ANL/IFR/CP--84089
CONF-950564--8

APPLYING NEURAL NETWORKS TO OPTIMIZE INSTRUMENTATION PERFORMANCE

by

Steven E. Start
Gregory G. Peters

Argonne National Laboratory - West
Engineering Division
P.O. Box 2528
Idaho Falls, ID 83403-2528
(208) 533-7096

Submitted for Presentation at
The Ninth Power Plant Dynamics,
Control & Testing Symposium

May 24-26, 1995
Hyatt Regency Hotel, Knoxville, Tennessee

DISCLAIMER

DISCLAIMER

APPLYING NEURAL NETWORKS
TO OPTIMIZE
INSTRUMENTATION PERFORMANCE

Steven E. Start, Gregory G. Peters
Engineering Division
Argonne National Laboratory - West
P.O. Box 2528
Idaho Falls, Idaho 83403-2528

ABSTRACT

Well calibrated instrumentation is essential in providing meaningful information about the status of a plant. Signals from plant instrumentation frequently have inherent non-linearities, may be affected by environmental conditions and can therefore cause calibration difficulties for the people who maintain them. Two neural network approaches are described in this paper for improving the accuracy of a non-linear, temperature sensitive level probe ised in Expermental Breeder Reactor II (EBR-II) that was difficult to calibrate.

INTRODUCTION

At EBR-II, the signal from an inductive level probe in the secondary sodium storage tank provides an important indication to plant operators when the secondary coolant loop is being filled or drained. The storage tank has large heaters which keep the sodium in liquid form during transport and during storage in the tank. An inductive probe measures the liquid level in the storage tank.

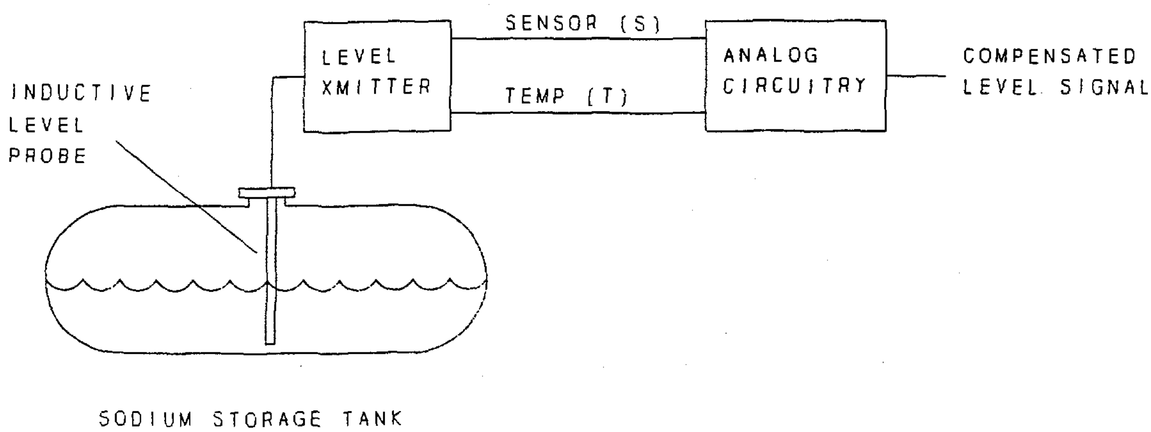


Figure 1: Secondary Storage Tank Level Probe

* Work supported by the U.S. Department of Energy, Reactor Research Technology under Contract No. W-31-109-ENG-38.

Specialized analog circuitry, purchased with the probe, compensates for probe non-linearities and temperature sensitivness to produce a linear level signal as shown in Figure 1. Calibration of the analog circuitry has been very difficult because it contains several potentiometer circuits that interact with each other.

This paper discusses two neural network approaches to solving the level probe calibration problem. The first approach uses an adaptive linear element (ADALINE) to calibrate the analog board. The second, a more general approach, uses a feed-forward neural network implemented on a microcontroller.

APPLYING WIDROW-HOFF TO CALIBRATE THE ANALOG CIRCUIT

The analog circuitry consists of two printed circuit boards: a signal conditioning board and a signal compensation board. The compensation board consists of analog multipliers, a summing amplifier and output amplifiers. The multipliers and summing amplifier are shown in Figure 2.

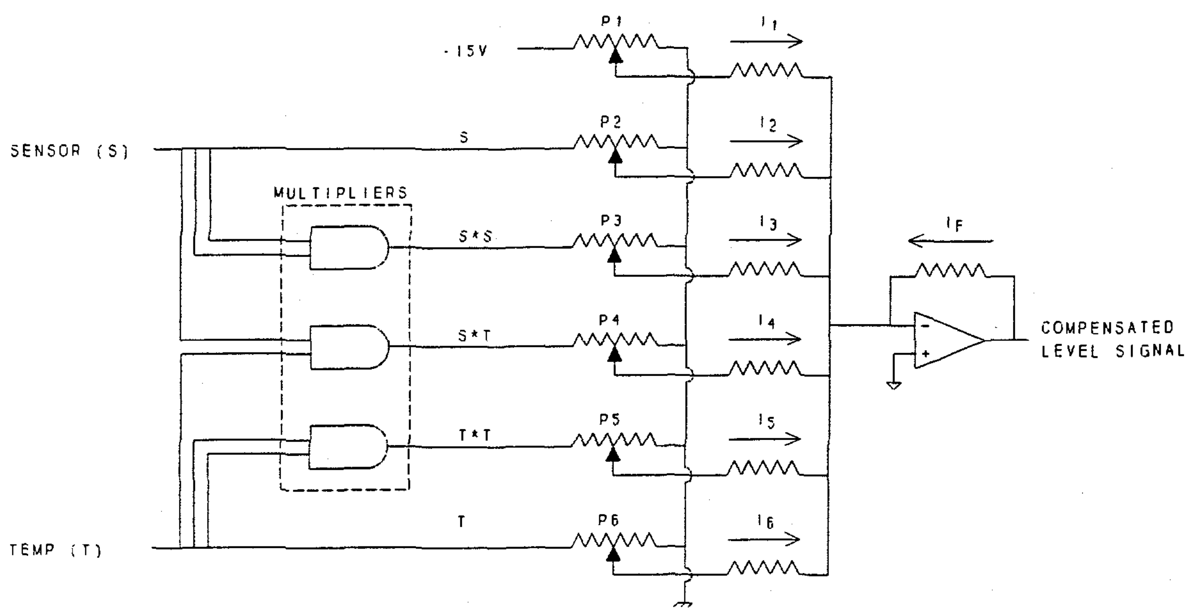


Figure 2: Analog Compensation Circuitry

The analog circuit implements the equation shown below. The constants (Ks) in the equation represent unitless ratios of resistances.

$$V_{OUT} = -15*K1 + S*K2 + S^2*K3 + S*T*K4 + T^2*K5 + T*K6$$

An ADALINE network architecture was selected for this problem because of its similarities to the analog compensation circuitry. The summing amplifier feedback current, I_F, is the sum of the input currents, I_1 through I_6. The input currents are the applied voltages times their respective conductances.

$$I_F = -15*G1 + S*G2 + S^2*G3 + S*T*G4 + T^2*G5 + T*G6$$

The weights and bias of the ADALINE are comparable to the summing amplifier conductances and a D.C. offset current. The output of the ADALINE is:

$$A1 = B1 + S*W2 + S^2*W3 + S*T*W4 + T^2*W5 + T*W6$$

The ADALINE architecture is shown in Figure 3. Two sets of data were collected for training the network. First, the tank was heated while maintaining a constant level. Next, the level probe was physically raised while maintaining a constant temperature. The Widrow-Hoff learning rule was applied to the training data. Training was done using PC-MATLAB and the Neural Network Tool Box. The potentiometers on the compensation board were adjusted to produce resistance ratios based on the weights of the ADALINE network.

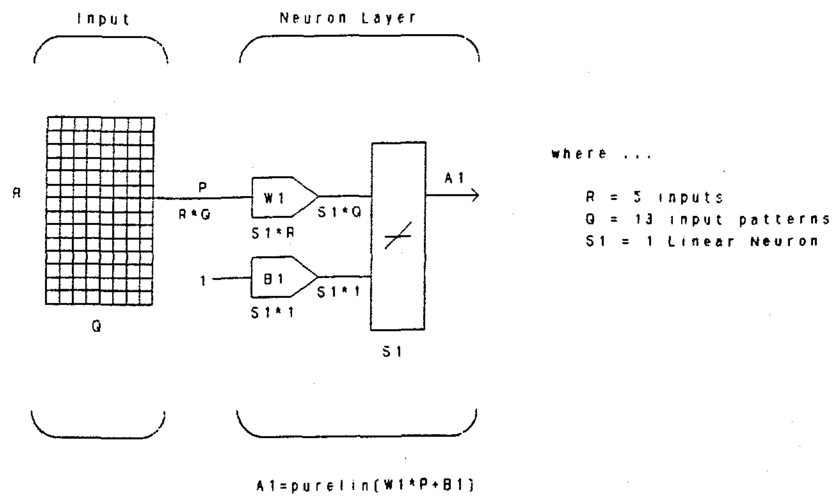
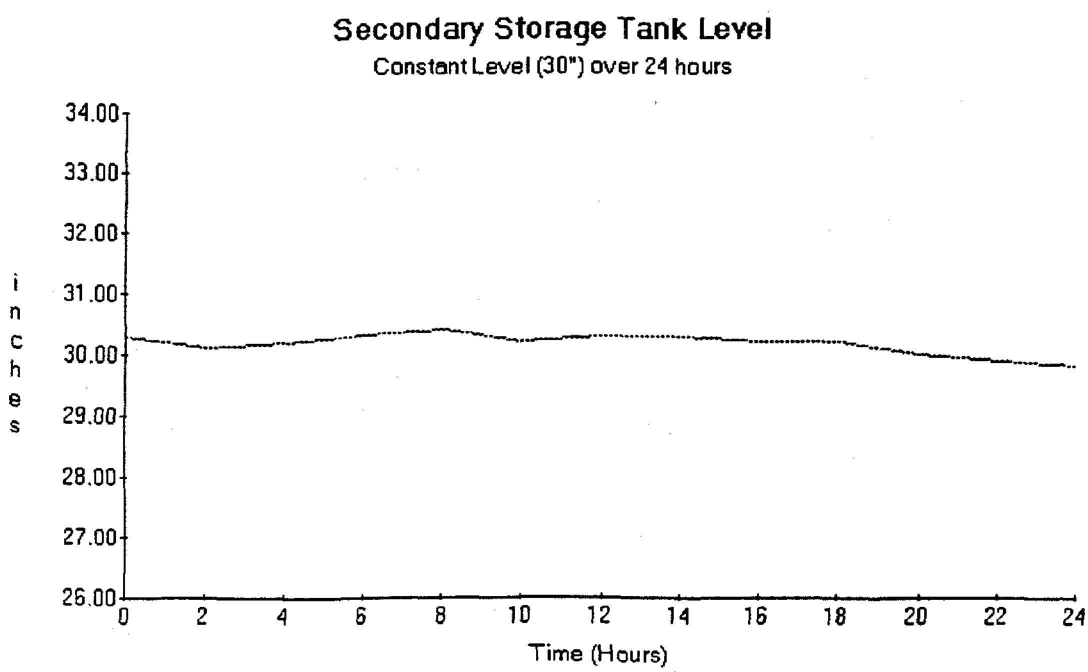


Figure 3: ADALINE Network Architecture

After calibration, a temperature compensation test was done to verify the adequacy of the calibration. The storage tank temperature was raised 80°F over a twenty-four hour period while the sodium level remained constant at 30 inches. The level indication error over this period was less than 0.5 inch as shown in Figure 4.



IMPLEMENTING A FEED-FORWARD NETWORK ON A MICROCONTROLLER

While the results obtained with the ADALINE network were sufficient to accurately calibrate the level probe, it was recognized that the accuracy of the level measurement could be improved while simplifying maintenance. By using a multiple layer feed-forward network trained using backpropagation, the non-linear temperature sensitive characteristics of the probe could be more accurately compensated for. In addition, analog-to-digital and digital-to-analog devices have proven to be easily calibratable devices. In this scenario, the analog compensation board would be replaced with a small general purpose embedded computer. Because of a shutdown and decommisioning order for EBR-II, the microcontroller approach could not be implemented in the actual system. However, the experience and information gained will be useful in evaluating future instrumentation needs.

A two layer log-sigmoid/linear network architecture was chosen as shown in Figure 5. The number of neurons in the hidden layer was chosen using Shannon's entropy function[1] and is given as follows:

$$S1 = R * \log_2(Q) \pm R$$

where $S1$ = Number of neurons in the hidden layer.
R = Input pattern size.
Q = Number of training patterns.

The network was trained with the same data that had been collected for training the ADALINE. Backpropagation with an adaptive learning rate, momentum and Nguyen-Widrow initial conditions[2] was used to train the network. The network was successfully trained to achieve an error of less than ±0.1 inch over all training patterns. The trained network was then implemented on the microcontroller.

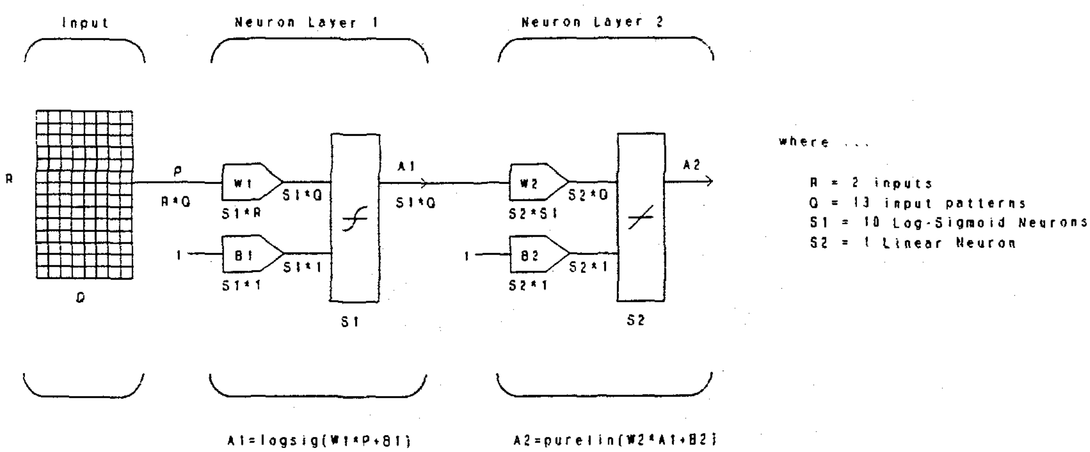


Figure 5: Feed-Forward Network Architecture

The embedded computer selected utilized a Z180 CPU operating at 12.288 MHz, on-board EPROM and RAM, eight 12-bit analog inputs, one analog 12-bit analog output, twenty-four digital I/O lines and four serial ports. The board uses floating point math libraries for all floating point computations. The cost of the microcontroller was approximately $500.

The microcontroller used a small real-time kernel to manage real-time and non-real-time tasks. The software structure consisted of two tasks: a real-time foreground task and a background task. The foreground task performed the following functions: read two analog inputs, convert analog input values to engineering units (which form the input to the network), perform feed-forward network calculations and convert numerical values to strings for output. (String conversions were done in the foreground because math libraries were non-reentrant.) The background task outputs the strings to a user screen. All programming was done in C. Coding took approximately two days.

Floating point performance for the board was approximately 85 μsec per multiply and 240 μsec per divide. This points out that microcontroller performance is inadequate for training networks but sufficient for execution. Despite the lack of a math coprocessor, the microcontroller was able execute both the foreground and background tasks approximately three times every second. No efforts were made to optimize the code for speed. Most math operations were done using floating point arithmetic. It is believed that significant speed improvements could be made by using integer arithmetic techniques for most operations without compromising required accuracy.

Matrix math functions that were originally developed for use on a personal computer (PC) were easily ported to the microcontroller. The only computational limitations that were encountered were related to floating point number size. Double precision floating point numbers were not supported as a data type by the cross-compiler. Consequently, calling functions were responsible for limiting the maximum and minimum size of values to be passed to exponential functions in order to avoid run-time errors.

Software development was done on a PC, cross-compiled for a Z180 processor and downloaded into the RAM on of the microcontroller board. During development, the microcontroller uses a monitor PROM for programming debugging. The final code resides in an EPROM which replaces the monitor PROM. Some of the motivation for applying a microcontroller to a neural network problem is that small embedded systems have been used frequently over the past several years in EBR-II[3]. Past systems used STD bus technology and systems used either Z80 or V53 CPU boards and separate I/O boards. The microcontroller with integrated I/O was a natural migration due to it's small size and dedicated function.

CONCLUSION

Utilizing Widrow-Hoff training techniques to train an ADALINE network provided an accurate means to calibrate an otherwise very difficult circuit. The key to this solution was in observing the analogy between the analog circuitry and the ADALINE architecture. Implementation of a feed-forward neural network on a microcontroller demonstrates the feasibility of this approach in addressing non-linear instrumentation problems. The microcontroller offers a simple and inexpensive solution.

REFERENCES

1. B. R. Upadhyaya, G. Mathai and E. Eryurek, "Development and Application of Neural Network Algorithms for Process Diagnostics," *Proceedings of the 29th Conference on Decision and Control*, Honolulu, Hawaii, December 1990.

2. H. Demuth and M. Beale, Neural Network Toolbox," The MathWorks, Inc., Natick, MA (1992)

3. R. B. Carlson and S. E. Start, "Embedded Computer Systems for Control Applications in EBR-II," *Proceedings of the Topical Meeting on Nuclear Plant Instrumentation, Control and Man-Machine Interface Technologies*, Oak Ridge, Tennessee, April 1993