UNIVERSITY OF THE WITWATERSRAND

JOHANNESBURG

SCHOOL OF MECHANICAL ENGINEERING

THE DEVELOPMENT OF AN ALGORITHM FOR THE CONTROL OF A SIX DEGREE OF FREEDOM MOTION SIMULATION PLATFORM

Michael John Acres

THE DEVELOPMENT OF AN ALGORITHM FOR THE CONTROL OF A SIX DEGREE OF FREEDOM MOTION SIMULATION PLATFORM

Michael John Acres

A dissertation presented in fulfillment of the requirements for the degree of master of science in engineering

Johannesburg, 1985

DECLARATION

I declare that this dissertation is my own unaided work. It is being submitted for the Master of Science in Engineering in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other University.

....

M J ACRES

22 day of August 1985.

ABSTRACT

An algorithm for the control of a six degree of freedom motion simulation platform is developed. The motivations for including motion cueing in a simulator are given. A model of a typical simulator is developed to investigate the performance of the motion filter. Aircraft motion is transformed, via the aircraft pilot and the simulator operator, into platform motion. The aircraft configuration is typical of a large passenger aircraft. The platform model is similar to the six axis synergistic platforms which are common on flight training simulators.

Inputs to the motion filter comprise the acceleration of the aircraft pilots centre of perception and a zero position command. Tilt generation is utilised to induce sustained translational cue perceptions. Investigations are conducted using forward acceleration and pitch acceleration inputs.

The correlation between the pilot and operator vestibular responses, as well as the movement of the actuators relative to their allowable envelopes, are investigated.

Recommendations for further work are given.

ACKNOWLEDGEMENTS

The author would like to thank Professor A Nurick for his help and supervision throughout this research project, and the National Institute for Aeronautics and Systems Technology for the funding of this project. He is grateful to Professor R Sivan for his suggestions for validating the results obtained and to John Hutchinson of Singer, Link-Miles Division for the information on training simulator motion platform dimensions. The author also expresses his thanks to Henriette du Plessis for her patient help in this typing effort.

CONTENTS PAGE

LIST OF FIGURES

NOTATION

LIST OF SYMBOLS

A - area (m^2)
AL - actuator length (m)
ALE - actuator length error (m)
B - actuator base mounting point (m)
FH - specific force input to translational washout filter (m/s^2)
FX - specific force on pilot (m/s^2)
FXH - high frequency component of FX (m/s^2)
FXL - low frequency component of FX (m/s^2)
g - gravity (m/s^2)
IP - input position (m)
kL - leakage coefficient (m^3/Pa)
L - transformation matrix (inertial to body)
OP - output position (m)
PM - actuator platform mounting point (m)
p - pressure (Pa)
p - roll rate (rad/s)
q - flow rate (m^3/s)
q - pitch rate (rad/s)
R - angular rate transformation matrix (inertial to body)
r - yaw rate (rad/s)
s - Laplace operator ($\sigma + j\omega$)
T - transformation matrix (body to inertial)
t - time (s)
x - x axis translation (m)
x_e - platform x translation error due to hydraulics (m)
$\ddot{x}p$ - pilot acceleration (x axis) (m/s^2)
$\ddot{x}_a$ - aircraft acceleration in the F R F (m/s^2)
x_s - platform x translation in inertial space (m)
y - y axis translation (m)
y_e - platform y translation error due to hydraulics (m)
y_s - platform y translation in inertial space (m)
yotoi - otolith response to aircraft motion input
yotoo - otolith response to platform motion input
yscci - semicircular canal response to aircraft motion input

y_{scco} - semicircular canal response to platform motion input

z - z axis translation (m)

z_e - platform z translation error due to hydraulics (m)

z_s - platform z translation in inertial space

α_a - angular orientation of aircraft in inertial space (rad)

α_s - platform attitude in inertial space without tilt = (θ_s, ϕ_s, ψ_s)

α_{st} - platform attitude in inertial space with tilt = (θ_{st}, ϕ_{st}, ψ_{st})

ζ - damping factor

ω - natural frequency (rad/s)

θ - pitch angle (rad)

θ_e - pitch error of platform due to hydraulics (rad)

θ_t - pitch tilt command (rad)

ϕ - roll angle (rad)

ϕ_e - roll error of platform due to hydraulics (rad)

ψ - yaw angle (rad)

ψ_e - yaw error of platform due to hydraulics (rad)

SUBSCRIPTS

1 - x axis

2 - y axis

3 - z axis

a - aircraft

p - pilot

o - operator

 - output

s - simulator platform

i - inertial

  - input

b - body

m - moving

r - rotation

t - translation

REFERENCE FRAMES

FRF - fuselage reference frame
- origin (CR) at centre of rotation of aircraft
- xa1 positive forwards along aircraft axis
- xa2 positive to the right when aircraft viewed from behind
- xa3 positive vertically down when aircraft viewed from behind

FIRF - fuselage inertial reference frame
- origin (CR) same as for FRF
- ai1,ai2 and ai3 have fixed orientation in inertial space with ai1 and ai2 in a horizontal plane

FFRF - fixed fuselage reference frame
- origin (S) at some fixed point on fuselage
- f1,f2 and f3 have the same orientation as FRF

ONRF - operator neutral reference frame
- origin (PN) at operator's centre of perception when platform is in its neutral position
- on1,on2 and on3 have the same orientation as NPLATRF

ORF - operator reference frame
- origin (P) at operators centre of perception
- o1,o2 and o3 have the same orientation as PLATRF

NPLATRF - neutral platform reference frame
- origin (NPC) at the centroid of the platform mounting points when in neutral position
- p1 positive forwards along platform axis in plane of mounting points
- p2 positive to the right (when viewed from behind) in plane of mounting points
- p3 positive downwards

PLATRF - platform reference frame
- origin (PC) as for NPLATRF but moves with platform
- same orientation as for NPLATRF but moves with platform

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND AND LITERATURE SURVEY

Flight simulators have applications in both the fields of aircraft design and development as well as flight crew training. Obvious benefits of training simulators are the substantial cost savings as well as the ability to subject the aircraft crew to numerous flight conditions, some of which might be potentially dangerous or even fatal in a real flight situation. Evolution of engineering flight simulators is providing analytical tools to evaluate design concepts and product improvements in existing aircraft with a degree of precision that helps in achieving the desired results.

Simulator installations generally include the following components, (a) aircraft dynamics model, (b) cockpit environment simulation including instruments, force feel control sticks and rudder pedals, communication systems and an audio simulator, (c) a visual display and (d) a motion platform. Training installations provide pilots with a simulation of an existing aircraft for either training, certification or refresher purposes, whereas in an engineering development simulator the pilot's reaction and response to the model provide information about the aircraft's acceptability and in later stages, constitutes a training facility for test pilots. Thus the realism required in simulators becomes apparent. The success of a flight simulator depends on the extent to which the operator can be led to believe that he is perceiving the real world. Under debate is the extent to which sensory cues need to be simulated to make this possible (Gundry [1], Stewart [2] and Perry and Naish [3], et al). There is consensus, however, that whatever cues are simulated, the performance of a simulator may have a lot to do with the quality of the cues provided. This is particularly true with regard to motion cues.

Problems which have had to be overcome to improve the qualities of sensory cues include the generation of detail and versatility in visual systems, as well as the investigation of the effects of motion cues and improving the performance of motion systems. Work in these fields has been carried out by a number of commercial simulator manufacturers as well as military and NASA establishments. Leaders in the field include Singer/Link-Miles Div, Rediffusion, CAE, General Dynamics and Rockwell International. Research carried out at the Technion, Israel Institute of Technology and the Massachussetts Institute of Technology into simulator motion and the optimal generation of motion cues must rank as among the most advanced to date.

The work in this thesis concerns some of the aspects relating to motion generation. An elementary model of a motion system is developed which includes the major motion transformations required to obtain platform motion as a function of the modelled aircraft's flight dynamics. The resultant platform motion is intended to supply the simulator pilot with the necessary motion cues to give the impression of real flight. The aim is to develop an understanding of the technology required in flight motion simulation and to develop a model which could be used for the investigation of different approaches to the transformation of real to platform motion.

Tremendous advances have been made in the field of aircraft simulation in both engineering and training applications. Developments in motion base technology have not been an exception in this regard and the capability exists of producing high fidelity motion on simulator installations. Linear washout filters (Schmidt and Conrad [4]), et al) with certain performance limitations have been replaced by adaptive and optimal control filtering techniques (Parrish and Martin [5] and LIU [6] et al). Various approaches have also been used in the development of motion base configurations. The number of degrees of freedom and excursion envelope are determined by the type of aircraft in question and the quality of motion required. According to Woomer and Williams [7], a full six degree of freedom motion system appears not to be only appropriate, but a necessity to VTOL trainers.

There is a vertical motion simulator, Figure 1.2.1(a), at the NASA AMES research centre which has a large amplitude motion in the vertical and one horizontal degree of freedom and more limited motion in the remaining four degrees of freedom. The United States Air Forces' flight dynamics laboratory at the Wright-Patterson Airforce base, Ohio, uses their LAMARS (large amplitude multimode aerospace research simulator) built by Northrop Corp as a primary tool in aeronautic research (Figure 1.2.1(b)). Work that can be performed includes pilot workload studies, aircraft handling quality studies and aircraft subsystem design development and integration. The majority of six degree of freedom motion platforms are of the type shown in Figure 1.2.1(c).

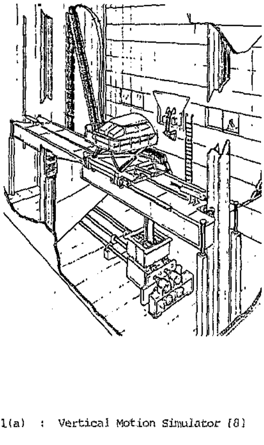

FIGURE 1.2.1(a) : Vertical Motion Simulator [8]

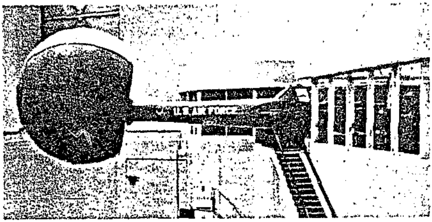

FIGURE 1.2.1(b) : Large Amplitude *Multimode* Aerospace Research Simulator [8]

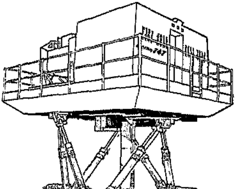

FIGURE 1.2.1(c) : Six Degree of Freedom Motion Platform [8]

According to Baret [9], this type of platform with hydrostatic bearings and hollow-rod jacks gives very smooth movement and high translational excursions due to the system geometry.

Work done at the University of Delft in Holland and at Parker-Hannifin USA has provided advanced hydrostatic actuators which yield very low friction, stiction and turn-around knock characteristics. This has considerably improved the quality of motion obtainable.

In flight, the pilot of an aircraft is subjected to the effects of a dynamic force environment which can be divided into two types. Firstly, there is disturbance motion, cues arising from sources which are external to the pilot control loop and secondly manoeuvre motion resulting from pilot initiated changes to the aircraft motion. Gundry [1] has shown that the motion of the simulator, as a function of disturbance and manoeuvre motion, allows the operator to control the simulator using similar sensory cues and skills to those that he would use in flight. This in effect allows the pilot to generate more lead and gain than for visual cues alone. Figures 1.2.2 and 1.2.3 (from Perry and Naish [3]) indicate the effect of motion cues on control response. Figure 1.2.2 (a) shows the response of the aircraft with no pilot action and (b), (c) and (d) each show a number of time histories when the pilot was attempting to control the same situation with differing simulation cues available to him. Figure 1.2.3 shows the reduction in amplitude of bank disturbances when motion cues are available to the operator. This is similar to observations by Woomer and Williams [7] indicating that confusion of apparent motion is experienced on simulated shipboard motion without the motion system activated and only a visual cue available. Hall [10] concluded that for the prediction and evaluation of the handling qualities of an aircraft using a piloted simulator it may not be sufficient for the pilot to achieve a similar performance in the simulator as in flight but it is also necessary that the pilot should adopt the same control strategy. To achieve this it has been found essential to provide the pilot with motion cues as no substitute in these circumstances has yet been found. Previous studies (Perry and Naish [3]) have shown also that the need for motion cues in simulation depends to a large extent on the control and response characteristics of the aircraft in question.

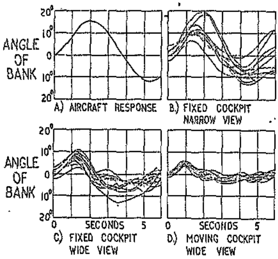

FIGURE 1.2.2 : *Control of Aircraft Response to a Step Side-Gust with Different Simulation Cues*

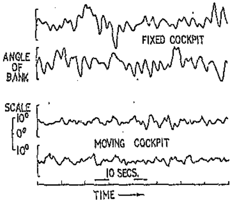

FIGURE 1.2.3 : Effect of Cockpit Movement on Control in Turbulence

It is obvious from the above that it is pertinent to include motions cues into the realms of flight simulation techniques.

1.2 OBJECT

The object of this research is to develop an understanding of the principles of motion cue generation for aircraft flight simulators by developing a software model of a simulator motion platform as well as the proposal and investigation of a possible motion filter. The model is simple and includes only essential cues and responses necessary for adequate simulation. Of interest with regard to the performance of the motion filter is whether the motion generated remains within the envelope of platform motion and whether the vestibular response of the operator is satisfactory when compared to the vestibular response of the pilot. To avoid ambiguity the aircraft pilot will is referred to as the 'pilot' and the simulator pilot as the 'operator'.

In designing the model a generic approach has been adopted so that the configuration of the component elements can be altered for further research if required. For example, research into various actuator types can be done to determine the effect of friction and stiction on operator vestibular response or the effects of changing the platform motion envelope.

1.3 SCOPE

For the purposes of this research the platform geometry has been modelled on the six actuator, six degree of freedom synergistic motion system type (Figure 1.2.1(c)), which is probably the most common six DOF type currently in use. Dimensional specifications for the model have been obtained from a Link-Miles platform such as the one in use at the South African Airways training facility. This type of motion system has been very successfully used, particularly for training simulators. Among the more advanced research simulators the tendency is to have at least one degree of translational motion extended in its envelope and often fewer than 6 degrees of freedom. This is necessary to enable more dramatic motion cues which are necessary in research environments for particular applications to be included. In the commercial training field the motion cue generation is approached less rigorously. Here motion is required to augment the reality of the simulator environment whereas motion in research facilities is used to give the pilot specific cues related to workload levels in, and stability characteristics of, the aircraft.

Implications of the above on the evaluation of the proposed motion generating filter are that the nett gains of the motion filters are lower than might be required in the case of motion for a research simulator because of the typically reduced motion envelope.

Figure 1.3.1 shows the general approach used by Link-Miles for the motion control on one of their simulators.

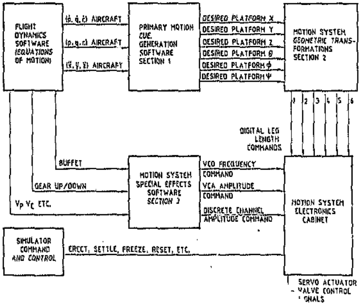

FIGURE 1.3.1 : Typical Motion Control System for Flight Trainer [2]

Provision is made for the inclusion of special effects dynamics, including vibration due to engines, turbulence and ground effects. These, however, are not of interest in this study. Of interest in Figure 1.3.1 are the blocks indicating primary motion cue generation, motion systems geometric transformations and motion system dynamics and control.

1.4 MOTION ALGORITHM EVALUATION CRITERIA

For motion cues to contribute to the simulation of the environment they must have a similar influence on the simulator operator as the real world motion would have on the aircraft pilot. This means that there should be a transferability of piloting experience from a simulator into the real world situation. The success of this transfer between the simulator and the real world, is therefore, dependent on the motion filtering technique used. This in turn is also influenced by the motion capabilities available on the simulator. Using a motion based simulator, comparisons between simulations with and without motion are possible (Perry and Naish [6]). Indications can be gained of the relative success of the motion generation algorithms used by evaluating these results. An even better valuation is possible if the correlation between these results and the real would can be investigated.

In this research however, no simulator hardware is available on which to conduct any experimentation. Evaluation of the algorithm is therefore done by comparing the vestibular responses of the modelled simulator operator and the aircraft pilot as well as monitoring the platform movement relative to its envelope limits.

No quantitative analysis technique has been developed with which to accurately predict the suitability of a motion generating algorithm by comparing the vestibular responses of a simulator operator and an aircraft pilot. However, the following criteria can be used to predict the suitability or otherwise of a motion generating algorithm.

(i) The cue onset in the real world and in the simulator must be synchronised.

(ii) The direction of the cues must be the same. If they are not the same then it is possible that the simulator operator may receive motion cues which are in conflict with his other cues.

The negative effects of this type of discrepancy are dependent on both the magnitude and duration of the conflicting cue.

(iii) Intended motion cues must be of such magnitudes that they are perceived by the simulator operator. Cue perception is both a function of magnitude and the number of other cues which are presented to the subject. Therefore, threshold values are usually related to some perception probability and are not absolute.

CHAPTER TWO

THE DEVELOPMENT OF A GENERAL PURPOSE SIMULATOR MOTION SYSTEM MODEL

2.1 INTRODUCTION

This chapter involves the development of the computer model of the flight simulator motion system. To facilitate the development, a modular approach has been adopted. Each of the constituent blocks was defined at the outset and then the required interface variables were determined. Thus, each element of the model could be developed in isolation and then be integrated to form the final model. Figure 2.1.1 shows the model flow diagram and indicates the interface variables between each of the blocks.

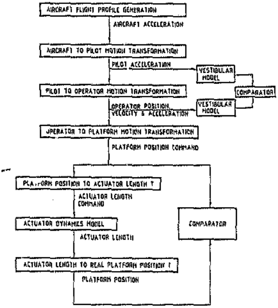

FIGURE 2.1.1 : Simulator Motion System Model Structure

Aircraft motion, in six degrees of freedom, is translated into eventual platform motion which is in turn transformed into the lengths of the six platform supporting actuators. The configuration of the platform to be modelled is shown in Figure 2.1.2. Control of the platform is by means of actuator length commands. A model of a typical actuator and an hydraulic control system, yields an actuator length output as a result of the input length command signal. The output actuator lengths are then operated on to yield the position the platform would assume. The resultant platform position can then be compared to the control position for evaluation of the platform, actuator and controller dynamics.

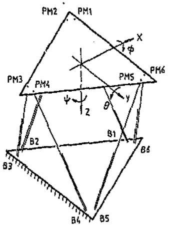

FIGURE 2.1.2 : Notation and Configuration of Modelled Platform

Chapter Two outlines the development of the mathematical models and transformations and Chapter Three gives details of the physical properties and dimensions of the particular system that has been modelled.

The numerical implementation of the model as a computer program is shown in Appendix D of this report. The model is written using ACSL [11] (Advanced Continuous Simulation Language). This language was designed for the modelling and evaluating of continuous time dependent systems and it allows the display, in graphic or tabular form, of any combination of variables as a function of time. It is a high level language and is thus optimised for accuracy and not for speed and as such does not conduct the simulation in real time. FORTRAN subroutines are employed to do some of the purely mathematical transformations.

The reference frames are right handed with positive rotation about an axis defined by the right hand screw rule and the first axis refers to the x-axis, the second axis to the y-axis and the third axis to the z-axis in a Cartesian axis system. Rotations about the three axes are defined as (a) roll (phi)-x axis, (b) pitch (theta)- y axis and (c) yaw (psi)-z axis.

2.2 AIRCRAFT FLIGHT PROFILE GENERATION

In order to evaluate the motion transformation and filtering, the response of the model to step inputs will be investigated. Details will be given in Chapter Four.

2.3 AIRCRAFT TO PILOT MOTION TRANSFORMATION

Aircraft motion defined in terms of translational accelerations of the aircraft in the FRF and the angular accelerations of the FRF with respect to the FIRF are transformed into aircraft pilot motion with respect to the FRF. Account is thus taken of the displacement of the pilot from the centre of rotation of the aircraft, which induces pilot translations as a result of the aircraft rotations.

Provision is made in the model for including relative motion between the centre of rotation of the aircraft and the fuselage. This occurs when an aircraft's mass distribution changes in flight as might be the case when fuel is consumed in flight or when some of the aircraft's payload is released.

The pilot's position in the fuselage reference frame is defined as follows :

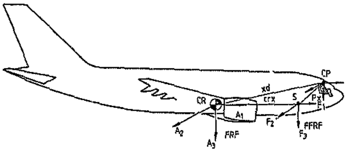

FIGURE 2.3.1 : Spatial Relation Between the Pilot, Centre of Rotation of the Aircraft and the Fuselage

where S is some arbitrarily defined point fixed relative to the fuselage. The centre of rotation of the aircraft (CR) is a distance |crx| from S in the FFRF and the pilot's centre of perception (CP) is a distance |px| from S in the FFRF. Thus

$$[xd] = [px] + [crx] \qquad (2.3\text{-}1)$$

where |xd| is the distance between the pilot's centre of perception and the aircraft's centre of rotation. Vector px is fixed relative to the FFRF while crx describes the position of CR with respect to S at any particular time.

The aircraft and pilot are defined in the inertial reference frame as follows.

Here xd defines the position of CP in the FRF and phi (ϕ), theta (θ) and psi (ψ) define the angular position of the FRF in the FIRF.

Using equation (B-5) from Appendix B the following expression *for* the pilot acceleration in the FRF, as a function of the aircraft velocity and acceleration in the FRF, is obtained.

$$\ddot{xp} = \ddot{xa} + \ddot{xd} + 2.w.\dot{xd} + (\dot{w} + w.w)xd$$

$$= \ddot{xa} + \ddot{crx} + 2.w.\dot{crx} + (\dot{w} + w.w)(crx + px) \qquad (2.3\text{-}2)$$

Here, from Etkin *(20)*,

$$w = \begin{bmatrix} 0 & -r & p \\ r & 0 & -q \\ -p & q & 0 \end{bmatrix} \qquad (2.3\text{-}3)$$

where

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & \sin\phi \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} . \begin{bmatrix} \dot{\phi} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \qquad (2.3\text{-}4)$$

2.4 *PILOT TO OPERATOR MOTION TRANSFORMATION*

Chapter Four details the investigation of the Pilot to Operator motion filtering.

2.5 OPERATOR TO PLATFORM MOTION TRANSFORMATION

Simulator operator motion, both translational and rotational, is transformed to give platform motion. The calculated platform motion will thus result in the required operator motion.

The relevant axes are defined as follows :

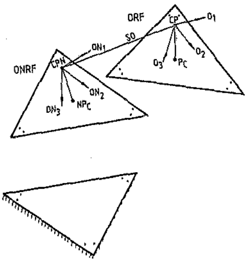

FIGURE 2.5.1 : Operator and Operator Neutral Reference Frames

In the above Figure CP is the operator's centre of perception and CPN is the position of CP when the platform assumes its neutral position. PC is the centre of the platform (geometric centre of the six platform mounting points) and yd describes the position of PC with respect to CP in the ORF.

Operator *motion* (so) from the neutral position (CPN) results in a translation and rotation of PC in the ONRF. The following equations describing the motion of PC with respect to the platforms neutral position in the ONRF.

$$p = T(yd) + xs \tag{2.5-1}$$

To obtain the movement of the platform from its neutral position, the displacement of PC from CP (yd) must be subtracted from equation (2.5-1), which yields;

$$p = T(yd) + xs - yd \tag{2.5-2}$$

where T is the transformation matrix *from* body to inertial axes (Equation (A-6), Appendix A).

2.6 PLATFORM MOTION TO ACTUATOR LENGTH TRANSFORMATION

Platform position is controlled by regulating the lengths of the six platform supporting actuators. Thus, the platform motion from its neutral position will need to be transformed into actuator lengths. The platform neutral reference frame has its origin at neutral platform position centre (NPC). Platform translation p from NPC in the NPLATRF and platform roll, pitch and yaw result in a translation of the actuator mounting points PM1 to PM6 with respect to their neutral positions in the NPLATRF (see Figure 2.6.1).

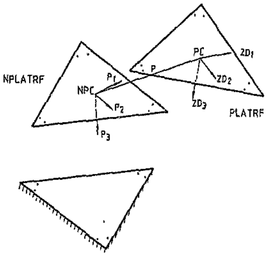

FIGURE 2.6.1 : Platform and Platform Neutral Reference Frames

The movement of each platform mounting point in the NPLATRF is expressed as follows :

$$PM = T(zd) + p \qquad (2.6\text{-}1)$$

where PM is the position of a m unting point in the NPLATRF and zd is the position of a moun'ing point in the PLATRF. Equatic : 2.6-1 is applied for each actuator mounting point (PM1 to PM6).

The base mounting point positions (B1 to B6) are defined in the NPLATRF. Actuator lengths are obtained by determining the distance between their respective base and platform mounting points in the NPLATRF.

Thus for each of the six actuators :

$$ali = \{ \sum_{n=1}^{3} (Bi_n - PMi_n)^2 \}^{1/2} \qquad (2.6\text{-}3)$$

where $i = 1$ to 6.

The subscripts 1,2 and 3 refer to the coordinates along the p1,p2 and p3 axes in the NPLATRF and i refers to the i'th actuator (i=1 to 6).

2.7 ACTUATOR CONTROL MODEL

To approximate a sophisticated hydrostatic actuator an idealised approach has been adopted in setting up the actuator control model. The servo valve is considered to be linear and the actuator model includes only the effects of viscosity and leakage (Schwarzenbach and Gill [12]). Effects such as compressibility, inertia and the possibility of the actuator being assymetrical have been ignored as the actuator control would become significantly more complex if they were included. This added complexity would have has no real influence on the rest of the model.

The actuator model is arrived at as follows ;

$$q(t) = A.\frac{dx(t)}{dt} + kl.p(t)$$

and $$p(t).A = \mu.\frac{dx(t)}{dt} \quad \text{where } \mu \text{ is a constant}$$

$$\therefore q(t) = A.\frac{dx(t)}{dt} + kl.\frac{\mu}{A}.\frac{dx(t)}{dt} \qquad (2.7\text{-}1)$$

Taking the Laplace transform of 2.7-1 and rearranging, gives the transfer function;

$$\frac{X(s)}{Q(s)} = \frac{1.0}{(A + \frac{kl.\mu}{A})\, s} \qquad (2.7\text{-}2)$$

where	A	=	area of piston	(m^2)
	kl	=	leakage coefficient and	(m^3/Pa)
	μ	=	viscosity of oil.	(N/m)
	q	=	hydraulic flow rate	(m^3/s)
	p	=	fluid pressure	(Pa)
	x	=	actuator displacement	(m)

A linear gain (kv) for the hydraulic servo valve has been assumed. The controller is a position control loop. Figure 2.7.1 shows the general arrangement of the control loop.

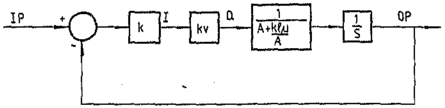

FIGURE 2.7.1 : Actuator Model

Thus the model transfer function in terms of the input position signal (ip) and the output position (op) is as follows;

$$\frac{OP(s)}{IP(s)} = \frac{1}{\frac{(A + kl.\mu/A)s}{k.kv} + 1} \qquad (2.7\text{-}3)$$

where k is the gain relating loop position error and value input current

ie k = mA/m and
kv = m^3/s/mA

Thus, the input to this model is an actuator length command and the output is the actual actuator length.

2.8 REAL ACTUATOR LENGTH TO REAL PLATFORM POSITION TRANSFORMATION

An inverse axis transformation is now performed on the real actuator lengths to obtain the real platform position. The difference between the required and actual platform positions results from the control system characteristics as well as the physical properties of the platform such as its inertia and actuator friction etc.

2.8.1 SETTING UP OF EQUATIONS

The six input actuator lengths are known as well as the base mounting point coordinates in the NPLATRF. To determine the platform position the actuator platform mounting points are required. There are six mounting points and therefore eighteen unknowns (6 points * 3 co-ordinates), thus requiring eighteen equations for a solution. Thus taking into account the facts that

(i) the actuator lengths are known
(ii) the mounting points must remain coplanar
(iii) and the distances between the mounting points on the plane are

known, the equations are constructed as follows :

Actuator Lengths

Six actuator mounting points in the NPLATRF are required to specify the lengths of the six actuators as the base mounting point coordinates remain fixed. Equation 2.8-1 expresses the actuator length in terms of the base and platform mounting point coordinates. Six equations are derived in this manner, one for each actuator. The equation is of the form

$$alri = \{ \sum_{n=1}^{3} (Bi_n - PMi_n)^2 \}^{1/2} \qquad (2.8\text{-}1)$$

where i = 1 to 6.

Coplanar Nature of Points PM1 to PM6

All of the six mounting points lie on the same plane. Any three points can be made to be on the same plane, here PM1, PM2 and PM3 are taken to be coplanar. Now consider the followin matrix :

$$A = \begin{bmatrix} PM1_1 & PM1_2 & PM1_3 & 1 \\ PM2_1 & PM2_2 & PM2_3 & 1 \\ PM3_1 & PM3_2 & PM3_3 & 1 \\ PMi_1 & PMi_2 & PMi_3 & 1 \end{bmatrix}$$

where PMi is the fourth point on the actuator mounting plane. The condition for the four points (PM1, PM2, PM3 and PMi) to be coplanor is that

$$\text{Det}(A) = 0. \qquad (2.8\text{-}2)$$

Thus, there are three equations resulting from the coplanar condition, (i) PM1, PM2, PM3 and PM4 coplanar, (ii) PM1, PM2, PM3 and PM5 coplanar and (iii) PM1, PM2, PM3 and PM6 coplanar.

Distance Between the Platform Mounting Points

Equations (2.8-1) specify the distance of each mounting point from its base point, equations (2.8-2) specify the coplanar nature of the six points. Now the equations specifying the distance between the mounting points on the mounting plane are required. To uniquely specify the position of six points on a plane, nine distances are required. Each point PMi is specified by its distance relative to three other points. If only distances from two points are used then there are two possible positions for the third point, the alternative point being the mirror image of the first about a line through the two base points. The following set of equations uniquely relates each of the six points to three other points.

$$\begin{aligned} dm12 &= PM2 - PM1 \\ dm13 &= PM3 - PM1 \\ dm15 &= PM5 - PM1 \\ dm24 &= PM4 - PM2 \\ dm26 &= PM6 - PM2 \\ dm34 &= PM4 - PM3 \\ dm35 &= PM5 - PM3 \\ dm46 &= PM6 - PM4 \\ dm56 &= PM6 - PM5 \end{aligned} \qquad (2.8\text{-}3)$$

where,

$$dmij = \left(\sum_{n=1}^{3} (PMj_n - PMi_n)^2 \right)^{1/2} \qquad (2.8\text{-}4)$$

with dmij being the distance on the mounting plane between points PMi and PMj.

Now the eighteen equations in eighteen unknowns can be solved to determine the platform mounting point coordinates in the NPLATRF.

2.8.2 SOLUTION OF EIGHTEEN SIMULTANEOUS EQUATIONS

The truncated Taylor Series (Appendix C) expansion of equations 2.8-1, 2.8-2 and 2.8-4 are combined to generate a set of eighteen simultaneous equations of the form:

$$[CM(18,18)].[E(18)] = [RHS(18)] \qquad (2.8\text{-}5)$$

where E(18) represents the error between the current root estimante and the predicted solution, CM(18,18) is the coefficient matrix and RHS(18) is the right hand side of the equation. The iteration is continued until the error is within the required limits and then the final estimate of the roots is obtained by adding the error to the last root estimate.

The following matrix equation is obtained.

$$[f_{ni}]\,[r_i - g_i] = [-f_n] \qquad (2.8\text{-}6)$$

Here r_i is the predicted ith root and g_i is the estimated ith root. Also f_n is the nth function evaluated at g_i and f_{ni} is the partial derivative, with respect to the ith root, of the nth function evaluated at g_i.

Solution of the matrix of linearised equations is performed using the Gauss Elimination method with partial pivoting (Gerald [13]). The augmented coefficient matrix B (equation 2.8-7), with the right hand side of equation 2.8-6 forming the 19'th column, is reduced by locating the pivot in each column and placing it on the diagonal and then reducing the elements below the pivot. The solution is placed in the augmentation column of the coefficient matrix. Checks to monitor the size of pivot elements are also conducted

Here,

$$B = \begin{bmatrix} \partial f1\ 1 & \partial f1\ 2 & \ldots & \partial f1\ 18 & : & -f1 \\ \partial f2\ 1 & . & \ldots. & & : & -f2 \\ . & . & \ldots. & & : & . \\ . & . & \ldots. & & : & . \\ \partial f181 & \partial f182 & \ldots & \partial f1818 & : & -f18 \end{bmatrix} \qquad (2.8\text{-}7)$$

2.8.3 DETERMINATION OF THE PLATFORM POSITION AND ORIENTATION

The centre of the platform (CP) is determined with reference to the NPLATRF. This is subtracted from the center of the platform when in the neutral position, CPNEUT, to give the translation of the platform. Using points PM1,PM3,PM4 and PM6 the pitch, roll and yaw attitudes of the platform are calculated.

2.9 SPECIFICATION OF MODEL PARAMETERS

Constants used thus far in the model development will be given values and the model configuration will be fixed. Data given, approximates that which is typical of most of the operational simulators of this type.

2.9.1 AIRCRAFT AND SIMULATOR PLATFORM GEOMETRIC SPECIFICATIONS

Data used for the aircraft is typical of a large transport type of aircraft. The point S is taken to be at a point at the rear of the cockpit. The centre of rotation of the aircraft (CR) is at a distance

crx = (-20,0; 0,0; 0,0)

meters from S in the FFRF. The pilot's position in the FFRF is at a distance

px = (1,5; 0,0; -1,0)

meters from point S.

The platform dimensions are as follows,

actuator length	min	=	2,430m
	max	=	3,950m
	neut	=	3,160m
	stroke	=	1,520m

with the coordinates, in meters. The actuator base and platform mounting points with the platform in the neutral position, in the neutral platform reference frame are

PM1 = (+1,126 ; -1,696 ; 0,000)
PM2 = (+0,906 ; -1,823 ; 0,000)
PM3 = (-2,031 ; -0,127 ; 0,000)
PM4 = (-2,031 ; +0,127 ; 0,000)
PM5 = (+0,906 ; +1,823 ; 0,000)
PM6 = (+1,126 ; +1,696 ; 0,000)

2.9.1 AIRCRAFT AND SIMULATOR PLATFORM GEOMETRIC SPECIFICATIONS

Data used for the aircraft is typical of a large transport type of aircraft. The point S is taken to be at a point at the rear of the cockpit. The centre of rotation of the aircraft (CR) is at a distance

crx = (-20,0; 0,0; 0,0)

meters from S in the FFRF. The pilot's position in the FFRF is at a distance

px = (1,5; 0,0; -1,0)

meters from point S.

The platform dimensions are as follows,

actuator length	min	=	2,430m
	max	=	3,950m
	neut	=	3,160m
	stroke	=	1,520m

with the coordinates, in meters. The actuator base and platform mounting points with the platform in the neutral position, in the neutral platform reference frame are

PM1 = (+1,126 ; -1,696 ; 0,000)
PM2 = (+0,906 ; -1,823 ; 0,000)
PM3 = (-2,031 ; -0.127 ; 0,000)
PM4 = (-2,031 ; +0,127 ; 0,000)
PM5 = (+0,906 ; +1,823 ; 0,000)
PM6 = (+1,126 ; +1,696 ; 0,000)

and

B1 = (+2,310 ; -0,165 ; +2,500)
B2 = (-1,012 ; -2,084 ; +2,500)
B3 = (-1,298 ; -1,919 ; +2,500)
B4 = (-1,298 ; +1,919 ; +2,500)
B5 = (-1,012 ; +2,084 ; +2,500)
B6 = (+2,310 ; +0,165 ; +2,500)

The operator's position on the platform is described by

<u>yd</u> = (0,0; 0,0; 3,0)

2.9.2 *ACTUATOR MODEL PARAMETERS*

For the purpose of this model the following values for the model have been used.

A = 0,0025 (m^2)
k = 20,0 (mA/m)
kv = 0,0038 ($m^3/s/mA$)
kl = $1,5 \times 10^{-11}$ (m^3/Pa)
and μ = 10,0 (N/m)

Figure 2.9.1 gives the step response of the model. MIL-STD-1588 [14] specifies that for a simulator motion system that there is no overshoot for a square wave input signal of magnitude equal to 0,05* (maximum command signal). This model meets this specification.

The position input command *IP* has the value
IP = 0,075 m.

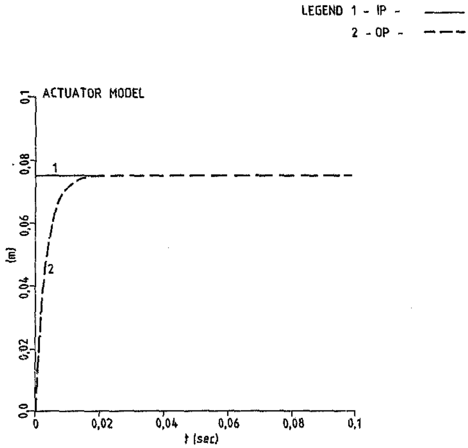

FIGURE 2.9.1 : ACTUATOR MODEL STEP RESPONSE.

CHAPTER THREE

INVESTIGATION OF PILOT TO OPERATOR MOTION TRANSFORMATION

3.1 INTRODUCTION

As previously stated, the major concern in providing motion is the prevention of false cues. Much of the undesirable motion in simulators stems from the mechanical system. These effects have been ignored with the emphasis being placed on the generation of the required cues.

The development of a filter which transforms aircraft pilot motion into simulator operator motion is outlined in the remainder of this section.

In generating motion for a simulator platform the following are some of the points which must be borne in mind to ensure the success of simulator motion cues :

(i) the platform must be prevented from running into its mechanical stops, as this detracts tremendously from the simulation fidelity,

(ii) the motion cues must have the correct sense in order to prevent the operator from becoming confused,

(iii) the cues must be washed out in such a way that the operator remains unaware of the difference between his realisation of the motion and the sensation which would be experienced in the real world and

(iv) the motion must be washed out so that subsequent motion cues can be accommodated even if they are in the same direction.

Validation of the filter in this exercise will be done by (i) determining the platform motion to establish whether it remains inside the envelope allowed by the actuators and (ii) by observing the correlation between the vestibular responses of the pilot and operator to their respective motions to ensure that the operator receives similar cues to the pilot.

3.2 PILOT TO OPERATOR MOTION TRANSFORMATION DEVELOPMENT

The motions of interest are the translation accelerations $\ddot{x}$, $\ddot{y}$ and $\ddot{z}$ and the rotational accelerations $\dot{p}$, $\dot{q}$ and $\dot{r}$ of the pilot and the resultant motion of the platform as seen by the operator.

In order to achieve the desired motion cue generation the motion filter must have the following characteristics :

(i) accelerations must be washed out to limit travels required and
(ii) the platform must be driven by a zero position signal to some neutral position.

An approach which is widely used to conserve travel requirements and generate sustained cues is to provide tilt angles, which are a function of sustained translational accelerations, and thereby use gravity to enhance the realism of the cues. Therefore, the motions in surge and pitch, sway and roll and heave and yaw are associated with one another. The motion filter will be developed for the pitch/surge axes and will then be expanded to include the other axes. The vestibular model which will be used to evaluate the filter makes no distinction between the three different axes for either the semicircular canals or otoliths. The differences between the pitch/surge, roll/sway and heave/yaw filters are limited only to sign changes and not dynamic characteristics.

3.2.1 PITCH/SURGE MOTION FILTER

Figure 3.2.1 shows the block structure of the Pitch/Surge motion filter.

The specific force, fx, in the pilot's body reference frame is filtered to provide the high and low frequency components. The high frequency component of fx, fxh, is used to generate the translational platform acceleration commands and the low frequency component of fx,fxl, is used to generate the platform tilt command. The specific force is calculated by adding the pilots body reference frame accelerations to the gravity components, which result from the aircrafts attitude in inertial space (α_a).

Thus,

$$fx = \dot{x}\dot{p} + [L]_{\alpha a} \cdot g \qquad (3.2\text{-}1)$$

where L is defined by equation (A-3) as given in Appendix A. The high frequency component fxh is multiplied by a gain KAT and has the gravity component effects of the simulator platform attitude without the tilt component (α_s) is subtracted from it before feeding into the position command loop. This loop is driven by a zero position command (xo) and the modified fxh, fh, is fed in as a disturbance signal. The output position from this controller is transformed from body to inertial coordinates by the transformation

$$x_s = [T]_{\alpha_{st}} \cdot x \qquad (3.2\text{-}2)$$

where T is defined by equation (A-6) Appendix A and α_{st} is the simulator's attitude in inertial space.

In order to generate the tilt position command (θ_t) the low frequency component of fx, fxl ,is multiplied by the gain KT, has its sign corrected by KI and is then multiplied by 1/g, as follows;

$$\theta_t = KT.\ KI.\ \frac{Fxl}{g} \qquad (3.2\text{-}3)$$

The rotational pitch acceleration $\dot{q}$ is filtered and the high frequency component (after being multiplied by KAR), is integrated to give the angular rate q. This is transformed from body to inertial axes and is then integrated to obtain the angular position of the simulator Θ_s. The transformation is as follows;

$$q = KAR.\int \dot{q}dt$$

$$\dot{\Theta}_s = [R]_{\alpha_s}.q \quad (3.2\text{-}4)$$

where

$$R = \begin{bmatrix} 1 & \sin\phi\tan\Theta & \cos\phi\tan\Theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\Theta & \cos\phi\sec\Theta \end{bmatrix} \quad (3.2\text{-}5)$$

The pitch tilt command obtained from the translation acceleration requirement is added to Θ_s to obtain the final platform angular position command,

$$\Theta_{st} = \Theta_s + \Theta_t \quad (3.2\text{-}6)$$

In support of the above technique, Stewart [2], indicates that it is possible to use a high frequency component of rotational acceleration to present rotational acceleration and a low frequency component of rotational acceleration to simulate translational accelerations by a component of gravity. Other references, Sivan [15] and Dustebury and White [16] et al, indicate that this is suitable for motion cue generation. However, the position washout loop which treats the acceleration as a disturbance is not included in the references.

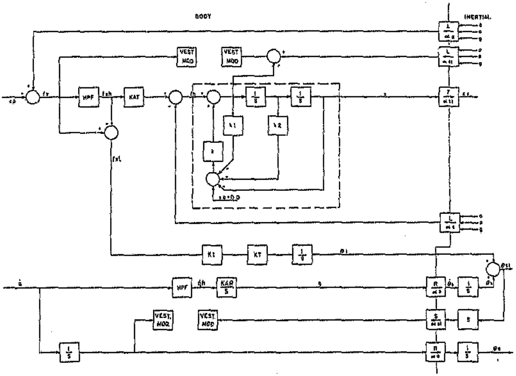

FIGURE 3.2.1 : *Pitch/Surge Motion Filter Structure*

Stewart [2], indicates that the break frequencies for the high and low pass filters could be 0,3rad/s and 0,25rad/s. Dustebury and White [16], indicate that the high pass break frequency of 0,7rad/s represents an upper limit and also that attenuating factors of 0,6 (gains KAT and KAR above) are acceptable.

A second order high pass filter has been used to generate the high frequency components of the accelerations and the low frequency component of fx is obtained by substracting fxh from fx. Initial values of ω = 0,4rad/s, ξ = 0,8 and KAT=KAR=0,6 are used. A filter of lowest possible order is desirable to minimize degradation of motion due to phase lead (Sivan [15]).

Thus the filter used is

$$\frac{FXH(s)}{FX(s)} = \frac{s^2}{s^2 + 2\xi\omega s + \omega^2} \qquad (3.2\text{-}7)$$

and $$fxl = fx - fxh. \qquad (3.2\text{-}8)$$

The high pass filters washout the specific force and rotational accelerations to zero and thus fulfill the first requirements stated above. Alone, this washout would yield a zero velocity state, but some displacement of the platform from the neutral position would result. The second filter in the translational controller then washes out this position to zero. This filter (Figure 3.2.2) treats the zero position signal as the command signal and the acceleration signal as a disturbance.

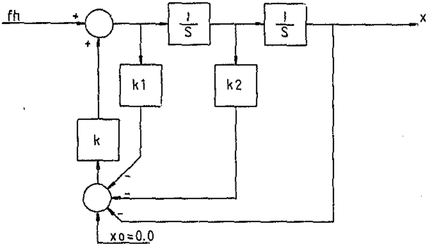

FIGURE 3.2.2 : Translational Position Washout Filter

The transfer functions of the translational position washout filters are;

$$\frac{O(s)}{I(s)} = \frac{k}{(1 + k\,k_1)\,s^2 + k\,k_2\,s + k} \qquad (3.2\text{-}9)$$

and

$$\frac{O(s)}{D(s)} = \frac{1}{k}\,\frac{k}{(1 + k\,k_1)\,s^2 + k\,k_2\,s + k} \qquad (3.2\text{-}10)$$

Rewriting the above equations in terms of the filter's natural frequency and damping ratios, the following is obtained;

$$\frac{O(s)}{D(s)} = \frac{1}{k}\,\frac{O(s)}{I(s)} = \frac{1}{k}\,\frac{Wn^2}{s^2 + 2\xi Wn s + Wn^2} \qquad (3.2\text{-}11)$$

giving $k_1 = \frac{1}{k} \left(\frac{k}{W_n^2} - 1\right)$

and $k_2 = 2\xi/Wn$

where the gain k can be used to bias the output response magnitude towards the disturbance input. An increase in the damping ratio will cause the motion to be overdamped. Alternatively a decrease will cause oscillations which the operator might pickup and interpret incorrectly. An initial value of $\xi = 0.8$ and $k = 1.0$ has been used. In determining the break frequency, account must be taken of the high pass filter through which the disturbance command has been passed. The higher the frequency, the sharper the resultant motion will be. To avoid attenuating the full range of motion, this frequency must not be less than that of the high pass filter break frequency. A value of Wn = 1.0 rad/s^2 has been chosen.

In deriving the above filter approach it has been assumed that the pilot is insensitive to the angular attitude of the aircraft in flight. This is true for yaw and is an approximation for pitch and roll. Therefore, it can be assumed that any permanent pitch or roll attitude can be employed to generate the sustained acceleration as discussed.

Thus far the argument has been developed using the translational acceleration cue as a baseline and then transferring to the rotational axis. Later in the report the results of a combined angular and translational acceleration input, used to excite the filter, will be commented on.

3.2.2 ROLL/SWAY AND HEAVE/YAW FILTER

The same filter has been used for these modes as for the pitch/surge mode. The differences are in the signs of some of the commands and in the case of the heave/yaw motion there is no cross coupling between the heave and yaw degrees of freedom. Thus, KT = 0,0 for the heave/yaw filter. Also, the specific force in the heave direction is taken to be the heave acceleration without adding the effect of gravity. If this were not done, situations would arise where large travels would be used in simulating the force of gravity. For example if some pitch tilt angle is implemented the platform would move to maintain a one g gravitational acceleration in the pilot's z axis.

In calculating the specific force on the pilot along the surge axis, a positive pitch induces a positive specific force component which is added to the surge acceleration. A positive roll however, induces a negative specific force component and therefore has to be subtracted from the sway acceleration to obtain the correct specific force.

Gravity, which acts in the positive direction is taken to produce a negative reaction in the z direction on the pilot and is therefore specified as -g in the model.

3.3 VESTIBULAR MODEL

The vestibular model of Sivan and Talwar [17] was used to evaluate the success or otherwise of the filtering technique. Acceleration along the three translational axes and velocity in the three angular directions about these axes, are sensed by the human vestibular system. Otolith sensors sense translational acceleration in the body axes and the rotational velocities are sensed by the semicircular canal sensors. No distinction has been made between any of the otolith or any of the semicircular canal transfer functions for the three respective axes.

According to Stewart [2] there is no significant difference between the perception of rotation thresholds for the x,y or z axes.

The transfer functions are as follows (Sivan and Talwar [17] :

$$Y\,oto(s) = G_o \frac{s + a_o}{s + b_o} A\,(s) \qquad (3.3\text{-}1)$$

where a_o = 0,076rad/s
b_o = 0,19rad/s
G_o = $2,16S^2/m$ and

Yoto (t) is the otolith response to input acceleration a(t) in the body reference frame.

$$Yscc(s) = G_s \frac{s}{s + b_s} \, Q(s) \qquad (3.3\text{-}2)$$

where G_s = 39,51s/rad
b_s = $0,169s^{-1}$ and

yscc(t) is the semicircular canal response to the angular velocity q(t) with respect to the body reference frame.

Inputs to the vestibular models are obtained firstly from the aircraft input motion and secondly from the resultant platform motion. The vestibular model outputs are expressed as

yotoi (t) = f (specific force on pilot)
yscci (t) = f (aircraft rotational velocities) (3.3-3)
yotoo (t) = f (specific force on operator) and
yscco (t) = f (simulator rotational velocities).

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 INTRODUCTION

Selected results from both the motion filter model, Appendix D.2, and the simulator platform model Appendix D.1 are given in Appendix E. Two input signals were used to drive the models. The first is a translational (surge) acceleration input and the second is a rotational (pitch) acceleration input. This enables conclusions to be drawn as to the effect on the operator of the cross coupling between the surge, heave and pitch axes which is a result of the aircrafts geometry and motion filter tilt angle generation. In order to enable comment on the suitability of the proposed washout scheme in terms of the simulator motion envelope restriction the translational and rotational acceleration inputs are combined and the results plotted. As has been mentioned, the investigations have been limited to include only surge and pitch acceleration inputs.

In order to investigate the influence of the actuator characteristics, on the performance, plots to show the actuator length and operator position error with respect to the desired input commands are also included in Appendix E.

Both the input acceleration commands are step functions shaped generated by a second order low pass filter with a break frequency of ω_b = 3,16 rad/s and a damping factor of ξ = 0,8. The magitudes of the translational and rotational inputs are representative of the accelerations that might be experienced on a commercial training simulator.

The acceleration input command magnitudes are as follows :

ACPM (SURGE) = 2,0 m/s/s

ACPM (PITCH) = 0,05 rad/s/s

The pulse length for the acceleration command is arbitrary and that for the rotational acceleration input was chosen so as to limit the pitch attitude of the aircraft to a reasonable limit.

In addition to the above results, the effect of eliminating the pitch angle generated by the translation acceleration motion filter is illustrated. This is done using the Motion Filter Model only and not the Simulator Model.

4.[illegible] MOTION FILTER PERFORMANCE

The motion filter has been developed to enable the high frequency components of the translational and rotational accelerations to be used to generate the basic motion cues. The low frequency translational acceleration cue component, which is demanding in terms of the required platform motion envelope, is implemented by means of the tilt generation. Therefore the total motion envelope requirement is reduced. Of major importance in this motion filtering scheme is the choice of break frequencies for the component filters. Provided that suitable cut-off frequencies and gains are chosen the reduction in the required motion envelope can successfully be traded off against the loss in motion cue fidelity. On the translational acceleration motion filtering axis there are two filters to be taken into account. Firstly; the specific force high pass filter and secondly; the position washout filter, which is a low pass filter. Also on the rotational acceleration filtering axis, there is a high pass filter similar to the translational axis high pass filter. The cross coupling between the two filter paths is via the tilt generating low pass compliment of the specific force. Results are shown for the high pass filter break frequency of 0.4rad/s and the position washout filter with a break frequency of 1.0rad/s.

Figures E.1.1 to E.1.5, obtained from the motion filter model, show some selected variables which indicate the response of the motion filter, the pilot and the operator, as a function of the translational acceleration input. In Figure E.1.1 the translational pilot x acceleration input $\ddot{x}_p$ and the specific force on the pilot FX are similar because there is no aircraft pitch attitude which would cause a component of gravity to be added to xp. FXH and FXL are the high and low frequency components of FX respectively. The pitch tilt command Θ_t is also shown. The pilot and operator x axis vestibular responses, $yotoi_x$ and $yotoo_x$ respectively are shown in Figure E.1.2. The y axis indicates the vestibular response in terms of threshold units (Sivan and Talwar [17]). One threshold unit indicates a signal with a detection probability of 0,7. As can be seen from this figure the operator receives an indication of cue onset which is similar to that experienced by the pilot. This is due to the translation of the platform which is shortlived. The sustained cue, which is a function of the tilt command, then becomes evident. The operator thus receives a similar translational cue to the pilot, except that the amplitude of the cue is reduced. This is apparently acceptable, Stewart [5]. What is important here is that there is no discrepancy in the sign of the cue. In generating the translational acceleration cue the position washout filter is driven by a zero position command and therefore negative acceleration of the platform is inevitable. This motion filter technique , however, by virtue of the tilt command eliminates the effect of this as seen by the operator. Figure E.1.3 shows the pilot and operator vestibular responses with the tilt angle generation disabled.

As is indicated by Figure E.1.2. The cue onset is below one threshold unit. Increasing the motion required to raise this above one threshold unit, increases the motion envelope required. Once again there is a trade-off between the desired cue and the necessary motion requirement. As can be seen in Figure E.1.5 the platform displacements remain within the limits imposed by the motion platform actuators. The scales for the translation envelope (± 0.76m) and for the rotational envelope (± 0,17rad) indicated, are typically the limits that can be expected on the motion platform that has been modelled. It is evident that the travels could not be extended much more before going out of limits. Also no account has been taken of the relative position of the operator and the platform which will tend to increase the platform motion to achieve the indicated pilot motion. In section 4.3, comments on simulator results where this was taken into account are given.

In order to achieve the results discussed, the inclusion of the tilt angle has introduced a major difference between the pilot's and the operators's perception of the situation. Figure E.1.4 shows the pitch axis vestibular responses $ysccI_\theta$ and $yscco_\theta$ of the pilot and the operator. This discrepancy is inevitable and needs to be introduced at acceptable levels so as not to detract from the simulation. Any reduction, however, will have a similar effect on the sustained cue generated. The difference indicated here appears to be unacceptably high.

Briefly then, the motion filter appears to yield a suitable operator response with a sustained cue and suitable cue onset accompanied by the predicted pitch cue discrepancy.

Thus far only an x axis pilot translational acceleration command has been implemented. Figure E.1.6 indicates the pilot pitch acceleration command q and its high frequency component qh. The pilot and operator pitch vestibular responses, $ysccI_\theta$ and $yscco_\theta$ respectively, in Figure E.1.7 show reasonable similarity. The x axis responses (Figure E.1.8) $yotoi_x$ and $yotoo_x$ show reasonable similarity. The x axis response of the pilot results from the pitch attitude of the aircraft. The simulator pitch, as well as the subtraction of the x axis gravity component from the position washout filter input, on the translational motion filter axis, causes the operator response as shown . This subtraction is implemented so that translational cue generation is freed from duplicating the gravity induced translational cue, which will be available by virtue of the platform tilt position. In transforming the g vector from inertial to body forces for this purpose, the platform position is taken to be only that component generated by the rotational acceleration filter, and not that generated by the tilt command. If this were not so, the effect of the platform tilt would be negated by applying a compensating translational acceleration to the system.

Figure E.1.9 indicates the motion of the operator in inertial space in six degrees of freedom.

Thus, the motion presented to the operator has similar cueing effects to those perceived by the pilot. Also, the motion envelope in which the operator moves, is within the envelope of motion allowed of the platform.

The filter break frequencies used in the model, yield reasonable results in terms of most of the pilot to operator motion transformations. In order to increase the translational acceleration cue onset, without increasing the platform motion envelope requirement the high pass break frequency could be raised in conjuction with the gain KAT. This would however, increase the bandwidth of the tilt command signal which would aggravate the already serious, tilt angle command induced, rotational cue discrepancy. Meiry [18] suggests that the semi-circular canals are known to behave like heavily damped angular accelerometers which result in the perception of angular velocities over the frequency range, 0,02 Hz (0,13rad/s) to 1,5 Hz (9,42rad/s), and that the otoliths behave like linear accelerometers which function dynamically over the frequency range 0,016Hz (0,1 rad/s) to 0,24Hz (1,57rad/s). Thus, the bandwidth of the rotational motion filter, 0,06Hz (0,4rad/s) up to the motion platform dynamic limits, and that of the translational motion filter, 0,06Hz (0,4rad/s) to 0,16Hz (1rad/s), both correlate with the vestibular dynamic ranges. Also the bandwidth of the input command signal, 1Hz (3,14rad/s) is such that it covers the whole of the relevant frequency spectrum.

4.3 SIMULATOR MODEL PERFORMANCE

In developing the motion filters there were two requirements which needed to be met to ensure that the proposed solution could prove successful. Firstly, the operator and pilot vestibular responses would have to show some definite correlation, and secondly, the platform motion restriction must not be violated. The first of these has been shown to be partially successful. The plots shown in Figures E.2.8 to E.2.11 give the response of the vestibular models and the platform as a result of the translational and then the rotational acceleration aircraft inputs to the system. The relative position of the pilot to the centre of gravity of the airframe; and the operator to the centre of the platform, cause rotational motion to be transformed into a combination of rotational and translational motion. This has the effect of increasing the motion requirement of the platform above those shown for the operator. (Figure E.1.9 for example.)

Figure E.2.1 shows the surge axis yotoi$_x$ and yotoo$_x$ translational vestibular responses of the pilot and the operator to the x axis acceleration input. Similarly Figure E.2.2 and E.2.3 show the z axis translational, and y axis pitch vestibular responses. Figure E.2.4 indicates the actuator lengths required to provide the necessary platform motion. Similar plots are given in Figures E.2.5 to E.2.8 for vestibular and platform motion responses to the aircraft pitch acceleration input signal. All of the vestibular responses are similar to those discussed in section 4.2.

The actuator lengths are plotted using a scale of 2,4 to 3,9m (Figures E.2.4, E.2.8 and E.2.11). This corresponds to the modelled actuator length and travel characteristics. In both cases the actuators remain with the allowable limits. There appear only to be 3 plots for the 6 actuators. This is because the platform geometry is symmetrical about the x-z plane and the actuators move in pairs (1 & 6, 2 & 5 and 3 & 4).

Provision is made in the simulator model, to include an actuator/servo model to investigate its influence on the motion fidelity. The actuator model that has been included, is a very simple one and does not have any extraordinary effect on the simulator model However, Figures E.2.9 and E.2.10 have been included to indicate the error between the command actuator length signal and the resultant actuator length ALE, and the resultant operator position error $(X,Y,Z,\theta,\phi,\psi)_e$, which are available from the model. For this simulation run, both the translational x axis and the pitch axis aircraft accelerations were simultaneously used to drive the model.

Figure E.2.11 shows the actuator length plots AL as a function of the combined inputs. This case could conceivably have been more stringent than for just a single axis input. Once again the motion requirements are shown to be within limits.

4.4 DISCUSSION

In order to conduct a true evaluation of the vestibular response correlation it is necessary to consider results that have been discussed in the light of the rest of the simulator environment. It must be stressed that the straight comparison between vestibular responses on a threshold perception scale may not indicate the true characteristics of the motion filtering technique used. Due to the fact that the full motion is rarely presented in a simulator, the characteristic which must be present is the correct representation of cue onset. The magnitude of this cue must also be such that it is perceived by the operator. This perception is influenced by the workload or stress on the operator. If the particular flight regime under consideration is straight and level flight with little operator activity then he will be for more likely to perceive a nominal turbulence 'bump'. For the case where the operator is approaching a landing under emergency condition which requires large concentration levels a similar bump might go unnoticed. Also in a simulator flight with a persistant light turbulence condition, supported by motion cueing, the operator might well become habituated to the turbulence and will therefore be less perceptive to that particular cue. Therefore the cue representation achieved in terms of cue onset and perception levels could well be satisfactory.

A similar argument can be used for those cues which display large discrepancies between the required pilot and resultant operator responses. In terms of the pitch tilt cue deviation it could be seen to be unsatisfactory. To improve the situation, either the magnitude of the tilt gain could be reduced or the frequency component could be reduced. Both of these options would be traded off against a corresponding reduction in the sustained translational cue perception of the operator. Once again, the only successful method of determining the best solution would be to implement the controller on a platform and elicit some response from a group of operators. This pitch tilt cue discrepancy is evident in a number of operational simulator installations (Link-Miles trainers for example) and the results obtained are reported to be satisfactory. Apart from the minimisation of the discrepancy there are a number of other reasons why this is so.

Firstly, the other cues which the operator receives enforce this perception of the required situation. The false cue is therefore compensated for. Secondly, the effect of the tilt generation will not necessarily have the same sharp rise which is evident in the plots shown and is as a result of the steplike aircraft acceleration inputs thus, the major effect will be the angular position discrepancy which the vestibular system is relatively insensitive to.

CHAPTER FIVE

CONCLUSIONS

5.1 CONCLUDING REMARKS

A motion filter has been developed which transforms aircraft motion into simulator platform motion. Although only surge and pitch motions are investigated, the model extends to all six translational and rotational degrees of freedom. Also shown, are the motion base actuator lengths which are required to generate the indicated platform motion. The simulator platform motion generation involves the following motion transformation and filtering. Aircraft motion is transformed into pilot motion (at the centre of perception of the pilot). This motion is then filtered to obtain simulator operator motion. This is achieved by using the high frequency components of the pilot's accelerations (rotational and translational) as operator motion inputs. Thus, the required operator displacements are reduced. In order to ensure that the operator can accommodate a series of translational motion inputs, the input acceleration is treated as a disturbance by a second order filter which drives the operator to a neutral position. It is assumed that the attitude requirements, imposed on the operator by the rotational acceleration inputs, remain within acceptable limits. This would certainly be the case for normal operation of a large passenger aircraft such as has been modelled. Also, to generate an operator awareness of sustained translational acceleration, a tilt angle is generated which is a function of the low frequency translational pilot acceleration.

The resultant operator motions are then transformed into actuator lengths. The effects on the actuator lengths of simple hydraulic valve and actuator characteristics are also shown.

In order to validate the results obtained from the motion filtering, the vestibular response of the pilot and operator were compared for various input accelerations. The actuator lengths in each of these cases were also monitored.

CHAPTER FIVE

CONCLUSIONS

5.1 CONCLUDING REMARKS

A motion filter has been developed which transforms aircraft motion into simulator platform motion. Although only surge and pitch motions are investigated, the model extends to all six translational and rotational degrees of freedom. Also shown, are the motion base actuator lengths which are required to generate the indicated platform motion. The simulator platform motion generation involves the following motion transformation and filtering. Aircraft motion is transformed into pilot motion (at the centre of perception of the pilot). This motion is then filtered to obtain simulator operator motion. This is achieved by using the high frequency components of the pilot's accelerations (rotational and translational) as operator motion inputs. Thus, the required operator displacements are reduced. In order to ensure that the operator can accommodate a series of translational motion inputs, the input acceleration is treated as a disturbance by a second order filter which drives the operator to a neutral position. It is assumed that the attitude requirements, imposed on the operator by the rotational acceleration inputs, remain within acceptable limits. This would certainly be the case for normal operation of a large passenger aircraft such as has been modelled. Also, to generate an operator awareness of sustained translational acceleration, a tilt angle is generated which is a function of the low frequency translational pilot acceleration.

The resultant operator motions are then transformed into actuator lengths. The effects on the actuator lengths of simple hydraulic valve and actuator characteristics are also shown.

In order to validate the results obtained from the motion filtering, the vestibular response of the pilot and operator were compared for various input accelerations. The actuator lengths in each of these cases were also monitored.

Simulation runs were done for the pilot to operator motion filter and then for the whole model, ie from the aircraft to the actuator motion. Two input accelerations were used for the pilot to operator motion filter model, a steplike pilot surge acceleration and then a steplike pilot pitch acceleration with magnitudes of 2,0m/s/s and 0,05rad/s/s respectively. The same two inputs were used for the simulator model. Vestibular response plots were obtained for each of the inputs separately. The two inputs were combined, however, to investigate the performance of the actuators relative to there motion limits.

For all of the simulation cases investigated the actuator remained within the prescribed limits. Also, the correlation between the pilot and vestibular plots has been reconciled with the filtering technique used. The tilt generation has been shown to improve the correlation between the pilot and operator vestibular responses. This success is detracted from by the resulting misrepresentation of the pitch axis vestibular response. Although this discrepancy is obvious, it is difficult to predict whether this would result in unacceptable cue levels in a simulator. Here, account must be taken of the supportive effect of other cues on the desired pilot perception of his environment.

In order to achieve a more realistic motion cue for simulator pilots than has been developed here, the following two courses of action are open.

Firstly, it is obvious that the limited travel which is allowed by the motion platform configuration used imposes severe restrictions on the translational cue realisation. As has been done at a number of simulator installations the travel allowed by the motion system has been increased (e.g. the Vertical motion simulator and the Large Multinode Aerospace Research Simulator [8]). The obvious implications of this added capability is a substantial increase in cost and therefore the 6 axis synergystic platform which was modelled in remained popular.

Secondly, the approach to the motion filtering technique could eliminate some of the major deficiencies of the proposed scheme. Because the platform is driven to some zero or neutral position a proportion of its total envelop remains largely unused. A deterministic or optimised motion generating technique which takes into account the aircraft motion and predicted manoeuvres could continually 'shift' the neutral position in order to make maximum use of the available motion envelope. A great deal of work is being done in this regard (Sivan and Talwar [17]) already.

5.2 SUGGESTIONS FOR FURTHER WORK

(i) Model performance can be investigated for the remaining four degrees of freedom.

(ii) Motion envelope requirements for pilot positions with respect to aircraft centre of gravity can be examined.

(iii) Various platform configurations can be investigated to determine relative suitability for particular applications.

(iv) Research needs to be conducted in order to determine some technique to enable the accurate predictions as to the suitability of various motion filtering techniques. These predictions need to be achieved without the use of a simulator platform.

APPENDIX

APPENDIX A

AXIS TRANSFORMATIONS

To obtain the transformation matrix [L] which will transform a vector from an inertial axis system to a body axis system, rotations are considered first about the yaw axis (psi), then about the pitch axis (theta) and lastly about the roll axis (phi). Taking rotations about the three body axes (x1, x2 and x3) the following is obtained (Goldstein [19]).

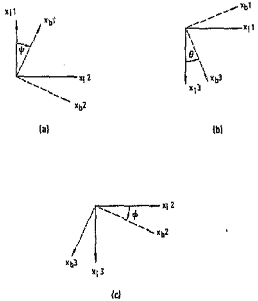

FIGURE A.1 : The Three Basic Rotations
(a) about xi3
(b) about xi2 and
(c) about xi1

with the transformation matrices.

$$L3(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$L2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \qquad \text{(A-1)}$$

$$L1(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}$$

The matrix L which transforms a vector from the body axis to the inertia axis is the product of L1, L2 and L3 in the reverse order of rotation as the transformation premultiplies the vector arrived at in the previous step. Going from inertia into body oriented axes systems the rotations are about xi3 (ψ), xi2 (θ) and xi1 (ϕ). Thus the transformation is,

$$L = L1(\phi)*L2(\theta)*L3(\psi) \qquad \text{(A-2)}$$

which yields,

$$L = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \qquad (A-3)$$

Thus the vector v_i in the inertial axis reference frame is transformed into v_b in the body axis reference frame by the matrix L.

$$v_b = L.v_i \qquad (A-4)$$

The inverse transformation from body coordinates to inertial coordinates,

$$v_i = (L)^{-1}.v_b,$$

is given by the *transpose* (matrix T) of matrix L. This is true because L is an orthogonal matrix.

$$v_i = T.v_b \qquad (A-5)$$

where,

$$T = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi-\cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi+\cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \qquad (A-6)$$

APPENDIX B

POSITION, VELOCITY AND ACCELERATION IN AN ARBITRARY REFERENCE FRAME WITH RESPECT TO A FIXED REFERENCE FRAME

Consider a point P in the body reference frame, FB(xb1,xb2,xb3), which moves arbitrarily in the inertial reference frame, FI(xi1,xi2,xi3). An expression is required which relates the motion of P to the inertial reference frame (see Figure B.1). According to Etkin [20] the following is obtained.

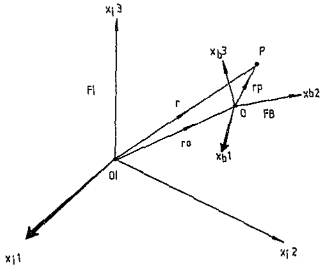

FIGURE B.1 : Moving Coordinate System

Let

$$r = ro + rp \qquad \text{(B-1)}$$

then the velocity of P in the inertial reference frame is

$$\begin{aligned} v_i &= \dot{r}_i \\ &= \dot{ro}_i + \dot{rp}_i \qquad \text{(B-2)} \\ &= vo_i + \dot{rp}_i \ . \end{aligned}$$

Now the velocity components of v_i (equations B-2) in FB are

$$v_b = L.v_i = L.(vo_i + \dot{rp}_i) = vo_b + L.\dot{rp}_i$$

where L is the transformation matrix as defined in Appendix A, equation A-3 and transforming the derivative,

$$v_b = vo_b + \dot{rp}_b + w_b.rp_b \qquad \text{(B-3)}$$

where the first term of (B-3) is the velocity of O relative to FI, the second term is the velocity of P relative to FB and the last is the transport velocity. Here

$$w_b = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} , \quad rp = \begin{bmatrix} x_{b1} \\ x_{b2} \\ x_{b3} \end{bmatrix} \qquad \text{(B-4)}$$

Differenciating vi and using (B-3), the acceleration of P in the body reference frame is obtained.

$$a_b = L.\dot{v}_i$$

$$= \dot{v}_b + w_b.v_b$$

$$= ao_b + \ddot{rp}_b + \dot{w}_b.rp_b + 2.w_b.\dot{rp}_b + w_b.w_b.rp_b$$

$$= ao_b + \ddot{rp}_b + 2.w_b.\dot{rp}_b + (\dot{w}_b + w_b.w_b).rp_b \qquad \text{(B-5)}$$

where :

ao_b	:	$\dot{vo}_b + w_b.vo_b$ is the acceleration of O relative to FI
$\ddot{rp}_b$	:	acceleration of P relative to FB
$\dot{w}_b.rp_b$	:	tangential acceleration owing to *rotational* acceleration of FB
$2.w_b.\dot{rp}_b$	:	coriolis acceleration and
$w_b.w_b.rp_b$	:	centripetal acceleration.

APPENDIX C

TAYLOR SERIES EXPANSIONS

Consider an equation of the form

$$f(x,y,z) = 0.$$

Using the Taylor series and expanding the function about the point x=a, y=b and z=c the following is obtained.

$$f(x,y,z) = f(a,b,c)+f_x(a,b,c).(x-a)+f_y(a,b,c).(y-b)+f_z(a,b,c).(z-c) = 0$$

and writing in matrix form :

$$[f_x(a,b,c)\ f_y(a,b,c)\ f_z(a,b,c)].\begin{bmatrix} x-a \\ y-b \\ z-c \end{bmatrix} = [-f(a,b,c)] \qquad (C-1)$$

where x, y and z are the predicted roots and a, b and c are estimates of the roots. Thus, to obtain the real roots the error matrix must be reduced to zero by successively updating the root estimate with the error obtained.

APPENDIX D

PROGRAM LISTINGS

D.1 SIMULATOR MODEL

Figure D.1 indicates the flow of the program as well as those parts of the program which are fortran subroutines. The program runs a time simulation of the model and has a communication interval of 0,050 seconds. This means that data from the model is stored at 0,050 second intervals for displaying in either a printed or plotted format. The calculation interval is defined in each of the derivative blocks of the program according to the requirements of that block. It is specified by the command NSTEPS which fixes the number of calculations to be done in each communication interval. This in effect allows each of the blocks to be sampled at the slowest possible rate which in turn drastically reduces the program execution time.

It must be stressed that the model is not intended to run in real time and therefore steps taken to reduce the cycle rate are to reduce program execution time and therefore cost, and not to obtain real time execution.

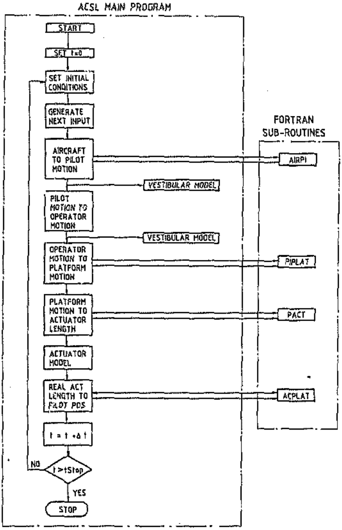

FIGURE D.1 : Simulator Model Program Structure

```
PROGRAM FLY "SIMULATOR MODEL "

""
"---- SET UP INTEGER VALUES FOR ACSL PROGRAM -----------------------"
""
      INTEGER N                      $" JACK NO. OR AXIS NO.             "
      INTEGER IN                     $" PLAT. POS. INITIALISING FLG.     "
      INTEGER FLG                    $" ACPLAT ERROR INDICATING FLG.     "
""
"---- SET UP ARRAYS FOR THE INTERFACE VARIABLES -------------------"
""
      ARRAY    DXA(6),DDXA(6),DDXAA(6),DDXAIC(6)                   ...
               ,CRX(3),DCRX(3),DDCRX(3)                            ...
               ,XD(3)                                              ...
               ,DXP(6),DDXP(6)                                     ...
               ,TN(6),PL(6),PW(6),ACPM(6),ONOF(6)                  ...
               ,XS(6)                                              ...
               ,P(6)                                               ...
               ,AL(6)                                              ...
               ,IP(6)                                              ...
               ,CI(6),Q(6),OV(6),OVIC(6)                           ...
               ,OP(6),DOP(6)                                       ...
               ,LERR(6),ERR(6)                                     ...
               ,AM(6)                                              ...
               ,PR(6)                                              ...
               ,PERR(6)
      ARRAY     TSCC1(2),TSCC2(2),QN(3),QD(3)
      DIMENSION  VESTO(6),VESTI(6),VESTER(6)



INITIAL
"    CALL LISTD(5)  "
""
"---- COMMUNICATION INTERVAL AND TERMINATION ----------------------"
"---- CONDITION SPECIFICATIONS ------------------------------------"
""
      CINTERVAL CINT  =  0.050    $"COMMUNICATION INTERVAL          "
      CONSTANT  TF    = 10.00     $"SIMULATION LENGTH               "



""
"---- CONSTANTS INITIAL CONDITIONS FOR THE GENERATION OF THE ------"
"---- AIRCRAFT FLIGHT MODEL PROFILES ------------------------------"
"           ( DERIVATIVE SECTION 1 )                                "
""
"      TN    -INITIALISATION TIME  (S)   "
"      PL    -PULSE LENGTH  (S)   "
"      PW    -PULSE WIDTH   (S)   "
"      ACPM  -ACC. PULSE MAGNITUDE   (M/S2)/(RAD/S2)   "
"      ONOF  -PULSE ON/OFF SWITCH    "
""
"      D.O.F. /           X      Y       Z     ROLL   PITC   YAW    "
      CONSTANT TN      = 1.0 , 1.0 , 1.0 , 0.0 , 0.0 , 0.0
      CONSTANT PL      =15.0 ,15.0 ,15.0 , 6.0 , 6.0 , 6.0
      CONSTANT PW      = 5.0 , 5.0 , 5.0 , 2.0 , 2.0 , 2.0
      CONSTANT ACPM    = 2.0 , 2.0 , 2.0 , 0.05 , 0.05 , 0.05
      CONSTANT ONOF    = 0.0 , 0.0 , 0.0 , 5.0 , 0.0 , 0.0
```

```
""
"---- AIRCRAFT ACCELERATION INITIAL CONDITION   "
     CONSTANT DDXAIC =     0.0     ,    0.0     ,    0.0        ...
                     ,     0.0     ,    0.0     ,    0.0


""
"---- CONSTANTS INITIAL CONDITIONS FOR THE TRANSFORMATION FROM ------"
"---- AIRCRAFT TO AIRCRAFT-PILOT MOTION ----------------------------"
"          ( DERIVATIVE SECTION 1 )                                  "
""



""
"---- CONSTANTS AND INITIAL CONDITIONS FOR THE TRANSFORMATION ------"
"---- FROM AIRCRAFT-PILOT TO SIMULATOR-PILOT MOTION ----------------"
"          ( DERIVATIVE SECTION 2 )                                  "
""

     CONSTANT QN=1.0,0.0,0.0       $" HIGH PASS FILTER NUM ARRAY     "
     CONSTANT QD=1.0,0.64,0.16     $"  ✓      ✓       ✓    DEN   ✓   "
     CONSTANT TSCC1=1.,0.        $" SEMIC.CANAL FILTER NUM ARRAY      "
     CONSTANT TSCC2=1.,0.169       $" SEMIC.CANAL FILTER DEN ARRAY   "
     CONSTANT GO=2.16          $" VALUES FOR THE OTOLITH FILTER      "
     CONSTANT AO=0.076         $"                 ✓                  "
     CONSTANT BO=0.19          $"                 ✓                  "
     CONSTANT GS=39.51         $"                 ✓                  "
     CONSTANT BS=0.169         $"                 ✓                  "
     CONSTANT G=-10.           $" GRAVITATIONAL CONSTANT             "
     CONSTANT ETA=0.8         $" POSITION WASHOUT FILTER DAMPING RATIO "
     CONSTANT WN=1.00         $"  POSITION FILTER NATURAL FREQUENCY  "
     CONSTANT MK=1.0
     MK1=((MK/(WN**2.0))-1.)/MK     $" POSITION FILTER ACC LOOP GAIN "
     MK2=(2.0*ETA)/WN         $" POSITION FILTER VELOCITY LOOP GAIN  "
     CONSTANT KT4=0.6         $" ROLL TILT COMMAND GAIN              "
     CONSTANT KT5=0.6         $" PITCH TILT COMMAND GAIN             "
     CONSTANT KI4=1.0         $" ROLL TILT COMMAND SENCE             "
     CONSTANT KI5=-1.0        $" PITCH TILT COMMAND SENCE            "
     CONSTANT KI6=1.0         $" YAW TILT COMMAND SENCE              "
     CONSTANT KAT=0.6         $" TRANS ACC COMMAND GAIN              "
     CONSTANT KAR=0.6         $" ROT ACC COMMAND GAIN                "



""
"---- CONSTANTS AND INITIAL CONDITIONS FOR THE ACTUATOR -----------"
"---- AND ACTUATOR CONTROLLER MODEL --------------------------------"
"          ( DERIVATIVE SECTION 4 )                                  "
""
     CONSTANT A         =     .0025 $"ACTUATOR AREA  (M2)            "
     CONSTANT ALNEUT =      3.162  $"ACTUATOR NEUTRAL LENGTH   (M)   "
     CONSTANT K         =    20.0   $"CURRENT GAIN  (MA/M)           "
     CONSTANT KR        =     0.50  $"RATE FDBCK GAIN  (M/M/S)       "
     CONSTANT KV        =   3.8E-3  $"VALVE GAIN  (M3/S/MA)          "
     CONSTANT KL        =  1.5E-11  $"LEAKAGE GAIN  (M3/S/PA)        "
     CONSTANT MU        =    10.0   $"VISCOSITY                      "
```

```
""
"---- ACTUATOR MODEL INIT. VELOCITY CONDITIONS   "
      CONSTANT OVIC   =    0.0    ,    0.0    ,    0.0         ...
                      ,    0.0    ,    0.0    ,    0.0
      CONSTANT OV     =    0.0    ,    0.0    ,    0.0         ...
                      ,    0.0    ,    0.0    ,    0.0


""
"---- CONSTANT (FLAG) FOR THE INITIALISATION OF THE INITIAL ---------"
"---- GUESS FOR THE ACTUATOR LENGTH IN THE TRANSFORMATION FROM ------"
"---- ACT. LENGTH TO PLATFORM POSITION  (SUBR. ACPLAT) -------------"
"           ( DERIVATIVE SECTION 5 )                                "
""
      CONSTANT IN= 0


END  $ "END OF INITIAL "




DYNAMIC
""
"-     COUNTER FOR THE INITIALISATION OF THE PLATFORM NEUTRAL        "
"-     POSITION CONSTANTS IN SUBROUTINE ACPLAT .                     "
""
       IN=IN+1


DERIVATIVE $"SECTION 1"
       NSTEPS NSTP1 = 1
""
"--    SET UP REQUIRED SIMULATION ACCELERATION INPUTS              --"
""
"-     AIRCRAFT MOTION (XA) IN THE F.R.F. IS GENERATED AS WELL       "
"-     AS THE MOVEMENT OF THE CENTER OF ROTATION OF THE AIRCRAFT     "
"-     WITH RESPECT TO THE FUSELAGE (CRX).                           "
""
      PROCEDURAL(XA,DXA,DDXA,CRX,DCRX,DDCRX=TN,PL,PW,ACPM,ONOF)
       DDXA(1)=CMPXPL(.1,.51,(PULSE(TN(1),PL(1),PW(1)))*ACPM(1)      ...
               *ONOF(1))
       DDXA(2)=0.
       DDXA(3)=0.
       DDXA(4)=0.
       DDXAT5=(PULSE(TN(5),PL(5),PW(5))-.5)*2.*ACPM(5)*ONOF(5)
       DDXA(5)=CMPXPL(.1,.51,RSW(T.GE.4.,0.,DDXAT5))
       DDXA(6)=0.

       CRX(1)    =-20.0
       CRX(2)    = 0.0
       CRX(3)    = 0.0
       DCRX(1)    = 0.0
       DCRX(2)    = 0.0
       DCRX(3)    = 0.0
       DDCRX(1)    = 0.0
       DDCRX(2)    = 0.0
       DDCRX(3)    = 0.0
```

```
     END

      CALL XFERB(DDXAA=DDXA,6)
      DXA=INTVC(DDXAA,DDXAIC)

""
"--    AIRCRAFT TO AIRCRAFT-PILOT TRANSFORMATION    ------------------"
""
"-     HERE THE AIRCRAFT MOTION IS TRANSFERED TO PILOT MOTION .        "
"-     ACCOUNT IS TAKEN OF THE AIRCRAFT-PILOT'S POSITION WITH          "
"-     RESPECT TO THE AIRCRAFT CENTER OF ROTATION . PROVISION          "
"-     IS MADE FOR CONSIDERING CASES WHERE THE CENTER OF ROTATION      "
"-     VARIES WITH TIME , SUCH AS WOULD BE THE CASE WHEN THE           "
"-     AIRCRAFT MASS AND MASS DISTRIBUTION CHANGE AS FUEL IS           "
"-     CONSUMED IN FLIGHT .                                            "
""
     PROCEDURAL(    DXP,DDXP,XD          =                        ...
                    DXA,DDXA,CRX,DCRX,DDCRX)
      CALL AIRPI(    DXP,DDXP,XD          =                       ...
                     DXA,DDXA,CRX,DCRX,DDCRX)
     END
END $"OF DERIVATIVE SECTION 1"




DERIVATIVE $"SECTION 2"
      NSTEPS NSTP2 = 5
""
"--    GENERATE SIMULATOR-PILOT MOVEMENTS    -------------------------"
""
PROCEDURAL(DDXP1,DDXP2,DDXP3,DDXP4,DDXP5,DDXP6=DDXP)
       DDXP1=DDXP(1)
       DDXP2=DDXP(2)
       DDXP3=DDXP(3)
       DDXP4=DDXP(4)
       DDXP5=DDXP(5)
       DDXP6=DDXP(6)
END
" CALCULATE THE AIRCRAFTS ROTATIONAL POSITION "
       DXP4=INTEG(DDXP4,0.0)
       DXP5=INTEG(DDXP5,0.0)
       DXP6=INTEG(DDXP6,0.0)
       "TRANSFORMING RATES FROM BODY TO INERTIAL AXES"
       DA4=DXP4+ SIN(A4)*TAN(A5) *DXP5+ COS(A4)*TAN(A5) *DXP6
       DA5=       COS(A4)           *DXP5- SIN(A4)          *DXP6
       DA6=      (SIN(A4)/COS(A5))*DXP5+(COS(A4)/COS(A5))*DXP6
       "INTEGRATING FOR POSITION"
       A4=INTEG(DA4,0.0)
       A5=INTEG(DA5,0.0)
       A6=INTEG(DA6,0.0)
" CALCULATE THE SPECIFIC FORCE INPUTS "
" PILOT VESTIBULAR INPUT "
       FXPV1=DDXP1-          SIN(A5)*G
       FXPV2=DDXP2+SIN(A4)*COS(A5)*G
       FXPV3=DDXP3+COS(A4)*COS(A5)*G-G
" MOTION FILTER INPUT "
       FXP1=DDXP1
       FXP2=DDXP2
       FXP3=DDXP3
" VESTIBULAR RESPONSE OF PILOT "
```

```
     YIOTO1=GO*(AO/BO)*LEDLAG(AO,BO,FXPV1,0.0)
     YIOTO2=GO*(AO/BO)*LEDLAG(AO,BO,FXPV2,0.0)
     YIOTO3=GO*(AO/BO)*LEDLAG(AO,BO,FXPV3,0.0)
     YISCC4=GS*TRAN(1,1,TSCC1,TSCC2,DXP4)
     YISCC5=GS*TRAN(1,1,TSCC1,TSCC2,DXP5)
     YISCC6=GS*TRAN(1,1,TSCC1,TSCC2,DXP6)
" GENERATE HIGH AND LOW PASS COMPONENTS OF SPECIFIC FORCE "
     FH1=TRAN(2,2,QN,QD,FXP1)
     FH2=TRAN(2,2,QN,QD,FXP2)
     FH3=TRAN(2,2,QN,QD,FXP3)
     FXPL1=FXP1-FH1
     FXPL2=FXP2-FH2
     FXPL3=FXP3-FH3
" SUBTRACT GRAVITY COMPONENT OF PLATFORM POSITION FROM ...
  PLATFORM ACCELERATION COMMAND.  "
     FXPH1=KAT*FH1-(-           SIN(XNR5)*G)
     FXPH2=KAT*FH2-( SIN(XNR4)*COS(XNR5)*G)
     FXPH3=KAT*FH3-( COS(XNR4)*COS(XNR5)*G-G)
" FILTER THE SPECIFIC FORCE DRIVING SIGNAL FOR OUTPUT POSITION "
     DDS1=(FXPH1-MK*(MK2*DS1+S1))/(1.0+MK*MK1)
     DDS2=(FXPH2-MK*(MK2*DS2+S2))/(1.0+MK*MK1)
     DDS3=(FXPH3-MK*(MK2*DS3+S3))/(1.0+MK*MK1)
     DS1=INTEG(DDS1,0.0)
     DS2=INTEG(DDS2,0.0)
     DS3=INTEG(DDS3,0.0)
     S1=INTEG(DS1,0.0)
     S2=INTEG(DS2,0.0)
     S3=INTEG(DS3,0.0)
" TRANSFORM THE DISPLACEMENTS FROM BODY TO INERTIAL AXES "
     XS1=(           COS(XS5)*COS(XS6)                        )*S1 ...
        +(SIN(XS4)*SIN(XS5)*COS(XS6)-COS(XS4)*SIN(XS6))*S2 ...
        +(COS(XS4)*SIN(XS5)*COS(XS6)+SIN(XS4)*SIN(XS6))*S3
     XS2=(           COS(XS5)*SIN(XS6)                        )*S1 ...
        +(SIN(XS4)*SIN(XS5)*SIN(XS6)+COS(XS4)*COS(XS6))*S2 ...
        +(COS(XS4)*SIN(XS5)*SIN(XS6)-SIN(XS4)*COS(XS6))*S3
     XS3=(                            -            SIN(XS5))*S1 ...
        +(                            SIN(XS4)*COS(XS5))*S2 ...
        +(                            COS(XS4)*COS(XS5))*S3
" GENERATE TILT CMDS AS FUNCTION OF SPECIFIC FORCE "
     PT4=KI4*KT4*(FXPL2/G)
     PT5=KI5*KT5*(FXPL1/G)
     PT6=0.0
" GENERATE HIGH PASS COMPONENT OF AIRCRAFT ROTATIONS "
     DDPH4=TRAN(2,2,QN,QD,DDXP4)
     DDPH5=TRAN(2,2,QN,QD,DDXP5)
     DDPH6=TRAN(2,2,QN,QD,DDXP6)
" GENERATE THE ROTATIONAL VELOCITY COMMAND FROM THE ...
  AIRCRAFTS ROTATIONAL ACCELERATION "
     DS4=INTEG(DDPH4,0.0)
     DS5=INTEG(DDPH5,0.0)
     DS6=INTEG(DDPH6,0.0)
" TRANSFORM ROTATIONAL RATES FROM BODY TO INERTIAL AXES "
     DXS4=DS4+ SIN(XNR4)*TAN(XNR5) *DS5+ COS(XNR4)*TAN(XNR5) *DS6
     DXS5=      COS(XNR4)            *DS5- SIN(XNR4)            *DS6
     DXS6=     (SIN(XNR4)/COS(XNR5))*DS5+(COS(XNR4)/COS(XNR5))*DS6
" INTEGRATE FOR POSITION "
     XNR4=INTEG(DXS4,0.0)
     XNR5=INTEG(DXS5,0.0)
     XNR6=INTEG(DXS6,0.0)
" ADD TILT COMPONENT TO POSITION COMMAND "
     XS4=XNR4+PT4
     XS5=XNR5+PT5
     XS6=XNR6
```

```
" DIF. TO OBTAIN ROTATIONAL VELOCITY IN BODY REF FRAME "
      XV4=DERIVT(0.0,XS4)
      XV5=DERIVT(0.0,XS5)
      XV6=DERIVT(0.0,XS6)
" TRANSFORM FROM INERTIAL TO BODY REF FRAMES "
      DV4=XV4+                        SIN(XS5)*XV6
      DV5=    COS(XS4)*XV5+SIN(XS4)*COS(XS5)*XV6
      DV6=   -SIN(XS4)*XV5+COS(XS4)*COS(XS5)*XV6
" CALCULATE THE SPECIFIC FORCE EXPERIENCED BY THE OPERATOR "
      FXS1=DDS1-             SIN(XS5)*G
      FXS2=DDS2+SIN(XS4)*COS(XS5)*G
      FXS3=DDS3+COS(XS4)*COS(XS5)*G-G
" VESTIBULAR RESPONCE OF OPERATOR "
      YOOTO1=GO*(AO/BO)*LEDLAG(AO,BO,FXS1,0.0)
      YOOTO2=GO*(AO/BO)*LEDLAG(AO,BO,FXS2,0.0)
      YOOTO3=GO*(AO/BO)*LEDLAG(AO,BO,FXS3,0.0)
      YOSCC4=GS*TRAN(1,1,TSCC1,TSCC2,DV4)
      YOSCC5=GS*TRAN(1,1,TSCC1,TSCC2,DV5)
      YOSCC6=GS*TRAN(1,1,TSCC1,TSCC2,DV6)
PROCEDURAL(XS=XS1,XS2,XS3,XS4,XS5,XS6)
      XS(1)=XS1
      XS(2)=XS2
      XS(3)=XS3
      XS(4)=XS4
      XS(5)=XS5
      XS(6)=XS6
END
PROCEDURAL(VESTI,VESTO,VESTER=YIOTO1,YIOTO2,YIOTO3,YOOTO1 ...
                           ,YISCC4,YISCC5,YISCC6,YOSCC4,YOSCC5,YOSCC6 ...
                           ,YOOTO2,YOOTO3)
      VESTI(1)=YIOTO1
      VESTI(2)=YIOTO2
      VESTI(3)=YIOTO3
      VESTI(4)=YISCC4
      VESTI(5)=YISCC5
      VESTI(6)=YISCC6
      VESTO(1)=YOOTO1
      VESTO(2)=YOOTO2
      VESTO(3)=YOOTO3
      VESTO(4)=YOSCC4
      VESTO(5)=YOSCC5
      VESTO(6)=YOSCC6
      VESTER(1)=VESTI(1)-VESTO(1)
      VESTER(2)=VESTI(2)-VESTO(2)
      VESTER(3)=VESTI(3)-VESTO(3)
      VESTER(4)=VESTI(4)-VESTO(4)
      VESTER(5)=VESTI(5)-VESTO(5)
      VESTER(6)=VESTI(6)-VESTO(6)
END
END $"OF DERIVATIVE SECTION 2"


DERIVATIVE $"SECTION 3"
      NSTEPS NSTP3 = 1
""
"--   SIMULATOR-PILOT TO PLATFORM MOVEMENT TRANSFORMATION   ---------"
""
"-    SIMULATOR-PILOT MOTION IS TRANSFERED TO PLATFORM MOTION .     "
"-    MOTION OF THE GEOMETRIC CENTER OF THE PLATFORM IS             "
"-    CALCULATED .                                                  "
""
     PROCEDURAL(P,PI=XS)
```

```
     CALL PIPLAT(P,PI=XS)
    END

""
"--   PLATFORM TO ACTUATOR TRANSFORMATION   ---------------------------"
""
"-    ACTUATOR LENGTHS ARE CALCULATED TO GIVE THE REQUIRED             "
"-    PLATFORM MOTION .                                                 "
""
    PROCEDURAL(AL=P)
     CALL PACT(AL=P)
    END
END $"OF DERIVATIVE SECTION 3 "



DERIVATIVE $"SECTION 4"
      NSTEPS NSTP4 = 5
""
"--   ACTUATOR MODEL   -------------------------------------------------"
""
"-    THE ACTUATOR LENGTHS CALCULATED IN PACT DRIVE A MODEL OF          "
"-    THE ACTUATOR CONTROL AND HYDRAULICS WHICH OUTPUTS THE             "
"-    ACTUAL ACTUATOR LENGTHS THAT WOULD BE ACHIEVED IN THE REAL        "
"-    WORLD WHEN TAKING INTO ACCOUNT THE PHYSICAL PROPERTIES            "
"-    OF THE SYSTEM .                                                   "
""
PROCEDURAL(OV,OP=AL)
DO 410 N=1,6
            IP(N)=AL(N)-ALNEUT     $"INPUT POSITION  (M)                "
            ERR(N)=IP(N)-OP(N)     $"ERROR COMMAND  (M)                 "
            CI(N)=K*ERR(N)         $"CONTROL CURRENT  (MA)              "
            Q(N)=KV*CI(N)          $"VALVE DELIVERY  (M3/S)             "
            OV(N)=Q(N)/(A+KL*MU/A)$"OUTPUT VELOCITY  (M/S)              "
410..CONTINUE
      CALL XFERB(DOP=OV,6)
      OP=INTVC(DOP,OVIC)           $"OUTPUT POSITION  (M)               "
DO 420 N=1,6
            AM(N)=OP(N)+ALNEUT     $"REAL ACTUATOR LENGTH  (M)          "
            LERR(N)=IP(N)-OP(N)    $"LENGTH ERROR  (M)                  "
420..CONTINUE
END
END $"OF DERIVATIVE SECTION 4 "



            "SECTION 5"
""
"--   REAL ACTUATOR TO REAL PLATFORM TRANSFORMATION   ------------------"
""
"-    THE REAL ACTUATOR LENGTHS ARE TRANSFORMED BACK TO ACTUAL          "
"-    PLATFORM POSITION .                                               "
""
     PROCEDURAL(PR,FLG=AM,IN)
      CALL ACPLAT(PR,FLG=AM,IN)
      DO 510 N=1,6
            PERR(N)=PR(N)-P(N)
 510..CONTINUE
```

```
      END


"     PROGRAM TERMINATION CONDITION                                        "
      TERMT (T.GE.TF-CINT)
END   $ "END OF DYNAMIC "
END   $ "END OF PROGRAM "
```

```
***********************************************************************
**    11111111111111111111111111111111111111111111111111   ** AIRPI **
**                                                                  **
**                                                                  **
**      TRANSFER OF AIRCRAFT TO PILOT MOTION                        **
**                                                                  **
C       SUBROUTINE AIRPI(DXA,DDXA,CRX,DCRX,DDCRX,
C     .                  DXP,DDXP,XD)
C
C
C       THIS SUBROUTINE CALCULATES THE AIRCRAFT-PILOT MOTION
C       AS A RESULT OF THE AIRCRAFT MOTION . IT TAKES INTO ACCOUNT
C       THE RELATIVE POSITIONS OF THE AIRCRAFT-PILOT'S CENTER OF
C       PERCEPTION AND THE CENTER OF ROTATION OF THE AIRCRAFT .
C       PROVISION IS MADE IN THE PROGRAM FOR A CHANGING
C       RELATIONSHIP BETWEEN THE AIRCRAFT-PILOT'S POSITION WITH
C       RESPECT TO TO THE CENTER OF ROTATION OF THE AIRCRAFT .
C
C       THE EQUATIONS ARE OF THE FORM :-
C       TRANSL.   [DDXP]= [A].([DDCRX]+[DDXA]+[W].[DXA]+
C                             2.[W].[DCRX]+([DW]+[WW]).[XD])
C       ROT.       [DXP]= [DXA]
C       ROT.      [DDXP]= [DDXA]
C
C
C                WHERE:- [XD]= PILOT POSITION IN AIRCRAFT
C                            = [CRX]+[PX]
C                AND    [CRX]= POSITION OF C.R. IN AIRCRAFT
C                        [PX]= POSITION OF PILOT IN AIRCRAFT
C                         [A]= AXIS TRANSFORMATION MATRIX
C                         [W]= ANGULAR ROTATION MATRIX
C                        [DW]= D([W])/DT
C                       [DDW]= D([DW])/DT
C                        [XA]= AIRCRAFT MOTION IN 6 D.O.F.
C                       [DXA]= D([XA])/DT
C                      [DDXA]= D([DXA])/DT
C                        [XP]= AIRCRAFT-PILOT MOTION IN 6 D.O.F.
C                       [DXP]= D([XP])/DT
C                      [DDXP]= D([DXP])/DT
C
C
C
C       INPUT  -     XA(1,2&3)   (M)
C                   DXA(1,2&3)   (M/S)
C                  DDXA(1,2&3)   (M/S/S)     AIRCRAFT POSITION,
C                    XA(4,5&6)   ( )         RATES AND ACCELERATIONS W.R.T.
C                   DXA(4,5&6)   ( /S)       THE INERTIAL REFERENCE FRAME.
C                  DDXA(4,5&6)   ( /S/S)
C
C                   CRX(1,2&3)   (M)         MOVEMENT OF C.R.
C                  DCRX(1,2&3)   (M/S)       W.R.T. THE AIRCRAFT
C                 DDCRX(1,2&3)   (M/S/S)     REFERENCE FRAME.
C
C       OUTPUT -     XP(1,2&3)   (M)
C                   DXP(1,2&3)   (M/S)       AIRCRAFT-PILOT POSITION,
C                  DDXP(1,2&3)   (M/S/S)     RATES AND ACCELERATIONS
C                    XP(4,5&6)   ( )         W.R.T. THE FIXED AIRCRAFT
C                   DXP(4,5&6)   ( /S)       REFERENCE FRAME.
C                  DDXP(4,5&6)   ( /S/S)
C
C                    XD(1,2&3)   (M)         PILOT CENTRE OF PERCEPTION
C                    XD(4,5&6)   ( )         W.R.T. THE CENTRE OF ROTATION.
C
```

```
      DIMENSION DXA(6),DDXA(6)
     .          ,CRX(3),DCRX(3),DDCRX(3)
     .          ,DXP(6),DDXP(6)
      DIMENSION PX(6),XI(6),S(6),SS(6),W(3,3)
     .          ,XD(6)
     .          ,X(6),DX(6)
     .          ,WW(3,3),WS(3,3),DW(3,3)
C
C     DATA    -  PX(1,2&3)   (M)       PILOT POSITION IN AIRCRAFT
C                                      W.R.T. FIXED AIRCRAFT
C                                      REFERENCE FRAME.
C
      DATA (PX(I),I=1,3)/1.5,0.,-1./
      DO 101 I=1,3
         XD(I)=CRX(I)+PX(I)
101   CONTINUE
**
**    SET UP MATRICES FOR TRANSFORMATONS
**
*
*     SET UP MATRIX W(3,3)
*
      W(1,1)= 0.0
      W(1,2)=-DXA(6)
      W(1,3)= DXA(5)
      W(2,1)=-W(1,2)
      W(2,2)= 0.0
      W(2,3)=-DXA(4)
      W(3,1)=-W(1,3)
      W(3,2)=-W(2,3)
      W(3,3)= 0.0
*
*     SET UP MATRIX DW(3,3)
*
      DW(1,1)= 0.0
      DW(1,2)=-DDXA(6)
      DW(1,3)= DDXA(5)
      DW(2,1)=-DW(1,2)
      DW(2,2)= 0.0
      DW(2,3)=-DDXA(4)
      DW(3,1)=-DW(1,3)
      DW(3,2)=-DW(2,3)
      DW(3,3)= 0.0
*
*     SET UP MATRIX WW(3,3)
*
      DO 102 I=1,3
            DO 102 J=1,3
                  WW(I,J)=00.0
                  DO 102 K=1,3
                        WS(I,J)=W(I,K)*W(K,J)
                        WW(I,J)=WW(I,J)+WS(I,J)
102   CONTINUE
**
**    PERFORM TRANSFORMATIONS :-
**    AIRCRAFT-PILOT TO SIMULATOR-PILOT MOTION.
**
*
*     CALCULATION OF RATES OF CHANGE
*
      DO 108 I=1,3
            SS(I)=0.0
            DO 109 J=1,3
```

```
                 S(J)=2.0*W(I,J)*DCRX(J)+
                 (DW(I,J)+WW(I,J))*XD(J)
                 SS(I)=SS(I)+S(J)
109           CONTINUE
              DDXP(I)=DDCRX(I)+DDXA(I)+SS(I)
108     CONTINUE
*
*       CALCULATION OF ROTATIONS
*
        DO 112 I=4,6
              DXP(I)=DXA(I)
              DDXP(I)=DDXA(I)
112     CONTINUE
        RETURN
        END
```

```
*******************************************************************
**    333333333333333333333333333333333333333333333333   ** PIPLAT **
**                                                               **
**     SIMULATOR PILOT TO PLATFORM TRANSFORMATION                **
**                                                               **
      SUBROUTINE PIPLAT(XS,P,PI)
C
C
C     THIS SUBROUTINE CALCULATES THE MOTION OF THE CENTER OF
C     THE PLATFORM AS A RESULT OF THE SIMULATOR-PILOT'S MOTION .
C     THE PROGRAM TAKES INTO ACCOUNT THE DISPLACEMENT BETWEEN
C     THE SIMULATOR-PILOT'S CENTER OF PERCEPTION AND THE CENTER
C     OF THE PLATFORM . NO PROVISION IS MADE FOR RELATIVE MOVEMENT
C     OF THE SIMULATOR-PILOT WITH RESPECT TO THE PLATFORM .
C
C     THE EQUATIONS ARE OF THE FORM :-
C
C     TRANS.       [P]= [C].[YD]-[YD]+[XS]                  :
C     ROT          [P]= [XS]                                :
C
C
C               WHERE:- [YD]= S-PILOT POSITION ON PLATFORM
C                        [C]= AXIS TRANSFORMATION MATRIX
C                        [R]= ANGULAR ROTATION MATRIX
C                       [DR]= D([R])/DT
C                       [XS]= S-PILOT MOTION IN 6 D.O.F.
C                        [P]= PLATFORM MOTION IN 6 D.O.F.
C
C
C
C     INPUT  -    XS(1,2&3)  (M)        OPERATOR POSITION W.R.T
C                 XS(4,5&6)  ( )        INERTIAL SPACE
C
C     OUTPUT -     P(1,2&3)  (M)        PLATFORM POSITION W.R.T.
C                  P(4,5&6)  ( )        INERTIAL SPACE
C
      DIMENSION XS(6)
     .          ,P(6),PI(3)
      DIMENSION YD(6)
     .          ,T(3,3)
     .          ,S(3),Y(3)
C
C     DATA   -  YD(6)     (M)      PILOT POSITION ON SIMULATOR
C                                  PLATFORM W.R.T. PLATFORM REFERENCE
C                                  FRAME.
C
      DATA (YD(I),I=1,3)/0.0,0.0,3.00/
**
**    SET UP MATRICES FOR TRANSFORMATONS
**
*
*     SET UP TRANSFORMATION MATRIX T(3,3)
*
      T(1,1)= COS(XS(5))*COS(XS(6))
      T(1,2)= SIN(XS(4))*SIN(XS(5))*COS(XS(6))-COS(XS(4))*
     .        SIN(XS(6))
      T(1,3)= COS(XS(4))*SIN(XS(5))*COS(XS(6))+SIN(XS(4))*
     .        SIN(XS(6))
      T(2,1)= COS(XS(5))*SIN(XS(6))
      T(2,2)= SIN(XS(4))*SIN(XS(5))*SIN(XS(6))+COS(XS(4))*
     .        COS(XS(6))
      T(2,3)= COS(XS(4))*SIN(XS(5))*SIN(XS(6))-SIN(XS(4))*
     .        COS(XS(6))
```

```
      T(3,1)=-SIN(XS(5))
      T(3,2)= SIN(XS(4))*COS(XS(5))
      T(3,3)= COS(XS(4))*COS(XS(5))
*
*           CALCULATION OF DISPLACEMENTS
*
      DO 302 I=1,3
            Y(I)=0.0
            DO 303 J=1,3
                  S(J)=T(I,J)*(YD(J)+XS(J))
                  Y(I)=Y(I)+S(J)
303         CONTINUE
            P(I)=Y(I)-YD(I)
            PI(I)=Y(I)
302   CONTINUE
*
*     CALCULATION OF ROTATIONS
*
      DO 310 I=4,6
            P(I)=XS(I)
310   CONTINUE
      RETURN
      END
```

```
*****************************************************************************
**333333333333333333333333333333333333333333333333333333333333333** PACT **
**                                                                        **
**       CALCULATION OF ACTUATOR LENGTHS.                                 **
**                                                                        **
      SUBROUTINE PACT (P,AL)
C
C
C     THIS SUBROUTINE CALCULATES THE MOTION REQUIRED OF THE
C     ACTUATORS TO GIVE THE REQUIRED PLATFORM MOTION . THE
C     LENGTHS,RATES AND ACCELERATION OF EACH OF THE SIX ACTUATORS
C     ARE CALCULATED . THE MOTION OF THE CENTER OF THE PLATFORM IS
C     TRANSFERED TO THE MOUNTING POINTS OF EACH ACTUATOR AND THEN THE
C     RELATED ACTUATOR LENGTHS ARE DETERMINED .
C
C
C     THE EQUATIONS ARE OF THE FORM :-
C
C                 [PM]= [F].[ZD]+[P]              :
C
C               WHERE:- [ZD]= MOUNTING POSITIONS ON PLATFORM
C                        [F]= AXIS TRANSFORMATION MATRIX
C                        [Q]= ANGULAR ROTATION MATRIX
C                        [P]= PLATFORM MOTION IN 6 D.O.F.
C                       [PM]= MOUNTING POINT MOTION IN 3 D.O.F.
C
C                            AND :-
C                 AL = SQRT((B1-PM1)**2+(B2-PM2)**2+(B3-PM3)**2)
C
C               WHERE:-   B = ACTUATOR BASE POINTS
C                        AL = ACTUATOR LENGTH
C
C     INPUT  -   P(1,2&3)  (M)     PLATFORM POSITION
C                P(4,5&6)  ( )     W.R.T. INERTIAL SPACE
C
C     OUTPUT -   AL(1-6)   (M)     ACTUATOR LENGTHS
C
      DIMENSION P(6)
     .          ,AL(6)
      DIMENSION ZD(6,6),PM(6,6)
     .          ,PR(6)
     .          ,B(6,6),ALL(6,6)
     .          ,F(6,6)
     .          ,S(6),Z(6,6)
C
C     DATA   -  ZD(6,3)    (M)     ACTUATOR MOUNTING POINTS
C                                  ON PLATFORM
C     DATA   -   B(6,3)    (M)   . PLATFORM NEUTRAL POSITION
C                                  CONSTANTS.
C
      DATA ((ZD(I,J),J=1,3),I=1,6)/+1.126,-1.696,0.0,
     .                             +0.906,-1.823,0.0,
     .                             -2.031,-0.127,0.0,
     .                             -2.031,+0.127,0.0,
     .                             +0.906,+1.823,0.0,
     .                             +1.126,+1.696,0.0/
      DATA  ((B(I,J),J=1,3),I=1,6)/+2.310,-0.165,2.500,
     .                             -1.012,-2.084,2.500,
     .                             -1.298,-1.919,2.500,
     .                             -1.298,+1.919,2.500,
     .                             -1.012,+2.084,2.500,
     .                             +2.310,+0.165,2.500/
*
```

```
*      SET UP TRANSFORMATION MATRIX F(3,3)
*
       F(1,1)= COS(P(5))*COS(P(6))
       F(1,2)= SIN(P(4))*SIN(P(5))*COS(P(6))-COS(P(4))*
     .         SIN(P(6))
       F(1,3)= COS(P(4))*SIN(P(5))*COS(P(6))+SIN(P(4))*
     .         SIN(P(6))
       F(2,1)= COS(P(5))*SIN(P(6))
       F(2,2)= SIN(P(4))*SIN(P(5))*SIN(P(6))+COS(P(4))*
     .         COS(P(6))
       F(2,3)= COS(P(4))*SIN(P(5))*SIN(P(6))-SIN(P(4))*
     .         COS(P(6))
       F(3,1)=-SIN(P(5))
       F(3,2)= SIN(P(4))*COS(P(5))
       F(3,3)= COS(P(4))*COS(P(5))
**
**     PERFORM TRANSFORMATION :-
**     PLATFORM MOTION TO ACTUATOR LENGTH MOTION
**
*
*      CALCULATION OF ACTUATOR MOUNTING POINT MOTION
*
       DO 310 N=1,6
*
*             CALCULATION OF DISPLACEMENTS
*
              DO 311 I=1,3
                     Z(N,I)=0.0
                     DO 312 J=1,3
                            S(J)=F(I,J)*ZB(N,J)
                            Z(N,I)=Z(N,I)+S(J)
312                  CONTINUE
                     PM(N,I)=P(I)+Z(N,I)
311           CONTINUE
310    CONTINUE
*
*      CALCULATION OF ACTUATOR LENGTH MOTION
*
       DO 319 N=1,6
*
*             CALCULATION OF ACTUATOR LENGTHS
*
              AL(N)=0.0
              DO 320 I=1,3
                     ALL(N,I)=(B(N,I)-PM(N,I))**2.0
                     AL(N)=AL(N)+ALL(N,I)
320           CONTINUE
              AL(N)=SQRT(AL(N))
319    CONTINUE
       RETURN
       END
```

```
*******************************************************************
**    5555555555555555555555555555555555555555555555555  ** ACPLAT **
**                                                                **
**     CALCULATION OF  ACTUAL PLATFORM POSITION AS A RESULT       **
**     OF ACTUAL ACTUATOR LENGTHS                                 **
**                                                                **
      SUBROUTINE ACPLAT (AM,IN,PR,FLG)
      INTEGER FLG
C
C
C
C     EQUATIONS DESCRIBING (1) THE ACTUATOR LENGTHS AS FUNCTIONS
C     OF THE BASE AND PLATFORM MOUNTING POINTS ,
C                          (2) THE COPLANAR NATURE OF THE PLATFORM
C     MOUNTING POINTS AND  (3) THEIR SPATIAL RELATIONSHIP WITH EACH
C     OTHER ARE APPROXIMATED USING TAYLOR SERIES EXPANSIONS .
C     THE RESULT IS A SET OF EIGHTEEN EQUATIONS IN EIGHTEEN UNKNOWNS
C     WHICH CAN BE EXPRESSED IN MATRIX FORM AS FOLLOWS ,
C               [T(18,18)].[E(18)] = [RHS]
C     WHERE  E(18)=E(6,3)  IS THE ERROR BETWEEN THE CURRENT VARIABLE
C     ESTIMATION AND THE SOLUTION PREDICTED BY THE TRUNCATED
C     TAYLOR SERIES APPROXIMATION OF THE FUNCTION THE COEFFICIENTS
C     OF WHICH ARE EXPRESSED IN THE SQUARE MATRIX T(18,18).
C     SOLVING THE MATRIX EQUATIONS USING AN INITIAL GUESS , PSI(6,3)
C     FOR THE ROOTS AND UPDATING THE ESTIMATE USING THE CALCULATED
C     ERROR THE TRUE SOLUTION CAN BE APPROACHED .
C
C     THUS               E(N,I) = PS(N,I)(J+1)-PS(N,I)(J)
C       OR         PS(N,I)(J+1)= E(N,I)+PS(N,I)(J) ,
C     WHERE , PS(N,I)(J) IS THE J'TH GUESS AND PS(N,I)(J+1) IS THE
C     J+1'ST GUESS OF THE ROOT PS(N,I) .
C
C     THE PROGRAM CONTINUES THE ITTERATION UNTIL THE GREATEST ERROR
C     IS LESS THAN OR EQUAL TO E-5 . THE FINAL ESTIMATION OF THE ROOTS
C     OF THE EQUATIONS (PO) IS OBTAINED BY SUMMING THE ERROR AND THE
C     LAST ESTIMATE OF THE ROOT . POINTS P7 , P8 AND CP ARE THEN
C     GENERATED TO FACILITATE THE CALCULATION OF THE PLATFORM
C     DISPLACEMENTS AND ROTATIONS ( ABOUT THE THREE AXES ) WITH
C     RESPECT TO THE NEUTRAL PLATFOM POSITION .
C
C
C     INPUT  -      AM(1-6)   (M)  ACTUATOR LENGTHS
C                   PS(6,3)   (M)  CURRENT PLATFORM POSITION ESTIMATE
C
C     OUTPUT -    PR(1,2&3)   (M)  PLATFORM DISPLACEMENTS
C                 PR(4,5&6)   ( )  PLATFORM ROTATIONS
C
C
      DIMENSION AM(6),PR(6)
      DIMENSION B(6,3),D(6,6),
     .          T(20,20),
     .          PS(6,3),PSI(6,3),
     .          F(18),G(18),H(18),
     .          FD(18,6,3),GD(18,6,3),HD(18,6,3),
     .          E(6,3),PO(6,3),
     .          P7(6),P8(6),CP(6),CPNEUT(6)
*
*
*     DATA    -  D(6,6)
*                B(6,3)
*
      FLG=1
      DATA D78 /3.157/
```

```
      DATA ((PSI(N,I),I=1,3),N=1,6)/+1.126,-1.696,0.0,
     .                              +0.906,-1.823,0.0,
     .                              -2.031,-0.127,0.0,
     .                              -2.031,+0.127,0.0,
     .                              +0.906,+1.823,0.0,
     .                              +1.126,+1.696,0.0/
      DATA   ((B(N,I),I=1,3),N=1,6)/+2.310,-0.165,2.5,
     .                              -1.012,-2.084,2.5,
     .                              -1.298,-1.919,2.5,
     .                              -1.298,+1.919,2.5,
     .                              -1.012,+2.084,2.5,
     .                              +2.310,+0.165,2.5/
      DATA ((D(N,I),I=1,6),N=1,5)
     .     / 0.000 , 0.254 , 3.525 , 3.646 , 3.526 , 3.392 ,
     .       0.000 , 0.000 , 3.392 , 3.525 , 3.646 , 3.526 ,
     .       0.000 , 0.000 , 0.000 , 0.254 , 3.525 , 3.646 ,
     .       0.000 , 0.000 , 0.000 , 0.000 , 3.392 , 3.525 ,
     .       0.000 , 0.000 , 0.000 , 0.000 , 0.000 , 0.254 /
      IF (IN.GT.1) GO TO 555
      DO 501 N=1,6
            DO 501 I=1,3
                  PS(N,I)=PSI(N,I)
501   CONTINUE
***
***   SET UP 18 SIMULTANEOUS EQUATIONS TO DETERMINE THE PLATFORM
***   POSITION AS A FUNCTION OF THE ACTUATOR LENGTHS
***
**
**     SET UP ACTUATOR LENGTH EQUATIONS
**
*
*     EQN 1 - LENGTH OF ACTUATOR 1
*     EQN 2 -     "      "      "    2
*     EQN 3 -     "      "      "    3
*     EQN 4 -     "      "      "    4
*     EQN 5 -     "      "      "    5
*     EQN 6 -     "      "      "    6
*
555   DO 502 N=1,6
            F(N)= 0.0
            DO 503 I=1,3
                  F(N)=F(N)+(B(N,I)-PS(N,I))**2.0
503         CONTINUE
            F(N)=F(N)-AM(N)**2.0
502   CONTINUE
      DO 504 N=1,6
            DO 504 I=1,6
                  DO 505 J=1,3
                        FD(N,I,J)= 0.0
                        IF(I.EQ.N) FD(N,I,J)= 2.0*(PS(I,J)-B(I,J))
505               CONTINUE
504   CONTINUE
**
**     SET UP COPLANAR CONDITION EQUATIONS
**
*
*     SET ZERO ELEMENTS OF EQUATIONS = 0.0
*
      DO 506 N=7,9
            G(N)= 0.0
            DO 506 I=1,6
                  DO 506 J=1,3
                        GD(N,I,J)= 0.0
```

```
506   CONTINUE
*
*     EQN 7 - POINTS 1,2,3&4 ON PLATFORM COPLANAR
*
      G(7)=PS(1,1)*( PS(2,2)*PS(3,3)-PS(2,2)*PS(4,3)-PS(2,3)*PS(3,2)
     .               +PS(2,3)*PS(4,2)+PS(3,2)*PS(4,3)-PS(3,3)*PS(4,2))-
     .     PS(1,2)*( PS(2,1)*PS(3,3)-PS(2,1)*PS(4,3)-PS(2,3)*PS(3,1)
     .               +PS(2,3)*PS(4,1)+PS(3,1)*PS(4,3)-PS(3,3)*PS(4,1))+
     .     PS(1,3)*( PS(2,1)*PS(3,2)-PS(2,1)*PS(4,2)-PS(2,2)*PS(3,1)
     .               +PS(2,2)*PS(4,1)+PS(3,1)*PS(4,2)-PS(3,2)*PS(4,1))
     .               -PS(2,1)*PS(3,2)*PS(4,3)+PS(2,1)*PS(3,3)*PS(4,2)
     .               +PS(2,2)*PS(3,1)*PS(4,3)-PS(2,2)*PS(3,3)*PS(4,1)
     .               -PS(2,3)*PS(3,1)*PS(4,2)+PS(2,3)*PS(3,2)*PS(4,1)
      GD(7,1,1)= PS(2,2)*PS(3,3)-PS(2,2)*PS(4,3)-PS(2,3)*PS(3,2)
     .          +PS(2,3)*PS(4,2)+PS(3,2)*PS(4,3)-PS(3,3)*PS(4,2)
      GD(7,1,2)=-PS(2,1)*PS(3,3)+PS(2,1)*PS(4,3)+PS(2,3)*PS(3,1)
     .          -PS(2,3)*PS(4,1)-PS(3,1)*PS(4,3)+PS(3,3)*PS(4,1)
      GD(7,1,3)= PS(2,1)*PS(3,2)-PS(2,1)*PS(4,2)-PS(2,2)*PS(3,1)
     .          +PS(2,2)*PS(4,1)+PS(3,1)*PS(4,2)-PS(3,2)*PS(4,1)
      GD(7,2,1)= PS(1,2)*(PS(4,3)-PS(3,3))+PS(1,3)*(PS(3,2)-PS(4,2))
     .          -PS(3,2)*PS(4,3)+PS(3,3)*PS(4,2)
      GD(7,2,2)= PS(1,1)*(PS(3,3)-PS(4,3))+PS(1,3)*(PS(3,1)+PS(4,1))
     .          +PS(3,1)*PS(4,3)-PS(3,3)*PS(4,1)
      GD(7,2,3)= PS(1,1)*(PS(4,2)-PS(3,2))+PS(1,2)*(PS(3,1)-PS(4,1))
     .          -PS(3,1)*PS(4,2)+PS(3,2)*PS(4,1)
      GD(7,3,1)= PS(1,2)*(PS(2,3)-PS(4,3))+PS(1,3)*(PS(4,2)-PS(2,2))
     .          +PS(2,2)*PS(4,3)-PS(2,3)*PS(4,2)
      GD(7,3,2)= PS(1,1)*(PS(4,3)-PS(2,3))+PS(1,3)*(PS(2,1)-PS(4,1))
     .          -PS(2,1)*PS(4,3)+PS(2,3)*PS(4,1)
      GD(7,3,3)= PS(1,1)*(PS(2,2)-PS(4,2))+PS(1,2)*(PS(4,1)-PS(2,1))
     .          +PS(2,1)*PS(4,2)-PS(2,2)*PS(4,1)
      GD(7,4,1)= PS(1,2)*(PS(3,3)-PS(2,3))+PS(1,3)*(PS(2,2)-PS(3,2))
     .          -PS(2,2)*PS(3,3)+PS(2,3)*PS(3,2)
      GD(7,4,2)= PS(1,1)*(PS(2,3)-PS(3,3))+PS(1,3)*(PS(3,1)-PS(2,1))
     .          +PS(2,1)*PS(3,3)-PS(2,3)*PS(3,1)
      GD(7,4,3)= PS(1,1)*(PS(3,2)-PS(2,2))+PS(1,2)*(PS(2,1)-PS(3,1))
     .          -PS(2,1)*PS(3,2)+PS(2,2)*PS(3,1)
*
*     EQN 8 - POINTS 1,2,3&5 ON PLATFORM COPLANAR
*
      G(8)=PS(1,1)*( PS(2,2)*PS(3,3)-PS(2,2)*PS(5,3)-PS(2,3)*PS(3,2)
     .               +PS(2,3)*PS(5,2)+PS(3,2)*PS(5,3)-PS(3,3)*PS(5,2))-
     .     PS(1,2)*( PS(2,1)*PS(3,3)-PS(2,1)*PS(5,3)-PS(2,3)*PS(3,1)
     .               +PS(2,3)*PS(5,1)+PS(3,1)*PS(5,3)-PS(3,3)*PS(5,1))+
     .     PS(1,3)*( PS(2,1)*PS(3,2)-PS(2,1)*PS(5,2)-PS(2,2)*PS(3,1)
     .               +PS(2,2)*PS(5,1)+PS(3,1)*PS(5,2)-PS(3,2)*PS(5,1))
     .               -PS(2,1)*PS(3,2)*PS(5,3)+PS(2,1)*PS(3,3)*PS(5,2)
     .               +PS(2,2)*PS(3,1)*PS(5,3)-PS(2,2)*PS(3,3)*PS(5,1)
     .               -PS(2,3)*PS(3,1)*PS(5,2)+PS(2,3)*PS(3,2)*PS(5,1)
      GD(8,1,1)= PS(2,2)*PS(3,3)-PS(2,2)*PS(5,3)-PS(2,3)*PS(3,2)
     .          +PS(2,3)*PS(5,2)+PS(3,2)*PS(5,3)-PS(3,3)*PS(5,2)
      GD(8,1,2)=-PS(2,1)*PS(3,3)+PS(2,1)*PS(5,3)+PS(2,3)*PS(3,1)
     .          -PS(2,3)*PS(5,1)-PS(3,1)*PS(5,3)+PS(3,3)*PS(5,1)
      GD(8,1,3)= PS(2,1)*PS(3,2)-PS(2,1)*PS(5,2)-PS(2,2)*PS(3,1)
     .          +PS(2,2)*PS(5,1)+PS(3,1)*PS(5,2)-PS(3,2)*PS(5,1)
      GD(8,2,1)= PS(1,2)*(PS(5,3)-PS(3,3))+PS(1,3)*(PS(3,2)-PS(5,2))
     .          -PS(3,2)*PS(5,3)+PS(3,3)*PS(5,2)
      GD(8,2,2)= PS(1,1)*(PS(3,3)-PS(5,3))+PS(1,3)*(PS(3,1)+PS(5,1))
     .          +PS(3,1)*PS(5,3)-PS(3,3)*PS(5,1)
      GD(8,2,3)= PS(1,1)*(PS(5,2)-PS(3,2))+PS(1,2)*(PS(3,1)-PS(5,1))
     .          -PS(3,1)*PS(5,2)+PS(3,2)*PS(5,1)
      GD(8,3,1)= PS(1,2)*(PS(2,3)-PS(5,3))+PS(1,3)*(PS(5,2)-PS(2,2))
     .          +PS(2,2)*PS(5,3)-PS(2,3)*PS(5,2)
```

```
      GD(8,3,2)= PS(1,1)*(PS(5,3)-PS(2,3))+PS(1,3)*(PS(2,1)-PS(5,1))
     .           -PS(2,1)*PS(5,3)+PS(2,3)*PS(5,1)
      GD(8,3,3)= PS(1,1)*(PS(2,2)-PS(5,2))+PS(1,2)*(PS(5,1)-PS(2,1))
     .           +PS(2,1)*PS(5,2)-PS(2,2)*PS(5,1)
      GD(8,5,1)= PS(1,2)*(PS(3,3)-PS(2,3))+PS(1,3)*(PS(2,2)-PS(3,2))
     .           -PS(2,2)*PS(3,3)+PS(2,3)*PS(3,2)
      GD(8,5,2)= PS(1,1)*(PS(2,3)-PS(3,3))+PS(1,3)*(PS(3,1)-PS(2,1))
     .           +PS(2,1)*PS(3,3)-PS(2,3)*PS(3,1)
      GD(8,5,3)= PS(1,1)*(PS(3,2)-PS(2,2))+PS(1,2)*(PS(2,1)-PS(3,1))
     .           -PS(2,1)*PS(3,2)+PS(2,2)*PS(3,1)
*
*     EQN 9 - POINTS 1,2,3&6 ON PLATFORM COPLANAR
*
      G(9)=PS(1,1)*( PS(2,2)*PS(3,3)-PS(2,2)*PS(6,3)-PS(2,3)*PS(3,2)
     .               +PS(2,3)*PS(6,2)+PS(3,2)*PS(6,3)-PS(3,3)*PS(6,2))-
     .     PS(1,2)*( PS(2,1)*PS(3,3)-PS(2,1)*PS(6,3)-PS(2,3)*PS(3,1)
     .               +PS(2,3)*PS(6,1)+PS(3,1)*PS(6,3)-PS(3,3)*PS(6,1))+
     .     PS(1,3)*( PS(2,1)*PS(3,2)-PS(2,1)*PS(6,2)-PS(2,2)*PS(3,1)
     .               +PS(2,2)*PS(6,1)+PS(3,1)*PS(6,2)-PS(3,2)*PS(6,1))
     .               -PS(2,1)*PS(3,2)*PS(6,3)+PS(2,1)*PS(3,3)*PS(6,2)
     .               +PS(2,2)*PS(3,1)*PS(6,3)-PS(2,2)*PS(3,3)*PS(6,1)
     .               -PS(2,3)*PS(3,1)*PS(6,2)+PS(2,3)*PS(3,2)*PS(6,1)
      GD(9,1,1)= PS(2,2)*PS(3,3)-PS(2,2)*PS(6,3)-PS(2,3)*PS(3,2)
                 +PS(2,3)*PS(6,2)+PS(3,2)*PS(6,3)-PS(3,3)*PS(6,2)
      GD(9,1,2)=-PS(2,1)*PS(3,3)+PS(2,1)*PS(6,3)+PS(2,3)*PS(3,1)
     .           -PS(2,3)*PS(6,1)-PS(3,1)*PS(6,3)+PS(3,3)*PS(6,1)
      GD(9,1,3)= PS(2,1)*PS(3,2)-PS(2,1)*PS(6,2)-PS(2,2)*PS(3,1)
     .           +PS(2,2)*PS(6,1)+PS(3,1)*PS(6,2)-PS(3,2)*PS(6,1)
      GD(9,2,1)= PS(1,2)*(PS(6,3)-PS(3,3))+PS(1,3)*(PS(3,2)-PS(6,2))
     .           -PS(3,2)*PS(6,3)+PS(3,3)*PS(6,2)
      GD(9,2,2)= PS(1,1)*(PS(3,3)-PS(6,3))+PS(1,3)*(PS(3,1)+PS(6,1))
     .           +PS(3,1)*PS(6,3)-PS(3,3)*PS(6,1)
      GD(9,2,3)= PS(1,1)*(PS(6,2)-PS(3,2))+PS(1,2)*(PS(3,1)-PS(6,1))
                 -PS(3,1)*PS(6,2)+PS(3,2)*PS(6,1)
      GD(9,3,1)= PS(1,2)*(PS(2,3)-PS(6,3))+PS(1,3)*(PS(6,2)-PS(2,2))
     .           +PS(2,2)*PS(6,3)-PS(2,3)*PS(6,2)
      GD(9,3,2)= PS(1,1)*(PS(6,3)-PS(2,3))+PS(1,3)*(PS(2,1)-PS(6,1))
     .           -PS(2,1)*PS(6,3)+PS(2,3)*PS(6,1)
      GD(9,3,3)= PS(1,1)*(PS(2,2)-PS(6,2))+PS(1,2)*(PS(6,1)-PS(2,1))
     .           +PS(2,1)*PS(6,2)-PS(2,2)*PS(6,1)
      GD(9,6,1)= PS(1,2)*(PS(3,3)-PS(2,3))+PS(1,3)*(PS(2,2)-PS(3,2))
     .           -PS(2,2)*PS(3,3)+PS(2,3)*PS(3,2)
      GD(9,6,2)= PS(1,1)*(PS(2,3)-PS(3,3))+PS(1,3)*(PS(3,1)-PS(2,1))
                 +PS(2,1)*PS(3,3)-PS(2,3)*PS(3,1)
      GD(9,6,3)= PS(1,1)*(PS(3,2)-PS(2,2))+PS(1,2)*(PS(2,1)-PS(3,1))
     .           -PS(2,1)*PS(3,2)+PS(2,2)*PS(3,1)
**
**     SET UP MOUNTING POINT DISTANCES
**
*
*     SET ZERO ELEMENTS OF EQUATIONS = 0.0
*
      DO 507 N=10,18
            H(N)= 0.0
            DO 507 I=1,6
                  DO 507 J=1,3
                        HD(N,I,J)= 0.0
507   CONTINUE
*
*     EQN  10 - DISTANCE BETWEEN MOUNTING POINTS 1&2
*
      H(10)= (PS(1,1)-PS(2,1))**2.0+(PS(1,2)-PS(2,2))**2.0
     .      +(PS(1,3)-PS(2,3))**2.0-D(1,2)**2.0
```

```
      HD(10,1,1)= 2.0*(PS(1,1)-PS(2,1))
      HD(10,1,2)= 2.0*(PS(1,2)-PS(2,2))
      HD(10,1,3)= 2.0*(PS(1,3)-PS(2,3))
      HD(10,2,1)=-HD(10,1,1)
      HD(10,2,2)=-HD(10,1,2)
      HD(10,2,3)=-HD(10,1,3)
*
*     EQN  11 - DISTANCE BETWEEN MOUNTING POINTS 1&3
*
      H(11)= (PS(1,1)-PS(3,1))**2.0+(PS(1,2)-PS(3,2))**2.0
     .      +(PS(1,3)-PS(3,3))**2.0-D(1,3)**2.0
      HD(11,1,1)= 2.0*(PS(1,1)-PS(3,1))
      HD(11,1,2)= 2.0*(PS(1,2)-PS(3,2))
      HD(11,1,3)= 2.0*(PS(1,3)-PS(3,3))
      HD(11,3,1)=-HD(11,1,1)
      HD(11,3,2)=-HD(11,1,2)
      HD(11,3,3)=-HD(11,1,3)
*
*     EQN  12 - DISTANCE BETWEEN MOUNTING POINTS 1&5
*
      H(12)= (PS(1,1)-PS(5,1))**2.0+(PS(1,2)-PS(5,2))**2.0
     .      +(PS(1,3)-PS(5,3))**2.0-D(1,5)**2.0
      HD(12,1,1)= 2.0*(PS(1,1)-PS(5,1))
      HD(12,1,2)= 2.0*(PS(1,2)-PS(5,2))
      HD(12,1,3)= 2.0*(PS(1,3)-PS(5,3))
      HD(12,5,1)=-HD(12,1,1)
      HD(12,5,2)=-HD(12,1,2)
      HD(12,5,3)=-HD(12,1,3)
*
*     EQN  13 - DISTANCE BETWEEN MOUNTING POINTS 2&4
*
      H(13)= (PS(2,1)-PS(4,1))**2.0+(PS(2,2)-PS(4,2))**2.0
     .      +(PS(2,3)-PS(4,3))**2.0-D(2,4)**2.0
      HD(13,2,1)= 2.0*(PS(2,1)-PS(4,1))
      HD(13,2,2)= 2.0*(PS(2,2)-PS(4,2))
      HD(13,2,3)= 2.0*(PS(2,3)-PS(4,3))
      HD(13,4,1)=-HD(13,2,1)
      HD(13,4,2)=-HD(13,2,2)
      HD(13,4,3)=-HD(13,2,3)
*
*     EQN  14 - DISTANCE BETWEEN MOUNTING POINTS 2&6
*
      H(14)= (PS(2,1)-PS(6,1))**2.0+(PS(2,2)-PS(6,2))**2.0
     .      +(PS(2,3)-PS(6,3))**2.0-D(2,6)**2.0
      HD(14,2,1)= 2.0*(PS(2,1)-PS(6,1))
      HD(14,2,2)= 2.0*(PS(2,2)-PS(6,2))
      HD(14,2,3)= 2.0*(PS(2,3)-PS(6,3))
      HD(14,6,1)=-HD(14,2,1)
      HD(14,6,2)=-HD(14,2,2)
      HD(14,6,3)=-HD(14,2,3)
*
*     EQN  15 - DISTANCE BETWEEN MOUNTING POINTS 3&4
*
      H(15)= (PS(3,1)-PS(4,1))**2.0+(PS(3,2)-PS(4,2))**2.0
     .      +(PS(3,3)-PS(4,3))**2.0-D(3,4)**2.0
      HD(15,3,1)= 2.0*(PS(3,1)-PS(4,1))
      HD(15,3,2)= 2.0*(PS(3,2)-PS(4,2))
      HD(15,3,3)= 2.0*(PS(3,3)-PS(4,3))
      HD(15,4,1)=-HD(15,3,1)
      HD(15,4,2)=-HD(15,3,2)
      HD(15,4,3)=-HD(15,3,3)
*
*     EQN 16 - DISTANCE BETWEEN MOUNTING POINTS 3&5
```

```
*
      H(16)= (PS(3,1)-PS(5,1))**2.0+(PS(3,2)-PS(5,2))**2.0
     .       +(PS(3,3)-PS(5,3))**2.0-D(3,5)**2.0
      HD(16,3,1)= 2.0*(PS(3,1)-PS(5,1))
      HD(16,3,2)= 2.0*(PS(3,2)-PS(5,2))
      HD(16,3,3)= 2.0*(PS(3,3)-PS(5,3))
      HD(16,5,1)=-HD(16,3,1)
      HD(16,5,2)=-HD(16,3,2)
      HD(16,5,3)=-HD(16,3,3)
*
*     EQN 17 - DISTANCE BETWEEN MOUNTING POINTS 4&6
*
      H(17)= (PS(4,1)-PS(6,1))**2.0+(PS(4,2)-PS(6,2))**2.0
     .       +(PS(4,3)-PS(6,3))**2.0-D(4,6)**2.0
      HD(17,4,1)= 2.0*(PS(4,1)-PS(6,1))
      HD(17,4,2)= 2.0*(PS(4,2)-PS(6,2))
      HD(17,4,3)= 2.0*(PS(4,3)-PS(6,3))
      HD(17,6,1)=-HD(17,4,1)
      HD(17,6,2)=-HD(17,4,2)
      HD(17,6,3)=-HD(17,4,3)
*
*     EQN 18 - DISTANCE BETWEEN MOUNTING POINTS 5&6
*
      H(18)= (PS(5,1)-PS(6,1))**2.0+(PS(5,2)-PS(6,2))**2.0
     .       +(PS(5,3)-PS(6,3))**2.0-D(5,6)**2.0
      HD(18,5,1)= 2.0*(PS(5,1)-PS(6,1))
      HD(18,5,2)= 2.0*(PS(5,2)-PS(6,2))
      HD(18,5,3)= 2.0*(PS(5,3)-PS(6,3))
      HD(18,6,1)=-HD(18,5,1)
      HD(18,6,2)=-HD(18,5,2)
      HD(18,6,3)=-HD(18,5,3)
**
**     SET UP MATRIX T(18,18)
**
      DO 508 N=1,6
             K=1
             DO 509 I=1,6
                    DO 509 J=1,3
                           T(N,K)=FD(N,I,J)
                           K=K+1
509          CONTINUE
             T(N,K)=-F(N)
508   CONTINUE
C
      DO 510 N=7,9
             K=1
             DO 511 I=1,6
                    DO 511 J=1,3
                           T(N,K)=GD(N,I,J)
                           K=K+1
511          CONTINUE
             T(N,K)=-G(N)
510   CONTINUE
C
      DO 512 N=10,18
             K=1
             DO 513 I=1,6
                    DO 513 J=1,3
                           T(N,K)=HD(N,I,J)
                           K=K+1
513          CONTINUE
             T(N,K)=-H(N)
512   CONTINUE
```

```
***
***     SOLUTION OF LINEARISED EQUATIONS
***
**
**         C.F. GERALD  APPLIED NUMERICAL ANALYSIS 2ND ED.
**                      PP132
**
*
*      ELIMINATE MATRIX VALUES LESS THAN 0.0001
*
       DO 544 N=1,18
             DO 544 I=1,19
                   IF(ABS(T(N,I)).LT.1.0E-4) T(N,I)= 0.0
544    CONTINUE
*
*      BEGIN REDUCTION OF THE MATRIX DIAGONAL DOMINENCE BY
*      CHANGIMG ROWS
*
       DO 514 I=1,17
             IPVT= I
             IP1= I+1
             DO 515 J=IP1,18
                   IF(ABS(T(IPVT,I)).LT.ABS(T(J,I))) IPVT= J
515          CONTINUE
             IF ((ABS(T(IPVT,I)).LT.1.0E-5)) GO TO 777
             IF (IPVT.EQ.I) GO TO 551
             DO 516 JCOL= I,19
                   SAVE= T(I,JCOL)
                   T(I,JCOL)= T(IPVT,JCOL)
                   T(IPVT,JCOL)= SAVE
516          CONTINUE
*
*            REDUCE DIAGONAL TO UNITY
*
551          DO 517 JROW= IP1,18
                   IF(T(JROW,I).EQ.0) GO TO 517
                   RATIO= T(JROW,I)/T(I,I)
                   DO 518 KCOL= IP1,19
                         T(JROW,KCOL)= T(JROW,KCOL)-RATIO*T(I,KCOL)
518                CONTINUE
517          CONTINUE
514    CONTINUE
       IF(ABS(T(18,18)).LT.1.0E-5) GO TO 888
*
*      CALCULATE VARIABLES FROM 18TH TO 1ST
*
       T(18,19)=T(18,19)/T(18,18)
       DO 519 NN=2,18
             NVBL= 19-NN
             L= NVBL+1
             VALUE= T(NVBL,19)
             DO 520 K=L,18
                   VALUE= VALUE-T(NVBL,K)*T(K,19)
520          CONTINUE
             T(NVBL,19)= VALUE/T(NVBL,NVBL)
519    CONTINUE
*
*      SET UP ERROR MATRIX E(6,3)
*
       JROW=1
       DO 521 I=1,6
             DO 521 J=1,3
                   KCOL=19
```

```
                E(I,J)= T(JROW,KCOL)
                JROW=JROW+1
521     CONTINUE
        LER=0
        DO 522 I=1,6
              DO 522 J=1,3
                    IF(ABS(E(I,J)).GT.1.0E-5) LER=3
522     CONTINUE
*
*       IF ERROR GREATER THAN E-4 SET UP NEW GUESS (PS(6,3) MATRIX)
*
        IF(LER.GT.1) GO TO 552
           GO TO 553
552        DO 523 I=1,6
                 DO 523 J=1,3
                       PS(I,J)= E(I,J)+PS(I,J)
523        CONTINUE
           GO TO 555
*
*       ELSE IF ERROR ACCEPTABLE SET UP FINAL POS MATRIX PO(6,3)
*
553        DO 524 I=1,6
                 DO 524 J=1,3
                 PO(I,J)= E(I,J)+PS(I,J)
524        CONTINUE
***
***     DETERMINE POSITION OF PLATFORM RELATIVE TO THE
***     NEUTRAL POSITION
***
*
*       SET UP POINTS P7 ,P8 AND CP
*
        DO 525 I=1,3
              P7(I)= (PO(3,I)+PO(4,I))/2.0
              P8(I)= (PO(1,I)+PO(6,I))/2.0
              CP(I)= 0.64333*(P8(I)-P7(I))+P7(I)
525     CONTINUE
*
*       DETERMINE DISPLACEMENTS FROM NEUTRAL POSITION
*
        DATA (CPNEUT(I),I=1,3)/ 0.0,0.0,0.0/
        DO 526 I=1,3
              PR(I)= CP(I)-CPNEUT(I)
526     CONTINUE
*
*       DETERMINE ROTATIONS OF PLATFORM
*
        PR(4)= ASIN((PO(3,3)-PO(4,3))/D(3,4))
        PR(5)= ASIN((P7(3)-P8(3))/D78)
        PR(6)= ASIN((P7(2)-P8(2))/D78)
        DO 527 I=1,6
              IF(ABS(PR(I)).LT.1.0E-4) PR(I)=0.
527     CONTINUE
        DO 528 I=1,6
              DO 528 J=1,3
                    PS(I,J)=PO(I,J)
528     CONTINUE
        GO TO 999
777     FLG= 777
888     FLG= 888
999     RETURN
        END
20.34.31.UCLP, IR, CONB45,       1.328KLNS.        ** END OF LISTING **
```

D.2 WASHOUT FILTER MODEL

The *following* is a listing of the washout filter model. It was developed independently from the rest of the simulator model and then the two models were integrated.

```
PROGRAM   " MOTION TRANSFORMATION FILTER                                        "


INITIAL

      ARRAY      TSCC1(2),TSCC2(2),QN(3),QD(3)
      DIMENSION  DDXP(6),XS(6),VESTO(6),VESTI(6),VESTER(6)

      CONSTANT ACCTON=0.0
      CONSTANT ACCRON=1.0
      CONSTANT ACCT=2.00      $" ACCELERATION COMMAND (M/SS)                   "
      CONSTANT ACCR=0.05      $" ACCELERATION COMMAND (RAD/SS)                 "


      CONSTANT QN=1.0,0.0,0.0        $" HIGH PASS FILTER NUM ARRAY        "
      CONSTANT QD=1.0,0.64,0.16      $"  ,      ,       ,   DEN   ,       "
      CONSTANT TSCC1=1.,0.        $" SEMIC.CANAL FILTER NUM ARRAY         "
      CONSTANT TSCC2=1.,0.169        $" SEMIC.CANAL FILTER DEN ARRAY      "
      CONSTANT G0=2.16        $" VALUES FOR THE OTOLITH FILTER                 "
      CONSTANT A0=0.076       $"            ,                                  "
      CONSTANT B0=0.19        $"            ,                                  "
      CONSTANT GS=39.51       $"            ,                                  "
      CONSTANT BS=0.169       $"            ,                                  "
      CONSTANT G=-10.         $" GRAVITATIONAL CONSTANT                        "
      CONSTANT ETA=0.8       $" POSITION WASHOUT FILTER DAMPING RATIO "
      CONSTANT WN=1.00       $"  POSITION FILTER NATURAL FREQUENCY      "
      CONSTANT MK=1.0
      MK1=((MK/(WN**2.0))-1.)/MK     $" POSITION FILTER ACC LOOP GAIN"
      MK2=(2.0*ETA)/WN          $" POSITION FILTER VELOCITY LOOP GAIN     "
      CONSTANT KT4=0.6         $" ROLL TILT COMMAND GAIN                       "
      CONSTANT KT5=0.6         $" PITCH TILT COMMAND GAIN                      "
      CONSTANT KI4=1.0         $" ROLL TILT COMMAND SENCE                      "
      CONSTANT KI5=-1.0        $" PITCH TILT COMMAND SENCE                     "
      CONSTANT KI6=1.0         $" YAW TILT COMMAND SENCE                       "
      CONSTANT KAT=0.6         $" TRANS ACC COMMAND GAIN                       "
      CONSTANT KAR=0.6         $" ROT ACC COMMAND GAIN                         "

      CINTERVAL          CINT  =     0.050
      CONSTANT           TSTOP    =     10.0
END $"OF INITIAL"
DYNAMIC
DERIVATIVE
      NSTEPS NSTP = 5
"GENERATE TRANSLATIONAL AND ROTATIONAL ACCELERATIONS IN THE FRF"
PROCEDURAL(DDXP=ACCT,ACCTON,ACCR,ACCRON)
      DDXP(1)=CMPXPL(.10,0.51,(PULSE(1.,15.,5.))*ACCT*ACCTON)
      DDXP(2)=0.0
      DDXP(3)=0.0
      DDXP(4)=0.0
      DDXP(5)=(PULSE(0.,6.,2.)-.5)*ACCR*ACCRUN*2.
      DDXP(6)=0.0
END
PROCEDURAL(DDXP1,DDXP2,DDXP3,DDXP4,DDXP5,DDXP6=DDXP)
      DDXP1=DDXP(1)
      DDXP2=DDXP(2)
      DDXP3=DDXP(3)
      DDXP4=DDXP(4)
      DDXP5=CMPXPL(.1,.51,RSW(T.GE.4.,0.0,DDXP(5)))
      DDXP6=DDXP(6)
END
" CALCULATE THE AIRCRAFTS ROTATIONAL POSITION "
      DXP4=INTEG(DDXP4,0.0)
```

```
      DXP5=INTEG(DDXP5,0.0)
      DXP6=INTEG(DDXP6,0.0)
      "TRANSFORMING RATES FROM BODY TO INERTIAL AXES"
      DA4=DXP4+ SIN(A4)*TAN(A5) *DXP5+ COS(A4)*TAN(A5) *DXP6
      DA5=      COS(A4)          *DXP5- SIN(A4)         *DXP6
      DA6=     (SIN(A4)/COS(A5))*DXP5+(COS(A4)/COS(A5))*DXP6
      "INTEGRATING FOR POSITION"
      A4=INTEG(DA4,0.0)
      A5=INTEG(DA5,0.0)
      A6=INTEG(DA6,0.0)
" CALCULATE THE SPECIFIC FORCE INPUTS "
" PILOT VESTIBULAR INPUT "
     FSX1=DDXP1-         SIN(A5)*G
     FSX2=DDXP2+SIN(A4)*COS(A5)*G
     FSX3=DDXP3+COS(A4)*COS(A5)*G-G
" VESTIBULAR RESPONSE OF PILOT "
      YIOTO1=GO*(AO/BO)*LEDLAG(AO,BO,FSX1,0.0)
      YIOTO2=GO*(AO/BO)*LEDLAG(AO,BO,FSX2,0.0)
      YIOTO3=GO*(AO/BO)*LEDLAG(AO,BO,FSX3,0.0)
      YISCC4=GS*TRAN(1,1,TSCC1,TSCC2,DXP4)
      YISCC5=GS*TRAN(1,1,TSCC1,TSCC2,DXP5)
      YISCC6=GS*TRAN(1,1,TSCC1,TSCC2,DXP6)
" GENERATE HIGH AND LOW PASS COMPONENTS OF SPECIFIC FORCE "
      FH1=TRAN(2,2,QN,QD,FSX1)
      FH2=TRAN(2,2,QN,QD,FSX2)
      FH3=TRAN(2,2,QN,QD,FSX3)
      FSXL1=FSX1-FH1
      FSXL2=FSX2-FH2
      FSXL3=FSX3-FH3
" SUBTRACT GRAVITY COMPONENT OF PLATFORM POSITION FROM ...
  PLATFORM ACCELERATION COMMAND.  "
      FSXH1=KAT*FH1-(-           SIN(XNR5)*G)
      FSXH2=KAT*FH2-( SIN(XNR4)*COS(XNR5)*G)
      FSXH3=KAT*FH3-( COS(XNR4)*COS(XNR5)*G-G)
" FILTER THE SPECIFIC FORCE DRIVING SIGNAL FOR OUTPUT POSITION "
      DDS1=(FSXH1-MK*(MK2*DS1+S1))/(1.0+MK*MK1)
      DDS2=(FSXH2-MK*(MK2*DS2+S2))/(1.0+MK*MK1)
      DDS3=(FSXH3-MK*(MK2*DS3+S3))/(1.0+MK*MK1)
      DS1=INTEG(DDS1,0.0)
      DS2=INTEG(DDS2,0.0)
      DS3=INTEG(DDS3,0.0)
      S1=INTEG(DS1,0.0)
      S2=INTEG(DS2,0.0)
      S3=INTEG(DS3,0.0)
" TRANSFORM THE DISPLACEMENTS FROM BODY TO INERTIAL AXES "
      XS1=(           COS(XS5)*COS(XS6)                    )*S1 ...
         +(SIN(XS4)*SIN(XS5)*COS(XS6)-COS(XS4)*SIN(XS6))*S2 ...
         +(COS(XS4)*SIN(XS5)*COS(XS6)+SIN(XS4)*SIN(XS6))*S3
      XS2=(           COS(XS5)*SIN(XS6)                    )*S1 ...
         +(SIN(XS4)*SIN(XS5)*SIN(XS6)+COS(XS4)*COS(XS6))*S2 ...
         +(COS(XS4)*SIN(XS5)*SIN(XS6)-SIN(XS4)*COS(XS6))*S3
      XS3=(                           -             SIN(XS5))*S1 ...
         +(                         SIN(XS4)*COS(XS5))*S2 ...
         +(                         COS(XS4)*COS(XS5))*S3
" GENERATE TILT POS AND RATE CMDS AS FUNCTION OF SPECIFIC FORCE "
      PT4=KI4*KT4*(FSXL2/G)
      PT5=KI5*KT5*(FSXL1/G)
      PT6=0.0
" GENERATE HIGH PASS COMPONENT OF AIRCRAFT ROTATIONS "
      DDPH4=TRAN(2,2,QN,QD,DDXP4)
      DDPH5=TRAN(2,2,QN,QD,DDXP5)
      DDPH6=TRAN(2,2,QN,QD,DDXP6)
" GENERATE THE ROTATIONAL VELOCITY COMMAND FROM THE ...
```

```
  AIRCRAFTS ROTATIONAL ACCELERATION "
       DS4=INTEG(DDPH4,0.0)
       DS5=INTEG(DDPH5,0.0)
       DS6=INTEG(DDPH6,0.0)
" TRANSFORM ROTATIONAL RATES FROM BODY TO INERTIAL AXES "
       DXS4=DS4+ SIN(XNR4)*TAN(XNR5) *DS5+ COS(XNR4)*TAN(XNR5) *DS6
       DXS5=     COS(XNR4)           *DS5- SIN(XNR4)           *DS6
       DXS6=    (SIN(XNR4)/COS(XNR5))*DS5+(COS(XNR4)/COS(XNR5))*DS6
" INTEGRATE FOR POSITION "
       XNR4=INTEG(DXS4,0.0)
       XNR5=INTEG(DXS5,0.0)
       XNR6=INTEG(DXS6,0.0'
" ADD TILT COMPONENT TO POSITION COMMAND "
       XS4=XNR4+PT4
       XS5=XNR5+PT5
       XS6=XNR6
" DIF. TO OBTAIN ROTATIONAL VELOCITY IN BODY REF FRAME "
       XV4=DERIVT(0.0,XS4)
       XV5=DERIVT(0.0,XS5)
       XV6=DERIVT(0.0,XS6)
" TRANSFORM FROM INERTIAL TO BODY REF FRAMES "
       DV4=XV4+                          SIN(XS5)*XV6
       DV5=    COS(XS4)*XV5+SIN(XS4)*COS(XS5)*XV6
       DV6=   -SIN(XS4)*XV5+COS(XS4)*COS(XS5)*XV6
" CALCULATE THE SPECIFIC FORCE EXPERIENCED BY THE OPERATOR "
       FXS1=DDS1-          SIN(XS5)*G
       FXS2=DDS2+SIN(XS4)*COS(XS5)*G
       FXS3=DDS3+COS(XS4)*COS(XS5)*G-G
" VESTIBULAR RESPONCE OF OPERATOR "
       YOOTO1=GO*(AO/BO)*LEDLAG(AO,BO,FXS1,0.0)
       YOOTO2=GO*(AO/BO)*LEDLAG(AO,BO,FXS2,0.0)
       YOOTO3=GO*(AO/BO)*LEDLAG(AO,BO,FXS3,0.0)
       YOSCC4=GS*TRAN(1,1,TSCC1,TSCC2,DV4)
       YOSCC5=GS*TRAN(1,1,TSCC1,TSCC2,DV5)
       YOSCC6=GS*TRAN(1,1,TSCC1,TSCC2,DV6)
PROCEDURAL(VESTER=YIOTO1,YIOTO2,YIOTO3,YOOTO1,YOOTO2,YOOTO3 ...
             ,YISCC4,YISCC5,YISCC6,YOSCC4,YOSCC5,YOSCC6)
       VESTER(1)=YIOTO1-YOOTO1
       VESTER(2)=YIOTO2-YOOTO2
       VESTER(3)=YIOTO3-YOOTO3
       VESTER(4)=YISCC4-YOSCC4
       VESTER(5)=YISCC5-YOSCC5
       VESTER(6)=YISCC6-YOSCC6
END
PROCEDURAL(XS=XS1,XS2,XS3,XS4,XS5,XS6)
       XS(1)=XS1
       XS(2)=XS2
       XS(3)=XS3
       XS(4)=XS4
       XS(5)=XS5
       XS(6)=XS6
END
PROCEDURAL(VESTI,VESTO=YIOTO1,YIOTO2,YIOTO3,YOOTO1,YOOTO2,YOOTO3 ...
                   ,YISCC4,YISCC5,YISCC6,YOSCC4,YOSCC5,YOSCC6)
       VESTI(1)=YIOTO1
       VESTI(2)=YIOTO2
       VESTI(3)=YIOTO3
       VESTI(4)=YISCC4
       VESTI(5)=YISCC5
       VESTI(6)=YISCC6
       VESTO(1)=YOOTO1
       VESTO(2)=YOOTO2
       VESTO(3)=YOOTO3
```

```
      VESTO(4)=YOSCC4
      VESTO(5)=YOSCC5
      VESTO(6)=YOSCC6
END

      TERMT(T.GE.(TSTOP-CINT))
END  $"OF DERIVATIVE"
END  $"OF DYNAMIC"
END  $"OF PROGRAM"
20.22.45.UCLP, IR, CONB45,        0.233KLNS.         ** END OF LISTING **
```

D.3 ACTUATOR MODEL

The ACSL implementation of the Actuator as in Figure 2.7.1 is listed below.

```
PROGRAM    " ACTUATOR CONTROL MODEL "
INITIAL

      CONSTANT A = .0025 $"ACTUATOR AREA "
      CONSTANT ALNEUT = 3.162 $"ACTUATOR NEUTRAL LENGTH "
      CONSTANT K =20.0 $"CURRENT GAIN "
      CONSTANT KR = 0.5 $"RATE FEEDBACK GAIN "
      CONSTANT KV = 3.8E-3 $"VALVE GAIN "
      CONSTANT KL = 1.5E-11 $"LEAKAGE GAIN "
      CONSTANT MU = 10.0 $"VISCOSITY "
      CONSTANT OV = 0.0

      CINTERVAL          CINT  =    0.010
      CONSTANT           TSTOP   =    0.1
END $"OF INITIAL"
DYNAMIC
DERIVATIVE
      NSTEPS NSTP = 5
      IP=PULSE(0.,10.,5.)*0.075
      ERR=IP-OP
      CI=K*ERR
      Q=KV*CI
      OV=Q/(A+KL*MU/A)
      OP=INTEG(OV,0.0)
      LERR=IP-OP
"ACTUATOR MODEL "

      TERMT(T.GE.(TSTOP-CINT))
END  $"OF DERIVATIVE"
END  $"OF DYNAMIC"
END  $"OF PROGRAM"
20.38.17.UCLP, IR, CONB45,       0.066KLNS.        ** END OF LISTING **
```

APPENDIX E

PLOTS

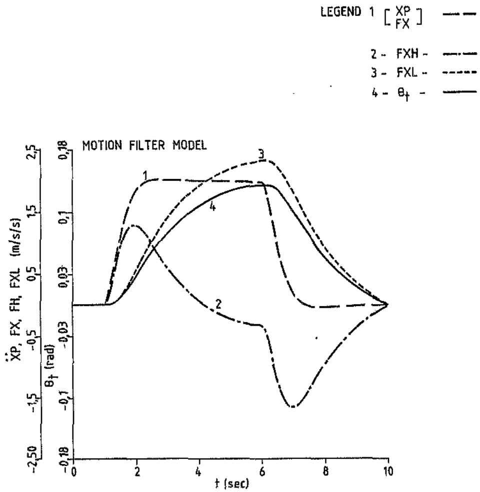

FIGURE E.1.1 : SURGE ACCELERATION INPUT AND WASHOUT.

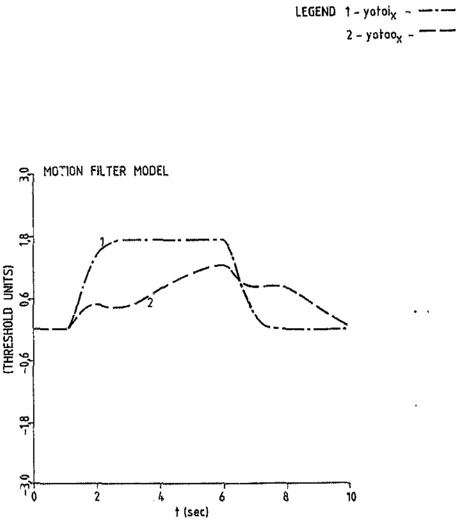

FIGURE E.1.2 : SURGE AXIS VESTIBULAR RESPONSE (SURGE INPUT)

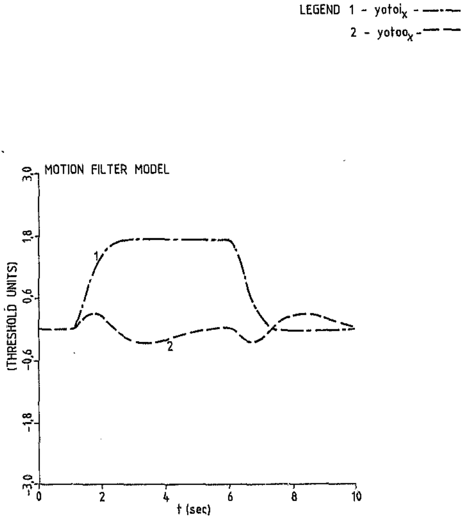

FIGURE E.1.3 : SURGE AXIS VESTIBULAR RESPONSE - NO TILT (SURGE INPUT)

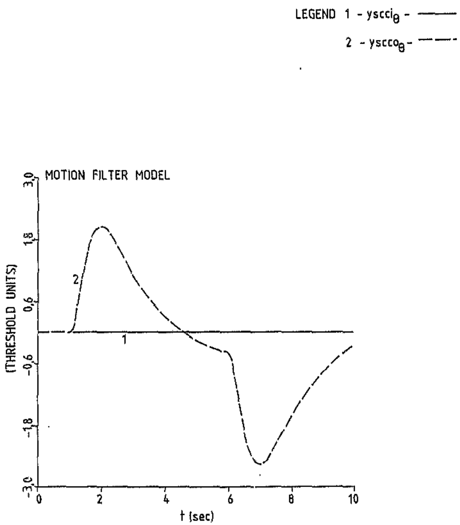

FIGURE E.1.4 : PITCH AXIS VESTIBULAR RESPONSE (SURGE INPUT)

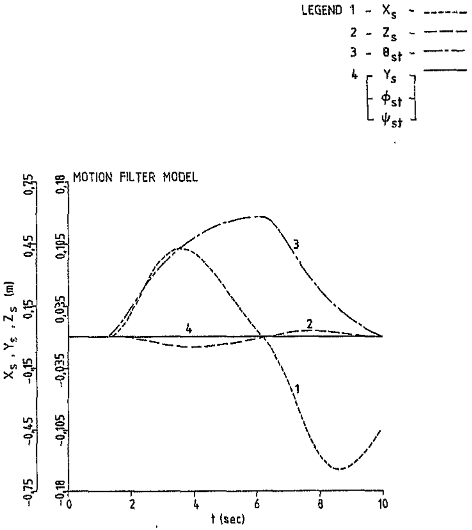

FIGURE E.1.5 : SIMULATOR OPERATOR MOTION (SURGE INPUT)

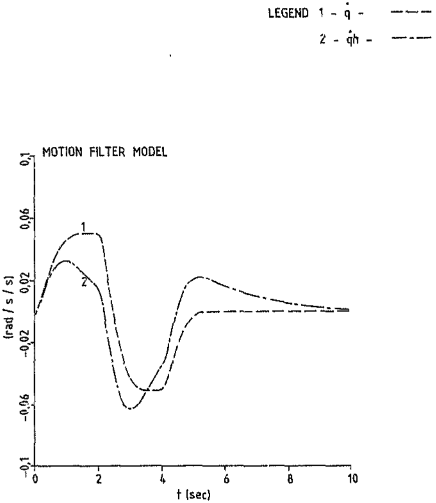

FIGURE E.1.6 : PITCH ACCELERATION INPUT AND WASHOUT

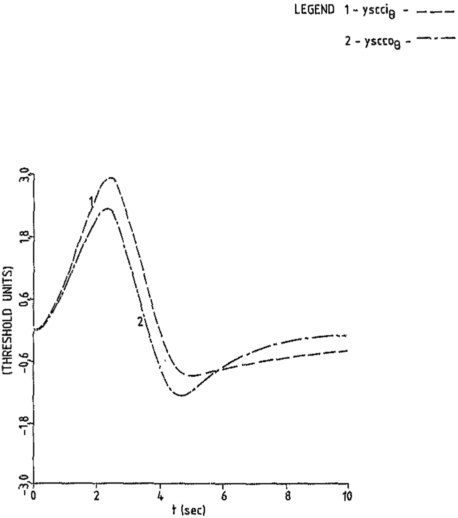

FIGURE E.1.7 : PITCH AXIS VESTIBULAR RESPONSE (PITCH INPUT)

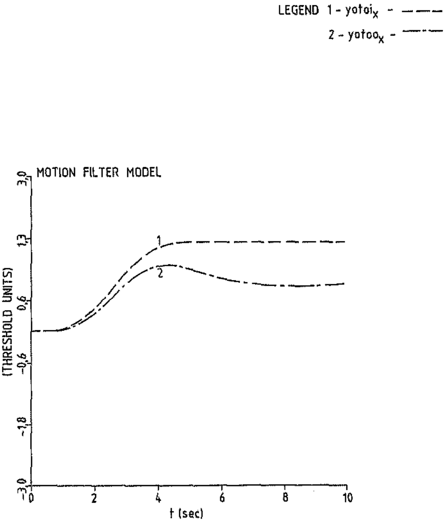

FIGURE E.1.8 : SURGE AXIS VESTIBULAR RESPONSE (PITCH INPUT)

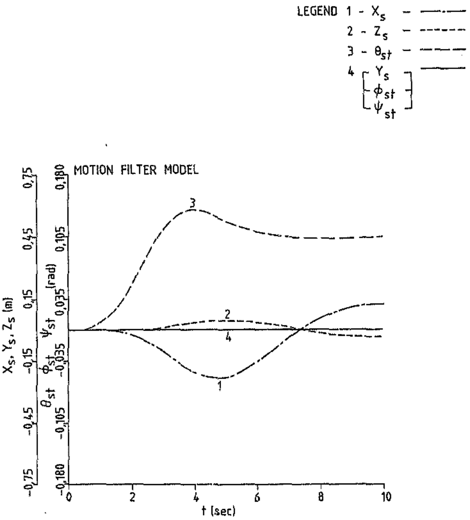

FIGURE E.1.9 : SIMULATOR OPERATOR MOTION (PITCH INPUT)

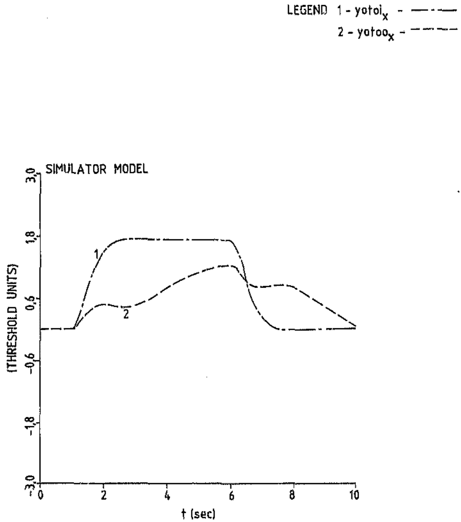

FIGURE E.2.1 : SURGE AXIS VESTIBULAR RESPONSE (AIRCRAFT SURGE INPUT)

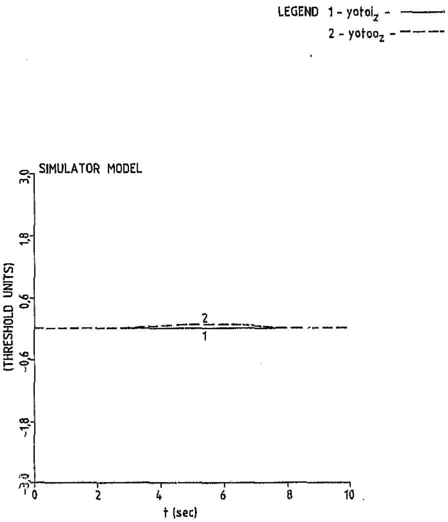

FIGURE E.2.2 : HEAVE AXIS VESTIBULAR RESPONSE (AIRCRAFT SURGE INPUT)

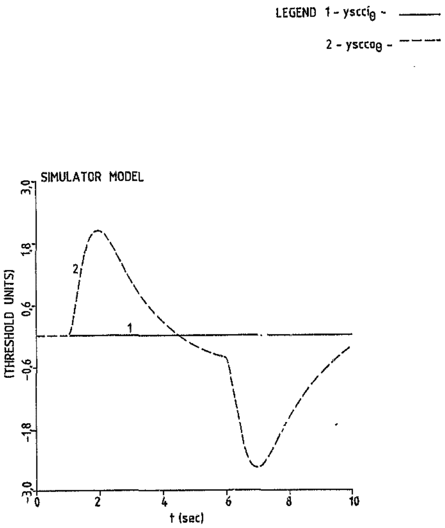

FIGURE E.2.3 : PITCH AXIS VESTIBULAR RESPONSE (AIRCRAFT SURGE INPUT)

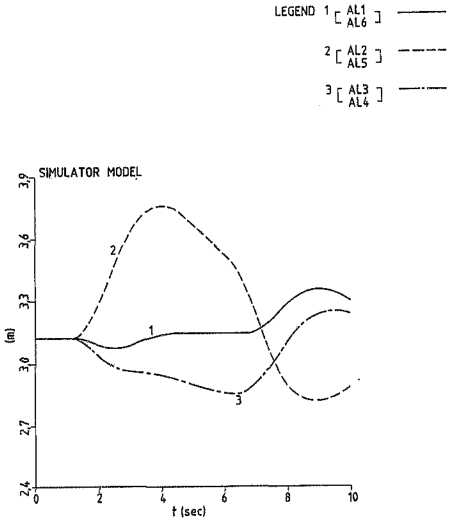

FIGURE E.2.4 : ACTUATOR LENGTHS (AIRCRAFT SURGE INPUT)

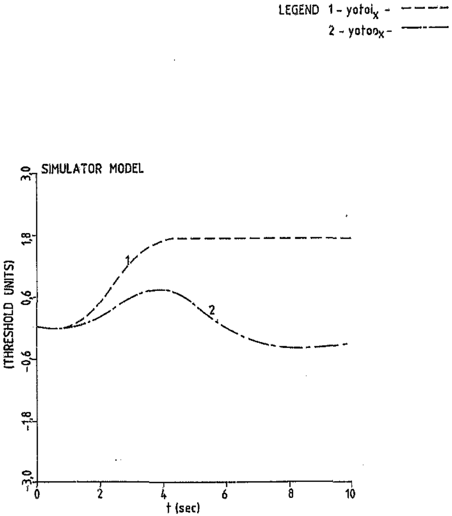

FIGURE E.2.5 : SURGE AXIS VESTIBULAR RESPONSE (AIRCRAFT PITCH INPUT)

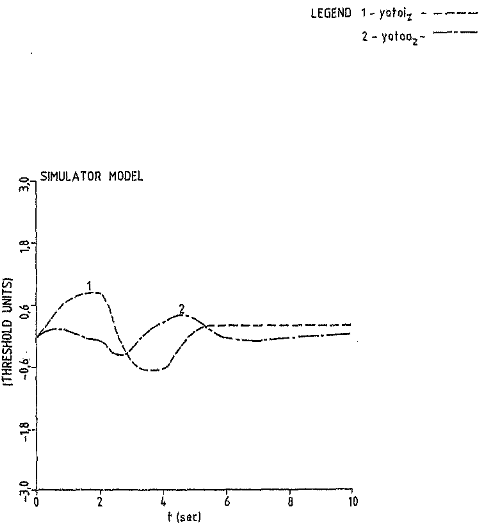

FIGURE E.2.6 : HEAVE AXIS VESTIBULAR RESPONSE (AIRCRAFT PITCH INPUT).

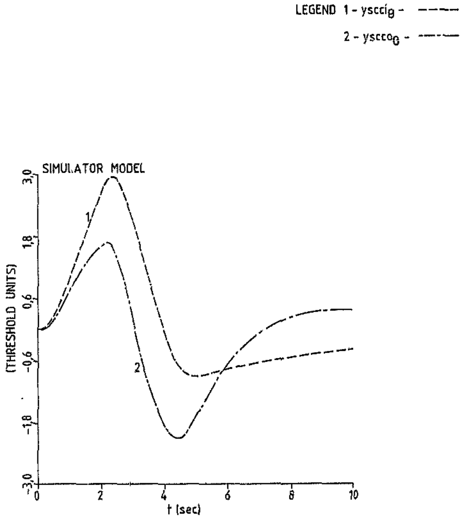

FIGURE E.2.7 : PITCH AXIS VESTIBULAR RESPONSE (AIRCRAFT PITCH INPUT).

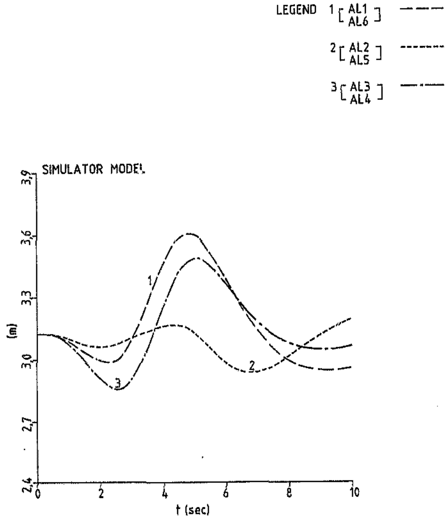

FIGURE E.2.8 : ACTUATOR LENGTHS (AIRCRAFT PITCH INPUT)

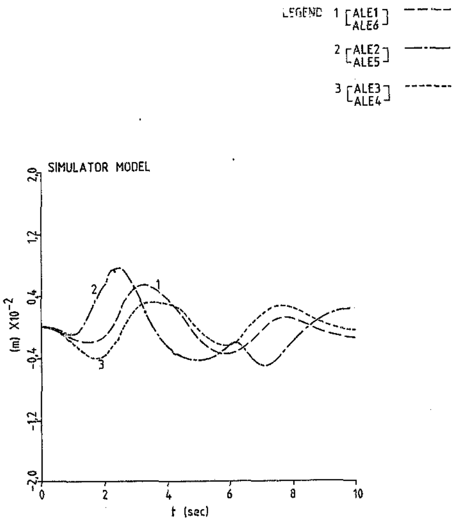

FIGURE E.2.9 : ACTUATOR LENGTH ERROR (COMBINED INPUT)

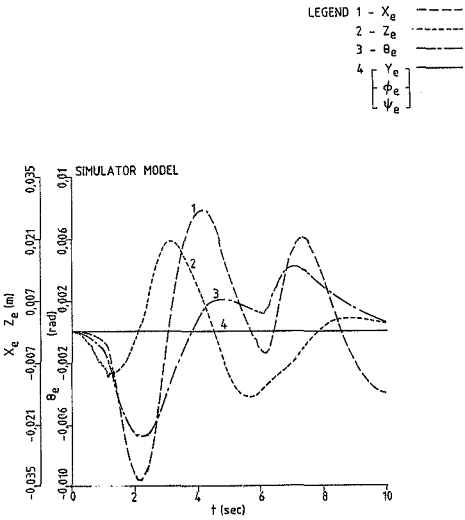

FIGURE E.2.10 : OPERATOR POSITION ERROR (COMBINED INPUT)

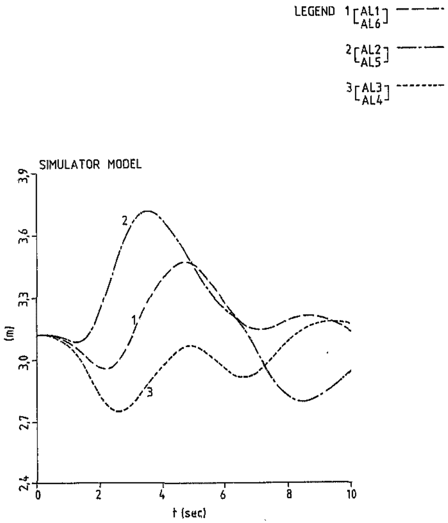

FIGURE E.2.11 : ACTUATOR LENGTHS (COMBINED INPUT)

REFERENCES.

1 Gundry, J. MAN AND MOTION CUES, RAeS Symposium, Theory and Practice in Flight Simulation : April 1976.

2 Stewart, J. D. HUMAN PERCEPTION OF ANGULAR ACCELERATIONS AND IMPLICATIONS IN MOTION SIMULATION, J Aircraft: 1971.

3 Perry, D. H. and Naish, J. M. FLIGHT SIMULATION FOR RESEARCH, The Journal of the Royal Aeronautical Society: October 1964.

4 Schmidt, S. F. and Conrad, B. MOTION DRIVE SIGNALS FOR PILOTED FLIGHT SIMULATORS, NASA CR-1601, Ames Research Centre, California, May 1970.

5 Parrish, R. V. and Martin, D. J. APPLICATION OF NON-LINEAR ADAPTIVE MOTION WASHOUT TO TRANSPORT GROUND-HANDLING SIMULATION, NASA TM 84568, February 1983.

6 Liu, E-Q. A STUDY OF WASHOUT FILTERS FOR A SIMULATOR MOTION BASE, (UTIAS-TN-246) Toronto University Downsvie, Ontario, December 1983.

7 Woomer, C. W. and Williams, R. L. ENVIRONMENTAL REQUIREMENTS FOR SIMULATED HELICOPTER/VTOL OPERATIONS FROM SMALL SHIPS AND CARRIERS, Piloted Aircraft Environment Simulation Techniques: October 1978.

8 Technical Survey : ADVANCED ENGINEERING SIMULATORS, Aviation Week and Space Technology, January 17, 1983.

9 Baret, M. SIX DEGREES OF FREEDOM LARGE MOTION SYSTEM FOR FLIGHT SIMULATORS.

10 Hall, J. R. MOTION VERSUS VISUAL CUES IN PILOTED FLIGHT SIMULATION.

11 Mitchell and Gauthier, Associates, Advanced Continuous Simulation Language (ACSL), User Guide/Reference Manual, Third Edition, 1981.

12 Schwarzenbach, J. and Gill, K. F. SYSTEM MODELLING AND CONTROL, London: Edward Arnold (Publishers) Ltd, 1978.

13 Gerald, C. F. APPLIED NUMERICAL ANALYSIS, California: Addison-Wesley Publishing Company, 1978.

14 MIL-STD-1558 SIX-DEGREE-OF-FREEDOM MOTION SYSTEM REQUIREMENTS FOR AIRCREWMEMBER TRAINING SIMULATORS, Washington: Department of Defence: February 1974.

15 Sivan, R. MODELLING OF THE SPACIAL ORIENTATION PERCEPTION OF HUMANS WITH APPLICATION TO FLIGHT SIMULATORS, Binational Science Foundation Report No 2135/80, August 1982.

16 Dusterberry, J. C. and White, M. D. THE DEVELOPMENT AND USE OF LARGE-MOTION SIMULATOR SYSTEMS IN AERONAUTICAL RESEARCH AND DEVELOPMENT.

17 Sivan, R. and Talwar, N. STOCHASTIC VERSUS DERMINISTIC APPROACH TO THE DESIGN OF THE CONTROL LAW OF MOVING FLIGHT SIMULATORS, Israel Institute of Technology Haifa, Unpublished.

18 Meiry, J. L. THE VESTIBULAR SYSTEM AND HUMAN DYNAMIC SPACE ORIENTATION NASA contract Report Number 628, Cambridge, Massachusettes, 1966.

19 Goldstein, N. H. CLASSICAL MECHANICS, 2nd ed Columbia: Addison-Wesley Publishing Company, 1950.

20 Etkin, B. DYNAMICS OF ATMOSPHERIC FLIGHT, Toronto: John Wiley and Sons Inc., 1972.