Test Data Reduction Using PEPSE® in (Almost) Real Time

John Mihalich, Jr.
Marcus B. Caudill

Tri-State Transmission & Generation Association, Inc.

Test Data Reduction Using PEASE in (Almost) Real Time

John Mihalich, Jr.
Results Engineer
Tri-State Generation and Transmission Association
P.O. Box 1307
Craig, Colorado 81626
(303) 824-4411

Marcus B. Caudill
Performance Engineer
Tri-State Generation and Transmission Association
P.O. Box 1149
Montrose, Colorado 81402
(303) 249-4501

ABSTRACT

Field verification of performance test data is critical to the quality of test results. Various methods of calculating results have been employed, ranging from hand calculations to exotic custom programming for on-line calculations. Hand calculations are problematic because the time required usually prevents field verification and calculation methods are not always consistent from person to person. Online calculations are much faster, however, the difficulty of programming the required steam table routines usually requires the programmer to resort to curve fits or nonstandard steam tables. This can cause initial results that deviate significantly from the more detailed final results. To provide immediate results of the same quality as the final results a better method is needed. The Tri-State solution is a combination of custom computer programming for data collection and manipulation used in conjunction with the industry standard PEPSE® thermal analysis software.

Using multiple personal computers for data collection and analysis, test data can be collected, converted, averaged, basic corrections applied, and formatted as a PEPSE® Special Input Processor. This can then be prepended to a base PEPSE® deck for the analysis. With 486 based computers the data manipulation, processing and results presentation can be done on a five minute cycle. Our paper presents the evolution of this process at Tri-State, from hand calculations, to the current version of online PEPSE® to the Tri-State plans for the future.

INTRODUCTION

Tri-State Generation and Transmission Association is the Operating Agent for the 1350 MW Craig Generating Station located in Northwestern Colorado. In 1990 an aggressive performance improvement program was initiated by the new performance staff. One of the priorities set by the new staff was the development of repeatable methods for collecting and analyzing test data, both for annual precision testing and for the monthly testing which is the basis for early detection of performance degradations. The test methods in place up to 1990 were not conducive to the more accurate and more frequent testing that we required.

PRE-HISTORY: THE DARK AGES

Existing data collection and analysis methods relied on many manually collected data points, requiring significant manpower to yield limited results. A limited set of precision test data was collected using computerized data acquisition, however, the acquisition software only provided data in hard copy format. Paper logs were generated by the plant computer, requiring hand averaging of all data, and all test data analysis was performed by hand. So not only were ten people required for collecting test data, but ten people were also required to decide if the data just collected was valid! After the lengthy data reduction process, only HP and IP turbine enthalpy drop efficiency, net unit capability (capacity), and input-output net unit heat rate were reported. The only corrections applied to the results used the vendor's deviation curves. It was time for a change.

1990: THE RENAISSANCE PERIOD

In order to progress into a modern era, it would be necessary to make significant improvements to the data acquisition software and hardware. The controller portion of data acquisition platform consisted of a Hewlett-Packard Vectra PC with an HP-82321 measurement co-processor (HP Viper Board) installed. The Viper Board is essentially an HP-9000 series 200 computer on a board that can be installed in a PC card slot. The Viper Board communicates with test instrumentation via its own HP-IB (IEEE-488) port and shares the disk drives, keyboard, I/O ports, and CRT with the PC. The Viper has true multi-tasking capability within the PC since it can run its programming concurrently with DOS applications. The down side of the Viper Board is its initial cost per unit; approximately $5K, fully equipped, and with process speed significantly slower than the PC's.

Since the performance program was expanding rapidly and additional computing power was required, a lower cost and more flexible alternative to the HP Viper board was needed. Trans-Era HT Basic provided that solution. HT basic allows HP 9000 series 200-300 Rocky Mountain Basic programs to be run on a PC compatible with few changes. The installed cost per machine is about 25% of a Viper Board. The decision was made to migrate to HT Basic, completely leaving the Viper platform.

The field portion of the data acquisition platform consisted of an HP-3852 Data Acquisition Unit, a Ruska pressure transmitter, sequencing valves and miscellaneous thermocouples and RTD's.

The software, as it existed at that time, was rather simplistic. There was no provision for electronic storage of test data or for real time feedback of test information that could be used for monitoring test/unit stability. In addition, all test configuration data were hardcoded within the software and had to be changed for each different test. Test duration was fixed and was not extendable after the test start. Rather than attempt to modify the existing software or develop new software from scratch, a decision was made to use data acquisition software that had been developed by others and modify that package to suit the Tri-State Generation and Transmission needs.

At the heart of the Tri-State software package are the "ACQ" and "APLOT" programs. "ACQ" is an online program that does the collection, conversion, presentation and storage of test data. The "APLOT" program is an offline plotting and data management package that manipulates data that has been stored to disk by "ACQ". The "ACQ" and "APLOT" programs have none of the shortcomings that the original software contained. These programs are menu driven, flexible and quite user friendly. Provisions were added to the online programming for water leg corrections, individual thermocouple calibration curves and multiple test configuration directories.

During this period, significant improvements were also made to the process measurement hardware. Rosemount transmitters replaced the Ruska and the sequencing valves for all of the critical pressure measurements on the HP and IP turbines. Rosemount transmitters replaced the Heise transmitters that were used to monitor condenser pressure. A second HP-3852 was added at the LP turbine skirt to pick up crossover pressures and temperatures as well as condenser pressures.

With all of the critical parameters being measured with high accuracy instrumentation, our next major step required a repeatable method for data analysis. After initially setting up some stop-gap spreadsheets to calculate HP and IP turbine efficiency, N2 packing leakage, and perform simple corrections, we turned our efforts toward the development of detailed PEPSE® models.

PEPSE® Model Development

Generalized PEPSE® models using GE turbines (Types 4-7) were already available for the Craig Units, and after some additional fine tuning, the existing models were used as the basis for test data reduction models. These new models were constructed in a conventional manner, with the first model a test data reduction model, the second a correction to standard conditions model, and the third as a component degradation model.

The first step in the process of creating a PEPSE® test data reduction run began with data collection. Data was collected from HP3852 Data Acquisition Units (DAU), from the Bailey 1055 plant computer, from various electric power meters, both digital and spinning type, along with other miscellaneous data. All data was collected for the test period (typically one hour), hand averaged, and then manually input into the PEPSE® model. The Special Input Processor was used to help minimize input errors and inconsistencies from run to run and test to test. The Special Output Processor was used to output all of the desired results, in addition to the parameters required for the correction to standard conditions run.

The test data model uses Type 8 turbines with solution methods requiring inputs of shell pressure and shell temperature (variable IPCASE = 6), or shell pressure and shell enthalpy (variable IPCASE = 5). Once this model was constructed and verified, the modifications for correction to standard conditions were begun.

For correction to standard conditions, turbine components were converted to solution methods requiring inputs of pressure ratios, flow coefficients and stage group efficiencies (variable IPCASE = 1 and 3). Reference conditions for the throttle valve and the HP, IP and LP turbines were manually transferred to the standard conditions run. In addition, as tested heater TTDs and

DCAs were input to the standard conditions run. In all, over 40 pieces of information were carried over from the test data run to the standard conditions run. This represents over 40 additional opportunities for data entry errors.

The purpose of the component upgrade runs is to determine the operating penalty for turbine cycle components that are performing at off design conditions. This is done by substituting the best achievable performance for an individual component for the as-tested performance. For example, as-tested HP turbine section efficiency might be 83 percent and design (or acceptance) HP turbine section efficiency might be 87 percent. By substituting the 87 percent design efficiency for the as-tested 83 percent in the PEPSE® model, the heat rate deviation, and consequently the operating cost, for the degraded component can be calculated.

At Tri-State, the primary areas of concern in the turbine cycle are the HP and IP turbine sections, the condenser, and the feedwater heaters. To individually quantify the losses in each of these areas required five additional PEPSE® runs, with a total of more than 40 pieces of manually changed input data. Again, over 40 additional opportunities for data entry errors. We had to find a better way.

Automating the Data Transfer

One of our initial goals in improving the data handling and processing was to have the ability to verify test data in the field and compare back to historical data. Because of this goal, the ability to rapidly correct to standard conditions was an important item. Therefore, early efforts were focused on passing information between the PEPSE® test data reduction model and the standard conditions model.

PEPSE® has tremendous flexibility in transferring data from one model to the next. Information can be passed in a number of ways using Save Case and Change Case. In the interest of faster run times, we chose to begin with save case and build from there.

From our initial modeling efforts, we knew what items had to be transferred from Run Number One to Run Number Two. These items are listed in Appendix A. The easiest way to transfer these values was using operations and operational variables. Each value being transferred is a PEPSE® output variable, and as such will be overwritten in the first iteration of the Save Case. To prevent this, each value required for the standard conditions run was set equal to a unique operational variable (OPVB). For example, the throttle valve reference pressure, which is the pressure in stream 10, was saved as follows:

```
881520  PP,  10,  EQL,  OPVB,  152
```

Because Save Case carries operational variables forward to the next case, the value of throttle pressure from the test data reduction run is preserved as OPVB(152). All of the other variables contained in Appendix A were carried forward in a like manner. All of the desired information was carried forward, however, undesirable information was also being carried forward. The undesirable information had to be deleted.

All of the test data was input to Run Number One using the Special Input Processor. Except for the boundary condition inputs, these Special Inputs had to be copied into Run Number Two and have delete cards (89XXX8 DELETE) added. The boundary condition cards were copied into Run Number Two and the special input values were changed to the standard conditions values. In addition, the straight expansion line was specified in the base run, and this option was deleted in the standard conditions run. Now, the required changes to generalize the model must be implemented.

Just as in the manual data transfer model, the turbine components must be converted to solution methods requiring inputs of pressure ratios, flow coefficients and stage group efficiencies (variable IPCASE = 1 and 3). The results of the test data reduction runs are input from operational variables in the reverse of the manner in which they were saved. For example, the throttle valve reference pressure saved in the previous example in OPVB(152) would be reinstalled as follows:

```
881520  OPVB,  152,  EQL,  PREF,  40
```

Note that the same operation (881520) is used in both runs, negating the need to delete this operation. This completes the construction of the correction to standard conditions run.

Extension of the data transfer concept to the component upgrade step is simply a matter of changing the stored as-tested values for efficiency, pressure ratio, flow coefficient, TTDs and DCAs. This is done simply by using the appropriate operational variables and adding an additional save case.

At this point in the model development, we had decreased the possible number of input errors due to typos by about 2/3. While this was a significant decrease, it still wasn't good enough.

Data Input Using LOTUS 123

Using a spreadsheet to transfer data to a PEPSE® input deck is not a new idea. A number of papers have discussed this subject (References 2, 3 and 4) so we will not discuss the details here. The spreadsheet has the advantage of being able to average raw data, apply calibration or barometric pressure corrections, and perform simple data pre-processing. The raw data can also usually be imported directly from an ASCII file, which minimizes the hand input data. Using this method, we can quickly obtain results within a short period of the test end. Our only complaint was the need to switch between multiple software packages. Since this was only an intermediate step on our way to real time PEPSE®, we could live with the software switching.

1992: THE INDUSTRIAL REVOLUTION

The last major link in the performance testing paper chain was the plant computer logs. A program was written in HT Basic to configure a PC so that the PC appeared to the plant computer as a line printer. The Bailey 1055 plant computer could continue to output its logs, on a five minute schedule, in its normal operating mode. The PC was programmed to gather all of the output into memory and then after a 30-second timeout period, it would begin to break the Bailey 1055 data dump apart.

Since the log format is fixed, as output from the Bailey 1055 computer, it is quite easy to search the output for key word groups that identify individual logs or other messages. In our case, the three key words that identified a valid log were "LOG xx", "PAGE y" and "- z" where x, y, and z are integers that designate the log number and page number y of z. If these three key words were found on one line, then a valid log was found and the program begins to search subsequent lines for the next key word which is ":". The colon is located in the time stamp for the data line. The data is arranged after the time stamp in groups of twelve on pages number one and two and in groups of eight on page number three. Once the log number and page number are known the data can is broken out by position number on the line and its value subsequently assigned to a variable name.

After the data line is broken, the program begins to search for the next log. This search process is continued until the last line of the data dump is examined. At the completion of the search, any variable that was bad quality or any variables not found are assigned a value of "-666". Because of program constraints, the bad quality value had to be a real number. The value "-666" was chosen because we could not foresee this value ever occurring at our facility in normal operation and even the untrained would look upon such a value with some suspicion.

When all of the plant computer logs were broken, and the values assigned to variables, the Plant Computer Interface PC would send a service request (SRQ) over the HP-IB (IEEE-488) link to the main system PC. Handshakes would be exchanged and the plant computer data would be transferred. The Main PC would then save all of the test data, for that scan, to disk in a single unified data set.

At the completion of testing, the "APLOT" program would be started, the test data would be loaded and processed. "APLOT" has the capability of averaging a block of data as specified by start and stop times. This averaged block of data is then passed to a sub-program that does pre-processing. This pre-processing consists of converting all pressures to absolute pressures, calculating sums and averages, converting units to PEPSE® consistent units, and like processes. Condenser cleanliness is also calculated at this time using HEI procedures. After the pre-processing is complete, a PEPSE® deck is generated by the sub-program.

To generate the input deck the user is first queried for the desired output file name. An ASCII file is then created of appropriate size. Next the data input statements are generated by reading the PEPSE® line number, PEPSE® component variable name and PEPSE® component number from a file and adding the appropriate variable in the following format:

```
89XXX1,      AAAAAA,   BBB, DDDDDDDDDD.DD,  I

where     XXX              is the card number
          AAAAAA           is the component variable name
          BBB              is the component number
          DDDDDDDDDD.DD    is the variable value
```

After each input line is composed it is output to the Test ASCII file. The process is repeated for the desired number of input lines. When all input lines are stored, the file that contains the remainder of the base deck is read and then stored to the Test ASCII file on a line by line basis. It is now necessary to return to the DOS environment to run PEPSE®.

After PEPSE® executes, it is necessary to return to the HT Basic environment to run a routine that searches the PEPSE® output deck for the desired output. When this output is found it is pulled out of the file, assigned to a variable and finally stored in a "Results" ASCII file. HT basic is stopped and Lotus 123 is started. The results file is imported into the spreadsheet and finally, charts, graphs and reports are generated for the test report.

To an outside observer, this process may seem rather involved, however, one must realize that at no point is any data handled by hand and the whole process can be completed in less than 15 minutes.

1993: THE SPACE AGE - PEPSE® IN PSEUDO-REAL TIME

The key to successfully running PEPSE® in real time has always been the ability to call PEPSE® from within another program, execute, and return to the original program. Because of memory manager conflicts, this has historically been a problem. What was needed was a multi-tasking environment.

Consideration was again given to the HP Viper board. Since it already had an integral HP-IB interface and operated independently of DOS applications, it had a very good chance of succeeding. Elementary tests showed that PEPSE® could indeed be called from the Viper, execute completely and correctly and return control to the Viper environment. A third PC, with a Viper, was added to the test data acquisition system and modifications were made to the "ACQ" program to allow data to be transferred on demand to the Viper via the HP-IB bus. New software was written for the Viper to import and pre-process the data, build the base deck, execute PEPSE®, extract the results from the output deck and finally, present the results on the CRT. From the moment of data request, to the reporting of final results, approximately five minutes elapses.

THE FUTURE: REAL PEPSE® IN REAL TIME

One of the more troublesome problems encountered throughout our quest for on line test data reduction was the unavailability of a suitable multi-tasking environment to handle all data processing and manipulation on a single computer. The majority of this problem was caused by the inability to run PEPSE® simultaneously with other applications in the various environments. With the recent release of PEPSE® Version 58, our problems have been solved to some degree. PEPSE® will now run in Microsoft Windows 3.1, 386 Enhanced Mode. As long as PEPSE® is the only DOS application running, you can even run PEPSE® in the background, with other Windows applications, such as data acquisition or presentation graphics, in the foreground. This has opened a whole new realm of possibilities.

The brightest of the currently available possibilities is Hewlett-Packard Instrument Basic for Windows. It provides a platform that allows the use of currently developed software to be used in the Windows environment with minimal changes. However, at this time, only revision 0.0 is available and the release is full of bugs that make its use difficult at best. The Revision 1.0 should eliminate these problems and provide a suitable platform. Until the release of the next revision, we are left to consider the possibilities of Windows NT. A beta copy of NT has been acquired but the time to investigate the possibilities has not been allocated.

So the final question becomes, "Where do we go from here?".

The possibilities of using the Windows DLL (Dynamic Link Library), off the shelf Windows software and some custom programming with the industry standard PEPSE® computer program to form an online non-proprietary performance monitor demands some contemplation. Presentation graphics packages, data bases, spreadsheets and word processors are all readily available for the Windows environment, and minimal integration would be required to provide the graphic front end, archival capabilities, and reporting functions that are desired in an online monitoring system. Custom data acquisition software would allow this system to be interfaced to any number of different plant monitoring computers or stand alone data acquisition systems. For utilities with PEPSE® models already in place and limited funds, this would be a much better alternative to a fully "home grown" monitoring system.

All of the items that form the basis of a good performance monitoring system also form the basis of an excellent performance testing front end. Windows features can be used to produce standard format reports, with all data transferred using the DLL. Especially in the case of monthly testing, minimal changes in report wording are required. A section for comments and test notes could be set up for online entry, with the comments entered into the final report. The possible time savings here is tremendous.

PEPSE® in real time is real. At Tri-State, we continue to strive for better methods to collect, analyze and use performance data, and we plan to continue the evolution of our performance program. As we implement our ideas for the Windows NT platform, we will continue to evolve and integrate additional ideas and methods. And as technology continues to move forward, we're sure to find better ways, so we'll keep you posted......

REFERENCES

1. PEPSE® Manual: Users Input Description, Volume 1, Version 58; HALLIBURTON NUS Corporation; 1993

2. "A Method for the Collection and Analysis of Performance Test Data"; R.D. Griebenow, L.A. Morris, M.B. Caudill; Proceedings of the 1989 Performance Software Users Group Meeting; EI International; 1989.

3. "Utilizing LOTUS 123 In Conjunction With PEPSE®"; D.L. Hill; Proceedings of the 1990 Performance Software Users Group Meeting; EI International; 1990.

4. "Integration of PEPSE® With a Turbine Testing System"; W.J. Kinash; Proceedings of the 1991 Performance Software Users Group Meeting; HALLIBURTON NUS Corporation; 1991.

5. "Users Guide, LOTUS 123 for Windows Release 1.0"; LOTUS Development Corporation; Cambridge , MA; 1991

6. "Trans-Era HT Basic Users Guide"; Trans-Era Corporation; Provo, UT; 1990

7. "Trans-Era HT Basic Reference Manual"; Trans-Era Corporation; Provo, UT; 1990

8. "HP Instrument Basic for Windows"; Hewlett-Packard Corporation; Palo Alto, CA; 1992

APPENDIX A

PEPSE Test Data Reduction Run Results

Description	Variable Name	Component ID
Throttle Valve Reference Conditions		
Throttle Valve Inlet Flow	WW	30
Throttle Valve Inlet Enthalpy	HH	10
Throttle Valve Inlet Pressure	PP	10
HP Turbine Section		
Governing Stage Efficiency	EHPP	100
Governing Stage Flow Coeff	FLOWCU	100
HP Stage Group Efficiency	EHPP	120
HP Stage Group Pressure Ratio	PRTURB	120
HP Reference Conditions		
HP Section Inlet Flow	WW	50
HP Section Inlet Enthalpy	HH	5
HP Section Inlet Pressure	PP	50
IP Turbine Section		
1st IP Stage Group Efficiency	EHPP	230
1st IP Stage Group Flow Coeff	FLOWCU	230
2nd IP Stage Group Efficiency	EHPP	240
2nd IP Stage Group Pressure Ratio	PRTURB	240
IP Reference Conditions		
IP Section Inlet Flow	WW	220
IP Section Inlet Enthalpy	HH	220
IP Section Inlet Pressure	PP	220
LP Turbine Section		
1st LP Stage Group Efficiency	EHPP	300
1st LP Stage Group Flow Coeff	FLOWCU	300
2nd LP Stage Group Efficiency	EHPP	310
2nd LP Stage Group Flow Coeff	FLOWCU	310
3rd LP Stage Group Efficiency	EHPP	320
3rd LP Stage Group Flow Coeff	FLOWCU	320
4th LP Stage Group Efficiency	EHPP	330
4th LP Stage Group Flow Coeff	FLOWCU	330
Last LP Stage Group Efficiency	EHPP	340
Last LP Stage Group Pressure Ratio	PRTURB	340
LP Reference Conditions		
LP Section Inlet Flow	WW	250
LP Section Inlet Enthalpy	HH	250
LP Section Inlet Pressure	PP	250
Feedwater Heater TTDs and DCAs		
A Feedwater Heater TTD	TTDOUT	650
A Feedwater Heater DCA	DCAOUT	650
B Feedwater Heater TTD	TTDOUT	640
B Feedwater Heater DCA	DCAOUT	640
C Feedwater Heater TTD	TTDOUT	630
C Feedwater Heater DCA	DCAOUT	630
E Feedwater Heater TTD	TTDOUT	530
E Feedwater Heater DCA	DCAOUT	530
F Feedwater Heater TTD	TTDOUT	520
F Feedwater Heater DCA	DCAOUT	520
G Feedwater Heater TTD	DCAOUT	500