493
10-6-82
JWB

I-5716

(1)

881

SANDIA REPORT

SAND82-1954 • Unlimited Release • UC-66c

Printed September 1982

SAND--82-1954

DE83 000405

The Wellbore Inertial Navigation System (WINS) Software Development and Test Results

MASTER

Ralph Wardlaw, Jr.

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE. It
has been reproduced from the best available
copy to permit the broadest possible avail-
ability.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

SAND82-1954
Unlimited Release
Printed September 1982

Distribution
Category UC-66c

The Wellbore Inertial Navigation System (WINS) Software Development and Test Results

Ralph Wardlaw, Jr.
Exploratory Systems Division II, 1622
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

The structure and operation of the real-time software developed for the Wellbore Inertial Navigation System (WINS) application are described. The procedure and results of a field test held in a 7000-ft well in the Nevada Test Site are discussed. Calibration and instrumentation error compensation are outlined, as are design improvement areas requiring further test and development. Notes on Kalman filtering and complete program listings of the real-time software are included in the Appendices. Reference is made to a companion document which describes the downhole instrumentation package.

NOTICE

PORTIONS OF THIS REPORT ARE ILLEGIBLE. It has been reproduced from the best available copy to permit the broadest possible availability.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Acknowledgment

The basic structure of the real-time software described in the document, along with the majority of the subroutines involved were the result of the previous investigative effort of A. C. Watts.

Contents

Introduction	7
System Description	7
Downhole Equipment.....	8
Surface Equipment.....	8
System Data Flow.....	8
Real-Time Software	8
General	8
Task Descriptions	9
Instrument Calibration and Compensation	13
Field Test.....	14
Topics for Further Investigation.....	19
Navigation Update Frequency.....	19
Kalman Filter Cycle Time.....	20
Additional Filter States	20
Other Measurements	20
Gyro Compassing	20
Conclusions.....	21
APPENDIX A — Notes on Kalman Filter	23
APPENDIX B — Computer Program Listings.....	31
References.....	77

Figures

1 WINS Functional Block Diagram	8
2 WINS Software Structure	10
3 Strapdown Systems	10
4 WINS Field Calibration Facility	13
5 Field Test Site	15
6 Tool in Wellhead Alignment Fixture	15
7 Run 1 North Deviation	16
8 Run 2 North Deviation	16
9 Run 1 East Deviation	16
10 Run 2 East Deviation	16
11 Run 2 Indicated North Deviation	16
12 Run 2 Indicated East Deviation	16
13 Run 2 X Gyro Downhole Estimated Bias vs Depth	17
14 Run 2 Y Gyro Downhole Estimated Bias vs Depth	17
15 Run 2 Z Gyro Downhole Estimated Bias vs Depth	17
16 Run 2 Y Accelerometer Downhole Estimated Bias vs Depth	18
17 Run 2 Z Accelerometer Downhole Estimated Bias vs Depth	18
18 Run 2 X Accelerometer Scale Factor Estimate vs Depth	18
19 Run 2 Downhole Vertical Velocity History	18
20 Run 2 Downhole North Velocity History	19
21 Run 2 Downhole East Velocity History	19

Tables

1	WINS Data Frame	11
2	Subroutines Used by Navigation Task	11
3	Subroutines Used by Kalman Filter Task	13
4	32-Hz Instrument Counts Tool in Motion	20

The Wellbore Inertial Navigation System (WINS) Software Development and Test Results

Introduction

For the past several years, Sandia National Laboratories has been involved in a program to investigate improved accuracy in the surveying of wellbores. The program began with a preliminary system design and analysis phase (Reference 1) which indicated that a Kalman-aided inertial navigation system could provide surveying accuracies several orders of magnitude better than was currently available. Because of these favorable predictions, a parallel hardware and software development of a system called Wellbore Inertial Navigation System (WINS) was initiated to demonstrate feasibility. The culmination of this activity was a successful field test that was performed on May 5, 1982, at DOE's Nevada Test Site.

The hardware development was conducted by Guidance and Control Division 2332 and is discussed in Reference 2 along with the field test results relating to tool performance. The purpose of this report is to document the real-time software that was developed in parallel with the hardware and to present field test results of system performance.

A following section briefly describes each of the system elements and the data flow between them.

Another section explains the real-time software that was used during the NTS feasibility test. To support the various inertial instrument models included in the software, a calibration procedure was developed and is described.

The next-to-the-last section presents the detailed test results, and the final section discusses conclusions that have been reached thus far as well as several areas that should be investigated further.

Appendix A presents the Kalman filter equations implemented by the software. Appendix B provides complete computer program listings.

System Description

The basic WINS concept involves using a three-axis inertial navigation system that is stopped periodically during a wellbore survey to obtain measurements of velocity errors. These velocity errors are operated upon by a Kalman filter to provide estimates of all system errors including instrument errors. This inertial navigation system is mechanized as a small diameter platform having a single gimbal to provide platform stabilization against rotation around the longitudinal axis. This platform and associated electronics represent the surveying tool. An associated uphole digital computer performs strapdown navigation computations and Kalman filter computations. The block diagram (Figure 1) depicts the major functional elements of WINS.

The term "strapdown" refers to navigation systems in which the gyros and accelerometers are body-fixed instead of being mounted on a fully gimballed platform. In WINS, the navigation equations assume a strapdown system even though one gimbal is employed. The advantages offered by this approach over that of a fully gimballed system include lower cost, smaller diameter, and higher reliability. In addition, the Kalman filter approach to aiding provides optimum accuracy because it estimates the error in each state variable at each update.

Figure 1. WINS Functional Block Diagram

Downhole Equipment

The downhole equipment consists of the single-axis platform with instrument cluster, together with appropriate electronics to provide roll axis stabilization, digitizing of instrument outputs, transmission of instrument and other data to the surface, and reception of commands from the surface. The design is based on use of a seven-conductor logging cable. Data transmission is serial current loop at 4096 baud.

Surface Equipment

Equipment required at the wellhead includes a logging truck, boom truck, calibration facility, and an operations trailer. Signal lines are patched from the logging truck to the operations trailer which contains the computer system.

The computer is a Data General Eclipse Model S-250 with 196K memory, 10M-byte disk, 9-track tape drive, and floating point hardware. The WINS I/O interface of special design is mounted in one of the mainframe I/O slots.

The CRT terminal is a standard product of Ann Arbor Terminals, Inc, and provides for cursor positioning under software control.

Analog recorders may be used to record the real-time serial data being sent by the downhole equipment. Playback of these tapes would allow 'replay' of the survey operation if necessary.

System Data Flow

Digitized instrument outputs and other data are transmitted uphole in basic 'frames' of twelve 10-bit words, each word consisting of a start bit, 8 data bits, and a stop bit. The frames are transmitted at a rate of

32 frames per second. On the surface, the data is routed to the WINS I/O interface module in the computer. As each 8-bit data word is received, the WINS I/O interface generates an interrupt to the computer, causing an interrupt routine to input and store the 8-bit data in a buffer area. After the twelfth such transfer, the interrupt routine alerts the WINS software that a new frame of data is available. The WINS software immediately transfers this data to working storage and initiates a navigation update cycle. Meanwhile, the interrupt routine continues to store the incoming 8-bit data bytes of the next frame. By the time this frame is complete, the navigation update cycle has been completed and the WINS software is awaiting the next buffer full notification. At 1-s intervals, selected navigation and system parameters are sent to the CRT for real-time display, and a record containing current values of certain system variables is written to the magnetic tape drive, providing data for subsequent analysis and reduction.

Commands to the downhole equipment can be selected and issued in real time by operator inputs at the CRT terminal. Commands are available to cage to a specified gimbal cycle, to initiate various slewing modes, and to uncage.

Real-Time Software

General

The software for WINS is structured to operate under the DATA General Real-Time Disk Operating System (RDOS) in a multitasking environment. Most of the routines are written in DG FORTRAN 5, which

includes provisions for calls to the operating system in real time.

Multitasking is a feature provided in RDOS which allows a large real-time process, such as WINS, to be broken down into several asynchronous functions, each having its own logically complete line of execution. These are called tasks, and each has an assigned priority level. During program execution, CPU control is given to the task of highest priority which is 'ready' to execute. The state of readiness of a task can be controlled by the task itself, by other tasks, and by the operating system. A task scheduler within RDOS keeps track of the state of readiness of each task and awards CPU control. During I/O operations with slow peripherals, the task involved loses CPU control while waiting for peripheral response.

The primary objectives of multitasking are to obtain maximum utilization of the CPU and to simplify programming of real-time processes that are too complex for single-task execution. The program structure of a typical task takes the form of a main routine whose name represents the name of the task and some number of subroutines that are called by the main routine.

The WINS software is broken down into the six functions shown in boxes in Figure 2. The name of the main routine of each task is given in parentheses. Subroutines called under each task are indicated. Certain utility routines have been omitted.

Intertask communication of data is accomplished through the use of labeled common block storage.

Task Descriptions

Initialization (Start)

This routine is the first one loaded by the loader and is considered to be the task of top priority by default. When execution begins, this routine executes first and then permanently removes itself from contention. It serves to initialize the various input parameters required, to identify various other tasks, to initialize the WINS I/O interface, and to identify the interrupt handling routine.

Navigation (SHREC)

In strapdown systems, gimbaling is done electronically by computer, and the instruments must have a dynamic range adequate to measure the body inertial

inputs and to function in the presence of same. Rotation-induced error mechanisms must either be designed out or be modeled in the software.

The WINS system is not fully gimbaled and is considered a strapdown system in which the singly gimbaled platform is the body. (The WINS navigation software is illustrated in Figure 2.)

Gyro data is converted to the proper units, and adjustments are made for bias and misalignment (see bottom of Figure 3). A further adjustment is made for the effects of angular acceleration about the other axis of that particular two-axis gyro.

The data is then operated on by a "Bortz" algorithm (Reference 3), which compensates for the noncommutativity of angular rotations. That is, a given set of rotations about the three axes will result in different final attitudes, depending upon the sequence in which they are applied. This algorithm then computes a noncommutativity vector which is added to the observed angular rate vector. The resulting data is fed to an attitude algorithm which computes a new direction cosine matrix for transformation from body to navigation coordinates. These values are adjusted for bias, misalignment, scale factor asymmetry, and temperature sensitivity. In addition, compensation is made for the effects of angular velocity and angular acceleration due to accelerometer offsets from body axes, and for the effect of angular acceleration about the axis of rotation of the pendulous element.

Following these operations, the data representing the change in velocity in body coordinates is transformed to navigation coordinates, the gravity and coriolis corrections are applied, and the integration steps are carried out to update velocity and positions.

The direction cosine matrix is updated periodically to account for the earth rate input to the gyros and for the change in orientation of the navigation frame as translation occurs.

Following execution of the initializing task, this navigation task assumes highest priority. It initially awaits word from the interrupt handler that the input buffer is full. During the waiting period, the task is considered 'not ready' by the task scheduler, allowing CPU control by lower priority tasks. As soon as word is received (accomplished via system call), CPU control returns to this task and it proceeds to transfer in the frame of data from the buffer area.

Gyro data arrives in the computer in the form of binary numbers representing the net angular rotation which occurred about each body axis during the preceding time increment. Accelerometer data arrives in the computer as a binary number representing the net velocity change along each body axis during the previous time increment.

At this point, the twelve 8-bit bytes occupy the lower 8 bits of twelve 16-bit words. Table 1 indicates the contents of the data frame. The data is in binary format, each byte (with two exceptions) representing half of a 16-bit number. In the case of the gyros, both halves of the 16-bit data are contained in every frame, while the accelerometer data alternates between high byte and low byte in successive frames. Thus, there is a synchronization problem to be resolved at the outset so that the software can start out on the right foot, so to speak.

Table 1. WINS Data Frame

Word No.	Contents
1	X gyro low byte
2	X gyro high byte
3	Y gyro low byte
4	Y gyro high byte
5	Z gyro low byte
6	Z gyro high byte
7	X accelerometer byte*
8	Y accelerometer byte*
9	Z accelerometer byte*
10	Resolver byte*
11	A/D byte
12	Status byte**

*Low byte if status byte least-significant bit = 0
High byte if status byte least-significant bit = 1

**Upper 4 bits = platform status
Lower 4 bits = A/D channel

Once synchronization is established, navigation begins. The 8-bit bytes must be appropriately combined and/or saved and the multiplexed A/D data must be identified and stored. Gyro and accelerometer count accumulations for use in output to CRT and magnetic tape must also be accomplished.

The navigation update cycle is initiated by a call to subroutine NAV, which alternately executes an attitude update or velocity update cycle, depending on whether it is an odd or even frame. Calls are made to routines that implement the appropriate strapdown navigation algorithms. Routines called and rates at which called are indicated in Figure 3. Subroutine NAV also controls activation of the Kalman filter task to be described later. After the return from NAV, the display and mag tape tasks are activated if 1-s has elapsed since the previous activation. The task then branches back to the condition of waiting for the next

buffer full message. In the interim, the lower priority active tasks are provided CPU time by the task scheduler.

Table 2 indicates the functions of each of the subroutines called under this task. Together they provide a discrete time solution to the basic differential equations for attitude, velocity, and position. A more detailed discussion of the routines and the underlying theory may be found in Reference 4.

Table 2. Subroutines Used by Navigation Task

Name	Function
ALIGN	Performs auto leveling during initial alignment
GRVTY	Computes vertical component of gravity vector
ERTHGEO	Computes trig functions of latitude and meridional and prime radii
GCOMP	Compensates gyro systematic error (Bortz algorithm)
DIRCOS	Integrates corrected gyro output to form new DCM
ACOMP	Compensates accelerometer systematic error
ERTHNAV	Applies coriolis and centripetal acceleration, translation, and earth rate attitude corrections
NEALT	Updates velocity and position
NAVROT	Rotates DCM per ERTHNAV output
ORTHO	Restores DCM orthonormality
NAV	Coordinates calls to above routines, initiates filter task, applies filter corrections

Control (KBD)

This task is of second priority, but requires very little CPU time. Its normal condition is waiting for a

character to be input from the keyboard, during which time it is classified as not ready or suspended. When a key is depressed, the task is reclassified to ready and is allotted CPU time when no higher priority task is ready.

When the character is retrieved, this task kills the display task and blanks the CRT. Depending on which character is input, a branch is taken to one of several blocks of CRT I/O dialog resulting in a message being written on the screen followed by a wait for operator response. When the response is received, appropriate actions are taken to comply with the operator command. The display task is then reestablished, and the control task resumes waiting for a keyboard input. This task provides control over such activities as:

1. Display mode – either of two formats
2. Magnetic tape operation – add to tape, start new tape, stop
3. Navigation mode – initialize, align, navigate, update
4. Output spooling
5. Test title input
6. Selection and execution of WINS commands enable/disable interface, initiate transmission, run inertial, cage, slew, etc.

Display (DSPLY)

This task provides real-time CRT display of selected system variables. Two formats are available, one presenting navigation data and the other presenting Kalman filter data. These are operator selectable from the keyboard. This task initially writes the labels on the screen for the selected presentation format and then enters a waiting state until triggered by the navigation task. The latter occurs at 1-s intervals upon which the current value of the variables involved are written on the screen using cursor control to overwrite the previous values. On completion, the task returns to the condition of awaiting the next trigger.

Recording (MTAPE)

This task is activated at 1-s intervals by the navigation task. Its sole function is to write a single record to magnetic tape containing the current values of selected navigation and Kalman filter variables.

Kalman Filter (NAVKG)

This task is activated at 8-s intervals by the navigation tasks. The functions it performs depend upon the current mode of navigation, which can be one of three:

1. Alignment – establishing initial altitude with assumed zero heading
2. Navigate – strapdown inertial navigation
3. Update – navigation with Kalman filter updates based on assumed zero velocity

A discussion of the Kalman filter and the formulations used in WINS is given in Appendix A. The reader who is unfamiliar with the Kalman filter should refer to that material at this point. A more detailed presentation of the general filter equations in both continuous and discrete time form may be found in Reference 5.

The procedure for running WINS in a borehole begins with placing the tool in the top of the hole oriented with the Z axis pointing north. The system is initialized and a mode 1 alignment is run to establish the initial attitude. Following this, mode 3 is entered to allow the Kalman filter to update the estimates of the instrument biases while the tool is stationary. When this process stabilizes, mode 2 is entered for navigation, and the tool is run into the hole. After approximately one minute, the tool is brought to a stop and mode 3 is entered to allow updates of the system states. After two or three update cycles, mode 2 is selected and the tool is again lowered into the hole. This cycle is repeated continuously until the tool reaches target depth and returns to the top of the hole. Mode selection is done by the operator at the computer terminal and must be coordinated with the movement of the tool and hence with the operator of the logging truck.

As indicated in Appendix A, during periods when no error measurements are being made and hence no update, it is still necessary to propagate the covariance matrix forward in time in order for it to maintain currency as the covariance matrix of the estimation error. Hence, during periods when the tool is moving (mode 2), this task, at 8-s intervals, computes a new F matrix, state transition matrix, driving noise covariance, and updates the error covariance matrix according to the premeasurement equation.

When the tool is stationary, the F matrix is static, and it is not necessary to compute the state transition matrix and driving noise covariance more than once. Hence, in mode 3 these are not done, and the task computes the premeasurement error covariance, the Kalman gain matrix, and the postmeasurement error covariance. This cycle continues at 8-s intervals as long as mode 3 is selected. At the end of each cycle, the navigation task computes corrections to the system state variables using the Kalman gain matrix and the current velocity errors.

During alignment (mode 1), only the F matrix, state transition matrix, and driving noise covariance are computed each cycle so that they will be available on switching to mode 3, where they are not computed. The error covariance is not propagated during alignment to avoid perturbations induced by the initial misalignment. Table 3 indicates the function of the subroutines used by this task.

Table 3. Subroutines Used by Kalman Filter Task

Name	Function
NAVEM	Computes current F (15×15) matrix
DISMA	Computes state transition matrix
DISMB	Computes driving noise covariance matrix
RKINT	Implements Runge-Kutta integration for DISMA, DISMB
DXA, DXB	Adapts RKINT to DISMA, DISMB
KALMAN	Implements Kalman filter equations, updates covariance, computes Kalman gains
MULT	Multiplies matrices
INV	Inverts (3x3) matrix

Instrument Calibration and Compensation

The pretest calibration procedure used for the WINS platform is an 18-step process designed to measure accelerometer bias; misalignment and scale factor; and gyro bias, misalignment, scale factor, and g-sensitivity. It involves mounting the platform in a fixture which functionally represents a perfect cube (Figure 4), and placing it on a level granite slab having an attached true north reference plate.

The cube is placed in six different positions of known orientation so that each axis of the platform experiences an up and down orientation. While in each position, instrument outputs are accumulated for a motionless period of 5 minutes, and for 2 periods of

40 seconds each during which the cube is rotated 360° clockwise in one case and counterclockwise in the other.

Figure 4. WINS Field Calibration Facility

The system computer accumulates the data, and at the conclusion of the procedure, computes the desired variables. The static data allows calculation of all accelerometer variables plus gyro bias and g-sensitivity. The rotational data allows calculation of gyro scale factor and misalignment.

Subroutine ACOMP applies corrections to the accelerometer data to account for the following:

- Bias
- Misalignment
- Scale factor asymmetry
- Scale factor temperature dependence
- Accelerations resulting from rotation

Bias is considered to be constant and independent of temperature and is merely subtracted out. The effects of misalignments, which allow an accelerometer to sense cross-axis components of acceleration, are likewise computed and subtracted out.

Scale factor is modeled as a linear function of platform temperature. Asymmetry is provided for by utilizing separate models for positive and negative scale factors.

Platform and accelerometer temperatures are not directly controlled, but are influenced by temperature control of the gyros and the resultant heat input to the platform. Platform temperature rises toward gyro temperature for some 90 minutes before stability is reached under room ambient conditions. During calibration, scale factor vs temperature data is taken during this warmup to arrive at the appropriate model parameters for each accelerometer. The measurements are made by observing accelerometer output under 1-g inputs after correcting for bias.

Acceleration outputs arising as the result of pure rotation are also computed and corrected for in subroutine ACOMP. Since the accelerometers are not all located at a common point, each will be influenced to a different degree by rotations. It is necessary to apply corrections which will, in effect, translate the data to such a common point.

A mass m located at some distance $\underline{R}$ from the origin 0 of a reference frame will, under some rotation rate $\underline{\omega}$, experience an acceleration given by

$$\underline{a}_m = \underline{a}_0 + \underline{\omega} \times \underline{\omega} \times \underline{R} + \dot{\underline{\omega}} \times \underline{R}$$

from which

$$\underline{a}_0 = \underline{a}_m - \underline{\omega} \times \underline{\omega} \times \underline{R} - \dot{\underline{\omega}} \times \underline{R}$$

Gyro data is used in ACOMP to compute corrections corresponding to the two rightmost terms above.

An additional correction is applied to account for the fact that the accelerometers utilize a pendulous mass hinged to the case to sense acceleration. Angular acceleration about an axis parallel to the hinge axis will cause an output from the instrument. The magnitude of this effect is specified by the manufacturer to be $0.893 \times 10^{-3} \text{ g/rad/s}^2$.

Gyro data is compensated in routine GCOMP, where corrections are made to account for ordinary bias, acceleration-induced bias, and misalignment. Scale factor is considered to be constant and symmetrical.

The two-axis tuned rotor gyros used in WINS display a characteristic whereby an angular acceleration applied to one of the input axes causes an indicated rate on the other axis. This effect appears in the equations describing the dynamics of such gyros, and can thus be calculated for a particular gyro. For the gyros in question, this effect was calculated to have a magnitude of $9.0909 \times 10^{-4} \text{ rad/s/rad/s}^2$. A correction is made for this effect each time GCOMP is executed, using the two most recent gyro data points to obtain angular acceleration. This value is also used in ACOMP to compute the angular acceleration effects on the accelerometers.

The GCOMP routine also contains the Bortz algorithm referenced earlier in this report.

Field Test

On May 5, 1982, a field test was conducted at the DOE Nevada Test Site. An existing well located at latitude $37^\circ 16' 48.95'' \text{N}$, longitude $116^\circ 29' 21.21'' \text{W}$, in an area known as Pahute Mesa, was used to run the WINS system. Wellhead elevation was 6557 ft, with a total depth of 7500 ft. Casing of 9-5/8 in. ID extended to 7400 ft. Water stood in the hole at a depth of 2000 ft, and bottom hole temperature was 140°F . The well was not deviated and had not been surveyed. On site, equipment (Figure 5) included an operation trailer, assembly trailer, calibration shed, and contract logging vehicles operated by Birdwell, Inc.

Alignment prior to each run was accomplished by using the Wellhead fixture shown in Figure 6, which aligns the tool case relative to north. The platform is then caged to a position which places the X, Y, and Z axes up, east and north, respectively. An auto leveling routine was then used to establish initial tilt.

The tool was used to navigate during two round trips to a depth of 7000 ft. Partial disassembly was required between trips to replace the battery pack. Each round trip required about 2 hours, using the stop and go procedure described earlier. Verbal instructions to stop and go were passed from the operations trailer to the operator of the logging truck winching equipment. The real-time display of navigation variables and instrument outputs, which was available in the trailer, made it easy to verify when the tool had stopped. During periods of motion downhole, speeds of about 400 ft/min were generally attained. Uphole velocity was substantially slower, particularly at depth, due to limitations in the winching power available.

Figure 5. Field Test Site

Figure 6. Tool in Wellhead Alignment Fixture

During pretest calibrations and other tests of the system to examine accelerometer scale factor variation, it became evident that factors other than platform temperature were influencing the scale factors to a substantial degree. The magnitude of the errors was such that the 1-g input in the vertical channel would be sufficient to 'excite' the scale factor error and make it observable in the results, while in the horizontal axis, the inputs would not be sufficient to make the errors visible. It was further noted that accelerometer bias was relatively stable in comparison with scale factor.

For this reason, it was decided to eliminate the X (vertical) accelerometer bias as a state variable in the Kalman filter, and to substitute the scale factor. In this way, the filter could provide estimates of scale factor error during the update periods. This was the filter formulation used during the two runs into the hole on May 5.

The north, south deviations recorded during the downhole legs of the two surveys are shown in Figures 7 and 8, while the east, west deviations are shown in Figures 9 and 10. As can be seen, there is substantial agreement between the two surveys. Disagreement in horizontal position at target depth was <3 ft. Figures 11 and 12 present the unsmoothed navigator output recorded during the test, the discontinuities correspond to corrections applied by the filter during updates.

Figure 7. Run 1 North Deviation

Figure 8. Run 2 North Deviation

Figure 9. Run 1 East Deviation

Figure 11. Run 2 Indicated North Deviation

Figure 10. Run 2 East Deviation

Figure 12. Run 2 Indicated East Deviation

At bottom hole, comparison between indicated depth and logging cable length showed a difference of 20 ft in Run 1 and 13 ft in Run 2, indicating that the vertical scale factor problem had not been fully resolved.

Prior to Run 2, the initial scale factor variance and driving noise variance were increased to allow the filter to make larger error estimates during the surveys. This improved the results for Run 2 but did not resolve the problem, which will require further investigation.

The return trip up the hole was characterized by slow movement in the deepest and hence hottest portion of the hole. This prolonged temperature soak resulted in further deviation of scale factor and further accumulation of depth error. The result was that upon arrival at the top of the hole, there was a 80-ft depth error for Run 1 and a 60-ft error for Run 2. It should also be noted that the return trip of Run 2 was speeded up by reducing the length of the stop periods.

Horizontal closure was good in both cases considering the overall trip of 14K ft. On completion of the surveys, indicated displacement was 10°N, 5°E for Run 1 and 2°N, 2.5°E for Run 2. Heading error was checked on completion of each run and was found to be 1.2 and 2.7 degrees. This error is too large for highly deviated holes and warrants further investigation. Possible error sources include scale factor asymmetry, coning, resolver (reference cage position) drift, and data transmission errors.

Figures 13 through 17 show the filter estimates of gyro and accelerometer bias as a function of depth during the downhole leg of Run 2.

Figure 13. Run 2 X Gyro Downhole Estimated Bias vs Depth

Figure 14. Run 2 Y Gyro Downhole Estimated Bias vs Depth

Figure 15. Run 2 Z Gyro Downhole Estimated Bias vs Depth

Figure 16. Run 2 Y Accelerometer Downhole Estimated Bias vs Depth

Figure 17. Run 2 Z Accelerometer Downhole Estimated Bias vs Depth

Figure 18 shows the estimated X accelerometer scale factor as a function of depth during the complete round trips of Run 2. The upper portion of the curve corresponds to the downhole leg. Note the pronounced change in SF during the uphole leg, while tool temperature is at or near maximum. These effects are discussed more fully in Reference 2.

Figure 18. Run 2 X Accelerometer Scale Factor Estimate vs Depth

Figures 19, 20, and 21 depict the velocity history during the downhole leg of Run 2.

Figure 19. Run 2 Downhole Vertical Velocity History

Figure 20. Run 2 Downhole North Velocity History

Figure 21. Run 2 Downhole East Velocity History

Topics for Further Investigation

A number of questions relating to the system hardware and software remain unanswered at this point. These should be resolved before committing to a final design for a system such as WINS. Some can be investigated by playing back the analog tapes of the tool output data stream during the field test. Others will require further field testing.

Some of these questions are outlined and discussed briefly below. The basic objectives of these investigations are to identify and eliminate the sources of error in heading and vertical closure.

Navigation Update Frequency

The system navigation cycle rate is 16 Hz. The 32-Hz gyro data is applied as such only in the coning compensation algorithm. By strapdown navigation standards, this is a low rate. Table 4 provides a brief glimpse at the instrument output counts being transmitted at 32 Hz while the tool is in motion. To be assured that the rate is high enough, we would like to see only small variation in successive counts, demonstrating that the accumulation interval is small relative to the rate of change of the instrument outputs. The navigation calculations are based on the assumption that angular rates and velocity are constant during each interval. This is not the case in the example. The 16-Hz rate is used in WINS because it appeared that 4096 BPS would be the maximum that could be transmitted reliably over a standard logging cable. From this limitation evolved the data format and rate previously described.

Table 4. 32-Hz Instrument Counts Tool in Motion

Time	Gyros			Accels		
	X	Y	Z	X	Y	Z
582.28	96	-5	-5	373	-8	6
582.31	64	-6	-5			
582.34	-26	0	-6	371	3	-30
582.38	-67	3	-2			
582.41	-56	1	3	384	-3	15
582.44	-50	1	0			
582.47	-43	0	-1	395	-15	-4
582.50	-17	-3	-8			
582.53	-21	2	0	390	0	-26
582.56	-13	8	7			
582.59	59	5	-4	385	3	6
582.63	54	-13	-5			
582.66	9	-9	-1	373	-7	6
582.69	-16	-7	3			
582.72	-42	9	7	362	-1	-30
582.75	52	9	-2			
582.78	72	-4	-8	366	-15	9
582.81	0	-3	2			
582.84	-33	0	19	382	9	-3
582.88	-19	5	4			
582.91	-22	4	-16	420	-25	-31
582.94	-47	0	-3			
582.97	-45	0	4	408	18	22

If an additional uphole data channel could be implemented, the rate could be at least doubled. Reformatting and reducing the field width allowed for instrument outputs could also increase the rate.

Kalman Filter Cycle Time

It is difficult to assess the improvement that would accrue if the filter cycle could be reduced from the present 8 s. This cycle period was determined by machine limitations since the filter had to run on a

spare time basis, and 8 s is required to accumulate sufficient spare time to execute the filter task.

Ideally, the filter would run at the navigation cycle rate to allow the state transition matrix to follow the detailed movement of the platform. The machine burden to accomplish this is too great, and it was reasoned that changes in attitude and velocity to be encountered in the borehole would be sufficiently limited to allow the 8-s interval.

To substantially increase the filter rate, it would be most advisable to use a multiprocessor configuration in which the filter software would be executed in a dedicated processor. This would provide the necessary throughput capability to try various filter cycle rates.

Additional Filter States

It would be desirable to retain the original 15 states and to add such other states as are deemed necessary. For example, the vertical accelerometer scale factor state would be added. In highly deviated holes, errors in the lateral accelerometer scale factor may become observable and these would be added. Other states may arise from attempts to model errors. For example, the scale factor error statistics may be considered white noise, in combination with a linear time or temperature dependence whose slope may be considered a state having some variance and noise statistics.

Other Measurements

Any measurement of an error in a state variable, if sufficiently accurate, should improve the aiding process and hence the accuracy. For example, the length of cable paid out is a measure of distance traveled by the tool. The integration of the body accelerations along the borehole axis should also yield distance traveled. If this quantity is added as a state variable and computed by the navigation software, a measurement of the errors in this state would be available during the update intervals and could be used to enhance performance.

Gyro Compassing

If significant horizontal movement is anticipated, as in a deviated well, an initial heading error can project to a large position error. A two-position gyro compassing alignment utilizing a Kalman filter can provide a more accurate initial heading estimate and is more convenient than the use of mechanical methods to attain a known initial heading.

Conclusions

The test reported here should be viewed as the first field test of a developmental system. East and north surveying accuracy was excellent and in agreement with theoretical predictions. However several questions remain to be resolved before a final design

can be proposed. Effort in this direction has been curtailed for reasons of funding, and will be limited to playback trials of certain software and input parameter changes. It is believed that the feasibility of this concept has been demonstrated with sufficient success to warrant continued development by the industry.

APPENDIX A

Notes on Kalman Filter

The Kalman filter is a particular form of an optimal estimator for linear systems. An optimal estimator is a computational algorithm that processes measurements to deduce a minimum (in some well-defined sense) error estimate of the state of a system by utilizing knowledge of the system dynamics, statistics, and initial conditions.

Linear System Description

The system state vector at time t_k is given by

$$\underline{X}_k = \Phi_{k-1} \underline{X}_{k-1} + \underline{\omega}_{k-1}, \quad (A1)$$

where $\underline{\omega}$ is a zero mean, white noise sequence of covariance B_k . Thus, the state vector at any time t_k is a linear function of the state vector at the previous time t_{k-1} , plus system noise. Φ is known as the *state transition matrix*.

Measurements are taken at t_k and can be represented as

$$\underline{Z}_k = H_k \underline{X}_k + \underline{v}_k.$$

The measurements $\underline{Z}_k$ are linear combinations of the system state variables, plus measurement noise, $\underline{v}_k$ (of covariance R_k). The problem solved by the Kalman filter is how to combine the measurement information $\underline{Z}_k$ with existing *estimates* of the system states, $\hat{\underline{X}}_k$, to produce new state estimates.

It is shown in Reference 5 (pp 107-108) that one may seek a new estimate of the form

$$\hat{\underline{X}}_k(+) = \hat{\underline{X}}_k + K_k [\underline{Z}_k - H_k \hat{\underline{X}}_k], \quad (A2)$$

where $\hat{\underline{X}}_k(+)$ is the new estimate, $\hat{\underline{X}}_k$ is the old (premeasurement) estimate, and $\underline{Z}_k$ and H_k were previously defined.

K_k is a *gain matrix* to be chosen to minimize the variance of the error in the estimate $\hat{\underline{X}}_k(+)$. When K_k is so chosen, it is called the *Kalman gain matrix*. Let $\underline{e}_k = \hat{\underline{X}}_k - \underline{X}_k$, the error in the estimate. Define $P_k = E\{\underline{e}_k \underline{e}_k^T\} = \text{error covariance matrix}$.

It is shown in Reference 5 that choosing

$$K_k = P_k H_k^T [H_k P_k H_k^T + R_k]^{-1}, \quad (A3)$$

where

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + B_{k-1} \quad (A4)$$

will minimize the diagonal elements of P_k , the error variance.

Application to the Wellbore Inertial Navigation System

The application of Kalman filtering to the wellbore mapping problem is dependent on the existence of a linearized error model for an inertial navigator. The model equation is

$$\frac{d}{dt} [\delta \underline{x}(t)] = F(\underline{X}, t) \delta \underline{x}(t) + G(t) \underline{U}(t). \quad (A5)$$

$\delta \underline{x}$ is a vector with elements and units as shown in Table A1. Note that these elements are the errors in the system states.

Table A1. System State Variables

State Variable	Symbol	Description	Units
δx_1	$\delta \lambda$	Longitude error	rad
δx_2	δL	Latitude error	rad
δx_3	δH	Altitude error	ft
δx_4	δv_E	East velocity error	ft/s
δx_5	δv_N	North velocity error	ft/s
δx_6	δv_Z	Up velocity error	ft/s
δx_7	ϵ_E	East axis attitude error	rad
δx_8	ϵ_N	North axis attitude error	rad
δx_9	ϵ_z	Up axis attitude error	rad
δx_{10}	ΔGB_x	X gyro bias error	rad/s
δx_{11}	ΔGB_y	Y gyro bias error	rad/s
δx_{12}	ΔGB_z	Z gyro bias error	rad/s
δx_{13}	ΔAB_x	X accelerometer bias error	ft/s ²
δx_{14}	ΔAB_y	Y accelerometer bias error	ft/s ²
δx_{15}	ΔAB_z	Z accelerometer bias error	ft/s ²

F is a matrix as given in Tables A2, A3, and A4. Its elements are continuously updated from outputs of the navigation software.

$G(t)\underline{U}(t)$ is a driving or input noise function where

$$G(t) = \begin{bmatrix} 0_{9 \times 6} \\ I_{6 \times 6} \end{bmatrix}, \text{ and}$$

$\underline{U}(t)$ is a vector, the six elements of which are random noise associated with the random bias of the inertial instruments (gyros and accelerometers).

The spectral density matrix associated with $\underline{U}(t)$, Q , is used in the calculation of the process noise covariance matrix, B_k , as demonstrated later.

Update Calculation Procedure

The state transition matrix Φ is related to the F matrix (as defined above) by $\dot{\Phi} = F\Phi$.

As the navigation calculations proceed, F is updated at the basic filter cycle rate. Then an updated value of Φ is obtained by integration of the above equation. Likewise, an updated value of the noise covariance B_k is calculated as described later. When the tool is moving, with each new value of Φ and B_k , a new error estimated $\hat{\underline{x}}$ and covariance P_k are computed.

$$\hat{\underline{x}}_k = \Phi_{k-1} \hat{\underline{x}}_{k-1} \quad (A6)$$

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^t + B_{k-1} \quad (A7)$$

Table A2. Linear System Matrix Used in Error Model

$F_{9 \times 9}$	$0_{3 \times 6}$	
	$0_{3 \times 3}$	$C_{P_{3 \times 3}}^N$
	$C_{P_{3 \times 3}}^N$	$0_{3 \times 3}$
$0_{6 \times 15}$		

C_P^N = Transformation from platform to navigation coordinates

Table A3. WINS Fundamental Error Model Matrix⁽²⁾

0	$\rho_z / \cos L$	$-\rho_n / R \cos L$	$1/R \cos L$	0	0	0	0	0
0	0	ρ_e / R	0	$1/R$	0	0	0	0
0	0	0	0	0	1	0	0	0
0	F_{42}	F_{43}	F_{44}	$(\omega_z + \Omega_z)$	$-(\omega_n + \Omega_n)$	0	$-f_z$	f_n
0	F_{52}	F_{53}	$-2\omega_z$	$-k_z$	ρ_e	f_z	0	$-f_e$
0	$-2\Omega_z v_e$	F_{63}	$2\omega_n$	$-2\rho_e$	0	$-f_n$	f_e	0
0	0	$-\rho_e / R$	0	$-1/R$	0	0	ω_z	$-\omega_n$
0	$-\Omega_z$	$-\rho_n / R$	$1/R$	0	0	$-\omega_z$	0	ω_n
0	F_{92}	$-\rho_z / R$	$\tan L / R$	0	0	ω_n	$-\omega_e$	0

Table A4. Elements of F_{9x9}

L	Latitude of vehicle
Ω	Earth rotation rate
R	Earth radius
g	Magnitude of gravity vector
v_e, v_n, v_z	Components of vehicle velocity with respect to earth
f_e, f_n, f_z	Components of specific force
$\left. \begin{aligned} \Omega_n &= \Omega \cos L \\ \Omega_z &= \Omega \sin L \end{aligned} \right\}$	Components of earth rate
$\left. \begin{aligned} \rho_e &= -v_n/R \\ \rho_n &= v_e/R \\ \rho_z &= v_e \tan L/R \end{aligned} \right\}$	Components of angular velocity of E-N-Z frame with respect to earth
$\left. \begin{aligned} \omega_e &= \rho_e \\ \omega_n &= \rho_n + \Omega_n \\ \omega_z &= \rho_z + \Omega_z \end{aligned} \right\}$	Components of angular velocity of E-N-Z frame with respect to inertial space
$k_z = v_z/R$	
$F_{42} = 2(\Omega_n v_n + \Omega_z v_z) + \rho_n v_n / \cos^2 L$	
$F_{43} = \rho_z \rho_e + \rho_n k_z$	
$F_{44} = -\rho_e \tan L - k_z$	
$F_{52} = -2\Omega_n v_e - \rho_n v_e / \cos^2 L$	
$F_{53} = \rho_n \rho_z - \rho_e k_z$	
$F_{63} = 2g/R - (\rho_n^2 + \rho_e^2)$	
$F_{92} = \omega_n + \rho_z \tan L$	

Every minute or so, the tool is brought to a stop. At these times, the tool velocity is assumed to be zero. Therefore, any navigator-indicated velocities are direct measurements of velocity errors. The measurements equation is

$$\underline{Z}_k = H_k \delta \underline{x}_k + \underline{v}_k \quad (A8)$$

$$H_k = [O_{3 \times 3} \mid I_{3 \times 3} \mid O_{3 \times 9}] \quad (A9)$$

Note that $\delta \underline{x}_k$ is the actual error, not the error estimate. H_k is a "selection" matrix which simply serves to limit the measurements to the three velocities only. All other terms are zero. $\underline{v}_k$ is a measurement noise sequence of assumed covariance R_k . R_k will be discussed later.

The measured velocity errors $\underline{Z}_k$ (obtained when the system is at rest) are used to update (or correct) the tool navigation parameters as follows. Recall that P_k , B_k , and $\delta \underline{x}_k$ are available as a consequence of the navigation and covariance propagation calculations. The Kalman gain may be calculated from

$$K_k = P_k H_k^T [H_k P_k H_k^T + R_k]^{-1} \quad (A10)$$

Next, the updated estimate of the states $\delta \underline{x}(+)$ is calculated:

$$\delta \underline{x}(+) = \delta \underline{x} + K_k [\underline{Z}_k - H_k \delta \underline{x}_k] \quad (A11)$$

Recall that the states $\delta \underline{x}$ are errors in the tool navigation parameters, i.e., errors in displacements, velocities, attitude angles, and accelerometer and gyro biases.

When the updated state vector estimate $\delta \underline{x}(+)$ is obtained, it is used to correct the estimates $\hat{\underline{x}}$ (of tool displacement, velocity, attitude) by

$$\hat{\underline{x}}_k(t) = \hat{\underline{x}}_k - \delta \underline{x}_k(+) \quad (A12)$$

In the above procedure, the system state update of Eq A12 effectively resets the error states $\delta \underline{x}(+)$ to zero. Hence, the error state propagation of Eq 6 will remain zero and need not be performed. Thus, Eq 11 reduces to

$$\delta \underline{x}(t) = K_k \underline{Z}_k \quad (A13)$$

and Eq A12 may be written

$$\hat{\underline{x}}_k(+) = \hat{\underline{x}}_k - K_k \underline{Z}_k \quad (A14)$$

Note that these updates occur only when the tool is at rest and error measurements are available. At other times, P and B are merely propagated forward in time.

NOTE: Even though $\underline{Z}_k$ consists only of velocity error measurements (a total of 3), all 15 states ($\underline{x}_k$) are corrected by this procedure. This is a consequence of the gain matrix K_k generally having all nonzero elements. In other words, the measured velocity errors $\underline{Z}_k$ are functionally related to other system states by the nature of the F matrix and therefore the Φ and P matrices. As these matrices propagate forward in time, off diagonal elements of P are developed which are reflected in K through Eq A10.

Noise Covariance Matrices Q_k and R_k

Recall that in the discrete time formulation the state vector $\delta \underline{x}_k$ (at time t_k) is related to the state vector $\delta \underline{x}_{k-1}$ (at previous time t_{k-1}) by

$$\delta \underline{x}_k = \Phi_{k-1} \delta \underline{x}_{k-1} + \underline{\omega}_{k-1} \quad (A15)$$

where $\underline{\omega}_k$ is a zero mean, white noise (or "process" noise) sequence of covariance B_k .

In terms of $G(t)$ $\underline{U}(t)$ of Eq A5, $\underline{\omega}_k$ is a 15 vector given by

$$\underline{\omega}_k = \int_k^{k+1} \Phi(t_k, \tau) G(\tau) \underline{U}(\tau) d\tau \quad (A16)$$

B_k appears as an add-on in the equation for the propagation of P , the estimation error covariance.

$$P_k = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + B_{k-1} \quad (A17)$$

It is shown in Reference 3 that the covariance of $\underline{W}_k$, $E\{\underline{W}_k, \underline{W}_k^T\}$ is given by

$$B_k = \int_{t_{k-1}}^{t_k} \Phi(t_k, \tau) G Q G^T \Phi^T(t_k, \tau) d\tau \quad (A18)$$

where

$$Q = \begin{bmatrix} \alpha_g^2 & & & & \\ & \alpha_v^2 & & & \\ & & \alpha_a^2 & & \\ & & & \alpha_{\dot{a}}^2 & \\ & & & & \alpha_{\ddot{a}}^2 \end{bmatrix} = \text{spectral density matrix of process noise } \underline{\omega}_k$$

The effect of the process noise covariance B_k is to cause the diagonal elements of the covariance matrix, P_k , to grow linearly with time; for example, $\sigma^2 = \alpha^2 t + \sigma_0^2$. The α 's were chosen so that the gyro bias uncertainty (σ) would triple in 1 hour and accelerometer bias uncertainty would double in 1 hour.

$$\alpha_g = 2.925 \times 10^{-8} \text{ rd/s}^{3/2}$$

$$\alpha_a = 1.597 \times 10^{-5} \text{ rd/s}^{5/2}$$

The purpose of including this process noise in the filter is to include the effects of random bias changes in the instruments and to hold the filter gains above some minimum value.

The measurement vector of the system states is given by

$$\underline{Z}_k = H_k \underline{\delta x}_k + \underline{v}_k,$$

where $\underline{v}_k$ is a zero mean, white noise (measurement noise) sequence of spectral density R_k .

$$R_k = \begin{bmatrix} \sigma_{v_e}^2 & & \\ & \sigma_{v_n}^2 & \\ & & \sigma_{v_z}^2 \end{bmatrix},$$

where the σ_v^2 's are the variances associated with making a particular velocity error measurement (when the tool is stopped).

As a typical value,

$$\sigma_{v_e}^2 = \sigma_{v_n}^2 = \sigma_{v_z}^2 = 0.0001 \text{ (ft/s)}^2.$$

This noise includes effects such as accelerometer quantization noise and residual motion present when the system is assumed to be at rest.

Initial Value of the Covariance Matrix, P_k

In the present version of the software, the *initial* system errors are assumed to be uncorrelated, i.e., the off-diagonal elements of P_k are zero. The covariances associated with the initial position are assumed to be 0. This is justified by the fact that although there may be uncertainty in the *absolute* location of the wellhead only the *relative* location of the bore with respect to the wellhead is of interest. Therefore,

$$P_{11} = P_{22} = P_{33} = 0.$$

Similarly for initial velocities, $P_{44} = P_{55} = P_{66} = 0$. The initial attitude error standard deviation σ is taken to be 1 degree, a reasonable value for a manual prealignment of the system. So, $P_{77} = P_{88} = P_{99} = 3 \times 10^{-4} \text{ rd}^2$.

The initial values of $P_{10,10}$ through $P_{15,15}$ are related to the quality of the inertial instruments. Again, bias uncertainties (deviations) of 0.3 degree/hour for the gyros and 15 μg for the accelerometers were assumed. In different units,

$$P_{10,10} = P_{11,11} = P_{12,12} = 2.35 \times 10^{-12} \text{ (rd/s)}^2$$

$$P_{13,13} = P_{14,14} = P_{15,15} = 2.33 \times 10^{-7} \text{ (ft/s)}^2.$$

Changes to Basic Error Model

The foregoing describes the standard filter formulation devised for use in WINS. As mentioned in the text, a change was made just prior to the field test in which the X (vertical) accelerometer bias was dropped as a state, and the X accelerometer scale factor was added in its place.

This required changing the elements of the F matrix relating the rate of change of velocity error in navigation coordinates to the X accelerometer scale factor error (F(4,13), F(5,13), and F(6,13)). These are related by the counts per second of compensated X accelerometer output in navigation coordinates. The three elements are computed in subroutine NAV as the average over the 8-s interval of the compensated X axis velocity change rate in ft/s² divided by the average scale factor used over the interval in ft/s/count, times the appropriate direction cosine. The resulting filter estimate of scale factor error is applied in NAV to adjust the reference scale factor used in the temperature compensation model in subroutine ACOMP.

APPENDIX B

Computer Program Listings

```

C      SUBROUTINE START
C      INITIALIZATION TASK
C      ID=0
C      PRIORITY=0
      EXTERNAL MTAPE
      EXTERNAL SHREC
      EXTERNAL KBD
      EXTERNAL NAVBG
C      INCLUDE COMMON BLOCKS
      INCLUDE "NAV DAT.FR"
      INCLUDE "KALCOM.FR"
      INCLUDE "TSKCOM.FR"
      INCLUDE "CALNUMS.FR"
      DIMENSION CALIN(33)
      EQUIVALENCE (CALIN(1),XAK)
      DATA IDMODE /1/
      DATA NSO/15/,NH/3/,NI/6/
      DATA F,G,H/675*0.0/
      DATA P,B,R,K,A,Q/1350*0.0/
C      DISABLE SPOOLING
      CALL GROUND(1)
      IF(I.EQ.0)CALL SPDIS("$TTO",IER)
      IF(I.EQ.1)CALL SPDIS("$TTO1",IER)
      CALL SPDIS("$LPT",IER)
C      READ KALMAN FILTER VARIABLES
      OPEN 5,"KALIN",ATT="I"
      READ(5,10)(P(I,I),I=1,NSO)
      READ(5,10)(Q(I,I),I=1,NI)
      READ(5,10)(R(I,I),I=1,NH)
      READ (5,11)STIME,TSTEP,IUPDT
10     FORMAT(8F10.0)
11     FORMAT(2F5.0,15)
C      READ CALIBRATION PARAMETERS
      OPEN 4,"CALDAT",ATT="I"
      READ(4,100)CALIN
100    FORMAT(6E12.6)
C      READ POS AND NEG SCALE FACTORS
      READ(4,110)XAKP,XAKN,YAKP,YAKN,ZAKP,ZAKN
110    FORMAT(6F10.8)
      CLOSE 4
C      INITIALIZE CONSTANT PARAMETERS
      G(10,1)=1.
      G(11,2)=1.
      G(12,3)=1.
      G(13,4)=1.
      G(14,5)=1.
      G(15,6)=1.
      H(1,4)=1.
      H(2,5)=1.
      H(3,6)=1.
C      DEFINE OTHER TASKS
      CALL ITASK(SHREC,10,10,IER)
      CALL ITASK(KBD,40,40,IER)
      CALL ITASK(NAVBG,60,60,IER)
      CALL ITASK(MTAPE,55,55,IER)

```

```

C      INITIALIZE WINS INTERFACE
      CALL STWIN(MSGSRC,JIN)
      CALL RESET
C      KILL THIS TASK
      CALL KILL
C
      END

      SUBROUTINE SHREC
C
C      INCLUDE "TSKCOM.FR"
      INCLUDE "BHMDAT.FR"
C
      DIMENSION KTPLAT(40)
      DIMENSION KGNA(3),KANA(3)
      DIMENSION KIN(6),IIN(12)
      DATA JDISK/0/
      DATA KTPLAT/40*0/
      DATA KIN/6*0/
C
C      RESET ACCUMULATORS,ETC.
C
      KAVG=1
      ITIME=0
      RTIME=0.
      DO 10 I=1,3
      IGN(I)=0
      IAN(I)=0
      JAN(I)=0
      JGN(I)=0
      KGNA(I)=0
      KANA(I)=0
      KGN(I)=0
      KAN(I)=0
10     CONTINUE
      JFLAG=0
      KFLAG=0
C
C      WAIT FOR BUFFER FULL MESSAGE
C
20     CALL REC(MSGSRC,IMES)
      IF(IMES.EQ.1)GO TO 40
      CALL AKILL(50)
      WRITE(10,30)IMES
30     FORMAT("<14>INTERRUPT TIMING ERROR ",I6)
40     CONTINUE
C      TRANSFER DATA FROM BUFFER
      DO 45 I=1,12
45     IIN(I)=JIN(I)
      KAD=IAND(IIN(12),17K)+1
      IF(JFLAG.EQ.1)GO TO 750

```

```

IF(KAD.EQ.1)GO TO 700
GO TO 10
700 JFLAG=1
750 CONTINUE
C INCREMENT TIME
220 ITIME =ITIME+1
RTIME=RTIME+.03125
C ASSEMBLE GYRO COUNTS
ITEMP=ISHFT(IIN(2),8)
KIN(1)=IOR(IIN(1),ITEMP)
ITEMP=ISHFT(IIN(4),8)
KIN(3)=-IOR(IIN(3),ITEMP)
ITEMP=ISHFT(IIN(6),8)
KIN(2)=-IOR(IIN(5),ITEMP)
C GET A/D CHANNEL NR AND STORE DATA
IF(KAD.EQ.16)GO TO 330
MUXDAT(KAD)=IIN(11)
330 IF(KAD.NE.9)GO TO 333
KTPLAT(KAVG)=MUXDAT(9)
KAVG=KAVG+1
IF(KAVG.EQ.41)KAVG=1
333 CONTINUE
C IF ITIME IS EVEN,ACCEL DATA IS IN
IF((ITIME.AND.1).EQ.0)GO TO 230
C IF NOT,HOLD LO BYTE TILL NEXT CYCLE
KINLO4=IIN(7)
KINLO5=IIN(8)
KINLO6=IIN(9)
C ALSO HOLD RESOLVER LO BYTE
JRDLO=IIN(10)
GO TO 250
C ASSEMBLE ACCEL COUNTS
230 ITEMP=ISHFT(IIN(7),8)
KIN(4)=IOR(KINLO4,ITEMP)
ITEMP=ISHFT(IIN(8),8)
KIN(5)=IOR(KINLO5,ITEMP)
ITEMP=ISHFT(IIN(9),8)
KIN(6)=IOR(KINLO6,ITEMP)
C ASSEMBLE RESOLVER READING
ITEMP=ISHFT(IIN(10),8)
JRD=IOR(JRDLO,ITEMP)
C
DO 62 I=1,3
62 IAN(I)=IAN(I)+KIN(I+3)
250 DO 60 I=1,3
IGN(I)=IGN(I)+KIN(I)
60 CONTINUE
C
CALL NAV(KIN,INMODE)
C 1 HZ TIME
IF((ITIME.AND.37K).NE.0)GO TO 20
C TRANSFER ACCUMUIATED 1 SEC COUNTS
DO 80 I=1,3
JGN(I)=IGN(I)
80 IGN(I)=0

```

```

JAN(I)=IAN(I)
IAN(I)=0
C ACCUMULATE 1 SEC COUNTS IN MULTI-SEC ACCUMULATORS
  KGNA(I)=KGNA(I)+JGN(I)
  KANA(I)=KANA(I)+JAN(I)
80 CONTINUE
C PLATFORM TEMPERATURE AVERAGE
  JTAVG=0
  DO 336 K=1,40
336 JTAVG=JTAVG+KTPLAT(K)
  MUXDAT(16)=JTAVG
C XFER MULTI SEC COUNT TO DISPLAY VARIABLE
  DO 84 I=1,3
    KGN(I)=KGNA(I)
    KAN(I)=KANA(I)
    KGNA(I)=0
    KANA(I)=0
84 CONTINUE
C IF MAG TAPE NOT REQUIRED SKIP TASK TRIGGER
  IF(MTMODE.EQ.0)GO TO 85
  CALL XMT(MSGMTP,1,85)
85 CONTINUE
C TRIGGER DISPLAY TASK
  CALL XMT(MSGDSP,1,90)
90 CONTINUE
C
C GO TO 20
C
C END

```

```

C SUBROUTINE NAV(JIND,NDMODE)
C
C DIMENSION JIND(6),JIND1(6),ERR(15)
C
C DATA TIAKAV/0.0/
C DATA JIND1 /6*0/
C DATA EN,EV /9.327617E-7,6.542630E-7/
C DATA DVX4S,DVX4SS/2*0.0/
C
C INCLUDE "CALNUMS.FR"
C INCLUDE "BIASCM.FR"
C INCLUDE "KALCOM.FR"
C INCLUDE "TSKCOM.FR"
C INCLUDE "NAVDAT.FR"
C INCLUDE "EHMDAT.FR"
C
C NMODE=NDMODE
C IF(NMODE.NE.0)GOTO 100
C CALL EPTHGEO
C CALL INITNAV
C CALL ERTFNAV

```

```

CALL GRVTV
GOTO 1000
100 CONTINUE
C IF TIME IS EVEN DO VEL UPDATE
C IF((ITIME.AND.1).EQ.0)GOTO 500
C
C ATTITUDE UPDATE PASS
C
CALL GCOMP(JIND,JIND1)
CALL DIRCOS
IF((ITIME.AND.37K).EQ.1)CALL CRTHO
C
GOTO 1000
C
C 500 CONTINUE
C
CALL ACOMP(JIND,MUXDAT)
TXAKAV=TXAKAV+TXAK
DVX4S=DVX4S+DVXI
IF((ITIME.AND.37K).NE.0)GO TO 2000
DVX4SS=DVX4SS+DVX4S
DVX4S=0.0
2000 CONTINUE
CALL ERTHNAV
CALL GRVTV
CALL NEALT
IF((NMODE.EQ.1).AND.((ITIME.AND.37K).EQ.2K))CALL ALIGN
C IF 8 SECS HAVE ELAPSED AND THE FILTER TASK
C HAS COMPLETED AN UPDATE CYCLE, APPLY
C CORRECTIONS TO SYSTEM STATES.
IF((ITIME.AND.377K).NE.366K)GOTO 600
IF(KFLAG.NE.1)GO TO 600
V(1)=VE
V(2)=VN
V(3)=-VD
DO 45 I=1,NSO
45 ERR(I)=0.0
DO 65 J=1,NSO
DO 65 I=1,NE
65 ERR(J)=ERR(J)+K(J,I)*V(I)
DLG=DLG-ERR(2)
DLNG=DLNG-ERR(1)
DALT=DALT-ERR(3)
VN=VN-EPR(5)
VE=VE-ERR(4)
VD=VD+ERR(5)
DTHN=DTHN+ERR(8)
DTHE=DTHE+EPR(7)
DTHD=DTHD-ERR(9)
XGB=XGB-ERR(10)
YGB=YGB-ERR(11)
ZGB=ZGB-ERR(12)
XAKP=XAKP-ERR(13)

```

```

        YAB=YAB+ERR(14)
        ZAB=ZAB+ERR(15)
        KFLAG=0
600    CALL NAVROT
        C
        C
1000    CONTINUE
        C    SAVE GYRO COUNTS
        DO 1010 I=1,3
1010    JIND1(I)=JIND(I)
        C
        C    IF 8 SECS HAVE ELAPSED, SET UP NEW DATA
        C    FOR F MATRIX AND INIT FILTER TASK.
        IF((ITIME.AND.377K).NE.0)GO TO 80
        V(1)=VE
        V(2)=VN
        V(3)=-VD
        FN(1)=0.
        FN(2)=0.
        FN(3)=32.12894
        POS(1)=LNG0+DLNG
        POS(2)=LG0+DLG
        POS(3)=ALT0+DALT
        CNP(1,1)=A21
        CNP(1,2)=A22
        CNP(1,3)=A23
        CNP(2,1)=A11
        CNP(2,2)=A12
        CNP(2,3)=A13
        CNP(3,1)=-A31
        CNP(3,2)=-A32
        CNP(3,3)=-A33
        INCOR=DVX4SS*16.0/TXAKAV
        DVX4SS=0.
        TXAKAV=0.0
        C    ELEMENTS FOR SCALE FACTOR STATE
        XN(1)=INCOR*CNP(1,1)
        XN(2)=INCOR*CNP(2,1)
        XN(3)=INCOR*CNP(3,1)
        CALL XMT(MSGNBG,1,70)
70    CONTINUE
        TIMREF=RTIME
        KFLAG=0
80    RETURN
        END

```

SUBROUTINE INITNAV

ROUTINE FOR INITIALIZATION OF THE NAVIGATION SOFTWARE.

INCLUDE "NAVDAT.FR"

NAVIGATOR INITIAL CONDITIONS

INITIAL POSITION

DATA LG0 /0.6506145/

DATA LNG0 /-2.031515/

DATA ALT0 /6557.0/

INITIALIZE NAVIGATOR

DLG=0.0

DLNG=0.0

DALT=0.0

A11=0.0

A12=0.0

A13=1.0

A21=0.0

A22=1.0

A23=0.0

A31=-1.0

A32=0.0

A33=0.0

VN=VE=VD=0.0

RETURN

END

SUBROUTINE GCOMP(JIN,JIN1)

GYRO INSTRUMENT COMPENSTATION ROUTINE

DIMENSION JIN(6),JIN1(6)

INCLUDE "BIASCM.FR"
INCLUDE "NAVDAT.FR"
INCLUDE "CALNUMS.FR"

REAL K1,K2
DATA K1 /0.083333/
DATA DPHXL,DPHYL,DPHZL/3*0.0/
DATA K2 /0.583333/

GYRO BIAS
XGB,YGB,ZGB

GYRO SCALE FACTORS
XGK,YGK,ZGK

GYRO G SENSITIVITIES
XGAX,XGAY,XGAZ,YGAX,YGAY,YGAZ,ZGAX,ZGAY,ZGAZ

GYRO MISALIGNMENTS
XGY,XGZ,YGX,YGZ,ZGX,ZGY

CONVERT INSTRUMENT COUNTS

DTHX=JIN(1)*XGK
DTHY=JIN(2)*YGK
DTHZ=JIN(3)*ZGK

DTHX1=JIN1(1)*XGK
DTHY1=JIN1(2)*YGK
DTHZ1=JIN1(3)*ZGK

DTHX, DTHY & DTHZ CONTAIN THE GYRO READINGS FROM THE START OF THIS
INTERRUPT. DTHX1, DTHY1 & DTHZ1 ARE THE GYRO READINGS FROM
THE PREVIOUS INTERRUPT WHERE THE ALTERNATE PASS THROUGH THE
NAVIGATOR WAS TAKEN. DTHX2, DTHY2 & DTHZ2 ARE SAVED IN THIS ROUTINE.

DATTX=K1*(DTHY2*DTHZ1-DTHZ2*DTHY1)+K2*(DTHY1*DTHZ-DTHZ1*DTHY)
 DATTY=K1*(DTHZ2*DTHX1-DTHX2*DTHZ1)+K2*(DTHZ1*DTHX-DTHX1*DTHZ)
 DATTZ=K1*(DTHX2*DTHY1-DTHY2*DTHX1)+K2*(DTHX1*DTHY-DTHY1*DTHX)

COMPUTE GYRO BIAS

XGBS=XGB/16+XGAX*DVXI+XGAY*DVYI+XGAZ*DVZI
 YGBS=YGB/16+YGAX*DVXI+YGAY*DVYI+YGAZ*DVZI
 ZGBS=ZGB/16+ZGAX*DVXI+ZGAY*DVYI+ZGAZ*DVZI

COMPUTE ANGLE INCREMENTS

DPHXS=DATTX+DTHX+DTHX1-XGBS
 DPHYS=DATTY+DTHY+DTHY1-YGBS
 DPHZS=DATTZ+DTHZ+DTHZ1-ZGBS

CORRECT FOR GYRO MISALIGNMENTS.

DPHX1=DPHXS-XGZ*DPHYS+XGY*DPHXS
 DPHY1=DPHYS-YGX*DPHXS+YGZ*DPHXS
 DPHZ1=DPHXS-ZGY*DPHXS+ZGX*DPHYS

COMPUTE ALPHA CORRECTION

ALFX=DPHX1-DPHXL
 ALFY=DPHY1-DPHYL
 ALFZ=DPHZ1-DPHZL
 DPHX=DPHX1+.014545*ALFZ
 DPHY=DPHY1-.014545*ALFX
 DPHZ=DPHZ1-.014545*ALFX
 DPHXL=DPEX
 DPHYL=DPHY
 DPHZL=DPHZ

SAVE DTHX, DTHY & DTHZ FOR NEXT COMPUTATION

DTHX2=DTHX
 DTHY2=DTHY
 DTHZ2=DTHZ

RETURN

END

SUBROUTINE DIRCOS

INCLUDE "NAV DAT.FR"

REAL K6,K24,K120,K720
DATA K6/.15666666667/
DATA K24/.04166666667/
DATA K120/.00933333333/
DATA K720/.00138888889/

DPHX2=DPHX*DPHX
DPHY2=DPHY*DPHY
DPHZ2=DPHZ*DPHZ
DPH2=DPHX2+DPHY2+DPHZ2
DPH4=DPH2*DPH2
ALF=-DPH2*K6+DPH4*K120
BET=.5-DPH2*K24+DPH4*K720
WX=DPHX+ALF*DPHX
WY=DPHY+ALF*DPHY
WZ=DPHZ+ALF*DPHZ
DXDY=DPHX*DPHY*BET
DXDZ=DPHX*DPHZ*BET
DYDZ=DPHY*DPHZ*BET
S11=-BET*(DPHZ2+DPHY2)
S22=-BET*(DPHZ2+DPHX2)
S33=-BET*(DPHY2+DPHX2)
S12=-WZ+DXDY
S13=WY+DXDZ
S21=WZ+DXDY
S23=-WX+DYDZ
S31=-WY+DXDZ
S32=WX+DYDZ

A11UP=A11+A11*S11+A12*S21+A13*S31
A12UP=A12+A11*S12+A12*S22+A13*S32
A13UP=A13+A11*S13+A12*S23+A13*S33
A21UP=A21+A21*S11+A22*S21+A23*S31
A22UP=A22+A21*S12+A22*S22+A23*S32
A23UP=A23+A21*S13+A22*S23+A23*S33
A31UP=A31+A31*S11+A32*S21+A33*S31
A32UP=A32+A31*S12+A32*S22+A33*S32
A33UP=A33+A31*S13+A32*S23+A33*S33

A11=A11UP
A12=A12UP
A13=A13UP
A21=A21UP
A22=A22UP

42
A23=A23UP
A31=A31UP
A32=A32UP
A33=A33UP

C
RETURN
END

C
SUBROUTINE ORTHO

C
INCLUDE "NAV DAT.FR"

C
C
COMPUTE A'A-I
ATA11=A11*A11+A21*A21+A31*A31-1.
ATA12=A11*A12+A21*A22+A31*A32
ATA13=A11*A13+A21*A23+A31*A33
ATA21=A12*A11+A22*A21+A32*A31
ATA22=A12*A12+A22*A22+A32*A32-1.
ATA23=A12*A13+A22*A23+A32*A33
ATA31=A13*A11+A23*A21+A33*A31
ATA32=A13*A12+A23*A22+A33*A32
ATA33=A13*A13+A23*A23+A33*A33-1.

C
C
COMPUTE A-0.5*A(A'A-I)
A11UP=A11-0.5*(A11*ATA11+A12*ATA21+A13*ATA31)
A12UP=A12-0.5*(A11*ATA12+A12*ATA22+A13*ATA32)
A13UP=A13-0.5*(A11*ATA13+A12*ATA23+A13*ATA33)
A21UP=A21-0.5*(A21*ATA11+A22*ATA21+A23*ATA31)
A22UP=A22-0.5*(A21*ATA12+A22*ATA22+A23*ATA32)
A23UP=A23-0.5*(A21*ATA13+A22*ATA23+A23*ATA33)
A31UP=A31-0.5*(A31*ATA11+A32*ATA21+A33*ATA31)
A32UP=A32-0.5*(A31*ATA12+A32*ATA22+A33*ATA32)
A33UP=A33-0.5*(A31*ATA13+A32*ATA23+A33*ATA33)

C
C
UPDATE DIRECTION COSINE MATRIX

A11=A11UP
A12=A12UP
A13=A13UP
A21=A21UP
A22=A22UP
A23=A23UP
A31=A31UP
A32=A32UP
A33=A33UP

C
RETURN

C
END

SUBROUTINE ACOMP(JIN1,MUXDAT)

ACCELEROMETER INSTRUMENT COMPENSATION ROUTINE

INCLUDE "BIASCM.FR"
INCLUDE "NAVDAT.FR"
INCLUDE "CALNUMS.FR"

DIMENSION JIN1(6)
DIMENSION MUXDAT(16)

TEMPERATURE COMP CONSTANTS

FOR SYSTEM 1:
DATA TFACTXP/1.84E-4/
DATA TFACTXN/2.23E-4/
DATA TFACTYP/2.01E-4/
DATA TFACTYN/2.01E-4/
DATA TFACTZP/1.94E-4/
DATA TFACTZN/2.28E-4/

DATA TREFX/53.5/
DATA TREFY/53.4/
DATA TREFZ/53.5/

SIZE EFFECT CONSTANT

DATA CONST /4.600736E-1/

ACCELEROMETER OFFSETS

DATA RXX/-.21667/
DATA RXY /-.025/
DATA RXZ /0.0/
DATA RYX /-.09166/
DATA RYY /-.04166/
DATA RYZ /.025/
DATA RZX /0.0/
DATA RZY /.025/
DATA RZZ /-.04166/

ACCELEROMETER MISALIGNMENTS
XAY,XAZ,YAX,YAZ,ZAX,ZAY

ACCELEROMETER SCALE FACTORS
XAK,YAK,ZAK

```

C      ASYMETRIC SCALE FACTORS
C      XAKP,XAKN,YAKP,YAKN,ZAKP,ZAKN
C
C      ACCELEROMETER BIAS
C      KAB,YAB,ZAB
C
C      SCALE FACTOR TEMPERATURE COMP
C
C      TC=MUXDAT(16)*.00976575
C      DELTX=TREFX-TC
C      DELTY=TREFY-TC
C      DELTZ=TREFZ-TC
C      IF(JIN1(4).LT.0)GO TO 10
C      TXAK=XAKP+TFACTXP*XAKP*DELTX
C      GO TO 20
10     TXAK=XAKN+TFACTXN*XAKN*DELTX
20     IF(JIN1(5).LT.0)GO TO 30
C      TYAK=YAKP+TFACTYP*YAKP*DELTY
C      GO TO 40
30     TYAK=YAKN+TFACTYN*YAKN*DELTY
40     IF(JIN1(6).LT.0)GO TO 50
C      TZAK=ZAKP+TFACTZP*ZAKP*DELTZ
C      GO TO 60
50     TZAK=ZAKN+TFACTZN*ZAKN*DELTZ
60     CONTINUE
C
C      ACCELEROMETER BIAS AND SCALE FACTOR COMPUTATION
C
C      DVXIS=TXAK*JIN1(4)-XAB/16
C      DVIIS=TYAK*JIN1(5)-YAB/16
C      DVZIS=TZAK*JIN1(6)-ZAB/16
C
C      *****
C
C      ACCELEROMETER MISALIGNMENT COMPUTATION
C
C      DVXI=DVXIS-XAZ*DVIIS+XAY*DVZIS
C      DVII=DVIIS+YAZ*DVXIS-YAX*DVZIS
C      DVZI=DVZIS-ZAY*DVXIS+ZAX*DVIIS
C
C      *****
C
C      ROTATIONAL CORRECTIONS
C
C      OMGX=DPHX
C      OMGY=DPHY
C      OMGZ=DPHZ
C      OMGX2=OMGX**2
C      OMGY2=OMGY**2
C      OMGZ2=OMGZ**2
C
C      FOP POSITION OFFSETS
C      DVXI=DVXI+(RXI*(OMGY2+OMGZ2)-RXY*OMGX*OMGY-FXZ*OMGX*OMGZ-FXZ*ALFY-FXY*ALFZ)*16.0

```

```

DVYI=DVYI+(RIY*(OMGX2+OMGZ2)-RIZ*OMGY*OMGZ-RYX*OMGX*OMGY-RYX*ALFZ+RYZ*ALFX)*16.0
DVZI=DVZI+(RZZ*(OMGX2+OMGY2)-RZY*OMGY*OMGZ-RZX*OMGX*OMGZ-RZY*ALFX+RZX*ALFY)*16.0
FOR SIZE EFFECT
DVXI=DVXI+ALFZ*CONST
DVYI=DVYI-ALFZ*CONST
DVZI=DVZI-ALFX*CONST
RETURN

```

END

SUBROUTINE ERTHNAV

INCLUDE "NAVDAT.FR"

VELOCITY UPDATE TIME INCREMENT
DATA DTV /0.0525/

EARTH RATE UPDATE TIME INCREMENT
DATA DTE /0.0525/

```

RLATHIV=1.0/(RLAT+DALT+ALT0)
RLNGHIV=1.0/(RLNG+DALT+ALT0)

```

VEDR=VE*RLNGHIV

DTHNN=VEDR+2.0*OMECLT

DTHEN=-VN*RLATHIV

DTHDN=-(VEDR*TNLT+2.*OMESLT)

```

DVNN=(-VD*DTHEN+VE*DTHDN)*DTV
DVEN=(-VN*DTHDN+VD*DTHNN)*DTV
DVDN=(-VE*DTHNN+VN*DTHEN)*DTV

```

```

DTHN=-(DTHNN-OMECLT)*DTE
DTHE=-DTHEN*DTE
DTHD=-(DTHDN+OMESLT)*DTE

```

RETURN
END

SUBROUTINE GRVTV

GRAVITY MODEL ROUTINE

REFERENCE VALUE FOR GRAVITY

DATA G0 /32.0982218/

INCLUDE "NAVDAT.FR"

TIME INTERVAL FOR VELOCITY UPDATES

DATA DTV /0.0525/

GRAVITY MODEL COEFFICIENTS

REAL KSL2,KH1

DATA KSL2 /5.2884E-03/

DATA KH1 /9.623E-08/

GRAVITY MODEL

$GD = G0 * (1.0 + KSL2 * SNLT * SNLT - KH1 * (DALT + ALT0))$

$GD = (GD - .001285) * DTV$

RETURN

END

[illegible]

SECRET

[illegible][illegible]

C C C C C C C C C C

C C C C C C C C C C C C

CCCCCCCC

0000000000

22 22 22 22 22 22

CCCCC

**c
c
c
c**

SUBROUTINE NAVROT

INCLUDE "NAVDAT.FR"

REAL K6,K24,K120,K720
DATA K6/.166666666667/
DATA K24/.041666666667/
DATA K120/.008333333333/
DATA K720/.001398888889/

DPHN2=DTHN*DTHN
DPHE2=DTHE*DTHE
DPHD2=DTHD*DTHD
DPH2=DPHN2+DPHE2+DPHD2
DPH4=DPH2*DPH2
ALF=-DPH2*K6+DPH4*K120
BET=.5-DPH2*K24+DPH4*K720
WN=DTHN+ALF*DTHN
WE=DTHE+ALF*DTHE
WD=DTHD+ALF*DTHD
DNDE=DTHN*DTHE*BET
DNDD=DTHN*DTHD*BET
DEDD=DTHE*DTHD*BET
S11=-BET*(DPHE2+DPHD2)
S22=-BET*(DPHN2+DPHD2)
S33=-BET*(DPHN2+DPHE2)
S12=-WD+DNDE
S13=WE+DNDD
S21=WD+DNDE
S23=-WN+DEDD
S31=-WE+DNDD
S32=WN+DEDD

A11UP=A11+S11*A11+S12*A21+S13*A31
A12UP=A12+S11*A12+S12*A22+S13*A32
A13UP=A13+S11*A13+S12*A23+S13*A33
A21UP=A21+S21*A11+S22*A21+S23*A31
A22UP=A22+S21*A12+S22*A22+S23*A32
A23UP=A23+S21*A13+S22*A23+S23*A33
A31UP=A31+S31*A11+S32*A21+S33*A31
A32UP=A32+S31*A12+S32*A22+S33*A32
A33UP=A33+S31*A13+S32*A23+S33*A33

A11=A11UP
A12=A12UP
A13=A13UP
A21=A21UP
A22=A22UP

A23=A23UP
A31=A31UP
A32=A32UP
A33=A33UP

RETURN
END

SUBROUTINE ERTHGEO

INCLUDE "NAV DAT.FR"

PHYSICAL PARAMETERS

DATA OME /7.2921151E-05/
DATA RE /20925722./
DATA EL /0.00335289187/

SNLT=SIN(DLG+LG0)
CSLT=COS(DLG+LG0)
SCLT=1.0/CSLT
TNLT=SNLT/CSLT

OMESLT=OME*SNLT
OMECLT=OME*CSLT
RLAT=RE*(1.0-2.0*EL+3.0*EL*SNLT*SNLT)
RLNG=RE*(1.0+EL*SNLT*SNLT)

RETURN

END

SUPERROUTINE ALIGN

AUTO LEVELING ALIGNMENT ROUTINE

INCLUDE "NAVDAT.FR"

ALIGNMENT GAINS

VELOCITY FILTER GAIN
DATA ALGN1 /0.363/

ATTITUDE CORRECTION GAIN
DATA ALGN2 /0.001020/

HOLD INITIAL POSITION

DLG=0.0
DLNG=0.0
DALT=0.0

ATTITUDE UPDATE
DTHN=DTHN-VE*ALGN2
DTHE=DTHE+VN*ALGN2

VELOCITY UPDATE
VN=VN-VN*ALGN1
VE=VE-VE*ALGN1
VD=0.

RETURN

END

```

SUBROUTINE KBD
C
C
EXTERNAL DSPLY
INCLUDE "TSKCOM.FR"
DATA MTNAME /2HMT,2H0:,2H00,0,0/
C
10  CONTINUE
WRITE(10,20)
20  FORMAT("<14>")
CALL ITASK(DSPLY,50,50,IER)
C
30  CALL GCHAR(ICCHAR,IER)
C
CALL AKILL(50)
C
IF(ICCHAR.EQ."<0>F")GOTO 30
WRITE(10,40)
40  FORMAT("<14>")
IF(ICCHAR.EQ."<0>I")GOTO 100
IF(ICCHAR.EQ."<0>M")GOTO 200
IF(ICCHAR.EQ."<0>S")GOTO 300
IF(ICCHAR.EQ."<0>T")GOTO 400
IF(ICCHAR.EQ."<0>N")GOTO 500
IF(ICCHAR.EQ."<0>C")GO TO 600
GOTO 10
C
C
100 CONTINUE
WRITE(10,110)
110 FORMAT("ENTER DISPLAY MODE ",Z)
CALL GRSVP(IDMODE,1)
GOTO 10
C
200 CONTINUE
WRITE(10,210)
210 FORMAT("ENTER NAV MODE. (0=INIT 1=ALIGN 2=NAV 3=FIX) ",Z)
CALL GRSVP(INMODE,1)
GOTO 10
C
300 CONTINUE
CALL FINRV(55K)
CALL SPEBL("$LPT",IER)
CALL GROUND(1)
IF(I.EQ.0)CALL SPEBL("$TTO",IER)
IF(I.EQ.1)CALL SPEBL("$TTO1",IER)
STOP
C
400 CONTINUE
WRITE(10,410)
410 FORMAT(1X,"MAG TAPE CONTROL:",/,
X      1X,1 - STOP",/,
X      1X,2 - CONTINUE",/,
X      1X,3 - ADD TO OLD TAPE",/,
X      1X,4 - START NEW TAPE ",Z)

```

```

C      CALL GRSVP(1,1)                                ; GET CHOICE
      IF(1.EQ.0.OR.1.GT.4)GO TO 400
      IF(1.GT.1.AND.MTMODE.NE.0)GO TO 480
      GO TO(420,430,440,440), 1

C      ***** STOP MAG TAPE *****
C
C      IF(MTMODE.EQ.0)GO TO 10
      MTMODE=0
      CALL WAIT(2,2,IER)
      ENDFILE 0
      GO TO 10

C      ***** CONTINUE MAG TAPE *****
C
C      MTNAME(3)=MTNAME(3)+1
      IF(IAND(MTNAME(3),377K).GT.71K)MTNAME(3)=IAND(MTNAME(3),177400K)-4
X      60K
      CALL MTSTAT(0,ISTAT)
      IF(ITEST(ISTAT,0).EQ.0)GO TO 470
      OPEN 0,MTNAME,ATT="S",ERR=490
      MTMODE=2
      GO TO 10

C      ***** ADD TO OLD MAG TAPE OR START NEW TAPE *****
C
C      CALL MTSTAT(0,ISTAT)
      IF(ITEST(ISTAT,0).EQ.0)GO TO 470
      CALL INIT("MT0",0,IER)
      IF(1.EQ.3)GO TO 455
      IF(1.EQ.3)GO TO 455
      OPEN 0,MTNAME,ATT="S",ERR=490
      CALL MTSTAT(0,ISTAT)
      IF(ITEST(ISTAT,0).EQ.0)GO TO 450
      MTMODE=1
      GO TO 10
; WAIT FOR MAG TAPE TO REWIND
; NEW TAPE STARTED

C      OPEN 0,MTNAME,ATT="S",ERR=460
      READ BINARY (0,END=460) JTEMP
      CLOSE 0
      IFILE=IFILE+1
      IFTENS=IFILE/10
      IFONES=IFILE-IFTENS*10
      MTNAME(3)=IFONES+60K+ISHFT(IFTENS+60K,8)
      GO TO 455
      CLOSE 0
      IFILE=IFILE-1
      IFTENS=IFILE/10
      IFONES=IFILE-IFTENS*10
      MTNAME(3)=IFONES+60K+ISHFT(IFTENS+60K,8)
      GO TO 10

```

```

C
C
C
470 ***** MAG TAPE NOT READY MESSAGE *****
CONTINUE
TYPE "MAG TAPE NOT READY !!!"
CALL WAIT(4,2,IER)
GO TO 10

C
C
C
480 ***** MAG TAPE RUNNING BEFORE START REQUEST *****
CONTINUE
TYPE "MAG TAPE ALREADY RUNNING"
CALL WAIT(2,2,IER)
GO TO 10

C
C
C
490 ***** MAG TAPE ERROR *****
CONTINUE
TYPE "MAG TAPE ERROR = ",IER
CALL WAIT(4,2,IER)
GO TO 10

C
C
C
500 CONTINUE
WRITE(10,505)
505 FORMAT(1X,"TEST NAME (32 CHARACTERS OR LESS)
X      = ",2)
510 READ(11,510) JNAME(1)
FORMAT(S33)
IND=0
DO 550 I = 1,16
IF(IND.EQ.1)GO TO 540
IF(IAND(JNAME(I),177400K).EQ.0)GO TO 530
IF(IAND(JNAME(I),377K).EQ.0)GO TO 520
GO TO 550
520 IND=1
JNAME(I)=JNAME(I)+40K
GO TO 550
530 IND=1
540 JNAME(I)=20040K
550 CONTINUE
GO TO 10

C
600 CONTINUE
JMODE =1
WRITE(10,602)
602 FORMAT("ENTER COMMAND NUMBER: 0=RUN,1=STDBY,2=TEST,"/
1 "3=CCW SLOW,4=CCW FAST,5=CCW SLOW,6=CCW FAST,"/
2 "7=XFER TO MON,8=ENABLE INTERFACE,9=DISABLE,"/
3 "10=CAGE,11=INITIATE WINS")
ACCEPT ICMD
IF (ICMD.GT.11)GO TO 600
IF (ICMD.LT.8)JMODE=2
IF (ICMD.EQ.10)GO TO 650
IF(ICMD.EQ.11)GO TO 690

```

```

CALL WINCMD(ICMD,JMODE)
GO TO 700
650 ACCLPT "ENTER INTEGER CAGE POSITION",JCAGE
    NHICAG=BYTE(JCAGE,1)
    NLOCAG=BYTE(JCAGE,2)
    CALL WINCAG(NHICAG,NLOCAG)
GO TO 700
690 CONTINUE
    CALL WINSTRT
700 CONTINUE
    GO TO 10

```

C
C

END

SUBROUTINE DSPLY

C
C
C
C

DISPLAY DRIVER TASK
ID=50 PRIORITY=50

INCLUDE "TSKCOM.FR"
INCLUDE "BHMDAT.FR"
DATA LASMODE,MTIME,TSTART /0.0.0./

C

MODE=INMODE
WRITE(10,10)
10 FORMAT("<14>"//)
C

IF(LASMODE.EQ.MODE)GO TO 30
LASMODE=MODE
TSTART=RTIME
MTIME=0

30

C

CONTINUE
DISPLAY MODE 3 FREEZES THE NAV DATA DSPLY

C

DISPLAY MODE 4 FREEZES THE FILTER DATA DISPLAY

IF((IDMODE.EQ.1).OR.(IDMODE.EQ.3))CALL NAVDIS(0,MODE,MTIME,JIN,JRD,MUXDAT)

C

IF((IDMODE.EQ.2).OR.(IDMODE.EQ.4))CALL KALDIS(0)

20

C

CALL REC(MSGDSP,IMSG)

MTIME=RTIME-TSTART
IF((IDMODE.EQ.3).OR.(IDMODE.EQ.4))GO TO 20
IF(IDMODE.EQ.1)CALL NAVDIS(1,MODE,MTIME,JIN,JRD,MUXDAT)

C

IF (IDMODE.EQ.2)CALL KALDIS(1)
GOTO 20

C

END

SUBROUTINE NAVDIS(IDM,MODE,MTIME,JIN,JRD,MUXDAT)

ROUTINE FOR GENERATION OF THE NAVIGATOR DATA DISPLAY

DOUBLE PRECISION PHI,THE,PSI

INCLUDE "BRMDAT.FR"

INCLUDE "NAVDAT.FR"

INCLUDE "BIASCM.FR"

INCLUDE "CALNUMS.FR"

DIMENSION DELP(3),JIN(12),MUXDAT(16)

DIMENSION JSTAT(3),NAMSTAT(3,9)

DATA NAMSTAT

1 /2HST,2HND,2HFY,

1 2HST,2HRT,2HUP,

1 2HIN,2HER,2HTL,

1 2HCA,2HGE,2H

1 2HCW,2HFA,2HST,

1 2HCW,2HSL,2HOW,

1 2HCC,2HWS,2HLO,

1 2HCC,2HWF,2HAS,

1 2HTE,2HST,2H /

TCON=.39063

RCON=.78125

VCON1=.15625

VCON2=.04688

TCON40=TCON/40.

IF(IDM.NE.0)GOTO 200

WRITE HEADINGS

WRITE(10,10)

FORMAT(

1 19X"X"9X"Y"9X"Z"/

1 "GYROS"/

1 "ACCEL"/

1 40X"ROLL RATE"/

1 "DEL N"35X"+29V"/

1 "DEL E"35X"-29V"/

1 "DEL H"35X"9.5V"/

1 "VN"38X"T XIG"/

1 "VE"38X"T XRG"/

1 "VD"38X"T XYGAF"/

1 "PHI"37X"T XRGAF"/

1 "THE"37X"T XACAF"/

1 "PSI"37X"XAKP"/

1 "GYRO BIAS EST"/

1 "ACCEL BIAS EST"/

1 "RESOLVER"32X"T PLAT"/

1 "STATUS"34X"T EXT"/

1 "MODE"36X"T R/D"/

1 "MODE TIME"31X"T BATT"/

1 40X"T PWRCON"/

1 "TIME")

```

C
C
200 CONTINUE
C
  NSTAT=ISHFT(JIN(12),-4)
  DO 5 I=1,3
5    JSTAT(I)=NAMSTAT(I,NSTAT+1)
    IF((A32.NE.0.0).OR.(A31.NE.0.0))
1    PHI=ATAN2(A32,-A31)*57.2958
    IF(ABS(A33).LE.1.0)THE=-ASIN(A33)*57.2958
    IF((A23.NE.0.0).OR.(A13.NE.0.0))
1    PSI=ATAN2(A23,A13)*57.2958
C
  DELP(1)=DLG*RLAT
  DELP(2)=DLNG*RLNG*CSLT
  DELP(3)=DALT
  DO 50 I=2,7
50  ANDAT(I)=MUXDAT(I)*TCON
C  ANDAT(9)-(TPLAT)-IS IN MUXDAT(16)-UPDATED EACH SEC
  ANDAT(9)=MUXDAT(16)*TCON40
  ANDAT(10)=MUXDAT(10)*TCON
  ANDAT(12)=MUXDAT(12)*TCON
  ANDAT(13)=MUXDAT(13)*TCON
  ANDAT(15)=MUXDAT(15)*TCON
  ANDAT(1)=MUXDAT(1)*RCON-100.0
  ANDAT(8)=-MUXDAT(8)*VCON1
  ANDAT(11)=MUXDAT(11)*VCON1
  ANDAT(14)=MUXDAT(14)*VCON2
C
  WRITE(10,210)JGN
210  FORMAT(3H<17><20><103>,3(4X,I6))
  WRITE(10,220)JAN
220  FORMAT(3H<17><20><104>,3(4X,I6))
  WRITE(10,230)DELP
230  FORMAT(3H<17><21><106>F9.2,3H<17><21><107>F9.2,
1    3H<17><21><110>F9.2)
  WRITE(10,240)VN,VE,VD
240  FORMAT(3H<17><22><111>F8.4,3H<17><22><112>F8.4,
1    3H<17><22><113>F8.4)
  WRITE(10,250)PHI,THE,PSI
250  FORMAT(3H<17><21><114>F9.4,3H<17><21><115>F9.4,
1    3H<17><21><116>F9.4)
  WRITE(10,251)XGB,YGB,ZGB
251  FORMAT(3H<17><20><117>3(1X,E9.3))
  WRITE(10,252)XAB,YAB,ZAB
252  FORMAT(3H<17><20><120>3(1X,E9.3))
  WRITE(10,254)JPD
254  FORMAT(3H<17><23><121>I6)
  WRITE(10,253)JSTAT
253  FORMAT(3H<17><23><122>3A2)
  WRITE(10,255)MODE
255  FORMAT(3H<17><23><123>I4)
  WRITE(10,256)MTIME
256  FORMAT(3H<17><23><124>I4)
  WRITE(10,260)ETIME
260  FORMAT(3H<17><23><126>F7.2)

```

```

C
WRITE(10,100)ANDAT(1),ANDAT(11)
100 FORMAT(3H<17><124><105>F6.2,3H<17><124><106>F6.2)
WRITE(10,105)ANDAT(8),ANDAT(14)
105 FORMAT(3H<17><124><107>F6.2,3H<17><124><110>F6.2)
WRITE(10,110)ANDAT(2),ANDAT(3)
110 FORMAT(3H<17><124><111>F6.2,3H<17><124><112>F6.2)
WRITE(10,115)ANDAT(4),ANDAT(5)
115 FORMAT(3H<17><124><113>F6.2,3H<17><124><114>F6.2)
WRITE(10,120)ANDAT(6),XAKP
120 FORMAT(3H<17><124><115>F6.2,3H<17><120><116>E12.6)
WRITE(10,125)ANDAT(9),ANDAT(13)
125 FORMAT(3H<17><124><121>F6.2,3H<17><124><122>F6.2)
WRITE(10,130)ANDAT(15),ANDAT(10)
130 FORMAT(3H<17><124><123>F6.2,3H<17><124><124>F6.2)
WRITE(10,135)ANDAT(12)
135 FORMAT(3H<17><124><125>F6.2)
WRITE(10,300)FLAGSTAT
300 FORMAT(3H<17><140><117>I1)
400 RETURN
C
END
C
SUBROUTINE KALDIS(IDM)
ROUTINE FOR GENERATION OF FILTER DATA DISPLAY
INCLUDE "BHMDAT.FR"
INCLUDE "NAV DAT.FR"
DIMENSION LINE(19)
DIMENSION SIG(20)
DATA LINE/2H D,2H E,2H F,2H G,2H H,2H I,2H J,2H K,2H L,2H M,
1 2H N,2H O,2H P,2H Q,2H R,2H S,2H T,2H U,2H V/
IF(IDM.NE.0)GO TO 800
WRITE(10,710)
710 FORMAT("SQRT P(N,N):",27X,"K MATRIX:"// "LNG"// "LAT"// "ALT"// "VE"// "VN"
1 // "VU"// "THE"// "THN"// "THU"// "XGB"// "YGP"// "ZGB"// "XAF"// "YAF"// "ZAF"
1 //20X,"TIME")
800 CONTINUE
DO 810 I=1,15
810 SIG(I)=SQRT(ABS(P(I,I)))
DO 900 I=1,3
900 WRITE(10,950)LINE(I),SIG(I)
950 FORMAT(2H<17><020>,R1,E9.3,Z)
WRITE(10,920)LINE(4)
920 FORMAT(2H<17><020>,R1,1X)
DO 910 I=5,10
910 WRITE(10,950)LINE(I),SIG(I-1)
WRITE(10,920)LINE(11)
DO 940 I=12,17
940 WRITE(10,950)LINE(I),SIG(I-2)
WRITE(10,920)LINE(18)
DO 980 I=1,3
980 WRITE(10,990)LINE(I),K(I,1),K(I,2),K(I,3)
DO 992 I=4,9
982 WRITE(10,990)LINE(I+1),K(I,1),K(I,2),K(I,3)
DO 984 I=10,15
984 WRITE(10,990)LINE(I+2),K(I,1),K(I,2),K(I,3)
990 FORMAT(2H<17><100>,R1,3(3X,E9.3))

```

850 WRITE(10,850)RTIME
 FORMAT(3H<17><060><126>,F7.2)

RETURN
 END

SUBROUTINE MTAPE

ROUTINE FOR WRITING DATA TO MAG TAPE

INCLUDE "TSKCOM.FR"
 INCLUDE "BHMDAT.FR"
 INCLUDE "KALCOM.FR"
 INCLUDE "BIASCH.FR"
 INCLUDE "NAVDAT.FR"
 INCLUDE "CALNUMS.FR"
 DIMENSION JOUT(1)
 EQUIVALENCE (JOUT(1),NMODE)
 DATA TCON40/.009766/

CALL REC(MSGMTP,KMSG)

CALL DATE(JDATE,IER)
 CALL TIME(JTIME,IER)

TPLAT=MUXDAT(16)*TCON40
 WRITE BINARY(0,ERR=10)JDATE,JTIME,JNAME,RTIME,KGN,KAN,
 1 (ANDAT(1),I=2,7),
 1 (P(I,I),I=1,15),((K(I,J),I=1,15),J=1,3),
 1 (JOUT(1),I=1,73),TPLAT,XGB,YGB,ZGB,XAB,YAB,ZAB,
 1 ANDAT(13),JRD,KFLAG,XAKP

GO TO 5

CONTINUE
 TYPE "MAG TAPE ERROR"
 STOP

END

SUBROUTINE NAVPG

```

C
C
INCLUDE "KALCOM.FR"
INCLUDE "NAVDAT.FR"
INCLUDE "BHMDAT.FR"
INCLUDE "TSKCOM.FR"
C
10 CALL REC(MSGNBG,I)
C
KINMODE=INMCDE
TSAVE=TIMREF
CALL ERTHGEO
GE=GD/.0625
C
IF(KINMODE.EQ.0)GO TO 40
IF (KINMODE.EQ.3)GO TO 41
CALL NAVEM(itime,f,g,h,NSO,NI,NE,V,FN,POS,CNP,GE,XN)
CALL DISMA(STIME,F,A,NSO,TSTEP)
CALL DISMB(STIME,F,G,Q,B,NSO,NI,TSTEP)
41 CONTINUE
IF(KINMODE.EQ.1)GO TO 40
CALL KALMAN(KINMODE,IUPDT,P,A,B,H,R,K,NSO,NE)
IF(KINMODE.EQ.2)GO TO 40
KFLAG=1
40 CONTINUE
C
FLAGSTAT=1
IF((RTIME-TSAVE).GT.7.5)FLAGSTAT=0
GOTO 10
C
END

```

SUBROUTINE NAVEM(T,F,G,H,NSO,NI,NH,V,FN,POS,CNP,G0,XN)

DIMENSION XN(3)

REAL F(15,15),G(15,15),H(15,15)

REAL V(3),FN(3),POS(3),CNP(3,3)

DATA R/20925722./

DATA OM/7.2921151E-5/

SL=SIN(POS(2))

CL=COS(POS(2))

TANL=SL/CL

RHE=-V(2)/R

RHN=V(1)/R

RHZ=(V(1)*TANL)/R

OMN=OM*CL

OMZ=OM*SL

WE=RHE

WN=RHN+OMN

WZ=RHZ+OMZ

F(1,2)=RHZ/CL

F(1,3)=-RHN/(R*CL)

F(1,4)=1./(R*CL)

F(2,3)=RHE/R

F(2,5)=1./R

F(3,6)=1.0

F(4,2)=2.*(OMN*V(2)+OMZ*V(3))+RHN*V(2)/(CL*CL)

F(4,3)=RHZ*RHE+RHN*V(3)/R

F(4,4)=-RHE*TANL-V(3)/R

F(4,5)=WZ+OMZ

F(4,6)=-WN-OMN

F(4,8)=-FN(3)

F(4,9)=FN(2)

F(5,2)=-2.*OMN*V(1)-RHN*V(1)/(CL*CL)

F(5,3)=RHN*RHZ-RHE*V(3)/R

F(5,4)=-2.*WZ

F(5,5)=-V(3)/R

F(5,6)=RHE

F(5,7)=FN(3)

F(5,9)=-FN(1)

F(6,2)=-2.*OMZ*V(1)

F(6,3)=2.*G0/R-RHN*RHN-RHE*RHE

F(6,4)=2.*WN

F(6,5)=-2.*RHE

F(6,7)=-FN(2)

F(6,8)=FN(1)

F(7,3)=-RHE/R

F(7,5)=-1./R

F(7,8)=WZ

F(7,9)=-WN

F(8,2)=-OMZ

F(8,3)=-RHN/R

F(8,4)=1./R

F(8,7)=-WZ

F(8,9)=WE

F(9,2)=WN+RHZ*TANL

F(9,3)=-RHZ/R

F(9,4)=TANL/R

```

F(9,7)=WN
F(9,8)=-WE
DO 210 I=1,3
DO 210 J=1,3
II=I+3
JJ=J+12
F(II,JJ)=CNP(I,J)
II=I+6
JJ=J+9
F(II,JJ)=CNP(I,J)
210  CONTINUE
      DO 220 J=1,3
      JJ=J+3
220  F(JJ,13)=XN(J)
      RETURN
      END

```

```

SUBROUTINE DISMA(T,FIN,AIN,NIN,TSTEP)
REAL FIN(15,15),AIN(15,15)
COMMON /DXCOM/ N,M,F(15,15),GQG(15,15)
1, TMP1(15,15),TMP2(15,15),TMP3(15,15)
EXTERNAL DXA
DIMENSION X(15)
DATA RE/1.0E-6/,AE/1.0E-20/
N=NIN
DO 5 I=1,N
DO 5 J=1,N
5  F(I,J)=FIN(I,J)
DO 10 J=1,N
DO 20 I=1,N
20 X(I)=0.0
X(J)=1.0
IER=1
TIME=0.0
IFLAG=1
CALL RKINT(DXA,TIME,T,X,N,TSTEP,IFLAG)
DO 40 I=1,N
40 AIN(I,J)=X(I)
10 CONTINUE
RETURN
END

```

```

SUBROUTINE DISMB(T,FIN,GIN,QIN,BIN,NIN,MIN,TSTEP)
REAL FIN(15,15),GIN(15,15),QIN(15,15),BIN(15,15)
COMMON /DXCOM/ N,M,F(15,15),GQG(15,15)
1,TMP1(15,15),TMP2(15,15),TMP3(15,15)
DIMENSION X(210)
EXTERNAL DXB
DATA RE/1.0E-6/,AE/1.0E-20/
N=NIN
M=MIN
DO 10 I=1,N
DO 10 J=1,N
10 F(I,J)=FIN(I,J)
CALL MULT(GIN,N,M,QIN,M,M,TMP1,N,M,1)
CALL MULT(TMP1,N,M,GIN,N,M,GQG,N,N,3)
JJ=0
DO 20 I=1,N
DO 20 J=1,I
20 JJ=JJ+1
X(JJ)=0.
NN=((N+1)*N)/2
IER=1
TIME=0.0
IFLAG=1
CALL RKINT(DXB,TIME,T,X,NN,TSTEP,IFLAG)
JJ=0
DO 30 I=1,N
DO 30 J=1,I
30 JJ=JJ+1
BIN(I,J)=X(JJ)
BIN(J,I)=BIN(I,J)
RETURN
END

```



```

SUBROUTINE RKINT(DERFUN,T,TOUT,Y,NSYS,TSTEP,IFLAG)
EXTERNAL DERFUN
DOUBLE PRECISION YU
DIMENSION Y(1)
COMMON /RCOM/YD(210),YU(210),DELY(4,210)
DIMENSION BET(3)
DATA BET /0.5,0.5,1.0/
IF(IFLAG.NE.1)GOTO 20
IFLAG=2
DO 10 I=1,NSYS
10 YU(I)=Y(I)
20 CONTINUE
TOLD=T
TS=AMIN1(TSTEP,TOUT-T)
DO 50 K=1,4
CALL DERFUN(T,Y,YD)
DO 30 I=1,NSYS
30 DELY(K,I)=YD(I)*TS
IF(K.EQ.4)GOTO 50
DO 40 I=1,NSYS
40 Y(I)=YU(I)+BET(K)*DELY(K,I)
T=TOLD+BET(K)*TS
50 CONTINUE
DO 60 I=1,NSYS
YU(I)=YU(I)+(DBLE(DELY(1,I))+2.0D0*DELY(2,I))
1 +2.0D0*DBLE(DELY(3,I))+DBLE(DELY(4,I)))/6.0D0
60 Y(I)=YU(I)
T=TOLD+TS
IF(T.LT.TOUT)GOTO 20
RETURN
END

```

```

SUBROUTINE DXA(TIME,X,DX)
COMMON /DXCOM/ N,M,F(15,15),GQG(15,15)
1,TMP1(15,15),TMP2(15,15),TMP3(15,15)
DIMENSION X(15),DX(15)
DO 10 I=1,N
DX(I)=0.0
DO 10 J=1,N
10 DX(I)=DX(I)+F(I,J)*X(J)
RETURN
END

```

```

SUBROUTINE DXB(TIME,X,DX)
COMMON /DXCOM/ N,M,F(15,15),GQG(15,15)
1,TMP1(15,15),TMP2(15,15),TMP3(15,15)
DIMENSION X(210),DX(210)
JJ=0
DO 1 I=1,N
DO 1 J=1,I
JJ=JJ+1
TMP1(I,J)=X(JJ)
1 TMP1(J,I)=TMP1(I,J)
CALL MULT(F,N,N,TMP1,N,N,TMP2,N,N,1)
CALL MULT(TMP1,N,N,F,N,N,TMP3,N,N,3)
DO 2 I=1,N
DO 2 J=1,N
2 TMP2(I,J)=TMP2(I,J)+TMP3(I,J)+GQG(I,J)
JJ=0
DO 3 I=1,N
DO 3 J=1,I
JJ=JJ+1
3 DX(JJ)=TMP2(I,J)
RETURN
END

```

```

SUBROUTINE KALMAN(MSG,IUPDT,P,STM,B,E,R,K,N,NH)
COMMON /DXCOM/ NDUM,MDUM,F(15,15),GQG(15,15),
1 TMP1(15,15),TMP2(15,15),TMP3(15,15)
REAL P(15,15),STM(15,15),B(15,15),R(15,15),K(15,15),E(15,15)

```

```

C
C   COMPUTE COV MATRIX,PRE MEAS,P=OPO*+P
CALL MULT(STM,N,N,P,N,N,TMP1,N,N,1)
CALL MULT(TMP1,N,N,STM,N,N,P,N,N,3)
DO 1 I=1,N
DO 1 J=1,N
1  P(I,J)=P(I,J)+B(I,J)
   IF(MSG.EQ.3)GO TO 7
RETURN
7 CONTINUE

C
C   COMPUTE KALMAN GAIN MATRIX,K=PH*/(HPE*+R)
CALL MULT(H,NH,N,P,N,N,TMP1,NH,N,1)
CALL MULT(TMP1,NH,N,H,NH,N,TMP2,NH,NH,3)
DO 2 I=1,NH
DO 2 J=1,NH
2  TMP2(I,J)=TMP2(I,J)+R(I,J)
CALL INV(TMP2,NH)
CALL MULT(H,NH,N,TMP2,NH,NH,TMP1,N,NH,2)
CALL MULT(P,N,N,TMP1,N,NH,K,N,NH,1)

C
C   COMPUTE COV MATRIX,POST MEAS,P=(I-KH)P(I-KH)*+KPK*
CALL MULT(K,N,NH,H,NH,N,TMP1,N,N,1)
DO 3 I=1,N
DO 3 J=1,N
3  TMP1(I,J)=-TMP1(I,J)
DO 5 I=1,N
5  TMP1(I,I)=1.+TMP1(I,I)
CALL MULT(TMP1,N,N,P,N,N,TMP2,N,N,1)
CALL MULT(TMP2,N,N,TMP1,N,N,P,N,N,3)
CALL MULT(K,N,NH,R,NH,NH,TMP1,N,NH,1)
CALL MULT(TMP1,N,NH,K,N,NH,TMP2,N,N,3)
DO 4 I=1,N
DO 4 J=1,N
4  P(I,J)=P(I,J)+TMP2(I,J)
RETURN
END

```

```

SUBROUTINE MULT(A,NAR,NAC,B,NBR,NBC,C,NCR,NCC,NOP)
DIMENSION A(15,15),B(15,15),C(15,15)
DOUBLE PRECISION SUM
C   NOP=1 C=A
C   NOP=2 C=A*B
C   NOP=3 C=A*B*
IF(NOP.EQ.2)GOTO 2
IF(NOP.EQ.3)GOTO 3
DO 4 I=1,NAR
DO 4 J=1,NBC
SUM=0.0D0
DO 4 K=1,NAC
SUM=SUM+DBLE(A(I,K))*DBLE(B(K,J))
4  C(I,J)=SUM
RETURN
2 CONTINUE
DO 6 I=1,NAC
DO 6 J=1,NBC
SUM=0.0D0
DO 6 K=1,NAR
SUM=SUM+DBLE(A(K,I))*DBLE(B(K,J))
6  C(I,J)=SUM
RETURN
3 CONTINUE
DO 8 I=1,NAR
DO 8 J=1,NBR
SUM=0.0D0
DO 8 K=1,NAC
SUM=SUM+DBLE(A(I,K))*DBLE(B(J,K))
8  C(I,J)=SUM
RETURN
END

```

```

SUBROUTINE INV(X,N)
DIMENSION X(15,15)
DIMENSION C(3,3)
C(1,1)=X(2,2)*X(3,3)-X(3,2)*X(2,3)
C(1,2)=-X(2,1)*X(3,3)+X(3,1)*X(2,3)
C(1,3)=X(2,1)*X(3,2)-X(3,1)*X(2,2)
C(2,1)=-X(1,2)*X(3,3)+X(3,2)*X(1,3)
C(2,2)=X(1,1)*X(3,3)-X(3,1)*X(1,3)
C(2,3)=-X(1,1)*X(3,2)+X(3,1)*X(1,2)
C(3,1)=X(1,2)*X(2,3)-X(2,2)*X(1,3)
C(3,2)=-X(1,1)*X(2,3)+X(2,1)*X(1,3)
C(3,3)=X(1,1)*X(2,2)-X(2,1)*X(1,2)
D=0.
DO 10 I=1,3
10  D=D+C(1,I)*X(1,I)
DO 20 I=1,3
DO 20 J=1,3
20  X(I,J)=C(J,I)/D
RETURN

```

C
C
C
C
C
C
C
C
C
C

SUBROUTINE GRSVP.EB - SUBROUTINE TO REQUEST AND GET AN INTEGER -

CALLING SEQUENCE:

CALL GRSVP(I,J)

WHERE: I = THE RETURNED INTEGER IN THE RANGE OF 0 TO 9

J = 0 TO PASS BACK ANY CHARACTER

0 TO PASS BACK ONLY DECODED NUMERIC DIGIT

```

SUBROUTINE GRSVP(I,J)
10  WRITE(10,20)                                ; TYPE A BLINKING "?"
20  FORMAT("?",2)
    CALL GCHAR(I,IERR)
    K = I * 400K
    WRITE(10,30) K                                ; ECHO THE INTEGER
30  FORMAT("<10>".A1,Z)
    IF (J .EQ. 0) GOTO 50
    I = I - 60K
    IF (I .GE. 0 .AND. I .LE. 9) GOTO 50
    WRITE(10,40)
40  FORMAT("<10>".Z)
    GOTO 10
50  WRITE(10,51)
51  FORMAT(1X)
    RETURN
    END

```

TITLE MTSTAT

; MTSTAT.SR - MAG TAPE STATUS SUBROUTINE - 9/1/76

; CALLING SEQUENCE: CALL MTSTAT (IUNIT, ISTAT)
; WHERE: IUNIT = MAG TAPE LOGICAL UNIT NO.
; ISTAT = RETURNED MAG TAPE STATUS

; CAUTION!!! - DO NOT USE THIS ROUTINE DURING A TIME WHILE THE MAG TAPE IS BEING
; DRIVEN BY ANOTHER TASK.

.EXTN .RLEF

DEFARGS

DEF IUNIT
DEF ISTAT

DEFTMPS

FENTRY MTSTAT

LDA 0,DCMTA

.SYSTEM

.DEBL

JMP .+1

LDAFP 3

LDA 0,IUNIT,3 ; GET LOGICAL UNIT NO.

DOA 0,MTA ; SELECT UNIT NO.

DIA 0,MTA ; GET STATUS

STA 0,ISTAT,3 ; PASS STATUS TO CALLING PROGRAM

LDA 0,DCMTA

.SYSTEM

.DDIS

JMP .+1

FRET

DCMTA: MTA

END

TITLE WINISR

WINS INTERFACE INITIALIZATION AND
INTERUPT SERVICE ROUTINE

CALLING SEQUENCE: CALL STWIN(IMESG,JIN)
IMESG=MESSAGE LOCATION LABEL
JIN=FIRST WORD OF OUTPUT ARRAY

.EXTN .IXMT,.UIEX
IWIN=66
NDAT=-14

DEFARGS

DEF IMESG
DEF JIN

DEFTMPS

FENTRY STWIN
LDA 0,IMESG,3 ;MESSAGE ADDR
STA 0,AMESG
LDA 0,JIN,3 ;FIRST WORD ADDRESS-DATA ARRAY
STA 0,AOUT
STA 0,TAOUT
SUBZL 0,0
STA 0,MSVAL ;INIT MESSAGE TO 1
LDA 0,DCWIN ;LOAD DEVICE CODE
.SYSTEM
.IRMV
JMP .+1
LDA 0,DCWIN
LDA 1,ADCT ;LOAD ADDP OF DEVICE CONTROL TABLE
.SYSTEM
.IDEF ;DEFINE INTERRUPTING DEVICE
JMP .+1
NIOC IWIN ;CLEAR DEVICE
NIOP IWIN ;RESET SYNC FLOP
NIOB IWIN ;SET BUSY
FRET

AMESG: 0
AOUT: 0
TAOUT: 0
DCWIN: IWIN
ADCT: DCTWIN
NWORD: NDAT
TNWORD: NDAT

ISRWIN: STA 2,PAC2
STA 3,PAC3
DIAS 1,IWIN ;READ WORD
STA 1,0TAOUT ;STORE AT ADDRESS IN TAOUT


```

ISZ    TAOUT      ;INCREMENT ADDRESS
ISZ    TNWORD     ;INCREMENT WORD COUNT,SZ
JMP    LEAVE
LDA    1,NWORD    ;IF ALL WORDS IN,
STA    1,TNWORD   ;RESET WORD COUNT
LDA    1,AOUT
STA    1,TAOUT    ;RESET ADDRESS
LDA    0,AMESG    ;SET UP TO XMIT
LDA    1,MSVAL
.IXMT
ISZ    MSVAL      ;ERROR. CHANGE MESSAGE
NIOP   IWIN       ;RESET SYNC FLOP
;
;
LEAVE: LDA    2,PAC2
      LDA    3,PAC3
      .UIEX
;
MSVAL: 1
PAC2:  0
PAC3:  0
DCTWIN: STSAV
      47
      ISRWIN
STSAV: .BLK 8.
      END

```

```

;
; TITLE WINCOM
; PACKAGE INCLUDES THREE SUBROUTINES,ONE FOR SENDING ONE BYTE
; COMMANDS,ONE FOR SENDING THE THREE BYTE CAGE COMMAND,AND
; ONE TO RESET THE INTERFACE. ALL COMMANDS ARE SENT THREE TIMES.
; ALSO INCLUDES A ROUTINE TO SEND A COMMAND TO INITIATE WINS OPS.
;
;

```

```

; CALLING SEQUENCE: CALL WINCMD(NCMD,MODE)
; NCMD=OFFSET OF COMMAND FROM BEGINNING OF LIST (BELOW).
; MODE=1 FOR INTERFACE COMMANDS, 2 FOR WINS COMMANDS
;

```

```

; CALLING SEQUENCE: CALL WINCAG(NHI,NLO)
; NHI=HI BYTE OF HEX 4 DIGIT CAGE POSITION
; NLO=LO BYTE OF SAME
;

```

```

; CALLING SEQUENCE: CALL RESET
;

```

```

; CALLING SEQUENCE: CALL WINSTRT
;

```

```

; WINDC=66 ;DEVICE CODE
; SUBROUTINE WINCMD
;

```

```

DEFARGS
DEF NCMD
DEF MODE

```

```

;
DEFTMPS

```

```

FENTRY WINCMD

```

```

LDA 0,ONCMD,3
LDA 1,OMODE,3      ;LOAD MODE
STA      ;STORE IT
LDA 2,ACMDLIST      ;LOAD ADDR OF TOP OF COMMAND LIST
ADD 0,2             ;INCREMENT BY COMMAND OFFSET
LDA 0,0,2           ;LOAD CMD CODE AC0
DSZ NMODE           ;TEST FOR MODE=1
JMP OUTA           ;MODE IS 2,USE DOA
DOB 0,WINDC         ;MODE IS 1,USE DOB
FRET

```

```

;
OUTA: LDA 2,TXCNT      ;LOAD NR OF TIMES TO SEND
LOOP: LDA 1,KTEST      ;LOAD NR OF RTC PULSES DELAY
      .SYSTEM
      .DELAY
      JMP .+1
      DOA 0,WINDC       ;SEND COMMAND
      INC 2,2,SZR       ;DONE?
      JMP LOOP          ;NO,SEND AGAIN
      FRET              ;YES,RETURN

```

```

;
;
NMODE: 0
ACMDLIST: CMDLST
CMDLST: 000           ;RUN
        200           ;STANDBY
        040           ;TEST
        020           ;CCW SLOW SLEW
        024           ;CCW FAST
        030           ;CW SLOW
        034           ;CW FAST
        003           ;XFER TO COSMAC MONITOR
        001           ;ENABLE INTERFACE TO RX
        000           ;DISABLE INTERFACE

```

```

;
;
SUBROUTINE WINCAG

```

```

DEFARGS

```

```

DEF NHI
DEF NLO

```

```

;
DEFTMPS

```

```

FENTRY WINCAG

```

```

LDA 0,ONHI,3
STA 0,HICAG          ;STORE HIBYTE
LDA 0,ONLO,3
STA 0,LOCAG          ;STORE LOBYTE
LDA 2,TXCNT          ;LOAD NR OF TIMES TO SEND
RPT: LDA 1,KTEST      ;LOAD WAIT PERIOD
      .SYSTEM

```



```

LDA 3,ADSTOR
LDA 2,CNTSTOR
INC 3,3
INC 2,2,SZR
JMP AGAIN
FRET

```

```

;INCREMENT ADD TO NEXT CHAR
;THROUGH YET?
;NO, GO AGAIN
;YES,RETURN

```

```

;
;
ADSTOR: 0
CNTSTOR:0
TRCNT: -4
MESSAD: MESS
MESSG: 44
      120
      104
      15
;
;
END

```

```

C FILE NAVDAT.FR
  REAL LG0,LNG0,K

```

```

C COMMON /NAVDAT/ NMODE,
1 DLG, DLNG, DALT, LG0, LNG0, ALT0.
1 VN, VE, VD.
1 A11, A21, A31, A12, A22.
1 A32, A13, A23, A33.
1 DTHN, DTHE, DTHD.
1 GD, SCLT, TRLT, SNLT, CSLT.
1 RLAT, RLNG, RLATHIV,RLNGHIV,
1 OMESLT, OMECLT.
1 DVNN, DVEN, DVDN,
1 DVXI, DVI, DVZI, DVNI, DVEI, DVDI.
1 DTHX, DTHY, DTHZ, DTHX1, DTHY1, DTHZ1.
1 DTHX2, DTHY2, DTHZ2.
1 DPHX, DPHY, DPHZ.
1 F(15,15), G(15,15), H(15,15).
1 POS(3), CNP(3,3), P(15,15).
1 B(15,15), R(15,15), K(15,15).
1 A(15,15), Q(15,15), V(3), FN(3)

```

C

```

C      FILE TSKCOM.FR
C
C      COMMON BLOCK STORAGE FOR INTER-TASK COMMUNICATION
C
COMMON /TSKCOM/ JIN(12),MSGSRC,MSGDSP,MSGNB3,MSGMTP,IDMODE,IAMODE,
1  INMODE,KFLAG,MTMODE,JDATE(3),JTIME(3),JNAME(16),MTNAME(5),
1  JRD,MUXDAT(16),XN(3),TXAK
C

```

```

C      FILE BHMDAT.FR
C
C      COMMON BLOCK STORAGE FOR BH NAVIGATOR
C
COMMON /BHMDAT/ ITIME,RTIME,IGN(3),IAN(3),JGN(3),JAN(3),
1  KGN(3),KAN(3),FLAGSTAT,TIMREF
C
C

```

```

C      FILE KALCOM.FR
COMMON /KALCOM/ NSO,NH,NI,STIME,IUPDT,TSTEP

```

```

C      FILE BIASCM.FR
COMMON/BIASCM/XGES,YGES,ZGES,ALFX,ALFY,ALFZ
C      FILE CALNUMS.FR
COMMON /CALNUMS/XAK,YAK,ZAK,XAB,YAB,ZAB,XAY,XAZ,YAX,YAZ,
1  ZAX,ZAY,XGK,YGK,ZGK,XGB,YGB,ZGB,XGY,XGZ,YGX,YGZ,ZGX,ZGY,
1  XGAX,XGAY,XGAZ,YGAX,YGAY,YGAZ,ZGAX,ZGAY,ZGAZ,
1  XAKP,XAKN,YAKP,YAKN,ZAKP,ZAKN,TXAKV,ANDAT(16)

```

FILE KALIN--FILTER INITIALIZATION DATA

```

0.0      0.0      0.0      0.0      0.0      0.0      3.000E-04 3.000E-04
0.000E-05 2.350E-12 2.350E-12 2.350E-12 1.000E-12 2.333E-07 2.333E-07
8.553E-16 8.553E-16 8.553E-16 2.700E-14 2.550E-10 2.550E-10
.0001      .0001      .0001
8.0      8.0      1

```

FILE CALDAT--INSTRUMENT CALIBRATION DATA

```

.514194E-02 .500172E-02 .520900E-02-.907251E-01-.386633E-01 .638970E-02
.305331E-02-.227311E-02 .503561E-02-.179287E-02 .578769E-02 .337685E-02
.284985E-04 .294697E-04 .291705E-04-.308259E-04 .403735E-04 .269194E-04
.193801E-02-.298583E-02 .472812E-02-.200523E-02 .585015E-02 .328026E-02
-.252270E-05 .194406E-07 .361214E-06-.534370E-07-.100175E-06-.731990E-07
-.370247E-06-.886280E-06-.249739E-05
.00514405 .00514342 .00500324 .00500399 .00521157 .00521052

```

References

¹A. C. Watts, *The Application of Inertial Navigation to Wellbore Positional Surveying*, SAND80-0913 (Albuquerque: Sandia National Laboratories, June 1982).

²S. M. Kohler, *Inertial Navigation System for Directional Surveying*, SAND82-1668 (Albuquerque: Sandia National Laboratories, September 1982).

³J. E. Bortz, "A New Mathematical Formulation for Strapdown Inertial Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, AES-7:1, January 1971.

⁴A. C. Watts, *The Tiger II Ring Laser Gyro Inertial Navigation System*, SAND80-0912 (Albuquerque: Sandia National Laboratories, May 1981).

⁵A. Gelb et al, *Applied Optimal Estimation* (Cambridge, MA: Massachusetts Institute of Technology, 1974).

DISTRIBUTION:

DOE/TIC-4500 (Rev 70) UC-66c (488)

Venture Innovations
PO Box 35845
Houston, TX 77035
Attn: T. Anderson

Shell Oil Co.
Two Shell Plaza
PO Box 2099
Houston, TX 77001
Attn: E. Bingman

Dyna-Drill
PO Box C-19576
Irvine, CA 92713
Attn: L. Diamond

NL Petroleum Services
PO Box 60087
Houston, TX 77205
Attn: J. E. Fontenot

Texas A&M University
Center for Tectonophysics
and Dept of Geology
College Station, TX 77843
Attn: M. Friedman
Professor of Geology

Phillips Petroleum Co.
Geothermal Operations
655 East 4500 South
Salt Lake City, UT 84107
Attn: T. Turner

Smith Tool
Geothermal Operations
PO Box C-19511
Irvine, CA 92713
Attn: J. Kingsolver

Dresser Industries, Inc
Security Division
PO Box 24647
Dallas, TX 75224
Attn: J. W. Langford

H. E. Mallory
PO Box 54696
Tulsa, OK 74155

N. L. Baroid
6400 Uptown Blvd NE, 365W
Albuquerque, NM 87110
Attn: G. Polk

Union Oil Co. of California
Union Geothermal Division
Union Oil Center
Los Angeles, CA 90017
Attn: D. E. Pyle

Los Alamos National Labs
Mail Stop 570
Los Alamos, NM 87545
Attn: J. C. Rowley

Otis, Inc
PO Box 34380
Dallas, TX 75234
Attn: W. D. Rumbaugh
Research and Development

Halliburton, Inc
Drawer 1431
Duncan, OK 73533
Attn: D. Smith

Amoco Production Co.
Research Center
PO Box 591
Tulsa, OK 74102
Attn: T. Warren

Superior Oil
Eastern Division
PO Box 51108 OCS
Lafayette, LA 70505
Attn: E. Martin

B. J. Livesay
129 Liverpool
Cardiff, CA 92007

US Department of Energy/ALO
Nuclear & Geosciences Division
PO Box 5400
Albuquerque, NM 87115
Attn: W. P. Grace

DISTRIBUTION (cont):

US Department of Energy (4)
Geothermal Hydropower
Technologies Division
Forrestal Bldg, CE 324
1000 Independence Ave SW
Washington, DC 20585
Attn: J. Bresee
D. Clements
R. Toms
D. Allen

0324 A. C. Watts
1600 D. B. Shuster
1620 M. M. Newsom
1622 R. D. Andreas
1622 R. Wardlaw (100)
2330 E. H. Barsis
2334 G. M. Heck (2)

2334 S. M. Kohler
2336 C. R. Blaine
9000 G. A. Fowler
9700 E. H. Beckner
9730 W. D. Weart
9740 R. K. Traeger
9741 J. R. Kelsey (20)
9743 H. C. Hardee
9746 B. Granoff
9747 P. J. Hommert
9750 V. L. Dugan
9753 D. A. Northrop
9760 R. W. Lynch
9770 G. E. Brandvold
8214 M. A. Pound
3141 L. J. Erickson (5)
3151 W. L. Garner (3)