For documenting of software routines prior to starting on a project, a copyright is the best and easiest way to establish the condition of the software on a certain date. This report was written to fully show the feature set and level of development of the software going into negotiations for the contract. The graphic images are actual screen images captured from the PC monitor while the program was running. Presentation here is ordered in the way the screens appear during program operation.

I. Startup Menu:

The startup menu calls the four areas of software development by clicking the appropriate button.

II. Registration & Security Access (Keyboard and Mouse input, Password array, CodeEnable) this routine prevents piracy of the program and prevents unauthorized access to the underlying data.

Figure 1 Registration and Security Panel
All of these data are stored to disk each time an update is made to the display. This is to allow the Optimizer to resume right where it stopped after power loss, with minimum loss of data.

The left block of information sets the Test Supervisor or Test Engineer’s name and other information as deemed appropriate by the customer. The time & date box updates continuously when measuring, or static when showing logged data.

The center block of buttons is the registration information with the code-key password generation utility and operator passwords to prevent software piracy.

The rightmost boxes set access to setup and data. Each authorized operator’s name is entered into the registration boxes, and a password selected and entered.

This password will be required each time to make any inquiries or changes to setup, operational or display functions and features.

Each data box uses the same Dialog Box to input requisite information in controlled manner as shown below:

**Figure 2 Registration Change Dialog Box**

### III. Autocal and Test ID Panel.

**AutoCal** (Keyboard and mouse click inputs, Calibration Array Output) This is a zero & span calibration routine. It allows the operator to set an input to an analog input of the computer, and then a keyboard entry sets the corresponding engineering units for that
voltage. The input voltage is moved to the other end of the measurement span, and then another keyboard entry sets the corresponding engineering units. The program computes the offset and gain to convert the voltage measured by the I/O board to engineering units defined by the two keyboard entries.

The upper left box is the Manufacturer Splash Panel. The array of buttons across the top are a utility to evaluate and control the level of confidence using methods from ASME PTC-18 and other sources, and constants for calibrations and dimensions for energy equations. The large array on the bottom is the individual channel calibration data.

The autocal routine works like this:

1. Set a low value on the input. When it is low enough (below 10%), the box for HiEU turns yellow.
2. The Volts values are a live display that updates periodically.
3. Clicking on the yellow Flow HiEU button calls the calibration input box:

![Figure 4 Autocal Input Dialog Box](image)

**Figure 4 Autocal Input Dialog Box**
III. StripChart Inner Loop

The following is a detailed step-by-step for the data capture and graphing portion of the program as it collects data and evaluates for “SteadyStateNess.”

Raw data is collected by the StripChart program. Voltage values are stored at maximum speed into an array. The program then analyzes the snapshot data to determine if the unit was operating steady state or not. The program constantly monitors operation, capturing and analyzing blocks of data then storing away any steady state values that are found. When a new data point is found, the program will redraw the desired output graphs.

1. Measure 1 to 5000 samples on 1 to 16 channels at maximum rate with the plug in I/O board in the PC. (For this demonstration, 8 channels were used. To add more channels, add more hardware and extend the for/next loops that step through the measurement and calculation sequences.)
2. If ShowRaw.backcolor = yellow then print the individual data points.
3. If CallMagnify.backcolor = yellow then use most of screen for selected channel
4. PreSet Max and Min values for X and Y channels.
5. Step through every measurement set, comparing the stored values with the max and min registers. Replace Max or Min with new values from the array.
6. If Value < Min then Min = Value. If Value > Max then Max = Value.
7. Add or subtract undershoot values to give some white space above and below the data set. Set window to printer dimensions and divide Y axis into 8 sections.
8. Draw grids for each of Head, Tail, Gate, Blade, Flow, Power, BladeServoPressure and GateServoPressure.
9. Use Max and Min values to draw Y-axis scale on graph. X-axis scale is just the number of measurements = full span.
10. Plot Dots on the grids, Round robin distribution as shown in Fig 1.
11. Run LeastSquare routine to find Center of Gravity and slope of data cloud.
   a. Load X and Y values into intake arrays for linear regression
   b. Determine NumPoints and call CurveFit
Appendix II Software Report 10/20/03 page 5 of 5

c. Compute averages of X and Y values. Plotted point of average X and Average Y is the “center of gravity” of the data set.

d. To find slope and Y intercept:
   1. Sum1 = Sum1 + (X Value – XAverage) * (YValue – YAverage)
   2. Sum2 = Sum2 + (X Value – XAverage)^2
   3. Slope = Sum1/Sum2
   4. YIntercept = YAverage - slope * Xaverage
   5. Discard outliers and repeat steps 1-4.
   6. Compare standard deviation and slope to pre-defined limits from calibration table.
   7. Store MidX, MidY, Slope and Standard Deviations.
   8. If enabled, draw center of gravity
   10. Draw slope line
   11. Draw all individual data point dots.

II. StripChart Outer Loop
The steady state analysis routine looks at the data points and determines the slope of the best fit through the data cloud. Source data for the graph at the left is thermal drift of 8 voltages provided by pots and op-amps in a test rig.

Figure 5 Average Values Crawling Regression.
The threshold of “SteadyState” will be a set number of sets of raw data with standard deviation and slope below their limits and a set number of averages, a set time interval apart, with standard deviations and slopes below their limits.

13. PreSet Max and Min values for X and Y channels.
14. Do each measurement channel set start to finish, and then step to the next one. They are in order, Head, Tail, Gate, Blade Flow, Power, GateServoPressure and BladeServoPressure. For each one in turn:
15. Get as many averages as the operator input calls for for Crawling Regression.
16. Determine Max and Min values for each measurement set.
17. Step through every measurement set, comparing the stored values with the Max and Min registers. Replace Max or Min with new values from the array. If Value < Min then Min = Value. If Value > Max then Max = Value.
18. Compare the stored value with the max and min registers. Replace Max or Min with Value by this logic: If Value < Min then Min = Value. If Value > Max then Max = Value.
19. Add or subtract undershoot values to give some white space above and below the Data set.
20. Define top, bottom, left and right for graph in Array1 through Array 8.
21. Draw grids for whichever array is being drawn to.
22. Use Max and Min values to draw Y axis scale on graph. X axis scale is just the number of samples = full span.
23. Plot Dots on the grids, as shown in figure 3.
24. Store data point

The next sequence computes the average values, standard deviations and slopes of result values derived from steps 1-23.

25. Run LeastSquare routine to find Center of Gravity and slope of data cloud.
   a. Load X and Y values into intake arrays RegX and RegY.
   b. Determine starting point and NumPoints and call CurveFit routine.
      i. Compute averages of X and Y values. Plotted point of average X and Average Y is the “center of gravity” of the data set.
      ii. To find slope and Y intercept:
          1. Sum1 = Sum1 + (X Value – XAverage) * (YValue – YAverage)
          2. Sum2 = Sum2 + (X Value – XAverage)²
          3. Slope = Sum1/Sum2
          4. YIntercept = YAverage - slope * Xaverage
          5. Discard outliers and repeat steps 1-4.
          6. Compare standard deviation and slope to pre-defined limits from calibration table.
          7. Store MidX, MidY, Slope and Standard Deviations.
          8. If enabled, draw center of gravity
          10. Draw slope line
          11. Draw all individual data point dots.
26. Repeat for all channels.
27. (The output value here is the SteadyState “center of gravity” value with accompanying standard deviation and slope for each channel.)
The above procedure monitors all input channels continuously. If at any time, unit operation is determined to be SteadyState, efficiency is computed, the data point is logged, and monitoring continues.

### III. Graphing Routine for StripChart

**GraphData** *(InputArray, ControlArray)* This procedure shifts selected columns of data from the InputArray into the Graphing Array, then analyzes the data and constructs a normalized X-Y (-Y,-Y,-Y…-Y) graph of the data in a Visual Basic Window. This routine parses data into 4 visual basic arrays, performs sort operations on desired columns of data, and manipulates columns of data into the data-graphing array.

The data graphing routine determines max and min values for X and Y parameters and draws the Cartesian coordinate grid, computes and draws the Y and Y axis scales to optimize display utilization of the graph space, then plots up to 10 “Y Axis” parameters against one “X Axis” parameter.

The image above shows the graphing routine output. The four arrays of parsed data are visible. As is the Graphing array, just set for WK Flow vs. Gate angle. To change what data gets plotted click on the third row down and enter an X for the x axis of the graph, or up to 10 Ys to set columns to be graphed along the Y axis.

28. Look at each cell in Row 3 of all four arrays to find “x”. This column is to be the X-axis of the graph. It is copied to the graph input data array. This is the white array on the right side of the screen.
29. Look at each cell in Row 3 of all four arrays for “y”. These columns will be plotted along the Y-axis against the X-axis that was set in step 26.

30. Predetermine Max and Min values for X and Y channels.

31. Plot each measurement channel set start to finish, and then step to the next one. They are in order, Head, Tail, Gate, Blade Flow, Power, GateServoPressure and BladeServoPressure. For each one in turn:
   a. Preset Max and Min registers for each measurement set.
   b. Step through every measurement set, comparing the stored values with the Max and Min registers. Replace Max or Min with new values from the array.
   c. If Value < Min then Min = Value. If Value > Max then Max = Value.
   d. Compare the stored value with the max and min registers. Replace Max or Min with Value by this logic: If Value < Min then Min = Value. If Value > Max then Max = Value.
   e. Add or subtract undershoot values to give some white space above and below the data set.
   f. Define top, bottom, left and right for graph in Array1 through Array 8.
   g. Draw grids for whichever array is being drawn to.
   h. Use Max and Min values to draw Y-axis scale on graph. X-axis scale is just the number of samples = full span.
   i. Plot Dots on the grids, as shown in figure 3.

The graphing program is a generic plotting routine that receives its data in a preset format from the analysis part of the program we’re going to develop together.
Shown above, Fig 1 is a graph of seven parameters along the Y axis plotted against Gate Position as the X axis.

All of the data are stored as columns of numbers in spreadsheet routine programmed in Visual Basic. The visual array, shown as Table 1, is programmed as a basic spreadsheet with features specialized to manipulate and graph the data.

To handle the sample data set from McNary Dam Unit 6, four arrays were used to emulate and extrapolate on the four original Excel spreadsheets that contain the entire data set.

Only the fourth “results (2)” from the no screens workbook is shown here. On the full display, all four arrays shows only 14 rows, but the scrollbars on the right side allow access to the 200 rows in the array.

Book marked text:

The visual component for the data array display is a “grid” from the Microsoft Visual Basic toolbox. This is a basic data holder, with very limited features.

Most of the nice features that make spreadsheets appealing: auto-math, boxes displaying the contents of another box etc, are missing here. Software must be written to create, and in many cases improve on these tools in Microsoft’s spreadsheet program.

Several macros have been written to manipulate, sort and graph the data in these grids. These data are read sequentially from a storage file into the arrays.

The number of columns to parse the data into is set by clicking a button to call the setup menu. Values loaded into this spreadsheet are parsed from left to right into as many columns as are set by the operator.

The leftmost entry in row# 3 is “Run#,” the header for the Run Number column. Putting a “y” in this cell causes that column of data to be plotted along the Y-axis, as shown in Figure 1.
Whether or not a set of data is displayed on the graph is set by the value in the 3rd row of each array of data in the array that is being graphed. If the value is “x” then this is plotted as the x axis of the graph. If the value is “y”, then this is one of the Y axis of the graph. Up to 10 Y axes can be displayed simultaneously. Y scale for #2 and all subsequent data set are shown to the right of the graph. (return)

The SteadyState routines produce a stream of “steady state” for the graphing routines by applying statistical and frequency domain analysis to the real-time measurement process.

(end of file)