Hybrid diagnostic techniques

by THOMAS K. SEEHUUS and WILLIAM A. HARMON

Boeing Company
Huntsville, Alabama

and

WILLIAM MAASBERG

International Business Machines
Huntsville, Alabama

INTRODUCTION

The Automated Maintenance Procedures at the Boeing Company's Huntsville Hybrid Facility were the result of a joint development effort by the Boeing Company and International Business Machines. (IBM). The analog and linkage portions are the responsibility of Boeing while the Digital System is maintained by IBM.

The combined effort has resulted in highly efficient Diagnostic Maintenance procedures that have given us a profitable degree of reliability for all elements of the Hybrid System.

Equipment configuration

The stringent requirements for diagnostics resulted from the complexity and size of our Hybird Installation. Although many requirements have changed as portions of the system have been updated, our maintenance philosophy has remained the same.[1,2] The present equipment complement of the Boeing Huntsville Hybrid System is listed in Figure 1. The major components of the system are four Applied Dynamic (ADI) analog computers, sixty channels of digital to analog conversion (DAC), fifty multiplexed channels of analog to digital conversion (ADC) and an IBM 360 Model 44 Digital Computing System.

The real time monitor currently operating on the 360/44 Hybrid Computer at Boeing Huntsville was developed jointly by Boeing and IBM almost two years ago. It was decided that some sort of system test capability was also needed to aid in debugging the software and assure a maximum level of hardware reliability.

ADI ANALOG COMPLEMENT (4 CONSOLES)

1024	OPERATIONAL AMPLIFIERS (256 SUMMER-INTEGRATORS, 256 SUMMERS, 512 INVERTERS
640	SERVOSET POTENTIOMETERS
160	HANDSET POTENTIOMETERS
172	ELECTRONIC QUARTER-SQUARE MULTIPLIERS
128	VARIABLE DIODE FUNCTION GENERATORS
40	FIXED SIN-COS DIODE FUNCTION GENERATORS
48	FIXED X^2 DIODE FUNCTION GENERATORS
16	FIXED LOG X DIODE FUNCTION GENERATORS
80	HARD LIMITERS
112	COMPARATORS
48	TRACK TRANSFER UNITS
4	HIGH-SPEED REP OP CONTROL UNITS

LINKAGE EQUIPMENT COMPLEMENT

50	ANALOG TO DIGITAL CONVERTER CHANNELS
60	DIGITAL TO ANALOG CONVERTER CHANNELS
6	DISCRETE OUT WORDS (32 BIT)
6	DISCRETE IN WORDS (32 BIT)
32	PRIORITY INTERRUPT LINES
1	ANALOG CONTROL CHANNEL

DIGITAL COMPLEMENT (IBM)

1	IBM DIGITAL COMPUTER, 360 MOD/44
3	MAGNETIC TAPE UNITS, 9-TRACK (2401)
1	MAGNETIC TAPE UNIT, 7-TRACK (2401)
1	LINE PRINTER (1403)
1	OUTPUT PRINTER (1053)
1	CONSOLE TYPEWRITER (1052)
2	VISUAL DISPLAY UNITS (2260)
4	I/O DATA ADAPTERS (2701)

Figure 1—Equipment complement

Basically one must be able to detect a malfunction and rapidly isolate it to a specific portion of either the software or hardware. Consequently, a System Test (SYSTST) program was developed.

The SYSTST program has been very successful, both during debugging of the initial Real Time Monitor and as aid in checking out improvements. The program is

readily available to application programmers to confirm suspected hardware malfunctions. Thus, the problem areas are immediately identified as either the application program or hardware. Corrective action is then initiated. Should the problem be hardware, the evidence collected by the SYSTST program is given to either the IBM Customer Engineers or the Boeing Hybrid Operations Personnel, who conduct detailed diagnostics and tests to locate and correct the malfunction.

360/44 Hybrid System Test Program (SYSTST)

A. SYSTST philosophy

SYSTST was written to fulfill the following primary objectives:

1. Aid in initial debugging of the 360/44 BPS Hybrid Monitor System
2. Aid in maintaining and improving the system
3. Test the Real Time I/O for a basic level of operation and accuracy

A secondary objective was to develop in SYSTST a basic test framework which could be easily modified, expanded, and developed into a rigorous test of the 360/44 HBPS software and hardware.

B. SYSTST organization

Although the individual tests were written to satisfy primary objectives, the final organization of SYSTST was dictated by the secondary objectives of flexibility and expandability. Consequently, SYSTST consists of one main program and a number of test subprograms. The main program contains the operator communications routines necessary for test completion and diagnostic action as a result of error. The test subprograms are entered from and returned to the basic program, as illustrated in Figure 2.

At the present there are six test subprograms which accomplish the following tasks:

1. Test display system
2. Test real time tape, single unit
3. Test real time tapes, all units
4. Test analog I/O from foreground
5. Test interval timers
6. Test analog I/O, interval timers, and tapes in real time

SYSTST is readily capable of expansion in either of two ways:

1. Expansion and refinement of the present test subprograms

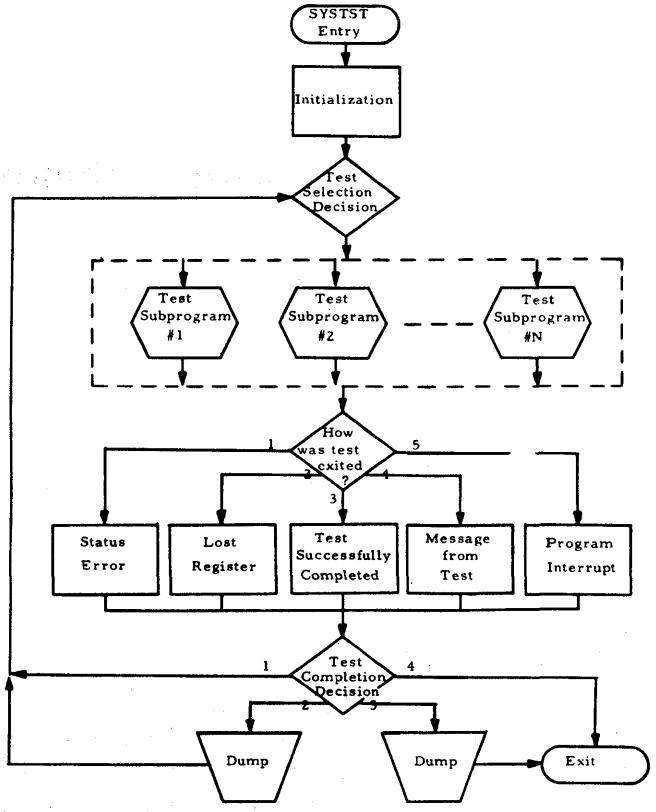

Figure 2—SYSTST flow chart

2. Addition of any number of new test subprograms

C. Diagnostics and error messages

It is readily apparent in Figure 2 that there are several basic error or diagnostic messages as well as provision for handling messages which are unique to a given test. These test exit messages are further described below:

1. STATUS ERROR

If during the course of a test, any I/O request results in an error or unusual condition indication from the I/O device, the test will be terminated, the test subprogram exited, and the following diagnostic message displayed:

STATUS ERROR

1. STATEMENT: CALL XXXXXX 2. LOGICAL UNIT: X

3. STATUS: X 4. I/O OLD PSW: XXXX XXXX XXXXXXXX

5. CSW: XXXXXXXX XXXX XXXX

2. LOST REGISTER

Each test subprogram is responsible for checking all the registers (except those used by the FORTRAN compiler) each time a call statement is made for a real time function. If a register is lost, the test is terminated, the test subprogram exited, and the following diagnostic message displayed:

REGISTER LOST DURING EXECUTION OF: CALL NNNNNN

REGS: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

LOST: XXXXXXXXXX X X X X X X

(X=O indicates register not lost, X=1 indicates register lost. Registers 0, 1, 13, 14 and 15 are normally employed in Fortran branching and linking and, consequently, are not checked. Any additional registers used by the compiler in setting up a specific call cannot be checked.)

3. TEST SUCCESSFULLY COMPLETED

The above diagnostic message is felt to be sufficiently self-explanatory.

4. ERROR OR DIAGNOSTIC MESSAGE FROM TEST

Use of this SYSTST return path is entirely at the discretion of the test subprogram writer. This option is used whenever the programmer wants to terminate his test subprogram and idicate a unique reason, or whenever he wishes to output a unique message upon successful test completion. It should be noted that a programmer may print or display any data or messages; he may also request any operator response desired during the course of his test subprogram.

5. PROGRAM INTERRUPT

Any errors which would normally result in a program interruption and the resultant abnormal end of job (ABEOJ), will cause immediate test termination, return to SYSTST MAIN, and display of the following message:

PROGRAM INTERRUPTION

INTERRUPTION CODE: XX INSTRUCTION COUNTER: XXXXXX

D. SYSTST OPERATION

SYSTST OPERATION is relatively easy and self-explanatory, as the main program and each of the test subprograms were written with the following philosophies:

1. The operator will communicate the SYSTST only through the display system (although some test data may be output on the printer, depending on the test subprogram).
2. The operator will be kept informed of test progess.
3. The operator will be given explicit instructions when a decision and/or response is required.
4. The operator will initiate no unrequested entries, except to request a hard copy of the display (which he may do at any time by depressing the shift and print keys).
5. Any test subprogram may be entered in any order and any number of times. The cumulative numbers of entries and successful completions are displayed each time a test selection decision is requested.

Analog and linkage diagnostics

The maintenance and calibration of analog computer components and linkage equipment is accomplished by the use of Hybrid Diagnostics. The use of diagnostics for analog computer maintenance is a relatively new procedure and initially their effectiveness and ease of application were not fully realized. It was found in developing the diagnostics that most of the antiquated manual checks and calibrations could be adapted to digital control. In addition, some components that previously could not be adequately tested can now be checked under digital control.

Another result of using Automated Diagnostics is that operational tests and calibration checks require much less time, making possible more frequent testing. Figure 3 shows a comparison of the time required to run manual and diagnostic tests. The use of diagnostics enable maintenance personnel to apply more time to areas previously neglected. In addition to time savings, a thorough maintenance program can be established that does not entirely depend upon the proficiency of the maintennance personnel. An automatic check is consistently thorough and accurate.

The Hybrid Diagnostics used at the Boeing Huntsville Facility are grouped into two types: general opera-

TYPE OF CHECK	MANUAL CHECK MINUTES	DIAGNOSTIC CHECK MINUTES
STATIC CHECK	60	3
INTEGRATOR RATE	60	2
INTEGRATOR DRIFT	120	2
COMPARATOR GAP	30	1
MULTIPLIER	60	1
NON-LINEAR	120	8

Figure 3—Comparison of diagnostics

tion and calibration. General operation diagnostics check the overall operation of equipment that requires no calibration, such as logic devices. The calibration diagnostic is capable of checking accuracy as well as general operation. The calibration diagnostic can be used, in many cases, to aid the technician performing calibration or repair.

The analog and linkage diagnostics have been written as subroutines of a main calling program. Each diagnostic need not be loaded separately when going from one to another. Figure 4 is a flow chart of the diagnostics. The diagnostic subroutines and calling program reside on magnetic tape in core format. They are loaded in approximately five seconds, after which they are accessible from the IBM 2260 Display Unit. Figure 5 is a photograph of the 2260 displaying the program options. Following are descriptions of the analog and linkage diagnostics.

Non-linear preventive maintenance diagnostic

Basic observations

Our approach is based upon the contention that all non-linear components have one thing in common: they produce a predetermined non-linear function when they receive a linear or ramp function as an input. Therefore any non-linear device may be tested using a highly accurate ramp synchronous with a highly accurate predetermined non-linear function. This enables one to test all members of the non-linear family as relatively simple functions.

The ability to generate a ramp function synchronous with a desired non-linear function became the key to a successful diagnostic test. This capability is available in the digital portion of our hybrid system, but the conversion equipment does not meet the accuracy requirements for specification testing. At this point we perceived the need for two digital to analog converters of high accuracy.

Design objectives for high accuracy DAC

A. Accuracy equal to or better than .005%
B. An integral part of our present hybrid system
C. Ease of maintenance and calibration
D. Controlled manually or automatically

The high accuracy DAC shown in Figure 1 met all design objectives. It requires 16 bi-polar amplifiers, 1 bi-polar integrator, 15 electronic switches, 2 potentiometers, and 3 free 100 K resistors. Fifteen bits plus sign was selected because this DAC has the accuracy required. We did not go to sixteen bits because the least significant bit of a sixteen bit DAC is well within the noise level of our system.

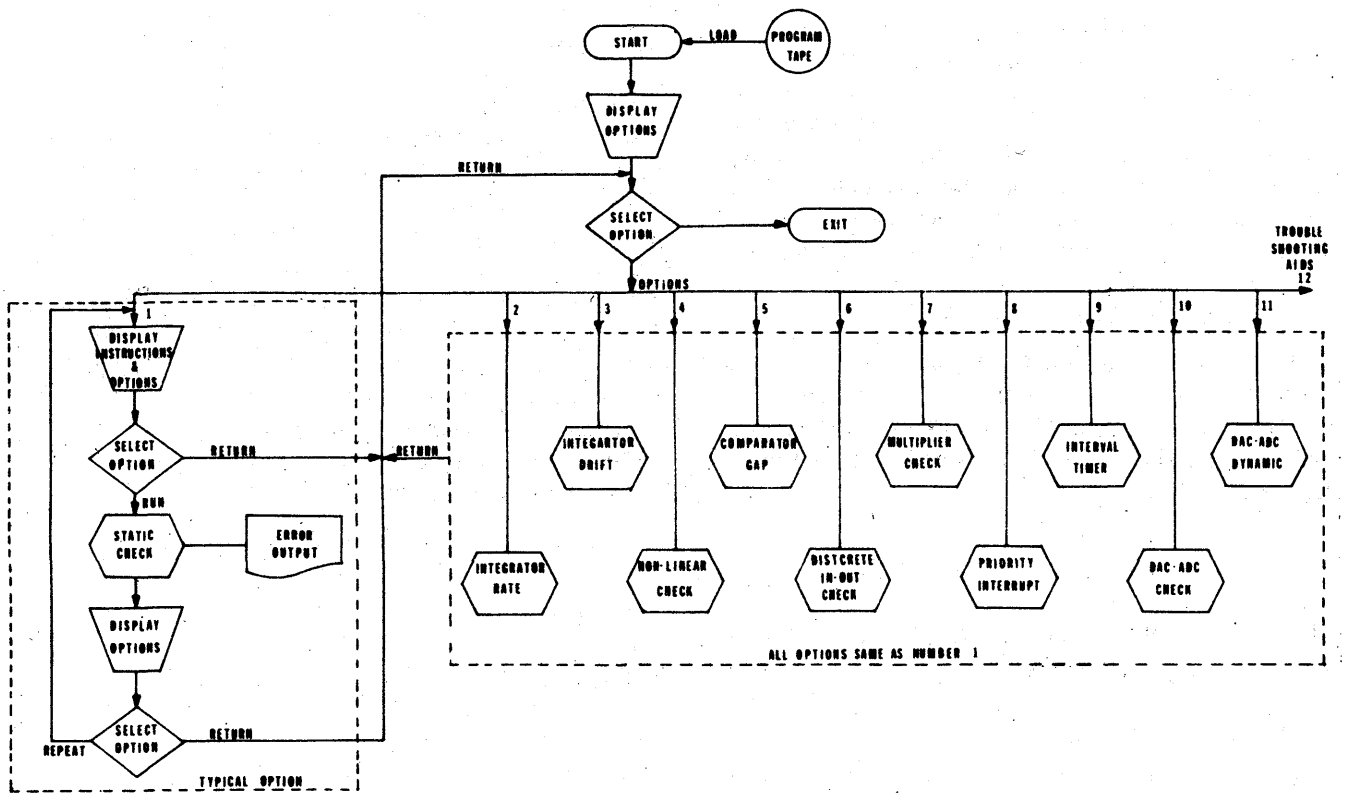

Figure 4—Diagnostic flow chart

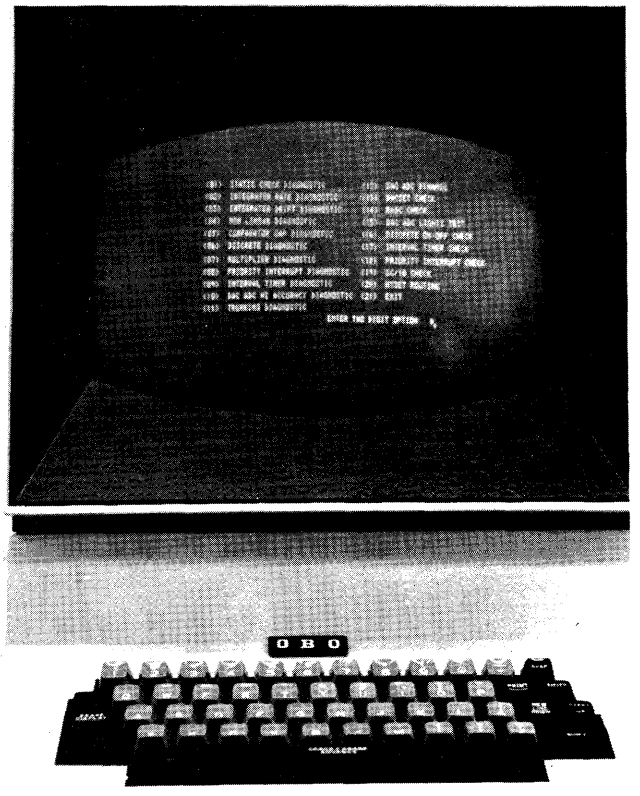

Figure 5—2260 display

The high accuracy DAC operates as follows: The plus volt reference is applied to the first summer on the left. Plus 50 volts is available at the input to electronic switch number one and is halved at the input to each succeeding electronic switch. The electronic switches may be operated manually at the analog console or by discrete lines from the digital. Each electronic switch has a 100 K resistor that is used as an input resistor to the summer at the top of Figure 6. The correct polarity of voltage is observed throughout the string by the use of bi-polar amplifiers. The negative output of the upper summer is fed into a gain of one, while the positive output determines the initial condition of the integrator. The sign bit of the digital to analog conversion determines whether the integrator is in initial condition or operate. If the sign is positive, the integrator will be in I C, with a negative sign in operate. The integrator is in times 100 mode to impart a filtering effect. This limits frequency response but keeps the DAC within design specifications.

The plus and minus 100 volt adjustments are necessary to compensate for the slight inaccuracies of the various components. They are adjusted in the following manner:

1. Throw all bits except the sign bit on
2. Set the +100 v adjustment to +99.997 volts

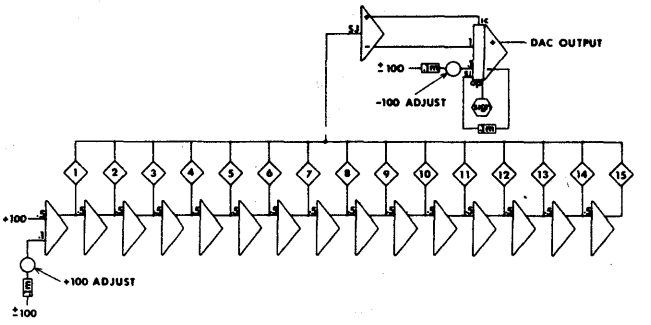

Figure 6—High accuracy DAC

3. Throw sign bit on
4. Set −100 v adjust to −99.997 volts

Caution: The plus 100 adjustment must be set first.

The accuracy of the DAC was verified in the following manner: the DAC test was conducted using the technique and equipment shown in Figure 7. The DAC is the input to one side of a null meter with a calibrated test voltage the other input.

The bit by bit error plot shown in Figure 8 verified

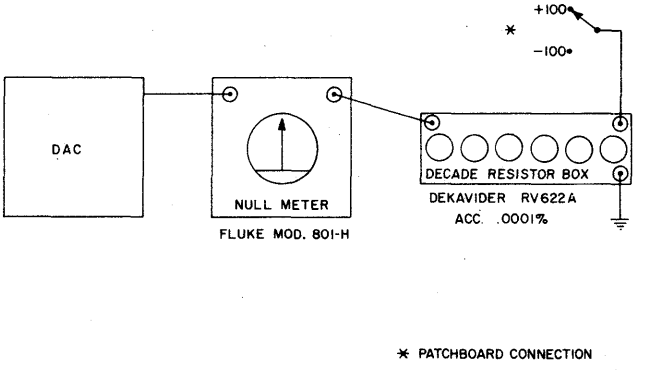

Figure 7—DAC test configuration

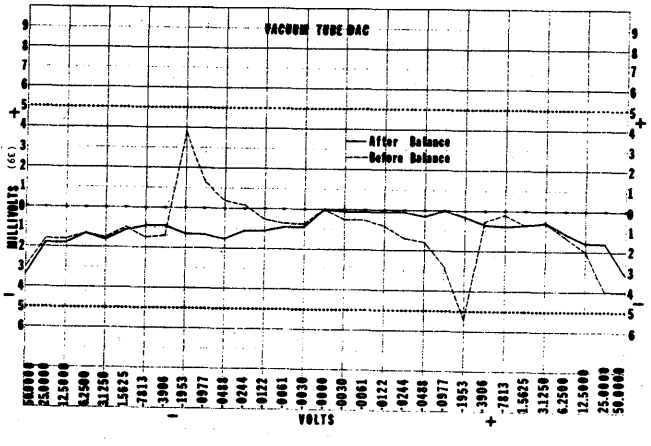

Figure 8—Vacuum tube DAC (Balance)

that our design specifications could be met on a vacuum tube analog after a careful balance of all amplifiers.

The bit by bit error plot shown in Figure 9 was taken on a different analog console after a complete balance of amplifiers, in addition to setting the plus and minus 100 v adjustments.

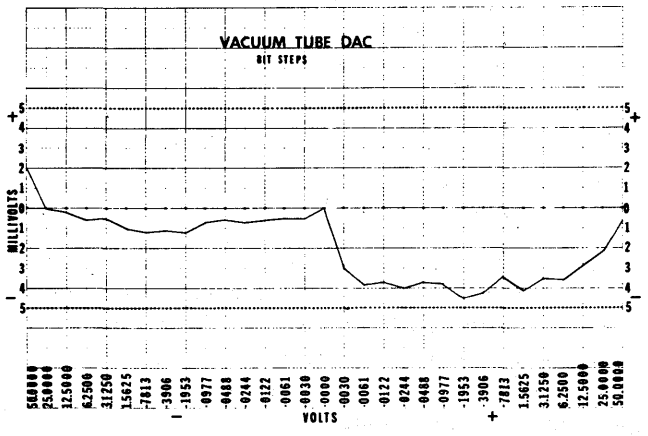

Figure 9—Vacuum tube DAC (Bit steps)

The plot shown in Figure 10 is an error plot in 10 volt steps from +100 v to—100 volts. This also falls within design specifications. Figures 11 and 12 were taken on a solid state console without balancing the amplifiers. The plus and minus 100 v adjustments were made, however.

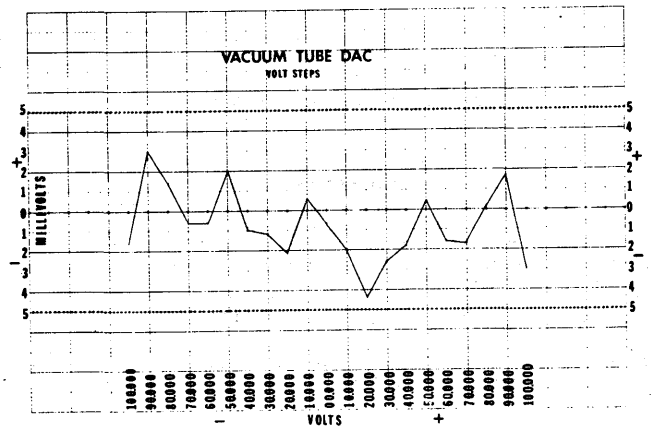

Figure 10—Vacuum tube DAC (Volt steps)

The preceding tests have given us confidence that the DAC will meet accuracy requirements on any well-maintained console.

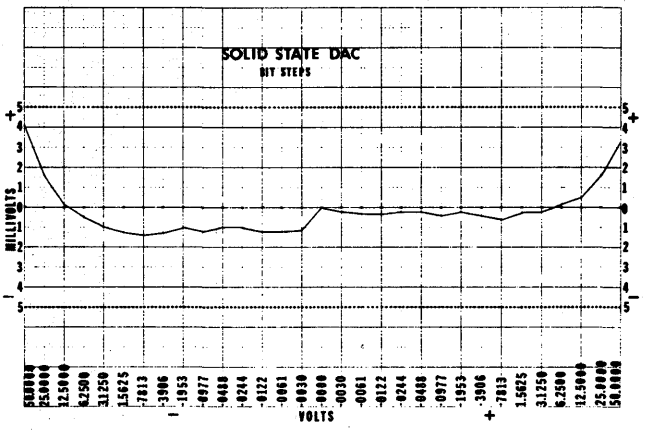

Figure 11—Solid state DAC (Bit steps)

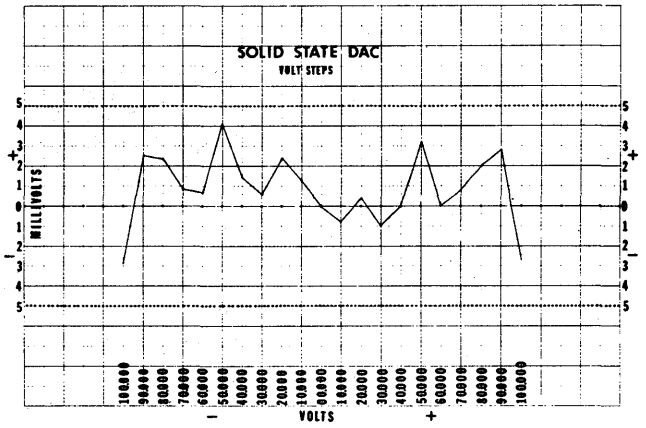

Figure 12—Solid state DAC (Volt steps)

Testing non-linear components

The test configuration shown in Figure 13 is the method used for daily checks. During the development stages the output of the error amplifier was plotted together with plots of desired function and the actual

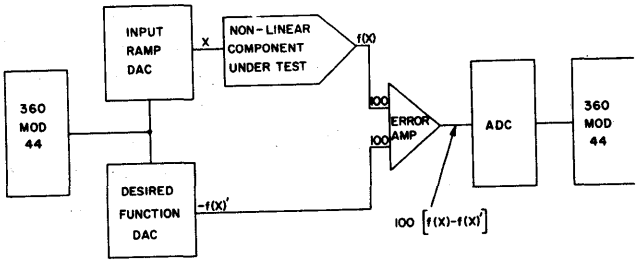

Figure 13—Non-linear test configuration

function. The first tests conducted were on an X^2 card. The results of these tests are shown in Figure 14. The plot took three hybrid runs. On the first run the desired curve was plotted, the second was a plot of the actual output of the X^2 card, and the third, a plot of the error between the two.

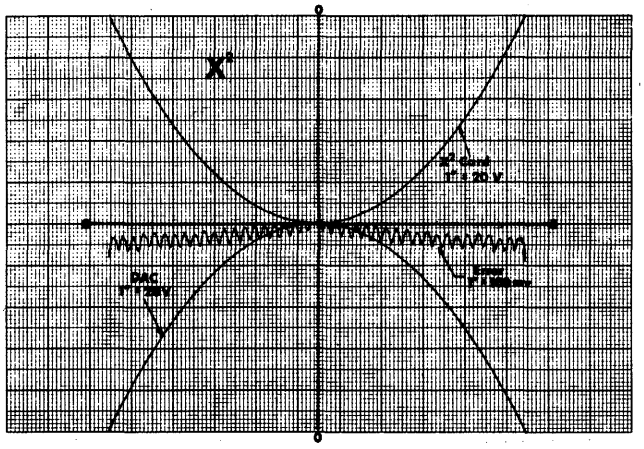

Figure 14—X^2 plot

Our next step in the verification on the testing procedures was to repeat our previously described plotting and testing for other members of the non-linear family. Figures 15, 16 and 17 show the results. All plots shown were generated using a ramp input to the tested non-linear device in 0.5 volt steps.

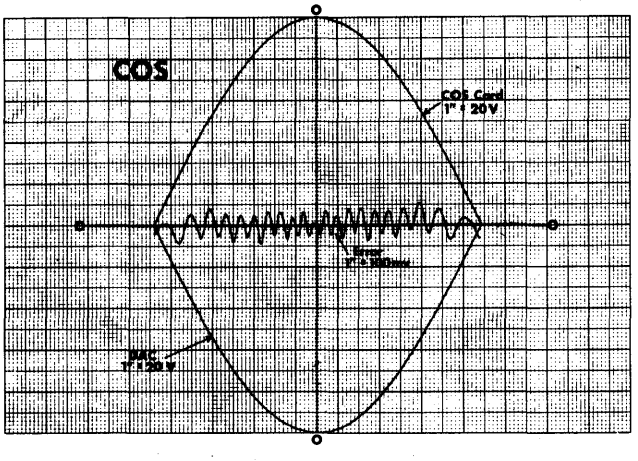

Figure 15—Cosine plot

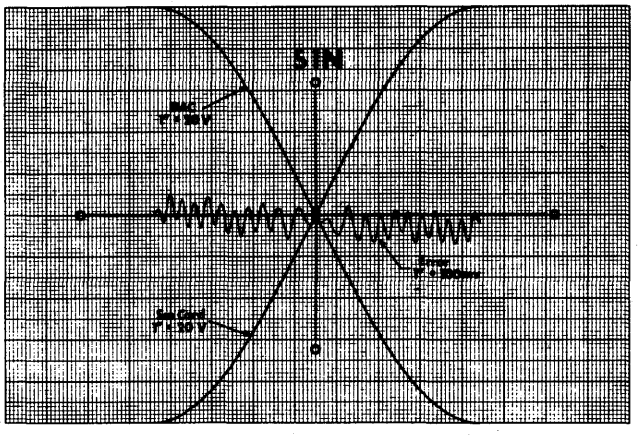

Figure 16—Sine plot

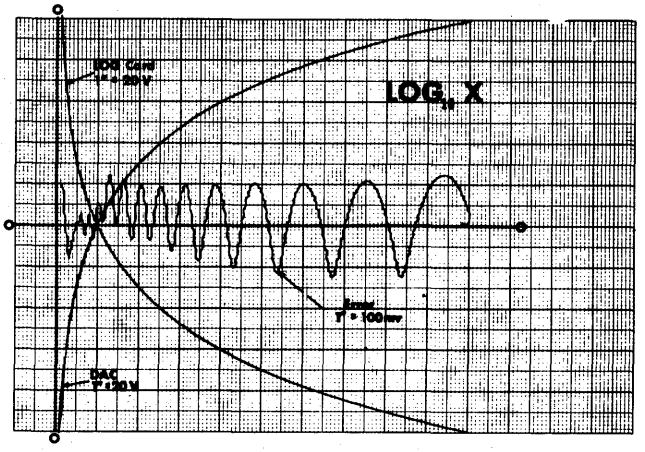

Figure 17—Log plot

Diagnostic technique

The program implements two of the highly accurate DAC's, one to provide a comparison curve (Figure 13). The outputs of the non-linear devices are summed with the comparison curve (opposite polarity into 100 gain amplifiers). The amplifier outputs, which represent 100 times the non-linear device error, are connected to ADC's. The following steps describe the method used to check Sine, Cosine, and Logarithmic devices:

1. First the circuitry and components used to test non-linear devices are checked by the program to insure proper operation. In the event of error, a pause results that allows the operator to investigate prior to program excution.
2. Error amplifiers associated with non-linear devices under test are enabled by descrete logic.
3. The input ramp is stepped from—100 to +100 volts, in one-half-volt increments and is applied to the non-linear device under test. At each increment

the program uses the value of the ramp to calculate the value of the comparision curve. Each ADC is read at every increment and the error magnitude is compared to the previous reading. The larger of the errors is saved.

4. Following the last one-half-volt increment, the largest error reading for each non-linear device is compared to that device's tolerance. Out-of-tolerance readings are recorded together with the address of the non-linear device and error amplifier.

The non-linear diagnostic program has an option for continuous operation. The ramp and comparison curve are operated continuously with the 100 gain amplifier outputs observed on an oscilloscope or recorder. The diagnostic may be used to analyze errors and perform necessary calibration or repair.

Total run time for the non-linear diagnostic program is approximately eight minutes.

The settling time, and therefore the frequency response of the high accuracy DAC, limits our testing to very low frequencies. We found no way to overcome this deficiency. The use of the integrator as the output amplifier was required to keep the noise level of the DAC within the 5 millivolt design specification. This problem should be rectified in the future and dynamic testing of nonlinear components using the method described will indeed become a reality.

Multiplier preventive maintenance diagnostic

The objective of the Multiplier Diagnostic Program is to insure that all multipliers function within their specified tolerance. The analog boards used in this diagnostic may be used manually to analyze errors and make necessary calibrations. All multipliers in an analog console may be checked simultaneously by the following steps:

1. A ramp varying between—100 and +100 volts is applied to the X input of all multipliers. A constant—100 volts is supplied to the Y input. Multiplier outputs are summed with the inverted ramp (+100 volts to—100 volts) by means of 100 gain amplifiers. The amplifier outputs, which represent 100 times multiplier error, are connected to ADC's.
2. The ADC's are continuously read as the input ramp varies from —100 volts to +100 volts. A comparison of the averages of selected readings is made and the largest average is saved. When the ramp ends, the X input and Y input to the multipliers are exchanged under program control. The ramp is restarted and the same procedure is used to obtain the largest average reading.
3. The largest average reading for each multiplier is compared to a predetermined tolerance. Out-of-tolerance readings for X or Y inputs are recorded together with the multiplier and error amplifier address.

The above check may be used manually by controlling the exchange of inputs with analog push buttons. The ramp is allowed to run continuously and the 100 gain amplifier outputs may be observed on an oscilloscope or a recorder. Total run time for Multiplier Diagnostic Program is approximately one minute.

Static preventive maintenance diagnostic

The Static Diagnostic Program will check the following equipment on an analog console:

1. Every input and output on all operational amplifiers
2. All servo set potentiometers for proper operation
3. All analog trunks
4. Mode control logic
5. The analog addressing system
6. All reference and electronic switches
7. Hand set potentiometer fuses

Permanently wired patchboards are used in conjunction with the Static Diagnostic to check the above components.

The Static Diagnostic requires three minutes hybrid time. To isolate bad components the computer modes and addressing are controlled manually. The actual value of the suspected components is read on the console digital volt meter, and appropriate action is taken using conventional trouble shooting techniques.

Integrator rate preventive maintenance diagnostic

The Integrator Rate Diagnostic is used to check all time scales as well as relay and electronic switching. The programming sequence is described in the following steps:

1. The integrators are allowed to integrate at one volt per second (Xl) for eight seconds. The integrators are then switched, via relays, to hold. All integrators are checked for the proper reading and if they are in error by more than a predetermined amount, the address, actual reading, specified value, and calculated error are recorded.
2. The above step is repeated for all additional time scales. (X10, X100, X1000)
3. Electronic switching is checked using the X10 time scale by the procedure outlined in step one.

Total hybrid time for the Integrator Rate Test is two minutes.

Integrator drift preventive maintenance diagnostic

The objective of the Drift Diagnostic is to periodically check all integrators for excessive drift. Drift is checked in both operate and hold using the X100 capacitor. While the integrators are being checked, they continue to drift. To prevent erroneous data, each integrator's drift rate is computed individually with respect to time.

Comparator gap preventive maintenance diagnostic

The Comparator Gap Diagnostic determines the input value at which the comparators are switched. These values are used to calculate gap and offset. Permanently wired patched boards are maintained with the following configuration: All comparators have their logic-one outputs tied to the hold control line of integrators. The associated integrators are input by a control integrator. The control integrator is also patched to a one-thousandth gain amplifier, whose output is the input to all comparators. The hybrid test stores the voltages at which the integrators are placed into hold. These recorded values are used to compute the gap and offset. The comparator address, on value, off value, and offset are recorded.

The total hybrid run time for the Comparator Gap Diagnostic is one minute.

Discrete preventive maintenance diagnostic

The objective of the Discrete P. M. program is to insure that each Discrete-out and Discrete-in bit is properly aligned and is capable of representing both logic levels. This objective is accomplished in the usual manner by trunking Discrete-out Units to corresponding Discrete-in Units. The digital computer then sets each Discrete-out bit and checks that the corresponding Discrete-in bit is on. The majority of time required for the completion of the Discrete P. M. program is a function of the amount of print-out. Total time is normally less than ten seconds.

Priority interrupt preventive maintenance diagnostic

The objective of the Priority Interrupt Diagnostic is to insure that all available priority interrupt lines will provide the desired number of interrupts at the proper priority level. There are eleven priority interrupt lines available which supply the capability of interrupting the computer program on one of eleven discrete levels of interrupt. A logic-one on any priority interrupt line will cause an interrupt request. This request will be honored immediately, if none of the following conditions exist:

1. An equal or higher priority interrupt still has control of the computer program.
2. The requested level of interrupt is temporarily disabled because the system is executing some nonreentrant code.
3. The request level of interrupt was never enabled by the program.

The Priority Interrupt Diagnostic checks the individual priority interrupt lines in the following steps:

1. The program enables all available priority interrupt lines. The seven highest level interrupts are trunked to seven discrete-out lines. The remaining four interrupt lines are hard wired to the overflows of interval timers and the end of conversion of the ADC's.
2. The program sets the seven discretes to a logic-one, starts the timers, and reads the ADC's.
3. The program analyzes the order in which the lines were serviced and checks that each line was serviced only once. Any error in the order or any multiple servicing of the interrupt lines will result in the recording of the improper line.
4. The program applies a signal to the first priority interrupt line and checks that the proper level of interrupt occurs only once. The remaining lines are checked in the same manner. Any error will result in the recording of the improper line.

Total run time for the Priority Interrupt Diagnostic is less than one second.

Interval timer preventive maintenance diagnostic

The Interval Timer Diagnostic provides a means for checking both interval timers in their four modes of operation. These modes provide a means for setting the timer register to a particular time which will provide a priority interrupt at the end of the desired interval (overflow).

The following is a description of the four modes of operation:

1. Upon selection of mode one, the desired time is immediately loaded into the timer register, regardless of the timer state. The timer will then stop after the desired interval has elapsed.
2. The selection of the second mode will cause the time register to be loaded with a new value immediately upon overflow. The new value will automatically be reloaded at subsequent overflows.
3. The third mode causes the timer register to be loaded immediately with the desired time interval, regardless of the timer state. The new value will automatically be reloaded at subsequent overflows.
4. The selection of the fourth mode results in an immediate stop of the timer.

Integrator drift preventive maintenance diagnostic

The objective of the Drift Diagnostic is to periodically check all integrators for excessive drift. Drift is checked in both operate and hold using the X100 capacitor. While the integrators are being checked, they continue to drift. To prevent erroneous data, each integrator's drift rate is computed individually with respect to time.

Comparator gap preventive maintenance diagnostic

The Comparator Gap Diagnostic determines the input value at which the comparators are switched. These values are used to calculate gap and offset. Permanently wired patched boards are maintained with the following configuration: All comparators have their logic-one outputs tied to the hold control line of integrators. The associated integrators are input by a control integrator. The control integrator is also patched to a one-thousandth gain amplifier, whose output is the input to all comparators. The hybrid test stores the voltages at which the integrators are placed into hold. These recorded values are used to compute the gap and offset. The comparator address, on value, off value, and offset are recorded.

The total hybrid run time for the Comparator Gap Diagnostic is one minute.

Discrete preventive maintenance diagnostic

The objective of the Discrete P. M. program is to insure that each Discrete-out and Discrete-in bit is properly aligned and is capable of representing both logic levels. This objective is accomplished in the usual manner by trunking Discrete-out Units to corresponding Discrete-in Units. The digital computer then sets each Discrete-out bit and checks that the corresponding Discrete-in bit is on. The majority of time required for the completion of the Discrete P. M. program is a function of the amount of print-out. Total time is normally less than ten seconds.

Priority interrupt preventive maintenance diagnostic

The objective of the Priority Interrupt Diagnostic is to insure that all available priority interrupt lines will provide the desired number of interrupts at the proper priority level. There are eleven priority interrupt lines available which supply the capability of interrupting the computer program on one of eleven discrete levels of interrupt. A logic-one on any priority interrupt line will cause an interrupt request. This request will be honored immediately, if none of the following conditions exist:

1. An equal or higher priority interrupt still has control of the computer program.
2. The requested level of interrupt is temporarily disabled because the system is executing some nonreentrant code.
3. The request level of interrupt was never enabled by the program.

The Priority Interrupt Diagnostic checks the individual priority interrupt lines in the following steps:

1. The program enables all available priority interrupt lines. The seven highest level interrupts are trunked to seven discrete-out lines. The remaining four interrupt lines are hard wired to the overflows of interval timers and the end of conversion of the ADC's.
2. The program sets the seven discretes to a logic-one, starts the timers, and reads the ADC's.
3. The program analyzes the order in which the lines were serviced and checks that each line was serviced only once. Any error in the order or any multiple servicing of the interrupt lines will result in the recording of the improper line.
4. The program applies a signal to the first priority interrupt line and checks that the proper level of interrupt occurs only once. The remaining lines are checked in the same manner. Any error will result in the recording of the improper line.

Total run time for the Priority Interrupt Diagnostic is less than one second.

Interval timer preventive maintenance diagnostic

The Interval Timer Diagnostic provides a means for checking both interval timers in their four modes of operation. These modes provide a means for setting the timer register to a particular time which will provide a priority interrupt at the end of the desired interval (overflow).

The following is a description of the four modes of operation:

1. Upon selection of mode one, the desired time is immediately loaded into the timer register, regardless of the timer state. The timer will then stop after the desired interval has elapsed.
2. The selection of the second mode will cause the time register to be loaded with a new value immediately upon overflow. The new value will automatically be reloaded at subsequent overflows.
3. The third mode causes the timer register to be loaded immediately with the desired time interval, regardless of the timer state. The new value will automatically be reloaded at subsequent overflows.
4. The selection of the fourth mode results in an immediate stop of the timer.

The timer modes, registers, and priority interrupts are checked in the following manner:

1. The program checks the first and fourth modes of operation by using them to check each bit of the timer register. This is accomplished by setting the register's highest order bit, stopping the timer, and recording the contents of the register. The highest order bit is then turned off and the next lower order bit is turned on. The contents of the register are recorded. This procedure continues until all bits have been checked.
2. The program checks the third mode of operations by using it to load the timer register with a known value. After the timer overflows, the value reloaded into the register is recorded. This cycle is continued for two additional overflows to insure that the values loaded at each overflow remain the same.
3. The last overflow, resulting from step (2), changes the value in the timer register under control of the second mode. After the register is loaded, its contents are recorded. The register value is recorded for two additional overflows. The proper operation of the second mode is assured if the register values are identical and have been changed from step (2).

Through the above steps the priority interrupts caused by interval timer overflows are checked for proper sequencing.

The total run time for the Interval Timer Diagnostic is less than one second.

DAC—ADC preventive maintenance diagnostic

A complete analysis of the operation of digital to analog converters and the analog to digital converters requires the use of two diagnostic programs. One program is a static check of each magnitude bit to insure proper operation and value. The other program is a dynamic check to reveal any intermittence in overall operation or crosstalk. The following is a description of these programs:

Static diagnostic

DAC

The Static Diagnostic uses the digital to analog converter, located within an analog console, for a standard. This console-DAC is controlled by the program and its accuracy is maintained to within ± .003 volts.

A single unit of linkage-DAC's contains 30 individual digital to analog converter channels. These channels may be checked simultaneously in the following manner:

1. The least significant bit of the console-DAC and the 30 linkage-DAC's is turned on.
2. A comparison is made between the output of the Console-DAC and the 30 linkage-DAC's by means of 30 error amplifiers with gains of 1000.
3. The outputs of the error amplifiers are read by analog to digital converters and are recorded.
4. The least significant bit is turned off and the next higher order bit is turned on. This error is recorded and procedure continues until all bits have been checked.
5. After checking the highest order bit, all magnitude bits are turned on and the linkage-DAC's are checked at full scale positive.
6. The sign bits of all the DAC's are turned on and the above steps are repeated until the error for full scale negative has been recorded.

The relays and amplifiers used in this diagnostic are checked by the program before execution. In the event of an error, a pause results that allows the operator to investigate the malfunction prior to program execution. Total time required is one minute.

ADC

The analog to digital converter static diagnostic also uses the console-DAC as a standard. A single unit of linkage-ADC's contains 25 individual analog to digital converters which may be checked simultaneously in the following manner:

1. The program makes use of the same analog equipment by switching out the linkage DAC's and changing the 1000 gain amplifiers to unity gain.
2. The least significant bit of the console-DAC is turned on and fed to 25 unity gain amplifiers.
3. The outputs of these amplifiers are read by the linkage-ADC's. The program then compares these values with the original console-DAC value and records the difference.
4. The console-DAC is stepped in the same manner as in the linkage-DAC test until the errors for each bit, both positive and negative, are recorded.

The total time for execution of the linkage-ADC Static Diagnostic is less than one-half minute.

Dynamic diagnostic

DAC–ADC

The Dynamic Check provides a method for checking dynamic response of the linkage equipment. Any intermittence of operation may be found quickly and isolated by various program options. The dynamic check supplies a varying voltage to an ADC, samples the ADC at a definite time interval, and fires the DAC to the sampled ADC value.

The check is accomplished by combinations of the following options:

1. The inputs to the first and second ADC units are program coupled to corresponding outputs of the *first* and *second* DAC units, respectively.
2. The inputs to the first and second ADC units are program coupled to corresponding outputs of the *second* and *first* DAC units, respectively.
3. Any input of either ADC unit can be selected to control the outputs of both DAC units.

The program is completely flexible in that any combination of units and their synchronization may be selected. An option is available to use the interval timer to control the timer interval. The synchronization of the DAC's and the ADC's may be controlled by the interval timer, an external signal, or by the program. Detection of any intermittence is accomplished by applying a signal of varying amplitude to the desired ADC and observing the output of the desired DAC with a recorder.

This test may be used continuously to locate a malfunction.

Present diagnostic schedule

The scheduled frequency of Diagnostic Routine execution can be seen in Figure 18. We must admit that the schedule is the result of experience (four years) rather than a highly technical evaluation. The schedule evolved from a trial and error method but has proven itself over the past year to be more than adequate for our needs. The high reliability of our hybrid system has proven the worth of both the diagnostics and the frequency of use.

PERFORMED DAILY	PERFORMED WEEKLY
STATIC CHECK	INTEGRATOR DRIFT
INTEGRATOR RATE	NON-LINEAR
DISCRETE	MULTIPLIER
INTERVAL TIMER	TRUNKING
DAC-ADC HI-ACCURACY	
PRIORITY INTERRUPT	

Figure 18—Diagnostic scheduled frequency

Trouble shooting aids

Figure 19 shows the balance of the options available in our Preventive Maintenance Program. These options are used as needed for trouble shooting and in some cases, to aid in automated calibration techniques.

DAC-ADC DYNAMIC CHECK	INTERVAL TIMER CHECK
DAC-SET CHECK	PRIORITY INTERRUPT CHECK
READ ADC-DISPLAY CHECK	SUB CHANNEL TEN CHECK
DAC-ADC LIGHTS TEST	HYBRID SET-UP ROUTINE
DISCRETE ON-OFF CHECK	

Figure 19—Trouble shooting aids

SUMMARY

In the process of preparing this paper we were forced to review our efforts over the past four years, and we feel that some of these reflections may be of interest. As we stated some years ago,[2] the approach we used was forced upon us by both the relatively new design of our equipment and the fact that we were embarking upon the establishment of a new technology; which, if not new to the world, was at least new to us. We began with a staff of nine, which consisted of three degreed engineers experienced in programming and design, plus six technicians. Only two members of the technician force were experienced in analog. Looking back, we would not have had it any other way. We found, in a hybrid environment, it was easier to establish new habits of work performance than it was to break old ones.

We designed our diagnostic routines in a serial manner and began using them as they became operational. As each new test was designed, we attempted to check components not covered by previous tests. In retrospect, we feel that this was and is a valid approach. Not all diagnostics used today were anticipated in the beginning. Some we designed as the need became apparent. We have tried to remain flexible in our approach to diagnostics and have redesigned and updated as our needs and equipment varied. The diagnostics outlined in this document have been stable for the past year. We see no need for further revision, unless our equipment is changed.

It is our firm belief that the time and money spent on developing this system for maintaining our hybrid equipment has been well worthwhile. Our prime objective has been and is to provide our customers with maximum operation time coupled with a minimum maintenance effort. In this goal we have succeeded. The Boeing Huntsville Facility operates 18 hours per day, six days a week (sometimes seven); the operation is supported by a six hour maintenance period on third shift. Our down time during the past year has averaged six hours per month on the analog and linkage equipment plus ten hours per month for the digital portion.

Down time is defined as that time between the notification of a maintenance technician and the appropriate repair.

We have found the Diagnostic Routines invaluable as a trouble shooting aid when the applications personnel have trouble with their simulations. The application engineers over the years have grown to trust our diagnostics and use them as they would a core dump to isolate problem areas. This we feel has been our greatest compliment.

REFERENCES

1 G H GALE
Boeing—Huntsville hybrid system
Simulation Vol 5 No 4 1965
2 T K SEEHUUS
Hybrid computer techniques to aid computer maintenance
Simulation Vol 7 No 5 November 1966
3 T K SEEHUUS M T BENNET
A hybrid oriented method for testing analog non linear equipment
Presented at a joint meeting of the Southeast and Midwest Simulation Councils January 1968 Miami Florida