Pergamon

0098-1354(94)E0042-L

Computers chem. Engng Vol. 19, No. 2, pp. 129–137, 1995

0098-1354/95 $9.50 + 0.00

PROCESS TRENDING WITH PIECEWISE LINEAR SMOOTHING

R. S. H. MAH†, A. C. TAMHANE, S. H. TUNG and A. N. PATEL
McCormick School of Engineering and Applied Sciences, Northwestern University, Evanston, IL 60208-3120, U.S.A.

(Received 20 September 1993; final revision received 7 March 1994; received for publication 8 April 1994)

Abstract—Compression and trending are techniques widely used in the initial treatment of raw process data. Two popular methods, Box Car with Backward Slope and Swinging Door, were reviewed recently by Kennedy. In spite of their popularity, neither of these methods are designed to cope with process variability and outliers. They also require one or more parameters to be specified based on practical considerations. In this paper we propose a new method, Piecewise Linear Online Trending (PLOT), which is statistically based and which performs significantly better. Unlike the two existing methods, it adapts to process variability and noisy data, recognizes and eliminates outliers, and it is robust even in the presence of outliers. It fits the data better for the same number of trends. The fidelity of its performance may be fine-tuned with a single level of significance which may be set by the user without requiring any expertise in statistics. It may be used in an online or a batch mode, and interfaces easily with most existing packages.

INTRODUCTION

For more than two decades now computers have been used to monitor and control industrial processes. One outcome of in-plant computer applications is the availability of improved process data in quantity and quality. At this point in time the mechanical and electronic aspects of data acquisition, organization, storage and retrieval (De Heer, 1987; Kennedy, 1993) are fairly well explored. Large volumes of sensed data are captured and stored in electronic devices with rapid access. How to utilize this rich source of information continues to be a major technological challenge. Potential uses of process data are numerous. Examples of applications are given by Langen (1984), for instance. They include:

1. Adverse events: diagnosing the cause of failure, tracing event path, selecting corrective actions.
2. Normal operation: monitoring process trends and conditions at different levels of process operation.
3. Startup: online trending and multiple point comparisons with test records.
4. Maintenance: detecting, diagnosing and predicting malfunctions using signatures of data trends.
5. Innovation: improving process operation through comparison with historical data.
6. Training: systematic analysis of scenarios and exploration of alternative operating strategies.
7. Regulations: providing a rational basis for developing regulatory guidelines and practice.

In this paper we shall focus on sensed data as time series.

Historically, the initial concern was how to cope with large volumes of sensed data. Instead of storing every piece of data, procedures were proposed for storing fewer pieces of data. The term commonly used was *data compression*. As the performance of storage devices and microprocessors continues to improve, and as their costs continue to decline, the emphasis is increasingly shifting towards *process trending*. In trending, the measured data are replaced by a fitted trend. The trend is summarized by an equation. Values of the parameters in the equation are stored instead of the raw data. At the present stage of the development, this shift represents a change of viewpoint rather than a change of methodology. However, in the long run this viewpoint may provide a more natural transition to the rapidly developing technology of pattern recognition.

In this paper we present a new method for process trending. We begin with a brief review of two existing procedures for data compression and trending commonly applied to sampled data systems.

† To whom all correspondence should be addressed.

PREVIOUS WORKS

Box Car and Backward Slope algorithm (Hale and Sellars, 1981)

The *Box Car* algorithm (Hale and Sellars, 1981) records a value only when the current value differs from the last recorded value by an amount greater than or equal to the recording limit for that variable, specified by the user. This algorithm is extremely simple and is quite effective for processes with long spells of stable operation. However, it is not very effective when the process follows a linear drift or undergoes a transition between steady states. For instance, in case of rapidly increasing or decreasing trend, e.g. the level of a large vessel when it is being filled or emptied at a large but steady rate, this algorithm will not achieve a high data compression. On the other hand, if the rate of change is slow, the box car algorithm may achieve compression at the expense of loss of information. Efforts to overcome this shortcoming led to the development of the so-called *Blackward Slope* algorithm (Hale and Sellars, 1981).

In this algorithm, current value of a variable is predicted based on linear extrapolation of the last two recorded values. If the actual value differs from the predicted value by an amount greater than the prespecified recording limit, then the current value is recorded. This algorithm retains the merit of simplicity. For processes involving ramp and step changes, this algorithm yields higher compression for data without noise. But in practice the Backward Slope algorithm does not always produce better results. This is especially true for noisy data. The noise in the data will cause the algorithm to project a slope which is misdirected. For instance, when the liquid level in a vessel is actually steady, the noise may give rise to a nonzero slope resulting in a recorded value which should not have been recorded. In this instance the Box Car algorithm will perform much better.

To capture the advantages of both techniques, they were combined into a single algorithm which dynamically selects the technique to be applied to the next data point. The combined algorithm records a value when an exceptional value is indicated by both the Box Car and the Backward Slope algorithms. As long as the trend continues, only one criterion need be checked. However, as soon as this criterion fails to be satisfied, the second criterion is tested. If that criterion also fails, a new value is recorded. This algorithm works better than either of the other two algorithms, but requires more computation. For many years it was widely used in the process industry, even though its performance was far from satisfactory. The next algorithm tries to address some of these issues.

Swinging Door algorithm (Bristol, 1990)

Swinging Door Trending (SDT) is an heuristic straight line trending and compression technique. It strives to give the longest straight line trend possible, given the data and the maximal error allowed. It also attempts to minimize computation.

In essence, the algorithm replaces a sequence of consecutive data points by a straight line, defined by *an initial point* and *a final point*. The algorithm specifies how long a sequence or *time interval* to take and where the final point of this time interval and the initial point of the next time interval are to be located. The very first data point is taken as the initial data point of the first time interval. Thereafter, the procedure is applied sequentially to each subsequent time interval.

With reference to Fig. 1, the initial data point is bracketed vertically by two points, known as *hinges* or *pivot points*, each located at a distance E from the initial point. A *door* is constructed with a straight line hinged at a pivot point. To start with we have only one data point in the sequence, namely, the initial data point i, and both doors are closed. That is, each door is represented as a vertical straight line through the pivot. As more data points are introduced, the doors will swing open or remain stationary as the case may be. Notice the width of each door (the straight line segment) may be extended, but, once open, neither door will close again until the end of the current time interval. For instance, with reference to the data points (i, a, b, c, d, e, f, g) in Fig. 1, the movements of the doors are summarized in Table 1. The swinging door operation continues with each succeeding new data point as long as the two doors are less than parallel. In other words, the sum of the two interior angles† between each door and the vertical line is less than 180°. In Fig. 1, data point d is the last point before this condition occurs, which demarcates the end of a time interval.

Given point d as the final point of this interval, the initial point of the next time interval may be selected in different ways. To avoid any offset (vertical discontinuity), point d may be taken also as the first

† The interior angles are the angles between each door and the vertical line. To start with both interior angles are zero. As each data point is processed, each interior angle is either enlarged or remains the same. In other words, the changes of each interior angle are monotonically increasing. The first interior angle to reach a value of 90° determines the trend, i.e. inclining or declining. When the sum of upper and lower interior angles reaches 180°, the shape of the box enclosing the compressed data points become fully defined.

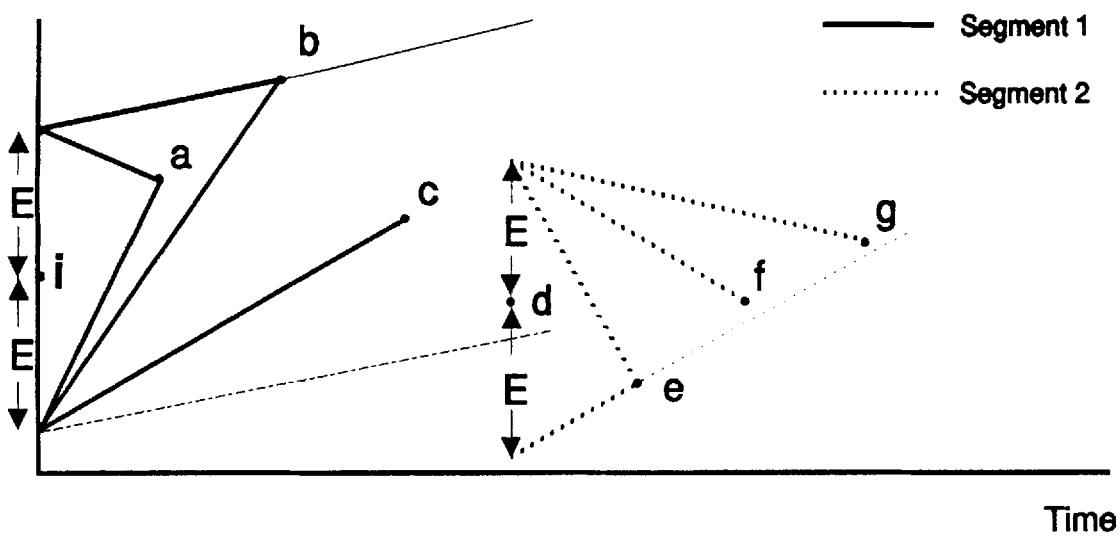

Fig. 1. Steps in SDT algorithm: basic mode, no offset.

point of the new time interval, or the first point of the new time interval may be offset from point d by an amount equal to one-half of the maximum allowable error (Bristol, 1990).

The above procedure is referred to as the *basic mode*. Compression is achieved by replacing the data points in each time interval by a straight line segment. Other compression modes may be devised to handle other specific trends, e.g. transients, step, pulse and so on. SDT algorithm switches between the basic mode and other compression modes. In SDT the parameter E is treated as the user stated required accuracy. It is chosen on the basis of practical considerations external to the SDT algorithm. SDT algorithm does not detect and handle outliers. As shown in Fig. 1, the trend and compression are critically affected by the choice of E. If E is chosen too large, the trend may be grossly incorrect. If E is chosen too small, very few data points may be eliminated. SDT gives very poor performance in the presence of noise and outliers.

PIECEWISE LINEAR ONLINE TRENDING (PLOT)

Given a time series of process data measurements y_t, the aim of the proposed method is to determine all major trends, or equivalently, to obtain maximum possible data compression by the use of piecewise linear smoothing. The measurements are assumed to be sampled at discrete and equal time periods and follow the model:

$$y_t = \mu_t + \varepsilon_t, \quad t = 1, 2, \ldots \tag{1}$$

where μ_t is the process mean at data sampling time t and ε_t is the measurement error. The measurement errors are assumed to be independent and identically distributed $N(0, \sigma^2)$ random variables. We further assume that the variation in the process mean μ_t with time is well-approximated by a piecewise linear function as follows: let the trend of the data points change at time t_j and let $0 = t_0 < t_1 < t_2 < \ldots$. During the jth trending interval, i.e. $t_{j-1} \leq t < t_j$, the form of the linear function is given by:

$$\mu_t = \mu_{t_{j-1}} + \delta_{j-1} + \beta_j (t - t_{j-1}), \quad j = 1, 2, \ldots \tag{2}$$

where $t_0 = 0$, $\delta_0 = 0$, δ_j is the jump or step change in the process mean at time t_j, and β_j is the slope of the line during the trend interval j. The model is depicted in Fig. 2.

The PLOT algorithm

Based on this model a trending algorithm PLOT was developed and implemented on computer. Our goal is to estimate the straight line, equation (2), as precisely as possible for any trend interval j and also to detect a change in the trend, i.e. determine the break point t_j, which marks the beginning of the new trend interval $j+1$, as early as possible. We also wish to detect outliers in the data and not let the trends be affected by such outliers. For this purpose

Table 1. Movements of swinging doors

Door	*i*	*a*	*b*	*c*	*d*	*e**	*f*	*g*
Upper	closed	open	open	—	—	open	open	open
Lower	closed	open	open	open	open	open	—	—

* Second point of a new sequence or time interval.

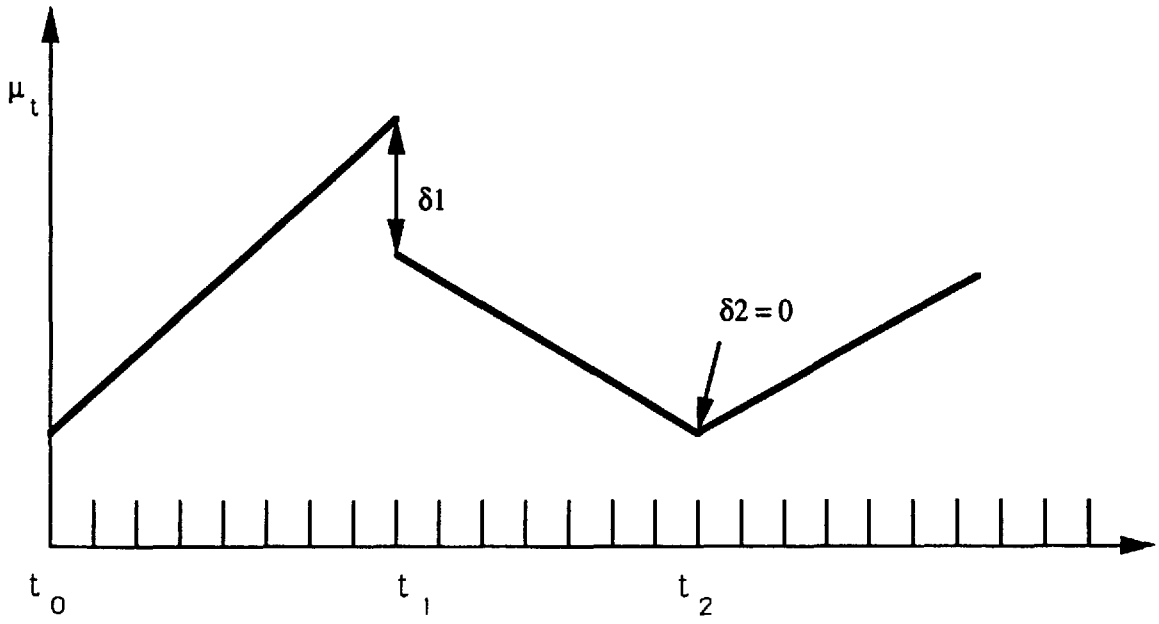

Fig. 2. Model of piecewise linear trending.

a least squares straight line is fitted† to the sampled data collected during the current trend interval. The estimated straight line is updated sequentially as each new data point becomes available. The least squares (LS) method also provides an estimate of the process variance σ^2 through the mean square error (MSE). Because of the assumption of a constant variance, we can pool estimates of σ^2 from successive time periods.‡ Whether a break point has occurred is determined by predicting the next data point from the straight line fit which is obtained using data points up to the current data point, and by checking to see if the next data point falls within the $100(1-\alpha)\%$ prediction interval. If the next data point falls outside the prediction interval, then this may indicate that the trend has changed or that the data point may be an outlier. Additional data points are observed before deciding on the follow up action.

Note that the width of the prediction interval corresponds to the parameter E used in the SDT algorithm. However, whereas E is fixed *a priori*, the width of the prediction interval in the PLOT algorithm depends on the estimate of the process variance among other factors. Thus the amount of data compression adapts automatically to process variability. In SDT algorithm E is chosen by the user. If it is chosen too small, false changes in the trend may be indicated, and vice versa.

For the purpose of computing data compression, it is necessary to introduce the notion of a time horizon consisting of H time periods even though the algorithm is applied continually online. We define the *data compression ratio* ρ as the number of measured data points H over the number of trending points to be stored. Note that our definition of data compression ratio adapts itself to different sampling, model and storage arrangements. In the simplest situation for which data points are sampled at H equal and consecutive time periods and no jumps are present ($\delta_j = 0$) between pairs of consecutive J trends, the data compression ratio is given by:

$$\rho = \frac{H}{2J+1}.$$

If jumps are always present, the data compression ratio changes to:

$$\rho = \frac{H}{3J+1}.$$

For any situation in between and with unequal sampling periods, the data compression ratio will depend on the storage method used and whether one can take advantage of the sparse structure.

The steps of the PLOT algorithm are as follows:

1. Set time period $j=1$. Begin by sampling the first n consecutive data points, $y_1, y_2, \ldots, y_n$. Since these measurements will be used to compute an estimate of the process standard deviation, σ, n should be at least 3. A larger value

† Note that we could have chosen another curve fitting method instead. But the least squares fit is attractive because of its simplicity.

‡ Notice that there is some latitude in this matter. If so desired, the estimate may be computed from the data collected in all current and past trend intervals. Alternatively, pooling may be limited to the current and a certain number of past time intervals.

is preferable, if we can be sure that the trend does not change during the sampling. The choice should be guided by the user's knowledge of the application and supported, if necessary, by experimentation.

2. Fit a straight line to the n data points using the LS method. Let $\hat{\sigma}=\sqrt{\text{MSE}}$ be the corresponding estimate of σ with $n-2$ degrees of freedom (df).
3. Compute the predicted value, $\hat{y}_{n+1}$, based on the extrapolation of the fitted LS line. Also compute a $100(1-\alpha)\%$ prediction interval for y_{n+1} using the generic formula:

$$\hat{y} \pm t_{\nu,\alpha/2}\hat{\sigma}\sqrt{1+\frac{1}{N}+\frac{(x-\bar{x})^2}{\sum_{i=1}^{N}(x_i-\bar{x})^2}}, \qquad (3)$$

when N is the number of data points, ν is the df on which the estimate of σ is based, $t_{\nu,\alpha/2}$ is the upper $\alpha/2$ critical point of the t-distribution with ν df, x_i, $i=1,2,\ldots,N$ are the time points at which the data are sampled, and x is the time point for which the prediction is to be made. When predicting y_{n+1} based on the data collected at times $i=1,2,\ldots,n$, assuming that no outliers have been detected before time period n, formula (3) simplifies to:

$$\hat{y}_{n+1} \pm t_{n-2,\alpha/2}\hat{\sigma}\sqrt{1+\frac{1}{n}+\frac{[(n+1)/2]^2}{n(n^2-1)/12}}. \qquad (4)$$

4. (a) If y_{n+1} falls inside its prediction interval, add this data point to the set $y_1, y_2, \ldots, y_n$, replace n by $n+1$, and go to Step 2.
   (b) If y_{n+1} falls outside its prediction interval, y_{n+1} may be an outlier or a change of trend may be indicated. To solve the dilemma go to Step 5.
5. Consider the next measurement y_{n+2}. Check whether it falls within its prediction interval using the LS line based on the first n data points. If y_{n+2} falls outside its prediction interval, go to Step 6; otherwise go to Step 7.
6. Conclude that a new trend has started. Advance the trend index by one to $j+1$. Relabel the time points $n+1, n+2$ as $1, 2$ and y_{n+1}, y_{n+2} as y_1, y_2, and fit a straight line to these two points. Use the previous estimate of σ and its df and go to Step 3.
7. Compute the following two prediction intervals using the general formula, equation (3) for y_{n+3}:
   (a) Based on the LS line fitted to data points $y_1, \ldots, y_{n+2}$, i.e. by omitting y_{n+1}, which is hypothesized for the time being to be an outlier in the previously established linear trend up to time $n+2$. The hypothesis will be confirmed in Step 8a or denied in Step 8b.
   (b) Based on extrapolating the straight line through points y_{n+1}, y_{n+2} and using the previous estimate of σ. Go to Step 8.
8. (a) If y_{n+3} falls inside the prediction interval of Step 7a but not of 7b, then we conclude that y_{n+1} is an outlier and eliminate it from further consideration. Relabel data point $n+2$ as $n+1$, and the data point $n+3$ as $n+2$. Replace n by $n+2$. We now have two more data points in the prediction intervals of the current trend. Re-estimate the slope and the intercept. Go to Step 3.
   (b) If y_{n+3} falls inside the prediction interval of Step 7b but not of Step 7a, then we conclude that y_{n+1} is not an outlier. Begin a new trend at y_{n+1}. Relabel $y_{n+1}, y_{n+2}, y_{n+3}$ as y_1, y_2, y_3. Set $n=3$, replace j by $j+1$, estimate the slope and the intercept, and go to Step 3.
   (c) If y_{n+3} falls in both prediction intervals, change time index from n to $n+1$. Then go to Step 7. That is, repeat Step 7 onward for the next data point y_{n+4}.
   (d) If y_{n+3} falls outside both prediction intervals, change the time index from n to $n+1$. Then go to Step 7. That is, repeat Step 7 onward for the next data point y_{n+4}, and so on until the dilemma is resolved.

A close examination of the above algorithm suggests that whenever a measurement falls outside its prediction interval, the algorithm slightly favors the hypothesis that it is the start of a new trend. An alternative algorithm is to favor the hypothesis that it is an outlier. Yet another alternative is to use a voting rule. In our experience these refinements do not drastically alter the performance of PLOT.

Performance evaluation

Computer runs were made in order to evaluate the effectiveness of the PLOT algorithm and to compare its performance with the SDT algorithm. Table 2 summarizes the results of 5 runs using noise-free synthetic data. These data were taken from models of increasing complexity. In Run 1 the data were taken from a constant straight line with $y=80$. In Run 2 the underlying model was a ramp function ($y=20+1.2x$). In Run 3 the model combined a ramp with a step ($y=80+1.2x$, $x\leq 250$; $y=420$, $x>250$). A sinusoidal model was used in Run 4 ($y=10+\sin x$). Finally, in Run 5 the model combined all four features listed above. These runs were

Table 2. Synthetic data (without noise)

Run	Method	Data points	SSE	R^2	Intervals (J)	$\frac{H}{2J+1}$	$\frac{H}{3J+1}$
1	PLOT	150	0	1.0	1	—	—
	SDT		0	1.0	1	—	—
2	PLOT	300	0	1.0	1	-	-
	SDT		0	1.0	1	—	—
3	PLOT	500	0	1.0	2	100	71.4
	SDT		0	1.0	2	100	71.4
4	PLOT	500	247.2	0.0111	2	100	71.4
	SDT		504.1	−1.0165	2	100	71.4
5	PLOT	750	$3E-4$	1.0000	126	2.8	1.8
	SDT		$8E-4$	1.0000	126	2.8	1.8

designed to verify the proper functioning of the programs.

Runs 6–10 used the same models as Runs 1–5 except for the inclusion of noises which were taken to be Gaussian noises with standard deviation of one. They also differed from the earlier runs in the number of data points. All these runs were made at $\alpha = 0.05$. The results are shown in Table 3.

Finally, Tables 4 and 5 present some typical results obtained on industrial plant data using PLOT and SDT methods, respectively. These data were taken on ten measured variables associated with one industrial plant, sampled at one minute intervals, over a time horizon of approxiately 4500 min. Similar results were obtained using data from two other plants. In these runs the outliers were recognized and eliminated by PLOT, but not by SDT. The estimated δ_j were plotted (but not presented in this paper because of the space limitations). They were unimodal and approximately symmetric. A majority of these jumps lie within the $\pm 3\sigma$.

Analysis of industrial data and display of trend line may be carried out in batch or online modes depending on the applications. To give the reader a feel of the data and trend lines, the 4491 raw data points used in Run C10 are displayed in Fig. 3a. A window is provided for selecting any subset of consecutive data points for enlargement and display, as illustrated in Fig. 3a and 3b. In this illustration 80 data points beginning with $t = 1461$ are selected. Options are provided for displaying, using color graphics, observed and predicted data points, and trend lines with or without data points. At appropriate levels of scale up outliers are marked with a cross (one outlier in Fig. 3b). Finally, the results are summarized in a separate file and used to construct Tables 4 and 5.

Discussion of results

Although noise-free data were used in Runs 1–5, in reality random errors were almost always present as a result of rounding off digital representations. It is important that σ be set to a small value which is of the order of the maximum value of rounding off errors in the computation, say 6 or 7 significant figures in 32 bit computation. Setting $\sigma = 0$ may create artificial trends and intervals and result in a reduction of data compression ratio. This remark is particularly relevant for Runs 1–3.

The coefficient of determination, R^2, is a general measure of how well a trend line fits the data. On the other hand, lack of fit is indicated by SSE. An interesting point to note is that there is a trade-off between the data compression ratio and the coefficient of determination. In order to compare the two methods we adjusted E in SDT so that the same data compression ratios (column 8 in Table 3, and column 7 in Tables 4 and 5), or the same number of intervals (see column 6 in Table 3, and columns 6 in Tables 4 and 5) are obtained. For each set of data we first applied the PLOT algorithm and computed the data compression ratio. Next, we applied the SDT

Table 3. Synthetic data (with noise)

Run	Method	Data points	SSE	R^2	Intervals	$\frac{H}{2J+1}$	$\frac{H}{3J+1}$	Outliers detected
6	PLOT	300	247.1	0.0283	5	27.3	18.8	8
	SDT		607.8	−1.1054	6	23.1	15.8	—
7	PLOT	500	391.3	1.0000	7	33.3	13.6	12
	SDT		725.0	1.0000	7	33.3	13.6	—
8	PLOT	500	384.6	0.9999	8	29.4	20	12
	SDT		656.9	0.9999	8	29.4	20	—
9	PLOT	500	469.2	0.2575	27	9.1	6.1	19
	SDT		938.6	−0.2798	27	9.1	6.1	—
10	PLOT	750	664.2	0.9999	15	24.2	16.3	21
	SDT		1611.4	0.9998	17	21.4	14.4	—

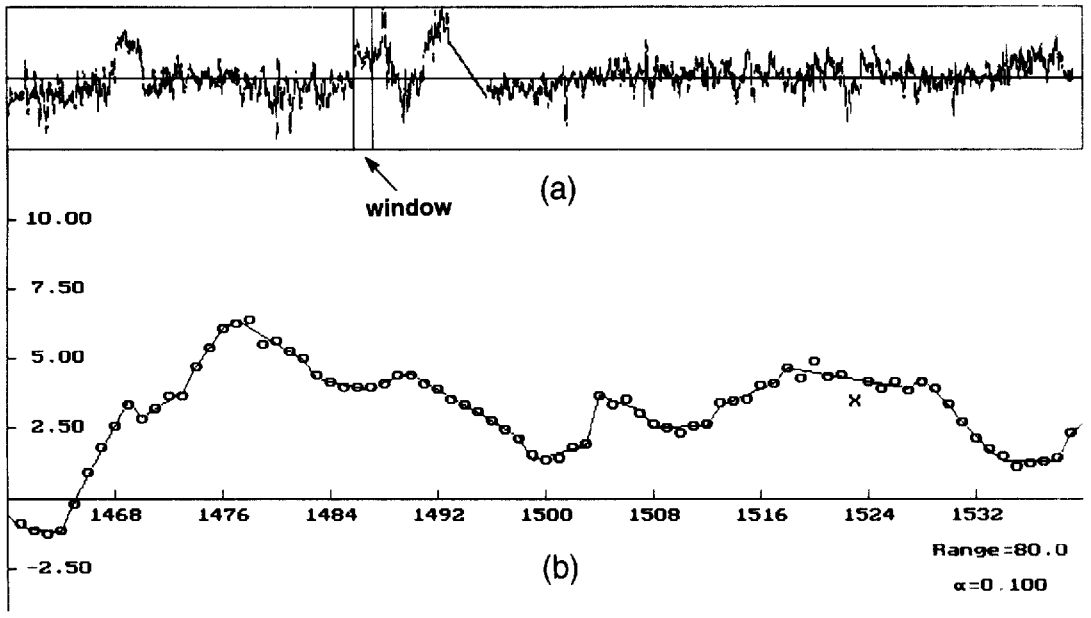

Fig. 3. Graphical display of process data points and trends using PLOT.

algorithm to achieve the same data compression ratio by adjusting the value of E. Finally, we computed and compared the value of SSE (see column 4 in Tables 3, 4 and 5) and the coefficients of determination, R^2 (see column 5 in Tables 3, 4 and 5) for the two methods. The results in Tables 3, 4 and 5 showed PLOT to be consistently superior with the largest discrepancy in R^2 for the two methods (0.7292 vs 0.0855) given by Run C10 for $\alpha = 0.01$.

Shrewd readers will notice the negative values of R^2, and wonder how such values can occur. For a value of R^2 to be negative SSE must be greater than SST. This can happen if the trend line is horizontal, and the data points in the interval are all above or below the trend line, a situation which can be readily demonstrated with the SDT algorithm. In fact, this is a critical flaw in the SDT algorithm. It can be eliminated by applying LS regression to all the data

Table 4. Industrial plant data using PLOT

Run	Method	α	SSE	R^2	Intervals	ρ	Outliers detected	CPU (s)
C1	PLOT	0.01	3.9363	0.9708	96	15.5	0	14.6
	PLOT	0.05	0.0037	1.0000	738	2.0	7	3.0
	PLOT	0.1	0.0025	1.0000	759	2.0	7	3.1
C2	PLOT	0.01	0.006	1.0000	429	3.5	0	3.0
	PLOT	0.05	0.0012	1.0000	452	3.3	0	2.9
	PLOT	0.1	0.0012	1.0000	454	3.3	1	2.9
C3	PLOT	0.01	155.3	0.9985	110	13.6	3	7.9
	PLOT	0.05	23.5	0.9998	330	4.5	18	5.1
	PLOT	0.1	8.5	0.9999	571	2.6	39	4.8
C4	PLOT	0.01	0.0002	0.9863	9	160.4	1	64.3
	PLOT	0.05	0.0001	0.9952	15	97.6	2	36.6
	PLOT	0.1	0.0001	0.9964	17	86.4	1	34.1
C5	PLOT	0.01	3281.5	0.9569	32	46.3	14	46.6
	PLOT	0.05	686.2	0.9909	155	9.6	60	10.5
	PLOT	0.1	404.4	0.9947	295	5.1	111	7.1
C6	PLOT	0.01	7728.8	0.9706	43	34.5	2	22.7
	PLOT	0.05	596.4	0.9977	469	3.2	22	4.8
	PLOT	0.1	209.1	0.9992	767	2.0	44	4.5
C7	PLOT	0.01	284.9	0.9990	69	21.6	13	13.2
	PLOT	0.05	107.9	0.9996	222	6.7	47	7.0
	PLOT	0.1	51.8	0.9998	431	3.5	111	5.7
C8	PLOT	0.01	999.2	0.9792	37	40.1	3	19.5
	PLOT	0.05	77.7	0.9984	343	4.4	10	5.0
	PLOT	0.1	27.6	0.9994	584	2.6	22	4.5
C9	PLOT	0.01	115.5	0.9656	45	33.0	2	21.5
	PLOT	0.05	24.7	0.9926	246	6.1	24	6.3
	PLOT	0.1	11.7	0.9965	426	3.5	47	5.4
C10	PLOT	0.01	9942.9	0.7292	42	35.4	1	22.7
	PLOT	0.05	2593.8	0.9294	174	8.6	2	6.5
	PLOT	0.1	47.2	0.9987	930	1.6	32	4.2

Table 5. Industrial plant data using SDT

Run	Method	E	SSE	R^2	Intervals	ρ
C1	SDT	0.0716	8.3614	0.9381	96	15.5
	SDT	0.0039	0.0032	1.0000	737	2.0
	SDT	0.0031	0.0014	1.0000	759	2.0
C2	SDT	0.005	0.001	1.0000	429	3.5
	SDT	0.0018	0.0002	1.0000	452	3.3
	SDT	0.0017	0.0002	1.0000	454	3.3
C3	SDT	0.5254	764.6	0.9928	110	13.6
	SDT	0.1839	77.4	0.9993	330	4.5
	SDT	0.1057	22.9	0.9998	570	2.6
C4	SDT	0.0006	0.0008	0.9462	9	160.4
	SDT	0.0004	0.0003	0.9833	15	97.6
	SDT	0.0003	0.0002	0.9845	17	86.4
C5	SDT	3.0443	11494.0	0.8508	32	46.3
	SDT	1.1589	1691.7	0.9780	155	9.6
	SDT	0.8340	1050.0	0.9864	295	5.1
C6	SDT	3.7882	27854.1	0.8942	43	34.5
	SDT	0.8717	1650.0	0.9937	469	3.2
	SDT	0.4870	443.5	0.9983	768	2.0
C7	SDT	0.7723	978.0	0.9967	70	21.3
	SDT	0.4096	288.9	0.9990	222	6.7
	SDT	0.2653	115.7	0.9996	435	3.4
C8	SDT	1.3117	3717.2	0.9229	37	40.1
	SDT	0.3226	233.4	0.9952	343	4.4
	SDT	0.1882	75.7	0.9984	584	2.6
C9	SDT	0.5182	486.8	0.8549	45	33.0
	SDT	0.1874	71.2	0.9788	246	6.1
	SDT	0.1259	33.5	0.9900	427	3.5
C10	SDT	4.1436	33586.6	0.0855	45	33.0
	SDT	1.8375	8780.9	0.7609	174	8.6
	SDT	0.2363	97.7	0.9973	930	1.6

points in each interval once the interval is determined.

Another interesting point is the sensitivity of SDT to the assumed value of E. Figure 4 shows trend lines obtained for two different values of E for the same run 12 (details omitted for brevity). The results can be strikingly different, and no guideline is given in the literature for picking a value for E. With PLOT the standard deviation σ is a measure of process variability and is estimated directly from process data. Moreover, unlike SDT, the outliers are recognized and stored separately. They are left out of further calculations by PLOT. One consequence is that the performance of PLOT is much more robust. Inability to cope with outliers can also give rise to negative values of R^2 for SDT.

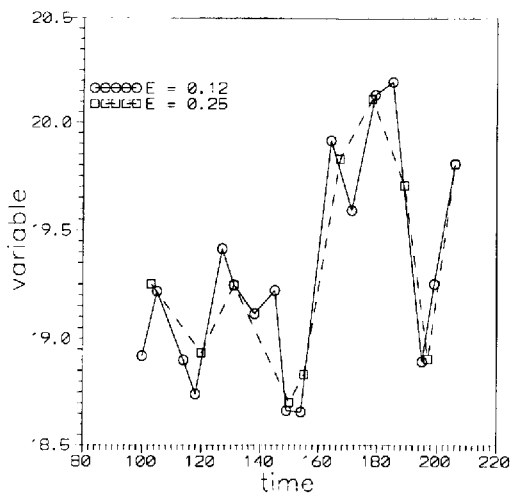

Fig. 4. Effect of E on the trends obtained using the SDT method.

Normally, one would expect the number of outliers declared by the algorithm to increase with the increase in α. However, occasionally this is not true for closely packed data points, especially, if there is a regular pattern, such as step changes. See, for instance, Run C4.

Two other observations should be made before we close. First, although we developed the PLOT algorithm on the basis of constant variance, a slight modification may be made to allow for slow drift or change of its value. Instead of pooling the data points collected up to the current sampling period to compute the variance, we could limit the pooling to a certain number of preceding sampling periods or trend intervals. In this way we can use a moving window to capture a gradual change of process variability. Second, the value of α, the level of significance, may be adjusted to achieve the desired level of data compression ratio. A small value of α would give rise to large prediction intervals, and a tendency to ignore the details. A large value of α would provide the details needed in short term analysis and applications. This is a very powerful tool which allows a user to focus the data to be displayed and analysed on a scale of his own choice

while maintaining a statistical consistency. Typical values of α would range from 0.01 to 0.1.

PLOT is designed to operate online, but is also readily applicable to batch (historical) data. For online applications computing time requirement is an important consideration. The exact computing time depends on the number of outliers as well as the number of data points. For Runs C1 through C10 computing times ranged from 3 to 64 CPU s to process 4500 data points on a PC 486/33 with a math coprocessor. These values are tabulated in the last column of Table 4. The average CPU time per data point is only in the order of a millisecond, which is much smaller than the data sampling period. It is expected that this computing time can be further reduced, if the computation is streamlined.

CONCLUSIONS

A new computing method is proposed for online data compression and trend recording. The method, Piecewise Linear Online Trending (PLOT), has been implemented on a Sun Workstation and on a PC, and evaluated using synthetic and industrial plant data. It performs significantly better than the two existing methds which are commonly used. It is based on statistics, and is readily applied sequentially to real time data as well as batch historical data. Unlike these methods, it adapts to process variability and noisy data, recognizes and eliminates outliers, and is robust even in the presence of outliers. While it is slower than the SDT algorithm, its computing time requirement is easily accommodated in process applications. In addition to the trend parameters, outliers and process data variability (in terms of σ) are also readily stored and available for subsequent analysis.

NOMENCLATURE

H = Number of sampling periods in the time horizon under consideration in computing the data compression ratio
i = Generic index with no fixed specific connotation
j = Index for trending interval
J = Number of trends or trend intervals in the time horizon under consideration in computing the data compression ratio
MSE = Mean square error, or the ratio of SSE over df
n = Index used to keep track of the number of data points in a trend interval
N = Generic index for the number of data points in computing prediction interval for y
R^2 = Coefficient of multiple determination, i.e. $1 - \text{SSE/SST}$

$$\text{SSE} = \text{Error sum of squares} = \sum_i (y_i - \hat{y}_i)^2$$

$$\text{SST} = \text{Total sum of squares} = \sum_i (y_i - \bar{y})^2$$

t = Index for sampling period in the time series for process data
x = Generic index for sampling period in the regression of y vs x
y = Process data measurement
$\hat{y}$ = Fitted data value or estimate of y
$\bar{y}$ = Sample average of observed values y

Greek letters

α = Level of significance
ε = Measurement error
μ = Expected value of y
ν = Degrees of freedom
σ = Standard deviation of measurement error

Acknowledgements—The authors thank Drs Norm Jerome and James Gerstle for helpful discussions.

REFERENCES

Bristol E. H., Swinging door trending: adaptive trend recording? *ISA National Conf. Proc.*, pp. 749–753 (1990).

DeHeer L. E., Plant-scale process monitoring and control systems: eighteen years and counting. *Foundations of Computer-Aided Process Operations*. CACHE/Elsevier (1987).

Hale J. C. and H. L. Sellars, Historical data recording for process computers. *Chem. Engng Prog.* pp. 38–43 (1981).

Kennedy J. P., Data treatment and applications—future of the desktop. *Proceedings of Foundations of Computer-Aided Process Operations* (1993).

Langen P. A., The use of historical data storage and retrieval systems at nuclear power plants. *IEEE Trans. Nuclear Sci.* **NS-31**, 844–848 (1984).