

Digital Autopilot Docked Jet Selection

DKJSLECT

Set TIME5 to cause program interrupt #2 in 6 centi-seconds

If DAPZRUPT > 0:

Proceed to "BAILOUT" (pattern 32000_g)

DAPARUPT = ARUPT_{dp}

DAPBQRPT = (BRUPT, QRUPT)

BRUPT = Instruction stored at location SUPERJOB

DAPZRUPT_{dp} = ZRUPT_{dp}

ZRUPT = Address of SUPERJOB + 1 (S register portion)

Resume (Causes the "SUPERJOB" "jask" to be entered promptly while allowing other interrupts, e.g. telemetry, to occur)

SUPERJOB

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 14(Free Mode complement) of TS = 0:

If INHIBIT > 0: (Tag here "YZFREE")

Proceed to "BURBLE"

If INHIBIT = +0:

TS₁ = - (channel 31) (gives "true" values of translation hand controller bits)

TS = (bits 10-9 of TS₁), shifted right 8 places (\pm Y)

TAUY = K_{200mst6}_{TS-1}

TS = (bits 12-11 of TS₁), shifted right 10 places (\pm Z)

TAUZ = K_{200mst6}_{TS-1}

Proceed to "NOBYPASS"

If INHIBIT > 0: (Tag here "INHIBCHK")

Proceed to "BURBLE"

Proceed to "NOBYPASS"

NOBYPASS

TRCOMPAC = 0

TRCOMPBD = 0

5PT = 0

5YWT = 0

61PT = 0

61YWT = 0

62PT = 0

62YWT = 0

DELFLG = 0

Proceed to "P/YWCOMP"

P/YWCOMP

i = 1 (in SPNDX cell, means yaw; 0 for pitch)

If $\text{TAU}_{1_i} = 0$: (rotation command zero)

If $\text{TAU}_{1_i} = 0$: (Tag here "YZCHECK")

Proceed to "SPNDXCK"

Proceed to "FORCE"

If bit K_{pck_i} of DAPDATR3 = 1: (bit 4 for i = 0; bit 1 for i = 1)

Proceed to "FORCE"

Proceed to "COUPLE"

COUPLE

TS = 0

If $\text{TAU}_{1_i} < 0$:

TS = 1

TS = TS + 2i (odd for negative command)

$\text{TS}_1 = K_{\text{ppitch}c_{\text{TS}}}$ (jets to implement command)

$5\text{PW}_1 = \text{TS}_1$ (for i = 1, load 5YWW)

If $TS_1 \wedge CH5FAIL = 0$: (i.e. selected jets not failed)

Proceed to "GOODA"

If bit K_{pck_i} of DAPDATR3 = 1: (bit 4 for $i = 0$; bit 1 for $i = 1$)

Proceed to "P/YWALRM"

Proceed to "FORCE"

FORCE

$TS = 0$

If $TAUZ_i < 0$:

$TS = 1$

$TS = TS + 2i$

$TS_1 = K_{pzforce_{TS}}$ (jets to implement command)

$62PW_i = TS_1$

If $TS_1 \wedge CH6FAIL = 0$: (i.e. selected jets not failed)

Proceed to "GOODB"

$TS_2 = TS_1 \wedge (-CH6FAIL)$

If $TS_2 = 0$: (i.e. all jets bad)

$62PT_i = 0$ (for $i = 1$, loads 62YWT) (Tag here "BADC")

If bit K_{pck_i} of DAPDATR3 = 1: (bit 4 for $i = 0$; bit 1 for $i = 1$)

Proceed to "COUPLE"

Proceed to "P/YWALRM"

$61PW_i = TS_2$ (for $i = 1$, loads 61YWW) (Tag here "GOODC")

If $TS_2 \wedge K_{prollbd_{2i}} = 0$:

Proceed to "NEGRDIST"

If $TAUZ_i > 0$: (for $i = 1$, checks TAUY)

$DELFLG = DELFLG + \text{bit}(8 - i)$ (bit 8 set 1 for $i = 0$; bit 7 for $i = 1$)

$61PT_i = |TAUZ_i|$ (for $i = 1$, loads 61YWT)

$TS = -|TAUZ_i|$

Proceed to "TRCOMP"

NEGRDIST

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (6 - i)$$

If $\text{TAUZ}_i < 0$:

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (8 - i)$$

$$61\text{PT}_i = |\text{TAUZ}_i|$$

$$\text{TS} = |\text{TAUZ}_i|$$

Proceed to "TRCOMP"

TRCOMP

$$\text{TS}_2 = C_{\text{alphap}_i} \text{TS}$$

If bit $(8 - i)$ of DELFLG = 0:

$$\text{TRCOMPAC}_i = \text{TRCOMPAC}_i + \text{TS} + \text{TS}_2 \quad (\text{for } i = 1, \text{ uses TRCOMPBD})$$

If bit $(8 - i)$ of DELFIG = 1:

$$\text{TRCOMPAC}_i = \text{TRCOMPAC}_i + \text{TS} - \text{TS}_2$$

If $\text{TAU}_i < 0$:

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (4 - i)$$

Proceed to "SPNDXCK"

GOODA

$$\text{TS} = \frac{1}{2} \text{TAU}_i \quad (\frac{1}{2} \text{ factor because of 2 jets})$$

If $\text{TS} < 0$:

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (12 - i)$$

$$5\text{PT}_i = |\text{TS}| \quad (\text{for } i = 1, \text{ loads 5YWT})$$

Proceed to "SPNDXCK"

GOODB

$$62\text{PT}_i = \left| \frac{1}{2} \text{TAUZ}_i \right| \quad (\text{for } i = 1, \text{ loads 62YWT with TAU}_i \text{ information})$$

If $\text{TAU}_i < 0$:

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (12 - i) + \text{bit } (4 - i)$$

If $\text{TAUZ}_i > 0$: (for $i = 1$, checks TAU_Y)

$$\text{DELFIG} = \text{DELFLG} + \text{bit } (10 - i)$$

$$\text{TRCOMPBD}_{-i} = \text{TRCOMPBD}_{-i} + C_{\text{alphap}_i} \text{TAUZ}_i \quad (\text{for } i = 1, \text{ uses TRCOMPAC})$$

Proceed to "SPNDXCK"

P/YWALRM Entered from "COUPLE" and "FORCE"

If bit 12(500**FLG) of FLAGWRD3 = 0: (set 0 e.g. by "DONOUN89")

Perform "ALARM" (pattern 0500_g)

Set bit 12(500**FLG) of FLAGWRD3 = 1

Proceed to "SPNDXCK"

SPNDXCK

If $i > 0$:

$i = i - 1$

Proceed to second line of "P/YWCOMP"

If TRCOMPAC \gg 0: (Tag here "ROLLCOMP")

If TRCOMPBD \leq 0: (Tag here "TCOMPAC+")

$TS = TRCOMPBD + TRCOMPAC$ (opposite signs, set larger to the sum and other to 0)

If $TS \leq 0$:

TRCOMPBD = TS (Tag here "RC1")

TRCOMPAC = 0

If $TS > 0$:

TRCOMPAC = TS (Tag here "RC3")

TRCOMPBD = 0

If TRCOMPAC $<$ 0:

If TRCOMPBD $>$ 0: (Tag here "TCOMPAC-")

$TS = TRCOMPBD + TRCOMPAC$

If $TS \leq 0$:

TRCOMPAC = TS

TRCOMPBD = 0

If $TS > 0$:

TRCOMPBD = TS

TRCOMPAC = 0

$i = (\text{bit } 13 \text{ of DAPDATR3}) \quad (1 \text{ or } 0) \quad (\text{Tag here "RPREFCK"})$
 If $\text{TAU} \neq 0$: (roll desired)
 Perform "ROLLMATH"
 Perform "ROLLMATH" (2nd time)
 Perform "MERGMATH" $(\text{Tag here "COMBINE"})$
 Perform "MERGMATH" (2nd time)
 $\text{TS} = \text{TAU} + \text{TRCOMPBD}_{-i} \quad (\text{for } i = 1, \text{ gets TRCOMPAC})$
 Perform "CPLMATH"
 $\text{T5TEMP2} = \text{TS}$
 If $\text{TS} = 0$: (i.e. jets good)
 Proceed to "PASS2"
 $62\text{PT}_i = 0 \quad (\text{for } i = 1, \text{ loads } 62\text{YWT})$
 $\text{T5TEMP2} = \text{TAU}$
 If $\text{TRCOMPBD}_{-i} \neq 0$: $(\text{for } i = 1, \text{ checks TRCOMPAC})$
 Perform "ROLLALRM"
 Proceed to "PASS2"

PASS2

$i = |i - 1| \quad (\text{if } 0, \text{ set to } 1; \text{ if } 1, \text{ set to } 0)$
 $\text{T5TEMP2} = \text{T5TEMP2} + \text{TRCOMPBD}_{-i} \quad (\text{for } i = 1, \text{ gets TRCOMPAC})$
 $\text{TS} = \text{T5TEMP2}$
 Perform "CPLMATH"
 If $\text{TS} \neq 0$:
 $62\text{PT}_i = 0 \quad (\text{for } i = 1, \text{ loads } 62\text{YWT})$
 Perform "ROLLALRM"
 If any computed times $(5\text{PT}, 5\text{YWT}, 61\text{PT}, 61\text{YWT}, 62\text{PT}, 62\text{YWT}) \neq 0$:
 If that time is $< K_{14\text{ms}}$:
 Set that time to $K_{14\text{ms}}$
 Proceed to "BURBLE"

ROLLMATH

$i = |i - 1|$

If $TRCOMPAC_i = 0$: (for $i = 1$, check TRCOMPBD)

Return

If $TRCOMPAC_i > 0$:

If $TAU > 0$: (Tag here "PLUSCOMP")

Return

$TRCOMPAC_i = TRCOMPAC_i + TAU$

If $TRCOMPAC_i > 0$:

$TAU = 0$

Return

$TAU = TRCOMPAC_i$

$TRCOMPAC_i = 0$

Return

If $TAU \leq 0$: ($TRCOMPAC_i < 0$ if here) (Tag here "MNUSCOMP")

Return

$TRCOMPAC_i = TRCOMPAC_i + TAU$

If $TRCOMPAC_i \leq 0$:

$TAU = 0$

Return

$TAU = TRCOMPAC_i$

$TRCOMPAC_i = 0$

Return

MERGMATH

If $62PT_i = 0$: (checks 62YWT for $i = 1$)

$i = |i - 1|$

Return

$j = -i$ (in T5TEMP cell)
TS = TRCOMPBD_{-i} (TRCOMPAC for $i = 1$)

If TS = 0:

Proceed to "TESTTAU"

If TS > 0:

Proceed to "SIDE+"

Proceed to "SIDE-"

TESTTAU

TS = j

$j = -4$ (TRCOMPBD₋₄ is TAU)

If TS = j: (means have already been through once)

$i = |i - 1|$

Return

TS = TAU

If TS > 0:

Proceed to "SIDE+"

If TS < 0:

Proceed to "SIDE-"

$i = |i - 1|$

Return

SIDE+

T5TEMP2 = |TS| (TS = TRCOMPBD_{-i} or TAU, for $j = 0/-1$ or -4)

If TAUZ_i > 0: (for $i = 1$, TAU_Y)

Set bit (8 - i) of DELFIG = 1

$61PW_i = K_{prollbd_{2i}} \cap 62PW_i$ (combine roll with Y/Z translation)
(for $i = 1$, 61YWW loaded using 62YWW)

TS = 2 (62PT_i) - T5TEMP2 (for $i = 1$, 62YWT; factor of 2 for 2 jets)

If $TS \leq 0$:

$$TRCOMPBD_j = -TS \quad (\text{tag "TAGD" - 1}) \quad (\text{for } j = -4, \text{ TAU; for } j = -1, \text{ TRCOMPAC})$$

$$61PT_i = 2 (62PT_i) + 61PT_i \quad (\text{for } i = 1, \text{ 61YWT and 62YWT})$$

$$62PT_i = 0$$

$$i = |i - 1|$$

Return

$$62PT_i = \frac{1}{2} TS \quad (\text{Tag here "TAGC"})$$

$$TRCOMPBD_j = 0$$

$$61PT_i = 61PT_i + T5TEMP2$$

Proceed to "TESTTAU"

SIDE-

$$T5TEMP2 = |TS|$$

Set bit $(6 - i)$ of DELFLG = 1

If $TAUZ_i < 0$: (for $i = 1, \text{ TAUY}$)

Set bit $(8 - i)$ of DELFLG = 1

$$61PW_i = K_{mrollbd_{2i}} \wedge 62PW_i \quad (i = 1, \text{ 61YWW loaded, using 62YWW data})$$

$$TS = 2 (62PT_i) - T5TEMP2 \quad (\text{for } i = 1, \text{ 62YWT})$$

If $TS \leq 0$:

$$TRCOMPBD_j = TS \quad (\text{Tag here "TAGD"}) \quad (\text{for } j = -4, \text{ TAU; for } j = -1, \text{ TRCOMPAC})$$

$$61PT_i = 2 (62PT_i) + 61PT_i \quad (\text{for } i = 1, \text{ 61YWT and 62YWT})$$

$$62PT_i = 0$$

$$i = |i - 1|$$

Return

$$62PT_i = \frac{1}{2} TS \quad (\text{Tag here "TAGC"})$$

$$TRCOMPBD_j = 0$$

$$61PT_i = 61PT_i + T5TEMP2$$

Proceed to "TESTTAU"

CPLMATH

$$TS = \frac{1}{2} TS$$

If $TS = 0$:

Return

If $TS > 0$:

$$62PT_i = \lfloor TS \rfloor \quad (i = 1, \text{ load } 62YWT) \quad (\text{Tag here "CPL+"})$$

Set bit $(10 - i)$ of $DELFLG = 0$

$$j = 2 i$$

If $TS < 0$:

$$62PT_i = \lfloor TS \rfloor \quad (\text{Tag here "CPL-"})$$

Set bit $(10 - i)$ of $DELFLG = 1$

$$j = 2 i + 1$$

$DELFLG = DELFLG + \text{bit}(14 - i)$ (Tag here "TAGE")

$$62PW_i = K_{\text{prollbd}_j} \quad (i = 1, \text{ load } 62YWW)$$

$$TS = 62PW_i \cap \text{CH6FAIL} \quad (\text{non-zero for jet failures})$$

Return

BURBLE

$$DFT = 0 \quad (\text{DFT same as } DFT_0; DFT1_0 \text{ same as } DFT_1; DFT1_1 \text{ same as } DFT_2)$$

$TS = (\text{bits } 8-7 \text{ of } - \text{ channel } 31), \text{ shifted right } 6 \text{ places}$ ($\pm X$ trans. "true" values)

If $TS = 0$:

$$5AXW = 0$$

$$5BXW = 0$$

$$i = 1$$

Proceed to "GOCYCLE"

$$ATTKALMN = -5$$

If bit 10 of DAPDATR3 = 0:

$$5AXW = 0$$

If bit 10 of DAPDATR3 = 1:

$$5PT = 0$$

$$5AXW = K_{pxac_{TS-1}} \cap (-CH5FAIL)$$

If bit 7 of DAPDATR3 = 0: (Note that if bits 10 and 7 both
0, get no translation)

$$5BXW = 0$$

If bit 7 of DAPDATR3 = 1:

$$5YWT = 0$$

$$5BXW = K_{pxac_{TS+1}} \cap (-CH5FAIL)$$

i = 1

Proceed to "GOCYCLE"

GOCYCLE

$$TLEFT = 0$$

$$TT = K_{100mst6}$$

$$TS = TT - 61PT_i \quad (\text{for } i = 1, \text{ uses } 61YWT)$$

If TS > 0:

$$TLEFT = TS$$

$$TT = K_{100mst6} - TLEFT \quad (\text{i.e. } 61PT_i)$$

If TT = 0:

Proceed to "NO61PT"

$$T5TEMP = TT \quad (\text{Tag here "LONG61PT"})$$

$$TS = T5TEMP$$

If bit (8 - i) of DELFIG = 1:

$$TS = -TS$$

$$T5TEMP = T5TEMP + C_{alphap_i} TS$$

$$TS = T5TEMP$$

If bit (6 - i) of DELFLG = 1:

$$TS = - TS$$

$$DFT = DFT + TS$$

$$TS = TT$$

If bit (4 - i) of DELFLG = 1:

$$TS = - TS$$

$$DFTL_i = DFTL_i + TS$$

Proceed to "NO61PT"

NO61PT

$$TT = TLEFT$$

$$TS = TT - 62PT_i \quad (\text{for } i = 1, 62YWT)$$

If $TS > 0$:

$$TT = TLEFT - TS \quad (\text{i.e. } 62PT_i)$$

If $TT = 0$:

Proceed to "NO62PT"

$$T5TEMP = 2 TT \quad (\text{Tag here "LONG62PT"})$$

If bit (14 - i) of DELFLG = 0:

$$T5TEMP = C_{\text{alphap}_i} T5TEMP$$

$$TS = 2 TT$$

If bit (12 - i) of DELFLG = 1:

$$TS = - TS$$

$$DFTL_i = DFTL_i + TS$$

$$TS = T5TEMP \quad (\text{Tag here "ROLLCPL"})$$

If bit (10 - i) of DELFLG = 1:

$$TS = - TS$$

$$DFT = DFT + TS$$

Proceed to "NO62PT"

NO62PT

TT = K_{100mst6}
TS = TT - 5PT_i (i = 1, 5YWT)
If TS > 0:
 TT = K_{100mst6} - TS (i.e. 5PT_i)
 If TT = 0:
 Proceed to "NO5PT"
TS = | C_{ecp_i} TT | (Tag here "LONG5PT")
If bit (2 - i) of DELFLG = 0:
 TS = 2 TS
If bit (12 - i) of DELFLG = 1:
 TS = - TS
DFT1_i = DFT1_i + TS
Proceed to "NO5PT"

NO5PT

If i > 0:
 i = i - 1
 Proceed to "GOCYCLE"
If INHIBIT > 0:
 INHIBIT = INHIBIT - 1, limited > +0 (Tag here "JETSON")
 If bit 14 of 5WORD = 1: (set in "DKT6")
 Set bit 14 of 5WORD = 0
 Perform "SETUPT6"
 Proceed to "NOTIMING"
TS₁ = 61PT + 62PT (in L register; Tag here "INHIBCOMP")
TS₂ = 61YWT + 62YWT

$$TS_3 = TS_2 - TS_1$$

If $TS_3 \leq 0$:

$$TS = TS_1$$

If $TS_3 > 0$:

$$TS = TS_3 + TS_1 \quad (\text{i.e. } TS_2)$$

If $TS \neq 0$: (Tag here "ZONK")

$$TS = (K_{1d160} \text{ TS})_{\text{integral part}} + 1 \quad (\text{units of 0.1 second passes now})$$

INHIBIT = TS (zero value likely to be -0)

Perform "SETUPT6"

Perform "TIMING"

Proceed to "NOTIMING"

NOTIMING

T5PHASE = -0

T5LOC = "RCSATT"

Return to task status (via EDRUPT instruction at "MAKERUPT")

ARUPT_{dp} = DAPARUPT (Tag here "ENDJASK")

(BRUPT, QRUPT) = DAPBQRPT (QRUPT not actually loaded, although effect equivalent)

ZRUPT_{dp} = DAPZRUPT_{dp}

DAPZRUPT = -16383

Resume

TIMING

Inhibit interrupts

5WORD = 0

6WORD = 0

NEXT6INT = TIMEHOLD

i = 1

Proceed to "T6LOOP"

T6LOOP

```
If 5PTi = 0:                (for i = 1, checks 5YWT)
    5WORD = 5WORD + 5AXWi    (i = 1, 5BXW)    (Tag here "NO5TIME")
If 5PTi ≠ 0:
    If (NEXT6INT - 5PTi) > 0:
        NEXT6INT = 5PTi
    5WORD = 5WORD + 5PWi    (for i = 1, 5YWW)
If 61PTi = 0:                (for i = 1, checks 61YWT)    (Tag here "TEST61PT")
    If 62PTi ≠ 0:            (Tag here "NO61TIME")
        If (NEXT6INT - 62PTi) > 0:            (i = 1, 62YWT)
            NEXT6INT = 62PTi
        6WORD = 6WORD + 62PWi    (i = 1, 62YWW)
If 61PTi ≠ 0:
    If (NEXT6INT - 61PTi) > 0:
        NEXT6INT = 61PTi
    6WORD = 6WORD + 61PWi    (i = 1, 61YWW)
If i > 0:                    (Tag here "COUNTDWN")
    i = i - 1
    Proceed to "T6LOOP"
```

Release interrupts

Return

SETUPT6

Inhibit interrupts

TIME6 = K_{14ms}

Perform "C13STALL"

Set bit 15(TIME6 Count Enable) of channel 13 = 1

TIMEHOLD = K_{100mst6}

Release interrupts

Return

DKT6 (Entered from "T6START" if WHICHDAP = 1)

If DAPZRUPT > 0:

Set bit 14 of 5WORD = 1

Resume

Channel 5 = 5WORD

Channel 6 = 6WORD

TIME6 = NEXT6INT

Perform "C13STALL"

Set bit 15(TIME6 Count Enable) of channel 13 = 1

Perform "TIMEUPDT"

Resume

TIMEUPDT

i = 1

TS = NEXT6INT - 5PT_i (for i = 1, 5YWT)

If TS ≤ 0:

5PT_i = - TS

TS = NEXT6INT - 61PT_i (i = 1, 61YWT) (Tag here "61PTCHK")

If TS ≤ 0:

61PT_i = - TS

Proceed to "GOROUND"

TS = NEXT6INT - 62PT_i (i = 1, 62YWT)

If TS ≤ 0:

62PT_i = - TS

Proceed to "GOROUND"

GOROUND

If i > 0:

i = i - 1

Proceed to second line of "TIMEUPDT"

TIMEHOLD = TIMEHOLD - NEXT6INT

If TIMEHOLD \leq 0:

TIMEHOLD = K_{100ms}t6

Proceed to "TIMING"

ROLLALRM Entered from "SPNDXCK" and "PASS2"

If bit 11(501**FIG) of FLAGWRD3 = 0: (set 0 e.g. by "DONOUN89")

Perform "ALARM" (pattern 0501_g)

Set bit 11(501**FIG) of FLAGWRD3 = 1

Return

Quantities in Computations

See also list of major variables and list of routines

- 5AXW: Single precision X translation jet bits for channel 5 using the A/C system, loaded in "BURBLE" based on X-translation channel bits and assembled into 5WORD in "T6LOOP". 5AXW₁ is 5BXW, and information is in octal.
- 5BXW: Single precision X translation jet bits for channel 5 using the B/D system (see 5AXW).
- 5PT: Single precision time (remaining) for transmission of jet information in 5PW, scale factor B10, units centi-seconds. It is decremented in "TIMEUPDT". 5PT₁ is 5YWT.
- 5PW: Single precision octal pitch-command jets for channel 5, transmitted under control of the time information in 5PT. 5PW₁ is 5YWW.
- 5WORD: Single precision cell assembled in "T6LOOP" giving the jet pattern to be loaded into channel 5 when "DKT6" is entered. Bit 14 of the word is set 1 in "DKT6" if the interrupt generated while the jet-selection jask is going on, and is checked in "NO5PT".
- 5YWT: Single precision time (remaining) for transmission of jet information in 5YWW, scale factor B10, units centi-seconds (see 5PT).
- 5YWW: Single precision octal yaw-command jets for channel 5, transmitted under control of the time information in 5YWT.
- 61PT: Single precision time (remaining) for transmission of one-jet "pitch"/roll control information in 61PW, scale factor B10, units centi-seconds. It may reflect a combined rotation and translation, and hence if it is non-zero it takes priority over time (and jet selection) information controlled by 62PT (see "T6LOOP" logic). 61PT₁ is 61YWT.
- 61PW: Single precision octal one-jet "pitch" command for channel 6, transmitted under control of the time information in 61PT. 61PW₁ is 61YWW.
- 61YWT: Single precision time (remaining) for transmission of jet information in 61YWW, scale factor B10, units centi-seconds, used for "yaw"/roll control (see 61PT).
- 61YWW: Single precision octal one-jet "yaw" command for channel 6, transmitted under control of the time information in 61YWT.
- 62PT: Single precision time (remaining) for transmission of two-jet "pitch"/roll control information in 62PW, scale factor B10, units centi-seconds. 62PT₁ is 62YWT.
- 62PW: Single precision octal two-jet "pitch" command for channel 6, transmitted under control of the time information in 62PT. 62PW₁ is 62YWW.

62YWT: Single precision time (remaining) for transmission of two-jet "yaw"/roll control information in 62YWW, scale factor B10, units centi-seconds.

62YWW: Single precision octal two-jet "yaw" command for channel 6, transmitted under control of the time information in 62YWT.

6WORD: Single precision cell assembled in "T6LOOP" giving the jet pattern to be loaded into channel 6 when "DKT6" is entered.

ARUPT_{dp}: Cells used to save accumulator and L register when interrupt received. It is necessary to make special provisions to retain these cells elsewhere because of the "jask" implementation in the coding (allowing the computations to be interrupted by other tasks, but taking immediate priority over the job that may have previously been running).

ATTKALMN: See Digital Autopilot RCS Routines.

BRUPT: Cell used to save hardware B register when interrupt received (see ARUPT).

C_{alphap₀}: Single precision erasable memory constant, program notation "ALPHAP", scale factor B0, used to determine effect on TRCOMPBD due to channel 6 use for pitch control (see e.g. "GOODB").

C_{alphap₁}: Single precision erasable memory constant, program notation "ALPHAYW", scale factor B0, used to determine effect on TRCOMPAC due to channel 6 use for yaw control (see e.g. "GOODB").

C_{ecp₀}: Single precision erasable memory constant, program notation "ECP", scale factor B0, used to convert the jet on-time computed by the DAP phase plane logic for pitch rotation into the equivalent amount of time required using channel 6 "translation" jet.

C_{ecp₁}: Single precision erasable memory constant, program notation "ECYW", scale factor B0. See C_{ecp₀} (except used for yaw, of course).

C31FLWRD: See Digital Autopilot RCS Routines.

CH5FAIL: Single precision octal cell, loaded by R2 of N87, whose individual bits are set to 1 to indicate that the corresponding RCS jet driven by that bit of channel 5 is disabled. A V59E sets the cell to zero.

CH6FAIL: Single precision octal cell, loading by R3 of N87, serving for channel 6 an analogous function to CH5FAIL (V59E also sets zero).

DAPARUPT: Cells used to retain ARUPT_{dp} during the DAP jask (see ARUPT).

DAPBQRPT: Cells used to retain BRUPT and QRUPT during the DAP jask.

DAPDATR3: Single precision quantity whose individual octal digits are assigned control significance for the jet selection logic of the docked DAP. The cell may be loaded by R1 of N87. The individual bits of DAPDATR3 have the following meanings (digit #1 is also referred to as "A"; #2 as "B", etc.).

<u>Octal Digit</u>	<u>Word Bit</u>	<u>Significance</u>
1	15-14 13	Not used (no effect). If 1, A/C quad jets preferred for roll; if 0, B/D.
2	12-11 10	Not used (no effect). Allow use of quad A/C for X translation, if 1.
3	9-8 7	Not used (no effect). Allow use of quad B/D for X translation (if bits 10 and 7 both 0, <u>no</u> X translation takes place), if 1.
4	6-5 4	Not used (no effect). Pitch control by channel 6 translation jets (Z) is preferred if 1 (if 0, couple control preferred).
5	3-2 1	Not used (no effect). Yaw control by channel 6 translation jets (Y) is preferred if 1 (if 0, couple control preferred).

DAPZRUPPT: Cells used to retain ZRUPPT^{dp} during the DAP jask (see ARUPPT). Since ZRUPPT^{sp} is positive (bits^{dp} 15-13 = 0), DAPZRUPPT^{sp} is set negative to indicate that the DAP jask is not in progress (if found positive in "DKJSLECT", a software restart is caused).

DELFIG: Single precision octal flagword used to retain information on nature of the decisions made concerning jet selection (such as signs) in order to determine the proper "feedback" DFT material. The even-numbered bits are used for the "pitch" channel (B/D quad), and the next-lower odd-numbered bit for the "yaw" channel (A/C quad).

<u>Bit</u>	<u>Use</u>
15	Not assigned.
14/13	Set 1 to indicate that 62PT _i controlling roll (couple, hence no cross-coupling to pitch/yaw channel needed).
12/11	Set 1 to indicate negative 5PT _i /62PT _i effect on DFT _i .
10/9	Set 1 to indicate negative 62PT _i effect on DFT (roll).
8/7	Set 1 to indicate negative 61PT _i cross-coupling (via C _{alphap_i}) to roll.
6/5	Set 1 to indicate negative 61PT _i complete effect on roll.
4/3	Set 1 to indicate negative 61PT _i effect on DFT _i .
2/1	Set 1 to indicate 5PT _i time is for 1 jet.

DFT: See Digital Autopilot RCS Routines.

DFT1: Same as DFT₁ (and DFT1₁ is same as DFT₂).

INHIBIT: Single precision counter, scale factor B14, of the number of passes that should be bypassed in "NO5PT" (also initialized there): a pass occurs each 0.1 second. Also used for control in "SUPERJOB".

K_{14ms}: See Digital Autopilot RCS Routines.

K_{100mst6}: Single precision constant, program notation "100MST6", scale factor B10, units centi-seconds. Value is 160×2^{-14} , corresponding to 10×2^{-10} , where first term is value in centi-seconds and second is scale factor.

K_{200mst6}_i (i = -1 to 2): Set of single precision constants, program notation (i = 0) "200MST6", scale factor B10, units centi-seconds. Value for i = -1 and i = 2 is 0; for i = 0, value is 320×2^{-14} (i.e. 20×2^{-10} , or 20 cs); for i = 1, value is -320×2^{-14} .

K_{1dl160}: Single precision constant, program notation "1/160", scale factor B4, units of DAP passes/centi-second. Value is 0.00625, corresponding to 0.1×2^{-4} , where first term is number of passes (i.e. one pass every 0.1 second or 0.1 pass/centi-second) and second is scale factor.

K_{mrollbd0}: Single precision constant, octal value 00012₈, program notation "-ROLLBD", corresponding to channel 6 jets 10 and 12 (negative roll torque from B/D quad).

K_{mrollbd2}: Single precision constant, octal value 00240₈, program notation "-ROLLAC", corresponding to channel 6 jets 14 and 16 (negative roll torque from A/C quad).

K_{pck0}: Single precision constant, program notation "PCK", octal value 00010₈, corresponding to bit 4 (used to select bit of DAPDATR3).

K_{pck1}: Single precision constant, program notation "PCK +1", octal value 00001₈, corresponding to bit 1.

K_{ppitchtc}_i (i = 0 - 3): Set of single precision constants giving pitch and yaw jet patterns (0,1 and 2,3) for positive (0,2) and negative (1,3) polarities respectively. Values are:

<u>i</u>	<u>Octal</u>	<u>Jets</u>	<u>Result</u>
0	00005 ₈	3,1	+ Pitch
1	00012 ₈	2,4	- Pitch
2	00120 ₈	7,5	+ Yaw
3	00240 ₈	6,8	- Yaw

Jets are in channel 5, for i = 0 program notation "+PITCHTC".

K_{prollbd_i} ($i = 0 - 3$): Set of single precision constants giving roll jet patterns for quads B/D (0,1) and A/C (2,3) for negative (1,3) and positive (0,2) polarities. Values are:

<u>i</u>	<u>Octal</u>	<u>Jets</u>	<u>Result</u>
0	00005 ₈	11,9	+ Roll (B/D)
1	00012 ₈	10,12	- Roll (B/D)
2	00120 ₈	15,13	+ Roll (A/C)
3	00240 ₈	14,16	- Roll (A/C)

Jets are in channel 6, for $i = 0$ program notation "+ROLLBD" (by assembler techniques, equated to same cells as K_{ppitchtc_i}).

K_{pxac_i} ($i = 0 - 3$): Set of single precision constants giving X-translation jet patterns for quads A/C (0,1) and B/D (2,3) for negative (1,3) and positive (0,2) polarities. Values are:

<u>i</u>	<u>Octal</u>	<u>Jets</u>	<u>Result</u>
0	00011 ₈	2,1	+X (A/C)
1	00006 ₈	3,4	-X (A/C)
2	00220 ₈	6,5	+X (B/D)
3	00140 ₈	7,8	-X (B/D)

Jets are in channel 5, for $i = 0$ program notation "+XAC".

K_{pzforce_i} ($i = 0 - 3$): Set of single precision constants giving Y/Z translation jet patterns (Y for 2,3; Z for 0,1) for negative (1,3) and positive (0,2) polarities. Values are:

<u>i</u>	<u>Octal</u>	<u>Jets</u>	<u>Result</u>
0	00011 ₈	10,9	+ Z
1	00006 ₈	11,12	- Z
2	00220 ₈	14,13	+ Y
3	00140 ₈	15,16	- Y

Jets are in channel 6, for $i = 0$ program notation "+ZFORCE" (by assembler techniques, equated to same cells as K_{pxac_i}).

NEXT6INT: Single precision cell, scale factor B10, units centi-seconds, giving the value to be loaded into TIME6 when "DKT6" is entered. Its setting is determined in "T6LOOP" (set to minimum amount of time left among the various jet timing words).

QRUPT: Cell used to save hardware Q register when interrupt received (see ARUPT). Rather than restoring it, coding restores Q register and branches to a point in the "Resume" coding after QRUPT otherwise would have been loaded into Q, but for convenience this detail not shown.

T5LOC: See Digital Autopilot Interface Routines.

T5PHASE: See Digital Autopilot RCS Routines.

T5TEMP: Temporary storage cell, when used for time information has units of centi-seconds, scale factor B10.

T5TEMP2: Temporary storage cell, when used for time information has units of centi-seconds, scale factor B10.

TAU: See Digital Autopilot RCS Routines. TAU same as TAU₀.

TAU1: Same as TAU₁ (and TAU₁ same as TAU₂), see TAU.

TAUY: Single precision cell, scale factor B10, units centi-seconds, used for Y translation/yaw time information. Loaded in "SUPERJOB" if in Free mode with Y translation; loaded in "JTIME" with yaw timing information. TAU_Y is TAUZ₁.

TAUZ: Single precision cell, similar function to TAUY but for Z translation or pitch control. TAUZ₁ is TAUY.

TIME5, TIME6: See Digital Autopilot Interface Routines.

TIMEHOLD: Single precision cell, scale factor B10, units centi-seconds, used to initialize NEXT6INT in "TIMING" and decremented (with a zero/negative value causing setting to 100 ms) in "GOROUND".

TLEFT: Single precision cell, scale factor B10, units centi-seconds, used to maintain a record of how much time is left (in 0.1 second interval) for doing commands specified by 62PT₁ (after 61PT₁ time employed).

TRCOMPAC: Single precision cell, scale factor B10, units centi-seconds, used to contain equivalent single-jet roll "disturbance" time for the A/C quad area. TRCOMPAC₁ is TRCOMPBD.

TRCOMPBD: Single precision cell, scale factor B10, units centi-seconds, used to contain equivalent single-jet roll "disturbance" time for the B/D quad area. TRCOMPBD₋₁ is TRCOMPAC; TRCOMPBD₋₄ is TAU.

TT: Single precision cell, scale factor B10, units centi-seconds, used for temporary time storage purposes (hence the notation) in computations starting at "GOCYCLE".

ZRUPT: Cells used to contain program counter (Z register) and BBANK when program interrupt received. See ARUPT.

Digital Autopilot Entry Routines

CM/DAPON Entered from "P62"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 00₂

T5LOC = "T5IDLOC"

T6LOC = "T5IDLOC"

Channel 6 = 0

Channel 5 = 0

Set bits 15-13 of DAPDATR1 = 0

If bit 11(GAMDIFSW) of FLAGWRD6 = 0:

Delay 0.5 second (by putting job to sleep via "DELAYJOB")
and repeat check of bit 11

RCSFLAGS = 77776₈ (sets bit 3 to 1)

P63FLAG = -1

JETAG = 0

PAXERR1 = 0

Inhibit interrupts

ALFACOM = ALFAd180

BETACOM = BETAd180 ("least significant half" of ALFAi)

ROLLHOLD = ROLLd180

ROLLC_{sp} = ROLLd180 (rescaled to 80 revolutions)

Set bit 12(CMDAPARM) of FLAGWRD6 = 1

Set bit 1(NODOV37) of FLAGWRD2 = 1

Release interrupts

Proceed to "P62.1"

READGYMB Called initially by "P62"

CMdGYMDT = CMdGYMDT + 10

If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)

Set bit 1(GYMDIF) of FLAGWRD6 = 0

(If bit 6 of IMODES33 = 1):

Channel 6 = 0

Channel 5 = 0

JETAG = 0

OLDELi = 0 (i = P,Q,R)

GAMDOT = 0

Call "READGYMB" in 0.1 second

End of task

Set AOG = CDU_x and TS = AOG

mDELAOG = TS - AOG (ones complement difference formed)

Set AIG = CDU_y and TS = AIG

mDELAIG = TS - AIG (ones complement difference formed)

Set AMG = CDU_z and TS = AMG

mDELAMG = TS - AMG (ones complement difference formed)

If bit 2(CMDSTBY) of FLAGWRD6 = 0:

Channel 6 = 0

Channel 5 = 0

End of task

If bit 1(GYMDIF) of FLAGWRD6 = 0:

Set bit 1(GYMDIF) of FLAGWRD6 = 1

JETAG = 0

OLDELi = 0 (i = P,Q,R)

GAMDOT = 0

Call "READGYMB" in 0.1 second

End of task

Call "READGYMB" in 0.1 second

Proceed to "BODYRATE"

BODYRATE

$$TS = - mDELAMG \sin_{sp} AOG - mDELAIG \cos_{sp} AOG \cos_{sp} AMG$$

$$TS_1 = TS - OLDELQ$$

$$OLDELQ = TS$$

$$QREL = TS + \frac{1}{2} TS_1$$

$$TS = mDELAIG \sin_{sp} AOG \cos_{sp} AMG - mDELAMG \cos_{sp} AOG$$

$$TS_1 = TS - OLDELR$$

$$OLDELR = TS$$

$$RREL = TS + \frac{1}{2} TS_1$$

$$TS = - mDELAOG - mDELAIG \sin_{sp} AMG$$

$$TS_1 = TS - OLDELP$$

$$OLDELP = TS$$

$$PREL = TS + \frac{1}{2} TS_1$$

If GAMDOT \neq 0:

$$PREL = PREL + K_{sintr} GAMDOT \sin_{sp} (- ROLLd180)$$

$$QREL = QREL - GAMDOT \cos_{sp} ROLLd180$$

$$RREL = RREL - K_{costr} GAMDOT \sin_{sp} (- ROLLd180)$$

If bit 12(CMDAPARM) of FLAGWRD6 = 0, End of task

Set TIME5 to cause program interrupt #2 in 10 milliseconds

T5LOC = "ATTRATES"

End of task

ATTRATES

$$ALFAd180 = ALFAd180 + QREL, \text{ in range } \pm 180^\circ$$

$$CALFA = \cos_{sp} ALFAd180$$

$$SALFA = \sin_{sp} ALFAd180$$

$$PHIDOT = PREL CALFA + RREL SALFA$$

$$BETADOT = RREL CALFA - PREL SALFA$$

$$BETAd180 = BETAd180 + BETADOT$$

$$ALFAd180 = ALFAd180 + PHIDOT \sin_{sp} BETAd180, \text{ in range } \pm 180^\circ$$

$$AK_1 = ALFACOM - ALFAd180, \text{ in range } \pm 180^\circ$$

$$QAXERR = AK_1$$

$$ROLLTM = ROLLd180 + PHIDOT, \text{ in range } \pm 180^\circ$$

$$ROLLd180 = ROLLTM$$

$$RAXERR = BETACOM - BETAd180$$

If bit 3(05GSW) of FLAGWRD6 = 0, proceed to "EXDAP"

$$CMDAPMOD = -1$$

$$AK_1 = 77776_g \quad (\text{one bit negative})$$

$$AK_2 = 77776_g \quad (\text{one bit negative})$$

$$TS_1 = RREL - K_{sintr} PREL$$

$$TS = 1 \text{ sgn } TS_1$$

If $|TS_1| - K_{ydtim} < 0$:

$$TS = 0 \quad (\text{inside deadband})$$

$$TS_3 = K_{yjetcd} TS$$

$$TS = 1 \text{ sgn } QREL$$

If $|QREL| - K_{ydtim} < 0$:

$$TS = 0 \quad (\text{inside deadband for pitch rate damping})$$

Proceed to "EXDAPIN"

EXDAPIN

$$\text{Channel 5} = \text{TS}_3 + K_{\text{prjcd}} \text{TS}$$

If JETAG \leq 0:

If JETAG = 0:

$$\text{PAXERR1} = \text{PAXERR1} - \text{CALFA PHIDOT}$$

Proceed to "CM/FDAIR"

$$\text{JNDX} = - 1$$

$$\text{mVTd180} = - K_{2\text{tdtcd}} \text{PREL} \quad (\text{scaled B-2 revolutions/DAP cycle})$$

TS = - ROLLd180, rescaled to scale factor 80 revolutions

If bit 4(LATSW) of FLAGWRD6 = 1:

$$\text{TS}_1 = \frac{1}{2} + \frac{1}{2} + \text{ROLLC}_{\text{sp}} \text{ modulo } 1 \quad (\text{the } 1 \text{ is } 360^\circ)$$

$$\text{TS}_2 = \text{TS} - \frac{1}{2} - \frac{1}{2}$$

$$\text{LCXd360} = \text{TS}_1 + \text{TS}_2 \text{ modulo } 1$$

$$\text{TS} = K_{180\text{d8att}} \text{mVTd180} \left| \frac{\text{mVTd180}}{\text{CALFA}} + \text{LCXd360} + \frac{1}{2} \text{sgn LCXd360} \right.$$

(sgn sets term 0 if
LCXd360 = 0)

If $|\text{TS}| \geq 1$: (the 1 is 360°)

$$\text{LCXd360} = \text{LCXd360} - 1 \text{sgn TS}$$

If bit 4(LATSW) of FLAGWRD6 = 0:

Set bit 4(LATSW) of FLAGWRD6 = 1

$$\text{LCXd360} = \text{ROLLC}_{\text{sp}} + \text{TS} \text{ modulo } 1$$

$$\text{LCXd360} = \text{LCXd360} \text{ CALFA}$$

If CMDAPMOD = 1:

$$\text{JETAG} = 0$$

Proceed to "CM/FDAIR"

If CMDAPMOD = -0:

$$\text{LCXd360} = 0$$

$$\text{ERROR}_2 = \text{LCXd360} \quad (\text{for telemetry})$$

$$\text{PAXERR1} = \text{LCXd360}$$

Rescale mVTd180 to scale factor B-1 revolutions/DAP cycle.

$mVTd180E = mVTd180$

$VSQd4API = K_{180d8att} mVTd180^2$

$TS = LCXd360 + K_{xsd360} - (K_{xsd360} - VSQd4API) \text{sgn } mVTd180$

If $TS \leq 0$: (point in left half plane) (if $mVTd180 = 0$,
sgn term set 0)

$mVTd180 = -mVTd180$

$LCXd360 = -LCXd360$

$JNDX = -JNDX$

Proceed to "DZ1"

If $2 K_{xsd360} - TS \leq 0$, proceed to "DZ1" (point in right half
plane)

If $VSQd4API - K_{vsqmn} \leq 0$:

If bit 1 of CMDAPMOD = 1: (i.e. -0, since +1 already exit)

ROLLHOLD = ROLLTM

If bit 1 of CMDAPMOD = 0: (i.e. +0 or -1)

ROLLHOLD = $ROLLC_{sp}$, rescaled to B-1 revolutions in
range $\pm 180^\circ$

Channel 6 = 0

$VDTd180 = 0$ (point in velocity deadzone)

JETAG = 0

Proceed to "CM/FDAIR"

$JNDX1 = -JNDX$ (point in buffer zone)

$VDTd180 = -mVTd180$

$T_{on1} = 0$

Proceed to "GETON2"

DZ1

If $VSQd4API - K_{vsqmn} \leq 0$:

If $LCXd360 + mVTd180 - K_{xmind360} \leq 0$:

If bit 1 of CMDAPMOD = 1:

ROLLHOLD = ROLLTM

If bit 1 of CMDAPMOD = 0:

ROLLHOLD = $ROLLC_{sp}$, rescaled to B-1 revolutions
in range $\pm 180^\circ$

(If $LCXd360 + mVTd180 - K_{xmind360} \leq 0$):

Channel 6 = 0

$VDd180 = 0$

JETAG = 0

Proceed to "CM/FDAIR"

$JNDX1 = - JNDX$

$XDd360 = \frac{1}{2} VSQd4API + LCXd360 - K_{xsd360}$

$TS_2 = XDd360 + K_{mvmd360k}$

If $TS_2 < 0$:

$VDd180 = K_{ktrcs} XDd360$

QREG = ZREG (positive non-zero)

If $TS_2 \geq 0$:

QREG = +0

$VDd180 = - K_{mvmd360k}$ (notation also "-VMT/180")

$T_{on1} = K_{180d8att} (VDd180 + mVTd180)$

If $T_{on1} \leq 0$:

If QREG > 0:

$VDd180 = - mVTd180$

$T_{on1} = 0$

If QREG = 0:

$JNDX = JNDX1$

$T_{on1} = - T_{on1}$

Proceed to "GETON2"

GETON2

$T_{on2} = 2 K_{180d8att} VDd180$

If $T_{on2} < 0$:

$T_{on2} = - T_{on2}$

$JNDX1 = JNDX$

If $VDd180 = 0$:

$T_{off} = 2 K_{2jett}$

If $VDTd180 \neq 0$:

$$TS = LCXd360 + \frac{1}{2} T_{on1} (mVTd180 - VDTd180) - \frac{1}{2} T_{on2} VDTd180$$

If $|TS| - 2 |VDTd180| \geq 0$: (implemented by division of
TS shifted right 14 places by
VDTd180 and check of A: the
"2" comes about from TS scaling
of BO and VDTd180 of B-1)

$$T_{off} = 2 K_{2jett}$$

If $|TS| - 2 |VDTd180| < 0$:

$$T_{off} = K_{2jett} TS / VDTd180$$

$$T_{on1} = K_{4jett} T_{on1}$$

$$T_{on2} = K_{4jett} T_{on2}$$

JETAG = 0

SWdNDX = 1

TUSED = TUSED + TIME1 + 8192 + 8192, modulo 2^{14}

TUSED = TUSED + K_{mtm3}

If $T_{on1} - 2 \leq 0$: (The "2" in this and subsequent equations is
in units of centi-seconds, as are the times).

$$T_{on1} = -1$$

If $T_{on1} - 2 > 0$:

$$T_{1BITS} = K_{prjcd} JNDX$$

TUSED = TUSED + T_{on1}

If TUSED > 0 :

$$T_{on1} = 0$$

$$T_{off} = -1$$

$$T_{on2} = -1$$

Proceed to "JETCALL1"

If $T_{off} - 2 \leq 0$:

$$T_{off} = -1$$

If $T_{\text{off}} - 2 > 0$:
TUSED = TUSED + T_{off}
If TUSED > 0 :
 $T_{\text{off}} = 0$
 $T_{\text{on2}} = -1$
 Proceed to "JETCALL1"

If $T_{\text{on2}} - 2 \leq 0$:
 $T_{\text{on2}} = -1$
 Proceed to "JETCALL1"

T2BITS = $K_{\text{prjcd}} \text{JNDX1}$
TUSED = TUSED + T_{on2}
If TUSED > 0 :
 $T_{\text{on2}} = 0$
 Proceed to "JETCALL1"

JETCALL1

OUTTAG = 0
NUJET = 0
TBITS = 0
Set $TS_1 = T_{\text{on1}}$ and $T_{\text{on1}} = 0$
 $TS_2 = \text{TBITS}$ (TBITS written over with meaningless data)
If $TS_1 > 0$, proceed to "JETCALL2"
If $TS_1 = 0$, proceed to "JETCALL3"
If $TS_1 < 0$, proceed to second line of "JETCALL3"

JETCALL3

NUJET = TS_2
Set $TS_1 = T_{\text{off}}$ and $T_{\text{off}} = -1$
 $TS_2 = \text{TBITS}$ (TBITS written over with meaningless data)
If $TS_1 > 0$, proceed to "JETCALL2"

If $TS_1 = 0$:

$NUJET = TS_2$

Set $TS_1 = T_{on2}$ and $T_{on2} = -1$

$TS_2 = T2BITS$ (T2BITS written over with meaningless data)

If $TS_1 > 0$, proceed to "JETCALL2"

If $TS_1 = 0$:

$NUJET = TS_2$

 Proceed to third line of "JETCALL2"

JETCALL2

$NUJET = NUJET + TS_2$

Call "JETCALL" in TS_1 centi-seconds

Channel 6 = $NUJET$

If $OUTTAG > 0$, End of task

 Proceed to "CM/FDAIR"

JETCALL

$OUTTAG = 2$

If bit 2(CMDSTBY) of $FLAGWRD6 = 1$:

$TS_2 = 0$

 Proceed to "JETCALL3"

Channel 6 = 0

End of task

CM/FDAIR

$AK_0 = \frac{1}{4} PAXERR1$ (gives full scale of $67\frac{1}{2}^\circ$)

If $SWdNDX < 0$:

$SWdNDX = - SWdNDX$

 Perform "NEEDLER"

$T5LOC = "T5IDLOC"$

 Resume

$SWdNDX = 3 - SWdNDX$

If SWdNDX \geq 0:

CMTMTIME = TIME1

SWdNDX = - 13

TS = SWdNDX - 1

ENDBUF_{TS} = (PREL, QREL, RREL)

T5LOC = "T5IDLOC"

Resume

EXDAP

CMDAPMOD = +0

If $|CALFA| - K_{c45im} < 0$: (i.e. in range 45 to 135 degrees)

CMDAPMOD = +1

JETAG = -1

If P63FLAG = 1, set P63FLAG = 0

TS₁ = - RAXERR sgn SALFA (0 for SALFA = 0)

PAXERR1 = TS₁, rescaled to scale factor B0, units revolutions

TS₂ = - BETADOT sgn SALFA (0 for SALFA = 0)

Perform "BIASEDZ"

Channel 6 = $K_{prjcd_{TS}}$

RAXERR = (ROLLHOLD - ROLLd180) sgn SALFA (0 for SALFA = 0)

If $|CALFA| - K_{c45im} \geq 0$:

If CALFA $>$ 0:

If P63FLAG = 0:

P63FLAG = -1

Call "WAKEP62" in K_{nsec} centi-seconds

If JETAG \leq 0:

JETAG = 0

Channel 6 = 0

If CALFA $<$ 0:

CMDAPMOD = -0

RAXERR = RAXERR sgn CALFA

$AK_2 = RAXERR$

$TS_1 = RAXERR$

$TS_2 = RREL$

If $CMDAPMOD = +0$:

$TS_2 = BETADOT$

Perform "BIASEDZ"

$TS_3 = K_{yjetcd_{TS}}$

$TS_1 = QAXERR$

$TS_2 = QREL$

Perform "BIASEDZ"

Proceed to "EXDAPIN"

BIASEDZ Entered with attitude information in TS_1 (L-register) and rate information in TS_2 (A-register).

$TS_1 = TS_1 - K_{cmdbs} \operatorname{sgn} TS_2$ (gives biased error; sgn sets term zero if $TS_2 = 0$)

$TS = -1 \operatorname{sgn} TS_1$

If $|TS_1| - K_{yawim} < 0$:

$TS = 0$

If $|TS_2| - K_{4dsim} \geq 0$:

$TS = 0$

$TS_4 = 1 \operatorname{sgn} TS_2$

If $|TS_2| - K_{ydtim} < 0$:

$TS_4 = 0$

$TS = TS + TS_4$, with magnitude limited $\ll 1$ (done in calling routine by an "or" function)

Return

Quantities in Computations

See also list of major variables and list of routines

AIG: Value of CDU_y sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("inner gimbal angle").

AK_i (i = 0-2): See Digital Autopilot Interface Routines.

ALFACOM: Single precision value of commanded "pitch" angle, scale factor B-1, units revolutions. Set to ALFAd180 in "CM/DAPON", and to C_{alfapad} in "P62.1" (see Entry Preparation).

ALFAd180: See Entry Computations.

AMG: Value of CDU_z sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("middle gimbal angle").

AOG: Value of CDU_x sampled by "READGYMB", single precision twos complement, scale factor B-1, units revolutions ("outer gimbal angle").

BETACOM: Single precision value of commanded "yaw" angle, scale factor B-1, units revolutions. Set to BETAd180 in "CM/DAPON", and to zero in "P62.1".

BETAd180: See Entry Computations.

BETADOT: Single precision value of rate of change of "yaw" angle, scale factor B-1, units revolutions/decisecond.

CALFA: Single precision value of cos ALFAd180, scale factor B0, after ALFAd180 updated only by QREL (see "ATTRATES").

CMDAPMOD: Single precision quantity used to control the performance of the entry DAP. Individual values have the following meanings:

<u>Value</u>	<u>Significance</u>
-1	Bit 3(05GSW) of FLAGWRD6 = 1 (value set in "ATTRATES").
-0	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA is at least $K_{c4.5im}$ in magnitude with negative sign (causes damping only, with LCXd360 set to 0). Value set in "EXDAP".
+1	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA magnitude less than $K_{c4.5im}$. Value set in "EXDAP".
+0	Bit 3(05GSW) of FLAGWRD6 = 0 and CALFA at least $K_{c4.5im}$ in magnitude with positive sign. Value set in "EXDAP".

Scale factor is B14.

CMdGYMDT: Single precision call time, scale factor B_{14} , units centi-seconds, used to permit proper restart time to be set for performance of "READGYMB". Initialized to 5 in "READACCS" (when the base time used for the restart is also set), giving "READGYMB" performance 5 cs out of phase with "READACCS".

CMTMTIME: Value of TIME1 loaded for telemetry purposes in "CM/FDAIR", scale factor B_{14} , units centi-seconds. Cell is the same as UPBUFF+0 (see Uplink Processing).

DAPDATR1: See Digital Autopilot Interface Routines.

ENDBUF: Set of cells used for telemetry of (PREL, QREL, RREL), loaded in "CM/FDAIR" on alternate entries to entry DAP (every 0.2 seconds the cell set is loaded). Cell set loaded is indexed by a negative quantity, with ENDBUF-14, ENDBUF-11, etc. used for successive PREL values. ENDBUF-14 is the same cell as UPBUFF+2 (see Uplink Processing).

ERROR₂: See Digital Autopilot RCS Routines (loaded in "EXDAPIN" with the original value of LCXd360 (before sign reversal, if any), for telemetry).

GAMDOT: See Entry Computations.

IMODES33: See IMU Computations.

JETAG: Single precision quantity, scale factor B_{14} , used to control the performance of the entry DAP. The "normal" value is zero; a setting to +1 is made by "SETJTAG" (called every two seconds by "READACCS") to cause an update of roll output to be made; a setting to -1 is made in "EXDAP" if the magnitude of CALFA is less than K_{c45im} .

JNDX: Value of jet table index for time interval specified by T_{on1} , scale factor B_{14} . Initialized to -1 near start of "EXDAPIN", and complemented subsequently if necessary. A negative value causes positive jets to be selected by indexing K_{prjcd} . The quantity is single precision.

JNDX1: Single precision quantity, scale factor B_{14} , serving a similar function to JNDX, but for the jets controlled by the time interval specified by T_{on2} .

K_{2jett} : Single precision constant, program notation "2JETT", scale factor B_{13} , units centi-seconds/entry DAP cycle. Value is 400×2^{-14} , corresponding to 200×2^{-13} , i.e. a two-second entry DAP cycle (for performance of roll command updates).

K_{2tdtcd} : Single precision constant, program notation "2T/TCDU", scale factor B_{13} , units (entry DAP period)/(CDU sample period). Value is 40×2^{-14} , corresponding to 20×2^{-13} , i.e. a two-second entry DAP cycle (to perform roll command updates) and a 0.1 second CDU sampling period (period of performance of "READGYMB"). Converts angular rates from units of revolutions/decisecond to revolutions/DAP cycle.

- K_{4dsim} : Single precision constant, program notation "4D/SLIM", scale factor B-1, units revolutions/decisecond. Value is 16348×2^{-14} , but used in the program in such a way (a check for overflow, i.e. magnitude of 16384 or more, on an argument decremented by 1) that the effective value is $(16384 - 16348 + 1)$, or 37 least increments, corresponding to $37 \times 180^\circ \times 2^{-14} \times 10 = 4.065^\circ/\text{sec}$. The argument is decremented by 1 for convenience in forming the absolute value.
- K_{4jett} : Single precision constant, program notation "4JETT", scale factor B12, units centi-seconds/entry DAP cycle. Value is 800×2^{-14} , corresponding to 200×2^{-12} (cf. K_{2jett}).
- $K_{180d8att}$: Single precision constant, program notation "180/8ATT", scale factor B4, units of $(\text{revolutions/DAP cycle}^2)^{-1}$. Nominal value is 0.61813187, corresponding to the reciprocal of: $(9.1 \times (1/360) \times 2^2 \times 2^4)$, where first term is acceleration (degrees/second^2), second converts to units of revolutions, third converts to units of DAP cycle^2 (2 seconds each), and fourth is scale factor. Actual stored value is about 0.61810. Alternatively, $9.1 = 2 \times 4.55$.
- K_{c45im} : Single precision constant, program notation "C45LIM", scale factor B0. Value is $112778 = 4799$, but used in program in such a way (a check for overflow, i.e. a magnitude of 16384 or more, on an argument decremented by 1 for convenience in forming the absolute value) that effective value is $(16384 - 4799 + 1) = 11586 \times 2^{-14} = 0.707153$, or about $\cos 45^\circ$.
- K_{cmdbs} : Single precision constant, program notation "CM/BIAS", scale factor B-1, units revolutions. Value is 55×2^{-14} , corresponding to about 0.604° .
- K_{costr} : Single precision constant, program notation "COSTRIM", scale factor B0, value 0.93969, corresponding to $\cos(-20^\circ)$.
- K_{ktrcs} : Single precision constant, program notation "KTRCS", scale factor B0, value 0.5. Value corresponds to 0.25×2 , where first term is "K" ("slope of line determining rate at which error reduced") and second is DAP roll cycle rate of 2 seconds.
- K_{mtm3} : Single precision constant, program notation "-T-3", scale factor B14, units centi-seconds. Value is -203×2^{-14} , corresponding to -2.03 seconds. See TUSED.
- $K_{mvm360k}$: Single precision constant, program notation "-VM/360K", scale factor B0. Nominal value is -0.22222222, corresponding to $(-1) \times 20 \times (1/360) \times (1/0.25)$, where first term is an equation factor, second is "maximum roll rate limit" (degrees/second), third converts to revolutions, and fourth is reciprocal of "K" (cf. K_{ktrcs}). Program notation also "-VMT/180", for which would have scale factor B-1, units revolutions/DAP cycle, where the "(1/0.25)" factor is 2×2^{-1} (the first factor is two-second cycle, the 2nd is scale).
- K_{nsec} : Single precision constant, program notation "NSEC", scale factor B14, units centi-seconds. Value is 2100×2^{-14} , corresponding to 21 seconds. Program comments indicate it is an approximation to the time required to travel 65° at a rate of $3^\circ/\text{second}$.

K_{prjcd} : Set of three single precision cells, program notation "P/RJCODE", selected by means of an index having values -1, 0, +1. Values are:

<u>Index</u>	<u>Cell</u>	<u>Channel 6 Jets</u>	<u>Channel 5 Jets</u>
-1	00005 _g	#11, #9 (+ R)	#3, #1 (+ P)
0	00000 _g	none	none
+1	00012 _g	#10, #12(- R)	#2, #4 (- P)

K_{sintr} : Single precision constant, program notation "SINTRIM", scale factor B_0 , value -0.34202, corresponding to $\sin(-20^\circ)$. Program comments indicate that this is trim angle for $L/D = 0.3$.

K_{vsqmm} : Single precision constant, program notation "VSQMIN", scale factor B_1 , units revolutions. Nominal value is $0.61050061E-3$, corresponding to $(2/360)^2 \times (9.1/360)^{-1} \times 2^{-2} \times 2^2 \times 2^{-1}$, where first term is value of the square of "VMIN" (expressed in units of revolutions), second is acceleration (in revolutions/second²), third converts acceleration to DAP cycles (cf. $K_{180d8att}$), fourth converts VMIN to units of DAP cycles (from rev/sec), and fifth is scale factor. Actual value is about $0.61035E-3$, i.e. 00012_g.

$K_{xmind360}$: Single precision constant, program notation "XMIN/360", scale factor B_0 , units revolutions. Value is 182×2^{-14} , corresponding approximately to $4 \times (1/360)$, where first term is value of "XMIN" (degrees), and second converts to revolutions.

K_{xsd360} : Single precision constant, program notation "XS/360", scale factor B_0 , units revolutions. Value is 91×2^{-14} , corresponding approximately to $2 \times (1/360)$, where first term is value of "XS" (degrees), and second converts to revolutions. Tag also "BUFLIM".

K_{yawim} : Single precision constant, program notation "YAWLIM", scale factor $B-1$, units revolutions. Value is 16055×2^{-14} , but used in the program in such a way (a check for overflow, i.e. a magnitude of 16384 or more, on an argument decremented by 1 for convenience in forming the absolute value) that effective value is $(16384 - 16055 + 1) = 330$ least increments, corresponding to $330 \times (360) \times 2^{-15} = 3.625^\circ$.

K_{ydtim} : Single precision constant, program notation "YDOTLIM", scale factor $B-1$, units revolutions/decisecond. Value is 16366×2^{-14} , but used in program in such a way (cf. K_{4dsim}) that effective value is 19 least increments, corresponding to about $2.087^\circ/\text{second}$.

K_{yjetcd} : Set of three single precision cells, program notation "YJETCODE", selected by means of an index having values -1, 0, +1. Values are:

<u>Index</u>	<u>Cell</u>	<u>Channel 5 Jets</u>
-1	00120 _g	#7, #5 (+ Y)
0	00000 _g	none
+1	00240 _g	#6, #8 (- Y)

LCXd360: Single precision value of roll attitude error (shortest path if LATS_W = 1 when enter computations, otherwise an enforced roll over the top) to be corrected, scale factor B₀, units revolutions.

mDELAIG: Complement of change in AIG during previous 0.1 second, scale factor B-1, units revolutions/decisecond: is $AIG_{n-1} - AIG_n$.

mDELAMG: Complement of change in AMG during previous 0.1 second, scale factor B-1, units revolutions/decisecond. Notation "-DELAMG".

mDELAOG: Complement of change in AOG during previous 0.1 second, scale factor B-1, units revolutions/decisecond.

mVTd180: Complement of value of PREL, converted to units of revolutions/DAP entry cycle (of 2 seconds). Initially computed with scale factor B-2, and subsequently rescaled to scale factor B-1. Program notation is "-VT/180". Complemented if point in left half plane.

mVTd180E: Value of mVTd180 for telemetry, scale factor B-1, original sign.

NUJET: Single precision cell used to contain the jet pattern to be loaded into channel 6 in "JETCALL2".

OLDELi (i = P,Q,R): Previous values of "raw" PREL, QREL, and RREL, scale factor B-1, units revolutions/decisecond, used in "BODYRATE" to compensate for angular accelerations. The values used are the uncompensated PREL, QREL, and RREL (i.e. those derived based solely on CDU measurement data this cycle and the previous cycle).

OUTTAG: Single precision quantity, scale factor B14, set to 2 if "JETCALL" performed and to 0 if "JETCALL1" performed, indicating respectively that the exit from "JETCALL2" should be to End of task and to "CM/FDAIR".

P63FLAG: Single precision cell, scale factor B14, used for control of entry DAP sequencing. It is set to -1 in "CM/DAPON" and in "EXDAP" after "WAKEP62" calling computations initiated. It is set to 1 in "P62.1" after a response to the VO6N61 display, and if 1 is set 0 in "EXDAP" if |CALFA| is less than K_{c45im} (a 0 value permits "WAKEP62" calling computations). Intended to prevent P63 from starting until response to VO6N61 display, and to avoid more than one "WAKEP62" call. It is set to -1 in "P63".

PAXERR1: Single precision quantity, scale factor B₀, units revolutions, containing the information to be loaded into AK₀: during atmospheric entry, it is set to LCXd360 each two seconds, and updated by (CALFA) x (- PHIDOT) each 0.1 second between two-second roll updates.

PHIDOT: Single precision value of rate of change of "roll" angle, scale factor B-1, units revolutions/decisecond. Cos Beta \approx 1.

PREL: Single precision value of "roll" body angular velocity, scale factor B-1, units revolutions/decisecond. It includes corrections for acceleration and GAMDOT, and is along the "X_p body axis".

QAXERR: Single precision value of pitch attitude error (saved for telemetry purposes), scale factor B-1, units revolutions. It is also used in "EXDAP", and is computed in "ATTRATES".

QREG: Computer single precision Q register (address 0002_g), used to retain program control information in "DZ1" (zero or non-zero value). After a TC (transfer control) order, Q is loaded with the S-register contents corresponding to the address of the following step (see 3420.5-27).

QREL: Single precision value of "pitch" body angular velocity, scale factor B-1, units revolutions/decisecond (including corrections for acceleration and GAMDOT), measured along the "Y_b body axis".

RAXERR: Single precision value of yaw attitude error (saved for telemetry purposes), scale factor B-1, units revolutions. It is loaded in "EXDAP" and "ATTRATES".

RCSFLAGS: See Digital Autopilot Interface Routines.

ROLLC: See Entry Computations.

ROLLd180: See Entry Computations.

ROLLHOLD: Single precision value of ROLLd180 sampled when the entry DAP is started (in "CM/DAPON"), scale factor B-1, units revolutions, and used to provide error signal information for RAXERR in "EXDAP". It is set to 0 in "KEP2", and to ROLLC or ROLLTM in "EXDAPIN"/"DZ1".

ROLLTM: Value of ROLLd180 placed in a special cell for telemetry purposes, scale factor B-1, units revolutions, single precision.

RREL: Single precision value of "yaw" body angular velocity, scale factor B-1, units revolutions/decisecond (including corrections for acceleration and GAMDOT), measured along the "Z_b body axis".

SALFA: Single precision value of sin ALFAd180, scale factor B0, after ALFAd180 updated only by QREL (see "ATTRATES").

SWdNDX: Single precision cell, scale factor B14, used for indexing the proper ENDBUF cells to be loaded and to cause alternation in "CM/FDAIR" between update of attitude error needles and loading of ENDBUF. It is set to 1 in "P62" and also in "GETON2", causing ENDBUF to be loaded (after initialization to -13). Negative values cause "NEEDLER" to be performed (by "CM/FDAIR"), and the cell is complemented each entrance to "CM/FDAIR". The cell is the same as UPBUFF+1 (see Uplink Processing).

T_{off}: Single precision value of time, scale factor B14, units centiseconds, required for "drifting" (between firing times of jets specified by T_{on1} and T_{on2}). Corresponding "jet pattern" (zero) is in TBITS. Cell set 0 if no waitlist call is to be made (i.e. if T_{on1} is small but the T_{off} value is such that the time delay would expire after the next two-second computing interval); cell set -1 if the time interval specified should be ignored (i.e. is too small or has already been used in "JETCALL3" or T_{on1} large).

T_{on1} : Single precision value of time, computed initially with scale factor B2, units DAP roll cycles, and subsequently rescaled to scale factor B14, units centi-seconds (by using K_{4jett}), giving the required "on time" for the jets in T1BITS. Cell set 0 if no waitlist call to be made (T_{on1} of such a size that jets should be left on for the full two-second DAP roll computing interval); cell set to -1 if the time interval involved should be ignored (i.e. is too small).

T_{on2} : Single precision value of time, same scaling as T_{on1} , giving the required "on time" for the jets in T2BITS. Cell set 0 if no waitlist call is to be made (because delay would expire after the next two-second computing interval); set to -1 if the time involved should be ignored (i.e. is too small, if $T_{on1} + T_{off}$ is too big, or after being used in "JETCALL3").

T1BITS: Value of jet pattern to be sent for the time duration specified by T_{on1} (contents loaded into NUJET for transmission via channel 6). These jets are required to go from present phase-plane point to the desired (or maximum) rate, where a coast interval (T_{off}) takes place and then the T2BITS jets are fired for T_{on2} to drive to the origin.

T2BITS: Single precision cell used to contain the value of the jet pattern to be sent for the time duration specified by T_{on2} : see discussion of T1BITS.

T5LOC, T6LOC: See Digital Autopilot Interface Routines.

TBITS: Value of the jet "pattern" to be sent for the time interval specified by T_{off} : it is set zero in "JETCALL1" (and used for symmetry in the program logic).

TIME5: See Digital Autopilot Interface Routines.

TUSED: Single precision cell, scale factor B14, units centi-seconds, used to keep track of the time elapsed by T_{on1} , T_{off} , and T_{on2} , in order to ensure that waitlist calls will not be set to elapse after the start of the next computing cycle (2 seconds) of the roll entry DAP, and also to make sure that minimum-firing time constraints on the jets are not violated. Cell is set to - TIME1 in "SETJTAG" (about 5 cs, plus computing delays, before the two-second entry roll DAP cycle is performed), and then is set to TIME1 + TUSED (a nominal 5 cs) before the start of processing, with an additional 2(8192) added to compensate, if necessary, for TIME1 overflow. This 5 centi-seconds, plus the two-second cycle, is then subtracted (giving nominally - 200 cs), to which 2 cs is added to avoid problems with the quantization of one cs for the waitlist and the fact that TUSED must exceed 0 for branches in equation logic to be taken: the 5, 200, and 2 are combined into the one constant K_{mtm3} . Additions of T_{on1} , T_{off} , and T_{on2} to TUSED (which starts at a nominal -198 cs) are tested for a positive non-zero result indicating expiration of the two-second computing interval.

VDTd180: Single precision cell, scale factor B-1, units revolutions/DAP cycle, giving the velocity argument for use in computing jet on and off times. Can be loaded with either drift velocity, maximum allowable velocity, or present velocity. If in deadzone, is loaded with 0 (for telemetry purposes).

VSQd4API: Single precision cell, scale factor B1, units revolutions, proportional to square of present "roll" body angular velocity divided by nominal acceleration ($K_{180d8att} mVTd180^2$).

XDd360: Single precision value of predicted intercept of attitude error axis (rate = 0) by vehicle "trajectory", decremented by K_{xsd360} , scale factor B0, units revolutions.

ZREG: Hardware Z register (program counter), which can be used as a convenient source of a positive non-zero number (magnitude not significant). See QREG.

Digital Autopilot Interface Routines*

T6RUPT Entered based on program interrupt #1, controlled by TIME6

Proceed to address specified by T6LOC

T5RUPT Entered based on program interrupt #2, controlled by TIME5

If TIME5 \geq 50 cs, Resume (means TIME5 reset after interrupt was generated)

Proceed to address specified by T5LOC

HANDRUPT Entered based on program interrupt #10, controlled by manual controller inputs

Resume

T5IDLOC

Resume (if entered from TIME5 interrupt, TIME5 will cause an interrupt again in 2^{14} centi-seconds)

SETJTAG Called by "READACCS" for entry DAP

TUSED = - TIME1

JETAG = 1

End of task

STABLISH Entered for a V46E

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2 : (TVC DAP)

If bit 14 of DAPDATR1 = 1: (LM-on)

Perform "SWICHOVR"

Proceed to "PINBRNCH"

Proceed to "ALM/END"

Set bit 9(S4B Takeover Enable) of channel 12 = 0

TS = (bits 14-13 of DAPDATR1, shifted right 12 places) (to bits 2-1)

Inhibit interrupts

If TS = 00_2 : (no DAP)

Perform "ZEROJET"

HOLDFLAG = 1

*NOTE: "LM" references are from Apollo 15 program, and have been retained for mnemonic usefulness.

(If $TS = 00_2$):

T5LOC = "T5IDLOC"

Set bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 0

Proceed to "PINBRNCH"

If $TS = 11_2$: (Saturn DAP)

Proceed to "SATSTKON"

Set bit 6(45/46FLG) of FLAGWRD3 = 0

Perform "RCS DAPON" (TS = 01_2 or 10_2 , for RCS DAP)

Proceed to "PINBRNCH"

DAPDISP Entered for a V48E (this is Routine 03).

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2 : (TVC DAP)

Proceed to "ALM/END"

Perform "TESTXACT"

Establish "DONOUN46" (priority 10_8)

End of job

DONOUN46

$TS = 0446_{vn}$

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If bit 4 of DAPDATR1 = 1: (maximum deadband specified)

Set bit 12(MAXDBFLG) of FLAGWRD9 = 1

If bit 4 of DAPDATR1 = 0:

Set bit 12(MAXDBFLG) of FLAGWRD9 = 0 (bit used e.g.
in "INITSUBA")

If WHICHX2 = 0:

Perform "S41.2"

$TS = 0647_{vn}$

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If bits 14-13 of DAPDATR1 \neq 00₂:

If bits 14-13 of DAPDATR1 \neq 11₂:

Inhibit interrupts (DAPDATR1 first digit = 1,2,5,6)

Perform "MASSPROP"

Release interrupts

If WHICHX2 = 0:

Perform "S40.14"

TS = 0648_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

Proceed to "ENDEXT"

DKDAPON Entered for a V45E

If MODREG = 40:

Proceed to "ALM/END"

Set bit 9(S4B Takeover Enable) of channel 12 = 0

Inhibit interrupts

Set bit 6(45/46FLG) of FLAGWRD3 = 1

Perform "RCSDAPON"

Proceed to "PINBRNCH"

DKDISP Entered for a V44E (this is Routine 04).

Perform "TESTXACT"

Change priority of present job to 10₈

TS = 0587_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

Proceed to "DONOUN89"

DONOUN89

Set bits 12(500**FLG) and 11(501**FLG) of FLAGWRD3 = 0

TS = 0689_{vn}

Perform "GOMARKFR": if terminate, proceed to "ENDEXT"
if proceed, skip next 2 lines
otherwise, proceed to "DONOUN89" (NOTE that
bits reset again)

TS = 100₂ and perform "BLANKET" (R3BLNK)

End of job

Proceed to "ENDEXT"

SETMAXDB

If WHICHDAP = 1:

ADB = DKDB

Return

ADB = K_{maxdb}

Set bit 4 of DAPDATR1 = 1

Return

SETMINDB

If WHICHDAP = 1:

ADB = DKDB

Return

THETADX = CDU

ADB = K_{mindb}

Set bit 4 of DAPDATR1 = 0

Return

STICKCHK Entered with TS set to control pattern

PMANNDX = bits 2-1 of TS

YMANNDX = (bits 4-3 of TS), shifted right 2 places (to bits 2-1)

RMANNDX = (bits 6-5 of TS), shifted right 4 places (to bits 2-1)

Return (Error if bits 15-7 of TS non-zero)

SPSOFF Entered from "AUTO37", "ENGINOFF", "V97E", and "V97T"

Inhibit interrupts

$T_{\text{evt}} = T_{\text{now}}$

Set bit 7(ENGONFLG) of FLAGWRD5 = 0

Set bit 13(SPS Engine On) of channel 11 = 0

Set bit 14(S4B Cutoff) of channel 12 = 1

If TVCPHASE \neq -1:

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2 : (TVC DAP)

If |REPFRAC| \neq 0:

PACTOFF = DELPBAR_{sp}

YACTOFF = DELYBAR_{sp}

Return

TVCZAP

Set bits 11(Disengage Optics DAC), 8(TVC Enable), and 2(Enable Optics CDU Error Counters) of channel 12 = 0

OPTIND = -1 ("DAC" is digital-to-analog converter)

NVWORD1 = +0

Set bit 11(TIMRFLAG) of FLAGWRD7 = 0

Return

PRE40.6 Called by "V97E" or restart group 6.2

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 10_2 : (TVC DAP)

End of task

MRKRTMP = -1

CNTR = +1 (serves as flag that entered "S40.6" from here)

Proceed to "S40.6"

S40.6 Called by "P4OSXTY"

OPTIND = -0

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 0

Set bits 11 (Disengage Optics DAC) and 8(TVC Enable) of channel 12 = 1

Delay 0.06 seconds

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 1

Delay 0.02 seconds

If CNTR > 0: (i.e. from "PRE40.6")

TVCPITCH = PACTOFF - 0 (avoid loading with +0)

TVCYAW = YACTOFF - 0 (avoid loading with +0)

Set bits 12-11 (Gate output from TVCPITCH & TVCYAW) of
channel 14 = 1

End of task

If MRKRIMP > 0:

Perform the following for i = PITCH and then i = YAW, with
j = 11, 12 respectively:

$TVCi = K_{p2actdg}$

Set bit j (Gate output from TVCi) of channel 14 = 1

Delay 2 seconds

$TVCi = K_{m4actdg}$

Set bit j (Gate output from TVCi) of channel 14 = 1

Delay 2 seconds

$TVCi = K_{p2actdg}$

Set bit j (Gate output from TVCi) of channel 14 = 1

Delay 2 seconds

Delay 4 seconds

TVCPITCH = PACTOFF - 0

TVCYAW = YACTOFF - 0

Set bits 12-11 (Gate outputs from TVCPITCH & TVCYAW) of
channel 14 = 1

End of task

S40.14 Entered from "DONOUN46" and "REDAP"

JdM₀ = K_{cntone} IXX
JdM₁ = K_{cntone} IAVG
JdM₂ = JdM₁
KMJ₀ = (K_{cnttwo}) / IXX
KMJ₁ = (K_{cnttwo}) / IAVG
KMJ₂ = KMJ₁
Return

S40.15 Entered from "SWICHOVR", "TVCEXEC", and "TVCINIT1"

ldCONACC = K_{2pidm} IXX
VARK = KTLXdI IAVGdTLX
Return

S41.2 Entered from "DONOUN46" and "REDAP"

RATEINDX = 2 (bits 2-1 of DAPDATR1)
Inhibit interrupts
If bits 14-13 of DAPDATR1 \neq 10₂:
 Set bit 2(LMATCH) of FLAGWRD7 = 0
If bits 14-13 of DAPDATR1 = 10₂:
 Set bit 2(LMATCH) of FLAGWRD7 = 1
Release interrupts
TS = K_{dc46}
If bit 4 of DAPDATR1 = 1: (use wide deadband)
 TS = TS + K_{dc409}
ADB = TS

If bit 7 of DAPDATR1 = 1: (use B/D for X-translation)
XTRANS = 1

If bit 7 of DAPDATR1 = 0:
XTRANS = 0

If bit 10 of DAPDATR1 = 1: (use A/C for X-translation)
XTRANS = XTRANS - 1

If XTRANS = 0:
Set bit 15(2JETSFLG) of FLAGWRD1 = 0

If XTRANS \neq 0:
Set bit 15(2JETSFLG) of FLAGWRD1 = 1

If bit 13 of DAPDATR2 = 0: (use B/D for roll)
ACORBD = 1

If bit 13 of DAPDATR2 = 1: (use A/C for roll)
ACORBD = - 4096 (i.e. negative non-zero)

If bit 10 of DAPDATR2 = 1: (Quad A OK)
RACFAIL = 0

If bit 4 of DAPDATR2 = 0: (Quad C not OK)
RACFAIL = -1

If bit 10 of DAPDATR2 = 0: (Quad A not OK)
RACFAIL = 1

If bit 7 of DAPDATR2 = 1: (Quad B OK)
RBDFAIL = 0

If bit 1 of DAPDATR2 = 0: (Quad D not OK)
RBDFAIL = -1

If bit 7 of DAPDATR2 = 0: (Quad B not OK)
RBDFAIL = 1

Return

NEEDLER

If bit 4 (Coarse Align) of channel 12 = 1:

Set bit 3 of RCSFLAGS = 1

Return

If bit 3 of RCSFLAGS = 1:

Set bit 6 (Enable CDU IMU Error Counters) of channel 12 = 0

Proceed to "NEEDLE11"

If bit 2 of RCSFLAGS = 1, proceed to "NEEDLER2"

If bit 6 (Enable CDU IMU Error Counters) of channel 12 = 0:

Set bit 3 of RCSFLAGS = 1

Return

Proceed to "NEEDLES"

NEEDLE11

$\underline{AK} = -0$

$\underline{EDRIVE} = -0$

$\underline{CDUXCMD} = -0$

Set bits 3-2 of RCSFLAGS = 01_2

Return

NEEDLER2

Set bit 6 (Enable CDU IMU Error Counters) of channel 12 = 1

Set bits 3-2 of RCSFLAGS = 00_2

Return

NEEDLES

$\underline{TS} = -K_{dacsc} \underline{AK}$, with magnitude of each component limited $\leq K_{dacmit}$

$\underline{CDUXCMD} = \underline{CDUXCMD} + (\underline{TS} - \underline{EDRIVE})$

$\underline{EDRIVE} = \underline{TS}$

Set bits 15-13 (Gate outputs from $\underline{CDUXCMD}$) of channel 14 = 1

Return

MASSPROP Entered from "AUTO37", "DONOUN46", "ENGINOFF", and "TVCINIT1"

If bit 13 of DAPDATR1 = 1: (means LM-off)

$$\text{VARST}_i = K_{\text{nolemval}_i} \quad (i = 0 - 9)$$

If bit 13 of DAPDATR1 = 0:

$$\text{VARST}_i = K_{\text{intvalue}_i} + K_{\text{slopeval}_i} \text{LEMMASS} \quad (i = 0 - 9)$$

If bit 15 of DAPDATR1 = 1: (descent stage off)

$$\begin{aligned} \text{VARST1} &= \text{VARST1} + \text{VARST8} && (\text{VARST1 and VARST2 not} \\ &&& \text{incremented if DAPDATR1} \\ \text{VARST2} &= \text{VARST2} + \text{VARST9} && \text{is 77777g, which it should not}) \end{aligned}$$

$$\text{VARST7} = \text{VARST7} + K_{\text{dxitfix}} \quad (\text{VARST8 and VARST9 destroyed if used})$$

Proceed to "FIXCW"

FIXCW Entered from "TVCEXEC" and at end of "MASSPROP"

$$\text{DELCMWT} = \text{CSMMASS} + K_{\text{negbpw}}$$

If $\text{DELCMWT} \leq 0$:

$$\text{IavgdTLX} = \text{VARST7} \text{ DELCMWT} + \text{VARST2}$$

$$\text{Iavg} = \text{VARST6} \text{ DELCMWT} + \text{VARST1}$$

$$\text{Ixx} = \text{VARST5} \text{ DELCMWT} + \text{VARST0}$$

If $\text{DELCMWT} > 0$:

$$\text{IavgdTLX} = \text{VARST3} \text{ DELCMWT} + \text{VARST2}$$

$$\text{Iavg} = \text{VARST4} \text{ DELCMWT} + \text{VARST1}$$

$$\text{Ixx} = \text{VARST5} \text{ DELCMWT} + \text{VARST0}$$

$$\text{TS} = \text{CSMMASS}$$

If bit 14 of DAPDATR1 = 1: (means LM-on)

$$\text{TS} = \text{TS} + \text{LEMMASS}$$

$$\text{MASS} = \text{TS}$$

Return (to routine calling "MASSPROP" or "FIXCW")

NOTE: "RCS DAP" refers to undocked DAP only in this segment.

Quantities in Computations

See also list of major variables and list of routines

- ldCONACC: Single precision value of reciprocal of roll-axis acceleration used for TVC roll DAP, scale factor B9, units of (revolutions/second²)⁻¹. Computed in "S40.15", and is based on two-jet torque (see K_{2pidm}). Updated every 10 seconds during burn (by entrance to "S40.15").
- ACORBD: Single precision quantity, scale factor B14, specifying the quad pair to be used for roll attitude control in the RCS DAP. If positive (+1), quad B/D is to be used; if negative (-4096, for programming convenience), quad A/C is used. The TVC DAP alternates roll commands between quads. Cell is set in "S41.2" based upon bit 13 of DAPDATR2 (0 for positive, 1 for negative).
- ADB: Single precision value of the attitude error deadband used in RCS/docked DAP, scale factor B-1, units revolutions (one least increment is about 0.011°). For RCS DAP, set based on bit 4 of DAPDATR1, but can subsequently be changed by the program (with changes reflected in DAPDATR1 for transfers between "wide" and "narrow" deadbands). For docked DAP, set to DKDB. In P20, set based on R2 of N79 (with zero being 0.5°), each pass through "R61CSM" (R61CNTR = 0) or in "R67START". Deadband restored to R03 or DKDB value in "FIXDB" and "INITSUBA". When "DONOUN46" is performed, the R03 value is retained; a noun 01 procedure could also be used to load the cell, if desired. For RCS DAP, "wide" is 5° and "narrow" is 0.5°.
- AK (AK_0 , AK_1 , AK_2): Single precision values of attitude errors used as communication cells with "NEEDLER", scale factor B-1, units revolutions. Nominal scaling of display (see K_{dacsc}) is such that "full scale deflection" is 16.875° of attitude error, with a positive AK input to "NEEDLER" giving a negative loading of CDUXCMD. During P11, entered from "NOPOLYM" with AK_0 loaded so as to give full scale of 67.5° (other axes normal scaling), every half second (approximately). Can be used to steer Saturn if control of Saturn switch set. If Saturn DAP activated, entered from "SATSTICK" instead. During RCS/docked DAP operation, "NEEDLER" is entered every 0.2 seconds from "KMATRIX" (on alternate 0.2 seconds, AK is loaded in that routine with appropriate error information). During TVC DAP operation, "NEEDLER" is entered every 0.5 second from "TVCEXEC", with AK loaded by "TVCEXEC", "PCOPY", and "YCOPY" respectively. During entry DAP operation, "NEEDLER" is entered from "CM/FDAIR" every 0.2 second, with AK_0 loaded so as to give full scale of 67.5° and others having normal scaling. AK_0 is set 0 every 2 seconds if CMDAPMOD = -0 (see Digital Autopilot Entry Routines), and AK_1 and AK_2 effectively set 0 if bit 3 (O5GSW) of FLAGWRD6 = 1: they are otherwise computed in "ATTRATES" and "EXDAP" respectively. The general polarity of loading AK is "commanded angle minus present angle". A DSKY one-shot drive of error needles can be done by V43E means. NO4 can be used to display - AK.

CDUXCMD: See IMU Computations. If coarse align bit is not set, the data goes to FDAI attitude error display (x, y, z associated with roll, pitch, and yaw respectively, with 384 pulses giving full scale deflection). If bit 9 of channel 12 = 1, the information in the cells is used for Saturn control.

CNTR: Single precision cell used for switch-over control purposes in TVC DAP. Prior to entering TVC DAP, is used in "S40.6" as a flag to indicate whether that routine entered from "PRE40.6" (value 1, scale factor B14) or from "P4OSXTY" (value +0). If it was entered from "PRE40.6", then the four-second delay before driving the SPS gimbals to their trim position is bypassed.

CSMMASS: Value of CSM mass, scale factor B16, units kilograms, computed double precision in "S40.8" but used single precision elsewhere (including DSKY loading). Since the least significant half is not otherwise employed, the displayed value can differ from the onboard estimate by ± 4 kg (about 8.8 pounds). It can be loaded manually in R1 of N47, and is updated in "S40.8". The updating consists of subtracting two seconds worth of mass flow rate (determined by a pad-load constant in units of kilograms/centi-second) if the sensed accelerometer output in the previous two-second interval exceeded the thrust-fail criterion (another pad-load constant). The updating occurs whenever "S40.8" is entered (and the accelerometer output is sufficient), and uses MASSTMP (see "S40.8") for restart protection purposes. "S40.8" is entered only for P40, starting when Average-G is turned on (due to AVEGEXIT setting in "P4OSXTY"), and ending when AVEGEXIT is loaded with some other quantity (e.g. "CALCN85" in "P4ORCS" for an ENTR to F V99 or PRO to F V16 N40 after the burn). Because of this logic, no other special provisions need be made to compensate for thrust fail conditions: it should be noted, however, that the gain updates in "TVCEXEC" are based on the current value of CSMMASS. Although the mass updating is done as part of the Average-G computation loop, the computations are entered only in P40. As a result, if an SPS burn were to be done using the SCS and P47, for example, CSMMASS would have to be updated manually.

DAPDATR1: Single precision quantity whose individual octal digits are assigned control significance for the RCS and TVC DAPs, and which is displayed in R1 of N46. Digit #1 is used for the TVC DAP to determine vehicle configuration (LM-off, LM-on descent & ascent, or LM-on ascent only), while the remaining digits are used only by the RCS DAP (together with digit #1). Digits #2 - #5 are processed in "S41.2", which is entered when RO3 is performed (undocked) as well as when the RCS DAP is initialized (e.g. after a restart or when IMU data becomes usable after a period of not being usable). The individual bits of DAPDATR1 have the following meanings (digit #1 is also referred to as "A", #2 as "B", etc.):

<u>Octal Digit</u>	<u>Word Bit</u>	<u>Significance</u>
1	15-13	Specify vehicle configuration/desired DAP. Because of the use of various bits (e.g. 14 and 13) at different parts of the program to obtain vehicle information, caution should be used that the digits specified be those below, and other patterns should be avoided. <ul style="list-style-type: none"> 0 is set by "CM/DAPON" for entry DAP, and terminates other DAP activity if "STABLISH" is entered 1 indicates LM-off (causes RCS DAP to be started if "STABLISH" is entered) 2 indicates LM-on ascent and descent, and causes RCS DAP to be started if "STABLISH" is entered 3 indicates Saturn DAP, and causes Saturn DAP to be started if "STABLISH" is entered 6 indicates LM-on ascent only, and causes RCS DAP to be started if "STABLISH" is entered (bit 15 is used in "MASSPROP" to select proper coefficient values)
2	12-11 10	Not used (no effect) Allow use of quad A/C for X-translation if 1
3	9-8 7	Not used (no effect) Allow use of quad B/D for X-translation if 1 (if bits 10 and 7 both 0, program acts as if both bits were 1)
4	6-5 4	Not used (no effect) Select narrow deadband (0.5°) if 0 and wide deadband (5.0°) if 1: loading of ADB is done

<u>Octal Digit</u>	<u>Word Bit</u>	<u>Significance</u>
5	3 2-1	Not used (no effect) Specify maneuver rate for RHC or automatic maneuvers:
		00 ₂ for 0.05 ^o /sec
		01 ₂ for 0.20 ^o /sec
		10 ₂ for 0.50 ^o /sec
		11 ₂ for 2.00 ^o /sec

DAPDATR2: Single precision quantity whose individual octal digits are assigned control significance for the RCS DAP only, and which is displayed in R2 of N46. It is processed in "S41.2", with the individual bits having the meanings given below:

<u>Octal Digit</u>	<u>Word Bit</u>	<u>Significance</u>
1	15-14 13	Not used (no effect) Use quad A/C for roll if 1 (if 0, use B/D)
2	12-11 10	Not used (no effect) Quad A may be used if 1 (not to be used if 0)
3	9-8 7	Not used (no effect) Quad B may be used if 1 (not to be used if 0)
4	6-5 4	Not used (no effect) Quad C may be used if 1 (if bit 10 is 1, quad C not to be used if bit 4 is 0)
5	3-2 1	Not used (no effect) Quad D may be used if 1 (if bit 7 is 1, quad D not to be used if bit 1 is 0)

DELCMWT: Value of $(CSMMASS + K_{negbpw})$ computed in "FIXCW", scale factor B15, units kilograms, single precision (the difference between present CSM mass and the "breakpoint" value for use in computing mass-property information). Program notation is "TEMP333".

DELPBAR, DELYBAR: See Digital Autopilot TVC Routines.

DKDB: See Digital Autopilot RCS Routines.

EDRIVE: Single precision values of previous scaled AK output information in "NEEDLES", scale factor B14, units output drive pulses (there are 2¹³ pulses per revolution for standard scaling). EDRIVE is the internal computer image of the number contained in the appropriate error counter driven from CDUXCMD, and is required since interface is incremental pulses from a previous setting.

HOLDFLAG: Single precision quantity, scale factor B14, used to control the nature of the attitude hold performed by the RCS/docked DAP. A positive non-zero value means that at the next opportunity ("AHFNOROT" entered not in Free Mode and with rate filter initialized), the present CDU angles should be loaded into THETADX for use as the new attitude reference. A +0 value means that the DAP is in attitude hold about the angles in THETADX. A negative non-zero value means that automatic maneuvers are being performed (and that WBODY, CDUXD, BIAS, and DELCDU all have an effect, see Digital Autopilot RCS Routines). A positive non-zero value is loaded if the DAP's control of the vehicle is removed (e.g. bit 15 of channel 31 = 1); when the DAP is initialized; if the IMU data is not usable (i.e. not fine align); if the Free Mode is entered; if RHC commands are provided; or if no DAP is desired (by "STABLISH"). If HOLDFLAG is negative non-zero and a switch to Hold Mode is made, attitude hold is also performed (same effect as having HOLDFLAG positive non-zero, although the setting is not made). A positive zero setting is made in "AHFNOROT" after loading THETADX provided that bits 13-11 of RCSFLAGS = 0 (i.e. damping done); in "STOPRATE" (if previous value, negative, was -1, is set +0); and in "T5PHASE2" if presently negative and the magnitude of the middle gimbal angle exceeds 75°. A negative non-zero value is set in R61 and also when attitude maneuver computations (e.g. R60 or R67) have loaded the DAP interface cells. Programs checking HOLDFLAG can conclude, if it is not negative non-zero, that an interruption of the automatic maneuver has occurred.

IAVG: Single precision value of the "average" of moments of inertia about the y and z axes, scale factor B20, units kilogram-meters². It is computed as a function of CSMMASS (and LEMMASS if DAPDATRI so indicates) in "FIXCW" and is used in "S40.14" to compute quantities for the RCS DAP (the separate quantity IAVGdTLX is used for the TVC DAP). Storage address corresponds to "IXX" +1.

IAVGdTLX: Single precision value of the "average" of moments of inertia about the y and z axes divided by the "thrust moment" (product of engine thrust and the "moment arm from hinge-point to c.g."), scale factor B2, units seconds², used in "S40.15" to multiply KTLXdI to obtain VARK. Program notation is "IAVG/TLX", and storage address corresponds to "IXX" +2. Computed in "FIXCW".

IXX: Single precision value of moment of inertia about the x axis (roll), scale factor B20, units kilogram-meters², computed in "FIXCW". It is used in "S40.14" and "S40.15" to compute quantities for the RCS and TVC DAPs respectively.

JdM (JdM₀, JdM₁, JdM₂): Single precision value of 80% (RCS DAP) of the reciprocal of one-jet acceleration about roll, pitch, and yaw axes respectively, scale factor B23 (RCS DAP) or B27 (docked DAP), units of (revolutions/(deci-second centi-second))⁻¹, so that multiplication of a rate in revolutions/deci-second by JdM yields a time in centi-seconds. The 80% RCS DAP factor is in K_{cntone}, to account for "tolerances in the torque produced by the jets and uncertainties in the vehicle moment of inertia". RCS DAP quantities computed in "S40.14"; docked DAP ones are erasable memory constants loaded into JdM in "REDAP".

JETAG: See Digital Autopilot Entry Routines.

K_{2pidm} : Single precision constant, program notation "2PI/M", scale factor B-8. Nominal value is 0.00331017×2^8 , corresponding to the reciprocal of: $(700 \times 2 \times 1.355817948 \times (1/2\pi) \times 2^{-8})$, where first term is nominal torque due to a single jet (in foot-pounds), second accounts for two jets, third converts to units of newton-meters, fourth converts to units of revolutions, and fifth is scale factor.

K_{cntone} : Single precision constant, program notation "CONTONE", scale factor B3, value 0.662034. Value corresponds to the reciprocal of: $(700 \times 1.355817948 \times (1/2\pi) \times (1/0.8) \times 10^{-2} \times 10^{-1} \times 2^3)$, where first term is nominal torque due to single jet (in foot-pounds), second converts to units of newton-meters, third converts to units of revolutions, fourth is the 80% factor discussed for JdM , fifth converts to units of centi-seconds for result, sixth converts rotation rate to units of deci-seconds (see JdM for the time units), and seventh is the scale factor.

K_{cnttwo} : Constant, program notation "CONTTWO", scale factor B7, value 0.00118 (double precision). Value corresponds to $700 \times 1.355817948 \times (1/2\pi) \times 10^{-1} \times 10^{-2} \times 2^{-7}$, where first term is nominal torque due to single jet (in foot-pounds), second converts to units of newton-meters, third converts to units of revolutions, fourth converts to deci-seconds (for rotation rate), fifth converts to centi-seconds (see KMJ for time units), and sixth is scale factor.

K_{dacmit} : Single precision constant, program notation "DACLIMIT+1", scale factor B14, units of output drive pulses. Value is 384×2^{-14} , corresponding to 384 pulses (the saturated error counter). Actual program implementation involves 3 constants: -384, 16000, and +384, with the 16000 used as a magnitude check. If, when 16000 is added to magnitude of argument, the answer overflows (exceeds 16383), then limiting is required. The 384 corresponds to 600_g .

K_{dacsc} : Single precision constant, program notation "QUARTER", scale factor B15, value $\frac{1}{4}$, corresponding to a true value of 2^{13} . It converts AK from units of revolutions to units of output drive pulses, with 2^{13} pulses corresponding to the nominal scaling for a revolution. Hence 384 pulses, the error counter saturation level, correspond to $16 \frac{7}{8}$ degrees, as discussed for AK .

K_{dc46} : Single precision constant, program notation "DEC46", scale factor B-1, units revolutions. Value is 46×2^{-14} , the same as K_{mindb} .

K_{dc409} : Single precision constant, program notation "DEC409", scale factor B-1, units revolutions. Value is 409×2^{-14} , which when added to K_{dc46} gives 455×2^{-14} , the same as K_{maxdb} .

$K_{dxitfix}$: Single precision constant, program notation "DXITFIX", scale factor B-12, units seconds²/kilogram, used to increment VARST7 if LM-on APS-only configuration is specified. Nominal value is $-1.88275E-5 \times 2^{12}$, corresponding to $-0.854E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in sec²/pound, second converts to kilograms, and third is scale factor. Octal value is 75420_8 , corresponding to about $-0.85367E-5$ sec²/pound.

$K_{intvalue_0}$: Single precision constant, program notation "INTVALUE", scale factor B20, units kilogram-meters², used as the constant value to initialize VARST0 for LM-on. Nominal value is 26850×2^{-20} , corresponding to about $19803.5 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft², second converts to kilogram-meters², and third is scale factor. Octal value is 00644_8 , corresponding to 26880 kilogram-meters² or about 19826 slug-ft².

$K_{intvalue_1}$: Single precision constant, program notation "INTVALUE +1", scale factor B20, units kilogram-meters², used as the constant value to initialize VARST1 for LM-on (if descent stage off, subsequently incremented by VARST8). Nominal value is 127518×2^{-20} , corresponding to about $94052.5 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft². Octal value is 3710_8 , corresponding to 127488 kilogram-meters² or about 94038 slug-ft².

$K_{intvalue_2}$: Single precision constant, program notation "INTVALUE +2", scale factor B2, used to initialize VARST2 for LM-on (if descent stage off, subsequently incremented by VARST8) by providing the constant value. Nominal value is 0.54059×2^{-2} , corresponding to 0.54059 seconds² (since units of constant are seconds²). Octal value is 4246_8 , corresponding to about 0.540527 seconds².

$K_{intvalue_3}$: Single precision constant, program notation "INTVALUE +3", scale factor B-12, used to initialize VARST3 for LM-on, by providing the constant value. Nominal value is $0.153964E-4 \times 2^{12}$, with units seconds²/kilogram. Nominal value corresponds to about $0.698369E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds²/pound, second converts to kilograms, and third is scale factor. Octal value is 2011_8 , corresponding to about $0.698210E-5$ seconds²/pound.

$K_{intvalue_4}$: Single precision constant, program notation "INTVALUE +4",
scale factor B6, used to initialize VARST4 for LM-on, by
providing the constant value, units (kilogram-meters²)/kilogram.
Nominal value is -0.742923×2^{-6} , corresponding to about -0.2485468
 $\times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in
slug-ft²/pound, second converts to kilogram-meters², third converts
to kilograms, and fourth is scale factor. Octal value is 77501₈,
corresponding to about -0.24830 slug-ft²/pound.

$K_{intvalue_5}$: Single precision constant, program notation "INTVALUE +5",
scale factor B6, used to initialize VARST5 for LM-on, by
providing the constant value, units (kilogram-meters²)/kilogram.
Nominal value is 1.5398×2^{-6} , corresponding to about $0.51514 \times$
 $1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in
slug-ft²/pound. Octal value is 612₈, corresponding to about
 0.5149 slug-ft²/pound. Octal value same as $K_{nolemval_5}$.

$K_{intvalue_6}$: Single precision constant, program notation "INTVALUE +6",
scale factor B6, used to initialize VARST6 for LM-on, by
providing the constant value, units (kilogram-meters²)/kilogram.
Nominal value is 9.68×2^{-6} , corresponding to about $3.2385 \times$
 $1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in
slug-ft²/pound. Octal value is 4656₈, corresponding to about
 3.2384 slug-ft²/pound.

$K_{intvalue_7}$: Single precision constant, program notation "INTVALUE +7",
scale factor B-12, used to initialize VARST7 for LM-on,
by providing the constant value, units seconds²/kilogram. Nominal
value is $0.647625E-4 \times 2^{12}$, corresponding to about $0.293758E-4$
 $\times (1/0.45359237) \times 2^{12}$, where first term is value in sec²/pound,
second converts to kilograms, and third is scale factor. Octal
value is 10372₈, corresponding to about $0.293748E-4$ sec²/pound.

$K_{intvalue_8}$: Single precision constant, program notation "INTVALUE +8",
scale factor B20, used to initialize VARST8 for LM-on, by
providing the constant value, units kilogram-meters². Nominal
value is -27228×2^{-20} , corresponding to about -20082
 $\times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft²,
second converts to kilogram-meters², and third is scale factor.
Octal value is 77126₈, corresponding to -27200 kilogram-meters²
or about -20062 slug-ft².

$K_{intvalue_9}$: Single precision constant, program notation "INTVALUE +9",
scale factor B2, used to initialize VARST9 for LM-on, by
providing the constant value, units seconds². Nominal value is
 -0.206476×2^{-2} , corresponding to -0.206476 seconds². Octal value
of constant is 76261₈, corresponding to about -0.206543 seconds².

- $K_{m4actdg}$: Single precision constant, program notation "-4ACTDEG", scale factor B14, units CDU actuator output pulses (one pulse is 85.41 seconds; there are about 42.14963 pulses/degree). Value is -168×2^{-14} , or twice the value of $K_{p2actdg}$ (with opposite sign), corresponding to about -4 degrees.
- K_{maxdb} : Single precision constant, program notation "MAXDB", scale factor B-1, units revolutions. Value is 455×2^{-14} , corresponding to about 4.9988 degrees.
- K_{mindb} : Single precision constant, program notation "MINDB", scale factor B-1, units revolutions. Value is 46×2^{-14} , corresponding to about 0.5054 degrees.
- K_{negbpw} : Single precision constant, program notation "NEGBPW", scale factor B16, units kilograms, giving the complement of the CSM "breakpoint" weight for computation of mass properties. Nominal value is -15402.17×2^{-16} , corresponding to -15402.17 kilograms or about 33955.97 pounds (program comments indicate that "dry weight" is 23956 pounds, giving effect for constant of 10,000 pounds of propellant). Octal value is 70364₈, corresponding to -15404 kilograms (-33960.0 pounds).
- $K_{nolemval_0}$: Single precision constant, program notation "NOLEMVAL", scale factor B20, units kilogram-meters², giving setting for VARST0 for LM-off. Nominal value is 25445×2^{-20} , corresponding to about $18767 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft², second converts to kilogram-meters², and third is scale factor. Octal value is 616₈, corresponding to about 18787 slug-ft².
- $K_{nolemval_1}$: Single precision constant, program notation "NOLEMVAL +1", scale factor B20, units kilogram-meters², used to give setting for VARST1 for LM-off. Nominal value is 87450×2^{-20} , corresponding to about $64500 \times 1.355817948 \times 2^{-20}$, where first term is value in slug-ft². Octal value is 2526₈, corresponding to about 64481 slug-ft².
- $K_{nolemval_2}$: Single precision constant, program notation "NOLEMVAL +2", scale factor B2, used to give setting for VARST2 for LM-off, units seconds². Nominal value is 0.30715×2^{-2} , corresponding to 0.30715 seconds². Octal value is 2352₈, corresponding to about 0.30713 seconds².

- K_{nolemval}₃ : Single precision constant, program notation "NOLEMVAL +3", scale factor B-12, units seconds²/kilogram, used to initialize VARST3 for LM-off. Nominal value is $1.22877E-5 \times 2^{12}$, corresponding to $0.55736E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds²/pound, second converts to kilograms, and third is scale factor. Octal value is 1471₈, corresponding to about 0.55762E-5 seconds²/pound.
- K_{nolemval}₄ : Single precision constant, program notation "NOLEMVAL +4", scale factor B6, used to initialize VARST4 for LM-off, units kilogram-meters² per kilogram. Nominal value is 1.6096×2^{-6} , corresponding to about $0.5385 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 634₈, corresponding to about 0.53842 slug-ft²/pound.
- K_{nolemval}₅ : Single precision constant, program notation "NOLEMVAL +5", scale factor B6, used to initialize VARST5 for LM-off, units kilogram-meters² per kilogram. Nominal value is 1.54×2^{-6} , corresponding to about $0.5152 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 612₈, corresponding to about 0.5149 slug-ft²/pound. Octal value same as K_{intvalue}₅.
- K_{nolemval}₆ : Single precision constant, program notation "NOLEMVAL +6", scale factor B6, used to initialize VARST6 for LM-off, units kilogram-meters² per kilogram. Nominal value is 7.77177×2^{-6} , corresponding to about $2.600 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 3706₈, corresponding to about 2.6006 slug-ft²/pound.
- K_{nolemval}₇ : Single precision constant, program notation "NOLEMVAL +7", scale factor B-12, used to initialize VARST7 for LM-off, units seconds²/kilogram. Nominal value is $3.46458E-5 \times 2^{12}$, corresponding to about $1.5715E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in sec²/pound, second converts to kilograms, and third is scale factor. Octal value is 4425₈, corresponding to about 1.57148E-5 sec²/pound.
- K_{nolemval}₈ : Single precision constant loaded into VARST8 for LM-off, but should not be used. Cell corresponds to K_{intvalue}₀, with same scaling.
- K_{nolemval}₉ : Single precision constant loaded into VARST9 for LM-off, but should not be used. Cell corresponds to K_{intvalue}₁, but with octal number considered scaled B2 instead of B20. Value would be about 0.48633 seconds².

- $K_{p2actdg}$: Single precision constant, program notation "+2ACTDEG", scale factor B14, units CDU actuator pulses (see $K_{m4actdg}$). Value is 84×2^{-14} , corresponding to about +2 degrees.
- $K_{slopeval_0}$: Single precision constant, program notation "SLOPEVAL", scale factor B6, units kilogram-meters² per kilogram, used as the slope value to initialize VARST0 for LM-on. Nominal value is 1.96307×2^{-6} , corresponding to about $0.65675 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 767_g, corresponding to about 0.65734 slug-ft²/pound.
- $K_{slopeval_1}$: Single precision constant, program notation "SLOPEVAL +1", scale factor B6, units kilogram-meters²/kilogram, used as the slope value to initialize VARST1 for LM-on. Nominal value is 27.5774×2^{-6} , corresponding to about $9.2261 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound. Octal value is 15624_g, corresponding to about 9.2263 slug-ft²/pound.
- $K_{slopeval_2}$: Single precision constant, program notation "SLOPEVAL +2", scale factor B-12, units seconds²/kilogram, used as the slope value to initialize VARST2 for LM-on. Nominal value is $2.3548E-5 \times 2^{12}$, corresponding to about $1.06812E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds²/pound, second converts to kilograms, and third is scale factor. Octal value is 3054_g, corresponding to about 1.06793E-5 seconds²/pound.
- $K_{slopeval_3}$: Single precision constant, program notation "SLOPEVAL +3", scale factor B-26, units seconds²/kilogram per kilogram, used as the slope value to initialize VARST3 for LM-on. Nominal value is $2.1777E-9 \times 2^{26}$, corresponding to about $0.44805E-9 \times (1/0.45359237)^2 \times 2^{26}$, where first term is value in seconds²/pound per pound. Octal value is 4532_g, corresponding to about 0.44798E-9 seconds²/pound per pound.
- $K_{slopeval_4}$: Single precision constant, program notation "SLOPEVAL +4", scale factor B-8, units kilogram-meters² per kilogram per kilogram, used as the slope value to initialize VARST4 for LM-on. Nominal value is $1.044E-3 \times 2^8$, corresponding to about $0.1584E-3 \times 1.355817948 \times (1/0.45359237)^2 \times 2^8$, where first term is value in slug-ft²/pound per pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 10433_g, corresponding to about 0.158433E-3 slug-ft²/pound per pound.
- $K_{slopeval_5}$: Single precision constant, program notation "SLOPEVAL +5", scale factor B-8, units kilogram-meters² per kilogram per kilogram, used as the slope value to initialize VARST5 for LM-on. Value is zero, meaning that VARST5 is not affected by LEMMASS.

K_{slopeval}₆ : Single precision constant, program notation "SLOPEVAL +6", scale factor B-8, units kilogram-meters² per kilogram per kilogram, used as the slope value to initialize VARST6 for LM-on. Nominal value is $2.21068E-3 \times 2^8$, corresponding to about $0.335472E-3 \times 1.355817948 \times (1/0.45359237)^2 \times 2^8$, where first term is value in slug-ft²/pound per pound. Octal value is 22070₈, corresponding to about $0.33546E-3$ slug-ft²/pound per pound.

K_{slopeval}₇ : Single precision constant, program notation "SLOPEVAL +7", scale factor B-26, units seconds²/kilogram per kilogram, used as the slope value to initialize VARST7 for LM-on. Nominal value is $1.5166E-9 \times 2^{26}$, corresponding to about $0.31203E-9 \times (1/0.45359237)^2 \times 2^{26}$, where first term is value in seconds²/pound per pound, second converts to kilograms, and third is scale factor. Octal value is 3204₈, corresponding to about $0.31212E-9$ seconds²/pound per pound.

K_{slopeval}₈ : Single precision constant, program notation "SLOPEVAL +8", scale factor B6, units kilogram-meters² per kilogram, used as the slope value to initialize VARST8 for LM-on. Nominal value is -1.284×2^{-6} , corresponding to about $-0.4296 \times 1.355817948 \times (1/0.45359237) \times 2^{-6}$, where first term is value in slug-ft²/pound, second converts to kilogram-meters², third converts to kilograms, and fourth is scale factor. Octal value is 77266₈, corresponding to about -0.4300 slug-ft²/pound.

K_{slopeval}₉ : Single precision constant, program notation "SLOPEVAL +9", scale factor B-12, units seconds²/kilogram, used as the slope value to initialize VARST9 for LM-on. Nominal value is $2.00E-5 \times 2^{12}$, corresponding to about $0.9072E-5 \times (1/0.45359237) \times 2^{12}$, where first term is value in seconds²/pound, second converts to kilograms, and third is scale factor. Octal value is 2476₈, corresponding to about $0.90706E-5$ seconds²/pound.

KMJ (KMJ₀, KMJ₁, KMJ₂): Single precision value of the one-jet acceleration about roll, pitch, and yaw axes respectively, scale factor B-13, units of revolutions/(deci-second centi-second), so that multiplication of a time in centi-seconds by KMJ yields a rotation rate in revolutions/deci-second. RCS DAP quantities computed in "S40.14"; docked DAP ones are erasable memory constants loaded into KMJ in "REDAP".

KTLXdI: See Digital Autopilot TVC Routines.

LEMMASS: Single precision value of LM mass loaded by R2 of N47, and used in "MASSPROP" if bit 13 of DAPDATR1 = 0 to compute LM-attached mass properties constants. If bit 14 of DAPDATR1 = 1, is used in "FIXCW" to compute MASS (if DAPDATR1 set to indicate that LM not attached, cell contents ignored). Scale factor B16, units kilograms, but value should not exceed 2^{15} (or be equal to it) kilograms because of left shift in "MASSPROP": this is about 72000 pounds. For valid MASS computation, sum of CSMMASS and LEMMASS must be less than 2^{16} kilograms.

MASS: Single precision value of vehicle mass, program notation "WEIGHT/G", scale factor B16, units kilograms, loaded in "FIXCW" as the sum of CSMMASS and (if specified by DAPDATR1) LEMMASS. Note that the value in this cell is not updated by RO3 ("DONOUN46") unless digit #1 of DAPDATR1 is 1, 2, 5, or 6. The cell is used as if it were double precision in "S40.1", "S40.1B", and "S40.13": in these cases, the least significant half is AK₀.

MRKRTMP: See Burn Control.

NVWORD1: See Burn Control.

OPTIND: See Optics Computations.

FACTOFF: See Digital Autopilot TVC Routines.

PMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 2-1 of TS (set before entering routine). These bits correspond to the "pitch" input, with 00₂ meaning no maneuver, 01₂ meaning a positive maneuver, 10₂ meaning a negative maneuver, and 11₂ treated the same as 00₂ except for rate-command use of the RHC (see "T5PHASE2"). TS is set to the "true" values of the input information (which for the controllers have an inversion from the binary values, so that a binary 0 channel input represents a logical 1).

RACFAIL: Single precision cell, scale factor B14, set in "S41.2" based on Quad failure information in DAPDATR2. It is set to 0 if Quads A and C are both indicated as functioning; it is set to +1 if Quad A is indicated as having failed; and it is set to -1 if Quad A is indicated as functioning and Quad C as not functioning. See information below.

RATEINDX: Single precision cell, scale factor B14, set in "S41.2" to bits 2-1 of DAPDATR1 (shifted left one place for convenience in indexing double precision constants). Value is 0 for a specified maneuver rate of 0.05°/second; value is 2 for a specified maneuver rate of 0.2°/second; value is 4 for a specified maneuver rate of 0.5°/second; and value is 6 for a specified maneuver rate of 2.0°/second.

RBDFAIL: Single precision cell, scale factor B14, set in "S41.2" based on Quad failure information in DAPDATR2. It is set to 0 if Quads B and D are both indicated as functioning; it is set to +1 if Quad B is indicated as having failed; and it is set to -1 if Quad B is indicated as functioning and Quad D as not functioning. See information below.

RCSFLAGS: Single precision flag word used for control of RCS/docked DAP functions. The individual bits have the following meanings:

<u>Bit</u>	<u>Meaning</u>
15	Set in "NEWANGL" on initial pass (also in R67) if RATEINDX = 6 (meaning large-rate R60 maneuver), and used in "T5PHASE2" to control setting of ATTKALMN (to -1 if bit is 1, and otherwise to 0) correspondingly. It avoids having attitude maneuver routine change ATTKALMN if the DAP filter is being started up. Reset e.g. in "STOPRATE".
14	Bit set in the DAP if rate filter must be initialized before computations are performed. It is set if e.g. bit 15 of channel 31 is 1 (no G&N control), or if bit 6 of IMODES33 = 1, indicating IMU data not usable; it is reset e.g. in "REDAP".
13	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that roll damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no roll firing required by phase-plane.
12	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that pitch damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no pitch firing required by phase-plane.
11	Bit set 1 in "T5PHASE2" for a change in RHC inputs, to signify that yaw damping must be completed before attitude"hold"resumed, and reset zero in "J23" if no yaw firing required by phase-plane.
10-9	Bits set 0 in "T5PHASE2" and then set, for a "change" in RHC inputs, to the changed bits for roll (if any), to force a firing in "J23" (if not otherwise required by phase-plane) for that cycle, provided EDOT for that axis not too close to 0. Bit 1 if input 1→0 or 0→1.
8-7	Bits set 0 in "T5PHASE2" and then set, for a "change" in RHC inputs, to the changed bits for yaw (if any): see bits 10-9. "Change" for 10-5 could be docked DAP.
6-5	Bits set 0 in "T5PHASE2" and then set, for a "change" in RHC inputs, to the changed bits for pitch (if any): see bits 10-9. "Change" also forced firings (see 8-7).
4	Bit used in "KMATRIX" to cause "NEEDLER" to be performed on one pass (if presently 0) and a new AK value to be loaded on the next pass (a pass takes place once every 0.1 second).
3	Bit set to 1 to cause "NEEDLER" routine to perform the initialization functions for the FDAI display (reset error counter to 0 etc.). Reset by "NEEDLE11".
2	Bit set to 1 in "NEEDLE11" to signify that first pass of initialization function has been completed (causes, with bit 3 = 0, "NEEDLER2" to be entered at the next transfer to "NEEDLER"). Reset by "NEEDLER2".
1	Bit set 0 at end of "T6SETUP" and in "ZEROJET" to indicate to "T6START" that the next TIME6 interrupt that is valid should cause RWORD1, PWORD1, and YWORD1 to be loaded. The "T6START" routine then resets the bit to 1.

REPFRAC: See Digital Autopilot TVC Routines.

RMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 6-5 of TS (set before entering routine), shifted right 4 places. These bits correspond to the "roll" input (see PMANNDX).

T5LOC: Address controlling where computations begin when program interrupt #2 (controlled by TIME5) is received. It is set to "T5IDLOC" to cause these program interrupts to be ignored: this setting is done as part of a "fresh start" and also in "STABLISH" if bits 14-13 of DAPDTR1 are 00₂. If a restart is encountered, the loading of T5LOC is controlled by bits 15-14 of FLAGWRD6, and T5LOC is set to "T5IDLOC", "REDORCS", "REDOTVC", and "REDOSAT" for bits 15-14 = 00₂, 01₂, 10₂, and 11₂ respectively.

For the Saturn DAP, T5LOC = "SATSTICK" as set by "REDOSAT" and "SATSTICK". "REDOSAT" is entered due to T5LOC setting to that address for a restart or by "SATSTKON" (entered from "STABLISH"). TIME5 setting is such as to cause "SATSTICK" to be entered every 0.1 second.

For the RCS DAP, startup is accomplished (from "STABLISH" or from "ENGINOFF", "V97E", or "V97T") by entering "RCS DAPON", which sets TIME5 to cause transfer to "RCSATT" (T5LOC setting) in 0.6 seconds. The delay if from "AUTO37" is 3.1 seconds. Two interrupts 20 ms apart cause entrance to "RCSATT", with subsequent branching controlled by T5PHASE (see Digital Autopilot RCS Routines). After the second "RCSATT" entrance, if filter not being initialized T5LOC is set to "JETSLECT" with another 20 ms delay; that routine in turn resets T5LOC to "RCSATT" with a 60 ms delay (for the 100 ms RCS DAP cycle rate).

For the TVC DAP, T5LOC is set to "T5IDLOC" in "IGNITION" shortly after engine-on, to disable RCS DAP entrances. About 0.4 seconds later, T5LOC is set to "TVCDAPON" (with delay of about 10 ms). "TVCDAPON" sets T5LOC to "TVCINIT1" (delay about 10 ms) which in turn, after entering "TVCINIT4", sets T5LOC to "DAPINIT" with delay from T5TVCDT (20 ms for LM-off). "DAPINIT" in turn sets T5LOC to "PITCHDAP" with delay of one complete TVC DAP cycle (40 ms for LM-off), causing entrance to "PITCHDAP" for LM-off about 0.48 seconds after engine-on. "PITCHDAP" sets T5LOC to "YAWDAP", which resets it to "PITCHDAP", each with delay from T5TVCDT, i.e. loop cycle rate of 40 ms for LM-off.

The TVC DAP also uses waitlist task means to perform "TVCEXEC" every 0.5 second (starting .50 seconds after "TVCINIT4"), which in turn calls the roll DAP task, "ROLLDAP", 0.03 seconds after being entered. If a restart, "REDOTVC" sets T5LOC to "ENABL2", which in turn sets to "CMDSOUT", which in turn sets to either "TVCINIT1" or "DAPINIT" depending on timing of restart.

For the entry DAP, "CM/DAPON" sets T5LOC to "T5IDLOC", and the waitlist task "READGYMB", after entering "BODYRATE", sets it to "ATTRATES" (with 0.01 second delay). After executing the computations, T5LOC again set to "T5IDLOC" (at end of "CM/FDAIR"). The entry DAP cycle rate is defined by the 0.1-second waitlist calls of "READGYMB".

For the docked DAP, startup accomplished by entering "RCS DAPON", and is similar to RCS DAP. Instead of "JETSLECT", however, T5LOC is set to "DKJSLECT".

T6LOC: Address controlling where computations begin when program interrupt #1 (controlled by TIME6) is received. The interrupt is not generated unless TIME6 is counting down; contrary to TIME5, the countdown of TIME6 must be enabled by a channel bit (bit 15 of channel 13). It is set to "T5IDLOC" in "CM/DAPON" (since the Entry DAP does not make use of program interrupt #1).

In the RCS DAP, T6LOC is set to "T6START" in "ZEROJET" (which is also entered if no DAP or if Saturn DAP specified, to zero output channels). During normal RCS DAP operation, TIME6 is set to 14 ms in "JETSLECT", so that jets scheduled to be fired from about 86 ms to 100 ms will all be fired for the full 100 ms. Otherwise, TIME6 is set in "T6START" to the appropriate delay. Docked DAP similar (in "SETUPT6" and "DKT6").

In the TVC DAP, T6LOC is used for timing of the roll RCS jets, and is set to "NOROLL1" in "JETROLL".

THETADX: See Digital Autopilot RCS Routines.

TIME5: Computer hardware clock register (cell 0030₈), incremented by 1 each 10 ms under hardware control. Its overflow (reaching a value of $2^{14} = 16384$) causes program interrupt #2 ("T5RUPT"), which is coded to start performing the computations whose starting address is in T5LOC. Setting TIME5 "to cause program interrupt #2 in 20 ms", for example, is accomplished by setting $TIME5 = 37776_8$.

TIME6: Computer hardware clock register (cell 0031₈), decremented by 1 each 0.625 ms (i.e. at a 1600 pps rate) under hardware control if bit 15 of channel 13 = 1. Because of decrement rate, can be considered to contain time-interval information with scale factor B10 in units of centi-seconds. Its reduction to zero causes program interrupt #1, which is coded to start performing the computations whose starting address is in T6LOC (via "T6RUPT").

TUSED: See Digital Autopilot Entry Routines.

TVCPHASE: See Digital Autopilot TVC Routines.

TVCPITCH: Single precision value of computer special erasable memory counter cell 0054₈ when used to provide pitch steering commands for the SPS engine gimbal (if bit 8 of channel 12 is 1). Pulses are provided from the cell to change the value of the associated error counter at a 3200 pps rate, until the cell contents reduced to zero, provided bit 11 of channel 14 is set. The computer's image of what is in the error counter is in PCMD (see Digital Autopilot TVC Routines), i.e. the "integral" of TVCPITCH output since error counter reset, by zeroing of bit 2 of channel 12, e.g. in "S40.6". One pulse gives 85.41 arc seconds of output, thus leading to the expression of scaling information as "B14 in units of CDU actuator output pulses" or B0 in units of ASCREV (actuator revolutions, namely 2^{14} pulses or about 1.079751111 revolution). There are about 42.14963 pulses per degree. The same cell is used for CDUSCMD (see Optics Computations), and hence optics use must be disabled when the cell is used for TVC purposes.

TVCYAW: Single precision value of computer special erasable memory counter cell 0053₈ when used to provide yaw steering commands for the SPS engine gimbal (if bit 8 of channel 12 is 1). Pulses are provided from the cell to change the value of the associated error counter at a 3200 pps rate, until the cell contents reduced to zero, provided bit 12 of channel 14 is set. The computer's image of what is in the error counter is in YCMD (see Digital Autopilot TVC Routines). Scaling same as TVCPITCH, and the same cell is also used for CDUTCMD (see Optics Computations).

VARK: See Digital Autopilot TVC Routines.

VARST0: Single precision quantity computed in "MASSPROP" for use in "FIXCW" (cell not time-shared in TVC DAP), scale factor B20, units kilogram-meters², giving the value of IXX at "breakpoint" (when DELCMWT = 0).

VARST1: Single precision quantity computed in "MASSPROP", scale factor B20, used as the value of IAVG at "breakpoint", units kilogram-meters².

VARST2: Single precision quantity computed in "MASSPROP", scale factor B2, units seconds², used as the "breakpoint" value of IAVGdTLX.

VARST3: Single precision quantity computed in "MASSPROP", scale factor B-12, units seconds²/kilogram, giving the slope of IAVGdTLX with CSMMASS for values above "breakpoint" (DELCMWT greater than 0).

VARST4: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters² per kilogram, giving the slope of IAVG with CSMMASS for values above "breakpoint" (DELCMWT greater than 0).

VARST5: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters² per kilogram, giving the slope of IXX with CSMMASS (both above and below "breakpoint"). Octal value is same for LM-off and LM-on (regardless of LEMASS), including APS only.

VARST6: Single precision quantity computed in "MASSPROP", scale factor B6, units kilogram-meters² per kilogram, giving the slope of IAVG with CSMMASS for values not above "breakpoint" (DELCMWT not greater than 0).

VARST7: Single precision quantity computed in "MASSPROP", scale factor B-12, units seconds²/kilogram, giving the slope of IAVGdTLX with CSMMASS for values not above "breakpoint" (DELCMWT not greater than 0).

VARST8: Single precision quantity computed in "MASSPROP", scale factor B20, units kilogram-meters², meaningful only for LM-on, and used to increment VARST1 if descent stage off: it is added to VARST1, but is computed for LM-on from two constants both of which are negative, hence a reduction in VARST1 results from the addition.

VARST9: Single precision quantity computed in "MASSPROP", scale factor B2, units seconds², meaningful only for LM-on, and used to increment VARST2 if descent stage off.

WHICHDAP, WHICHX2: See Digital Autopilot RCS Routines.

XTRANS: Single precision cell, scale factor B14, set 0 if bits 10 and 7 of DAPDATR1 are the same value (allows four-jet X-translation); set to +1 if bit 7 is 1 (allows use only of quad B/D for X-translation); and set to -1 if bit 10 is 1 (allows use only of quad A/C for X-translation). Setting performed in "S41.2". See information below.

YACTOFF: See Digital Autopilot TVC Routines.

YMANNDX: Single precision cell, scale factor B14, loaded in "STICKCHK" with bits 4-3 of TS (set before entering routine), shifted right 2 places. These bits correspond to the "yaw" input (see PMANNDX).

Summary of RCS DAP Performance for Jet Failures

The only knowledge that the computer program has of RCS failures is by means of the information in DAPDATR2: there is no attempt made to deduce failures from observed spacecraft responses. The Entry DAP has no input provided to specify failed CM RCS jets, and the TVC roll DAP alternately uses (for rolls in a given direction) the A/C and B/D Quads, without regard to the RCS jet status information provided in DAPDATR2. Consequently, it is only the RCS DAP which makes use of jet status information furnished in DAPDATR2.

Routine "S4.1.2" serves as the interface between the information in DAPDATR2 and the RCS DAP jet selection logic (which is normally entered every 0.1 seconds while the RCS DAP is capable of providing output commands, and which starts at "JETSLECT"). Certain features of "S4.1.2" simplify the subsequent coding, but also have an influence upon the performance of the program that is not necessarily evidenced in the displays provided to the crew. These features include:

- a) If Quad A is reported failed, the program assumes (without checking and with no special indication to the crew) that Quad C is functional.
- b) If Quad B is reported failed, the program assumes (without checking and with no special indication to the crew) that Quad D is functional.
- c) If neither Quad A/C or Quad B/D is specified for X-translation, the program assumes that both are to be used.
- d) If only Quad A/C is specified for X-translation, no attempt is made to use Quad B/D for this purpose (even if Quad A or Quad C has a reported failure). Similarly for roll Quad selection.
- e) If only Quad B/D is specified for X-translation, no attempt is made to use Quad A/C for this purpose (even if Quad B or Quad D has a reported failure). Similarly for roll Quad selection.

In addition to the features of "S4.1.2" that affect program performance, the jet selection logic itself has certain characteristics which strongly influence program performance in the presence of reported RCS failures. These include the following:

- 1) If only Quad A/C is specified for X-translation, and Quad A or Quad C has a reported failure, no X-translation is performed except that which may incidentally arise from the single-jet firings to perform pitch control (or yaw control in the event that Quad B or Quad D also has a reported failure).
- 2) Similar to #1, except with Quad B/D instead of Quad A/C.
- 3) If both Quad A/C and Quad B/D are specified for X-translation, proper translation from the unfailed Quad (plus that which may incidentally arise from single-jet firings) will be performed. The routine used to estimate burn time ("S4.0.13"), however, will perform its computations based on four-jet ullage. If Quads A/C and B/D both have reported failures, the situation of #1 applies.

- 4) If Quad A/C is specified for roll, and Quad A or Quad C has a reported failure, then no Y-translation is performed except that which may incidentally arise from the single-jet firings to perform roll control, "in which case the astronaut should switch to B/D roll."
- 5) If Quad B/D is specified for roll, and Quad B or Quad D has a reported failure, then no Z-translation is performed except that which may incidentally arise from the single-jet firings to perform roll control, "in which case the astronaut should switch to A/C roll."
- 6) If Quad A/C is specified for roll, and Quad B or Quad D has a reported failure, then single-jet Z-translation will be performed for that portion of the 0.1-second DAP cycle not required for roll attitude control (even though a roll torque is produced).
- 7) If Quad B/D is specified for roll, and Quad A or Quad C has a reported failure, then single-jet Y-translation will be performed for that portion of the 0.1-second DAP cycle not required for roll attitude control (even though a roll torque is produced).
- 8) If Quad A/C is specified for roll, and Quad B or Quad D has a reported failure, then the jet used in #6 will also be fired during the portion of the 0.1-second DAP cycle that is used for roll attitude control if and only if the roll torque on the vehicle is not reduced to 0 during this interval. It would be fired if no Quad A/C failures were reported (since two-jet torque is used) and no Y-translation is simultaneously specified; other cases can best be analyzed by review of the jet table information itself (see Digital Autopilot RCS Routines).
- 9) If Quad B/D is specified for roll, and Quad A or Quad C has a reported failure, then the jet used in #7 will also be fired during the **portion** of the 0.1-second DAP cycle that is used for roll attitude control if and only if the roll torque on the vehicle is not reduced to 0 during this interval. It would be fired if no Quad B/D failures were reported and no Z-translation is simultaneously specified; see jet table information for other conditions.
- 10) As suggested by #1 - #3 above, the philosophy of #6 - #9 above for Y-translation and Z-translation is not employed for X-translation.

Digital Autopilot RCS Routines *

RCS DAPON

Set TIME5 to cause program interrupt #2 in 0.6 seconds

T5PHASE = 16324 (i.e. positive non-zero)

Set bit 3 of RCSFLAGS = 1 (cause FDAI error display initialization)

T5LOC = "RCSATT"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 01₂

Return

REDORCS

If T5PHASE \leq 0:

T5PHASE = -1

T5LOC = "RCSATT"

Proceed to "RCSATT"

RCSATT

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 15 (G&N Autopilot Control complement) of TS = 1:

Set bit 14 of RCSFLAGS = 1

HOLDFLAG = +MAX

ERROR = +0

If bit 14 (Free Mode complement) of TS = 1: (TS re-computed)

T5PHASE = 8192 (i.e. positive non-zero)

Set TIME5 to cause program interrupt #2 in 0.1 second

Perform "ZEROJET"

Proceed to "KMATRIX"

If T5PHASE $>$ 0: (Tag here "SETT5")

HOLDFLAG = 1

Proceed to "REDAP"

*NOTE: "LM" references are from Apollo 15 program, retained for mnemonic usefulness.

If T5PHASE = +0, proceed to "T5PHASE2"

If T5PHASE < 0, proceed to "REDAP"

T5PHASE = +0 (was -0, indicating phase 1)

T5TIME = TIME5

Set TIME5 to cause program interrupt #2 in 0.02 seconds

If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)

Set bit 14 of RCSFLAGS = 1

HOLDFLAG = 8192

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 14(Free Mode complement) of TS = 0, Resume

Set TIME5 to cause program interrupt #2 in 0.2 seconds

T5PHASE = 16364 (i.e. positive non-zero)

Resume

If bit 14 of RCSFLAGS = 1, proceed to "KMATRIX" (Tag here "RATEFILT")

$\underline{DRHO} = \underline{DRHO} - K_{gn1_ATTKALMN} \underline{DRHO}_{sp} - \underline{ADOT}$

Set $\underline{RHO}_0 = \underline{CDU}_x$ and $TS = \underline{RHO}_0$

$TS_1 = - (TS - \underline{RHO}_0)$ (ones complement difference formed)

Set $\underline{RHO}_1 = \underline{CDU}_y$ and $TS = \underline{RHO}_1$

$TS_2 = - (TS - \underline{RHO}_1)$ (ones complement difference formed)

$\underline{DELTEMP}_x = TS_1 + \underline{AMGB1} TS_2$

Set $\underline{RHO}_2 = \underline{CDU}_z$ and $TS = \underline{RHO}_2$

$TS_3 = - (TS - \underline{RHO}_2)$ (ones complement difference formed)

$\underline{DELTEMP}_y = \underline{AMGB4} TS_2 + \underline{AMGB5} TS_3$

$\underline{DELTEMP}_z = \underline{AMGB7} TS_2 + \underline{AMGB8} TS_3$

$\underline{DRHO} = \underline{DRHO} + \underline{DELTEMP}$

$\underline{MERROR} = \underline{MERROR} + \underline{DELTEMP}$

$$\underline{ADOT} = \underline{ADOT} + K_{gn2} \underline{ATTKALMN} \underline{DRHO}_{sp} + \underline{KMJ} \underline{DFT}$$

(last term involves e.g. $\underline{KMJ}_1 \underline{DFT}_1$
for \underline{ADOT}_1)

Proceed to "KMATRIX"

KMATRIX

TS = bits 4-1 of ATTSEC

If TS = 0:

Establish "AMBGUPDT" (priority 34_g)

ATTSEC = 9

If TS ≠ 0:

ATTSEC = TS - 1

If HOLDFLAG < 0:

$\underline{CDUXD} = \underline{CDUXD} + \underline{DELCDU}$, in range $\pm 180^\circ$

$\underline{THETADX} = \underline{CDUXD}_{sp}$

If bit 4 of RCSFLAGS = 0:

Set bit 4 of RCSFLAGS = 1

Perform "NEEDLER"

Resume

Set bit 4 of RCSFLAGS = 0

If bit 9(NEEDLFLG) of FLAGWRDO = 0: (means do not display total
attitude error; V62 and V63
set 1; V61 sets 0)

$\underline{AK} = - \underline{ERROR}$

Resume

If bit 6(N22ERNDS) of FLAGWRD9 = 1: (set 1 by V62, 0 by V63)

$\underline{ANGREF} = \underline{THETAD}$ (THETAD is N22)

If bit 6(N22ERNDS) of FLAGWRD9 = 0:

$\underline{ANGREF} = \underline{CPHIX}$ (CPHIX is N17)

$$TS = CDU$$

$$AK_0 = (ANGREF_x - TS_x) + AMGB1 (ANGREF_y - TS_y)$$

$$AK_1 = AMGB4 (ANGREF_y - TS_y) + AMGB5 (ANGREF_z - TS_z)$$

$$AK_2 = AMGB7 (ANGREF_y - TS_y) + AMGB8 (ANGREF_z - TS_z)$$

Resume

AMBGUPDT (Established once a second by "KMATRIX", and also by "REDAP").

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 \neq 01₂, End of job

$$AMGB1 = \sin_{sp} CDU_z$$

$$AMGB8 = \cos_{sp} (CDU_x - K_{quadan}) \quad (\text{ones complement difference formed when } K_{quadan} \text{ used})$$

$$AMGB4 = \cos_{sp} (CDU_x - K_{quadan}) \cos_{sp} CDU_z$$

$$AMGB5 = \sin_{sp} (CDU_x - K_{quadan})$$

$$AMGB7 = -\sin_{sp} (CDU_x - K_{quadan}) \cos_{sp} CDU_z$$

End of job

REDAP

If bit 6(45/46FLG) of FLAGWRD3 = 0: (Set 0 e.g. in "STABLISH")

$$WHICHDAP = 0$$

$$WHICHX2 = 0$$

Perform "S41.2"

Perform "S40.14"

If bit 6(45/46FLG) of FLAGWRD3 = 1: (Set 1 e.g. in "DKDAPON")

$$WHICHDAP = 1 \quad (\text{Tag here "DKDLOAD"})$$

$$WHICHX2 = 2$$

$$KMJ = (C_{kmjdckd}, C_{kmjldckd}, C_{kmj2dckd})$$

$$JdM = (C_{jdmdckd}, C_{jdmlckd}, C_{jdm2dckd})$$

$$ADB = DKDB$$

$$DAPZRUP = -16383$$

$$TIMEHOLD = K_{oct240}$$

ERROR = 0 (Tag here "ZEROT5")
 BIAS = 0
 TRCOMPBD = 0
 TRCOMPAC = 0
 TAU = 0
 ATTSEC = 0
 DRHO = 0
 DFT = 0
 MERROR = 0
 ADOT = 0
 WBODY = 0
 Perform "ZEROJET"
 CHANTEMP = 77777₈
 CH31TEMP = 77777₈
 SLOPE = K_{p24}WHICHDAP
 T5TIME = 4
 ATTKALMN = 11
 RHO = CDU
 T5PHASE = +0
 If bit 6(NOIMUDAP) of IMODES33 = 1: (IMU data not usable)
 ATTKALMN = 0
 RCSFLAGS = 2000₄₈ (bits 14 and 3 = 1)
 If bit 6(NOIMUDAP) of IMODES33 = 0:
 Establish "AMBGUPDT" (priority 3₄₈)
 RCSFLAGS = 0000₄₈ (bit 3 = 1)
 Set TIME5 to cause program interrupt #2 in 0.06 seconds
 Resume

ZEROJET

If WHICHDAP = 0: (dummy check for documentation convenience)

BLASTi+1 = 0 (i = 0, 1, 2)

BLASTi+0 = 0 (i = 0, 1, 2)

iWORD2 = 0 (i = R, P, Y)

iWORD1 = 0 (i = R, P, Y)

If WHICHDAP = 1: (dummy check for documentation convenience:
cells are same addresses as for undocked
DAP indicated above (BLASTi+1 thru iWORD1))

62YWW = 0

62PW = 0

61YWW = 0

61PW = 0

5YWW = 0

5PW = 0

62YWT = 0

62PT = 0

61YWT = 0

61PT = 0

5YWT = 0

5PT = 0

If WHICHDAP = 0:

BLAST1+1 = 4

BLAST2+1 = 11

If WHICHDAP = 1:

5WORD = 0 (Tag here "DKDZERO")

6WORD = 0

5AXW = 0

5BXW = 0

INHIBIT = 0

NEXT6INT = K_{bt9}

Set bit 1 of RCSFLAGS = 0 (Tag here "ZOT")
T6LOC = "T6START"
Set TIME6 to 23 (cause program interrupt #1 in about 14 ms)
Perform "C13STALL"
Set bit 15(TIME6 Count Enable) of channel 13 = 1
Return

T5PHASE2

If ATTKALMN > 0:
 ATTKALMN = ATTKALMN - 1 (Tag here "KALUPDT")
 TS = 8 - T5TIME
 If TS ≤ 0:
 TS = 2
 Set TIME5 to cause program interrupt #2 in TS centi-seconds
 T5PHASE = -0
 Resume
Set TS = TIME5 and TIME5 to cause program interrupt #2 in 0.02 seconds
T5TIME = T5TIME + TS
If $(|CDU_z| + K_{m75degs}) > 0$:
 Set bit 14(STIKFLAG) of FLAGWRD1 = 1
 If HOLDFLAG < 0:
 HOLDFLAG = +0
If WHICHDAP = 0: (Tag here "OKGIMB")
 If bit 15 of RCSFLAGS = 1:
 ATTKALMN = -1
 If bit 15 of RCSFLAGS = 0:
 ATTKALMN = 0
If WHICHDAP = 1:
 ATTKALMN = -4

Set bits 10-5 of RCSFLAGS = 0 (initialize forced firing bits)

TS = channel 31 (bits 6-1 are RHC command complements)

$TS_1 = \text{bits 6-1 of } (CH31TEMP \wedge (-TS) + (-CH31TEMP) \wedge TS)$
(gives ones in bits whose TS values \neq CH31TEMP)

If $TS_1 \neq 0$:

CH31TEMP = TS

$RCSFLAGS = RCSFLAGS + 2^4 TS_1$ (puts TS_1 into bits 10-5)

Set bits 13-11 of RCSFLAGS = 1 (rate damping)

If WHICHDAP = 1:

INHIBIT = -0

TS = bits 6-1 of (- CH31TEMP) (Tag here "NOCHANGE"; get binary 1 for logic 1, i.e. binary 0, in input)

If TS = 0:

Proceed to "AHFNOROT" (no RHC inputs)

HOLDFLAG = TS

Perform "STICKCHK"

Set bit 14(STIKFLAG) of FLAGWRD1 = 1

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 14(Free Mode complement) of TS = 0:

Proceed to "FREEFUNC"

If bit 14 of RCSFLAGS = 1:

Set TIME5 to cause program interrupt #2 in 0.2 second

T5PHASE = 16364 (i.e. positive non-zero)

Resume

If RATEINDX - 5 > 0: (note that RATEINDX not meaningful for docked DAP)

ATTKALMN = -1

Perform the following for $i = 2, 1, 0$ ($j = Y, P, R$ respectively):

If $j\text{MANNDX} = 0$:

$$\text{WBODY}_i = 0$$

If $j\text{MANNDX} \neq 0$:

If $\text{WHICHDAP} = 0$:

$$\text{TS} = j\text{MANNDX} + \text{RATEINDX} - 1$$

$$\text{WBODY}_i = K_{\text{mantab}_{\text{TS}}}$$

If $\text{WHICHDAP} = 1$:

If $j\text{MANNDX} = 01_2$: (plus command) (Tag here "GETDKRAT")

$$\text{WBODY}_i = \text{DKRATE}$$

If $j\text{MANNDX} = 10_2$: (minus command)

$$\text{WBODY}_i = -\text{DKRATE}$$

If bits 13-11 of $\text{RCSFLAGS} = 0$: (damping done)

If $\text{WHICHDAP} = 0$:

$$\text{MERROR}_i = \text{MERROR}_i - K_{\text{mantab}_{\text{TS}}}$$

If $\text{WHICHDAP} = 1$:

If $j\text{MANNDX} = 01_2$:

$$\text{MERROR}_i = \text{MERROR}_i - \text{DKRATE}$$

If $j\text{MANNDX} = 10_2$:

$$\text{MERROR}_i = \text{MERROR}_i + \text{DKRATE}$$

If bits 13-11 of $\text{RCSFLAGS} \neq 0$:

$$\text{MERROR}_i = 0$$

$$\text{ERROR}_i = \text{MERROR}_{i_{\text{sp}}}$$

Proceed to "JETS"

FREEFUNC

$$\text{TS}_2 = (\text{RCSFLAGS shifted right 4 places}) \quad (\text{changed bits back to 6-1})$$

TS = TS₂ \wedge (- CH3TEMP) (bits are 1 for 1 \rightarrow 0 transition)

If WHICHDAP = 0:

Proceed to "RHCMINP"

TS = (bits 6-1 of TS) (Tag here "DKRHCI MP")

Perform "STICKCHK"

TAU₀ = K_{200mst6}PMANNDX-1

TAU₁ = K_{200mst6}PMANNDX-1

TAUZ = TAU₁

TAU₂ = K_{200mst6}YMANNDX-1

TAUY = - TAU₂

ERROR = 0

If WHICHX2 = 0: (should not be)

T5LOC = "JETSLECT"

If WHICHX2 = 2:

T5LOC = "DKJSLECT"

Resume

AHFNOROT

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 14 (Free Mode complement) of TS = 0:

HOLDFLAG = 1 (Tag here "FREECONT")

TS₁ = channel 32 (bits 6-1 are minimum impulse command complements)

TS = CHANTEMP \wedge (- TS₁) (bits are 1 for 1 \rightarrow 0 transitions)

CHANTEMP = TS₁

Proceed to "RHCMINP"

If bit 14 of RCSFLAGS = 1:

Set TIME5 to cause program interrupt #2 in 0.2 seconds

T5PHASE = 16364 (i.e. positive non-zero)

Resume

TS = channel 31

If bits 14-13 of C31FLWRD \neq 00₂:

TS = C31FLWRD

If bit 13(Hold Mode complement) of TS = 0:

If HOLDFLAG \neq +0: (Tag here "HOLDFUNC")

WBODY_y = 0

BIAS_y = 0

If bits 13-11 of RCSFLAGS \neq 0:

ERROR_y = 0

Proceed to "JETS"

HOLDFLAG = +0

THETADX_y = CDU_y

If bit 13(Hold Mode complement) of TS = 1: (since bit 14 = 1,
this means Auto)

If HOLDFLAG \neq 0:

WBODY_y = 0

BIAS_y = 0

If bits 13-11 of RCSFLAGS \neq 0:

ERROR_y = 0

Proceed to "JETS"

HOLDFLAG = +0

THETADX_y = CDU_y

TS₁ = CDU_y - THETADY (ones comp. difference formed) (Tag here "ATTHOLD")

TS₂ = CDU_z - THETADZ (ones comp. difference formed)

ERROR₀ = (CDU_x - THETADX) + AMGB1 TS₁ (ones comp. difference)

$$\text{ERROR}_1 = \text{AMGB4} \text{ TS}_1 + \text{AMGB5} \text{ TS}_2$$

$$\text{ERROR}_2 = \text{AMGB7} \text{ TS}_1 + \text{AMGB8} \text{ TS}_2$$

If $\text{HOLDFLAG} < 0$:

$$\text{ERROR} = \text{ERROR} + \text{BIAS}$$

Proceed to "JETS"

RHCMINP

$\text{TS} = (\text{bits } 6-1 \text{ of } \text{TS})$

If $\text{WHICHDAP} = 0$:

Perform "STICKCHK"

$$\text{TAU}_0 = K_{\text{mintau}_{\text{RMANNDX}}}$$

$$\text{TAU}_1 = K_{\text{mintau}_{\text{PMANNDX}}}$$

$$\text{TAU}_2 = K_{\text{mintau}_{\text{YMANNDX}}}$$

If $\text{WHICHDAP} = 1$:

If $\text{TS} \neq 0$: (Tag here "DKMICIMP")

$$\text{INHIBIT} = -0$$

Perform "STICKCHK"

$$\text{TAU}_0 = K_{50\text{mst6}_{\text{RMANNDX}-1}}$$

$$\text{TAU}_1 = K_{50\text{mst6}_{\text{PMANNDX}-1}}$$

$$\text{TAUZ} = \text{TAU}_1$$

$$\text{TAU}_2 = K_{50\text{mst6}_{\text{YMANNDX}-1}}$$

$$\text{TAUY} = - \text{TAU}_2$$

$$\text{ERROR} = 0$$

If $\text{WHICHX2} = 0$:

$\text{T5LOC} = \text{"JETSLECT"}$

If $\text{WHICHX2} = 2$:

$\text{T5LOC} = \text{"DKJSLECT"}$

Resume

JETS

$$TS_1 = ADB + K_{flat_WHICHDAP}$$

i = 2

Proceed to "JLOOP"

JLOOP

$$EDOT = ADOT_i$$

If HOLDFLAG \neq 0:

$$EDOT = EDOT - WBODY_i$$

$$AERR = ERROR_i$$

$$ADBVEL = TS_1 \operatorname{sgn} EDOT \quad (-0 \text{ is negative})$$

$$AERRVEL = AERR \operatorname{sgn} EDOT \quad (-0 \text{ is negative})$$

If $(AERRVEL - ADB + C_{wlhdslop_WHICHDAP}) \leq 0$: (Tag here "J6.")

If $(|EDOT| - K_{wlh_WHICHX2}) > 0$: (Tag here "J8")

$$TS = K_{wl_WHICHX2} \operatorname{sgn} EDOT - EDOT \quad (\text{Tag here "J22"})$$

Proceed to "JTIME"

If $(|EDOT| / \text{SLOPE} + TS_1 + AERRVEL) \gg 0$: (Tag here "NJ22")

Proceed to "J23"

If $(|EDOT| - K_{wlmh_WHICHX2}) > 0$:

Proceed to "J23"

If $(AERRVEL + TS_1 + C_{wlmhdslop_WHICHDAP}) \leq 0$: (Tag here "NJ23")

$$TS = K_{wl_WHICHX2} \operatorname{sgn} EDOT - EDOT \quad (\text{Tag here "J22"})$$

Proceed to "JTIME"

$$TS = -EDOT - C_{slope2_WHICHDAP} (AERR + ADBVEL) \quad (\text{Tag here "J24"})$$

Proceed to "JTIME"

If (AERRVEL - TS₁) ≤ 0:

If (ADB - AERRVEL - |EDOT| / SLOPE) ≤ 0:

TS = - EDOT (Tag here "J18")

Proceed to "JTIME"

If WHICHDAP = 0:

Proceed to "J23"

If (|EDOT| / SLOPE + AERRVEL + TS₁) < 0: (Tag here "NEWCHECK")

TS = - EDOT - C_{slope2}_{WHICHDAP} (AERR + ADBVEL) (Tag here "J24")

Proceed to "JTIME"

Proceed to "J23"

If (AERRVEL - TS₁ - C_{wlmhdslp}_{WHICHDAP}) ≤ 0: (Tag here "J7")

TS = C_{slope2}_{WHICHDAP} (ADBVEL - AERR) - EDOT (Tag here "J20")

Proceed to "JTIME"

TS = - K_{wl}_{WHICHX2} sgn EDOT - EDOT (Tag here "J21")

Proceed to "JTIME"

J23 (Entered if conclude in deadzone)

j = 13 - i

Set bit j of RCSFLAGS = 0 (reset rate damping bit)

If i = 2:

TS = 00300₈ (bits 8-7, yaw command)

If i = 1:

TS = 00060₈ (bits 6-5, pitch command)

If i = 0:

TS = 01400₈ (bits 10-9, roll command)

If RCSFLAGS \wedge TS ≠ 0: (i.e. forced firing from RHC)

TS = - EDOT

Proceed to "JTIME"

If WHICHDAP = 1: (Tag here "CHECKDAP")

$TS_2 = 0$

$TAU_i = 0$

Proceed to 5th line of "JTIME"

JTIME

$TS_2 = JdM_i TS$ (B24 docked, B20 docked)

If WHICHDAP = 0:

Shift TS_2 right 4 places (gives scaling of B24 cs)

$TAU_i = TS_2$, with magnitude limited $< 2^{10}$ centi-seconds

If $i > 0$:

If WHICHDAP = 1:

$TS = i - 1$

$TAUZ_{TS} = C_{ecp_{TS}} TS_2$, with magnitude limited $< 2^{10}$ cs
(for $TS = 1$, load TAU_Y)

$i = i - 1$

Proceed to "JLOOP"

If WHICHX2 = 0:

T5LOC = "JETSLECT"

If WHICHX2 = 2:

T5LOC = "DKJSLECT"

Resume

JETSLECT Entered for undocked (to OWS) DAP

$TS = 6 - T5TIME$

If $TS \leq 0$:

$TS = 2$

Set TIME5 to cause program interrupt #2 in TS centi-seconds

$TIME6 = K_{14ms}$ (hence jets desired for ~86-100 ms on for 100 ms)

Perform "C13STALL"

Set bit 15(TIME6 Count Enable) of channel 13 = 1

TS = bits 12-7 of - (channel 31) (gives "true" translation bits)

If TS = 0:

XNDX1 = 0

XNDX2 = 0

YNDX = 0

ZNDX = 0

If TS \neq 0:

XNDX1 = bits 8-7 of TS, shifted right 6 places (to bits 2-1)

XNDX2 = XNDX1

YNDX = bits 10-9 of TS, shifted right 8 places (to bits 2-1)

ZNDX = bits 12-11 of TS, shifted right 10 places (to bits 2-1)

If bit 14 of DAPDATR1 = 0: (means LM-off)

ATTKALMN = -2

If bit 14 of DAPDATR1 = 1: (means LM-on)

ATTKALMN = -3

If XTRANS = 1:

XNDX1 = 0

If XTRANS = -1:

XNDX2 = 0

If $\text{TAU}_1 = 0$:

PINDEX = 0

If $\text{TAU}_1 > 0$:

PINDEX = 1

If $\text{TAU}_1 < 0$:

PINDEX = 2

If RACFAIL = 0:

TS = $K_{\text{xtndx}_{\text{XNDX1}}}$

If RACFAIL $>$ 0:

TS = 9

If RACFAIL < 0:

TS = 12

TS = TS + PINDEX

PWORD1 = bits 10, 9, and 4-1 of $K_{pytab_{TS}}$

NPJETS = bits 10-9 of PWORD1, shifted right 8 places (to bits 2-1)

If $TAU_2 = 0$:

YINDEX = 0

If $TAU_2 > 0$:

YINDEX = 1

If $TAU_2 < 0$:

YINDEX = 2

If RBDFAIL = 0:

TS = $K_{xtndx_{XNDX2}}$

If RBDFAIL > 0:

TS = 9

If RBDFAIL < 0:

TS = 12

TS = TS + YINDEX

YWORD1 = bits 12, 11, and 8-5 of $K_{pytab_{TS}}$

NYJETS = bits 12-11 of YWORD1, shifted right 10 places (to bits 2-1)

If $TAU_0 = 0$:

RINDEX = 0

If $TAU_0 > 0$:

RINDEX = 1

If $TAU_0 < 0$:

RINDEX = 2

If ACORBD \geq +0, proceed to "BDROLL"

If RACFAIL = 0:

$$TS = K_{xtndx_{YNDX}}$$

If RACFAIL > 0:

$$TS = 9$$

If RACFAIL < 0:

$$TS = 12$$

$$TS = TS + RINDEX$$

$$RWORD1 = \text{bits } 11-5 \text{ of } K_{rtab_{TS}}$$

If ZNDX ≤ 0:

$$NRJETS = (\text{bits } 11-9 \text{ of } RWORD1, \text{ shifted right } 8 \text{ places}) - 2$$

Proceed to "ROLLTIME"

If RBDFAIL = 0:

$$TS = 0$$

If RBDFAIL > 0:

$$TS = 3$$

If RBDFAIL < 0:

$$TS = 6$$

$$TS = TS + K_{xnlndx_{ZNDX}}$$

$$TS_1 = RWORD1 + (\text{bits } 11-9, 4-1 \text{ of } K_{yztab_{TS}})$$

$$NRJETS = (\text{bits } 11-9 \text{ of } TS_1, \text{ shifted right } 8 \text{ places}) - 4$$

If NRJETS ≠ 0: (have a net roll torque)

$$RWORD1 = TS_1$$

Proceed to "ROLLTIME"

If $\text{TAU}_0 = 0$: (i.e. no roll requested)

$\text{RWORD1} = \text{TS}_1$

Proceed to "ROLLTIME"

$\text{NRJETS} = (\text{bits } 11-9 \text{ of } \text{RWORD1}, \text{ shifted right } 8 \text{ places}) - 2$

Proceed to "ROLLTIME" (Z-translation ignored in favor of roll)

BDROLL

If $\text{RBDFAIL} = 0$:

$\text{TS} = K_{\text{xtndx}_{\text{ZNDX}}}$

If $\text{RBDFAIL} > 0$:

$\text{TS} = 9$

If $\text{RBDFAIL} < 0$:

$\text{TS} = 12$

$\text{TS} = \text{TS} + \text{RINDEX}$

$\text{RWORD1} = \text{bits } 14-12 \text{ and } 4-1 \text{ of } K_{\text{rtab}_{\text{TS}}}$

If $\text{YNDX} = 0$:

$\text{NRJETS} = (\text{bits } 14-12 \text{ of } \text{RWORD1}, \text{ shifted right } 11 \text{ places}) - 2$

Proceed to "ROLLTIME"

If $\text{RACFAIL} = 0$:

$\text{TS} = 0$

If $\text{RACFAIL} > 0$:

$\text{TS} = 3$

If $\text{RACFAIL} < 0$:

$\text{TS} = 6$

$\text{TS} = \text{TS} + K_{\text{xnln dx}_{\text{YNDX}}}$

$\text{TS}_1 = \text{RWORD1} + (\text{bits } 14-12, 8-5 \text{ of } K_{\text{yztab}_{\text{TS}}})$

NRJETS = (bits 14-12 of TS₁, shifted right 11 places) - 4

If NRJETS ≠ 0: (have a net roll torque)

RWORD1 = TS₁

Proceed to "ROLLTIME"

If TAU₀ = 0: (no roll requested)

RWORD1 = TS₁

Proceed to "ROLLTIME"

NRJETS = (bits 14-12 of RWORD1, shifted right 11 places) - 2

Proceed to "ROLLTIME" (Y-translation ignored in favor of roll)

ROLLTIME

If TAU₀ = 0:

DFT₀ = K_{dftm}NRJETS

RWORD2 = RWORD1

BLASTO+0 = 0

Proceed to "PITCHTIM"

BLASTO+0 = K_{njt}NRJETS TAU₀

TS = BLASTO+0 + K_{mptlsc}

If TS > 0:

DFT₀ = K_{dftm}NRJETS

TAU₀ = TAU₀ - DFT₀

BLASTO+0 = K_{ptlsc}

If TS ≤ 0:

Limit BLASTO+0 > K_{14ms}

DFT₀ = NRJETS BLASTO+0

TAU₀ = 0

If YNDX = 0:

RWORD2 = 0

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

If ACORBD \leq -0:

If |RACFAIL| $>$ 0:

RWORD2 = 0

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

TS = $K_{xtndx_{YNDX}}$

RWORD2 = bits 11-5 of $K_{rtab_{TS}}$

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

If RACFAIL = 0:

TS = 0

If RACFAIL $>$ 0:

TS = 3

If RACFAIL \leq 0:

TS = 6

TS = TS + $K_{xnlndx_{YNDX}}$

RWORD2 = bits 14-12, 8-5 of $K_{yztab_{TS}}$

NRJETS = (bits 14-12 of RWORD2, shifted right 11 places) - 2

DFT₀ = DFT₀ + NRJETS (K_{ptlsc} - BLASTO+0)

If ZNDX = 0, proceed to "PITCHTIM"

Proceed to "ACBD2Z"

ACBD2Z

If ACORBD \geq +0:

If (RBDFAIL) $>$ 0, proceed to "PITCHTIM"

$$TS = K_{xtndx_{ZNDX}}$$

$$RWORD2 = RWORD2 + (\text{bits } 14-12, 4-1 \text{ of } K_{rtab_{TS}})$$

Proceed to "PITCHTIM"

If RBDFAIL = 0:

$$TS = 0$$

If RBDFAIL $>$ 0:

$$TS = 3$$

If RBDFAIL $<$ 0:

$$TS = 6$$

$$TS = TS + K_{xnlnx_{ZNDX}}$$

$$RWORD2 = RWORD2 + (\text{bits } 11-9, 4-1 \text{ of } K_{yztab_{TS}})$$

$$NRJETS = (\text{bits } 14-9 \text{ of } RWORD2, \text{ shifted right } 8 \text{ places}) - 2$$

$$DFT_0 = DFT_0 + NRJETS (K_{ptlsc} - BLAST0+0)$$

Proceed to "PITCHTIM"

PITCHTIM

If $TAU_1 = 0$:

$$DFT_1 = 0$$

$$PWORD2 = PWORD1$$

$$BLAST1+0 = 0$$

Proceed to "YAWTIME"

If $TAU_1 <$ 0:

$$NPJETS = - NPJETS$$

$$\text{BLAST1+0} = K_{\text{njt}} \text{NPJETS} \text{TAU}_1$$

$$\text{TS} = \text{BLAST1+0} + K_{\text{mptlsc}}$$

If $\text{TS} > 0$:

$$\text{DFT}_1 = K_{\text{dftm}} \text{NPJETS}$$

$$\text{TAU}_1 = \text{TAU}_1 - \text{DFT}_1$$

$$\text{BLAST1+0} = K_{\text{ptlsc}}$$

If $\text{TS} \leq 0$:

$$\text{Limit } \text{BLAST1+0} \gg K_{14\text{ms}}$$

$$\text{DFT}_1 = \text{NPJETS} \text{BLAST1+0}$$

$$\text{TAU}_1 = 0$$

$$\text{TS} = 0$$

If $\text{RACFAIL} = 0$:

$$\text{TS} = K_{\text{xtndx}} \text{XNDX1}$$

$$\text{PWORD2} = \text{bits } 10, 9, \text{ and } 4-1 \text{ of } K_{\text{pytab}_{\text{TS}}} \quad (\text{is } 0 \text{ for } \text{TS} = 0)$$

Proceed to "YAWTIME"

YAWTIME

If $\text{TAU}_2 = 0$:

$$\text{DFT}_2 = 0$$

$$\text{YWORD2} = \text{YWORD1}$$

$$\text{BLAST2+0} = 0$$

Proceed to "T6SETUP"

If $\text{TAU}_2 < 0$:

$$\text{NYJETS} = - \text{NYJETS}$$

$$\text{BLAST2+0} = K_{\text{njt}} \text{NYJETS} \text{TAU}_2$$

$$\text{TS} = \text{BLAST2+0} + K_{\text{mptlsc}}$$

If $TS > 0$:

$$DFT_2 = K_{dftm} NYJETS$$

$$TAU_2 = TAU_2 - DFT_2$$

$$BLAST2+0 = K_{ptlsc}$$

If $TS \leq 0$:

$$\text{Limit } BLAST2+0 \gg K_{14ms}$$

$$DFT_2 = NYJETS \cdot BLAST2+0$$

$$TAU_2 = 0$$

$TS = 0$

If $RBDFAIL = 0$:

$$TS = K_{xtndx} XNDX2$$

$YWORD2 = \text{bits } 12, 11, \text{ and } 8-5 \text{ of } K_{pytab_{TS}}$ (is 0 for $TS = 0$)

Proceed to "T6SETUP"

T6SETUP

$$BLAST0+1 = 0$$

$$BLAST1+1 = 4$$

$$BLAST2+1 = 11$$

If $BLAST1+0 - BLAST0+0 \leq 0$:

Set $BLAST0 = BLAST1$ and $BLAST1 = BLAST0$

If $BLAST2+0 - BLAST1+0 \leq 0$:

Set $BLAST1 = BLAST2$ and $BLAST2 = BLAST1$

If $BLAST1+0 - BLAST0+0 \leq 0$:

Set $BLAST0 = BLAST1$ and $BLAST1 = BLAST0$

(now ordered $BLAST0+0 \leq BLAST1+0 \leq$

$$BLAST2+0 = BLAST2+0 - BLAST1+0 \quad \quad \quad BLAST2+0)$$

$$BLAST1+0 = BLAST1+0 - BLAST0+0$$

T5LOC = "RCSATT"
Set bit 1 of RCSFLAGS = 0
T5PHASE = -0
Resume

T6START

If |TIME6| > 0, Resume
If WHICHDAP = 1:
 Proceed to "DKT6"
If bit 1 of RCSFLAGS = 0:
 Set bit 1 of RCSFLAGS = 1
 Channel 6 = bits 8-1 of RWORD1
 Channel 5 = bits 8-1 of (PWORD1 + YWORD1)
If BLASTO+0 > 0:
 TIME6 = BLASTO+0
 BLASTO+0 = 0
 Perform "C13STALL"
 Set bit 15(TIME6 count enable) of channel 13 = 1
 Resume
If BLASTO+0 = 0:
 TS = BLASTO+1
 Perform "REPLACE"
 BLASTO+0 = - 1/16
If BLAST1+0 > 0:
 TIME6 = BLAST1+0
 BLAST1+0 = 0
 Perform "C13STALL"
 Set bit 15(TIME6 count enable) of channel 13 = 1
 Resume

If BLAST1+0 = 0:

TS = BLAST1+1

Perform "REPLACE"

BLAST1+0 = -1/16

If BLAST2+0 > 0:

TIME6 = BLAST2+0

BLAST2+0 = 0

Perform "C13STALL"

Set bit 15 (TIME6 count enable) of channel 13 = 1

Resume

If BLAST2+0 = 0:

TS = BLAST2+1

Perform "REPLACE"

BLAST2+0 = -1/16

Resume

REPLACE

If TS = 0: (roll information)

Channel 6 = bits 8-1 of RWORD2

Return

If TS = 4: (pitch information)

TS₁ = bits 8-5 of channel 5

Channel 5 = bits 8-1 of (TS₁ + PWORD2)

Return

If TS = 11: (yaw information)

TS₁ = bits 4-1 of channel 5

Channel 5 = bits 8-1 of (TS₁ + YWORD2)

Return

Quantities in Computations

See also list of major variables and list of routines

5AXW, 5BXW, 5PT, 5PW, 5WORD, 5YWT, 5YWW: See Digital Autopilot Docked Jet Selection.

61PT, 61PW, 61YWT, 61YWW, 62PT, 62PW, 62YWT, 62YWW, 6WORD: See Digital Autopilot Docked Jet Selection.

ACORBD: See Digital Autopilot Interface Routines.

ADB: See Digital Autopilot Interface Routines.

ADBVEL: Single precision value of $(ADB + K_{flat})$ sgn EDOT used in "JLOOP", B-1 revolutions. WHICH DAP

ADOT (ADOT₀, ADOT₁, ADOT₂): Filtered value of roll, pitch, and yaw body-angle rates (about "control axes"), scale factor B-3, units revolutions/decisecond (one decisecond is 0.1 seconds, the iteration rate of the DAP). The cells are OMEGAB in TVC DAP. Could also be considered scaled BO in "units" of 450°/second. Displayed by N56.

AERR: Single precision value of ERROR₁ about appropriate axis used in "JLOOP", scale factor B-1, units revolutions.

AERRVEL: Single precision value of AERR sgn EDOT used in "JLOOP", scale factor B-1, units revolutions.

AK (AK₀, AK₁, AK₂): See Digital Autopilot Interface Routines.

AMGB1, AMGB4, AMGB5, AMGB7, AMGB8: Single precision matrix elements which are used by the DAP to transform information from gimbal to control axes, scale factor BO, computed once per second by "AMBGUPDT". The matrix from gimbal to control axes may be written as:

$$\begin{bmatrix} 1 & \text{AMGB1} & 0 \\ 0 & \text{AMGB4} & \text{AMGB5} \\ 0 & \text{AMGB7} & \text{AMGB8} \end{bmatrix}$$

In order to "account for the roll displacements of the reaction jets with respect to navigation base coordinates", the outer gimbal angle (CDU_x) used to compute the matrix elements is decremented by K_{quadan}, thus giving the "control axes" result.

ANGREF: Value of reference angle (THETAD or CPHIX) used in "KMATRIX" to compute display information of total attitude error (if required), scale factor B-1, units revolutions.

ATTKALMN: Single precision quantity, scale factor B14, used to select appropriate gains in "RCSATT". To initialize the filter, the cell is set to 11 (decremented to 10 before used) in "REDAP", and decremented in "T5PHASE2" (with subsequent performance of further computations, including jet firings, bypassed until ATTKALMN = 0). A value of 0 is the normal undocked steady-state filter gain value; a setting of -1 is used for high-rate (RATEINDX = 6) automatic maneuvers (set in "T5PHASE2" if bit 15 of RCSFLAGS = 1 and undocked or if RHC with RATEINDX = 6); a setting of -2 is used (set in "JETSLECT") if IM-off translation hand controller inputs are provided; a setting of -3 is used if IM-on translation inputs are provided; a setting of -4 is used for normal docked (to OWS) steady-state filter gains; and a setting of -5 is set in "BURBLE" for X translation inputs in docked DAP.

ATTSEC: Single precision quantity, scale factor B14, initialized to 0 in "REDAP" and used in "KMATRIX" to cause "AMBGUPDT" to be established once a second (every tenth entrance to "KMATRIX").

BIAS (BIAS₀, BIAS₁, BIAS₂): Set of single precision angles, scale factor B-1, units revolutions, added to attitude errors to provide additional lead and prevent overshoot when starting/stopping automatic maneuvers. They are added to ERROR (used as the attitude error in "JLOOP") if HOLDFLAG is negative, and the values are computed in "LOCSKIRT" and "MANUSTOP" (where set 0). Except for R60 maneuvers, value normally 0.

BLASTO, BLAST1, BLAST2: Set of 3 quantities handled double precision, used for RCS jet burn time control. Before "T6SETUP" is entered, BLASTO is associated with roll, BLAST1 with pitch, and BLAST2 with yaw. The most significant halves of each word have required firing time for RWORD1, PWORD1, and YWORD1 respectively, scale factor B10, units centi-seconds. The least significant halves contain an index which identifies the axis in the most significant half (set at start of "T6SETUP" to 0, 4, and 11 respectively for convenience in the coding). Cells subsequently arranged in order of increasing time durations, then replaced with time intervals. In "T6START", most significant half set +0 if associated time in TIME6, and -1 (B14, i.e. -1/16 B10) after the expiration of that interval.

C_{ecp0}, C_{ecp1}: See Digital Autopilot Docked Jet Selection.

C_{jdm1dckd}, C_{jdmldckd}, C_{jdm2dckd}: Single precision erasable memory constants, program notations "J/MDCKD", "J/MLDCKD", and "J/M2DCKD" respectively, used in "REDAP" to initialize JdM. Scale factor is B27, units of (revolutions/(deci-second centi-second))⁻¹ (see JdM). They are the reciprocal of average roll, pitch, and yaw acceleration of a jet couple respectively, and are employed only in docked (to OWS) DAP.

$C_{kmjdckd}$, $C_{kmjldckd}$, $C_{kmj2dckd}$: Single precision erasable memory constants, program notations "KMJDCKD", "KMJLDCKD", and "KMJ2DCKD" respectively, used in "REDAP" to initialize KMJ. Scale factor is B-13, units of (revolutions/(deci-second centi-second)) (see KMJ). They correspond to the one-jet average roll, pitch, and yaw acceleration respectively, and are employed only in docked (to OWS) DAP.

C_{slope2_0} : Single precision erasable memory constant, program notation "SLOPE2", scale factor B-2, units (revolutions/deci-second)/revolution, used for undocked DAP phase plane computations.

C_{slope2_1} : Single precision erasable memory constant, program notation "SLOPE2 +1", scale factor B-2, units (revolutions/deci-second)/revolution, used for docked DAP phase plane computations.

$C_{wlhdslop_0}$: Single precision erasable memory constant, program notation "WLH/SLOP", scale factor B-1, units revolutions, used in undocked DAP phase plane computations (nominally is sum of rate limit and half-width, divided by SLOPE, hence the notation).

$C_{wlhdslop_1}$: Single precision erasable memory constant, program notation "WLH/SLOP +1", scale factor B-1, units revolutions, used in docked DAP phase plane computations.

$C_{wlmhdslp_0}$: Single precision erasable memory constant, program notation "WL-H/SLP", scale factor B-1, units revolutions, used in undocked DAP phase plane computations (nominally is rate limit minus half-width, divided by SLOPE, hence the notation).

$C_{wlmhdslp_1}$: Single precision erasable memory constant, program notation "WL-H/SLP +1", scale factor B-1, units revolutions, used in docked DAP phase plane computations.

C31FLWRD: Single precision flagword used for backup of bits 15-13 of channel 31 (by bits 15-13, i.e. digit "A") and bits 5-4 of channel 33 (by bits 5-4, i.e. digit "D"). It can be loaded by NO1 means (address 0373_g). For digit A, a value of 0 (or 4) means should use bits 15-13 of channel 31; values of 1-3 mean G&N Control (Free, Attitude Hold, and Auto respectively); values of 5-7 mean SCS Control. For digit D, a value of 0 means use bits 5-4 of channel 33; of 1-3 mean optics CMC, Zero, and Manual respectively (values of 4-7 mean the same as 0-3 respectively). Should be pad-loaded to zero; digits B, C, and E ("ABCDE") are ignored.

CDUXD: See Attitude Maneuvers (used in "KMATRIX" to update THETADX if HOLDFLAG is negative).

CH31TEMP: Single precision cell used in "T5PHASE2" to contain the previous value of channel 31 inputs (bits 6-1 are rotational hand controller inputs) if bits 6-1 change, in order to detect binary 1 → 0 transitions (logic 0 to 1) for RHC control purposes: binary 0 → 1 are also detected, of course. It is initialized to 77777_g by "REDAP". Can be set to cause "forced firings" for docked DAP (e.g. in "NEWANGL", "STOPRATE", and "CRS61.2A").

CHANTEMP: Single precision octal cell used in Free Mode (at the start of "AHFNOROT") to retain previous value of channel 32 input (cf. CH31TEMP), for use in providing minimum impulse control. It is initialized to 77777_8 in "REDAP".

CPHIX: Set of reference angles for attitude error (AK) display, scale factor B-1, units revolutions. They can be loaded by N17, or set to present CDU values by verb 60. Input of a verb 63 causes them to be used to compute AK. They are single precision.

DAPZRUP: See Digital Autopilot Docked Jet Selection.

DELCDU: See Attitude Maneuvers (used in "KMATRIX" to increment CDUXD values if HOLDFLAG negative).

DELTEMP: Value (computed double precision) of CDU angle change, converted to body rates, computed in "RCSATT". Scale factor is B-1, units revolutions/decisecond.

DFT (DFT₀, DFT₁, DFT₂): Single precision equivalent times, scale factor B10, units centi-seconds, giving the equivalent single-jet burning time computed for roll, pitch, and yaw axes respectively, used in the "RCSATT" rate filter to update body-angle rates.

DKDB: Single precision value of deadband to be used for docked DAP, scale factor B-1, units revolutions. It can be loaded via R2 of N89, and is also expected to form part of the prelaunch load.

DKRATE: Single precision value of maneuver rate (for R60 and RHC purposes) for docked DAP, scale factor B-9, units revolutions/decisecond. It can be loaded via R1 of N89, and is also expected to form part of the prelaunch load.

DRHQ (DRHQ₀, DRHQ₁, DRHQ₂): Value of body rate used as input to ADOT filter, scale factor B-1, units revolutions/decisecond, computed in "RCSATT" as the difference between measured (from CDU data) and ADOT body rates, plus a weighted estimate of previous DRHQ.

EDOT: Value of attitude rate error used in "JLOOP", scale factor B-3, units revolutions/decisecond (set to ADOT_i, decremented by WBODY_i if HOLDFLAG \neq 0). Its absolute value is stored in EDOTVEL cell.

ERROR (ERROR₀, ERROR₁, ERROR₂): Single precision value of attitude error for automatic maneuvers or attitude hold (computed in "AHFNOROT"), or RHC maneuvers (loaded with MERROR in "T5PHASE2"), used to load AERR at the start of "JLOOP". Scale factor is B-1, units are revolutions. Set 0 if free mode inputs are made, and can be loaded into AK (- ERROR) if bit 9(NEEDLFLG) of FLAGWRDO = 0 (e.g. V61).

HOLDFLAG: See Digital Autopilot Interface Routines.

IMODES33: See IMU Computations.

INHIBIT: See Digital Autopilot Docked Jet Selection.

JdM (JdM₀, JdM₁, JdM₂): See Digital Autopilot Interface Routines.

K_{14ms}: Single precision constant, program notation "=14MS", scale factor B10, units centi-seconds. Value is 23×2^{-14} , corresponding to about 14.4 milliseconds.

K_{200mst6_i}: See Digital Autopilot Docked Jet Selection.

K_{50mst6_i} (i = -1 to 2): Set of single precision constants, program notation (i = 0) "50MST6", scale factor B10, units centi-seconds. Values for i = -1 and i = 2 are zero. For i = 0, value is 80×2^{-14} (corresponding to $5 \times 2^4 \times 2^{-14} = 5 \times 2^{-10}$, where first term is value in centi-seconds and second is scale factor). For i = 1, value is -80×2^{-14} (corresponding to - 5 centi-seconds).

K_{bt9}: Single precision constant, program notation "BIT9", scale factor B10, units centi-seconds. Value is 00400_8 , corresponding to 16 centi-seconds.

K_{dftm_i}: Set of single precision constants, program notation "DFTMAX", scale factor B10, units centi-seconds. Values are:

<u>i</u>	<u>Value</u>	<u>True Value</u>
-3	-480×2^{-14}	-0.3 seconds
-2	-320×2^{-14}	-0.2 seconds
-1	-160×2^{-14}	-0.1 seconds
0	0	0.0 seconds
1	160×2^{-14}	0.1 seconds
2	320×2^{-14}	0.2 seconds
3	480×2^{-14}	0.3 seconds

K_{flat₀}: Single precision constant, program notation "FLAT", scale factor B-1, units revolutions, used to specify "flat region" in "JLOOP" (the offset of the dead-zone line when the rate error changes polarity). Value is 4×2^{-14} , corresponding to about 0.044_0 .

K_{flat₁}: Single precision constant, program notation "FLAT +1", scale factor B-1, units revolutions (see K_{flat₀}). Value is 4×2^{-14} , or about 0.044_0 .

K_{gnl_i}: Set of single precision constants, program notation (i = 0) "GAIN1", scale factor B0, used in "RCSATT" to obtain a new value of DRHQ (from samples taken 10 times/second). The value of the index is given by the contents of ATTKALMN. In the table below, the "True Value" obtained by converting the octal memory information to decimal.

<u>i</u>	<u>Value</u>	<u>True Value</u>
10	0.9342	0.93420
9	0.8151	0.81512
8	0.6933	0.69330
7	0.5970	0.59698
6	0.5223	0.52228
5	0.4634	0.46338
4	0.4161	0.41608
3	0.3774	0.37738
2	0.3452	0.34521
1	0.3180	0.31799
0	0.0640	0.06403
-1	0.2112	0.21118
-2	0.8400	0.84003
-3	0.2112	0.21118
-4	0.0640	0.06403
-5	0.0640	0.06403

K_{gn2_i} : Set of single precision constants, program notation ($i = 0$) "GAIN2", scale factor B_0 , used in "RCSATT" to obtain a new value of \underline{ADOT} (from \underline{DRHQ} information acquired 10 times/second). The value of the index is given by the contents of $\underline{ATTKALMN}$. In the table below, the "True Value" obtained by converting the octal memory information to decimal.

<u>i</u>	<u>Value</u>	<u>True Value</u>
10	0.8683	0.86829
9	0.4817	0.48169
8	0.2955	0.29547
7	0.1985	0.19849
6	0.1422	0.14221
5	0.1069	0.10687
4	0.0832	0.08319
3	0.0666	0.06659
2	0.0545	0.05450
1	0.0454	0.04541
0	0.0016	0.00159
-1	0.0174	0.01740
-2	0.3600	0.35999
-3	0.0174	0.01740
-4	0.0016	0.00159
-5	0.0016	0.00159

$K_{m75degs}$: Single precision constant, program notation "-75DEGS", scale factor B_{-1} , units revolutions. Nominal value is -0.41666 , corresponding to approximately -75° . Octal value is 62524_8 , which corresponds (after a one-bit correction reflecting formation of the absolute value, but ignoring twos complement) to -15254_8 , or about -75.015° .

K_{mantab_i} ($i = 0-7$): Set of single precision constants, program notation ($i = 0$) "MANTABLE", scale factor B-9, units revolutions/decisecond. The single precision constants are shifted right (double precision) by 6 places for loading into WBODY and by 8 places for decrementing MERROR in "T5PHASE2". The nominal value times 3600/512 gives the value in degrees/second in the table below.

<u>i</u>	<u>Value</u>	<u>Corresponds to</u>
0	0.0071111	0.05°/sec
1	-0.0071111	-0.05°/sec
2	0.0284444	0.20°/sec
3	-0.0284444	-0.20°/sec
4	0.0711111	0.50°/sec
5	-0.0711111	-0.50°/sec
6	0.2844444	2.00°/sec
7	-0.2844444	-2.00°/sec

K_{mintau_i} : Set of single precision constants, program notation ($i = 0$) "MINTAU", scale factor B10, units centi-seconds. Values for $i = 0$ and $i = 3$ are zero; for $i = 1$, value is 23×2^{-14} (about 14.4 milliseconds), and for $i = 2$, value is -23×2^{-14} (about -14.4 ms).

K_{mptlsc} : Single precision constant, program notation "=-.1SEC", scale factor B10, units centi-seconds. Value is -160×2^{-14} , corresponding to -0.1 second.

K_{njt_i} : Set of single precision constants, program notation ($i = 0$) "NJET", scale factor B0, used to correct jet burn times (which are derived based on one-jet operation) for the number of jets actually to be used, and also to compensate for polarity of TAU (negative for negative jets). Values are:

<u>i</u>	<u>Value</u>
-3	-0.333333
-2	-0.5
-1	-0.999999
0	0.0
1	0.999999
2	0.5
3	0.333333

K_{oct240} : Single precision constant, program notation "OCT240", scale factor B10, units centi-seconds. Value is 00240_g , corresponding to ten centi-seconds.

K_{p240} : Single precision constant, program notation "=.24", scale factor B-2, value 0.24. Used in "REDAP" as initial condition for SLOPE for undocked DAP. Value corresponds to $(-1) \times -0.6 \times 0.1 \times 2^2$, where first term is an equation factor, second is slope of attitude rate error limit line vs. attitude error (in units of rev/sec per rev), third converts to deci-seconds, and fourth is scale factor.

K_{p24_1} : Single precision constant, program notation " $=.24 +1$ ", scale factor B-2, value 0.12. Used in "REDAP" as initial condition for SLOPE for docked DAP. Value corresponds to $(-1) \times -0.3 \times 0.1 \times 2^2$, where terms have same significance as for K_{p24_0} .

K_{ptlsc} : Single precision constant, program notation " $=+.1SEC$ ", scale factor B10, units centi-seconds. Value is 160×2^{-14} , corresponding to 0.1 second.

K_{pytab_i} : Single precision set of octal constants, program notation "PYTABLE", giving information on pitch and yaw jets selected, together with the number of jets used for the function. See information below.

K_{quadan} : Single precision constant, program notation "QUADANGL", scale factor B-1, units revolutions, used to account for the roll displacements of the reaction jets with respect to navigation base coordinates. Value is 660×2^{-14} , corresponding to about 7.251^0 .

K_{rtab_i} : Single precision set of octal constants, program notation ($i = 0$) "RTABLE", giving information on roll jets selected (for AC or BD quads), together with number of roll jets (with polarity). See information below.

K_{wl_0} : Constant, program notation "WL", scale factor B-3, units revolutions/decisecond, used for undocked DAP. Value is 0.0008888888, corresponding to about 0.4^0 /second (nominal rate limit).

K_{wl_2} : Constant, program notation "WL +2", scale factor B-3, units revolutions/decisecond, used for docked DAP. Value is 0.0002222222, corresponding to about 0.1^0 /second (nominal rate limit).

K_{wlh_0} : Constant, program notation "WLH", scale factor B-3, units revolutions/decisecond, used for undocked DAP. Value is 0.0011111111, corresponding to about 0.5^0 /second (nominal rate limit plus half-width of 0.1^0 /second).

K_{wlh_2} : Constant, program notation "WLH +2", scale factor B-3, units revolutions/decisecond, used for docked DAP. Value is 0.0003333333, corresponding to about 0.15^0 /second (nominal rate limit plus half-width of 0.05^0 /second).

K_{wlmh_0} : Constant, program notation "WLMH", scale factor B-3, units revolutions/decisecond, used for undocked DAP. Value is 0.0006666666, corresponding to about 0.3^0 /second (nominal rate limit minus half-width of 0.1^0 /second).

K_{wlmh_2} : Constant, program notation "WLMH +2", scale factor B-3, units revolutions/decisecond, used for docked DAP. Value is 0.0001111111, corresponding to about 0.05^0 /second (nominal rate limit minus half-width of 0.05^0 /second).

K_{xnlndx_i} : Set of single precision constants, program notation "XLN1NDX", scale factor B14, used to provide index information from input data on control configuration. Value is equal to i for $i = 0, 1, \text{ and } 2$, and is equal to zero for $i = 3$.

K_{xtndx_i} : Set of single precision constants, program notation "XLNNDX", scale factor B14, used to provide index information for combining translation and rotation commands. Value is equal to $(3 i)$ for $i = 0, 1, \text{ and } 2$, and is equal to zero for $i = 3$.

K_{yztab_i} : Set of single precision constants, program notation "YZTABLE", giving information permitting the combination of Y or Z translations with roll, as well as number of jets giving a "net roll torque". See information below.

KMJ (KMJ_0, KMJ_1, KMJ_2): See Digital Autopilot Interface Routines.

MERROR: Value of integrated rate error used with RHC ("manual") control, scale factor B-1, units revolutions, computed in double precision (a single precision version is loaded into ERROR in "T5PHASE2" if manual RHC inputs are provided). The cells are initialized to 0 in "REDAP" and are also set 0 in "T5PHASE2" if bits 13-11 of RCSFLAGS are not all 0 (i.e. damping not done); they are updated in "RCSATT" by the measured (from CDU first differences) attitude rate, from which is subtracted desired rate (if damping done) in "T5PHASE2".

NEXT6INT: See Digital Autopilot Docked Jet Selection.

NPJETS: Single precision value of number of pitch jets to be fired, scale factor B14. Sign of TAU_1 added in "PITCHTIM".

NRJETS: Single precision value of number of roll jets to be fired (with sign indicating polarity), scale factor B14.

NYJETS: Single precision value of number of yaw jets to be fired, scale factor B14. Sign of TAU_2 added in "YAWTIME".

PINDEX: Single precision index parameter, scale factor B14, indicating the polarity of the pitch command desired: 0 if none, 1 if plus pitch, and 2 if minus pitch.

PMANNDX: See Digital Autopilot Interface Routines.

PWORD1, PWORD2: Single precision cells containing in bits 8-1 the pitch-command bits to be loaded in channel 5 when "T6START" is entered and after the pitch time duration has expired, respectively. The pitch time duration originally is in BLAST1+0.

RACFAIL: See Digital Autopilot Interface Routines.

RATEINDX: See Digital Autopilot Interface Routines.

RBDFAIL: See Digital Autopilot Interface Routines.

RCSFLAGS: See Digital Autopilot Interface Routines.

RHO (RHO₀, RHO₁, RHO₂): Value of previous CDU information (initialized in "REDAP") used to compute input to attitude rate filter in "RCSATT", scale factor B-1, units revolutions in twos complement.

RINDEX: Single precision index parameter, scale factor B14, indicating the polarity of the roll command desired: 0 if none, 1 if plus roll, and 2 if minus roll.

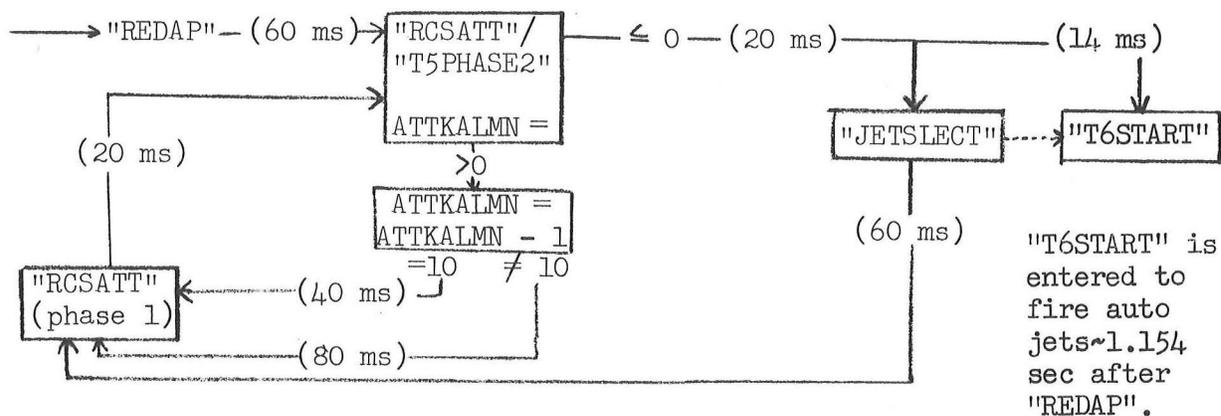
RMANNDX: See Digital Autopilot Interface Routines.

RWORD1, RWORD2: Single precision cells containing in bits 8-1 the roll-command bits (perhaps combined with Y and/or Z axis translation) to be loaded into channel 6 when "T6START" is entered and after the roll time duration has expired, respectively. The roll time duration originally is in BLASTO+0.

SLOPE: Single precision cell, scale factor B-2, units (revolutions/decisecond)/revolution, set to K_{p24i} in "REDAP" and not subsequently changed. It is a "variable" (i.e. in erasable memory) because the computer hardware divide order is designed to accept operands only from erasable memory.

T5LOC: See Digital Autopilot Interface Routines.

T5PHASE: Single precision cell, scale factor B14, used to control the program branching that takes place when program interrupt #2 transfers to "RCSATT". A positive non-zero value (of any magnitude) causes the DAP to be initialized; a +0 value causes "T5PHASE2" to be entered; a negative non-zero value (of any magnitude) causes a DAP "restart" (some of the initialization, i.e. HOLDFLAG, is bypassed), as set in "REDORCS"; and a -0 value causes the computations in "RCSATT" to be continued (to perform the rate filtering functions if appropriate). The continuation of "RCSATT" is also known as "phase 1" of the DAP, since the normal sequencing of the RCS (undocked) DAP is:



For docked DAP, "DKJSLECT" replaces "JETSLECT".

T5TIME: Single precision time, scale factor B14, units centi-seconds, used to compensate for delays in action on program interrupt #2, so as to increase the likelihood that the CDU angles for determining attitude rates will be sampled at uniform 0.1-second intervals. It is initialized in "REDAP" to 4, thus causing "RCSATT" (phase 1) to be entered (8 - 4 = 4 cs) after "T5PHASE2" is entered the first time, i.e. 10 cs after RHO initialized. The nominal contents at other times should be zero (it is reset to TIME5 after "RCSATT" phase 1 is entered).

T6LOC: See Digital Autopilot Interface Routines.

TAU (TAU₀, TAU₁, TAU₂): Single precision values of desired one-jet burn times to accomplish required rotations in roll, pitch, and yaw respectively, scale factor B10, units centi-seconds.

TAUY, TAUZ: See Digital Autopilot Docked Jet Selection.

THETADX (THETADX, THETADY, THETADZ): Single precision values, scale factor B-1, twos complement, units revolutions, of desired CDU angles in DAP, used to form attitude errors for "JLOOP". These cells should not be confused with THETAD (components THETAD_x etc.) of e.g. N22.

TIME5, TIME6: See Digital Autopilot Interface Routines.

TIMEHOLD: See Digital Autopilot Docked Jet Selection.

TRCOMPAC, TRCOMPBD: See Digital Autopilot Docked Jet Selection.

WBODY (WBODY₀, WBODY₁, WBODY₂): Values of desired angular rates, scale factor B-3, units revolutions/decisecond, used to modify the rate error information in "JLOOP" if HOLDFLAG ≠ 0. Cells are set in "NEWANGL" and "MANUSTOP" (to 0) to desired rates, and in "T5PHASE2" for RHC inputs. Also set in R61. Same cells used for OMEGAC in TVC DAP.

WHICHDAP: Single precision cell, scale factor B14, set in "REDAP" to 0 for undocked DAP and to 1 for docked DAP.

WHICHX2: Single precision cell, scale factor B14, set in "REDAP" to 0 for undocked DAP and to 2 for docked DAP. It is the least significant half of WHICHDAP.

XNDX1, XNDX2: Single precision index cells, scale factor B14, set at the start of "JETSLECT" to the complement of bits 8-7 (shifted right 6 places) of channel 31, i.e. the X-axis translation hand controller input. XNDX1 is the "AC Quad" translation index (reset 0 if XTRANS = 1, and used in conjunction with "pitch" information), and XNDX2 is the "BD Quad" translation index (reset 0 if XTRANS = -1, and used in conjunction with "yaw" information).

XTRANS: See Digital Autopilot Interface Routines.

YINDEX: Single precision index parameter, scale factor Bl4, indicating the polarity of the yaw command desired: 0 if none, 1 if plus yaw, and 2 if minus yaw.

YMANNDX: See Digital Autopilot Interface Routines.

YNDX: Single precision index cell, scale factor Bl4, set at the start of "JETSLECT" to the complement of bits 10-9 (shifted right 8 places) of channel 31, the Y-axis translation hand controller input.

YWORD1, YWORD2: Single precision cells containing in bits 8-1 the yaw-command bits to be loaded in channel 5 when "T6START" is entered and after the yaw time duration has expired, respectively. The yaw time duration originally is in BLAST2+0.

ZNDX: Single precision index cell, scale factor Bl4, set at the start of "JETSLECT" to the complement of bits 12-11 (shifted right 10 places) of channel 31, the Z-axis translation hand controller input.

Value of Jet Table Data for Undocked DAP

Information on pitch and yaw jets is stored in $K_{pytab,i}$, where i is in the range 0 - 14. Bits 12-11 give the number of yaw jets used to perform the rotation, and bits 10-9 give the number of pitch jets used. Bits 8-5 give the yaw output bits (to be loaded in the corresponding bit positions of channel 5), while bits 4-1 give the pitch output bits (to be loaded into the corresponding bits of channel 5).

The value of i depends on the failure information (or, if no failures, the X-translation requested) as well as the rotation desired.

$$i = 3a + b$$

- $a = 0$ for no X-translation
- $= 1$ for no failures, +X translation
- $= 2$ for no failures, -X translation
- $= 3$ for A (pitch) or B (yaw) failures
- $= 4$ for C (pitch) or D (yaw) failures

- $b = 0$ for no desired rotation (for PWORD2 and YWORD2,
- $= 1$ for + desired rotation $b = 0; a = 0$ if any
- $= 2$ for - desired rotation failures)

<u>i</u>	<u>K_{pytab}</u>	<u># of Yaw</u>	<u>Yaw Jet #</u>	<u># of Pitch</u>	<u>Pitch Jet #</u>
0	0000 ₈	0	---	0	---
1	5125 ₈	2	7,5	2	3,1
2	5252 ₈	2	6,8	2	2,4
3	0231 ₈	0	6,5	0	2,1
4	2421 ₈	1	5	1	1
5	2610 ₈	1	6	1	2
6	0146 ₈	0	7,8	0	3,4
7	2504 ₈	1	7	1	3
8	2442 ₈	1	8	1	4
9	0000 ₈	0	---	0	---
10	2421 ₈	1	5	1	1
11	2442 ₈	1	8	1	4
12	0000 ₈	0	---	0	---
13	2504 ₈	1	7	1	3
14	2610 ₈	1	6	1	2

<u>Jet #</u>	<u>Bit</u>	<u>Quad</u>	<u>Rotation</u>	<u>Translation</u>
1	1	C	+ Pitch	+X
2	4	A	- Pitch	+X
3	3	A	+ Pitch	-X
4	2	C	- Pitch	-X
5	5	D	+ Yaw	+X
6	8	B	- Yaw	+X
7	7	B	+ Yaw	-X
8	6	D	- Yaw	-X

Information on roll jets (for rotation combined with translation) is stored in K_{rtab}_i , where i is in the range 0 - 14. Bits 14-12 give the number of roll jets for using Quads B/D, while bits 11-9 give the number for using Quads A/C (the number in both cases is given by the value of the bits - 2, thus providing sign information). Bits 8-5 give the Quad A/C output (to be loaded into the corresponding bit positions of channel 6), while bits 4-1 give the Quad B/D output (likewise to be loaded into the corresponding bit positions of channel 6). The value of i is defined as for K_{pytab}_i , except that Y translation is associated with Quad A/C and the Z translation with Quad B/D.

<u>i</u>	<u>K_{rtab}</u>	<u># of A/C</u>	<u>A/C Jet #</u>	<u># of B/D</u>	<u>B/D Jet #</u>
0	11000 ₈	0	---	0	---
1	22125 ₈	2	15,13	2	11,9
2	00252 ₈	-2	14,16	-2	10,12
3	11231 ₈	0	14,13	0	10,9
4	15421 ₈	1	13	1	9
5	04610 ₈	-1	14	-1	10
6	11146 ₈	0	15,16	0	11,12
7	15504 ₈	1	15	1	11
8	04442 ₈	-1	16	-1	12
9	11000 ₈	0	---	0	---
10	15504 ₈	1	15	1	11
11	04610 ₈	-1	14	-1	10
12	11000 ₈	0	---	0	---
13	15421 ₈	1	13	1	9
14	04442 ₈	-1	16	-1	12

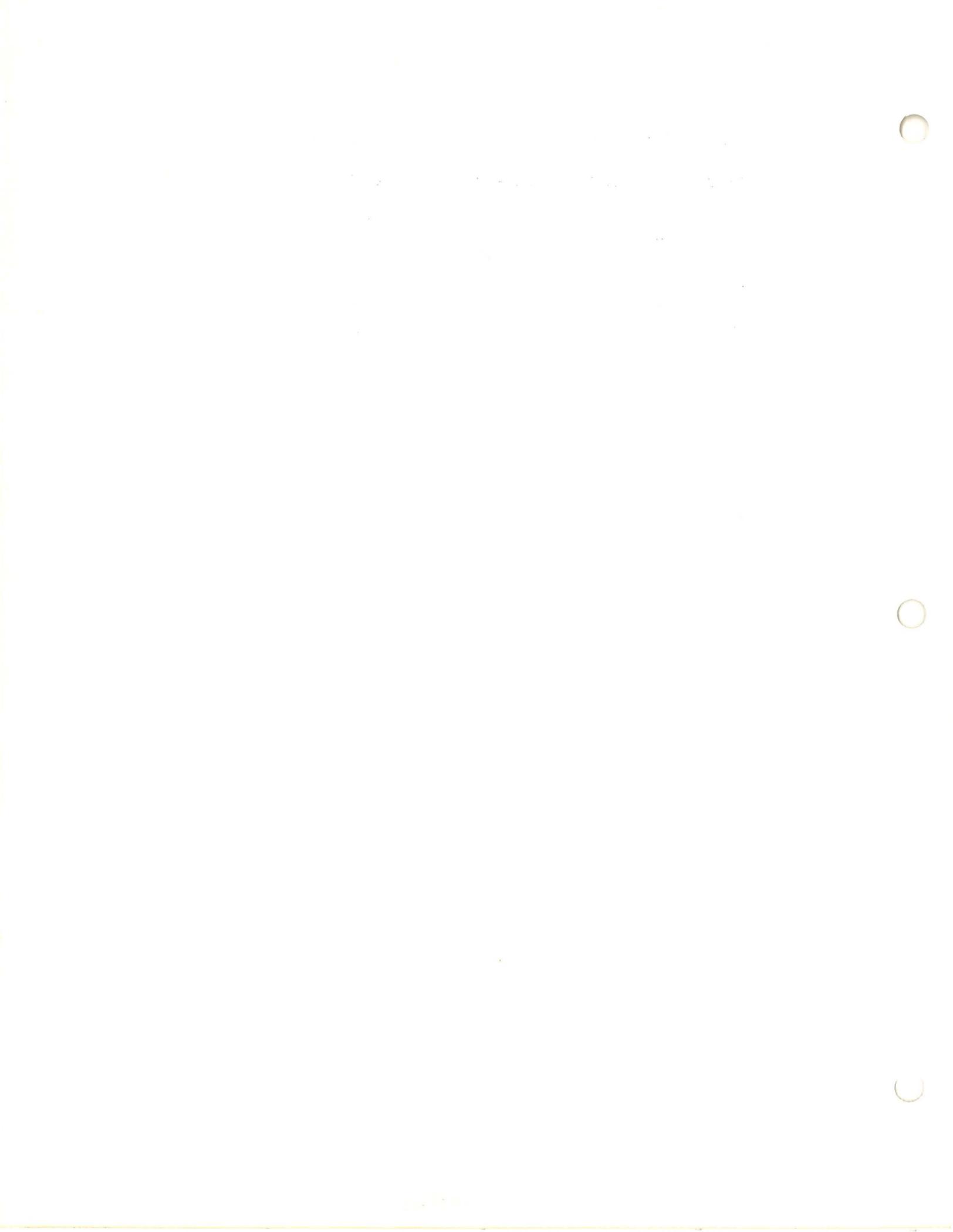
The information in K_{yztab}_i is in the same format as K_{rtab} , except that $i = 3c + d$, where $c = 0$ for no failures; $c = 1$ for A or B failures; and $c = 2$ for C or D failures. The quantity $d = 0$ for no translation; $d = 1$ for + translation; and $d = 2$ for - translation.

<u>i</u>	<u>K_{yztab}</u>	<u># of A/C</u>	<u>A/C Jet #</u>	<u># of B/D</u>	<u>B/D Jet #</u>
0	11000 ₈	0	---	0	---
1	11231 ₈	0	14,13	0	10,9
2	11146 ₈	0	15,16	0	11,12
3	11000 ₈	0	---	0	---
4	04610 ₈	-1	14	-1	10
5	15504 ₈	1	15	1	11
6	11000 ₈	0	---	0	---
7	15421 ₈	1	13	1	9
8	04442 ₈	-1	16	-1	12

The "# of A/C" and "# of B/D" information is used to check that the resulting roll torque on the vehicle is not forced to be zero when the desired roll rotation is non-zero. The K_{yztab} information permits the Quad not used for roll rotation to be used to satisfy translation commands.

The following roll-jet assignments are made (channel 6):

<u>Jet #</u>	<u>Bit</u>	<u>Quad</u>	<u>Rotation</u>	<u>Translation</u>
9	1	B	+ Roll	+Z
10	4	D	- Roll	+Z
11	3	D	+ Roll	-Z
12	2	B	- Roll	-Z
13	5	A	+ Roll	+Y
14	8	C	- Roll	+Y
15	7	C	+ Roll	-Y
16	6	A	- Roll	-Y



Digital Autopilot TVC Routines*

TVCDAPON Entered about 0.41 seconds after "IGNITION" or if
restart (from "CMDSOUT" for TVCPHASE = -1)

TTMP1 = 0

YDELOFF = 0

PDELOFF = 0

DELYBAR = 0

DELPBAR = 0

YERRB = 0

PERRB = 0

TVCDUMMY = 0 (not used)

TEMREG = 0

ROLLWORD = 0

ROLLFIRE = 0

YTMP_i = 0 (i = 6 - 1)

PTMP_i = 0 (i = 6 - 1)

OMEGAB = 0

OMEGAC_z = 0

OMEGAC_y = 0

T5LOC = "TVCINIT1"

Set TIME5 to cause program interrupt #2 in 0.01 second

Resume

TVCINIT1 Entered about 0.42 seconds from start of "IGNITION"

Perform "MASSPROP"

j = (bit 14 of DAPDATR1) (1 for LM on, 0 for LM off)

*NOTE: "LM" references are from Apollo 15 program, and have been
retained for mnemonic usefulness.

If j = 1: (IM on)

$$N1O_i = C_{hbn1O_i} \quad (i = 0 - 14)$$

If j = 0:

$$N1O_i = K_{csmn1O_i} \quad (i = 0 - 14)$$

$$KTLXdI = C_{ektlxdi_j}$$

Perform "S40.15"

If j = 1:

$$TS_1 = K_{8csd2}$$

If j = 0:

$$TS_1 = K_{4csd2}$$

$$T5TVCDT = 16384 - TS_1$$

Set bit 15(SWTOVER) of FLAGWRD9 = 0

$$KPRIME DT = 2 C_{ekprime_j} TS_1 \quad (\text{the 2 is due to scaling})$$

$$REPFRAC = C_{erefrac_j}$$

$$CNTR = K_{tcorr_j}$$

$$VCNTR = 19$$

$$PDELOFF_{sp} = PACTOFF$$

$$PCMD = PACTOFF$$

$$DELPBAR_{sp} = PACTOFF$$

$$YDELOFF_{sp} = YACTOFF$$

$$YCMD = YACTOFF$$

$$DELYBAR_{sp} = YACTOFF$$

If bit 13 of DAPDATR1 = 1: (IM off)

$$\text{If } \left| K_{ldattlim} \text{ ERRBTMP}+i \right| \gg 2^{-14}: \quad (i = 0, 1: \text{loaded with } -(\text{ERROR}_1, \text{ERROR}_2) \text{ in "IGNITION"})$$
$$\text{ERRBTMP}+i = K_{attlim} \text{ sgn ERRBTMP}+i$$

(If bit 13 of DAPDATR1 = 1):

PERRB_{sp} = ERRBTMP+0

YERRB_{sp} = ERRBTMP+1

Set bit 3 of RCSFLAGS = 1

Perform "NEEDLER"

Proceed to "TVCINIT4"

TVCINIT4 Can be entered from "CMDSOUT" if restart

TVCPHASE = +0

OGANOW = CDU_x

Call "TVCEXEC" in 0.50 seconds

T5LOC = "DAPINIT"

TIME5 = T5TVCDT

Resume

DAPINIT Entered about $(0.42 + \frac{1}{2} \text{ DAP cycle})$ sec after start of "IGNITION"

TIME5 = T5TVCDT + (T5TVCDT - 16384) (gives interrupt in one DAP cycle)

T5LOC = "PITCHDAP" (entered about $(0.42 + 1.5 \text{ DAP cycle})$ sec after start of "IGNITION": for IM-off, 0.48 second)

PCDUYPST = CDU_y

YCDUYPST = PCDUYPST

PCDUZPST = CDU_z (Note that on first pass yaw rates 50% too big)

YCDUZPST = PCDUZPST

Resume

SWICHOVR Entered from "STABLISH" or "CMDSOUT"

Inhibit interrupts (released when return to caller)

PHASETMP = TVCPHASE

TVCPHASE = -2

YTMPi = 0 (i = 6 - 1)

PTMPi = 0 (i = 6 - 1)

Set bit 15(SWTOVER) of FLAGWRD9 = 1

KTLXdI = C_{ektlxdi}₂

Perform "S40.15" (starting at second line)

KPRIMEDT = K_{fkprimdt}

REPPFRAC = K_{frepfrac}

PDELOFF = DELPBAR

YDELOFF = DELYBAR

N10_i = K_{lbn10_i} (i = 0 - 14)

TVCPHASE = PHASETMP

Return

TVCEXEC Entered first about 0.92 second after "IGNITION" (see "TVCINIT4")

If bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 \neq 10₂:

TVCEXPHS = 0

End of task

Call "TVCEXEC" in 0.5 seconds

OGAPAST = OGANOW

OGANOW = CDU_x

AK₀ = OGAD - OGANOW (ones complement difference formed)

OGA = - AK₀, rescaled to scale factor BO revolutions

Call "ROLLDAP" in 0.03 seconds

Perform "NEEDLER"

If VCNTR > 0:

VCNTR = VCNTR - 1, limited \geq +0 (Note lack of restart protection)

Proceed to "LSHOTCHK"

Perform "FIXCW" (VCNTR was +0)

Perform "S40.15"

VCNTR = 19

Proceed to "LSHOTCHK"

LSHOTCHK

If CNTR > 0:

TS₁ = CNTR - 1

TVCEXPHS = 4

Proceed to "CNTRCOPY"

If CNTR < 0: (Tag here "REPCHEK")

If REPFRAC \leq 0: (Set -0 in "S40.8" if low thrust)

TVCEXPHS = 0

End of task

TS₂ = REPFRAC

TVCEXPHS = 2

Proceed to third line of "TEMPSET"

If bit 7(ENGNFLG) of FLAGWRD5 = 0: (CNTR = 0; Tag "LSHOTOK")

TVCEXPHS = 0

End of task

TVCEXPHS = TVCEXPHS + 1 (sets to 1)

Proceed to "TEMPSET"

TEMPSET

$TS_2 = K_{fcorfrac}$

TVCEXPHS = TVCEXPHS + 1 (sets to 2)

If bit 13 of DAPDATR1 = 1: (means LM-off)

$TS_2 = 2 TS_2$

$TS_1 = -1$

$TS_3 = PDELOFF + TS_2 (DELPBAR - PDELOFF)_{sp}$

$TS_4 = YDELOFF + TS_2 (DELYBAR - YDELOFF)_{sp}$

TVCEXPHS = TVCEXPHS + 1 (sets to 3)

$PDELOFF = TS_3$

$PACTOFF = TS_3_{sp}$

$YDELOFF = TS_4$

$YACTOFF = TS_4_{sp}$

TVCEXPHS = TVCEXPHS + 1 (sets to 4)

Proceed to "CNTRCOPY"

CNTRCOPY

$CNTR = TS_1$

TVCEXPHS = 0

End of task

PITCHDAP Entered initially due to logic in "DAPINIT"; then
from "YAWDAP" logic.

T5LOC = "YAWDAP"

TIME5 = T5TVCDT

Set PCDUYPST = CDU_y and $TS = PCDUYPST$

MCDUYDOT = $TS - PCDUYPST$ (ones complement difference formed)

If $|K_{ldrtim} \text{MCDUYDOT}| \gg 2^{-14}$:

$\text{MCDUYDOT} = 0$

Set $\text{PCDUZPST} = \text{CDU}_z$ and $\text{TS} = \text{PCDUZPST}$

$\text{MCDUZDOT} = \text{TS} - \text{PCDUZPST}$ (ones complement difference formed)

If $|K_{ldrtim} \text{MCDUZDOT}| \gg 2^{-14}$:

$\text{MCDUZDOT} = 0$

$\text{ERRBTMP} = \text{PERRB} + \text{OMEGAC}_y$

$\text{OMEGAB}_y = -\text{MCDUYDOT} \text{COSCDUZ} \text{COSCDUX} - \text{MCDUZDOT} \text{SINCDUX}$

$\text{ERRBTMP} = \text{ERRBTMP} - \text{OMEGAB}_y$

If $|K_{lderrim} \text{ERRBTMP}| \gg 2^{-14}$:

$\text{ERRBTMP}_{sp} = K_{errim} \text{sgn} \text{ERRBTMP}$

$\text{TMP1} = \text{PTMP1}$

$\text{TMP3} = \text{PTMP3}$

$\text{TMP5} = \text{PTMP5}$

Perform "FWDFLTR"

$\text{CMDTMP} = \text{PDELOFF} + \text{CMDTMP}$

Round CMDTMP to single precision

If $|K_{ldactsat} \text{CMDTMP}| \gg 2^{-14}$:

$\text{CMDTMP} = K_{actsat} \text{sgn} \text{CMDTMP}$

$\text{TVCPITCH} = \text{TVCPITCH} + \text{CMDTMP} - \text{PCMD}$

Set bit 11 (Gate output from TVCPITCH) of channel 14 = 1

$\text{TMP2} = \text{PTMP2}$

$\text{TMP4} = \text{PTMP4}$

$\text{TMP6} = \text{PTMP6}$

Perform "PRECOMP"

$\text{DELBRTMP} = K_{emat} \text{DELPBAR} + K_{onemeat} \text{CMDTMP}$

Perform "PCOPY"

Resume

PCOPY

TVCPHASE = TVCPHASE + 1 (sets to 1)

PTMPi = TMPi (i = 1 - 6)

PERRB = ERRBTMP

AK₁ = ERRBTMP_{sp}

PCMD = CMDTMP

DELPBAR = DELBRTMP

TVCPHASE = TVCPHASE + 1 (sets to 2)

Return

YAWDAP Entered due to "PITCHDAP" logic.

T5LOC = "PITCHDAP"

TIME5 = T5TVCDT

Perform the following for i = Y(or y) and then i = Z (or z):

Set YCDUiPST = CDU₁ and TS = YCDUiPST

MCDUiDOT = TS - YCDUiPST (ones complement difference formed)

If $|K_{ldrtim} \text{MCDUiDOT}| \geq 2^{-14}$:

MCDUiDOT = 0

ERRBTMP = YERRB + OMEGAC_z

OMEGAB_z = MCDUYDOT COSCDUZ SINCDUX - MCDUZDOT COSCDUX

ERRBTMP = ERRBTMP - OMEGAB_z

If $|K_{lderrim} \text{ERRBTMP}| \geq 2^{-14}$:

ERRBTMP_{sp} = K_{errim} sgn ERRBTMP

TMP1 = YTMP1

TMP3 = YTMP3

TMP5 = YTMP5

Perform "FWDFLTR"

CMDTMP = YDELOFF + CMDTMP

Round CMDTMP to single precision

If $|K_{ldactsat} \text{ CMDTMP}| \geq 2^{-14}$:

$$\text{CMDTMP} = K_{actsat} \text{ sgn CMDTMP}$$

$$\text{TVCYAW} = \text{TVCYAW} + \text{CMDTMP} - \text{YCMD}$$

Set bit 12(Gate output from TVCYAW) of channel 14 = 1

$$\text{TMP2} = \text{YTMP2}$$

$$\text{TMP4} = \text{YTMP4}$$

$$\text{TMP6} = \text{YTMP6}$$

Perform "PRECOMP"

$$\text{DELBRTMP} = K_{emat} \text{ DELYBAR} + K_{onemeat} \text{ CMDTMP}$$

Perform "YCOPY"

Resume

YCOPY

$$\text{TVCPHASE} = \text{TVCPHASE} + 1 \quad (\text{sets to } 3)$$

$$\text{YTMP}_i = \text{TMP}_i \quad (i = 1 - 6)$$

$$\text{YERRB} = \text{ERRBTMP}$$

$$\text{AK}_2 = \text{ERRBTMP}_{sp}$$

$$\text{YCMD} = \text{CMDTMP}$$

$$\text{DELYBAR} = \text{DELBRTMP}$$

$$\text{TVCPHASE} = 0$$

Return

FWDFLTR

$$\text{DAP1} = \text{N10}_0 \text{ ERRBTMP} + \text{TMP1} \quad (\text{N10}_0 \text{ is "N10"})$$

$$\text{DAP2} = \text{N10}_5 \text{ DAP1} + \text{TMP3} \quad (\text{N10}_5 \text{ is "N20"})$$

If bit 14 of DAPDATR1 = 1: (LM attached)

$$\text{DAP3} = \text{N10}_{10} \text{ DAP2} + \text{TMP5} \quad (\text{N10}_{10} \text{ is "N30"})$$

If bit 14 of DAPDATR1 = 0: (LM off)

$$\text{DAP3} = \text{DAP2} \quad (\text{rescaled to scale factor B-3 revolutions})$$

$$\text{CMDTMP} = - \text{VARK DAP3}$$

Return

PRECOMP

TMP1 = N10₁ ERRBTMP - N10₃ DAP1 + TMP2 (N10₁ is "N11" and
N10₃ is "D11")

TMP2 = N10₂ ERRBTMP - N10₄ DAP1 (N10₂ is "N12" and
N10₄ is "D12")

TMP3 = N10₆ DAP1 - N10₈ DAP2 + TMP4

TMP4 = N10₇ DAP1 - N10₉ DAP2

If bit 13 of DAPDATR1 = 0: (IM on)

TMP5 = N10₁₁ DAP2 - N10₁₃ DAP3 + TMP6

TMP6 = N10₁₂ DAP2 - N10₁₄ DAP3

Return

REDOTVC

If TVCEXPHS > 0:

Call "EXRSTRT" in 0.09 seconds

If TVCPHASE < 0:

If bit 2 of TVCPHASE = 1: (i.e. -1, not -2)

PCMD = PACTOFF

YCMD = YACTOFF

If TVCPHASE = 1:

Perform "PCOPY" (starting at second line)

If TVCPHASE = 3:

Perform "YCOPY" (starting at second line)

Set bits 11(Disengage Optics DAC) and 8(TVC Enable) of channel 12 = 1

T5LOC = "ENABL2"

Set TIME5 to cause program interrupt #2 in 0.06 seconds

Resume

ENABL2

Set bit 2(Enable Optics CDU Error Counters) of channel 12 = 1

T5LOC = "CMDOUT"

Set TIME5 to cause program interrupt #2 in 0.02 seconds

Resume

CMDSOUT

TVCPITCH = PCMD - 0 (avoids loading with +0)

TVCYAW = YCMD - 0 (avoids loading with +0)

Set bits 12-11 (Gate outputs from TVCPITCH & TVCYAW) of channel
14 = 1

If TVCPHASE < 0:

If |TVCPHASE| - 1 = 0:

Proceed to "TVCDAPON"

Perform "SWICHOVR" (starting at 4th line)

Proceed to "TVCINIT4"

EXRSTRT

Proceed to appropriate point in "TVCEXEC" routine (just after
the point where TVCEXPHS was incremented to its present value)

ROLLDAP Called by "TVCEXEC" every 0.5 second (0.03 second after start
of "TVCEXEC")

OGARATE = K_{rrt} (OGANOW - OGAPAST) (ones complement difference)

If |ROLLFIRE| = 0:

If |TEMREG| ≠ 0:

TEMREG = 0

End of task

TS = $K_{db} - \frac{1}{2} \text{ldCONACC} \text{OGARATE}^2$

SGNRT = 1 sgn OGARATE (0 considered negative)

$$\text{DELOGA} = \text{OGA} - \text{TS SGNRT}$$

$$\text{SGNDOG} = 1 \text{ sgn DELOGA} \quad (0 \text{ considered negative})$$

If $\text{SGNRT sgn SGNDOG} \geq 0$:

$$\text{DELOGA} = \text{DELOGA} + K_{\text{db}} \text{ sgn SGNDOG}$$

$$\text{DELOGART} = - K_{\text{mcrat}} \text{ sgn SGNDOG}$$

Proceed to "ONROLL"

If $\text{OGARATE sgn SGNDOG} + K_{\text{maxim}} \leq 0$:

$$\text{RATDES} = - K_{\text{maxim}} \text{ sgn SGNDOG}$$

Proceed to "ROLLSET"

$$\text{TS} = (K_{\text{msope}} \text{ OGA} + \text{OGARATE}) \text{ sgn SGNDOG} - K_{\text{intercp}}$$

If $\text{TS} \leq 0$, proceed to "NOROLL"

If $\text{OGARATE sgn SGNDOG} + K_{\text{minim}} \leq 0$, proceed to "NOROLL"

If $\text{OGA sgn SGNDOG} - K_{\text{db}} \leq 0$, proceed to "NOROLL"

$$\text{DELOGA} = \text{OGA}$$

$$\text{DELOGART} = \text{OGARATE}$$

Proceed to "ONROLL"

ONROLL

$$\text{TS}_1 = K_{\text{msope}} \text{ ldCONACC DELOGART} - 1 \text{ sgn SGNDOG} \quad (\text{scaled B}_4)$$

$$\text{TS}_2 = K_{\text{msope}} \text{ ldCONACC DELOGART}^2 + K_{\text{msope}} (\text{DELOGA sgn SGNDOG} - K_{\text{db}}) + K_{\text{mcrat}} \quad (\text{scaled B}_0)$$

If $|\text{TS}_2| - 2^{-4} |\text{TS}_1| < 0$: (implemented by division of TS_2 shifted right 14 places by TS_1 and check of A: the 2^{-4} comes about from scale factors)

$$\text{RATDES} = \text{TS}_2 / \text{TS}_1$$

If $|\text{TS}_2| - 2^{-4} |\text{TS}_1| \geq 0$:

$$\text{RATDES} = +\text{MAX sgn} (\text{TS}_2 / \text{TS}_1) \quad (\text{magnitude} \cong 22\frac{1}{2}^\circ/\text{sec})$$

Proceed to "ROLLSET"

ROLLSET

TEMREG = RATDES (same cell)

TS = RATDES - OGARATE

If $|TS| \geq 2^{-4}$ rev/sec:

TS = $(2^{-4} - 2^{-18})$ sgn TS rev/sec

TS = $\frac{1}{2} K_{t6sc}$ ldCONACC TS, modulo 2^{11} centi-seconds

TEMREG = 2 TS

If $|TEMREG| \geq 2^{10}$ centi-seconds:

TEMREG = $(2^{10} - 2^{-4})$ sgn TEMREG centi-seconds

If TEMREG = 0:

Proceed to "NOROLL"

TS₁ = TEMREG ROLLFIRE

If TS₁ < 0, proceed to "NOROLL"

If $|TS_1| = 0$:

If $|TEMREG| - K_{tminfr} \leq 0$:

Proceed to "NOROLL"

If TS₁ > 0:

SGNDOG = 0 (used as flag in "JETROLL")

If $|K_{ldtmxfir} \cdot TEMREG| \geq 2^{-14}$:

TEMREG = K_{tmxfir} sgn TEMREG

ROLLFIRE = +MAX sgn TEMREG

TEMREG = $|TEMREG|$

If SGNDOG \neq 0, proceed to "JETROLL"

If (TIME6 - TEMREG) \leq 0, End of task (don't extend burn duration)

Proceed to "JETROLL"

NOROLL

ROLLFIRE = -0

TEMREG = -0

Proceed to "JETROLL"

JETROLL

T6LOC = "NOROLL1"

TIME6 = TEMREG

If SGNDOG = 0: (set in "ROLLSET" to continue present jets)

End of task

If ROLLFIRE = 0:

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

If ROLLFIRE > 0:

If bit 1 of ROLLWORD = 1:

Set bit 1 of ROLLWORD = 0

Channel 6 = K_{pro2}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

Set bit 1 of ROLLWORD = 1

Channel 6 = K_{pro1}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

If bit 2 of ROLLWORD = 1: (ROLLFIRE < 0)

Set bit 2 of ROLLWORD = 0

Channel 6 = K_{mro2}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

Set bit 2 of ROLLWORD = 1

Channel 6 = K_{mrol}

Set bit 15(TIME6 Count Enable) of channel 13 = 1

End of task

NOROLL1

ROLLFIRE = +0

Channel 6 = 0

Resume

Quantities in Computations

See also list of major variables and list of routines

ldCONACC: See Digital Autopilot Interface Routines.

AK₀, AK₁, AK₂: See Digital Autopilot Interface Routines.

C_{ekprime₀}: Single precision erasable memory constant, program notation "EKPRIME", scale factor B-5, units (revolutions/centi-second)/radian, giving the "steer law gain" for LM-off (used in "TVCINIT1" to initialize KPRIMEDT for "S40.8"). The maximum value that can be stored in the cell is $(100 \times 2\pi \times 2^{-5}) = (100\pi/16)$ radians/sec per radian, where "maximum value" actually is one least increment more than 37777₈.

C_{ekprime₁}: Single precision erasable memory constant, program notation "EKPRIME +1", scale factor B-5, units (revolutions/centi-second)/radian, giving the "steer law gain" set in "TVCINIT1" for LM-on (see C_{ekprime₀} and K_{fkprimdt}).

C_{ektlxdi₀}: Single precision erasable memory constant, program notation "EKTIX/I", used in "TVCINIT1" to initialize KTLXdI for "S40.15", for the LM-off configuration. Scale factor is B4, units ASCREV/sec², where one ASCREV = 1.07975111 revolutions (see K_{actsat}). To convert to memory information, value in units of (1/sec²) should be divided by (1.07975111×2^4) , giving a fraction which in turn is placed in the cell. Alternatively, cell could be considered scaled B18 in units of CDU actuator pulses per revolution times (newton-meters/kg-meters²), where second term compensates for the multiplication by IAVG/TIX (and reduces to 1/sec², of course). There are 2¹⁴ actuator pulses in 1.07975111 revolutions.

C_{ektlxdi₁}: Single precision erasable memory constant, program notation "EKTIX/I +1", used in "TVCINIT1" to initialize KTLXdI for "S40.15" for LM on (see also C_{ektlxdi₀}). Scale factor is B2, units ASCREV/sec² (see C_{ektlxdi₀}).

C_{ektlxdi₂}: Single precision erasable memory constant, program notation "EKTIX/I +2", used in "SWICHOVR" to initialize KTLXdI for "S40.15" for "low bandwidth" LM on. Scale factor is B2, units ASCREV/sec² (see C_{ektlxdi₀}).

$C_{erepfrac_0}$: Single precision erasable memory constant, program notation "EREPPFRAC", scale factor B2, used as LM-off initialization for REPFRAC in "TVCINIT1" and "S40.8" (note that TS_2 , i.e. REPFRAC, is doubled in "TEMPSET" if LM-off, hence effective "system" scale factor is B3).

$C_{erepfrac_1}$: Single precision erasable memory constant, program notation "EREPPFRAC +1", scale factor B2, used as LM-on initialization for REPFRAC in "TVCINIT1" and "S40.8". See also $K_{frepfrac}$.

C_{hbnl0_i} ($i = 0 - 14$): Set of single precision erasable memory constants used in "TVCINIT1" to initialize $Nl0_i$ for LM-on (if "SWICHOVR" entered, $Nl0_i$ loaded with K_{lbnl0_i} instead).

<u>i</u>	<u>Notation</u>	<u>Scale Factor</u>
0	"HBN10"	B0
1	"HBN11/2"	B1
2	"HBN12"	B0
3	"HBD11/2"	B1
4	"HBD12"	B0
5	"HBN20"	B0
6	"HBN21/2"	B1
7	"HBN22"	B0
8	"HBD21/2"	B1
9	"HBD22"	B0
10	"HBN30"	B0
11	"HBN31/2"	B1
12	"HBN32"	B0
13	"HBD31/2"	B1
14	"HBD32"	B0

CMDTMP: Value of required output information computed initially in "FWDFLTR" double precision, and subsequently rounded to single precision in "PITCHDAP" or "YAWDAP", scale factor B0, units ASCREV (or B14 in units of actuator pulses): VARK converts from units of revolutions in "FWDFLTR".

CNTR: Single precision counter, scale factor B14, units counts (of $\frac{1}{2}$ second each), preset in "TVCINIT1" and decremented in "LSHOTCHK". When reduced to 0, the next entrance to "TVCEXEC" causes a "one-shot" correction to PACTOFF and YACTOFF (as well as PDELOFF and YDELOFF), using $K_{fcorfrac}$. CNTR is set to +MAX in "S40.81" when the cutoff decision is made, to inhibit further trim updates after that time (a value of -1 is set in "TEMPSET"/"CNTRCOPY" to signify that the one-shot correction has been made). Since K_{tcorr_1} (initial condition for LM-on of CNTR) is zero, no decrementing of CNTR is required for that configuration.

COSCDUX, COSCDUZ: See Coordinate Transformations. Updated once a second while "CLOCKJOB" is entered, which includes period of time when TVC DAP is on.

DAP1, DAP2, DAP3: Filter quantities computed in "FWDFLTR", scale factor B-1, units revolutions. If LM off, then DAP3 is set equal to DAP2 (otherwise, it is derived), scaled B-3.

DAPDATR1: See Digital Autopilot Interface Routines.

DELBRTMP: Double precision erasable memory cell used to contained the updated value for DELPBAR or DELYBAR (for restart protection purposes), scale factor B14, units CDU actuator output pulses (or scale factor B0 in ASCREV units).

DELOGA: Single precision value of roll attitude error used in "ONROLL", scale factor B0, units revolutions (computed initially in "ROLLDAP" as "distance from switch parabola").

DELOGART: Single precision value of roll attitude rate error used in "ONROLL", scale factor B-4, units revolutions/second.

DELPBAR: Double precision value of estimated "c.g. offset relative to electrical null" in pitch, scale factor B14 in units of CDU actuator pulses (or scale factor B0 in ASCREV units). Initialized to PACTOFF in "TVCINIT1" and updated in "PCOPY" with DELBRTMP (for restart protection purposes). After CNTR reduced to zero, used in "TEMPSET" to updated PDELOFF and PACTOFF (using $K_{fcorfrac}$ and then REPFRAC for weighting). Used in "SPSOFF" to load PACTOFF if suitable initialization completed (i.e. "TVCINIT1" completed, initializing DELPBAR), and no thrust failure.

DELYBAR: Double precision value of estimated "c.g. offset relative to electrical null" in yaw, scale factor B14, units CDU actuator output pulses (or scale factor B0 in ASCREV units). See DELPBAR (substitute yaw axis for pitch axis).

ERRBTMP: Value of pitch or yaw error in body coordinates, scale factor B-1, units revolutions, limited to K_{errim} . Could also be considered to be the "integral of body-axis rate error". Value is displayed on FDAI error needles (due to loading of AK_1 or AK_2 by appropriate copy routine). Loaded in "IGNITION" with RCS DAP pitch and yaw negative (single precision) error information for use in "TVCINIT1", after limiting, to initialize PERRB and YERRB for LM-off.

$K_{ldactsat}$: Single precision constant, program notation "1/ACTSAT", scale factor B0, octal value 00101₈, corresponding to 0.003967285. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 253 least increments or more (see K_{actsat}).

$K_{ldattlim}$: Single precision constant, program notation "1/ATTLIM", scale factor B0, octal value 00170₈. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 137 least increments (at B-1 rev., about 1.505°): see K_{attlim} .

$K_{lderrim}$: Single precision constant, program notation "1/ERRLIM", scale factor B0, octal value 00004₈. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 2^{-2} or more (see K_{errim}): the 2^{-2} would correspond to 2^{-3} revolutions.

K_{ldrtim} : Single precision constant, program notation "1/RTLIM", scale factor B0, octal value 00115₈. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 213 least increments (at B-1 rev., about 2.34°).

$K_{ldtmxfir}$: Single precision constant, program notation "1/TMXFIR", scale factor B0, octal value 00004₈. Effect in equations such as to cause an answer of one least increment (2^{-14} in programmed equations) for an argument of 2^{-2} or more (see K_{tmxfir}): the 2^{-2} would correspond to 2.56 seconds.

K_{4csd2} : Single precision constant, program notation "BIT2", scale factor B14, units centi-seconds, giving the required time interval between DAP computations for different axes for LM-off. Value is 2×2^{-14} , corresponding to 2 centi-seconds: this gives a time interval of 4 centi-seconds from one pitch axis solution to the next.

K_{8csd2} : Single precision constant, program notation "BIT3", scale factor B14, units centi-seconds, giving the required time interval between DAP computations for different axes for LM-on. Value is 4×2^{-14} , corresponding to 4 centi-seconds: this gives a time interval of 8 centi-seconds from one pitch axis solution to the next.

K_{actsat} : Single precision constant, program notation "ACTSAT", scale factor B14, units CDU actuator output pulses (one pulse is 85.41 seconds). Value is 253×2^{-14} , corresponding to about 6.0024° . Note that 2^{14} CDU actuator output pulses correspond to about 1.079751111 revolutions; there are about 42.14963 pulses/degree.

K_{attlim} : Single precision constant, program notation "ATTLIM", scale factor B-1, units revolutions. Value is 136 least increments (see $K_{ldattlim}$), or octal 00210₈, corresponding to about 1.494° .

K_{csmn10} ($i = 0 - 14$): Set of single precision constants used in "TVCINIT1" i to initialize N10 _{i} for LM-off, program notation ($i = 0$) "CSMN10". Only values for $i = 0^i - 9$ are significant (since N10₁₀ to N10₁₄ are employed only for LM-on); values for $i = 10 - 14$ are the same cells as K_{1bn10_0} to K_{1bn10_4} .

<u>i</u>	<u>Nominal Value</u>	<u>Scale Factor</u>	<u>Nominal True Value</u>	<u>Stored</u>
0	0.99999	B0	0.99999 (N10)	0.99994
1	-0.2549	B1	-0.5098 (N11)	-0.50977
2	0.0588	B0	0.0588 (N12)	0.05878
3	-0.7620	B1	-1.524 (D11)	-1.52405
4	0.7450	B0	0.7450 (D12)	0.74500
5	0.99999	B0	0.99999 (N20)	0.99994
6	-0.4852	B1	-0.9704 (N21)	-0.97046
7	0.0	B0	0.0 (N22)	0.0
8	-0.2692	B1	-0.5384 (D21)	-0.53845
9	0.0	B0	0.0 (D22)	0.0

The "nominal value" is the scaled decimal input to the assembly program; the "stored" value was obtained by converting the octal memory information to decimal and applying the proper scale factor.

- K_{db} : Single precision constant, program notation "DB", scale factor B0, units revolutions, giving roll deadband for TVC DAP. Octal value is 00344₈, corresponding to about 5.0098°.
- K_{emat} : Single precision constant, program notation "E(-AT)", scale factor B0, value 37535₈, corresponding to $(1 - K_{onemeat})$.
- K_{errim} : Single precision constant, program notation "ERRLIM", scale factor B-1, units revolutions. Value is 10000₈, corresponding to 2⁻³ revolutions or 45°.
- $K_{fcorfrac}$: Single precision constant, program notation "FCORFRAC", scale factor B2. Used in "TEMPSET" as the LM-on c.g. one-shot correction gain (see CNTR above). A value of 2 $K_{fcorfrac}$ is used for LM-off (hence scale factor could be considered B3 for the LM-off application). Value is octal 10000₈, corresponding to 1.0 x 2⁻², where first term is equation value and second is scale factor.
- $K_{fkprimdt}$: Single precision constant, program notation "FKPRIMDT", scale factor B-4, units (revolutions/DAP cycle) per radian. Used in "SWICHOVR" to load KPRIMEDT. Value is 0.0102, corresponding approximately to 0.05 x 0.08 x (1/2 π) x 2⁴, where first term is value of gain in (radians/sec)/radian, second is DAP cycle rate in seconds, third is conversion to revolutions, and 4th is scale factor: see KPRIMEDT.
- $K_{frepfrac}$: Single precision constant, program notation "FREPFRAC", scale factor B2, used in "SWICHOVR" and "S40.8" to load REPFRAC for "low-bandwidth" mode of TVC DAP. Nominal value is 0.0375 x 2⁻², where first term is equation value and second is scale factor. The actual stored value corresponds to about 0.0376.
- $K_{intercp}$: Single precision constant, program notation "INTERCEP", scale factor B-3, units revolutions. Octal value is 00510₈, corresponding to about 0.2002 x 45 = 0.9009°.
- K_{lbn10} (i = 0 - 14): Set of single precision constants used in "SWICHOVR" to initialize N10_i for "low bandwidth" LM-on, program notation (i = 0) "LBN10".

<u>i</u>	<u>Nominal Value</u>	<u>Scale Factor</u>	<u>Nominal True Value</u>	<u>Stored</u>
0	0.99999	B0	0.99999 (N10)	0.99994
1	-0.3285	B1	-0.6570 (N11)	-0.65698
2	-0.3301	B0	-0.3301 (N12)	-0.33008
3	-0.9101	B1	-1.8202 (D11)	-1.82019
4	0.8460	B0	0.8460 (D12)	0.84601

<u>i</u>	<u>Nominal Value</u>	<u>Scale Factor</u>	<u>Nominal True Value</u>	<u>Stored</u>
5	0.03125	B0	0.03125 (N20)	0.03125
6	0.0	B1	0.0 (N21)	0.0
7	0.0	B0	0.0 (N22)	0.0
8	-0.9101	B1	-1.8202 (D21)	-1.82019
9	0.8460	B0	0.8460 (D22)	0.84601
10	0.5000	B0	0.5000 (N30)	0.50000
11	-0.47115	B1	-0.9423 (N31)	-0.94226
12	0.4749	B0	0.4749 (N32)	0.47491
13	-0.9558	B1	-1.9116 (D31)	-1.91162
14	0.9372	B0	0.9372 (D32)	0.93719

See note with K_{csmn10_i} .

K_{maxim} : Single precision constant, program notation "MAXLIM", scale factor B-4, units revolutions/second. Value is 0.01388889×2^4 , corresponding to about 5 degrees/second.

K_{mcrat} : Single precision constant, program notation "LMCRATE", scale factor B-4, units revolutions/second (gives limit cycle roll rate for TVC roll DAP). Value is 0.0111_8 , corresponding to about 0.10025 degree/second.

K_{minim} : Single precision constant, program notation "MINLIM", scale factor B-4, units revolutions/second. Value is 0.00277778×2^4 , corresponding to about 1 degree/second.

K_{mrol} : Single precision constant, program notation "-ROLL1", octal value 00012_8 , corresponding to channel 6 roll jets #10 and #12, giving a SM negative roll using quad B/D.

K_{mro2} : Single precision constant, program notation "-ROLL2", octal value 00240_8 , corresponding to channel 6 roll jets #14 and #16, giving a SM negative roll using quad A/C.

K_{mslope} : Single precision constant, program notation "-SLOPE", scale factor B0, units (revolutions/second)/revolution. Value is 0.2. Notation arises from fact that "true" slope is negative (an increasing attitude error results in a decreasing attitude rate).

$K_{onemeat}$: Single precision constant, program notation "1-E(-AT)", scale factor B0, octal value 00243_8 . Value corresponds to $163/16384$, or approximately 0.01, equivalent to the first term in the series expansion of $(1 - e^{-at})$ for $at = 0.01$.

K_{pro1} : Single precision constant, program notation "+ROLL1", octal value 00005₈, corresponding to channel 6 roll jets #11 and #9, giving a + Roll (for SM). Uses Quad B/D.

K_{pro2} : Single precision constant, program notation "+ROLL2", octal value 00120₈, corresponding to channel 6 roll jets #15 and #13, giving a + Roll (for SM). Uses Quad A/C.

K_{rrt} : Single precision constant, program notation "BIT5", scale factor B11, value 2^{-10} , corresponding to 2×2^{-11} , where first term is conversion of rate to units of revolutions/second, and second is scale factor.

K_{t6sc} : Single precision constant, program notation "T6SCALE", scale factor B7, value 31000₈. Value corresponds to 100×2^{-7} , where first term converts from seconds to centi-seconds and second is scale factor.

K_{tcorr} : Single precision constant, program notation "TCORR", scale factor B14, units counts (used as a preset for CNTR in "TVCINIT1" for LM-off). Value is 5×2^{-14} , meaning that the "single-shot" correction will take place the 6th time that "TVCEXEC" is entered, or about 3.42 seconds after "IGNITION" is done.

K_{tcorr} : Single precision constant, program notation "TCORR +1", scale factor B14, units counts (used as a preset for CNTR in "TVCINIT1" for LM-on). Value is 0, meaning that the "single-shot" correction will take place the first time that "TVCEXEC" is entered, or about 0.92 seconds after "IGNITION" is done.

K_{tminfr} : Single precision constant, program notation "TMINFIRE", scale factor B10, units centi-seconds. Value is 00030₈, corresponding to 1.5 centi-seconds (i.e. 15 milliseconds).

K_{tmxfir} : Single precision constant, program notation "TMAXFIRE", scale factor B10, units centi-seconds. Value is 250×2^{-10} , corresponding to 250 centi-seconds (i.e. 2.5 seconds). See $K_{ldtmxfir}$.

KPRIMEDT: Quantity used in "S40.8" to convert guidance attitude error information into OMEGAC. Scale factor is B-4, units (revolutions/DAP cycle) per radian. Loaded in "TVCINIT1" and "SWICHOVR". Least significant half expected to be 0 (from "TVCINIT1" loading method).

KTLXdI: Single precision value of appropriate $C_{ektlxdi}$ cell, loaded in "TVCINIT1" or "SWICHOVR" and used in "S40.15". Scale factor is B4 (LM off) or B2 (LM on), program notation "KTLX/I".

MCDUYDOT, MCDUZDOT: Complement of change in CDU_y and CDU_z respectively since previous sample, scale factor B-1, units revolutions/DAP cycle (in general). Computed in "PITCHDAP" and "YAWDAP" based on separate CDU angle samples, and used to compute appropriate component of OMEGAB. If the magnitude exceeds about 2.33° , it is set 0 (see K_{ldrtim}). The DAP cycle is 40 ms for LM-off and 80 ms for LM-on. Due to initialization in "DAPINIT", the rate derived the first pass through "YAWDAP" will be 50% too big.

NLO_i ($i = 0 - 14$): Set of filter constants loaded in "TVCINIT1" with K_{csm10_i} (LM-off) or C_{hbn10_i} (LM-on), or in "SWICHOVR" with K_{lbn10_i} . For LM-off, parameters for $i = 10 - 14$ are not used. Scale factors for $i = 1, 3, 6, 8$; and $11, 13$ are B1, with remainder of constants (all single precision) having scale factor B0.

OGA: Single precision value of roll error (OGANOW - OGAD), scale factor B0, units revolutions, computed in "TVCEXEC". Program notation where computed is "OGAERR".

OGAD: Single precision value of desired outer gimbal angle (i.e. CDU_x), scale factor B-1, units revolutions, set to CDU_x in "IGNITION" (when SPS engine is turned on). Used to cause the roll TVC DAP to maintain "roll attitude hold".

OGANOW: Single precision value of CDU_x , scale factor B-1, units revolutions, sampled in "TVCINIT4" and "TVCEXEC".

OGAPAST: Previous value of OGANOW, scale factor B-1, units revolutions, loaded with previous value of OGANOW in "TVCEXEC".

OGARATE: Single precision value of rate of change of OGANOW computed at the start of "ROLLDAP", scale factor B-4, units revolutions/second.

OMEGAB: Value of body rate determined from first difference of CDU_y and CDU_z information, scale factor B-1, units revolutions/DAP cycle (computed double precision). The y component is loaded in "PITCHDAP" and the z component in "YAWDAP", program notations "OMEGAYB" and "OMEGAZB" respectively. The "x" component is used for the storage of OGARATE (single precision), and is assigned that tag. The cells used for OMEGAB are used in the RCS DAP for ADOT (a common cell assignment is made for telemetry considerations). Note that "x" component is B-4 rev/second, not B-1 rev/DAP cycle, and single precision.

OMEGAC: Value of guidance steering output computed in "S40.8", scale factor B-1, units revolutions/DAP cycle (see KPRIMEDT). The x component is not used; the y component is used in "PITCHDAP"; and the z component is used in "YAWDAP".

FACTOFF: Single precision value of pitch c.g. offset, scale factor B14, units CDU actuator pulses (one pulse is 85.41 seconds). Could also be considered scaled BO in units of ASCREV (see C_{ektlxdi}). Can be loaded by R1 of N48 (e.g. in R03). It is measured "relative to electrical null" specified by K_{bias} (see Steering Computations). Quantity also called "total pitch trim angle." It is used in "TVCINIT1" to initialize filter quantities in TVC DAP, and is updated in "TEMPSET" after the one-shot has taken place (CNTR reduced to 0). Can also be updated in "SPSOFF" (see DELPBAR).

PCDUYPST, PCDUZPST: Previous values of CDU_y and CDU_z for use in deriving MCDUYDOT and MCDUZDOT for pitch DAP, single precision with scale factor B-1, units revolutions, in twos complement. Separate cells are used for yaw DAP (see YCDUYPST). PCDUiPST cells are initialized in "DAPINIT", with "PITCHDAP" entered at a time corresponding to one complete DAP cycle (pitch and yaw) later, giving proper first-difference computation.

PCMD: Single precision value of previous pitch output command information, scale factor B14, units CDU actuator pulses (or BO in units of ASCREV). It is intended to maintain an image of the information in the optics error counter driven from TVCPITCH output. It is set to FACTOFF in "TVCINIT1" ("S40.6" left engine bell at that position), and updated in "PCOPY" with CMDTMP.

PDELOFF: Value of pitch c.g. offset, scale factor B14, units CDU actuator pulses (or BO in units of ASCREV). The quantity is a double precision version of FACTOFF (updated in "TEMPSET" and initialized to FACTOFF in "TVCINIT1") used to minimize the effect of computer word length.

PERRB: Value of pitch error in body coordinates, loaded in "PCOPY" with ERRBTMP (which in turn is sum of previous PERRB and (OMEGAC_y - OMEGAB_y)), scale factor B-1, units revolutions. Could also be considered to be the "integral of body-axis pitch-rate error", since the OMEGA_i terms are in units of revolutions/DAP cycle.

PHASETMP: Single precision cell used in "SWICHOVR" to retain the value of TVCPHASE until the computation is complete, to permit setting of TVCPHASE = -2 as an indication to restart logic ("REDOTVC" and "CMDSOUT") that "SWICHOVR" was being performed.

PTMP_i (i = 1 - 6): Values of pitch filtered information, initialized to 0 in "TVCDAPON" and "SWICHOVR", and loaded with TMP_i in "PCOPY". Scale factors are all B-1, units revolutions.

RATDES: Single precision value of desired roll rate used in "ROLLSET", scale factor B-4, units revolutions/second. Program uses TEMREG cell.

RCSFLAGS: See Digital Autopilot Interface Routines.

REPFRAC: Single precision value of "repetitive c.g. correction fraction", scale factor B2, used in "LSHOTCHK" to control update of the c.g. tracker (if REPFRAC 0 or negative, no update is done) after CNTR becomes 0. It is initialized in "TVCINIT1" to appropriate $C_{erefrac}$ and (for "long" burns) is loaded in "S40.8" with the same information. If "SWICHOVR" entered, loaded with $K_{frepfrac}$. In "S40.8", is set to -0 if thrust below minimum value, to inhibit updates.

ROLLFIRE: Single precision cell, scale factor B0, set to +MAX for positive roll torque and -MAX for negative roll torque in roll TVC DAP. It is initialized to 0 in "TVCDAPON" and when roll jets are turned off in "NOROLL" or "NOROLL1", and used as a control cell to indicate that roll jet firings have been specified (used at start of "ROLLDAP", with TEMREG, to force at least $\frac{1}{2}$ second of no firing between successive firings).

ROLLWORD: Single precision cell initialized to 0 in "TVCDAPON" and used to permit alternation of the pairs of jets (i.e. quads) used for torque in the TVC roll DAP. Bit 1 is used for positive roll commands and bit 2 for negative roll commands: quad B/D is used if the bit is now 0 and quad A/C if bit is now 1. The bit used for selection is complemented after being used.

SGNDOG: Cell giving sign information for DELOGA (implemented in program by loading a "clear add" order for positive and a "clear subtract" order for negative), program notation "I". It is set 0 in "ROLLSET" if the polarity of the present jet firing is the same as that of desired firing during this evaluation, in order to flag the fact that present jets should be continued (subsequent evaluations can reduce the required roll firing time, but not increase it, due to logic at end of "ROLLSET").

SGNRT: Single precision cell, scale factor B14, giving information on the polarity of OGARATE.

SINCDUX: See Coordinate Transformations (and COSCDUX).

T5LOC: See Digital Autopilot Interface Routines.

T5TVCDT: Single precision value of required setting for TIME5, scale factor B14, units centi-seconds. TIME5 is incremented by hardware means once each centi-second, and when it reaches 16384 (2^{14}), program interrupt #2 is generated. Consequently, to cause program interrupt #2 in 2 centi-seconds from "now", TIME5 is set to $16384 - 2 = 16382$; in 4 centi-seconds requires 16380; etc.

T6LOC: See Digital Autopilot Interface Routines.

TEMREG: Single precision cell used for several functions within TVC roll DAP, but left at end with value of time loaded into TIME6, scale factor B10, units centi-seconds. It is preset to 0 in "TVCDAPON", and used with ROLLFIRE at the start of "ROLLDAP" to force at least $\frac{1}{2}$ second of no firing between successive firings.

TIME5: See Digital Autopilot Interface Routines.

TIME6: See Digital Autopilot Interface Routines.

TMPi (i = 1 - 6): Set of cells used to retain PTMPi or YTMPi information, scale factor B-1, units revolutions, in order to achieve restart protection and allow use of the common subroutines "FWDFLTR" and "PRECOMP". All cells double precision.

TTMP1: Double precision cell used for intermediate storage of quantities in TVC DAP (not shown in programmed equations), set 0 in "TVCDAPON" (no functional purpose served by the zeroing).

TVCDUMMY: Single precision cell set 0 in "TVCDAPON", but not otherwise referenced by TVC DAP (cell formerly assigned to a stroking-test variable, and employed for erasable memory layout considerations).

TVCEXPHS: Single precision cell, scale factor B14, used to control the restart logic associated with the TVC DAP for the waitlist task started every half second at "TVCEXEC". It is initialized to 0 in "IGNITION" and at the end of each performance of the "TVCEXEC" task. If it is non-zero when "REDOTVC" is entered, it is used as an indexing parameter to select the appropriate address within the "TVCEXEC" package in "EXRSTRT". For clarity, these addresses are not shown in the programmed equations, but correspond to the line following the one which incremented TVCEXPHS to its present value.

TVCPHASE: Single precision cell, scale factor B14, used to control the restart logic associated with the TVC DAP for the initialization, switchover to low bandwidth (LM-on), pitch copy cycle, and yaw copy cycle (values -1, -2, +1, and +3 respectively). Set to -1 in "IGNITION" and to 0 at the start of "TVCINIT4", to control branching in "REDOTVC". If "SWICHOVR" entered, set to -2, after the present value saved in PHASETMP. The appropriate "PCOPY"/"YCOPY" routine sets to control the pitch/yaw copy cycle restart logic.

TVCPITCH, TVCYAW: See Digital Autopilot Interface Routines.

VARK: Single precision variable gain for TVC pitch and yaw channels, scale factor B4 (LM off) or B2 (LM on), units ASCREV/revolution (or B18/B16 in units of CDU actuator pulses/revolution). It is computed in "S40.15" and used in "FWDFLTR".

VCNTR: Single precision cell, scale factor B14, used for control of the "TVCEXEC" computations. It is normally employed to cause an update of DAP parameters (due to mass change computed every 2 seconds in "S40.8") every 20 entrances to "TVCEXEC" (i.e. every 10 seconds). Since the modifications to the counter are not restart protected, however, a restart in a small interval could cause another modification to the counter (modifying the period, although this should not be a problem). The cell is preset to 19 in "TVCINIT1" and in "TVCEXEC" after updating parameters (when cell reaches +0).

YACTOFF: Single precision value of yaw c.g. offset, scale factor B14, units CDU actuator pulses (or scale B0 in ASCREV). It is measured "relative to electrical null" specified by $K_{y\text{bias}}$ (see Steering Computations). See PACTOFF above (updated at analogous times to those for PACTOFF, but of course with yaw parameters).

YCDUYPST, YCDUZPST: Previous values of CDU^y and CDU^z for use in deriving MCDUYDOT and MCDUZDOT for yaw DAP, single precision with scale factor B-1, units revolutions, in twos complement. Separate cells are used for pitch DAP (see PCDUYPST). YCDUⁱPST cells are initialized in "DAPINIT", but "YAWDAP" is entered for the first time at a time corresponding to 1.5 complete DAP cycles (i.e. pitch and yaw, plus pitch), giving a derived rate for this first pass (at DAP turn-on or after a restart) that is 50% too big.

YCMD: Single precision value of previous yaw output command information, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). See PCMD.

YDELOFF: Value of yaw c.g. offset, scale factor B14, units CDU actuator pulses (or B0 in units of ASCREV). Quantity is a double precision version of YACTOFF (cf. PDELOFF).

YERRB: Value of yaw error in body coordinates, loaded in "YCOPY" with ERRBTMP (which in turn is sum of previous YERRB and $(\text{OMEGAC} - \text{OMEGAB})_z$), scale factor B-1, units revolutions (cf. PERRB).^z

YTMP_i (i = 1 - 6): Values of yaw filtered information, initialized to 0 in "TVCDAPON" and "SWICHOVR", and loaded with TMP_i in "YCOPY". Scale factors are all B-1, units revolutions.

Entry Computations

CM/POSE (Entered by setting AVEGEXIT = "CM/POSE" in "P62")

$$mVREL = K_{mkvsc} \underline{V} + K_{kwe} (C_{unitw} * UNITR)$$

$$UXA = unit(mVREL)$$

$$TS = - unit(UXA * UNITR)$$

If $(mVREL)_{sp} - K_{spvq} \ll 0$:

$$TS = OLDUYA$$

$$UYA = TS$$

$$OLDUYA = UYA$$

$$UZA = - UYA * UXA$$

$TS = (AOGdPIP, AIGdPIP, AMGdPIP)$, converted to ones complement double precision

$$UBY = (-\cos TS_x \cos TS_y \sin TS_z + \sin TS_x \sin TS_y, \\ \cos TS_x \cos TS_z,$$

$$\cos TS_x \sin TS_y \sin TS_z + \sin TS_x \cos TS_y) \quad [REFSMAT]$$

$$UBX = (\cos TS_y \cos TS_z, \sin TS_z, -\sin TS_y \cos TS_z) \quad [REFSMAT]$$

$$UBZ = UBX * UBY$$

$$TS_1 = unit(UXA * UBY)$$

$$COSTH = TS_1 \cdot UZA$$

$$SINTH = TS_1 \cdot UYA$$

Perform "ARCTRIG"

$$TS_x = THETA_{sp} \quad (- \text{roll angle})$$

$$TS_y = (\sin^{-1}(UBZ \cdot UXA))_{sp} \quad (- \text{beta angle})$$

$$SINTH = UBX \cdot TS_1$$

$$COSTH = TS_1 \cdot UBZ$$

Perform "ARCTRIG"

$$TS_z = THETA_{sp} \quad (- \text{alfa angle})$$

$$TS_2 = GAMA$$

$$GAMA = -\left(\cos^{-1} (\text{UNITR} \cdot \text{UZA})\right)_{\text{sp}}$$

Inhibit interrupts

If bit 11(GAMDIFSW) of FLAGWRD6 = 1:

$$\text{GAMDOT} = K_{\text{tcdU}} (GAMA - TS_2)$$

If $|\text{GAMDOT}| - K_{\text{gmmn}} < 0$:

$$\text{GAMDOT} = 0$$

If bit 11(GAMDIFSW) of FLAGWRD6 = 0:

Set bit 11(GAMDIFSW) of FLAGWRD6 = 1

$$\text{GAMDOT} = 0$$

$$TS_3 = -TS_x, \text{ in range } \pm 180^\circ$$

$$\text{ROLLd180} = TS_3 - \text{ROLLdPIP} + \text{ROLLd180}, \text{ in range } \pm 180^\circ$$

$$TS_4 = -TS_z, \text{ in range } \pm 180^\circ$$

$$\text{ALFAd180} = TS_4 - \text{ALFAdPIP} + \text{ALFAd180}, \text{ in range } \pm 180^\circ$$

$$\text{BETAd180} = -TS_y - \text{BETAdPIP} + \text{BETAd180} \quad (\text{no overflow checks})$$

Release interrupts

$$\text{VMAGI} = |\underline{V}|$$

Proceed to address specified by POSEXIT

STARTENT (Entered by setting POSEXIT = "STARTENT" in "P63")

Set bits 9(RELVESW), 8(EGSW), 7(NOSWITCH), 6(HIND), 5(INRLSW),
and 3(O5GSW) of FLAGWRD6 = 0

Set bits 13(ENTRYDSP), 10(GONEPAST), and 4(LATSW) of FLAGWRD6 = 1

$$\text{LOD} = C_{\text{odpad}}$$

$$\text{LAD} = C_{\text{adpad}}$$

$$\text{LdDCMINR} = K_{\text{cs15p2}} \text{LAD}$$

$$\text{KLAT} = K_{\text{atspe}} \text{LAD}$$

$$Q7 = K_{\text{q7f}}$$

$$\text{FACTOR} = (1 - 2^{-28})$$

$$\text{LdD} = -\text{LAD} \text{sgn HEADSUP}$$

Perform "STARTEN1"

$$\text{LATANG} = (\text{unit}(\underline{V} * \text{UNITR})) \cdot \underline{R}_t$$

$$\text{K2ROLL} = -1 \text{ sgn LATANG}_{\text{sp}} \quad (+0 \text{ is } +, -0 \text{ is } -)$$

$$Q2 = K_{q21} \text{ LAD} + K_{q22}$$

GOTOADDR = "INITROLL"

POSEXIT = "SCALEPOP"

Proceed to "SERVEXIT"

STARTEN1 Entered from "STARTENT" and "NEWRNVN"

GOTOADDR = Return address

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

LAT = LATSPL

LONG = LINGSPL

ALT = DELVLVC_x

ALT = 0

TIMEdRTO = T_{pptm}

TS = TIMEdRTO

Perform "LALOTORV"

$\underline{R}_{t,i} = \text{unitALPHA}\underline{V}$

DTEAROT = K_{500sec}

Perform "EARROT1"

$$\text{THETAH} = \cos^{-1} (\underline{R}_t \cdot \text{UNITR})$$

Proceed to address specified by GOTOADDR

SCALEPOP Entered due to setting of POSEXIT in "STARTENT"

Perform "TARGETNG"

Proceed to address specified by GOTOADDR

TARGETNG

If bit 9(RELVELSW) of FLAGWRD6 = 0:

$$\underline{VEL} = K_{kvsc} \underline{V}$$

If bit 9(RELVELSW) of FLAGWRD6 = 1:

$$VEL = -mVREL$$

$$UNITV = \text{unit}VEL$$

$$VSQUARE = |VEL|^2$$

$$LEQ = VSQUARE - 1$$

$$NV = |VEL|$$

$$RDOT = VEL \cdot UNITR$$

$$D = K_{kasc} |DELV|, \text{ limited } \geq K_{mnd} \quad (\text{limit effective only if } D_{tp} = 0).$$

$$UNI = \text{unit}(VEL * UNITR)$$

If bit 9(RELVELSW) of FLAGWRD6 = 1:

$$TS = 0$$

If bit 9(RELVELSW) of FLAGWRD6 = 0:

$$D = D + D (RDOT / K_{mhsca} + K_{mksc} D / NV)$$

If bit 8(EGSW) of FLAGWRD6 = 0:

$$TS = K_{kteta} \text{ THETAH}$$

If bit 8(EGSW) of FLAGWRD6 = 1:

$$\text{If } NV - K_{vmin} < 0:$$

Set bit 9(RELVELSW) of FLAGWRD6 = 1

$$TS = K_{ktl} \text{ THETAH} / NV$$

$$DTEAROT = TS + T_{pptm} - \text{TIME}dRTO$$

Perform "EARROT2"

$$LATANG = R_t \cdot UNI$$

Set bit 8(GONEBYTG) of FLAGWRD7 = 0

If $(R_t * UNITR) \cdot UNI < 0$, set bit 8(GONEBYTG) of FLAGWRD7 = 1

$$TS = R_t \cdot UNITR$$

If $TS - K_{nq} < 0$:

$$\text{THETAH} = \cos^{-1} TS$$

If $TS - K_{nq} \geq 0$:

$$THETAH = K_{kacs} \sqrt{|2(TS - 1)|}$$

$TS = THETAH$

If bit 8 (GONEBYTG) of FLAGWRD7 = 0:

$$TS = -TS$$

$RTGON67 = TS$

If $D - K_{pt05g} < 0$:

Set bit 3(O5GSW) of FLAGWRD6 = 0

Return

Set bit 3(O5GSW) of FLAGWRD6 = 1

$$TS = \frac{|\underline{DELVREF}|^2 - (\underline{DELVREF} \cdot \underline{UXA})^2}{(\underline{DELVREF} \cdot \underline{UXA})^2}$$

If no overflow has taken place (e.g. $|TS| < 1$):

$$LdDCALC = \sqrt{TS}$$

Return

INITROLL

If bit 5(INRLSW) of FLAGWRD6 = 1:

If $RDOT + K_{vrcont} \geq 0$:

$$DIFFOLD = 0$$

$$DLEWD = K_{dlewd0}$$

$$LEWD = K_{ewdl}$$

GOTOADDR = "HUNTEST"

Proceed to "HUNTEST"

If $(KAT - D) < 0$, proceed to "CONSTD"

Proceed to "LIMITL/D"

If bit 3(O5GSW) of FLAGWRD6 = 0, proceed to "LIMITL/D"

$TS = 64$ and perform "NEWMODEX"

$$ENTRYVN = 0674_{vn}$$

$$KAT = K_{ka2} + LEQ^3 / K_{ldkal}, \text{ limited } \leq K_{kalim}$$

Set bit 10(GONEPAST) of FLAGWRD6 = 0 ("STARTENT" initialized
this bit to 1)

If $NV - K_{vfnl} < 0$:

GOTOADDR = "KEP2"

Set bit 5(INRLSW) of FLAGWRD6 = 1 (must be 0 to get here)

Proceed to "LIMITL/D"

$DO = K_{ka3} LEQ + K_{ka4}$ (NV at least 27,000 fps to get here)

$CdDO = K_{cone} / DO$

Reset overflow indicator

LdD = LAD

$TS = (RDOT / NV)^3 / K_{ldk44} - K_{vfin} + NV$

If no overflow has taken place since indicator reset:

If $TS \geq 0$:

LdD = - LAD

Set bit 5(INRLSW) of FLAGWRD6 = 1 (must be 0 to get here)

Proceed to "LIMITL/D"

HUNTEST

A1 = D

TS = LAD

If $RDOT \geq 0$:

TS = LEWD

$V1 = NV + RDOT / TS$

$AO = (V1^2 / VSQUARE) (D + RDOT^2 / (K_{2chs} TS))$

If $RDOT < 0$:

A1 = AO

If $LdD < 0$:

$V1 = V1 - K_{vquit}$

Proceed to "HUNTEST1"

HUNTEST1

$$ALP = K_{2chs} AO / (LEWD V1^2)$$

$$FACT1 = V1 / (1 - ALP)$$

$$FACT2 = ALP (ALP - 1) / AO$$

$$VL = FACT1 (1 - \sqrt{FACT2 Q7 + ALP})$$

$$GAMMAL1 = LEWD (V1 - VL) / VL$$

If $VL - K_{vmn} < 0$:

Proceed to "PREFINAL"

$$VBARS = VL^2$$

$$TS = K_{hav} - VL$$

If $TS < 0$:

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

$$DVL = TS$$

$$VS1 = K_{hav}$$

$$TS = VS1 - V1$$

If $TS \geq 0$:

$$DVL = DVL - TS \quad (\text{i.e. } V1 - VL)$$

$$VS1 = V1$$

$$DHOOK = \frac{(1 - VS1 / FACT1)^2 - ALP}{FACT2}$$

$$AHOOKDV = \frac{1}{4} (DHOOK / Q7) - K_{chk}$$

$$TS = GAMMAL1 - (AHOOKDV + 1) K_{chone} DVL^2 / (DHOOK VBARS)$$

If $TS \geq 0$:

$$GAMMAL = TS$$

If TS < 0:

$$VL = VL + \frac{(1/3) VL TS}{(1/3) LEWD - (AHOOKDV + 2/3)(K_{chone} DVL)/(DHOOK VL)}$$

$$Q7 = \frac{(1 - VL / FACT1)^2 - ALP}{FACT2}$$

$$VBARS = VL^2$$

$$GAMMAL = 0$$

$$GAMMAL1 = GAMMAL1 + K_{q19} (GAMMAL - GAMMAL1)$$

Proceed to "RANGER"

RANGER

$$COSG = 1 - \frac{1}{2} GAMMAL^2$$

$$CPE = \sqrt{1 + (VBARS - 2) COSG^2 VBARS}$$

$$ASKEP = 2 \sin^{-1} (VBARS COSG GAMMAL / CPE)$$

$$ASP1 = Q2 + K_{q3} VL$$

$$ASPUP = (K_{c12} / GAMMAL1) \left(-\log_e \left((VL^2 Q7) / (AO VBARS) \right) \right)$$

$$ASPDWN = K_{kc3} RDOT NV / (AO LAD)$$

$$ASP3 = K_{q5} (K_{q6} - GAMMAL)$$

$$ASPSpTMp = \left((ASKEP_{sp}, ASP1_{sp}), (ASPUP_{sp}, ASPDWN_{sp}), ASP3 \right)$$

$$DIFF = ASP3 + ASPDWN + ASPUP + ASP1 + ASKEP - THETAH$$

If $|DIFF| - K_{25nm} < 0$:

TS = 65 and perform "NEWMODEX"

Establish "P65.1" (priority 13g)

GOTOADDR = "UPCONTRL"

Proceed to address specified by GOTOADDR

If bit 6(HIND) of FLAGWRD6 = 0:

If $DIFF \geq 0$: (note that polarity of DIFF the opposite of that in official equation documentation)

$$DIFFOLD = DIFF$$

$$Q7 = K_{q7f}$$

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

DLEWD = (DLEWD DIFF) / (DIFFOLD - DIFF)

If DLEWD + LEWD < 0:

DLEWD = - $\frac{1}{2}$ LEWD

Proceed to second previous line (recheck of sum with LEWD)

If overflow has taken place since calculation of DIFF:

LEWD = (1 - 2⁻²⁸)

DIFFOLD = DIFF

Q7 = K_{q7f}

GOTOADDR = "HUNTEST"

Proceed to "CONSTD"

LEWD = DLEWD + LEWD

Change priority of present job to 16_g (lower than "SERVICER")

GOTOADDR = "ENDEXIT"

DIFFOLD = DIFF

Set bit 6(HIND) of FLAGWRD6 = 1

Q7 = K_{q7f}

Proceed to "HUNTEST"

P65.1 Established when P65 entered from "RANGER"

Set bit 13(ENTRYDSP) of FLAGWRD6 = 0

TS = 1669_{vn}

Perform "GOFLASHR": if terminate, proceed to previous line
if proceed, skip next line
otherwise, proceed to previous line

End of job

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

End of job

UPCONTRL

If $D - K_{c21} \geq 0$:

Set bit 7(NOSWITCH) of FLAGWRD6 = 1 (reset in "L355")

If $NV - V1 \geq 0$:

Reset overflow indicator

$$TS = LAD + K_{k2d} \left(RDOT + LAD (NV - V1) \right) \quad (\text{scaled B8})$$

$$TS_1 = AO \text{ VSQUARE} / V1^2 - (V1 - NV)^2 LAD / K_{2chs} \quad (\text{scale B0})$$

$$TS_2 = K_{k1d} (D - TS_1) + TS \quad (\text{scaled B8})$$

Rescale TS_2 to scale factor B0

If overflow has taken place since indicator reset:

$$LdD = LAD \text{ sgn } TS_2$$

Proceed to "GLIMITER"

$$LdD = TS_2$$

Proceed to "NEGTESTS"

If $D - Q7 < 0$:

$TS = 66$ and perform "NEWMODEX"

$$ENTRYVN = 0622_{vn}$$

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

GOTOADDR = "KEP2"

Proceed to "KEP2"

If $RDOT < 0$:

If $NV - VL - K_{c18} < 0$:

Proceed to "PREFINAL"

If $D - AO \geq 0$:

$$LdD = LAD$$

Proceed to "LIMITL/D"

$$VREF = FACT1 (1 - \sqrt{D FACT2 + ALP})$$

$$RDOTREF = LEWD (V1 - VREF)$$

$$TS = VS1 - VREF$$

If $TS \geq 0$:

$$RDOTREF = RDOTREF - \frac{K_{chone} TS^2 (1 + TS AHOOKDV / DVL)}{DHOOK VREF}$$

$$TS = D - K_{q7min} \quad (\text{Tag here "CONTINU2"})$$

Reset overflow indicator

If $TS \geq 0$: (should not be, since $K_{q7min} = +MAX$)

$$FACTOR = (D - Q7) / (A1 - Q7)$$

$$TS = FACTOR (NV - VREF + FACTOR (RDOT - RDOTREF) / K_{ldkb}) / K_{mldkb2} \quad (\text{scale B4})$$

If overflow has taken place since indicator reset:

$$LdD = LAD \text{ sgn } TS$$

Proceed to "GLIMITER"

If $|TS| - K_{ptb4} \geq 0$:

$$TS = (K_{ptb4} + K_{ptone} (|TS| - K_{ptb4})) \text{ sgn } TS$$

Rescale TS to scale factor B0

$$TS = TS + LEWD$$

If overflow has taken place since indicator reset:

$$LdD = LAD \text{ sgn } TS$$

Proceed to "GLIMITER"

$$LdD = TS$$

Proceed to "NEGTESTS"

NEGTESTS

If $D - K_{c20} < 0$:

Proceed to "LIMITL/D"

Set bit 4(LATSW) of FLAGWRD6 = 0

If LdD < 0:

LdD = 0

Proceed to "LIMITL/D"

CONSTD

Reset overflow indicator

$TS = CdDO \text{ LEQ} + K_{k2d} (RDOT + K_{2hs} DO / NV)$ (scale B8)

$TS_2 = K_{kld} (D - DO) + TS$ (scaled B8)

Rescale TS_2 to scale factor B0

If overflow has taken place since indicator reset:

LdD = LAD sgn TS_2

Proceed to "GLIMITER"

LdD = TS_2

Proceed to "NEGTESTS"

KEP2

If $K_{q7fkdm} - D < 0$:

Proceed to "PREFINAL"

$TS_{tp} = (ROLLC_{dp}, \text{ROLLHOLD})$

If bit 3(O5GSW) of FLAGWRD6 = 0:

$TS_{tp} = 0$

$(ROLLC_{dp}, \text{ROLLHOLD}) = TS_{tp}$

Proceed to "P62.3" (ENTRYVN set to 0622 by "UPCONTRL" before "KEP2" entered, if vn in P66)

PREFINAL

GOTOADDR = "PREFINAL" (for restart considerations)

TS = 67 and perform "NEWMODEX"

ENTRYVN = 0666_{vn}

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

Set bit 8(EGSW) of FLAGWRD6 = 1

GOTOADDR = "PREDICT3"

Proceed to "PREDICT3"

PREDICT3

If $NV - K_{vquit} < 0$:

Establish "P67.1" (priority 16_g)

GOTOADDR = "P67.2"

Proceed to "P67.2"

JJ = 12

If $K_{vrfr_{JJ}} - NV_{sp} > 0$:

JJ = JJ - 1 and repeat check

$$GRAD = \frac{|K_{vrfr_{JJ}} - NV_{sp}|}{K_{vrfr_{JJ+1}} - K_{vrfr_{JJ}}}$$

i = JJ

JJ = JJ + 13

$$F1V = K_{drda_i} + GRAD (K_{drda_{i+1}} - K_{drda_i})$$

JJ = JJ + 13

$$F2V = K_{drdd_i} + GRAD (K_{drdd_{i+1}} - K_{drdd_i})$$

JJ = JJ + 13

$$RDRFV = K_{rdtr_i} + GRAD (K_{rdtr_{i+1}} - K_{rdtr_i})$$

JJ = JJ + 13

$$RTOGOV = K_{rtgo_i} + GRAD (K_{rtgo_{i+1}} - K_{rtgo_i})$$

JJ = JJ + 13

$$DREFRV = K_{aref_i} + GRAD (K_{aref_{i+1}} - K_{aref_i})$$

JJ = JJ + 13

$$F3V = K_{\text{drd}_i} + \text{GRAD} (K_{\text{drd}_{i+1}} - K_{\text{drd}_i})$$

Set TS = DREFRV and DREFRV = 0

$$\text{PREDANG} = \text{RTOGOV} + F2V (\text{RDRFV} - \text{RDOT}) + \text{FLV} (D_{\text{sp}} + \text{TS})$$

$$\text{TS} = \text{PREDANG}_{\text{dp}} - \text{THETAH}$$

If bit 10(GONEPAST) of FLAGWRD6 = 1:

$$\text{LdD} = - \text{LAD}$$

Proceed to "GLIMITER"

If bit 8(GONEBYTG) of FLAGWRD7 = 1:

Set bit 10(GONEPAST) of FLAGWRD6 = 1

$$\text{DNRNGERR} = K_{\text{maxrng}}$$

$$\text{LdD} = - \text{LAD}$$

Proceed to "GLIMITER"

$$\text{DNRNGERR} = \text{TS}$$

Reset overflow indicator

$$\text{TS} = - 4 \text{ DNRNGERR} / F3V$$

If overflow has taken place since indicator reset ($|\text{TS}| \geq 1$):

$$\text{LdD} = \text{LAD} \text{ sgn } \text{TS}$$

Proceed to "GLIMITER"

$$\text{TS} = \text{TS} + \text{LOD}$$

If $|\text{TS}| \geq 1$:

$$\text{LdD} = \text{LAD} \text{ sgn } \text{TS}$$

Proceed to "GLIMITER"

$$\text{LdD} = \text{TS}$$

Proceed to "GLIMITER"

GLIMITER

$$TS = K_{gmx2} - D$$

If $TS \geq 0$:

Proceed to "LIMITL/D"

$$TS = TS + K_{gmx2}$$

If $TS < 0$:

$$LdD = LAD$$

Proceed to "LIMITL/D"

$$CPX = \sqrt{K_{2hs} TS (K_{ldgmx} LEQ + LAD) + K_{2hsgmxq} / VSQUARE}$$

If $CPX + RDOT \geq 0$:

Proceed to "LIMITL/D"

$$LdD = LAD$$

Proceed to "LIMITL/D"

LIMITL/D

$$LdD1 = LdD$$

If bit 10(GONEPAST) of FLAGWRD6 = 1: (also set in "STARTENT" and
reset in "INITROLL" after
0.05G sensed)

Proceed to "L355"

$$CPY = KLAT VSQUARE + K_{atbias}$$

If $|LdD| - LdDCMINR \geq 0$:

If $LATANG \operatorname{sgn} K2ROLL < 0$:

$$LdD1 = LdDCMINR \operatorname{sgn} LdD$$

Proceed to "L355"

$$CPY = \frac{1}{2} CPY$$

If $LATANG \operatorname{sgn} K2ROLL - CPY < 0$:

Proceed to "L355"

If bit 7(NOSWITCH) of FLAGWRD6 = 1: (set by "UPCONTRL")

Set bit 7(NOSWITCH) of FLAGWRD6 = 0

Proceed to "L355"

K2ROLL = - K2ROLL

Proceed to "L355"

L355

TS = Ldd1 / LAD, with magnitude limited ≤ 1 (due to interpreter divide order)

Set bit 7(NOSWITCH) of FLAGWRD6 = 0

ROLLC = $(\cos^{-1} TS)$ sgn K2ROLL

Proceed to "ENDEXIT"

ENDEXIT

If bit 13(ENTRYDSP) of FLAGWRD6 = 1:

TS = ENTRYVN

Perform "REGODSPR"

Proceed to "SERVEXIT" (after special check for new job)

P67.1 Established by "PREDICT3"

TS = 1667_{vn}

Proceed to "GOFLASH": if terminate, proceed
if proceed, proceed
otherwise, proceed to previous line

Set bits 2(CMDSTBY) and 1(GYMDIF) of FLAGWRD6 = 0

AVEGEXIT = "SERVEXIT"

Proceed to "GOTOPOOH"

P67.2

Set bit 13(ERADCOMP) of FLAGWRD1 = 0

ALPHA_V = R

TS = T_{pptm}

Perform "LAT-LONG"

Proceed to "SERVEXIT"

Quantities in Computations

See also list of major variables and list of routines

- AI: Value of drag (set to D or AO in "HUNTEST") used to compute FACTOR, scale factor BO, G-units (25 g's).
- AHOOKDV: Term in GAMMAL calculation (equal to AHOOK DVL), scale factor B4.
- AIGdPIP: See General Program Control.
- ALFAd180: Single precision angle, program notation "ALFA/180", giving the third Euler angle of the CM attitude (about UBY in "pitch"), scale factor B-1, units revolutions.
- ALFAdPIP: See General Program Control.
- ALP: Quantity used to compute VREF in "UPCONTRL" (and computed itself in "HUNTEST1", where it is also used), scale factor BO.
- ALPHA \bar{V} : See Coordinate Transformations.
- ALT: See Display Computations: discussion concerning loading with DELVLVC_x also applicable to "STARTEN1".
- AMGdPIP: See General Program Control.
- AO: Initial drag for "UPCONTRL", scale factor BO, G-units (25 g's).
- AOGdPIP: See General Program Control.
- ASKEP: Value of "Kepler range", scale factor BO, units revolutions, stored in push-down list location OD. A single precision value is available in the ASKEP cell itself for generating telemetry data.
- ASPl: Value of "final phase" range, scale factor BO, units revolutions, stored in push-down list location 2D. A single precision value is available in the ASPl cell itself for generating telemetry data.
- ASP3: Value of "gamma correction", scale factor BO, units revolutions.
- ASPDWN: Value of "range down to pull-up", scale factor BO, units revolutions, stored in push-down list location 6D. A single precision value is available in the ASPDWN cell itself for generating telemetry data.
- ASPSpT \bar{M} p: "Vector" information loaded for telemetry purposes (see "RANGER" for format). Cells loaded are the same as those used for WBODY (RCS DAP) and OMEGAC (TVC DAP); program notation "ASPS(TM)".

ASPUP: Value of "up-range", scale factor B0, units revolutions, stored in push-down list location 4D. A single precision value is available in the ASPUP cell itself for generating telemetry data.

AVEGEXIT: See General Program Control.

BETAd180: Single precision angle, program notation "BETA/180", giving the second Euler angle of the CM attitude (about $UXA * UBY$ in "yaw"), scale factor B-1, units revolutions.

BETAdPIP: See General Program Control.

C_{adpad}: Single precision erasable memory constant, program notation "LADPAD", scale factor B0, used to initialize LAD in "STARTENT".

C_{odpad}: Single precision erasable memory constant, program notation "LODPAD", scale factor B0, used to initialize LOD in "STARTENT".

C_{unitw}: See General Program Control.

CdDO: Value of K_{cone} / DO , computed in "INITROLL" and used in "CONSTD", scale factor B6.

COSG: Approximation to value of cos GAMMAL, scale factor B1.

COSTH: See Coordinate Transformations.

CPE: Value of eccentricity, scale factor B2 (stored in push-down list).

CPX: Value of intermediate quantity used in "GLIMITER", scale factor B1, V-units (see NV).

CPY: Value of lateral miss limit, scale factor B2, units of radians.

D: Value of total acceleration, scale factor B0, G-units (25 g's).

DELVLVC: See Burn Control (and ALT).

DELVREF: See General Program Control.

DHOOK: Quantity used in computation of GAMMAL, scale factor B0.

DIFF: Difference between predicted and actual expected range, scale factor B0, units revolutions. Note that polarity of program definition is the reverse of that in official equation documentation.

DIFFOLD: Previous value of DIFF, scale factor B0, units revolutions.

DLEWD: Computed desired change in LEWD, scale factor B0.

DNRNGERR: Difference between predicted angle of travel and actual angle to target (i.e. PREDANG - THETAH) computed in "PREDICT3", scale factor BO, units revolutions, provided that bit 10(GONEPAST) of FLAGWRD6 = 0. If bit is set 1, DNRNGERR is set to K_{maxrng} (and not subsequently loaded) in P67.

DO: Value of "controlled constant D", scale factor BO, G-units (25 g's).

DREFRV: Value of K_{aref} table for present NV, same scaling as K_{aref} .

DTEAROT: See Coordinate Transformations.

DVL: Value of (VS1 - VL), scale factor B1, V-units (see NV).

ENTRYVN: Single precision cell loaded with verb-noun pattern (in proper format) for use in "ENDEXIT" for entry display.

F1V: Value of K_{drda} table for present NV, same scaling as K_{drda} .

F2V: Value of K_{drdd} table for present NV, same scaling as K_{drdd} .

F3V: Value of K_{drd} table for present NV, same scaling as K_{drd} .

FACT1: Value of quantity computed in "HUNTEST1", scale factor B1, V-units (see NV).

FACT2: Value of quantity computed in "HUNTEST1", scale factor BO.

FACTOR: Quantity used in "UPCONTRL" (initialized to 1 in "STARTENT"), scale factor BO.

GAMA: Single precision angle computed in "CM/POSE", scale factor BO, units revolutions, used to compute (by forming first difference with previous value) GAMDOT.

GAMDOT: Rate of change of GAMA per 0.1 second, scale factor B-1, units revolutions. Set to zero if magnitude is less than K_{gmmn} .

GAMMAL: Value of "flight path angle at VL", scale factor BO, units radians.

GAMMAL1: Value of "simple form of GAMMAL", scale factor BO, units radians. Stored in push-down list location 22D.

GOTOADDR: Single precision cell used to contain the starting address of the computations performed after "SCALEPOP" is performed, and used to control the phases of the entry computations. Prior to start of entry computations, used as storage for return address information.

GRAD: Single precision "gradient" of present velocity with respect to a table value, scale factor BO, used to permit linear interpolation in "PREDICT3".

HEADSUP: See Entry Preparation.

JJ: Single precision indexing parameter used in "PREDICT3", scale factor B14. It is incremented by 13 to select the next set of tabular information within an index loop (the incrementing is shown because the quantity is on the downlink).

K_{ldgmx} : Constant, program notation "1/GMAX", scale factor B-2, units of g^{-1} . Value is 0.5, corresponding to $(1/8) \times 2^2$, where first term is reciprocal of "GMAX" and second is scale factor.

K_{ldk44} : Constant, program notation "1/K44", scale factor B-1, units of (1/V-units). Stored value is 0.00260929464, corresponding to $(1/19749550) \times 25766.1973 \times 2^1$, where first term is reciprocal of "K44" (in units of fps), second is V-unit conversion, and third is scale factor.

K_{ldkal} : Constant, program notation "1/KAL", scale factor B6, units of G-units (25 g's) reciprocal. Value is 0.30048077, corresponding to $(1/1.3) \times 25 \times 2^{-6}$, where first term is reciprocal of "KAL" (in units of g's), second is G-unit conversion, and third is the scale factor.

K_{ldkb} : Constant, program notation "1/KB1", scale factor B0, value 0.400. Value corresponds to $(1/2.5)$, or the reciprocal of the value of "KB1".

K_{2chs} : Constant, program notation "2CLHS"; scale factor B2, value 0.0215983264. Value corresponds to $2 \times (1.25) \times 28500 \times 805 \times (25766.1973)^{-2} \times 2^{-2}$, where first term is equation factor of 2, second is value of C1, third is value of HS, fourth converts answer from fps^2 to G-units (of 25 g's @ $32.2 \text{ fps}^2 = 805 \text{ fps}^2$), fifth converts velocity to fps from V-units, and sixth is scale factor.

K_{2hs} : Constant, program notation "2HS", scale factor B2, value 0.0172786611. Value corresponds to $2 \times 28500 \times 805 \times (25766.1973)^{-2} \times 2^{-2}$, where terms (except for the omission of the 1.25 factor for C1) have analogous significance to those for K_{2chs} .

$K_{2hsgmxq}$: Constant, program notation "2HSGMXSQ", scale factor B4, value $3.05717E-5$. Value corresponds to the square of: $(2 \times 28500 \times 8 \times 32.2 \times (25766.1973)^{-2} \times 2^{-2})$, where first term is equation factor of 2, second is value of HS, third is value of GMAX (in g's), fourth converts GMAX to fps^2 , fifth converts V-units to fps, and sixth is square root of scale factor.

K_{25nm} : Constant, program notation "25NM", scale factor B0, units revolutions. Value is 0.0011574074, corresponding to $25 \times (1/21600)$, where first term is "basic" value and second converts from nmi to revolutions (it is the reciprocal of $3437.7468 \times 2\pi$, or 60×360 for 60 nmi/degree).

K_{500sec} : Constant, program notation "500SEC", scale factor B28, units centi-seconds. Value is 50000×2^{-28} , corresponding to 500 seconds.

- K_{aref} : Single precision set of constants, scale factor B0, G-units (of 25 g's). To convert to fps^2 , multiply table entries by $32.2 \times 25 = 805$. Table starts at "VREFER +65". See below.
- K_{atbias} : Constant, program notation "LATBIAS", scale factor B2, value 0.00003, units radians. Corresponds to 0.00012 radians, or (multiplying by 3437.7468) 0.412529616 nmi.
- K_{atspe} : Constant, program notation "LATSLOPE", scale factor B-1, value 0.083333. Value corresponds to $(1/12) \times 2^{-1}$, or $(1/24)$.
- K_{c12} : Constant, program notation "C12", scale factor B-5, value 0.00684572901. Value corresponds to $28500 \times (1/21\ 202\ 900) \times (1/2\ \pi) \times 2^5$, where first term is HS, second is reciprocal of RE in feet (standard "entry radius"), third converts from radians to revolutions, and fourth is scale factor.
- K_{c18} : Constant, program notation "C18", scale factor B1, V-units (see NV). Value is 0.0097026346, corresponding to $500 \times (1/25766.1973) \times 2^{-1}$, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.
- K_{c20} : Constant, program notation "C20", scale factor B0, G-units (of 25 g's). Value is 0.26086957, corresponding to $210 \times (1/805)$, where first term is "basic" value (in fps^2), and second is conversion to G-units.
- K_{c21} : Constant, program notation "C21", scale factor B0, G-units (of 25 g's). Value is 0.17391304, corresponding to $140 \times (1/805)$, where first term is "basic" value (in fps^2), and second is conversion to G-units.
- K_{chk} : Constant, program notation "CHOOK", scale factor B4, value 2^{-6} , corresponding to 0.25×2^{-4} , where first term is "basic" value and second is scale factor.
- K_{chone} : Constant, program notation "CH1", scale factor B-4, value 0.32×2^{-1} . Value corresponds to $1.0 \times (1/25) \times 2^4$, where first term is the "basic" value, second converts to G-units, and third is scale factor.
- K_{ccone} : Constant, program notation "COO1", scale factor B6, value -0.000625. Value corresponds to $(-1) \times (1/25) \times 2^{-6}$, where first term is "basic" value, second converts to G-units, and third is scale factor.
- K_{csl5p2} : Constant, program notation "COS15", scale factor B0, value 0.965. Value corresponds to cosine 15.2 degrees.
- K_{dlewd0} : Constant, program notation "DLEWDO", scale factor B0, value -0.05.

- K_{drd} : Single precision set of constants, scale factor B-3, units revolutions. Multiply table entries by $21600 \times 2^{-3} = 2700$ to convert to nmi. Table starts at "VREFER +78". See below.
- K_{drda} : Single precision set of constants, scale factor B-3, units revolutions/G-unit. Multiply table entries by $2700/805$ to convert to units of nmi/fps². Table starts at "VREFER +13". See below.
- K_{drdd} : Single precision set of constants, scale factor B-1, units revolutions/V-unit. Multiply table entries by $10800/25766.1973$ to convert to units of nmi/fps ($10800 = \frac{1}{2} 21600$). Table starts at "VREFER +26". See below.
- K_{ewdl} : Constant, program notation "LEWD1", scale factor B0, value 0.15.
- K_{gmmn} : Single precision constant, program notation "FIVE", scale factor B-1, units revolutions per 0.1 second. Value is 5×2^{-14} , corresponding to a rate of about 0.5493 degrees/second.
- K_{gmx2} : Constant, program notation "GMAX/2", scale factor B0, G-units. Value is 0.16, corresponding to $\frac{1}{2} \times 8 \times (1/25)$, where first term gives equation factor of $\frac{1}{2}$, second is GMAX (in g's), and third converts to G-units (of 25 g's).
- K_{hav} : Constant, program notation "HALVE", scale factor B1, units V-units. Value is 0.5, corresponding to 25766.1973 fps (one V-unit), i.e. "VSAT".
- K_{kld} : Constant, program notation "KLD", scale factor B8, value 0.0314453125. Value corresponds to $0.01 \times 805 \times 2^{-8}$, where first term is "basic" value (corresponding to "C16"), second converts to fps², and third is scale factor.
- K_{k2d} : Constant, program notation "K2D", scale factor B7, value -0.402596836. Value corresponds to $(-1) \times 0.002 \times 25766.1973 \times 2^{-7}$, where first term is equation factor, second is "basic" value ("C17"), third converts from V-units to fps, and fourth is scale factor.
- K_{ka2} : Constant, program notation "KA2", scale factor B0, G-units. Value is 0.008, corresponding to $0.2 \times (1/25)$, where the first term is "basic" value (in g's) and second converts to G-units of 25 g's.
- K_{ka3} : Constant, program notation "KA3", scale factor B-2, G-units. Value is 0.44720497, corresponding to $90 \times (1/805) \times 2^2$, where first term is "basic" value (in fps²), second converts to G-units, and third is scale factor.
- K_{ka4} : Constant, program notation "KA4", scale factor B0, G-units. Value is 0.049689441, corresponding to $40 \times (1/805)$, where first term is "basic" value (in fps²), and second converts to G-units.

- K_{kacs} : Constant, program notation "KACOS", scale factor B5, value 0.004973592, corresponding to $(1/2\pi) \times 2^{-5}$.
- K_{kalim} : Constant, program notation "KALIM", scale factor B0, G-units. Value is 0.06, corresponding to $1.5 \times (1/25)$, where the first term is "basic" value (in g's) and second converts to G-units of 25 g's.
- K_{kasc} : Constant, program notation "KASCALE", scale factor B-13, value 0.97657358. Value corresponds to $0.0585 \times (1/0.3048) \times (1/805) \times \frac{1}{2} \times 2^{13}$, where first term is accelerometer scale factor (meters/second per count), second converts to feet, third converts to G-units, fourth compensates for the two-second accelerometer sampling cycle, and fifth is scale factor.
- K_{kc3} : Constant, program notation "KC3", scale factor B-2, value -0.0247622232. Value corresponds to $(-1) \times (1/21\ 202\ 900) \times (1/2\pi) \times (25766.1973)^2 \times (1/805) \times 2^2$, where first term is "equation" value, second is reciprocal of RE in feet ("entry radius"), third converts to revolutions, fourth converts V-units to fps, fifth converts to G-units, and sixth is scale factor.
- K_{ktl} : Constant, program notation "KT1", scale factor B29, value $15.7788327 \times 2^{-14}$. Value corresponds to $(21\ 202\ 900) \times (1/25766.1973) \times 100 \times 2\pi \times 2^{-29}$, where first term is RE ("entry radius"), second converts V-units, third converts to centi-seconds, fourth converts THETAH to radians, and fifth is scale factor.
- K_{kteta} : Constant, program notation "KTETA", scale factor B28, value $38.3495203 \times 2^{-14}$. Value corresponds to $10^3 \times 100 \times 2\pi \times 2^{-28}$, where first term is "basic" value, second converts to centi-seconds, third converts THETAH to radians, and fourth is scale factor.
- K_{kvsc} : Constant, program notation "KVSCALE", scale factor B-6, value 0.81491944. Value corresponds to $(1/25766.1973) \times 100 \times (1/0.3048) \times 2^6$, where first term is one V-unit (in fps) conversion, second converts from centi-seconds to seconds, third converts from meters to feet, and fourth is scale factor.
- K_{kwe} : Constant, program notation "KWE", scale factor B0, value $0.120056652 \times 2^{-1}$. Value corresponds to $1546.70168 \times (1/25766.1973)$, where first term is value in fps (corresponds to radius of 21 210 605 feet for a period of 86164.10 seconds) and second converts to V-units.
- K_{maxrng} : Constant, program notation "MAXRNG", scale factor B0, units revolutions. Octal value is $16631_8\ 06755_8$, corresponding to about 0.46247666 revolutions. Value selected so that noun display of DNRNGERR for this information will be +9999.9 nmi (DSKY noun conversion factor is $2\pi \times 6,373,338$ meters = 21622.4965 nmi/rev).
- K_{mldkb2} : Constant, program notation "-1/KB2", scale factor B-3, value -0.0077621078×2^4 . Value corresponds to $(-1) \times (1/0.0025) \times (1/25766.1973) \times 2^3$, where first term is equation factor, second is reciprocal of basic constant ("KB2"), third converts V-units, and fourth is scale factor.

- K_{mhsca} : Constant, program notation "-HSCALED", scale factor B1, value -0.55305018. Value corresponds to $(-1) \times 28500 \times (1/25766.1973) \times 2^{-1}$, where first term is equation factor, second is value of HS, third converts V-units, and fourth is scale factor.
- K_{mksc} : Constant, program notation "-KSCALE", scale factor B1, value -0.0312424837. Value corresponds to $(-2) \times 805 \times (1/25766.1973) \times 2^{-1}$, where first term is "equation" value, second converts G-units to fps^2 , third converts V-units to fps , and fourth is scale factor.
- K_{mkvsc} : Constant, program notation "-KVSCALE", scale factor B-6, value -0.81491944, or $-K_{kvsc}$.
- K_{mnd} : Constant, program notation "1BITDP", scale factor B0, G-units. Value is 2^{-28} , corresponding to about 10^{-7} g.
- K_{nq} : Constant, program notation "NEAR1/4", scale factor B2, value $(\frac{1}{4} - 2^{-14})$. Value corresponds to $(1 - 2^{-12})$, or about 0.99976, i.e. $\cos 1\frac{1}{4}^\circ$.
- K_{pt05g} : Constant, program notation ".05G", scale factor B0, G-units. Value is 0.002, corresponding to $0.05 \times (1/25)$, where first term is "basic" value (in g's), and second converts to G-units (of 25 g's).
- K_{ptb4} : Constant, program notation "PT1/16", scale factor B4, value 0.1×2^{-4} , corresponding to a "basic" value of 0.1.
- K_{ptone} : Constant, program notation "POINT1", scale factor B0, value 0.1.
- K_{q3} : Constant, program notation "Q3", scale factor B-1, value 0.167003132. Value corresponds to $0.07 \times 25766.1973 \times (1/21600) \times 2^1$, where first term is "basic" value (in units of nmi/fps), second converts V-units, third converts to revolutions, and fourth is scale factor.
- K_{q5} : Constant, program notation "Q5", scale factor B0, value 0.326388889. Value corresponds to $7050 \times (1/21600)$, where first term is "basic" value (in units of nmi/rad, corresponding to 0.3×23500) and second converts from nmi to revolutions.
- K_{q6} : Constant, program notation "Q6", scale factor B0, units radians. Value is 0.0349, or about 2° .
- K_{q7f} : Constant, program notation "Q7F", scale factor B0, G-units. Value is 0.0074534161, corresponding to $6 \times (1/805)$, where first term is "basic" value (in fps^2), and second converts to G-units.
- K_{q7fkdm} : Constant, program notation "Q7FKDMIN", scale factor B0, G-units. Value is 0.0080745342, corresponding to $6.5 \times (1/805)$, where first term is "basic" value (in fps^2), and second converts to G-units.

- K_{q7min} : Constant, program notation "Q7MIN", scale factor B0, G-units.
Constant is $(1 - 2^{-28})$, corresponding to about 25 g's (to disable the check in "UPCONTRL" to recompute FACTOR), or 805 fps².
- K_{q19} : Constant, program notation "Q19", scale factor B0, value 0.5.
- K_{q21} : Constant, program notation "Q21", scale factor B0, units revolutions. Value is 0.162037037, corresponding to 3500 x (1/21600), where first term is "basic" value (nmi), and second converts to revolutions.
- K_{q22} : Constant, program notation "Q22", scale factor B0, units revolutions. Value is -0.092222222, corresponding to -1992 x (1/21600), where first term is "basic" value (nmi), and second converts to revolutions.
- K_{rdtr} : Single precision set of constants, scale factor B-2, V-units. Multiply table entries by $(\frac{1}{4} \times 25766.1973)$ to convert to fps. Table starts at "VREFER +39". See below.
- K_{rtgo} : Single precision set of constants, scale factor B-3, units of revolutions. Multiply table entries by 2700 (i.e. $2^{-3} \times 21600$) to convert to nmi. Table starts at "VREFER +52". See below.
- K_{spvq} : Single precision constant, program notation "SPVQUIT", scale factor B1, V-units. Octal value is 00476g, corresponding to about 1000.2 fps.
- K_{tcdu} : Single precision constant, program notation "TCDU", scale factor B-1, value 0.1. Value corresponds to $\frac{1}{2} \times 0.1 \times 2^1$, where first term reduces angle rate from units of computing cycles (2 seconds) to seconds, second is "basic" value (to determine angle increment per 0.1 second), and third is scale factor.
- K_{vfin} : Constant, program notation "VFINAL", scale factor B1, V-units. Value is 0.51618016, corresponding to 26600 x (1/25766.1973) x 2^{-1} , where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.
- K_{vfnl} : Constant, program notation "VFINAL1", scale factor B1, V-units. Value is 0.523942273, corresponding to 27000 x (1/25766.1973) x 2^{-1} , where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.
- K_{vmin} : Constant, program notation "VMIN", scale factor B1, V-units. Value is 0.25, corresponding to a velocity of $\frac{1}{2}$ (25766.1973) fps, or about 12883.1 fps.
- K_{vmin} : Constant, program notation "VIMIN", scale factor B1, V-units. Value is 0.34929485, corresponding to 18000 x (1/25766.1973) x 2^{-1} , where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vquit} : Constant, program notation "VQUIT", scale factor B1, V-units.
 Value is 0.019405269, corresponding to $1000 \times (1/25766.1973) \times 2^{-1}$, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vrcont} : Constant, program notation "VRCONT", scale factor B1, V-units.
 Value is 0.0135836886, corresponding to $700 \times (1/25766.1973) \times 2^{-1}$, where first term is "basic" value (in fps), second converts to V-units, and third is scale factor.

K_{vrfr} : Single precision set of constants, scale factor B1, V-units.
 Multiply table entries by (2×25766.1973) to convert to fps.
 Table starts at "VREFER". See below.

$K2ROLL$: Quantity, scale factor B0 (only the sign is used, with quantity initialized to a non-zero magnitude in "STARTENT"), used to determine sign of ROLLC (used for lateral control).

KAT : Quantity, scale factor B0, G-units, computed in "INITROLL" the first cycle that D is not less than K_{pt05g} , and used as the "drag to lift up if down".

$KLAT$: Value of lateral switch gain computed in "STARTENT", scale factor B0.

LAD : Value of "maximum L/D (minimum actual vehicle L/D)" initialized in "STARTENT" from C_{adpad} , scale factor B0.

LAT : See Coordinate Transformations.

$LATANG$: Value of "lateral range", scale factor B2, units radians. Program notation also "XRNGERR".

$LATSPL$: See Display Computations.

LdD : Value of "desired lift-to-drag ratio (vertical plane)", scale factor B0. Program notation "L/D".

$LdD1$: Value of LdD used to compute ROLLC (set to LdD at start of "LIMITL/D", and perhaps subsequently changed), scale factor B0.

$LdDCALC$: Value of L/D computed in "TARGETING" (for telemetry purposes only), scale factor B0. Uses same cell as TTE (see Display Computations). Program notation "L/DCALC".

$LdDCMINR$: Value of $LAD \cos 15.2^\circ$ computed in "STARTENT", scale factor B0. Program notation "L/DCMINR".

LEQ : Value of $(VSQUARE - 1)$, scale factor B2, units of gravity (divide by 25 to convert to G-units).

$LEWD$: Value of "UPCONTRL" reference L/D, scale factor B0. Updated in "RANGER" and initialized in "INITROLL".

LINGSPL: See Display Computations.

LOD: Value of "Final Phase L/D" loaded in "STARTENT", scale factor B0, from C_{odpad} .

LONG: See Coordinate Transformations.

mVREL: Complement of relative velocity, program notation "-VREL", scale factor B1, V-units. Computed in "CM/POSE".

NV: Magnitude of VEL (velocity normalized by satellite velocity of 25766.1973 fps), scale factor B1, V-units. Program notation "V". One V-unit, i.e. 25766.1973 fps, corresponds to \sqrt{r}/RE , where $RE = 21\ 202\ 900$ (300 000' above 2.09029E7 feet).

OLDUYA: Previous value of UYA saved in "CM/POSE", scale factor B1.

POSEXIT: Single precision exit address from "CM/POSE". Set to "P62.3" at start of "P62"; set to "STARTENT" in "P63"; and set to "SCALEPOP" (the normal beginning of the entry computations) in "STARTENT".

PREDANG: Single precision value of "predicted range" computed in "PREDICT3", scale factor B-3, units revolutions (one revolution is 21600 nmi). Terms are double precision if possible.

Q2: Value of final-phase ranging information used in "RANGER" to compute ASP1, scale factor B0, units revolutions. It is initialized in "STARTENT" based on the specified value for LAD.

Q7: Value of minimum drag for "UPCONTRL", scale factor B0, G-units. Initialized in "STARTENT".

R_t , R_{ti} : See Coordinate Transformations.

RDOT: Value of altitude rate, scale factor B1, V-units.

RDOTREF: Value of reference RDOT computed in "UPCONTRL", scale factor B1, V-units.

RDRFV: Value of K_{rdtr} table for present NV, same scaling as K_{rdtr} .

ROLLC: Value of roll command output of equations, scale factor B0, units revolutions.

ROLLd180: Single precision angle, program notation "ROLL/180", giving the first Euler angle of the CM attitude (about UBX in "roll"), scale factor B-1, units revolutions.

ROLLdPIP: See General Program Control.

ROLLHOLD: See Digital Autopilot Entry Routines.

RTGON67: Value of THETAH (with negative sign if not yet past target) loaded in "TARGETNG" for use with nouns 64 and 67, scale factor B0, units revolutions. Program notation also "RTGON64".

RTGOV: Value of K_{rtgo} table for present NV, same scaling as K_{rtgo} .

SINTH: See Coordinate Transformations.

THETA: See Coordinate Transformations.

THETAH: Value of "desired range" (angle between present position vector and appropriately rotated target vector), scale factor B0, units revolutions.

TIME_dRTO: Value of computer time for which R_{ti} is valid, scale factor B28, units centi-seconds. Loaded in "STARTEN1", with program notation "TIME/RTO".

UB_X, UB_Y, UB_Z: Unit vectors, scale factor B1, computed in "CM/POSE". They define the "body axis triad" in reference coordinates: if all three gimbal angles are zero, UB_X = unit_X [REFSMMA_T] etc; otherwise, the sequence is about y,z,x axes (inner, middle, and outer respectively) to go from platform to body axes.

UN_I: Unit vector, scale factor B1, giving the "normal to the trajectory plane".

UN_{ITV}: Unit VEL vector, scale factor B1.

UX_A, UY_A, UZ_A: Unit vectors, scale factor B1, computed in "CM/POSE". They define the "trajectory triad" in reference coordinates. UX_A is in the direction of mVREL; UY_A is in the direction of $\underline{v} * \underline{r}$ (left constant if magnitude of mVREL is less than K_{spvq}), and UZ_A completes the right-handed coordinate system (in direction of UY_A * \underline{v}). Note that UX_A corresponds to "- UVA" and UZ_A corresponds to "UNA".

V1: Value of initial velocity for "UPCONTRL", scale factor B1, V-units.

VBARS: Value of $V1^2$, scale factor B2 (since V1 in units of VSAT, no explicit division by VSAT is necessary to obtain VBARS).

VEL: Value of velocity used in entry computations (inertial or relative), scale factor B1, V-units.

VL: Value of exit velocity for "UPCONTRL", scale factor B1, V-units.

VMAGI: Magnitude of \underline{V} computed at the end of "CM/POSE", scale factor B7, units meters/centi-second, computed for display purposes.

VREF: Value of reference velocity for "UPCONTRL", scale factor B1, V-units.

VS1: Value of the lesser of V1 or K_{hav} (V-unit factor), scale factor B1, V-units.

VSQUARE: Square of NV, scale factor B2(cf. discussion on VBARS).

Constants used in "PREDICT3" Reference
Table

<u>i</u>	<u>K_{vrfr}</u>	<u>K_{drda}</u>	<u>K_{drdd}</u>	<u>K_{rdtr}</u>	<u>K_{rtgo}</u>	<u>K_{aref}</u>	<u>K_{drd}</u>
<u>Stored Values</u>							
			Multiply by 2 ⁻³	Multiply by 2 ⁺³			
0	.019288	-.010337	-.0478599	-.0134001	.00137037	-.051099	.004491
1	.040809	-.016550	-.0683663	-.013947	.00385185	-.074534	.008081
2	.076107	-.026935	-.1343468	-.013462	.00874074	-.101242	.016030
3	.122156	-.042039	-.2759846	-.011813	.017148	-.116646	.035815
4	.165546	-.058974	-.4731437	-.0095631	.027926	-.122360	.069422
5	.196012	-.070721	-.6472087	-.00806946	.037	-.127081	.104519
6	.271945	-.098538	-1.171693	-.006828	.063296	-.147453	.122
7	.309533	-.107482	-1.466382	-.00806946	.077889	-.155528	.172407
8	.356222	-.147762	-1.905171	-.0109791	.098815	-.149565	.252852
9	.404192	-.193289	-2.547990	-.0151496	.127519	-.118509	.363148
10	.448067	-.602557	-4.151220	-.0179817	.186963	-.034907	.512963
11	.456023	-.99999	-5.813617	-.0159061	.238148	-.007950	.558519
12	.67918	-.99999	-5.813617	-.0159061	.294185	-.007950	.558519
					185		
(9 figures)							

<u>Basic Values</u>							
0	993.9	-.03460	-.002507	-690.4	3.63	-41.12	12.19
1	2104.2	-.05548	-.003582	-718.7	10.38	-59.99	21.75
2	3922.1	-.09028	-.007035	-693.5	23.57	-81.51	43.34
3	6293.7	-.14105	-.014454	-608.6	46.31	-93.89	96.73
4	8530.0	-.19775	-.024790	-492.6	75.48	-98.56	187.37
5	10099.5	-.23726	-.033898	-416.0	99.87	-102.30	282.13
6	14015.4	-.33041	-.061399	-351.9	170.89	-118.71	329.43
7	15949.8	-.36050	-.076826	-416.0	210.28	-125.19	465.55
8	18355.9	-.49561	-.099825	-565.8	266.80	-120.38	682.75
9	20828.1	-.64833	-.13349	-780.8	344.26	-95.42	980.53
10	23089.6	-2.0209	-.21752	-926.7	504.77	-28.10	1384.94
11	23498.4	-3.3538	-.30459	-819.7	643.03	-6.39	1508.04
12	35000.8	-3.3538	-.30459	-819.7	794.31	-6.39	1508.04
	"VREF"	"DR/DA"	"DR/DRDOT"	"RDOTREF"	"RTOGO"	"DREFR"	"DR/DL/D"
	fps	nmi/fps ²	nmi/fps	fps	nmi	fps ²	nmi

The "Basic Values" were obtained by converting the octal memory information into the corresponding decimal values (using the appropriate scaling information), and therefore differ slightly from the results that would be obtained by multiplying the (nominal) "Stored Values" by the same scaling information.

Entry Preparation

P61

EXTVBACT = 20000_g (lock out other extended verbs)

HEADSUP = -1

Perform "S61.1"

TS = 0661_{vn}

Perform "VNFLASHR": (if terminate, proceed to "GOTOPOOH")
if proceed, skip next line
otherwise, proceed to previous line

End of job

If HEADSUP > 0:

ROLLC = K_{bt14}

If HEADSUP = +0 or if HEADSUP < 0: (ROLLC not loaded for -0 HEADSUP)

ROLLC = K_{rvsc} (|HEADSUP| -1), limited \geq +0 (i.e. 0 if HEADSUP = -1)

EXTVBACT = 20000_g

Proceed to "NEWRVN"

NEWRVN

MM = T_{pptm}

Perform "STARTEN1"

Set bit 3(05GSW) of FLAGWRD6 = 0

R_{one} = R

URONE = unitR_{one}

V_{one} = V

UNI = unit (V_{one} * URONE)

If (MM - T_{pptm}) < 0:

Proceed to "NEWRVN" (an Average-G update has occurred)

Perform "S61.2"

Perform "CLEARMRK" (resets EXTVBACT to 0)

TS = 0660_{vn}

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

TS = 1663_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, skip next 2 lines
otherwise, proceed

EXTVBACT = 20000_g

Proceed to "NEWNRNVN"

TS = 62 and perform "NEWMODEX"

DNLSTCOD = 1

Proceed to "P62"

P62

Perform "S61.1"

POSEXIT = "P62.3"

Inhibit interrupts (until released, routine is "CM/DAPIC")

TS = 8192 + 8192 + TIME1 - (least significant half of T_{pptm}),
modulo 16384
(equivalent to $T_{now} - T_{pptm}$)

If (TS - 5) > 0:

TS = TS - 10 and repeat check(proceed to previous line)

TS = |TS - 5| , limited > 1 centi-second

CMdGYMDT = TS

Call "READGYMB" in TS centi-seconds (5 cs out of phase with
"READACCS")

Set bits 13(ENTRYDSP), 12(CMDAPARM), 11(GAMDIFSW), 3(O5GSW), and
1(GYMDIF) of FLAGWRD6 = 0

Set bits 4(LATSW) and 2(CMDSTBY) of FLAGWRD6 = 1

BETAd180 = 0

SWdNDX = 1

Release interrupts

AVEGEXIT = "CM/POSE" (Tag here "P62.2")

TS = 00041_g

Perform "GOPERFLR": if terminate, proceed to "GOTOPOOH"
if proceed, skip next line
otherwise, proceed to previous line

End of job

Proceed to "CM/DAPON" (returns to "P62.1")

P62.1

TS = 0661_{vn}

Proceed to "GOFLASH": if terminate, proceed to previous line
if proceed, proceed
otherwise, proceed to previous line

If HEADSUP > 0:

ROLLC_{sp} = K_{bt14} (note that, contrary to "P61", low half
not set)

If HEADSUP = +0 or if HEADSUP < 0: (ROLLC not loaded if HEADSUP -0)

ROLLC_{sp} = K_{rvsc} (|HEADSUP| - 1), limited \geq +0 (i.e. 0 if
HEADSUP = -1)

ALFACOM = C_{alfapad}

BETACOM = 0

P63FLAG = 1

ENTRYVN = 0622_{vn}

Set bit 13(ENTRYDSP) of FLAGWRD6 = 1

If bit 1 of CMDAPMOD = 1, End of job (CMDAPMOD = -0 or +1)

Proceed to "P63" (CMDAPMOD = +0; the -1 value should not exist
at this point)

P62.3 (address set in "P62" into POSEXIT; entered from "KEP2")

$$\underline{Y}_{dc} = \underline{UYA} \cos \text{ROLLC} + \underline{UZA} \sin \text{ROLLC}$$

$$\underline{X}_{dc} = (\underline{Y}_{dc} * \underline{UXA}) \sin C_{\text{alfapad}} + \underline{UXA} \cos C_{\text{alfapad}}$$

$$\underline{Z}_{dc} = \underline{X}_{dc} * \underline{Y}_{dc}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{sm} = \begin{bmatrix} \text{REFSMMAT} \end{bmatrix}$$

Perform "CALCGA"

Proceed to "ENDEXIT"

WAKEP62 Called by "EXDAP"

Establish "P63" (priority 13_g)

End of task

P63 Established by "WAKEP62" or entered from "P62.1"

TS = 63 and perform "NEWMODEX"

POSEXIT = "STARTENT"

ENTRYVN = 0664_{vn}

P63FLAG = -1

Perform "CLEANDSP"

End of job

S61.1

60GENRET = Return address

Perform "RO2BOTH"

If bit 1(AVEGFLAG) of FLAGWRD1 = 1: (true for entrance from
"P62" if V37 not used to
start P62)

Proceed to "S61.1A"

Perform "MIDTOAV2"

Call "S61.1C" in TS+1_{sp} centi-seconds (TS set when exit from
"MIDTOAV2")

End of job

S61.1C

Establish "S61.1A" (priority 13_g)

AVEGEXIT = "SERVEXIT"

Proceed to "PREREAD"

S61.1A

$TS_y = \text{unit}([REFSMAT] (V * R))$

If $TS_y \geq 0$:

If $TS_y - K_{cs30im} \geq 0$, or if unit vector overflowed:

Proceed to address specified by 60GENRET

If $TS_y < 0$:

If $-TS_y - K_{cs30im} \geq 0$, or if unit vector overflowed:

Perform "ALARM" (pattern 1427_g) (IMU "reversed")

Skip next line

Perform "ALARM" (pattern 1426_g)

$TS = 0509_{vn}$

Perform "GODSPR"

Delay 10 seconds (by putting job to sleep via "DELAYJOB")

Proceed to address specified by 60GENRET

Quantities in Computations

See also list of major variables and list of routines

60GENRET: Single precision cell used to retain return address information from "S61.1".

ALFACOM: See Digital Autopilot Entry Routines.

AVEGEXIT: See General Program Control.

BETACOM: See Digital Autopilot Entry Routines.

BETAd180: See Entry Computations.

C_{alfapad}: Single precision erasable memory constant, program notation "ALFAPAD", scale factor B-1, units revolutions, giving the nominal trim angle (expected to be a negative number).

CMDAPMOD: See Digital Autopilot Entry Routines.

CMdGYMDT: See Digital Autopilot Entry Routines.

DNLSTCOD: See Telemetry.

ENTRYVN: See Entry Computations.

EXTVBACT: See Verb Definitions.

HEADSUP: Single precision quantity used to indicate desired initial attitude of vehicle: it is positive non-zero if a ROLLC of 180 degrees (lift down) is desired, which is a "heads up" attitude. Should be set to -1 (or 0) if a ROLLC of 0 degrees (heads down/lift up) is desired. Scale factor is B14, and it is initialized at start of "P61" to -1 (before the N61 display which can be used to change it there).

K_{bt14}: Single precision constant, program notation "BIT14", scale factor B0, units revolutions. Value is 0.5, corresponding to 180 degrees. Used in "P61" as double precision information, with least significant half zero, for initializing ROLLC; in "P62.1", however, is single precision, with least significant half of ROLLC left alone.

K_{cs30im}: Constant, program notation "C(30)LIM", scale factor B1, value 0.566985. Used in program in such a way (a check on performing overflow) that effective value is 0.86603, or cosine 30 degrees: the value in memory is $1.0 - \frac{1}{2} \cos 30^\circ$.

K_{rvsc} : Dummy constant used to indicate rescaling of HEADSUP information from a scale factor of B14 to a scale factor of B0 revolutions, "value" 2^{-14} (see HEADSUP). If HEADSUP were loaded in N61 with a value of -04097, for example, then ROLLC would be set to $+90^{\circ}$.

MM: Storage for state vector time in "NEWRNVN", to ensure a consistent state vector for "S61.2", scale factor B28, units centi-seconds. It is the time tag for R_{one} and V_{one} , and hence is used in "S61.2" to compute TTE2 (see Display Computations).

P63FLAG: See Digital Autopilot Entry Routines.

POSEXIT: See Entry Computations.

R_{one} : See Display Computations.

ROLLC: See Entry Computations.

SWdNDX: See Digital Autopilot Entry Routines.

UNI: See Entry Computations (computed in "NEWRNVN" for "S61.2").

URONE: See Display Computations.

UXA, UYA, UZA: See Entry Computations.

V_{one} : See Display Computations.

General Program Control

PREREAD Entered from "TIGAVEG", "TIGON", and "S61.1C"

Establish "LASTBIAS" (priority 21_g)

Perform "PREREAD1"

Establish "NORMLIZE" (priority 32_g)

Call "READACCS" in 2 seconds

End of task

PREREAD1 Entered from "PREREAD" and "P11"

Perform "PIPASR"

Set bit 1(AVEGFLAG) of FLAGWRD1 = 1 (means Average-G not to be stopped)

Set bit 15(DRIFTFLG) of FLAGWRD2 = 0

Set bit 6(V37FLAG) of FLAGWRD7 = 1 (means Average-G running)

DVTOTAL = 0

Return

NORMLIZE Established by "PREREAD" and entered from "P11"

Inhibit interrupts

$\underline{R} = \underline{R}_{n1}$

$\underline{V} = \underline{V}_{n1}$

$\underline{T}_{pptom} = \underline{T}_{pptom1}$

Release interrupts

$\underline{TS} = \underline{R}$

Perform "CALCGRAV"

$\underline{GDT} = \underline{GDT1}$

$\underline{GOBL} = \underline{GOBL1}$

End of job

READACCS

Perform "PIPASR"

If bit 3(SBFLAG) of FLAGWRD2 = 1:

PIPAPI = PIPAPI + DELV_{sp} (each component modulo 2^{14} counts,
or about 958.464 m/sec)

PIPCTR = 2

If bit 2(CMDSTBY) of FLAGWRD6 = 1:

AOGdPIP = AOG

AIGdPIP = AIG ("least significant halves" of AOGdPIP
and AOG respectively, since a double
precision order is used)

AMGdPIP = AMG

ROLLdPIP = ROLLd180

ALFAdPIP = ALFAd180 ("least significant halves" of
ROLL information)

BETAdPIP = BETAd180

If bit 12(CMDAPARM) of FLAGWRD6 = 1:

Call "QUIKREAD" in 0.5 second

XOLDBUF = XPIPBUF

XPIPBUF = DELV_{sp} (loads uncompensated PIPA data as
read)

CMdGYMDT = 5

Call "SETJTAG" in 1.2 seconds

If bit 1(AVEGFLAG) of FLAGWRD1 = 0: (Average-G to be stopped)
(set 0 by "ISITPOO")

AVEGEXIT = "AVGEND"

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Call "READACCS" in 2 seconds

Establish "SERVICER" (priority 20_g)

Set bit 9(Test connector Outbit) of channel 11 = 1

End of task

QUIKREAD (Called by "READACCS" if entry DAP is on)

XOLDBUF = XPIPBUF

XPIPBUF = PIPA

If PIPCTR > 0:

PIPCTR = PIPCTR - 1

Delay 0.5 seconds

Proceed to "QUIKREAD"

End of task

AVGEND

ldPIPADT = (least significant half of T_{pptom})

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

Perform "AVETOMID"

Perform "PIPFREE"

MRKBUF1 = -256 (i.e. negative non-zero, keeping "REND1" from using)

Set bit 9(Test connector Outbit) of channel 11 = 0

Set bit 2(CMDSTBY) of FLAGWRD6 = 0

Set bit 6(V37FLAG) of FLAGWRD7 = 0 (Average-G not running)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

Set restart group 1 to phase 11 (i.e. 1.11, causing "PIKUP20"
to be established with priority 10_g via restart logic)

Set restart group 2 to phase 0 (i.e. no effect, since inactive)

If bit 7(RNDVZFLG) of FLAGWRD0 = 1: (and UTFLAG = 0)

Set restart group 1 to phase 11 (i.e. 1.11, see above)

Set restart group 2 to phase 13 (i.e. 2.13, causing "REDOR22"
to be established with priority 10_g via restart logic)

Proceed to "CANV37"

SERVEXIT

End of job (after resetting restart group 5 for accelerometer reading)

SERVICER

If one or more components of $DELV_{sp}$ has magnitude above K_{mxdlv} :

Perform "ALARM" (pattern 0205_g)

Proceed to "AVERAGEG"

Perform "1/PIPA"

If bit 3(05GSW) of FLAGWRD6 = 0:

$$TTE = TTE2 + T_{pptml}$$

$$DVTOTAL = DVTOTAL + K_{ppl} \left(DELV \right) \quad \text{(could be incremented twice if get a restart)}$$

Proceed to "AVERAGEG"

AVERAGEG

Perform "CALCRVG"

Inhibit interrupts

$$\underline{R} = \underline{R}_{nl}$$

$$\underline{V} = \underline{V}_{nl}$$

$$T_{pptm} = T_{pptml}$$

$$GDT_{\underline{1}} = GDT_{\underline{1}}$$

$$GOBL_{\underline{1}} = GOBL_{\underline{1}}$$

Release interrupts

Proceed to address specified by AVEGEXIT

CALCRVG

$$DELVREF = K_{ppl} DELV \left[REFSMMAT \right]$$

$$\underline{R}_{nl} = \underline{R} + K_{2s22} \left(\underline{V} + \frac{1}{2} DELVREF + \frac{1}{2} GDT_{\underline{1}} \right)$$

$$TS = \underline{R}_{nl}$$

Perform "CALCGRAV"

$$\underline{V}_{nl} = \underline{V} + DELVREF + \frac{1}{2} GDT_{\underline{1}} + \frac{1}{2} GDT_{\underline{1}}$$

Return

CALCGRAV

$$\text{UNITR} = \text{unitTS}$$

$$\text{SIN}\phi = \text{UNITR} \cdot C_{\text{unitw}}$$

$$\text{GOBL}_1 = K_{20j} \frac{K_{\text{resq}}}{|\text{TS}|^2} (0.05 - \frac{1}{4} \text{SIN}\phi^2) \text{UNITR} +$$

$$K_{2j} \frac{K_{\text{resq}}}{|\text{TS}|^2} \text{SIN}\phi C_{\text{unitw}}$$

$$\text{GDT}_1 = (K_{\text{mmudt}} / |\text{TS}|^2) (\text{GOBL}_1 + \text{UNITR})$$

Return

(GDT₁ denominator formed triple precision, then normalized and used double precision for division, after which answer shifted right before multiplication by the vector quantities).

T4RUPT Entered based upon program interrupt #4, controlled by TIME4

If DSRUPTSW < 0:

If bit 14 of DSRUPTSW = 1:

If NOUT = 0, proceed to "NODSPY"

Perform "DSPOUTSB": if return to calling address +1,
proceed to "NODSPY"
otherwise, proceed

Set bit 14 of DSRUPTSW = 0

Set TIME4 to cause program interrupt #4 in 20 ms

DSRUPTSW = DSRUPTSW + K_{1b6}

Resume

OUTO = 0

Set bit 14 of DSRUPTSW = 1

(If DSRUPTSW < 0):

Set TIME₄ to cause program interrupt #4 in 20 ms

DSRUPTSW = DSRUPTSW + K_{1b6}

Resume

TS = DSRUPTSW

If TS > 0:

DSRUPTSW = DSRUPTSW - 1, limited > 0

If |TS| = 0:

DSRUPTSW = 3

TS₁ = DSRUPTSW (program tag for TS₁ is RUPTRG1)

If DSPTAB+11 is flagged for output (i.e. is negative): (Tag "CDRVE")

Set TS = DSPTAB+11 and DSPTAB+11 = |DSPTAB+11| - 1

TS = (bits 11-1 of TS) (note if get a restart before DSPTAB+11
reloaded, "GOPROG" could put IMU into
coarse align)

DSPTAB+11 = TS

OUTO = TS + K_{rtu₁₁}

Proceed to "HANG20"

If bit 15(DSKYFLAG) of FLAGWRD5 = 0: (Tag here "DSPOUT")

Proceed to "NODSPOUT"

If NOUT = 0:

Proceed to "NODSPOUT"

Perform "DSPOUTSB": if return to calling address +1, proceed
to "NODSPOUT"
otherwise, proceed

Proceed to "HANG20"

NODSPY

OUTO = 0

i = 0

i = i + 1

DSRUPTSW = DSRUPTSW + K_{1b6}

If DSRUPTSW < 0:

Proceed to 3rd line of "NODSPY"

Set TIME4 to cause program interrupt #4 in (20 i) ms

Resume

DSPOUTSB

NOUT = NOUT - 1

TS₂ = -0 (program tag for TS₂ is DSRUPTEM)

If DSPTAB_{DSPCNT} flagged for output (is negative):

Remove output flag from DSPTAB_{DSPCNT} (form absolute value)

OUTO = (bits 15-11 of K_{rtu_{DSPCNT}}) + (bits 11-1 of DSPTAB_{DSPCNT})

Return to calling address +2 (DSKY display sent)

If DSPCNT > 0:

DSPCNT = DSPCNT - 1

Proceed to third line of "DSPOUTSB"

If TS₂ = -0:

TS₂ = +0

DSPCNT = 10

Proceed to third line of "DSPOUTSB"

NOUT = 0

Return to calling address +1 (No DSKY display sent)

HANG20

DSRUPTSW = DSRUPTSW - 22400₈ (subtracts 5 scaled B6, and also forces bit 14 to be 0)

Set TIME4 to cause program interrupt #4 in 20 ms

Proceed to "PROCEEDE"

NODSPOUT

OUTO = 0

Set TIME4 to cause program interrupt #4 in 120 ms

Proceed to "PROCEEDE"

PROCEDE

TS = channel 32 (bit 14 is PRO key complement)

If bit 14 of TS \neq bit 14(PROCDBIT) of IMODES33:

Set bit 14(PROCDBIT) of IMODES33 = bit 14 of TS

If bit 14(PROCDBIT) of IMODES33 = 0: (i.e. 1 to 0 binary transition)

Establish "PROCKEY" (priority 30₈)

If TS₁ = 0, proceed to "OPTTEST" (TS₁, program notation RUPTREG1,

If TS₁ = 1, proceed to "OPTMON" has DSRUPTSW from "T4RUPT"

If TS₁ = 2, proceed to "OPTTEST" before any modification by

If TS₁ = 3, proceed to "IMUMON" "HANG20")

SIAP1 (Entered via V36E)

Inhibit interrupts

Perform "STARTSUB"

DSPTAB+11 = bits 6(Gimbal lock) and 4(No Attitude) of DSPTAB+11,
and flag for output at next opportunity

ERCOUNT = 0

FAILREG+i = 0 (i = 0 - 2)

REDOCTR = 0

Channel 77 = 0 (i.e. reset restart monitor flip-flops)

DSRUPTSW = - K_{10b5}

Proceed to "DOFSTART"

DOFSTART

ERESTORE = 0

SMODE = 0

UPSVFLAG = 0

Channel 5 = 0

Channel 6 = 0

Channel 11 = 0

Channel 12 = 0

Channel 13 = 0

Channel 14 = 0

WTOPTION = 0

DNLSTCOD = 0

NVWORD+2 = 0

DSPFLG+2 = 0

TRKMKCNT = 0

VHFCNT = 0

EXTVBACT = 0

If bits 6 and 4 of DSPTAB+11 = 11₂: (coarse align w/gimbal lock)

Set bits 6(Enable IMU CDU Error Counters) and 4(Coarse Align)
of channel 12 = 1

Make all restart groups inactive (by setting PHASEi = +0 and -PHASEi =
-0 for i = 1 - 6)

MODREG = -0 (see "DSPMMJB")

RESTREG = 30000₈ (i.e. priority 30₈)

IMODES30 = 37411₈ (bits 14, 13, 12, 11, 10, 9, 4, and 1 set 1)

OPTIND = -1

OPTMODES = 00130₈ (bits 7, 5, and 4 to 1)

IMODES33 = 16000₈ (bits 13, 12, and 11 to 1)

T5LOC = "T5IDLOC"

FLAGWRDO = 0

FLAGWRD1 = bit 12(NODOPO1) of FLAGWRD1 (other bits set 0)

FLAGWRD2 = 0

FLAGWRD3 = bit 13(REFSMFLG) of FLAGWRD3 (other bits set 0)

FLAGWRD4 = 0

FLAGWRD5 = 0

FLAGWRD6 = 0000₄₈ (bit 3(O5GSW) is set 1)

FLAGWRD7 = 0

FLAGWRD8 = 0

FLAGWRD9 = 0

FLAGWRD10 = bit 11(HDSUPFLG) of FLAGWRD10 (other bits set 0)

FLAGWRD11 = 0

Proceed to second line of "DUMMYJOB"

GOPROG (Entered based on program interrupt #11, for a hardware
restart (interrupts inhibited before entrance))

REDOCTR = REDOCTR + 1

TS = BBANK + SUPERBNK (or'ed into bits 7-5) (BBANK is that when
restart generated)

RSBBQ = (TS, QREG)

Perform "VAC5STOR"

TS = Channel 33

Reset flip-flop bits of channel 33 (bits 15-11, done by write-type order)

If bit 15(Oscillator stop complement) of TS = 1:

If bit 14(Computer warning complement) of channel 33 = 0:

Perform "STARTSUB"

Proceed to "DOFSTART" (conclude in restart loop due e.g.
to memory parity failure)

Perform "LIGHTSET" (which may exit to "DOFSTART")

If bits 15-11 of ERESTORE \neq 0: (Note ERESTORE cell is not excluded from normal memory test sequence in "ERASLOOP")

Perform "STARTSUB"

Proceed to "DOFSTART"

If ERESTORE \neq 0:

If ERESTORE \neq SKEEP7:

Perform "STARTSUB"

Proceed to "DOFSTART"

EBANK = (bits 11-9 of SKEEP4)

$E_{\text{SKEEP7}} = \text{SKEEP5}_{\text{dp}}$

ERESTORE = 0

Perform "STARTSUB"

TS = bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 (Tag here "ELRSKIP")

Set T5LOC according to the following table:

<u>TS</u>	<u>T5LOC</u>
00 ₂	"T5IDLOC"
01 ₂	"REDORCS"
10 ₂	"REDOTVC"
11 ₂	"REDOSAT"

Set bit 14(INTINUSE) of FIGWRD10 = 0

TS = (bits 5-4 of OPTMODES)

OPTMODES = TS + 00100_g (bit 7, OCDUFBIT, set 1)

TS = bit 6(NOIMUDAP) of IMODES33

IMODES33 = TS + 16000_g (bits 13-11 to 1)

DSPTAB+11 = bits 9(Program alarm), 6(Gimbal lock), and 4(No Attitude) of DSPTAB+11, and flag for output at next opportunity

If bit 4(No Attitude) of DSPTAB+11 = 1:

Perform "SETCOARS"

Call "CA+ECE" in 0.06 seconds

TS = (bits 9, 5, 4, 3, and 1 of IMODES30) (Tag here "NOCOARSE")

IMODES30 = TS + 37000₈ (sets bits 14-10 to 1)

If bit 7(ENGNONFLG) of FLAGWRD5 = 1:

Set bit 13(SPS Engine On) of channel 11 = 1

Proceed to "GOPROG3"

ENEMA (Entered from "BAILOUT", "POODOO", and "TRACKTRM")

Inhibit interrupts

Perform "LIGHTSET" (which may exit to "DOFSTART")

Perform "STARTSB2"

Set bit 14(INTINUSE) of FLAGWRD10 = 0

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 10₂: (TVC DAP)

Call "TVCEXEC" in 0.5 seconds

Proceed to "GOPROG3"

GOPROG2 (Entered from "REV37" and "ROO")

Inhibit interrupts

Perform "STARTSB2"

Set bit 14(INTINUSE) of FLAGWRD10 = 0

Proceed to "GOPROG4"

GOPROG3

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 = 01₂: (RCS DAP)

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Perform "STOPRATE"

Proceed to "GOPROG4"

GOPROG4

If PHASE_i ≠ (-1) (-PHASE_i): (i = 1 - 6)

Perform "ALARM" (pattern 1107₈)

Proceed to "DOFSTART"

Establish "DSPMMJB" (priority 30_g)

Establish appropriate jobs and call appropriate tasks based on information in the active restart groups

If all PHASE_i = 0: (i = 1 - 6, no active restart groups)

If bit 15 of MODREG = 0: (i.e. cell not -0)

Proceed to "GOTOPOOH" (Note "VNFLASH" will put the job in core set 0, i.e. "DSPMMJB", to sleep, leaving program number display blank)

Proceed to second line of "DUMMYJOB"

LIGHTSET (Entered from "GOPROG" and "ENEMA")

If bit 7(Mark reject) of channel 16 = 1:

If bits 5-1 of channel 16 = 22_g: (Error reset)

Perform "STARTSUB"

Proceed to "DOFSTART"

If channel 15 = 22_g: (Error reset from main panel DSKY)

Perform "STARTSUB"

Proceed to "DOFSTART"

Return

STARTSUB

DNTMGOTO = "DNPHASE1"

Set TIME5 to cause program interrupt #2 in 40 ms

Set TIME4 to cause program interrupt #4 in 30 ms

Proceed to "STARTSB2"

STARTSB2

Set bits 7-3 (Operator Error, Flash, Key Release, Temperature Caution, and Uplink Activity) of channel 11 = 0 (other bits left alone)

Set TIME3 to cause program interrupt #3 in 10 ms

Set bits 11-10 (Enable Standby, Test DSKY lights) of channel 13 = 0
(other bits left alone)

Set bits 14(R21MARK), 12(P21FLAG), and 5(REVFLAG) of FLAGWRD2 = 0

Set bit 10(SKIPVHF) of FLAGWRD2 = 1 (flag to "VHFREAD" that got
restart)

Set bit 4(MARKFLG) of FLAGWRD1 = 0

Set bit 4(CYC61FLG) of FLAGWRD0 = 0

Set waitlist times to 81.93 seconds

Set waitlist task starting addresses to "SVCT3"

PRIORITY_i = -0 (i = 0 - 6)

DSRUPTSW = -0

NEWJOB = -0

Make all VAC areas available (by setting VACIUSE = "VACIUSE"
for I = 1 - 5)

DSPTAB+i = 2¹¹ and flag for output at next opportunity (i = 0 - 10)
(blanks DSKY displays)

DELAYLOC+i = 0 (i = 0 - 3)

INLINK = 0

DSPCNT = 0

CADRSTOR = 0

REQRET = 0

CLPASS = 0

DSPLOCK = 0

MONSAVE = 0

MONSAVE1 = 0

VERBREG = 0

NOUNREG = 0

DSPLIST = 0

IMUCADR = 0
LGYRO = 0
FLAGWRD4 = 0
MARKINDX = 0
EXTVBACT = 0
NOUT = 11
SELFRET = "SELFCHK"
DSPCOUNT = -19
Return

GOTOPOOH

Set bit 7(AUTOSEQ) of FLAGWRD10 = 0
Set restart group 4 to cause a start at next line (tag here "MNKGOPOO")
Perform "INITSUB"
Set bit 1(XDSPFLAG) of FLAGWRD4 = 0
Perform "AUTOCHK"
TS = 3799_{vn} (noun not permanently displayed, but must be non-zero)
Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed to previous line
otherwise, proceed to previous line

(An input of XXE will cause "V37" entrance from "MMCHANG")

V37 (Entered from "MMCHANG" or "VERB96" with TS = new program number)

MMNUMBER = TS
Set bit 7(AUTOSEQ) of FLAGWRD10 = 0
Set bit 3(TPIMNFLAG) of FLAGWRD10 = 0
Set bit 15(PCMANFLAG) of FLAGWRD10 = 0
Set bit 9(VHFRFLAG) of FLAGWRD9 = 0
MRKBUF1 = -1

If $(80 - \text{MMNUMBER}) \leq 0$:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

Proceed to "AUTO37"

If $(\text{MMNUMBER} - 30) \leq 0$:

Proceed to "AUTO37"

If $(\text{MMNUMBER} - 38) \leq 0$:

Proceed to "REND3OS"

(Note that should not do this with Average-G running and RNDVZFLG = 0, since do P20 startup)

Proceed to "AUTO37"

AUTO37 (Entered from "AUTOSET", "REND3OS", and "V37")

RESTREG = 30000_8 (i.e. priority 30_8)

If bit 6(IMUNITBT) of IMODES30 = 1:

Perform "ALARM" (pattern 1520_8)

Perform "RELDSP"

Proceed to "PINBRNCH"

If bit 7(ENGONFLG) of FLAGWRD5 = 0:

If bits 15-14(DAPBIT1, DAPBIT2) of FLAGWRD6 $\neq 10_2$:

Proceed to "ISITPOO" (not TVC DAP)

Perform "SPSOFF" (exits with interrupts inhibited)

Perform "MASSPROP"

Set TIME5 to cause program interrupt #2 in 3.1 seconds

T5PHASE = 16074 (i.e. positive non-zero)

Set bit 3 of RCSFLAGS = 1

T5LOC = "RCSATT"

Set bits 15-14 (DAPBIT1, DAPBIT2) of FLAGWRD6 = 01_2

Perform "TVCZAP"

MMNUMBER = 0

Release interrupts

Delay 0.05 seconds (by putting job to sleep via "DELAYJOB")

Channel 5 = 0

Channel 6 = 0

Proceed to "ISITPOO"

INITSUB (Entered from "GOTOPOOH" and "ROO")

Inhibit interrupts

Set bits 13(P55.LFLG), 12(P50.LFLG), 11(ATMFLAG), 10(P5OFLAG),
and 1(P29FLAG) of FLAGWRD0 = 0

Set bit 6(IDLEFAIL) of FLAGWRD1 = 0

Set bits 12(P21FLAG), 11(STEERSW), 9(IMPULSW), and 3(SBFLAG) of
FLAGWRD2 = 0

Set bits 14(GLOKFAIL) and 9(POOFLAG) of FLAGWRD3 = 0

Set bits 10(NEWTFILG) and 7(ENGONFLG) of FLAGWRD5 = 0

Set bit 13(STRULLSW) of FLAGWRD6 = 0

Set bits 13(IGNFLAG), 12(ASTNFLAG), and 11(TIMRFLAG) of FLAGWRD7 = 0

Set bit 15(SWTOVER) of FLAGWRD9 = 0

Proceed to "INITSUBA"

INITSUBA (Entered from "INITSUB", "ROO", and "TRACKTRM")

Inhibit interrupts

If bit 5(TRACKFLG) of FLAGWRD1 = 0:

Perform "STOPRATE"

(If bit 5(TRACKFIG) of FLAGWRD1 = 0):

If bit 12(MAXDBFLG) of FLAGWRD9 = 1:

Perform "SETMAXDB"

If bit 12(MAXDBFLG) of FLAGWRD9 = 0:

Perform "SETMINDB"

Release interrupts

OPTIND = -1

Return (to caller of "INITSUB" or "INITSUBA")

ISITPOO

If MMNUMBER \neq 0:

If bit 1(NODOV37) of FLAGWRD2 = 1:

Perform "ALARM" (pattern 1520_g)

Perform "RELDSP"

Proceed to "PINBRNCH"

$TS_1 = K_{nov37m}$ (Tag here "CHECKTAB")

If (bits 7-1 of $K_{prml_{TS_1}}$) \neq MMNUMBER:

If (bits 7-1 of $K_{prml_{TS_1}}$) $>$ MMNUMBER:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

If $TS_1 = 0$:

Set bit 7(Operator error) of channel 11 = 1

Perform "RELDSP"

Proceed to "PINBRNCH"

$TS_1 = TS_1 - 1$

Proceed to 7th line of "ISITPOO" (check next table entry)

MINDEX = TS_1

Set bit 15(V5ON18FL) of FLAGWRD3 = 1

If bit 6(V37FLAG) of FLAGWRD7 = 1: (Average-G running)

Set bit 1(AVEGFLAG) of FLAGWRD1 = 0

End of job (goes to "CANV37" from "AVGEND")

Proceed to "CANV37"

CANV37

CADRFLSH+2 = "ROO" + 3

Set restart group 4 to phase 1 (4.1, causing "INITDSP" to enter "ROO" if a restart)

Proceed to "ROO"

ROO

Perform "INTSTALL"

Set bits 10 (Caution Reset) and 9(Test connector Outbit) of channel 11 = 0

Set bits 14(S4B Cutoff), 13(S4B Injection Sequence start), 11(Disengage optics DAC), 10(Zero optics), 8(TVC enable), 3(Enable star tracker, not used), and 2(Enable Optics CDU Error Counters) of channel 12 = 0

Set bits 9(not assigned) and 8(not assigned) of channel 13 = 0

Perform "INITSUB"

Perform "CLEARMRK"

STARIND = 0

Set bit 14(STIKFLAG) of FLAGWRD1 = 0 (V5ON18FL = 1 from "ISITPOO" unless POO selected)

Set bit 3(Uplink Activity) of channel 11 = 0

Set bit 14(R21MARK) of FLAGWRD2 = 0

Set bit 9(EXTRANGE) of FLAGWRD10 = 0

If MMNUMBER = 0:

Perform "RELDSP" (tag here "POOH")

PHSPRDT2 = 05000₈ (i.e. priority 05₈)

Set bit 1(NODOV37) of FLAGWRD2 = 0

Set restart group 2 to phase 5 (i.e. 2.5, causing "STATINT1" to be established with priority 05₈ by restart logic)

(If MMNUMBER = 0):

Set bits 8(IMUSE) and 7(RNDVZFLG) of FLAGWRDO = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

DNLSTCOD = 0

If MMNUMBER > 0:

If bit 7(RNDVZFLG) of FLAGWRDO = 0: (Tag here "NOUVEAU")

If bit 9(UTFLAG) of FLAGWRD8 = 0:

Set bit 8(IMUSE) of FLAGWRDO = 0

DNLSTCOD = (bits 15-13 of K_{prml}MINDEX), cycled left 3 places
(i.e. to bits 3-1)

DSPFLG+2 = 76657₈ (includes setting bit 4 and 6, either one of which
causes "NORMRET" to bypass establishing "PLAYJUM1")

Set bits 10(LMTRG), 7(UPDATFLG), and 5(TRACKFLG) of FLAGWRD1 = 0

Set bit 2(R67FLAG) of FLAGWRD8 = 0

Set bits 11(CYCLFLAG), 6(SNAPFLAG), 5(P48FLAG), 4(P25FLAG),
3(TDFLAG), 2(R27UP2), and 1(R27UP1) of FLAGWRD11 = 0

Make restart groups 3, 5, and 6 inactive

If MMNUMBER = 0:

Make restart groups 1,3,4,5, and 6 inactive (leaving only group
2 set, to 2.5, for "STATINT1")

Perform "INITSUBA" (TRACKFLG now zero, so "STOPRATE" etc.)

MODREG = MMNUMBER

Proceed to "GOPROG2" (restart group 2 will cause "STATINT1"
to be established)

If MMNUMBER = 20: (Tag here "RENDVOO")

If MODREG = 20:

Make restart group 1 inactive (Tag here "KILL20")

Perform "INITSUBA"

Make restart group 2 inactive

Proceed to "REV37"

(If MMNUMBER = 20):

If bit 9(UTFLAG) of FLAGWRD8 = 0:

If bit 7(RNDVZFLG) of FLAGWRD0 = 1:

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 1
(Tag here "STATQUO")

Make restart group 4 inactive (leaves 1 and 2 set by
e.g. "AVGEND")

MODREG = MMNUMBER

Proceed to "GOPROG2" ("RELDSP" effect done in
"STARTSB2" cell resets)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

Set restart group 2 to phase 5 (i.e. 2.5, to cause
"STATINT1" to be established with priority 05₈ by
restart logic) (Tag here "STATQUO1")

Set bit 5(TRACKFLG) of FLAGWRD1 = 1

Make restart group 4 inactive

MODREG = MMNUMBER

Proceed to "GOPROG2"

If bit 9(UTFLAG) of FLAGWRD8 = 1: (Tag here "POOFIZZ")

Make restart group 2 inactive

Proceed to "REV37"

If bit 7(RNDVZFLG) of FLAGWRD0 = 1:

Proceed to "REV37"

Make restart group 1 inactive

Perform "INITSUBA"

Make restart group 2 inactive

Proceed to "REV37"

REV37

CADRFLSH+2 = "V37XEQ" + 3 (Group 4.1 set from "CANV37")

Proceed to "GOPROG2" (restart logic will cause "V37XEQ" to be
entered via "INITDSP")

V37XEQ

Inhibit interrupts

PHSPRDT₄ = 13000₈ (i.e. priority 13₈)

MMTEMP = K_{prml}MINDEX

BASETEMP = K_{fcadm}MINDEX

TS₂ = (bits 15-11 of BASETEMP) + (bits 10-8 of MMTEMP, shifted right 7 places)

TS₃ = (bits 10-1 of BASETEMP) + 2000₈

Establish a VAC area job, with starting address given in 2CADR format by (TS₃, TS₂), and with priority of 13₈

TS = (bits 7-1 of MMTEMP) and perform "NEWMODEX"

Perform "RELDSP"

End of job

PO6

Set bit 1(NODOV37) of FLAGWRD2 = 1

Inhibit interrupts

TIME2SAV = T_{now}

Perform "SCALPREP": if return to calling address +1, proceed to second line of "PO6" otherwise, proceed

SCALSAVE = TS_{dp}

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

Set bit 8(IMUSE) of FLAGWRD0 = 0

Set bit 7(RNDVZFLG) of FLAGWRD0 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

Set bit 11(Enable Standby) of channel 13 = 1

Set restart group 4 to cause it to establish "POSTAND" with priority 20₈ after a restart (erasable memory restart)

TS = 00062₈

Proceed to "GOPERF1" if terminate, proceed to previous line
if proceed, proceed to previous line
otherwise, proceed to previous line

POSTAND Entered after a restart (e.g. power-up) due to "PO6" settings
for restart group 4

Set bit 11(Enable Standby) of channel 13 = 0

Inhibit interrupts

$T_{\text{now}} = 0$

Perform "SCALPREP": if return to calling address +1, proceed to
second line of "POSTAND"
otherwise, proceed

TS = TS - SCALSAVE, rescaled to B28 centi-seconds, with sign agreement
forced

If $TS \leq -0$:

TS = TS + 2^{23} centi-seconds (correct for channel 3 overflow)

TS = TS + TIME2SAV, with sign agreement forced

$T_{\text{now}} = T_{\text{now}} + TS$

Set bit 1(NODOV37) of FLAGWRD2 = 0

Proceed to "GOTOPOOH" (where restart group 4 is set)

SCALPREP

$TS_{\text{dp}} = (\text{Channel 3, Channel 4})$ Channel 4 read twice, then a 3rd
time if 2 readings disagree. The
channel 3 reading not done until
channel 4 \neq +MAX.

Release interrupts

$TS_{\text{dp}} = TS_{\text{dp}} + K_{\text{rndbt}}$

If bits 5-1 of $TS+1 = 0$, return to calling address +1 (Channel 4
was read in first interval after
 T_{now} was incremented)

Set bits 5-1 of $TS+1 = 0$

Return to calling address +2

DELAYJOB

i = 3

If DELAYLOC+i \neq 0:

 If i = 0:

 Proceed to "BAILOUT" (pattern 3110₄₈)

 i = i - 1

 Proceed to second line of "DELAYJOB"

Call "WAKER" in (delay time) centi-seconds, with EBANK = i

DELAYLOC+i = Return address (to routine calling "DELAYJOB")

Put present job to sleep (starting address id = DELAYLOC+i)

WAKER

i = EBANK (mechanized by using BBANK with FBANK of "WAKER" = 00)

Set DELAYLOC+i = 0 and TS = DELAYLOC+i

Awaken job with starting address id = TS

End of task

ALARM

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Proceed to "ALARM2"

ALARM2

LREG = Alarm pattern (quantity in calling address +1)

ALMCADR+1 = BBANK + SUPERBNK (or'ed into bits 7-5)

If FAILREG+0 = 0:

 FAILREG+0 = LREG

 Skip next 2 lines

If FAILREG+1 = 0:

 FAILREG+1 = LREG

FAILREG+2 = LREG

Set bit 9(Program alarm) of DSPTAB+11 = 1, and flag for output
at next opportunity

Release interrupts

Return (to routine calling "ALARM" or "ALARM2")

PRIOLARM (Entered from "COM52" for alarm 0404_g)

Inhibit interrupts

LREG = TS (alarm pattern)

ALMCADR = "Calling address +1" (S-register portion)

ALMCADR+1 = BBANK of caller + SUPERBNK (or'ed into bits 7-5)

Perform "ALARM2" (starting at 3rd line)

TS = 0509_{vn}

Proceed to "PRIODSPR" (will return to routine calling "PRIOLARM")

VARALARM

Inhibit interrupts

LREG = TS

ALMCADR = "Calling address +1" (S-register portion)

Perform "ALARM2" (starting at 2nd line)

Return (address in ALMCADR)

BALLOUT

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "VAC5STOR"

LREG = Alarm pattern (quantity in calling address +1)

Perform "ALARM2" (starting at second line)

Inhibit interrupts

If in interrupt mode, return to job status (set appropriate address into cell 0017_g and then do a Resume)

Proceed to "ENEMA"

POODOO

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "VAC5STOR"

LREG = Alarm pattern (quantity in calling address +1)

Perform "ALARM2" (starting at second line)

If bit 6(V37FLAG) of FLAGWRD7 = 0: (i.e. Average-G not running)

If (bits 13-1 of EXTVBACT) = 0: (error if bit 15 = 1)

Set bit 5(STATEFLG) of FLAGWRD3 = 0

Set bit 7(AUTOSEQ) of FLAGWRD10 = 0

Set bit 13(INTGRAB) of FLAGWRD10 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

Set bit 7(RNDVZFLG) of FLAGWRD0 = 0

Set bit 5(TRACKFLG) of FLAGWRD1 = 0

Set bit 1(NODOV37) of FLAGWRD2 = 0

Make all restart groups inactive (by setting PHASEi = +0
and -PHASEi = -0 for i = 1-6)

Inhibit interrupts

If in interrupt mode, return to job status (set appropriate address
into cell 0017₈ and then do a Resume)

Proceed to "ENEMA"

SVCT3 (This task is used as part of the waitlist control, and is
entered every 81.93 seconds for that purpose)

If bit 15(DRIFTFIG) of FLAGWRD2 = 0:

End of task

If IMUCADR = +0:

Establish "NBDONLY" (priority 35₈)

End of task

Delay 5 seconds

Proceed to "SVCT3"

DUMMYJOB This routine is entered from the executive system if
no active jobs are found.

NEWJOB = -0

Release interrupts

Set bit 2(Computer Activity) of channel 11 = 0

Proceed to second line of "CHECKNJ"

CHECKNJ

SELFRET = Return address

If NEWJOB = -0:

Return (to address specified by SELFRET, BPCON of "SELFCHK")

Set bit 2(Computer Activity) of channel 11 = 1

Perform functions necessary to start new job's computations

Quantities in Computations

See also list of major variables and list of routines

ldPIPADT: See IMU Computations.

AIG: See Digital Autopilot Entry Routines.

AIGdPIP: Value of AIG sampled in "READACCS", i.e. the value of CDU sampled about 0.05 seconds before accelerometers read, scale y factor B-1, units revolutions, single precision. Program notation is "AIG/PIP".

ALFAd180: See Entry Computations (quantity is updated in "ATTRATES").

ALFAdPIP: Value of ALFAd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. It is computed based upon the same angle sample as for AIGdPIP. Program notation "ALFA/PIP".

ALMCADR, ALMCADR+1: Pair of erasable memory cells used to store the information on the calling address to the "ALARM" type routines (including "BAILOUT" and "POODOO"). Information is in 2CADR format (see 3420.5-27), with ALMCADR giving S-register information and ALMCADR+1 giving BBCON (i.e. BBANK and SUPERBNK) information. Cells can be displayed by NO8 to permit identification of the specific area of the program responsible for the generation of the fault condition.

AMG: See Digital Autopilot Entry Routines.

AMGdPIP: Value of AMG sampled in "READACCS", i.e. the value of CDU_z sampled about 0.05 seconds before accelerometers read, scale factor B-1, units revolutions, single precision.

AOG: See Digital Autopilot Entry Routines.

AOGdPIP: Value of AOG sampled in "READACCS", i.e. the value of CDU_x sampled about 0.05 seconds before accelerometers read, scale factor B-1, units revolutions, single precision.

AVEGEXIT: Cell containing in 2CADR format the starting address of the computations to be performed after "AVERAGEG" is done (program notation also AVEGEXIT). It is set to "AVGEND" by "READACCS"; to "CALCN83" by "TIGON"; to "CALCN85" by "P4ORCS" and "P4OSXTY"; to "CM/POSE" by "P62"; to "SERVEXIT" by "P67.1", "POST41", "S61.1C", and "TIGNOW"; to "S40.8" by "P4OSXTY"; and to "VHHDOT" by "P11".

BASETEMP: Cell used in "V37XEQ" to store temporarily the appropriate value of K_{fcadm} for use in generating starting address information for the job to be initiated by V37 processing.

BBANK: A computer hardware cell (address 0006_g) containing in bits 15-11 the fixed memory bank (FBANK) currently being used and in bits 3-1 the erasable memory bank number. Bits 7-5 of the constants in the proper format for BBANK loading can contain SUPERBNK information, and constants in this format are referred to as "BBCON" format (see 3420.5-27). The fact that the program retains BBANK information is used in "DELAYJOB" to retain the indexing data for the DELAYLOC cell employed (since in that routine the erasable memory bank is not significant and the fixed memory bank number is 00).

BETAd180: See Entry Computations (quantity is updated in "ATTRATES").

BETAdPIP: Value of BETAd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. Cf. ALFAdPIP.

C_{unitw}: Erasable memory vector constant, scale factor B0, program notation "UNITW". It gives the polar axis in the reference coordinate system. The x component (also called "-AYO" in old program) gives the "true to mean pole rotation about the -Y axis;" the y component (also called "AXO" in old program) gives the "true to mean pole rotation about the +X axis." Also "LMATRIX +12".

CADRFLSH+2: See Display Interface Routines.

CADRSTOR: See Data Input/Output.

CLPASS: See Data Input/Output.

CMdGYMDT: See Digital Autopilot Entry Routines.

DELAYLOC+i (i = 0 - 3): Set of four single precision cells used to contain the starting address identifications for jobs put to sleep using "DELAYJOB", in FCADR format. If an attempt is made to put more than 4 jobs to sleep using the routine, a software restart (pattern 31104_g) is produced.

DELVREF: Value of accelerometer output (scaled and compensated) converted to reference coordinates, scale factor B7, units meters/centi-second.

DNLSTCOD: See Telemetry.

DNTMGOTO: See Telemetry.

DSPCNT: Single precision counter used to cycle through the DSPTAB_i (i = 10-0) to be changed in "DSPOUTSB", scale factor B14. The DSPTAB_i word loaded at one entrance to "DSPOUTSB" is the first one checked at the next entrance, but that word will not be checked again until all the other words have been examined and, if necessary, loaded into OUTO.

DSPCOUNT: See Data Input/Output.

DSPFILG+2: See Display Interface Routines.

DSPLIST, DSPLOCK: See Data Input/Output.

DSRUPTSW: Single precision cell that is used for control of the computations that are performed when "T4RUPT" is entered. If it is not negative non-zero when "T4RUPT" is entered, it contains a number between 3 and 0, scale factor B14, used to control the branching at the end of "PROCEEDE" so as to complete a cycle (3 through 0 and back to 3 again) in 0.48 seconds. If the cell is negative non-zero, bits 2-1 contain the same "T4RUPT" branching control ("PROCEEDE" is entered and these bits employed every 0.12 seconds), while bits 12-9 contain, scale factor B6, the complement of the number of 20-millisecond intervals still to be loaded into TIME4 to achieve synchronism with the basic T4RUPT period of 0.12 seconds. Bit 14 is used to achieve an alternate selection of clearing OUTO or loading new information into OUTO. If OUTO was loaded with a non-zero value when the basic 0.12-second T4RUPT cycle was performed, then DSRUPTSW is set (in "HANG20") to DSRUPTSW - 22400_g, where the first "2" causes bit 14 to be set 0 (note that computation same as adding 55377_g), and the "24" is 5 with scale factor B6, signifying that five more 20-ms loadings of TIME4, plus the current one, are needed to satisfy the 0.12 second synchronism. If "HANG20" is entered, TIME4 is set to generate another interrupt in 0.02 second. When it is received, bit 14 of DSRUPTSW will be 0, so OUTO will be cleared, bit 14 set to 1, and TIME4 reset to 20 ms. In addition, DSRUPTSW is incremented by K_{1b6} (making the upper bits equivalent to $-4B6$, since bit 14 = 1, for a negative DSRUPTSW, represents a magnitude of 0). The next time the interrupt is generated, bit 14 will be one in "T4RUPT", so a new OUTO can be generated if required (if not, the number of remaining 20-ms intervals is determined and TIME4 and DSRUPTSW loaded immediately). In order to load all 11 (DSPTAB+0 - DSPTAB+10) DSKY display registers, a total time of $11 \times 0.04 = 0.44$ seconds is required (measured from the first loading of the first register to the end of the 20 ms wait after reset of OUTO for the last register). As part of a verb 36 fresh start, DSRUPTSW = $-K_{10b5}$, to force clearing of DSPTAB-driven relays (bit 14 = 1 due to this setting, of course) before starting the normal T4RUPT 0.12-second cycle.

DVTOTAL: Sum of magnitudes of scaled and compensated DELV each two-second entrance to "SERVICER", scale factor B7, units meters/centi-second. Quantity is initialized to 0 in "PREREAD1", and can be displayed in R3 of N40 (XXXX.X fps) or N80 (XXXXX. fps). It is the "line integral" of accumulated velocity, not the magnitude of the velocity vector increment: the magnitude is taken each two-second Average-G cycle of DELV and this scalar is added to DVTOTAL.

EBANK: See Data Input/Output.

ERCOUNT: See Testing Routines.

ERESTORE: See Testing Routines.

EXTVBACT: See Verb Definitions.

FAILREG+i (i = 0 - 2): Set of three single precision cells used to retain alarm pattern code information. They are all reset to zero by a verb 36 fresh start (or V25N9E E E E), and they can be displayed by the use of noun 09. Use of the error reset DSKY key causes FAILREG+0 and FAILREG+1 to be reset zero (FAILREG+2 is left alone). FAILREG+0 contains the first alarm pattern received after the reset; FAILREG+1 contains the second; and FAILREG+2 always has the most recent (and hence would be the same as FAILREG+0 if one alarm has occurred, or FAILREG+1 if only two alarms have occurred). If FAILREG+0 = 0, then no alarms have occurred since the last error reset, and the alarm to occur most recently is in FAILREG+2.

GDT: Value of gravity times computing interval, scale factor B8, units meters/centi-second. Program notation is "GDT/2", in recognition of the fact that it is " $\frac{1}{2}$ GDT" for a standard velocity scaling of B7 meters/centi-second. When "CALCRVG" is first entered, the quantity in GDT is the value computed the previous computation cycle.

GDT1: Value of gravity times computing interval computed in "CALCGRAV", same scaling and units as GDT (program notation "GDT1/2").

GOBL: Value of "normalized" oblateness component of gravity, scale factor B1, dimensionless (program notation "GOBL/2"). It must be multiplied by $-\mu/r^2$ to convert to units of acceleration. The quantity is used in "S40.9" to correct the output of the Lambert routine.

GOBL1: Value of GOBL computed in "CALCGRAV", same scaling as GOBL (program notation "GOBL1/2").

IMODES30, IMODES33: See IMU Computations.

IMUCADR: See IMU Computations.

INLINK: See Data Input/Output.

K_{1b6}: Single precision constant, program notation "BIT9", scale factor B6, value 00400g, corresponding to 1×2^{-6} , used to increment DSRUPTSW for 20-ms OTO update rate (cf. DSRUPTSW).

K_{2j}: Constant, program notation "2J", scale factor B-1, value $3.2469201E-3 \times 2^1$. Value corresponds to $2 \times 1.62346005E-3 \times 2^1$, where first term is an equation factor of 2, second is the nominal value of the computer-employed first gravitational harmonic coefficient (note that the "standard" value, such as that employed in K_{j2} of Orbital Integration, is $1.62345E-3$), and third is scale factor. Octal value corresponds to a harmonic coefficient value of about $1.6234554E-3$.

- K_{2s22} : Constant, program notation "2SEC(22)", scale factor B22, units centi-seconds. Value is 200×2^{-22} , corresponding to a value of 2 seconds.
- K_{10b5} : Single precision constant, program notation "PRI012", scale factor B6, value 12000_8 , corresponding to 20×2^{-6} . It is used to initialize DSRUPTSW to ensure clearing of all DSPTAB registers for a V36 fresh start (see DSRUPTSW): it forces 10 sets of OUTO settings (DSPTAB+0 and then remaining DSPTAB's) and subsequent OUTO = 0 settings before starting the normal T4RUPT 0.12-second cycle, hence the notation.
- K_{20j} : Constant, program notation "20J", scale factor B-1, value $3.2469201E-2 \times 2^1$. Value corresponds to $20 \times 1.62346005E-3 \times 2^1$, where first term is an equation factor of 20 (to give a net effect of "1 - 5 SIN ϕ^2 " for the factor it multiplies), second is the nominal value of the computer-employed first gravitational harmonic coefficient (cf. K_{2j}), and third is scale factor. Octal value corresponds to a harmonic coefficient value of about $1.62346009E-3$.
- K_{fcadm} : Table of fixed-memory starting addresses used in "V37XEQ" to set BASETEMP, arranged in FCADR format (bits 15-11 give the fixed-memory bank and bits 10-1, when added to 2000_8 , give S-register contents). Program notation for first cell in table ($i = 0$) is "FCADRMML".
- K_{mmudt} : Constant, program notation "-MU DT(E)", scale factor B44, units meters³/centi-second. Value is $-7.9720645E12 \times 2^{-44}$, corresponding to $-1 \times 3.98603225E10 \times 200 \times 2^{-44}$, where first term is an equation factor to give proper sign of result, second is value of μ (for time in centi-seconds) for the earth, third is computing interval of two seconds expressed in centi-seconds, and fourth is scale factor.
- K_{mxdlv} : Single precision constant, program notation "-MAXDELV", scale factor B14, units accelerometer counts. Stored value is -6398×2^{14} ; used in program in such a way that an effective value of 6399 is employed, giving an alarm (at the start of "SERVICER") if an accelerometer output of 6400 pulses for 2 seconds (i.e. 3200 pps) is received.

K_{nov37m} : Single precision constant, program notation "NOV37MM", scale factor B14, giving the number of table entries minus 1 in K_{prml_i} (and associated V37 tables) for new programs that can be started by the V37 logic (including the minimum key rendezvous driver programs, but excluding P00). Value is 00043₈, corresponding to decimal 35 (meaning 36 programs).

K_{pp1} : Constant, program notation "KPIPl", scale factor B-7, units meters/centi-second per accelerometer count. Value is 0.07488, corresponding to $5.85 \times 10^{-2} \times 10^{-2} \times 2^7$, where first term is nominal accelerometer scale factor in units of centimeters/second per count, second converts to meters, third converts to centi-seconds, and fourth is scale factor.

K_{prml_i} : Table of program information used in association with V37 program i changes. Table is arranged in order of decreasing program numbers (a higher i is a lower number), with first cell having program notation "PREMM1". Bits 7-1 of the cell give the program number (since program numbers are in decimal, seven bits are necessary to represent numbers above 63); bits 10-8 are the erasable memory bank number (required when the job is established); and bits 15-13 are the job's required DNLSTCOD setting for telemetry. See information below.

K_{resq} : Constant, program notation "RESQ", scale factor B59, units meters². Nominal value is $40.6809913E12 \times 2^{-59}$, where the decimal portion corresponds to the square of 6 378 165.2 meters. Octal value is 00001₈ 05000₈, however, which corresponds to the square of about 6 378 238.8 meters.

K_{rndbt} : Single precision constant, program notation "BIT5", scale factor (as used) B23, units centi-seconds. Value is 00020₈, corresponding to (as used, i.e. added to the least significant half of channel 4 time information) 0.5 centi-seconds. Used to account for the fact that channel 4 bits 5-1 "read 20₈ for the first interval after a TIME1 increment", thus causing bit 6 of channel 4 to be incremented 5 ms out of phase with TIME1 increments.

K_{rtu_i} ($i = 0 - 11$): Table of single precision constants, program notation "RELTAB", containing in bits 15-12 the required values to be loaded into bits 15-12 of OUTO for DSPTAB_i. Value in these bits is $(i + 1)$. Bit 11 is zero. The least significant 5 bits of these same constants are used for K_{rtb_I} (see Data Input/Output).

IGYRO: See IMU Computations.

LREG: Computer "L" register (address 0001₈), used to retain alarm pattern information in alarm-generation package.

MARKINDX: See Optics Computations.

MINDEX: Single precision cell, scale factor B_{14} , used to select the appropriate table entries for a V37-selected program change (loaded based on equality of MMNUMBER and bits 7-1 of K_{prml_i} with the value of i).

MMNUMBER: Single precision cell, scale factor B_{14} , used to retain the desired new program number for a V37 program change (it is the number entered in "MMCHANG" or is 0 if "V37" entered from "VERB96"). If the TVC DAP or SPS engine-on are indicated, it is set 0.

MMTEMP: Single precision cell loaded in "V37XEQ" with the value of the appropriate K_{prml} for use in generating appropriate information in connection with the job to be established by V37 processing.

MONSAVE, MONSAVE1: See Data Input/Output.

MRKBUF1: See Optics Computations.

NEWJOB: Cell set positive non-zero to indicate that a job of higher priority than the one presently being performed (the "DUMMYJOB" loop has minimum priority) has been established and is awaiting execution. Cell 0067₈ is used for the word, and must be addressed periodically in order to avoid generating a hardware restart (see hardware documentation for details).

NOUNREG: See Data Input/Output.

NOUT: Single precision cell, scale factor B_{14} , used to provide a count of the number of DSPTAB display outputs ($DSPTAB+0 - DSPTAB+10$) that are remaining to be changed: if it is non-zero but none are to be changed, it is set 0.

NVWORD+2: See Display Interface Routines.

OPTIND: See Optics Computations.

OPTMODES: See Optics Computations.

OUTO: Computer output channel 10, used to transmit relay driving information to the display system. Bits 15-12 specify the group of relays to be driven, while bits 11-1 specify their new state.

PHASE_i, -PHASE_i ($i = 1 - 6$): Set of cells used to contain program octal "phase" information, used to control program restarts (each "i" is called a "group"). To check the validity of erasable memory information (and the phase information itself), a check is made that the information in PHASE_i, when complemented, is the same as the information in -PHASE_i (for all i). A setting of +0 for PHASE_i means that the group restart controlled by that cell is "inactive" (see Section VIIC of 3420.5-27 for details of restart logic computations not covered in this writeup). In the interests of avoiding excessive detail (since the "normal" mode of operation of the computer system does not require restarts), most of the settings for restart groups are not shown in the writeups, although settings in some cases form an integral part of the program control logic, and hence are shown in such cases.

As a general guide to allocation of restart groups, the following areas where each group is used are supplied as a partial list:

- #1: P20 (setting to 1.11 causes "PIKUP20" to be entered). A setting to 1.7 is done in "CHKLINUS" for R61-specified R60 maneuver. A setting to 1.5 is used to protect "S40.9" (set by "SETUP.9"), which of course only occurs with Average-G on, when P20 doesn't run. If "ROO" concludes that P20 should not run, it resets group 1 to inactive, thus over-riding the "AVGEND" setting.
- #2: Orbital integration (including the periodic state vector update started by "STATINT1" for a 2.5 setting). A setting of 2.7 causes "R22" to be established, and 2.13 causes "REDOR22" to be established (the latter setting is done by "AVGEND", and written over by "ROO" if necessary). Group 2 is also used for protection within the R22 computations, which are done at sufficiently high priority that they would not be expected to be interrupted by mission programs except while R22 is performing "WAITONE". A restart priority lower than that at which job runs is assigned to permit the mission programs to set e.g. TRACKFLG. Restart group 2 also used to protect the R27 computations.
- #3: "S40.13"; "P11"; "ENGINOFF".
- #4: All V37-selectable programs except POO (and also other mission program sequences running outside of Average-G cycle, such as "P65.1"); "GCOMPVER"; "AZMTHCG1". Also V37 processing itself.
- #5: "READACCS" (and subsequent "SERVICER" computations); Prelaunch alignment.
- #6: "CLOKTASK"; Entry autopilot; P27; "S40.6" (via "PRE40.6").

PHSPRDT2: Single precision erasable memory cell used to contain priority or time information for erasable memory restarts using group 2 (see Section VIIC of 3420.5-27). It is set to indicate priority 05_g for POO selection in "ROO" in order to achieve proper priority information for group 2 use in the periodic POO orbital integration.

PHSPRDT4: Single precision erasable memory cell used to contain priority or time information for erasable memory restarts using group 4 (see Section VIIC of 3420.5-27). It is set to the priority of the mission program that is established by "V37XEQ", for subsequent use by erasable memory restarts.

PIPA: See IMU Computations.

PIPAPI: See Steering Computations.

PIPCTR: Single precision cell, scale factor B14, used to control the performance of "QUIKREAD". It is set to 2 in "READACCS" to allow "QUIKREAD" to be performed 1.0 and 1.5 seconds after "READACCS" ("READACCS" itself causes "QUIKREAD" to be done 0.5 seconds after "READACCS").

PRIORITY_i (i = 0 - 6): Set of cells containing in bits 14-10 the priority information on computer jobs associated with the job register set (see Section VIIB of 3420.5-27). If the cell contains -0, this indicates that the corresponding job register set is available.

QREG: Computer hardware address 0002_g, loaded with return address information after a "TC" machine language transfer instruction. See 3420.5-27 for details (program notation "Q"): the S-register portion is loaded into QREG.

R_{n1}: Temporary storage for updated value of R in "CALCRVG", used for restart protection purposes and to ensure a homogeneous state vector on the downlink (same units and scaling as R). It is used in "NORMLIZE" to initialize R for Average-G: in such cases, R_{n1} loaded either by "P11" or "MIDTOAV2".

RCSFLAGS: See Digital Autopilot Interface Routines.

REDOCTR: Single precision cell, scale factor B14, used to count the number of hardware restarts (entrances to "GOPROG") that have been performed. It is initialized to 0 as part of a verb 36 fresh start.

REQRET: See Data Input/Output.

RESTREG: See Display Interface Routines.

ROLLd180: See Entry Computations: quantity updated in "ATTRATES".

ROLLdPIP: Value of ROLLd180 sampled in "READACCS", scale factor B-1, units revolutions, single precision. See ALFAdPIP.

RSBBQ: Value of address where hardware restart (causing entrance to "GOPROG") was generated. Most significant half contains BBCON (see BBANK above) information and least significant half has Q-register information.

SCALSAVE: Value of (Channel 3, Channel 4) saved in "PO6", scale factor B23, units centi-seconds. Used to restore the proper value of the computer clock (modulo 2²³ centi-seconds or about 23.3 hours) after a period of low-power operation.

SELFRET: Cell containing return address information to routine calling "CHECKNJ" (a location in the computer self-check memory bank, hence the name). Must be preset as part of the initial conditions routine so that the "DUMMYJOB" routine will function properly.

SINØ: Value of sine of latitude computed in "CALCGRAV", scale factor B1.

SKKEPi: See Testing Routines.

SMODE: See Testing Routines.

STARIND: See Inflight Alignment.

SUPERBNK: Value of computer channel 07, containing in bits 7-5 the specification of the "super bank" setting: value of 0-3 causes FBANK settings of 30_8 - 37_8 to read out correspondingly numbered fixed memory banks; a setting of 4 causes FBANK settings of 30_8 - 33_8 to read out fixed memory banks 40-43 (used e.g. for most DSKY processing). Also referred to as FEXT.

T_{pptml}: See IMU Computations.

T5LOC: See Digital Autopilot Interface Routines.

T5PHASE: See Digital Autopilot RCS Routines.

TIME2SAV: Value of T_{now} retained in "PO6", scale factor B28, units centi-seconds (the corresponding value of channels 3 and 4 is in SCALSAVE), used to permit restoration of the proper value of the computer clock (T_{now}) after a period of low-power operation.

TIME3, TIME4, TIME5: Computer hardware erasable memory counter clock cells, whose overflow cause program interrupts #3, #4, and #2 respectively (all are scale factor B14 in units of centi-seconds, and are set to $(16384 - t)$, with t in centi-seconds, to cause overflow in "t" centi-seconds). See 3420.5-27.

TRKMKCNT: See Measurement Incorporation.

TTE, TTE2: See Display Computations.

UPSVFLAG: See Uplink Processing.

V_{nl}: Temporary storage for updated value of V in "CALCRVG", used for restart protection purposes and to ensure a homogeneous state vector on the downlink (same units and scaling as V). It is used in "NORMLIZE" to initialize V for Average-G: in such cases, V_{nl} is loaded either by "P11" or "MIDTOAV2".

VACIUUSE (I = 1-5): Control cell indicating, if zero, that the associated VAC area is assigned; otherwise, it contains its own address (hence VAC3USE +1 would be the address of the first available cell in the 43-cell VAC area). Quantities indicated as in "push-down list", or by their "relative address" identification, are stored in a VAC area. See 3420.5-27 for more details.

VERBREG: See Data Input/Output.

VHFCNT: See Measurement Incorporation.

WTOPTION: See Optics Computations.

XOLDBUF: Value of previous XPIPBUF, containing accelerometer samples with scale factor Bl4, units pulses. Cells are all single precision, with program notations XOLDBUF, YOLDBUF, and ZOLDBUF respectively. Use same cells as ADOT₁+1, ADOT₂, and ADOT₂+1 (for telemetry purposes): see Digital Autopilot^{sp} RCS Routines.

XPIPBUF: Set of three consecutive erasable memory cells, scale factor Bl4, units pulses (accelerometer counts, uncompensated). Cells are all single precision, with program notations XPIPBUF, YPIPBUF, and ZPIPBUF respectively. Use same cells as ADOT₀, ADOT₀+1, and ADOT_{1sp} (for telemetry purposes): see Digital Autopilot RCS Routines. If Entry DAP is running (bit 12 of FLAGWRD6 = 1), loaded in "READACCS" with raw accelerometer sample, and updated 0.5, 1.0, and 1.5 seconds later with present accelerometer count (in "QUIKREAD"), with previous value of XPIPBUF being stored in XOLDBUF.

Verb 37 Major Mode Tables

<u>Mode</u>	<u>Address</u>	<u>EBANK</u>	<u>DNLSTCOD</u>	<u>Mode</u>	<u>Address</u>	<u>EBANK</u>	<u>DNLSTCOD</u>
00	---	-	0	48	"P48CSM"	6	3
01	"GTSCPSS"	5	0	50	"P50"	7	0
06	"P06"	4	0	51	"P51"	5	0
20	"PROG20"	6	2	52	"PROG52"	5	0
21	"PROG21"	4	2	53	"P51"*	5	0
25	"P25CSM"	6	2	54	"PROG52"*	5	0
29	"P29"	7	2	55	"P55"	7	0
30	"P30"	7	2	61	"P61"	6	3
31	"P31"	4	2	62	"P62"	6	1
32	"P32"	4	2	77	"P77"	7	2
33	"P33"	4	2	81	"P81"	6	2
34	"P34"	4	2	82	"P82"	6	2
35	"P35"	4	2	83	"P83"	6	2
36	"P36"	4	2	84	"P84"	6	2
37	"P37"	7	2	85	"P85"	6	2
38	"P38"	4	2	86	"P86"	6	2
40	"P40CSM"	6	3	87	"P87"	6	2
41	"P41CSM"	6	3	88	"P88"	6	2
47	"P47CSM"	6	3				

*Table entries for P53 and P54 indicate starting address of "P53" and "P54" respectively, but these tags are defined to be equal to "P51" and "PROG52", as shown.

P81-P88 are control drivers for minimum key rendezvous cycling: they are not intended for manual selection.



IMU Computations

RO2BOTH Entered from "P4OS/F" (P40/P41), "P47CSM" (P47/P48), "P50",
"P55", "PROG20", "PROG52" (P52/P54), "S61.1" (P61/P62), and
"V89CALL"

If bit 13(REFSMFLG) of FLAGWRD3 = 0:

If bit 9(IMUOPBIT) of IMODES30 = 0: (IMU turned on)

TS = 0220₈

If bit 9(IMUOPBIT) of IMODES30 = 1:

TS = 0210₈

Perform "VARALARM"

Proceed to "GOTOPOOH"

Set bit 8(IMUSE) of FLAGWRD0 = 1

Return

PIPASR

$T_{pptoml} = T_{now}$

Set least significant halves of DELV = 0 (flag for telemetry that
DELV not compensated)

Set DELV = PIPA and PIPA = -0 PIPA reset so that no counts are lost.
Only most significant halves of DELV
loaded. Special precautions taken to
make routine restartable: if get a
restart, ldPIPADT set to K_{2sec} .

Return

1/PIPA

If GCOMPSW < 0, Return

Inhibit interrupts

Perform the following for $i = z, y, x$:

$$DELV_i = DELV_{i_{sp}} + C_{pipascf_i} DELV_{i_{sp}} - C_{pipabias_i} ldPIPADT$$

Release interrupts

$$GCOMP_x = GCOMP_x - C_{ad,x} DELV_x + C_{sr,x} DELV_y - C_{nbd,x} ldPIPADT$$

$$GCOMP_y = GCOMP_y - C_{ad,y} DELV_y + C_{sr,y} DELV_z - C_{nbd,y} ldPIPADT$$

$$GCOMP_z = GCOMP_z - C_{ad,z} DELV_z - C_{sr,z} DELV_y + C_{nbd,z} ldPIPADT$$

If magnitude of any $GCOMP_i$ ($i = x,y,z$) $\gg K_{cpck}$:

Establish "1/CHECK" (priority 21_g)

Return

NBDONLY Established by "SVCT3"

If $GCOMP_{SW} < 0$, End of job

Inhibit interrupts

If bit 15(DRIFTFLG) of FLAGWRD2 = 0, End of job

Set $TS = ldPIPADT$ and $ldPIPADT = TIME1$

Release interrupts

$TS = ldPIPADT - TS$ (i.e. new value minus old value)

Proceed to "NBD2"

LASTBIAS Established by "PREREAD"

Perform "PIPUSE"

If $GCOMP_{SW} < 0$, End of job

$TS = (\text{least significant half of } T_{pptml}) - ldPIPADT$

$ldPIPADT = K_{2secp}$

Proceed to "NBD2"

NBD2

If $TS = 0$, End of job

If $TS < 0$:

$TS = TS + 16384$ centi-seconds (correct for TIME1 overflow)

$GCOMP_x = GCOMP_x - C_{nbd,x} TS$ (TS scaling here B19)

$GCOMP_y = GCOMP_y - C_{nbd,y} TS$

$GCOMP_z = GCOMP_z + C_{nbd,z} TS$

If magnitude of any $GCOMP_i$ ($i = x,y,z$) $\gg K_{cpck}$:

Proceed to "1/GYRO"

End of job

1/CHECK

If IMUCADR \neq 0:

End of job

Proceed to "1/GYRO"

1/GYRO

Shift GCOMP right 7 places (bits corresponding to 2^{-8} pulses and below are lost), making scaling B21 pulses

TS = "GCOMP"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, proceed
otherwise, proceed

Shift least significant half of GCOMP left 7 places, making scaling of double precision number (for most significant halves 0) B14 pulses. Bits 14-8 of least significant half lost (expected to be 0 for normal operation)

End of job

IMUMON

Entered from "PROCEEDE" every 0.48 seconds

TS₁ = (bits 15-11 and 9 of channel 30)

If TS₁ = (bits 15-11 and 9 of IMODES30):

Proceed to "TNONTEST"

TS = 0 (in RUPTREG2 cell)

Set those bits of TS = 1 that are different in IMODES30 and TS₁
(in the range of bits 15-11 and 9)

Set (bits 15-11 and 9 of IMODES30) = TS₁

If bit 15 of TS = 1: (IMU temperature)

TS = bits 14-1 of TS (Tag here "TLIM")

If bit 15(TLIMBIT) of IMODES30 = 1:

Set bit 4(Temperature Caution) of channel 11 = 1

If bit 15(TLIMBIT) of IMODES30 = 0:

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 4(Temperature Caution) of channel 11 = 0

If bit 14 of TS = 1: (IMU turn-on request)

If bit 2(DLAYFAIL) of IMODES30 = 0: (Tag here "ITURNON")

If bit 14(TONISSBT) of IMODES30 = 0:

Set bit 7(ITNONLBT) of IMODES30 = 1

If bit 14(TONISSBT) of IMODES30 = 1:

If bit 15(ISS Turn-on Delay Complete) of channel 12 = 0:

Set bit 2(DLAYFAIL) of IMODES30 = 1

Perform "ALARM" (pattern 0207_g)

If bit 13 of TS = 1: (IMU fail)

Perform "SETISSW"

If bit 12 of TS = 1: (IMU CDU fail)

Perform "SETISSW"

If bit 11 of TS = 1: (IMU cage command)

If bit 11(CAGEBIT) of IMODES30 = 1: (Tag here "IMUCAGE")

Set bit 4 (No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU = 0

Call "UNZ2" in 0.32 seconds

Proceed to "C33TEST"

Set bits 15-10 (IMU and CDU drive enable and gyro drive enable) of channel 14 = 0

Set bits 8(TVC Enable), 6-4 (Enable IMU CDU Error Counters, Zero IMU CDU, and Coarse Align IMU), and 2(Enable Optics Error Counters) of channel 12 = 0

Set bit 7(ENGONFIG) of FLAGWRD5 = 0

Set bit 13(SPS Engine On) of channel 11 = 0

Perform "CAGESUB" (starting at 3rd line)

(If bit 11 of TS = 1):

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and
13(REFSMFLG) of FLAGWRD3 = 0

CDUXCMD = -0

GYROCMD = -0

Set bits 9-6 (Gyro drive selection) of channel 14 = 0

If bit 9 of TS = 1: (IMU operate)

If bit 9(IMUOPBIT) of IMODES30 = 1: (Tag here "IMUOP")

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2,
and 13(REFSMFLG) of FLAGWRD3 = 0

Set bit 9(UTFLAG) of FLAGWRD8 = 0

If bit 8(IMUSE) of FLAGWRD0 = 1:

Perform "ALARM" (pattern 0214_g)

Set bits 8(IMUSE) and 7(RNDVZFLG) of FLAGWRD0 = 0

If bit 9(IMUOPBIT) of IMODES30 = 0:

If bit 2(DLAYFAIL) of IMODES30 = 0:

Set bit 7(ITNON1BT) of IMODES30 = 1

Proceed to "TNONTEST"

TNONTEST

If bit 7(ITNON1BT) of IMODES30 = 0:

Proceed to "C33TEST"

If bit 8(ITNON2BT) of IMODES30 = 0:

Set bit 8(ITNON2BT) of IMODES30 = 1

Proceed to "C33TEST"

Set bits 8-7(ITNON2BT, ITNON1BT) of IMODES30 = 0 (Tag here "PROCTNON")

If bit 14(TONISSBT) of IMODES30 = 1:

If bit 4(Coarse Align) of channel 12 = 1: (Tag here "OPONLY")

Proceed to "C33TEST"

If bit 8(IMUSE) of FLAGWRDO = 1:

Proceed to "C33TEST"

Set bits 6-3 (IMUNITBT, NOACCALM, IMUFINHT, ICDUINHT) and bit 1(ACCFINHT) of IMODES30 = 1

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU = 0

Call "UNZ2" in 0.32 seconds

Proceed to "C33TEST"

If bit 9(IMUOPBIT) of IMODES30 = 1:

Perform "ALARM" (pattern 0213₈)

Perform "CAGESUB"

Call "ENDTNON" in 90 seconds

Proceed to "C33TEST"

C33TEST

TS₁ = (bits 13-11 of channel 33)

Reset channel 33 flip-flops (bits 15-11, done by a Write-type order)

If TS₁ = (bits 13-11 of IMODES33):

Proceed to "GLOCKMON"

TS = 0

Set those bits of TS = 1 that are different in IMODES33 and TS₁
(in the range of bits 13-11)

Set (bits 13-11 of IMODES33) = TS₁

If bit 13 of TS = 1: (Accelerometer fail)

Set bit 10(PIPAFLBT) of IMODES30 = bit 13(PIP2FLBT) of IMODES33
(Tag here "PIPFALL")

Perform "SETISSW"

If bit 1(ACCFINHT) of IMODES30 = 1:

If bits 10-7 (PIPAFLBT, IMUOPBIT, ITNON2BT, ITNON1BT) and
bit 5(NOACCALM) of IMODES30 all = 0:

Perform "ALARM" (pattern 0212_g)

If bit 12 of TS = 1: (Downlink too fast)

If bit 12(DNLKFAIL) of IMODES33 = 0: (Tag here "DNTMFAST")

Perform "ALARM" (pattern 1105_g)

If bit 11 of TS = 1: (Uplink too fast)

If bit 11(UPLKFAIL) of IMODES33 = 0: (Tag here "UPTMFAST")

Perform "ALARM" (pattern 1106_g)

Proceed to "GLOCKMON"

GLOCKMON Entered every 0.48 second after computations of "IMUMON"

$$TS = |CDU_z| + K_{m70degs}$$

If $TS \leq 0$:

If bit 6(Gimbal Lock) of DSPTAB+11 = 0, Resume

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 6(Gimbal Lock) of DSPTAB+11 = 0, and flag for
output at next opportunity

Resume

If $(TS + K_{m15degs}) > 0$:

If bit 4(Coarse Align) of channel 12 = 0:

If bits 14-13 of DAPDATR1 = 11₂: (Saturn DAP)

If bit 1(AVEGFLAG) of FLAGWRD1 = 1:

Skip next 2 lines

Perform "SETCOARS"

Call "CA+ECE" in 0.06 seconds

If bit 6(Gimbal Lock) of DSPTAB+11 = 1, Resume

If bit 6(IMUNITBT) of IMODES30 = 1, Resume

Set bit 6(Gimbal Lock) of DSPTAB+11 = 1, and flag for output at next opportunity

Resume

SETISSW

TS_3 = (bits 4(IMUFINHT), 3(ICDUINHT) and 1(ACCFINHT) of IMODES30)

Shift TS_3 left 9 places (loading bits 13, 12, and 10)

TS_4 = TS_3 or IMODES30 (bits 13, 12, and 10 of IMODES30 are IMUFLBIT, ICDUFLBT, and PIPAFLB)

TS_5 = bits 13, 12, and 10 of ($-TS_4$)

If $TS_5 \neq 0$: (means that 13/4, 12/3, and 10/1 bit pairs not have at least one of the two bits of the pair a binary 1)

$TS = TS_5 - 1$ (communication cell with "VARALARM", not affecting the TS of e.g. "IMUMON")

Perform "VARALARM"

Set bit 1(ISS Warning) of channel 11 = 1

Return

If bit 1(LMPTSTBT) of IMODES33 = 0:

Set bit 1(ISS Warning) of channel 11 = 0

Return

CAGESUB

Set bit 15(ISS Turn-on Delay Complete) and bit 6(Enable IMU CDU Error Counters) of channel 12 = 0

Set bit 5(Zero IMU CDU) and bit 4(Coarse Align) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 1, and flag for output at next opportunity (Tag here "CAGESUBL")

Set bits 6-3 (IMUNITBT, NOACCALM, IMUFINHT, ICDUINHT) and bit 1(ACCFINHT) of IMODES30 = 1

Set bit 6(NOIMUDAP) of IMODES33 = 1

Return

ENDTNON

If bit 2(DLAYFAIL) of IMODES30 = 1:

Set bit 2(DLAYFAIL) of IMODES30 = 0

If bit 14(TONISSBT) of IMODES30 = 0:

Delay 90 seconds

Proceed to "ENDTNON"

If bit 8(IMUSE) of FLAGWRDO = 1:

Proceed to "IMUBAD"

End of task

Set bit 15(ISS Turn-on Delay Complete) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at next opportunity

Proceed to "UNZ2"

UNZ2

CDU = 0

Set bit 5(Zero IMU CDU) and bit 4(Coarse Align) of channel 12 = 0

Delay 7.9 seconds

Set bits 6(IMUNITBT), 4(IMUFINHT), and 3(ICDUINHT) of IMODES30 = 0

Set bit 6(NOIMUDAP) of IMODES33 = 0

Perform "SETFISSW"

Set bit 15(ISS Turn-on Delay Complete) of channel 12 = 0

Call "PFAILOK" in 4 seconds

End of task

PFAILOK Called by "UNZ2"

If bit 6(IMUNITBT) of IMODES30 = 1, End of task

Set bit 10(PIPAFLBT) of IMODES30 = 1

Set bit 13(PIP2FLBT) of IMODES33 = 1

Set bit 5(NOACCALM) of IMODES30 = 0

Perform "SETISSW"

End of task

IFAILOK Called by "IMUFINE"

If bit 6(IMUNITBT) of IMODES30 = 1, End of task

If bit 4(Coarse Align) of channel 12 = 1, End of task

Set bit 13(IMUFLBIT) of IMODES30 = 1

Set bit 4(IMUFINHT) of IMODES30 = 0

Perform "SETISSW"

End of task

IMUZERO Entered from "GEOIMUTT", "GTSCPSS", and "VBZERO"

Inhibit interrupts

If bit 6(Gimbal Lock) and bit 4(No Attitude) of DSPTAB+11 = 11₂:

 Perform "ALARM" (pattern 0206₈)

 IMUCADR = -0

 Release interrupts

 Return

If bit 6(IMUNITBT) of IMODES30 = 1:

 IMUCADR = -0

 Release interrupts

 Return

Set bits 6-5 (NOIMUDAP, IMUZROBT) of IMODES33 = 1

Set bits 4-3 (IMUFINHT, ICDUINHT) of IMODES30 = 1

Set bit 6(Enable IMU CDU Error Counters) and bit 4(Coarse Align)
of channel 12 = 0

Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at
next opportunity

Set bit 5(Zero IMU CDU) of channel 12 = 1

CDU = 0

Call "IMUZERO2" in 0.32 seconds

If bit 9(IMUOPBIT) of IMODES30 = 1:

 Perform "ALARM" (pattern 0210₈)

Release interrupts

Return

IMUZERO2

If bit 6(IMUNITBT) of IMODES30 = 1:

 Proceed to "IMUBAD"

CDU = 0

Set bit 5(Zero IMU CDU) of channel 12 = 0

Delay 7.9 seconds

If bit 6(IMUNITBT) of IMODES30 = 1:

 Proceed to "IMUBAD"

Set bits 4-3 (IMUFINHT, ICDUINHT) of IMODES30 = 0

Set bits 6-5 (NOIMUDAP, IMUZROBT) of IMODES33 = 0

Perform "SETISSW"

Proceed to "ENDIMU"

IMUCOARS

Inhibit interrupts

If bit 6(IMUNITBT) of IMODES30 = 1:

 IMUCADR = -0

 Release interrupts

 Return

Perform "SETCOARS"

Call "COARS" in 0.06 seconds

Release interrupts

Return

COARS

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "IMUBAD"

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 1

COMMAND = THETAD - CDU ones complement difference formed, and
rounded shift used to rescale to Bl rev.

Delay 0.02 seconds

Proceed to "COARS2"

COARS2

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "IMUBAD"

$TS_1 = 0$ (in ITEMP1 cell)

Perform the following for $i = Z, Y, X$:

If $|COMMAND_i| = 0$:

CDUICMD = -0

If $|COMMAND_i| \neq 0$:

$TS_1 = TS_1 + 1$

$TS = |COMMAND_i| + K_{mcommax}$

If $TS > 0$:

$COMMAND_i = TS \text{ sgn } COMMAND_i$

$CDUICMD = -K_{mcmxm} \text{ sgn } COMMAND_i$

If $TS \leq 0$:

CDUICMD = COMMAND_i

COMMAND_i = 0

If $TS_1 > 0$: (i.e. non-zero commands to be sent)

Set bits 15-13(Gate outputs from CDUXCMD) of channel 14 = 1

Delay 0.6 seconds

Proceed to "COARS2"

Delay 1.5 seconds

If any($|CDU_i - THETAD_i| + K_{corst}$) > 0: (i = x,y,z)

Perform "ALARM" (pattern 0211g)

Proceed to "IMUBAD"

Proceed to "ENDIMU"

CA+ECE

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 1

End of task

SETCOARS

If bit 4(Coarse Align) of channel 12 = 1, Return

Set bit 6(Enable IMU CDU Error Counters) of channel 12 = 0

Set bit 10(Gyro Output Drive) of channel 14 = 0

GYROCMD = -0

Set bit 4(Coarse Align) of channel 12 = 1

Set bit 4(No Attitude) of DSPTAB+11 = 1, and flag for output at next opportunity

Set bit 6(NOIMUDAP) of IMODES33 = 1

Set bit 4(IMUFINHT) of IMODES30 = 1

Set bits 5(TRACKFLG) of FLAGWRD1, 15(DRIFTFLG) of FLAGWRD2, and 13(REFSMFLG) of FLAGWRD3 = 0

Return

IMUFINE

Tag also "IMUFIN20"

Inhibit interrupts

If bit 6(IMUNITBT) of IMODES30 = 1:

IMUCADR = -0

Release interrupts

Return

Set bits 5(Zero IMU CDU) and 4(Coarse Align) of channel 12 = 0
Set bit 6(NOIMUDAP) of IMODES33 = 0
Set bit 4(No Attitude) of DSPTAB+11 = 0, and flag for output at
next opportunity
Call "IFAILOK" in 5.12 seconds
Call "IMUFINED" in 2 seconds
Release interrupts
Return

IMUFINED

If bit 6(IMUNITBT) of IMODES30 = 1:
 Proceed to "IMUBAD"
Proceed to "ENDIMU"

PIPUSE Entered from "ESTIMS" and "LASTBIAS"

PIPA = -0
If bit 6(IMUNITBT) of IMODES30 = 1, Return
Inhibit interrupts
Set bit 1(ACCFINHT) of IMODES30 = 0
Perform "SETISSW"
Release interrupts
Return

PIPFREE Entered from "AVGEND"

Inhibit interrupts
Set bit 1(ACCFINHT) of IMODES30 = 1
If bit 10(PIPAFLBT) of IMODES30 = 1:
 Release interrupts
 Return
Perform "ALARM" (pattern 0212_g)
Inhibit interrupts

Perform "SETISSW"
Release interrupts
Return

IMUPULSE

If bit 6(IMUNITBT) of IMODES30 = 1:

IMUCADR = -0

Return

If LGYRO = 0:

Set bit 6(Enable Gyro Power Supply) of channel 14 = 1

Call "STRTGYRO" in 0.04 seconds

If LGYRO > 0:

Put present job to sleep (starting address id = "GWAKE")

When awakened, put to sleep again if LGYRO > 0

Call "STRTGYRO" in 0.01 seconds

LGYRO = TS (TS set on entrance to "IMUPULSE" with address of first gyro command)

Force sign agreement of $E_{\text{LGYRO}_{dp}}$ (i.e. each double precision axis)

Return

STRTGYRO

Set bits 10-7 (Gyro select bits) of channel 14 = 0

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "DONTPULS"

Proceed to "STRTGYR2"

STRTGYR2

If bits 14-13 of LGYRO = 00₂: (initial condition value)

TS₃ = 2

TS₄ = Bit 8 (Gyro Y-axis selection)

Proceed to "GSELECT"

If bits 14-13 of LGYRO = 01₂:

$$TS_3 = 2$$

$$TS_4 = \text{Bits 8-7 (Gyro Z-axis selection)}$$

Proceed to "GSELECT"

If bits 14-13 of LGYRO = 10₂:

$$TS_3 = 0$$

$$TS_4 = \text{Bit 7 (Gyro X-axis selection)}$$

$$LGYRO = LGYRO - 4 \quad (\text{sets for address of X component})$$

Proceed to "GSELECT"

$$LGYRO = 0 \quad (\text{bits 14-13 of LGYRO} = 11_2)$$

Awaken job (if any) put to sleep with starting address id = "GWAKE"

Proceed to "IMUFINED"

GSELECT

$$LGYRO = LGYRO + TS_3 + 10000_8 \quad (\text{octal number is bit 13, to cause y,z,x output sequence})$$

$$ADRLGYR = (\text{bits 11-1 of LGYRO})$$

$$TS = E_{ADRLGYR_{dp}}$$

$$\text{If } (|TS| + K_{mgyrmm}) \leq 0:$$

Proceed to "STRGTGYR2"

$$TS = TS + K_{gyrfrc} \text{sgn } TS \quad (\text{modulo 1 revolution or } 2^{21} \text{ pulses})$$

If $TS < 0$:

$$TS_4 = TS_4 + \text{Bit 9} \quad (\text{Negative gyro torquing})$$

Set those bits of channel 14 = 1 that are 1 in TS_4 (selects proper axis and sign)

$$E_{ADRLGYR+1} = (\text{bits 7-1 of } |E_{ADRLGYR+1}|) \text{sgn } E_{ADRLGYR_{dp}} \quad (\text{save fractional part})$$

$$TS = K_{gyrsc} |TS|$$

$$TS_1 = \text{integral part of } (TS/8192)$$

$TS_2 = TS - 8192 TS_1$ (in range 0 - 8191)

If $TS_1 > 1$:

$E_{ADRLGYR+0} = TS_1 - 2$ (Tag here "LONGGYRO")

$GYROCMD = 8192 + TS_2$

Call "8192AUG" in ($K_{bt10} GYROCMD - 3$) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

If $TS_1 = 1$:

$TS_2 = TS_2 + 8192$

$E_{ADRLGYR+0} = 0$

$GYROCMD = TS_2$

If bit 4(IMU Coarse Align) of channel 12 = 0:

If bit 6(IMUNITBT) of IMODES30 = 0:

Call "TWO PULSE" in ($K_{bt10} GYROCMD + 3$) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

Proceed to "DONT PULS"

8192AUG

If bit 4(IMU Coarse Align) of channel 12 = 1:

Proceed to "DONT PULS"

If bit 6(IMUNITBT) of IMODES30 = 1:

Proceed to "DONT PULS"

If $E_{ADRLGYR+0} > 0$:

$E_{ADRLGYR+0} = E_{ADRLGYR+0} - 1$

$GYROCMD = GYROCMD + 8192$

Call "8192AUG" in ($K_{bt10} GYROCMD - 3$) centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

GYROCMD = GYROCMD + 8192

If bit 4(IMU Coarse Align) of channel 12 = 0: (redundant with check already done)

If bit 6(IMUNITBT) of IMODES30 = 0:

Call "TWOPULSE" in $(K_{bt10} \text{ GYROCMD} + 3)$ centi-seconds

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

Proceed to "DONTPULS"

TWOPULSE (Purpose is to have "last" torquing always negative)

Set bit 9(Negative Gyro Torquing) of channel 14 = 0

GYROCMD = 2 (sent in less than 1 ms)

Set bit 10(Gyro Output Drive) of channel 14 = 1

Call "STRTGYRO" in 0.01 second (logic for this requires over 1 ms)

Set bit 9(Negative Gyro Torquing) of channel 14 = 1

GYROCMD = 2

Set bit 10(Gyro Output Drive) of channel 14 = 1

End of task

DONTPULS

IGYRO = 0

Awaken job (if any) put to sleep with starting address id = "GWAKE"

Proceed to "IMUBAD"

ENDIMU

If bit 1(ISS Warning) of channel 11 = 1:

Proceed to "IMUBAD"

If IMUCADR = +0:

IMUCADR = -1

End of task

Awaken job put to sleep in "IMUSTALL" and cause it to start at (IMUCADR +1), indicating a success

IMUCADR = +0

End of task

IMUBAD

If IMUCADR = +0:

IMUCADR = -0

End of task

Awaken job put to sleep in "IMUSTALL" and cause it to start at (IMUCADR), indicating an error

IMUCADR = +0

End of task

IMUSTALL

If IMUCADR > 0, proceed to "POODOO" (pattern 21210₈)

If IMUCADR = +0, set IMUCADR = (Calling address +1, in FCADR format), and put present job to sleep

If IMUCADR = -0:

IMUCADR = +0

Return (to calling address +1, indicating an error)

If IMUCADR = -1:

IMUCADR = +0

Return (to calling address +2, indicating a success)

If IMUCADR < -1, proceed to "POODOO" (pattern 21210₈)

Quantities in Computations

See also list of major variables and list of routines

- ldPIPADT: Cell containing time information used with IMU compensation routines. During free-flight portions, contains value of TIME1 when previous gyro drift compensation was made, scale factor B14, units centi-seconds (with optional tag "OLDBT1"). During those portions of flight when "l/PIPA" is entered, contains value of computing interval, scale factor B8, units centi-seconds. Both quantities are single precision. Cell is set to 2 seconds in "P11" and in "LASTBIAS", and to $\frac{1}{2}$ second in "GTSCPSS".
- ADRLGYR: Dummy quantity used to indicate that gyro compensation information taken from least significant 11 bits of IGYRO. Bits 11-9 are loaded into EBANK (see Data Input/Output) and bits 8-1 are added to 1400₈ to give address within the erasable memory bank.
- C_{ad} : Set of three single precision erasable memory constants, program notation "ADIAX", "ADIAY", and "ADIAZ", to correct for "IRIG acceleration sensitive drift along the input axis", scale factor B-3, units gyro pulses/accelerometer count. One gyro pulse is 2^{-21} revolution and one accelerometer count is a velocity increment of 5.85 centimeters/second. The units of the constant can also be expressed as "meru/g", where one meru is 10^{-3} x earth rate. In these units, the scaling of the constant is about 19.0304345 least increments per (meru/g): this figure is $2^{17} \times 0.024339048 \times (5.85 \times 10^{-2})/9.80665$, where first term is the reciprocal of the least increment for units of gyro pulses/accelerometer count; the second term is the number of gyro pulses/meru (0.1 K_{omegms}, see Prelaunch Alignment); the third is the accelerometer scale factor in units of meters/second; and the fourth is gravity (meters/sec²). If earth rate is simplified to 15 degrees/hour, the number of least increments per (meru/g) is reduced to about 18.9785.
- C_{nbd} : Set of three single precision erasable memory constants, program notation "NBDX", "NBDY", and "NBDZ", to correct for "IRIG bias drift", scale factor B-5, units gyro pulses/centi-second. The units of the constant can also be expressed as "meru", in which case the scaling is about 127.606708 least increments per meru: this figure is $2^{19} \times 0.01 \times 0.024339048$ (see C_{ad} : the 0.01 factor is for centi-seconds). For earth rate simplified to 15 degrees/hour, the number of least increments per meru is reduced to about 127.2583.
- $C_{pipabias}$: Set of three single precision erasable memory constants, program notation for first "PIPABIAS" (or "PBIASX", "PBIASY", and "PBIASZ"), to correct for "accelerometer bias", scale factor B-6, units accelerometer counts/centi-second. The units of the constant can also be expressed as "cm/sec²", in which case the scaling is about 1792.4376 least increments per cm/sec²: this figure is $2^{20} \times 0.01 \times (1/5.85)$, where first term is reciprocal of normal least increment, second compensates for centi-seconds,

and third is reciprocal of accelerometer scale factor. In-flight evaluation could be performed by determining the number of accelerometer counts accumulated in a period of 320 seconds, multiplying by two to give counts in $(2^6 \times 10)$ seconds, and then dividing by 1000 to give scaled counts/centi-second. N21 can be used to read PIPA cells and NO1 with decimal input can be used to load the appropriate cells (1452, 1454, and 1456 in ECADR form for x, y, z respectively).

C_{pipascf}: Set of three single precision erasable memory constants, program notation "PIPASCFX", "PIPASCFY", and "PIPASCFZ", scale factor B-9, units accelerometer counts/accelerometer count, used to correct for "accelerometer scale factor error." The units of the constant can also be expressed as "parts per million (PPM)", in which case the scaling is 8.388608 least increments per PPM: this figure is $2^{23} \times 10^{-6}$, where first term is reciprocal of normal least increment and second is PPM conversion.

C_{sr}: Set of three single precision erasable memory constants, program notation "ADSRAX", "ADSRAY", and "ADSRAZ", to correct for "IRIG acceleration sensitive drift along the spin reference axis", scale factor B-3, units gyro pulses/accelerometer count. See discussion with C_{ad} (which has same scaling).

CDUXCMD (CDUXCMD, CDUYCMD, CDUZCMD): Single precision value of computer special erasable memory cells 0050₈ - 0052₈. Output pulses are generated based on the contents of these cells if bits 15-13 of channel 14 (respectively) are set 1, and the respective IMU CDU error counter is loaded from the pulse train information if bit 6 of channel 12 is 1. Pulses may be used to coarse align the IMU if bit 4 of channel 12 is set (in this case, IMU stable member movement causes the error counter information to be decremented), with 2^{13} pulses giving one revolution. Pulses are generated at a rate of 3200 pps. See also Digital Autopilot Interface Routines.

COMMAND: Single precision value of required coarse-align command, scale factor B1, units revolutions. Could also be considered (see above discussion of CDUXCMD) to have a scale factor B14, units coarse-align pulses.

DAPDATR1: See Digital Autopilot Interface Routines.

GCOMP: Value of required gyro compensation command, computed with a scale factor B14 but used in "IMUPULSE" with a scale factor B21 (or, alternatively, with a scale factor B0 revolutions rather than B21 gyro pulses, since there are 2^{21} gyro pulses/revolution).

GCOMPSW: Single precision control cell used to bypass the performance of "1/PIPA" and "NBDONLY" if is negative. Set 0 in "GTSCPSS".

GYROCMD: Single precision value of computer special erasable memory cell 0047₈, used to control generation of gyro torquing pulses at a 3200 pps rate if bit 10 of channel 14 = 1. Gyro power supply for the pulses is enabled by bit 6 of channel 14, bits 8-7 indicate the axis, and bit 9 is 1 if a negative command is to be produced. One pulse is 2^{-21} revolution.

IMODES30: Single precision flag word used for control of "T4RUPT" routines associated with channel 30 inputs. Individual bits have the meanings given below. A restart ("GOPROG") sets bits 14-10 to 1 and sets bits 15, 8-6, and 2 to 0.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
15	TLIMBIT	Last sampled value of channel 30 bit 15 (0 if stable member within designed temperature limits).
14	TONISSBT	Last sampled value of channel 30 bit 14 (0 if ISS has been turned on or commanded to be turned on).
13	IMUFLBIT	Last sampled value of channel 30 bit 13 (0 if an IMU fail indication has been produced by the IMU hardware).
12	ICDUFLBT	Last sampled value of channel 30 bit 12 (0 if an IMU CDU fail indication has been generated by the IMU CDU hardware).
11	CAGEBIT	Last sampled value of channel 30 bit 11 (0 if an IMU cage command generated by the crew).
10	PIPAFLBT	Same as bit 13 of IMODES33 (0 if an accelerometer fail indication produced by the hardware, channel 33 bit 13).
9	IMUOPBIT	Last sampled value of channel 30 bit 9 (0 if IMU turned on and operating with no malfunctions).
8	ITNON2BT	Bit 7 of IMODES30 sensed as 1 (i.e. turn-on request received). Used to achieve a wait of 0.48 seconds (during which bit is 1) before acting on turn-on information.
7	ITNON1BT	IMU turn-on request received (a change in value to a logic 1, i.e. binary 0, of bit 14 or bit 9 of IMODES30).
6	IMUNITBT	Bit set 1 to indicate that IMU use should not be attempted: set 1 during the turn-on sequence, if bit 11 of IMODES30 indicates a cage command, or ISS zeroing in "T4RUPT" loop.
5	NOACCALM	Bit set 1 to inhibit generation of program alarm 212 ₈ in "C33TEST" (for accelerometer fail signal).
4	IMUFINHT	Bit set 1 to inhibit generation of ISS Warning signal based on IMU fail signal (bit 13 of IMODES30).

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
3	ICDUINHHT	Bit set 1 to inhibit generation of ISS Warning signal based on receipt of IMU CDU fail signal (bit 12 of IMODES30).
2	DLAYFAIL	Bit set 1 to indicate failure of the turn-on delay sequence (IMU turn-on signals not present for the required time interval).
1	ACCFINHT	Bit set 1 to inhibit generation of ISS Warning signal based on receipt of accelerometer fail signal (bit 10 of IMODES30).

IMODES33: Single precision flag word used for control of "T4RUPT" routines associated with channel 33 inputs (and other functions). Individual bits have the following meanings.

<u>Bit</u>	<u>Symbol</u>	<u>Meaning</u>
14	PROCDBIT	Last sampled value of channel 32 bit 14 (0 if a Proceed pushbutton command given). This bit is used in "PROCEEDE".
13	PIP2FLBT	Last sampled value of channel 33 bit 13 (0 if an accelerometer fail indication has been produced by the hardware).
12	DNLKFAIL	Last sampled value of channel 33 bit 12 (0 if telemetry end pulse rejected, downlink too fast).
11	UPLKFAIL	Last sampled value of channel 33 bit 11 (0 if uplink pulse rejected, uplink too fast).
6	NOIMUDAP	Bit set 1 to indicate that IMU use for vehicle attitude information should not be attempted (sensed by RCS/docked DAP and Entry DAP).
5	IMUZROBT	Bit set 1 in "IMUZERO" and reset 0 about 8.22 seconds later in "IMUZERO2". Indicates that IMU zeroing outside of T4RUPT package is taking place.
1	LMPTSTBT	Bit set 1 if a lamp test (verb 35) is in progress, to inhibit turn-off of appropriate indicators.

IMUCADR: Cell containing control information used in association with "IMUSTALL": it contains return address information from the routine or information on the success (if = -1) or failure (if = -0) of an IMU function.

$K_{2\text{secp}}$: Single precision constant, program notation "PRIO31", octal value 31000₈, scale factor B8, units centi-seconds. Value corresponds to 2 seconds.

$K_{\text{bt}10}$: Single precision constant, program notation "BIT10", scale factor B0, units centi-seconds/gyro torquing pulse. Value is 2^{-5} (i.e. 1/32), since output pulse rate is 3200 pps, or 32 pulses/centi-second.

K_{corst} : Single precision constant, program notation "COARSTOL", scale factor B-1, units revolutions. Octal value is 77511₈, which corresponds (after a one-bit modification used in the program for convenience in forming the absolute value) to about -2.0105° .

K_{cpck} : Single precision constant, program notation "NEGONE", scale factor B14, units gyro pulses, used to check if compensation of gyros is required. Octal value is 77776₈, but used in the program in such a way (a mask on decremented |GCOMP_i|) that effective value corresponds to 3 pulses: the most significant half of GCOMP_i must be 3 or more pulses for the "1/GYRO" computations to be initiated, although lack of sign agreement could make the actual value only slightly more than 2 pulses.

K_{gyrfrc} : Constant, program notation "GYROFRAC", scale factor B21, units gyro pulses. Stored value is 28×2^{-28} , corresponding to about 0.219 pulses (nominal value 0.215).

K_{gyrsc} : Dummy constant indicating a rescaling of original contents of E_{ADRLGYR} from B21 to B28 pulses. "Value" is 1.0, scale factor B7.

$K_{\text{m}15\text{degs}}$: Single precision constant, program notation "-15DEGS", scale factor B-1, units revolutions. Octal value is 75252₈, corresponding to about -14.996° .

$K_{\text{m}70\text{degs}}$: Single precision constant, program notation "-70DEGS", scale factor B-1, units revolutions. Octal value is 63434₈, corresponding to about -70.005° (after a one-bit modification used in the program for convenience in forming the absolute value).

K_{mcmxm} : Single precision constant, program notation "-COMMAX-", scale factor B14, units CDU output pulses. Value is -192×2^{-14} .

K_{mcommax} : Single precision constant, program notation "-COMMAX", scale factor B14, units CDU output pulses. Value is -191×2^{-14} , but used in program (for convenience in forming the absolute value) in such a way that effective value is -192. Limits change in CDU angle to 192 least increments per axis (or 8.4375°) each 0.6 seconds. The maximum pulses required (to go 180 degrees) would be 4096, which would require "21.3333", i.e. 22 intervals of 0.6 seconds each. With the additional 1.5 seconds for the K_{corst} check, time required for coarse align would be less than about $0.06 + 0.02 + 22(0.6) + 1.5$, or 14.78 seconds.

K_{mgyrmm} : Single precision constant, program notation "-GYROMIN", scale factor B7, units gyro pulses. Octal value is 77601₈, which corresponds (after a one-bit adjustment used in program for convenience in forming the absolute value) to (-127/128), i.e. one least increment less than 1.00 gyro pulse.

LGYRO: Cell containing (if non-zero) address information for gyro torquing commands. If cell is zero, gyros are "available" (i.e. not being driven).

PIPA: Value of special erasable memory cells (0037₈ - 0041₈) containing accumulated output counts from the accelerometers, scale factor B4, units counts. One count is 5.85 centimeters/second.

T_{pptom1} : Value of computer clock when accelerometers sampled, program notation "PIPTIME1", scale factor B28, units centi-seconds. Subsequently loaded into T_{pptom} , hence can also be used as storage for the time tag associated with R_{n1} and V_{n1} (see General Program Control) for use by "PREREAD".

Inflight Alignment

P51 (Tag also "P53")

If bit 9(IMUOPBIT) of IMODES30 = 1: (IMU not on)

Perform "ALARM" (pattern 0210_g)

Proceed to "GOTOPOOH"

Proceed to "P51A"

P51A

Set bit 8(IMUSE) of FLAGWRDO = 1

TS = 00015_g

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed to "P51B"
otherwise, proceed

THETAD = 0

TS = 0622_{vn}

Perform "GODSPRET"

TS = 4100_{vn}

Perform "GODSPRET"

If IMUCADR \neq 0:

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck IMUCADR)

Perform "IMUCOARS"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Perform "IMUFINE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Proceed to second line of "P51A"

P51B

STARIND = 0

ldPIPADT = TIME1

GCOMP = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

If MODREG = 53: (Tag here "P51C", from below)

Perform "R56"

If MODREG \neq 53:

Perform "R53"

Perform "SXTSM"

If STARIND = 0:

STARSAV1 = TS

TS = T_{sight}

Perform "PLANET"

PLANVEC = TS

STARIND = 1

Proceed to 5th line of "P51B"

STARSAV2 = TS

TS = T_{sight} (Tag here "P51F")

Perform "PLANET"

STARBC = TS

STARAC = PLANVEC

STARAD = STARSAV1

STARBD = STARSAV2

Perform "R54"

If bit 3(FREEFLAG) of FLAGWRD0 = 0: (e.g. a V32E response in "R54")

Proceed to second line of "P51A"

Perform "AXISGEN"

Set bit 13(REFSMFLG) of FLAGWRD3 = 0

$$[\text{REFSMMAT}] = [X_{dc}]$$

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

Proceed to "GOTOPOOH"

PROG52 (Tag also "P54")

Set bits 7(UPDATFLG) and 5(TRACKFLG) of FLAGWRD1 = 0

Perform "R02BOTH"

If bit 7(AUTOSEQ) of FLAGWRD10 = 1:

Proceed to "P52AUTO" (page MINK-7)

If bit 9(UTFLAG) of FLAGWRD8 = 1:

If OPTNTYPE \neq 0: (means "option 2 of P20", i.e. R67 rotation)

Set bit 5(TRACKFLG) of FLAGWRD1 = 1 (should not have R67 active in P54 because of delay it can introduce in "R56")

If bit 4(PFRATFLG) of FLAGWRD2 = 0:

OPTION2 = 3

If bit 4(PFRATFLG) of FLAGWRD2 = 1:

OPTION2 = 1

Proceed to "P52B"

P52B

TS = 1

Proceed to "GOPERF4": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to previous line

TS = (bits 2-1 of OPTION2)

If TS = 3: ([REFSMAT] orientation)
 Proceed to "P52C"

If TS = 1: ("preferred" orientation)
 Proceed to "P52D"

If TS = 0 or 2, proceed ("nominal" orientation)
 $DSPT\text{EML}_{dp} = (-0, 0)$
 $TS = 0634_{vn}$
 Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
 if proceed, proceed
 otherwise, proceed to previous line

If $DSPT\text{EML}_{sp} = 0$:
 $DSPT\text{EML}_{dp} = T_{now}$
 $T_{decl} = DSPT\text{EML}_{dp}$ (Tag here "S52.3")
 Perform "CSMPREC"

$Z_{smd} = - \text{unit}R_{att}$
 $Y_{smd} = \text{unit}(V_{att} * R_{att})$
 $X_{smd} = \text{unit}(Y_{smd} * Z_{smd})$
 Proceed to "P52D"

P52D

Perform "S52.2"

$TS = 0622_{vn}$
 Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
 if proceed, proceed
 otherwise, proceed to "P52D"

If bit 7(AUTOSEQ) of FLGWRD10 = 1:
 Proceed to "PERF20" (page MINK-7)

$TS = 00013_g$
 Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
 if proceed, proceed
 otherwise, proceed to "GYCRS"

Proceed to "CAL53A" (exits to "P52C")

P52C

TS = 00015₈

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "R51"

$T_{\text{sight}} = T_{\text{now}} + K_{\text{tsgtl}}$

Perform "S50"

Perform "CDUTRIG" (This is start of the former "PICAPAR"
routine: the tag has been deleted)

Perform "CALCSMSC"

Set bit 10(VFLAG) of FLAGWRD3 = 1 (means no star found)

BESTI = 0

BESTJ = 0

$SAX = \text{unit} \left((K_{\sin 33} X_{\text{dc}} + K_{\cos 33} Z_{\text{dc}}) [\text{REFSMAT}] \right)$

X1 = 228

Proceed to "PIC1"

PIC1

If $X1 \leq 6$, proceed to "PICEND"

$X1 = X1 - 6$

$TS = K_{\text{cat}_{X1}}$

Perform "OCCULT"

If bit 7(CULTFLAG) of FLAGWRD3 = 1, proceed to "PIC1"

$X2 = X1$

Proceed to "PIC3"

PIC3

If $X2 \leq 6$, proceed to "PIC1"

$X2 = X2 - 6$

$TS = K_{\text{cat}_{X2}}$

Perform "OCCULT"

If bit 7(CULTFLAG) of FLAGWRD3 = 1, proceed to "PIC3"

$$TS = \frac{K_{cat_{X1}}}{K_{cat_{X2}}} \cdot K_{cat_{X2}} - K_{css66}$$

If $TS < 0$, proceed to "PIC3" (separation more than 76°)

If $TS + K_{css6640} \geq 0$, proceed to "PIC3" (separation 30° or less)

If $\frac{K_{cat_{X1}}}{K_{cat_{X2}}} \cdot SAX - K_{css33} < 0$, proceed to "PIC1"

If $\frac{K_{cat_{X2}}}{K_{cat_{X1}}} \cdot SAX - K_{css33} < 0$, proceed to "PIC3"

If bit 10(VFLAG) of FLAGWRD3 = 1: (means no star yet found)

Set bit 10(VFLAG) of FLAGWRD3 = 0

BESTI = X1

BESTJ = X2

Proceed to "PIC3"

$$TS_1 = \frac{K_{cat_{BESTI}}}{K_{cat_{BESTJ}}} \cdot K_{cat_{BESTJ}}$$

Set bit 10(VFLAG) of FLAGWRD3 = 1

$$TS_2 = \frac{K_{cat_{X1}}}{K_{cat_{X2}}} \cdot K_{cat_{X2}}$$

Set bit 10(VFLAG) of FLAGWRD3 = 0

If $TS_2 - TS_1 \geq 0$, proceed to "PIC3" (new pair closer than old pair)

BESTI = X1

BESTJ = X2

Proceed to "PIC3"

OCCULT

If $CSUN - (TS \cdot VSUN) < 0$:

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

If $CEARTH - (TS \cdot VEARTH) < 0$:

Set bit 7(CULTFLAG) of FLAGWRD3 = 1

Return

Set bit 7(CULTFLAG) of FLAGWRD3 = 0

Return

PICEND Entered from "PIC1" at the end of the star table scan

If bit 10(VFLAG) of FLAGWRD3 = 1: (means no stars found)

Perform "ALARM" (pattern 0405_g)

TS = 0509_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "P52C"

Proceed to "R51"

S50 (Tag also "LOCSAM")

TS = T_{sight}

Perform "LSPOS"

VSUN = TS₁ (unit sun position, B1)

T_{decl} = T_{sight}

Perform "CSMCONIC"

VEARTH = - unit_{R_{att}}

CEARTH = $\cos \left(\sin^{-1} \left(K_{\text{rsube}} / |R_{\text{att}}| \right) \right) + K_{5\text{degs}}$

VELdc = K_{ldc} V_{att} + VSUN * K_{eclipol}

CSUN = K_{cssun}

Return

S52.2

Perform "CDUTRIG"

Perform "CALCSMSC"

$\underline{X}_{dc} = \text{unit} \left(\underline{X}_{dc} \left[\text{REFSMMAT} \right] \right)$

$\underline{Y}_{dc} = \text{unit} \left(\underline{Y}_{dc} \left[\text{REFSMMAT} \right] \right)$

$\underline{Z}_{dc} = \text{unit} \left(\underline{Z}_{dc} \left[\text{REFSMMAT} \right] \right)$

$\left[\underline{X}_{sm} \right] = \left[\underline{X}_{smd} \right]$

Perform "CALCGA"

Return

PLANET

$T_{\text{sight}} = TS$

Perform "S50"

$TS = 6$ (bits 6-1 of STARCODE)

If STARIND = 0:

BESTI = TS

If STARIND = 1:

BESTJ = TS

If TS = 0: (means planet)

$TS = 0688_{\text{vn}}$

Proceed to "VNFLASH": (if terminate, proceed to "GOTOPOOH")
if proceed, proceed
otherwise, proceed to previous line

$TS = \text{unit} \left(\text{unit}(K_{\text{ldsqr}3} \text{STARSAV}3) + \text{VELdC} \right)$

Return

If $(TS - 228) < 0$: (original code in range 01-45_g, meaning
a star in star table)

$TS = \text{unit}(K_{\text{cat}_{TS}} + \text{VELdC})$

Return

If TS = 228: (original code 46_g, meaning sun)

$TS = \text{unit}(\text{VSUN} + \text{VELdC})$

Return

If TS = 234: (original code 47_g, meaning earth)

$TS = \text{unit}(\text{VEARTH} + \text{VELdC})$

Return

(If $TS > 234$, improper results)

SXTSM

Set bit 11(ATMFLAG) of FLAGWRDO = 1

$T_{\text{sight}} = \text{MRKBUF1}_{\text{dp}}$

$X1 = - \text{"MRKBUF1"}$

If STARIND = 0: (Tag here "SXTSM1", entered from "GETUM")

$\text{MARKDOWN}+i = E_{i-X1}$ (i = 0 - 6)

If STARIND = 1:

$\text{MARK2DWN}+i = E_{i-X1}$ (i = 0 - 6)

$\text{CDUSPOT} = E_{2-X1}$

If bit 11(ATMFLAG) of FLAGWRDO = 0: (e.g. "GETUM" entrance)

Perform "SXTNB"

Perform "TRG*NBSM" (leaves line-of-sight in TS)

Return

Set bit 11(ATMFLAG) of FLAGWRDO = 0

If bits 14-7 of STARCODE = 0: (e.g. 000xx₈)

Perform "SXTNB"

Perform "TRG*NBSM"

Return

If bits 14-8 of STARCODE = 0: (e.g. 001xx₈)

$\text{TS} = \text{NBOA}_6$ (Tag here "ATMSS")

Perform "TRG*NBSM"

Return

$\text{NBOA}_0 = \text{unit}(\text{NBOA}_3 * \text{NBOA}_6)$ (STARCODE equal to e.g. 002xx₈)

Perform "OASTAR"

$\text{TS} = \text{unit}(\text{TS} [\text{NBOA}])$

Perform "TRG*NBSM"

Return

CAL53A (This is routine R50, entered from "P52D")

Perform "S52.2"

TS = CDU

If any $(|TS_i - THETAD_i|) \gg K_{deg1}$ and $< K_{deg359}$: (i = x,y,z; check actually done using double precision)

If IMUCADR \neq 0:

Delay 1 second (by putting job to sleep via "DELAYJOB")

Proceed to second previous line (recheck IMUCADR)

Perform "IMUCOARS"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed otherwise, proceed

Perform "IMUFINE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed otherwise, proceed

CDUSPOT = (THETAD_y, THETAD_z, THETAD_x) (Tag here "FINEONLY")

TS = unit_X

Perform "TRG*NBSM"

STARAD = TS

TS = unit_Y

Perform "TRG*NBSM" starting at second line

STARBD = TS

Perform "CDUTRIG"

Perform "CALCSMSC"

STARAC = $\frac{X}{dc}$

STARBC = $\frac{Y}{dc}$

Perform "AXISGEN"

Perform "CALCGTA"

Set bit 13(REFSMFLG) of FLAGWRD3 = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 0

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

ldPIPADT = TIME1

GCOMP = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

[REFSMMAT] = [X_{smd}]

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

Proceed to "P52C"

R51 Entered from "PICEND" and "P52C"

STARIND = 1

If STARIND = 0:

STARCODE = BESTI / 6

If STARIND = 1:

STARCODE = BESTJ / 6

Proceed to "R51DSP"

R51DSP

TS = 0170_{vn}

Perform "VNFLASHR": (if terminate, proceed to "GOTOPOOH")
if proceed, skip next 2 lines
otherwise, proceed to previous line

TS = 110₂ and perform "BLANKET" (R3BLNK, R2BLNK)

End of job

If STARCODE ≤ -0 or if (bits 6-1 of STARCODE) > 47₈:

Set bit 7(Operator Error) of channel 11 = 1

Proceed to "R51DSP"

If bit 7 of STARCODE = 1:

If STARCODE ≠ 00146₈: (i.e. sun not specified)

Set bit 7(Operator Error) of channel 11 = 1

Proceed to "R51DSP"

```

TS = Tnow
Perform "PLANET"

If STARIND > 0:
    STARSAV1 = TS

If STARIND = 0:
    STARSAV2 = TS

If (MODREG modulo 4) ≠ 0:      (i.e. P54 rather than P52)
    Perform "R56"      (processing of mark ENTR could be delayed if
                       R67 is active)

If (MODREG modulo 4) = 0:
    If bits 14-7 of STARCODE = 0:      (Tag here "R51A1"; value
                                         e.g. 000xx8)
        Perform "R52"
    If bits 14-7 of STARCODE ≠ 0:
        Perform "R53"

Perform "SXTSM"      (Tag here "R51B")

STARSAV2 = TS

TS = Tsight
Perform "PLANET"

If STARIND > 0:
    PLANVEC = unit( [REFSMMAT] TS )
    STARSAV1 = STARSAV2
    STARIND = 0

    Proceed to 2nd line of "R51"

STARAD = unit( [REFSMMAT] TS )
STARAC = STARSAV2
STARBC = STARSAV1
STARBD = PLANVEC

```

Perform "R54"

If bit 3(FREEFLAG) of FLAGWRDO = 0:

Proceed to "R51K"

Perform "AXISGEN"

Proceed to "R55" (exits to "R51K")

R51K

TS = 00014_g (Tag here also "R51KA")

Proceed to "GOPERF1": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "GOTOPOOH"

Proceed to "P52C"

R54 (Tag also "CHKSDATA"; entered from "P51B" and "R51DSP")

Set bit 3(FREEFLAG) of FLAGWRDO = 1

$TS_1 = \cos^{-1} (\text{STARAD} \cdot \text{STARBD})$

Set bit 3(FREEFLAG) of FLAGWRDO = 0

$TS_2 = \cos^{-1} (\text{STARAC} \cdot \text{STARBC})$

DSPTM1 = $\left| TS_2 - TS_1 \right|$, with sign agreement forced, scaled B-1 rev.

DSPTM1+1 = TS_1 (single precision, B-1 rev.)

Set bit 3(FREEFLAG) of FLAGWRDO = 1

TS = 0

Perform "CLEANDSP"

TS = 0605_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, skip next line
otherwise, proceed

Set bit 3(FREEFLAG) of FLAGWRDO = 0 (e.g. a V32E response)

Return

R55 Entered from "R51DSP"

Perform "CALCGTA"

TS = 0693_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "R55RET"

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

Proceed to "R55RET"

R55RET

Set bit 4(PFRATFLG) of FLAGWRD2 = 0

Proceed to "R51K"

GYCRS Entered from "PERF20" and "P52D"

$X_{dc} = \text{unit}([REFSMMAT] X_{smd})$

$Y_{dc} = \text{unit}([REFSMMAT] Y_{smd})$

$Z_{dc} = \text{unit}([REFSMMAT] Z_{smd})$

Perform "CALCGTA"

Set bit 15(DRIFTFLG) of FLAGWRD2 = 0

Set bit 13(REFSMFLG) of FLAGWRD3 = 0

TS = 1620_{vn}

Perform "GODSPR"

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL": if error return, perform "217ALARM"; proceed
otherwise, proceed

$[REFSMMAT] = [X_{smd}]$

Set bit 4(PFRATFLG) of FLAGWRD2 = 0

Set bit 13(REFSMFLG) of FLAGWRD3 = 1

ldPIPADT = TIME1

GCOMP = 0

Set bit 15(DRIFTFLG) of FLAGWRD2 = 1

If bit 7(AUTOSEQ) of FLGWRD10 = 1:

Set bit 1(PCFLAG) of FLGWRD10 = 0 (means did pulse torquing)

Proceed to second line of "GOTOPOOH"

Proceed to "R51K"

217ALARM

Inhibit interrupts

ALMCADR = "Calling address +1" (S-register portion)

Perform "ALARM2" (pattern 0217_g)

Return (address in ALMCADR)

P50

OPTION2 = 1

TS = 00012_g

Proceed to "GOPERF4": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to 2nd line of "P50"

TS = (bits 2-1 of OPTION2)

If TS = 0:

Set bit 7(Operator error) of channel 11 = 1

Proceed to 2nd line of "P50"

If TS = 1:

Set bit 12(P50.1FLG) of FLAGWRD0 = 1 (Tag here "P50.1")
(bit set 0 in "INITSUB")

If TS = 3:

$NBOA_0 = \text{unit}(NBOA_3 * NBOA_6)$ (Tag here "P50.3")

$[X_{sm}] = [NBOA]$

Proceed to "OAVECS"

Perform "R02BOTH" (Tag here "P50.2", for options 1 and 2)

STARIND = 0

STARCODE = 46₈ (sun)

Perform "R53"

CDUSPOT = MRKBUF1+2

T_{sight} = MRKBUF1_{dp}

TS = T_{sight}

Perform "PLANET"

TS = unit([REFSMMAT] TS)

Perform "CDUTRIG" starting at 2nd line

Perform "*SMNB*"

STARBC = TS

TS = unit([REFSMMAT] VSUN)

Perform "*SMNB*"

STARAC = TS

If bit 12(P50.1FLG) of FLAGWRDO = 1:

STARBD = unitY (Tag here "P50.1A")

STARBC = NBOA₃

If bit 12(P50.1FLG) of FLAGWRDO = 0:

Perform "OASTAR"

STARBD = TS

STARAD = unitZ (Tag here "P50GEN")

Perform "AXISGEN"

$$\begin{bmatrix} X_{sm} \end{bmatrix} = \begin{bmatrix} X_{dc} \end{bmatrix}$$

Proceed to "OAVECS"

OAVECS

Set bit 10(P50FLAG) of FLAGWRDO = 1

X_{dc} = unitX

$\underline{Y}_{dc} = \text{unit}\underline{Y}$

$\underline{Z}_{dc} = \text{unit}\underline{Z}$

Perform "CALCGA"

$\underline{N23TEMP} = \underline{THETAD}$

$\underline{THETAD}_x = 40000_8 - \underline{THETAD}_x$ (40000₈ is 180°, ones complement difference formed)

$\underline{TS} = 0623_{vn}$

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to previous line

$\underline{THETAD}_x = 40000_8 - \underline{THETAD}_x$ (Tag here "DOVECS")

If ($\underline{THETAD} - \underline{N23TEMP}$) has at least one component non-zero:

$\underline{CDUSPOT} = (\underline{THETAD}_y, \underline{THETAD}_z, \underline{THETAD}_x)$

Perform "CDUTRIG" starting at 2nd line

Perform "CALCSMSC"

$\underline{Y}_{sm} = \text{unit}(\underline{X}_{dc}) \text{unit}\underline{Y}$

$\underline{Z}_{sm} = \text{unit}(\underline{X}_{dc}) \text{unit}\underline{Z}$

$[\underline{NBOA}] = [\underline{X}_{sm}]$ (Tag here "P50OUT")

Proceed to "GOTOPOOH"

OASTAR

(MRKBUF1+3 and MRKBUF1+5 both B-1 rev when enter)

$\underline{TS}_1 = \text{MRKBUF1+5}$, converted to ones complement double precision(elev.)

$\underline{TS}_2 = \text{MRKBUF1+3}$, converted to ones complement double precision(az.)

$\underline{TS} = \text{unit}(\cos \underline{TS}_1 \sin \underline{TS}_2, -\cos \underline{TS}_1 \cos \underline{TS}_2, -\sin \underline{TS}_1)$

Return

P55

OPTION2 = 1

$\underline{TS} = 00013_8$

Proceed to "GOPERF4": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to 2nd line of "P55"

TS = (bits 2-1 of OPTION2)

If TS = 0 or 3:

Set bit 7(Operator error) of channel 11 = 1

Proceed to 2nd line of "P55"

If TS = 1:

Perform "RO2BOTH" (Tag here "P55.1")

Set bit 13(P55.1FLG) of FLAGWRDO = 1 (bit set 0 in "INITSUB")

Proceed to "TRKSTAR"

STARIND = 0 (option 2, tag "P55.2")

STARCODE = 46_8 (sun)

Perform "R53"

TS = MRKBUF1_{dp}

Perform "PLANET"

STARBC = TS

STARAC = VSUN

NBOA₀ = unit(NBOA₃ * NBOA₆)

X1 = - "MRKBUF1"

Perform "SXTNB"

STARBD = unit([NBOA] TS)

STARAD = unitZ

Perform "AXISGEN"

Proceed to "TRKSTAR"

TRKSTAR

TS = 0170_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to previous line

If STARCODE ≤ -0 or if STARCODE $> 47_8$:

Set bit 7(Operator error) of channel 11 = 1

Proceed to "TRKSTAR"

TS = T_{now}

Perform "PLANET"

If bit 13(P55.1FLG) of FLAGWRDO = 0:

STARSAV₁ = unit([X_{dc}] TS)

If bit 13(P55.1FLG) of FLAGWRDO = 1:

STARSAV₁ = unit([REFSMMAT] TS) (Tag here "P55.1A")

NBOA₀ = unit(NBOA₃ * NBOA₆)

TS = STARSAV₁

Perform "CDUTRIG"

Perform "*SMNB*"

STARSAV₁ = unit([NBOA] TS)

STARSAV₂ = unit(STARSAV_{1x}, STARSAV_{1y}, 0) (Tag here "P55GIMS")

COSTH = - unit_Y · STARSAV₂

SINTH = STARSAV₂ · unit_X

Perform "ARCTRIG"

TRKAZ = THETA

COSTH = STARSAV₂ · STARSAV₁

SINTH = - unit_Z · STARSAV₁

Perform "ARCTRIG"

TRKEL = THETA

If |TRKAZ_{sp}| + K_{m87deg} ≤ 0 :

If |TRKEL_{sp}| + K_{m40deg} ≤ 0 :

Skip next line

Perform "ALARM" (pattern 0107₈)

TS = 0614_{vn}

(Tag here "N14DISP")

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "TRKSTAR"

TS = K_{d21600} | TRKAZ_{sp} | (truncated integer minutes, form ABCDE_g)

TRKAZOCT = 00ABC_g (N19 cells also used for intermediate
computation results before display)

TRKAZOCT+1 = 00DE0_g

If TRKAZ_{sp} < 0:

TRKAZOCT = TRKAZOCT + 01000_g

TS = 0419_{vn}

Proceed to "GOFLASH": if terminate, proceed to "GOTOPOOH"
if proceed, proceed
otherwise, proceed to "TRKSTAR"

Proceed to "GOTOPOOH"

Quantities in Computations

See also list of major variables and list of routines

ldPIPADT: See IMU Computations.

ALMCADR: See General Program Control.

BESTI: Single precision value of index parameter for star #1 of the "best" star pair (as determined by "P52C" computations), scale factor B14. It will be zero if no star pairs found that are satisfactory, and otherwise will be equal to six times the "star number" (see K_{cat}) when loaded in "P52C" computations. See BESTJ. For option 2 of P50, is loaded in "R53C" after N71 response (since STARIND = 0); options 1 and 2 also load it in "PLANET". For option 2 of P55, loading parallels that done for P50; both options also load it (via "PLANET") after response to the N70 display of "TRKSTAR".

BESTJ: Single precision value of index parameter for star #2 of the "best" star pair (as determined by "P52C" computations), scale factor B14. It will be zero if no star pairs found that are satisfactory, and otherwise will be equal to six times the "star number" (see K_{cat}) when loaded in "P52C" computations. Unless it is zero, it will be less than BESTI as loaded in "P52C" computations. In P52/P54, it is loaded with (STARCODE x 6), see "PLANET", for the first body that is used, while BESTI has the second; in P51/P53, however, it contains (STARCODE x 6) for the second body, while BESTI has the first. In both cases, however, BESTJ will contain the body code associated with the sighting data for which the address selected in "SXTSM" is MARK2DWN. The address of BESTJ is one greater than the address of BESTI.

CDUSPOT: See Coordinate Transformations.

CEARTH: Value of the cosine of the minimum angle between the star and VEARTH allowed in "P52C" logic (i.e. in "OCCULT"), scale factor B2, computed in "S50". It is stored in push-down list location 16D.

COSTH: See Coordinate Transformations.

CSUN: Value of the cosine of the minimum angle between the star and VSUN allowed in "OCCULT", scale factor B2, set to K_{cssun} in "S50". It is a "variable" for indexing convenience in "OCCULT", and is stored in push-down list location 14D.

GCOMP: See IMU Computations.

IMODES30, IMUCADR: See IMU Computations.

K_{ldc} : Constant, program notation "1/C", scale factor B-6, value $4.2696E-5 \times 2^{-1}$. Value corresponds to $2^7 / (9.835712E8 \times 10^{-2} \times 0.3048)$, where first term (numerator) combines the B-6 scale factor and the 2^{-1} for constant, second is velocity of light (fps), third converts to centi-seconds, and 4th converts to meters, giving the $4.2696E-5$.

- $K_{\text{ldsqr}3}$: Constant, program notation "1/SQR3", scale factor B0, value 0.57735021. Value corresponds approximately to $(1/1.732051)$, or the reciprocal of $\sqrt{3}$. Used in "PLANET" to ensure that the formation of the unit vector does not overflow.
- $K_{5\text{degs}}$: Constant, program notation "5DEGREES", scale factor B0, units revolutions. Value is 0.013888889, corresponding to 5 degrees.
- K_{cat} : Table of positions of 37 stars, program notation "CATLOG", stored as the X, Y, and Z components of a unit vector, scale factor B1. The index for the table equals six times the star serial number (since each component is double precision, six cells per star are used). See table below. Because of the nature of the interpretive index order (subtractive), the tag "CATLOG" actually is one cell after the least significant half of the z component of star #1, and #1 is the star with highest memory address.
- $K_{\text{cos}33}$: Constant, program notation "COS33", scale factor B0, value 0.8431756920. Value corresponds to cosine of $32^{\circ} 31' 23.19''$, the same angle as that used with [NB1NB2] (see Coordinate Transformations).
- $K_{\text{css}33}$: Constant, program notation "CSS33", scale factor B2, value 0.197002688. True value is 0.788010752, corresponding to cosine 38° (one half the angle reflected in $K_{\text{css}66}$, hence the notation).
- $K_{\text{css}66}$: Constant, program notation "CSS66", scale factor B2, value 0.060480472. True value is 0.241921888, corresponding to $\cos 76^{\circ}$ (angle gate was formerly 66° , hence the notation).
- $K_{\text{css}6640}$: Constant, program notation "CSS6640", scale factor B2, value -0.15602587. True value is -0.62410348, corresponding to $0.241921888 - 0.866025368$, or $(\cos 76^{\circ} - \cos 30^{\circ})$. Angle gates were formerly 66° and 40° , hence the notation.
- K_{cssun} : Constant, program notation "CSSUN", scale factor B2, value 0.24148. True value is 0.96592, corresponding to $\cos 15^{\circ}$.
- $K_{\text{d}21600}$: Single precision constant, program notation "D21600", scale factor B17, units of arc minutes per revolution. Value is 21600×2^{-17} , corresponding to $360 \times 60 \times 2^{-17}$, where first term is number of degrees in a revolution and second is number of minutes in a degree, while third is scale factor.
- K_{degl} : Constant, program notation "DEGREE1", scale factor B0, units revolutions. Octal value is $00056_8 37722_8$, corresponding to about 1.033° .
- $K_{\text{deg}359}$: Constant, program notation "DEG359", scale factor B0, units revolutions. Octal value is $37722_8 00004_8$, corresponding to about 358.9893° . The single precision versions of this constant and K_{degl} sum to one revolution.

- K_{eclipol} : Vector constant, program notation "ECLIPOL", scale factor B_0 , value (0, -0.395319722E-4, 0.911652662E-4). Values correspond approximately to 0.993674E-4 (0, -0.3978364, 0.9174564). The first term is the radian equivalent of about 20.496 arc seconds (the constant of aberration), while the terms in parentheses are - sin and cos of the obliquity.
- K_{m40deg} : Single precision constant, program notation "-40DEG", scale factor B_0 , units revolutions. Nominal decimal value is -0.111111, corresponding to $-40/360$, or -40° . Octal value is 74343₈, corresponding (after a one-bit change for absolute value convenience) to about -40.012° .
- K_{m87deg} : Single precision constant, program notation "-87DEG", scale factor B_0 , units revolutions. Nominal decimal value is -0.241666, corresponding to $-87/360$, or -87° . Octal value is 70210₈, corresponding (after a one-bit change for absolute value convenience) to about -87.012° .
- K_{rsube} : Constant, program notation "RSUBE", scale factor B_{29} , units meters. Value is 6378166×2^{-29} , the Fischer equatorial radius.
- K_{sin33} : Constant, program notation "SIN33", scale factor B_0 , value 0.5376381241. Value is the sine of the angle discussed for K_{cos33} .
- K_{tsgtl} : Constant, program notation "TSIGHT1", scale factor B_{28} , units centi-seconds. Value is 24000×2^{-28} , corresponding to 240 seconds.
- MARK2DWN+i (i = 0-6): Set of optics mark information (double precision time, CDU_y, optics shaft, CDU_z, optics trunnion, and CDU_x respectively) loaded in "SXTSM" for downlink purposes from alignment mark (P51-P54) identified by BESTJ. Shaft and trunnion have N14 information if used.
- MARKDOWN+i (i = 0-6): See Measurement Incorporation. Note that loading is done from MRKBUF1 set of cells (see Optics Computations for special "trunnion" scaling after N14 display).
- MRKBUF1: See Optics Computations.
- N23TEMP: Single precision vector storage for THETAD in "OAVECS", same scaling, used to determine if a manual change was made at the N23 display.
- NBOA₀: First row of [NBOA] matrix, scale factor B_1 , derived as cross-product of second and third rows (which are not time shared) of the matrix.
- NBOA₃: Second row of [NBOA] matrix, program notation "NBOA +6", scale factor B_1 , initialized as part of padload and subsequently updated if necessary. It gives the +Y-axis direction of the ATM Sensor Coordinate System in navigation-base coordinates.
- NBOA₆: Third row of [NBOA] matrix, program notation "NBOA +12", scale factor B_1 , initialized as part of padload and subsequently updated if necessary. It gives the +Z-axis direction of the ATM Sensor Coordinate System in navigation-base coordinates.

OGC: See Coordinate Transformations.

OPTION2: See Display Interface Routines.

OPTNTYPE: See Orbital and Rendezvous Navigation.

PLANVEC: Temporary storage for the output of the "PLANET" routine for the first body that is marked, scale factor B1. In P51/P53 it is in reference coordinates, while in P52/P54 it is converted to IMU coordinates before being stored.

SAX: Unit vector, scale factor B1, giving the direction of the optics shaft axis: stars are considered visible (for selection purposes) if they lie within 38° of this axis. The quantity is expressed in the reference coordinate system.

SINTH: See Coordinate Transformations.

STARAC, STARAD, STARBC, STARBD: See Coordinate Transformations.

STARCODE: Single precision cell, scale factor B14 (but considered to be octal), displayed in R1 of N70 and N71 and used to contain the serial number of the celestial body being sighted (bits 6-1): 00 for "planet"; 01-45₈ for star in K_{cat} ; 46₈ for sun; and 47₈ for earth. BESTI and BESTJ have six times this number (for double precision vector indexing reasons). Bits 9-7 are 0 for CSM optics, 1 for ATM sun sensor, and 2 for ATM star tracker (i.e. digit C in the ABCDE format).

STARIND: Single precision cell, scale factor B14, used in P51-P54 to cause two celestial bodies to be processed. If it is 1, BESTJ is used; if it is zero, BESTI is used. Initialized to 0 in "ROO", "P51B" (for P51/P53), "P50", and "P55" (and incremented to 1 for second body in P51/P53). Initialized to 1 in "R51" (P52/P54) for first body and decremented to 0 for second body.

STARSAV1: Temporary storage for sighting information (unit vector in IMU coordinates), scale factor B1. Loaded in "P51B" (P51/P53) with sighting vector to first body; in "R51DSP", loaded briefly with reference-coordinate information (from "PLANET") for first body, and then loaded with sighting vector to first body. The reference-coordinate information is used in R52 for optics pointing purposes. Also loaded in "TRKSTAR" (P55) with unit LOS vector of body in ATM sensor coordinates.

STARSAV2: Temporary storage for sighting information (unit vector in IMU coordinates), scale factor B1. See STARSAV1 for P51-P54 uses (the vector STARSAV2 is for the second body rather than the first, of course). Also loaded in "TRKSTAR" (P55) with unit vector formed by zeroing z component of STARSAV1 (and forming unit vector of result).

STARSAV3: Cells used by N88 for retention of "planet" information. N88 treats decimal input as double precision fraction, and since a unit vector is formed before quantity used (with components previously multiplied by $K_{\text{dsqrt}3}$), the DSKY input can be with arbitrary (but consistent) scaling.

T_{sight} : Value of sighting time, scale factor B28, units centi-seconds.

THETA: See Coordinate Transformations.

TRKAZ: Quantity computed double precision, but used single precision (no rounding), scale factor B0, units revolutions, displayed in R1 of N14. It gives the star tracker "azimuth" angle (outer gimbal angle or ψ_3).

TRKAZOCT: Single precision cell loaded in "TRKSTAR" (part of P55) with OSABC information, where S is sign of TRKAZ (0 if positive, 1 if negative), and ABC is the first three octal digits of $|TRKAZ|$ in units of arc minutes. It is displayed in R1 of N19.

TRKAZOCT+1: Single precision cell loaded in "TRKSTAR" with OODEO information, where DE is the last 2 octal digits of $|TRKAZ|$ in units of arc minutes. It is displayed in R2 of N19.

TRKEL: Quantity computed double precision, but used single precision (no rounding), scale factor B0, units revolutions, displayed in R2 of N14. It gives the star tracker "elevation" angle (inner gimbal angle or ψ_1).

VEARTH: Quantity computed in "S50" as the earth position (scale factor B1, a unit vector, when exit from routine) at time T_{sight} .

VELdC: Vector, scale factor B1, giving "aberration correction vector" (see K_{eclipol}). It is computed in "S50" and used in "PLANET"; it is also computed and used in "UTAREAL".

VSUN: Quantity computed in "S50" as the sun position (scale factor B1, a unit vector, when exit from routine) at time T_{sight} .

X_{smd} , Y_{smd} , Z_{smd} : Unit vectors, scale factor B1, giving desired stable member orientation. Loaded by "P52B" or "S40.2,3(B)", or can be loaded by P27 (X_{smd} address is 0306₈, the same as "UPBUFF" +2).

Values of K_{cat} (Star Table)

Display	Index	X Component	Y Component	Z Component	Identification
01	6	0.8759127335	0.0221767921	0.4819596176	α Andromedae
02	12	0.9344221007	0.1693349531	-0.3133384932	β Ceti
03	18	0.4792524242	0.1143361938	0.8701978789	γ Cassiopeiae
04	24	0.4911095585	0.2182968779	-0.8433017695	α Eridani
05	30	0.0150728422	0.0077535749	0.9998563354	α Ursae Minoris
06	36	0.5461199432	0.5290559424	-0.6495019764	θ Eridani
07	42	0.7065135180	0.7044331667	0.0679158478	α Ceti
10	48	0.4142717678	0.4969671969	0.7624975459	α Persei
11	54	0.3552697275	0.8910553149	0.2824957457	α Tauri
12	60	0.2051938364	0.9681249050	-0.1436302809	β Orionis
13	66	0.1416151994	0.6807526554	0.7186939246	α Aurigae
14	72	-0.0603437771	0.6034300291	-0.7951294413	α Carinae
15	78	-0.1784151505	0.9412937204	-0.2865905892	α Canis Majoris
16	84	-0.4076119204	0.9083859733	0.0932064683	α Canis Minoris
17	90	-0.3601048130	0.5763392844	-0.7335922251	γ Velorum
20	96	-0.4622023601	0.4795583503	0.7459173994	ι Ursae Majoris
21	102	-0.7717862824	0.6187078821	-0.1467872297	α Hydrae
22	108	-0.8583271674	0.4674686420	0.2115361489	α Leonis
23	114	-0.9649190792	0.0569171730	0.2563037376	β Leonis
24	120	-0.9533561730	-0.0550800729	-0.2967797042	γ Corvi
25	126	-0.4542931560	-0.0473466895	-0.8895931763	α' Crucis
26	132	-0.9189360613	-0.3461098015	-0.1891150984	α Virginis
27	138	-0.5810208598	-0.2883158706	0.7611101885	η Ursae Majoris
30	144	-0.6928505073	-0.4151341670	-0.5895946046	θ Centauri
31	150	-0.7878732623	-0.5182718149	0.3326410200	α Bootis
32	156	-0.5350048409	-0.7136778495	0.4521435028	α Coronae Borealis
33	162	-0.3561948806	-0.8224933890	-0.4434296246	α Scorpii
34	168	-0.1179551093	-0.3394503938	-0.9331988117	α Trianguli Australis
35	174	-0.1163388632	-0.9689823983	0.2180329804	α Ophiuchi
36	180	0.1195024086	-0.7708087413	0.6257579872	α Lyrae
37	186	0.2021911120	-0.8729034786	-0.4440250795	σ Sagittarii
40	192	0.4500934054	-0.8799661957	0.1519059600	α Aquilae
41	198	0.5479677261	-0.7957767725	-0.2578191213	β Capricorni
42	204	0.3165704246	-0.4450237255	-0.8376974691	α Pavonis
43	210	0.4530708165	-0.5412631657	0.7083509163	α Cygni
44	216	0.8118187290	-0.5593567634	0.1675421212	ϵ Pegasi
45	222	0.8322543673	-0.2429657768	-0.4983174684	α Piscis Austrini

The "Display" column gives the star number that is displayed by the program (as an octal quantity). The "Index" column gives the value of BESTI (or BESTJ) for the corresponding star, and is equal to six times the star serial number (i.e. six times the decimal equivalent of the first column).

A display of 00 indicates an N88 input; displays of 46 and 47 are for sun and earth respectively.

Mathematical Functions

Sine

Routine entered with argument in MPAC_{dp}, scale factor B0, units revolutions. Leaves MPAC_{dp} with sine of argument, scale factor B1.

$$X = \text{MPAC}_{dp}$$

Perform "SICOM"

$$\text{MPAC}_{dp} = \text{TS}$$

Return

SICOM

$$\text{If } |X| \geq \frac{1}{2}, X = \frac{1}{2} \text{sgn } X - X$$

$$\text{If } |X| \geq \frac{1}{4}, X = \frac{1}{2} \text{sgn } X - X$$

$x = X$, rescaled to scale factor B-1 revolutions, i.e. B1 in $(\pi/2)$ units.

$$\text{TS} = K_{sn1} x + K_{sn3} x^3 + K_{sn5} x^5 + K_{sn7} x^7 + K_{sn9} x^9$$

Return

<u>Constant</u>	<u>Scaling</u>	<u>Stored Value</u>	<u>True Value x (2/π)ⁱ</u>
K _{sn1}	B2	0.39269 90796	0.99999 9995
K _{sn3}	B0	-0.64596 37111	-0.16666 6567
K _{sn5}	B-2	0.31875 8717	0.00833 3025
K _{sn7}	B-4	-0.07478 0249	-0.00019 8074
K _{sn9}	B-6	0.00969 4988	0.00000 2603

According to the program comments, the constants are derived from a Hastings series.

Cosine

Routine entered with argument in MPAC_{dp}, scale factor B0, units revolutions. Leaves MPAC_{dp} with cosine of argument, scale factor B1.

$$X = \frac{1}{4} - |\text{MPAC}_{dp}|$$

Perform "SICOM"

MPAC_{dp} = TS

Return

Single Precision Sine (sin_{sp})

Routine entered with argument in computer hardware accumulator, ACC, single precision, scale factor B-1, units revolutions. Leaves sine in hardware accumulator, scale factor B0.

X = ACC

Perform "SPSICOM"

ACC = TS

Return

SPSICOM

If $|X| \geq \frac{1}{4}$, $X = \frac{1}{2} \text{sgn } X - X$

If $|X| \geq \frac{1}{4}$:

TS = MAX sgn X

Return

x = X, rescaled to scale factor B-2 revolutions, i.e. B0 in $(\pi/2)$ units.

TS = $K_{c1sp} x + K_{c3sp} x^3 + K_{c5sp} x^5$, limited in magnitude ≤ 1

Return

<u>Constant</u>	<u>Nominal Value</u>	<u>Stored Value</u>	<u>True Value x $(2/\pi)^i$</u>
K_{c1sp}	0.7853134	0.78533935	0.999925
K_{c3sp}	-0.3216147	-0.32159423	-0.165951
K_{c5sp}	0.0363551	0.03637695	0.007608

All constants have scale factor B1, for $\pi/2$ units.

Single Precision Cosine (cos_{sp})

Routine entered with argument in computer hardware accumulator, ACC, single precision, scale factor B-1, units revolutions. Leaves cosine in hardware accumulator, scale factor B0.

$$X = ACC + \frac{1}{4}$$

If $|X| \geq \frac{1}{2}$:

$$X = - (X - \frac{1}{2} \operatorname{sgn} X)$$

Perform "SPSICOM"

$$ACC = TS$$

Return

Arc Sine

Routine entered with argument in $MPAC_{tp}$, scale factor B1. Leaves $MPAC_{dp}$ with arc sine of argument, scale factor B0, units revolutions, in range $\pm \frac{1}{4}$ (i.e. $\pm 90^\circ$).

$$X = MPAC_{tp}$$

Perform "ARCCOM"

$$MPAC_{dp} = \frac{1}{4} - TS$$

Return

ARCCOM

If $|X| = 0$, $TS = \frac{1}{4}$; Return

$$TS_1 = X$$

$$X = |X|$$

If $X \gg (1 + 2^{-12})$, or 1.000244, or if least significant half of X is negative and $X \gg (1 + 2^{-13})$, or 1.000122:

Perform "ALARM" (pattern 1301₈)

If $X \gg 1$, set $TS = 0$ (for TS_1 positive) or $TS = \frac{1}{2}$ (for TS_1 negative); Return

$TS = (1 - X)^{\frac{1}{2}}$ (scale factor B1 when enter square root routine)

$$TS = TS(K_{as0} + K_{as1} X + K_{as2} X^2 + K_{as3} X^3 + K_{as4} X^4 + K_{as5} X^5 + K_{as6} X^6 + K_{as7} X^7)$$

If TS_1 negative, $TS = \frac{1}{2} - TS$

Return

<u>Constant</u>	<u>True Value</u>	<u>$\sqrt{2} \pi$ x True Value</u>
K_{as0}	0.35355 3385	1.57079 6302
K_{as1}	-0.04830 17006	-0.21459 8801
K_{as2}	0.02002 73085	0.08897 8987
K_{as3}	-0.01129 31863	-0.05017 4305
K_{as4}	0.00695 311612	0.03089 1881
K_{as5}	-0.00384 617957	-0.01708 8126
K_{as6}	0.00150 1297736	0.00667 0090
K_{as7}	-0.00028 4160334	-0.00126 2491

All constants are stored with scale factor B-I in program (e.g. K_{as5} has scale factor B-5); the $\sqrt{2}$ factor is required because $(1 - X)$ is scaled at B1 when take its square root. The numbers in the last column agree closely with published Hastings series values.

Arc Cosine

Routine entered with argument in $MPAC_{tp}$, scale factor B1. Leaves $MPAC_{dp}$ with arc cosine of argument, scale factor B0, units revolutions, in range $0 - \frac{1}{2}$ (i.e. $0^\circ - 180^\circ$).

$X = MPAC_{tp}$

Perform "ARCCOM"

$MPAC_{dp} = TS$

Return

Square Root

Routine entered with argument in $MPAC_{tp}$, and with scale factor (in general) an even number. Leaves $MPAC_{tp}$ with square root of argument, having scale factor one-half the scale factor of argument (a $\sqrt{2}$ factor must be accounted for elsewhere if the scale factor was odd when enter the routine).

$X = MPAC_{tp}$, considered to have scale factor B0

If $X = 0$, $MPAC_{tp} = 0$; Return
 If $X < -2^{-14}$, proceed to "POODOO" (pattern 21302₈)
 If $X < 0$, $MPAC_{dp} = 0$; Return
 $TS = X$, normalized to lie between $\frac{1}{4}$ and < 1 . This is accomplished by shifting X left in increments of 2 shifts, and storing the required number of shifts in TS_1 (an even number $\gg 0$)
 If $TS \gg \frac{1}{2}$, $TS_2 = 0.5884 TS_{sp} + 0.4192$
 If $TS < \frac{1}{2}$, $TS_2 = 0.8324 TS_{sp} + 0.2974$
 $TS_2 = (\frac{1}{2} TS_{sp}) / TS_2 + \frac{1}{2} TS_2$ (TS_2 single precision)
 $TS = \frac{1}{2} TS_2 + \frac{1}{2} TS / TS_2$
 If $TS \gg 1$, $TS = (1 - 2^{-28})$
 $MPAC_{tp} = TS$, shifted right by $(\frac{1}{2} TS_1)$ places
 Return

Natural Logarithm (\log_e)

This routine is entered from "RANGER", with the argument in $MPAC_{dp}$, scale factor B0. It leaves the natural logarithm of the reciprocal of the argument (or the complement of the logarithm of the argument) in $MPAC_{dp}$, scale factor B5.

LOG

$TS = MPAC_{dp}$, normalized to lie between $\frac{1}{2}$ and < 1 . This is accomplished by shifting $MPAC_{dp}$ left and storing the required number of shifts in TS_1 (a number $\gg 0$).

$X = 1 - TS$ (The "1" actually is $1 - 2^{-28}$)

$MPAC_{dp} = K_{cog1} TS_1 + K_{og1} X + K_{og2} X^2 + K_{og3} X^3$

Return (with value of $-\log_e$ of original $MPAC_{dp}$)

All constants are stored with scale factor B5. The value of K_{cog1} corresponds to $\log_e 2$.

<u>Constant</u>	<u>Stored Value</u>	<u>True Value</u>
K_{cog1}	0.0216608494	0.69314718056
K_{og1}	0.031335467	1.00273494
K_{og2}	0.0130145859	0.4164667488
K_{og3}	0.0215738898	0.6903644736

Measurement Incorporation

INCORP1

$$\underline{Z}_0 = [W_0] \text{ BVECTOR}_{\underline{0}} + [W_3] \text{ BVECTOR}_{\underline{1}}$$

$$\underline{Z}_1 = [W_1] \text{ BVECTOR}_{\underline{0}} + [W_4] \text{ BVECTOR}_{\underline{1}}$$

$$\text{TRIPA} = |\underline{Z}_0|^2 + |\underline{Z}_1|^2 + \text{VARIANCE}$$

$$\text{TS} = \sqrt{\text{TRIPA VARIANCE}}$$

computed quasi-floating point
(TRIPA rounded to double precision
before multiplication use)

$$\text{GAMMA} = 1/(\text{TS} + \text{TRIPA})$$

computed quasi-floating point

$$\text{DELQDA} = \text{DELTAQ} / \text{TRIPA}$$

computed quasi-floating point

Set TS_1 to the largest $|\underline{Z}_i|$ ($i = 0, 1$) (tag here "NEWZCOMP")

$\text{TS}_2 = (\text{number of leading zeros in } \text{TS}_1) - 2$ (value used for subsequent scaling)

Shift each \underline{Z}_i left by TS_2 places ($i = 0, 1$)

$$\text{OMEGA}_{\underline{0}} = \underline{Z}_0 [W_0] + \underline{Z}_1 [W_1]$$

$$\text{OMEGA}_{\underline{1}} = \underline{Z}_0 [W_3] + \underline{Z}_1 [W_4]$$

$$\text{DELTA}_{\underline{0}} = \text{DELQDA} \text{ OMEGA}_{\underline{0}}$$
 computed quasi-floating point

$$\text{DELTA}_{\underline{1}} = \text{DELQDA} \text{ OMEGA}_{\underline{1}}$$
 computed quasi-floating point

Return

INCORP2

EGRESS = Return address

Perform "INTSTALL"

$$\text{OMEGAM}_{\underline{0}} = \text{GAMMA} \text{ OMEGA}_{\underline{0}}$$

$$\text{OMEGAM}_{\underline{1}} = \text{GAMMA} \text{ OMEGA}_{\underline{1}}$$

Set bit 13(INTGRAB) of FLGWRD10 = 1 (means restart phase change)

$$[W_0] = [W_0] - (Z_0) \text{ OMEGAM}_{\underline{0}}$$

NOTE: Notation means computation carried out as:

$$[W_1] = [W_1] - (Z_1) \text{ OMEGAM}_{\underline{0}}$$

$$[W_3] = [W_3] - (Z_0) \text{ OMEGAM}_{\underline{1}}$$

$$[W_4] = [W_4] - (Z_1) \text{ OMEGAM}_{\underline{1}}$$

$$[W_0] = [W_0] - \begin{bmatrix} Z_{0,x} \\ Z_{0,y} \\ Z_{0,z} \end{bmatrix} \text{ OMEGAM}_{\underline{0}}$$

If bit 8(CSMUPDAT) of FLAGWRD1 = 1: (Tag here "FAZC")

Perform "MOVEPCSM"

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Perform "MOVEPLEM"

Reset overflow indicator

$$T\underline{S} = T\underline{\Delta TAV} + (\underline{\Delta TAX}_0 \text{ shifted left 7 places})$$

If overflow has taken place since indicator reset:

$$R\underline{CV} = R\underline{CV} + \underline{\Delta TAX}_0$$

$$V\underline{CV} = V\underline{CV} + \underline{\Delta TAX}_1$$

Perform "RECTIFY"

Proceed to "FAZAB3"

$$T\underline{\Delta TAV} = T\underline{S}$$

$$T\underline{S} = T\underline{NUV} + (\underline{\Delta TAX}_1 \text{ shifted left 4 places})$$

If overflow has taken place:

$$V\underline{CV} = V\underline{CV} + \underline{\Delta TAX}_1$$

Perform "RECTIFY"

Proceed to "FAZAB3"

$$T\underline{NUV} = T\underline{S}$$

Proceed to "FAZAB3"

FAZAB3

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

Perform "MOVEACSM"

(If bit 8(CSMUPDAT) of FLAGWRD1 = 1):

If bit 1(AVEMIDSW) of FLAGWRD9 = 0:

$$\underline{R} = \underline{RCV} + \underline{TDELTA\underline{V}}$$

$$\underline{V} = \underline{VCV} + \underline{TNUV}$$

$$T_{\text{pptm}} = T_{\text{et}}$$

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Perform "MOVEALEM"

$$\underline{R}_{\text{other}} = \underline{RCV} + \underline{TDELTA\underline{V}}$$

$$\underline{V}_{\text{other}} = \underline{VCV} + \underline{TNUV}$$

QPRET = EGRESS

Proceed to "INTWAKE" (returns to address in QPRET)

V67CALL Established by "V67" for a verb 67.

Reset overflow indicator

Perform "INTSTALL"

WWOPT = 0

$$\underline{WWPOS} = \sqrt{|\underline{W}_0|^2 + |\underline{W}_3|^2 + |\underline{W}_6|^2 + |\underline{W}_9|^2 + |\underline{W}_{12}|^2 + |\underline{W}_{15}|^2}$$

$$\underline{WWVEL} = \sqrt{|\underline{W}_{27}|^2 + |\underline{W}_{30}|^2 + |\underline{W}_{33}|^2 + |\underline{W}_{36}|^2 + |\underline{W}_{39}|^2 + |\underline{W}_{42}|^2}$$

(i.e., \underline{W}_{3i} , $i = 9-14$)

If overflow has taken place since indicator reset:

$$\underline{WWPOS} = +\text{MAX} \quad (2^{19} - 2^{-9} \text{ meters})$$

$$\underline{WWVEL} = +\text{MAX} \quad (\text{N99 R2 display } 328.1 \text{ fps})$$

If $\underline{WWPOS} - K_{\text{ft999}} \geq 0$:

$$\underline{WWPOS} = K_{\text{ft999}} \quad (\text{constant gives display of } 99997 \text{ feet})$$

Perform "INTWAKE" (returns to next line)

TS = 0699_{vn}

Proceed to "GOXDSPF": if terminate, proceed to "ENDEXT"
if proceed, proceed
otherwise, proceed to previous line

If WWOPT = 0: (no change in W-matrix initialization
cells desired)

Proceed to "ENDEXT"

WRENDPOS = WWPOS_{sp}

WRENDVEL = WWVEL_{sp} (must set RENDWFLG = 0 by e.g. V93E)

Proceed to "ENDEXT"

GETUM Entered from "REND7"

Perform "SXTSM" starting at fourth line (STARIND = 0 from "ROO")

TS = TS [REFSMMAT]

Return

BVECTORS Entered from "REND7"

BVECTOR₀ = unit(USTAR * unitRCLP)

USTAR = BVECTOR₀

DELTAQ = $K_{2\pi} |RCLP| \left(\cos^{-1} (BVECTOR_0 \cdot UM) - 1/4 \right)$ (1/4 = 90°)

BVECTOR₁ = 0

Return

R22 (Established by "NDUTINPT" for P20 option 0/4, and entered due to
restart group 2.7)

Change priority of present job to 26_g (restart group 2.7 has 10_g
for establishing "R22")

MRKBUF1 = -3

VHFTIME = T_{now}

Proceed to "REND1"

REND1

Change restart group 2 to cause a start at next line (subsequent
restart logic not shown unless significant for normal logic)

Perform "WAITONE"

If MRKBUF1 \leq 0: (Tag here "REND1A")
Proceed to "REND3"
Inhibit interrupts
MARKTIME+i = MRKBUF1+i (i = 0 - 6)
MRKBUF1 = -3
Set bit 7(R22CAFLG) of FLAGWRD9 = 1 (for "MKREJECT" use)
Release interrupts
Set bit 8(VHFSOURC) of FLAGWRD9 = 0
Proceed to "REND4"

REND3

Change restart group 2 to cause a start at next line
Perform "WAITONE"
If bit 9(VHFRFLAG) of FLAGWRD9 = 0:
Proceed to 3rd line of "REND1"
If ($T_{\text{now}} - K_{60\text{secdp}} - \text{VHFTIME}$) < 0:
Proceed to 3rd line of "REND1"
Perform "RANGERD": if return to calling address +1, proceed to "R22BAD"
if return to calling address +2, proceed to "R22GOOD"

RANGERD This is RO8, entered from "REND3", "R27GO", and "CYCVHF".

RREADRET = return address (calling address +1)
Inhibit interrupts
Set bits 4-1 (Radar selection) of channel 13 = 0
Proceed to "RADSTART"

RADSTART

TS = channel 4
TS₁ = (bits 5-1 of TS)

RADDEL = bits 5-1 of $(00040_8 - TS_1)$

If $RADDEL \leq 00002_8$: (bit 6 of channel 4, in phase with 100
pps signal used for radar control,
Proceed to "RADSTART" about to be incremented)

Set bits 1 and 4 (VHF select and Radar initiate) of channel 13 = 1

RADTIME = -TS

Release interrupts

MARKTIME = T_{now}

Set bit 10(SKIPVHF) of FLAGWRD2 = 0 (set 1 in "STARTSB2")

Put present job to sleep (starting address id = RREADRET)

VHFREAD (Entered based on program interrupt #9, Radar Rupt)

If bit 10(SKIPVHF) of FLAGWRD2 = 1: (set in "STARTSB2")

Resume

RM = RNRAD

If bit 2(VHF Data Good complement) of channel 33 = 0:

TS = "RANGERD1"

If bit 2(VHF Data Good complement) of channel 33 = 1:

$VHFRANGE_{sp} = 2$

TS = "LIGHTON"

Awaken job with starting address id = RREADRET (put to sleep in
"RADSTART")

Set starting address of awakened job = TS

Resume

LIGHTON (Entered via "VHFREAD" logic for unsuccessful radar read)

If $VHFRANGE_{sp} \neq 0$:

Set bit 8(Tracker) of DSPTAB+11 = 1, and flag for output at
next opportunity

Proceed to address specified by RREADRET (error return)

RANGERD1 (Entered via "VHFREAD" logic for successful radar read)

Inhibit interrupts

Set bit 8(Tracker) of DSPTAB+11 = 0, and flag for output at next opportunity

Set bit 7(OCDUFBIT) of OPTMODES = 1

Release interrupts

If RM = +0:

Proceed to address specified by RREADRET (error return)

If bit 15 of RM = 0:

$TS = K_{\text{convrnge}} (RM, 0)$

If bit 15 of RM = 1: (including case of 77777₈)

$TS = (\text{bits } 14-1 \text{ of } RM)$

$TS = K_{\text{convrnge}} (TS, 0) + 2^{14} K_{\text{convrnge}}$

Proceed to "RANGERD7"

RANGERD7

VHFRANGE = TS

If bit 9(EXTRANGE) of FLGWRD10 = 1: (set in "CRS61.1"; reset in "ROO")

If $(R63RANGE - TS - K_{\text{vhfmodlo}}) \gg 0$:

$TS = VHFRANGE + K_{328nm}$ (constant is 327.68 nmi)

Proceed to "RANGERD7"

Proceed to address specified by (RREADRET +1) (good return)

R22BAD (Entered from "REND3" for error return, and from "R22GOOD")

VHF TIME = MARK TIME

Proceed to "REND1"

R22GOOD (Entered from "REND3" for good return)

If bit 7(UPDATFLG) of FLAGWRD1 = 0:

Proceed to "R22BAD"

If bit 12(PDSPFLAG) of FLAGWRD4 = 1: (R61 doing R60)

Proceed to "R22BAD"

Set bit 8(VHFSOURC) of FLAGWRD9 = 1

VHFTIME = MARKTIME

Proceed to "REND4"

REND4 (Entered from "R22GOOD" for VHF measurement, and from "REND1"
for an optics measurement)

Perform "SETINTG"

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

Set bit 3(CSMINTSW) of FLAGWRD3 = 0

Perform "INTEGRV"

Perform "SETINTG"

If bit 1(RENDWFLG) of FLAGWRD5 = 1:

Set bit 1(WMATINT) of FLAGWRD3 = 1

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

Set bit 3(CSMINTSW) of FLAGWRD3 = 0

Perform "INTEGRV" (when exit, OWS and CSM vectors both integrated
"permanently" to MARKTIME)

$CSMPOS = RCV_{cm} + (DELTA V_{cm} \text{ shifted right } 7 \text{ places})$

Set bit 11(FSTINCRP) of FLAGWRD5 = 1

$LEMPOS = RCV_{lm} + (DELTA V_{lm} \text{ shifted right } 7 \text{ places})$

If bit 7(AUTOSEQ) of FIGWRD10 = 1:

Proceed to "AUTOW" (exits to "REND5C" if initialize, or "REND7"
if don't)

If bit 1(RENDWFLG) of FLAGWRD5 = 1:

Set bit 1(RENDWFLG) of FLAGWRD5 = 1 (redundant, Tag "REND6")

Proceed to "REND7"

Proceed to "REND5C"

REND5C

VHFCNT = 0

TRKMKCNT = 0 (as "least significant half" of VHFCNT)

$W_i = 0$ (i = 0 - 17) (i.e. $[W_0]$ and $[W_1]$)

$W_i = 0$ (i = 27 - 44) (i.e. $[W_3]$ and $[W_4]$)

$W_i = \text{WRENDPOS}$ (i = 0, 4, 8, diagonal elements of $[W_0]$)

$W_i = \text{WRENDVEL}$ (i = 36, 40, 44, diagonal elements of $[W_4]$)

Set bit 8(P35FLAG) of FLGWRD10 = 0

AGEOFW = MARKTIME

Set bit 4(PTV93FLG) of FLGWRD10 = 0

COUNT3MK = 1

Set bit 1(RENDWFLG) of FLAGWRD5 = 1

Proceed to "REND7"

REND7

RCLP = LEMPOS - CSMPOS

If bit 8(VHFSOURC) of FLAGWRD9 = 0: (i.e. have optics data)

UCL = unitRCLP

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

X1 = - "MARKTIME"

Perform "GETUM"

UM = TS

TS = UCL * UM

If all components of TS $< 2^{-19}$ rad: (so unit vector overflow)

Proceed to "RENDISP3"

USTAR = unitTS

Perform "BVECTORS" (Tag here "REND9")

(If bit 8(VHFSOURC) of FLAGWRD9 = 0):

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

$$\text{BVECTOR}_{\underline{0}} = - \text{BVECTOR}_{\underline{0}}$$

If bit 14(R21MARK) of FLAGWRD2 = 1: (i.e. prime optics)

$$\text{TS}_{\underline{1}} = K_{\text{sxtvar}} + K_{\text{imuvar}}$$

If bit 14(R21MARK) of FLAGWRD2 = 0: (i.e. R23 optics)

$$\text{TS}_{\underline{1}} = C_{\text{altvar}} + K_{\text{imuvar}} \quad (\text{Tag here "REND15"})$$

If bit 8(VHFSOURC) of FLAGWRD9 = 1: (i.e., have VHF ranging data)

$$\text{BVECTOR}_{\underline{0}} = \text{unitRCLP} \quad (\text{Tag here "REND14"})$$

$$\text{BVECTOR}_{\underline{1}} = 0$$

TS = VHFRANGE, rescaled to B29

Set bit 11(FSTINCRP) of FLAGWRD5 = 1

$$\text{DELTAQ} = \text{TS} - |\text{RCLP}|$$

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

$$\text{BVECTOR}_{\underline{0}} = - \text{BVECTOR}_{\underline{0}}$$

$$\text{TS}_{\underline{1}} = C_{\text{rvar}}$$

$$\text{TS} = |\text{RCLP}|^2 \text{TS}_{\underline{1}} \quad \text{Computed quasi-floating point} \quad (\text{Tag here "REND10"})$$

$$\text{VARIANCE} = \text{TS}_{\text{tp}} + C_{\text{intvar}} \quad (\text{B40})$$

If bit 8(VHFSOURC) of FLAGWRD9 = 1:

If $(\text{VARIANCE} + C_{\text{rvarmin}}) < 0$: (C_{rvarmin} negative)

$$\text{VARIANCE} = |C_{\text{rvarmin}}|$$

Perform "INCORP1"

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

$$\text{N49DISP}+2 = |\text{DELTA}_{\underline{1}}|$$

$$\text{N49DISP} = |\text{DELTA}_{\underline{0}}|$$

$$\text{N49DISP}+4 = 2$$

(If bit 11(FSTINCRP) of FLAGWRD5 = 1):

If bit 8(VHFSOURC) of FLAGWRD9 = 0: (processing optics)

If bit 12(REJECTFLG) of FLGWRD10 = 1: (set in "MKREJECT")

Proceed to "RENDISP3"

$N49DISP+4 = 1$

If $(C_{rmax} - N49DISP) < 0$: ("N49DISP" same as "BVECTOR₀")

Proceed to "RENDISP"

If $(C_{vmax} - N49DISP+2) < 0$:

Proceed to "RENDISP"

Proceed to "REND12"

REND12

Perform "INCORP2" (incorporates update, same time tag as from "REND4" integration)

If bit 11(FSTINCRP) of FLAGWRD5 = 1:

OLDMKTME = MARKTIME

Set bit 5(MANEUFLG) of FLGWRD10 = 0

TS = COUNT3MK

If TS \neq 0:

If $(TS - 3) \gg 0$:

COUNT3MK = 0

If $(TS - 3) < 0$:

COUNT3MK = COUNT3MK + 1

If bit 8(VHFSOURC) of FLAGWRD9 = 1: (Tag here "REND12A")

VHFCNT = VHFCNT + 1

Proceed to "RENDISP3"

If bit 11(FSTINCRP) of FLAGWRD5 = 0: (i.e. 2nd optics incorporation has been completed)

TRKMKCNT = TRKMKCNT + 1

Proceed to "RENDISP3"

If bit 8(CSMUPDAT) of FLAGWRD1 = 1:

$CSMPOS = RCV_{cm} + (\text{DELTA}V_{cm} \text{ shifted right } 7 \text{ places})$

If bit 8(CSMUPDAT) of FLAGWRD1 = 0:

$LEMPOS = RCV_{lm} + (\text{DELTA}V_{lm} \text{ shifted right } 7 \text{ places})$

Set bit 11(FSTINCRP) of FLAGWRD5 = 0

Proceed to "REND7"

RENDISP

TEMPOR1 = +0

Establish "RENDISP2" (priority 27_g)

Allow performance of higher priority jobs (e.g. "RENDISP2")

If TEMPOR1 = 0: (i.e. display not answered yet)

Proceed to second previous line (note this loop locks out lower priority jobs, including R52)

If TEMPOR1 < 0: (i.e. a PRO display response)

Proceed to "REND12"

Proceed to "RENDISP3"

RENDISP2 (Established by "RENDISP")

TS = 0649_{vn}

Proceed to "PRIODSP": if terminate, proceed to "TRACKTRM"
if proceed, TS = -1 and proceed
otherwise, TS = QREG and proceed

TEMPOR1 = TS

End of job

RENDISP3 (Entered at end of each mark-processing pass)

Set bit 7(R22CAFLG) of FLAGWRD9 = 0

Set bit 12(REJCTFLG) of FLAGWRD10 = 0

If bit 8(VHFSOURC) of FLAGWRD9 = 1:

Proceed to "REND1"

Proceed to "REND3"

WAITONE

POINTEX = Return address

Delay 4 seconds (by putting job to sleep via "DELAYJOB")

If bit 7(RNDVZFLG) of FLAGWRDO = 0, End of job

If bit 13(REFSMFLG) of FLAGWRD3 = 0, End of job

If bit 5(TRACKFLG) of FLAGWRD1 = 0, End of job

If R61CNTR < 0: (R61 maneuver in process)

Proceed to second line of "WAITONE"

If bit 12(R27FLAG) of FIGWRD11 = 1:

Perform "R27GO"

If bit 6(SNAPFLAG) of FIGWRD11 = 1:

Proceed to second line of "WAITONE"

If bit 7(UPDATFLG) of FLAGWRD1 = 1:

Proceed to address specified by POINTEX

Proceed to "REDOR22"

REDOR22 (Entered from "WAITONE" and due to restart group 2.13)

Set restart group 2 to phase 13 (2.13, causing "REDOR22" to be established with priority 10₈ if a restart)

Change priority of present job to 26₈

Proceed to second line of "WAITONE"

SETINTG

Perform "INTSTALL"

Set bit 5(STATEFLG) of FLAGWRD3 = 1

T_{decl} = MARKTIME

Set bits 4(CONICINT) and 1(WMATINT) of FLAGWRD3 = 0

Set bit 3(CSMINTSW) of FLAGWRD3 = 1

Return

R27GO Entered from "R27JOB" (P25), "R27JOBA" (P48), and "WAITONE" (R22)

R27END = Return address

Perform "RANGERD": if return to calling address +1, proceed to address
specified by R27END
if return to calling address +2, proceed

TIM1 = MARKTIME (PD 0D)

RANG1 = VHFRANGE (PD 2D)

NPASS = -4

If bit 1(R27UP1) of FLGWRD11 = 1:

If bit 4(P25FLAG) of FLGWRD11 = 1:

Perform "VHFMOD"

If bit 1(R27UP1) of FLGWRD11 = 0:

RBUF = VHFRANGE

Proceed to "CYCVHF"

CYCVHF

NPASS = NPASS + 1

Perform "RANGERD": if return to calling address +1, proceed to address
specified by R27END
if return to calling address +2, proceed

TIM2 = MARKTIME (PD 4D)

RANG2 = VHFRANGE (PD 6D)

If (RANG2 - RANG1) \neq 0:

If bit 4(P25FLAG) of FLGWRD11 = 1:

Perform "VHFMOD"

TS = RANG2 - RANG1

Proceed to "TAKE2"

TIM1 = TIM2

TS = NPASS

If TS \neq 0:

Proceed to "CYCVHF"

Proceed to "TAKE2"

TAKE2

If TS \geq 0:

RANG1 = RANG2

TIM1 = $\frac{1}{2}$ (TIM2 + TIM1) (overflow if exceed 2^{27} cs) (now T_m)

RANG1 = RANG1, shifted left 7 places (B20 meters) (now R_m)

If bit 2(R27UP2) of FLGWRD11 = 1: (Tag here "SAMPLE")

Proceed to "WUNZMOR"

If bit 1(R27UP1) of FLGWRD11 = 0:

OPVEC = RANG1

OPVEC+2 = TIM1

Set bit 1(R27UP1) of FLGWRD11 = 1

Proceed to address specified by R27END

SVTEM = $\frac{1}{2}$ RANG1 + $\frac{1}{2}$ OPVEC (Tag here "HOWMUCHR")

TEMTYM = $\frac{1}{2}$ (TIM1 + OPVEC+2)

TS = (TIM1 - OPVEC+2), shifted left 14 places (time difference)

SVTEM+2 = (RANG1 - OPVEC)/TS (B6 m/cs)

SVTEM+4 = 0

WMAT_i = 0 (i = 1, 2, 3, 5, 6, 7)

WMAT₀ = K_{upperlef}

WMAT₄ = (K_{middle}) / TS (TS has time difference)

WMAT₈ = K_{bottomri}

VSAVTEM = DELVIMU_x

Set bit 7(FIXFLAG) of FLGWRD11 = 0

Set bit 2(R27UP2) of FLGWRD11 = 1

Proceed to "STOR77"

VHFMOD

TS = RBUF - VHFRANGE (in 10D)

If $(|TS| - \frac{1}{4} K_{\text{vhfmodlo}}) < 0$: ($\frac{1}{4}$ due to fact that TS is B27,
but constant is B29)

RBUF = VHFRANGE

Return

RBUF = $K_{328\text{nm}}$ sgn TS (constant is 327.68 nmi)

If NPASS \neq -4: (i.e. not from "R27GO")

RANG1 = RANG1 - RBUF

If bit 2(R27UP2) of FLGWRD11 = 0:

OPVEC = OPVEC - RBUF

If bit 2(R27UP2) of FLGWRD11 = 1:

SVEC = SVEC - RBUF

RBUF = VHFRANGE

Return

WUNZMOR

Inhibit interrupts

WMAT_i = VHF.W_i (i = 0 - 6)

Release interrupts (allow e.g. telemetry)

Inhibit interrupts

WMAT_i = VHF.W_i (i = 7 - 8)

SVTEM = SVEC

SVTEM+2 = SVEC+2

SVTEM+4 = SVEC+4

TEMTYM = TYMR

VSAVTEM = VSAV

Release interrupts

If bit 5(P48FLAG) of FLGWRD11 = 1:

SVTEM+2 = SVTEM+2 - (DELVIMU_x - VSAVTEM)

VSAVTEM = DELVIMU_x (Note value could be different from above)

VHF.OMEG = RANG1 (measured value, B20 meters) (Tag "TMCORP")

VHF.B+2 = (TIM1 - TEMTYM), shifted left 13 places (B15 cs)

VHF.B+4 = $\frac{1}{2}$ (VHF.B+2)² (B30 cs²)

PD2DTM = TEMTYM (save in 2D)

VHF.B+0 = 1 (B1)

PD4DLQ = VHF.OMEG - (VHF.B+0, VHF.B+2, VHF.B+4) · (SVTEM, SVTEM+2, SVTEM+4)

VHF.Z = (VHF.B+0, VHF.B+2, VHF.B+4) [WMAT]

TS₂ = K_{vargap} + |VHF.Z|² (B22, in 6D)

TS₃ = 1 / TS₂ (computed quasi-floating point)

VHF.OMEG = [WMAT] VHF.Z

VHF.OMEG = TS₃ VHF.OMEG (components VHF.OMEG+0, +2, +4)

(SVTEM, SVTEM+2, SVTEM+4) = (SVTEM, SVTEM+2, SVTEM+4) + PD4DLQ VHF.OMEG

TS = $1 / \left(\sqrt{K_{\text{vargap}} / TS_2} + 1 \right)$ (in TEMTYM cell, B1)

WMAT₆ = WMAT₆ - VHF.Z (VHF.OMEG+4 TS)

WMAT₃ = WMAT₃ - VHF.Z (VHF.OMEG+2 TS)

WMAT₀ = WMAT₀ - VHF.Z (VHF.OMEG+0 TS)

If bit 7(FIXFLAG) of FLGWRD11 = 0: (Tag here "SWITCHGO")

Proceed to "WOTNOW" (not optimizing)

TEMTYM = FIXTIME

TS = TIM1 - FIXTIME

If TS < 0:

Proceed to "SNAPPY" (FIXTIME in future)

If $(TS - K_{95secs}) < 0$:

If bit 5(P48FLAG) of FLGWRD11 = 0:

Set bit 3(TDFLAG) of FLGWRD11 = 1

Set bit 6(SNAPFLAG) of FLGWRD11 = 0 (Tag here "P48OPDUN")

Set bit 9(N77FLAG) of FLGWRD11 = 0

Proceed to "THISJAZZ"

OPVEC = SVTEM (Tag here "OPDONE"; load R1 and R2 of N77)

OPVEC+2 = SVTEM+2

TEMTYM = TIM1

VHF.B+2 = (TIM1 - FIXTIME), shifted left 13 places (B15 cs)

VHF.B+4 = $\frac{1}{2}$ (VHF.B+2)² (B30 cs²)

FIXTEMP = FIXTIME + K_{4mins}

Set restart group 2 to phase 17 (2.17, causing "ENDFIX" to be established with priority 26₈ if a restart)

FIXTIME = FIXTEMP

Set bit 9(N77FLAG) of FLGWRD11 = 1

Set bit 7(FIXFLAG) of FLGWRD11 = 0

Proceed to "XTRAP"

SNAPPY

If $TIM1 - (FIXTIME - K_{20secs}) \gg 0$:

Set bit 6(SNAPFLAG) of FLGWRD11 = 1

(i.e. within 20 seconds before FIXTIME)

PHETA = K_{dpm01}

Proceed to "THISJAZZ"

WOTNOW

TS = TIM1 - FIXTIME

If $TS \gg 0$:

TEMTYM = TIM1

Proceed to "XTRAP"

If $(TS + K_{95secs}) < 0$:

TEM_{TYM} = TIM_L

Proceed to "XTRAP"

TEM_{TYM} = FIXTIME (time to start new optimization)

VHF.B+2 = (FIXTIME - PD2DTM), shifted left 13 places (B15 cs)

VHF.B+4 = $\frac{1}{2}$ (VHF.B+2)² (B30 cs²)

Set restart group 2 to phase 11 (2.11, causing "STARTFIX" to be established with priority 26₈ if a restart)

Set bit 9(N77FLAG) of FLGWRD11 = 1

Set bit 7(FIXFLAG) of FLGWRD11 = 1

Proceed to "XTRAP"

XTRAP

PMATRIX₀ = VHF.B (same cells; shown for documentation convenience)

PMATRIX₃ = 0

PMATRIX₄ = 1

PMATRIX₅ = VHF.B+2 (B16 cs)

PMATRIX₆ = unitZ (i.e. 0,0,1)

(SVTEM, SVTEM+2, SVTEM+4) = [PMATRIX] (SVTEM, SVTEM+2, SVTEM+4)

WMAT₀ = PMATRIX₀ [WMAT]

WMAT₃ = PMATRIX₃ [WMAT]

TS = - (WMAT₃ * WMAT₀)

If all components of TS < 2⁻¹¹ m²/cs: (so interpreter vector magnitude operation gives 0)

TS = 2¹⁴ TS

Proceed to second previous line (stay in loop if all TS components 0)

PDCVC = TS (in 0D)

PDDVL = PDCVC · WMAT₆ (in 6D)

PDSVL = |PDCVC|² (in 8D)

PDFVL = (C_{rtridot} VHF.B+2)² (in 10D)

$TS = (PDFVL/PDDVL^2) PDSVL$ (note scaling of PDCVC cancels)

If overflow has taken place (e.g. $|TS| \geq 2^{18}$):

$$TS_1 = +MAX \quad (2^9 - 2^{-19})$$

If overflow has not taken place:

$$TS_1 = \sqrt{TS + 1} - 1 \quad (B9)$$

$$WMAT_6 = WMAT_6 + \left((PDDVL TS_1) / PDSVL \right) PDCVC$$

(addition of 1 not
protected from overflow)
(PDCVC scaling
again cancels)

Proceed to "THISJAZZ"

THISJAZZ

If bit 4(P25FLAG) of FLGWRD11 = 1:

Proceed to "STOR77"

If bit 3(TDFLAG) of FLGWRD11 = 1:

Proceed to "STOR77"

If bit 6(SNAPFLAG) of FLGWRD11 = 1:

Proceed to "STOR77"

If bit 5(P48FLAG) of FLGWRD11 = 0:

If bit 14(INTINUSE) of FLGWRD10 = 0:

CDUSNAP = CDU

CDUSNAPT = CDUT

CDUSNAPS = CDUS

Perform "PHICOMP"

Proceed to "STOR77"

$TS_1 = R$ (Tag here "STHETA")

$TS_2 = V$ (TS_1 in 18D; TS_2 in 24D; unit TS_1 in OD)

$TS = unitX$

Perform "ANGLER"

If bit 7(FIXFLAG) of FLGWRD11 = 1:

If $(T_{now} - FIXTIME) \geq 0$:

Set bit 3(TDFLAG) of FLGWRD11 = 1

Proceed to "STOR77"

STOR77

If bit 9(N77FLAG) of FIGWRD11 = 0:

OPVEC = SVTEM (load R1 and R2 of N77)

OPVEC+2 = SVTEM+2

Inhibit interrupts (Tag here "GARDOPT") (Restart group 2 set
to cause start at next line: other
settings not shown)

VHF.W_i = WMAT_i (i = 0 - 6)

Release interrupts (allow e.g. telemetry)

Inhibit interrupts

VHF.W_i = WMAT_i (i = 7 - 8)

SVEC = SVTEM

SVEC+2 = SVTEM+2

SVEC+4 = SVTEM+4

TYMR = TEMTYM

VSAV = VSAVTEM

Release interrupts

Proceed to address specified by R27END

STARTFIX Entered due to restart group 2.11 (set in "WOTNOW")

Set bit 7(FIXFLAG) of FIGWRD11 = 0

Proceed to address specified by R27END

ENDFIX Entered due to restart group 2.17 (set in "WUNZMOR")

Set bit 7(FIXFLAG) of FIGWRD11 = 1

If (FIXTIME - FIXTEMP) = 0:

FIXTIME = FIXTEMP - K_{4mins}

Proceed to address specified by R27END

C13STALL

If bit 4(Radar initiate) of channel 13 = 0:

Return (no VHF read in progress)

$TS_3 = \text{channel } 4$

$TS_3 = TS_3 + \text{RADTIME} + 2^{14}, \text{ modulo } 2^{14}$ (corrects for possible channel 4 overflow since RADTIME loaded, and gives "true" time difference)

$TS_4 = K_{90\text{mscalr}} + \text{RADDEL} - TS_3$

If $TS_4 \leq 0$:

Return

If $TS_4 + K_{\text{mdtscalr}} > 0$:

Return

Proceed to third line of "C13STALL" (a NOOP order used to avoid TC Trap)

Quantities in Computations

See also list of major variables and list of routines

- AGEOFW: Time of last W-matrix initialization for rendezvous measurement incorporation, scale factor B28, units centi-seconds. Set equal to MARKTIME in "REND5C", and can be displayed by N31. It is used in "AUTOW" logic.
- BVECTOR₀, BVECTOR₁: Vectors used as communication cells with "INCORP1" to provide geometry information associated with DELTAQ, i.e. relating the deviation in the measurement (a scalar) to the vector deviation in the state vector. The vectors normally are associated with position and velocity respectively, but for R22 BVECTOR₁ is always set 0. The scale factor of BVECTOR₀ is B1.
- C_{altvar}: Single precision erasable memory constant, program notation "ALTVAR", scale factor B-16, units rad², used in "REND7" (part of R22) if optics backup (R23) data is used, to specify information on the a priori accuracy of the backup optics data.
- C_{intvar}: Single precision erasable memory constant, program notation "INTVAR", scale factor B15, units meters², added to the variance computed in "REND7" (part of R22) for "integration variance", i.e. "square of expected integration position extrapolation error".
- C_{rmax}: Single precision erasable memory constant, program notation "RMAX", scale factor B19, units meters, giving the maximum value of position state vector change (as expressed in N49DISP) that is incorporated automatically by R22. Values larger than this constant cause a display for crew approval. If it is desired to generate a display for all measurement information, this can be accomplished by setting the constant to 77776₈ (one bit negative).
- C_{rtridot}: Erasable memory (double precision) constant, program notation "RTRIDOT", scale factor B-31, units meters/centi-second³, giving an a priori value of rate of change of acceleration used to degrade the W-matrix employed in R27. It is employed in "XTRAP" to compute PDFVL.
- C_{rvar}: Erasable memory (double precision) constant, program notation "RVAR", scale factor B-16, giving the "percentage error" (squared) in the VHF ranging measurement: for VHF marks, $(|RCLP| C_{rvar} + C_{intvar})$ is the value of VARIANCE (subject to C_{rvarmin} constraint). Actually would involve "percent"/100, of course.
- C_{rvarmin}: Triple precision erasable memory constant, program notation "RVARMIN", scale factor B40, units meters², giving the negative of the minimum value of VARIANCE for the VHF range measurements in R22 (negative is for coding convenience). If the most significant one-third if the constant is zero, then least significant two-thirds can be considered as a double precision number, scale factor B26.

C_{vmax} : Single precision erasable memory constant, program notation "VMAX", scale factor B7, units meters/centi-second, giving the maximum value of velocity state vector change (as expressed in N49DISP+2) that is incorporated automatically by R22. Values larger than this constant cause a display for crew approval.

CDUS: See Optics Computations.

CDUSNAP: See Orbital and Rendezvous Navigation.

CDUSNAPS, CDUSNAPT: See Orbital and Rendezvous Navigation.

CDUT: See Optics Computations.

COUNT3MK: See Minimum Key Rendezvous.

CSMPOS: Value of CSM position vector, scale factor B29, units meters.

DELQDA: Value of DELTAQ/TRIPA, variable scale factor, stored in push-down list location OD. If numerator and denominator are already normalized, scale factor is B-10.

DELTAQ: Value of deviation in measurement quantity, scale factor B29, units meters.

DELTAV_{cm}, DELTAV_{lm}: See Orbital Integration. Component overflows if exceeds 2^{22} meters.

DELTA_{X0}, DELTA_{X1}: State vector deviation estimates, scale factor (DELTA_{X0}) B29 in units of meters; for DELTA_{X1}, scale factor B7 in units of meters/centi-second. They are updates to position and velocity estimates of vehicle respectively.

DELVIMU: See Display Computations.

EGRESS: Single precision cell used for storage of return address information from "INCRP2" routine.

FIXTEMP: Temporary storage cell for FIXTIME, used for restart protection purposes.

FIXTIME: See Orbital and Rendezvous Navigation.

GAMMA: Quantity computed in "INCRP1", variable scale factor (scale factor information partially contained in cell NORMGAM). If NORMGAM = 0, scale factor of GAMMA equivalent to B-40. NORMGAM subsequently modified to reflect rescaling of Z_1 (for use in "INCRP2").

$K_{2\pi}$: Constant, program notation "PI/4.0", scale factor B3, value 0.785398164. Value corresponds to $2 \times 3.141592656 \times 2^{-3}$, or approximately $2\pi \times 2^{-3}$, where first term converts between angle measurements in units of revolutions and radians and second term is scale factor.

- K_{20sex}: Constant, program notation "20SEX", scale factor B28, units centi-seconds. Value is 2000×2^{-28} , corresponding to 20 seconds.
- K_{328nm}: Constant, program notation "328NM", scale factor B27, units meters. Value is 606863.36×2^{-27} (nominal), corresponding to $327.68 \times 1852 \times 2^{-27}$, where first term is value in nautical miles, second converts to meters, and third is scale factor. The nautical mile value is $2^{15} K_{\text{convrnge}}$ (since the 15-bit interface with radar provides data modulo that value).
- K_{4mins}: Constant, program notation "4MINS", scale factor B28, units centi-seconds. Value is 24000×2^{-28} , corresponding to 4 minutes.
- K_{60secdp}: Constant, program notation "60SECDP", scale factor B28, units centi-seconds. Value is 6000×2^{-28} , corresponding to 60 seconds: if T_{now} - VHFTIME is at least this amount, then an attempt can be made to obtain a new VHF mark in "REND3".
- K_{90mscalr}: Single precision constant, program notation "90MSCALR", scale factor B9, units centi-seconds. Value is 440_8 , or 9×2^{-9} , corresponding to 9 centi-seconds (time delay, when added to RADDEL, between initiation of radar read and completion of the shift pulses).
- K_{95secs}: Constant, program notation "95SECS", scale factor B28, units centi-seconds. Value is 9500×2^{-28} , corresponding to 95 seconds.
- K_{bottomri}: Constant, program notation "BOTTOMRI", scale factor B-17, units meters/centi-second², used in "TAKE2" to initialize WMAT₈. Value is $0.102E-5 \times 2^{17}$, corresponding to $0.102E-1 \times 10^{-4} \times 2^{17}$, where first term is value in meters/second² (about 1/30 fps²), second converts from seconds to centi-seconds, and third is scale factor.
- K_{convrnge}: Constant, program notation "CONVRNGE", scale factor B13, units of (meters/bit). Value is 18.52×2^{-13} , corresponding to a VHF ranging system bit weight of 18.52 meters (or 0.0100 nmi). A value of $(2^{14} K_{\text{convrnge}})$ corresponds to the weight of bit 15, normally the "sign", of RM.
- K_{dpm01}: See Burn Control.
- K_{ft999}: Constant, program notation "FT99999", scale factor B19, units meters. Value is 30479×2^{-19} , corresponding to $99996.72 \times 0.3048 \times 2^{-19}$, where first term is value in feet (display would be expected to show 99997 feet), second converts to meters, and third is scale factor.
- K_{imuvar}: Constant, program notation "IMUVAR", scale factor B-16, units radians². Value is $0.04E-6 \times 2^{16}$, corresponding to 0.04 mr^2 .

K_{mdtscalr} : Single precision constant, program notation "-DTSCALR", scale factor B9, units centi-seconds. Value is 77754_8 , corresponding to -23_8 , or -0.59375 centi-seconds (a measure of when the shift pulses are sent to the radar).

K_{middle} : Constant, program notation "MIDDLE", scale factor B12, units meters, used in "TAKE2" in the initialization of WMAT. Value is 12.93×2^{-12} , corresponding approximately to $\sqrt{2} \times 30 \times 0.3048 \times 2^{-12}$, where first term is an equation factor, second is VHF ranger variance (in feet), third converts to meters, and fourth is scale factor.

K_{sxtvar} : Constant, program notation "SXTVAR", scale factor B-16, units radians². Value is $0.04E-6 \times 2^{16}$, corresponding to 0.04 mr^2 .

K_{upperlef} : Constant, program notation "UPPERLEF", scale factor B12, units meters, used in "TAKE2" to initialize WMAT. Value is 6.47×2^{-12} , corresponding approximately to $(1/\sqrt{2}) \times 30 \times 0.3048 \times 2^{-12}$, where first term is an equation factor, second is VHF ranger variance (in feet), third converts to meters, and fourth is scale factor.

K_{vargap} : Constant, program notation "VARGAP", scale factor B22, units meters². Value is 83.6127×2^{-22} , corresponding approximately to $(30 \times 0.3048)^2 \times 2^{-22}$, where first term is VHF ranger variance converted to meters (i.e. $30'$) and second is scale factor.

K_{vhfmodlo} : Constant, program notation "VHFMODLO", scale factor B29, units meters. Value is 555600×2^{-29} , corresponding to $300 \times 1852 \times 2^{-29}$, where first term is value in nautical miles, second converts to meters, and third is scale factor.

LEMPOS: Value of OWS position vector, scale factor B29, units meters.

MARKDOWN+i (i = 0-6): Set of optics mark information (double precision time, CDU_y, optics shaft, CDU_z, optics trunnion, and CDU_x respectively) intended for downlink transmission in programs making use of optics data. Address is selected in "SXTSM" if STARIND = 0. For P51 - P54, this means the BESTI body being marked; in R22, since STARIND = 0 in "ROO", these cells are also loaded (due to performance of part of "SXTSM" via "GETUM"). MARKDOWN+7 is the same cell as RM.

MARKTIME: Cells used to contain time of mark information for R22, and as a communication cell with "SETINTG" (for which only the first 2 cells are employed, scale factor B28, units centi-seconds). Used in R22 as the first 2 cells of a 7-cell buffer to contain optics mark data being processed, as loaded from MRKBUF1 cells in "REND1".

MRKBUF1: See Optics Computations.

N49DISP: Magnitude of position state vector change for display in R1 of N49, scale factor B29, units meters. The magnitude is calculated as the square root of the sum of the squares of individual components of DELTAX₀, where the squares are carried triple precision. Hence if a given component is of insufficient magnitude to be retained triple precision when squared, that component will contribute 0 to the sum: since for triple precision scaled B0 the least increment is 2^{-42} , this means that for a component to have an input it must be at least 2^{-21} , i.e. a value of 2^8 (256) meters since scaling B29. Hence if all three components of DELTAX₀ are less than 256 meters (0.14 nmi), N49DISP will display zero (despite the fact that least increment of R1 of N49 is 0.01 nmi).

N49DISP+2: Magnitude of velocity state vector change for display in R2 of N49, scale factor B7, units meters/centi-second. Magnitude is calculated in a manner analogous to N49DISP, using DELTAX₁ instead of DELTAX₀. The least increment effect, however, is considerably less, amounting to only about 0.02 fps (display least increment is 0.1 fps).

N49DISP+4: Single precision cell, scale factor B14, used to display in R3 of N49 information on the source of the update generating the display. It is set to 1 in "REND7" for optics data and 2 for VHF ranging data.

NPASS: Single precision cell, scale factor B14, used to limit to 4 the number of tries in "CYCVHF" to obtain a change in the input range from that read in "R27GO".

OLDMKTME: Time of previous mark, scale factor B28, units centi-seconds, loaded in "REND12" and used in "AUTOW" logic.

OMEGA₀, OMEGA₁: Weighting vectors for measurement incorporation of the DELTAQ measurement (to give a "statistically optimum linear estimate of the deviation from the estimated state vector"). For no Z₁ rescaling, scale factors are B39 and B20 respectively.

OMEGAM₀, OMEGAM₁: Values of OMEGA₁ multiplied by GAMMA, used in "INCORP2" for updating W matrix information. Scale factor depends on scaling of GAMMA: if NORMGAM = 0 (see GAMMA), then scaling is B-1 and B-20 respectively. Program notations "OMEGAM1" and "OMEGAM2" respectively.

OPTMODES: See Optics Computations.

OPVEC: Value of range information displayed in R1 of N77, scale factor B20, units meters. It is used in "TAKE2" to retain the first range point when R27 is initialized, for subsequent initialization purposes. It is loaded in "WUNZMOR" with SVTEM after completion of the optimization interval (when bit 9, N77FLAG, of FIGWRD11 is also set 1), and it is also loaded in "STOR77" with SVTEM if N77FLAG = 0. It should be noted that the N77FLAG lockout on loading at end is not effective since "WUNZMOR" also increments FIXTIME, causing "TIMETASK" (entered each second) to detect a change and enter "N72STUFF", where bit reset 0 (barring special timing circumstances).

OPVEC+2: Value of range rate information displayed in R2 of N77, scale factor B6, units meters/centi-second. Loading history parallels that of OPVEC, except that initially used in "TAKE2" to store computer clock for first sample (B28 centi-seconds), yielding of course a non-meaningful "range rate" display.

PD2DTM: Value of TEMTYM sampled in "WUNZMOR", scale factor B28, units centi-seconds, used in "WOTNOW" to compute VHF.B+2 when start new optimization interval. It is stored in push-down list location 2D.

PD4DIQ: Value of measurement deviation (analogous to DELTAQ for R22) computed in "WUNZMOR", scale factor B12, units meters, stored in push-down list location 4D.

PDCVC: Value of "C vector" computed in "XTRAP", scaling (unless rescaled) B11: computations are such that the scaling of the quantity cancels as far as the final results are concerned. It is stored in push-down list location 0D.

PDDVL: Value of "D" computed in "XTRAP", scaling dependent on PDCVC, stored in push-down list location 6D. For no rescaling, would be scaled B-6.

PDFVL: Value of "F" computed in "XTRAP", scale factor B-32, stored in push-down list location 10D.

PDSVL: Value of "S" (square of magnitude of PDCVC) computed in "XTRAP", scaling B22 (unless rescaled), stored in push-down list location 8D.

PHETA: See Orbital and Rendezvous Navigation.

[PMATRIX]: Matrix computed in "XTRAP", program notation "P-MATRIX". First row is assigned same cells as VHF.B (loading of them shown in "XTRAP" for documentation convenience), scaling B1, B15, and B30 respectively. First element of 2nd row is zero, second is scaled B1, and third scaled B16. In third row, first 2 elements are zero, and third is scaled B1.

POINTEX: Single precision cell used to retain return address information from "WAITONE".

QPRET: See Orbital Integration.

QREG: Computer Q register (cell 0002_g), loaded with S-register information for computer TC orders, and also used in some cases for temporary storage of other information. As used in "RENDISP2", apparently expected to contain a positive non-zero number (as would be true for a TC order use). The actual sampling of QREG takes place in the program service routine "BANKJUMP", which is entered at the end of "ENDRET2" to return to the "PRIODSP" caller.

R_{other}: See Orbital Integration.

R27END: Single precision cell used to retain return address information from "R27GO".

R61CNTR: See Orbital and Rendezvous Navigation.

R63RANGE: Value of range estimate computed from vehicle state vectors loaded in "CRS61.1", scale factor B29, units meters. After loading, bit 9(EXTRANGE) of FLGWRD10 is set 1 to indicate that cell has valid data. It is used in "RANGERD7" to correct incoming range point to reflect interface-cell overflow (see K_{328nm}) if incoming and R63RANGE value differ by $K_{vhfmod10}$ or more.

RADDEL: Single precision time until the 100 pps pulse initiating radar read sequence (loaded in "RADSTART"), scale factor B9, units of centi-seconds.

RADTIME: Single precision storage for the complement of the contents of channel 4, scale factor B9, units centi-seconds, as sampled when bit 4 of channel 13 was set to initiate radar reading.

RANG1: Value of VHFRANGE sampled in "R27GO", scale factor B27, units meters. Subsequently replaced in "TAKE2" with the larger of the two range samples, and then rescaled to B20 meters for use subsequently. It is stored in push-down list location 2D.

RANG2: Value of VHFRANGE sampled in "CYCVHF", scale factor B27, units meters. It is stored in push-down list location 6D.

RBUF: Value of previous range sample saved in "VHFMOD", scale factor B27, units meters (loaded in "R27GO" for first pass). Also used for temporary storage purposes in "VHFMOD".

RCLP: Value of ($\text{LEMPOS} - \text{CSMPOS}$) for R22, scale factor B29, units meters.

RCV, RCV_{cm} , RCV_{lm} : See Orbital Integration.

RM: Single precision value of measured range sampled from RNRAD cell, scale factor B14, units counts (one VHF count is 0.01 nmi). The cell is the same as MARKDOWN+7 (for telemetry purposes).

RNRAD: Single precision special erasable memory cell 0046₈, into which VHF range data is shifted under hardware control if bits 4-1 of channel 13 are set to 1001₂. When shifting is complete, program interrupt #9 is generated, causing entrance to "VHFREAD".

RREADRET: Single precision cell used to retain return address information from "RANGERD". An "error" return transfers to the address in the cell, while a successful return transfers to calling address +2 (i.e. RREADRET contents incremented by 1).

SVEC: Value of "permanent" R27 filter range output, scale factor B20, units meters. It is loaded into SVTEM in "WUNZMOR", and loaded from SVTEM in "STOR77". It is displayed by R1 of N76.

SVEC+2: Value of "permanent" R27 filter range rate output, scale factor B6, units meters/centi-second. It is loaded into SVTEM+2 in "WUNZMOR", and loaded from SVTEM+2 in "STOR77". It is displayed by R2 of N76.

SVEC+4: Value of "permanent" R27 filter range acceleration output, scale factor B-9, units meters/centi-second². It is loaded into SVTEM+4 in "WUNZMOR", and loaded from SVTEM+4 in "STOR77".

SVTEM: Value of R27 working storage for filter range output, scale factor B20, units meters. See SVEC.

SVTEM+2: Value of R27 working storage for filter range rate output, scale factor B6, units meters/centi-second. See SVEC+2.

SVTEM+4: Value of R27 working storage for filter range acceleration output, scale factor B-9, units meters/centi-second². See SVEC+4.

T_{et}, TDELTA_V: See Orbital Integration.

TEMPOR1: Single precision cell, scale factor B14, used in the "RENDISP"/"RENDISP2" interface to determine when an answer to the 0649^{vn} display is received (when value becomes non-zero) and to conclude that this answer was PRO if the cell negative and that it was a recycle if the cell is positive (see QREG). This method is employed to avoid display conflicts (with e.g. R60 P20 display), and will cause the computer activity light to remain on until the display is answered.

TEMTYM: Value of R27 working storage for time of optimization (loaded with TYMR in "WUNZMOR" and into TYMR in "STOR77"), scale factor B28, units centi-seconds. Loaded with TIM1 in "WOTNOW" or "WUNZMOR" if optimization to FIXTIME not being done; otherwise loaded with FIXTIME. After being saved in PD2DTM, used in "WUNZMOR" for temporary storage purposes also.

TIM1: Time tag for RANG1, loaded initially in "R27GO" and subsequently written over with TIM2 for no range change in "CYCVHF". In "TAKE2" is set to the time tag used subsequently for the measurement. Scale factor is B28, units centi-seconds, and quantity stored in push-down list location OD.

TIM2: Time tag for RANG2, loaded in "CYCVHF", scale factor B28, units centi-seconds. It is stored in push-down list location 4D.

TNUV: See Orbital Integration.

TRIPA: Triple precision value of sum of VARIANCE and $|Z_1|^2$ computed in "INCORP1", scale factor B40. It is used as the denominator value for computing the necessary weighting functions (a rounded version is used to multiply VARIANCE).

TRKMKCNT: Single precision cell, scale factor B14, giving a count of the number of optics marks (either primary optics or via R23) that have been completely incorporated into the state vector in R22 (it is incremented in "REND12"). It is set 0 in "DOFSTART" and "REND5C" (when initialize W matrix for R22 use). Cell is the "least significant half" of VHFCNT, and can be displayed as the 2 least significant digits in R3 of N45.

TYMR: Value of "permanent" R27 value of TEMTYM, scale factor B28, units centi-seconds. See TEMTYM.

UCL: Value of unitRCLP computed in "REND7", scale factor B1 (a unit vector, of course).

UM: Value of unit measurement vector derived in "GETUM", scale factor B1: it gives the direction of the optics-measured information in reference coordinates.

USTAR: Unit "fictitious star direction", scale factor B1, updated in "BVECTORS" (initialized in "REND7"). Two such quantities are used, "chosen to be perpendicular to each other and to the current estimated line-of-sight vector."

V_{other} : See Orbital Integration.

VARIANCE: Triple precision variance associated with the navigation measurement reflected in DELTAQ, scale factor B40, units of meters². The value for use in "INCORP1" is computed by the calling program based on a priori measurement errors.

VCV, VCV_{cm} : See Orbital Integration.

VHF.B (components VHF.B+0, VHF.B+2, VHF.B+4): Propagation vector for R27, components scaled B1, B15, and B30 respectively. Loaded with 1, Δt , and $\frac{1}{2} \Delta t^2$ respectively, where Δt is the time interval required for extrapolation (e.g. TIM1 - TEMTYM in "WUNZMOR"), centi-seconds.

VHF.OMEG: Weighting vector used in R27 computations ("WUNZMOR"). VHF.OMEG (first component) initially used for retention of raw measurement, B20 meters. When initially computed from [WMAT], components B23, B9, B-6 respectively. When used to update SVTEM, components scaled B3, B-11, B-26 respectively, as they are when updating [WMAT], for no rescaling.

[VHF.W]: Value of "permanent" R27 weighting matrix, loaded into [WMAT] in "WUNZMOR" and from [WMAT] in "STOR77". See [WMAT].

VHF.Z: R27 measurement incorporation vector, scale factor B11, meters.

VHFCNT: Single precision cell, scale factor B14, giving a count of the number of VHF range marks that have been incorporated into the state vector in R22 (it is incremented in "REND12"). It is reset to 0 at the points listed for TRKMKCNT (which is the "least significant half" of VHFCNT), and it can be displayed as the 2 most significant digits in R3 of N45.

VHFRANGE: Value of incoming VHF range measurement (a scaled version of the 15-bit integer in RM), scale factor B27, computed in "RANGERD1" and corrected for overflow of the data interface (see K_{328nm}) in "RANGERD7".

VHF TIME: Value of computer clock when last VHF range sample for R22 made, scale factor B28, units centi-seconds. It is initialized to present computer clock in "R22", and if at least $K_{60secdp}$ have elapsed since it was last loaded (see "REND3" logic), then another sample attempt for R22 is initiated. It is set to MARKTIME in "R22BAD" (sample not successful) or "R22GOOD" (sample successful): MARKTIME is set to T_{now} in "RADSTART" (just after setting channel 13 bits to initiate sample, but after releasing interrupts that were inhibited during channel 13 setting).

VSAV: Value of "permanent" R27 value of VSAVTEM, scale factor B7, units meters/centi-second, loaded into VSAVTEM in "WUNZMOR" and from VSAVTEM in "STOR77".

VSAVTEM: Value of DELVIMU^x sampled in "TAKE2" or "WUNZMOR", used to update SVTEM+2 during R27 performance with P48, scaled B7 meters/centi-second.

$[W_i]$, W_i , \underline{W}_i : Notations used to describe portions of the "W" (error transition) matrix used in R22 and stored in the computer memory. The matrix as defined for R22 is made up of four separate 3x3 (each double precision) matrices, a total of 36 elements. In former programs, the matrix was 9x9: to minimize the impact of changes, the last three rows and last 3 columns have been deleted, but the memory cell assignments have not otherwise been modified. This means that the first two matrix portions ($[W_0]$ and $[W_1]$) occupy cells W_0 through W_{17} (indices are B13), and the remaining two matrix portions ($[W_3]$ and $[W_4]$) occupy cells W_{27} through W_{44} . Note that computations carried out by vector operations on 3 elements at a time are designated by the usual vector notation with the subscript being that of the first element involved: \underline{W}_3 , for example, is the "vector" made up of (W_3, W_4, W_5) .

Initialization of the W matrix involves initializing the diagonal elements of $[W_0]$ with the appropriate position uncertainty information and the diagonal elements of $[W_4]$ with the appropriate velocity uncertainty information. The scale factor of $[W_0]$ and $[W_1]$ elements are all B19 in units of meters; for $[W_3]$ and $[W_4]$, the scaling is B0 in units of meters/centi-second.

[WMAT]: Weighting matrix used in R27 computations. Elements of first(WMAT₀₋₂) row are scaled B12; of second row are scaled B-2; and of third row are scaled B-17 (units meters, meters/centi-second, and meters/centi-second² respectively). Matrix initialized in "TAKE2". See [VHF.W].

WRENDPOS: Single precision value of W-matrix initialization for R22 position computations, scale factor B19, units meters. Can be updated by "V67CALL", and would be expected to form part of the prelaunch load. It is also loaded (if final pass and in minimum key rendezvous mode) in "VN1645" with K_{posvel2} (see Burn Computations), and is used in "REND5C".

WRENDVEL: Single precision value of W-matrix initialization for R22 velocity computations, scale factor B0, units meters/centi-second. Can be updated by "V67CALL", and would be expected to form part of the prelaunch load. It is also loaded (if final pass and in minimum key rendezvous mode) in "VN1645" with K_{posvel3} (see Burn Computations), and is used in "REND5C".

WVOPT: Single precision cell, scale factor B14, used in "V67CALL" to specify whether W-matrix initialization elements are to be changed. The value when displayed initially in R3 of N99 is 0: a non-zero value after PRO to 0699_{vn} causes WRENDPOS and WRENDVEL to be changed.

WWPOS: Quantity displayed in R1 of N99, scale factor B19, units meters. It provides the present W-matrix RSS position error. If the loading option is selected, is used as source for WRENDPOS information.

WVVEL: Quantity displayed in R2 of N99, scale factor B0, units meters/centi-second. It provides the present W-matrix RSS velocity error. If the loading option is selected, is used as source for WRENDVEL information.

Z₀, Z₁: Measurement vector information, scale factor B20, units meters, computed at the start of "INCORP1". Subsequently rescaled.