An Overview of a DEC-based Satellite Ranging System

by

David W. Morrison
Drews Systems, Inc.
599 North Matilda Avenue
Suite 245
Sunnyvale, CA 94086 USA

1 THE PDP-11/45 COMPUTER AND RSX-11M

SEVERAL FEATURES OF THE DEC COMPUTER AND ITS OPERATING SYSTEM HAVE STRONGLY SHAPED THE SATELLITE RANGING SYSTEM (SRS) LOCATED IN WETTZELL, BUNDESREPUBLIK DEUTSCHLAND. THE FIRST SECTION DISCUSSES THOSE FEATURES, THE SECOND DESCRIBES SRS ITSELF, AND THE THIRD DEALS WITH THE DESIGN AND CODING METHODS THAT WERE USED IN THE CREATION OF SRS.

1.1 THE EQUIPMENT

FIGURE 1.1-1 SHOWS THE CURRENT AND ORIGINAL COMPUTER CONFIGURATIONS IN WETTZELL.

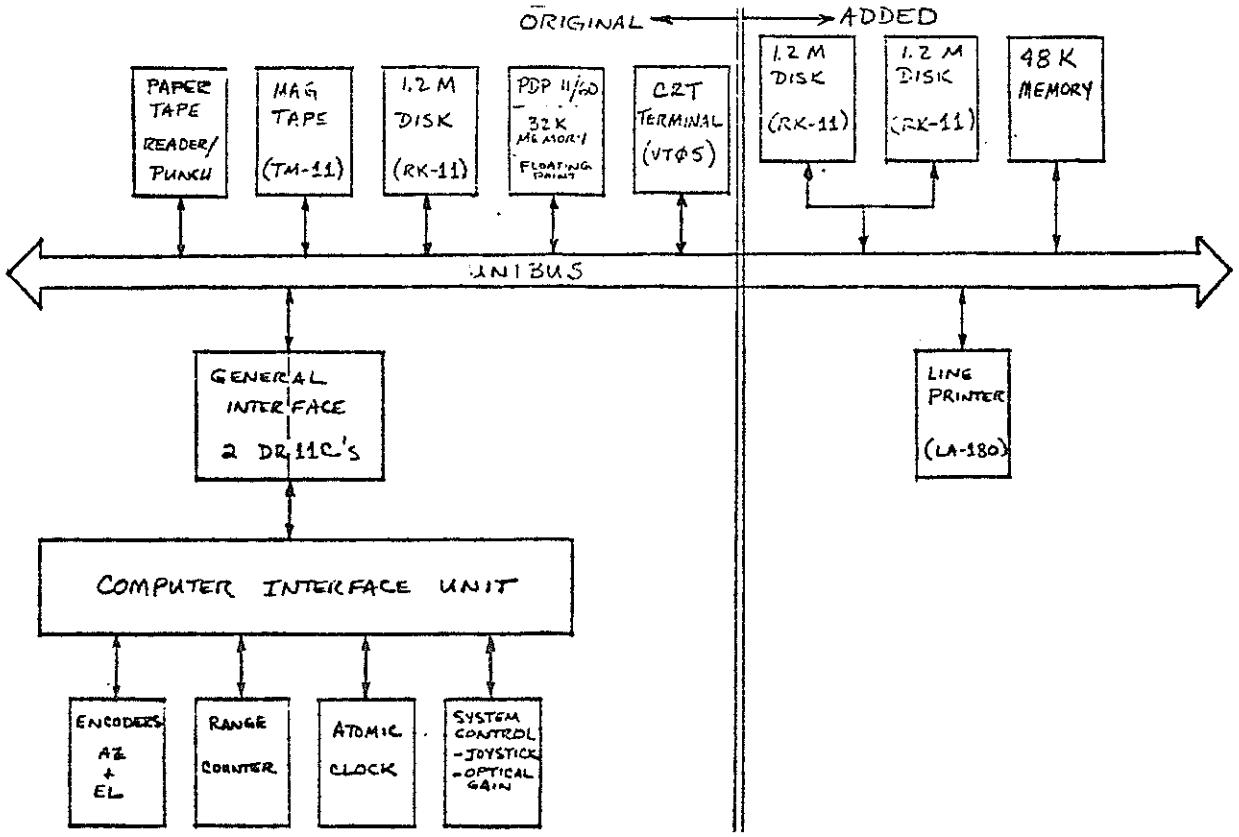


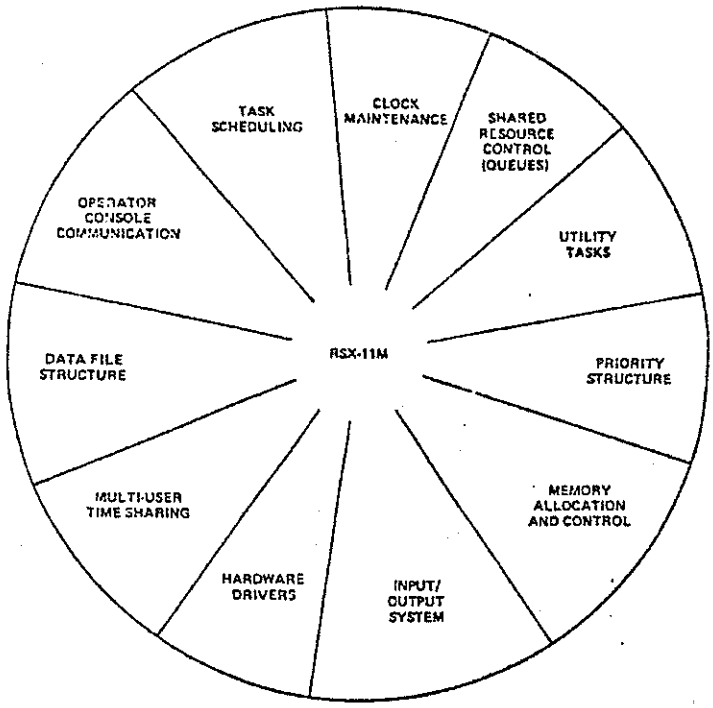
Figure 1.1-1

THE FACILITY HAS BEEN SIGNIFICANTLY EXPANDED SINCE THE ORIGINAL SRS WAS WRITTEN PARTICULARLY BY ADDING MEMORY, DISK STORAGE, AND A FAST LINE PRINTER.

1.2 RSX-11M--THE OPERATING SYSTEM

FIGURE 1.2-1 DEPICTS THE MAJOR FEATURES OF THE RSX-11M OPERATING SYSTEM.

MOST PLAYED A ROLE IN SRS DEVELOPMENT. SEVERAL ARE DISCUSSED IN SUBSEQUENT ARTICLES.



RSX-11M OPERATING SYSTEM

NOTE:
RSX PROVIDES THE FUNCTIONS SHOWN IN THE DIAGRAM. APPLICATION PROGRAMS MAKE "EXECUTIVE" CALLS TO COMMUNICATE WITH RSX AND INVOKE ONE OR MORE OF THESE FUNCTIONS.

Figure 1.2-1

1.3 PROGRAM DEVELOPMENT UTILITIES

FIGURE 1.3-1 SUMMARIZES THE RELATIONSHIPS BETWEEN THE UTILITY PROGRAMS, COMPILER, AND ASSEMBLER WHICH ARE USED TO CREATE SOURCE TEXT, OBJECT FILES, LIBRARIES, AND EXECUTABLE PROGRAMS.

ANY FILE CAN BE COPIED, DELETED, MERGED OR LISTED USING PIP (PERIPHERAL INTERCHANGE PROGRAM.) PIP IS FOUND IN SEVERAL DEC SYSTEMS. IT IS NOT PART OF THE I/O SYSTEM (CALLED FILE CONTROL SERVICES) THAT IS AVAILABLE TO INDIVIDUAL PROGRAMS. IT GENERALLY PROVIDES WAYS FOR MANIPULATING FILES AS A WHOLE AS OPPOSED TO MAKING CHANGES WITHIN THEM.

- EDI AND EDT ARE TWO TEXT EDITORS THAT PROVIDE THE NORMAL MEANS OF SOURCE ENTRY AND MAINTENANCE.

- F4P IS A SEPARATELY AVAILABLE FORTRAN COMPILER WHICH GENERATES EFFICIENT, INLINE CODE.

- MAC IS THE ASSEMBLER FOR THE DEC ASSEMBLY LANGUAGE CALLED MACRO.

- TKB BUILDS EXECUTABLE PROGRAM IMAGES (CALLED TASKS) FROM RELOCATABLE

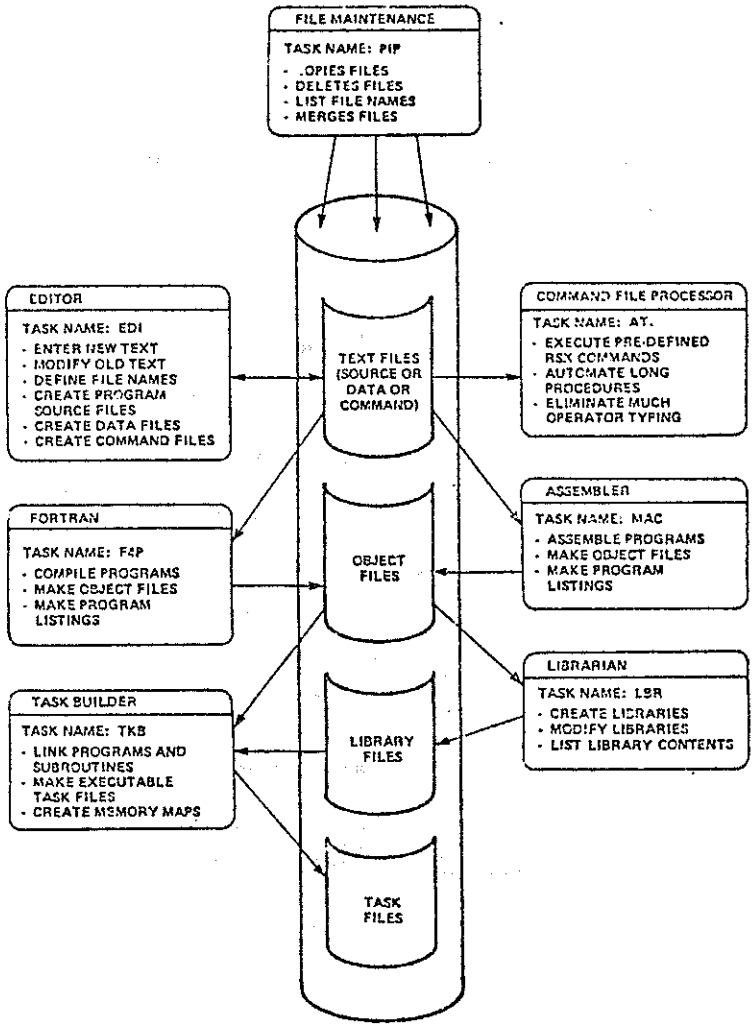


Figure 1.3-1

OBJECT FILES AND OBJECT MODULE LIBRARIES.

- LBR IS THE LIBRARIAN UTILITY USED TO CREATE AND MAINTAIN OBJECT MODULE LIBRARIES.

- AT. IS A COMMAND LANGUAGE UTILITY THAT ALLOWS LENGTHY CONTROL SEQUENCES TO BE PRE-DEFINED IN FILES. THE RSX KEYBOARD MONITOR ACCEPTS SUCH FILES AS DO ALL THE MAJOR UTILITY PROGRAMS. MUCH OPERATOR TYPING AND TIME IS SAVED THROUGH THE USE OF AT. SYSTEM-WIDE PROCEDURAL STANDARDS CAN ALSO BE MORE EASILY IMPOSED.

1.4 EVENT FLAGS

EVENT FLAGS ARE ONE OF THE MAIN SYNCHRONIZING AGENTS IN THE RSX SYSTEM. THEY CAN BE ASSOCIATED WITH AN ASYNCHRONOUS EVENT (FOR EXAMPLE, THE END OF A TIME INTERVAL.) WHEN THE EVENT OCCURS, THE FLAG'S STATUS GENERALLY CHANGES FROM "CLEAR" TO "SET". EVENT FLAGS CAN ALSO BE SET AND CLEARED BY TASKS AS GATING INDICATORS. EACH TASK HAS A LOCAL SET OF EVENT FLAGS WHILE ANOTHER

GROUP IS COMMON TO ALL TASKS.

EVENT FLAGS ARE MAINTAINED BY THE RSX EXECUTIVE VIA EXECUTIVE REQUEST CALLS FROM THE APPLICATION TASKS. CAPABILITIES INCLUDE SETTING, CLEARING, AND TESTING THE STATE OF AN ARBITRARY EVENT FLAG; WAITING FOR AN EVENT FLAG OR FLAGS TO BE SET; AND ASSOCIATING AN ARBITRARY EVENT FLAG WITH CERTAIN EVENTS. FIGURE 1.4-1 SUMMARIZES THE CAPABILITIES.

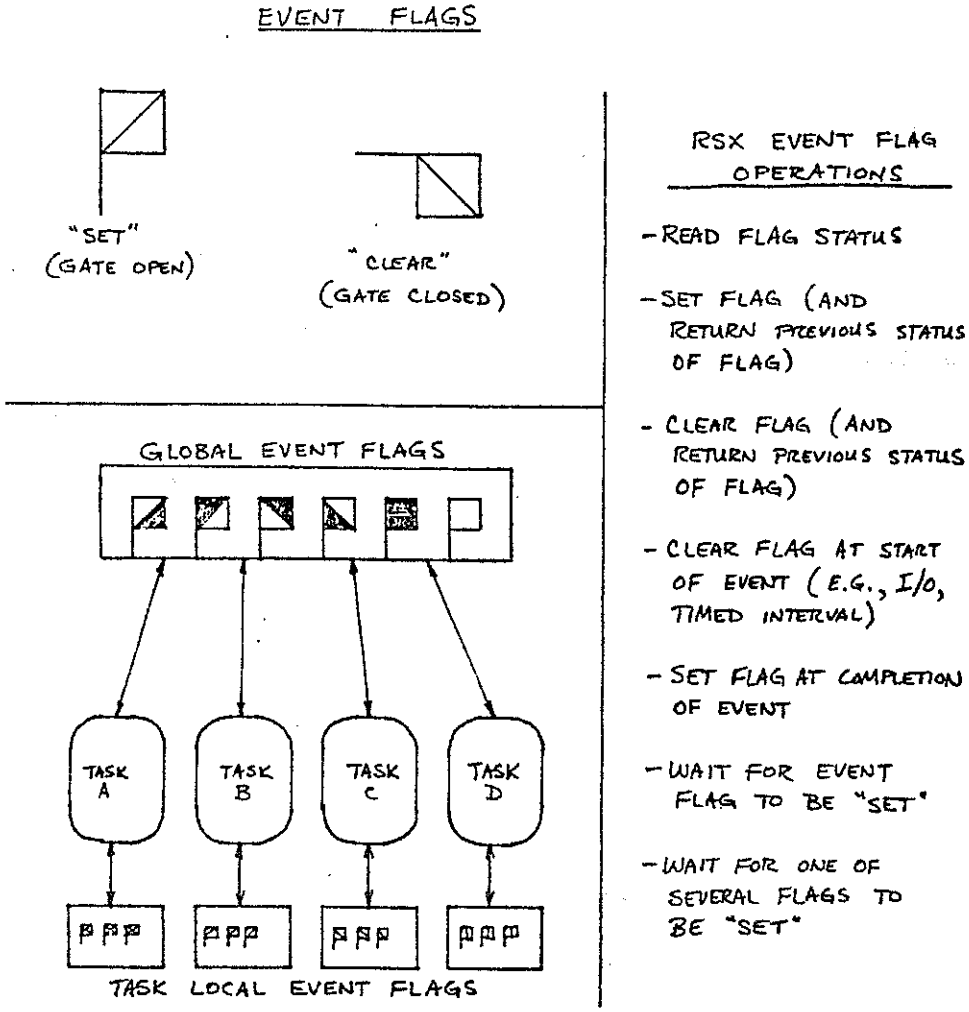


Figure 1.4-1

EVENT FLAGS ARE OFTEN USED TO SYNCHRONIZE INPUT/OUTPUT OPERATIONS WITH EXECUTION OF THE REQUESTING TASK. THE I/O CAN PROCEED IN PARALLEL WITH TASK OPERATION. THE EVENT FLAG IS SET BY THE RSX EXECUTIVE WHEN THE I/O COMPLETES. THE TASK CAN CHECK THE EVENT FLAG STATUS WHENEVER AND WHEREVER THE SYNCHRONIZATION MUST OCCUR. IT IS ALSO POSSIBLE TO CHECK THE FLAG'S STATUS WITHOUT SYNCHRONIZING.

A SECOND COMMON USE FOR EVENT FLAGS IS TO GATE ACCESS TO COMMON DATA BASES. USUALLY A CLEAR FLAG MEANS THE DATA IS WRITE-LOCKED WHILE SET MEANS THE DATA IS READ-WRITE ENABLED. COMPETING TASKS WAIT FOR THE EVENT FLAG TO BE SET (THE TASK USUALLY RELINQUISHES THE CPU WHILE IT WAITS), THEN TRY TO CLEAR

THE FLAG AND CHECK ITS PREVIOUS STATE. IF THE FLAG WAS PREVIOUSLY CLEAR THEN ANOTHER, HIGHER PRIORITY TASK WO' THE RACE AND LOCKED THE DATA BASE FIRST. THE LOWER PRIORITY REQUESTER MUST AGAIN WAIT FOR THE SET CONDITION AND THEN REPEAT THIS ACCESSING PROCESS.

EVENT FLAGS ARE USED BY SRS IN ALL THE WAYS MENTIONED HERE. FOR EXAMPLE, DURING THE TRACKING OPERATION EVENT FLAGS ARE USED BY

1. THE CRT TASK TO REQUEST MORE DISPLAY DATA FROM THE TRACKING TASK;
2. ALL REAL-TIME TASKS TO CHECK FOR THE END-OF-TRACK CONDITION;
3. THE CRT TASK TO SYNCHRONIZE ITS OUTPUT TO THE OPERATOR CONSOLE;
4. THE TRACKING TASK TO SYNCHRONIZE ITS READING OF PERIODIC DATA.

1.5 I/O REQUESTS AND DATA TRANSFERS

INPUT AND OUTPUT IN THE RSX SYSTEM CAN BE FULLY ASYNCHRONOUS RELATIVE TO THE REQUESTING TASK'S EXECUTION. THE PRINCIPAL I/O INITIATOR ROUTINE FOR RSX IS NAMED QIO. FIGURE 1.5-1 SHOWS THE INITIATION OF AN I/O REQUEST BEGINNING WITH THE QIO CALL IN THE APPLICATION TASK.

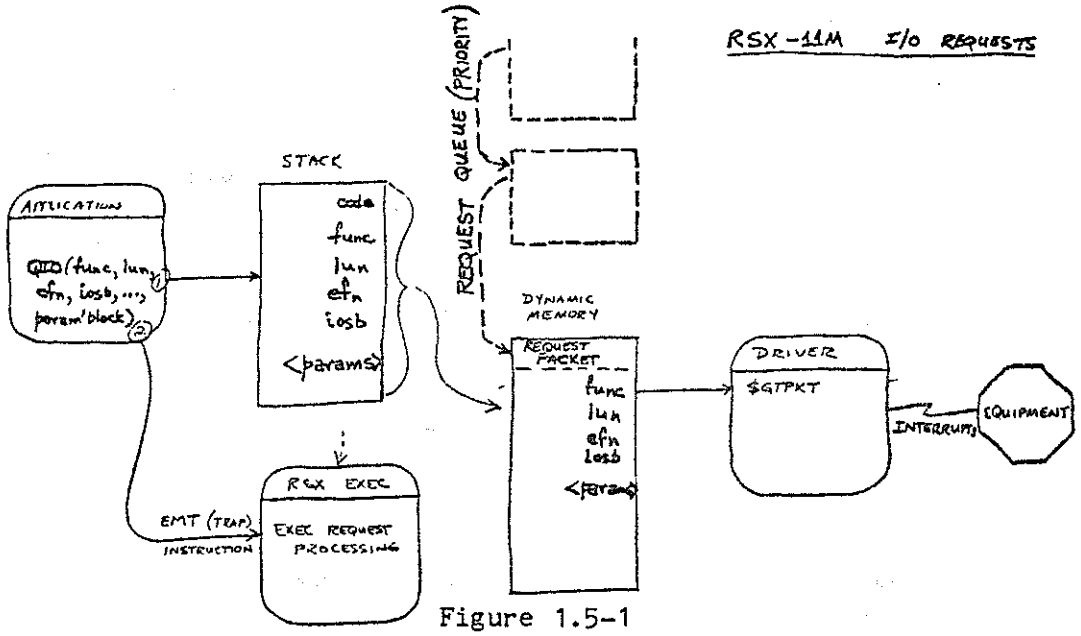


Figure 1.5-1

QIO PLACES THE CALLING PARAMETERS ON THE STACK AND EXECUTES A TRAP INSTRUCTION WHICH BEGINS RSX EXECUTIVE EXECUTION. THE EXECUTIVE PLACES THE PARAMETERS IN A QUEUE AND CALLS THE DRIVER. THE DRIVER READS THE QUEUE AND BEGINS INTERFACING TO THE APPROPRIATE EQUIPMENT.

FIGURE 1.5-2 SHOWS TRANSFERRAL OF DATA IN RESPONSE TO A QIO CALL. THE DRIVER NORMALLY WRITES DATA DIRECTLY TO (OR READS DATA DIRECTLY FROM) THE APPLICATION BUFFER ALTHOUGH IT IS ALSO POSSIBLE TO TRANSFER DATA THROUGH

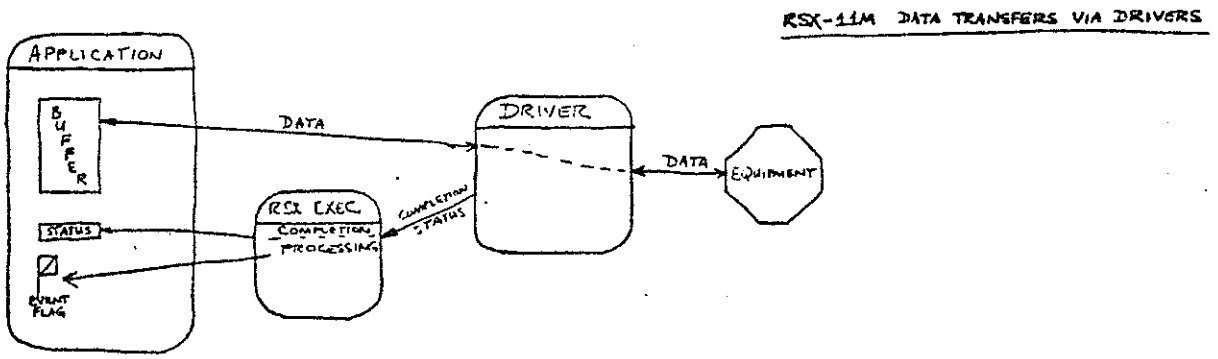


Figure 1.5-2

DYNAMIC MEMORY THUS ALLOWING THE APPLICATION TO BE NON-MEMORY RESIDENT DURING THE BUFFER-HARDWARE TRANSFER. THE RSX EXECUTIVE IS NOTIFIED BY THE DRIVER WHEN I/O COMPLETES. IT, IN TURN, SETS SYNCHRONIZING EVENT FLAGS AS REQUIRED AND MAKES THE APPLICATION TASK ELIGIBLE FOR EXECUTION.

FIGURE 1.5-3 DEPICTS THE STANDARD RSX EQUIPMENT HANDLER ORGANIZATION.

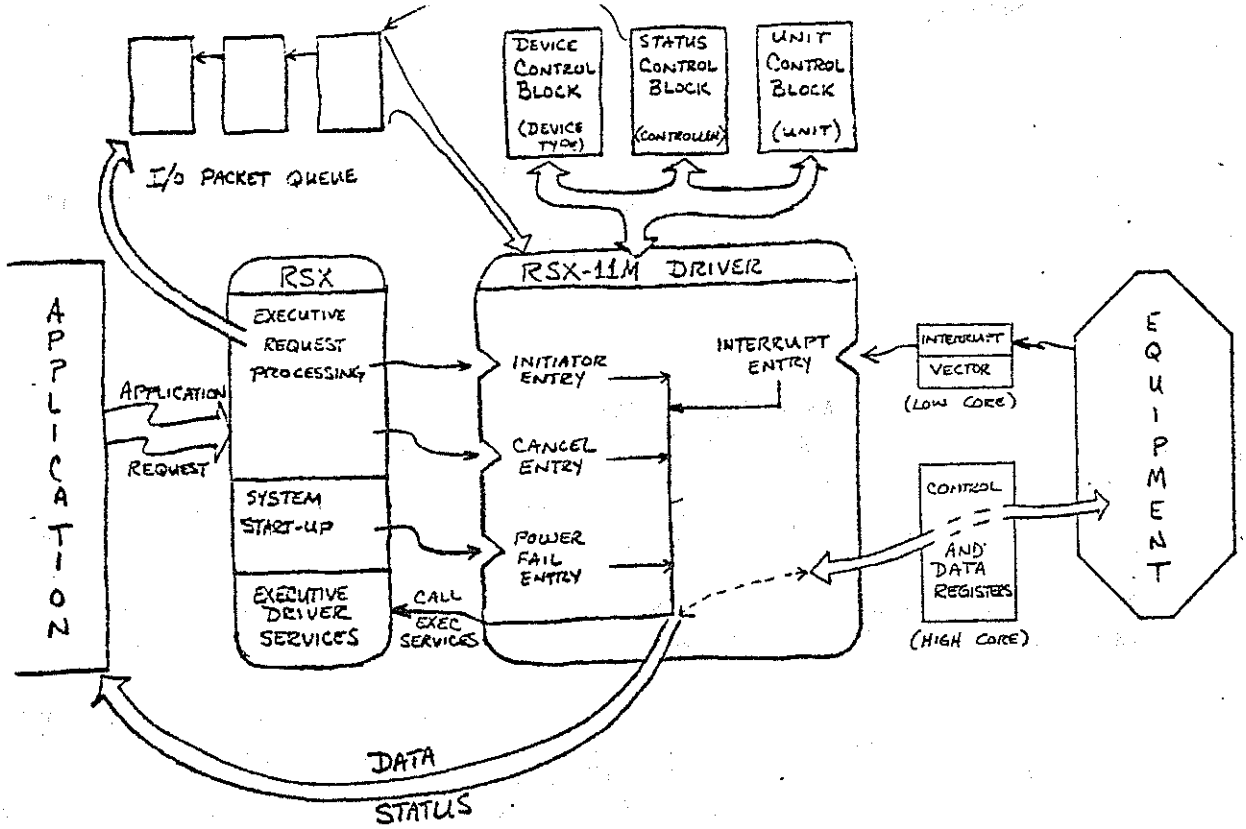


Figure 1.5-3

ESSENTIAL POINTS INCLUDE

1. THE EXECUTIVE REQUEST PROCESSING WHICH IS THE INTERFACE BETWEEN THE APPLICATION AND THE DRIVER-LEVEL SOFTWARE,

2. THE DEVICE CONTROL BLOCK (DCB), THE STATUS CONTROL BLOCK (SCB) AND THE UNIT CONTROL BLOCK (UCB) WHICH REFLECT THE STATUS OF THE HARDWARE,

3. THE I/O PACKET QUEUE WHICH CONTAINS PRIORITY-ORDERED REQUESTS FOR DRIVER ACTION,

4. THE INITIATOR ENTRY POINT WHERE THE EXECUTIVE STARTS DRIVER EXECUTION,

5. THE INTERRUPT ENTRY POINT WHERE HARDWARE INTERRUPTS ARE PROCESSED,

6. THE CONTROL AND DATA REGISTERS WHERE THE FUNCTION CODES, DATA, AND STATUSES ACTUALLY CHANGE HANDS BETWEEN COMPUTER AND EXTERNAL HARDWARE,

7. THE INTERRUPT VECTOR WHICH GUIDES TRANSFER OF COMPUTER CONTROL WHEN AN INTERRUPT OCCURS,

8. THE EXECUTIVE DRIVER SERVICES WHICH PERFORM STANDARD FUNCTIONS FOR THE DRIVER,

9. THE TRANSFERRAL OF DATA AND STATUS INFORMATION BETWEEN APPLICATION AND DRIVER.

SRS MAKES EXTENSIVE USE OF THIS I/O SYSTEM PARTICULARLY THROUGH THE COMPUTER INTERFACE UNIT DRIVER PROGRAM (CUDRV) WHICH IS THE MAIN INTERFACE ROUTINE TO THE TRACKING HARDWARE. CUDRV IS DISCUSSED FURTHER IN SECTION 2.

1.6 FILE STRUCTURE

RSX SUPPORTS A REASONABLY EXTENSIVE FILE STRUCTURE ON DISKS. THE ENTIRE DISK IS MAPPED FOR BLOCK ALLOCATION TO NAMED FILES. FILES MAY BE SEQUENTIAL OR RANDOM, FORMATTED OR BINARY, OR ANY COMBINATION. FILES ARE IDENTIFIED WITH A 1-9 CHARACTER NAME, A 3-CHARACTER TYPE, AND A VERSION NUMBER. FILES ARE STORED IN DIRECTORIES WHICH ARE WRITE LOCKED TO ALL USERS EXCEPT THOSE WHO HAVE ENTERED THE PROPER DIRECTORY NUMBER IN ANY OF SEVERAL WAYS. RSX SUPPORTS FURTHER FILE PROTECTIONS AND MULTI-USER FEATURES, BUT BECAUSE THE SRS ENVIRONMENT IS "FRIENDLY," NO USE IS MADE OF THEM.

FILES MAY BE OPENED AT ANY OF SEVERAL LOGICAL LEVELS. FORTRAN PROVIDES AN OPEN STATEMENT AT THE HIGHEST LEVEL. THE FILE CONTROL SERVICES (FCS) PACKAGE PROVIDES MORE FLEXIBILITY AT THE ASSEMBLY LANGUAGE LEVEL. AT THE LOWEST, STEP-BY-STEP LEVEL THE FILE CONTROL PRIMITIVES SUPPORT NOT ONLY FILE ACCESSES BUT FILE DIRECTORY MANIPULATIONS AS WELL.

SRS USES ALL LEVELS OF FILE ACCESS. FORTRAN IS USED FOR BOTH SEQUENTIAL AND RANDOM ACCESS WHENEVER ITS LIMITED ALTERNATIVES AND RELATIVELY SLOW SPEED CAN BE TOLERATED. FCS PROVIDES OCCASIONAL SPECIAL ACCESS (USUALLY IN THE FORM OF A FORTRAN-CALLABLE ASSEMBLY LANGUAGE ROUTINE.) FINALLY, SOME PRIMITIVES ARE USED BY THE TRACKING TASK TO ACHIEVE MAXIMUM SPEED AND PARTICULARLY TO

AVOID THE MEMORY COST OF THE FORTRAN AND FCS EXECUTION TIME ROUTINES.

1.7 FORTRAN COMPILER OBJECT CODE

THE FORTRAN-IV-PLUS COMPILER (F4P) GENERATES INLINE, MACHINE LANGUAGE CODE. MANY OTHER MINICOMPUTER FORTRANS GENERATE "THREADED" CODE; THAT IS, A SERIES OF CALLS TO EXECUTION-TIME ROUTINES FOR ALL OPERATIONS NO MATTER HOW SIMPLE. THREADED CODE ALLOWS FAST COMPILE TIMES BUT IS MUCH TOO SLOW FOR MOST REAL-TIME APPLICATIONS. THUS THE INLINE CODE OF F4P IS AN ESSENTIAL FEATURE OF RSX WHICH HAS SIGNIFICANTLY AIDED THE CODING, IMPLEMENTATION, AND MAINTENANCE OF THE SRS SYSTEM.

FIGURE 1.7-1 IS A PARTICULARLY IMPRESSIVE EXAMPLE OF F4P'S CODE QUALITY, IN THIS CASE INVOLVING TRIPLE SUBSCRIPTING.

NOTE:
given: INTEGER X(A,B,C)
then: $X(I,j,k) = locX + 2[A(j+Bk) - A(1+B) - (I-1)]$

```
      INTEGER TAPEBF(36,8,2)
C
      TAPEBF(PSTATS,NXSEPT,CRNTPT)=
     *      IAND(TAPEBF(PSTATS,NXSEPT,CRNTPT),'77777)
      CALL UPDATN(TAPEBF(PSTATS,NXSEPT,CRNTPT),
     !            TAPEBF(PCVPWR,NXSEPT,CRNTPT)
```

```
MOV     CRNTPT,R0
ASH     #3,R0          ; Bk ... NOTE USE OF SHIFT TO MULTIPLY - BY POWER OF 2.
ADD     NXSEPT,R0      ; j+Bk
MOV     R0,R1
MUL     #110,R1        ; 2[A(j+Bk)]
BIC     #100000,TAPEBF-1210(R1) ; R1 = 2[A(1+B)+I-1]
                       ; 0043
MOV     R1,$IDATA+12
ADD     #TAPEBF-1210,$IDATA+12 ; loc(ARG1)
ADD     #TAPEBF-1140,R1
MOV     R1,$IDATA+14   ; loc(ARG2)
MOV     #$IDATA+10,R5
JSR     PC,UPDATN      ; CALL UPDATN (---)
```

Figure 1.7-1

1.8 OVERLAYS

THE RSX TASK BUILDER AND RUN-TIME LOADER SUPPORT PROGRAM OVERLAY STRUCTURES. AT THE EXPENSE OF EXECUTION SPEED, OVERLAYS ALLOW VERY LARGE

PROGRAMS TO BE EXECUTED IN RELATIVELY SMALL MEMORY AREAS WITHOUT THE NEED FOR BREAKING THE APPLICATION INTO MANY SMALL, INTERACTING TASKS. BECAUSE SRS WAS ORIGINALLY WRITTEN FOR AN APPLICATION MEMORY SPACE OF ONLY 18,000 WORDS, THIS FEATURE WAS EXTENSIVELY USED IN SEVERAL TASKS.

RSX SUPPORTS TWO METHODS OF OVERLAY LOADING--MANUAL AND AUTOMATIC. THE FIRST IS FASTER BUT REQUIRES EXPLICIT CODE IN THE PROGRAM TO LOAD EACH OVERLAY. THE SECOND IS SLOWER BUT REQUIRES NO KNOWLEDGE OF THE OVERLAY STRUCTURES IN THE PROGRAMS. THE FIRST LOADING METHOD IS USED BY THE SRS TRACKING TASKS, THE SECOND BY MOST OTHER SRS TASKS WHICH USE OVERLAYS.

1.9 MEMORY MAPPING REGISTERS

THE DEC HARDWARE USES EIGHT MEMORY MAPPING REGISTERS TO MAP UP TO 32K WORDS OF LOGICAL ADDRESS SPACE ONTO ACTUAL PHYSICAL MEMORY (FIGURE 1.9-1.)

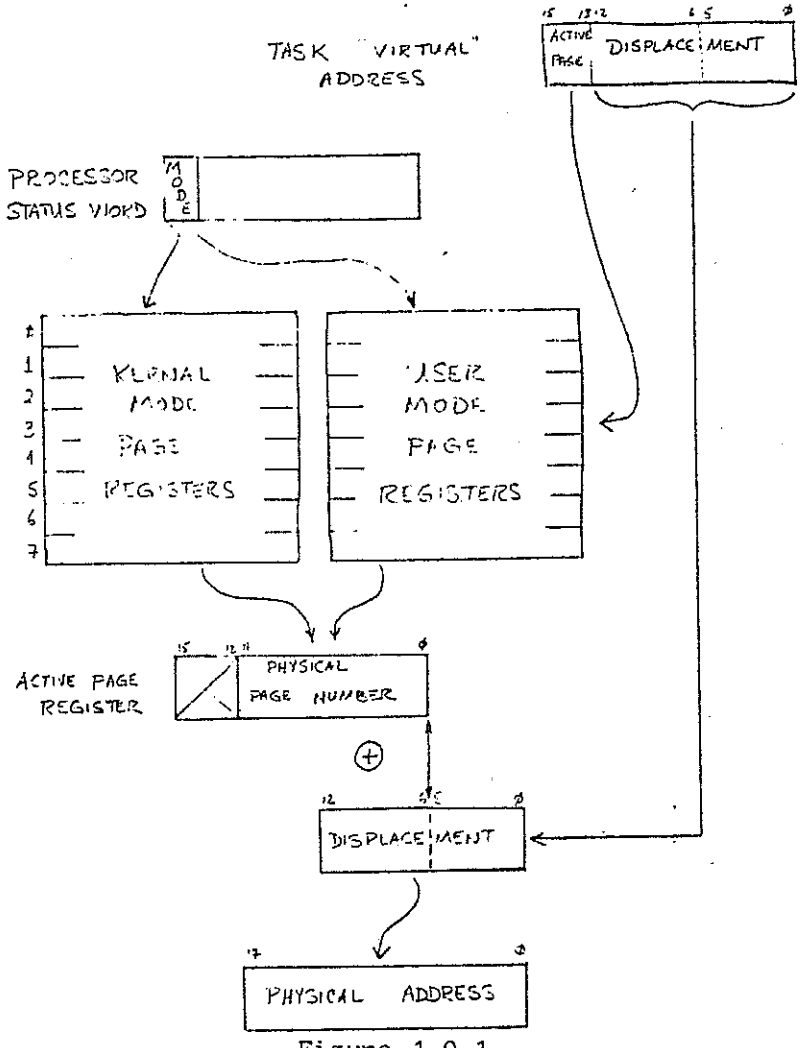


Figure 1.9-1

THIS MEANS THAT LOGICAL MEMORY CAN ONLY BE ALLOCATED IN 4K-WORD BLOCKS. THIS RELATIVELY LARGE INCREMENT IS CONFINING IN SOME CASES AND WASTEFUL OF LOGICAL MEMORY SPACE IN OTHERS.

EXAMPLES: (1) THE GLOBAL DATA AREA IS ABOUT 2.5K WORDS LONG WHICH MEANS THAT, BECAUSE MEMORY CAN ONLY BE ALLOCATED IN 4K BLOCKS, ABOUT 1.5K WORDS OF LOGICAL ADDRESS SPACE IS LOST IN EVERY TASK WHICH LINKS TO THIS AREA; (2) FOR THIS REASON THE DIAGNOSTIC MODE TASK MUST BE BUILT IN A SPECIAL WAY TO AVOID LINKING TO THE GLOBAL AREA.

2 SRS ORGANIZATION

THE SRS SYSTEM IS BROKEN INTO SEVERAL RELATED UNITS CALLED "MODES". EACH MODE IS IMPLEMENTED AS A SEPARATE PROGRAM CALLED A "TASK". FIGURE 2.0-1 DEPICTS THE RELATIONSHIPS BETWEEN ALL THE TASKS.

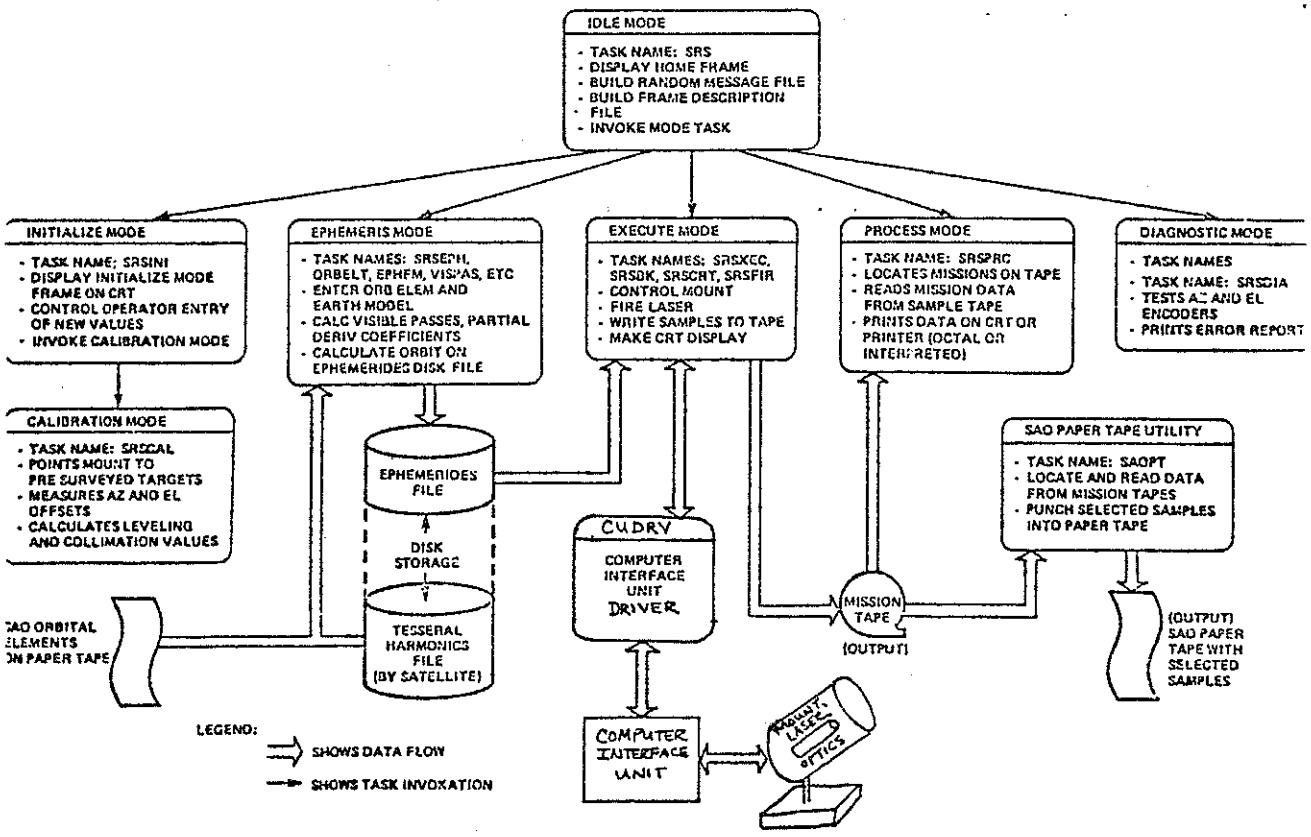


Figure 2.0-1

FURTHER ARTICLES IN THIS SECTION SUMMARIZE EACH MODE PLUS RELATED UTILITIES AND THE COMPUTER INTERFACE UNIT DRIVER PROGRAM.

2.1 IDLE MODE

- PURPOSE:

1. ALLOW THE OPERATOR TO SELECT A MAJOR OPERATING MODE,

2. WHEN NECESSARY, READ THE MESSAGE FILE AND CREATE THE RANDOM MESSAGE FILE AND THE FR..ME DESCRIPTOR FILE.

- COMMENTS:

* THE IDLE MODE TASK--NAMED SRS--IS CALLED WHEN THE OPERATOR TYPES "RUN SRS". WHENEVER THIS CALL IS MADE, A MAJOR SUBFUNCTION OF THE IDLE MODE NAMED THE "MESSAGE HANDLER" IS CALLED. THE MESSAGE HANDLER IS RESPONSIBLE FOR PREPARING THE MESSAGES AND DISPLAYS WHICH WILL BE USED DURING EXECUTION OF THE SRS SYSTEM.

* THE MESSAGE HANDLER BEGINS BY ATTEMPTING TO OPEN A FILE NAMED "FRAMEF.SRS". THIS FILE CONTAINS DESCRIPTIVE INFORMATION FOR THE VARIOUS CRT FRAMES IN SRS. IF THE FILE IS SUCCESSFULLY OPENED, THE MESSAGE HANDLER HAS NO MORE RESPONSIBILITIES. IT CLOSES THE FILE AND RETURNS TO THE IDLE MODE. IF, HOWEVER, "FRAMEF.SRS" IS NOT SUCCESSFULLY OPENED--THAT IS, NO FILE BY THAT NAME EXISTED ON THE DISK--AN ASCII TEXT FILE CALLED THE "MESSAGE FILE" IS READ AND TWO NEW FILES ARE CREATED:

  1. A RANDOM FILE CONTAINING THE MESSAGES WHICH CAN BE DISPLAYED DURING SRS EXECUTION (RANDOM.SRS);

  2. A FILE CONTAINING DESCRIPTIONS OF THE DISPLAY FRAMES IN THE SYSTEM (FRAMEF.SRS.)

* THIS PROCESSING WILL NOT BE REPEATED ON SUBSEQUENT EXECUTIONS OF SRS UNLESS FRAMEF.SRS IS DELETED OR DESTROYED (SEE FIGURE 2.1-1.)

* AFTER THE ABOVE PROCESSING HAS BEEN COMPLETED, THE HOME FRAME (FIGURE 2.1-2) IS DISPLAYED TO ALLOW THE OPERATOR TO SELECT A SPECIFIC PORTION OF THE SYSTEM FOR FURTHER PROCESSING.

* THE FRAME DATA CREATED BY THE MESSAGE HANDLER IS ACCESSED THROUGHOUT SRS VIA A COLLECTION OF ROUTINES CALLED THE "DISPLAY PACKAGE." THE DISPLAY PACKAGE CAN BE INCLUDED IN ANY TASK WHICH MUST CREATE AND INTERACT WITH A STANDARD SRS DISPLAY FRAME.

* THE MESSAGE HANDLER AND DISPLAY PACKAGE PROVIDE SEVERAL ADVANTAGES FOR SRS. THEY SERVE AS A STRONG IMPETUS FOR STANDARDIZING ALL DISPLAYS. THIS IN TURN PROVIDES A COHERENT "FEEL" TO THE ENTIRE SYSTEM. THEY GREATLY SIMPLIFY CREATION OF NEW DISPLAYS AS WELL AS MODIFICATION OF EXISTING ONES. THEY ALLOW FRAME CONTENTS TO BE STORED ON DISK UNTIL NEEDED THUS SAVING MUCH VALUABLE MEMORY SPACE FOR EXECUTABLE CODE.

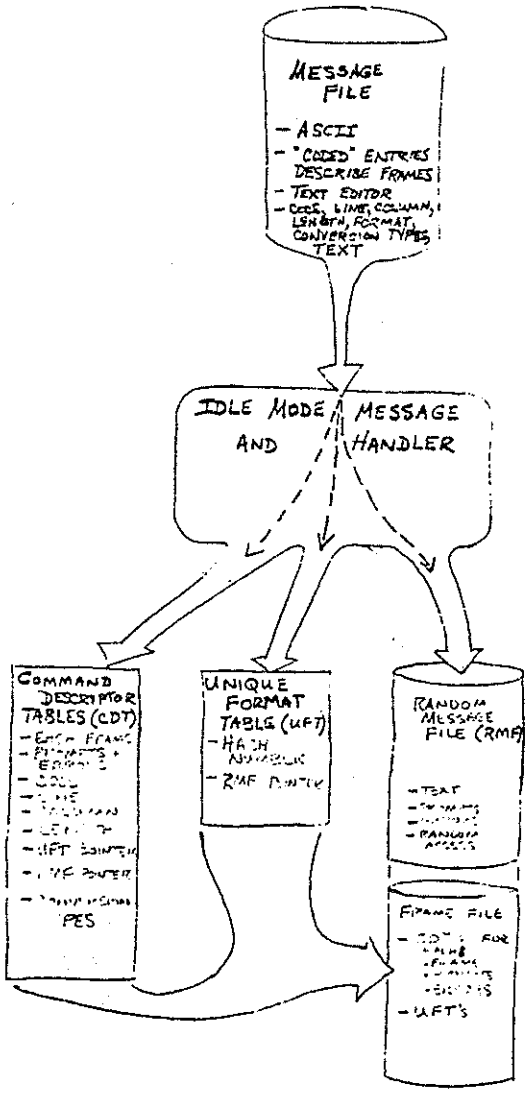


Figure 2.1-1

HOME FRAME

PREPARED BY: _____ CHARGE NUMBER: _____

DATE: _____ PAGE _____ OF _____

```
< OPERATOR ENTRY LINE >
ENTER COMMAND+CR OR CNTL/E+CR TO END          < ERROR AREA >

***** SATELLITE RANGING SYSTEM - MODE SELECTION *****

IN-INITIALIZE
EX-EXECUTE
PR-PROCESSING
DI-DIAGNOSTICS
EP-EPHEMERIDES
```

Figure 2.1-2

2.2 INITIALIZE MODE

- PURPOSE:

  * DEFINE AND VERIFY HARDWARE AND SOFTWARE PARAMETERS WHICH WILL BE USED WHEN TRACKING A SATELLITE.

- COMMENTS:

  * INITIALIZE MODE DISPLAYS THE INITIALIZE FRAME (FIGURE 2.2-1) AND ALLOWS THE OPERATOR TO CHANGE ANY OF THE DISPLAYED PARAMETERS.

INITIALIZE FRAME

PREPARED BY _____ CHARGE NUMBER: _____

DATE: _____ PAGE ___ OF ___

```
< OPERATOR ENTRY LINE >
ENTER COMMAND+CR  OR CNTL/E+CR TO END
***** SATELLITE RANGING SYSTEM - INITIALIZATION *****
*** CIU VALUES ***                    *** CORE VALUES ***
FV-FIELD OF VIEW-MR          XX.X     LA-LEVELING AMPLITUDE-DG   ±XXX.XXXX
AT-ATTENUATION-DB            XX.X     LP-LEVELING PHASE-DG       ±XXX.XXXX
DV-DIVERGENCE-MR             X.XXX    CA-COLLIMATION ANGLE-DG    ±X.XXXXX
ST-START THRESHOLD-MV        XXX.X    RF-RANGE OFFSET-M          ±XXX.XXX
SP-STOP THRESHOLD-MV         XXX.X    AF-AZIMUTH OFFSET-DG       ±XXX.XXXX
TF-TIME OFFSET-SEC           ±XXX.XXX EF-ELEVATION OFFSET-DG     ±XXX.XXXX
MR-MINIMUM RANGE-KM          XXXXX.XX SG-SAG-DG                  ±X.XXXX
RG-RANGE GATE WIDTH-KM       XXXX.XXX MA-MINIMUM ATTENUATION-DB  XX.XX
                                      RV-RCVR PWR REF-DB         ±XX.XX
                                      FI-LASER FIRE INTVL-MS     XXXX
                                      RC-REFRACTION CONSTANT     X.XXX
                                      SA-RANGE SURVEY AZ-DG      ±XXX.XXX
                                      SE-RANGE SURVEY EL-DG      ±XXX.XXX
CL-CALIBRATION                        SC-RANGE SURVEY SAMPLE COUNT XXXXX
```

Figure 2.2-1

TWO MAJOR TYPES OF PARAMETERS ARE SHOWN, VALUES WHICH ARE CIU HARDWARE SETTINGS AND VALUES WHICH ARE USED INTERNALLY BY THE COMPUTER PROGRAMS DURING TRACKING.

* THE INITIALIZE MODE TASK IS HEAVILY OVERLAID. IN GENERAL, EACH COMMAND CAUSES AN OVERLAY TO BE LOADED FROM DISK.

* INITIALIZE MODE WILL INVOKE THE CALIBRATION MODE TASK IN RESPONSE TO THE APPROPRIATE COMMAND.

* TYPING "CNTL/E" CAUSES THE CURRENT PARAMETERS TO BE SAVED AND

THE IDLE MODE TASK TO BE RE-INVOKED.

2.3 CALIBRATION MODE

- PURPOSE:

  * CONTROL MOUNT LEVELING AND COLLIMATION.

- COMMENTS:

  * CALIBRATION MODE USES CODE WHICH IS A SIMPLIFIED VERSION OF THE MOUNT POINTING SUBROUTINES IN EXECUTE MODE. UP TO TEN PRE-SURVEYED TARGETS CAN BE UTILIZED. COMMANDS ARE PROVIDED TO DEFINE NEW SITES, POINT IN EITHER DIRECT OR DUMPED POSITION TO A TARGET, DELETE OR CHANGE EXISTING TARGET PARAMETERS, TAKE OFFSET MEASUREMENTS, AND CALCULATE LEVELING AND COLLIMATION CORRECTIONS (SEE FIGURE 2.3-1.)

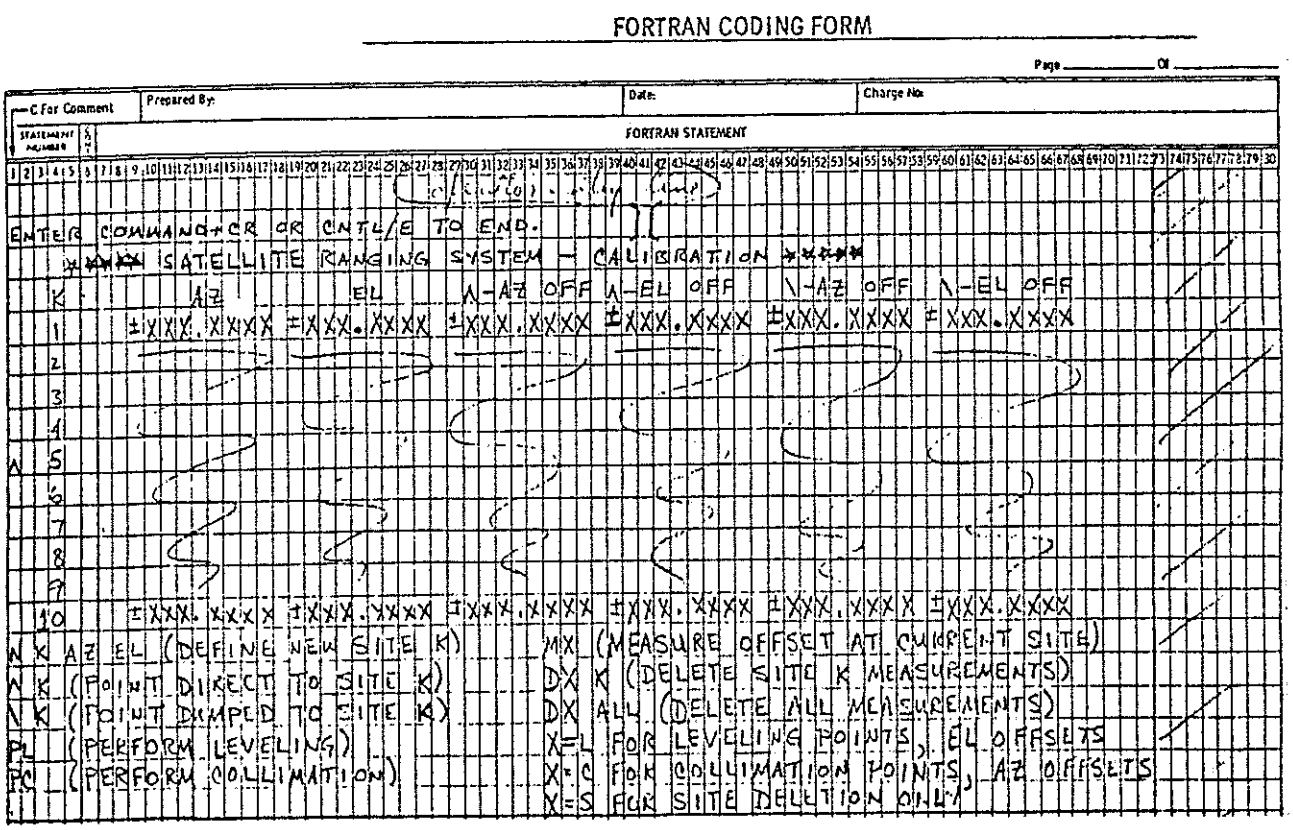


Figure 2.3-1

  * THE CALIBRATION MODE TASK IS NOT OVERLAID.

  * TYPING "CNTL/E" CAUSES THE INITIALIZE MODE TASK TO BE RE-

INVOKED.

2.4 EPHEMERIDES MODE

- PURPOSE:

  * CONTROL PREPARATIONS FOR AND CALCULATIONS OF SATELLITE EPHEMERIDES.

- COMMENTS:

  * EPHEMERIDES MODE CONSISTS OF SEVERAL TASKS--EACH OF WHICH PERFORMS A PARTICULAR PART OF THE OVERALL EPHEMERIS FUNCTION (SEE FIGURE 2.4-1.)

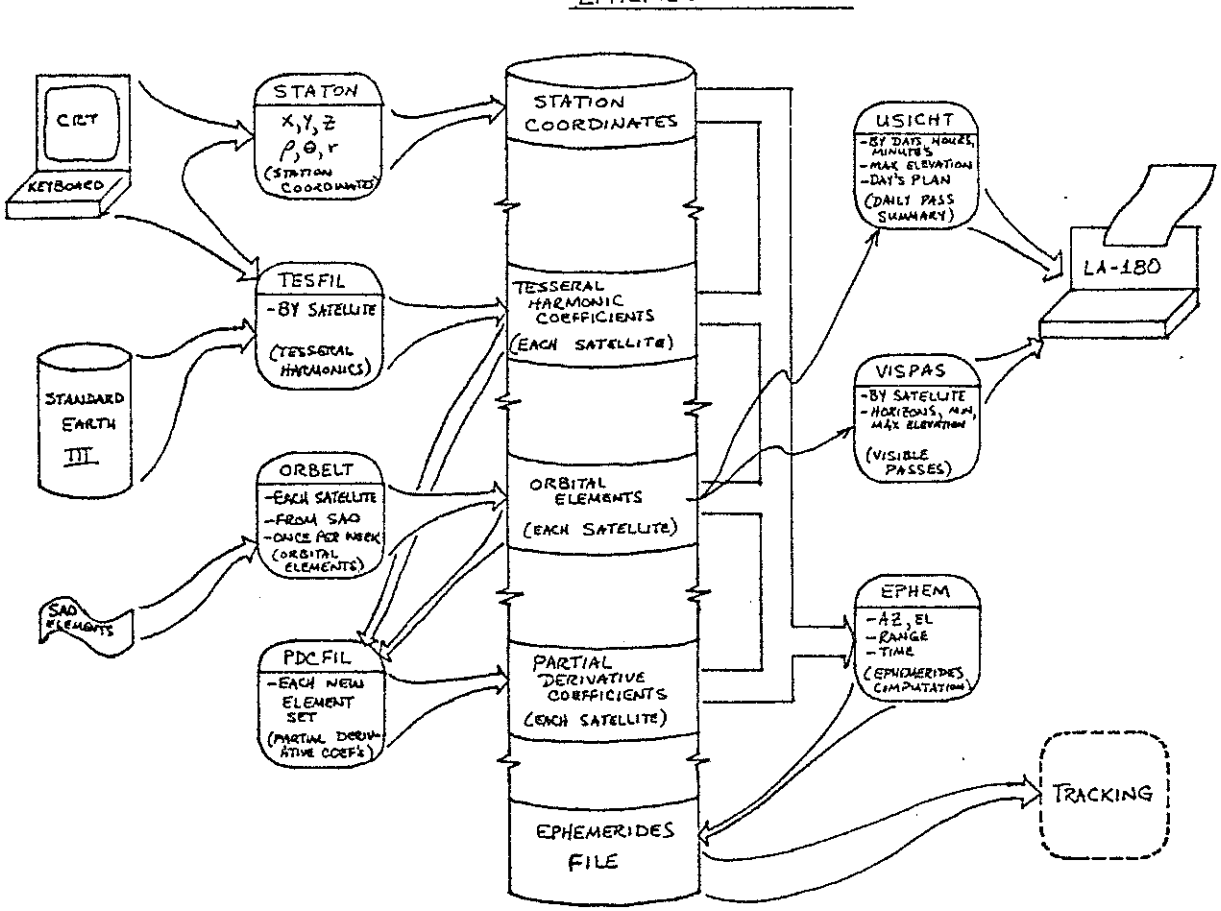


Figure 2.4-1

  * "SRSEPH" (NOT DEPICTED IN FIGURE 2.4-1) DISPLAYS A CRT FRAME WHICH ALLOWS THE OPERATOR TO SELECT THE PARTICULAR TASK WITH WHICH HE WISHES TO WORK. THE TASKS ARE SUMMARIZED IN THE FOLLOWING PARAGRAPHS.

* "STATON" ACCEPTS THE STATION COORDINATES FROM THE OPERATOR.

* "TESFIL" CONTROLS CREATION OF THE TESSERAL HARMONICS FILE FOR EACH SATELLITE. TESSERAL HARMONICS ARE CONSTANTS AND COEFFICIENTS WHICH ARE USED IN EQUATIONS WHICH DESCRIBE THE EARTH'S GRAVITA- TIONAL FIELD. THIS FUNCTION IS USUALLY PERFORMED ONCE PER YEAR OR LESS.

* "ORBELT" CONTROLS ENTRY OF THE ORBITAL ELEMENTS FROM THE SMITHSONIAN ASTROPHYSICAL OBSERVATORY. THIS FUNCTION IS USUALLY PERFORMED ONCE PER WEEK.

* "PDCFIL" CALCULATES COEFFICIENTS OF PARTIAL DIFFERENTIAL EQUATIONS WHICH WILL BE USED IN COMPUTING SPECIFIC SATELLITE ORBITS. THIS FUNCTION MUST BE PERFORMED WHENEVER NEW ORBITAL ELEMENTS HAVE BEEN ENTERED AND BEFORE ANY NEW ORBITS ARE CALCULATED USING THOSE ELEMENTS.

* "VISPAS" CREATES A LISTING OF VISIBLE PASSES FOR ANY SATELLITE FOR ANY TIME PERIOD. THE CALCULATIONS ARE APPROXIMATE AND FASTER THAN FULL ORBITAL COMPUTATIONS. THIS FUNCTION IS USUALLY PERFORMED ONCE FOR EACH SATELLITE WHEN NEW ORBITAL ELEMENTS ARE ENTERED.

* "USICHT" IS SIMILAR TO VISPAS EXCEPT THAT ITS REPORT PROVIDES A DAY-BY-DAY SUMMARY OF SATELLITE PASSES.

* "EPHEM" CALCULATES ONE FULL, VISIBLE PORTION OF A SATELLITE ORBIT. OUTPUT FILES FROM TASKS STATON, TESFIL, ORBELT, AND PDCFIL ARE USED AS INPUTS TO EPHEM. ALSO, THE OPERATOR MUST ENTER SEVERAL PARAMETERS INCLUDING START TIME, DATE, SATELLITE NUMBER, AND 72 CHARACTERS OF FREE-FORM TEXT (A SPECIAL FORMAT FOR THIS TEXT MUST BE USED IF SELECTED MISSION SAMPLES ARE TO BE PUNCHED USING THE SAOPT UTILITY PROGRAM. SAOPT IS DISCUSSED IN ANOTHER PORTION OF THIS DOCUMENT.) EPHEM WRITES THE PARAMETERS, TEXT, CALCULATED TIMES, AZIMUTHS, ELEVATIONS, AND EXPECTED RANGES ONTO A DISK FILE CALLED THE EPHEMERIDES FILE. THIS FILE WILL BE READ BY THE EXECUTE MODE TASKS DURING ACTUAL SATELLITE TRACKING. THIS FUNCTION IS PERFORMED ONCE BEFORE EACH SATELLITE TRACK.

.5 EXECUTE MODE

- PURPOSE:

  * CONTROL SATELLITE TRACKING AND SAMPLE DATA COLLECTION.

- COMMENTS:

  * EXECUTE MODE CONSISTS OF FOUR TASKS, TWO WHICH ARE TIME

CRITICAL, ONE THAT IS BACKGROUND, AND ONE WHICH ALWAYS EXECUTES WHENEVER NONE OF THE OTHER THREE REQUIRE THE COMPUTER (SEE FIGURE 2.5-1.)

EXECUTE (TRACKING) MODE

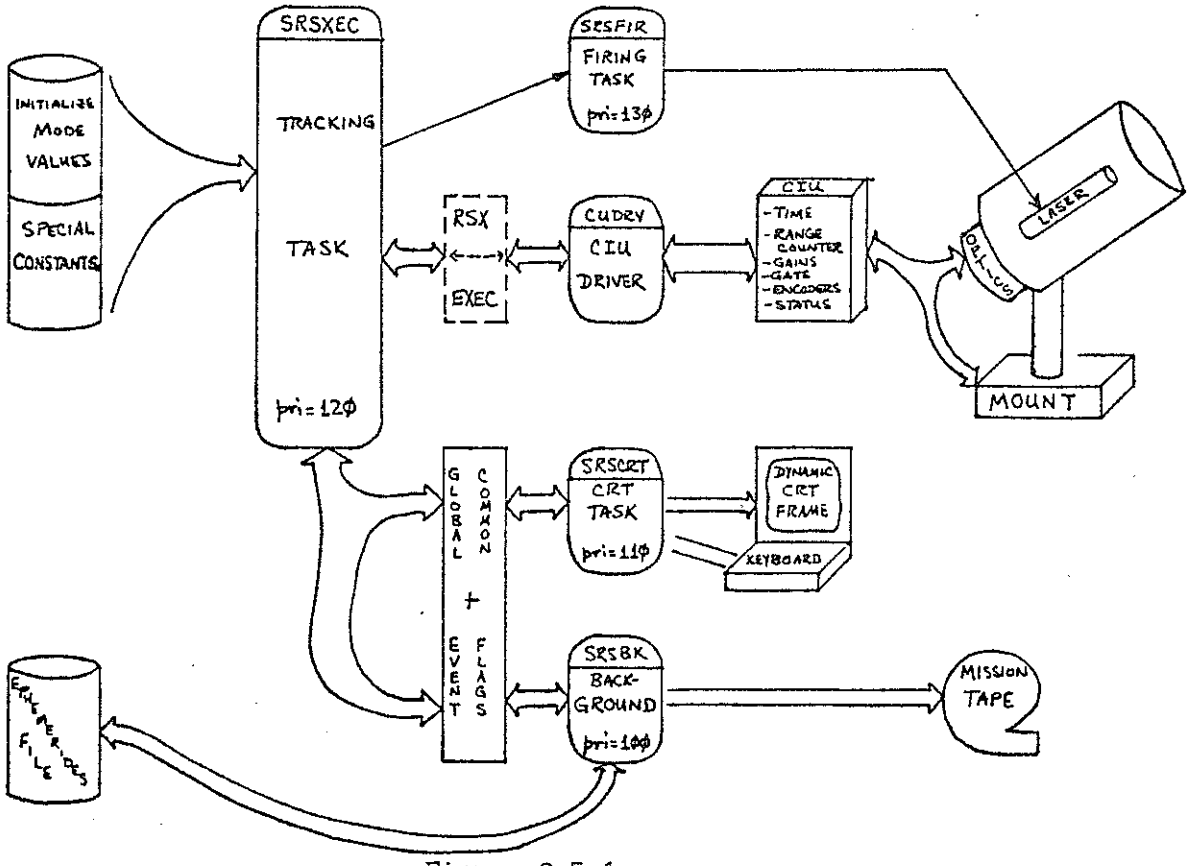


Figure 2.5-1

* "SRSXEC" IS THE TASK WHICH CONTROLS ALL MOUNT AND LASER FUNCTIONS AND GATHERS SAMPLE DATA. IT IS CODED IN FORTRAN BUT USES ONLY SELECTED FORTRAN LIBRARY ROUTINES AND NO FORTRAN I/O. A SPECIAL SUBROUTINE CALLED "CRAMP" IS USED TO AVOID LOADING THE NORMAL FORTRAN OBJECT TIME ROUTINES. THUS THE FORTRAN COMPILER IS USED IN THIS CASE AS AN EFFICIENT GENERATOR OF IN-LINE MACHINE LANGUAGE CODE. THIS IS ESSENTIAL FOR ACHIEVING THE TIME- CRITICAL PROCESSING WHICH IS REQUIRED FOR REAL-TIME CONTROL. SRSXEC HAS SEVERAL OVERLAYS. HOWEVER, DURING REAL-TIME OPERATION ALL REQUIRED CODE IS RESIDENT IN MEMORY SO THAT NO OVERLAY LOADING IS NECESSARY UNTIL THE TRACK HAS FINISHED. SRSXEC DOES NOT DIRECTLY LOAD THE EPHEMERIDES FILE OR WRITE THE SAMPLE DATA TO THE FINAL MISSION TAPE. INSTEAD IT SHARES COMMON BUFFER AREAS WITH SRSBK (DISCUSSED BELOW) AND EXPECTS THAT TASK TO PERFORM THE READ/WRITE FUNCTIONS WHEN REQUIRED.

* "SRSFIR" FIRES THE LASER. IT IS SCHEDULED BY SRSXEC TO BE RUN PERIODICALLY AT A RATE WHICH IS EQUAL TO THE LASER FIRING

INTERVAL AS SPECIFIED ON THE INITIALIZATION FRAME.

* "SRSBK" IS THE BACKGROUND TASK WHICH READS THE EPHEMERIDES FILE AND WRITES THE COLLECTED SAMPLE DATA TO THE MISSION TAPE. IT ALSO PERFORMS DYNAMIC MOUNT LEVELING ADJUSTMENTS ON AN AS-FAST- AS-POSSIBLE BASIS. IT COMMUNICATES WITH SRSXEC VIA GLOBAL DATA BUFFERS AND SOME ASSOCIATED POINTER VARIABLES.

* "SRSCRT" EXECUTES WHENEVER NONE OF THE THREE TASKS DISCUSSED ABOVE REQUIRE THE PROCESSOR. SRSCRT CONTROLS DISPLAY OF ALL DATA VALUES ON THE CRT DURING SATELLITE TRACKING (FIGURE 2.5-2.)

DYNAMIC FRAME

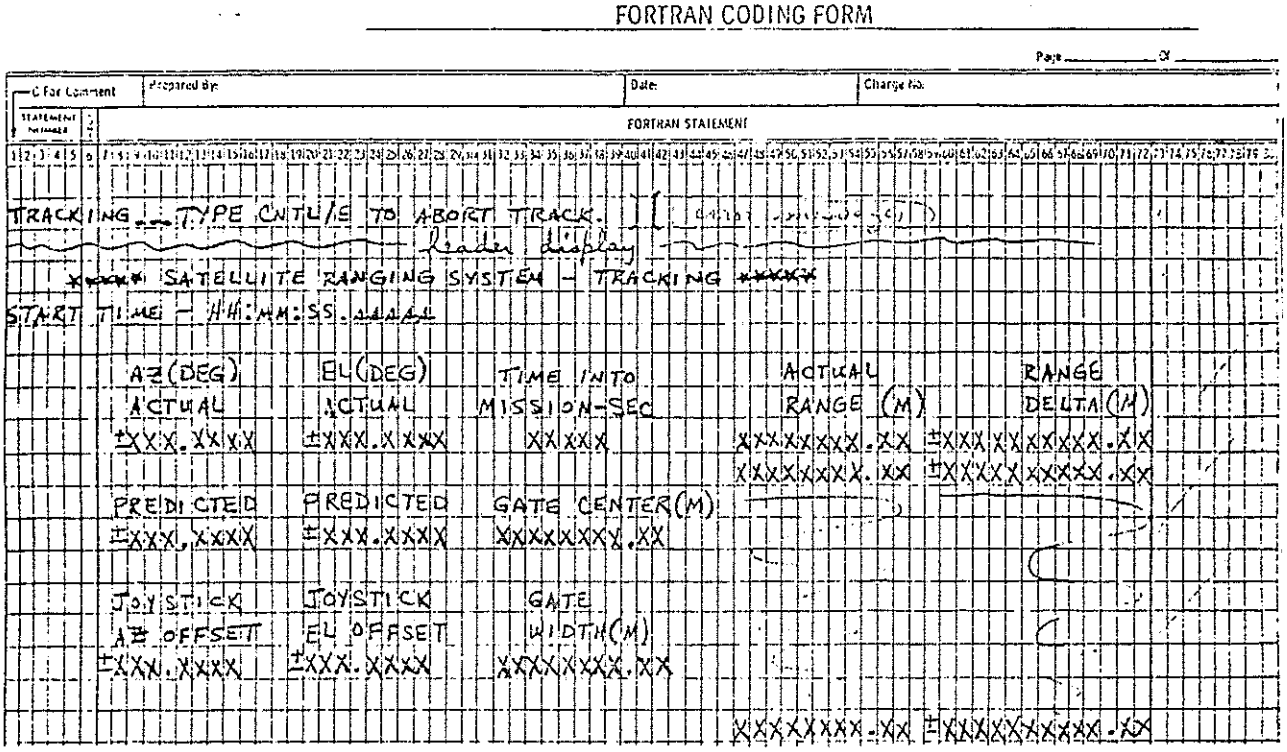


Figure 2.5- 2

IT SHARES A GLOBAL DATA AREA WITH SRSXEC THROUGH WHICH THE DISPLAY VALUES ARE OBTAINED.

2.6 PROCESS MODE

- PURPOSE:

  * EXAMINE SELECTED SAMPLES FROM A MISSION TAPE.

- COMMENTS:

* AFTER DISPLAYING ITS MENU OF OPTIONS (FIGURE 2.6-1) PROCESS MODE RESPONDS TO OPERATOR COMMANDS TO POSITION THE TAPE, LOCATE A MISSION, LOCATE DATA WITHIN A MISSION, AND DUMP DATA IN OCTAL OR INTERPRETED FORMAT TO EITHER THE PRINTER OR CRT.

PROCESSING FRAME

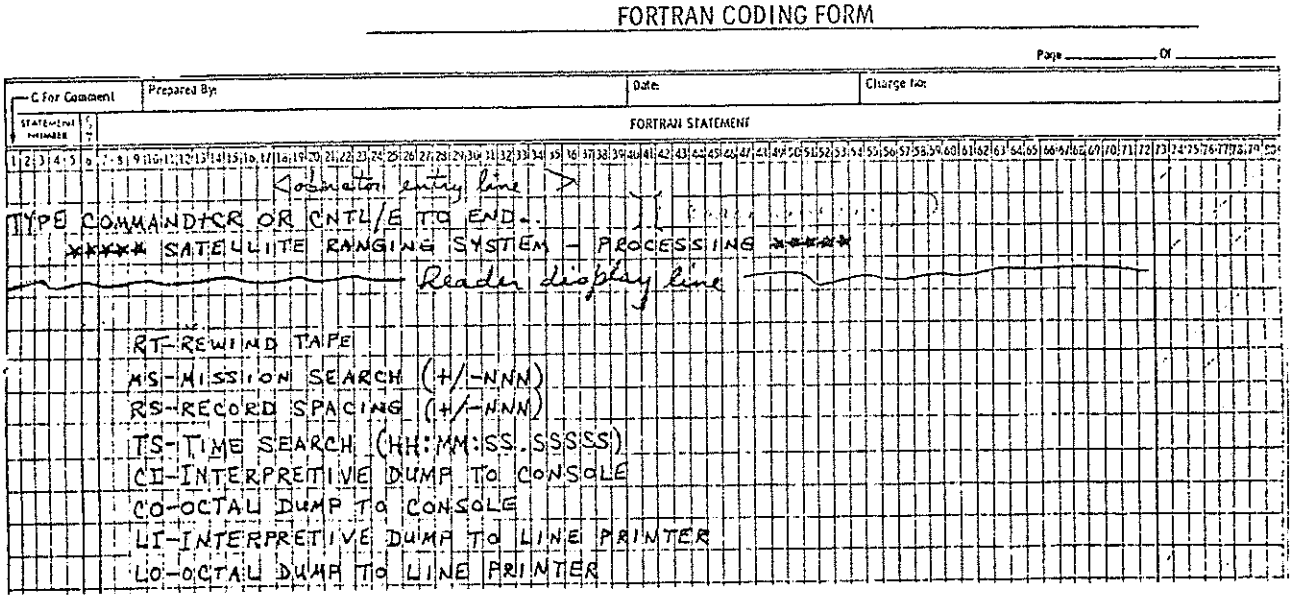


Figure 2.6-1

* THE PROCESS MODE TASK IS OVERLAID. MANY, BUT NOT ALL, OF THE INDIVIDUAL FIELDS WITHIN A SAMPLE REQUIRE A SEPARATE OVERLAY WHEN AN INTERPRETIVE DUMP IS BEING PERFORMED.

* OCTAL DUMPS SHOW EACH WORD OF EACH SAMPLE IN UNINTERPRETED, OCTAL FORMAT. EACH BIT OF DATA IS AVAILABLE FOR INSPECTION. NO SELECTION OF VALUES IS POSSIBLE. LINE SPACING IS USED TO IMPROVE READABILITY. THE LENGTH OF EACH PRINTED LINE IS ADJUSTED TO FILL EITHER THE PRINTER OR CRT LINE SIZE.

* INTERPRETED DUMPS SHOW SELECTED VALUES FROM EACH SAMPLE AFTER THEY HAVE BEEN CONVERTED TO DECIMAL FORM IN MEANINGFUL UNITS. THE SELECTION IS MADE ON THE PROCESS FRAME BEFORE THE DUMP IS STARTED. TIME AND RANGE FIELDS ARE ALWAYS PRINTED. LINE SPACING IS USED TO IMPROVE READABILITY. THE LINE LENGTHS ARE ADJUSTED TO FILL EITHER THE PRINTER OR CRT LINE SIZE.

2.7 DIAGNOSTIC MODE

- PURPOSE:

  * TEST THE ENCODERS FOR PROPER FUNCTION.

- COMMENTS:

  * EITHER AN AZIMUTH OR ELEVATION TEST IS PERFORMED. TO START THE TEST, THE MOUNT IS MOVED TO A LIMIT. IT IS THEN DRIVEN SMOOTHLY IN ONE DIRECTION UNTIL 360 DEGREES HAVE BEEN COVERED. THE ENCODERS ARE SAMPLED RAPIDLY ENOUGH TO DETECT CHANGE IN THE LEAST SIGNIFICANT BIT. THE TEST FAILS IF NON-MONOTONICITY OF THE ENCODER READINGS IS DETECTED. UP TO TWELVE ERRORS ARE REPORTED IN A CRT LISTING (FIGURE 2.7-1.)

DIAGNOSTIC FRAME

FORTRAN CODING FORM

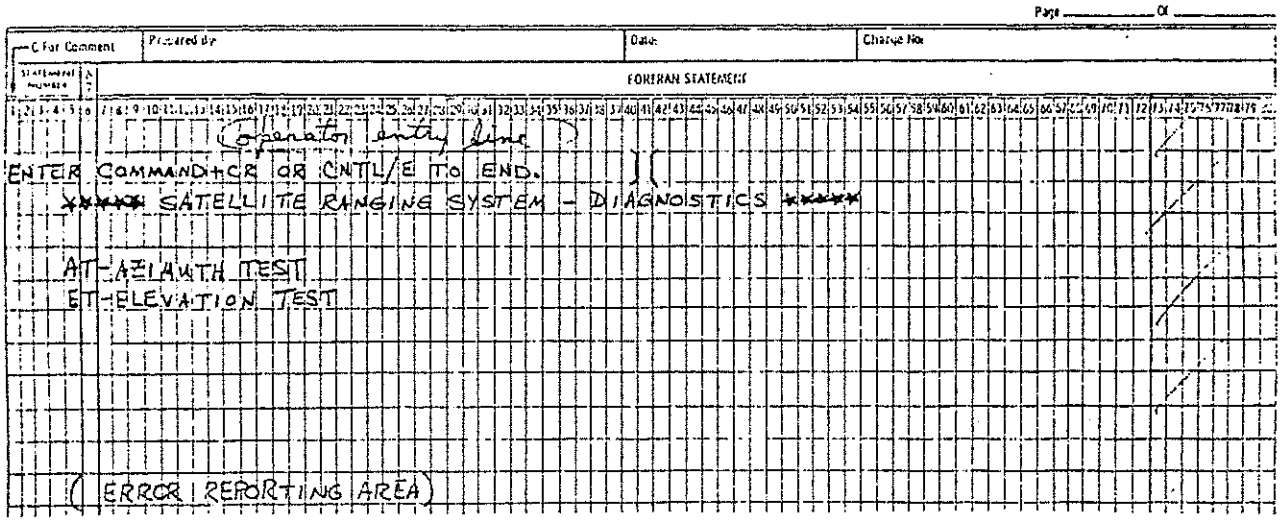


Figure 2.7-1

  * DUE TO LOGICAL MEMORY MAPPING LIMITATIONS, THIS TASK DOES NOT REFERENCE CERTAIN RESIDENT LIBRARIES AND GLOBAL DATA AREAS BUT DEFINES AND/OR LOADS THESE VALUES LOCALLY.

2.8 SAO "QUICK-LOOK" PAPER TAPE PUNCHING UTILITY

- PURPOSE:

  * PUNCH SELECTED SAMPLES FROM A MISSION INTO PAPER TAPE FOR TRANSMITTAL TO SMITHSONIAN ASTROPHYSICAL OBSERVATORY IN CAMBRIDGE, MASSACHUSETTS.

- COMMENTS:

* "SAOPT" DISPLAYS A MENU OF COMMANDS (FIGURE 2.8-1.).

72-Character Header Format

Column(s)	Contents
1-7	Satellite identification number
8	blank
9-13	Station identification number
14	blank
15-16	Year of measurement
17-19	Day of measurement (1 ≤ day ≤ 366)
20	blank
21-25	Pressure in millibars
26	blank
27-30	Temperature in degrees C
31	blank
32-35	Humidity in percent
36	blank
37-45	Calibration Range in meters
46	blank
47-55	Clock Correction in seconds
56-72	Text entered by operator or blanks

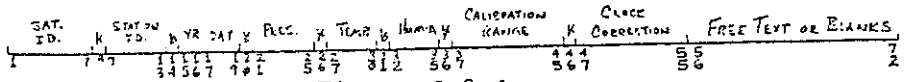


Figure 2.8-1

COMMANDS ARE AVAILABLE FOR LOCATING A MISSION, DEFINING SAMPLE SELECTION PARAMETERS, COUNTING THE NUMBER OF SAMPLES WHICH FALL WITHIN SPECIFIED BOUNDS, AND PUNCHING THE TAPE.

* IT IS AN SAOPT REQUIREMENT THAT THE 72-CHARACTER TEXT FIELD IN THE HEADER BLOCK OF THE MISSION BE DEFINED ACCORDING TO THE FORMAT GIVEN IN FIGURE 2.8-2.

2.9 CIU DEVICE HANDLER (CUDRV)

- PURPOSE:

* DIRECT HARDWARE HANDLING OF THE COMPUTER INTERFACE UNIT (CIU).

SAOPT CONTROL FRAME

```
ENTER COMMAND+CR OR CNTL/E+CR TO END

***** SATELLITE RANGING SYSTEM - SAO PAPER TAPE PUNCHING *****

RT-REWIND TAPE
MS-MISSION SEARCH (+N)
MN-MINIMUM RANGE DELTA BOUND  ±XXXXXXXXXX.XX
MX-MAXIMUM RANGE DELTA BOUND  ±XXXXXXXXXX.XX
SD-SELECTION DELTA                      XXXX
RM-RANGE SURVEY MEAN                 XXXXXX.X
CO-COUNT IN BOUNDS SAMPLES
PT-PUNCH TAPE
```

Figure 2.8-2

- COMMENTS:

  * "CUDRV" SERVES AS THE INTERFACE BETWEEN APPLICATION PROGRAMS ON THE ONE HAND AND THE LASER, MOUNT AND OPTICAL HARDWARE ON THE OTHER. CUDRV IS THE PROGRAM WHICH SENDS FUNCTION CODES DIRECTLY TO THE CIU AND WHICH RECEIVES INTERRUPTS, DATA AND STATUS CODES DIRECTLY FROM IT.

  * CUDRV IS CODED AND INSTALLED AS AN INTEGRAL PART OF THE RSX-11M OPERATING SYSTEM. THIS ALLOWS CUDRV TO TAKE ADVANTAGE OF RSX FEATURES SUCH AS REQUEST QUEUING, I/O PAGE ACCESS, AND INTERRUPT SERVICING. CODING IS IN MACRO ASSEMBLY LANGUAGE TO ACHIEVE THE FASTEST POSSIBLE EXECUTION SPEED.

  * TO FURTHER INCREASE SPEED THE OVERHEAD OF SOME I/O REQUESTS TO CUDRV IS ELIMINATED BY THE USE OF "DIRECTIVE PACKETS." DIRECTIVE PACKETS ARE A SUPERSET OF THE NORMAL I/O REQUEST MECHANISM. NORMALLY A SINGLE I/O REQUEST SPECIFIES THE TRANSFER OF AT MOST ONE CONTIGUOUS BLOCK OF DATA. THE PARAMETERS FOR THE TRANSFER ARE STORED DIRECTLY IN THE I/O PACKET WHICH IS PLACED BY RSX IN THE DRIVER'S QUEUE. REQUESTS TO CUDRV, HOWEVER, ADD ONE LAYER TO THIS STRUCTURE. THE PARAMETERS IN THE QUEUED I/O PACKET ONLY LOCATE A "DIRECTIVE PACKET" IN THE APPLICATION'S ADDRESS SPACE. THE DIRECTIVE PACKET CONTAINS A BLOCK OF PARAMTERS WHICH WILL CONTROL THE CIU TRANSFER. THESE PARAMETERS MAY SPECIFY SEVERAL EXCHANGES OF CONTIGUOUS BLOCKS OF DATA STARTING AT RANDOM CIU ADDRESSES. CUDRV WILL COMPLETE ALL OF THEM BEFORE PERFORMING THE I/O COMPLETION SEQUENCE WHICH ALERTS THE APPLICATION TASK THAT THE TRANSFER IS COMPLETE. THUS, DIRECTIVE PACKETS BYPASS THE RSX OVERHEAD OF SEVERAL I/O REQUESTS.

3 . RS DESIGN AND CODING PROCEDURES

SEVERAL EXPLICIT GUIDELINES WERE FOLLOWED IN THE DETAILED DESIGN AND CODING PHASES OF THE WETTZELL PROJECT. IN PARTICULAR, THE THEORIES OF STRUCTURED, TOP-DOWN PROGRAMMING WERE CONVERTED TO SPECIFIC METHODS WHICH WERE FOLLOWED RIGOROUSLY. A GREAT DEAL OF DETAILED DESIGN WAS HAND-WRITTEN IN A DESIGN (OR META) LANGUAGE AND THOROUGHLY CHECKED FOR CONSISTENCY, PARTICULARLY IN THE REAL-TIME PART OF THE SYSTEM. DESIGN AND CODE WAS REVIEWED BY DIFFERENT PROJECT PROGRAMMERS BEFORE THE ORIGINATOR MOVED TO THE NEXT PHASE OF HIS WORK.

THIS PROCESS WAS A TIME CONSUMING ONE AND CREATED INITIAL FEELINGS OF DISTRUST ON THE PART OF PROJECT MANAGEMENT WHO WERE ANXIOUS FOR END RESULTS. IN THE END, HOWEVER, SEVERAL GENERALLY AGREED UPON REWARDS EMERGED FROM THIS APPROACH:

1. A TRUE AND EARLY COST PICTURE FOR THE SOFTWARE DEVELOPED--THE HARDWARE COSTS WERE MUCH LONGER IN MATERIALIZING;

2. THE SOFTWARE WAS RELATIVELY EASY TO IMPLEMENT;

3. THE SOFTWARE WAS DEVELOPED IN A TIME FRAME REASONABLY CLOSE TO A ONCE-REVISED SCHEDULE;

4. THE SOFTWARE WAS EXTREMELY ADAPTABLE TO NEW REQUIREMENTS BOTH WITHIN THE PROJECT AND LATER, IN WETTZELL, WHEN THE SYSTEM WAS EXPANDED FOR LUNAR RANGING.

THE PROCEDURE OR SUBROUTINE UNIT WAS ALWAYS OF A LIMITED SIZE AND SINGLE PURPOSE. THE PURPOSE WAS EXPLICITLY WRITTEN OUT AS THE FIRST STEP IN PRODUCING EACH ROUTINE. THAT ACT OF WRITING-- OF BEING CONSCIOUS OF THE PURPOSE OF THE ROUTINE ABOUT TO BE PRODUCED--WAS PERHAPS THE SINGLE MOST IMPORTANT TECHNIQUE USED IN CREATION OF THE WETTZELL SYSTEM.

THE OTHER GUIDELINES WERE NOT ELABORATE. YET THEY PROVED TEDIOUS AT TIMES. THE NEED FOR SO MUCH WRITING TO CREATE EVEN SMALL ROUTINES--EVEN THOUGH THE GUIDELINES WERE INTENTIONALLY MADE "BAREBONES"--WAS SOMETIMES FRUSTRATING. IN RETROSPECT, HOWEVER, THE MAJOR ADVANTAGE IN FOLLOWING THE STRICT RULES WAS THAT THE PROGRAMMER'S THOUGHTS AND ACTIVITIES WERE THEREBY CHANNELLED, DIRECTED, CONTROLLED.

THE REMAINDER OF THIS SECTION SUMMARIZES THE RULES THAT WERE FOLLOWED. THE READER SHOULD REMEMBER THAT THEY WERE SUCCESSFUL ON A PROJECT WHICH HAD AT MAXIMUM ONLY THREE PROGRAMMERS. LARGER PROJECTS WILL NO DOUBT DICTATE MODIFICATIONS OF THESE PROCEDURES. SMALLER PROJECTS MAY SAFELY CIRCUMVENT CERTAIN FEATURES. YET THE BASIC CONCEPT OF A SINGLE, WRITTEN PURPOSE FOR EACH ROUTINE (BEWARE THE WORDS "AND" AND "OR") AND A CAREFUL, STEP-BY-STEP, TOP-DOWN APPROACH TO DESIGN ARE VALID NO MATTER WHAT SIZE THE PROJECT.

3.. SRS DESIGN GUIDELINES

- EVERY PROCEDURE HAS ONE AND ONLY ONE FUNCTION.
- NO PROCEDURE DESIGN LARGER THAN A SINGLE SHEET OF PAPER.
- TOP-DOWN DESIGN (PROGRESS FROM HIGHEST CONTROL LEVELS TO MOST SPECIFIC.)
- DO ONLY THE REQUIRED PROCESSING AT EACH LEVEL. DO NOT TRY TO DO TOO MUCH IN ANY ROUTINE.
- NO "GO TO" STATEMENTS IN THE DESIGN (NOTE THAT "GO TO" STATEMENTS ARE PERMITTED UNDER CERTAIN CONDITIONS IN THE FINAL FORTRAN SOURCE CODE.)
- HAVE THE DESIGN REVIEWED BY AT LEAST ONE OTHER PROGRAMMER BEFORE CODING.
- USE ONLY THE DESIGN LANGUAGE ELEMENTS GIVEN IN THE NEXT ARTICLE EXCEPT IN VERY SPECIAL CASES.
- DO THE FIRST DESIGN WITHOUT ARGUMENT LISTS ON PROCEDURE CALLS. MAKING THE NAMES OF PROCEDURES MORE MEANINGFUL WILL HELP. ADD ARGUMENTS AFTER THE LOGICAL FLOWS OF CONTROL ARE ESTABLISHED.
- EVERY COMMON (THAT IS, GLOBAL) AREA HAS ONE AND ONLY ONE REASON FOR BEING CREATED.

3.2 SRS DESIGN LANGUAGE ELEMENTS

STATEMENT *********	EXPLANATION/COMMENTS **********
PROCEDURE	WILL BECOME A "SUBROUTINE".
MAIN PROCEDURE	WILL BECOME THE "PROGRAM" ROUTINE.
<TYPE> PROCEDURE	WILL BECOME A "FUNCTION". <TYPE> IS IN THE SET (INTEGER, LOGICAL, REAL, DOUBLE PRECISION)
BEGIN	STARTING POINT OF A COMPOUND STATEMENT OR OF A PROCEDURE.
END	STOPPING POINT OF A COMPOUND STATEMENT OR OF A PROCEDURE.
"STMT"	ONE OR MORE EXECUTABLE EXPRESSIONS. COMPOUND STATEMENTS ARE ENCLOSED BY

	A BEGIN...END PAIR. SINGLE STATEMENTS ARE TERMINATED BY A SEMI-COLON (";").
"COND"	A LOGICAL EXPRESSION ("CONDITION").
WHILE "COND" DO "STMT"	EXECUTE "STMT" AS LONG AS "COND" IS TRUE. "COND" IS TESTED BEFORE "STMT" IS EXECUTED.
REPEAT "STMT" UNTIL "COND"	SAME AS WHILE...DO.... EXCEPT THAT "COND" IS TESTED AFTER EXECUTION OF "STMT".
IF "COND" THEN "STMTT" ELSE "STMTF"	WHEN "COND" IS TRUE, "STMTT" IS EXECUTED. WHEN "COND" IS FALSE, "STMTF" IS EXECUTED.
CASE "STMT" OF BEGIN VAL1: "STMT1" VAL2: "STMT2" . . . VALN: "STMTN" END	"STMT" IS EVALUATED TO OBTAIN AN INTERNAL ANSWER (CALL IT "VAL".) THEN WHEN "VAL"="VAL(I)", "STMT(I)" IS EXECUTED. CONTROL THEN PASSES TO THE LINE AFTER "END".
VARIABLE'NAMES'CAN'BE'LONG PROCEDURE'NAMES'CAN'BE'LONG	VARIABLE OR SUBROUTINE NAMES MAY BE ANY NUMBER OF CHARACTERS AND/OR WORDS LONG. WORDS ARE CONCATENATED USING A SINGLE APOSTROPHY AS SHOWN.
PROCEDURE CALLS	IT IS NOT NECESSARY TO WRITE "CALL" BEFORE A PROCEDURE NAME. SIMPLY WRITING THE NAME IN-LINE WITH OTHER DESIGN CODE IS ENOUGH TO INDICATE INVOKATION OF THE PROCEDURE.
GLOBAL VARIABLES AND BLOCK DEFINITIONS	DEFINE GLOBAL BLOCKS ON A SEPARATE PAGE. TO SHOW INCLUSION OF THAT BLOCK IN A PROCEDURE, WRITE THE BLOCK NAME IMMEDIATELY AFTER THE "PROCEDURE" STATEMENT.
PROCEDURE ARGUMENTS	ALL ARGUMENTS MUST APPEAR IN EXPLICIT TYPE STATEMENTS.
FOR VAR:=VAR1 TO VARN STEP VARS DO "STMT"	EXECUTE "STMT" WITH "VAR" SET TO "VAR1", "VAR1+VARS", "VAR1+2*VARS", AND SO ON UNTIL VAR > VARN.
<<COMMENTS>>	DOUBLE BRACKETS ENCLOSE COMMENTS.

3.3 SRS CODING GUIDELINES

SRS IS WRITTEN IN FORTRAN WITH ONLY OCCASSIONAL REVERSION TO ASSEMBLY LANGUAGE. THE FOLLOWING GUIDELINES WERE FOLLOWED TO IMPOSE CONTROL AND A DEGREE OF UNIFORMITY ON THE CODE.

- NO ROUTINE LONGER THAN 60 LINES, MOST LESS THAN 30.
- INCLUDE THE STANDARD HEADER (SEE FOLLOWING ARTICLE) BEFORE EVERY SUBROUTINE.
- NO "DROP THROUGH" CODE INTO A LABELLED STATEMENT. WRITE A "GO TO" STATEMENT. THE FORTRAN COMPILER IGNORES SUCH "GO TO" STATEMENTS BUT THE CODE IS CLEARER TO READ.
- EVERY VARIABLE EXCEPT SIMPLE LOOP COUNTERS MUST BE IN AN EXPLICIT TYPE STATEMENT.
- FORTRAN "COMMON" STATEMENTS ARE DEFINED IN ONE AND ONLY ONE "INCLUDE" FILE WHICH CAN BE COMPILED AS PART OF ANY ROUTINE. (THIS IS A FEATURE OF THE DEC FORTRAN COMPILER.)
- NEW LIBRARY ROUTINES OR GLOBAL DATA DEFINITIONS ARE SUBMITTED TO THE SYSTEM MANAGER FOR INCLUSION IN THE SYSTEM.
- INCLUDE EXPLANATORY COMMENTS IN EACH ROUTINE TO EXPLAIN WHAT IS HAPPENING, WHY, AND ANY SPECIAL CONSIDERATIONS.
- AVOID ALL "TRICKY" CODE WHEREVER POSSIBLE. COMMENT IT WELL WHEN IT MUST BE USED.
- EVERY ROUTINE IS ENTERED ONLY THROUGH THE "SUBROUTINE" OR "FUNCTION" STATEMENT, NEVER THROUGH AN "ENTRY" STATEMENT. NEVER USE AN "ENTRY" STATEMENT.
- EVERY ROUTINE CONTAINS ONLY ONE "RETURN" STATEMENT AT THE END OF THE ROUTINE IMMEDIATELY BEFORE THE "END" STATEMENT.

3.4 SRS STANDARD SUBROUTINE HEADER

THE FOLLOWING FORM WAS USED FOR CREATING A HEADER BEFORE EVERY SUBROUTINE. FOR MACRO (THAT IS, ASSEMBLY LANGUAGE) ROUTINES, THE "C" IN COLUMN 1 IS REPLACED WITH A SEMI-COLON. THE ARTICLE NUMBERS ARE AN AID FOR COMPUTER DOCUMENTATION. VARIABLE PARTS OF THE FORM ARE INCLUDED IN <BRACKETS>.

```
C**
C     .0 <ROUTINE NAME>
C          .1 FUNCTION
C             <SINGLE PURPOSE FUNCTION>
```

```
C         .2 CALLING SEQUENCE
C            CALL <ROUTINE NAME AND CALLING SEQUENCE>
C                 <EACH PARAMETER IS EXPLAINED BRIEFLY.>
C         .3 GLOBAL INPUTS
C            <ALL INPUT VARIABLES THAT ARE NOT PART OF THE CALLING>
C            <SEQUENCE ARE EXPLAINED HERE.>
C         .4 UNIQUE VARIABLES/DATA
C            <SPECIAL NOTES THAT EXPLAIN UNUSUAL OR DIFFICULT>
C            <FEATURES OR DATA IN THE ROUTINE.>
C         .5 OUTPUTS
C            <EXPLANATION BY VARIABLE NAME OF EACH CHANGE>
C            <WHICH THIS ROUTINE MAKES ON DATA ITEMS.>
C
        SUBROUTINE <NAME AND CALLING PARAMETER LIST>
C
C       <THE DATA DEFINITION STATEMENTS GO HERE.>
C
C
C
C       <THE EXECUTABLE STATEMENTS GO HERE.>
C
C
C FINISHED.
C
900     CONTINUE
        RETURN
        END
```

HP 9825B-based Software System for Laser Radar

by

R.Neubert
Central Institute for Physics of the Earth
Potsdam, GDR

ABSTRACT

The software system described here has been designed for automatic control of the Potsdam laser station. Minimum computer requirements could be attained by rather simple approximations. Nevertheless reasonable reliable blind tracking has been obtained during the MERIT short campaign for all satellites included in that program. The laser beam divergency used is 1 mrad or even less. In the following we outline the method used for the main parts of the system. For more detailed information the reader is referred to reference [4] of this paper which contains commented program listings and examples.

1 Satellite Position Prediction

The predictions are calculated on the basis of SAO orbital elements consisting in a polynomial part and additional long periodic terms. The polynomial part is linear for all elements except mean anomaly, which is given up to 3rd degree for low orbiting satellites. Among the 16 long periodic terms supplied by the prediction center we use the 4 most important only. These are the terms B_{21} (Ω correction), B_{31} (i correction), B_{51} (χ correction), and the constant part of the η correction. The latter refer to the new SAO element format [1].

The calculation of the satellite position is carried out in two steps.

- In the first step, the Keplerian elements are evaluated for an epoch very near or identical to the time for which the position is wanted (for a pass prediction the culmination time is used in general). In this step (routine "BE(T)") the polynomial part and long periodic terms are taken into account. The major semi axis is evaluated using the well-known first order approximation [2].

- In the second step ("SAPOS") linear changes to the node and perigee and short periodic J_2-perturbations are applied and the satellite position is calculated. In this part the topocentric position including refraction correction is determined as well.

The transformation into the coordinate system determined by the orientation of the 4-axis mount is done by a separate routine if necessary. For this the orientation of the 3rd mount axis (polar axis, main tracking axis) has to be known or calculated before. For a given pass this orientation is determined from 3 predicted topocentric satellite positions around the culmination point fulfilling the geometrical condition, that the 3 angles between the 3rd axis and all the position vectors should become equal (first part of routine "EPHB").

2 Culmination Time Estimation

To print a list of culmination times , we use a very approximate but fast method which is well-known [2]. The calculation is carried out in 3 steps:

Step 1: Let t_0 be the start time of the search, for instance MJD of first day. Then the time difference to the next crossing of the station through the orbital plane (ascending part) is:

$$t_1 = (\lambda - \lambda_b + \Omega_{t_0} - \theta_{t_0})/(\theta_1 - \Omega_1)$$

$$\sin(\lambda) = \tan(\phi)/\tan(i),\ \lambda = 90^\circ \text{ for } i < \phi$$

where

λ_b = longitude of the station
ϕ = latitude
Ω = longitude of ascending node
Ω_1 = rate of change of Ω ($^\circ$/day)
θ = sidereal time angle
θ_1 = rotational speed of the Earth ($^\circ$/day)
i = orbital inclination

For the descending part of the orbit we have:

$$t_{12} = t_1+(180^\circ-2\lambda)/(\theta_1-\Omega_1), \quad t_{12} = t_1 \text{ for } i < \phi$$

Step 2: In the next step the time is added, which is necessary for the satellite to move from its position at t_0+t_1 to the crossing point:

$$t_2 = (M - M_{t_1})/n$$
$$M = \nu - e*\sin(\nu)$$
$$M_{t_1} = M_{t_0} + n*t_1$$
$$\nu = \lambda - \Omega$$
$$\sin(\lambda) = \sin(\phi)/\sin(i), \quad \lambda = 90^\circ \text{ for } i < \phi$$

where

Ω = argument of perigee
n = dM/dt = anomalistic frequency.

Step 3: In the last step of approximation the Earths rotation during the time t_2 is taken into account. Using plane geometry instead of spherical we obtain:

$$t_3 = t_2*((\theta_1-\Omega_1)*\cos(i))/360n, \quad (\theta_1 \text{ and } \Omega_1 \text{ are in } ^\circ\text{/day})$$

The culmination time $t_0+t_1+t_2+t_3$ is accurate to about 1 min. In the "KULM"- routine, t_1 is calculated for the first day only. For the next day a multiple of $360/(\theta_1-\Omega_1)$ is added to t_1 of first day and the other steps are repeated.

3 Real Time Control of the Equipment

For real time control, the computer outputs an instruction each second via the same channel to the control equipment. This repetition frequency is completely sufficient, because the laser shots are spaced by 6 sec. at our station.

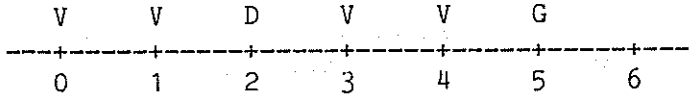
In the normal mode

a), the following types of output information are used: angular velocity along track, cross track correction angle, and receiver gate time (range). The type of information is coded by the first digit of the seven digit word. The remaining 6 digits contain the value of the information (see [4]). Not all of the possible 9 code digits are necessary in the normal mode. The remaining are used to switch the control logic to other modes of operation. In this way the computer can switch to:

b) Start-stop drive of both axes. In this mode being introduced for the planned LASSO experiment, the laser can be programmed to shot at an arbitrary time within the 6 sec. interval using an additional output information (in normal mode the laser repetition rate is fixed).

c) Continuous drive of both axes. This mode is useful for star tracking.

In the following we consider the normal mode a) only. The cross track information and gate time have to be output once each in a 6 sec interval. Therefore, we have the opportunity to change the along track velocity 4 times in each cycle. This is important for a fast and stable adaptation of the system after a perturbation by a manual correction. The tracking algorithm insures, that at the next laser shot both the position and the velocity of the satellite are approximated within the accuracy level of the (shifted) prediction. Let us consider the control cycle in more detail according to the following time diagram:



The full seconds of a cycle are denoted by 0 to 6, where 6 is identical with point 0 of the next cycle. In the upper half of the diagram the type of information accepted by the control circuit in the corresponding point is printed. V stands for velocity, D for cross track correction (δ), and G for gate time. In the moment of acceptance the instruction is immediately executed. For "V" this means, that the velocity is switched to the new value, for "G" the laser is triggered and the gate counter started. In the foregoing cycle the computer calculates the satellite positions in points 4 and 5 taking into account a time shift, which may be changed by the observer in real time. The "real" position of the laser beam at point 0 is calculated by the computer by integrating the velocity values given before. Now the condition is set,

that the laser beam should be directed to the predicted positions in both the points 4 and 5. To obtain a definite solution, the simple additional requirement was introduced, that the same velocity change X should be applied at points 0 and 1, and another change Y at points 3 and 4. Denoting the velocity at point 0 by V_0 and the position in point i by S_i, we obtain for the velocity changes:

$$X = (2(S_4-S_0)-(S_5-S_4)-7V_0)/12$$

$$Y = ((S_5-S_4)-V_0)/2-X$$

In these equations the time interval is removed for simplicity, because it is assumed to be the time unit for velocity (1 sec. in our case). The second equation is identical with the condition, that the total sum of velocity changes should be equal to the difference of the velocities at points 5 and 0. If the required total velocity change but not the positional displacement is very low, the changes X and Y are approximately equal in absolute value, but opposite in sign. This is the case for time corrections near the culmination.

The limited interrupt possibilities of the desktop computer require, that there is no routine in the control program consuming more than 1 sec. for execution. This is no problem at present, because the most critical routine "SAPOS" is well below this limit. So the full program could be used in real time and interpolation techniques have been unnecessary. Some care has been given to the distribution of the computing time to the intervals. The reading and printing of the measurements for instance had to be placed after point 6 (1 sec. after the laser shot).

4 Filtering of the measurements

The selection of the laser returns from noise pulses is done mainly by comparing to predictions. The raw data stored after observation on tape cartridge contain all the measurements. They are read in again by the analysis program and compared to predictions. The differences are below 1000 m. in general for real returns. For best fitting of the predictions to the measurements a time shift may be introduced (Sect. 6.1). The remaining differences are weakly dependent on time and may be easily represented by a low degree polynomial. In this way the instrumental noise level down to some 10 cm. rms is easily reached for the residuals and the false alarm points may be reliably eliminated if the total number of returns is not too low. The filtered measurements are corrected for instrumental effects (calibration, time of flight counter frequency error, eccentric correction of the mount) and stored on a separate tape cartridge.

5 .rediction Improvement Using Own Laser Data

Using data from only one station, prediction improvement has some limitations. The results are very stable, however, if only the mean anomaly is changed. Fortunately, the SAO elements are leading to accurate predictions of the cross track components even if using the simple algorithm outlined. On the other hand, strong along track errors are observed for low orbiting satellites and, therefore, updating of mean anomaly coefficients is useful and necessary.

5.1 Time Shift

The most simple improvement is a constant correction of mean anomaly or an equivalent time shift. For this it is completely sufficient to use two points of one pass only. Two methods have been used in parallel. The first is the standard least square differential improvement leading to the time shift dt:

$$dt = ((D_1-S_1)Q_1+(D_2-S_2)Q_2)/(Q_1^2+Q_2^2)$$

where:

S_1, S_2 = predicted ranges for the two epochs
D_1, D_2 = measured ranges
Q_1, Q_2 = time derivatives of range

This formula gives good results if the remaining range differences (cross track error) are small. If this is not the case, the points should be chosen symmetrically to the culmination at approximately equal range. If measurements are available only before or after the culmination, the following method may lead to better results. It is deduced from the condition, that after introduction of the time shift the range differences should become equal, i.e.:

$$S_1 + Q_1*dt - D_1 = S_2 + Q_2*dt - D_2$$

or:

$$dt = (D_1-S_1-D_2+S_2)/(Q_1-Q_2)$$

This method tends to bring the measured and predicted times of minimum range into close agreement. The latter is exactly the case, if the measured and predicted range versus time curves are identical in shape differing in a constant offset only. The results of this formula may be bad, however, if the points are too close together.

Both formulas are nearly identical, if the points are symmetrically located to the culmination on opposite sides. This may be verified easily by

introducing $Q_1 = -Q_2$ in both equations.

5.2 Orbital Element Improvement

When data from several passes are available, the time shift values may be plotted versus time and extrapolated to estimate the time shift of future passes. More convenient and powerful is a least square adjustment of mean anomaly coefficients using data from all passes simultaneously. For this, the routine "IMPROVE" is included in the analysis program. It uses the first and last point from each pass and minimizes the least square sum of range by standard iterative methods. For each pass, the orbital program has to be called 4 times to evaluate the time derivatives of range consuming about 3 sec. of computing time. Thus for 20 passes about 1 min. is necessary per iteration. Two or even one iteration are sufficient in most cases only.

Let us consider as an example GEOS C using data collected during the spring of 1980. In the following table time shifts (ms) determined by the two point method (Sect. 6.1) are printed as a convenient measure of the along track error. In the first column original SAO elements (reference epoch 44358) are used. It is obvious, that the prediction center used observations from the period of about 2 weeks before the reference epoch. Outside of this period the prediction error increases strongly. For comparison, in the next column the time shifts relative to AIMLASER-predictions are printed. These shifts were calculated treating the ranges predicted by AIMLASER like measured ones in formula 2 of Sect. 6.1. Not for all the passes AIMLASER predictions were available but it is obvious, that an almost constant shift of about 100 ms. exists between the programs. This indicates that the strongly increasing prediction error is mainly due to the SAO elements itself. The opposite sign of the time shift in the past and future, respectively, is caused by the 3rd order term in mean anomaly. Removing this term, the third column results. The divergency is reduced somewhat, but remains strong. In the next 2 columns the time shifts are given for elements updated using all passes before MJD=44359.9 and using the third order term or not. It is seen that the 3rd order polynomial gives slightly better extrapolation behavior in this case. Current practice shows, that the most stable method seems to be the use of 2nd order connected with frequent readjustment of the mean anomaly coefficients as soon as new data are obtained. As an example in the next column the results of fitting to passes up to MJD=386.9 (pass No. 1 to 18) are given. Second order is much better in this case. Using frequent updating, we used the same SAO element set for 6 weeks or even more with no significant loss in accuracy.

5.3 Comparison to AIMLASER

Let us now compare the two programs more directly than in the last section. GEOS C is again considered, because it is the most critical laser satellite with respect to predictions. In Table 2, angular differences are given in seconds of arc and in addition range differences in meters. In the first set, original elements are used, in the second a time shift of 67 ms. is introduced in the Potsdam program.

Table 1: Time Shifts for Different Variants of Improvement.
GEOS C. Reference Epoch: 44358

No	MJD	Orig. elem.	AIM-Po	M3=0	M3=0 No1-10	M3#0 No1-10	M3=0 No1-18	M3#0 No1-18
1	334.8	2210	92	-3199	- 14	- 12	- 47	- 9
2	334.9	2114		-3151	- 30	- 28	- 61	- 25
3	335.8	1832		-2885	- 17	- 19	- 34	- 24
4	344.8	210	131	- 773	12	8	67	2
5	344.9	243		- 740	46	42	102	37
6	345.8	147	135	- 629	- 1	- 3	54	- 4
7	349.8	93	131	- 140	7	10	48	27
8	350.8	63	97	- 94	- 30	- 24	4	- 6
9	358.9	76	113	77	5	7	- 76	- 18
10	359.9	25	95	28	7	5	- 93	- 35
11	361.9	- 119		- 94	26	15	- 120	- 69
12	362.9	- 139	139	- 90	118	101	- 47	- 7
13	363.8	- 275		- 190	126	103	- 70	- 39
14	363.9	- 202	144	- 113	213	188	13	43
15	364.9	- 346		- 207	236	204	11	25
16	365.0	- 314	139	- 164	309	274	71	84
17	368.9	-1241		- 690	396	318	37	- 57
18	368.9	-1187	107	- 629	461	383	104	8
19	372.0	-2361	90	-1191	547	418	73	- 165
20	373.0	-2840	87	-1406	575	427	60	- 238
21	373.9	-3392	52	-1659	562	394	9	- 352
22	379.0	-6851	70	-2929	802	496	10	- 796
23	379.0	-6965		-2955	824	515	25	- 792

This table shows that the cross track angular differences are indeed very small. The along track differences are significant, but, as already pointed out in Sect 6.2, they are usually small compared to the real prediction error. Using updating of elements, long periodic effects are easily absorbed in the polynomial coefficients.

6 Coordinate Transformations Related to Star Observations

Star observations are necessary to check and maintain the pointing accuracy of the mount. This is becoming increasingly important in connection with narrower laser beams. The first two axes of the mount used at our station can be set with a precision of 1 minute of arc only. The mechanical quality of the step motor driven 3rd and 4th axes permits pointing to 5 arcsec. on the other hand. Thus, a possibility to achieve this accuracy would be the observation of a few reference stars immediately before the satellite tracking.

For flexible programming we found it useful to have a set of subroutines for the orthogonal transformations between the astronomical systems. The following systems need to be considered:

1. The equatorial system with x-axis to the equinox at 1950.0

Table 2: Comparison to AIMLASER
GEOS C, Reference Epoch: 44358

		No time shift			Time shift =67ms		
		Angul. along track	Diff. cross track	Range	Angul. along track	Diff. cross track	Range
80 4 12	20 33 45	85	14	-674	16	14	-426
	20 35 5	144	16	-401	53	16	-349
	20 36 45	148	12	97	73	13	-120
80 4 20	20 13 15	88	12	-825	30	13	-471
	20 15 15	221	13	-350	106	13	-284
	20 17 35	157	3	520	94	2	185
80 5 1	20 48 50	59	31	-338	-15	31	-118
	20 50 10	93	29	-163	0	29	-159
	20 51 50	87	21	114	18	21	-141
80 5 10	21 52 50	27	12	-327	-47	12	- 98
	21 54 10	59	10	-243	-34	10	-228
	21 55 50	68	5	- 50	- 2	5	-299
80 5 20	0 35 5	69	25	-697	0	25	-401
	0 36 45	158	27	-368	51	27	336
	0 38 45	135	18	237	67	18	- 61
	rms	118	18	424	52	19	276

2. The equatorial system with x-axis to instantaneous equinox

3. The horizontal system

4. The "SBG"-system defined by the setting of the 1st and 2nd axis of the mount.

For all practical problems it is sufficient to have transformations between all the 3 pairs of systems adjacent in this list (forward and backward). Rectangular coordinates are used throughout introducing no singularities in the vicinity of the pole. In the following we comment the approximation used for the transformation from system 1 to 2 and vice versa only. It is not a clean rotation of the coordinate system because of inclusion of seasonal aberration. The diagonal elements of the precession- nutation matrix are set to unity simply and the non diagonal are approximated by:

$$m_{12} = -m_{21} = -6.119E\text{-}7*t+7.67E\text{-}5*\sin(F)$$

$$m_{13} = -m_{31} = -2.6604E\text{-}7*t+3.33E\text{-}5*\sin(F)$$
$$m_{23} = -m_{32} = -4.47E\text{-}5*\cos(F)$$

$F = 12.11279^{\circ} - 0.0529539(^{\circ}/day) * t$
$t = MJD - 33282$ (days after 1950.0).

For the inverse transformation the signs of the non-diagonal elements have to be reversed.

For aberration, the simple vector relation holds:

$$dX_i = (V/c)(V_i - (V_k X_k) X_i), \quad i,k = 1,2,3$$

In this equation the vectors are represented by their rectangular coordinates using lower case letters as indices. The summation convention over equal indices occurring in a product is adopted. The used symbols are:

V = velocity of the station in the used inertial reference system
c = velocity of light
V_i = unit vector of station velocity
X_i = unit vector directed from the station to the star
$(V_k X_k)$ = scalar product
dX_i = aberration correction

In the equatorial system the direction of the orbital velocity of the Earth is approximately:

$V_1 = \sin(L)$
$V_2 = -\cos(L)\cos(\varepsilon)$
$V_3 = -\cos(L)\sin(\varepsilon)$

where:

L = longitude of the sun in the ecliptic
$\varepsilon = 23.4425$ = inclin. of the ecliptic

For L the approximation is used:

$$L = 360(t/365.242 - 0.222).$$

7 References

[1] M.R. Pearlman: "Updating Ephemerides for Laser Tracking Memorandum directed to SAO elements users June 1, 1979

[2] K. Arnold: "Methoden der Satellitengeodaesie" Berlin (Akademieverlag) 1970

[3] R. Neubert, I. Prochazka: "Software for Automatic Laser Satellite Tracking" Submitted to the Intern. Scient. Conf. Sect.6 INTERCOSMOS, Albena (Varna) ,Sept. 15-21, 1980

[4] R.Neubert: "Laser Radar Software (Listings, Examples)" Report ZIPE, Sept.1981

Software Package for Station SAO No. 7831

by

A. Novotny and I. Prochazka
Czech Technical University
Brehova 7
115 19 Prague 1, Czechoslovakia

ABSTRACT

This software package was developed for the satellite laser ranging station no. 7831 in Helwan, Egypt. It guarantees the computer control of the automatic tracking system. The simple method for on-site prediction correction is included. The computer hardware used includes:

- Hewlett-Packard 2100S processor,
- 64 KByte of internal memory,
- magnetic disc HP7900 with capacity of 5 MByte,
- standard I/O paper tape devices,
- the Multiprogrammer HP6940B and communication channel HP-IB (IEEE 488).

The RTE II real time disc operating system is applied. The laser radar electronics is connected to the CPU partly via the HP-IB channel, partly via Multiprogrammer special interfaces. The whole software package consists of four principal program blocks:

1. programs for satellite position prediction,
2. programs for system calibration and check,
3. satellite tracking and ranging programs,
4. postpass ranging data handling and analysis.

All the programs were written in FORTRAN IV language.

1 Satellite Position Prediction Software Package

For blind, fully automatic satellite tracking and laser ranging the position of the satellite must be predicted with the accuracy better than is the beam divergence of the laser beam transmitted: two arcmin for low-orbit and one arcmin for Lageos satellite. The standard algorithm of evaluation of the satellite position of the AIMLASER program is used [see 1,2]. This algorithm is based on the use of so-called Kozai mean elements, which are routinely provided by SAO and distributed via telex. To fit into the minicomputer HP2100 with only 16k words of user-available memory, the original program was significantly modified and divided into several separate programs. All the input/output procedures and transfers of the data from one program to another one are carried out via the disc, partly within the file structure, partly on the dedicated data region used for direct data access (DDA). Each of the programs may be used separately, in an interactive mode, or may be scheduled by the transfer file in sequence in the batch processing mode. The satellite position prediction software package was constructed to deal with the set of maximum 6 satellites. The standard ones are: Geos A, Geos C, Beacon C, Starlette and Lageos. The satellites are identified within the program by their numbers. To simplify this, only the first four digits of them are used. Any time the satellite identification number is mentioned in the following sections, only these first four digits of the original id. number are kept in mind.

1.0.1 Prediction Software Scheme

The prediction procedure consists of four groups of programs:

- Group 1 - culmination times evaluation programs

  * SEEKC: computes the culmination times for one satellite within given period of time.

  * ARANG: rearranges the culmination times for all satellites in chronological order, checks the daytime conditions.

  * CULM: computes the rough values of azimuth and elevation in satellite culmination and prints out a "time table" for the satellite ranging on the station.

- Group 2

  * PDCTS: computes the slowly varying coefficients for evaluation of perturbations due to tesseral harmonics.

- Group 3 - position prediction programs

  * AIMFL: computes position of the satellite in 10 points near the culmination taking into account all the perturbations available.

* SCHED: small help program which schedules the program AIMFL.

* ATS: computes the time shift parameter from the foregoing satellite ranging for prediction accuracy improvement.

- Group 4 - input/output data sets handling and help programs.

* ORBEL: interactive input of satellite orbital elements.

* CORCT: test of the data for orbital elements by means of four letter checkword provided together with data, correction of bad digits.

* TSHAR: reformating of the tesseral harmonics coefficients and storing them on the direct disc access area.

* TESFL: interactive creating of special tesseral harmonics set from Standard Earth III for special purposes.

* STATN: the station coordinates files manipulation program.

* INFOR: informs about the program, satellite name and program phase just being executed in CPU.

On the end of the prediction procedure the user gets the table of satellite passes for a given station, period of time and satellites and the table of universal times and positions of these satellites in the x,y,z coordinate system in 10 points within each pass. The actual position az/el/rang of the satellite is computed in the on-line procedure by means of the Chebychevian approximation.

1.0.2 Prediction software data structure

- ELWORK: file, created by the ORBEL program, contains orbital elements, which are just being entered.

- EL++++: orbital elements set for satellite with id. number equal to ++++ .

- ST****: station coordinates file for station with id. number equal to ****.

- SE++++: culmination times for satellite ++++.

- CTIMES: file with culmination times for all 6 satellites arranged in the chronological order.

- TTABLE: the "time schedule table" for satellite laser ranging on the desired station.

- DATA3: file with tesseral harmonics coefficients set, contains the

Standard Earth III coefficient set and special tesseral harmonics sets for some satellites.

- TRDATA: file with computed positions of the satellites.

1.0.3 Dedicated disc area structure

To simplify the program PDCTS and AIMFL, the direct disc access to several data sets is arranged. Totally, the first 21 disc tracks on the non-system disc cartridge are used for this purpose. They are protected against wrong manipulation by means of initialization of the corresponding cartridge from track no. 21. The scheme of the *DDA* area is on fig. 2. Generally, there exists 3 data blocks in the DDA region:

1. intermediate results for program PDCTS and AIMFL,
2. six different tesseral harmonics coefficients sets,
3. values of slowly varying coefficients for computation of perturbations due to tesseral harmonics for six different satellites.

2 Description of Programs

2.1 Group 1

2.1.1 Program SEEKC

It computes the times of culminations of the satellite above the given place on the Earth. The searching algorithm described in [4,5] is used. This algorithm seems to be a good compromise between the computing accuracy and the computing time required. Program SEEKC runs separately for each satellite. Input parameters date, no. of days, and satellite id. no. are entered via a RMPAR routine to the program. The program calculates all the possible culmination times during one day and stores these times in a linear array. Routine *RRRR* is responsible for it. Then the positions of the satellite are computed for these culmination times in the *SATPS* routine omitting all the perturbations. The elevation angles are then computed and compared with the minimal values required for corresponding satellites (default values are 50° for Lageos, 30° for all the other satellites). The culmination times, in which the satellites culminate above the limit, are stored on the file SE++++, where ++++ is the satellite id. number. The times are stored in the form of non-integer value of modified julian date. The program is started as follows:

RU,SEEKC,iyr,mo,idy,nday,nsat

where

- iyr,mo,idy.... is the date of beginning of prediction,

- nday..... is the no. of day,

- nsat.....satellite id. no.

2.1.2 Program ARANG

It rearranges the culmination times for all the satellites into one file in a time increasing order. All the files SE++++ are opened, read and stored in one two-dimensional array. The elements of this array are moved to another linear array in a time increasing order (thus, the smaller elements are moved earlier). The culmination times for one and the same satellite which differs only in some minutes of time are detected and replaced by a single value obtained as an average of these values. (This culmination time duality may occur for the searching procedure used.) Then, the culmination times are converted into the form of the date, hour and minute. Optionally, the daytime passes are suppressed. The sequence of the culmination times is stored on the file *CTIMES*.

2.1.3 Program CULM

This program computes the position (azimuth, elevation) of the culmination for satellite passes listed in the file *CTIMES*. Only the mean orbital elements are used, all the perturbation effects are omitted. As an output the program CULM prints out a table with the time schedule for satellite ranging. Rough position of the culmination is included to guess the observation priority in the case that several satellite passes overlap. When the program CULM is started in a batch processing mode, the results are stored on the file TTABLE. The print out sample of the file TTABLE is on fig.3. The program can be scheduled by the command:

RU,CULM,irec

where

- irec.... is the no.of record of the CTIMES file from which the procedure should start (default 1).

2.2 Group 2

2.2.1 Program PDCTS

This program computes the slowly varying coefficients for computation of perturbations due to tesseral harmonics. From the theory of Kaula [7], it can be gathered that analytical expressions, describing the short-period tesseral harmonics effects, consist of very slowly changing functions of the mean elements to be multiplied among others by several trigonometrical terms in the

tr ıe and the mean anomaly. It can be shown that the slowly varying terms may be considered to be constant for periods of time of the order of several weeks. Therefore, the so-called intermediate results are computed only once for one orbital elements set for the time period of the next two weeks.

Program PDCTS is composed of the main program and two program segments. The main program contains only the common block area declaration and the next segment call. First segment PDCT1 reads the satellite id. number and the optional print out parameter from the RMPAR routine. Then the appropriate orbital elements data set is read from the file EL++++ and the corresponding set of the tesseral harmonics coefficient from the DDA area. The epoch, for which the computation is carried out, is set to be equal to the epoch of the orbital elements. Optionally, the orbital elements, the tesseral harmonics coefficients and the convergence criteria used are printed out. Then the PDCT2 program segment is called. Segment PDCT2 calls, via some help routines, the function GRAHV for evaluation of the slowly varying coefficients. Finally, these coefficients are stored on the DDA area. Although the algorithm is the same as in the original Aimlaser program, several changes were to be made in the version for the minicomputer to reduce the memory size required and to overcome the problems of the restricted dynamical range of computation (only 48 bits in double precision in contrast with 64 bits on the CDC or the IBM computer).

1. The original function TESSRB was shortened to its initial part, to computing of the coefficients and storing them on the DDA area, only.

2. The two dimensional array CLMP, in original version dimension of 6x300, is not present in the computer memory in its original form. Only one column of it is present, while all the others are stored on the DDA area. For fast and effective handling with the elements of the matrix CLMP two special routines were created. Routine *CLMPD* reads the specified column of the matrix from the DDA area, the *CLMPT* routine stores the values of one column of the CLMP matrix on the DDA area. To reduce the number of disc transfers, the data are transferred to the disc not before the new CLMP matrix column is dealt with. Otherwise, the values are stored in a help 6 element array within the *CLMPT* routine.

3. In the GRAHV, BAT and FECK routines some changes were made, according to [3], to overcome the troubles with the restricted dynamical range of computation. Namely, in evaluation of binomial coefficients, factorial computation.

4. For sequential reading and writing of the results on the DDA area, two new routines were created: DWRIT and DREAD. The function of both of them is quite self-explaining.

Analogically to the original Aimlaser program, different reduced tesseral harmonics sets may be used for computation. Subroutine *SIZES* determines which of six sets present will be used. One additional output parameter was implemented to the original SIZES routine: NKAT. It denotes the sequence

number of the tesseral harmonics coefficients set within the catalogue (NKAT=1,...,6). Satellites Be-A, Ge-A, Starl., Ge-C and Lageos have the number 1 to 5, respectively, to all other satellites the tesseral harmonics coefficients set present on the sixth position in the catalogue will be assigned. On this position the Standard Earth III set or any reduced set (created by the special program *TESFL*) may be placed. Running the program:

RU,PDCTS,nsat,iprint

where

- nsat .... is satellite id. number
- iprint .. is optional print out parameter (default zero)

2.3 Group 3

2.3.1 Program AIMFL

It computes the position of the satellite in the Cartesian coordinate system x,y,z in 10 time points equidistantly spaced round the culmination time. The program consists of the main program and three program segments. The main program declares all the common block variables and calls the first segment AIMF1, only. In segment AIMF1 the satellite id. number, the date, hour and minute of culmination time is read from the file CTIMES. The number of the record to be read from this file is entered to the program via the *RMPAR* routine together with code number identifying the perturbations which are to be used in computation. Then the appropriate satellite orbit elements set, station coordinates and the tesseral harmonics coefficients are read from the corresponding files. The beginning time of the pass and the time step for position evaluation are calculated from the culmination time and the rough duration of the pass.

The second program segment *AIMF2* evaluates the positions of the satellite. With comparison to the original Aimlaser program, several significant modifications were made in the program structure to reduce the memory size required and to decrease the level of nesting of the routines.

For purpose of the segment AIMF2 the routine *INST* was written. In fact, it is separated fourth part of the *MYORB* routine, only. It evaluates the instantaneous values of orbital elements. In the segment AIMF2 the time for position computation is calculated, then the instantaneous values of the orbital elements are calculated within the INST routine. According to the original program version, the call to the function *ZATPZB* should follow. In AIMFL program, the routines ZATPZB, SIDTIM and TSSRLB were introduced directly into the program segment AIMF2. This modification saved roughly 300 words of the computer memory required at the expense of the program elegance. The original function *TESSRB* was shortened. Only the part, which reads the already computed partial derivative coefficients and evaluates the

perturbations due to the tesseral harmonics is used. This modified routine is called *TESE*. Within computation in TESE function and some help routines, the routines for the direct disc access are used analogically to the *PDCTS* program. Some small corrections of the algorithm to overcome the dynamical range problem were made as well.

The computed coordinates of the satellite are stored in the two dimensional array *UU*, the corresponding values of the time are stored in the array *VV* and they are transferred to the next program segment. To reduce the computer memory required, no formated input/output statements are used in this segment. All the I/O procedures are carried out via a common block from the first to the third program segment or by means of the direct disc access. Program segment AIMF3 stores the computed values on the disc file *TRDATA* only. The number of the record of this file, on which the data are stored, is equal to the record number of the input data file CTIMES for this calculation. Program AIMFL may be started by the command:

RU,AIMFL,irec,kode

where

- irec ... is the record number of the CTIMES file to be used (default 1)
- kode ... indicates which perturbation must be included. (default 7... all)

2.3.2 Program SCHED

This is the help program to schedule the program AIMFL in the sequence, using the operator command:

RU,SCHED,ir,kode

Program AIMFL will be periodically scheduled with *RMPAR* parameters irec equal to ir, ir+1, ir+2, ... sequentially. When the final record of the CTIMES file is found, this sequence is terminated.

2.3.3 Program ATS (Along Track Shift)

See Section 4.1.

2.4 Group 4 - Programs for input & output data file manipulation

2.4.1 Program ORBEL

Program for manual interactive input of the orbital elements of the satellites. The data are stored on the help file *ELWORK* in the format acceptable for the *MYORB* routine in all the prediction programs. The key word for the mean elements is supposed to be in the form: 2 4 6 8 12. If some of the elements required by the *ORBEL* program are not supported, they must be entered as zeroes. The program may be started by the command:

RU,ORBEL

Then, the full satellite identification number, the epoch, the argument of perigee, ... are to be entered. Entering the data into the file ELWORK by the ORBEL program, the file must be listed and the data checked. Then, the file ELWORK must be copied to the corresponding EL++++ file.

2.4.2 Program CORCT

Checks the data set with the orbital elements received from SAO via Telex using the four letter checkword [6]. The incorrect lines are printed out. The missing one or non trustful digits or signs may be replaced by "*" by the interactive editor program. In the next program CORCT pass, the correct form of the line will be found and printed out. Up to 4 digits may be corrected in one line, all the combinations of characters corresponding to the given checkword (usually only one) are printed out.

2.4.3 Program TSHAR

It reformats the tesseral harmonics coefficients sets from the file *DATA3* and stores them on the appropriate place in the second block of the DDA area. (see fig. 2) In fact, the *DATA3* file contains many special tesseral harmonics coefficient sets, but only the sets for standard satellites and the Standard Earth III are used, the remaining being ignored. The program was arranged for input from the standard input device (lu=5). If input from the disc file is required, the program must be executed in the batch processing mode with the appropriate starting file.

2.4.4 Program TESFL

This is an interactive program for creating of the reduced set of tesseral harmonic coefficients from the set placed on the sixth position in the second block of the DDA area (default, the Smithsonian Standard Earth III set). The results are stored on the corresponding position in the DDA area according to the satellite identification number. Optionally, the convergence criteria may be modified, too. The program may be started by the command:

RU,TESFL

Then, the identification number of the satellite and the convergence criteria are entered from the console. Then the pairs of the subscripts of the

tesseral harmonics coefficients to be included in the reduced set may be entered. The subscript pair 0,0 terminates the interactive procedure, the reduced set is stored on the DDA area.

2.4.5 Program STATN

This is the interactive program for the station coordinates files manipulation. The program creates the file ST####, where #### is the SAO station id. number. During the program execution the station id. number, name and cartesian coordinates are entered. Then the geographic latitude, longitude and the distance from the earth centre is computed and stored on the file. Running the program :

```
RU,STATN
```

The station files are created automatically. If the corresponding file already exists, it can be or modified or purged by this program.

2.4.6 Program INFOR

This is the help program, which informs the computer operator about the program name, segment number, satellite id.number and the program phase just being executed in *CPU*. As mentioned above, the satellite position prediction software is a rather difficult system of several segmented programs, some of them having no output or control message on the console for relatively long time (up to 5 hours for the PDCTS program). That is why, a simple method of display of the basical information about the program just running was introduced. The *S* register of the computer is used for this purpose. Its value may be set by the programs using the system function *ISSR* call. Independently, the value of the *S* register may be read into another program. Thus, each noninteractive program, which is executed routinely for the prediction procedure, sets on some value on the S register. The meaning of the bits in the register:

```
bits    0-5  ... program computing phase
        6-8  ... program segment number
        9-11 ... satellite id.number
       12-14 ... program name identifier
```

The program may be started by the command :

```
RU,INFOR
```

The program writes on the operator console name of the program, segment and satellite number and program computing phase just executed.

2.5 Preparing the Prediction

1. All the files required must be created. For this purpose the transfer file #FILES may be used.

2. The station coordinate file must be created by the STATN program.

3. The tesseral harmonics coefficients sets must be placed into the DDA area by the program TSHAR.

4. The satellite orbital elements must be entered to the file ELWORK. This may be done interactively by the program ORBEL or by reading of the telex tape with the elements by the program TLXR and by editing of the resulting file.

5. The elements on the file ELWORK must be checked simply by visual comparison to the original telex print-out or using the CORCT program.

6. The file ELWORK must be copied to the appropriated EL++++ file.

7. If another satellite is to be dealt with, go to item 4.

8. The program SEEKC must be started consequently for each satellite. Then,the program ARANG and program CULM must be executed.

9. Program PDCTS must be started consequently for each satellite.

10. Program SCHED must be started to schedule repeatedly the program AIMFL.

On the end of this procedure, the user gets in the file TTABLE table of the satellite passes over given station, period of time and satellites and in the file TRDATA the table of universal times and positions of these satellites in the x,y,z coordinate system. It is obvious, that for routine prediction on the station, the sequence listed above is started once a week from the item no. 4. For automatical scheduling of this sequence items 8-10 the transfer file may be used.

2.6 Prediction procedure computer requirements

From the point of view of computer memory and computing time requirements, only the programs PDCTS and AIMFL are of interest. Program PDCTS requires roughly 15k words of computer internal memory . The time required for computation strongly depends on the tesseral harmonics coefficients set used. For LAGEOS satellite the program requires roughly 0.5 hour of computing time, for GEOS-C satellite roughly 5 hours. Running the PDCTS program for all the standard satellites takes about 12 hours of computing time.

The program AIMFL requires 16k words of computer internal memory. The

computing time required for 1 satellite pass is 100 seconds. The time consumption of all the other programs is of order of seconds or minutes of time. Thus, the off-line prediction procedure for one station, 5 standard satellites and one week period of time, takes about 14 to 16 hours of computing time.

2.6.1 Computer operating systems

Generally, the HP2100S computer is used for two rather different jobs - for satellite position prediction calculation, and for on-line laser radar control. The operating system requirements for these two jobs are quite different as well. The former needs as large computer memory as possible, the latter many special input/output facilities, foreground program availability, etc. To fulfill these requirements, two different operating systems were generated, they are denoted as "A" and "C" system for calculation and control, respectively. In practice the system "A" is used for the program AIMFL run, only. All the other programs, including the PDCTS program, may be executed in the "C" operating system.

The systems may be exchanged by the SWTCH system program. No disc cartridge exchange is required. The "A" system is routinely loaded only once a week, to compute the prediction for the next 7 days. The most time consuming program - PDCTS may be executed in the system "C" within the time gaps between the satellite laser ranging passes. This way, the prediction procedure for one week may be completed within one day (24 hours) on the station computer without affecting the plan of observations.

3 Control Software Package

This control software package was developed for the computer hardware described in section 1.1 and the laser radar hardware consisting of :

1. Laser clock: a multipurpose device which serves as a time central for the whole laser radar. The data to and from the laser clock are transferred via a HP-IB channel (IEEE-488 or IMS-2 standard).

2. Ranging counter: the HP5360 counter with time interval measurement unit HP5379A is used. The measured data are transferred from the counter to the computer via a special interface in the grammer device.

3. Time gate: resolution 100 ns, the window width and the delay time are programmable via a standard 8-bit communication interface in the Multiprogrammer device.

4. Step Motor Control Unit: (SMCU) the control device generating the sequences of pulses to control the step motor power supplies for continuous satellite tracking.

The SMCU is connected to the computer via a HP-IB channel. The data transfer

to and from the devices via a HP-IB is ordered by means of ordinary input/output commands to transfer the ASCII characters (formated READ/WRITE commands in the FORTRAN IV language). For data transfers via the Multiprogrammer device the special routines were written: the *GATE, *BUFR*, RESET ... routines for time gate device programming and data collecting from the range counter, respectively. The function of these routines, using the description comments on the heads of them, is quite self-explaining.

3.1 Programs for system calibration and check

3.1.1 Program TIME

It is an interactive program for programming and set-on of the laser clock unit. There is a closed loop inside this program, the user can choose one of the seven program options:

1. Start the laser clock. The starting values (hour,minute) are entered from the console. The clock is started by the next coincidence of the "1 min" and "1 sec" pulse from the station master clock.

2. Reading and display of the current value of the time from the laser clock and the computer internal clock.

3. Synchronising of the computer internal clock to the laser clock. The former is synchronized by help of the system command TI executed through the function MESSS. The computer internal clock may be synchronized with accuracy bether then 10 ms.

4. Programming of the repetition rate for the laser trigger, the laser period 1 to 9 seconds may be set on.

5. Laser control - by the following characters ASCII :

   - A ... start of the HV power supplies,
   - B ... stop of the power supplies,
   - T ... single shot trigger of the laser,
   - Q ... start of the lasing sequence with period programmed on the item 4,
   - R ... stop of the lasing sequence.

6. The laser clock synchronization check. The epoch of arriving of the one second pulse from the cesium master clock is determined. The fractional part of second of this epoch determines the time shift between the cs.clock and the laser clock.

7. The end of the program.

3.1.2 Program KALIB

It is the program for the fixed target ranging calibration procedure. It can be started by the command:

RU,KALIB,iopt,lu

where

iopt ... option code 0...calibration only(default)
1...precalibration
2...postcalibration
lu ... number of the lu output device for the result list. (default 1)

The program starts the laser power supplies, sets the time gate and completes the total of 23 laser rangings to the target. The ranging results, which differs more than ±20 ns from the usual value are rejected, the error message being transferred to the operator console. The mean range value and the RMS are calculated and displayed. If the option code equals 1 or 2, the values of the temperature, air humidity and pressure must be entered. The mean range value and the atmospheric data are then stored on the record no. 2 of the file RANGE or the file RANG** for pre- and post-calibration, respectively.

3.1.3 Program MOUNT

It is an interactive program for the mount positioning, alignment and pointing accuracy check. The mount motion is controlled through the step motor control unit (see 9). This device generates the sequence of pulses for the step motors power stages. The frequency of stepping is obtained by the digital frequency division of the 1 MHz frequency from the high stability quartz oscillator. The division ratio is programmable, together with some other functions of SMCU, via a HP-IB channel. There are two counters within the SMCU, which count the total number of steps already generated. The stepping frequency of the both drivers is set on each one second.

There exists a set of several routines, which are available for the mount control. Except for the direct SMCU programming, the algorithm is identical for both the step drivers. All the data transfers within this set of control routines is carried out via one two dimensional, double precision array *DM*.

- Subroutine STEPS: the principal routine, which computes the division ratio to be programmed for the mount setting from the initial to the final position. Inside this routine the stepping frequency is evaluated in such a way, that it is from the interval between the minimal and the maximal frequency, the resonant frequencies of the

mechanics and the maximal acceleration allowed is taken into account. The algorithm is resistant to the rounding errors. The meaning of the variables used is described in the head of this routine. (This description is valid for the input/output values only. Within the computation, the variables may be used to other purpose, too!)

- Function MAXA: its value is equal to the maximal acceleration allowed for current frequency of stepping. In the first call, the value of MAXA is equal to 10 to guarantee proper start of the drivers.

- Subroutine REZON: it checks the input stepping frequency with respect to the the resonant frequencies. If the input frequency falls within the resonant region, the output value of frequency will be set lower, then is the lower limit of the resonant frequency region.

- Subroutine SECND: it guarantees the time synchronization of the SMCU programming with respect to the synchro pulses from the laser clock. When this routine is called, it periodically tests if the new synchro pulse was detected in the SMCU. As soon as it occurs, the test is terminated and the control returns to the calling program.

- Subroutine CTI: reads the values of the built-in step counters. Calling the CTI routine, the caller gets the values of the step counters, which were achieved in the moment of arriving of the last synchro pulse.

- Subroutine DRIVE: guarantees the communication with the SMCU. It has 3 separate parts, which are executed according to the value of the input code: the initial reset of the SMCU, the start of stepping and setting the stepping frequency, stop of the stepping.

The routines STEPS and DRIVE are called each second to compute and to set on the division ratios for the SMCU. Consequently, the other help routines CTI, MAXA, SECND, REZON are called as well.

- Subroutine POSIT: is the main routine, which calls periodically the STEPS routine. This routine guarantees the positioning of the mount from one position to another one at the shortest time possible. The input values are the relative shift angles for both axes (in degrees) only. The positioning procedure from one position to another one is carried out in three phases:

  * first phase: the highest rate acceleration up to the maximal speed, motion with the maximal speed for N seconds until one half of the motion required is completed.

  * second phase: motion with the maximal speed for the next N seconds.

* third phase: the highest rate decreasing speed down to minimal start-stop frequency, positioning up to the desired final position.

Then, the total number of steps carried out in each coordinate (the content of the step counters) is compared to the value required according to the input. The difference is compensated by the "single step mode" stepping. Running this program:

RU,MOUNT RU,MOUNT

The program has several user selectable options: definitions of the starting and the final position of the mount and a direction of revolving of the mount round the azimuthal axis (not to twist the cables and pipes inside the mount too much). The fixed, preprogrammed, initial/final coordinates of the position reference target, ranging target and the reference stars position may be used or any other coordinates may be entered manually from the console. The option "aim to the reference star" may be used for absolute accuracy check.

3.1.4 Program FVT (Field of View Test

This is the program for the receiver efficiency and the field of view test. The program controls the mount in such a way that it scans a small area round the initial position. Simultaneously, the receiver noise frequency is recorded. The scanning step is 0.01 degree in the azimuth direction and 0.075 degree in elevation. The current value of the noise frequency is displayed on the console. Scanning the area, the results are printed out together with simple graphs of dependence of the receiver noise on the coordinate in the form of densitographs. As a reference light source the POLAR star may be used. Measuring on the 1PE detection level, the receiver channel overall efficiency may be determined from the difference between the maximal and minimal values of the noise. Running this program:

RU,FVT

3.2 Satellite tracking and ranging programs

3.2.1 Program SATEL

This program controls the laser radar system during the satellite ranging procedure. It is composed of the short formal main program and three program segments for the prepass radar preparation, ranging and postpass radar control. The data for the satellite automatical tracking are evaluated on-line in this program. As the input data, the results of the AIMFL program are used. The computed positions of the satellite in the x,y,z coordinate system are approximated by the Chebychevian polynomial [3]. For computing of the position of the satellite at any time during the pass, the corresponding value of this polynomial is evaluated for the time required taking into account the time shift, computed by the program ATS. The resulting position in the x,y,z

coordinates is then transformed to the azimuth, elevation and range. The routines INTER, AZALT, SIDTM, JNTMT and DASIN are responsible for this computation. The INTER routine has two program options, controlled by the KODE variable. For KODE = 1, the actual beginning of the pass and the associated values of azimuth and elevation are computed. The time is rounded to two tenths of seconds. This option is used in the prepass set procedure in the first program segment. For kode = 2, the azimuth, elevation and range of the satellite are evaluated for the given time. This option is used in the on-line satellite tracking procedure. The program SATEL may be started by the command:

RU,SATEL

Then, the sequence number of the satellite pass must be entered. The tracking input data, computed by the AIMFL program, are read from the corresponding record of the TRDATA file. The station coordinates are read from the file. Then, the actual beginning of the pass is computed together with the starting position for the tracking. The time gate window width and the initial value of the along track tracking delay must be entered.

The proper function of the laser clock unit is then tested, the availability of the help program TRACK is checked. The mount is set to the initial position with help of the POSIT routine. Each 10 seconds, the message to the operator console about the time left to the start of the pass is sent. Twenty seconds before the start, the first program segment is completed and the segment SATE2 is loaded and started.

The segment SATE2 locks the computer to be unaccessible for other foreground programs. The laser power supplies are switched on, the SMCU step counters are reset. At the moment of beginning of the pass, the closed program loop of on-line control is started. The loop runs with the period of lasing - once in 4 seconds. Within it, the laser is triggered, the help program TRACK is scheduled, the ranging results of the foregoing cycle are stored on the disc file RANGE, the value of the step counters is read and compared to the value required and the optional correction of the stepping is introduced. The ranging results: the epoch of the laser fire and the propagation time are read from the laser clock and the range-counter, the range to the satellite is evaluated for the epoch of lasing in the INTER routine, the difference (prediction - measurement) is displayed on the console. Then, the positions of the satellite within the next 4 seconds are evaluated and the corresponding control commands for SMCU are created, the range counter is reset. Then, the loop is scheduled again. This is repeated until the satellite does not decrease below the observation limit(28 degrees).

The programming of the SMCU in this segment rather differs from that in the MOUNT program or the segment SATE1. As the period of ranging (4 sec.) differs from the period of the SMCU control, special algorithm was to be applied. The SMCU is not programmed directly from the program segment SATE2, but via a small help program TRACK. This program transfers the already created control commands to the SMCU only. Program TRACK is loaded as a foreground program, thus, it can be present in the computer internal memory together with the background program SATEL/SATE2. The control commands for the SMCU are

created in the SATE2 program segment for the next four one-second intervals. They are transferred to the TRACK program via a system COMMON block area. The TRACK program is scheduled to run each one second with the highest priority.

Due to the mechanical construction of the mount, there exists a small near-zenith area of the sky, to which the mount can not be aimed. Subroutine CULM guarantees, that the step drivers will not try to set the mount to this position, but to the nearest available one. The execution of the program loop in the SATE2 program segment may be broken manually by the operator command:

BR,SATEL.

To compensate the along track prediction error, the time shift may be introduced into the tracking procedure. The S register switches are used for this purpose. Setting on the bit 0 or 1, the tracking delay will be increased or decreased at one elementary time interval, respectively. The elementary tracking time interval is evaluated with respect to the satellite angular velocity and the laser beam divergence used.

When the program loop of the SATE2 program segment is terminated, the laser HV power supplies are switched off, the motion of the mount is smoothly stopped, the next program segment is loaded. Program segment SATE3 stores the ranging results on the disc file RANG**, where ** is two digits sequential number of the file, generated automatically. Then, the mount is positioned to its initial position and the program is terminated.

4 Programs for Postpass Ranging Data Handling and Analysis

4.1 Program ATS (Along Track Shift)

This program evaluates the along track time shift between the satellite position predicted and the actual measured value. This parameter may be used in the next automatical tracking of the same satellite. The formula derived in [4] is used for the time shift calculation. As the input, two satellite rangings, together with the corresponding epochs are used. The predicted range and the radial velocity of the satellite for these two epochs is then evaluated by the algorithm identical to that in the satellite tracking program. The routines INTER, AZALT, etc. are used ,as well. The program may be started by the command:

RU, ATS

The identification number of the satellite pass must be entered together with the two epochs and measured distances of the satellite. To increase the accuracy of evaluation, it is recommended to use the points of roughly the same satellite range, one before and the other after the culmination. The computed time shift in seconds is displayed on the operator console.

4. Program RDF (Ranging Data Fit)

This is an interactive program for the ranging data handling, noise rejection, ranging accuracy check, etc. The program fits the measured ranging data with a polynomial of chosen degree (1 to 8) and prints out the difference between the measurement and the fitting polynomial. Optionally, the "range to prediction residuals" may be fitted. This option is highly appreciated in the case of very low signal to noise ratio. For example, using the 3rd order polynomial, the residuals for 6 minutes Ge-A pass may be fit up to about +/-30 nsec. This way, the noise points may be rejected effectively. The total number of the rangings and the standard deviation is printed out as well. The program may be started by the command:

RU,RDF,lu,iout

where

lu ... is the no. of terminal (default console)
iout... is the no. of the output device(default lu)

4.3 Program RDT (Ranging Data Transmission)

This is the program for measured ranging data transmission in the SAO standard quick-look data format. The program reformats the ranging data files into the file TELEX. This file may be then punched out in the telex code by the TLXW program. Starting the program by the command "RU,RDT", the user may choose if the quick-look only or the whole output file is to be created. (In the quick-look format only 15 measurements are selected from the full ranging data set.) Then the number of the ranging data file must be entered. The no. 00 terminates the program.

4.4 Program TLXW (TeLeX Write)

This is the program for punching of the paper tape in the standard telex code. It may be started by the command:

RU,TLXW

The user must answer the name of the file, from which the data for punching are to be read.

5 References

[1] Smithsonian Standard Earth, SAO special report no.200

[2] B.Ambrosius, H. Piersma, K. F. Wakker, Description of the Aimlaser

Satellite Orbit Prediction Program and its Implementation on the Delft University IBM Computer, Report LR-218, Delft, May 1976

[3] P. Wilson: private communication

[4] R. Neubert, I. Prochazka, A Simplified Satellite Orbit Prediction Program FJFI-CVUT, Prague 1979, Report no. 79/83

[5] Arnold, Methoden der Satellitengeodaesie, Berlin, Akademie-Verlag, 1978

[6] M. R. Pearlman, Memorandum, Updating Ephemerides Software for Laser Ranging, SAO, 1 June 1979

[7] W. M. Kaula, Analysis of Gravitational and Geometrical Aspects of Geodetics Utilization of Satellites, Geophysical Journal, Vol. 5, 1961

[8] I. Prochazka, P. Sobek: Step Motor Control Unit for Continuous Satellite Tracking, FJFI Report, no. 81/130, Prague, 1981

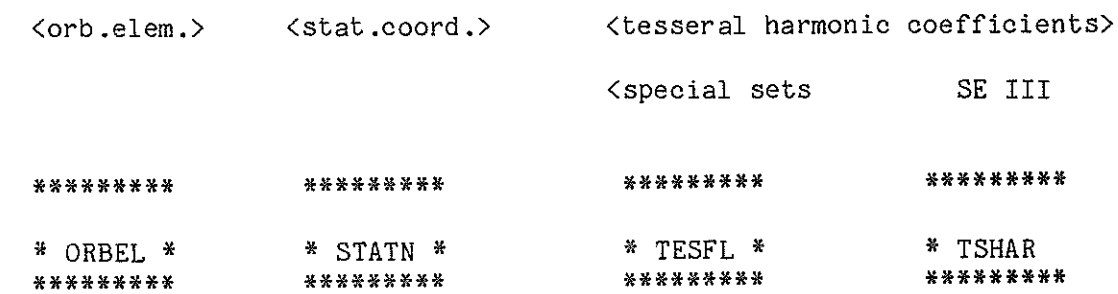
I. Appendix A

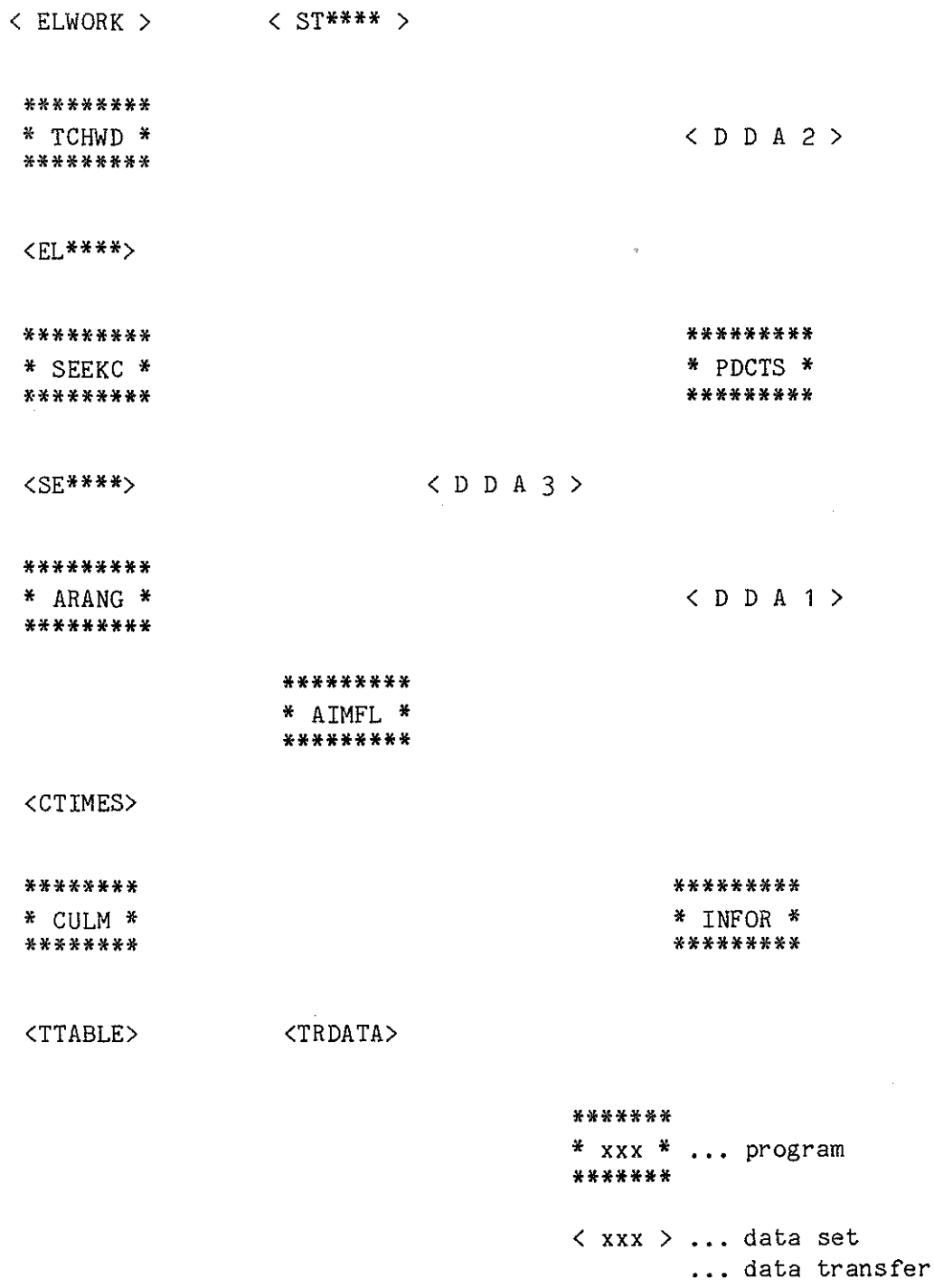
Dedicated disc area (DDA) structure :

track	beg.sector	satellite	content
0- 1	0	all	CLMP matrix.
2- 3	0	Beacon c	slowly
4- 5	0	Geos a	varying
6- 7	0	Starlette	coeff.
8- 9	0	Geoc c	for t.h.
10-11	0	Lageos	perturb.
12-13	0	spare	calcul.
14	1/10	Beacon c	A/ITESS
15	1/10	Geos a	A/ITESS
16	1/10	Starlette	A/ITESS
17	1/10	Geos c	A/ITESS
18	1/10	Lageos	A/ITESS
19	1/10	spare	A/ITESS
20	not used		

fig.2

SATELLITE POSITION PREDICTION SOFTWARE SCHEME




< ELWORK >
< ST**** >
****
* TCHWD *
****
< D D A 2 >
<EL****>
****
* SEEKC *
****
****
* PDCTS *
****
<SE****>
< D D A 3 >
****
* ARANG *
****
< D D A 1 >
****
* AIMFL *
****
<CTIMES>
****
* CULM *
****
****
* INFOR *
****
<TTABLE>
<TRDATA>
****
* xxx * ... program
****
< xxx > ... data set
... data transfer

Multiprocessor-Based Mobile System Software Design

by

K.H. Otten
Working Group for Satellite Geodesy
Kootwijk
Delft University of Technology
Delft, The Netherlands.

ABSTRACT

Considering that the satellite tracking activity can be divided into three major tasks, a choice of three separately operating computer systems has been made. The predictions of the satellite points as well as the corrections are supplied to the servo system of the telescope by the predictor micro processor. The observation data is collected and formatted by a second micro processor and transmitted to a main processor. The main processor hosts a monitor program during observation activity. The monitor program takes care of data transmission and schedules real-time tasks. Programs for initial satellite predictions and observation data screening are included in the main processor's application software package. A brief description of the system hardware is given to clarify the interaction with the computer configuration.

1 Introduction

The laser ranging system described in this paper is a mobile one and therefore can be brought into action in remote locations. This requires it to be almost completely self-supporting for a certain period of time. Concerning the observation activities, this means that the system must be equipped with all facilities to do a maximum number of valuable measurements (on even low satellites like STARLETTE) within a short campaign and with few contacts to the outside world as possible. To meet these requirements a considerable amount of time has been and still will be spent on investigation and development. The hardware system has been designed in more or less conveniently arranged segments with only a few interconnections and a great deal of independence, which will be helpful in locating errors during malfunctioning. For low satellite tracking (STARLETTE), a method has been developed to calculate orbital elements, valid for a whole week of observations and needing no more than seven numbers of data transmission for one pass. Detectors, indicating the rough position of the satellite in the telescope's field of view and the range-gate window, will aid in finding and tracking the satellite.

2 System Description and Functioning

In general, the ranging system consists of four compound units, namely:

- the telescope with the optical system and the laser,
- the mount positioner and laser control unit,
- the detection system including the station UTC-clock,
- the computer configuration with its peripherals.

The mount control and detection units will be described briefly, only to clarify the interaction with the computer configuration, which is the main subject of this paper.

2.1 Mount Positioner and Laser Control Unit

The mount positioner and laser control unit has the responsibility to direct the telescope to the object to be observed and to enable or disable the laser. At this point it may be mentioned, that the laser is continuously triggered at a 10Hz rate during ranging operation.

-Servo system.

The telescope's axes are driven by a servo system using DC-servo motors. Speed and orientation information for both axes are received from the prediction processor at a rate of 10Hz. Every tenth of a second, the actual and predicted positions are compared and differences will be automatically

corrected for. This feature makes special measures unnecessary for the observation of zenith passes. When the speed predictions are set equal to zero, the system chooses its own speed and stops at the desired position.

-Shutter.

A shutter is located in the oscillator cavity to disable the laser, without influencing the operation of the laser itself.

2.2 Detection System

The detection and the optical system are the most vital parts of the ranging system: they offer the means to find and track the satellite, to improve the measurements and to meet some safety requirements (Visser,1981).

-Position, time and meteo-data keeping.

For completeness, it must be mentioned that firing time (UTC), meteorological data and the actual position of the telescope are recorded. Some of the information will also be used during real-time operation of the computer system.

-Simultaneous calibration timer.

A fraction of the laser beam will be reflected after passing the start detector and is directed to the main photo-multiplier (PMT), which in turn stops the simultaneous calibration timer. This gives an indication of the stability of the basic detection system.

-Time interval counter.

The received signal will be attenuated to the single photo electron level by reflecting a part of the beam to the quadrant detector and transmitting the remainder to the main PMT which stops the time interval counter. This counter will deliver the round travel times to the satellite.

-Quadrant-detector timers.

The reflected part of the received signal, described in the previous section, is split up into four equally sized segments and directed to separate PMT's, which stop the quadrant detector timers. These time intervals are related to the start of the window and are expected to give an idea, in which quadrant of the field of view and where in the window the satellite might be found.

-Multiple-stop timer.

The multiple-stop timer consists of four, serially arranged timers,

linked to the main PMT. This configuration allows the recording of four events, relative to the beginning of the window. Speeding up of radial searching of the satellite's position or improvement of the remaining tracking parameters after the satellite has been found, is possible.

-Safety detectors.

An aircraft and a sun detector will be installed to meet government safety regulations and to protect sensitive system components. There will be no discontinuity in tracking when the telescope is passing the sun (although no observations are made).

2.3 Computer Configuration

In analyzing the tracking activity, which is the most critical task the computer system has to support, it can be easily seen that division into three more or less logical parts of cooperating functions is possible:

- final prediction and correction of direction and speed of the telescope,
- collection and formatting of observation data,
- monitoring and data recording.

Considering this division, a choice of three separately operating computer systems, each of them executing one task, is desirable. Since the prediction and formatting sections are quite unlikely subject to changes and not too complex, they will be installed in PROM and running on relative small 8-bit micro-processors: the predictor and the formatter. Unfortunately, micro's are slow in handling floating point data. Therefore a fast floating point processor has to be added to the predictor to handle this type of computations. The two micro's will be slaved to a 16-bit main computer system. The main processor will support extensive software for program development and data-handling. Knowing that computers are overloaded within a short period of time after installation, a choice of the main system will be made very carefully. To enable data communication, the predictor and the formatter are connected to each other by 8-bit bi-directional parallel data lines. The predictor is linked to the mount positioner and the formatter to the detection system. The main processor will have a HP-IB controller, used for data transmission to and from the slave processors. The main processor may be extended by a number of processors, if necessary. As peripheral devices, there will be two diskette units, a graphics display console and an optional KSR-terminal with an acoustic coupler, all of them interfaced to the main system. The UTC-time source will be used for epoch timing and synchronization of computer operation by means of a 10Hz interrupt supply. Both slave processors, in cooperation with the main processor, will execute test routines during system start-up/reset to guarantee the functioning of data communication and the processors themselves. Also the firm-ware will include defined checkpoints, which can be initialized during test mode of operation.

At these points the micro processors will transmit intermediate computational results and memory dump data to th main processor.

-Main processor.

In the computer configuration, the main processor must have the property of a general purpose development system with multi-programming/tasking capabilities, offering all the facilities for real-time and off-line operation. In the basic concept, the main processor will host the monitor program only, which initializes the slave processors, handles their status and message information, supplies the prediction processor with the predictions and corrections, receives the observation data from the formatter and records it on a diskette. This leaves most of the capacity of the main processor for real-time programs, which will be used to improve the ranging system's adjustments. Years of satellite ranging have shown, that supporting software is changed very frequently. Because the monitor is the only more or less time-critical program in the main processor, replacing and adding real-time software can easily be done without disturbing the tracking operation itself. In times, when the main processor is not involved in satellite ranging, it will be used to run application programs for astro-positioning, initial prediction calculation, data validation and general utilities.

-Prediction processor.

The prediction processor provides the mount positioner at 10Hz frequency with the desired direction of the telescope (azimuth, elevation), speed of the two axes, range-gate delay, window and laser control information. Because this action is very time-critical, it is controlled by an interrupt routine, which is initiated by a 10Hz signal from the UTC-system clock. The interrupt routine receives the UTC-time and status information from the formatter. Also it transmits the prediction status to the formatter. The UTC-time is used for internal time-keeping and -verification. Two modes of pointing are supported by the firm-ware, i.e. tracking and positioning. In positioning mode, the final predictions for the mount positioner will not include speed information. This allows pointing to a fixed target for a certain period of time. The initiation of the telescope's positioning can be done automatically by time values included in the predictions or manually by operator intervention. When the processor is running in tracking mode, the interrupt routine will be provided with updated pointing information before an interrupt occurs, whereas in positioning mode it receives one prediction for every predicted telescope direction only. The predictor also has the option to enable or disable the laser during ranging operation by using the information from the formatter, which indicates whether the telescope's orientation is within limits of the predictions or not. The laser will be enabled in positioning mode, when the telescope is pointed at the desired target and in tracking mode, when the minimum elevation is exceeded. After start-up, the predictor is idle, with only the interrupt routine working, waiting for initial predictions. When an input-request is received from the main processor, it reads a complete set of topocentric predictions for one observation activity, i.e. tracking one satellite pass or pointing to one or more targets. Tracking predictions are divided into groups of four satellite points, enclosing three subsequent intervals. For every group, a right-handed rectangular topocentric coordinate

system is defined with the x-axis pointing to the first satellite position and the y-axis being in the plane of the horizon. The remaining satellite positions are transformed into this system and the mean direction of the satellite track is determined. Using the transformed coordinates, the coefficients of a third degree Newton collocation polynomial are calculated. The elements of the rotation matrix (azimuth and elevation), the track direction and the polynomial coefficients replace the original predictions. This method allows real-time interpolation of the direction at tenth of a second intervals with no more than 0.001 degree deviation relative to the initially predicted track. For the range-gate delay, the same type of polynomial is used, resulting in a deviation of no more than 5 ns. The accuracy of the interpolation does not really require a third degree polynomial. However with a lower degree collocation polynomial the available RAM would have been exceeded (more groups per satellite pass). The predictor also applies along- and across-track, wire, delay and window corrections, which are received from the main processor. When the observation activity has ended, the system is idle again.

-Formatter processor.

Like in the prediction processor, an interrupt routine is handling the data communication to the hardware system. At a 10Hz rate, the detection system transmits the UTC-time, telescope direction, meteo-data and the various counter- and time-information to the formatter, even if there is no observation activity. The formatter compares the actual and the predicted direction of the telescope. The result of the comparison together with the time information is transmitted to the predictor. After the formatter has been initiated for an observation activity, relevant data is selected from the received information, formatted and collected in blocks of 1.6 second time periods and placed into an output buffer, to be transmitted to the monitor. To meet a delay in the response of the main processor, the buffer can accommodate several blocks.

3 Software Support

The software of the mobile ranging system must support all activities, which range from astro-positioning to quick-look selection. Careful segmentation and structuring is essential to facilitate future replacement and improvement of the programs. Interaction with the operator will be kept to a minimum, because the station crew may consist of less experienced personnel.

3.1 Astro-Positioning and Orientation

When installed on-site the first time or periodically, it will be necessary to determine the azimuth reference of the telescope and the latitude and the longitude of the ranging system. Preliminary values, obtained from e.g. compass and maps will be corrected iteratively using star observations.

-Star selection.

A catalogue, containing FK3 stars with FK4 accuracy, stored in a random-access file on a diskette will be searched for suitable stars. Depending on the type of parameter to be solved for and adjustment method to be used, the stars must satisfy some conditions for the time of observation, e.g. they must be near the prime vertical or the elongation and be the brightest stars in the near surrounding of the telescope's direction to ease the observation. The star coordinates will be transformed into the topocentric system and stored on a diskette in a manual or automatic positioning mode prediction file.

-Observation of stars.

During the observation of the stars, the ranging system is operating in the real-time mode. The star positions are transmitted to the prediction processor and pointing of the telescope to the stars is initiated automatically or manually by the operator. The observer at the telescope's eyepiece has the ability to correct the direction of the telescope by hand-paddle and to trigger the system by push-button. The observation data is recorded on a diskette.

-Parameter approximation.

The parameters are solved for separately in a non-linear least-square approximation. It might be necessary to repeat the whole cycle before the desired accuracy of the parameters is obtained.

3.2 Satellite Tracking

In this section the system software involved in activities necessary to enable satellite tracking in a remote location will be described. It will include software, which already has been developed and tested and software, that will be implemented after some experience has been gained during operation.

-Orbital elements.

Once a week, for every satellite and pass, a new set of osculating orbital elements is received via public telecommunication facilities and stored in the bubble memory of the KSR-terminal. This message is retransmitted to the prediction center for a transmission check and recorded on a diskette of the computer system, ready to be used by the prediction program.

-Satellite predictions.

The prediction program selects the orbital elements from the diskette, calculates the topocentric coordinates of the satellite positions for one pass and writes it onto an empty diskette. A fourth-order Cowell integration method is used to integrate the equations of motion in a gravity field comprising coefficients up to degree and order 4. The predictions are produced with a 5 or 30 seconds time-interval for STARLETTE or LAGEOS respectively,

rea y to be used for the real-time interpolation operation of the prediction processor (Vermaat,1981).

-Real-time observation scheme.

The real-time operation is initiated by the monitor program, when it transmits the predictions to the prediction processor. From this moment on all observation data is recorded on the same diskette the predictions are read from, except the information from the quadrant detector and the multiple-stop timer, which will be available in the main processor only. When after system calibration tracking has started, the data from the quadrant detector and the multiple-stop timer is displayed on the console to aid in finding the satellite. Corrections will be applied to the predictions, until sufficient returns have been received from the satellite. Subsequently a task is scheduled to select the returns. These observations will be used in a least-squares approximation of Keplerian orbital elements, to extrapolate the range-gate delay and to possibly correct the range-gate delay and window. The information from the quadrant detector will be helpful to correct the predictions in such a way that the satellite stays near the optical axis of the telescope. The real-time operation is terminated after the post-calibration has been completed and corrections have been saved. These corrections will be relayed to the operator as initial values to be used for the next pass.

-Data validation and quick-look selection.

In periods of less activity, a final screening of the calibration and ranging data will be separately performed, one pass at the time. The screening of the pre- and post-calibration data is done in a least-squares sense including testing procedures. For the screening of the satellite observations, a Keplerian model is used including the short-periodic J2-effect. An iterative method of statistical testing is applied, leading to converging solutions even in case of high noise rates (Vermaat,1978). A number of more or less uniformly distributed and reliable observations is selected for the quick-look data report to be transmitted to the prediction center (before the new orbital elements are received).

3.3 Utilities and Diagnostics

The ranging system's software will also include utilities for data-handling, as preparation of a ground-target prediction file and copy operations. A number of diagnostic programs will be available for trouble shooting in case of system malfunctioning.

4 References

Vermaat, E. Laser Data Preprocessing at the Kootwijk Observatory. Proceedings of the Third International Workshop on Laser Ranging Instrumentation. Lagonissi, May 1978.

Vermaat, E. On site Numerical Integration of the STARLETTE Orbit in a Taylored Force Field. Presented at the Fourth International Workshop on Laser Ranging Instrumentation. Austin, October 1981.

Visser, H. and F.W. Zeeman. Detection Package for the German/Dutch Mobile System. Presented at the Fourth International Workshop on Laser Ranging Instrumentation. Austin, October 1981.

Preliminary Data Handling at Borowiec

by

S. Schillak and E. Wnuk
Astronomical Observatory
Poznan University
Sloneczna Str. 36
Poznan, Poland

ABSTRACT

The results of the satellite laser ranging must be initially prepared for their further utilization. The main purpose of these computations is elimination of the erroneous points, determination of the standard deviation of residuals for the satellite pass, and initial corrections of the results. The presented method was prepared especially for calculating of the results of Borowiec laser system. The following conditions were taken into account in this method:

- small number of points per satellite pass,
- possibility of appearance of the great number of noise points,
- quick and simple calculations on small computer,
- good estimation of the standard deviation.

The method can be used also for estimation of the results from other satellite laser ranging stations.

1 Elimination of the Erroneous Points

For the epochs obtained from observations we calculate expected satellite ranges on the base of actual orbital elements. The satellite range is calculated with secular and long-period perturbations. The difference in range between the observation and ephemeris gives possible the rejection of very big errors, of the order of several kilometers and it is the first step for further computation. The main purpose is to bring to minimum the difference between ephemeris and observation. Ephemeris, in consequence of inaccurate orbital elements and not taken into account all factors perturbing the satellite motion from epoch elements to epoch of observation is burden with the systematical error in epoch ΔT and range ΔR (Fig.1).

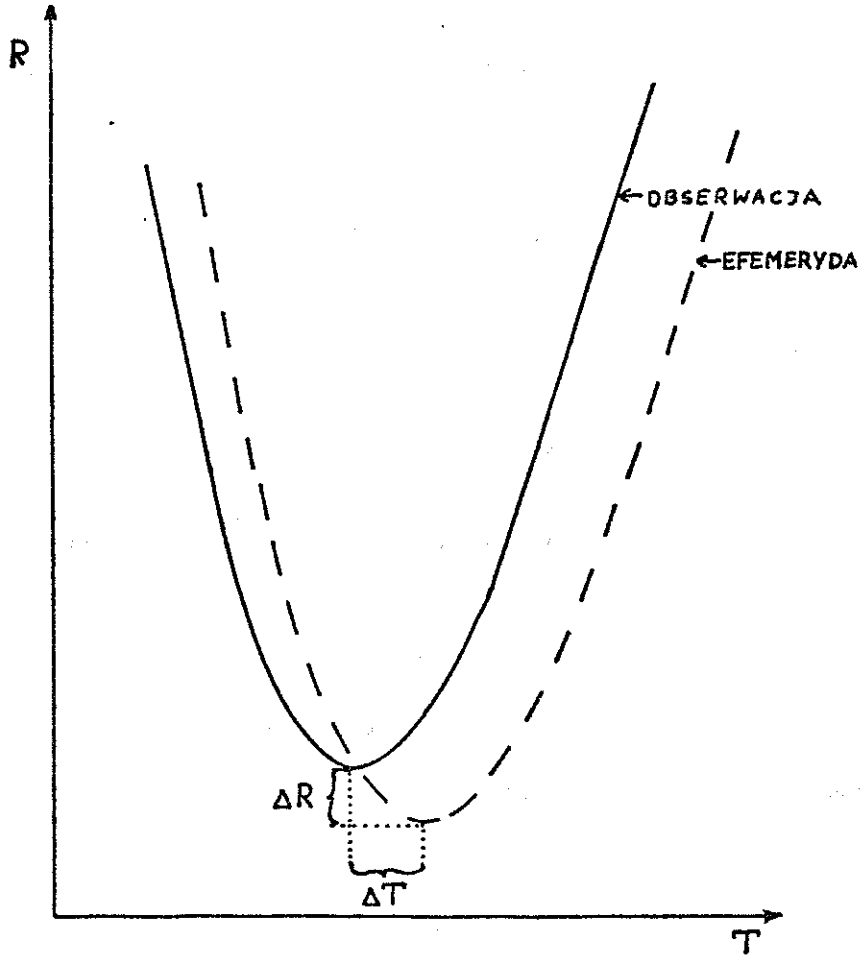


Figure 1: Systematic ephem. error in epoch (ΔT) and range (ΔR).

For the purpose of eliminating of this error, the method supposes step by step change of the epoch so far as both curves will be in agreement. For every step is calculated the sum of squares of the residuals from mean difference between observation and ephemeris. The epoch changes are performed with the 0.1 millisecond step in the direction to diminish the sum square. As final result we obtained the difference in the epochs between ephemeris and observation, the mean value of the difference in the range, it means systematical difference in the satellite range for corrected ephemeris epochs, and residuals from this mean for particular points. The continuous change of

these residuals depends on accuracy of the orbital elements and it comprises in the limit from several meters to several hundred meters (Fig.2).

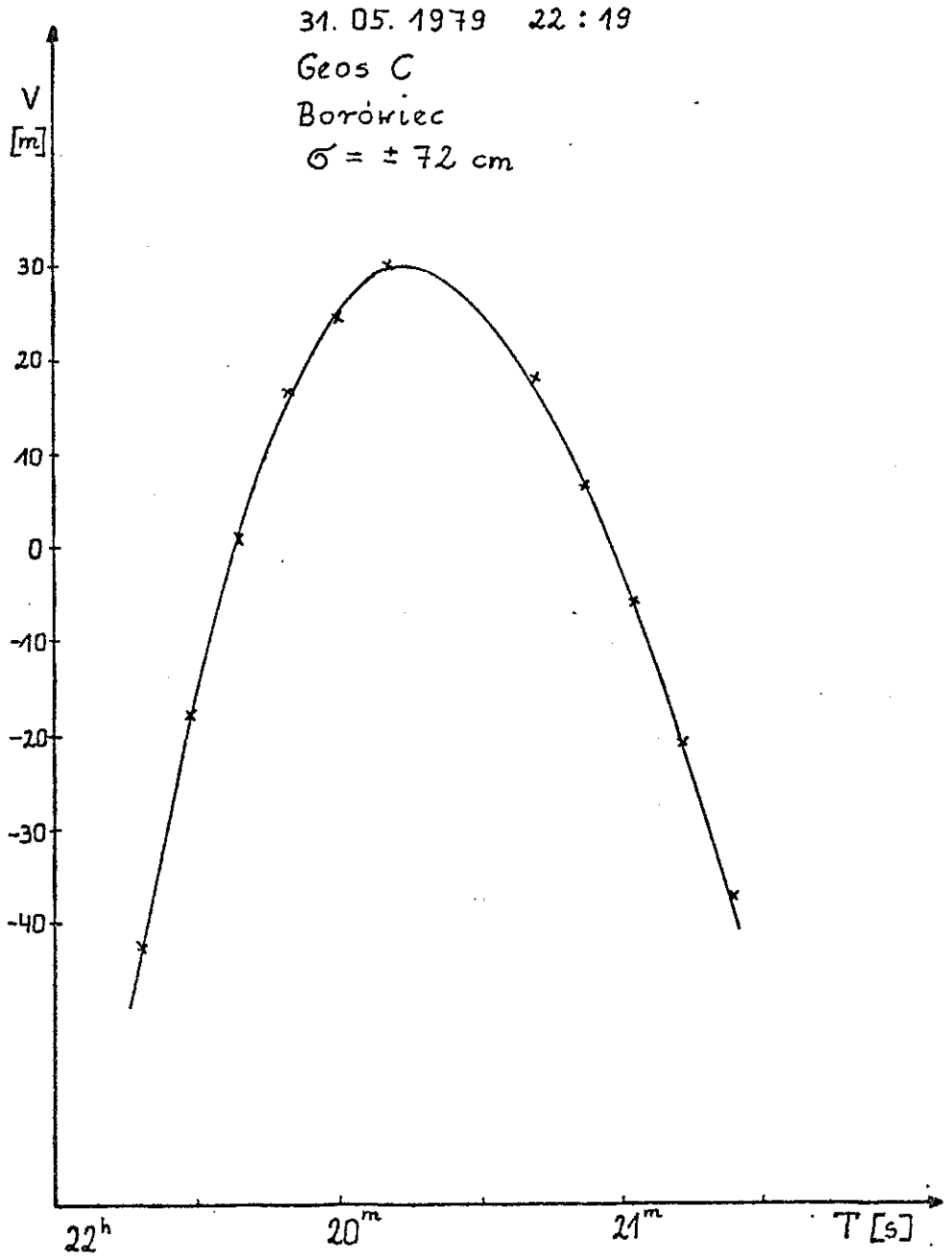


Figure 2: Residuals from the mean dif. observation-ephemeris.

The diagram of the residuals, or visual computer monitor control gives the possibility of easily founding erroneous points, even if errors are small, of the order of several meters, and their number is great (Figs.3,4,5).

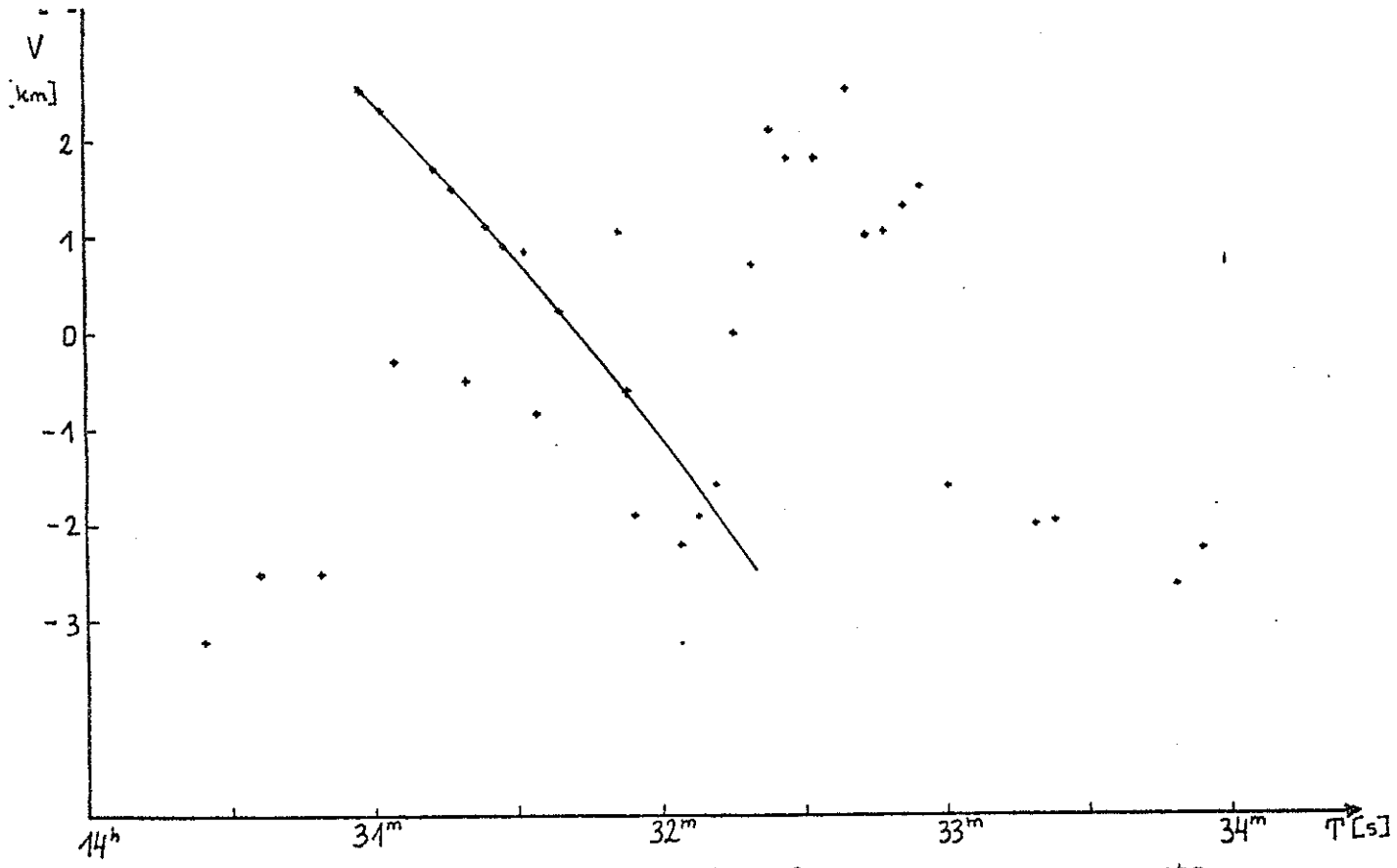


Figure 3: Determination of good points from erroneous measurements

2 Estimation of the Ranging Precision

The important problem for estimation of the observation is possible accurate determination of the standard deviation as result of random dispersion of the observing points for particular pass. For this purpose we used the smoothing of the differences between observation and ephemeris by method of least squares with Tchebychew polynomials.

$$\sigma^2 = \Sigma\, v_p^2 / (n-p-1)$$

The essential element of this method is the determination of the polynomial degree. For this purpose we used statistical method based on Fisher's random variable, which gives the possibility of determination of the polynomial degree for condition when dispersion of the results achieve random character (Schillak & Wnuk, 1977). Fisher's variable we calculate for polynomial degree 'p' by the formula:

$$F = \Sigma\, v_{p-1}^2 - \Sigma\, v_p^2 / \sigma_p^2$$

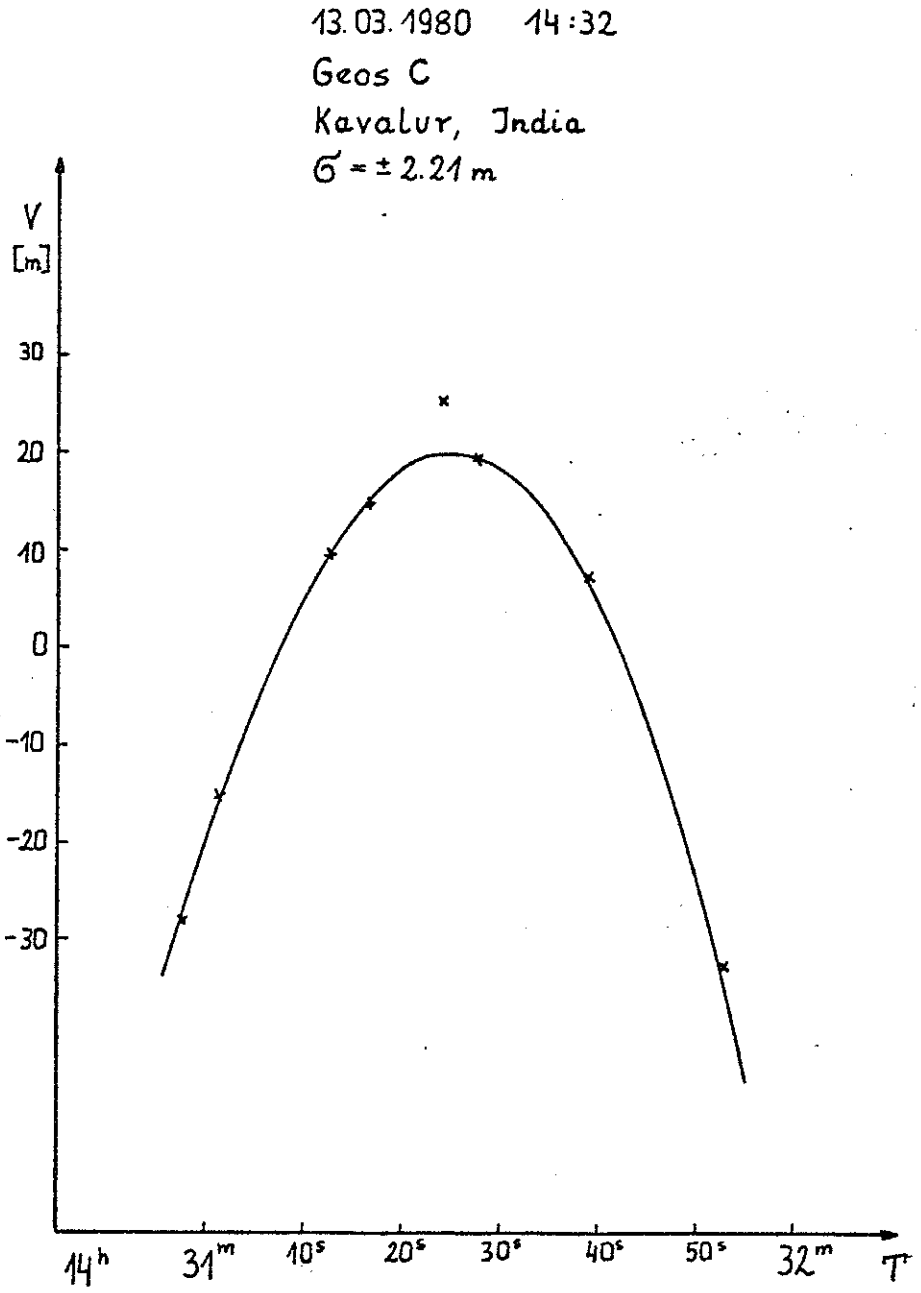


Figure 4: Residuals for good points from Fig. 3, σ = ±221 cm.

where:

- v = residuals from 'p' or 'p-1' polynomial
- n = number of points
- p = polynomial degree

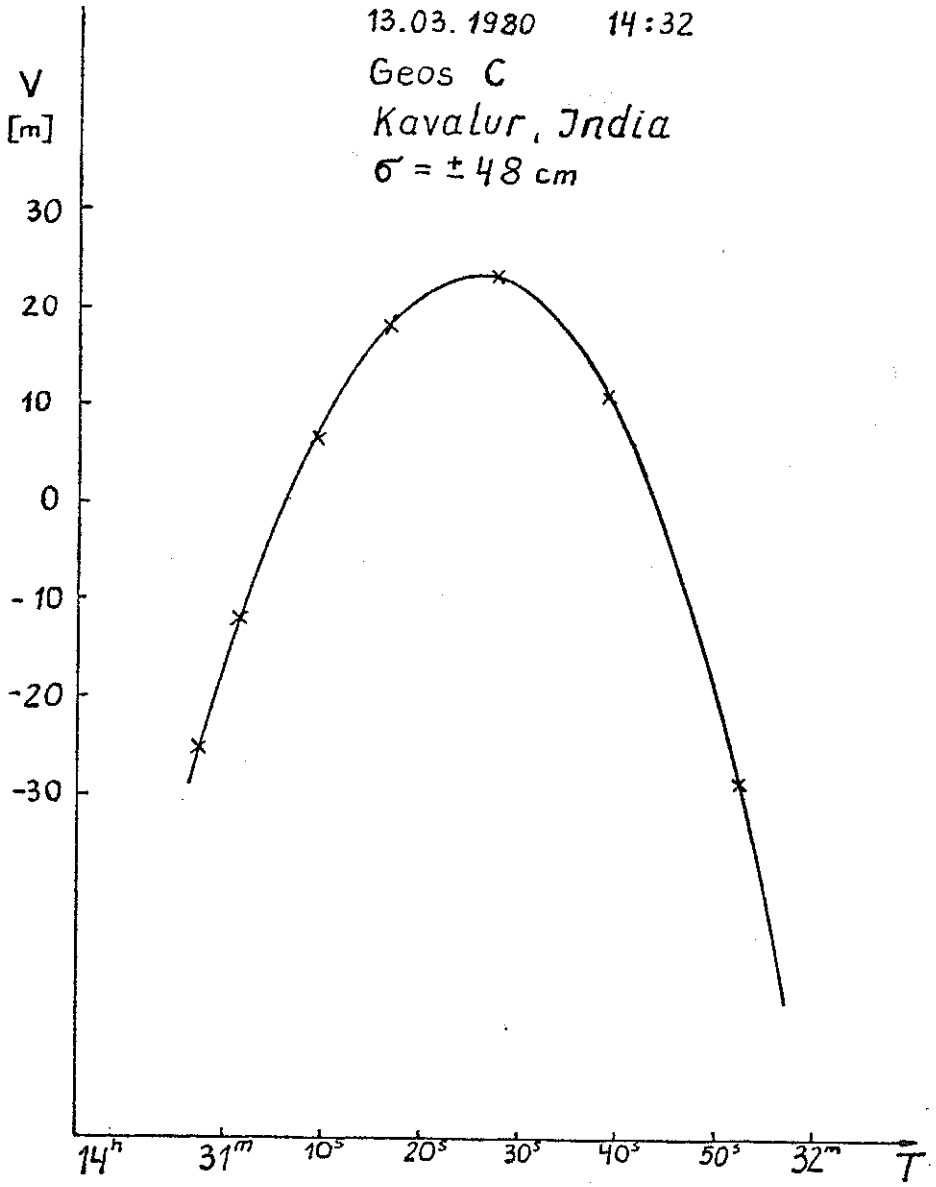


Figure 5: Final residuals for pass, σ = ±48 cm.

The smoothing is stopped, when for successive two polynomial degrees $F < F_o$, where F_o is definite for significance level 0.05 and degree of freedom $\nu_1 = 1$ and $\nu_2 = n-p-1$, it means when the difference between three successive standard deviations is not statistically significant.

3 Correction of the Results

The results of the laser ranging are computed in the two forms. The first one is without any corrections in the SAO 33333 Quick Look Format - epochs for transmitting moment in UTC and two-way range in nanoseconds. This format is used for quick send of the results to cooperation stations and centers of the laser data. In the second form are computed the results with necessary corrections.

1. Calibration Correction - is calculated from the mean of about 20 measurements before and after satellite pass. The ground target distance is corrected for changes of the light velocity in actual atmospheric conditions on the base of the knowledge of ground temperature and pressure by used SAO formula (Gaposchkin, 1973).

2. Atmospheric Correction - is calculated for every point according to ground temperature, pressure and elevation angle of satellite by using simplified SAO formula (Gaposchkin, 1973). This correction results from the changes of light velocity in two-way pass across the atmosphere.

3. Epoch Correction - for the moment of reflecting the laser beam from satellite.

4 Final Data

Computer data consist of all main informations about the satellite pass;

- calibration measurements (mean calibration, standard deviation, calibration correction),
- meteorological data (temperature, humidity, pressure, weather),
- apparatus data (laser voltage, PMT voltage, transmitting and receiving amplitudes, start and stop channel sensitivities),
- epochs and ranges in centimeters with above corrections,
- atmospheric corrections and elevation angle of satellite for every point,
- differences observation-ephemeris,
- residuals from mean difference between observation and ephemeris for correcting ephemeris epochs,
- systematical ephemeris error in range and epoch,
- residuals of the smoothing points for succeeding polynomial degrees, standard deviations and index Fisher's test,
- 33333 Quick Look Format.

5 References

Gaposchkin E. M. - SAO Special Report No. 353, 1973

Schillak, S., E. Wnuk -Veroff. des Zentralinstituts fur Physik der Erde, Nr 52, Teil 3, Potsdam 1977.

6 \ppendix A

```
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 V1(30),V2(30),T(100),RO(100),RRA(100),EL(5,6),RC(100),
     1      RC1(100),RC2(100),TT(100),PP(100),VA(100,20),
     1   SIG(100),D1(100),AUP(100),ARP(100),WS(20)
      INTEGERIV(100,20),IS(20),ID(20),JX(20)
200   FORMAT('1',22X,'ASTRONOMICAL LATITUDE OBSERVATORY')
201   FORMAT(' ',33X,'AT BOROWIEC')
202   FORMAT(' ',30X,'LASER OBSERVATIONS'/////)
203   FORMAT(' ',10X,'SERIAL NUMBER',5X,I5)
204   FORMAT(' ',10X,'DATE',13X,3I5)
205   FORMAT(' ',10X,'CULM. EPOCH',7X,I2,'/',I2)
206   FORMAT(' ',10X,'SATELLITE',9X,I7)
207   FORMAT(' ',10X,'OPERATOR',10X,2A8/////)
208   FORMAT(' ',10X,'OBSERVER',10X,2A8)
209   FORMAT(' ',40X,'C A L I B R A T I O N'/////)
210   FORMAT(' ',34X,'PRE CALIBR.',13X,'POST CALIBR.')
211   FORMAT('0',10X,'CALIBR. EPOCH',9X,I2,'/',I2,20X,I2,'/',I2)
212   FORMAT(' ',10X,'DIAPHRAGM',12X,I2,23X,I2)
213   FORMAT(' ',10X,'TRANS. AMPLITUDE',6X,F3.1,22X,F3.1)
214   FORMAT(' ',10X,'RETURN AMPLITUDE',6X,I3,22X,I3)
215   FORMAT(' ',10X,'NUMBER OF MEASUR.'5X,I3,22X,I3)
216   FORMAT(' ',30X,I2,F8.1,F7.1,10X,2F7.1)
217   FORMAT('0',10X,'MEAN',19X,2F7.1,10X,2F7.1)
218   FORMAT(' ',10X,'STANDARD DEVIATION',4X,F9.2,F6.1, 9X,F9.2,F6.1)
219   FORMAT(' ',10X,'MEAN ERROR',12X,F9.2,15X,F9.2)
220   FORMAT('0',10X,'MEAN CALIBRATION',21X,2F9.1)
221   FORMAT(' ',10X,'CALIBRATION CORRECTION',15X,F9.1)
222   FORMAT('0',10X,'NOTICE')
223   FORMAT('1',34X,'O B S E R V A T I O N'//)
224   FORMAT('0',10X,'METEO DATA',25X,'APPARAT. DATA')
225   FORMAT('0',10X,'TEMPERATURE',F8.1,16X,'LASER VOLTAGE',5X,I6)
226   FORMAT(' ',10X,'HUMIDITY',3X,I8,16X,'PMT VOLTAGE',7X,I6)
227   FORMAT(' ',10X,'PRESSURE',3X,I8,16X,'TRANSM. AMPLITUDE',1X,F6.1)
228   FORMAT(' ',10X,'WEATHER',4X,I8,16X,'RETURN AMPLITUDE',2X,I6)
229   FORMAT(' ',10X,'MOON',7X,A8,16X,'START SENSITIVITY',1X,F6.1)
230   FORMAT(' ',10X,'VISIBILITY',1X,I8,16X,'STOP SENSITIVITY',2X,I6)
231   FORMAT(' ',10X,'TRACKING',3X,I8,16X,'CLOCK CORRECTION',2X,F6.1//)
232   FORMAT('0',11X,'NO',10X,'UTC',12X,'H',3X,'M',6X,'S',15X,'RO'//)
233   FORMAT(' ',10X,I3,F18.10,6X,I2,I4,F12.6,6X,F11.5)
234   FORMAT('1',32X,'COMPARISON WITH EPHEMERIES'//)
235   FORMAT('0',10X,'EPOCH OF ELEMENTS',I7,2I4)
236   FORMAT('0',11X,'NO',4X,'H',7X,'PA',8X,'ROA',10X,'REF',11X,'DRO'//)
237   FORMAT(' ',10X,I3,F7.1,F9.5,3F13.5)
238   FORMAT('1',15X,'DETERMINATION OF THE SYSTEMATIC CORRECTION IN EPOC
     1H AND RANGE')
239   FORMAT('0',11X,'NO',8X,'RC',12X,'ROA-RC',11X,'V'//)
240   FORMAT(' ',10X,I3,3F15.5)
241   FORMAT('0',10X,'EPOCH CORRECTION',4X,F10.4)
242   FORMAT(' ',10X,'RANGE CORRECTION',4X,F10.4)
243   FORMAT(' ',10X,'SQUARE SUM V',8X,F10.4)
```

```
241   FORMAT('1',27X,'DETERMINATION OF THE OBSERVATION ERROR'//)
245   FORMAT('0',11X,'NO',16X,'DEVIATION (V) FOR POLYNOMIAL STEPS')
246   FORMAT('0',20X,I2)
247   FORMAT(' ',18X,I3,8I8)
248   FORMAT(' ',//)
249   FORMAT(' ',10X,I3,I10)
250   FORMAT(' ',10X,I3,8I8)
251   FORMAT('0',10X,'SQUARE SUM',10X,F10.1)
252   FORMAT(' ', 8X,'VARIAN.',I6,8I8)
253   FORMAT(' ', 8X,'ST.DEV.',I6,8I8)
254   FORMAT('1',37X,'SAO 33333 FORMAT'////)
255   FORMAT('+',22X)
256   FORMAT('+',8X)
257   FORMAT('+',23X)
260   FORMAT('1',10X,'ELEMENTS OF ORBIT'///)
261   FORMAT('0',2F20.8,4D16.5)
262   FORMAT('0',10X,'INITIAL DATA'//)
263   FORMAT(' ',10X,I6,2F20.1)
264   FORMAT(' ',8X,'WS',3X,9F8.1)
100   FORMAT(7I10)
      READ(1,100) NRPRZ,NROK,NMC,NDZ,NGZ,NMIN,NRSAT
101   FORMAT(2A8,4X,2A8)
      READ(1,101) AOBS,AOB31,AOPER,AOPER1
102   FORMAT(3I10,F10.1,2I10)
      READ(1,102) K1G,K1M,IPRZ1,AMN1,IAO1,N1
103   FORMAT(F10.1,3I10,2X,A8,2I10)
104   FORMAT(2I10,F10.1,I10,F10.1,I10,F10.1)
105   FORMAT(6F10.1)
      WRITE(3,200)
      WRITE(3,201)
      WRITE(3,202)
      WRITE(3,203) NRPRZ
      WRITE(3,204) NROK,NMC,NDZ
      WRITE(3,205) NGZ,NMIN
      WRITE(3,206) NRSAT
      WRITE(3,208) AOBS,AOB31
      WRITE(3,207) AOPER,AOPER1
      WRITE(3,209)
      WRITE(3,210)
      IPAT=1
      PI=3.1415926535898D0
      RD=180.0D0/PI
      AE=6378.14D0
      S1=0D0
      AKT1=0D0
      AMO1=0D0
      AMK1=0D0
      IF(N1.EQ.0) GO TO 501
      DO 401 K=1,5
      K1=(K-1)*6
401   READ(1,105) (V1(K1+J),J=1,6)
      DO 502 K=1,N1
      V1(K)=V1(K)*5D0
502   AKT1=AKT1+V1(K)
```

```
      AKT1=AKT1/N1
      DO 503 K=1,N1
      V1(K)=AKT1-V1(K)
503   S1=S1+V1(K)*V1(K)
      AM01=DSQRT(S1/(N1-1))
      AN1=N1
      AMK1=AM01/(DSQRT(AN1))
501   READ(1,102) K2G,K2M,IPRZ2,AMN2,IA02,N2
      S2=0D0
      AKT2=0D0
      AM02=0D0
      AMK2=0D0
      IF(N2.EQ.0) GO TO 504
      DO 402 K=1,5
      K1=(K-1)*6
402   READ(1,105) (V2(K1+J),J=1,6)
      DO 505 K=1,N2
      V2(K)=V2(K)*5D0
505   AKT2=AKT2+V2(K)
      AKT2=AKT2/N2
      DO 506 K=1,N2
      V2(K)=AKT2-V2(K)
506   S2=S2+V2(K)*V2(K)
      AM02=DSQRT(S2/(N2-1))
      AN2=N2
      AMK2=AM02/(DSQRT(AN2))
504   CONTINUE
      READ(1,103) TEMP,IWIL,ICIS,IPOG,AKSIE,IWID,IPROW
      READ(1,104) NAPLAS,NAPPMT,AMPNAD,IAMPOD,CZUNAD,ICZUOD,POP
      IF(N1.GT.0.AND.N2.GT.0) AKT=(AKT1+AKT2)/2D0
      IF(N1.GT.0.AND.N2.EQ.0) AKT=AKT1
      IF(N2.GT.0.AND.N1.EQ.0) AKT=AKT2
      WA=0D0
      IF(IPAT.EQ.1) WA=80.29D-6*ICIS/(TEMP+273.2D0)
      AKC=1478.913D0*(1D0+WA+6.917D-4)/0.15D0
      PK=AKT-AKC
      IF(N1.GT.N2) GO TO 507
      NK=N2
      GO TO 508
507   NK=N1
508   CONTINUE
      WRITE(3,211) K1G,K1M,K2G,K2M
      WRITE(3,212) IPRZ1,IPRZ2
      WRITE(3,213) AMN1,AMN2
      WRITE(3,214) IA01,IA02
      WRITE(3,215) N1,N2
      DO 509 K=1,NK
      IF(K.GT.N1) V1(K)=0D0
      IF(K.GT.N2)V2(K)=0D0
      DV1=AKT1-V1(K)
      DV2=AKT2-V2(K)
509   WRITE(3,216) K,DV1,V1(K),DV2,V2(K)
      WRITE(3,217)AKT1,S1,AKT2,S2
      CM01=AM01*15D0
```

```
      CM02=AM02*15D0
      WRITE(3,218) AM01,CM01,AM02,CM02
      WRITE(3,219) AMK1,AMK2
      WRITE(3,220) AKT
      WRITE(3,221) PK
      WRITE(3,222)
      WRITE(3,223)
      WRITE(3,224)
      WRITE(3,225) TEMP,NAPLAS
      WRITE(3,226) IWIL,NAPPMT
      WRITE(3,227) ICIS,AMPNAD
      WRITE(3,228) IPOG,IAMPOD
      WRITE(3,229) AKSIE,CZUNAD
      WRITE(3,230) IWID,ICZUOD
      WRITE(3,231) IPROW, POP
      WRITE(3,232)
108   FORMAT(2F10.1)
107   FORMAT(4I10)
      READ(1,107) IAMP,ITPH,ITPM,ITPS
      TP=(ITPH+(ITPM+ITPS/60D0)/60D0)/24D0
      AMP=IAMP
106   FORMAT(I10)
      READ(1,106) N
      DO 510 K=1,N
      READ(1,108)U,R
      AUP(K)=U
      ARP(K)=R
      U=U/1D4-AMP+ POP/1D3
      U=U/864D2+TP
      RR=R*5D-9
      RRA(K)=RR*149896.229D0
      R=(R*5D0-PK)*1D-9
      U1=U+R/1728D2
      R=R*149896.229D0
      RO(K)=R
      T(K)=U1
      U1=U1*2D0*PI
      CALL RHMS(U1,UH,UM,US)
      IUH=UH
      IUM=UM
510   WRITE(3,233)K,T(K),IUH,IUM,US,R
      WRITE(3,222)
109   FORMAT(4F15.6)
      READ(1,109)XA,YA,ZA,DLU
      DLU=DLU/RD
110   FORMAT(2F20.9)
      READ(1,110) AJDE,DJDE
111   FORMAT(2F15.8,4D12.5)
      DO 511 K=1,4
511   READ(1,111)(EL(K,J),J=1,5)
      READ(1,111)(EL(5,J),J=1,6)
      WRITE(3,260)
      WRITE(3,110) AJDE,DJDE
      DO 560 K=1,4
```

```
56)   WRITE(3,261) (EL(K,J),J=1,5)
      WRITE(3,261) (EL(5,J),J=1,6)
      WRITE(3,262)
      WRITE(3,203) NRPRZ
      WRITE(3,248)
      DO 561 K=1,N
561   WRITE (3,263) K,AUP(K),ARP(K)
      AKE=107.0883D0
      ANDZ=NDZ
      NMC1=NMC
      CALL JDAY(NROK,NMC,ANDZ,AJD)
      AJDEE=IDINT(AJDE+0.5D0+DJDE)
      CALL DATE(AJDEE,IARKE,IAMCE,IADZE)
      WRITE(3,234)
      WRITE(3,235) IARKE,IAMCE,IADZE
      WRITE(3,236)
      S1=0D0
      S2=0D0
      DO 512 K=1,N
      TA=T(K)
      CALL SMC(AJD,TA,DLU,S)
      S=S-2D0*PI*(IDINT(S/(2D0*PI)))
      DT=AJD-AJDE+TA-DJDE
      CALL ELEM(EL(1,1),EL(1,2),EL(1,3),EL(1,4),EL(1,5),
     1          EL(2,1),EL(2,2),EL(2,3),EL(2,4),EL(2,5),
     2          EL(3,1),EL(3,2),EL(3,3),EL(3,4),EL(3,5),
     3          EL(4,1),EL(4,2),EL(4,3),EL(4,4),EL(4,5),
     4          EL(5,1),EL(5,2),EL(5,3),EL(5,4),EL(5,5),EL(5,6),
     5          DT,WM,WD,AI,E,AM)
      CALL KEPL(AM,E,ED)
      V=2D0*DATAN((DSQRT((1D0+E)/(1D0-E)))*DTAN(ED/2D0))
      IF(V.LT.0D0) V=V+2D0*PI
      UM=V+WM
      IF(UM.GT.2D0*PI)UM=UM-2D0*PI
      A=EL(5,2)+EL(5,3)*DT
      A=(AKE/(A*36D1 /RD))**(2D0/3D0)
      DCAI=DCOS(AI)
      DCAI=DCAI*DCAI
      A=A*(1D0+0.1082637D-2*(1D0-3D0*DCAI)/(4D0*A*A*((1D0-
     1  E*E)**1.5D0)))
      CALL WSPOL(A,E,AI,UM,ED,WD,R,XM,YM,ZM)
      S=S-DLU
      XD=(XA*DCOS(S)-YA*DSIN(S))/AE
      YD=(XA*DSIN(S)+YA*DCOS(S))/AE
      ZD=ZA/AE
      ROA=DSQRT((XM-XD)*(XM-XD)+(YM-YD)*(YM-YD)+(ZM-ZD)*(ZM-ZD))
      AH=XD*(XM-XD)+YD*(YM-YD)+ZD*(ZM-ZD)
      BH=DARSIN(AH/ROA)
      RC(K)=ROA*AE
      WA=0D0
      IF(IPAT.EQ.1) WA=0.0414D0*ICIS/(TEMP+273.2D0)
      PA=(2.219D0+WA)/(DSIN(BH)+1D0/(1D3*DTAN(BH)))/1D3
      ROA=RO(K)-PA
      DRO=RO(K)-PA-RC(K)
```

```
      RO(K)=ROA
      S1=S1+RO(K)-RC(K)
      RC2(K)=DRO
      BH=BH*RD
512   WRITE(3,237) K,BH,PA,ROA,RC(K),DRO
      WRITE(3,222)
      TX=0D0
      DO 515 K=1,N
515   S2=S2+(RO(K)-RC(K)-S1/N)*(RO(K)-RC(K)-S1/N)
      NK=0
      DO 525 J=1,5
      J1=J-1
      TX=TX+1D0*(10D0**(-J1))
      NK=0
514   TZ=TX/864D2
      NK=NK+1
      X1=0D0
      X2=0D0
      DO 513 K=1,N
      TA=T(K)+TZ
      CALL SMC(AJD,TA,DLU,S)
      S=S-2D0*PI*(IDINT(S/(2D0*PI)))
      DT=AJD-AJDE+TA-DJDE
      CALL ELEM(EL(1,1),EL(1,2),EL(1,3),EL(1,4),EL(1,5),
     1          EL(2,1),EL(2,2),EL(2,3),EL(2,4),EL(2,5),
     2          EL(3,1),EL(3,2),EL(3,3),EL(3,4),EL(3,5),
     3          EL(4,1),EL(4,2),EL(4,3),EL(4,4),EL(4,5),
     4          EL(5,1),EL(5,2),EL(5,3),EL(5,4),EL(5,5),EL(5,6),
     5          DT,WM,WD,AI,E,AM)
      CALL KEPL(AM,E,ED)
      V=2D0*DATAN((DSQRT((1D0+E)/(1D0-E)))*DTAN(ED/2D0))
      IF(V.LT.0D0) V=V+2D0*PI
      UM=V+WM
      IF(UM.GT.2D0*PI)UM=UM-2D0*PI
      A=EL(5,2)+EL(5,3)*DT
      A=(AKE/(A*36D1 /RD))**(2D0/3D0)
      DCAI=DCOS(AI)
      DCAI=DCAI*DCAI
      A=A*(1D0+0.1082637D-2*(1D0-3D0*DCAI)/(4D0*A*A*((1D0-
     1  E*E)**1.5D0)))
      CALL WSPOL(A,E,AI,UM,ED,WD,R,XM,YM,ZM)
      S=S-DLU
      XD=(XA*DCOS(S)-YA*DSIN(S))/AE
      YD=(XA*DSIN(S)+YA*DCOS(S))/AE
      ZD=ZA/AE
      ROA=DSQRT((XM-XD)*(XM-XD)+(YM-YD)*(YM-YD)+(ZM-ZD)*(ZM-ZD))
      RC(K)=ROA*AE
513   X1=X1+RO(K)-RC(K)
      X1=X1/N
      DO 516 K=1,N
516   X2=X2+(RO(K)-RC(K)-X1)*(RO(K)-RC(K)-X1)
      IF(NK.GT.1) GO TO 517
518   IF(X2.LT.S2) GO TO 520
      ZN=-1D0
```

```
      TX=TX-2D0*(10D0**(-J1))
      DO 519 K=1,N
519   RC1(K)=RC(K)
      GO TO 514
520   ZN=1D0
517   IF(X2.LE.S2) GO TO 521
522   IF(J1.EQ.4) GO TO 525
      TX=TX-1D0*ZN*(10D0**(-J1))
      DO 523 K=1,N
523   RC1(K)=RC(K)
      GO TO 525
521   S1=X1
      S2=X2
      TX=TX+1D0*ZN*(10D0**(-J1))
      DO 524 K=1,N
524   RC1(K)=RC(K)
      GO TO 514
525   CONTINUE
      WRITE(3,238)
      WRITE(3,239)
      DO 526 K=1,N
      ROB=RO(K)-RC1(K)
      ROC=RO(K)-RC1(K)-S1
526   WRITE(3,240) K,RC1(K),ROB,ROC
      WRITE(3,241) TX
      WRITE(3,242) X1
      WRITE(3,243) X2
      DO 530 K=1,N
      TT(K)=T(K)*1440D0
530   PP(K)=1D0
      WRITE(3,244)
      WRITE(3,245)
      L1=N-1
      CALL INTERF(L1,N,TT,RC2,PP,VA,SIG,D1,WS,M)
      MP1=2
      WRITE(3,248)
      IF(M.GT.9) MP1=M-7
      DO 532 J=MP1,M
 532  JX(J)=J-1
      WRITE(3,247) (JX(J),J=MP1,M)
      WRITE(3,248)
      DO 531 I=1,N
      DO 536 J=MP1,M
536   IV(I,J)=VA(I,J)*1D5
531   WRITE(3,250)I,(IV(I,J),J=MP1,M)
      WRITE(3,248)
      DO 534 J=MP1,M
534   IS(J)=SIG(J)*1D10
      WRITE(3,252)(IS(J),J=MP1,M)
      DO 535 J=MP1,M
535   ID(J)=D1(J)*1D5
      WRITE(3,253)(ID(J),J=MP1,M)
      WRITE(3,264) (WS(J),J=MP1,M)
      WRITE(3,254)
```

```
      NSTA=7811
      NMC=NMC1
      CALL FORMSA(NSTA,NRSAT,NROK,NMC,NDZ,IWIL,TEMP,ICIS,AKT,N,T,RRA)
      END
      SUBROUTINE RHMS(A,A1,A2,A3)
      IMPLICIT REAL*8(A-H,O-Z)
      PI=3.1415926535898D0
      RD=180.0D0/PI
      A0=(A*RD)/15.0D0
      IA1=IDINT(A0)
      A1=IA1
      IA2=IDINT((A0-A1)*60.0D0)
      A2=IA2
      A3=((A0-A1)*60.0D0-A2)*60.0D0
      RETURN
      END
      SUBROUTINE JDAY(Y,M,D,JD)
      REAL*8 D,JD,C1,J1,J3,J5
      INTEGER Y,M,C,YA
      IF(M.GT.2) GO TO 10
      M=M+9
      Y=Y-1
      GO TO 20
10    M=M-3
20    C1=Y/100.0
      C=IDINT(C1)
      YA=Y-C*100
      J1=(146097D0*C)/4.0
      J2=IDINT(J1)
      J3=(1461D0*YA)/4.0
      J4=IDINT(J3)
      J5=(153D0*M+2)/5.0
      J6=IDINT(J5)
      JD=J2+J4+J6+D+1721118D0+0.5D0
      RETURN
      END
      SUBROUTINE DATE(JD,Y,M,D)
      REAL*8 Y1,D1,J1,M1,JD,D2
      INTEGER Y,M,D,J
      JD=JD-1721119D0
      Y1=(4*JD-1D0)/146097.0D0
      Y=IDINT(Y1)
      JD=4D0*JD-1D0-146097.0D0*Y
      D1=JD/4.0D0
      D2=D1/1000.0D0
      D=IDINT(D2)
      D2=D1-D*1000.0D0
      M=IDINT(D2)
      D1=D*1000.0D0+M
      J1=(4D0*D1+3)/1461.0D0
      J=IDINT(J1)
      D1=4*D1+3D0-1461*J
      D2=(D1+4D0)/4.0D0
      D=IDINT(D2)
```

```
      M1=(5D0*D-3D0)/153.0D0
      M=IDINT(M1)
      D=5D0*D-3D0-153D0*M
      D1=(D+5D0)/5.0D0
      D=IDINT(D1)
      Y=100D0*Y+J
      IF(M.LT.10) GO TO 10
      M=M-9
      Y=Y+1D0
      GO TO 20
10    M=M+3
20    RETURN
      END
      SUBROUTINE SMC(JD,T,LA,S)
      REAL*8 JD,T,LA,S,T1,S0
      T1=(JD-2415020.0D0)/36525.0D0
      S0=(99.6909833D0+36000.7689D0*T1+0.00038708D0
     1*T1*T1)/57.29577951D0
      S=S0+LA+T*6.30038808D0
      RETURN
      END
      SUBROUTINE ELEM(WM0,DWM,WM1,WM2,WM3,WD0,DWD,WD1,WD2,WD3,I0,DI,
     1I1,I2,I3,E0,DE,E1,E2,E3,M0,DM,DM1,M1,M2,M3,DT,WM,WD,I,E,M)
      REAL*8 PI,RD,U,M4,WM,WD,I,E,M
      REAL*8 WM0,DWM,WM1,WM2,WM3,WD0,DWD,WD1,WD2,WD3,I0,DI,I1,I2,I3
      REAL*8 E0,DE,E1,E2,E3,M0,DM,DM1,M1,M2,M3,DT
      PI=3.141592653589D0
      RD=57.2957795D0
      U=(WM0+DWM*DT)/RD
      WM=WM0+DWM*DT+WM1*DCOS(U)+WM2*DSIN(2D0*U)+WM3*DCOS(3D0*U)
      WD=WD0+DWD*DT+WD1*DCOS(U)+WD2*DSIN(2D0*U)+WD3*DCOS(3D0*U)
      I=I0+DI*DT+I1*DSIN(U)+I2*DCOS(2D0*U)+I3*DSIN(3D0*U)
      E=E0+DE*DT+E1*DSIN(U)+E2*DCOS(2D0*U)+E3*DSIN(3D0*U)
      M4=M0+DM*DT+DM1*DT*DT+M1*DCOS(U)+M2*DSIN(2D0*U)+M3*DCOS(3D0*U)
      M=(M4-IDINT(M4))*360D0
      WM=WM/RD
      WD=WD/RD
      M=M/RD
      I=I/RD
      RETURN
      END
      SUBROUTINE KEPL(M,EE,E)
      REAL*8 S,E,M,E1,EE
      S=0.0D0
      E=M
30    E1=M+EE*DSIN(E)
      S=S+1D0
10    E=E1
      IF(S.GT.100.0D0) GO TO 10
      IF(DABS(E-E1).GT.10D-10) GO TO 20
      GO TO 40
20    E=E1
      GO TO 30
40    RETURN
```

```
      END
      SUBROUTINE WSPOL(A,EE,I,U,E,W,R,X,Y,Z)
      REAL*8 A,EE,I,U,E,W,R,X,Y,Z
      R=A*(1.0D0-EE*DCOS(E))
      X=R*(DCOS(U)*DCOS(W)-DSIN(U)*DSIN(W)*DCOS(I))
      Y=R*(DCOS(U)*DSIN(W)+DSIN(U)*DCOS(W)*DCOS(I))
      Z=R*DSIN(U)*DSIN(I)
      RETURN
      END
      SUBROUTINE INTERF(L1,N,T,Y,P,V,SIG,D1,WS,M)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 T(1),Y(1),P(1),V(100,10),SIG(1),D1(20),X(100),F(100,20),
     1 C(20),SIGMA2(20),D2(20),A(20,20)
      REAL*8 F01(34),SKW(20),FP(20),WS(20)
      DATA F01/161.0,18.51,10.13,7.71,6.61,5.99,5.59,5.32,5.12,4.96,
     *4.84,4.75,4.67,4.60,4.54,4.49,4.45,4.41,4.38,4.35,
     *4.32,4.30,4.28,4.26,4.24,4.22,4.21,4.20,
     *4.18,4.17,4.08,4.00,3.92,3.84/
      WS(1)=-1.0
      WS(2)=-1.0
      IF(L1.GT.15) L1=15
      SY=0D0
      SL=0D0
      SM=0D0
      AX=T(N)-T(1)
      DO 10I=1,N
      X(I)=(T(I)-T(1))/AX
      F(I,1)=1.0D0
      SL=SL+P(I)*Y(I)
      SM=SM+P(I)
10    SY=SY+Y(I)
      YB=SY/N
      SY=0D0
      DO 11 I=1,N
11    SY=SY+(Y(I)-YB)*(Y(I)-YB)
      S2=SY/N
      S=DSQRT(S2)
      C(1)=SL/SM
      DO 12 I=1,N
12    V(I,1)=Y(I)-C(1)
      DO 50 J=2,L1
      J1=J-1
      DO 40 K=1,J1
      SL=0D0
      SM=0D0
      DO 21 I=1,N
      SL=SL+P(I)*(X(I)**J1)*F(I,K)
21    SM=SM+P(I)*F(I,K)*F(I,K)
40    A(J1,K)=SL/SM
      DO 23 I=1,N
      SA=0D0
      DO 22 K=1,J1
22    SA=SA+A(J1,K)*F(I,K)
23    F(I,J)=X(I)**J1-SA
```

```
      SL=0D0
      SM=0D0
      DO 24 I=1,N
      SL=SL+P(I)*Y(I)*F(I,J)
24    SM=SM+P(I)*F(I,J)*F(I,J)
      C(J)=SL/SM
      DO 26 I=1,N
      SA=0D0
      DO 25 K=1,J
25    SA=SA+C(K)*F(I,K)
26    V(I,J)=Y(I)-SA
      SL=0D0
      SM=0D0
      DO 27 I=1,N
      SL=SL+P(I)*V(I,J)*V(I,J)
27    SM=SM+P(I)*F(I,J)*F(I,J)
      SM=SM*(N-J)
      SKW(J)=SL
      SIGMA2(J)=SL/SM
      SIG(J)=DSQRT(SIGMA2(J))
      D2(J)=SL/(N-J)
      D1(J)=DSQRT(D2(J))
      DS=D2(J)/S2
      SIG(J)=D2(J)
      IF(J.GT.2) GO TO 81
      GO TO 50
  81  FP(J)=(SKW(J-1)-SKW(J))/(SKW(J)/(N-J))
      K=N-J
      WS(J)=-1.0
      IF(FP(J).LE.F01(K)) WS(J)=1.0
      IF(WS(J).GT.0.0.AND.WS(J-1).GT.0.0) GO TO 60
50    CONTINUE
60    M=J
      RETURN
      END
      SUBROUTINE FORMSA(NSTA,NRSAT,NROK,NMC,NDZ,IWIL,TEMP,ICIS,
     1                  AKT,N,T,RO)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 T(1),RO(1)
      PI=3.1415926535898D0
301   FORMAT('0',10X,'33333',I5,I1,3I2)
      NROK=NROK-1900
      IDE=NROK/10
      IDK=NROK-10*IDE
      WRITE(3,301) NSTA,IDE,IDK,NMC,NDZ
302   FORMAT(' ',10X,I5,I3,'0',2I2,I3,'0',I5,'0 ',I5,' 00000')
      IDE=NRSAT/100
      IDK=NRSAT-IDE*100
      KT=10*AKT
      ITE=0
      IF(TEMP.LT.0D0) ITE=1
      JTE=10*TEMP+0.1
      WRITE(3,302) IDE,IDK,IWIL,ITE,JTE,ICIS,KT
303   FORMAT(' ',10X,2I2,I1,' ',I5,' 00000 0',I4,' ',I4,'0')
```

```
      DO 310 K=1,N
      U=T(K)
      R=RO(K)
      U=U*2D0*PI
      CALL RHMS(U,UH,UM,US)
      IUH=UH
      IUM=UM
      US=US-R/299792.458D0
      IUS=US/10
      IUSP=(US-IUS*10)*10000+0.1
      R=R/149896.229D0
      R=R*1D9
      IDE=R/10000
      IDK=R-IDE*10000+0.1
310   WRITE(3,303)IUH,IUM,IUS,IUSP,IDE,IDK
      RETURN
      END
```