Copy No. 11

RESEARCH REPORT

ON

OPTIMAL CONTROL FOR SATELLITE ATTITUDE MANEUVERS

(Contract No. NAS 5-10191 and NAS 5-10349)

APR 1968

VOLUME 2: COMPUTER PROGRAMS

GPO PRICE	$
CFSTI PRICE(S)	$
Hard copy (HC)	3.00
Microfiche (MF)	.65

ff 653 July 65

(ACCESSION NUMBER)	(THRU)
137	1
(PAGES)	(CODE)
CR 94678	30
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

FACILITY FORM 602

RQ750402A

PART VII

DIGITAL AND HYBRID SIMULATOR COMPUTER PROGRAMS

Preface

One of the principal results of this work was a computer program incorporating the analysis described in the first book of the report. This program contained the control computation in a digital program and the dynamics available either digitally or in an analog computer for hybrid computation. The program writeup contained here was separated from the main body since it was felt that the actual programming and computation instructions would not have as wide an audience.

The description of the analog computer flow and the hybrid setup contained in section 5 - 9 were prepared by Computing and Software, Inc. to whom we express our appreciation for their contribution to the completeness of this report.

The description of the digital program in sections 1 - 4 is, of course, the responsibility of The Mathematical Sciences Group.

PART VII

TABLE OF CONTENTS

Page No.

Preface.. 1

7.1 Digital Program Description.......................... 3

7.2 Verification Information............................. 18

7.3 Flow Charts For Digital Program...................... 40

7.4 Digital Program Listings............................. 52

Ideal Model.. 53

Sensor Model... 70

Elasticity Model..................................... 91

7.5 Simulation Model For Analog Computer................. 108

7.6 Overall Hybrid Simulation............................ 112

7.7 Simulation Setup Procedures.......................... 114

7.8 Analog Patching...................................... 115

Initial Condition Setup.............................. 116

Dynamics & Control................................... 118

Thrustor Misalignments............................... 126

Thrustor Real Effects & Fuel Consumption......... 130

7.1 *Description of a Computer Program for Optimal Control Synthesis and Satellite Attitude Manuevering Simulation.*

I. *Subprogram Description*

The program is written in Fortran IV for the SDS 9300; an alternate version is acceptable to the IBM 7094. Following are the names of the various subprograms together with their approximate word length: "MAIN" - 1400; "INIT" - 510; "STUP" - 116; "TOR" - 1403; "REG" - 1130; "DTOA" - 143; "ATOD" - 66; "OUT" - 86; "RK" - 270; "DER" - 222. Operation of the program may be purely digital, or hybrid in conjunction with the HYDAC 2000. In the former mode both the control synthesis and the dynamics are done digitally; in the latter, the attitude simulation is done by the HYDAC 2000, angles and rates being sent to the 9300, it determining applicable controls and transmitting these back to the analog computer.

"MAIN" is the executive branch of the program. It reads all input quantities, determines the mode of operation, calls upon the various subroutines at the proper time, does some computation necessary for initialization and ground track determination and decides when a run is to be terminated and a new run is to begin. Further, in optional versions of the program which include sensor and elastic effects these calculations are also done in "MAIN".

Subroutine "INIT" does the major portion of the initialization prior to actual execution. Input to the program is in terms of latitude and longitude on the earth, whereas control is with respect to a coordinate system whose origin is in the satellite. "INIT" determines the initial angles and angular rates in this coordinate system.

"STUP" determines the elements of the three by three matrix, C, referred to in the section on coordinate systems. Three angles are furnished as arguments of the subroutine; the nine matrix elements are computed.

"TOR" determines torques on the three principal axes which are caused by solar pressure and the earth's gravity gradient. It was demonstrated by a series of runs that these torques are sufficiently small so that they may be replaced by small constant torques without essential change in either angles or rates during the course of a single slewing maneuver. Since this subroutine requires over 1400 storages, as various additions and modifications were made to the program it was found necessary to replace it by a dummy routine in order to comply with storage restrictions on the 9300.

"REG" incorporates all normalization and logic for the determination of control. After all angles and rates have been sampled "REG" is called to determine for each of the three axes whether the control torque should be positive, negative or zero, considering the weighting of the cost function parameters. "REG" then further considers the estimated arrival time of all three axes assuming these control policies are adopted. It then modifies these policies to prevent large differences in arrival times while guaranteeing that the maximum overall time to reach the target is not increased. Controls are then transmitted to "MAIN".

"DTOA" is a digital-analog interface subroutine which transmits three controls and three disturbance torques to the analog computer each time the subroutine is called upon by the timed interrupt.

"ATOD" is a similar subroutine to "DTOA", except that here three angles and three angular rates are transmitted from the analog to the digital computer.

"OUT" is the output subroutine which does all the printing.

"RK" does the numerical integration of the six differential equations via the usual fourth order Runge Kutta formulas. When elastic effects are included in the program ten differential equations are integrated.

"DER" merely calculates the values of the six or ten derivatives involved in the dynamics. It is called upon by "RK" four times for each complete forward step.

II. *Input to Ideal Model Program*

Card 1: FORMAT (2I5, 5F10.5)

Field 1: NA serves two purposes. If zero the hybrid computer is to be used, control being handled digitally while the dynamics are simulated on the analog. If positive the program is to be executed digitally only, in which case NA is the number of integration steps to be taken between samplings for possible control changes.

Field 2: NB is the number of samplings between printouts. Thus if NB is 1 output will be printed at each sample time; if NB is 90000 no output will occur in a normal run.

Fields 3-5: PA, PB and PC are all scale factors which are used in transmitting torques, angles and angular rates digital to analog so as to keep the voltages on the analog computer within the range of the equipment. These factors depend naturally upon the wiring of the analog patch board. The most recently used factors were -5.0×10^3, 5.0 and 250.0 .

Field 6: Z is merely a dummy variable to indicate whether the disturbance torques are to be calculated from the subroutine "TOR" or whether they are to be read in as constants on card 6. If Z equals 0 the subroutine is to be used, if not the torques are to be read in.

Card 2: FORMAT (9F8.5)

Fields 1-3: ZI(1-3) are the moments of inertia about the three principal axes, the latest values being 3580, 1970 and 2000 slugs/ft^2.

Fields 4-6: ZM(1-3) are the thrusts exerted by the jets on the three principal axes, the latest values being .08, .04 and .12 lb. respectively.

Field 7: AL(1) is a multiplier of the cross-coupling terms in the differential equations and gives the user the capability of removing these terms from the digital dynamics by setting it to zero. For the complete dynamics AL(1) = 1; for single axis dynamics AL(1) = 0.

Field 8: AL(2) is unused and should be unpunched.

Field 9: AL(3) is the longitude of the satellite, i.e., the satellite sub-point, with the convention that a negative quantity represents a west longitude and a positive an east longitude. Since the satellite has usually been assumed to be over the United States, -100. has been most commonly used.

Card 3: FORMAT (9F8.5)

Fields 1-9: ((ZL(I,J), J = 1,3), I = 1,3) are the weighting parameters which determine the cost function to be minimized. The index I denotes the axis, the index J refers to the parameter being weighted, i.e., time, fuel or energy. Although the capability for independent weighting on different axes exists, no application has been made of this feature.

Thus, fields 4-6 and 7-9 are repetitions of 1-3. For a run to be strictly time optimal card 3 should read 1, 0, 0, 1, 0, 0, 1, 0, 0. Such a run, for a Mohave-Quito maneuver, takes 131 seconds. The same run with card 3 reading 1, 2, .00005, 1, 2, .00005, 1, 2, .00005 takes 178 seconds and uses about one-half the fuel. With card 3 reading 1, 8, .00005, 1, 8, .00005, 1, 8, .00005, 1, 8, .00005 the run takes 306 seconds and uses about one-third the fuel.

While it is true that the weighting parameters for a cost function such as this should sum to one, for computational purposes it is not necessary here. In order to study the effects of varying these parameters it is more feasible to fix two of them and vary a third. It is reasonable to set ZL(1,J) equal to 1 for all cases. If one wishes to study variations of ZL(2,J) then fixing ZL(3,J) at .00005 will permit any positive values for ZL(2,J). However, values larger than 10 will result in no significant fuel savings. Similarly, to study ZL(3,J) setting ZL(2,J) at .00005 permits all positive values for ZL(3,J). In this case the maximum rate is limited by approximately

$$\sqrt{\frac{1}{ZL(3,J)}}\ .$$

Card 4: FORMAT (9F8.5)

Field 1: AX is used in time synchronization, the first step of which is to find which of the three angles will require the longest time to reach the origin. The estimated time for this angle is then multiplied by AX and this product called TMAX. TMAX is now compared to the estimated time for each of the other angles to determine whether control policy should be changed to make arrival times more nearly the same.

Field 2: BX is a safety factor for the time synchronization. The estimated time of arrival for any axis whose thruster is to be cut off for time synchronization purposes will be at least BX seconds less than TMAX.

Field 3: XF was originally used to prevent zig-zagging on the final leg of the path to the origin. 10. is a good value for this parameter.

Field 4-6: PIB, ROB, YIB are the angles (in radians) of misalignment of the "B" system with respect to the "P" system and are normally assumed to be zero.

Card 5: FORMAT (9F8.5)

Field 1: YO is the angle of rotation about the pointing axis 2, in degrees, when the body is in the holding mode. This angle must always be very small and is usually input as zero.

Fields 2-3: AT and ONG are the initially assumed latitude and longitude in degrees to which the satellite is pointing when the maneuver is to begin.

Fields 4-5: XLAT and XLON are the latitudes and longitudes of the target in degrees.

Field 6: DT is the time increment (in seconds) which is used as the step size in the integration subroutine.

Field 7: AE was an experimental parameter which is now fixed at 5. Its purpose is to prevent chattering along the rate limiting line $\omega = S$. It sets a minimum band width for which thrusters are to be cut off. Its units are radians per second. However it must be multiplied by the ratio of moment of inertia to thrust before being input. Since this ratio is approximately 20,000, the value 5 actually represents a band width of about .00025 radians/sec.

Card 6: FORMAT (9F8.5) - Optional

Fields 1-3: T(1), T(2), T(3) are the disturbance torques in foot-pounds which are used to replace the calculated torques from TOR when Z of card 1 is nonzero. If Z is zero this card should not appear.

Card 7: FORMAT (9F8.5)

Fields 1-3: Elevation misalignment angles for the positive axes 1-3.

Fields 4-6: Azimuth misalignment angles for the positive axes 1-3. All angles are in degrees.

Card 8: FORMAT (9F8.5)

Fields 1-6: This card is the counterpart of card 7 for negative axes.

III. *Input for Sensors*

The input for the operation of the program with sensor real effects differs only in card 4 from that of the ideal model. Fields 1-3 are AX, BX and XF as before. Fields 4 and 5 are ZK1 and ZK2, which correspond to parameters K1 and K2 in the sensor write-up of the technical report, and are currently .1 and .005. Field 6 is SD, and is used in the random number generator to determine the standard deviation currently 0.00063. Fields 7-9 are PIB, ROB and YB, which in the ideal model input appeared in Fields 4-6.

The values of ZK1, ZK2, and SD are not arbitrary but are very much determined by satellite system specifications. The value for SD has been carefully derived on the basis of oral communications with the appropriate GSFC technical persons. That derivation is given in Appendix F, Part 1 (Filter specification (choice of K)). The values of ZK1 and ZK2 are intimately involved with the signal noise SD, the sampling interval δ and also the modeling errors. This dependence is described also in Appendix F, Part 1 (computation of K.) Referring to Appendix F, Part 1, Fig. 3, we can see the region from which possible (not necessarily good) values of K may be selected.

IV. *Input for Elastic Effects*

Card input here is identical to that of the ideal model through card 5. Card 6 uses fields 1-3 for the three disturbance torques as before. Field 4 is for the length of the boom in feet. Field 5 is for the mass of the boom in slugs. Field 6 is for the spring constant in slugs/sec^2 and field 7 the damping coefficient, which is dimensionless. It should be noted that the moment of inertia of the elastic boom is already included in the moments of inertia about the principal axes.

V. *Operating Instructions*

Several sense switch options are available to the operator. When running digitally sense switch 1 may be set, whereupon computations will halt and a new print frequency, NB, is to be typed in FORMAT (15) on the on-line typewriter. Before depressing the carriage return switch 1 should be reset to prevent halting again. The print frequency may be changed any number of times during a run using this procedure.

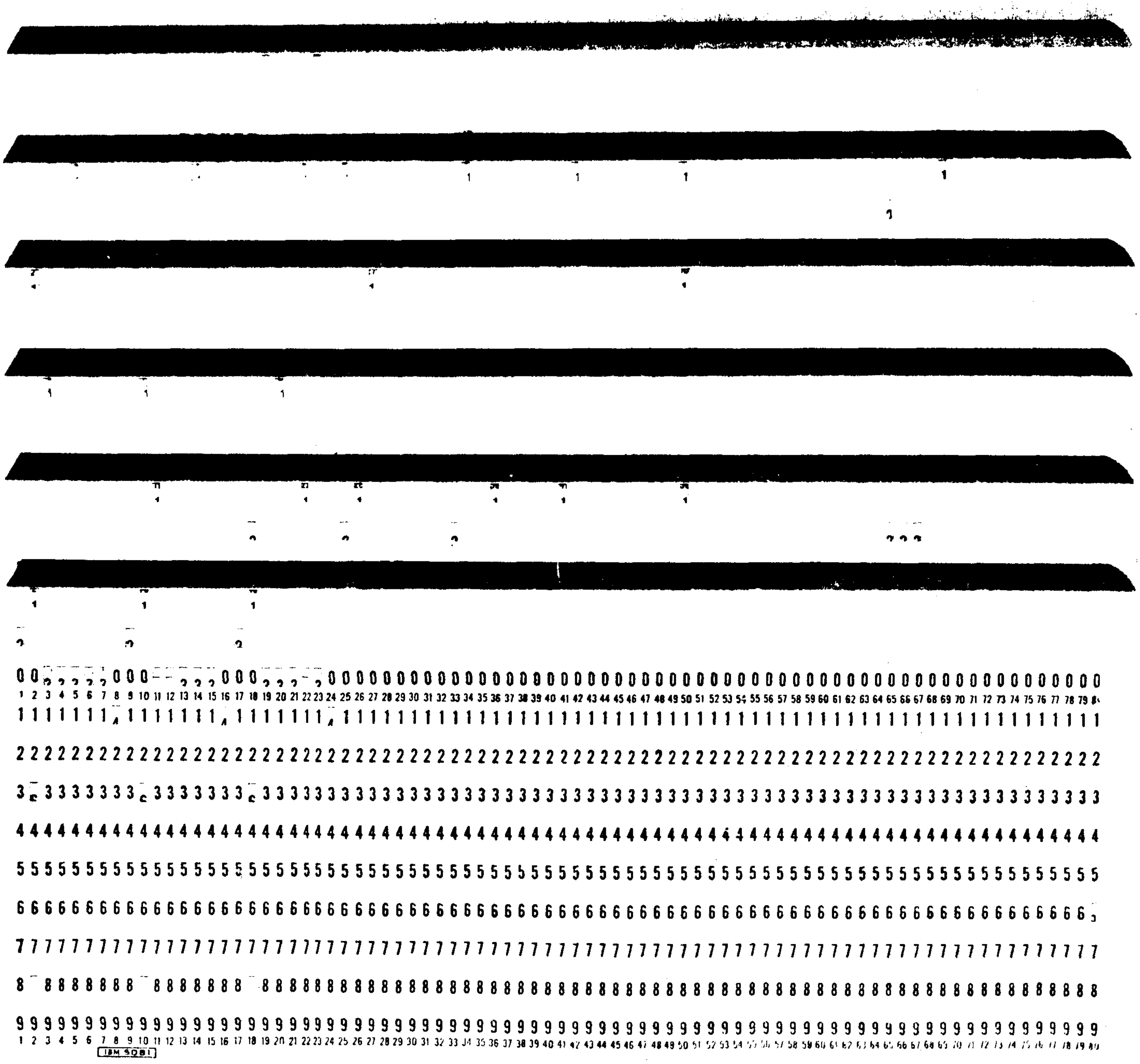
Switch 2 is used for terminating a run and reading in new data. When switch 2 is set computations will cease, and terminal conditions printed if in digital mode. After inserting new data into the card reader it is only necessary to push the carriage return to start a new run.

When operating in the hybrid mode when the satellite has reached its target control will be cut off, and no communication is possible between HYDAC and the 9300 until at least one plotter pen has been removed from the origin and then switch 3 set. Hybrid operation will resume at this time.

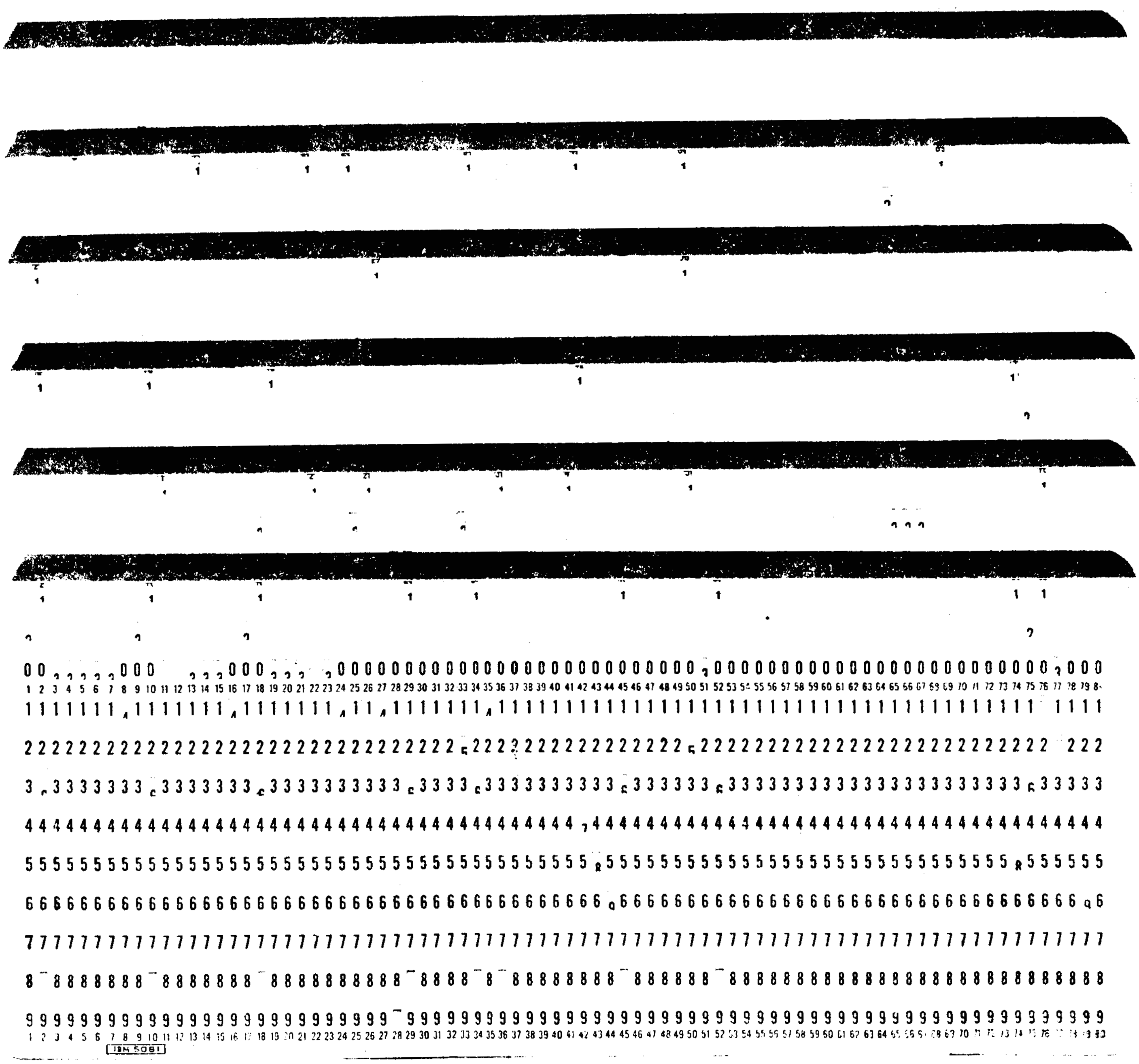
Setting sense switch 4 allows one to type in YO, AT, ONG, XLAT and XLON in FORMAT (5F5.3) after all data cards have been read in. These typed values will override the corresponding values already read in. However, this must take place at the beginning of a run before internal looping has begun.

In all digital runs when the target is reached terminal conditions will be printed on the on-line printer and "PAUSE 16" will appear on the typewriter. Inserting a new set of data and pushing the carriage return will start a new run. In hybrid runs after initial conditions are printed computations will cease and "SET ANALOG - PAUSE 11" will be typed. After setting up the HYDAC the run is initiated by the carriage return.

SAMPLE INPUT FOR HYBRID RUN (IDEAL MODEL)



SAMPLE INPUT OF ELASTICITY RUN (DIGITAL)



OUTPUT

Input from cards 7 and 8 are printed first in the same order. Two 3×3 matrices are then printed, these being multipliers corresponding to positive and negative control torques. If we refer to the elements of the first matrix as A_{ij1} and those of the second as A_{ij2}, then let

$$b_{ij} = \begin{cases} A_{ij1} & \text{if } U_i \geq 0 \\ A_{ij2} & \text{if } U_i < 0 \end{cases},$$

where U_i is the current control torque for axis i. The effective control torque $\overline{U}_i$ is then given by

$$\overline{U}_i = \sum_{j=1}^{3} b_{ij} U_j$$

Following a page eject the other input parameters appear as follows.

Line 1: All weighting factors ((ZL(I,J),J=1,3),I=1,3)

Line 2: Yaw angle, initial latitude and longitude, target latitude and longitude, angles ρ,Y and π in the BS system, DT.

Line 3: AX, BX XF, AE.

Line 4: T(1-3), followed by the scaled torques for HYDAC.

The body of the run appears next, and for the ideal model its format follows.

Line 1: Time, ρ, Y, π, ω_1, ω_2, ω_3, latitude and longitude of the pointing vector.

Line 2: ρ, Y, π, ω_1, ω_2, ω_3.

Line 3: Control torques (u_1, u_2, u_3), T(3), normalized ρ, normalized ω_1, S.

The body of a run with sensors has identical lines 1, 2, and 3 as that of an ideal model run. An additional line appears, however, which contains filtered angles ρ, Y, π, estimated ω_1, ω_2, ω_3 and unfiltered noisy angles ρ, Y and π.

The body of an elastic run has the following appearance.

Line 1: Time, ρ, Y, π, ω_1, ω_2, ω_3, q_1 and q_3, (displacements of the point mass with respect to axes 1 and 3), $\dot{q}_1$ and $\dot{q}_3$.

Line 2: $\dot{\rho}$, $\dot{Y}$, $\dot{\pi}$, $\dot{\omega}_1$, $\dot{\omega}_2$, $\dot{\omega}_3$, $\dot{q}_1$, $\dot{q}_1$, $\ddot{q}_3$ and $\ddot{q}_3$.

Line 3: XM, XL, D(13)-spring constant, D(14)-damping coefficient.

Line 4: u_1, u_2, u_3, ground track latitude and longitude.

7.2 *Verification Information*

7.2.1 *Introduction.*

An integral part of the construction of any digital program is the checkout. This involves establishing that the formulae which represent the analysis have been correctly mechanized and also verifying that the analysis is true, perhaps by showing that it works for simple cases such as, in our program, single axis slews.

A great variety of experience with the program, only a portion of which is described in the two parts of this report, has given us great confidence that it simulates the motion correctly. Part of this confidence stems from the modular construction, enabling us to use established subroutines in many places.

7.2.2 *Euler Angle Calculations.*

An elementary test for initial condition calculation involves the single-axis relations. That is, the computation of the Euler angles (ϕ_{br}, ψ_{br}, θ_{br}) of the body in the target system when only one angle is perturbed. This procedure works, for instance, when the satellite points to its subpoint with no yaw and the target lies in the equatorial plane, $\theta_{br} \neq 0$.

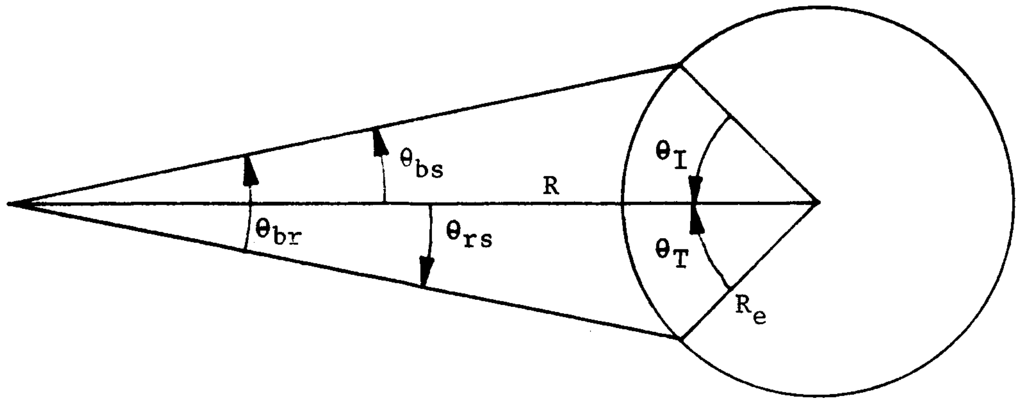
Figure 7.2.1 shows

$$\theta_{br} = \tan^{-1} \frac{\sin \theta_T}{K - \cos \theta_T} - \tan^{-1} \frac{\sin \theta_I}{K - \cos \theta_I}$$

where θ_T is target longitude, positive east of Greenwich.

θ_I is initial longitude, positive east of Greenwich.

K = 6.610808, the ratio of the orbit radius of a synchronous satellite to the radius of the earth, $= \frac{R}{R_e}$.



$$\theta(\theta_T, \theta_I) = \theta(\theta_T, 0) - \theta(\theta_I, 0),$$

and

$$\theta(-\theta_T, 0) = - \theta(\;_T, 0) \quad .$$

Therefore, only the basic curve $\theta(\theta_T, 0)\ 0 \leq \theta_T \leq 90$ is really necessary.

For latitude calculations, note that

$$\phi(\phi_T, \phi_I) = - \theta(\theta_T, \theta_I) \quad .$$

For check purposes, several points on the curves were computed by the machine and marked with x's, the calculations being done both in latitude and longitude.

A more detailed analysis for three axis freedom was then performed. It checks explicitly only the calculation of the initial euler angles $(\phi, \psi, \theta)_{br}$ as determined by target latitude and longitude, initial latitude and longitude, initial "yaw" angle y, and the longitude of the satellite subpoint. Actually, however, these checks are rather more comprehensive, since the calculations are performed in a subroutine which is used elsewhere in the program.

The following procedure was followed in checking the transformations.

The pertinent formulae were derived by persons not involved in the original problem difinition. These formulae were programmed on a small digital computer using a non-Fortran language and not referring to the code presently used in the simulator.

Then a number of spot checks were run and the results compared with those from the simulator. These results appear in Table 7.2.1.

The formulae used in the small computer are:

$$\tan \theta_{rs} = - \frac{\cos \phi_T \sin \theta_T}{K - \cos \phi_T \cos \theta_T}$$

$$\tan \phi_{rs} = \frac{\sin\phi_T}{\cos \theta_{rs}(K-\cos \phi_T \cos \theta_T)-\cos \phi_T \sin \theta_T \sin\theta_{rs}}$$

ϕ_T = target latitude

θ_T = target longitude

K = 6.610808 ratio of synchronous orbit radius to earth radius.

θ_{rs}, ϕ_{rs} are the pitch and roll angles through which the S system must be rotated, pitch (θ_{rs}) first, to align with the R (target) system.

$$\tan \theta_{Is} = - \frac{\cos \phi_I \sin \theta_I}{K-\cos \phi_I \cos \theta_I}$$

$$\tan \phi_{Is} = \frac{\sin \phi_I}{\cos \theta_{Is}(K-\cos \phi_I \cos \theta_I)-\cos \phi_I \sin \theta_I \sin \theta_{Is}}$$

ϕ_I = Initial latitude

θ_I = Initial longitude.

θ_{Is}, ϕ_{Is} are the pitch and roll angles through which the S system must be rotated to point at the initial latitude and longitude and maintain minimum angle between the body 3-axis and north.

We now assume an arbitrary rotation of $-y$ takes place about the body 2-axis. This rotates the three axis out of a plane defined by the satellite center and a north pointing vector through the target, and thus increases the angle between the polaris sensor and north.

This will not be the body yaw because yaw is the second euler angle. This is the yaw angle of a second set of euler angles.

Using this yaw angle along with θ_{Is} and ϕ_{Is} we can define a new set of angles θ_{bs}, ψ_{bs}, and ϕ_{bs} which are the euler angles of the body with respect to the S system including the yaw angle y.

These are given by:

$$\psi_{bs} = \sin^{-1}(\sin y \cos \phi_{Is})$$

$$\phi_{bs} = \sin^{-1}(\sin \phi_{Is}/\cos \psi_{bs})$$

$$\theta_{bs} = \sin^{-1}\left((\sin \theta_{Is} \cos y - \sin y \sin \phi_{Is} \cos \theta_{Is})/\cos \psi_{bs}\right).$$

Having the euler angles of the body with respect to the S system and of the R system with respect to the S system, we can solve for the euler angles of the body with respect to the R-system.

These are:

$$\psi_{br} = \sin^{-1}\left(\sin \phi_{rs} \cos \psi_{bs} \sin(\theta_{rs} - \theta_{bs}) + \cos \phi_{rs} \sin \psi_{bs}\right)$$

$$\theta_{br} = \mathrm{sgn}(\theta_{bs} - \theta_{rs}) \cos^{-1}\left((\cos \psi_{bs} \cos(\theta_{rs} - \theta_{bs}))/\cos \psi_{br}\right)$$

$$\phi_{br} = \sin^{-1}\left((\cos \phi_{rs} \sin \phi_{bs} \cos \psi_{bs} - \sin \phi_{rs} \cos \phi_{bs} \cos(\theta_{rs} - \theta_{bs})\right.$$

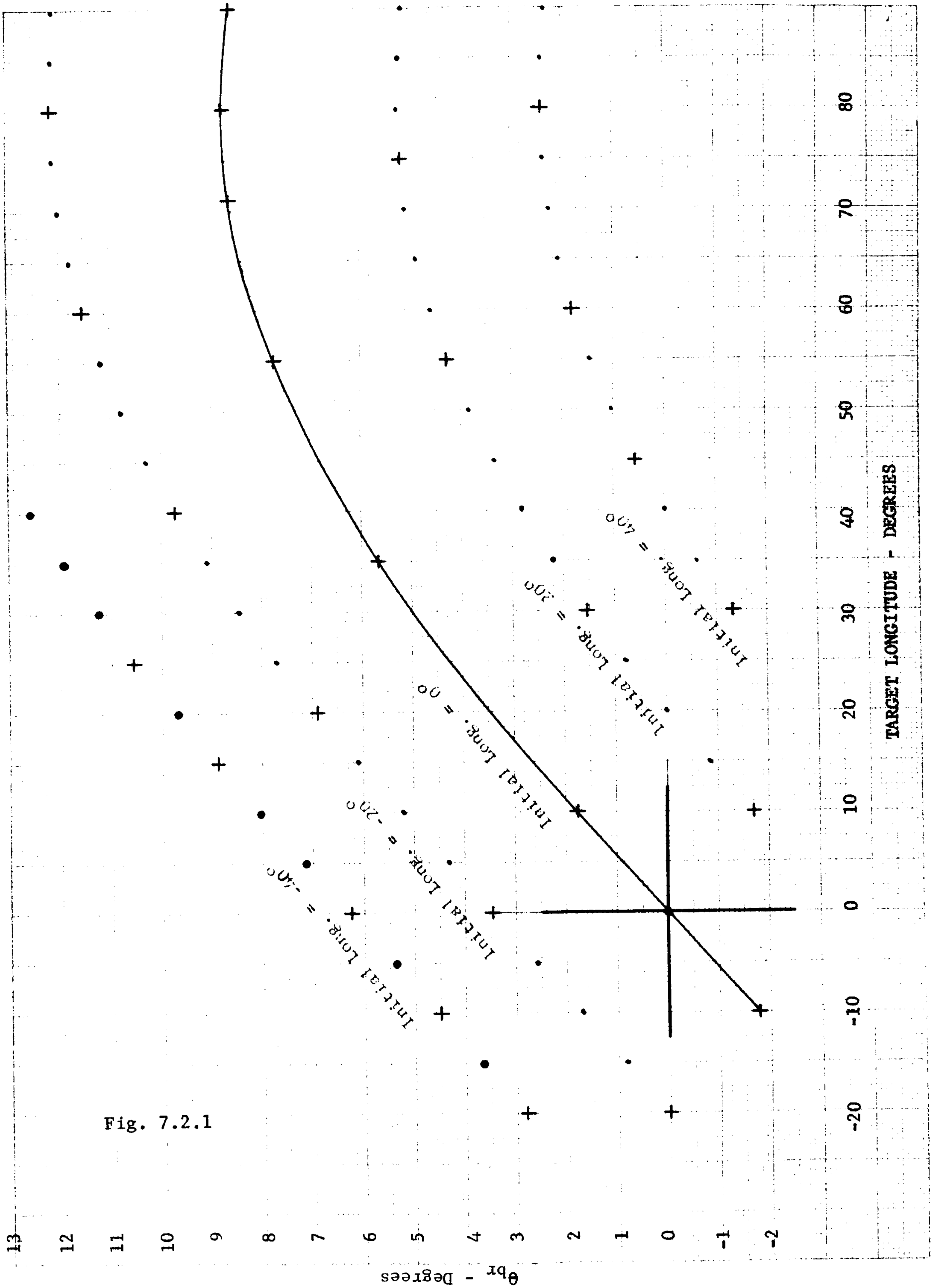
$$\left. - \sin \phi_{rs} \sin \phi_{bs} \sin \psi_{bs} \sin(\theta_{rs} - \theta_{bs}))/\cos \psi_{br}\right)$$

Theoretical investigations show that the largest yaw (y_{br}) which can occur "naturally" (initial yaw zero) is in magnitude less than 2.3°. Notice that the largest value appearing here is about 0.8° and occurs on the Quito-Mojave maneuver.

Since both of these calculations were carried out on digital computers, it is proper to ask why even small discrepancies appear. A possible source is the different computers and different input routines. However, we feel that the primary cause is a poorly chosen formula used in the checks - i.e., the simulator is probably more accurate. (Use of inverse cosine in the vicinity of $\theta = 0$.)

Init Lat (deg)	Init Long (deg)	Init Yaw (deg)	Target Lat (deg)	Target Long (deg)	Calc ϕ (rad)	Calc ψ (rad)	Calc θ (rad)	Simulator ϕ (rad)	Simulator ψ (rad)	Simulator θ (rad)
20	30	1	5	-10	.04337	.01916	-.11233	.04337	.01916	-.11233
20	30	5	5	-10	.04353	.08891	-.11537	.04354	.08891	-.11536
20	30	10	5	-10	.04404	.17609	-.11922	.04405	.17609	-.11922
30	10	30	-20	15	.16878	.51842	-.06761	.16876	.51841	-.06761
-30	10	30	20	15	-.16858	.51645	.10047	-.16857	.51644	.10046
-30	-30	30	20	15	-.16560	.51069	.19868	-.16559	.51068	.19867
-.2	-78	0	35.3	-1.17	-.09884	-.01470	.14800	-.09882	-.01470	.14800
-.2	-78	0	35.3	-117	-.08259	-.00392	.04765	-.08258	-.00392	.04761
35.3	-117	0	-.2	-78	.08268	-.00003	-.04781	.08268	-.00003	-.04780
35.3	-116.9	0	-.2	-78	.09932	.00007	.10649	.09932	.00007	.10648
35.3	-116.9	0	-.2	-78	.08270	-.00003	-.04770	.08270	-.00003	-.04770
0	0	-10	80	0	-.15412	-.17251	-.02666	-.15412	-.17250	-.02666
0	0	10	80	0	-.15412	.17251	.02666	-.15412	.17250	.02666
0	80	10	0	0	0	.17453	-.15181	0	.17453	-.15181
0	-80	10	0	0	0	.17453	.15181	0	.17453	.15181

TABLE 7.2.1.



TARGET LONGITUDE - DEGREES
θ_{br} - Degrees
Initial Long. = -40°
Initial Long. = -20°
Initial Long. = 0°
Initial Long. = 20°
Initial Long. = 40°
-20
-10
0
10
20
30
40
50
60
70
80
13
12
11
10
9
8
7
6
5
4
3
2
1
0
-1
-2

Fig. 7.2.1

Then

$$I_o = \hat{g}^T I \hat{g}$$

which agrees with the equations 2) and 3) in Appendix C.

To check this work further the transformations between the MSG and (3652) systems were substituted directly into (C2) and the same results were obtained in the MSG system.

II. Torque Calculation.

From (3652) we have

$$T_b^T \cdot B = - [U_\theta, U_\Psi, U_\phi], \text{ so}$$

$$T_b = - \begin{bmatrix} \sin\Psi & \cos\Psi \ \sin\phi & \cos\Psi \ \cos\phi \\ 0 & -\cos\phi & \sin\phi \\ 1 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} U_\theta \\ U_\Psi \\ U_\phi \end{bmatrix}$$

or

$$T_b = - \begin{bmatrix} 0 & 0 & 1 \\ \dfrac{\sin\phi}{\cos\Psi} & -\cos\phi & -\dfrac{\sin\phi\ \sin\Psi}{\cos\Psi} \\ \dfrac{\cos\phi}{\cos\Psi} & \sin\phi & -\dfrac{\sin\Psi\ \cos\phi}{\cos\Psi} \end{bmatrix} \begin{bmatrix} U_\theta \\ U_\Psi \\ U_\phi \end{bmatrix}$$

This checks equation 11) of the Appendix C, compared with (C7) of (3652).

$$U_\theta = 2I_{11}a_1(\cos\theta\ \cos\Psi) + 2I_{22}a_2(-\cos\theta\ \sin\phi\ \sin\Psi$$
$$- \sin\theta\ \cos\phi) + 2I_{33}a_3(\sin\theta\ \sin\phi - \cos\theta\ \cos\phi\ \sin\Psi)$$

7.2.3 *Verification of Gravity Gradient Torques*

Our first task is to verify that the correct equations are used. To do this we compare NASA TN D-3652 with Appendix C.

I. Moment of inertia Calculation

On page 33 of 3652 appears the equation

$$I_o = I_{xx}a_1^2 + I_{yy}a_2^2 + I_{zz}a_3^2 + 2I_{xy}a_1a_2 +$$

$$2I_{xz}a_1a_3 + 2I_{yz}a_2a_3 \quad , \qquad (C2)$$

which represents the moment of inertia about the local (geocentric) vertical in terms of the body system moments of inertia.

The vector [0, 1, 0] is the local vertical unit vector in the MSG S-system.

Referring to Appendix G we find that the local vertical unit vector in the MSG body system will be

$$\hat{g} = \begin{bmatrix} \sin\theta_{bs}\cos\psi_{bs} \\ -\sin\theta_{bs}\sin\phi_{bs}\sin\psi_{bs} + \cos\theta_{bs}\cos\phi_{bs} \\ -\cos\theta_{bs}\sin\phi_{bs} - \sin\theta_{bs}\sin\psi_{bs}\cos\phi_{bs} \end{bmatrix}$$

$$+ 2I_{12}(a_1(-\cos\theta \sin\phi \sin\Psi - \sin\theta \cos\phi) + a_2(\cos\theta \cos\Psi))$$

$$+ 2I_{13}(a_1(-\cos\theta \cos\phi \sin\Psi + \sin\theta \sin\phi) + a_3(\cos\theta \cos\Psi))$$

$$+ 2I_{23}(a_2(-\cos\theta \cos\phi \sin\Psi + \sin\theta \sin\phi) + a_3(-\cos\theta \sin\phi \sin\Psi - \sin\theta\cos\phi))$$

$$U_\Psi = 2I_{11}a_1(-\sin\theta \sin\Psi) + 2I_{22}a_2(-\sin\theta \sin\phi \cos\Psi)$$

$$+ 2I_{33}a_3(-\sin\theta \cos\phi \cos\Psi) + 2I_{12}(a_1(-\sin\theta \sin\phi \cos\Psi)$$

$$+ a_2(-\sin\theta \sin\Psi)) + 2I_{13}(a_1(-\sin\theta \cos\phi \cos\Psi)$$

$$+ a_3(-\sin\theta \sin\Psi) + 2I_{23}(a_2(-\sin\theta \cos\phi \cos\Psi) + a_3(-\sin\theta \sin\phi \cos\Psi))$$

$$U_\phi = 2I_{22}a_2(-\sin\theta \cos\phi \sin\Psi - \cos\theta \sin\phi)$$

$$+ 2I_{33}a_3(\sin\theta \sin\phi \sin\Psi - \cos\theta \cos\phi)$$

$$+ 2I_{12}a_1(-\sin\theta \cos\phi \sin\Psi - \cos\theta \sin\phi)$$

$$+ 2I_{13}a_1(\sin\theta \sin\phi \sin\Psi - \cos\theta \cos\phi)$$

$$+ 2I_{23}(a_2(\sin\theta \sin\phi \sin\Psi - \cos\theta \cos\phi)$$

$$+ a_3(-\sin\theta \cos\phi \sin\Psi - \cos\theta \sin\phi))$$

$$T_{b_1} = \frac{3\mu}{r^3}[-I_{22}a_2a_3 + I_{33}a_2a_3 - I_{12}a_1a_3 + I_{13}a_1a_2 + I_{23}(a_2^2 - a_3^2)]$$

$$T_{b_2} = \frac{3\mu}{r^3}[I_{11}a_1a_3 - I_{33}a_1a_3 + I_{12}a_2a_3 + I_{13}(a_3^2 - a_1^2) + I_{23}(-a_1a_2)]$$

$$T_{b_3} = \frac{3\mu}{r^3}[I_{11}(-a_1a_2) + I_{22}(a_1a_2) + I_{12}(a_1^2 - a_2^2) + I_{13}(-a_2a_3)$$

$$+ I_{23}(a_1a_3)]$$

This checks the torque equations. Remember, however, that (θ, Ψ, ϕ) appearing in I) and II) are $(\theta_{bs}, \Psi_{bs}, \phi_{bs})$, whereas the running angles of the simulation are $(\theta_{br}, \Psi_{br}, \phi_{br})$.

III. Numerical Checks.

Having established that the equations are correct, we must verify the programming. To this end a program was written on our terminal and the results compared with the simulator.

An arbitrary case was selected with

$$\phi_{br} = -.072248$$

$$\Psi_{br} = .17572$$

$$\theta_{br} = -.029535$$

$$\phi_{rs} = .064267$$

$$\theta_{rs} = .036348$$

This gave

$$\phi_{bs} = -.0353552$$

$$\Psi_{bs} = .17453$$

$$\theta_{bs} = .0283034$$

Using a principal axis body system with moments of inertia

$$I_{11} = 2686$$

$$I_{22} = 1846$$

$$I_{33} = 1617 \quad ,$$

both programs gave the same gravitational torques

$$T_1 = -1.10678\ \text{E-7}$$

$$T_2 = 1.44113\ \text{E-8}$$

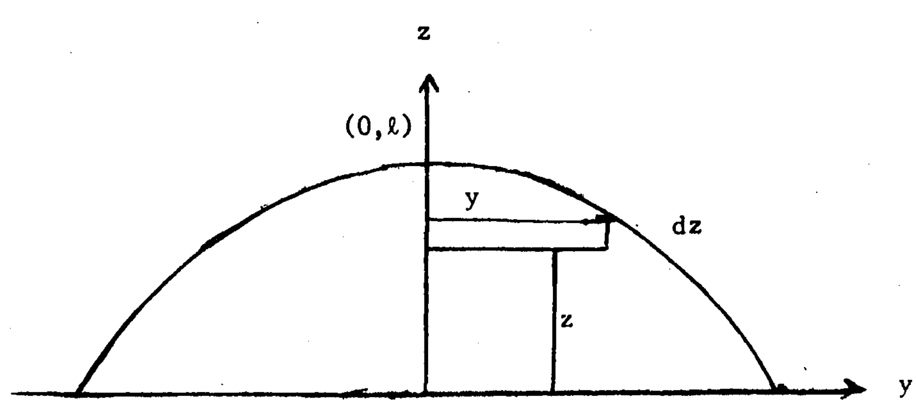
$$T_3 = -3.71911\ \text{E-7} \quad .$$

7.2.4 *Verification of Solar Pressure Perturbations*

Our object is to review the analysis appearing in Appendix B and to establish that the mechanization of that analysis is correct.

The physical assumptions are fairly standard. The behavior of the photons is considered to be that of small particles with momentum determined by the frequency. Thus momentum is transferred by absorption or by reflection from a smooth surface. The approximation of the satellite by a paraboloid is a necessary approximation to bring the geometry of the problem within reason.

Each equation in the absorptivity section was checked and changes made where necessary (mostly minor typographical changes). For a check on the magnitude and sign of the torque, the absorptivity torque was calculated for the radiation beam coming in at 90° ($\delta = 90°$) to the axis of symmetry of the paraboloid as follows. Since the force due to absorbed photons is the same for any two bodies casting the same shadow, we may calculate the torque using a parabola in the yz plane.



The equation of the parabola is $z = \ell - \gamma y^2$. The element of force is $dF = (2ydz)S$, where S is the solar constant. The element of torque about the y axis is $dT = zdF$ and is positive (since x, the incoming beam, and the force are all out of the page). Other torques about the origin are zero by symmetry. Thus:

$$T_a = 2S \int_o^{\ell} yzdz$$

$$= 2S \int_o^{\ell} \sqrt{\frac{\ell-z}{\gamma}}\, zdz$$

changing variable to $u = (\ell-z)/\gamma$, we integrate to

$$T_a = 2S\left[-\frac{2}{3}\gamma\ell\left(\frac{\ell-z}{\gamma}\right)^{3/2} + \frac{2}{5}\gamma^2\left(\frac{\ell-z}{\gamma}\right)^{5/2}\right]_o^{\ell}$$

or

$$T_a = S\sqrt{\frac{\ell}{\gamma}}\ \frac{8}{15}\ell^2 \quad \text{for} \quad \delta = 90° \quad .$$

Taking the limit as $\delta \to 90°$ for the last equation in T_a on p. 161 of the MSG report, we obtain the same result. For the actual dimensions of the satellite and $S = 9.4 \times 10^{-8}$ pounds/ft^2, this is approximately $T_a = 7.9 \times 10^{-6}$ foot-pounds.

The sign of any absorptivity torque is always positive or zero. If the parabola in the last figure is revolved about the z axis we have the given paraboloid. The x axis is out of the page and the $\vec{b}$ vector (unit vector from the sun to the satellite) makes an acute angle with the positive x axis and y is normal to $\vec{b}$ (definition of x and y axes). Since the force is in the direction of $\vec{b}$, the torque about the y axis is always positive or zero. The torque is zero when either the upper or lower surface of the paraboloid is fully illuminated by symmetry of the shadow on the $z = 0$ (or $z = \ell$) plane. The torque becomes zero when the sun's rays are just tangent to the lower lip of the paraboloid. Referring to the preceding figure, we may evaluate $\frac{dz}{dy}$ at the point $(y,z) = (-\sqrt{\ell/\gamma}, 0)$. At that point $\left.\frac{dz}{dy}\right|_{z=0} = 2\sqrt{\ell\gamma}$. The angle δcrit is between the z axis and the incoming beam so that

$$\tan \delta\text{crit} = \frac{1}{2\sqrt{\ell\gamma}} \quad .$$

Another critical angle is obtained from the point $(+\sqrt{\ell/\gamma}, 0)$ equal to $\pi - \delta$crit. Thus T_a is non-zero only for

$$\delta\text{crit} < \delta < \pi - \delta\text{crit} \quad .$$

To check the fact that non-zero T_a are positive, note in Appendix B, sect. 3 that all terms in T_a are positive with the possible exception of $\tan^2 \delta\ \ell^2 8/15 - \ell/\gamma\ 4/15$. Using the smallest value for $\tan^2 \delta = \tan^2 \delta\text{crit}$ and the actual dimensions of the satellite, we see that these terms are also positive.

Each equation in the reflectivity section was checked. There are four cases occuring in the reflectivity situation depending upon the relation of δ to δ crit. These cases may be summarized.

Case	δ range	Illuminated from	Illum. of Upper Surface	Illum. of Lower Surface
I	$(0, \delta\text{crit})$	Below	None	Complete
II	$(\delta\text{crit}, \frac{\pi}{2})$	Below	Partial	Partial
III	$(\frac{\pi}{2}, \pi - \delta\text{crit})$	Above	Partial	None
IV	$(\pi - \delta\text{crit}, \pi)$	Above	Complete	None

To determine the magnitude of this torque, we will determine the maximum value and the signs for case I. For this case, we integrate the equation in Appendix B, section 5 over even limits $x_o = -\sqrt{\ell/\gamma}$, $x_1 = +\sqrt{\ell/\gamma}$. Since the limits are even, on expanding $(2\gamma x \sin\delta + \cos\delta)^2$

we see that only the cross term $4\gamma x \sin\delta \cos\delta$ makes any contribution. Differentiating

$$T_r = (-4S)(2\gamma \sin 2\delta) \int_{-\sqrt{\ell/\gamma}}^{+\sqrt{\ell/\gamma}} \int_{o}^{\sqrt{\ell/\gamma - x^2}} \frac{x[2\gamma^2 x^3 - x(2\gamma\ell - 1) + 2\gamma^2 xy^2]}{1+4\gamma^2(x^2+y^2)} dxdy$$

where we have substituted $2\sin\delta \cos\delta = \sin 2\delta$ and $\mathrm{sign}(\cos\theta) = -1$ (for illumination from below, case I).

The integral in the last equation is over half of the base circle. We may transform to polar coordinates $r^2 = x^2 + y^2$, $x = r\cos\theta$ to obtain:

$$T_r = (-4S)(2\gamma \sin 2\delta) \int_{o}^{\sqrt{\ell/\gamma}} \int_{o}^{\pi} \frac{(s^2 \cos^2\theta)[2\gamma^2 r^2 - (2\gamma\ell - 1)]}{1+4\gamma^2 r^2} r dr d\theta$$

Since $1-2\gamma\ell > 0$ for this paraboloid, the sign of T_r is always negative (for case I). The maximum value of T_r with respect to δ is at $\delta = 45°$ (allowed for case I since $45° \leq \delta crit \simeq 66.6°$). Thus for case I, T_r is a sine function with amplitude determined by the last integral above.

This last integral may be integrated explicitly. Integrating $\int_o^\pi \cos^2\theta d\theta$ we obtain $\frac{\pi}{2}$. Changing variable to $u = 1+4\gamma^2 r^2$, we have

$$T_r = -4S\pi\gamma \sin 2\delta \int_1^{1+4\gamma\ell} [2\gamma^2 \left(\frac{u-1}{4\gamma^2} \right)^2 + (1-2\gamma\ell)\left(\frac{u-1}{4\gamma^2} \right)] \frac{du/8\gamma^2}{u}$$

or

$$T_r = -\frac{1}{2} S\pi \frac{1}{\gamma} \sin 2\delta \int_1^{1+4\gamma\ell} [\frac{1}{8\gamma^2}\left(\frac{1-2u+u^2}{u} \right) + \left(\frac{1-2\gamma\ell}{4\gamma^2} \right)\left(-\frac{1}{u} + 1\right)]du$$

$$= -\frac{1}{2} S\pi \frac{1}{\gamma} \sin 2\delta \left[\frac{1}{8\gamma^2}\left(\ell n u - 2u + \frac{u^2}{2}\right) + \left(\frac{1-2\gamma\ell}{4\gamma^2} \right)(-\ell n u + u) \right]_1^{1+4\gamma\ell}$$

$$T_r = -\frac{1}{2} S\pi \frac{1}{\gamma} \sin 2\delta \left[\frac{1}{8\gamma^2} \left(\ell n(1+4\gamma\ell) - 2(4\gamma\ell) + \frac{1}{2}(1+4\gamma\ell)^2 - \frac{1}{2} \right) \right.$$

$$\left. + \left(\frac{1-2\gamma\ell}{4\gamma^2} \right)(-\ell n(1+4\gamma\ell) + 4\gamma\ell) \right]$$

Thus $|T_r|\max = \frac{S\pi}{2\gamma} \left[\frac{\ell n(1+4\gamma\ell) - 4\gamma\ell + 8\gamma^2\ell^2}{8\gamma^2} + \left(\frac{1-2\gamma\ell}{4\gamma^2} \right)(-\ell n(1+4\gamma\ell) + 4\gamma\ell) \right]$

or

$$|T_r|\max = \frac{S\pi}{16\gamma^3} \left[(-1+4\gamma\ell)\ell n(1+4\gamma\ell) + 4\gamma\ell - 8\gamma^2\ell^2 \right] \quad .$$

For this paraboloid:

$$|T_r|\max = 1.83 \times 10^{-4} \text{ ft-pounds.}$$

Since $\gamma\ell \simeq 4.696 \times 10^{-2}$ is small compared to one, we can get an interesting approximation to $|T_r|\max$. Expanding the ℓn term and retaining $\gamma^2\ell^2$ powers, we obtain

$$|T_r|\max \simeq \frac{S\pi\ell^2}{\gamma} = S\pi(\sqrt{\ell/\gamma})^2\ell \quad .$$

Now $\sqrt{\ell/\gamma}$ is the radius of the base circle for the paraboloid and ℓ is the depth. Thus

$$|T_r|\max \simeq SV_c$$

where V_c is the volume of the right circular cylinder which circumscribes the paraboloid and has the same axis of symmetry. This approximation gives

$$|T_r|\max \simeq 2.16 \times 10^{-4} \text{ ft-pounds.}$$

The value of T_r integral for case IV is the negative of that for case I. Since $\sin 2\delta$ is also negative for case IV, we have that T_r for case IV is always negative and that the same value of $|T_r|$max is valid. Since $|T_r|$ for case I is decreasing as we approach the case II region (and similarly for IV and III), the above value of $|T_r|$max appears to be the maximum for all δ.

Numerical checks.

In order to check the numerical results for the solar torques computed by the simulator program, several things were done. First, the absorptivity torque (T_a) and the reflectivity torque (T_r) were calculated separately; second, certain formulas (described below) were re-programmed for the remote console at MSG and the results compared with those of the simulator program; third, the checks previously discussed in this write-up were applied to the numerical results.

The equation used on the remote console for T_a is:

$$T_a = S\cos^2\delta \sqrt{\frac{\ell}{\gamma} - \frac{1}{4\gamma^2\tan^2\delta}} \times$$

$$\left[\frac{\ell^2 8}{15} \tan^2\delta - \frac{\ell}{\gamma}\frac{4}{15} + \frac{1}{30\gamma^2}\frac{1}{\tan^2\delta} \right]$$

The four cases occurring for reflectivity are treated according to Table 7.2.4.1 where the functions therein are defined as follows:

$$w(x_o,x_1,y_o,y_1,\delta) \equiv \int_{x_o}^{x_1} (2\gamma x\sin\delta+\cos\delta)^2[h(x,y_1) - h(x,y_o)]dx$$

where x_o, x_1, δ are constants, y_o, y_1 are functions and

$$h(x,y(x)) \equiv \frac{2\gamma^2 x^3 - x(2\gamma\ell - 1)}{4\gamma^2} \left[\frac{1}{\sqrt{\frac{1+4\gamma^2x^2}{4\gamma^2}}} \tan^{-1} \frac{y}{\sqrt{\frac{1+4\gamma^2x^2}{4\gamma^2}}} \right]$$

$$+ \frac{x}{2} \left[y - \sqrt{\frac{1+4\gamma^2x^2}{4\gamma^2}} \tan^{-1} \frac{y}{\sqrt{\frac{1+4\gamma^2x^2}{4\gamma^2}}} \right]$$

The angle α between the plane of the satellite's orbit and the plane of the earth's orbit was arbitrarily set at 20.5° and a selected set of data appears in Table 7.2.4.2. The angle θ_{se} in the table is equal to the angle of the satellite as it travels around the earth in its orbit ($\theta_{se} = 0°$ being the point closest to the sun). The relation between θ_{se} and δ for this case is $\cos\delta = -\cos 20.5° \cos \theta_{se}$.

We see from Table 7.2.4.2 that there is basic agreement between the two sets of results. The maximum percentage error between the two runs is about 3%. The differences are probably due to the different machines and different integration subroutines used.

For other checks mentioned in this write-up, note that the absorptivity torque T_a at $\delta = 90°$ ($\theta_{se} = 90°$) is $T_a \simeq 7.9 \times 10^{-6}$ ft-pounds. We see that the non-zero T_a is always positive and that T_a is zero outside of the range

$$\delta\text{crit} = 66.6 \leq \delta \leq 113.4°$$

$$(64.9° \leq \theta_{se} \leq 115.1°) \quad .$$

The reflectivity torque maximum approximately checks $|T_r|\max = 1.83 \times 10^{-4}$ and occurs near $\delta = 45°$ ($\theta_{se} = 41.4°$). The T_r torques are always negative for cases I ($0 < \delta < 66.6°$) and IV ($113.4 < \delta < 180°$) i.e., I ($115.1 < \theta_{se} < 180°$) and IV ($0 < \theta_{se} < 64.9°$).

Figure 7.2.4.1 is a plot of total torque $T = 0.75\ T_r + 0.25\ T_a$ versus $\theta = 180° - \delta$. In the ranges $0 < \theta < 66.6$ and $115.1 < \theta < 180°$, T_a is zero and $T = 0.75\ T_r$. Thus we see that the T_r curve in these ranges is indeed a constant times the sine function $\sin 2\delta$.

It appears that a very good way to enter the solar torques in the future would be by means of an approximating curve.

From reviewing the tabulated data and from applying the other checks mentioned in the write-up, it appears that the equations in the MSG report are correct and have been correctly programmed.

	I $\delta\varepsilon(0,66.575°]$	II $\delta\varepsilon(66.575°, 90°]$	III $\delta\varepsilon(90°,113.425°]$	IV $\delta\varepsilon(113.425°,180°]$
T_r	$-4Sw(x_o,x_1,y_o,y_1,\delta)$	$\begin{bmatrix} +4Sw(x_o,x_1,y_o,y_1,\delta) \\ -4Sw(x_o',x_1',y_o',y_1',\delta) \end{bmatrix}$	$+4Sw(x_o,x_1,y_o,y_1,\delta)$	$+4Sw(x_o,x_1,y_o,y_1,\delta)$
y_o	$\equiv 0$	$\equiv 0$	$\equiv 0$	$\equiv 0$
y_1	$\sqrt{\frac{\ell}{\gamma} - x^2}$	$\sqrt{\frac{\ell}{\gamma} - x^2}$	$\sqrt{\frac{\ell}{\gamma} - x^2}$	$\sqrt{\frac{\ell}{\gamma} - x^2}$
x_o	$-\sqrt{\frac{\ell}{\gamma}}$	$-\sqrt{\frac{\ell}{\gamma}}$	$-\sqrt{\frac{\ell}{\gamma}}$	$-\sqrt{\frac{\ell}{\gamma}}$
x_1	$+\sqrt{\frac{\ell}{\gamma}}$	$-\frac{1}{2\gamma\tan\delta}$	$-\frac{1}{2\gamma\tan\delta}$	$+\sqrt{\frac{\ell}{\gamma}}$
x_o'	-	$-\frac{1}{2\gamma\tan\delta}$	-	-
x_1'	-	$+\sqrt{\frac{\ell}{\gamma}}$	-	-
y_o'	-	$\sqrt{\frac{\ell}{\gamma} - (x + \frac{1}{\gamma\tan\delta})^2}$ *	-	-
y_1'	-	$\sqrt{\frac{\ell}{\gamma} - x^2}$	-	-

Table 7.2.4.1

*set to zero if quantity under square root is negative

	T_a (ft.-lb.)		T_r (ft.-lb.)	
θ_{se}(deg)	Remote Console	Simulator Program	Remote Console	Simulator Program
0		0	-1.22×10^{-4}	-1.20×10^{-4}
10		0	-1.33×10^{-4}	-1.30×10^{-4}
25		0		-1.64×10^{-4}
30		0	-1.77×10^{-4}	-1.73×10^{-4}
40		0		-1.82×10^{-4}
45		0		-1.81×10^{-4}
50		0		-1.76×10^{-4}
70	6.60×10^{-7}	6.79×10^{-7}	-1.13×10^{-4}	-1.11×10^{-4}
80	5.18×10^{-6}	5.23×10^{-6}	-6.58×10^{-5}	-6.50×10^{-5}
85	7.16×10^{-6}	7.20×10^{-6}	-4.47×10^{-5}	-4.42×10^{-5}
90	7.90×10^{-6}	7.94×10^{-6}	-2.73×10^{-5}	-2.70×10^{-5}
100	5.18×10^{-6}	5.23×10^{-6}	-7.15×10^{-5}	-7.07×10^{-5}
110	6.60×10^{-7}	6.79×10^{-7}	-1.13×10^{-4}	-1.12×10^{-4}
130		0		-1.76×10^{-4}
135		0		-1.81×10^{-4}
140		0		-1.82×10^{-4}
150		0	-1.77×10^{-4}	-1.73×10^{-4}
170		0	-1.33×10^{-4}	-1.30×10^{-4}
175		0		-1.22×10^{-4}
180		0	-1.22×10^{-4}	-1.20×10^{-4}

Table 7.2.4.2

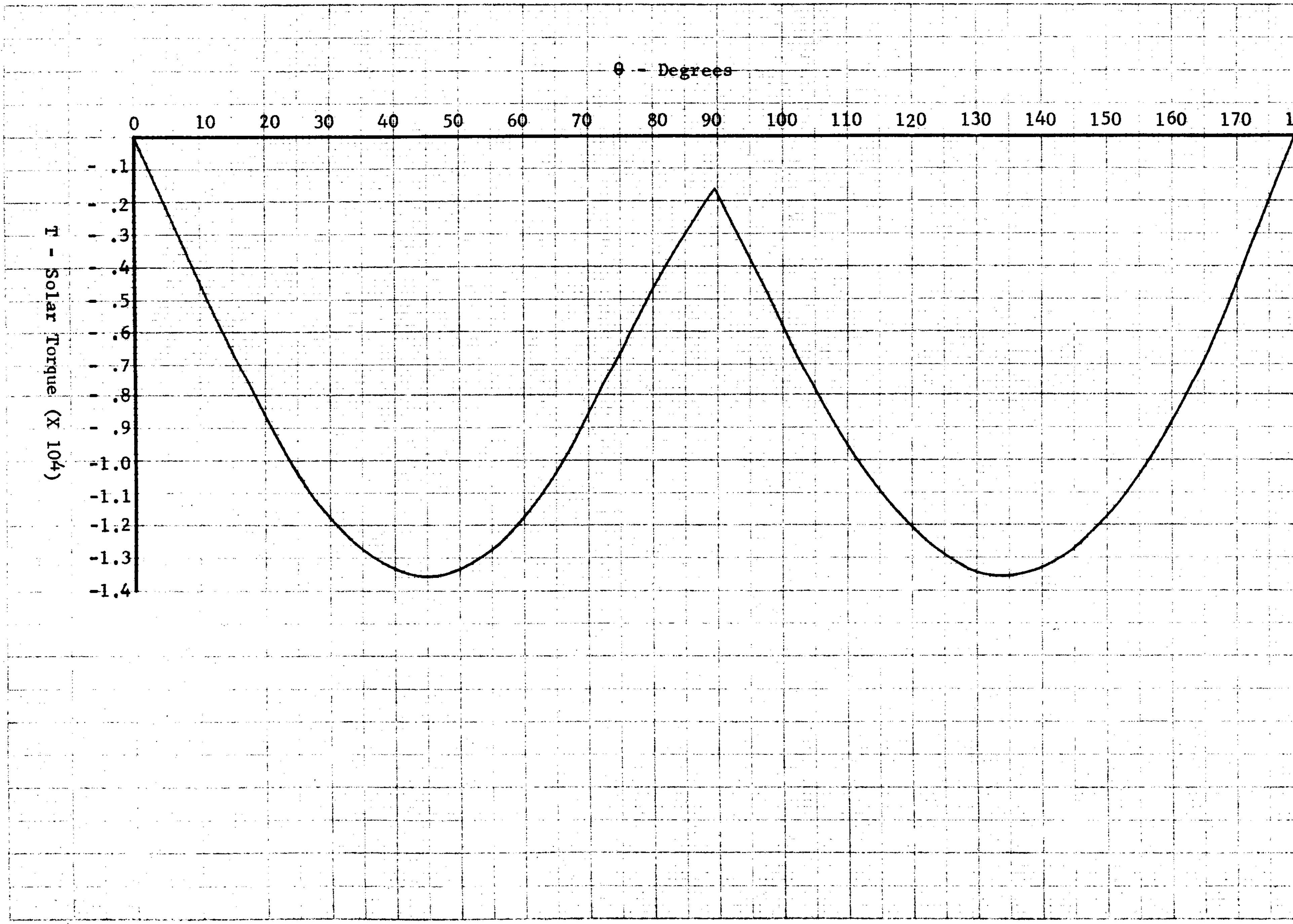
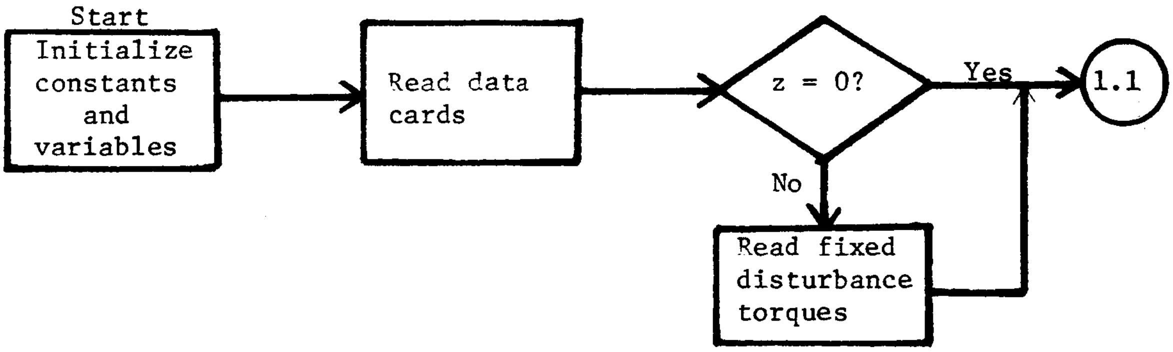

θ - Degrees
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
T - Solar Torque (X 10⁴)
- .1
- .2
- .3
- .4
- .5
- .6
- .7
- .8
- .9
-1.0
-1.1
-1.2
-1.3
-1.4

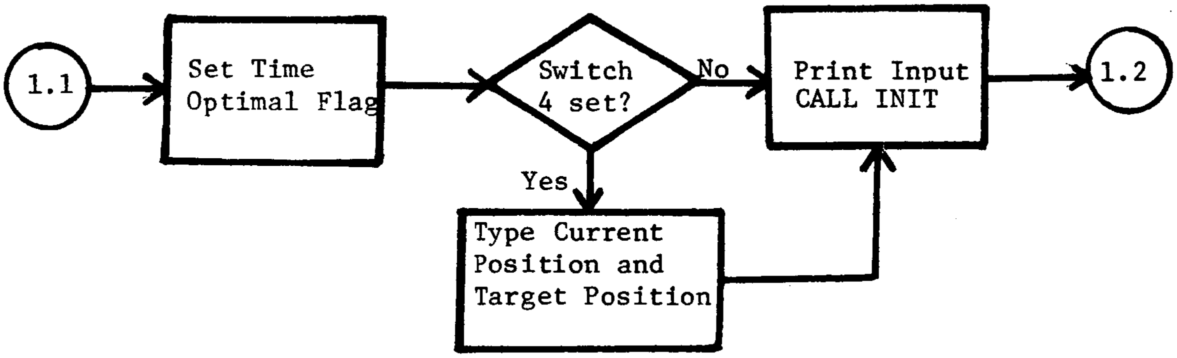
Figure 7.2.4.1

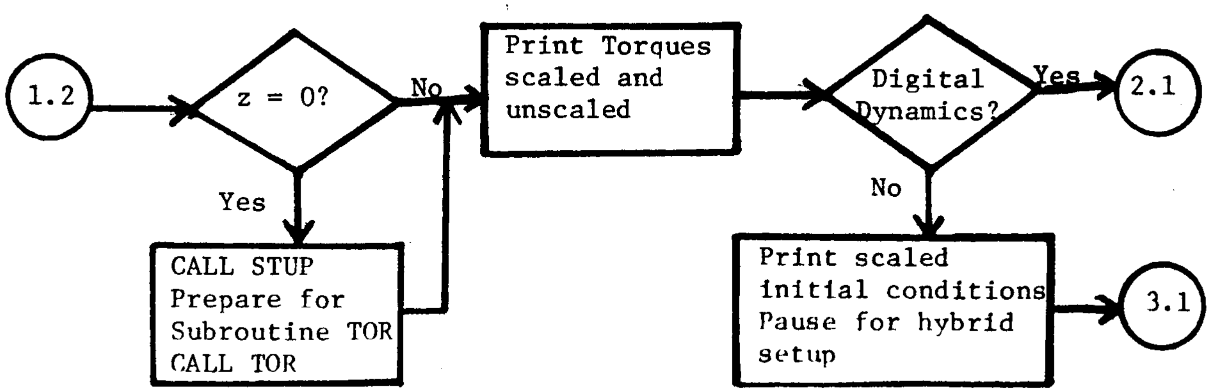
7.3 *Flow Charts*

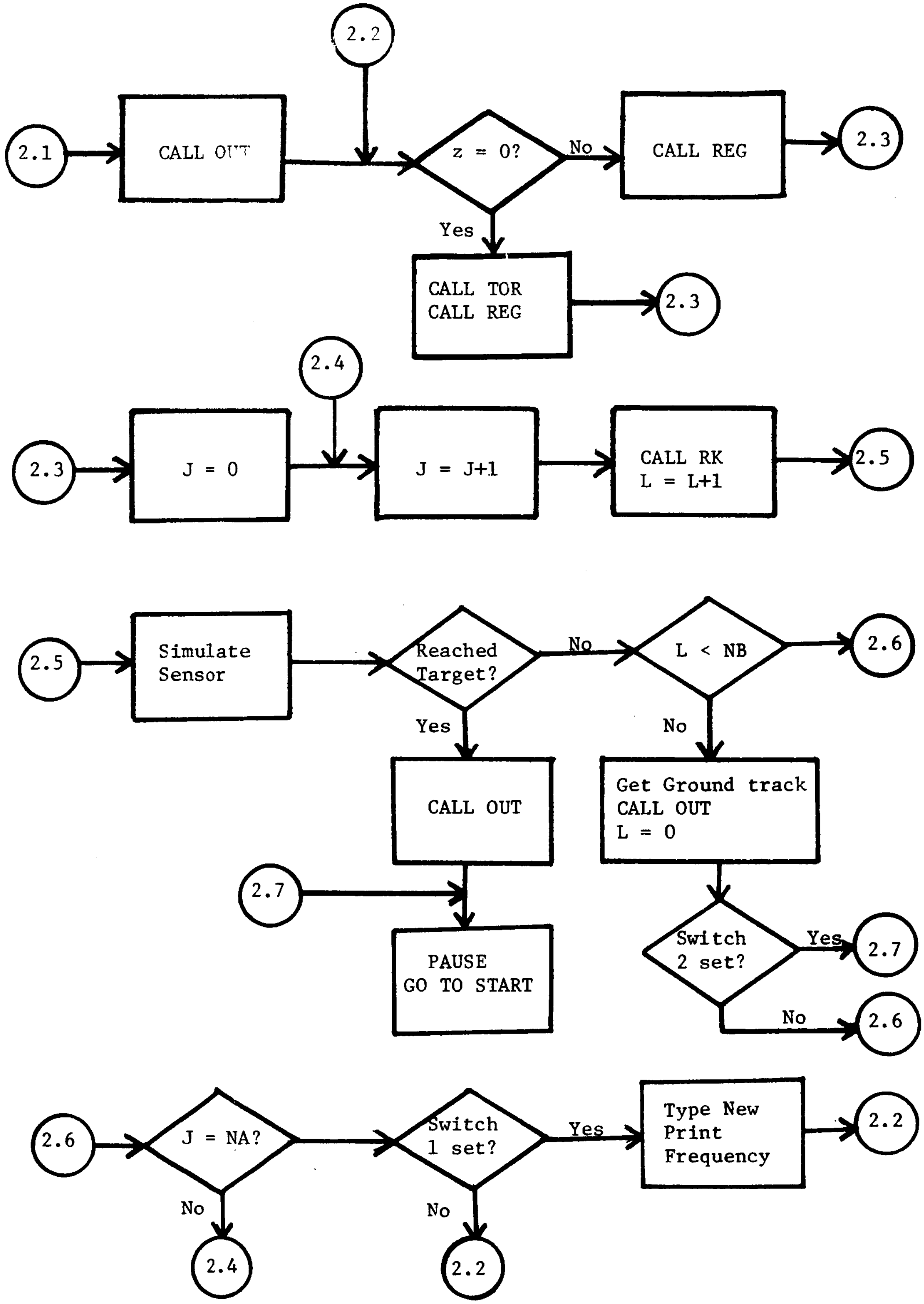
The flow charts which follow are for the main program and subroutine "REG" only. All other subroutines are straight-forward sequential computation and warrant no flow chart. The charts exhibited are those of the sensor model. The logic for the ideal model is identical, except that any reference to sensor evaluation should be omitted.

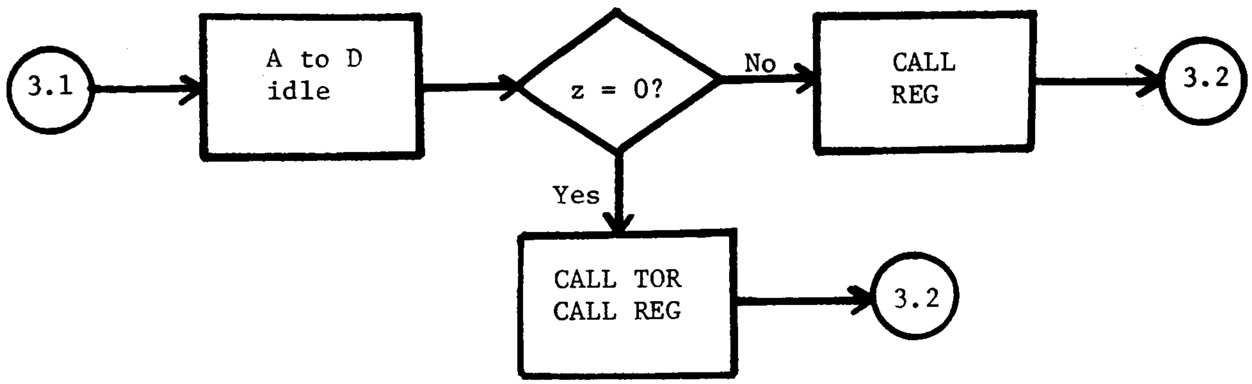
Main Program

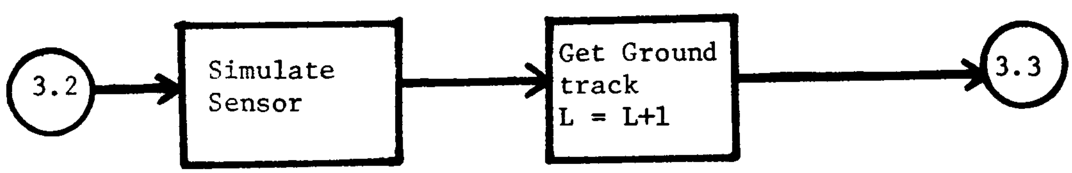


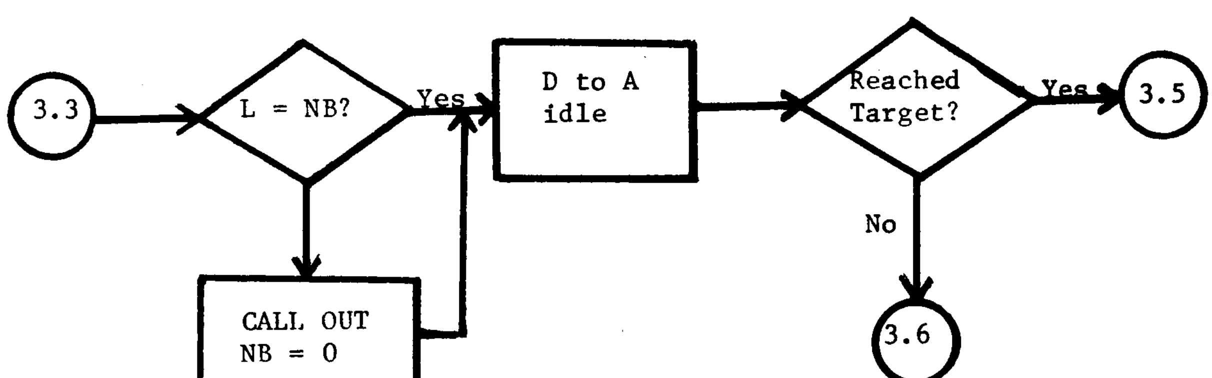


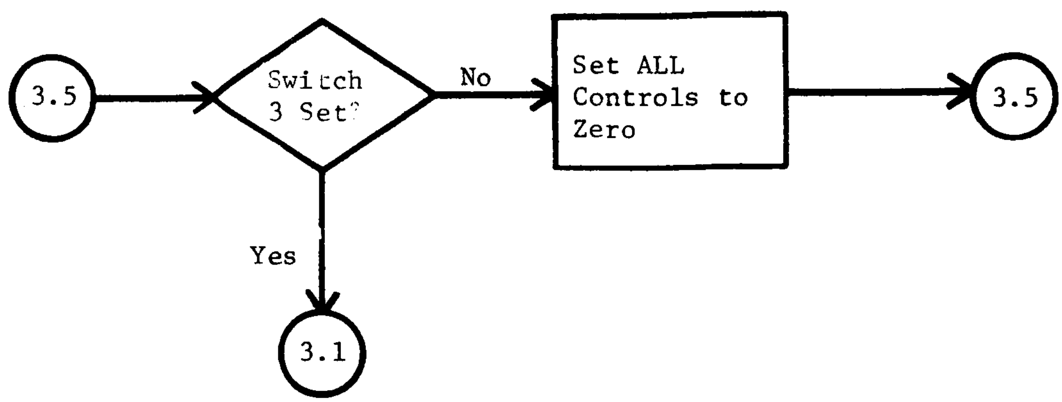


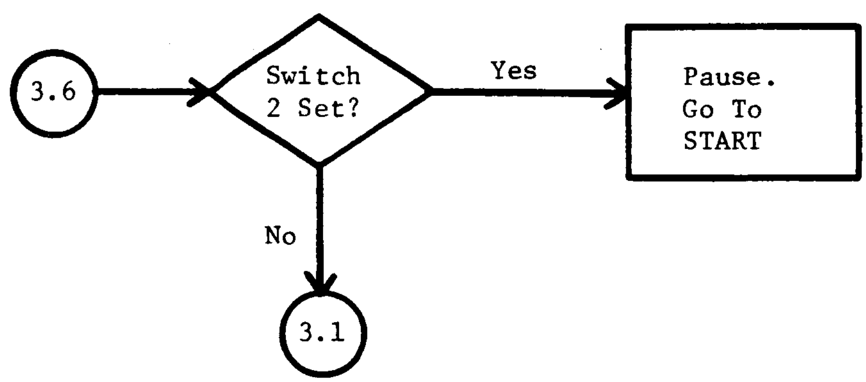

2.2
2.1
CALL OUT
z = 0?
No
CALL REG
2.3
Yes
CALL TOR
CALL REG
2.3
2.4
2.3
J = 0
J = J+1
CALL RK
L = L+1
2.5
2.5
Simulate
Sensor
Reached
Target?
No
L < NB
2.6
Yes
No
CALL OUT
Get Ground track
CALL OUT
L = 0
2.7
PAUSE
GO TO START
Switch
2 set?
Yes
2.7
No
2.6
2.6
J = NA?
Switch
1 set?
Yes
Type New
Print
Frequency
2.2
No
2.4
No
2.2


3.1
A to D
idle
z = 0?
No
CALL
REG
3.2
Yes
CALL TOR
CALL REG
3.2

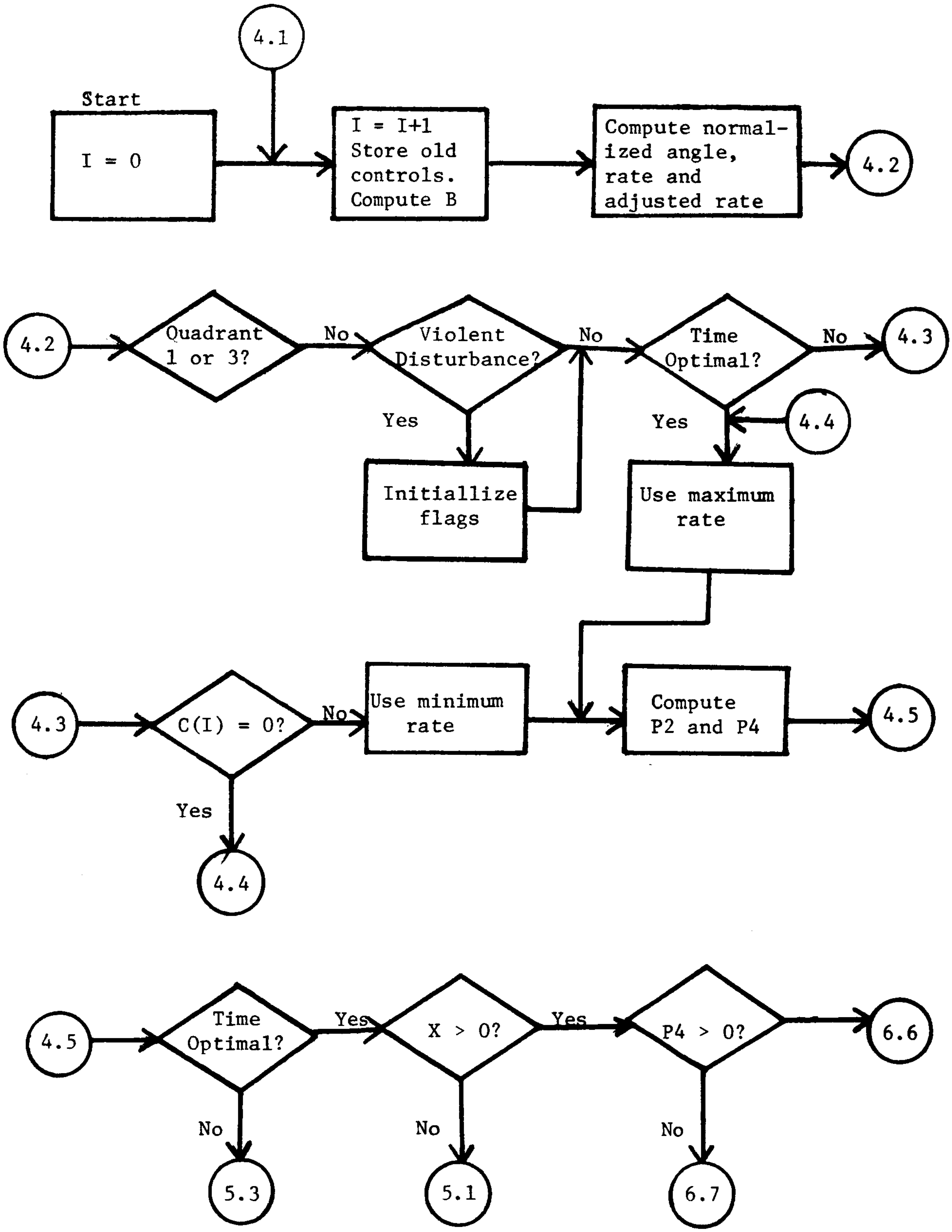

3.2
Simulate
Sensor
Get Ground
track
L = L+1
3.3


3.3
L = NB?
Yes
D to A
idle
Reached
Target?
Yes
3.5
CALL OUT
NB = 0
No
3.6

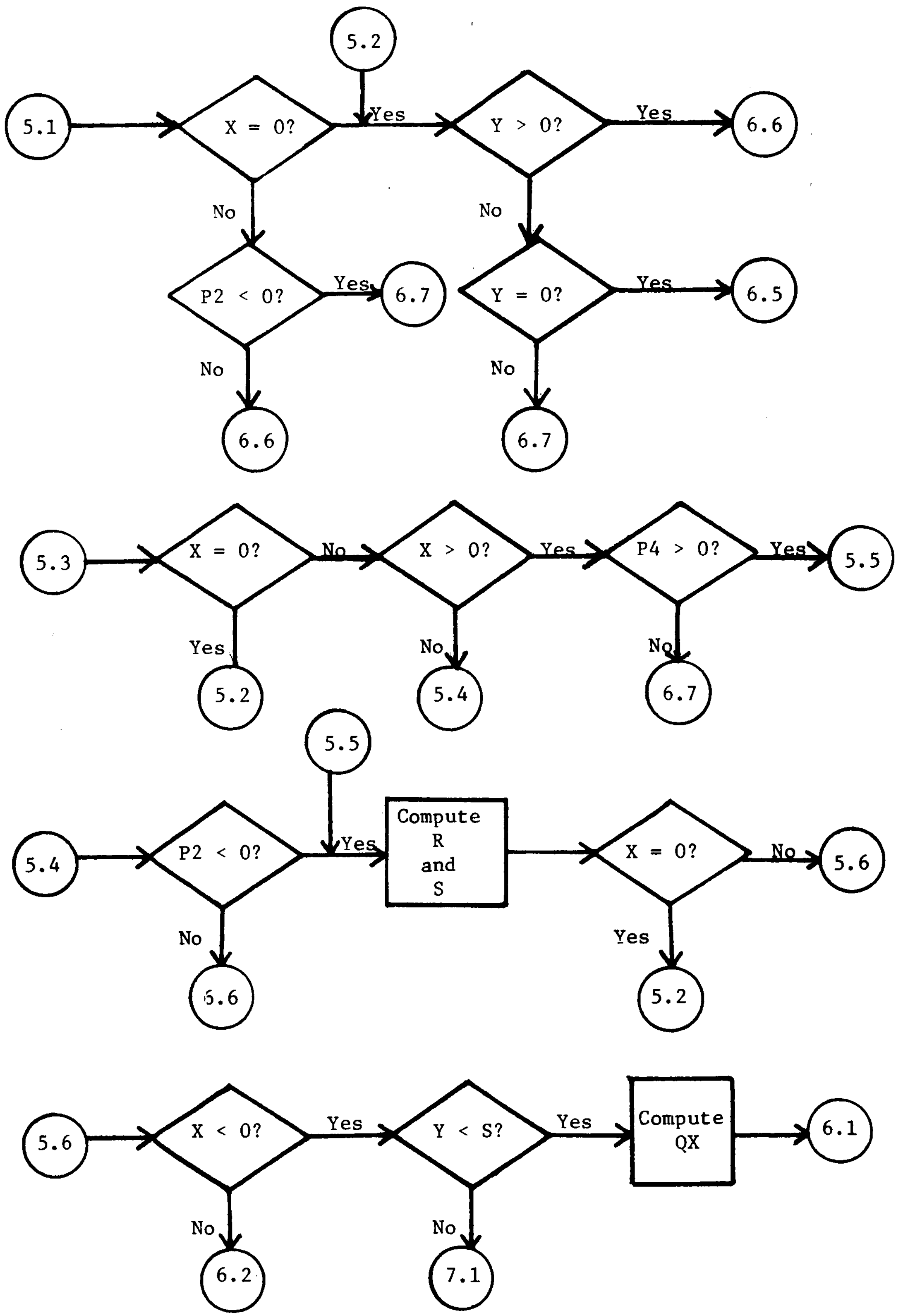

3.5
Switch 3 Set?
No
Set ALL Controls to Zero
3.5
Yes
3.1


3.6
Switch 2 Set?
Yes
Pause. Go To START
No
3.1

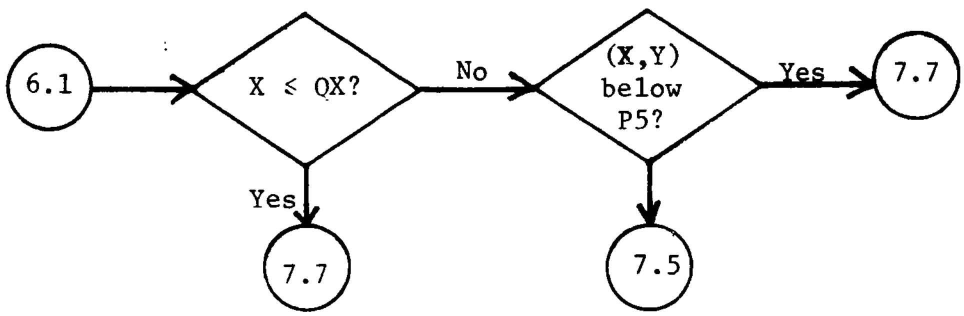
REG

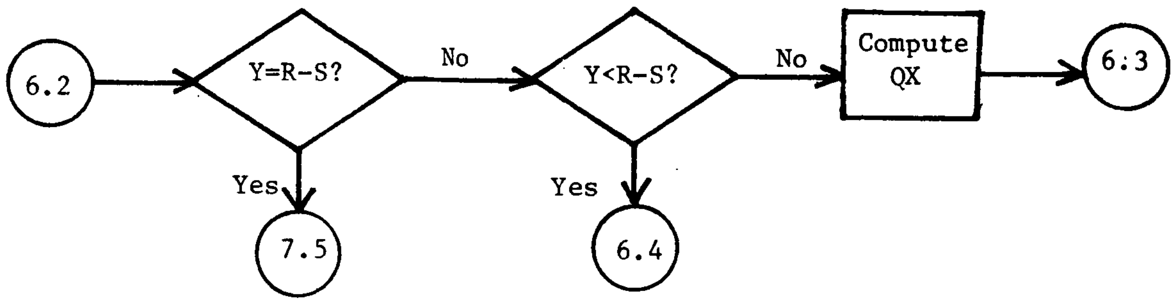

4.1
Start
I = 0
I = I+1
Store old
controls.
Compute B
Compute normal-
ized angle,
rate and
adjusted rate
4.2
4.2
Quadrant
1 or 3?
No
Violent
Disturbance?
No
Time
Optimal?
No
4.3
Yes
Initiallize
flags
Yes
4.4
Use maximum
rate
4.3
C(I) = 0?
No
Use minimum
rate
Compute
P2 and P4
4.5
Yes
4.4
4.5
Time
Optimal?
Yes
X > 0?
Yes
P4 > 0?
6.6
No
5.3
No
5.1
No
6.7

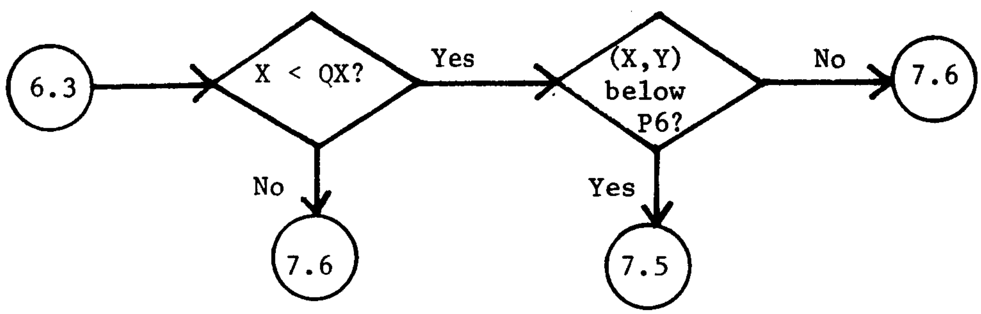
REG

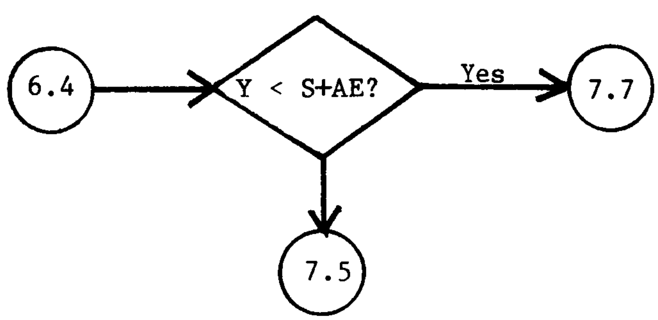

5.2
5.1
X = 0?
Yes
Y > 0?
Yes
6.6
No
No
P2 < 0?
Yes
6.7
Y = 0?
Yes
6.5
No
No
6.6
6.7
5.3
X = 0?
No
X > 0?
Yes
P4 > 0?
Yes
5.5
Yes
No
No
5.2
5.4
6.7
5.5
5.4
P2 < 0?
Yes
Compute R and S
X = 0?
No
5.6
No
Yes
6.6
5.2
5.6
X < 0?
Yes
Y < S?
Yes
Compute QX
6.1
No
No
6.2
7.1

REG

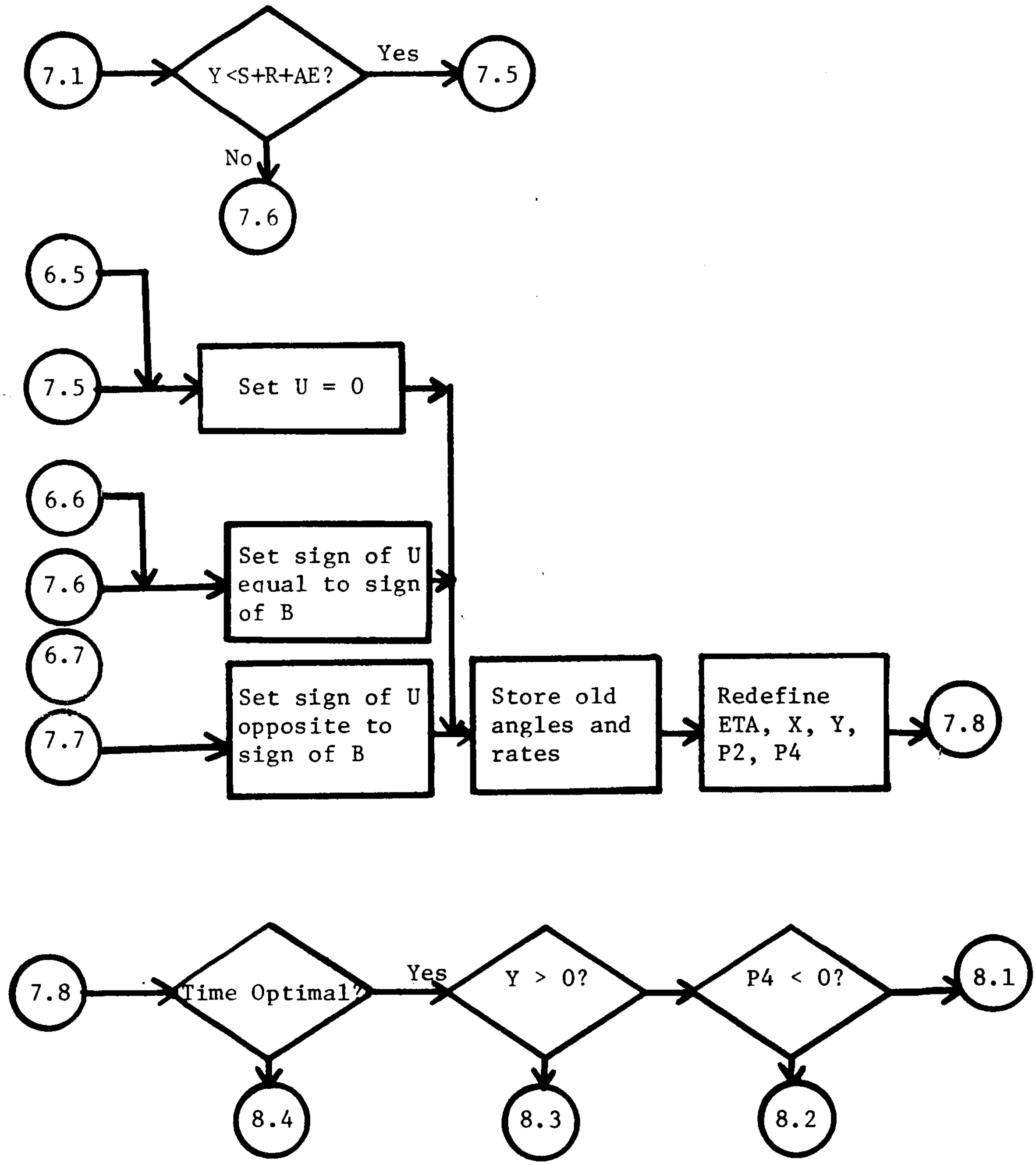

6.1
X ≤ QX?
No
(X,Y)
below
P5?
Yes
7.7
Yes
7.7
7.5


6.2
Y=R-S?
No
Y<R-S?
No
Compute
QX
6.3
Yes
7.5
Yes
6.4


6.3
X < QX?
Yes
(X,Y)
below
P6?
No
7.6
No
7.6
Yes
7.5

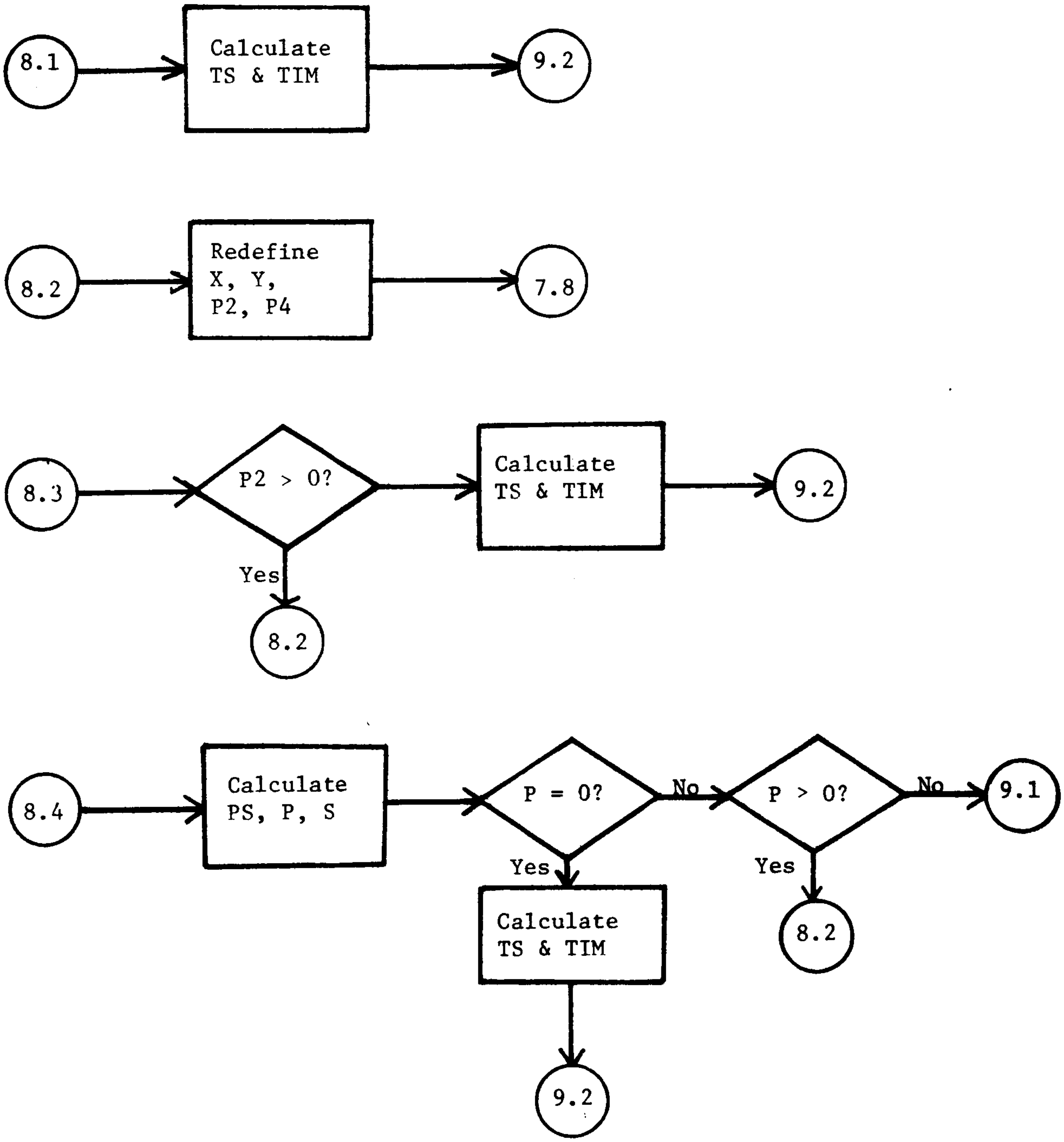

6.4
Y < S+AE?
Yes
7.7
7.5

REG

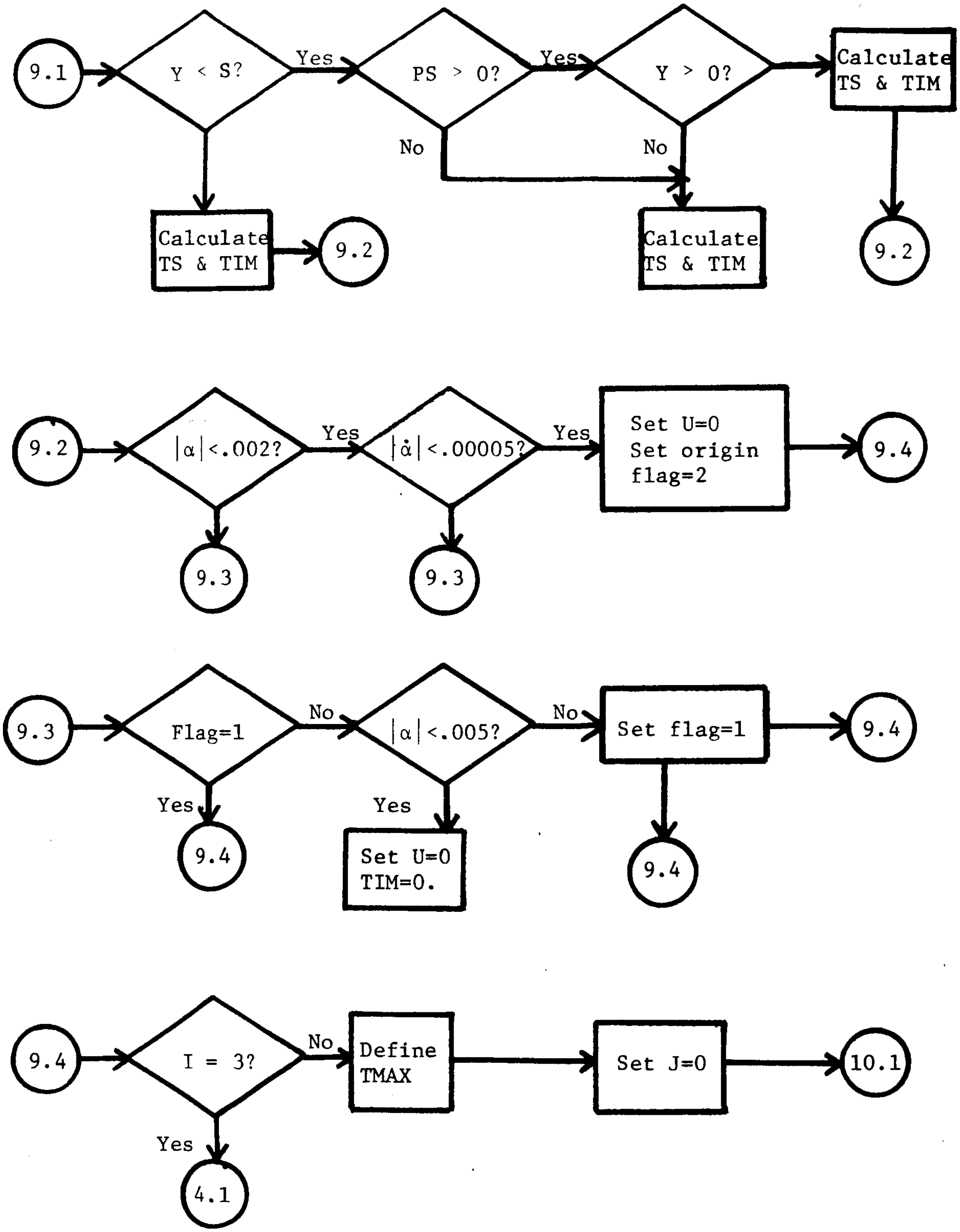


7.1
Y <S+R+AE?
Yes
7.5
No
7.6
6.5
7.5
Set U = 0
6.6
7.6
Set sign of U equal to sign of B
6.7
7.7
Set sign of U opposite to sign of B
Store old angles and rates
Redefine ETA, X, Y, P2, P4
7.8
7.8
Time Optimal?
Yes
Y > 0?
P4 < 0?
8.1
8.4
8.3
8.2

REG

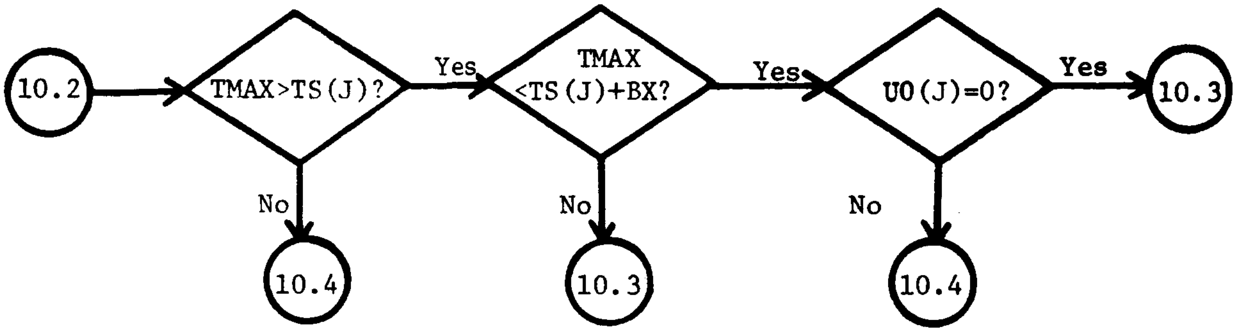

8.1
Calculate TS & TIM
9.2
8.2
Redefine X, Y, P2, P4
7.8
8.3
P2 > 0?
Yes
8.2
Calculate TS & TIM
9.2
8.4
Calculate PS, P, S
P = 0?
No
P > 0?
No
9.1
Yes
Calculate TS & TIM
9.2
Yes
8.2

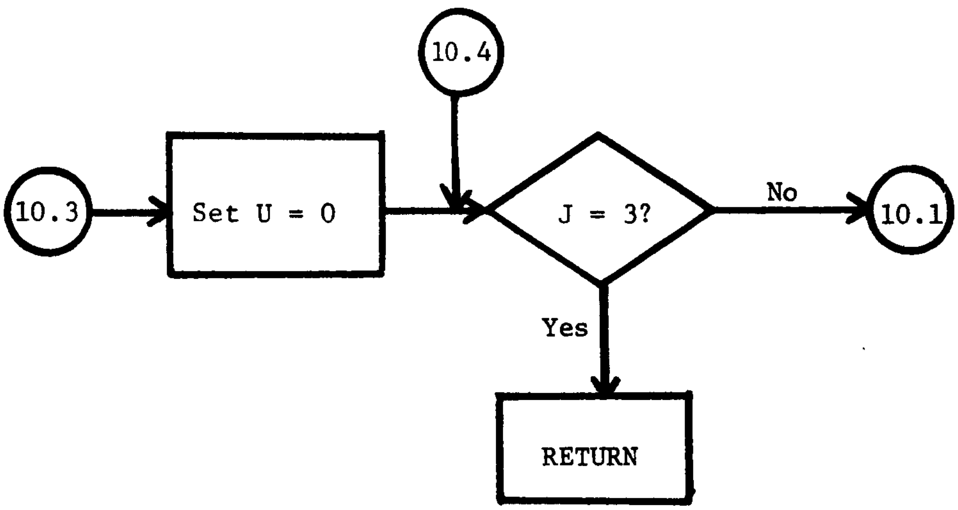
REG



$|\alpha|$ = angle

$|\dot{\alpha}|$ = rate


10.2
TMAX>TS(J)?
Yes
No
10.4
TMAX
<TS(J)+BX?
Yes
No
10.3
UO(J)=0?
Yes
No
10.4
10.3


10.4
10.3
Set U = 0
J = 3?
No
10.1
Yes
RETURN

7.4 *Program Listings*

Three distinct computer programs appear on the following pages. The first is the ideal model program. The second program includes several real effects, i.e. those due to the sensors, solar torques and gravity gradient. Subroutine "TOR" in this listing is the subroutine which evaluates the later two quantities, whereas in the ideal program "TOR" is a dummy subroutine. The third program listing is one for simulating elastic effects on the satellite. Because of storage limitations this is simply a digital program, no hybrid operation being possible, nor does it take into consideration any other real effects.

```
C                                 IDEAL MODEL
C                                                                 DATE 3 MAR 68
C
C
C
      DIMENSION ELV(3,2),AZ(3,2)
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      DIMENSION RA(3,3),RB(3,3),RC(3,3),RD(3,3),ETA(3),C(3),NF(3)
      DIMENSION AP(3),XX(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX
      COMMON /MISAL/ TH(3,3,2),UX(3)
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,RBS,PIB,YB,ROB
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,TO
   13 TI=0.; L=0; T4=6.610808; NC=2; RAD=.017453      ;AC=1.
      PRINT 113
      DO 1 I=1,3
      NF(I)=1.; U(I)=ZM(I); C(I)=1.
    1 CONTINUE
      READ 100,NA,NB,PA,PB,PC,Z
      READ 101,(ZI(I),I=1,3),(ZM(I),I=1,3),(AL(I),I=1,3)
      READ 101,((ZL(I,J),J=1,3),I=1,3)
      READ 101,AX,BX,XF,PIB,ROB,YB,RDBS,YDBS,PDBS
      READ 101,YO,AT,ONG,XLAT,XLON,DT,AE
      IF(Z.NE.0.)      READ 101,T(1),T(2),T(3)
C     ELV(I,J)=ELEVATION IN DEGREES FOR AXIS I,J=1 PLUS,2 MINUS
C     AZ(I,J)=AZIMUTH
      READ 101,(ELV(I,1),I=1,3),(AZ(I,1),I=1,3)
      READ 101,(ELV(I,2),I=1,3),(AZ(I,2),I=1,3)
      DO 40 J=1,2
      DO 40 I=1,3
      ELV(I,J)=ELV(I,J)*RAD
   40 AZ(I,J)=AZ(I,J)*RAD
      DO 42 K=1,2
      DO 41 I=1,3
   41 TH(I,I,K)=COS(ELV(I,K))*COS(AZ(I,K))
      TH(2,1,K)=SIN(ELV(1,K))
      TH(3,1,K)=COS(ELV(1,K))*SIN(AZ(1,K))
      TH(1,2,K)=COS(ELV(2,K))*SIN(AZ(2,K))
      TH(3,2,K)=SIN(ELV(2,K))
      TH(1,3,K)=SIN(ELV(3,K))
   42 TH(2,3,K)=COS(ELV(3,K))*SIN(AZ(3,K))
      PRINT 120
      PRINT 101,(ELV(I,1)/RAD,I=1,3),(AZ(I,1)/RAD,I=1,3)
      PRINT 101,(ELV(I,2)/RAD,I=1,3),(AZ(I,2)/RAD,I=1,3)
      PRINT 117
      PRINT 118,((TH(I,J,1),J=1,3),I=1,3)
      PRINT 119
      PRINT 118,((TH(I,J,2),J=1,3),I=1,3)
      PRINT 113
  117 FORMAT (1H1,5X, $MULTIPLIERS FOR POSITIVE THRUSTS,///)
  118 FORMAT (3F15.8)
  119 FORMAT (1H0,5X, $MULTIPLIERS FOR NEGATIVE THRUSTS,///)
```

```
120 FORMAT (1H1, $ELEVATION 1-2-3: AZIMUTH 1-2-3: POSITIVE AXES LINE 1
   1. NEGATIVE AXES LINE 2.$ )
    ZIM(1)=ZI(1)/ZM(1):ZIM(2)=ZI(2)/ZM(2):ZIM(3)=ZI(3)/ZM(3)
    IF(ZL(1,2).EQ.0..AND.ZL(1,3).EQ.0.) NC=1
    IF (SENSE SWITCH 4) 2,23
  2 WRITE(102,102): READ(101,103)YO,AT,ONG,XLAT,XLON
 23 T7=AT: T8=ONG: T9=XLAT: T10=XLON: ONG=ONG-AL(3): XLON =XLON-AL(3)
  3 AT=AT*RAD: ONG=ONG*RAD: YO=YO*RAD: SAT=SIN(AT): SON=SIN(ONG)
    CAT=COS(AT):   CON=COS(ONG)
    T1=SIN(YO): T2= COS(YO): T3=T4-CAT*CON
    P2=ATAN(-CAT*SON/T3): SP2=SIN(P2)
    R2=ATAN(SAT/(COS(P2)*T3-CAT*SON*SP2) )
    T5=T1*COS(R2)
    YBS=ATAN(T5/SQRT(1.-T5**2))
    T5=SIN(R2)/COS(YBS)
    RBS=ATAN(T5/SQRT(1.-T5**2))
    T5=(SP2*T2-T1*SIN(R2)*COS(P2))/COS(YBS)
    PBS=ATAN(T5/SQRT(1.-T5**2))
    PRINT 105: PRINT 106,((ZL(I,J). J=1,3),I=1,3)
    PRINT 107: PRINT 106,YO/RAD,T7,T8,T9,T10,RBS,YBS,PBS,DT
    PRINT 108:PRINT 106,(ZI(I),I=1,3),(ZM(I),I=1,3),(AL(I),I=1,3)
    PRINT 109:PRINT 106,AX,BX,XF,AE: CALL INIT
    CALL STUP(0.,.35692,0.,RA): CALL STUP(AL(3),0.,0.,RB)
    DO 4 I=1,3: DO 4 J=1,3: RD(I,J)=0.: DO 4 K=1,3
  4 RD(I,J)=RD(I,J)+RA(I,K)*RB(K,J)
    CALL STUP(PIR,ROR,0.,RA)
    DO 5 I=1,3: DO 5 J=1,3: RB(J,I)=0.: DO 5 K=1,3
  5 RB(J,I)=RB(J,I)+RD(I,K)*RA(K,J)
C 5 TAKES THE TRASPOSE AS IT COMPUTES THE PRODUCT
    CALL STUP (PIB,ROB,YB,RA): SPIR=SIN(PIR): SROR=SIN(ROR)
    CPIR=COS(PIR): CROR=COS(ROR)
    IF(Z.NE.0.)           GO TO 16
  6 CALL TOR (RA,RB,SPIR,SROR,CPIR,CROR)
 16 CONTINUE
    T4=T7
    T5=T8
    T7=T(1)*PA/ZI(1): T8=T(2)*PA/ZI(2): T9=T(3)*PA/ZI(3)
    PRINT 110: PRINT 106,T(1),T(2),T(3),T7,T8,T9
    IF (NA)15,14,15
 15 PRINT 111: PRINT 115: PRINT 116: CALL OUT: GO TO 10
 14 PRINT 112: PRINT 105,TI,X(1)*PB,X(2)*PB,X(3)*PB,X(4)*PC,
   1X(5)*PC,X(6)*PC
    PRINT 113: WRITE(102,114): PAUSE 11
100 FORMAT (2I5,5F10.5)
101 FORMAT (9F8.5)
102 FORMAT($ TYPE POS YO,AT,ONG,TARGET LAT AND LONG 9F5.3$ )
103 FORMAT (9F5.3)
105 FORMAT (1H0,2X 10HLAMDA(1,1),4X10HLAMDA(1,2),4X10HLAMDA(1,3),4X10H
   1LAMDA(2,1),4X10HLAMDA(2,2),4X10HLAMDA(2,3),4X10HLAMDA(3,1),4X10HLA
   2MDA(3,2),4X10HLAMDA(3,3) )
106 FORMAT(1X,1P9E14.5)
107 FORMAT (1H0,4X2HYO,12X2HAT,11X3HONG,11X3HLAT,11X3HLON,11X3HRBS,11X
   23HYBS,11X3HPBS,12X2HDT )
108 FORMAT (1H0,4X8HMOMENT 1,6X8HMOMENT 2,6X8HMOMENT 3,8X4HZM 1,10X4HZ
```

```
      2M 2.10X4HZM 3.8X7HALPHA 1.7X7HALPHA 2.7X7HALPHA 3   )
  109 FORMAT (1H0,5X2HAX,12X2HBX,12X2HXF,12X2HAE   )
  110 FORMAT (1H0,4X8HTORQUE 1.6X8HTORQUE 2.6X8HTORQUE 3.15X14HSCALED TO
     1RQUES   )
  111 FORMAT (1H0, 5X 4HTIME, 10X4HX(1), 10X4HX(2), 10X 4HX(3), 10X
     14HX(4), 10X 4HX(5), 10X 4HX(6) )
  112 FORMAT (1H0,4X4HTIME, 19X 26HSCALED  VALUES  OF X(1-6)             )
  113 FORMAT (1H1 )
  114 FORMAT (10HSET ANALOG  )
  115 FORMAT (20X 4HD(1), 10X 4HD(2), 10X 4HD(3), 10X 4HD(4),
     110X 4HD(5), 10X 4HD(6) )
  116 FORMAT (20X 4HU(1), 10X 4HU(2), 10X 4HU(3), 10X 3HX1S, 11X 3HX4S,
     110X 1HS   )
    7 CALL CONDITION
                 CONNECT(40,ADCIN); CONNECT(43,DTOA); CONNECT(44,ATOD)
      CALL ENABLE; CALL ARM(40B,43B,44B); AC=1.; CALL REG; CALL DTOA
S     FOM 030010
    8 CONTINUE
S     FOM 030013
      CALL IDLE; AC=0.
   24 IF(7.NE.0.)  CALL REG; GO TO 19
    9 CALL TOR(RA,RB,SPIR,SROR,CPIR,CROR); CALL REG
   19 CONTINUE
      ASSIGN 32 TO NG
      GO TO 30
   32 CONTINUE
      L=L+1
      IF(L.GE.NB)CALL OUT; L=0
      AC=1.; CALL IDLE
      IF(NF(1).EQ.NF(2).EQ.NF(3).EQ.2) GO TO 21
C THIS IS THE DTOA IDLE
   22 IF(SENSE SWITCH 2) 20,8
   21 IF(SENSE SWITCH 3)       8,25
   25 U(1)=0; U(2)=0; U(3)=0; CALL DTOA; GO TO 21
   20 PAUSE 16; GO TO 13
   10 IF(7.NE.0.)            CALL REG; GO TO 17
      CALL TOR(RA,RB,SPIR,SROR,CPIR,CROR); CALL REG
   17 CONTINUE
      DO 11 J=1,NA
      CALL RK; L=L+1
      IF(NF(1).EQ.NF(2).EQ.NF(3).EQ. 2)  CALL OUT; GO TO 20
   27 IF(L.LT.NB)GO TO 11
      ASSIGN 31 TO NG
      GO TO 30
   31 CALL OUT
      L=0
      IF (SENSE SWITCH 2) 20 ,11
   11 CONTINUE
      IF(SENSE SWITCH 1) 12,10
C SET SWITCH 1 FOR NEW PRINT FREQUENCY
   12 READ(101,100)NB; GO TO 10
C FOLLOWING COMPUTES GROUND TRACK, WHERE T4  IS LAT AND T5  IS LONG
   30 CONTINUE
      S1=SIN(X(3))
```

```
S2=SIN(X(2))
S3=SIN(X(1))
C1=COS(X(3))
C2=COS(X(2))
C3=COS(X(1))
S7=SROR*S1*C2+CROR*S2
C7=SQRT(ABS(1.-S7**2))
S8=(SROR*(C1*C3-S1*S2*S3)+CROR*S3*C2)/C7
C8=SQRT(ABS(1.-S8**2))
S6=(SPIR*C1*C2+CPIR*CROR*S1*C2-CPIR*SROR*S2)/C7
C6=SQRT(ABS(1.-S6**2))
F2=ATAN(S7/C7/C8)
T2=SIN(F2)
D2=COS(F2)
T1=S6*C7*D2+T2*(C6*S8+S6*C8*S7)
D1=SQRT(ABS(1.-T1**2) )
T3=S8*C7
D3=SQRT(ABS(1.-T3**2))
Y1=T3*(ZK*D3*D1-SQRT(ABS(1.-ZK**2*(1.-D3**2*D1**2))))
Z1=SQRT(ABS(1.-Y1**2))
T4 =ATAN(Y1/Z1)/RAD
A1=SQRT(ABS(Z1**2-ZK**2*T1**2))
T5 =ATAN(T1*(A1-ZK*D1)/(A1+ZK*T1*T1)/D1)/RAD +AL(3)
GO TO NG,(32,31)
END
```

```
      SUBROUTINE REG
C (1-17),60,(63-66),68,70,73,(101-129),134,135,138
C ,200-203),(207-217),(222-225),345
      DIMENSION AL(3),NF(3),ETA(3),U0(3),C(3),TIM(3),TS(3),D(6)
      DIMENSION XX(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),X0(6),ZL(3,3),XA(6)
      COMMONXX,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX
      COMMON /MISAL/ TH(3,3,2),UX(3)
      IF(U(1))92,91,91
   91 K=1
      GO TO 93
   92 K=2
   93 IF(U(2))95,94,94
   94 L=1
      GO TO 96
   95 L=2
   96 IF(U(3))98,97,97
   97 M=1
      GO TO 99
   98 M=2
   99 UX(1)=TH(1,2,L)*U(2)+TH(1,3,M)*U(3)
      UX(2)=TH(2,1,K)*U(1)+TH(2,3,M)*U(3)
      UX(3)=TH(3,1,K)*U(1)+TH(3,2,L)*U(2)
      DO 115 I=1,3
  200 U0(I) = U(I)
      GO TO (201,202,203),I
  201 B=(T(1)+(ZI(2)-ZI(3))*XX(5)*XX(6)*AL(1))/ZM(1)
      B=B+UX(1)/ZM(1)
      T1=XX(4)-XX(2)*XX(6)*AL(1)
      GO TO 11
  203 B= ( T(3)+(ZI(1)-ZI(2))*XX(4)*XX(5)*AL(1))/ZM(3)
      B=B+UX(3)/ZM(3)
      T1=XX(6)+XX(1)*XX(5)*AL(1)
   11 X=XX(I)*ZIM(I)*SIGN(1.,-B)
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
      GO TO 18
  202 B=(T(2)+(ZI(3)-ZI(1))*XX(6)*XX(4)*AL(1))/ZM(2)
      B=B+UX(2)/ZM(2)
      T1=XX(5)-XX(1)*XX(6)*AL(1)
      X=XX(I)*ZIM(I)*SIGN(1., B)
      IF (B.EQ.0.) X=-X
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
   18 IF(X*Y)2,2,1
    1 U(I)=SIGN(ZM(I),-X)*SIGN(1.,-B)
      ETA(I)=-1.
      C(I)=1.
      GO TO 10
    2 T2=200.*DT
      IF(ZIM(I)*ABS(X0(I+3)-XX(I+3))-T2)12,12,13
   12 IF(ABS(X0(I)-XX(I))-T2*ABS(XX(I+3)))14,14,13
   13 ETA(I)=-1.
      C(I)=1.
```

```
      GO TO 9
   14 C(I)=C(I)*U(I)
    9 CONTINUE
    8 GO TO(209,345),NC
C NC=1 TIME OPTIMAL.208 USES MIN.209 USES MAX.
  345 IF(C(I))208,209,208
  208 IF(ABS(XX(I+3))-ABS(T1))212,212,210
  209 IF(ABS(XX(I+3))-ABS(T1))210,212,212
  210 Y=SIGN(T1*ZIM(I),Y)
  212 IF(ETA(I))207,207,213
  213 Y=Y*XF
  207 T3=Y*Y/2.
      KA = 1
      IF ( U(I).LE.0 ) KA = 2
      X1 = TH(I,I,KA)
      BA = -ABS(B)
      P2 = X - T3/(BA - X1 )
      P4 = X - T3/(BA + X1 )
      GO TO (63,60),NC
   63 IF(X)64,66,65
   64 IF(P2)17,16,16
   65 IF(P4)17,17,16
   66 IF(Y) 17,15,16
   60 IF(X)68,66,70
   68 IF(P2)73,16,16
   70 IF(P4)17,17,73
   73 CONTINUE
      T4=ABS(ZL(I,2)*B/ZL(I,3))
      R=SQRT(2.*T4)
      S=SQRT(ZL(I,1)/ZL(I,3)+T4)
      IF(X)3,66,5
    3 IF(Y-S) 19,4,4
   19 IF(B.EQ.0.) QX=-10000.; GO TO 20
      BA=-ABS(B); QX=(BA*S*S-2.*R*S+R*R)/(2.*BA*(BA-X1 ))
   20 IF(X-QX)17,17,21
   21 G=-ZL(I,1)-ZL(I,3)*Y**2+ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA-X1)*X   )
      IF(( BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA-X1)*X)17,15,15
    4 IF(Y-S-R-AE)15,15,16
    5 IF(Y-(R-S)) 6,15,22
   22 IF(B.EQ.0.) QX=10000.; GO TO 23
      BA=-ABS(B); QX=(BA*(R-S)**2-2.*R*S+R*R)/(2.*BA*(BA+X1))
   23 IF(X-QX)24,16,16
   24 G=-ZL(I,1)-ZL(I,3)*Y**2-ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA+X1)*X   )
      IF((-BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA+X1)*X)16,15,15
    6 IF(Y-(-S-AE))17,15,15
   15 U(I)=0.
      GO TO 7
   16 U(I)=SIGN(ZM(I),B)
      IF(B.EQ.0.) U(I)=-U(I)
      GO TO 7
   17 U(I)=SIGN(ZM(I),-B)
    7 IF(I.EQ.2) ETA(I)=-U(I)*XX(I); GO TO 10
```

```
      FTA(I)=U(I)*XX(I)
   10 XØ(I)=XX(I); XØ(I+3)=XX(I+3)
C 123 BEGINS TIME SYNCH
  123 X=XX(I)*ZIM(I)
      IF (I.EQ.2) X=-X
      Y=XX(I+3)*ZIM(I)
      T8=U(I)
  121 T5=Y*Y/2.
      P2=X-T5
      P4=X+T5
      GØ TØ(117,101),NC
  117 IF(Y)107,107,109
  107 IF(P2)110,110,111
  111 X=-X
      Y=-Y
      T8=-U(I)
      GØ TØ 121
  109 IF(P4)112,111,111
  110 TS(I)=-1.
      GØ TØ 113
  112 TS(I)=ABS(Y/2.)-X/Y
      IF(T8*Y.LE.0.) TS(I)=-1.
  113 TIM(I)=2.*SQRT(ABS(P2))-Y
      GØ TØ 211
  101 PS=X+(.5+2.*ZL(I,2)/(ZL(I,1)-ZL(I,3)*Y**2))*Y**2
      P=Y*ABS(Y)+2.*X
      S=SQRT(ZL(I,1)/ZL(I,3))
      IF(P)87,79,111
   87 IF(Y-S)74,89,89
   89 TS(I)=-1.
      TIM(I)=Y-P4/S
      GØ TØ 211
   74 IF(PS)75,75,76
   75 T9=Y**2-2.*X
      T8=2.*ZL(I,1)+4.*ZL(I,2)
      T0=SQRT((T8+ZL(I,3)*T9)**2-8.*ZL(I,1)*ZL(I,3)*T9      )
      T0=SQRT((T8+ZL(I,3)*T9-T0)/4./ZL(I,3))
C T0 IS ØMEGA S1
      TIM(I)=T0-P2/T0-Y
C REGIØN I
      IF(Y)77,77,78
   77 TS(I)=-1.
      GØ TØ 211
   78 TS(I)=Y/2.-X/Y
      IF(T8*Y.LE.0.) TS(I)=-1.
      GØ TØ 211
   76 IF (Y) 75,75,79
   79 TIM(I)=          Y/2.-X/Y
      GØ TØ 77
  211 IF(ABS(XX(I))-.0020) 222,223,223
  222 IF(ABS(XX(I+3))-.00005) 224,223,223
  224 U(I)=0.
      NF(I)=2
      TIM(I)=0.
```

```
      GO TO 115
  223 GO TO (115,216) , NF(I)
 216  IF(ABS(XX(I)) - .005)227,225,225
  227 IF(ABS(XX(I+3)) -.000125) 226,225,225
  226 U(I)=0.
      TIM(I)=0.
      GO TO 115
  225 NF(I)=1
  115 CONTINUE
  127 TMAX=AMAX(TIM(1),TIM(2),TIM(3))*AX
  129 DO 124 J=1,3
  128 GO TO (138,124),NF(J)
  138 IF(TS(J))124,125,125
  125 IF(TMAX-TS(J))124,124,134
  134 IF(TMAX-TS(J)-BX)135,126,126
  135 IF(UO(J))124,126,124
  126 U(J)=0.
  124 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DTOA
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
    1 IF(AC)1,10,2
    2 CALL DAC (0,T(1)*PA/ZI(1))
      CALL DAC (1,T(2)*PA/ZI(2))
      CALL DAC (2,T(3)*PA/ZI(3))
      CALL DAC(3,.01*T4)
      CALL DAC(4,.006*T5)
      IF(U(1))3,4,5
    3 CONTINUE
S       FOM 034110
S       POT =034110
S       FOM 034110
S       POT =034101
      GO TO 6
    4 CONTINUE
S       FOM 034110
S       POT =034110
S       FOM 034110
S       POT =034111
      GO TO 6
    5 CONTINUE
S       FOM 034110
S       POT =034100
S       FOM 034110
S       POT =034111
    6 IF(U(2))7,8,9
    7 CONTINUE
S       FOM 034110
S       POT =034112
S       FOM 034110
S       POT =034103
      GO TO 11
    8 CONTINUE
S       FOM 034110
S       POT =034112
S       FOM 034110
S       POT =034113
      GO TO 11
    9 CONTINUE
S       FOM 034110
S       POT =034102
S       FOM 034110
S       POT =034113
   11 IF(U(3))12,13,14
   12 CONTINUE
S       FOM 034110
S       POT =034114
S       FOM 034110
S       POT =034105
      GO TO 15
```

```
   13 C0NTINUE
S        F0M 0341
S        P0T =034
S        FPM 034110
S        P0T =034115
         G0 T0 15
   14 C0NTINUE
S        F0M 034110
S        P0T =034104
S        F0M 034110
S        P0T =034115
   15 RETURN
   10 PRINT 100
  100 F0RMAT (14H TIMING IS 0FF )
      RETURN
      END
```

```
SUBRØUTINE ATØD
DIMENSIØN X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
CØMMØN X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
CALL ADC (0,VA)
CALL ADC (3,VD)
CALL ADC (1,VB)
CALL ADC (4,VE)
CALL ADC (2,VC)
CALL ADC (5,VF)
X(1)=VA/PB; X(2)=VB/PB; X(3)=VC/PB
X(4)=VD/PC; X(5)=VE/PC; X(6)=VF/PC ;RETURN
END
```

```
      SUBROUTINE OUT
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /TEMP/ T1,T2,T3,T4,T5,T6,T7,T8,T9,TO
      IF(NA)1,2,1
    1 WRITE(6,112)TI,X,14,T5
      WRITE(6,113)D
      WRITE(6,113)U, T(1),X(1)*ZIM(1),X(4)*ZIM(1),S
      RETURN
    2 WRITE (6,114) TI,X,U
      RETURN
  112 FORMAT  (1H0,1P9E14.5)
  113 FORMAT (15X, 1P7E14.5)
  114 FORMAT (1H0, F6.2, 1P9E12.3)
      END
```

```
      SUBROUTINE STUP (CP,CR,Y,CA)
      DIMENSION CA(3,3)
      SINR=SIN(CR)
      SINR=SIN(CR); SINP=SIN(CP); SINY=SIN(Y)
      COSR=COS(CR); COSP=COS(P); COSY=COS(Y)
C WE COMPUTE THE MATRIX I+B
      CA(1,1)=COSP*COSY; CA(1,2)=-COSP*SINR*SINY-SINP*COSR
      CA(1,3)=-COSP*COSR*SINY+SINP*SINR; CA(2,1)=SINP*COSY
      CA(2,2)=-SINP*SINR*SINY+COSP*COSR
      CA(2,3)=-SINP*COSR*SINY-COSP*SINR; CA(3,1)=SINY
      CA(3,2)=SINR*COSY; CA(3,3)=COSR*COSY
      RETURN
      END
```

```
      SUBROUTINE    INIT
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,PBS,PIB,YB,ROB
      X(4)=0.; X(5)=0.; X(6)=0.
      XLON = XLON *0.017453
      XLAT =XLAT *0.017453
      SILAT = SINF(XLAT)
      SILON = SINF (XLON)
      COLAT = COSF (XLAT)
      COLON = COSF (XLON)
      CONST=6.610808
      PITCH =-ATANF((+SILON * COLAT)/(CONST - COLON * COLAT))
      COPI = COSF (PITCH)
      SINPI = SINF (PITCH)
      ROLL = ATANF (SILAT/((CONST -COLON*COLAT)* COPI -SILON*COLAT *
     1SINPI))
      PIR= PITCH
      ROR =ROLL
      CRBS = COSF(RBS)
      GO TO 200
  200 CONTINUE
      SRBS = SINF (RBS)
      CYBS = COSF (YBS)
      SYBS = SINF (YBS)
      CPIB = COSF (PIB)
      CYB = COSF (YB)
      CROB = COSF (ROB)
      SYB = SINF (YB)
      SROB = SINF (ROB)
      SPIB = SINF (PIB)
      SPIR = SINF (PIR)
      SROR = SINF (ROR)
      CPIR = COSF (PIR)
      CROR = COSF (ROR)
      SPRB = SINF (PIR - PBS)
      CPRB = COSF (PIR - PBS)
      G1 = SROR * CYBS * SPRB + CROR * SYBS
   78 FORMAT (F9.5)
      G2 = SQRTF (1.-G1**2)
      X(2) =  ATANF (G1/G2)
      G3=(-CROR*CYBS*SPRB+SROR*SYBS)/G2
      G4=SQRTF(1.-G3*G3)
      X(3)=ATANF(G3/G4)
      G6 =(-SROR * CRBS  * CPRB - SROR* SRBS *SYBS *SPRB + CROR*SRBS *
     1CYBS)/G2
      G7= SQRTF (1 - G6**2)
      X(1)= ATANF (G6/G7)
      RETURN
      END
```

```
      SUBROUTINE DER
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /MISAL/ TH(3,3,2),UX(3)
      IF(U(1))2,1,1
    1 K=1
      GO TO 3
    2 K=2
    3 IF(U(2)) 5,4,4
    4 L=1
      GO TO 6
    5 L=2
    6 IF(U(3)) 8,7,7
    7 M=1
      GO TO 9
    8 M=2
    9 DO 10 I=1,3
   10 UX(I)=TH(I,1,K)*U(1)+TH(I,2,L)*U(2)+TH(I,3,M)*U(3)
      D(1)=-AL(1)* X(2)*X(6) + X(4)
      D(2)= AL(1) * X(1) *X(6) - X(5)
      D(3)= AL(1)*X(1)*X(5) + X(6)
      D(4)= (AL(1) *(ZI(2)-ZI(3))* X(5)* X(6) +UX(1) + T(1))/ZI(1)
      D(5) = (AL(1) *(ZI(3) -ZI(1))*X(4)*X(6)+UX(2) +T(2))/ZI(2)
      D(6)= (AL(1)* (ZI(1)-ZI(2))* X(4) * X(5) +UX(3) + T(3))/ZI(3)
      RETURN
      END
```

```
SUBROUTINE TOR(RA,RB,SPIR,SROR,CPIR,CROR)
DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
DIMENSION RA(3,3),RB(3,3),RC(3,3),RD(3,3),G(3,3),XI(3,3)
COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
T(1)=1.E-7
T(2)=1.E-7
T(3)=1.E-7
RETURN
END
```

```
      SUBROUTINE RK
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      DIMENSION R(6),SA(6), SB(6),SC(6),SD(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,Z,S,D,NA,TI
      DO 2 N=1,6
    2 R(N)=X(N)
      CALL DER
      DO 3 N=1,6
    3 SA(N)=D(N)*DT
      DO 4 N=1,6
    4 X(N)=R(N)+SA(N)/2.
      TI=TI+DT/2.
      CALL DER
      DO 5 N=1,6
    5 SB(N)=D(N)*DT
      DO 6 N=1,6
    6 X(N)=R(N)+SB(N)/2.
      CALL DER
      DO 7 N=1,6
    7 SC(N)=D(N)*DT
      TI=TI+DT/2.
      DO 8 N=1,6
    8 X(N)=R(N)+SC(N)
      CALL DER
      DO 9 N=1,6
      SD(N)=D(N)*DT
    9 X(N)=R(N)+(SA(N)+SD(N))/6.+(SB(N)+SC(N))/3.
      RETURN
      END
```

```
C                                SENSOR MODEL
C                                                             DATE 16 OCT 67
C
C
C
C
C1 NA,NB,PA,PB,PC                                   2I5,5F10.5
C2 ZI(1-3), ZM(1-3)), AL(1-3),                      9F8.5
C3 ZL(1,1), (1,2).......(3,3)                       9F8.5
C4 AX,BX,XF,ZK1,ZK2,SD,PIB,ROB,YB                  9F8.5
C5 YO,AT,ANG,XLAT,XLON,DT,AE                        9F8.5
C   SW1 TYPE NEW PRINT FREQUENCY I5
C   SW2 READ NEW DATA AND RESTART
C   SWITCH 3 IGNORES FUEL CALCULATION
C   SW4 TYPE YO, AT, ANG, XLAT, XLON 9F5.3
C   XF PREVENTS JAGS COMING IN; AE INCREASES ZERO BAND;AX,BX LEAD FACTORS
C   (1-25) (30-37) (100-116)
C
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      DIMENSION PA(3,3),RB(3,3),RC(3,3),RD(3,3),ETA(3),C(3),NF(3)
      DIMENSION AP(3),XX(6)
      DIMENSION ELV(3,2),AZ(3,2)
      COMMON /MISAL/ TH(3,3,2),UX(3)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX ,XX
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,RBS,PIB,YB,ROB
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
   13 TI=0.; L=0; T4=6.610806; NC=2; RAD=.017453       ;AC=1.
      PRINT 113
      ZK=T4
      DO 1 I=1,3
      NF(I)=1.; U(I)=ZM(I); C(I)=1.
    1 CONTINUE
      READ 100,NA,NB,PA,PB,PC,Z
      READ 101,(ZI(I),I=1,3),(ZM(I),I=1,3),(AL(I),I=1,3)
      READ 101,((ZL(I,J),J=1,3),I=1,3)
      READ 101,AX,BX,XF,ZK1,ZK2,SD,PIB,ROB,YB
      OK=1.-ZK1
      READ 101,YO,AT,ANG,XLAT,XLON,DT,AE
      SIG=DT
      IF(NA.NE. 0) SIG=SIG*FLOAT(NA)
      S2=SIG**2/2.
      IF(Z.NE.0.)       READ 101,T(1),T(2),T(3)
C     ELV(I,J)=ELEVATION IN DEGREES FOR AXIS I,J=1 PLUS,2 MINUS
C     AZ(I,J)=AZIMUTH
      READ 101,(ELV(I,1),I=1,3),(AZ(I,1),I=1,3)
      READ 101,(ELV(I,2),I=1,3),(AZ(I,2),I=1,3)
      DO 40 J=1,2
      DO 40 I=1,3
      ELV(I,J)=ELV(I,J)*RAD
   40 AZ(I,J)=AZ(I,J)*RAD
      DO 42 K=1,2
      DO 41 I=1,3
```

```
   41 TH(I,I,K)=COS(ELV(I,K))*COS(AZ(I,K))
      TH(2,1,K)=SIN(ELV(1,K))
      TH(3,1,K)=COS(ELV(1,K))*SIN(AZ(1,K))
      TH(1,2,K)=COS(ELV(2,K))*SIN(AZ(2,K))
      TH(3,2,K)=SIN(ELV(2,K))
      TH(1,3,K)=SIN(ELV(3,K))
   42 TH(2,3,K)=COS(ELV(3,K))*SIN(AZ(3,K))
      PRINT 120
      PRINT 101,(ELV(I,1)/RAD,I=1,3),(AZ(I,1)/RAD,I=1,3)
      PRINT 101,(ELV(I,2)/RAD,I=1,3),(AZ(I,2)/RAD,I=1,3)
      PRINT 117
      PRINT 118,((TH(I,J,1),J=1,3),I=1,3)
      PRINT 119
      PRINT 118,((TH(I,J,2),J=1,3),I=1,3)
      PRINT 113
  117 FORMAT (1H1,5X, $MULTIPLIERS FOR POSITIVE THRUSTS$,////)
  118 FORMAT (3F15.8)
  119 FORMAT (1H0,5X, $MULTIPLIERS FOR NEGATIVE THRUSTS$,////)
  120 FORMAT (1H1, $ELEVATION 1-2-3: AZIMUTH 1-2-3: POSITIVE AXES LINE 1
     1. NEGATIVE AXES LINE 2.$ )
      ZIM(1)=ZI(1)/ZM(1);ZIM(2)=ZI(2)/ZM(2);ZIM(3)=ZI(3)/ZM(3)
      IF(ZL(1,2).EQ.0..AND.ZL(1,3).EQ.0.) NC=1
      IF (SENSE SWITCH 4) 2,23
    2 WRITE(102,102); READ(101,103)YO,AT,ONG,XLAT,XLON
   23 T7=AT; T8=ONG; T9=XLAT; T10=XLON; ONG=ONG-AL(3); XLON =XLON-AL(3)
    3 AT=AT*RAD; ONG=ONG*RAD; YO=YO*RAD; SAT=SIN(AT); SON=SIN(ONG)
      CAT=COS(AT);  CON=COS(ONG)
      T1=SIN(YO); T2=COS(YO);T3=T4-CAT*CON
      P2=ATAN(-CAT*SON/T3); SP2=SIN(P2)
      R2=ATAN(SAT/(COS(P2)*T3-CAT*SON*SP2) )
      T5=T1*COS(R2)
      YBS=ATAN(T5/SQRT(1.-T5**2))
      T5=SIN(R2)/COS(YBS)
      RBS=ATAN(T5/SQRT(1.-T5**2))
      T5=(SP2*T2-T1*SIN(R2)*COS(P2))/COS(YBS)
      PBS=ATAN(T5/SQRT(1.-T5**2))
      PRINT 105; PRINT 106,((ZL(I,J), J=1,3),I=1,3)
      PRINT 107; PRINT 106,YO/RAD,T7,T8,T9,T10,RBS,YBS,PBS,DT
      PRINT 108;PRINT 106,(ZI(I),I=1,3),(ZM(I),I=1,3),(AL(I),I=1,3)
      PRINT 109;PRINT 106,AX,BX,XF,AE; CALL INIT
      DO 37 I=1,3
      XX(I)=X(I)
   37 XX(I+3)=0.
      CALL STUP(0.,.35692,0.,RA); CALL STUP(AL(3),0.,0.,RB)
      DO 4 I=1,3; DO 4 J=1,3; RD(I,J)=0.; DO 4 K=1,3
    4 RD(I,J)=RD(I,J)+RA(I,K)*RB(K,J)
      CALL STUP(PIR,ROR,0.,RA)
      DO 5 I=1,3; DO 5 J=1,3; RB(J,I)=0.; DO 5 K=1,3
    5 RB(J,I)=RB(J,I)+RD(I,K)*RA(K,J)
C 5 TAKES THE TRASPOSE AS IT COMPUTES THE PRODUCT
      CALL STUP (PIB,ROB,YB,RA); SPIR=SIN(PIR); SROR=SIN(ROR)
      CPIR=COS(PIR); CROR=COS(ROR)
      IF(Z.NE.0.)           GO TO 16
    6 CALL TOR (RA,RB,SPIR,SROR,CPIR,CROR)
```

```
 16 CONTINUE
    T4=T7
    T5=T8
    T7=T(1)*PA/ZI(1); T8=T(2)*PA/ZI(2); T9=T(3)*PA/ZI(3)
    PRINT 110; PRINT 106,T(1),T(2),T(3),T7,T8,T9
    IF (NA)15,14,15
 15 PRINT 111; PRINT 115; PRINT 116; CALL OUT; GO TO 10
 14 PRINT 112; PRINT 106,TI,X(1)*PB,X(2)*PB,X(3)*PB,X(4)*PC,
   1X(5)*PC,X(6)*PC
    PRINT 113; WRITE(102,114); PAUSE 11
100 FORMAT (2I5,5F10.5)
101 FORMAT (9F8.5)
102 FORMAT($ TYPE POS YO,AT,ONG,TARGET LAT AND LONG 9F5.3$ )
103 FORMAT (9F5.3)
105 FORMAT (1H0,2X 10HLAMDA(1,1),4X10HLAMDA(1,2),4X10HLAMDA(1,3),4X10H
   1LAMDA(2,1),4X10HLAMDA(2,2),4X10HLAMDA(2,3),4X10HLAMDA(3,1),4X10HLA
   2MDA(3,2),4X10HLAMDA(3,3) )
106 FORMAT(1X,1P9E14.5)
107 FORMAT (1H0,4X2HYO,12X2HAT,11X3HONG,11X3HLAT,11X3HLON,11X3HRBS,11X
   23HYBS,11X3HPBS,12X2HDT )
108 FORMAT (1H0,4X8HMOMENT 1,6X8HMOMENT 2,6X8HMOMENT 3,8X4HZM 1,10X4HZ
   2M 2,10X4HZM 3,8X7HALPHA 1,7X7HALPHA 2,7X7HALPHA 3  )
109 FORMAT (1H0,5X2HAX,12X2HBX,12X2HXF,12X2HAE  )
110 FORMAT (1H0,4X8HTORQUE 1,6X8HTORQUE 2,6X8HTORQUE 3,15X14HSCALED TO
   1RQUES  )
111 FORMAT (1H0, 5X 4HTIME, 10X4HX(1), 10X4HX(2), 10X 4HX(3), 10X
   14HX(4), 10X 4HX(5), 10X 4HX(6) )
112 FORMAT (1H0,4X4HTIME, 19X 26HSCALED  VALUES  OF X(1-6)            )
113 FORMAT (1H1 )
114 FORMAT (10HSET ANALOG  )
115 FORMAT (20X 4HD(1), 10X 4HD(2), 10X 4HD(3), 10X 4HD(4),
   110X 4HD(5) 10X 4HD(6) )
116 FORMAT (20X 4HU(1), 10X 4HU(2), 10X 4HU(3), 10X 3HX1S, 11X 3HX4S,
   110X 1HS  )
  7 CALL CONDITION
                CONNECT(40,ADCIN); CONNECT(43,DTOA); CONNECT(44,ATOD)
    CALL ENABLE; CALL ARM(40B,43B,44B); AC=1.; CALL REG; CALL DTOA
S   FOM 030010
  8 CONTINUE
S   FOM 030013
    CALL IDLE; AC=0.
 24 IF(Z.NE.0.)  CALL REG; GO TO 19
  9 CALL TOR(RA,RB,SPIR,SROR,CPIR,CROR); CALL REG
 19 ASSIGN 34 TO NG
    GO TO 35
 34 CONTINU   E
    ASSIGN 32 TO NG
    GO TO 30
 32 CONTINUE
    L=L+1
    IF(L.GE.NB)CALL OUT; L=0
    AC=1.; CALL IDLE
    IF(NF(1).EQ.NF(2).EQ.NF(3).EQ.2) GO TO 21
C THIS IS THE DTOA IDLE
```

```
   22 IF(SENSE SWITCH 2) 20,8
   21 IF(SENSE SWITCH 3)         8,25
   25 U(1)=0; U(2)=0; U(3)=0; CALL DTOA; GO TO 21
   20 PAUSE 16; GO TO 13
   10 IF(7.NE.0.)          CALL REG; GO TO 17
      CALL TOR(RA,RB,SPIR,SROR,CPIR,CROR); CALL REG
   17 CONTINUE
      DO 11 J=1,NA
      CALL RK; L=L+1
      ASSIGN 33 TO NG
      GO TO 35
   33 CONTINUE
      IF(NF(1).EQ.NF(2).EQ.NF(3).EQ. 2)  CALL OUT; GO TO 20
   27 IF(L.LT.NB)GO TO 11
      ASSIGN 31 TO NG
      GO TO 30
   31 CALL OUT
      L=0
      WRITE(6,106)XX,AP
      IF (SENSE SWITCH 2) 20 ,11
   11 CONTINUE
      IF(SENSE SWITCH 1) 12,10
C SET SWITCH_1 FOR NEW PRINT FREQUENCY
   12 READ(101,100)NB; GO TO 10
C
C  FOLLOWING IS RANDOM NUMBER GENERATOR
   35 IA=ABS(IA*KA)
      T1=FLOAT(IA)/CA
      IA=ABS(IA*KA)
      T2=FLOAT(IA)/CA
      T9=SQRT(-2.*ALOG(T1))
      T8=6.28318531*ALOG(T2)
      T0=T9*COS(T8)
C
C  FOLLOWING IS SENSOR PROGRAM
C     AP ARE ALPHA PRIMES (ANGLES PLUS NOISE)
C     XX(1-3) ARE FILTERED ANGLES;XX(4-6) ARE ESTIMATED RATES
      DO 36 JA=1,3
      AP(JA)=X(JA)+SD*T0
      T1=XX(JA)+SIG*XX(JA+3)+S2*U(JA)/ZI(JA)
      XX(JA)=T1*OK+ZK1*AP(JA)
   36 XX(JA+3)=XX(JA+3)+SIG*U(JA)/ZI(JA)+ZK2*(AP(JA)-T1)
      GO TO NG,(33,34)
C
C FOLLOWING COMPUTES GROUND TRACK, WHERE T4  IS LAT AND T5  IS LONG
   30 CONTINUE
      S1=SIN(X(3))
      S2=SIN(X(2))
      S3=SIN(X(1))
      C1=COS(X(3))
      C2=COS(X(2))
      C3=COS(X(1))
      S7=SROR*S1*C2+CROR*S2
      C7=SQRT(ABS(1.-S7**2))
```

```
S8=(SROR*(C1*C3-S1*S2*S3)+CROR*S3*C2)/C7
C8=SQRT(ABS(1.-S8**2))
S6=(SPIR*C1*C2+CPIR*CROR*S1*C2-CPIR*SROR*S2)/C7
C6=SQRT(ABS(1.-S6**2))
F2=ATAN(S7/C7/C8)
T2=SIN(F2)
D2=COS(F2)
T1=S6*C7*D2+T2*(C6*S8+S6*C8*S7)
D1=SQRT(ABS(1.-T1**2) )
T3=S8*C7
D3=SQRT(ABS(1.-T3**2))
Y1=T3*(ZK*D3*D1-SQRT(ABS(1.-ZK**2*(1.-D3**2*D1**2))))
Z1=SQRT(ABS(1.-Y1**2))
T4 =ATAN(Y1/Z1)/RAD
A1=SQRT(ABS(Z1**2-ZK**2*T1**2))
T5 =ATAN(T1*(A1-ZK*D1)/(A1+ZK*T1*T1)/D1)/RAD +AL(3)
DATA IA,KA,CA/1,4099,8388638./
GO TO NG,(32,31)
END
```

```
      SUBROUTINE REG
C (1-17),60,(63-66),68,70,73,(101-129),134,135,138
C ,200-203),(207-217),(222-225),345
      DIMENSION AL(3),NF(3),ETA(3),US(3),C(3),TIM(3),TS(3),D(6)
      DIMENSION XX(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),X0(6),ZL(3,3),XA(6)
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX ,XX
      COMMONXA,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,II
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
      COMMON /MISAL/ TH(3,3,2),UX(3)
      IF(U(1))92,91,91
   91 K=1
      GO TO 93
   92 K=2
   93 IF(U(2))95,94,94
   94 L=1
      GO TO 96
   95 L=2
   96 IF(U(3))98,97,97
   97 M=1
      GO TO 99
   98 M=2
   99 UX(1)=TH(1,2,L)*U(2)+TH(1,3,M)*U(3)
      UX(2)=TH(2,1,K)*U(1)+TH(2,3,M)*U(3)
      UX(3)=TH(3,1,K)*U(1)+TH(3,2,L)*U(2)
      DO 115 I=1,3
  200 US(I) = U(I)
      GO TO (201,202,203),I
  201 B=(T(1)+(ZI(2)-ZI(3))*XX(5)*XX(6)*AL(1))/ZM(1)
      B=B+UX(1)/ZM(1)
      T1=XX(4)-XX(2)*XX(6)*AL(1)
      GO TO 11
  203 B= ( T(3)+(ZI(1)-ZI(2))*XX(4)*XX(5)*AL(1))/ZM(3)
      B=B+UX(3)/ZM(3)
      T1=XX(6)+XX(1)*XX(5)*AL(1)
   11 X=XX(I)*ZIM(I)*SIGN(1.,-B)
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
      GO TO 18
  202 B=(T(2)+(ZI(3)-ZI(1))*XX(6)*XX(4)*AL(1))/ZM(2)
      B=B+UX(2)/ZM(2)
      T1=XX(5)-XX(1)*XX(6)*AL(1)
      X=XX(I)*ZIM(I)*SIGN(1., B)
      IF (B.EQ.0.) X=-X
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
   18 IF(X*Y)2,2,1
    1 U(I)=SIGN(ZM(I),-X)*SIGN(1.,-B)
      ETA(I)=-1.
      C(I)=1.
      GO TO 10
    2 T2=200.*DT
      IF(ZIM(I)*ABS(X0(I+3)-XX(I+3))-T2)12,12,13
   12 IF(ABS(X0(I)-XX(I))-T2*ABS(XX(I+3)))14,14,13
   13 ETA(I)=-1.
      C(I)=1.
```

```
      GO TO 9
   14 C(I)=C(I)*U(I)
    9 CONTINUE
    8 GO TO(209,345),NC
C NC=1 TIME OPTIMAL.208 USES MIN.209 USES MAX.
  345 IF(C(I))208,209,208
  208 IF(ABS(XX(I+3))-ABS(T1))212,212,210
  209 IF(ABS(XX(I+3))-ABS(T1))210,212,212
  210 Y=SIGN(T1*ZIM(I),Y)
  212 IF(ETA(I))207,207,213
  213 Y=Y*XF
  207 T3=Y*Y/2.
      KA = 1
      IF ( U(I).LE.0 ) KA = 2
      X1 = TH(I,I,KA)
      BA = -ABS(B)
      P2 = X - T3/(BA - X1 )
      P4 = X - T3/(BA + X1 )
      GO TO (63,60),NC
   63 IF(X)64,66,65
   64 IF(P2)17,16,16
   65 IF(P4)17,17,16
   66 IF(Y) 17,15,16
   60 IF(X)68,66,70
   68 IF(P2)73,16,16
   70 IF(P4)17,17,73
   73 CONTINUE
      T4=ABS(ZL(I,2)*B/ZL(I,3))
      R=SQRT(2.*T4)
      S=SQRT(ZL(I,1)/ZL(I,3)+T4)
      IF(X)3,66,5
    3 IF(Y-S) 19,[illegible],4
   19 IF(B.EQ.0.) QX=-10000.; GO TO 20
      BA=-ABS(B); QX=(BA*S*S-2.*R*S+R*R)/(2.*BA*(BA-X1 ))
   20 IF(X-QX)17,17,21
   21 G=-ZL(I,1)-ZL(I,3)*Y**2+ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA-X1)*X   )
      IF(( BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA-X1)*X)17,15,15
    4 IF(Y-S-R-AE)15,15,16
    5 IF(Y-(R-S)) 6,15,22
   22 IF(B.EQ.0.) QX=10000.; GO TO 23
      BA=-ABS(B); QX=(BA*(R-S)**2-2.*R*S+R*R)/(2.*BA*(BA+X1))
   23 IF(X-QX)24,16,16
   24 G=-ZL(I,1)-ZL(I,3)*Y**2-ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA+X1)*X   )
      IF((-BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA+X1)*X)16,15,15
    6 IF(Y-(-S-AE))17,15,15
   15 U(I)=0.
      GO TO 7
   16 U(I)=SIGN(ZM(I),B)
      IF(B.EQ.0.) U(I)=-U(I)
      GO TO 7
   17 U(I)=SIGN(ZM(I),-B)
    7 IF(I.EQ.2) ETA(I)=-U(I)*XX(I); GO TO 10
```

```
      FTA(I)=U(I)*XX(I)
   10 X0(I)=XX(I); X0(I+3)=XX(I+3)
C 123 BEGINS TIME SYNCH
  123 X=XX(I)*ZIM(I)
      IF (I.EQ.2) X=-X
      Y=XX(I+3)*ZIM(I)
      T8=U(I)
  121 T5=Y*Y/2.
      P2=X-T5
      P4=X+T5
      GO TO(117,101),NC
  117 IF(Y)107,107,109
  107 IF(P2)110,110,111
  111 X=-X
      Y=-Y
      T8=-U(I)
      GO TO 121
  109 IF(P4)112,111,111
  110 TS(I)=-1.
      GO TO 113
  112 TS(I)=ABS(Y/2.)-X/Y
      IF(T8*Y.LE.O.) TS(I)=-1.
  113 TIM(I)=2.*SQRT(ABS(P2))-Y
      GO TO 211
  101 PS=X+(.5+2.*ZL(I,2)/(ZL(I,1)-ZL(I,3)*Y**2))*Y**2
      P=Y*ABS(Y)+2.*X
      S=SQRT(ZL(I,1)/ZL(I,3))
      IF(P)87,79,111
   87 IF(Y-S)74,89,89
   89 TS(I)=-1.
      TIM(I)=Y-P4/S
      GO TO 211
   74 IF(PS)75,75,76
   75 T9=Y**2-2.*X
      T8=2.*ZL(I,1)+4.*ZL(I,2)
      T0=SQRT((T8+ZL(I,3)*T9)**2-8.*ZL(I,1)*ZL(I,3)*T9      )
      T0=SQRT((T8+ZL(I,3)*T9-T0)/4./ZL(I,3))
C T0 IS OMEGA S1
      TIM(I)=T0-P2/T0-Y
C REGION I
      IF(Y)77,77,78
   77 TS(I)=-1.
      GO TO 211
   78 TS(I)=Y/2.-X/Y
      IF(T8*Y.LE.O.) TS(I)=-1.
      GO TO 211
   76 IF (Y) 75,75,79
   79 TIM(I)=          Y/2.-X/Y
      GO TO 77
  211 IF(ABS(XX(I))-.0020) 222,223,223
  222 IF(ABS(XX(I+3))-.00005) 224,223,223
  224 U(I)=0.
      NF(I)=2
      TIM(I)=0.
```

```
      GO TO 115
223   GO TO (115,216) , NF(I)
216   IF(ABS(XX(I))-.005)226,225,225
226   U(I)=0.
      TIM(I)=0.
      GO TO 115
225   NF(I)=1
115   CONTINUE
127   TMAX=AMAX(TIM(1),TIM(2),TIM(3))*AX
129   DO 124 J=1,3
128   GO TO (138,124),NF(J)
138   IF(TS(J))124,125,125
125   IF(TMAX-TS(J))124,124,134
134   IF(TMAX-TS(J)-BX)135,126,126
135   IF(U0(J))124,126,124
126   U(J)=0.
124   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DTOA
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,TO
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
    1 IF(AC)1,10,2
    2 CALL DAC (0,T(1)*PA/ZI(1))
      CALL DAC (1,T(2)*PA/ZI(2))
      CALL DAC (2,T(3)*PA/ZI(3))
      CALL DAC(3,.01*T4)
      CALL DAC(4,.006*T5)
      IF(U(1))3,4,5
    3 CONTINUE
S     FOM 034110
S     POT =034110
S     FOM 034110
S     POT =034101
      GO TO 6
    4 CONTINUE
S     FOM 034110
S     POT =034110
S     FOM 034110
S     POT =034111
      GO TO 6
    5 CONTINUE
S     FOM 034110
S     POT =034100
S     FOM 034110
S     POT =034111
    6 IF(U(2))7,8,9
    7 CONTINUE
S     FOM 034110
S     POT =034112
S     FOM 034110
S     POT =034103
      GO TO 11
    8 CONTINUE
S     FOM 034110
S     POT =034112
S     FOM 034110
S     POT =034113
      GO TO 11
    9 CONTINUE
S     FOM 034110
S     POT =034102
S     FOM 034110
S     POT =034113
   11 IF(U(3))12,13,14
   12 CONTINUE
S     FOM 034110
S     POT =034114
S     FOM 034110
S     POT =034105
      GO TO 15
```

```
   13 CONTINUE
S        FOM 034110
S        POT =034114
S        FOM 034110
S        POT =034115
         GO TO 15
   14 CONTINUE
S        FOM 034110
S        POT =034104
S        FOM 034110
S        POT =034115
   15 RETURN
   10 PRINT 100
  100 FORMAT (14H TIMING IS OFF )
      RETURN
      END
```

```
SUBROUTINE ATOD
DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
CALL ADC (0,VA)
CALL ADC (3,VD)
CALL ADC (1,VB)
CALL ADC (4,VE)
CALL ADC (2,VC)
CALL ADC (5,VF)
X(1)=VA/PB; X(2)=VB/PB; X(3)=VC/PB
X(4)=VD/PC; X(5)=VE/PC; X(6)=VF/PC ;RETURN
END
```

```
      SUBROUTINE OUT
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /TEMP/ T1,T2,T3,T4,T5,T6,T7,T8,T9,TO
      IF(NA)1,2,1
    1 WRITE(6,112)TI,X,T4,T5
      WRITE(6,113)D
      WRITE(6,113)U, T(1),X(1)*ZIM(1),X(4)*ZIM(1),S
      RETURN
    2 WRITE (6,114) TI,X,U
      RETURN
  112 FORMAT  (1HO,1P9E14.5)
  113 FORMAT (15X, 1P7E14.5)
  114 FORMAT (1HO, F6.2, 1P9E12.3)
      END
```

```
      SUBROUTINE STUP (CP,CR,CY,CA)
      DIMENSION CA(3,3)
      SINR=SIN(CR)
      SINR=SIN(CR); SINP=SIN(CP); SINY=SIN(Y)
      COSR=COS(CR); COSP=COS(P); COSY=COS(Y)
C WE COMPUTE THE MATRIX I+B
      CA(1,1)=COSP*COSY; CA(1,2)=-COSP*SINR*SINY-SINP*COSR
      CA(1,3)=-COSP*COSR*SINY+SINP*SINR; CA(2,1)=SINP*COSY
      CA(2,2)=-SINP*SINR*SINY+COSP*COSR
      CA(2,3)=-SINP*COSR*SINY-COSP*SINR; CA(3,1)=SINY
      CA(3,2)=SINR*COSY; CA(3,3)=COSR*COSY
      RETURN
      END
```

```
      SUBROUTINE    INIT
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,RBS,PIB,YB,ROB
      X(3)=0.
      X(4)=0.
      X(5)=0.
      XLON = XLON *0.017453
      XLAT =XLAT *0.017453
      SILAT = SINF(XLAT)
      SILON = SINF (XLON)
      COLAT = COSF (XLAT)
      COLON = COSF (XLON)
      CONST=6.610808
      PITCH =-ATANF((+SILON * COLAT)/(CONST - COLON * COLAT))
      COPI = COSF (PITCH)
      SINPI = SINF (PITCH)
      ROLL = ATANF (SILAT/((CONST -COLON*COLAT)* COPI -SILON*COLAT *
     1SINPI))
      PIR= PITCH
      ROR =ROLL
      CRBS = COSF(RBS)
      GO TO 200
  200 CONTINUE
      SRBS = SINF (RBS)
      CYBS = COSF (YBS)
      SYBS = SINF (YBS)
      CPIB = COSF (PIB)
      CYB = COSF (YB)
      CROB = COSF (ROB)
      SYB = SINF (YB)
      SROB = SINF (ROB)
      SPIB = SINF (PIB)
      SPIR = SINF (PIR)
      SROR = SINF (ROR)
      CPIR = COSF (PIR)
      CROR = COSF (ROR)
      SPRB = SINF (PIR - PBS)
      CPRB = COSF (PIR - PBS)
      G1 = SROR * CYBS * SPRB + CROR * SYBS
   78 FORMAT (F9.5)
      G2 = SQRTF (1.-G1**2)
      X(2) =  ATANF (G1/G2)
      G3=(-CROR*CYBS*SPRB+SROR*SYBS)/G2
      G4=SQRTF(1.-G3*G3)
      X(3)=ATANF(G3/G4)
      G6 =(-SROR * CRBS  * CPRB - SROR* SRBS *SYBS *SPRB + CROR*SRBS *
     1CYBS)/G2
      G7= SQRTF (1 - G6**2)
      X(1)= ATANF (G6/G7)
      RETURN
      END
```

```
      SUBROUTINE DER
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,II
      COMMON /MISAL/ TH(3,3,2),UX(3)
      IF(U(1))2,1,1
    1 K=1
      GO TO 3
    2 K=2
    3 IF(U(2)) 5,4,4
    4 L=1
      GO TO 6
    5 L=2
    6 IF(U(3)) 8,7,7
    7 M=1
      GO TO 9
    8 M=2
    9 DO 10 I=1,3
   10 UX(I)=TH(I,1,K)*U(1)+TH(I,2,L)*U(2)+TH(I,3,M)*U(3)
      D(1)=-AL(1)* X(2)*X(6) + X(4)
      D(2)= AL(1) * X(1) *X(6) - X(5)
      D(3)= AL(1)*X(1)*X(5) + X(6)
      D(4)= (AL(1) *(ZI(2)-ZI(3))* X(5)* X(6) +UX(1) + T(1))/ZI(1)
      D(5) = (AL(1) *(ZI(3) -ZI(1))*X(4)*X(6)+UX(2) +T(2))/ZI(2)
      D(6)= (AL(1)* (ZI(1)-ZI(2))* X(4) * X(5) +UX(3) + T(3))/ZI(3)
      RETURN
      END
```

```
      SUBROUTINE TOR(RA,RB,SPIR,SROR,CPIR,CROR)
      DIMENSION  X(6),UF(3),T(3),ZI(3)
      DIMENSION RA(3,3),RB(3,3),RC(3,3),RD(3,3),G(3,3),XI(3,3)
      COMMON  X,UF,T,ZI
      FN=20.
      FL=13./4.
      GM=EL/225.
      FG2=4.*GM*GM
      TGL=2.*GM*EL
      S=9.4E-8
      SP=SINF(X(3))
      SY=SINF(X(2))
      SR=SINF(X(1))
      CP=COSF(X(3))
      CY=COSF(X(2))
      CR=COSF(X(1))
      SYBS=SROR*SP*CY+CROR*SY
      Z=SYBS*SYBS
   31 CYBS=SQRTF(1.-Z)
      SRBS=(SROR*(CP*CR-SP*SR*SY)+CROR*SR*CY)/CYBS
      SPBS=(SPIR*CP*CY+CPIR*CROR*SP*CY-CPIR*SROR*SY)/CYBS
      Z=SRBS*SRBS
      YBS=ATANF(SYBS/CYBS)
      CRBS=SQRTF(1.-SRBS*SRBS)
      CPBS=SQRTF(1.-SPBS*SPBS)
      RBS=ATANF(SRBS/CRBS)
      PBS=ATANF(SPBS/CPBS)
      CALL STUP(PBS,RBS,YBS,RC)
C WE NEED RC TRANSPOSE.
       DO 51 NI=1,2
      IP=NI+1
      DO 51 NJ=IP,3
      TM=RC(NI,NJ)
      RC(NI,NJ)=RC(NJ,NI)
      GO TO 51
   51 RC(NJ,NI)=TM
      DO 52 NI=1,3
      DO 52 NJ=1,3
      RD(NI,NJ)=0.
      DO52 NK=1,3
   52 RD(NI,NJ)=RD(NI,NJ)+RA(NI,NK)*RC(NK,NJ)
      DO53 NI=1,3
      DO 53 NJ=1,3
      G(NI,NJ)=0.
      DO53 NK=1,3
   53 G(NI,NJ)=G(NI,NJ)+RD(NI,NK)*RB(NK,NJ)
C THIS COMPLETES THE G MATRIX.
      IF(G(1,2) )1,99,1
C WE ASSUME ATAN(X) GUVES RABGE -90 TO +90 DEGREES
   99 COR=0.
      IF(G(3,2))60,62,61
   61 SIR=1.
      GO TO 20
```

```
60 SIB=-1.
   GO TO 20
62 SIB=0.
   GO TO 20
 1 IF(G(3,2)) 2,5,2
 5 IF(G(1,2))95,5,24
95 SIB=0.
   COB=-1.
   GO TO 20
24 SIB=0.
   COB=1.
   GO TO 20
 2 BETA=ATANF(G(3,2) /G(1,2) )
   IF(BETA)85,85,89
89 IF(G(3,2))83,82,82
82 SIB=SINF(BETA)
   COB=COSF(BETA)
   GO TO 20
83 BETA=BETA+3.1415927
   GO TO 82
85 IF(G(3,2)) 82,82,83
20 C2D=G(2,2)*G(2,2)
   TAND=SQRTF(1.-C2D)/(-G(2,2)    )
   TAN2=TAND*TAND
   XB=1./FG2/TAN2
   IF(225.-XB)6,6,7
 6 TAA=0.
   GO TO 63
 7 TAA=S*C2D*SQRTF(225.-XB)*(TAN2*EL*EL*.53333-50. +.03333/GM/GM/
  1TAN2)
63 CONTINUE
   CD=-G(2,2)
   SD=SQRTF(1.-C2D)
   DT=ATANF(TAND)
   X0=-15.
   FIN=0.
   IF (DT)8,9,9
 8 DT=DT+3.14159
 9 IF(G(2,2)) 10,11,11
11 TR=1.
   IF(1.97965-DT)12,12,13
12 X1=15.
   GO TO 14
13 X1=-.5/GM/TAND
14 H=(X1-X0)/EN
   DO 16 I=1,20,2
   AI=I
   F=X0+AI*H
   ASSIGN 17 TO M
   GO TO 30
17 FIN=FIN+4.*F
   F=F+H
   ASSIGN 18 TO M
   GO TO 30
```

```
18 FIN=FIN+2.*F
16 CONTINUE
   FIN=H*(FIN-F)/3.
   TR= 4.*S*FIN*TR
   GO TO 26
10 IF(DT-1.16194)15,15,19
15 TR=-1.
   GO TO 12
19 X1=-.5/GM/TAND
   X01=X1
   X11=15.
   FIN=0.
   H=(X1-X0)/EN
   DO 41 I=1,20,2
   AI=I
   F=X0+AI*H
   ASSIGN 42 TO M
   GO TO 30
42 FIN=FIN+4.*F
   F=F+H
   ASSIGN 43 TO M
   GO TO 30
43 FIN=FIN+2.*F
41 CONTINUE
   FIN=H*(FIN-F)/3.
   H=(X11-X01)/EN
   F=X01
   ASSIGN 21 TO M
   GO TO 40
21 GIN=F
   DO 44 I=1,20,2
   AI=I
   F=X01+AI*H
   ASSIGN 22 TO M
   GO TO 40
22 GIN=GIN+4.*F
   F=E+H
   ASSIGN 23 TO M
   GO TO 40
30 F=(2.*GM*E*SD+CD)**2
   X2=E*E
   XB=SQRTF(1./FG2+X2)
   XC=SQRTF(225.-X2)
   XD=ATANF(XC/XB)
   XA=(FG2/2.*E*X2-E*(TGL-1.))/(2.*GM)/SQRTF(1.+FG2*X2)*XD
   F=F*(XA+E/2.*(XC-XB*XD))
   GO TO M
40 X2=E*E
   XC=SQRTF(225.-X2)
   XB=SQRTF(1./FG2+X2)
   XD=ATANF(XC/XB)
   IF(15.-(E+1./GM/TAND))70,70,71
70 XF=0.
   GO TO 72
```

```
71 CONTINUE
   XF=SQRTF(225.-(E+1./GM/TAND)**2)/XB
72 CONTINUE
   XE=ATANF(XF)
   XA=(FG2/2.*E*X2-E*(TGL-1.))/(2.*GM)/SQRTF(1.+FG2*X2)*(XD-XE)
   F=(2.*GM*E*SD+CD)**2
   F=F*(XA+E/2.*(XC-XF*XB)*(XD-XE))
   GO TO M
23 GIN=GIN+2.*F
44 CONTINUE
   GIN=H*(GIN-F)/3.
   TR= 4.*S*(FIN-GIN)
26 TAB=.25*TAA+.75*TR
   T(1)=-SIB*TAB
   T(2)=0.
   T(3)=COB*TAB
   SP=SPBS
   SY=SYBS
   SR=SRBS
   CY=CYBS
   CP=SQRTF(1.-SP*SP)
   CR=SQRTF(1.-SR*SR)
   DO 27 NA=1,3
   DO 27 NB=1,3
27 G(NB,NA)=RA(NA,NB)
   DO 28 L=1,3
28 XI(L,L)=G(L,1)**2*ZI(1)+G(L,2)**2*ZI(2)+G(L,3)**2*ZI(3)
   XI(1,2)=G(1,1)*G(2,1)*ZI(1)+G(1,2)*G(2,2)*ZI(2)+G(1,3)*G(2,3)*
  1ZI(3)
   XI(1,3)=G(1,1)*G(3,1)*ZI(1)+G(1,2)*G(3,2)*ZI(2)+G(1,3)*G(3,3)*
  1ZI(3)
   XI(2,1)=XI(1,2)
   XI(3,1)=XI(1,3)
   XI(2,3)=G(2,1)*G(3,1)*ZI(1)+G(2,2)*G(3,2)*ZI(2)+G(2,3)*G(3,3)*
  1ZI(3)
   XI(3,2)=XI(2,3)
   A1=SP*CY
   A2=-SP*SR*SY+CP*CR
   A3=-SP*CR*SY-CP*SR
   BA=XI(1,1)*A1+XI(1,2)*A2+XI(1,3)*A3
   BB=XI(2,1)*A1+XI(2,2)*A2+XI(2,3)*A3
   BC=XI(3,1)*A1+XI(3,2)*A2*XI(3,3)*A3
   T(1)=15.9E-9*(-BB*A3+BC*A2)    +T(1)
   T(2)=15.9E-9*(BA*A3-BC*A1)    +T(2)
   T(3)=15.9*(BB*A1-BA*A2)*1.E-9 +T(3)
   RETURN
   END
```

```
      SUBROUTINE RK
      DIMENSION X(6),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(6)
      DIMENSION R(6),SA(6), SB(6),SC(6),SD(6)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,Z,S,D,NA,TI
      DO 2 N=1,6
    2 R(N)=X(N)
      CALL DER
      DO 3 N=1,6
    3 SA(N)=D(N)*DT
      DO 4 N=1,6
    4 X(N)=R(N)+SA(N)/2.
      TI=TI+DT/2.
      CALL DER
      DO 5 N=1,6
    5 SB(N)=D(N)*DT
      DO 6 N=1,6
    6 X(N)=R(N)+SB(N)/2.
      CALL DER
      DO 7 N=1,6
    7 SC(N)=D(N)*DT
      TI=TI+DT/2.
      DO 8 N=1,6
    8 X(N)=R(N)+SC(N)
      CALL DER
      DO 9 N=1,6
      SD(N)=D(N)*DT
    9 X(N)=R(N)+(SA(N)+SD(N))/6.+(SB(N)+SC(N))/3.
      RETURN
      END
```

```
C                             ELASTICITY MODEL
C                                                        DATE 11 MAR 68
C
C
C
C
C 7094     7094     7094     7094
C1 NA,NB,PA,PB,PC                              2I5,5F10.5
C2 ZI(1-3), ZM(1-3), AL(1-3),                  9F8.5
C3 ZL(1,1), (1,2).......(3,3)                  9F8.5
C4 AX,BX,PIB,ROB,YB,PDBS,RDBS,YDBS             9F8.5
C5 YO,AT,ONG,XLAT,XLON,DT,AE                   9F8.5
C6 T(1) T(2) T(3)                                             9F8.5
C  XF PREVENTS JAGS COMING IN; AE INCREASES ZERO BAND;AX,BX LEAD FACTORS
C  Z=0 USES TOR SUBROUTINE
C  (3-6) (10-16)  20,23,27,30,31,35, (100-116)
      DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
      DIMENSION RA(3,3),RB(3,3),RC(3,3),RD(3,3),ETA(3),C(3),NF(3)
      DIMENSION XX(10)
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX,XX
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,RBS,PIB,YB,ROB
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,TO
   13 TI=0.
   32 CONTINUE
      I=0.
      T4=6.610808
      ZK=T4
      NC=2
      RAD=.017453
      AC=1.
      DO 1 I=1,3
      NF(I)=1.
      C(I)=1.
    1 CONTINUE
      READ (5,100)NA,NB,PA,PB,PC,Z
      READ(5,101) (ZI(I),I=1,3),(ZM(I),I=1,3),(AL(I),I=1,3)
      READ (5,101) ((ZL(I,J),J=1,3),I=1,3)
      READ (5,101) AX,BX,XF,PIB,ROB,YB,RDBS,YDBS,PDBS
      READ(5,101) YO,AT,ONG,XLAT,XLON,DT,AE
      READ (5,101) T, (D(I),I=11,14)
C  D(11)-D(14) LENGTH, MASS, SPRING CONSTANT, DAMPING COEF.
      DO 35 I=1,3
      ZIM(I)  =ZI(I)/ZM(I)
      U(I)=ZM(I)
   35 CONTINUE
      IF(ZL(1,2).EQ.0..AND.ZL(1,3).EQ.0.) NC=1
   23 T7=AT
      T8=ONG
      T9=XLAT
      T10=XLON
      ONG=ONG-AL(3)
      XLON=XLON-AL(3)
```

```
  3 AT=AT*RAD
    ONG=ONG*RAD
    YO=YO*RAD
    SAT=SIN(AT)
    SON=SIN(ONG)
    CAT=COS(AT)
    CON=COS(ONG)
    T1=SIN(YO)
    T2=COS(YO)
    T3=T4-CAT*CON
    P2=ATAN(-CAT*SON/T3)
    SP2=SIN(P2)
    R2=ATAN(SAT/(COS(P2)*T3-CAT*SON*SP2) )
    T5=T1*COS(R2)
    YBS=ATAN(T5/SQRT(1.-T5**2))
    T5=SIN(R2)/COS(YBS)
    RBS=ATAN(T5/SQRT(1.-T5**2))
    T5=(SP2*T2-T1*SIN(R2)*COS(P2))/COS(YBS)
    PBS=ATAN(T5/SQRT(1.-T5**2))
    WRITE (6,105)
    WRITE (6,106)((ZL(I,J), J=1,3),I=1,3)
    WRITE (6,107)
    TEM=YO/RAD
    WRITE (6,106)TEM    ,T7,T8,T9,T10,RBS,YBS,PBS,DT
    WRITE (6,108)
    WRITE (6,106) (ZI(I),I=1,3), (ZM(I),I=1,3), (AL(I),I=1,3)
    WRITE (6,109)
    WRITE (6,106)AX,BX,XF,AE
    CALL INIT
    DO 37 I=1,3
    XX(I)=X(I)
    XX(I+3)=0.
    X(I+6)=0.
 37 XX(I+6)=0.
    XX(10)=0.
    X(10)=0.
    CALL STUP(0.,.35692,0.,RA)
    CALL STUP(AL(3),0.,0.,RB)
    DO 4 I=1,3
    DO 4 J=1,3
    RD(I,J)=0.
    DO 4 K=1,3
  4 RD(I,J)=RD(I,J)+RA(I,K)*RB(K,J)
    CALL STUP(PIR,ROR,0.,RA)
    DO 5 I=1,3
    DO 5 J=1,3
    RB(J,I)=0.
    DO 5 K=1,3
  5 RB(J,I)=RB(J,I)+RD(I,K)*RA(K,J)
C 5 TAKES THE TRASPOSE AS IT COMPUTES THE PRODUCT
    CALL STUP (PIB,ROB,YB,RA)
    SPIR=SIN(PIR)
    SROR=SIN(ROR)
    CPIR=COS(PIR)
```

```
      CROR=COS(ROR)
      IF(7.NE.0.)             GO TO 16
    6 CALL TOR (RA.RB.SPIR.SROR.CPIR.CROR)
   16 CONTINUE
      T4=T7
      T5=T8
      T7=T(1)*PA/ZI(1)
      T8=T(2)*PA/ZI(2)
      T9=T(3)*PA/ZI(3)
      WRITE (6,110)
      WRITE (6,106)T(1),T(2),T(3),T7,T8,T9
      IF (NA)15,14,15
   15 CALL OUT
      GO TO 10
   14 WRITE (6,112)
      T1=X(1)*PB
      T2=X(2)*PB
      T3=X(3)*PB
      T4=X(4)*PC
      T5=X(5)*PC
      T6=X(6)*PC
      WRITE (6,106) TI,T1,T2,T3,T4,T5,T6
      WRITE (6,113)
      PAUSE 11
   10 IF(7.EQ.0.) CALL TOR(RA,RB,SPIR,SROR,CPIR,CROR)
      DO 38 JB=1,10
   38 XX(JB)=X(JB)
      CALL REG
      DO 11 J=1,NA
      CALL RK
      L=L+1
      ASSIGN 31 TO NG
      GO TO 30
   31 CONTINUE
      IF(NF(1)+NF(2)+NF(3)-6) 27,20,27
   27 IF(L.LT.NB)GO TO 11
      CALL OUT
      L=0
   11 CONTINUE
      IF (SENSE SWITCH 1) 41,40
   41 READ (101,100) NB
      GO TO 10
   40 IF(SENSE SWITCH 2) 20,10
   20 CALL OUT
      PAUSE 16
      GO TO 13
   30 CONTINUE
C FOLLOWING COMPUTES GROUND TRACK, WHERE T4  IS LAT AND T5  IS LONG
      S1=SIN(X(3))
      S2=SIN(X(2))
      S3=SIN(X(1))
      C1=COS(X(3))
      C2=COS(X(2))
      C3=COS(X(1))
```

```
      S7=SRØR*S1*C2+CR[illegible]
      C7=SQRT(1.-S7*S7)
      S8=(SRØR*(C1*C3-S1*S2*S3)+CRØR*S3*C2)/C7
      C8=SQRT(1.-S8*S8)
      S6=(SPIR*C1*C2+CPIR*CRØR*S1*C2-CPIR*SRØR*S2)/C7
      C6=SQRT(1.-S6*S6)
      F2=ATAN(S7/C7/C8)
      T2=SIN(F2)
      D2=CØS(F2)
      T1=S6*C7*D2+T2*(C6*S8+S6*C8*S7)
      D1=SQRT(1.-T1*T1)
      T3=S8*C7
      D3=SQRT(1.-T3*T3)
      Y1=T3*(ZK*D3*D1-SQRT(1.-ZK*ZK*(1.-D3*D3*D1*D1)))
      Z1=SQRT(1.-Y1*Y1)
      T4 =ATAN(Y1/Z1)/RAD
      A1=SQRT(Z1*Z1-ZK*ZK*T1*T1)
      T5 =ATAN(T1*(A1-ZK*D1)/(A1+ZK*T1*T1)/D1)/RAD +AL(3)
      GØ TØ NG,(32,31)
100   FØRMAT (2I5,5F10.5)
101   FØRMAT  (9F8.5)
102   FØRMAT(46H TYPE PØS YO,AT,ØNG,TARGET LAT AND LØNG 9F5.3        )
103   FØRMAT  (9F5.3)
106   FØRMAT  (1P9E14.5)
105   FØRMAT  (1H0,2X 10HLAMDA(1,1),4X10HLAMDA(1,2),4X10HLAMDA(1,3),4X10H
     1LAMDA(2,1),4X10HLAMDA(2,2),4X10HLAMDA(2,3),4X10HLAMDA(3,1),4X10HLA
     2MDA(3,2),4X10HLAMDA(3,3) )
107   FØRMAT  (1H0,4X2HY0,12X2HAI,11X3HØNG,11X3HLAT,11X3HLØN,11X3HRBS,11X
     23HYBS,11X3HPBS,12X2HDT )
108   FØRMAT  (1H0,4X8HMØMENT 1,6X8HMØMENT 2,6X8HMØMENT 3,8X4HZM 1,10X4HZ
     2M 2,10X4HZM 3,8X7HALPHA 1,7X7HALPHA 2,7X7HALPHA 3   )
109   FØRMAT  (1H0,5X2HAX,12X2HBX,12X2HXF,12X2HAE   )
110   FØRMAT  (1H0,4X8HTØRQUE 1,6X8HTØRQUE 2,6X8HTØRQUE 3,15X14HSCALED TØ
     1RQUES   )
111   FØRMAT  (1H0, 5X 4HTIME, 10X4HX(1), 10X4HX(2), 10X 4HX(3), 10X
     14HX(4), 10X 4HX(5), 10X 4HX(6) )
112   FØRMAT  (1H0,4X4HTIME, 19X 26HSCALED  VALUES  ØF X(1-6)          )
113   FØRMAT  (1H1 )
114   FØRMAT  (10HSET ANALØG   )
115   FØRMAT  (20X 4HD(1), 10X 4HD(2), 10X 4HD(3), 10X 4HD(4),
     110X 4HD(5), 10X 4HD(6) )
116   FØRMAT  (20X 4HU(1), 10X 4HU(2), 10X 4HU(3), 10X 3HX1S, 11X 3HX4S,
     110X 1HS   )
      END
```

```
      SUBROUTINE REG
C 7094     7094     7094     7094
C (1-17).60,(63-66),68,70,73,(101-129),134,135,138
C ,200-203),(207-217),(222-225),345
      DIMENSIONXX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),X0(10),ZL(3,3)
      DIMENSION AL(3),NF(3),ETA(3),U0(3),C(3),TIM(3),TS(3),D(20),XA(10)4
      COMMONXA,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
      COMMON /CON/C,ETA,NF,XF,AE,AX,BX,XX
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
      DO 115 I=1,3
  200 U0(I) = U(I)
      GO TO (201,202,203),I
  201 B=(T(1)+(ZI(2)-ZI(3))*XX(5)*XX(6)*AL(1))/ZM(1)
      T1=XX(4)-XX(2)*XX(6)*AL(1)
      GO TO 11
  203 B= ( T(3)+(ZI(1)-ZI(2))*XX(4)*XX(5)*AL(1))/ZM(3)
      T1=XX(6)+XX(1)*XX(5)*AL(1)
   11 X=XX(I)*ZIM(I)*SIGN(1.,-B)
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
      GO TO 18
  202 B=(T(2)+(ZI(3)-ZI(1))*XX(6)*XX(4)*AL(1))/ZM(2)
      T1=XX(5)-XX(1)*XX(6)*AL(1)
      X=XX(I)*ZIM(I)*SIGN(1., B)
      IF (B.EQ.0.) X=-X
      Y=XX(I+3)*ZIM(I)*SIGN(1.,-B)
   18 IF(X*Y)2,2,1
    1 U(I)=SIGN(ZM(I),-X)*SIGN(1.,-B)
      ETA(I)=-1.
      C(I)=1.
      GO TO 10
    2 T2=200.*DT
      IF(ZIM(I)*ABS(X0(I+3)-XX(I+3))-T2)12,12,13
   12 IF(ABS(X0(I)-XX(I))-T2*ABS(XX(I+3)))14,14,13
   13 ETA(I)=-1.
      C(I)=1.
      GO TO 9
   14 C(I)=C(I)*U(I)
    9 CONTINUE
    8 GO TO(209,345),NC
C NC=1 TIME OPTIMAL.208 USES MIN.209 USES MAX.
  345 IF(C(I))208,209,208
  208 IF(ABS(XX(I+3))-ABS(T1))212,212,210
  209 IF(ABS(XX(I+3))-ABS(T1))210,212,212
  210 Y=SIGN(T1*ZIM(I),Y)
  212 IF(ETA(I))207,207,213
  213 Y=Y*XF
  207 T3=Y*Y/2.
      P2=X-T3/(B-1.)
      P4=X-T3/(B+1.)
      GO TO (63,60),NC
   63 IF(X)64,66,65
   64 IF(P2)17,16,15
   65 IF(P4)17,17,16
```

```
66    IF(Y) 17,15,16
60    IF(X)68,66,70
68    IF(P2)73,16,16
70    IF(P4)17,17
73    CONTINUE
      T4=ABS(ZL(I,2)*B/ZL(I,3))
      R=SQRT(2.*T4
      S=SQRT(ZL(I,1)/ZL(I,3)+T4)
      IF(X)3,66,5
3     IF(Y-S) 19,4,4
19    IF(B) 26,25,26
25    QX=-10000.
      GO TO 20
26    BA=-ABS(B)
      QX=(BA*S*S-2.*P*S+R*R)/(2.*BA*(BA-1.))
20    IF(X-QX)17,17,21
21    G=-ZL(I,1)-ZL(I,3)*Y**2+ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA-1.)*X   )
      IF(( BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA-1.)*X)17,15,15
4     IF(Y-S-R-AE)15,15,16
5     IF(Y-(R-S)) 6,15,22
22    IF(B) 27,28,27
28    QX=10000.
      GO TO 23
27    BA=-ABS(B)
      QX=(BA*(R-S)**2-2.*R*S+R*R)/(2.*BA*(BA+1.))
23    IF(X-QX)24,16,16
24    G=-ZL(I,1)-ZL(I,3)*Y**2-ZL(I,2)*BA
      Z=-2.*ZL(I,2)+ZL(I,3)*(Y**2-2.*(BA+1.)*X   )
      IF((-BA*Z*Z+2.*G*Z          +G*G)*Y*Y-2.*G*G*(BA+1.)*X)16,15,15
6     IF(Y-(-S-AE))17,15,15
15    U(I)=0.
      GO TO 7
16    U(I)=SIGN(ZM(I),B)
      IF(B.EQ.0.) U(I)=-U(I)
      GO TO 7
17    U(I)=SIGN(ZM(I),-B)
7     IF( I-2)31,32,31
32    FTA(I)=U(I)*X
      GO TO 10
31    FTA(I)=U(I)*XX(I)
10    XO(I)=XX(I)
      XO(I+3)=XX(I+3)
C 123 BEGINS TIME SYNCH
123   X=XX(I)*ZIM(I)
      IF (I.EQ.2) X=-X
      Y=XX(I+3)*ZIM(I)
      T6=U(I)
121   T5=Y*Y/2.
      P2=X-T5
      P4=X+T5
      GO TO(117,101),NC
117   IF(Y)107,107,109
107   IF(P2)110,110,111
```

```
111 X=-X
    Y=-Y
    T6=-U(I)
    GO TO 121
109 IF(P4) 112,111,111
110 TS(I)=-1.
    GO TO 113
112 TS(I)=ABS(Y/2.)-X/Y
    IF(T6*Y.LE. 0.) TS(I)=-1.
113 TIM(I)=2.*SQRT(ABS(P2))-Y
    GO TO 211
101 PS=X+(.5+2.*ZL(I,2)/(ZL(I,1)-ZL(I,3)*Y**2))*Y**2
    P=Y*ABS(Y)+2.*X
    S=SQRT(ZL(I,1)/ZL(I,3))
    IF(P) 87,79,111
 87 IF(Y-S)74,89,80
 89 TS(I)=-1.
    TIM(I)=Y-P4/S
    GO TO 211
 74 IF(PS)75,75,76
 75 T9=Y**2-2.*X
    T8=2.*ZL(I,1)+4.*ZL(I,2)
    T0=SQRT((T8+ZL(I,3)*T9)**2-8.*ZL(I,1)*ZL(I,3)*T9)
    T0=SQRT((T8+ZL(I,3)*T9-T0)/4./ZL(I,3))
C T0 IS OMEGA S1
    TIM(I)=T0-P2/T0-Y
C REGION I
    IF(Y)77,77,78
 77 TS(I)=-1.
    GO TO 211
 78 TS(I)=Y/2.-X/Y
    IF(T6*Y.LE. 0.) TS(I)=-1.
    GO TO 211
 76 IF (Y) 75,75,79
 79 TIM(I)=          Y/2.-X/Y
    GO TO 77
 84 X=-X
    Y=-Y
    GO TO 121
211 IF(ABS(XX(I))-.0020) 222,223,223
222 IF(ABS(XX(I+3))-.00005) 224,223,223
224 U(I)=0.
    TIM(I)=0.
    NF(I)=2
    GO TO 115
223 K=NF(I)
    GO TO (115,216),K
216 IF(ABS(XX(I))-.005)226 ,225,225
226 U(I)=0.
    TIM(I)=0.
    GO T O 115
225 NF(I)=1
115 CONTINUE
127 TMAX=AMAX1(TIM(1),TIM(2),TIM(3))*AX
```

```
129 DO 124 J=1,3
    K=NF(J)
128 GO TO (138,124),K
138 IF(TS(J))124,125,125
125 IF(TMAX-TS(J))124,124,134
134 IF(TMAX-TS(J)-BX)135,126,126
135 IF(U0(J))124,126,124
126 U(J)=0.
124 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE DER
      DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /TEMP/T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
      EQUIVALENCE (XL,D(11)),(XM,D(12)),(ALF,D(13)),(BET,D(14))
C   THIS PROGRAM IS ONLY TO CHECK ELASTIC EFFECTS
C   XL IS BOOM LENGTH. XM IS MASS (SLUGS)
C   ALF IS SPRING CONSTANT8 BETA IS DAMPING
      ZI(1)=ZI(1)-XM*XL**2
      ZI(3)=ZI(3)-XM*XL**2
      T7=X(4)*X(7)+X(5)*XL       +X(6)*X(8)
      T8=X(4)**2+X(5)**2+X(6)**2
      T9= XM*(-2.*(X(6)*X(9)-X(4)*X(10))-X(5)*T7+XL*T8-(D(6)*X(7)-D(4)D
     1*X(8)))
      T1=XL*(ALF*X(8)+BET*X(10))-X(8)*T9
      T2=BET*(X(8)*X(9)-X(7)*X(10))
      T3=-XL*(ALF*X(7)+BET*X(9))+X(7)*T9
      D(1)=-AL(1)* X(2)*X(6) + X(4)
      D(2)= AL(1) * X(1) *X(6) - X(5)
      D(3)= AL(1)*X(1)*X(5) + X(6)
      D(4)= (AL(1) *(ZI(2)-ZI(3))* X(5)* X(6) + U(1) + T(1))/ZI(1)
      D(5) = (AL(1) *(ZI(3) -ZI(1))*X(4)*X(6) +U(2) +T(2))/ZI(2)
      D(6)= (AL(1)* (ZI(1)-ZI(2))* X(4) * X(5) + U(3) + T(3))/ZI(3)
      D(4)=D(4)+T1/ZI(1)
      D(5)=D(5)+T2/ZI(2)
      D(6)=D(6)+T3/ZI(3)
      D(7)=X(9)
      D(8)=X(10)
      D(9)=-(ALF*X(7)+BET*X(9))/XM-X(4)*T7+X(7)*T8-(D(5)*X(8)-D(6)*XL)
     1-2.*X(5)*X(10)
      D(10)=-(ALF*X(8)+BET*X(10))/XM-X(6)*T7+X(8)*T8-(D(4)*XL-D(5)*X(7))
     1+2.*X(5)*X(9)
      ZI(1)=ZI(1)+XM*XL**2
      ZI(3)=ZI(3)+XM*XL**2
      RETURN
      END
```

```
      SUBROUTINE RK
      DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
      DIMENSION R(10),SA(10),SB(10),SC(10),SD(10)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,Z,S,D,NA,TI
      DO 2 N=1,10
    2 R(N)=X(N)
      CALL DER
      DO 3 N=1,10
    3 SA(N)=D(N)*DT
      DO 4 N=1,10
    4 X(N)=R(N)+SA(N)/2.
      TI=TI+DT/2.
      CALL DER
      DO 5 N=1,10
    5 SB(N)=D(N)*DT
      DO 6 N=1,10
    6 X(N)=R(N)+SB(N)/2.
      CALL DER
      DO 7 N=1,10
    7 SC(N)=D(N)*DT
      TI=TI+DT/2.
      DO 8 N=1,10
    8 X(N)=R(N)+SC(N)
      CALL DER
      DO 9 N=1,10
      SD(N)=D(N)*DT
    9 X(N)=R(N)+(SA(N)+SD(N))/6.+(SB(N)+SC(N))/3.
      RETURN
      END
```

```
SUBROUTINE DTBA
DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DI,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
AC=AC
RETURN
END
```

```
SUBROUTINE ATOD
DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI/HYB/PA,PB,PC,AC
AC=AC
RETURN
END
```

```
      SUBROUTINE OUT                                                    70
      DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /TEMP/ T1,T2,T3,T4,T5,T6,T7,T8,T9,T0
       WRITE (6,112) TI,X
      WRITE (6,113) D
      WRITE (6,113) U,T4,T5
  112 FORMAT (1H0, F6.2, 1P10E12.3)
  113 FORMAT (7X, 1P10E12.3)
      RETURN
      END
```

```
      SUBROUTINE STUP (CP,CR,CY,CA)
      DIMENSION CA(3,3)
      SINR=SIN(CR)
      SINR=SIN(CR          )
      SINP=SIN(CP)
      SINY=SIN(Y)
      COSR=COS(CR)
      COSP=COS(P)
      COSY=COS(Y)
C WE COMPUTE THE MATRIX I+B
      CA(1,1)=COSP*COSY
                                  CA(1,2)=-COSP*SINR*SINY-SINP*COSR
      CA(1,3)=-COSP*COSR*SINY+SINP*SINR
                                                  CA(2,1)=SINP*COSY
      CA(2,2)=-SINP*SINR*SINY+COSP*COSR
      CA(2,3)=-SINP*COSR*SINY-COSP*SINR
                                                  CA(3,1)=SINY
      CA(3,2)=SINR*COSY
                                  CA(3,3)=COSR*COSY
      RETURN
      END
```

```
SUBROUTINE TOR(RA,RB,SPIR,SROR,CPIR,CROR)
DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
DIMENSION RA(3,3),RB(3,3),PC(3,3),PD(3,3),G(3,3),XI(3,3)
COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
T(1)=1.E-7
T(2)=1.E-7
T(3)=1.E-7
RETURN
END
```

```
      SUBROUTINE    INIT
      DIMENSIONX(10),U(3),T(3),ZI(3),ZM(3),ZIM(3),AL(3),ZL(3,3),D(20)
      DIMENSION A(3,3),B(3,3),C(3,3),H(3),W(3)
      COMMON X,U,T,ZI,ZM,ZIM,AL,ZL,DT,NC,R,S,D,NA,TI
      COMMON /POSIN/XLAT,XLON,PIR,ROR,PBS,YBS,RBS,PIB,YB,ROB
      XLON = XLON *0.017453
      XLAT =XLAT *0.017453
      SILAT=SIN(XLAT)
      SILON=SIN(XLON)
      COLAT=COS(XLAT)
      COLON=COS(XLON)
      CONST=6.610808
      PITCH =-ATAN ((+SILON * COLAT)/(CONST - COLON * COLAT))
      COPI=COS(PITCH)
      SINPI=SIN(PITCH)
      ROLL = ATAN  (SILAT/((CONST -COLON*COLAT)* COPI -SILON*COLAT *
     1SINPI))
      PIR= PITCH
      ROR =ROLL
      CRBS=COS(RBS)
      SRBS=SIN(RBS)
      CYBS=COS(YBS)
      SYBS=SIN(YBS)
      CPIB=COS(PIB)
      CYB=COS(YB)
      CROB=COS(ROB)
      SYB=SIN(YB)
      SROB=SIN(ROB)
      SPIB=SIN(PIB)
      B(1,1) = CPIB * CYB
      B(1,2) = -CPIB * SROB * SYB - SPIB * CROB
      B(1,3) = -CPIB * CROB * SYB + SPIB * SROB
      B(2,1) = SPIB * CYB
      B(2,2) = -SPIB * SROB * SYB + CPIB * CROB
      B(2,3) = -SPIB * CROB * SYB - CPIB * SROB
      B(3,1) = SYB
      B(3,2) = SROB * CYB
      B(3,3)=CROB*CYB
      A(1,1) = 1.
      A(1,2)=0.
      A(1,3) = SYBS
      A(2,1) = 0.
      A(2,2) = -CRBS
      A(2,3) = CYBS * SRBS
      A(3,1) = 0.
      A(3,2) = SRBS
      A(3,3) = CYBS * CRBS
      DO 17 K = 1,3
      DO 17 I = 1,3
      C(I,K) = 0.
      DO 17 J = 1,3
      C(I,K) = B(I,J) * A(J,K) + C(I,K)
   17 CONTINUE
```

```
      H(1)=RDBS
      H(2)=YDBS
      H(3)=PDBS
      DO 61 I = 1,3
      W(I) = 0.
      DO 62 J = 1,3
      W(I)=W(I)+C(I,J)*H(J)
   62 CONTINUE
      X(I+3) = W(I)
   61 CONTINUE
      SPIR=SIN(PIR)
      SROR=SIN(ROR)
      CPIR=COS(PIR)
      CROR=COS(ROR)
      SPRB=SIN(PIR-PBS)
      CPRB=COS(PIR-PBS)
      G1 = SROR * CYBS * SPRB + CROR * SYBS
   78 FORMAT (F9.5)
      G2 = SQRT  (1.-G1**2)
      X(2) =   ATAN  (G1/G2)
      G3=(-CROR*CYBS*SPRB+SROR*SYBS)/G2
      G4=SQRT (1.-G3*G3)
      X(3)=ATAN (G3/G4)
      G6 =(-SROR * CRBS  * CPRB - SROR* SRBS *SYBS *SPRB + CROR*SRBS *
     1CYBS)/G2
      G7= SQRT  (1.- G6**2)
      X(1)= ATAN  (G6/G7)
      RETURN
      END
```

7.5 *Simulation Model For Analog Computer*

Equations of Motion

The equations of motion for the satellite rotational dynamics are written

$$I_1\dot{\omega}_1 = (I_2-I_3)\omega_2\omega_3 + B_1 + T_1 \tag{7.5.1}$$

$$I_2\dot{\omega}_2 = (I_3-I_1)\omega_1\omega_3 + B_2 + T_2 \tag{7.5.2}$$

$$I_3\dot{\omega}_3 = (I_1-I_2)\omega_1\omega_2 + B_3 + T_2 \tag{7.5.3}$$

where Bi and Ti represent <u>lumped</u> control and disturbance torques respectively.

The Euler rate transformation equations for the rotational sequence θ, $-\Psi$, φ (pitch, -yaw, roll) are written

$$\dot{\varphi} = \omega_1 - \omega_3\Psi \tag{7.5.4}$$

$$\dot{\Psi} = -\omega_2 + \omega_3\varphi \tag{7.5.5}$$

$$\dot{\theta} = \omega_3 + \omega_2\varphi \tag{7.5.6}$$

when small angle approximations (cos x=1, sin x=x) are used for the angles θ, $-\Psi$, and φ.

Lumped Control Torque

For the case of no thrustor misalignment the lumped control torques B_1, B_2, and B_3 are written

$$B_1 = (t_1 \times d_1)\ u_1 \tag{7.5.7}$$

$$B_2 = (t_2 \times d_2)\ u_2 \tag{7.5.8}$$

$$B_3 = (t_3 \times d_3)\ u_3 \tag{7.5.9}$$

where,

$$u_i = \begin{cases} 1 & (+)\ \text{thrust} \\ 0 & \text{no thrust} \\ -1 & (-)\ \text{thrust} \end{cases}$$

t_i = thrust

d_i = moment arm.

When each of the six attitude control jets ($\pm$ pitch, $\pm$ yaw, $\pm$ roll) can be misaligned in either aximuth (α) or elevation (ϵ) the equations for B_i must be rewritten.

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} b_{11}^+ & b_{12}^+ & b_{31}^+ \\ b_{21}^+ & b_{22}^+ & b_{32}^+ \\ b_{31}^+ & b_{32}^+ & b_{33}^+ \end{bmatrix} \begin{bmatrix} c_1 & u_1 \\ c_2 & u_2 \\ c_3 & u_3 \end{bmatrix} + \begin{bmatrix} b_{11}^- & b_{12}^- & b_{31}^- \\ b_{21}^- & b_{22}^- & b_{32}^- \\ b_{31}^- & b_{32}^- & b_{33}^- \end{bmatrix} \begin{bmatrix} c_1 & u_1 \\ c_2 & u_2 \\ c_3 & u_3 \end{bmatrix} \qquad 7.5.10$$

where,

$c_i = t_i \times d_i$

b_{ij}^+ = misalignment coefficients for (+) thrustors

b_{ij}^- = misalignment coefficients for (-) thrustors

Equation (10) can be expanded to

$$B_1 = c_1\ u_1\ (b_{11}^+ \oplus b_{11}^-) + c_2 u_2 (b_{12}^+ \oplus b_{12}^-) + c_3 u_3 (b_{13}^+ \oplus b_{13}^-) \qquad (7.5.11)$$

$$B_2 = c_1\ u_1\ (b_{21}^+ \oplus b_{21}^-) + c_2 u_2 (b_{22}^+ \oplus b_{22}^-) + c_3 u_3 (b_{23}^+ \oplus b_{23}^-) \qquad (7.5.12)$$

$$B_3 = c_1\ u_1\ (b_{31}^+ \oplus b_{31}^-) + c_2 u_2 (b_{32}^+ \oplus b_{32}^-) + c_3 u_3 (b_{33}^+ \oplus b_{33}^-) \qquad (7.5.13)$$

where $(b_{ij}^+ + b_{ij}^-)$ is the misalignment coefficient for $u = +1$ or ($\oplus$) $u = -1$.

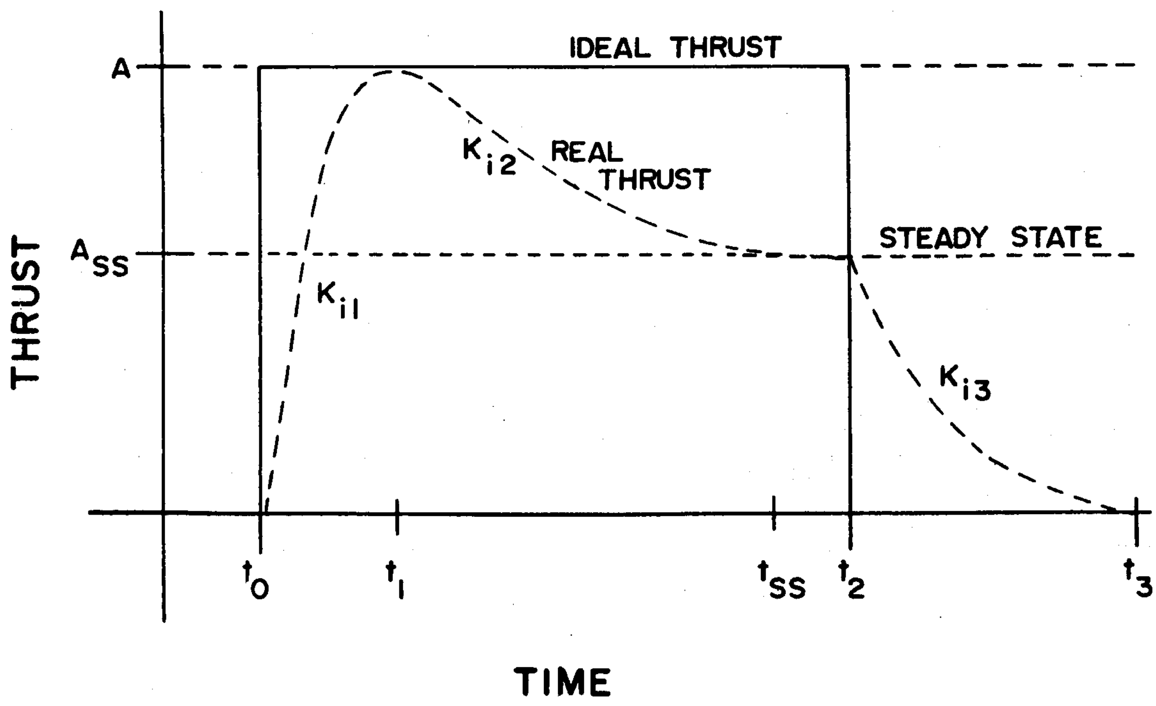
Lumped Disturbance Torque

The lumped disturbance torques T_i represent total disturbance acting on the satellite. In general this is the sum of the gravity gradient torque and solor pressure torques.

Thrustor Real Effects (TRE)

The simulation of the Thrustor Real Effects requires three separate time constants for each of the three axes. The time constants are designated k_{ij} where "i" is the axis. j=1 is the rise time constant; j=2 is the steady state decay time constant; j=3 is the fall time constant.

The diagram below shows a typical curve used for the TRE simulation.



Fuel Consumption

The fuel consumption is determined from the following equation

$$FC = \frac{1}{ISP}\int |u_1 t_1| + |u_2 t_2| + |u_3 t_3|\ dt.$$

This equation can be rewritten and scaled as follows

$$[10FC] = \sum_{i=1}^{3}\left[\frac{100I_i}{ISP \times d_i \times 5 \times 10^4}\right]\int |[5B_i \times 10^3/I_i]|\ dt.$$

Scaled Equations of Motion

$$[5\dot{\omega}_1 \times 10^3] = \frac{.8A_1}{10}[250\omega_2][250\omega_3] + (5B_1 \times 10^3/I_1) + (5T_1 \times 10^3/I_1)$$

$$[5\dot{\omega}_2 \times 10^3] = \frac{.8A_2}{10}[250\omega_1][250\omega_3] + (5B_2 \times 10^3/I_2) + (5T_2 \times 10^3/I_2)$$

$$[5\dot{\omega}_3 \times 10^3] = \frac{.8A_3}{10}[250\omega_1][250\omega_2] + (5B_3 \times 10^3/I_3) + (5T_3 \times 10^3/I_3)$$

$$[250\dot{\phi}] = [250\omega_1] - \frac{1}{5}[250\omega_3][5\Psi]$$

$$[250\dot{\Psi}] = -[250\omega_2] + \frac{1}{5}[250\omega_3][5\phi]$$

$$[250\dot{\theta}] = [250\omega_3] + \frac{1}{5}[250\omega_2][5\phi]$$

where,

$$A_1 = \frac{I_2 - I_3}{I_1}$$

$$A_2 = \frac{I_3 - I_1}{I_2}$$

$$A_3 = \frac{I_1 - I_2}{I_3}$$

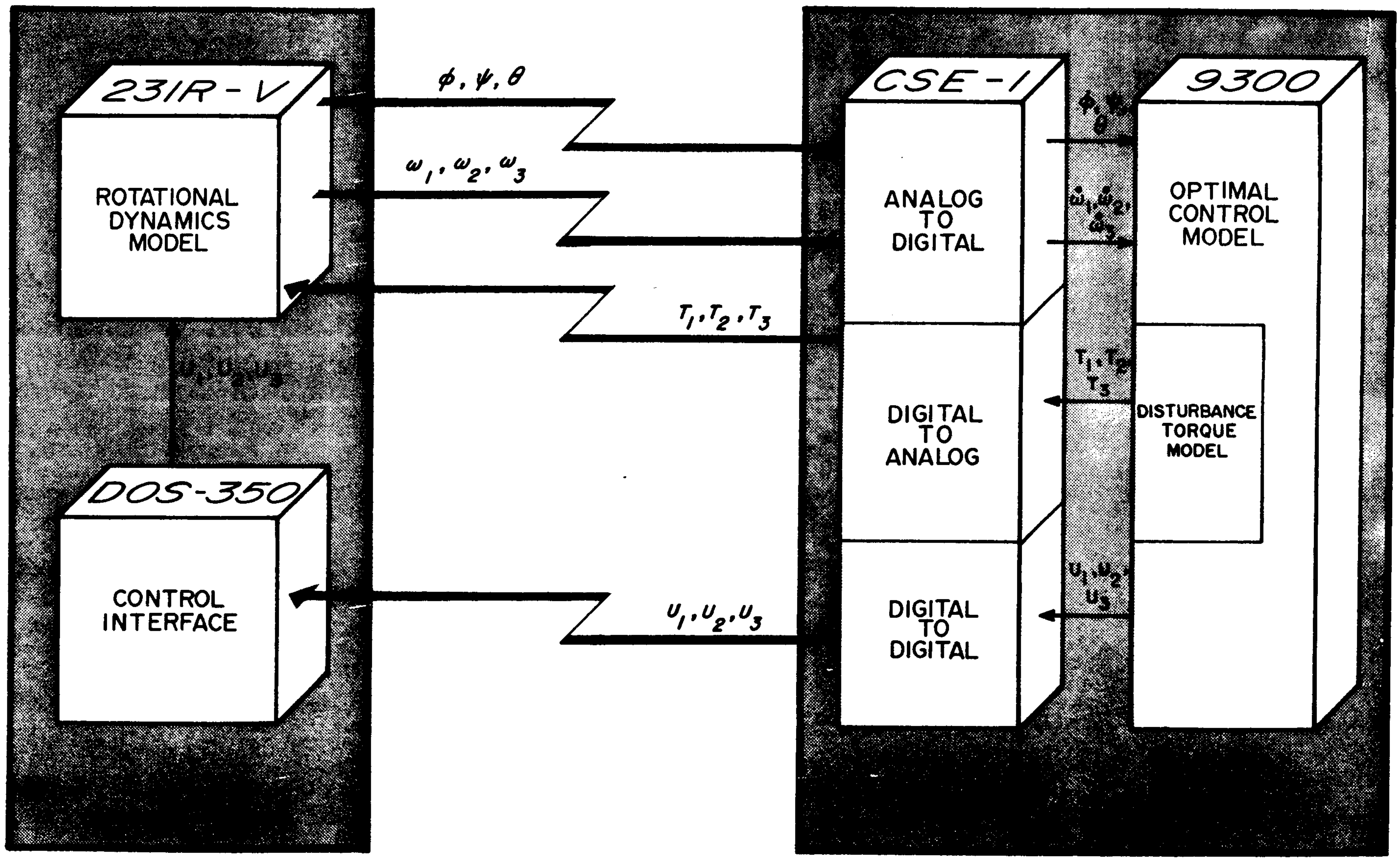
7.6 *Overall Hybrid Simulation*

The objective of the Hybrid Simulation was to gather performance and design data for the optimal attitude slew controller discussed in Parts I thru VI of this report. In order to accomplish this objective a Hybrid Computer system was used to model the Optimal Control Equations, Disturbance Torques, Satellite Rotational Dynamics, and Thrustor Real Effects. Each of these models were tied together as a total hybrid simulation by a Digital/Analog Interface.

The optimal control equations and external disturbance torques (Solar & Gravity Gradient) were programmed on the SDS 9300 digital computer. Position and rate information required in the optimal control equation were generated from the satellite rotational dynamics model programmed on the EAI 231R-V analog computer. The thrustor real effects were incorporated into the analog dynamics model.

The digital interface (CSE-1) and analog interface (DOS-350) made-up the linkage between the analog and digital computers. The digital interface provided A/D conversion, D/A conversion and timing. The control commands to the dynamics model were interfaced through the analog interface.

A block diagram of the simulation hardware and its function is shown on the next page. The symbols in this diagram are defined in Section 7.5.



SIMULATION BLOCK DIAGRAM

7.7 *Simulation Setup Procedures*

a. Load digital program using standard 9300 procedures. When loading is complete computer will PAUSE.

b. Place the data deck in the card reader and clear the PAUSE. Initial conditions for both the analog and digital computers will be output on the line printer and computer will PAUSE.

c. Set analog potentiometers for the dynamics constants. (ref. 7.8)

d. Set interval timers for the desired sampling time.

e. Set up strip chart and X-Y recorders.

f. Place analog computer in IC mode.

g. Set analog potentiometers and switches in accordance with the printout. (ref. 7.8)

h. Clear digital PAUSE.

i. Place analog computer into operate. Run will now start.

7.8 *Analog Patching*

This section is devided into four parts.

A. Initial Condition Setup

B. Dynamics & Control

C. Thrustor Misalignments

D. Thrustor Real Effects & Fuel Consumption

INITIAL CONDITION SETUP

VARIABLE		POTENTIOMETER
MSG	REPORT	
X1(0)	$\phi(0)$	P50
X2(0)	$\psi(0)$	P51
X3(0)	$\theta(0)$	P54
X4(0)	$\omega_1(0)$	P55
X5(0)	$\omega_2(0)$	P58
X6(0)	$\omega_3(0)$	P59

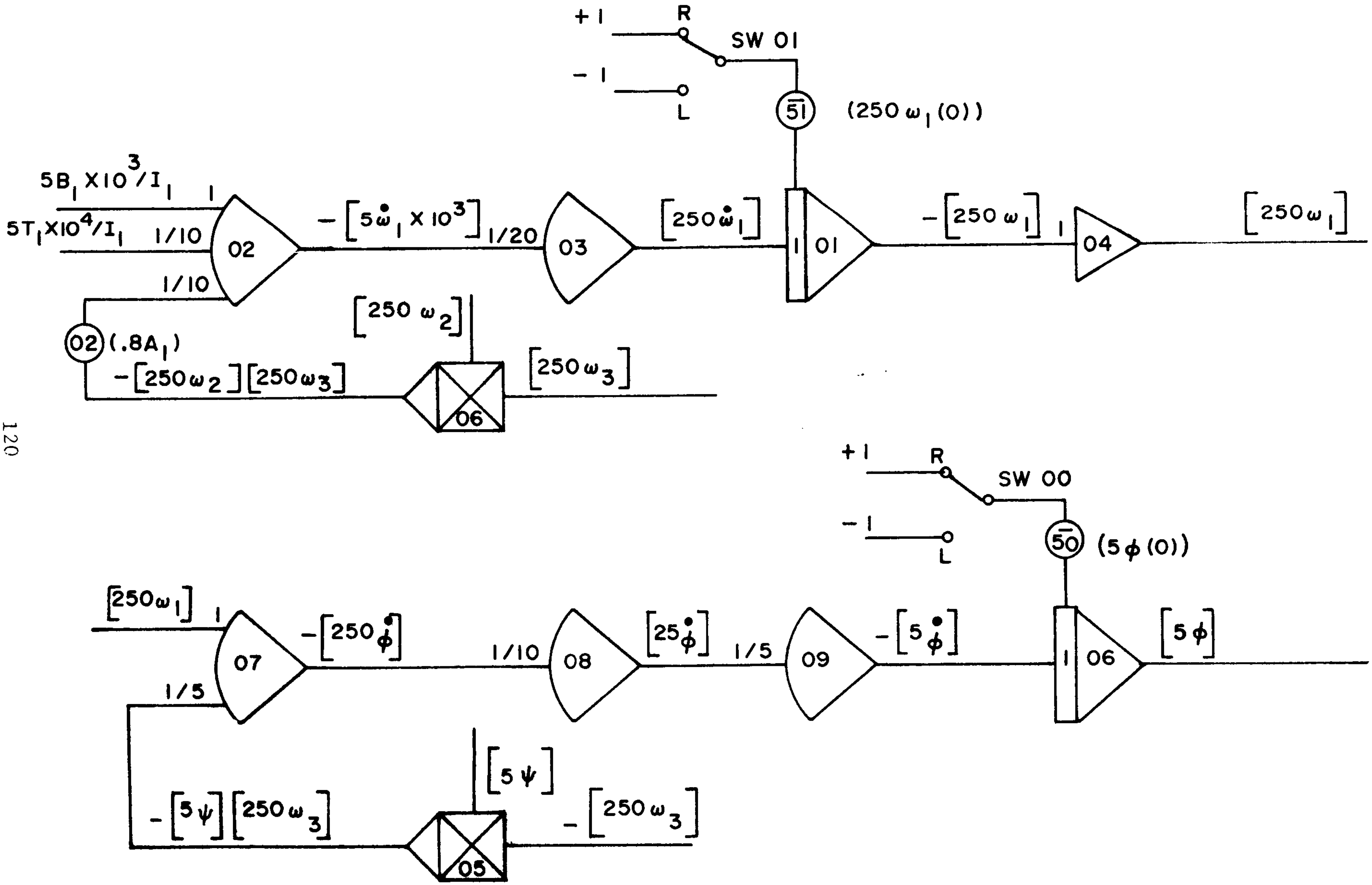
SWITCH / VARIABLE	00		10		20		01		11		21	
	R	L	R	L	R	L	R	L	R	L	R	L
X1(0)	−	+										
X2(0)			−	+								
X3(0)					−	+						
X4(0)							−	+				
X5(0)									−	+		
X6(0)											−	+

INITIAL CONDITION SET-UP

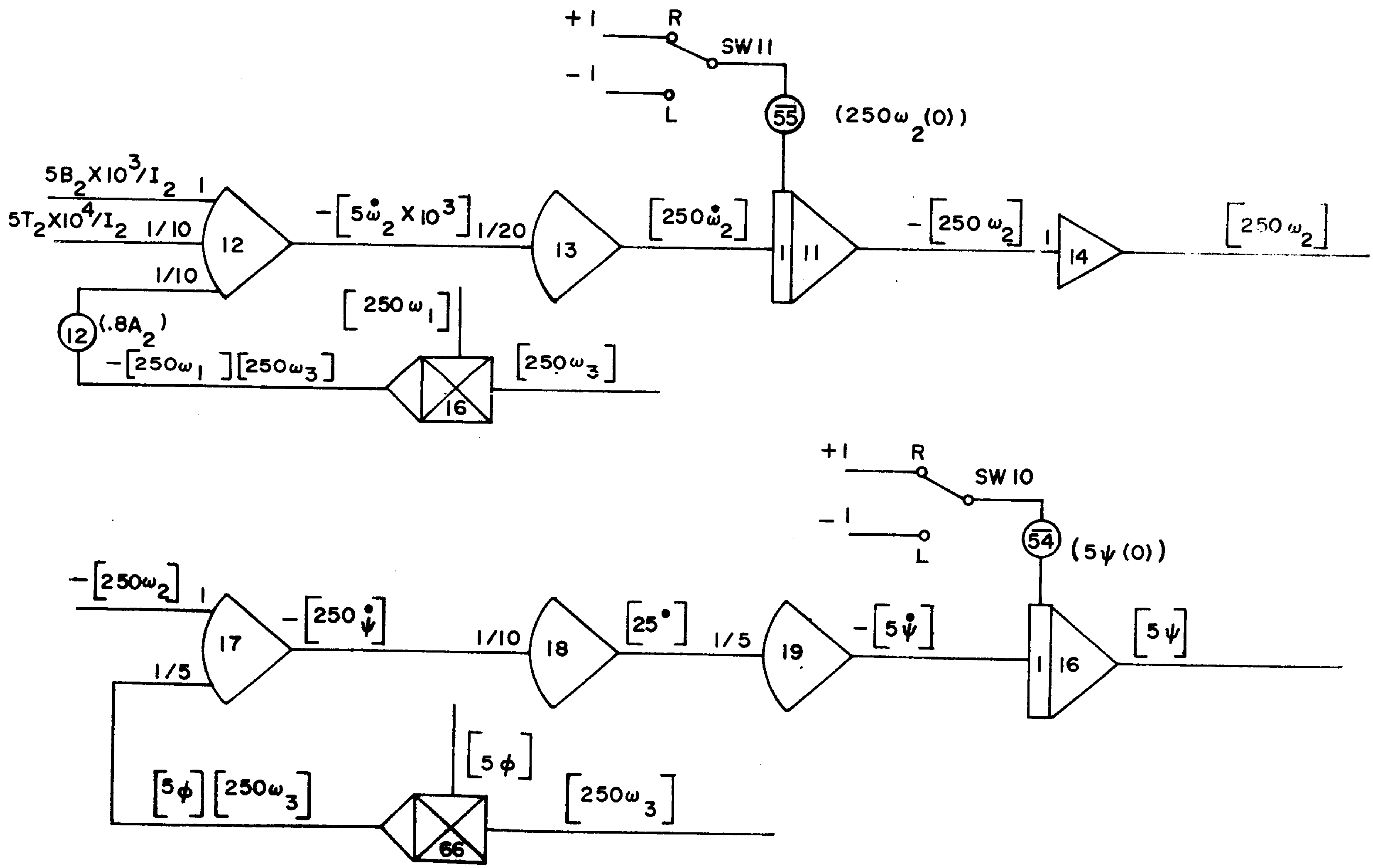
DYNAMICS & CONTROL

POTENTIOMETER	VARIABLE	VALUE
Q06	$(t_1 X d_1)\ (5)$	.4000
Q16	$(t_2 X d_2)\ (5)$	.2000
Q26	$(t_3 X d_3)\ (5)$	.6000
P06	$1000/I_1$	.2793
P16	$1000/I_2$	.5076
P26	$1000/I_3$	.5000
Q02	$(.8)(I_2-I_3)/I_1$	.0067
Q12	$(.8)(I_3-I_1)/I_2$	.6416
Q22	$(.8)(I_1-I_2)/I_3$	.6440

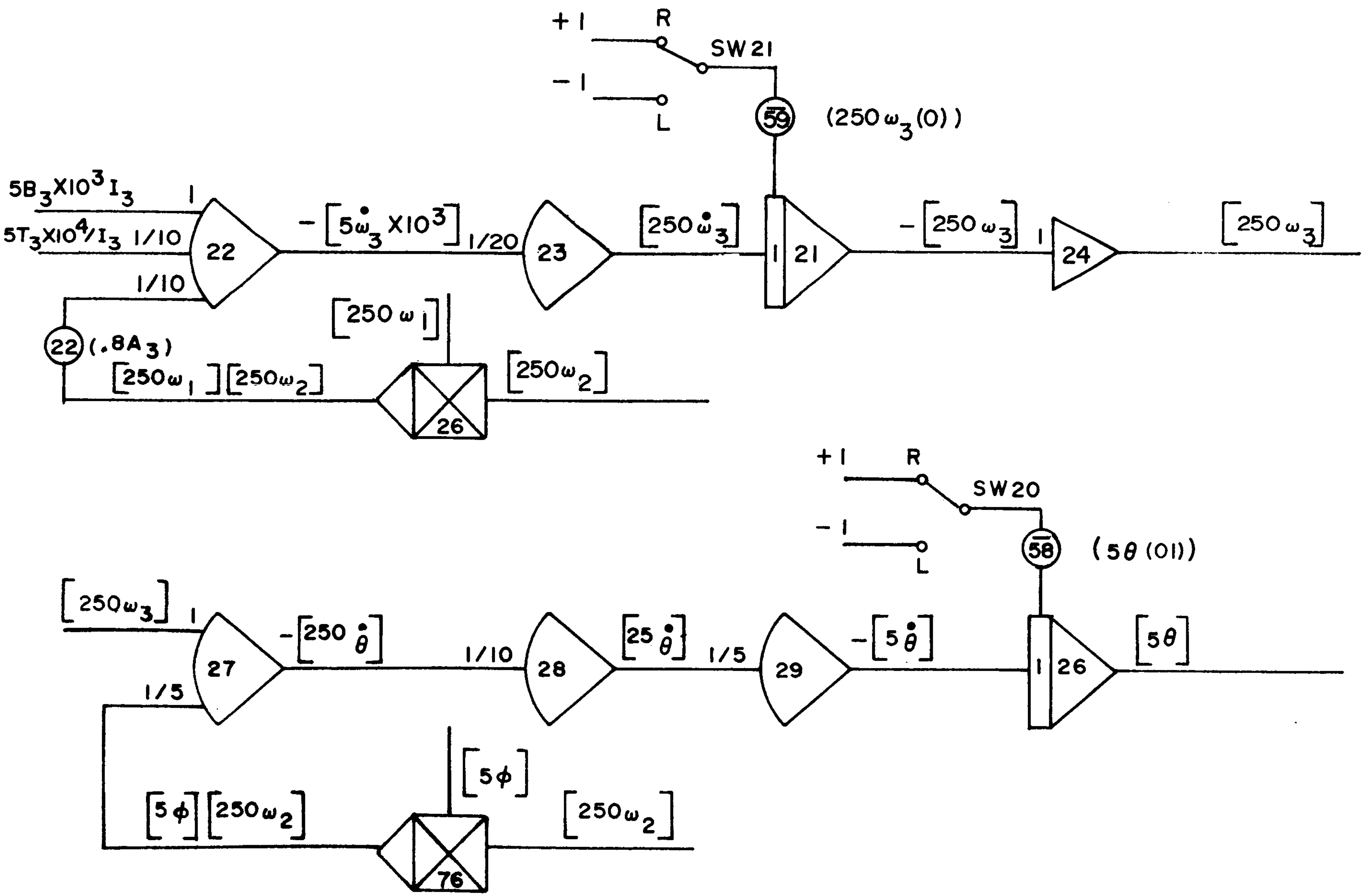
POTENTIOMETER SET-UP (DYNAMICS)



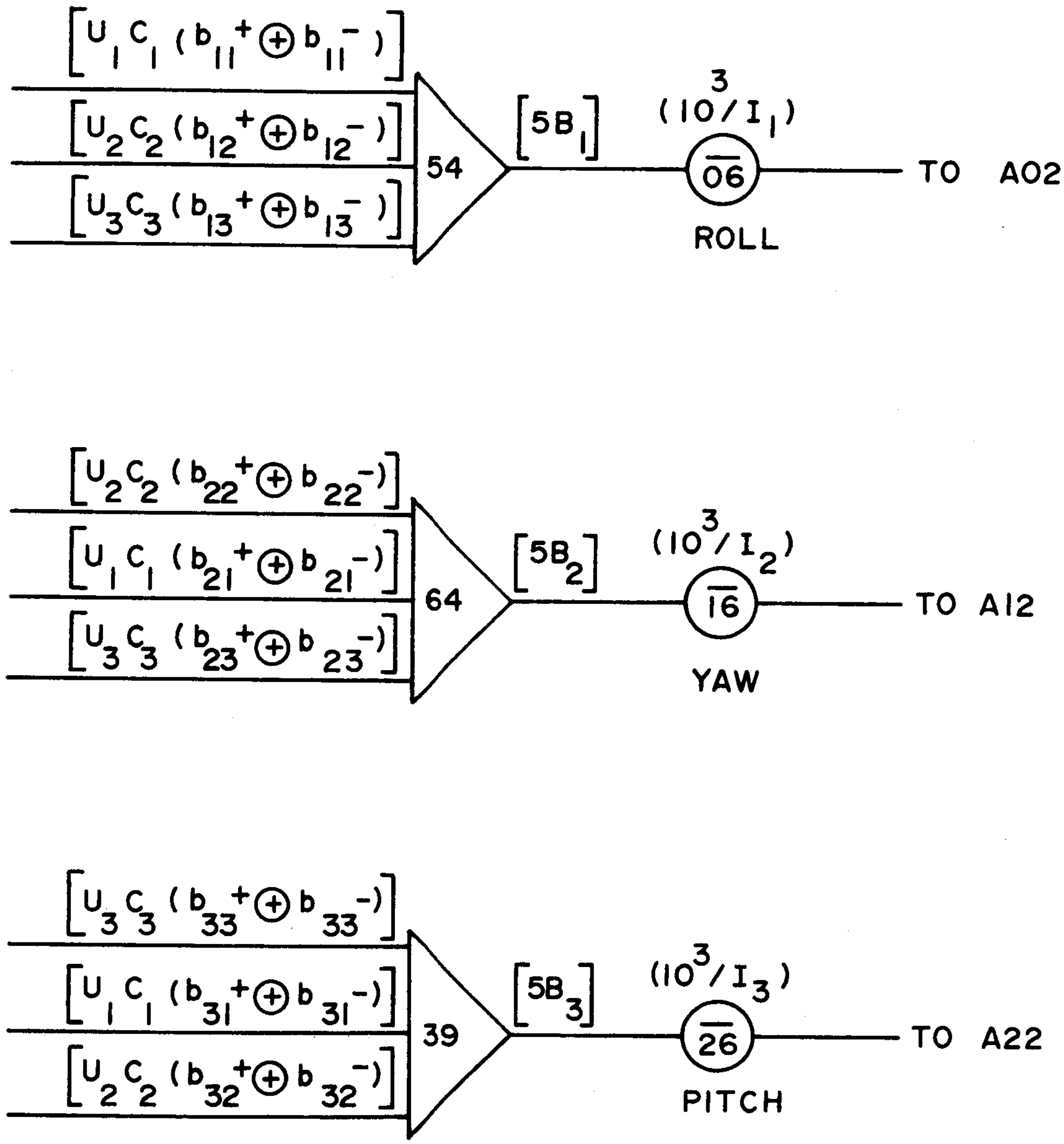
DYNAMICS AXIS I (ROLL)



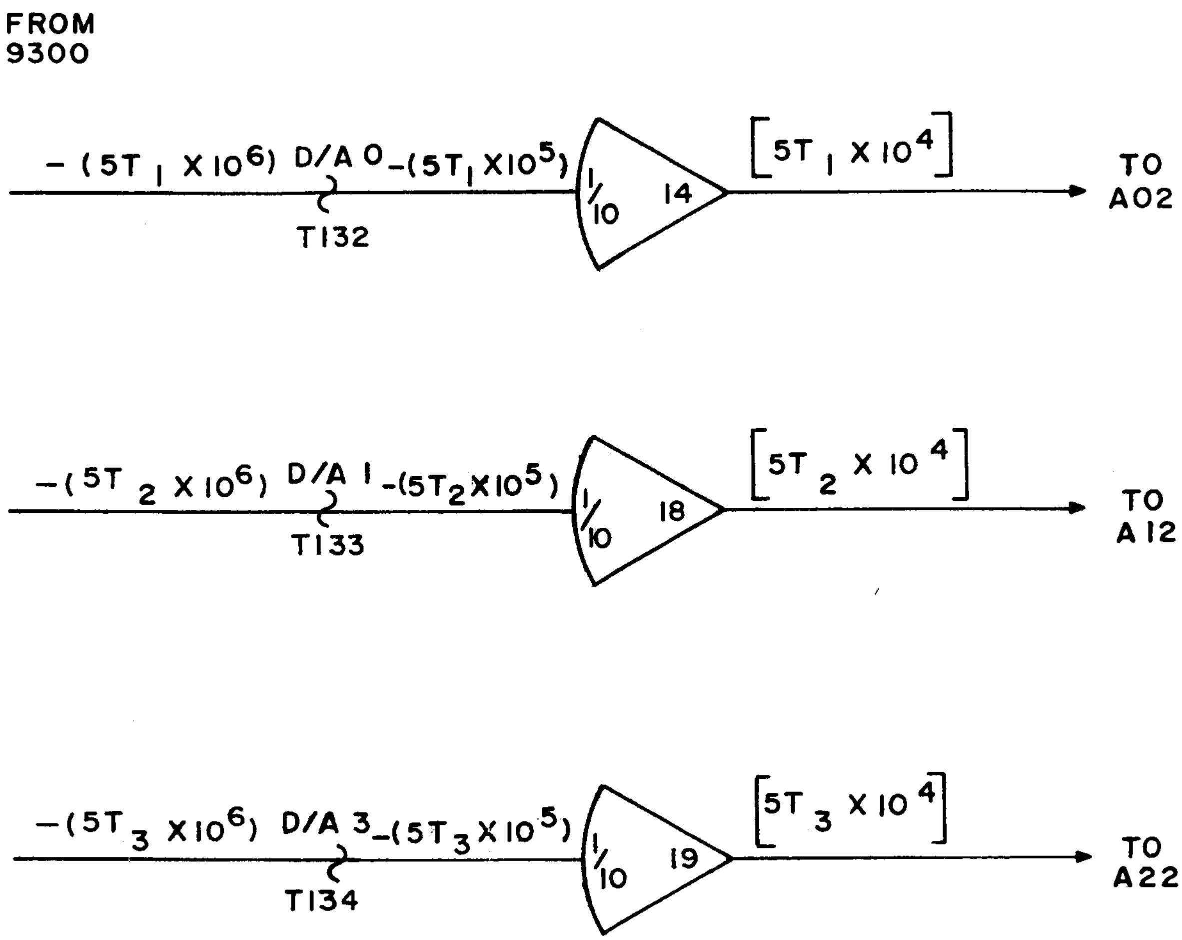
DYNAMICS AXIS 2 (YAW)



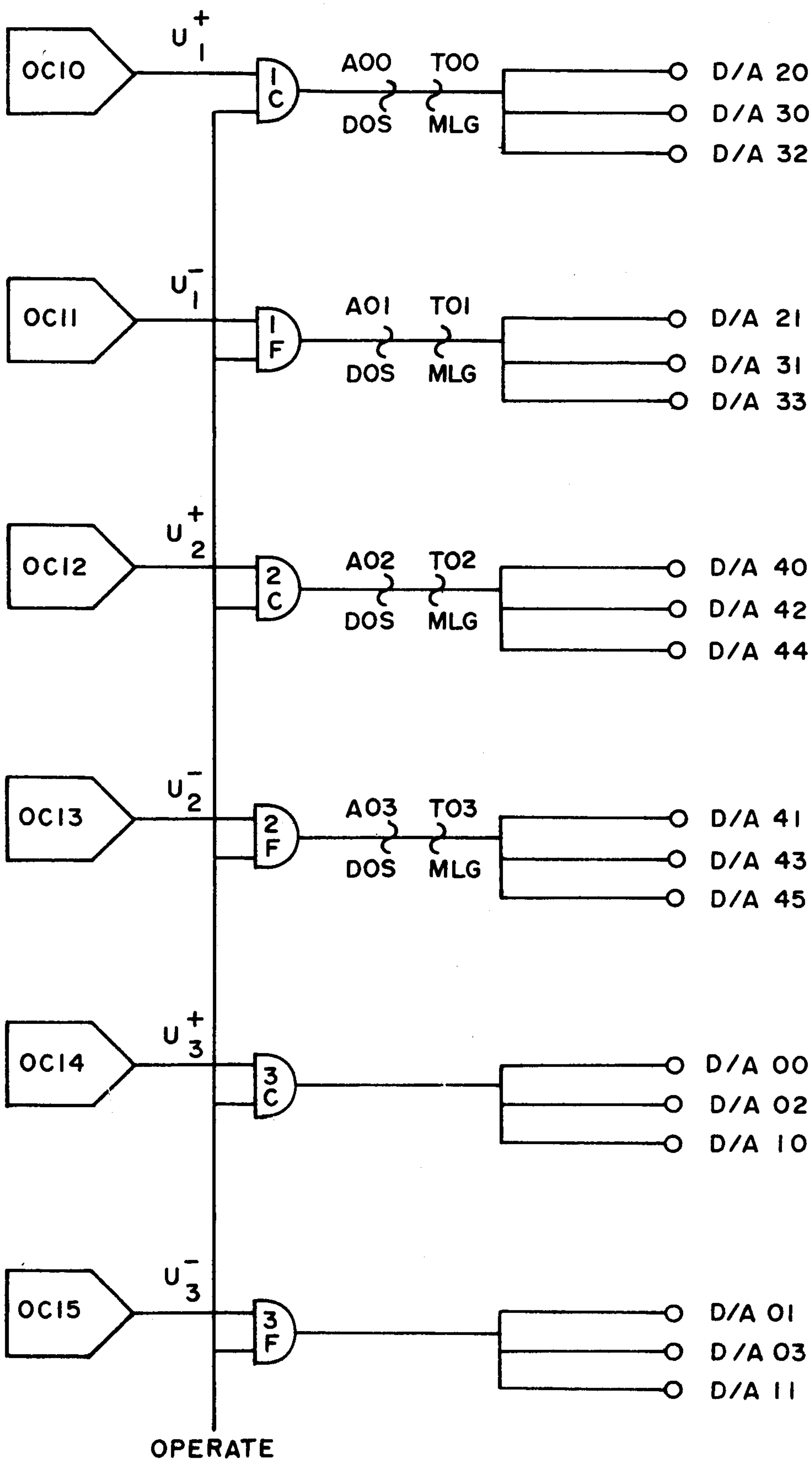
DYNAMICS AXIS 3 (PITCH)



LUMPED THRUST TORQUE



DISTURBANCE TORQUES

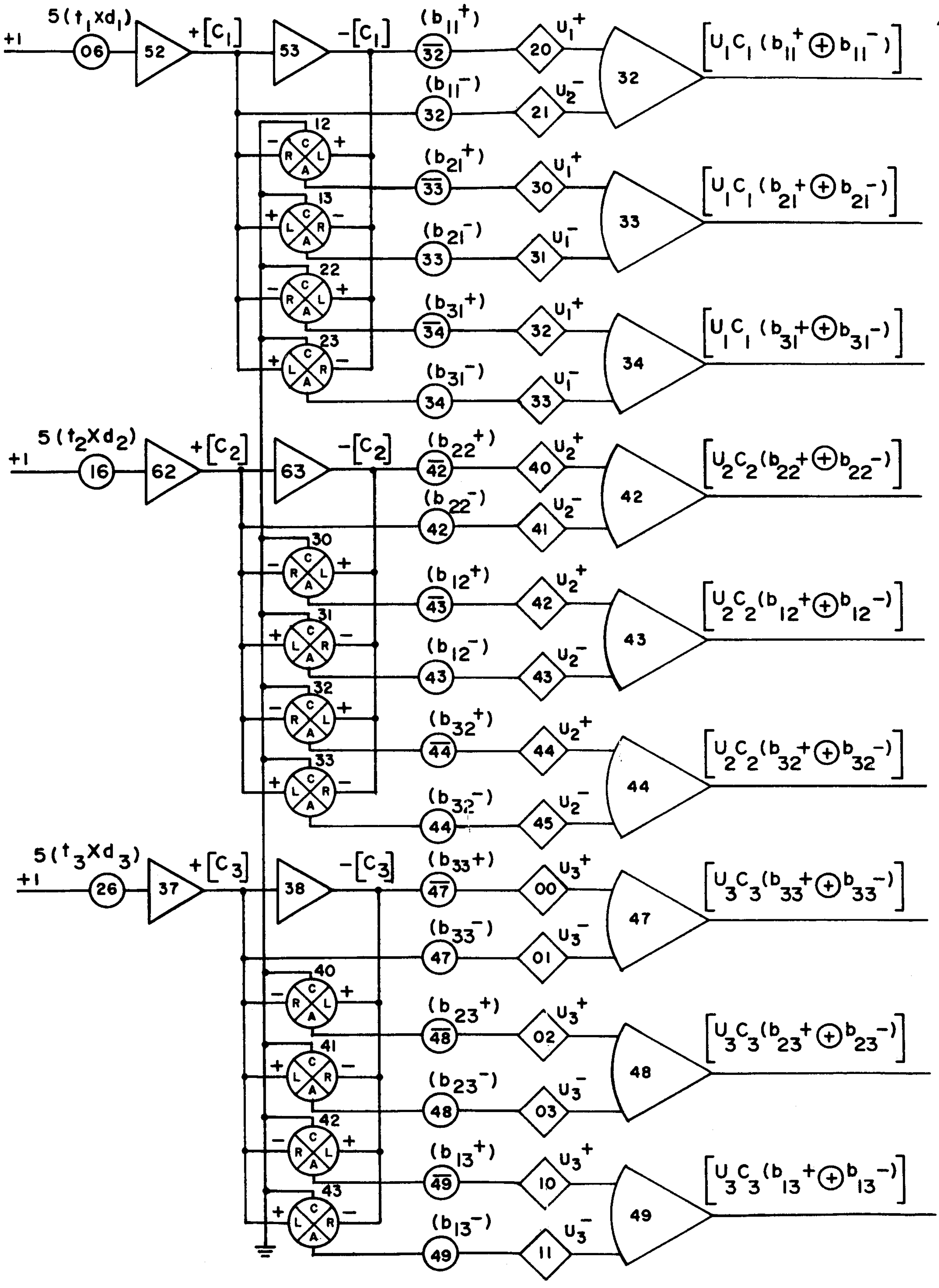


THRUST CONTROL LOGIC

THRUSTOR MISALIGNMENTS

POTENTIOMETER	VARIABLE	VALUE
P32	b_{11}^{+}	
P33	b_{21}^{+}	
P34	b_{31}^{+}	
Q32	b_{11}^{-}	
Q33	b_{21}^{-}	
Q34	b_{31}^{-}	
P42	b_{22}^{+}	
P43	b_{12}^{+}	
P44	b_{32}^{+}	
Q42	b_{22}^{-}	
Q43	b_{12}^{-}	
Q44	b_{32}^{-}	
P47	b_{33}^{+}	
P48	b_{23}^{+}	
P49	b_{13}^{+}	
Q47	b_{33}^{-}	
Q48	b_{23}^{-}	
Q49	b_{13}^{-}	

POTENTIOMETER SET-UP (MISALIGNMENT)

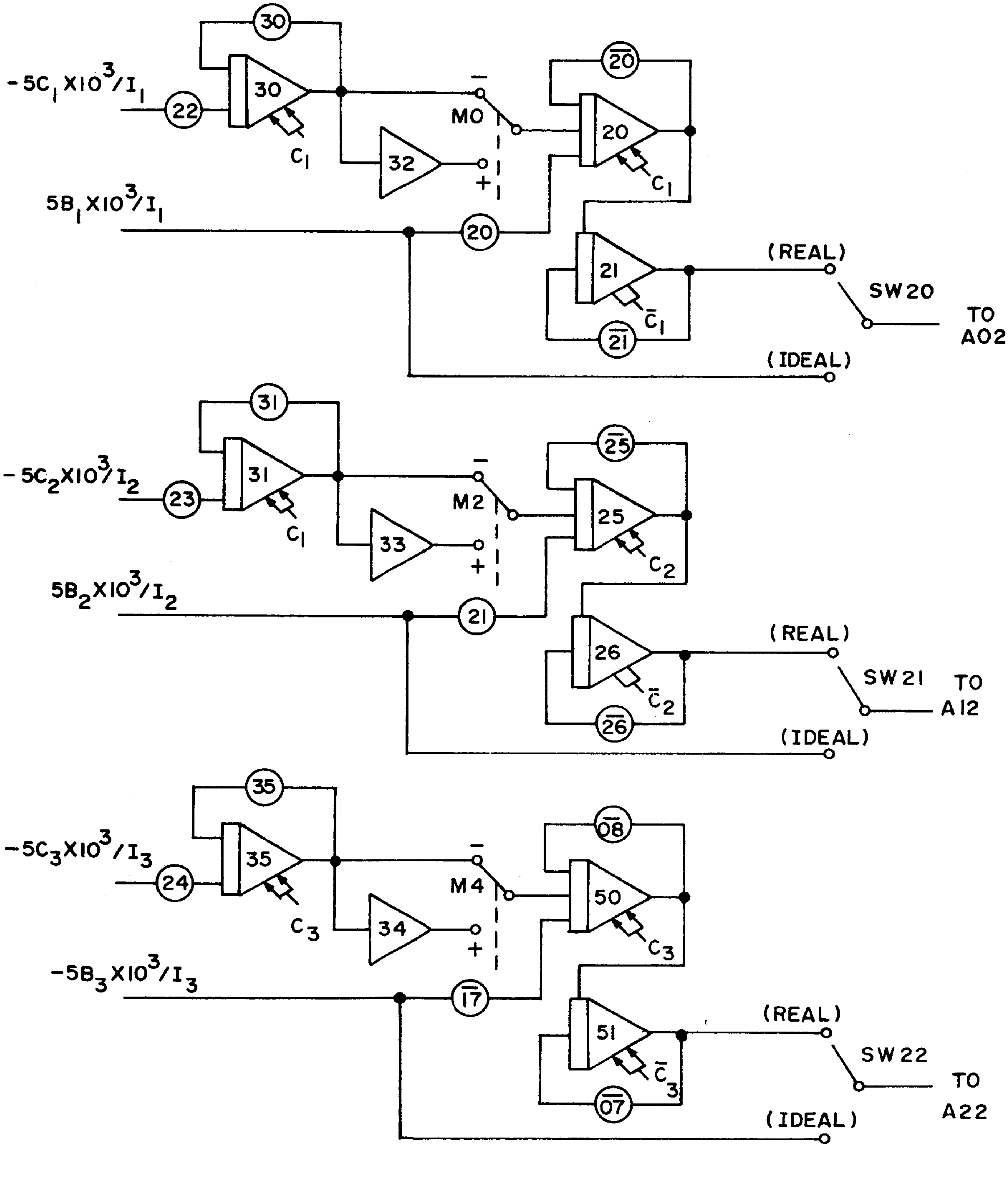


THRUSTOR MISALIGNMENT

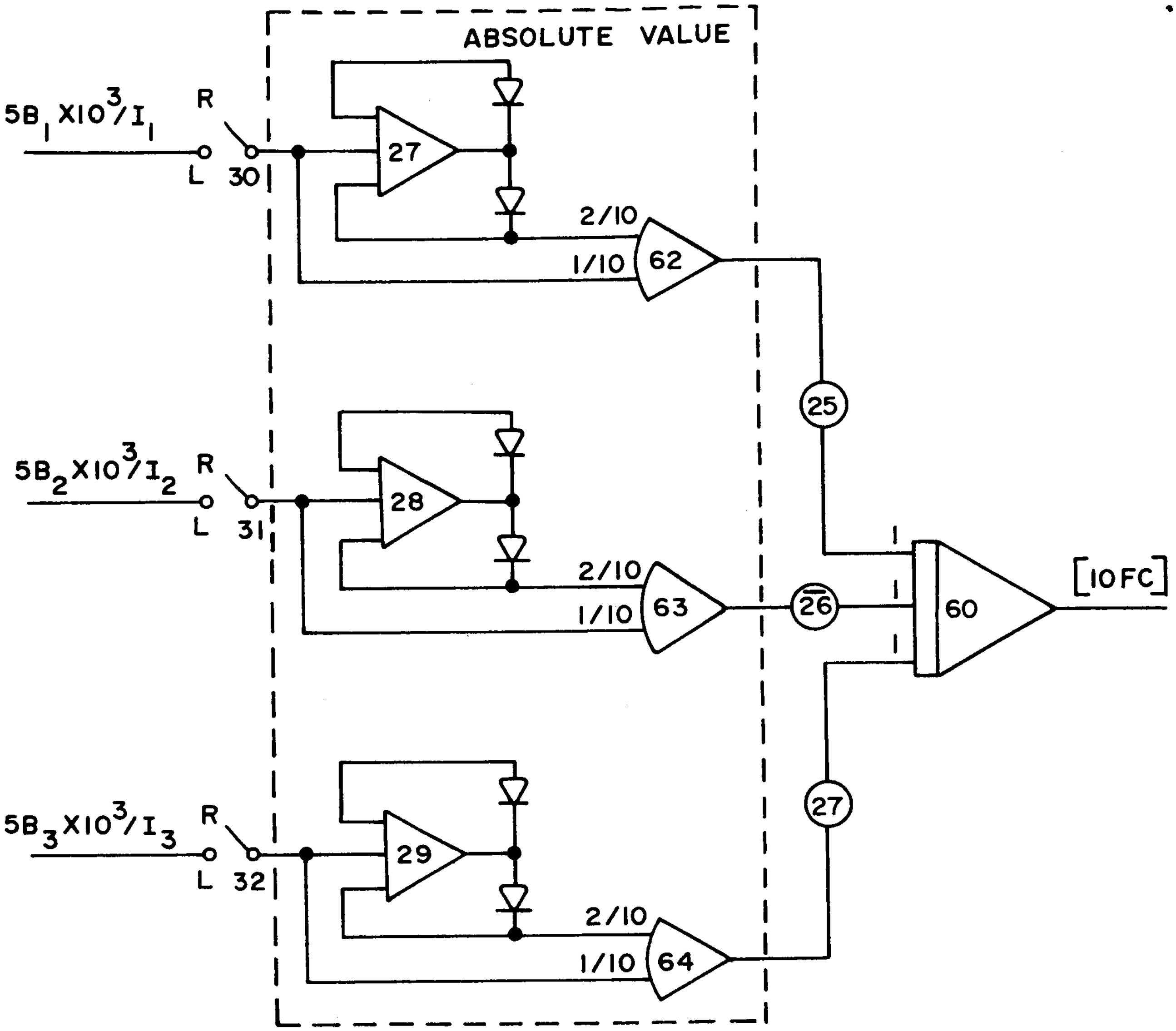
SWITCH	POSITION	
	RIGHT	LEFT
12	$-b_{21}^{+}$	$+b_{21}^{+}$
13	$-b_{21}^{-}$	$+b_{21}^{-}$
22	$-b_{31}^{+}$	$+b_{31}^{+}$
23	$-b_{31}^{-}$	$+b_{31}^{-}$
30	$-b_{12}^{+}$	$+b_{12}^{+}$
31	$-b_{12}^{-}$	$+b_{12}^{-}$
32	$-b_{32}^{+}$	$+b_{32}^{+}$
33	$-b_{32}^{-}$	$+b_{32}^{-}$
40	$-b_{23}^{+}$	$+b_{23}^{+}$
41	$-b_{23}^{-}$	$+b_{23}^{-}$
42	$-b_{13}^{+}$	$+b_{13}^{+}$
43	$-b_{13}^{-}$	$+b_{13}^{-}$

SWITCH SETTINGS
(THRUSTOR MISALIGNMENT)

THRUSTOR REAL EFFECTS & FUEL CONSUMPTION



THRUSTER REAL EFFECTS



$$Q25 = \frac{100\ I_1}{d_1 \times ISP \times 5 \times 10^4} = .0300$$

$$P26 = \frac{100\ I_2}{d_2 \times ISP \times 5 \times 10^4} = .0210$$

$$Q27 = \frac{100\ I_3}{d_3 \times ISP \times 5 \times 10^4} = .0180$$

FUEL CONSUMPTION

POTENTIOMETER	VARIABLE	VALUE
Q25	$100I_{1/d1} X I_{sp} X 5 X 10^4$	.0300
P26	$100I_{2/d2} X I_{sp} X 5 X 10^4$	.0210
Q27	$100I_{3/d3} X I_{sp} X 5 X 10^4$	.0180
Q20	K_{12}	.2800
Q21	K_{22}	.2800
Q22	K_{11}	.0018
Q23	K_{21}	.0018
Q24	K_{31}	.0018
Q30	K_{11}	.0150
Q31	K_{21}	.0150
Q35	K_{31}	.0150
P08	K_{32}	.2800
P07	K_{33}	.0750
P17	K_{32}	.2800
P20	K_{12}	.2800
P21	K_{13}	.0750
P25	K_{22}	.2800
P26	K_{23}	.0750

POT SET-UP

(TRE AND FUEL CONSUMPTION)

TRE* \ SWITCH	20		21		22	
	R	L	R	L	R	L
YES (AXIS 1)		X				
NO (AXIS 1)	X					
YES (AXIS 2)				X		
NO (AXIS 2)			X			
YES (AXIS 3)						X
NO (AXIS 3)					X	

* THRUSTOR REAL EFFECTS

FUEL CONSUMPTION \ SWITCH	30		31		32	
	R	L	R	L	R	L
YES (AXIS 1)		X				
NO (AXIS 1)	X					
YES (AXIS 2)				X		
NO (AXIS 2)			X			
YES (AXIS 3)						X
NO (AXIS 3)					X	

SWITCH SETTINGS
(TRE AND FUEL CONSUMPTION)