专 题 汇 编

IJCNN 模式识别

The Intelligent Arc Furnace™ Controller:

A Neural Network Electrode Position Optimization System for the Electric Arc Furnace

William E. Staib

Neural Applications Corporation
180 North Wabash Avenue
Chicago, IL 60601
email: staib@whirlwind.stanford.edu

Dept. of Electrical Engineering
Stanford University
P.O. Box 13915
Stanford, CA 94305

Robert B. Staib

Milltech-HOH,INC.
4711 North Brady Street
Davenport, IA 52806

ABSTRACT

Electric Arc Furnaces use electric power to melt steel. The amount of power delivered to the scrap metal is controlled by positioning three large electrodes. A Neural Network control system can learn to predict the relationships between electrode position and stability of furnace operation. Real-time adaption to changes in furnace conditions enables the system to increase productivity while reducing furnace and electrode wear, lowering power consumption, and achieving an unprecedented smooth melt. The result of Neural Network control is a savings of millions of dollars per year per furnace.

Keywords: Steel, Neural network, Process control, Real-time adaption, Electric power, Predictive control, Chaotic systems.

I. INTRODUCTION

To control an Electric Arc Furnace is to control a lightning storm. Tens of megawatts of power are violently delivered into scrap metal through huge carbon electrodes weighing thousands of pounds. Effective positioning of the electrodes is essential for proper furnace operation. However, due to the complex, unbalanced three-phase nature of Electric Arc Furnace circuitry and oversimplified, often incorrect system models, electrode regulators have historically achieved rather crude furnace control. In two previous papers, we have identified flaws in prior Electric Arc Furnace control schemes and suggested how these problems might be overcome with Neural Network techniques. Here, we will present quantative performance results of the Intelligent Arc Furnace™ (IAF™) Regulator and analyze qualitatively how the Neural Network improved furnace operation.

In August, 1990, Milltech-HOH (now a sister company of Neural Applications Corporation) and North Star Steel - Iowa, agreed to research the application of Neural Network techniques to the closed-loop process control of an Electric Arc Furnace. One year later, in September, 1991, the project was successfully completed. All data presented here were obtained from North Star Steel - Iowa's 80 ton Whiting furnace. The furnace is 16.6 feet in diameter, and utilizes a 30 MVA transformer.

NOTE: It is important to realize that before the IAF™, North Star Steel - Iowa was equipped with a state-of-the-art closed-loop control computer, a modern electrode regulator, and had efficiency numbers among the best in the industry. The savings realized at North Star were <u>not</u> due to

0-7803-0559-0/92 $3.00 © 1992 IEEE

improving an inefficient operation. All savings data occurred while the IAF™ system was receiving its setpoints from the existing computer control system.

II. FURNACE PHENOMENA SUITABLE FOR NEURAL NETWORKS

Although a quantative analysis is impractical here, it is important to gain a qualitative understanding of how Neural Networks are ideal for furnace electrode regulation.

A. FURNACES ARE COMPLEX SYSTEMS

The optimal control signal when the furnace is running at 30,000 amps is quite different from when it is at 50,000 amps. A Neural Network can learn these relationships easily, while designing a process model for an Electric Arc Furnace is extremely difficult.

B. FURNACES ARE DYNAMIC SYSTEMS

The Electric Arc Furnace is difficult to accurately model because its electrical characteristics are constantly changing. At the beginning of a "heat" of steel, cold scrap metal is placed in the furnace--a large bucket. The scrap may have many small windings and therefore has a large amount of reactance. As the scrap melts, the load becomes almost completely resistive. Furthermore, no two buckets of scrap are the same, so there is no good way to enter sufficient data for a rule-based system to calculate what optimal control should be for a given heat of steel.

A Neural Network can not only learn the general change in furnace response as the steel melts, but also it can quickly recognize and adapt to unusual scrap makeup.

C. NEED TO BE THREE-PHASE AWARE

Examine the Arc Furnace Schematic in Figure 1. To operate, current must flow from one electrode, arc into the scrap, and then flow out one or both of the other electrodes. All previous electrode regulation schemes were based on the assumption that the Electric Arc Furnace can be modeled as three separate single-phase systems. Let's explore this assumption.

Suppose a setpoint of 30,000 amps was desired and the furnace was in the following condition:

PHASE A AMPS	PHASE B AMPS	PHASE C AMPS
30,000	30,000	45,000

If the single-phase approximation was perfect, Phase C would be corrected to 30,000 amps but because current flows out one electrode (phase), and returns through the other phases, lowering Phase C current will also lower the currents through Phases A and B. So with a single phase control system, this might happen:

PHASE A AMPS	PHASE B AMPS	PHASE C AMPS
25,000	25,000	30,000

A Neural Network, however, learns to ask for increased current on phases A and B while correcting the error on phase C, thus attaining approximately 30,000 amps on

each phase. This property of recognizing three phase relationships we call being Three-Phase Aware. Because of the mathmatical complexity of the relationships that a control change on one phase has on another, people have not been able to solve this Three-Phase Aware problem. A Neural Network can determine these relationships after only 10 to 20 minutes of on-line learning.

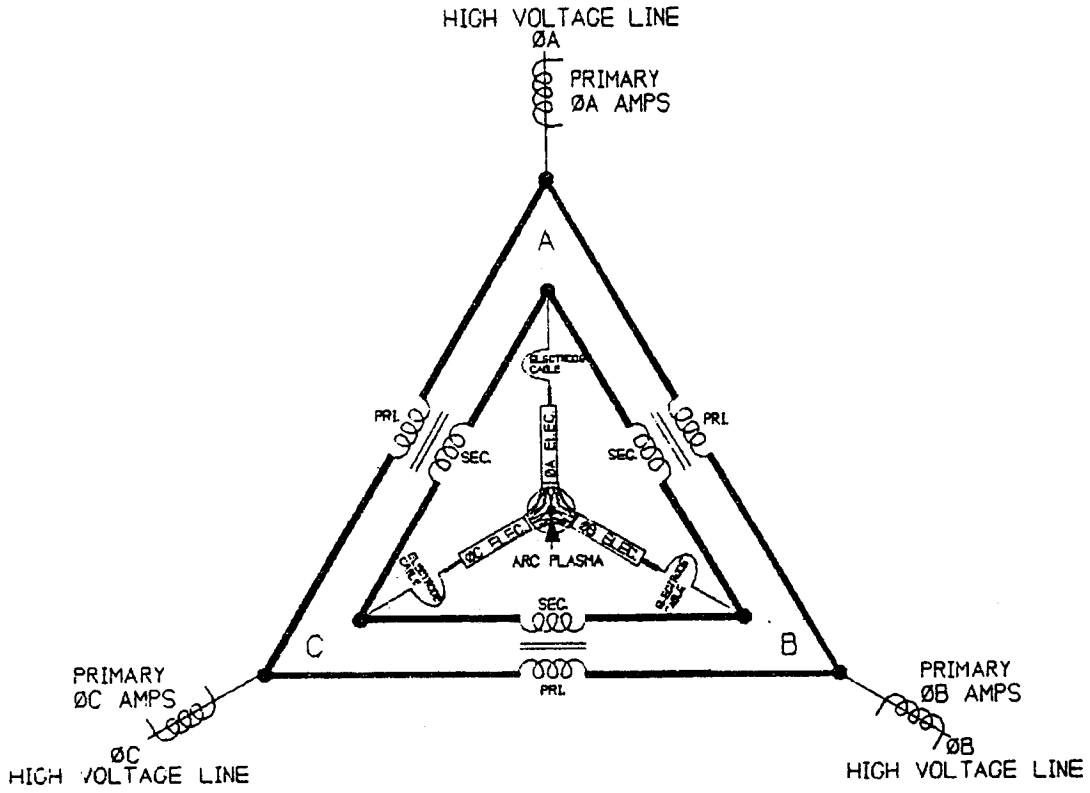


Figure 1: Arc Furnace Schematic

D. FLICKER REMOVER

The electric arc used to make steel is like the arc of a spark plug in that if the contacts of a spark plug are too far apart, the spark plug won't arc. Similarly, if the distance between the end of the electrode and the scrap metal is too great, no current will flow. Flicker happens when the furnace is at the border of instability. As in Figure 2, current does not flow for part of the cycle; then a rapid change occurs as current suddenly flows again. This distortion of the sinewave causes less power transfer into the metal, more electrode wear, and disturbances on the power grid–flicker.

The Neural Network can utilize high speed readings from the Data Acquisition System to predict and correct flicker problems.

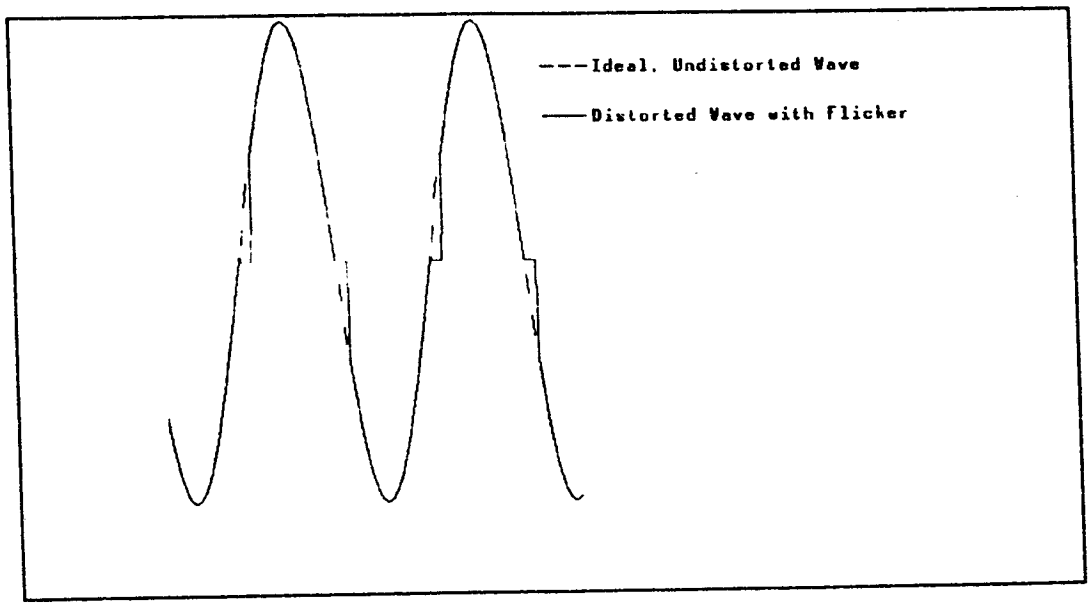


Figure 2: Phase Amps Versus Time

E. PREDICTIVE CONTROL

Because the Neural Network can learn to predict furnace response, unwanted trends can be identified and solved—even before they happen. The Neural Network learns the mechanical time constant of the electrode drive system and, through predictive control, can dramatically reduce this delay. (We note however, that sometimes incorrect predictions actually increase response time. No system can be perfect.)

III. ARC FURNACE DATA ACQUISITION SYSTEM

A high-speed data collection computer was designed and built. This machine contained a 400,000 sample-per-second, 12-bit-accuracy, 16-channel analog input subsystem, an Intel 80486 I/O processor, an Intel i860 80 million floating-point operation-per-second processor for numerical processing, and 12 megabytes of RAM memory. On-line storage was to 1.0 gigabyte hard disks and off-line storage was to a multi-gigabyte tape backup system. An interface subsystem was developed to connect the above hardware to an electric arc furnace in order to collect the following data items:

A. Primary Phase to Phase Volts
B. Primary Phase Amps
C. Secondary Phase to Phase Volts
D. Secondary Phase to Hearth (Ground) Volts
E. Secondary Phase Amps
F. Existing Regulator Outputs (UP/DOWN)
G. Microphone (operating sounds)
H. Regulator automatic / manual signals

Note: For all voltage, current, and regulator output signals, all three phases were sampled.

Figure 3 shows a block diagram of the data collection system and how it was connected to the arc furnace and its regulator.

This metering system enabled the researchers to store high-speed readings from entire heats of steel (approximately two hours in length) to disk for off-line analysis and training.

NOTE: The metering system in this project is more than 1000 times faster and much more accurate than the transducers used in other electrode positioning systems. Because improved metering techniques resulted in significant savings, we have shown savings calculations from a rule-based system using the new metering in addition to data from the Neural Network regulator.

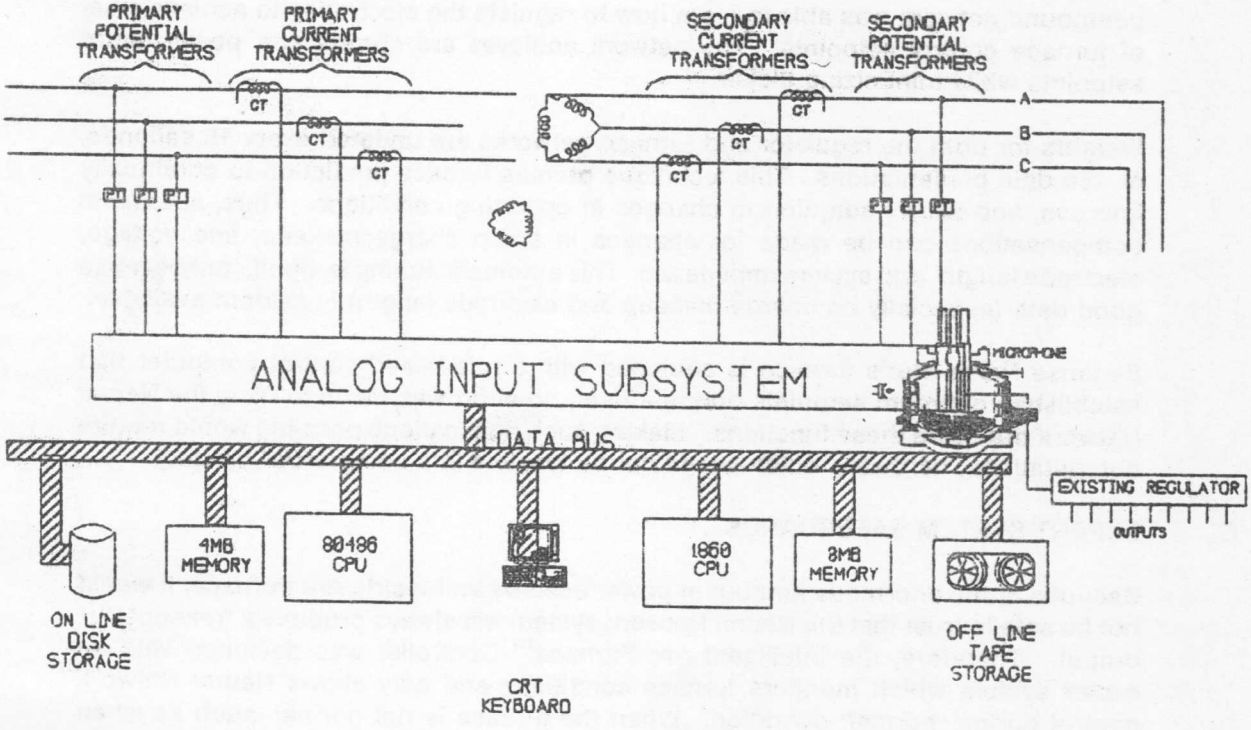


Figure 3: Arc Furnace Data Acquisition System

IV. OVERVIEW OF NEURAL NETWORK ARCHITECTURE

Three configurations of an Emulator and a Controller network were used. All utilized the Extended Delta-Bar-Delta adaption algorithm. Altogether, the models consisted of over 200 fully connected neurons. Block diagrams are shown in Figure 4.

A. REGULATOR EMULATOR NETWORK (CONTROLLER)

First, a regulator-emulating network was designed and trained to match the responses of North Star's existing electrode regulator. This was done as a safety measure to ensure that the Neural Network would initially output reasonable control signals.

B. FURNACE EMULATOR NETWORK

Next, a furnace-emulating network was trained. The network was given a time history of regulator outputs and furnace state conditions plus the regulator outputs for timeslice N + 1. Using these as inputs, the network was trained to predict the furnace state values for timeslice N + 1.

C. COMBINED REGULATOR/FURNACE NETWORK

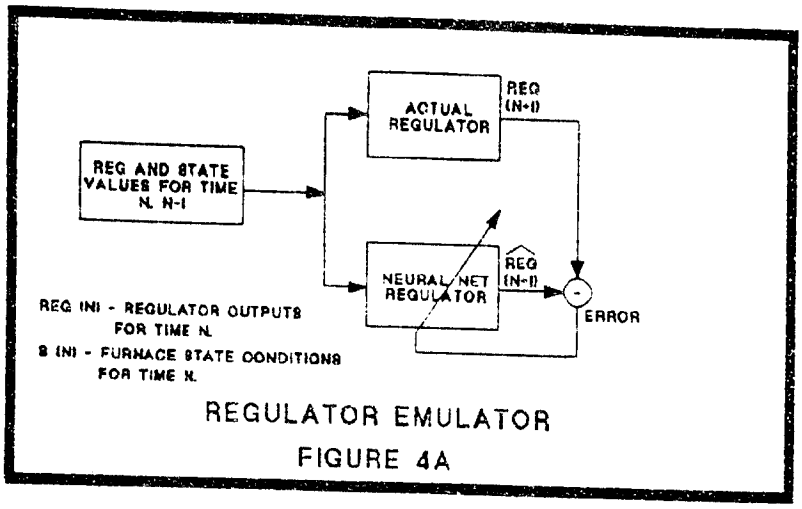
Finally, the furnace and regulator emulation networks were combined. The resulting compound network was able to learn how to regulate the electrodes to achieve a set of furnace control setpoints. The network achieves arc current and power factor setpoints while minimizing flicker.

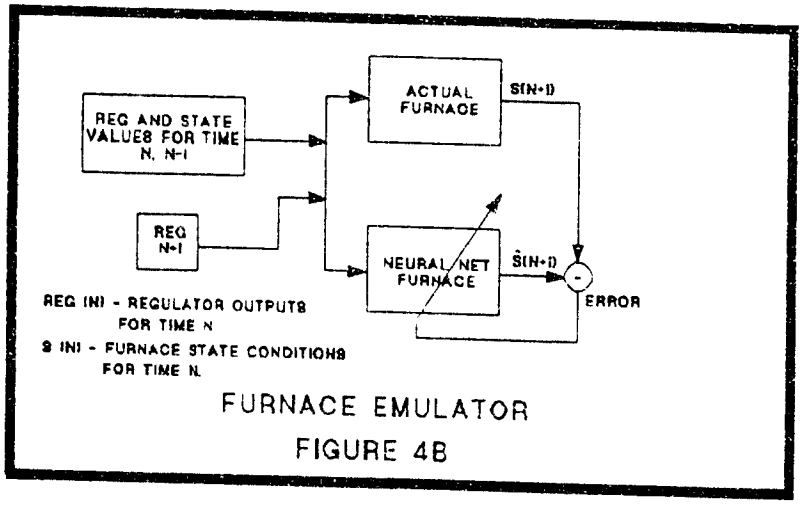
Weights for both the regulator and furnace networks are updated every 15 seconds, or 150 data presentations. This technique permits furnace prediction to continually improve, and allows adaption to changes in operating conditions. Thus, automatic compensations can be made for changes in scrap charge makeup, line voltage, electrode length, and system impedance. This automatic tuning is significant because good data (especially on charge makeup and electrode length) is seldom available.

Because North Star's furnace is equipped with a rule-based control computer that establishes optimum setpoints during a heat, no effort was made to have the Neural Network duplicate these functions. Making such duplications possible would require computational hardware which would not be feasible in a commercial product.

D. EXPERT SYSTEM SAFEGUARDS

Because of the enormous amount of power used by an Electric Arc Furnace, it would not be safe to trust that the Neural Network system will always produce a "reasonable" output. Therefore, the Intelligent Arc Furnace™ Controller was designed with an expert system which monitors furnace conditions and only allows Neural Network control during "normal" operation. When the furnace is not normal--such as when scrap has caved-in, creating a short circuit by touching an electrode--a rule-based regulator is invoked. The Neural Network system has proven surprisingly robust and stable. Control is only rarely switched to rule-based mode--and the system has never created any safety problems.





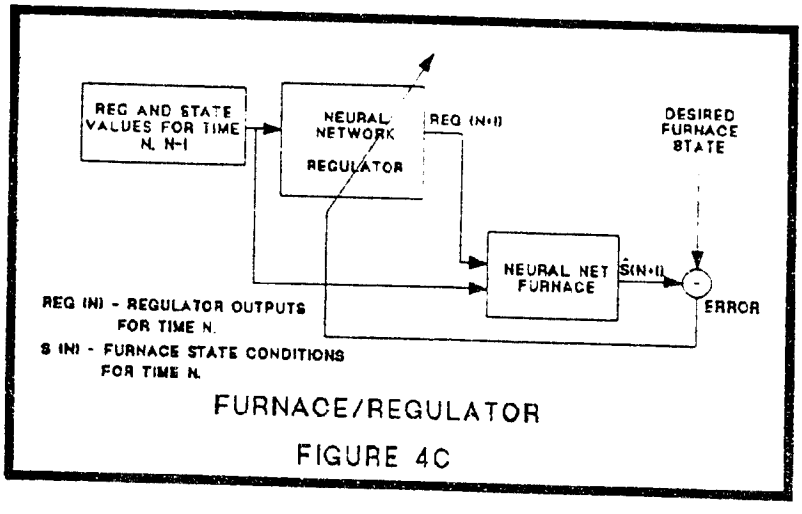


Figure 4· Block Diagrams of: A. Regulator Emulator B. Furnace Emulator C. Furnace/Regulator

SAVINGS DATA

All of the following are actual improvements at North Star Steel, Wilton, Iowa. However, the ar amounts are calculated assuming the following hypothetical firm: production is 300,000 tons vear; profit per ton is $50; cost of electrodes is $1 per pound; and electrical energy is billed at a of $.03/kilowatt•hour. These figures are very conservative. This hypothetical plant would achieve ngs considerably less than those realized at North Star.

---

st we present the savings of a rule-based controller which shows the improvement due to better tering and rule-based control techniques: (See Note Section III.)

nber of rule-based heats (RULE) : 123

umber of control heats (OLD) : 823
ne two months before the rule-based tests)

EM	PERCENT IMPROVEMENT	SAVINGS
wer Usage	1.30%	$38,700
ectrode Usage	2.94%	$48,000
oductivity	2.20%	$330,000

ULE" NET SAVINGS PER FURNACE PER YEAR ----------> $416,700

---

ext, we present the savings of the IAF™:

umber of rule-based heats (IAF) : 250

umber of control heats (OLD) : 823

EM	PERCENT IMPROVEMENT	SAVINGS
wer Usage	3.3%	$118,800
ectrode Usage	30.89%	$384,000
roductivity	4.8%	$720,000

AF" NET SAVINGS PER FURNACE PER YEAR ----------> $1,222,800

RCENT OF SAVINGS ATTRIBUTABLE TO NEURAL NETWORK ONLY = 65.9 %

VI. CONCLUSIONS

The application of Neural Networks to the control of an Electric Arc Furnace has been a great success. Not only has the Intelligent Arc Furnace™ Controller been able to create substantial savings, but also it has proven to be remarkably stable during five months of full-time operation. It should be noted that the system has successfully been installed on other furnaces.

We hope this work will give credibility to Neural Networks' abilities to perform critical tasks. We are presently working to develop similar systems for several other industrial applications.

References:

Harper, T. A., Bliss, N. G., "Optimization of Electric Arc Furnace Electrode Control Performance through Closed-Loop Control," *AISE Iron and Steel Exposition*, Pittsburgh, PA, 1989.

Lee, S., Kil, R., "Robotic Kinematic Control Based on Bidirectional Mapping Neural Network," IJCNN, vol. 3, pp. 327-336, June 1990.

Minai, A., Williams, R., "Backpropagation Heuristics: A Study of the Extended Delta-Bar-Delta Algorithm," IJCNN, vol 1., pp. 595-600, June 1990.

Nguyen, D., Widrow, B., "The Truck Backer-Upper: An Example of Self-Learning in Neural Networks," Hand Out #5, EE392W, Stanford University, 1990,

Persson, J. A., "Analyzing the Characteristics of Arc Furnaces," *Electric Arc Furnace Proceedings*, vol. 47, pp. 31-36, 1989.

Staib, W. E., Hawkes, K., "The Application of Neural Networks to Music Composition: The Melody Predictor and Harmony Arranger," EE392W Final Project, Stanford University, 1990.

Staib, W. E., Bliss, N. G., Staib, R. B., "Neural Network Conversion of the Electric Arc Furnace into the Intelligent Arc Furnace," *74th AIME Steelmaking Conference*, Washington, D.C., 1991.

Staib, W. E., Bliss, N. G., Staib, R. B., "Recent Developments in Neural Network Applications - Neural Network Conversion of the Electric Arc Furnace into the Intelligent Arc Furnace," *1991 AISE Spring Conference*, Detroit, MI, 1991.

Widrow, B., "Adaptive Expert System: The 'Broom Balancer'," Hand Out #13, EE392W, Stanford University, 1990.

Widrow, B., Hoff., M., "Adaptive switching circuits," *1960 IRE WESTCON Convention Record*, New York, pp. 96-104, 1960.

FAST NEURAL SOLUTION OF A NONLINEAR WAVE EQUATION

Nikzad Toomarian and Jacob Barhen

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

A neural algorithm for rapidly simulating a certain class of nonlinear wave phenomena using analog VLSI neural hardware is developed, presented, and applied to the Korteweg-de Vries partial differential equation. The corresponding neural architecture is obtained from a pseudospectral representation of the spatial dependence, along with a leap-frog scheme for the temporal evolution. Numerical simulations demonstrate the robustness of the proposed approach.

1. Introduction

A large variety of physical phenomena can be described be means of partial differential equations[11]. In practical applications involving the numerical solutions of such equations, one is not only interested in computational speed, but also in the accuracy of the results, as well as the stability of the numerical scheme employed. Recent advances in neural analog VLSI hardware architectures provide a strong incentive to develop massively parallel neural algorithms, which can fully realize the capabilities of such hardware.

In order to provide a concrete framework for the proposed formalism, we focus our attention on the Korteweg-de Vries (KdV) equation [5]. This nonlinear partial differential equation has the advantage of exhibiting both sufficient computational complexity, and possessing analytical solutions. This enables a rigorous benchmark of the proposed neural algorithm. The KdV (or soliton) equation was originally introduced to describe the behavior of one-dimensional shallow water waves with small but finite amplitudes. Since its discovery, solitons have enabled many advances in areas such as plasma physics and fluid dynamics[e.g., 9,10,12]. In this paper, we limit ourselves to the solution of the one-dimensional, 1-soliton problem.

To fix the ideas, consider the following expression of the KdV equation:

$$u_t + a\ u\ u_x + b\ u_{xxx} = 0 \tag{1}$$

where u_t and u_x denote partial derivatives of u with respect to time and space, respectively. If one selects the values of the constant a and b as 6 and 1 respectively, an analytical solution of Eq. (1) over an infinite interval can be obtained [2]:

$$u(x,t) = 2k^2 sech^2(kx - 4k^3t + \eta_0) \tag{2}$$

where k and η_0 are constants, with $k > 0$. The above expression represents a solitary wave of amplitude $2k^2$ initially located at $x = -\eta_0/k$, moving with the velocity $4k^2$. In order to numericlly integrate Eq. (1), we require the solution to satisfy periodic boundary conditions, i.e., $u(x + 2L, t) = u(x,t)$ for $x \in \Re$ and $t \in [0,T]$. Specifically, we will consider numerical solutions in the region $-L \leq x \leq L$, with the positive constant L chosen to be sufficiently large for periodic conditions to be valid.

0-7803-0559-0/92 $3.00 © 1992 IEEE

2. The Pseudospectral Scheme

In this section, the pseudospectral representation described, will be as it applies to the KdV equation (1). However, the method is general and can be implemented for a broad class of periodic initial value PDEs [3]. In the pseudospectral computational scheme $u(x,t)$ is transformed into Fourier space with respect to the spatial variable, x. The main advantage of this operation is that the derivatives with respect to x are algebraic in the transformed variable. Furthermore, we normalize the spatial period to the interval $[0, 2\pi]$, using the transormation $\frac{\pi}{L}(x+L) \rightarrow x$. The scaled KdV equation can then be expressed in terms of the new state variable $v(x,t)$ as

$$v_t + 6\ s\ v\ v_x + s^3\ v_{xxx} = 0 \quad (x,t) \in \Re \times [0,T], \tag{3}$$

where s denotes π/L.

In order to numerically solve Eq. (3), the interval $[0, 2\pi]$ is discretized by $2N$ equidistant points, with spacing $\Delta_x = \pi/N$. The function $v(x,t)$, which is defined only at these points, is approximated by $V(x_n,t)$, where $x_n = n\Delta_x$, and $n = 0, 1, \cdots 2N-1$. The function $V(x_n,t)$ is now transformed to discrete Fourier space by

$$\hat{V}(\mu,t) = F[V] = \frac{1}{\sqrt{(2N)}} \sum_{n=0}^{2N-1} V(x_n,t)\ e^{-i\mu x_n} \tag{4}$$

where μ takes the values $\mu = 0, \pm 1, \cdots, \pm N$. The inversion formula is

$$V(x_n,t) = F^{-1}\{\hat{V}\} = \frac{1}{\sqrt{(2N)}} \sum_{\mu} \hat{V}(\mu,t)\ e^{i\mu x_n} \tag{5}$$

This enables an efficient calculation of the derivatives of v with respect to x

$$v_x \approx V_x = F^{-1}\{i\mu F[V]\} \tag{6}$$

$$v_{xxx} \approx V_{xxx} = F^{-1}\{-i\mu^3 F[V]\} \tag{7}$$

Fornberg has shown [12] how such a representation of derivatives of periodic functions can be related to central difference approximations of infinite order. Combining Eqs. (6-7) with a leap- frog approximation to the temporal derivative, results in the pseudospectral scheme for solving the KdV equation:

$$\begin{aligned} V(x_n, t+\Delta_t) = V(x_n, t-\Delta_t) &- 12s\Delta_t V(x_n,t)[F^{-1}(i\mu\hat{V}(\mu,t))] \\ &+ 2s^3\Delta_t[F^{-1}(i\mu^3\hat{V}(\mu,t))] \end{aligned} \tag{8}$$

where Δ_t denotes the time step size. Fornberg and Whitman [3] modified the final term in Eq. (8) to produce the scheme

$$\begin{aligned} V(x_n, t+\Delta_t) = V(x_n, t-\Delta_t) &- 12s\Delta_t V(x_n,t)[F^{-1}(i\mu\hat{V}(\mu,t))] \\ &+ 2i[F^{-1}(sin(s^3\mu^3\Delta_t)\hat{V}(\mu,t))] \end{aligned} \tag{9}$$

which is more accurate, specially when dispersion effects dominate the nonlinearity. Moreover, the linearized stability condition for Eq. (9) is

$$\Delta_t < (\frac{2L}{N})^3\ 0.1520 \tag{10}$$

Note that a similar condition for Eq. (8) would be more restrictive. A detailed review of numerical schemes for the KdV equation can be found in [8]. Our emphasis here is only on approaches readily amenable to neural hardware implenetaiton.

3. Neural Network Architecture

In order to map the numerical solution scheme of the KdV equation into a neural network architecture, we return to the evaluation of the spatial derivatives. Let us expand Eq. (6) by substituting the appropriate terms from Eqs. (4) and (5). One obtains

$$V_x(x_n,t) = \frac{1}{\sqrt{(2N)}}\sum_{\mu} i\mu \frac{1}{\sqrt{(2N)}}\sum_{m=0}^{2N-1} V(x_m,t)\, e^{-i\mu x_m}\, e^{i\mu x_n} \tag{11a}$$

By rearranging the terms, we get

$$V_x(x_n,t) = \sum_{m} V(x_m,t)(\frac{i}{2N})\sum_{\mu} \mu\, e^{i\mu(x_n-x_m)} \tag{11b}$$

Now, we can represent the last term in Eq. (11b) by a constant matrix which depends solely on the distance between two spatial grid points, i.e.,

$$T^{(1)}_{nm} = (\frac{i}{2N})\sum_{\mu} \mu\, e^{i\mu(x_n-x_m)} \tag{12}$$

Since the function $V(x_n,t)$ is real, its derivatives should be real as well. Thus, only the real part of the matrix in Eq. (12) is needed for the computation of $V_x(x_n,t)$. Hence,

$$T^{(1)}_{nm} = Re[(\frac{i}{2N})\sum_{\mu} \mu\, e^{i\mu(x_n-x_m)}] \tag{13a}$$

resulting in

$$T^{(1)}_{nm} = -(\frac{1}{2N})\sum_{\mu} \mu\, sin[\mu(x_n-x_m)] \tag{13b}$$

Following the same procedure, one can evaluate the third order derivative with respect to x and find the grid interconnectivity matrix which relates to Eq. (7) i.e.,

$$T^{(3)}_{nm} = -(\frac{1}{2N})\sum_{\mu} \mu^3\, sin[\mu(x_n-x_m)] \tag{14}$$

We can now proceed with the specification of the neural architecture. Let each grid point represent a neuron with activation function $v_n(t)$ equal to $V(x_{n,t})$. Combining Eqs. (13b) (14), and (3), the network dynamics is readily obtained:

$$\dot{v}_n(t) + 6sv_n(t)\sum_{m} T^{(1)}_{nm}v_m(t) + s^3\sum_{m} T^{(3)}_{nm}v_m(t) = 0 \tag{15}$$

The initial values for Eq. (15) are similar to those of Eq. (3). Thus, our architecture consist of $2N$ neurons, the dynamics of which is governed by a system of coupled nonlinear differential equations, i.e.,Eq. (15). Two overlapping synaptic arrays, $T^{(1)}$ and $T^{(3)}$ fully interconnect all neurons. The elements of these synaptic arrays are calculated from Eqs. (13b) and (14), respectively.

The main purpose of the above discussion was to cast a nonlinear partial differential equation (i.e., the KdV equation) into a form amenable to implementation on novel charge domain VLSI circuits, currently being developed at Caltech, capable of performing high speed vector-matrix multiplications[1,6-7]. The vector-matrix multiplier consists of a matrix of CCD cells having an architecture inspired by charge injection device (CID) imager pixels, in that one of the cell's gates is connected vertically from cell to cell forming a column electrode, while another gate is connected horizontally forming a row electrode. The charge stored beneath the row and column gates encodes the matrix, with the column and row electrodes representing the input vector and the output vector, respectively. In its most basic configuration, shown in Fig. 1, such a circuit computes the product of a binary input vector and an analog matrix of charge. The computation done by each CCD cell in the matrix is a multiply-accumulate in which the charge, Q_{ij}, is multiplied by a binary input vector element, U_j, encoded on the column line and this product is summed with other products in the same row to form I_i. Multiplication by a binary number is equivalent to adding or not adding the charge at a particular matrix element to its associated row line. Since all column electrodes are pulsed at the same time, and the associated voltage changes are then capacitively summed on the row lines in parallel, the entire vector-matrix multiplication is accomplished in one clock cycle. Many improvements have been incorporated into this basic structure, the most important being the ability to handle digital (instead of binary) input with 2^n levels, and digital output using novel, compact A/D designs. A 256x256 element circuit is currently being fabricated, which is expected to exceed 10^{12} operations/second/bit of precision.

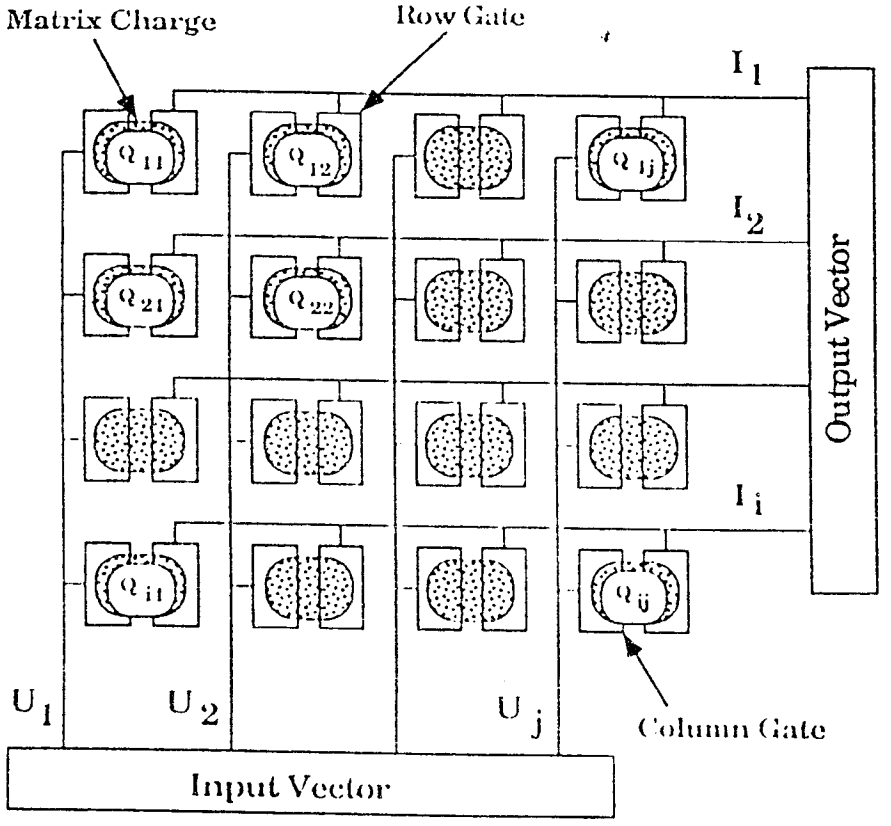


Figure 1. The CID architecture consists of an array of CCD elements that are connected together by row and column electrodes. The matrix values are encoded as charge packets that sit underneath these gates in the silicon substrate. The computation occurs when the charges are transferred from the column gates to the row gates which perform a capacitive sum operation. (Courtesy, C. Neugebauer, Caltech).

In terms of these CCD/CID circuits, the neural network dynamics described by Eq. (15) can be recast as follows, using the leap-frog scheme for temporal dependence

$$v_n(t+\Delta_t) = v_n(t-\Delta_t) - 12\Delta_t s v_n(t) \sum_m T^{(1)}_{nm} v_m(t) - 2\Delta_t s^3 \sum_m T^{(3)}_{nm} v_m(t) \tag{16}$$

If we include the approximation which resulted in Eq. (9), we obtain

$$v_n(t+\Delta_t) = v_n(t-\Delta_t) - v_n(t) \sum_m W^{(1)}_{nm} v_m(t) - \sum_m W^{(3)}_{nm} v_m(t) \tag{17}$$

where

$$W^{(1)}_{nm} = -(\frac{12s\Delta_t}{2N}) \sum_\mu \mu \; sin[\mu(x_n - x_m)] \tag{18}$$

$$W^{(3)}_{nm} = (\frac{2}{2N}) \sum_\mu sin(s^3\mu^3\Delta_t) \; sin[\mu(x_n - x_m)] \tag{19}$$

where $W^{(1)}$ and $W^{(3)}$ are the synaptic matrices corresponding to the first and third spatial derivatives in the KdV equation, and include the scaling factor as well as the time step.

4. Simulations

In order to evaluate the capability of the pseudospectral neural architecture in a CCD/CID neural circuit framework, a computer code was written to simulate Eq. (17) as an approximation of the KdV equation, Eq. (3). In these simulations, the spatial region $[-20, +20]$ was divided into 64 equidistant points, i.e., $\Delta_x = 40/64$. The initial conditions were chosen to be

$$u(x_n, 0) = sech^2(\sqrt{2}x_n) \tag{20}$$

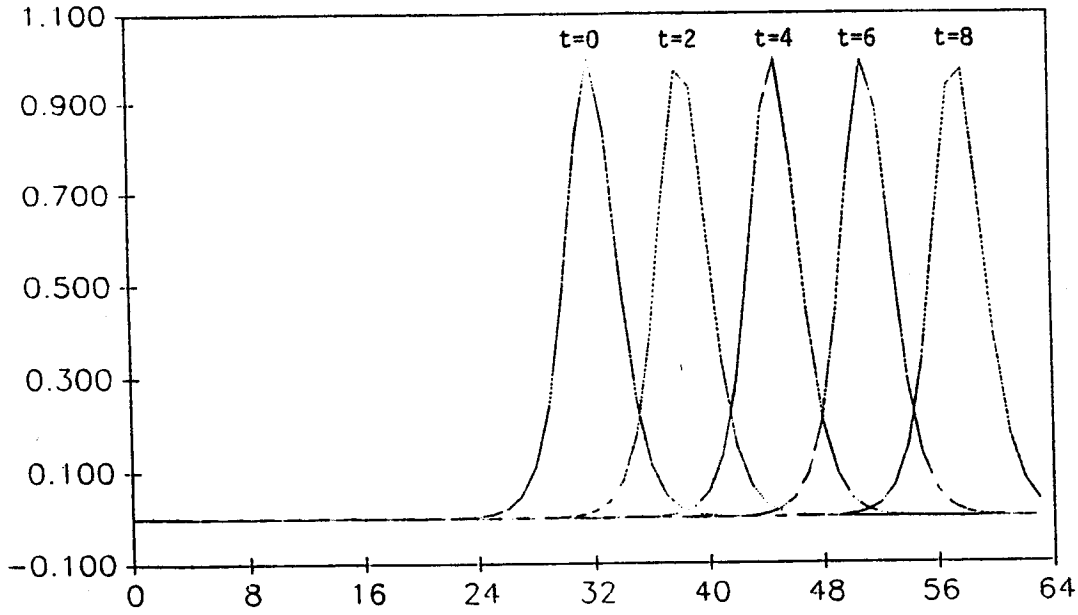


Figure 2. Superposition of Neural Simulation and analytical solution of the KdV equation.

which coresponds to a solitary wave initially located at $x = 0$ and moving with speed 2 [i.e., 3.2 Δ_x per unit time]. The integration time step was selected to be $\Delta_t = 0.005$, which satisfies the linearized stability criterion, Eq. (10). The numerical results obtained at different times, (i.e., $t = 0., 2., 4., 6., 8.$) are depicted in Figure 2, and compared with the analytical solution given by Eq. (2). As is obvious from Figure 2, the simulated neural solutions very closely overlap the analytical ones. It is impossible to distinguish the two curves on the plot. This indicates that Eq. (17), and the corresponding neural architecture provide an excellent computational framework for solving certain classes of PDEs, such as the KdV equation.

5. Conclusions

A neural algorithm for rapidely simulating a certain class of nonlinear wave phenomena using analog VLSI neural hardware was developed and presented. A neural architecture for solving partial differential equations of the Korteweg-de Vries type was obtained from a pseudospectral representation of the spatial dependence, along with a leap-frog scheme for the temporal evolution. The formalism was developed for full compatibility with CCD/CID analog VLSI neural circuits, enabling fast (potentially 10^{12} ops/sec/bit precision), massively parallel computation. Numerical simulations demonstrated the robustness of the proposed approach.

Acknowledgments

This research was carried out at the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology. Support for the work came from the National Aeronautics and Space Administration.

References

1. Agranat, A., Neugebauer, C.F., Nelson, R.D., and Yariv, A., *IEEE Trans. Circ. Syst.*, **37**, 1073 (1990).

2. Drazin, P.G., "Solitons", Cambridge Univ. Press., (1983).

3. Fornberg, B., and Whitham, G.B., *Philos. Trans. Roy. Soc., London,* **289**, 373 (1978).

4. Fornberg, B., *Geophysics*, **52**, 483 (1987).

5. Korteweg, D.J. and de Vries,G., *Phil. Mag.* **39**, 422 (1895).

6. Neugebauer, C.F., and Yariv, A., Proceedings IEEE International Joint Conference on Neural Networks, pp. 447-451, IEEE Press (1991).

7. Neugebauer, C.F., Agranat, A., Yariv, A., and Barhen, J.,"A High Speed Linear Algebra VLSI Engine", in preparation (1991).

8. Nouri, F.Z., and Sloan, D.M., *J. Compt. Phys.* **83**, 324 (1989).

9. van Wijngaarden, L., *J. Fluid Mech.,* **33**, 465 (1968).

10. Washimi, H., and Taniuti, T., *Phys. Rev. Lett.*, **17**, 996 (1966).

11. Whithman G.B. " Linear and Nonlinear Waves", John Wiley and Sons, New York, (1974).

12. Zabusky, N.J. "Nonlinear P.D.E.'s" (Ames, Ed.), Academic Press, New York, (1967).

A Modular Approach to the Design of Neural Networks for Fault Diagnosis in Power Systems†

C. Rodríguez, S. Rementería, C. Ruiz, A. Lafuente, J.I. Martín, J. Muguerza

Dpto. de Arquitectura y Tecnología de Computadores
Facultad de Informática. Universidad del País Vasco UPV-EHU.
Apdo. 649. 20080 Donostia. Spain.

LABEIN Technological Centre
Artificial Intelligence Department
Apdo. 1234. 48080 Bilbao. Spain.

ABSTRACT

In this work, a modular approach to the design of Neural Networks for fault diagnosis in electrical networks of real size is described. Modularization is strictly based on functional criteria, rather than topological criteria as it is usually found in literature.
This approach allows to obviate the problems inherent to this kind of applications (big amounts of information to be processed, high degree of uncertainty in data, changes in the topological features, sources of uncertainty). The most important characteristics of our model are: simplicity of the modules, replicability of the training results, easy adaptation to topological changes, and high scalability. Furthermore, it allows for parallel implementations. A portion of a real transportation electrical network has been simulated.

1 Introduction

One of the problems that come up when monitoring large power systems is the high number of alarms that can be generated during a disturbance. A certain skill is required from the operator to recognize events and the status of the components of a power system from the sequence of alarms reaching the dispatching centre. In this paper we will concentrate on the application of neural networks to the detection of faulty elements in a power system from information contained in alarm messages. This type of fault diagnosis can be reformulated as a pattern classification problem in which input data patterns (alarms) representing the behaviour of a physical system (electrical transportation networks) must be mapped to possible fault conditions.

The sequence of alarms generated in a disturbance is not a simple one. The operator can miss important information, as the one available from the alarm messages is noisy and uncertain. The following are the main sources of complexity in the application we will deal with in this paper:

(a) Big amounts of information: in a real world size electrical network, there are thousands of possible sources of alarms and hundreds of possible locations of a fault. The number of possible patterns is certainly unmanageable.

(b) Simultaneous disturbances: it may happen that the alarms reaching the system are due to unrelated and almost simultaneous disturbances (more than one cause for the observed effects). Sometimes it is not clear whether we are facing more than one disturbance or more than one effect of the same disturbance.

(c) Urgency of the situation: Answers must be provided fast enough. The reaction time is very important for serious disturbances, although usually there is not a fixed time constraint.

---

† This work has been partially supported by the CEC, ESPRIT project no. 5433, NEUFODI.

0 7803-0559-0/92 $3.00 © 1992 IEEE