Supply Chain Monitoring Using Principal Component Analysis

Authors:

Jing Wang, Christopher Swartz, Brandon Corbett, Kai Huang

Date Submitted: 2020-07-16

Keywords: monitoring, Multivariate Statistics, Supply Chain

Abstract:

Various types of risks exist in a supply chain, and disruptions could lead to economic loss or even breakdown of a supply chain without an effective mitigation strategy. The ability to detect disruptions early can help improve the resilience of the supply chain. In this paper, the application of principal component analysis (PCA) and dynamic PCA (DPCA) in fault detection and diagnosis of a supply chain system is investigated. In order to monitor the supply chain, data such as inventory levels, market demands and amount of products in transit are collected. PCA and DPCA are used to model the normal operating conditions (NOC). Two monitoring statistics, the Hotelling's T-squared and the squared prediction error (SPE), are used to detect abnormal operation of the supply chain. The confidence limits of these two statistics are estimated from the training data based on the χ^2- distributions. The contribution plots are used to identify the variables with abnormal behavior when at least one statistic exceeds its limit. Two case studies are presented - a multi-echelon supply chain for single product that includes a manufacturing process, and a gas bottling supply chain with multiple products. In order to validate the proposed method, supply chain simulation models are developed using the programming language Python 3.7, and simulated data is collected for analysis. PCA and DPCA are applied to the data using the scikit-learn machine learning library for Python. The results show that abnormal operation due to transportation delay, supply shortage, and poor manufacturing yield can be detected. The contribution plots are useful for interpreting and identifying the abnormality.

Record Type: Preprint

Submitted To: LAPSE (Living Archive for Process Systems Engineering)

Citation (overall record, always the latest version):	LAPSE:2020.0822
Citation (this specific file, latest version):	LAPSE:2020.0822-1
Citation (this specific file, this version):	LAPSE:2020.0822-1v1

DOI of Published Version: https://doi.org/10.1021/acs.iecr.0c01038

Supply Chain Monitoring Using Principal Component Analysis

Jing Wang,[†] Christopher L.E. Swartz,[*,‡] Brandon Corbett,[¶] and Kai Huang[§]

†*School of Computational Science and Engineering, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada, L8S 4K1*

‡*Department of Chemical Engineering, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada, L8S 4L7*

¶*ProSensus Inc., 4325 Harvester Road, Unit 12, Burlington, Ontario, Canada, L7L 5M4*

§*DeGroote School of Business, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada, L8S 4M4*

E-mail: swartzc@mcmaster.ca

Abstract

Various types of risks exist in a supply chain, and disruptions could lead to economic loss or even breakdown of a supply chain without an effective mitigation strategy. The ability to detect disruptions early can help improve the resilience of the supply chain. In this paper, the application of principal component analysis (PCA) and dynamic PCA (DPCA) in fault detection and diagnosis of a supply chain system is investigated. In order to monitor the supply chain, data such as inventory levels, market demands and amount of products in transit are collected. PCA and DPCA are used to model the normal operating conditions (NOC). Two monitoring statistics, the Hotelling's T^2 and the squared prediction error (SPE), are used to detect abnormal operation

of the supply chain. The confidence limits of these two statistics are estimated from the training data based on the χ^2- distributions. The contribution plots are used to identify the variables with abnormal behavior when at least one statistic exceeds its limit. Two case studies are presented - a multi-echelon supply chain for single product that includes a manufacturing process, and a gas bottling supply chain with multiple products. In order to validate the proposed method, supply chain simulation models are developed using the programming language Python 3.7, and simulated data is collected for analysis. PCA and DPCA are applied to the data using the scikit-learn machine learning library for Python. The results show that abnormal operation due to transportation delay, supply shortage, and poor manufacturing yield can be detected. The contribution plots are useful for interpreting and identifying the abnormality.

1 Introduction

A supply chain is a network of suppliers, manufacturing facilities, warehouses, retailers, and customers that deals with the procurement of raw materials, the manufacture of products, and the distribution of products to the customers. There is material flow from upstream suppliers to downstream customers, and information flow in the opposite direction [1,2]. The importance of a well operating supply chain to the overall economic performance of an enterprise has underpinned a large body of research on supply chain operation and design across several disciplines. Reviews of supply chain studies from a process systems engineering perspective are given by Grossmann [3], Shah [4], and Papageorgiou [5].

There are various types of risks in supply chain operation, including delays, poor yield or quality at a supply source, procurement failures, inaccurate forecasts, uncertain consumer demands, and disruptions like natural disaster [6,7]. For example, in the case study of Carvalho et al. [8], the supply delay is regarded as the main disturbance that negatively affects the

automaker. Wilson [9] investigates the impact of transportation disruptions on supply chain performance. The disturbances may cause financial loss or even breakdown of the supply chain without an effective mitigation strategy [10]. Improving the ability to detect disturbances in a supply chain quickly can help reduce the risks and substantially increase the resilience of the supply chain [11,12]. The purpose of this paper is to investigate the application of data analytics to supply chain monitoring, fault detection and diagnosis.

Data analytics is becoming more and more important in the era of big data. Data analytics in supply chain management (SCM) has gained much attention in both academic research and industrial applications [13–15]. In SCM, data is increasingly employed to capture trends in costs and performance, monitor inventory, support process control and improve the process, and optimize production [16]. Supply chain analytics (SCA), defined by Souza [13] as the use of information and analytical tools for improved supply chain decision-making, can generally be categorized into three classes [13–15]: (1) descriptive analytics, which aims at explaining what has happened/is happening in the supply chain and why; (2) predictive analytics, which focuses on answering what will be happening or likely to happen in the supply chain; and (3) prescriptive analytics, which explores what should be happening and how to influence it. The relevant techniques involve statistics, programming, mathematical optimization, and simulation [15].

The applications of data analytics in SCM can also be classified by supply chain functions: procurement, manufacturing, logistics and transportation, warehousing, and demand management, or general SCM [17]. For example, in research on procurement, Jain et al. [18] investigate a data mining approach to discover the hidden relationships between data used for suppliers' selection and their overall rating based on prior performance. The approach helps in optimizing the selection process of suppliers. Mori et al. [19] utilize machine learning techniques like support vector machine and logistic regression to build a prediction model of customer-supplier relationships, and help find potential business partners. As for manufac-

turing, Zhong et al.[20] use radio frequency identification (RFID) production shop floor data to obtain better estimation of the arrival of customer orders and standard operation times. These parameters are then used to develop a two-level planning and scheduling model for RFID-enabled real-time ubiquitous shop floor manufacturing. In logistics and transportation, Zhao et al.[21] extract the upper and lower limits of uncertain parameters from historical data, and use them for the re-design of a green supply chain. Toole et al.[22] develop a system to estimate the travel demand and infrastructure usage for transportation planning, using massive data generated by mobile computing. Li et al.[23] employ Lasso Granger causality models to pick the most relevant data to build a traffic prediction model, thus achieving a good balance between model complexity and model performance. In terms of warehousing, Chiang et al.[24] define an association index and propose a data mining-based storage assignment approach, which improves the efficiency of order picking. Tsai and Huang[25] use data analytics to capture customer purchase and moving behavior, and optimize the shelf space allocation. As for demand management, Salehan and Kim[26] use a sentiment mining approach to investigate predictors of performance of online consumer reviews; Alain Yee et al.[27] employ a neural network to investigate which variables, such as online reviews, promotion strategies and sentiments, are important predictors of product sales. Ma et al.[28] propose a demand trend mining technique for data-driven product design.

A promising application of SCA is data-driven supply chain optimization under uncertainty, in which techniques of data analytics and machine learning are used to characterize the uncertainty based on process data in order to more accurately represent the uncertainty and reduce conservativeness in the optimization. Ning and You[29] present a framework for data-driven adaptive robust optimization with case studies that include robust planning of chemical process networks under uncertainty, and a data-driven stochastic robust optimization framework is proposed by Ning and You[30]. Shang and You[31] develop a formulation for distributionally robust optimization under uncertainty, which seeks to hedge against inexactness of the probability distribution of the uncertainty. In Gao et al.[32], the scheme is

extended and applied to robust optimization of shale gas supply chains under uncertainty. A recent review of data-driven optimization under uncertainty with perspectives on future research directions is given in Ning and You [33].

The various applications of SCA as described above indicate the power and potential for data analytics in aiding decision-making in many aspects of SCM. This paper focuses on a multivariate statistical method, principal component analysis (PCA). PCA is designed for extracting uncorrelated components from correlated data, as described by Wold et al. [34]. It is known as a latent variable method (LVM), and is also considered as a type of machine learning method [35,36]. PCA has been successfully applied to industrial process modeling, monitoring and diagnosis [37–42]. Extensions of PCA have been developed for industrial process analysis, including dynamic PCA [43–48], kernel/nonlinear PCA [49], recursive PCA, multi-block, multi-way PCA [41], dynamic inner PCA [50], and mixtures of probabilistic PCA [51].

In the current literature, PCA has been applied in some aspects of SCA. In Pozo et al. [52], PCA is employed to reduce the computational complexity of a multi-objective optimization problem formulated for supply chain design. Redundant metrics are identified and omitted while retaining the main features of the problem. Lei and Moon [53] use PCA to help determine market segments for new products, and develop a decision support system for market-driven product positioning and design. How and Lam [54] use PCA to reduce the redundancies of performance indicators, thus aiding the multi-objective optimization of a supply chain. In data-driven supply chain optimization, PCA is applied to help characterize uncertain parameters in a supply chain by reducing the dimensionality of the correlated uncertainty data [32,55]. Mele et al. [56] demonstrate the application of PCA techniques for detection of manufacturing and transportation delays in a simulated supply chain system. Their study includes a wavelet based multi-scale PCA technique and a Genetic Algorithm based search scheme to account for time delays.

The above applications demonstrate some of the benefits of using PCA in SCA. However, research on applying PCA to monitoring, fault detection and diagnosis of a supply chain as a system is still lacking. In SCA applications of PCA, the monitoring statistics of the scores and residual are largely ignored and the ability of PCA in statistical monitoring has not been fully taken advantage of yet, in contrast to applications of statistical process monitoring using PCA in plant-wide processes such as manufacturing.

The objective of this paper is to investigate the use of PCA in supply chain monitoring, through keeping track of the operating status of a supply chain as a system. Since there are usually many measurements in a supply chain, monitoring all of them individually could be difficult to implement and inefficient. Supply chains comprise a collection of entities interlinked through material and information flow, leading to correlation in supply chain data. Hence analogous to process monitoring, PCA can potentially be employed to model and monitor the operation status of a supply chain, and detect variation from the normal supply chain operating condition (NOC). Within the context of SCA, this application can be classified as descriptive analytics. In order to validate the effectiveness of PCA in supply chain monitoring, simulation is carried out to generate supply chain data, with PCA and dynamic PCA (DPCA) performed on the simulated data.

PCA and DPCA are used in this study since they represent basic forms of a broader class of latent variable methods, and would thus be good indicators of the potential of these methods for supply chain monitoring applications. We note that canonical variate analysis (CVA), for example, has been recognized for some time as an effective basis for fault detection, isolation and diagnosis of dynamic systems[45,57–59]. While this technique could, in principle, be applied to supply chain systems written in state-space form, we limit our focus in this introductory study to the fundamental PCA and DPCA approaches.

The remainder of this paper is organized as follows: Section 2 provides a summary of fault

detection and diagnosis using PCA and DPCA. Section 3 describes the supply chain simulation model developed in this study, which is employed to generate simulated data for analysis. In Section 4, two case studies are introduced, and used to validate the proposed supply chain monitoring method. Conclusions are presented in Section 5.

2 Principal component analysis

A brief introduction of PCA and fault detection and analysis (FDD) using PCA is given in Sections 2.1–2.3. Dynamic PCA is described in Section 2.4. The description of supply chain monitoring using PCA/DPCA is presented in Section 2.5. The formulas are extracted from Kresta et al.[37], Kourti and MacGregor[60], Qin[41], Li et al.[48], as well as other references where stated.

2.1 Principal component analysis

PCA is designed for extracting uncorrelated components from correlated data. Denote the data collected at time i as a K-dimensional vector $\mathbf{x}_i$, and the data collected over N time periods as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$, which contains N samples. Assume that $\mathbf{X}$ has been scaled to zero-mean and unit-variance, then performing PCA on $\mathbf{X}$ corresponds to construction of the following relationships:

$$\mathbf{T} = \mathbf{XP} \tag{1a}$$

$$\mathbf{X} = \mathbf{TP}^T + \mathbf{E} = \sum_{a=1}^{A} \mathbf{t}_a \mathbf{p}_a^T + \mathbf{E} \tag{1b}$$

where $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_A]$ is the loading matrix; $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_A]$ is the score matrix, and $\mathbf{t}_a = \mathbf{Xp}_a$, for $a = 1, \ldots, A$, is the sample data of the a-th principal component (PC); and

$\mathbf{E}$ is the residual. The loadings are determined by maximizing the variance of the scores. Usually $A < K$ to achieve dimensionality reduction.

PCA can be implemented through an eigen-decomposition on the sample covariance matrix $\mathbf{S} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}$, or a singular value decomposition (SVD) on $\mathbf{X}$. Alternatively, the PCs can be efficiently extracted by the nonlinear iterative partial least squares (NIPALS) algorithm, which is actually a variant of the Power method [34].

2.2 Fault detection

For fault detection, first, a PCA model is built to characterize the normal operating conditions (NOC) from normal operating data. A sample $\mathbf{x}_i$ can be projected to the principal component subspace (PCS) and the residual subspace (RS) respectively:

$$\mathbf{t}_i^T = \mathbf{x}_i^T\mathbf{P} \tag{2a}$$

$$\tilde{\mathbf{x}}_i^T = \mathbf{x}_i^T - \mathbf{t}_i^T\mathbf{P}^T = \mathbf{x}_i^T(\mathbf{I} - \mathbf{P}\mathbf{P}^T) \tag{2b}$$

Qin [61] gives a comprehensive review of fault detection indices in statistical process monitoring. In PCA, the Hotelling's T^2 and the squared prediction error (SPE) are widely employed to detect variations from the NOC. The Hotelling's T^2, also known as the D-statistic, of $\mathbf{x}_i$ is defined as:

$$T^2(\mathbf{x}_i) = \mathbf{t}_i^T\mathbf{\Lambda}^{-1}\mathbf{t}_i = \mathbf{x}_i^T\mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T\mathbf{x}_i = \sum_{a=1}^{A}\frac{t_{ia}^2}{\lambda_a} \tag{3}$$

where $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_\mathrm{A}\}$ contains the eigenvalues of $\mathbf{S}$ in descending order, and t_{ia} is the a-th component of $\mathbf{t}_i$.

Even when the data are not multivariate normally distributed, the principal components are approximately independently normally distributed when the number of variables is large,

according to the central limit theorem[61,62]. When N is large, the distribution of T^2 can be approximated by a χ^2-distribution, and the confidence limit at a significance level α can be calculated as:

$$T_\alpha^2 = \chi_\alpha^2(A) \tag{4}$$

Alternatively, the confidence limit can also be estimated by the F-distribution[38] or the Beta-distribution[63]. The significance level α can be determined based on the users' needs[64].

The SPE, also referred to as the Q-statistic, of $\mathbf{x}_i$ is defined as:

$$\text{SPE}(\mathbf{x}_i) = \|\tilde{\mathbf{x}}_i\|_2^2 = \|(\mathbf{I} - \mathbf{P}\mathbf{P}^T)\mathbf{x}_i\|_2^2 = \mathbf{x}_i^T\tilde{\mathbf{P}}\tilde{\mathbf{P}}^T\mathbf{x}_i \tag{5}$$

where $\tilde{\mathbf{P}} = [\mathbf{p}_{A+1}, \ldots, \mathbf{p}_K]$ comprises the $(K-A)$ eigenvectors that are not retained. The confidence limit of SPE at a significance level α can be estimated by a weighted χ^2-distribution[40]:

$$\text{SPE}_\alpha = \frac{v}{2m}\chi_\alpha^2\left(\frac{2m^2}{v}\right) \tag{6}$$

where $m = mean(\text{SPE}), v = var(\text{SPE})$, denote the mean and variance of the SPE, respectively. The approximating distribution using eq 6 works well in practice even when the errors are not normal[41,62].

The T^2 and SPE model the variation in the PCS and RS, respectively. They can be used as a pair to indicate significant deviation from the NOC[63]. Their roles in process monitoring are not symmetric. Exceeding the T^2 limit does not necessarily indicate a fault, while probably a shift in the operation region; thus SPE is considered preferable over T^2 for fault detection[61]. In this paper, the Hotelling' T^2 and SPE are adopted for supply chain monitoring, with greater emphasis on the SPE, since it is more reliable for non-normal data[41].

2.3 Fault diagnosis

Fault detection tells whether there is an abnormality or not. If a fault is detected, the next step is to identify potential fault-related variables with the goal of determining the source cause of the abnormality. A widely used approach for fault identification is the contribution plot [38,65]. The variables with large contributions to the T^2 and SPE are considered most likely to be fault related. For a sample $\mathbf{x}_i$, the contribution of the j-th variable x_{ij} to score T_a can be defined as follows [66]:

$$contribution_{a,j} = x_{ij} p_{aj} \tag{7}$$

where p_{aj} is the j-th component of the a-th loading $\mathbf{p}_a$, while the contribution of x_{ij} to the SPE can be defined as:

$$contribution_j = \tilde{x}_{ij}^2 \tag{8}$$

where $\tilde{x}_{ij}$ is the j-th component of the residual $\tilde{\mathbf{x}}_i$.

We remark that use of the contribution plot to identify potential fault-related variables may be viewed as an indirect approach to fault diagnosis [63], thus is commonly categorized as such. By contrast, Chiang et al. [45] refer to this approach as fault identification, and reserve fault diagnosis to describe methods that seek to identify specific faults, such as Fisher discriminant analysis (FDA).

Other methods developed for fault identification and/or diagnosis in process monitoring include the fault signature [67], hierarchical contribution plots [61], causal map combined with data-driven approach [68], trajectory loading and score contribution plots [69], reconstruction based contribution (RBC) [70], and the framework integrating dynamic PCA, RBC and Granger causality analysis [48].

The contribution plots have been demonstrated to be very useful in many applications [67].

They greatly help narrow the scope of potential fault-related variables. Therefore, in this paper, the contribution plots for the score (eq 7) and SPE (eq 8) are adopted for fault identification. The trajectory contribution plot is also employed.

2.4 Dynamic PCA

The standard (static) PCA deals with correlated data that are time independent. When the observations are time series, the auto-correlation and cross-correlation can be taken into account by dynamic PCA (DPCA). DPCA uses the time lag shift technique to enable the PCA model to capture dynamic behavior of a system[43]. In DPCA, the observation at time k is augmented by the l previous observations:

$$\mathbf{z}_k = [\mathbf{x}_k^T, \mathbf{x}_{k-1}^T, \ldots, \mathbf{x}_{k-l}^T]^T \tag{9}$$

where l is the number of time lags.

DPCA then corresponds to PCA implemented on the following augmented matrix $\mathbf{Z}$[48]:

$$\mathbf{Z} = [\mathbf{z}_{l+1}, \mathbf{z}_{l+2}, \ldots, \mathbf{z}_N]^T = \begin{bmatrix} \mathbf{x}_{l+1} & \mathbf{x}_{l+2} & \ldots & \mathbf{x}_N \\ \mathbf{x}_l & \mathbf{x}_{l+1} & \ldots & \mathbf{x}_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_{N-l} \end{bmatrix}^T \tag{10}$$

The augmented matrix can be equivalently expressed as $\mathbf{Z} = [\mathbf{X}_{l+1}, \mathbf{X}_l, \ldots, \mathbf{X}_1]$[43], where $\mathbf{X}_k = [\mathbf{x}_k, \mathbf{x}_{k+1}, \ldots, \mathbf{x}_{k+N-l-1}]^T$.

In DPCA models, the time lag needs to be taken into account when calculating the variable contributions. The contribution of one variable to T_a and SPE is summed over the lags[71],

as shown in eq 11.

$$Contribution_j = contribution_{j,t} + contribution_{j,(t-1)} + \cdots + contribution_{j,(t-l)} \tag{11}$$

2.5 Supply chain monitoring using PCA/DPCA

The structure of a supply chain is shown in Figure 1. Generally, a supply chain consists of suppliers, manufacturers, warehouses (distribution centers), retailers, and consumers (customers). At each agent, different inventory management policies can be adopted [72]. There are various types of flows in a supply chain: material flow, product flow, and service flow which are from upstream to downstream; with information flow and cash flow from downstream to upstream [73,74]. In this paper, the material flow from upstream to downstream, and the order information flow in the opposite direction are considered.

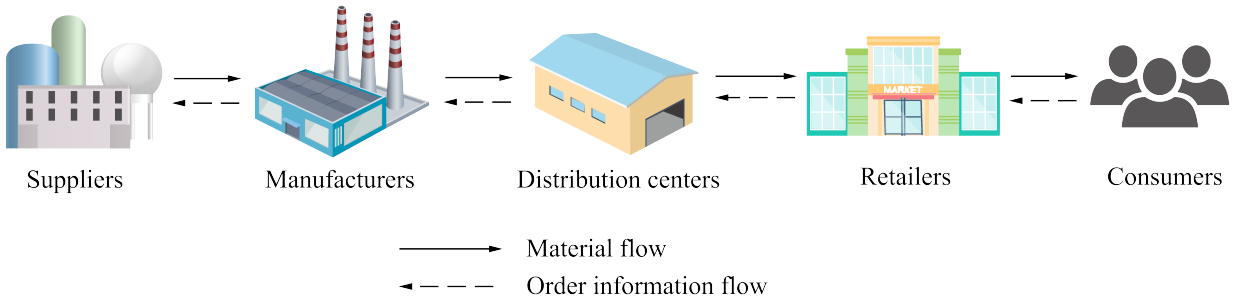


Figure 1: The structure of a supply chain.

Supply chains can be viewed and modeled as dynamic systems, and as such respond to disturbances and inputs in an analogous way to process plants, and admit control techniques such as model predictive control (MPC) that is widely applied in a process plant setting [2,75,76]. From this perspective, the 'process variables' of a supply chain, such as the time-varying market demands and inventory levels, are correlated. Assuming that the information of all echelons and agents can be obtained, the correlation of data indicates the potential for PCA in dimensionality reduction and subsequent monitoring.

The procedure adopted for using PCA and DPCA for supply chain monitoring is presented as follows:

1. Training step:

(1) Collect NOC data of the supply chain, such as inventory levels, market demands, and the amount of products in transit;

(2) Preprocessing: augment data for DPCA; normalize the data to have zero mean and unit variation for each variable;

(3) Perform PCA/DPCA, obtain the loadings and scores;

(4) Calculate the monitoring statistics (T^2 and SPE) of the NOC data;

(5) Determine the confidence limits of T^2 and SPE[41,61,64].

2. Monitoring step:

(1) For new data, normalize it with the mean and standard deviation of each variable from the training step;

(2) Project the new data into the PCS and RS to get scores and residuals;

(3) Calculate the T^2 and SPE of the new data;

(4) Check whether T^2 and SPE are both within the confidence limits; if so, then the data can be seen as normal; otherwise, there is an abnormal event, check the contribution plots to identify the fault-related variables.

3 Supply Chain Simulation

In order to validate the effectiveness of PCA and DPCA in supply chain monitoring, a simulation model of a supply chain is developed and the simulated data is collected for

analysis. This section gives a detailed description of the simulation model developed in this paper.

3.1 The role and basics of supply chain simulation

Usually the optimal design of a supply chain network under uncertainty can be formulated as a two-stage stochastic program [77], or a robust/adaptive optimization problem with more stages [33]. Solving the optimization problem will produce an optimal supply chain configuration. However, it cannot tell what will happen after the supply chain design is implemented under arbitrary scenarios [73]. In comparison, simulation can help understand and evaluate the supply chain under different scenarios.

For example, inventory policies, such as (R, Q) and (s, S) policies with periodic or continuous review, might be used for the inventory management in a multi-echelon supply chain in reality [72]. The (R, Q) policy requires that an order with lotsize Q be placed to the upstream when the inventory position falls below the re-order point R. The (s, S) policy is to maintain a target inventory level S when the inventory position falls below the re-order point s. Simulation can help analyze the performance of a supply chain over a time period when using different inventory policies, even with a time-varying re-order point. Wilson [9] uses simulation to study how transportation disruptions affect the performance of supply chain. Carvalho et al. [8] use simulation to investigate different mitigation strategies when disruptions occur in the supply chain.

In order to validate the proposed supply chain monitoring method, this paper uses simulation to generate supply chain data for analysis.

3.2 Supply chain simulation using Python

There are some softwares designed for supply chain simulation, such as Arena combined with Microsoft Excel[74], AnyLogic[78], and Supply Chain Analyzer[73]. In this paper, the supply chain simulation is implemented using the open-source programming language Python 3.7.

The simulation model is developed using discrete-continuous combined modeling as presented by Lee et al.[73], where the supply chain elements can be classified into two groups, continuous and discrete. Inventory levels, order information and customer demands are considered as continuous elements, while transportation between agents is considered as a discrete element. Object-oriented programming is implemented, which makes the simulation model flexible to be customized for different supply chains. The Python simulation model can be categorized into two parts: (1) The Python modules where the classes for the supply chain participants are defined. Each type of participant is defined as a class, while some participants share some common attributes and methods. These classes can be debugged individually first before the systematic simulation. (2) The 'main' Python module where the classes are imported, the supply chain structure is defined, and the simulation is run. For example, the attributes and methods defined for the 'Warehouse' class is shown in Table 1.

Figure 2 shows the causal loop diagram of the supply chain simulation model when all the participants adopt the (s, S) inventory policy. The 'inv' is used as the attribute name for inventory level. The 'Retailer' class satisfies the orders from the 'Customer', and places orders to the 'Warehouse'. Similarly, the 'Warehouse' orders from the 'Factory' and satisfies the orders from the 'Retailer'. The 'Factory' orders raw materials from the 'Supplier', manufactures the product and delivers it to the 'Warehouse'. The 'Supplier' has raw material in stock while no product. A waiting line (with an attribute name 'waitline') of a participant is a list of its customers. It uses the first in, first out (FIFO) method, which means that the first order is served first.

Table 1: Attributes and methods defined for the 'Warehouse' class

Attributes	name	string, the name of this Warehouse
	total_periods	NumPy array, the time periods for which the simulation is run
	products	list, contains the names of all products stored at this Warehouse
	product	dictionary, created for each of the product in the 'products' list, containing information (key-value pairs) including inventory level, demand, inventory policy, re-order point, target inventory level, lot-size, order size, the amount of product in transit, arrival time of order, etc.
Methods	__init__()	initialize the attributes of the class
	update()	update the inventory profiles after satisfying downstream orders and receiving upstream shipments
	review()	review the inventory level, if the inventory level is below the re-order point, place an order to upstream according to the inventory policy
	collect_data()	collect data and save it as a 2-dimensional numpy array, write it into a .txt file, or comma-separate values (.csv) file
	plot()	plot the profiles

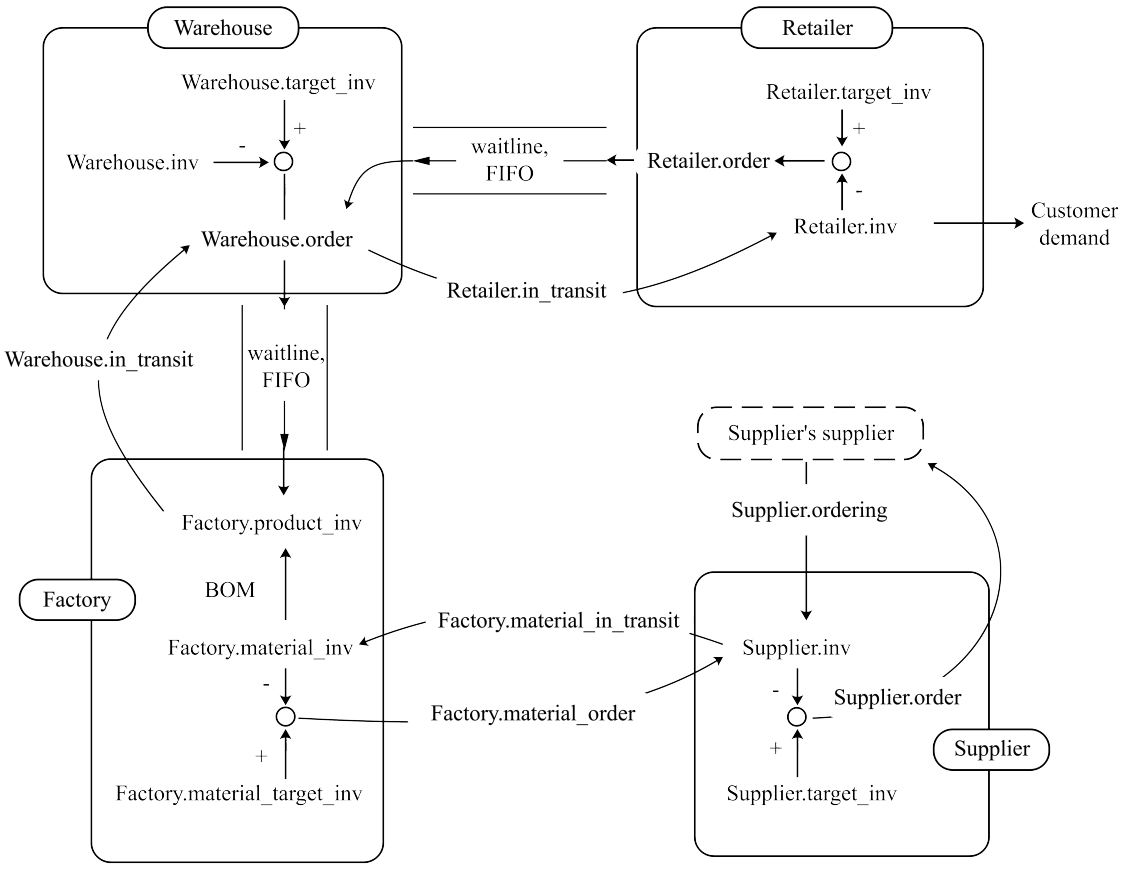


Figure 2: The causal loop diagram of the supply chain simulation model.

It is assumed that an agent does not place another order if there is an outstanding order which has not arrived yet. Further demands are satisfied using available stocks. The methods update() and review() for the participants are given as follows, which contain the material balance equations and show how the class attributes are updated at time period i.

Method: Retailer.update(i):

```
Retailer.inv[i] = Retailer.inv[i-1] - Customer.demand[i]
if Retailer.order_ready == True then
    Retailer.in_transit = Retailer.ordersize
    Retailer.arrival_time = i + transportation_delay
    Retailer.order_ready = False
end if
if i == Retailer.arrival_time then
    Retailer.inv[i] = Retailer.inv[i] + Retailer.in_transit
    Retailer.in_transit = 0
    Retailer.outstand_order = False
end if
```

Method: Retailer.review(i):

```
if Retailer.inv[i]≤ Retailer.reorder_point and Retailer.outstand_order == False then
    (s, S) policy: Retailer.ordersize = Retailer.target_inv - Retailer.inv[i]
    (r, Q) policy: Retailer.ordersize = Retailer.lotsize
    Retailer.outstand_order = True
end if
```

The attribute 'inv' indicates the inventory level, which is defined by eq 12, in the same way as defined by Axsäter [72]:

$$\text{inventory level} = \text{stock on hand} - \text{backorders} \tag{12}$$

The backorders are the orders that have been placed but cannot be fulfilled yet due to a shortage. For the Retailer, negative 'inv' means backorders. The Warehouse has a waiting line for its customers, and the orders are satisfied in a FIFO method. If the inventory is not sufficient, then the rest of the orders are recorded as backorders. The backorders will be satisfied once sufficient inventory is available. The method Warehouse.update() is given below. The review and replenishment of inventory for the Warehouse are similar to those of the Retailer, and hence they are not described here. The attribute 'stock' indicates the stock on hand.

Method: Warehouse.update(i):

while len(Warehouse.waitline) > 0 **do**
  if Warehouse.stock[i] ≥ Warehouse.waitline[0].ordersize **then**
    Warehouse.stock[i] = Warehouse.stock[i-1] - Warehouse.waitline[0].ordersize
    Warehouse.waitline[0].order_ready = True
    Warehouse.waitline.pop(0)
  else
    break
  end if
end while
Warehouse.backorder[i] = $\sum_{\text{Retailer in waitline}}$ Retailer.ordersize
Warehouse.inv[i] = Warehouse.stock[i] - Warehouse.backorder[i]

The Factory uses a 'make-to-order' production system. It starts to manufacture only when an order is received. It has raw materials in stock, while no excess product in stock. The waiting line for the Factory is similar to that of the Warehouse, and the review and replenishment of raw materials for the Factory are also similar to those of the Retailer. The Supplier is similar to the Warehouse; it is seen as the most upstream and provides raw materials. The material balance equations for manufacturing at the Factory are given in the following Factory.update() method, where Factory.material_BOM is the mass balance coefficient of material.

Method: Factory.update(i):

for each material **do**
  Factory.material_inv[i] = Factory.material_inv[i-1] - Factory.production[i] × Factory.material_BOM
end for
Factory.product_inv[i] = Factory.product_inv[i-1] + Factory.production[i]

The Python libraries used for developing the simulation model and analyzing data are listed in Table 2. No other third-party library is used in the simulation.

Table 2: Python libraries used in simulation and analysis

Python library	Purpose
numpy	some profiles such as inventory levels are stored as numpy arrays; multivariate normal distribution
scipy	the χ^2- distribution (scipy.stats.chi2) is used to calculate the confidence limits of T^2 and SPE
matplotlib.plot	to plot the profiles
pandas	to store data as dataframes, which can then be saved as txt or csv files
sci-kit learn (sklearn)	sklearn.preprocessing.StandScaler for preprocessing data, and sklearn.decomposition.PCA for implementing PCA

For a supply chain shown in Figure 1, the pseudo-code of the simulation is presented in the following procedure. The simulation starts from the most downstream and proceeds to the most upstream echelon by echelon. First, the demands of the Customers in each time period are generated, for example, as constant values or randomly from statistical distributions, like multivariate Gaussian. Then the Retailers satisfy the demands of Customers, review their inventories and place orders to the Warehouse according to the inventory policy. The Retailers are appended to the waiting lines of the Warehouses. The Warehouses satisfy the orders of Retailers in the waiting lines in a FIFO way, and place orders to the Factories. Once a Retailer's orders are satisfied, it is moved out from the waiting line. The Factories update the attributes in a similar way and manufacture products according to orders from Warehouses. After that the Suppliers update.

Procedure: Supply chain simulation

```
import Customer, Retailer, Warehouse, Factory, Supplier
initialize the supply chain participants
generate Customer.demand
for i = 1 to total_periods do
  for each Retailer do
    Retailer.demand[i] ← ∑_downstream Customer.demand[i]
    Retailer.update(i)
    Retailer.review(i)
  end for
  for each Warehouse do
    for each downstream Retailer of Warehouse do
      Warehouse.waitline.append(Retailer)
    end for
    Warehouse.update(i)
    Warehouse.review(i)
  end for
  for each Factory do
    for each downstream Warehouse of Factory do
      Factory.waitline.append(Warehouse)
    end for
    Factory.update(i)
    Factory.review(i)
  end for
  for each Supplier do
    for each downstream Factory of Supplier do
      Supplier.waitline.append(Factory)
    end for
    Supplier.update(i)
    Supplier.review(i)
  end for
end for
```

3.3 PCA and DPCA using Python

The data analysis can be conducted in Python. In this paper, PCA is implemented using scikit-learn [79]. Also known as sklearn, scikit-learn is a machine learning library for Python, which provides various machine learning algorithms. The application program interface (API) sklearn.preprocessing.StandScaler is used to preprocess the collected supply chain

data, and then sklearn.decomposition.PCA is used to implement PCA on the preprocessed data. As for DPCA, there is no off-the-shelf API in scikit-learn, therefore it is coded in Python. The supply chain data is augmented with previous time lags, and then PCA is implemented on the augmented data by sklearn.decomposition.PCA.

Also, sklearn.decomposition.PCA does not provide methods for calculating the Hotelling's T^2 and SPE, nor the variable contributions. Therefore, the methods for calculating the two statistics and plotting the variable contributions are coded in Python. The PCA model containing all these methods is coded as a class.

4 Case studies

4.1 Case study 1: A multi-echelon supply chain with manufacturing process

4.1.1 Simulation

In the first case study, a multi-echelon supply chain example with a manufacturing process for a single product is investigated. The structure of this supply chain is shown in Figure 3. It consists of 2 suppliers, 1 factory, 1 warehouse, 3 retailers and their customers. At the most upstream are the suppliers, and the most downstream are the customers. The Supplier1 and Supplier2 provide raw materials M_1 and M_2, respectively. The final product A is made from M_1 and M_2 at the Factory, and the production scheme is $0.5M_1 + M_2 \rightarrow A$.

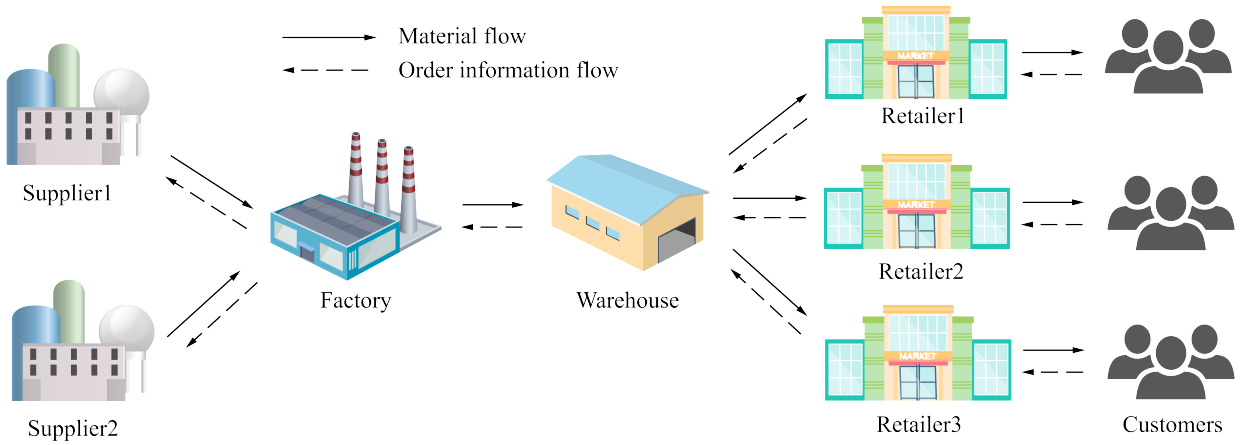


Figure 3: A multi-echelon supply chain with manufacturing process.

The supply chain network in this case is similar to that of the famous 'Beer Distribution Game'[80], but more complicated in some aspects. The beer game is designed for role-playing simulation of a production and distribution system. They both have 4 echelons. The supply chain in the beer game is a serial system[72], having one agent for each echelon. In Case 1, the wholesaler is not included, while multiple retailers and suppliers are included, which makes the supply chain network more complex.

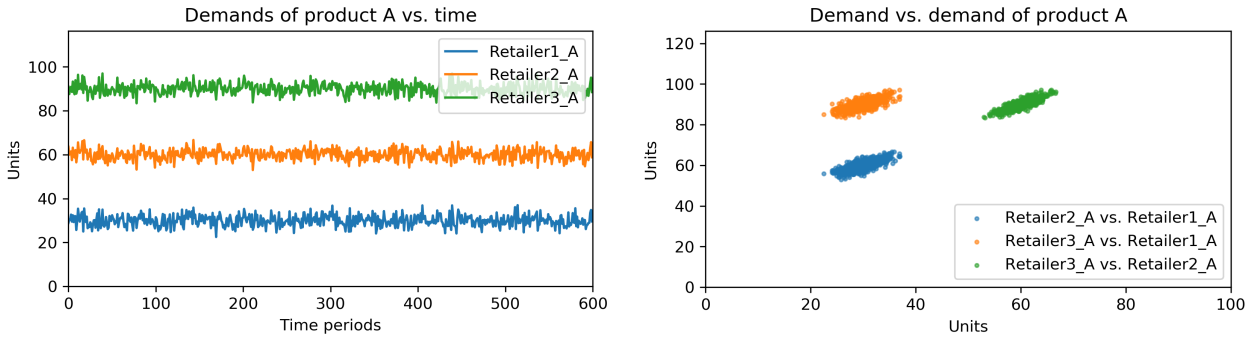
It is assumed that all the participants adopt the (s, S) inventory policy with continuous review, which means that the inventory is reviewed in every period. The demands of Customers are generated from a multivariate Gaussian distribution[29,81]. The mean vector of the Gaussian distribution is $[30, 60, 90]^T$, as given in Table 3, and the covariance matrix is a randomly generated positive semi-definite matrix: $[6.23, 4.37, 4.47; 4.37, 5.62, 4.95; 4.47, 4.95, 6.18]$. For each time period, the Retailers satisfy the demand of Customers if sufficient inventory is available, otherwise backorders are recorded; then they review the inventory and place an order to the Warehouse for replenishment. The Warehouse orders products from the Factory. The Factory follows the 'make-to-order' policy, which means that only when there is an incoming order from the Warehouse, it starts to manufacture the product A. It is assumed that the Factory manufactures at a fixed production rate, 400 units per time period, until

the order is fulfilled. It has raw materials in stock, but no excess product in stock. The Factory orders raw material M_1 from Supplier1 and M_2 from Supplier2. The parameters of the supply chain, such as the re-order points and target inventory levels, are listed in Table 3. The time interval of simulation is one time period, and 600 time periods are simulated. The normal transportation time between two agents is set as one time period.

Table 3: Parameters of supply chain

Participant	demand of A (mean)	initial stocks (units)			target stocks (units)			re-order points (units)		
		A	M_1	M_2	A	M_1	M_2	A	M_1	M_2
Retailer1	30	300	–	–	300	–	–	100	–	–
Retailer2	60	400	–	–	500	–	–	180	–	–
Retailer3	90	400	–	–	800	–	–	200	–	–
Warehouse	–	4000	–	–	5000	–	–	2500	–	–
Factory	–	0	1000	2000	–	1500	3000	–	600	1200
Supplier1	–	–	4000	–	–	4000	–	–	2000	–
Supplier2	–	–	–	6000	–	–	8000	–	–	2500

The demands of Customers at the 3 Retailers over the simulated time periods are shown in Figure 4a. It can be seen the demands fluctuate with time. Figure 4b shows the scatter plot of each pair of the demands, which indicates the assumed positive correlation between the demands of Customers. When the demand at one Retailer is high or low, the demands at the other two Retailers also tend to be high or low. The simulated supply chain data are collected, and the inventory profiles of the agents are shown in Figure 5a. A total of 21 variables are collected, including the orders received at the Retailers and the Warehouse (4 variables), and the inventory levels of A, M_1, and M_2 of the agents (9 variables), and the amount of products in transit or in processing (8 variables).



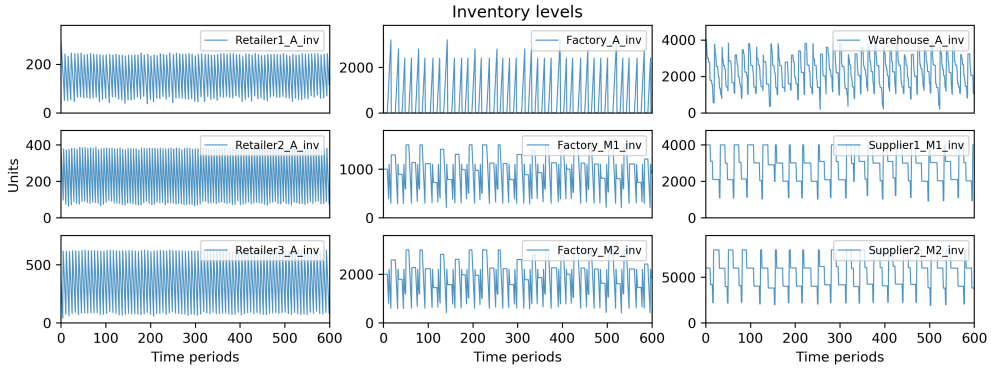
(a) Demands over the simulated time periods (b) Demand versus demand for Retailer pairs

Figure 4: Case 1: Demands at the 3 Retailers.

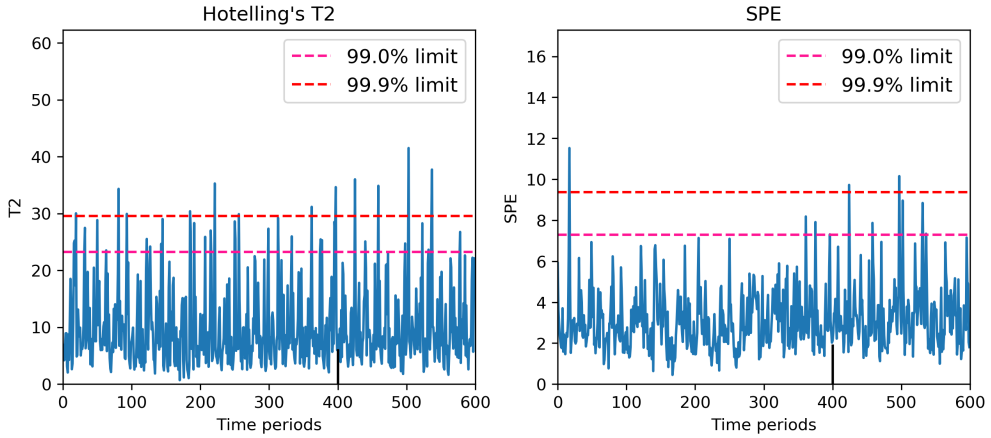
The supply chain operation over 600 time periods is simulated and analyzed. Each time period can be seen as 1 day. The NOC data from time period 0 to 400 is taken as the training set, which is a matrix of 400 rows and 21 columns, denoted as $\mathbf{X}_{train}^{raw}$. For PCA, $\mathbf{X}_{train}^{raw}$ is mean centred and scaled to unit variance to get $\mathbf{X}_{train}$. Then PCA is performed on $\mathbf{X}_{train}$. With PCA, 10 PCs are retained to achieve a R^2 value (the ratio of variance explained) of 85%. For DPCA, the matrix $\mathbf{X}_{train}^{raw}$ is augmented with 2 time lags to get $\mathbf{X}_{train,aug}^{raw}$, which has $21 \times 3 = 63$ columns. It is then preprocessed to get $\mathbf{X}_{train,aug}$, and 18 PCs are extracted to achieve a R^2 value of 82%. A testing set $\mathbf{X}_{test}^{raw}$ from time period 401 to 600 is compiled, and also comprises NOC data. $\mathbf{X}_{test}^{raw}$ is preprocessed to get $\mathbf{X}_{test}$, using the means and standard deviations obtained from the training set. Then $\mathbf{X}_{test}$ is projected into the latent space using the same transformation as that implemented on $\mathbf{X}_{train}$.

The monitoring charts for the Hotelling's T^2 and SPE using PCA and DPCA are shown in Figure 5b and Figure 5c, respectively. The confidence limits are estimated from the training set and plotted as dashed lines. Since the supply chain data are not all normally distributed, in order to reduce the false alarm rates, the significance levels in eq 4 and eq 6 are tuned to 99% and 99.9%. Exceeding the 99% limit raises a warning that the supply chain could be behaving abnormally, while exceeding the 99.9% limit raises an alarm that it is very likely that some fault has occurred in the supply chain. It is observed that a small portion of NOC

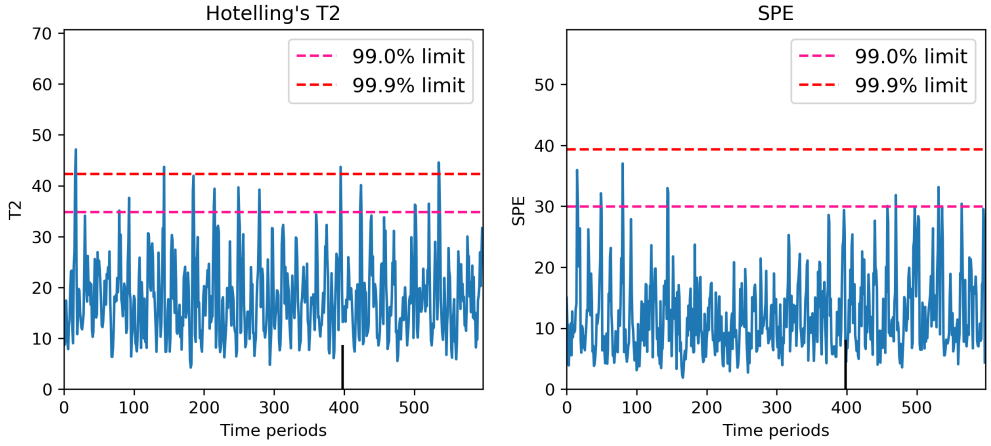
samples slightly violate the confidence limits, which are the false alarms. False alarms are not uncommon in fault detection, as in the research of Russell et al.[44] and Dong and Qin[50], for example. DPCA shows lower false alarm rate than PCA, which could be potentially due to the significant dynamics in the supply chain data, as can be seen in Figure 5a. This is consistent with Ku et al.[43], who show in their application studies that DPCA outperforms static PCA for dynamic systems. In general, the statistics of the testing set by DPCA are below the limits, which means that no unexpected event is detected and the testing data can be seen as normal.



(a) Inventory levels



(b) PCA



(c) DPCA

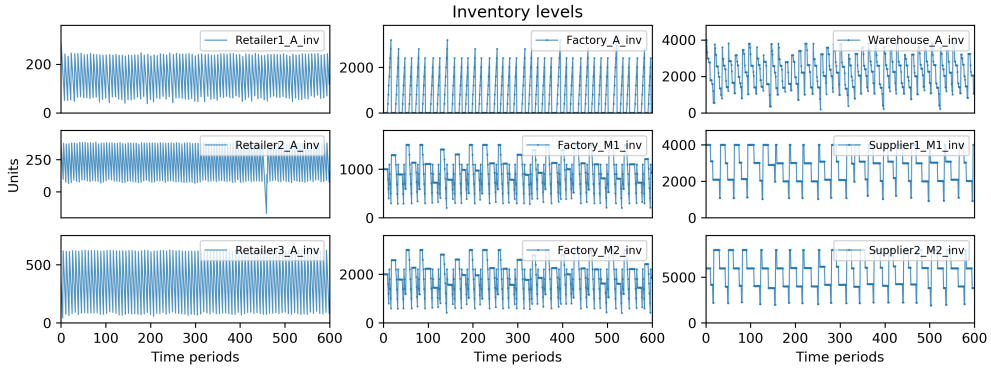
Figure 5: Case 1, NOC

4.1.2 FDD using PCA and DPCA

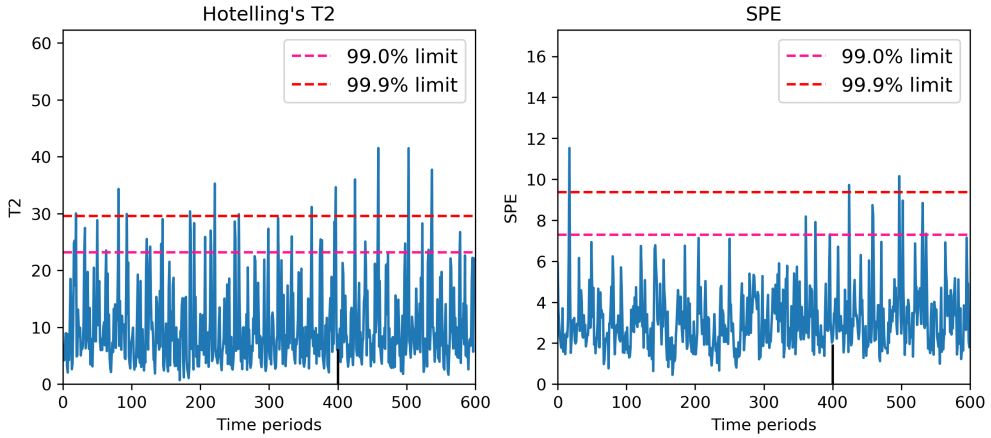
Three fault scenarios are simulated and analyzed using PCA and DPCA.

Scenario 1: The first fault scenario is a transportation delay between two echelons, which could lead to variations in the inventory levels of the affected participants. Suppose the order placed by the Retailer2 to the Warehouse at time period 454 is delayed, and the transportation time increases from 1 to 5 time periods. The simulated inventory profiles of the agents are shown in Figure 6a. The inventory level of the Retailer2 becomes low and a backorder situation occurs due to the delay.

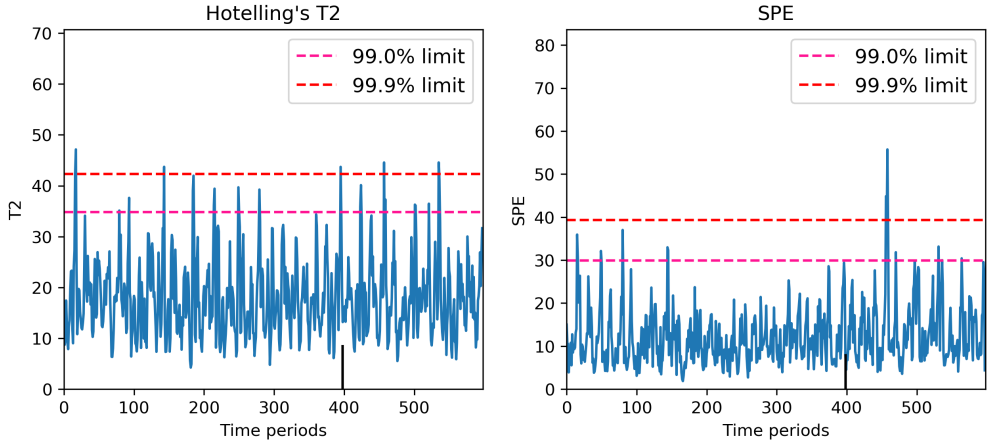
The monitoring charts using PCA and DPCA are shown in Figure 6b and Figure 6c, respectively. It can be seen that when the delay occurs, the SPE by DPCA exceeds its limits, which means that a large variation from the NOC is detected. In comparison, the SPE by PCA is not able to detect the fault. In order to identify the fault-related variables, the SPE contribution plot of a fault sample detected by DPCA is shown in Figure 7a. It can be seen that the variables related to Retailer2's inventory contribute the most to SPE, which implies that they are most likely to be fault-related, and some unexpected event might have happened at the Retailer2. The trajectory SPE contribution plot of these two abnormal variables is given in Figure 7b. From the trajectory, the SPE contributions of the 2 variables increase abnormally during the period of the fault. This helps narrow the scope of the root cause diagnosis. The SPE falls below the limits after the Retailer2 returns to NOC.



(a) Inventory levels

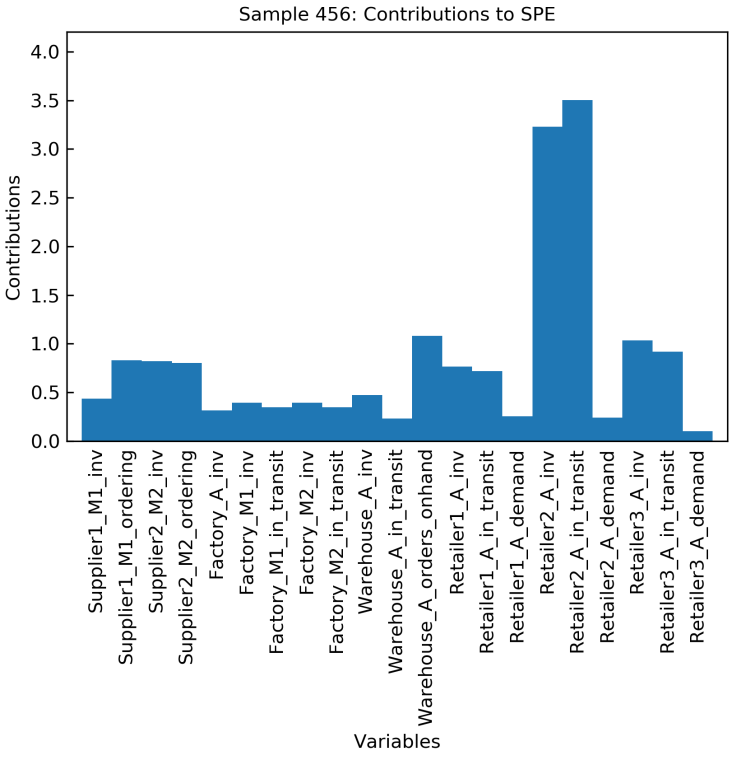


(b) PCA

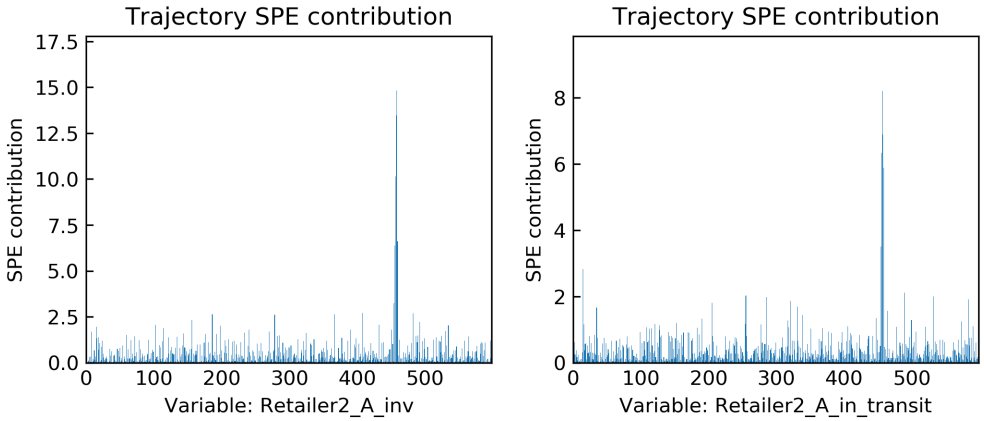


(c) DPCA

Figure 6: Case 1, scenario 1: transportation delay



(a) Sample SPE contribution



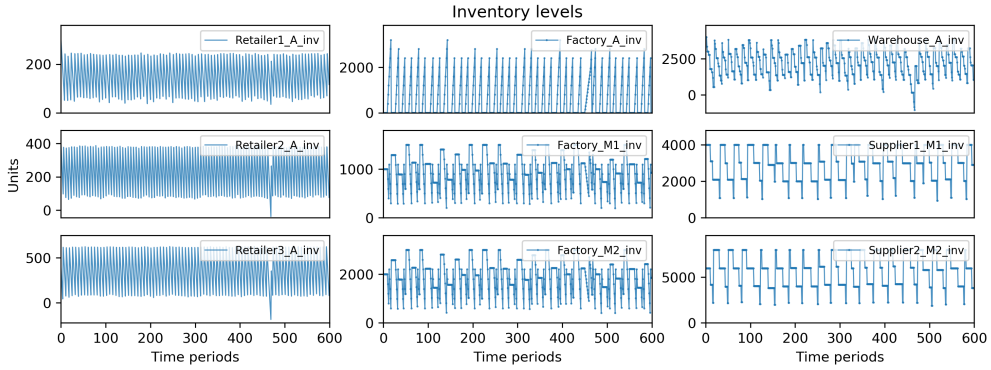
(b) Trajectory SPE contribution of fault-related variables

Figure 7: Case 1, scenario 1, DPCA, SPE contribution plots.

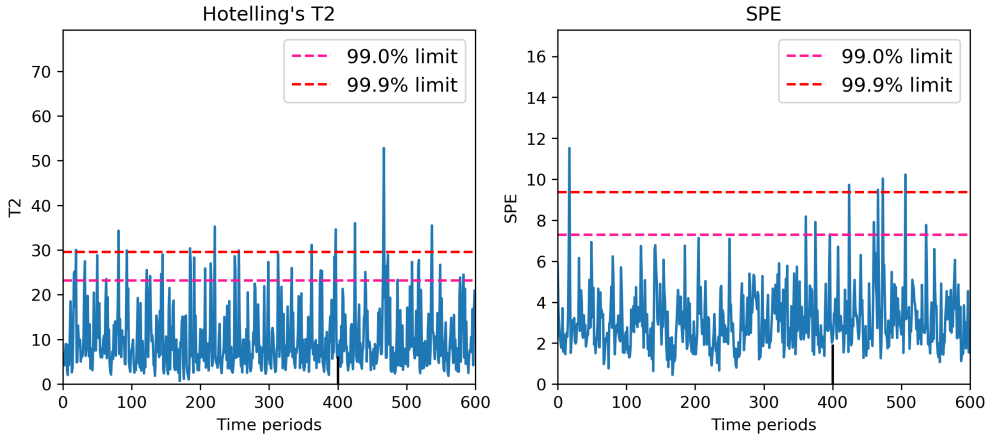
Scenario 2: The second simulated fault scenario is a low production rate (or poor yield) of the Factory. Suppose the production rate of the factory decreases from 400 to 150 units per time period suddenly within time period 451–463 due to some problem. The inventory

profiles of the agents are shown in Figure 8a.

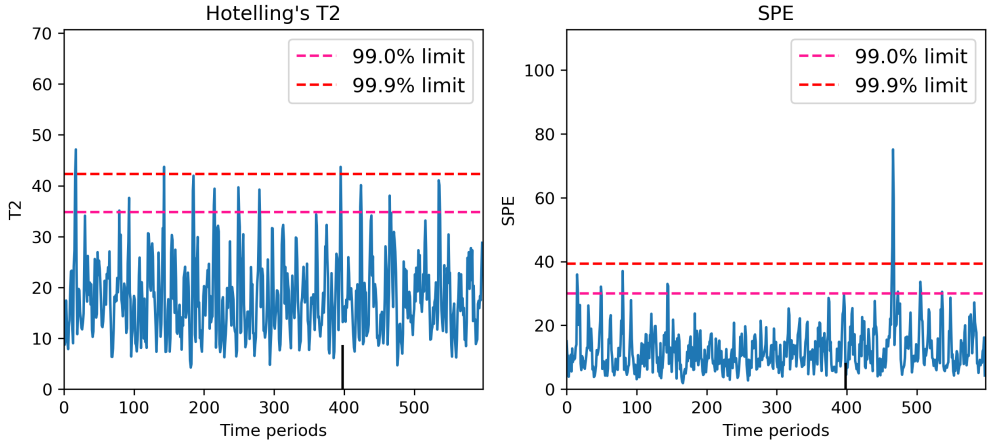
The monitoring charts using PCA and DPCA are shown in Figure 8b and Figure 8c, respectively. It can be seen that during the period with low production rate, the SPE by DPCA exceeds its limit when the inventory level of the Warehouse becomes low, and the orders from the Retailers cannot be satisfied. This means that the effect of the poor yield of the Factory on the supply chain is detected. In comparison, the SPE by PCA cannot detect the fault. For diagnosis, the SPE contribution plot of an abnormal sample and the trajectory SPE contribution plot using DPCA are shown in Figure 9a and Figure 9b, respectively. It can be seen that the contribution of the variable related to the inventory level of Warehouse is the largest. This implies that it is likely to have a large variation from the NOC, and hence indicates the low yield problem of the Factory and its effect on the operation of the Warehouse. The SPE falls below the limits after the supply chain returns to NOC. In terms of the Hotelling's T^2, PCA detects the fault while DPCA does not. This implies that PCA takes this fault as a large variation from NOC in the PCS, while DPCA takes it as a large variation in the RS.



(a) Inventory levels

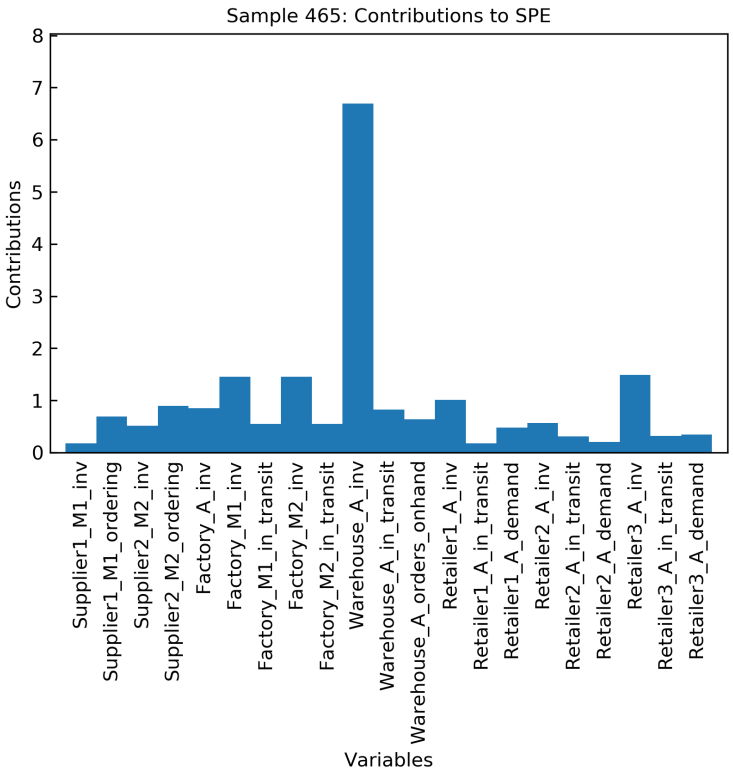


(b) PCA

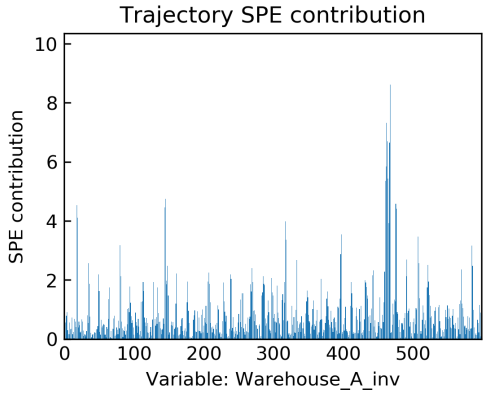


(c) DPCA

Figure 8: Case 1, scenario 2: low production rate



(a) Sample SPE contribution



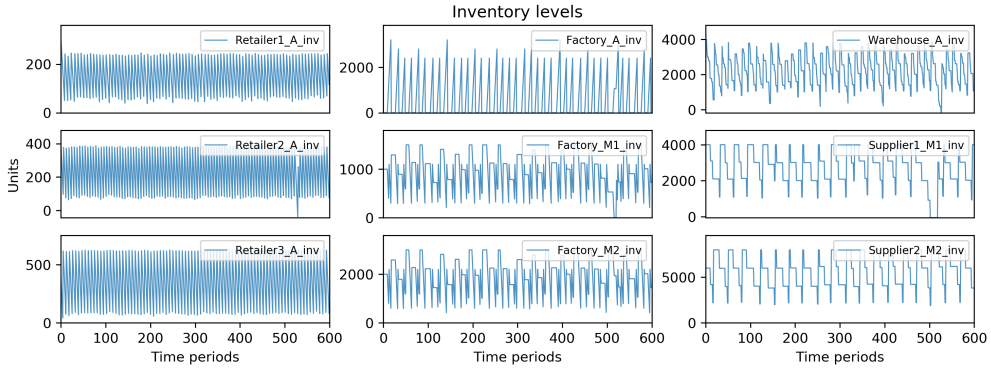
(b) Trajectory SPE contribution of fault-related variables

Figure 9: Case 1, scenario 2, DPCA, SPE contribution plots.

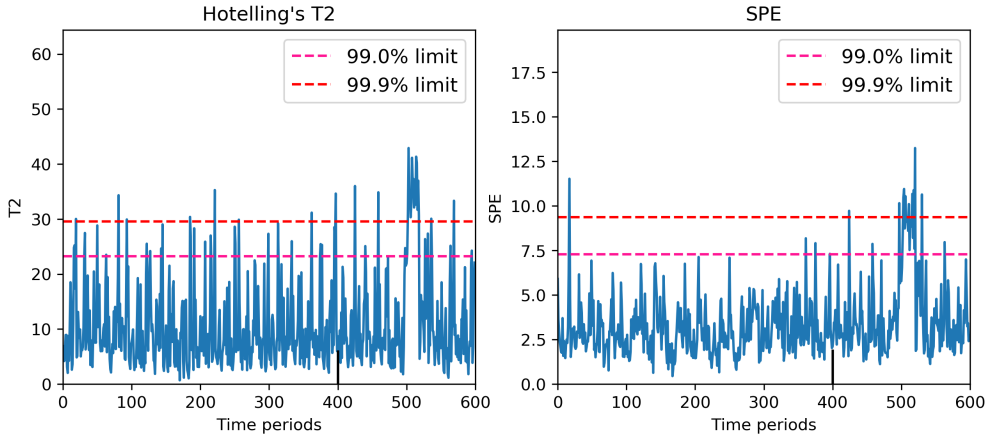
Scenario 3: The third fault scenario is a raw material shortage of a supplier. It is simulated by a shortage occurring at the Supplier1 during time period 498–519. Figure 10a shows the inventory levels of the supply chain participants. Because of the shortage, the Factory cannot

get raw material 'M1' from the Supplier1 and has to stop manufacturing. The Warehouse subsequently cannot get replenished, and a backorder situation occurs at the Retailer2.

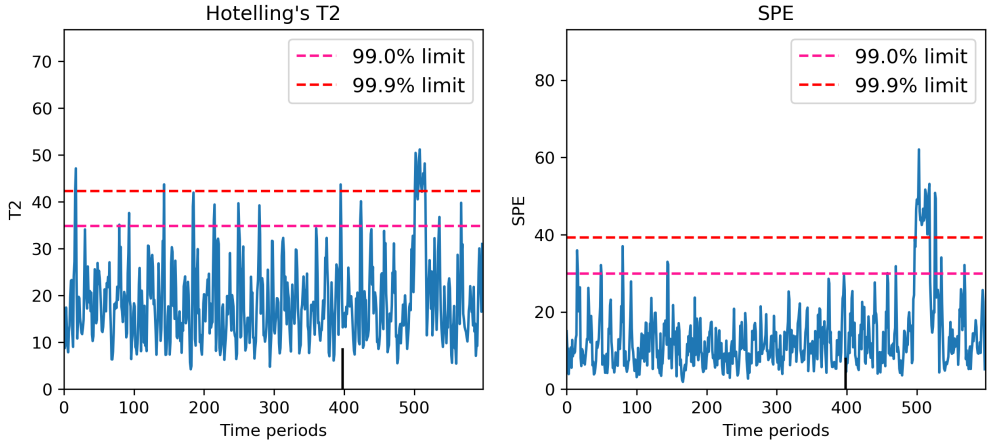
The monitoring charts by PCA and DPCA are shown in Figure 10b and Figure 10c, respectively. It can be seen that by DPCA, both the T^2 and SPE raise an alarm when the supplier's inventory level becomes low and abnormal, which is at the beginning of the stockout of the supplier. This means that the abnormal behavior of the supply chain is detected soon after the stockout occurs at the supplier, before it affects downstream agents. The T^2 and SPE by PCA also pick up this fault. The SPE contribution plot of the abnormal sample 501 and the trajectory SPE contribution plot using DPCA are shown in Figure 11a and Figure 11b, respectively. It can be seen that the contribution of the variable related to the material in processing of Supplier1 is the largest, which implies it is likely to have a large variation from the NOC. Hence, the effect of the shortage of Supplier1 on the supply chain is detected. After the supply recovers, the supply chain returns to NOC, and the monitoring statistics fall below the limits.



(a) Inventory levels

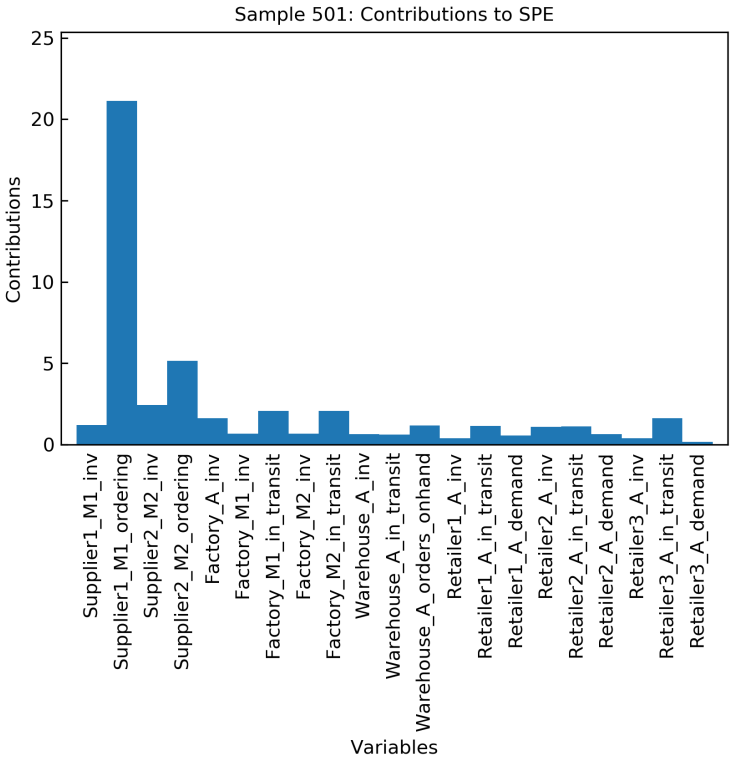


(b) PCA

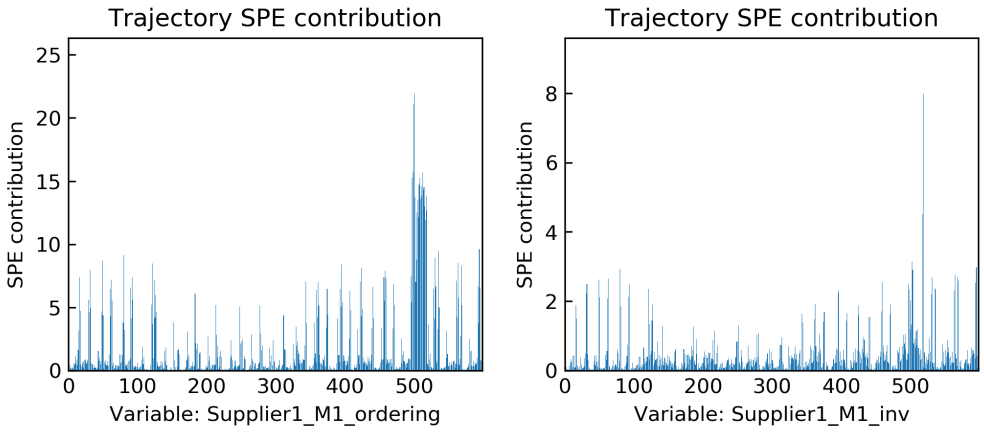


(c) DPCA

Figure 10: Case 1, scenario 3: supply shortage of Supplier1



(a) Sample SPE contribution

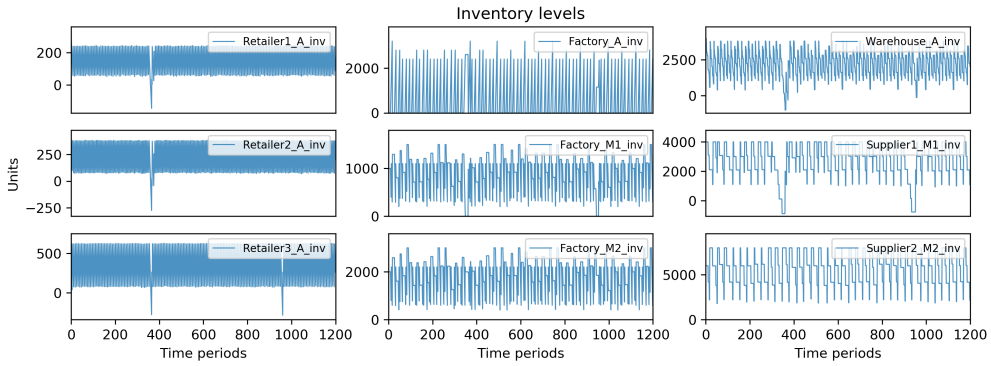


(b) Trajectory SPE contribution plot

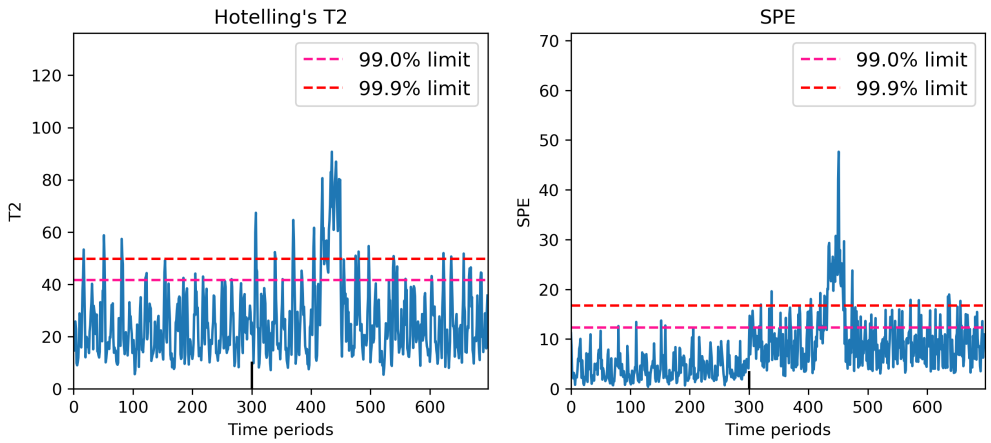
Figure 11: Case 1, scenario 3, DPCA, SPE contribution plots.

It is worth mentioning that the training data does not have to be contiguous in time. The training set can be organized by removing fault data within some time periods and stacking the NOC data. A comparison of the DPCA results using different training sets while the

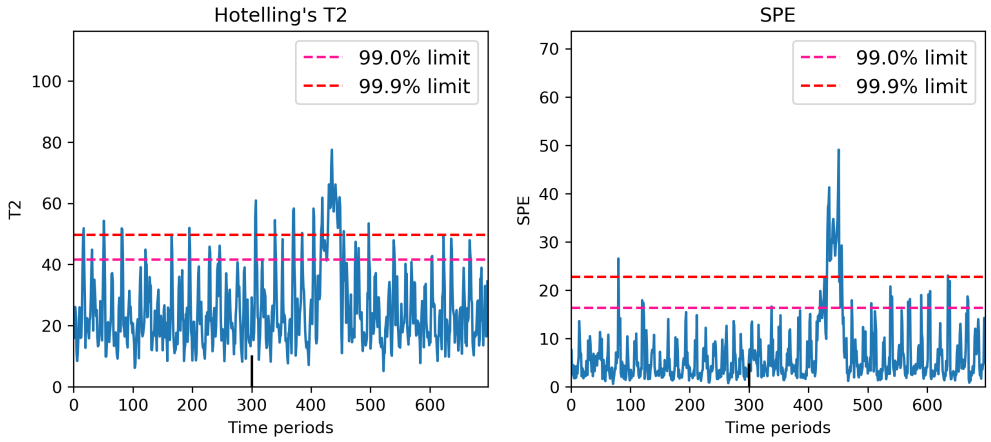
same number of PCs is given in Figure 12. Figure 12a shows the inventory levels over the simulated 1200 time periods, during which a shortage occurs twice at the Supplier1. The result when taking samples 0–300 as the training set, and samples 801–1200 as the testing set is shown in Figure 12b. The horizontal axes are labelled with a range of 0–700 for contiguous representation of these samples. It can be seen that although the abnormality is detected, the SPE of the NOC data in the testing set is generally higher than that of the training data. The reason for this is that when the supply chain recovers to NOC after the shortage, the relation between the variables and the operating status of the supply chain may change. Therefore, an increase in SPE is observed. In comparison, when combining samples 0–150 and 650–800 as the training set, the SPE of the NOC data in the testing set remain close to that of the training data, as shown in Figure 12c. This is because the NOC data after the shortage have been included in the training set. The comparison indicates that organizing the training set by removing fault data and stacking NOC data could help reduce overfitting to some extent. Moreover, it implies that new NOC data can be included into the training set to update the DPCA model.



(a) Inventory levels



(b) DPCA, using samples 0–300 as training set, 801–1200 as testing set



(c) DPCA, using samples 0–150 and 650–800 as training set, 801–1200 as testing set

Figure 12: Case 1, comparison of DPCA results using different training data.

From the analysis and results shown above, the faults like transportation delay, low production rate, and supply shortage can be detected by the proposed supply chain monitoring method using DPCA. The monitoring charts raise an alarm when the abnormality occurs.

4.2 Case Study 2: A packaged liquefied gas supply chain

4.2.1 Simulation

The second case study is based on the packaged liquefied gas supply chain investigated by Misra et al.[82]. The gas products are stored in containers (stock keeping units, SKUs). The structure of this supply chain is shown in Figure 13. It consists of 3 echelons: (1) the Customer locations, where the filled SKUs are consumed and then empty SKUs are generated. (2) the Warehouse, which stores both filled and empty SKUs. Filled SKUs are transported to the Customer locations to be replenished, and empty SKUs are collected from the Customer locations. The empty SKUs are transported to the Plants for refilling. (3) the Plants, where the empty SKUs from the Warehouse are refilled and then transported back.

The difference between this case study and the first case study is that the empty SKUs from downstream are transported back upstream for refilling. Therefore, there is both flow of filled SKUs from upstream to downstream, and flow of empty SKUs from downstream to upstream.

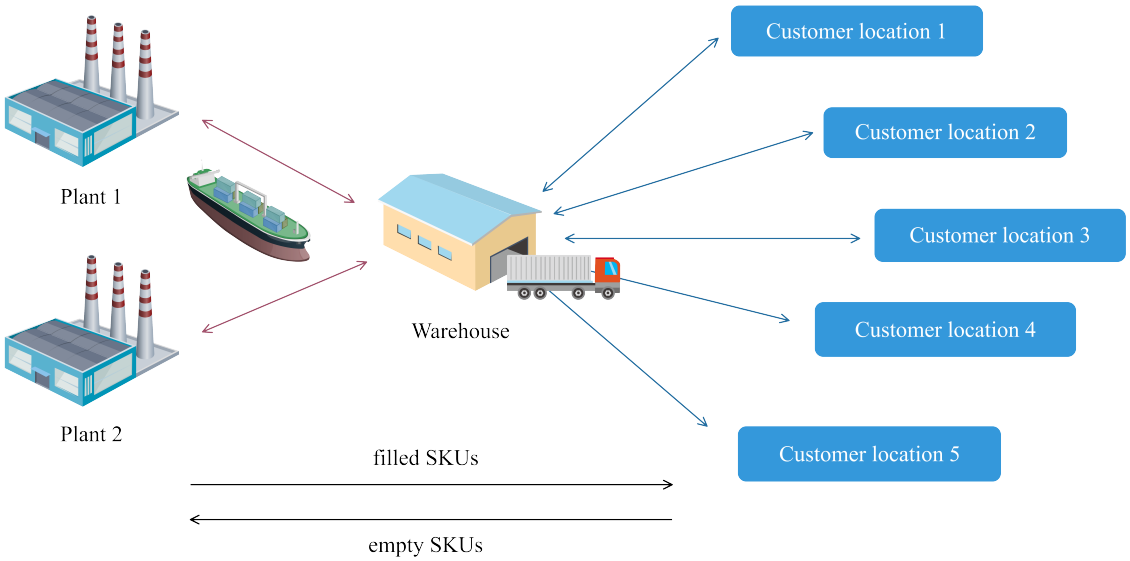


Figure 13: The packaged liquefied gas supply chain from Misra et al. [82].

In Misra et al. [82], vendor managed inventory (VMI) is adopted to make sure the filled SKU inventory levels of the Customer locations are above the number of filled SKUs currently being used. VMI means that the suppliers make the replenishment decision. In the simulation in this paper, the inventory policy is implemented in the following method: when the inventory level of filled SKUs of a product at a Customer location is below its 're-order point', filled SKUs of this product are shipped from the Warehouse to the Customer location to replenish the stock. The number of delivered filled SKUs is exactly equal to the collected empty SKUs. Simultaneous delivery and pickup is adopted for the Customer locations, which means that the trucks bring the filled SKUs to the Customer location, and collect the empty SKUs from it in the same time period of arrival, then bring them back to the Warehouse. All the delivered products originate from the Warehouse and all the collected empty containers are sent back to the Warehouse. When the inventory level of filled SKUs of the Warehouse is below its 're-order point', the empty SKUs are transported to the Plants for refilling. Ships are used for upstream transportation, while trucks are used for downstream distribution [82]. Hence in this paper, the transportation times of upstream and downstream are set as 3 time periods and 1 time period, respectively.

In order to simplify the simulation model, some assumptions are made here. Two types of products A and B are transported across this supply chain, and refilled at the Plant 1 and 2, respectively; there is no product in transit before time period 0; in normal operating conditions, the transportation times are constant. The supply chain parameters used in this paper, such as the initial inventory levels and re-order points, are listed in Table 4. The time interval of simulation is one time period. The normal period is from 0 to 600 time periods. The numbers of the SKUs in the supply chain are assumed to be fixed, and there are no new or departing customers.

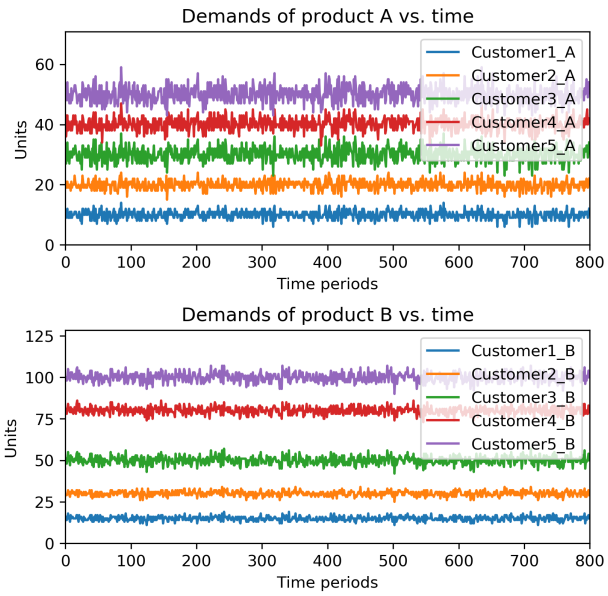
Table 4: Parameters of the supply chain

Participant	demand (mean)		initial filled SKUs		initial empty SKUs		re-order points	
	A	B	A	B	A	B	A	B
Customer1	10	15	100	150	0	0	50	80
Customer2	20	30	200	350	0	0	100	200
Customer3	30	50	300	450	0	0	180	250
Customer4	40	80	400	800	0	0	240	500
Customer5	50	100	600	1500	0	0	400	800
Warehouse	–	–	4000	8000	2000	3500	2000	4000

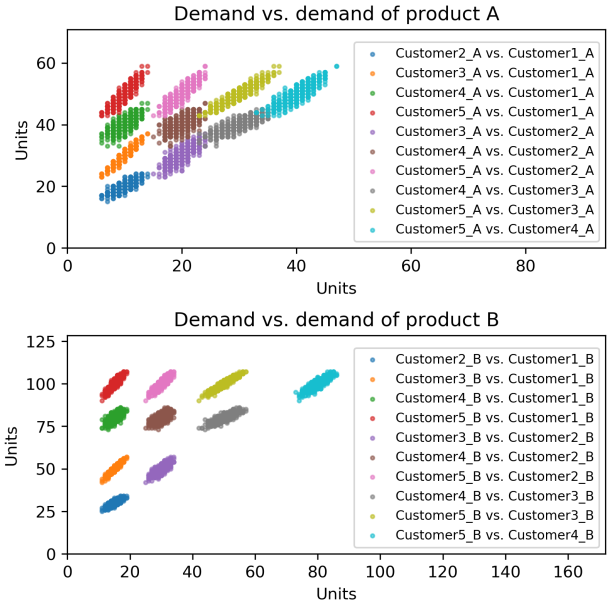
The demands for A and B at the 5 Customer locations are shown in Figure 14. They are generated from multivariate Gaussian distributions, respectively, and then rounded to integers. The demand means the amount of filled SKUs consumed at a Customer location in each time period, which is equal to the amount of empty SKUs generated. The simulated inventory profiles of the agents for the two products are shown in Figure 15a and Figure 15b, respectively. The data of the Warehouse and Customer locations are collected for analysis. There are a total of 60 variables collected, including the demands for A and B at each Customer location (10 variables), inventory levels of filled and empty SKUs for A and B at each agent (total of 24 variables), and the amount of filled and empty SKUs for A and B in transit and in refilling at the Plants (total of 26 variables).

The supply chain operation over 800 time periods is simulated and analyzed. Each time

period can be seen as 1 day. The NOC data during time period 10–600 is taken as the training set. Using DPCA with 2 time lags, the raw data is augmented to $60 \times 3 = 180$ variables, and 50 PCs are extracted to achieve a R^2 value of 95%. The data between period 601–800 is taken as the testing set, which are also normal data. The monitoring charts for the D- and Q- statistics using DPCA are shown in Figure 16. The 99% and 99.9% confidence limits are estimated from the training data and plotted as dashed lines. It can be seen that in NOC, the two statistics of the testing set are within the confidence limits. This suggests no unexpected event occurs in the supply chain and the testing data can be seen as normal.

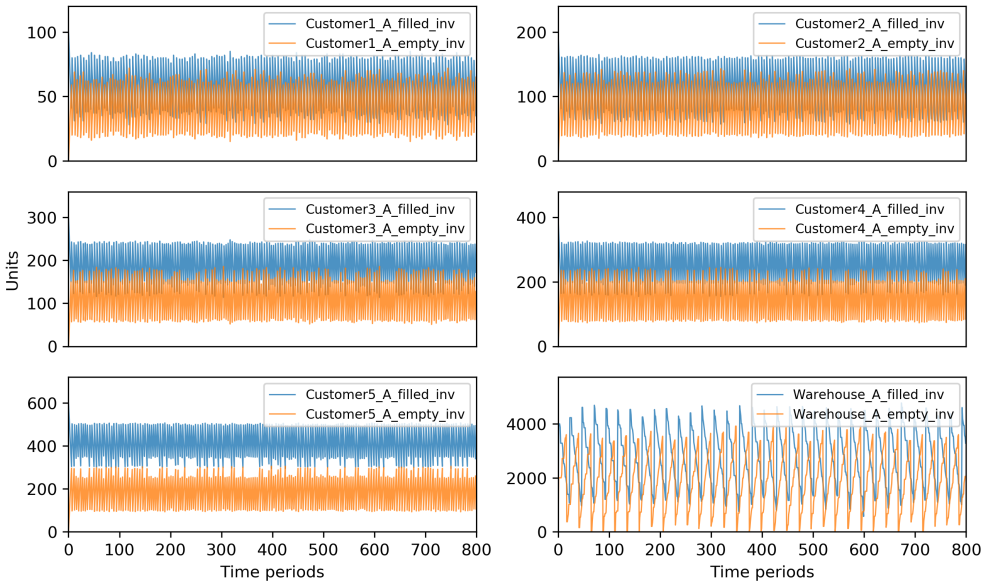


(a) Demands versus time

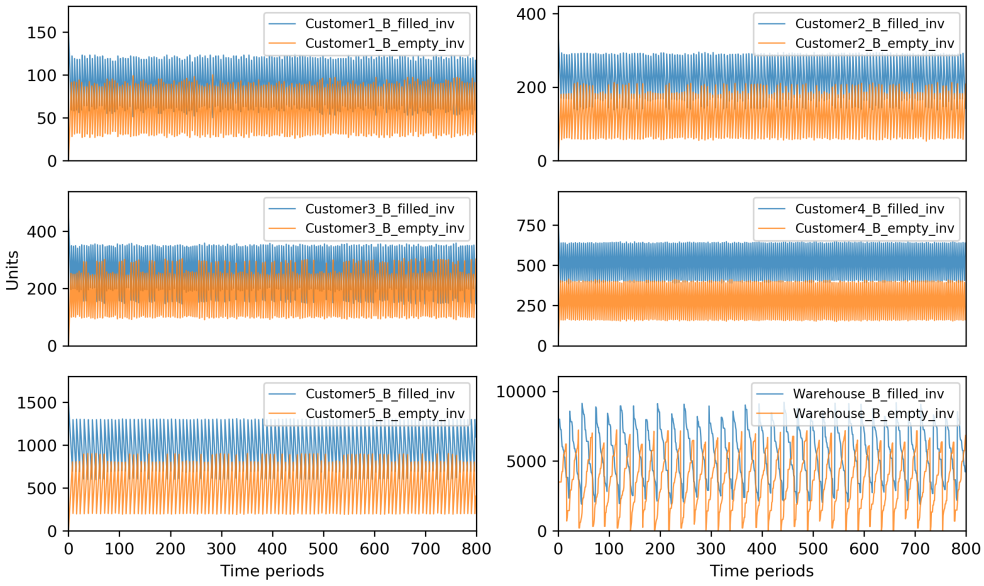


(b) Demand versus demand for Customer pairs

Figure 14: Case 2: demands



(a) Product A



(b) Product B

Figure 15: Case 2, NOC, inventory levels

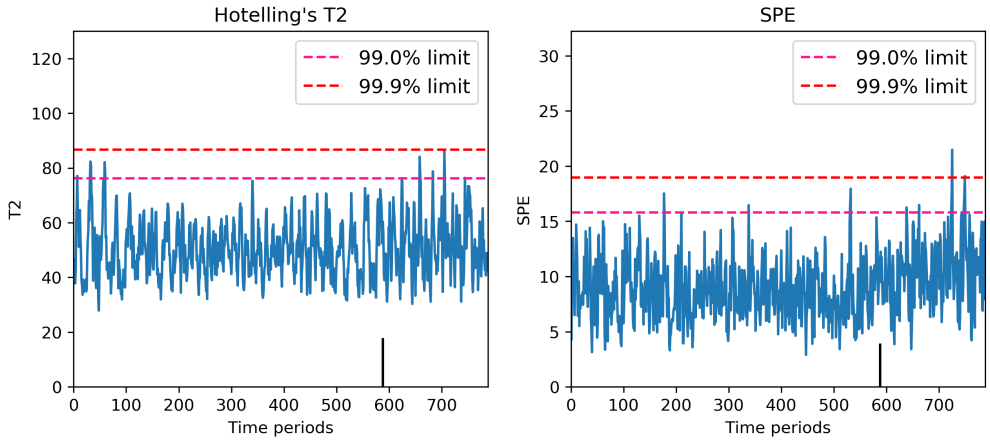


Figure 16: Case 2, NOC, DPCA

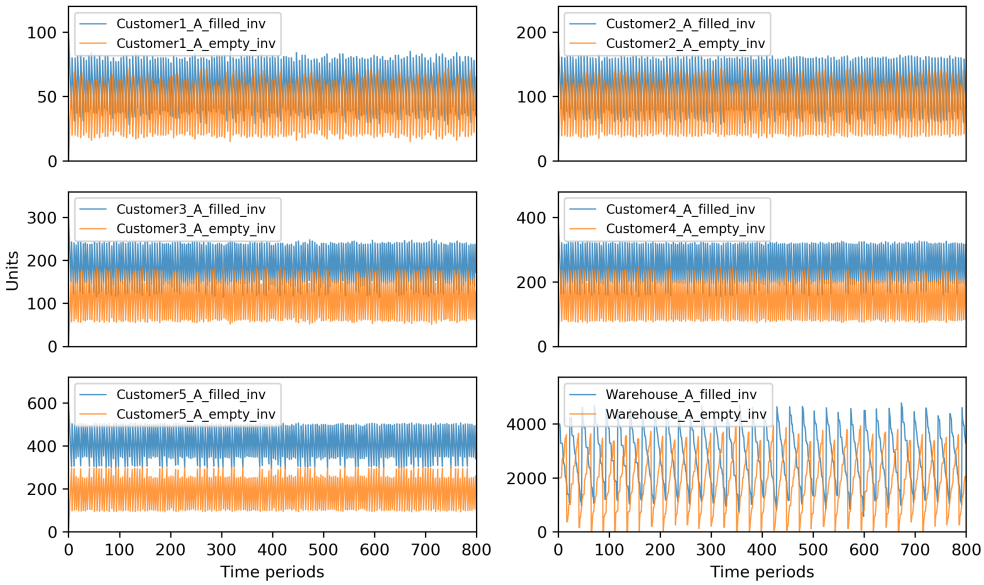
4.2.2 FDD using DPCA

The transportation delay is stated as a supply chain risk by Chopra and Sodhi[6]. In this case study, two fault scenarios are designed in the simulation and analyzed: the transportation delay under Gaussian demands and seasonal demands, respectively. In the analysis, DPCA is focused on.

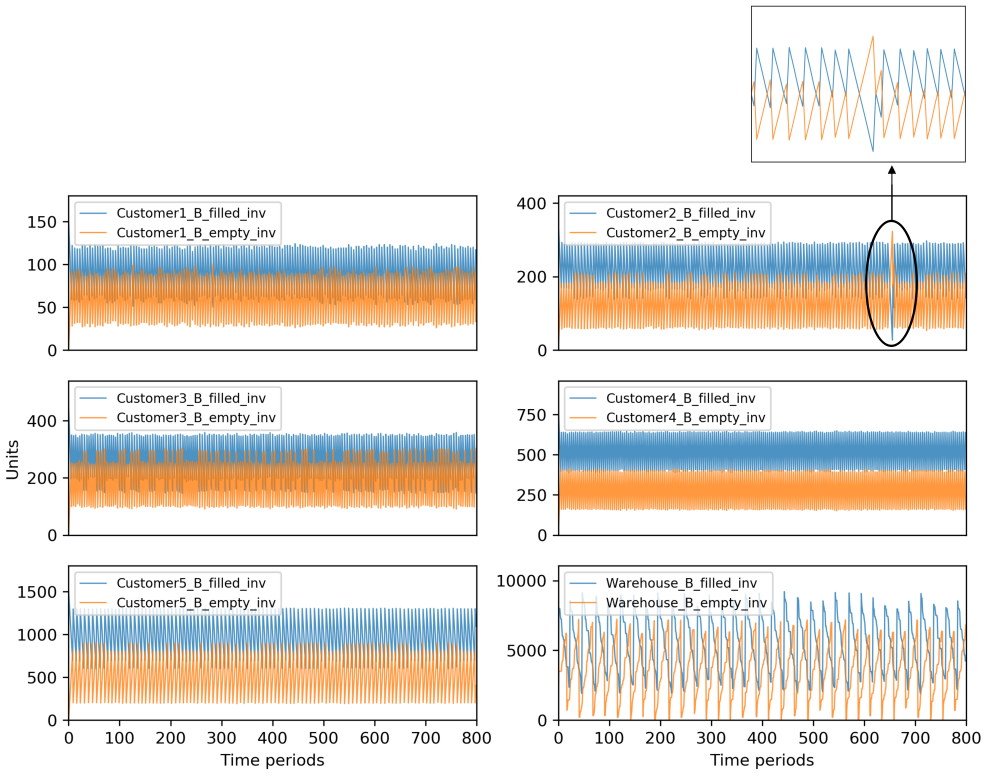
Scenario 1: The first scenario is the transportation delay under multivariate Gaussian demands. Suppose a problem occurs in the transportation link between the Warehouse and the Customer location 2, and the transportation time of the shipment of product B to the Customer location 2 at time period 650 increases from 1 day to 5 time periods. The inventory profiles of the agents are shown in Figure 17a and Figure 17b. It can be seen from Figure 17b that when the delay occurs, the Customer location 2's inventory level of filled SKUs of product B becomes low and the empty SKUs pile up.

The monitoring charts using DPCA are shown in Figure 18. It can be seen that when the delay occurs, the SPE by DPCA exceeds the 99.9% confidence limit by a wide margin, which means that there is a large variation from the NOC. In order to identify the fault-related

variables, the SPE contribution plots by DPCA of a sample above the limit is presented in Figure 19. From this plot, the variables related to the Customer location 2's inventory of product B contribute the largest to the SPE, which implies that they are most likely to be fault-related, and some abnormal event might have occurred at Customer location 2. This helps in the diagnosis of the root cause.



(a) Product A



(b) Product B

Figure 17: Case 2, scenario 1, inventory levels

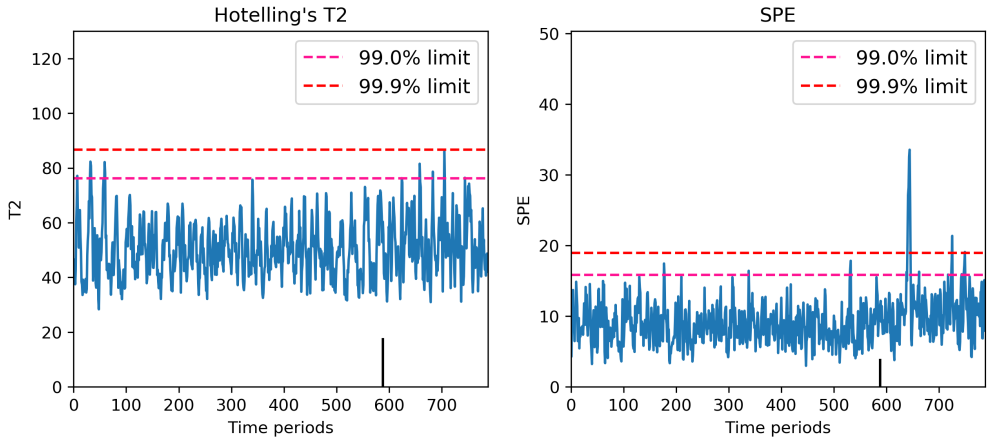


Figure 18: Case 2, scenario 1, DPCA

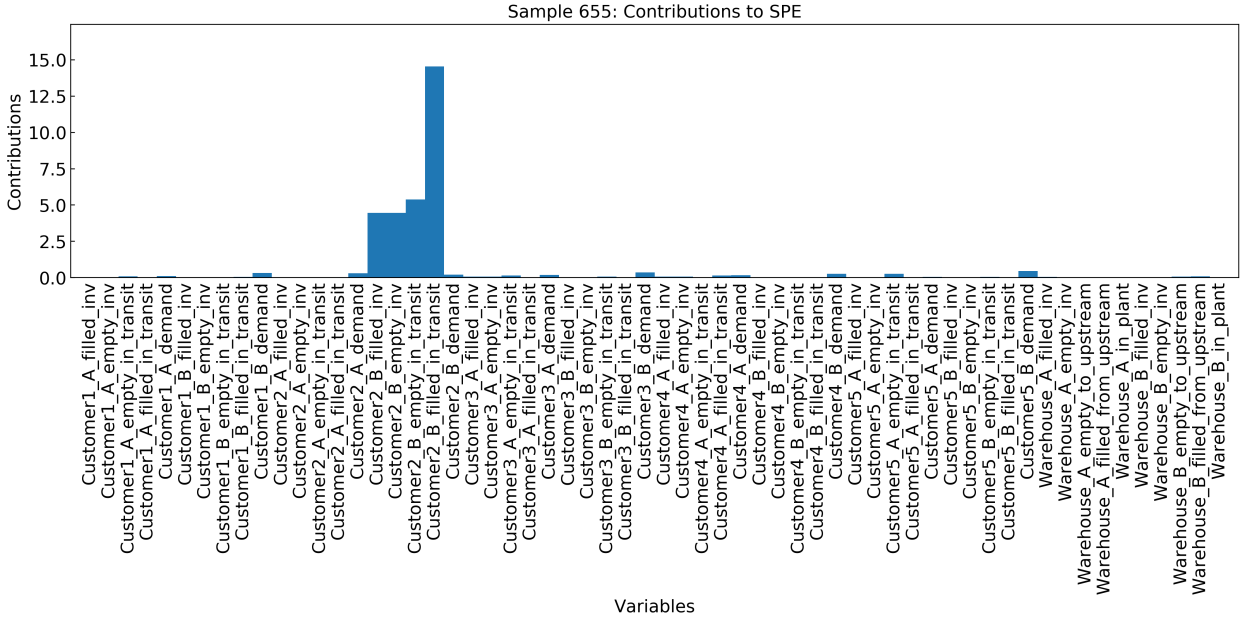
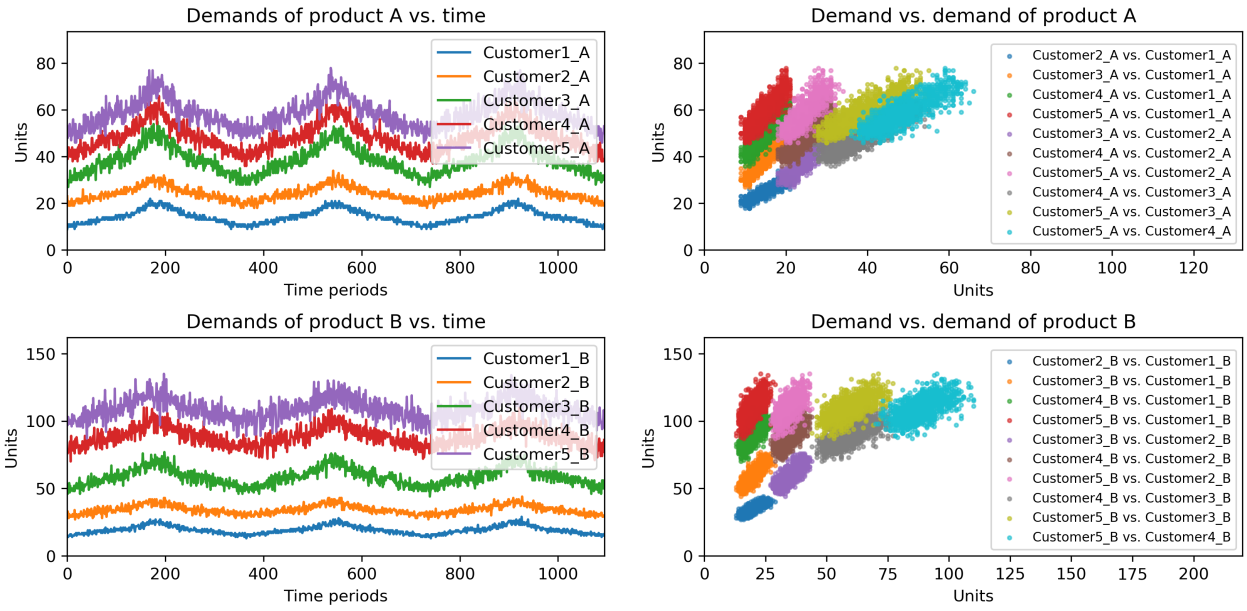


Figure 19: Case 2, scenario 1, DPCA, sample SPE contribution.

Scenario 2: The other fault scenario simulated is a transportation delay under seasonal demands, which could cause a variation in the inventory level at the affected Customer location. The seasonal demands at the 5 Customer locations are shown in Figure 20. For this scenario, a total of 1095 time periods are simulated. The Customer location 2 experiences a transportation delay for the shipment of product B from the Warehouse at time period 784. The shipment is delayed to 5 time periods. The inventory profiles of the agents are shown in

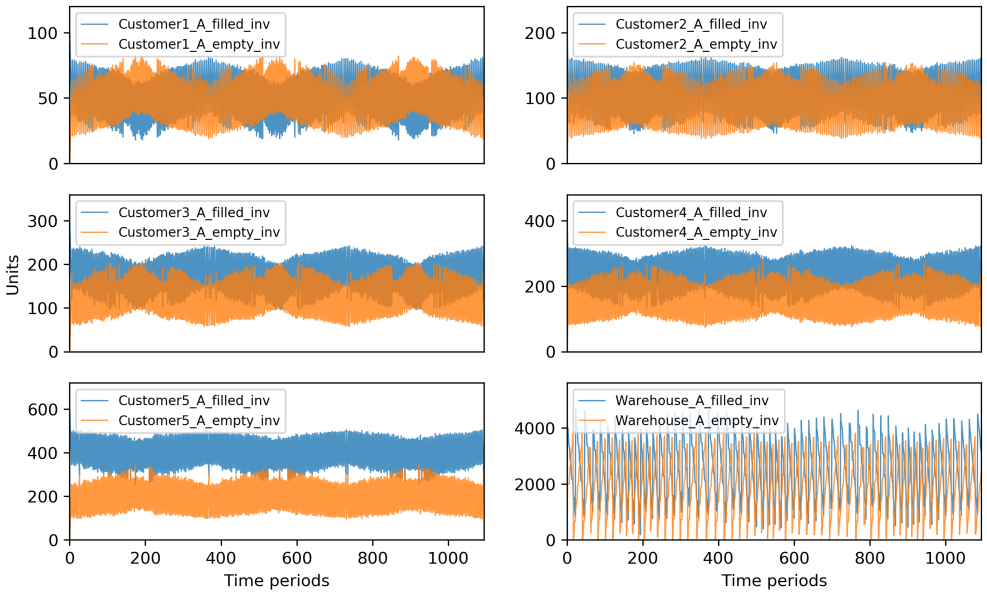
Figure 21. It can be seen from Figure 21b that at Customer location 2, the inventory level of filled SKUs of product B is lower than usual when the delay occurs, and the empty SKUs pile up.

The NOC data during time period 10–730 are taken as the training set, and the data during time period 731–1095 are taken as the testing set. 45 PCs are retained to achieve a R^2 value of 95%. The monitoring charts using DPCA are shown in Figure 22. It can be seen that during the period when the delay occurs, the SPE greatly exceeds its limit. This means that the effect of the delay on the inventory levels of the Customer location 2 is successfully detected. For diagnosis, the SPE contribution plot of an abnormal sample is shown in Figure 23. It can be seen that the contributions of variables related to the Customer location 2's inventory of product B are the largest. This implies the inventory of the Customer location 2 has a large variation from the NOC, and hence the fault is identified.

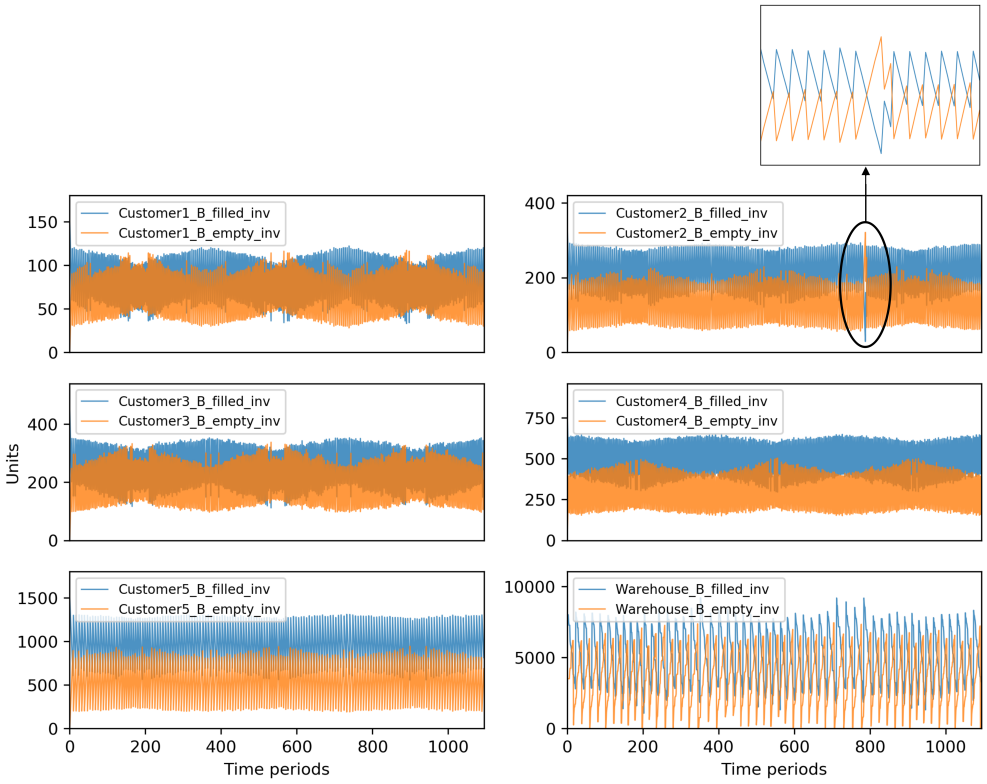


(a) Demands versus time (b) Demand versus demand for Customer pairs

Figure 20: Case 2, scenario 2: seasonal demands at the 5 Customer locations.



(a) Product A



(b) Product B

Figure 21: Case 2, scenario 2: inventory levels

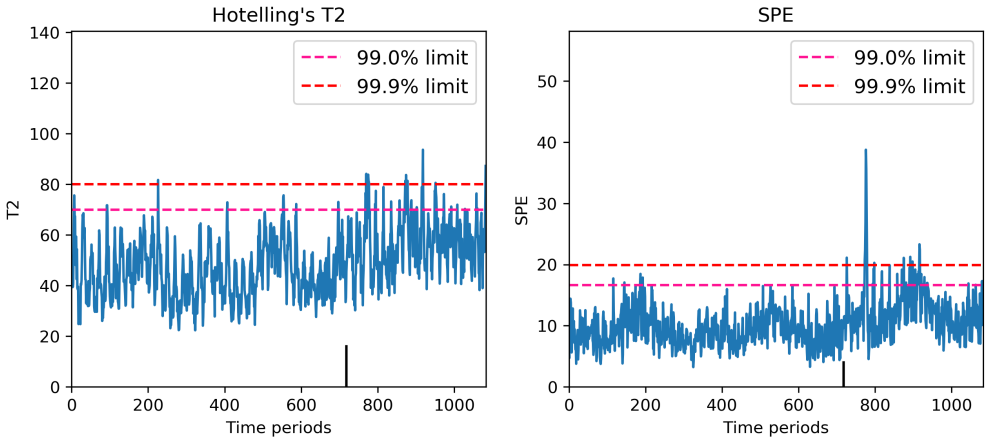


Figure 22: Case 2, scenario 2, DPCA

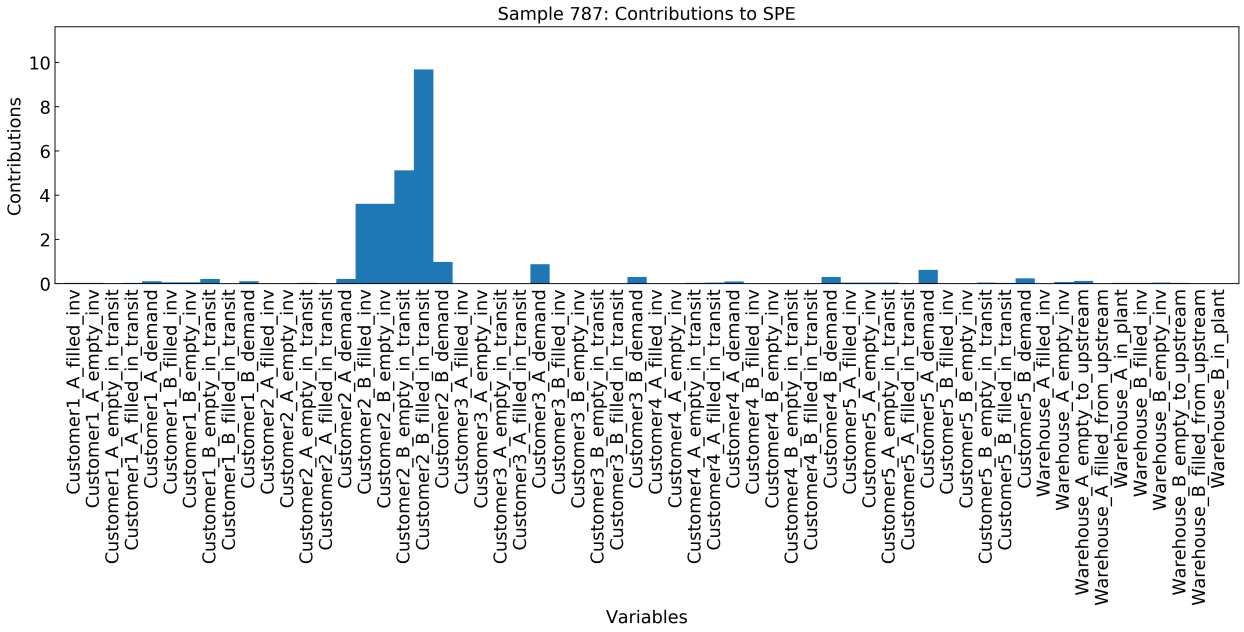


Figure 23: Case 2, scenario 2, DPCA, sample SPE contribution.

From the analysis and results shown above, the effect of transportation delay on the supply chain, under Gaussian and seasonal demands, is detected by the proposed supply chain monitoring method using DPCA.

5 Conclusion

In this paper, PCA and dynamic PCA are applied for the fault detection and diagnosis of supply chain systems. In order to monitor the supply chain, data such as inventory levels, market demands, and material in transit are collected. PCA and DPCA are employed to model the NOC of the supply chain, and the Hotelling's T^2 and SPE are used to detect abnormal behavior of the supply chain. Contribution plots are adopted to identify the fault-related variables when at least one index exceeds the limits. The proposed supply chain monitoring method is validated on two case studies, one of which is a 4-echelon supply chain with single product, and the other is a two-product supply chain with materials transported not only from upstream to downstream but also in the opposite direction. A Python-based supply chain simulator is developed to generate supply chain data. Different scenarios are simulated and analyzed. The results show that the SPE by DPCA is more reliable than the other fault detection indices considered in this paper, while that by PCA is not sensitive enough. Abnormal behavior of the supply chain, such as transportation delay, low production rate and supply shortage, can be successfully detected by DPCA. The proposed method applies to non-contiguous data and seasonal market demands. Moreover, the contribution plots can help interpret the abnormality and identify the fault-related variables.

Acknowledgments

This work was funded by the Natural Sciences and Engineering Council of Canada, Grants RGPIN-05627-2017, RGPIN-06524-2015 and CREATE 466264-2015.

References

(1) Patel, S.; Swartz, C. L. E. Supply chain design with time-limited transportation contracts. *Computers & Chemical Engineering* **2019**, https://doi.org/10.1016/j.compchemeng.2019.106579.

(2) Mastragostino, R.; Patel, S.; Swartz, C. L. E. Robust decision making for hybrid process supply chain systems via model predictive control. *Computers & Chemical Engineering* **2014**, *62*, 37–55.

(3) Grossmann, I. E. Enterprise-wide optimization: A new frontier in process systems engineering. *AIChE Journal* **2005**, *51*, 1846–1857.

(4) Shah, N. Process industry supply chains: Advances and challenges. *Computers & Chemical Engineering* **2005**, *29*, 1225–1236.

(5) Papageorgiou, L. G. Supply chain optimisation for the process industries: Advances and opportunities. *Computers & Chemical Engineering* **2009**, *33*, 1931–1938.

(6) Chopra, S.; Sodhi, M. S. Managing risk to avoid supply-chain breakdown. *MIT Sloan Management Review* **2004**, *46*, 53–62.

(7) Tang, C. S. Perspectives in supply chain risk management. *International Journal of Production Economics* **2006**, *103*, 451–488.

(8) Carvalho, H.; Barroso, A. P.; Machado, V. H.; Azevedo, S.; Cruz-Machado, V. Supply chain redesign for resilience using simulation. *Computers & Industrial Engineering* **2012**, *62*, 329–341.

(9) Wilson, M. C. The impact of transportation disruptions on supply chain performance. *Transportation Research Part E: Logistics and Transportation Review* **2007**, *43*, 295–320.

(10) Tang, C. S. Robust strategies for mitigating supply chain disruptions. *International Journal of Logistics Research and Applications* **2006**, *9*, 33–45.

(11) Chopra, S.; Sodhi, M. S. Reducing the risk of supply chain disruptions. *MIT Sloan Management Review* **2014**, *55*, 73–80.

(12) Sheffi, Y. Preparing for disruptions through early detection. *MIT Sloan Management Review* **2015**, *57*, 31–42.

(13) Souza, G. C. Supply chain analytics. *Business Horizons* **2014**, *57*, 595–605.

(14) Wang, G.; Gunasekaran, A.; Ngai, E. W.; Papadopoulos, T. Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *International Journal of Production Economics* **2016**, *176*, 98–110.

(15) Tiwari, S.; Wee, H.; Daryanto, Y. Big data analytics in supply chain management between 2010 and 2016: Insights to industries. *Computers & Industrial Engineering* **2018**, *115*, 319–330.

(16) Mishra, D.; Gunasekaran, A.; Papadopoulos, T.; Childe, S. J. Big Data and supply chain management: a review and bibliometric analysis. *Annals of Operations Research* **2018**, *270*, 313–336.

(17) Nguyen, T.; Zhou, L.; Spiegler, V.; Ieromonachou, P.; Lin, Y. Big data analytics in supply chain management: A state-of-the-art literature review. *Computers and Operations Research* **2018**, *98*, 254–264.

(18) Jain, R.; Singh, A. R.; Yadav, H. C.; Mishra, P. K. Using data mining synergies for evaluating criteria at pre-qualification stage of supplier selection. *Journal of Intelligent Manufacturing* **2014**, *25*, 165–175.

(19) Mori, J.; Kajikawa, Y.; Kashima, H.; Sakata, I. Machine learning approach for finding

business partners and building reciprocal relationships. *Expert Systems with Applications* **2012**, *39*, 10402–10407.

(20) Zhong, R. Y.; Huang, G. Q.; Lan, S.; Dai, Q.; Zhang, T.; Xu, C. A two-level advanced production planning and scheduling model for RFID-enabled ubiquitous manufacturing. *Advanced Engineering Informatics* **2015**, *29*, 799–812.

(21) Zhao, R.; Liu, Y.; Zhang, N.; Huang, T. An optimization model for green supply chain management by using a big data analytic approach. *Journal of Cleaner Production* **2017**, *142*, 1085–1097.

(22) Toole, J. L.; Colak, S.; Sturt, B.; Alexander, L. P.; Evsukoff, A.; González, M. C. The path most traveled: Travel demand estimation using big data resources. *Transportation Research Part C: Emerging Technologies* **2015**, *58*, 162–177.

(23) Li, L.; Su, X.; Wang, Y.; Lin, Y.; Li, Z.; Li, Y. Robust causal dependence mining in big data network and its application to traffic flow predictions. *Transportation Research Part C: Emerging Technologies* **2015**, *58*, 292–307.

(24) Chiang, D. M.-H.; Lin, C.-P.; Chen, M.-C. The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterprise Information Systems* **2011**, *5*, 219–234.

(25) Tsai, C.-Y.; Huang, S.-H. A data mining approach to optimise shelf space allocation in consideration of customer purchase and moving behaviours. *International Journal of Production Research* **2015**, *53*, 850–866.

(26) Salehan, M.; Kim, D. J. Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decision Support Systems* **2016**, *81*, 30–40.

(27) Alain Yee, L. C.; Li, B.; Ngai, E. W. T.; Ch'ng, E.; Filbert, L. Predicting online product sales via online reviews, sentiments, and promotion strategies. *International Journal of Operations & Production Management* **2016**, *36*, 358–383.

(28) Ma, J.; Kwak, M.; Kim, H. M. Demand trend mining for predictive life cycle design. *Journal of Cleaner Production* **2014**, *68*, 189–199.

(29) Ning, C.; You, F. Data-driven adaptive nested robust optimization: General modeling framework and efficient computational algorithm for decision making under uncertainty. *AIChE Journal* **2017**, *63*, 3790–3817.

(30) Ning, C.; You, F. Data-driven stochastic robust optimization: General computational framework and algorithm leveraging machine learning for optimization under uncertainty in the big data era. *Computers & Chemical Engineering* **2018**, *111*, 115–133.

(31) Shang, C.; You, F. Distributionally robust optimization for planning and scheduling under uncertainty. *Computers & Chemical Engineering* **2018**, *110*, 53–68.

(32) Gao, J.; Ning, C.; You, F. Data-driven distributionally robust optimization of shale gas supply chains under uncertainty. *AIChE Journal* **2019**, *65*, 947–963.

(33) Ning, C.; You, F. Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering* **2019**, *125*, 434–448.

(34) Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **1987**, *2*, 37–52.

(35) Ge, Z. Review on data-driven modeling and monitoring for plant-wide industrial processes. *Chemometrics and Intelligent Laboratory Systems* **2017**, *171*, 16–25.

(36) Qin, S. J.; Chiang, L. H. Advances and opportunities in machine learning for process data analytics. *Computers and Chemical Engineering* **2019**, *126*, 465–473.

(37) Kresta, J. V.; MacGregor, J. F.; Marlin, T. E. Multivariate statistical monitoring of process operating performance. *The Canadian Journal of Chemical Engineering* **1991**, *69*, 35–47.

(38) Kourti, T.; MacGregor, J. F. Process analysis, monitoring and diagnosis, using multivariate projection methods. *Chemometrics and Intelligent Laboratory Systems* **1995**, *28*, 3–21.

(39) MacGregor, J. F.; Kourti, T. Statistical process control of multivariate processes. *Control Engineering Practice* **1995**, *3*, 403–414.

(40) Nomikos, P.; MacGregor, J. F. Multivariate SPC charts for monitoring batch processes. *Technometrics* **1995**, *37*, 41–59.

(41) Qin, S. J. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control* **2012**, *36*, 220–234.

(42) Ge, Z.; Song, Z.; Ding, S. X.; Huang, B. Data mining and analytics in the process industry: The role of machine learning. *IEEE Access* **2017**, *5*, 20590–20616.

(43) Ku, W.; Storer, R. H.; Georgakis, C. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **1995**, *30*, 179–196.

(44) Russell, E. L.; Chiang, L. H.; Braatz, R. D. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **2000**, *51*, 81–93.

(45) Chiang, L. H.; Russell, E. L.; Braatz, R. D. *Fault Detection and Diagnosis in Industrial Systems*; Springer-Verlag, London, 2001.

(46) Li, W.; Qin, S. J. Consistent dynamic PCA based on errors-in-variables subspace identification. *Journal of Process Control* **2001**, *11*, 661–678.

(47) Li, G.; Qin, S. J.; Chai, T. Multi-directional reconstruction based contributions for root-cause diagnosis of dynamic processes. 2014 American Control Conference. 2014; pp 3500–3505.

(48) Li, G.; Qin, S. J.; Yuan, T. Data-driven root cause diagnosis of faults in process industries. *Chemometrics and Intelligent Laboratory Systems* **2016**, *159*, 1–11.

(49) Lee, J.-M.; Yoo, C.; Choi, S. W.; Vanrolleghem, P. A.; Lee, I.-B. Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science* **2004**, *59*, 223–234.

(50) Dong, Y.; Qin, S. J. A novel dynamic PCA algorithm for dynamic data modeling and process monitoring. *Journal of Process Control* **2018**, *67*, 1–11.

(51) Kodamana, H.; Raveendran, R.; Huang, B. Mixtures of probabilistic PCA with common structure latent bases for process monitoring. *IEEE Transactions on Control Systems Technology* **2019**, *27*, 838–846.

(52) Pozo, C.; Ruíz-Femenia, R.; Caballero, J.; Guillén-Gosálbez, G.; Jiménez, L. On the use of Principal Component Analysis for reducing the number of environmental objectives in multi-objective optimization: Application to the design of chemical supply chains. *Chemical Engineering Science* **2012**, *69*, 146–158.

(53) Lei, N.; Moon, S. K. A Decision Support System for market-driven product positioning and design. *Decision Support Systems* **2015**, *69*, 82–91.

(54) How, B. S.; Lam, H. L. Sustainability evaluation for biomass supply chain synthesis: Novel principal component analysis (PCA) aided optimisation approach. *Journal of Cleaner Production* **2018**, *189*, 941–961.

(55) Ning, C.; You, F. Data-driven decision making under uncertainty integrating robust

optimization with principal component analysis and kernel smoothing methods. *Computers and Chemical Engineering* **2018**, *112*, 190–210.

(56) Mele, F. D.; Musulin, E.; Puigjaner, L. Supply chain monitoring a statistical approach. In *European Symposium on Computer Aided Process Engineering-15*; Puigjaner, L., Espuna, A., Eds.; Computer Aided Chemical Engineering; Elsevier, 2005; Vol. 20B; pp 1375–1380.

(57) Wang, Y.; Seborg, D. E.; Larimore, W. E. Process monitoring based on canonical variate analysis. 1997 European Control Conference (ECC). 1997; pp 3089–3094.

(58) Negiz, A.; Çinar, A. Statistical monitoring of multivariable dynamic processes with state-space models. *AIChE Journal* **1997**, *43*, 2002–2020.

(59) Lu, Q.; Jiang, B.; Gopaluni, R. B.; Loewen, P. D.; Braatz, R. D. Sparse canonical variate analysis approach for process monitoring. *Journal of Process Control* **2018**, *71*, 90–102.

(60) Kourti, T.; MacGregor, J. F. Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology* **1996**, *28*, 409–428.

(61) Qin, S. J. Statistical process monitoring: basics and beyond. *Journal of Chemometrics* **2003**, *17*, 480–502.

(62) van Sprang, E. N.; Ramaker, H.-J.; Westerhuis, J. A.; Gurden, S. P.; Smilde, A. K. Critical evaluation of approaches for on-line batch process monitoring. *Chemical Engineering Science* **2002**, *57*, 3979–3991.

(63) MacGregor, J. F.; Cinar, A. Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods. *Computers and Chemical Engineering* **2012**, *47*, 111–120.

(64) Wang, T.; Wu, H.; Ni, M.; Zhang, M.; Dong, J.; Benbouzid, M. E. H.; Hu, X. An adaptive confidence limit for periodic non-steady conditions fault detection. *Mechanical Systems and Signal Processing* **2016**, *72-73*, 328–345.

(65) Westerhuis, J. A.; Gurden, S. P.; Smilde, A. K. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems* **2000**, *51*, 95–114.

(66) Miller, P.; Swanson, R. E.; Heckler, C. E. Contribution plots: a missing link in multivariate quality control. *Applied Mathematics and Computer Science* **1998**, *8*, 775–792.

(67) Yoon, S.; MacGregor, J. F. Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *Journal of Process Control* **2001**, *11*, 387–400.

(68) Chiang, L. H.; Braatz, R. D. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics and Intelligent Laboratory Systems* **2003**, *65*, 159–178.

(69) García-Muñoz, S.; Kourti, T.; MacGregor, J. F.; Mateos, A. G.; Murphy, G. Troubleshooting of an industrial batch process using multivariate methods. *Industrial & Engineering Chemistry Research* **2003**, *42*, 3592–3601.

(70) Alcala, C. F.; Qin, S. J. Reconstruction-based contribution for process monitoring. *Automatica* **2009**, *45*, 1593–1600.

(71) Huang, J.; Yan, X. Dynamic process fault detection and diagnosis based on dynamic principal component analysis, dynamic independent component analysis and Bayesian inference. *Chemometrics and Intelligent Laboratory Systems* **2015**, *148*, 115–127.

(72) Axsäter, S. *Inventory Control*, 3rd ed.; Springer, Cham, 2015.

(73) Lee, Y. H.; Cho, M. K.; Kim, S. J.; Kim, Y. B. Supply chain simulation with discrete–continuous combined modeling. *Computers and Industrial Engineering* **2002**, *43*, 375–392.

(74) Pundoor, G.; Herrmann, J. W. A hierarchical approach to supply chain simulation modelling using the Supply Chain Operations Reference model. *International Journal of Simulation and Process Modelling* **2006**, *2*, 124–132.

(75) Perea-Lopez, E.; Ydstie, B. E.; Grossmann, I. E. A model predictive control strategy for supply chain optimization. *Computers & Chemical Engineering* **2003**, *27*, 1201–1218.

(76) Schildbach, G.; Morari, M. Scenario-based model predictive control for multi-echelon supply chain management. *European Journal of Operational Research* **2016**, *252*, 540–549.

(77) Tsiakis, P.; Shah, N.; Pantelides, C. C. Design of multi-echelon supply chain networks under demand uncertainty. *Industrial & Engineering Chemistry Research* **2001**, *40*, 3585–3604.

(78) Ivanov, D. *Operations and Supply Chain Simulation with AnyLogic: Decision-Oriented Introductory Notes for Master Students*, 2nd ed.; E-Textbook, Berlin School of Economics and Law, 2017.

(79) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(80) Sterman, J. D. Modeling managerial behavior: misperceptions of feedback in a dynamic decision making experiment. *Management Science* **1989**, *35*, 321–339.

(81) You, F.; Grossmann, I. E. Design of responsive supply chains under demand uncertainty. *Computers & Chemical Engineering* **2008**, *32*, 3090–3111.

(82) Misra, S.; Kapadi, M.; Gudi, R. D.; Saxena, D. Resource optimization and inventory routing of the packaged liquefied gas supply chain. *Industrial & Engineering Chemistry Research* **2019**, *58*, 7579–7592.

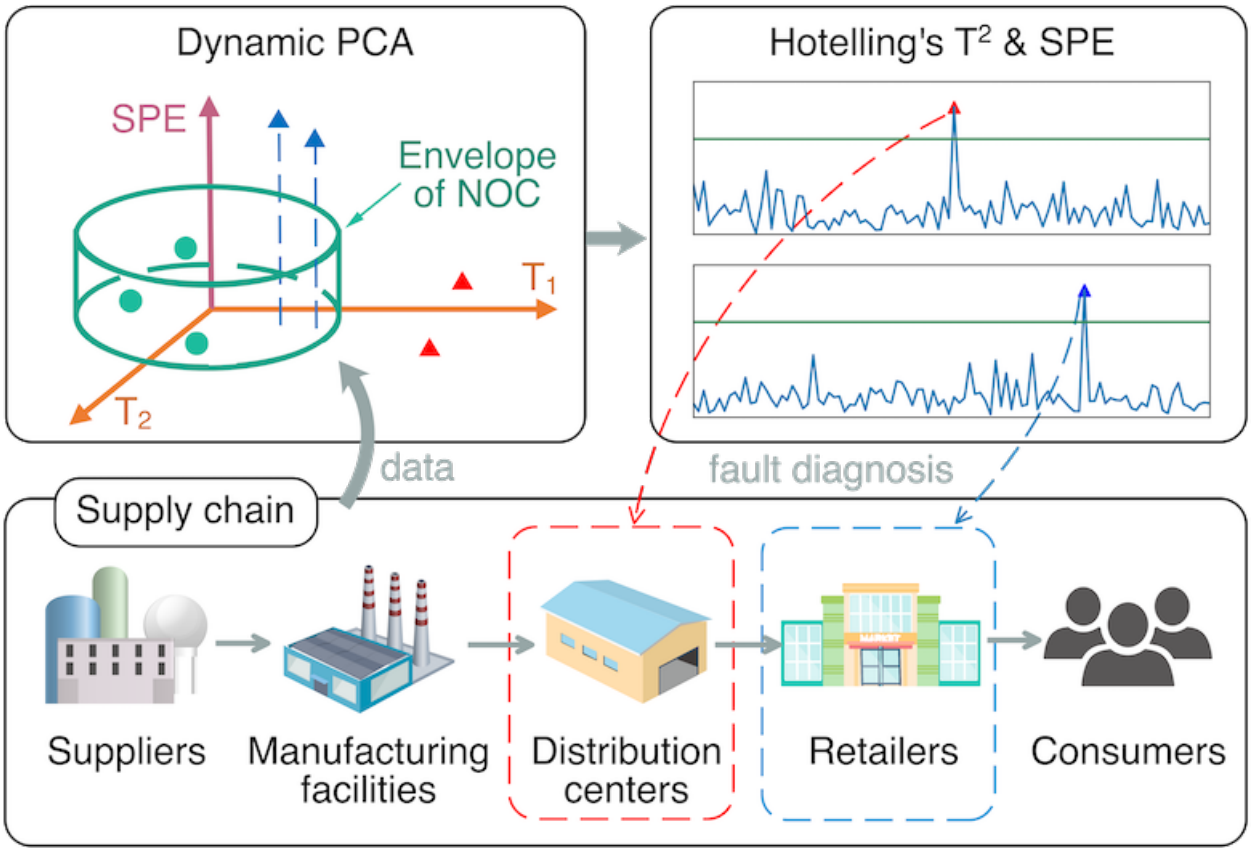


Figure 24: For Table of Contents Only.