Received: 6 November 2017 | Revised: 23 February 2018 | Accepted: 23 March 2018

DOI: 10.1002/asmb.2333

SPECIAL ISSUE PAPER

WILEY

Multistate multivariate statistical process control

Gabriel J. Odom[1,2] | Kathryn B. Newhart[3] | Tzahi Y. Cath[3] | Amanda S. Hering[1]

[1]Department of Statistical Science, Baylor University, Waco, TX, USA

[2]Sylvester Comprehensive Cancer Center, University of Miami, Miami, FL, USA

[3]Department of Civil and Environmental Engineering, Colorado School of Mines, Golden, CO, USA

Correspondence
Amanda S. Hering, Department of Statistical Science, Baylor University, Waco, TX 76798, USA.
Email: mandy_hering@baylor.edu

Funding information
King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR), Grant/Award Number: OSR-2015-CRG4-2582; Partnerships for Innovation: Building Innovation Capacity, National Science Foundation, Grant/Award Number: 1632227

Abstract

For high-dimensional, autocorrelated, nonlinear, and nonstationary data, adaptive-dynamic principal component analysis (AD-PCA) has been shown to do as well or better than nonlinear dimension reduction methods in flagging outliers. In some engineered systems, designed features can create a known multistate scheme among multiple autocorrelated, nonlinear, and nonstationary processes, and incorporating this additional known information into AD-PCA can further improve it. In simulations with one of three types of faults introduced, we compare accounting for the states versus ignoring them. We find that multistate AD-PCA reduces the proportion of false alarms and reduces the average time to fault detection. Conversely, we also investigate the impact of assuming multiple states when only one exists, and find that as long as the number of observations is sufficient, this misspecification is not detrimental. We then apply multistate AD-PCA to real-world data collected from a decentralized wastewater treatment system during in control and out of control conditions. Multistate AD-PCA flags a strong system fault earlier and more consistently than its single-state competitor. Furthermore, accounting for the physical switching system does not increase the number of false alarms when the process is in control and may ultimately assist with fault attribution.

KEYWORDS

multistate adaptive-dynamic principle component analysis, multivariate statistical process control, nonlinear time-varying process

1 | INTRODUCTION

In the past 2 decades, online system monitoring and statistical process control (SPC) of industrial systems have expanded rapidly due to increased personnel and energy costs and questions of environmental impact.[1] One such area of application is in the monitoring of centralized and decentralized wastewater treatment (WWT) operations. Centralized treatment plants are often placed at the geographical location that maximizes water conveyance by gravity, but most treatment systems still require extensive pumping operations.[2] Decentralized plants are growing in popularity because they reduce infrastructure costs and allow treated water to be reused closer to the point of generation.[3-5]

The nature of decentralized water and WWT facilities often makes online SPC from a central off-site location or by on-call staff the only cost-effective strategy to ensure that satellite facilities operate within permitted parameters. These systems generate data that are nonlinear, nonstationary, autocorrelated, and multivariate. In an initial monitoring study, Kazor et al[6] compared adaptive-dynamic principal component analysis (PCA), abbreviated AD-PCA, with nonlinear dimension reduction approaches. The aforementioned authors[6] found that AD-PCA yielded either similar or better monitoring results than adaptive dynamic versions of kernel PCA[7-9] and locally linear embedding.[10,11] Many other

choices for nonlinear dimension reduction could have been made (eg, independent component analysis, semidefinite embedding, and spectral multidimensional scaling); however, a PCA-based approach remains an interpretable and accessible solution that can be implemented with minimal computational cost.

Many variations of PCA have been proposed to adapt it for specific scenarios. Gertler et al[12] discussed monitoring PCA residuals, and Kano et al[13] compared moving PCA, wavelet-based multiscale PCA, and a dissimilarity scale (DISSIM). Dynamic PCA was employed by Garcia-Muñoz et al[14] to detect process faults by comparing process observations to their predictions, and Wang and He[15] augmented PCA with statistical pattern analysis.

However, none of the previous authors applied these techniques to SPC for WWT. One of the early applications of PCA to WWT process monitoring was by Wise et al,[16] and PCA was also applied to the similar task of chemical systems monitoring by Kresta et al.[17] Baggiani and Marsili-Libelli[18] discussed implementing adaptive PCA to detect faults in a large-scale centralized WWT plant by monitoring 3 process variables. Sanchez-Fernández et al[19] divided the fault detection problem of a large-scale centralized WWT plant by monitoring subsystems within the plant with distributed PCA. Kazor et al[6] were the first to focus exclusively on SPC of a decentralized WWT facility monitoring in a very complex and realistic setting with over 25 variables measured every 10 minutes.

1.1 | Motivating example

In this paper, we develop and study the concept of multistate PCA in which PCA or one of its variations is independently applied to subsets of observations. These states can occur in many processes when an operator controls a physical mechanism in the system. While one could consider any number of states, we illustrate this concept with 3 states. Figure 1 shows a schematic of a decentralized WWT facility with a hybrid sequencing batch reactor (SBR) and membrane bioreactor, known as a sequencing batch membrane bioreactor. The system is partitioned, and a blower for each bioreactor aerates prescreened wastewater. The operation of a blower in SBR 1 has no direct effect on SBR 2, and vice versa. At any given time, the system is in one of three states, ie, neither blower on (S_0), blower 1 on (S_1), or blower 2 on (S_2). The operator can control the operation and duration of each blower. Figure 2 shows the values of a few monitored variables over time, and it is clear that a blower being on or off can change the dissolved oxygen levels and blower flow rates in the bioreactor tanks dramatically. In addition, Figure 3 shows 2 heat maps where the differences in correlation matrices between states S_0 versus S_1 (*left*) and S_0 versus S_2 (*right*) are given. The pairwise correlation between variables also changes substantially between different states.

To further illustrate, assume blower 1 malfunctions. Because of this malfunction, the dissolved oxygen values in this state would fall toward 0. If state information is ignored, these lower values would not be considered unusual because the dissolved oxygen values are often 0 when both blowers are off. However, by incorporating the known state of the system, a multistate monitoring method could correctly identify that the blower is malfunctioning.

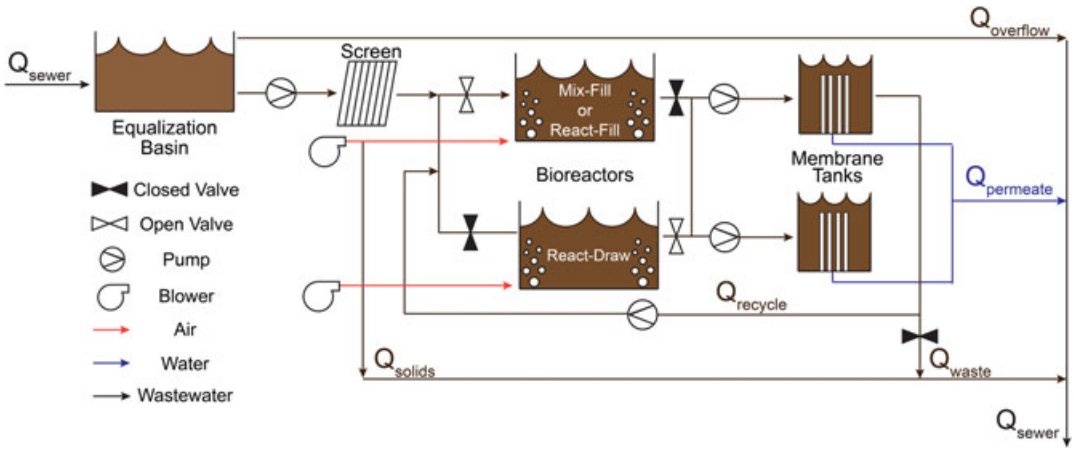

FIGURE 1 Process flow diagram of the decentralized sequencing batch membrane bioreactor wastewater treatment pilot at Mines Park. Influent from the municipal sanitary sewer is diverted to a 2500 gallon equalization tank and filtered through a 2-mm drum screen to remove large solids before addition to the bioreactors. The bioreactors are dosed sequentially. The first bioreactor is fed influent while the second recirculates through the membrane tanks. The membranes are hollow fiber, ultrafiltration membranes with a nominal pore size of 0.04 μm, and a total surface area of 74 m^2 [Colour figure can be viewed at wileyonlinelibrary.com]

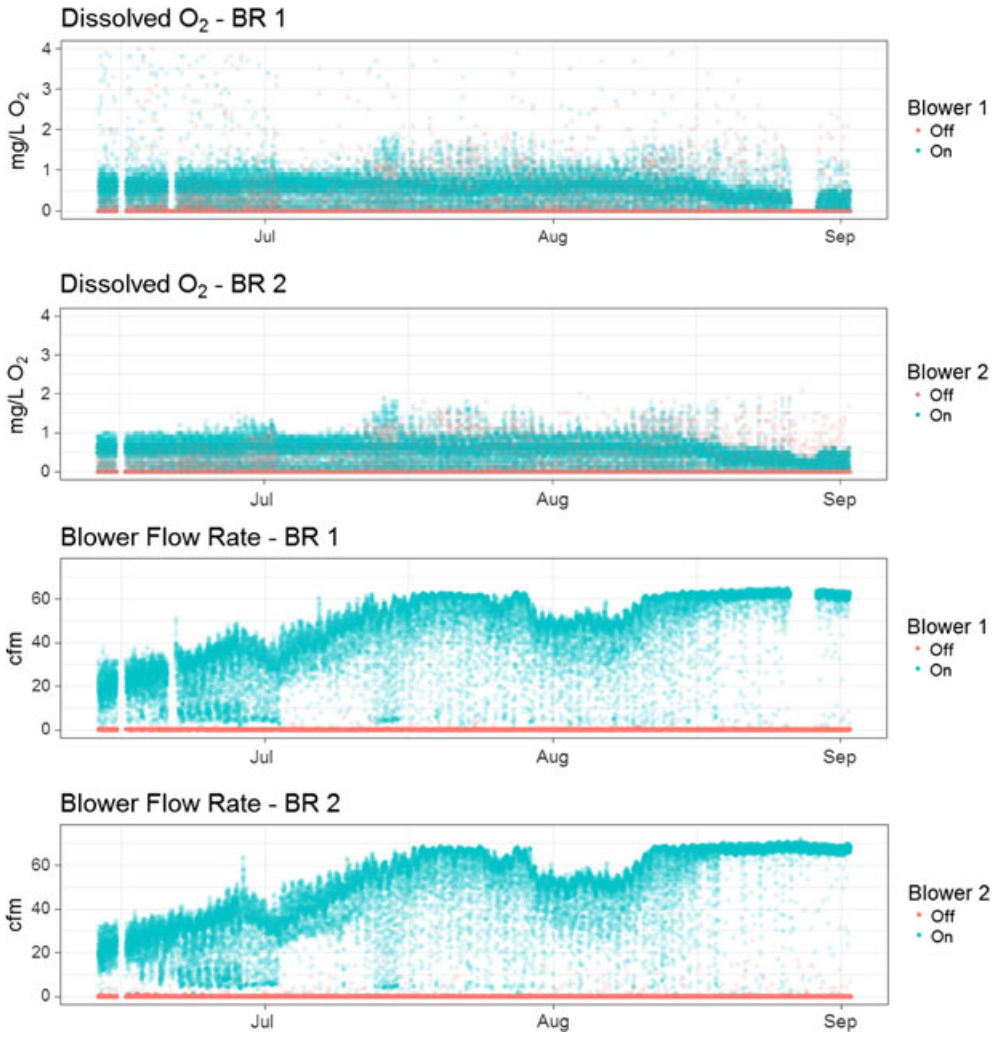

FIGURE 2 Each panel shows a time series of a monitored variable, the top two of dissolved oxygen and the bottom two of blower flow rate. Within each pair, blower 1 or 2 is either on or off, denoted by the color of each dot [Colour figure can be viewed at wileyonlinelibrary.com]

1.2 | Multistate versus multistage monitoring

A similar approach to *multistate* SPC is *multistage* SPC, which accounts for different stops along an ordered manufacturing process or assembly line. Jearkpaporn et al[20] accounted for the differences in feature means across different stages in a multivariate production process, where each stage directly influences the stages that follow it. Asadzadeh et al[21] built on this with their work on cause-selecting control charts, but these methods do not account for changes in variances of features or correlation among features across stages. For a discussion of multivariate SPC over sequential stages, see the work of Li and Tsung[22] and the references therein. The order of the stages in a multistage process matters, and stages cannot deviate from the prescribed order. However, states in multistate monitoring do not have a required order and usually depend entirely on operator input.

In this paper, we build on the work of Kazor et al[6] by constructing a modification to AD-PCA that accounts for different known and controlled process states, denoted multistate adaptive-dynamic principal components analysis (MSAD-PCA). We will show that monitoring nonlinear, nonstationary, and serially autocorrelated multistate multivariate processes by pairing the squared prediction error (*SPE*) and Hotelling's T^2 (T^2) monitoring statistics after dimension reduction via MSAD-PCA is superior to AD-PCA, especially when a fault occurs in only one state. We demonstrate the behavior of T^2 and *SPE* in a simulation study and then illustrate with a real case study on a decentralized WWT plant.

This paper is organized as follows. In Section 2, the basic framework for MSAD-PCA is presented. In Section 3, a simulation study to test the utility and reliability of the new method is described. Section 4 presents the simulation results,

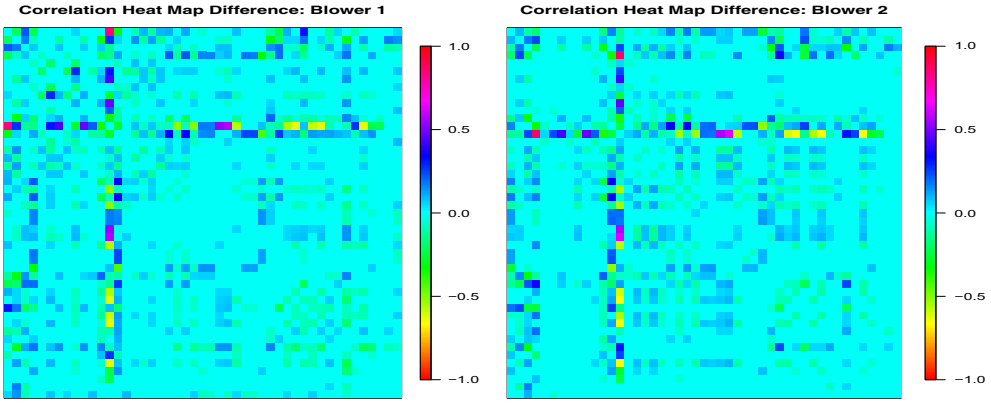

FIGURE 3 Differences in correlation matrices of variables being monitored between $\mathcal{S}_0$ and $\mathcal{S}_1$ (*left*) and $\mathcal{S}_0$ and $\mathcal{S}_2$ (*right*) [Colour figure can be viewed at wileyonlinelibrary.com]

and in Section 5, we return to the WWT problem to demonstrate the benefit of monitoring the process with MSAD-PCA. In Section 6, we offer concluding remarks and discuss future research directions.

2 | METHODS

Let $\mathbf{X}_s = [X_1(s), X_2(s), \ldots, X_p(s)] \in \mathbb{R}_{1\times p}$ be a standardized realization from a multivariate stochastic process indexed by $s = 1, 2, \ldots, n$. When the process dimension p is large, variables are often noisy and dependent. One widely accepted technique to reduce the data dimension and produce linearly independent variables is PCA.[23] Briefly, PCA is an orthogonal linear transformation mapping the existing data matrix $\mathbf{X}$, defined as

$$\mathbf{X} := [\mathbf{X}_1 \ \vdots \ \mathbf{X}_2 \ \vdots \ \cdots \ \vdots \ \mathbf{X}_n]^T \in \mathbb{R}_{n\times p}, \tag{1}$$

onto a new coordinate system wherein the greatest source of the variability of $\mathbf{X}$ lies in the first coordinate of the new system, the second greatest source lies in the second coordinate, and so forth. To reduce the data dimension from p to $d < p$ dimensions, $\mathbf{X}$ is multiplied by the d eigenvectors of the correlation matrix of $\mathbf{X}$ whose corresponding eigenvectors are the largest, denoted by $\mathbf{P}_d$. Then, the new $n \times d$ matrix is $\mathbf{Y}_d = \mathbf{X}\mathbf{P}_d$.

2.1 | Adaptive-dynamic PCA

Ge and Song[24] gave a good survey of accounting for nonlinearity, nonstationarity, or autocorrelation in process data, but Kazor et al[6] found that an adaptive-dynamic variation of PCA adequately accounted for these features in simulated data. Both the adaptive and dynamic variations are described here.

The adaptive modification to PCA creates a rolling training window over which to estimate a reduced data matrix $\mathbf{Y}_d$ to mitigate the effects of nonlinearity and nonstationarity. We estimate the $p\times d$ projection matrix $\mathbf{P}_d$ by taking the SVD of w observations, $[\mathbf{X}_{s-w+1} \ \vdots \ \mathbf{X}_{s-w+2} \ \vdots \ \cdots \ \vdots \ \mathbf{X}_{s-1} \ \vdots \ \mathbf{X}_s]^T$, rather than of the full data matrix $\mathbf{X}$. Then, the next n_u observations are projected from p to d dimensions by right-multiplying each by $\mathbf{P}_d$, yielding $1\times d$ observations $\mathbf{Y}_{s+1}, \mathbf{Y}_{s+2}, \ldots, \mathbf{Y}_{s+n_u}$, where $n_u << w$ is the number of observations to monitor before moving the window forward and re-estimating $\mathbf{P}_d$. After the next set of observations, $\mathbf{X}_{s+n_u+1}, \ldots, \mathbf{X}_{s+2n_u}$, is recorded, the oldest n_u observations, $\mathbf{X}_{s-w+1}, \mathbf{X}_{s-w+2}, \ldots, \mathbf{X}_{s-w+n_u}$, are removed from the training set; the newest n_u observations, $\mathbf{X}_{s+1}, \mathbf{X}_{s+2}, \ldots, \mathbf{X}_{s+n_u}$, are added to the training set; and the aforementioned dimension reduction process is repeated. At each step in the adaptive process, the number of training observations is fixed at w.

As an example, consider a multivariate process that is recorded hourly. For short subsets of consecutive observations, $\mathbf{X}$ may be locally linear. Thus, we could estimate the $p \times d$ projection matrix $\mathbf{P}_d$ using a week's worth of data points, setting $w = 24 \times 7 = 168$. From this first week of observations, we reduce the data dimension for the eighth day of the process, and we perform tests on the set of 24 reduced observations $\{\mathbf{Y}_{169}, \mathbf{Y}_{170}, \ldots, \mathbf{Y}_{192}\}$. Then, the oldest 24 observations are removed from $\mathbf{X}$ and are replaced with the most recent 24, and the process is repeated until the end of the data is reached.

A dynamic modification to PCA can account for autocorrelation in the process. Assume that a feature $X_i(s)$ depends upon $X_i(s-\ell)$, for some lag $\ell \in \mathbb{N}$. These lagged features are column concatenated onto the data matrix $\mathbf{X}$. In the most general case, all lags up to lag ℓ for each feature could be included, and the new data matrix is of size $(n-\ell)\times(p+p\ell)$. To curb the rapid dimensional growth possible in $\mathbf{X}$, Rato and Reis[25] advocated including only a few lagged features per original feature. To this end, Kazor et al[6] defined

$$\mathrm{L}^{\boldsymbol{\ell}}(\mathbf{X}_s) := \left[X_1(s), X_1(s-\ell_1), X_2(s), X_2(s-\ell_2), \ \dots, X_p(s), X_p(s-\ell_p)\right], \tag{2}$$

where $\boldsymbol{\ell} = [\ell_1, \ell_2, \ \dots, \ell_p]$ is the vector of maximally autocorrelated lags for each feature $i = 1, \ \dots, p$. This lag feature selection criterion yields a data matrix $\mathbf{X}$ with $n - \max\{\boldsymbol{\ell}\}$ rows and $2p$ columns, which doubles the number of features. Thus, the projection matrix $\mathbf{P}_d$ has dimension $2p \times d$, but these new features are maximally correlated with the existing features, so the number of retained principle components, d, is not expected to double.

2.2 | Multistate AD-PCA

We now consider how known dichotomous states affect system behavior. Multistate AD-PCA accounts for the multistate structure of the process while still incorporating the adaptive and dynamic PCA modification. When the relationships among features change as the state changes, the projection matrix will have different linear combinations of features under different states.

Again, let $\mathbf{X}_s$ be defined as $[X_1(s), X_2(s), \ \dots, X_p(s)]$. Given a set, $\mathcal{S}$, of K mutually exclusive process states, $\mathcal{S}_1, \ \dots, \mathcal{S}_K$, we assign corresponding pairwise disjoint sets of indices $\mathcal{T}_1, \ \dots, \mathcal{T}_K$ such that $\bigcup_{k=1}^{K} \mathcal{T}_k = \{1, \ \dots, n\}$ and $n_k = C(\mathcal{T}_k)$, where $C(\cdot)$ denotes the cardinality of a set. Specifically, $s^* \in \mathcal{T}_k \iff \mathbf{X}_{s^*}$ was observed under state $\mathcal{S}_k$.

In order to find the feature subspace maximizing the principal component loadings, the observations from each process state should be treated differently. Therefore, for states $\mathcal{S}_1, \ \dots, \mathcal{S}_K$, we partition $\mathbf{X}$ as $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ \dots, \mathbf{X}^{(K)}\}$, where $\mathbf{X}^{(k)} := [\mathbf{X}_{\min(\mathcal{T}_k)}, \ \dots, \mathbf{X}_{\max(\mathcal{T}_k)}]^T$ is the set of all observations from the process under state k, ordered in s. Appropriate lags can then be included and state-specific projection matrices calculated along a rolling window for a full MSAD-PCA approach.

In summary, MSAD-PCA splits the observations by state and estimates state-specific projection matrices that update over a rolling window of training observations. However, one should exercise caution when including state information. For K equally sized states, MSAD-PCA will partition the observations into K groups, effectively dividing the total sample size by K. Therefore, the number of states should be kept as small as possible. We return to this point at the end of Section 6.

2.3 | Process monitoring statistics and thresholds

Given the $2p \times d$ linear projection matrix $\mathbf{P}_d$, trained from the observations indexed by $\{\max_i\{\boldsymbol{\ell}\}, \ \dots, s\}$, 2 process monitoring statistics of the new observation, $\mathbf{X}_{s+1}$, are calculated, namely, Hotelling's T^2 and squared prediction error (*SPE*). The T^2 statistic is calculated as $T^2_{s+1} = \mathbf{Y}_{s+1}\Lambda_d^{-1}\mathbf{Y}_{s+1}^T$, where $\boldsymbol{\Lambda}_d = \mathrm{diag}(\lambda_1, \lambda_2, \ \dots, \lambda_d)$ and λ_i is the ith eigenvalue of $\mathbf{P}$. Thus, T^2 is the Mahalanobis distance of the mapped value $\mathbf{Y}_{s+1}$ from the original space into the PCA subspace, and it measures deviations in the lower d-dimensional subspace. Because this subspace should characterize the major components of the process under in control (IC) conditions, outliers identified by large T^2 values are often indicative of out of control (OoC) conditions.

The *SPE* statistic is calculated as $\mathrm{SPE}_{s+1} = (\mathbf{X}_{s+1} - \mathbf{Y}_{s+1}\mathbf{P}_d^T)(\mathbf{X}_{s+1} - \mathbf{Y}_{s+1}\mathbf{P}_d^T)^T$, which computes the squared distance between the original and reduced-dimension vectors. It measures the goodness-of-fit of the d-dimensional model to the p-dimensional process observations. Abnormal values of *SPE* can indicate either an unusual observation or that the lower-dimensional model does not account for an important component of process variability.

Typically, parametric distributions, such as the χ^2 or F-distribution, are used to compute thresholds for these process monitoring statistics, but when the underlying distribution of the data is nonnormal, these thresholds are not reliable.[26] To avoid making any distributional assumptions about T^2, we follow Kazor et al[6] who calculate nonparametric thresholds based on observations in the IC training window for the *SPE* and T^2 monitoring statistics via kernel density estimation with a Sheather-Jones bandwidth[27] and a Gaussian kernel. A threshold for *SPE* is computed similarly. For further discussion of nonparametric SPC, see the works of Qiu and Li[28] and Chen et al.[29]

3 | SIMULATION DESIGN

The fault-detection behavior of MSAD-PCA compared with ordinary AD-PCA is studied in simulation. Performance is measured with (a) the average false alarm rates for IC conditions, (b) the proportion of faults detected, and (c) the average time to detection for true system faults. Lower values of (a) and (c) and higher values of (b) are desirable. Here, we simulate a 3-variable dataset that contains nonlinear relationships among features and autocorrelation and nonstationarity in each feature, as introduced by Kazor et al[6] who extended the study by Dong and McAvoy.[30] We simulate both a single-state process and a new multistate process with 3 states. Specific details needed to replicate the simulated data are given in Supplementary Material with code and documentation for complete replicability in the vignettes of the `mvMonitoring` software package for `R`.

3.1 | Fault scenarios

We generate data of length $60 \times 24 \times 7 = 10\,080$, yielding 1 observation per minute for 1 week, and introduce a fault at $s = 8500$, or roughly 85% through the cycle. Table 1 shows the types of faults and how these faults affect states and features. The columns show the general type of fault introduced, while the rows show how these faults affect each process feature. Specific details on the construction of the faults is given in Supplementary Material.

In summary, the "1" faults are a shift to all features in all states ("A"), 1 feature in all states ("B"), or 2 features in 1 state ("C"). The "2" faults are a slow drift to all features in all states ("A"), 2 features in all states ("B"), or 1 feature in 1 state ("C"). Fault 3A is a mutation of an underlying latent variable, affecting all features in all states. Fault 3B is a different mutation of the latent variable, but it affects only one feature in all states. Fault 3C is a shift and amplitude increase to the error vector of 1 feature in 1 state. The "C" faults should showcase differences between AD-PCA and MSAD-PCA.

The top row of Figure 4 shows about one day's worth of observations of the time series of a single random draw of the IC multistate multivariate process data generated via the simulation scheme. Clearly, the means and variances for each of the process features change as the state changes. The vertical black line at 21:40 on 2 December marks when a fault will be introduced (as shown in the bottom row). Thus, in this top row of plots, the values to the right of the black line show the observations *as they should have been*, if a fault had not been introduced.

The bottom row of graphs in Figure 4 shows a realization of state-specific Fault 1C. Features x and z are affected in S_3 only. For feature x, the mean increases, but it still lies within the range of values of x observed across all 3 states. The mean of feature z in state S_3 begins to increase over time, so it acts like a slow drift. A monitoring scheme that ignores the states should perform worse for this type of fault. Realizations of all other faults can be seen in Supplementary Material

Given the data generation described in Supplementary Material, the simulation proceeds as follows.

1. Draw a set of IC observations with associated fault introduced under S_1 (single state) or under S_1, S_2, S_3 alternating (multistate).
2. Train the fault-detection system on 75% of the observations (7560 observations for AD-PCA). For MSAD-PCA, note that the number of training observations per class is $7560/3 = 2520$.
3. Apply AD-PCA and MSAD-PCA to the training and test data sets for each data set under both the single-state scenario and multistate scenario.

TABLE 1 Fault types by how each fault affects different sets of features

	Type of Fault		
Features Affected	**Shift Fault**	**Drift Fault**	**Latent/Error Faults**
All features equally	Fault 1A (x, y, z)	Fault 2A (x, y, z)	Fault 3A (x, y, z)
Each feature differently	Fault 1B (x)	Fault 2B (y, z)	Fault 3B (z)
Each state differently	Fault 1C $(x,\ z) \in S_3$	Fault 2C $y \in S_2$	Fault 3C $y \in S_2$

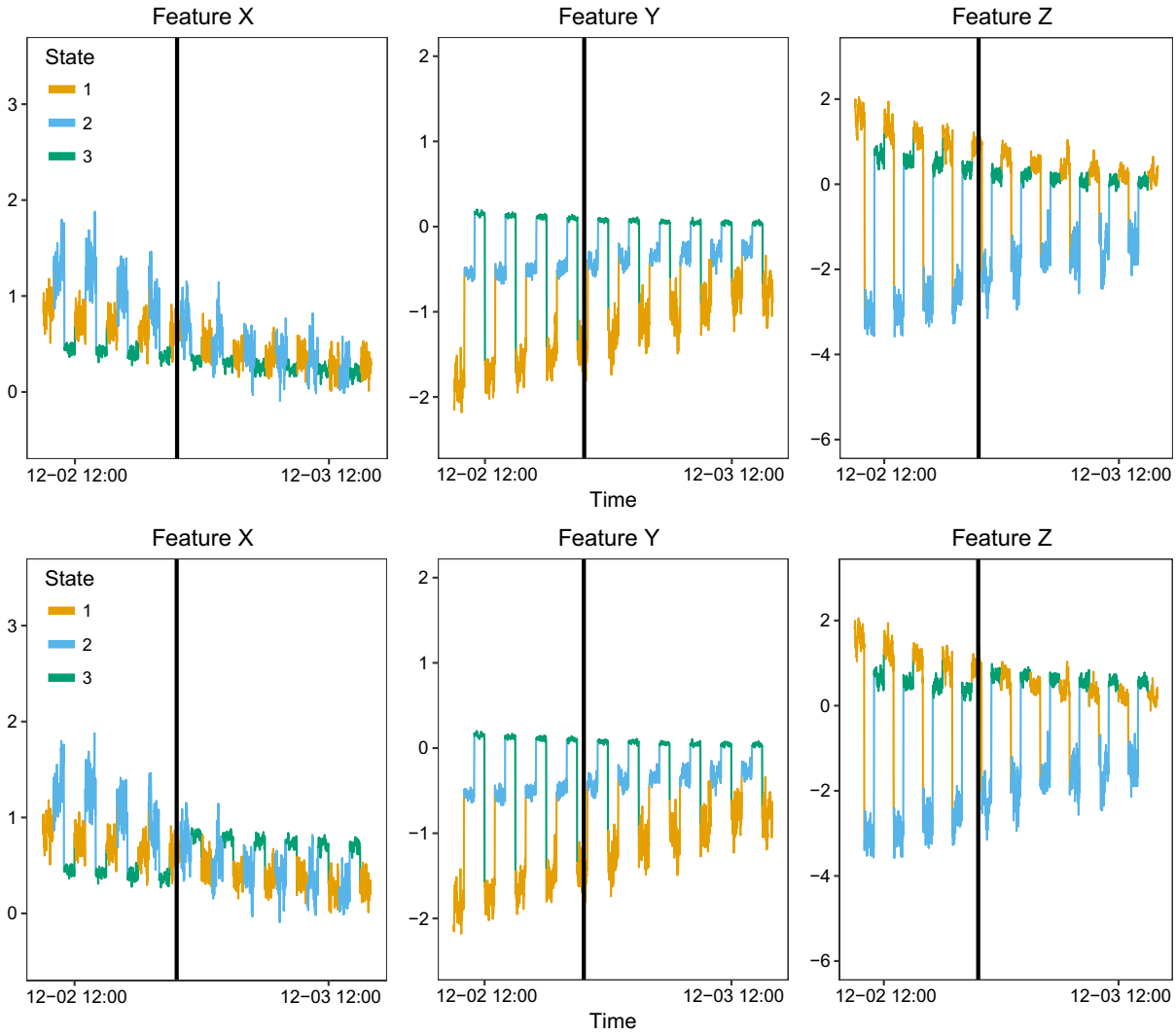

FIGURE 4 Process time series comparison of in control (IC) and out of control conditions for Fault 1C, showing a 31-hour subset immediately before and after the fault is introduced. The dates on the horizontal axis of each plot are arbitrary. *Top row:* Multivariate process feature time series under IC conditions. The vertical black line marks the time at which the fault is induced. *Bottom row:* Multivariate process feature time series before and after fault 1C is introduced [Colour figure can be viewed at wileyonlinelibrary.com]

4. Measure the following.
   (a) False alarm rate: When IC, we expect to see 0 alarms, so the *SPE* and T^2 statistic alarm rates on the IC data are their false alarm rates.
   (b) Fault detection: For *SPE* and T^2, this is a 1 if the monitoring statistic detected a fault, and 0 otherwise.
   (c) Detection time: For *SPE* and T^2, this is the time in minutes after a fault is introduced until a statistic raises an alarm. Note that the earliest the methods can detect a fault is within 3 minutes because we set the number of sequential faults necessary to trigger an alarm to be equal to 3.
5. Repeat Steps 1 through 4 1000 times.

4 | SIMULATION RESULTS

In this section, we present and discuss the results for each of the 4 monitoring statistics (MSAD-PCA and AD-PCA, each paired with *SPE* and T^2). In Table 2, we present the false alarm rates, expected number of false alarms per day, and detection percentages for each of the 4 monitoring statistics under the multistate or single-state data generation scenarios.

TABLE 2 False alarm rates and detection probabilities. Note, Faults 1C, 2C, and 3C are state-specific and thus cannot be applied to observations generated under a single state, so we put an NA in the table for these scenarios

		Multistate Scenario				Single-State Scenario			
		MSAD-PCA		AD-PCA		MSAD-PCA		AD-PCA	
		SPE	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	T^2
False alarms	%	0.15	0	0	1.25	0.10	11.5	0.12	14.5
	/Day	2.3	0.1	0.1	18.1	1.6	165.7	1.9	208.9
Shift faults	1A	100.0	100.0	1.0	99.8	100.0	100.0	99.3	100.0
	1B	100.0	100.0	0.0	97.7	100.0	100.0	100.0	100.0
	1C	100.0	32.0	0.0	96.5	NA	NA	NA	NA
Drift faults	2A	100.0	100.0	0.0	97.7	25.8	100.0	20.3	100.0
	2B	100.0	100.0	0.0	7.7	100.0	100.0	100.0	100.0
	2C	100.0	100.0	0.0	97.0	NA	NA	NA	NA
Latent/error faults	3A	100.0	16.2	3.0	30.6	92.9	98.5	93.0	98.8
	3B	100.0	99.7	98.3	99.0	100.0	100.0	100.0	100.0
	3C	100.0	14.0	0.0	97.0	NA	NA	NA	NA

Abbreviations: AD-PCA, adaptive-dynamic principal component analysis; MSAD-PCA, multistate adaptive-dynamic principal components analysis.

TABLE 3 Average and 95th percentile (labeled "Upper Tail") detection times for shift, drift, and latent/error faults. Cells shaded in gray have perfect fault detection based on results in Table 2

				Shift Faults			Drift Faults			Latent / Error Faults		
				1A	1B	1C	2A	2B	2C	3A	3B	3C
Multistate Scenario	MSAD-PCA	*SPE*	Average	3.0	3.0	171.1	369.9	281.7	699.3	869.4	16.7	34.1
			Upper Tail	3.0	3.0	1032.0	621.1	384.0	953.0	1225.0	54.0	191.0
		T^2	Average	3.0	97.0	392.2	492.6	570.6	935.8	1351.0	679.0	1251.0
			Upper Tail	3.0	233.0	1354.0	533.0	692.0	1084.0	1518.0	999.6	1432.0
	AD-PCA	*SPE*	Average	18.0	∞	∞	∞	∞	∞	1372.0	738.2	∞
			Upper Tail	18.0	∞	∞	∞	∞	∞	1518.0	1116.0	∞
		T^2	Average	5.2	26.0	122.0	1114.0	191.7	81.8	660.9	87.5	9.7
			Upper Tail	5.0	26.0	490.0	1406.0	1514.0	195.0	1459.0	258.0	23.0
Single-state Scenario	MSAD-PCA	*SPE*	Average	9.9	3.0	NA	1171.0	304.7	NA	1060.0	17.5	NA
			Upper Tail	32.0	3.0	NA	1568.0	401.8	NA	1481.0	48.0	NA
		T^2	Average	3.0	4.1	NA	85.8	93.4	NA	435.4	95.2	NA
			Upper Tail	3.0	5.0	NA	156.5	173.0	NA	942.3	220.8	NA
	AD-PCA	*SPE*	Average	12.4	3.0	NA	1141.0	312.3	NA	1055.0	17.9	NA
			Upper Tail	46.0	3.0	NA	1562.0	405.8	NA	1476.0	50.8	NA
		T^2	Average	3.0	4.1	NA	83.2	90.1	NA	412.5	91.2	NA
			Upper Tail	3.0	5.0	NA	152.8	168.0	NA	929.9	222.8	NA

Abbreviations: AD-PCA, adaptive-dynamic principal component analysis; MSAD-PCA, multistate adaptive-dynamic principal components analysis.

In Table 3, the rows show the mean and the 95th percentile ("Upper Tail") of the time until fault detection (in minutes) for each of the 4 monitoring statistics within the multistate or single-state data generation scenarios.

4.1 | False alarm rates and detection probabilities

In Table 2, we present the false alarm rates and detection probabilities under both the single-state scenario and multistate scenario for each of the monitoring statistics. These false alarm rates were calculated by applying the 4 process monitoring statistics to IC data for a full week (10 080 observations) with 3 days for training and 4 days for testing with the model updating every 24 hours. We set $\alpha = 0.001$, which is a prespecified significance level controlling the critical value needed to "flag" an observation as a potential system fault. We found that the median detection percentage for all 4 monitoring statistics across all faults is 100%.

For the multistate scenario, AD-PCA *SPE* yielded very few false alarms on average, but this is not unexpected as this statistic is not very sensitive. In fact, the only fault that AD-PCA *SPE* detects with high probability (98.3%) is Fault 3B,

but it fails completely for the other 8 faults. In contrast, MSAD-PCA T^2 also maintained a false alarm rate of effectively 0 but was still sensitive enough to detect most faults. Faults 1C, 3A, and 3C were the most difficult for the MSAD-PCA T^2 statistic to detect, detecting only 32%, 16.2%, and 14% of these faults, respectively. In comparison, AD-PCA T^2 is overly sensitive and issues over 18 false alarms per day on average. This seriously degrades any positive consideration we have for this statistic in terms of time to fault detection. This statistic also has difficulty detecting Faults 2B (7.7% detection) and 3A (30.6% detection). In the "sweet spot," MSAD-PCA *SPE* recorded 2.3 false alarms per day on average, while also detecting all 9 faults in 100% of the simulation replicates. Focusing on the state-specific faults (1C, 2C, and 3C), we see that they are more difficult to detect without the multistate adaptation; AD-PCA *SPE* always fails in these cases. The AD-PCA T^2 statistic does detect faults well but at the cost of a much higher false alarm rate.

For the single-state scenario, Table 2 shows that the false alarm rates for the T^2 statistic under both methods increase. The AD-PCA *SPE* statistic false alarm rates increase slightly for observations generated from a single-state process as opposed to a multistate process, while false alarm rates slightly decrease for the MSAD-PCA *SPE* statistic. Comparatively, these changes are very small relative to the changes in false alarm rates for the T^2 statistic. The increased false alarm rates for MSAD-PCA T^2 for observations from a single-state process illustrate a possible consequence of making a model overly complex by incorrectly assuming a multistate model. Therefore, thorough exploratory data analysis is recommended to justify the necessity of a multistate modification to an existing fault-detection system.

4.2 | Time to detection

Table 3 shows the average and 95th percentile across the 1000 simulated datasets of their time to detection, and the gray shaded boxes correspond to those instances in Table 2 for which the corresponding statistic identified 100% of the faults. Fault 1A for the multistate scenario is detected by three of the methods almost instantly, except for AD-PCA *SPE*. The AD-PCA *SPE* monitoring statistic does not detect a fault until minute 18 on average. Fault 1B is similar but slightly more difficult to detect, as expected. Moreover, as expected, the state-specific Fault 1C is the most difficult of the shift faults to detect; comparing AD-PCA T^2 with MSAD-PCA *SPE*, the average time to fault detection is 122 minutes (AD-PCA T^2) versus 171 minutes (MSAD-PCA *SPE*). Furthermore, the 95th percentile of MSAD-PCA *SPE* is more than double than that of AD-PCA T^2. The reason for this is in the detection probability. As shown in Table 2, MSAD-PCA *SPE* detects 100% of these faults (increasing the time to fault detection in the upper tail), whereas AD-PCA T^2 detects Fault 1C in only 96.5% of the replicates.

For the multistate drift faults (Faults 2A, 2B, and 2C), our expectation that the drift fault would take a few hours to detect seems to hold true. Interestingly, MSAD-PCA *SPE* detected Fault 2B faster than Fault 2A, whereas MSAD-PCA T^2 detected Fault 2A faster than Fault 2B. In addition, AD-PCA T^2 detected Fault 2B on average hours before either MSAD-PCA *SPE* or T^2, but Table 2 shows it only detected Fault 2B in 7.7% of replicates. Fault 2C began in S_2, so the earliest a method could *correctly* detect a fault would be 21 minutes after the fault start index. As expected, Fault 2C was the most difficult drift fault to detect for MSAD-PCA *SPE* and MSAD-PCA T^2 but was uncharacteristically easy to detect by AD-PCA T^2 in comparison. This may be due to the fact that AD-PCA T^2 is overly sensitive, which is shown by the high false alarm rate in Table 2 (over 18 per day) for this statistic.

Of the 2 latent feature faults (3A and 3B) and the error fault (3C), Fault 3A was the most difficult fault for all 4 methods to detect, as shown by the low detection proportions for MSAD-PCA T^2 and AD-PCA *SPE* and T^2. Fault 3B was markedly easier to detect by comparison, and MSAD-PCA *SPE* detected this fault quickly and consistently. Fault 3B is also the only fault under the multistate scenario in which AD-PCA *SPE* and T^2 perform comparably to the other monitoring statistics. However, Fault 3C (a perturbation of the error structure in S_2 only) shows AD-PCA *SPE* is again ineffective. This fault is especially difficult even for MSAD-PCA T^2 to detect, detecting a fault in only 14% of cases. However, MSAD-PCA *SPE* and AD-PCA T^2 detect this fault relatively quickly.

The fault detection times for single-state scenarios are also included in Table 3 to show that the multistate methods detect faults without substantial deterioration in average time to detection. Thus, as long as each state has sufficient observations, MSAD-PCA does not suffer, even when the process is erroneously assumed to have the multistate property.

Overall, the best performing monitoring statistics by average time to fault detection are MSAD-PCA *SPE* and AD-PCA T^2. The best performing monitoring statistics by detection percent are MSAD-PCA *SPE*, MSAD-PCA T^2, and AD-PCA T^2. The best performing monitoring statistics by false alarm rates are MSAD-PCA *SPE*, MSAD-PCA T^2, and AD-PCA *SPE*. In summary, the MSAD-PCA *SPE* monitoring statistic presents the best combination of high detection probability, short time to detection, and low false alarm rates, especially for state-specific faults. While the AD-PCA T^2 monitoring statistic

does perform very well in some scenarios, it also performs very poorly in others. Because of this and its high false alarm rate, this statistic is not recommended for monitoring multistate processes.

5 | CASE STUDY

We apply these methods to data collected at Mines Park at Colorado School of Mines, a demonstration-scale decentralized WWT facility, diagrammed in Figure 1. Data were recorded at 3 different temporal frequencies: 13 features at the 5-second level, 19 at the 1-minute level (two of which indicate the operation of each blower), and 17 at the 10-minute level. We are interested in detecting process faults remotely; that is, can the MSAD-PCA model be applied to automatically detect when the WWT process is OoC? We hypothesize that the underlying multivariate process may change based on the operation of a set of 2 blowers that are used to regulate dissolved oxygen as described in Section 1.1.

5.1 | Specific system considerations

When either of the blowers is on, some process features change dramatically. The blowers have independent controls, so we have 4 states: S_0 = both blowers off, S_1 = only blower 1 on, S_2 = only blower 2 on, and S_B = both blowers on, as shown in Figure 5. We observe 130 153 sequential 1-minute level IC observations over an 81-day period at this facility from June 1 to September 1, 2013, which was described in the work of Kazor et al.[6] The operation of the blowers is known and controlled by the engineers, so we investigate the lengths of stay in each blower state.

When both blowers are off (S_0), $n_0 = 54\,408$. This is the most common state. The process changes from S_0 to S_1 or S_2 with similar frequency but never changes directly to S_B. As we can see from the top 3 densities of Figure 5 (right), the modal lengths of time to remain in S_0, S_1, and S_2 are 10 or 20 minutes. If neither blower is on, it appears that both blowers will remain off for either 10 or 20 minutes, and then one of the blowers will turn on. S_1 (only blower 1 on) and S_2 (only blower 2 on) also occur frequently ($n_1 = 35\,974, n_2 = 37\,701$). The last state, S_B (both blowers on), is comparatively rare, with $n_B = 2070$ observations recorded, which is less than 2% of the observations. Thus, accounting for this rare state is unlikely to increase model accuracy. We randomly assign the observations in S_B to either S_1 or S_2 with equal probability. The blower operation indicator is recorded at the 1-minute level, and the effects of the blower are almost immediate for features affected by blower status, so we downscale all of the 10-minute observations to the 1-minute level as described in detail in Supplementary Material, and we upscale the 5-second observations to 1-minute averages.

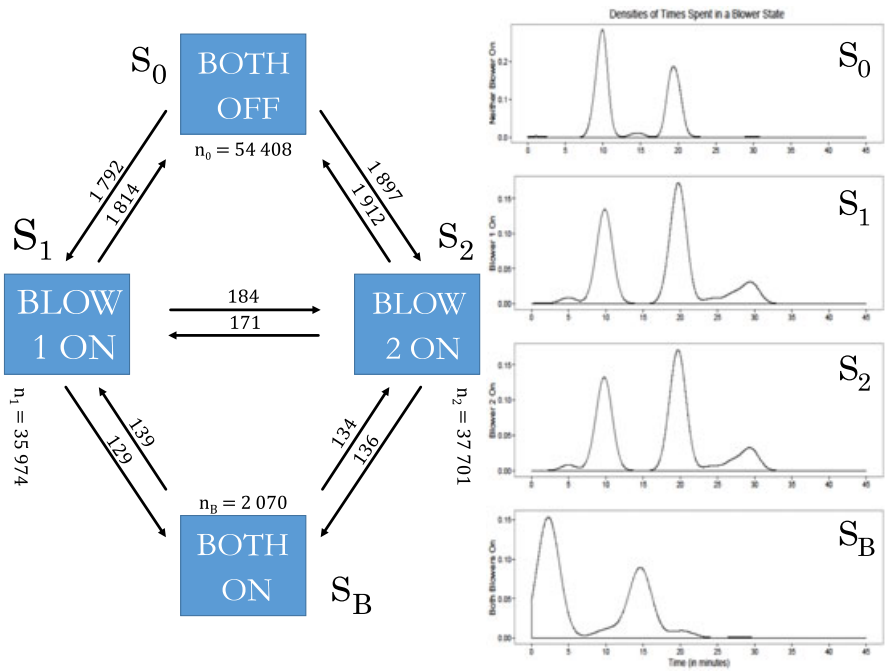

FIGURE 5 The blower process flowchart shows the number of times the process changes states and in which directions (left). The lengths of times spent in each state are bimodal, as shown by their densities (right) [Colour figure can be viewed at wileyonlinelibrary.com]

With all observations on the 1-minute scale, we can see some of the differences among the variable values across states in Figure 2 and in S_1 and S_2 versus S_0 in the 47×47 correlation heatmaps in Figure 3. As discussed in Section 1.1, both the feature values themselves and the relationships between the features change based upon the multivariate process state. Changing from S_0 (neither blower on) to S_1 (blower 1 on) has a substantial effect on the recorded process values for SBR 1 (row 13 of Figure 3, *left*), and for other features, such as raw, influent, and waste flow values. We see a similar effect on SBR 2 (row 14 of Figure 3, *right*) when the state changes from S_0 to S_2. Therefore, the PCA decomposition will change from one state to another. Accounting for these process changes should improve our ability to correctly detect a system fault.

5.2 | False alarm rates

We first apply our method to the IC dataset, monitoring the false alarm rates using the first 40 days of the 81-day period as training data and updating the model daily. Kazor et al[6] monitored the process using 10-minute averages, but for the 1-minute values, the AD-PCA false alarm rates were 0.3% for both *SPE* and T^2. The MSAD-PCA rates were 0.3% for *SPE* and 0.1% for T^2. The slight decrease in the false alarm rate of the T^2 monitoring statistic agrees with the simulation results, showing that the MSAD-PCA T^2 has higher specificity than the single-state versions.

However, the false alarm rates for both AD-PCA and MSAD-PCA are higher than the specified α threshold of 0.1%. These higher false alarm rates are likely due to observations being serially autocorrelated. In other words, if 1 observation is flagged as beyond the threshold for its *SPE* or T^2 monitoring statistics, the monitoring statistics for the next 2 to 5 observations also have a higher probability of being beyond these same thresholds. We expect the impact of autocorrelation to be greater on 1-minute averaged observations than it will be on 10-minute averaged observations. Because of this, we increase the number of sequential flags necessary to trigger an alarm from 3 to 5. Of note, in the Kazor et al[6] paper, 3 flagged 10-minute observations in a row correspond to 30 minutes of data. Here, 5 consecutive flags correspond to 5 minutes.

5.3 | Time to fault detection

We also applied our method to a fault analyzed by Kazor et al.[6] We train on the observations from April 10 at 01:10 to April 19 at 00:00, for a total of 12 854 observations recorded at the 1-minute level. Operators detected a fault at 10:00 on April 24, 2010 when pH dropped below an important biological threshold. The AD-PCA method first detects a fault at 16:31 on April 21 and triggers a few alarms between then and the point at which the operators detected the fault. This is shown in Figure 6 (*top*). AD-PCA never triggers *both* the *SPE* and T^2 alarms.

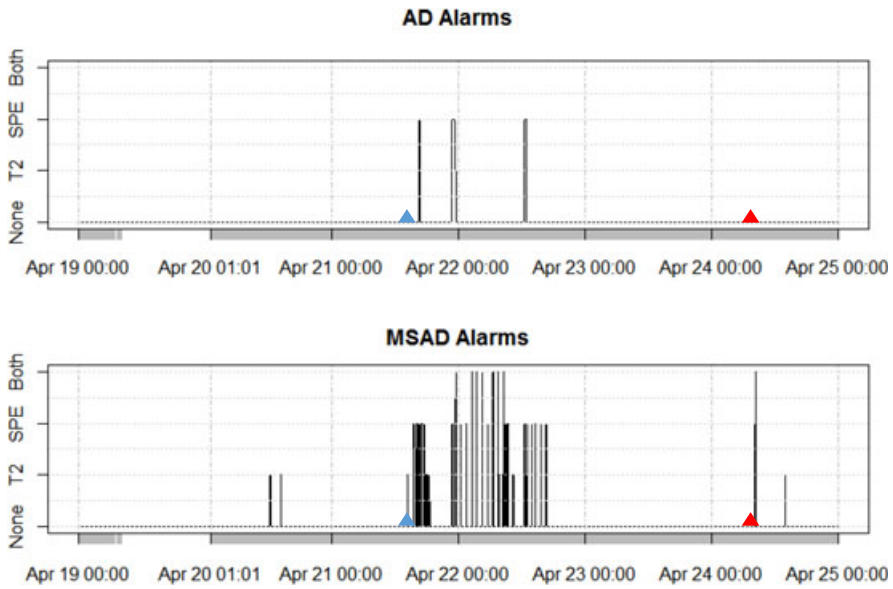

FIGURE 6 Alarms issued after applying adaptive-dynamic principal component analysis (*top*) and multistate adaptive-dynamic principal components analysis (MSAD-PCA) (*bottom*) to a system fault that occurred between April 21 and April 24, 2010. The *y*-axis represents the categories for when an observation triggers no alarms, a T^2 alarm, an *SPE* alarm, or both alarms. The blue triangles are at 14:15 on April 21, when the MSAD-PCA monitoring method first detects a fault. The red triangle at 10:00 on April 24 is when the human operators detected a fault [Colour figure can be viewed at wileyonlinelibrary.com]

After applying the MSAD-PCA method, the first alarms were issued around noon on April 20, but after inspecting the feature process graphs, we believe that these first alarms were triggered by the observations directly following a period of missing data on the morning of the 20th and not by the onset of a system fault. However, Figure 6 (*bottom*) shows that the MSAD-PCA method triggers alarms starting at 14:15 on April 21, 2 hours before AD-PCA. Moreover, these alarms are persistent for longer than a 24-hour period (from 14:15 on April 21 to 17:00 on April 22), and some of these observations trigger both the *SPE* and T^2 alarms. Based on these results, we believe that MSAD-PCA will warn system operators of faults earlier, and it will more consistently detect faults as they occur in real time.

6 | CONCLUSION

In conclusion, we have illustrated how incorporating state information when monitoring a complex process characterized by nonlinear, nonstationary, autocorrelated, and non-Gaussian features can improve fault detection results. Using both a new simulation design with faults introduced within a state and a real motivating example from a decentralized WWT facility, we have seen that faults occurring within a state can be detected faster and more consistently when using a multistate monitoring approach. In addition, as long as the data are sufficient to estimate the state-specific parameters, results when applying a multistate approach to a truly single-state process are not inferior to results from a single-state monitoring method.

When choosing which states to include in a model, we recommend having at least $p^2/2$ IC observations per state for training since MSAD-PCA estimates state-specific covariance matrices. Consequently, we also recommend that the frequency of membership for each state not be drastically dissimilar. For example, 5 states with membership probabilities of $[0.2, 0.2, 0.2, 0.2, 0.2]$, $[0.25, 0.11, 0.07, 0.36, 0.21]$, or $[0.6, 0.1, 0.1, 0.1, 0.1]$ would be acceptable, but a membership probability vector $[0.96, 0.01, 0.01, 0.01, 0.01]$ could cause problems. Furthermore, we recommend considering states in which the process distribution, mean vector, and/or covariance matrix change significantly between states. Finally, in the interest of model parsimony, we do not recommend including too many states. In short, block on states meeting these criteria: states that have different vector or matrix moments, states with each training sample size at least $p^2/2$, and states with similar frequencies.

To conclude, we recommend monitoring multistate multivariate processes by pairing the *SPE* and T^2 monitoring statistics after dimension reduction with MSAD-PCA. The simulation and case study show that the *SPE* monitoring statistic offers an excellent combination of sensitivity and high detection probability, while the T^2 monitoring statistic offers an excellent combination of specificity and high detection probability. This approach can provide engineers with early warning of faults, allowing them to respond quickly. Monitoring within mutually exclusive states can also help identify the cause of a fault when one occurs in a specific state, and this is our next step in process monitoring research.[31]

Software

To implement the MSAD-PCA process in `R`, our package `mvMonitoring` is available from CRAN via https://CRAN.R-project.org/package=mvMonitoring. Functions to simulate the data and faults used in our simulation study are also provided. See https://gabrielodom.github.io/mvMonitoring/ for selected vignettes.

ACKNOWLEDGEMENTS

This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under award OSR-2015-CRG4-2582 and by the Partnerships for Innovation: Building Innovation Capacity program of the National Science Foundation under award 1632227. We would also like to thank an associate editor and referee whose anonymous comments helped improve the content and presentation of this work.

REFERENCES

1. Stavropoulos P, Chantzis D, Doukas C, Papacharalampopoulos A, Chryssolouris G. Monitoring and control of manufacturing processes: a review. *Procedia CIRP*. 2013;8:421-425.
2. United States Environmental Protection Agency. Wastewater technology fact sheet: In-plant pump stations. Tech Rep EPA 472 832-F-00-069; 2000.

3. Gikas P, Tchobanoglous G. The role of satellite and decentralized strategies in water resources management. *J Environ Manage*. 2009;90(1):144-152.
4. Leverenz HL, Asano T. Wastewater reclamation and reuse system. In: Wilderer P, ed. *Treatise on Water Science*. Oxford, UK: Elsevier; 2011:63-71.
5. Vuono D, Henkel J, Benecke J, et al. Flexible hybrid membrane treatment systems for tailored nutrient management: a new paradigm in urban wastewater treatment. *J Membrane Sci*. 2013;446:34-41.
6. Kazor K, Holloway RW, Cath TY, Hering AS. Comparison of linear and nonlinear dimension reduction techniques for automated process monitoring of a decentralized wastewater treatment facility. *Stoch Environ Res Risk Assess*. 2016;30(5):1527-1544.
7. Choi SW, Lee IB. Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chem Eng Sci*. 2004;59(24):5897-5908.
8. Chouaib C, Mohamed-Faouzi H, Messaoud D. Adaptive kernel principal component analysis for nonlinear dynamic process monitoring. Paper presented at: 2013 9th Asian Control Conference (ASCC); 2013; Istanbul, Turkey.
9. Scholkopf B, Smola A, Muller KR. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*. 1998;10(5):1299-1319.
10. Miao A, Song Z, Ge Z, Zhou L, Wen Q. Nonlinear fault detection based on locally linear embedding. *J Control Theory Appl*. 2013;11(4):615-622.
11. Zhang W, Liu XY, Qi RL, Jiang Y. Improved locally linear embedding based method for nonlinear system fault detection. *Chi Acad Sci Int J Adv Comp Tech*. 2013;5(1).
12. Gertler J, Li W, Huang Y, McAvoy TJ. Isolation enhanced principal component analysis. *AIChE J*. 1999;45(2):323.
13. Kano M, Nagao K, Hasebe S, et al. Comparison of statistical process monitoring methods: application to the Eastman challenge problem. *Comput Chem Eng*. 2000;24(2):175-181.
14. Garcia-Muñoz S, Kourti T, MacGregor JF. Model predictive monitoring for batch processes. *Ind Eng Chem Res*. 2004;43(18):5929-5941.
15. Wang J, He QP. Multivariate statistical process monitoring based on statistics pattern analysis. *Ind Eng Chem Res*. 2010;49(17):7858-7869.
16. Wise BM, Veltkamp DJ, Davis B, Ricker NL, Kowalski BR. Principal components analysis for monitoring the West Valley Liquid Fed Ceramic Melter. In: Proceedings of the Symposium on Waste Management. 1988; Tucson, AZ.
17. Kresta JV, MacGregor JF, Marlin TE. Multivariate statistical monitoring of process operating performance. *Can J Chem Eng*. 1991;69(1):35-47.
18. Baggiani F, Marsili-Libelli S. Real-time fault detection and isolation in biological wastewater treatment plants. *Water Sci Technol*. 2009;60(11):2949-2961.
19. Sanchez-Fernández A, Fuente MJ, Sainz-Palmero GI. Fault detection in wastewater treatment plants using distributed PCA methods. Paper presented at: 20th IEEE International Conference on Emerging Technologies Factory Automation (ETFA); 2015; Luxembourg City, Luxembourg.
20. Jearkpaporn D, Borror CM, Runger GC, Montgomery DC. Process monitoring for mean shifts for multiple stage processes. *Int J Prod Res*. 2007;45(23):5547-5570.
21. Asadzadeh S, Aghaie A, Yang SF. Monitoring and diagnosing multistage processes: a review of cause selecting control charts. *J Ind Sys Eng*. 2008;2(3):214-235.
22. Li Y, Tsung F. False discovery rate-adjusted charting schemes for multistage process monitoring and fault identification. *Technometrics*. 2012;51(2):186-205.
23. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 5th ed. Upper Saddle River, NJ: Prentice Hall; 2002.
24. Ge Z, Song Z. *Multivariate Statistical Process Control: Process Monitoring Methods and Applications*. London, UK: Springer-Verlag London; 2012.
25. Rato TJ, Reis MS. Defining the structure of DPCA models and its impact on process monitoring and prediction activities. *Chemometr Intell Lab*. 2013;125:74-86.
26. Qiu P. *Introduction to Statistical Process Control*. Boca Raton, FL: CRC Press; 2013.
27. Sheather S. *A Modern Approach to Regression with R*. New York, NY: Springer; 2009.
28. Qiu P, Li Z. On nonparametric statistical process control of univariate processes. *Technometrics*. 2012;43(4):390-405.
29. Chen N, Zi X, Zou C. A distribution-free multivariate control chart. *Technometrics*. 2016;58(4):448-459.
30. Dong Dong, McAvoy TJ. Batch tracking via nonlinear principal component analysis. *AIChE J*. 1996;42(8):2199-2208.
31. Zou C, Jiang W, Tsung F. A LASSO-based diagnostic framework for multivariate statistical process control. *Technometrics*. 2011;53(3):297-309.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Odom GJ, Newhart KB, Cath TY, Hering AS. Multistate multivariate statistical process control. *Appl Stochastic Models Bus Ind*. 2018;1–13. https://doi.org/10.1002/asmb.2333