A Novel Interval-Halving Framework For Automated Identification of Process Trends

Sourabh Dash, Mano Ram Maurya, and Venkat Venkatasubramanian
School of Chemical Engineering, Purdue University, West Lafayette, IN 47907

Raghunathan Rengaswamy
Dept. of Chemical Engineering, Clarkson University, Potsdam, NY 13699

Qualitative process trend representation is an useful approach to model the temporal evolution of sensor data and has been applied in areas such as process monitoring, data compression, and fault diagnosis. However, the sheer volume of real-time sensor data that needs to be processed necessitates an automated *approach for trend extraction. The step of recovering important temporal features is a difficult procedure to automate because of the absence of a priori knowledge about the sensor trend characteristics such as noise and varying scales of evolution. A novel approach is proposed to automatically identify the qualitative shapes of sensor trends using a polynomial-fit based interval-halving technique. To estimate the significance of fit-error, an estimate of the noise obtained from wavelet-based denoising is used. The procedure identifies the qualitative trend as a sequence of piecewise unimodals or quadratic segments. The least-order (among constant, first-order and quadratic) polynomial with fit-error statistically insignificant compared to noise (as dictated by F-test) is used to represent the segment. If the fit-error is large even for the quadratic polynomial, then the length is* halved *and the process is repeated on the first half segment until fit-error is acceptable. A constrained polynomial fit is used to ensure the continuity of the fitted data and an outlier detection methodology is used to detect any jump (step) changes in the signal. The whole procedure is recursively applied to the remaining data until the entire data record is covered. Finally, a unique assignment of qualitative shape is made to each of the identified segments. The application of the interval-halving technique for trend extraction is illustrated on a variety of both simulated and industrial data.* © 2004 American Institute of Chemical Engineers *AIChE J,* 50: 149–162, 2004

Keywords: qualitative trend analysis, trend extraction, wavelet-based denoising, interval-halving, process monitoring, data compression

Introduction

Process data contain valuable information about the state, operation, and behavior of the process plant, more so in cases with limited available process knowledge. In most of the cases, only lumped parameter models are available for certain sections of the plant. Increased automation and faster sampling rates have made large volumes of precise data collection (so that the data captures important small time-scale events such as inverse response over a short period of time, and so on) and rapid access from electronic devices possible (Kennedy, 1993). The problem of interpretation, that is, extraction of meaningful information, however, has largely been left to the operator, who usually suffers from information overload. The potential uses of data are numerous (Mah et al., 1995; Rengaswamy et

Correspondence concerning this article should be addressed to R. Rengaswamy at raghu@clarkson.edu and V. Venkatasubramanian at venkat@ecn.purdue.edu.
Current address of S. Dash: ExxonMobil Research & Engineering Company, Fairfax, VA 22037.
Current address of M. R. Maurya: San Diego Supercomputer Center, MC0505, 9500 Gilman Drive, La Jolla, CA 92093.

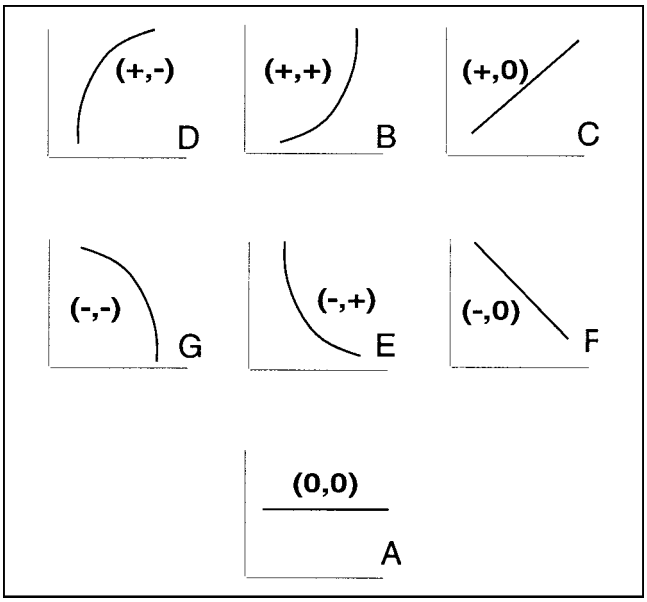

Figure 1. Fundamental language: primitives.

al., 2001): detection and diagnosis of faults (Bakshi and Stephanopoulos, 1994b; Davis et al., 1995; Misra et al., 2002), adaptive control (Najim and Saad, 1991), product quality control (Muske et al., 1991; Yabuki et al., 2002), and operator training (Sebzalli et al., 2000) to name a few. This motivates development of methods which extract useful and relevant information from sensor data.

Process trend analysis is an useful approach to exploit the temporal information and reason about process state. The main activities involved in such an analysis are: (a) a technique to identify trends and, (b) a mapping from trends to operational conditions. While formal trend representation schemes have been proposed (Cheung and Stephanopoulos, 1990; Janusz and Venkatasubramanian, 1991), the question of automatically and correctly identifying such features still remains to be addressed comprehensively. A brief review of the previous work in this area is presented below (see Maurya (2003) and Dash et al. (2003a) for a detailed review).

Cheung and Stephanopoulos (1990) have proposed a formal methodology to transform the measured signals into a qualitative representation consisting of "triangular" episodes. This representation was later used by Bakshi and Stephanopoulos (1994a) to extract trends based on multiscale analysis using wavelets. Janusz and Venkatasubramanian (1991) developed a "minimal" qualitative representation scheme (language) whose fundamental elements are called *primitives* (Figure 1). Figure 1 shows the shapes of the seven primitives *viz.* **A**(0, 0), **B**(+, +), **C**(+, 0), **D**(+, −), **E**(−, +), **F**(−, 0), **G**(−, −) where the signs are of the first and second derivatives, respectively. A trend is represented as a sequence (combination) of these seven primitives. Rengaswamy and Venkatasubramanian (1995) and Rengaswamy et al. (2001) extended this method using syntactic pattern recognition involving grammar-based error correction to rectify inconsistencies. The primitives were identified from the sensor data using a neural network. A fixed *window size* (the number of nodes in the input layer which is the same as the number of samples used to identify the primitive) was used in this work.

Konstantinov and Yoshida (1992) proposed a qualitative analysis procedure with the help of an expandable "composite" shape library. The identified features are compared against the shape descriptors in the library for classification. The time-scale of analysis is fixed *a priori* which limits its generic applicability. Also, with increasingly complex shapes, the library can become quite big and the simple reasoning based on the extent of derivatives' sign match may not be suitable. Whiteley and Davis (1992) present a knowledge-based interpretation of sensor patterns. Mah et al. (1995) developed a technique for data compression and trending called piecewise linear online trending (PLOT).

Vedam (1999) presented a B-Spline based technique for data compression and automatic trend extraction which avoids the use of neural networks. Although the technique is very good in capturing important features automatically, it suffers from the need to tune a number of parameters such as feature threshold (during trend extraction), and thresholds on window size and magnitude (during event detection and diagnosis) for each sensor. Recently, Dash et al. (2003b) have presented a fuzzy-logic-based multivariate inferencing framework for temporal-reasoning. In this work, the primitive-based language (Janusz and Venkatasubramanian, 1991; Rengaswamy and Venkatasubramanian, 1995) is used as a basis for process trend analysis, and it is assumed that the qualitative trends can be automatically identified.

From the above review, it is clear that various trend extraction methodologies discussed in the literature are far from being comprehensive. In this article, a novel approach to automate the identification of process trends based on an *interval-halving* procedure is proposed. The idea is to parameterize the data as a sequence of primitives by using the goodness-of-fit determined by comparing the fit-error with noise (Dash et al., 2001; Dash, 2001). An estimate of noise level can be obtained using wavelet analysis. The structure of the article is as follows. The main issues involved in the problem of trend identification are discussed in the next section. Thereafter, the main contribution, the interval-halving framework, is discussed. In the following section, a number of metrics have been defined which are used to test the effectiveness of the methodology. Then, a number of examples (both simulated and real-data) are presented to elucidate and evaluate the methodology. Finally, conclusions are presented.

Issues in Process Trend Identification

The temporal behavior of measured variables in a chemical process is the superposition of many underlying driving processes such as process dynamics, sensor noise, faults, external loads, disturbances, and so on, evolving at different time-scales (Bakshi and Stephanopoulos, 1994a). This disparity in rates of fault evolution is the most important issue in trend identification. This emphasizes the need for the automated identification method to be extremely robust while using minimal or no *a priori* information. The presence of sensor noise further complicates the task. Clearly, the problem of trend identification is a difficult task. Some of the important issues are briefly discussed below (see Dash et al. (2003a) for a detailed discussion):

- **Time-scale of identification:** The scales (rates) at which real trends evolve vary considerably from slow to extremely

fast depending on the underlying driving event. Hence, any effective technique must be adaptive to the time scale of trend evolution as the window cannot be fixed *a priori.* The window should be wide enough to observe significant variations in the monitored signal and it should be small enough so that a single primitive can fit the data well.

- **Noise:** Sensor data are invariably corrupted with measurement noise. In most practical cases, knowledge of the noise characteristics is minimal. The amount of noise (signal to noise ratio (SNR)) is an important criteria in deciding the usefulness of a sensor from trend analysis perspective.
- **Scale variant nature:** The representations of qualitative shapes depend on the scale at which they are observed. A careful choice of window width which allows proper detection of concavity and convexity in process trends is required.
- **Simplicity and computational complexity:** Trends are visual features. Hence, simplicity in extraction and reasoning, while not compromising on accuracy and applicability, is highly desirable. Also, since most of the intended applications are real-time, the computational complexity of the algorithm should not be prohibitive to restrict its usefulness.

An Interval-Halving Algorithm for Trend Extraction

In this section the algorithm to *automatically* extract the primitives (Figure 1) from data is described. First, the basic idea is presented and then the details of the technique are discussed.

Basic idea

To be able to extract trends from data in terms of the primitives, the raw signal should be transformed into a form which would facilitate conversion into the symbolic language of primitives. A functional form is associated to the data so as to extract the symbolic representation. The simplest approach is to fit polynomials to the data. The shapes of the primitives, by definition, are characterized by constant signs of the first and second derivatives. Thus, if a sequence of consecutive segments with a fixed sign of the first and second derivatives (at every point in a segment) can be identified, then the assignment of primitives would be straightforward. All the discussion in this section is valid for both continuous functional representation of a signal, as well as sampled (discrete) data (not necessarily sampled at an uniform rate). Also, discrete data is usually denoted as $\boldsymbol{y}$ and y_i. The requirement on the first derivative can be relaxed, that is, allow it to take different signs at the ends while retaining the second derivative constraint. A function which satisfies this criteria is the *unimodal* function.

A function $g(x)$ is unimodal on the interval $a \leq x \leq b$ if and only if it is monotonic on either side of the single optimal point (extremum) x^* in the interval. In other words, this implies that x^* is the single extremum point of $g(x)$ in $a \leq x \leq b$.

A unimodal region involving primitives has a constant sign for second derivative while the first derivative sign could change, that is, be opposite at both ends. Any reasonably continuous and smooth signal can be observed as a sequence of unimodal regions (see Dash et al. (2003a) for an illustrative example). While the simple unimodals that is, those with constant first and second derivative signs, can be directly assigned primitives, for the composite shapes, that is, those with constant second derivative sign but changing first derivative sign, primitives can still be assigned by splitting them at the zero first derivative point and identifying the two simple pieces separately. Each of the identified unimodal regions may also be characterized by zero-crossings (Witkin, 1983; Bakshi and Stephanopoulos, 1994a). The n^{th} order zero crossings in a signal $y(t)$ are given by points that satisfy

$$\frac{\partial^k y}{\partial t^k} = 0 \; (\; \forall \; k = 1, 2, \ldots, n), \; \frac{\partial^{n+1} y}{\partial t^{n+1}} \neq 0$$

that is, $(n + 1)^{\text{th}}$ derivative is the first nonzero derivative. These locations usually correspond to the extrema (odd n) and inflexion points (even n) in the signal. For unimodal functions, odd order zero crossing ($n = 1, 3$, and so on) corresponds to a point of extremum (no inflexion points).

The objective here is to extract out the piecewise unimodals from the data record as the first step. In each of the unimodal regions the data could be approximated using a polynomial, based on which the shape can then be identified. The problem, of course, lies in identifying these crucial segments automatically. The basic motivating idea is that, if the function is smooth, it can then be approximated by a polynomial, and the approximating polynomial can then replace that portion of the data segment for the purpose of trend extraction. The Weierstrass approximation theorem (Bartle, 1976) guarantees that, if a function is continuous on the interval, then it can be approximated as closely as desired by polynomials of sufficiently high order. Consequently, if the function is unimodal and a reasonably good approximating polynomial is at hand, then the functional behavior can be reasonably well predicted by the polynomial. Improvements in the fit can be obtained through approximating polynomials in two ways: by using a higher-order polynomial or by reducing the interval over which the function is to be approximated. Of these, the second alternative is generally to be preferred, because the polynomial-fitting algebra becomes more complicated compared to the interval reduction which is rather easily accomplished. Hence, the idea of halving the interval. For the present application, given the requirement on the signs of the derivatives mentioned above, the quadratic is the simplest and the most obvious choice as the approximating polynomial. To determine the time segments for identification, the interval-halving technique, described next, is used. As will be seen, the location of the piecewise unimodals is driven by the polynomial fit itself and, thus, the whole procedure of (a) identifying these regions and (b) fitting polynomials is condensed to a single polynomial fit-error driven unimodal regions location scheme. The standard least-squares technique is used to identify the polynomial fit. The fitted data in two consecutive unimodal segments identified through the standard least-squares formulation need not be continuous. One way to circumvent this problem is to consider the average y value at the intersection of the two segments, but it might be meaningless for large discontinuities. Hence, once a unimodal segment is identified, a constrained least-squares (polynomial fit) approach is used to fine-tune the fitted polynomials (both order and the coefficients) to ensure continuity of the fitted data (or primitives in every two consecutive unimodal segments). Since the constrained least-squares fit results in strict continuity, to preserve jump (step) changes that are present in the

original signal itself, the jump changes (if any) at both the ends of the unimodal segment are identified by using an outlier identification methodology. The discussion on the interval-halving framework is presented below.

Interval-halving framework

The idea of interval halving has been utilized in many applications. In general, the method is a region-elimination strategy and aspires to remove exactly half of the interval, that is, region not containing the desired item or satisfying a criterion at each stage. It has found use in various forms in (i) optimization: bisection searches for maxima/minima in unimodal functions (Peters and Timmerhaus, 1990), searches for sorting algorithms, nonlinear optimization (Krongold et al., 2000), (ii) numerical methods: bisection method for root-finding algorithms (Peters and Timmerhaus, 1990), initial value guess for the iterative initial value approach in solving ODEs (ordinary differential equations) involving 2 point BVPs (boundary value problems) (Jimenez et al., 1998), extended trapezoidal rule for numerical integration, and so on. The method is computationally efficient, intuitive, and simple. It has been shown by Kiefer (1957) that, out of all equal-interval searches (two-point, three-point, four-point, and so on), the three point search or interval-halving is the most efficient. Paritosh and Rengaswamy (1999) have proposed and discussed the various issues in the use of interval-halving technique for qualitative process trend identification. In this section, the interval halving algorithm for trend identification is discussed. The objective here is to automatically identify the trends using an adaptive approach. As discussed above then:

Any time-series function $y(t)$ can be approximated to any level of detail using a polynomial of certain order n. Moreover, $y(t)$ can be represented arbitrarily closely using a sequence of piecewise polynomials $p_i(t)$ over the unimodal regions U_i each of order $n_i \leq n$ that is, $y(t) \approx \{p_1(t), p_2(t), \ldots, p_M(t)\}$.

Thus the identification of the unimodal regions U_i can be equivalently posed as the location, that is, the start and end points and specification, that is, the polynomial coefficients of these p_i (quadratics). The algorithm consists of two parts: (i) determining the sequence of the p_i, that is, U_i using the interval-halving procedure, and (ii) assigning primitives to these identified U_i based on the signs of the derivatives of p_i. The standard least-squares result is used to fit a polynomial p_i with coefficients $\hat{\boldsymbol{\beta}} = [\hat{\beta}_0 \cdots \hat{\beta}_n]$ to the data

$$\mathbf{y} = \mathbf{T}\boldsymbol{\beta} + \mathbf{e}, \hat{\mathbf{y}} = \mathbf{T}\hat{\boldsymbol{\beta}}$$

$$p_i(t) = \sum_{k=0}^{k=n\leq 2} \hat{\beta}_k t^k \quad \text{where } \hat{\boldsymbol{\beta}} = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{y} \tag{1}$$

The $(j,\ k)^{\text{th}}$ element of $\boldsymbol{T}$, $(\boldsymbol{T})_{jk} = t_j^{k-1}$. The identification window is normalized to [0 1] to avoid ill conditioning of $\boldsymbol{T}$. To determine the goodness of fit the estimate of noise provided by the wavelet analysis (see appendix) is used. The significance of the fit-error ϵ_{fit}^2 is tested against the estimated noise variance σ_{noise}^2. Once a unimodal segment (and the corresponding $p_i(t)$) is identified, jump changes at the ends of the current unimodal segment are identified (see Dash et al., 2003a) for a detailed discussion) and a constrained least-squares fit is used over the current segment and the previous segment to refine the polynomial coefficients (including the order) for both the segments. Next, the details of both the parts are discussed. A review of the material on F-test, t-test, identification of jump changes presented by Dash et al. (2003a) and constrained polynomial fit (see appendix) would help the reader understand the rest of this section.

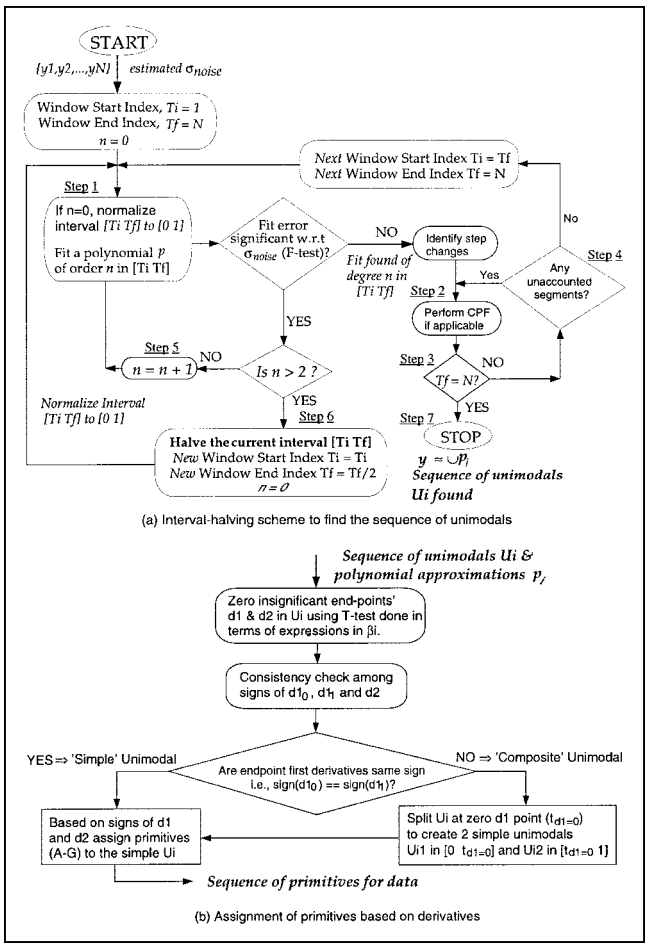

Figure 2. Two-part interval-halving scheme.

Polynomial fits—Identification of unimodals. Assume that the data $y(t)$ are given as a sequence of uniformly sampled data points $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$. Set start time $T_i = 1$ and end time $T_f = N$. Set the initial window length $l = N$ and the polynomial order $n = 0$. In the steps discussed below, the subscripts on y refer to the indices of the data points in the identification window (and not with respect to the whole signal). Set the threshold on the length of a segment l_{th} to some small number, say 10. The procedure is illustrated in the flowchart shown in Figure 2a.

(1) *Polynomial fit:* If $n = 0$, normalize the window of identification $W_{id} = [T_i\ T_f]$ to [0 1] (else it is already normalized). Fit polynomial of order n, p_n to $y(t)$ $(T_i \leq t \leq T_f)$ in normalized $[T_i\ T_f]$, that is, [0 1]. Calculate fit error ϵ_{fit}^2 as

$$\epsilon_{\text{fit}}^2 = \frac{1}{\nu_1}\sum_{i=1}^{i=l}(y_i - p_{n_i})^2 \tag{2}$$

where $\nu_1 = l - (n + 1)$ (window length − number of coefficients) is the degrees of freedom (DOF). To consider the significance of the error a statistical hypothesis test is carried out using *F-test* (Dash et al., 2003a) since the elements of the error vector $\boldsymbol{e}$ are assumed to be IID (independent and identically distributed), that is, they form a white-noise sequence. If the null hypothesis (H_0) is accepted, then the fit-error is insignificant. Use the jump identification methodology (Dash et al., 2003a) to identify jump changes at the start and the end of the segment. Go to Step 2. Else (that is, H_0 is rejected) the fit-error is significant. Go to Step 5.

(2) *Constrained polynomial fit (CPF):* If $T_i = 1$ (that is, the very first segment of the entire data record), then CPF cannot be performed yet. Go to Step 3 with the remaining data. Else, proceed with CPF (details presented in the Appendix) over the current segment (call it segment 2) and the previous segment (call it segment 1) as follows. The basic idea is to let the order of the polynomial for segment 1 n_1 increase and perform CPF. To provide maximum flexibility in segment 1, the order of the polynomial for segment 2, $n_2 = 2$ (a quadratic) except when $T_f = N$ (that is, it is the very last segment, in which case, the fit corresponding to its actual order is desired (since no new segments are to be identified) and, hence, n_2 is kept equal to the order identified in Step 1). n_1 is kept at its previous value obtained in Step 1. $n_{1,\max}$ and $n_{2,\max}$ are set to 2 or 1 depending upon whether or not the length of the corresponding segments are above l_{th}. The recursive procedure is:

(a) Perform CPF with current values of n_1 and n_2. If F-test is satisfied for segment 1, go to Step 2b. Else go to Step 2c.

(b) If F-test is satisfied for segment 2, quit CPF. Else go to Step 2d.

(c) If $n_1 < n_{1,\max}$, set $n_1 = n_1 + 1$ and go to Step 2a. Else go to Step 2b.

(d) If $n_2 < n_{2,\max}$, set $n_2 = n_2 + 1$ and go to Step 2a. Else quit CPF.

One can verify that Steps 2b and 2d are executed more than once only if segment 2 is the last segment. A minor variation of the above procedure could be to set $n_{1,\max}$ equal to the original value of n_1 (identified in Step 1) so that the fit order would not be allowed to increase and the Steps 2a–2d would be executed only once.

The useful quantities to be retained from CPF are n_1 and n_2, the polynomial coefficient vectors $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$, the Lagrange multiplier(s) λ (and μ), the fitted data for the two segments, and the covariance matrices of $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$ (see appendix). Store the updated information for segment 1. If segment 2 is the last segment of the data record, store the updated information for segment 2 as well, else, store the original order of segment 2 (identified in Step 1, not the returned value of n_2 from CPF) and the fitted value (after CPF) at the intersection, that is, y_1 for segment 2 (this value would act as d_0 (see appendix) during the CPF for the next segment). Go to Step 3.

(3) If $T_f = N$ then the whole data record length N of $\boldsymbol{y}$ is covered, go to Step 7. Else go to Step 4.

(4) *Precondition for identification of a new unimodal segment:* If there is a segment over which CPF has not been performed even once, mark it as the current segment and go to Step 2. Else go to Step 1 with the remaining time segment $[T_f \ T_N]$ and $n = 0$ (identify a new unimodal segment).

(5) *Increasing the order of the polynomial:* If $l \leq l_{th}$, if $n = 1$ (linear), stop refining the current segment, accept the linear fit (assume that F-test is satisfied) and go to Step 2. The rationale behind l_{th} is that if the window does fall to such a small number, the current polynomial fit is accepted and the interval is not refined any further. If $l > l_{th}$, if $n < 2$, set $n = n + 1$ and go to Step 1, else go to Step 6.

(6) *Interval-halving step:* Split the current interval $W_{id} = [T_i \ T_f]$ at the *midpoint,* that is, halve the interval and assign $T_{\text{half}} = T_f/2$, $W_{id} = [T_i \ T_{\text{half}}]$. If desired, estimate σ_{noise} for the current segment. Go to Step 1 with the data segment in $[T_i \ T_{\text{half}}]$ and $n = 0$.

(7) The data $\boldsymbol{y}$ has thus been completely parameterized as a sequence of piecewise quadratics p_i, that is, $f \approx \cup_{i=1}^{M} p_i$. Thus, M unimodal regions U_i have been extracted. The regions are unimodal by definition as quadratics can only have a single extremum and the sign conditions on the derivatives are also satisfied: $d^2p_i/dt^2 = 2\hat{\beta}_2$ (constant sign) while $dp_i/dt = \hat{\beta}_1 + 2\hat{\beta}_2 t$. Quit.

In the above procedure, by the way of implementation, l_{th} equals twice the minimum window size desired. Thus, when the window size falls between $l_{th}/2$ and l_{th}, further halving is stopped. Next, the second part is presented.

Assignment of Shapes: Identification of Primitives. The second part comprises of assigning primitive shapes to the piecewise polynomials. The procedure above guarantees that *unique* assignment of primitives is possible to each of the identified unimodals (quadratics). It should be noted that the sequence/location of the unimodals is not unique, that is, the trend identification by using another methodology or visual inspection (manual identification) might identify slightly different locations and/or sequence. Similarly, the trends extracted from a noisy signal corresponding to different realization of the noise sequence (no change in true signal) may differ from each other with respect to location and shape of the primitives, but these differences are minor and are acceptable for most of the applications that rely on the use of qualitative trends (Dash et al., 2003b).

The labeling of shapes is based on the sign of the first derivative ($d1_t$) and second derivative ($d2_t$). First, one needs to examine if the unimodals identified are simple, that is, $\text{sign}(d1_{t=0}) = \text{sign}(d1_{t=1})$ as in the seven primitives or composite, that is, $\text{sign}(d1_{t=0}) \neq \text{sign}(d1_{t=1})$. To examine the significance of the end-derivatives, that is, whether $d1_{t=0,1} \approx 0$ (ensures robustness), a statistical test is carried out on the coefficients $\hat{\beta}_i$ with variances s_i^2. The entire procedure is illustrated in Figure 2b. It consists of the following:

(1) *Significance of derivatives:* Since derivatives will not be identically zero, the significance of the derivatives is examined using the *t-test* (Dash et al., 2003a) to compensate for the effect of noise. Only those derivatives should be assigned a non-zero value which are statistically significant.

(2) *Assignment of primitives:* For $i = 1, \ldots, M$ check if the signs of the end derivatives in U_i match, that is, if $\text{sign}(d1_{0,\text{new}}) = \text{sign}(d1_{1,\text{new}})$?. If they match this segment is a simple unimodal region, else it is a composite region implying that the identified quadratic p_i in U_i is composed of 2 primitives (**EB** or **DG**). Next, find the zero-derivative point, that is, $\{t_{d1=0} | dp_i/dt|_{t=t_{d1=0}} = 0\}$ and split the composite U_i into to 2 simple regions $\{[0 \ t_{d1=0}], [t_{d1=0} \ 1]\}$. Thus, a sequence of simple unimodal regions is generated and the primitive assignment is carried out in a straightforward manner using the derivative signs (see Figure 1).

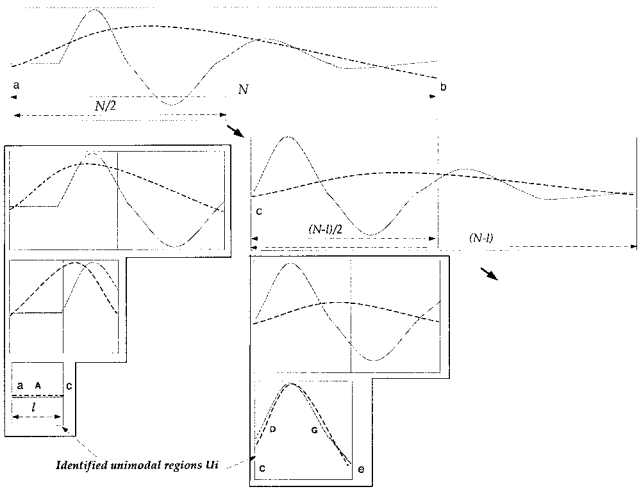

Figure 3. Interval-halving.

Figure 3 illustrates the idea of interval-halving and assignment of shapes on a sample data. As shown, the data length is recursively halved until a unimodal region (quadratic) with acceptable error is found. For example, the first unimodal region (segment ac, primitive **A**) has a span of $N/8$ ($= l$). This procedure is repeated on the remaining data (such as on the segment of length ($N - l$) after the first unimodal is identified) until the whole data record is covered. The next unimodal segment is ce (the subsequence **DG**). It is a composite shape since the slopes at the two ends are opposite. It is easy to see that, within a finite number of iterations, this interval-halving procedure on an interval I will terminate, that is, the method will not get into an infinite loop. This is because there always exists an interval $[a\ \ b] \subset I$ which will be encountered during this process wherein either (a) the *F-test* succeeds and the search ends or (b) if the window becomes too narrow (size less than l_{th}) then the search is forcibly terminated. The search in this procedure is for the features that characterize the unimodals (extrema, inflection points, and so on), along the same lines as the usual interval-halving searches.

Selection of Various Parameters. The various parameters used in the interval-halving framework for trend identification are discussed below. Some guidelines for tuning these parameters also are presented.

- *The wavelet and the decomposition level (number of scales) in wavelet-based denoising:* For all the case studies, *db3 wavelet* (order 3 wavelet from the Daubechies orthogonal wavelet family) has been used (see appendix). In the case studies presented, the maximum decomposition level (automatically estimated in Matlab) is used for single scale-factor based thresholding. When level dependent scale factor is used for thresholding the coefficients, the following rule is used to set the decomposition level (l is the length of the data set to be denoised and J is the decomposition level). If $l \geq 512$, $J = 5$, else if $l \geq 128$, $J = 4$, else if $l \geq 64$, $J = 3$.

In general, single scale-factor is used for thresholding the wavelet coefficients when white noise is present in the signal, whereas level-dependent scale-factor is used when autocorrelated noise is present. If the magnitude of noise varies across different parts of the signal, then noise is re-estimated in various segments before polynomial fit is carried out in Step 1 until the window size becomes small (see below).

- *Minimum window length for noise estimation:* Usually, at least 64 (2^k, where k is an integer) data points are used for wavelet-based denoising. In all the case studies presented here, this threshold has been set to 128.
- *Window size threshold (l_{th}):* l_{th} is used to stop interval-halving when window size becomes very small. As discussed in the interval-halving procedure, the window size can never fall below $l_{th}/2$. Recall that $n \leq 1$ for such a small segment (Step 5) since if $n = 2$ results in a composite unimodal, then the primitive length may fall below $l_{th}/2$. This parameter can be kept the same for all sensors with equal sampling rates and can be chosen easily. For example, to extract trends from an industrial data, where sampling rate is one sample per minute and the process evolution time is of the order of hours, l_{th} is kept at 10.

Similar to l_{th}, a cut off value can be chosen for DOF to avoid unrealistic relaxation in F-test or t-test (important when the segment length falls below l_{th}), such as in all the case studies, this cut off value is set to 3 for all the t-tests and the F-test during CPF.

- *Significance level (α) for F-test and t-test (significance of derivatives):* This statistical parameter has been set at 0.05 (corresponds to the widely used 95% confidence level) for all the case studies.
- *Significance level (α) for t-test to impose restriction on constant fit:* As explained in the appendix, $\alpha = 0.80$ for segment 1 and $\alpha = 0.50$ for segment 2. Without much experimentation, $\alpha \geq 0.50$ is recommended.

Summary and Benefits of the Approach. The end result of the entire method is a sequence of shapes (**A–G**) describing the data *qualitatively* as also the piecewise quadratics explaining it *quantitatively* in a succinct manner. There is data compression because of the parameterization (coefficients of the polynomials and end points need only be stored instead of raw data). In addition, denoising is an added benefit as a result of polynomial smoothing and statistical testing. Variations of this basic procedure can be employed for fine-tuning some special cases. The described technique is adaptive to the scales of trend evolution and is robust to process noise. Also, as discussed above, there are only few tuning parameters (quite universal in nature) to manipulate, thus allowing automation and possible explanation generation from the process trends. One important issue that has not been addressed in this article is on-line implementation. A succinct discussion is given below.

One way to implement a trend-extraction or de-noising technique is to move the data window at regular intervals as more data becomes available, perform the trend extraction, and concatenate the trends in the current window with the previously extracted trends. This is called sliding window approach (Vedam, 1999). Another approach for on-line trend extraction is to slide the window at regular intervals, but eject some of the primitives in the current window so that only last few primitives are allowed to evolve (Keogh et al., 2001). In this approach, the window size is not fixed and concatenation is not required. Nonuniformly sampled data and missing samples are some of the other issues that need to be addressed. As such, the methodology presented in this article does not assume that the samples are available at uniform intervals. Thus, both the data and time can be recorded. The only modification that would be

needed is to split the window at the sample that is nearest and on the left side to the middle time point. If the measurements are made at uniform intervals, then some simplification is achieved in various expressions. Huang et al. (2003) have discussed the use of pseudo-measurements for extended Kalman filter (EKF)-based estimation when sampling rate is low. This idea can also be exploited to handle irregular measurements and missing values.

As discussed in the introduction, several applications of process trends have been reported in the literature, such as process monitoring and fault diagnosis (Bakshi and Stephanopoulos, 1994a; Dash et al., 2003b), control loop monitoring (Rengaswamy et al., 2001), and so on. In the next section a number of metrics are defined which are used to evaluate the efficacy of the interval-halving methodology. In the following section, the effectiveness of the interval-halving technique is demonstrated on a variety of simulated noisy and real plant data.

Performance Metrics

To evaluate the effectiveness of the algorithm the following measures are used as performance metrics.

Scaled average global error (SAGE)

This error is calculated between the wavelet estimated denoised data $\hat{f}$ and the piecewise quadratic approximation to the data (fitted data) $f_{\text{fit}} = \cup_{i=1}^{n} p_i$. $\hat{f}$ is used because noise should not be taken into account (it needs to be effectively rejected). SAGE is closely related to root mean squared error (RMSE) which is calculated using the raw noisy data (Keogh et al., 2001) instead of the denoised data

$$\text{SAGE} = \frac{1}{\sigma_{\text{noise}}} \sqrt{\frac{\sum (f_{\text{fit}} - \hat{f})^2}{N}}; \text{RMSE} = \sqrt{\frac{\sum (f_{\text{fit}} - f)^2}{N}}$$

To avoid the effect of the magnitude of noise, σ_{noise} is used for normalization. Thus, a value of SAGE close to 1 indicates that most of the noise has been rejected while retaining the underlying approximately true signal.

Scaled L_∞ error (SLE)

SLE is defined as the normalized (with respect to σ_{noise}) maximum local absolute error between $\hat{f}$ and f_{fit}

$$\text{SLE} = \frac{1}{\sigma_{\text{noise}}} \max|f_{\text{fit}} - \hat{f}|; L_\infty - \text{error} = \max|f_{\text{fit}} - f|$$

Except for the scaling by σ_{noise} and the use of denoised data, SLE is similar to the traditional L_∞-error. RMSE and L_∞-error are used in least-squares approximation and data compression as well (Misra et al., 2001). A large value of SLE indicates that there is at least one point where the fitted signal deviates considerably from the denoised signal. If desired, such regions can be further investigated. Thus, this metric is indicative of the local degradation of the technique and can be regarded as a useful warning, which possibly can be exploited.

Compression ratio (ρ)

Considering the need to efficiently store and retrieve large volumes of data, data compression is an important area of research (Misra et al., 2001; Saxena et al., 2000). As a bonus resulting from the piecewise quadratic representation, data compression is also achieved since the data is now parameterized in terms of the coefficients of the polynomials. For a data record of length N, if the approximation contains M piecewise segments and the number of nonzero coefficients in each is n_i, then this ratio is defined as

$$\rho = \frac{N}{\sum_{i=1}^{M} n_i + M + 1}$$

where the $M + 1$ in the denominator is for the number of end time points information required. In fact, if data compression is not desired, then the use of CPF (which promotes the overall integrity of the interval-halving framework) is not very critical, as the qualitative shapes identified right after the standard F-test based identification of unimodal segments itself qualitatively explain the temporal evolution quite well (Dash et al., 2001).

An important point to note is that, although these metrics give an idea of the performance, they do not give the complete picture from a trend analysis perspective. In general, an accurate trend representation need not be the same as accurate approximation in terms of polynomial fits, for which these metrics are usually used. Thus, the above metrics can be treated as a guiding necessary condition. It is desired that the final temporal behavior captured by the trend representation in terms of features like extrema, inflexion points, and so on be accurate and that is the real criteria in determining efficacy. A single number, as represented by these metrics, cannot condense all such information and, thus, may not be adequate to evaluate effectiveness. One can realize that different representations (shapes) can result in the same mean squared error (MSE) or fit-error (see Dash et al., 2003a). If the acceptable fit-error is guided by the noise content of the signal, then it becomes a fairly sufficient criterion. For this reason, a new measure of success in identifying the extrema, *viz.,* ratio of the number of extrema (RNE) is used. To calculate RNE, it is assumed that an operator can identify all the points of extrema with 100% accuracy. RNE is defined as follows.

Ratio of the number of extrema (RNE)

Let n_a and n_h be the number of extrema identified by the algorithm and an operator, respectively. Let n_a^* be the number of extrema identified by the algorithm which are within l_{th} width (number of samples in between) of the nearest extrema identified by the operator with similar nature (that is, maxima or minima). Then,

$$\text{RNE} = \frac{n_a^*}{n_h}$$

The RNE as defined above is a fairly strict criterion for qualitative similarity of two trends and its value should be very close to 1 for a methodology to be effective. One can note that

Table 1. Functional Forms for Simulated Signals

Case	Functional Form	Data Length
1	Gaussian signal: $y = 20e^{-[(x-\mu)^2/2\sigma^2]}$, ($\mu = 150$, $\sigma = 40$)	300
2	$y = \begin{cases} 80 + 1.2t & \text{if } 1 \le t \le 150 \\ 400 & \text{if } 150 < t \le 300 \end{cases}$	300
3	$y = \text{Case 2} \;\&\; 30 \sin\left(\frac{t}{20}\right) \;\&\; \begin{cases} 40 & \text{if } 50 \le t \le 100 \\ 600 & \text{if } 200 \le t \le 250 \end{cases}$	300

the inflexion points have not been included in the above definition due to two reasons. First, around the point of inflexion, the slope is non-zero and fairly constant in a wide region. Accurate identification of such points from noisy data is quite difficult for any method. Secondly, such points are not very critical for qualitative comparison (Dash et al., 2003a). In the case studies presented in the next section, the metrics SAGE, SLE and ρ are reported for all cases, but RNE is calculated and reported for first simulated signal only.

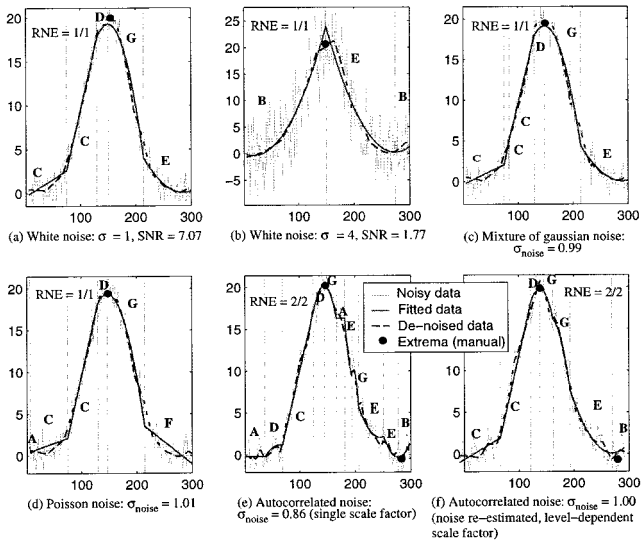

Figure 4. Simulated Case 1: a Gaussian signal.

Case Studies

In this section the application of the above algorithm is demonstrated on a variety of simulated and industrial data to evaluate its efficacy. To estimate the noise in all the signals, the *db3 wavelet* with filter coefficients [0.2352, 0.5706, 0.3252, −0.0955, −0.0604, 0.0249] and soft thresholding is used (see the Appendix). Unless otherwise stated, values of all the parameters are as stated in the section on parameters. Two types of case studies are presented: (i) trend extraction from general simulated data for which the functional form of the true (exact) signal is known, and (ii) trend extraction from the data from a real plant.

Simulated data

Three base case simulated signals of varying complexities are generated. The functional forms of the base case signals are shown in Table 1. Noise (with different characteristics and magnitude) is added to different signals to reflect variety and evaluate the efficacy of the method. To denoise these signals, unless otherwise stated, a single level estimation of noise scale is used and σ_{noise} is calculated only once by using the entire signal. Table 2 shows the amount of added noise (σ) and estimated noise (σ_{noise}) along with the performance metrics for all the signals. σ_{noise} is seen to be close to the simulated noise. A detailed discussion on all the 3 cases is presented next.

Case 1: A Gaussian. This is a simple signal with only few primitives. It is easier to compare the effect of noise characteristics on a simple signal like this. The extracted trends for the signal with a low white noise ($\sigma = 1$, SNR = 7.07) are shown in Figure 4a. The denoised signal and the manually identified extrema are also shown. The manually identified trend itself is not shown since the figure is already crowded. One can see that the fitted data is in good agreement with the denoised data. Further, the point of maxima is identified. Next, some variations of this signal are considered.

Figure 4b shows the Gaussian signal with more noise ($\sigma = 4$). Although the single maximum point has been identified, it is easy to see that some distortion has occurred. Near the maxima, the fitted data deviates considerably from the denoised data. Also, the number of primitives has decreased. Thus, as SNR decreases, performance degradation should be expected. Figure 4c shows the Gaussian signal with a mixture of Gaussian noise added to it (see Dash et al., 2003a) for details of noise generation). Similarly, Figure 4d shows the trends extracted for the Gaussian signal with Poisson noise added to it. As it can be seen in Figures 4c–d, the results are quite similar to the case shown in Figure 4a. Notice that σ_{noise} for all

Table 2. Performance Metrics for Simulated Signals

Case	Fig. no.	Added noise (σ)	Est. noise (σ_{noise})	SAGE	SLE	ρ	RNE
1	4(a)	1.00	0.97	1.05	3.25	20.00	1/1
	4(b)	4.00	3.81	0.99	2.77	33.33	1/1
	4(c)	N/A	0.99	1.11	4.46	16.67	1/1
	4(d)	N/A	1.01	1.14	4.50	17.65	1/1
	4(e)	N/A	0.86	1.16	3.10	9.38	2/2
	4(f)	N/A	1.00	1.14	2.70	15.79	2/2
2	5(a)	4.00	4.70	1.40	13.62	16.67	N/A
	5(b)	16.00	17.78	1.06	4.89	33.33	N/A
	5(c)	64.00	64.31	1.11	3.58	42.86	N/A
3	6(a)	4.00	7.69	5.12	40.93	4.48	N/A
	6(b)	16.00	25.38	1.52	8.07	5.88	N/A
	6(c)	64.00	71.48	1.22	3.40	27.27	N/A

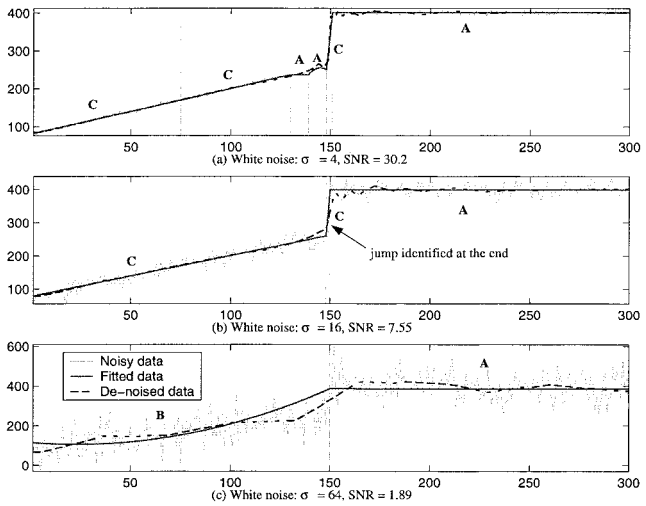

Figure 5. Simulated Case 2: a ramp and step signal.

the three cases (a, c, and d) is close to 1. In Figure 4d, a false **A** primitive (for a short duration only) has been identified by the jump identification procedure at the left end of the signal.

Figure 4e shows the Gaussian corrupted with autocorrelated noise (see Dash et al., 2003a) for parameters). The extracted trends are poor in several regions. Also, a number of primitives are identified. Here, σ_{noise} has been estimated using a single scale factor. So, σ_{noise} is estimated using a level-dependent scale factor which results in larger value of σ_{noise}. Further, σ_{noise} is re-estimated in each segment (if the length criterion is satisfied). The results of trend extraction by using the new noise estimate are shown in Figure 4f. Two extrema points are identified by the interval-halving methodology, as well as manually. It is clear that the performance has improved. Thus, when in doubt about the nature of the noise, it is best to re-estimate σ_{noise} and to use level-dependent scale factor in wavelet-based denoising. As listed in Figures 4a–4f, RNE is 1 for all variants of the Gaussian signal.

Case 2: A Ramp Followed by a Positive Step Change. The purpose of including this example is to show that the framework is able to identify significant jump changes near the ends of the segments. A smaller value of l_{th} (=6) is used. In Figure 5a, the step change is automatically identified since the nearby region in which interval halving took place was small, so F-test was sufficient. Figure 5b shows a scenario where noise is large enough so that F-test does not identify the small segment exhibiting the step change. The jump identification algorithm identifies the step change because it is significant. No jumps are identified in Figure 5c since the step change is not significant (as compared to noise). Certainly, it is true that this step change can be detected manually.

Case 3: Superposition of Case 2, a Sine Signal and Some Steps. As the origin of the signal suggests, this example (Figure 6) is used to show that if the methodology works well on certain characteristic signals (such as sinusoid, triangular, ramp, step change, and so on) then good performance can be expected for composite signals too. $l_{th} = 6$. The extrema points are not shown in the figure, but it is clear that most of the important points (mostly step changes in this case) are identified with good accuracy when SNR is fairly high (>10). The SNR for the signal shown in Figure 6c is not close to 1, but still performance has degraded considerably. A closer look at the noisy signal reveals that even an operator cannot identify the true trends with good accuracy, meaning that the correspondence between the SNR and encapsulation of qualitative trends is not very strong when a number of step changes are present. Still, the general result that performance degrades as SNR decreases holds true.

It is apparent from all these cases that the method performs very well in extracting all crucial features provided SNR is large enough. Table 2 reports the performance metrics. Of particular importance are SAGE (column 5) and RNE (column 8). SAGE is close to 1 in most of the cases. SLE (column 6) is also not very large. There are few exceptions to this. SAGE and SLE are quite large for the scenarios shown in Figures 5a, 6a, and 6b because of step changes. Since the step changes occur over just one sample time and the fitted data tries to approximate these step changes over a window of at least $l_{th}/2$ data points, the mismatch is unavoidable. Hence, SAGE and SLE are quite large for these cases. For Figure 6a, these values are extremely high (SAGE = 5.12 and SLE = 40.93) since there are many step changes and noise is small. In fact, even wavelet-based denoising results in excessive smoothing near step changes and overestimates the noise (see the respective figures and compare column 3 and 4 of Table 2). Notice that it is believed that such a level of SLE is acceptable because reconstruction of the data from the compressed data results in a denoised signal and not in a noisy signal. The SIX SIGMA rule indicates that SLE as large as 3 (or slightly more) can be accepted. Compression ratios also are quite high (4–42).

Industrial data

In this section, through four different data sets each corresponding to a different sensor, the methodology would be tested and the effect of various options (such as use of single vs. level-dependent scale factor, one time estimation vs. re-estimation of noise, and so on) would be analyzed. In general these signals are more rough compared to the smooth nature of the simulated signals above. Also, since the nature of the noise

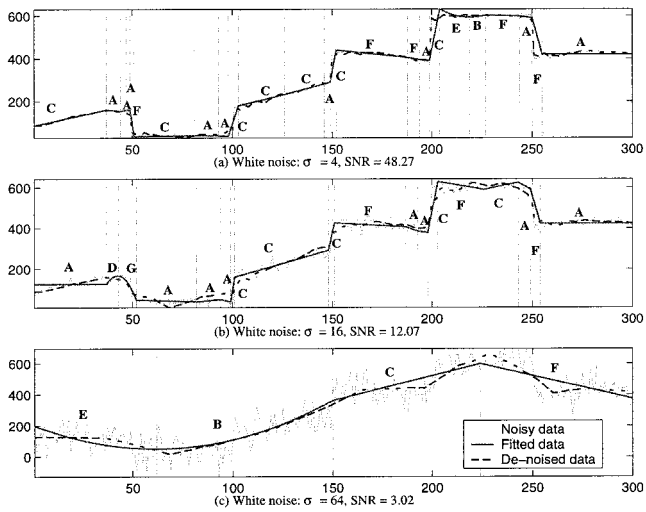

Figure 6. Simulated Case 3: superposition of Case 2, 30 sin(*t*/20) and steps.

Table 3. Performance Metrics for Industrial Signals

Case	Fig. No.	SNR	SAGE	SLE	ρ
1	7a	32.15	1.09	7.54	8.95
	7b	32.15	0.97	4.74	7.62
	7c	14.14	0.91	3.22	20.88
	7d	14.14	0.69	4.54	11.26
2	8a	10.84	1.01	3.79	20.88
	8b	10.84	0.90	3.15	16.38
	8c	10.23	1.00	3.67	25.28
	8d	10.23	0.85	3.10	17.15
3	9	23.68	2.12	24.23	10.60
4	10a	6.44	1.30	5.80	7.17
	10b	6.44	1.12	5.58	14.41

is not known *a priori,* nonwhite noise is assumed and the level-dependent estimation of noise scale (see the appendix) is used for calculation of σ_{noise}. In the first two data sets, single scale factor-based estimate of σ_{noise} is used to study the effect of scale factor. The values of the performance metrics for all the data sets (and all the cases for each data set) are listed in Table 3. In Table 3, SAGE and SLE are listed in columns 4 and 5, respectively. The values of these metrics (SAGE is close to 1 and SLE is not much greater than 3.5 for most of the cases) indicate that the fits are very good. The compression ratio is also good (7–25). Discussion on trend extraction for each data set is presented below.

Case 1: A Signal with High SNR. As shown in Figure 7, four (all possible) different combinations of the two options *viz.* choice of scale factor and choice on the noise estimation have been considered. Since the figures are already complicated, manual identification of the trends is not attempted. A glance at the four plots shows that data is fitted quite well in each case. The trends shown in Figure 7c are good despite some smoothing at the two ends. The trends shown in Figure 7a–7b reveal that the low value of σ_{noise} (single scale factor) has resulted in too many small segments. The trend shown in Figure 7d is the best among the four trends shown. Although the noise content in different parts of the signal appears the same, the local noise estimate (re-estimation of noise) provides better results as compared to the global noise estimate since the

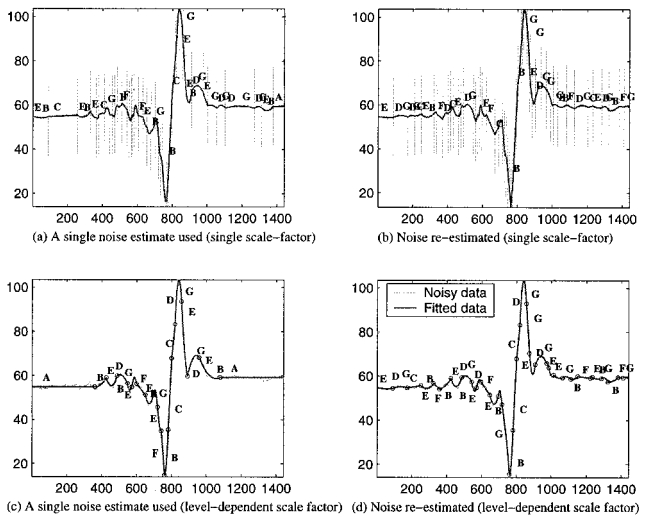

Figure 7. Industrial Case 1: trends in a signal with high SNR (adaptive windows).

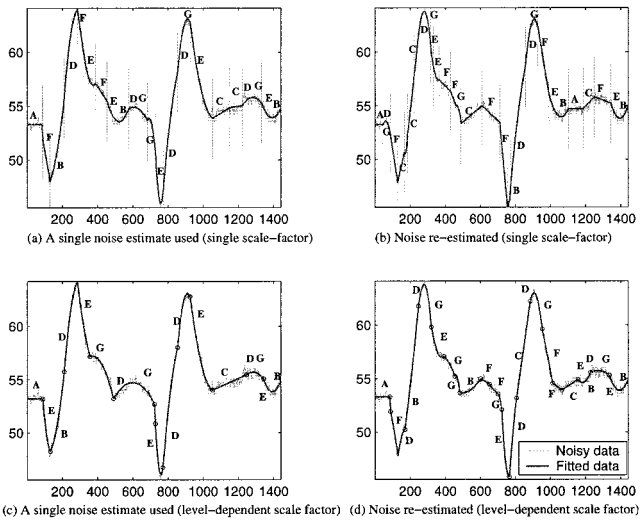

Figure 8. Industrial Case 2: a signal with stationary noise and moderate SNR.

former is a better representative of the noise in the region containing the segment under consideration.

Case 2: A Signal with Stationary Noise and Moderate SNR. Similar to the previous case, trends are extracted using four combinations of scale factor and noise estimation. The trends are shown in Figures 8a–8d. The trends in each subplot represent the signal closely. This means that a white noise sequence can characterize the noise. Also, the noise content in different part of the signal does not vary much (that is, quite stationary), although one may say that a particular trend is better as compared to the others in certain parts of the signal. For example, in Figure 8b, the **C** primitive in the time interval [490 608] represents the data better as compared to the corresponding primitive (in that region) in other plots.

Case 3: Use of Eq. 13. This example is used to show the improvement in fit when Eq. 13 (see the appendix) is used. The extracted trend (σ_{noise} re-estimated with level-dependent scale factor) is shown in Figure 9. Equation 13 has been used to constrain the constant polynomial fits. Consider the first four

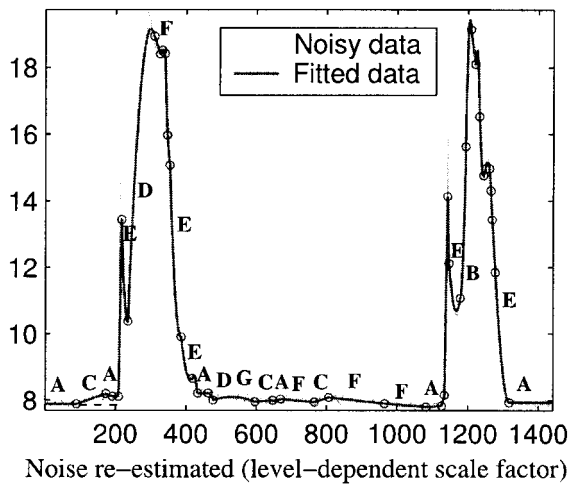

Figure 9. Industrial Case 3: use of Eq. 13 to restrict constant fit.

primitives (**ACAA**). These primitives fit the data quite well. If the second primitive (**C**) were to be an **A** primitive, then (due to the continuity condition) the fitted data would have been far from the noisy data (see the dotted lines). Indeed, this is what happens if Eq. 13 is not used. As mentioned in the Appendix, the reason is that the sub-global estimate of σ_{noise} (based on 128 data points which include the current segment and some future points) is much larger than the local noise content. Since such problems cannot be avoided completely even by reducing the minimum length for noise estimation, use of Eq. 13 is a good alternative.

Case 4: Improvement Due to Re-estimation of σ_{noise}. In the previous three cases, it has been found that the sub-global estimate (which is even a local estimate if $l \geq 128$) of σ_{noise} improves the quality of the fitted data. Still, the effect of re-estimation was not well clear since the variation in the noise was not much. In this example, a signal with large variation in the noise in different parts of the signal is considered. Figures 10a and 10b show the extracted trends from the signal without and with re-estimation of σ_{noise}, respectively. Needless to say, the trend shown in Figure 10b is much better, as compared to that shown in Figure 10a. Notice that level-dependent scale factor is used in both the cases. As shown in Figure 10b, the noise in the data segment [1 550] is much larger as compared to the noise in the rest of the signal. That is why re-estimation of noise has resulted in much better fit. In the first two segments, most of the noise is rejected. In the remaining portion, the signal is tracked well.

Through the four industrial examples presented above, it can be concluded that the best options for trend extraction are use of level-dependent scale factor for the estimation of σ_{noise} and re-estimation of σ_{noise}. In fact, to decide whether or not re-estimation is necessary, one can estimate noise in different parts of the signal and find out whether or not they differ (F-test can be conveniently used here). A comparative analysis of the four cases is carried out using Table 3. The value of SAGE is very good except for Case 3. Ironically, the fit appears very good (Figure 9). This can be explained as follows. The value of SAGE is calculated using σ_{noise} and the denoised signal estimated from the entire noisy signal, but σ_{noise} is re-estimated during trend identification. When a signal is very long, denoising can lead to considerable smoothing and, hence, although a fitted trend (data) might seem to capture the important features well, it might deviate from the denoised data. Essentially, the aggregate metric SAGE does not really reflect the fit criteria well in such a case. In fact, if SAGE were to be calculated using the fitted data and the noisy data, its value would have been much closer to 1. The above explanation also accounts for large values of SLE for Case 3 and Case 4, although in Case 3, it is excessively large because of few sharp peaks, such as at sample 1141 (not visible in Figure 9). A large value of SLE for the trend shown in Figure 7a is due to poor fit (forced termination of interval-halving at small lengths). Another important observation in Table 3 (column 3, SNR) is that the noise in the signal used in Case 1 is non-white noise since the values of SNR with a single scale factor and a level-dependent scale factor differ from each other considerably. In Case 2, the noise is almost white noise.

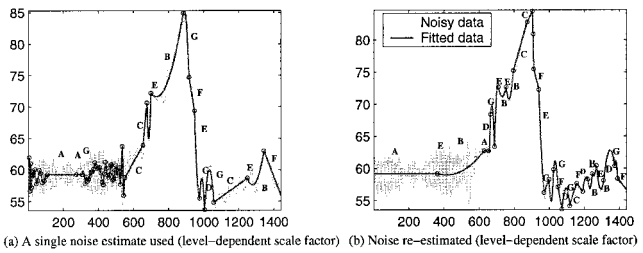

Figure 10. Industrial Case 4: improvement due to re-estimation of σ_{noise}.

Conclusions

Qualitative trend analysis is a simple yet powerful method to reason about system behavior using historical data, more so when fundamental knowledge is absent or minimal. Identification of qualitative trends from any type of signal is the first step in qualitative trend analysis. The main challenge lies in automating the task while not losing accuracy. The primitives-based trend language is used to represent trends, since it is a simple scheme and could capture any type of sensor behavior. To this end, a simple strategy based on interval-halving is presented in this article. The idea is to locate a sequence of unimodal regions characterizing the data using the least-squares polynomial fits to parameterize the data quantitatively. To estimate the significance of fit-errors, the noise estimated using wavelet-based denoising is used. Step changes (if any) in the signal are identified using an outlier detection methodology. Once the unimodal regions are obtained, a constrained polynomial fit is used to fine-tune the polynomials to ensure continuity of the fitted data across consecutive unimodal regions. The primitives are easily assigned to the unimodal segments using the derivative signs which are also tested for their statistical significance. While the strategy outlined is sound, certain variations can be employed to deal with some specific cases. Utilization of orthogonal polynomials such as Legendre instead of the usual $(1, x, x^2)$ quadratics might simplify calculations and yield better results. However, the fundamental idea of interval-halving is a robust one and would work across a broad variety of cases. The main benefit of this procedure is the complete automation of the task of trend extraction. All the parameters used in the methodology can be easily tuned and guidelines have been provided for the same. The technique has been exhaustively tested on several cases, both simulated and real, and is seen to perform very well. As an added bonus, data compression (with high compression ratio) is also achieved.

Notation

$\boldsymbol{I}$ = an identity matrix of appropriate order
J = decomposition level in wavelet analysis
L = Lagrangian function
M = total number of unimodal segments
N = total number of data points in the signal, normal distribution
RNE = ratio of the number of extrema
SAGE = scaled average global error
SLE = scaled L_∞ error
$\boldsymbol{T}$ = time matrix in least-squares estimation: $\boldsymbol{y} = \boldsymbol{T}\beta + \boldsymbol{e}$
U_i = i^{th} unimodal region
W_{id} = identification window
W_1, W_2 = windows for outlier detection near the two ends of a primitive
d = DWT detail coefficients
$d1$, $d2$ = first and second derivatives

$e(t)$, $\boldsymbol{e}$ = error signal
$f(t)$ = unknown true data
$\hat{f}(t)$ = wavelet estimated denoised data
l = length of a segment
p_i = piecewise polynomial in the i^{th} region
s = DWT smooth coefficients
s_i = standard deviation of $\hat{\beta}_i$
t, T = time
$t_{d1=0}$ = normalized time where first derivative is zero
$y(t)$, $\boldsymbol{y}$ = sensor data
$\bar{\boldsymbol{y}}$ = unknown true data (described exactly by a polynomial)
$\hat{\boldsymbol{y}}$ = unbiased estimate of $\bar{\boldsymbol{y}}$

Subscripts

f = final (last) time, function
half = middle point of the segment to be halved
high, low = upper and lower limits, respectively
i = generic index with no specific connotation, initial time
j = level j in multiscale wavelet analysis, generic index
k = generic index with no specific connotation
max = maximum value of n_1 or n_2
n = order of a polynomial
new = values of $d1_0$, $d1_1$ and $d2$ after t-test
t = normalized time
th = threshold
1 = unimodal segment, or segment 1 in CPF
2 = signal used for estimation of σ_{noise}, or segment 2 in CPF

Greek letters

Σ = variance-covariance matrix
α = significance level
$\hat{\boldsymbol{\beta}}$ = least-squares estimated quadratic coefficients: $\hat{\boldsymbol{y}} = \boldsymbol{T}\hat{\boldsymbol{\beta}}$
$\boldsymbol{\beta}$ = actual quadratic coefficient: $\boldsymbol{y} = \boldsymbol{T\beta} + \boldsymbol{e}$
$\delta_c(x)$ = coefficients shrinkage function
ϵ^2_{fit} = polynomial fit error
λ, μ = Lagrange parameters in CPF
ν = degrees of freedom
ϕ = scaling function
ψ = wavelet function
ρ = compression ratio
σ = standard deviation of the added noise
σ^2_{noise} = estimated noise variance
$\hat{\sigma}$ = Median Absolute Deviation (MAD) function for noise scale estimation
σ_j = noise scale at level j

Literature Cited

Bakshi, B. R., and G. Stephanopoulos, "Representation of Process Trends—III. Multiscale Extraction of Trends from Process Data," *Comput. & Chem. Eng.,* **18**(4), 267 (1994a).

Bakshi, B. R., and G. Stephanopoulos, "Representation of Process Trends—IV. Induction of Real-Time Patterns from Operating Data for Diagnosis and Supervisory Control," *Comput. & Chem. Eng.,* **18**(4), 303 (1994b).

Bartle, R., *Elements of Real Analysis,* Wiley, New York (1976).

Cheung, J. T., and G. Stephanopoulos, "Representation of Process Trends—Part I. A Formal Representation Framework," *Comput. & Chem. Eng.,* **14**(4/5), 495 (1990).

Dash, S., "Data-Driven Qualitative and Model-Based Quantitative Approaches to Fault Diagnosis," PhD Thesis, Purdue University (2001).

Dash, S., M. R. Maurya, R. Rengaswamy, and V. Venkatasubramanian, "A Novel Interval-Halving Framework for Automated Identification of Process Trends: Extended Version," Technical Report CIPAC-03-3, Purdue University (2003a).

Dash, S., R. Rengaswamy, and V. Venkatasubramanian, "A Novel Interval Halving Algorithm for Process Trend Identification," *4th IFAC Workshop on On-Line Fault Detection & Supervision in the Chemical Process Industries,* Korea, 155 (2001).

Dash, S., R. Rengaswamy, and V. Venkatasubramanian, "Fuzzy-Logic based Trend Classification for Fault Diagnosis of Chemical Processes," *Comput. & Chem. Eng.,* **27**(3), 347 (2003b).

Davis, J. F., B. R. Bakshi, K. A. Kosanovich, and M. J. Piovoso, "Process Monitoring, Data Analysis and Data Interpretation," *Proc. of the First Int. Conf. on Intelligent Systems in Process Eng.,* Snowmass Village, CO, G. Stephanopoulos, J. F. Davis, and V. Venkatasubramanian, eds., CACHE Corp., University of Texas, Austin, TX, 1 (July 9–14, 1995).

Donoho, D. L., and I. M. Johnstone, "Ideal Spatial Adaptation via Wavelet Shrinkage," *Biometrika,* **81**(3), 425 (1994).

Huang, Y. J., G. V. Reklaitis, and V. Venkatasubramanian, "A Heuristic Extended Kalman Filter Based Estimator for Fault Identification in a Fluid Catalytic Cracking Unit," *Ind. Eng. Chem. Res.,* **42**(14), 3361 (2003).

Janusz, M., and V. Venkatasubramanian, "Automatic Generation of Qualitative Description of Process Trends for Fault Detection and Diagnosis," *Eng. Applic. Artif. Intell.,* **4**(5), 329 (1991).

Jimenez, S., S. Bulgakov, and L. Vazquez, "Efficient Shooting Algorithms for Solving the Nonlinear One-Dimensional Scalar Helmholtz Equation," *Applied Mathematics and Computation,* **95**(2–3), 101 (1998).

Kennedy, J. P., "Data Treatment and Applications—Future of Desktop," *Proc. of Foundations of Computer-Aided Process Operations,* Mount Crested Butte, CO, D. W. T. Rippin, J. C. Hale, and J. F. Davis, eds., CACHE Corp., University of Texas, Austin, TX 1 (1993).

Keogh, E. J., S. Chu, D. Hart, and M. J. Pazzani, "An Online Algorithm for Segmenting Time Series," *IEEE Int. Conf. on Data Mining ICDM,* San Jose, CA, IEEE Computer Society, Los Alamitos, CA. 289 (Nov. 29–Dec. 2, 2001).

Kiefer, J., "Optimum Sequential Search and Approximation Methods under Minimum Regularity Assumptions," *J. Soc. Ind. Appl. Math.,* **5**(3), 105 (1957).

Konstantinov, K. B., and T. Yoshida, "Real-Time Qualitative Analysis of the Temporal Shapes of (Bio)process Variables," *AIChE J.,* **38**(11), 1703 (1992).

Krongold, B. S., K. Ramchandran, and D. L. Jones, "Computationally Efficient Optimal-Power Allocation Algorithms for Multicarrier Communication Systems," *IEEE Trans. Communications,* **48**(1), 23 (2000).

Mah, R. S. H., A. C. Tamhane, S. H. Tung, and A. N. Patel, "Process Trending with Piecewise Linear Smoothing," *Comput. & Chem. Eng.,* **19**(2), 129 (1995).

Maurya, M. R., "Integrating Causal Models and Trend Analysis for Process Fault Diagnosis," PhD Thesis, Purdue University (2003).

Misra, M., S. Kumar, S. J. Qin, and D. Seemann, "Error Based Criterion for On-line Wavelet Data Compression," *J. of Process Control,* **11**(6), 717 (2001).

Misra, M., H. H. Yue, S. J. Qin, and C. Ling, "Multivariate Process Monitoring and Fault Diagnosis by Multi-scale PCA," *Comput. & Chem. Eng.,* **26**(9), 1281 (2002).

Muske, K., J. Young, P. Grosdidier, and S. Tani, "Crude Unit Product Quality Control," *Comput. & Chem. Eng.,* **15**(9), 629 (1991).

Najim, K., and M. M. Saad, "Adaptive Control: Theory and Practical Aspects," *J. of Process Control,* **1**(2), 84 (1991).

Paritosh, P. K., and R. Rengaswamy, "Interval-Halving Techniques for Process Trend Identification," Technical Report PROCISS-99-01, I.I.T. Bombay, Mumbai, India (1999).

Peters, M. S., and K. D. Timmerhaus, *Plant Design and Economics for Chemical Engineers,* 4th ed., McGraw-Hill, New York (1990).

Rengaswamy, R., T. Hagglund, and V. Venkatasubramanian, "A Qualitative Shape Analysis Formalism for Monitoring Control Loop Performance," *Eng. Applic. Artif. Intell.,* **14**(1), 23 (2001).

Rengaswamy, R., and V. Venkatasubramanian, "A Syntactic Pattern-Recognition Approach for Process Monitoring and Fault Diagnosis," *Eng. Applic. Artif. Intell.,* **8**(1), 35 (1995).

Saxena, S. C., V. Kumar, and S. T. Hamde, "ECG Data Compression using Non-redundant Templates," *IETE Tech. Rev.,* **17**(5), 299 (2000).

Sebzalli, Y., R. Li, F. Chen, and X. Wang, "Knowledge Discovery from Process Operational Data for Assessment and Monitoring of Operator's Performance," *Comput. & Chem. Eng.,* **24,** 409 (2000).

Vedam, H., "OP-AIDE: An Intelligent Operator Decision Support System for Diagnosis and Assessment of Abnormal Situations in Process Plants," PhD Thesis, Purdue University (1999).

Whiteley, J. R., and J. F. Davis, "Knowledge-Based Interpretation of Sensor Patterns," *Comput. & Chem. Eng.,* **16**(4), 329 (1992).

Witkin, A. P., "Scale-space Filtering," *Proc. Int. Joint Conf. Artificial*

Intell., Karlsruhe, Germany, A. Bundy, ed., Morgan Kaufmann Publishers, San Francisco, 1019 (Aug. 8–12, 1983).
Yabuki, Y., T. Nagasawa, and J. F. MacGregor, "Industrial Experiences with Product Quality Control in Semi-batch Processes," *Comput. & Chem. Eng.,* **26**(2), 205 (2002).

Appendix

Wavelet denoising for noise estimation

Wavelet denoising is used to estimate the noise parameter in the interval-halving based trend extraction approach. The most important advantage of wavelet-based analysis is the requirement of minimal *a priori* information. Their excellent time-frequency localization allows good nonparametric statistical estimation of the true, that is, denoised, data. Any process signal can be represented as

$$y(t) = f(t) + e(t) \tag{A1}$$

where $y(t)$ = noisy data, $f(t)$ = unknown "true" signal and $e(t)$ (noise) is usually assumed to be independent and identically distributed (IID) normal errors $\sim N(0, \sigma^2)$. The orthogonal wavelet series approximation to a continuous time signal $y(t)$ is given by

$$y(t) \approx \sum_k s_{J,k}\phi_{J,k}(t) + \sum_{j=1}^{j=J}\sum_k d_{j,k}\psi_{j,k}(t) \tag{A2}$$

where J is the number of multiresolution components (or *scales*), and k is the number of coefficients at level j. $s_{J,k}$ are called smooth coefficients and $d_{J,k}, \ldots, d_{1,k}$ are the detail coefficients. The functions $\phi_{j,k}(t)$ and $\psi_{j,k}(t)$ are approximating wavelet functions generated from ϕ and ψ by scaling (factor 2^j) and translation (parameter $2^j k$). When a discrete signal $y_1, y_2, \ldots, y_n$ (sampled data) is used, this approximation is referred to as the Discrete Wavelet Transform (DWT).

There exists an impressive theory for nonparametric regression and smoothing based on wavelet shrinkage (Donoho and Johnstone, 1994). The principle consists of (1) applying the DWT, (2) shrinking the small coefficients to zero, and (3) applying the inverse discrete wavelet transform (IDWT). The shrinking of wavelet coefficients usually is defined by the shrinkage function $\delta_c(x)$, (c = shrinkage threshold). The threshold is given usually as $c_j = \lambda_j\sigma_j$ where λ_j is the threshold rule and σ_j is the noise scale. Usually, λ is taken to be the universal threshold defined by $\sqrt{2 \log N}$ (N = sample size) and results in high degree of smoothness. In cases of white noise, the finest scale detail coefficients $\boldsymbol{d}_1$ are used to estimate a single scale factor for all levels whereas for non-white noise, a level-dependent scale factor is estimated as $\sigma_j = \hat{\sigma}(\boldsymbol{d}_j)$. The $\hat{\sigma}$ is the Median Absolute Deviation (MAD) function, a highly robust estimate of scale. The estimated sensor noise (σ_{noise}) and signal/noise ratio (SNR) can then be simply estimated as ($\hat{f}(t)$ is the wavelet estimated denoised function of f)

$$\sigma_{\text{noise}} = \sigma(y(t) - \hat{f}(t))$$

$$\text{SNR} = \frac{\sigma_f}{\sigma_{\text{noise}}} \tag{A3}$$

Constrained least-squares-based polynomial fit

Constrained polynomial fit (CPF) is used to refine the order and the coefficients of the fitted polynomials to ensure continuity between two adjacent segments (and hence continuity in the overall fitted signal). See Dash et al. (2003a) for other applications of CPF. In this section, first the CPF formulation and its solution is presented. Then, various related quantities (such as degrees of freedom and covariance) are calculated and issues are discussed. Let the standard polynomial fit problem in the two consecutive segments (call them segment 1 and 2) be (see Figure A1)

$$\mathbf{y}_1 = \mathbf{T}_1\boldsymbol{\beta}_1 + \mathbf{e}_1;\ \mathbf{y}_2 = \mathbf{T}_2\boldsymbol{\beta}_2 + \mathbf{e}_2 \tag{A4}$$

where the subscripts 1 and 2 refer to the segments 1 and 2, respectively. Throughout this section, context should be used to resolve among various notations (including those that are used elsewhere). $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are defined similar to as $\boldsymbol{T}$ in Eq. 1 and they are based on the normalized window in the respective segments. Let the number of data points and the order of the fitted polynomial in the i^{th} segment be l_i and n_i, respectively. Clearly, $\boldsymbol{T}_i \in \mathbb{R}^{l_i \times (n_i+1)}$ and $\beta_i \in \mathbb{R}^{(n_i+1)\times 1}$. An equality constraint at the end of segment 1 and at the beginning of segment 2, and a fixed value constraint at the start of segment 1 can be written as

$$\mathbf{c}_1^T\boldsymbol{\beta}_1 - \mathbf{c}_2^T\boldsymbol{\beta}_2 = 0 \tag{A5}$$

$$\mathbf{c}_0^T\boldsymbol{\beta}_1 - d_0 = 0 \tag{A6}$$

where $\mathbf{c}_0^T$ and $\mathbf{c}_1^T$ are the first and last rows of $\boldsymbol{T}_1$, respectively, and $\boldsymbol{c}_2^T$ is the first row of $\boldsymbol{T}_2$ (Figure A1). Thus, if $n_1 = n_2 = 2$ then $\boldsymbol{c}_0^T = [0\ 0\ 1]^T$, $\boldsymbol{c}_1^T = [1\ 1\ 1]^T$ and $\boldsymbol{c}_2^T = [0\ 0\ 1]^T$ due to normalization. d_0 (Figure A1) is the fitted value of the signal at the end of the then segment 1 during the previous pass through the CPF step (Step 2 of the interval-halving procedure). Thus, Eq. 8 is not used during the first pass through Step 2. The Lagrangian for the constrained least-square problem is

$$L(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \lambda, \mu) = \mathbf{e}_1^T\mathbf{e}_1 + \mathbf{e}_2^T\mathbf{e}_2 + \lambda(\mathbf{c}_1^T\boldsymbol{\beta}_1 - \mathbf{c}_2^T\boldsymbol{\beta}_2) + \mu(\mathbf{c}_0^T\boldsymbol{\beta}_1 - d_0) \tag{A7}$$

where λ and μ are the Lagrange multipliers corresponding to Eqs. 7 and 8, respectively (Figure A1). Using Eq. 6 and minimizing $L(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \lambda, \mu)$ (by differentiating L with respect to the unknowns and equating the differentials to zero) yields the following equation

$$\begin{bmatrix} 2\mathbf{T}_1^T\mathbf{T}_1 & \mathbf{0} & \mathbf{c}_1 & \mathbf{c}_0 \\ 0 & 2\mathbf{T}_2^T\mathbf{T}_2 & -\mathbf{c}_2 & \mathbf{0} \\ \mathbf{c}_1^T & -\mathbf{c}_2^T & \mathbf{0} & \mathbf{0} \\ \mathbf{c}_0^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \hat{\boldsymbol{\beta}}_1 \\ \hat{\boldsymbol{\beta}}_2 \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} 2\mathbf{T}_1^T\mathbf{y}_1 \\ 2\mathbf{T}_2^T\mathbf{y}_2 \\ 0 \\ d_0 \end{bmatrix} \tag{A8}$$

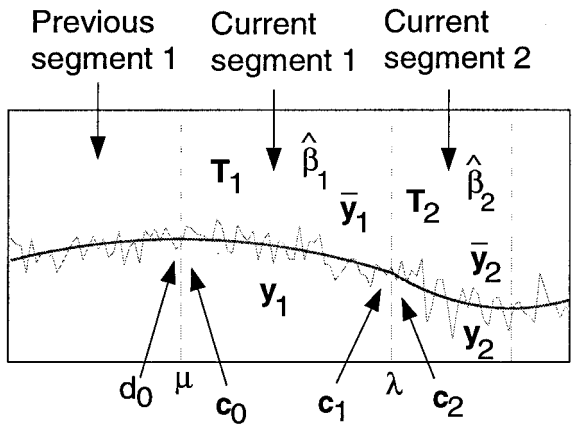

Figure A1. Constrained polynomial fit.

For the sake of simplicity, lets rewrite the above equation as $\boldsymbol{AX} = \mathbf{b}$ with $\boldsymbol{X} = [\hat{\boldsymbol{\beta}}_1^T \ \hat{\boldsymbol{\beta}}_2^T \ \lambda \ \mu]^T$, and so on. Equation A8 is the main equation of CPF. This approach (based on CPF over two consecutive segments) is preferred as compared to applying CPF over all the segments simultaneously since the former approach results in an equation with only few (≤ 8) unknowns and, hence, can be easily solved for $\boldsymbol{X}$. Further, the refinement of n_1 and n_2 can be easily carried out (Step 2). Next, DOF (needed for covariance calculation and t-test) for the two segments are calculated.

Calculation of the Degrees of Freedom. Since both the segments are used for CPF, theoretically, DOF can be estimated only for $[\boldsymbol{e}_1^T \ \boldsymbol{e}_2^T]^T$ (both the segments jointly). DOF increases due to the equality constraints since the number of free parameters to be estimated decreases accordingly. Thus, DOF $= l_1 + l_2 - (n_1 + n_2 + 2) + 2$. Although the joint DOF and the fit-error can be used for the F-test (over the two segments jointly) and the t-test, it practically has been observed that the F-test and t-test based upon the individual segments perform better when the noise in the two segments differ considerably. An explanation for this observation is that the joint test uses an averaged value of fit error (thus reduces the effect of re-estimation of σ_{noise}). Also, individual fit errors are required in the calculation of the covariance. So a fictitious DOF is calculated for the two segments—it is assumed that segment 1 is independent of segment 2, but segment 2 is dependent on segment 1. Thus, DOF for segment 1, ν_1, is $l_1 - (n_1 + 1)$, and DOF for segment 2, ν_2, is $l_2 - (n_2 + 1) + 1$. Another way to explain ν_1 is to analyze the constraint that explains the DOF for segment 1, viz. (derived from Eq. 10)

$$[2\mathbf{T}_1^T \ \mathbf{c}_1 \ \mathbf{c}_0][\mathbf{e}_1^T \ \lambda \ \mu]^T = \mathbf{0} \tag{A9}$$

The above equation comprises of $(n_1 + 1)$ scalar equations, so the DOF for any quantity estimated by using all of $[\boldsymbol{e}_1^T \ \lambda \ \mu]^T$ is $l_1 + 2 - (n_1 + 1)$ and that the DOF for any quantity estimated by using only $\boldsymbol{e}_1$ is $l_1 - (n_1 + 1)$ $(= \nu_1)$. A similar expression for segment 2 yields $\nu_2 = l_2 - (n_2 + 1)$ which does not reflect the dependence of segment 2 on segment 1 so the former approach ($\nu_2 = l_2 - (n_2 + 1) + 1$) is preferred. Next, $\hat{\sigma}_1^2 = \epsilon_{\text{fit},1}^2 = \boldsymbol{e}_1^T\boldsymbol{e}_1/\nu_1$ and $\hat{\sigma}_2^2 = \epsilon_{\text{fit},2}^2 = \boldsymbol{e}_2^T\boldsymbol{e}_2/\nu_2$. These results are used to perform F-test. ν_2 refers here to the DOF for the estimation of $\epsilon_{\text{fit},2}^2$ rather than that of σ_{noise}^2.

The Covariance of $\boldsymbol{X}$. The expression for the covariance matrix of $\boldsymbol{X}$, $\boldsymbol{\Sigma}_{\boldsymbol{X}}$, is

$$\boldsymbol{\Sigma}_{\mathbf{X}} = \mathbf{A}^{-1}\boldsymbol{\Lambda}\mathbf{A}^{-1} \tag{A10}$$

where $\boldsymbol{\Lambda}$ is a block diagonal matrix with the leading diagonal blocks being $4\boldsymbol{T}_1^T\boldsymbol{T}_1\hat{\sigma}_1^2$, $4\boldsymbol{T}_2^T\boldsymbol{T}_2\hat{\sigma}_2^2$, 0 and 0. Once $\boldsymbol{\Sigma}_{\boldsymbol{X}}$ is known, $\boldsymbol{\Sigma}_{\hat{\beta}_1}$ and $\boldsymbol{\Sigma}_{\hat{\beta}_2}$ (to be used in t-test) can be extracted as appropriate square sub-matrices (along the leading block diagonal) from $\boldsymbol{\Sigma}_{\boldsymbol{X}}$.

Further Restriction on Constant Fits. Through several case studies, it has been observed that the above methodology performs well except when: (1) the final value of n_1 (and n_2 in the case of the last segment of the entire data set) is 0 (constant polynomial fit passes the F-test) for two or more consecutive segments; (2) the estimated σ_{noise} is very different from the noise content in the region around the segment(s) under consideration. This problem is mostly encountered when σ_{noise} is re-estimated as the intervals are halved. So, apart from satisfying the F-test, the following restriction is imposed for accepting a constant fit in a segment.

The (constant) fitted value of the signal (that is, $(\hat{\boldsymbol{\beta}}_i)_1$) should be close to the mean value of the signal in the segment (this is the value that is achieved in Step 1 for $n_i = 0$, $i = 1, 2$). The above restriction is realized by using a two-sided t-test. Let $\hat{y}$ and σ be the mean value and the standard deviation, respectively, of the data in the i^{th} segment. The constant fit should be accepted only if the following inequality is satisfied

$$|\hat{y} - (\hat{\boldsymbol{\beta}}_i)_1| \leq t_{1-(\alpha/2),\nu_i}\sigma \tag{A11}$$

The above inequality is only a necessary condition. The other condition that should be satisfied is the standard F-test. A high value should be chosen for α to make the test severe. For the case studies presented in this article, the values of α for segments 1 and 2 are 0.80 and 0.50, respectively.

Manuscript received Feb. 10, 2003, and revision received June 9, 2003.