Nonlinear data driven techniques for process monitoring

Michael C. Thomas
Louisiana State University and Agricultural and Mechanical College, mthomasmall@gmail.com

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses
Part of the Chemical Engineering Commons

Recommended Citation
Thomas, Michael C., "Nonlinear data driven techniques for process monitoring" (2014). LSU Master's Theses. 2554.
https://digitalcommons.lsu.edu/gradschool_theses/2554

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master’s Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
NONLINEAR DATA DRIVEN TECHNIQUES FOR PROCESS MONITORING

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Cain Department of Chemical Engineering

by

Michael Carl Thomas
B.E., Vanderbilt University, 2012
May 2014
To my family
ACKNOWLEDGEMENTS

I am grateful for Prof. Jose Romagnoli for his brilliant mentorship during my studies at LSU. I also like thank Dr. Flake and Dr. Chen for serving on my committee and enriching my experience as a graduate student. I also thank Greg Robertson, who played a pivotal role in the development of this research.

Finally, I am most grateful for the support of my parents for helping me to get where I am now.
# Table of Contents

Acknowledgements .................................................................................................................. iii

List of Tables ................................................................................................................................. vi

List of Figures ................................................................................................................................. vii

Abstract ......................................................................................................................................... ix

1. Introduction ................................................................................................................................. 1
   1.1 Thesis Aim ............................................................................................................................... 4
   1.2 Thesis Organization .................................................................................................................. 5

2. Background ................................................................................................................................ 7
   2.1 Principal Component Analysis in Process Monitoring .......................................................... 7
   2.2 The Self-Organizing Map ...................................................................................................... 8
   2.3 SOM Visualization Tools for Data Analysis .......................................................................... 10
   2.4 Dimensionality Reduction Example ...................................................................................... 13
   2.5 Nonparametric Density Estimation ....................................................................................... 14
   2.6 Gaussian Mixture Models for Density Estimation ................................................................. 18

3. Process Monitoring Methods ...................................................................................................... 20
   3.1 Univariate Case ...................................................................................................................... 20
   3.2 PCA in Process Monitoring .................................................................................................. 21
   3.3 SOM in Process Monitoring .................................................................................................. 25
   3.4 MSOM in Process Monitoring .............................................................................................. 26
   3.5 SOM Methods for Soft Sensing ........................................................................................... 31
   3.6 Literature Review .................................................................................................................. 33

4. Case Study: Tennessee Eastman Process ..................................................................................... 36
   4.1 Process Description ................................................................................................................ 36
   4.2 Fault Descriptions ................................................................................................................ 37

5. Results and Discussion ............................................................................................................... 39
   5.1 TEP Fault Detection .............................................................................................................. 39
   5.2 TEP Fault Identification ........................................................................................................ 41
   5.3 TEP Fault Diagnosis ............................................................................................................. 43
   5.4 TEP Soft Sensing .................................................................................................................. 48
6. CONCLUSIONS .................................................................................................................. 51

REFERENCES .................................................................................................................... 53

APPENDIX A: ADDITIONAL TENNESSEE EASTMAN PROCESS DETAILS ....................... 57

VITA ........................................................................................................................................ 59
LIST OF FIGURES

Figure 1: The vectors comprising an SOM spread out over a set of data. During the training process, some map vectors cluster together according to the data while others are spread over the empty space between clusters. ..................................................................................................................11

Figure 2: Visualizations of the SOM in Figure 1. The dark borders between white areas on the U-matrix are the representation of the distance between the clusters of the data. The component planes for variable 1, 2, and 3 show the values of different variables at each map node. ..................................................................................................................12

Figure 3: (a) Projecting the trajectory of the data to the SOM allows us to follow the data through transitions to different operating regions. (b) A hit diagram on the map allows us to view the location of groups of data on the map. In (b) the circles represent group 1, the triangles are group 2, and squares are for group 3. .................................................................................13

Figure 4: Plot of Fisher's iris data (Fisher 1936). .................................................................................................................................15

Figure 5: Fisher iris data after PCA dimensionality reduction to two principal components, PC1 and PC2. .................................................................................................................................15

Figure 6: U-matrix and component planes of the Fisher iris data set. .................................................................................................16

Figure 7: Hit diagram of Fisher iris data on the U-matrix. ........................................................................................................................16

Figure 8: Estimated PDF using the Scott Rule. .................................................................................................................................18

Figure 9: SOM based GMM PDF estimation on a parabolic set of data ...............................................................................................19

Figure 10: Shewhart chart .........................................................................................................................................................21

Figure 11: 1SOM process monitoring ........................................................................................................................................26

Figure 12: MSOM process monitoring ........................................................................................................................................27

Figure 13: Illustration of the residual analysis ....................................................................................................................................29
Figure 14: 1SOM based soft sensing ................................................................. 32

Figure 15: Schematic for MSOM soft sensing. .................................................. 33

Figure 16: Process schematic with control scheme ..................................... 37

Figure 17: Plots of the PDF of the state of the TEP as it progresses through fault 1. .......... 40

Figure 18: Comparative analysis for contribution plots ...................................... 42

Figure 19: (a) The U-Matrix of the SOM used for 1SOM fault diagnosis with (b) a hit diagram of the training data projected onto the map. ................................................................. 44

Figure 20: The trajectory from normal operation to (a) Fault 1, (b) Fault 7, and (c) Fault 13. ................................................................. 46

Figure 21: Data classification using (a) MPCA, (b) 1-SOM, and (c) MSOM on the testing data set with time colored according to class. Solid lines of color indicate a consistent classification while discontinuous lines and dots indicate an inconsistent classification. ................................................................. 47

Figure 22: Prediction of the A in the reactor feed as and the residuals of both prediction methods during Fault 8. ................................................................. 49

Figure 23: Trajectory plot of a period of 50 points of data following point 4700 of Fault 8. ................................................................. 50
ABSTRACT

The goal of this research is to develop process monitoring technology capable of taking advantage of the large stores of data accumulating in modern chemical plants. There is demand for new techniques for the monitoring of non-linear topology and behavior, and this research presents a topological preservation method for process monitoring using Self Organizing Maps (SOM). The novel architecture presented adapts SOM to a full spectrum of process monitoring tasks including fault detection, fault identification, fault diagnosis, and soft sensing.

The key innovation of the new technique is its use of multiple SOM (MSOM) in the data modeling process as well as the use of a Gaussian Mixture Model (GMM) to model the probability density function of classes of data. For comparison, a linear process monitoring technique based on Principal Component Analysis (PCA) is also used to demonstrate the improvements SOM offers. Data for the computational experiments was generated using a simulation of the Tennessee Eastman process (TEP) created in Simulink by (Ricker 1996). Previous studies focus on step changes from normal operations, but this work adds operating regimes with time dependent dynamics not previously considered with a SOM.

Results show that MSOM improves upon both linear PCA as well as the standard SOM technique using one map for fault diagnosis, and also shows a superior ability to isolate which variables in the data are responsible for the faulty condition. With respect to soft sensing, SOM and MSOM modeled the compositions equally well, showing that no information was lost in dividing the map representation of process data. Future research will attempt to validate the technique on a real chemical process.
1. Introduction

With the advent of the information age almost every area of human civilization has been touched in some way by the vast computing power that is commonly available. A spillover of the increasing digitization of human activity is the creation of enormous databases and online logging of exabytes \((10^{18})\) of data (Vastag 2011). Outside of this titanic buildup of data is a vast gap between data storage capabilities and people’s ability to understand and interpret patterns from such large amounts of data.

Chemical engineers have not ignored this trend of digitization. For decades, engineers have used data from chemical plants to monitor the state of processes and maintain safe and efficient operation. Nearly all modern processes operate in a highly integrated and automated environment with a distributed control system (DCS) to maintain a consistent product quality and continuously adjust for disturbances entering in the process from the surrounding environment, with the human operator acting as a supervisor. Controllers can adjust for many different disturbances, but occasionally the process is perturbed in a way controllers cannot handle adequately in an event called a fault. A fault is defined as an unpermitted deviation of at least one characteristic property or variable of the system. The detection, diagnosis, and correction faults, often with the assistance of computer algorithms, is the domain of process monitoring (Chiang 2001).

Under faulty conditions, the operator is required to rapidly assess the situation to determine the root causes and take necessary and appropriate corrective actions. Failure to respond properly can have large economic, safety, and environmental impacts, as occurred in major chemical accidents such as BP’s Deepwater Horizon or Occidental Petroleum’s Piper Alpha accident. In addition to infrequent major disasters, other less serious faults occur much
more often and result in occupational injuries and economic losses. The decisions taken by operators may depend on factors such as differences in plant operation between shifts and the skills mental condition and time available for the action to be implemented. The development of technology for process monitoring is critical to assist operators, process engineers and managers in handling abnormal situations.

Process monitoring tasks include fault detection, fault identification, fault diagnosis, and process recovery. Fault detection recognizes a deviation from the normal operating regime using process measurements. Fault identification helps personnel identify the fault by finding the measured variables most related to the fault. Fault diagnosis determines the root causes of the fault and process recovery is the manual correction for the effect of the fault. Process recovery is removing the effect of the fault through the intervention of human operators. It is not necessary to implement all process monitoring tasks at all times. A fault may be detected while fault identification is used by plant operators to diagnose the problem. Alternatively, a model of the process data may be good enough that fault identification is unnecessary as any operating condition could be easily diagnosed (Chiang 2001).

Process monitoring techniques can be divided into three classes: knowledge based methods, qualitative model based methods, and process history based approaches (Venkatasubramanian 2003). The first two require first principles or empirical models which are generally difficult to create in a modern industrial plant and can quickly lose their real-time applicability due the complexity of the solution. Analytical methods create a prediction of the process with a model often derived from first principles. Knowledge-based methods are mostly based on causality and expert systems. Most applications of these systems are for smaller input-output systems. For large-scale systems, these techniques require a detailed model or highly
specialized knowledge of a process. Recent trends in process monitoring research have favored process history based approaches based on machine learning and multivariate statistics or “big data” approaches which take full advantage of the large repositories of process data and can be simpler to apply to real plants. Data-driven methods are derived directly from process data without using any underlying laws. Popular data based techniques include principal components analysis, partial least squares, and neural networks. A comprehensive review of process monitoring techniques can be found in (Venkatasubramanian 2003) and (Qin 2012).

Another process monitoring task which has many overlaps with fault detection and diagnosis is soft sensing, sometimes known as inferential monitoring. A soft sensor is a predictive model for one or several variables in a plant usually composed of a software program. Often used to determine process variables found at low sampling rates or with a large time from off line analysis, such as composition measurements by gas chromatography, an accurate soft sensor can save time and money by instantly a desired measurement from process data and reducing or eliminating the need for certain expensive measuring instruments (Dong 1995). In addition to generating missing measurements, soft sensors can also be used to interpret the process state and assist in process monitoring through sensor fault detection and measurement reconstruction (Kadlec 2009)

Also in common with fault detection and diagnosis, soft sensors can be separated into two broad categories: model driven and data driven models based on historical data. Model driven techniques attempt to reconstruct the dynamics of a measured quantity from the chemical and physical laws that govern the system and suffer from drawbacks similar to model based fault detection mentioned above. One of the drawbacks to model based soft sensors is that they are often developed during the planning and design of plants and are focused more on the ideal
steady states of processes. Data based soft sensing includes statistical methods like partial least squares (PLS) as well as artificial intelligence methods like neural networks and SOM.

Utilizing large amounts of plant data for fault diagnosis is a challenge in itself. Plants often have the capability to record plant data for every second of years of operations leading to numbers of data points in the millions. For an engineer attempting to create a program to classify the state of the process, the data initially contains a paradoxically small amount of information because the state of the process (normal or faulty) is almost never recorded. Furthermore, the monitoring of complex plants involves the recording many process variables, some containing complicated nonlinear relationships with other variables and others which supply no information at all. Accompanying the development of intelligent process monitoring strategies is the creation of ingenious preprocessing techniques to define the normal operating conditions of a plant from unlabeled historical process data.

Research contained in this thesis was done in collaboration with Robertson (2014).

1.1 Thesis Aim

Following the discussion in the previous section, the aim of this thesis is the development, implementation, and testing of an automated data-driven nonlinear framework for process analysis and supervision. Towards that goal, this thesis presents new techniques for process monitoring, including traditional process monitoring tasks as well as methods for data preprocessing and soft sensing. The approaches are tested using the simulated data from the Tennessee Eastman process (TEP). The new technique gives engineers useful information about the topology of the data and readily lends itself to automated process monitoring.

Specifically in this thesis:
• The current abilities of the nonlinear topology preservation technique of Self Organizing Maps (SOM) are expanded towards the development of a complete SOM-based fault diagnosis and monitoring framework. This includes the development and implementation of novel detection variable contribution and diagnosis tools within SOM framework.

• A novel SOM-based soft sensor for multivariable prediction is proposed and tested and its capabilities are compared to traditional techniques.

• A complete set of new tools were implemented into the Matlab environment which will form the main components of an agent-based process supervision framework being developed within the PSE group.

• The techniques are tested and compared using a simulated industrial case study. The Tennessee Eastman process is a well-studied benchmark process first created by the Tennessee Eastman Chemical company and includes a set of 20 faults, 8 of which will be considered in this work, that simulate faults in a real plant. This standard simulation enables comparison with standard process monitoring techniques.

1.2 THESIS ORGANIZATION

This thesis is organized in a number of chapters as follows:

• In Chapter 2, the data-driven techniques used in this work, including PCA, SOM, as well and probability density estimation are introduced. The visualization capabilities of SOM are discussed and will be used in later chapters.

• Chapter 3 explains how the methods described in Chapter 2 are utilized for process monitoring tasks. More importantly, new developments on SOM are
introduced leading to a complete monitoring and fault diagnosis strategy for process supervision.

- Chapter 4 introduces the Tennessee Eastman process, the source of the data used in the computational experiments done in this research.

- Chapter 5 covers the results of performing fault detection, fault identification, fault diagnosis, and soft sensing with data generated by the Tennessee Eastman process. The results illustrate the improvement made by the proposed strategies developed in this thesis.

- Chapter 6 summarizes the conclusions of the research and promising new avenues for future research.
2. BACKGROUND

This section introduces the data-driven techniques utilized in this work. First, the typical linear distance-preservation technique, Principal Component Analysis (PCA), is introduced. Next a nonlinear topological preservation technique based on Self Organizing Maps (SOM) is presented along with its visualization capabilities. An example of PCA and SOM dimensionality reduction will then be presented to illustrate the use of the two techniques. Last, techniques for estimating the probability density function are introduced, along with a method for using SOM in density estimation.

2.1 PRINCIPAL COMPONENT ANALYSIS IN PROCESS MONITORING

PCA is a linear distance-preservation technique which determines a set of orthogonal vectors which optimally capture the variability of the data in order of the variance explained in the loading vector directions (Chiang 2001). PCA is feature extraction and dimensionality reduction. Some monitored variables can generate highly noisy data with few significant variations, and PCA is adept at ignoring the noise and capturing a signal in data.

Given a set of $n$ observations and $m$ process variables in the $n \times m$ matrix $X$ with covariance $S$, the loading vectors are determined from an eigenvalue decomposition of $S$:

$$S = \frac{1}{n-1}X^TX = \Lambda V^T$$

Where $\Lambda$ is the diagonal matrix containing the non-negative real eigenvalues of the covariance matrix in order of decreasing magnitude, and $V$ holds their corresponding eigenvectors. The $i$th eigenvalue in the diagonal of $\Lambda$ indicates the fraction of variance stored in the $i$th eigenvector. In order to reduce the misclassification rate, it is often desirable to remove directions that may contain little useful information or simple statistical noise. In PCA this is achieved by selecting
the columns of the loading matrix which correspond to the $a$ largest eigenvalues $P \in R^{m \times a}$. The projections of the observation in $X$ into the lower dimensional space, also known as the score matrix, can be found from:

$$T = XP$$

An engineer can remove directions that primarily contain noise by selecting only the first $a$ principal components and reduce them to a space with smaller dimensionality, $T$. The smaller space composed of latent variables that are formed from combinations of the original data set and more sensitive to process variations.

Conversely, the data can be mapped back into the $m$-dimensional observation space producing data stripped of unnecessary variance, $\hat{X}$:

$$\hat{X} = TP^T$$

The space spanned by is sometimes referred to as the score space. The variance removed by the model, also called the residual space $E$, can be found as

$$E = (X - \hat{X})$$

The residual matrix captures the variations in the observed space spanned by the loading vectors associated with the $m - a$ smallest eigenvalues. The information in $E$ has a small signal to noise ratio, and the removal of this space from $X$ can produce a more accurate representation of the process, $\hat{X}$.

In Section 3.1 the projections of the process variables into the score space will be later used to generate statics for process fault detection, identification, and diagnosis.

2.2 **The Self-Organizing Map**

Self-organizing maps (SOMs), also known as Kohonen Network, are a type of neural network used to visualize complicated, high-dimensional data. SOM may be described formally
as nonlinear, ordered, smooth mapping of high dimensional input data manifolds onto the elements of a regular, low dimensional array. It simultaneously performs vector quantization and topographical preservation while representing complex multivariate data on a two dimensional grid (Kohonen 2001).

A map is a graph composed of an array of nodes connected together in a rectangular or hexagonal grid. Each node, called prototype vector, models the observation vectors in the data space. These prototypes have a representation in the input (observation) and output (latent) space. The closest prototype vector to a data point in the observed space, known as the Best Matching Unit (BMU), can represent that data point on the map. The BMU of \(i^{th}\) data point \(v_i\) is found by finding the closest prototype vector \(m_k\) according to:

\[
BMU_i = \text{arg min} \left( \delta (m_k, v_i) \right), \forall k
\]

To implement the SOM procedure, first the map shape is selected. Then the prototypes vectors’ positions are initially embedded in the data space along the PCA hyperplane. Next, the map is trained to capture curves in the manifold. The prototype vectors’ positions are updated using each data point from a training set according to the formula:

\[
m_k(t + 1) = m_k(t) + \alpha(t)h_{k,BMU_i}(v_i(t) - m_k(t))
\]

Where \(t\) is the discrete time coordinate of the mapping steps and \(\alpha\) is the monotonically decreasing learning rate. The scalar \(h_{k,BMU_i}\) denotes a neighborhood kernel function centered at the BMU. Data vectors are matched to the prototype vectors with the smallest Euclidean distance from the data vector of interest. A neighborhood kernel \(h_{k,BMU_i}\) centered at \(m_k(t)\) is usually chosen in the Gaussian form:

\[
h_{m_k,BMU_i} = \exp \left( \frac{\|m_k - BMU_i\|^2}{2\sigma^2} \right)
\]
σ(t) denotes the monotonically decreasing width of the kernel that allows for a regular smoothing of the prototypes. The algorithm continues to make passes over the data updating the locations of the prototype vectors and terminates after some predefined number of time steps have passed or prototype updating becomes negligible (Kohonen 2001).

A popular alternative to the update rule outlined above is the SOM batch training algorithm. During the batch training algorithm, the training data is passed through once to determine which data vectors lie in each prototype’s neighborhood. Then each map unit is replaced by the weighted average of the data vectors that were in its neighborhood, where the weighting factors are the neighborhood kernel values. In the notation used previously:

\[ m_i(t + 1) = \frac{\sum_{j=1}^{n} h_{i,c(j)}(t)x_j}{\sum_{j=1}^{n} h_{i,c(j)}(t)} \]

Where c(j) is the BMU of the sample vector \( x_j \), \( h_{i,c(j)} \) is the neighborhood kernel defined above, and \( n \) is the number of data vectors used for training. This variant of the training algorithm is often used because it is much faster to calculate and the results are typically just as good or better than the sequential training algorithm (Kohonen 2001).

At the end of training, in contrast to a PCA model which creates hyperplane oriented to capture the most variance in the data, a trained SOM is like an elastic net spread through multidimensional space designed to capture the dominant topology and clustering characteristics of the data. During training, map vectors can associate with dense groups of data, compressing their representation. The next section will explore the visualization techniques that allow engineers to observe the shape and distribution of map nodes in the higher dimensional space.

**2.3 SOM VISUALIZATION TOOLS FOR DATA ANALYSIS**

A 2-D SOM offers excellent visualization tools for data exploration. The approach to high dimensional data analysis with the SOM consists of a set of 2-D visualization displays
The set of visualization displays are constructed by projecting data onto the map. For instance, the displays allow identifying the shape of the data distribution, cluster borders, projection directions and possible dependencies between the variables. Common visualization discussed here are component places, distance matrices, and projections as proposed by Kaski (1997), Vesanto (1999), Vesanto (2002) and Himberg (2001). The data and map shown in Figure 1 will be used to illustrate SOM data visualizations.

Figure 1: The vectors comprising an SOM spread out over a set of data. During the training process, some map vectors cluster together according to the data while others are spread over the empty space between clusters.

2.3.1 Component Planes

A component plane displays the value of individual variables at each prototype vector in the map. Each component plane is associated to one variable in the observation space. The value of the variable at all map nodes is visualized using different colors which allows the user to visually identify possible dependencies between variables (Vesanto 1999); (Lampinen 2000). The dependencies between variables can be seen as similar patterns in identical locations on the component planes. In that sense, the SOM reduces the effect of noise and outliers in the
observations and, therefore, may actually make any existing dependency clearer than in the original data. Figure 2 gives an example of component planes created using the data in Figure 1.

2.3.2 Distance Matrices

A distance matrix visualizes on the SOM array the distances between each prototype vector and its closest neighbors. In a distance matrix, distances are encoded into gray levels or colors and each unit on the array is colored according the distances between neighbors. The most widely used distance matrix for the SOM is the Unified Distance Matrix, U-matrix where all pairwise distances are visualized. In a U-matrix, the dominant clustering structure can be seen as areas with small distances within and separated by large distances. As in the basic U-matrix, visualization of the clusters can be greatly improved by augmenting the distance matrix with an

Figure 2: Visualizations of the SOM in Figure 1. The dark borders between white areas on the U-matrix are the representation of the distance between the clusters of the data. The component planes for variable 1, 2, and 3 show the values of different variables at each map node.
additional entry between each prototype vector and each of its neighbors. Alternatives to the basic U-matrix and other distance matrices are reported in the literature (Kraaijveld 1995; Moutarde and Ultsch, 2005).

2.3.3 Projections

Finally, projections onto the map are very important visualization tools. Highlighting the BMU of data points represents them on the map. Two important figures derived from projections are hit diagrams and trajectories. Hit diagrams project a collection of data points to show a region belonging to a class. Trajectories take a series of time and project them on to the map, connecting them with lines. In process monitoring, the trajectory makes it possible to visually indicate the current state of the process and observe how that state has been reached. Figure 3 gives examples of SOM trajectory plots and hit diagrams.

2.4 Dimensionality Reduction Example

The use of SOM and PCA for dimensionality reduction will be shown using Fisher’s iris data set (Fisher 1936). The data contain measurements of the dimensions iris sepals and flower

![Figure 3](image.png)

**Figure 3:** (a) Projecting the trajectory of the data to the SOM allows us to follow the data through transitions to different operating regions. (b) A hit diagram on the map allows us to view the location of groups of data on the map. In (b) the circles represent group 1, the triangles are group 2, and squares are for group 3.
petals for three different types of irises. There are four measured variables and three classes, each consisting of 50 observations. There data are plotted in Figure 4.

As the data is four dimensional, all four measurements cannot be viewed simultaneously. PCA can be used to create a simple dimensionality reduction to view how the data separate. Figure 5 shows a plot of the data set after it has been reduced to two dimensions (PC1 and PC2) and colored according to class. It is easy to see that the three species of flowers form three groups in the data, with the Setosa variety being especially different from Versicolor and Virginica. By summing the largest two eigenvectors and dividing by 4, the dimensionality of the original data, it can be calculated that PC1 and PC2 have captured 77% of the variance of the original data set.

Performing an SOM projection on the same data set yields the plots given in Figure 6. Similar to the PCA projection, the SOM analysis shows evidence of three distinct groups of data. First, the top of the map can be seen to be distinct from the rest of the map based on the distances between map nodes visualized in the U-matrix. Second, and more subtly, The Petal Length and Petal Width component planes, in addition to distinguishing the group at the top, show a difference in the measurement values between the very bottom of the map and the middle. The hit diagram in Figure 7 shows that the projection of the data onto the map separates the all three groups at least as well as PCA.

2.5 Nonparametric Density Estimation

Conventional methods used to define the normal region for a chemical process assume that the process measurements will conform to an assumed distribution, usually the Gaussian distribution. However, not all sets of measurements fit into the Gaussian distribution, and nonparametric methods, which require no knowledge of a population’s mean or standard
Figure 4: Plot of Fisher's iris data (Fisher 1936).

Figure 5: Fisher iris data after PCA dimensionality reduction to two principal components, PC1 and PC2.
Figure 6: U-matrix and component planes of the Fisher iris data set.

Figure 7: Hit diagram of Fisher iris data on the U-matrix.
Kernel density estimation is the most widely used nonparametric data-driven technique used to find a nonparametric estimation of the probability density function for a set of data. The multivariate kernel estimator with kernel $K$ is defined as:

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

Where $h$ is the window width, also called the smoothing parameter or bandwidth, and $d$ is the number of dimensions in the data (Silverman 1986). The kernel function used here was the multivariate normal distribution given by this equation:

$$y = f(x, \mu, \Sigma) = \frac{1}{\sqrt{|\Sigma|(2\pi)^d}} \exp\left(-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)^\prime\right)$$

Where $x$ and $\mu$ are 1 by $d$ vectors and $\Sigma$ is a $d$ by $d$ symmetric positive definite matrix. The kernel function $K$ must have the property that:

$$\int_{-\infty}^{\infty} K(x) \, dx = 1$$

A simple way to imagine kernel density estimation is to think of it as the sum of bumps, defined by the kernel function, centered at each of the data points in the set of interest. It is generally accepted that the window width $h$ is much more important in determining the accuracy of a density estimate than the nature of the kernel function (Chen 1996).

Many elaborate methods for determining the window width have been developed. Here, for simplicity, the Scott Rule is used to calculate an optimum window width in each of the Gaussian PDFs estimated (Scott 1979):

$$h_n = \frac{3.49s}{n^{1/3}}$$
Figure 8, gives a simple example where a probability distribution of a group of data randomly generated around a parabola was estimated using kernel density estimation and the Scott Rule to calculate an optimum window width.

![Figure 8: Estimated PDF using the Scott Rule.](image)

### 2.6 Gaussian Mixture Models for Density Estimation

An alternative method for calculating estimating the probability density function of a set of data uses a well-trained SOM. As explored by Holmstrom, (1993) and Heskes, (2001), a SOM can be used to create a Gaussian mixture model (GMM) using SOM’s vector quantization abilities which provide a compressed representation of a group of data. For an SOM with $M$ map nodes, the GMM is the weighted sum of $M$ component Gaussian densities as given by the equation:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(x|\mu_i, \Sigma_i),$$

Where $x$ is a $D$-dimensional continuous-valued data vector (i.e. measurement or features), the $w_i$ are the mixture weights, and $g(x|\mu_i, \Sigma_i)$, are the component Gaussian densities. Each component density is a $D$-variate Gaussian function of the form,

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_i)'\Sigma_i^{-1}(x - \mu_i) \right\}$$
With mean vector $\mu_i$ and covariance matrix $\Sigma_i$. The mixture weights satisfy the constraint that $\sum_{i=1}^{M} w_i = 1$. The complete model is parameterized by the mean vectors, covariance, and mixture weights from all component densities (Reynolds, 2008), which can give GMM modeling a higher degree of accuracy compared to heuristic Scott Rule.

Here, we use SOM map vectors as the centers of the Gaussian kernels instead of the data, a feature useful for very large sets of multivariate data. A Gaussian mixture model is formed by summing kernels centered on the map nodes along with a regularization term (Heskes, 2001). The advantage of using a GMM within the SOM for discrimination comes from its ability to perform Kernel Density Estimation (KDE) using a predefined number of kernels, $M$. Figure 9 shows a probability density function (PDF) estimated by an SOM based GMM, which more accurately represents the shape of the data compared to the PDF estimated with conventional KDE method in Figure 8. A boundary for the group of data can be created from any of the contours in the PDF by selecting a probability threshold.

![Figure 9: SOM based GMM PDF estimation on a parabolic set of data](image)
3. PROCESS MONITORING METHODS

This section explains how the methods described in Chapter 2 can be utilized in the process monitoring tasks. A key issue associated with the multivariate statistical monitoring schemes is the construction or definition of process normal operating region, which can be thought of as the acceptable level of variation around the target values. Statistics are generated to quantify the degree of belongingness in the normal region and are employed as an indicator of the process performance and product quality.

In univariate process monitoring, the normal operating region is constructed based upon the assumption of normality and independence. Although a similar approach has been extended to multivariate case using Principal Components Analysis (PCA), the underlying assumptions appear to be restrictive and inappropriate for complex processes. Still, one of the main advantages of PCA is its simplicity and the range of tools it provides which covers all fault detection, identification, and diagnosis tasks discussed above. Our intention is to develop a nonlinear approach that will mimic PCA by defining similar measures for process monitoring and fault detection.

Section 3.1 introduces a basic univariate process monitoring technique. Section 3.2 briefly overviews PCA tools and Section 3.2 expands these tools within the SOM framework. Section 3.3 introduces a new method for process monitoring using multiple maps. Section 3.4 explains how SOM methods can be used for soft sensing.

3.1 UNIVARIATE CASE

The simplest and most common technique for process monitoring is a Shewhart control chart, where a single variable is plotted against time (Shewhart 1931). The target is the centerline, and upper and lower limits define a normal region for acceptable variation from the
target, as shown in Figure 10. The upper and lower limits represent 3 sigma (sample standard deviation) from the sample, which is defined as:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1}}$$

Here is an individual sample, is the mean (arithmetic average) of the data and n is the total number of measurements. The 3 sigma control limit can also be called 99.73% confidence level. It means that 99.73% possibility the normal operation data will fall within the normal operation region. When the measurement above or below the 3 sigma control limits, the process is said to be in a faulty state. An important assumption of this method, however, is that the data follow normal distribution.

![Shewhart chart](image)

Figure 10: Shewhart chart

### 3.2 PCA IN PROCESS MONITORING

Given the large number of variables monitored in a large plant, it is important to develop techniques that monitor more than one variable at a time in order to execute effective process monitoring. Here, the application of PCA to the three process monitoring tasks, fault detection, identification, and diagnosis, is explained. Besides its statistical ability to detect large deviations
from normal operations, PCA is also useful for feature extraction and dimensionality reduction. Some monitored variables can generate highly noisy data with few significant variations, and PCA is adept at ignoring the noise and capturing a signal in data.

Besides the techniques presented here, PCA has been applied to process monitoring other forms which more specifically take into account dynamics and non-linearity.

3.2.1 PCA Fault Detection

Using the projection of the process data found in Section 2.1, the Hotelling’s $T^2$ statistic is used to detect faults for multivariate process data. Given the $1 \times a$ projection of a $1 \times m$ observation row vector into the score space $t$, the $T^2$ statistic can be calculated:

$$T^2 = t^T \Lambda_a^{-1} t$$

Where $\Lambda_a$ is the diagonal matrix containing the first $a$ largest eigenvalues in order of decreasing magnitude. The $T^2$ statistic threshold for normal operation can be represented as an ellipsoid with a $1 - \alpha$ statistical significance using the $T^2_\alpha$ threshold and a level:

$$t^T \Lambda_a^{-1} t \leq T^2_\alpha$$

When the actual covariance matrix is estimated from the sample covariance matrix, the $T^2_\alpha$ threshold is found from the $F$ distribution by the equation:

$$T^2_\alpha = \frac{\alpha(n-1)(n+1)}{n(n-a)} F_{a}(a, n-a)$$

Where $n$ is the number of data points and $a$ is the number of principal components. The $T^2$ statistic is highly sensitive to inaccuracies in the PCA space corresponding to the smaller eigenvalues (Chiang 2001) because it directly measures scores corresponding to the smaller singular values. The $m-a$ smallest principal components can be monitored using a supplement to fault detection based on the $T^2$ statistic, the $Q$ statistic. Also known as the squared prediction error (SPE), the $Q$ statistic can be computed to measure how well each sample conforms to the
PCA model and the amount of variation not captured by the principal components retained in the model (Wise 1996). The part of the observation space corresponding to the $m - a$ smallest singular values, which correspond to the $m - a$ ignored by the reduction in dimension, can be monitored using the Q statistic:

$$r = (I - PP^T)x \quad Q = r^Tr$$

Where, $r$ is the residual vector, a projection of observation $x$ into the residual space. The threshold for the Q statistic is given by the following equations from Wise and Gallagher, (1996).

$$Q_\alpha = \Theta_1 \left[ \frac{c_\alpha \sqrt{2\Theta_2 h_0^2}}{\Theta_1} + \frac{\Theta_2 h_0 (h_0 - 1)}{\Theta_1^2} + 1 \right]^{\frac{1}{h_0}}$$

where $c_\alpha$ is the standard normal deviate corresponding to the upper $1 - \alpha$ percentile and

$$h_0 = 1 - \frac{2\Theta_1 \Theta_3}{3 \Theta_2^2}$$

and

$$\Theta_i = \sum_{j=a+1}^{N} \lambda_j^i \quad \text{for } i = 1 - 3$$

where $\lambda_i$ are the eigenvalues of the covariance matrix of $X$ and $N$ is the total number of principal components, which is either the number of variables or samples in $X$, whichever is smaller. The thresholds defined for the $T^2$ and $Q$ statistics set the boundaries for normal variations and random noise, and a violation of the threshold indicates that the random noise has changed (Chiang et al 2001) and a statistically separate operating status reached.

3.2.2 PCA Fault Identification
Contribution plots are a PCA approach to fault identification based on determining the process variable responsible for the faulty condition. For a $T^2$ violation, follow the approach in Chiang et al. (2001) in which the contribution of each variable is calculated according to:

$$cont_{i,j} = \frac{t_i}{\lambda_i} p_{i,j} (x_j - \mu_j)$$

Where $p_{i,j}$ is the $i^{th}$ element of the loading matrix $P$. The total contribution of the $j^{th}$ process variable is determined from:

$$CON_{Tj} = \sum_{i=1}^{r} (cont_{i,j})$$

The $CON_{Tj}$ for all process variables are then plotted on a single graph in order to decide which variables are responsible for the faulty condition.

3.2.3 PCA Fault Diagnosis – Discriminant Analysis

To determine the root cause of a given fault, data previously collected from out of control operations can be grouped and used by a pattern classification scheme to diagnose future faults. Pattern classification proceeds via three steps: feature extraction, discriminant analysis, and maximum selection. Feature extraction is the process of isolating the important trends in the data and removing noise which could disrupt classification. PCA performs feature extraction in the projection into the score space using $P$ in the fault detection step. The pattern classification system assigns an observation $x$ to class $i$ if

$$g(x)_i > g(x)_j \quad \forall j \neq i$$

Where, $g_j$ is the discriminant function for class $j$. $g(x)_i$ can be approximated using the $-T^2$ statistic assuming the probability of each class is the same and the total amount of variability in each class is the same.
PCA, and similar techniques that use matrix decomposition, assume linearity and preserve distances. Preserving topologies with non-linear techniques preserves the order of data, which may aid our classification goals. In the next section, SOM as a topological preservation method is discussed.

3.3 SOM IN PROCESS MONITORING

As described in the previous chapter, SOM provides a powerful nonlinear projection tool for data mining and analysis as compared to PCA and other linear techniques. However, capabilities of SOM for fault detection and diagnosis are limited without the development of methodologies that use SOM. In the following sections the current abilities of SOM are expanded towards the development of a complete SOM-based fault diagnosis and monitoring framework.

1-SOM process monitoring, as outlined in Figure 11 trains a single map to fit all training data, both normal and faulty, using the training algorithm outlined in Section 2.2. Fault detection and diagnosis occur in one step where the incoming data are matched to their BMUs. The class of incoming data is determined by the class of the new data’s BMU. For example, if most of the training data associated with a particular BMU belong to group 1, new points projected onto the map during process monitoring that are associated with that BMU will be assigned to group 1. All previous applications of SOM to fault detection and diagnosis are of this variety.

Using one SOM for process monitoring offers fewer capabilities compared to a monitoring scheme using multiple PCA models or multiple SOMs. The use of a single SOM means that all data, normal and faulty, is projected onto the same map, which does not allow the model’s error or differences between several models to be fully exploited for fault detection or identification. Because 1-SOM uses only one map, it does allow for the visual tools presented in
Section 2.3 to be used for detailed analysis of the structure and distribution of the data. The visualization abilities of 1-SOM are extremely helpful for the pre-processing of unlabeled plant data and viewing the clustering structure of a data set.

Figure 11: 1SOM process monitoring

3.4 MSOM IN PROCESS MONITORING

Figure 12 illustrates the overall monitoring structure. The approach first trains maps to the separate operating regimes or fault conditions identified in historical data. The maps then serve one of two general purposes: data analysis or classification. Data analysis includes using visualizations, such as component planes, hit diagrams, as well as unsupervised clustered maps to identify variables important to clusters in the data. Data analysis is used to select variables and label clusters so that SOM maps can be trained and used for the classification of new data points.
3.4.1 MSOM Fault Detection

For fault detection, a GMM is created from an SOM trained by the method discussed in Section 2.2 using data from normal operation of the process to create a non-parametric model proportional to the probability distribution of the data. Here, the discriminate function used is:

\[ g(x)_i = \log(p(x|\lambda)) \]

Therefore, a normal point will have a small positive value and a faulty point will have a larger positive value. The main advantage of a density estimation approach over parametric approaches

![Diagram of MSOM process monitoring](image)

Figure 12: MSOM process monitoring
lies in density estimation’s ability to describe irregular data shapes caused by nonlinearities, as opposed to the hyper ellipsoid shape assumed by the Hotelling’s $T^2$ statistic or linear assumptions of the Q statistic.

Since nonparametric points do not assume a distribution, the thresholds for the normal region must be determined. Below this threshold, the process is considered at normal operation and above the bound the process in under abnormal operation. To set the threshold, values of the density estimation function were calculated for the normal and faulty training data sets. The threshold for normal operation is initialized to three standard deviations above average discriminant function on the training data. This initial threshold is then adjusted based on the type I or type II errors of the initial threshold. If the type I error rate is higher, the threshold would be decreased to capture faulty data with a lower probability. If the type II error rate is higher, the threshold is raised to reduce the number of false alarms. This optimization converges once the type 1 error = 10*type 2 error. Type 2 error is weighted heavily to avoid a large number of false faulty alarms.

3.4.2 MSOM Contribution Plots

The variables most responsible for the fault are determined by analyzing the residual of the faulty point and the SOM model of the normal region used in fault detection. The data points are projected on the model by locating the BMU of the point. The residual vector is the squared difference of the point and its BMU. For variable $j$, the residual is calculated by:

$$r_j(t) = (x_j(t) - m_{c_j})^2$$

Figure 13 illustrates the advantage of analyzing the residuals for the SOM model for fault identification. Suppose a data set exists with related dimensions $x_1$ and $x_2$ and other independent variables. A detected faulty data point, shown as a large blue circle, inconsistent with the general
trend of the data caused by an alteration in an unmeasured variable that affects $x_1$ and $x_2$, creating deviations in both directions $x_1$ and $x_2$ directions.

In Figure 13, three residuals are plotted using the black braces: the centroid, the nearest data point, and the BMU in the SOM. In this case, the centroid of the nonlinearly related data does not lie within the data cloud itself. Therefore, the deviations from the data point are negligible and do represent the point’s deviation from the data. Not only is the residual vector small, it would erroneously capture only $x_1$ as the deviated variable which ignores the deviation in $x_2$. Using raw uncompressed data causes the faulty point to be matched with a noisy outlier. The SOM residual used is an approximation of an orthogonal projection onto the topological structure of the data cloud. The SOM residual accurately captures both $x_1$ and $x_2$’s contribution.

Figure 13: Illustration of the residual analysis.

3.4.3 MSOM Fault Diagnosis

Data of different classes may vary in size. In order for class identification to be accurate, large classes must be prevented from dominating the space. Therefore, it is advantageous to compress large data sets to fewer points. Rather than eliminating points in some fashion, SOM captures density and topological properties of the data cloud.
In traditional applications of SOM in process monitoring, one map (1-SOM) represents the entire observed space by training with data from all observed states of the process. Fault detection and diagnosis are performed in one step: new observations are classified according to the label of their BMU. The discriminate function is the distance from a point to the map units. This has major drawbacks in practical applications. If the new observation belongs to a fault that has not previously occurred, 1-SOM will erroneously classify it to an existing condition. Additionally, some map vectors are located between multiple classes, particularly during the transition states of a process, increasing the chances for misleading classifications.

MSOM takes a different approach in that each operating regime has its own map expressing the topology of the data under those conditions, but poorly represents any data under other conditions. Multiple SOM maps have been previously applied to classification problems by (Wan 1999), who found it to perform better than standard SOM for an image analysis application. For our application, to determine how to classify a new data vector detected as faulty, the vector’s probability is evaluated for each of the Gaussian mixture models created from the SOM of each regime under consideration. Normal operation was also included to lower the chances of a false alarm. Once the probability of the faulty point occurring on each map is calculated, the maximum discriminant function assigns a point to an operating regime associated to a class:

\[ g(x)_i > g(x)_j \quad \forall j \neq i \]

Where, \( g(x)_i \) is the discriminant function for class \( i \).

When practically implementing a data-driven process-monitoring scheme, engineers must balance a tradeoff between time to develop the model and the unknown potential benefits. Long periods of developing faulty databases can be an arduous task. We instead envision putting a
limited data set consisting of only normal operating regime and include suboptimal incidents as they occur. This would involve the process adapting as new conditions are met. This allows less upfront time input. For single map representations, adding new process conditions affects the model for all data representations. MSOM allows the process monitoring scheme to identify new points, allow the process supervisor to classify them, and incorporate them into the data-driven scheme without affecting the representations of other operating regimes.

3.5 SOM METHODS FOR SOFT SENSING

An SOM can be modified to create a soft sensor and the process used here is illustrated in Figure 14. First, the map is trained using the methods outlined in Chapter 2.2. The difference between using SOM for fault detection and diagnosis and soft sensing is that all variables, those with a time delay and instant variables, are used in training. While using the soft sensor, the BMUs of incoming data vectors are found using only the measured variables, while the predicted values of interest are taken from the values at the BMU calculated during training. Any number of variables can be sensed in this way, although intuition suggests that if the number of predicted variables is near or larger than the number of measured variables accuracy could decrease.

Just like with 1SOM fault detection, one map is created to cover all operating regimes of interest. During training, a map is created using all monitored variables, including the variables to be soft sensed. Therefore, this map can also effectively utilize the visualization capabilities of SOM to view a visual understanding of the process and observe many variables and their interactions. While using the map for predicting measurements, the measured variables are used to locate the BMU of input data vector. Once the BMU has been found using the measured
variables, the soft sensed variables found during training are used as the prediction for the soft sensed variables.

Similar to the 1SOM map, the same maps created for MSOM fault detection and diagnosis can be used for soft sensing compositions as illustrated in Figure 15. Ideally, multiple maps will allow a more precise characterization of the topologies of the different fault regions and grouping similar operating regimes together would allow the individual models to be adapted over time without affecting the characterization of other operating modes. Unlike fault detection and diagnosis, the soft sensing architecture used here would use the map quantization

\[
	ext{Train map using all } N \text{ variables} \\
\downarrow \\
	ext{Set aside the } \alpha \text{ variables to be soft sensed from the codebook vectors} \\
\downarrow \\
\text{Distributed Control System} \\
\downarrow \\
\text{Data vector of measured variables} \ [x_1 \ x_2 \ x_3 \ldots x_{N-\alpha}] \\
\downarrow \\
\text{Process SOM} \\
\downarrow \\
\text{Find BMU on map using the } N-\alpha \text{ measured variables} \\
BMU_i = \arg \min \left( \partial(m_k, v_i) \right), \forall k \\
\downarrow \\
\text{Match the BMU with the values of the } \alpha \text{ predicted variables found during training}
\]

Figure 14: 1SOM based soft sensing
error to decide which map to use to project the incoming data vector. Once the map is selected, the BMU for the data vector is found and used for prediction in the same way as for 1SOM soft sensing.

Figure 15: Schematic for MSOM soft sensing.

3.6 Literature Review

Distance-preservation fault-detection techniques, including PCA, have previously been applied to the Tennessee Eastman process with great success. The ease with which training data
can be generated using the Tennessee Eastman Process has allowed it to become the benchmark for any new process monitoring scheme. Raich and Cinar, (1996) demonstrated the use of standard PCA for fault detection and diagnosis on the Tennessee Eastman Process. (Zhu 2011) applied an ensemble clustering method based on a dynamic PCA-ICA model to label process transitions to the TEP and achieve transition process monitoring using PCA based dimensionality reduction. A thorough treatment of PCA and other statistical techniques to the Tennessee Eastman process is presented in Chiang, (2001). A comparison of these standard techniques applied to TEP can be found in Yin et al (2012).

SOM has been previously applied to fault detection and diagnosis with success. Deventer et. al., (1996) used SOM in tandem with textural analysis for monitoring of a mineral flotation process. Jamsa-Jounela et. al. (2003) utilizes SOM to detect several faults in a smelter and an online tool to assist in its implementation. Garcia et. al. (2004) used SOM and k-means clustering for system state estimation and monitoring with an application to a wastewater treatment plant. Ng and Srinivasan (2008a and 2008b) created an effective training strategy using SOM for multistate operations in addition to a sequence comparison algorithm with applications to a lab scale distillation unit, a refinery hydrocracker, and the Tennessee Eastman Process. They also resample the training data in order to achieve an equal representation of the different operating regimes. Corona et. al. (2010; 2012) applied SOM to the classification of different operating regimes of an industrial deethanizer and included a method to consider quality specifications. Throughout these applications, the use of SOM’s visualization tools was a key advantage in the analysis over other process monitoring techniques.

Previous applications of linear data-driven techniques to the TEP for process monitoring have included detection, identification, and diagnosis rates. These previous works may be limited
by the use of distance preservation techniques and are more applicable to step changes faults. Some researchers have applied SOMs nonlinear topological-preservation features to the TEP. Gu, (2004) presented the visualization of several step faults in the Tennessee Eastman process. Ng, (2008a-b) applied SOM to a different set of measurements generated by the TEP. Chen (2012; 2013) improved the performance of their SOM algorithm using two linear dimensionality reduction techniques, CCA and FDA. These works include multiple step faults on a single map. They have not included a method of fault identification and were not applied to faults with any time dependent dynamics.

An effective soft sensor can produce valuable information about the operating state of the process. Because of the value they can create, soft sensors have been developed for a broad range of applications. Dong (1995) research a particularly important problem in emissions monitoring to predict the amount of certain pollutants released. Wilson (1999) used partial least squares (PLS) along with a radial basic function neural network to soft sense measurements with delays in the Tennessee Eastman process. Abonyi (2003) used an SOM along with local linear models of the process state around the map vectors to predict the properties of the products of a polyethylene plant.

Comprehensive reviews of every area of process monitoring, including fault detection, identification, diagnosis, as well as soft sensing can be found in Venkatasubramanian (2002), (Qin 2012), and Kadlec (2009).
4. Case Study: Tennessee Eastman Process

The source of the data used in our analysis was the Tennessee Eastman Process (TEP). First introduced as a process control challenge problem by Downs and Vogel (1993), the process is a realistic simulation based on a real chemical process and used as an important benchmark in the evaluation of process control and process monitoring techniques.

4.1 Process Description

The process uses four gaseous reactants (A, C, D, and E) to produce two liquid products (G and H) in competition with an unwanted byproduct (F) and in the presence of an inert (B). There are four competing reactions related to temperature through Arrhenius laws, two producing desired products, the others producing the byproduct. The entire simulation includes five unit operations: a reactor, condenser, separator, compressor, and a stripper and has 41 measured variables along with 12 manipulated variables for a total of 53 process variables. Downs and Vogel also defined 20 disturbances or faults for the challenge process. Tables of the measured variables and fault of the Tennessee Eastman process can be found in Appendix A.

The data was generated using the control scheme and Simulink code from Ricker, who applied a decentralized control strategy to the process which involved partitioning the plant into subunits and the creation of a controller for each. The process flow sheet and controller scheme (Ricker 1996) is given in Figure 16. Because online application is the focus of this work, composition measurements with a time delay are not considered, leaving 22 of the original 41 measurements for consideration. Additionally, of the 12 manipulated variables, three are held constant under Ricker’s control scheme (compressor recycle valve, stripper steam valve, and agitator speed) and not considered in the analysis presented.
Figure 16: Process schematic with control scheme

### 4.2 Fault Descriptions

The process simulations of the TEP process contain 20 preprogrammed faults. Most of these faults are particular feed-step-change faults. In this study, the in-depth analysis to characterize other more difficult faults such as random variations, sticky valves and slow drift in kinetics is addressed. Consequently, the specific faults considered in this work are:

- Fault 1 is a step change in the A/C feed ratio in stream 4.
- Fault 4 is a step change in the inlet temperature of the reactor cooling water.
- Fault 7 is a loss of header pressure in stream 4, which reduces the availability of C.
- Fault 8 is a random variation in the feed composition of stream 4.
• Fault 11 imposes a random variation in the cooling water to the reactor, which requires the control system to constantly adjust the cooling water flow to compensate for the reduced or increased cooling capacity of the water used.

• Fault 13 is a slow drift in the kinetics of the reaction which requires many adjustments by the control system to accommodate the changing composition of the reactor output.

• Fault 14 is a sticky valve in the reactor cooling water. Its effect on the rest of the process is similar to the earlier random variation faults where it creates variations that would not be observed under normal operation.

• Faults 14 and 1 (14+1) is the combination of faults 1 and 14, where the state is disturbed by a step change in the composition of steam 4 while coping with a sticky valve.
5. RESULTS AND DISCUSSION

This section presents the results of performing process monitoring on sets of data generated by the Tennessee Eastman process simulation. Step faults are well-studied using traditional SOM methods (Gu 2004, Chen 2012, 2013), but the dynamic faults like Faults 8, 11, 13, and 14 have not been considered with an SOM analysis, nor has the effect of including a combination of Faults 1 and 14 been previously studied. Since each operating regime’s model is in competition with others, adding additional faults with similar dynamics or overlapping regions could have a large effect on fault diagnosis. Various faults were run and tested against PCA and SOM detection and identification methods. Next, the fault diagnosis rates of PCA, standard 1SOM, and MSOM are compared. Finally, the ability of 1SOM and MSOM to soft sense the compositions of two important streams in the Tennessee Eastman process are evaluated.

5.1 TEP FAULT DETECTION

Fault detection was tested by comparing PCA based fault detection methods with MSOM. Data was collected for the six faulty operating regimes and run through mPCA or MSOM fault detection algorithms. The correct classification rate of the two methods are given in Table 1. All variables without a time delay were used except the compressor recycle valve, stripper steam valve, and agitator speed as these numbers were held constant by Ricker’s control system. For fault detection, 480 data points were collected from normal operation, corresponding to 48 hours of operation and used the data generated as a training set for an SOM map. After training, a GMM was used to represent the probability distribution and characteristics the normal operating region. The bound for normal operation was found using the optimization described in Chapter 3.2.
Table 1: Fault detection rate of mPCA and MSOM

<table>
<thead>
<tr>
<th>TEP fault</th>
<th>Detection Rate (%)</th>
<th>mPCA</th>
<th>MSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.9%</td>
</tr>
<tr>
<td>4</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.9%</td>
</tr>
<tr>
<td>7</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>8</td>
<td>99.5%</td>
<td>99.3%</td>
<td>99.3%</td>
</tr>
<tr>
<td>11</td>
<td>99.7%</td>
<td>99.0%</td>
<td>99.0%</td>
</tr>
<tr>
<td>13</td>
<td>99.6%</td>
<td>99.5%</td>
<td>99.5%</td>
</tr>
<tr>
<td>14</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.9%</td>
</tr>
<tr>
<td>14+1</td>
<td>99.9%</td>
<td>99.9%</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

The probability distribution for a vector projected onto the map fitted to normal operation used by MSOM for fault detection is shown in Figure 17. Each map node has a different probability of being the BMU for the projected vector, with the real BMU having the largest probability. Figure 17a gives the PDF of the normal point on the map. It can be seen that the data point with the reddest color indicates the highest probability of being associated with that map node and corresponds to the map node that best represents the data point. Figure 17b however is the same illustration point at the beginning of Fault 1’s deviation. The state of the process is leaving the normal region and approaching the boundaries of normal operation. Finally, in Figure 17c, the representation of the faulty point long after Fault 1 has occurred is shown, showing a
zero probability of the point occurring anywhere on the map. Later, in fault diagnosis, MSOM is used to find this probability distribution behind the scenes for every point requiring classification.

5.2 TEP Fault Identification

Fault identification was performed on data sets from the Tennessee Eastman Process according to the PCA method described in Section 3.1 and the MSOM method proposed in Section 3.2. Contribution plots of faulty data points from each fault were analyzed and compared with contribution plots calculated by a PCA method and our process knowledge to evaluate the novel method’s ability to isolate the causes of the faulty condition. Sometimes, the root causes of the problem are in variables that are not monitored, so the goals of fault identification are to isolate a subsection of the plant where the root causes of the issue are likely located and to give engineers useful clues on how to identify them. In Figure 18, a data vector roughly 10 hours after the imposition of the faulty condition was used to create contribution plots as illustrative examples of the method proposed.

During Fault 1, in response to the changes in the composition of the feed, the composition controller on the reactor feed creates changes in streams 1 and 4 in response to the disturbance. For both PCA and MSOM, the variable with the largest contribution is the controller output to Stream 1, which, based on the knowledge that stream 1 and 4 are in the same control loop, points to a feed composition issue. Fault 7 is a reduction in the flow of stream 4. Here, both tag the stream 4 controller output as abnormal, but PCA does so while highlighting insignificant variations in other feed streams.

Applying a similar analysis to Fault 13, it can be seen that MSOM successfully reproduces the results of PCA in isolating the fault to changes in the pressure of the reactor and
Figure 18: Comparative analysis for contribution plots
in the separation units following the reactor. Dramatic changes in all of these variables at the same time could point to a change in the composition of the outlet of the reactor. In fact, Fault 13 creates a “slow drift” in the kinetics of the reactor. The result is wild variations spread over many process variables, but both methods successfully isolate it to a subset of variables related to the reactor and its composition.

Faults 11 and 14 are illustrative of the value of using MSOM. Both faults relate to the cooling water of the reactor, which may not be very important to the PCA model, as under normal operation reactor cooling water faults remain relatively constant. The result in Figure 18b and 18d is that when a fault related to the cooling water occurs, cooling water variables are sidelined in favor of flow variables. For fault identification, MSOM uses all online measured variables and weights them equally. The result is that while PCA appears to point to more changes in feed streams, MSOM’s largest contribution variables clearly point to a problem with the reactor and its cooling water.

The plots in Figure 18 illustrate MSOM’s superior ability in identifying variables responsible for a deviation from normal operation. The data compression abilities of SOM help identification by reducing the noise in individual data, without the loss of information that comes with performing a dimensionality reduction using PCA. The next section will illustrate the effectiveness of MSOM in diagnosing the faulty condition of the process.

5.3 TEP Fault Diagnosis

In this section, the MSOM method proposed was used to classify data sets from the faults of interest and compared to two standard techniques: mPCA, a standard fault detection and diagnosis technique, and the standard form of SOM which uses a single map for classification. In each case, a series of data from an operating regime of the Tennessee Eastman process is given
to each set of models, and every data point is classified into the different fault categories using each technique’s discriminant function. PCA uses Hotelling’s $T^2$ statistic, 1-SOM uses a given data point’s BMU on the map of the process, and MSOM uses the probability estimated from the GMM fit to each class’s map.

The U-matrix trained to all data for 1SOM fault diagnosis is given in Figure 19a. From the hit diagram in Figure 19b, it can be seen that each fault is restricted to certain regions of the map. Operating regimes at a relatively steady state tend to form a well-defined cluster, indicated on the U-matrix by darkly colored regions. Fault 7 is particularly well separated from the rest of the data as indicated by the red, orange, and yellows separating Fault 7’s region in Figure 19b. Normal operations, Fault 1, and Fault 4 all consistently project to the same areas. Faults with more dynamics occupy a larger area of the map. Often, these regions are not a contiguous area of the map, a phenomenon known as tearing (Lee 2007). Faults 8 and 13 are both torn between two areas of the map characterized by warm, brighter colors. The warm colors visualize the constant changes the process experiences over the propagation of the faulty operations which causes the

Figure 19: (a) The U-Matrix of the SOM used for 1SOM fault diagnosis with (b) a hit diagram of the training data projected onto the map.
SOM nodes to spread out more. Interestingly, Fault 14 and 14+1, which both contain a random disturbance in the reactor cooling water, surround Normal operations and Fault 1 respectively, which how SOM visualizes the way the random perturbations increase the variability of operations.

Figure 20a-c show trajectory plots of the first 10 hours of step Faults 1 and 7 as well as Fault 13, the slow drift in reactor kinetics. These three cases illustrate the different way operation transitions can appear on an SOM trajectory. Observe that the starting point of each trajectory lies in the region labeled as normal in Figure 19b. Fault 1 is characterized by movement through a transitional state towards a steady state in its region on the map. In the process it passes through areas of higher distance (warm colors) before settling into Fault 1’s cluster region. Fault 7 is an abrupt step to its region, without any type of transition. In Fault 13’s constant changes cause the state of the process to drift across large areas of the map. In Figure 20c it can be seen moving to different sides of the map, contacting regions more closely associated with Fault 8 during the transition, which will be misdiagnosed if process monitoring is done using 1SOM. Other overlapping fault areas, including Fault 11 and 14, can be interpreted from Figure 19b. The way some map vectors are shared between multiple classes can create a disadvantage for 1SOM relative to MSOM.

From the results in Table 2, both 1SOM and MSOM techniques represent a dramatic improvement over conventional PCA fault detection. Figure 21 shows how each method classified different portions of the process data. Visualizing the results in this way allows us analyze possible reasons for low classification rates in the results. For example, mPCA’s successful diagnosis of Fault 13 suggests that this fault has been characterized well by its PCA model, while Figure 21a shows that this high detection rate for Fault 13 came at the expense of a
Figure 20: The trajectory from normal operation to (a) Fault 1, (b) Fault 7, and (c) Fault 13.

Table 2: Correct diagnosis rates for selected TEP faults

<table>
<thead>
<tr>
<th>TEP fault</th>
<th>mPCA</th>
<th>1-SOM</th>
<th>MSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.9%</td>
<td>93.0%</td>
<td>99.3%</td>
</tr>
<tr>
<td>4</td>
<td>0%</td>
<td>95.0%</td>
<td>97.5%</td>
</tr>
<tr>
<td>7</td>
<td>37.2%</td>
<td>99.9%</td>
<td>100.0%</td>
</tr>
<tr>
<td>8</td>
<td>79.8%</td>
<td>79.2%</td>
<td>91.4%</td>
</tr>
<tr>
<td>11</td>
<td>69.5%</td>
<td>74.7%</td>
<td>94.6%</td>
</tr>
<tr>
<td>13</td>
<td>97.4%</td>
<td>84.1%</td>
<td>94.3%</td>
</tr>
<tr>
<td>14</td>
<td>21.3%</td>
<td>92.6%</td>
<td>99.3%</td>
</tr>
<tr>
<td>14+1</td>
<td>96.8%</td>
<td>90.3%</td>
<td>99.6%</td>
</tr>
</tbody>
</table>

large number of points from Faults 8, 11, and 14 being incorrectly classified into Fault 13 due to Fault 13’s large $T^2$ bounds. The PCA model for Fault 8 has a similar problem with the result that all points from Fault 4 were classified incorrectly.

Another challenge comes from regions where multiple fault operating regimes overlap. For example, Fault 14’s classification has been spread by mPCA between Fault 8, 11, 13, and 14 due to overlap of the four models. A similar case is the high detection rate for the combination Fault 14+1 where a high correct diagnosis rate for Fault 14+1 was achieved at the cost of many points from Fault 1 being incorrectly diagnosed due to overlapping regions for Faults 1 and
Figure 21: Data classification using (a) MPCA, (b) 1-SOM, and (c) MSOM on the testing data set with time colored according to class. Solid lines of color indicate a consistent classification while discontinuous lines and dots indicate an inconsistent classification.
In each of these cases, the non-linear topology preservation abilities of 1SOM and MSOM offer impressive improvements over mPCA, with MSOM having a consistently higher correct classification rate. MSOM has a better fault diagnosis performance because of the separation between classes, as the sharing of a single map vector between training data of different classes can cause the training algorithm to incorrectly label that area of the map with whichever label is most common in the training data.

5.4 TEP SOFT SENSING

In this section, the results from the soft sensing of the delayed concentration measurements of the Tennessee Eastman process are discussed. As before, only variables without a time delay were used except the compressor recycle valve, stripper steam valve, and agitator speed. The SOM maps used for soft sensing were the same as those used in both 1SOM and MSOM fault detection and diagnosis. Table 3 shows the percent error of predictions by the soft sensor in the reactor feed composition and product stream composition. Because Ricker’s control scheme does not use the GC measurements from the purge gas, these results are not given, but a similar level accuracy was observed in those measurements as well.

Table 3: Percent Error of Selected Compositions

<table>
<thead>
<tr>
<th>Composition</th>
<th>1SOM</th>
<th>MSOM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A in Rxnr Feed</td>
<td>2.04</td>
<td>1.59</td>
</tr>
<tr>
<td>B in Rxnr Feed</td>
<td>2.50</td>
<td>2.47</td>
</tr>
<tr>
<td>C in Rxnr Feed</td>
<td>4.24</td>
<td>3.27</td>
</tr>
<tr>
<td>D in Rxnr Feed</td>
<td>1.78</td>
<td>1.78</td>
</tr>
<tr>
<td>E in Rxnr Feed</td>
<td>2.64</td>
<td>2.70</td>
</tr>
<tr>
<td>G in Product</td>
<td>0.99</td>
<td>0.98</td>
</tr>
<tr>
<td>H in Product</td>
<td>1.27</td>
<td>1.27</td>
</tr>
</tbody>
</table>
Interestingly, 1SOM and MSOM soft sensors did not perform significantly different. Both produced a similar and reliable prediction, which means both methods created a similar representation of process state, in spite of the separations between the maps made by MSOM.

Figure 22 compares the predictions by the two methods with the measurements from Fault 8. The random variation imposed by Fault 8 creates fluctuations in the composition of the reactor feed which both soft sensors could appropriately accommodate. An even greater level of accuracy was observed during step faults that reach a more steady state.

Figure 22: Prediction of the A in the reactor feed as and the residuals of both prediction methods during Fault 8.
An advantage to using an SOM based method is that areas in the data that are poorly modeled can be investigated using SOM’s tools for clues about a soft sensor problem. In Figure 23, the trajectory of a period after point 4700 where the residuals are relatively high is projected to the 1SOM map used for soft sensing. It can be seen that the data from this period is restricted to a limited number of BMUs, which could suggest a factor that the SOM fit may not be accounting for. In a real plant, insights from this analysis could guide future adjustments to map training and updating.

Figure 23: Trajectory plot of a period of 50 points of data following point 4700 of Fault 8.
6. Conclusions

Modern process monitoring is marked by the extensive use of advanced computer systems and large numbers of sensors to determine if a plant is operating in a safe state and to detect when a problem occurs. Identifying these states can be difficult due to the large number of variables measured, but data driven methods offer ways to quickly and efficiently extract useful information from large data sets common in process industries. Improved process monitoring can minimize downtown, increase safety, reduce manufacturing costs, and improve performance, which all contribute to safer and more profitable operations.

This research explored the use of SOM in process monitoring and proposed a new technique for using SOM in process monitoring. The proposed method differs from other SOM-based process monitoring schemes due to the use of multiple maps to characterize the nonlinear topology of each class of faulty data individually. Usually, one SOM is fit to all process states during training. In MSOM, each different operating regime or fault class is modeled by a separate SOM. To enhance the diagnosis ability of the multiple maps, a GMM is created for each map in the state space from its constituent vectors and used in detection and diagnosis as a discriminant function to classify new data vectors with the appropriate map.

In this work, MSOM process monitoring architecture was compared to PCA and a more conventional 1SOM on data generated from the Tennessee Eastman Process with a focus on faulty conditions containing non-linear variations with time that have not been previously analyzed with SOM. Besides fault detection and diagnosis, the ability 1SOM and MSOM to soft sense several stream compositions were compared. The results indicate that MSOM outperforms mPCA and improves upon SOM fault identification and diagnosis through its more effective modeling of process dynamics. With regard to soft sensing, MSOM and SOM performed equally.
well, suggesting that MSOM and SOM create similar representations of composition measurements.

Future work will attempt to apply the methods presented to real plant data and evaluate how the method copes with challenges stemming from data quality, nonlinearities, time sensitive variations, and other problems that cannot be effectively modeled by a simulation. A framework for plant engineers to adapt the models to the long term dynamics of a process and improve the quality of the training data will also need to be explored.
REFERENCES


APPENDIX A: ADDITIONAL TENNESSEE EASTMAN PROCESS DETAILS

Table A1:

Table 4: TEP Measured variables

<table>
<thead>
<tr>
<th>Number</th>
<th>Variable Name</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A feed (stream 1)</td>
<td>kscmh</td>
</tr>
<tr>
<td>2</td>
<td>D feed (stream 2)</td>
<td>kg/hr</td>
</tr>
<tr>
<td>3</td>
<td>E feed (stream 3)</td>
<td>kg/hr</td>
</tr>
<tr>
<td>4</td>
<td>A and C feed (stream 4)</td>
<td>kscmh</td>
</tr>
<tr>
<td>5</td>
<td>Recycle Flow (stream 8)</td>
<td>kscmh</td>
</tr>
<tr>
<td>6</td>
<td>Reactor feed rate (stream 6)</td>
<td>kscmh</td>
</tr>
<tr>
<td>7</td>
<td>Reactor Pressure</td>
<td>kPa gauge</td>
</tr>
<tr>
<td>8</td>
<td>Reactor Level</td>
<td>%</td>
</tr>
<tr>
<td>9</td>
<td>Reactor Temperature</td>
<td>Deg. C</td>
</tr>
<tr>
<td>10</td>
<td>Purge Rate (stream 9)</td>
<td>kscmh</td>
</tr>
<tr>
<td>11</td>
<td>Product Separator Temperature</td>
<td>Deg. C</td>
</tr>
<tr>
<td>12</td>
<td>Product Separator Level</td>
<td>%</td>
</tr>
<tr>
<td>13</td>
<td>Product Separator Pressure</td>
<td>kPa gauge</td>
</tr>
<tr>
<td>14</td>
<td>Product Separator Underflow (stream 10)</td>
<td>m^3/s</td>
</tr>
<tr>
<td>15</td>
<td>Stripper Level</td>
<td>%</td>
</tr>
<tr>
<td>16</td>
<td>Stripper Pressure</td>
<td>kPa gauge</td>
</tr>
<tr>
<td>17</td>
<td>Stripper Underflow (stream 11)</td>
<td>m^3/s</td>
</tr>
<tr>
<td>18</td>
<td>Stripper Temperature</td>
<td>Deg. C</td>
</tr>
<tr>
<td>19</td>
<td>Stripper Steam Flow</td>
<td>kg/hr</td>
</tr>
<tr>
<td>20</td>
<td>Compressor Work</td>
<td>kW</td>
</tr>
<tr>
<td>21</td>
<td>Reactor Cooling Water Outlet Temperature</td>
<td>Deg. C</td>
</tr>
<tr>
<td>22</td>
<td>Separator Cooling Water Outlet Temperature</td>
<td>Deg. C</td>
</tr>
</tbody>
</table>

Table 5: TEP Manipulated variables

<table>
<thead>
<tr>
<th>Number</th>
<th>Variable Name</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>D Feed Flow (stream 2)</td>
<td>%</td>
</tr>
<tr>
<td>24</td>
<td>E Feed Flow (stream 3)</td>
<td>%</td>
</tr>
<tr>
<td>25</td>
<td>A Feed Flow (stream 1)</td>
<td>%</td>
</tr>
<tr>
<td>26</td>
<td>A and C Feed Flow (stream 4)</td>
<td>%</td>
</tr>
<tr>
<td>27</td>
<td>Purge Valve (stream 9)</td>
<td>%</td>
</tr>
<tr>
<td>28</td>
<td>Separator Pot Liquid Flow (stream 10)</td>
<td>%</td>
</tr>
<tr>
<td>29</td>
<td>Stripper Liquid Product Flow (stream 11)</td>
<td>%</td>
</tr>
<tr>
<td>30</td>
<td>Reactor Cooling Water Flow</td>
<td>%</td>
</tr>
<tr>
<td>31</td>
<td>Agitator Speed</td>
<td>%</td>
</tr>
</tbody>
</table>
Table 6: TEP process faults description

<table>
<thead>
<tr>
<th>Fault No.</th>
<th>Description</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A/C Feed Ration, B Composition Constant (Stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>2</td>
<td>B Composition, A/C Ratio Constant (Stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>3</td>
<td>D Feed Temperature (Stream 2)</td>
<td>Step</td>
</tr>
<tr>
<td>4</td>
<td>Reactor Cooling Water Inlet Temperature</td>
<td>Step</td>
</tr>
<tr>
<td>5</td>
<td>Condenser Cooling Water Inlet Temperature</td>
<td>Step</td>
</tr>
<tr>
<td>6</td>
<td>A Feed Loss (Stream 1)</td>
<td>Step</td>
</tr>
<tr>
<td>7</td>
<td>C Header Pressure Loss - Reduced Availability (Stream 4)</td>
<td>Step</td>
</tr>
<tr>
<td>8</td>
<td>A, B, C Feed Composition (Stream 4)</td>
<td>Random Variation</td>
</tr>
<tr>
<td>9</td>
<td>D Feed Temperature (Stream 2)</td>
<td>Random Variation</td>
</tr>
<tr>
<td>10</td>
<td>C Feed Temperature (Stream 2)</td>
<td>Random Variation</td>
</tr>
<tr>
<td>11</td>
<td>Reactor Cooling Water Inlet Temperature</td>
<td>Random Variation</td>
</tr>
<tr>
<td>12</td>
<td>Condenser Cooling Water Inlet Temperature</td>
<td>Random Variation</td>
</tr>
<tr>
<td>13</td>
<td>Reaction Kinetics</td>
<td>Slow Drift</td>
</tr>
<tr>
<td>14</td>
<td>Reactor Cooling Water Valve</td>
<td>Sticking</td>
</tr>
<tr>
<td>15</td>
<td>Condenser Cooling Water Valve</td>
<td>Sticking</td>
</tr>
<tr>
<td>16</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Unknown</td>
<td></td>
</tr>
</tbody>
</table>
VITA

Michael Thomas was born and raised in Baton Rouge, Louisiana and graduated from Catholic High School. He went on to attend Vanderbilt University where he became interested in the application of computers to chemical engineering problems while working as an undergraduate research assistant. In 2012 he graduated with a Bachelor of Engineering in Chemical and Biomolecular Engineering.

Following graduation, he began studies at the LSU Cain Department of Chemical Engineering where he will work to earn a Ph. D.