TR-640-2

USERS MANUAL FOR FLIGHT CONTROL DESIGN PROGRAMS

24 September 1975

(NASA-CR-144913) USERS MANUAL FOR FLIGHT CONTROL DESIGN PROGRAMS (Analytic Sciences Corp.) 103 p HC $5.50 CSCL 01C

N76-24264

Unclas

G3/08 28244

Prepared under:

Contract No. NAS1-13807

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Langley Research Center
Hampton, Virginia 23665

Prepared by:

James Y. Nalbandian

Approved by:

Robert F. Stengel
E. Wayne Vinje

THE ANALYTIC SCIENCES CORPORATION
6 Jacob Way
Reading, Massachusetts 01867

FOREWORD

The computer programs described in this report were developed by The Analytic Sciences Corporation (TASC) during the period from March 3, 1975 to November 25, 1975 under Contract No. NAS1-13807 for the National Aeronautics and Space Administration, Langley Research Center, Hampton, Virginia. The work was sponsored by the Navigation and Guidance Research Branch of the Flight Instrumentation Division as a contribution to the VTOL Automatic Landing Technology (VALT) Program. Dr. David R. Downing served as Technical Monitor for this contract.

The author acknowledges the assistance and technical guidance provided by Mr. John R. Broussard, Mr. Paul W. Berry, and Mr. Joel M. Winett during the course of the computer program development.

ABSTRACT

Computer programs for the design of analog and digital flight control systems have been developed and are documented in this report. The program DIGADAPT uses Linear-Quadratic-Gaussian (LQG) synthesis algorithms in the design of command-response controllers and state estimators, and it applies covariance propagation analysis to the selection of sampling intervals for digital systems. Program SCHED executes correlation and regression analyses for the development of gain and trim schedules to be used in open-loop explicit-adaptive control laws. A linear-time-varying simulation of aircraft motions is provided by the program TVHIS, which includes guidance and control logic, as well as models for control actuator dynamics. The programs are coded in Fortran and have been compiled and executed on both IBM and CDC computers.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF TABLES (Continued)

1. PROGRAM OVERVIEW

This report describes computer programs which can be used in the analysis, design, and evaluation of flight control laws. Design techniques for analog (continuous-time) and digital (discrete-time) control systems are implemented in Program DIGADAPT using Linear-Quadratic-Gaussian (LQG) Synthesis. Program SCHED computes correlations between control constants and flight conditions, indicating possible relationships for scheduling gain and trim settings in the actual system. The program also generates the gain scheduling functions by regression analysis. Program TVHIS is a linear-time-varying simulation of flight motions in the neighborhood of a reference flight path. It is a useful tool for evaluating the performance of control systems alone or in combination with predetermined guidance laws.

Design procedures using these programs can be flexible, as several options for estimation and control logic are provided. Control laws include dynamic, proportional-integral (PI), and proportional-double integral (PII) command-response controllers, as derived in Ref. 1 and summarized in succeeding chapters. Estimation logic is based upon the Kalman filter, and the sampling interval for digital systems is determined using a state covariance propagation technique. Once system gains have been computed at several flight conditions, the gain scheduling feature allows an open-loop explicit-adaptive control law to be defined.

These design programs have been developed for a specific application, the control of a tandem-rotor helicopter, and they can be extended to a variety of aircraft configurations, including other VTOL designs and conventional transport, fighter, and general aviation aircraft. Application to re-entering spacecraft, missiles, and submarines also is straight-forward. In each case, details of control effectors and elastic modes must be specified by the designer. The basic routines and subroutines have been programmed to handle systems of arbitrary dimension, so they are generally applicable to a wide variety of vehicle control laws.

1.1 GENERAL SYSTEM DESCRIPTION

DIGADAPT and SCHED are used to develop a flight control structure and a gain scheduling table. The purpose of TVHIS is to evaluate and determine the success of the design procedure. Figure 1.1-1 gives an overview of the control flow in the design procedure, and Fig. 1.1-2 gives the data flow for this system.

1.2 PROGRAM IMPLEMENTATION

The design programs were developed on an IBM 370/145 using the Fortran H compiler. They also have been converted to run on a CDC 6600 under the SCOPE 3.3 operating system using the FTN Fortran compiler (Ref. 2). A number of design subroutines were adapted from Ref. 3.

A permanent disk file is used to store aerodynamic coefficients. This file must be established using the procedure described in Appendix A before DIGADAPT or TVHIS is executed.

On the CDC 6600 computer DIGADAPT requires 44,352 decimal ($126{,}500_8$) locations for execution; approximately

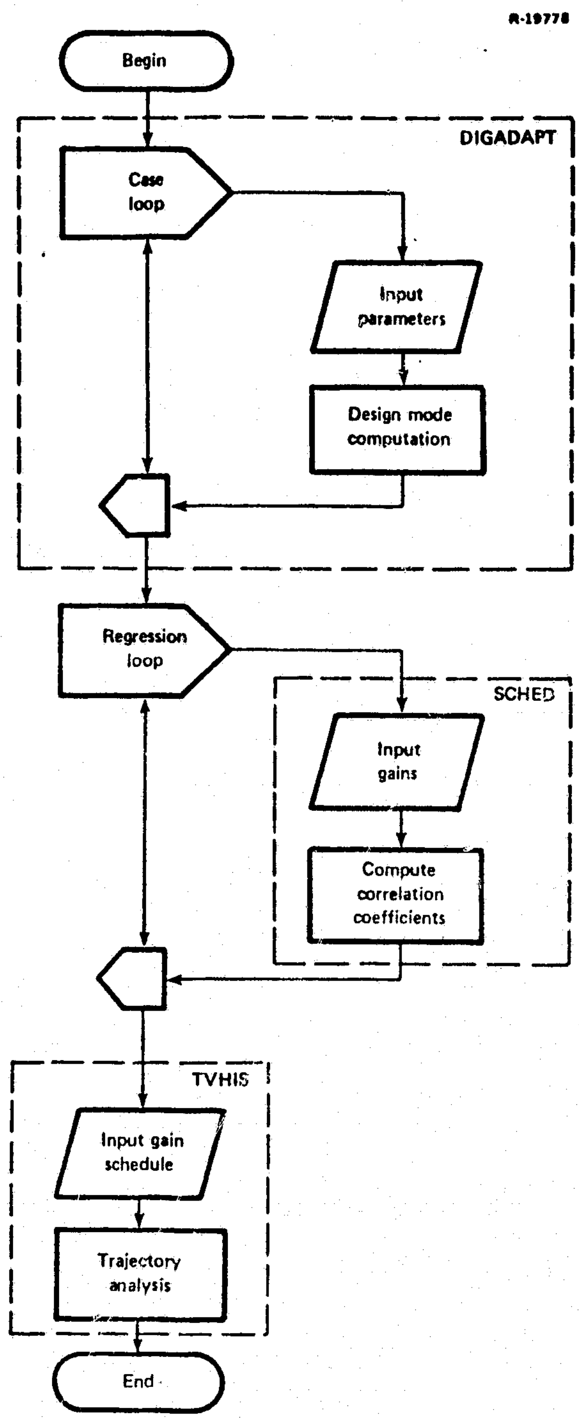


Figure 1.1-1 Control Flow

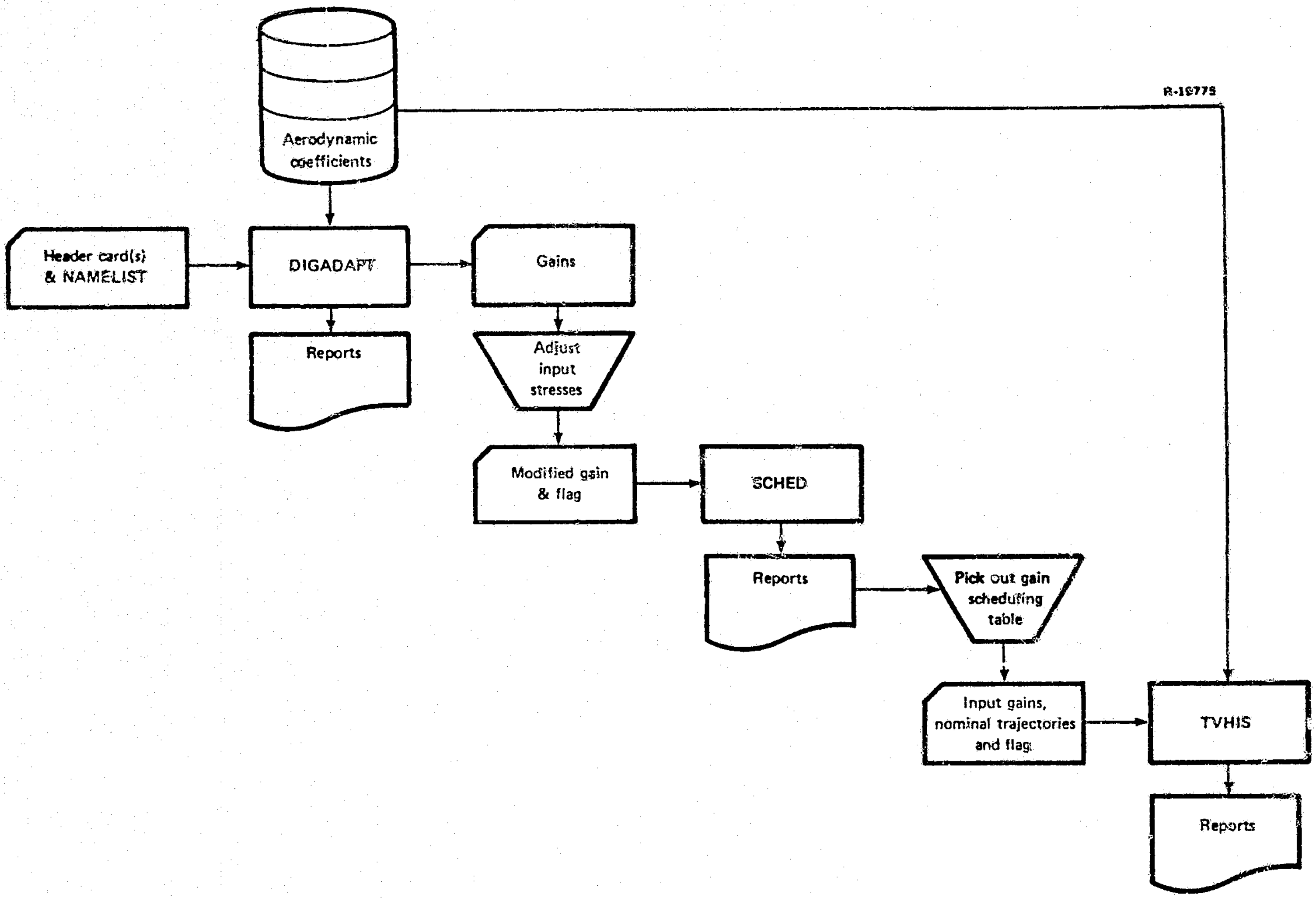


Figure 1.1-2 Data Flow

$40{,}000_8$ locations are required for array storage to execute a design procedure for a 13-state, 4-control system; and approximately $30{,}000_8$ locations are used by the SCOPE operating system. An overlay or segment procedure has not been defined for the following reasons:

- Two distinct algorithms are used to solve matrix Riccati equations. This assures convergence of solutions in minimum time but increases the storage requirement.
- System overhead which would be incurred with overlay and segment structure is eliminated; hence, run time for each design is reduced.

SCHED requires 33,664 decimal ($101{,}600_8$) locations for execution and TVHIS requires 33,024 decimal ($100{,}400_8$) locations for execution.

1.3 DOCUMENTATION

This document is divided into seven chapters. Chapter 1 gives an overview of the three programs, DIGADAPT, SCHED and TVHIS, and a discussion of program implementation and documentation. Chapter 2 describes all the inputs and outputs of DIGADAPT, and Chapter 3 describes in detail the significant subroutines of DIGADAPT. Similarly, Chapters 4 and 6 give the inputs and outputs of SCHED and TVHIS, respectively, while Chapters 5 and 7 give a detailed description of SCHED and TVHIS, respectively.

Appendix A is concerned with the creation of the permanent disk file containing aerodynamic coefficients. It describes how to pick appropriate cases once the file has been created. Appendix B illustrates the flowcharting symbols used, and Appendix C outlines the design procedure used in DIGADAPT and SCHED. Appendix D describes the list of deliverables.

2. DIGADAPT USAGE

DIGADAPT has been written using top-down structure and depends on user input to determine which of the program chapters are to be used for a given design (see Table 2-1). There are seven such program chapters:* The main executive, system matrix computation, sampling interval determination, linear-optimal gain design, estimator/observer design, calculation of eigenvalues, and time history simulations. Two flags control the main logic, ICHPTR(7,6) and ITYP(2). ICHPTR(I,J) indicates which chapters are to be executed; I represents the i^{th} chapter and J represents the j^{th} section within the i^{th} chapter. The default values for ICHPTR(I,J) are all 0 (representing "off"), and it is necessary for the user to override these values via NAMELIST by setting ICHPTR(I,J) to 1 (representing "on"). ITYP is a working flag whose definition depends upon the procedure being executed. Table 2-2 is a summary of flag settings for specific features of DIGADAPT.

2.1 DIGADAPT INPUT

Input to DIGADAPT takes two forms: card input on Fortran logical unit 5 (TAPE5) and permanent file disk inputs on Fortran logical unit 1 (TAPE1). The card input consists of a header card for the first case followed by NAMELIST input cards used to override program variable defaults set in

---

*Program chapters refer to the logical structure of the program DIGADAPT and should not be confused with chapters of this document.

TABLE 2-1

PROGRAM CHAPTERS OF DIGADAPT

1. Main Executive - Run/Case Setup
   Initialize constants
   Read in dimensions, flags and case number

2. Full Model - Stability Derivatives, F and G (body axes)
   2.1 F and G data input
       Primary routine - AERO
       Secondary routine - FGCOMP

3. Sampling Interval Determination
   3.1 Sampling time calculation
       Primary routine - SID
       Secondary routine - SAMPLE

4. Digital - Adaptive Control Law
   4.1 Q and R input
   4.2 Addition of integrator states for Proportional-plus-Double Integral (PII) controller
   4.3 Linear-optimal regulator gains: continuous-time case
   4.4 Discretization of continuous-time system
   4.5 Linear-optimal regulator gains: discrete-time case
   4.6 Proportional-integral (PI) controller
       Primary subroutine - GAIN
       Secondary routines - modified ESL package

5. Estimator Design
   5.1 Kalman filter gains - discrete-time case
       Primary routine - EOD
       Secondary routines - ESTOBS, KFD

6. Eigenvalues - Stability Criteria For LTI Models
   6.1 Eigenvalues of the closed-loop-continuous system ACL
   6.2 Eigenvalues of the closed-loop-discrete system ACLD
   6.3 Eigenvalues of the closed-loop-discrete Kalman filter FCL
       Primary subroutine - FREE
       Secondary routines - ATEIG, HSBG

7. Time History Computation
   7.1 LTI time history of PII controller or dynamic controller
   7.2 A discrete time history of PII system will be performed
   7.3 A discrete time history of the PII system is computed
   7.4 An approximate discrete-time history of the PII system is computed
   7.5 A discrete time history of the PI and/or control-rate system is computed
       Primary subroutine - TIME
       Secondary subroutines - TIMHIS, DTMHIS, PIDHIS

TABLE 2-2
SUMMARY OF DESIGNS

	CONTINUOUS TIME			DISCRETE TIME												
	DYNAMIC CONTROLLER	PII ATTITUDE CONTROL	PII VELOCITY CONTROL	PI ATTITUDE CONTROL	PI VELOCITY CONTROL	PII ATTITUDE CONTROL	PII VELOCITY CONTROL	DYNAMIC CONTROLLER VELOCITY CONTROL	DYNAMIC CONTROLLER ATTITUDE CONTROL	TRIM VALUES	TRIM MATRICES F AND G	SAMPLING INTERVAL	KALMAN FILTER	SAMPLE 3	SAMPLE 2	SAMPLE 1
ICHPTR(2,1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ICHPTR(3,1)														1		
ICHPTR(4,1)	1	1	1	1	1	1	1	1	1						1	1
ICHPTR(4,2)		1	1													1
ICHPTR(4,3)	1	1	1												1	1
ICHPTR(4,4)				1	1	1	1	1	1						1	1
ICHPTR(4,5)				1	1	1	1	1	1						1	1
ICHPTR(4,6)				1	1			1	1						1	
ICHPTR(5,1)													1			1
ICHPTR(6,1)	1*	1*	1*													1
ICHPTR(6,2)				1*	1*	1*	1*	1*	1*							
ICHPTR(6,3)													1			
ICHPTR(7,1)	1	1	1												1	1
ICHPTR(7,2)						1	1									1
ICHPTR(7,3)						1*	1*									
ICHPTR(7,4)						1	1									1
ICHPTR(7,5)				1	1			1	1						1	
ITYP(1)	1	3	2	1	1	3	2	1	1						1	2
ITYP(2)	1		2	2	4	5			7	6	3	8	9*		4	2
NPUNCH				1*	1*	1*	1*	1*	1*	2*	1*		1*			

*Optional

BLOCK DATA. For multiple case runs, header cards for subsequent cases follow the NAMELIST input. Note that the NAMELIST input is read only on the first case run and, therefore, if a multiple case run is made, the NAMELIST input values will be the same for all cases. In addition, if a multiple case run is made, case numbers must be input in increasing order (i.e., Case 47 should be input before Case 55). The disk file input contains 297 different cases of aerodynamic coefficients for the system matrix F. (See Appendix A).

All input to DIGADAPT is read by the main executive. Figure 2.1-1 shows card input data for a multiple case run and Tables 2.1-1 to 2.1-10 give a description of all DIGADAPT input. Table 2.1-1 describes the header card and Tables 2.1-2 to 2.1-10 to describe namelists NAM1 to NAM9 respectively.

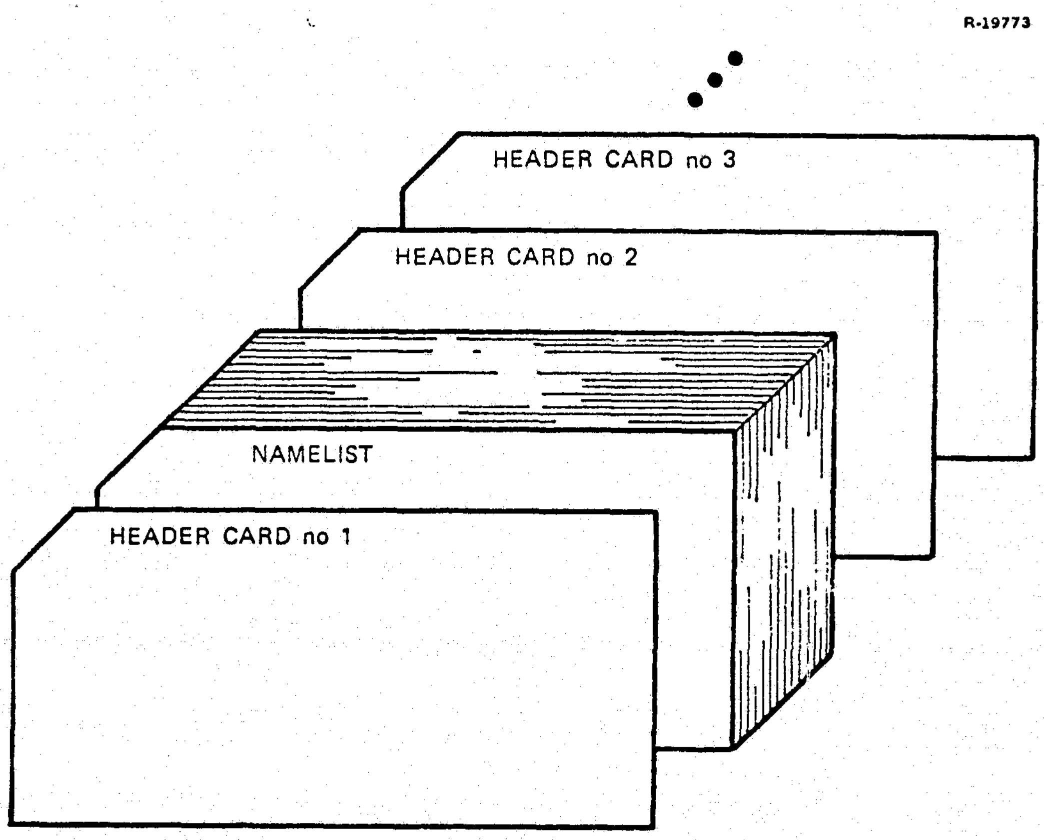


Figure 2.1-1 Input on Logical Unit 5 (TAPE5) for DIGADAPT

TABLE 2.1-1

HEADER CARD INPUT TO DIGADAPT

VARIABLE	DESCRIPTION	UNITS	TYPE*
L	Number of observations	-	I col. 1-5
M	Number of controls	-	I col. 6-10
N	Number of states	-	I col. 11-15
NCASE	Case number from 1 to 297 to determine which set of aerodynamic coefficients to use, see Appendix A	-	I col. 16-20
NSIGMA	Flag =0, if SIGMA(I), I=1 to 16 is to be computed in QRCOMP =1, if SIGMA(I), I=1 to 16 has been input via NAMELIST	-	I col. 21-25
NDOT	Flag =0, if DBDOT, DCDOT, DSDOT DRDOT are to be used to compute the diagonal values of R matrix =1, if diagonal values of R matrix are to be read in via NAMELIST	-	I col. 26-30
NPUNCH	Flag =0, no punched output -1, punched output for SCHED	-	I col. 31-35
NCURV	Flag - set to zero - not used	-	I col. 36-40

*I = represents a right justified integer value.

TABLE 2.1-2

NAM1 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
ICHPTR	7 x 6 array of flags used to turn on (=1) and off (=0) the given sections of each chapter where I = chapter and J = section	-	I	42 * 0

TABLE 2.1-3

NAM2 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE*	DEFAULT
ROLRAT	Maximum roll rate	deg/sec	R	-25
PITRAT	Maximum pitch rate	deg/sec	R	-25
YAWRAT	Maximum yaw rate	deg/sec	R	-25
DIFCOL	Maximum differential collective = pitch control	in	R	-6.5
COLLEC	Maximum collective = vertical velocity control	in	R	-4.56
CYCLIC	Maximum cyclic = roll control	in	R	-3.6
DIFCYC	Maximum differential cyclic = yaw control	in	R	-4.18
PITANG	Maximum pitch angle	deg	R	-10
ROLANG	Maximum roll angle	deg	R	-10
HEDANG	Maximum yaw angle	deg	R	-10
T1	If ITYP(1) = 2, then weighting on integrators for velocity control system - X position If ITYP(1) = 3, then weighting on integrators for attitude system - pitch integral	deg-sec	R	1
T2	If ITYP(1) = 2, then weighting on integrators for velocity control system - y-position	ft	R	1
	If ITYP(1) = 3, then weighting on integrators for attitude system - roll integral	deg-sec	R	
T3	If ITYP(1) = 2, then weighting on integrators for velocity control system - z-position	ft	R	2
	If ITYP(1) = 3, then weighting on integrators for attitude system - yaw integral	deg-sec	R	
T4	If ITYP(1) = 2, then weighting on integrators for velocity control system - yaw integral	deg-sec	R	1
	If ITYP(1) = 3, then weighting on integrators for attitude system - z-position	ft	R	

*R represents a real variable

TABLE 2.1-4

NAM3 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
SIGMA	16 vector - values used in computing diagonal values of Q matrix (see subroutine QRCOMP)	-	R	-
KFLAG	9 vector flag to determine which states are assumed to be measured in discrete form	-	I	9 * 1

TABLE 2.1-5

NAM4 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
IXX	Second moment of inertia about x-axis = I_{xx}	slug-ft^2	R	4.04E4
IYY	Second moment of inertia about y-axis = I_{yy}	slug-ft^2	R	2.19E5
IZZ	Second moment of inertia about z-axis = I_{zz}	slug-ft^2	R	2.06E5
IXZ	Product of inertia = I_{xz}	slug-ft^2	R	1.59E4

TABLE 2.1-6

NAM5 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
DBDOT	Pitch control rate = $\dot{\delta}_B$	in/sec	R	-2.0
DCDOT	Vertical velocity control rate = $\dot{\delta}_C$	in/sec	R	-2.0
DSDOT	Roll control rate = $\dot{\delta}_S$	in/sec	R	-2.0
DRDOT	Yaw control rate = $\dot{\delta}_R$	in/sec	R	-2.0
UDOT	U acceleration = $\dot{u}$	ft/sec^2	R	-0.2
VDOT	V acceleration = $\dot{v}$	ft/sec^2	R	-0.2
WDOT	W acceleration = $\dot{w}$	ft/sec^2	R	-0.2
UMAX	U velocity = u	ft/sec	R	-1.5
VMAX	V velocity = v	ft/sec	R	-0.5
WMAX	W velocity = w	ft/sec	R	-0.2

TABLE 2.1-7

NAM6 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
QBAR	Noise covariance matrix dimension (N+3) x (N+3). Note singly subscripted*	-	R	256 * 0.0
SVB	13 state vector bound	-	R	2.,2.,2.,5.,5.,5.,.5,.5,.5.,1.,1.,1.,1.
SIGMAD	Covariance matrix for process noise (9x9)	-	R	81 * 0.0
THETAD	Covariance matrix for observation noise (Lx9) - Note singly subscripted*	-	R	290 * 0.0

*To input a value in the (I,J)th position of the matrix, use the following algebraic conversion to compute appropriate subscript for singly subscripted variables.

$$K = (J-1)* N + I$$

where N is the number of rows of the matrix

TABLE 2.1-8

NAM7 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
F	(NxN) system matrix - must use single subscript*	-	R	-
G	(NxM) control matrix - must use single subscript*	-	R	-
Q	(NxN) state weighting matrix - must use single subscript*	-	R	-
R	(MxM) control weighting matrix - must use single subscript*	-	R	-
T	Interval sampling time	sec	R	.0833333

*To input a value in the (I,J)th position of the matrix use the following algrbraic conversion compute appropriate subscript for singly subscripted variables

K = (J-1) * N + I

where N is the number of rows of the matrix

TABLE 2.1-9

NAM8 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
UMAG1	Is a 17 x 4 array of steady-state values of the system where I corresponds to the ith state and J corresponds to the jth run	-	R	68 * 0.0
UMAG2	Is a 4 x 4 guidance command array where I corresponds to the ith state and J corresponds to the jth run	-	R	16 * 0.0
DXZERO	Is a 17 x 4 array of initial conditions on the ith states, controls and integrators and to the jth run	-	R	68 * 0.0

TABLE 2.1-10

NAM9 INPUT

VARIABLE	DESCRIPTION	UNITS	TYPE	DEFAULT
TRDT	Iteration interval for continuous-time propagation of Riccati equation	sec	R	0.5
SMPLT	Iteration interval for sampling time determination	sec	R	.001
MAX	Maximum number of steps allowed for propagation of Riccati equation	-	I	30
MAXD	Maximum number of steps allowed for propagation of discrete Riccati equation	-	I	50
EPS	Convergence criterion for both discrete and continuous designs	-	R	.01
IMAX	Maximum number of time steps for simulation	-	I	200
IRUN	Number of simulation runs	-	I	1
DELT	Iteration interval for continuous simulation	sec	R	.18
ITYP(1)	controller type flag ITYP(1) = 1 dynamic compensation achieved by control-rate weight ITYP(1) > 1 dynamic compensation achieved by PII ITYP(1) = 2 velocity control system ITYP(1) = 3 attitude control system	-	I	2
ITYP(2)	ITYP(2) = 1 UMAG1 is used in Section 7.1 ITYP(2) = 2 UMAG1 is not used in Section 7.1 ITYP(2) = 3 punches out nominal control and angle position ITYP(2) = 4 or 5 determine and simulate discrete PI controller ITYP(2) = 4 attitude control system ITYP(2) = 5 velocity control system ITYP(2) = 6 or 7 determine and simulate discrete control-rate system ITYP(2) = 6 attitude control system ITYP(2) = 7 velocity control system ITYP(2) = 8 punch out F 6x6 and G 6x4 for trim in AERO ITYP(2) = 9 punch discrete control matrix	-	I	1

2.2 DIGADAPT OUTPUT

The DIGADAPT output takes two forms: printed reports on Fortran logical unit 6 (TAPE6) and punched output on Fortran logical unit 7 (TAPE7). Table 2.2-1 contains a summary of the printout from the program including a paraphrase, a description, the variables and the subroutine associated with the report. Table 2.2-2 contains a list of all the variables that can be punched for the SCHED regression analysis. It includes a description of the variables, the format of the punched output, the subroutine generating the output and the control flag(s) necessary to turn the punch command "on".

TABLE 2.2-1

OUTPUT REPORTS FROM DIGADAPT

REPORT	DESCRIPTION - VARIABLE	SUBROUTINE
CASE NUMBER =	Give particular case of aerodynamic coefficients chosen - will print out only in multiple case run starting with second case - NCASE	MAIN
DIGADAPT	Abstract of the program	MAIN
INPUT PARAMETERS	The flags, defaults and/or override values that will be used in all cases to follow - ICHPTR and other parameters	MAIN
WEIGHTING ON STATE	Values used in computing the Q matrix	MAIN
WEIGHTING ON INTEGRATORS FOR VELOCITY-COMMAND SYSTEM	x position, y position, z position, yaw integral - T1, T2, T3, T4	MAIN
TRIM VALES	δ_B, δ_C, δ_S, δ_R, θ, ϕ	MAIN
NORMAL EOJ	Job has terminated as expected	MAIN
SYSTEM MATRIX F	N x N system matrix F	AERO
INITIAL VELOCITIES	u_0, v_0, w_0	AERO
XDOT, YDOT, ZDOT, THETA PHE, PSI	$\dot{x}$, $\dot{y}$, $\dot{z}$, θ, ϕ, ψ	FGCOMP
PO, QO, RO	Nominal control	FGCOMP
NOISE COVARIANCE MATRIX	QBAR	
STATE VECTOR BOUND	SVB	SID
SAMPLING TIME	Sampling time T as determined by SAMPLE	SID
STATE POSITION WHOSE BOUND WAS EXCEEDED	The K^{th} diagonal value of covariance matrix which was greater than state vector bound	SID
TIME HISTORY OF DIAGONAL ELEMENTS COVARIANCE	Beginning subroutine SAMPLE with value of iteration interval for sampling time - the results to follow are U, V, W, P, Q, R, THETA, PHI, PSI TIME	SAMPLE
STATE WEIGHTING MATRIX	NxN Q matrix	GAIN
CONTROL WEIGHTING MATRIX	MxM R matrix	GAIN
OPTIMAL GAIN MATRIX	MxN C matrix	GAIN

TABLE 2.2-1

OUTPUT REPORTS FROM DIGADAPT (Continued)

REPORT	DESCRIPTION - VARIABLE	SUBROUTINE
RICCATI SOLUTION EQUATION	NxN X matrix	GAIN
DISCRETE RICCATI EQUATION	MxN XD matrix	GAIN
DISCRETE OPTIMAL GAIN	MxN CD matrix	GAIN
TRANSFORMATION MATRIX FOR PI CONTROLLER	NRxP Q matrix where NR=N-M-P	GAIN
FEEDBACK GAIN MATRIX FOR PI CONTROLLER	MxM R matrix	GAIN
FEEDBACK GAIN MATRIX FOR PI CONTROLLER	MxNR QHAT matrix	GAIN
PROPAGATION OF CONTINUOUS TIME RICCATI EQUATION	TRDT - time step for iteration, MAX - maximum of number of iterations allowed EPS - convergence criterion NUM - number of iterations computed	RICTCT
BEGIN SOLUTION OF ALGEBRATIC RICCATI EQUATION	Use Kleinman scheme with convergence criterion 10^{-5}	TMRIC
ALGEBRAIC RICCATI SOLUTION IN	Give number of iterations to satisfy convergence criterion	TMRIC
RICCATI SOLUTION IS PSD	Riccati solution is positive semi-definite	TMRIC
RICCATI NOT CONVERGED	Computation continues as if convergence occurred - value of solution matrix x is output	TMRIC
RICCATI BLOWUP	Solution X has diverged	TMRIC
LINEAR EQN ALGORITHM NON-CONVERGENT	Gives number of iterations, IT, attempted to solve linearized Riccati equation	MLINEQ
NO RICCATI SOLUTION	Gives number of iterations, I, attempted to solve Riccati equation	TDREQ
PROPAGATION OF DISCRETE-TIME RICCATI EQUATION	MAXD - maximum number of iterations allowed EPS - convergence criterion I - number of iterations	TDREQ
BEGIN SOLUTION OF STEADY-STATE DISCRETE MATRIX RICCATI EQUATION	Use Kleinman scheme with convergence criterion of 10^{-5}	DRIC

TABLE 2.2-1

OUTPUT REPORTS FROM DIGADAPT (Continued)

REPORT	DESCRIPTION - VARIABLE	SUBROUTINE
STEADY-STATE DISCRETE RICCATI SOLN	Gives number of iteration, IT, performed	DRIC
RICCATI SOLUTION IS PSD	Riccati solution is positive semi-definite	DRIC
RICCATI NON-CONVERGENT	Computation continues despite a non-convergent solution	DRIC
RICCATI BLOWUP	Solution diverging	DRIC
STEADY-STATE MATRIX FOR GUIDANCE SYSTEM S12	DUM1	PID
STEADY CONTROL MATRIX FOR A CONTROL SYSTEM S22	DUM3	PID
COVARIANCE MATRIX FOR PROCESS NOISE	SIGMAD	EOD
COVARIANCE MATRIX FOR OBSERVATION NOISE	THETAD	EOD
OBSERVATION MATRIX FOR DISCRETE SYSTEM	MHAT	EOD
RICCATI EQUATION STEADY STATE SOLUTION	XD	EOD
KALMAN GAIN FILTER MATRIX	DT	EOD
DISCRETE CONTROL MATRIX	XMLIH	EOD
EIGENVALUES OF ACL	Closed-loop system	EIGEN
EIGENVALUES OF FCL	Closed-loop discrete Kalman filter system	EIGEN
EIGENVALUES	Printout eigenvalues, period, damping ratio, amplitude and frequency	FREE
SYMBOL MEANING AND UNITS OF SIMULATION	Gives name of symbols used in time history calculation	TIME
STEADY-STATE VALUES	UMAG1	TIME
GUIDANCE COMMAND MATRIX	UMAG2	TIME
CONTINUOUS-TIME SIMULATION PII	If ITYP(1) equals 2	TIMHIS
CONTINUOUS-TIME SIMULATION OF DYNAMIC CONTROLLER	If ITYP(1) equals 1	TIMHIS

TABLE 2.2-1

OUTPUT REPORTS FROM DIGADAPT (Continued)

REPORT	DESCRIPTION - VARIABLE	SUBROUTINE
WRITE OUT TIME HISTORIES U,V,W,P,Q,R, THETA,PHI PSI,DELTA-B, DELTA-S, DELTA-R,V1,V2,V3,V4, DXDOT,DYDOT,DZDOT,U1,U2, U3, UDOT,VDOT,WDOT, DELBDOT, DELCOT,DELSDOT, TIME	PX matrix is written after each run (maximum of 4) is completed	TIMHIS
STEADY STATE HAS NOT BEEN ATTAINED	Gives i^{th} variable of PX for which absolute difference is greater than 10^{-4}.	STEADY
STEADY STATE	Writes out last time step value for PX matrix, SS, on the L^{th} component.	STEADY
PEAK VALUE	Writes out maximum value for i^{th} component of PX=PV	STEADY
TIME TO REACH NINETY PERCENT OF STEADY STATE	Time to reach 90% of steady state - T90	STEADY
TIME TO REACH A FIVE PERCENT BOUND OF STEADY STATE	T05	STEADY
DISCRETE SIMULATION OF PII INTEGRATOR	Begin subroutine DTMHIS	DTMHIS
WRITE OUT TIME HISTORIES (SAME AS TIMHIS)	PX is written after each run	DTMHIS
DISCRETE SIMULATION OF PI	ITYP (2) equals 4 or 5	PIDHIS
DISCRETE SIMULATION OF DYNAMIC CONTROLLER	ITYP (2) equals 6 or 7	PIDHIS
WRITE OUT TIME HISTORIES	PX is written after each run	PIDHIS
Holerith title	In print matrix routine holerith title is printed along with number of rows and columns	PRMR
FATAL ERROR - SINGULARITY DISCOVERED	The propagation of covariance matrix has become singular - W1	PROP
SINGULARITY DISCOVERED	Singularity has been found in Kalman filter update - W2	SMPL
FATAL ERROR - MEASUREMENT NOISE MATRIX IS SINGULAR	R	RSET

TABLE 2.2-2

PUNCHED OUTPUT FROM DIGADAPT

VARIABLE	DESCRIPTION	FORMAT	SUBROUTINE	CONTROLLING* FLAG
F	Upper 6x6 matrix of aircraft system matrix used in dynamic trim computation and the upper 4x6 pseudo-inverse of aircraft control system matrix used in dynamic trim computation	6E12.5	AERO	ITYP(2)=8
NCASE	Case number corresponding to aerodynamic coefficients	I5	FGCOMP	NPUNCH=1
XDOT,YDOT ZDOT,THETA PHE,PSI	$\dot{x}$, $\dot{y}$, $\dot{z}$, θ, ϕ, ψ	6E12.5	FGCOMP	ICHPTR(2,1)=1
PSIDOT	$\dot{\psi}$	E12.5	FGCOMP	ICHPTR(2,1)=1
PO,QO,RO	Nominal control	3E12.5	FGCOMP	ITYP(2)=3 ICHPTR(2,1)=1
UO,VO,WO THETA,PHE, PSI	u_0, v_0, w_0, θ, ϕ, ψ	6E12.5	FGCOMP	ICHPTR(2,1)=1
C	mxn optimal gain matrix	6E12.5	GAIN	ICHPTR(4,3)=1
CD	mxn discrete optimal gain matrix	6E12.5	GAIN	ICHPTR(4,5)=1
R	Gain matrix for delayed error state	6E12.5	GAIN	ICHPTR(4,6)=1
RHAT	Gain matrix for current error state	6E12.5	GAIN	ICHPTR(4,6)=1
QHAT	State feedback gain matrix	6E12.5	GAIN	ICHPTR(4,6)=1
DT	(9xL) Kalman filter gain matrix	6E12.5	EOD	ICHPTR(5,1)=1
FWORK	Discrete 9x9 discrete system matrix	6E12.5	EOD	ICHPTR(5,1)=1 ITYP(2)=9
XMLIH	Discrete 9xm control matrix	6E12.5	EOD	ICHPTR(5,1)=1 ITYP(2)=9

*In addition to the controlling flags listed NPUNCH must be set to one in the header input card for punched output.

3. DETAILED DESCRIPTION OF DIGADAPT

DIGADAPT is comprised of seven program chapters: the main executive, system matrix computation, sampling interval determination, linear-optimal gain design, calculation of eigenvalues, and time history simulations. This chapter discusses the significant subroutines of DIGADAPT and provides flowcharts and tables of input and output. Figures 3-1 to 3-3 gives a structural overview of the relationships between routines in outline form. A brief discussion of the service utilities also is included.

3.1 MAIN EXECUTIVE

The main program reads the data for a given run and calls all subsequent program chapters. On the first case for a given execution of DIGADAPT, the program reads a header card and then reads data via NAMELIST to override the BLOCK DATA initialization. On subsequent cases of a multiple case run, only the header card is needed (see Fig. 2.1-1). The program terminates when the last case input on Fortran logical unit 5 has been processed.

The output from the main executive consists of an abstract of the DIGADAPT program along with a listing of initialization parameters (printed out for the first case only). Figure 3.1-1 gives an overview of the main executive.

R-21328

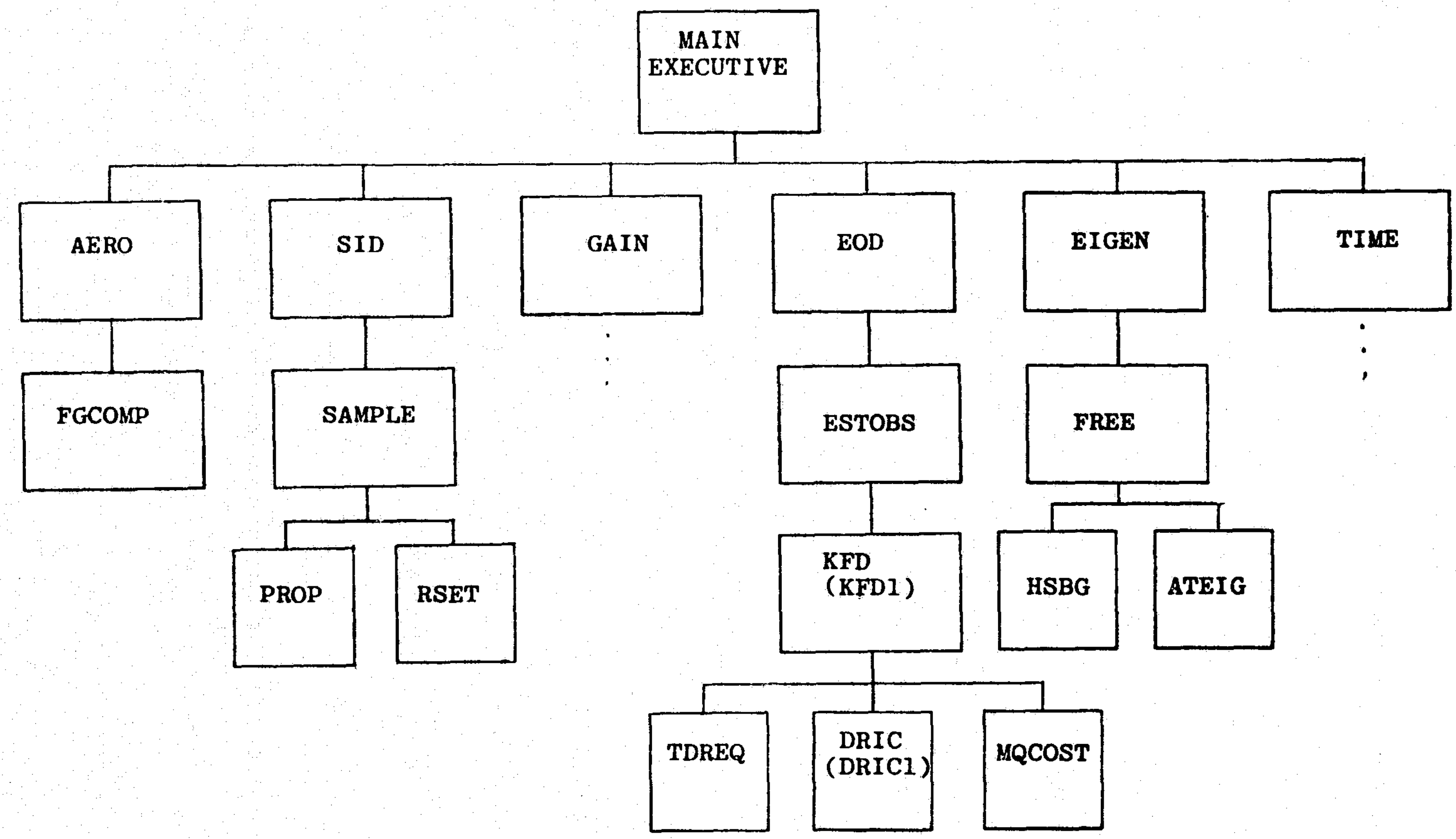


Figure 3-1   Overview of Program DIGADAPT

R-21329

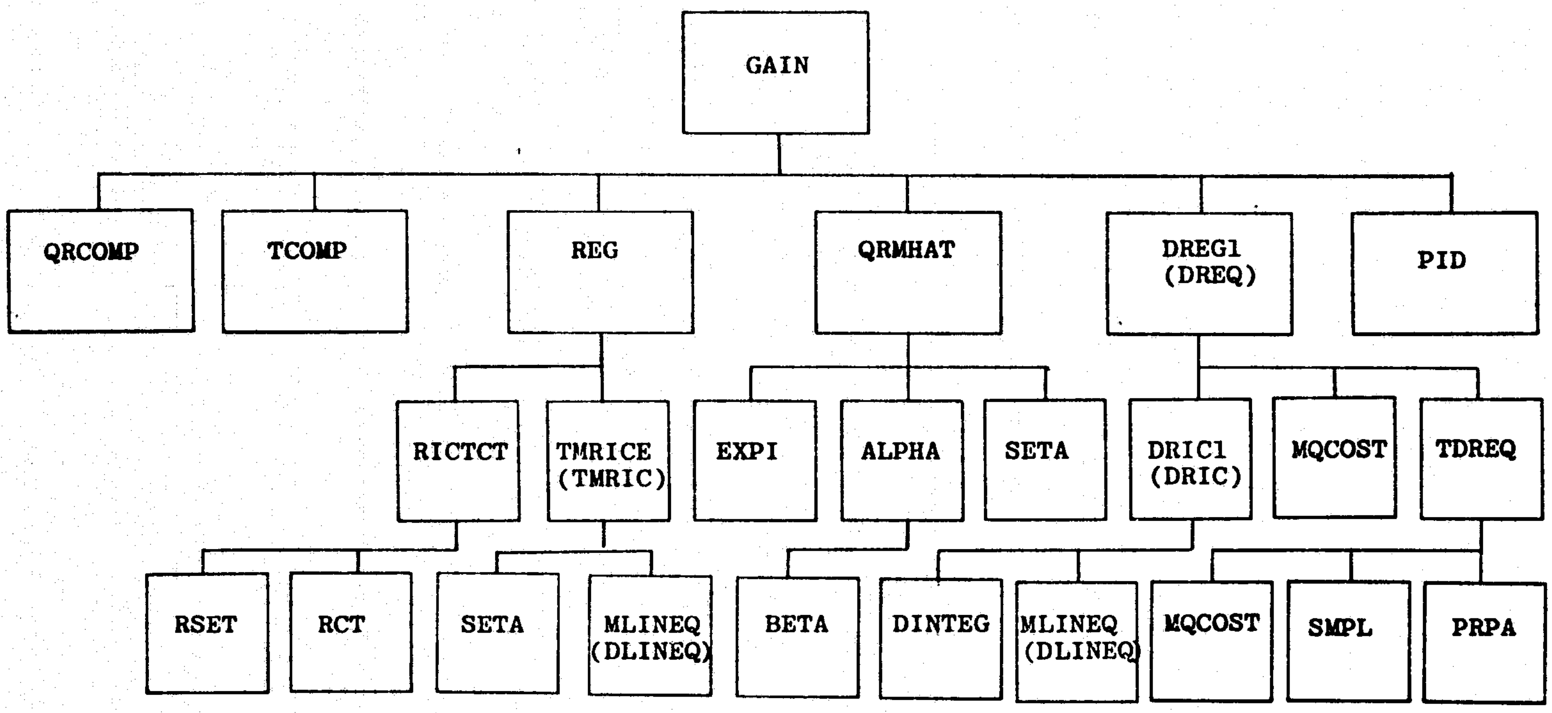
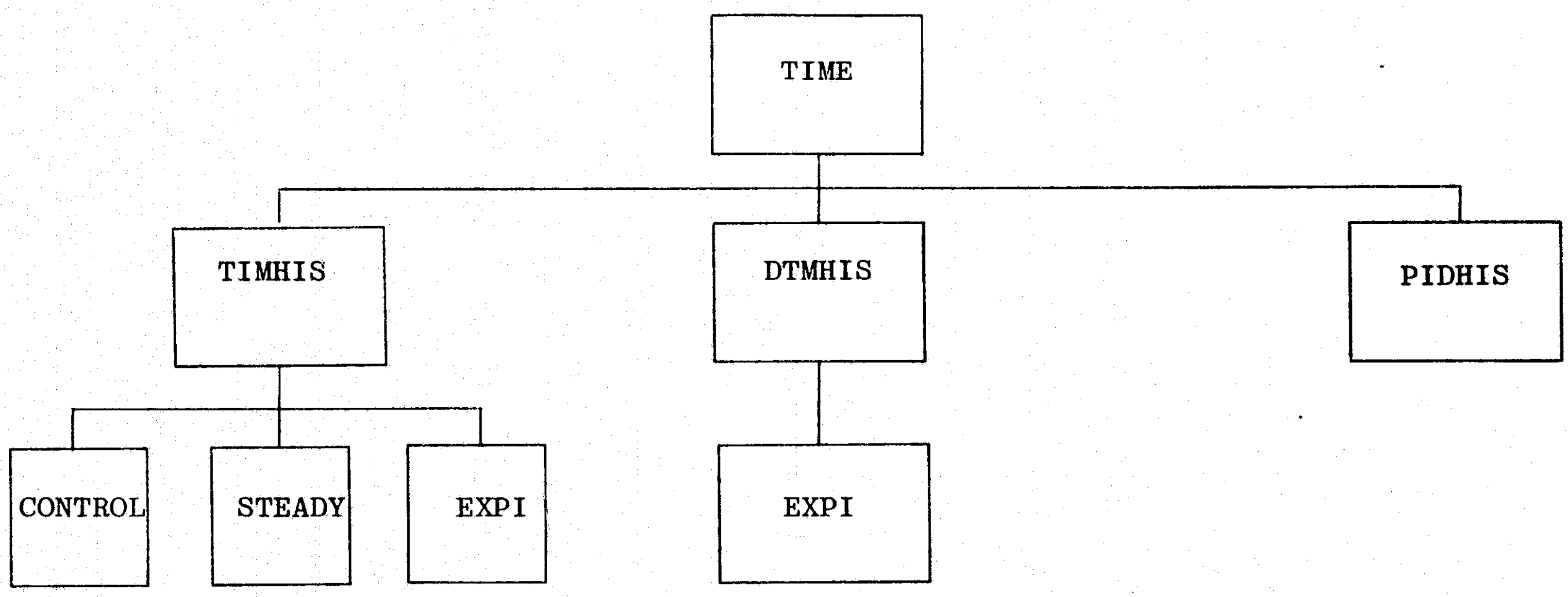


Figure 3-2 Overview of Subroutine GAIN



SERVICE ROUTINES

ADD	SUBT	MULT	ZERO	EQAT	JUXR	JUXC	INV	DOT
PRMR	EXPJ	TRNP	TMLT	SMLT	NORM	MNUS	IDEN	BLOCK DATA

Figure 3-3   Subroutine TIME and Service Routines

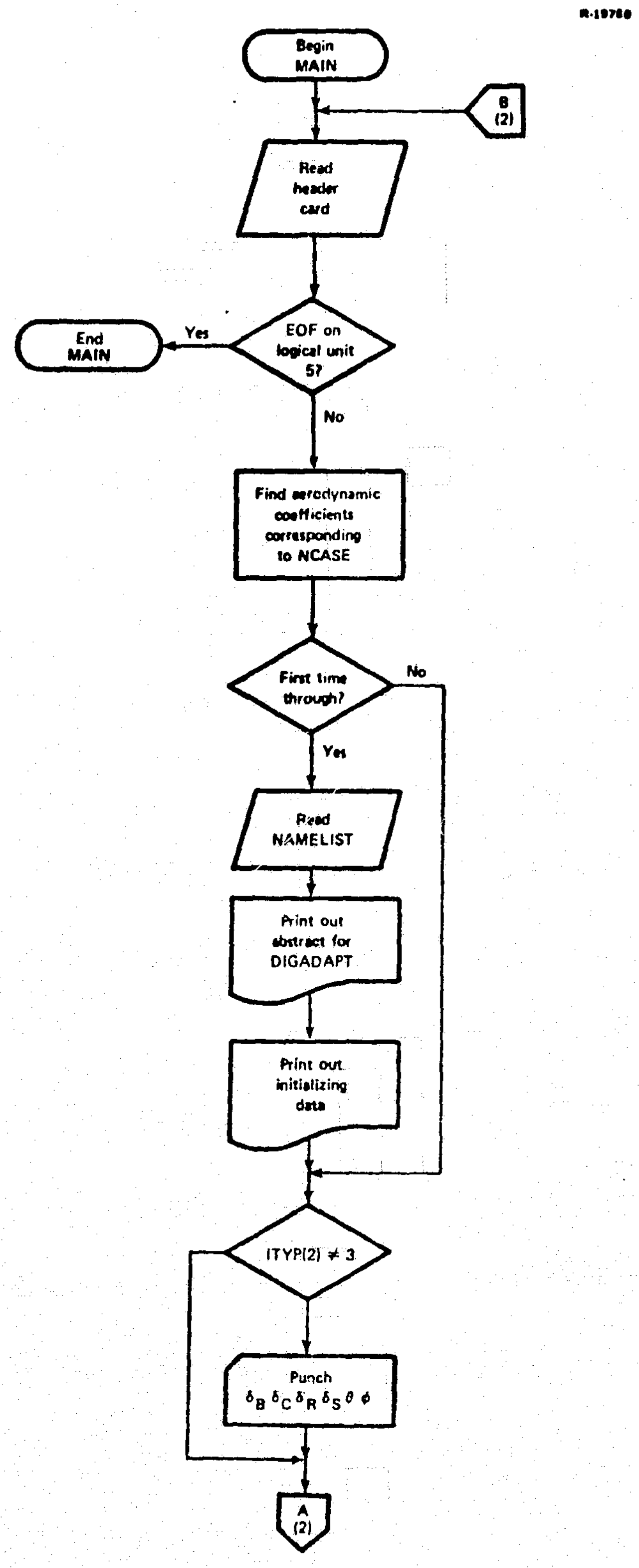


Figure 3.1-1 Flowchart of DIGADAPT Main Executive (Segment 1 of 2)

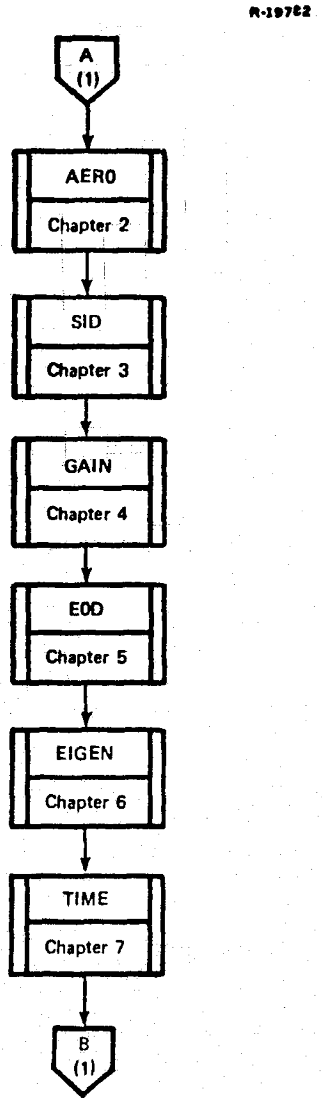


Figure 3.1-1 Flowchart of Main Executive (Segment 2 of 2)

3.2 SYSTEM MATRIX COMPUTATION

The purpose of subroutines AERO and FGCOMP is to place the stored aerodynamic coefficients into the correct positions of the system matrix F. The ordering of the states for this program chapter is as shown in Table 3.2-1.

TABLE 3.2-1

ORDER OF STATES

STATE VARIABLE	DESCRIPTION
u	BODY-AXIS VELOCITY - POSITIVE FORWARD
v	BODY-AXIS VELOCITY - POSITIVE RIGHT
w	BODY-AXIS VELOCITY - POSITIVE DOWN
θ	INERTIAL-BODY PITCH EULER ANGLE
ϕ	INERTIAL-BODY ROLL EULER ANGLE
ψ	INERTIAL-BODY YAW EULER ANGLE
p	BODY-AXIS ROLL RATE
q	BODY-AXIS PITCH RATE
r	BODY-AXIS YAW RATE
δ_B	DIFFERENTIAL COLLECTIVE
δ_C	GANG COLLECTIVE
δ_S	GANG CYCLIC
δ_R	DIFFERENTIAL CYCLIC

Appendix A describes how the disk storage for the aerodynamic coefficients is prepared and describes how to choose appropriate calls. This disk file must be created before running DIGADAPT. Figure 3.2-1 gives the flowchart of AERO, and Table 3.2-2 gives the inputs and outputs for FGCOMP.

3.3 SAMPLING INTERVAL DETERMINATION

The purpose of subroutines SID and SAMPLE is to aid in the determination of the sampling time of a continuous, linear-time-invariant system. In this program chapter, the system matrix F has three added gust states. The sampling time is found by propagating the covariance matrix equation

$$\dot{X}(t) = FX(t) + X(t)F^T + Q$$

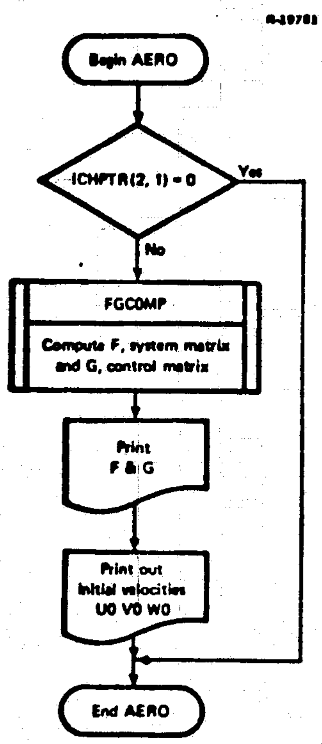


Figure 3.2-1 Flowchart of Subroutine AERO

TABLE 3.2-2

INTERFACE TO ROUTINE FGCOMP

INPUTS		
N	=	number of states
M	=	number of controls
NCASE	=	case number
NPUNCH	=	0 if no punch output is desired 1 if punch output is desired
OUTPUTS		
F	=	NxN state matrix
G	=	NxM control matrix
U0	=	body velocity along body x-axis
V0	=	body velocity along body y-axis
W0	=	body velocity along body z-axis
XDOT	=	body velocity along inertial x-axis
YDOT	=	body velocity along inertial y-axis
ZDOT	=	body velocity along inertial z-axis
ANGLE	=	3x3 transformation matrix from inertial coordinates to body coordinates
ANGLI	=	3x3 transformation matrix from body to inertial coordinates

where F is the augmented system matrix, X is the state covariance* Q is the covariance matrix of the system excitation noise, and X(0) = 0. The sampling time is determined when any diagonal element of X exceeds the input state vector bounds (SVB). Figure 3.3-1 shows how this is accomplished and Table 3.3-1 gives the interface of SAMPLE with SID.

R-19783

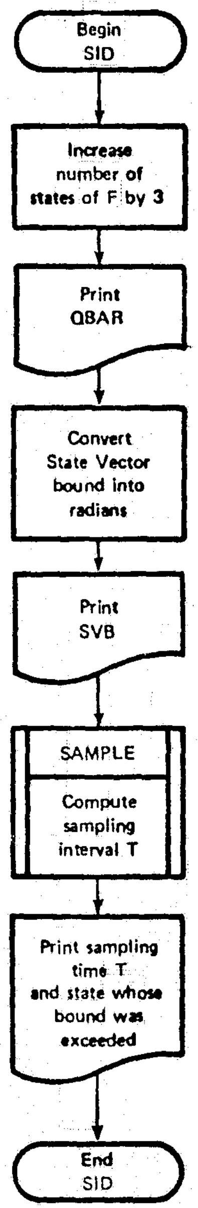


Figure 3.3-1 Flowchart of Subroutine SID

*X is equivalent to engineering notation P.

TABLE 3.3-1

INTERFACE TO ROUTINE SAMPLE

INPUTS	
N =	dimension of system matrix
SVB =	state N-vector bound
QBAR =	NxN noise covariance matrix
F =	NxN system matrix
OUTPUTS	
T =	sampling time
K =	the number of the state whose state vector bound was exceeded

3.4 LINEAR-OPTIMAL GAINS

The purpose of this program chapter is to design various linear-optimal gains (Fig. 3.4-1). GAIN is divided into continuous and discrete modes of design, with and without integrators.

The following equations show the controllers which can be simulated, where the K's shown are continuous gains, the C's are discrete gains, and Φ and Γ are discrete equivalents of F and G.

<u>Continuous Time Dynamic Controller</u>

$$\dot{\underline{x}}(t) = F\underline{x}(t) + G\underline{u}(t)$$
$$\dot{\underline{u}}(t) = K_2\underline{u}(t) + K_1\underline{x}(t) + L\underline{y}_d$$

<u>Continuous Time PII Controller</u>

$$\dot{\underline{x}}(t) = F\underline{x}(t) + G\underline{u}(t)$$
$$\dot{\underline{u}}(t) = K_3\underline{u}(t) + K_4\underline{x}(t) + K_5\underline{y}(t)$$
$$\dot{\underline{y}}(t) = T\underline{x}(t) - \underline{y}_d$$

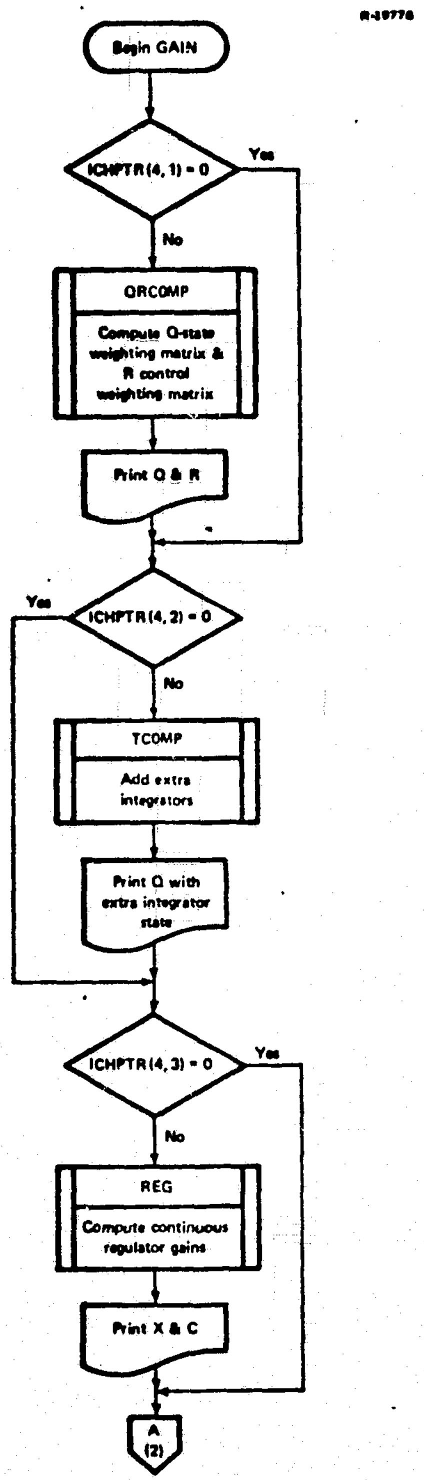


Figure 3.4-1 Flowchart of Subroutine GAIN (Segment 1 of 3)

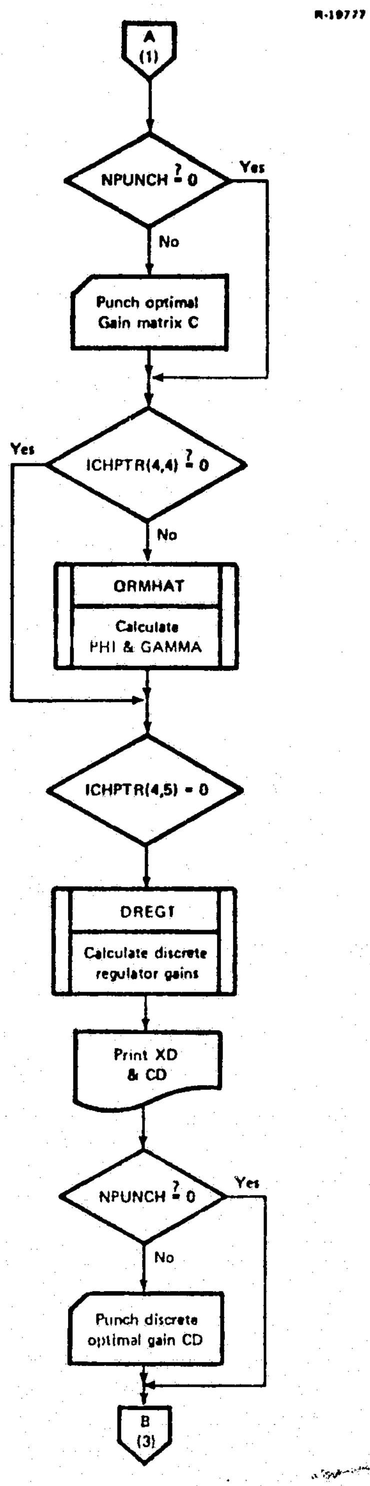


Figure 3.4-1 Flowchart of Subroutine GAIN (Segment 2 of 3)

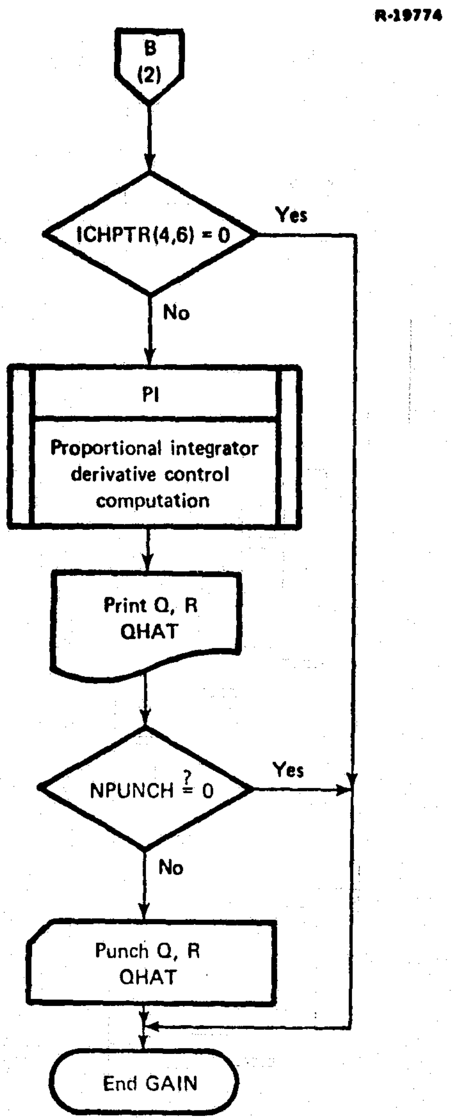


Figure 3.4-1 Flowchart of Subroutine GAIN (Segment 3 of 3)

Discrete PII Controller

$$\begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi & \Gamma & 0 \\ 0 & I & 0 \\ \Phi_1 & \Phi_2 & I \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_k + \begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} \begin{bmatrix} -C_1 & -C_2 & -C_3 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_k - \begin{bmatrix} 0 \\ 0 \\ \Delta t I \end{bmatrix} \underline{y}_d$$

Simplified Discrete PII Controller

$$\begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_{k+1} = \begin{bmatrix} \Phi & \Gamma & 0 \\ 0 & I & 0 \\ \Delta t T & 0 & I \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_{k} + \begin{bmatrix} 0 \\ \Delta t I \\ 0 \end{bmatrix} \begin{bmatrix} -C_1 & -C_2 & -C_3 \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{y} \end{bmatrix}_{k}$$

$$- \begin{bmatrix} 0 \\ 0 \\ \Delta t I \end{bmatrix} \underline{y}_d$$

Simplified Discrete Dynamic Controller

$$\underline{x}_{k+1} = \underline{x}_k + \Gamma \underline{u}_k$$
$$\underline{u}_{k+1} = \underline{u}_k + C_r \underline{u}_k + C_5(\underline{y}_d - T\underline{x}_k) + C_6(\underline{x}_k)$$

Simplified Discrete PI Controller

$$\underline{x}_{k+1} = \underline{x}_k + \Gamma \underline{u}_k$$
$$\underline{u}_k = \underline{u}_{k-1} + C_7(\underline{y}_d - T\underline{x}_{k-1}) + C_8(\underline{y}_d - T\underline{x}_k) + C_9(\underline{x}_k)$$

3.4.1 Q and R Computation

The purpose of QRCOMP is to compute the state weighting matrix Q and the control weighting matrix R. Note that if NSIGMA (input variable of header card) is set to one then the value of SIGMA(I), I = 1 to 16 will default to the BLOCK DATA value; alternatively, NAMELIST may be used to override the default value. Note that the BLOCK DATA values default to negative values; hence, the program sets SIGMA(I), I = 1 to 16 to zero. The SIGMA values are then used to compute the diagonal terms of the Q matrix. The diagonal values of the R matrix are computed by taking the inverse squared of the appropriated maximum control rate. Table 3.4-1 gives the interface between QRCOMP and GAIN.

TABLE 3.4-1

INTERFACE TO ROUTINE QRCOMP

```
INPUTS
     N = number of states
     M = number of controls
NSIGMA = 0 if SIGMAS are to be calculated
         1 if SIGMAS are to be input
  NDOT = 0 if diagonal values of the R matrix
           are to be computed
         1 if diagonal values of the R matrix
           are to be input
OUTPUTS
     Q = NxN state weighting matrix
     R = MxM control weighting matrix
```

3.4.2 Guidance Commands

The purpose of TCOMP is to augment the states of the system matrix, F, the control matrix, G, and the state weighting matrix Q. The added states have the form $\dot{\underline{y}} = T\underline{x}$ - UMAG2 where UMAG2 are the guidance commands, and T is the transformation matrix converting $\underline{x}$ to UMAG2 states. Table 3.4-2 shows the interface between TCOMP and GAIN.

TABLE 3.4-2

INTERFACE TO ROUTINE TCOMP

```
INPUTS
     N = number of states without T
     M = number of controls
     F = NxN state matrix
     F = NxM control matrix
     Q = NxN state weighting matrix
  ITYP = type of guidance used
    T1)
    T2}= weighting on extra states for attitude system
    T3
    T4)
OUTPUTS
     N = number of states with T
     P = number of states in extra states added to system
     F = NxN new state matrix
     G = NxN new control matrix
     Q = NxN new state weighting matrix
 ANGLE = 3x3 transformation matrix from inertial
         coordinates to body coordinates
 ANGLI = 3x3 transformation matrix from body to
         inertial coordinates
```

3.4.3 Continuous Regulator Gains

REG designs an optimal linear regulator by finding the steady-state solution to the Riccati equation

$$\dot{P} = PF + F^TP + Q - PGR^{-1} G^TP$$

In addition, the optimal feedback gain K and the closed-loop system matrix ACL are calculated. REG first propagates the Riccati equation (using subroutine RICTCT) from P = 0 until either the maximum number of iterations, MAX, is reached or the convergence criterion, EPS, is satisfied. It then takes the last value P from RICTCT as an initial input guess for TMRICE, an entry point in TMRIC. Table 3.4-3 shows the interface between REG and GAIN.

TABLE 3.4-3

INTERFACE TO ROUTINE REG

INPUTS	
N =	number of states
M =	number of inputs
A =	NxN system matrix F
B =	NxM input matrix G
Q =	NxN state weighting matrix
R =	MxM control weighting matrix
OUTPUTS	
X =	NxN matrix Riccati equation solution
G =	MxN optimal gain matrix
ACL =	NxN optimal closed-loop matrix

The purpose of RICTCT is to solve the Riccati equation

$$-\dot{P}(t) = P(t)F + F^TP(t) + Q - P(t)\, GR^{-1}GP(t)$$

using backward integration (Ref. 1). Consider the equations

$$\frac{d}{dt}\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} F & -GR^{-1}G \\ -Q & -F^{1} \end{bmatrix} \quad , \quad \begin{bmatrix} \underline{x}(t_N) \\ \underline{y}(t_N) \end{bmatrix} = \begin{bmatrix} I \\ P_O \end{bmatrix}$$

If a solution exists on the interval (t, t_1), then the solution has the property that

$$P(t) = Y(t)\, X^{-1}(t)$$

For computer implementation, the system matrix is discretized by

$$\Theta\,(t,\tau) = \exp\begin{bmatrix} F & -GR^{-1}G^{T} \\ -Q & -F^{T} \end{bmatrix}(t-\tau)$$

where $\Theta\,(t,\tau)$ is partitioned into four nxn submatrices

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}$$

If $t-\tau$ is set to a constant value Δt, then the solution for P can proceed as follows:

$$P(t_N) = P_O$$

$$P\big((n-1)\Delta t\big) = \Big[\,\Theta_{21}(\Delta t) + \Theta_{22}(\Delta t)P(n\Delta t)\Big]\Big[\,\Theta_{11}(\Delta t) + \Theta_{12}(\Delta t)\,P(n\Delta t)\Big]^{-1}$$

This computation is implemented by calls to RSET which compute the transition matrices and RCT which solves the Ricatti equation.

Finally, upon leaving RCT, if NUM=MAX, no steady-state solution has been reached; if NUM<MAX a steady-state solution has been reached, where NUM is the number of iterations

performed. In addition the values of EPS, MAX, NUM and TRDT are printed. Then the closed-loop system ACL matrix is computed by $F - GR^{-1}G^TP$. Table 3.4-4 gives the interface between RICTCT and REG.

TABLE 3.4-4

INTERFACE TO ROUTINE RICTCT

INPUTS	
F	= NxN system matrix
G	= NxM control matrix
Q	= NxN state weighting matrix
R	= MxM control weighting matrix
TRDT	= time step of propagation
MAX	= maximum number of time steps allowed
EPS	= criterion for reaching steady state
OUTPUTS	
X	= NxN Riccati solution at time NUM*TRDT
C	= $R^{-1}*G^T*$ X = MxN optional feedback gain
ACL	= F - G*C = NxN closed-loop system

The purpose of TMRIC is to solve the algebraic Riccati equation

$$0 = PF + F^TP + Q - PGR^{-1}GP$$

using Newton's Method. Let P_K, K=0,1,... be the (unique) positive semidefinite solution of the linear equation

$$0 = F_k^TP_k + P_kF_k + Q + L_k^TRL_k$$

where, recursively,

$$L_k = R^{-1}G^TP_{k-1} \qquad k=1,2,...$$
$$F_k = F - GL_k$$

L_0 is chosen so that the matrix $F - GL_0$ has eigenvalues with negative real parts. Thus

$$0 \le P \le P_{k+1} \le P_k \le \ldots P_0 \qquad k=0,1,2,...$$

and

$$\lim_{k\to\infty} P_k = P$$

The above scheme linearizes the Riccati equation and can be shown to have quadratic convergence.

The linearized form of the Riccati equation is solved by subroutine MLINEQ with the following decisions made in TMRIC

- Has the solution converged?
- Has the solution diverged?
- Has the solution not diverged and not converged?

Table 3.4-5 shows the interface between TMRIC and REG.

TABLE 3.4-5

INTERFACE TO ROUTINE TMRIC

INPUTS	
N	= dimension of square arrays
A	= NxN system matrix
S	= NxN matrix = $G*R^{-1}*G^T$
Q	= NxN state weighting matrix
OUTPUTS	
X	= NxN positive-definite solution of matrix Riccati equation
Z	= A-S*X = NxN closed-loop matrix

3.4.4 Calculation of PHI and GAMMA

The purpose of QRMHAT is to discretize a linear-time-invariant continuous system and its associated cost function

$$\dot{\underline{x}} = F\underline{x} + G\underline{u}$$

$$J(u) = \int_0^{\infty} \underline{x}^T Q\underline{x} + \underline{u}^T R\underline{u}\, dt$$

In its discretized form, the first equation becomes

$$\underline{x}_{k+1} = \Phi\underline{x}_k + \Gamma\underline{u}_k$$

Similarly, the second equation becomes

$$J(u) = \sum_{k=0}^{\infty} (\underline{x}_k^T \hat{Q} \underline{x}_k + 2\underline{x}_k^T \hat{M} \underline{u}_k + \underline{u}_k^T \hat{R} \underline{u}_k)$$

Upon discretization, subroutine QRMHAT produces the following results (computer symbols are used on the left, engineering notation on the right):

$$\text{PHI} = \sum_{k=0}^{\infty} \frac{(FT)^k}{k!} = e^{FT}$$

$$\text{DT} = \int_o^T \sum_{k=0}^{\infty} \frac{(F\tau)^k}{k!} d\tau = \int_o^T e^{F\tau} d\tau$$

$$\text{QHAT} = \int_o^T e^{F^T\tau} Q e^{F\tau} d\tau$$

$$\text{MHAT} = \left[\int_o^T e^{F^T s} * Q\left(\int_o^s e^{Fw} dw\right) ds\right] G$$

$$\text{RHAT} = \text{TR} + G^T \int_o^T \left\{\left[\int_o^s e^{F^T w} dw\right] Q \cdot \left[\int_o^s e^{Fw} dw\right] ds\right\} G$$

This computation is completed by EXPI computing PHI, by SETA computing QHAT, and by ALPHA computing RHAT and MHAT. Table 3.4-6 shows the interface between this program and GAIN.

The purpose of MQCOST is to find equivalent matrices FEQ and QEQ so that the optimal regulator problem

$$\dot{\underline{x}} = F\underline{x} + G\underline{u}$$

and

$$J = \int_o^{\infty} (\underline{x}^T Q\underline{x} + 2\underline{x}^T S\underline{u} + \underline{u}^T R\underline{u})\, dt$$

is similar to

TABLE 3.4-6

INTERFACE TO ROUTINE QRMHAT

```
INPUTS
   M = number of controls
   N = number of states
   T = sampling time
   F = NxN state matrix
   G = NxM matrix
   Q = NxM matrix
   R = MxM weighting matrix
OUTPUTS
 PHI = NxN discretized system matrix
  DT = NxM discretized output matrix
QHAT = NxM discretized state cost matrix
MHAT = NxM discretized cross coupling cost matrix
RHAT = MxM discretized control cost matrix
```

$$\dot{\underline{x}} = (FEQ)\underline{x} + G\underline{u}$$

and

$$J = \int_0^\infty (\underline{x}^T(QEQ)\underline{x} + \underline{u}^T R\underline{u})dt$$

Table 3.4-7 shows the interface between MQCOST and TDREQ.

TABLE 3.4-7

INTERFACE TO ROUTINE MQCOST

```
INPUTS
   N = number of states
   M = number of controls
   F = NxN matrix
   Q = NxN matrix
OUTPUTS
 FEQ = NxN equivalent F matrix = F - G*R-1*ST
 QEQ = NxN equivalent Q matrix = Q - S*R-1*ST
```

3.4.5 Discrete Regulator Gains

The purpose of DREQ is to design an optimal linear regulator for the discrete time, linear system

$$\underline{x}_{k+1} = \Phi\underline{x}_k + \Gamma\underline{u}_k$$

according to the quadratic cost function

$$J = \sum_{k=0}^{\infty} (\underline{x}_k^T Q\underline{x}_k + \underline{u}_k^T R\underline{u}_k)$$

The regulator is defined by the optimal control $\underline{u}_k = -K\underline{x}_k$ where

$$K = (R + \Gamma^T P\Gamma)^{-1} \Gamma^T P\Phi$$

and

$$\underline{x}_{k+1} = (\Phi - \Gamma K)\underline{x}_k = (ACL)\underline{x}_k$$

For DREQ the cost function is

$$J = \sum_{k=0}^{\infty} (\underline{x}_k^T Q\underline{x}_k + 2\underline{x}_k^T S\underline{u}_k + \underline{u}_k^T R\underline{u}_k)$$

and

$$K = (R + \Gamma^T P\Gamma)^{-1} (\Gamma^T P\Phi + S^T)$$

Subroutine DREQ insures that the first P used in DRIC is chosen correctly. The routine propagates the Riccati solution P (using subroutine TDREQ) from P = 0 until either EPS is satisfied or MAXD is reached. The solution P, an output from TDREQ, is then used by subroutine DRIC, where a final solution is obtained.* Table 3.4-8 shows the interface between GAIN and DREQ.

Subroutine TDREQ propagates the discrete-time Riccati equation

$$P_k = (\Phi^T P_{k+1}\Phi + Q) - (\Gamma^T P_{k+1}\Phi)^T(R + \Gamma^T P_{k+1} \Gamma)^{-1}(\Gamma P_{k+1}\Phi)$$

---

*Note the program variables A, B, X, and G correspond to the above variables Φ,Γ,P,K respectively.

TABLE 3.4-8

INTERFACE TO ROUTINE DREQ

INPUTS	
N =	number of states
M =	number of inputs
A =	NxN system matrix
B =	NxM input matrix
R =	MxM control weighting matrix
Q =	NxN state weighting matrix
S =	NxM control and state crossed weighting matrix
OUTPUTS	
X =	NxN matrix Riccati equation solution
G =	MxN optimal gain
ACL =	NxN optimal closed-loop matrix

where the variables, PHI, XD, QHAT, RHAT, DT are equivalent to variables Φ, P, Q, R, Γ respectively.

This calculation is accomplished by looping on the routine SMPL and PRPA, which updates and propagates the matrix P. When the convergence criterion EPS is satisfied, as when MAXD, the maximum number of iterations is reached, the closed-loop system ACLD and the feedback gain CD are computed using the equations:

$$CD = (R + \Gamma P \Gamma^T)^{-1} (\Gamma^T P \Phi)$$

$$ACLD = \Phi - \Gamma(CD)$$

Table 3.4-9 gives the interface between TDREQ and DREQ.

The purpose of subroutine DRIC is to solve for the steady-state solution of the discrete-time Riccati equation by Newton's method. This is achieved by the following algorithm. $P_k (k=0,1,2,...)$ is the solution of the linear equation

TABLE 3.4-9

INTERFACE TO ROUTINE TDREQ

INPUTS	
N =	number of states
M =	number of controls
PHI =	NxN discrete system matrix
DT =	NxM discrete output matrix
RHAT =	MxM discrete weighting on controls
QHAT =	NxN discrete weighting on states
MHAT =	NxM discrete cross coupling weighting
OUTPUTS	
XD =	NxN final propagation solution to the Riccati iteration
CD =	MxN discrete feedback gain = (RHAT + DT^T*XD*DT)$^{-1}$*(DT^T*XD*PHI) for TDREQ or (RHAT + DT^T*XD*DT)$^{-1}$*(DT^T*XD*PHI+MHAT^T) for TDREG1
ACLD =	NxN closed-loop system = PHI - DT*CD

$$P_k = \Phi_k^T P_k \Phi_k + Z_k$$

where, recursively

$$\Phi_k = (I + \Gamma R^{-1} \Gamma^T P_{k-1})^{-1} \Phi \qquad k=1,2,...$$

$$Z_k = \Phi_k^T P_{k-1} \Gamma R^{-1} \Gamma^T P_{k-1} \Phi_k + Q$$

and where P_0 is chosen so that the matrix $(1 + \Gamma R^{-1} \Gamma^T P_0)^{-1} \Phi$ has eigenvalues inside the unit circle. Table 3.4-10 shows the interface between DRIC and DREQ.

3.4.6 Proportional-Integral Controller

PID computes the gains for the proportional-integral controller using the following set of equations. If ITYP(1) = 4 or 5, then

TABLE 3.4-10

INTERFACE TO ROUTINE DRIC

INPUTS	
N =	dimension of system
A =	NxN system matrix
S =	NxN matrix = $B*R^{-1}*B^T$
Q =	NxN weighting matrix
X =	NxN initial guess matrix for calling DRIC1
TR =	NxN gain matrix
OUTPUTS	
X =	NxN positive definite solution of matrix Riccati equation
Z =	NxN closed-loop system matrix = $(I+S*X)^{-1}*A$
IOC =	-1 if anything goes wrong

$$\underline{x}_{k+1} = e^{Ft}\,\underline{x}_k + \left(\int_o^T e^{Fs}\,ds\right) G\underline{u}_k \tag{3.6-1}$$

and

$$\underline{u}_k = \underline{u}_{k-1} + R\left((UMAG2) - Q\underline{x}_{k-1}\right) + (RHAT)\left((UMAG2) - Q\underline{x}_k\right) + (QHAT)\underline{x}_k \tag{3.6-2}$$

If ITYP(1) = 6 or 7, then Eq. (3.6-1) remains the same, but Eq. (3.6-2) becomes

$$\underline{u}_k = \underline{u}_{k-1} + R\underline{u}_{k-1} + (RHAT)(UMAG2 - Q\underline{x}_{k-1}) + (QHAT)\,\underline{x}_{k-1} \tag{3.6-3}$$

Table 3.4-11 gives the interface between PID and GAIN.

TABLE 3.4-11

INTERFACE TO ROUTINE PID

INPUTS		
N	=	dimension of composite system
M	=	dimension of controller
P	=	dimension of guidance system
KPUNCH	=	logical unit number for punched output
NPUNCH	=	0 if no punched output is desired
T	=	sampling time of discrete system
NR	=	dimension of system = N-M
F	=	NxN system matrix
OUTPUTS		
R	=	MxP gain matrix for delayed error state
RHAT	=	MxP gain matrix for the current error state
QHAT	=	MxNR state feedback gain matrix
Q	=	PxNR transformation matrix of control system

3.5 ESTIMATOR - OBSERVER DESIGN

The purpose of this program chapter, subroutine EOD, is to perform the estimator observer design calculations (Fig. 3.5-1).

The routine ESTOBS is called to find the steady-state discrete-time Kalman filter gains for a continuous-time system. The system matrix F is discretized with interval T. Table 3.5-1 shows its interface with EOD.

Finally ESTOBS calls KFD, whose purpose is to design a steady-state discrete Kalman filter. Table 3.5-2 shows the relationship between ESTOBS and KFD.

R-19770

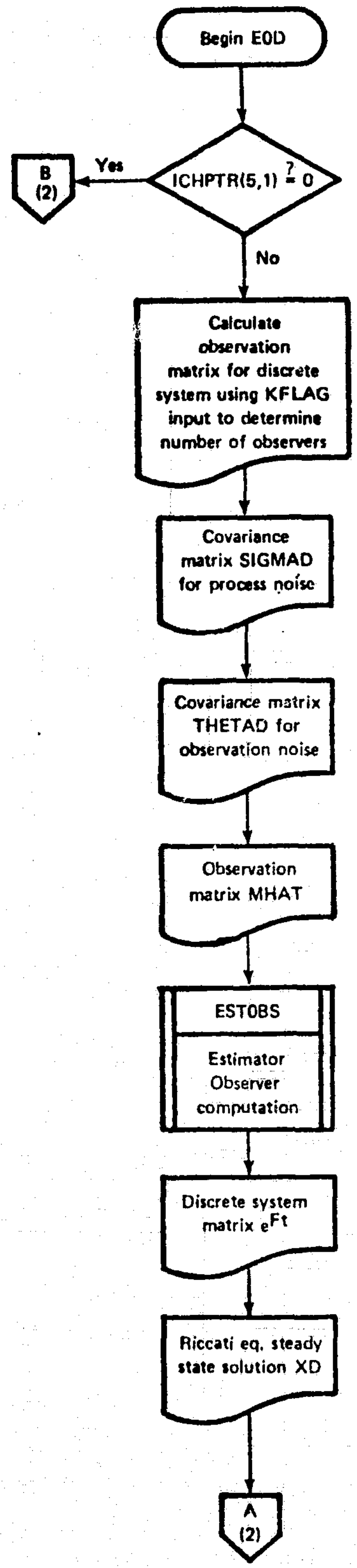


Figure 3.5-1 Flowchart of Subroutine EOD (Segment 1 of 2)

R-19771

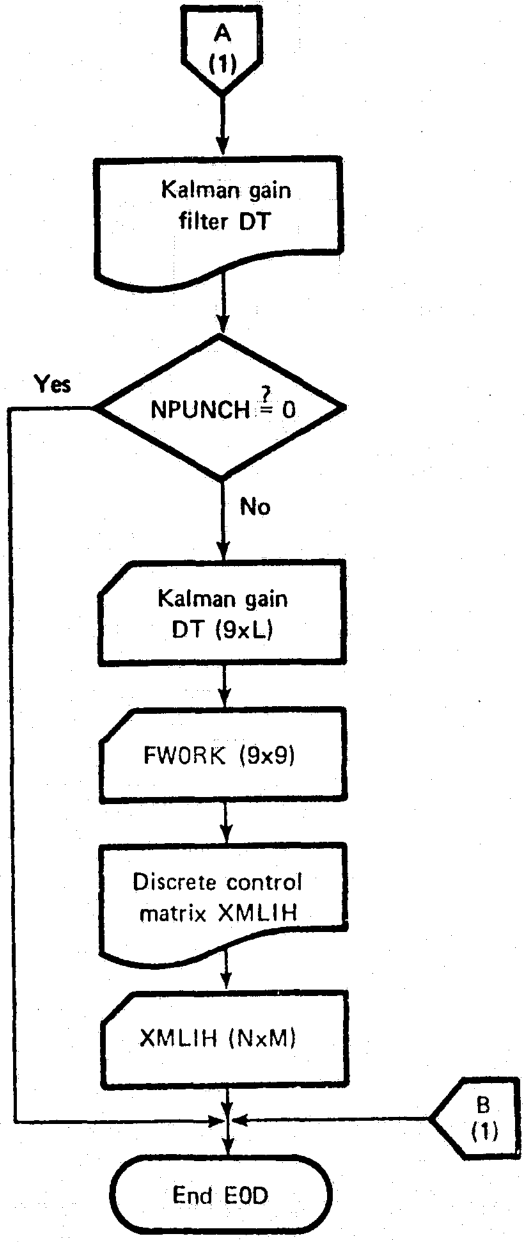


Figure 3.5-1 Flowchart of Subroutine EOD
(Segment 2 of 2)

TABLE 3.5-1

INTERFACE TO ROUTINE ESTOBS

```
INPUTS
     N = dimension of system
     L = dimension of observation
     F = NxN system matrix
     H = NxN observation matrix
SIGMAD = NxN process noise covariance matrix
THETAD = LxL observation noise covariance matrix
         (dimensioned at least NxN)
   EPS = convergence criterion for TDREQ
  MAXD = maximum number of iterations for TDREQ
     T = sampling time for filter

OUTPUTS
     S = 9x9 discrete time steady-state Riccati
         equation solution
    DK = 9xL discrete time Kalman filter gain
   FCL = 9x9 discrete Kalman filter closed-loop
         matrix = F-K*H
```

TABLE 3.5-2

INTERFACE TO ROUTINE KFD

```
INPUTS
   N = number of states
   L = number of observed variables
   A = NxN system matrix
   H = LxN observation matrix
   R = LxL observation noise covariance matrix
   Q = NxN process noise covariance matrix
   D = NxL correlated noise matrix

OUTPUTS
   S = NxN discrete Riccati equation steady
       state solution
   K = NxL Kalman filter discrete gain
 FCL = A-K*H
```

3.6 EIGENVALUE COMPUTATION

The purpose of this program chapter, subroutine EIGEN is to calculate the eigenvalue of a given matrix. Consider Fig. 3.6-1, which shows that the eigenvalues of three matrices are of interest to DIGADAPT. They are ACL, the closed loop system, ACLD, the closed-loop discrete system, and FCL, the closed-loop Kalman filter system. Note that if the user so desires, he can calculate the eigenvalue of any matrix by a direct call to FREE, where it should be noted that, in calculating the eigenvalues, the matrix of interest is destroyed.

The purpose of FREE is to calculate the eigenvalues of various modes of motion of the given aircraft system matrix, along with the corresponding natural frequencies, damping rates, periods, and times to half amplitude. In turn, FREE calls two programs HSBG and ATEIG, which have been incorporated from the System/360 Scientific Subroutine Package (Ref. 4).

The subroutine HSBG reduces an NxN real matrix A by similarity transformation to upper almost-triangular (Hessenberg) form. Each row is reduced in turn, starting from the last one by applying a suitable right elimination matrix, and similarity is achieved by also applying the left inverse transformation. Thus the eigenvalues are preserved. Following the subroutine HSBG, ATEIG computes the eigenvalues of a real upper almost-triangular matrix using the double-QR iteration of J.G.F. Francis (Ref. 4). Table 3.6-1 shows the interface between EIGEN and FREE.

3.7 TIME HISTORY SIMULATIONS

The purpose of this program chapter, subroutine TIME, is to control time history calculations for continuous-time

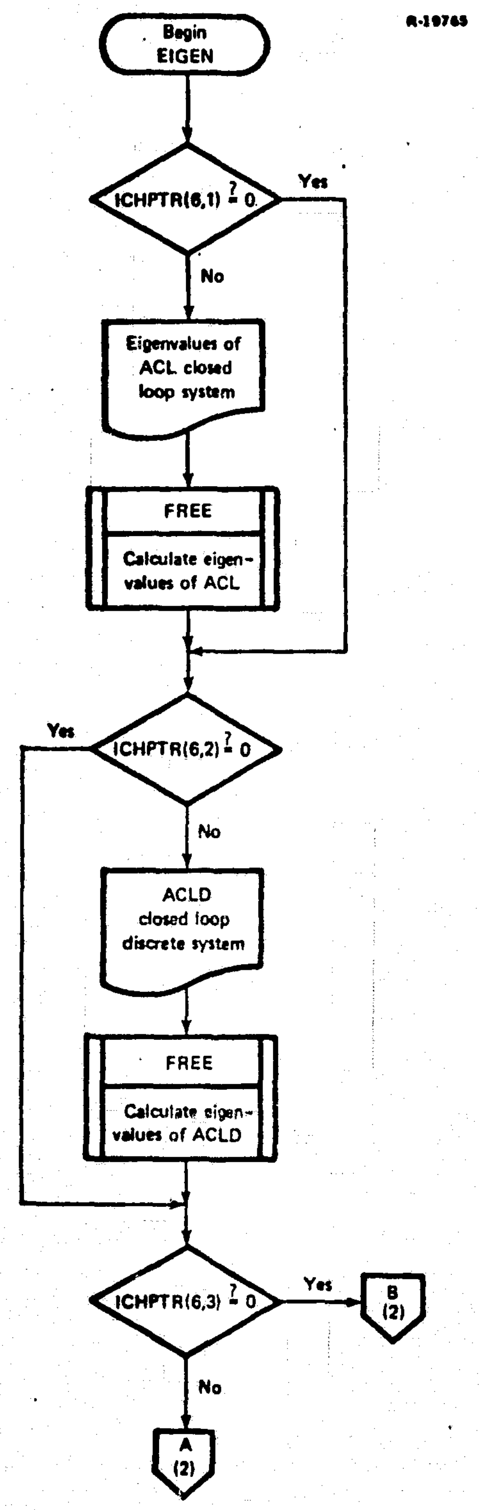


Figure 3.6-1   Flowchart of Subroutine EIGEN
(Segment 1 of 2)

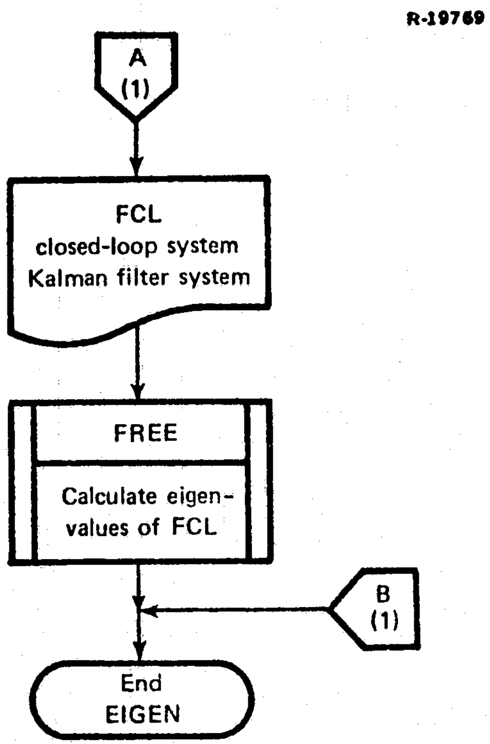


Figure 3.6-1 Flowchart of Subroutine EIGEN (Segment 2 of 2)

TABLE 3.6-1

INTERFACE TO ROUTINE FREE

INPUTS N = number of state in the system F = NxN aircraft system matrix of NxN closed-loop system matrix
OUTPUTS RR = n-vector with real parts of the eigenvalues of F RI = n-vector with imaginary parts of the eigenvalues

and discrete-time systems. Figure 3.7-1 gives an overview of its structure.

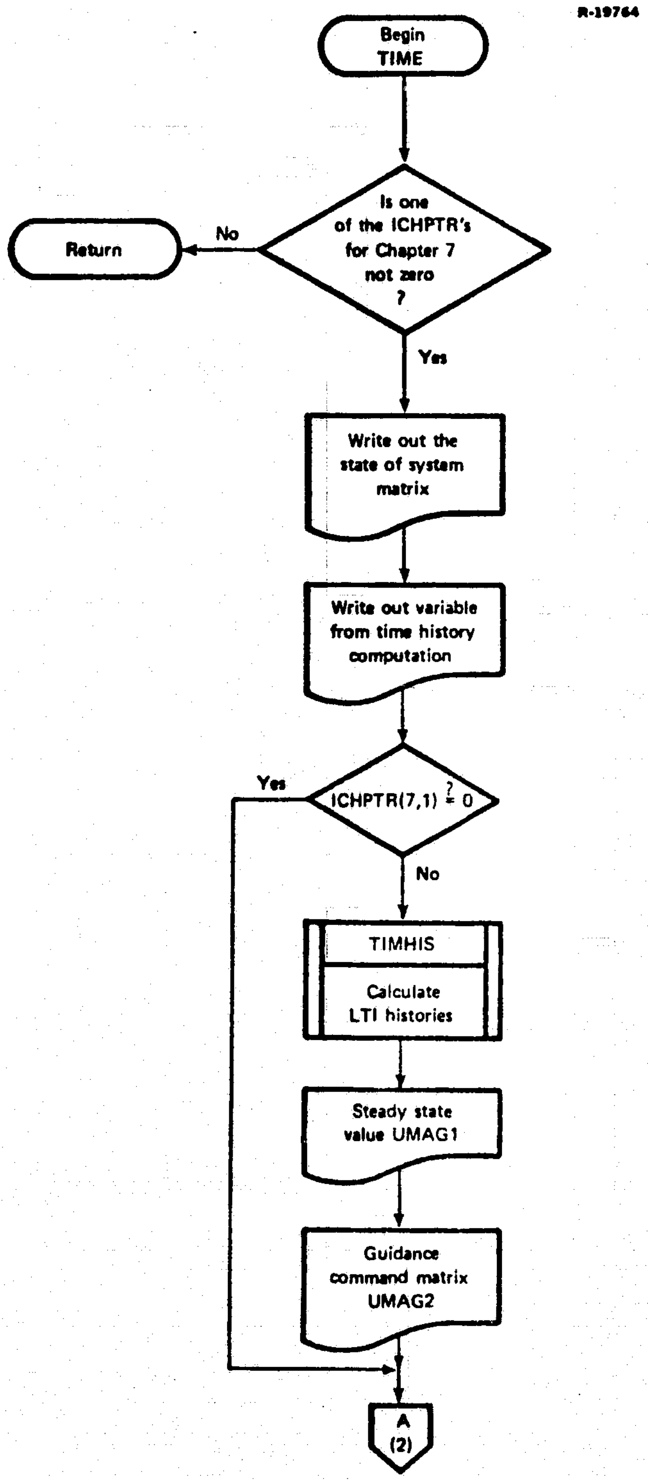


Figure 3.7-1 Flowchart of Subroutine TIME (Segment 1 of 2)

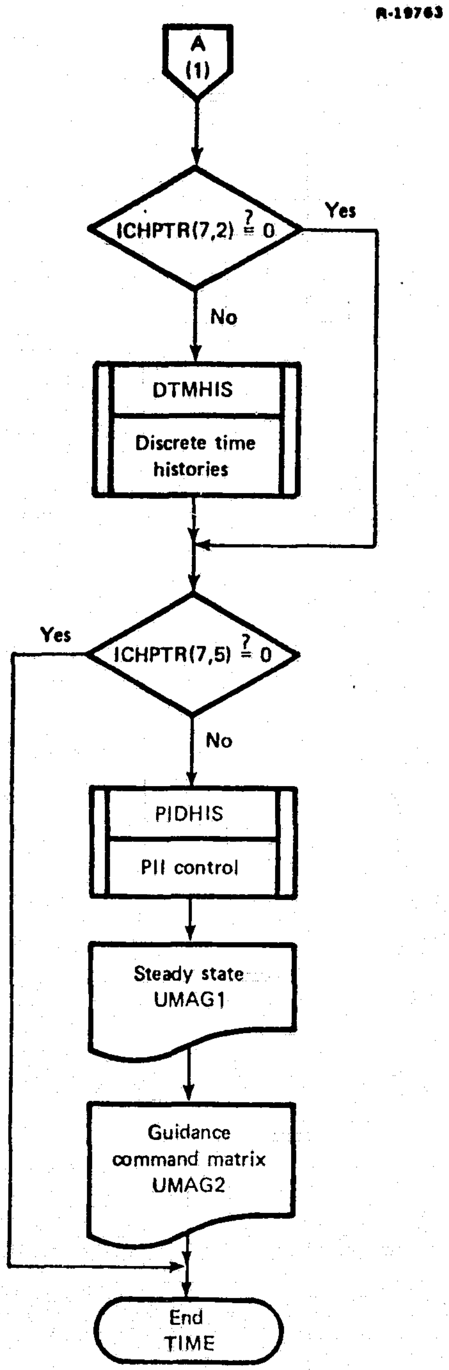


Figure 3.7-1 Flowchart of Subroutine TIME (Segment 2 of 2)

3.7.1 Linear Time Invariant Calculation

TIMHIS simulates a continuous linear time invariant (LTI) system

$$\begin{bmatrix} \dot{\underline{x}} \\ \dot{\underline{u}} \\ \dot{\underline{v}} \end{bmatrix} = \begin{bmatrix} ACL \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \begin{bmatrix} UMAG2 \end{bmatrix} + \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix} \begin{bmatrix} UMAG1 \end{bmatrix}$$

using state-transition-matrix propagation of the state. The propagation equation is

$$\begin{bmatrix} \underline{x}_{k+1} \\ \underline{u}_{k+1} \\ \underline{v}_{k+1} \end{bmatrix} = PHI \begin{bmatrix} \underline{x}_k \\ \underline{u}_k \\ \underline{v}_k \end{bmatrix} + \Gamma_2(DELU) + \Gamma_1(UMAG1)$$

where

$$PHI = e^{(ACL)\ t}$$

$$\Gamma_1 = \left(\int_o^{\Delta t} e^{(ACL)s}\, ds \right) \begin{bmatrix} 0 \\ I \\ 0 \end{bmatrix}$$

$$\Gamma_2 = \left(\int_o^{\Delta t} e^{(ACL)s}\, ds \right) \begin{bmatrix} 0 \\ 0 \\ -I \end{bmatrix}$$

For each run (up to a maximum of 4) there is a maximum of IMAX time steps; for each time step 31 variables are calculated including u, v, w, p, q, r, θ, ψ, ϕ, δ_B, δ_C, δ_R, δ_S, y_1, y_2, y_3, y_4, v_x, v_y, v_z, u_1, u_2, u_3, u_4, $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{\delta}_B$, $\dot{\delta}_G$, $\dot{\delta}_R$. Table 3.7-1 shows the interface between TIMHIS and TIME.

TABLE 3.7-1

INTERFACE TO ROUTINE TIMHIS

INPUTS	
N	= total number of states and controls
M	= number of controls
P	= number of commands
NR	= N - M - P
ACL	= NxN closed loop matrix
G	= NxM control matrix
UO, VO, WO	= initial body velocity
IRUN	= number of time histories up to 4
UMAG2	= Px4 guidance inputs to the system
UMAG1	= Nx5 feedforward control input
C	= MxN feedback gain of system
DELT	= interval of discretization
ITYP(1)	= 1 no integrator states
ITYP(1)	= 2 velocity guidance system
ITYP(1)	= 3 attitude guidance system
ITYP(2)	= 2 UMAG1 is not used and DELU is set to zero
ITYP(2)	= 1 UMAG1 is turned on
DELU	= C*UMAG1
DELV	= UMAG2
IMAX	= maximum number of iteration steps
DXZERO	= Nx5 initial condition of the states, controls and integrators
OUTPUTS	
PX	= see discussion of output reports in Table 2.2-1

3.7.2 Discrete PII Calculation

The purpose of DTMHIS is to simulate a discrete-time system. The system simulated is

$$\begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix}_{k+1} = \begin{bmatrix} ACLD \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix}_{k} + \begin{bmatrix} 0 \\ 0 \\ TI \end{bmatrix} \begin{bmatrix} UMAG2 \end{bmatrix}$$

The exact system is simulated if ICHPTR(7,3)=1; an approximation to the discrete-time system is simulated if ICHPTR(7,4)=1. The approximate equation is

$$\begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix}_{k+1} = \begin{bmatrix} PHI & DT & 0 \\ 0 & I & 0 \\ TT & 0 & I \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix}_k - \begin{bmatrix} 0 \\ TI \\ 0 \end{bmatrix} \begin{bmatrix} CD \end{bmatrix} \begin{bmatrix} \underline{x} \\ \underline{u} \\ \underline{v} \end{bmatrix}_k$$

$$- \begin{bmatrix} 0 \\ 0 \\ TI \end{bmatrix} \begin{bmatrix} UMAG2 \end{bmatrix}_k$$

This routine uses the same run logic as TIMHIS. Table 3.7-2 shows the relationship between DTHMIS and TIME.

TABLE 3.7-2

INTERFACE TO ROUTINE DTMHIS

INPUTS	
N	= number of states
M	= number of controls
NR	= number of aircraft states
P	= number of integrator states
T	= sampling time
F	= NxN system matrix
CD	= MxN feedback gains matrix
UMAG2	= Px1 set point or guidance command
DXZERO	= Nx1 initial condition of states
ACLD	= control discrete input matrix
OUTPUTS	
PX	= see discussion of output reports in Table 2.2-1

3.7.3 Discrete PI and Control Rate Calculation

The purpose of PIDHIS is to perform a simulation for the PI control system. If ITYP(2) = 4 or 5, then the simulation uses

$$\underline{x}_{k+1} = \Phi \underline{x}_k + \Gamma \underline{u}_k$$

$$\underline{u}_k = \underline{u}_{k-1} + R(UMAG2 - Q\underline{x}_{k-1}) + RHAT(UMAG2 - Q\underline{x}_k) + QHAT\ \underline{x}_k$$

$$\text{where PHI} = \begin{pmatrix} \Phi & \Gamma \\ 0 & I \end{pmatrix}, \quad \Phi = e^{FT} \quad \text{and} \quad \Gamma = \left(\int_0^T e^{Fs}\, ds \right) * G$$

If ITYP(2) = 6 or 7, then the program simulates

$$\underline{x}_{k+1} = \Phi\, \underline{x}_k + \Gamma\, \underline{u}_k$$

$$\underline{u}_k = \underline{u}_{k-1} + R\underline{u}_{k-1} + \text{RHAT}(\text{UMAG2} - Q\underline{x}_{k-1}) + (\text{QHAT})\underline{x}_{k-1}$$

PIDHIS also has the same time history simulation logic as TIMHIS and DTMHIS. Table 3.7-3 shows the interface between PIDHIS and TIME.

TABLE 3.7-3
INTERFACE TO ROUTINE PIDHIS

INPUTS	
PHI	= NxN discrete
R	= MxP matrix
RHAT	= MxP matrix
DXZERO	= Nx4 = initial condition of states and controls
UMAG2	= Px1 guidance command
QHAT	= MxNR matrix
Q	= MxNR command transformation matrix
N	= dimension of system + control
P	= dimension of guidance command
NR	= N-M
T	= sampling time
OUTPUTS	
PX	= see discussion of output reports in Table 2.2-1

3.8 SERVICE UTILITIES

A number of utility routines are used by the subroutines described in earlier sections. Table 3.8-1 lists these service programs, gives an appropriate description of each, and make references to Tables 3.8-2 through 3.8-33 that describe their interface.

TABLE 3.8-1

UTILITY ROUTINES

SUBROUTINE	DESCRIPTION	TABLE OF I/O
DOT	Take dot product between two vectors	Table 3.8-2
CONTRL	Calculate control and guidance inputs	Table 3.8-3
STEADY	Calculate steady state in a time history simulation	Table 3.8-4
HSBG	Transform matrix into Hessenberg form.	Table 3.8-5
ATEIG	Find eigenvalues of a matrix in Hessenberg form	Table 3.8-6
ALPHA	Compute integral of a function of the form $(\int e^{\tilde{A}}) * B * (\int e^{A})$	Table 3.8-7
MLINEQ	Solve matrix equation $A^{T}X + A + C = 0$ for X	Table 3.8-8
PRMR	Print matrix routine	Table 3.8-9
EQAT	Set two matrices equal to each other	Table 3.8-10
IDEN	Set a matrix to the identity matrix	Table 3.8-11
JUXC	Concatenate two matrices columnwise, i.e., $A \vert B$	Table 3.8-12
JUXR	Concatenate two matrices rowwise, i.e., $\frac{A}{B}$	Table 3.8-13
MNUS	Multiply matrix by (-1)	Table 3.8-14
MULT	Multiply two matrices	Table 3.8-15
NORM	Compute norm of a matrix	Table 3.8-16
SMLT	Multiply a matrix by a scalar value	Table 3.8-17
SUBT	Subtract one matrix from another	Table 3.8-18

TABLE 3.8-1

UTILITY ROUTINES (Continued)

SUBROUTINE	DESCRIPTION	TABLE OF I/O
TMLT	Multiply a matrix by the transpose of another	Table 3.8-19
TRNP	Compute transpose of a matrix	Table 3.8-20
ZERO	Zeros a matrix	Table 3.8-21
SETA	Compute transition matrix and discrete form of the noise matrix	Table 3.8-22
PROP	Propagate covariance matrix	Table 3.8-23
EXPI	Compute state transition and its integral	Table 3.8-24
PRPA	Propagate covariance matrix	Table 3.8-25
SMPL	Compute optimal Kalman gain matrix and performs filter update	Table 3.8-26
RCT	Solve covariance equation	Table 3.8-27
INV	Calculate inverse of a matrix	Table 3.8-28
RSET	Compute transition matrix for continuous Riccati equation	Table 3.8-29
EXPJ	Compute exponential of a matrix	Table 3.8-30
ADD	Add two matrices	Table 3.8-31
BETA	Assist subroutine ALPHA in computing integral	Table 3.8-32
DINTEG	Compute discrete integral	Table 3.8-33

TABLE 3.8-2

INTERFACE TO ROUTINE DOT

INPUTS
- N = dimension of vectors A and B
- A = N-vector
- B = N-vector

OUTPUTS
- DOT = result of the dot product of A and B

TABLE 3.8-3

INTERFACE TO ROUTINE CONTRL

INPUTS		
N	=	number of states in system
M	=	number of controls in system
P	=	dimension of guidance command
IP	=	run number
C	=	MxN gain matrix
OUTPUTS		
DELU	=	M-vector = (feedback gain)*(feedward control)
DELV	=	P-vector of guidance inputs to the system

TABLE 3.8-4

INTERFACE TO ROUTINE STEADY

INPUTS		
ITYP	=	flag to determine velocity or angular system
PX	=	201x31 time history computation matrix
DELT	=	sample period
IMAX1	=	number of time steps in sample periods, plus one
OUTPUTS		
SS	=	steady-state value for variables 1 to NM3
PV	=	peak value
TPV	=	time of peak value
T05	=	time to reach a five percent bound of steady state
T90	=	time to reach ninety percent of steady state

TABLE 3.8-5

INTERFACE TO ROUTINE HSBG

INPUTS		
N	=	order of the matrix
A	=	the input matrix, N x N
IA	=	size of the first dimension assigned to the array A in the calling program when the matrix is in double-subscripted data storage mode. IA=N when the matrix is in SSP vector storage mode.
OUTPUTS		
A	=	returned N x N matrix in 'almost' Hessenberg form

TABLE 3.8-6

INTERFACE TO ROUTINE ATEIG

INPUTS	
M =	order of the matrix
A =	the input matrix, M by M
IANA =	vector whose dimension must be greater than or equal to M, containing on return indications about the way the eigenvalues appeared (see Ref. 4).
IA =	size of the first dimension assigned to the array A in the calling program when the matrix is in double-subscripted data storage mode. IA=M when the matrix is in SSP vector storage mode.
OUTPUTS	
RR =	vector containing the real parts of the eigenvalues
RI =	vector containing the imaginary parts of the eigenvalues

TABLE 3.8-7

INTERFACE TO ROUTINE ALPHA

INPUTS	
N =	dimension of square matrix A
A =	constant matrix
SIG =	upper limit of integration
EPSLO =	convergence criterion
GEST =	index for starting test for convergence
LLL =	1,2
H =	1.0, 0.0
OUTPUTS	
SUM =	the integral of the desired function

TABLE 3.8-8

INTERFACE TO ROUTINE MLINEQ

INPUTS	
TOL =	convergence tolerance
N =	dimension of square array
A =	NxN matrix
C =	NxN matrix
OUTPUTS	
X =	positive semi-definite solution of matrix linear equation

TABLE 3.8-9

INTERFACE OF ROUTINE PRMR

INPUTS
A = Holerith string title of length L
L = character length of Holerith field in
N = number of rows
M = number of columns
X = NxM matrix

TABLE 3.8-10

INTERFACE TO ROUTINE EQAT

INPUTS
N = number of rows
M = number of columns
A = NxM matrix
OUTPUTS
B = NxM Matrix = A

TABLE 3.8-11

INTERFACE TO ROUTINE IDEN

INPUTS
N = dimension of matrix A
OUTPUTS
A = NxN identity matrix

TABLE 3.8-12

INTERFACE TO ROUTINE JUXC

INPUTS
N = number of rows in matrices A and B
MA = number of columns in A
MB = number of columns in B
A = NxMA matrix
B = NxMB matrix
OUTPUTS
C = Nx(MA + MB) matrix

TABLE 3.8-13

INTERFACE TO ROUTINE JUXR

```
INPUTS
   M = number of columns in matrices A and B
  NA = number of rows in A
  NB = number of rows in B
   A = NAxM matrix
   B = NBxM matrix
OUTPUTS
   C = (NA+NB)xM matrix
```

TABLE 3.8-14

INTERFACE TO ROUTINE MNUS

```
INPUTS
   N = number of rows
   M = number of columns
   A = NxM matrix
OUTPUTS
   B = NxM matrix = -A
```

TABLE 3.8-15

INTERFACE TO ROUTINE MULT

```
INPUTS
   N = number of rows in A
   L = number of columns A and number of rows in B
   M = number of columns in B
   A = NxL matrix
   B = LxM matrix
OUTPUTS
   C = NxM matrix = A*B
```

TABLE 3.8-16

INTERFACE TO ROUTINE NORM

```
INPUTS
   N = number of columns
   M = number of rows
   A = NxM matrix
OUTPUTS
   S = norm of matrix A
```

TABLE 3.8-17

INTERFACE TO ROUTINE SMLT

INPUTS
S = scalar to multiply A
N = number of rows
M = number of columns
A = NxM matrix
OUTPUT
B = NxM matrix = S * A

TABLE 3.8-18

INTERFACE TO ROUTINE SUBT

INPUTS
N = number of rows
M = number of columns
A = NxM matrix
B = NxM matrix
OUTPUTS
C = NxM matrix = A-B

TABLE 3.8-19

INTERFACE TO ROUTINE TMLT

INPUTS
N = number of rows in A
L = number of columns in A and B
M = number of rows in B
A = NxL matrix
B = MxL matrix
OUTPUTS
C = NxM matrix = $A*B^T$

TABLE 3.8-20

INTERFACE TO ROUTINE TRNP

INPUTS
N = number of rows
M = number of columns
A = NxM matrix
OUTPUTS
B = MxN matrix = A^T

TABLE 3.8-21

INTERFACE TO ROUTINE ZERO

```
INPUTS
   N = number of rows
   M = number of columns
OUTPUTS
   A = NxM zero matrix
```

TABLE 3.8-22

INTERFACE TO ROUTINE SETA

```
INPUTS
    N = dimension of system
    F = NxN system matrix
    Q = NxN noise matrix
   DT = time step
OUTPUTS
   RK = NxN discrete noise matrix
   TR = NxN transition matrix
```

TABLE 3.8-23

INTERFACE TO ROUTINE PROP

```
INPUTS
    N = number of states
    P = NxN covariance matrix
   TR = (2*N) x (2*N)
OUTPUTS
    P = NxN propagated covariance matrix
```

TABLE 3.8-24

INTERFACE TO ROUTINE EXPI

```
INPUTS
     N = number of states
     F = NxN system matrix
    DT = specified time interval
OUTPUTS
    TR = NxN transition matrix
   TRI = NxN integral matrix
```

TABLE 3.8-25

INTERFACE TO ROUTINE PRPA

```
INPUTS
   N = dimension of system
   P = NxN covariance matrix
  TR = NxN transition matrix
  QD = discrete noise matrix
OUTPUT
   P = NxN propagated covariance matrix
```

TABLE 3.8-26

INTERFACE TO ROUTINE SMPL

```
INPUTS
   N = dimension of covariance matrix
   M = dimension of measurement noise matrix
   P = NxN covariance matrix
   H = MxN measurement matrix
   R = measurement noise matrix
OUTPUTS
   P = NxN updated covariance matrix
  GK = NxM optimal Kalman gain matrix
```

TABLE 3.8-27

INTERFACE TO ROUTINE RCT

```
INPUTS
   N = dimension of covariance matrix P
 MAX = maximum number of time steps through
       which P is to be propagated
   P = NxN covariance matrix
  TR = (2*N) x (2*N) transition matrix
 EPS = convergence error criterion
OUTPUTS
   P = NxN propagated covariance matrix
 NUM = number of time steps through which
       P has been propagated
```

TABLE 3.8-28

INTERFACE TO ROUTINE INV

```
INPUTS
   N = order of matrix A
   A = NxN matrix which is destroyed and replace
       by its inverse
OUTPUTS
   A = NxN inverse of input matrix A
   D = Determinant of A - equals 0 if A is singular
```

TABLE 3.8-29

INTERFACE TO ROUTINE RSET

```
INPUTS
   N = number of states
   M = dimension of measurement matrix
   F = NxN system matrix
   R = MxM measurement noise matrix
   H = MxN matrix relating measurement vector
       to state vector
  QG = NxN noise matrix
OUTPUTS
  TR = (2*N) x (2*N) transition matrix
```

TABLE 3.8-30

INTERFACE TO ROUTINE EXPJ

```
INPUTS
   N = dimension of matrix A
   A = NxN matrix
OUTPUTS
EXPA = NxN matrix exponential of A
```

TABLE 3.8-31

INTERFACE TO ROUTINE ADD

```
INPUTS
   N = number of rows
   M = number of columns
   A = NxM matrix
   B = NxM matrix
OUTPUTS
   C = NxM matrix = A+B
```

TABLE 3.8-32

INTERFACE TO ROUTINE BETA

```
 INPUTS
    L = flag equal to 1 or 2
    N = size of matrix A
    A = NxN matrix
 ASIG = NxN matrix
EPSLO = convergence criterion
 OUTPUTS
   SS = NxN matrix
```

TABLE 3.8-33

INTERFACE TO ROUTINE DINTEG

```
INPUTS
  NT = number of summations
   N = dimension of system
   A = NxN matrix
   C = NxN matrix
OUTPUTS
   S = NxN discrete integral matrix
```

4. SCHED USAGE

SCHED transforms independent variables, performs polynomial or multiple regressions, and determines the correlation coefficients between gains, trim settings, and flight conditions. These features are controlled by the ICHPTR(I) flags (I = 1 to 6). The purpose of this program is to develop a set of regression coefficients to be used in an open-loop explicity-adaptive control algorithm.

4.1 SCHED INPUT

SCHED uses Fortran logical unit 5 (TAPE5) for all its inputs. In creating the input deck for SCHED, it is necessary to choose appropriate flight conditions (independent variables X) and gains (dependent variables Y) which have been punched by DIGADAPT. Table 4.1-1 summarizes all card types for SCHED, including a list of variables, their descriptions, their format, and their appropriate card image positions.

TABLE 4.1-1

INPUT FOR SCHED

VARIABLE(S)	DESCRIPTION	FORMAT	COLUMNS	CARD TYPE
N	Number of observations	I4	1-4	1
M	Number of independent variables	I4	5-8	1
L	Number of dependent variables	I4	9-12	1
ICHPTR ICHPTR(1)	6-vector control flag =0, do not perform polynomial regression =1, performs polynomial regression of highest order 2 of independents and dependents	I2	1-2	2
ICHPTR(2)	=0, printout only correlation coefficients of polynomial regression =1, print regression coefficient, an analysis of variance, table of residuals and correlation coefficients	I2	5-6	2
ICHPTR(3)	=0, do not perform simple regression =1, perform simple regression between dependent variables	I2	9-10	2
ICHPTR(4)	=0, printout only correlation coefficients of simple regression =1, print regression coefficients, an analysis of variance, table of residuals, correlation coefficients, and an analysis of a multiple regression.	I2	13-14	2
ICHPTR(5)	=0, do not perform multiple regression =1, perform multiple regression analysis	I2	17-18	2
ICHPTR(6)	=0, printout only correlation coefficients of multiple regression =1, print regression coefficients, analysis of multiple regression, table of residuals, and correlation coefficients.	I2	21-22	2
NIND	Number of independent variables for polynomial regression $\leq$ M	I2	1-2	3
IIND	NIND vector used to choose which independent variable will be used in polynomial regression	10(I2,2X)	1-2,5-6, 9-10,etc.	4
JORDER	(NIND + 2) array of flags where the second subscript turns 'on' (=1) or 'off' (=0) the first and second-order polynomial regression. The first subscript corresponds to the NIND variables of the IIND vector. These values are read in rowwise; hence, each row corresponds to an independent variable.	20I2	1-2,3-4, 5-6,etc.	5

TABLE 4.1-1

INPUT FOR SCHED

VARIABLE(S)	DESCRIPTION	FORMAT	COLUMNS	CARD TYPE
JIND	(NIND x 5) array of flags which control the transformation on the independent variable. Each row of five switches corresponds to the appropriate independent variable of the IIND vector. 0 implies 'off' and 1 implies 'on'. The five switches pertain to the following transformations of the independent variable 1. No change 2. Squared 3. Inverse 4. Inverse squared 5. X * ABS(X)	35I2	1-2,3-4 5-6,etc.	6
ISI	Total number of multiple regressions to be performed	I2	1-2	7
MUL	Number of independent variables for each multiple regression. Card Type 8 and card type 9 is repeated ISI times in the sequence 8, 9, 8, 9, etc.	I2	1-2	8
MCC	MUL - vector which specifies which independent variables are to be used for the multiple regression	10(I2,2X)	1-2,5-6 9-10,etc.	9
IRUM	Case numbers (same as NCASE from DIGADAPT). Cards type 10, 11, 12 are repeated N times in the sequence 10, 11, 12, 10, 11, 12, ...	I5	1-5	10
X	M independent variables	6E12.5	1-12,13-24	11
Y	L dependent variables	6E12.5	1-12,12-24, ...	12

4.2 SCHED OUTPUT

The output of SCHED is on Fortran logical unit 6 (TAPE6). Table 4.2-1 gives a summary of the reports and a description and/or variable listing. All output from SCHED is printed from the main routine.

TABLE 4.2-1

OUTPUT FROM SCHED

REPORT	DESCRIPTION-VARIABLE
Problem/Parameter Cards	N, number of observations, M, number of independent variables and L, number of dependent variables
Independent Variables for Polynomial Regressions	IIND, JORDER, JIND
Number of Multiple Regressions	ISI
Number of Independent Variables	MUL, MCC
List of Independent Variables	Case number and independent variables are printed out along with headings for $\dot{x}$, $\dot{y}$, $\dot{z}$, θ, ϕ, ψ, $\dot{\psi}$, velocity
List of Dependent Variables	Dependent variables y and the following variables:
Gain Number Mean Standard Deviation Percent	I AMEAN STDV PRCNT (=100*STDV/AMEAN)
Begin Polynomial Regression	---
Independent = Change = Gain No = IRDER =	Independent variables modified Change of variable code - IHNGE Gain number - IAIN Order of polynomial - IRDER
Variable power Mean Standard Deviation Correlation	I XBAR STD D
Intercept	ANS(1)
Regression Coefficients	B
Multiple Correlation Coefficient	ANS(2)
Source of Variation Degree of Freedom Sum of Squares Mean Square	IRDER ANS(4) ANS(6) ANS(10)
Deviation About Regression	NI, ANS(7) - array of square of deviation from regression, ANS(9) mean square of ANS(7)
Total	NT = N-1, SUMSQ(NRDER)
Table of Residuals	I, observation number XX(I), x value XX(IRDER * N + I), y value COE(NRDER), y estimate RES, residuals
Independent Variable	IIND(I)
Polynomial Order	NZ
Correlation Coefficients	CORRCO
Begin Simple Regression	Written on top of page
Independent and Dependent Variable	XX
Begin Multiple Regression	Header Title
Intercept Multiple Correlation Standard Error of Estimate	ANS(1) ANS(2) ANS(3)
Multiple Regression No.	INI
Independent Variable =	MCC, also prints out correlation coefficients CORMUL

5. DETAILED DESCRIPTION OF SCHED

SCHED performs a correlation and regression analysis between independent and dependent variables. The independent variables are flight conditions at particular points in the flight envelope of the aircraft, and the dependent variables are control gains, trim settings, system matrices, and filter gains.

In addition to performing regressions and determining correlation coefficients, the program can perform transformations on any number of the independent variables. The change of variables that can be chosen are squared, inverted, inverted and squared, and squared preserving the sign of the variable. The transformed variables are used to obtain a polynomial regression of order one or two. In addition, SCHED can perform simple regressions between the dependent variables and multiple regressions between a select group of independent variables and all the dependent variables. The outputs of the program are correlation coefficients, regression coefficients and tables of residuals. These results are computed by the following routines: GDATA, MULTR, CORRE, and ORDER. (See Fig. 5-1).

GDATA generates independent variables up to the nth power (for the polynomial regression) and calculates means, standard deviations, sums of cross-products of deviations from means, and product-moment correlation coefficients (Ref.4).

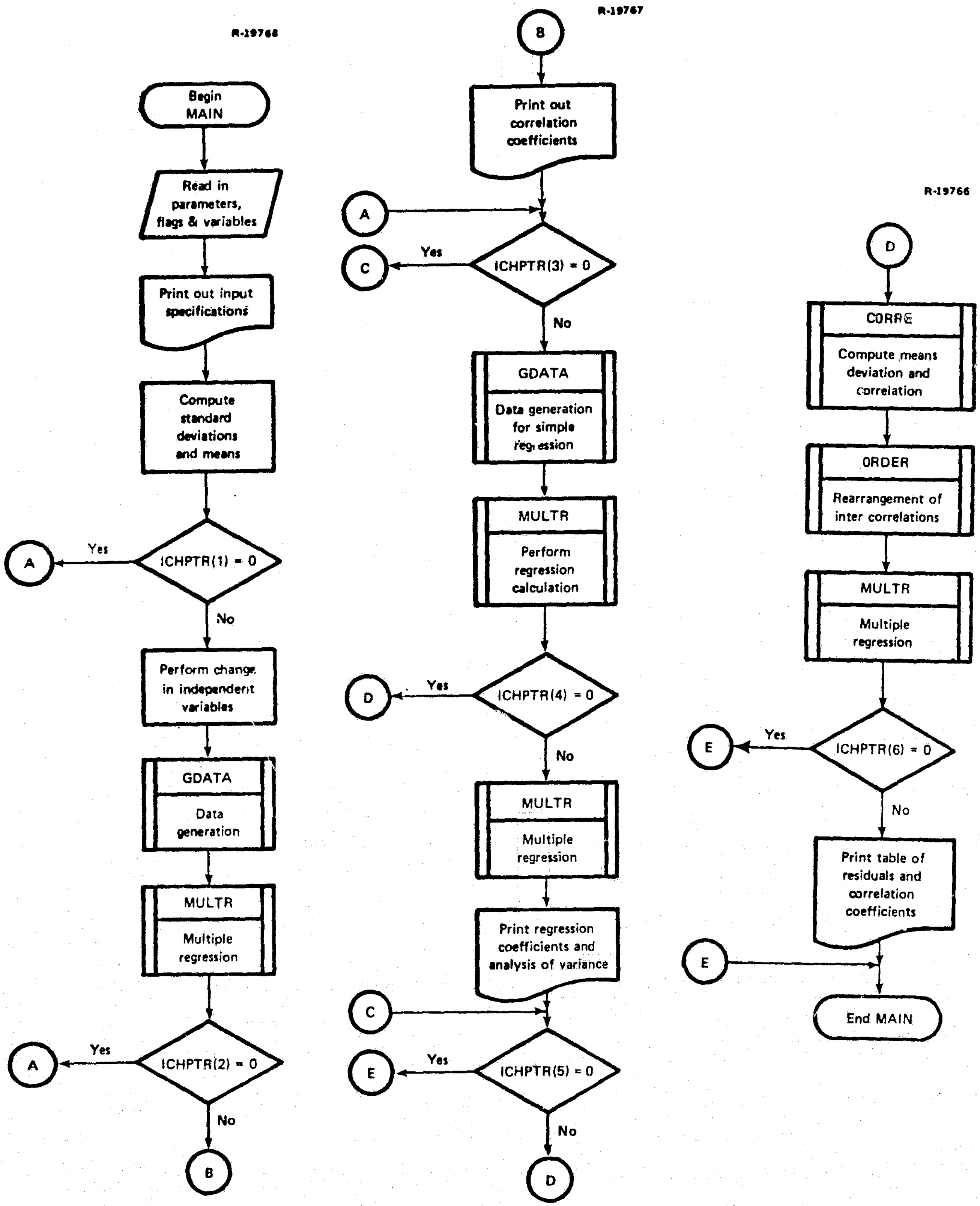


Figure 5-1 Flowchart of Program SCHED

For the polynomial regressions, ORDER chooses a dependent variable and a subset of independent variables, and MULTR computes the regression coefficients. For the multiple regressions, the routine CORRE is used in place of GDATA to calculate standard deviations, means, sums of cross-products of deviations from means, and product moment correlation coefficients from input data.

6. TVHIS USAGE

TVHIS performs a flight path simulation based on a linear-time-varying model that includes actuator and rotor dynamics. The logic is controlled by the flags IREF, IAUG, IEST, IACT, IGNPT, and INOMPT. The purpose of this program is to test and evaluate the gain-scheduled control law table that has resulted from the analysis performed with DIGADAPT and SCHED.

6.1 TVHIS INPUT

Input for TVHIS is in two forms: card input on Fortran logical unit 5 (TAPE5) and permanent file disk inputs on Fortran logical unit 1 (TAPE1). Table 6.1-1 describes the input variables, their descriptions, their units, their formats, and their appropriate card image positions. Reference is made to Table 6.1-3 for the units of the units of the perturbation states. The variables ACONST and REGCOF should be obtained from the output of program SCHED. The disk file inputs, which are the same aerodynamic coefficients used in program DIGADAPT, are linearly interpolated in TVHIS to determine the system matrices F0 and F1.

In addition, there are variables initialized in BLOCK DATA. Table 6.1-2 gives the BLOCK DATA variables, their descriptions and their units. (Note that recompilation is necessary to change a variable in BLOCK DATA.)

TABLE 6.1-1

CARD INPUT FOR TVHIS

VARIABLES	DESCRIPTION	FORMAT	UNITS	COLUMN	CARD TYPE
XIC	24-vector of initial conditions on perturbation state vector	16F4.0	*	1-4,5-8,etc.	1
DIGUID	Guidance time interval	F7.0	sec	1-7	2
DTCNTR	Control time interval	F7.0	sec	8-14	2
DTOUTP	Output time interval	F7.0	sec	15-21	2
NEV	Number of special events	I3	-	1-3	3
ITYP,TEVENT EVMAG	Key, time, and magnitude of special event - Card Type 4 is repeated NEV times	I3,2F10.0	-,sec,-	1-3,4-7,8-11	4
NWPF,NWPL	Flight waypoint numbers	2I3	-	1-3,4-6	5
ACONST	68-vector of gain regression (9 cards) coefficients, from SCHED	8F10.0	-	1-10,11-20, 21-30, etc.	6
REGCOF	68 x 3 array of u-regression coefficients, w-regression coefficients, and V^2-regression coefficients from SCHED (26 cards)	8F10.0	-	1-10,11-20, 21-30, etc.	7

*See Table 6.1-3

TABLE 6.1-2

BLOCK DATA INITIALIZATION

VARIABLES	DESCRIPTION	UNITS
KIN	Input logical unit = 5	-
KOUT	Output logical unit = 6	-
KPUNCH	Punch logical unit = 7	-
ACCU	Convergence criterion for RKGS	-
NDIM	Number of states = 24	-
WT	NDIM vector of state error weights for RKGS	-
GVD	Guidance gains at each waypoint	sec^{-1}
LIMIT	Actuator rate and displacement limit key	-
GRAV	Gravitational constant g	ft/sec^2
RADS	Conversion factor from degrees to radians	rad/degree
CKNOTS	Conversion factor from knots to feet/sec	feet/sec-knots

TABLE 6.1-2

BLOCK DATA INITIALIZATION (Continued)

VARIABLES	DESCRIPTION	UNITS
IXX	Second moment of inertia about x-axis	slug-ft^2
IYY	Second moment of inertia about y-axis	slug-ft^2
IZZ	Second moment of inertia about z-axis	slug-ft^2
IXZ	Product of inertia	slug-ft^2
XVECN	Nominal state vector	Same as state vector
XIDTN	Nominal inertial velocity vector	ft/sec
IREF	=0 nominal state and control set to estimated nominal =1, nominal state and control set to true nominal	-
IAUG	=1, nominal state estimates used to get transformation matrix T =2, total state estimates used to get transformation	-
IEST	=0, for true nominal state used as independent regression variable =1, for the total state used as independent regression variable	-
IACT	=0 actuator model off =1 actuator model on	-
UNCON	4 vector static trim control regression intercepts	stick equivalent inches
UNREGC	4x3 array static trim control regression coefficients	-
ANGCON	2 vector static trim Euler angle regression intercepts	degrees
ANGRGC	2x3 array static trim Euler angle regression coefficients	-

TABLE 6.1-2

BLOCK DATA INITIALIZATION (Continued)

VARIABLES	DESCRIPTION	UNITS
VIND	Inertial velocity at each waypoint	ft/sec
XIND	Inertial position at each waypoint	feet
TIMD	Elapsed time at each waypoint	sec
IPLOT	Not used	-
IGNPT	=0, no gains printed out =1, gains printed out	-
INOMPT	=0, do not print independent variable for regression =1, print independent variable for regression of nominal state and control values	-

TABLE 6.1-3

PERTURBATION STATES

VARIABLES	DESCRIPTION	UNITS
x	Earth x-axis position	feet
y	Earth y-axis position	feet
z	Earth z-axis position	feet
u	Body x-axis velocity	ft/sec
v	Body y-axis velocity	ft/sec
w	Body z-axis velocity	ft/sec
p	Body x-axis angular rate	rad/sec
q	Body y-axis angular rate	rad/sec
r	Body z-axis angular rate	rad/sec
θ	Pitch Euler angle	rad
ϕ	Roll Euler angle	rad

TABLE 6.1-3

PERTURBATION STATES (Continued)

VARIABLES	DESCRIPTION	UNITS
ψ	Yaw Euler angle	rad
δ_B	Differential collective (rotor)	inches
δ_C	Collective (rotor)	inches
δ_S	Lateral cyclic (rotor)	inches
δ_R	Differential cyclic (rotor)	inches
δ'_B	Diff. collective rate (rotor)	in/sec
δ'_C	Collective rate (rotor)	in/sec
δ'_S	Cyclic rate (rotor)	in/sec
δ'_R	Differential cyclic rate (rotor)	in/sec
δ_{BA}	Diff. collective actuator	inches
δ_{CA}	Collective actuator	inches
δ_{SA}	Cyclic actuator	inches
δ_{RA}	Differential cyclic actuator	inches

6.2 TVHIS OUTPUT

The output of TVHIS is stored on Fortran logical unit 6 (TAPE6). Table 6.2-1 contains a summary of the output reports, a description and/or variable listing, and the subroutine which generates the report.

TABLE 6.2-1

OUTPUT FROM TVHIS

REPORT	DESCRIPTION-VARIABLE	SUBROUTINE
Integration Report	TIMD - initial time TIM1 - final time PRMT(3) - integration interval XVEC - state initial condition	MAIN

TABLE 6.2-1

OUTPUT FROM TVHIS (Continued)

REPORT	DESCRIPTION-VARIABLE	SUBROUTINE
Input Report	DTGUID - guidance time interval DTCNTR - control time interval DTOUTP - output time interval	RDDAT
Special Event Specification	ITYP - key to special event TEVENT - time of special event EVMAG - magnitude of special event	RDDAT
Flight Waypoint No.	NWPF, NWPL	RDDAT
Gain Regression Coefficients	ACONST, REGCOF	RDDAT
Initial Waypoint System Matrix	F0	MISION
Initial Waypoint Input Matrix	G0	MISION
Initial Waypoint Nominal Inertial Velocity	XIDT0	MISION
Initial Waypoint Nominal State	XVEC0	MISION
Final Waypoint System Matrix	F1	MISION
Final Waypoint Input Matrix	G1	MISION
Rows = Column =	Print matrix routine output description head plus number of row and columns	PRMR
Special Event Report	TIME	OUTP
DYDS = Guidance Off	Guidance perturbation input	OUTP
DXS = Guidance Off	State perturbation input	OUTP

TABLE 6.2-1

OUTPUT FROM TVHIS (Continued)

REPORT	DESCRIPTION-VARIABLE	SUBROUTINE
DVS = Guidance Off, Controller Off	Control perturbation input	OUTP
Time Histories	The total state and control vectors (PX) are x, y, z, u, v, w, p, q, r, θ, ϕ, ψ, δ_B, δ_C, δ_S, δ_R, $\dot{\delta}_B$, $\dot{\delta}_C$, $\dot{\delta}_S$, $\dot{\delta}_R$, δ_{B_T}, δ_{C_T}, δ_{S_T}, δ_{R_T}, $\dot{\delta}_{B_T}$, $\dot{\delta}_{C_T}$, $\dot{\delta}_{S_T}$, $\dot{\delta}_{R_T}$, $\dot{u}$, $\dot{v}$, $\dot{w}$, $\dot{x}_I$, $\dot{y}_I$, $\dot{z}_I$. Similarly, perturbation state and control vectors (PDX) are output	OUTP
Control Report	TIME - time in seconds V - augmented error state UCOM - control command	CNTRL
State Feedback Gains	CKX	CNTRL
Integrated Feedback Gains	CKV	CNTRL
Control Feedback Gains	CKU	CNTRL
X NOM EST U NOM EST	XVNEST - x nominal estimate	CNTRL
Perturbation States and Control	DX, DU	CNTRL
Control For Application	UCOM	
Integrated Output States	GDCMVC - guidance command vector V - augmented error state DUM2 - approximate time derivative of the augmented error state	CNTRL
Transformation Matrix T	T - transformation from state vector to output vector	TCOMP
Guidance Report	TIME	GUIDE
GDCMVC	Guidance command vector	GUIDE
Turbulence Included	Variation from subroutine TURC included	TURC

7. DETAILED DESCRIPTION OF TVHIS

TVHIS is designed as a realistic yet efficient helicopter simulation. It is based on a linear-time-varying helicopter model, including actuator and rotor dynamics models. The capability for simulating rate and displacement limits on the actuators also is incorporated. TVHIS includes a guidance subroutine which calculates total inertial velocity commands by summing the nominal velocity (specified by a nominal trajectory) and a velocity correction due to cross range and altitude errors. The guidance algorithm may be bypassed to allow testing of the control algorithm alone. TVHIS includes a digital-adaptive control routine that includes a state estimation section, a gain adaptation procedure, and the controller logic. The computations in TVHIS are accomplished by the following major subroutines: RDDAT, MISION, FLTINT, FGCOMP, RKGS, FCT, OUTP, CNTRL and GUIDE (see Figure 7-1).

RDDAT is the input routine which reads the flags and input parameters and which prints them out. The purpose of MISION is to pick the waypoints of the trajectory where position and velocity have been specified. Using the flight conditions, FLTINT linearly interpolates between the aerodynamic coefficients and then FGCOMP computes the system matrix.

RKGS has been incorporated from the IBM Scientific Subroutine Package (Ref. 4). The purpose of the Runge-Kutta method is to obtain an approximate solution of a system of first-order ordinary differential equations with given

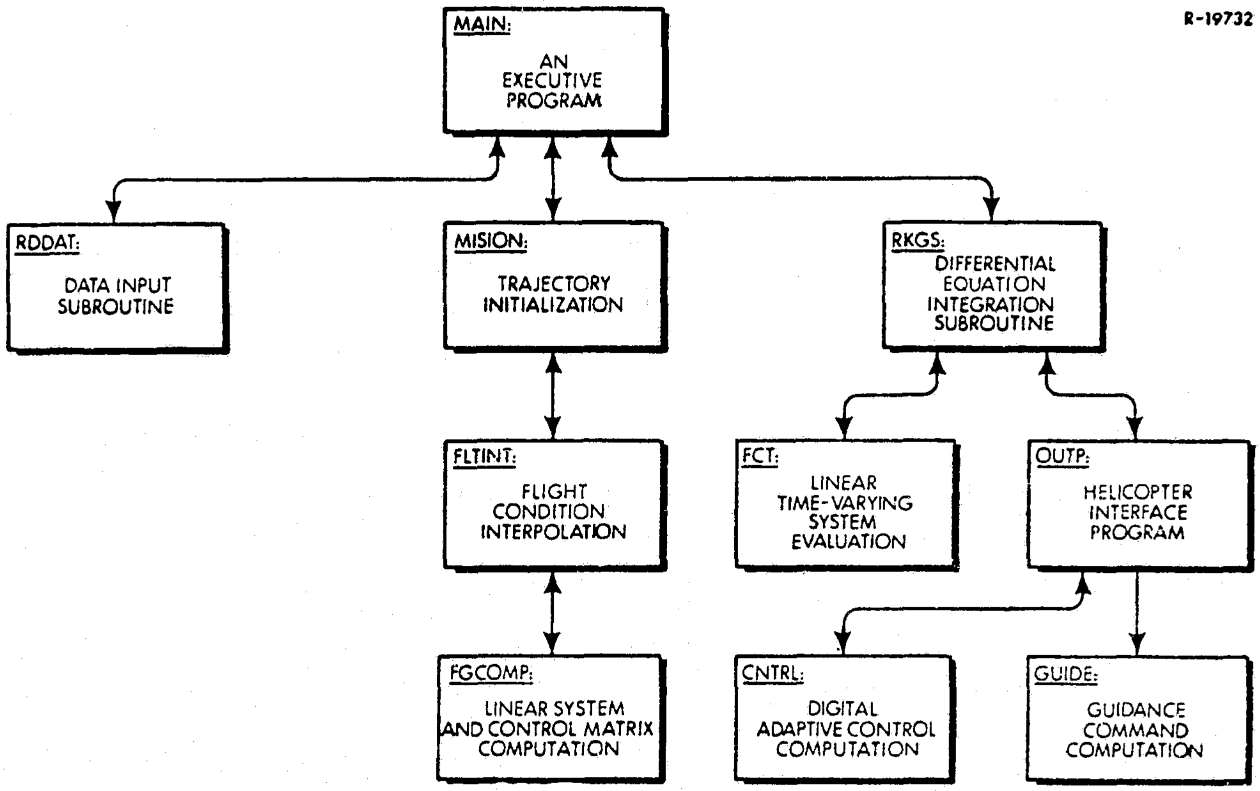


Figure 7-1 Overview of Program TVHIS

initial values. It is a fourth-order integration procedure which is stable and self-starting; that is, only the functional values at a single previous point are required to obtain the functional values ahead. For this reason, it is easy to change the step size (h) at any step in the calculations. Control of accuracy and adjustment of h is done by comparison of the results due to double and single step-sizes, 2h and h.

FCT evaluates the differential equation using the state and control vectors. OUTP is used to apply special events, to produce time history tables, and to call guidance and control algorithms, which are found in subroutine GUIDE and CNTRL, respectively.

GUIDE computes the vehicle guidance commands by combining the nominal command with a perturbation command due to inertial position error. The guidance algorithm may use time-varying gains.

CNTRL includes the state estimation logic and the controller algorithm. This logic computes the perturbation states, which are the differences between the nominal and total state estimates. If desired, the true state values may be taken as the estimates. The controller algorithm updates the control gains by applying the results of the controller gain regression of SCHED and calculates the control commands by applying the feedback law.

APPENDIX A

IMPLEMENTATION OF PERMANENT FILE FOR AERODYNAMIC COEFFICIENTS

The program which stores aerodynamic coefficients (AERO) must be run before DIGADAPT and TVHIS can be used. The purpose of this program is to create a permanent file on a disk. The file consists of 297 different cases of aerodynamic coefficients whose parameter variations are explained below.

The aerodynamic coefficients, corresponding to specific values of $\dot{\psi}$, $\dot{x}$ and $\dot{z}$ stored on the permanent disk file, are selected by computing a case number (NCASE) defined as

$$NCASE = (J-1)*99 + (K-1)*11 + L$$

where J is an index of 3 values for $\dot{\psi}$, K is an index of 9 values for $\dot{z}$, and L is an index of 11 values for $\dot{x}$. The $\dot{\psi}$ (radian/sec) values of 0.0, +0.05, -0.05 correspond to J = 1 to 3. The $\dot{z}$ (feet/min) value of -2000, -1500, -1000, -500, 0, 500, 1000, 1500, 2000 correspond to K = 1 to 9. The $\dot{x}$ (knots) values of -40, -20, 0, 20, 40, 60, 80, 100, 120, 140, 160 correspond to L = 1 to 11. For each case, there are 71 aerodynamic coefficients used in the formulation of the system matrix F.

APPENDIX B

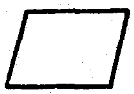
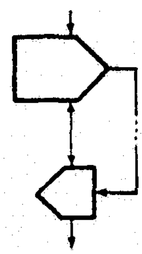
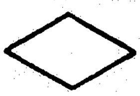
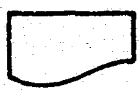
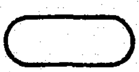
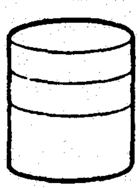
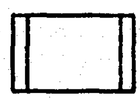
FLOWCHARTING SYMBOLS

Table B-1 gives the flowcharting symbols used in this manual. They have been defined in the ANSI standard (Ref. 5 and 6), with the exception of the off-page connector, which is an IBM defined symbol, and the loop symbol, which is a TASC defined symbol.

The symbol used within the off page connector, as in Fig. 3.1-1, identify the connector (by letter) and the connecting segment (by number in parentheses).

TABLE B-1

PREDEFINED PROCESS FLOWCHARTING SYMBOLS

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
	INPUT · OUTPUT		LOOP STRUCTURE
	PROCESS		
	DECISION		PUNCHED CARD OUTPUT
	DOCUMENT		
	BEGIN, END OR ENTRY		DISC STORAGE
	ON PAGE CONNECTOR		
	OFF PAGE CONNECTOR		PREDEFINED PROCESS FLOWCHARTING SYMBOLS

APPENDIX C

DESIGN PROCEDURES

The design procedure for determining a control law using the programs DIGADAPT and SCHED is outlined below (Ref. 1). The procedure indicates at each step which method was chosen for this report; alternate steps which could have been attempted are also provided.

- Sampling Interval

  Decide on covariance matrix for gust states.
  Add any other system noises.
  Specify the state vector bounds.
  Propagate state covariance for various flight conditions until the state vector bound is reached.

- Flight Conditions for Point Design

  Choose a representative number of the most common flight conditions.
  Separate the flight conditions into two groups, one near hover, the other at higher speeds. Near hover, constant gains can be chosen; mode switching occurs to the regressed gains after a specified forward velocity is exceeded.

- Command-Controller

  Choose the command system and controller type for continuous or discrete case, e.g.,
  Velocity Command (PII Controller),
  Attitude Command (PI Controller), or
  Dynamic Controller (Regulator).

- Criteria

  Specify suitable step response requirements, closed-loop eigenvalues, or quadratic weightings.

Determine the state variables to be weighted by quadratic synthesis. Specify Q and R adjustment policy if required.

- <u>Controller Gain Scheduling</u>

  Obtain control gains from DIGADAPT. Iterate Q,R as required. Punch out controller gains and flight conditions.

  Choose flight conditions for regression (body axes or inertial axes). Reorder punched cards with chosen flight condition type.

  Run SCHED for all combinations of independent variables (flight conditions), printing out correlation coefficients.

  Using SCHED output choose independent variables for regression based on correlation coefficients and availability of measurements onboard the vehicle.

  Run SCHED for regressions with chosen independent variables.

  Record regression coefficients for use in open-loop explicit-adaptive gain scheduling algorithm.

- <u>Trim Values</u>

  Run DIGADAPT for as many flight conditions as desired, punching out control trim positions and state positions.

  Choose body or inertial axes for independent variable conditions.

  Schedule static trim values against flight condition.

  Schedule dynamic trim coefficients against flight condition.

- <u>Low-Pass Filters</u>

  Establish signal and noise variances for measurement states.

  Define filter gains using discrete-time Kalman filter algorithms.

- Complementary Filters

  Choose a sampling time for the filters if they are to be discrete. Filters are simple enough to be adjusted on line or by full simulation techniques.

- Kalman Filter Design for Unmeasured States

  Define states which are measured, disturbance covariance matrix, and observation noise covariance matrix.

  Run DIGADAPT to obtain Kalman filter gains for flight conditions desired. Run SCHED with Kalman gains as noted above.

APPENDIX D
LIST OF DELIVERABLES

Delivered with this document are thirteen distinct, computer-related items: one magnetic tape, six executable card decks, and six computer-printed listings. These thirteen items are detailed in the following list.

1. Magnetic tape, 7-track, 556 bpi (HI) containing five files.

   File 1 - (binary) DIGADAPT program library
   File 2 - (binary) SCHED program library
   File 3 - (binary) TVHIS program library
   File 4 - (binary) AERO program library
   File 5 - (coded) aerodynamic coefficient data

2. Executable deck which reads item 1 (above) and creates four permanent files.

   DIGOBJ - DIGADAPT object code
   SCHOBJ - SCHED object code
   TVHOBJ - TVHIS object code
   AERO - data file of aerodynamic coefficients, read by DIGADAPT and TVHIS

3. Executable deck - DIGADAPT sample execution #1

4. Executable deck - DIGADAPT sample execution #2

5. Executable deck - DIGADAPT sample execution #3

6. Executable deck - SCHED sample execution

7. Executable deck - TVHIS sample execution

8. Computer printout generated by execution of item 2 - includes compiler-generated source listings of DIGADAPT, SCHED, TVHIS, and AERO and the execution-time output of program AERO.

9. Computer printout generated by execution of item 3 - DIGADAPT sample output #1.

10. Computer printout generated by execution of item 4 - DIGADAPT sample output #2.

11. Computer printout generated by execution of item 5 - DIGADAPT sample output #3.

12. Computer printout generated by execution of item 6 - SCHED sample output.

13. Computer printout generated by execution of item 7 - TVHIS sample output.

REFERENCES

1. Stengel, R.F., Broussard, J.R. and Berry, P.W., "The Design of Digital Adaptive Controllers for VTOL Aircraft," The Analytic Sciences Corporation, TR-640-1, Reading, Massachusetts, September 1975.

2. CYBERNET Scope 3.3 Reference Manual - Batch and Remote Batch Operating System, Publication No. 8661800, CDC CYBERNET Publications Department, Minneapolis, Minnesota, 1973.

3. Sandell, N.R. and Athans, M., "Modern Control Theory: Computer Manual," M.I.T. Center for Advanced Study, Cambridge, Massachusetts, 1974.

4. System/360 Scientific Subroutine Package, Version III, Publication No. H20-0166-5, IBM Technical Publications Department, White Plains, New York, March 1970.

5. American National Standard Flowchart Symbols and Their Usage in Information Processing, Publication No. AIM1171/375, ANSI, Inc., New York, New York, 1971.

6. IBM Flowcharting Techniques, Publication No. C20-8152-1, IBM Technical Publications Department, White Plains, New York, March 1970.