Process Fault Detection Using Time-Explicit Kiviat Diagrams

Ray C. Wang
McKetta Dept. of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712

Thomas F. Edgar
McKetta Dept. of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712

Energy Institute, The University of Texas at Austin, Austin, TX 78712

Michael Baldea
McKetta Dept. of Chemical Engineering, The University of Texas at Austin, Austin, TX 78712

Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712

Mark Nixon, Willy Wojsznis, and Ricardo Dunia
Emerson Process Management, Round Rock, TX 78681

DOI 10.1002/aic.15054
Published online October 14, 2015 in Wiley Online Library (wileyonlinelibrary.com)

Significant amounts of data are collected and stored during chemical process operations. The corresponding datasets are typically difficult to represent and analyze using traditional visualization methods. This article introduces time-explicit Kiviat diagrams as a means to visualize the multidimensional time series data acquired from plant operations. This framework is then used to build multivariate control charts for large scale time series datasets, and to develop a fault detection mechanism that lends itself to real-time implementation. The proposed methodology is applied to an industrial case study as well as to data obtained from the Tennessee Eastman process simulator, showing very good performance. © 2015 American Institute of Chemical Engineers *AIChE J*, 61: 4277–4293, 2015
Keywords: fault diagnosis, data analysis, data visualization

Introduction

Recent years have brought a tremendous increase in the availability of large-scale datasets describing the state and evolution of a broad array of social, economic, and industrial systems. The contributing factors to the advent of the "Big Data" age are numerous—they include the explosive growth of, among others, business and transactional-type operations (e.g., e-commerce, enterprise resource planning) as well as social media, coupled with a significant rise of data transmission and storage capabilities. In a different vein, high throughput experimentation in the natural and life sciences has also played an important role in generating large amounts of data that require analysis.

The availability of such massive and often heterogeneous datasets has, in turn, triggered intense research activity to develop fast and scalable analysis methods. Loosely termed as "data mining," such methods aim to identify new (and often unexpected) correlations and insights from existing data.[1] The range of applications reported in the literature is broad, including, for example, e-commerce,[2] traffic analysis,[3] finance,[4] healthcare.[5]

Process industries have been a leading source of large datasets. One of the most frequent uses of recorded data is the development of mechanisms for identifying and characterizing process faults. The task of process monitoring has become increasingly challenging as the volume of the data increases. Depending on plant size, records of the time evolution of a few hundreds to that of tens of thousands of variables (tags) may be collected (often with sampling intervals of less than 1 min) and made available in data historians. Process operators thus often find themselves "drowning in data"[6] due to the lack of time and resources required to analyze and generate value from these large volumes of information. Nevertheless, tools such as principal component analysis (PCA) and partial least squares regression have been successfully used to detect and isolate faults pertaining to individual process variables and units. Applications on both *in silico* test cases and real-life industrial problems have been reported in the literature (see, e.g., the reviews in Refs. 7–10).

The tools listed above are typically implemented for process monitoring by control room operators, who often rely on visual data representations to draw empirical inferences about the process state and potential operational difficulties. Here, too, dimensionality reduction techniques such as PCA have proven to be valuable, forming the basis for, for example, score and square prediction error (SPE) plots.

This is a "Best Paper" initiative article.
This contribution was identified by Dr. Jin Wang (Auburn University) as the Best Presentation in the session "Process Monitoring and Fault Detection I" of the 2013 AIChE Annual Meeting in San Francisco, CA.

Correspondence concerning this article should be addressed to M. Baldea at mbaldea@che.utexas.edu

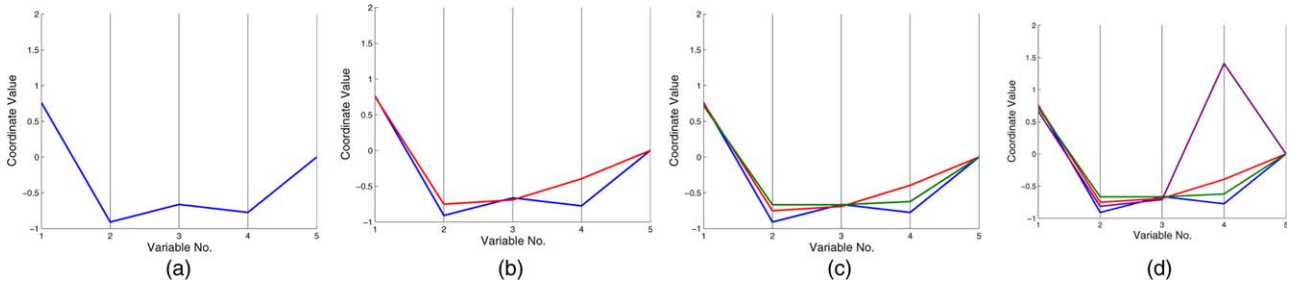


Figure 1. Data visualization in parallel coordinates for a five-dimensional dataset.

Each coordinate can be regarded as the ordinate of a regular time-series plot. Data samples are added to the plot as they are acquired; subfigures a–d represent a sequence of four samples. Data from Appendix A (Samples 7–10). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Visual representations are hampered by the limitations of the Cartesian system: data can be plotted in only three dimensions (equivalently, only three attributes of each data sample can be represented simultaneously). Monitoring multiple such two-dimensional (2-D) or three-dimensional (3-D) plots makes it difficult for process operators to observe and interpret a broad repertoire of time-dependent process trends and events.

To break this "curse of dimensionality," more recent research has focused on process monitoring based on representing multidimensional data in parallel coordinates. Initial efforts[11–13] considered plotting the data or leading PCA components in parallel coordinates, were later expanded[14] to include PCA-based statistical tests, such as Hotelling's T^2 and the SPE, and improved definitions of the confidence regions.

While parallel coordinate plots alleviate one of the shortcomings of presenting data in the Euclidean space (i.e., the limitation on the number of dimensions that can be plotted), they nevertheless present the disadvantage of not explicitly representing time. Motivated by this, in the present work we introduce time-explicit Kiviat diagrams as a class of multivariate plots with an explicit time dimension. We discuss novel applications of time-explicit Kiviat plots for characterizing the operation of chemical processes via appropriate confidence regions, and use these plots for the timely detection of process faults and events. Finally, we demonstrate the application of this novel framework to fault detection on data, both simulated and collected from industry.

The article is organized as follows: Background for this work is provided through a review of parallel coordinates and their use. An introduction to 3-D time-explicit Kiviat plots is followed by a discussion of their properties and their use in fault detection. Subsequently, we present case studies based on an industrial dataset as well as on data obtained from simulating the Tennessee Eastman benchmark process.[46] The case studies illustrate the capabilities of the proposed framework; comparisons with results from the literature are provided as well. Finally, we discuss the results and delineate directions for future work.

Background: Visualization of Multivariate Data and Visualization-Based Fault Detection

Fault detection is a key area of process monitoring. Early detection and prevention of process faults is highly valued by practitioners, given the economic cost and safety risks that faults may incur. Fault detection methods fall in three broad categories: data-based methods include the use of PCA, T^2 values and Q statistics, spectral analysis, and various other analytical methods to extract information from process operating data. Model-based methods rely on building a (typically first-principles) model of the plant and then comparing data from the plant to the model through the use of residuals or state observers to determine if a fault has occurred. Knowledge-based methods use expert knowledge from plant engineers to establish a rule-based system to determine if a faulty condition has occurred. There are several papers available that review the different methods known in each category,[9,10,15,16] and a comprehensive presentation is beyond the scope of this article.

Of the data-driven methods, we will focus in particular on visualization-based approaches. Visualization is vital to the analysis of data, and the representation of time series data from the operation of chemical processes has received significant attention over time.[17–19] There are several important characteristics that a visual representation of data should provide:

1. Capture the multivariate nature of data—this becomes difficult as the number of "tags" or variables increases.
2. Explicitly represent the time dimension to capture important trends.
3. Provide high resolution and minimize cluttering, which in turn provide for fast and easy interpretation.

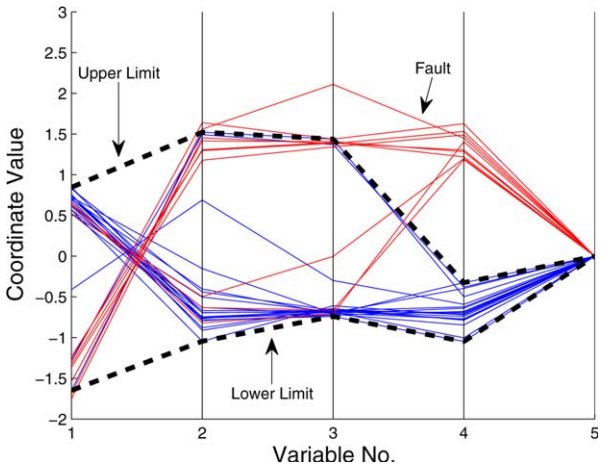
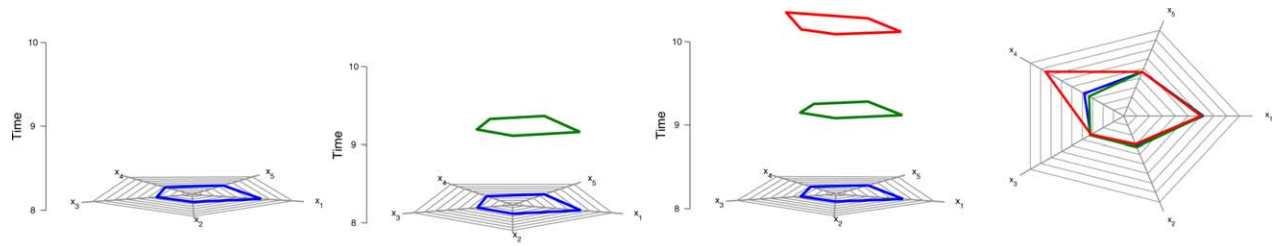


Figure 2. Fault detection using parallel coordinates: univariate upper and lower control limits are used to define a ribbon-like confidence region around the plot (shown in dashed lines).

Data samples that fall outside this confidence region indicate a fault in the process. Data from Appendix A. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]



(a) $t = 8$ (b) $t = 9$ (c) $t = 10$ (d) Top view of plot

Figure 3. Representing multidimensional data in time-explicit Kiviat plots.

A five-dimensional dataset with 1-min sampling time is used for illustration purposes. The first sample is represented in radial coordinates and a time axis normal to the plane of the plot is added (a). Subsequent samples are added as radial plots aligned along the time axis (b–c). The plot can be updated by adding such "data slices" in a first-in, first-out fashion. The top view of the plot is equivalent to projecting all data on the radial coordinate plane (d). This graphical representation is related to the parallel coordinate system introduced by Inselberg[20]; however, it brings the crucial advantage of an *explicit representation of time*. Data from Appendix A (Samples 8–10). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Parallel coordinate plots

The use of parallel coordinates, first proposed by Inselberg,[20] has been explored as a method of displaying multivariate information. In a parallel coordinate plot, each (multivariate) data sample is represented by an open line that connects the values of each variable in the respective sample. The variables are plotted on a set of parallel ordinates, with no abscissae (Figure 1).

Initial efforts aimed at visualization-based fault detection using parallel coordinate plots[11,12] considered plotting the data or leading PCA components, followed by defining a confidence region around the plot. A fault is declared once the data fall outside the confidence region (Figure 2).

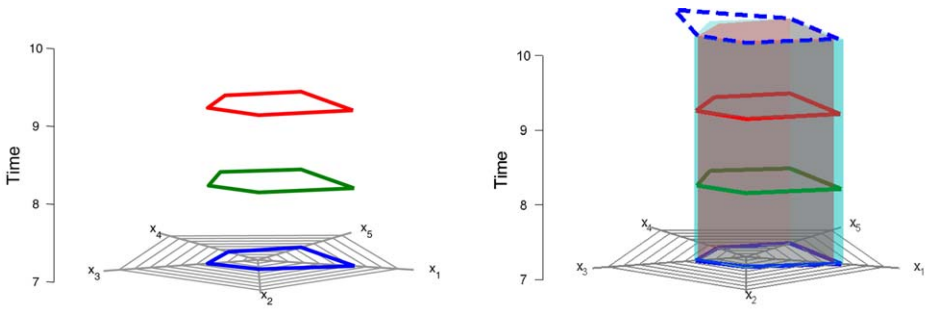
Later, these concepts were expanded to include PCA-based statistical tests, such as Hotelling's T^2 and the SPE.[14] With the aid of appropriately defined (visual) confidence regions, these techniques were proven to be successful in the detection of process faults and events.[21]

However, parallel coordinates are ill-suited for use with datasets with significant variability in time; while the multivariate aspect of such datasets is adequately addressed, parallel coordinate plots are affected heavily by cluttering—data points overlap and eventually obscure any potentially meaningful information. This issue is generally dealt with by designating the first two parallel axes as date and time, which may then be used to select and filter for the desired period of time.[22]

Data representation using time-explicit Kiviat plots

Kiviat plots (named after Philip J. Kiviat,[23] also referred to as Kiviat diagrams, star plots, or spider plots), can be regarded as an evolution of the parallel coordinates plot. However, here the coordinate axes extend radially from a center point, similar to spokes of a bicycle wheel. Akin to parallel coordinates, Kiviat plots enable plotting multivariate (normalized and mean-centered) data. However, as the axes are no longer adjacent to one another, the variable values in each sample are connected to form a closed line/polygon instead of the open lines found in parallel coordinates. Furthermore, unlike parallel coordinate plots, the time dimension can be captured via an additional axis, which is normal to the plot plane at the center of the plot[24] (Figure 3a).

Making use of the fact that each sample is a polygon as opposed to an open line, samples in time can be represented by stacking multiple polygons on top of one another. The result is a 3-D figure resembling a cylinder comprised of many



(a) Polygons represent normal operation and are used to generate inner and outer hulls

(b) New sample acquired at $t = 10min$ (dashed polygon) violates hulls and is considered a fault

Figure 4. Inner and outer hulls (shown in red and, respectively, light blue in Figure 4b) for the time-explicit Kiviat plot can be defined based on upper and lower bounds of the confidence interval for each variable.

Assuming that samples for $t \leq 9$ min represent normal operation, the *normal* operating region is defined as the shell between the inner and outer convex hulls of the plots for $t \leq 9$ min. The sample collected at t=10 min is outside the normal operating region and is an indication of faulty operation. Data from Appendix A (Samples 7–10). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

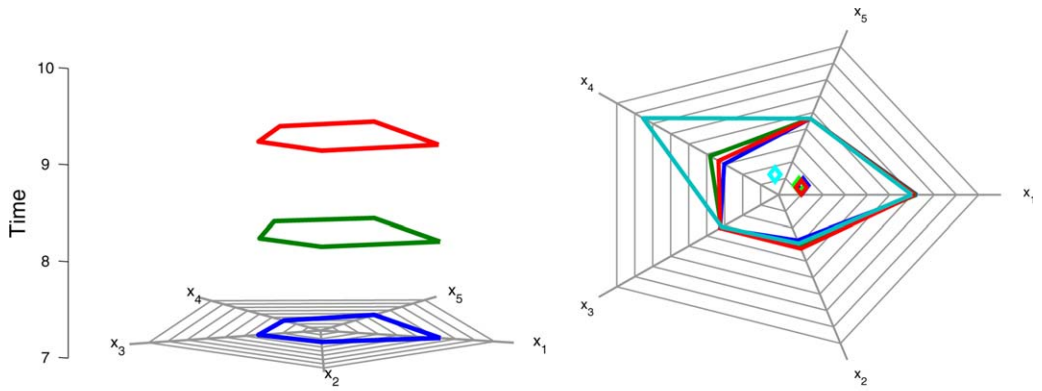


Figure 5. The centroid of each slice (diamonds in plot b) constitutes a single-point, multivariate representation of each data slice.

Data from Appendix A (Samples 7–10). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

polygons (Figures 3b, c). The 2-D polygon of the Kiviat plot at one sample time can, therefore, be considered a "data slice" corresponding to the respective time sample of the time series data. A detailed example of constructing a time explicit Kiviat plot is provided in Appendix B.

Such time-explicit Kiviat plots have found applications in computer science, typically for the visualization of software performance.[25,26] As of yet, we are not aware of any applications in which they have been used for fault detection, a topic that is explored in the remainder of the article.

Fault Detection Using Time-Explicit Kiviat Plots

The applications of time-explicit Kiviat plots that were mentioned above are typically static, in the sense that they consider a dataset whose membership does not evolve in time.

In this work, we note that time-explicit Kiviat plots can be generated and updated in real-time, as measurements from a physical process are collected. The time horizon of the plot is defined in terms of the number of data slices to be plotted at a given time (i.e., the height of the cylinder). Then, once a new scan of the data is obtained, the corresponding data slice is added at the top of the plot, while the oldest data slice is removed from the bottom in a "first-in-first-out" fashion. The time horizon is user-defined and dependent on the time scale and resolution desired.

Univariate approach

The above observation opens the possibility of carrying out fault detection, based on evaluating the deviation of the current data from a desired or nominal operating state. We begin with defining the state vector of a process from a geometric point of view, followed by establishing criteria for the nominal operation of a process and deviations therefrom.

Referring to Figures 3a–c, we note that the state of the process (assuming for now that all the variables are represented in the time-explicit Kiviat plot) is fully defined in terms of the location of the data points on each of the axes.

Thus, confidence limits for each state variable can be defined using available methods (e.g., defining upper and lower bounds for each variable).[27–29] Uniting all the upper

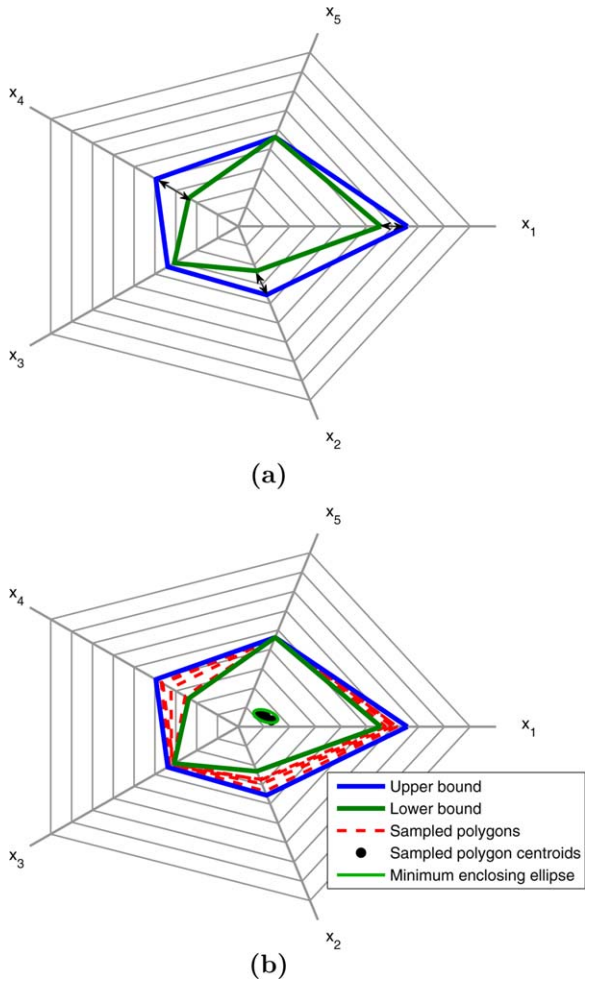


Figure 6. (a) Extrema of confidence ellipsoid projected in time-resolved Kiviat plot. Black arrows indicate the variable range for each variable. Blue and green lines are the extrema of the confidence ellipsoid. (b) Sampled points within the annular region (in red) are used to generate the confidence ellipse.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

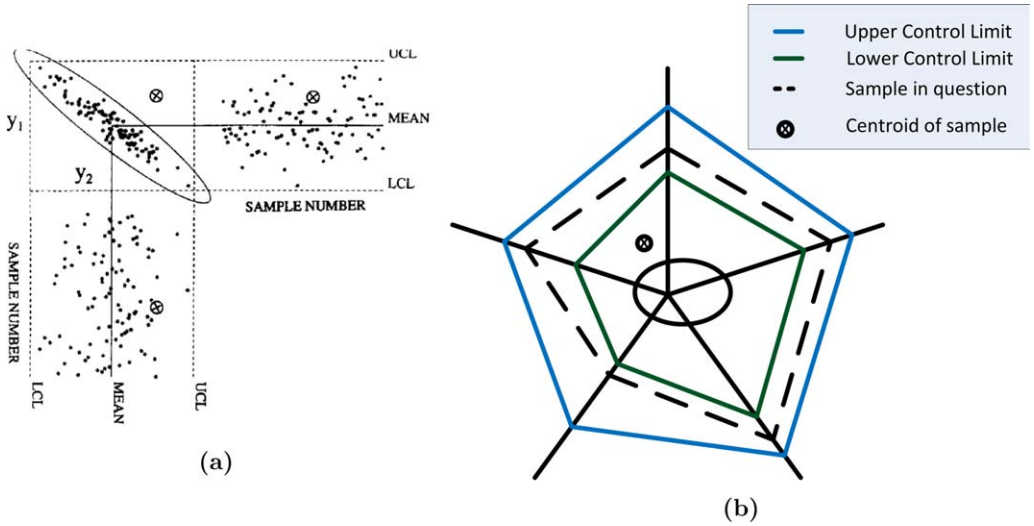


Figure 7. Univariate vs. multivariate control limits: (a) Univariate control limits suffer from "blind spots" in a multivariate setting: a data sample (marked ⊗) can be within the control limits from the perspective of every variable on the respective univariate control charts, but fall outside the multivariate confidence region.

Reprinted with permission from J. F. MacGregor and T. Kourti, Contr Eng Prac, Statistical process control of multivariate processes, 1995, Vol. 3(1), 403–414, © Elsevier. (b) The proposed fault detection framework can be interpreted from the perspective of an *n*-dimensional control chart. Here, too, the sample represented in dashed line falls within the control limits of each individual variable, but the corresponding centroid (marked ⊗) is not within the multivariate confidence ellipse. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

limits and, respectively, all lower limits of these confidence intervals results in an outer and, respectively, inner hull, which define a confidence region for the operation of the process (Figure 4). Thus, a data point whose Kiviat slice falls outside this confidence region is potentially indicative of a faulty behavior.

This approach to fault detection is then completely equivalent to that discussed above in the context of parallel coordinates (Figure 2). However, owing to Kiviat plots using a closed polygonal line to represent each data scan (rather than an open line as is the case in parallel coordinate plots), the ribbon-like confidence region used in the parallel coordinate framework is transformed into an annular region as shown in Figure 4.

Multivariate analysis

Defining the confidence region for an entire process based on confidence limits for individual variables amounts to a univariate analysis. Interactions between variables are, however, important in the occurrence (and detection) of process faults, and the potential of univariate analyses (i.e., monitoring multiple process variables based on their individual control limits) for reaching "blind spots" has been recognized early on (see, e.g., the classic article by Kourti and MacGregor[30]).

Dunia et al.[14] have addressed the issue by solving a series of optimization problems to define the representation of multivariate constraints (either linear or quadratic) in parallel coordinates. In this work, we propose a different approach, which exploits a unique feature of time-explicit Kiviat plots. Specifically, the fact that each data point is represented as a closed polygonal line allows for calculating a center of gravity of the corresponding data slice (Figure 5), which then constitutes a multivariate representation of the process state at that time instant.

We compute the centroid positions in a 2-D Cartesian coordinate system, whose origin is at the center of the radial plot. For an n-dimensional dataset with m samples, the positions of the n polygon vertices translate trivially to this Cartesian coordinate system as $\{X_i, Y_i\}$, $i \in \{1, \ldots, n\}$, and the coordinates of the centroid for data point $j \in \{1, \ldots, m\}$ can be determined as

$$X_{\text{centroid},j} = \frac{\sum_{i=1}^{n} X_{i,j}}{n}$$

$$Y_{\text{centroid},j} = \frac{\sum_{i=1}^{n} Y_{i,j}}{n} \quad (1)$$

Since in our framework the data in a set are always normalized prior to plotting (See Appendix B), the steady-state region of the data generally consists of near-regular polygons centered around the intersection of the axes of the Kiviat plot. This in turn means that the centroids of the steady-state region are always clustered near this intersection point, with slight deviations from point to point due to noise.

In the case of a fault, one or several process variables deviate from their steady-state, nominal values. As a consequence, the corresponding vertices of the data representation in the time-explicit Kiviat plot will change position and shift the shape of the polygon representing the respective data sample. This in turn shifts the location of the centroid of the polygon.

A fault is thus signaled by the shift in position of the centroids of the data samples corresponding to the faulty state. Therefore, assuming that the process has a single, nominal, steady state, the centroids corresponding to normal operation form a single, tight cluster (which, based on the data preprocessing is located at the intersection of the axes of the Kiviat plot), whereas the centroids corresponding to data collected when the process deviates from this nominal state lie outside this cluster.

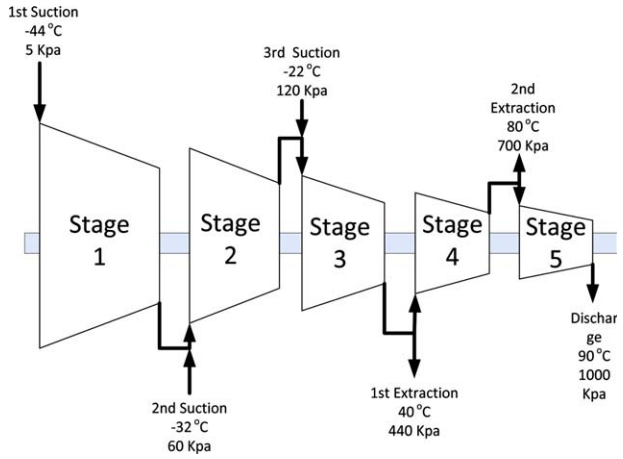


Figure 8. Schematic of compressor system.[21]

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Based on this observation, time-explicit Kiviat plots lend themselves naturally to defining the following fault detection scheme:

- Define a confidence region around the cluster of centroids corresponding to the nominal operating state.
- Monitor current process operating data and represent them on the time-explicit Kiviat plot.
- If the centroid of the representation of a newly acquired data point is not within the aforementioned confidence region, a fault is declared.

In what follows, we describe the steps of a procedure for computing a confidence region for the centroids in the Kiviat plot.

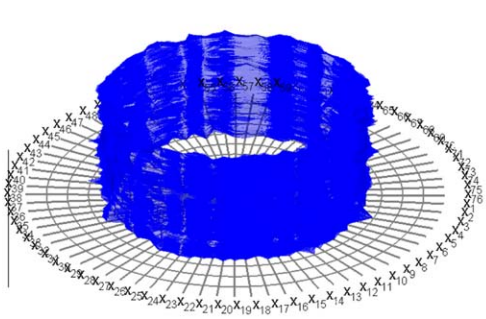
Step 1. Assume that matrix $X \in R^{m \times n}$ (which contains m time-resolved measurements of n process variables) represents a "golden period"[31] of steady-state operation of the process with good performance. We then define the covariance matrix $\Sigma = XX^{\top}$ and compute its eigenvalues λ_i and eigenvectors $\mathbf{v}_i$, $i \in \{1, \ldots, n\}$, that is

$$\lambda \mathbf{v} = \Sigma \mathbf{v} \tag{2}$$

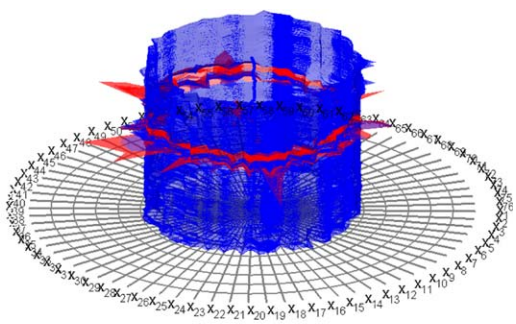
Step 2. Use the λ and $\mathbf{v}$ values to define an n-dimensional confidence ellipsoid around the steady-state operating region. The equation for the n-dimensional confidence ellipsoid can be written as

$$(x - \bar{x})^{\top} \Sigma^{-1} (x - \bar{x}) = 1 \tag{3}$$

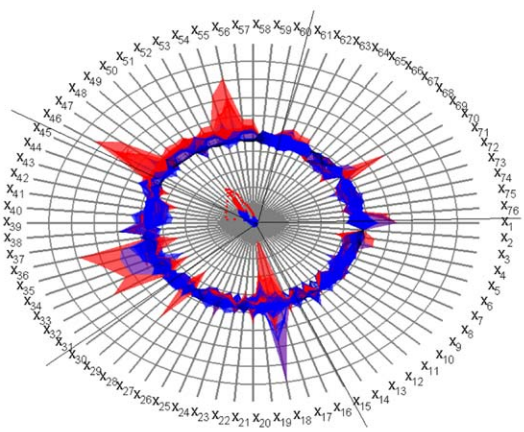
Specifically, the coordinates of the center of the ellipsoid will be given by the vector $\bar{X} = [\bar{x}_1, \ldots, \bar{x}_n]$ of the means of each variable, the orientation of the axes of the ellipsoid in the



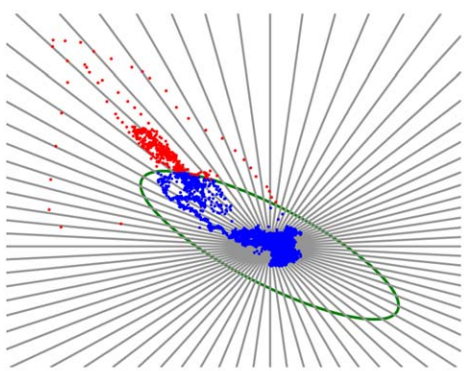
(a) Time-explicit Kiviat plot with no fault present



(b) Time-explicit Kiviat plot of a data set with a fault



(c) Top-view of the Kiviat plot when a fault is present



(d) Centroid plot showing confidence ellipse and the deviation of the centroids when a fault is present

Figure 9. (a) High-dimensional time-explicit Kiviat plot of all variables in a fault-free compressor dataset. (b–d) High-dimensional time-explicit Kiviat plot of all variables in compressor dataset 3, where a fault was present.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 1. Fault Detection Times and Early Detection Time (EDT) for 30 Compressor Surge Datasets

Dataset#	EDT (min): PCA with Kiviat Diagram	EDT (min): Raw Data with Kiviat Diagram	EDT (min): PCA T^2 (No Kiviat Diagram or Visualization)	EDT (min): PCA Q (No Kiviat Diagram or Visualization)
1	31	31	31	
2	42	42	42	42
3			1768	1773
4	748	757	716	585
5	−7	45	−68	−67
6	44	44	44	45
7	52	52	52	53
8	1335	1337	1336	1351
9	325	325	42	84
10	1434	878	33	33
11	721	440	70	41
12	229	143	62	33
13	1519	1335	1337	1332
14	39	43	39	36
15	877	932	263	−13
16	17	53		−6
17	1369	1410	1307	135
18	35	32	35	36
19	26	26	25	26
20			2463	2116
21	1079	1093	206	
22	61	34	103	114
23	1133	1109	180	180
24	561	575	143	204
25	3	37	−15	−28
26	−52	−50	−50	−49
27	−27	−64	−28	−62
28	33	−43	−59	−41
29	25	25	25	−25
30	178	176	58	102

A negative value indicates that the fault was detected after the operator-estimated fault time. Blank cells indicate that no fault was detected by the method.

n-dimensional hyperspace is provided by the eigenvectors $\mathbf{v}$, while the length of the axis is determined by the eigenvalues of the covariance matrix. Specifically, axes lengths l_i are scaled using the critical value κ of the χ^2 distribution that corresponds to the desired confidence level of the ellipsoid,

$$l_i = 2\sqrt{\kappa\lambda_i} \quad \forall i \in \{1 \ldots n\} \tag{4}$$

Step 3. Project the n-dimensional ellipsoid on the radial plot plane to define an appropriate confidence region for the centroids. To do so, we observe that the extremes of the n-dimensional ellipsoid in any given dimension can be represented on the Kiviat plot, and describe an annular region (Figure 6a). Note that this region will likely be different from the univariate confidence region described above.

Step 4. The multivariate confidence ellipse is then computed from the locus of the centroids of polygons contained within the annular region. We use a sampling-based procedure for determining this ellipse (Figure 6b). Specifically, we sample the annular region at random, by generating data scans using values uniformly distributed within the multivariate bounds of each variable and thus located inside the annular region.

It is possible that polygons situated close to the edges of the annular region would in effect be outside the confidence ellipsoid. To prevent this, each random polygon is verified to correspond to a point inside the confidence ellipsoid in the n-dimensional by using the following steps:

a. Apply the transformation matrix W^{-1} to the coordinates Y of the randomly generated polygon, to obtained the transformed coordinates Z

$$Z = YW^{-1} \tag{5}$$

where

$$W = \mathbf{v}\sqrt{\lambda} \tag{6}$$

b. Compute the norm $D = ||Z||$ and compare the resulting value with the radius of the unit sphere. Then, if $D \leq 1$, the randomly generated polygon corresponds to a point *within* the confidence ellipsoid and is retained for computing the confidence ellipse as described below. Else, the polygon is discarded and a new polygon is generated.

c. Repeat the procedure until the prescribed number of random polygons (typically, 5000) is reached.

The calculation of the minimum area enclosing ellipse,[32] of center c and axes lengths given by the matrix A, $(X-c)^\top A(X-c)=1$, is an optimization problem, aimed at minimizing

$$\underset{A,c}{\text{minimize}} \ \log(\det(A))$$

$$\text{subject to } (P_i-c)^\top A(P_i-c) \leq 1 \ \ i=1,2 \tag{7}$$

where P is the matrix of centroid locations.

The application of this procedure to a simple process with three variables is demonstrated in Appendix C.

Remark 1. *Earlier work by Kourti and MacGregor[30] has shown that univariate control charts may result in "blind spots," whereby the current state of a (bivariate) process falls within the control limits of both variables, but not within the joint normal operating region, resulting in a failure to detect an abnormal event (Figure 7). The fault detection framework discussed in this article can be interpreted from this perspective as a multivariate control chart, where the confidence ellipse denotes a n-dimensional normal operating region.*

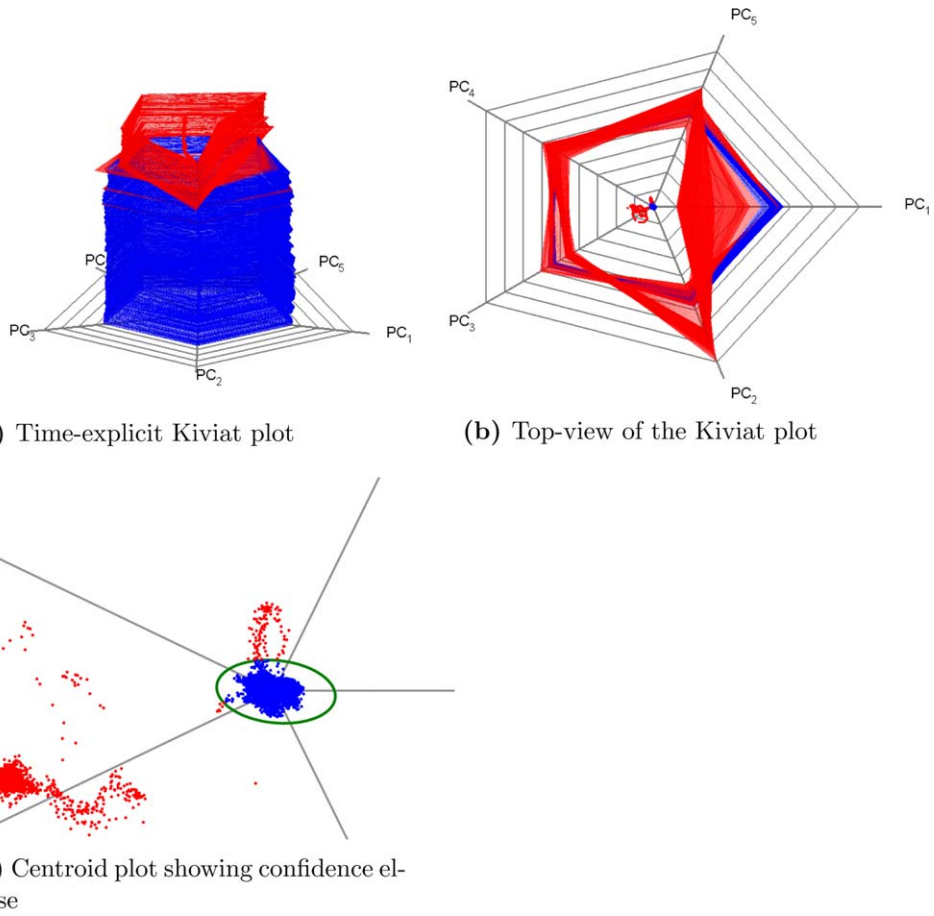
As a consequence, this framework is expected to have superior fault detection performance to that provided by the univariate approach described earlier in the article, and, implicitly, to the fault detection performance provided by setting individual upper and lower control limits on each variable represented in a parallel coordinate plot.

Representing large datasets and reducing dimensionality of the data

Process datasets often contain a large number of variables; high dimensionality makes data analysis challenging and dimensionality reduction is therefore desirable.

This issue can be addressed in several ways. First, assuming that sufficient process knowledge is available, the subset of the variables to be plotted can be user-selected. This approach is, however, less practical when an unknown/unfamiliar process is under consideration, in which case a more rigorous approach for reducing the dimensionality of the dataset should be used.

PCA can be used to reduce the dimensionality of the data. PCA accomplishes this through the linear transformation of



(a) Time-explicit Kiviat plot

(b) Top-view of the Kiviat plot

(c) Centroid plot showing confidence ellipse

Figure 10. Time-explicit Kiviat plots of principal components in compressor dataset 8.

Note that by reducing the number of axes to plot, the graphical load is significantly reduced and it is very clear which principal components are changing. However, each axis loses physical meaning as principal components are being plotted instead of variables. Nevertheless, significant loading values for a given direction may be physically meaningful and provide a starting point for basic fault diagnosis purposes. (a) Time-explicit Kiviat plot. (b) Top-view of the Kiviat plot. (c) Centroid plot showing confidence ellipse. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

variables into principal components. A subset of the principal components is selected for analysis, generally based on the total variance captured.[33] PCA has been used extensively in literature to conduct fault detection and isolation (see e.g., Russell et al.[34] and Wise et al.[35]), and excellent reviews on its use are available.[30,36,37] Returning to the proposed time-explicit Kiviat framework, in the case of large-scale datasets with a large number of variables, it may be more practical to represent the PCA scores rather than the data themselves. Typically, the number of components/scores to be represented is selected based on the amount of variance that is desired to be captured. If there are practical limitations on the number of coordinates that can be used in the Kiviat plot, a fixed/limited number of scores can be employed.

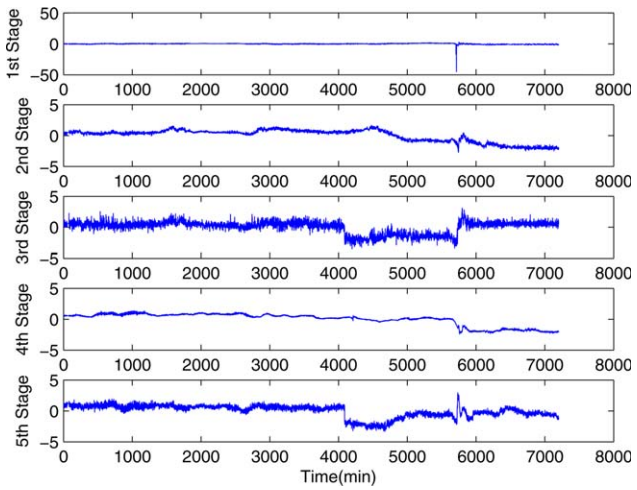


Figure 11. Plots of five flow variables values corresponding to each of the five stages.

Values have been normalized. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Remark 2. *We note here that two similar polygons—that is, polygons that are of the same shape and orientation, but of different sizes—have the same centroid. In principle, this could impede on the proposed fault detection framework. However, this is highly unlikely to occur in practice as it would require that the variables in the dataset to be completely correlated and have the same scaling. In a different vein, for cases where multiple variables are closely correlated,*[38–40] *PCA can be used to alleviate collinearity. The number of correlated variables used in fault detection can also be reduced based on process knowledge (this includes, e.g., considering only one of several redundant sensors).*

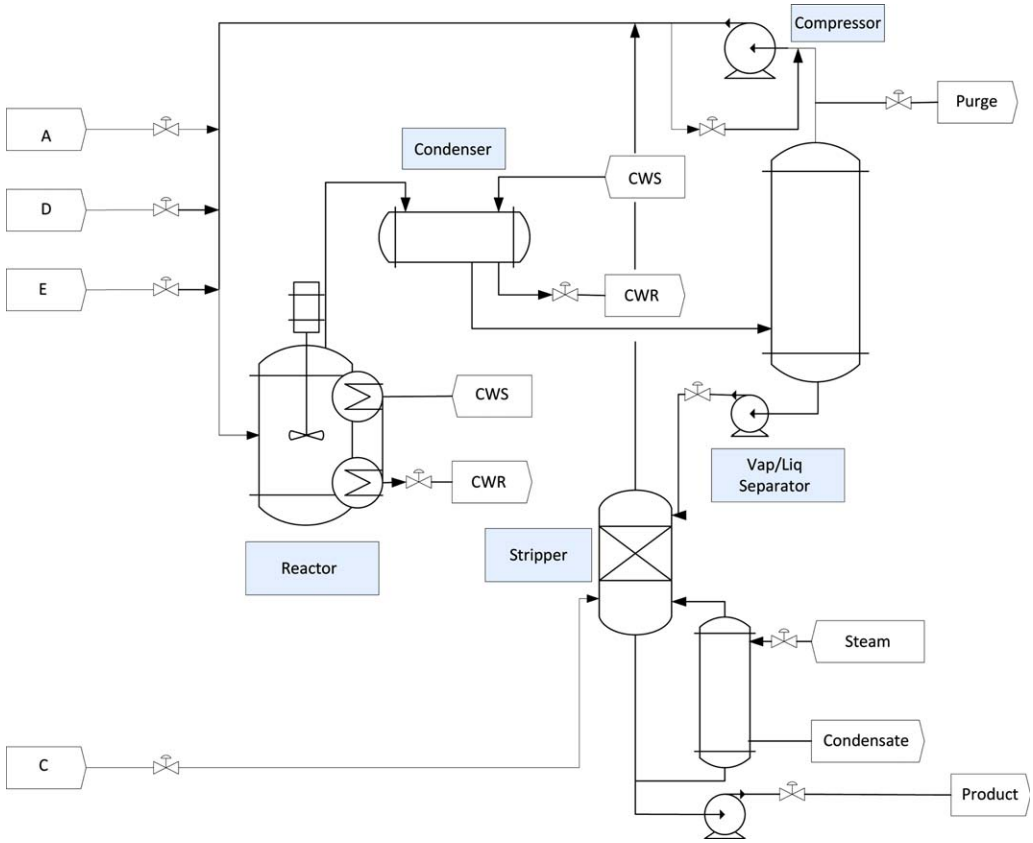


Figure 12. Schematic of the Tennessee Eastman Process.[46] [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Remark 3. *Although the same centroid can be calculated from an infinite number of polygons, the set of polygons relevant to the representation in this framework is in fact bounded by the upper and lower (operating) limits of each variable in the process.*

Remark 4. *Using PCA to reduce the dimensionality of the data allows for extending the proposed framework to fault isolation using contribution plots,[41] which provide an indication of variables contributing to the deviation of the process from its normal operating state. Furthermore, other dimension reduction methods can be used, along with the corresponding fault diagnosis methods.[42–44] These extensions constitute the scope of our future research.*

Case Studies

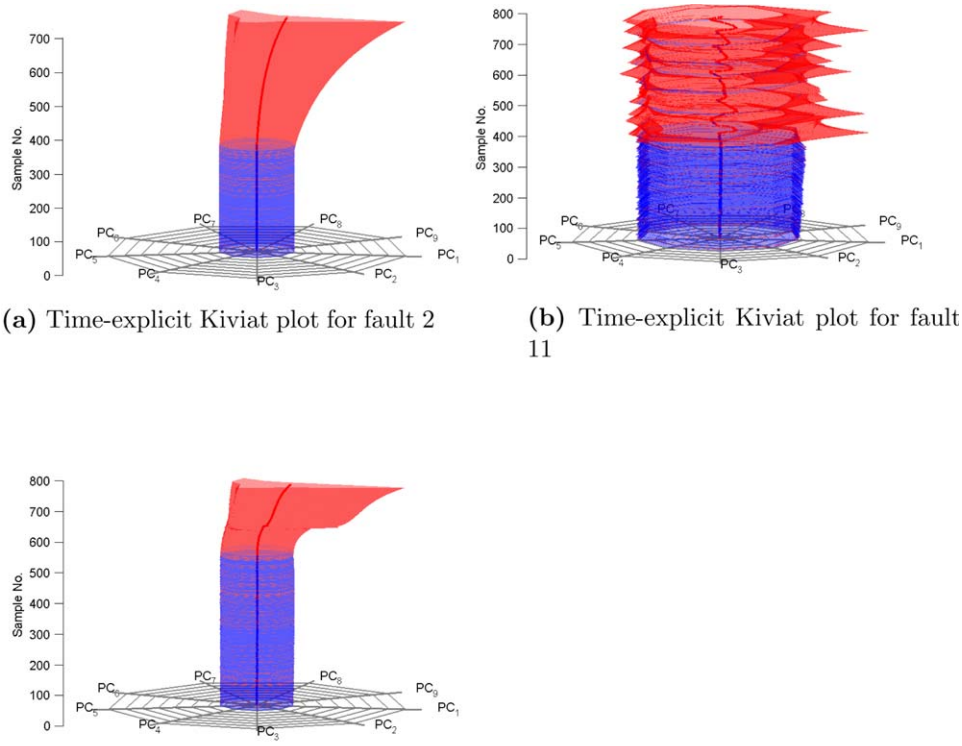
Compressor surge

Surge is an undesirable compressor condition, whereby the gas flow in the system reverses due to high back pressure. This is primarily due to low flow rates through the compressor system relative to the power input.[45] As a result, the compressor back pressure may be higher than the output pressure, resulting in the reversal of flow. This leads to machine vibration and may cause irreversible physical damage. Recovery from a surge event requires all compressed gas to be removed from the compressor system and restarting the compression process.[21]

Anticipating and preventing the occurrence of such events is, therefore, of very high interest to process operators. In this section, we analyze 30 surge event datasets occurring in a industrial five-stage compressor (Figure 8). In each case, the timing of the surge event was determined based on operator experience. Each dataset captures a single surge event, and the process historian database was queried for all the variables pertaining to the compressor operation for 4 days prior to, and 1 day after the occurrence of the surge event. Data are sampled

Table 2. List of Available Faults in TEP[51]

Fault No.	Description	Type
1	A/C feed ratio, B Composition constant (Stream 4)	Step
2	B Composition, A/C ratio constant (Stream 4)	Step
3	D feed temperature (Stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (Stream 1)	Step
7	C header pressure loss—reduced availability (Stream 4)	Step
8	A, B, C feed composition (Stream 4)	Random variation
9	D feed temperature (Stream 2)	Random variation
10	C feed temperature (Stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	
17	Unknown	
18	Unknown	
19	Unknown	
20	Unknown	



(a) Time-explicit Kiviat plot for fault 2

(b) Time-explicit Kiviat plot for fault 11

(c) Time-explicit Kiviat plot for fault 20

Figure 13. High-dimensional time-explicit Kiviat plots of principal components for several TEP faults.

(a) Time-explicit Kiviat plot for fault 2. (b) Time-explicit Kiviat plot for fault 11. (c) Time-explicit Kiviat plot for fault 20. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

at 1 min intervals, and each dataset contains 7200 samples.[21] The number of variables in each dataset is 76, consisting of flow, pressure, and temperature sensor readings for each compressor stage. Communication with the industry partner that provided the data indicated that the operation during the 4 days prior to each surge event appeared to be at steady state.

Time-explicit Kiviat plots of the data (both fault-free and faulty) from one of the available datasets are presented in Figures 9a, b. To facilitate fault diagnosis by operators, the sequence in which the variables are displayed may be reordered to follow the main process flow from compression stage 1 through stage 5 (note that this may not be possible for more complex flow sheets involving, e.g., multiple recycle streams). PCA can be performed on the data to reduce the number of variables that need to be plotted, an example of which is seen in Figure 10.

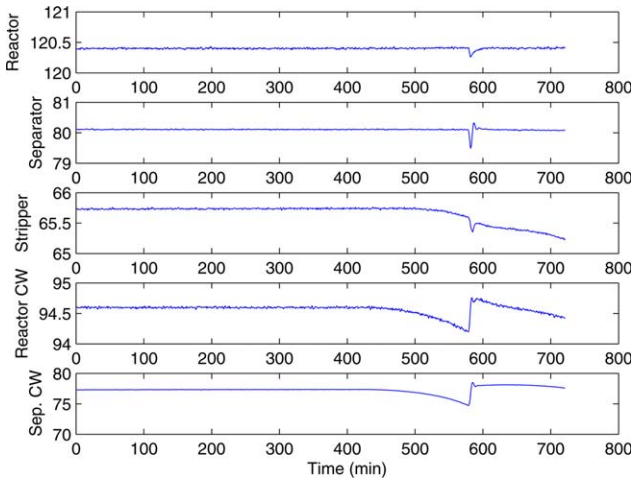


Figure 14. Temperature plots of various process units for Fault 20.

All temperatures are given in °C. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

A plot of the flow rates corresponding to the five stages of the compressor is provided as well in Figure 11. We constructed a 95% confidence ellipse using the steady-state samples from all 30 datasets as training data, and used the framework described above to analyze the advent of surge. The results of fault detection using the proposed method are shown in Table 1, which provides the time required for fault detection (i.e., the time instant where the centroids violate the normal operating region defined by the confidence ellipse). For reference, the time when the surge event occurred was estimated by the operators to be $t = 5760$ min in each dataset, and we define a performance indicator, the early detection time (EDT), as the difference between the operator-estimated time of fault occurrence and the time of fault detection by the proposed method. A positive EDT is evidently desirable, and is illustrative of the predictive capabilities of the proposed method. Negative detection times for a given fault suggest that the fault would have been detected by the proposed method after it was identified by process operators.

No false alarms (i.e., flagged samples that cannot be related to the surge event) were raised for the majority of datasets. The false alarm rate for the data is 0.77%.

Table 3. Fault Detection Time for Tennessee Eastman Process

			Detection Time (Min)											
			Russell et al.[34]							Tamura and Tsujita[47]		Zhang[48]		
Fault Number	Proposed Method	Avg. Time of All Other Methods	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q	PCA Q	PCA T^2	KPCA	KICA	Imp. KICA
1	3	13	21	9	18	15	6	9	6	21	21	15	9	6
2	8	41	51	36	48	39	39	45	75	27	33	30	36	33
3	17													
4	2	234		9	453	3	1386	3				9	6	6
5	2	8	48	3	6	6	3	3	0			3	3	3
6	2	9	30	3	633	3	3	3	0	15	15	3	3	0
7	2	5	3	3	3	3	3	3	0	15	15	3	3	0
8	46	64	69	60	69	63	60	60	63	54	60	75	69	60
9	201	899								894	903			
10	52	131	288	147	303	150	75	69	132	99	150	60	51	42
11	39	271	912	33	585	21	876	33	81			69	57	45
12	14	15	66	24	9	24	6	6	0			9	6	3
13	124	110	147	111	135	120	126	117	129	48	48	123	114	99
14	8	6	12	3	18	3	6	3	3			3	3	3
15	32	865		2220			2031					27	27	21
16	56	304	936	591	597	588	42	27	33	153	624	27	21	9
17	112	89	87	75	84	72	81	60	69	189	192	57	51	51
18	113	242	279	252	279	252	249	237	252	237	252	222	198	195
19		140				246		33						
20	180	189	261	261	267	252	246	198	216	39	45	177	165	135

Blank cells indicate that no fault was detected

PCA was also applied to this dataset (see a visualization example in Figure 10) and the scores used for fault detection. The fault detection times and EDT when using PCA scores are provided in Table 1. For comparison purposes, we have included detection times when using standard PCA T^2 and Q statistics, as well as when process variables are directly used in the framework. Five principal components were used, which captured 74% of the variance in the steady-state data. The number of principal components selected was based on a minimum threshold of 70% for the amount of variance captured.

Applying PCA, a fault was detected in 28 of the 30 datasets, in 25 of which the EDT was positive, indicating that the fault was noticed early (i.e., prior to operators becoming aware of its advent) and preventative measures could be taken. It is key, however, to note that the representation of data loses physical meaning after conducting PCA—in Figure 9, the physical variables could be arranged in sequence of their appearance on the flow sheet, but if PCA is applied, each principal component is a combination of multiple variables and the visual representation is not physically meaningful.

In the case of using the raw data directly, a fault was detected in 28 of the 30 datasets. Furthermore, in 25 cases, the EDTs were positive. When using PCA T^2, a fault was detected in 29 of the 30 cases, but there were five cases where the EDT

Table 4. Missed Detection Rates for Tennessee Eastman Process

		Missed Detection Rates (Lower is Better)									
		Russell et al.[34]							Zhang[48]		
Fault Number	Proposed Method	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q	KPCA	KICA	Imp. KICA
1	0.018	0.008	0.003	0.006	0.005	0.001	0	0.003	0	0	0
2	0.018	0.02	0.014	0.019	0.015	0.011	0.010	0.026	0.02	0.02	0.02
3	0.542	0.998	0.991	0.991	0.990	0.981	0.986	0.985	0.96	0.94	0.92
4	0.040	0.956	0.038	0.939	0	0.688	0	0.975	0.91	0.18	0.19
5	0.002	0.775	0.746	0.758	0.748	0	0	0	0.75	0.71	0.71
6	0.005	0.011	0	0.013	0	0	0	0	0.01	0	0
7	0.007	0.085	0	0.159	0	0.386	0	0.486	0	0	0
8	0.102	0.034	0.024	0.028	0.025	0.021	0.016	0.486	0.03	0.03	0.02
9	0.829	0.994	0.981	0.995	0.994	0.986	0.993	0.993	0.96	0.95	0.95
10	0.138	0.666	0.659	0.580	0.665	0.166	0.099	0.599	0.57	0.19	0.20
11	0.171	0.794	0.356	0.801	0.193	0.515	0.195	0.669	0.76	0.19	0.18
12	0.069	0.029	0.025	0.01	0.024	0	0	0.021	0.03	0.03	0.02
13	0.283	0.06	0.045	0.049	0.049	0.047	0.040	0.055	0.06	0.05	0.05
14	0.040	0.158	0	0.061	0	0	0	0.122	0.21	0	0
15	0.188	0.988	0.973	0.964	0.976	0.928	0.903	0.979	0.95	0.95	0.94
16	0.138	0.834	0.755	0.783	0.708	0.166	0.084	0.429	0.70	0.20	0.20
17	0.266	0.259	0.108	0.240	0.053	0.104	0.024	0.138	0.26	0.05	0.05
18	0.266	0.113	0.101	0.111	0.100	0.094	0.092	0.102	0.10	0.10	0.09
19		0.996	0.873	0.993	0.735	0.849	0.019	0.923	0.97	0.25	0.23
20	0.411	0.701	0.570	0.644	0.558	0.44	0.342	0.547	0.59	0.42	0.50

Blank cells indicate that no fault was detected.
Missed detection rates not available for Tamura and Tsujita.[47]

Table 5. False Detection Rates for Tennessee Eastman Process

	False Detection Rates (Lower is Better)	
	Proposed Method	0.032
Russell et al.[34]	PCA T^2	0.014
	PCA Q	0.16
	DPCA T^2	0.006
	DPCA Q	0.281
	CVA T^2	0.083
	CVA T_r^2	0.126
	CVA Q	0.087
Zhang[48]	PCA T^2	0.005
	KPCA T^2	0.0152
	KICA T^2	0.0031
	Improved KICA T^2	0.0027

False detection rates not available for Tamura and Tsujita.

was negative, indicating that the faults would not have been anticipated prior to detection by operators. When using fault detection based on the PCA Q statistic, a fault was detected in 28 of 30 datasets, but the EDT was negative in eight cases, indicating again that detection by the operators would have preceded the indication of this fault detection mechanism.

We note here that for the above case study, the details of the compressor operation and the conditions surrounding any given surge event were not provided; furthermore, no cleaning (such as the elimination of outliers) was performed on the data. It is, therefore, difficult to determine reasons as to why the method performed well in some cases, but failed to predict or detect a surge in other cases. Potential (but not confirmed) reasons include human error in estimating surge time as well as database errors when recording the data.

Benchmarking with the Tennessee Eastman process simulator

The Tennessee Eastman Process (TEP)[46] was used to further validate the fault detection method proposed in this article. The motivation behind this exercise is to test the ability of the proposed method to detect known faults; furthermore, the TEP constitutes a very useful benchmark given that a number of fault detection studies have been published in the literature. In this article, we will refer to the work of Russell et al.[34] and Tamura and Tsujita,[47] who implemented and tested a complement of linear PCA-based fault detection methods, as well as to the work by Zhang,[48] where nonlinear (kernel) PCA is used. The TEP model was originally provided as a black-box FORTRAN code,[30] and ported to MATLAB at a later date.[49] The latter version was used in this study. The process is open-loop unstable and two temperature cascade control loops as suggested by McAvoy and Ye[50] were used to stabilize the process. A process diagram is provided in Figure 12.

The TEP simulator has provisions for applying 20 faults (of which five are not described specifically); they are listed in Table 2.

In performing our fault detection tests, we proceeded as follows: for each fault, the simulator was run for at least 12 h (720 min), and an individual fault was injected at $t = 300$ min. The random seed of the noise signal generator was changed between every run. Fault detection using the proposed framework was conducted on the data obtained from the simulation, with the steady-state operating region being defined based on training data from a fault-free run of 12 h. As in the previous case study, using a threshold of 70% for the variance captured, the number of principal components selected was nine, and the variance captured for normal (fault-free) operation was 70.1%. A 95% confidence ellipse was used for all cases. Sample Kiviat diagrams are provided in Figure 13.

Sample plots of the temperatures throughout the process are presented in Figure 14.

The results of our fault detection analysis are provided in Table 3. The data reported in the aforementioned papers are included for comparison. The detection time refers to the time required for a fault to be reported after it was injected at $t = 300$ min.

The data in Table 3 show that the proposed fault detection method performs very well compared to other methods in terms of fault detection time. One fault (fault 19) was only detected by one of the methods, which can likely be attributed to the small impact that it has on the system states and outputs. The false detection and missed detection rates were also computed. A fault is only considered detected when two consecutive samples are flagged as a fault. As the time when a fault is implemented is known, a binary approach is taken—all samples taken prior to $t = 300$ min are considered to be normal operating data, and all samples taken after $t = 300$ min are considered faulty data. Therefore, any samples before $t = 300$ min that are flagged as a fault result in a false detection, while any sample after that point in time that is not detected is considered a missed detection. This definition is used to calculate the false detection and missed detection rates and is similar to the method used by Zhang to calculate these parameters.[48] From Table 4, we note that our proposed method compares favorably with the other approaches used in the literature.

Table A. Data Used for Figures 1–5

Sample Number	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$
1	0.95	0.03	313.91	289.41	1
2	0.90	0.098	315.26	289.42	1
3	1.01	0.04	313.07	289.23	1
4	0.89	0.02	313.08	288.89	1
5	0.92	0.019	311.81	288.11	1
6	0.96	0.01	311.3	289.88	1
7	0.98	0.04	313.45	288.85	1
8	0.98	0.02	312.46	292.88	1
9	0.96	0.06	313.31	290.47	1
10	0.94	0.01	311.82	312.04	1
11	0.92	0.07	313.56	309.72	1
12	0.88	0.13	335.93	309.84	1
13	0.01	0.97	408.07	312.52	1
14	0.14	0.81	381.81	312.94	1
15	0.15	0.86	383.75	310.07	1
16	0.02	0.92	383.26	313.37	1
17	0.19	0.86	383.40	310.95	1
18	0.07	0.91	384.07	310.74	1
19	0.16	1.00	385.13	314.38	1
20	0.18	0.94	382.83	293.07	1
21	0.03	0.95	385.17	291.81	1
22	0.52	0.61	325.91	290.8	1
23	0.97	0.27	312.15	288.92	1
24	0.89	0.12	312.82	289.77	1
25	0.96	0.16	311.84	293.64	1
26	0.96	0.01	310.87	285.96	1
27	1.0132	0.02	314.11	288.73	1
28	1.01	0.15	313.35	289.62	1
29	0.90	0.07	312.14	290.3	1
30	0.93	0.05	312.77	286.43	1

As seen in Table 5, the false detection rate of the proposed method is in line with other methods, while providing improved fault detection times and lower missed detection rates.

Discussion and Perspective

The case studies presented above, using both data collected *in silico* from an industrial process simulator, and a rich complement of industrial datasets, demonstrate the effectiveness of the proposed method for fault detection. Along with fault detection capabilities, our Kiviat diagram-based framework can provide a time explicit representation of multivariate process data.

Our informal interactions with industrial plant personnel have shown that these multidimensional visualizations are easily understood and followed by process operators, and that operators can more easily detect the advent of a process fault based on them. The visualization of multidimensional data using time-explicit Kiviat diagrams was considered preferable to parallel coordinates or the use of multiple 2-D plots.

However, despite proven (and often superior) fault detection capabilities, the method is not without drawbacks. As with all data-driven mechanisms, it is susceptible to noise, so preprocessing of the data to remove outliers and noise may be necessary. Selecting a low number of principal components would also help reduce the effect of the noise. Similar to other data-driven methods, the proposed method is affected by the quality of the "model," which in this case consists of the confidence ellipse in the centroid representation. The correct construction of the ellipse depends on the selection of an appropriate steady-state region, and the proposed method may be less effective (e.g., resulting in an increased number of Type I and Type II errors) in processes where the operation is oscillatory or strongly affected by noise. This in turn affects the occurrences of false positive and false negatives during fault detection. Our experience suggests that the probability of false positives is relatively small when the process noise is low, and can be lowered for noisy data when filtering is appropriately applied. Under the same conditions, the probability of occurrence for false negatives is similarly small. As with many fault detection methods, slow, drifting disturbances are hard to detect in a timely fashion in a noisy environment.

Conclusions

In this article, we propose a novel method for the detection of process faults based on operating data. The method consists of creating a multivariate control chart for large datasets through the use of a time-dependent extension of Kiviat diagrams. PCA can be used to reduce the dimensionality of the data as well when needed. Through the use of centroids, we were able to capture snapshots of the process at any given point in time and monitor the process through the use of multivariate control charts. This is superior to the use of univariate control limits in parallel coordinates. An added benefit of the framework is a time-explicit visual representation of multivariate data, which was not possible using most traditional plotting methods.

Through extensive validation, we show that the proposed method performs comparably with existing FDI algorithms from the literature. Further, we showed that its application to real plant operating data for a multistage compressor could lead to anticipating the advent of systemic faults such as surge prior to these events being detected by operators.

Our studies show that the computational cost of data analysis within the proposed framework is reasonable, and that most of the computational burden (i.e., determining the confidence region) can be completed offline. This, together with favorable feedback received from operators, suggests that the proposed method has, in its present form, a strong potential for real-time implementation in plant operations.

Acknowledgments

This work was supported by Emerson Process Management and the Center for Operator Performance (COP). Michael Baldea was supported by the Moncrief Grand Challenges Award from the Institute for Computational Engineering and Sciences at The University of Texas at Austin. Guidance and support throughout the project from Mr. Terry Blevins of Emerson Process Management is greatly appreciated.

Literature Cited

1. Chien C, Chen L. Data mining to improve personnel selection and enhance human capital: a case study in high-technology industry. *Expert Syst Appl*. 2008;34:280–290.
2. Gagan M. 6 Uses of big data for online retailers. 2013. Available at http://www.practicalecommerce.com/articles/3960-6-Uses-of-Big-Data-for-Online-Retailers. Last Accessed: Sep. 28, 2015.
3. Andrienko G, Andrienko N, Wrobel S. Visual analytics tools for analysis of movement data. *SIGEX*. 2007;9(2):38–46.
4. Alsakran J, Zhao Y, Zhao X. Tile-based parallel coordinates and its application in financial visualization. *Proceedings of the SPIE, Visualization and Data Analysis 2010*, *Vol. 7530*, San Jose, CA, January 17, 2007.
5. Groves P, Kayyali B, Knott D, Van Kuiken S. The 'big data' revolution in healthcare. *McKinsey Quarterly*. 2013:2.
6. Venkatasubramanian V. Drowning in data: informatics and modeling challenges in a data-rich networked world. *AIChE J*. 2009; 55(1):2–8.
7. Kano M, Nagao K, Hasebe S, Hashimoto I, Ohno H, Strauss R. Comparison of multivariate statistical process monitoring methods with applications to the Eastman challenge problem. *Comput Chem Eng*. 2002;26(2):161–174.
8. Yoon S, MacGregor JF. Fault diagnosis with multivariate statistical models. Part I: using steady state fault signatures. *J Process Control*. 2001;11(4):387–400.
9. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part III: process history based methods. *Comput Chem Eng*. 2003;27(3):324–346.
10. Qin SJ. Survey on data-driven industrial process monitoring and diagnosis. *Annu Rev Control*. 2012;36(2):220–234.
11. Albazzaz H, Wang XZ. Historical data analysis based on plots of independent and parallel coordinates and statistical control limits. *J Process Control*. 2006;16(2):103–114.
12. Wang XZ, Medasani S, Marhoon F, Albazzaz H. Multidimensional visualization of principal component scores for process historical data analysis. *Ind Eng Chem Res*. 2004;43(22):7036–7048.
13. He QP. Multivariate visualization techniques in statistical process monitoring and their applications to semiconductor manufacturing. *SPIE 31st International Symposium on Advanced Lithography*. San Jose, CA: International Society for Optics and Photonics, 2006: 615506–615506.
14. Dunia R, Rochelle G, Edgar TF, Nixon M. Multivariate monitoring of a carbon dioxide removal process. *Comput Chem Eng*. 2014;60: 381–395.
15. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: part I: quantitative model-based methods. *Comput Chem Eng*. 2003;27(3):293–311.
16. Venkatasubramanian V, Rengaswamy R, Kavuri N. A review of process fault detection and diagnosis: part II: qualitative models and search strategies. *Comput Chem Eng*. 2003;27(3):313–326.

17. Keogh E, Kasetty S. *On the need for time series data mining benchmarks: a survey and empirical demonstration. Data Min Knowl Discov*. 2003;7(4):349–371.
18. Peng RD. A method for visualizing multivariate time series data. *J Stat Softw*. 2008;25:1.
19. Few S. Visualizing change - an innovation in time-series analysis. *Visual Business Intelligence Newsletter*. 2007. Available at https://www.perceptualedge.com/articles/visual_business_intelligence/visualizing_change.pdf. Last Accessed: Sep. 28, 2015
20. Inselberg A. *Parallel Coordinates*. US: Springer, 2009.
21. Dunia R, Edgar TF, Nixon M. Process monitoring using principal components in parallel coordinates. *AIChE J*. 2013;59(2):445–456.
22. Brooks RW, Mahoney A, Wilson J, Zhao N. Operator alarms are the first line of defence. In *IChemE Proceedings of Hazards XXIII/316*, Southport UK: Southport, 2012.
23. Kolence KW. The software empiricist. *ACM SIGMETRICS Perform Eval Rev*. 1973;2(2):31–36.
24. Tominski C, Abello J, Schumann H. Interactive poster: 3d axes-based visualizations for time series data. *Poster Compendium of IEEE Symposium on Information Visualization (InfoVis' 05)*, InfoVis, Minneapolis, MN, 2005.
25. Hackstadt ST, Malony AD. Visualizing parallel programs and performance. *IEEE Comput Graph Appl*. 1995;15(4):12–14.
26. Fanea E, Carpendale S, Isenberg T. An interactive 3d integration of parallel coordinates and star glyphs. *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*, Minneapolis, Minnesota, USA: IEEE, 2005:149–156.
27. Heij C, De Boer P, Franses PH, Kloek T, Van Dijk HK. *Econometric Methods with Applications in Business and Economics*. Oxford, England: Oxford University Press (OUP), 2004.
28. Dixon WJ, Massey FJ. *Introduction to Statistical Analysis*, Vol. 344. New York, USA: McGraw-Hill New York, 1969.
29. Wonnacott TH, Wonnacott RJ. *Introductory Statistics*, Vol. 19690. New York: Wiley, 1972.
30. MacGregor JF, Kourti T. Statistical process control of multivariate processes. *Control Eng Pract*. 1995;3(3):403–414.
31. Yu J, Qin SJ. Statistical mimo controller performance monitoring. Part I: data-driven covariance benchmark. *J Process Control*. 2008;18(3–4):277–296.
32. Moshtagh N. Minimum volume enclosing ellipsoid. *Convex Optimization*. 2005. Available at http://www.researchgate.net/publication/254980367_MINIMUM_VOLUME_ENCLOSING_ELLIPSOIDS. Last Accessed: Sep. 28, 2015
33. Ku W, Storer RH, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometr Intell Lab*. 1995;30(1):179–196.
34. Russell EL, Chiang LH, Braatz RD. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometr Intell Lab*. 2000;51(1):81–93.
35. Wise BM, Gallagher NB, Butler SW, White DD, Barna GG. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *J Chemometr*. 1999;13(3–4):379–396.
36. Kourti T, MacGregor JF. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometr Intell Lab*. 1995;28(1):3–21.
37. Nomikos P, MacGregor JF. Monitoring batch processes using multiway principal component analysis. *AIChE J*. 1994;40(8):1361–1375.
38. Bühlmann P, Rütimann P, van de Geer S, Zhang C. Correlated variables in regression: clustering and sparse estimation. *J Indian Soc Periodontol*. 2013;143(11):1835–1858.
39. García J, Salmerón R, García C, López Martín M. Standardization of variables and collinearity diagnostic in ridge regression. *Int Stat Rev*. 2015. doi: 10.1111/insr.12099
40. Brunsdon C, Charlton M, Harris P. Living with collinearity in local regression models. Accuracy 2012 - 10th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences. 2012, Florianopolis, SC, Brazil.
41. Misra M, Yue HH, Qin SJ, Ling C. Multivariate process monitoring and fault diagnosis by multi-scale PCA. *Comput Chem Eng*. 2002;26(9):1281–1293.
42. Kourti T, Nomikos P, MacGregor JF. Analysis, monitoring and fault diagnosis of batch processes using multiblock and multiway PLS. *J Process Control*. 1995;5(4):277–284.
43. He QP, Qin SJ, Wang J. A new fault diagnosis method using fault directions in Fisher discriminant analysis. *AIChE J*. 2005;51(2):555–571.
44. Chiang LH, Russell EL, Braatz RD. Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. *Chemometr Intell Lab*. 2000;50(2):243–252.
45. Cahill J. *Surge Control Considerations in Centrifugal Compressors*. Blog, Emerson Process Experts, 2010. Available at http://www.emersonprocessxperts.com/2010/04/surge_control_c/. Last Accessed: Sep. 28, 2015
46. Downs JJ, Vogel EF. A plant-wide industrial process control problem. *Comput Chem Eng*. 1993;17(3):245–255.
47. Tamura M, Tsujita S. A study on the number of principal components and sensitivity of fault detection using PCA. *Comput Chem Eng*. 2007;31(9):1035–1046.
48. Zhang Y. Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM). *Ind Eng Chem Res*. 2008;47(18):6961–6971.
49. Ricker NL. Tennessee Eastman challenge archive. Available at http://depts.washington.edu/control/LARRY/TE/download.html. Last Accessed: Sep. 28, 2015.
50. McAvoy TJ, Ye N. Base control for the Tennessee Eastman problem. *Comput Chem Eng*. 1994;18(5):383–413.
51. Lee J, Yoo C, Lee I. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chem Eng Sci*. 2004;59(14):2995–3006.
52. Henson M, Seborg DE. *Nonlinear Process Control*. New Jersey, USA: Prentice-Hall, Inc., 1997.
53. Hendengren J. Nonlinear model library. 2008. Available at http://www.hedengren.net/research/models.htm. Last Accessed: Sep. 28, 2015.
54. Said D. Radar plot. 2011. Available at http://www.mathworks.com/matlabcentral/fileexchange/33134-radar-plot. Last Accessed: Sep. 28, 2015.

Appendix A: Sample Data Used in Example

Table A provides the data used to generate Figures 1–5.

These data were obtained by simulating the simple CSTR model adapted from the textbook *Nonlinear Process Control* by Henson and Seborg[52]—the MATLAB model is provided by Dr. John Hendengren.[53] A step change (simulating a fault) was made at sample 10 that resulted in an increased temperature reading (Variable 4) that later propagated to the rest of the system, this was then corrected at sample 20 and the system returned to its original state. The first two variables are concentration readings, the next two columns are temperature readings, and the final column is a valve constant. Samples from this table were used to generate Figures 1–5.

Appendix B: Construction of Time Explicit Kiviat Diagrams

A simple 3-D, 10 sample (i.e., $n = 3$, $m = 10$) example is provided to demonstrate how the time explicit Kiviat diagrams are constructed. For simplicity, the data for this example are a

Table B1. Data Used in Appendix B

Sample Number	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$
5	0.92	0.019	311.81
6	0.96	0.01	311.3
7	0.98	0.04	313.45
8	0.98	0.02	312.46
9	0.96	0.06	313.31
10	0.94	0.01	311.82
11	0.92	0.07	313.56
12	0.88	0.13	335.93
13	0.01	0.97	408.07
14	0.14	0.81	381.81
15	0.15	0.86	383.75

Table B2. Normalized Data Used in Appendix B

Sample Number	x_1	x_2	x_3
5	0.52	−0.64	−0.66
6	0.63	−0.68	−0.68
7	0.67	−0.58	−0.62
8	0.66	−0.63	−0.65
9	0.63	−0.55	−0.62
10	0.57	−0.68	−0.66
11	0.52	−0.51	−0.62
12	0.43	−0.37	−0.01
13	−1.78	1.76	1.96
14	−1.95	1.37	1.25
15	−1.41	1.50	1.30

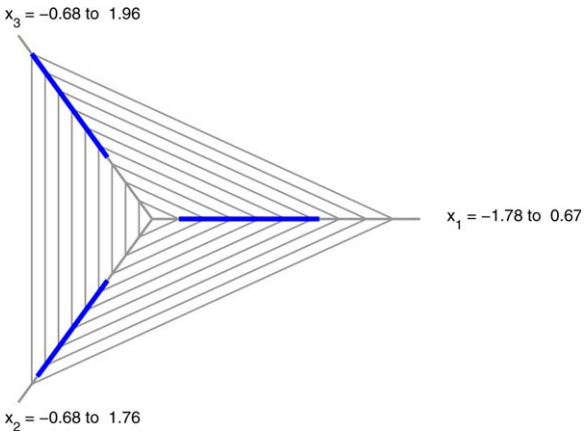


Figure B1. Ranges of variables in the Kiviat plot.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

subset of the data provided in Appendix A. Specifically, data from the first three columns and Samples 5–15 were selected to reflect changes in the process.

Step 1. Normalization of the dataset. Each variable is transformed to have zero mean and unit standard deviation. The data in Table B1 are normalized using the equation B1

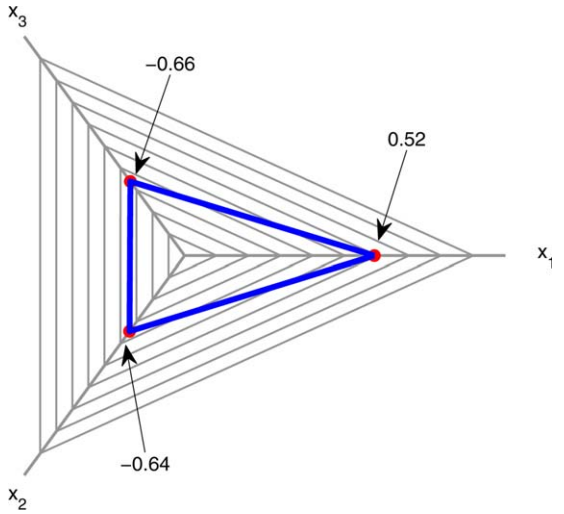


Figure B2. Polygon corresponding to data sample 1.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

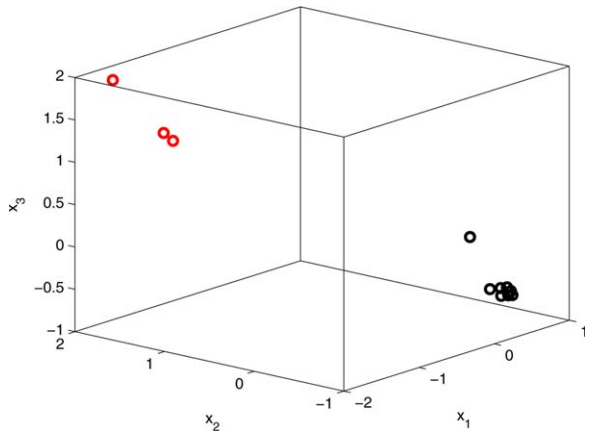


Figure C1. Sample data from Table B1, points in black correspond to normal operation, points in red correspond to faulty data.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

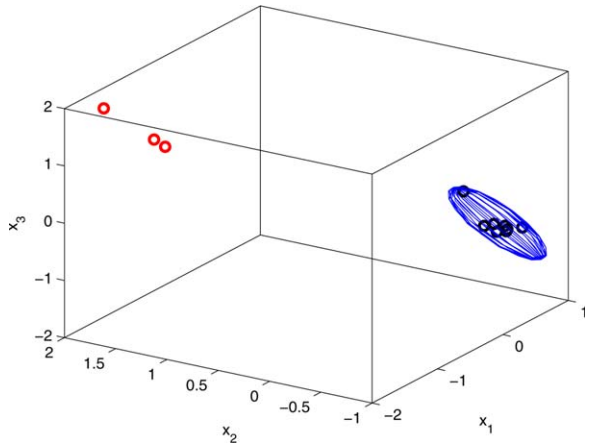


Figure C2. The cluster of points bounded by an ellipsoid.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

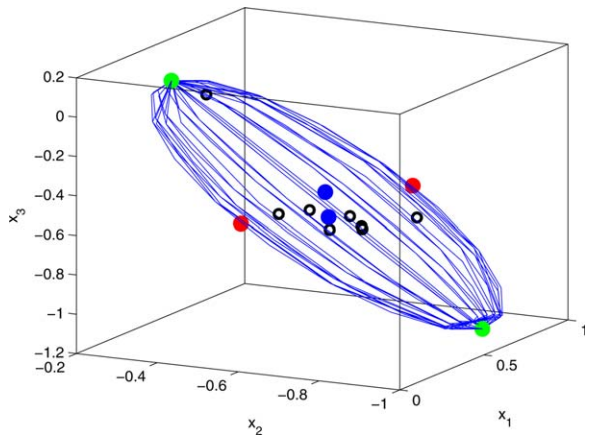


Figure C3. The extreme points of the ellipsoid are depicted by solid marked points.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

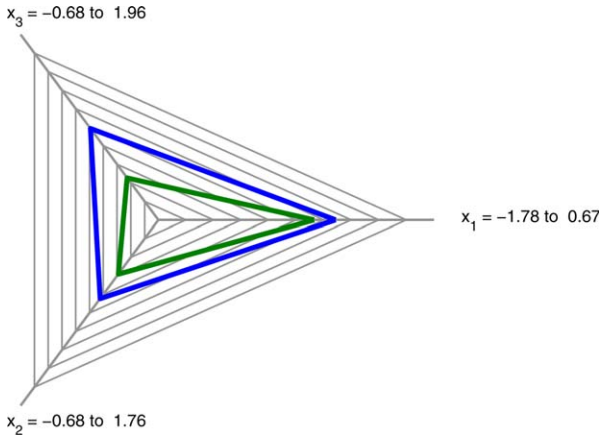


Figure C4. The extreme points of the ellipsoid are depicted in the Kiviat diagram by the blue and green polygons.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$x_{j,i}=\frac{\hat{x}_{j,i}-\bar{x}_i}{\sigma_i} \quad \forall j \in \{1,\ldots,m\}, \quad \forall i \in \{1,\ldots,n\} \tag{B1}$$

where $\hat{x}_{j,i}$ represents the "raw" data point, $\bar{x}_i$ is the mean value of variable $i \in \{1,\ldots,n\}$ across all m samples, and σ_i is the corresponding standard deviation. After normalization, the data take the values in Table B2.

Step 2. Calculate the position (in terms of radial angles) of the axes in polar coordinates, with the first axis starting at 0 radians. Axes are spaced $\frac{2\pi}{n}$ radians apart.

Step 3. Determine the overall radius of the Kiviat plots using the maximum and minimum values of each variable over the entire dataset.

The overall minimum of each variable is subtracted from each sample of the variable, and the difference is divided by the variable range (the difference between the overall maximum and minimum). Thus, the representation of each sample of each variable on the corresponding axis is obtained. A gain is used to ensure that data points stay within the bounds of the radial plot, and a bias is added to ensure that the minimum points on each axis do not "cross through" the origin owing to fluctuations in each variable. Typical values for the gain and bias are 0.8 and 0.1, respectively.[54]

$$\text{Radial Position}=\frac{\text{gain}\times(\text{data point}-\text{overall minimum})}{(\text{overall maximum}-\text{overall minimum})}+\text{bias} \tag{B2}$$

Figure B1 shows the ranges each variable on the axes of the Kiviat plot.

Step 4. Connect the points corresponding to all variables for a particular sample time to form a polygon (Figure B2).

The time dimension is captured via a time axis normal at the origin to the plane of the Kiviat plot. Once an initial polygon has been produced from the first data sample, subsequent samples can be represented as polygons whose origin is situated on the time axis at a predetermined vertical distance from the first polygon. It is convenient to assume that data are sampled with a uniform sample rate, but nonuniformly sampled data can be displayed easily.

Appendix C: Determining the Multivariate Confidence Ellipse

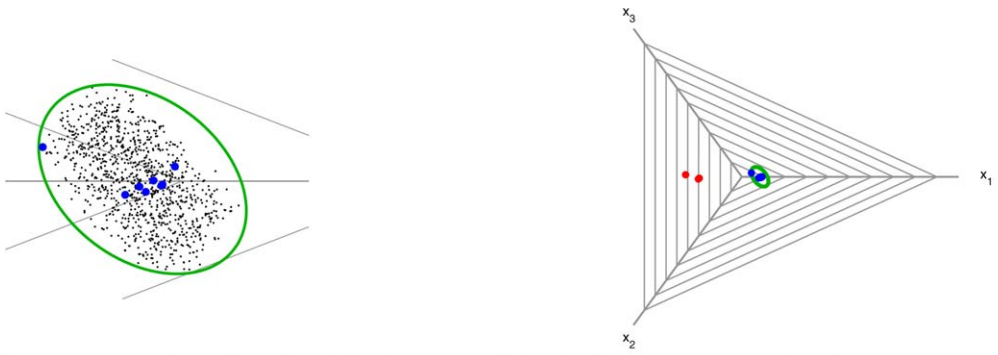
We use the plots and data from Appendix B to demonstrate the construction of a multivariate confidence ellipse. We begin by representing the normalized points in the n-dimensional Cartesian space. For the example considered here, this representation can be visualized as in Figure C1 (this is not possible in higher dimensions).

We then determine the confidence ellipsoid around the subset of data that correspond to normal operation as seen in Figure C2.

To this end, we calculate the covariance matrix of these data and use eigendecomposition (Eq. 2) to obtain the vector direction and length of the ellipsoid axes.

Then, we calculate the extremities of the ellipsoid as points in the n-dimensional space, and represent these extrema as polygons in the Kiviat diagram (Eqs. 3 and 4), obtaining the limits described in Step 3 of the Multivariate Analysis Section (Figures C3 and C4).

We now apply Step 4 from that same section, generating a series of random polygons that are bounded by the two



(a) Minimum enclosing ellipse found for the sample polygon centroids (black).

(b) Confidence ellipse used to distinguish faulty and non-faulty points

Figure C5. Minimum enclosing ellipse generation. (a) Minimum enclosing ellipse found for the sample polygon centroids (black). (b) Confidence ellipse used to distinguish faulty and nonfaulty points.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

extreme polygons and determining their centroids. The minimum enclosing ellipse that bounds these centroids is calculated by solving the optimization problem in Eq. 5. With a sufficient number of random points, we are able to generate a projection of the n-dimensional ellipsoid into the 2-D radial plot space. We use this minimum enclosing ellipse as our multivariate confidence limit for fault detection. It is seen in Figure C5b that the second cluster of data falls outside of the confidence ellipse, and will be classified as faulty data.

Manuscript received Apr. 10, 2015, and revision received Sep. 8, 2015.