

1995113223

N95-19639

USING EXPERT SYSTEMS TO ANALYZE ATE DATA

34074
p. 8

Jim Harrington
Honeywell Military Avionics
MS 673-4
13350 US Highway 19 N.
Clearwater FL, 34624
harrington@space.honeywell.com

ABSTRACT

The proliferation of Automatic Test Equipment (ATE) is resulting in the generation of large amounts of component data. Some of this component data is not accurate due to the presence of noise. Analyzing this data requires the use of new techniques. This paper describes the process of developing an expert system to analyze ATE data and provides an example rule in the CLIPS language for analyzing trip thresholds for high gain/high speed comparators.

INTRODUCTION

We are seeing a proliferation of "Simple" Automatic Test Equipment (ATE) based on personal computers. Large quantities of test data is being generated by these test stations. Some of this data will not accurately represent the true characteristics of the equipment being tested, particularly when the power supply in the personal computer is being used to power the ATE circuitry. This paper discusses a methodology for developing an expert system to examine the data files generated by the ATE. This expert system can be used to produce a data file containing the "most probable" data values with the effect of power supply noise removed. These "most probable" data values are based on specific statistical processes and special heuristics selected by the rule base.

THE NEED FOR A NEW APPROACH OF DATA ANALYSIS

Power supply noise can become a significant source of error during testing of high speed/high accuracy Analog to Digital (A/D) and Digital to Analog (D/A) converters (10-bits or greater) or high speed/high gain comparators. This power supply noise can cause:

- erratic data values from an A/D converter,
- wandering analog outputs (which can translate to harmonic distortion) from a D/A converter, and
- false triggers from a comparator.

A 10-bit A/D converter, optimized to measure voltages in the 0 to 5 V range, has a voltage resolution of 4.88 mV (i.e., the least significant bit--LSB--represents a voltage step of 4.88 mV). The 5 volt power bus on a personal computer (i.e., an IBM or third party PC) can have noise spikes in excess of 50 millivolts (mV). These noise spikes can, therefore, represent an error of greater than ten times the value of the LSB.

Even though the noise spikes on the PC power bus are considered high frequency (above a hundred Megahertz), high speed A/D converters and high speed comparators can capture enough energy from them to affect their operation.

The power supply noise is both conducted and radiated throughout the PC enclosure. Simple power line filters do not prevent noise from entering the ATE circuitry and affecting the tests

being performed. Much of the noise is cyclic in nature, such as that resulting from Dynamic Random Access Memory (DRAM) refresh and I/O polling. Other noise can be tied to specific events such as disk access.

The best method of improving the accuracy of the ATE is proper hardware design. Using power line filters does provide some noise isolation. Metal enclosures can also reduce susceptibility to radiated emissions. Noise canceling designs will also help. Self powered, free standing ATE which is optically coupled to the PC is probably the cleanest solution (from a power supply noise perspective). However, there are cases when an existing design must be used, or when cost and/or portability factors dictate that the ATE be housed within the PC. In these situations, an expert system data analysts can be used to enhance the accuracy of the test data.

The Traditional Software Approach

The traditional software approach to remove noise from a data sample is to take several readings and average the values. This approach may be acceptable if the system accuracy requirement is not overly stringent. A procedural language such as FORTRAN would be a good selection for implementing a system that just calculates averages. If A/D or D/A linearity is being tested, the averaging approach may not be accurate enough. The problem with this particular approach is that all of the data values that have been "obviously" affected by noise (as defined by "an expert") are pulling the average away from the true value.

The Expert System Approach

An expert system could be devised to cull out the "noisy" data values before the average is taken, resulting in a more accurate average. The problem is to develop an expert system that can be used to identify when noise might be effecting a data value.

THE PROCESS

The first step in developing an expert system to remove noisy data from the calculations is to define how the noise affects the circuits. For example, a voltage ramp is frequently used to test the linearity of an A/D converter and the trip points of a comparator. How does noise on that voltage ramp affect the performance of the circuitry being tested?

As a voltage source ramps down from 5 volts to 0 volts, noise can cause high speed A/D converters to output a data set with missing and/or repeated data values. Figure 1 shows the possible effect of random noise on the output of an A/D converter; the data line in this graph represents the source voltage that has been corrupted by noise and then quantized to represent the output of an A/D converter. Even though the general trend may be linear, the actual data is not.

Noise on a 0 to 5 volt ramp can cause a high speed comparator to change states too early or too late. Another common effect of noise on a comparator is a "bouncy" output (turning on and off in quick succession before settling to either the "on" or "off" state). Figure 2 shows the effect of noise on the output of the comparator.

The next step is to identify how the noise appears on the output of the equipment being tested. As seen in Figure 1, noise can cause runs of numbers with the same output from the A/D converter (e.g., three values in succession that the A/D converter interprets as 4002 mV). We also see missing data codes (as in the run of 3919, 3919, 3933, 3933--repeating the data values 3919 and 3933 but missing the data values 3924 and 3929 mV). Figure 2 shows an unexpected

“on” and “off” cycle from the comparator before the voltage ramp reached the true trip point of 3970 mV. An additional cycle to the off state occurs at the end of the graph as seen in the comparator output going to 0 V, it should have stayed in the high (+5 V) state.

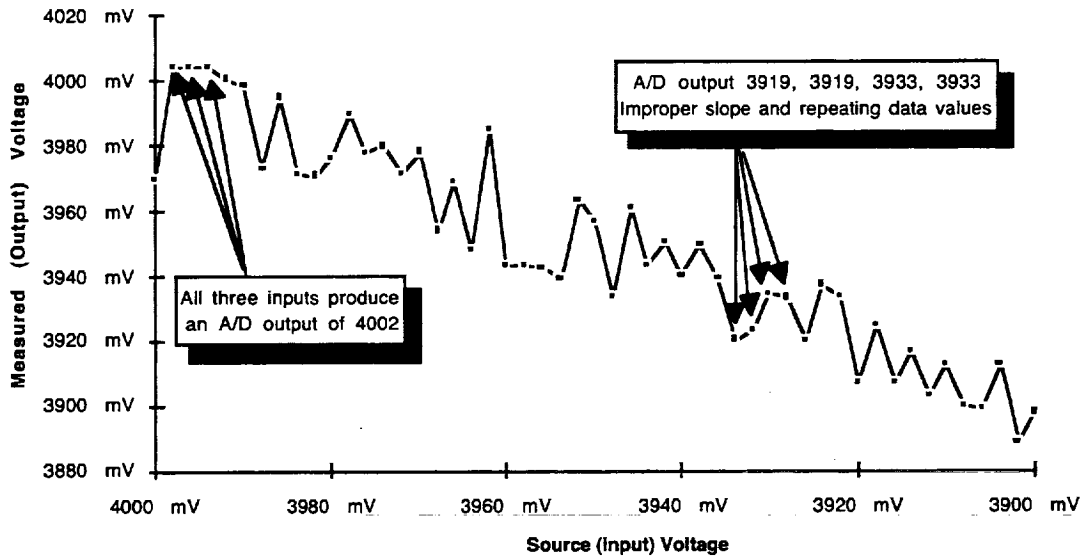


Figure 1. A/D Output Errors Caused by a Noisy Voltage Source

For brevity, the rest of this paper focuses on the comparator example; the approaches discussed for the comparator are directly applicable to the A/D converter as well.

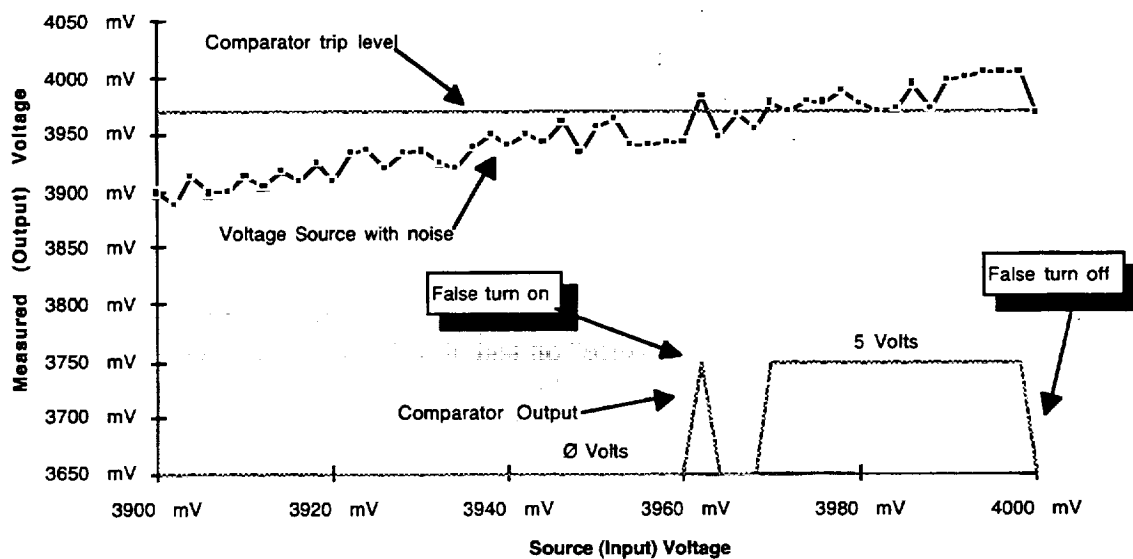


Figure 2. Comparator Output With a Noisy Source Signal

In most test set-ups, there will be no synchronization between any cyclic noise in the PC and the data acquisition process because most computer systems fill a buffer before dumping the data to the disk drive (this is true unless the code generated for data acquisition goes through the specific process of “read data” “forced write to the disk” “read data” “forced write to the disk” etc.). This means that the cyclic noise will affect the data at random times during the data taking process. The non-cyclic random noise will, of course, appear at random times. Therefore, if we repeat the test multiple times, the noise should manifest itself at different places in the test sequence each time.

The noise shown in Figures 1 and 2 is random, with peaks as high as 50 mV. These plots were constructed to demonstrate the effect of noise on the circuitry. Under normal conditions, with the test voltage generated in a PC, the actual source voltage would look like the data shown in Figure 3. The “Source Voltage ramp” in Figure 3 shows noise spikes as high as 65 mV at regular intervals. The area between the noise spikes has some low-level asynchronous noise. The results of the combination of low level random noise and high level cyclic noise on a comparator is shown at the bottom of the graph (the voltage trip level is still set at 3970 mV).

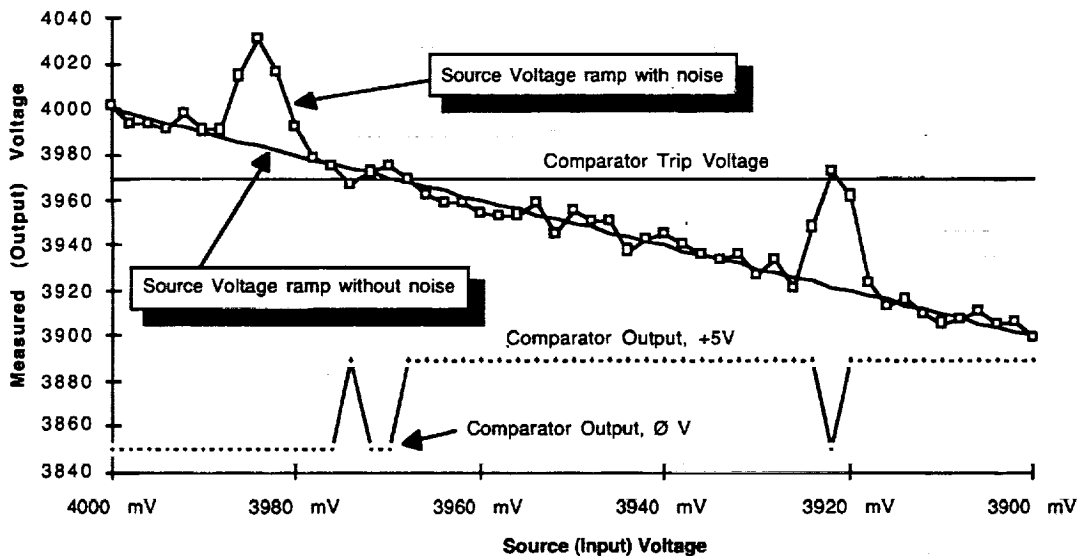


Figure 3. Noise Waveform During Test

Table I lists the voltages at which the comparator output shifts between 0 V and 5 V. The table also shows the voltages of the initial “bounce” on and off (“Off to On #1” and “On to Off #1”), and the inadvertent bounce off and back on (“On to Off #2” and “Off to On #3”). A basic set of rules starts to emerge when this noise spectrum is combined with knowledge of the effect of noise on the data taking process.

DEVELOPING THE RULE SET

Three sets of data were accumulated for each threshold analysis. These data sets were separated in time by only seconds as the ATE reset itself; this meant that the basic environmental factors were fairly consistent (an important factor when using any statistical process on data). The following rules are shown in order of precedence, from most important to least important.

Saliency levels were used to ensure the most important rule would fire before a rule of lesser importance.

Once the optimum data value is selected by a rule, the data is written to an output file. The old data is then removed from the fact list to prevent a rule of lower precedence from firing and changing the data value again.

Table I. Comparator Input Threshold Voltage Test

<i>Comparator ID</i>	<i>Off to On # 1</i>	<i>On to Off # 1</i>	<i>Off to On # 2</i>	<i>On to Off # 2</i>	<i>Off to On # 3</i>
001	3974	3972	3968	3922	3920
.					
.					
.					

Rule 1

The first rule developed was one of common sense. If the comparator does not show any bounces in any of the three data sets, and the data values are identical, that value should be used.

Rule 2

The second rule is similar. If two of the three data sets contain no bounces and the comparator output changes from \emptyset V to 5 V at the same input ramp voltage, the data value found in those two data sets is used. This rule takes effect (is instantiated) regardless of the number of bounces in the third data set (i.e., the third data set could contain a single trip level or 3 trip levels--as in Figure 3--or 100 trip levels). In mathematical terms this is called selecting the mode of the data (i.e., the most commonly repeated value in a data set [1]).

The second rule handles the possible situation of the noise spike occurring just prior to the time the true source voltage exceeds the trip level. The noise spike would cause the comparator to trip early. The philosophy behind the rule is that the high level noise spikes seen in Figure 3 can cause the comparator to switch early in one data set, but the probability of the noise spike affecting two data sets the same way is extremely low.

This rule will also fire if none of the three data sets have bounces, and the values are the same. If both rule 1 and rule 2 fire, the trip level will be recorded in the output file twice. Repeated data in the output file can cause other data analysis programs to have problems. This is the reason the old data (the data used to infer the optimum trip level) is removed by each rule that fires; so that the rule with the highest precedence will remove the trip level data, preventing a secondary rule from also firing.

Rule 3

The next rule handles the case where no bounces are detected, but the trip levels in the three data sets are all different. In this situation, it is difficult to determine which data set contains good data and which might contains noisy data. We decided to average the two closest together

values. This averaging represents the effect of finding a trip level that is between two steps in the voltage ramp (e.g., the addition of low-level noise causing the comparator to trip at one level one time and another level the second time). If all three data values are equally dispersed, the average of all three values is taken. The mathematics used to implement this rule in effect takes the median of the data when all three data sets were used (an easy way of selecting the average of three equally dispersed numbers).

In hindsight, another method of approaching the same rule is to track the interval of the 65 mV noise spike. By tracking the last occurrence of comparator output bounces, the period of the large cyclic noise spike can be determined (this approach requires an additional pass through the data to determine the period of the cyclic noise). The period of the cyclic noise can be used as an additional constraint, and the data set(s) that are closest to the middle of their periods can be averaged.

At this point, we have covered all of the easy rules. The following rules rely more on heuristics developed by the project research team (i.e., the domain experts).

Rule 4

The fourth rule covers the case when two of the data sets contain no bounces, but the data values are different, and the third file contains bouncy data. In this situation, we decided to average the data from the two non-bouncy data sets. An additional precaution was added to this rule in that if either of the two values is more than $\pm 5\%$ from the average value, the operator is notified of the comparator being tested and the two trip voltage levels.

Rule 5

If only one of the data sets doesn't contain bouncy data, the data value of that one data set is used. This rule covers the case when "the experts" feel that two of the data sets are affected by some high level noise spikes. It is "felt" that the one file non-bouncy data set contains the most accurate trip level on the most frequent basis.

Rule 6

The final case is when all three data sets contain bounces. This means that all three data sets have been affected by noise. In Figure 3, the noise spikes are only one reading increment wide. Both the on-off cycle and the off-on cycle are in the inverted state for one data reading. The difference between the first two data values in Table I is 2 mV; this is also true for the last two data values. The difference between the second and third data values is 4 mV, and the difference between the third and fourth data values is 46 mV. The comparator settles to the "On" state at 3968 mV. The rule that forms out of this analysis is to select for the trip level (when the comparator turns "On") the voltage that has the greatest difference between it and the adjacent "Off" voltage (e.g., in Figure 3, the trip level would be selected as 3968 mV, the point which starts the 46 mV span before the next "Off" spike).

The average of all 5 data values is 3951.20 mV; an error of 16.8 mV compared to the 3968 mV trip level. The average of the 3 values which cause an "Off to On" transition of the comparator is 3954 mV; an error of 14 mV. The increase in accuracy is easily seen in this example. But, we are not done yet. What if all of the transitions of the comparator occur at equal voltage intervals?

There is no clear cut heuristic when all of the "Off to On" and "On to Off" data values are evenly spaced. The system we developed defaulted to the first "Off to On" transition. Depending on the system implementation, the middle "Off to On" transition may be the best selection.

The coding of this rule is a little tricky. The code in Figure 4 shows how this rule could be implemented in CLIPS.

```
(defrule Compromise "This rule finds the most probable data value out of a set of bounces"
  (Evaluate_Data) ;flag fact that allows the compromise rule to function
  ?Orig_Data <- (?set&set_1set_2set3 $?data) ;get the data
  (test (>= (length $?data) 2)) ;test for bounces
=>
  (retract ?Orig_Data) ;Get rid of the data set that is to be replaced
  (bind ?elements (length $?data))
  (if (evenp ?elements)
    then
      (assert (?set 0)) ;Bit ended in original state--no data
    else
      (bind ?delta 0) ;Set up the variable to find the largest difference
      (bind ?y 2) ;Pointer to the first even number (position) data value
      (while (<= ?y ?elements)
        (bind ?x (- ?y 1)) ;Pointer to the position preceding ?y
        (bind ?z (+ ?y 1)) ;Pointer to the position following ?y
        (bind ?test_val (abs (- (nth ?x $?data) (nth ?y $?data))))
        (if (> ?test_val ?delta)
          then
            (bind ?pointer ?x)
            (bind ?delta ?test_val)
          )
        (bind ?test_val (abs (- (nth ?y $?data) (nth ?z $?data))))
        (if (> ?test_val ?delta)
          then
            (bind ?pointer ?z)
            (bind ?delta ?test_val)
          )
        (bind ?y (+ ?y 2)) ;Increment ?y to the next even position
      )
      (assert (?set =(nth ?pointer $?data))) ;put the new data on the fact list
    )
  )
)
```

Figure 4. CLIPS Code Implementing Rule Number 6

The actual searching of the data set is performed in the "while" statement found on the right hand side of the rule (the "then" part of the rule, found after the "=>" symbols).

Rule 6 is performed on all three data sets separately. Remember that this rule is invoked only when bounces are seen in all three data sets. This point helps to alleviate some of the worry surrounding the case where all of the trip levels are equally dispersed. Once all three data sets are evaluated and a single voltage level is selected, the three new values are put on the fact list to be operated on by the first 5 rules. Only rules number 1 through 3 actually apply since none of the data generated by rule 6 would contain bounces.

CONCLUSIONS

This process demonstrates a new approach to analyzing data taken by ATE. Increases in accuracy of 14 to 17 mV is demonstrated in rule 6. The other rules select the heuristic or statistical procedure used to determine the best data value from the multiple data sets provided. This paper illustrates the method of developing a set of rules, implementable in CLIPS, for defining the presence of noise in a data set and for removing the noisy data from the calculations.

ACKNOWLEDGMENTS

I would like to thank Richard Wiker and Roy Walker for their diligent work in helping to define the heuristics involved in the data analysis.

REFERENCES

1. Spiegel, Murray R., "Chapter 3: The Mean, Median, Mode and Other Measures of Central Tendency," SCHAUM'S OUTLINE SERIES: THEORY AND PROBLEMS OF STATISTICS, McGraw-Hill Book Company, New York, 1961, pp. 47 - 48.