NASA TECHNICAL MEMORANDUM

NASA TM X-2872

NASA TM X-2872

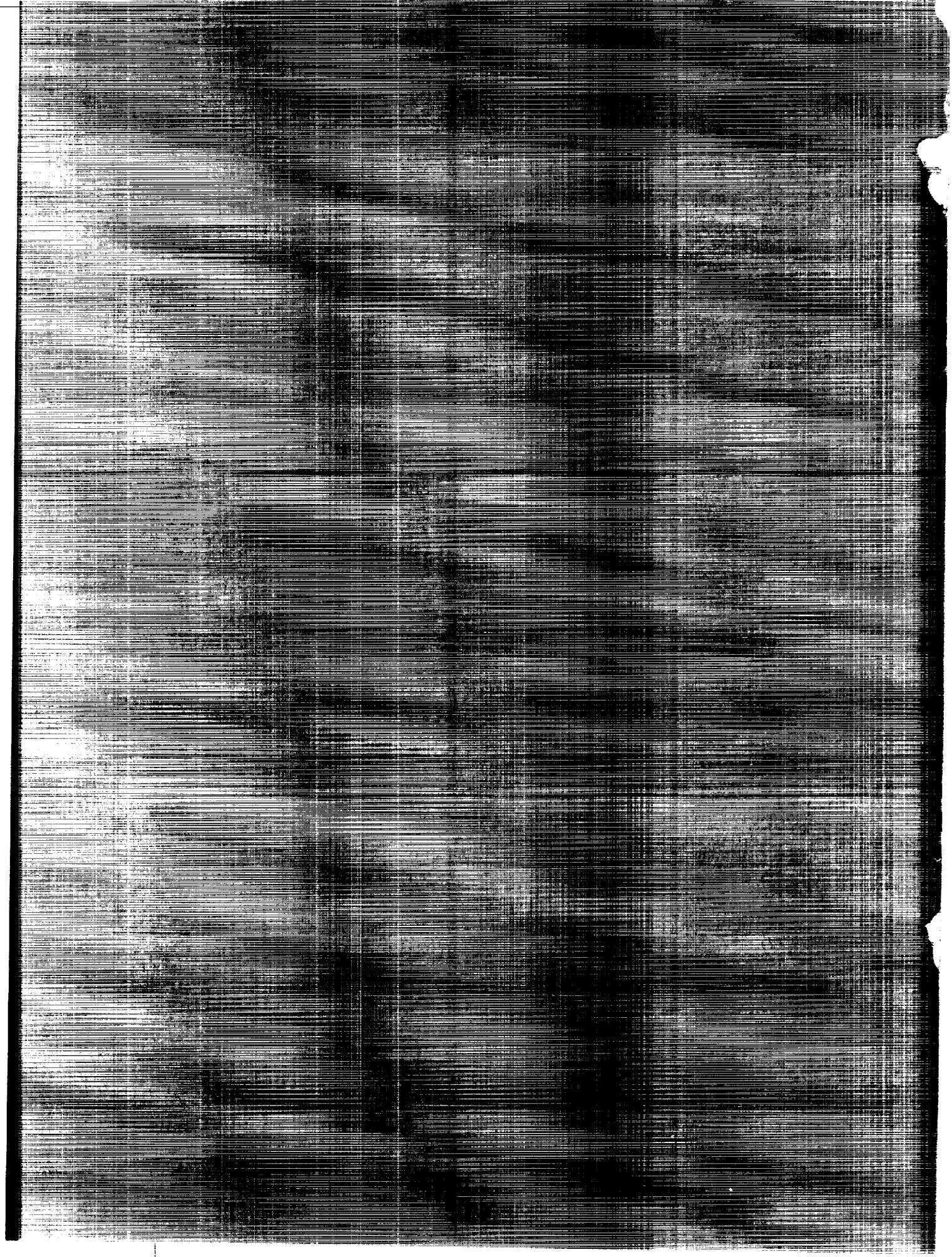
A REAL-TIME DIGITAL COMPUTER PROGRAM FOR THE SIMULATION OF A SINGLE-ROTOR HELICOPTER

by Jacob A. Houck, Lucille H. Gilbert, and George G. Steinmetz

Langley Research Center

Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1974



1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM X-2872		

4. Title and Subtitle	5. Report Date
A REAL-TIME DIGITAL COMPUTER PROGRAM FOR THE SIMULATION OF A SINGLE-ROTOR HELICOPTER	June 1974
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Jacob A. Houck; Lucille H. Gibson, Electronic Associates, Inc.; and George G. Steinmetz	L-9002
	10. Work Unit No. 598-85-40-24

9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, Va. 23665	
	13. Type of Report and Period Covered Technical Memorandum

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Washington, D.C. 20546	

15. Supplementary Notes

Appendix B by Lawrence E. Barker, Jr., and Kemper S. Kibler
Appendix D by Gary A. McDaniel, Electronic Associates, Inc.

16. Abstract

A computer program (Langley program C1152) has been developed for the study of a single-rotor helicopter on the Langley Research Center real-time digital simulation system. This report includes descriptions of the helicopter equations and data, program subroutines (including flow charts and listings), real-time simulation system routines, and program operation. Program usage is illustrated by standard check cases and a representative flight case.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Helicopter Simulation Computer program	Unclassified – Unlimited STAR Category 08

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	163	$5.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

A REAL-TIME DIGITAL COMPUTER PROGRAM FOR THE SIMULATION OF A SINGLE-ROTOR HELICOPTER

By Jacob A. Houck, Lucille H. Gibson,*
and George G. Steinmetz
Langley Research Center

SUMMARY

A computer program (Langley program C1152) has been developed for the study of a single-rotor helicopter on the Langley Research Center real-time digital simulation system. This report includes descriptions of the helicopter equations and data, program subroutines (including flow charts and listings), real-time simulation system routines, and program operation. Program usage is illustrated by standard check cases and a representative flight case.

INTRODUCTION

In view of the expanding interest in helicopters at the Langley Research Center, the development of a man-in-the-loop real-time helicopter simulation was necessary. An existing computer program for a single-rotor helicopter was modified and implemented on the Langley Research Center real-time simulation (RTS) system. This program was then used to gain experience in the area of nelicopter mathematical modeling (specifically in the area of rotor mathematical modeling), and it was used to determine the impact (computer memory and timing requirements) of such a simulation program on the computing facility at the Langley Research Center.

The objective of this paper is to present the development of the computer program for the RTS system. Equations, aerodynamic data, flow charts, computer listings, and operating procedures for the simulation program are given.

The simulation program, with modifications, is presently being used in support of two piloted studies of the Sikorsky S-61 helicopter, which is a five-blade, single-rotor, commercial passenger-type helicopter currently being used by New York Airways. One study concerns pilot workload, route structures, and air traffic control procedures for instrument flight in congested short-haul markets. The second study concerns development and evaluation of various computer-generated displays and instruments for aiding pilots during flight, especially during approach and landing.

---

*Electronic Associates, Inc., Hampton, Va.

SYMBOLS

Measurements, calculations, and programing were made in the U.S. Customary Units. They are presented herein in the International System of Units (SI) with the equivalent values in the U.S. Customary Units given parenthetically.

Most of the symbols used herein are not standard but were adapted from the FORTRAN notation used in the computer program of reference 1.

A_{SPD} indicated airspeed, kilometers/hour (knots)

A_{OS} main rotor collective pitch, radians

A_{OS_C} collective pitch control input, radians

A_{1S} main rotor lateral cyclic pitch, radians

A_{1S_C} lateral cyclic pitch control input, radians

$A_{1S_{CS}}$ SAS lateral cyclic pitch control input, radians

$ADVP_1$ aerodynamic variable parameter, per meter^2 (per foot^2)

$ADVP_2$ aerodynamic variable parameter, per second

a_X, a_Y, a_Z linear acceleration along X, Y, and Z body axis, respectively, meters/second^2 (feet/second^2)

B_{1S} main rotor longitudinal cyclic pitch, radians

B_{1S_C} longitudinal cyclic pitch control input, radians

$B_{1S_{CS}}$ SAS longitudinal cyclic pitch control input, radians

C_j^i SAS constants used in transfer function 1, where $i,j = 1,2,3$

$C_{D_{\Psi Y}}$ main rotor drag coefficient at Ψ_R and Y

$C_{L_{\Psi Y}}$ main rotor lift coefficient at Ψ_R and Y

C_P power coefficient

C_Q torque coefficient

C_T thrust coefficient

C_{X_W} wing drag coefficient

C_{X_1} fuselage drag coefficient as a function of sideslip

C_{X_2} fuselage drag coefficient as a function of angle of attack

C_{Y_F} fuselage side-force coefficient

$C_{Y_{TR}}$ tail rotor side-force coefficient

$C_{Y_{VS}}$ vertical-stabilizer side-force coefficient

C_{Z_F} fuselage vertical-force coefficient

$C_{Z_{HS}}$ horizontal-stabilizer vertical-force coefficient

C_{Z_W} wing vertical-force coefficient

$C_{l_{F_1}}$ fuselage rolling-moment coefficient

$C_{l_{F_2}}$ fuselage roll-rate damping coefficient, kilogram-meters (slug-feet)

$C_{m_{F_1}}$ fuselage pitching-moment coefficient

$C_{m_{F_2}}$ fuselage pitch-rate damping coefficient, kilogram-meters (slug-feet)

$C_{n_{F_1}}$ fuselage yawing-moment coefficient

$C_{n_{F_2}}$ fuselage yaw-rate damping coefficient, kilogram-meters (slug-feet)

$D_{PDR_{\Psi Y}}$ summing variable (main rotor blade drag), meters^4/second^2 (feet^4/second^2)

D_W rotor downwash coefficient

d_X longitudinal distance from center of gravity to main rotor shaft (positive rearward of shaft), meters (feet)

$d_{X_{HS}}$ longitudinal distance from horizontal stabilizer to main rotor shaft (positive rearward of shaft), meters (feet)

$d_{X_{TR}}$ longitudinal distance from tail rotor to main rotor shaft (positive rearward of shaft), meters (feet)

$d_{X_{VS}}$ longitudinal distance from vertical stabilizer to main rotor shaft (positive rearward of shaft), meters (feet)

d_{X_W} longitudinal distance from wing to main rotor shaft (positive rearward of shaft), meters (feet)

d_Z vertical distance from center of gravity to main rotor hub (positive upward to hub), meters (feet)

$d_{Z_{TR}}$ vertical distance from tail rotor to main rotor hub (positive upward to hub), meters (feet)

$d_{Z_{VS}}$ vertical distance from vertical stabilizer to main rotor hub (positive upward to hub), meters (feet)

d_{Z_W} vertical distance from wing to main rotor hub (positive upward to hub), meters (feet)

E Earth coordinate in east direction, meters (feet)

$\dot{E}$ velocity component in east direction, meters/second (feet/second)

e flapping hinge offset, meters (feet)

F_{MRS} main rotor hub moment parameter, kilogram-meters^2 (slug-feet^2)

F_{XR} main rotor force along X body axis, newtons (pounds)

F_{YR} main rotor force along Y body axis, newtons (pounds)

G_{EH} main rotor ground effect due to height

G_{EV} main rotor ground effect due to forward velocity

G_{RWT} vehicle gross weight, kilograms (pounds)

g acceleration due to gravity, meters/second^2 (feet/second^2)

h Earth vertical coordinate, meters (feet)

$\dot{h}$ velocity component in vertical direction, meters/second (feet/second)

$\dot{h}_D$ desired velocity component in vertical direction, meters/second (feet/second)

I_B blade moment of inertia, kilogram-meters^2 (slug-feet^2)

I_{HS} horizontal-stabilizer built-in incidence angle, radians

I_R rotor moment of inertia inboard of flapping hinge, kilogram-meters^2 (slug-feet^2)

I_{ROT} main rotor moment of inertia, kilogram-meters^2 (slug-feet^2)

I_{VS} vertical-stabilizer built-in incidence angle, radians

I_W wing built-in incidence angle, radians

I_{XX}, I_{YY}, I_{ZZ} moment of inertia about X, Y, and Z body axis, respectively, kilogram-meters^2 (slug-feet^2)

K_j^i SAS constants used in transfer function 2, where i = 1,2,3 and j = 1,2,3,4

$K_{A_O}, K_{AB_1}, K_{AB_2}, K_{AB_3}$ main rotor constants for flap angle calculation

K_D, K_{DL}, K_L area constants used to determine main rotor lift and drag forces, meters^2 (feet^2)

K_{E_1} constant used in engine torque calculation, kilogram-meters^2/second (slug-feet^2/second)

K_M, K_Q, K_{QL} volume constants used to determine main rotor moment and torque, meters^3 (feet^3)

$K_{R/F}$ rotor downwash fraction acting on fuselage and wing

K_{TR_1} ratio of elevator area to horizontal-stabilizer area

K_{TR_2} constant used in calculation of Y_{TR}, newtons (pounds)

$K_{W_{IWM}}$ wing downwash coefficient

L_A total aerodynamic rolling moment, newton-meters (pound-feet)

L_F fuselage rolling moment, newton-meters (pound-feet)

L_{MR} main rotor lift force, newtons (pounds)

$L_{PDR_{\Psi Y}}$ summing variable (main rotor blade lift), meters^4/second^2 (feet^4/second^2)

L_{RH} main rotor hub rolling moment, newton-meters (pound-feet)

M_A total aerodynamic pitching moment, newton-meters (pound-feet)

M_F fuselage pitching moment, newton-meters (pound-feet)

$M_{PDR_{\Psi Y}}$ summing variable (main rotor blade flapping moment), meters^5/second^2 (feet^5/second^2)

M_{RH} main rotor hub pitching moment, newton-meters (pound-feet)

M_{TIP} main rotor tip Mach number

m vehicle mass, kilograms (slugs)

m_B blade mass, kilograms (slugs)

N	Earth coordinate in north direction, meters (feet)
$\dot{N}$	velocity component in north direction, meters/second (feet/second)
N_A	total aerodynamic yawing moment, newton-meters (pound-feet)
N_{AZ}	number of rotor azimuth stations
N_B	number of rotor blades
N_F	fuselage yawing moment, newton-meters (pound-feet)
N_{RAD}	number of blade radial stations
p	roll rate, radians/second
$\dot{p}$	roll acceleration, radians/second^2
Q_E	engine torque, newton-meters (pound-feet)
Q_{E_D}	desired engine torque, newton-meters (pound-feet)
Q_{max}	maximum engine torque, newton-meters (pound-feet)
Q_{MR}	main rotor torque, newton-meters (pound-feet)
$Q_{PDR_{\Psi Y}}$	summing variable (main rotor blade torque), meters^5/second^2 (feet^5/second^2)
q	pitch rate, radians/second
$\dot{q}$	pitch acceleration, radians/second^2
$\bar{q}_L$	lateral dynamic pressure, newtons/meter^2 (pounds/foot^2)
$\bar{q}_V$	vertical dynamic pressure, newtons/meter^2 (pounds/foot^2)
R_B	main rotor blade radius, meters (feet)
r	yaw rate, radians/second

$\dot{r}$	yaw acceleration, radians/second^2
S_{HP}	shaft horsepower, watts (horsepower)
$S_{X_{F_1}}$	fuselage lateral-force area, meters^2 (feet^2)
$S_{X_{F_2}}$	fuselage vertical-force area, meters^2 (feet^2)
S_{X_W}	wing longitudinal-force area, meters^2 (feet^2)
S_{Y_F}	fuselage lateral-force area, meters^2 (feet^2)
$S_{Y_{VS}}$	vertical-stabilizer force area, meters^2 (feet^2)
S_{Z_F}	fuselage vertical-force area, meters^2 (feet^2)
$S_{Z_{HS}}$	horizontal-stabilizer force area, meters^2 (feet^2)
S_{Z_W}	wing vertical-force area, meters^2 (feet^2)
t	time, seconds
$U_{P_{\Psi Y}}$	blade element velocity perpendicular to blade span axis and to $U_{T_{\Psi Y}}$ at Ψ_R and Y, meters/second (feet/second)
$U_{T_{\Psi Y}}$	blade element velocity perpendicular to blade span axis and to rotor rotation axis at Ψ_R and Y, meters/second (feet/second)
u	linear velocity along X body axis, meters/second (feet/second)
$\dot{u}$	linear acceleration along X body axis, meters/second^2 (feet/second^2)
V_S	speed of sound, meters/second (feet/second)
V_T	total velocity of air through main rotor, meters/second (feet/second)
V_{TR}	tail velocity along Y body axis, meters/second (feet/second)
V_{l_F}	fuselage rolling-moment volume parameter, meters^3 (feet^3)

V_{m_F} fuselage pitching-moment volume parameter, meters^3 (feet^3)

V_{n_F} fuselage yawing-moment volume parameter, meters^3 (feet^3)

v linear velocity along Y body axis, meters/second (feet/second)

$\dot{v}$ linear acceleration along Y body axis, meters/second^2 (feet/second^2)

W_F main rotor inflow distribution at any radial station

W_{HS} tail velocity along Z body axis, meters/second (feet/second)

W_{IF} main rotor inflow redistribution as a function of velocity, meters/second (feet/second)

W_{IM} main rotor mean inflow velocity, meters/second (feet/second)

W_{IWM} wing downwash velocity, meters/second (feet/second)

$W_{I_{\Psi Y}}$ main rotor local inflow velocity meters/second (feet/second)

w linear velocity along Z body axis, meters/second (feet/second)

$\dot{w}$ linear acceleration along Z body axis, meters/second^2 (feet/second^2)

X_A total aerodynamic force along X body axis, newtons (pounds)

X_{AOS} cockpit collective stick input (positive up), centimeters (inches)

X_{CS} cockpit cyclic stick longitudinal input (positive forward), centimeters (inches)

X_F fuselage aerodynamic force along X body axis, newtons (pounds)

X_{TR} cockpit tail rotor pedal input (positive right), centimeters (inches)

X_W wing aerodynamic force along X body axis, newtons (pounds)

Y main rotor radial distance (measured from hinge), meters (feet)

Y_A total aerodynamic force along Y body axis, newtons (pounds)

Y_{CS} cockpit cyclic stick lateral input (positive right), centimeters (inches)

Y_F fuselage aerodynamic force along Y body axis, newtons (pounds)

Y_{PE} main rotor radial station plus hinge offset, meters (feet)

Y_{TR} tail rotor force along Y body axis, newtons (pounds)

Y_{TW} blade twist at any radial station, radians

Y_{VS} vertical-stabilizer force along Y body axis, newtons (pounds)

Z_A total aerodynamic force along Z body axis, newtons (pounds)

Z_F fuselage aerodynamic force along Z body axis, newtons (pounds)

Z_{HS} horizontal-stabilizer force along Z body axis, newtons (pounds)

Z_W wing aerodynamic force along Z body axis, newtons (pounds)

α_F fuselage angle of attack, radians or degrees

α_{HS} horizontal-stabilizer angle of attack, radians

α_W wing angle of attack, radians

α_{OSS} main rotor coning angle, radians

α_{1SS} main rotor longitudinal flap angle, radians

$\dot{\alpha}_{1SS}$ main rotor longitudinal flap rate, radians/second

$\alpha_{\Psi Y}$ main rotor blade angle of attack at Ψ_R and Y, radians

β_F fuselage angle of sideslip, radians or degrees

β_{VS} vertical-stabilizer angle of sideslip, radians

β_{1SS} main rotor lateral flap angle, radians

$\dot{\beta}_{1SS}$ main rotor lateral flap rate, radians/second

β_{Ψ} main rotor total blade flap angle at Ψ_R, radians

$\dot{\beta}_{\Psi}$ main rotor total blade flap rate at Ψ_R, radians/second

Δt time interval, seconds

δ_E horizontal-stabilizer deflection angle, radians or degrees

θ fuselage pitch angle, radians

$\dot{\theta}$ rate of change of pitch angle, radians/second

θ_{TR} tail rotor pitch angle, radians

θ_{TR_C} tail rotor pitch control input, radians

$\theta_{TR_{CS}}$ SAS tail rotor pitch control input, radians

$\theta_{\Psi Y}$ main rotor blade pitch angle at Ψ_R and Y, radians

μ main rotor tip-speed ratio

ρ air density, kilogram/meter^3 (slug/feet^3)

τ time lag, seconds

Φ fuselage roll angle, radians

$\dot{\Phi}$ rate of change of roll angle, radians/second

$\Phi_{\Psi Y}$ main rotor blade inflow angle at Ψ_R and Y, radians

Ψ fuselage yaw angle, radians

$\dot{\Psi}$ rate of change of yaw angle, radians/second

Ψ_R main rotor blade azimuth angle, radians

Ω main rotor angular velocity, radians/second

$\dot{\Omega}$ main rotor angular acceleration, radians/second^2

Ω_D desired main rotor angular velocity, radians/second

Dot over symbol denotes time derivative.

PROBLEM DESCRIPTION

A description is presented of the development of an all digital simulation program for a single-rotor helicopter to be run on the Langley Research Center real-time simulation (RTS) system. The program was designed to represent the entire operational flight envelope from hover through transition to forward flight.

The equations of motion for a single-rotor helicopter were developed in reference 2. They are suitable for analog, digital, and hybrid simulation programs. The equations include dynamic modeling of the main rotor and airframe to provide a mathematical model which will accept control inputs and calculate the resulting vehicle dynamics and orientation.

In reference 1, the helicopter equations of motion were applied to a specific vehicle. The simulation consisted of an all digital airframe mathematical model programed on the Electronic Associates, Inc. 8400 digital computer and a stability augmentation system (SAS) programed on an analog computer. The digital programing language was FORTRAN IV.

References 1 and 2 have been used to form the basis of the Langley helicopter simulation. A complete collection of aircraft equations is contained in appendix A. Several modifications and updatings of the computer program have been instituted and are discussed herein. The simulation program has also been structured to run on the Langley RTS system which is described in reference 3. Operating procedures are discussed in appendix A of reference 4, and in the Program Organization and Program Usage sections of this paper.

One of the major modifications to the computer program involved a change in integration formulas. Reference 1 uses an Euler formula to integrate the state and rotor variables with a solution rate of 25 iterations per second. The Langley program uses a second-order Adams-Bashforth predictor integration formula with a solution rate of 32 iterations per second to solve the state variables. This allows for a more accurate

solution and is one of several integration formulas available through the RTS system software. This integration of the state variables occurs only in the operate mode of the simulation. There are five sets of rotor variables which are integrated continually, regardless of the simulation computer mode, in order to stabilize the rotor dynamics. They are the inflow velocity, the main rotor coning angle, the two components of the main rotor flapping angle, and the main rotor angular velocity. These variables are integrated by using an Euler formula at a solution rate of 32 iterations per second.

Other modifications involve function generation and trim condition calculation. The function generation system used in reference 1 has been replaced by a technique used for several years with the Langley RTS system. This technique allows for rapid generation of aerodynamic functions used in real-time simulations. The technique is detailed in reference 5 and in appendix B by Lawrence E. Barker, Jr., and Kemper S. Kibler. The function data are presented in appendix C. The trim program of reference 1 has been replaced because of its slow convergence rate and its poor accuracy. The new trim circuit, a modified secant method, is detailed in reference 6 and has been reproduced for convenience in appendix D by Gary A. McDaniel. It has been used in several RTS system programs and is rapidly becoming one of the standard techniques used with this system. The technique is extremely fast and requires only a few seconds to reach trim criteria at speeds up to 140 knots in level flight. A forward velocity of 140 knots appears to be the upper limit for trim convergence in level flight in this program due to the uncertainty in the helicopter aerodynamic data. Trim conditions at higher velocities have been reached in descents; and with an improved data package, it is felt that trim conditions at higher forward velocities can be reached in level flight.

A stability augmentation system (SAS) has been implemented in the Langley simulation. This system is not discussed except to mention that it was necessary to develop special integration techniques due to the numerical instability of the standard RTS integration formulas when handling time constants of the magnitude involved. These convolution integration techniques are derived from reference 7.

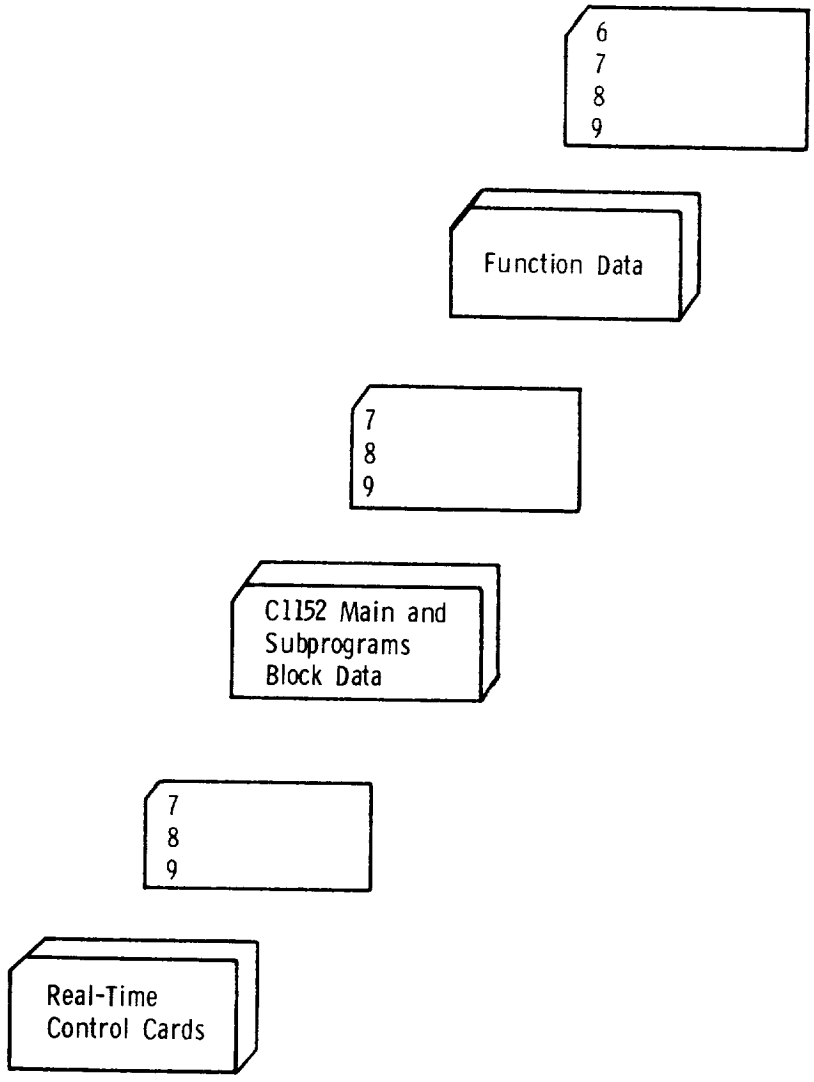
One final feature has been added to the simulation program. This feature is the ability to conduct preprogramed flight maneuvers from the program control station (fig. 1) and allows the analyst to run research and validation tests on the mathematical model without having a pilot present to fly the simulation. The operating procedure for this feature is detailed in the Program Usage section of this paper.

With minor changes to the mathematical model (data, control systems equations, fuselage equations, etc.), the program described in this paper can be used to simulate a variety of single-rotor helicopters. Future studies include air traffic control, control systems, navigation systems, and pilot display systems.

PROGRAM ORGANIZATION

Langley program C1152 was written in the FORTRAN IV language to be run in a real-time mode on the CDC 6600 computer system. The program requires a field length of 65 000 octal locations. The computing time depends primarily on the integration scheme used. For the normally used one-pass Adams-Bashforth integration scheme, the computing time required is approximately 11.25 milliseconds (total available is 31.25 milliseconds) per iteration.

This section contains a list of the FORTRAN variables in labeled COMMON and the various DSPLAY arrays, a brief description of the real-time system subroutines, and individual program subroutine descriptions with corresponding flow charts and listings. The following diagram shows the deck configuration needed for execution:



Deck Configuration

Labeled COMMON

The following list contains the FORTRAN variables appearing in labeled COMMON and descriptions of each variable:

COMMON label	FORTRAN variable	Description
/CHECK/	TABLE(150)	Array of floating point numbers to be displayed and/or changed
	INTEG(09)	Array of integer numbers to be displayed and/or changed
/INTCOMM/	T	Time, update in subroutine IGRATE1
	DT	Integration step size used in subroutine IGRATE1
	INT	Flow control parameter used in subroutine IGRATE1
	NEQ	Number of integrations performed in subroutine IGRATE1
	ISCHEME	Selects integration scheme in subroutine IGRATE1
	DERINT(2,12)	Array of integrals and derivatives in subroutine IGRATE1
/INTINTR/	INTERN(5,12)	Temporary storage for elements of subroutine IGRATE1
/REALTIM/	ADC(32)	Array of analog-to-digital converter inputs
	DAC(64)	Array of digital-to-analog converter outputs
	LDISI(108)	Array of logical discrete inputs
	LDISO(196)	Array of logical discrete outputs

COMMON label	FORTRAN variable	Description
/REALTIM/	NOPER	Return addresses from subroutine RTMODE
	NHOLD	
	NRESET	
	NTERM	
	NPRINT	
	NREAD	
/ACCEL/	AX	a_X
	AY	a_Y
	AZ	a_Z
/ADV C/	OMEG	Ω
	WIM	W_{IM}
	VT	V_T
	QV	$\bar{q}_V$
	QL	$\bar{q}_L$
	ASPD	A_{SPD}
/ADV P/	ADVP1	$ADVP_1$
	ADVP2	$ADVP_2$

COMMON label	FORTRAN variable	Description
/ANGSAS/	PHIL	Last values of Euler angles and their derivatives, used in SAS calculation
	THETAL	
	PSIL	
	PHIDL	
	THETDL	
	PSIDL	
/COEF/	CT	C_T
	CP	C_P
	CQ	C_Q
	MU	μ
	MTIP	M_{TIP}
	VS	V_S
/CONTRL/	AOS	A_{OS}
	A1S	A_{1S}
	B1S	B_{1S}
	THTR	θ_{TR}
	OMEGD	Ω_D

COMMON label	FORTRAN variable	Description
/CNTL C/	AOSC	A_{OS_C}
	A1SC	A_{1S_C}
	B1SC	B_{1S_C}
	THTRC	θ_{TR_C}
	A1SCL	Storage locations for last values of A_{1S_C}, B_{1S_C}, and θ_{TR_C}
	B1SCL	
	THTRCL	
/CNTL P/	AOSC0	Control constants in A_{OS_C}, A_{1S_C}, B_{1S_C}, and θ_{TR_C}
	A1SC0	
	B1SC0	
	THTRC0	
	XAOSG	Control gains
	YCSG	
	XCSG	
	XTRG	
	XAOSR	
	YCSR	
	XCSR	
	XTRR	

COMMON label	FORTRAN variable	Description
/CNTSAS/	A1SCS	$A_{1S_{CS}}$
	B1SCS	$B_{1S_{CS}}$
	THTRCS	$\theta_{TR_{CS}}$
/CPIC/	ACI	Instrument lag parameters
	BCI	
	DCI	
	XRCM1	Instrument constants
	YRCM1	
	XRTM1	
	YRTM1	
	XPBM1	
	YPBM1	
	CSK4	
	CSK6	
	CSK7	
	F05(18)	
/ENG C/	QMAX	Q_{max}
	QE	Q_E
	SHP	S_{HP}

COMMON label	FORTRAN variable	Description
/ENG C/	GRWT	G_{RWT}
	OMEGDT	$\dot{\Omega}$
/ENG P/	IROT	I_{ROT}
	CKE1	K_{E_1}
/ERROR/	NERR	Error return parameter
/EUL CS/	COSFI	$\cos(\Phi)$
	SINFI	$\sin(\Phi)$
	COSTH	$\cos(\theta)$
	SINTH	$\sin(\theta)$
	COSSI	$\cos(\Psi)$
	SINSI	$\sin(\Psi)$
/FCNNAME/	CLF1	$C_{l_{F_1}} = f(\beta_F)$
	CMF1	$C_{m_{F_1}} = f(\alpha_F)$
	CNF1	$C_{n_{F_1}} = f(\beta_F)$
	CX1	$C_{X_1} = f(\beta_F)$
	CX2	$C_{X_2} = f(\alpha_F)$
	CYTR	$C_{Y_{TR}} = f(u, V_{TR}, \theta_{TR})$
	CYF	$C_{Y_F} = f(\beta_F)$
	CZF	$C_{Z_F} = f(\alpha_F)$

COMMON label	FORTRAN variable	Description
/FCNNAME/	CXW	$C_{X_W} = f(\alpha_W)$
	CZW	$C_{Z_W} = f(\alpha_W)$
	CYVS	$C_{Y_{VS}} = f(\beta_{VS})$
	CZHS	$C_{Z_{HS}} = f(\alpha_{HS})$
	CDSIY	$C_{D_{\Psi Y}} = f(\alpha_{\Psi Y}, U_{T_{\Psi Y}})$
	CLSIY	$C_{L_{\Psi Y}} = f(\alpha_{\Psi Y}, U_{T_{\Psi Y}})$
	DW	$D_W = f(u)$
	GEV	$G_{EV} = f(u)$
	GEH	$G_{EH} = f(h)$
/F N M C/	XA	X_A
	YA	Y_A
	ZA	Z_A
	LA	L_A
	MA	M_A
	NA	N_A
/F N M P/	DX	d_X
	DXHS	$d_{X_{HS}}$
	DXTR	$d_{X_{TR}}$
	DXVS	$d_{X_{VS}}$

COMMON label	FORTRAN variable	Description
/F N M P/	DZ	d_Z
	DZTR	$d_{Z_{TR}}$
	DZVS	$d_{Z_{VS}}$
	DXW	d_{X_W}
	DZW	d_{Z_W}
	IR	I_R
/FTRMPLS/	EQLMR	Trim parameters
	EQAOSS	
	EQWIM	
	HDOTD	$\dot{h}_D$
/FUNCS/	F001(28)	Array of data values for $C_{m_{F_1}}$
	F002(28)	Array of data values for C_{X_2}
	F003(14)	Array of data values for C_{Z_F}
	F004(14)	Array of data values for $C_{l_{F_1}}$
	F005(14)	Array of data values for $C_{n_{F_1}}$
	F006(28)	Array of data values for C_{X_1}
	F007(14)	Array of data values for C_{Y_F}
	F008(28)	Array of data values for C_{X_W}
	F009(14)	Array of data values for C_{Z_W}

COMMON label	FORTRAN variable	Description
/FUNCS/	F010(28)	Array of data values for $C_{Y_{VS}}$
	F011(35)	Array of data values for $C_{Z_{HS}}$
	F013(07)	Array of data values for D_W
	F014(07)	Array of data values for G_{EV}
	F015(07)	Array of data values for G_{EH}
	F016(35,10)	Array of data values for $C_{D_{\Psi Y}}$
	F017(35,10)	Array of data values for $C_{L_{\Psi Y}}$
	F018(7,3,4)	Array of data values for $C_{Y_{TR}}$
	D001(9)-D011(9)	Arrays containing the generated function values and function parameters for F001-F011
	D013(9)-D015(9)	Arrays containing the generated function values and function parameters for F013-F015
	D016(18)-D017(18)	Arrays containing the generated function values and function parameters for F016-F017
	D018(27)	Array containing the generated function value and function parameters for F018
/FUS C/	XF	X_F
	YF	Y_F
	ZF	Z_F
	LF	L_F
	MF	M_F

COMMON label	FORTRAN variable	Description
/FUS C/	NF	N_F
	XW	X_W
	WIWM	W_{IWM}
	ZW	Z_W
	ALFF	α_F
	BETF	β_F
	ALFW	α_W
/FUS P/	CLF2	$C_{l_{F_2}}$
	CMF2	$C_{m_{F_2}}$
	CNF2	$C_{n_{F_2}}$
	CKRF	$K_{R/F}$
	CKWIWM	$K_{W_{IWM}}$
	IW	I_W
	VLF	V_{l_F}
	VMF	V_{m_F}
	VNF	V_{n_F}
	SXF1	$S_{X_{F_1}}$
	SXF2	$S_{X_{F_2}}$
	SYF	S_{Y_F}

COMMON label	FORTRAN variable	Description
/FUS P/	SZF	S_{Z_F}
	SXW	S_{X_W}
	SZW	S_{Z_W}
/GPARAM/	DELT	Δt
	RHO	ρ
	G	g
	PIE	π
/LAGSTK/	NILAG	Time lag input parameter
	NSTICK	Console potentiometer input parameter
/MODEC/	MRESET	Mode control reset
	MHOLD	Mode control hold
	MOPER	Mode control operate
/MROT C/	FXR	F_{XR}
	FYR	F_{YR}
	LMR	L_{MR}
	LRH	L_{RH}
	MRH	M_{RH}
	QMR	Q_{MR}

COMMON label	FORTRAN variable	Description
/MROT C/	AOSS	α_{OSS}
	ALFSIY	$\alpha_{\Psi Y}$
	UTSIY	$U_{T_{\Psi Y}}$
	BETSI	β_{Ψ}
	BDTSI	$\dot{\beta}_{\Psi}$
	A1SS	α_{1SS}
	AD1SS	$\dot{\alpha}_{1SS}$
	B1SS	β_{1SS}
	BD1SS	$\dot{\beta}_{1SS}$
/RCOEFF/	CKL(20)	K_L
	CKD(20)	K_D
	CKQ(20)	K_Q
	CKDL(20)	K_{DL}
	CKQL(20)	K_{QL}
	CKM(20)	K_M
	Y(20)	Y
	WF(20)	W_F
/ROTCON/	YPE(20)	Y_{PE}
	YTW(20)	Y_{TW}

COMMON label	FORTRAN variable	Description
/ROTCON/	WIF(20)	W_{IF}
	BOVN	Ratio of N_B to N_{AZ}
/ROT P/	TWIST	Blade twist
	FMRS	F_{MRS}
	CKAO	K_{A_O}
	CKAB1	K_{AB_1}
	CKAB2	K_{AB_2}
	CKAB3	K_{AB_3}
	NRAD	N_{RAD}
	NAZ	N_{AZ}
	NB	N_B
	RB	R_B
	EFH	e
/SAS P/	A1C1	C_1^1
	A1C2	C_2^1
	A1K1	K_1^1
	A1K2	K_2^1
	A1K3	K_3^1
	A1K4	K_4^1

COMMON label	FORTRAN variable	Description
/SAS P/	B1C1	C_1^2
	B1C2	C_2^2
	B1K1	K_1^2
	B1K2	K_2^2
	B1K3	K_3^2
	B1K4	K_4^2
	THC1	C_1^3
	THC2	C_2^3
	THK1	K_1^3
	THK2	K_2^3
	THK3	K_3^3
	THK4	K_4^3
/SCALES/	SF1	Cockpit scale factors
	SF2	
	SF3	
	SF4	
	SF5	
	SF6	
	SF7	

COMMON label	FORTRAN variable	Description
/SCPSIR/	SINSIR(50)	$\sin(\Psi_R)$
	COSSIR(50)	$\cos(\Psi_R)$
/SHIP P/	IXX	I_{XX}
	IYY	I_{YY}
	IZZ	I_{ZZ}
	MASS	m
/STICKS/	XAOS	X_{AOS}
	YCS	Y_{CS}
	XCS	X_{CS}
	XTR	X_{TR}
	DPR	Degrees per radian conversion factor
	RPD	Radians per degree conversion factor
/STINPUT/	X1	Initial input on X_{AOS}
	X2	Initial input on Y_{CS}
	X3	Initial input on X_{CS}
	X4	Initial input on X_{TR}
	T1	Time at which to apply stick input
	T2	Time at which to remove stick input

COMMON label	FORTRAN variable	Description
/TAIL C/	ALFHS	α_{HS}
	BTVS	β_{VS}
	ZHS	Z_{HS}
	YVS	Y_{VS}
	VTR	V_{TR}
	WHS	W_{HS}
	YTR	Y_{TR}
	DELE	δ_E
/TAIL P/	CKTR1	K_{TR_1}
	CKTR2	K_{TR_2}
	IHS	I_{HS}
	IVS	I_{VS}
	SZHS	$S_{Z_{HS}}$
	SYVS	$S_{Y_{VS}}$
/TRIMAT/	XMAT(11,12)	An (NEQN × NEQP1) matrix of independent trim variables
	EMAT(12,13)	An (NEQP1 × NEQP2) matrix of functional evaluations
	TRSUM(12)	An NEQP1 dimensional vector of error sums
	NEQN	Number of equations to be trimmed

COMMON label	FORTRAN variable	Description
/TRIMAT/	NEQP1	NEQN + 1
	NEQP2	NEQN + 2
	XMATG(11)	Increments added to XTRIM to start the iteration
/VALUES/	A1CS1	SAS value of first transfer function on A_{1S}
	A1CS2	SAS value of second transfer function on A_{1S}
	B1CS1	SAS value of first transfer function on B_{1S}
	B1CS2	SAS value of second transfer function on B_{1S}
	THCS1	SAS value of first transfer function on θ_{TR}
	THCS2	SAS value of second transfer function on θ_{TR}
/XIC/	A1X1	
	A1X2	
	A1X3	
	B1X1	
	B1X2	Second transfer function parameters
	B1X3	
	THX1	
	THX2	
	THX3	

COMMON label	FORTRAN variable	Description
/XIC/	AX1TN	First transfer function parameters
	AX2TN	
	BX1TN	
	BX2TN	
	TX1TN	
	TX2TN	

DSPLAY Arrays

The real-time simulation program uses specified arrays for displaying and/or changing the values of desired program variables. The desired program variables as defined in these specified arrays are assigned display addresses as shown in the following table:

DSPLAY arrays	Display address	Maximum number of elements
TABLE(I)	I	199
INTEG(I)	I + 200	99
LOGIC(I) (not used)	I + 300	99
ADC(I)	I + 400	99
DAC(I)	I + 500	99
LDISI(I)	I + 600	99
LDISO(I)	I + 700	199

This type of addressing is called "IN TABLES" addressing. For program variables not in the DSPLAY arrays, a type of addressing called "NO TABLES" addressing is used. The DSPLAY arrays are listed below with their associated FORTRAN variables and descriptions. The array elements are equivalenced to the FORTRAN variables, except where equality signs are indicated.

TABLE is a floating point number array.

TABLE

TABLE element	FORTRAN variable	Description
1	= A1SS	α_{1SS}
2	= AD1SS	$\dot{\alpha}_{1SS}$
3	= B1SS	β_{1SS}
4	= BD1SS	$\dot{\beta}_{1SS}$
5	= BETSI	β_{Ψ}
6	= BDTSI	$\dot{\beta}_{\Psi}$
7	= E	E
8	= EDOT	$\dot{E}$
9	= H	h
10	= HDOT	$\dot{h}$
11	= N	N
12	= NDOT	$\dot{N}$
13	= OMEG	Ω
14	= OMEGDT	$\dot{\Omega}$
15	= P	p
16	= PDOT	$\dot{p}$

TABLE element	FORTRAN variable	Description
17	= PHI	Φ
18	= PHID	$\dot{\Phi}$
19	= PSI	Ψ
20	= PSIDT	$\dot{\Psi}$
21	= Q	q
22	= QDOT	$\dot{q}$
23	= R	r
24	= RDOT	$\dot{r}$
25	= THETA	θ
26	= THETDT	$\dot{\theta}$
27	= U	u
28	= UDOT	$\dot{u}$
29	= V	v
30	= VDOT	$\dot{v}$
31	= W	w
32	= WDOT	$\dot{w}$
33	= AOS	A_{OS}
34	= A1S	A_{1S}
35	= B1S	B_{1S}

TABLE element	FORTRAN variable	Description
36	= THTR	θ_{TR}
37	= OMEGD	Ω_D
38	= T	t
40	AOS0	Initial conditions
41	A1S0	
42	B1S0	
43	THTR0	
44	U0	
45	V0	
46	W0	
47	PSI0	
48	PHI0	
49	THETA0	
50	UKNOTS	
51	HDOTD0	
52	H0	
53	OMEG0	
54	CG0	
55	GRWT0	

TABLE element	FORTRAN variable	Description
56	P0	Initial conditions
57	Q0	
58	R0	
60	A1SSTEP	Control step inputs
61	B1SSTEP	
62	THTRSTP	
63	A1SGN	Scale factors for variables on time history recorders
64	B1SGN	
65	THTRGN	
66	AOSGN	
67	PGN	
68	QGN	
69	RGN	
70	UGN	
71	VGN	
72	WGN	
73	HGN	
74	PSIGN	

TABLE element	FORTRAN variable	Description
75	PHIGN	Scale factors for variables on time history recorders
76	THEGN	
77	A1SSGN	
78	B1SSGN	
79	ALFFGN	
80	BETFGN	
81	BETSGN	
82	BDTGN	
83	DTCON	Divisor on step size DT
84	X10	Control stick inputs, initial conditions
85	X20	
86	X30	
87	X40	
88	T10	Time at which to apply input, initial condition
89	T20	Time at which to remove input, initial condition
90	SF10	Cockpit scale factors, initial conditions
91	SF20	
92	SF30	

TABLE element	FORTRAN variable	Description
93	SF40	Cockpit scale factors, initial conditions
94	SF50	
95	SF60	
96	SF70	
125	= CT	C_T
126	= CP	C_P
127	= CQ	C_Q
128	= MU	μ
129	= MTIP	M_{TIP}
130	= ASPD	A_{SPD}
131	= SHP	S_{HP}
132	= XAOS	X_{AOS}
133	= YCS	Y_{CS}
134	= XCS	X_{CS}
135	= XTR	X_{TR}

INTEG is an integer number array.

INTEG

INTEG element	FORTRAN variable	Description
1	NSTICK0	Denotes cockpit or potentiometer inputs
2	NT	Determines print interval
3	IRUN	Run number
4	NILAG0	Denotes lag input
5	INTSCME	Selects integration scheme
6	KNOTS	Selects airspeed

ADC is an input array from potentiometers or cockpit inputs.

ADC

ADC element	FORTRAN variable	Description
1	XAOS	X_{AOS}
2	YCS	Y_{CS}
3	XCS	X_{CS}
4	XTR	X_{TR}
5	YCSCOL	Lateral stick trim adjustment input
6	XCSCOL	Longitudinal stick trim adjustment input
7	XTRCOL	Tail rotor pedal trim adjustment input

DAC is an output array. Elements 1 through 24 are for the time history recorders. Elements 25 through 40 are outputs to the cockpit.

DAC

DAC element	FORTRAN variable	Description
1	XAOS*XAOSGN	Normalized time history recordings
2	AOS*AOSGN	
3	XCS*XCSGN	
4	B1S*B1SGN	
5	YCS*YCSGN	
6	A1S*A1SGN	
7	XTR*XTRGN	
8	THTR*THTRGN	
9	P*PGN	
10	Q*QGN	
11	R*RGN	
12	PHI*PHIGN*DPR	
13	THETA*THEGN*DPR	
14	PSI*PSIGN*DPR	
15	ALFF*ALFFGN*DPR	
16	BETF*BETFGN*DPR	

DAC element	FORTRAN variable	Description
17	H*HGN	Normalized time history recordings
18	U*UGN	
19	V*VGN	
20	W*WGN	
21	A1SS*A1SSGN	
22	B1SS*B1SSGN	
23	BETSI*BETSGN	
24	BDTSI*BDTGN	
25	-COS(PHI)	Euler angle drivers
26	-SIN(PHI)	
27	-COS(THETA)	
28	-SIN(THETA)	
29	-COS(PSI)	
30	-SIN(PSI)	
31	COS(AA4)	Altimeter drivers
32	SIN(AA4)	
33	-COS(AA5)	Rate of climb meter drivers
34	SIN(AA5)	

DAC element	FORTRAN variable	Description
35	LIM(COS(AA6),-.886,.886)	Limited two minute turn drivers
36	LIM(SIN(AA6),-.500,.500)	
37	-COS(AA7)	Bank angle indicator drivers
38	-SIN(AA7)	
39	LIM(OMEGRPM/500.,0.0,.9996)	Limited RPM indicator driver
40	LIM(ASPD/140.,0.0,0.9996)	Limited airspeed indicator driver

LDISI is a logical discrete input array where the descriptions are for true (.T.) values of the switches.

LDISI

LDISI element	Description
1-16	Data entry keyboard
17	OPER (OPERATE) mode switch, integration of equations of motion
18	HOLD mode switch, stops integration of equations of motion and holds all variables at present values
19	RESET mode switch, initialization of variables (conditions for t = 0)
20	TERM (TERMINATE) mode switch, terminates control at the program control console and transfers control to the graphic display unit
21	CHANGE mode switch, allows changing of variable values
22	SCAN mode switch, scans through the display addresses in conjunction with the subroutine SCANNER

LDISI element	Description
23	RELEASE mode switch, releases CHANGE or SCAN mode switch
27	ERASE mode switch, erases real-time disk file
29	IDLE mode switch, uses minimum computing time
30	PRINT mode switch, prints output of previously stored variables
31	READ mode switch, selects printout storage format
32	RELEASE mode switch, releases ERASE, IDLE, PRINT, or READ mode switch
33	Dynamic check doublet inputs on A_{1S}, B_{1S}, or θ_{TR}
37	Trim circuit switch
38	Selects second set of printout variables
41	Stability augmentation system (SAS) switch
42	Cockpit instrument test
43	Cockpit mode control
44	Cockpit tie-in
45	ADC inputs
46	AUTOTYPE, a logical variable used with the typewriter
47	TYPESW, a logical variable used with the typewriter
48	"IN TABLES" addressing
55	Cockpit SAS switch

LDISO is a logical discrete output array. Elements 61 through 99 are used to turn the white indicator lights on and off. Elements 101 through 110 are recorder event markers. Elements 111 through 118 are used to turn the red indicator lights on and off. Only the following elements are used.

LDISO

LDISO element	Description
61	Variables are trimmed
65	SAS is activated
73	α_F out of range
74	β_F out of range
75	α_W out of range
76	α_{HS} out of range
77	β_{VS} out of range
78	$U_{T_{\Psi Y}}$ out of range
79	$\alpha_{\Psi Y}$ out of range
80	$C_{Y_{TR}}$ out of range
81	X_{AOS} is trimmed
82	Y_{CS} is trimmed
83	X_{CS} is trimmed
84	X_{TR} is trimmed
102	Timing marker on recorder no. 1

LDISO element	Description
110	Timing marker on recorder no. 2
111	Trim unsuccessful
112	A_{OS} out of range
113	A_{1S} out of range
114	B_{1S} out of range
115	θ_{TR} out of range
116	Automatic hold
117	Negative altitude
118	Engine power out of range

Real-Time System Subroutine Descriptions

The real-time system subroutines and brief descriptions of their functions are as follows:

Subroutine	Description
APRINT	Provides an alternate entry point to subroutine RTMODE which returns control to subroutine RTMODE for further processing
AREAD	Provides an alternate entry point to subroutine RTMODE which returns control to subroutine RTMODE for further processing
ATERM	Does final processing and halts execution
CYCLE	Sets up return address from subroutine RECYCLE
DATABLX	Specifies the variable arrays for subroutine DSPLAY

Subroutine	Description
DAYTIM	Provides date and time of day
HALT	Signals completion of real-time portion of program
IGRATE1	Integrates variables in array DERINT(2,J) and stores in array DERINT(1,J) (J = 1,2,...,NEQ)
INOUT	Sets up arrays for input/output conversion
LOSTIME	Sets address to which control will be returned in the case of iteration time exceeded
NAMECRT	Identifies and initializes usage of the CRT console
NM218	Initializes usage of typewriter
PLAYBAK	Plays back data recorded on real-time disk one frame at a time
PRINTER	Routes MFI disk file to printer
READOUT	Specifies variables to be recorded and frequency of recording for the real-time disk file
READY	Signals that the program is ready for real-time operation
RECORD	Records variables as specified in subroutine READOUT
RECYCLE	Signals end of a cycle in the operating loop and returns to address specified by subroutine CYCLE
RELEASE	Releases a real-time data file that has been opened by subroutine READOUT
RTMODE	Real-time mode control
RTROUTE	Indicates dynamic printing will occur during the job run

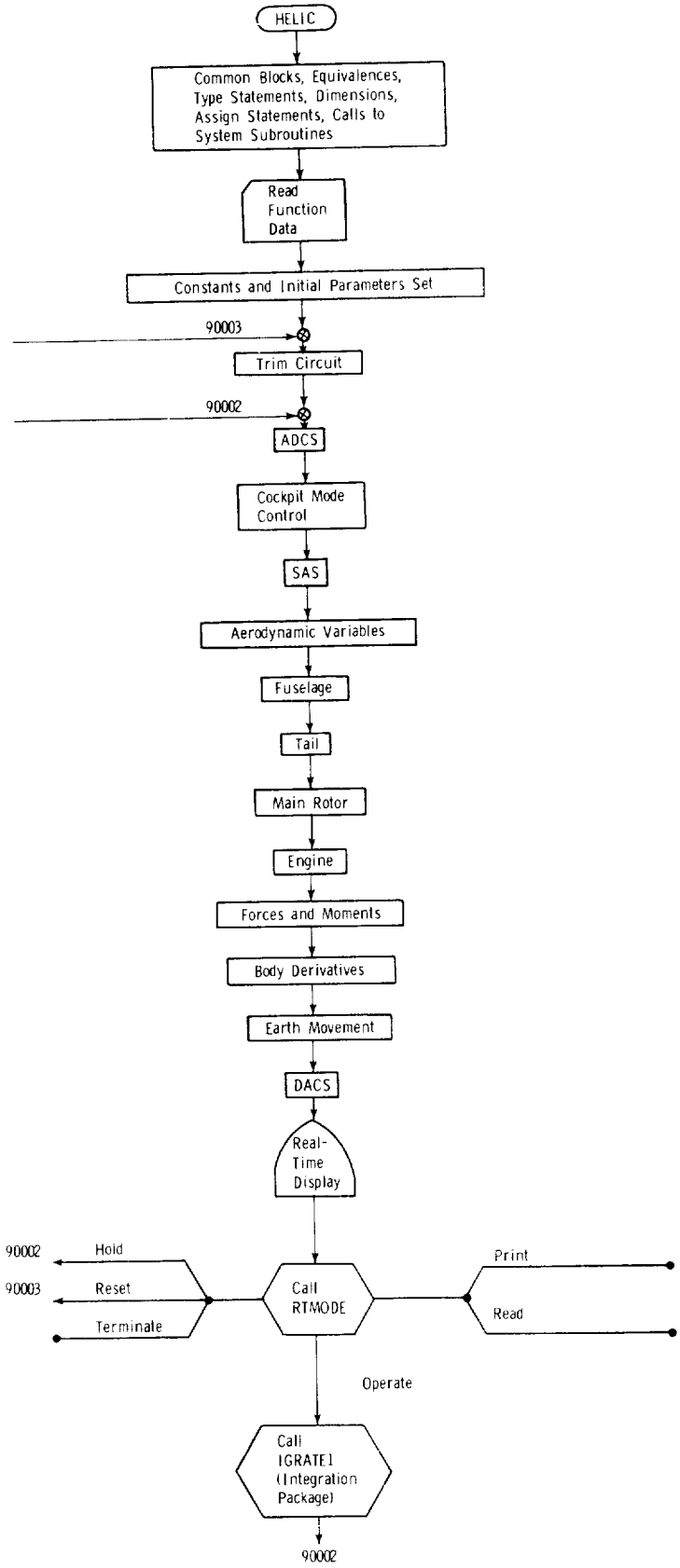
Subroutine	Description
SCANNER	Increments display address
SYNCH	Issues a dayfile message that indicates the current instruction that was processing when synchronization with real time was lost
TYPEVAR	Types data displayed on digital decimal display unit
XDSPLAY	Initializes data entry keyboard and digital decimal display unit, routes program listing

Program Subroutine Descriptions, Flow Charts, and Listings

This section contains a brief description of the main program and each of the subroutines. A simplified flow chart and a FORTRAN listing are given at the end of each description. The principal aircraft equations used in the simulation program and the references for their sources are contained in appendix A.

HELIC

Helic is the main program from which all the subroutines are called. HELIC contains the COMMON and EQUIVALENCE statements needed for communication with the various subroutines and the other FORTRAN declarative statements such as DIMENSION, REAL, and LOGICAL. The main parts of the HELIC program are the real-time system software, the program constants, the variable initializations, the aircraft trim circuit, and the calls to the subroutines.


HELIC
Common Blocks, Equivalences, Type Statements, Dimensions, Assign Statements, Calls to System Subroutines
Read Function Data
Constants and Initial Parameters Set
90003
Trim Circuit
90002
ADCS
Cockpit Mode Control
SAS
Aerodynamic Variables
Fuselage
Tail
Main Rotor
Engine
Forces and Moments
Body Derivatives
Earth Movement
DACS
Real-Time Display
90002
Hold
90003
Reset
Terminate
Call RTMODE
Print
Read
Operate
Call IGRATE1 (Integration Package)
90002

```
      PROGRAM HELIC (INPUT=201, OUTPUT=201)
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL IHS, IVS, IROT, IW, IXX, IYY, IZZ, LA, LF, LMR, LRH, MASS,
     X     MRH, MU , MA , MF, MTIP, N,  NDOT, NA, NF, NO
      LOGICAL AUTOH, AUTOTYP, INTABLS, LDISI, LDISO, LOGIC,
     X        TYPESW,VARCHNG, MRESET , MHOLD, MOPER
      DIMENSION IVARBUF(5), LOGIC(01), TIME(2), MESAGE(2),
     X          XTRIM(11) , FTRIM(11), QVEC(12)
C
C** REAL TIME CONTROL COMMUNICATION
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X       /CHECK  / TABLE(150), INTEG(09)
     X       /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                 ISCHEME, DERINT(2,12)
     X       /INTINTR/ INTERN(5,12)
     X       /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                 NOPER  , NHOLD  , NRESET    , NTERM     ,
     X                 NPRINT , NREAD
C
C                              SUBROUTINE COMMUNICATION
C
      COMMON
     X       /ACCEL/   AX      , AY     , AZ
     X       /ADV C  / OMEG    , WIM    , VT      , QV     , QL     , ASPD
     X       /ADV P /  ADVP1   , ADVP2
     X       /ANGSAS / PHIL    , THETAL , PSIL    , PHIDL  , THETDL , PSIDL
     X       /CNTL C / AOSC    , A1SC   , B1SC    , THTRC  , A1SCL  , B1SCL ,
     X                 THTRCL
     X       /CNTL P / AOSC0   , A1SC0  , B1SC0   , THTRC0 , XAOSG  , YCSG  ,
     X                 XCSG    , XTRG   , XAOSR   , YCSR   , XCSR   , XTRR
     X       /CNTSAS / A1SCS   , B1SCS  , THTRCS
     X       /COEF   / CT      , CP     , CQ      , MU     , MTIP   , VS
     X       /CONTRL/  AOS     , A1S    , B1S     , THTR   , OMEGD
     X       /CPIC   /  ACI    , BCI    , DCI     , XRCM1  , YRCM1  , XRTM1 ,
     X                 YRTM1   , XPBM1  , YPBM1   , CSK4   , CSK6   , CSK7  ,
     X                 F05(18)
     X       /ENG C  / QMAX    , QE     , SHP     , GRWT   , OMEGDT
     X       /ENG P /  IROT    , CKE1
     X       /ERROR /  NERR
      COMMON
     X       /EUL CS/  COSFI   , SINFI  , COSTH   , SINTH  , COSSI  , SINSI
     X       /FCNNAME/ CLF1    , CMF1   , CNF1    , CX1    , CX2    , CYTR  ,
     X                 CYF     , CZF    , CXW     , CZW    , CYVS   , CZHS  ,
     X                 CDSIY   , CLSIY  , DW      , GEV    , GEH
     X       /F N M C/ XA      , YA     , ZA      , LA     , MA     , NA
     X       /F N M P/ DX      , DXHS   , DXTR    , DXVS   , DZ     ,
     X                 DZTR    , DZVS   , DXW     , DZW    , IR
     X       /FTRMPLS/ EQLMR   , EQAOSS , EQWIM   , HDOTD
     X       /FUNCS /  F001(28), F002(28), F003(14), F004(14), F005(14),
     X                 F006(28), F007(14), F008(28), F009(14), F010(28),
     X                 F011(35), F013(07), F014(07), F015(07), D018(27),
     X                 D001(09), D002(09), D003(09), D004(09), D005(09),
     X                 D006(09), D007(09), D008(09), D009(09), D010(09),
     X                 D011(09), D013(09), D014(09), D015(09), D016(18),
     X                 D017(18), F016(35,10),  F017(35,10),  F018(7,3,4)
     X       /FUS C /  XF      , YF     , ZF      , LF     , MF     , NF    ,
     X                 XW      , WIWM   , ZW      , ALFF   , BETF   , ALFW
      COMMON
     X       /FUS P /  CLF2    , CMF2   , CNF2    , CKRF   , CKWIWM , IW    ,
     X                 VLF     , VMF    , VNF     , SXF1   , SXF2   , SYF   ,
     X                 SZF     , SXW    , SZW
     X       /GPARAM/  DELT    , RHO    , G       , PIE
     X       /MROT C/  FXR     , FYR    , LMR     , LRH    , MRH    , QMR   ,
     X                 AOSS    , ALFSIY , UTSIY   , BETSI  , BDTSI  , AISS  ,
     X                 A01SS   , B1SS   , B01SS
     X       /RCOEFF / CKL(20), CKD(20), CKQ(20), CKDL(20), CKQL(20),
     X                 CKM(20), Y(20)  , WF(20)
     X       /ROTCON / YPE(20), YTW(20),WIF(20), BOVN
     X       /ROT P /  TWIST   , FMRS   , CKA0    , CKAB1  , CKAB2  , CKAB3 ,
     X                 NRAD    , NAZ    , NB      , RB     , EFH
     X       /LAGSTK / NILAG   , NSTICK
     X       /MODEC  / MRESET , MHOLD , MOPER
     X       /SAS P  / A1C1    , A1C2   , A1K1    , A1K2   , A1K3   , A1K4  ,
     X                 B1C1    , B1C2   , B1K1    , B1K2   , B1K3   , B1K4  ,
     X                 THC1    , THC2   , THK1    , THK2   , THK3   , THK4
      COMMON
     X       /SCALES / SF1     , SF2    , SF3     , SF4    , SF5    , SF6   ,
     X                 SF7
     X       /SCPSIR / SINSIR(50),  COSSIR(50)
     X       /SHIP P/  IXX     , IYY    , IZZ     , MASS
     X       /STICKS / XAOS    , YCS    , XCS     , XTR    , DPR    , RPD
     X       /STINPUT/ X1      , X2     , X3      , X4     , T1     , T2
```

```
     X       /TAIL C / ALFHS  , BTVS  , ZHS   , YVS   , VTR   ,
     X                 WHS    , YTR   , DELE
     X       /TAIL P/  CKTR1  , CKTR2 , IHS   , IVS   , SZHS  , SYVS
     X       /TRIMAT/  XMAT(11,12), EMAT(12,13), TRSUM(12), NEQN,
     X                 NEQP1      , NEQP2      , XMATG(11)
     X       /VALUES / A1CS1  , A1CS2 , B1CS1 , B1CS2 , THCS1 , THCS2
     X       /XIC    / A1X1   , A1X2  , A1X3  , B1X1  , B1X2  , B1X3  ,
     X                 THX1   , THX2  , THX3  , AX1TN , AX2TN , BX1TN ,
     X                 BX2TN  , TX1TN , TX2TN
C
C                              EQUIVALENCES
C
      EQUIVALENCE(INTEG( 1),NSTICKO),   (INTEG( 2),NT    ),
     X           (INTEG( 3),IRUN   ),   (INTEG( 4),NILAGO),
     X           (INTEG( 5),INTSCME),   (INTEG( 6),KNOTS )
C
      EQUIVALENCE(LDIS1(46),AUTOTYP), (LDIS1(47),TYPESW),
     X           (LDIS1(48),INTABLS)
C
      EQUIVALENCE
     X           (DERINT(1, 1),P    ),    (DERINT(2, 1),PDOT  ),
     X           (DERINT(1, 2),Q    ),    (DERINT(2, 2),QDOT  ),
     X           (DERINT(1, 3),R    ),    (DERINT(2, 3),RDOT  ),
     X           (DERINT(1, 4),PHI  ),    (DERINT(2, 4),PHID  ),
     X           (DERINT(1, 5),THETA),    (DERINT(2, 5),THETDT),
     X           (DERINT(1, 6),PSI  ),    (DERINT(2, 6),PSIDT ),
     X           (DERINT(1, 7),U    ),    (DERINT(2, 7),UDOT  ),
     X           (DERINT(1, 8),V    ),    (DERINT(2, 8),VDOT  ),
     X           (DERINT(1, 9),W    ),    (DERINT(2, 9),WDOT  ),
     X           (DERINT(1,10),N    ),    (DERINT(2,10),NDOT  ),
     X           (DERINT(1,11),E    ),    (DERINT(2,11),EDOT  ),
     X           (DERINT(1,12),H    ),    (DERINT(2,12),HDOT  )
C
      EQUIVALENCE(AOS0, TABLE(40)),    (A1S0  , TABLE(41)),
     X           (B1S0, TABLE(42)),    (THTR0 , TABLE(43)),
     X           (U0  , TABLE(44)),    (V0    , TABLE(45)),
     X           (W0  , TABLE(46)),    (PSI0  , TABLE(47)),
     X           (PHI0, TABLE(48)),    (THETA0, TABLE(49)),
     X           (UKNOTS,TABLE(50)),   (HDOTD0, TABLE(51)),
     X           (H0  , TABLE(52)),    (OMEG0 , TABLE(53)),
     X           (CG0 , TABLE(54)),    (GRWT0 , TABLE(55)),
     X           (P0  , TABLE(56)),    (Q0    , TABLE(57)),
     X           (R0  , TABLE(58))
C
      EQUIVALENCE(A1SSTEP, TABLE(60)),    (B1SSTEP, TABLE(61)),
     X           (THTRSTP, TABLE(62)),    (A1SGN  , TABLE(63)),
     X           (B1SGN  , TABLE(64)),    (THTRGN , TABLE(65)),
     X           (AOSGN  , TABLE(66)),    (PGN    , TABLE(67)),
     X           (QGN    , TABLE(68)),    (RGN    , TABLE(69)),
     X           (UGN    , TABLE(70)),    (VGN    , TABLE(71)),
     X           (WGN    , TABLE(72)),    (HGN    , TABLE(73)),
     X           (PSIGN  , TABLE(74)),    (PHIGN  , TABLE(75)),
     X           (THEGN  , TABLE(76)),    (A1SSGN , TABLE(77)),
     X           (B1SSGN , TABLE(78)),    (ALFFGN , TABLE(79)),
     X           (BETFGN , TABLE(80)),    (BETSGN , TABLE(81)),
     X           (BDTGN  , TABLE(82)),    (DTCON  , TABLE(83)),
     X           (X10    , TABLE(84)),    (X20    , TABLE(85)),
     X           (X30    , TABLE(86)),    (X40    , TABLE(87)),
     X           (T10    , TABLE(88)),    (T20    , TABLE(89)),
     X           (SF10   , TABLE(90)),    (SF20   , TABLE(91)),
     X           (SF30   , TABLE(92)),    (SF40   , TABLE(93)),
     X           (SF50   , TABLE(94)),    (SF60   , TABLE(95)),
     X           (SF70   , TABLE(96))
C
      DATA MESAGE/ 2HON, 3HOFF /
C
C** INITIAL SET-UP OF REAL TIME SYSTEM
C
      CALL NM218(5LTYPER)
      CALL CYCLE(90006S)
      CALL LOSTIME(77777S)
C
C** CALL   READOUT SETS UP THE REAL TIME RECORDING FILE
C
      NT= 32
      CALL READOUT(4,NT, T    , ASPD , GRWT , CG    )
      CALL READOUT(4,NT, H    , HDOT , OMEG , OMEGDT)
      CALL READOUT(4,NT, U    , V    , W    , ALFF  )
      CALL READOUT(4,NT, UDOT , VDOT , WDOT , BETF  )
      CALL READOUT(4,NT, P    , Q    , R    , DELE  )
      CALL READOUT(4,NT, PDOT , QDOT , RDOT , LMR   )
      CALL READOUT(4,NT, XA   , YA   , ZA   , QMR   )
      CALL READOUT(4,NT, LA   , MA   , NA   , FXR   )
      CALL READOUT(4,NT, PHI  , THETA, PSI  , FYR   )
      CALL READOUT(4,NT, PHID , THETDT, PSIDT, QE   )
      CALL READOUT(4,NT, AOSS , A1SS , B1SS , WIM   )
      CALL READOUT(4,NT, AOS  , A1S  , B1S  , THTR  )
      CALL READOUT(4,NT, XAOS , YCS  , XCS  , XTR   )
```

```
      CALL READOUT(4,NT, N     , NDOT  , E     , EDOT  )
      CALL READOUT(4,NT, QMAX  , SHP   , CT    , VT    )
      CALL READOUT(2,NT, QV    , QL                    )
C
C** FORMAT NO.SUPPLIED TO SUPERVISOR FOR PRINTING JOB CARD FROM MF FILE
C
      CALL RTROUTE(MFI,900?4S)
C
C** CALL INOUT SETS UP I/O TO REQUESTED EQUIPMENT
C
      CALL INOUT(ADC,8, DAC,48, LDISI,60, LDISO,180)
      CALL DATABLX(TABLE,1?0, INTEG,9, LOGIC,01, ADC,8, DAC,48,
     X             LDISI, 60, LDISO, 180)
      CALL XDSPLAY(LDISI, LDISO, VARCHNG, ITYPE, IVARBUF, INTABLS)
C
C** ASSIGN STATEMENTS SET UP MODE CONTROL RETURNS TO PROGRAM
C
      ASSIGN 90001 TO NOPER
      ASSIGN 90002 TO NHOLD
      ASSIGN 90003 TO NRESET
      ASSIGN 90004 TO NTERM
      ASSIGN 90014 TO NPRINT
      ASSIGN 90015 TO NREAD
C
C** LOGICAL DISCRETES SET TO FALSE
C
      DO 10 IND = 1,8
   10 ADC(IND)  = 0.
      DO 20 IND = 1,48
   20 DAC(IND)  = 0.
      DO 30 IND = 1,60
   30 LDISI(IND)= .F.
      DO 40 IND = 1,180
   40 LDISO(IND)= .F.
C
C** FUNCTION DATA READ IN
C
      READ 98,  F001
      READ 98,  F002
      READ 98,  F003
      READ 98,  F004
      READ 98,  F005
      READ 98,  F006
      READ 98,  F007
      READ 98,  F008
      READ 98,  F009
      READ 98,  F010
      READ 98,  F011
      READ 98,  F013
      READ 98,  F014
      READ 98,  F015
      READ 98,  F016
      READ 98,  F017
      READ 98,  F018
   98 FORMAT(7F10.0)
C
C** CONSTANTS AND INITIAL PARAMETERS
C
C** INITIAL LOGICAL VARIABLES
      LOGIC(01) = .F.
      VARCHNG   = .F.
      MRESET    = .T.
      MHOLD     = .F.
      MOPER     = .F.
C
C** INITIAL INTEGER PARAMETERS
      ICOUNT  = 1
      INTSCME = 5
      IRUN    = 1
      KNOTS   = 3
      KSCAN   = 32
      NEQ     = 12
C
C** INITIAL DOUBLET INPUTS
      A1SSTEP = 0.
      B1SSTEP = 0.
      THTRSTP = 0.
C
C** SAS CONSTANTS
      A1X1    = 0.
      A1X2    = 0.
      A1X3    = 0.
      B1X1    = 0.
      B1X2    = 0.
      B1X3    = 0.
      THX1    = 0.
      THX2    = 0.
      THX3    = 0.
```

```
      AX1TN    = 0.
      AX2TN    = 0.
      BX1TN    = 0.
      BX2TN    = 0.
      TX1TN    = 0.
      TX2TN    = 0.
C
C* PROGRAMMED CONTROL STICK DEFLECTIONS
      X10      = 0.0
      X20      = 0.0
      X30      = 0.0
      X40      = 0.0
      T10      = 0.0
      T20      = 0.0
C
C** I.C. S ON INTEGRATED VARIABLES
      P0       = 0.
      Q0       = 0.
      R0       = 0.
      PSI0     = 0.
      U0       = 0.
      V0       = 0.
      W0       = 0.
      N0       = 0.
      E0       = 0.
      H0       = 5000.
      REVTRAD  = 30./PIE
      OMEGO    = 33.9*REVTRAD
C
C**   IC'S ON DERIVATIVES USED IN TRIM
      PDOT     = 0.
      QDOT     = 0.
      RDOT     = 0.
      PHID     = 0.
      THETDT   = 0.
      PSIDT    = 0.
      UDOT     = 0.
      VDOT     = 0.
      WDOT     = 0.
      NDOT     = 0.
      EDOT     = 0.
      HDOT     = 0.
C
C**   RECORDER GAINS
      A0SGN    = 1.
      A1SGN    = 10.
      B1SGN    = 10.
      THTRGN   = 5.
      XA0SGN   = .0666666666
      XCSGN    = .10
      YCSGN    = .10
      XTRGN    = .10
      PGN      = 2.5
      QGN      = 2.5
      RGN      = 2.5
      PHIGN    = .025
      THEGN    = .05
      PSIGN    = .05
      UGN      = .003125
      VGN      = .025
      WGN      = .025
      HGN      = .0001
      ALFFGN   = .05
      BETFGN   = .05
      A1SSGN   = 10.
      B1SSGN   = 20.
      BETSGN   = 10.
      BDTGN    = .333333
C
C*  COCKPIT CONSTANTS
      NILAG0   = 0
      NSTICK0  = 0
      H7       = 32./1024.
      TLAG     = 1.0
      AC1      = 1.0 - (TLAG/H7)*(1.0 - EXP(-H7/TLAG))
      BC1      = (TLAG/H7)*(1.0 - (1.0 + H7/TLAG)*EXP(-H7/TLAG))
      DC1      = EXP(-H7/TLAG)
      CSK4     = PIE/500.
      CSK6     = -10.
      CSK7     = 20.
      XPBM1    = 0.0
      YPBM1    = 0.0
      XRCM1    = 0.0
      YRCM1    = 0.0
      XRTM1    = 3.1416
      YRTM1    = 3.1416
      SF10     = 16.625
      SF20     = 8.00
```

```
      SF30    =-8.00
      SF40    = 5.00
      SF50    = 5.00
      SF60    =-5.00
      SF70    = 4.00
      F05( 1) = 0.0
      F05( 2) = 5.2170
      F05( 3) = 10.500
      F05( 4) = 13.734
      F05( 5) = 19.183
      F05( 6) = 23.383
      F05( 7) = 27.833
      F05( 8) = 33.567
      F05( 9) = 37.350
      F05(10) = 41.750
      F05(11) = 44.350
      F05(12) = 61.990
      F05(13) = 77.930
      F05(14) = 83.834
      F05(15) = 106.784
      F05(16) = 116.600
      F05(17) = 128.336
      F05(18) = 128.336
C
      AOSS    = .0001
      AISS    = 0.
      BISS    = 0.
      BETS1   = 0.
      CGO     = 193.85
      CNTERR  = .01
      DPR     = 57.2957795
      DTCON   = 1.
      DTO     = .03125
      GRWTO   = 8822.8
      HDOTDO  = 0.
      LMR     = 9000.
      QF      = 0.
      RPD     = 0.01745329
      WIM     = 0.
      DO 75 J=1,11
   75 XMATG(J) = .001
C
C** COMPUTE ROTOR BLADE PARAMETERS
C
      DO 50 J = 1, NRAD
      YPE(J) = Y(J) + EFH
      YTW(J) = TWIST*YPE(J)/RB
      WIF(J) = 3.685*Y(J)/((OMEGO/REVTRAD)*RB*RB)
   50 CONTINUE
C
C** COMPUTE SINES AND COSINES AROUND AZIMUTH
C
      DO 60 J = 1, NAZ
        PSIR      = 2.*PIE*FLOAT(J-1)/FLOAT(NAZ)
        COSSIR(J) = COS(PSIR)
        SINSIR(J) = SIN(PSIR)
   60 CONTINUE
C
      BOVN  = FLOAT(NB)/FLOAT(NAZ)
C
C** BEGINNING OF REAL TIME LOOP
C
      CALL READY
C
C** BEGINNING OF RESET LOOP
C
90003 CONTINUE
      IF(MHOLD) GO TO 90002
      IF(MOPER) GO TO 90001
C
C** SELECT INITIAL CONDITIONS
      GO TO (901,902,903,904,905,906), KNOTS
C
  901 CONTINUE
C** HOVER INITIAL CONDITIONS
C
      AOSO  =  0.2738429
      AISO  = -0.031216394
      BISO  = -0.063409593
      THTRO =  0.1392389
      PHIO  = -0.019283726
      THETAO= -0.062152122
      UKNOTS= 0.
      VO    =  0.
      WO    =  0.
      GO TO 906
C
  902 CONTINUE
```

```
C** 33 KNOTS INITIAL CONDITIONS
C
      AOSO   = .2462616
      A1SO   =-.024896768
      B1SO   =-.044236309
      THTR0  = .080909505
      PHI0   =-.014087214
      THETA0=-.073933757
      UKNOTS= 33.15571344
      V0     = 0.00
      W0     =-4.148259
      GO TO 906
C
  903 CONTINUE
C** 80 KNOTS INITIAL CONDITIONS
C
      AOSO   = .2396854
      A1SO   =-.016621453
      B1SO   =-.011830726
      THTR0  = .029313655
      PHI0   =-.017073581
      THETA0=-.097307773
      UKNOTS= 79.928952
      V0     = 0.00
      W0     =-13.18009
      GO TO 906
C
  904 CONTINUE
C** 105 KNOTS INITIAL CONDITIONS
C
      AOSO   = .2591912
      A1SO   =-.018925199
      B1SO   = .021399861
      THTR0  = .031807306
      PHI0   =-.026224227
      THETA0=-.1158838
      UKNOTS= 105.1509769
      V0     = 0.00
      W0     =-20.68070
      GO TO 906
C
  905 CONTINUE
C** 120 KNOTS INITIAL CONDITIONS
C
      AOSO   = .2791526
      A1SO   =-.021712153
      B1SO   = .045708954
      THTR0  = .038308344
      PHI0   =-.034026789
      THETA0=-.1291323
      UKNOTS= 120.0
      V0     = 0.00
      W0     =-26.33423
C
  906 CONTINUE
      U0 = UKNOTS*1.689
      KNOTS = 6
C
      AOSC     = AOSO
      A1SC     = A1SO
      B1SC     = B1SO
      THTRC    = THTR0
C
C** TRIM CIRCUIT
C** IF FUNCTION SWITCH 5 IS ON, ITERATE CONTROL DEFLECTIONS
C** UNTIL STATE VARIABLE DERIVATIVES ARE ZERO.
C
      IF(LDISI(37))   GO TO 80
      ICOUNT = 1
      MCOUNT  = 1
      C2      = .01
      NT1     = 1
      NITER   = 0
      LDISO(61)  = .F.
      LDISO(111) = .F.
      GO TO 100
   80 CONTINUE
      IF (MCOUNT.GT.1) GO TO 100
 1000 CALL HALT
      LDISO(61)  = .F.
      LDISO(111) = .F.
            = 10
                + 1
```

```
C
      CALL SETUPA(XTRIM,THETA0,PHI0,A0S0,A1S0,B1S0,THTR0,WIM,AOSS,
     1            A1SS ,B1SS  ,LMR ,1)
 1005 CALL XMATRIX(XTRIM)
C
C** CALCULATE THE EMATRIX
C
      DO 1010 NCOL = 1,NEQP1
      CALL XVAL(XTRIM,NCOL)
      CALL SETUPA(XTRIM,THETA,PHI,A0S,A1S,B1S,THTR,WIM,A0SS,A1SS,
     1            B1SS ,LMR  ,2)
      CALL TRIMDER
      CALL SETUP2(FTRIM,UDOT,VDOT,WDOT,PDOT,QDOT,RDOT,EQWIM,EQAOSS,
     1            A01SS,B01SS,EQLMR)
      CALL EVAL(FTRIM,NCOL)
 1010 CONTINUE
C
C** FIND THE MINIMUM SUM OF F-MATRIX COLUMNS
C
      SUMMIN = TRSUM(1)
      LL = 1
      DO 1020 K=2,NEQP1
      IF (SUMMIN.LT.TRSUM(K)) GO TO 1020
      LL = K
      SUMMIN = TRSUM(K)
 1020 CONTINUE
C
C** FIND THE MAXIMUM SUM
C
 1030 SUMMAX = TRSUM(1)
      KK = 1
      DO 1040 J=2,NEQP1
      IF (SUMMAX.GT.TRSUM(J)) GO TO 1040
      KK = J
      SUMMAX = TRSUM(J)
 1040 CONTINUE
C
C** SET UP MATRIX EQUATION - EMAT*QVEC = EMAT + 1
C
      EMAT(1,NEQP2) = 1.0
 1045 DO 1050 J=2,NEQP1
 1050 EMAT(J,NEQP2) = (1.-C2)*EMAT(J,LL)
C
C** SOLVE MATRIX EQUATION FOR QVEC AND COMPUTE NEW TRIM VECTOR
C
      CALL SIMEQA(EMAT,NEQP1,QVEC,XMAT,XTRIM)
      CALL  SETUPA(XTRIM,THETA,PHI,A0S,A1S,B1S,THTR,WIM,A0SS,A1SS,
     1             B1SS ,LMR  ,2)
      CALL TRIMDER
      CALL SETUP2(FTRIM,UDOT,VDOT,WDOT,PDOT,QDOT,RDOT,EQWIM,EQAOSS,
     1            A01SS,B01SS,EQLMR)
      SUMNEW = 0.
      DO 1070  J = 1,NEQN
 1070 SUMNEW = SUMNEW + FTRIM(J)**2
      IF(SUMNEW .LT. TRMTOL) GO TO 1120
      MCOUNT = MCOUNT + 1
      IF(MCOUNT .GT. 75) GO TO 1135
      IF (SUMNEW.LT.SUMMAX)  GO TO 1090
 1080 C2  = .5*C2
      NT1 = 2
      IF (C2.GT.1.E-5)  GO TO 1045
      GO TO 1140
 1090 IF (NT1.EQ.2) GO TO 1095
      C2  = AMIN1(2.*C2,1.0)
 1095 NT1 = 1
C
C** REPLACE WORST VALUES WITH NEW FUNCTIONS AND ITERATE
C
      TRSUM(KK)  = SUMNEW
      DO 1100  I = 1,NEQN
      XMAT(I,KK) = XTRIM(I)
 1100 EMAT(I+1,KK) = FTRIM(I)
      IF (SUMNEW.GT.SUMMIN)  GO TO 1030
      LL = KK
      SUMMIN = SUMNEW
      GO TO 1030
 1120 CONTINUE
C
C** WHITE LIGHT 1 ON INDICATES VARIABLES ARE TRIMMED
C
      LDIS0(61)  = .T.
 1130 GO TO 1160
C
C** RED LIGHT 1 ON INDICATES UNSUCCESSFUL TRIM
C
 1135 LDIS0(111) = .T.
      GO TO 1160
 1140 CONTINUE
```

```
      C2    = .01
      NTI   = 1
      CALL XVAL(XTRIM,LL)
      NITER = NITER + 1
      IF (NITER.LE.2)  GO TO 1005
      LDISO(111) = .T.
 1160 CONTINUE
      XTRIM(11) = LMR
      CALL SETUPA(XTRIM,THETA0,PHI0,AOS0,A1S0,B1S0,THTR0,WIM,AOSS,A1SS,
     1             B1SS ,LMR  ,2)
C
      W0      = W
      DELT    = DT
      CALL READY
C
  100 CONTINUE
      XAOST = (AOS0 *DPR - AOSC0 )/XAOSG
      XCST  = (B1S0 *DPR - B1SC0 )/XCSG
      YCST  = (A1S0 *DPR - A1SC0 )/YCSG
      XTRT  = (THTR0*DPR - THTRC0)/XTRG
C
C
C** IF FUNCTION SWITCH 13 IS ON ACCEPT
C** CONTROL STICK INPUTS FROM COCKPIT
C
      IF(LDIS1(45))   GO TO 110
      AOSC    = AOS0
      A1SC    = A1S0
      B1SC    = B1S0
      THTRC   = THTR0
  110 CONTINUE
      AOS   = AOSC
      A1S   = A1SC
      B1S   = B1SC
      THTR  = THTRC
C
C
C
      AUTOH   = .F.
C
      NERR    = 0
      INT     = 0
      IPRINT  = 1
C
C**   ISCHEMES: 1=2ND RK. 2=4TH RK. 3=2ND AM.  4=4TH AM.  5 = 1ST AB
      ISCHEME = INTSCME
      IF(INTSCME.EQ.5)   ISCHEME = 3
C
      CG      = CG0
      DT      = DT0/DTCON
      DELT    = DT
      DX      = (CG - 200.)/12.
      GRWT    = GRWT0
      HDOTD   = HDOTD0/60.
      MASS    = GRWT/G
      OMEGD   = OMEG0/REVTRAD
      OMEG    = OMEGD
      T       = 0.
      VS     = 1100.
C
      P       = P0
      Q       = Q0
      R       = R0
      PHI     = PHI0
      THETA   = THETA0
      PSI     = PSI0
      U       = U0
      V       = V0
      W       = W0
      N       = N0
      E       = E0
      H       = H0
C
      NILAG   = NILAG0
      NSTICK  = NSTICK0
C
      X1      = X10
      X2      = X20
      X3      = X30
      X4      = X40
      T1      = T10
      T2      = T20
C
      SF1     = SF10
      SF2     = SF20
      SF3     = SF30
      SF4     = SF40
      SF5     = SF50
```

```
      SF6     = SF60
      SF7     = SF70
C
      A1SCL   = A1SC
      B1SCL   = B1SC
      THTRCL  = THTRC
      PHIL    = PHI
      THETAL  = THETA
      PSIL    = PSI
      PHIDL   = 0.0
      THETDL  = 0.0
      PSIDL   = 0.0
C
C** BEGINNING OF OPERATE LOOP
90006 CONTINUE
90002 CONTINUE
90008 CONTINUE
C
C** IF FUNCTION SWITCH 1 IS ON APPLY
C** DOUBLET INPUT TO SELECTED CONTROL
C
      IF(.NOT.LDISI(33))   GO TO 500
      IF((T.GT.1.).AND.(T.LT.2.))   GO TO 501
      IF((T.GT.2.).AND.(T.LT.3.))   GO TO 502
      A1SC  = A1S0
      B1SC  = B1S0
      THTRC = THTR0
      GO TO 500
  501 CONTINUE
      A1SC  = A1S0  - A1SSTEP
      B1SC  = B1S0  - B1SSTEP
      THTRC = THTR0 - THTRSTP
      GO TO 500
  502 CONTINUE
      A1SC  = A1S0  + A1SSTEP
      B1SC  = B1S0  + B1SSTEP
      THTRC = THTR0 + THTRSTP
  500 CONTINUE
C
C** CALCULATE CONTROLS FROM COCKPIT INPUTS
C
      IF(.NOT.LDISI(45))   GO TO 140
      IF(INT.LE.1)   CALL ADCIN
      LDIS0(81) = .F.
      LDIS0(82) = .F.
      LDIS0(83) = .F.
      LDIS0(84) = .F.
C
C** IF COCKPIT IS ACTIVATED, FUNCTION SWITCH 13 IS ON -
C** COMPARE CONTROL STICK POSITIONS WITH TRIM POSITIONS
C** AND TURN ON WHITE LIGHTS 21,22,23,24 WHEN TRIMMED
C
      IF(ABS(XAOS-XAOST).LE.XAOS*CNTERR)   LDIS0(81)=.T.
      IF(ABS(YCS -YCST ).LE. YCS*CNTERR)   LDIS0(82)=.T.
      IF(ABS(XCS -XCST ).LE. XCS*CNTERR)   LDIS0(83)=.T.
      IF(ABS(XTR -XTRT ).LE. XTR*CNTERR)   LDIS0(84)=.T.
      GO TO 150
C
  140 CONTINUE
      XAOS = (A0SC*DPR - A0SC0)/XAOSG
      XCS  = (B1SC*DPR - B1SC0)/XCSG
      YCS  = (A1SC*DPR - A1SC0)/YCSG
      XTR  = (THTRC*DPR - THTRC0)/XTRG
  150 CONTINUE
C
C** IF FUNCTION SWITCH 11 IS ON, SEND MODE CONTROL TO COCKPIT
C
      IF(.NOT.LDISI(43))GO TO 151
      IF(INT.LE.1)   CALL CPMODE
      GO TO 152
  151 CONTINUE
      MRESET=LDISI(19)
      MHOLD =LDISI(18)
      MOPER =LDISI(17)
  152 CONTINUE
C
C
C** IF FUNCTION SWITCH 9 IS ON, SAS SYSTEM IS
C** ACTIVATED AND WHITE LIGHT 5 IS ON
C
      JMES    = 2
      IF(LDISI(41).OR.(LDISI(44).AND.LDISI(55)))    JMES = 1
      IF(INT.LE.1)   CALL SAS
      LDIS0(65) = .F.
      IF(LDISI(41).OR.(LDISI(44).AND.LDISI(55)))    LDIS0(65) = .T.
      IF(LDISI(41).OR.(LDISI(44).AND.LDISI(55)))    GO TO 160
      A1SCS   = 0.
      B1SCS   = 0.
```

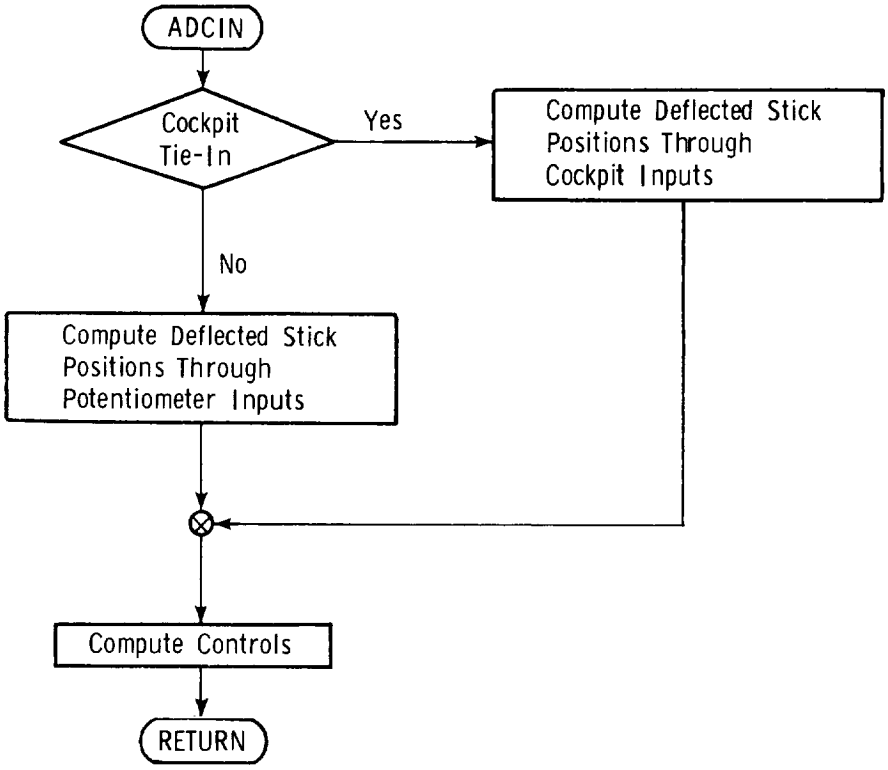
```
      THTRCS  = 0.
      IF(.NOT. LDISI(37))   GO TO 160
      AOSC    = AOS0
      A1SC    = A1S0
      B1SC    = B1S0
      THTRC   = THTR0
  160 AOS  = AOSC
      A1S     = A1SC  + A1SCS
      B1S     = B1SC  + B1SCS
      THTR    = THTRC + THTRCS
      PHIL    = PHI
      THETAL  = THETA
      PSIL    = PSI
      PHIDL   = PHID
      THETDL  = THETDT
      PSIDL   = PSIDT
      A1SCL   = A1SC
      B1SCL   = B1SC
      THTRCL  = THTRC
C
C** RED LIGHTS 2,3,4, OR 5 ON RESPECTIVELY
C** INDICATES AOS, A1S, B1S, OR THTR IS
C** BEYOND SET LIMITS
C
      LDIS0(112) = .F.
      LDIS0(113) = .F.
      LDIS0(114) = .F.
      LDIS0(115) = .F.
      IF(AOS .LT. .1396.OR.AOS .GT..3490)     LDIS0(112) = .T.
      IF(A1S .LT.-.1746.OR.A1S .GT..1222)     LDIS0(113) = .T.
      IF(B1S .LT.-.2305.OR.B1S .GT..2410)     LDIS0(114) = .T.
      IF(THTR.LT.-.1222.OR.THTR.GT..4010)     LDIS0(115) = .T.
C
C** CALL SUBROUTINES
C
      CALL AEROVAR
      CALL FUS
      CALL TAIL
      CALL MAINROT
      CALL ENGINE
      CALL F AND M
      CALL BODYDER
      IF(NERR.EQ.1)  AUTOH=.T.
      CALL EARTH M
      IF(INT.LE.1)   CALL DACOUT
      LDIS0(116) = .F.
      LDIS0(117) = .F.
      IF(AUTOH)   GO TO 170
      IF(INT.GT.1)   90005,180
  170 CONTINUE
C
C** RED LIGHT 6 ON INDICATES PHI OR THETA HAS BECOME TOO LARGE
C** RED LIGHT 7 ON INDICATES NEGATIVE ALTITUDE
C
      IF(NERR.EQ.1)     LDIS0(116) = .T.
      IF(H.LE.0.00)     LDIS0(117) = .T.
  180 CONTINUE
C
C** SCALING OF REAL TIME OUTPUT
C
      DAC( 1) = XAOS*XAOSGN
      DAC( 2) = AOS*AOSGN
      DAC( 3) = XCS*XCSGN
      DAC( 4) = B1S*B1SGN
      DAC( 5) = YCS*YCSGN
      DAC( 6) = A1S*A1SGN
      DAC( 7) = XTR*XTRGN
      DAC( 8) = THTR*THTRGN
      DAC( 9) = P*PGN
      DAC(10) = Q*QGN
      DAC(11) = R*RGN
      DAC(12) = PHI*PHIGN*DPR
      DAC(13) = THETA*THEGN*DPR
      DAC(14) = PSI*PSIGN*DPR
      DAC(15) = ALFF*ALFFGN*DPR
      DAC(16) = BETF*BETFGN*DPR
      DAC(17) = H*HGN
      DAC(18) = U*UGN
      DAC(19) = V*VGN
      DAC(20) = W*WGN
      DAC(21) = A1SS*A1SSGN
      DAC(22) = B1SS*B1SSGN
      DAC(23) = BETSI*BETSGN
      DAC(24) = BDTSI*BDTGN
C
C** REAL TIME DISPLAY
C
      TABLE(1 )= A1SS                    $TABLE(17)=PHI
```

```
      TABLE(2 )= ADISS                    $TABLE(18)=PHID
      TABLE(3 )= BISS                     $TABLE(19)=PSI
      TABLE(4 )= BDISS                    $TABLE(20)=PSIDT
      TABLE(5 )= BETSI                    $TABLE(21)=Q
      TABLE(6 )= BDTSI                    $TABLE(22)=QDOT
      TABLE(7 )= E                        $TABLE(23)=R
      TABLE(8 )= EDOT                     $TABLE(24)=RDOT
      TABLE(9 )= H                        $TABLE(25)=THETA
      TABLE(10)= HDOT                     $TABLE(26)=THETDT
      TABLE(11)= N                        $TABLE(27)=U
      TABLE(12)= NDOT                     $TABLE(28)=UDOT
      TABLE(13)= OMEG                     $TABLE(29)=V
      TABLE(14)= OMEGDT                   $TABLE(30)=VDOT
      TABLE(15)= P                        $TABLE(31)=W
      TABLE(16)= PDOT                     $TABLE(32)=WDOT
      TABLE(33)=AOS                       $TABLE(34)=AIS
      TABLE(35)=BIS                       $TABLE(36)=THTR
      TABLE(37)=OMEGD                     $TABLE(38)=T
      TABLE(125)=CT                       $TABLE(126)=CP
      TABLE(127)=CQ                       $TABLE(128)=MU
      TABLE(129)=MTIP                     $TABLE(130)=ASPD
      TABLE(131)=SHP                      $TABLE(132)=XAOS
      TABLE(133)=YCS                      $TABLE(134)=XCS
      TABLE(135)=XTR
C
      IF(LDISI(22))   CALL SCANNER(KSCAN)
      CALL DSPLAY
      IF(LDISI(17))   GO TO 200
      IF(VARCHNG) CALL TYPEVAR
      IF(ENABLE.AND.TYPESW)   CALL TYPEVAR
90050 CONTINUE
      ENABLE = .NOT.TYPESW
      IF(AUTOTYP.AND.LDISI(14))   CALL TYPEVAR
  200 CONTINUE
      CALL RTMODE
90001 CONTINUE
C
C** RECORD ON REAL TIME FILE
C
      CALL RECORD
C
C
      IF(AUTOH)       CALL RECYCLE
C
C** TIC MARKS RECORDED ON CHART DURING EACH SECOND OF OPERATION
C
      LDISO(102)=.F.
      LDISO(110)=.F.
      IF(AMOD(T,1.).NE.0.) GO TO 230
      LDISO(102)=.T.
      LDISO(110)=.T.
  230 CONTINUE
C
90005 CALL IGRATE1
C
C** IF INTSCME = 5 , USE ADAMS-BASHFORTH INTEGRATION
      IF(INTSCME.EQ.5)   INT = 1
      IF(H.LE.0.000)    AUTOH = .T.
      PSI = AMOD(PSI,6.28318)
C
C** INTEGRATION FLOW CONTROL
C
      IF(INT.LE.1) CALL RECYCLE
      GO TO 90008
C
90014 CONTINUE
      WRITE(MFI,240) IRUN
      IF(IPRINT.GT.1) GO TO 260
      CALL DAYTIM(TIME)
      WRITE(MFI,250) IRUN,TIME(1),TIME(2)
      WRITE(MFI,255) MESAGE(JMES)
      IPRINT = 2
  260 IRUN = IRUN + 1
C
90030 CALL PLAYBAK(900325)
      WRITE(MFI,300)
     X  T     , ASPD , GRWT , CG    ,
     X  H     , HDOT , OMEG ,OMEGDT,
     X  U     , V    , W    , ALFF ,
     X  UDOT  , VDOT , WDOT , BETF ,
     X  P     , Q    , R    , DELE ,
     X  PDOT  , QDOT , RDOT , LMR  ,
     X  XA    , YA   , ZA   , QMR  ,
     X  LA    , MA   , NA   , FXR  ,
     X  PHI   , THETA, PSI  , FYR  ,
     X  PHID  ,THETDT, PSIDT, QE   ,
     X  AOSS  , AISS , BISS , WIM  ,
     X  AOS   , AIS  , BIS  , THTR
```

```
C
C** FUNCTION SWITCH 6 ON SELECTS ALTERNATE PRINTED OUTPUT
C
      IF(LOISI(38))   GO TO 90029
      WRITE(MFI,301)
     X  XAOS  , YCS   , XCS   , XTR   ,
     X  N     , NDOT  , E     , EDOT  ,
     X  QMAX  , SHP   , CT    , VT    ,
     X  QV    , QL
      GO TO 90030
C
90029 CONTINUE
      WRITE(MFI,302)
     X  XF    , YF    , ZF    , VTR   ,
     X  LF    , MF    , NF    , YTR   ,
     X  WHS   , YVS   , ZHS   , WIWM  ,
     X  XW    , ZW
      GO TO 90030
90032 CONTINUE
C
C** FORMAT STATEMENTS
C
  240 FORMAT(125X16)
  250 FORMAT(4X8HRUN NO = I6,8X12HTHE DATE IS A10,8X12H THE TIME IS A10/
     1)
  255 FORMAT(4X7HSAS IS A10/)
  300 FORMAT(1H210X1HT,E20.8,8X4HASPD,E20.8,8X4HGRWT,E20.8,10X2HCG,E20.8
     X/  11X1HH,E20.8,8X4HHDOT,E20.8,8X4HOMEG,E20.8,6X6HOMEGDT,E20.8/
     X   11X1HU,E20.8,11X1HV,E20.8,11X1HW,E20.8,8X4HALFF,E20.8/
     X   8X4HUDOT,E20.8,8X4HVDOT,E20.8,8X4HWDOT,E20.8,8X4HBETF,E20.8/
     X   11X1HP,E20.8,11X1HQ,E20.8,11X1HR,E20.8,8X4HDELE,E20.8/
     X   8X4HPDOT,E20.8,8X4HQDOT,E20.8,8X4HRDOT,E20.8,9X3HLMR, E20.8/
     X   10X2HXA,E20.8,10X2HYA,E20.8,10X2HZA,E20.8,9X3HQMR,E20.8/
     X   10X2HLA,E20.8,10X2HMA,E20.8,10X2HNA,E20.8,9X3HFXR,E20.8/
     X   9X3HPHI,E20.8,7X5HTHETA,E20.8,9X3HPSI,E20.8,9X3HFYR,E20.8/
     X   8X4HPHID,E20.8,6X6HTHETDT,E20.8,7X5HPSIDT,E20.8,10X2HQE,E20.8/
     X   8X4HAOSS,E20.8,8X4HA1SS,E20.8,8X4HB1SS,E20.8,9X3HWIM,E20.8/
     X   9X3HAOS,E20.8,9X3HA1S,E20.8,9X3HB1S,E20.8,8X4HTHTR,E20.8)
  301 FORMAT(1H2
     X   7X4HXAOS,E20.8,9X3HYCS,E20.8,9X3HXCS,E20.8,9X3HXTR,E20.8/
     X   11X1HN,E20.8,8X4HNDOT,E20.8,11X1HE,E20.8,8X4HEDOT,E20.8/
     X   8X4HQMAX,E20.8,9X3HSHP,E20.8,10X2HCT,E20.8,10X2HVT,E20.8/
     X   10X2HQV,E20.8,10X2HQL,E20.8//)
  302 FORMAT(1H2
     X   09X2HXF,E20.8,10X2HYF,E20.8,10X2HZF,E20.8,9X3HVTR,E20.8/
     X   10X2HLF,E20.8,10X2HMF,E20.8,10X2HNF,E20.8,9X3HYTR,E20.8/
     X   9X3HWHS,E20.8,9X3HYVS,E20.8,9X3HZHS,E20.8,8X4HWIWM,E20.8/
     X   10X2HXW,E20.8,10X2HZW,E20.8//)
C
      CALL APRINT
C
90015 CONTINUE
      CALL RELEASE
      CALL READOUT(4,NT, T     , ASPD  , GRWT  , CG    )
      CALL READOUT(4,NT, H     , HDOT  , OMEG  , OMEGDT)
      CALL READOUT(4,NT, U     , V     , W     , ALFF  )
      CALL READOUT(4,NT, UDOT  , VDOT  , WDOT  , BETF  )
      CALL READOUT(4,NT, P     , Q     , R     , DELE  )
      CALL READOUT(4,NT, PDOT  , QDOT  , RDOT  , LMR   )
      CALL READOUT(4,NT, XA    , YA    , ZA    , QMR   )
      CALL READOUT(4,NT, LA    , MA    , NA    , FXR   )
      CALL READOUT(4,NT, PHI   , THETA , PSI   , FYR   )
      CALL READOUT(4,NT, PHID  , THETDT, PSIDT , QE    )
      CALL READOUT(4,NT, AOSS  , A1SS  , B1SS  , WIM   )
      CALL READOUT(4,NT, AOS   , A1S   , B1S   , THTR  )
C
      IF(LOISI(38))   GO TO 90016
      CALL READOUT(4,NT, XAOS  , YCS   , XCS   , XTR   )
      CALL READOUT(4,NT, N     , NDOT  , E     , EDOT  )
      CALL READOUT(4,NT, QMAX  , SHP   , CT    , VT    )
      CALL READOUT(2,NT, QV    , QL                    )
      GO TO 90017
C
90016 CONTINUE
      CALL READOUT(4,NT, XF    , YF    , ZF    , VTR   )
      CALL READOUT(4,NT, LF    , MF    , NF    , YTR   )
      CALL READOUT(4,NT, WHS   , YVS   , ZHS   , WIWM  )
      CALL READOUT(2,NT, XW    , ZW                    )
90017 CONTINUE
      CALL AREAD
C
90004 CONTINUE
      CALL ATERM
77777 CALL SYNCH
      CALL RTMODE
90034 FORMAT(55H JOB,43,28800,65000, C1152, 13912, LUCI GIBSON,  RM2131)
C
      END
```

ADCIN

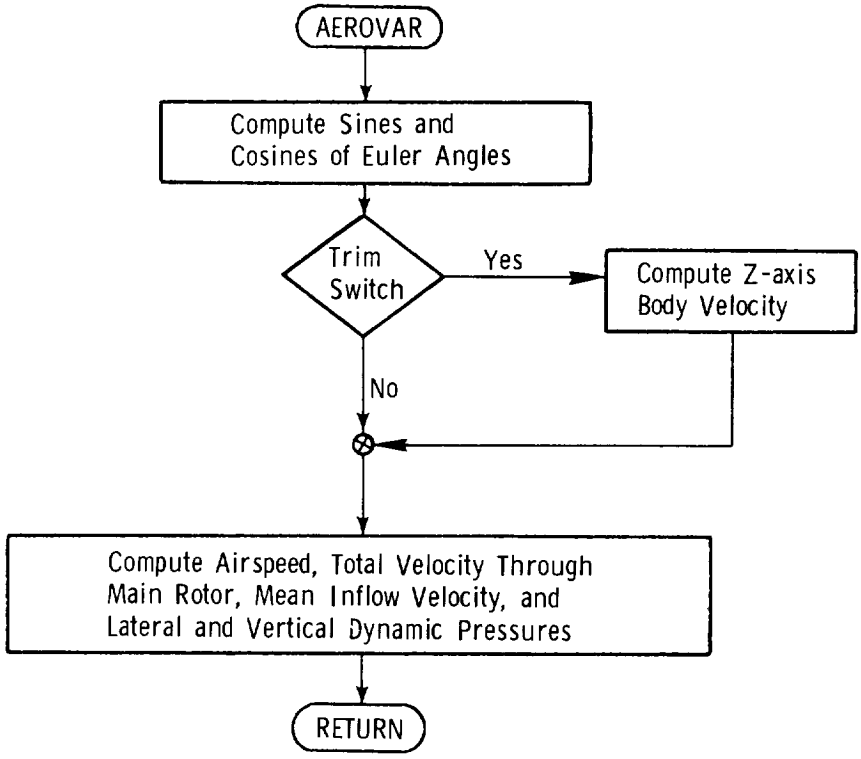
Subroutine ADCIN contains the equations for the helicopter collective and cyclic stick inputs and the tail rotor pedal inputs (ref. 1 and appendix A). The subroutine allows two modes of operation. Primary control is from the control sticks and pedals in the cockpit, and alternate control is from the program control station (fig. 1(a)) through the use of potentiometers (located on the program control console (fig. 1(b)) representing the control sticks and pedals. The alternate mode of operation allows the analyst to conduct preselected flight operations to be used, for example, for comparison with actual flight data.



```
      SUBROUTINE ADCIN
C
C                              PROGRAM PROVIDES ADC INPUTS THROUGH
C                                   POTENTIOMETERS OR COCKPIT
C
C                               TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL LIM
      LOGICAL LDISI
C
C                               INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET    , NTERM     ,
     X                  NPRINT  , NREAD
C
C                               SUBROUTINE COMMUNICATION
      COMMON
     X        /CNTL C / AOSC    , A1SC    , B1SC    , THTRC   , A1SCL   , B1SCL ,
     X                  THTRCL
     X        /CNTL P / AOSCO   , A1SCO   , B1SCO   , THTRCO, XAOSG   , YCSG  ,
     X                  XCSG    , XTRG    , XAOSR   , YCSR    , YCSR    , XTRR
     X        /LAGSTK / NLAG    , NSTICK
     X        /SCALES / SF1     , SF2     , SF3     , SF4     , SF5     , SF6   ,
     X                  SF7
     X        /STICKS / XAOS    , YCS     , XCS     , XTR     , DPR     , RPD
     X        /STINPUT/ X1      , X2      , X3      , X4      , T1      , T2
C
C
      LIM(X,A,B) = AMIN1(B,AMAX1(X,A))
      IF(LDISI(44))    GO TO 102
      IF(NSTICK.NE.1) GO TO 100
      IF(T.LT.T1)      GO TO 100
      IF(T.GT.T2)      GO TO 100
C
      X11      = X1
      X22      = X2
      X33      = X3
      X44      = X4
      GO TO 101
C
  100 CONTINUE
      X11      = 0.
      X22      = 0.
      X33      = 0.
      X44      = 0.
C
C                               STICK DEFLECTIONS THROUGH POTENTIOMETERS
  101 CONTINUE
      XAOS     = ADC(1)*XAOSR  + X11
      XCS      = ADC(3)*XCSR   + X33
      YCS      = ADC(2)*YCSR   + X22
      XTR      = ADC(4)*XTRR   + X44
      GO TO 103
C
C                               COCKPIT STICK DEFLECTIONS
  102 CONTINUE
      XAOSCOL  = LIM(SF1*ADC(1)                ,  0.0, 13.3 )
      XCSCOL   = LIM(SF3*ADC(3) + SF6*ADC(6), -4.8, 4.8  )
      YCSCOL   = LIM(SF2*ADC(2) + SF5*ADC(5), -4.8, 4.8  )
      XTRCOL   = LIM(SF4*ADC(4) + SF7*ADC(7), -3.0, 3.0  )
      XAOS     = XAOSCOL
      XCS      = XCSCOL + 4.8
      YCS      = YCSCOL + 4.8
      XTR      = XTRCOL + 3.0
C
  103 CONTINUE
      AOSC     = (AOSCO + XAOSG*XAOS)*RPD
      A1SC     = (A1SCO + YCSG *YCS )*RPD
      B1SC     = (B1SCO + XCSG *XCS )*RPD
      THTRC    = (THTRCO+ XTRG *XTR )*RPD
C
      RETURN
      END
```

AEROVAR

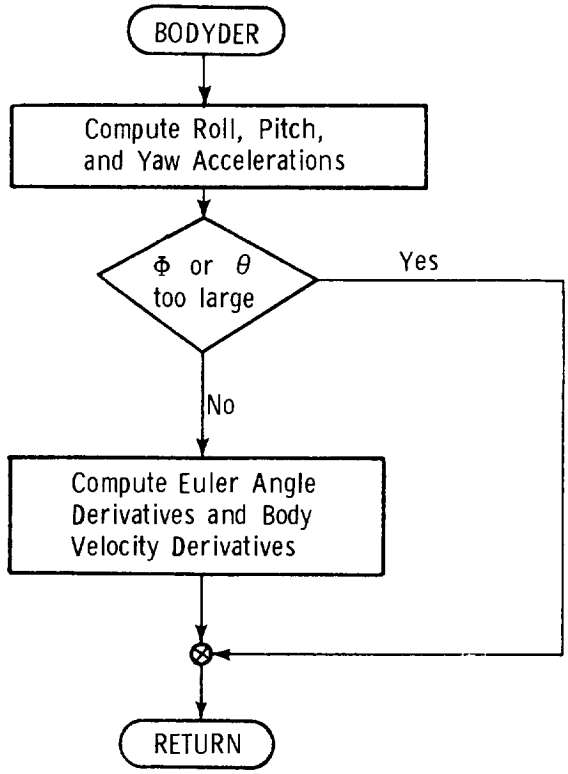
Subroutine AEROVAR computes the main rotor inflow velocity, the lateral and vertical dynamic pressures, and the fuselage airspeed (refs. 1 and 2 and appendix A).



```
      SUBROUTINE AEROVAR
C
C                              PROGRAM COMPUTES ROTOR INFLOW VELOCITY,
C                              FUSELAGE AIRSPEED, AND DYNAMIC PRESSURES
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL LMR
      LOGICAL   LDISI
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT        , NEQ        ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET     , NTERM      ,
     X                  NPRINT , NREAD
C
C                              SUBROUTINE  COMMUNICATION
C
      COMMON
     X        /ADV C  / OMEG    , WIM    , VT      , QV     , QL     , ASPD
     X        /ADV P /  ADVP1   , ADVP2
     X        /EUL CS/  COSFI   , SINFI , COSTH   , SINTH , COSSI , SINSI
     X        /FTRMPLS/ EQLMR   , EQAOSS, EQWIM   , HDOTD
     X        /GPARAM/  DELT    , RHO    , G       , PIE
     X        /MROT C/  FXR     , FYR    , LMR     , LRH    , MRH    , QMR    ,
     X                  AOSS    , ALFSIY, UTSIY   , BETSI , BDTSI , A1SS   ,
     X                  AD1SS   , B1SS   , BD1SS
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 4),PHI   ),
     X            (DERINT(1, 5),THETA),
     X            (DERINT(1, 6),PSI   ),
     X            (DERINT(1, 7),U     ),
     X            (DERINT(1, 8),V     ),
     X            (DERINT(1, 9),W     )
C
C                               COMPUTE SINES AND COSINES OF EULER ANGLES
C
      COSFI = COS(PHI)
      SINFI = SIN(PHI)
      COSTH = COS(THETA)
      SINTH = SIN(THETA)
      COSSI = COS(PSI)
      SINSI = SIN(PSI)
C
C                              COMPUTE LINEAR VELOCITY ALONG Z-AXIS
C                                 AND NET AIR VELOCITY PAST HUB
C
      IF(.NOT. LDISI(37)) GO TO 100
      W      = (U*SINTH - V*SINFI*COSTH - HDOTD)/(COSTH*COSFI)
  100 CONTINUE
      WM     = W - WIM
C
C                              COMPUTE AIRSPEED, TOTAL VELOCITY THROUGH
C                                 MAIN ROTOR,AND MEAN INFLOW VELOCITY
C
      ASPD   = SQRT(U*U + V*V + W*W)/ 1.689
      VT     = SQRT(U*U + V*V + WM*WM)
      IF( VT.LE.0. )       VT = .00001
      WIMP   = ADVP1*LMR/RHO/VT
      IF(INT.LE.1)     WIM = ADVP2*(WIMP - WIM)*DELT + WIM
C
C                              LATERAL AND VERTICAL DYNAMIC PRESSURES
C
      QL     = 0.5*RHO*(U*U + V*V)
      QV     = 0.5*RHO*(U*U + WM*WM)
C
      RETURN
      END
```

BODYDER

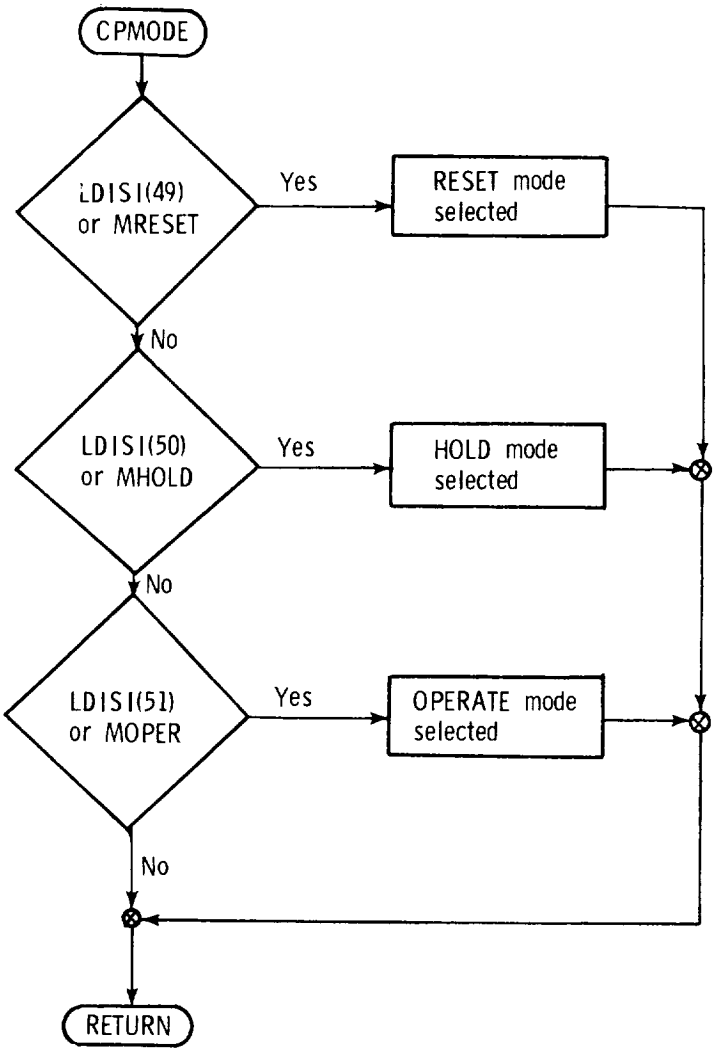
Subroutine BODYDER contains the equations necessary to compute the total body angular and linear accelerations (refs. 1 and 2 and appendix A). The program also contains a safety feature to prevent program abortion if Φ or θ becomes too large.



```
      SUBROUTINE BODYDER
C
C                              PROGRAM COMPUTES BODY MOTION ANGULAR
C                              AND LINEAR ACCELERATIONS AND VELOCITIES
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL LA, MA, NA, IXX, IYY, IZZ, MASS
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT        , NEQ        ,
     X                  ISCHEME, DERINT(2,12)
C
C                              SUBROUTINE  COMMUNICATION
C
      COMMON
     X        /ERROR /  NERR
     X        /EUL CS/  COSFI   , SINFI , COSTH   , SINTH , COSSI , SINSI
     X        /F N M C/ XA      , YA    , ZA      , LA    , MA    , NA
     X        /GPARAM/  DELT    , RHO   , G       , PIE
     X        /SHIP P/  IXX     , IYY   , IZZ     , MASS
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 1),P     ),     (DERINT(2, 1),PDOT   ),
     X            (DERINT(1, 2),Q     ),     (DERINT(2, 2),QDOT   ),
     X            (DERINT(1, 3),R     ),     (DERINT(2, 3),RDOT   ),
     X            (DERINT(1, 4),PHI   ),     (DERINT(2, 4),PHID   ),
     X            (DERINT(1, 5),THETA),     (DERINT(2, 5),THETDT),
     X            (DERINT(1, 6),PSI   ),     (DERINT(2, 6),PSIDT  ),
     X            (DERINT(1, 7),U     ),     (DERINT(2, 7),UDOT   ),
     X            (DERINT(1, 8),V     ),     (DERINT(2, 8),VDOT   ),
     X            (DERINT(1, 9),W     ),     (DERINT(2, 9),WDOT   )
C
C                              COMPUTE ROLL, PITCH, AND YAW ACCELERATIONS
C
      PDOT   = (LA + (IYY-IZZ)*Q*R)/IXX
      QDOT   = (MA + (IZZ-IXX)*R*P)/IYY
      RDOT   = (NA + (IXX-IYY)*P*Q)/IZZ
C
C                                IF PHI OR THETA GETS TOO LARGE PROGRAM
C                             PUT INTO AUTOMATIC HOLD TO PREVENT ABORTION
C                                        WITH A MODE 4
C
      IF(ABS(THETA).GT.1.5.OR.ABS(PHI).GT.1.5)  GO TO 10
C
C                              COMPUTE EULER ANGLE DERIVATIVES
C
      PSIDT = (R*COSFI + Q*SINFI)/COSTH
      THETDT= Q*COSFI - R*SINFI
      PHID  = P + PSIDT*SINTH
C
C                              COMPUTE BODY VELOCITY DERIVATIVES
C
      UDOT   = XA/MASS + R*V - Q*W - G*SINTH
      VDOT   = YA/MASS + P*W - R*U + G*COSTH*SINFI
      WDOT   = ZA/MASS + Q*U - P*V + G*COSTH*COSFI
C
      RETURN
   10 CONTINUE
      NERR   = 1
      RETURN
      END
```

CPMODE

Subroutine CPMODE contains the logic statements necessary to allow mode control of the computer program from the cockpit by the pilot.



```
      SUBROUTINE CPMODE
C
C                                PROGRAM PROVIDES MODE CONTROL TO COCKPIT
C
C                                TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      LOGICAL   LDISI, MRESET, MHOLD, MOPER
C
C                                INTEGRATION COMMUNICATION
C
      COMMON
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER  , NHOLD  , NRESET    , NTERM     ,
     X                  NPRINT , NREAD
C
C                                SUBROUTINE COMMUNICATION
C
      COMMON
     X        /MODEC  / MRESET , MHOLD , MOPER
C
      IF(LDISI(49))   GO TO 100
      IF(LDISI(50))   GO TO 200
      IF(LDISI(51))   GO TO 300
      IF(MRESET)   GO TO 100
      IF(MHOLD)    GO TO 200
      IF(MOPER)    GO TO 300
      GO TO 400
C
C                                RESET MODE
  100 MRESET = .T.
      MOPER = .F.
      MHOLD = .F.
      LDISI(19) = .T.
      LDISI(17) = .F.
      LDISI(18) = .F.
      GO TO 400
C
C                                HOLD MODE
  200 MHOLD = .T.
      MOPER = .F.
      MRESET = .F.
      LDISI(18) = .T.
      LDISI(17) = .F.
      LDISI(19) = .F.

      GO TO 400
C
C                                OPERATE MODE
  300 MOPER = .T.
      MHOLD = .F.
      MRESET = .F.
      LDISI(17) = .T.
      LDISI(18) = .F.
      LDISI(19) = .F.
C
  400 CONTINUE
C
      RETURN
      END
```

DACOUT

Subroutine DACOUT contains the equations and logic necessary to drive the cockpit instrumentation. The following instruments are included:

(1) attitude direction indicator

(2) altimeter

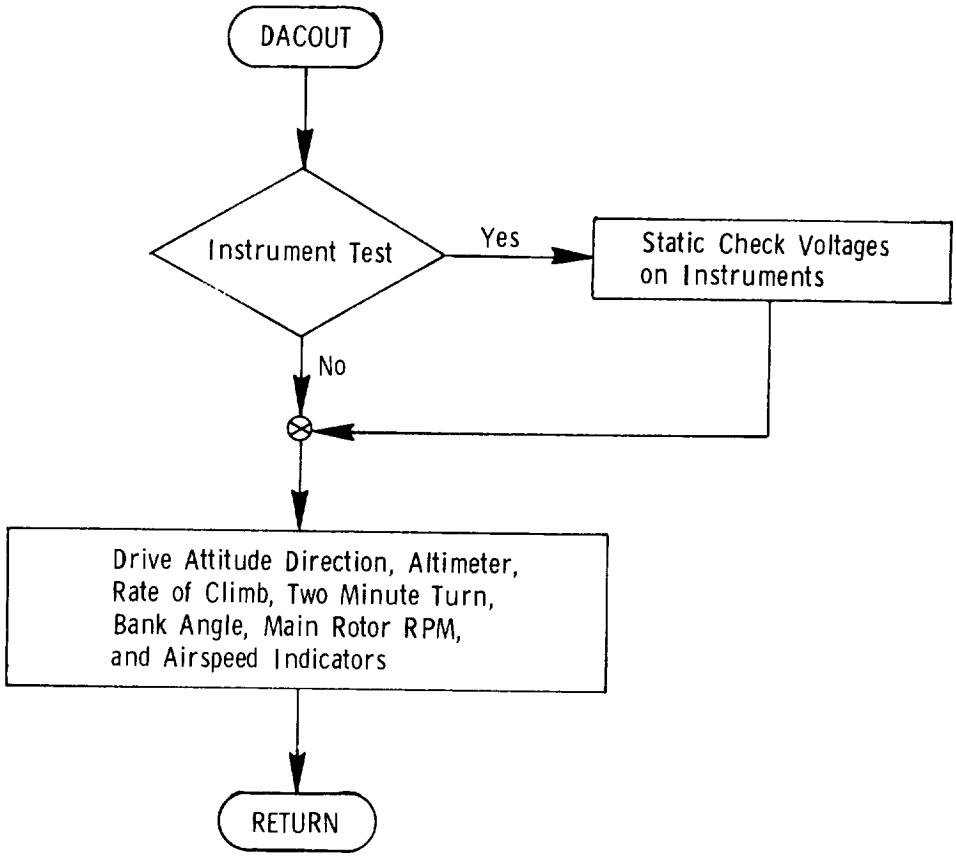
(3) rate of climb indicator

(4) two minute turn indicator

(5) bank angle indicator

(6) main rotor RPM indicator

(7) airspeed indicator

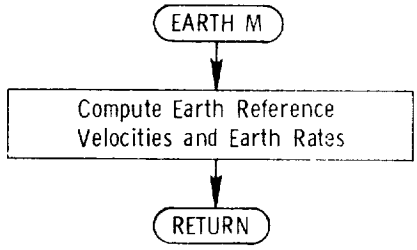


```
      SUBROUTINE DACOUT
C
C                              PROGRAM PROVIDES DACS NEEDED TO
C                                 DRIVE COCKPIT INSTRUMENTS
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
      REAL LIM
      LOGICAL   LDISI, LDISO, MRESET, MHOLD, MOPER
C
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET    , NTERM     ,
     X                  NPRINT  , NREAD
C
C                              SUBROUTINE COMMUNICATION
      COMMON
     X        /ACCEL/   AX      , AY      , AZ
     X        /ADV C  / OMEG    , WIM     , VT        , QV        , QL      , ASPD
     X        /CPIC   /  ACI    , BCI     , DCI       , XRCM1     , YRCM1   , XRTM1 ,
     X                  YRTM1   , XPBM1   , YPBM1     , CSK4      , CSK6    , CSK7  ,
     X                  E05(18)
     X        /GPARAM/  DELT    , RHO     , G         , PIE
     X        /LAGSTK / NILAG   , NSTICK
     X        /MODEC  / MRESET  , MHOLD   , MOPER
     X        /STICKS / XAOS    , YCS     , XCS       , XTR       , DPR     , RPD
C
C                              EQUIVALENCES
      EQUIVALENCE
     X            (DERINT(1, 4),PHI   ),
     X            (DERINT(1, 5),THETA),
     X            (DERINT(1, 6),PSI   ),
     X            (DERINT(1, 3),R     ),
     X            (DERINT(1,12),H     ),
     X            (DERINT(2,12),HDOT  )
C
C
      DATA PALT / 0.0 /
      LIM(X,A,B) = AMIN1(B,AMAX1(X,A))
      PHIBAL  =-AY/AZ
      OMEGRPM = OMEG*DPR/6.0
C
C                               INSTRUMENT TEST
C
      IF(.NOT. LDISI(42))   GO TO 100
      PHI     = 0.524
      THETA   = 0.7854
      PSI     = 0.524
      HDOT    = 25.
      H       = 50.
      PHIBAL  = 0.1
      R       = 0.0262*2.0
      OMEGRPM = 324.
      ASPD    = 130.
  100 CONTINUE
C
C                               ADI
C
      DAC(25) = -COS(PHI)
      DAC(26) = -SIN(PHI)
      DAC(27) = -COS(THETA)
      DAC(28) = -SIN(THETA)
      DAC(29) = -COS(PSI)
      DAC(30) = -SIN(PSI)
C                               ALTIMETER
C
      PALT    = PALT + LIM(H - PALT, -15., 15.)
      AA4     = PIE - AMOD(PALT,1000.)*CSK4
      DAC(31) = COS(AA4)
      DAC(32) = SIN(AA4)
C
C                               RATE OF CLIMB
C
      ZSDP    = LIM(HDOT*60., -3999.99, 3999.99)
      IF(ZSDP)  200, 201, 201
  200 CSK5    = -1.0
      GO TO 202
  201 CSK5    = 1.0
  202 CONTINUE
      IF(ZSDP.GE.1000..OR.ZSDP.LE.-1000.)   GO TO 300
      XX      = 100.
      IDX     = 0
      GO TO 301
  300 XX      = 500.
```

```
      IDX      = 8
  301 SS1      = ZSDP/XX
      IS1      = SS1
      IS2      = IDX*CSK5
      IS       = IABS(IS1 + IS2) + 1
      SS       = ABS(SS1 - IS1)
      ZSDF     = CSK5*((1.0 - SS)*F05(IS) + SS*F05(IS + 1))
      IF(NILAG.EQ.1)   302, 303
  302 AA5X     = ZSDF*RPD
      AA5      = DCI*YRCM1 + BCI*XRCM1 + ACI*AA5X
      XRCM1    = AA5X
      YRCM1    = AA5
      GO TO 304
  303 CONTINUE
      AA5      = ZSDF*RPD
  304 CONTINUE
      DAC(33) = -COS(AA5)
      DAC(34)=  SIN(AA5)
C
C                              TWO MINUTE TURN
C
      IF(NILAG.EQ.1)   400, 401
  400 AA6X     = LIM((R*CSK6 + PIE), 0.0, 6.2831)
      AA6      = DCI*YRTM1 + BCI*XRTM1 + ACI*AA6X
      XRTM1    = AA6X
      YRTM1    = AA6
      GO TO 402
  401 AA6      = LIM((R*CSK6 + PIE), 0.0, 6.2831)
  402 CONTINUE
      DAC(35) = LIM(COS(AA6), -.886, .886)
      DAC(36) = LIM(SIN(AA6), -.500, .500)
C
C                              BANK
C
      IF(NILAG.EQ.1)   500, 501
  500 AA7X     = (LIM(PHIBAL, -.15, .15))*CSK7
      AA7      = DCI*YPBM1 + BCI*XPBM1 + ACI*AA7X
      XPBM1    = AA7X
      YPBM1    = AA7
      GO TO 502
  501 AA7      = (LIM(PHIBAL, -.15, .15))*CSK7
  502 CONTINUE
      DAC(37) = -COS(AA7)
      DAC(38) = -SIN(AA7)
C
C                              RPM
C
      DAC(39) = LIM(OMEGRPM/ 500., 0.0, .9996)
C
C                              INDICATED AIRSPEED
C
      DAC(40)=LIM(ASPD/140.,0.0,0.9996)
C
C                              MODE CONTROL DISCRETES
C
      LDISO(1) = MRESET
      LDISO(2) = MHOLD
      LDISO(3) = MOPER
C
      RETURN
      END
```

EARTH M

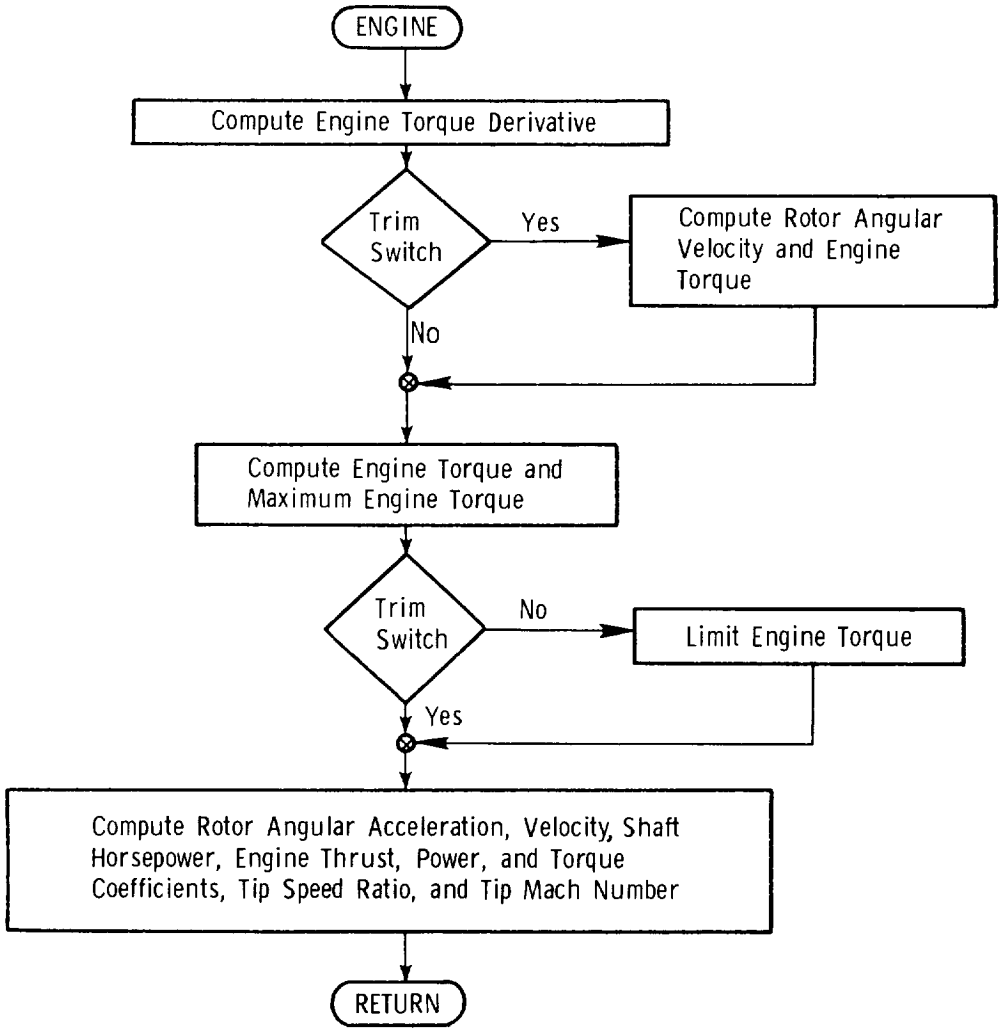
Subroutine EARTH M contains the equations used to calculate the earth rates by resolution of body linear velocities through the Euler angles (refs. 1 and 2 and appendix A).




```
      SUBROUTINE EARTH M
C
C                              PROGRAM COMPUTES EARTH RATES BY RESOLUTION
C                             OF BODY AXIS VELOCITIES THROUGH EULER ANGLES
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL NDOT
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
C
C                              SUBROUTINE  COMMUNICATION
C
      COMMON
     X        /EUL CS/  COSFI   , SINFI , COSTH   , SINTH , COSSI , SINSI
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X             (DERINT(1, 7),U     ),
     X             (DERINT(1, 8),V     ),
     X             (DERINT(1, 9),W     ),
     X             (DERINT(2,10),NDOT  ),
     X             (DERINT(2,11),EDOT  ),
     X             (DERINT(2,12),HDOT  )
C
C                              COMPUTE EARTH REFERENCE VELOCITIES
C
      VPE    = V*COSFI - W*SINFI
      WPE    = V*SINFI + W*COSFI
      UPE    = U*COSTH + WPE*SINTH
C
C                              COMPUTE EARTH RATES
C
      HDOT   = U*SINTH   - WPE*COSTH
      NDOT   = UPE*COSSI - VPE*SINSI
      EDOT   = UPE*SINSI + VPE*COSSI
C
      RETURN
      END
```

ENGINE

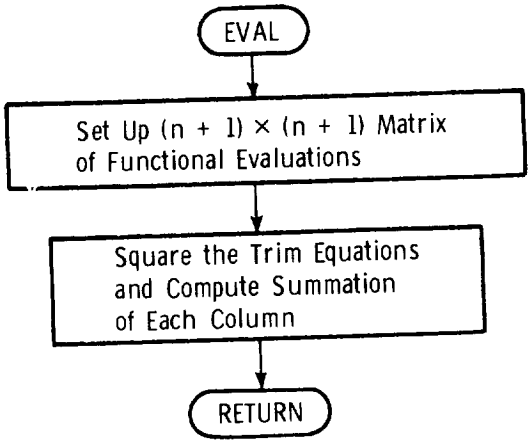
Subroutine ENGINE contains the equations used to calculate required engine torque, maximum engine torque available, and the main rotor angular velocity (refs. 1 and 8 and appendix A).



```
      SUBROUTINE ENGINE
C
C                              PROGRAM COMPUTES REQUIRED ENGINE TORQUE,
C                               MAXIMUM ENGINE TORQUE AVAILABLE, AND
C                                    MAIN ROTOR ANGULAR VELOCITY
C
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL IROT, MU, MTIP
      LOGICAL LDISI, LDISO
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET    , NTERM     ,
     X                  NPRINT  , NREAD
C
C                              SUBROUTINE  COMMUNICATION
C
      COMMON
     X        /ADV C  / OMEG    , WIM     , VT        , QV      , QL      , ASPD
     X        /COEF   / CT      , CP      , CQ        , MU      , MTIP    , VS
     X        /CONTRL/  AOS     , A1S     , B1S       , THTR    , OMEGD
     X        /ENG C  / QMAX    , QE      , SHP       , GRWT    , OMEGDT
     X        /ENG P /  IROT    , CKE1
     X        /GPARAM/  DELT    , RHO     , G         , PIE
     X        /MROT C/  FXR     , FYR     , LMR       , LRH     , MRH     , QMR     ,
     X                  AOSS    , ALFSIY, UTSIY     , BETSI   , BDTSI   , A1SS    ,
     X                  ADISS   , B1SS    , BDISS
     X        /ROT P /  TWIST   , FMRS    , CKA0      , CKAB1   , CKAB2   , CKAB3   ,
     X                  NRAD    , NAZ     , NB        , RB      , EFH
     X        /SHIP P/  IXX     , IYY     , IZZ       , MASS
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 7),U     ),
     X            (DERINT(1,12),H     )
C
C                              COMPUTE ENGINE TORQUE REQUIRED
C                                   AND LIMIT TO MAXIMUM
C
      QED    = CKE1*(OMEGD - OMEG) + QMR
      IF(.NOT. LDISI(37))   GO TO 100
      OMEG   = OMEGD
      QE     = QMR
  100 CONTINUE
      IF(INT.LE.1)   QE = QE + (QED - QE)*DELT
      QMAX   = (202.63158*(H - 2500.) + 412500.)/ OMEG
      IF( H.LT.2500. )     QMAX = 605000./ OMEG
      IF(LDISI(37))   GO TO 200
      IF( QE.GT.QMAX )     QE   = QMAX
  200 CONTINUE
C
C                              RED LIGHT B ON INDICATES
C                          ENGINE TORQUE GREATER THAN MAXIMUM
C
      LDISO(118) = .F.
      IF(QE.GE.QMAX)   LDISO(118) = .T.
C
C                              COMPUTE  ROTOR ANGULAR ACCELERATION
C
      OMEGDT= (QE - QMR)/ IROT
      IF(INT.LE.1)   OMEG = OMEG + OMEGDT*DELT
C
C                              COMPUTE SHAFT HORSEPOWER
C
      SHP    = QMR*OMEG/550.
C
C                              COMPUTE ENGINE THRUST, POWER, AND TORQUE
C                                        COEFFICIENTS
C
      CT     = GRWT/(RHO*PIE*RB*RB*((OMEG*RB)**2))
      CP     = (550.*SHP)/(RHO*PIE*RB*RB*((OMEG*RB)**3))
      CQ     = QMR/(RHO*PIE*RB*RB*RB*((OMEG*RB)**2))
C
C                              COMPUTE TIP SPEED RATIO AND MACH NUMBER
      MU     = (1.689*ASPD)/(OMEG*RB)
      MTIP   = (1.689*ASPD + OMEG*RB) / VS
C
      RETURN
      END
```

EVAL

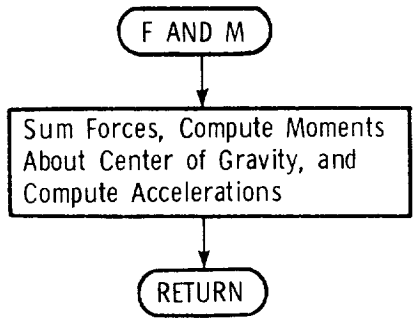
Subroutine EVAL is used by the trim circuit algorithm and sets up an $(n + 1) \times (n + 1)$ matrix of functional evaluations and computes $(n + 1)$ sums where each sum is the sum of the trim equations squared for each column (ref. 6 and appendix D).




```
      SUBROUTINE EVAL(F,NCOL)
C
C                              PROGRAM SETS UP AN (N+1)X(N+1) MATRIX OF
C                              FUNCTIONAL EVALUATIONS AND COMPUTES (N+1)
C                              SUMS WHERE EACH SUM IS THE SUM OF THE TRIM
C                                   EQUATIONS SQUARED FOR EACH COLUMN
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION F(11)
C
C                              SUBROUTINE COMMUNICATION
C
      COMMON
     X        /TRIMAT/  XMAT(11,12), EMAT(12,13), TRSUM(12), NEQN,
     X                  NEQP1      , NEQP2      , XMATG(11)
C
      EMAT(1,NCOL) = 1.0
      TRSUM(NCOL)  = 0.
C
      DO 100  J=2,NEQP1
      EMAT(J,NCOL) = F(J-1)
      TRSUM(NCOL) = TRSUM(NCOL) + EMAT(J,NCOL)**2
  100 CONTINUE
C
      RETURN
      END
```

F AND M

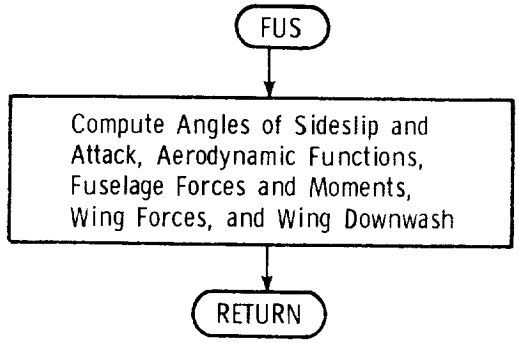
Subroutine F AND M contains the equations used to calculate the total aerodynamic forces and moments (refs. 1 and 2 and appendix A).



```
      SUBROUTINE F AND M
C
C                              PROGRAM COMPUTES NET FORCES AND MOMENTS
C                                        ACTING ON HELICOPTER
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL IR, LMR, LRH, MRH, LA, MA, NA, LF, MF, NF, MASS
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT         , NEQ         ,
     X                  ISCHEME, DERINT(2,12)
C
C                              SUBROUTINE  COMMUNICATION
C
      COMMON
     X        /ACCEL/   AX      , AY      , AZ
     X        /ADV C  / OMEG    , WIM     , VT      , QV      , QL      , ASPD
     X        /ENG C  / QMAX    , QE      , SHP     , GRWT    , OMEGDT
     X        /F N M P/ DX      , DXHS    , DXTR    , DXVS    , DZ      ,
     X                  DZTR    , DZVS    , DXW     , DZW     , IR
     X        /F N M C/ XA      , YA      , ZA      , LA      , MA      , NA
     X        /FUS C /  XF      , YF      , ZF      , LF      , MF      , NF      ,
     X                  XW      , WIWM    , ZW      , ALFF    , BETF    , ALFW
     X        /MROT C/  FXR     , FYR     , LMR     , LRH     , MRH     , QMR     ,
     X                  AOSS    , ALFSIY, UTSIY   , BETSI   , BDTSI   , A1SS    ,
     X                  AD1SS   , B1SS    , BD1SS
     X        /SHIP P/  IXX     , IYY     , IZZ     , MASS
     X        /TAIL C / ALFHS   , BTVS    , ZHS     , YVS     , VTR     ,
     X                  WHS     , YTR     , DELE
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 1),P     ),
     X            (DERINT(1, 2),Q     )
C
C                              SUM FORCES
C
      XA       = XF + XW  + FXR
      YA       = YF + YTR + FYR + YVS
      ZA       = ZF + ZW  + ZHS - LMR
C
C                              COMPUTE MOMENTS ABOUT C.G.
C
      LA       = LF + LRH + DZ*FYR + (DZ - DZTR)*YTR + (DZ - DZVS)*YVS
     X                    + IR*OMEG*Q
      MA       = MF + MRH - DZ*FXR - (DZ - DZW )*XW  + (DXHS - DX)*ZHS
     X                    + (DXW - DX)*ZW + DX*LMR - IR*OMEG*P
      NA       = NF + QE  + DX*FYR - (DXVS - DX)*YVS - (DXTR - DX)*YTR
C
C                              COMPUTE ACCELERATIONS
C
      AX       = XA/MASS
      AY       = YA/MASS
      AZ       =-ZA/MASS
C
      RETURN
      END
```

FUS

Subroutine FUS contains the equations and functions used in calculating the fuselage forces and moments. FUS also calculates the wing lift, drag forces, and downwash velocity (refs. 1 and 2 and appendix A).



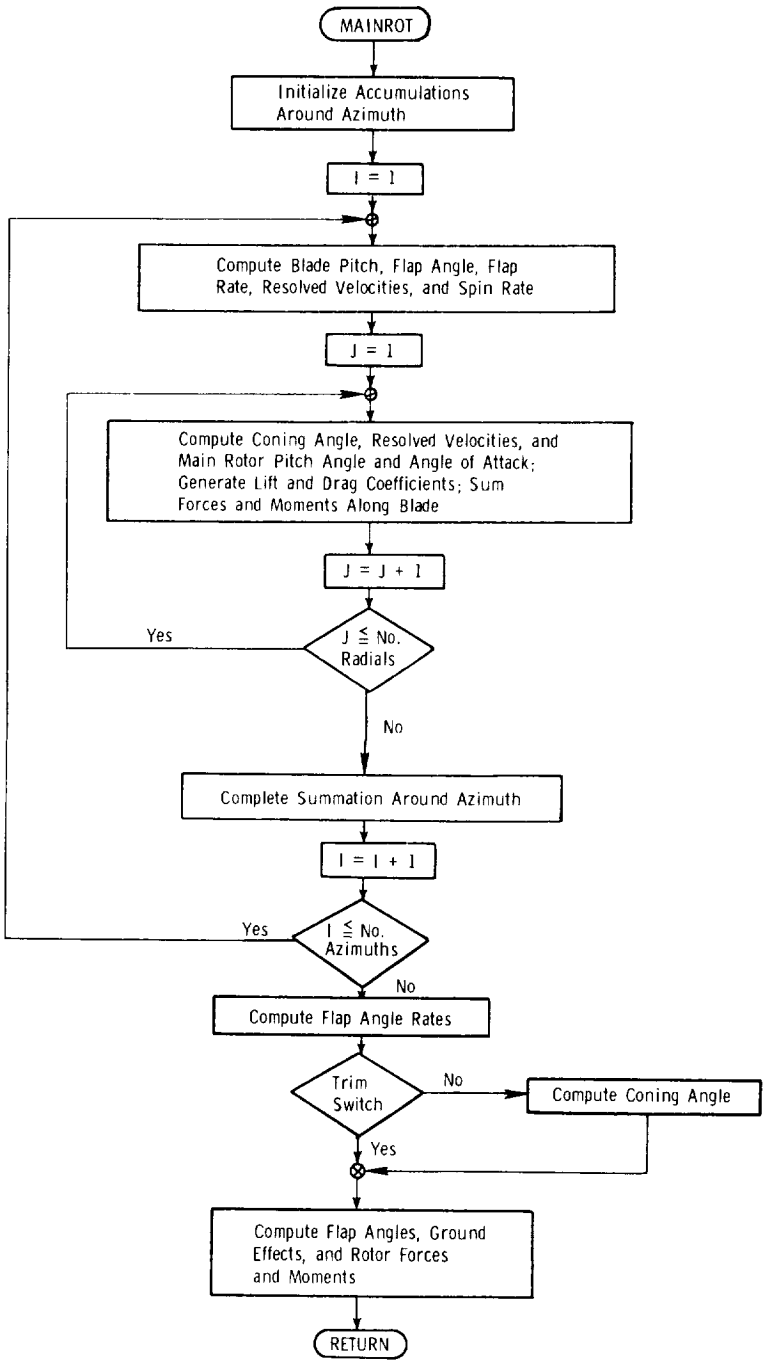
```
      SUBROUTINE FUS
C
C                              PROGRAM COMPUTES FORCES AND MOMENTS FOR
C                                   FUSELAGE AND WING COMBINATION
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL LF, MF, NF, IW
      LOGICAL LDISI, LDISO
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET    , NTERM     ,
     X                  NPRINT  , NREAD
C
C                              SUBROUTINE  COMMUNICATION
      COMMON
     X        /FCNNAME/ CLF1    , CMF1    , CNF1    , CX1     , CX2     , CYTR    ,
     X                  CYF     , CZF     , CXW     , CZW     , CYVS    , CZHS    ,
     X                  CDSIY   , CLSIY   , DW      , GEV     , GEH
     X        /FUNCS /  F001(28), F002(28), F003(14), F004(14), F005(14),
     X                  F006(28), F007(14), F008(28), F009(14), F010(28),
     X                  F011(35), F013(07), F014(07), F015(07), D018(27),
     X                  D001(09), D002(09), D003(09), D004(09), D005(09),
     X                  D006(09), D007(09), D008(09), D009(09), D010(09),
     X                  D011(09), D013(09), D014(09), D015(09), D016(18),
     X                  D017(18), F016(35,10),  F017(35,10),  F018(7,3,4)
C
      COMMON
     X        /ADV C  / OMEG    , WIM     , VT      , QV      , QL      , ASPD
     X        /FUS C /  XF      , YF      , ZF      , LF      , MF      , NF      ,
     X                  XW      , WIWM    , ZW      , ALFF    , BETF    , ALFW
     X        /FUS P /  CLF2    , CMF2    , CNF2    , CKRF    , CKWIWM, IW      ,
     X                  VLF     , VMF     , VNF     , SXF1    , SXF2    , SYF     ,
     X                  SZF     , SXW     , SZW
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 1),P     ),
     X            (DERINT(1, 2),Q     ),
     X            (DERINT(1, 3),R     ),
     X            (DERINT(1, 7),U     ),
     X            (DERINT(1, 8),V     ),
     X            (DERINT(1, 9),W     )
C
C                              COMPUTE SIDESLIP AND ANGLES OF ATTACK
C
      UL    = U
      IF(U.EQ.0.)   UL = .00001
      ALFF  = ATAN2(W-CKRF*WIM,UL)
      BETF  = ATAN2(V,UL)
      ALFW  = ALFF + IW
C
C                              WHITE LIGHTS 13,14,15 INDICATE ALFF, BETF,
C                                   ALFW RESPECTIVELY BEYOND LIMIT
C
      LDISO(73) = .F.
      LDISO(74) = .F.
      LDISO(75) = .F.
      IF(ALFF.LT.-3.12.OR.ALFF.GT.3.12)     LDISO(73) = .T.
      IF(BETF.LT.-3.12.OR.BETF.GT.3.12)     LDISO(74) = .T.
      IF(ALFW.LT.-3.12.OR.ALFW.GT.3.12)     LDISO(75) = .T.
C
C                              COMPUTE CMF1, CZF, CLF1, CNF1, CYF, CZW
C
      ALFFP = ALFF
      BETFP = BETF
      ALFWP = ALFW
      ALFSIGN =-SIGN(1.,ALFFP)
      BETSIGN =-SIGN(1.,BETFP)
      ALFWSGN =-SIGN(1.,ALFWP)
      ALFFP = ALFF
      BETFP =-ABS(BETF)
      ALFWP =-ABS(ALFW)
      IF(ALFFP.LE.-3.12)       ALFFP=-3.1199
      IF(ALFFP.GE. 3.12)       ALFFP= 3.1199
      CALL FUNC1(F001,D001,ALFFP)
      IF(BETFP.LE.-3.12)       BETFP=-3.1199
      IF(BETFP.GE.0.00 )       BETFP=-1.0E-10
      IF(ALFWP.LE.-3.12)       ALFWP=-3.1199
      IF(ALFWP.GE.0.00 )       ALFWP=-1.0E-10
      ALFFP =-ABS(ALFF)
      IF(ALFFP.LE.-3.12)   ALFFP=-3.1199
```

```
      IF(ALFFP.GE.0.0)      ALFFP=-1.0E-10
      CALL FUNC1(F003,D003,ALFFP)
      CALL FUNC1(F004,D004,BETFP)
      CALL FUNC1(F005,D005,BETFP)
      CALL FUNC1(F007,D007,BETFP)
      CALL FUNC1(F009,D009,ALFWP)
      CMF1   = D001(1)
      CZF    = D003(1)*ALFSIGN
      CLF1   = D004(1)*BETSIGN
      CNF1   = D005(1)*BETSIGN
      CYF    = D007(1)*BETSIGN
      CZW    = D009(1)*ALFWSGN
C
C                              COMPUTE CX2, CX1, CXW
C
      ALFFP = ABS(ALFF)
      BETFP = ABS(BETF)
      ALFWP = ABS(ALFW)
      IF(ALFFP.LE.0.000)    ALFFP=1.0E-10
      IF(ALFFP.GE.3.160)    ALFFP=3.1599
      IF(BETFP.LE.0.000)    BETFP=1.0E-10
      IF(BETFP.GE.3.160)    BETFP=3.1599
      IF(ALFWP.LE.0.000)    ALFWP=1.0E-10
      IF(ALFWP.GE.3.160)    ALFWP=3.1599
      CALL FUNC1(F002,D002,ALFFP)
      CALL FUNC1(F006,D006,BETFP)
      CALL FUNC1(F008,D008,ALFWP)
      CX2    = D002(1)
      CX1    = D006(1)
      CXW    = D008(1)
C
C                              CALCULATE FUSELAGE FORCES AND MOMENTS
C
      LF     = QL*VLF*CLF1  + U*P*CLF2
      MF     = QV*VMF*CMF1  + U*Q*CMF2
      NF     = QL*VNF*CNF1  + U*R*CNF2
      XF     = QL*SXF1*CX1  + QV*SXF2*CX2
      YF     = QL*SYF*CYF
      ZF     = QV*SZF*CZF
C
C                              CALCULATE WING FORCES AND WING DOWNWASH
C
      ZW     = QV*SZW*CZW
      XW     = QV*SXW*CXW + IW*ZW

      WIWM   = CKWIWM*U*CZW
C
      RETURN
      END
```

MAINROT

Subroutine MAINROT contains the equations and functions used to calculate the main rotor forces and moments generated due to a given collective and cyclic pitch and to given helicopter angular and translational rates. This subroutine includes the calculation of lift and drag coefficients, coning angle and flap angle rates, and total main rotor forces and moments including ground effect influences (refs. 1 and 2 and appendix A).



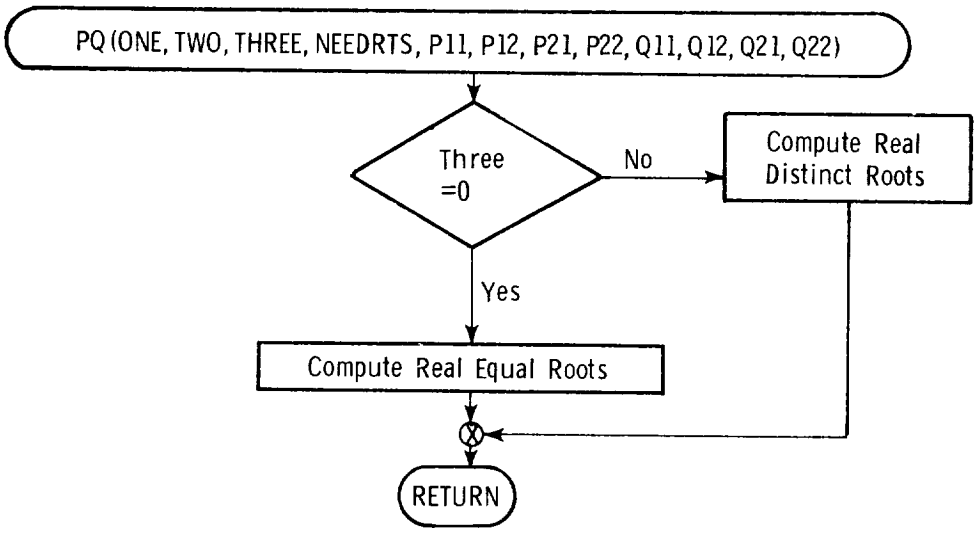
```
      SUBROUTINE MAINROT
C
C                              PROGRAM COMPUTES FORCES AND TORQUES ON
C                              MAIN ROTOR FOR GIVEN COLLECTIVE AND CYCLIC
C                              PITCHES,ANGULAR AND TRANSLATIONAL RATES.
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL LMR, LRH, MRH, LPDR, MPDR
      LOGICAL LDISI, LDISO
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER  , NHOLD  , NRESET    , NTERM     ,
     X                  NPRINT , NREAD
C
C                              SUBROUTINE  COMMUNICATION
      COMMON
     X        /FCNNAME/ CLF1    , CMF1    , CNF1    , CX1     , CX2     , CYTR    ,
     X                  CYF     , CZF     , CXW     , CZW     , CYVS    , CZHS    ,
     X                  CDSIY   , CLSIY   , DW      , GEV     , GEH
     X        /FUNCS /  F001(28), F002(28), F003(14), F004(14), F005(14),
     X                  F006(28), F007(14), F008(28), F009(14), F010(28),
     X                  F011(35), F013(07), F014(07), F015(07), D018(27),
     X                  D001(09), D002(09), D003(09), D004(09), D005(09),
     X                  D006(09), D007(09), D008(09), D009(09), D010(09),
     X                  D011(09), D013(09), D014(09), D015(09), D016(18),
     X                  D017(18), F016(35,10),  F017(35,10),  F018(7,3,4)
      COMMON
     X        /ADV C  / OMEG    , WIM     , VT      , QV      , QL      , ASPD
     X        /CONTRL/  AOS     , A1S     , B1S     , THTR    , OMEGD
     X        /FTRMPLS/ EQLMR   , EQABSS, EQWIM   , HDOTD
     X        /GPARAM/  DELT    , RHO     , G       , PIE
     X        /MROT C/  FXR     , FYR     , LMR     , LRH     , MRH     , QMR     ,
     X                  AOSS    , ALFSIY, UTSIY   , BETSI   , BDTSI   , A1SS    ,
     X                  ADISS   , B1SS    , BDISS
     X        /ROTCON / YPE(20), YTW(20),WIF(20), BOVN
     X        /ROT P /  TWIST   , FMRS    , CKA0    , CKAB1   , CKAB2   , CKAB3   ,
     X                  NRAD    , NAZ     , NB      , RB      , EFH
     X        /RCOEFF / CKL(20), CKD(20), CKQ(20), CKDL(20), CKQL(20),
     X                  CKM(20), Y(20)    , WF(20)
     X        /SCPSIR / SINSIR(50),  COSSIR(50)
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 1),P     ),
     X            (DERINT(1, 2),Q     ),
     X            (DERINT(1, 7),U     ),
     X            (DERINT(1, 8),V     ),
     X            (DERINT(1, 9),W     ),
     X            (DERINT(1,12),H     )
C
C
C
C                              INITIALIZE ACCUMULATIONS AROUND AZIMUTH
C
      SLDR  = 0.0
      SLSD  = 0.0
      SLCD  = 0.0
      SXDR  = 0.0
      SYDR  = 0.0
      SQDR  = 0.0
      SMDR  = 0.0
      SMSD  = 0.0
      SMCD  = 0.0
C
C                              MAIN LOOP AROUND AZIMUTH SUMS FORCES AND
C                              MOMENTS ON ROTOR.
C
      DO 20 I = 1,NAZ
C                              BLADE PITCH-EXCLUDING TWIST
C
      TEMT  = AOS  - A1S *COSSIR(I) - B1S *SINSIR(I) - (AOSS-.043)
C
C                              FLAP ANGLE
C
      BETSI = AOSS - A1SS*COSSIR(I) - B1SS*SINSIR(I)
C
C                              FLAP RATE
C
      BDTSI = (A1SS*SINSIR(I) - B1SS*COSSIR(I))*OMEG
C
```

```
C                              RESOLVED VELOCITIES AND SPIN RATE
C
            USVC   = U*SINSIR(I) + V*COSSIR(I)
            UCVS   = U*COSSIR(I) - V*SINSIR(I)
            QCPS   = Q*COSSIR(I) + P*SINSIR(I)
            WTEM   = W - BETSI*UCVS
C
C                              INITIALIZE SUMS OVER RADIAL STATIONS
C
            LPDR   = 0.0
            DPDR   = 0.0
            MPDR   = 0.0
            QPDR   = 0.0
C
C                              INNER LOOP SUMS FORCES ALONG BLADE
C
            DO  10 J = 1,NRAD
C
C                              CONING ANGLE AND RESOLVED VELOCITIES
C
                  THSIY = TEMT + YTW(J)
                  UTSIY = YPE(J)*OMEG + USVC
                  WISIY = WIM*WF(J)*(1. + UCVS*WIF(J))
                  UPSIY = YPE(J)*QCPS - Y(J)*BDTSI + WTEM-WISIY
C
C                              IF UTSIY NEGATIVE, ALFSIY, CL, CD ASSUMED
C                                    TO BE ZERO
C
                  IF (UTSIY .LE. 0.0)  GO TO 10
                  PHISIY= UPSIY/UTSIY
                  ALFSIY= THSIY + PHISIY
C
      ALFSIYP = ALFSIY
      UTSIYP  = UTSIY
      IF(ALFSIYP.LE.-.5060*)     ALFSIYP=-.50604
      IF(ALFSIYP.GE. .5060*)     ALFSIYP= .50604
      IF(UTSIYP.LE.110.)         UTSIYP=109.999
      IF(UTSIYP.GE.1100.)        UTSIYP=1099.999
C
C                              WHITE LIGHT 18 OR 19 ON INDICATES
C                            UTSIY OR ALFSIY RESPECTIVELY IS BEYOND
C                                    SET LIMITS
C
      LDISO(78) = .F.
      LDISO(79) = .F.
      IF(UTSIY .LT.110.000.OR.UTSIY .GT.1100.0)    LDISO(78) = .T.
      IF(ALFSIY.LT.-.50605.OR.ALFSIY.GT..50605)    LDISO(79) = .T.
C
C                              GENERATE LIFT AND DRAG COEFFICIENTS
C
      CALL FUNC2(F016,D016,ALFSIYP,UTSIYP)
      CALL FUNC2(F017,D017,ALFSIYP,UTSIYP)
      CDSIY    = D016(1)
      CLSIY    = D017(1)
C
C                              SUM FORCES AND MOMENTS ALONG BLADE
C
                  LPDR   = CKL(J)*CLSIY*UTSIY*UTSIY + LPDR
                  DPDR   = (CKD(J)*CDSIY*UTSIY - CKDL(J)*CLSIY*UPSIY)
     1                     *UTSIY + DPDR
                  MPDR   = CKM(J)*CLSIY*UTSIY*UTSIY    +  MPDR
                  QPDR   = (CKQ(J)*CDSIY*UTSIY - CKQL(J)*CLSIY*UPSIY)
     1                     *UTSIY + QPDR
   10       CONTINUE
C
C                              COMPLETE SUMMATION AROUND AZIMUTH
C
            SLDR   = LPDR + SLDR
            SLSD   = LPDR*SINSIR(I) + SLSD
            SLCD   = LPDR*COSSIR(I) + SLCD
            SMDR   = MPDR + SMDR
            SMSD   = MPDR*SINSIR(I) + SMSD
            SMCD   = MPDR*COSSIR(I) + SMCD
            SXDR   = BETSI*LPDR*COSSIR(I) - DPDR*SINSIR(I) + SXDR
            SYDR   =-BETSI*LPDR*SINSIR(I) - DPDR*COSSIR(I) + SYDR
            SQDR   = QPDR + SQDR
   20 CONTINUE
C
C                              COMPUTE CONING ANGLE AND FLAP ANGLE RATES
C
      ROVN   = RHO/NAZ
      RBON   = RHO*BOVN
C
      EQAOSS   = ROVN*SMDR*CKAO/(OMEG*OMEG) - AOSS
      ADISS =  CKAB1*SMSD*ROVN/OMEG - CKAB2*Q + CKAB3*B1SS
      BDISS = -CKAB1*SMCD*ROVN/OMEG - CKAB2*P - CKAB3*A1SS
      IF(LDISI(37)) GO TO 100
      AOSS     = ROVN*SMDR*CKAO/(OMEG*OMEG)
```

```
      IF(INT.LE.1)    A1SS = A1SS + DELT*AD1SS
      IF(INT.LE.1)    B1SS = B1SS + DELT*BD1SS
  100 CONTINUE
C
C
C                              COMPUTE ROTOR FORCES AND MOMENTS
C                              INCLUDING GROUND EFFECT INFLUENCE
C
      HP = H
      UP = U
      IF(HP.LE.0.00)  HP=.0001
      IF(HP.GE.50.0)  HP=49.999
      IF(UP.LE.0.00)  UP=.0001
      IF(UP.GE.200.)  UP=199.999
      CALL FUNC1(E015,D015,HP)
      CALL FUNC1(E014,D014,UP)
      GFH   = D015(1)
      GFV   = D014(1)
C
      LMR   = RBON*(1. + GFH*GEV)*SLDR
      EQWIM = WIM-.000328* LMR/(RHO*VT)
      FYR   = RBON*SYDR
      FXR   = RBON*SXDR
C
      LRH   =-EFH*SLSD*RBON + OMEG*OMEG*FMRS*B1SS
      MRH   =-EFH*SLCD*RBON + OMEG*OMEG*FMRS*A1SS
      QMR   = RBON*SQDR
C
      RETURN
      END
```

PQ

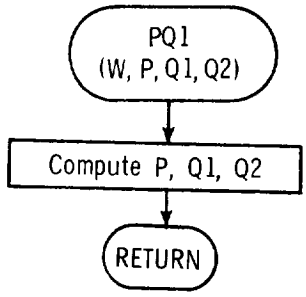
Subroutine PQ contains the equations used to calculate the constants needed for the convolution solution of a linear second-order equation. The subroutine requires the integration step size and the roots of the second-order system in order to calculate the constants (ref. 7).



```
      SUBROUTINE PQ(ONE,TWO,THREE,NEEDRTS,P11,P12,P21,P22,Q11,Q12,Q21,Q22)
C
C                                TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      LOGICAL  NEEDRTS   , LDISI       , LDISO
C
C                                INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT        , NEQ        ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196),
     X                  NOPER   , NHOLD   , NRESET     , NTERM      ,
     X                  NPRINT  , NREAD
C
      H       = DT
      SIGMA   = 0.
      SIGMA1  = 0.
      SIGMA2  = 0.
      IF(THREE.NE.0.0)   GO  TO 30
C
C                                REAL EQUAL ROOTS
C
   10 SIGMA   = ONE
   20 SH      = SIGMA*H
      IF(SIGMA.EQ.0)   GO TO 50
      PSH     = EXP(SH)
      SIGMAS  = SIGMA*SIGMA
      P12     = PSH*H
      P21     =-SIGMAS*P12
      P11     = PSH - SIGMA*P12
      P22     = PSH + SIGMA*P12
      Q12     = (H*(1. + PSH) - 2.*(PSH - 1.)/SIGMA)/SIGMAS
      Q21     = P12
      Q11     = (1. - P11)/SIGMAS
      Q22     = Q11
      GO TO 60
C
C                                REAL DISTINCT ROOTS
C
   30 SIGMA1  = ONE
      SIGMA2  = TWO
   40 S1H     = SIGMA1*H
      S2H     = SIGMA2*H
      PS1H    = EXP(S1H)
      PS2H    = EXP(S2H)
      S1S2    = SIGMA1*SIGMA2
      S1MS2   = SIGMA1-SIGMA2
      IF(S1S2.EQ.0..OR.S1MS2.EQ.0.)   GO TO  50
      P12     = (PS1H - PS2H)/S1MS2
      P21     =-S1S2*P12
      P11     = (SIGMA1*PS2H - SIGMA2*PS1H)/S1MS2
      P22     = (SIGMA1*PS1H - SIGMA2*PS2H)/S1MS2
      P1M1    = PS1H - 1.
      P2M1    = PS2H - 1.
      Q12     = (P1M1/SIGMA1 - H)/(S1MS2*SIGMA1) - (P2M1/SIGMA2 - H)/
     X          (S1MS2*SIGMA2)
      Q21     = P12
      Q11     = (P1M1/SIGMA1 - P2M1/SIGMA2)/S1MS2
      Q22     = Q11
      GO TO 60
   50 LDISO(90) = .T.
   60 CONTINUE
C
      RETURN
      END
```

PQ1

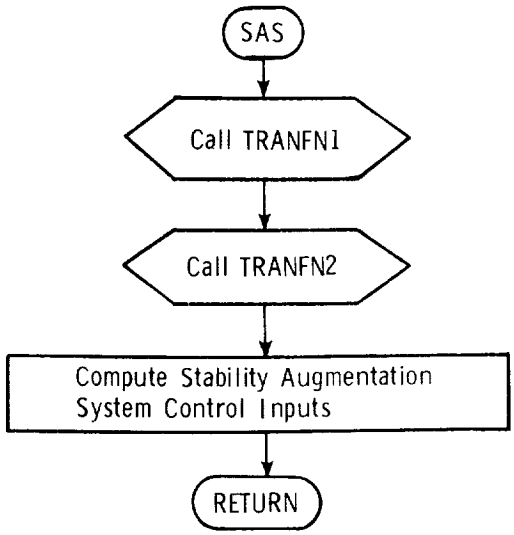
Subroutine PQ1 contains the equations used to calculate the constants needed for the convolution solution of a linear first-order equation. The subroutine requires the integration step size and the root of the first-order system in order to calculate the constants (ref. 7).




```
      SUBROUTINE PQ1(W,P,Q1,Q2)
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT         , NEQ
     X                  ISCHEME, DERINT(2,12)
C
      P  = EXP(-W*DT)
      Q1 = (1.-P)/W
      Q2 = (DT-Q1)/W
C
      RETURN
      END
```

SAS

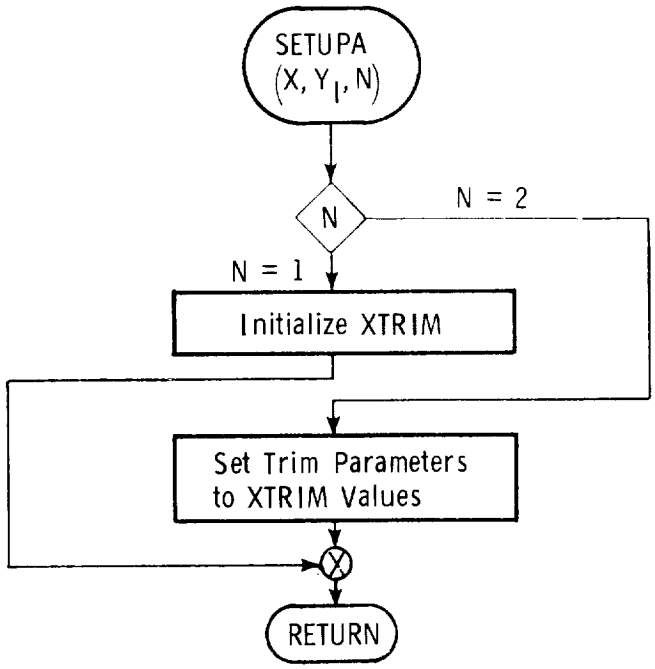
Subroutine SAS contains the equations used to evaluate the stability augmentation system transfer functions and to calculate the corrections made to the control inputs.



```
      SUBROUTINE SAS
C
C                              PROGRAM EVALUATES STABILITY AUGMENTATION
C                              SYSTEM TRANSFER FUNCTIONS AND CALCULATES
C                                   THE CONTROL SURFACE DEFLECTIONS
C
C
C                              INTEGRATION COMMUNICATION
      COMMON
     X       /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                 ISCHEME, DERINT(2,12)
C
C                              SUBROUTINE COMMUNICATION
C
      COMMON
     X       /ANGSAS / PHIL    , THETAL, PSIL    , PHIDL , THETDL, PSIDL
     X       /CNTSAS / A1SCS   , B1SCS , THTRCS
     X       /CNTL C / A0SC    , A1SC  , B1SC    , THTRC , A1SCL , B1SCL ,
     X                 THTRCL
     X       /SAS P  / A1C1    , A1C2  , A1K1    , A1K2  , A1K3  , A1K4  ,
     X                 B1C1    , B1C2  , B1K1    , B1K2  , B1K3  , B1K4  ,
     X                 THC1    , THC2  , THK1    , THK2  , THK3  , THK4
     X       /VALUES / A1CS1   , A1CS2 , B1CS1   , B1CS2 , THCS1 , THCS2
     X       /XIC    / A1X1    , A1X2  , A1X3    , B1X1  , B1X2  , B1X3  ,
     X                 THX1    , THX2  , THX3    , AX1TN , AX2TN , BX1TN ,
     X                 BX2TN   , TX1TN , TX2TN
C
C                              EQUIVALENCES
C
      EQUIVALENCE
     X            (DERINT(1, 4),PHI   ),
     X            (DERINT(1, 5),THETA)
     X            (DERINT(1, 6),PSI   )
C
C
      CALL TRANFN1(A1SC,A1SCL,A1C1,A1C2,A1CS1,AX1TN,AX2TN)
      CALL TRANFN2(PHI,PHIL,PHIDL,A1K1,A1K2,A1K3,A1K4,A1CS2,
     1             A1X1,A1X2,A1X3)
      A1SCS = A1CS1 - A1CS2
C
      CALL TRANFN1(B1SC,B1SCL,B1C1,B1C2,B1CS1,BX1TN,BX2TN)
      CALL TRANFN2(THETA,THETAL,THETDL,B1K1,B1K2,B1K3,B1K4,B1CS2,
     1             B1X1,B1X2,B1X3)
      B1SCS = B1CS1 + B1CS2
C
      CALL TRANFN1(THTRC,THTRCL,THC1,THC2,THCS1,TX1TN,TX2TN)
      CALL TRANFN2(PSI,PSIL,PSIDL,THK1,THK2,THK3,THK4,THCS2,
     1             THX1,THX2,THX3)
      THTRCS = THCS1 + THCS2
C
      RETURN
      END
```

SETUPA

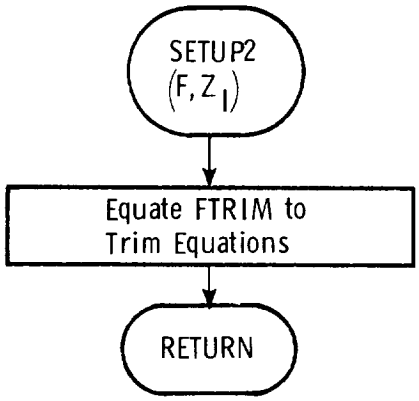
Subroutine SETUPA is used by the trim circuit algorithm and initializes XTRIM to the starting values of the trim parameters. During the trim iterations, the trim parameters are set to the XTRIM values (ref. 6 and appendix D).




```
      SUBROUTINE SETUPA (X,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,N)
C
C                              PROGRAM INITIALIZES XTRIM TO THE STARTING
C                              VALUES OF THE PARAMETERS. DURING THE TRIM
C                              ITERATION THE TRIM PARAMETERS ARE SET TO
C                                        THE XTRIM VALUES
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION X(11)
C
      GO TO(100,200),N
  100 CONTINUE
      X( 1) = Y1    $X( 2) = Y2    $X( 3) = Y3    $X( 4) = Y4
      X( 5) = Y5    $X( 6) = Y6    $X( 7) = Y7    $X( 8) = Y8
      X( 9) = Y9    $X(10) = Y10   $X(11) = Y11
      RETURN
C
  200 CONTINUE
      Y1 = X( 1)    $Y2 = X( 2)    $Y3 = X( 3)    $Y4 = X( 4)
      Y5 = X( 5)    $Y6 = X( 6)    $Y7 = X( 7)    $Y8 = X( 8)
      Y9 = X( 9)    $Y10= X(10)    $Y11= X(11)
C
      RETURN
      END
```

SETUP2

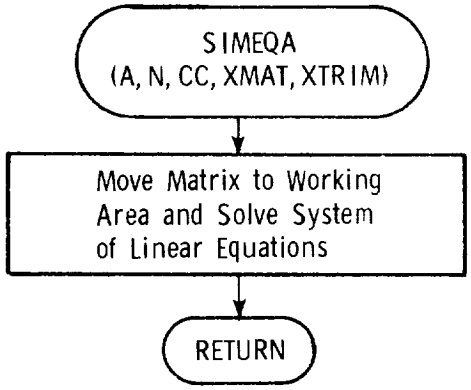
Subroutine SETUP2 is used by the trim circuit algorithm and equates FTRIM to the trim equations (ref. 6 and appendix D).




```
      SUBROUTINE SETUP2(F,Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9,Z10,Z11)
C
C                              PROGRAM EQUATES FTRIM TO THE TRIM EQUATIONS
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION F(11)
C
      F( 1) = Z1     $F( 2) = Z2     $F( 3) = Z3     $F( 4) = Z4
      F( 5) = Z5     $F( 6) = Z6     $F( 7) = Z7     $F( 8) = Z8
      F( 9) = Z9     $F(10) = Z10    $F(11) = Z11
C
      RETURN
      END
```

SIMEQA

Subroutine SIMEQA is used by the trim circuit algorithm and solves a system of linear equations by the Gauss-Jordan reduction method (ref. 6 and appendix D).

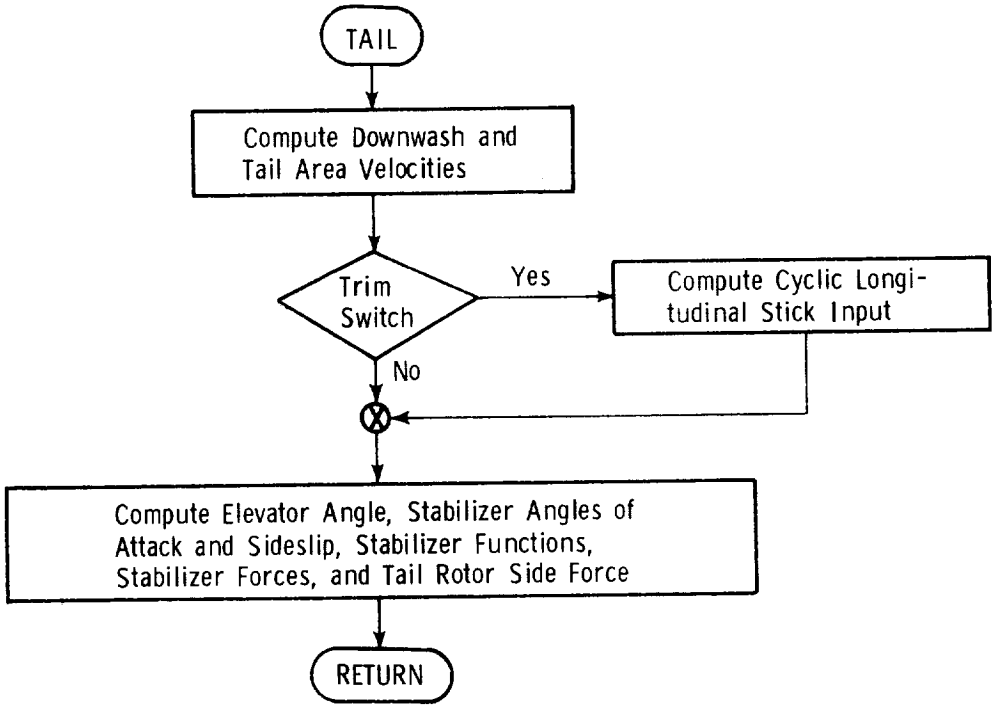


```
      SUBROUTINE SIMEQA(A,N,CC,XMAT,XTRIM)
C
C                              PROGRAM SOLVES A SYSTEM OF LINEAR EQUATIONS
C                              BY THE GUASS-JORDAN REDUCTION METHOD IN THE
C                                        TRIM ALGORITHM
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION A(12,13), CC(12), XMAT(11,12), XTRIM(11)
      DOUBLE PRECISION B(12,13), C(12), DXMAT(11,12), DXTRIM(11), D, TOL
C
      TOL      = 5.D-20
      KA       = N + 1
      K2       = 2
      K3       = 1
      DO 10 I=1,N
      DO 10 J=1,KA
C
C                              MOVE MATRIX TO WORKING AREA
   10 B(I,J)   = DBLE(A(I,J))
   20 IF(DABS(B(K3,K3))- TOL) 30, 100, 100
   30 IF(K2 - N)  40, 40, 300
   40 DO 50 I=1,KA
      D        = B(K3,I)
      B(K3,I) = B(K2,I)
      B(K2,I) = D
   50 CONTINUE
      K2       = K2 + 1
      GO TO 20
  100 K1       = K3
      K3       = K3 + 1
      K2       = K3 + 1
      IF(K3 - N) 120, 120, 200
  120 DO 130 I=K3,KA
      B(K1,I) = B(K1,I)/B(K1,K1)
  130 CONTINUE
      DO 140 I=K3,N
      DO 140 J=K3,KA
      B(I,J)   = B(I,J) - B(I,K1)*B(K1,J)
  140 CONTINUE
      GO TO 20
  200 C(N)     = B(N,N+1)/B(N,N)
      K3       = N - 1

  210 K1       = K3 + 1
      D        = 0.
      DO 220 I=K1,N
  220 D        = D + B(K3,I)*C(I)
      C(K3)    = B(K3,KA) - D
      K3       = K3 - 1
      IF(K3)  350, 350, 210
  300 TOL      = TOL/10.
      IF(TOL - 1.D-30) 320,310,310
  310 K2       = K3 + 1
      GO TO 20
  320 CONTINUE
  350 CONTINUE
      NM1      = N - 1
      DO 360 I=1,NM1
      DO 360 J=1,N
  360 DXMAT(I,J) = DBLE(XMAT(I,J))
      DO 370 I=1,NM1
      DXTRIM(I) = 0.0D0
      DO 370 J=1,N
      DXTRIM(I) = DXTRIM(I) + DXMAT(I,J)*C(J)
  370 CONTINUE
      DO 380 J=1,NM1
  380 XTRIM(J) = SNGL(DXTRIM(J))
      DO 390 J=1,N
  390 CC(J)    = SNGL(C(J))
C
      RETURN
      END
```

TAIL

Subroutine TAIL contains the equations and functions used to calculate the forces and moments generated due to the tail rotor and the horizontal and vertical stabilizers. General tail velocities are also computed to account for downwash velocities and angular rates (refs. 1 and 2 and appendix A).

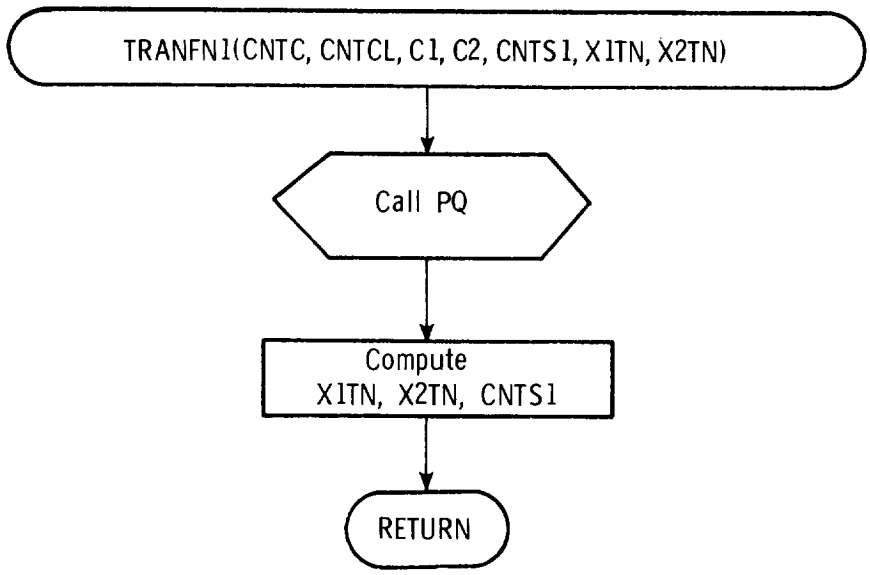


```
      SUBROUTINE TAIL
C
C                              PROGRAM COMPUTES FORCES AND MOMENTS DUE TO
C                              TAIL ROTOR, HORIZONTAL, AND VERTICAL STA-
C                              BILIZERS. GENERAL TAIL VELOCITIES ARE
C                                        ALSO COMPUTED
C
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL IHS, IVS
      LOGICAL LDIS1, LDISO
C
C                              INTEGRATION COMMUNICATION
C
       COMMON
     X        /INTCOMM/ T       , DT      , INT       , NEQ       ,
     X                  ISCHEME, DERINT(2,12)
     X        /REALTIM/ ADC(32), DAC(64), LDIS1(108), LDISO(196),
     X                  NOPER  , NHOLD  , NRESET    , NTERM     ,
     X                  NPRINT , NREAD
C
C                              SUBROUTINE  COMMUNICATION
       COMMON
     X        /FCNNAME/ CLF1   , CMF1   , CNF1    , CX1    , CX2    , CYTR   ,
     X                  CYF    , CZF    , CXW     , CZW    , CYVS   , CZ IS  ,
     X                  CDSIY  , CLSIY  , DW      , GEV    , GEH
     X        /FUNCS /  F001(28), F002(28), F003(14), F004(14), F005(14),
     X                  F006(28), F007(14), F008(28), F009(14), F010(28),
     X                  F011(35), F013(07), F014(07), F015(07), D018(27),
     X                  D001(09), D002(09), D003(09), D004(09), D005(09),
     X                  D006(09), D007(09), D008(09), D009(09), D010(09),
     X                  D011(09), D013(09), D014(09), D015(09), D016(18),
     X                  D017(18), F016(35,10),  F017(35,10),  F018(7,3,4)
       COMMON
     X        /ADV C  / OMEG   , WIM    , VT      , QV     , QL     , ASPD
     X        /CNTL P / AOSCO  , A1SCO  , B1SCO   , THTRCO , XAOSG , YCSG   ,
     X                  XCSG   , XTRG   , XAOSR   , YCSR   , XCSR   , XTRR
     X        /CONTRL/  AOS    , A1S    , B1S     , THTR   , OMEGD
     X        /FUS C /  XF     , YF     , ZF      , LF     , MF     , NF     ,
     X                  XW     , WIWM   , ZW      , ALFF   , BETF   , ALFW
     X        /F N M P/ DX     , DXHS   , DXTR    , DXVS   , DZ     ,
     X                  DZTR   , DZVS   , DXW     , DZW    , IR
     X        /STICKS / XAOS   , YCS    , XCS     , XTR    , DPR    , RPD
     X        /TAIL C / ALFHS  , BTVS   , ZHS     , YVS    , VTR    ,
     X                  WHS    , YTR    , DELE
     X        /TAIL P/  CKTR1  , CKTR2  , IHS     , IVS    , SZHS   , SYVS
C
C                              EQUIVALENCES
C
       EQUIVALENCE
     X            (DERINT(1, 2),Q     ),
     X            (DERINT(1, 3),R     ),
     X            (DERINT(1, 7),U     ),
     X            (DERINT(1, 8),V     ),
     X            (DERINT(1, 9),W     )
C
C                              COMPUTE DOWNWASH AS A FUNCTION OF
C                                       FORWARD VELOCITY
C
C
      UP    = U
      IF(UP.LE.0.00)    UP=0.0001
      IF(UP.GE.152.0)   UP=151.999
      CALL FUNC1(F013,D013,UP)
      DW    = D013(1)
C
C                              COMPUTE TAIL AREA VELOCITIES
C
      VTR     = V - (DXTR - DX)*R
      WHS     = W + (DXHS - DX)*Q - DW*WIM - WIWM
C
C                              COMPUTE STABILIZER ANGLES OF ATTACK
C                               (HOR. STAB. MODIFIED BY ELEVATOR)
C
      IF(LDIS1(37))   XCS = (B1S*DPR - B1SCO)/XCSG
      DELE  = (10.6852 - 2.0405*XCS + 0.2445*XCS*XCS)*RPD
      UL    = U
      IF(U.EQ.0.)   UL = .00001
      ALFHS = ATAN2(WHS,UL) + CKTR1*DELE + IHS
      BTVS  = ATAN2(VTR,UL) + IVS
C
C                              WHITE LIGHT 16 OR 17 ON INDICATES
C                             ALFHS OR BTVS RESPECTIVELY IS BEYOND
C                                        SET LIMITS
      LDISO(76) = .F.
      LDISO(77) = .F.
      IF(ALFHS.LT.-3.12.OR.ALFHS.GT.3.12)   LDISO(76) = .T.
```

```
      IF(BTVS.LT.-3.06.OR.BTVS.GT.3.06)     LDISO(77) = .T.
C
C                              COMPUTE CYVS, CZHS
C
      ALFHSP=ALFHS
      BTVSP = BTVS
      ALFSGN= SIGN(1.,ALFHSP)
      BTVSGN= SIGN(1.,BTVSP )
      ALFHSP= ABS(ALFHS)
      BTVSP = ABS(BTVS)
      IF(BTVSP.LE.0.00)      BTVSP=0.0001
      IF(BTVSP.GE.3.06)      BTVSP=3.0599
      IF(ALFHSP.LE.0.00)     ALFHSP=0.0001
      IF(ALFHSP.GE.3.12)     ALFHSP=3.1199
      CALL FUNC1(F010,D010,BTVSP)
      CALL FUNC1(F011,D011,ALFHSP)
      CYVS   = D010(1)*BTVSGN
      CZHS   = D011(1)*ALFSGN
C
C                              COMPUTE STABILIZER FORCES
C
      YVS    = QL*SYVS*CYVS
      ZHS    = QV*SZHS*CZHS
C
C                              WHITE LIGHT 20 ON INDICATES U, VTR, OR
C                              THTR BEYOND LIMITS WHEN COMPUTING CYTR
C
      LDISO(80) = .F.
      IF(VTR .LT.-20.0.OR.VTR .GT.20.0)     LDISO(80) = .T.
      IF(U   .LT. 00.0.OR.U   .GT.360.)     LDISO(80) = .T.
      IF(THTR.LT.-.353.OR.THTR.GT..432)     LDISO(80) = .T.
C
C                              COMPUTE TAIL ROTOR SIDE FORCE
C
C
      UP     = U
      VTRP   = VTR
      THTRP  = THTR
      IF(UP.LE.0.00)          UP=.0001
      IF(UP.GE.360.0)         UP=359.999
      IF(VTRP.LE.-20.)        VTRP=-19.999
      IF(VTRP.GE. 20.)        VTRP= 19.999
      IF(THTRP.LE.-.353)      THTRP=-.3529
      IF(THTRP.GE. .432)      THTRP= .4319
      CALL FUNC3(F018,D018,THTRP,VTRP,UP)
      CYTR   = D018(1)
      YTR    = CKTR2*CYTR
C
C
      RETURN
      END
```

TRANFN1

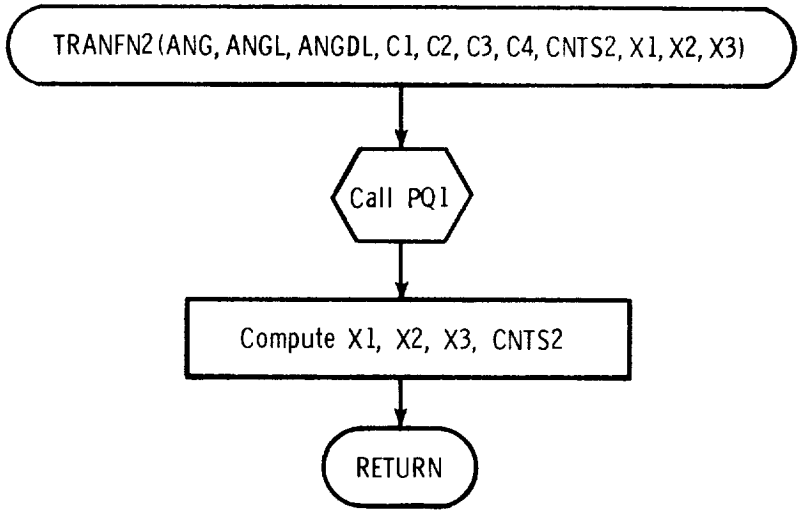
Subroutine TRANFN1 contains the equations used to compute the values of the first of two transfer functions which make up the total SAS transfer function.




```
      SUBROUTINE TRANFN1(CNTC,CNTCL,C1,C2,CNTS1,X1TN,X2TN)
C
C                              SUBROUTINE COMPUTES VALUES OF THE
C                                   FIRST TRANSFER FUNCTION
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      LOGICAL NEEDRTS
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X       /INTCOMM/ T       , DT      , INT         , NEQ         ,
     X                 ISCHEME, DERINT(2,12)
C
      CNTDL = (CNTC-CNTCL)/DT
      NEEDRTS = .F.
      RTYPE = 1.0
      ROOT1 = -0.4
      ROOT2 = -1./C2
      CALL PQ(ROOT1,ROOT2,RTYPE,NEEDRTS,P11,P12,P21,P22,Q11,Q12,Q21,Q22)
      X1TL = X1TN
      X2TL = X2TN
      X1TN = P11*X1TL + P12*X2TL + Q11*CNTCL + Q12*CNTDL
      X2TN = P21*X1TL + P22*X2TL + Q21*CNTCL + Q22*CNTDL
      CNTS1= C1/(2.5*C2)*X2TN
C
      RETURN
      END
```

TRANFN2

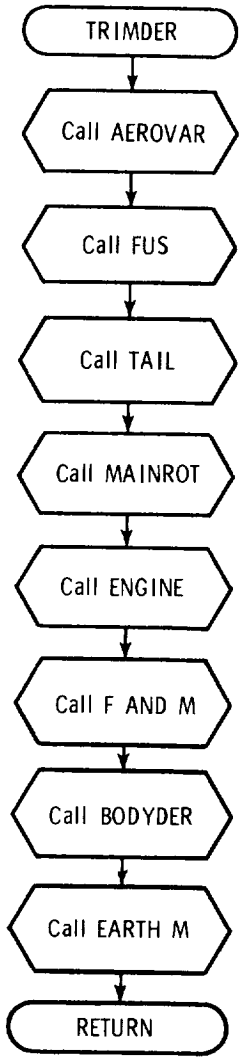
Subroutine TRANFN2 contains the equations used to compute the values of the second of two transfer functions which make up the total SAS transfer function.




```
      SUBROUTINE TRANFN2(ANG,ANGL,ANGDL,C1,C2,C3,C4,CNTS2,X1,X2,X3)
C
C                              SUBROUTINE COMPUTES VALUES OF THE
C                                   SECOND TRANSFER FUNCTION
C
C
C                              INTEGRATION COMMUNICATION
C
      COMMON
     X        /INTCOMM/ T       , DT      , INT         , NEQ         ,
     X                  ISCHEME, DERINT(2,12)
C
      XI(XX,XP,XQ1,XQ2) = XP*XX + XQ1*ANGL + XQ2*ANGDL
      YI(TC,XJ,AI)      = (ANG - 1./TC*XJ)*AI
      AP = 2.5
      A1 = (C1*(C2-AP))/(AP*(AP-C3)*(AP-C4))
      A2 = (C1*(C2-C3))/(C3*(C3-AP)*(C3-C4))
      A3 = (C1*(C2-C4))/(C4*(C4-C3)*(C4-AP))
C
      CALL PQ1(1./AP,P,Q1,Q2)
      X1 = XI(X1,P,Q1,Q2)
      CALL PQ1(1./C3,P,Q1,Q2)
      X2 = XI(X2,P,Q1,Q2)
      CALL PQ1(1./C4,P,Q1,Q2)
C
      X3 = XI(X3,P,Q1,Q2)
      Y1 = YI(AP,X1,A1)
      Y2 = YI(C3,X2,A2)
      Y3 = YI(C4,X3,A3)
      CNTS2 = Y1 + Y2 + Y3
C
      RETURN
      END
```

TRIMDER

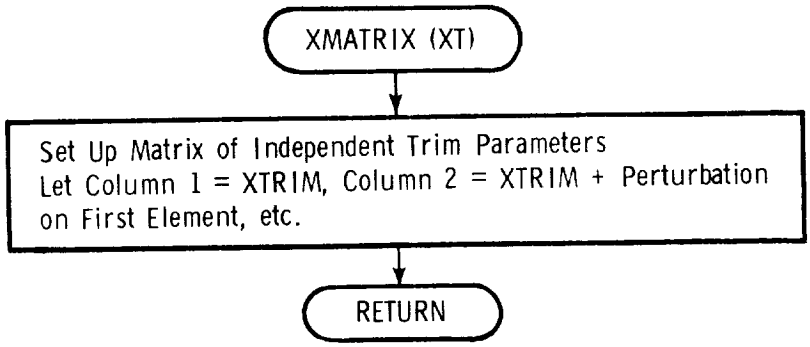
Subroutine TRIMDER is used by the trim circuit algorithm and executes all subroutines necessary to compute the trim equations (ref. 6 and appendix D).




```
      SUBROUTINE TRIMDER
C
C                                PROGRAM EXECUTES ALL SUBROUTINES NECESSARY
C                                       TO COMPUTE THE TRIM EQUATIONS
C
      CALL AEROVAR
      CALL FUS
      CALL TAIL
      CALL MAINROT
      CALL ENGINE
      CALL F AND M
      CALL BODYDER
      CALL EARTH M
C
      RETURN
      END
```

XMATRIX

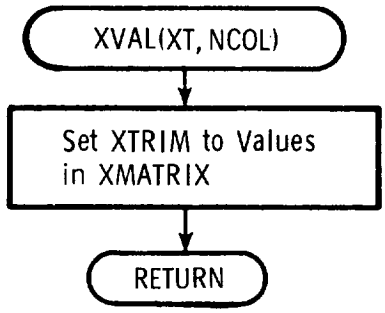
Subroutine XMATRIX is used by the trim circuit algorithm and sets up a matrix of independent trim parameters with the first column being XTRIM, the second column being XTRIM with a perturbation on the first element, etc. (ref. 6 and appendix D).




```
      SUBROUTINE XMATRIX(XT)
C
C                              PROGRAM SETS UP A MATRIX OF INDEPENDENT
C                              TRIM PARAMETERS WITH THE FIRST COLUMN
C                              BEING XTRIM, THE SECOND BEING XTRIM WITH
C                              A PERTURBATION ON THE FIRST ELEMENT,ETC.
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION XT(11)
C
C                              SUBROUTINE COMMUNICATION
C
      COMMON
     X        /TRIMAT/  XMAT(11,12), EMAT(12,13), TRSUM(12), NEQN,
     X                  NEQP1      , NEQP2       , XMATG(11)
C
      DO 100  I = 1,NEQN
      DO 100  J = 1,NEQP1
      XMAT(I,J) = XT(I)
  100 CONTINUE
      DO 200  J = 1,NEQN
      XMAT(J,J+1) = XMATG(J)*XT(J)
      IF(XT(J).EQ.0.0)   XMAT(J,J+1) = .001
  200 CONTINUE
C
      RETURN
      END
```

XVAL

Subroutine XVAL is used by the trim circuit algorithm and sets XTRIM to the values in XMATRIX to generate n + 1 equations where n = number of trim equations (ref. 6 and appendix D).




```
      SUBROUTINE XVAL(XT,NCOL)
C
C                              PROGRAM SETS XTRIM TO THE  VALUES IN
C                              XMATRIX TO GENERATE N + 1 EQUATIONS
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      DIMENSION XT(11)
C
C                              SUBROUTINE COMMUNICATION
C
      COMMON
     X        /TRIMAT/  XMAT(11,12), EMAT(12,13), TRSUM(12), NEQN,
     X                  NEQP1      , NEQP2      , XMATG(11)
C
      DO 100  J = 1,NEQN
      XT(J) = XMAT(J,NCOL)
  100 CONTINUE
C
      RETURN
      END
```

This section of the report describes the input data (BLOCK DATA and function data), the program setup, and the procedures for obtaining a static check, dynamic check, and cockpit checkout. Also included are console and cockpit running procedures.

Input Data Description

The function data are loaded into the program on data cards using a FORTRAN IV READ statement and a 7F10.0 format. All other input parameters are loaded in the FORTRAN IV BLOCK DATA subprogram. This subprogram assigns constant values to variables by use of the FORTRAN IV DATA statement. These values are then transferred to other subroutines through labeled COMMON. The BLOCK DATA input parameters are listed as follows with their appropriate COMMON label and typical value. Input parameters for function generation are also stored in the BLOCK DATA subprogram.

COMMON label	FORTRAN variable	Input value
/ADV P/	ADVP1	0.000328
	ADVP2	3.0
/CNTL P/	AOSC0	8.5
	A1SC0	-10.8
	B1SC0	-13.2
	THTRC0	23.0
	XAOSG	0.92
	YCSG	1.85
	XCSG	2.81
	XTRG	-5.00
	XAOSR	13.3

COMMON label	FORTRAN variable	Input value
/CNTL P/	YCSR	9.6
	XCSR	9.6
	XTRR	6.0
/ENG P/	IROT	2 820.0
	CKE1	10 529.0
/F N M P/	DX	-0.5125
	DXHS	16.55
	DXTR	26.75
	DXVS	25.08
	DZ	7.584
	DZTR	2.863
	DZVS	4.40
	DXW	-0.67
	DZW	7.554
	IR	0.0
/FUS P/	CLF2	-20.0
	CMF2	0.0
	CNF2	-30.0
	CKRF	0.50

COMMON label	FORTRAN variable	Input value
/FUS P/	CKWIWM	0.0
	IW	0.244
	VLF	100.0
	VMF	100.0
	VNF	100.0
	SXF1	10.0
	SXF2	10.0
	SYF	10.0
	SZF	10.0
	SXW	27.8
	SZW	27.8
/GPARAM/	DELT	0.03125
	RHO	0.00238
	G	32.2
	PIE	3.14159
/RCOEFF/	CKL(1)	5.32
	CKL(2)	4.95
	CKL(3)	4.35

COMMON label	FORTRAN variable	Input value
/RCOEFF/	CKD(1)	5.32
	CKD(2)	4.95
	CKD(3)	4.35
	CKQ(1)	52.68
	CKQ(2)	70.79
	CKQ(3)	92.57
	CKDL(1)	5.32
	CKDL(2)	4.95
	CKDL(3)	4.35
	CKQL(1)	52.68
	CKQL(2)	70.79
	CKQL(3)	92.57
	CKM(1)	52.68
	CKM(2)	70.79
	CKM(3)	92.57
	Y(1)	9.90
	Y(2)	14.30
	Y(3)	18.70

COMMON label	FORTRAN variable	Input value
/RCOEFF/	WF(1)	1.0
	WF(2)	1.0
	WF(3)	1.0
/ROT P/	TWIST	-0.175
	FMRS	0.0
	CKAO	0.000704
	CKAB1	0.000704
	CKAB2	1.0
	CKAB3	0.0
	NRAD	3
	NAZ	4
	NB	2
	RB	22.0
	EFH	0.0
/SAS P/	A1C1	
	A1C2	
	A1K1	
	A1K2	

COMMON label	FORTRAN variable	Input value
/SAS P/	A1K3	
	A1K4	
	B1C1	
	B1C2	
	B1K1	
	B1K2	
	B1K3	
	B1K4	
	THC1	
	THC2	
	THK1	
	THK2	
	THK3	
	THK4	
/SHIP P/	IXX	2 530.0
	IYY	11 716.0
	IZZ	10 164.0
	MASS	274.0

COMMON label	FORTRAN variable	Input value
/TAIL P/	CKTR1	1.0
	CKTR2	4 000.0
	IHS	0.0
	IVS	-0.0785
	SZHS	13.5
	SYVS	22.5

```
      BLOCK DATA
C
C                              BLOCK DATA SUBPROGRAM ASSIGNS VALUES
C                              FOR PARAMETERS IN LABELLED COMMON
C
C                              TYPE STATEMENTS AND DIMENSIONED VARIABLES
C
      REAL IR, IW, IHS, IVS, IROT, IXX, IYY, IZZ, MASS
C
C                              SUBROUTINE COMMUNICATION
C
C
      COMMON
     X        /ADV P /  ADVP1   , ADVP2
     X        /CNTL P / AOSC0   , A1SC0 , B1SC0  , THTRC0, XAOSG , YCSG  ,
     X                  XCSG    , XTRG  , XAOSR  , YCSR  , XCSR  , XTRR
     X        /ENG P /  IROT    , CKE1
     X        /F N M P/ DX      , DXHS  , DXTR   , DXVS  , DZ    ,
     X                  DZTR    , DZVS  , DXW    , DZW   , IR
     X        /FUS P /  CLF2    , CMF2  , CNF2   , CKRF  , CKWIWM, IW    ,
     X                  VLF     , VMF   , VNF    , SXF1  , SXF2  , SYF   ,
     X                  SZF     , SXW   , SZW
     X        /GPARAM/  DELT    , RHO   , G      , PIE
      COMMON
     X        /RCOEFF / CKL(20), CKD(20), CKQ(20), CKDL(20), CKQL(20),
     X                  CKM(20), Y(20)  , WF(20)
     X        /ROT P /  TWIST   , FMRS  , CKA0   , CKAB1 , CKAB2 , CKAB3 ,
     X                  NRAD    , NAZ   , NB     , RB    , EFH
     X        /SAS P  / A1C1    , A1C2  , A1K1   , A1K2  , A1K3  , A1K4  ,
     X                  B1C1    , B1C2  , B1K1   , B1K2  , B1K3  , B1K4  ,
     X                  THC1    , THC2  , THK1   , THK2  , THK3  , THK4
     X        /SHIP P/  IXX     , IYY   , IZZ    , MASS
     X        /TAIL P/  CKTR1   , CKTR2 , IHS    , IVS   , SZHS  , SYVS
C
      COMMON
     X        /FUNCS /  F001(28), F002(28), F003(14), F004(14), F005(14),
     X                  F006(28), F007(14), F008(28), F009(14), F010(28),
     X                  F011(35), F013(07), F014(07), F015(07), D018(27),
     X                  D001(09), D002(09), D003(09), D004(09), D005(09),
     X                  D006(09), D007(09), D008(09), D009(09), D010(09),
     X                  D011(09), D013(09), D014(09), D015(09), D016(18),
     X                  D017(18), F016(35,10),  F017(35,10),  F018(7,3,4)
C
C                                THE FOLLOWING DATA IS FOR COBRA
C                                DATA SOURCE IS TECHNICAL REPORT
C                                ECOM-0387-F2A
C
C
C
C     /ADVP   /--AERODYNAMIC PARAMETERS
      DATA
     1 ADVP1 / 3.28E-4  /, ADVP2 / 3.0      /
C
C
C     /CNTL P /--CONTROL STICK PARAMETERS
      DATA
     1 AOSC0 /  8.5     /, A1SC0 / -10.8    /, B1SC0 /  -13.2   /,
     2 THTRC0/  23.     /, XAOSG /  0.92    /, YCSG  /   1.85   /,
     3 XCSG  /  2.81    /, XTRG  / -5.00    /, XAOSR /   13.3   /,
     4 YCSR  /  9.6     /, XCSR  /  9.6     /, XTRR  /  6.0     /
C
C
C     / ENG P /--ENGINE PARAMETERS
      DATA
     1 IROT  / 2820.0   /, CKE1  / 10529.0  /
C
C
C     /F N M P/--PARAMETERS FOR CALCULATION OF FORCES AND MOMENTS
      DATA
     1  DX   /  -.5125  /,  DXHS /  16.55   /, DXTR  /  26.75   /,
     2  DXVS /   25.08  /,  DZ   /  7.584   /, DZTR  /  2.863   /,
     3  DZVS /    4.40  /,  DXW  /  -0.67   /, DZW   /  7.554   /,
     4  IR   /    0.00  /
C
C
C     / FUS P /--FUSELAGE PARAMETERS
      DATA
     1 CLF2  /   -20.0  /, CMF2  /   0.0    /, CNF2  /  -30.0   /,
     2 CKRF  /    0.50  /, CKWIWM/   0.0    /, IW    /   0.244  /,
     3 VLF   /   100.   /, VMF   /   100.   /, VNF   /   100.   /,
     4 SXF1  /    10.0  /, SXF2  /   10.0   /, SYF   /   10.0   /,
     5 SZF   /    10.0  /, SXW   /   27.8    /, SZW   /   27.8   /
C
C
C     /GPARAM /--GENERAL PARAMETERS
      DATA
     1 DELT  / 0.03125  /, RHO   / 0.00238  /, G     /   32.2   /,
     2 PIE   / 3.14159  /
```

```
C
C
C     /RCOFFF/--MAIN ROTOR ELEMENTAL LIFT AND DRAG COEFFICIENTS
C                DATA IS FOR 3 ELEMENT ROTOR
      DATA
     1 CKL(1)/ 5.32      /, CKL(2)/  4.95     /, CKL(3)/ 4.35      /,
     2 CKD(1)/ 5.32      /, CKD(2)/  4.95     /, CKD(3)/ 4.35      /,
     3 CKQ(1)/ 52.68     /, CKQ(2)/ 70.79     /, CKQ(3)/ 92.57     /,
     4CKDL(1)/ 5.32      /,CKDL(2)/  4.95     /,CKDL(3)/ 4.35      /,
     5CKQL(1)/ 52.68     /,CKQL(2)/ 70.79     /,CKQL(3)/ 92.57     /,
     6 CKM(1)/ 52.68     /, CKM(2)/ 70.79     /, CKM(3)/ 92.57     /,
     7    Y(1)/ 9.90     /,    Y(2)/ 14.30     /,    Y(3)/ 18.70    /,
     8  WF(1)/ 1.0       /,  WF(2)/   1.0     /,  WF(3)/ 1.0       /
C
C
C     / ROT P /--MAIN ROTOR PARAMETERS
      DATA
     1 TWIST /-0.175     /, FMRS  /0.0        /,
     2 CKA0  / 7.04E-4   /, CKAB1 / 7.04E-4   /, CKAB2 / 1.0       /,
     3 CKAB3 / 0.0       /, NRAD  / 3         /,  NAZ  / 4         /,
     4  NB   / 2         /,  RB   /  22.0     /,  EFH  / 0.0       /
C
C     /SAS P /--SAS GAINS AND TIME CONSTANTS
      DATA
     1 A1C1  /           /,  A1C2 /           /,  A1K1 /           /,
     2 A1K2  /           /,  A1K3 /           /,  A1K4 /           /,
     3 B1C1  /           /,  B1C2 /           /,  B1K1 /           /,
     4 B1K2  /           /,  B1K3 /           /,  B1K4 /           /,
     5 THC1  /           /,  THC2 /           /,  THK1 /           /,
     6 THK2  /           /,  THK3 /           /,  THK4 /           /
C
C
C     /SHIP P /--INERTIA AND MASS PARAMETERS
      DATA
     1  IXX  / 2530.0    /,  IYY  / 11716.0   /,  IZZ  / 10164.0   /,
     2  MASS / 274.0     /
C
C
C     /TAIL P /--TAIL ROTOR AND STABILIZER PARAMETERS
      DATA
     1 CKTR1 /    1.0    /, CKTR2 /  4000.    /, IHS   /   0.0     /,
     2 IVS   / -.0785    /, SZHS  /  13.5     /, SYVS  /  22.5     /
C
C
C
C                              INTERNAL DATA FOR FUNCTION GENERATION
C
C
      DATA (D001(IONE),IONE=2,9)/
     1-2.60000E-01, 3.12000E+00, 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D002(IONE),IONE=2,9)/
     1 7.84000E-01, 1.00000E+00,-0.         , 4.90000E-02, 2.67200E+00,
     1 2.16000E-01, 3.59000E+00, 2.70000E-01/
C
      DATA (D003(IONE),IONE=2,9)/
     1-2.60000E-01, 0.         , 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D004(IONE),IONE=2,9)/
     1-2.60000E-01, 0.         , 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D005(IONE),IONE=2,9)/
     1-2.60000E-01, 0.         , 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D006(IONE),IONE=2,9)/
     1 7.84000E-01, 1.00000E+00,-0.         , 4.90000E-02, 2.67200E+00,
     1 2.16000E-01, 3.59000E+00, 2.70000E-01/
C
      DATA (D007(IONE),IONE=2,9)/
     1-2.60000E-01, 0.         , 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D008(IONE),IONE=2,9)/
     1 7.84000E-01, 1.00000E+00,-0.         , 4.90000E-02, 2.67200E+00,
     1 2.16000E-01, 3.59000E+00, 2.70000E-01/
C
      DATA (D009(IONE),IONE=2,9)/
     1-2.60000E-01, 0.         , 3.12000E+00, 2.60000E-01, 3.12000E+00,
     1 2.60000E-01, 0.         , 0.         /
C
      DATA (D010(IONE),IONE=2,9)/
     1 1.60000E+00, 2.22000E+00,-0.         , 1.00000E-01, 8.32000E+00,
     1 6.20000E-01, 1.35000E+00, 2.10000E-01/
C
```

```
      DATA (D011(IONE),IONE=2,9)/
     1 4.71000E-01, 5.10000E-01,-0.          , 4.71000E-01,-4.32000E-01,
     1 3.90000E-02,-3.30000E-01, 9.00000E-02/
C
      DATA (D013(IONE),IONE=2,9)/
     1 1.40000E+01, 3.40000E+01,-0.          , 7.00000E+00, 2.60000E+01,
     1 2.00000E+01, 3.20000E+02, 1.18000E+02/
C
      DATA (D014(IONE),IONE=2,9)/
     1 3.30000E+01, 1.00000E+02,-0.          , 3.30000E+01, 3.40000E+01,
     1 6.70000E+01, 1.00000E+02, 1.00000E+02/
C
      DATA (D015(IONE),IONE=2,9)/
     1 1.00000E+01, 5.00000E+01,-0.          , 1.00000E+01, 1.00000E+01,
     1 2.00000E+01, 0.          , 0.          /
C
      DATA (D016(ITWO),ITWO=2,18)/
     1 5.06050E-01, 5.06050E-01, 5.06050E-01, 3.49000E-02, 0.          ,
     1 0.          , 0.          , 0.          , 3.50000E+01, 1.10000E+03,
     1 1.10000E+03,-1.1000nE+02, 1.10000E+02, 0.          , 0.          ,
     1 0.          , 0.          /
C
      DATA (D017(ITWO),ITWO=2,18)/
     1 5.06050E-01, 5.06050E-01, 5.06050E-01, 3.49000E-02, 0.          ,
     1 0.          , 0.          , 0.          , 3.50000E+01, 1.10000E+03,
     1 1.10000E+03,-1.1000nE+02, 1.10000E+02, 0.          , 0.          ,
     1 0.          , 0.          /
C
      DATA (D018(ITHREE),ITHREE=2,27)/
     1-9.20000E-02, 8.4000nE-02, 3.53000E-01, 2.61000E-01, 1.80000E-01,
     1 8.80000E-02, 9.6000nE-01, 3.48000E-01, 7.00000E+00, 2.00000E+01,
     1 2.00000E+01, 2.0000nE+01, 2.00000E+01, 0.          , 0.          ,
     1 0.          , 0.          , 3.00000E+00, 3.60000E+02, 3.60000E+02,
     1-0.          , 1.2000nE+02, 0.          , 0.          , 0.          ,
     1 0.          /
C
      END
```

```
-.020385  -.28538   -.54407   -.72308   -.79846   -.75846   -.71904
-.69404   -.66849   -.62618   -.42655   -.12774    .16600    .46371
 .73487    .85767    .96585    .96040    .95042    .84925    .74405
 .53809    .34481    .17769    .01269
-1.0400   -1.0540   -1.0680   -1.1010   -1.1371   -1.1836   -1.2479
-1.3121   -1.3180   -1.3158   -1.3135   -1.3113   -1.3090   -1.3068
-1.3045   -1.3023   -1.3000   -1.0809   -0.8188   -0.42778   0.8200
 1.0900    1.3200    1.3200    1.2007    1.0841
 .23077   3.2308    6.0000    6.0000    6.0000    6.0000    6.0000
6.0000    6.0000    6.0000    5.9793    4.1679     .00
  .071154   .99615   1.9352    3.0327    3.8529    3.9279    3.9971
 3.9221    3.8179    2.9781    1.8379     .79389    .00
-0.18846  -2.6385   -5.0296   -6.6927   -7.7777   -8.2377   -8.6650
-8.2750   -7.8627   -6.6058   -4.9015   -2.4313    .00
 0.0000   -.023333  -.046667  -.10439   -.16785   -.26436   -.41779
-.57121   -.59000   -.59000   -.59000   -.59000   -.59000   -.59000
-.59000   -.59000   -.59000   -.44234   -.23112   -.026556   .17508
 .37654    .59000    .59000    .34484    .09968
  .23846   3.3385    6.2593    7.0238    7.6349    8.0144    8.1927
 7.9973    7.6468    6.7574    5.0648    2.8779    .00
-.00500    .0066567  .018333   .044098   .072213   .12246    .21082
 .29918    .34845    .39229    .43612    .47995    .52378    .56761
 .61144    .65528    .69911    .61887    .36240    .036667  -.27692
-.58846   -.70000   -.60000   -.13508   .32984
  .05000    .70000   1.3186    1.5462    1.4442    1.2962    1.2038
 1.3058    1.4538    1.5452    1.2948    .84351    0.0
 0.0000   -.24333   -.48667   -.73000   -.79250   -.85500   -.91750
-.93000   -.93000   -.92676   -.89437   -.86197   -.82958   -.79718
-.76479   -.73239   -.70000   -.93000   -.93000   -.89000   -.51620
-.14240
 0.0000   -1.4100   -1.4064   -1.3980   -1.3896   -1.3812   -1.3728
-1.3540   -1.3300   -1.3060   -1.2820   -1.2580   -1.2340   -1.2100
-1.2291   -1.2482   -1.2674   -1.2865   -1.3056   -1.3247   -1.3439
-1.3630   -1.3762   -1.3422   -1.3081   -1.2741   -1.2400   -1.0026
-0.76511  -0.52766  -0.29021  -.052766
1.0       .7        .34       .16       0.
1.0       .4        0.        0.
1.0       .5        .125      0.
.622      .563      .508      .455      .406      .361      .130
.062      .013      .011      .010      .009      .008      .008
.008      .008      .008      .008      .009      .010      .011
.013      .062      .130      .361      .406      .455      .508
.563      .622
.622      .563      .508      .455      .406      .361      .130
.062      .013      .011      .010      .009      .008      .008
.008      .008      .008      .008      .009      .010      .011
.013      .062      .130      .361      .406      .455      .508
.563      .622
.622      .564      .508      .456      .406      .361      .130
.062      .013      .011      .010      .009      .008      .008
.008      .008      .008      .008      .009      .010      .011
.013      .062      .130      .361      .406      .456      .508
.564      .622
.623      .565      .509      .456      .407      .361      .180
.112      .044      .011      .010      .009      .008      .008
.008      .008      .008      .008      .009      .010      .011
.044      .112      .180      .361      .407      .456      .509
.565      .623
.626      .567      .511      .457      .407      .361      .280
.212      .144      .076      .010      .009      .008      .008
.008      .008      .008      .008      .009      .010      .076
.144      .212      .280      .361      .407      .457      .511
.567      .626
.632      .571      .514      .459      .409      .362      .340
.312      .244      .176      .108      .041      .008      .008
.008      .008      .008      .008      .041      .108      .176
.244      .312      .340      .362      .409      .459      .514
.571      .632
.640      .578      .519      .463      .411      .362      .340
.340      .340      .276      .208      .141      .074      .008
.008      .008      .008      .074      .141      .208      .276
.340      .340      .340      .362      .411      .463      .519
.578      .640
.653      .588      .527      .468      .414      .363      .340
.340      .340      .340      .308      .241      .174      .107
.041      .041      .107      .174      .241      .308      .340
.340      .340      .340      .363      .414      .468      .527
.588      .653
.672      .603      .538      .476      .418      .365      .340
.340      .340      .340      .340      .340      .274      .207
.141      .141      .207      .274      .340      .340      .340
.340      .340      .340      .365      .418      .476      .538
.603      .672
.699      .624      .554      .487      .425      .367      .340
.340      .340      .340      .340      .340      .340      .340
.340      .340      .340      .340      .340      .340      .340
.340      .340      .340      .367      .425      .487      .554
.624      .699
```

-.995 -.962 -.926 -.885 -.841 -.794 -.840
-.980 -1.120 -1.183 -.967 -.752 -.537 -.322
-.107 .107 .322 .537 .752 .967 1.183
1.120 .980 .840 .794 .841 .885 .926
.962 .995
-.995 -.963 -.926 -.885 -.841 -.794 -.822
-.928 -1.034 -1.140 -.982 -.764 -.546 -.327
-.109 .109 .327 .546 .764 .982 1.14
1.034 .928 .822 .794 .841 .885 .926
.963 .995
-.996 -.963 -.926 -.885 -.841 -.794 -.808
-.885 -.963 -1.04 -1.009 -.785 -.560 -.336
-.112 .112 .336 .560 .785 1.009 1.040
.963 .885 .808 .794 .841 .885 .926
.963 .996
-.997 -.964 -.927 -.886 -.842 -.794 -.797
-.851 -.904 -.958 -1.012 -.817 -.583 -.350
-.116 .116 .350 .583 .817 1.012 .958
.904 .851 .797 .794 .842 .886 .927
.964 .997
-1.000 -.966 -.929 -.887 -.842 -.794 -.787
-.822 -.857 -.892 -.927 -.864 -.617 -.370
-.123 .123 .370 .617 .864 .927 .892
.857 .822 .787 .794 .842 .887 .929
.966 1.000
-1.005 -.970 -.932 -.890 -.844 -.794 -.779
-.798 -.818 -.837 -.856 -.875 -.668 -.401
-.133 .133 .401 .668 .875 .856 .837
.818 .798 .779 .794 .844 .890 .932
.970 1.005
-1.013 -.977 -.937 -.893 -.846 -.795 -.773
-.779 -.785 -.792 -.798 -.804 -.749 -.449
-.149 .149 .449 .749 .804 .798 .792
.785 .779 .773 .795 .846 .893 .937
.977 1.013
-1.025 -.987 -.945 -.899 -.850 -.796 -.768
-.764 -.760 -.756 -.752 -.748 -.744 -.740
-.283 .283 .740 .744 .748 .752 .756
.760 .764 .768 .796 .850 .899 .945
.987 1.025
-1.042 -1.002 -.957 -.908 -.855 -.798 -.762
-.748 -.733 -.718 -.704 -.689 -.675 -.466
-.155 .155 .466 .675 .689 .704 .718
.733 .748 .762 .798 .855 .908 .957
1.002 1.042
-1.067 -1.023 -.974 -.920 -.862 -.801 -.755
-.727 -.699 -.670 -.642 -.614 -.500 -.300
-.100 .100 .300 .500 .614 .642 .670
.699 .727 .755 .801 .862 .920 .974
1.023 1.067
-.3253 -.0337 .0039 .0635 .4818
-.3538 -.0555 -.0026 .0477 .4556
-.2789 -.0720 -.0094 .0267 .4264
-.4091 -.0651 .0452 .1548 .6081
-.4561 -.1155 -.0056 .1045 .5627
-.5027 -.1658 -.0562 .0541 .5168
-.5328 -.0933 .0527 .1985 .7829
-.5919 -.1532 -.0074 .1386 .7241
-.6510 -.2131 -.0672 .0787 .6657
-.6802 -.1289 .0547 .2382 .9725
-.7436 -.1925 -.0092 .1744 .9091
-.8072 -.2566 -.0730 .1105 .8458
-

Program Setup

The program deck is read into the computer after selecting the RESET mode and both RELEASE switches on the program control console (fig. 1(b)). When real-time status is achieved, the READ mode is selected to store variables for printed output followed by depression of the lower RELEASE switch. The check cases are then run before further use of the helicopter simulation program.

The static check is initiated by depressing Function Sense Switch 6 (LDISI(38)), followed by selection of the READ mode and the lower RELEASE switch (fig. 1(b)). With an untrimmed (148.16 km/hr (80 knots)) condition selected, the simulation is momentarily placed in the HOLD mode and then in the OPERATE mode for approximately 1 second. Data are then printed by selecting the PRINT mode followed by depression of the lower RELEASE switch. The values at time equal zero (t = 0) are used for comparison with those listed in table I. With successful completion of the static check, the analyst is ready to conduct the dynamic check.

Before executing the dynamic check, the strip chart recorders should be set up as detailed on the overlay plots for each control perturbation selected. (See figs. 2, 3, and 4.) The dynamic response of the simulation is verified by imposing a doublet perturbation equivalent to a 2.54-centimeter (1 inch) stick deflection on each of three control inputs, that is, A_{1S_C}, B_{1S_C}, and θ_{TR_C}.

Function Sense Switch 1 (LDISI(33)) is depressed and the desired control deflection is input by entering the value in the appropriate TABLE location. The values equivalent to a 2.54-centimeter (1 inch) stick deflection are as follows:

A1SSTEP (TABLE(60)) = 0.032 radian for a perturbation on A_{1S_C}

B1SSTEP (TABLE(61)) = 0.049 radian for a perturbation on B_{1S_C}

THTRSTP (TABLE(62)) = 0.087 radian for a perturbation on θ_{TR_C}

The simulation is placed in the HOLD mode for a few seconds and then in the OPERATE mode for approximately 20 seconds for each of the three control perturbations. Output of the strip chart recorders can then be compared with the appropriate independent check overlay plots. (See figs. 2, 3, and 4.) Strip chart recorders are then reset for normal recording of data. With successful completion of the dynamic check, the analyst is ready to conduct the cockpit checkout. If no cockpit is desired, the analyst is ready to conduct test cases from the program control console.

Cockpit Checkout

After the static and dynamic checks have been executed, the simulation is now ready for the cockpit checkout. Telephone communication between the cockpit and the program control console is established and Function Sense Switches 12 and 13 (LDISI(44) and

LDISI(45)) are depressed to activate inputs to the computer from the cockpit. Function Sense Switch 10 (LDISI(42)) is depressed for several minutes to allow the cockpit engineer to statically check his instruments. During this time, voltages on all cockpit instruments (DAC(25)-DAC(40)) are checked on the digital decimal display unit for comparison with those listed in table II. When the cockpit engineer is ready, Function Sense Switch 10 is released and voltages for full deflections of the controls are checked on the digital decimal display unit for comparison with those listed in table III. After the cockpit checkout has been successfully completed, the simulation is ready for piloted test flights.

Console Run Procedure

Upon completion of the static and dynamic checks, the simulation program is ready for the analyst to set up preprogramed flights through the use of potentiometers representing the control sticks and pedals. The procedure is as follows:

Step 1: Select the RESET mode. All other switches should be in release positions.

Step 2: Select a value for KNOTS (INTEG(6)) from 1 to 6 corresponding to the preset conditions closest to the desired forward velocity in order to initialize the trim circuit.

Step 3: Input the desired forward velocity (TABLE(50)) and altitude (TABLE(52)).

Step 4: Activate the trim circuit by depressing Function Sense Switch 5 (LDISI(37)) until White Light 1 (LDISO(61)) illuminates. Then release Function Sense Switch 5.

Step 5: Set NSTICK0 (INTEG(1)) equal to 1; this allows control stick inputs from the potentiometers located on the program control console.

Step 6: Depress Function Sense Switch 13 (LDISI(45)) to activate subroutine ADCIN; this allows control stick inputs to the main program.

Step 7: Set potentiometers 1, 2, 3, and 4 until White Lights 21 through 24 (LDISO(81)-LDISO(84)) illuminate, respectively; this indicates that the controls are trimmed.

Step 8: Select one of the following variables and input the desired stick displacement in inches to program the desired flight condition:

X1 (TABLE(84)) – collective stick
X2 (TABLE(85)) – lateral stick
X3 (TABLE(86)) – longitudinal stick
X4 (TABLE(87)) – tail rotor pedals

Then set the time of input T1 (TABLE(88)) and the time of removal of input T2 (TABLE(89)).

Step 9: The SAS system may be activated, if desired, by depressing Function Sense Switch 9 (LDISI(41)).

Step 10: The simulation is now ready for the analyst to conduct a test run. The HOLD mode is depressed momentarily and is followed by the OPERATE mode for the desired number of seconds. At the completion of the run, the analyst depresses the RESET switch which returns the computer to the conditions at the beginning of the run. Desired printout may be obtained at this point by depressing the PRINT switch and the lower RELEASE switch. For sample strip chart output, see figure 5.

Cockpit Run Procedure

Upon completion of all check procedures, the analyst is ready to set up the initial conditions for the desired piloted flight cases. The procedure is as follows:

Step 1: Select the RESET mode. All other switches should be in release positions.

Step 2: Select a value for KNOTS (INTEG(6)) from 1 to 6 corresponding to the preset conditions closest to the desired forward velocity in order to initialize the trim circuit.

Step 3: Input the desired forward velocity (TABLE(50)) and altitude (TABLE(52)).

Step 4: Activate the trim circuit by depressing Function Sense Switch 5 (LDISI(37)) until White Light 1 (LDISO(61)) illuminates. Then release Function Sense Switch 5.

Step 5: Depress Function Sense Switch 13 (LDISI(45)) to activate subroutine ADCIN; this allows control stick inputs to the main program.

Step 6: Depress Function Sense Switch 12 (LDISI(44)) to allow cockpit control of the stick inputs.

Step 7: Pilot trims control sticks and pedals to program computed values. Cockpit trim is reached when White Lights 21 through 24 (LDISO(81)-LDISO(84)) are illuminated.

Step 8: Depress Function Sense Switch 11 (LDISI(43)) to activate cockpit control of computer control modes, that is, RESET, HOLD, OPERATE.

Step 9: The SAS system may be activated, if desired, by depressing Function Sense Switch 9 (LDISI(41)) or by having the pilot depress the SAS switch in the cockpit.

Step 10: The simulation is now ready for flight runs with the pilot depressing the HOLD switch and then the OPERATE switch. At the completion of a flight, the pilot depresses the RESET switch which returns the computer to the conditions at the beginning of the flight. Desired printout may be obtained at this point by depressing the PRINT switch and then the lower RELEASE switch on the program control console.

PROGRAM VALIDATION

This section presents a comparison of the Langley simulation program (C1152) with the simulation program of reference 1. Figure 6 presents data X_{AOS}, X_{CS}, Y_{CS}, X_{TR}, and θ from the Langley simulation program (C1152), the simulation program of reference 1, and an unpublished program based on reference 1 provided by the U.S. Army Electronics Command, Fort Monmouth, New Jersey. The good match between the Langley data and the reference 1 updated (unpublished) program data validates the implementation of the simulation program onto the Langley RTS system.

CONCLUDING REMARKS

A man-in-the-loop real-time helicopter simulation program has been developed and integrated with a suitable fixed-base cockpit by using the Langley Research Center real-time simulation system. With minor changes to the mathematical model and detailed aircraft data, this program can be used to simulate a variety of single-rotor helicopters. This fact has been substantiated by a program presently being used in support of two piloted studies of the Sikorsky S-61 helicopter, a commercial passenger-type helicopter. One study concerns pilot workload, route structures, and Instrument Flight Rules (IFR) procedures in congested areas. The second study concerns development and evaluation of computer-generated pilot displays and instruments for approach and landing. In both studies, the pilots' comments are favorable; that is, the simulation more than adequately represents a general single-rotor helicopter.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., February 27, 1974.

APPENDIX A

AIRCRAFT EQUATIONS

This appendix contains the equations used in program HELIC. The equations are listed according to the subroutines in which they are used along with references for their source.

Control equations:

Subroutine ADCIN (ref. 1)

$$A_{OS_C} = \left(8.5^{\circ} + 0.92X_{AOS}\right)RPD$$

$$A_{1S_C} = \left(-10.8^{\circ} + 1.85Y_{CS}\right)RPD$$

$$B_{1S_C} = \left(-13.2^{\circ} + 2.81X_{CS}\right)RPD$$

$$\theta_{TR_C} = \left(23.0^{\circ} - 5.0X_{TR}\right)RPD$$

where RPD is the factor for converting degrees to radians.

General aerodynamic variable equations:

Subroutine AEROVAR (refs. 1 and 2)

$$A_{SPD} = \frac{\sqrt{u^2 + v^2 + w^2}}{1.689}$$

$$V_T = \sqrt{u^2 + v^2 + \left(w - W_{IM}\right)^2}$$

$$ADVP_1 = \frac{1}{2\pi R_B{}^2}$$

$$ADVP_2 = \frac{1}{\tau}$$

$$W_{IM} = W_{IM} + \left(\frac{L_{MR}}{2\pi R_B{}^2 \rho V_T} - W_{IM}\right)\frac{\Delta t}{\tau}$$

$$\bar{q}_L = \frac{1}{2}\rho\left(u^2 + v^2\right)$$

$$\bar{q}_V = \frac{1}{2}\rho\left[u^2 + \left(w - W_{IM}\right)^2\right]$$

Body derivative equations:

Subroutine BODYDER (refs. 1 and 2)

$$\dot{p} = \frac{L_A + \left(I_{YY} - I_{ZZ}\right)qr}{I_{XX}}$$

$$\dot{q} = \frac{M_A + \left(I_{ZZ} - I_{XX}\right)rp}{I_{YY}}$$

$$\dot{r} = \frac{N_A + \left(I_{XX} - I_{YY}\right)pq}{I_{ZZ}}$$

$$\dot{\Psi} = \frac{r\cos\Phi + q\sin\Phi}{\cos\theta}$$

$$\dot{\theta} = q\cos\Phi - r\sin\Phi$$

$$\dot{\Phi} = p + \dot{\Psi}\sin\theta$$

$$\dot{u} = \frac{X_A}{m} + rv - qw - g\sin\theta$$

$$\dot{v} = \frac{Y_A}{m} + pw - ru + g\cos\theta\sin\Phi$$

$$\dot{w} = \frac{Z_A}{m} + qu - pv + g\cos\theta\cos\Phi$$

Earth orientation equations:

Subroutine EARTHM (refs. 1 and 2)

$$\dot{N} = \left[u\cos\theta + \left(v\sin\Phi + w\cos\Phi\right)\sin\theta\right]\cos\Psi - \left(v\cos\Phi - w\sin\Phi\right)\sin\Psi$$

$$\dot{E} = \left[u \cos \theta + (v \sin \Phi + w \cos \Phi)\sin \theta\right]\sin \Psi + (v \cos \Phi - w \sin \Phi)\cos \Psi$$

$$\dot{h} = u \sin \theta - (v \sin \Phi + w \cos \Phi)\cos \theta$$

Engine equations:

Subroutine ENGINE (refs. 1 and 8)

$$Q_{E_D} = K_{E_1}(\Omega_D - \Omega) + Q_{MR}$$

$$Q_E = Q_E + \left(Q_{E_D} - Q_E\right)\frac{\Delta t}{\tau}$$

$$Q_{max} = \frac{202.63158(h - 2500) + 412\ 500}{\Omega}$$

$$\dot{\Omega} = \frac{Q_E - Q_{MR}}{I_{ROT}}$$

$$\Omega = \Omega + \dot{\Omega}\Delta t$$

$$S_{HP} = \frac{Q_{MR}\Omega}{550}$$

$$C_T = \frac{G_{RWT}}{\rho\left(\pi R_B^{\,2}\right)\left(\Omega R_B\right)^2}$$

$$C_P = \frac{550 S_{HP}}{\rho\left(\pi R_B^{\,2}\right)\left(\Omega R_B\right)^3}$$

$$C_Q = \frac{Q_{MR}}{\rho R_B\left(\pi R_B^{\,2}\right)\left(\Omega R_B\right)^2}$$

$$\mu = \frac{1.689 A_{SPD}}{\Omega R_B}$$

$$M_{TIP} = \frac{1.689 A_{SPD} + \Omega R_B}{V_S}$$

APPENDIX A

Force and moment equations:

Subroutine FANDM (refs. 1 and 2)

$$X_A = X_F + X_W + F_{XR}$$

$$Y_A = Y_F + Y_{TR} + Y_{VS} + F_{YR}$$

$$Z_A = Z_F + Z_W + Z_{HS} - L_{MR}$$

$$L_A = L_F + L_{RH} + d_Z F_{YR} + \left(d_Z - d_{Z_{TR}}\right)Y_{TR} + \left(d_Z - d_{Z_{VS}}\right)Y_{VS} + I_R \Omega q$$

$$M_A = M_F + M_{RH} - d_Z F_{XR} + d_X L_{MR} - \left(d_Z - d_{Z_W}\right)X_W + \left(d_{X_W} - d_X\right)Z_W + \left(d_{X_{HS}} - d_X\right)Z_{HS} - I_R \Omega p$$

$$N_A = N_F + d_X F_{YR} - \left(d_{X_{TR}} - d_X\right)Y_{TR} - \left(d_{X_{VS}} - d_X\right)Y_{VS} + Q_E$$

$$a_X = \frac{X_A}{m}$$

$$a_Y = \frac{Y_A}{m}$$

$$a_Z = \frac{-Z_A}{m}$$

Fuselage equations:

Subroutine FUS (refs. 1 and 2)

$$\alpha_F = \tan^{-1}\left(\frac{w - K_{R/F} W_{IM}}{u}\right)$$

$$\beta_F = \tan^{-1}\left(\frac{v}{u}\right)$$

$$\alpha_W = \alpha_F + I_W$$

$$L_F = \bar{q}_L V l_F C l_{F_1} + upC l_{F_2}$$

$$M_F = \bar{q}_V V m_F C m_{F_1} + uqC m_{F_2}$$

$$N_F = \bar{q}_L V n_F C n_{F_1} + urC n_{F_2}$$

$$X_F = \bar{q}_L S X_{F_1} C X_1 + \bar{q}_V S X_{F_2} C X_2$$

$$Y_F = \bar{q}_L S Y_F C Y_F$$

$$Z_F = \bar{q}_V S Z_F C Z_F$$

$$Z_W = \bar{q}_V S Z_W C Z_W$$

$$X_W = \bar{q}_V S X_W C X_W + I_W Z_W$$

$$W_{IWM} = K W_{IWM} uC Z_W$$

Main rotor equations:

Subroutine MAINROT (refs. 1 and 2)

$$K_{A_O} = \frac{1}{I_B}$$

$$K_{AB_1} = \frac{1}{I_B}$$

$$K_{AB_2} = \frac{1 + m_B R_B e}{I_B}$$

$$K_{AB_3} = \frac{m_B R_B e \Omega}{2 I_B}$$

$$F_{MRS} = \frac{1}{2} N_B m_B R_B e$$

$$\beta_\Psi = \alpha_{OSS} - \alpha_{1SS} \cos \Psi_R - \beta_{1SS} \sin \Psi_R$$

$$\dot{\beta}_\Psi = \left(\alpha_{1SS} \sin \Psi_R - \beta_{1SS} \cos \Psi_R\right)\Omega$$

$$W_{I_{\Psi Y}} = W_{IM} W_{F_Y}\left[1.0 + W_{IF_Y}\left(u \cos \Psi_R - v \sin \Psi_R\right)\right]$$

$$U_{T_{\Psi Y}} = Y_{PE}\Omega + u \sin \Psi_R + v \cos \Psi_R$$

$$U_{P_{\Psi Y}} = Y_{PE}\left(q \cos \Psi_R + p \sin \Psi_R\right) - Y\dot{\beta}_\Psi + w - \beta_\Psi\left(u \cos \Psi_R - v \sin \Psi_R\right) - W_{I_{\Psi Y}}$$

$$\theta_{\Psi Y} = Y_{TW} + A_{OS} - A_{1S} \cos \Psi_R - B_{1S} \sin \Psi_R - \left(\alpha_{OSS} - 0.043\right)$$

$$\Phi_{\Psi Y} = \frac{U_{P_{\Psi Y}}}{U_{T_{\Psi Y}}}$$

$$\alpha_{\Psi Y} = \theta_{\Psi Y} + \Phi_{\Psi Y}$$

$$L_{PDR_{\Psi Y}} = K_{L_Y} C_{L_{\Psi Y}} U^2_{T_{\Psi Y}}$$

$$D_{PDR_{\Psi Y}} = K_{D_Y} C_{D_{\Psi Y}} U^2_{T_{\Psi Y}} - K_{DL_Y} C_{L_{\Psi Y}} U_{P_{\Psi Y}} U_{T_{\Psi Y}}$$

$$M_{PDR_{\Psi Y}} = K_{M_Y} C_{L_{\Psi Y}} U^2_{T_{\Psi Y}}$$

$$Q_{PDR_{\Psi Y}} = K_{Q_Y} C_{D_{\Psi Y}} U^2_{T_{\Psi Y}} - K_{QL_Y} C_{L_{\Psi Y}} U_{P_{\Psi Y}} U_{T_{\Psi Y}}$$

$$\alpha_{OSS} = \left(\frac{\rho}{N_{AZ}}\right)\left(\frac{K_{A_O}}{\Omega^2}\right) \sum^{\Psi_R} \sum^{Y} M_{PDR_{\Psi Y}}$$

$$\dot{\alpha}_{1SS} = \left(\frac{\rho}{N_{AZ}}\right)\left(\frac{K_{AB_1}}{\Omega}\right) \sum^{\Psi_R} \sum^{Y} \left(M_{PDR_{\Psi Y}} \sin \Psi_R\right) - K_{AB_2} q + K_{AB_3} \beta_{1SS}$$

$$\dot{\beta}_{1SS} = -\left(\frac{\rho}{N_{AZ}}\right)\left(\frac{K_{AB_1}}{\Omega}\right) \sum^{\Psi_R} \sum^{Y} \left(M_{PDR_{\Psi Y}} \cos \Psi_R\right) - K_{AB_2} p - K_{AB_3} \alpha_{1SS}$$

$$\alpha_{1SS} = \alpha_{1SS} + \dot{\alpha}_{1SS} \Delta t$$

$$\beta_{1SS} = \beta_{1SS} + \dot{\beta}_{1SS} \Delta t$$

$$F_{XR} = \left(\frac{\rho N_B}{N_{AZ}}\right) \sum^{\Psi_R} \sum^{Y} \left(\beta_\Psi L_{PDR_{\Psi Y}} \cos \Psi_R - D_{PDR_{\Psi Y}} \sin \Psi_R\right)$$

$$F_{YR} = \left(\frac{\rho N_B}{N_{AZ}}\right) \sum^{\Psi_R} \sum^{Y} \left(-\beta_\Psi L_{PDR_{\Psi Y}} \sin \Psi_R - D_{PDR_{\Psi Y}} \cos \Psi_R\right)$$

$$L_{RH} = -\left(\frac{e\rho N_B}{N_{AZ}}\right) \sum^{\Psi_R} \sum^{Y} \left(L_{PDR_{\Psi Y}} \sin \Psi_R\right) + F_{MRS}\Omega^2\beta_{1SS}$$

$$M_{RH} = -\left(\frac{e\rho N_B}{N_{AZ}}\right) \sum^{\Psi_R} \sum^{Y} \left(L_{PDR_{\Psi Y}} \cos \Psi_R\right) + F_{MRS}\Omega^2\alpha_{1SS}$$

$$L_{MR} = \left(\frac{\rho N_B}{N_{AZ}}\right)\left(1.0 + G_{EH}G_{EV}\right) \sum^{\Psi_R} \sum^{Y} L_{PDR_{\Psi Y}}$$

$$Q_{MR} = \left(\frac{\rho N_B}{N_{AZ}}\right) \sum^{\Psi_R} \sum^{Y} Q_{PDR_{\Psi Y}}$$

Tail equations:

Subroutine TAIL (refs. 1 and 2)

$$V_{TR} = v - \left(d_{X_{TR}} - d_X\right)r$$

$$W_{HS} = w + \left(d_{X_{HS}} - d_X\right)q - D_W W_{IM} - W_{IWM}$$

$$\delta_E = 10.6852^{\circ} - 2.0405X_{CS} + 0.2445X_{CS}^2$$

$$\alpha_{HS} = \tan^{-1}\left(\frac{W_{HS}}{u}\right) + K_{TR_1}\delta_E + I_{HS}$$

$$\beta_{VS} = \tan^{-1}\left(\frac{V_{TR}}{u}\right) + I_{VS}$$

$$Z_{HS} = \bar{q}_V S_{Z_{HS}} C_{Z_{HS}}$$

$$Y_{VS} = \bar{q}_L S_{Y_{VS}} C_{Y_{VS}}$$

$$Y_{TR} = K_{TR_2} C_{Y_{TR}}$$

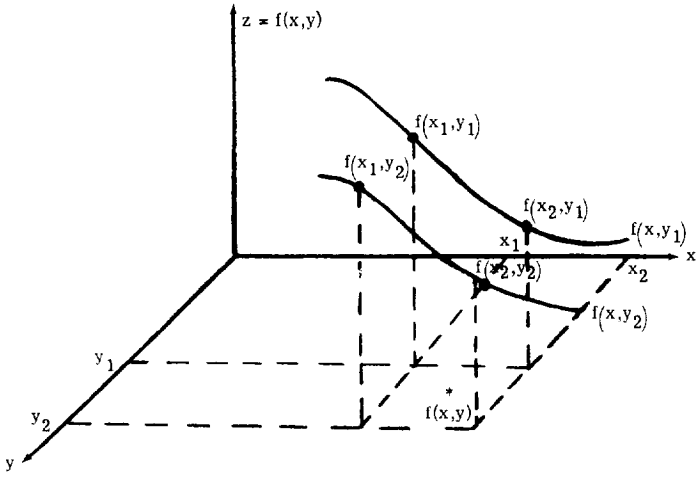
APPENDIX B

INDEXING TECHNIQUE FOR FUNCTION GENERATION

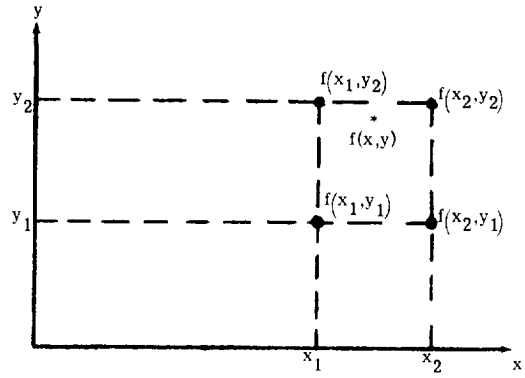
By Lawrence E. Barker, Jr., and Kemper S. Kibler
Langley Research Center

This appendix gives the mathematical and programing details of the technique used at Langley Research Center for the generation of aerodynamic functions for real-time simulation (ref. 5). The FORTRAN code for the subroutines is given followed by a description of the COMPASS version of this technique.

Since function generation represents the largest portion of the problem central processing unit (CPU) time in real-time digital programs with a large amount of aerodynamic data, prime attention was given to developing a technique to reduce this time. The mathematical description upon which this function method is based can be illustrated by the following sketches for an arbitrary function of two variables:



Sketch (a) – Three-dimensional view.



Sketch (b) – Bottom view.

Contour lines where one of the variables is held constant are drawn, such as $f(x,y_1)$ and $f(x,y_2)$. It is assumed that the values $f(x_1,y_1)$ corresponding to preselected grid points are stored in some array and that the area of interest is the area marked with an asterisk. The interpolated value of the function $f(x,y)$ is obtained by performing a linear interpolation between $f(x,y_1)$ and $f(x,y_2)$ where

$$f(x,y_1) = f(x_1,y_1) + \frac{x - x_1}{x_2 - x_1}\left[f(x_2,y_1) - f(x_1,y_1)\right] \tag{B1}$$

$$f(x,y_2) = f(x_1,y_2) + \frac{x - x_1}{x_2 - x_1}\left[f(x_2,y_2) - f(x_1,y_2)\right] \tag{B2}$$

This interpolation can then be written

$$f(x,y) = f(x,y_1) + \frac{y - y_1}{y_2 - y_1}\left[f(x,y_2) - f(x,y_1)\right] \tag{B3}$$

Equations (B1), (B2), and (B3) are sufficient to give the value for $f(x,y)$ in the particular area. For these equations to be of use they must be implemented to change as the independent variables move to a new area.

The method is based on a setup assuming the function for equal intervals of the independent variables. This will include most aerodynamic functions. A more generalized technique would be to provide for several areas of equal intervals per independent variable. The function technique described provides for three areas of equal intervals per independent variable. The areas and the corresponding interval for each area are defined by an array. The generated function value is returned through the first word in this array. A FORTRAN CALL STATEMENT is used to pass the array of function values, the array of function parameters, and the independent variables to the FUNC subroutine.

The key statement in the coding is the division in fixed point mode of the real FORTRAN variable AX (or AY or AZ) by its equal increment. This division allows selection of the proper interpolation area without a search routine. For this function generation technique the execution time is independent of the number of curves per function or the number of points per curve. The function value of interest will be obtained by linear interpolation between the stored data values by equations similar to equations (B1), (B2), and (B3). These equations are implemented in another form to be more efficient. The implemented FORTRAN equations for function of two variables are

$$G1 = TX*FUNC(I1,J1) + SX*FUNC(I2,J1) \tag{B4}$$

$$G2 = TX*FUNC(I1,J2) + SX*FUNC(I2,J2) \tag{B5}$$

$$RESULT = G1 + (AY/DELY - JS)*(G2 - G1) \tag{B6}$$

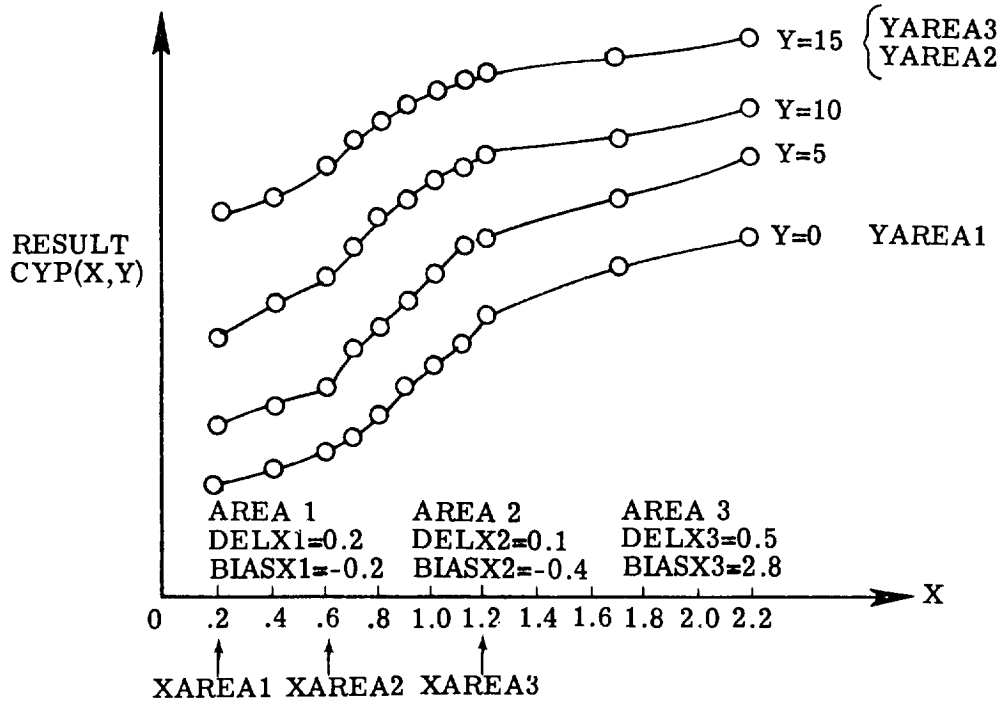
where

$$SX = AX/DELS - IS$$

and

$$TX = (1.0 - SX)$$

EXAMPLE: Consider the function of two variables defined in sketch (c). For this example, the independent variable X is divided into three areas. These areas are subdivided into equal segments.



Sketch (c) – Typical function of two variables.

The tabular form of the foregoing function is as follows:

X Y	0	5	10	15
0.2	FXY01(1,1)	FXY01(1,2)	FXY01(1,3)	FXY01(1,4)
.4	FXY01(2,1)			
.6				
.7				
.8				
.9				
1.0				
1.1				
1.2				
1.7				
2.2	FXY01(11,1)	FXY01(11,2)	FXY01(11,3)	FXY01(11,4)

XAREA1
DELX1 = 0.2
BIASX1 = -0.2

XAREA2
DELX2 = 0.1
BIASX2 = -0.4

XAREA3
DELX3 = 0.5
BIASX3 = 2.8

where

XAREA = (XAREA1,XAREA2,XAREA3) is the value of the independent variable X at the beginning of an area (XAREA point)

DELX is the desired value of increment which begins at the XAREA point

NUMPTS is the number of data points previous to the XAREA point

The BIAS(BIASX1,BIASX2,BIASX3) are functions of the independent variable X and the parameters DELX1, DELX2, and DELX3. The formula for determining the BIAS is

$$BIAS = (DELX)(NUMPTS) - XAREA$$

For the example shown,

$$BIASX1 = (0.2)(0) - 0.2 = -0.2$$

$$BIASX2 = (0.1)(2) - 0.6 = -0.4$$

$$BIASX3 = (0.5)(8) - 1.2 = 2.8$$

The BIAS for changes in DELY or DELZ are calculated in a similar manner. The NUMPTS in the BIAS formula would be replaced by the number of function curves previous to the YAREA point and the number of families previous to the ZAREA point, respectively.

For a two-variable function the parameter array (locations 2-18) is

PXY01(XAREA2,XAREA3,BIASX1,DELX1,BIASX2,DELX2,BIASX3,DELX3,XPOINTS,
YAREA2,YAREA3,BIASY1,DELY1,BIASY2,DELY2,BIASY3,DELY3)

Substituting numerical values gives

DATA(PXY01(ITWO),ITWO=2,18)/.6,1.2,-.2,.2,-.4,.1,2.8,.5,11.,15.,15.,0.,5.,0.,0.,0.,0.1

The FORTRAN CALL STATEMENT is

```
CALL FUNC2(FXY01,PXY01,X,Y)

CYP = PXY01(1)
```

APPENDIX B

```
C     *  ONE VARIABLE
C     *     THE FOLLOWING IS CODED IN COMPASS. THE FORTRAN
C     *     CODE IS SHOWN AS AN EXAMPLE OF WHAT IS CODED.
C
      IF(XINDEP-XAREA2)  1,1,2
    1 DELX = DELX1
      AX   = XINDEP + BIASX1
      GO TO 5
    2 IF(XINDEP-XAREA3)  3,3,4
    3 DELX = DELX2
      AX   = XINDEP + BIASX2
      GO TO 5
    4 DELX = DELX3
      AX   = XINDEP + BIASX3
    5 IS   = AX/DELX
      SX   = AX/DELX - IS
      RESULT = (1.-SX)*FUNC(IS+1) + SX*FUNC(IS+2)
```

```
C     *' TWO VARIABLE
C     *     THE FOLLOWING IS CODED IN COMPASS. THE FORTRAN
C     *     CODE IS SHOWN AS AN EXAMPLE OF WHAT IS CODED.
C
      IF(XINDEP-XAREA2)  1,1,2
    1 DELX = DELX1
      AX   = XINDEP + BIASX1
      GO TO 5
    2 IF(XINDEP-XAREA3)  3,3,4
    3 DELX = DELX2
      AX   = XINDEP + BIASX2
      GO TO 5
    4 DELX = DELX3
      AX   = XINDEP + BIASX3
    5 IF(YINDEP-YAREA2)  6,6,7
    6 DELY = DELY1
      AY   = YINDEP + BIASY1
      GO TO 13
    7 IF(YINDEP-YAREA3)  8,8,9
    8 DELY = DELY2
      AY   = YINDEP + BIASY2
      GO TO 13
    9 DELY = DELY3
      AY   = YINDEP + BIASY3
   13 IS   = AX/DELX
      JS   = AY/DELY
      I1   = IS + 1
      J1   = JS + 1
      I2   = IS + 2
      J2   = JS + 2
      SX   = AX/DELX - IS
      TX   = 1. - SX
      G1   = TX*FUNC(I1,J1) + SX*FUNC(I2,J1)
      G2   = TX*FUNC(I1,J2) + SX*FUNC(I2,J2)
      RESULT = G1 + (AY/DELY-JS)*(G2-G1)
```

```
C     *  THREE VARIABLE
C     *     THE FOLLOWING IS CODED IN COMPASS. THE FORTRAN
C     *     CODE IS SHOWN AS AN EXAMPLE OF WHAT IS CODED.
C
      IF(XINDEP-XAREA2)  1,1,2
    1 DELX = DELX1
      AX   = XINDEP + BIASX1
      GO TO 5
    2 IF(XINDEP-XAREA3)  3,3,4
    3 DELX = DELX2
      AX   = XINDEP + BIASX2
      GO TO 5
    4 DELX = DELX3
      AX   = XINDEP + BIASX3
    5 IF(YINDEP-YAREA2)  6,6,7
    6 DELY = DELY1
      AY   = YINDEP + BIASY1
      GO TO 13
    7 IF(YINDEP-YAREA3)  8,8,9
    8 DELY = DELY2
      AY   = YINDEP + BIASY2
      GO TO 13
    9 DELY = DELY3
      AY   = YINDEP + BIASY3
   13 IF(ZINDEP-ZAREA2)  14,14,15
   14 DELZ = DELZ1
      AZ   = ZINDEP + BIASZ1
      GO TO 18
   15 IF(ZINDEP-ZAREA3)  16,16,17
   16 DELZ = DELZ2
      AZ   = ZINDEP + BIASZ2
      GO TO 18
   17 DELZ = DELZ3
      AZ   = ZINDEP + BIASZ3
   18 IS   = AX/DELX
      JS   = AY/DELY
      KS   = AZ/DELZ
      I1   = IS + 1
      J1   = JS + 1
      K1   = KS + 1
      I2   = IS + 2
      J2   = JS + 2
      K2   = KS + 2
      SX   = AX/DELX - IS
      TX   = 1. - SX
      SY   = AY/DELY - JS
      G11  = TX*FUNC(I1,J1,K1) + SX*FUNC(I2,J1,K1)
      G21  = TX*FUNC(I1,J2,K1) + SX*FUNC(I2,J2,K1)
      G12  = TX*FUNC(I1,J1,K2) + SX*FUNC(I2,J1,K2)
      G22  = TX*FUNC(I1,J2,K2) + SX*FUNC(I2,J2,K2)
      H1   = G11 + SY*(G21-G11)
      H2   = G12 + SY*(G22-G12)
      RESULT  = H1 + (AZ/DELZ-KS)*(H2-H1)
```

APPENDIX B

SUBROUTINES FUNC1, FUNC2, FUNC3

Language: COMPASS

Purpose: To provide functions of one, two, and three variables. The method for generating these functions which is described in the reference is basic for an understanding of the material presented below.

Use: CALL FUNC1(FUNC,RESULT,XINDEP) for functions of one variable
CALL FUNC2(FUNC,RESULT,XINDEP,YINDEP) for functions of two variables
CALL FUNC3(FUNC,RESULT,XINDEP,YINDEP,ZINDEP) for functions of three variables

FUNC — the array which contains the function values.

RESULT — the generated function value which is returned through the first word of the array. The remaining locations in the array must contain the function parameters (see below).

XINDEP,YINDEP,ZINDEP — the X,Y,Z independent variables for the function.

For functions of one variable, the RESULT array must be dimensioned for nine locations and locations 2 through 9 must contain the following parameters (in order):

XAREA2,XAREA3,BIASX1,DELX1,BIASX2,DELX2,BIASX3,DELX3

For functions of two variables, the RESULT array must be dimensioned for 18 locations. Locations 2 through 9 are the same as those for the function of one variable. Locations 10 through 18 must contain the following parameters:

XPOINTS,YAREA2,YAREA3,BIASY1,DELY1,BIASY2,DELY2,BIASY3,DELY3

For functions of three variables, the RESULT array must be dimensioned for 27 locations. Locations 2 through 18 are the same as those for the function of two variables. Locations 19 through 27 contain the following parameters:

YCURVES,ZAREA2,ZAREA3,BIASZ1,DELZ1,BIASZ2,DELZ2,BIASZ3,DELZ3

where

AREA – that is, XAREA1,XAREA2,...,ZAREA2 are the values of the independent variables at which point the function values are tabulated at a different incremental value than the previous area.

DEL – that is, DELX1,DELX2,...,DELZ3 are the incremental values of the independent variable at which the functions are tabulated.

BIAS – that is, BIASX1,BIASX2,...,BIASZ3 are the values that when added on to the independent variable enable the function to be treated as an equal increment function.

XPOINTS – the number of points per function curve.

YCURVES – the number of function curves per family.

Restrictions: Limits should be placed on the X, Y, and Z upper and lower bounds of the function to avoid calling an address out of range.

Method: See reference.

Accuracy: Of the order of linear interpolation.

Reference: Indexing Technique for Function Generation. (See pp. 125-130 of appendix B.)

Storage: FUNC1 – 13_8 locations.
FUNC2 – 26_8 locations.
FUNC3 – 52_8 locations.

Subprograms used: None.

Other coding information: The timing requirements are as follows:
FUNC1 – 20 microseconds.
FUNC2 – 40 microseconds.
FUNC3 – 60 microseconds.

Source: See reference.

Author of subroutine: Dave E. Eckhardt, Jr.

Subroutine date: December 15, 1970.

APPENDIX C

FUNCTION DATA

This appendix contains the data input to the function generation scheme described in appendix B.

Angle* breakpoint	$C_{m_{F_1}}(\alpha_F)$	$C_{Z_F}(\alpha_F)$	$C_{l_{F_1}}(\beta_F)$	$C_{n_{F_1}}(\beta_F)$	$C_{Y_F}(\beta_F)$	$C_{Z_W}(\alpha_W)$
-3.12	-0.020385	0.23077	0.071154	-0.18846	0.23846	0.0500
-2.86	-.28538	3.2308	.99615	-2.6385	3.3385	.7000
-2.60	-.54407	6.0000	1.9352	-5.0296	6.2593	1.3185
-2.34	-.72308	6.0000	3.0327	-6.6927	7.0238	1.5462
-2.08	-.79846	6.0000	3.8529	-7.7777	7.6349	1.4442
-1.82	-.75846	6.0000	3.9279	-8.2377	8.0144	1.2962
-1.56	-.71904	6.0000	3.9971	-8.6650	8.1927	1.2038
-1.30	-.69404	6.0000	3.9221	-8.2750	7.9973	1.3058
-1.04	-.66849	6.0000	3.8179	-7.8627	7.6468	1.4538
-.78	-.62618	6.0000	2.9781	-6.6058	6.7674	1.5452
-.52	-.42655	5.9793	1.8379	-4.9015	5.0648	1.2948
-.26	-.12774	4.1679	.79389	-2.4313	2.8779	.84351
.00	.16600	.0000	.0000	.0000	.0000	.0000
.26	.46371	-4.1679	-.79389	2.4313	-2.8779	-.84351
.52	.73487	-5.9793	-1.8379	4.9015	-5.0648	-1.2948
.78	.85767	-6.0000	-2.9781	6.6058	-6.7674	-1.5452
1.04	.96585	-6.0000	-3.8179	7.8627	-7.6468	-1.4538
1.30	.96040	-6.0000	-3.9221	8.2750	-7.9973	-1.3058
1.56	.95042	-6.0000	-3.9971	8.6650	-8.1927	-1.2038
1.82	.84925	-6.0000	-3.9279	8.2377	-8.0144	-1.2962
2.08	.74405	-6.0000	-3.8529	7.7777	-7.6349	-1.4442
2.34	.53808	-6.0000	-3.0327	6.6927	-7.0238	-1.5462
2.60	.34481	-6.0000	-1.9352	5.0296	-6.2593	-1.3185
2.86	.17769	-3.2308	-.99615	2.6385	-3.3385	-.7000
3.12	.01269	-.23077	-.071154	.18846	-.23846	-.05000

*Angle refers to α_F, β_F, or α_W.

Angle* breakpoint	$C_{X_2}(\alpha_F)$	$C_{X_1}(\beta_F)$	$C_{X_W}(\alpha_W)$	Angle* breakpoint	$C_{X_2}(\alpha_F)$	$C_{X_1}(\beta_F)$	$C_{X_W}(\alpha_W)$
-3.160	1.0841	0.09968	0.32984	0.000	-1.0400	0.00000	-0.0050000
-2.890	1.2007	.34484	-.13508	.049	-1.0540	-.023333	.0066667
-2.620	1.3200	.59000	-.60000	.098	-1.0680	-.046667	.018333
-2.350	1.3200	.59000	-.70000	.147	-1.1010	-.10439	.044098
-2.080	1.0900	.37654	-.58846	.196	-1.1371	-.16785	.072213
-1.810	.8200	.17508	-.27692	.245	-1.1836	-.26436	.12246
-1.540	-.42778	-.026556	.036667	.294	-1.2479	-.41779	.21082
-1.270	-.8188	-.23112	.36240	.343	-1.3121	-.57121	.29918
-1.000	-1.0809	-.44234	.61887	.392	-1.3180	-.59000	.34846
-.784	-1.3000	-.59000	.69911	.441	-1.3158	-.59000	.39229
-.735	-1.3023	-.59000	.65528	.490	-1.3135	-.59000	.43612
-.686	-1.3045	-.59000	.61144	.539	-1.3113	-.59000	.47995
-.637	-1.3068	-.59000	.56761	.588	-1.3090	-.59000	.52378
-.588	-1.3090	-.59000	.52378	.637	-1.3068	-.59000	.56761
-.539	-1.3113	-.59000	.47995	.686	-1.3045	-.59000	.61144
-.490	-1.3135	-.59000	.43612	.735	-1.3023	-.59000	.65528
-.441	-1.3158	-.59000	.39229	.784	-1.3000	-.59000	.69911
-.392	-1.3180	-.59000	.34846	1.000	-1.0809	-.44234	.61887
-.343	-1.3121	-.57121	.29918	1.270	-.8188	-.23112	.36240
-.294	-1.2479	-.41779	.21082	1.540	-.42778	-.026556	.036667
-.245	-1.1836	-.26436	.12246	1.810	.8200	.17508	-.27692
-.196	-1.1371	-.16785	.072213	2.080	1.0900	.37654	-.58846
-.147	-1.1010	-.10439	.044098	2.350	1.3200	.59000	-.70000
-.098	-1.0680	-.046667	.018333	2.620	1.3200	.59000	-.60000
-.049	-1.0540	-.023333	.0066667	2.890	1.2007	.34484	-.13508
				3.160	1.0841	.09968	.32984

*Angle refers to α_F, β_F, or α_W.

β_{VS} breakpoint	$C_{Y_{VS}}(\beta_{VS})$	β_{VS} breakpoint	$C_{Y_{VS}}(\beta_{VS})$
-3.06	0.14240	0.10	-0.24333
-2.85	.51620	.20	-.48667
-2.64	.89000	.30	-.73000
-2.43	.93000	.40	-.79250
-2.22	.93000	.50	-.85500
-1.60	.70000	.60	-.91750
-1.50	.73239	.70	-.93000
-1.40	.76479	.80	-.93000
-1.30	.79718	.90	-.92676
-1.20	.82958	1.00	-.89437
-1.10	.86197	1.10	-.86197
-1.00	.89437	1.20	-.82958
-.90	.92676	1.30	-.79718
-.80	.93000	1.40	-.76479
-.70	.93000	1.50	-.73239
-.60	.91750	1.60	-.70000
-.50	.85500	2.22	-.93000
-.40	.79250	2.43	-.93000
-.30	.73000	2.64	-.89000
-.20	.48667	2.85	-.51620
-.10	.24333	3.06	-.14240
.00	.00000		

α_{HS} breakpoint	$C_{Z_{HS}}(\alpha_{HS})$	α_{HS} breakpoint	$C_{Z_{HS}}(\alpha_{HS})$
-3.12	0.052766	0.471	-1.4100
-3.03	.29021	.51	-1.4064
-2.94	.52766	.60	-1.3980
-2.85	.76511	.69	-1.3896
-2.76	1.0026	.78	-1.3812
-2.67	1.2400	.87	-1.3728
-2.58	1.2741	.96	-1.3540
-2.49	1.3081	1.05	-1.3300
-2.40	1.3422	1.14	-1.3060
-2.31	1.3762	1.23	-1.2820
-2.22	1.3630	1.32	-1.2580
-2.13	1.3439	1.41	-1.2340
-2.04	1.3247	1.50	-1.2100
-1.95	1.3056	1.59	-1.2291
-1.86	1.2865	1.68	-1.2482
-1.77	1.2674	1.77	-1.2674
-1.68	1.2482	1.86	-1.2865
-1.59	1.2291	1.95	-1.3056
-1.50	1.2100	2.04	-1.3247
-1.41	1.2340	2.13	-1.3439
-1.32	1.2580	2.22	-1.3630
-1.23	1.2820	2.31	-1.3762
-1.14	1.3060	2.40	-1.3422
-1.05	1.3300	2.49	-1.3081
-.96	1.3540	2.58	-1.2741
-.87	1.3728	2.67	-1.2400
-.78	1.3812	2.76	-1.0026
-.69	1.3896	2.85	-.76511
-.60	1.3980	2.94	-.52766
-.51	1.4064	3.03	-.29021
-.471	1.4100	3.12	-.052766
.00	.0000		

APPENDIX C

u breakpoint	$D_W(u)$
0	1.0
7	.7
14	.34
34	.16
152	.0

u breakpoint	$G_{EV}(u)$
0	1.0
33	.4
100	.0
200	.0

h breakpoint	$G_{EH}(h)$
0	1.0
10	.5
30	.125
50	.0

u breakpoint	V_{TR} breakpoint	$C_{Y_{TR}} = f(u, V_{TR}, \theta_{TR})$ for –				
		θ_{TR} = -0.353	θ_{TR} = -0.092	θ_{TR} = -0.004	θ_{TR} = 0.084	θ_{TR} = 0.432
0	-20	-0.3253	-0.0337	0.0039	0.0635	0.4818
	0	-.3538	-.0555	-.0026	.0477	.4556
	20	-.3789	-.0720	-.0094	.0267	.4264
120	-20	-.4091	-.0651	.0452	.1548	.6081
	0	-.4561	-.1155	-.0056	.1045	.5627
	20	-.5027	-.1658	-.0562	.0541	.5168
240	-20	-.5328	-.0933	.0527	.1985	.7829
	0	-.5919	-.1532	-.0074	.1386	.7241
	20	-.6510	-.2131	-.0672	.0787	.6657
360	-20	-.6802	-.1289	.0547	.2382	.9725
	0	-.7436	-.1925	-.0092	.1744	.9091
	20	-.8072	-.2566	-.0730	.1105	.8458

$\alpha_{\Psi Y}$ breakpoint	$C_{D_{\Psi Y}} = f(\alpha_{\Psi Y}, U_{T_{\Psi Y}})$ for –									
	$U_{T_{\Psi Y}} = 110$	$U_{T_{\Psi Y}} = 220$	$U_{T_{\Psi Y}} = 330$	$U_{T_{\Psi Y}} = 440$	$U_{T_{\Psi Y}} = 550$	$U_{T_{\Psi Y}} = 660$	$U_{T_{\Psi Y}} = 770$	$U_{T_{\Psi Y}} = 880$	$U_{T_{\Psi Y}} = 990$	$U_{T_{\Psi Y}} = 1100$
-0.50605	0.622	0.622	0.622	0.623	0.626	0.632	0.640	0.653	0.672	0.699
-.47115	.563	.563	.564	.565	.567	.571	.578	.588	.603	.624
-.43625	.508	.508	.508	.509	.511	.514	.519	.527	.538	.554
-.40135	.455	.455	.456	.456	.457	.459	.463	.468	.476	.487
-.36645	.406	.406	.406	.407	.407	.409	.411	.414	.418	.425
-.33155	.361	.361	.361	.361	.361	.362	.362	.363	.365	.367
-.29665	.130	.130	.130	.180	.280	.340	.340	.340	.340	.340
-.26175	.062	.062	.062	.112	.212	.312	.340	.340	.340	.340
-.22685	.013	.013	.013	.044	.144	.244	.340	.340	.340	.340
-.19195	.011	.011	.011	.011	.076	.176	.276	.340	.340	.340
-.15705	.010	.010	.010	.010	.010	.108	.208	.308	.340	.340
-.12215	.009	.009	.009	.009	.009	.041	.141	.241	.340	.340
-.08725	.008	.008	.008	.008	.008	.008	.074	.174	.274	.340
-.05235	.008	.008	.008	.008	.008	.008	.008	.107	.207	.340
-.01745	.008	.008	.008	.008	.008	.008	.008	.041	.141	.340
.01745	.008	.008	.008	.008	.008	.008	.008	.041	.141	.340
.05235	.008	.008	.008	.008	.008	.008	.008	.107	.207	.340
.08725	.008	.008	.008	.008	.008	.008	.074	.174	.274	.340
.12215	.009	.009	.009	.009	.009	.041	.141	.241	.340	.340
.15707	.010	.010	.010	.010	.010	.108	.208	.308	.340	.340
.19195	.011	.011	.011	.011	.076	.176	.276	.340	.340	.340
.22685	.013	.013	.013	.044	.144	.244	.340	.340	.340	.340
.26175	.062	.062	.062	.112	.212	.312	.340	.340	.340	.340
.29665	.130	.130	.130	.180	.280	.340	.340	.340	.340	.340
.33155	.361	.361	.361	.361	.361	.362	.362	.363	.365	.367
.36645	.406	.406	.406	.407	.407	.409	.411	.414	.418	.425
.40135	.455	.455	.456	.456	.457	.459	.463	.468	.476	.487
.43625	.508	.508	.508	.509	.511	.514	.519	.527	.538	.554
.47115	.563	.563	.564	.565	.567	.571	.578	.588	.603	.624
.50605	.622	.622	.622	.623	.626	.632	.640	.653	.672	.699

$\alpha_{\Psi Y}$ breakpoint	$C_{L_{\Psi Y}} = f(\alpha_{\Psi Y}, U_{T_{\Psi Y}})$ for –									
	$U_{T_{\Psi Y}} = 110$	$U_{T_{\Psi Y}} = 220$	$U_{T_{\Psi Y}} = 330$	$U_{T_{\Psi Y}} = 440$	$U_{T_{\Psi Y}} = 550$	$U_{T_{\Psi Y}} = 660$	$U_{T_{\Psi Y}} = 770$	$U_{T_{\Psi Y}} = 880$	$U_{T_{\Psi Y}} = 990$	$U_{T_{\Psi Y}} = 1100$
-0.50605	-0.995	-0.995	-0.996	-0.997	-1.000	-1.005	-1.013	-1.025	-1.042	-1.067
-.47115	-.962	-.963	-.963	-.964	-.996	-.970	-.977	-.987	-1.002	-1.023
-.43625	-.926	-.926	-.926	-.927	-.929	-.932	-.937	-.945	-.957	-.974
-.40135	-.885	-.885	-.885	-.886	-.887	-.890	-.893	-.899	-.908	-.920
-.36645	-.841	-.841	-.841	-.842	-.842	-.844	-.846	-.850	-.855	-.862
-.33155	-.794	-.794	-.794	-.794	-.794	-.794	-.795	-.796	-.798	-.801
-.29665	-.840	-.822	-.808	-.797	-.787	-.779	-.773	-.768	-.762	-.755
-.26175	-.980	-.928	-.885	-.851	-.822	-.798	-.779	-.764	-.748	-.727
-.22685	-1.120	-1.034	-.963	-.904	-.857	-.818	-.785	-.760	-.733	-.699
-.19195	-1.183	-1.140	-1.04	-.958	-.892	-.837	-.792	-.756	-.718	-.670
-.15705	-.967	-.982	-1.009	-1.012	-.927	-.856	-.798	-.752	-.704	-.642
-.12215	-.752	-.764	-.785	-.817	-.864	-.875	-.804	-.748	-.689	-.614
-.08725	-.537	-.546	-.560	-.583	-.617	-.668	-.749	-.744	-.675	-.500
-.05235	-.322	-.327	-.336	-.350	-.370	-.401	-.449	-.740	-.466	-.300
-.01745	-.107	-.109	-.112	-.116	-.123	-.133	-.149	-.283	-.155	-.100
.01745	.107	.109	.112	.116	.123	.133	.149	.283	.155	.100
.05235	.322	.327	.336	.350	.370	.401	.449	.740	.466	.300
.08725	.537	.546	.560	.583	.617	.668	.749	.744	.675	.500
.12215	.752	.764	.785	.817	.864	.875	.804	.748	.689	.614
.15705	.967	.982	1.009	1.012	.927	.856	.798	.752	.704	.642
.19195	1.183	1.140	1.040	.958	.892	.837	.792	.756	.718	.670
.22685	1.120	1.034	.963	.904	.857	.818	.785	.760	.733	.699
.26175	.980	.928	.885	.851	.822	.798	.779	.764	.748	.727
.29665	.840	.822	.808	.797	.787	.779	.773	.768	.762	.755
.33155	.794	.794	.794	.794	.794	.794	.795	.796	.798	.801
.36645	.841	.841	.841	.842	.842	.844	.846	.850	.855	.862
.40135	.885	.885	.885	.886	.887	.890	.893	.899	.908	.920
.43625	.926	.926	.926	.927	.929	.932	.937	.945	.957	.974
.47115	.962	.963	.963	.964	.966	.970	.977	.987	1.002	1.023
.50605	.995	.995	.996	.997	1.000	1.005	1.013	1.025	1.042	1.067

APPENDIX D

TRIM CIRCUIT ALGORITHM

By Gary A. McDaniel
Electronic Associates, Inc.

A generalized trim algorithm taken from reference 6 and described in this appendix has been used successfully in several RTS programs at the Langley Research Center. This method, a generalized secant method, is applicable since it requires no first partial derivatives in making the linear approximation to the trim equations. This is a desirable feature inasmuch as many RTS programs contain function data rather than analytical expressions representing the aerodynamic coefficients and derivatives.

The generalized secant algorithm is an extension of regula falsi to multivariable functions. By letting 0 be an m-dimensional zero vector, the problem can be stated: Given an m-dimensional vector $x = (x_1, \ldots, x_m)$ of independent elements (called the variable vector) and an m-dimensional vector $F(x) = (F_1(x), \ldots, F_m(x))$ whose elements depend on x (called the function vector), find an m-dimensional solution vector x_o such that $F(x_o) = 0$. The vector x_o is computed by the following steps. Pick a nominal value of x^1. Obtain m additional variable vectors by successively varying only one component of x^1. There is now a set of $m + 1$ variable vectors, that is, $x^1, x^2, \ldots, x^{m+1}$. With each member of this set, associate a function vector $F^1, F^2, \ldots, F^{m+1}$. From this set of function vectors, compute an $m + 1$ dimensional sum vector $S = (F^1 \cdot F^1, F^2 \cdot F^2, \ldots, F^{m+1} \cdot F^{m+1})$. Use this S vector to find

$$S^{max} = \max_{i=1}^{m+1} S_i \quad \text{and} \quad S^{min} = \min_{i=1}^{m+1} S_i$$

With S^{max} and S^{min}, there is associated in an obvious way x^{max}, F^{max}, x^{min}, and F^{min} Pick a value for K and solve the following linear system for an m+1—dimensional vector $q = (q_1, q_2, \ldots, q_{m+1})$, which is called the q-vector:

$$\sum_{j=1}^{m+1} q_j = 1 \tag{D1}$$

$$\sum_{j=1}^{m+1} q_j F_i^j = (1 - K)F_i^{min} \tag{D2}$$

where $0 < K \leq 1$. Then, compute a new variable vector from

$$x_i^{new} = \sum_{j=1}^{m+1} q_j x_i^j \tag{D3}$$

Associated with this new variable vector will be a new function vector F^{new} and a new sum vector $S^{new} = F^{new}$. If $S^{new} > S^{max}$, halve K and resolve equation (D3) for x^{new}. Continue this procedure until $S^{new} \leq S^{max}$. When this occurs, replace F^{max} by F^{new} in equation (D2). Compute a new S, S^{max}, and S^{min}. Replace F^{min} on the right-hand side of equation (D2) by the new F^{min}. Replace x^{max} on the right-hand side of equation (D3) by x^{new} and resolve equation (D2) until $F^{new} = 0$ to within some tolerance. Therefore, K is doubled up to a maximum of 1 each time $S^{new} \leq S^{max}$ within three trials.

REFERENCES

1. Baker, E. B.; Hay, I.; and Mitchell, E. E. L.: Development of Hybrid Computer Programs for Simulation of Cobra. Vol. 1: Cobra Airframe Simulation. Tech. Rep. ECOM-0387-F2a, U.S. Army, Jan. 1969. (Available from DDC as AD 852 533.)

2. Toler, James R.; McIntyre, Walter; and Coffee, Merlin P.: Simulation of Helicopter and V/STOL Aircraft. Vol. I – Helicopter Analysis Report. Tech. Rep. NAVTRADEVCEN 1205-1, U.S. Navy, Sept. 1963. (Available from DDC as AD 601 022.)

3. White, Ellis: Eastern Simulation Council Meeting. Simulation, vol. 12, no. 2, Feb. 1969, pp. 53-56.

4. Grove, Randall D.; and Mayhew, Stanley C.: A Real-Time Digital Program for Estimating Aircraft Stability and Control Parameters From Flight Test Data by Using the Maximum Likelihood Method. NASA TM X-2788, 1973.

5. Barker, L. E., Jr.; and Kibler, K. S.: Description of the Tactical Effectiveness Simulation Program. NASA paper presented at the Eastern Simulation Council (Hampton, Va.), Sept. 26, 1968.

6. Burrows, Roger R.; and McDaniel, Gary A.: A Method of Trajectory Analysis With Multi-Mission Capability and Guidance Application. AIAA Paper No. 68-844, Aug. 1968.

7. Barker, Lawrence E., Jr.; Bowles, Roland L.; and Williams, Louise H.: Development and Application of a Local Linearization Algorithm for the Integration of Quaternion Rate Equations in Real-Time Flight Simulation Problems. NASA TN D-7347, 1973.

8. Gessow, Alfred; and Myers, Garry C., Jr.: Aerodynamics of the Helicopter. Macmillan Co., c.1952. (Republished 1967 by Frederick Ungar Pub. Co.)

TABLE I.- COMPUTER PROGRAM STATIC CHECK

[SAS is off]

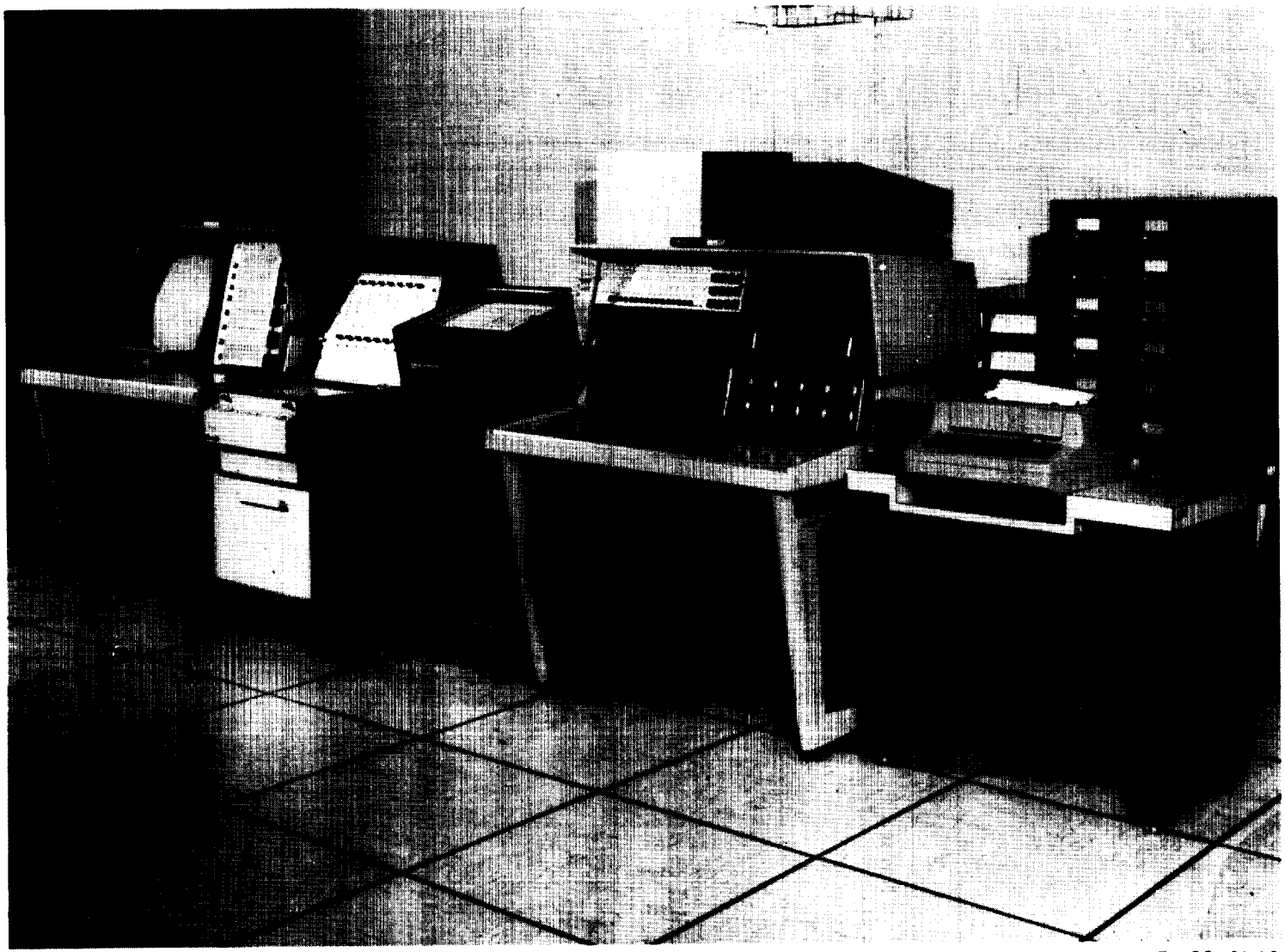
T	0.	ASPD	8.03089770E+01	GRWT	8.82280000E+03	CG	1.93850000E+02
H	5.00000000E+03	HDOT	-4.53263453E-07	OMEG	3.39000000E+01	OMEGDT	4.64422826E-13
U	1.35000000E+02	V	0.	W	-1.31800900E+01	ALFF	-1.30567737E-01
UDOT	-8.15470116E-07	VDOT	-4.73316781E-08	WDOT	-8.63041720E-02	BETF	0.
P	0.	Q	0.	R	0.	DELE	1.12530912E-01
PDOT	1.17304527E-01	QDOT	-3.39395264E-02	RDOT	-2.84419212E-03	LMR	9.02949721E+03
XA	-8.57173014E+02	YA	1.49916881E+02	ZA	-8.80342978E+03	QMR	6.40387922E+03
LA	2.96780455E+02	MA	-3.97635492E+02	NA	-2.89083687E+01	FXR	-5.75123449E+02
PHI	-1.70735810E-02	THETA	-9.73077730E-02	PSI	0.	FYR	-9.35080094E+01
PHID	0.	THETDT	0.	PSIDT	0.	QE	6.40387922E+03
AOSS	4.45834208E-02	A1SS	6.14689089E-02	B1SS	2.52623282E-03	WIM	9.09481658E+00
AOS	2.39685400E-01	A1S	-1.66214530E-02	B1S	-1.18307260E-02	THTR	2.93136550E-02
XF	-2.42817472E+02	YF	-2.40051956E-07	ZF	4.66293973E+02	VTR	0.
LF	-6.62201666E-07	MF	4.11892346E+01	NF	2.02799645E-06	YTR	1.50214929E+02
WHS	-1.33897332E+01	YVS	9.32099616E+01	ZHS	-1.23087275E+01	WIWM	0.
XW	-3.92320927E+01	ZW	-2.27917809E+02				

TABLE II.- COCKPIT INSTRUMENT CHECK

DAC			Instrument	
Element	Variable	Voltage	Name	Reading
25	$\cos(\Phi)$	-86.54	Attitude direction indicator – roll	$+30^{o}$
26	$\sin(\Phi)$	-50.04		
27	$\cos(\theta)$	-70.71	Attitude direction indicator – pitch	$+45^{o}$
28	$\sin(\theta)$	-70.71		
29	$\cos(\Psi)$	-86.58	Heading indicator	$+30^{o}$
30	$\sin(\Psi)$	-50.04		
31	cos(h)	-95.11	Altimeter	50 ft
32	sin(h)	+30.90		
33	$\cos(\dot{h})$	-46.96	Rate of climb indicator	1500 ft/min
34	$\sin(\dot{h})$	+88.29		
35	cos(r)	-86.58	Rate of turn indicator	2 needle widths right
36	sin(r)	+50.00		
37	$\cos(-a_Y/a_Z)$	+41.62	Bank indicator	1 ball width right
38	$\sin(-a_Y/a_Z)$	-90.93		
39	Ω	+64.80	Main rotor rpm indicator	323 rpm
40	A_{SPD}	+72.22	Airspeed indicator	130 knots

TABLE III.- CONTROL DEFLECTION CHECK

Cockpit control		ADC		
Name	Range	Element	Voltage	Variable
Collective stick	Full down to full up	1	0 to +80	X_{AOS}
Lateral cyclic stick	Full left to full right	2	-60 to +60	Y_{CS}
Longitudinal cyclic stick	Full forward to full back	3	-60 to +60	X_{CS}
Tail rotor pedals	Full left to full right	4	-60 to +60	X_{TR}
Lateral cyclic trim dial	Full left to full right	5	-100 to +100	-----
Longitudinal cyclic trim dial	Full left to full right	6	-100 to +100	-----
Tail rotor pedal trim dial	Full left to full right	7	-100 to +100	-----



L-69-8762

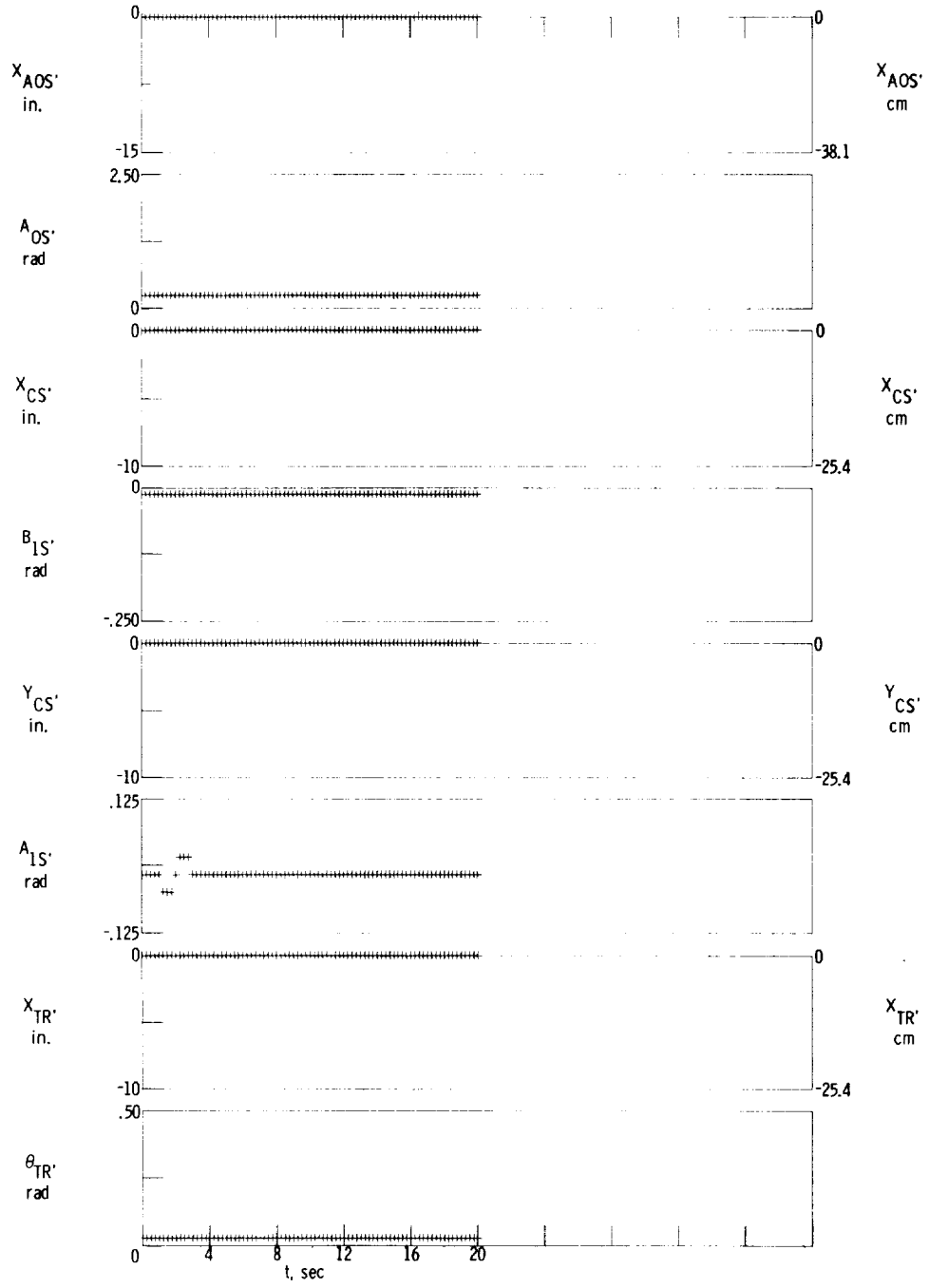
(a) Typical program control station.

Figure 1.- Operational control features.

L-68-10 517

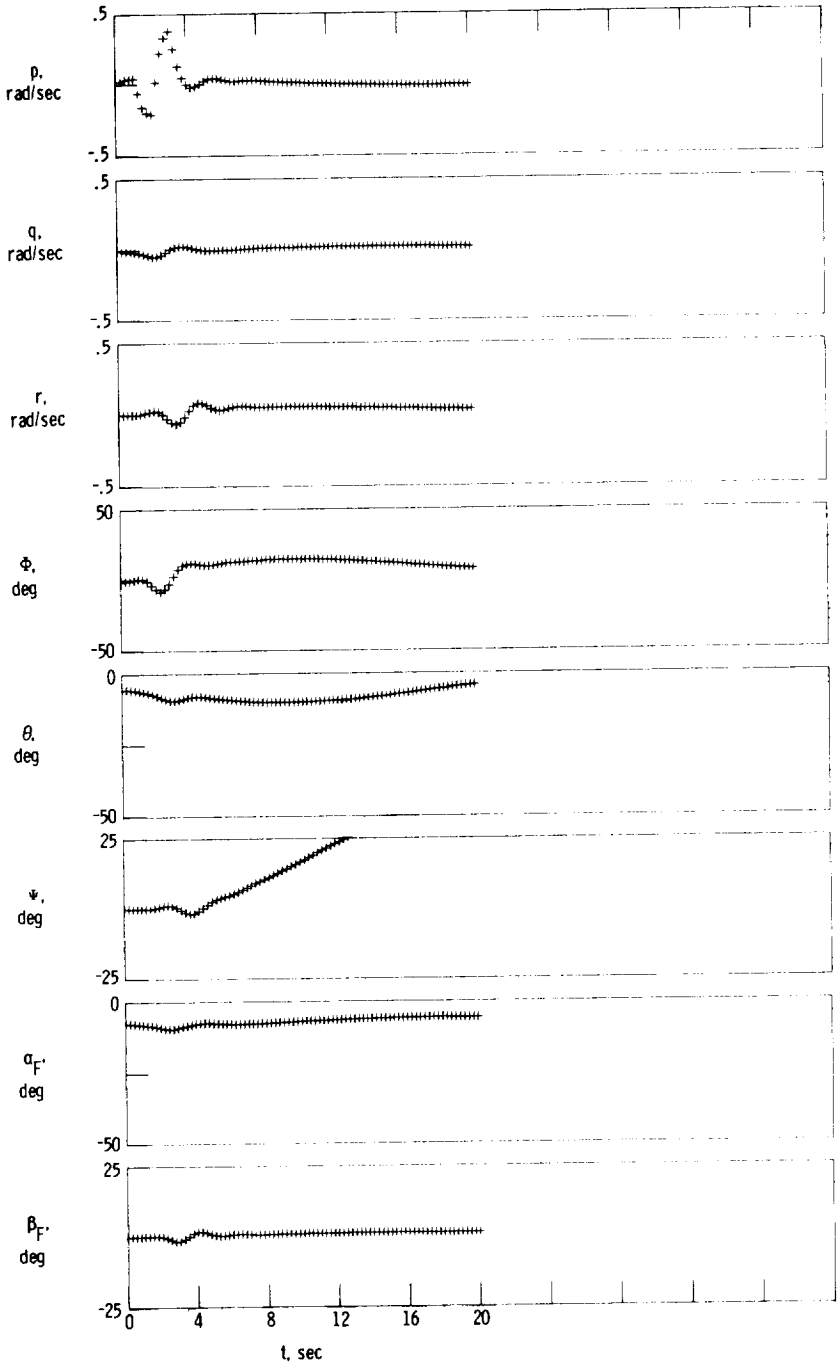
(b) Closeup of control panel on the program control console.

Figure 1.- Concluded.



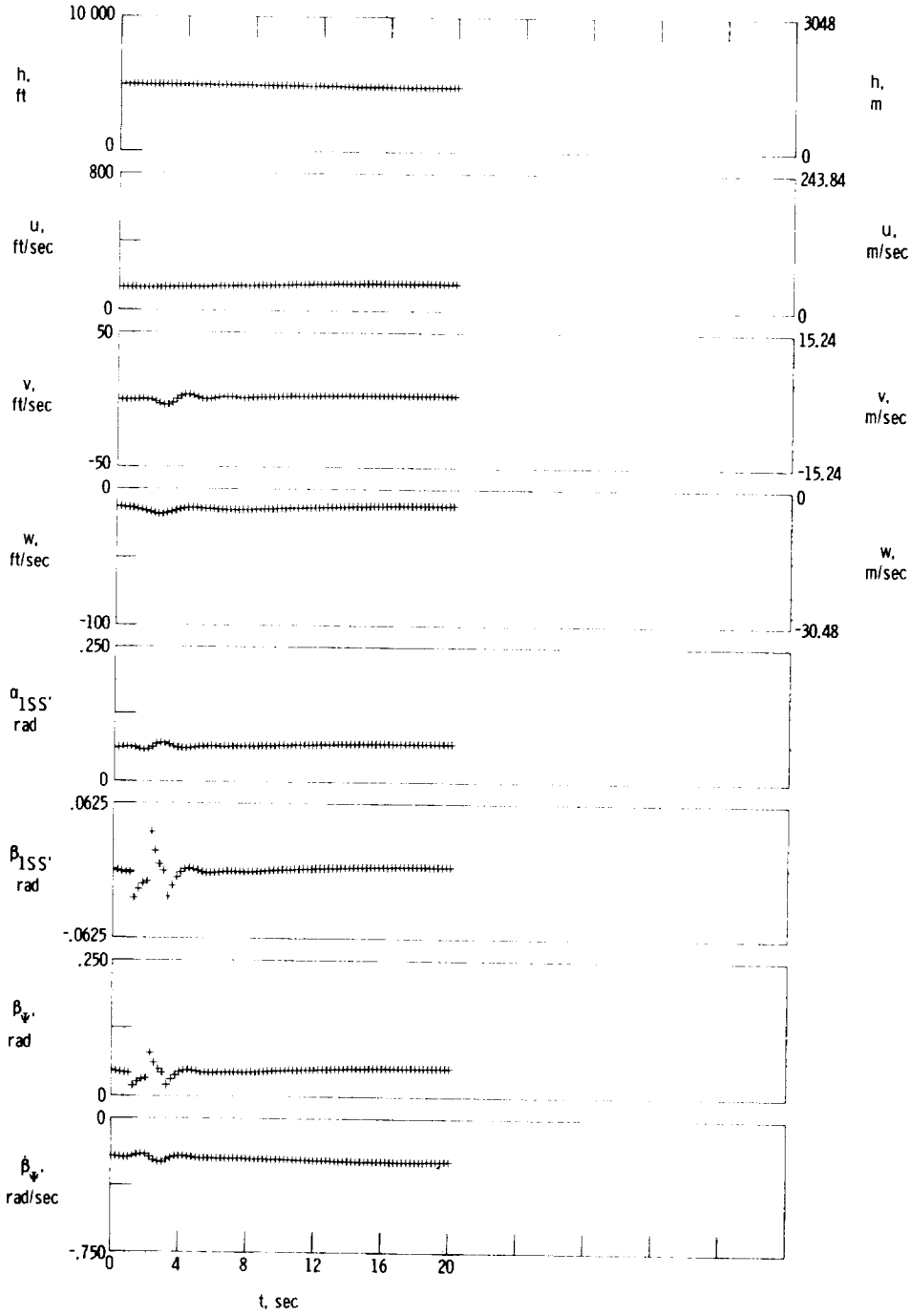
(a) Recorder 1.

Figure 2.- Dynamic check at 148.16 km/hr (80 knots) with A_{1S} doublet input.



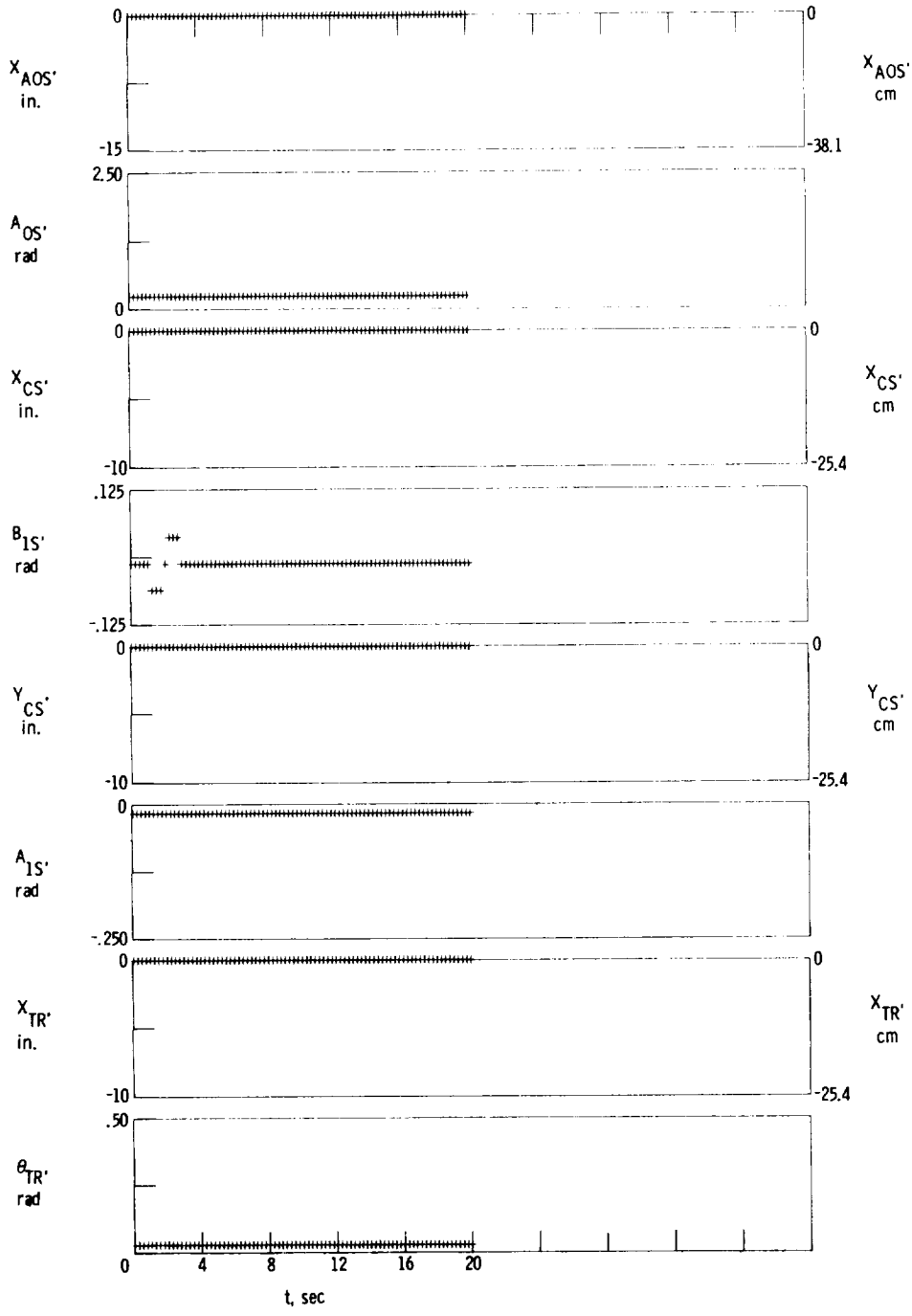
(b) Recorder 2.

Figure 2.- Continued.



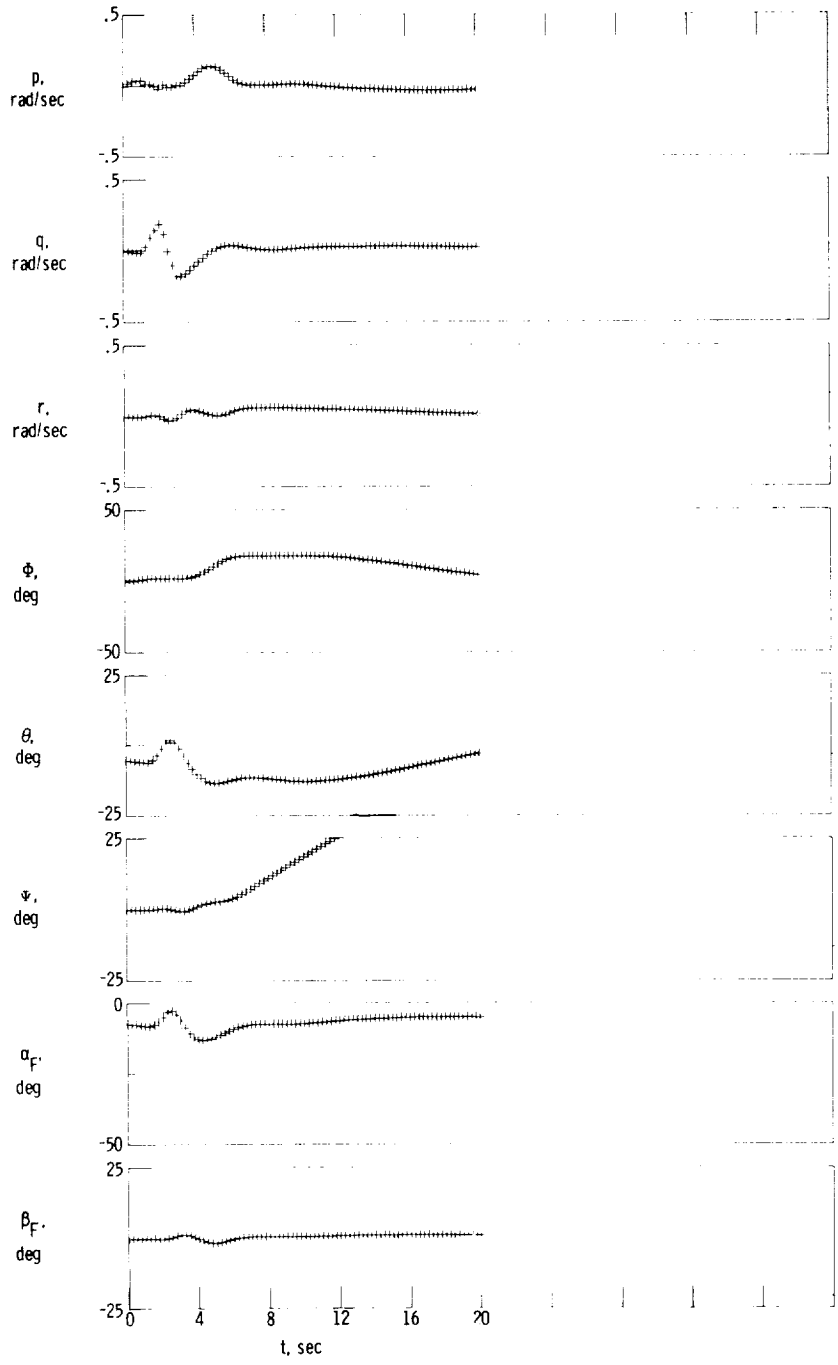
(c) Recorder 3.

Figure 2.- Concluded.



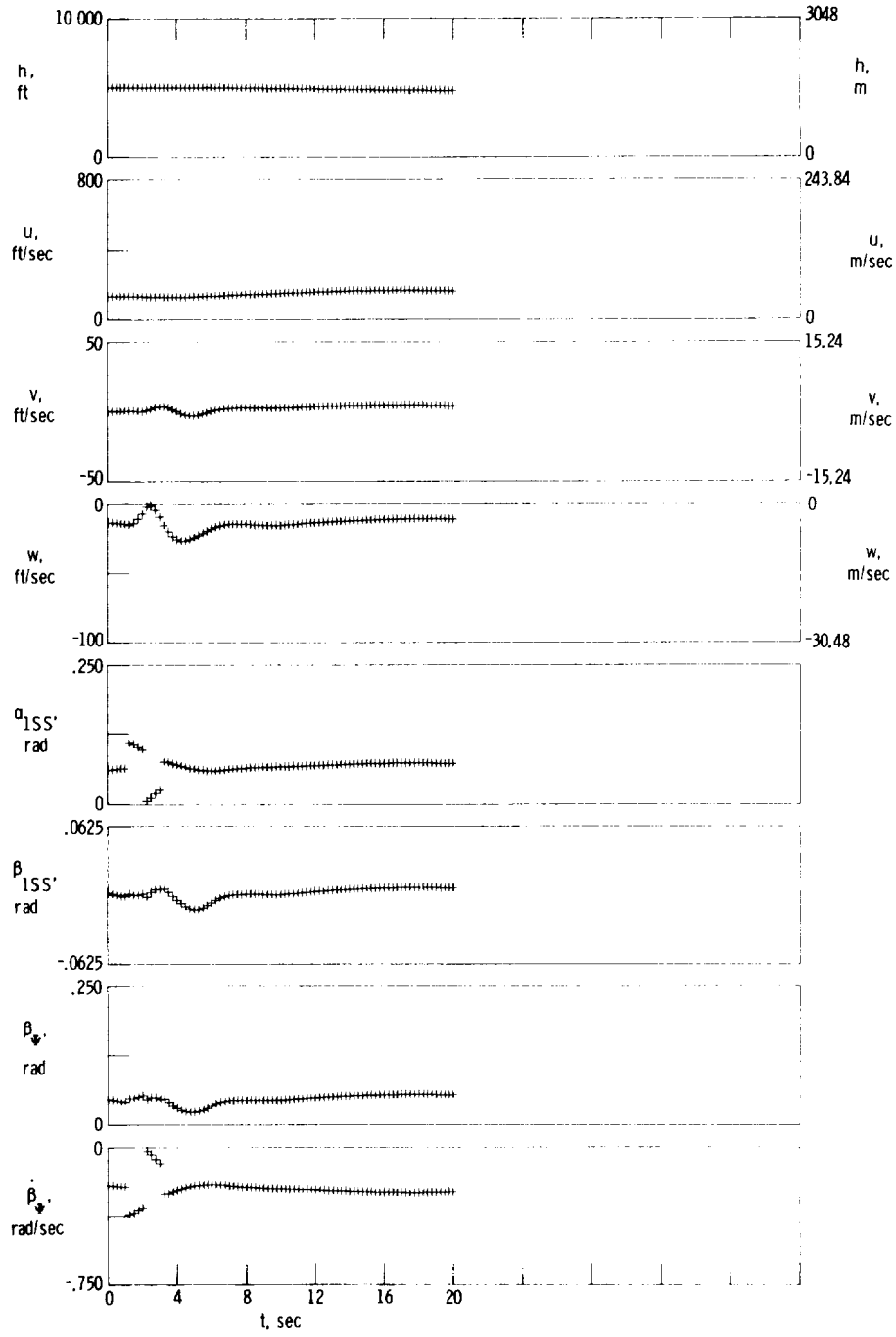
(a) Recorder 1.

Figure 3.- Dynamic check at 148.16 km/hr (80 knots) with B_{1S} doublet input.



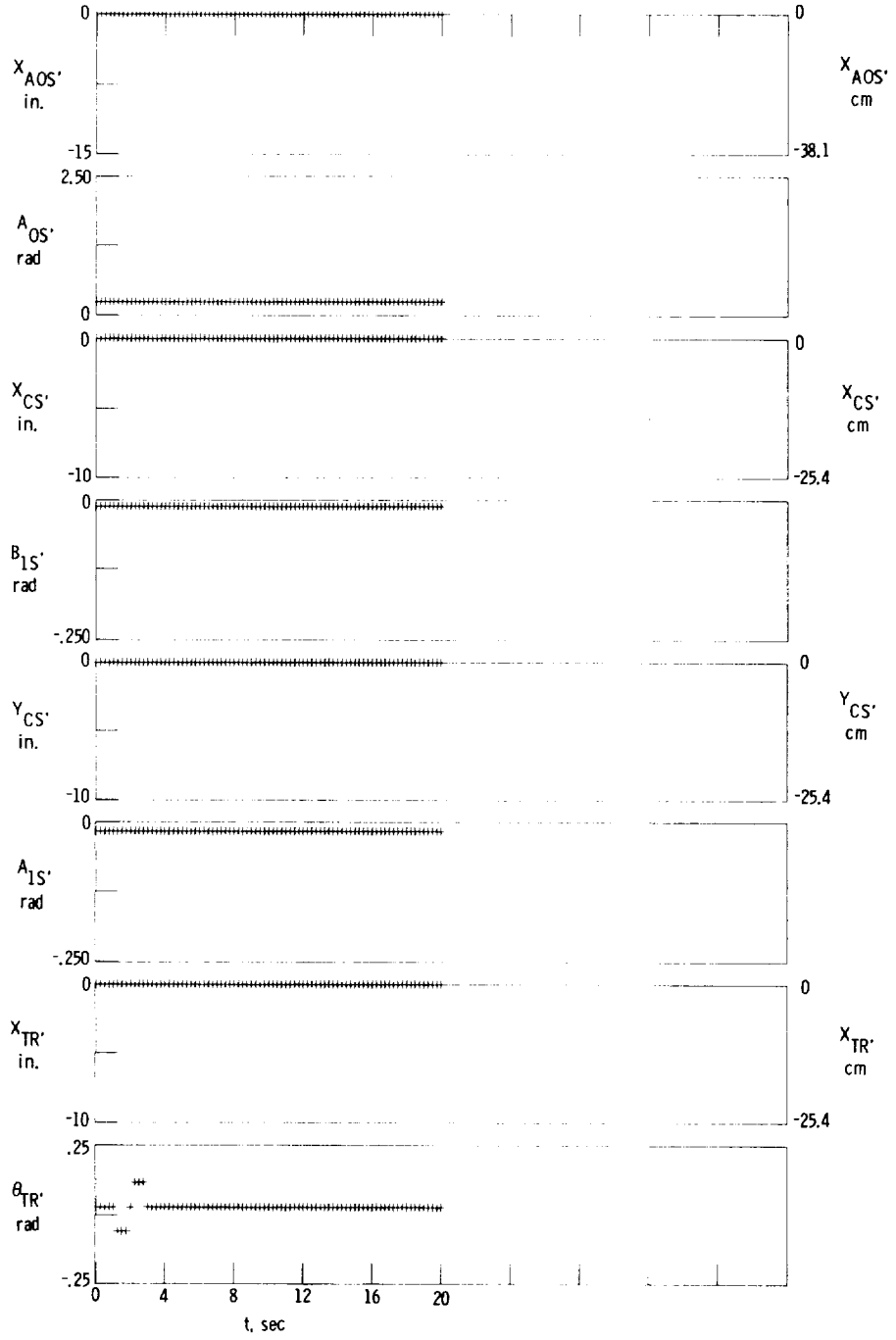
(b) Recorder 2.

Figure 3.- Continued.



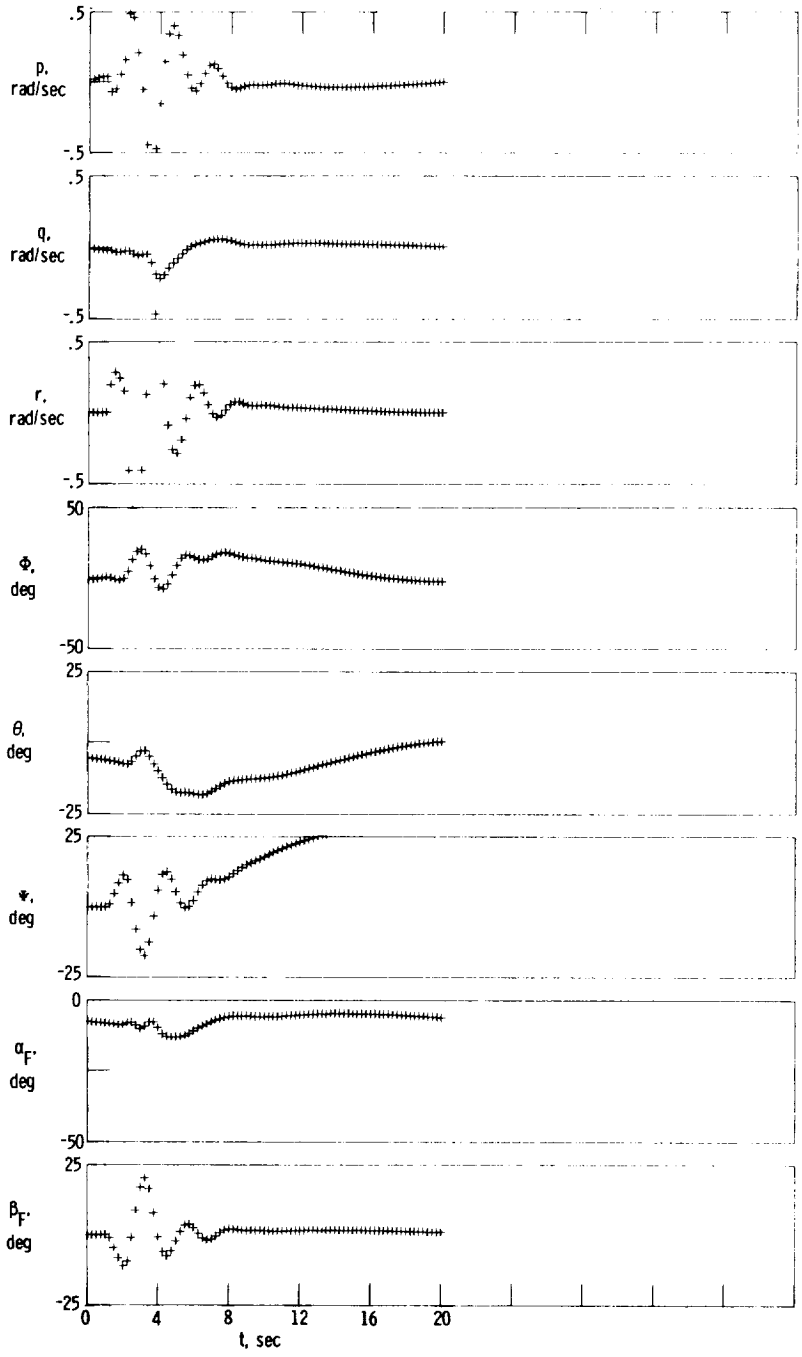
(c) Recorder 3.

Figure 3.- Concluded.



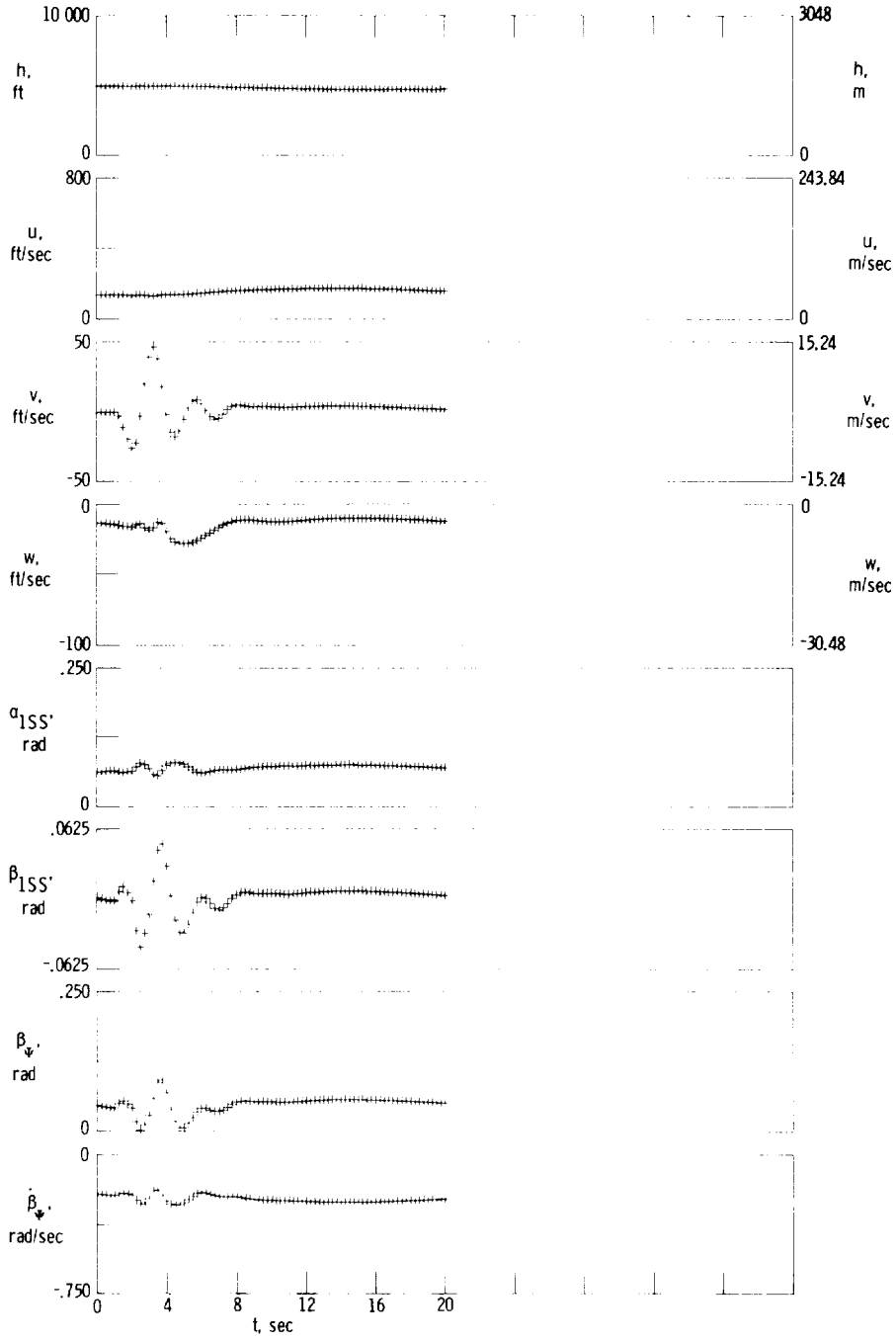
(a) Recorder 1.

Figure 4.- Dynamic check at 148.16 km/hr (80 knots) with θ_{TR} doublet input.



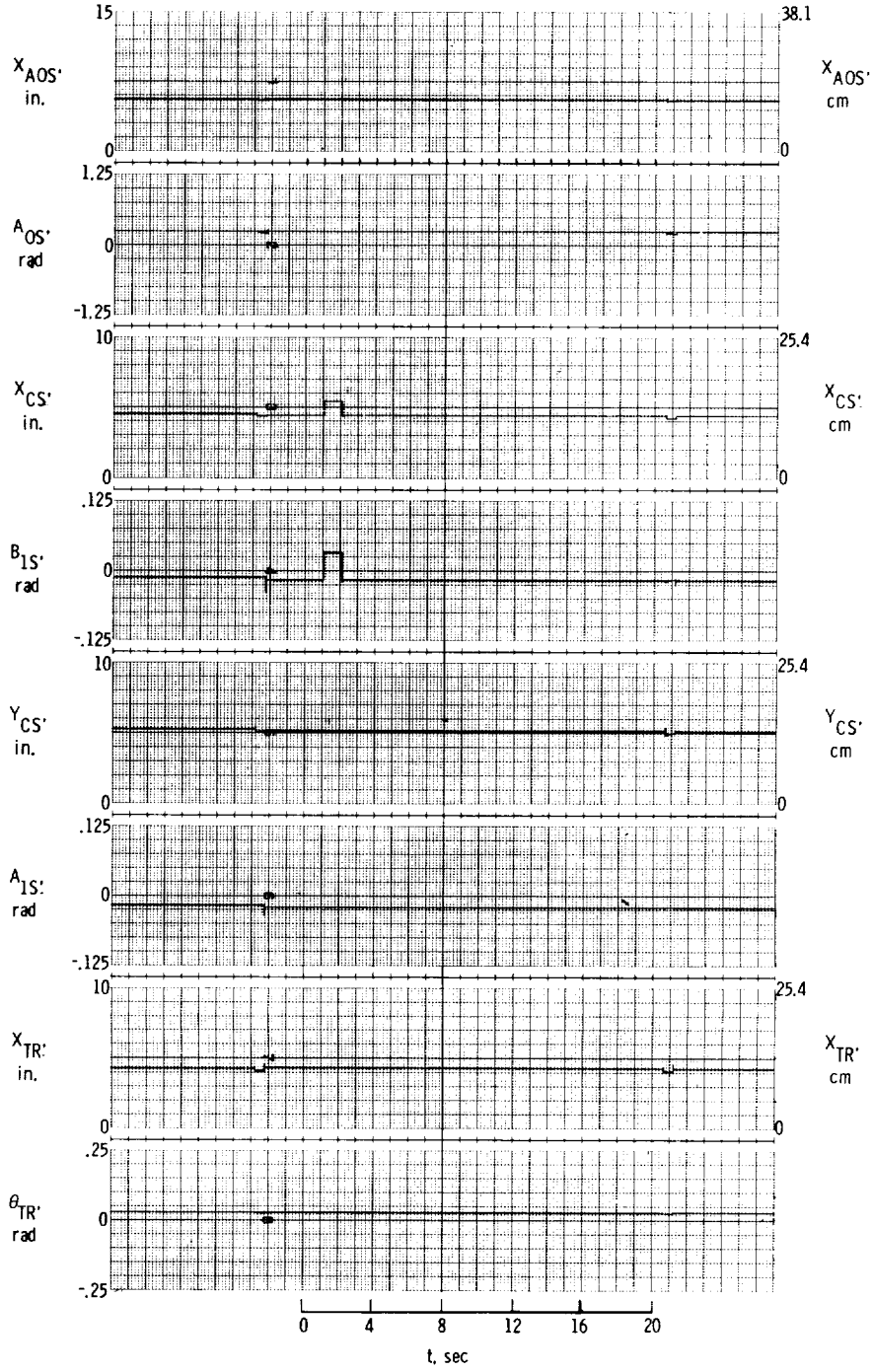
(b) Recorder 2.

Figure 4.- Continued.



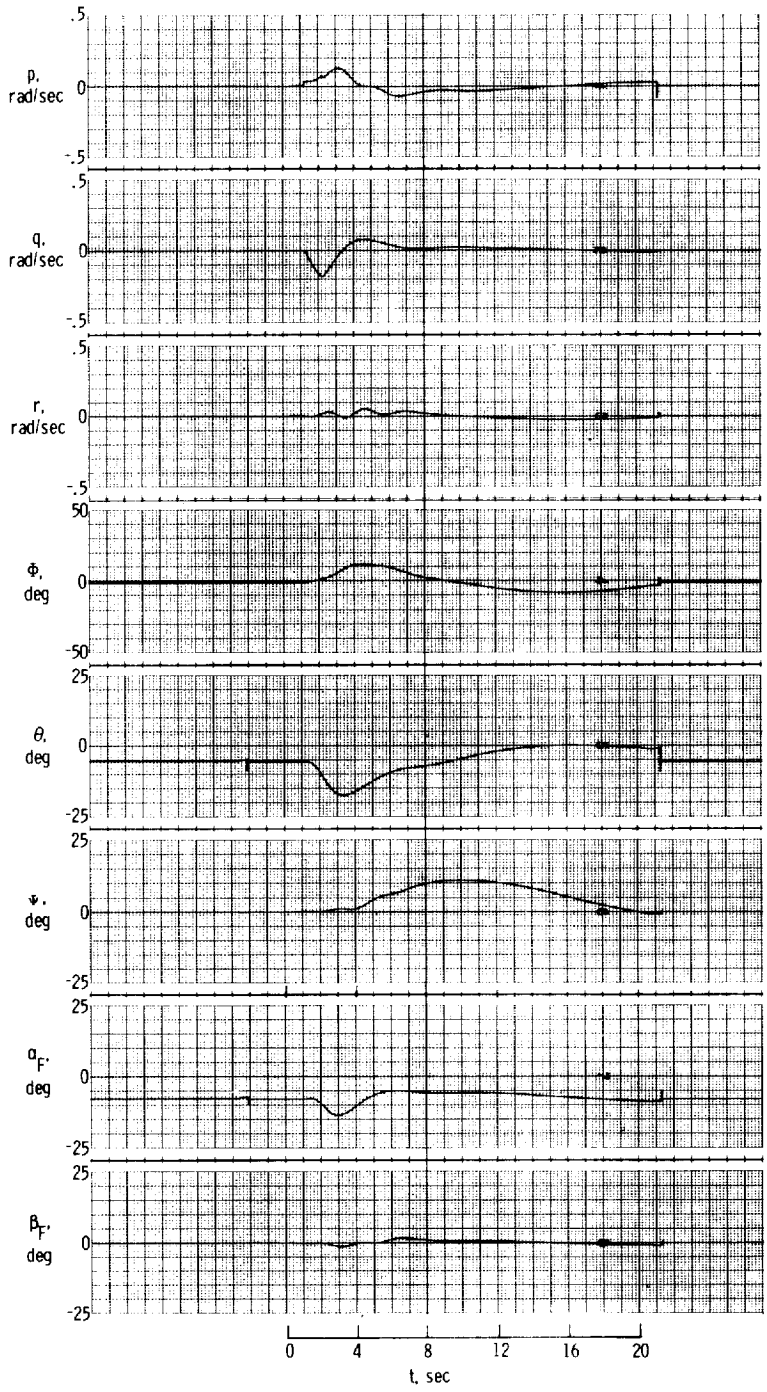
(c) Recorder 3.

Figure 4.- Concluded.



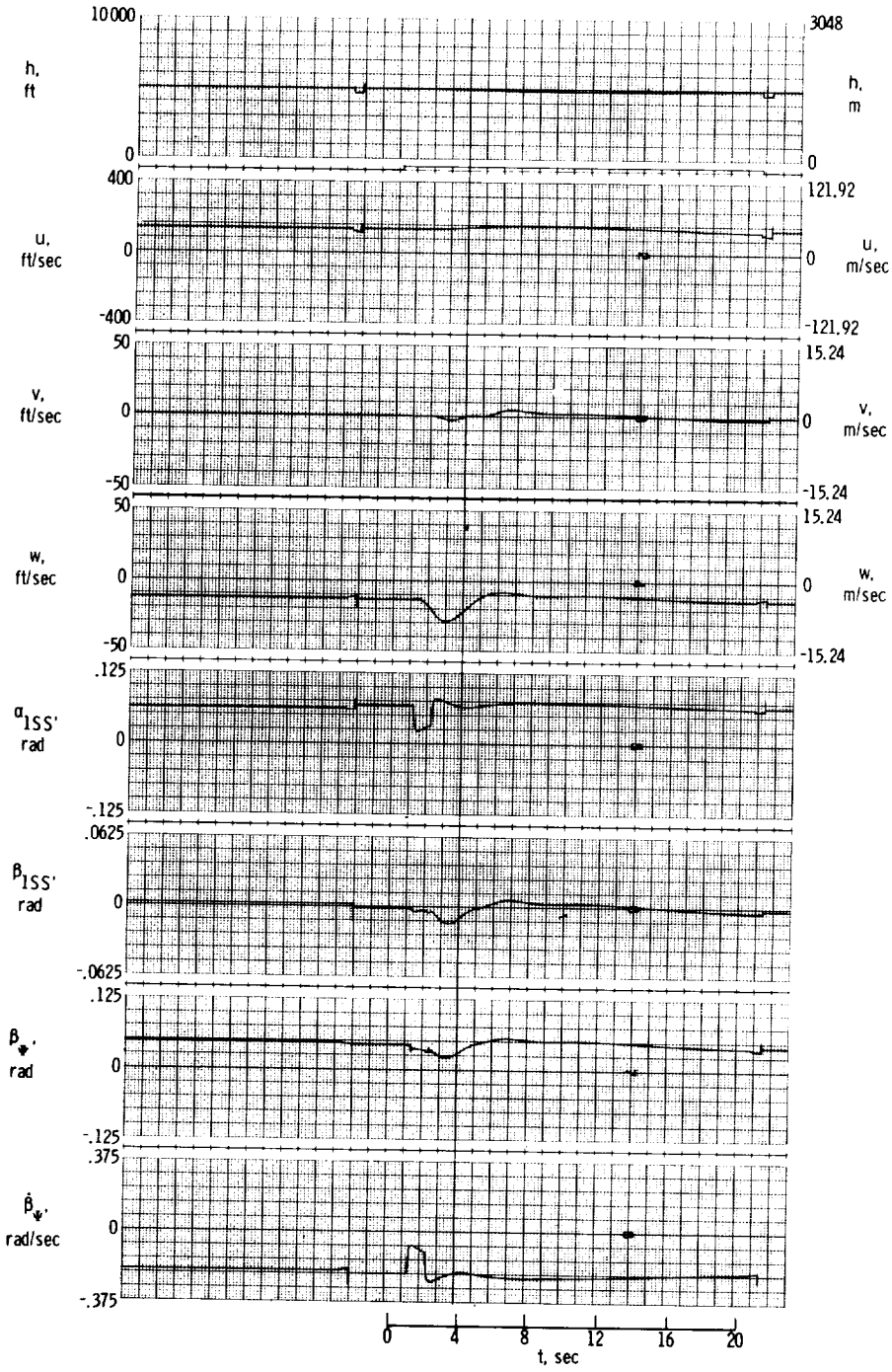
(a) Recorder 1.

Figure 5.- Test flight.



(b) Recorder 2.

Figure 5.- Continued.



(c) Recorder 3.

Figure 5.- Concluded.

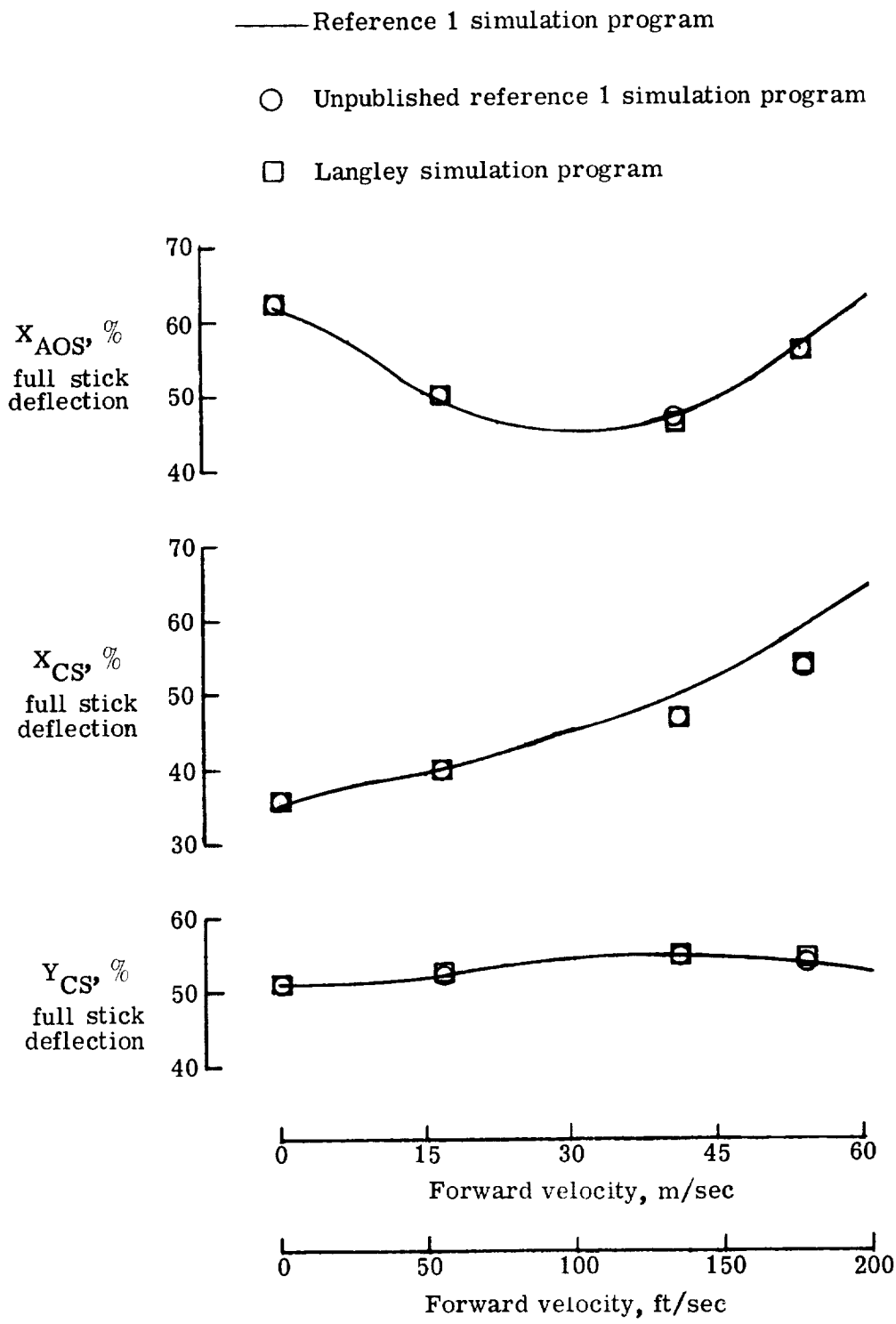


Figure 6.- Comparison of static trim stability.

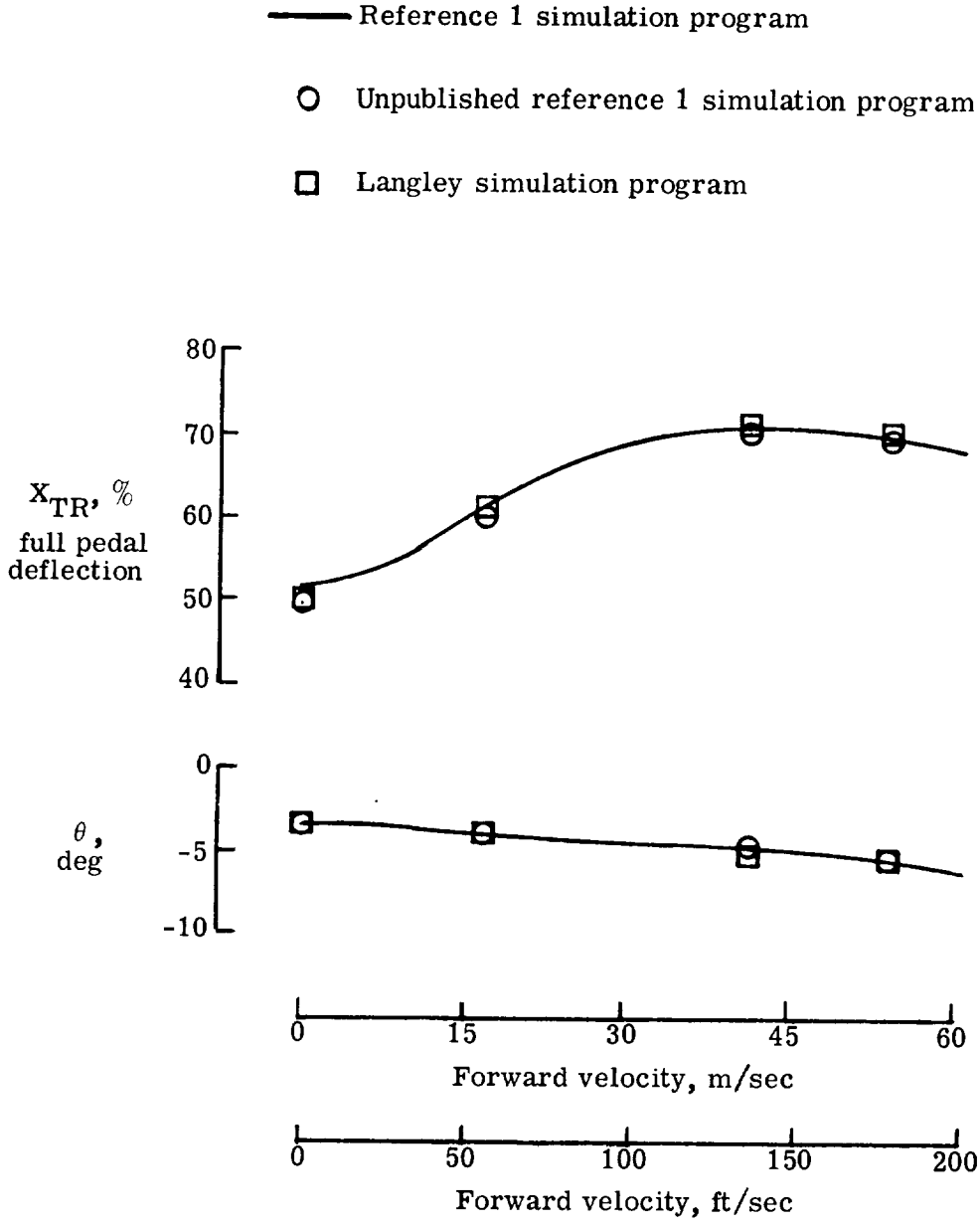


Figure 6.- Concluded.