1964

A study of the application of pattern recognition to automatic control.

Murray, James.
Monterey, California: U.S. Naval Postgraduate School

http://hdl.handle.net/10945/13136
A STUDY OF THE APPLICATION OF PATTERN RECOGNITION TO AUTOMATIC CONTROL

JAMES MURRAY.
A STUDY OF THE APPLICATION
OF PATTERN RECOGNITION
TO AUTOMATIC CONTROL

* * * * *

James Murray
A STUDY OF THE APPLICATION
OF PATTERN RECOGNITION
TO AUTOMATIC CONTROL

by

James Murray

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

United States Naval Postgraduate School
Monterey, California
1964
A STUDY OF THE APPLICATION
OF PATTERN RECOGNITION
TO AUTOMATIC CONTROL

by

James Murray

This work is accepted as fulfilling
the thesis requirements for the degree of
MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING
from the
United States Naval Postgraduate School
ACKNOWLEDGEMENTS

Dr. John R. Ward stimulated the author's interest in pattern recognition and was a source of assistance and encouragement throughout this work.
ABSTRACT

Pattern recognition schemes have been concerned, mainly, with the problem of identifying alphabetic characters, or numerals. To this end adaptive pattern recognition devices have been trained to recognize such characters. The topic discussed here is the training of an adaptive pattern recognition device, Adaline, to mimic the performance of the controller of a plant.

This study discusses Adaline and the minimum square error method of adaption. Adaline is trained by observing the behaviour of the controller in numerous situations. The problem of coding the information to be presented to Adaline is discussed, and finally, a suitably trained Adaline takes control of the plant.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>v</td>
</tr>
<tr>
<td>I Pattern Recognition</td>
<td>1</td>
</tr>
<tr>
<td>1.1 The Adaptive Mechanism</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Separability of Patterns</td>
<td>5</td>
</tr>
<tr>
<td>1.3 Experimental Simulation of Adaline</td>
<td>11</td>
</tr>
<tr>
<td>II A Note on Minimum Square Error Adaption</td>
<td>20</td>
</tr>
<tr>
<td>2.1 Analytical Minimization of $\sum e^2$</td>
<td>21</td>
</tr>
<tr>
<td>2.2 Steepest Descent Minimization of $\sum e^2$</td>
<td>26</td>
</tr>
<tr>
<td>2.3 Modified Steepest Descent</td>
<td>28</td>
</tr>
<tr>
<td>III Adaline in Control of a Plant</td>
<td>30</td>
</tr>
<tr>
<td>3.1 The Equations of the Plant</td>
<td>30</td>
</tr>
<tr>
<td>3.2 Coding of the Phase Plane and the Training of Adaline</td>
<td>34</td>
</tr>
<tr>
<td>3.3 Adaline Acting as the Controller</td>
<td>45</td>
</tr>
<tr>
<td>IV Conclusions</td>
<td>57</td>
</tr>
<tr>
<td>4.1 Adaline and a Dead Zone</td>
<td>57</td>
</tr>
<tr>
<td>4.2 Coarse and Fine Division of the Phase Plane</td>
<td>58</td>
</tr>
<tr>
<td>4.3 Polar Coding Scheme</td>
<td>59</td>
</tr>
<tr>
<td>4.4 General Remarks</td>
<td>63</td>
</tr>
<tr>
<td>Bibliography</td>
<td>68</td>
</tr>
</tbody>
</table>
INTRODUCTION

The problem of automatic character recognition has received a great deal of attention recently. Many schemes have been proposed and they can be divided into two main groups, "open loop" types and "closed loop" types. Open loop schemes compare the character to be recognized with a library stock and make a decision on this basis. These schemes are very useful when the characters are, essentially, standard—possibly typewritten. Closed loop schemes are trained to recognize characters by applying as many patterns as possible to the machine while "teaching" it to generate the correct answers before asking it to operate alone. These schemes have been described by such names as "linear decision network" or "adaptive linear neuron" (Adaline). Their advantage over closed loop schemes is that they are able to attempt the classification of patterns other than those on which the machine has been trained.

The latter scheme has application to process control. The human operator of a steel rolling mill is presented with such data as speed and temperature of the incoming billet. On the basis of this knowledge and past experience he can adjust the rolls to produce the desired sheet steel. A pattern recognition device could be placed beside him and be trained to recognize suitably coded patterns of information about the billet and thus to produce the same "response" as the operator in control of the rolls.
The present investigation considers a simple relay activated plant, which is already controllable, and considers the training of a pattern recognition device to produce the same result as the relay controller. The block diagram of a stable plant is shown in Figure (a), where $d(\tau)$ is the control signal which causes the relay to apply driving power to the plant. If the linear control system error is defined to be the difference between desired and actual output at any given time, then a convenient way of describing the behaviour of a plant is to consider the system error and its derivatives as functions of time.

The pattern recognizer is, therefore, presented with suitably coded patterns of error, error rate (and higher derivatives if necessary) and it is then trained to produce an output signal, $a(\tau)$, close to $d(\tau)$. This is indicated in Figure (b).

---

Figure (a)

Figure (b)
Training involves presenting the pattern recognizer with as many typical operating conditions as possible, while adjusting it so that $a(t)$ matches $d(t)$ as closely as possible over the entire range of conditions. After the training period the compensator can be disconnected as the learning machine takes its place. This is shown in Figure (c).

![Diagram](image)

**Figure (c)**

The present study discusses the properties of the pattern recognition device (in Chapter I) and a specific method of adaption (in Chapter II.). The training of the device, the coding of error and error rate and the control of the plant by the learning machine are discussed in Chapter III. The conclusions appear in Chapter IV.
CHAPTER I

PATTERN RECOGNITION

An adaptive pattern recognition device of the type to be considered has four essential components: a sensory unit, an association unit, a response unit and an adjustment unit. The whole device is shown in Figure 1. It is expected of such a device that it can be trained to recognize stimuli or patterns which are part of the environment in which it is placed.

Figure 1

The sensory unit is a transducer which produces, possibly, a set of electrical signals in response to a visual or audio pattern. If the patterns which are to be recognized are alphabetical or numerical, then they could be displayed on a matrix of photo-cells which could, in turn, generate positive or negative voltages, depending on the presence or absence of an element of the pattern. This is shown in Figure 2.

Figure 2
Throughout the remainder of the discussion the characteristics of the transducer will be bypassed and the pattern, or input, for the association unit will be considered to be simply an array of positive and negative voltages of unit magnitude.

The association unit is a logical decision element which produces an output on receipt of the input pattern. It will be assumed to consist of a set of adjustable weights of number \( n+1 \) when the number of elements of the input pattern is \( n \). The \( n \) elements of the input pattern are supplied to the weights \( W_1, W_2, W_3, \ldots, W_n \). The weight, \( W_0 \), is called the threshold and its input is fixed at \( +1 \). The values of the weights and threshold are determined by previous training.

In this study all the weights and threshold will have any past experience removed by setting them to zero before a training sequence begins. If the elements of a pattern are supplied to the device, the sum of the outputs of the weights and the threshold is called the analogue output, \( \alpha \).

This is shown in Figure 3.

![Figure 3](image-url)
The function of the response unit is to indicate which decision has been made on the pattern applied to the sensor. In this discussion the characteristics of the response unit will be bypassed and the analogue output, \( a \), will be used as an indication of the pattern which has been applied.

These ideas have been discussed by many authors, [1, 2, 3, 4, 5]. Widrow has named the device "Adaline" (Adaptive linear neuron).

1.1 The Adaptive Mechanism

The problem of pattern recognition using Adaline could be stated in the following way: given \( m \) input patterns, each having \( n \) pattern elements, separate them into two classes—some of the patterns being mapped to positive values of analogue output and the remainder to negative values of analogue output.

Each input pattern can be thought of as a column vector, 

\[
[X]_i, \quad \text{where} \quad [X]_i = \begin{bmatrix} X_{i1} \\ X_{i2} \\ \vdots \\ X_{ij} \\ \vdots \\ X_{in} \end{bmatrix}
\]

with \( X_{ij} = \pm 1 \), and \([X]_i\), the \( i \)th pattern of \( m \). If the \( i \)th pattern, \([X]_i\), is applied to the device, the resulting analogue output, \( a_i \), is

\[
a_i = W_1 X_{i1} + W_2 X_{i2} + \ldots + W_j X_{ij} + \ldots + W_n X_{in} + W_0
\]

\[
= \sum_{j=1}^{n} W_j X_{ij} + W_0
\]

where \( W_0 \) and \( W_1, W_2, \ldots, W_j, \ldots, W_n \) are the threshold and
set of weights respectively. To be a competent pattern recognizer Adaline must be trained, in some way, to make $a_i$ match $d_i$, where $d_i$ is the required or desired output for the $i$th pattern. After training, which entails adaption of the weights, it is hoped that when presented with a pattern, $[X]_i$, Adaline will make the correct classification by generating an output as nearly as possible equal to $d_i$.

If $(d_i - a_i)$ is defined as the analogue error, $e_i$, any training scheme which will make all the $e_i$ small is worth considering. The adaption scheme studied here attempts to minimize the sum of the squares of the errors for all the patterns presented. i.e. the scheme minimizes $\sum_{i=1}^{m} e_i^2$. This topic is discussed fully in Chapter II. Other adaption schemes are discussed by Treado [6]. For each pattern, $[X]_i$, the analogue error, $e_i$, is measured. An equal adjustment is then made to each weight. This adjustment is proportional to the error, $e_i$, and is of such a sign as will reduce $e_i$ and $\sum_{i=1}^{m} e_i^2$. Thus the change to the weight, $W_j$, at each step is given by:

$$\Delta W_{ij} = g e_i X_{ij}$$

1.1

and for the threshold, $W_o$, it is:

$$\Delta W_o = g e_i$$

where $g$ is a constant and where it will be assumed that the weights can be varied continuously. The total adjustment after all patterns have been presented once is:

$$\Delta W_j = g \sum_{i=1}^{m} e_i X_{ij}$$

1.2

and

$$\Delta W_o = g \sum_{i=1}^{m} e_i$$

The adjustments of equation 1.1 are made if $Q_i$ is not equal
to $\delta_i$. Adjustments could, therefore, continue until the analogue error, $e_i$, is zero. Hence the minimum of $\sum_{i=1}^{n} e_i^2$ found, if training is carried on for long enough, will be zero. (See also Chapter II)

Three aspects of equation 1.2 will be examined in a later section:

1. The speed with which the error, $e_i$, is reduced as a succession of training patterns is presented and the dependence of this speed of convergence on $\beta$.

2. The fact that the error, $e_i$, in response to one of the $m$ patterns, may be too large even though $\sum_{i=1}^{n} e_i^2$ is acceptably small, but not zero.

3. The fact that the constant, $\beta$, can take on values which make $\sum_{i=1}^{n} e_i^2$ diverge. If the constant, $\beta$, takes the value, $k/\eta^2$, where $\eta$ is the number of weights and $k$ is a constant, Mays [7] showed that $k \leq 2$ for convergence.

1.2 Separability of Patterns

Adaline can be considered to be the realization of a linear decision function. Input patterns can be regarded as sets of points in $n$ space and a linear decision function is any partitioning of the space by a hyperplane of dimension $n - 1$. The pattern recognition problem requires the selection of a set of weights which will define an appropriate hyperplane. In an adaption scheme which uses the mean square error adaption rules the weights defining a separating hyperplane are those
which cause \( \sum_{i=1}^{m} \varepsilon_i \) and the analogue errors, \( \varepsilon_i \), for each pattern, \([x]_i\), to be zero. Hence weights must be chosen to satisfy the following equations:

\[
X_{i1} W_1 + X_{i2} W_2 + \ldots + X_{i2} W_j + \ldots + X_{in} W_n + w_0 = d_i
\]

\[
X_{i2} W_1 + X_{i2} W_2 + \ldots + X_{i2} W_j + \ldots + X_{in} W_n + w_0 = d_i
\]

\[
X_{i1} W_1 + X_{i2} W_2 + \ldots + X_{i2} W_j + \ldots + X_{in} W_n + w_0 = d_i
\]

The iterative training scheme of section 1.3, if carried to its conclusion, will yield a set of weights defined by the above equations if it is possible to map the patterns, \([x]_1, \ldots, [x]_i, \ldots, [x]_m\), to outputs of \( d_1, d_2, \ldots, d_i, \ldots, d_m \). If it is possible to map all the patterns with which Adaline is trained, to the desired outputs then the patterns are said to be separable. After training has been completed with a set of separable patterns, the values found for the threshold and the weights are fixed at \( W_0^f, W_1^f, W_2^f, \ldots, W_n^f \) and the equation of the hyperplane dividing \( \mathbb{R} \) space is, therefore,

\[
X_1 W_1^f + X_2 W_2^f + \ldots + X_j W_j^f + \ldots + X_n W_n^f + w_0^f = 0
\]

where \( X_1, X_2, \ldots, X_n \) are the coordinate axes of \( \mathbb{R} \) space.
If the $i$ th pattern, $[x]_i$, with coordinates, $x_{i1}, x_{i2}, \ldots, x_{in}$, in $\mathbb{H}$ space, is now presented, it should yield an analogue output,

$$a_i = x_{i1}w^f_1 + x_{i2}w^f_2 + \cdots + x_{in}w^f_n + w^f_0$$

which is approximately equal to the desired output, $d_i$. Equation 1.5 is the equation of a hyperplane on one side of the dividing hyperplane given by equation 1.4. Hence another definition of separability is that a set of patterns can be said to be separable if one group of them lie on one or more parallel hyperplanes on one side of, and parallel to, the dividing hyperplane and the other group lie on one or more parallel hyperplanes on the other side of, and parallel to, the dividing hyperplane. If a further restriction is placed on the minimum mean square error adaptation scheme, the definition of separability is even simpler. Consider an example with $m$ patterns (as described in equation 1.3) where $k$ of them have desired outputs of $+d$ and $l$ of them have desired outputs of $-d$ (and $k + l = m$). If the patterns are separable, then the threshold and weights, $w^f_0, w^f_1, w^f_2, \ldots, w^f_n$, can be found. Further, after training and on presenting the $k$ patterns to these weights, a hyperplane on one side of the dividing hyperplane is defined. If the $l$ patterns are now presented a hyperplane on the other side of the dividing hyperplane is defined. Both hyperplanes are parallel to the dividing hyperplane and equal distances from it. These ideas can be clarified if specific examples in 2-space are considered. In 2-space Adaline consists of two weights, $w_1$ and $w_2$, and a threshold, $w_0$. In the example illustrated in Figure 4 there
are two patterns:

\[
\begin{bmatrix}
\mathbf{X}_1 \\
\mathbf{X}_2
\end{bmatrix}
\begin{align*}
X_{11} &= 1 \\
X_{12} &= 1
\end{align*}
\begin{align*}
d_1 &= d \\
\mathbf{d}_1 &= d
\end{align*}
\begin{align*}
X_{21} &= 1 \\
X_{22} &= 1
\end{align*}
\begin{align*}
d_2 &= -d \\
\mathbf{d}_2 &= -d
\end{align*}
\begin{align*}
X_{31} &= 1 \\
X_{32} &= 1
\end{align*}
\begin{align*}
d_3 &= -d \\
\mathbf{d}_3 &= -d
\end{align*}
\begin{align*}
\mathbf{X}_1 \\
\mathbf{X}_2 \\
\mathbf{X}_3
\end{align*}

From equation 1.3 the equations defining the weights required for separation are:

\[
W_1 + W_2 + W_0 = d
\]
\[
- W_1 + W_2 + W_0 = -d
\]

These can be solved only by choosing a value for one of the weights and solving for the other two. The equations are consistent\(^1\) but yield an infinite number of dividing lines which pass the point, \(X_1 = 0\), \(X_2 = 0\).

In the example illustrated in Figure 5 there are three patterns:

\[
\begin{bmatrix}
\mathbf{X}_1 \\
\mathbf{X}_2 \\
\mathbf{X}_3
\end{bmatrix}
\begin{align*}
X_{11} &= 1 \\
X_{12} &= 1
\end{align*}
\begin{align*}
d_1 &= d \\
\mathbf{d}_1 &= d
\end{align*}
\begin{align*}
X_{21} &= 1 \\
X_{22} &= 1
\end{align*}
\begin{align*}
d_2 &= -d \\
\mathbf{d}_2 &= -d
\end{align*}
\begin{align*}
X_{31} &= 1 \\
X_{32} &= 1
\end{align*}
\begin{align*}
d_3 &= -d \\
\mathbf{d}_3 &= -d
\end{align*}

\(^1\)The rank of the coefficient matrix is equal to the rank of the augmented matrix.
The equations defining the weights required for separation are:

\[ w_1 + w_2 + w_0 = d \]
\[ w_1 - w_2 + w_0 = -d \]
\[ -w_1 + w_2 + w_0 = -d \]

These can be solved and unique values of \( w_0, w_1, w_2 \) are found to be \( w_0^f = -d, w_1^f = d, w_2^f = d \). Hence the equation of the dividing line is:

\[ x_2 + x_1 - 1 = 0 \]

The two parallel hyperplanes for each class of patterns are indicated in Figure 5.

Two inseparable cases are now considered. In the case which is illustrated in Figure 6 there are four patterns:

\[
\begin{bmatrix}
\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} & = & \begin{bmatrix} 1 \\ 1 \end{bmatrix} & d_1 = d \\
\begin{bmatrix} x_{21} \\ x_{22} \end{bmatrix} & = & \begin{bmatrix} 1 \\ -1 \end{bmatrix} & d_2 = -d \\
\begin{bmatrix} x_{31} \\ x_{32} \end{bmatrix} & = & \begin{bmatrix} -1 \\ 1 \end{bmatrix} & d_3 = -d \\
\begin{bmatrix} x_{41} \\ x_{42} \end{bmatrix} & = & \begin{bmatrix} -1 \\ -1 \end{bmatrix} & d_4 = -d 
\end{bmatrix}
\]

The four equations to be solved for the weights are:

\[ x_2 + x_1 - 1 = 0 \]
\[ \text{Text content here} \]
\[ W_1 + W_2 + W_0 = d \]
\[ W_1 - W_2 + W_0 = -d \]
\[ -W_1 + W_2 + W_0 = -d \]
\[ -W_1 - W_2 + W_0 = -d \]

and they are inconsistent. A dividing line cannot be placed between them and satisfy the given conditions.

Another inseparable case is shown in Figure 7. There are four patterns:

\[ [x]_1 \]
\[ X_{11} = 1 \quad d_1 = d \]
\[ X_{12} = 1 \]

\[ [x]_2 \]
\[ X_{21} = -1 \quad d_2 = d \]
\[ X_{22} = -1 \]

\[ [x]_3 \]
\[ X_{31} = 1 \quad d_3 = -d \]
\[ X_{32} = -1 \]

\[ [x]_4 \]
\[ X_{41} = -1 \quad d_4 = -d \]
\[ X_{42} = 1 \]

The resulting four equations are inconsistent and cannot be solved for the weights. It is obvious, from Figure 7, that one line cannot separate the patterns.

It should be emphasised that the examples discussed use minimum mean square error adaption. Since this scheme requires

\[ ^2 \text{The rank of the coefficient matrix is different from the rank of the augmented matrix.} \]
convergence to a desired value of output (with the accompanying precise positioning of the hyperplane) it makes complete separation of apparently separable patterns (such as shown in Figure 6) impossible. Other "less precise" schemes, as discussed by Treado[6], do not encounter some of these difficulties.

The problem of separability will be mentioned again in Chapter III where coding is considered.

1.3 Experimental Simulation of Adaline

A computer program was written to simulate Adaline and to test its ability to separate pattern sets. Two pattern sets were applied: a group of five 'C's was to produce an output of +10 and a group of five 'T's to produce an output of -10. The patterns and resultant inputs to Adaline are shown in Figure 8. During the training phase one of the patterns was presented and an appropriate adjustment made to the weights. Each of the ten patterns was then presented and the error, $E_x$, in each case was measured. $\sum E_x^2$ was then calculated. This process was carried out 100 times. The value of the constant, $c$, in equation 1.2 was initially chosen to be 0.05.

A graph of $\sum E_x^2$ against number of adaptions is shown in Figure 9. This has been called a "learning curve" by Widrow [5]. It indicates how many presentations of the patterns are necessary before Adaline is able to recognize all of the patterns with small error. From the learning curve shown it can be seen that the quantity $\sum E_x^2$, has
Patterns mapped to +10

\[
\begin{array}{cccccc}
\text{X X X . . . X X X X . . . X X X . . . . . . . .} \\
\text{X . . . . . . X . . . X . . . X . . . X . . . X . . . X} \\
\text{X . . . . . . X . . . X . . . X . . . X . . . X . . . X} \\
\text{X X X . . . X X X . . . . . . . . X X X X X X X X} \\
\end{array}
\]

(1) (2) (3) (4) (5)

(1) 1 1 1 -1 1 -1 -1 -1 1 -1 -1 1 1 1 1 -1
(2) -1 1 1 1 -1 1 -1 -1 -1 1 1 1 1 1 1 1
(3) 1 1 1 1 1 -1 -1 1 1 -1 -1 1 -1 -1 -1 -1
(4) -1 1 1 1 -1 -1 1 -1 1 -1 -1 1 -1 1 1 1 1
(5) -1 -1 -1 -1 1 -1 -1 1 1 -1 -1 1 1 1 1 1 1

Patterns mapped to -10

\[
\begin{array}{cccccc}
\text{X X X . . . X X X . . . . . . . . . . . . . .} \\
\text{X . . . . . . X . . . X . . . X . . . X . . . X} \\
\text{X . . . . . . X . . . X . . . X . . . X . . . X} \\
\text{X X X . . . X X X . . . . . . . . X X X X X X X} \\
\end{array}
\]

(6) (7) (8) (9) (10)

(6) 1 1 1 -1 1 -1 -1 -1 1 -1 -1 -1 1 -1 -1 -1
(7) -1 -1 -1 -1 -1 1 -1 -1 -1 1 1 1 1 -1 -1 -1
(8) -1 -1 -1 -1 -1 1 1 1 1 -1 1 -1 -1 -1 -1 -1
(9) -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1
(10) -1 -1 -1 -1 1 -1 -1 1 1 -1 1 1 -1 -1 -1 -1

Figure 8
Figure 9(a)
Learning Curve for $q = 0.05$

Figure 9(b)
Part of the Learning Curve of Figure 9(a)
dropped rapidly, after 20 adaptations, from the large initial value of 1000 to a value of 47. This might seem acceptable but the use of $\sum e_i^2$ as a criterion is dangerous since it may be generated almost entirely by one or two unacceptably large errors. Training can be considered complete only when all the errors are zero or, if this requires too many adaptations, when the error for each pattern lies within an acceptable limit.

The five 'C's and five 'T's were used again as test patterns to find the effect of the constant, $g$, on the rate of adaptation. Values of $\sum e_i^2$, after various numbers of iterations, plotted against $g$, are shown in Figure 10. The "best" value of $g$, that producing the smallest $\sum e_i^2$, after a fixed number of iterations, is found to be $\approx 0.06$. If $g = \frac{k}{n}$, where $k$ is a constant and $n$ is the number of weights, the best value of $k$ is 1.02 for an Adaline with 16 weights.

Learning curves for other values of $g$ are shown in Figures 11 and 12. For small values of $g$, the adjustments are small and $\sum e_i^2$ is reduced with few fluctuations. When $g$ approaches the maximum permissable value, the adjustments are large and $\sum e_i^2$ experiences large fluctuations before a suitably low value is approached. For values of $g$ greater than 0.1167 the adjustments are too large and $\sum e_i^2$ diverges.

The problem of large analogue errors "concealed" by what seems an acceptably low $\sum e_i^2$ has been mentioned. In all cases examined, however, no individual error was ever prohibitively large. Figure 13 is a table of values of
Figure 11
Learning Curve for $g = 0.003$
analogue error corresponding to a value of $\sum \frac{e_i^2}{\delta z}$ for various values of $g$ and various numbers of iterations.

### 100 ITERATIONS

<table>
<thead>
<tr>
<th>$g$</th>
<th>$\frac{\sum e_i^2}{\delta z}$</th>
<th>ANALOGUE ERROR $e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C's +10$</td>
<td>$T's -10$</td>
</tr>
<tr>
<td>0.09</td>
<td>24.27</td>
<td>-0.13 -2.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.87 -1.85 -0.80 0.11 1.01</td>
</tr>
<tr>
<td>0.06</td>
<td>10.83</td>
<td>-1.18 -0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.07 -0.56 0.43 0.47 0.03</td>
</tr>
<tr>
<td>0.01</td>
<td>43.82</td>
<td>-2.77 1.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1.74 1.09 0.03 -2.02 -0.13</td>
</tr>
</tbody>
</table>

### 50 ITERATIONS

<table>
<thead>
<tr>
<th>$g$</th>
<th>$\frac{\sum e_i^2}{\delta z}$</th>
<th>ANALOGUE ERROR $e_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C's +10$</td>
<td>$T's -10$</td>
</tr>
<tr>
<td>0.09</td>
<td>45.34</td>
<td>-0.39 2.25 -0.86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1.92 1.41 3.97 1.53 -0.156</td>
</tr>
<tr>
<td>0.06</td>
<td>19.60</td>
<td>-1.79 -0.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.34 -0.12 0.59 0.58 0.037</td>
</tr>
<tr>
<td>0.01</td>
<td>78.21</td>
<td>-1.15 2.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-3.48 -0.09 -1.52 -3.03 -1.29</td>
</tr>
</tbody>
</table>

Figure 13

If training were continued indefinitely, the errors, $e_i$ and $\frac{\sum e_i^2}{\delta z}$, could, presumably, be brought close to zero. This is hardly practical, however, and a more realistic approach is to limit the number of iterations and accept an analogue error. Here the concept of a "dead zone" could be introduced. Thus, if the desired value of output is $+10$, a tolerance is placed on this value so that if, after training, a pattern produces an output of, say, $+10 \pm 3$ it would be classed in the $+10$ group. In addition to making an excessive number of training
iterations unnecessary this may enable Adaline to recognize patterns which it has not seen during training—such as a training pattern which has been contaminated by noise. If Adaline is used with threshold device, two modifications could be made. Consider the Adaline shown in Figure 14.

![Diagram of Adaline with threshold device](image)

**Figure 14**

The threshold device yields an output of +1 if its input is greater than +7 and an output of -1 if its input is more negative than -7. Only the lower half of the dead zone, or tolerance, is used here. An analogue input of 10 to the threshold element is aimed at but, in this case, if values of output, \( a \), are greater than 10, adaption to reduce this to 10 would seem pointless. A new training scheme would be to adjust the weights (by the old rule) to make the output approach 10 from below. If the output is greater than 10 no adaption should take place.
There are many methods of approximating a polynomial by a straight line or curve. One common method is described by texts on numerical analysis [8] and receives the name "minimum square error approximation." If the curve shown in Figure 15 is a polynomial, \( p(x) \), and is approximated by a straight line, \( l(x) = ax + b \), then the constants, \( a \) and \( b \), can be found for which \( l(x) \) satisfies the minimum mean square error criterion. i.e. The sum, \( \sum_{i=1}^{m} e_i^2 = e_1^2 + e_2^2 + \ldots + e_i^2 + \ldots + e_m^2 \), where \( e_i = p(x_i) - l(x_i) \), must be minimized. If \( F = \sum_{i=1}^{m} e_i^2 \), then the values of \( a \) and \( b \) which minimize \( F \) can be found using the techniques of differential calculus.

Consider the problem of training Adaline. Patterns, \( [x]_1 \), \( [x]_2 \), \( \ldots \), \( [x]_m \), have desired outputs of \( d_1 \), \( d_2 \), \( \ldots \), \( d_m \).
At any point during the training phase, when the output of Adaline in response to these patterns is $a_1, a_2, \ldots, a_m$, the errors are $e_1, e_2, \ldots, e_m$. Adaptation requires the adjustment of the threshold and weights in such a way as to make $a_1, a_2, \ldots, a_m$ approach $d_1, d_2, \ldots, d_m$. By carrying over the concepts of polynomial approximation the sum of the squares of these errors can be found and, by using this as a criterion, the weights can be found which minimize $\sum_{i=1}^{m} e_i^2$.

### 2.1 Analytical Minimization of $\sum_{i=1}^{m} e_i^2$

Before training Adaline is assumed to have the threshold and weights of value, $W_0$, $W_1$, $W_2$, $\ldots$, $W_i$, $\ldots$, $W_n$. If the $m$ training patterns are now presented the outputs are given by the following equations:

\[
\begin{align*}
a_1 &= X_1 W_1 + X_2 W_2 + \ldots + X_j W_j + \ldots + X_{m} W_m + W_0 \\
a_2 &= X_1 W_1 + X_2 W_2 + \ldots + X_j W_j + \ldots + X_{m} W_m + W_0 \\
a_m &= X_1 W_1 + X_2 W_2 + \ldots + X_j W_j + \ldots + X_{m} W_m + W_0 
\end{align*}
\]

The expressions for the errors are: $e_1 = d_1 - a_1$, $e_2 = d_2 - a_2$, $e_m = d_m - a_m$.

The sum, $\sum_{i=1}^{m} e_i^2$, is formed and this must be minimized by adjusting the threshold and the weights. If $\sum_{i=1}^{m} e_i^2$ is called $S$, then

\[ S = (d_1 - a_1)^2 + (d_2 - a_2)^2 + \ldots + (d_m - a_m)^2 \]

Analytically, the values of the threshold and weights which yield the minimum value of $S$ can be found by taking the
partial derivatives of $S$ with respect to the threshold and weights and equating the derivatives to zero.

i.e.

$$\frac{\partial S}{\partial w_0} = 0$$
$$\frac{\partial S}{\partial w_i} = 0$$
$$\frac{\partial S}{\partial w_n} = 0$$

There are $n+1$ equations and $n+1$ unknowns and, if these equations are consistent, they can be solved for $w_0^f, w_i^f, w_2^f, \ldots w_n^f$ -- the values of threshold and weights which produce minimum $S$. The equations are:

$$(d_1 - a_1) + (d_2 - a_2) + \ldots + (d_i - a_i) + \ldots + (d_m - a_m) = 0$$
$$(d_1 - a_1)x_{11} + (d_2 - a_2)x_{21} + \ldots + (d_i - a_i)x_{i1} + \ldots + (d_m - a_m)x_{m1} = 0$$
$$(d_1 - a_1)x_{1j} + (d_2 - a_2)x_{2j} + \ldots + (d_i - a_i)x_{ij} + \ldots + (d_m - a_m)x_{mj} = 0$$
$$(d_1 - a_1)x_{1n} + (d_2 - a_2)x_{2n} + \ldots + (d_i - a_i)x_{in} + \ldots + (d_m - a_m)x_{mn} = 0$$

They can be expressed compactly, with the $j$th equation given by:

$$\frac{\partial S}{\partial w_j} = 0 \quad \text{or} \quad \sum_{i=1}^{m} e_i x_{ij} = 0 \quad \text{valid for } j = 1, n$$

and

$$\frac{\partial S}{\partial w_0} = 0 \quad \text{or} \quad \sum_{i=1}^{m} e_i = 0 \quad \text{for } j = 0$$

It is instructive to consider a few examples of this
technique as applied to patterns in two space. For such patterns there will be three equations, with three unknowns, to be used to find the best weights. i.e.

\[ \sum_{i=1}^{N} e_i = 0 \]
\[ \sum_{i=1}^{N} e_i x_{i1} = 0 \]
\[ \sum_{i=1}^{N} e_i x_{i2} = 0 \]

In the example illustrated in Figure 16 the patterns are:

\[ [X]_1 \]
\[ x_{11} = 1 \]
\[ x_{12} = -1 \]
\[ d_1 = d \]

\[ [X]_2 \]
\[ x_{21} = -1 \]
\[ x_{22} = 1 \]
\[ d_2 = -d \]

The errors are given by:

\[ e_1 = d - w_1 + w_2 - w_0 \]
\[ e_2 = -d + w_1 - w_2 - w_0 \]

and the three equations are:

\[ w_0 = 0 \]
\[ w_1 - w_2 = d \]
\[ w_1 - w_2 = d \]

These equations yield an infinite number of solutions for the weights so that the weights define an infinite number of hyperplanes which pass through the point \( X_1 = 0, X_2 = 0 \). If \( W^f_2 \) is
chosen to be \( d \), then \( W_i^f = 2d \) and the hyperplane has equation, \( X_2 = -2X_1 \). This is the same result as was obtained in example 1 in section 1.2.

In the example shown in Figure 17 the patterns are:

\[
\begin{align*}
[X]_1 & \quad X_{11} = 1 \quad d_1 = d \\
X_{12} & = 1
\end{align*}
\]

\[
\begin{align*}
[X]_2 & \quad X_{21} = 1 \quad d_2 = -d \\
X_{22} & = -1
\end{align*}
\]

\[
\begin{align*}
[X]_3 & \quad X_{31} = -1 \quad d_3 = -d \\
X_{32} & = 1
\end{align*}
\]

The errors are given by:

\[
\begin{align*}
e_1 & = d - W_1 - W_2 - W_0 \\
e_2 & = -d - W_1 + W_2 - W_0 \\
e_3 & = -d + W_1 - W_2 - W_0
\end{align*}
\]

and the equations to be solved for the weights and threshold are:

\[
\begin{align*}
W_1 + W_2 + 3W_0 & = -d \\
3W_1 - W_2 + W_0 & = d \\
W_1 - 3W_2 + W_0 & = -d
\end{align*}
\]

Solution yields: \( W_0^f = -d \), \( W_i^f = d \), \( W_2^f = d \) and hence, the equation of the dividing hyperplane is \( X_2 = -X_1 \).

Again this agrees with the results of example 2 in 1.2.
In the example shown in Figure 18 the patterns are:

$$[X]_1 \quad x_{11} = 1 \quad d_1 = d \quad \times \quad X$$

$$x_{12} = 1$$

$$[X]_2 \quad x_{21} = 1 \quad d_2 = -d \quad \times \quad X$$

$$x_{22} = -1$$

$$[X]_3 \quad x_{31} = 1 \quad d_3 = -d \quad \times \quad X$$

$$x_{32} = 1$$

$$[X]_4 \quad x_{41} = -1 \quad d_4 = -d \quad \times \quad X$$

$$x_{42} = -1$$

Figure 18

The equations for the errors are:

$$e_1 = d - w_1 - w_2 - w_0$$

$$e_2 = -d - w_1 + w_2 - w_0$$

$$e_3 = d + w_1 - w_2 - w_0$$

$$e_4 = -d + w_1 + w_2 - w_0$$

and the three equations to be solved for the weights are:

$$4 w_0 + 2d = 0$$

$$4 w_1 - 2d = 0$$

and so $$w_0^f = -d/2$$, $$w_1^f = d/2$$, $$w_2^f = d/2$$. The equation of the hyperplane which separates the two classes of patterns is $$x_2 = -x_1 + 1$$. This apparent contradiction of the result of example 3 of section 1.2 can be explained in the following way. Minimizing $$S$$ analytically results in finding
the particular hyperplane which will separate the patterns into two classes in such a way as to minimize \( \sum_{i=1}^{m} e_i^2 \). The equations of section 1.2 describe an adaptive scheme which will separate patterns with a precisely located hyperplane only if the error, \( e_i \), for each pattern and \( \sum_{i=1}^{m} e_i^2 \) can be reduced to zero.

Finally, in the example shown in Figure 19 the patterns are:

\[
\begin{align*}
\mathbf{X}_1, \quad & X_{11} = 1, \quad d_1 = d \\
& X_{12} = 1 \\
\mathbf{X}_2, \quad & X_{21} = -1, \quad d_2 = d \\
& X_{22} = -1 \\
\mathbf{X}_3, \quad & X_{31} = -1, \quad d_3 = -d \\
& X_{32} = 1 \\
\mathbf{X}_4, \quad & X_{41} = 1, \quad d_4 = -d \\
& X_{42} = -1
\end{align*}
\]

Figure 19

The three equation for the weights yield \( w_1^f = w_2^f = w_0^f = 0 \). This means, in effect, that no hyperplane exists which will separate the patterns. This agrees with the result of example 4 in section 1.2.

2.2 Steepest Descent Minimization of \( \sum_{i=1}^{m} e_i^2 \)

If the function, \( S = \sum_{i=1}^{m} e_i^2 \), has a minimum it can be found by an iterative procedure. One possible method is described here.
If the threshold and weights have initial values, \( W_0, W_1, W_2, \ldots, W_n \), and these weights do not define the minimum, then all the patterns, \([X_1], [X_2], \ldots, [X_i], \ldots, [X_m] \), are presented. The function \( S \), and the \( n+1 \) partial derivatives are calculated. The method of steepest descent involves evaluating the magnitude of the gradient vector at the point defined by \( W_0, W_1, W_2, \ldots, W_j, \ldots, W_n \) and descending a short distance along the gradient vector towards the minimum of

by adjusting the weights, \( W_0, W_1, \ldots, W_n \). The adjustment made to the \( j \)th weight is proportional to \( \frac{\partial S}{\partial W_j} \) and is \( \Delta W_j = g \sum_{i=1}^{m} e_i X_{ij} \).

The adjustment to the threshold is \( \Delta W_0 = g \sum_{i=1}^{m} e_i \). \( g \) is a constant of proportionality and affects the size of the increments to all the weights and the threshold. It should be noted that the increments, \( \Delta W_0, \Delta W_1, \ldots, \Delta W_j, \ldots, \Delta W_n \), are not necessarily equal. At each new set of values, \( W_0, W_1, \ldots, W_j, \ldots, W_n \), the patterns are again presented and the process is repeated. This process is continued until the minimum is found. A memory is required to implement this process.

After presenting each pattern the error, \( e_i \), and the \( n \) elements of the \( i \)th pattern must be stored until all the patterns are presented so that \( S, \Delta W_0 \) and \( \Delta W_j \) can be calculated.

The calculation of \( \Delta W_0 \), \( \Delta W_j \) could be accomplished using \( n+1 \) totaling registers to collect \( \sum_{i=1}^{m} e_i, \sum_{i=1}^{m} e_i X_{i1}, \ldots, \sum_{i=1}^{m} e_i X_{in} \).

It is for this reason that a modified form of steepest descent (which does not require a memory) is used when training Adaline to find the values of weights which yield a minimum for \( S \).
2.3 Modified Steepest Descent

The sum of the squares of the errors, $S$, is given by:

$$S = \sum_{i} e_i^2$$

The adaptive scheme described in this section attempts to find the weights which yield the minimum for $S$ without the need for a memory. The first pattern is presented and $e_i^2$ is calculated. This is treated as the function to be minimized and the steepest descent method is applied to this. Calling $f_i = e_i^2 = (d_i - a_i)^2$, the derivatives are:

$$\frac{\partial f}{\partial W_0} = 2(d_i - a_i)$$

$$\frac{\partial f}{\partial W_i} = 2(d_i - a_i)X_i$$

$$\frac{\partial f}{\partial W_n} = 2(d_i - a_i)X_{in}$$

It should be noted that the derivatives are equal in magnitude.

As was mentioned in section 2.2, the method of steepest descent involves making an adjustment, which is proportional to $\frac{\partial f}{\partial W_i}$, to each of the weights. i.e. $\Delta W_i = g e_i X_{in}$, and the adjustment to the threshold is $\Delta W_0 = g e_1$, where $g$ is a constant.

The adjustments, $\Delta W_0, \Delta W_1, \ldots, \Delta W_n$, are equal in magnitude. These adjustments are made and then the second pattern is applied and $f_2 = e_2^2$ is calculated. Equal adjustments are then made by applying one step of the steepest descent procedure to $f_2$. This process is repeated until the $n$ patterns have all been been presented. If the minimum has not been found presentation of the patterns and adjustment continues as described. The "real" steepest descent method uses the function, $S$. 

28.
as the criterion. The criterion for the modified method, however, changes from \( f_1 \) to \( f_2 \) --- to \( f_n \) --- to \( f_m \). If the constant, \( g \), is such that the adjustments are very small then \( S \approx f_1 + f_2 + \cdots + f_m \), where \( f_i = e_i^2 \) and is evaluated after \( i - 1 \) adjustments. The minimum of \( S \) can be found by continuing to apply this method. In many cases the true minimum (as obtained by the analytical method of examples 1 and 2 of section 2.1) is found but in some cases (as in example 3 of section 2.1) no hyperplane defining the minimum is found. The hyperplane oscillates around a mean plane which would define the minimum of \( S \).

The adaptation scheme can be defined as follows:

Present the patterns in turn and, after each pattern is presented, adjust all of the weights and the threshold by an equal amount in a direction such that the error will be reduced. Adjustment is to take place of the error, on application of a pattern, is not zero.

This scheme is most frequently called "Minimum Square Error Adaption" in the literature. The main difference between "Minimum Square Error Adaption" and the method of steepest descent is that in the former adjustment is made after each pattern is presented whereas in the latter, all the patterns are presented before any adjustment can be made.

3 In "real" steepest descent \( S \) would be given by \( S = f_1' + f_2' + \cdots + f_m' \) where \( f_1', f_2', \ldots, f_m' \) were evaluated before any adjustment was made.
3.1 The Equations of the Plant

In this chapter Adaline is taught to recognize the behaviour of a stable plant and to produce the correct control signal. The plant chosen is a second order system which consists of a motor and a load. The power supplied to the motor is controlled by a relay. The system is stable by virtue of unity and velocity feedback. It is shown in Figure 19.

The differential equations of the plant can be obtained, as a function of time, from the block diagram. Hence the output, \( c(t) \), and the output rate, \( \dot{c}(t) \), can be calculated, as functions of time, for various initial values of \( c(t) \) and \( \dot{c}(t) \).--rather than supplying the plant with various desired outputs, \( r(t) \). The voltage which is applied to the plant is \( \pm V \) volt, depending on the sign of the relay input, \( d(t) \).

From Figure 19,

\[
d(t) = r(t) - c(t) - K_t \dot{c}(t)
\]  

3.1
and choosing to set \( r(t) = 0 \), the switching condition for the voltage, \( V \), is given by:
\[
\dot{c}(t) = - \frac{1}{K_t} c(t)
\]
The equations for output and output rate are:
\[
c(t) = Kt + (C_0 + C_i - K) + (K - C_i) e^{-bt}
\]
\[
\dot{c}(t) = K - b(K - C_i) e^{-bt}
\]
where \( C_0 = C(0) \) the initial value of \( c(t) \) at \( t = 0 \),
\( C_i = \dot{c}(0) \) the initial value of \( \dot{c}(t) \) at \( t = 0 \),
and \( K = \pm K_t \sqrt{V} \).
A convenient way of presenting this information is to use the phase plane. If the system error, \( e(t) \), is defined as
\[
e(t) = r(t) - c(t)
\]
where \( r(t) \) is the desired output and \( c(t) \) the actual output, then a plot of error, \( e(t) \), versus error rate, \( \dot{e}(t) \), is called a trajectory in the phase plane. Since "inputs" are applied to the system as initial conditions of \( c(t), \dot{c}(t) \) with \( r(t) = 0 \) then \( e(t) = -c(t) \), \( \dot{e}(t) = -\dot{c}(t) \). Hence equations 3.3 and 3.4 define a trajectory in the phase plane which describes the system behaviour for a given set of initial conditions. Since \( K \) can be positive or negative, depending on the sign of the input to the relay, \( d(t) \), the trajectory obeys two equations, A and B. This is indicated in Figure 20. The switching condition has been stated in the equation 3.2. This is a line with a slope, \(-1/K_t\), in the phase plane. The values chosen for the control system were: \( K_t = 0.2 \), \( K = \pm 10 \), and \( b = 1 \).
Using the analytical solutions for $C(t)$ and $\dot{C}(t)$ a computer program was constructed to produce sample trajectories in the phase plane for comparison with the attempts produced by Adaline. One of them is shown in Figure 21.

The objective of training Adaline is to enable it to recognize combinations of $e(t)$ and $\dot{e}(t)$ in order that it may produce a response, $a(t)$, which is close to the correct input to the relay, $d(t)$. During training it is desirable to present many combinations of $e(t)$ and $\dot{e}(t)$, so that when Adaline later acts as a controller it will have enough "experience" (or have seen most of the phase plane) to make a correct decision and produce a good value of $a(t)$ to drive the relay. To train Adaline in a practical situation various initial conditions, $C(0)$ and $\dot{C}(0)$, would be set and the resulting trajectories would generate values
of $e(t)$, $\dot{e}(t)$ and $d(t)$, which would be presented to Adaline. This is illustrated in Figure 22. It is unnecessary to follow this procedure exactly in a digital computer simulation. Since it is known that the input to relay, $d(t)$, is given by $d(t) = e(t) + k_t \dot{e}(t)$ everywhere in the phase plane, Adaline is, in this study, presented with a large number of randomly selected points with coordinates, $e(t)$ and $\dot{e}(t)$, and is trained to produce an analogue output, $a(t)$, of value as close as possible to $d(t)$ for these points.

3.2 Coding of the Phase Plane and the Training of Adaline

A practical Adaline is limited in size by having a finite number of weights and the number of weights limits the number of patterns which Adaline can attempt to separate. When the control system responds to a set of initial conditions,
the error, $e(t)$, and error rate, $\dot{e}(t)$, vary continuously until a steady state condition is reached. If the values of $e(t)$, $\dot{e}(t)$ and $d(t)$ are sampled frequently, patterns could be assigned to the points with coordinates, $e(t)$ and $\dot{e}(t)$, in the phase plane. This would yield a very large number of patterns and an Adaline with a large number of weights would be required. Although, in this study, the values of $e(t)$ and $\dot{e}(t)$ were sampled frequently, the above situation was avoided by dividing the phase plane into a small number of regions to which certain patterns were assigned. This ensures that only a small number of patterns will be presented to Adaline. It was also necessary to choose the patterns, or codes, in such a way that it will be possible for Adaline to separate groups of the patterns into the appropriate two classes. If this last requirement is met, it is said that the patterns are linearly separable.

A linearly separable code which has been used by Widrow [9] is indicated in Figure 23. The variable, $X$, which can take

\[ a \leq X < b \]
\[ b \leq X < c \]
\[ c \leq X < d \]
\[ d \leq X < f \]

is represented by the following code:

<table>
<thead>
<tr>
<th>$X$</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \leq X &lt; b$</td>
<td>0001</td>
</tr>
<tr>
<td>$b \leq X &lt; c$</td>
<td>0010</td>
</tr>
<tr>
<td>$c \leq X &lt; d$</td>
<td>0100</td>
</tr>
<tr>
<td>$d \leq X &lt; f$</td>
<td>1000</td>
</tr>
</tbody>
</table>

Figure 23
on an infinite number of values between \( a \) and \( f \), is coded by choosing a reference or origin at \( a \) on the \( X \) axis and dividing the region to the right of this reference into segments. The total number of segments determines the number of digits in the code. The code assigned to each region is obtained by moving the digit 1 one space to the left as the variable, \( X \), moves from one segment to another. This ensures that there is a difference of two digits in the code between any two regions--which may be a factor in explaining the separability of the code (Treado[6]).

The phase plane can now be divided into squares, for example, and a pattern assigned to each square. One method of coding \( e(t) \) and \( \dot{e}(t) \) separately has been discussed. If this were done, a group of points in a square in the phase plane could have a coded value for \( e(t) \), and a coded value for \( \dot{e}(t) \). A pattern which could be associated with points lying within a given square in the phase plane can be obtained by placing the coded values for \( e(t) \) and \( \dot{e}(t) \) side by side. This means that if \( e(t) \) has a value such that it is in segment 0001, and \( \dot{e}(t) \) has a value such that it is in segment 0100, the pattern for the point with coordinates, \( e(t), \dot{e}(t) \) in the given square is \([e \dot{e}]\) or 00010100.

In all the tests described here the possible initial conditions for the control system were limited so that the maximum absolute value of \( e(t) \) and \( \dot{e}(t) \) was 10. This section of the phase plane was then divided into squares. By choosing a reference at \( e(t) = -10 \), \( \dot{e}(t) = -10 \), \( e(t) \) and \( \dot{e}(t) \) were
suitably coded and hence a pattern was assigned to each square. A division of the phase plane into 25 squares is shown in Figure 24. The fineness with which the phase plane is divided determines the number of digits per pattern and this determines the number of weights in the Adaline. If the $e(t)$ and $\dot{e}(t)$ axes are divided into $n$ sections each, then the pattern for each square has $2n$ digits. It can be seen that the size of Adaline increases linearly with the fineness of the division. The more alarming feature is that the number of patterns increases with the square of the sections, $n^2$, and the question arises as to whether Adaline is capable of separating $n^2$ patterns into distinct classes when the number of weights is $2n$.

By using various sizes of grids on the $e(t), \dot{e}(t)$ plane, several Adalines were trained (according to the Mean Square Error scheme of Chapter I) to "duplicate" the performance of the control system of section 3.1. As in Chapter I, each of the zeros in the patterns was replaced by the digit $-1$. 

37.
A pseudo-random number generator was used to produce values of \( e(t) \) and \( \dot{e}(t) \) which were then coded. The function, \( d(t) \), was calculated from equation 3.1 and, together with the coded values of \( e(t) \) and \( \dot{e}(t) \), presented to Adaline. After each pattern was presented the weights were adjusted and the next pattern read in. After a suitable training period the weights were fixed and Adaline was tested with one pattern from each square of the grid.

The output of the trained Adaline, \( a(t) \), is positive or negative—depending on which pattern is presented. If a line is drawn in the phase plane separating squares for which \( a(t) \) is positive from those with negative \( a(t) \), this can be called the switching line which Adaline has produced, and the training of Adaline can be considered as the training of a function generator.

Several points were examined using the scheme which has been outlined:

1. The effect of the adjustment constant, \( g \), on the switching line which is obtained when the training process is complete.
2. The effect of the number of training cycles on the final switching line.
3. Convergence of values of weights and threshold as training proceeds.
4. The capacity of Adaline.

These four points were examined using a digital computer simulation of the coding scheme and Adaline as described.
in the previous paragraph. A training cycle consists of fixing the adjustment constant, \( q \), and presenting Adaline with patterns generated by a fixed number of points in the phase plane. The training cycle is repeated for various numbers of points and various \( q \).

Assuming that Adaline is trained with a very large number of randomly distributed points in the phase plane the switching line which Adaline will produce, when trained, can be predicted.

For a grid of 16 squares, and hence an Adaline of eight weights and a threshold, Adaline's attempt at duplicating a switching line of slope, -5, is shown in Figure 25. For points occurring in all the squares, except those numbered 2, 6, 11 and 15, the input to the relay, \( d(t) \), is either positive for all points in a square or negative for all points in the square. There is no difficulty in training Adaline to differentiate between such squares. Points in squares 2, 6, 11 and 15 can,
however, present Adaline with conflicting information during the training phase. In square 2, for example, all points, when coded, yield the same pattern, 00101000. Since the switching line passes through this square the points, and hence the pattern, can correspond to both positive and negative values of \( d(t) \). Adaline has, therefore, the problem of placing the same pattern into two different classes. Using the assumption of a statistically uniform distribution of points it can be seen that Adaline will see more values of negative \( d(t) \) than positive \( d(t) \) during training.

After training, therefore, the weights and threshold will have values such that the pattern of square 2 is classed negative. The same argument applies to squares 6, 11 and 15. It is on this basis that a prediction of Adaline's attempt at producing a switching line is made.

For a grid of 25 squares, and an Adaline with ten weights and a threshold, the prediction of the switching line produced, after training, is shown in Figure 26.
In this case the weights and threshold, after training, should be such that squares 3 and 8 will yield positive output and squares 18 and 23 negative output. The output of Adaline in response to a pattern from square 13 is uncertain. Of the patterns presented to Adaline from square 13, statistically, half of them would correspond to positive values of $d(t)$ and half of them to negative values of $d(t)$.

For a grid of 36 squares and an Adaline of 12 weights and one threshold, the prediction of the switching line produced by Adaline, after training, is shown in Figure 27.

![Figure 27](image)

From previous arguments it can be seen that, for squares 9 and 15, Adaline will produce a negative output after training and, for squares 22 and 28, Adaline will produce a positive output. Since the actual switching line bisects squares 3 and 34, the output Adaline will produce, after training, when presented with the patterns of squares 3 and 34 is uncertain.
Examples of the switching lines actually produced by Adaline, after training, are shown in Figure 28.

Variations of these switching lines were produced by using different values of the adjustment constant, $g$, and with
different numbers of training cycles. The four points mentioned earlier in the section are now discussed.

The size of adjustment can be considered as affecting the position of the switching line in the following way. For values of \( g \) close to or less than the optimum value (as discussed in section 1.3) the switching line produced, after training on a large number of points, is very close to the predicted line. Variations occurred when the values of \( g \) were greater than the optimum value. When the adjustment is large, the last few patterns presented before training is stopped will have a disproportionately large effect on the weights. The last few patterns may not be distributed evenly among the squares and hence Adaline may be biased and produce a peculiar switching line. Two examples are shown in Figure 29. When the size of adjustment is small this effect is less.

\[ e^* = \frac{e}{\nu} \]

FIGURE 29
The effect on the switching line of the number of points presented during training is similar to that due to different values of constant, $g$, but is not so pronounced. The squares which are affected most are those for which, during training, there are both positive and negative values of $d(t)$. The total number of points presented determines the number which occur in a given square. Although the random number generator produces a statistically random spread of points, it may be cut off at a stage when there has been a distinct bias toward one region of the phase plane. The number of points presented during testing varied between 100 and 1000. In the case of a grid of 16 squares the switching line is unaffected. In the grid of 25 squares, square 13 of Figure 26 yields outputs both negative and positive—depending on the number of training points. In the case of a grid of 36 squares the outputs produced by squares 3 and 34 oscillate between positive and negative values.

When examined during training the weights and threshold are seen to oscillate about a mean value. Given a value of less than or equal to the optimum value, the weights and threshold oscillate around values which yield a switching line close to the predicted switching line. The oscillation is a result of the presentation of conflicting information from squares cut by the switching line of the plant.

Widrow [9] has stated that "the statistical capacity of Adaline is twice the number of weights." By this 'rule' the Adaline with eight weights would be able to classify the
16 patterns which it receives but the 10- and 12-weight Adalines would have difficulty in attempting to classify the 25 and 36 patterns which they receive. The results obtained indicate that this does not appear to be the case. In an attempt to discover if there is a limit to the number of patterns which Adaline can separate, Adalines with 14, 16, 18, 20, 22 and 24 weights were trained. Several thousand points were presented to each Adaline during training and, in all cases, the switching line produced after training was very close to that predicted. Variations occurred only when squares were cut by the switching line of the control system. Since an upper limit did not appear (an Adaline with 24 weights separated 144 patterns into two classes) further study of Adaline's capacity is indicated.

3.3 Adaline Acting as the Controller

A digital computer simulated Adaline was now placed in control of the simulated plant described in section 3.1. The desired output, \( r(t) \), was set to zero, and 'inputs' were applied by setting initial conditions, \( C(0) \) and \( \dot{C}(0) \), at the output shaft. Trajectories in the phase plane were obtained for various initial conditions.

A block diagram showing Adaline connected to the plant is shown in Figure 30. The weights and threshold chosen for these tests were those which gave switching lines as close as possible to the predicted lines of Figures 25, 26, 27.
The essence of computer calculations is as follows. The initial conditions \( c(0), \dot{c}(0) \) are presented to the error, error-rate calculator and yield \( e(0), \dot{e}(0) \). The values \( e(0), \dot{e}(0) \) are presented to the coder and time, \( t \), is set to zero. A pattern, which depends on the square in which the point with coordinates, \( e(t), \dot{e}(t) \), lies, is generated and applied to Adaline. Adaline produces an output, \( a \), and this causes the output from the relay to assume a definite sign. Using equations 3.3 and 3.4, and a small increment of time (10ms), \( c(t), \dot{c}(t) \) are calculated. These values define the second point of the phase trajectory. \( c(t) \) and \( \dot{c}(t) \) are now applied to the error, error-rate calculator and the calculations are repeated. The values of \( e(t) \) and \( \dot{e}(t) \) are stored for printing and for drawing a graph of \( \dot{e}(t) \) against \( e(t) \) -- the phase trajectory. The input to the relay, \( a(t) \), is examined after each increment of time. If it has changed sign, the trajectory has crossed the switching line and the sign of \( \dot{e} \) in equations 3.3 and 3.4 is then reversed.

Time is set to zero and the initial conditions, \( c(0), \dot{c}(0) \), assume the values of \( c(t) \) and \( \dot{c}(t) \) that existed immediately before switching occurred. Since the increments of time are small, these values of \( c(t) \) and \( \dot{c}(t) \) are close to the values at the time of switching. The calculations described are continued until it is clear that the trajectory is either converging to zero error or to a steady state value of error.

In one test, an Adaline with eight weights and a threshold was used to control the plant. The values of the weights and
threshold were those which yielded the switching line of Figure 28(a) and are given in Figure 31. This is the switching line of a relay operated plant, with no velocity feedback. The system in Figure 30 should exhibit the same response. The trajectory in response to initial conditions, \( c(0) = -8 \), \( \dot{c}(0) = -2 \) is shown in Figure 31. The response is highly oscillatory, but stable, and the error is, eventually, reduced to zero. A similar response would be obtained for other initial conditions.

In considering an Adaline with ten weights and a threshold, corresponding to the switching line of Figure 28(b), it can be seen that the two vertical portions of the switching line will result in the relay switching early in the course of a trajectory, so that, after switching, the trajectory will tend towards the origin. The horizontal portion of the switching line, however, causes trouble. Consider a trajectory which crosses the switching line. If it is trajectory B of Figure 20, then the relay will change sign and the trajectory becomes trajectory A of Figure 20. Trajectory A then crosses the switching line and switching again occurs. This chattering will continue until the trajectory crosses the vertical line.

This is best illustrated by Figure 32 for which the initial conditions are \( c(0) = 8 \), \( \dot{c}(0) = 2 \). The final solution is damped but settles to a steady state error of 2. Tests were then run with many values of initial conditions (see also Figure 33) but the main features of the trajectories were always as described. It should be noted that all trajectories,
The explanation of the situation and the circumstances surrounding the events that occurred. It highlights the context and the reasons behind the actions taken. This section aims to provide a comprehensive understanding of the events, their implications, and the lessons learned from them. It serves as a foundation for any future decisions or actions that need to be taken in similar situations. The detailed analysis and insights shared in this section are crucial for anyone seeking to grasp the complexities of the situation and make informed decisions.
except one, will result in chattering and a final steady state error of 2. The exception is illustrated in Figure 34 where the initial conditions are \( c(0) = -8 \), \( \dot{c}(0) = -2 \). It can be seen that switching occurs early, and, after switching, the trajectory tends towards the origin. The trajectory continues past the \( e(t) \) axis, meets the horizontal switching line, chatters and, finally, settles to a steady state error of 2. There will, however, be a trajectory, close to that described, which would arrive at the origin after the relay has switched once.

Tests were also made using an Adaline with 12 weights and a threshold, corresponding to the switching line of Figure 28(c). As in the example with 10 weights and a threshold, the two vertical portions of the switching line will cause early switching and the two horizontal portions are so placed that the relay will chatter until \( c(t) \) and \( \dot{c}(t) \) reach values such that the trajectory crosses the central, vertical line. The trajectory will, in all cases, settle to a steady value of zero error and error rate. This is illustrated in Figure 35--for which the initial conditions are \( c(0) = 8 \), \( \dot{c}(0) = 2 \). It can be seen that certain values of initial conditions would result in a trajectory switching only on the central portion of the switching line. This is illustrated in Figure 36--for which the initial conditions are \( c(0) = 2 \), \( \dot{c}(0) = -5 \).
Finer division of the phase plane would result in more accurate reproduction of the desired switching line of the controlled plant, but Adaline's attempt would still consist of vertical and horizontal portions. The examples given show the type of response which can be expected.
ADALINE IN CONTROL OF THE PLANT.

FIGURE 30.
Figure 32
Figure 35
Figure 36

Switching Line
CHAPTER IV
CONCLUSIONS

The work described in Chapter III has shown that it is possible for an Adaline to learn to control a simple plant. Its ability as a controller is poor, however, and in this section methods of improving the basic scheme are considered. The concept of training a network to perform a desired function is also discussed.

4.1 Adaline and a Dead Zone

In all cases discussed in section 3.3 the combination of Adaline and the plant resulted in a stable control system, but Adaline's control ability was poor in regions near the origin of the phase plane—where either lightly damped oscillations about the origin or lightly damped oscillations about a steady state error. One method of eliminating this would be to use Adaline in conjunction with a relay with a small dead zone. When the trajectory enters the central square, or squares, Adaline would be disconnected and, with the relay in the dead zone, the output shaft of the plant would coast. This method would probably yield a better response than can be obtained using Adaline with an ideal relay. At worst a limit cycle condition could persist but, if the central square were small, this could be tolerated. In the physical realization of this scheme, a source of difficulty would lie in deciding when the trajectory leaves and enters the central square so that Adaline could be
connected or disconnected.

4.2 Coarse and Fine Division of the Phase Plane

Another method of improving Adaline's response near the origin could consist of dividing the central square, or squares, into the same number of squares as the major grid. The pattern codes of squares in the large grid could then also be assigned to corresponding squares in a fine grid. This is illustrated in Figure 37 for a division of the phase plane into 25 squares. If the weights and threshold controlled.

![Figure 37](image)

Squares 7 and h have code 0001001000

Figure 37

correspond to the switching line of Figure 28(b), by referring to section 3.3, it can be seen that most trajectories will now finish with a steady state error of 2/5, instead of 2, since the grid is now five times finer near the origin. This scheme can be realized by altering
the coding box in Figure 30. When the trajectory enters
the central square, the values of $e(t)$ and $\dot{e}(t)$ are
amplified by 5 before being applied to the coder. This is
indicated in Figure 38. As in the scheme of section 4.1,
difficulties would arise in deciding when the trajectory
enters and leaves the central square. It should also be
noted that this method is applicable only when Adaline is
trained on a linear switching line.

4.3 Polar Coding Scheme

In Chapter III the controlled plant had a switching
line which Adaline found difficult to reproduce due to the
coarseness of the division of the phase plane. One possible
approach to the division problem, in the case of this specific
plant, would be to divide the phase plane into regions bounded
by concentric circles, with their centre at the origin and
their radii extending from the origin. The resulting 'curved'
rectangles are those to which patterns can be assigned
using a coding scheme such as that discussed in section 3.2.
This is best illustrated by considering the example with six angular divisions of \( \pi / 3 \) radians and six radial divisions in Figure 39.

If a point with coordinates, \( e(t), \dot{e}(t) \), in the phase plane is to be coded and assigned a pattern, it is first necessary to convert the coordinates of the point into polar coordinates, \( r, \phi \). This complication is trivial as far as a computer simulation is concerned, but if hardware implementation is to be considered, the additional complexity would make this method of dividing the phase plane less attractive than rectilinear division.

Using a digital computer simulation, and Adaline with 12 weights and a threshold was trained on the controlled plant of section 3.1. The program was very similar to that described in section 3.2. A pseudo-random number generator produced the coordinates, \( e(t), \dot{e}(t) \), of a point in the phase plane. They were then converted to polar coordinates, and presented to the coder. The resulting pattern and \( d(t) \), as calculated from equation 3.1, were presented to Adaline and adjustments were then made to the weights. The calculation was then repeated for different points. After a few training cycles the weights were fixed and the response of Adaline, when presented with a pattern from each square, examined. The switching line produced, after training with 700 points and an adjustment constant of 0.03, is shown in Figure 40.
Area 21 has code \([r\phi] = 000100 001000\).

\[r = 2 \implies \phi = \text{normalized values of } e(x), \text{ and } \dot{e}(x) \text{ so that the distance between concentric circles is 1.}\]

<table>
<thead>
<tr>
<th>Radius (r)</th>
<th>Code</th>
<th>Angle (\phi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0 \leq r &lt; 1)</td>
<td>000001</td>
<td>(0 \leq \phi &lt; \pi/3)</td>
</tr>
<tr>
<td>(1 \leq r &lt; 2)</td>
<td>000010</td>
<td>(\pi/3 \leq \phi &lt; 2\pi/3)</td>
</tr>
<tr>
<td>(2 \leq r &lt; 3)</td>
<td>000100</td>
<td>(2\pi/3 \leq \phi &lt; \pi)</td>
</tr>
<tr>
<td>(3 \leq r &lt; 4)</td>
<td>001000</td>
<td>(\pi \leq \phi &lt; 4\pi/3)</td>
</tr>
<tr>
<td>(4 \leq r &lt; 5)</td>
<td>010000</td>
<td>(4\pi/3 \leq \phi &lt; 5\pi/3)</td>
</tr>
<tr>
<td>(5 \leq r)</td>
<td>100000</td>
<td>(5\pi/3 \leq \phi &lt; 2\pi)</td>
</tr>
</tbody>
</table>

Figure 39.
As has been discussed in section 3.2, the distribution of points affects the training—particularly the response from patterns corresponding to squares or areas cut by the switching line of the plant. The switching line produced, after training with 300 points and an adjustment constant of 0.03, again illustrates this point (Figure 41).
It can be seen that an Adaline adequately trained in this manner would be able to control the plant. The results of a computer program, using the weights and threshold which correspond to the switching line of Figure 40 verified this. The phase trajectory was identical to that of a second order plant, compensated with unity and velocity feedback, and with the tachometer constant determined by the slope of the switching line in Figure 40.

4.4 General Remarks

The main purpose of this study has been to verify that a pattern recognition device, if suitably trained, can take the place of the controller of a plant. Some of the possibilities of control using Adaline, and some of the associated problems have been revealed in the course of this work.

It is instructive to consider the peculiar nature of this particular recognition problem and to this end comparison is made with a simple form of the weather forecasting scheme discussed by Hu [10]. Weather maps, containing information on barometric pressure over a wide area are the source of the patterns to be presented to Adaline. The weather (wet or dry) on the following day at location B is the "desired output." The map is divided into squares and the squares generate +1 or -1 pattern elements according as the pressure is higher or lower than normal. The pattern, consisting of the array of pattern elements and the resulting weather at B are presented to Adaline. If wet is assigned to -10 and dry to +10, Adaline
is trained to produce the appropriate output. Training consists of presenting Adaline with many weather maps, and the resulting conditions at B. After training is complete, Adaline can, given a weather map, estimate the weather on the following day. In this recognition problem, given a large amount of information (over the area of the map), the response at a particular place can be evaluated. In the control system, the recognition problem is the following: given information on the error and error rate of the plant at an instant (the coordinates of a point in the phase plane) what is the input to the plant to force the error and the error rate to tend to zero? It is not possible in this case to examine the overall situation to assist in making the correct decision. To date this problem has been approached by assigning patterns to regions of the phase plane, deciding into which region a point falls, and making a decision on receipt of this pattern. It can be seen that a very small amount of information is presented to Adaline at each stage. A possible area of study would be to search for a better way of presenting the information about the point, and possibly about the desired final point, to Adaline at each stage. For the control problem studied the specific difficulties encountered are now discussed.

As has been mentioned, Adaline found it difficult to reproduce the linear switching line of the controlled plant because of the coarseness of division of the phase plane. It would, however, require a very fine division for Adaline
to be able to reproduce a line accurately. The resulting size of the Adaline would make practical realization, using memistors [11] as the adaptive elements, prohibitively expensive even where only two states are coded. The polar coding scheme offers a partial solution to the problem of reproducing this particular switching line. There may also be cases in which the controlled plant has an unknown and complex switching line, which would be better tackled by a polar division than by rectilinear division. The problem of the coarseness of the division and the resulting undesirable features of the trajectory poses a difficult problem for which a solution is not immediately obvious.

Directly associated with the problem of the coarseness of the divisions is the rapid increase of patterns, and the number of digits per pattern, with increasing fineness of the quantization. It was shown, experimentally, that 144 patterns generated in the quantized phase plane, could be successfully separated using 24 weights and a threshold. The statement on the capacity of Adaline by Widrow [9] and discussed in section 3.2 deals, presumably, with all possible patterns which can be generated by permutations of the digits applied to Adaline. In presenting Adaline with 144 patterns from the phase plane, only a small and carefully controlled sample of the total number of patterns which can be obtained from the permutations of the 24 digits is used.

In this study Adaline was trained on a plant which was known to perform well with a conventional controller. It did,
in fact, prove possible to train Adaline to act as a (rather crude) controller, but if this is to remain more than an academic exercise, the question must be asked if this idea can be applied to more complicated situations. If Adaline is to be considered seriously as a controller in some given situation, it must be able to control competently, and it must show a definite advantage over conventional controllers. There is one such area where Adaline might be used—where the aim is to duplicate the response of a human operator, or, more generally, where the controller must learn to control in the face of an absence of information about the dynamics of the plant and its existing controller. In the case of the human operator the main problem would be to present Adaline with the correct signals. The operator can be considered as having a switching surface (or hypersurface) which depends on his past experience in acting on observations of many variables. It may prove possible to train Adaline to duplicate his performance so that Adaline could eventually take over the operator's task. A by-product of this process may be the identification of some of the characteristics of the controller by examining Adaline's weights at the end of the training cycle.

The problem of training Adaline to act as the controller of an unknown and, as yet, uncontrolled plant is more difficult. A set of weights could be placed in Adaline, and the phase space, with certain of the plant variables as its coordinate axes, could be quantized and coded. On starting the plant
from certain initial conditions, Adaline would produce a response and drive the plant. As the resulting trajectory passes through different coded regions, different responses would be produced which would force the plant to behave in a certain way. If the resulting trajectory is undesirable, Adaline must be adjusted in some way to produce a satisfactory trajectory. The main problem would be to decide how to devise a suitable training procedure.

Although the possible uses for Adaline seem plausible, they depend on solving the problem of Adaline's poor response in certain areas of the phase plane. The schemes mentioned in sections 4.1 and 4.2 for improving Adaline's response near the origin do not attack the problem at its source, and an attempt should be made to improve the performance of Adaline itself. To date the information on the state of the plant has been presented to Adaline as a pattern after some form of coding. An area of research would be to consider more sophisticated ways of presenting Adaline with information about the state of the plant and, possibly, presenting additional information about the desired final state.
BIBLIOGRAPHY


A study of the application of pattern re...