An Improved Methodology for Outlier Detection in Dynamic Datasets

Shu Xu, Michael Baldea, and Thomas F. Edgar
McKetta Dept. of Chemical Engineering, The University of Texas at Austin, Austin, TX, 78712

Willy Wojsznis, Terrence Blevins, and Mark Nixon
Process Systems and Solutions, Emerson Process Management, Round Rock, TX, 78759

DOI 10.1002/aic.14631
Published online October 12, 2014 in Wiley Online Library (wileyonlinelibrary.com)

A time series Kalman filter (TSKF) is proposed that successfully handles outlier detection in dynamic systems, where normal process changes often mask the existence of outliers. The TSKF method combines a time series model fitting procedure with a modified Kalman filter to deal with additive outlier and innovational outlier detection problems in dynamic process dataset. Compared with current outlier detection methods, the new method enjoys the following advantages: (a) no prior knowledge of the process model is needed; (b) it is easy to tune; (c) it can be applied to both univariate and multivariate outlier detection; (d) it is applicable to both on-line and off-line operation; (e) it cleans outliers while maintains the integrity of the original dataset. © 2014 American Institute of Chemical Engineers *AIChE J*, 61: 419–433, 2015

Keywords: outlier detection, Kalman filter, time series modeling, additive outlier, innovational outlier, dynamic process modeling

Introduction

In the process industries, the development of cheap electronic storage devices allows engineers to access a wealth of high-frequency sampling data without compression, but it also poses a new challenge for process engineers on drawing valuable information from gigabits of stored data. A direct solution is to go through a knowledge discovery process that includes data cleaning and integration, selection and transformation, data mining, evaluation, and presentation steps.[1] Since data cleaning is a fundamental step for knowledge discovery, it is necessary to explore fast and effective data cleaning methods.

One of the most important work in data cleaning is to remove outliers which might compromise model behavior, parameter estimation, and data analysis results. In a statistical sense, an outlier is defined as an observation or a subset of observations that exhibits an inconsistent behavior with the remainder of the set of data.[2] In the process industries, outliers might be generated by malfunction of sensors, or human-related errors such as inappropriate treatment of missing data. Outliers should be removed before mining steps such as data reconciliation and gross error detection,[3] regression,[4] and system identification.[5,6]

If outliers only occur within a single variable context, they are called univariate outliers; in contrast, multivariate outliers occur when combinations of variables cross a certain boundary. Multivariate outliers are commonly treated as univariate ones for simplicity but applying such a procedure has several disadvantages.[7] First, outliers in one variable might be caused by abnormalities in other variables, and ignoring such a possibility will give rise to over-specification of number of outliers. Second, an outlier with moderate size that affects all variables might be masked by process changes in univariate detection procedure, so it is necessary to take into consideration interactions or joint dynamics between variables.

Outlier detection differs from noise removal because the noise filters contain certain reconciliation procedures that will compromise the integrity of normal observations. Outlier detection methods only identify and replace outliers based on analysis on normal data behavior, and no interpolation or approximation is applied on normal data.

A key assumption underlying many statistical outlier detection methods[2,8–11]is that the data are identically and independently distributed (*i.i.d.*), which is often compromised by dynamics hidden in time-varying datasets. A related issue is how to differentiate process dynamics and outliers. Traditionally, a moving window technique is applied, which assumes that data in a small enough moving window can still be treated as identically and independently distributed. However, applying moving window techniques does not always give satisfactory results, especially when the variations in the dataset are significant, and it is always computationally expensive for large datasets. Another solution is approximating variations in the data by time-series models, such as an autoregressive model (AR), and then separating observations that are inconsistent with the imposed model using outlier detection techniques proposed in the time series

Correspondence concerning this article should be addressed to T. F. Edgar at tfedgar@austin.utexas.edu.

literature, such as likelihood-based methods.[7,12–18] However, those methods are only applicable to small datasets with a small number of outliers; besides, there is a lack of discussion on industrial applications and related parameter tuning issues. The on-line filter-cleaner[19] has been applied to univariate outlier detection in an industry dataset, and outperforms other time series outlier detection methods in robustness and efficiency. However, the filter-cleaner[19] can still be improved in several aspects such as the model fitting algorithm, parameter tuning, and an extension to the multivariate cases. Besides, since off-line data analyses are more encountered in practice, it is useful to develop a related off-line outlier detection method. Finally, detecting innovational outliers (IOs), which are commonly encountered in dynamic processes, is also worth studying.

The article is organized as follows: Basic concepts in outlier detection section is an introduction of basic concepts in outlier detection, time-series model fitting algorithms, and model order selection criterion. Outlier detection in time series data section provides a review of outlier detection in time-series data. Time series Kalman filter section describes the time series Kalman filter (TSKF) with both off-line and on-line versions, and related parameter tuning. In Simulation testing results, both univariate and multivariate datasets are simulated for on-line and off-line testing. For additive outlier (AO) detection, the univariate results are compared with the on-line filter-cleaner[19] and the Hampel identifier[8]; the multivariate results are compared with principal component analysis (PCA), and dynamic PCA methods,[20,21] which detect outliers based on the Hotelling's T^2 metric.[22] For IO detection, the univariate and multivariate results are compared with the Hampel identifier and dynamic principal component analysis (DPCA) & PCA methods, respectively. In Plant data testing results section, actual plant data are used to test the TSKF method. Conclusions and future work are shown in Conclusion and future work section.

Basic Concepts in Outlier Detection

Robustness and breakdown points

For outlier detection in *i.i.d.* datasets, the location and the scatter estimation are critical steps for almost every statistical method. In univariate cases, one pair of commonly used location and scatter parameters is the sample mean μ and the variance σ^2, another pair is the median med and the median absolute deviation MAD, and they can be calculated based on Eqs. 1 and 2[9]

$$\mu=\frac{\sum_{t=1}^{N} x_t}{N},\sigma^2=\frac{\sum_{t=1}^{N} (x_t-\mu)^2}{N-1} \tag{1}$$

$$\text{med}=\frac{x_{[(N+1)/2]:N}+x_{[N/2]+1:N}}{2}, \quad \text{MAD}=\text{med}(|x_1-\text{med}|,...,|x_N-\text{med}|) \tag{2}$$

in which $[A]$ rounds A to the nearest integer less than or equal to A, and $x_{1:N},...,x_{N:N}$ are the ordered sequence of $\{x_t\}$.

Well-known outlier detection methods based on above location and scatter estimates are the 3σ edit rule and the Hampel identifier,[23] the detection conditions of which are shown in Eqs. 3 and 4

$$|x_t-\mu| > 3\sigma \tag{3}$$

$$|x_t-\text{med}| > g(N,\alpha_N)\,\text{MAD} \tag{4}$$

where $g(N,\alpha_N)$ is a function related to observation numbers N and significance level α_N. Both location and scatter parameters are affected by the outliers, and to estimate the robustness of a method against outliers, the concept of breakdown point was proposed by Hampel.[8] The breakdown point was defined as the smallest percentage of outliers that an estimator could withstand. Outliers affect the estimators in two different ways: (1) the masking effect, which affects the estimator's capability to detect certain outliers; (2) the swamping effect, which leads to normal data mistaken for outliers. Correspondingly, the breakdown points can be categorized as masking and swamping breakdown points.[24] For 3 σ edit rule, its masking breakdown point is $1/(N+1)$ and swamping breakdown point 100%, which suggests outliers are likely to protect themselves from being detected, but once being detected as an outlier, it will not be a false alarm. For the Hampel identifier, its masking breakdown point is 50%, and its swamping break down point approaches 50% when the sample size increases. In practice, the Hampel identifier is considered as a robust method for removing univariate outliers in *i.i.d.* datasets. A self-regulatory method is proposed which can robustly estimate mean, variance, and detect univariate outliers for different large datasets.[25]

In multivariate cases, the reweighted minimum covariance determinant (MCD) estimator[10,11,26] has a masking breakdown point of 50%, making it a desirable choice in outlier detection and estimating the location and scatter parameters in multivariate *i.i.d.* datasets. If the time-dependence between observations of variables is not considered, there are other ways to remove multivariate outliers, including the closed distance to center combined with the ellipsoidal multivariate trimming method,[27] proposed (PROP) function-based method,[28] iteratively reweighted partial least squares (PLS),[29] maxiumum correntropy estimator,[30] self-organizing map,[31] and PCA.

Contamination rate, detection rate, misidentification rate, and normal data estimation rate

The contamination rate κ is defined as the percentage of outliers in total data, and the detection rate χ is defined as the percentage of outliers being successfully identified. The misidentification rate β is defined as the percentage of normal data falsely tagged as outliers (type I error), and γ is defined as a prior estimation of the percentage of normal data in the original dataset. Normally, γ is set to be larger than 80%; otherwise, the data will be considered as of poor quality and useless. Mathematical expressions for κ, χ, β, and γ are shown in Eqs. 5–8

$$\kappa=\frac{N_{\text{outliers}}}{N_{\text{total data}}} \tag{5}$$

$$\chi=\frac{N_{\text{successfully identified}}}{N_{\text{total outliers}}} \tag{6}$$

$$\beta=\frac{N_{\text{false alarm}}}{N_{\text{normal data}}} \tag{7}$$

$$\gamma=\frac{N_{\text{normal data}}}{N_{\text{total data}}} \tag{8}$$

Based on the definition, κ=5% in a dataset with 10,000 observations means the number of actual outliers is 500. To simplify the verification process of detection methods, we are using simulated datasets with outliers added at every 20th sample points.

Univariate AR fitting

In practice, many discrete random processes can be approximated by a stationary ARMA (p, q) model shown in Eq. 9

$$\left(1+\sum_{i=1}^{p}\phi_i z^{-i}\right)x_t=\left(1+\sum_{i=1}^{q}\theta_i z^{-i}\right)\epsilon_t \tag{9}$$

where p, q are model orders, ϕ_i, θ_i are coefficients, ϵ_i is the white noise of the model, $\epsilon_t \sim N\left(0, \sigma_\epsilon^2\right)$, and z^{-1} is a shift operator.[32]

Based on the Wold decomposition theorem[33] and Kolmogorov's theorem,[34] any ARMA process can be represented by an AR process of infinite order. Thus, a feasible solution is to fit an AR (p) model as shown in Eq. 10 to approximate the changes exhibited in the ARMA process

$$\left(1+\sum_{i=1}^{p}\phi_i z^{-i}\right)x_t=\epsilon_t \tag{10}$$

Three different methods are commonly used in estimating ϕ_i in AR (p) model shown in Eq. 10: the least-squares method, the Yule–Walker method, and Burg's method.[35] The previous two methods involve an inverse of the auto-covariance matrix step, while the Burg's method calculates the reflection coefficients and then applies Levinson recursion to obtain the AR parameter estimates. De Hoon et al.[36] compared the above methods and presented simulation results showing that because the first two methods invert an auto-covariance matrices which can be poorly conditioned, Burg's method is preferable to the least-squares and the Yule–Walker approach.

Multivariate (vector) autoregressive model fitting

For Multivariate (vector) autoregressive (MVAR) model fitting, an extension of Yule–Walker method to multivariate cases can be applied[37] as well as the multivariate Burg's method.[38,39] A new estimator (Arfit) has been proposed,[40,41] and by comparing the above multivariate estimators, the multivariate Burg's method still outperforms the others.[42,43]

Model order selection

When selecting a model order, a balance has to be made in between improving the coefficient of determination R^2 and a prevention of model over-fitting. For AR model order selection, a commonly used criterion is the Akaike information criterion (AIC)[44] shown in Eq. 11

$$\mathrm{AIC}(p)=N\log\hat{\boldsymbol{\rho}}_p+2p \tag{11}$$

Another way to select model order is the Schwarz's Bayesian information criterion (BIC) shown in Eq. 12,[45] which can be applied in both AR and MVAR models

$$\mathrm{BIC}(p)=\frac{l_p}{m}-\left(1-\frac{n_p}{N}\right)\log N \tag{12}$$

where p is the model order, m is the number of variables, N is the number of observations, n_p is the number of model parameters, and

$$l_p=\log\left\{\det\left[\left(N-n_p\right)\hat{\boldsymbol{\rho}}_p\right]\right\} \tag{13}$$

$\hat{\boldsymbol{\rho}}_p$ stands for residual covariance matrix, $\det[.]$ calculates the matrix's determinant. The QR factorization algorithm is applied in evaluating $\hat{\boldsymbol{\rho}}_p$, and a regularization term δD^2, where δ is a coefficient and D^2 is a positive definite diagonal matrix, is added to deal with the situation when $\hat{\boldsymbol{\rho}}_p$ becomes ill-conditioned.[40,41]

For the univariate case, $m=1$, and

$$l_p=\log\left\{\left|\left(N-n_p\right)\hat{\boldsymbol{\rho}}_p\right|\right\} \tag{14}$$

Because no AIC calculation equation has been found for multivariate cases, the BIC is used in the new algorithm. Normally, the best model order corresponds to the lowest BIC or AIC value. The outliers will contaminate the dataset and give rise to a large model order p. Thus, a prewhitening procedure is needed to reduce the outlier effects on model order estimation. After prewhitening, a small p usually suffices, similar results have been reported.[19] In this article's simulation study, p is selected to be 2.

Outlier Detection in Time Series Data

Univariate outlier detection

Since dynamic data are time dependent, it is worth exploring techniques used in time series analysis to detect outliers in dynamic systems. There are two basic types of outliers in time series models shown in Eq. 9, the AO and the IO, defined by Eqs. 15 and 16, respectively

$$y_t=v_t\delta_t^{(T)}+x_t \tag{15}$$

$$y_t=\frac{\left(1+\sum_{i=1}^{q}\theta_i z^{-i}\right)}{\left(1+\sum_{i=1}^{p}\phi_i z^{-i}\right)}v_t\delta_t^{(T)}+x_t \tag{16}$$

where $\delta_t^{(T)}$ is a pulse function: $\delta_t^{(T)}=1$ if $t=T$; $\delta_t^{(T)}=0$ if $t \neq T$. v_t is an outlier. From the above equations, we can see that AOs affect the observed time series only at time T, while the IOs impact a finite number of observations in a stationary process. If the time series is not stationary, for example, observations follow an autoregressive integrated moving average (ARIMA) model, the IOs will permanently affect the observations and lead to new types of outliers such as a transient level change (TC) or permanent level shift(LS).[13,32] In practice, the AOs are likely to be caused by inappropriate treatment of missing data, and the IOs are likely to caused by process disturbances. In this article, the AO detection and IO detection in stationary ARMA and VARMA processes are discussed.

Generally, four types of methods have been developed for univariate outlier detection in a given time-series dataset. The first kind is the likelihood-based methods: they detect outlier based on maximizing likelihod functions. The first likelihood-based method is developed by Fox[14] for single outlier detection (AO, IO) in the autoregressive (AR) process, and it is then modified for AO and IO detection in the ARIMA process,[12,16,18] and for detecting TCs or LSs.[13,16,18] Since the likelihood function is key to Bayesian inference, Bayesian methods were developed to detect outliers based on analyzing posterior distribution.[46–48]

The second type is the deletion diagnostic-based methods, which involve a iterative deletion step and a diagnostic step. They have been used for AO and IO detection in ARMA processes[49] and ARIMA processes.[50] The results obtained by the likelihood-based methods and the deletion diagnostic-based method are compared and shown to be very close.[51]

The third kind of methods is the influence functional (IF)-based method, which involve evaluating certain IFs such as the Mahalanobis distance. The methods have been applied to detect AOs and IOs in AR processes[52] and ARIMA processes.[53]

The previous outlier detection methods have only been applied to small datasets with only a few outliers due to a high computational cost and difficulties in deriving metrics such as the likelihood function. Moreover, none of them provides information on parameter tuning for different datasets. The fourth type of method is the on-line filter-cleaner,[19] which combines the time-series modeling with a AR model-based filter-cleaner.[54] The method has been tested in the process datasets and obtained a good outlier detection result. However, the method uses the Yule–Walker method to estimate the model, which does not work well when the auto-covariance matrices become ill-conditioned. In addition, the method cannot be applied to detect multivariate outliers.

Multivariate outlier detection

Similar to the univariate case, there exist the same types of multivariate outliers: the AO, the IO, the permanent LS, and the TC. A common way to remove outliers in multivariate datasets is to treat each component separately and apply univariate outlier removal techniques. However, such a procedure fails to take into consideration the joint dynamics between variables and leads to some multivariate outliers being masked. For multivariate outlier detection in given time series datasets, only a few methods have been reported. The first kind are the likelihood-based methods, which have been applied in detecting a level shift at unknown time in general vector autoregressive (VAR) processes,[17] and all four types of outliers in a simple VAR process[15] or a vector autoregressive integrated moving average (VARIMA) process.[7]

The second type of method is projection-pursuit based,[55] which detects and processes multivariate outliers in some projection direction. The method outperforms a direct testing on original VARIMA processes.

A gentic algorithm (GA)[56] has been reported recently to detect multiple isolated and consecutive AOs in time series. The iterative GA procedure detects outliers by searching the chromosome which minimizes a certain fitness function and is more flexible and adaptive than the sequential detection algorithm.[7]

However, all methods shown above have only been applied to small testing datasets with a low outlier contamination rate ($\kappa < 5\%$), and there is no discussion about how to tune related parameters.

The PCA method has been applied to detect multivariate outliers. First, PCA decomposes the original dataset using singular value decomposition shown in Eq. 17. Second, it calculates the Hotelling's T^2 metric defined in Eq. 18, and by monitoring such a metric, disturbances or abnormalities can be detected and isolated if the metrics violate the threshold calculated in Eq. 19

$$\mathbf{X}_{\mathbf{PCA}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{17}$$

where **X** is the training data matrix with n observations and m variables

$$t^2 = \mathbf{x}^T V \Sigma_a^{-2} V^T \mathbf{x} \tag{18}$$

where V contains the loading vectors associated with the a largest singular values, Σ_a includes the first a rows and columns of Σ, and **x** is an observation vector of dimension m

$$T_\alpha^2 = \frac{(n-1)a}{(n-a)} F_\alpha(a, n-a) \tag{19}$$

where $F_\alpha(a, n-a)$ is the critical point of the F-distribution with a and $n - a$ degrees of freedom, and α is the level of significance.

However, the static PCA method fails to take into consideration the serial correlation at different time instances, and one way to improve the PCA method is to recursively sum the last a few scores and construct new metrics.[57] The DPCA[20,21] is another way proposed for detection and isolation of process disturbances in time series data. The DPCA augments each observation **X** matrix at time t with the previous l time instances, as shown in Eq. 20, and similar to PCA, it decomposes $\mathbf{X}_{\mathbf{DPCA}}$ and detects outliers by monitoring the Hotelling's T^2

$$\mathbf{X}_{\mathrm{DPCA}}(l) = [\mathbf{X}(t)\mathbf{X}(t-1)\cdots\mathbf{X}(t-l)] \tag{20}$$

where $\mathbf{X}(t)$ is the observation matrix at time instance t.

Time Series Kalman Filter

Inspired by the filter-cleaner[19] (shown in Appendix), the TSKF is proposed which differs in several aspects:

1. Burg's model estimation is applied. In multivariate cases, the auto-covariance matrices might become ill-conditioned; thus, Burg's method is preferred to the Yule–Walker method.

2. Parameter estimation is directly obtained from the preliminary clean dataset, which reduces efforts to get a robust estimation of the auto-covariance matrix via reweighted MCD.

3. Instead of applying the filter-cleaner[54] which simultaneously detects and replaces outliers with related predicted values, the detection and clean steps can be separated: for on-line use, the users can do both at the same time, and for off-line use, they also can replace the outliers after the detection process is finished, options are available for applying the best imputation techniques to replace the outliers for a specific application. In the new procedure, for simplicity, neighboring normal points will be used to replace outliers instead of using a model predicted value, because in practice, no actual model exists; imposing a fitted model, although a robust one, will compromise the integrity of the original dataset and might lead to a new spike such as shown in Figure 1 (close to time point 600).

4. Besides the AOs, the simulation cases include IOs.

Both off-line and on-line versions of the method are provided. For simplicity, the method is written in the format for multivariate outlier detection.

Off-line version

Given a dataset $\{\mathbf{y}_t\}_{t=1}^N$, we can detect and replace the outliers with the following steps:

1. Data partition: partition the dataset into M subsets, $\{\mathbf{y}_t\}_{t=1}^{N_i}, i=1,2,\ldots,M$.

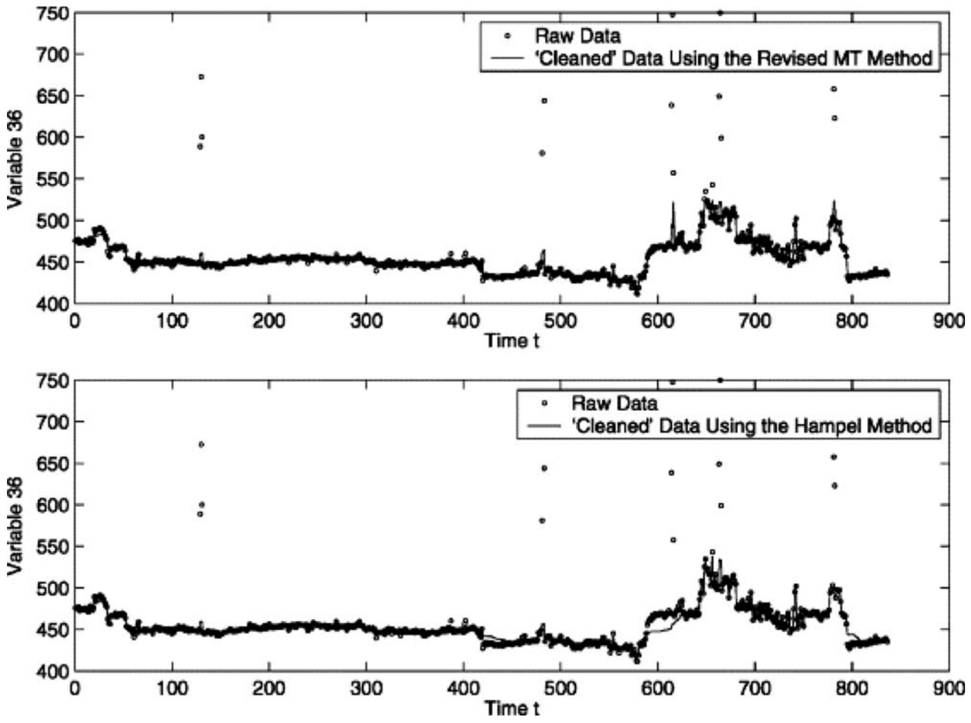


Figure 1. Testing results of filter-cleaner and the Hampel method.[19]

2. Prewhitening: for each subset $\{\mathbf{y}_t\}_{t=1}^{N_i}, i=1,2,...,M$, prewhiten the data using the reweighted MCD estimator, replace the outliers with robust center $\boldsymbol{\mu}_i$, and centralize the data with $\boldsymbol{\mu}_i$.

3. Model fitting: based on the preliminary clean data $\{\mathbf{y}_t^c\}_{t=1}^{N_i}$,

3.1. (Optional) select the model order p according to BIC.

3.2. Calculate the model coefficients $\boldsymbol{\Phi}_i$ based on Burg's method.

4. Outlier detection: for each subset $\{\mathbf{y}_t\}_{t=1}^{N_i}$, $i=1,2,...,M$:

4.1. Reformat

$$\begin{aligned}\mathbf{Y}_t&=\Theta\mathbf{Y}_{t-1}+\mathbf{U}_t\\ \mathbf{y}_t&=\mathbf{H}\mathbf{Y}_t\end{aligned}\tag{21}$$

where

$$\mathbf{Y}_t^T=\left[\mathbf{y}_t,\mathbf{y}_{t-1},...,\mathbf{y}_{t-p+1}\right]_{1\times pm}\tag{22}$$

$$\mathbf{U}_t^T=[\hat{\boldsymbol{\epsilon}},\mathbf{0},...,\mathbf{0}]_{1\times pm};\hat{\boldsymbol{\epsilon}}\sim N(\mathbf{0},\mathbf{Q})\tag{23}$$

$$\mathbf{H}=[\mathbf{I}_{m\times m},\mathbf{0},...,\mathbf{0}]_{1\times pm}\tag{24}$$

$$\Theta=\begin{bmatrix}\boldsymbol{\Phi}_{1,m\times m} & \boldsymbol{\Phi}_{2,m\times m} & \cdots & \boldsymbol{\Phi}_{p-1,m\times m} & \boldsymbol{\Phi}_{p,m\times m}\\ \mathbf{I}_{m\times m} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}_{m\times m} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{m\times m} & \mathbf{0}\end{bmatrix}_{pm\times pm}\tag{25}$$

4.2. Predict

$$\hat{\mathbf{Y}}_{t|t-1}=\Theta\hat{\mathbf{Y}}_{t-1|t-1}\tag{26}$$

$$\mathbf{P}_{t|t-1}=\Theta\mathbf{P}_{t-1|t-1}\Theta^T+\mathbf{Q}\tag{27}$$

4.3. Update

$$\mathbf{E}_t=\mathbf{y}_t-\mathbf{H}\hat{\mathbf{Y}}_{t|t-1}\tag{28}$$

$$\mathbf{S}_t=\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T+\tau\mathbf{I}\tag{29}$$

$$d_t=\sqrt{\mathbf{E}_t^T\mathbf{S}_t^{-1}\mathbf{E}_t}\tag{30}$$

$$\mathbf{K}_t=\mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}\tag{31}$$

$$\hat{\mathbf{Y}}_{t|t}=\hat{\mathbf{Y}}_{t|t-1}+\mathbf{K}_t\mathbf{E}_t\tag{32}$$

$$\mathbf{P}_{t|t}=(\mathbf{I}-\mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}\tag{33}$$

4.4. Detect

4.4.1. Set $\Delta=0$.

4.4.2. Find a number of n observations whose Mahalanobis distance $d_t\geq\Delta$.

4.4.3. Calculate the percentage of normal data

$$\xi=\frac{N^i-n}{N^i}\tag{34}$$

4.4.4. If $\xi\geq\gamma$,stop; else increase Δ by $d\Delta$.

4.4.5. The outliers correspond to observations with Mahalanobis distance

$$d_t\geq\Delta_{\text{final}}.$$

Table 1. Tuning Parameters of the TSKF Method

Implementation	Tuning Parameters
on-line	distance threshold Δ; moving window size (MW)
off-line	γ; partition window size (PW)

Table 2. Brief Summation of Simulation Cases

Models	AO	IO	Test Points Number	
			on-line	off-line
ARMA(1,1)	✓	✓	10000	10000
VARMA(1,1)	✓	✓	1000	10000

4.5. Replace: replace the outliers with neighboring normal values.

On-line version

The on-line version of the method is very similar to the on-line filter-cleaner[19]:

Given a data sequence at time, we can detect and replace the outliers in the following steps:

1. Choose a dataset $\{\mathbf{y}_t\}_{t-M+1:t}^{M}$ with window size moving window size (MW).

2. The prewhitening and modeling fitting steps are the same as the off-line version.

3. Based on a preset threshold Δ, if the Mahalanobis distance $d_t \geq \Delta$, the observation is identified as an outlier.

4. Replace the outliers with neighboring normal values.

The new procedure keeps most of the original structure of the Kalman filter[58] unchanged, and only makes a few modifications shown in Eqs. 29 and 30. In Eq. 29, the variance of observation noise term $\mathbf{R}$ shown in Eq. 36 of the original Kalman filter is deleted, which makes the new algorithm detect outliers without changing the original observations. A Tikhonov regularization term[59] $\tau\mathbf{I}$ is added to deal with ill-conditioned covariance matrices in multivariate outlier detection cases. If we add the observation noise terms back, Eqs. 21 and 29 become

$$\begin{aligned}\mathbf{Y}_t &= \Theta\mathbf{Y}_{t-1} + \mathbf{U}_t \\ \mathbf{y}_t &= \mathbf{H}\mathbf{Y}_t + \mathbf{z}_t\end{aligned} \tag{35}$$

$$\mathbf{S}_t = \mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R} \tag{36}$$

where $\mathbf{z}_t \sim N(\mathbf{0}, \mathbf{R})$.

A univariate version of the method is similar, except that a univariate Mahalanobis distance is calculated as shown in Eq. 37

$$d_t = |e_t| / \sqrt{s_t} \tag{37}$$

As we can see from the method description shown above, the TSKF method has a high computational cost largely due to the outlier detection step, which tracks the variances of each observation point. Such a step becomes even slower when dealing with multivariate outlier detection cases.

Parameter tuning

An important evaluation standard for an outlier detection method is whether contains tuning parameters. Table 1 gives the tuning parameters for the implementation of the TSKF method.

For on-line use, since we have no knowledge of how to preset the distance threshold Δ, we need to run a training dataset and record the Mahalanobis distance d_t, which can be used to set the value of Δ.

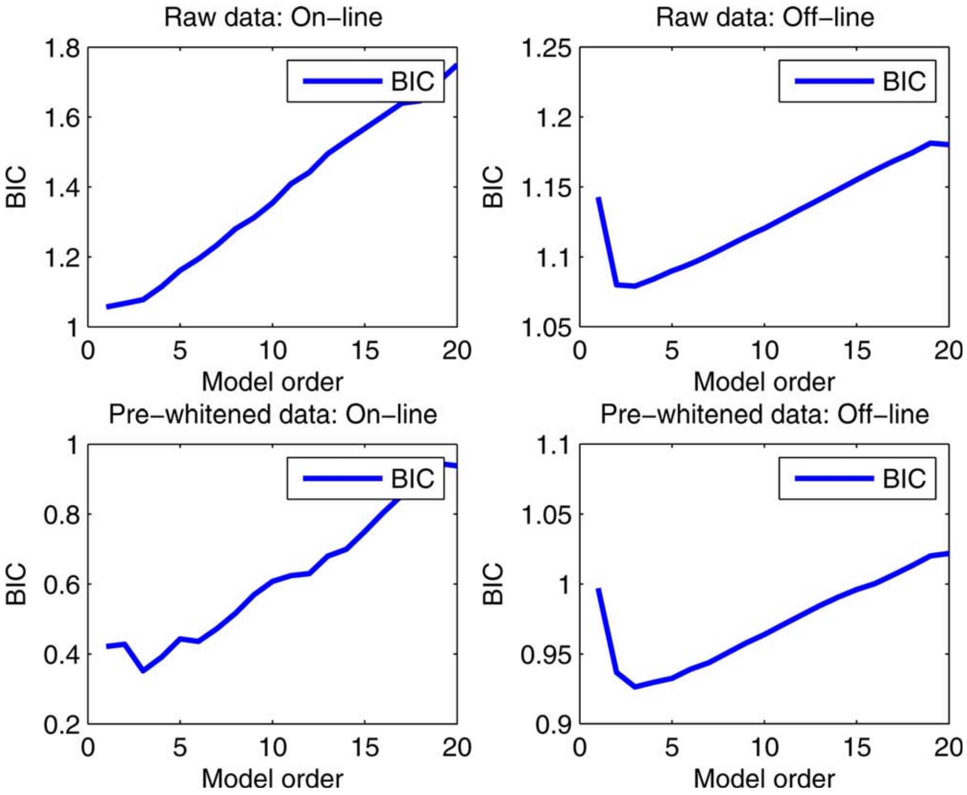


Figure 2. Model order selection for AOs.

Simulation condition: ARMA (1,1), ϕ=0.9; θ = 0; Amp = 4; κ=5%. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 3. Model Impact Analysis, κ=5%

	On-Line[a]		Off-Line[b]	
	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$
True model	85.40	2.04	78.37	1.05
Prewhitened data	82.34	2.11	75.47	1.17
Raw data	78.34	2.15	70.12	1.33

[a] Simulation condition: $N = 10{,}000$; rep = 500; MW = 100.
[b] Simulation condition: $N = 10{,}000$; rep = 500; PW = 1000, γ=95%.

For off-line use, we can set the γ based on prior knowledge of the raw dataset, such as γ=95% means we estimate that the maximum amount of outliers in the dataset should not exceed 5% of the total number of observations. γ can be tuned easily without repeating the recursion calculation of the Mahalanobis distance d_t.

Simulation Testing Results

For illustration, the on-line and off-line versions of the TSKF method will be applied to both univariate and multivariate outlier detection in the dynamic datasets. For simplicity, only the univariate simulation process is described: following the approach used by Liu et al.,[19] we obtain data by simulating the AO and IO models shown in Eqs. 15 and 16; v_t has equal probabilities being $+$Amp or $-$Amp (Amp is the amplitude of outliers); x_t follows ARMA(1,1) processes for univariate cases. The multivariate simulation process is the similar as the univariate one.

The univariate model was run with 10,000 test points for both on-line and off-line cases. For processes with AOs, the data are corrupted with outliers at different contamination rates defined in Eq. 5. For processes with IOs cases, IOs are added every 100th sample points in ARMA(1,1) processes.

The multivariate models are run with 10,000 test points for off-line case and with 1000 test points for on-line case. Similar to the univariate model, data are contaminated with different rates of outliers for the AO process simulation. For processes with IOs cases, IOs are added every 100th sample points in VARMA(1,1) processes.

A summation of simulation cases is shown in Table 2. In addition, for on-line testing, the MW is set to be 100, and for off-line testing, the partition window size is set to be 1000.

Furthermore, although the definitions of outlier detection rate χ and misidentification rate β work perfectly for AO detection, some modifications need to be made, as shown in Eqs. 38 and 39, so that they can be applied to IO detection

$$\chi_{\mathrm{IO}} = \frac{N_{\text{successfully identified } \delta_t^{(T)}=1}}{N_{\text{total } \delta_t^{(T)}=1}} \tag{38}$$

$$\beta_{\mathrm{IO}} = \frac{N_{\text{false alarm prior to } \delta_t^{(T)}=1}}{N_{\text{normal data}}} \tag{39}$$

As shown in Eq. 38, the χ_{IO} is defined as the percentage of successfully identified locations of pulse function ($\delta_t^{(T)}=1$) leading to the IOs. In simulation cases, the original dataset is divided into subsets with 100 observations each, and one pulse function is added on each subset. Based on the definition of β_{IO} shown in Eq. 39, any identified outliers prior to the preset locations of pulse function $\delta_t^{(T)}=1$ within each subset will be regarded as false alarms.

Model impact analysis and order selection

The cleanness of the preliminary clean data will affect the performance of the outlier detection, and to demonstrate the necessity of a prewhitening step, a detailed discussion is given based on the simulation case: an ARMA(1,1) model with ϕ=0.9, θ=0, and Amp = 4.

The true model of the process is expressed as Eq. 40

$$\left(1-0.9z^{-1}\right) x_t = \epsilon_t \tag{40}$$

We first estimate the model order based on BIC shown in Eq. 12 for raw data of single MW (on-line) and prewhitened data of single partition window size (off-line). Figure 2 shows that the BIC results are not significantly affected by the prewhitening step; also a lower model order of 1 or 2 usually suffices. Thus, to prevent model over-fitting, the order is selected to be 2 for both on-line and off-line implementation.

Next, we perform the TSKF method with a true model shown in Eq. 40 and a estimated model based on raw data, the simulation results are summarized in Table 3.

As shown in Table 3, surprisingly, applying the true model can only obtain a slightly better result than the one built on prewhitened data and raw data. A possible explanation is that the Mahalanobis distance d_t is not sensitive enough and some changes of local variances caused by the outliers are overshadowed by local process dynamics. In other words, although some increases are shown in d_t as outliers, they are not large enough to violate the threshold and raise an alarm. Furthermore, excluding the prewhitening step from the method can negatively affect the detection results, although not significantly, because outliers affect the

Table 4. Additive Outlier Detection Rates for Data from ARMA (1, 1) Process at κ=5%

					On-Line[a]				Off-Line[b]			
					Liu's filter-cleaner[c]		TSKF		Hampel[c]		TSKF	
Case No.	ϕ	θ	Δ	Amp	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$
1	0.0	0.0	2.5	3	0.82	65.13	1.51	70.12	0.72	70.91	1.42	69.03
2	0.0	0.0	2.6	4	0.55	82.83	1.07	91.78	0.63	84.23	0.50	88.17
3	0.0	0.0	2.6	5	0.43	95.41	1.12	99.15	0.44	96.41	0.11	95.76
4	0.0	−0.5	2.5	3	1.01	63.35	1.77	68.08	0.92	50.32	1.64	67.47
5	0.0	−0.5	2.7	4	0.54	78.24	1.48	89.4	0.86	73.85	0.65	85.34
6	0.0	−0.9	3.0	4	0.58	65.87	1.80	80.75	0.81	53.89	1.10	76.68
7	0.5	0.0	2.5	3	1.12	64.38	1.82	67.45	0.94	49.52	1.62	64.65
8	0.5	0.0	2.7	4	0.49	82.44	1.48	89.69	0.71	73.65	0.73	83.92
9	0.9	0.0	3.0	4	0.59	79.84	2.11	82.34	0.47	12.57	1.17	75.47

[a] Simulation condition: N = 10000; rep = 500; MW = 100.
[b] Simulation condition: N=10000; rep=500; PW=1000, γ=95%.
[c] Some results come from Liu et al.[19]

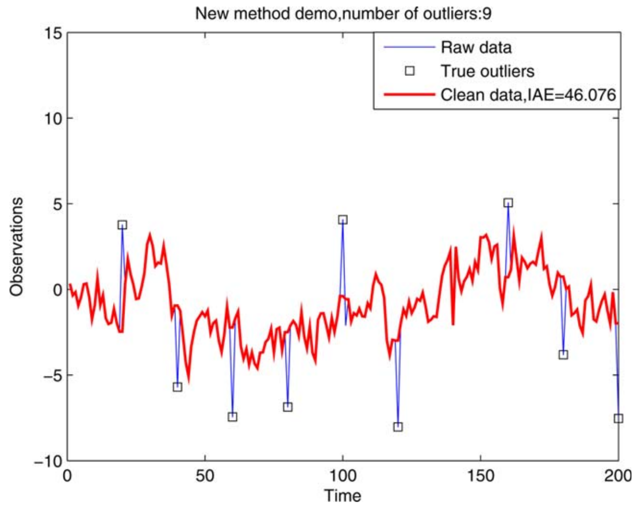


Figure 3. TSKF method for AOs.

Simulation condition: ARMA (1,1), ϕ=0.9; $\theta = 0$; Amp = 5; κ=5%; MW = 100. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

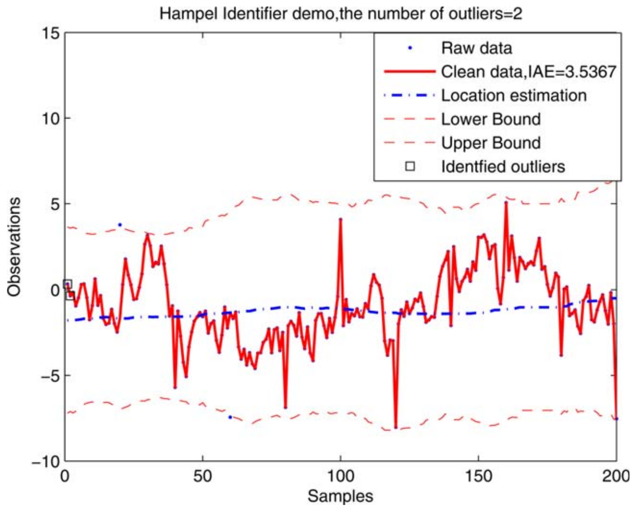


Figure 4. The Hampel identifier for AOs.

Simulation condition: ARMA (1,1), ϕ=0.9; $\theta = 0$; Amp = 5; κ=5%; MW = 100. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

estimation of center μ of the data, leading to a gap showing between raw data and prewhitened data results.

ARMA(1,1)model

$$(1-\phi z^{-1})x_t=(1-\theta z^{-1})\epsilon_t \tag{41}$$

where ϕ, θ are coefficients, ϵ_t is the white noise, $\epsilon_t \sim N(0,1)$ and z^{-1} is a shift operator.

AO Detection. The on-line and off-line AO detection results for data from ARMA(1,1) processes are shown in Tables 4.

From Table 4, we can see that for on-line outlier detection rate χ, the TSKF method obtains results close to Liu's filter-cleaner,[19] and both work better than the Hampel identifier when the process autocorrelation becomes high (shown in the first-order correlation coefficient ϕ). The result suggests that system dynamics affect the Hampel identifier more significantly.

Moreover, a larger outlier size Amp will help the outlier detection by increasing the outlier detection rate χ, but it will not necessarily decrease the misidentification rate β for the TSKF method.

In addition, although detecting more AOs than the Hampel identifier and on-line filter-cleaner, the TSKF method has a higher misidentification rate β than the later two, and it increases with a higher correlation ϕ. Such a phenomenon can be attributed to that Δ is set to be a constant value and does not change as the process evolves. However, possible metrics to monitor the process changes, such as the variance and mean, will all be negatively affected by the outliers. Thus, the auto-adjustment of Δ although may lower the misidentification rate β, will decrease the detection rate χ, as validated by several additional simulations.

Last but not least. Although the on-line and off-line results of the TSKF method are close, it is worth mentioning that the off-line results can be obtained much faster than the on-line results since no moving window is applied.

Figures 3 and 4 show that the TSKF method is able to detect more AOs in dynamic process than the Hampel identifier.

IO Detection. The on-line and off-line IO detection results for data from ARMA(1,1) processes are shown in Tables 5.

By analyzing results shown in Table 5, we can see that similar to the AO detection results in Table 4, the TSKF method works much better than the Hampel identifier in IO detection, especially when the process autocorrelation becomes high (shown in the first-order correlation coefficient ϕ).

Furthermore, for the TSKF method, the detection rates of IOs do not show a lot of differences. Because unlike the AOs, the effect coming from interactions between IOs and the system dynamics, only lasts for a finite number of observations and is neutralized by a lower contamination rate(1%). This makes the IOs more difficult to detect. Even

Table 5. Innovational Outlier Detection Results for Data from ARMA (1,1) Processes

					On-Line[a]				Off-Line[b]	
					TSKF		Hampel[b]		TSKF	
Case No.	ϕ	θ	Δ	Amp	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$	$\beta(/\%)$	$\chi(/\%)$
1	0.0	0.0	2.6	5	1.06	74.08	0.33	73.98	0.18	72.72
2	0.0	−0.5	2.7	5	0.84	75.54	0.45	69.02	0.17	72.58
3	0.0	−0.9	3.0	5	0.98	72.64	0.75	56.21	0.22	68.97
4	0.5	0.0	2.7	5	0.77	74.44	0.44	67.17	0.18	72.56
5	0.9	0.0	3.0	5	0.32	72.67	0.61	13.88	0.18	72.12

[a] Simulation condition: N = 10,000; rep = 500; MW = 100.
[b] Simulation condition: N = 10,000; rep = 500; γ=99%

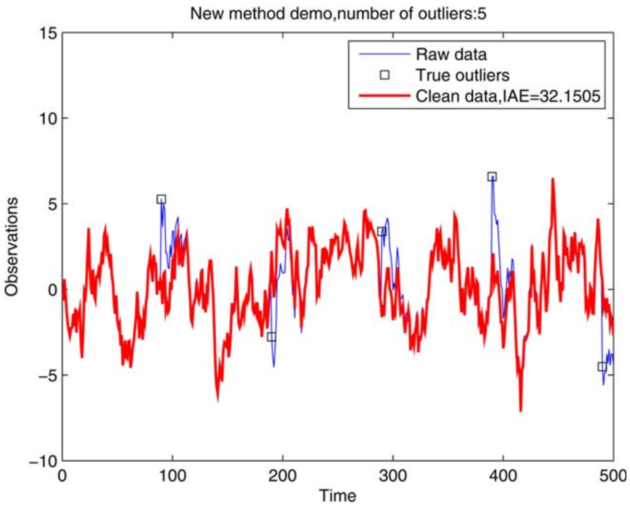


Figure 5. TSKF method for IOs.

Simulation condition: ARMA (1, 1), ϕ=0.9; $\theta = 0$; Amp = 5; MW = 100. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

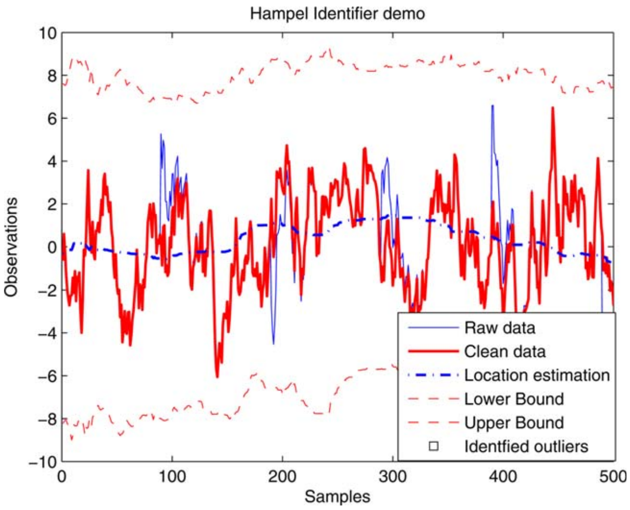


Figure 6. The Hampel identifier for IOs.

Simulation condition: ARMA (1,1), ϕ=0.9; $\theta = 0$; Amp = 5; MW = 100. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

though the IOs have amplitudes of 5, the TSKF method cannot guarantee a 100% IO detection rate χ every simulation run, and sometimes the detection rate χ is only close to 50%. Thus, the average detection rates are less than 80% and the differences are negligible.

In comparison with Figure 3, Figure 5 exhibits that unlike the AOs, the IOs will impact a finite number of observations afterwards. Compared with Figure 6, Figure 5 also illustrates that the TSKF method is able to detect more IOs than the Hampel identifier.

VARMA(1,1)model

$$\left(\mathbf{I}-\boldsymbol{\Phi}z^{-1}\right)\mathbf{x}_t=\left(\mathbf{I}-\boldsymbol{\Omega}z^{-1}\right)\epsilon_t \tag{42}$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\Omega}$ are coefficient matrices, ϵ_t is the driving noise of the model, $\epsilon_t \sim N\left(\mathbf{0}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$ and z^{-1} is a shift operator.

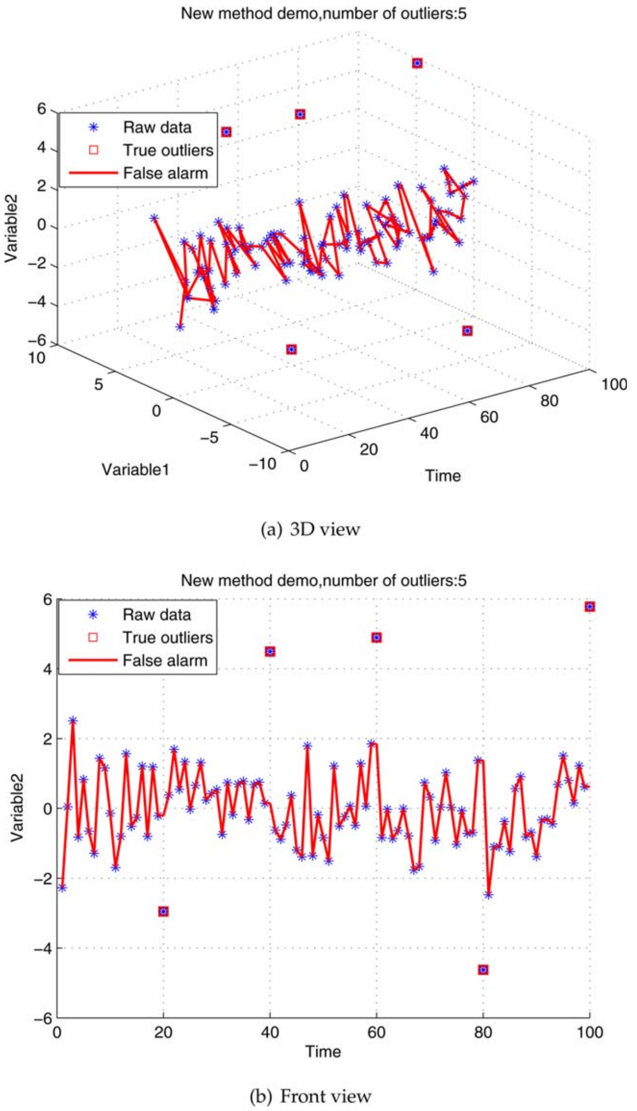
AO Detection. The on-line and off-line AO detection results for data from VARMA(1,1) processes are shown in Table 6.

As for parameter settings, based on additional simulations, increasing the model complexity by choosing a larger parameter l does not obtain a better performance. The parameter a is chosen to be 2 for on-line and 3 for off-line to ensure the DPCA model captures close to 90% total variance of the dataset and to prevent model over-fitting at the same time. Although increasing the significance level α will help raise the detection rate,

Table 6. Additive Outlier Detection Rate $\chi(/\%)$ for Data from VARMA (1,1) Process at κ=5%

					On-Line[a]			Off-Line[b]		
Case No.	$\boldsymbol{\Phi}$	$\boldsymbol{\Omega}$	$\boldsymbol{\Delta}$	$\begin{bmatrix}\text{Amp}\\\text{Amp}\end{bmatrix}$	TSKF	DPCA(I)	PCA(II)	TSKF	DPCA(III)	PCA(IV)
1	**0.0**	**0.0**	3.2	$\begin{bmatrix}3.0\\3.0\end{bmatrix}$	90.22	71.82	72.24	88.32	45.16	66.57
2	**0.0**	**0.0**	3.2	$\begin{bmatrix}4.0\\4.0\end{bmatrix}$	97.11	86.92	88.72	93.74	66.93	87.11
3	**0.0**	**0.0**	3.2	$\begin{bmatrix}5.0\\5.0\end{bmatrix}$	100	95.72	95.78	97.60	81.79	96.27
4	$\begin{bmatrix}0.5 & 0\\0 & 0.5\end{bmatrix}$	$\begin{bmatrix}0.5 & 0\\0 & 0.5\end{bmatrix}$	3.5	$\begin{bmatrix}4.0\\4.0\end{bmatrix}$	98.22	87.12	89.88	93.69	66.74	87.39
5	$\begin{bmatrix}0.5 & 0.5\\0.5 & 0.5\end{bmatrix}$	$\begin{bmatrix}0.5 & 0.5\\0.5 & 0.5\end{bmatrix}$	3.5	$\begin{bmatrix}4.0\\4.0\end{bmatrix}$	95.56	87.80	89.06	93.72	66.75	87.41
6	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	3.3	$\begin{bmatrix}3.0\\3.0\end{bmatrix}$	88.47	69.02	71.42	85.21	45.24	66.33
7	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	3.3	$\begin{bmatrix}4.0\\4.0\end{bmatrix}$	94.22	87.52	88.66	93.64	66.57	87.38
8	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	$\begin{bmatrix}0.9 & 0.1\\0.1 & 0.9\end{bmatrix}$	3.3	$\begin{bmatrix}5.0\\5.0\end{bmatrix}$	99.11	95.22	95.78	97.62	81.84	96.35

[a] Simulation condition: $N = 1000$; rep = 100; MW = 100.
[b] Simulation condition: $N = 10{,}000$; rep = 100; PW = 1000; γ=95%.
(I) Parameter setting: α=0.01; $l = 1$; $a = 2$.
(II) Parameter setting: α=0.01; $a = 1$.
(III) Parameter setting: α=0.01; $l = 1$; $a = 3$.
(IV) Parameter setting: α=0.01; $a = 1$.



(a) 3D view

(b) Front view

Figure 7. AO detection results obtained by the TSKF method.

Simulation condition: VARMA(1,1); on-line; case No. = 5. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

especially during off-line implementation, it will significantly affect the results by leading to a much larger misidentification rate β (larger than 10%). Thus, α was chosen to be 0.01.

By analyzing results shown in Table 6, we can see that the TSKF method generally obtains a higher outlier detection rate χ than the DPCA and PCA method. Surprisingly, the DPCA method does not obtain as good detection rates as PCA, especially during off-line implementation. Such a result can be largely attributed to that without a prewhitening step, an outlier contamination rate of 5% will severely damage the serial correlation in augmented matrix shown in Eq. 20 and lead to inaccurate t^2 results.

Moreover, a larger outlier size Amp will help the outlier detection by increasing the outlier detection rate χ, while increasing autocorrelation $\mathbf{\Phi}$ will negatively affect the outlier detection results.

Figure 7 shows the on-line multivariate AO detection results of the TSKF method for a VARMA(1,1) process. Figure 8 shows the Hotelling's T^2 record on the first moving window of the DPCA method, it needs to be pointed that although time points 40 and 80 are missed, they can be successfully detected in moving windows afterwards.

IO Detection. The on-line and off-line IO detection results for data from VARMA(1,1) processes are shown in Table 7. It is found that the DPCA with a significance level α=0.01 obtains desirable results for both on-line and off-line cases. In addition, the parameter a is chosen to be 3 and l to be 1 for the same reason discussed in univariate outlier cases.

Furthermore, for the same reasons as discussed in ARMA(1,1) cases, IO detection results of TSKF, PCA, and DPCA in Table 7 are close and close to 75%. It is worth mentioning that the PCA and DPCA methods are faster than the TSKF method in obtaining the results.

Figures 9 and 10 show the multivariate IO detection results of the TSKF method and the DPCA method for a VARMA(1,1) process, respectively. Comparing these two figures, we can see both methods correctly identified the location of pulse function in the first moving window.

Summary and discussion of simulation testing results

Based on the simulation results, we can see that while the AOs only affect single observations, IOs will have an impact on a finite number of observations after they appear. A larger autocorrelation contained in the process data will negatively affect the detection results, while a larger outlier size will help the outlier detection.

Furthermore, although the interactions between IOs and the system dynamics make them more difficult to be detected than the AOs, a desirable detection rate χ can still be obtained if the contamination rate κ is low and the amplitude of IOs is high.

Although only the outlier detection results of ARMA(1,1) and VARMA(1,1) processes are shown, the TSKF method works well for higher order stationary process data, such as ARMA(2,1) and VARMA(2,1). The TSKF method works better than the Hampel identifier, PCA and DPCA methods

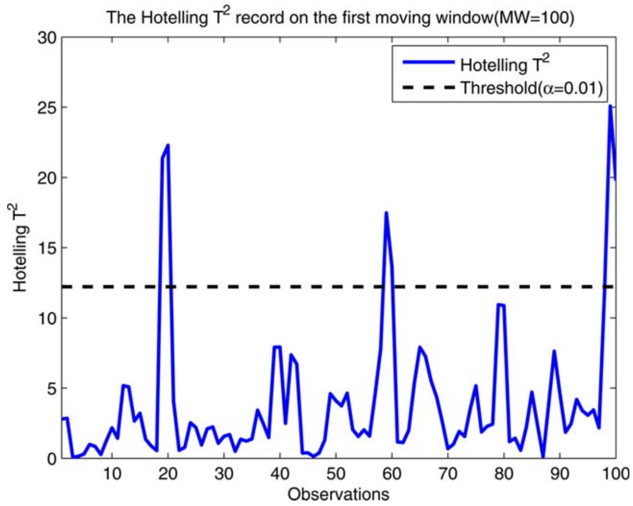


Figure 8. Hotelling's T^2 record on the first moving window of dynamic PCA.

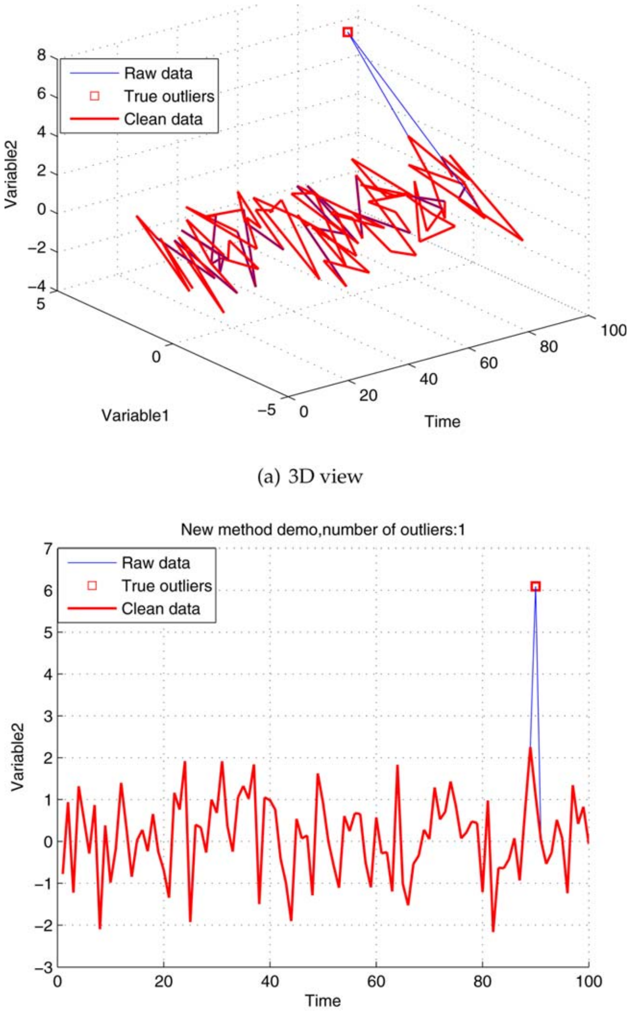
Simulation condition: VARMA(1,1); on-line; κ=5%; case No.= 5; AO. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Table 7. Innovational Outlier Detection Rate $\chi(/\%)$ for Data from VARMA (1,1) Processes

Case No.	Φ	Ω	Δ	$\begin{bmatrix}\text{Amp}\\ \text{Amp}\end{bmatrix}$	On-Line[a]			Off-Line[b]		
					TSKF	DPCA(I)	PCA(II)	TSKF	DPCA(III)	PCA(IV)
1	**0.0**	**0.0**	3.2	$\begin{bmatrix}5.0\\ 5.0\end{bmatrix}$	74.07	75.00	75.10	74.99	75.25	74.14
2	$\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}$	$\begin{bmatrix}0.5 & 0\\ 0 & 0.5\end{bmatrix}$	3.5	$\begin{bmatrix}5.0\\ 5.0\end{bmatrix}$	77.78	75.40	74.70	75.13	74.99	75.04
3	$\begin{bmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{bmatrix}$	$\begin{bmatrix}0.5 & 0.5\\ 0.5 & 0.5\end{bmatrix}$	3.5	$\begin{bmatrix}5.0\\ 5.0\end{bmatrix}$	79.33	76.80	76.0	74.44	75.02	74.73
4	$\begin{bmatrix}0.9 & 0.1\\ 0.1 & 0.9\end{bmatrix}$	$\begin{bmatrix}0.9 & 0.1\\ 0.1 & 0.9\end{bmatrix}$	3.3	$\begin{bmatrix}5.0\\ 5.0\end{bmatrix}$	75.11	76.30	76.40	75.58	74.28	74.84

[a] Simulation condition: $N = 1000$; rep = 100; MW = 100.
[b] Simulation condition: $N = 10{,}000$; rep = 100; γ=99%.
(I) Parameter setting: α=0.01; $l = 1$; $a = 3$.
(II) Parameter setting: α=0.01; $a = 1$.
(III) Parameter setting: α=0.01; $l = 1$; $a = 3$.
(IV) Parameter setting: α=0.01; $a = 1$.

in detection of AOs. For IOs, the Hampel identifier becomes incompetent in detecting them when a high autocorrelation exists. The TSKF, PCA, and DPCA still obtain close results



(a) 3D view

(b) Front view

Figure 9. IO detection results obtained by the TSKF method.

Simulation condition: VARMA(1,1); on-line; case No. = 3; $N = 1000$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

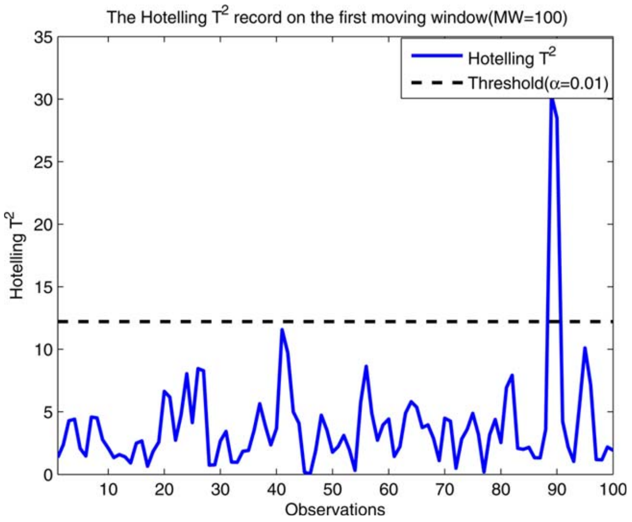


Figure 10. Hotelling's T^2 record on the first moving window of dynamic PCA.

Simulation condition: VARMA(1,1); on-line; case No.=3; $N = 1000$; IO. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

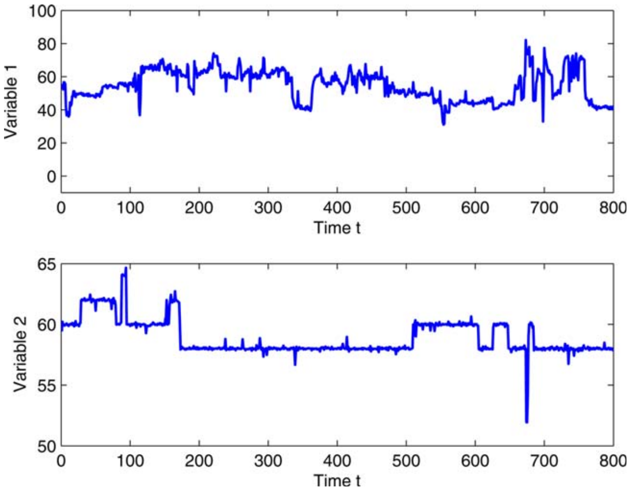


Figure 11. Raw plant data.

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

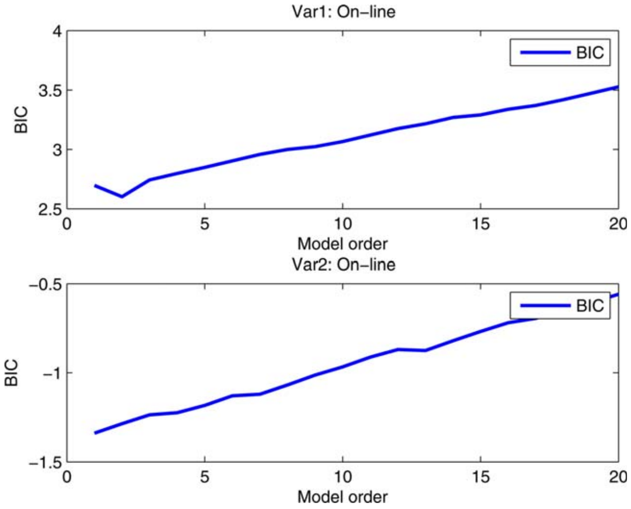


Figure 12. BIC of raw plant data at single moving window.

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

in detecting IOs, but the computational cost of the TSKF method is higher.

Plant Data Testing Results

In this section, an industrial dataset obtained from a chemical plant is used to perform on-line testing of the TSKF method.

Two variables shown in Figure 11 have been selected from the process dataset because outliers and dynamic changes are present in both variables. The missing values have been replaced with local means. While the second variable shows many step changes, the first variable contains dynamic responses caused by a series of step changes.

The model order selection is based on BIC for two variables, as shown in Figure 12. Although Var1 and Var2 vary with time, a model order of 1 or 2 suffice. In this case, we pick 2.

The on-line outlier detection process is carried out as follows: each variable is cleaned alone by the TSKF method and the Hampel identifier. The results are plotted in Figures 13–16. Compare Figure 13 with 14, we can see that the TSKF method is able to detect more spikes between time points 200 and 300 than the Hampel identifier, and successfully replaces an outlier shown at time point 700. Similar results can also be found near time point 700 in Figures 15 and 16.

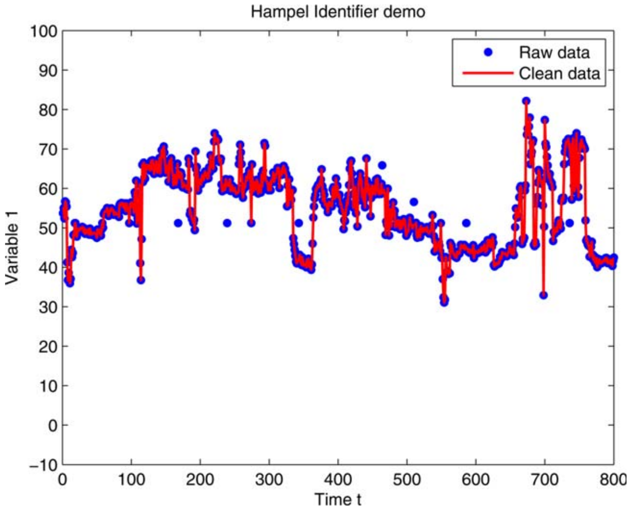


Figure 14. The Hampel identifier for V1.

Simulation condition: MW = 10; on-line testing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Conclusions and Future Work

In this article, a new method (TSKF) for outlier detection has been proposed that is suitable for both univariate and multivariate outlier detection in dynamic datasets. Both on-line and off-line versions and related parameter tuning for the new method have been given.

Different from the on-line filter-cleaner,[19] the TSKF method incorporates the Burg-type time series model fitting algorithm, which ensures stability of the method when dealing with ill-conditioned auto-covariance matrices in multivariate cases. In addition, neighboring normal points will be

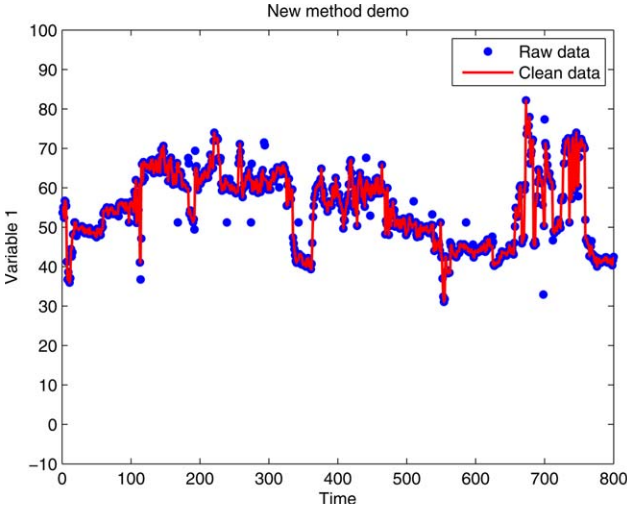


Figure 13. The TSKF method for V1.

Simulation condition: Δ = 5; MW = 10; on-line testing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

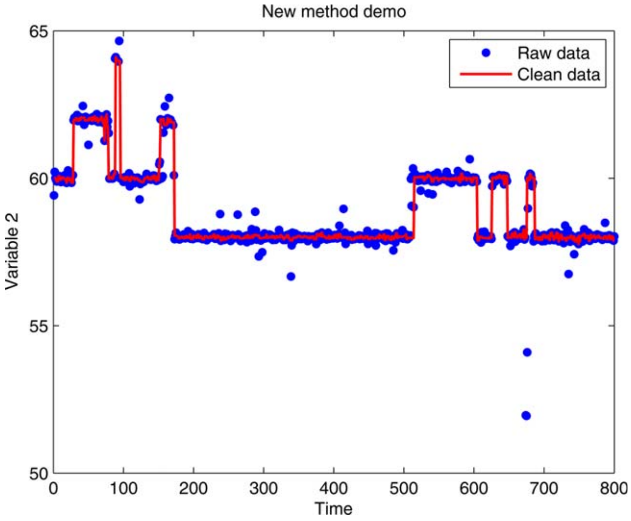


Figure 15. The TSKF method for V2.

Simulation condition: Δ=1.5; MW = 10; on-line testing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

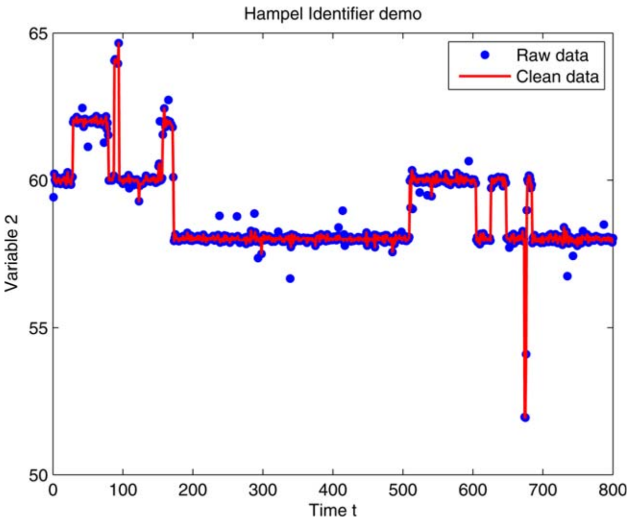


Figure 16. The Hampel identifier for V2.

Simulation condition: MW = 10; on-line testing. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

used to replace outliers instead of using imposed model predictive value. Moreover, obtaining model parameters directly from the preliminary clean dataset has a lower computational complexity in comparison with the procedure used in on-line filter-cleaner (converting the original data to more complicated matrices and applying reweighted MCD method to estimate parameters).

Based on the simulation testing results, the TSKF method outperforms the Hampel identifier and the dynamic PCA method in AO detection in an ARMA(1,1) process and a VARMA(1,1) process, respectively. Interestingly, for IOs, the TSKF method although outperforms the Hampel identifier in an ARMA(1,1) process, does not differ a lot from PCA and DPCA in a VARMA(1,1) process, due to combining effects of interactions between IOs and system dynamics, as well as contamination rate and outlier amplitude, as discussed in simulation section. It is worth mentioning that the computational cost of the TSKF method is higher than the PCA and DPCA method, which makes it a less competitive choice.

Based on actual plant data testing results, the TSKF method is able to detect more univariate outliers than the Hampel identifier.

Last but not least, tests have been made on nonstationary processes, in which the IOs will lead to permanent parameter drifts and shifts, and the results of TSKF on detecting those changes are not desirable. Thus, the method still needs to be improved on dealing with such problems in the future work.

Acknowledgments

The authors would like to gratefully acknowledge the financial and technical support from the Emerson Process Management. The authors would also like to thank Ricardo Dunia for their insightful feedback and constructive suggestions. Material presented in this publication reflects opinions of the authors, and not of their institutional affiliations.

Notation

α = Significance level
β = Mis-identification rate(Type I error)
γ = Normal data estimation rate
Δ = Threshold for outlier identification
ϵ_t = white noise in the ARMA model
$\boldsymbol{\epsilon}_t$ = white noise in the VARMA model
θ = The autoregressive model coefficients
Θ = State transition matrix
κ = Normal data estimation rate
μ = Sample mean
σ^2 = Sample variance
χ = Outlier detection rate
ϕ = The autoregressive model coefficients
$\boldsymbol{\Phi}$ = Multivariate (vector) autoregressive model coefficient matrix
$\boldsymbol{\Omega}$ = Multivariate (vector) autoregressive model coefficient matrix
AO = Additive outlier
AR(p) = Autoregressive model with order p
AIC = Akaike information criterion
Amp = Outlier size
ARMA(p, q) = Autoregressive moving average model with order p, q
ARIMA = Autoregressive integrated moving average model
BIC = Bayesian information criterion
$i.i.d$ = Identically and independent distributed
IO = Innovational outlier
LS = Permanent level shift
m = Variable number in a given dataset
med = Sample median
MAD = Sample median absolute deviation
MVAR = Multivariate (vector) autoregressive model with order p
N = Observation number in a given sample dataset
n_p = Number of AR model parameters
rep = Repetition time
TS = Transient level change
VARMA(p, q) = Vector autoregressive moving average model with order p, q

Literature Cited

1. Han J, Kamber M, Pei J. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann, 2006.
2. Barnett V, Lewis T. *Outliers in Statistical Data. Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, 2nd ed. Chichester: Wiley, 1984.
3. Albuquerque JS, Biegler LT. Data reconciliation and gross-error detection for dynamic systems. *AICHE J*. 1996;42(10):2841–2856.
4. Cui WT, Yan XF. Adaptive weighted least square support vector machine regression integrated with outlier detection and its application in QSAR. *Chemometr Intell Lab Syst*. 2009;98:130–135.
5. Liu Y, Chen JH. Correntropy kernel learning for nonlinear system identification with outliers. *Ind Eng Chem Res*. 2014;53(13):5248–5260.
6. AlMutawa J. Identification of errors-in-variables state space models with observation outliers based on minimum covariance determinant. *J Process Control*. 2009;19(5):879–887.
7. Tsay RS, Peña D, Pankratz AE. Outliers in multivariate time series. *Biometrika*. 2000;87(4):789–804.
8. Hampel FR. A general qualitative definition of robustness. *Ann Math Stat*. 1971;42(6):1887–1896.
9. Pearson RK. Outliers in process modeling and identification. *IEEE Trans Control Syst Technol*. 2002;10(1):55–63.
10. Rousseeuw PJ. Least median of squares regression. *J Am Stat Assoc*. 1984;79(388):871–880.
11. Rousseeuw PJ. Multivariate estimation with high breakdown point. *Math Stat Appl*. 1985;B:283–297.
12. Chang I, Tiao GC, Chen C. Estimation of time series parameters in the presence of outliers. *Technometrics*. 1988;30(2):193–204.
13. Chen C, Liu L-M. Joint estimation of model parameters and outlier effects in time series. *J Am Stat Assoc*. 1993;88(421):284–297.
14. Fox AJ. Outliers in time series. *J R Stat Soc Ser B (Methodological)*. 1972;34(3):350–363.
15. Franses PH, Lucas A. Outlier detection in cointegration analysis. *J Bus Econ Stat*. 1998;16(4):459–468.
16. Jesús SM, Peña D. The identification of multiple outliers in ARIMA Models. *Commun Stat - Theory Methods*. 2003;32(6):1265–1287.
17. Lütkepohl H, Saikkonen P, Trenkler C. Testing for the cointegrating rank of a VAR process with level shift at unknown time. *Econometrica*. 2004;72(2):647–662.

18. Tsay RS. Outliers, level shifts, and variance changes in time series. *J Forecasting*. 1988;7(1):1–20.
19. Liu H, Shah S, Jiang W. On-line outlier detection and data cleaning. *Comput Chem Eng*. 2004;28(9):1635–1647.
20. Ku W, Storer RH, Georgakis C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometr Intell Lab Syst*. 1995;30(1):179–196.
21. Russell EL, Chiang LH, Braatz RD. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometr Intell Lab Syst*. 2000;51(1):81–93.
22. Hotelling H. The generalization of student's ratio. *Ann Math Stat*. 1931;2(3):360–378.
23. Hampel FR. The influence curve and its role in robust estimation. *J Am Stat Assoc*. 1974;69(346):383–393.
24. Davies L, Gather U. The identification of multiple outliers. *J Am Stat Assoc*. 1993;88(423):782–792.
25. Buzzi-Ferraris G, Manenti F. Outlier detection in large data sets. *Comput Chem Eng*. 2011;35(2):388–390.
26. Rousseeuw PJ, Driessen KV. A fast Algorithm for the minimum covariance determinant estimator. *Technometrics*. 1999;41(3):212–223.
27. Chiang LH, Pell RJ, Seasholtz MB. Exploring process data with the use of robust outlier detection algorithms. *J Process Control*. 2003; 13(5):437–449.
28. Singh A. Outliers and robust procedures in some chemometric applications. *Chemometr Intell Lab Syst*. 1996;33:75–100.
29. Pell RJ. Multiple outlier detection for multivariate calibration using robust statistical techniques. *Chemometr Intell Lab Syst*. 2000;52:87–104.
30. Munoz JC, Chen JH. Removal of the effects of outliers in batch process data through maximum correntropy estimator. *Chemometr Intell Lab Syst*. 2012;111:53–58.
31. Yan XF. Multivariate outlier detection based on self-organizing map and adaptive nonlinear map and its application. *Chemometr Intell Lab Syst*. 2011;107:251–257.
32. Box GEP, Jenkins GM, Reinsel GC. *Time Series Analysis: Forecasting and Control*, 4th ed. New York: Wiley, 2013.
33. Wold HOA. A study in the analysis of stationary time series. *Almqvist & Wiksells boktrycheri-a.-b*. 1938.
34. Kolmogoroff A. Interpolation und extrapolation von stationaren zufalligen folgen. *Izvestiya Rossiiskoi Akademii Nauk Seriya Matematicheskaya*. 1941;5(1):3–14.
35. Kay SM. *Modern Spectral Estimation: Theory and Application*. NJ: Prentice Hall, 1988.
36. de Hoon MJL, van der Hagen THJJ, Schoonewelle H, van Dam H. Why Yule-Walker should not be used for autoregressive modelling. *Ann Nuclear Energy*. 1996;23(15):1219–1228.
37. Marple SL. *Digital Spectral Analysis with Applications*. NJ: Prentice Hall, 1987.
38. Nuttall AH. FORTRAN Program for multivariate linear predictive spectral analysis employing forward and backward averaging. *DTIC Document*. 1976.
39. Nuttall AH. Multivariate linear predictive spectral analysis employing weighted forward and nackward averaging: a generalization of Burg's algorithm. *DTIC Document*. 1976.
40. Neumaier A, Schneider T. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans Math Softw*. 2001;27(1):27–57.
41. Schneider T, Neumaier A. Algorithm 808: ARfit - a MATLAB package for the estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Trans Math Softw*. 2001;27(1):58–65.
42. Marple SL, Nuttall AH. Experimental comparison of three multichannel linear prediction spectral estimators. *IEE Proc F Commun Radar Signal Process*. 1983;130(3):218–229.
43. Schlögl A. A comparison of multivariate autoregressive estimators. *Signal Process*. 2006;86(9):2426–2429.
44. Akaike H. A new look at the statistical model identification. *IEEE Trans Automatic Control*. 1974;19(6):716–723.
45. Schwarz G. Estimating the dimension of a model. *Ann Stat*. 1978; 6(2):461–464.
46. Abraham B, Box GEP. Bayesian analysis of some outlier problems in time series. *Biometrika*. 1979;66(2):229–236.
47. Chaloner K, Brant R. A Bayesian approach to outlier detection and residual analysis. *Biometrika*. 1988;75(4):651–659.
48. Khatibisepehr S, Huang B. A Bayesian approach to robust process identification with ARX models. *AIChE J*. 2013;59(3):845–859.
49. Abraham B, Chuang A. Outlier detection and time series modeling. *Technometrics*. 1989;31(2):241–248.
50. Bianco AM, Garca Ben M, Martnez EJ, Yohai VJ. Outlier detection in regression models with ARIMA errors using robust estimates. *J Forecasting*. 2001;20(8):565–579.
51. Ljung GM. On outlier detection in time series. *J R Stat Soc Ser B (Methodological)*. 1993;55(2):559–567.
52. Martin RD, Yohai VJ. Influence functionals for time series. *Ann Stat*. 1986;14(3):781–818.
53. Peña D. Influential observations in time series. *J Bus Econ Stat*. 1990;8(2):235–241.
54. Martin RD, Thomson DJ. Robust-resistant spectrum estimation. *Proc IEEE*. 1982;70(9):1097–1115.
55. Galeano P, Peña D, Tsay RS. Outlier detection in multivariate time series by projection pursuit. *J Am Stat Assoc*. 2006;101(474):654–669.
56. Cucina D, di Salvatore A, Protopapas MK Outliers detection in multivariate time series using genetic algorithms. *Chemometr Intell Lab Syst*. 2014;132:103–110.
57. Wachs A, Lewin DR. Improved PCA methods for process disturbance and failure identification. *AIChE J*. 1999;45(8):1688–1700.
58. Kalman RE. A new approach to linear filtering and prediction problems. *J Basic Eng*. 1960;82(1):35–45.
59. Tikhonov AN, Arsenin VY. *Solution of Ill-Posed Problems*. Washington: Winston & Sons, 1977.
60. Huber PJ. *Robust Statistics*. New York: Wiley, 1981.

APPENDIX

The on-line filter-cleaner procedure

Given a univariate process data sequence $\{x_t\}_{t=1}^N$ at time t, the filter-cleaner detects outliers on-line following steps below[19]:

1. Choose a dataset $\{x_t\}_{t-M+1:t}^M$ with window size M.
2. Selection of AR order r.
3. Estimation of AR(r) model coefficient ϕ based on the dataset $\{x_t\}_{t-M+1:t}^M$.

3.1. Estimate the mean μ and variance c_0 of $\{x_t\}_{t-M+1:t}^M$ based on Hubers M-estimator.[60]

3.2. Form new multivariate datasets $\{X_i^k=(x_i, x_{i-k})\}_{i=t-M+k+1}^M (k=1,2,...,r)$. Obtain a robust estimation of the covariance matrix $\begin{bmatrix} c_{11}^k & c_{12}^k \\ c_{21}^k & c_{22}^k \end{bmatrix}$ of the kth multivariate dataset $\left\{X_i^k=(x_i,x_{i-k})\right\}_{i=t-M+k+1}^M$ by the reweighted MCD method.[10,11,26] The kth autocorrelation coefficient $\omega_k=c_{12}^k/\sqrt{c_{11}^k c_{22}^k}$.

3.3. Estimation of AR(r) model coefficient ϕ by solving Yule–Walker equations:

$$\omega_j=\phi_1\omega_{j-1}+\phi_2\omega_{j-2}+...+\phi_r\omega_{j-r}, j=1,...,r \tag{A1}$$

reformat Eq.A1

$$\phi=\begin{bmatrix}\phi_1\\ \phi_2\\ \vdots\\ \phi_r\end{bmatrix}, \omega=\begin{bmatrix}\omega_1\\ \omega_2\\ \vdots\\ \omega_r\end{bmatrix}, P=\begin{bmatrix}1 & \omega_1 & \cdots & \omega_r\\ \omega_1 & 1 & \cdots & \omega_{r-1}\\ \vdots & \vdots & \vdots & \vdots\\ \omega_r & \omega_{r-1} & \cdots & 1\end{bmatrix} \tag{A2}$$

so that $\phi=P^{-1}\omega$, and the process model can be expressed as

$$z_t=\frac{\mu}{1-\phi_1-\phi_2-...-\phi_r}+\phi_1 z_{t-1}+\phi_2 z_{t-2}+...+\phi_r z_{t-r}+\varepsilon_t. \tag{A3}$$

4. Filter and clean the current data point x_t.

4.1. Reformat the process model in the state-space form

$$Z_t = \Phi Z_{t-1} + U_t \tag{A4}$$

where

$$Z_t^T = [z_t, z_{t-1}, \dots, z_{t-r+1}] \tag{A5}$$

$$U_t^T = [\varepsilon_t, 0, \dots, 0] \tag{A6}$$

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \cdots & \phi_{r-1} & \phi_r \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & & 0 \\ \vdots & 0 & \cdots & \vdots & \vdots \\ \vdots & & \cdots & \vdots & \vdots \\ 0 & & \cdots & 1 & 0 \end{bmatrix} \tag{A7}$$

4.2. The filter-cleaner computes robust estimates of the vector X_t based on

$$\hat{Z}_t = \Phi \hat{Z}_{t-1} + \tilde{m}_t s_t \Psi\left(\frac{x_t - \hat{x}_t^{t-1}}{s_t}\right) \tag{A8}$$

where $\tilde{m}_t = m_t / s_t^2$, and $\tilde{m}_t$ is the first column of M_t

$$M_{t+1} = \Phi P_t \Phi^T + Q \tag{A9}$$

$$P_t = M_t - \pi\left(\frac{x_t - \hat{x}_t^{t-1}}{s_t}\right) \tag{A10}$$

where Q is a matrix with all zero entries except $Q_{11} = \sigma_\varepsilon^2$. $s_t^2 = m_{11,t}$.

$\hat{x}_t^{t-1}$ denotes a robust one-step ahead prediction of x_t and $\hat{x}_t^{t-1} = \left(\Phi \hat{Z}_{t-1}\right)_1$.

The psi-function,Ψ and weight function,π are chosen to be:

$$\Psi(\tau) = \begin{cases} \tau, & |\tau| < 3 \\ 0, & |\tau| \geq 3 \end{cases} \tag{A11}$$

$$\pi(\tau) = \frac{\Psi(\tau)}{\tau} \tag{A12}$$

4.3. Finally, the cleaned data at time t is given by:

$$\hat{z}_t = \left(\hat{Z}_t\right)_1 \tag{A13}$$

Manuscript received May 16, 2014, and revision received Aug. 20, 2014.