

# INTERPRETING STATISTICAL PROCESS CONTROL (SPC) CHARTS USING MACHINE LEARNING AND EXPERT SYSTEM TECHNIQUES

Mark Shewhart

Air Force Logistics Command (AFLC)  
Center for Supportability and Technology Insertion (CSTI)  
Special Projects Office (CSTI/PIPR)  
Wright Patterson AFB, Ohio 45433

**Abstract.** Statistical Process Control (SPC) Charts are one of several tools used in Quality Control. Other tools include flow charts, histograms, cause-and-effect diagrams, check sheets, Pareto diagrams, graphs, and scatter diagrams. A control chart is simply a graph which indicates process variation over time. The purpose of drawing a control chart is to detect any changes in the process, signalled by abnormal points or patterns on the graph. The Special Projects Office of the Center for Supportability and Technology Insertion (CSTI) has developed a hybrid machine-learning and expert-system software tool which automates the process of constructing and interpreting control charts.

## INTRODUCTION

The Air Force Logistics Command (AFLC) has provided TQM and Quality Control training to its employees for several years now. In particular, Statistical Process Control has been emphasized in this effort. While many data collection efforts have been undertaken within AFLC, the SPC Quality Control tool has been under-utilized due to the lack of experienced personnel able to identify and interpret patterns within the control charts. CSTI has developed a software tool which draws control charts, identifies various chart patterns, advises what each pattern means, and suggests possible corrective actions. The application is easily modifiable for process specific applications through simple modifications to the knowledge base portion using any word processing software.

The remainder of this paper consists of the following sections :

- (1) CONTROL CHARTS
- (2) SOFTWARE FUNCTIONALITY
- (3) SOFTWARE DESIGN
- (4) MACHINE LEARNING
- (5) EXPERT SYSTEM
- (6) CONCLUSION

Section (1) provides a more in-depth explanation of the purpose of control charts. Section (2) details the initial functional requirements for the SPC software, and section (3) outlines the design approach used to implement the system requirements. Sections (4) and (5) examine in detail the roles of machine learning

and expert system techniques respectively. Finally, section (6) offers some basic conclusions resulting from this effort.

## CONTROL CHARTS

An example of a control chart is given below in FIGURE 1. A run chart is a plot of a process measurement (e.g. bore diameter or time to process an insurance claim) on the vertical axis (y-axis) against time on the horizontal axis (x-axis). A control chart is simply a run chart with statistically determined upper (Upper Control Limit - UCL) and lower (Lower Control Limit - LCL) lines drawn on either side of the process average. These limits are calculated by running a process untouched, taking samples of the process measurement, and applying the appropriate statistical formulas (references [3-9]).

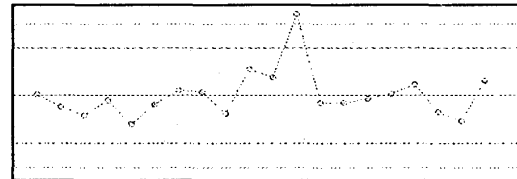
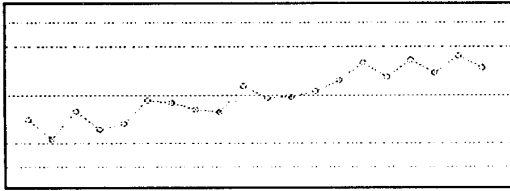


Figure 1 Sample Control Chart

The random fluctuation of points within the limits results from variation built into the process. Such random variation is natural, results from common causes within the system (e.g. design, choice of machine, preventative maintenance, etc.), and can only be affected by changing the system itself. However, points which fall outside of the control limits or which form "unnatural" patterns indicate that some of the variation within the process may be due assignable causes. Assignable causes of variation (e.g. measurement errors, unplanned events, freak occurrences, etc.) can be identified and result from occurrences that are not part of the process.

The purpose of drawing the control chart is to detect any unusual causes of variation in the process, signalled by abnormal points or patterns on the graph. The CSTI developed software tool automatically identifies nine types of patterns which suggest the presence of assignable causes of variation in a

process. An example of one of these patterns is given in **FIGURE 2**. Each of the nine patterns is associated with generic advice about what may be happening at that point in the process.



**Figure 2** An Increasing Trend  
Example of a Pattern Which Suggests the Presence of Assignable Causes of Variation

## SOFTWARE FUNCTIONALITY

An overview of the functionality of the application (referred to as *SPC*) is given below :

- (1) *SPC* determines which type of control chart is appropriate by asking a series of questions about the nature of the user's process data. The appropriate control chart is selected from 8 types of charts (See References [3,4,5,6]) : Run Chart, Moving Range Chart, X-Bar R Chart, X-Bar S Chart, p Chart, pn Chart, u Chart, and c Chart.
- (2) *SPC* graphically displays the chart(s) selected in (1).
- (3) *SPC* identifies 9 pattern types in the chart(s) which suggest the presence of assignable causes of variation : increasing trends, decreasing trends, shifts up, shifts down, cycles, runs, stratification, freak patterns, and freak points.
- (4) *SPC* graphically displays and highlights each chart pattern identified in (3).
- (5) *SPC* displays text in a hypertext fashion which provides generic advice on the meaning of each chart pattern identified in (3).

## SOFTWARE DESIGN

The basic approach to developing *SPC* was to integrate machine learning, expert systems, hypertext, and conventional programming techniques. The machine learning portion of *SPC* was developed using the Abductive Induction Mechanism (AIM) by AbTECH Inc. The expert system portion of *SPC* was developed using an embedded application of the forward chaining expert system tool CLIPS along with a generic end-user interface also developed by CSTI. The hypertext portion of *SPC* was developed using a hypertext authoring and display tool called *HYPERSHOW* developed by the Special Projects Office of CSTI. Turbo C++ was used as the conventional language into which the machine

learning, expert system, and hypertext applications were embedded.

The task for the machine learning portion of *SPC* is to classify every sub-sequence of the control chart according to the presence or absence of five specific chart patterns : increasing trends, decreasing trends, shifts up, shifts down, and cycles. The remaining four chart patterns are identified by conventional methods.

The expert system is initially utilized to help the user select the appropriate type of control chart. This determination is based upon the type of data being collected, the number of data points, and the constancy of the sample sizes.

Another function of the expert system is to interpret the classification results of the trained AIM Network. A control chart with 40 data points will generate over 600 classification results; with nine types of patterns this amounts to over 5500 individual pieces of classification information. This interpretation function represents an ideal expert system application. What requires a few hundred lines of difficult-to-comprehend C code can be implemented using an expert system with only three simple rules! This classification information is then boiled down to about one to ten patterns which are reported to the final expert system application.

The final role of the expert system is to provide advice via the hypertext facility based upon the types of charts and the chart patterns present. The advice currently provided by *SPC* is of a generic nature. For example,

*"A shift up in the R chart indicates that the process is becoming less consistent. This may be due to some sudden change in the process."*

However, the knowledge base is designed to allow for quick modifications to provide process specific advice. For example,

*"A shift up in the R chart has historically been associated (90%) with a loose bearing in the preprocessing machine."*

Conventional software is used to graphically display the control charts, utilize the AIM Networks, provide an end-user interface, and integrate the entire application.

## MACHINE LEARNING

### Role Of Machine Learning

The task of chart interpretation can be summarized as follows. A control chart is simply a sequence or array of floating point numbers. The art of chart interpretation is to determine whether or not sub-sequences similar to several standard patterns are present within the chart. These patterns include trends, shifts, and cycles.

The function of the machine learning tool is to generate code (trained AIM Networks) which can effectively classify a specific sub-sequence of a control chart (array) according to the presence or absence of several standard patterns. With this classification function generated by machine learning techniques, all sub-sequences of the control chart are exhaustively (conventionally) classified by five AIM Networks. The AIM Network classification results are asserted into the fact-list of the CLIPS expert system application.

#### Justification For The Use Of Machine Learning Techniques

Machine learning techniques are used to classify five types of chart patterns - increasing trends, decreasing trends, shifts up, shifts down, and cycles. We could find no references which provide an algorithm for determining whether or not a sequence of real numbers is representative of one of these patterns. In fact, most references on control charts define these patterns by example! The most mathematical approaches to this problem are found in references [1,2] on time series analysis and forecasting. Despite being mathematical in nature, these references still do not describe a deterministic decision procedure. Rather, they provide mathematical heuristics. A sampling of these rules-of-thumb for a times series of length N are given below :

- (1) The number of increasing steps in an increasing trend may be significantly larger than  $(N-1)/2$ .
- (2) The number of discordances in a decreasing trend is usually larger than the expected number of discordances in a random sequence which is  $N*(N-1)/4$ .
- (3) The autocorrelation coefficient sequence of a cycle is usually cyclic.
- (4) The average of the first half of a shift down is always greater than the average of the second half.

Notice that most of these heuristics are in the form of rules with confidence factors. This would seem to suggest the possibility of using a production system for the classification procedure. However, it is almost always the case that the pattern-type (the attribute for which we wish to determine a value) is on the left-hand side of the rule.

This is very similar to some medical diagnosis problems whose domain knowledge is in the form "*Disorder A usually causes symptoms 1, 3, & 4 and may cause symptom 2.*" In cases such as these, the best knowledge-based approach is to use some form of a Hypothesize-and-Test (HT) model. Although the HT approach appears to model the domain very well, we did not pursue this option since we do not have access to a HT development tool which can be easily embedded into other applications.

Attempting to implement such applications using a rule-based system with confidence factors ultimately boils down to an iterative process of re-adjusting confidence factors and re-testing the rule base on a set of examples. This iterative process, however, is quite analogous to the process of training a neural network or a machine learning tool on a set of examples. Given this analysis and the fact that most references on control charts define these patterns by example, we elected to implement a portion of the classification process using a machine learning tool.

#### Representation Of Control Chart Sub-sequence

The function of the machine learning tool is to classify a specific sub-sequence of a control chart according to the presence or absence of several standard patterns. A key question relating to the use of machine learning tools, is how do we represent an arbitrary length sub-sequence of an arbitrary length sequence of numbers as a fixed length vector of real numbers. The approach is to represent a sub-sequence of a control chart as a fixed length vector of statistical features.

Twenty (20) statistical features are extracted from each sub-sequence  $X[1..N]$  under consideration. Features 1 - 10 are raw statistical features such as linear regression coefficients, the standard deviations of the first and second halves of the sequence, and the Box-Pierce Q-statistic [1, p 269]. Features 11 - 20 are Boolean type indicator variables such as whether the sequence has ten or more points and whether one of the regression coefficients is "low" in magnitude. Most of these Boolean indicators were included as a result of a database analysis on the training sets which produced "statistics" such as "*90% of increasing trends have a linear regression coefficient greater than 0.8*". A more detailed explanation of the 20 statistical features used to represent a chart subsequence to the AIM Networks is given in **ATTACHMENT A**.

#### Training And Test Sets For Machine Learning Tool

Over 70,000 sample chart sub-sequences were generated to train and test the AIM Networks. Most of these sub-sequences were generated by adding random noise of various magnitudes to existing control charts with existing patterns. Each chart sub-sequence generated a training/test vector of dimension 25 - 20 real-valued Network inputs (statistical features) and 5 bi-polar (-1 or 1) outputs. One AIM Network was trained for each of the 5 outputs. Each AIM Network required from two to six hours to train on a 386 machine with math co-processor.

#### Machine Learning Test Results

The results of the AIM Networks applied to control chart patterns not present in the training sets were quite good. The increasing trend network classified the test data %97.5 correctly, the decreasing trend network %97.3, the shift up network %98.8, the shift down network %98.8, and the cyclic network

%92.0. Most (99%) of the errors in the cyclic network occurred in the short saw-toothed patterns with added noise. If the saw-toothed pattern is deemed by experts/customers to be very important, a separate network could be developed for the saw-toothed pattern. This would increase the overall cyclic percent correct to about 96% and provide a better recognition rate for noisy, short saw-toothed patterns.

## EXPERT SYSTEM

### Role Of Expert System

The role of the expert system in *SPC* is three-fold. One knowledge base helps the user select the type of control chart to be used, another interprets the AIM Networks' classification results, and the third knowledge base provides expert advice on the meaning of any identified patterns.

### Selecting Appropriate Control Chart Type

The knowledge base for this portion of the expert system application in *SPC* is given below in TABLE 1. In short, the type of control chart is selected based upon (1) the number of data points, (2) the logical group size, (3) whether the data is attribute data or measurement data, (4) whether the logical group size is constant or variable, and (5) whether the (attribute) data is measuring defectives or defects.

```
(defrule XBAR_R_Chart
  (data_type measurement)
  (test (< sample_size 10))
  =>
  (assert (chart_type XBAR_R)) )
```

TABLE 1 Sample Rule To Select Chart Type

### Interpreting AIM Network Classification Results

A major issue during the development of *SPC* was how to interpret the AIM Networks' classification results. An example of a portion of the results of the AIM Networks' classification during the exhaustive conventional search is given in TABLE 2. The classification results of the AIM Networks appear in the following format :

(*chart pattern begin end network-score*).

Notice that from points 6 to 17 there are 17 sub-sequences which the AIM increasing trend Network gave high scores to! Clearly we cannot report to the user all 17 patterns. The expert system application which interprets the AIM Networks' classification results is shown in TABLE 3 and is composed of three rules :

```
( X cycle 1 13 0.743 )
( X inc_trend 5 17 0.098 )
( X shift_up 6 18 0.282 )
( X inc_trend 6 17 0.819 )
( X inc_trend 6 16 1.000 )
( X inc_trend 7 17 0.829 )
( X inc_trend 6 15 1.000 )
( X inc_trend 7 16 0.874 )
( X inc_trend 6 14 0.991 )
( X inc_trend 7 15 1.000 )
( X inc_trend 6 13 0.951 )
( X inc_trend 7 14 1.000 )
( X inc_trend 8 15 0.973 )
( X inc_trend 6 12 0.807 )
( X inc_trend 7 13 0.997 )
( X inc_trend 8 14 0.961 )
( X inc_trend 9 15 0.841 )
( X inc_trend 7 12 0.904 )
( X inc_trend 10 15 0.917 )
( X inc_trend 10 14 0.895 )
```

TABLE 2 Sample Fact List Generated By AIM Network

(1) The first rule eliminates from consideration any pattern whose AIM Network score is below a certain threshold. The sensitivity of the pattern recognition can be adjusted by altering these thresholds in the defacts statement.

(2) The second rule eliminates from consideration any pattern which is contained entirely within another existing pattern of the same type. It is assumed that the first rule has previously been applied. For example, the fact (*X inc\_trend 8 14 0.961*) would be retracted due to the presence of the fact (*X inc\_trend 6 16 1.0*).

(3) The third rule eliminates from consideration any pattern which overlaps another existing pattern of the same type but with a higher AIM Network score. It is assumed that the first two rules have previously been applied. For example, this rule would retract the fact (*X inc\_trend 8 14 0.961*) due to the presence of the fact (*X inc\_trend 7 13 0.997*).

### Expert Advice On Meaning Of Chart Patterns

The majority of the expert system interaction that the user will see involves explanations and advice regarding any patterns that the AIM Networks have identified as indicators of assignable causes of variation. At the most basic level, this expert knowledge simply consists of triples of the form *<chart-type, pattern-type, advice-text>*. The current CSTI SPC software consists of knowledge at this level of complexity only. The rule-base representation was selected to allow for the most flexible knowledge representation scheme. A sample of the CLIPS implementation of such knowledge is illustrated in TABLE 4.

```

(deffacts thresholds
  (threshold run 0.99)
  (threshold inc_trend 0.95)
  (threshold dec_trend 0.95)
  (threshold shift_up 0.95)
  (threshold shift_down 0.95)
  (threshold stratification 0.99)
  (threshold freak_point 0.99)
  (threshold freak_pattern 0.99)
  (threshold cycle 0.95) )

(defrule simple_threshold
  (resolve thresholds)
  ?pattern <- (?chart ?type ?a ?b ?score)
  (threshold ?type ?thresh)
  (test (< ?score ?thresh))
  =>
  (retract ?pattern) )

(defrule subset
  (resolve subsets)
  (?chart ?type ?a1 ?b1 ?score1)
  ?subset_pattern <- (?chart ?type ?a2 ?b2 ?score2)
  (test (not (and (= ?a1 ?a2) (= ?b1 ?b2))))
  (test (and (<= ?a1 ?a2) (<= ?b2 ?b1) ))
  =>
  (retract ?subset_pattern) )

(defrule overlap
  (resolve overlap)
  (?chart ?type ?a1 ?b1 ?score1)
  ?pattern2 <- (?chart ?type ?a2 ?b2 ?score2)
  (test (not (and (= ?a1 ?a2) (= ?b1 ?b2))))
  (test (>= ?score1 ?score2))
  (test (or (and (<= ?a1 ?a2 ?b1) (< ?b1 ?b2) )
            (and (<= ?a2 ?a1 ?b2) (< ?b2 ?b1) ) ))
  =>
  (retract ?pattern2) )

```

TABLE 3 Rules To Interpret AIM Network Results

```

(defrule R_shift_up
  (R_shift_up ?a ?b ?score)
  =>
  (write_paragraph "advice.idx" "R_shift_up" )

```

\*The function *write\_paragraph* is provided by the CLIPS Application User Interface (AUI) also developed by CSTI at Wright-Patterson AFB, Ohio.

TABLE 4 Sample Rule To Provide Expert Advice

## CONCLUSION

*SPC* is a good example of a hybrid system which integrates machine learning, expert system, and conventional programming techniques. It is a classic example of pattern recognition and is an excellent

demonstration of problem representation techniques necessary when using machine learning or neural network tools.

Two features distinguish *SPC* from most other control chart software. (1) *SPC* automatically identifies and highlights unusual chart patterns. Most related commercial software simply draws the chart and explains to the user what unusual patterns to look for. We found no commercial software which automatically identified trends, shifts, or cycles. (2) *SPC* provides expert advice on the meaning of all identified unusual chart patterns. Over 50% of available commercial software only construct the control chart for the user and go no further.

*SPC* Version 1.01 was released in October 1991 and Version 1.1 is scheduled for release in March 1992. CSTI plans to provide software enhancements to *SPC* based upon continuous customer feedback and demand. Already *SPC* has a wide variety of customers. For example, the Wright-Patterson Medical Center (WPMC) has selected *SPC Version 1.1* over existing commercial software packages as the official SPC software tool for the WPMC. The Quality Programs Directorate of the Air Force Logistics Command (AFLC/QP) recommends that the software be distributed to students in all AFLC sponsored SPC courses. Copies of *SPC* and reprints of this paper are available to government agencies and domestic US companies upon written request.

## REFERENCES

- [1] Spyros Makridakis and Stephen C. Wheelwright, "Forecasting: Methods and Applications", Wiley/Hamilton, 1978.
- [2] Sir Maurice Kendall and J Keith Ord, "Time Series", Oxford University Press, 1990.
- [3] SPC Course Materials, Decision Dynamics Inc., 1990
- [4] Kaoru Ishikawa, "Guide to Quality Control", Asian Productivity Organization, 1982.
- [5] Perry Johnson Inc., "SPC Chart Interpretation", Perry Johnson, Inc., 1987.
- [6] J.M. Juran, Dr. Frank M. Gryna, Jr., and R.S. Bingham, Jr., "Quality Control Handbook", Third Edition, McGraw-Hill, 1974.
- [7] Western Electric Company, "Statistical Quality Control Handbook", Western Electric Co., Inc., 1958.
- [8] H. Besterfield, "Quality Control", Second Edition, Prentice-Hall.
- [9] Douglas C. Montgomery, "Introduction to Statistical Quality Control".

## ATTACHMENT A

### Statistical Features Used To Represent Chart Subsequences

- (1) RMS\_SU - This is the root-mean-squared difference between  $X[1..N]$  and an "ideal" shift-up pattern.
- (2) RMS\_SD - This is the root-mean-squared difference between  $X[1..N]$  and an "ideal" shift-down pattern.
- (3) A - This is the simple linear regression coefficient when trying to approximate the time series  $X[t]$  using  $X[t] = A + Bt$ .
- (4) B - This is the simple linear regression coefficient when trying to approximate the time series  $X[t]$  using  $X[t] = A + Bt$ .
- (5) SIGMA 1 - This is the standard deviation of the first half  $X[1..N/2]$  of the sequence  $X[1..N]$ .
- (6) SIGMA 2 - This is the standard deviation of the second half  $X[N/2 + 1..N]$  of the sequence  $X[1..N]$ .
- (7) R\_root\_N\_r - The percentage of the first  $N/4 + 1$  autocorrelation coefficients  $r(k)$  for which  $\text{abs}(r(k)) > 1.96/\text{sqrt}(N)$ .
- (8) CHI\_SQ\_TEST - This is the Box-Pierce Q-statistic which is capable of determining whether several autocorrelation coefficients are significantly different from zero. This is defined in reference [1,p 269]
- (9) CONCORD - This is the number of concordances Q in  $X[1..N]$  divided by the maximum possible number  $N(N-1)/2$  of concordances. This is defined in reference [2,pp 21-23].
- (10) DISCORD - This is the number of discordances P in  $X[1..N]$  divided by the maximum possible number  $N(N-1)/2$  of discordances. This is defined in reference [2,pp 21-23].
- (11) TEN\_PLUS - An indicator variable used to indicate if  $X[1..N]$  has length less than ten. This is important since many statistical significance tests are ineffective for small sample sizes.
- (12) CCRD\_LOW - An indicator variable used to indicate whether CONCORD is less than 0.7. The value of 0.7 was chosen since a database analysis indicated that a high percentage of increasing trends had  $\text{CONCORD} > 0.7$ .
- (13) DCRD\_LOW - An indicator variable used to indicate whether DISCORD is less than 0.7. The value of 0.7 was chosen since a database analysis indicated that a high percentage of decreasing trends had  $\text{DISCORD} > 0.7$ .
- (14) HIGH\_ISD - An indicator variable used to indicate whether RMS\_SD is greater than 1.8. The value of 1.8 was chosen since a database analysis indicated that a high percentage of shifts-up had  $\text{RMS\_SD} > 1.8$ .
- (15) HIGH\_ISU - An indicator variable used to indicate whether RMS\_SU is greater than 1.8. The value of 1.8 was chosen since a database analysis indicated that a high percentage of shifts-down had  $\text{RMS\_SU} > 1.8$ .
- (16) GOOD\_INC\_MM - An indicator variable used to indicate when the sequence minimum was early and the sequence maximum was late. The first 20% and last 20% was chosen since a database analysis indicated that a high percentage of increasing trends had their minimum and maximum within the first 20% and last 20% respectively of the sequence.
- (17) GOOD\_DEC\_MM - An indicator variable used to indicate when the sequence maximum was early and the sequence minimum was late. The first 20% and last 20% was chosen since a database analysis indicated that a high percentage of decreasing trends had their maximum and minimum within the first 20% and last 20% respectively of the sequence.
- (18) HIGH\_R\_root\_N\_r - An indicator variable used to indicate whether  $R\_root\_N\_r$  is greater than 0.1. The object of introducing this variable was to help draw a distinction between random sequences and cycles. The value of 0.1 was chosen since a database analysis indicated that a high percentage of cycles and a low percentage of random sequences had  $R\_root\_N\_r > 0.1$ .
- (19) SMALL\_A - An indicator variable used to indicate whether the absolute value of A is less than 0.8. The object of introducing this variable was to help draw a distinction between random sequences or cycles and the other chart patterns. The value of 0.8 was chosen since a database analysis indicated that a high percentage of cycles and random sequences and a low percentage of other types of patterns had  $\text{abs}(A) < 0.8$ .
- (20) MAYBE\_CYCLE - An indicator variable used to indicate when both  $R\_root\_N\_r > 0.1$  and  $\text{ABS}(A) < 0.8$ . This is the logical AND of variables 18 and 19.