Reprinted from PROCEEDINGS OF THE IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Philadelphia, Pennsylvania, May 4-7, 1987

THE ORIGINAL ADAPTIVE NEURAL NET BROOM-BALANCER

Bernard Widrow

Department of Electrical Engineering
Stanford University

ABSTRACT

This paper reviews work on artificial adaptive neurons done at Stanford University almost 25 years ago. The development of an adaptive linear threshold element (ADALINE) and several of of its mathematical properties are described. Applications were made to pattern recognition, speech recognition, classification of EKG waveforms, weather forecasting, and to control systems. An ADALINE element, a single artificial neuron, was used as a trainable controller to stabilize an inverted pendulum. This was the original adaptive broom-balancer.

The broom-balancer of the future will be a system that can learn to balance the "broom" by observing the real time control decisions made by the teacher, an expert who knows how to do the control function. The teacher has access to the critical control state variables. The trainable system, an ADALINE network, observes the cart and pendulum through visual (photocell array or TV) inputs and must of its own accord obtain the relevant state information by dynamic scene analysis. With an adequate training sample, the ADALINE net will be able to take over the control function from the teacher and thus become a trained expert. This research will lead to a new class of trainable expert systems.

INTRODUCTION

The original adaptive broom balancer was built at Stanford University in 1963 by Bernard Widrow, a young faculty member, and Fred. W. Smith, a Ph.D. candidate in Electrical Engineering. The purpose of the broom-balancer, a working electromechanical machine, was to demonstrate the capability of the ADALINE "neuron" in performing the task of optimal controller for an unstable system. At that time, the idea of modeling a component of the nervous system or of using an "artificial neuron" in an engineering application was highly controversial. Funding was difficult to obtain, and the capability of computer and electronic circuits and systems was very limited. In spite of all this, considerable progress was made in 1963, which was reported by Widrow and Smith [1] and by Smith [2,3]. Now, almost 25 years later, it makes sense to look back at this work and see where it might go in the future. The situation today is a different one. The study of neural nets for computing has gained a worldwide following, and the advent of VLSI technology makes their implementation practical.

The purpose of this paper is to review the status of ADALINE work at Stanford about 25 years ago, particularly with reference to applications in control systems. After a review of the old work, we then show how the concepts could be extended for the creation of TRAINABLE EXPERT SYSTEMS.

ADALINE, AN ADAPTIVE LOGIC ELEMENT

The basic building block of the systems to be considered is an adaptive threshold element, sometimes called an adaptive "neuron". For many years, we at Stanford University have called this element ADALINE (adaptive linear neuron). A functional diagram of this element is shown in Fig. 1. It includes an adjustable threshold level and the adaptation machinery which automatically adjusts the variable input weights. It has been demonstrated experimentally and theoretically that this element can be trained to react specifically to a wide variety of binary input signals and that it can be trained to generalize in certain ways, i.e, to react as desired with high reliability to inputs that it has not been specifically trained on.

In Fig. 1 the binary input signals on the input lines have values of +1 or -1 rather than the usual values of 1 or 0. Within the neuron shown, a linear combination of the input signals is formed. The weights are the gains $w_1, w_2, \ldots$, which could have both positive and negative values. The output signal is +1 if this weighted sum is greater than a certain threshold, and -1 otherwise. The threshold level is determined by the setting of the weight w_0, whose input is permanently connected to a +1 source. Varying w_0 varies a constant added to the linear

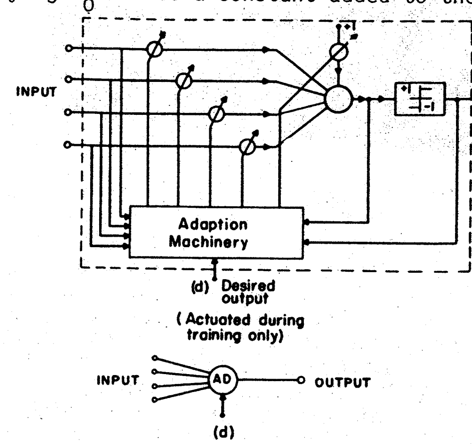

Fig. 1 The ADALINE Concept

CH2394-5/87/0000-0351 $1.00 © 1987 IEEE

combination of input signals.

For fixed gain settings, each of the 2^n possible combinations would cause either a +1 or a -1 output. Thus, all possible inputs are classified into two categories. The input-output relationship is determined by choice if the gains $w_0, \ldots, w_n$. In the adaptive neuron, these gains are set during the training procedure.

In general, there are 2^{2^n} different input-output relationships or truth functions by which the n input variables can be mapped into the single output variable. Only a subset of these, the linearly separable logic functions [4], can be realised by all possible choices of the gains. Although this subset is not all inclusive, it is a useful subset, and it is "searchable", i.e., the "best" function in many practical cases can be found iteratively without trying all functions within the subset. An iterative search procedure has been devised and is described below. This procedure is quite simple to implement, and can be analyzed by statistical methods that were originally developed for the analysis of adaptive sample-data systems.[5]

An adaptive pattern classification machine had been constructed for the purpose of illustrating adaptive behavior and artificial learning. A photograph of this machine, which is an adjustable threshold element (called "KNOBBY ADALINE") is shown in Fig. 2.

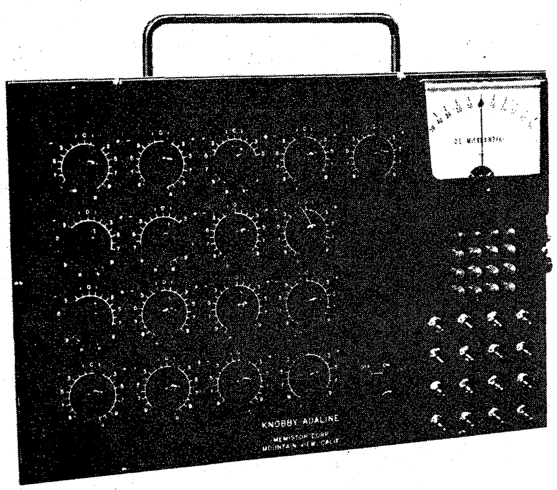

Fig. 2 "Knobby" ADALINE

During a training phase, simple geometric patterns were fed to the machine by setting the toggle switches in the 4 x 4 input switch array. All gains, including the threshold level were to be changed by the same absolute magnitude so that the analog error (the difference between the desired meter reading and the actual meter reading) was brought to zero. This was accomplished by changing each gain in the direction to diminish the error by 1/17. The 17 gains could be changed in any sequence, and after all changes were made, the error for the present input pattern was zero. The weights associated with switches up (+1 input signals) were incremented by rotation in the same direction as the desired meter needle rotation, the weights connected to switches in the down position were incremented opposite to the desired direction of rotation of the meter needle. The next pattern and its desired output was then presented, and the error was read. The same adjustment routine was followed and the error was brought to zero. If the first pattern had been reapplied at this point, the error would have been small but not necessarily zero. More patterns were inserted in like manner. Convergence was indicated by small errors. This is a least-mean-square [6] adaption procedure (LMS). It requires that adaption be made even if the quantized neuron output is correct. If, for example, the desired response is +1, the neuron is adapted to bring the analog response closer to the desired response, even if the analog response is more positive than +1.

The iterative training routine is purely mechanical. Electronic automation of the LMS algorithm andsoftware implementation of it has been widely practiced [7].

The results of a typical adaption on six noiseless patterns is given in Fig. 3. During adaption, the patterns were selected in a random sequence, and were classified into 3 categories. Each T was to be mapped to +30 on the meter dial, each G to 0 and each F to -30. As a measure of performance, after each adaptation, all six patterns were read in (without adaptation) and six analog errors were read. The sum of their squares denoted by $\sum e^2$ was computed and plotted. Fig. 3 shows the learning curve for the case in which all gains were initially zero.

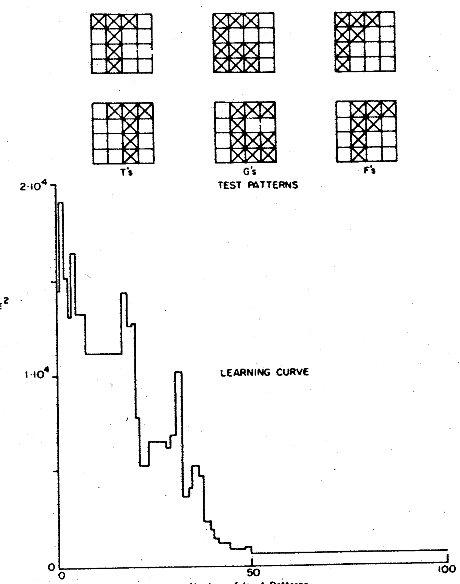

Fig. 3 A Learning Experiment

It is shown in [5,6] that making full correction with each adaption using the LMS procedure is in effect a stable "performance feedback" process having an adaptive time constant approximately equal to the number of weights. In the experiment of Fig. 3, the time constant is 17 adaptions. It is also shown that changing each weight by the same magnitude in the appropriate directions is equivalent to utilization of the method of steepest descent on a mean squre error surface.

THE ADAPTIVE MEMORY CAPACITY

An important question was, how many patterns or stimuli can the single adaptive neuron be trained to react to correctly at a time? This is a statistical question. Each pattern and desired output combination represents an inequality constraint on the weights. It is possible to have inconsistencies in sets of simultaneous inequalities just as with simultaneous equalities. When the patterns (i.e, the equations) are picked at random, the number which can be picked before inconsistency is created is a random variable. As few as 2 patterns can form a nonlinearly separable set, regardless of the pattern size.

A series of experiments was devised by J. S. Koford and R. J. Brown where patterns containing unbiased random bits and random desired responses were applied to ADALINES with varying numbers of inputs. It was found that the average number of random patterns that can be absorbed by an ADALINE is equal to twice the number of weights. This is one basic measure of memory capacity. It was proven by Cover [8] and Brown [9] that this experimental result is rigorously correct. Analytical curves showing the probability of being able to train-in N patterns as a function of N/(n+1) are presented in Fig. 4. Notice the sharpening of the break point of these curves at exactly the average capacity as the numbers of inputs to the ADALINE increases.

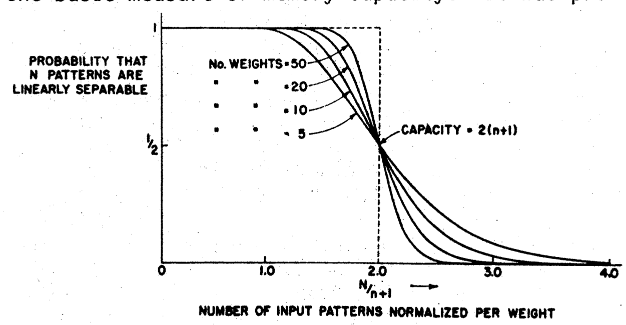

Fig. 4 Memory Capacity of an ADALINE

GENERALIZATION EXPERIMENTS WITH ADALINES

With suitable pattern-response examples and the proper training procedures, generalizations can be trained into ADALINES. The kinds of generalizations that had been done were insensitivity to noise and to translation, rotation, and size. ADALINES were trained to react consistently to a training set of patterns for all possible positions, for example, and then they reacted consistently in all positions with high reliability on new patterns never seen before.

Generalization with Respect to Noise

Statistical separation of patterns consisting of a finite set of basic "prototypes" and noisy versions of these basic patterns can be readily accomplished by the single ADALINE after training on the basic patterns and/or samples of the noisy patterns. A new pattern would be associated with one of the prototype classes by proximity in a Hamming distance sense.

With the objective of minimizing the probability of incorrect classification, there is an optimum set of weights that would result from training on a very large sample. The effect of training on a small sample set can be summarized with the following formula, derived in [5,6].

$$M = \frac{(n+1)}{N} \qquad (1)$$

The number of training samples is N, randomly selected from all possible samples, and the total number of weights is (n + 1). The quantity M is called the "misadjustment". In this context, it is the per unit increase in error probability, based on a minimum error probability attainable by training on a very large sample. This formula leads directly to the idea that the number of patterns required to train an ADALINE to discriminate noisy patterns is about five times (making M only 20 percent) the number of weights. The number of training patterns required to produce this form of generalization is of the order of twice the statistical memory capacity.

Generalization with Respect to Rotation of Patterns

Insensitivity to rotation by 90° is a characteristic that can be perfectly trained into an ADALINE. An experiment was made as depicted in Fig. 5 by using the 4 x 4 KNOBBY ADALINE shown in Fig. 2. C's rotated in all four positions were trained-in to give +1 response, while T's were trained-in to give the -1 response in all four rotations. The initial weights were set to zero, and during training, the minimum mean-square error adaption procedure with an adaptive time constant of 32 patterns was utilized. The process converged with the desired responses trained-in precisely, and the set of weights shown in Fig. 5 resulted. Without further training, new patterns totally unrelated to the training patterns were inserted, and it was observed that not only were

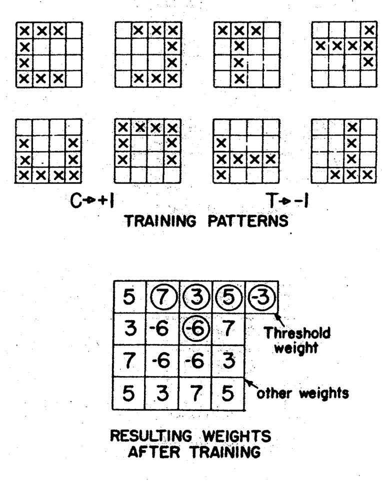

Fig. 5 Training Insensitivity to Pattern Rotation

the decisions made by the ADALINE perfectly consistent for each pattern over the four rotations, but the four meter readings (confidence levels or analog outputs) for each pattern were identical. The reason for this is simple: Rotation of the weights by 90° yields an identical set of weights. Let the a-matix represent the set of weights (not including the threshold weight). The threshold weight remains the same for all rotations. The superscript R represents rotation by 90°.

$$[a] = [a]^R = \left[[a]^R\right]^R = \left[\left[[a]^R\right]^R\right]^R \tag{2}$$

Generalization with Respect to Left-Right Translation

Perfect solutions to the problem of training an ADALINE to be insensitive to left-right pattern shift exist. A solution requires the columns of the a-matrix to be identical. On a 4 x 4 input array, there is a choice of 4 independent weights, each choice setting a row of weight values. It follows that the statistical discrimination capacity subject to the constraint of insensitivity to left-right translation is that of a 4-input ADALINE or 8 basic patterns. The total capacity of the 4 x 4 ADALINE is 32 patterns, and this corresponds to the four positional possibilities for each of the 8 basic patterns. Patterns can be placed in four positions by considering the input pattern space to be continuous and folded over a cylinder having a vertical axis.

Generalization with Respect to Pattern Size

An ADALINE can be trained to be highly insensitive to pattern size. The training procedure required slow minimum mean-square-error adaption. In Fig. 6, a set of "small" and "large" patterns is shown that comprised examples for the training experiment. On a 3 x 3 array in the upper left hand corner of a 4 x 4, a T and a C were inserted as shown. In the full 4 x 4 array, expanded versions of these patterns were trained-in to give corresponding responses. After training, it was found that new patterns gave widely fluctuating analog responses. For about 90 percent of new pattern inputs, the same binary response resulted for the small as for the large versions, and the corresponding confidence levels were extremely close.

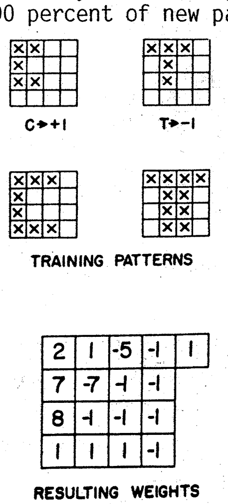

Fig. 6 Training Insensitivity to Pattern Size

To be perfectly insensitive to size, the weights of an ADALINE must be such that an element of area of the small pattern "sees" the same total weight (input patterns are thought of as continuous two-dimensional functions and weights are thought of as continuous distribution functions) as the corresponding area element of the large pattern that it maps into. It can be shown that perfect solutions result when the weight function radiates from a point and has an intensity that decays with an inverse-square law. These effects are approximated in the weights of Fig. 6.

PATTERN RECOGNITION APPLICATIONS

In addition to the applications to automatic control systems which will be described in the next section, the above principles have been applied to weather forecasting [10], speech recognition [11], and diagnosis of EKG waveforms [12,13].

APPLICATION TO CONTROL SYSTEMS

The state of a dynamic system can be completely described at any instant by the values of the state variables of the system. (The state variables of a control system are such quantities as the error, the error derivative, etc.) A control decision therefore need depend only on the present values of the state variables. The value of each state variable can be encoded as a sequence of binary digits. The collection of these encoded state variables forms a pattern. Proper control of a dynamic system by an ADALINE or a network of ADALINES becomes a matter of the proper classification of the patterns which represent the different states of a dynamic system. Just as an ADALINE can be taught to classify patterns into two groups, it can also be taught to control a dynamic system in a "bang-bang" or +1, -1 manner.

When the state variables are encoded using what has been called a "linearly independent code", the task of learning control strategies is quite natural for an ADALINE.

(i) The large sets of patterns representing the control strategy for all possible regions of state space are often linearly separable, or separable with simple ADALINE networks. The number of control patterns which the ADALINE is able to correctly classify is generally an order of magnitude or more greater than its statistical capacity.

(ii) The ADALINE generalizes in a known and predictable way. Namely, the ADALINE can correctly classify all the patterns of a control strategy after learning to correctly classify only the patterns bordering on the switching surface.

Because of this strong generalizing property and because of the special interrelationships among the many patterns, the ADALINE is much easier to train than it would be for a similar number of random or near random patterns.

The Trainable Controller

Fig. 7 shows in block diagram form the general situation in which an ADALINE would be used as a trainable controller for a dynamic system. The state variables $y_1,...y_m$ are assumed to be the system error.

The teaching controller is capable of performing the control function while supplying the desired output to the ADALINE during its training period. This controller could be an automatic controller or possibly a human expert. The ADALINE controller and the teaching controller need not have the same inputs, provided both receive the same or related information. For instance, the ADALINE controller could be receiving the state variables as electronic signals while a human teacher could be receiving information about the system by actually watching its motions.

For the purposes of discussion the teacher will be assumed to be represented by a function $f(y_1,...,y_m)$. The switching surface $f(y_1,...y_m) = 0$ describes the transition where the teacher changes his reaction from "force plus" to "force minus". During the training, the ADALINE analog output $\hat{f}(y_1,...y_m)$ is adjusted so that its switching surface $\hat{f}(y_1,...y_m) = 0$ is made to approximate the switching surface of the teacher.

The ADALINE controller consists of an encoder and an ADALINE. For simplicity, a single ADALINE is shown here in the controller; more typically a network might be used. The network with its encoder is basically a trainable function generator which forms the function $\hat{f}(y_1,...y_m)$. The pattern inputs to the network change continually as the state variables change. The encoder produces patterns by quantizing or dividing the range over which each of the state variables varies into a finite number of zones. Each zone of a state variable y_i is represented by a binary number or partial pattern. The m partial patterns make up the total pattern.

Fig. 8 illustrates the quantization of a two-dimensional state space. Each square in the figure is represented by a particular pattern for the ADALINE. The continuous curve $f(y_1,y_2) = 0$ represents a typical desired switching surface (a curved line in this case). The jagged curve $\hat{f}(y_1y_2) = 0$ is the switching curve that an ADALINE controller might use to approximate the teaching controller.

The system has two modes of operation:
(i) During the training mode, the teaching

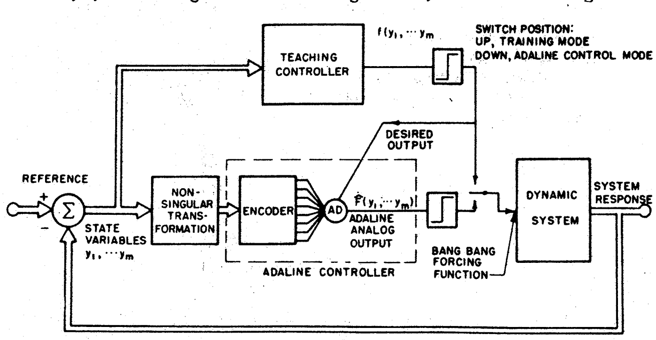

Fig. 7 ADALINE Controller: A "Trainable Expert System"

controller, an expert, controls the dynamic system. The adapt logic in the ADALINE continuously compares the binary output of the ADALINE with that of the teacher. Whenever they differ, the ADALINE is adapted in the direction which would make them agree. Because the patterns change rapidly, there may not be time for a full correction. However, the pattern is bound to recur, at which time adaption can be continued. During the training mode the ADALINE controller "watches" the teacher in order to zero the error after various large disturbances or large initial conditions.

(ii) During the ADALINE control mode, the teaching controller is not used and may be completely removed from the system. The ADALINE then is a "trained expert system", performing the control function.

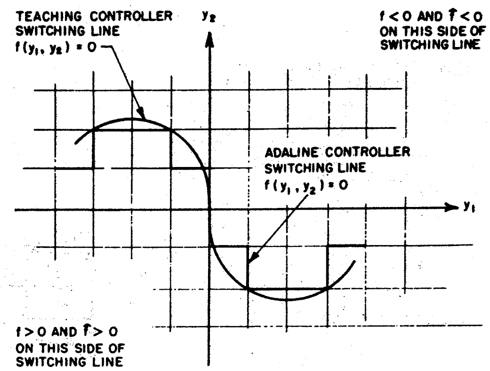

Fig. 8 An ADALINE Switching Function and Teacher's Switching Function

Coding

The choice of codes used to represent the values of the state variables largely determines how well the ADALINE controller will be able to imitate its teacher. Fig. 9 illustrates two possible "linearly independent codes". A linearly independent code is any code which has a nonsingular partial pattern matrix. This matrix has the partial patterns as rows plus a column of ones (if necessary). The partial pattern matrix for the codes of Figs. 10a and b are respectively:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Both matrices are obviously invertible. When linearly independent coding is used, the ADALINE will be able to exactly imitate (except for quantization effects) any teacher whose function does not contain cross-product terms, i.e., terms of the form $y_iy_j, i=j$, regardless of the number of patterns. This is a consequence of encoding each state variable independent of the others. An ADALINE with encoded inputs is shown in Fig. 10.

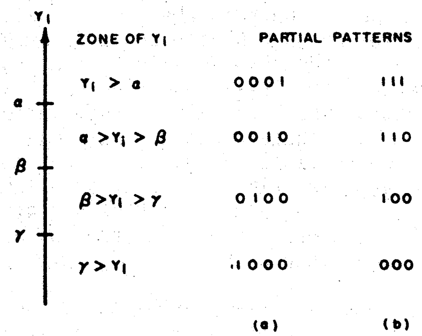

Fig. 9 Linearly Independent Encoding of an Analog State Variable

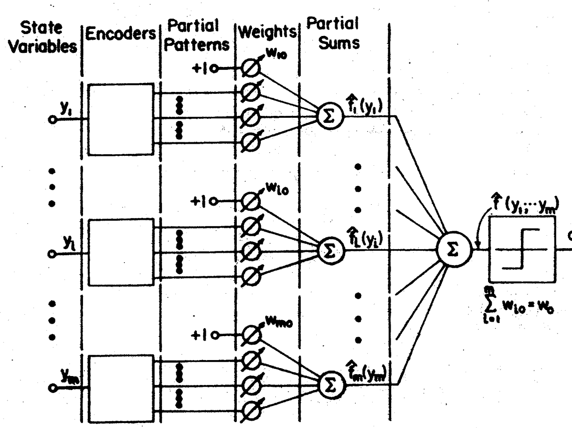

Fig. 10 An ADALINE with Binary Encoded Inputs

An Example of An Adaline Controller

The above ideas can best be illustrated by showing how an ADALINE controller would control the oscillatory undamped second-order system with differential equation

The minimum-time optimum-switching curve is the well-known Bushaw [14] switching curve, the teacher switching curve shown in Fig. 11. A controller containing one ADALINE is capable of closely approximating the nonlinear function of the optimum controller. The switching curve of the ADALINE controller is shown in Fig. 11 with functions $f_1(y_1), \hat{f}_1(y_1) f_2(y_2)$, and $\hat{f}_2(y_2)$.

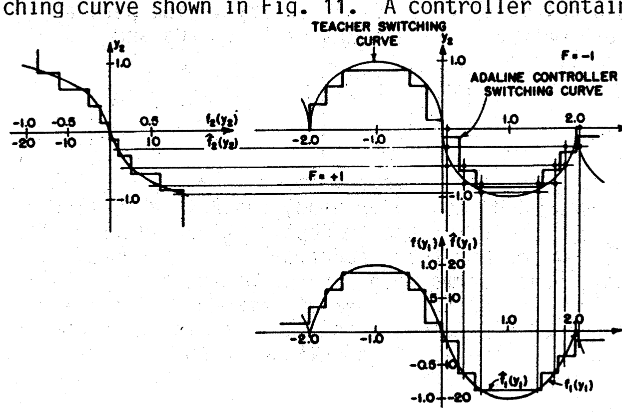

Fig. 11 ADALINE Approximation of the Bushaw Switching Curve

"Broom-Balancing Machine"

To demonstrate these ideas, a relatively complex dynamic system with an ADALINE controller had been assembled in 1963. The dynamic system was a motorized cart carrying an inverted pendulum. The controller for the system was required to keep the pendulum balanced and keep the cart within certain bounds by applying a horizontal force to the cart. An ADALINE circuit utilizing electrochemical weights called Memistors [15] was used in the training controller. Fig. 7 gives a block diagram of the dynamic system and its controllers. The nonsingular transformation in Fig. 7 is the identity transformation in this case.

The cart and pendulum system was an undamped and inherently unstable fourth-order dynamic system. The four-state variables were the angle of the pendulum from vertical, θ; the rate of change of angle θ; the position of the cart, x; and the rate of change of position x. These and other relevant quantities are defined in Fig. 12. The linearized differential equations representing this system were:

$$\ddot{\theta} = \frac{3g}{4l}\theta - \frac{3}{4lM}F$$

$$\ddot{x} = \frac{1}{M}F$$

It was assumed that there was no damping, and that the reaction of the pendulum motions on the cart was negligible.

The teaching controller used in these experiments had a linear switching surface approximately

$$f = -2.0\,\dot{\theta} - 1.0\,\theta + 1.0\dot{x} + 1.0x$$

The ADALINE controller contains one 24-input ADALINE. The range of each of the state variables was divided into seven approximately equal zones. The state variables were encoded into 6-bit partial patterns using a linearly independent code similar to the one illustrated in Fig. 9b. The controller was taught by having it observe the teacher return the system to the origin of state space after it had received various large disturbances.

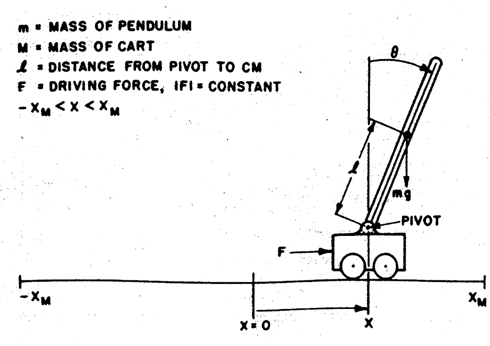

Fig. 12 The Cart with Inverted Pendulum

THE BROOM-BALANCER OF THE FUTURE: A TRAINABLE EXPERT SYSTEM

In the utilization of present-day rule based expert systems, decision rules must always be known for the application of interest. Sometimes there are no rules however. The rules are either not explicit or they simply do not exist. For such applications, trainable expert systems might be usable. Rather than working with decision rules, an adaptive expert system might observe the decisions made by a human expert. Looking over the expert's shoulders, an adaptive system can learn to make similar decisions to those of the human when facing given sets of input circumstances.

A sample application for a trainable expert system is the following: Driving a car down a narrow congested street is done every day by experienced human drivers. Imagine an adaptive pattern recognition system equipped with a visual input (a retina of photo receptors or a TV camera) looking through the windshield of the car seeing the dynamic scene confronting the driver and at the same time observing the driver's responses via the steering wheel, brake pedal, and accelerator pedal. With an adequate training sample, the adaptive system should be able to make decisions very much like those of the teacher.

Instead of developing at this time a system to learn to drive a car in traffic, a simpler and more quantifiable problem to begin with is the broom-balancing problem. Refering to Fig. 13, a trainable controller consisting of an ADALINE network learns to respond like the teacher by observing the teacher's control decisions. However, the teacher has the proper state variables as its set of inputs. The ADALINE net has an "eyeball" input, a photocell retina or a TV camera, with which to observe the positions and motions of the cart and the pendulum. Acting on its own, the ADALINE will need to obtain the equivalent state-variable information from visual observations of the scene and its time rate of change, the scene being the picture of the cart and the pendulum. With an adequate training sample, the ADALINE net will be able to take over the control function from the teacher and thus become a trained expert.

It is not known how to design the ADALINE network to solve the broom-balancing problem. Doing the research to develop an adaptive controller of this type will enable us to progress toward the goal of being able to design trainable expert systems for much more general applications. Members of governmental laboratories, industrial laboratories, and university laboratories who would like to engage in this work are encouraged to contact Professor Widrow.

REFERENCES

[1] B. Widrow and F.W. Smith, "Pattern recognizing control systems," Computer and Information Sciences (COINS) Symposium Proceedings, Spartan Books, Wash. DC, 1963.

[2] F.W. Smith, "A trainable nonlinear function generator," IEEE Trans. on Automatic Control, Vol. AC-11, No. 2, pp. 212-218, April, 1966.

[3] F.W. Smith, "Design of quasi-optimal minimum-time controllers," IEEE Trans. on Automatic Control, Vol. AC-11, No. 1, pp. 71-77, Jan. 1966.

[4] N. Nilsson, Learning Machines, New York: McGraw Hill Book Co., 1965.

[5] B. Widrow, "Adaptive sampled-data systems - a statistical theory of adaption," 1959 WESCON Convention Record, Part 4.

[6] B. Widrow and M.E. Hoff, "Adaptive switching circuits," 1960 WESCON Convention Record, Part IV, Aug. 23, 1960, pp. 96-104.

[7] B. Widrow and S.D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall, Inc., 1985.

[8] T.M. Cover, "Geometrical and statistical properties of linear threshold devices," Stanford Electronics Labs., Stanford, CA., Tech. Report 6107-1, Ph.D. thesis, May 1964.

[9] R.J. Brown, "Adaptive multiple-output threshold systems and their storage capacities," Stanford Electronics Labs., Tech. Report 6771-1, Ph.D. thesis, June 1969.

[10] M.J.C. Hu, "A trainable weather-forecasting system," Stanford Electronics Labs, Stanford, CA., Tech. Report 6759-1, June 1963.

[11] L.R. Talbert, G.F. Groner, J.S. Koford, R.J. Brown, P.R. Low, and C.H. Mays, "A real-time adaptive speech recognition system," Stanford Electronics Labs., Stanford, CA., Tech. Report 6760-1, May 1963.

[12] D.F. Specht, "Generation of polynomial discrimination functions for pattern recognition," IEEE Trans. on Electronic Computers, Vol. EC-16, No. 3, pp. 308-319, June 1967.

[13] D.F. Specht, "Vectrocardiographic diagnosis using the polynomial discriminant method of pattern recognition," IEEE Trans. on Bio-Medical Engineering, Vol. BME-14, No. 2, pp. 90-95, April 1967.

[14] L.S. Pontryagin, et. al., The Mathematical Theory of Optimal Processes, (translation), New York: Interscience, 1962.

[15] B. Widrow, "An adaptive 'Adaline' neuron using chemical 'memistors'," Stanford Electronics Labs., Stanford, CA., Tech. Report 1553-2, Oct. 17, 1960.

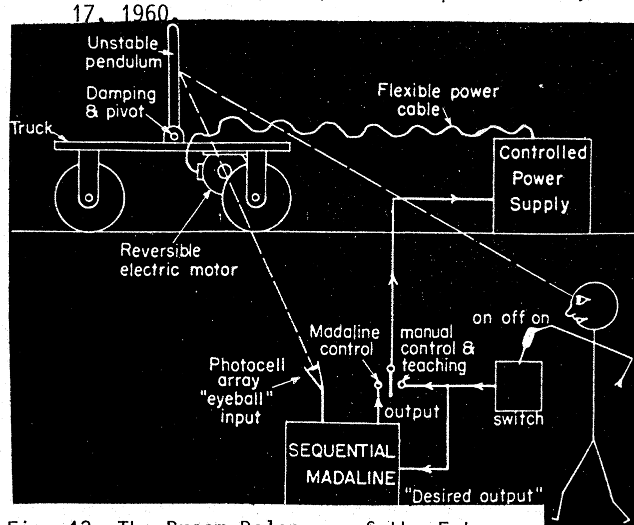

Fig. 13 The Broom-Balancer of the Future - A "Trainable Expert System"