Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

(NASA-CR-158474) SIMULATION EVALUATION OF COMBINED 4D RNAV AND AIRBORNE TRAFFIC SITUATION DISPLAYS AND PROCEDURES APPLIED TO TERMINAL AERIAL MANUEVERS Semi-annual Progress Report, 1 Sep. 1978 (Massachusetts G3/04 N79-21033 Unclas 14735

Semi-Annual Progress Report

Simulation Evaluation of Combined 4D RNAV and Airborne Traffic Situation Displays and Procedures Applied to Terminal Area Manuevers

Principal Investigators: Michael Athans
Mark E. Connelly

Period: September 1, 1978 to March 1, 1979

Grant NSG-2180
NASA-Ames Research Center
Moffett Field, CA 94035

Massachusetts Institute of Technology
Laboratory for Information and Decision Systems
Cambridge, Massachusetts 02139

Semi-Annual Status Report

(NASA Ames Grant NSG-2180)

April 1979

Introduction

The principal objective of this research is to prepare and evaluate a set of simulation scenarios in which subject pilots must carry out the following critical approach functions simultaneously:

1. Follow a 3D terminal airspace structure and arrive at fixed waypoints within the structure precisely at pre-scheduled times in the presence of a full range of wind conditions aloft.

2. Monitor nearby traffic on an Airborne Traffic Situation Display, especially during merging and spacing operations, and detect blunders and resolve conflicts in a safe manner.

These functions represent two key tasks in the application of distributed management to the problem of providing adequate ATC capacity, safety, and efficiency at busy terminals.

Open-loop simulator tests of the single-stage 4D RNAV algorithm developed by the project indicate that a descending pilot can comply quite closely with an assigned time of arrival at a 3D waypoint simply by tracking a pre-calculated speed profile. In these tests, the pilot cuts back to idle thrust at a given DME distance from the waypoint and keeps the aircraft descending at constant Mach and/or constant EAS almost solely with stabilizer trim adjustments. Our initial experiments show that the aircraft arrives at the 3D waypoint within a few seconds of the anticipated time. The presence of headwinds or tailwinds does not affect the arrival time error as long as the wind is accurately modeled in the descent algorithm. The accuracy achieved in the open-loop, single-stage descents was much better than expected. These results all but guarantee that a 5 second standard deviation in arrival time error can be realized in closed-loop descents at very moderate pilot workload levels. The term "closed-loop" means that the descent profile required to get the aircraft to the 3D waypoint at the scheduled time is periodically recomputed throughout the descent and the pilot receives a continuous indication of the correct airspeed. The principal advantage of the closed-loop approach is that errors in wind estimation and pilot errors can be compensated for as long as the aircraft stays within its normal speed-altitude envelope. The disadvantage of the closed-loop technique is that an on-board computer, properly interfaced with other aircraft systems, is required, whereas the open-loop technique can be implemented in the immediate future employing existing hand-calculators such as the TI59. For all practical purposes, it appears that the main limitation on the performance of an open-loop descent is the degree to which winds aloft can be accurately estimated.

Research Activity

A. Open-Loop Descent Algorithm Development

The following working programs have been written by the project based on the descent algorithm analyzed in the last semi-annual progress report:

1. Complete Algorithm for the TI59 Hand Calculator (predicts horizontal distance and elapsed time for an idle thrust descent from 36,000' to 10,000' given the desired Mach during the first phase of the descent and the desired equivalent airspeed during the second phase). Running Time is 5 minutes, 13 seconds.

2. Simplified Algorithm for the TI59 applicable to a descent between any two integer altitudes in the range 40,000' to sea level with constant headwinds/tailwinds. Running time has been reduced to 2 minutes, 40 seconds by using only one iteration at each altitude level and simplifying the computation in the transition zone between constant EAS and constant Mach.

3. Complete Algorithm in FORTRAN for running baseline solutions on the Adage AGT-30 computer. The values obtained from this program have been used to check the results of open-loop descents in the 707 simulator from 36,000' to 10,000' and to check the TI59 results.

4. Real-time, closed-loop algorithm in Adage AGT-30 assembly language (ADEPT) continuously computes and displays the Mach or EAS value required to arrive at a 3D waypoint at the assigned time.

The basic building block for all of these programs is a computation sequence which estimates the horizontal distance and elapsed time corresponding to a given Mach-EAS descent profile. The algorithm is based on two equations which sum the longitudinal wind axis force components (zero thrust assumed):

$$-D - mg \sin \gamma = m(\dot{TAS} + \dot{w} \cos \gamma)$$

$$-L + mg \cos \gamma = m(\dot{w} \sin \gamma - \dot{\gamma}\, TAS)$$

These equations are solved at discrete altitude levels between the aircraft's cruise altitude and the desired altitude at the destination waypoint. At present, even numbered altitudes between sea level and 40,000 ft. are employed as computation points. Using a piecewise-linear approximation to the trajectory, the time and horizontal distance required to traverse each 2,000 ft. altitude increment are then found. These values are summed to obtain the overall elapsed time and the horizontal distance covered during the descent.

As with most finite difference algorithms of this type, some of the values required during the calculations are not available until the calculation is is completed. As a consequence, the computation must be repeated (iterated) one or more times to obtain more accurate approximations for the missing values. In the M.I.T. descent algorithm, no more than two passes through the routine at each altitude level are employed. Additional iterations, it was found, have very little effect on the results.

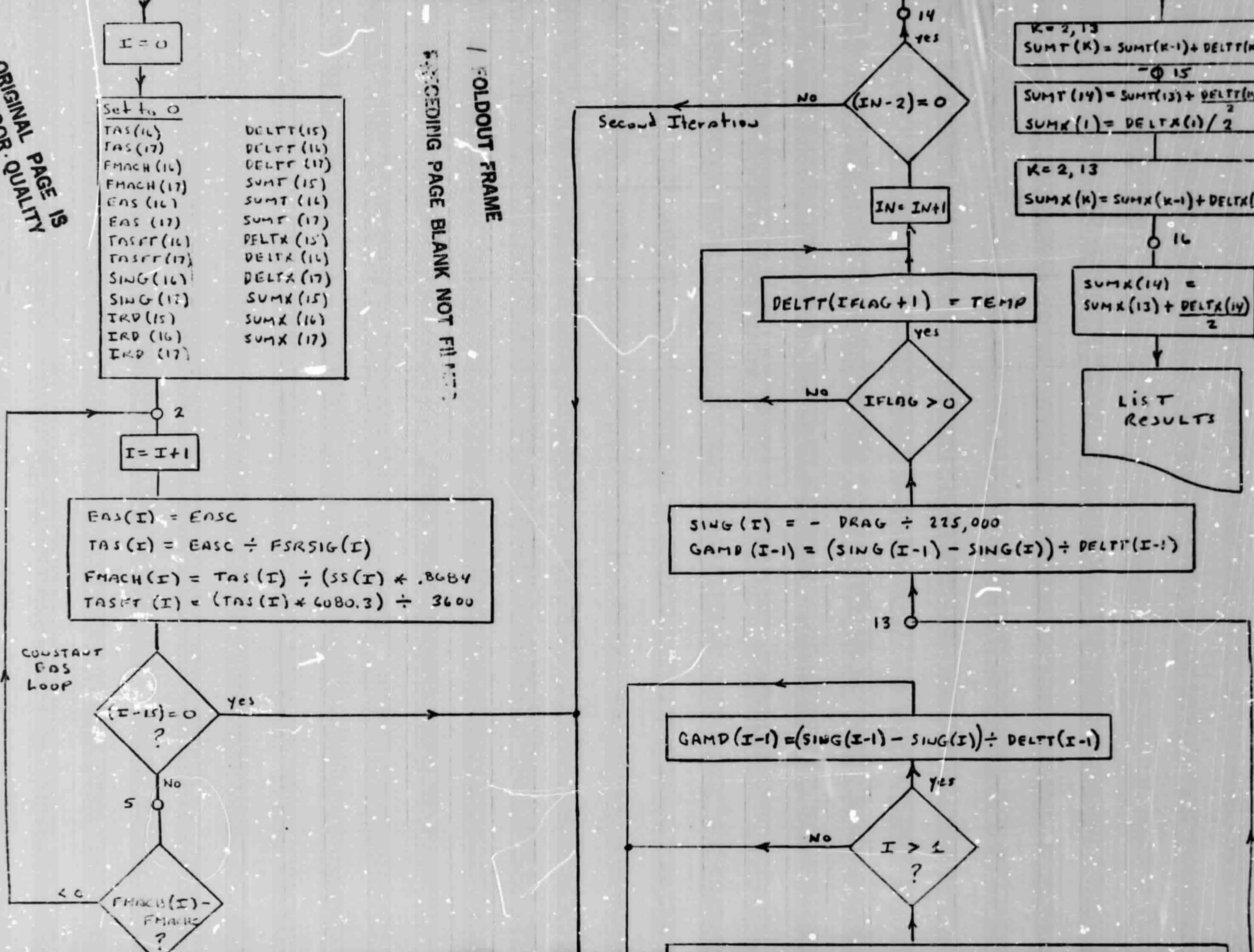
The same basic algorithm is implemented quite differently in FORTRAN, in the TI59, and in the Adage assembly language ADEPT because of the unique constraints imposed in each case. The FORTRAN program, for example, makes one complete pass through all the altitude levels, storing the iterim results in ten tables for use in the second pass. This extravagant use of storage tables was impossible in the TI59, which is memory limited. In the latter program, consequently, two iterations are executed at each altitude in sequence in the full version and one iteration per altitude in the short version. Minimizing the running time was the main consideration in organizing the real-time ADEPT program. Since it operates repetitively at a one solution per second rate during the descent, values generated in the prior frame are used to approximate the missing variables in the current frame, hence only one pass through the algorithm is required per frame.

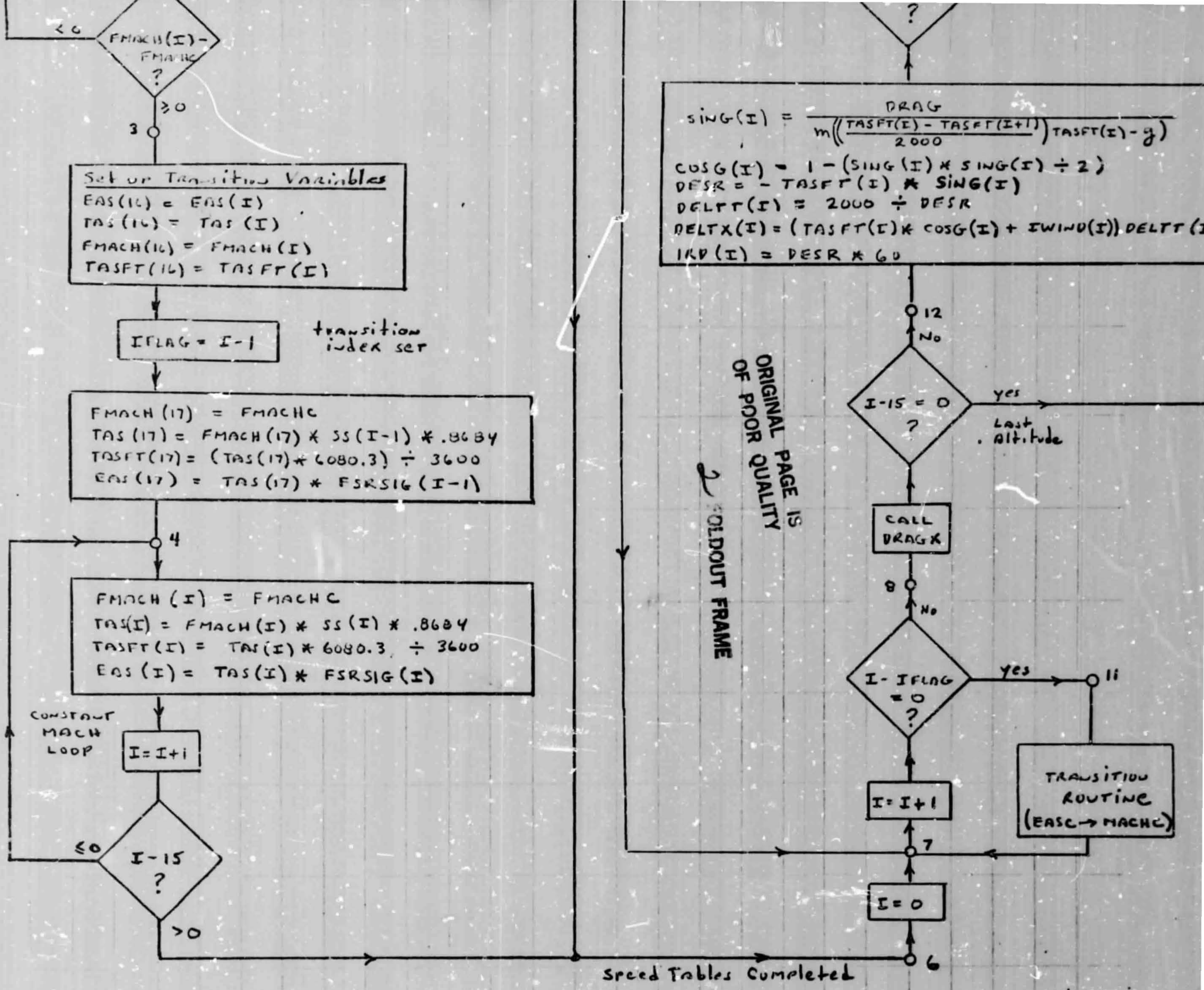
A flow chart for the FORTRAN version of the algorithm is given in Figure 1 and a flow chart of the short TI59 program is presented in Figure 2. A listing of the FORTRAN program is given in Appendix A as well as a set of outputs for three descent profiles (standard, medium speed, slow speed). A listing of the short TI59 program is given in Appendix B. The real-time Adage descent program is still being tested and refined, hence it will be documented at a later date.

The results from the FORTRAN and TI59 program agree quite well. For example, on a standard .83 Mach - 320 knot EAS descent from 36,000' to 10,000' the FORTRAN program predicts an elapsed time of 591 seconds and the TI59 program predicts 591.6 seconds. Horizontal distance according to the FORTRAN program will be

FIGURE 1

FORTRAN 4DRNAV (1)





2 FOLDOUT FRAME

FORTRAN 4DRNAV (2)

11 TRANSITION ROUTINE

↓

CALL DRAGX

↓

$$SING(I) = \frac{DRAG}{6994\left(\left(\frac{TASFT(I) - TASFT(16)}{2000}\right) TASFT(I) - 32.17\right)}$$

COSG(I) = 1 − (SING(I) * SING(I) ÷ 2)

DESR = − TASFT(I) * SING(I)

DELTT(I) = 2000 ÷ DESR

IRD (I) = DESR * 60

GAMD (I-1) = (SING (I-1) − SING (I)) ÷ DELTT (I-1)

↓

I = 16

↓

CALL DRAGX

I = IFLAG

↓

$$SING(16) = \frac{-DRAG}{225000}$$

GAMD (I) = (SING (I) − SING (16)) ÷ DELTT (I)

↓

I = 17

↓

CALL DRAGX

I = IFLAG + 1

↓

$I = ITLAG + 1$

$$SING(17) = \frac{DRAG}{6994\left(\left(\frac{TASFT(17) - TASFT(I)}{2000}\right) TASFT(17) - 32.17\right)}$$

$DESR = - TASFT(17) * SING(17)$

$DELTT(17) = 2000 \div DESR$

CALL DRAGX

$$SING(I) = \frac{DRAG}{6994\left(\left(\frac{TASFT(I) - TASFT(I+1)}{2000}\right) TASFT(I) - 32.17\right)}$$

$COSG(I) = 1 - (SING(I) * SING(I) \div 2)$

$DESR = - TASFT(I) * SING(I)$

$DELTT(I) = 2000 \div DESR$

$IRD(I) = DESR * 60$

$GAMD(17) = (SING(17) - SING(I)) \div DELTT(17)$

$$HTMHK = \frac{2000 * (TAS(17) - TAS(I-1)}{TAS(IL) - TAS(I-1) + TAS(17) - TAS(I)}$$

$$DELTT(I-1) = \frac{-(HTMHK + 1000)}{TASFT(I-1) * SING(I-1)}$$

$DELTX(I-1) = (TASFT(I-1) * COSG(I-1) + IWIND(I-1)) * DELTT(I-1)$

$$TEMP = \frac{-(3000 - HTMHK)}{TASFT(I) * SING(I)}$$

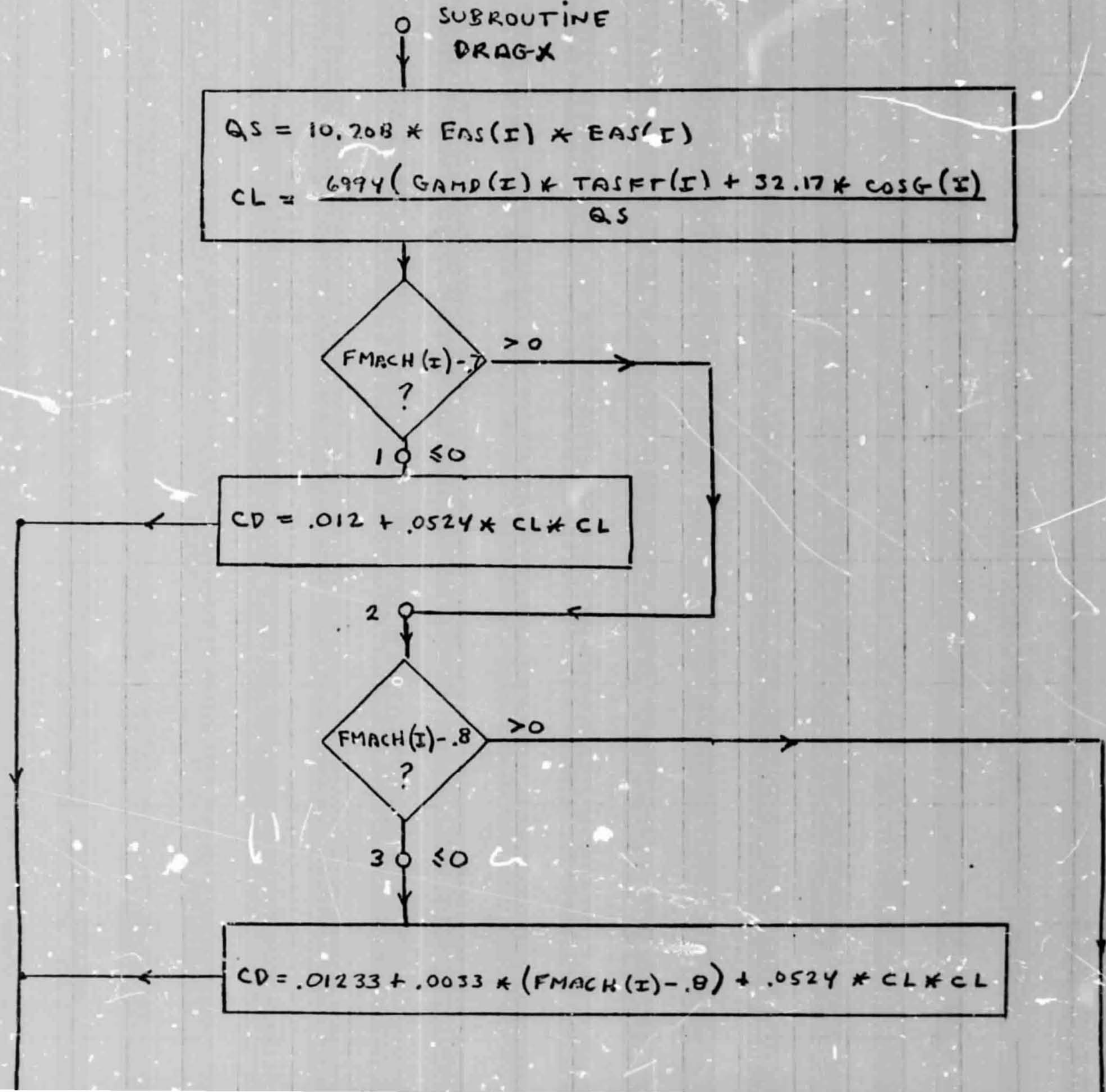
$DELTX(I) = (TASFT(I) * COSG(I) + IWIND(I)) * TEMP$

GOTO 7

ORIGINAL PAGE IS OF POOR QUALITY

2 FOLDOUT FRAME

FORTRAN 4DRNAV (3)



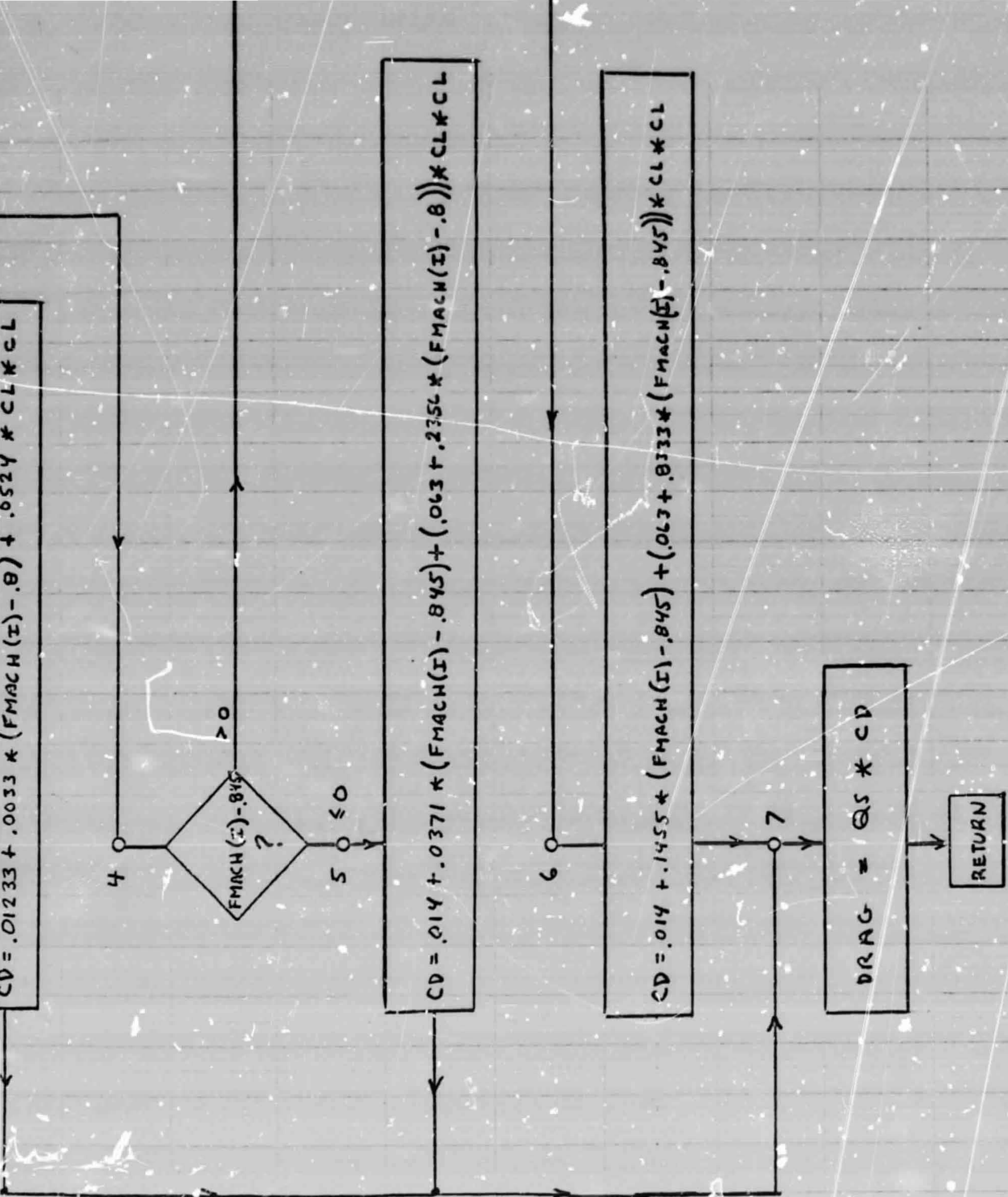
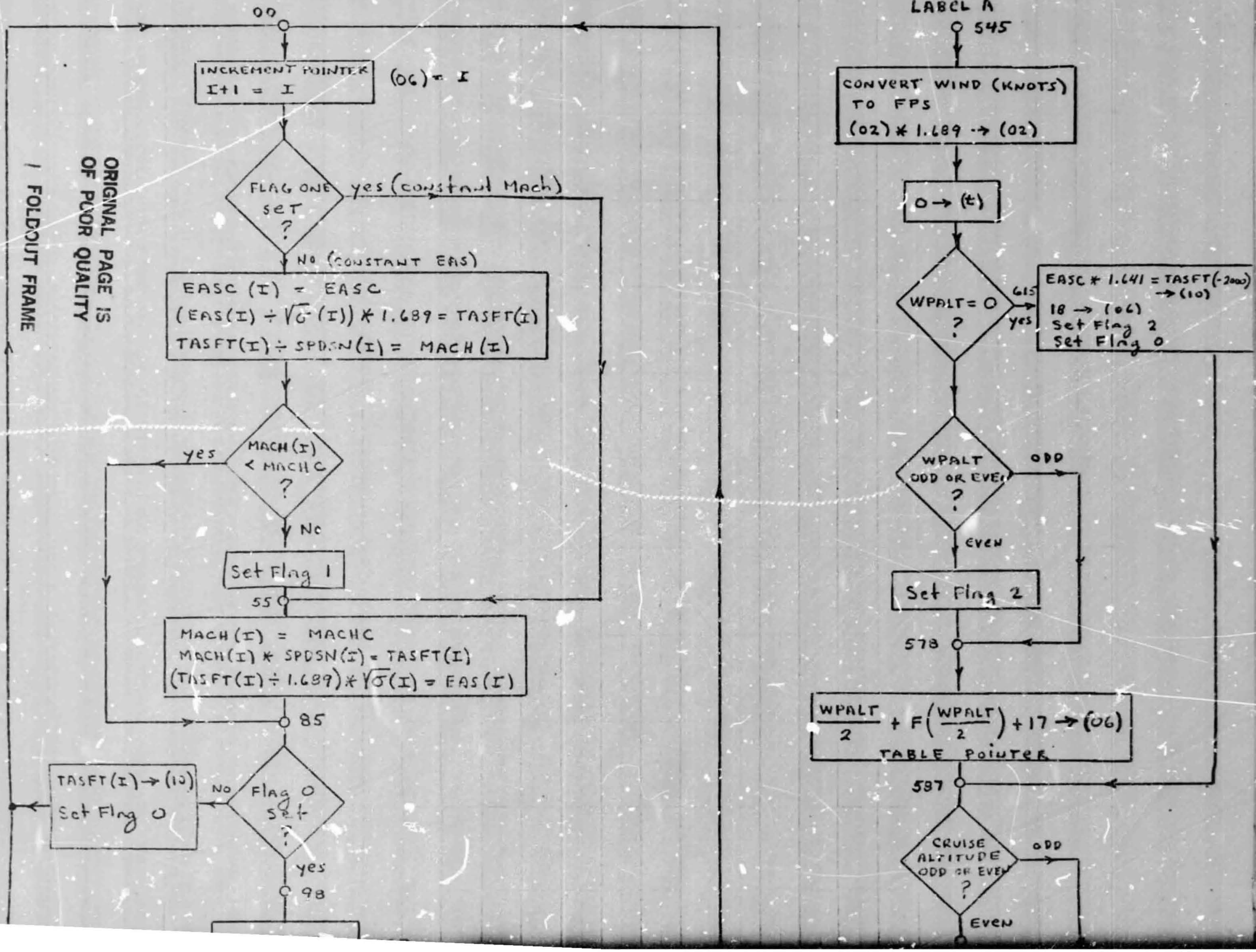
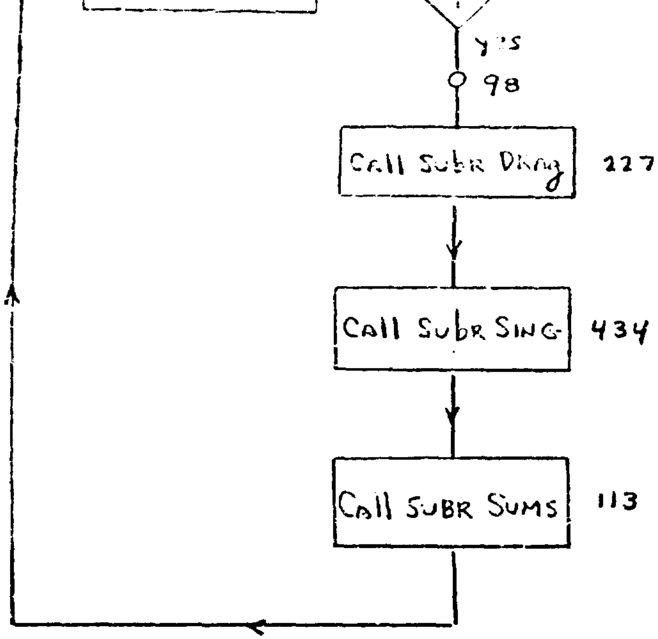

CD = .01233 + .0033 * (FMACH(I) - .8) + .0524 * CL * CL
4
FMACH(I)-.845 ?
> 0
5
≤ 0
CD = .014 + .0371 * (FMACH(I) - .845) + (.063 + .2356 * (FMACH(I) - .8)) * CL * CL
6
CD = .014 + .1455 * (FMACH(I) - .845) + (.063 + .8133 * (FMACH(I) - .845)) * CL * CL
7
DRAG = QS * CD
RETURN

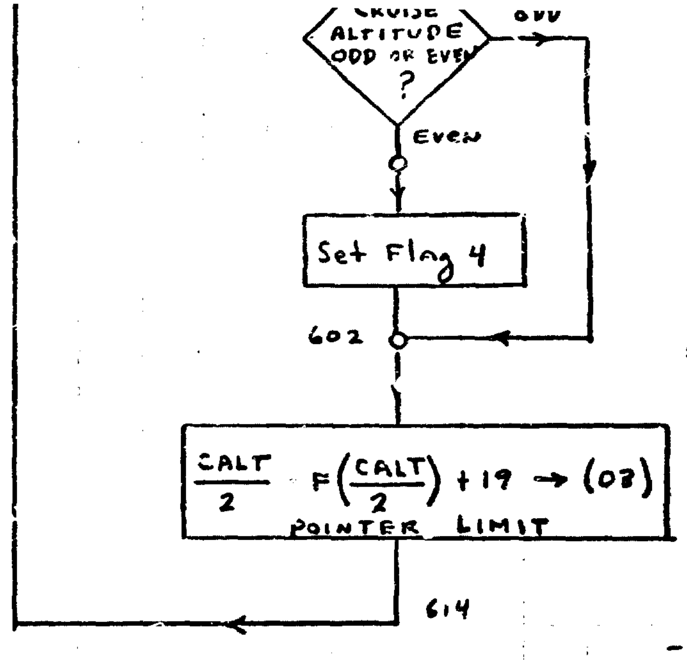
FIGURE 2

TI57 NEW SHORT ALGORITHM (1)

(PUSH RST BEFORE RUN TO LOWER FLAGS)







FOLDOUT FRAME

ORIGINAL PAGE IS OF POOR QUALITY

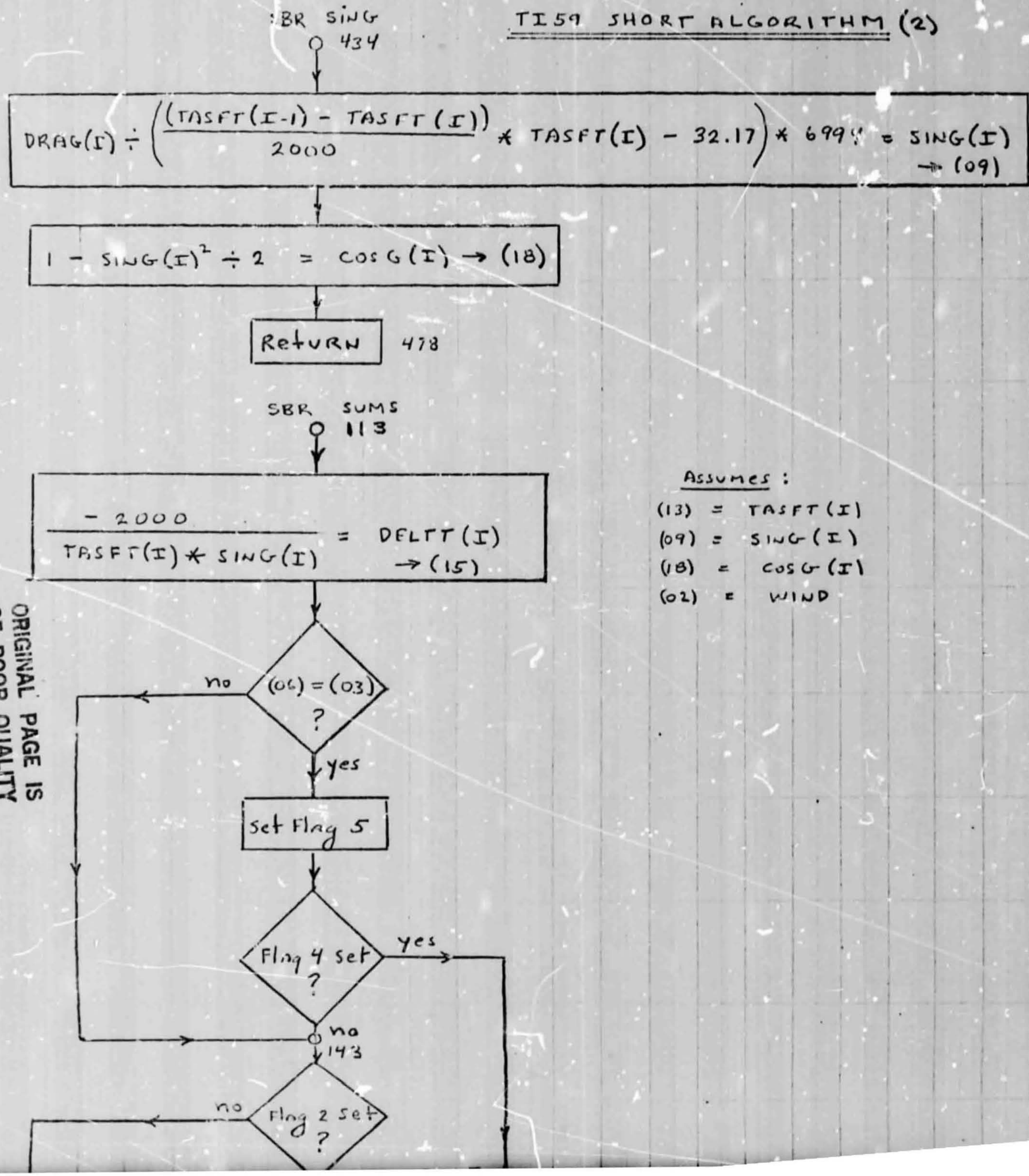
FLAG ASSIGNMENTS

0	FIRST PASS COMPLETED
1	CONSTANT MACH ZONE
2	WAYPOINT ALTITUDE EVEN
4	CRUISE ALTITUDE EVEN
5	LAST ALTITUDE

MEMORY ASSIGNMENTS

00	MACHC
01	EASC KNOTS
02	HW/TW KNOTS → FPS
03	CRUISE ALTITUDE (1000's) → PTR LIMIT
04	WAYPOINT ALTITUDE (1000's) → DRAG(I)
05	TEMPORARY STORE
06	SPDSN - $\sqrt{\sigma}$ TABLE POINTER
07	
08	CL^2
09	SING(I)
10	TASFT (I-1)
11	
12	MACH (I)
13	TASFT (I)
14	EAS(I)
15	QS (I) → DT(I)
16	SUM T
17	SUM X
18	COSG (I)
19-39	SPDSN (ft/sec * 10) integer part $\sqrt{\sigma}$ FRACTION PART SEA LEVEL TO 40,000'

TI59 SHORT ALGORITHM (2)

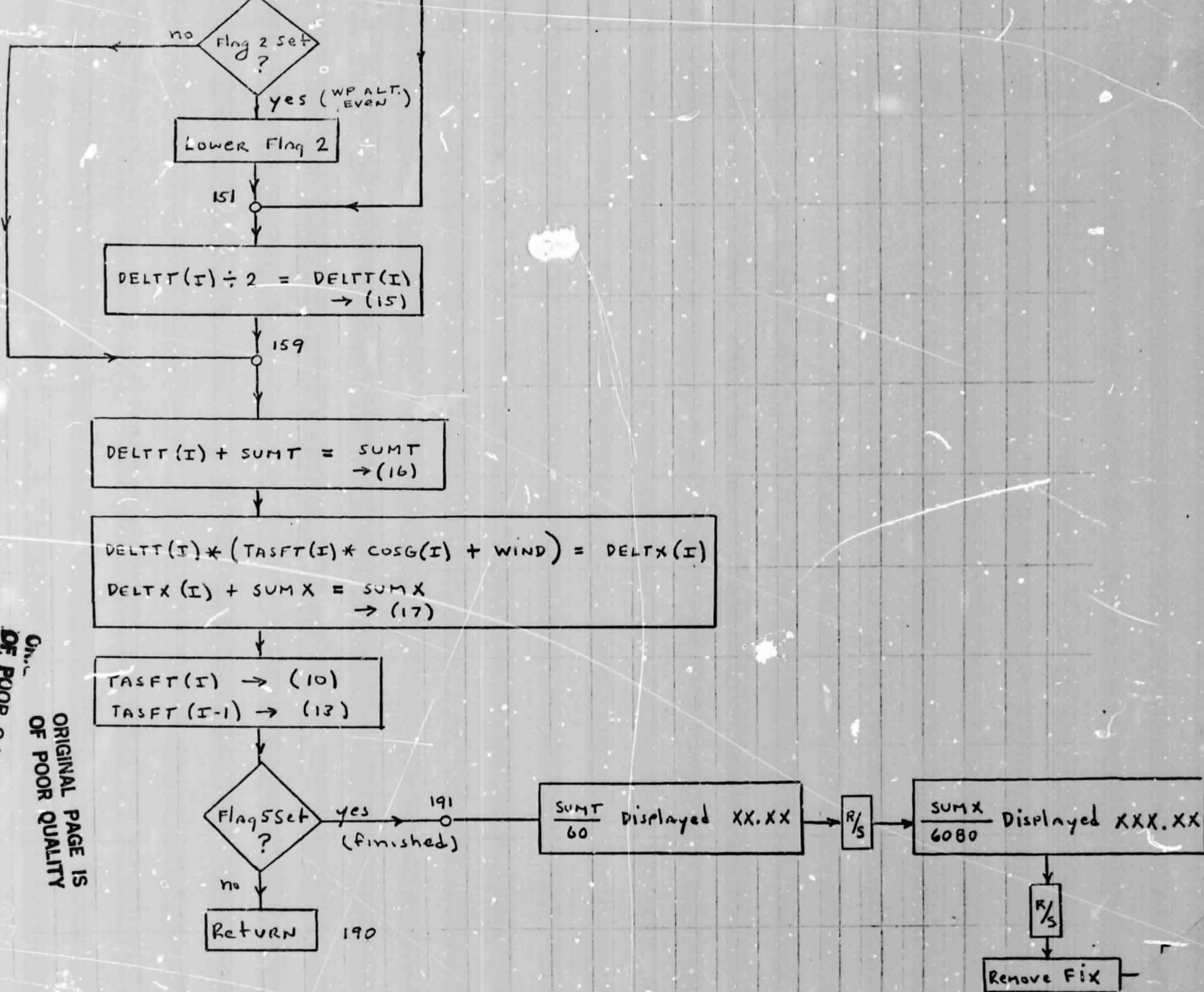


Assumes:
(04) = DRAG(I)
(10) = TASFT(I-1)
(13) = TASFT(I)
(09) = SING(I)

Assumes:
(13) = TASFT(I)
(09) = SING(I)
(18) = COSG(I)
(02) = WIND

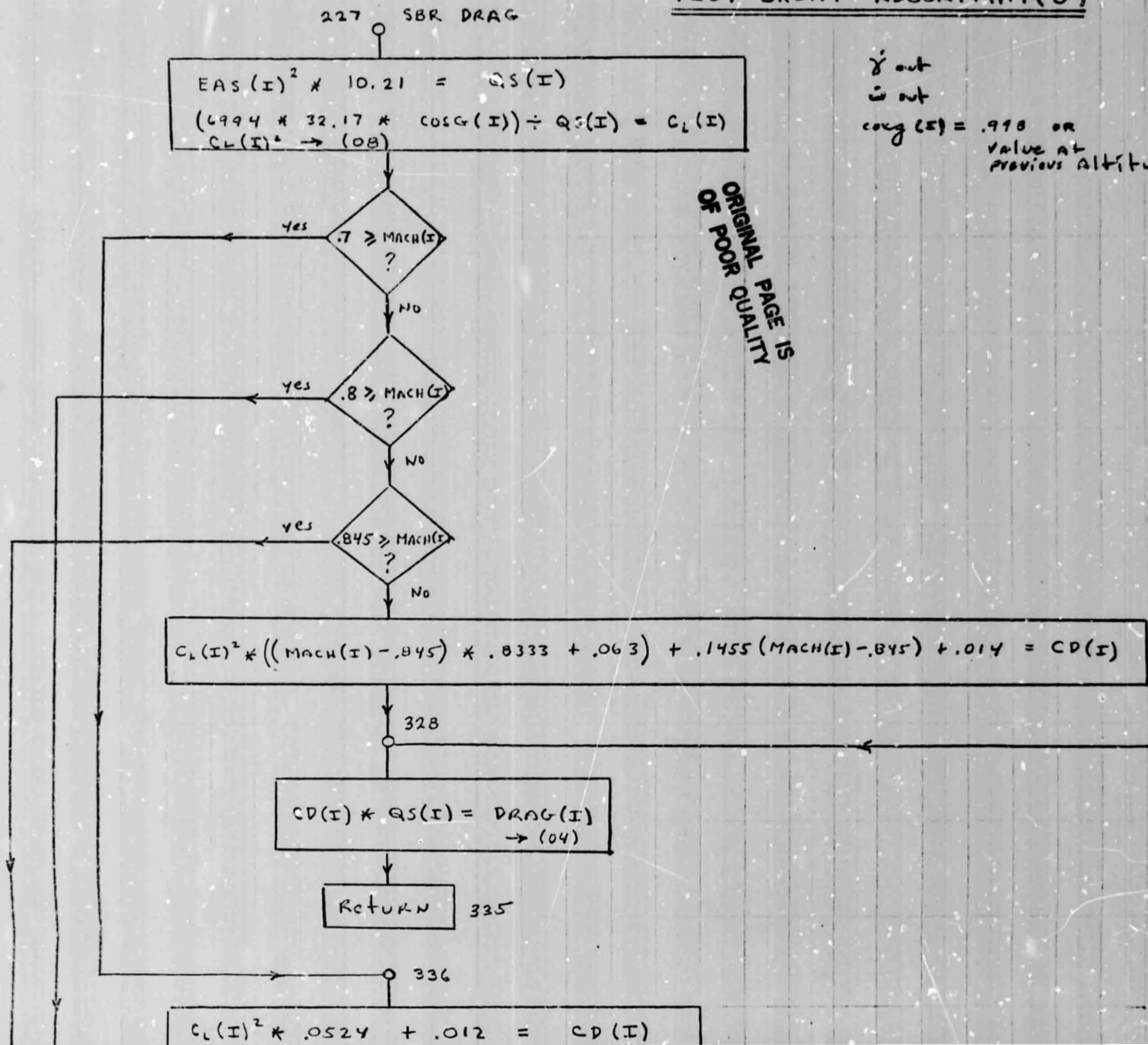
FOLDOUT FRAME

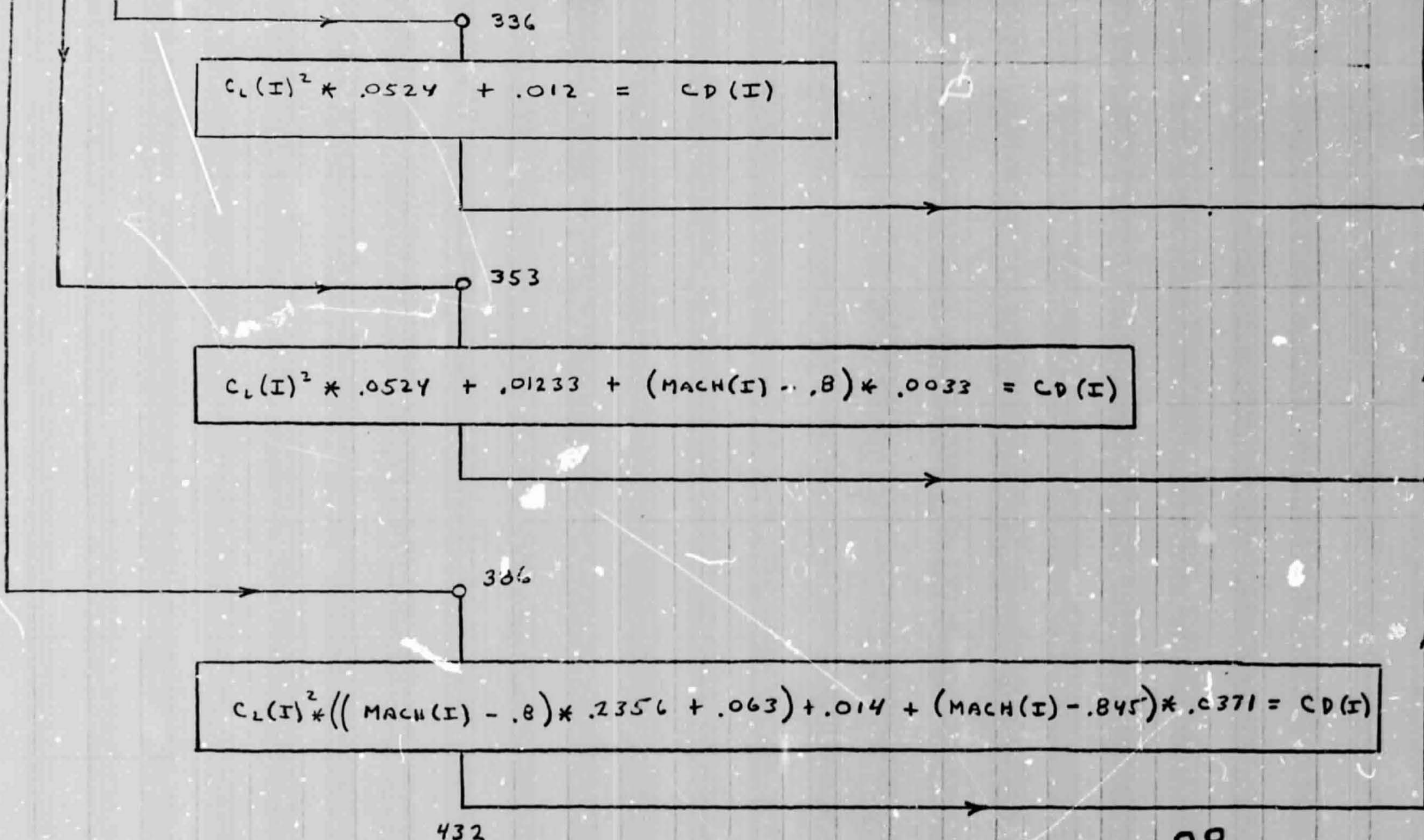
ORIGINAL PAGE IS OF POOR QUALITY


Flag 2 Set ?
no
yes (WP ALT. EVEN)
Lower Flag 2
151
DELTT(I) ÷ 2 = DELTT(I) → (15)
159
DELTT(I) + SUMT = SUMT → (16)
DELTT(I) * (TASFT(I) * COSG(I) + WIND) = DELTX(I)
DELTX (I) + SUM X = SUM X → (17)
TASFT(I) → (10)
TASFT (I-1) → (13)
Flag5Set ?
yes
(finished)
191
no
Return
190
SUMT / 60 Displayed XX.XX
R/S
SUM X / 6080 Displayed XXX.XX
R/S
Remove Fix

TI57 SHORT ALGORITHM (3)

$\dot{\gamma}$ out
$\dot{\omega}$ out
cos g (I) = .998 OR value at previous altitude




336
CL(I)² * .0524 + .012 = CD(I)
353
CL(I)² * .0524 + .01233 + (MACH(I) - .8) * .0033 = CD(I)
386
CL(I)² * ((MACH(I) - .8) * .2356 + .063) + .014 + (MACH(I) - .845) * .0371 = CD(I)
432

439,996 feet, whereas the TI59 estimates 441, 165 feet.

Two visiting students from the Centre d'Etudes et de Recherches de Toulouse M. Roger Dubois and M. Jean-Michel Loussant, will be working full time from April through June on the evaluation and refinement of the TI59 algorithm.

B. 707 Simulation Program Refinements

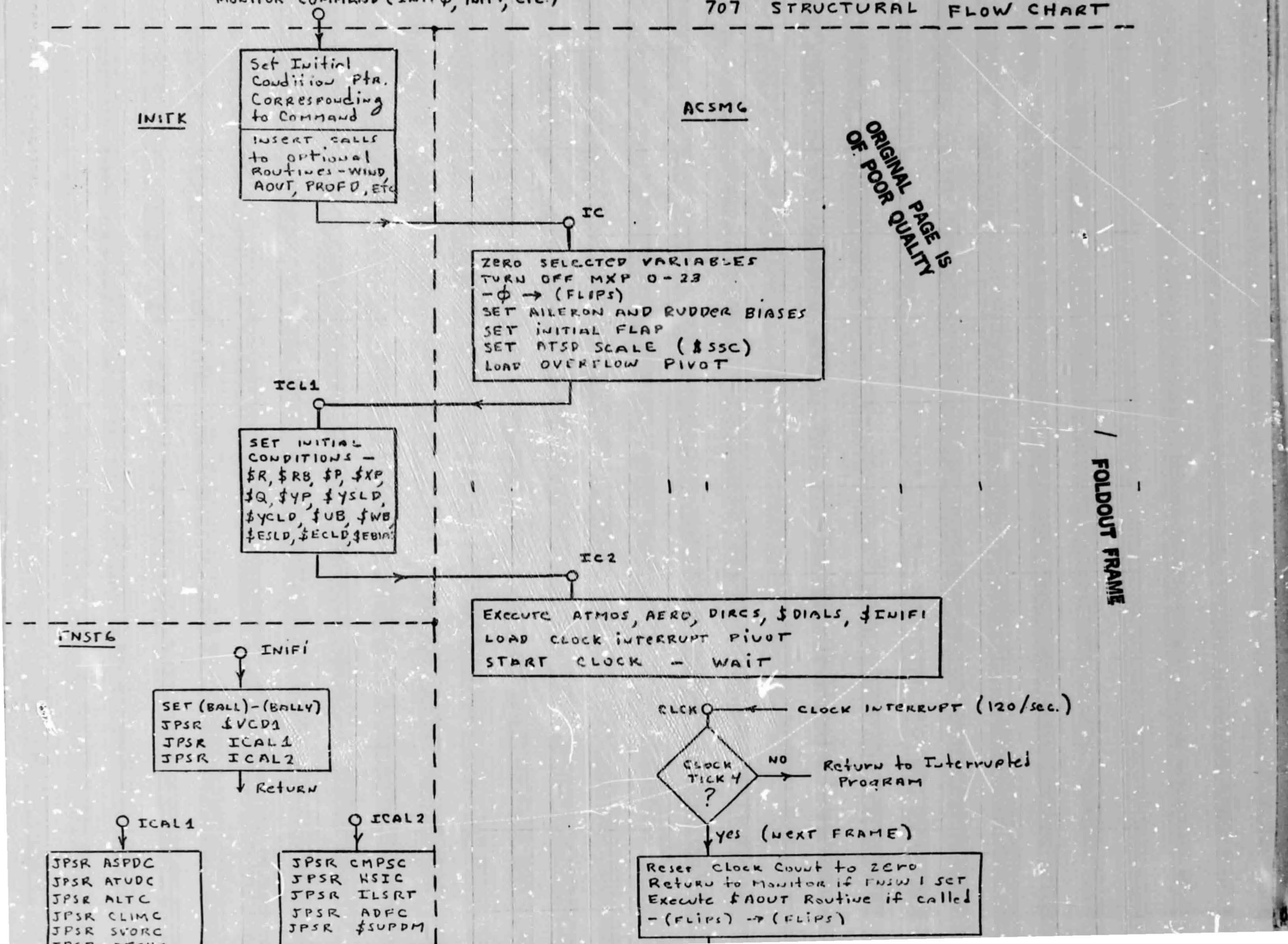
The decision to include open-loop descent tests in the research program was based on the need for an interim 4D RNAV technique that could be implemented in the near future. Open-loop testing, however, places much more severe requirements on the fidelity of the simulation model. Modeling errors that would be washed out in a closed-loop system, cause cumulative performance errors in an open-loop system that would completely mask the human factor effects we are attempting to measure. For this reason, a very significant effort has been made to correct errors and to improve the accuracy of the M.I.T. 707 simulation program. The original 707 program written by Captain Charles Corley (reference 1) has been thoroughly checked and over a hundred major modifications made to enhance its fidelity. A separate report on the current aircraft model and simulation program is now in preparation, so we will not go into great detail here. A structural flow chart for the new 707 program is given in Fig. 3. The following list identifies the most important changes:

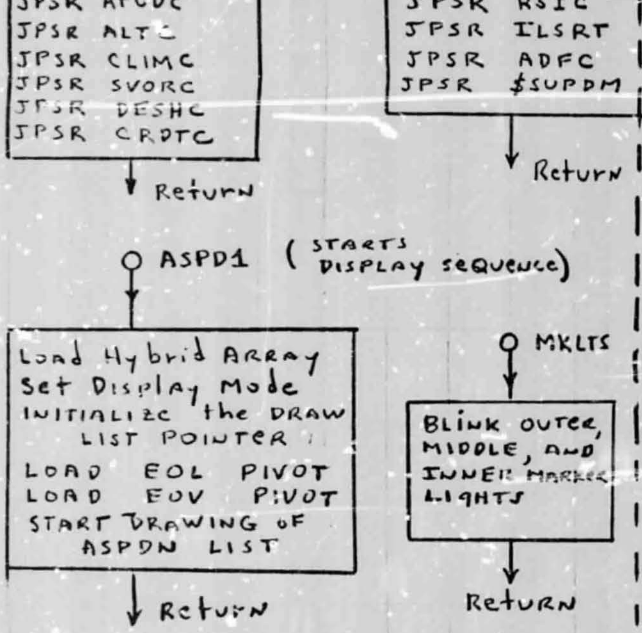
1. The roll moment equation was modified to include the interaction between inboard and outboard ailerons as a function of flap setting and the effects of spoiler blowdown.

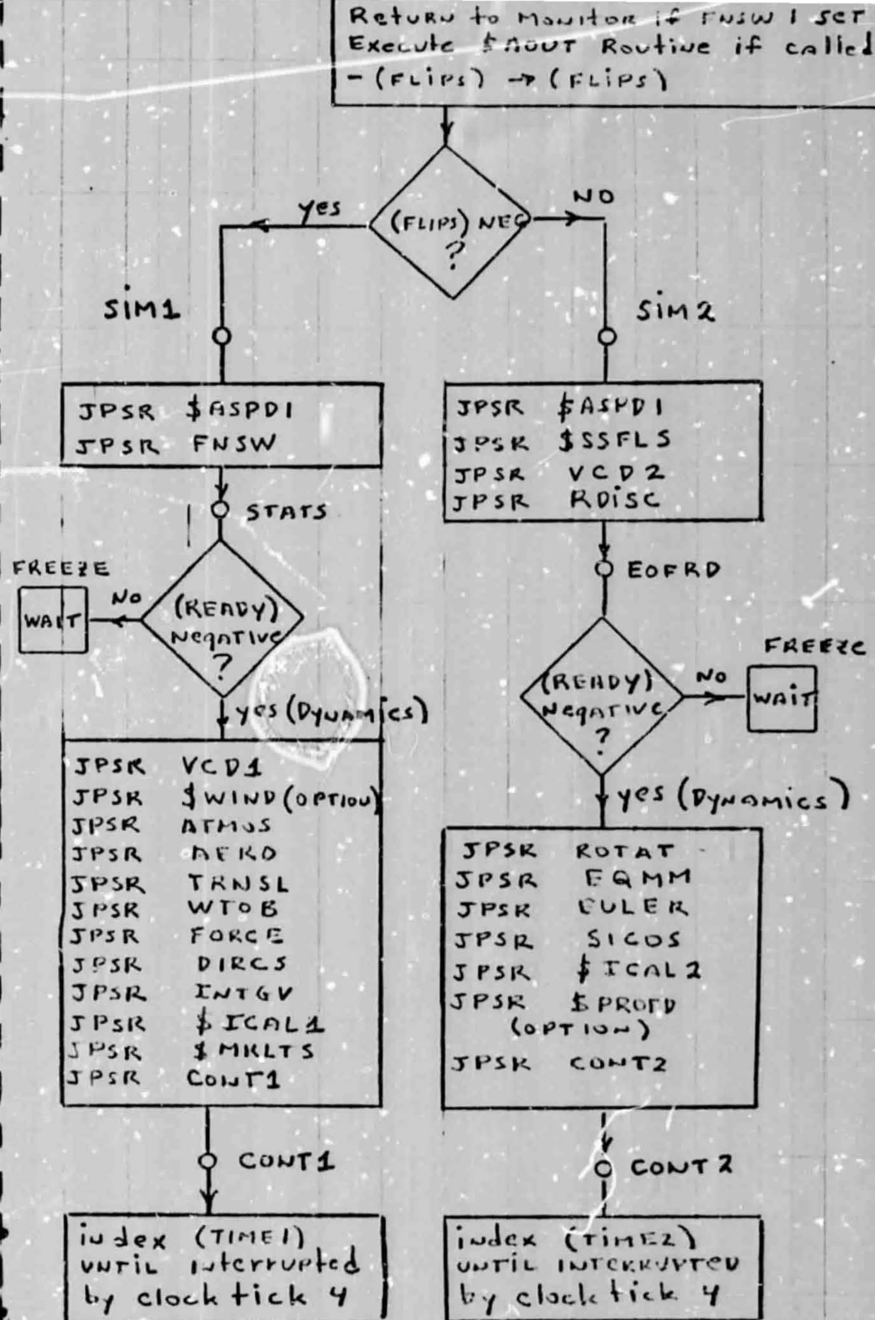
2. The airspeed dial now reads equivalent airspeed (EAS) instead of indicated airspeed (IAS) since the profile descents utilize constant EAS, which is a true indicator of dynamic pressure.

3. The alignment of the MACH dial and the EAS dial is now correct at the current airspeed needle position. The MACH dial range

FIGURE 3
707 STRUCTURAL FLOW CHART







The Draw Setup Routines are Called in the following order:

	Setup	Draw Lists
	ASPD1	ASPDN
	MACHI	LOMAC
	ATUD1	LIST2
	CIRCL	$LISTO
	ALT1	ALTN
	CLIM1	CLIMN
	HSI	HSIN
	CMPAS	$LIST5
	DESH	DHBUG
	CIRCL	$LISTO
	CMPS2	$LIST5
	CIRCL	$LISTO
	RMI	RMIH
	ADF	ADFN
	CORDT	LIST6
	ATUD2	$LIST7
CRGD6	DMERT	CLPO - CLPPL
	DMEEL	CLPO - CLPPL
MMAP6	PARLR	PARL
	ACTDR	ACTON
	MILDR	MILIS
	WITDR	WITMN
	LOGDR	LOGAN
	OMKDI	OMKI
	GARDR	VOR
	ALBDR	VOR
	PRODR	VOR
	[illegible]	VOR
	RING 1, 2, 3	$LISTO
	[illegible], MSEL	HBUG, SCLST

ORIGINAL PAGE IS OF POOR QUALITY

FOLDOUT FRAME 2

was reduced to .6 through .9.

4. The rotating drum on the altimeter (100's of feet) was replaced by an discretely updated digit.

5. The atmospheric variables and the instrument calculations are now updated every 1/15 seconds.

6. All the analog control inputs have been rescaled as well as the corresponding aerodynamic coefficients.

7. Errors in drag, the Euler angle routine SICOS, thrust, elevator limits, and several logic errors have been corrected.

8. The old piecewise-linear traffic generator has been removed to make way for the new traffic routine which will be based on stored trajectories.

9. Three new function switch modes have been created:

(a) Fly with position frozen (useful in establishing the steady-state flight conditions required by the initial condition routine INITK).

(b) Return to monitor leaving the real-time program in a clean condition so that it can be restarted at the point where the interruption occured.

(c) Execution of a single frame (used for debugging).

10. The steady-state elevator bias can now be set as an initial condition, thus avoiding a severe pitch transient when the program is started.

11. A more accurate square root routine has been written and the duplication of square root software in FNST and ACSM eliminated.

12. The trapezoidal integration of acceleration components has been rescaled to reduce the risk of overflow.

13. The initial condition routine, the wind model, analog outputs, and profile descent algorithm are now separate programs, no longer imbedded in the ACSM program. This was done to minimize the number of versions of the rather large program ACSM that must be stored on disk.

14. Several aerodynamic coefficients were changed and idle thrust was set to zero.

15. An interval clock count was added to provide the real-time program with an absolute time base for matching 4D RNAV schedules.

As a consequence of the revisions listed above, the current 707 simulation performs and handles much like the actual aircraft and meaningful open-loop testing can be carried out.

C. Advanced Integrated Display

John-Thones Amenyo continued his work during the report period on the design and programming of a single integrated display that would place ATSD information in the center of the pilot's scan field. This display is scheduled to be completed at the end of the spring term.

D. Presentations

(a) NASA-Langley Research Center - October 11, 1978

At the invitation of NASA Headquarters, a review of M.I.T.'s 4D RNAV-ATSD work was presented to a group of about 30 FAA and NASA staff members by Mark Connelly. Professor John Kreifeldt also made a presentation of his research on distributed management.

(b) FAA Engineering and Development Initiatives Process - August 17, 1978

A talk on the relationship between ATC capacity and ATSD-4D RNAV techniques was presented to the Airport Capacity Topic Group in Washington, D.C. by

Mark Connelly. Later in the year, at the request of the Chairman, Joseph Blatt, a report summarizing this talk was submitted for inclusion in the final report of the Initiatives Study.

(c) GENAVAC Meeting - February 27, 1979 - Washington, D.C.

A presentation was made to a group representing the major general aviation organizations by Mark Connelly. A review of the M.I.T. ATSD work was followed by a discussion period which focused on the need for greater ATC capacity to accomodate the growth of both the airlines and general aviation without restricting the efficiency or utility of either. Representatives of AOPA, NPA, NBAA, GAMA and HAA were present at the meeting.

(d) Channel 2 - Boston - April 20, 1979

The M.I.T. cockpit and ATSD display were included in an evening news special report on mid-air collisions.

References

ORIG
OF POOR QUALITY

1. Charles Corley, A Simulation Study of Time-Controlled Aircraft Navigation, M.S. Thesis, M.I.T. Department of Electrical Engineering, December 1974.

Appendix A

```
      PROGRAM  PROFILE
      IMPLICIT FRACTION(F)
      DIMENSION TAS(17),FMACH(17),EAS(17),TASFT(17),SING(17)
      DIMENSION DELTT(17),SUMT(17),DELTX(17),SUMX(17),GAMD(17)
      DIMENSION IWIND(15),WDOT(17),FSRSIG(15),SS(15),EASC(1)
      DIMENSION IFLAG(1),IN(1),IRD(17),COSG(17),IALT(17)
        DIMENSION FMACHC(1),DRAG(1),I(1)
        COMMON/X/EAS,GAMD,TASFT,COSG,FMACH,DRAG,I
      DATA IFLAG/0/,IN/0/,WDOT/17*0.0/
        DATA COSG/17*1.0/,GAMD/17*0.0/
      DATA IWIND/15*0.0/,FSRSIG/.8593F,.8325F,.8061F,.7802F,
        1.7548F,.7298F,.7053F,.6811F,.6575F,.6343F,.6115F,
        2.5891F,.5672F,.5442F,.5188F/
      DATA IALT/10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,0,0/
      DATA SS/734.2,728.8,723.3,717.8,712.2,706.6,700.9,695.2,
        1689.5,683.7,677.8,671.9,666.0,662.0,662.0/
      DATA EASC/320./,FMACHC/.83F/
      GLOBAL EASC,FMACHC
1       I=0
        DO 60 K=16,17
        TAS(K)=0
      FMACH(K)=0.0F
      EAS(K)=0.
        TASFT(K)=0.
        SING(K)=0.
60      CONTINUE
        DO 70 K=15,17
        IRD(K)=0
        DELTT(K)=0.
        SUMT(K)=0.
        DELTX(K)=0.
        SUMX(K)=0.
70      CONTINUE
2       I=I+1
        EAS(I)=EASC
        TAS(I)=EASC/FSRSIG(I)
        FMACH(I)=TAS(I)/(SS(I)*.8684F)
        TASFT(I)=(TAS(I)*6080.3)/3600.
        IF (I-15) 5,6
5       IF (FMACH(I)-FMACHC) 2,3,3
```

```
3     EAS(16)=EAS(I)
      TAS(16)=TAS(I)
      FMACH(16)=FMACH(I)
      TASFT(16)=TASFT(I)
      IFLAG=I-1
      FMACH(17)=FMACHC
      TAS(17)=FMACH(17)*SS(I-1)*.8684F
      TASFT(17)=(TAS(17)*6080.3)/3600.
      EAS(17)=TAS(17)*FSRSIG(I-1)
4     FMACH(I)=FMACHC
      TAS(I)=FMACH(I)*SS(I)*.8684F
      TASFT(I)=(TAS(I)*6080.3)/3600.
      EAS(I)=TAS(I)*FSRSIG(I)
      I=I+1
      IF (I-15) 4,4,6
6     I=0
7     I=I+1
      IF (I-IFLAG) 8,11
8     CALL DRAGX
      IF (I-15) 12,13
12    SING(I)=DRAG/((6994.)*(((TASFT(I)-TASFT(I+1))/2000.)*
     1TASFT(I)-32.17))
      COSG(I)=1-SING(I)*SING(I)/2.
      DESR=-TASFT(I)*SING(I)
      DELTT(I)=2000./DESR
      DELTX(I)=(TASFT(I)*COSG(I)+IWIND(I))*DELTT(I)
      IRD(I)=DESR*60.
      IF(I.GT.1) GAMD(I-1)=(SING(I-1)-SING(I))/DELTT(I-1)
      GO TO 7
```

```
 11   CALL DRAGX
      SING(I)=DRAG/((6994.)*(((TASFT(I)-TASFT(16))/2000.)*
     1TASFT(I)-32.17))
      COSG(I)=1-SING(I)*SING(I)/2
      DESR=-TASFT(I)*SING(I)
      DELTT(I)=2000./DESR
      IRD(I)=DESR*60.
      GAMD(I-1)=(SING(I-1)-SING(I))/DELTT(I-1)
      I=16
      CALL DRAGX
      I=IFLAG
      SING(16)=-DRAG/225000.
      GAMD(I)=(SING(I)-SING(16))/DELTT(I)
      I=17
      CALL DRAGX
      I=IFLAG+1
      SING(17)=DRAG/((6994.)*(((TASFT(17)-TASFT(I))/2000.)*
     1TASFT(17)-32.17))
      DESR=-TASFT(17)*SING(17)
      DELTT(17)=2000./DESR
      CALL DRAGX
      SING(I)=DRAG/((6994.)*(((TASFT(I)-TASFT(I+1))/2000.)*
     1TASFT(I)-32.17))
      COSG(I)=1-SING(I)*SING(I)/2.
      DESR=-TASFT(I)*SING(I)
      DELTT(I)=2000./DESR
      IRD(I)=DESR*60.
      GAMD(17)=(SING(17)-SING(I))/DELTT(17)
      HTMHK=2000.*(TAS(17)-TAS(I-1))/(TAS(16)-TAS(I-1)+TAS(17)-
     1TAS(I))
      DELTT(I-1)=-(HTMHK+1000.)/(TASFT(I-1)*SING(I-1))
      DELTX(I-1)=(TASFT(I-1)*COSG(I-1)+WIND(I-1))*DELTT(I-1)
      TEMP=-(3000.-HTMHK)/(TASFT(I)*SING(I))
      DELTX(I)=(TASFT(I)*COSG(I)+WIND(I))*TEMP
      GO TO 7
 13   SING(I)=-DRAG/225000.
      GAMD(I-1)=(SING(I-1)-SING(I))/DELTT(I-1)
      IF (IFLAG.GT.0) DELTT(IFLAG+1)=TEMP
      IN=IN+1
      IF (IN-2) 6,14
 14   SUMT(1)=DELTT(1)/2
      DO 15 K=2,13
      SUMT(K)=SUMT(K-1)+DELTT(K)
 15   CONTINUE
      SUMT(14)=SUMT(13)+DELTT(14)/2
      SUMX(1)=DELTX(1)/2
      DO 16 K=2,13
      SUMX(K)=SUMX(K-1)+DELTX(K)
 16   CONTINUE
      SUMX(14)=SUMX(13)+DELTX(14)/2
```

```
      WRITE (10,20)
      K=1
   21 WRITE (10,30) IALT(K),FMACH(K),EAS(K),TAS(K),TASFT(K),
     1SING(K),COSG(K),IRD(K),GAMD(K)
      IF (K.EQ.17) GO TO 23
      K=K+1
      GO TO 21
   23 WRITE (10,40)
      K=1
   24 WRITE(10,50) IALT(K),DELTT(K),SUMT(K),DELTX(K),SUMX(K)
      IF (K.EQ.17) GO TO 26
      K=K+1
      GO TO 24
   30 FORMAT(I3,F8.4,3F8.1,2F9.4,I7,E12.3)
   50 FORMAT(I3,2X,F7.2,F9.1,I8,I10)
   20 FORMAT(1X,3HALT,2X,5HFMACH,4X,3HEAS,5X,3HTAS,4X,5HTASFT,
     14X,4HSING,5X,4HCOSG,4X,3HR/D,6X,4HGAMD)
   40 FORMAT(////1X,3HALT,3X,5HDELTT,5X,4HSUMT,3X,5HDELTX,5X,
     14HSUMX)
   26 EXIT
      END
```

```
      SUBROUTINE DRAGX
      IMPLICIT FRACTION(F)
      COMMON/X/EAS(17),GAMD(17),TASFT(17),COSG(17),
     1FMACH(17),DRAG(1),I(1)
      QS=10.208*EAS(I)*EAS(I)
      CL=(GAMD(I)*TASFT(I)+32.17*COSG(I))*6994./QS
      IF (FMACH(I)-.7) 1,1,2
    1 CD=.012+.0524*CL*CL
      GO TO 7
    2 IF (FMACH(I)-.8) 3,3,4
    3 CD=.01233+.0033*(FMACH(I)-.8)+.0524*CL*CL
      GO TO 7
    4 IF (FMACH(I)-.845) 5,5,6
    5 CD=.014+.0371*(FMACH(I)-.845)+(.063+.2356*(FMACH(I)-.8))*
     1CL*CL
      GO TO 7
    6 CD=.014+.1455*(FMACH(I)-.845)+(.063+.8333*(FMACH(I)-
     1.845))*CL*CL
    7 DRAG=QS*CD
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

STANDARD Descent (.83 Mach/320 EAS)

```
PROFILWLISTV(0,1)
RESET(6,001)
PROFILE
```

ALT	FMACH	EAS	TAS	TASFT	SING	COSG	R/D	GAMD
10	0.5841	320.0	372.4	629.0	-0.0559	0.9984	2110	-0.133E-04
12	0.6074	320.0	384.4	649.2	-0.0552	0.9985	2148	-0.144E-04
14	0.6320	320.0	397.0	670.5	-0.0544	0.9985	2186	-0.148E-04
16	0.6580	320.0	410.2	692.8	-0.0536	0.9986	2225	-0.169E-04
18	0.6855	320.0	424.0	716.1	-0.0526	0.9986	2261	-0.151E-04
20	0.7146	320.0	438.5	740.6	-0.0518	0.9987	2303	-0.137E-04
22	0.7455	320.0	453.7	766.3	-0.0511	0.9987	2350	-0.127E-04
24	0.7783	320.0	469.8	793.5	-0.0505	0.9987	2403	0.619E-04
26	0.8129	320.0	486.7	822.0	-0.0536	0.9986	2642	0.640E-03
28	0.8300	312.5	492.7	832.2	-0.0828	0.9966	4136	-0.141E-03
30	0.8300	298.7	488.5	825.0	-0.0786	0.9969	3898	-0.116E-03
32	0.8300	285.2	484.2	817.9	-0.0752	0.9972	3690	-0.159E-03
34	0.8300	272.2	480.0	810.7	-0.0700	0.9975	3406	-0.175E-03
36	0.8300	259.6	477.1	805.8	-0.0639	0.9980	3088	-0.338E-04
38	0.8300	247.5	477.1	805.8	-0.0626	1.0000	0	0.000E+00
0	0.8497	320.0	504.5	852.1	-0.0826	1.0000	0	0.000E+00
0	0.8300	326.7	496.9	839.3	-0.0875	1.0000	0	-0.171E-03

ALT	DELTT	SUMT	DELTX	SUMX
10	56.86	28.4	35709	17854
12	55.84	84.3	36199	54053
14	54.87	139.1	36737	90791
16	53.91	193.1	37294	128085
18	53.06	246.1	37941	166027
20	52.09	298.2	38527	204554
22	51.04	349.3	39066	243620
24	49.93	399.2	39568	283189
26	43.79	443.0	35945	319134
28	30.05	473.0	24918	344053
30	30.78	503.8	25315	369368
32	32.52	536.3	26520	395888
34	35.22	571.5	28485	424373
36	38.85	591.0	31244	439996
38	0.00	0.0	0	0
0	0.00	0.0	0	0
0	27.24	0.0	0	0

```
LISTA(1)
LISTV(0,1)
REMOVE(0,1,((11,13)))
COPY(11,11,112,001)
SFORT(001,11,001)
```

```
START(12,001(001,007))
OPEN(EASC,"0")
2114040000S
PROFILE
```

Intermediate Descent (260 KEAS)

ALT	FMACH	EAS	TAS	TASFT	SING	COSG	R/D	GAMD
10	0.4745	260.0	302.6	511.1	-0.0476	0.9989	1460	-0.549E-05
12	0.4935	260.0	312.3	527.5	-0.0472	0.9989	1493	-0.604E-05
14	0.5135	260.0	322.5	544.8	-0.0467	0.9989	1526	-0.627E-05
16	0.5347	260.0	333.3	562.9	-0.0462	0.9989	1560	-0.728E-05
18	0.5570	260.0	344.5	581.8	-0.0456	0.9990	1593	-0.801E-05
20	0.5806	260.0	356.3	601.7	-0.0450	0.9990	1625	-0.907E-05
22	0.6057	260.0	368.7	622.7	-0.0444	0.9990	1657	-0.909E-05
24	0.6323	260.0	381.7	644.7	-0.0437	0.9990	1690	-0.105E-04
26	0.6605	260.0	395.5	667.9	-0.0430	0.9991	1721	-0.118E-04
28	0.6904	260.0	409.9	692.3	-0.0421	0.9991	1750	-0.100F-04
30	0.7224	260.0	425.2	718.2	-0.0414	0.9991	1785	-0.919E-05
32	0.7565	260.0	441.4	745.5	-0.0408	0.9992	1826	-0.196E-04
34	0.7926	260.0	458.4	774.2	-0.0395	0.9992	1836	0.379E-03
36	0.8300	259.6	477.1	805.8	-0.0639	0.9980	3088	-0.338E-04
38	0.8300	247.5	477.1	805.8	-0.0626	1.0000	0	0.000E+00
0	0.8311	260.0	477.8	806.9	-0.0643	1.0000	0	0.000E+00
0	0.8300	272.2	480.0	810.7	-0.0701	1.0000	0	-0.178E-03

ALT	DELTT	SUMT	DELTX	SUMX
10	82.17	41.1	41947	20973
12	80.37	121.5	42349	63323
14	78.63	200.1	42790	106114
16	76.92	277.0	43248	149362
18	75.33	352.3	43779	193142
20	73.81	426.1	44367	237509
22	72.40	498.5	45038	282548
24	70.98	569.5	45718	328266
26	69.71	639.2	46514	374781
28	68.56	707.8	47426	422207
30	67.19	775.0	48215	470423
32	65.71	840.7	48945	519368
34	96.01	936.7	74277	593646
36	20.61	947.0	16573	601933
38	0.00	0.0	0	0
0	0.00	0.0	0	0
0	35.17	0.0	0	0

SLOW DESCENT (200 KEAS)

```
START(2,(001,007)))
OPEN(EASC,"U")
2106200000
PROFILE
```

ALT	FMACH	EAS	TAS	TASFT	SING	COSG	R/D	GAMD
10	0.3651	200.0	232.8	393.1	-0.0470	0.9989	1107	-0.255E-05
12	0.3796	200.0	240.3	405.8	-0.0467	0.9989	1136	-0.285E-05
14	0.3950	200.0	248.1	419.1	-0.0464	0.9989	1166	-0.298E-05
16	0.4113	200.0	256.4	433.0	-0.0461	0.9989	1196	-0.351E-05
18	0.4285	200.0	265.0	447.6	-0.0457	0.9990	1227	-0.392E-05
20	0.4466	200.0	274.0	462.9	-0.0453	0.9990	1259	-0.451E-05
22	0.4659	200.0	283.6	479.0	-0.0449	0.9990	1290	-0.459E-05
24	0.4864	200.0	293.6	496.0	-0.0445	0.9990	1323	-0.540E-05
26	0.5081	200.0	304.2	513.8	-0.0440	0.9990	1356	-0.621E-05
28	0.5311	200.0	315.3	532.6	-0.0434	0.9991	1388	-0.672E-05
30	0.5557	200.0	327.1	552.4	-0.0429	0.9991	1420	-0.707E-05
32	0.5819	200.0	339.5	573.5	-0.0423	0.9991	1454	-0.151E-04
34	0.6097	200.0	352.6	595.5	-0.0410	0.9992	1465	-0.215E-04
36	0.6393	200.0	367.5	620.7	-0.0393	0.9992	1462	0.139E-03
38	0.6706	200.0	385.5	651.1	-0.0507	1.0000	0	0.000E+00
0	0.0000	0.0	0.0	0.0	0.0000	1.0000	0	0.000E+00
0	0.0000	0.0	0.0	0.0	0.0000	1.0000	0	0.000E+00

ALT	DELTT	SUMT	DELTX	SUMX
10	108.35	54.2	42546	21273
12	105.59	159.8	42799	64072
14	102.91	262.7	43077	107150
16	100.25	362.9	43364	150515
18	97.74	460.7	43699	194214
20	95.31	556.0	44069	238284
22	92.99	649.0	44491	282776
24	90.66	739.6	44919	327695
26	88.49	828.1	45420	373116
28	86.45	914.6	45995	419112
30	84.47	999.0	46620	465732
32	82.52	1081.6	47280	513013
34	81.87	1163.4	48719	561732
36	82.08	1204.5	30907	587186
38	0.00	0.0	0	0
0	0.00	0.0	0	0
0	0.00	0.0	0	0

Appendix B

TI 59 Descent Program

New Short Form w/o Printout

```
000  69  OP
001  26   26
002  87  IFF
003  01   01
004  00   00
005  55   55
006  73  RC*
007  06   06
008  22  INV
009  59  INT
010  35  1/X
011  65   ×
012  43  RCL
013  01   01
014  42  STO
015  14   14
016  65   ×
017  01   1
018  93   .
019  06   6
020  08   8
021  09   9
022  95   =
023  42  STO
024  13   13
025  55   ÷
026  53   (
027  73  RC*
028  06   06
029  59  INT
030  55   ÷
031  01   1
032  00   0
033  54   )
034  95   =
035  42  STO
036  12   12
037  32  X:T
038  43  RCL
039  00   00
040  32  X:T
041  22  INV
042  77   GE
043  00   00
044  85   85
045  86  STF
046  01   01
047  61  GTO
048  00   00
049  55   55
050  68  NOP
```

ORIGINAL PAGE IS
OF POOR QUALITY

051	68	NOP
052	61	GTO
053	00	00
054	85	85
055	73	RC*
056	06	06
057	59	INT
058	55	÷
059	01	1
060	00	0
061	65	×
062	43	RCL
063	00	00
064	42	STO
065	12	12
066	95	=
067	42	STO
068	13	13
069	55	÷
070	01	1
071	93	.
072	06	6
073	08	8
074	09	9
075	65	×
076	53	(
077	73	RC*
078	06	06
079	22	INV
080	59	INT
081	54	)
082	95	=
083	42	STO
084	14	14
085	87	IFF
086	00	00
087	00	00
088	98	99
089	43	RCL
090	13	13
091	42	STO
092	10	10
093	86	STF
094	00	00
095	61	GTO
096	00	00
097	00	00
098	71	SBR
099	02	02
100	27	27

101	71	SBR
102	04	04
103	34	34
104	71	SBR
105	01	01
106	13	13
107	61	GTO
108	00	00
109	00	00
110	00	0
111	00	0
112	00	0
113	43	RCL
114	13	13
115	65	×
116	43	RCL
117	09	09
118	55	÷
119	02	2
120	00	0
121	00	0
122	00	0
123	95	=
124	35	1/X
125	94	+/-
126	42	STO
127	15	15
128	43	RCL
129	03	03
130	32	X:T
131	43	RCL
132	06	06
133	22	INV
134	67	EQ
135	01	01
136	43	43
137	86	STF
138	05	05
139	87	IFF
140	04	04
141	01	01
142	51	51
143	22	INV
144	87	IFF
145	02	02
146	01	01
147	59	59
148	22	INV
149	86	STF
150	02	02

```
151  43  RCL
152  15   15
153  65   ×
154  93   .
155  05   5
156  95   =
157  42  STO
158  15   15
159  43  RCL
160  15   15
161  44  SUM
162  16   16
163  68  NOP
164  65   ×
165  53   (
166  43  RCL
167  18   18
168  65   ×
169  43  RCL
170  13   13
171  85   +
172  43  RCL
173  02   02
174  54   )
175  95   =
176  44  SUM
177  17   17
178  68  NOP
179  43  RCL
180  13   13
181  48  EXC
182  10   10
183  42  STO
184  13   13
185  87  IFF
186  05   05
187  01   01
188  91   91
189  68  NOP
190  92  RTN
191  43  RCL
192  16   16
193  55   ÷
194  06   6
195  00   0
196  95   =
197  58  FIX
198  02   02
199  91  R/S
200  43  RCL
```

201	17	17
202	55	÷
203	06	6
204	00	0
205	08	8
206	00	0
207	95	=
208	91	R/S
209	22	INV
210	58	FIX
211	91	R/S
212	00	0
213	00	0
214	00	0
215	00	0
216	00	0
217	00	0
218	00	0
219	00	0
220	00	0
221	00	0
222	00	0
223	00	0
224	00	0
225	00	0
226	00	0
227	43	RCL
228	14	14
229	68	NOP
230	33	X^2
231	65	×
232	01	1
233	00	0
234	93	.
235	02	2
236	01	1
237	95	=
238	42	STO
239	15	15
240	35	1/X
241	65	×
242	06	6
243	09	9
244	09	9
245	04	4
246	65	×
247	68	NOP
248	68	NOP
249	03	3
250	02	2

251	93	.
252	01	1
253	07	7
254	65	×
255	43	RCL
256	18	18
257	68	NOP
258	95	=
259	33	X²
260	42	STO
261	08	08
262	43	RCL
263	13	13
264	68	NOP
265	43	RCL
266	12	12
267	68	NOP
268	32	X⇌T
269	93	.
270	07	7
271	77	GE
272	03	03
273	36	36
274	93	.
275	08	8
276	77	GE
277	03	03
278	53	53
279	93	.
280	08	8
281	04	4
282	05	5
283	77	GE
284	03	03
285	86	86
286	32	X⇌T
287	75	-
288	93	.
289	08	8
290	04	4
291	05	5
292	95	=
293	42	STO
294	04	04
295	65	×
296	93	.
297	08	8
298	03	3
299	03	3
300	03	3

301	85	+
302	93	.
303	00	0
304	06	6
305	03	3
306	95	=
307	65	×
308	43	RCL
309	08	08
310	98	ADV
311	48	EXC
312	04	04
313	65	×
314	93	.
315	01	1
316	04	4
317	05	5
318	05	5
319	85	+
320	93	.
321	00	0
322	01	1
323	04	4
324	85	+
325	43	RCL
326	04	04
327	95	=
328	65	×
329	43	RCL
330	15	15
331	95	=
332	42	STO
333	04	04
334	68	NOP
335	92	RTN
336	43	RCL
337	08	08
338	65	×
339	93	.
340	00	0
341	05	5
342	02	2
343	04	4
344	85	+
345	93	.
346	00	0
347	01	1
348	02	2
349	95	=
350	61	GTO

351	03	03
352	28	28
353	43	RCL
354	08	08
355	65	×
356	93	.
357	00	0
358	05	5
359	02	2
360	04	4
361	85	+
362	93	.
363	00	0
364	01	1
365	02	2
366	03	3
367	03	3
368	85	+
369	53	(
370	32	X:T
371	75	−
372	93	.
373	08	8
374	54	)
375	65	×
376	93	.
377	00	0
378	00	0
379	03	3
380	03	3
381	68	NOP
382	95	=
383	61	GTO
384	03	03
385	28	28
386	43	RCL
387	08	08
388	65	×
389	53	(
390	53	(
391	32	X:T
392	75	−
393	93	.
394	08	8
395	54	)
396	65	×
397	93	.
398	02	2
399	03	3
400	05	5

401	06	6
402	85	+
403	93	.
404	00	0
405	06	6
406	03	3
407	54	)
408	85	+
409	93	.
410	00	0
411	01	1
412	04	4
413	85	+
414	53	(
415	43	RCL
416	12	12
417	75	-
418	93	.
419	08	8
420	04	4
421	05	5
422	54	)
423	65	×
424	93	.
425	00	0
426	03	3
427	07	7
428	01	1
429	95	=
430	61	GTO
431	03	03
432	28	28
433	00	0
434	43	RCL
435	10	10
436	75	-
437	43	RCL
438	13	13
439	95	=
440	55	÷
441	02	2
442	00	0
443	00	0
444	00	0
445	65	×
446	43	RCL
447	13	13
448	75	-
449	03	3
450	02	2

451	93	.
452	01	1
453	07	7
454	95	=
455	65	×
456	06	6
457	09	9
458	09	9
459	04	4
460	95	=
461	35	1/X
462	65	×
463	43	RCL
464	04	04
465	95	=
466	42	STO
467	09	09
468	68	NOP
469	33	X²
470	55	÷
471	02	2
472	75	-
473	01	1
474	95	=
475	94	+/-
476	42	STO
477	18	18
478	92	RTN
479	00	0
480	00	0
481	00	0
482	00	0
483	00	0
484	00	0
485	00	0
486	00	0
487	68	NOP
488	68	NOP
489	68	NOP
490	68	NOP
491	68	NOP
492	68	NOP
493	68	NOP
494	68	NOP
495	68	NOP
496	68	NOP
497	68	NOP
498	68	NOP
499	68	NOP
500	00	0

```
545  76 LBL
546  11  A
547  43 RCL
548  02  02
549  65  x
550  01  1
551  93  .
552  06  6
553  08  8
554  09  9
555  95  =
556  42 STO
557  02  02
558  00  0
559  32 X:T
560  43 RCL
561  04  04
562  67  EQ
563  06  06
564  15  15
565  55  ÷
566  02  2
567  95  =
568  42 STO
569  06  06
570  22 INV
571  59 INT
572  22 INV
573  67  EQ
574  05  05
575  78  78
576  86 STF
577  02  02
578  85  +
579  43 RCL
580  06  06
581  85  +
582  01  1
583  07  7
584  95  =
585  42 STO
586  06  06
587  43 RCL
588  03  03
589  55  ÷
590  02  2
591  95  =
592  42 STO
593  03  03
594  22 INV
595  59 INT
596  22 INV
597  67  EQ
598  06  06
599  02  02
600  86 STF
```

601	04	04
602	75	-
603	43	RCL
604	03	03
605	75	-
606	01	1
607	09	9
608	95	=
609	94	+/-
610	42	STO
611	03	03
612	61	GTO
613	00	00
614	00	00
615	43	RCL
616	01	01
617	65	×
618	01	1
619	93	.
620	06	6
621	04	4
622	01	1
623	95	=
624	42	STO
625	10	10
626	01	1
627	08	8
628	42	STO
629	06	06
630	86	STF
631	02	02
632	86	STF
633	00	00
634	61	GTO
635	00	~~00~~ 05
636	00	~~00~~ 97
637	00	0
638	00	0
639	00	0

0.83	00	
320.	01	
0.	02	
36.	03	
10.	04	
0.	05	
0.	06	
0.	07	
0.	08	
0.	09	
0.	10	
0.	11	
0.	12	
0.	13	
0.	14	
0.	15	
0.	16	
0.	17	
0.998	18	
11160.9999	19	SL
11082.9709	20	2
11004.9424	21	4
10927.9142	22	6
10848.8865	23	8
10769.8593	24	10
10690.8[illegible]25	25	12
10609.8061	26	14
10528.7802	27	16
10446.7548	28	18
10364.7298	29	20
10281.7053	30	22
10196.6811	31	24
10114.6575	32	26
10028.6343	33	28
9941.6115	34	30
9855.5891	35	32
9769.5672	36	34
9710.5442	37	36
9709.5188	38	38
9709.4946	39	40