NASA TECHNICAL NOTE

NASA TN D-3875

C.1

NASA TN D-3875

MIDCOURSE NAVIGATION USING STATISTICAL FILTER THEORY, A MANUAL THEODOLITE, AND SYMBOLIC COMPUTER CONTROL

by Jay V. Christensen and E. David Kipping

Ames Research Center
Moffett Field, Calif.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1967

MIDCOURSE NAVIGATION USING STATISTICAL FILTER THEORY,

A MANUAL THEODOLITE, AND SYMBOLIC COMPUTER CONTROL

By Jay V. Christensen and E. David Kipping

Ames Research Center
Moffett Field, Calif.

MIDCOURSE NAVIGATION USING STATISTICAL FILTER THEORY,

A MANUAL THEODOLITE, AND SYMBOLIC COMPUTER CONTROL

By Jay V. Christensen and E. David Kipping

Ames Research Center

SUMMARY

This report considers the application of a specific hardware-computational system to midcourse guidance and navigation of a manned spacecraft. The system consists of three major subsystems: a digital computer to process the observed data, a computer control and display panel, and a manually operated theodolite. The guidance and navigation computations used statistical filtering and linear prediction. The computer control and display panel used a symbolic abbreviation technique. The theodolite was manually operated. Each of three study phases considered a set of eight closed-loop trajectory runs for the transearth portion of a lunar mission in which actual theodolite observations of a simulated celestial scene were processed, and velocity corrections were simulated. The resulting trajectories were integrated to vacuum perigee to determine system guidance and navigation performance.

The results of this investigation have confirmed theoretical studies regarding the application of statistical filter theory midcourse guidance and navigation, and have shown that the hardware-computational system, as described, will be adequate for on-board midcourse guidance and navigation. No serious anomalies or discontinuities were detected in the statistical filter processing. The results also showed that the velocity correction parameters (thrust magnitude and direction) could be obtained with high accuracy, and, therefore, a precise vernier velocity correction system should be considered if the spacecraft is to make effective use of the system performance. The investigation demonstrated the successful use of a symbolic computer control and display panel concept.

INTRODUCTION

For a manned lunar or interplanetary mission, midcourse guidance and navigation will be necessary in order to meet required terminal conditions. Considerable research has been devoted to the evaluation of statistical filter theory as applied to the computations required for the on-board computer. However, in these studies (e.g., refs. 1-3) the observational errors have been based on an assumed error distribution, and have not been taken directly from actual instrument observations in a manned real-time mission environment. To evaluate the statistical filter theory more realistically, actual hardware is used wherever possible in a simulator at Ames Research Center.

A-1092

Data have been obtained for a particular hardware-computational system which uses statistical filter theory and linear prediction, a representative digital computer, symbolic computer control and display, and manual theodolite observations. This study was specifically oriented to the lunar midcourse guidance and navigation problem. Symbolic abbreviations and electroluminescent displays were used in the "on-board" computer control and display during the data runs. Computer requirements for this on-board control and display concept were investigated.

A moon-to-earth return mission was studied in three investigation phases. Each phase consisted of data runs in which theodolite observations were processed, velocity corrections were simulated, and the resulting trajectories were integrated to perigee to determine system performance. In each phase, the reference trajectory, the observation schedule, the data processing technique, and the optical instrument remained the same. The only variables which changed were the optical observation data and the velocity correction model. Guidance and navigation performance was evaluated on a standard deviation basis for each phase of the study. Performance was measured by the the trajectory state estimation accuracy, the resulting velocity corrections, the deviation of the actual trajectory from the reference trajectory, and the ability of the system to meet the required reentry trajectory state at perigee.

DESCRIPTION OF SIMULATION ON-BOARD SYSTEM

The system simulation used in this study is shown in figure 1. The specific on-board, hardware-computational system investigated is shown within the solid line and includes the following: (a) a guidance and navigation digital computer statistical filter theory and linear prediction, (b) a computer control and display, that uses symbolic abbreviations, and (c) a manually operated theodolite.

A medium-sized, ground-based general purpose digital computer was used to simulate the on-board digital computer. The mathematical formulas and techniques for the on-board guidance and navigation computations were essentially those used by Smith, McLean, Schmidt, and McGee (refs. 1-3), except that a stored polynomial was used for evaluating the sun and moon position, and the numerical integration scheme was different.

The integration technique used was a Stormer-Cowell integration with a starter that builds up the table of differences backwards in time from the point of integration. This technique, reported in reference 4, was chosen for its high accuracy which essentially eliminated one variable (integration error) and provided accurate actual position data for the analysis. For computation of the trajectory, knowledge of the position of the sun and moon was required. A Chebysheff polynomial in time was fitted to each Cartesian component of the sun-moon data over a range of 14 days, a time span chosen to allow for the 7-day round trip, launch delays, and stay time on the moon. The coefficients were computed from Naval Observatory data.

Two sets of nonlinear equations of motion were computed and integrated: the on-board reference trajectory set, which was a nominal trajectory around which the equations of motion were linearized for prediction and guidance; and the estimated trajectory set, which was computed to obtain the best estimate of the position and velocity of the vehicle. These equations included the effect of the sun, the moon, and the second harmonic of the earth's oblateness and are summarized in appendix A of reference 2.

For prediction and guidance, the equations of motion were linearized about the on-board reference trajectory by expanding in a Taylor series and retaining only the linear term, a technique which was used and reported in reference 2.

The computer control and display panel was designed and constructed specifically for midcourse guidance and navigation using symbolic abbreviations for control and display. This panel, shown in figure 2, consisted of the following subassemblies: (1) a real-time clock control and display panel, (2) electroluminescent and incandescent status and warning lights, (3) electroluminescent symbolic display panel, (4) single function, priority interrupt pushbutton panel, (5) thumbwheel input panel, (6) keyboard input panel, and (7) a time interrogate pushbutton. The computer control and display programs were written in machine language. All logic, conversion and formating was oriented to the on-board computer. This allowed realistic definition of the computer requirements for symbolic control and display. All communications between the observer and the computer were symbolic; this provided operating experience with the concept and associated hardware, such as the electroluminescent displays, thumbwheel control, keyboard control, etc. This symbolic control and display concept is described in more detail in appendixes A, B, and C. Control and display panel operation and data format is given in appendix A. The symbolic code structure for the computer control and display panel is described in appendix B. Appendix C contains memory requirements and functional descriptions of the required digital computer control and display programs.

The sighting instrument used was a Hilger and Watts, Microptic no. 2 theodolite (fig. 3). This instrument was manually operated and alined in the simulation coordinate system defined in the next paragraph. The instrument was calibrated with an Ultradex Indexing Table with calibration points every degree. The calibration showed a mean error in right ascension (horizontal) of -0.5 second of arc with a ±0.9 second of arc uncertainty (standard deviation) about the mean. The mean error in declination (horizontal) was 0.6 second of arc with a ±1.5 second of arc uncertainty (standard deviation) about the mean. The uncertainty errors appeared to consist largely of operator and instrument repeatability errors.

COORDINATE SYSTEM

The coordinate system chosen was a nonrotating, right-hand orthogonal Cartesian frame with the origin at the center of the earth. The Z axis

was alined along the north polar axis. The X and Y axes were in the equatorial plane with the X axis alined with the positive direction through the first point of Aries. The theodolite was assumed to be alined in this inertial reference frame with its azimuth axis parallel to the Z axis. Declination was measured as a positive or negative angle above or below the equatorial plane.

DIGITAL COMPUTATIONS

The required digital simulation computations are shown functionally within the dotted line of figure 1, and include processing of control and display information, computation of the on-board guidance and navigation data, computation and processing of the celestial scene data, and the computation and print-out of research data. The computer word length was 24 binary digits. Double precision was used on all guidance and navigation computations. Because of its total length, the computer program was divided into seven links which were stored on magnetic tape and loaded into memory as needed. If it is assumed that the entire program is in memory simultaneously, without any duplications, then storage requirements can be estimated. Such an estimation is given in table I. Forty-eight percent of the total storage required is associated with the simulation, and 52 percent is associated with the on-board computations. Detailed considerations of the simulation computation requirements are given in appendix D.

The time required to process an observation or velocity correction, and update the trajectory estimate, was almost totally dependent on the time required to integrate the equations of motion - normally about 7 minutes.

CELESTIAL SCENE SIMULATION

Figure 4 shows the celestial scene simulation (which includes the planet simulator and star simulators), the computer control and display panel, and the theodolite at the observing station. To simulate the planet in the celestial scene, it was necessary to move a light source with respect to the simulated inertial reference base established for the sighting instrument and the stationary point light star sources. Because it was not desired in this investigation to introduce moving line-of-sight tracking errors, the observations were always taken under static conditions. Observations of a non-moving planet were used to obtain system performance without introducing errors associated with the tracking task. All observations were taken in real time, the planet being repositioned appropriately before each observation so that the observer was required to obtain a new unknown angle. The time of observation was recorded to within 0.01 second. A collimated point light source was used to simulate the planet because a more elaborate collimated planet source was not available at the time of the investigation. The planet source was optically identical to the star sources and was mounted on a rotational table to simulate the angular motion of the planet with

respect to the fixed star field. The rotational table was mounted on a translational table which translated the planet source so that the observer always remained in the field of collimation. To keep the center of the collimated planet bundle at the observing station, the rotational and translational tables were driven at suitable synchronized rates. The rotational table was instrumented with a digital encoder having natural binary output. This provided a resolution of 1.23 seconds of arc over the 12° range of the table. The rotational table encoder was appropriately interfaced with the computer for interrogation under computer control. To avoid parallax errors, collimated point light sources were necessary and the construction is illustrated in the photograph of figure 5. Details of the generation and processing of the observation errors is given in appendix E. For the data of this report the simulation errors were computed to be 2-5 arc seconds (standard deviation). The simulation demonstrated a potential of reducing the simulation errors to 1-2 seconds of arc (standard deviation). Additional considerations·of the celestial scene simulation errors are given in appendix F. The digital simulation techniques used are described and discussed in some detail in reference 5.

INVESTIGATION PROCEDURES

A representative moon-to-earth return mission and trajectory were chosen. The observation schedule, initial conditions, and system errors were selected to be typical of a moon-to-earth return guidance and navigation problem; however, no attempt was made to study or optimize these factors. The reference trajectory was based upon a circumlunar ballistic trajectory of a spacecraft launched from Cape Kennedy on February 12, 1966. The spacecraft was assumed to enter a parking orbit before injection toward the moon. Closest approach to the moon (perilune) was 185.3 km (100.0 nautical miles) from the surface. Vacuum perigee was 60.0 km above sea level at latitude 24.2° N, and longitude 80.98° W, which is approximately over Havana, Cuba. A summary of this reference trajectory is given in table II.

It was assumed that if the spacecraft were injected into a moon-to-earth trajectory at perilune with no injection errors, it would follow the return portion of this circumlunar reference ballistic trajectory. All mission times were from injection of the reference circumlunar trajectory; hence, the spacecraft injection at perilune was taken to occur at 70.68 hours. For each study phase it was necessary to use appropriate initial errors at the moon-to-earth injection point. These errors were obtained by adding initial condition errors to the reference trajectory at the perilune mission time of 70.68 hours. A random number generator computed these initial condition errors on the basis of a normal error distribution with a standard deviation of 1 km, and 0.001 km/sec in each Cartesian coordinate - typical of a boost guidance and navigation system. The specific errors that resulted are given in table III and were used as initial condition errors at the perilune moon-to-earth injection point on all investigation phases. The observation and velocity correction schedule was based on fixed observation and velocity correction time as illustrated in figure 6, and summarized in table IV.

The velocity correction maneuver was not actually mechanized, but was simulated with error values generated from random numbers. Three error sources were considered: (a) alinement of the thrust vector; (b) thrust cutoff; and (c) measurement of the velocity correction. The standard deviations used to describe these errors were chosen to be representative and are summarized in table V.

Three study phases are reported and are summarized in the following table.

	Phase I	Phase II	Phase III
Mission segment	122.0 hr to perigee	122.0 hr to perigee	70.68 hr to perigee
Number of observations	12	12	39
Velocity correction	Last inbound	Last inbound	All inbound (3)
Type of velocity correction maneuver error model	Error model as defined	Perfect	Perfect
Number of data runs	8	8	8

The Phase I study used actual instrument observations for the last 12 observations (fig. 6). The trajectory state for this phase was intialized at the 122.0-hour point by starting the inbound trajectory at the moon-to-earth injection point (70.68 hr) with initial condition errors as previously described. A random number generator was then used to generate instrument sighting errors (as outlined in table VI), velocity correction errors, and velocity correction measurement errors for the inbound trajectory up to 122.0 hours, assuming standard deviations as previously described. The first 27 inbound observations were computed and processed, and both velocity corrections were simulated through a mission time of 122.0 hours according to the observation and velocity correction schedule outlined in table IV. The resulting errors in position and velocity obtained at the 122.0-hour point constituted the initial condition errors used in the subsequent Phase I theodolite data runs from this point, and these errors are given in table VII. An observer took the last 12 observations. The computer then simulated the velocity correction using the errors in alinement, cutoff, and measurement, as outlined in table V. The actual trajectory was then integrated to the time of reference vacuum perigee. Eight separate missions were run in this manner using two different observers. The actual trajectories were compared statistically with the reference trajectory, and with the estimated trajectories through the time of reference vacuum perigee for system performance analysis. To further evaluate performance, the trajectories were integrated to actual perigee, and altitude errors were evaluated at this point.

The Phase II investigation used Phase I observation data, and was identical to Phase I except that in Phase II the final velocity corrections were executed without introducing any velocity correction errors.

6

The Phase III investigation was a computer study covering the entire inbound trajectory. No observer was used directly. The observer sighting data obtained in the Phase I investigation were used. The previously described errors at the moon-to-earth injection point were used for the inbound trajectory. The 96 sets of observer sighting errors obtained in the Phase I missions were then used sequentially in the inbound observations. All inbound observations were processed and all velocity corrections were executed without introducing any velocity correction errors. After the final velocity correction, the resulting trajectories were integrated to the time of reference vacuum perigee. Eight missions from injection to perigee were executed in this manner, using the actual observation error set of Phase I to generate all of the observational errors required. The resulting trajectories were compared statistically with the reference and estimated trajectories to obtain system performance data.

System Performance Reference Data

Statistical filter processing and the resulting guidance and navigation system performance depend on matching the assumed standard deviation of the observation error with the actual instrument observation error. For the Phase I, II, and III data runs, the assumed standard deviation of the observation error was set at 10 seconds of arc. In these runs, the instrument observation error turned out to be closer to 5 seconds of arc, so that a mismatch did occur. System performance also depends on the magnitude of the instrument observation error. To establish theoretical baseline reference data that account for both of the above considerations, and to establish a theoretical baseline representative of optimum system performance, three types of system performance reference data (SPRD) studies were completed.

One SPRD computer run was made assuming a very small normally distributed instrument error of 1 second of arc, and using an assumed standard deviation of the observation error of 1 second of arc. These data are referred to as SPRD-A and represent system performance as limited primarily by the system computational accuracy of the statistical filter theory processing.

To provide System Performance Reference Data with mismatching (as described above) and as a function of the magnitude of the observation error, an additional SPRD computer run was required. This computer run was made assuming an instrument error model of 0 second of arc, and an assumed standard deviation of the observation error of 10 seconds of arc. The resulting data represent the best performance with the system limited by a 10 second of arc assumed standard deviation in the observation error. These data are referred to as SPRD-B. In both SPRD-A and SPRD-B, the errors were small and consistent with statistical errors computed by linear methods so that one run in each case was considered adequate.

To provide System Performance Reference Data when a poor sighting instrument is used, a series of 8 computer runs was made assuming a normally distributed instrument error model with a 1 σ value of 50 seconds of arc, and an assumed standard deviation of the observational error of 10 seconds of

arc. These data are presented as standard deviation data and are referred to as SPRD-C. They are used only in the evaluation of Phase I and Phase II.

The relationship of the assumed standard deviation of the observation error to the instrument observation error, as it was applied to each of the different System Performance Reference Data and to the actual study phases, is summarized in the following table. The model used for the instrument observation error was based on a 1 σ normally distributed value. This model and a random number generator were used to generate individual sighting instrument errors for all of the SPRD computer runs.

	Assumed standard deviation of the observation error (sec of arc)	Instrument observation error (sec of arc, standard deviation)
Phase I, II, and III data runs	10	~5 (obtained from actual observations)
SPRD-A	1	1
SPRD-B	10	0
SPRD-C	10	50

OPERATIONAL SEQUENCE AND PROCEDURES

The sequence and procedures used were centered around the requirements of the Phase I investigation. At the start of Phase I, the celestial scene was simulated appropriately and the trajectory values were those for the 122.0-hour trajectory mission time. The real-time mission clock was set to 122.0 hours and 55 minutes (the approximate time of the next scheduled observation) and was put into the run mode. The planet simulator was driven to a new location which was unknown to the observer. The observer entered the planet identification (earth or moon) and the number of the observation into the computer using the computer control and display panel. He then made an observation by setting the theodolite cross hairs on the planet, and pressed the time interrogate button at the time of the observation. The observer then read the theodolite angles and entered this information into the computer. Next he requested the computer to update the trajectory estimate and the trajectory was updated to the time of the observation. This procedure was used in all 12 observations. The mission time, the number of the observation, and the planet identification were changed as appropriate. After the last observation was processed and the trajectory updated, the velocity correction parameters were computed and a paper tape record of the trajectory state was made so that Phase II could, at a later date, be started properly. The velocity correction was then executed and the trajectory was integrated to the time of reference perigee. All the observation errors from Phase I data runs were recorded and entered on cards for later processing in the Phase III runs.

RESULTS

System performance was evaluated by comparing performance results with the previously described SPRD runs, and with acceptable reentry corridor conditions. All three investigation phases provided consistent system performance data that demonstrated adequate guidance and navigation performance, and demonstrated that the velocity correction parameters could be obtained quite accurately. No serious anomalies or discontinuities were detected in the statistical filter processing or linear prediction technique, with one exception.[1]

PHASES I AND II

The Phase I estimated trajectory errors are compared in figure 7 with the Systems Performance Reference Data, and with the actual position of the vehicle. The 1 σ upper boundary represents the error value when the computed standard deviation is added to the mean of the error. The 1 σ lower boundary represents the error value when the computer standard deviation is subtracted from the mean of the error. System estimation performance approaches the performance capability obtained in SPRD-A and SPRD-B, and is significantly better than the performance obtained in SPRD-C. Table VIII compares the resulting actual velocity correction magnitudes with the System Performance Reference Data runs, and with the data from the other two phases. Phase I velocity correction requirements approach very closely those for SPRD-A and SPRD-B, and are significantly smaller than those for SPRD-C. The general trend was to undercorrect during this last velocity correction, with the result that the fuel expenditure was less than required for SPRD-A. The difference between Phase I and Phase II velocity correction magnitudes is representative of the amount of velocity correction error contained in the Phase I correction maneuver. The magnitude of the first two inbound velocity corrections is not given for Phase I or Phase II because these two corrections were executed prior to the start of those phases. Figure 8 compares the resulting actual trajectory data with the desired reference trajectory, and with the System Performance Reference Data runs. In Phase I, the increase in the upper boundary deviation as the actual trajectory approaches the reference perigee point is to be expected. Most of the error is in the downrange direction, and the upper boundary still represents adequate system performance when compared to allowable standard deviation entry corridor errors (crossrange: 71 km, downrange: 161 km, altitude: 4 km, as obtained from refs. 6 and 7). Table IX compares the actual positional errors at time of reference perigee. The different phase data are compared with each other and with System Performance Reference Data. Phase I results consistently demonstrated

---

[1]When it was attempted to obtain computational base-line data for SPRD-A using a 0 second of arc instrument, and using a 0 second of arc instrument estimation error in the covariance matrix, the guidance and navigation computations generated errors due to a mathematical condition equivalent to dividing by zero.

fixed time of arrival guidance and navigation performance that approaches the performance obtained in SPRD-A and SPRD-B, and is significantly better than SPRD-C. Errors in crossrange and downrange are small compared to the allowable 1 σ corridor errors and are not evaluated further. Inasmuch as the altitude deviation is particularly critical for a safe reentry, the trajectory was integrated out to the actual trajectory vacuum perigee point, and the altitude errors at this point with respect to the reference perigee altitude were computed. These data are given in table X and show that Phase I system performance was significantly better than that required for a safe reentry as defined above.

Phase II results differed from Phase I results only because the velocity correction maneuver errors were not introduced in the Phase II studies. The resulting Phase II performance is, as expected, better than that obtained in Phase I and, together with Phase I data, confirms that the basic hardware-computational system under study will provide system performance approaching very closely the system computation accuracy of the statistical filter theory processing technique. Figure 7 compares the estimated trajectory errors with System Performance Reference Data, and with the actual position of the vehicle. Table VIII compares the resulting actual velocity correction magnitudes with the System Performance Reference Data runs, and with the data from the other two phases. Figure 9 compares the Phase II actual trajectory data with the desired reference trajectory, and with the System Performance Reference Data runs. Tables IX and X compare the Phase II actual positional errors at perigee and show Phase II system performance to be adequate for safe reentry conditions.

PHASE III

Phase III position estimation errors are compared in figure 10 with the System Performance Reference Data, and with the actual position of the vehicle. System estimation performance approaches very closely the performance obtained in SPRD-B. Even though the assumed standard deviation of the observation error that was used did not match the instrument observation error as well as desired, system performance was more than adequate. System estimation performance for Phase III compares very closely with the performance data obtained in the Phase I and Phase II investigations. Table VIII compares the magnitude of the actual velocity correction with that of the System Performance Reference Data runs, and of the data from the other two phases. They were found to be within a few centimeters per second of SPRD-B and are reasonably close to those obtained in SPRD-A. Figure 11 compares the resulting actual trajectory positional errors with the System Performance Reference Data runs. Tables IX and X compare the actual position errors at reference perigee as previously explained for Phase I. These comparisons confirm that Phase III system performance is adequate for a safe reentry, and that this performance approaches very closely computational accuracy limitations discussed earlier.

COMPUTER CONTROL AND DISPLAY

The symbolic computer control and display concept proved very satisfactory operationally. A check list technique was used. However, the abbreviations were learned rapidly and the check list was used mainly to check proper event sequencing. Both the thumbwheels and the keyboard were available for use as the observer saw fit. All observers ended up entering commands with numerical data through the keyboard. Command requests not involving data were entered both ways without any marked preference. The available electroluminescent displays were satisfactory under shaded or dim cab ambient conditions; however, better contrast and intensity will be required for operation without washout in room light or direct sunlight. Symbolic abbreviations with EL display technology depend on translating the computer dc level codes into the appropriate EL segment codes and gating appropriate ac voltage across the segments. A·highly reliable translator is required for actual application of this technique. A large laboratory type relay translator was used for these tests.

CONCLUDING REMARKS

The hardware-computational system investigated provided adequate midcourse guidance and navigation performance. No serious anomalies or discontinuities were detected in the statistical filter processing and linear prediction guidance and navigation when manual observations were used with an inertially fixed theodolite. System performance was sensitive to matching the assumed standard deviation of the observation error to the instrument observation error. System performance demonstrated that the velocity correction parameters could be obtained quite accurately, and, as a result, a precise vernier velocity correction system should be considered if the spacecraft is to make the most effective use of the guidance and navigation capability offered by the statistical filter processing technique. The symbolic computer control and display panel concept provides a communication link between the observer and the computer that is effective and easy to use and interpret. A reliable, low-power, solid-state translator is required before this concept can be considered for use in a spacecraft.

Ames Research Center
    National Aeronautics and Space Administration
        Moffett Field, Calif., Nov. 18, 1966
            125-17-04-01

COMPUTER CONTROL AND DISPLAY OPERATION AND FORMAT

A typical statement format for a computer data input is shown in sketch (a). As can be seen, a full set of input data required 17 sets of BCD
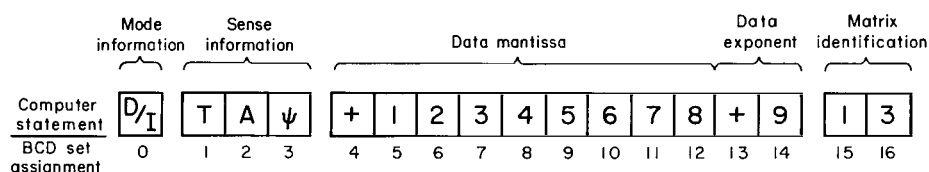
Sketch (a)

data. The mode of computer operation was specified by the first character in the total statement. Although there was a physical possibility of 16 operational modes available in this first character set, only three modes were required in this investigation: Data Input (D/I), Data Output (D/0), and Compute (C). The next three characters in the statement specified sense information and the combination of these three characters represented a code which was interpreted by the computer as a function of the particular operation mode. During data input and data output modes, this information was interpreted by the computer as being representative of the storage location of the data involved. In the Compute mode, the three sense characters were interpreted as the type of computation required. All information entered in these first four characters was symbolic.

Several formats were used for data input or output: time was expressed in days, hours, minutes, and seconds; angles were expressed in degrees, minutes and seconds; whole numbers were expressed as right-justified integers; and other numerical data were represented as floating point numbers with an 8-digit mantissa and a 1-digit exponent. The last two characters were designed for identification of matrix element data.

The interface between the control panel and the computer was mechanized by extensive use of program interrupts and conversion programs which converted the diverse numerical data formats into standard forms for computation. Position data were converted to kilometers, time was converted to seconds, and angles were converted to radians. Data were stored as floating point numbers, except for a few required program flags which were stored as integer numbers.

All data entered into the computer from the computer control and display panel or data displayed as output information from the computer were temporarily stored in three 24-bit memory locations, the contents of which were displayed immediately after each entry. These locations were referred to as D-1, D-2, and D-3, and were formated relative to BCD set configurations as shown in sketch (a) and sketch (b). This technique allowed information

Computer word	Bit configuration 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
D-1	0 1 2 3 15 16
D-2	4 5 6 7 8 9
D-3	10 11 12 13 14 /////

BCD set assignment

Sketch (b)

to be verified prior to any positive computer action that would destroy previous information and provided all input or output devices such as the thumbwheels and keyboard with a common format for communicating with the computer.

In general, the computer control and display programs were written in modular form so as to be easily called by any program of the set. Also, program enable and disable techniques were used frequently to protect the continuous execution of the more critical programs. After each program, the computer was returned to a standard wait loop.

APPENDIX B

COMPUTER CONTROL AND DISPLAY ABBREVIATION CODE STRUCTURE

All computer control and display information was divided into mode information, sense information, and numerical data information as shown in sketch (a). The mode information was entered as one character. The sense information was either entered or displayed as three-character information. The remaining input-output characters were either sign information or contained numerical data. The code structure used and the computer interpretation are given below.

MODE INFORMATION

Mode code	Computer interpretation
D/I	Data Input
D/O	Data Output
C	Compute
EY	Emergency

SENSE INFORMATION

Sense code	Computer interpretation
T M E	Mission time in day, hours, minutes, and seconds from injection
Z E X	Sextant angle (observed)
T A θ	Theodolite angle (observed declination)
T A ψ	Theodolite angle (observed right ascension)
θ - -	Declination (θ) of the velocity correction
ψ - -	Right ascension (ψ) of the velocity correction
V C I	Velocity correction indicated (or estimated) magnitude
C V X	Computed velocity correction in the X axis
C V Y	Computed velocity correction magnitude in the Y axis
C V Z	Computed velocity correction magnitude in the Z axis
V - M	Velcoity gained magnitude
V M X	Velocity gained magnitude in the X axis
V M Y	Velocity gained magnitude in the Y axis
V M Z	Velocity gained magnitude in the Z axis
P X E	Vehicle position in the X axis (estimated)
P Y E	Vehicle position in the Y axis (estimated)
P Z E	Vehicle position in the Z axis (estimated)
P X -	Vehicle position in the X axis (reference)

Sense code	Computer interpretation
P Y -	Vehicle position in the Y axis (reference)
P Z -	Vehicle position in the Z axis (reference)
V X E	Vehicle velocity in the X axis (estimated)
V Y E	Vehicle velocity in the Y axis (estimated)
V Z E	Vehicle velocity in the Z axis (estimated)
V X -	Vehicle velocity in the X axis (reference)
V Y -	Vehicle velocity in the Y axis (reference)
V Z -	Vehicle velocity in the Z axis (reference)
M V E	Moon to vehicle radial distance (estimated)
E V E	Earth to vehicle radial distance (estimated)
T V M	Total velocity gained (measured)
T V E	Total (cummulative) velocity correction uncertainty
M I E	Miss distance at the target as indicated by the estimated trajectory
E I T	Uncertainty in velocity correction
- E M	Earth or moon identification. The observation uses the earth or the moon, depending on an integer 1 or 2 that follows this code.
- I -	The number of the observation is determined by an integer number that follows this code.
M - -	The celestial star that is used in the observation is identified by an integer number that follows this code.

SENSE INFORMATION DURING COMPUTE MODES

Sense code	Computer interpretation
T M -	Process a theodolite observation and update the estimated trajectory to the time of the observation. This compute request requires the following additional information: TME, -I-, -EM, TA, and TA.
Z T A	Process a sextant observation and update the estimated trajectory to the time of observation. This compute request requires the following additional information: TME, -I-, -EM, M--, and ZEX.
I T -	Integrate the estimated trajectory to some future mission time. This request requires the following additional information: TME and -I-.
V C P	Compute the velocity correction parameters. This request requires TME and -I-.
V C -	Execute the velocity correction parameters. This request requires TME and -I-.
X Y Z	Print out all state vectors and matrices.

COMPUTER CONTROL AND DISPLAY PROGRAM STORAGE REQUIREMENTS

AND FUNCTIONAL DESCRIPTIONS

The object of this section is to describe functionally the basic programs that were used and to summarize the storage requirements. The computer program storage requirements are shown below. Functional descriptions follow the table.

Program	Words of storage
Interrogate mission time and celestial scene	10
Sense pushbutton interpretation	30
Thumbwheel data input	12
Keyboard enable	10
Keyboard data	51
Keyboard release	10
Execute	366
Set real-time mission clock	49
Start clock	26
0.01-second computer mission time update . .	15
1-minute computer mission time update	37
Stop clock	15
Display	19
Clear display	14
Start countdown	78
Release countdown	10
Binary time to BCD	13
BCD time to binary	21
BCD to binary floating point	81
Binary to BCD floating point	166
Program interrupts and linkage	125
Constants	262
Total	1420

Interrogate mission time and celestial scene.- This program was entered when the observer pressed the time interrogation button at the time of the observation. The button energized an interrupt, which was assigned the highest priority of any of the interrupts that were used, and initiated this program which first interrogated a double precision fixed point value of computer mission time that was carried in a continuously updated computer memory location known as T. The computer location T had a resolution of 0.01 second, a range of 20 days, and was accurate after each updating cycle to better than 0.002 second. After the computer mission time had been

interrogated and stored for future reference, the computer interrogated the celestial scene and stored the value of the planet position at the time of the observation.

Sense pushbutton interpretation.- The sense pushbuttons 4-9 triggered a common interrupt and initiated the sense pushbutton interpretation program. This program immediately determined the status of all sense pushbuttons and keyboard pushbuttons and transferred the sense pushbutton status in bit information form into D-2 (sketch (b)). Bits 0-3 were then interpreted as BCD data to determine the sense program that should be entered. Five programs could be entered; they were the thumbwheel input, clear display, start countdown, release countdown, and execute.

Thumbwheel data input.- This program was entered from the above sense pushbutton interpretation program. Input data were selected by rotating the thumbwheels to the appropriate positions. When the thumbwheel input button was pressed, all other computer control and display panel programs were disabled for 3/4 of a second; the thumbwheel input pushbutton light was on during this period and all of the thumbwheel data were transferred in three parallel transfers into locations D-1, D-2, and D-3 (sketches (a) and (b)). The computer BCD sets as shown in sketches (a) and (b) corresponded directly to the thumbwheel position numbered from left to right. After the input transfer of the thumbwheel data into locations D-1, D-2, and D-3, the display program was used to display the contents of these locations for verification that the information had been transferred into the computer without any errors.

Keyboard enable.- The keyboard enable pushbutton triggered an interrupt program used with all of the keyboard data pushbuttons and cleared all of the displays by use of the clear display routine. The enter keyboard pushbutton light was turned on to show that control had been set up for the keyboard and remained on until the keyboard release pushbutton was pressed.

Keyboard data.- The keyboard data pushbuttons all triggered a common interrupt. If this keyboard data interrupt had been preceded by the keyboard enable program, then data could be serially entered from the keyboard. Data were entered serially in single BCD character sets and stored in computer memory locations D-1, D-2, D-3 in the same format as previously described. After actuation of each data pushbutton, the specific character of interest was formated, entered into memory, and then, with all previously entered characters, was displayed using the display program. The operator thus saw the character by character construction of the keyboard information as it was entered from the keyboard.

This program displayed only entered characters; all other display windows displayed a cleared (all segments off) condition. After the display had been filled with 17 characters, additional characters were not accepted. After the input of one specific character set with any of the keyboard data pushbuttons, all of the control and display panel was program disabled for 3/4 second to protect against contact bounce and erroneous input errors.

One keyboard data button was used for as many as four different character statements. The sequence of the serial entry determined, under computer program control, which of the different characters assigned to the pushbutton was to be used in forming the statement. The first entry was always interpreted as a computer mode statement such as Data Input (D/I), Data Output (D/O), or Compute (C). The second through fourth entries were interpreted as symbolic sense information as outlined in appendix B. The fifth entry was always interpreted as either a plus or minus sign. The sixth through thirteenth entries were all interpreted as numerical data. The fourteenth entry was interpreted as a plus or minus sign and the fifteenth through seventeenth entries were interpreted as numerical data. This sequential computer interpretation technique allowed the input of 22 different character sets using but one 4-bit BCD code, one common interrupt, and 16 keyboard data pushbuttons.

Keyboard release. - The keyboard release pushbutton triggered an interrupt that initiated an interpretive program for determining which of three possible pushbuttons had been actuated. If the interpretation program determined that the keyboard release pushbutton was still being pressed, then the keyboard release program was entered. This program disabled the keyboard data interrupt and turned off the keyboard enable pushbutton light.

Execute. - The execute pushbutton actuated a program (see appendix A) interpreting the characters associated with bits 0 through 15 (the first four BCD input characters) of D-2 and the proper operation was performed upon the remaining data of D-1, D-2, and D-3. During data input or data output commands, data were stored or loaded from selected addresses in the Fortran common block of memory. In the computer mode, the program was transferred to the appropriate starting locations associated with the Fortran guidance and navigation computations.

For most data input and data output the data entered or displayed were always a signed-fractional 8-decimal digit (coded in BCD) mantissa and a signed 1-decimal digit (coded in BCD) integer exponent. The BCD data to binary, floating point and binary, floating point to BCD data subroutines were used in processing this data for use in computer computations. Special BCD to binary and binary to BCD subroutines handled the data for input and output of time, angle, and integer information.

Set real-time mission clock. - The set clock button triggered an interrupt and initiated a program that assumed that the mission time desired for the clock setting had been entered into the computer, formated in BCD days, hours, and minutes, and had been stored in D-1, D-2, and D-3. This mission time data could be entered either from the thumbwheels or from the keyboard. Upon execution, this program stopped the clock, converted the time that was stored in D-1, D-2, and D-3 from BCD days, hours, and minutes to binary hundreds of seconds by using the time BCD to binary program. The resulting binary time was stored in a computer time location cell T and the real-time clock system was set to the BCD day, hour, and minutes as indicated by the data in D-1, D-2, and D-3. The real-time clock hardware was logically wired to keep track of time in BCD days, hours, minutes, and seconds. In every

setting, the seconds were set to zero. After setting the real-time clock, this program then turned on the clock set pushbutton light.

Start clock.- The mission clock start/stop pushbutton energized an interrupt and initiated an interpretive program for determining whether the start or the stop pushbutton had been actuated. The real-time clock hardware, the 0.01-second computer mission time update program, and the 1-minute computer mission time update program were program enabled.

0.01-second computer mission time update.- The start clock program started 0.01-second pulses from the real-time clock hardware system which energizes an interrupt that initiates this update program. This program incremented the computer mission time data stored in location T by 0.01 of a second. (It was assumed that location T had been preset properly through the set real-time mission clock program.) The total time required for the interrupt, interpretation, and execution of the program was less than 250 microseconds which was consistent with the requirement of providing a computer mission time base that was accurate to 0.01 second.

1-minute computer mission time update.- The start clock program started 1 pulse per minute from the real-time clock hardware that actuated an interrupt and started this program which interrogated the real-time clock hardware. Then, the time BCD to binary program converted the BCD clock data into fixed point, double precision binary data. Computer mission time (location T) was then reset with this updated data to protect against dropout errors.

Stop clock.- The stop/start clock pushbutton disabled the 0.01-second computer mission time update program, and the 1-minute computer mission time update programs and stopped the clock.

Display.- This program assumed data had been stored in memory location D-1, D-2, and D-3 in the standard format of sketch (b) and the total function of this program was to display the data contained in these storage registers.

Clear display.- This program cleared BCD characters in D-1, D-2, and D-3. This information was then displayed by executing the display program described above.

Start countdown.- This program assumed that the future time of an event had been entered and stored in computer memory locations D-1, D-2, and D-3 either through use of the thumbwheel input or the keyboard data program. This information consisted of days, hours, minutes, and seconds stored in BCD format in bits 4 through 23 of D-2 and bits 0 through 11 of D-3. The start countdown pushbutton caused the data to be converted using the BCD to binary floating point program and the results were stored in a location called E.

A one per second pulse interrupt from the real-time clock hardware, enabled the difference between the current value of computer mission time (T) and the entered value of the event time (E) to be computed and stored in a difference register in integer binary seconds. This difference register data was then converted to a 5 digit BCD integer using the binary time to BCD

program. These digits were then formated and displayed using the display program. The resulting data was displayed in positions 5 through 9 of the electroluminescent displays with the sign being displayed in position 4. All other displays were cleared to the off condition. If E was greater than T, the sign was displayed with a negative value. If E was less than T, the sign was displayed with a positive value. A special light was actuated when the two times were identical.

Release countdown.- The release countdown pushbutton energized an interrupt that initiated an interpretive program to determine which sense pushbutton was actuated. When it was determined that the release countdown pushbutton was actuated, the resulting program stopped the countdown program and cleared the displays using the display program.

Binary time to BCD.- This program accepted a double precision binary integer in 0.01 second units, and converted it to 5 BCD characters in 1 second units.

BCD time to binary.- This program converted a series of BCD characters representing days, hours, minutes, and seconds to a double precision binary integer in 0.01 second units.

BCD to binary, floating point.- This program converted a signed fractional mantissa and a signed integer exponent to a floating point binary number.

Binary to BCD, floating point.- This program performed the inverse of the BCD to binary, floating point operation described above. After the BCD characters were stored in D-2 and D-3, the information was displayed through the display program.

APPENDIX D

DIGITAL COMPUTATIONS

It is the purpose of this appendix to discuss the digital computations used in the simulations. All of the simulation computation was accomplished using one digital computer and included on-board digital computer simulation, celestial scene simulation, and research data compilation and recording.

The computer system used was a medium-sized, commercially available, 24 binary digit, general purpose digital computer system with 12,288 words of random accessible core storage. The simulation system also included 32 hardware priority interrupts, a real-time clock system, and the computer control and display panel. In addition to these features, the computer system utilized peripheral equipment that included a card reader, line printer, one magnetic tape unit, paper tape input/output, and a console typewriter.

The program included the mathematical computations associated with the on-board guidance and navigation data processing, for the on-board as well as the simulation (actual trajectory) case, and the programing required in data manipulation and conversion associated with the external computer hardware. The guidance and navigation computations were either arithmetic calculations, simple logic operations, or input/output associated with initial conditions. Programing of data reduction and printout of research results were also required. Hence, to reduce programing time, it was decided to use Fortran II for this section of the programing. Although Fortran is not as efficient in execution time or storage allocations as machine language programing, the coding is fairly efficient for arithmetic computations and greatly eases the programing effort. All arithmetic computations were in double precision floating point. The programing for the control and display panel and the celestial scene interrogation and calibration was accomplished using machine language. The control and display panel programs involved servicing of interrupts, logical operations, and format conversions, such as binary to decimal. The celestial scene interrogation and calibration program required interrogation of the digital encoder on the planetary simulation table, followed by a table search and linear interpolation to find the proper cali- bration value. Proper matching of the data transfer interfaces for the Fortran and machine language programing allowed operation in real time.

PROGRAM EXECUTION TIMES

Once the simulation had been set up and was ready for inputs from the computer control and display panel, approximately 95 percent of the execution time of the program (not counting time for operator tasks) was devoted to the integration of the equations of motion. The exact time spent on this depended on the step size used which, in turn, depended on the portion of the trajec- tory being traversed. For each integration sequence, a beginning step, a

number of intermediate continuation steps, and a terminal step were required. Each of the beginning and terminal steps required about 3 minutes, while the intermediate steps required about 3 seconds apiece. The time to perform other sighting computations was very small so that the total time for processing optical data or updating the trajectory and executing a velocity correction was less than 7 minutes.

FORMULATION AND INTEGRATION OF THE EQUATIONS OF MOTION

Cowell's method was chosen for formulating the equations of motion, because it was used in the original theoretical studies reported in references 1, 2, and 3, and because it had been thoroughly investigated and checked. It was not the objective of this study to develop or to evaluate advanced formulation techniques. Cowell's method is a simple straightforward technique and for theoretical work it is probably the best one to use, since it is easy to program and check and eliminates many of the potential trouble and error sources of other techniques. However, this method requires large amounts of storage, is slow, and always requires a high accuracy integration scheme.

For an on-board spacecraft computer there may be better ways of both formulating and integrating the equations of motion. For example, the reference trajectory and the transition matrix could be periodically reset using stored values and integrating between reset points using a very simple integration scheme. A preliminary investigation of this technique showed that using a simple Runge-Kutta integration scheme and resetting about 10 times would produce sufficient accuracy on an earth-moon journey for a typical trajectory. Another type of improvement would be to use an Encke integration method and integrate only the deviations from the two-body solution, thus lessening the integration accuracy requirements. This method would be faster, although it probably would not significantly reduce storage. Finally, if a computational technique which eliminated the necessity for integrating the transition matrix were developed or a guidance technique which eliminated the need for the transition matrix altogether were developed, complexity, storage, and execution time could be reduced considerably.

GENERATION AND PROCESSING OF OBSERVATIONAL ERRORS

Sketch (c) is a four-step functional flow diagram of how the observational errors were generated and incorporated in the on-board statistical filter process.
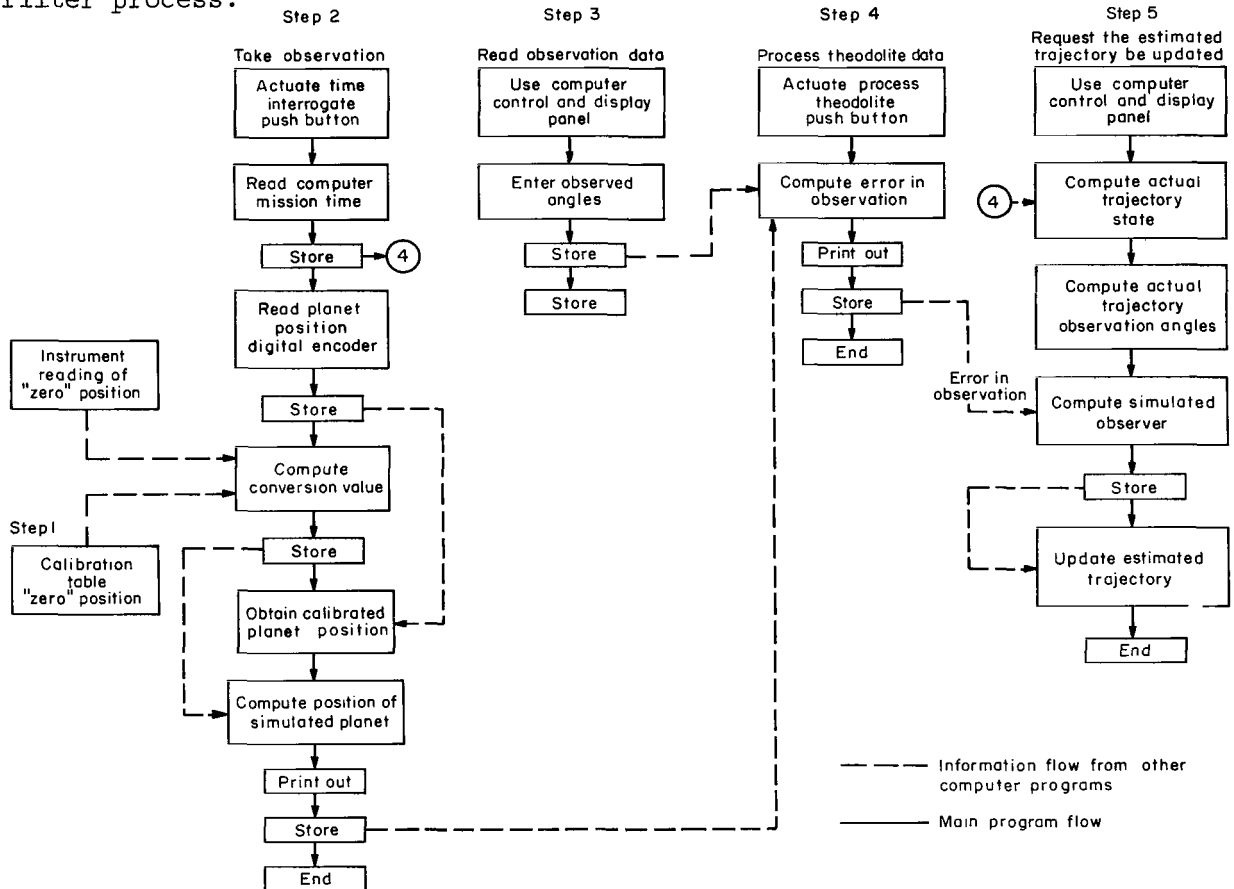
Sketch (c)

In Step 1, the planet is positioned at a defined zero position. A number of measurements are taken of the angle between this planet zero position and the star position. The arithmetic mean of this data is entered into the computer and is used to bias the calibration curve to compensate for any shifts from day to day.

In Step 2, the observer took the observation of the simulated celestial scene and pressed the time interrogate pushbutton to start a computer program that interrogated and stored computer mission time of the observation for use in Step 4. The digital encoder was then read for the planet position and this information was stored for later use. The program then computed the conversion value which was required for the translation of the table

calibration information into the instrument reference frame. This was done by comparing the table calibration zero values with the values obtained from the zero position readings that were taken daily with the sighting instrument. The program then used the previously stored encoder information to obtain the position of the planet in the calibration frame. The calibration table was a 60-point table having values for right ascension and declination for every 1/4 of a degree of planet motion. Linear interpolation was used for information between calibration points. The previously computed conversion values were then used to convert the simulated planet position from the calibration frame into the instrument reference frame. This latter position was referred to as the position of the simulated planet and was printed out for research purposes and stored for use in Step 4.

In Step 3, the observer read the instrument angles and entered them into the computer through the computer control and display panel. They were then stored for use in Step 4.

In Step 4, the observer pressed a pushbutton to process the position of the simulated planet. The position was then compared with the observed angles to determine the error in the observation and this error was stored for use in Step 5.

In Step 5, the observer used the computer control and display panel to request that the estimated trajectory be updated. As a result, the actual trajectory state was updated to the computer mission time obtained in Step 2. The values of what the actual observation angles should be at this point on the trajectory were then computed. These computed actual observation angles were modified by incorporation of the errors in the observation obtained in Step 4. The resulting observation information was called the simulated observation and was used in updating the estimated trajectory using statistical filter theory processing.

APPENDIX F

CELESTIAL SCENE SIMULATION ERRORS

Rather than trying to determine all individual error contributions associated with the celestial scene and then establishing a system calibration table to correct for these errors, it was decided to calibrate a complete system with as precise an instrument as was available. The theodolite used in this study was calibrated in $1°$ increments, using an Ultradex that was accurate to 0.25 second of arc. The results of this calibration are given in column A of sketch (d). The error is small and appeared to be a function of readability and repeatability so the theodolite was used as an absolute reference without introducing any instrument calibration corrections. As a result, it was assumed that if any bias errors showed in the celestial scene calibration they would be celestial scene simulation errors.

Sketch (d) also compares the celestial scene simulation errors with the errors in the calibration instrument and with errors obtained under different calibration situations. Column A gives the calibration instrument errors. The errors obtained during the data runs is given statistically in column B.

	(A) Theodolite	(B) Data runs normal range (12°)	(C) Data runs restricted range (1/4°)	(D) Computed simulation errors for normal range	(E) Computed simulation errors for restricted range
Right ascension (mean)	-.5	-2.2	-.8	-1.7	-.3
Right ascension (standard deviation)	.9	5.5	1.5	5.5	1.2
Declination (mean)	.6	1.4	.4	.8	.2
Declination (standard deviation)	1.5	3.5	1.4	2.1	0

Sketch (d)

Simulation celestial scene errors were computed by assuming that the instrument and simulation scene errors combined in a root-mean-square relationship. This relationship was used to compute the simulation errors for the data run normal range case (using columns A and B data). The computed simulation errors are given in column D.

The above simulation errors should be acceptable for the evaluation of instrument systems in the ±10-30 second accuracy class. Even in the case of evaluating the application of ±2 second instrument, such as was done when the calibration theodolite was used as the sighting instrument, the validity of the resulting guidance and navigation performance can be justified on the basis of the following considerations: (1) the celestial scene simulation error was not large enough to introduce significantly adverse effects in guidance and navigation performance; (2) even with simulation errors present in the processed observation data, the error was small enough so that

instrument-computational system discontinuities and anomalies that might result in adverse systems performance could have been detected if present; (3) if a hardware-computational system processes errors containing simulation errors and still provides statisfactory system performance, it can be expected that the basic guidance and navigation system under evaluation should perform better than the simulation will show; and, (4) having determined system performance, using a very precise instrument, base-line performance data are available for comparison with follow-on investigations involving less precise hand-held instruments. A simulation capability with better than 1-second-of-arc accuracy would be the obvious answer to this simulation problem rather than logical and statistical arguments for the validity of the research results.

The possibility of restricting the planet table operation to within $1/4^{\circ}$ of the daily zero point was investigated. The resulting system errors are shown in column C of sketch (d) and the computed simulation errors are given in column E. In order to take daily zeros (one every $1/4^{\circ}$) over the desired 12° range of the simulation would require a daily zeroing effort equivalent to one recalibration run each day. This technique was not used for practical operational reasons. However, the data do point out that a mechanical system, such as described, if restricted in dynamic range, does have the potential of limiting simulation errors to the 1-2 second of arc range.

REFERENCES

1. McLean, John D.; Schmidt, Stanley F.; and McGee, Leonard A.: Optimal Filtering and Linear Prediction Applied to a Midcourse Navigation System for the Circumlunar Mission. NASA TN D-1208, 1962.

2. Smith, Gerald L.; Schmidt, Stanley F.; McGee, Leonard A.: Application of Statistical Filter Theory to the Optimal Estimation of Position and Velocity On Board a Circumlunar Vehicle. NASA TR R-135, 1962.

3. Smith, Gerald L.: Multivariable Linear Filter Theory Applied to Space Vehicle Guidance. Paper presented at SIAM Symposium on Multivariable System Theory. November 1962.

4. Mersman, William A.: Self-Starting Multistop Methods for the Numerical Integration of Ordinary Differential Equations. NASA TN D-2936, 1965.

5. Christensen, Jay V.: Digital Simulation Techniques for Lunar Midcourse Guidance and Navigation Computer Systems Research. 1966 IEEE Aerospace Systems Conference Record, Supplement to IEEE Transactions on Aerospace and Electronics Systems, Vol. AES-2, No. 4, July 1966.

6. Duncan, Robert C.: Apollo Navigation, Guidance and Control. Apollo Lunar Landing Mission Symposium, Manned Spacecraft Center, Houston, Texas, Vol. I, June 25-27, 1966.

7. Frank, M. P.: Transearth Injection Through Reentry. Apollo Lunar Landing Mission Symposium, Manned Spacecraft Center, Houston, Texas, Vol. II, June 25-27, 1966.

TABLE I.- COMPUTER WORD STORAGE REQUIREMENTS

Program	Total storage	On-board storage
Computer control and display	1,420	1,420
Celestial scene calibration	2,560	0
Guidance and navigation	14,999	6,520
Fortran library and run time system	4,261	3,219
Data	2,535	2,388
Total	25,775	13,547

TABLE II.- CHARACTERISTICS OF THE REFERENCE TRAJECTORY

Injection conditions	Outbound leg in the earth centered coordinate system		Inbound leg in the moon centered coordinate system at perilune (70.68 hr)	
X	-560.7075	km	-154.890	km
Y	5931.5798	km	-758.885	km
Z	2595.8516	km	-761.638	km
Range	6499.0	km	1926.0	km
$\dot{X}$	-10.620809	km/sec	-2.470377	km/sec
$\dot{Y}$	.27002576	km/sec	.062852	km/sec
$\dot{Z}$	-2.9111238	km/sec	.348196	km/sec
Velocity	11.015858	km/sec	2.4956	km/sec
Transit time	70.68	hr	74.195	hr

TABLE III.- INITIAL CONDITION ERRORS AT THE TRANSEARTH INJECTION POINT

Parameter	Initial condition errors	
X	-0.588	km
Y	-.304	km
Z	1.254	km
$\dot{X}$	.001359	km/sec
$\dot{Y}$	.000949	km/sec
$\dot{Z}$	.000449	km/sec

TABLE IV.- OBSERVATION AND VELOCITY CORRECTION SCHEDULE

Mission time in hours	Mission function	Observed body
70.68	Transearth injection time	
71.0	Observation	Moon
71.5	Observation	Moon
72.0	Observation	Moon
72.5	Observation	Moon
73.0	Observation	Moon
73.5	Observation	Moon
74.0	Observation	Moon
74.5	Observation	Moon
75.0	Observation	Moon
75.5	Observation	Moon
76.0	Observation	Earth
76.5	Observation	Earth
77.0	Observation	Earth
77.5	Observation	Earth
78.0	Observation	Earth
78.5	Observation	Earth
79.0	Observation	Earth
79.5	Observation	Earth
80.0	Observation	Earth
80.5	Observation	Earth
81.0	1st return velocity correction	
109.0	Observation	Moon
110.0	Observation	Moon
111.0	Observation	Earth
112.0	Observation	Earth
113.0	Observation	Earth
114.0	Observation	Earth
115.0	Observation	Earth
116.0	2nd return velocity correction	
123.0	Observation	Moon
124.0	Observation	Moon
125.0	Observation	Moon
126.0	Observation	Earth
127.0	Observation	Earth
128.0	Observation	Earth
129.0	Observation	Earth
130.0	Observation	Earth
131.0	Observation	Earth
132.0	Observation	Moon
133.0	Observation	Moon
134.0	Observation	Moon
135.0	Final return velocity correction	
144.87527	Time of reference perigee (aim point).	

TABLE V.- VELOCITY CORRECTION ERRORS

Parameter	Standard deviation
Engine alinement in right ascension	0.6°
Engine alinement in declination	$.6^\circ$
Engine cutoff	.00010 km/sec
Thrust measurement in the X axis	.00006 km/sec
Thrust measurement in the Y axis	.00006 km/sec
Thrust measurement in the Z axis	.00006 km/sec

TABLE VI.- INITIAL OBSERVATIONAL ERRORS USED IN PHASE I AND PHASE II

Parameter description	Parameter	Value
Assumed standard deviation value of the observational errors	Right ascension	10.0 seconds of arc (1 σ)
	Declination	10.0 seconds of arc (1 σ)
Instrument sighting error	Right ascension uncertainty	8.0 seconds of arc (1 σ)
	Right ascension bias	2.0 seconds of arc
	Declination uncertainty	8.0 seconds of arc (1 σ)
	Declination bias	2.0 seconds of arc

TABLE VII.- INITIAL CONDITION ERRORS AT 122.0 HOURS AS USED IN PHASE I AND PHASE II

Parameter	Error	
X	72.594	km
Y	1.855	km
Z	-19.889	km
$\dot{X}$	-.000648	km/sec
$\dot{Y}$	.000328	km/sec
$\dot{Z}$	.000276	km/sec

TABLE VIII.- ACTUAL VELOCITY CORRECTION MAGNITUDES, M/SEC

Parameter		Phase I data	Phase II data	Phase I and Phase II			Phase III data	Phase III SPRD-A	Phase III SPRD-B
				SPRD-A	SPRD-B	SPRD-C			
First return velocity correction	Mean	---	---	---	---	---	2.72	2.51	2.62
	Standard deviation	---	---	---	---	---	.40	---	---
Second return velocity correction	Mean	---	---	---	---	---	.72	.07	.46
	Standard deviation	---	---	---	---	---	.10	---	---
Third return velocity correction	Mean	0.37	0.34	0.44	0.33	0.94	.31	.05	.40
	Standard deviation	.17	.14	---	---	.38	.25	---	---

TABLE IX.- POSITIONAL ERRORS (ACTUAL - REFERENCE) AT TIME OF REFERENCE, PERIGEE, KM

Parameter		Phase I data	Phase II data	Phase I and Phase II reference data			Phase III data	Phase III SPRD-A	Reference data SPRD-B	Allowable limits
				SPRD-A	SPRD-B	SPRD-C				
Altitude	Mean	-0.46	-0.42	0.75	-0.50	0.27	0.40	-0.15	0.21	0
	Standard deviation	.85	.60	---	---	7.86	.69	---	---	4.0
Cross-range	Mean	.28	.28	.04	.17	.27	.09	0	.01	0
	Standard deviation	.19	.13	---	---	.44	.06	---	---	71.0
Down-range	Mean	-42.09	-45.11	-12.47	-22.36	-34.0	-47.99	1.06	-33.17	0
	Standard deviation	89.61	38.19	---	---	102.0	41.20	---	---	161.0

TABLE X.- ALTITUDE ERROR (ACTUAL - REFERENCE) AT VACUUM PERIGEE, KM

Mean	Phase I	Phase II	Phases I and II reference data			Phase III	Phase III SPRD-A	Reference data SPRD-B	Allowable error
			SPRD-A	SPRD-B	SPRD-C				
	-0.245	-0.296	0.063	-0.460	0.725	0.636	-0.24	0.29	0
Standard deviation	.986	.721	---	---	7.892	.943	---	---	4.0
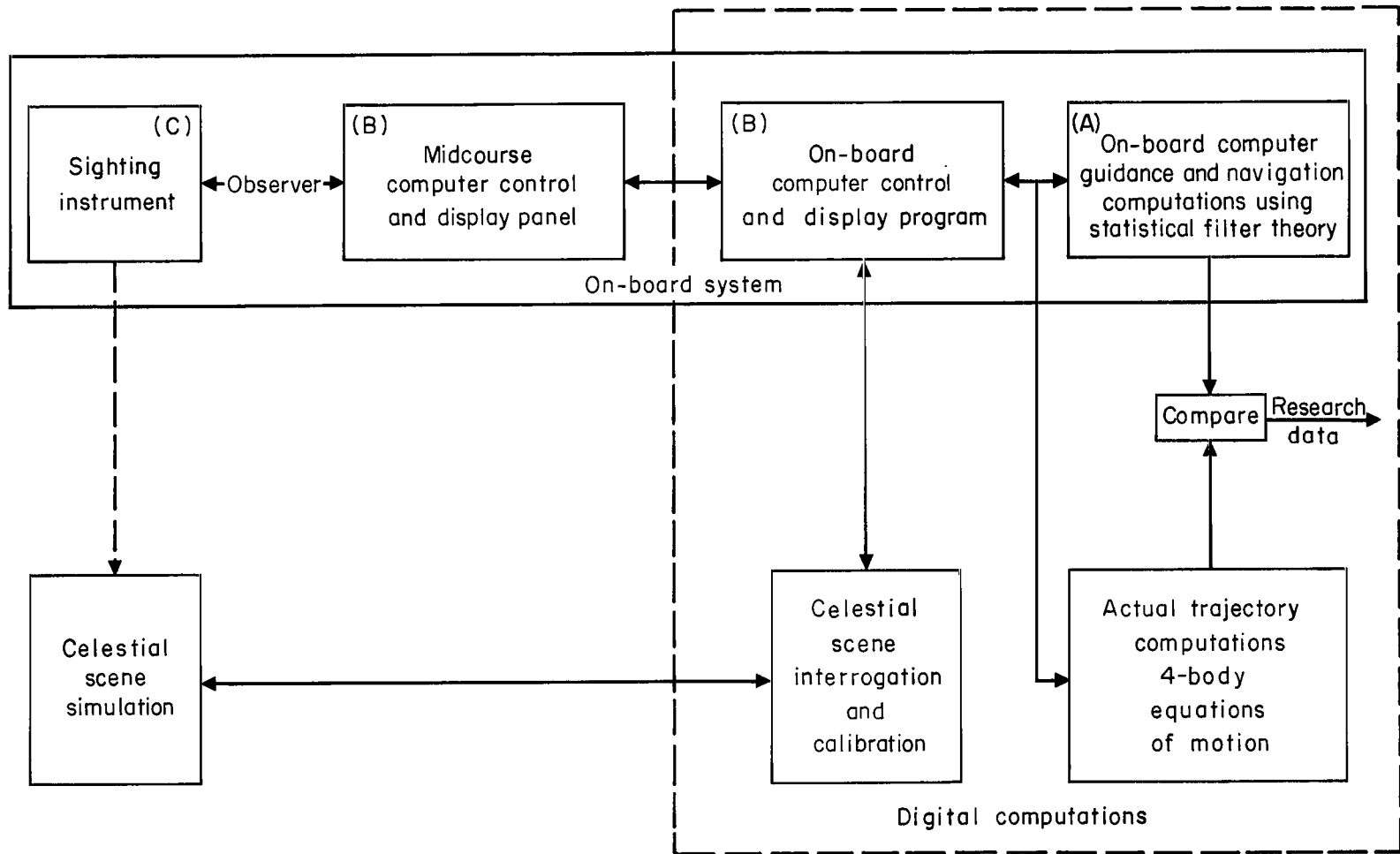
Figure 1.- Simulation configuration.
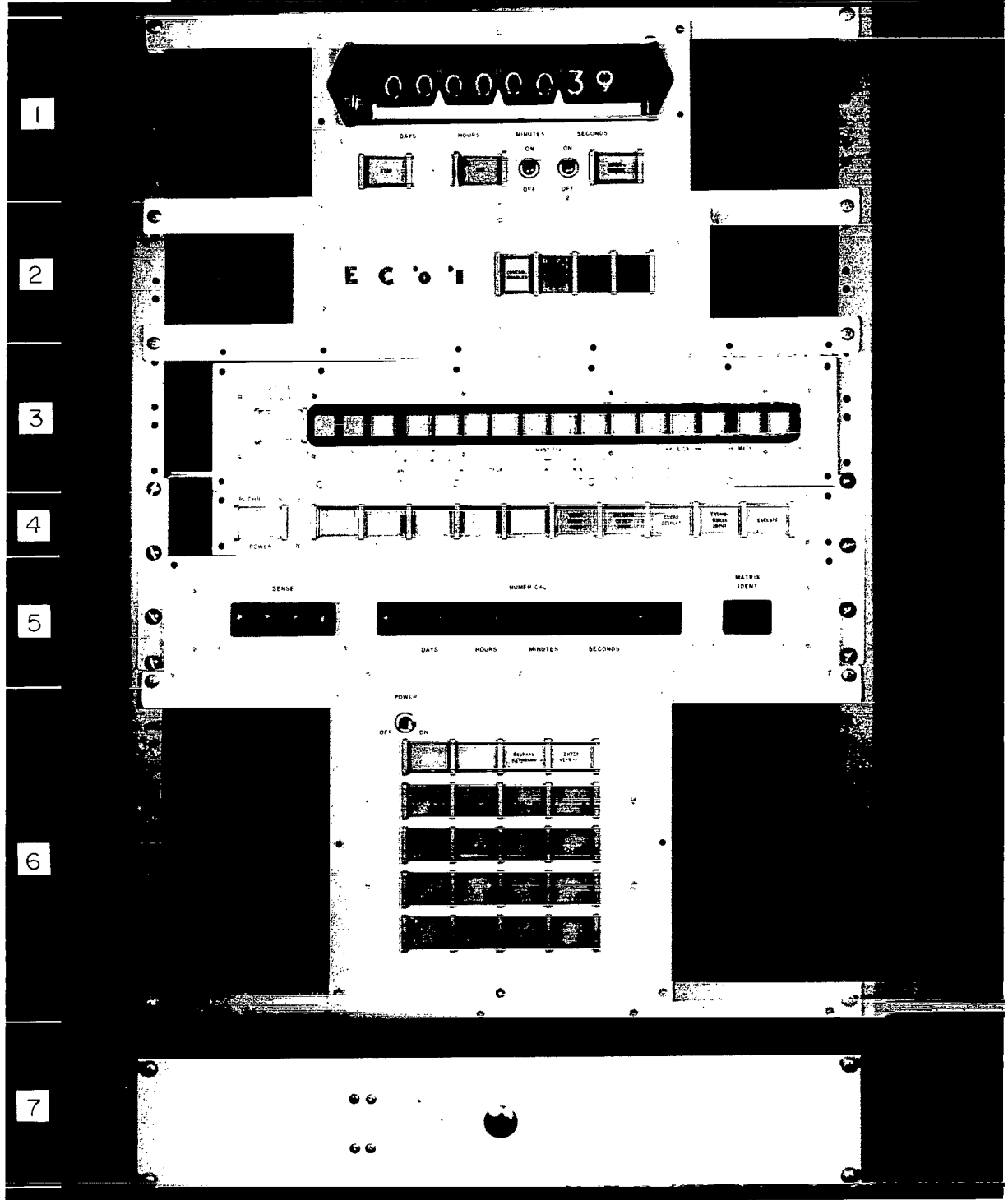
Figure 2.- Computer control and display panel.

A-33837.1
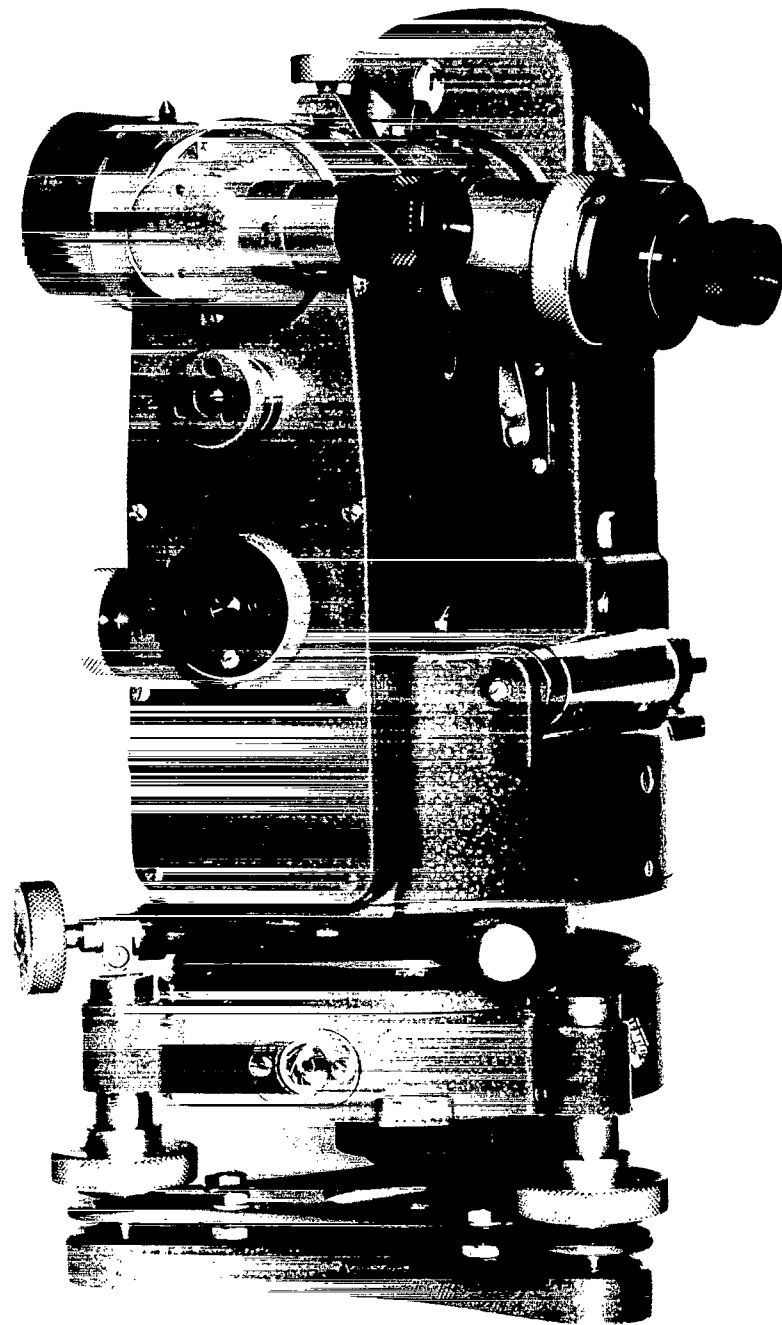
Figure 3.- Theodolite.

A-35714
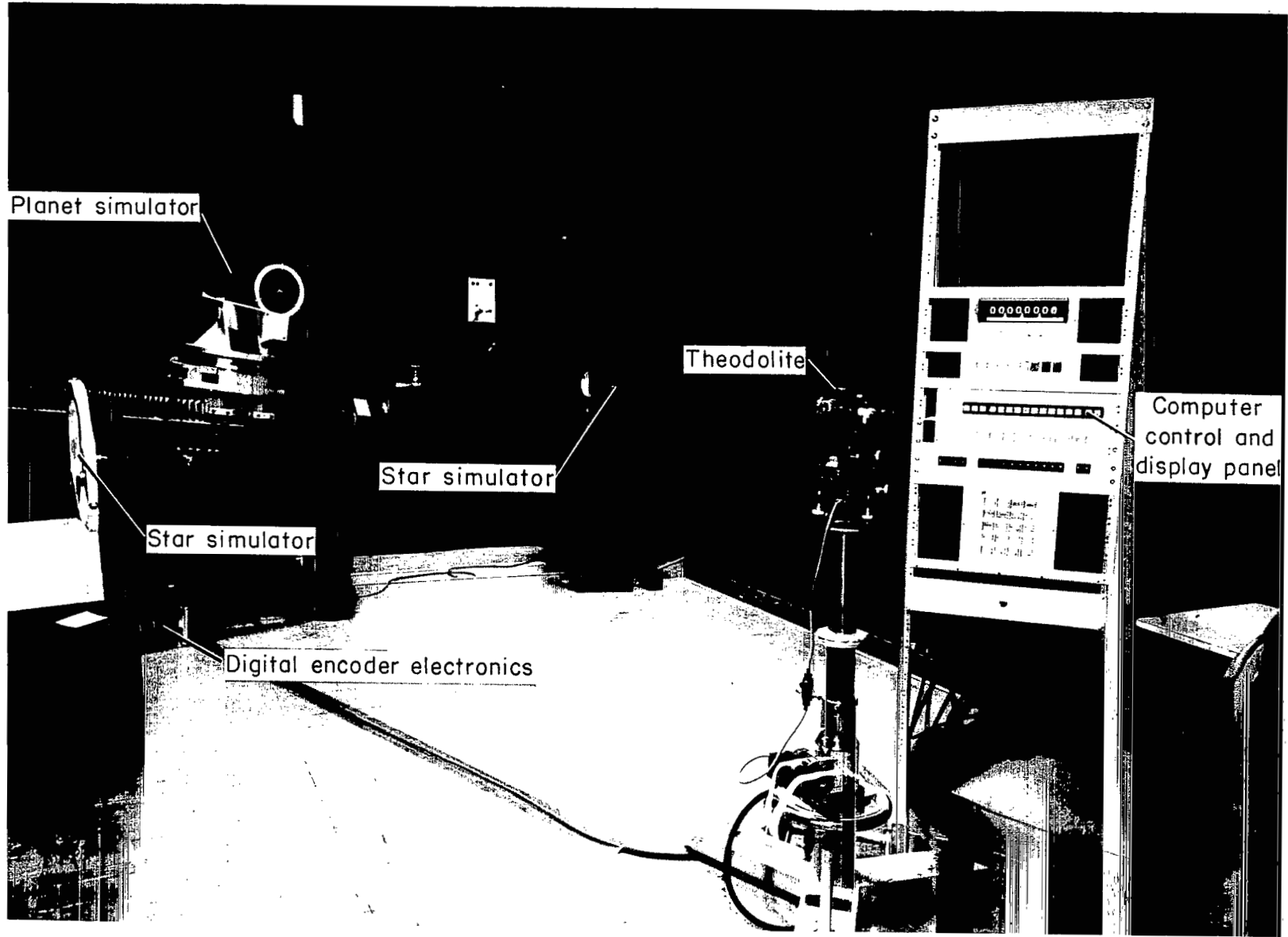
Figure 4 - Simulation equipment.

A-33296.1

Point light source

Parabolic mirror
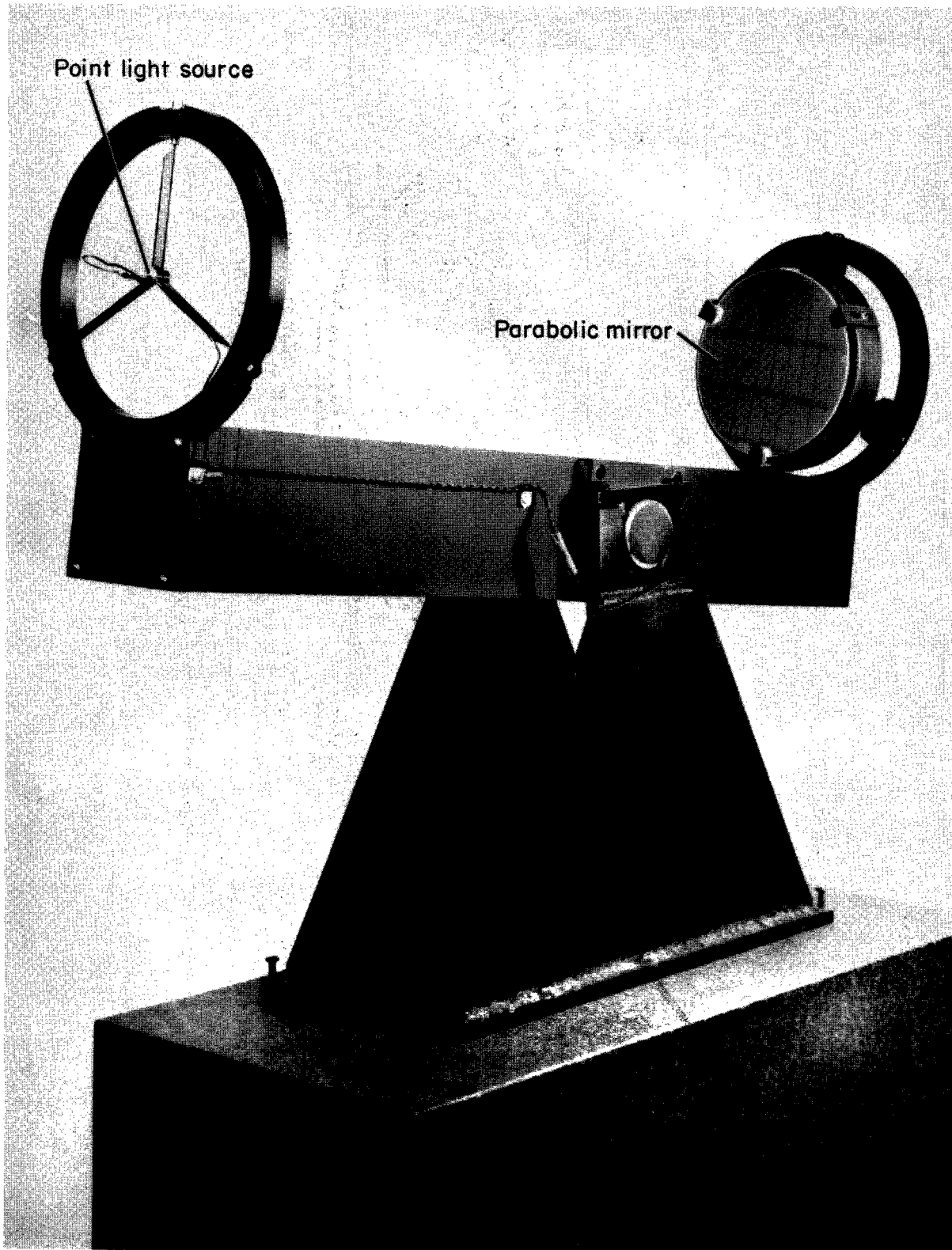
Figure 5.- Star simulator.

A-33839.1

Initial conditions at 70.68000 hr**

71.0 hr

20 observations

80.5 hr

First return
velocity correction
at 81.0 hr

109.0 hr

7 observations

115.0 hr

122.0 hr*

Second return
velocity
correction
at 116.0 hr

123.0 hr

12 observations

134.0 hr

Reference
perigee
144.87527 hr

Final velocity
correction at
135.0 hr

* Starting point for phases I and II

** Starting point for phase III
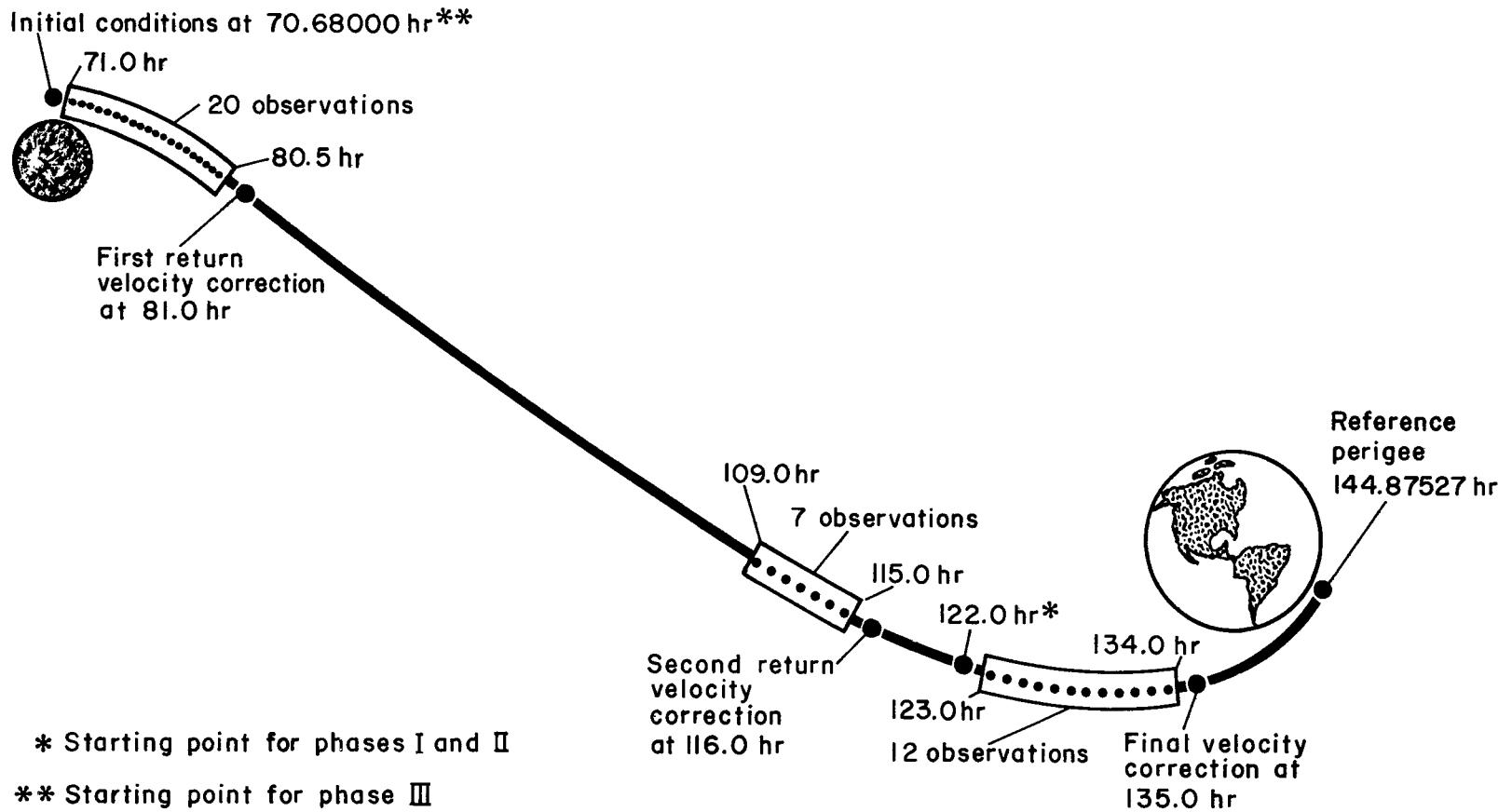
Figure 6.- Midcourse simulation return trajectory.
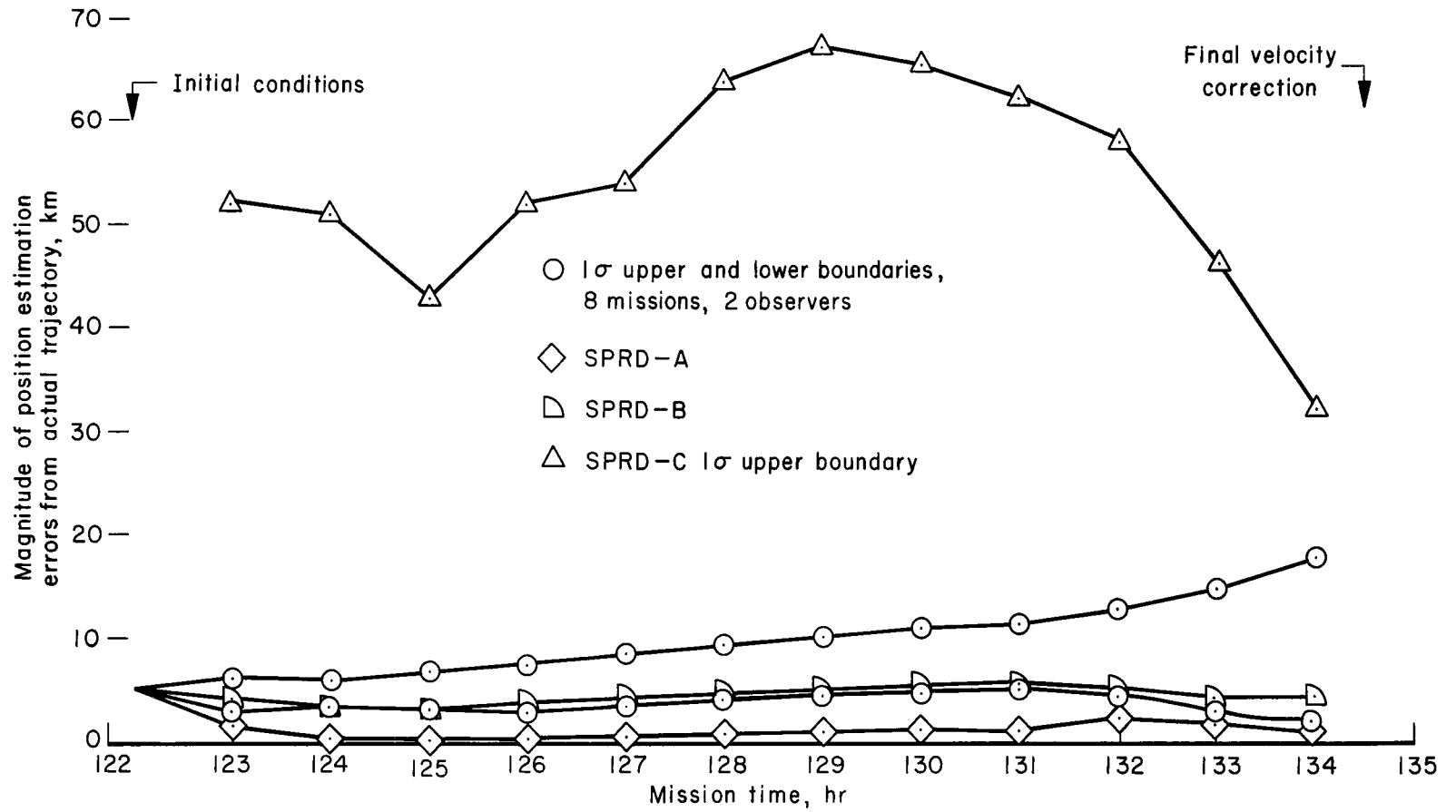
Figure 7.- Position estimation errors, phases I and II.
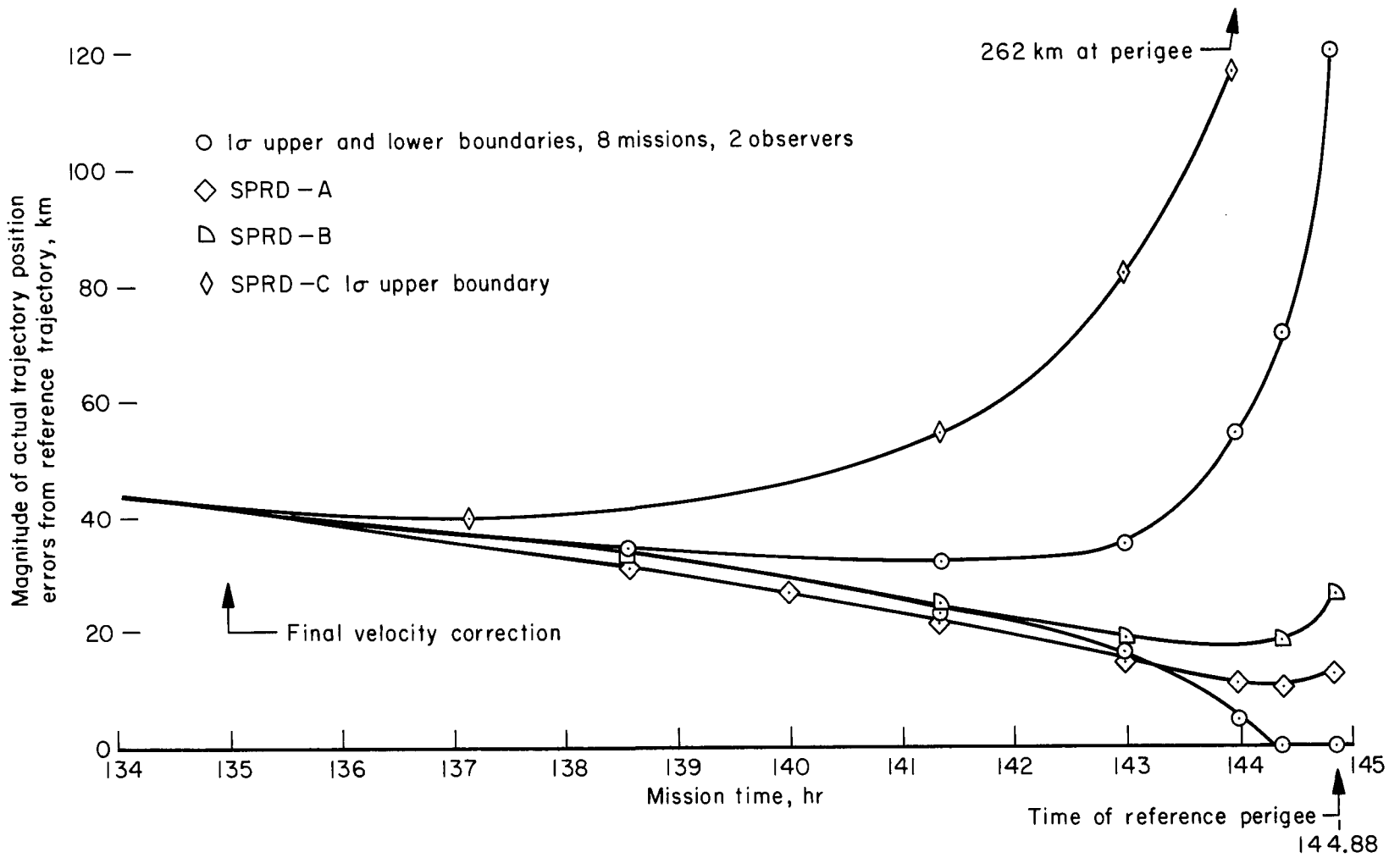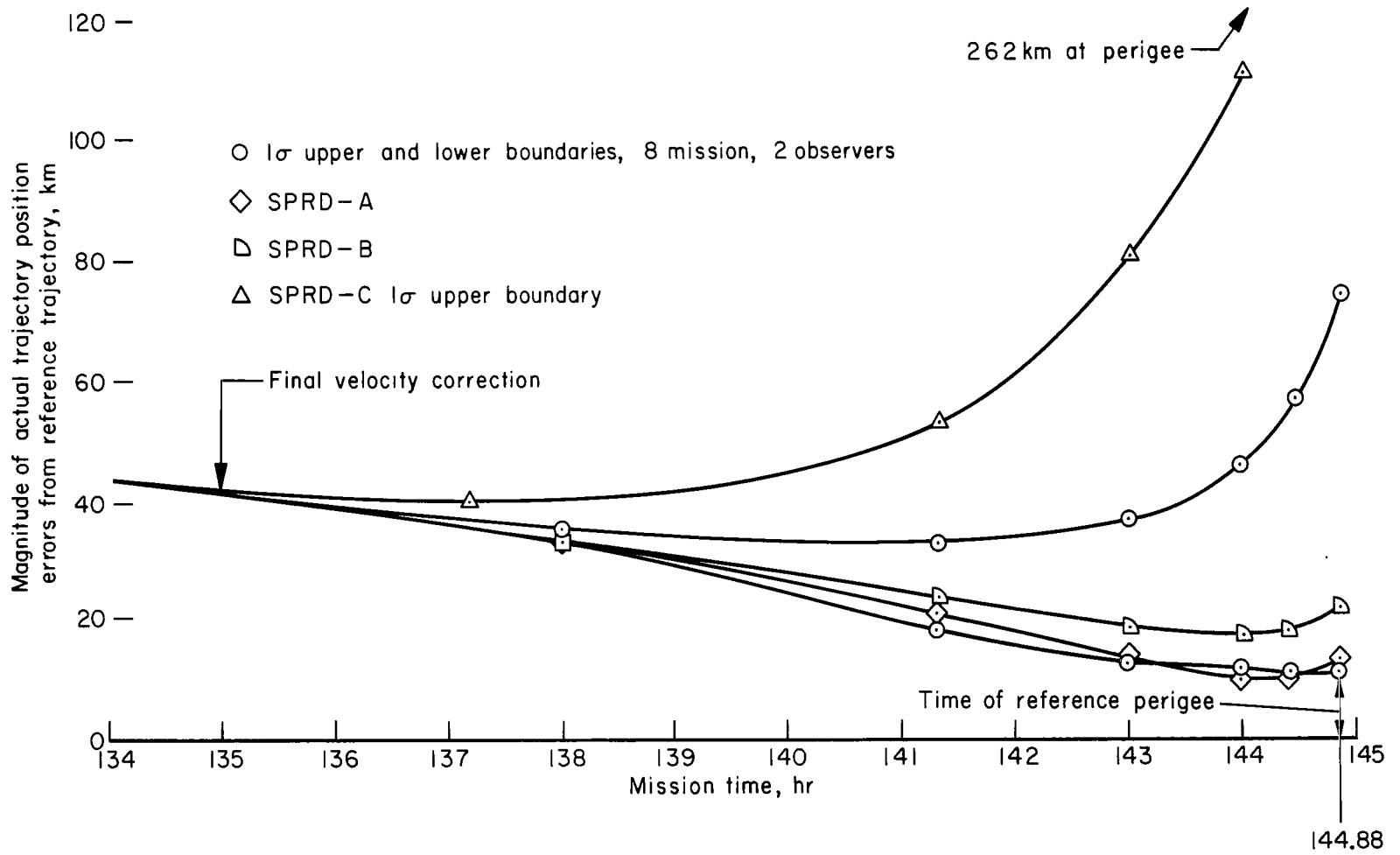
Figure 8.- Actual trajectory position errors, phase I.

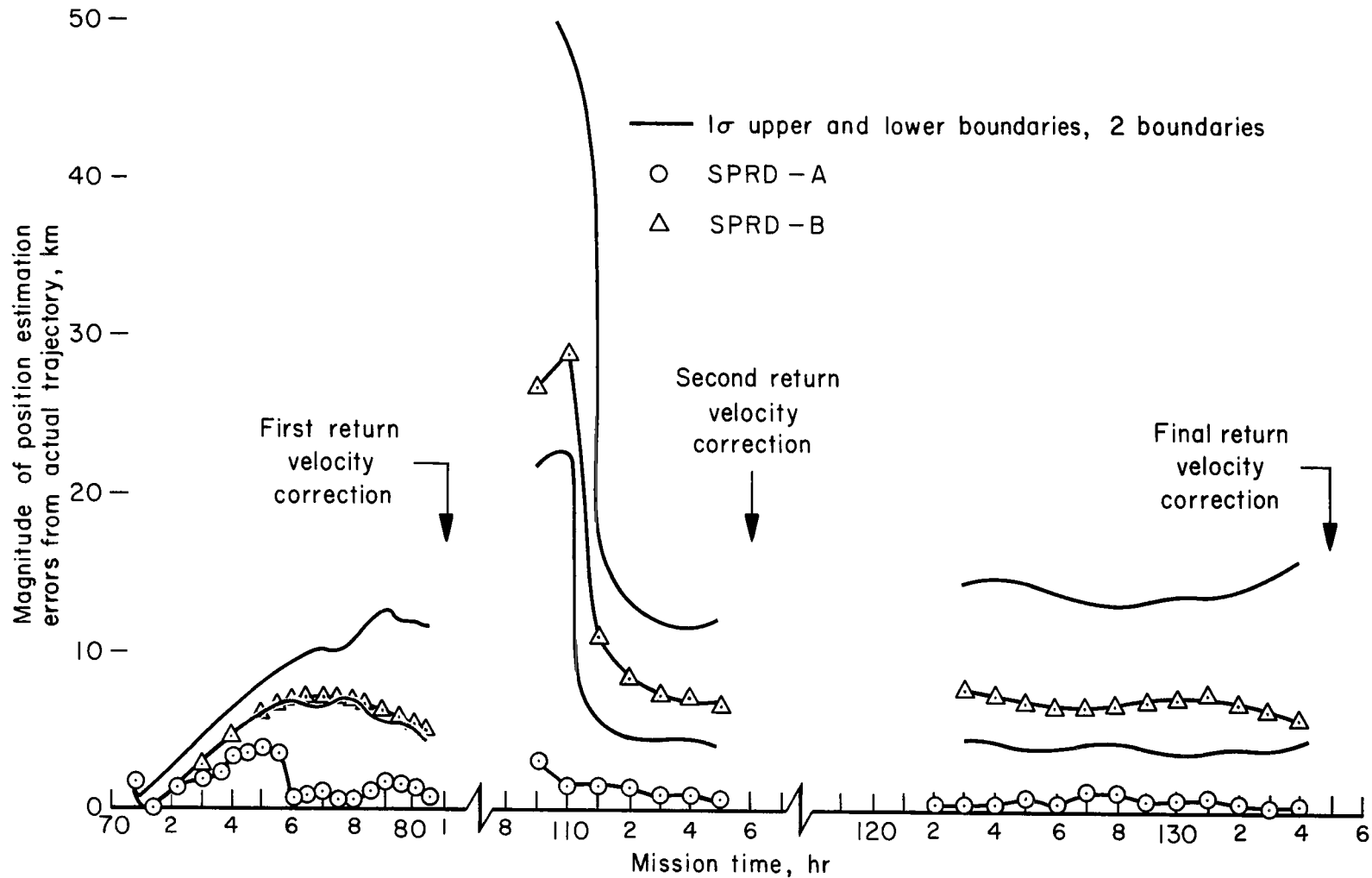Figure 9.- Actual trajectory position errors, phase II.
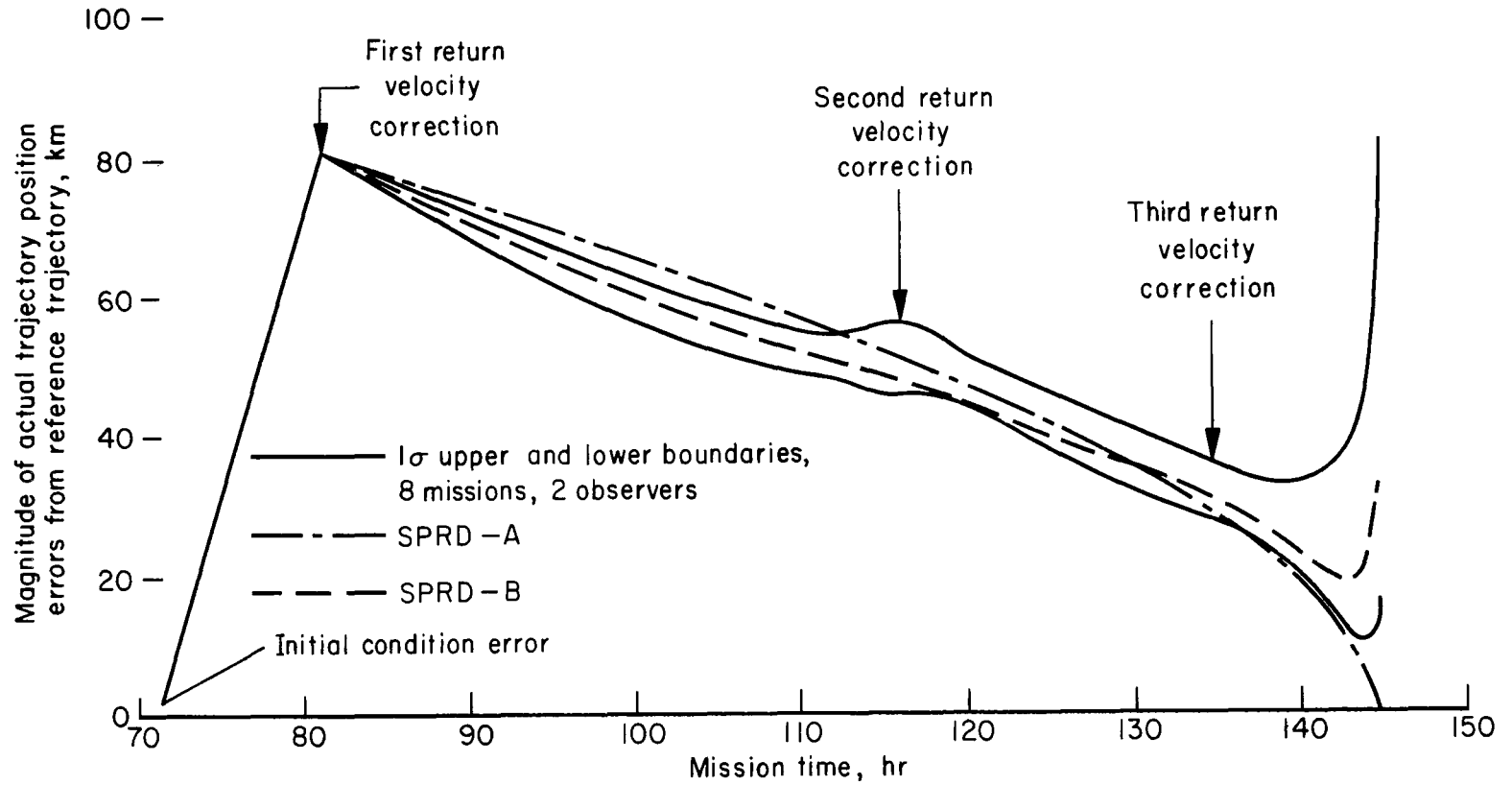
Figure 10.- Position estimation errors, phase III.

Figure 11.- Actual trajectory position errors, phase III.