IDENTIFICATION OF CONTROL CHART PATTERNS USING NEURAL NETWORKS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

in

Mechanical Engineering

By

ASIM PUJAPANDA

&

SIBASHISH PRIYADARSHI ACHARYA

Department of Mechanical Engineering

National Institute of Technology

Rourkela

2007

IDENTIFICATION OF CONTROL CHART PATTERNS USING NEURAL NETWORKS

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

In

Mechanical Engineering

By

ASIM PUJAPANDA

&

SIBASHISH PRIYADARSHI ACHARYA

Under the Guidance of

Prof. K.R. PATEL

Department of Mechanical Engineering

National Institute of Technology

Rourkela

2007

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, “IDENTIFICATION OF CONTROL CHART PATTERNS USING NEURAL NETWORKS” submitted by Shri Asim Pujapanda &
Shri Sibashish Priyadarshi Acharya in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Mechanical Engineering at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

Prof. K.R. Patel
Dept. of Mechanical Engineering
National Institute of Technology
Rourkela - 769008

National Institute of Technology
Rourkela

ACKNOWLEDGEMENT

I would like to articulate my deep gratitude to my project guide Prof. K.R. Patel who has always been my motivation for carrying out the project.

I would also like to express my gratitude to Prof. S.S. Mahapatra for his valuable support and guidance throughout the project .

It is my pleasure to refer Microsoft word 2003 of which the compilation of this report would have been impossible.

An assemblage of this nature could never have been attempted without reference to and inspiration from the works of others whose details are mentioned in reference section. I acknowledge my indebtedness to all of them.

Last but not the least to all of my friends who were patiently extended all sorts of help for accomplishing this undertaking.

Date:

Asim Pujapanda

Sibashish Priyadarshi Acharya

Dept. of Mechanical Engineering
National Institute of Technology
Rourkela – 769008

CONTENTS

Abstract.

To produce products with consistent quality, manufacturing processes need to be closely monitored for any deviations in the process. Proper analysis of control charts that are used **to** determine the state of the process not only requires a thorough knowledge and understanding of the underlying distribution theories associated with control charts, but also the experience of an expert in decision making. The present work proposes a modified backpropagation neural network methodology to identify and interpret various patterns of variations that can occur in a manufacturing process.

Control charts primarily in the form of X-bar chart are widely used to identify the situations when control actions will be needed for manufacturing systems. Various types of patterns are observed in control charts. Identification of these control chart patterns (CCPs) can provide clues to potential quality problems in the manufacturing process. Each type of control chart pattern has its own geometric shape and various related features can represent this shape.

This project formulates Shewhart mean (X-bar) and range (R) control charts for diagnosis and interpretation by artificial neural networks. Neural networks are trained to discriminate between samples from probability distributions considered within control limits and those which have shifted in both location and variance. Neural networks are also trained to recognize samples and predict future points from processes which exhibit long term or cyclical drift. The advantages and disadvantages of neural control charts compared to traditional statistical process control are discussed.

In processes, the causes of variations may be categorized as chance (unassignable) causes and special (assignable) causes. The variations due to chance causes are inevitable, and difficult to detect and identify. On the other hand, the variations due to special causes prevent the process being a stable and predictable. Such variations should be determined effectively and eliminated from the process by taking the necessary corrective actions to maintain the process in control and improve the quality of the products as well. In this study, a multilayered neural network trained with a back propagation algorithm was applied to pattern recognition on control charts. The neural network was experimented on a set of generated data.

LIST OF FIGURES :

Chapter 1

Overview of Control Charts and their usage in Process Control

1.1 **Introduction**

Control charts are commonly used in production environments to analyze process parameters to determine if a controlled process is within or out of control, i.e. to distinguish between assignable and common, also called chance, causes. Some manufacturing processes which benefit from control chart tracking are filtration, extraction, fermentation, distillation, refining, reaction, pressing, metal cutting, heat treatment, welding, casting, forging, extrusion, injection molding, spraying, and soldering (Although computationally simple, control charts are sometimes complex to use correctly because the sample points come from non-specified probabilistic distributions and usually require interpretation by a skilled user.

1.2 **Control Charts**

A control chart is a statistical tool used to distinguish between variation in a process resulting from common causes and variation resulting from special causes. It presents a graphic display of process stability or instability over time .
Every process has variation. Some variation may be the result of causes which are not normally present in the process. This could be **special cause variation**. Some variation is simply the result of numerous, ever-present differences in the process. This is **common cause variation**. Control Charts differentiate between these two types of variation. One goal of using a Control Chart is to achieve and maintain **process stability**.
Process stability is defined as a state in which a process has displayed a certain degree of consistency in the past and is expected to continue to do so in the future.
This consistency is characterized by a stream of data falling within **control limits** based on **plus or minus 3 standard deviations (3 sigma)** of the centerline]. We will discuss methods for calculating 3 sigma limits .
Control limits represent the limits of variation that should be expected from a process in a state of statistical control. When a process is in statistical control, any variation is the result of common causes that effect the entire production in a similar way. Control limits should not be confused with **specification limits**, which represent the desired process performance.

Chart's Usage of Control

A stable process is one that is consistent over time with respect to the center and the spread of the data. Control Charts help one monitor the behavior of your process to determine whether it is stable. Like Run Charts, they display data in the **time sequence in which they occurred**. However, Control Charts are more efficient that Run Charts in assessing and achieving process stability.

- Monitor process variation over time.
- Differentiate between special cause and common cause variation.
- Assess the effectiveness of changes to improve a process.

Communicate how a process performed during a specific period

Types of Control Charts

There are two main categories of Control Charts, those that display *attribute data*, and those that display *variables data*.

Attribute Data: This category of Control Chart displays data that result from counting the number of occurrences or items in a single category of similar items or occurrences. These "count" data may be expressed as pass/fail, yes/no, or presence/absence of a defect.

Variables Data: This category of Control Chart displays values resulting from the measurement of a continuous variable. Examples of variables data are elapsed time, temperature, and radiation dose.

While these two categories encompass a number of different types of Control Charts

There are three types that will work for the majority of the data analysis cases you will encounter. In this module, we will study the construction and application in these three types of Control Charts:

X-Bar and R Chart
Individual X and Moving Range Chart for Variables Data
Individual X and Moving Range Chart for Attribute Data

<u>Elements of a Control Chart</u>

Each Control Chart actually consists of *two graphs*, an **upper** and a **lower**, which are described below under *plotting areas*. A Control Chart is made up of eight elements.

1. **Title.** The title briefly describes the information which is displayed.
2. **Legend.** This is information on how and when the data were collected.
3. **Data Collection Section.** The counts or measurements are recorded in the data collection section of the Control Chart prior to being graphed.
4. **Plotting Areas.** A Control Chart has two areas—an upper graph and a lower graph—where the data is plotted.

a. The **upper graph** plots either the individual values, in the case of an Individual X and Moving Range chart, or the average (mean value) of the sample or subgroup in the case of an X-Bar and R chart.

b. The **lower graph** plots the moving range for Individual X and Moving Range charts, or the range of values found in the subgroups for X-Bar and R charts.

5. **Vertical or Y-Axis.** This axis reflects the magnitude of the data collected. The Y-axis shows the **scale of the measurement** for *variables data*, or the **count (frequency) or percentage of occurrence** of an event for *attribute data.*
6. **Horizontal or X-Axis.** This axis displays the chronological order in which the data were collected.
7. **Control Limits.** Control limits are set at a distance of 3 sigma above and 3 sigma below the centerline . They indicate variation from
the centerline and are calculated by using the actual values plotted on the Control Chart graphs.
8. **Centerline.** This line is drawn at the average or mean value of all the plotted data. The upper and lower graphs each have a separate centerline.

Steps for calculating and plotting an X-Bar and R Control Chart for Variables Data:

The X-Bar (arithmetic mean) and R (range) Control Chart is **used with variables data when subgroup or sample size is between 2 and 15**. The steps for constructing this type of Control Chart are:

Step 1 - Determine the data to be collected. Decide what questions about the process you plan to answer.
Step 2 - Collect and enter the data by subgroup. A subgroup is made up of variables data that represent a characteristic of a product produced by a process. The *sample size* relates to how large the subgroups are. Enter the individual subgroup measurements in time sequence in the portion of the data collection section of the Control Chart labeled *MEASUREMENTS*
STEP 3 - Calculate and enter the average for each subgroup. Use the formula below to calculate the average (mean) for each subgroup and enter it on the line labeled *Average* in the data collection section

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots x_n}{n}$$

Where: $\bar{x}$ = The average of the measurements within each subgroup
x_i = The individual measurements within a subgroup
n = The number of measurements within a subgroup

.

Step 4 - Calculate and enter the range for each subgroup. Use the following formula to calculate the range (R) for each subgroup. Enter the range for each subgroup on the line labeled *Range* in the data collection section .

RANGE = (Largest Value in each group) - (Smallest Value in each subgroup)

Range Example

Subgroup	1	2	3	4	5	6	7	8	9
X_1	15.3	14.4	15.3	15.0	15.3	14.9	15.6	14.0	14.0
X_2	14.9	15.5	15.1	14.8	16.4	15.3	16.4	15.8	15.2
X_3	15.0	14.8	15.3	16.0	17.2	14.9	15.3	16.4	13.6
X_4	15.2	15.6	18.5	15.6	15.5	16.5	15.3	16.4	15.0
X_5	16.4	14.9	14.9	15.4	15.5	15.1	15.0	15.3	15.0
Average:	15.36	15.04	15.82	15.36	15.98	15.34	15.52	15.58	14.56
Range:	1.5	1.2	3.6	1.2	1.9	1.6	1.4	2.4	1.6

Step 5 - Calculate the *grand mean* of the subgroup's average. The *grand mean* of the subgroup's average (X-Bar) becomes the centerline for the upper plot.

$$\bar{\bar{x}} = \frac{\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots \bar{x}_k}{k}$$

Where: $\bar{\bar{x}}$ = The grand mean of all the individual subgroup averages
$\bar{x}$ = The average for each subgroup
k = The number of subgroups

Step 6 - Calculate the average of the subgroup ranges. The average of all subgroups becomes the centerline for the lower plotting area.

$$\bar{R} = \frac{R_1 + R_2 + R_3 + \ldots R_k}{k}$$

Where: R_i = The individual range for each subgroup
$\bar{R}$ = The average of the ranges for all subgroups
k = The number of subgroups

Step 7 - Calculate the upper control limit (UCL) and lower control limit (LCL) for the averages of the subgroups. At this point, your chart will look like a Run Chart. Now, however, the uniqueness of the Control Chart becomes evident as you calculate the control limits. Control limits define the parameters for determining whether a process is in statistical control. To find the X-Bar control
limits, use the following formula:

Step 7 - Calculate the upper control limit (UCL) and lower control limit (LCL) for the averages of the subgroups. At this point, your chart will look like a Run Chart. Now, however, the uniqueness of the Control Chart becomes evident as you calculate the control limits. Control limits define the parameters for determining whether a process is in statistical control. To find the X-Bar control limits, use the following formula:

$$UCL_{\bar{X}} = \bar{\bar{X}} + A_2 \bar{R}$$
$$LCL_{\bar{X}} = \bar{\bar{X}} - A_2 \bar{R}$$

Step 8 - Calculate the upper control limit for the ranges. When the subgroup or sample size (n) is less than 7, there is no lower control limit. To find the upper control limit for the ranges, use the formula:

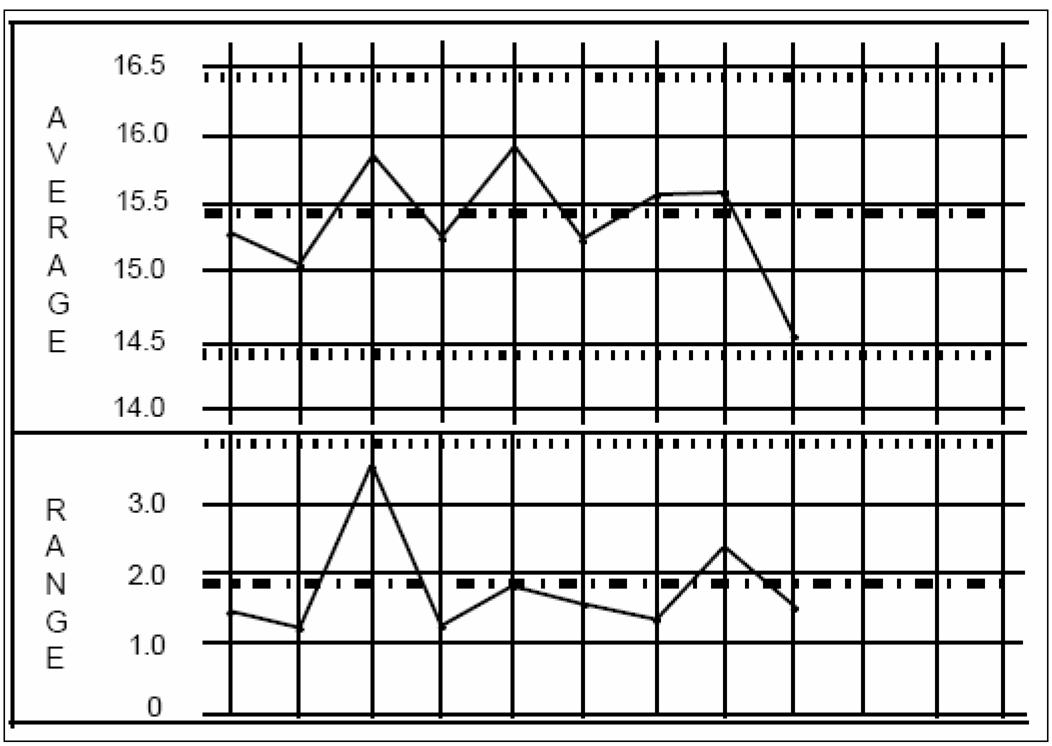
$$UCL_{\bar{R}} = D_4 \bar{R}$$
$$LCL_{\bar{R}} = D_3 \bar{R} \text{ (for subgroups} \geq 7)$$

Step 9 - Select the scales and plot the control limits, centerline, and data points, in each plotting area. The scales must be determined before the data points and centerline can be plotted. Once the upper and lower control limits have been computed, the easiest way to select the scales is to have the current data take up approximately 60 percent of the vertical (Y) axis. The scales for both the upper and lower plotting areas should allow for future high or low out-of control data points.

Plot each **subgroup average** as an individual data point in the **upper plotting area**. Plot individual **range** data points in the **lower plotting area**

Step 10 - Provide the appropriate documentation. Each Control Chart should be labeled with who, what, when, where, why, and how information to describe where the data originated, when it was collected, who collected it, any identifiable equipment or work groups, sample size, and all the other things necessary for understanding and interpreting it. It is important that the legend include all of the information that clarifies what the data describe.

SAMPLE X BAR AND R CHART :



1.3 PROCESS CONTROL

.

Interpretation of Control Charts

Process stability is reflected in the relatively constant variation exhibited in Control Charts. Basically, the data fall within a band bounded by the control limits. If a process is stable, the likelihood of a point falling outside this band is so small that such an occurrence is taken as a **signal of a special cause of variation**. In other words, something abnormal is occurring within your process. However, even though all the points fall inside the control limits, special cause variation may be at work. The presence of unusual patterns can be evidence that your process is not in statistical control. Such patterns are more likely to occur when one or more special causes is present. Control Charts are based on control limits which are 3 standard deviations (3 sigma) away from the centerline. One should resist the urge to narrow these limits in the hope of identifying special causes earlier. Experience has shown that **limits based on less than plus and minus 3 sigma may lead to false assumptions about special causes** operating in a process [Ref. 6, p. 82]. In other words, using control limits which are less than 3 sigma from the centerline may trigger a hunt for special causes when the process is already stable. The three standard deviations are sometimes identified by zones. Each zone's dividing line is exactly one-third the distance from the centerline to either the upper control limit or the lower control limit (Viewgraph 19).

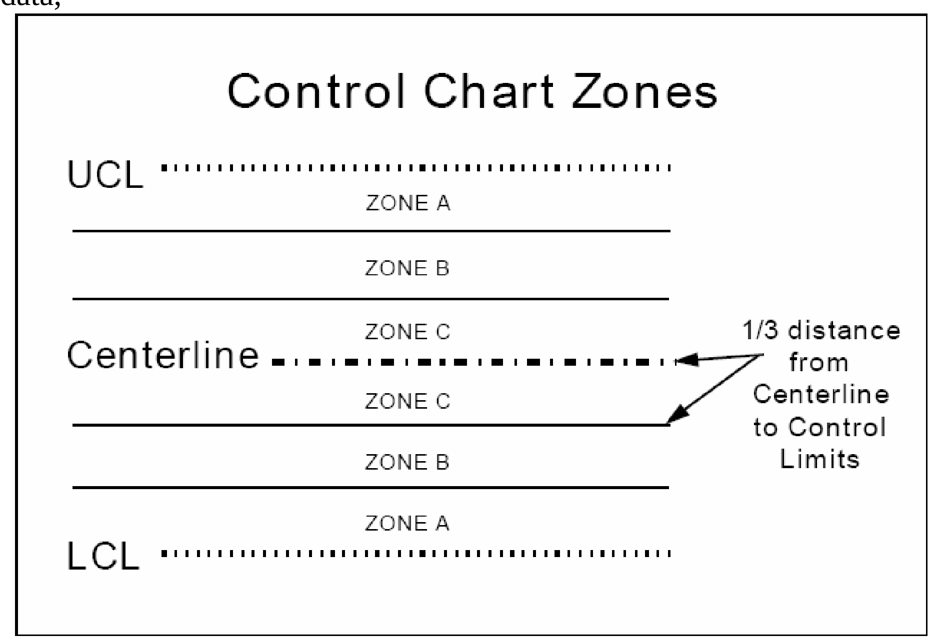
Zone A is defined as the area between 2 and 3 standard deviations from the centerline on both the plus and minus sides of the centerline.

Zone B is defined as the area between 1 and 2 standard deviations from the centerline on both sides of the centerline.

Zone C is defined as the area between the centerline and 1 standard deviation from the centerline, on both sides of the centerline.

There are two basic sets of rules for interpreting Control Charts: Rules for interpreting **X-Bar and R** Control Charts.

A similar, but separate, set of rules for interpreting **XmR** Control Charts. When a special cause is affecting the data,

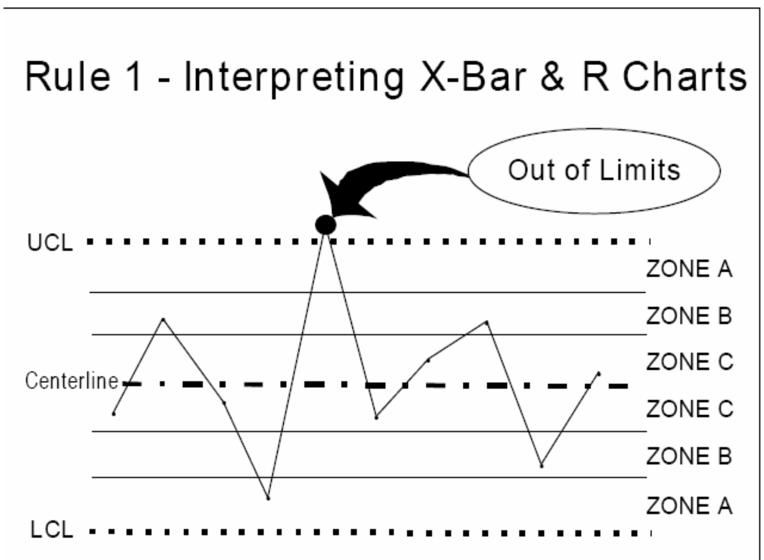


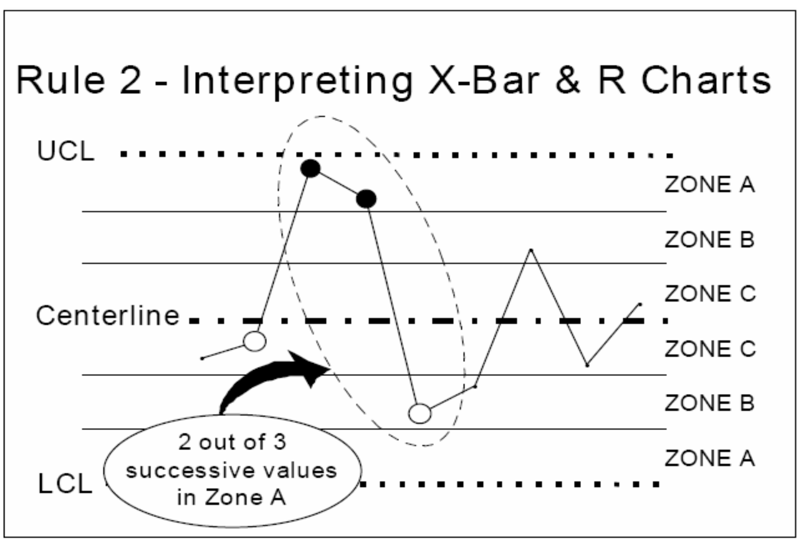
Rules for interpreting X-Bar and R Charts :

The nonrandom patterns displayed in a Control Chart will be fairly obvious. The key to these rules is recognizing that they serve as a signal for when to investigate what occurred in the process. When you are interpreting X-Bar and R Control Charts, you should apply the following set of rules:

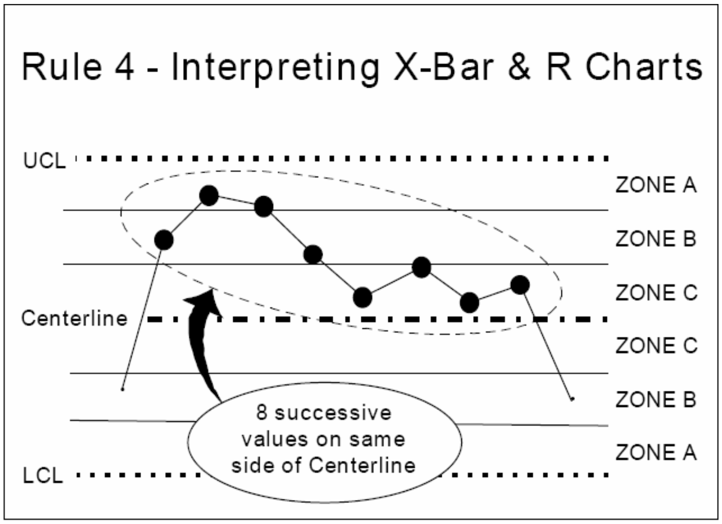
RULE 1: Whenever **a single point falls outside the 3 sigma control limits**, a lack of control is indicated. Since the probability of this happening is rather small, it is very likely not due to chance.

RULE 2: Whenever at least **2 out of 3 successive values** fall on the same side of the centerline and more than 2 sigma units away from the centerline (in Zone A or beyond), a lack of control is indicated. Note that the third point can be on **either side** of the centerline.





RULE 3: Whenever at least **4 out of 5 successive values** fall on the same side of the centerline and more than one sigma unit away from the centerline (in Zones A or B or beyond), a lack of control is indicated. Note that the fifth point can be on **either side** of the centerline.
RULE 4: Whenever at least **8 successive values** fall on the same side of the centerline, a lack of control is indicated.



Change the control limits
There are only three situations in which it is appropriate to change the control limits:
When removing out-of-control data points. When a special cause has been identified and removed while you are working to achieve process stability, you may want to delete the data points affected by special causes and use the remaining data to compute new control limits.
When replacing trial limits. When a process has just started up, or has changed, you may want to calculate control limits using only the limited data available. These limits are usually called *trial control limits*. You can calculate new limits every time you add new data. Once you have 20 or 30 groups of 4 or 5 measurements without a signal, you can use the limits to monitor future performance. You don’t need to recalculate the limits again unless fundamental changes are made to the process.

When there are changes in the process

When there are indications that your process has changed, it is necessary to recompute the control limits based on data collected since the change occurred. Some examples of such changes are the application of new or modified procedures, the use of different machines, the overhaul of existing machines, and the introduction of new suppliers of critical input materials.

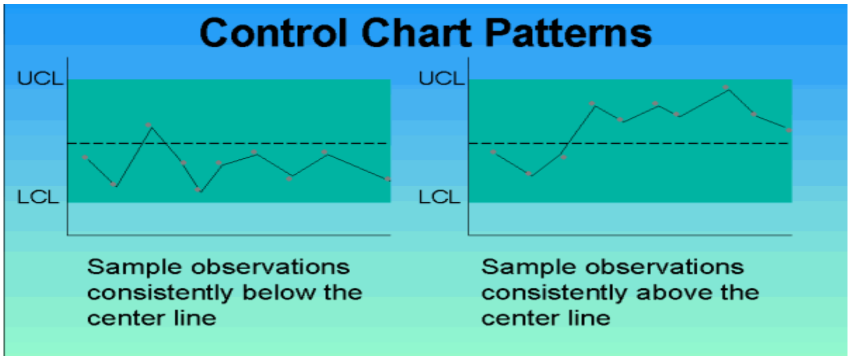
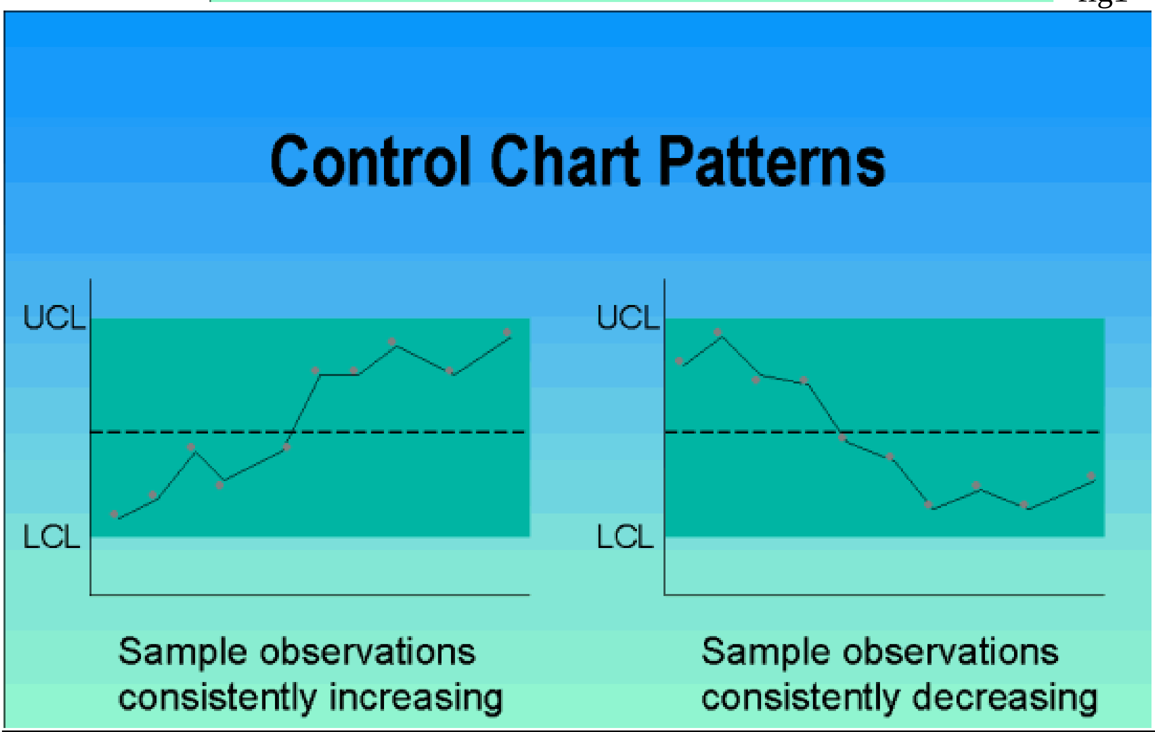
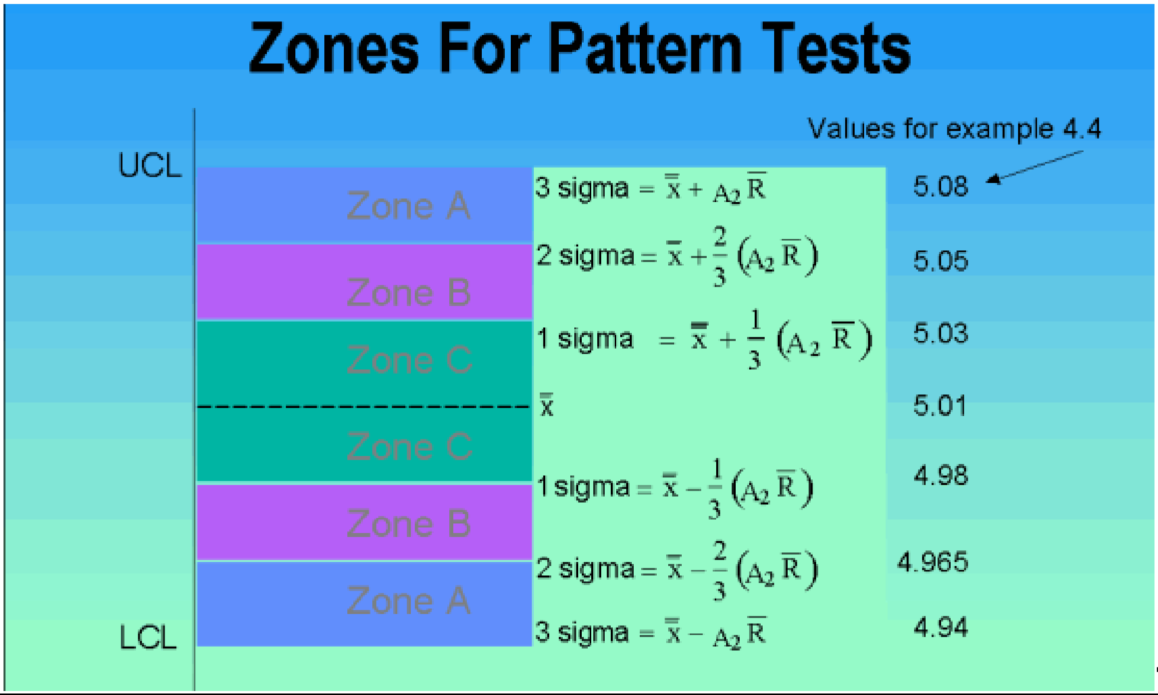


fig1



fg2



fg3

1.4 PROCESS CAPABILITY

Process capability examines

- the variability in process characteristics
- whether the process is capable of producing products which conforms to specifications

Process capability studies distinguish between conformance to ***control limits*** *and conformance to* ***specification limits*** *(also called* ***tolerance limits****)*

- if the process mean is in control, then virtually all points will remain within control limits
- staying within control limits does not necessarily mean that specification limits are satisfied
- specification limits are usually dictated by customers

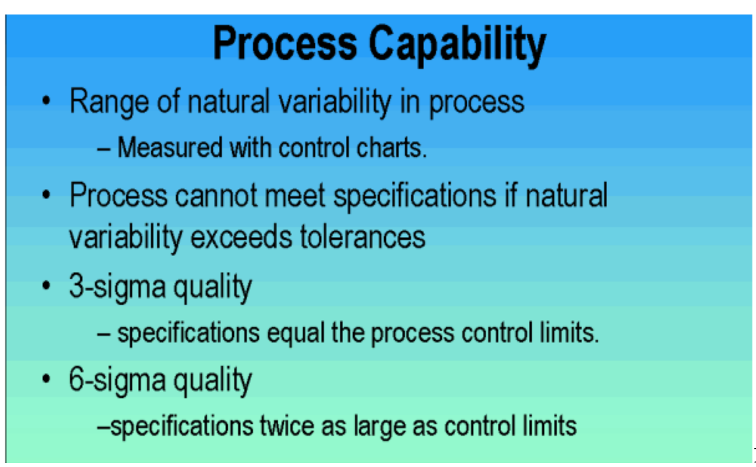


fig4

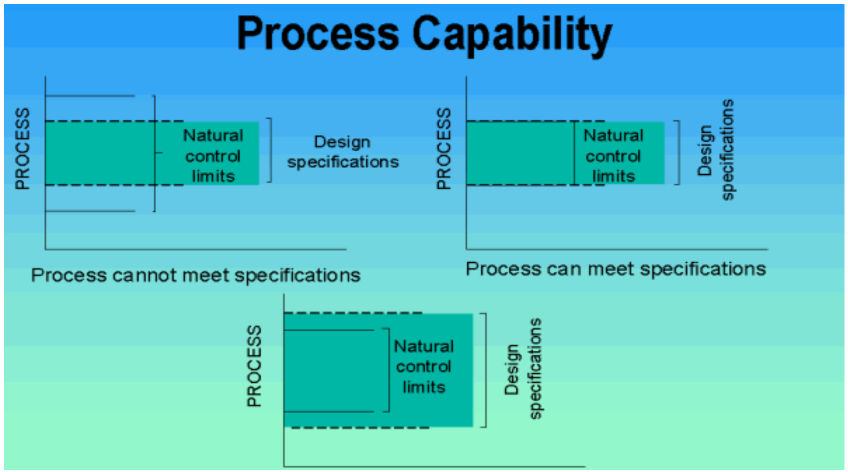
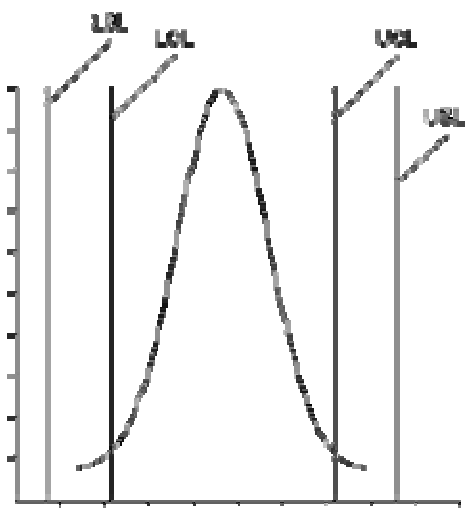


fig5

PROCESS CAPABILITY CONCEPTS :

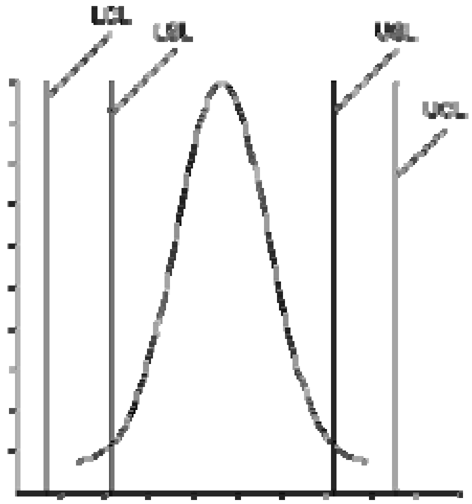
The following distributions show different process scenarios. Note the relative positions of the control limits and specification limits.

1. 

In control and product meets specifications.

Control limits are within specification limits

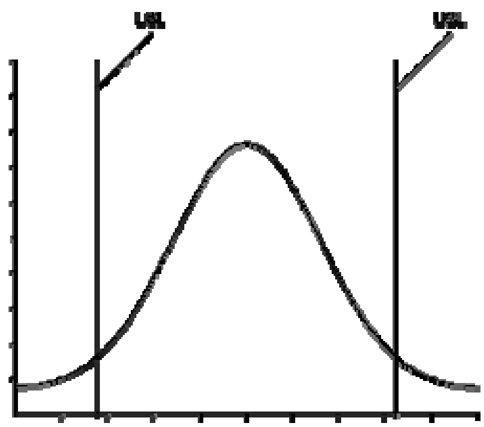
2.



In control but some products do not meet specifications.

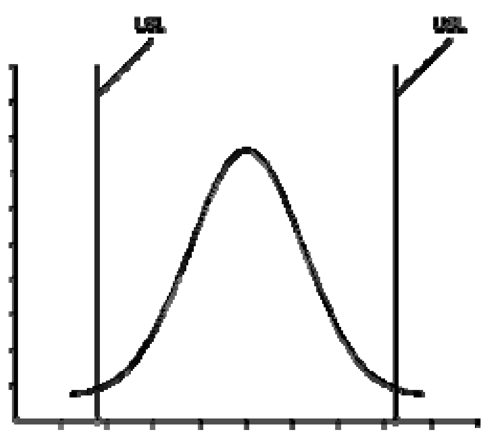
Specification limits are within control limits

3.



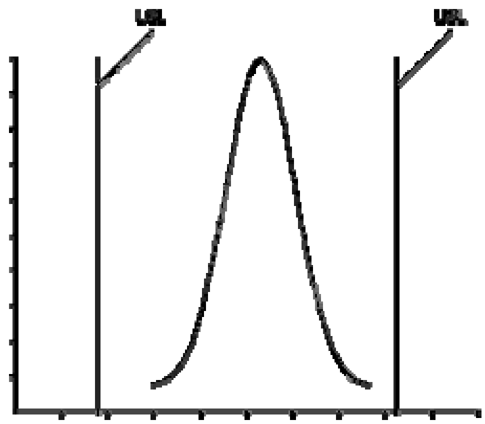
Data from process with low capability

4.



Data from process with medium capability

5.



Data from process with high capability

6. Process capability: capability index

The capability index is defined as:
Cp = (allowable range)/6s = (USL - LSL)/6s
The capability index show how well a process is able to meet specifications. The higher the value of the index, the more capable is the process:

- Cp < 1 (process is unsatisfactory)
- 1 < Cp < 1.6 (process is of medium relative capability)
- Cp > 1.6 (process shows high relative capability)

7. Process capability: process performance index

The capability index

- considers only the spread of the characteristic in relation to specification limits
- assumes two-sided specification limits

The product can be bad if the mean is not set appropriately. The process performance index takes account of the mean (m) and is defined as:
Cpk = min[(USL - m)/3s, (m - LSL)/3ss]
The process performance index can also accommodate one sided specification limits

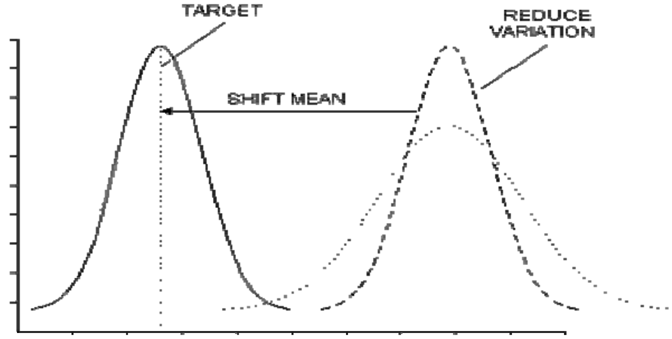
- for upper specification limit: Cpk = (USL - m)/3s
- for lower specification limit: Cpk = (m - LSL)/3s

8. Process capability: the message

The message from process capability studies is:

- first reduce the variation in the process
- then shift the mean of the process towards the target

This procedure is illustrated in the diagram



Chapter 2

Artificial Neural Networks

2.1 Introduction

Artificial neural networks trained by supervised techniques have been documented as good alternatives for pattern classification and prediction. These skills can be put to use for the interpretation of process control from input of control chart samples. Through learning of varying input conditions matched with control status, a network can decide on control status when faced with new inputs. Besides exceeding control boundaries, control chart points can provide information about the long term condition of the process through symptomatic shapes, runs and drifts. If these can be correctly identified from a small sample by neural networks, then the process can be investigated expediently. Neural networks can also forecast future control chart point(s), thus contributing to the diagnosis of process condition in borderline conditions.

Introduction to Neural Networks

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Historical background

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of During this period when funding and professional support was minimal, important advances were made by relatively few researchers. These pioneers were able to develop convincing technology which surpassed the limitations enthusiasm, the field survived a period of frustration and disrepute. identified by Minsky and Papert. Minsky and Papert, published a book (in 1969) in which they summed up a general feeling of frustration (against neural networks) among researchers, and was thus accepted by most without further analysis. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding. The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do too much.

Usage of Neural Networks

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse. This expert can then be used to provide projections given new situations of interest and answer "what if" questions. Other advantages include:

1. Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
2. Self-Organisation: An ANN can create its own organisation or representation of the information it receives during learning time.
3. Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Neural networks versus conventional computers

Neural networks take a different approach to problem solving than that of conventional computers. Conventional computers use an algorithmic approach i.e. the computer follows a set of instructions in order to solve a problem. Unless the specific steps that the computer needs to follow are known the computer cannot solve the problem. That restricts the problem solving capability of conventional computers to problems that we already understand and know how to solve. But computers would be so much more useful if they could do things that we don't exactly know how to do.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

On the other hand, conventional computers use a cognitive approach to problem solving; the way the problem is to solved must be known and stated in small unambiguous instructions. These instructions are then converted to a high level language program and then into machine code that the computer can understand. These machines are totally predictable; if anything goes wrong is due to a software or hardware fault.

Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks are more suited to an algorithmic approach like arithmetic operations and tasks that are more suited to neural networks. Even more, a large number of tasks, require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency.

Human and Artificial Neurons - the similarities

The process of human brain learning:

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes

Components of a neuron :

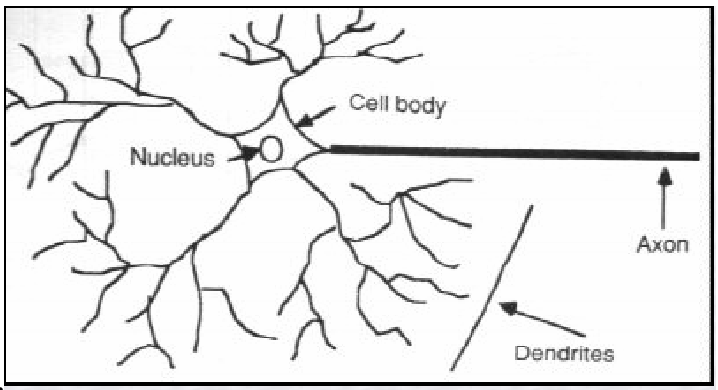


fig6

Synapse :

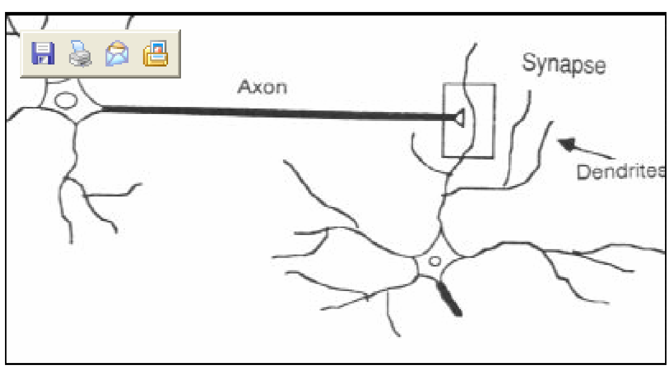


fig 7

A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use . It resembles the brain in two respects.:-*1. Knowledge is acquired by the network through a learning process.*

2. Interneuron connection strengths, known as synaptic weights, are used to store the knowledge.

Artificial Neurons

We conduct these neural networks by first trying to deduce the essential features of neurons and their interconnections. We then typically program a computer to simulate these features. However because our knowledge of neurons is incomplete and our computing power is limited, our models are necessarily gross idealisations of real networks of neurons.

The Neuron Model :

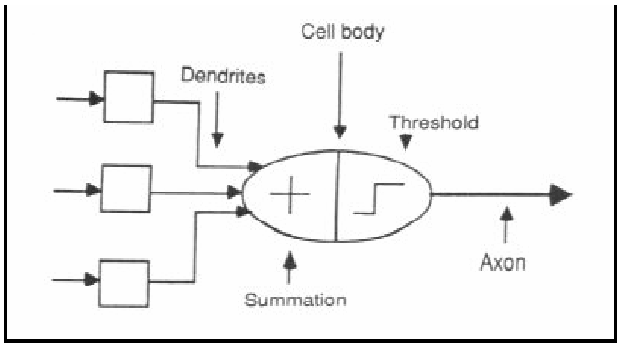


fig8

A simple neuron

An artificial neuron is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

Complicated neurons

The previous neuron doesn't do anything that conventional conventional computers don't do already. A more sophisticated neuron (figure 2) is the McCulloch and Pitts model (MCP). The difference from the previous model is that the inputs are 'weighted', the effect that each input has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

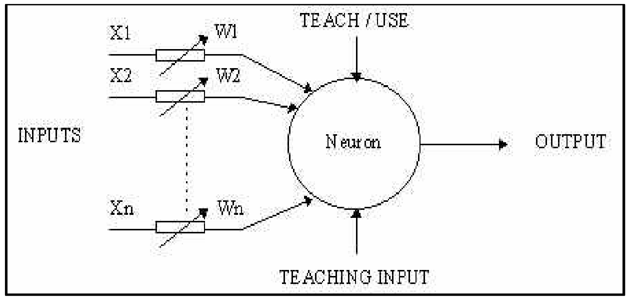
An MCP neuron : 

fig9

In mathematical terms, the neuron fires if and only if;

X1W1 + X2W2 + X3W3 + ... > T

The addition of input weights and of the threshold makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation. The former is used in feed-forward networks and the latter in feedback networks.

2.2 <u>Architecture of neural networks</u>

Feed-forward networks

Feed-forward ANNs (figure 1) allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward ANNs tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organisation is also referred to as bottom-up or top-down.

Feedback networks

Feedback networks (figure 1) can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organisations.

An example of a simple Feed Forward network :

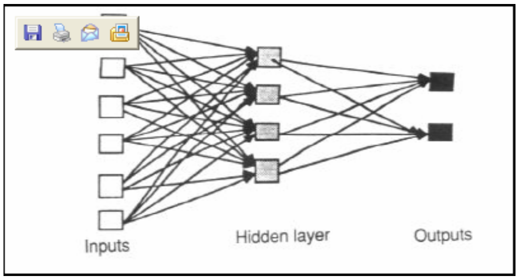


fig10

Example of a complicated network :

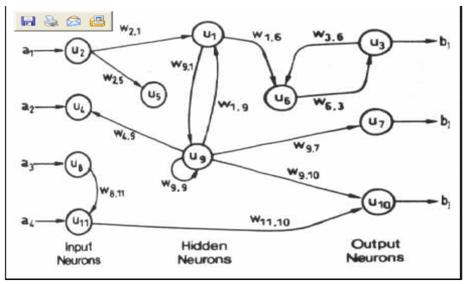


fig11

Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of **"output**" units.

- The activity of the input units represents the raw information that is fed into the network.

- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.

- The behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

NETWORK LAYERS :

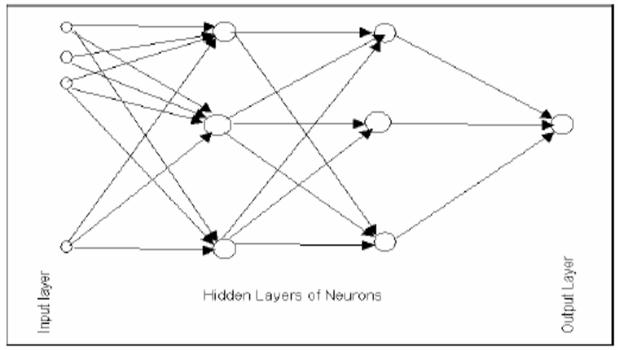


fig12

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

We also distinguish single-layer and multi-layer architectures. The single-layer organisation, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organisations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

Perceptrons**:** The most influential work on neural nets in the 60's went under the heading of 'perceptrons' a term coined by Frank Rosenblatt. The perceptron (figure 4.4) turns out to be an MCP model (neuron with weighted inputs) with some additional, fixed, pre--processing. Units labelled A1, A2, Aj , Ap are called association units and their task is to extract specific, localised featured from the input images. Perceptrons mimic the basic idea behind the mammalian visual system. They were mainly used in pattern recognition even though their capabilities extended a lot more.

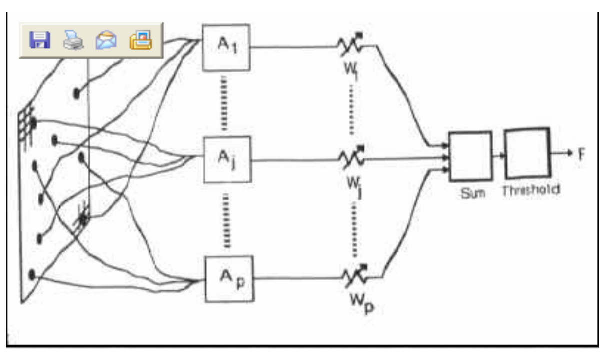


fig13

2.3 The Learning Process

The memorisation of patterns and the subsequent response of the network can be categorised into two general paradigms:

associative mapping in which the network learns to produce a particular pattern on the set of input units whenever another particular pattern is applied on the set of input units. The associative mapping can generally be broken down into two mechanisms:

auto-association: an input pattern is associated with itself and the states of input and output units coincide. This is used to provide pattern completion, ie to produce a pattern whenever a portion of it or a distorted pattern is presented. In the second case, the network actually stores pairs of patterns building an association between two sets of patterns.

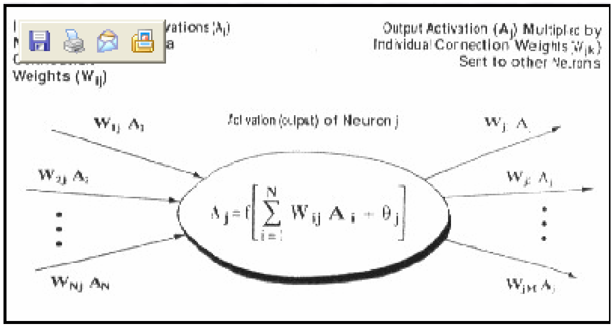
hetero-association: is related to two recall mechanisms:

nearest-neighbour recall, where the output pattern produced corresponds to the input pattern stored, which is closest to the pattern presented, and

interpolative recall, where the output pattern is a similarity dependent interpolation of the patterns stored corresponding to the pattern presented. Yet another paradigm, which is a variant associative mapping is classification, ie when there is a fixed set of categories into which the input patterns are to be classified.

regularity detection in which units learn to respond to particular properties of the input patterns. Whereas in associative mapping the network stores the relationships among patterns, in regularity detection the response of each unit has a particular 'meaning'. This type of learning mechanism is essential for feature discovery and knowledge representation.

Every neural network possesses knowledge which is contained in the values of the connections weights. Modifying the knowledge stored in the network as a function of experience implies a learning rule for changing the values of the weights.



Information is stored in the weight matrix W of a neural network. Learning is the determination of the weights. Following the way learning is performed, we can distinguish two major categories of neural networks:

- **fixed networks** in which the weights cannot be changed, ie dW/dt=0. In such networks, the weights are fixed a priori according to the problem to solve.

- **adaptive networks** which are able to change their weights, ie dW/dt not= 0.

All learning methods used for adaptive neural networks can be classified into two major categories:

- **Supervised learning** which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning.
An important issue concerning supervised learning is the problem of error convergence, ie the minimisation of error between the desired and computed unit values. The aim is to determine a set of weights which minimises the error. One well-known method, which is common to many learning paradigms is the least mean square (LMS) convergence.

- **Unsupervised learning** uses no external teacher and is based upon only local information. It is also referred to as self-organisation, in the sense that it self-organises data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian lerning and competitive learning.
From Human Neurons to Artificial Neurons the aspect of learning concerns the distinction or not of a separate phase, during which the network is trained, and a subsequent operation phase. We say that a neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas supervised learning is performed on-line.

Transfer Function

The behaviour of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- ***linear (or ramp)***
- ***threshold***
- ***sigmoid***

For **linear units**, the output activity is proportional to the total weighted output.

For **threshold units**, the output is set at one of two levels, depending on whether the total input is greater than or less than some threshold value.

For **sigmoid units**, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another (see figure 4.1), and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

We can teach a three-layer network to perform a particular task by using the following procedure:

1. We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units.
2. We determine how closely the actual output of the network matches the desired output.
3. We change the weight of each connection so that the network produces a better approximation of the desired output.

The Back-Propagation Algorithm

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights (**EW**). In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back propagation algorithm is the most widely used method for determining the **EW**.The back-propagation algorithm is easiest to understand if all the units in the network are linear. The algorithm computes each **EW** by first computing the **EA**, the rate at which the error changes as the activity level of a unit is changed. For output units, the **EA** is simply the difference between the actual and the desired output. To compute the **EA** for a hidden unit in the layer just before the output layer, we first identify all the weights between that hidden unit and the output units to which it is connected. We then multiply those weights by the **EA**s of those output units and add the products. This sum equals the **EA** for the chosen hidden unit. After calculating all the **EA**s in the hidden layer just before the output layer, we can compute in like fashion the **EA**s for other layers, moving from layer to layer in a direction opposite to the way activities propagate through the network. This is what gives back propagation its name. Once the **EA** has been computed for a unit, it is straight forward to compute the **EW** for each incoming connection of the unit. The **EW** is the product of the EA and the activity through the incoming connection. The back-propagation algorithm includes an extra step. Before back-propagating, the **EA** must be converted into the **EI**, the rate at which the error changes as the total input received by a unit is changed.

2.4 Applications of neural networks

Neural Networks in Practice

Given the description of neural networks and the way they work, neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.

Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs including:

- sales forecasting
- industrial process control
- customer research and target marketing
- data validation
- risk management

But to give some more specific examples-ANN are also used in the following specific paradigms recognition of speakers in communications, diagnosis of hepatitis recovery of telecommunications from faulty software, interpretation of multi-meaning Chinese words undersea mine detection; texture analysis; three-dimensional object recognition; hand-written word recognition; and facial recognition.

Chapter 3

Artificial Neural Networks in Process Control

3.1 **INTRODUCTION** :

Artificial neural networks trained by supervised techniques have been documented as good alternatives for pattern classification and prediction. These skills can be put to use for the interpretation of process control from input of control chart samples. Through learning of varying input conditions matched with control status, a network can decide on control status when faced with new inputs. Besides exceeding control boundaries, control chart points can provide information about the long term condition of the process through symptomatic shapes, runs and drifts. If these can be correctly identified from a small sample by neural networks, then the process can be investigated expediently. Neural networks can also forecast future control chart point(s), thus contributing to the diagnosis of process condition in borderline conditions.

The project focuses first on the data representation required to successfully train a ***back propagation neural network*** to recognize control chart patterns from input of control chart samples. Number of inputs and preprocessing which enhance network performance are discussed. The ability of networks to discriminate within control from out of control situations based on small, probabilistic samples is presented. A second focus is the ability of the network to recognize sequences of noisy data points as belonging to patterns which reflect long term undesirable drift in the process,

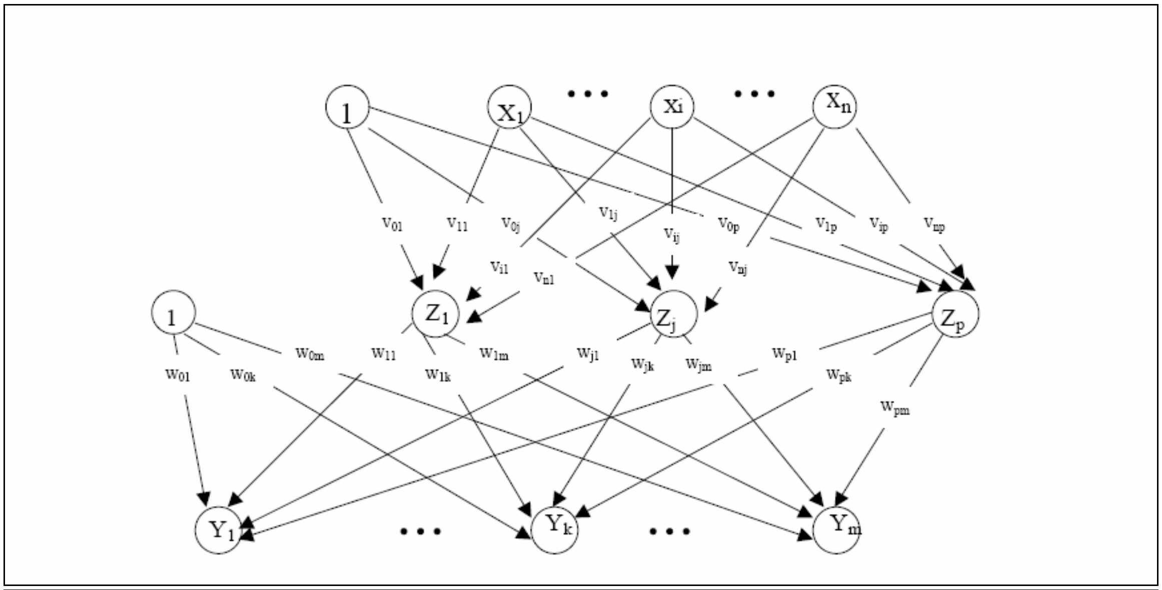
Thus earning of varying input conditions matched with control status, a network can decide on control status when faced with new inputs. Besides exceeding control boundaries, control chart points can provide information about the long term condition of the process through symptomatic shapes, runs and drifts.

Neural networks have been noted as being particularly advantageous for modeling systems which contain noisy, fuzzy and uncertain elements. They learn models by iterating through a large number of exemplar vectors. Relationships can be ***auto-associative*** (relating an input with itself), or hetero-associative (relating an input with another output). Learning can take place through internal grouping (self organizing or competitive learning) or through paired training sets (***supervised learning***). For modeling control data, a supervised approach is preferable since calibrated training data is usually available and it is advantageous to pre-specify the desired output.

The most well known of supervised techniques is ***back propagation***, which adjusts initially randomized weights during training according to the steepest gradient along the error surface. Weights are adjusted in proportion to their contribution to the output by recycling the squared error signal back through the layers of weights. Typical back propagation neural networks, which are more properly termed ***multi-layered perceptrons*** trained by back propagation, are fully connected, feed forward only, and use a ***sigmoidal transfer function*** at the nodes to evaluate weighted input sums. An input layer, an output layer and at least one hidden layer are required to model nonlinear systems however it has been suggested that for analog input, a two hidden layer network is superior.

3.2 Back-Propagation Algorithm

Back propagation Neural Network with One Hidden Layer



Back propagation was created by generalizing the ***Widrow-Hoff learning rule*** to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding output vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by you. Networks with biases, a sigmoid layer, and a linear output layer are capable ofapproximating any function with a finite number of discontinuities.

The central idea behind this solution is that the errors for the units of the hidden layer are determined by back-propagating the errors of the units of the output layer. For this reason the method is often called the back-propagation learning rule. Back-propagation can also be considered as a generalisation of the delta rule for non-linear activation functions1 and multilayer networks

Notation

We use the following notation in our formulae.

i an input unit;

h a hidden unit;

o an output unit;

$\mathbf{x}^p$ the pth input pattern vector;

x_j^p the jth element of the pth input pattern vector;

$\mathbf{s}^p$ the input to a set of neurons when input pattern vector p is clamped (i.e., presented to the network); often: the input of the *network* by clamping input pattern vector p;

$\mathbf{d}^p$ the desired output of the *network* when input pattern vector p was input to the network;

d_j^p the jth element of the desired output of the network when input pattern vector p was input to the network;

$\mathbf{y}^p$ the activation values of the *network* when input pattern vector p was input to the network;

y_j^p the activation values of element j of the network when input pattern vector p was input to the network;

W the matrix of connection weights;

$\mathbf{w}_j$ the weights of the connections which feed into unit j;

w_{jk} the weight of the connection from unit j to unit k;

$\mathcal{F}_j$ the activation function associated with unit j;

γ_{jk} the learning rate associated with weight w_{jk};

$\boldsymbol{\theta}$ the biases to the units;

θ_j the bias input to unit j;

U_j the threshold of unit j in $\mathcal{F}_j$;

E^p the error in the output of the network when input pattern vector p is input;

$\mathcal{E}$ the energy of the network.

TERMINOLOGY :

Output vs. activation of a unit. Since there is no need to do otherwise, we consider the output and the activation value of a unit to be one and the same thing. That is, the output of each neuron equals its activation value.

Bias, offset, threshold. These terms all refer to a constant (i.e., independent of the network input but adapted by the learning rule) term which is input to a unit. They may be used interchangeably, although the latter two terms are often envisaged as a property of the activation function. Furthermore, this external input is usually implemented (and can be written) as a weight from a unit with activation value 1.

Number of layers. In a feed-forward network, the inputs perform no computation and their layer is therefore not counted. Thus a network with one input layer, one hidden layer, and one output layer is referred to as a network with two layers. This convention is widely though not yet universally used.

1989; Funahashi, 1989; Cybenko, 1989; Hartman, Keeler, & Kowalski, 1990) that only one
layer of hidden units su_ces to approximate any function with _nitely many discontinuities to

Multi-layer feed-forward networks :A feed-forward network has a layered structure. Each layer consists of units which receive their input from units from a layer directly below and send their output to units in a layer directly above the unit. There are no connections within a layer. The Ni inputs are fed into the first layer of Nh;1 hidden units. The input units are merely 'fan-out' units; no processing takes place in these units. The activation of a hidden unit is a function Fi of the weighted inputs plus a bias,. The output of the hidden units is distributed over the next layer of Nh;2 hidden units, until the last layer of hidden units, of which the outputs are fed into a layer of No output units

Although back-propagation can be applied to networks with any number of layers, just as for networks with binary units it has been shown that only one layer of hidden units suffices to approximate any function with finitely many discontinuities to arbitrary precision, provided the activation functions of the hidden units are non-linear (the universal approximation theorem). In most applications a feed-forward network with a single layer of hidden units is used with a sigmoid activation function for the units.

A multi-layer network with l layers of units :

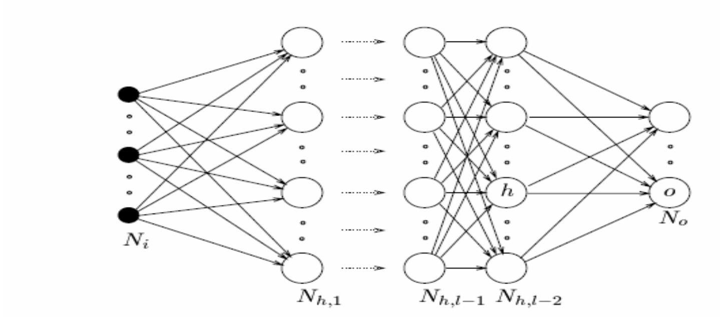


fig 14

3.3 <u>The Generalised delta rule</u>

Since we are using units with nonlinear activation functions, we have to generalise the delta rule for linear functions to the set of non-linear activation functions.

The activation is a differentiable function of the total input, given by

$$y_k^p = \mathcal{F}(s_k^p), \tag{4.1}$$

in which

$$s_k^p = \sum_j w_{jk} y_j^p + \theta_k. \tag{4.2}$$

To get the correct generalisation of the delta rule

set

$$\Delta_p w_{jk} = -\gamma \frac{\partial E^p}{\partial w_{jk}}. \tag{4.3}$$

The error measure E^p is defined as the total quadratic error for pattern p at the output units:

$$E^p = \tfrac{1}{2} \sum_{o=1}^{N_o} (d_o^p - y_o^p)^2, \tag{4.4}$$

where d_o^p is the desired output for unit o when pattern p is clamped. We further set $E = \sum_p E^p$ as the *summed squared error*. We can write

$$\frac{\partial E^p}{\partial w_{jk}} = \frac{\partial E^p}{\partial s_k^p} \frac{\partial s_k^p}{\partial w_{jk}}. \tag{4.5}$$

By equation (4.2) we see that the second factor is

$$\frac{\partial s_k^p}{\partial w_{jk}} = y_j^p. \tag{4.6}$$

When we define

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p}, \tag{4.7}$$

we will get an update rule which is equivalent to the delta rule resulting in a gradient descent on the error surface if we make the weight changes according to:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p. \tag{4.8}$$

The trick is to figure out what δ_k^p should be for each unit k in the network. The interesting result, which we now derive, is that there is a simple recursive computation of these δ's which can be implemented by propagating error signals backward through the network.

To compute δ_k^p we apply the chain rule to write this partial derivative as the product of two factors, one factor reflecting the change in error as a function of the output of the unit and one reflecting the change in the output as a function of changes in the input. Thus, we have

$$\delta_k^p = -\frac{\partial E^p}{\partial s_k^p} = -\frac{\partial E^p}{\partial y_k^p}\frac{\partial y_k^p}{\partial s_k^p}. \tag{4.9}$$

Let us compute the second factor. By equation (4.1) we see that

$$\frac{\partial y_k^p}{\partial s_k^p} = \mathcal{F}'(s_k^p), \tag{4.10}$$

which is simply the derivative of the squashing function $\mathcal{F}$ for the kth unit, evaluated at the net input s_k^p to that unit. To compute the first factor of equation (4.9), we consider two cases. First, assume that unit k is an output unit $k = o$ of the network. In this case, it follows from the definition of E^p that

$$\frac{\partial E^p}{\partial y_o^p} = -(d_o^p - y_o^p), \tag{4.11}$$

which is the same result as we obtained with the standard delta rule. Substituting this and equation (4.10) in equation (4.9), we get

$$\delta_o^p = (d_o^p - y_o^p)\,\mathcal{F}_o{}'(s_o^p) \tag{4.12}$$

for any output unit o. Secondly, if k is not an output unit but a hidden unit $k = h$, we do not readily know the contribution of the unit to the output error of the network. However, the error measure can be written as a function of the net inputs from hidden to output layer; $E^p = E^p(s_1^p, s_2^p, \ldots, s_j^p, \ldots)$ and we use the chain rule to write

$$\frac{\partial E^p}{\partial y_h^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} \frac{\partial s_o^p}{\partial y_h^p} = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} \frac{\partial}{\partial y_h^p} \sum_{j=1}^{N_h} w_{ko} y_j^p = \sum_{o=1}^{N_o} \frac{\partial E^p}{\partial s_o^p} w_{ho} = -\sum_{o=1}^{N_o} \delta_o^p w_{ho}. \tag{4.13}$$

Substituting this in equation (4.9) yields

$$\delta_h^p = \mathcal{F}'(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho}. \tag{4.14}$$

Equations (4.12) and (4.14) give a recursive procedure for computing the δ's for all units in the network, which are then used to compute the weight changes according to equation (4.8). This procedure constitutes the generalised delta rule for a feed-forward network of non-linear units.

Concept of The Delta Rule : When a learning pattern is clamped, the activation values are propagated to the output units, and the actual network output is compared with the desired output values, we usually end up with an error in each of the output units. Let's call this error eo for a particular output unit o. We have to bring e0 to zero. The simplest method to do this is the greedy method: we strive to change the connections in the neural network in such a way that, next time around, the error eo will be zero for this *particular pattern.* We know from the delta rule that, in order to reduce an error,

we have to adapt its incoming weights according to

$$\Delta w_{ho} = (d_o - y_o) y_h. \tag{4.15}$$

That's step one. But it alone is not enough: when we only apply this rule, the weights from input to hidden units are never changed, and we do not have the full representational power of the feed-forward network as promised by the universal approximation theorem. In order to adapt the weights from input to hidden units, we again want to apply the delta rule. In this case, however, we do not have a value for _ for the hidden units. This is solved by the chain rule which does the following: distribute the error of an output unit o to all the hidden units that is it connected to, weighted by this connection. Differently put, a hidden unit h receives a delta from each output unit o equal to the delta of that output unit weighted with (= multiplied by) the weight of the connection between those units. In symbols:

$$\delta_h = \sum_o \delta_o w_{ho}$$

3.4 Working with Back Propagation

The application of the generalised delta rule thus involves two phases: During the first phase the input $\mathbf{x}$ is presented and propagated forward through the network to compute the output values y_o^p for each output unit. This output is compared with its desired value d_o, resulting in an error signal δ_o^p for each output unit. The second phase involves a backward pass through the network during which the error signal is passed to each unit in the network and appropriate weight changes are calculated.

Weight adjustments with sigmoid activation function. The results from the previous section can be summarised in three equations:

- The weight of a connection is adjusted by an amount proportional to the product of an error signal δ, on the unit k receiving the input and the output of the unit j sending this signal along the connection:

$$\Delta_p w_{jk} = \gamma \delta_k^p y_j^p. \tag{4.16}$$

- If the unit is an output unit, the error signal is given by

$$\delta_o^p = (d_o^p - y_o^p)\, \mathcal{F}'(s_o^p). \tag{4.17}$$

Taking the activation function F as the 'sigmoid' function

$$y^p = \mathcal{F}(s^p) = \frac{1}{1 + e^{-s^p}}. \tag{4.18}$$

In this case the derivative is equal to

$$\begin{aligned} \mathcal{F}'(s^p) &= \frac{\partial}{\partial s^p} \frac{1}{1 + e^{-s^p}} \\ &= \frac{1}{(1 + e^{-s^p})^2} (-e^{-s^p}) \\ &= \frac{1}{(1 + e^{-s^p})} \frac{e^{-s^p}}{(1 + e^{-s^p})} \\ &= y^p(1 - y^p). \end{aligned} \tag{4.19}$$

such that the error signal for an output unit can be written as:

$$\delta_o^p = (d_o^p - y_o^p)\, y_o^p(1 - y_o^p). \tag{4.20}$$

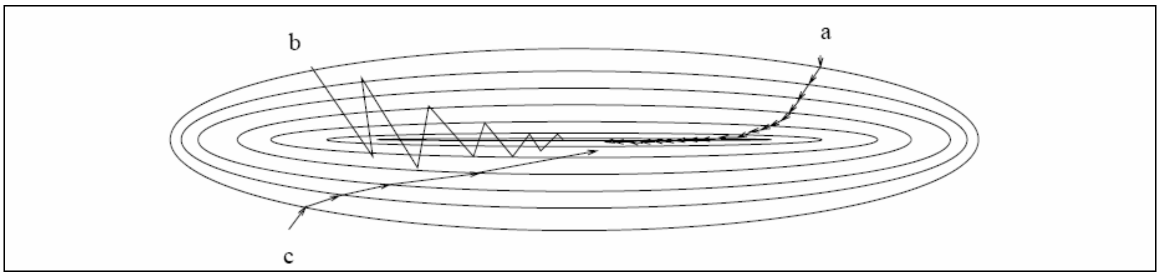
- The error signal for a hidden unit is determined recursively in terms of error signals of the units to which it directly connects and the weights of those connections. For the sigmoid activation function:

$$\delta_h^p = \mathcal{F}'(s_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho} = y_h^p(1 - y_h^p) \sum_{o=1}^{N_o} \delta_o^p w_{ho}. \tag{4.21}$$

Learning rate and momentum. The learning procedure requires that the change in weight is proportional to $\partial E^p/\partial w$. True gradient descent requires that infinitesimal steps are taken. The constant of proportionality is the learning rate γ. For practical purposes we choose a learning rate that is as large as possible without leading to oscillation. One way to avoid oscillation at large γ, is to make the change in weight dependent of the past weight change by adding a *momentum* term:

$$\Delta w_{jk}(t+1) = \gamma \delta_k^p y_j^p + \alpha \Delta w_{jk}(t), \tag{4.22}$$

where t indexes the presentation number and _ is a constant which determines the effect of the previous weight change.. When no momentum term is used, it takes a long time before the minimum has been reached with a low learning rate, whereas for high learning rates the minimum is never reached because of the oscillations. When adding the momentum term, the minimum will be reached faster.



The descent in weight space. a) for small learning rate; b) for large learning rate: the oscillations, and c) with large learning rate and momentum term added

Back Propagation Neural Network Architecture

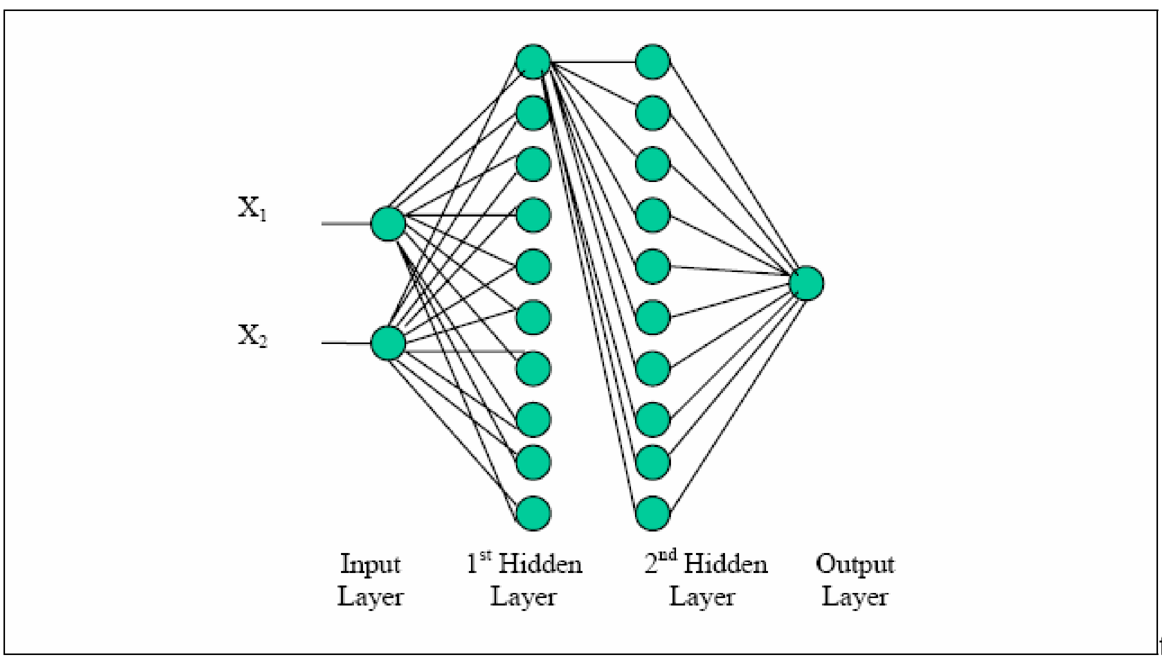


fig 15

Chapter 4

Application of the Neural Network Model to the Identification of Control Chart Patterns(CCP)

4.1 PATTERN ANALYSIS IN CONTROL CHARTS

. In such a case, one can only suspect that the process has gone out of control and if it is not affordable to continue the process, it has to be halted to diagnose the cause, which may lead to Control charts do not solve problems, but provide information on the basis of which we can reasonably (in a statistical sense) presume that process is “in control” or “out of control”. Actual analysis of the control chart is done manually and this is somewhat difficult. Proper analysis not only requires thorough knowledge and understanding of the underlying distribution theories associated with control charts, but the experience that only an expert can provide as well. While using control charts, the judgement is usually left to the user. Another major limitation of the control charts is that they only use the information about the process contained in the last observation to decide about the state of the process. This feature makes the control chart relatively insensitive to small shifts in the process. Furthermore, a point falling out of control limits does not necessarily indicate an out-of-control situation unnecessary downtime in case of a false alarm.

In view of the above observations, it can be deduced that there is a great need for an automatic and effective identification and interpretation methodology of the control chart patterns, which will necessarily indicate the state of the manufacturing process. Neural networks, one of the Artificial Intelligence (AI) paradigms, can be handily used for this purpose, because of their ability to imitate the skill of experts by capturing the knowledge implicitly contained in the examples taken from the process. Neural networks store this knowledge which can be used in understanding the process behaviour. For example a synthetic backpropagation network to analyse process data. The input nodes correspond to observations in a subgroup and a single output represents the status of a process mean (shifted or unshifted). The network was trained with shifted and unshifted data. It was shown that the performance of the neural network is comparable to that of control charts. There are two limiting features in this approach. First, the network size is highly dependent on the subgroup size. Second, since the network analyses one subgroup at a time, the Type I error (concluding that the process is out of control when it is really in control) will be very high if the training includes small magnitudes of shifts.

Even though many research efforts were directed at the application of neural networks for control chart pattern identification and interpretation, pertinent issues which are not addressed so far are the following.

(i) A unique neural network model to detect all possible trends and variations in the process

(ii) Models for detecting small shifts in process states.

(iii) Consideration of a small number of subgroups in analyzing the data.

The present work deals with the identification and interpretation of the control chart patterns (which necessarily reflects the state of the manufacturing process) through neural network approach, which models the knowledge and judgement needed in understanding the process behaviour. The objective is to recognize both small and large deviations and also to identify the nature of process change to help take proper corrective actions to remedy the problem. The next section describes the outline of the backpropagation neural network with learning rate adaptation.

A process is generally defined as a combination of people, machines, and other equipment, raw materials, methods, and environment that produces products as planned. In any process, regardless of how well the process is designed and maintained, a certain amount of inherent or natural variability may occur. In Shewhart control charts, two causes of process variations, chance (unassignable) causes and special (assignable) causes may be seen . When chance causes, which are inevitable, difficult to detect or identify, are in affect, a process is considered to be in a state of statistical control. On the other hand, even if a process is in control, variations due to machine and operator performances and characteristics of incoming materials or other causes may occur within a stable and predictable process.

In general, a process is said to be in control, if all data points measured for a selected quality characteristic of a product are within the control limits established by the inherent variation in a process. However, fluctuations or unnatural behaviors due to special causes may exist even if all data points are within the control limits. When data points show unnatural behaviors (also called special patterns), a process is said to be out of control or beyond the expected nature of variation because of excessive variation. If unnatural patterns are observed, special causes responsible for the condition must be found and the necessary corrective actions need to be taken in order to eliminate these disturbances . Each special pattern, such as trend, sudden shifts, occurs due to different circumstances. The occurrence of a trend or recurring cycles, for example, may indicate special causes, such as operator fatigue, tool wear, different incoming materials, voltage fluctuations, or systematic adjustment of the process. Therefore, the necessary corrective actions should be taken to eliminate the special causes appeared.

4.2 Neural Network Model used in CCP identification :

1. An ***Artificial Neural Network (ANN)*** model having **2 layers** is used.
2. The **input layer** has **4 Nodes** for the sample size of 4 inputs
3. The **output layer** has **3 Nodes** for the 3 types of shifts – **Mean shift, Sigma shift** or **No shifts**.
4. The **hidden layer** was fixed with **4** number of Nodes.
5. The model was trained with 75% of the normally distributed random data i.e., 750 samples of input data.
6. Testing was performed with the rest 25% data – 250 sample set.
7. Initial Learning Rate was 0.5%
8. Momentum Rate was 0.1%
9. The whole ANN model for CCP identification was realized using Back-Propagation (BPN) algorithm with the standard Delta rule implemented in it.

4.3 DATA GENERATION AND REPRESENTATION :

Any quantity whose variation depends on random causes is distributed according to the normal Law

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left((x-\mu)^2/2\sigma^2\right)}$$

where p and c are the mean and standard deviation, respectively. The neural network should be trained across a wide spectrum of possible change magnitudes in order to yield satisfactory performance over the entire range of process changes and the network must be trained with sufficient examples in order to achieve good generalization. Without loss of generality, the incontrol data generated is normally distributed with a mean (machine setting) of p and standard deviation of u (calculated from process capability of the process). The out-of-control data corresponding to the changes in mean and standard deviation are normally distributed with mean changed from ,u and standard deviation changed from 0, respectively.

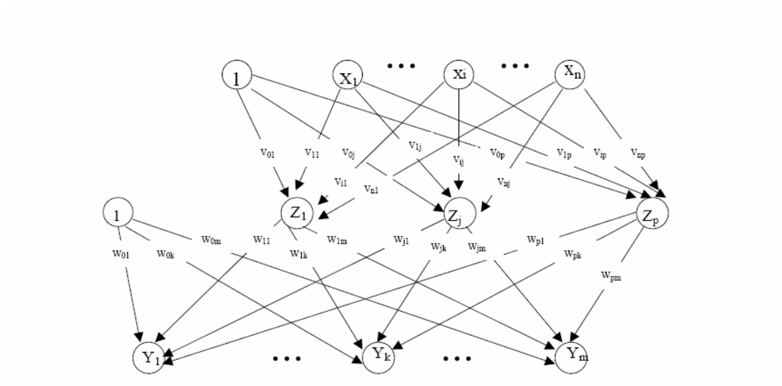
The data used in this study were generated using a computer. The data was generated according to normal distribution as data from manufacturing is assumed to follow normal distribution. 1000 number of samples each having 4 data points for the 4 input nodes were generated., to be able to consider different types of special patterns, which might appear on control charts, were produced and saved into files.

The saved files consisted of a number of samples for mean shifts, sigma shifts and no shifts each having their inherent patterns.

The samples were presented to the neural network in the form of data points both for training (75%) and testing (25%) purposes.

4.4 <u>NEURAL NETWORK MODEL ARCHITECTURE</u> :

The neural network used in this study was composed of three layers. Each layer was connected to the upper layer via randomly generated real values (weights). A sigmoidal function was used to determine the new activation values of the neurons. The input layer was made up of 4 neurons, represented patterns. There were three output neurons.



The ANN Model used for the CCP identification has :

1. 4 input nodes
2. 1 hidden layer having 4 nodes
3. 3 output nodes

The model was trained with BPN algorithm with the following parameters :

1. Learning rate = 0.5
2. Momentum rate = 0.1

All samples were assumed to conform to rational subgrouping. Training sets of 750 samples were created with the following components:

250 Normally distributed random data with μ = 10 and σ = 0.1 (No shifts or In Control)
250 Normally distributed random data with μ = 11 and σ = 0.1 (Mean shift)
250 Normallly distributed random data with μ = 10 and σ = 0.2 (Variance shift)

After training another random set of 250 normally distributed data samples were used to test the ANN model for CCP identification.

4.5 <u>TRAINING & TESTING RESULTS</u> :

Confusion Matrix :

[PATTERN]

	ACTUAL				
Prediction	MEAN SHIFT	NO SHIFTS	SIGMA SHIFT	Prediction Totals	Prediction Error%
MEAN SHIFT	83	0	0	83	0.00%
NO SHIFTS	0	83	1	84	1.19%
SIGMA SHIFT	0	0	82	82	0.00%
Actual Totals	83	83	83	249	0.40%
Actual Error%	0.00%	0.00%	1.20%	0.40%	

The confusion matrix is a table summarizing the tendency of the recognizer to classify a recognized pattern into a correct class or into any of the other seven possible (wrong) classes. Confusion matrices, as given in provide the overall mean percentages for confusions among the pattern classes

For a large shift in mean or variance, the neural networks performed comparably to a standard control chart, however when the shift was more subtle, and the extra information of the raw data itself was provided, the neural networks made slightly more than half the errors of the control charts

The type of errors of each formulation was also different. In our experiments the manual control chart made primarily Type II Errors, that is missed disturbances. The neural network made both Type I and Type II Errors. This improvement of the neural networks' Type II Error rate relative to the manual control charts'. The networks were trained with the assumption that the penalties for Type I and Type II Errors were identical, and therefore made both kinds of errors in approximately equal proportions. This could easily be remedied to reflect different relative penalties for Type I and Type II Errors by shifting the interpretive schedule for an out of control decision, that is moving the decision boundary further from the class with the lower relative error penalty. The errors near decision borders could be analyzed more closely by human experts or with neural prediction of points, to achieve even better precision.

The actual data used for testing of the ANN model for CPP identification is given in the tables :

PATTERN	Predicted	1	2	3	4	IndexCounter1
NO SHIFTS	NO SHIFTS	9.92	9.945	10.065	9.899	1
NO SHIFTS	NO SHIFTS	9.977	9.976	9.951	10.081	2
NO SHIFTS	NO SHIFTS	9.878	10.139	10.271	10.01	3
NO SHIFTS	NO SHIFTS	9.89	9.974	10.025	10.129	4
NO SHIFTS	NO SHIFTS	9.987	9.984	10.134	9.914	5
NO SHIFTS	NO SHIFTS	10.014	10.123	10.221	10.086	6
NO SHIFTS	NO SHIFTS	9.992	10.079	9.974	9.968	7
NO SHIFTS	NO SHIFTS	10.005	10.019	9.858	9.963	8
NO SHIFTS	NO SHIFTS	10.111	10.12	10.087	10.092	9
NO SHIFTS	NO SHIFTS	10.131	10.001	10.15	10.081	10
NO SHIFTS	NO SHIFTS	9.996	10.077	10.08	10.051	11
NO SHIFTS	NO SHIFTS	9.996	9.906	10.023	9.843	12
NO SHIFTS	NO SHIFTS	10.065	9.954	9.85	9.834	13
NO SHIFTS	NO SHIFTS	10.279	10.319	9.839	10.025	14
NO SHIFTS	NO SHIFTS	10.076	9.846	9.7	10.052	15
NO SHIFTS	NO SHIFTS	9.818	10.027	10.137	9.919	16
NO SHIFTS	NO SHIFTS	10.194	10.053	9.982	10.142	17
NO SHIFTS	NO SHIFTS	10.043	10.051	10.107	9.993	18
NO SHIFTS	NO SHIFTS	10.048	9.97	9.979	9.917	19
NO SHIFTS	NO SHIFTS	9.966	10.026	10.013	9.817	20
NO SHIFTS	NO SHIFTS	10.041	10.052	9.962	10.113	21
NO SHIFTS	NO SHIFTS	9.946	10.158	9.828	10.039	22
NO SHIFTS	NO SHIFTS	9.893	9.866	9.968	10.043	23
NO SHIFTS	NO SHIFTS	9.869	10.167	9.938	9.915	24
NO SHIFTS	NO SHIFTS	10.033	10.182	10.004	10.103	25
NO SHIFTS	NO SHIFTS	10.112	10.028	9.971	9.886	26
NO SHIFTS	NO SHIFTS	9.942	10.031	10.03	10.068	27
NO SHIFTS	NO SHIFTS	9.915	10.036	9.864	9.985	28
NO SHIFTS	NO SHIFTS	10.094	9.98	10.05	10.097	29
MEAN SHIFT	MEAN SHIFT	10.956	10.951	10.939	11.041	30
MEAN SHIFT	MEAN SHIFT	10.862	10.914	10.873	11.074	31
MEAN SHIFT	MEAN SHIFT	10.868	10.87	11.077	11.007	32
MEAN SHIFT	MEAN SHIFT	11.012	10.952	10.898	11.124	33
MEAN SHIFT	MEAN SHIFT	10.979	10.778	11.038	11.034	34
MEAN SHIFT	MEAN SHIFT	10.832	11	10.942	10.906	35
MEAN SHIFT	MEAN SHIFT	11.069	10.986	11.049	10.949	36
MEAN SHIFT	MEAN SHIFT	10.986	10.924	11.029	10.916	37
MEAN SHIFT	MEAN SHIFT	11.14	11.015	10.914	11.028	38
MEAN SHIFT	MEAN SHIFT	11.012	11.025	10.9	11.279	39
MEAN SHIFT	MEAN SHIFT	10.848	10.78	11.006	10.867	40
MEAN SHIFT	MEAN SHIFT	11.29	10.882	10.926	10.971	41
MEAN SHIFT	MEAN SHIFT	10.936	11.026	11.207	11.035	42
MEAN SHIFT	MEAN SHIFT	11.002	11.031	11.15	11.005	43
MEAN SHIFT	MEAN SHIFT	11.098	10.73	10.934	11.148	44
MEAN SHIFT	MEAN SHIFT	10.865	10.793	10.951	10.995	45
MEAN SHIFT	MEAN SHIFT	10.916	11.136	10.92	11.103	46
MEAN SHIFT	MEAN SHIFT	10.944	11.155	11.032	11.131	47
MEAN SHIFT	MEAN SHIFT	11.009	11.039	10.942	10.949	48
MEAN SHIFT	MEAN SHIFT	10.913	11.048	11.061	10.961	49
MEAN SHIFT	MEAN SHIFT	11.082	11.188	11.036	10.992	50
MEAN SHIFT	MEAN SHIFT	11.13	10.935	11.059	10.934	51
MEAN SHIFT	MEAN SHIFT	11.136	10.864	11.089	11	52
MEAN SHIFT	MEAN SHIFT	10.964	10.912	11.066	11.084	53
MEAN SHIFT	MEAN SHIFT	10.992	10.865	10.806	10.887	54
MEAN SHIFT	MEAN SHIFT	11.119	10.994	11.011	10.905	55
MEAN SHIFT	MEAN SHIFT	10.92	10.945	11.065	10.899	56
SIGMA SHIFT	SIGMA SHIFT	9.996	9.837	9.973	10.128	57
SIGMA SHIFT	SIGMA SHIFT	9.911	9.901	9.879	10.083	58
SIGMA SHIFT	SIGMA SHIFT	9.724	9.829	9.747	10.149	59
SIGMA SHIFT	SIGMA SHIFT	9.735	9.741	10.153	10.015	60
SIGMA SHIFT	SIGMA SHIFT	10.023	9.904	9.797	10.248	61
SIGMA SHIFT	SIGMA SHIFT	9.959	9.557	10.076	10.068	62
SIGMA SHIFT	SIGMA SHIFT	9.663	10	9.883	9.811	63
SIGMA SHIFT	SIGMA SHIFT	10.137	9.972	10.097	9.898	64
SIGMA SHIFT	SIGMA SHIFT	9.972	9.848	10.059	9.833	65
SIGMA SHIFT	SIGMA SHIFT	10.28	10.03	9.828	10.056	66
SIGMA SHIFT	SIGMA SHIFT	10.023	10.051	9.799	10.557	67
SIGMA SHIFT	SIGMA SHIFT	9.695	9.56	10.012	9.733	68
SIGMA SHIFT	SIGMA SHIFT	10.58	9.763	9.853	9.942	69
SIGMA SHIFT	SIGMA SHIFT	9.871	10.052	10.413	10.07	70
SIGMA SHIFT	SIGMA SHIFT	10.005	10.062	10.299	10.01	71
SIGMA SHIFT	SIGMA SHIFT	10.196	9.46	9.868	10.296	72
SIGMA SHIFT	SIGMA SHIFT	9.731	9.586	9.903	9.99	73
SIGMA SHIFT	SIGMA SHIFT	9.832	10.272	9.839	10.206	74
SIGMA SHIFT	SIGMA SHIFT	9.888	10.31	10.064	10.262	75
SIGMA SHIFT	SIGMA SHIFT	10.018	10.078	9.884	9.898	76
SIGMA SHIFT	SIGMA SHIFT	9.826	10.095	10.123	9.921	77
SIGMA SHIFT	SIGMA SHIFT	10.165	10.375	10.072	9.984	78

PATTERN	Predicted	1	2	3	4	IndexCounter1
SIGMA SHIFT	SIGMA SHIFT	10.259	9.87	10.117	9.868	79
SIGMA SHIFT	SIGMA SHIFT	10.273	9.727	10.178	10	80
SIGMA SHIFT	SIGMA SHIFT	9.929	9.824	10.132	10.167	81
SIGMA SHIFT	SIGMA SHIFT	9.985	9.73	9.613	9.774	82
SIGMA SHIFT	SIGMA SHIFT	10.238	9.989	10.021	9.81	83
MEAN SHIFT	MEAN SHIFT	10.977	10.976	10.951	11.081	84
MEAN SHIFT	MEAN SHIFT	10.878	11.139	11.271	11.01	85
MEAN SHIFT	MEAN SHIFT	10.89	10.974	11.025	11.129	86
MEAN SHIFT	MEAN SHIFT	10.987	10.984	11.134	10.914	87
MEAN SHIFT	MEAN SHIFT	11.014	11.123	11.221	11.086	88
MEAN SHIFT	MEAN SHIFT	10.992	11.079	10.974	10.968	89
MEAN SHIFT	MEAN SHIFT	11.005	11.019	10.858	10.963	90
MEAN SHIFT	MEAN SHIFT	11.111	11.12	11.087	11.092	91
MEAN SHIFT	MEAN SHIFT	11.131	11.001	11.15	11.081	92
MEAN SHIFT	MEAN SHIFT	10.996	11.077	11.08	11.051	93
MEAN SHIFT	MEAN SHIFT	10.996	10.906	11.023	10.843	94
MEAN SHIFT	MEAN SHIFT	11.065	10.954	10.85	10.834	95
MEAN SHIFT	MEAN SHIFT	11.279	11.319	10.839	11.025	96
MEAN SHIFT	MEAN SHIFT	11.076	10.846	10.7	11.052	97
MEAN SHIFT	MEAN SHIFT	10.818	11.027	11.137	10.919	98
MEAN SHIFT	MEAN SHIFT	11.194	11.053	10.982	11.142	99
MEAN SHIFT	MEAN SHIFT	11.043	11.051	11.107	10.993	100
MEAN SHIFT	MEAN SHIFT	11.048	10.97	10.979	10.917	101
MEAN SHIFT	MEAN SHIFT	10.966	11.026	11.013	10.817	102
MEAN SHIFT	MEAN SHIFT	11.041	11.052	10.962	11.113	103
MEAN SHIFT	MEAN SHIFT	10.946	11.158	10.828	11.039	104
MEAN SHIFT	MEAN SHIFT	10.893	10.866	10.968	11.043	105
MEAN SHIFT	MEAN SHIFT	10.869	11.167	10.938	10.915	106
MEAN SHIFT	MEAN SHIFT	11.033	11.182	11.004	11.103	107
MEAN SHIFT	MEAN SHIFT	11.112	11.028	10.971	10.886	108
MEAN SHIFT	MEAN SHIFT	10.942	11.031	11.03	11.068	109
MEAN SHIFT	MEAN SHIFT	10.915	11.036	10.864	10.985	110
MEAN SHIFT	MEAN SHIFT	11.094	10.98	11.05	11.097	111
MEAN SHIFT	MEAN SHIFT	10.947	11.006	10.928	11.037	112
NO SHIFTS	NO SHIFTS	9.998	9.918	9.987	10.064	113
NO SHIFTS	NO SHIFTS	9.956	9.951	9.939	10.041	114
NO SHIFTS	NO SHIFTS	9.862	9.914	9.873	10.074	115
NO SHIFTS	NO SHIFTS	9.868	9.87	10.077	10.007	116
NO SHIFTS	NO SHIFTS	10.012	9.952	9.898	10.124	117
PATTERN	Predicted	1	2	3	4	IndexCounter1
NO SHIFTS	NO SHIFTS	9.979	9.778	10.038	10.034	118
NO SHIFTS	NO SHIFTS	9.832	10	9.942	9.906	119
NO SHIFTS	NO SHIFTS	10.069	9.986	10.049	9.949	120
NO SHIFTS	NO SHIFTS	9.986	9.924	10.029	9.916	121
NO SHIFTS	NO SHIFTS	10.14	10.015	9.914	10.028	122
NO SHIFTS	NO SHIFTS	10.012	10.025	9.9	10.279	123
NO SHIFTS	NO SHIFTS	9.848	9.78	10.006	9.867	124
NO SHIFTS	NO SHIFTS	10.29	9.882	9.926	9.971	125
NO SHIFTS	NO SHIFTS	9.936	10.026	10.207	10.035	126
NO SHIFTS	NO SHIFTS	10.002	10.031	10.15	10.005	127
NO SHIFTS	NO SHIFTS	10.098	9.73	9.934	10.148	128
NO SHIFTS	NO SHIFTS	9.865	9.793	9.951	9.995	129
NO SHIFTS	NO SHIFTS	9.916	10.136	9.92	10.103	130
NO SHIFTS	NO SHIFTS	9.944	10.155	10.032	10.131	131
NO SHIFTS	NO SHIFTS	10.009	10.039	9.942	9.949	132
NO SHIFTS	NO SHIFTS	9.913	10.048	10.061	9.961	133
NO SHIFTS	NO SHIFTS	10.082	10.188	10.036	9.992	134
NO SHIFTS	NO SHIFTS	10.13	9.935	10.059	9.934	135
NO SHIFTS	NO SHIFTS	10.136	9.864	10.089	10	136
NO SHIFTS	NO SHIFTS	9.964	9.912	10.066	10.084	137
NO SHIFTS	NO SHIFTS	9.992	9.865	9.806	9.887	138
NO SHIFTS	NO SHIFTS	10.119	9.994	10.011	9.905	139
SIGMA SHIFT	SIGMA SHIFT	9.895	10.012	9.857	10.074	140
SIGMA SHIFT	SIGMA SHIFT	10.333	9.856	9.668	9.597	141
SIGMA SHIFT	SIGMA SHIFT	9.874	10.039	10.084	10.102	142
SIGMA SHIFT	SIGMA SHIFT	10.159	9.956	9.754	10.079	143
SIGMA SHIFT	SIGMA SHIFT	10.151	10.196	9.896	10.538	144
SIGMA SHIFT	SIGMA SHIFT	10.102	9.616	10.043	9.968	145
SIGMA SHIFT	SIGMA SHIFT	9.968	10.034	9.933	9.92	146
SIGMA SHIFT	SIGMA SHIFT	9.973	10.153	9.843	9.917	147
SIGMA SHIFT	SIGMA SHIFT	9.694	9.868	10.452	10.17	148
SIGMA SHIFT	SIGMA SHIFT	10.073	10.215	10.033	10.032	149
SIGMA SHIFT	SIGMA SHIFT	10.194	9.647	9.842	10.013	150
SIGMA SHIFT	SIGMA SHIFT	9.702	9.98	9.863	9.84	151
SIGMA SHIFT	SIGMA SHIFT	10.191	9.823	9.993	10.254	152
SIGMA SHIFT	SIGMA SHIFT	10.292	10.02	10.042	9.727	153
SIGMA SHIFT	SIGMA SHIFT	9.86	10.232	10.06	10.123	154
SIGMA SHIFT	SIGMA SHIFT	10.276	10.126	9.987	9.793	155
SIGMA SHIFT	SIGMA SHIFT	9.665	10.152	10.27	9.736	156

PATTERN	Predicted	1	2	3	4	IndexCounter1
SIGMA SHIFT	SIGMA SHIFT	10.077	9.865	9.955	10.137	157
SIGMA SHIFT	SIGMA SHIFT	9.802	9.758	10.121	9.944	158
SIGMA SHIFT	SIGMA SHIFT	9.778	10.127	10.102	9.96	159
SIGMA SHIFT	SIGMA SHIFT	9.783	9.567	10.441	9.84	160
SIGMA SHIFT	SIGMA SHIFT	9.821	10.283	10.238	10.354	161
SIGMA SHIFT	SIGMA SHIFT	9.993	9.864	9.938	9.95	162
SIGMA SHIFT	SIGMA SHIFT	10.073	10.072	10.113	10.076	163
SIGMA SHIFT	SIGMA SHIFT	9.735	10.018	10.014	10.029	164
SIGMA SHIFT	SIGMA SHIFT	10.406	10.293	10.072	10.128	165
SIGMA SHIFT	SIGMA SHIFT	9.62	9.927	10.323	10.157	166
SIGMA SHIFT	SIGMA SHIFT	9.839	9.891	10.129	9.797	167
SIGMA SHIFT	SIGMA SHIFT	9.953	9.953	9.903	10.162	168
SIGMA SHIFT	SIGMA SHIFT	9.757	10.278	10.543	10.02	169
SIGMA SHIFT	SIGMA SHIFT	9.78	9.949	10.051	10.259	170
SIGMA SHIFT	SIGMA SHIFT	9.974	9.968	10.268	9.828	171
SIGMA SHIFT	SIGMA SHIFT	10.028	10.245	10.442	10.172	172
SIGMA SHIFT	SIGMA SHIFT	9.983	10.158	9.947	9.936	173
SIGMA SHIFT	SIGMA SHIFT	10.009	10.037	9.716	9.926	174
SIGMA SHIFT	SIGMA SHIFT	10.221	10.24	10.174	10.184	175
SIGMA SHIFT	SIGMA SHIFT	10.262	10.001	10.301	10.162	176
SIGMA SHIFT	SIGMA SHIFT	9.992	10.155	10.159	10.101	177
SIGMA SHIFT	SIGMA SHIFT	9.991	9.811	10.045	9.687	178
SIGMA SHIFT	SIGMA SHIFT	10.13	9.908	9.699	9.668	179
SIGMA SHIFT	SIGMA SHIFT	10.558	10.637	9.678	10.049	180
SIGMA SHIFT	SIGMA SHIFT	10.151	9.693	9.4	10.104	181
SIGMA SHIFT	SIGMA SHIFT	9.637	10.054	10.275	9.838	182
SIGMA SHIFT	SIGMA SHIFT	10.389	10.107	9.964	10.285	183
SIGMA SHIFT	SIGMA SHIFT	10.087	10.103	10.215	9.986	184
SIGMA SHIFT	SIGMA SHIFT	10.096	9.94	9.958	9.833	185
SIGMA SHIFT	SIGMA SHIFT	9.932	10.053	10.025	9.634	186
SIGMA SHIFT	SIGMA SHIFT	10.083	10.105	9.925	10.226	187
SIGMA SHIFT	SIGMA SHIFT	9.891	10.315	9.656	10.079	188
SIGMA SHIFT	SIGMA SHIFT	9.785	9.731	9.936	10.086	189
SIGMA SHIFT	SIGMA SHIFT	9.737	10.334	9.876	9.83	190
SIGMA SHIFT	SIGMA SHIFT	10.066	10.364	10.009	10.206	191
SIGMA SHIFT	SIGMA SHIFT	10.224	10.056	9.942	9.772	192
SIGMA SHIFT	SIGMA SHIFT	9.885	10.061	10.06	10.137	193
SIGMA SHIFT	NO SHIFTS	9.831	10.071	9.728	9.969	194
SIGMA SHIFT	SIGMA SHIFT	10.187	9.96	10.099	10.195	195

PATTERN	Predicted	1	2	3	4	IndexCounter1
MEAN SHIFT	MEAN SHIFT	11.166	10.928	10.834	10.798	196
MEAN SHIFT	MEAN SHIFT	10.937	11.02	11.042	11.051	197
MEAN SHIFT	MEAN SHIFT	11.079	10.978	10.877	11.04	198
MEAN SHIFT	MEAN SHIFT	11.076	11.098	10.948	11.269	199
MEAN SHIFT	MEAN SHIFT	11.051	10.808	11.022	10.984	200
MEAN SHIFT	MEAN SHIFT	10.984	11.017	10.966	10.96	201
MEAN SHIFT	MEAN SHIFT	10.986	11.076	10.922	10.958	202
MEAN SHIFT	MEAN SHIFT	10.847	10.934	11.226	11.085	203
MEAN SHIFT	MEAN SHIFT	11.037	11.108	11.017	11.016	204
MEAN SHIFT	MEAN SHIFT	11.097	10.823	10.921	11.006	205
MEAN SHIFT	MEAN SHIFT	10.851	10.99	10.932	10.92	206
MEAN SHIFT	MEAN SHIFT	11.096	10.911	10.997	11.127	207
MEAN SHIFT	MEAN SHIFT	11.146	11.01	11.021	10.863	208
MEAN SHIFT	MEAN SHIFT	10.93	11.116	11.03	11.062	209
MEAN SHIFT	MEAN SHIFT	11.138	11.063	10.994	10.897	210
MEAN SHIFT	MEAN SHIFT	10.833	11.076	11.135	10.868	211
MEAN SHIFT	MEAN SHIFT	11.039	10.932	10.977	11.069	212
MEAN SHIFT	MEAN SHIFT	10.901	10.879	11.06	10.972	213
MEAN SHIFT	MEAN SHIFT	10.889	11.063	11.051	10.98	214
MEAN SHIFT	MEAN SHIFT	10.892	10.784	11.221	10.92	215
MEAN SHIFT	MEAN SHIFT	10.911	11.142	11.119	11.177	216
MEAN SHIFT	MEAN SHIFT	10.997	10.932	10.969	10.975	217
MEAN SHIFT	MEAN SHIFT	11.037	11.036	11.057	11.038	218
MEAN SHIFT	MEAN SHIFT	10.867	11.009	11.007	11.014	219
MEAN SHIFT	MEAN SHIFT	11.203	11.146	11.036	11.064	220
MEAN SHIFT	MEAN SHIFT	10.81	10.963	11.161	11.078	221
MEAN SHIFT	MEAN SHIFT	10.998	10.918	10.987	11.064	222
NO SHIFTS	NO SHIFTS	9.947	10.006	9.928	10.037	223
NO SHIFTS	NO SHIFTS	10.166	9.928	9.834	9.798	224
NO SHIFTS	NO SHIFTS	9.937	10.02	10.042	10.051	225
NO SHIFTS	NO SHIFTS	10.079	9.978	9.877	10.04	226
NO SHIFTS	NO SHIFTS	10.076	10.098	9.948	10.269	227
NO SHIFTS	NO SHIFTS	10.051	9.808	10.022	9.984	228
NO SHIFTS	NO SHIFTS	9.984	10.017	9.966	9.96	229
NO SHIFTS	NO SHIFTS	9.986	10.076	9.922	9.958	230
NO SHIFTS	NO SHIFTS	9.847	9.934	10.226	10.085	231
NO SHIFTS	NO SHIFTS	10.037	10.108	10.017	10.016	232
NO SHIFTS	NO SHIFTS	10.097	9.823	9.921	10.006	233
NO SHIFTS	NO SHIFTS	9.851	9.99	9.932	9.92	234

NO SHIFTS	NO SHIFTS	10.096	9.911	9.997	10.127	235
NO SHIFTS	NO SHIFTS	10.146	10.01	10.021	9.863	236
NO SHIFTS	NO SHIFTS	9.93	10.116	10.03	10.062	237
NO SHIFTS	NO SHIFTS	10.138	10.063	9.994	9.897	238
NO SHIFTS	NO SHIFTS	9.833	10.076	10.135	9.868	239
NO SHIFTS	NO SHIFTS	10.039	9.932	9.977	10.069	240
NO SHIFTS	NO SHIFTS	9.901	9.879	10.06	9.972	241
NO SHIFTS	NO SHIFTS	9.889	10.063	10.051	9.98	242
NO SHIFTS	NO SHIFTS	9.892	9.784	10.221	9.92	243
NO SHIFTS	NO SHIFTS	9.911	10.142	10.119	10.177	244
NO SHIFTS	NO SHIFTS	9.997	9.932	9.969	9.975	245
NO SHIFTS	NO SHIFTS	10.037	10.036	10.057	10.038	246
NO SHIFTS	NO SHIFTS	9.867	10.009	10.007	10.014	247
NO SHIFTS	NO SHIFTS	10.203	10.146	10.036	10.064	248
NO SHIFTS	NO SHIFTS	9.81	9.963	10.161	10.078	249

Discussion of results :

Thus after training the ANN model with 750 samples of normal random sets of data the model has become quite good at pattern recognition with only 1 mistaken prediction which amounts to 0.4% overall actual error .
Although the training process takes considerable computer time, the recall process is very fast.

In order to improve applicability of this approach, more patterns should be included in training. Also, even though it is a tedious process, some research should be done to determine the optimal values of the parameters, such as learning rate, momentum term, number of hidden layers and neurons, which affect the performance of the neural network. The data used in this study may be applied to different neural networks and pattern recognition techniques and results obtained from them should be compared with each other to determine a reliable way of recognizing and interpreting patterns, if applicable.

Chapter 5

CONCLUSIONS

Identification and interpretation of manufacturing process patterns is an important aspect of diagnosis in a quality improvement environment. Given the occurrence of a particular event, the diagnostic search can be reduced in length of time if one has knowledge of whether the process change is in the mean (such as shift, trend, cycles, etc.) or in variability (such as shift in process variability). This identification and interpretation combined with the corresponding knowledge . of the process factors that affect mean and variability of the output, can help in correcting the manufacturing process very early in its erratic behavior and save lot of time, wastage and effort for the organisation. The neural network methodology presented in this work is used to solve this statistical classification problem by identifying the shift in the process mean or variance and also the various patterns such as trends, whenever they are present. It is seen that, the network is quite effective in detecting small process changes when compared with control charts. Also these changes are correctly identified from a small sample, which helps in investigating the process expediently.

The simulated examples discussed in this paper show that the neural network developed in the present work is a generalised pattern classification program to suit manufacturing needs and is a good control procedure for detecting small changes in the process mean as well. Early identification of the patterns, such as trends and shifts, can provide valuable information for real-time process control. In a computer integrated manufacturing environment, the proposed methodology could be used to send signals when unwanted patterns are identified which signal an out-of-control condition, to facilitate the prevention of nonconforming products from being produced, and also to take corrective action immediately.

Abnormal control chart patterns can provide clues to reveal potential quality problems at an early stage, so as to eliminate the defects before they are produced. Owing to the interference, it is not efficient to recognize control chart patterns, especially mixed patterns, by traditional techniques, such as simple heuristics or control limits.

This paper has demonstrated that neural networks can be comparable to Shewhart control charts for large shifts in mean or variance, and can out perform them for small shifts. Neural networks work best when they have benefit of both raw sample data and sample statistics, although the statistics themselves are adequate to detect large shifts. A significant benefit of the neural approach is that a single network can model multiple control strategies simultaneously.

For shape interpretation and prediction, networks performed best with minimal noise and maximum number of inputs. All neural networks proved capable of good quality decisions regarding pattern identification even in light of sparse and noisy data. The predictive networks could simultaneously predict out to five sample increments without much lost of accuracy.

Since the network training can be implemented off-line, this feature facilities the use of the proposed model in an on-line real time mode. There are several directions for future research:

1. Many factors, e.g**. learning rate and momentum factor**, etc., may affect the results of neural networks training. In the literature, the best combination of these factors was generally obtained by experiments. It is possible to use experimental design approach to analyze the relationships of these factors.
2. BPN, although it has good performance in pattern recognition of control charts, has a weak point by nature. That is, it has **no `memory' ability**. Once a new pattern set appears, it has to be trained again by both the old training sets and the new ones. Hence, one may consider other neural networks which possess memory, e.g. adaptive resonance theory network, to improve the model adaptation ability in the changing environment.
3. It is possible to combine the proposed model with other neural networks or **expert systems** which are to inference the relevant causes of the variations and then facilitate the automatic quality control.
4. In this research, the proposed model has only been tested by the independent samples which already contain the full abnormal patterns. In practice, one may consider a neural network model with moving range sampling. For such a sampling practice, several issues, e.g. the initial state and ARL (average run length to identify an abnormal pattern), are to be studied further.

Given the widespread use of control charts in both manufacturing and service industries, and the current difficulties with proper interpretation of plotted results, the promising results of a neural computing approach bear further research. Since neural networks are commonly available in software form running on PC and workstation platforms, and will soon be practical in VLSI format, they are viable options for statistical control in production environments.

REFERENCES :

Alexander, S. M. and V. Jagannathan, 1986, Advisory system for control chart. *Computers & Industrial Engineering,* **10**, 171-177.

Andersen, Kristinn, George E. Cook, Gabor Karsai and Kumar Ramaswamy, 1990, Artificial neural networks applied to arc welding process modeling and control. *IEEE Transactions on Industry Applications,* **26**, 824-830.

Burke, Laura I., 1989, *Automated Identification of Tool Wear States in Machining Processes: An Application of Self-Organizing Neural Network.* Ph.D. Thesis, University of California - Berkeley.

Dagli, Cihan H. and Alice E. Smith, 1991, A prototype quality control expert system integrated with an optimization module. *Proceedings of the World Congress on Expert Systems,* 1959-1966.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth and Brooks/Cole Advanced Books and Software.

Cheng, C. S. (1997). A neural network approach for the analysis of control chart patterns. International Journal of Production Research, 35, 667–697.

Evans, J. R., & Lindsay, W. M. (1988). A framework for expert system development in statistical quality control. Computers and Industrial Engineering, 14(3), 335–343.

Grant, G. E., & Leavenworth, R. S. (1996). Statistical quality control (7th ed.). New York: McGraw-Hill.

Guh, R. S., & Tannock, J. D. T. (1999). Recognition of control chart concurrent patterns using a neural network approach. International Journal of Production Research, 37(8), 1743–1765.

Guh, R. S., Zorriassatine, F., Tannock, J. D. T., & O'Brien, C. (1999a). On-line control chart pattern detection and discrimination – a neural network approach. Artificial Intelligence in Engineering, 13, 413–425.

Guh, R. S., Tannock, J. D. T., & O'Brien, C. (1999b). IntelliSPC: a hybrid intelligent tool for on-line economical statistical process control. Expert Systems with Applications, 17, 195–212.

Guh, R. S., & Shiue, Y. R. (2005). On-line identification of control chart patterns using self-organizing approaches. International Journal of Production Research, 43, 1225–1254.

Hwarng, H. B., & Hubele, N. F. (1993). Back-propagation pattern recognisers for X control charts: methodology and performance. Computers and Industrial Engineering, 24(2), 219–235.

Haykin, S. (1999). Neural network: a comprehensive foundation (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

SUMMARY OF THE PROJECT WORK DONE ON -

IDENTIFICATION OF CONTROL CHART PATTERNS USING NEURAL NETWORKS

ABSTRACT

Artificial neural networks trained by supervised techniques have been documented as good alternatives for pattern classification and prediction. These skills can be put to use for the interpretation of process control from input of control chart samples. Through learning of varying input conditions matched with control status, a network can decide on control status when faced with new inputs. Besides exceeding control boundaries, control chart points can provide information about the long term condition of the process through symptomatic shapes, runs and drifts. If these can be correctly identified from a small sample by neural networks, then the process can be investigated expediently. Neural networks can also forecast future control chart point(s), thus contributing to the diagnosis of process condition in borderline conditions.

INTRODUCTION

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse.

ANN MODEL USED

An Artificial Neural Network (ANN) model having 3 layers is used. The input layer has 4 Nodes for the sample size of 4 inputs. The output layer has 3 Nodes for the 3 types of shifts – Mean shift, Sigma shift or No shifts. The **hidden layer** was fixed with **4** number of Nodes. The model was trained with 75% of the normally distributed random data i.e., 750 samples of input data. Testing was performed with the rest 25% data – 250 sample set. Initial Learning Rate was 0.5%.Momentum Rate was 0.1%.The whole ANN model for CCP identification was realized using Back-Propagation (BPN) algorithm with the standard Delta rule implemented in it. Each layer was connected to the upper layer via randomly generated real values (weights). A sigmoidal function was used to determine the new activation values of the neurons.

RESULT

Thus after training the ANN model with 750 samples of normal random sets of data the model has become quite good at pattern recognition with only 1 mistaken prediction which amounts to 0.4% overall actual error .Although the training process takes considerable computer time, the recall process is very fast

EFFECTIVENESS OF USING NEURAL NETWORKS IN PROCESS CONTROL

In general control charts do not solve problems, but provide information on the basis of which we can reasonably (in a statistical sense) presume that process is “in control” or “out of control”. Actual analysis of the control chart is done manually and this is somewhat difficult. Proper analysis not only requires thorough knowledge and understanding of the underlying distribution theories associated with control charts, but the experience that only an expert can provide as well. While using control charts, the judgement is usually left to the user. Another major limitation of the control charts is that they only use the information about the process contained in the last observation to decide about the state of the process. This feature makes the control chart relatively insensitive to small shifts in the process. Furthermore, a point falling out of control limits does not necessarily indicate an out-of-control situation unnecessary downtime in case of a false alarm.

Neural networks process information in a similar way the human brain does. The network is composed of a large number of highly interconnected processing elements (neurons) working in parallel to solve a specific problem. Neural networks learn by example. They cannot be programmed to perform a specific task. The examples must be selected carefully otherwise useful time is wasted or even worse the network might be functioning incorrectly. The disadvantage is that because the network finds out how to solve the problem by itself, its operation can be unpredictable.

CONCLUSION

Identification and interpretation of manufacturing process patterns is an important aspect of diagnosis in a quality improvement environment. Given the occurrence of a particular event, the diagnostic search can be reduced in length of time if one has knowledge of whether the process change is in the mean (such as shift, trend, cycles, etc.) or in variability (such as shift in process variability). This identification and interpretation combined with the corresponding knowledge of the process factors that affect mean and variability of the output, can help in correcting the manufacturing process very early in its erratic behavior and save lot of time, wastage and effort for the organisation. The neural network methodology presented in this work is used to solve this statistical classification problem by identifying the shift in the process mean or variance and also the various patterns such as trends, whenever they are present. It is seen that, the network is quite effective in detecting small process changes when compared with control charts. Also these changes are correctly identified from a small sample, which helps in investigating the process expediently.

Given the description of neural networks and the way they work, neural networks have broad applicability to real world business problems. In fact, they have already been successfully applied in many industries.