Detection and Evaluation of Abnormal Events in Complex Industrial Processes

Hallgrimsson, Asgeir Daniel

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Technical University of Denmark

Ásgeir Daniel Hallgrímsson

Detection and Evaluation of Abnormal Events in Complex Industrial Processes

PhD Thesis, October 2020

DTU Electrical Engineering
Department of Electrical Engineering

Detection and Evaluation of Abnormal Events in Complex Industrial Processes

Ásgeir Daniel Hallgrímsson

Technical University of Denmark
Kgs. Lyngby, Denmark, 2020

Technical University of Denmark
Automation and Control (AUT)
Elektrovej Building 326
DK-2800, Kgs. Lyngby
Denmark
Phone: (+45) 45 25 38 00
Email: elektro@elektro.dtu.dk
www.elektro.dtu.dk

Summary

Detection and evaluation of abnormal events in industrial process systems is vital for safe and undisturbed operation. Failure to treat a process operating in an abnormal state can lead to a loss of system functions that are necessary for recovering the process to a nominal state. Advances in process monitoring technologies pose the issue of information overload, namely, the difficulty for an operator to monitor and understand the process information that is provided in real-time. In extreme cases, the operator accidentally disregards critical information, which leads to incorrect diagnosis. Disasters such as the Deepwater Horizon explosion and the crashing of Lion Air Flight 610 and Ethiopian Airlines Flight 320 share a common post-accident diagnostic report: though the primary cause of these incidents was not attributed to operator error, their outcomes would have been significantly mitigated if operators had been provided with relevant and accurate diagnostic information.

To address the issue of information overload, this thesis proposes a statistical method for detecting and evaluating abnormal changes in the signal characteristics of process variables. The method is independent of process knowledge; it only requires a collection of samples for process variables gathered from past operations rather than a physical understanding of the system. The motive for this approach is based on the complexity of modern industrial process systems, as a detailed physical description of the influence of process inputs on process outputs for a system comprising thousands of process variables may not be available. The thesis argues that the performance of the proposed method is associated with its ability to extract features from samples gathered from while the process was consistent with nominal operating conditions. Existing methods provide satisfactorily performance of detecting abnormal events. The thesis argues that their performance of evaluating abnormal events could be improved.

The scientific contributions of the research cover two topics, namely, the detection of abnormal events in nonlinear, dynamic systems and the evaluation of abnormal changes in process variables. A method is proposed for extracting features from samples for process variables with an artificial neural network - a mathematical model that describes a nonlinear function. Abnormal event detection is facilitated by

comparing the features of new observations against those of samples gathered from while the process was consistent with nominal operations. It is concluded that an abnormal event has occurred if the disparity of this comparison exceeded a certain threshold. Abnormal changes in process variables are evaluated by combining a structural analysis of the artificial neural network with a contribution analysis of process variables on the detected abnormal event. The novelty of the proposed method is that it does not require prior instances of abnormal events to evaluate abnormal changes in process variables.

Resumé

Det er altafgørende at detektere og evaluere unormale hændelser i industrielle processystemer til at sikre en sikker og uforstyrret drift. Manglende behandling af en proces, som opererer i en unormal tilstand, kan føre til tab af systemfunktioner som er nødvendige for at processen returnerer til en nominel tilstand. Fremskridt indenfor procesovervågningsteknologier udgør en risiko for overbelastende mængder af information og data, hvilket vanskeliggør operatørens overvågning og forståelse af den procesinformation, som vises i realtid. I ekstreme tilfælde overser operatøren kritiske information, hvilket kan fører til forkerte diagnoser. Katastrofer som Deepwater Horizon-eksplosionen, Lion Air Flight 610 og Ethiopian Airlines Flight 320 flystyrtene deler fælles diagnostiske ulykkes rapporter. Selvom den primære årsag til disse hændelser ikke blev tilskrevet operatørfejl, kunne deres udfald være blevet betydeligt anderledes hvis operatørerne fik relevante og nøjagtige diagnostiske oplysninger.

For at løse problemet med informationsoverbelastning præsenteres der i denne afhandling en statistisk metode til at detektere og evaluere unormale ændringer i signalkarakteristika for procesvariabler. Metoden er uafhængig af viden om processen, da det kun kræver en samling af procesvariable observationer indsamlet fra tidligere operationer uden en større fysisk forståelse af systemet. Motivationen til denne afhandling er baseret på kompleksiteten af moderne industrielle processystemer, da en detaljeret fysisk beskrivelse fra procesinputs til procesoutputs i et system kan omfatte tusindvis af procesvariabler som muligvis ikke er tilgængeligt. Afhandlingen argumenterer for, at den foreslåede metode er baseret på dens evne til at udtrække funktioner fra observationer indsamlet, da processen foregik under normale driftsforhold. Eksisterende metoder giver tilfredsstillende detektering af unormale hændelser. I denne afhandling argumenteres der for, at deres evner til at evaluere unormale hændelser kan forbedres.

Det videnskabelige bidrag i denne afhandling dækker over to emner, nemlig påvisning af unormale hændelser i ikke-lineære, dynamiske systemer og evaluering af unormale ændringer i procesvariabler. Der beskrives en metode til at ekstrahere "features" fra observationer af procesvariabler ved brug af et kunstigt neuralt netværk,

hvilket er en matematisk model, der har ikke-lineære funktion. Unormale hændelser påvises lettest ved at sammenligne ”features” i nye observationer med dem, der blev indsamlet, da processen foregik under normale driftsforhold. En unormal hændelse konkluderes til at have fundet sted, hvis forskellen i denne sammenligning overstiger en bestemt tærskel. Unormale ændringer i procesvariabler evalueres ved at kombinere en strukturel analyse af det kunstige neurale netværk med en bidragsanalyse af procesvariabler baseret på den detekterede unormale hændelse. Det innovative ved den beskrevne metode er, at den ikke kræver tidligere forekomster af unormale hændelser for at evaluere unormale ændringer i procesvariabler.

Samantekt

Uppgötvun og mat á óeðlilegum atburðum í iðnaðarkerfum skipta miklu máli fyrir örugga og ótruflaða nokun kerfanna. Takist ekki að meðhöndla ferli sem starfar í óeðlilegu ástandi getur það leitt til þess að kerfisaðgerðir tapast sem eru nauðsynlegar til að endurheimta ferlið í eðlilegt ástand. Framfarir í tækni við ferlaeftirlit geta haft í för með sér ofgnótt upplýsinga, þ.e. erfiðleika fyrir kerfisstjóra að fylgjast með og skilja upplýsingar um ferli sem eru veittar í rauntíma. Við sérstakar aðstæður kann kerfisstjóri að líta óvart framhjá mikilvægum upplýsingum með þeim afleiðingum að greining reynist ekki rétt. Hörmungar eins og Deepwater Horizon sprengingin og hrap Lion Air 610 og Ethiopian Airlines 320 hafa verið greindar eftirá með sama hætti: þótt aðalorsök þessara atvika hafi ekki verið rakin til mistaka kerfisstjóra hefði verið hægt að milda afleiðingar þeirra verulega ef kerfisstjórar hefðu verið veittar viðeigandi og nákvæmar greiningarupplýsingar.

Til að takast á við ofgnótt upplýsinga leggur þessi ritgerð til tölfræðilega aðferð til að greina og meta óeðlilegar breytingar á merkiseinkennum ferlabreytna. Aðferðin er óháð ferlaþekkingu; það þarf aðeins safn sýnishorna fyrir ferlabreytur sem safnað er frá fyrri aðgerðum frekar en eðlisfræðilegan skilning á virkni kerfisins. Hvatinn að þessari nálgun er byggður á flækjustigi nútíma iðnaðarkerfa þar sem ítarleg eðlisfræðileg lýsing á áhrifum ferlainntaks á ferlaúttak fyrir kerfi, sem samanstendur af þúsundum ferlabreytna, er hugsanlega ekki til staðar. í ritgerðinni er því haldið fram að virkni þeirrar aðferðar sem lögð er til sé tengd getu aðferðarinnar til að draga fram eiginleika úr sýnishornum sem safnað var þegar ferlið hafði eðilega virkni. þótt núverandi aðferðir greini óeðlilega atburði með fullnægjandi hætti er því haldið fram í ritinu að auka megi getu þeirra til að meta óeðlilega atburði.

Vísindalegt framlag rannsóknarinnar er tvíþætt: annars vegar greining á óeðlilegum atburðum í ólínulegum, tímaháðum kerfum og hins vegar mat á óeðlilegum breytingum á ferlabreytum. Lögð er til aðferð til að draga fram eiginleika úr sýnishornum fyrir ferlabreytur með gerfitauganet - stærðfræðilegt líkan sem lýsir ólínulegu falli. óeðlileg uppgötvun á atburði er auðvelduð með því að bera saman þætti sem varða nýjar athuganir við þætti sem byggja á sýnishornum sem safnað var þegar ferlið hafði eðlilega virkni. óeðlilegur atburður er talinn hafa átt sér stað ef misræmi í þessum

samanburði fór yfir ákveðin mörk. óeðlilegar breytingar á ferlabreytum eru metnar með því að sameina formgerðargreiningu á gerfitauganetinu og framlagsgreiningu á ferlabreytum á greindan, óeðlilegan atburð. það sem er nýtt við aðferðina sem lögð er til er að það þarf ekki fyrri tilvik af óeðlilegum atburðum til að meta óeðlilegar breytingar á ferlabreytum.

Preface

This thesis was prepared at the Department of Electrical Engineering at the Technical University of Denmark in partial fulfillment of the requirements for acquiring a Ph.D. degree. The Ph.D. study commenced in mid-August 2017 and the dissertation was submitted in mid-October 2020. The research project was a part of a research venture on "Water-Management" funded by the Danish Hydrocarbon Research and Technology Center. The goal of the "Water-Management" project was to improve monitoring and operations of oil and gas production platforms. The project supervisors were:

- Associate professor Hans Henrik Niemann (principal supervisor), Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark.
- Professor Emeritus, Senior Researcher Morten Lind (co-supervisor), Department of Electrical Engineering, Automation and Control Group, Technical University of Denmark.

This thesis consists of five research papers and a summary report describing the contributions of the research.

Ásgeir Hallgrímsson

Kongens Lyngby, October 2020
Ásgeir Daniel Hallgrímsson

List of Publications

Publications included in the thesis

(A) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2019). Autoencoder based residual generation for fault detection of quadruple tank system. *IEEE Conference on Control Technology and Applications, p:994-999*.

(B) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Improved process diagnosis using fault contribution plots from sparse autoencoders. *21st IFAC World Congress*.

(C) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Unsupervised Isolation of Abnormal Process Variables using Sparse Autoencoders. *Journal of Process Control*. Submitted paper under review.

(D) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Modeling Correlations of Nonlinear Process Variables with Expanding Autoencoders. *Journal of Process Control*. To be submitted for review.

(E) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Fault detection with recurrent autoencoders. *Journal of Process Control*. To be submitted for review.

Acknowledgments

Sá sem aldrei er forvitinn verður aldrei fróður.
He who is never curious will never know.

Throughout my Ph.D. studies I have received a great deal of support and assistance. I would like to start by thanking my supervisors Henrik Niemann and Morten Lind for their mentorship. Your guidance has been an important factor in my development as a researcher. It has been a pleasure to work with you, and I would like to express my thanks for always being readily available with your assistance.

One of my most cherished aspects of the Ph.D. study has been my interaction with my co-workers and co-students at the Automation and Control group. I was fortunate enough to be surrounded by a group of talented researchers, each providing valuable insights into my work freely. The open environment and frequent group meetings cultivated my skills at communicating the complex nature of scientific research, of which I am grateful to part with. I will cherish the time I spent with the friends I made at the group during our “coffee breaks", birthday celebrations and Christmass lunches, as well as other endeavors. Special thanks go to Carlos Corchado, Thomas Thuesen Enevoldsen, Marie Claire Capolei, Dimitris Papageorgiou, Fletcher Thompson, Emil Krabbe Nielsen, Robert Miklos, Adriana Zsurzsan, Christopher Reinartz, and Rasmus Hjorth Andersen for their camaraderie, cooperation, and discussions.

I would like to thank my family and friends for their support during these years. I would like to especially thank my parents for all they have done for me.

It would be deceptive to suggest that my Ph.D. study has been all but an easy journey. I came to realize that as I was confronted, time and time again, with a strenuous period, I was always able to count on my girlfriend Mette for support, whether it was in the form of encouraging words or a delicious meal. I am grateful for you and our cat Vera for being my little family away from home.

Table of Contents

List of Abbreviations

AE Autoencoder. 39, 44, 45, 46, 51, 52, 53, 55, 59, 60, 61, 69, 72, 73, 75, 76, 77, 78, 80, 81, 82, 83, 85, 86, 87, 88, 90, 92, 94, 95, 96, 97, 98, 99, 100, 101, 103, 105, 107, 108, 109, 110, 111, 112, 113, 117, 119, 120, 121, 122, 123, 124, 125, 126, 128, 129, 131, 133, 134, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157

AEM Abnormal Event Management. 4, 5, 6, 7, 12, 16, 64

ANN Artificial Neural Network. 39, 40

CPV Cumulative Percentage Variance. 51

DHRTC Danish Hydrocarbon Research and Technology Center. 11

FNN Feedforward Neural Network. 40, 41, 42, 44, 45, 138, 139

ICA Independent Component Analysis. 35, 36, 37, 38, 39, 45, 47, 51, 52, 53, 54, 132, 133

KPCA Kernel Principal Component Analysis. 38, 39, 46, 51, 52, 53, 133

LCL Lower Control Limit. 24, 25, 26

LP Latent Projection. 33, 34, 35, 36, 37, 38, 39, 44, 45, 46, 47, 48, 49, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 66, 70, 71, 72, 73, 77, 78, 79, 82, 85, 86, 104, 105, 112, 121, 124, 128

MQC Multivariate Quality Control. 86

MSE Mean Squared Error. 141, 145, 152

PC Principal Component. 71, 72, 75, 76, 80

PCA Principal Component Analysis. 34, 35, 36, 37, 38, 45, 46, 47, 51, 52, 53, 60, 62, 69, 71, 72, 73, 75, 76, 80, 81, 82, 83, 86, 87, 88, 92, 105, 106, 107, 125, 132, 133, 144, 156

PLS Partial Least Squares. 105

QTP Quadruple Tank Process. 60, 69, 73, 74, 75, 79, 80, 81, 82, 83, 88, 90

RAE Recurrent Autoencoder. 61, 62, 63, 67, 68

RNN Recurrent Autoencoder. 87

SDG Signed Directed Graph. 9, 10, 12, 13

SPC Statistical Process Control. 70, 86, 87

SPE Squared Prediction Error. 25, 26, 27, 32, 48, 49, 50, 53, 54, 56, 57, 78, 81, 82, 94, 99, 100, 106, 111, 112, 113, 116, 121, 122, 123, 124, 125, 126, 144, 145

SPM Statistical Process Monitoring. 132

TTP Triple Tank Process. 61, 85, 88, 89, 90, 96, 97, 100, 117, 118, 119, 120, 121, 122

UCL Upper Control Limit. 24, 25, 26

VAE Variational Autoencoder. 87

Chapter 1

Introduction

Human beings regularly operate systems they understand little of. An example of such a system is the automobile. A 2020 report on car ownership statistics states that 93.3% of households in the U.S.A. have access to at least one vehicle [8]. An individual earns the right to drive a car by obtaining a driver's license, which is granted if the individual has (a) demonstrated their ability to drive under normal conditions via a road test; and (b) confirmed their knowledge of driving and relevant rules via a theory test. There is, however, no explicit requirement that the driver knows the innate functionality of the vehicle they have earned the right to drive.

A driver is essentially an operator who supervises the operation of a car. Cars are equipped with a user control interface that is operated by a combination of the hands and feet. Those controls include a steering wheel, pedals for engaging the brakes and rotary speed of an engine, and several buttons and dials for turning on lights and other functions. Cars are also equipped with a dashboard - a control panel that displays control and monitoring instruments, such as vehicle speed, engine speed, and fuel level. Other features exist to ensure safe operation of the vehicle, such as indicators for low oil pressure, high engine temperature, and engagement of the handbrake. Though the driver's primary objective is to operate the car with a certain goal in mind (such a transportation) under certain constraints (such as time), they must also monitor the condition of the car to ensure its safe operation.

An engineering artifact is developed with a certain functionality that permits it to satisfy a certain goal. Over time, an artifact becomes unable to meet the goal it was designed for and is considered to no longer being consistent with normal operating conditions. Such an incident ensues either via a steady degradation of functionality, such as a reduction in the power output of a car engine that develops with use, or via an event that the artifact was not designed for, such as the puncturing of a tire with a sharp object. Human beings are regularly confronted with the problem of diagnosing

abnormal conditions of engineering artifacts. The task can appear burdensome if one knows little about the innate function of said artifact, such as the case of a driver operating a car. However, human perception and the ability to memorize past experiences provides a different means to diagnosis. A driver learns with use the sensory information they expect to receive whilst operating a car. Via experience, the driver memorizes the typical response of engaging the engine in terms of sound (auditory information), acceleration (visual information from the dashboard and vestibular information), and mechanical vibrations (vestibular information), and develops an expectancy for the distance their car may drive with a full tank. With time, the driver may never learn the functional aspects of a car, but may develop an understanding on what constitutes a normal operating experience. The driver then detects abnormal conditions when the learned experience is not met by their current experience. An extreme example would be the unexpected event of smoke rising out from the hood of the car. A subtle example would be a decreased capability to accelerate that occurs under a certain gear configuration.

The ability to (a) determine when an abnormal event has occurred; and (b) extrapolate any exposed symptoms becomes invaluable when qualitative diagnosis is performed by a certified expert, i.e., a car mechanic in the car example. Symptomatic information aids in the diagnostic process because it provides an initial starting point for diagnosis. For example, notifying the sight of smoke could imply problems with the engine's water cooling system, and a lack of acceleration and presence of a hissing noise at the back of the car implies an issue with the exhaust system.

This thesis considers the diagnosis of abnormal events in industrial process systems. Modern industrial process systems are complex arrangement of engineering artifacts that are continuously monitored by its operators. Current state of the art systems for diagnosis correspond closely to the car-example provided above, namely, that operators are provided with symptomatic information that they must evaluate when performing diagnosis. This is made possible from the fact that process systems are equipped with numerous measurement devices, which permits a high-level of observability in terms of inferring the system's internal states. However, the method for how this information is provided often results with it being unclear, which risks incorrect diagnosis. This thesis is directed towards improving this.

1.1 Industrial systems and abnormal event management

An industrial process system is an engineered arrangement of control systems and associated instrumentation that facilitates the control and monitoring of industrial processes - procedures involving chemical, physical, electrical, or mechanical steps

that transform raw material and energy into a usable product. Industrial systems are an integral component of manufacturing, that is, the production of things essential to human activity. Manufacturing is important to the economic and technological welfare of society, so much so that a strong manufacturing base is vital if a nation is to provide a high standard of living for its people [44]. Figure 1.1 shows that manufacturing is a stable contributor to overall GDP [14], [70]. In fact, manufacturing has contributed roughly 16% of the world's overall GDP during the past decade.

Quality management is the process of overseeing the quality of products, as well as the means to achieve it. It is central to the production process. It has several components, but of relevance in this thesis is quality assurance and quality control. ISO 9000 defines quality control as "a part of quality management focused on fulfilling quality requirements" [57]. Quality control focuses on detecting defects in end products, such as an abnormal composition of carbon, hydrogen, oxygen, and

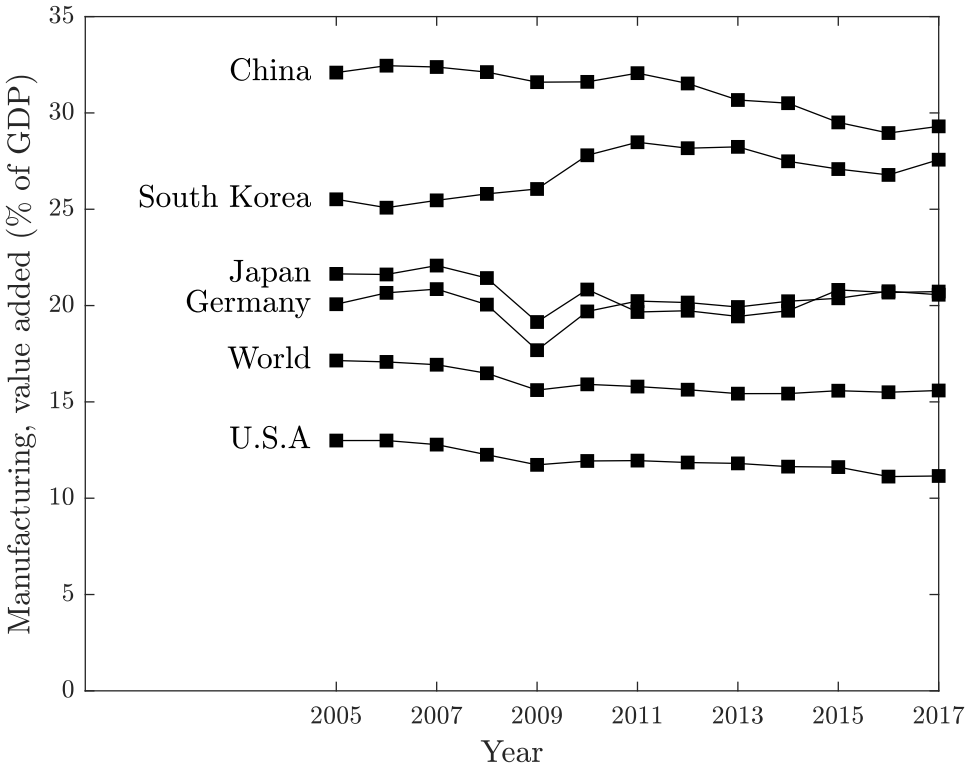


Figure 1.1: Manufacturing, value added (% of GDP) for the world and top five manufacturing countries (by total value of manufacturing). Data obtained from World Bank Open Data [140].

sulfur in refined petroleum. Quality control ensures that an industrial process is following the operations that it was designed for. ISO 9000 defines quality assurance as "a part of quality management focused on providing confidence that quality requirements will be fulfilled" [56]. Quality assurance is preventative in nature, and ensures that industrial processes are implemented correctly by recognizing flaws in the process. An example would maintaining the temperatures in the fractionating columns of a distillation tower within specified limits, thus ensuring that crude oil is separated into its different components.

Process control and process monitoring are two pillars that facilitate quality control and quality assurance in industrial process systems. Process control is the integration of control engineering with process engineering disciplines that uses industrial control systems to achieve a consistent production level in terms of performance, safety, and quality with minimal human assistance. Low-level micro-actions that used to be performed by human operators, such as the opening and closing of valves, are now performed in an automated manner. Automation allows for accurate control actions that facilitate a level of production unachievable purely by human manual control. Process monitoring accounts for the continuous surveillance of process variables, namely, the collection of control inputs and measurement outputs, in order to evaluate abnormal events that interfere with nominal operating conditions. Process monitoring contributes to quality control and assurance by being a central component of Abnormal Event Management (AEM) – a procedure consisting of: (a) timely detection for judging the occurrence of an abnormal event; (b) diagnosing its causal origins; and (c) taking appropriate control decision to bring the process back to a nominal state.

An abnormal event is an abnormal change in the nominal function of a process system that causes a deterioration in its performance [20]. It is synonymous to a fault. Abnormal events include structural changes such as a leakage in a heat exchange pipe that permits the mixing of hot and cool fluids, as well as parametric changes such as a malfunction in a power supply that reduces its supply of electric current. In principle, disturbances - actions of the environment on a system - and faults can have similar effects on a system; they cause an undesired change in a system. However, an important distinction between faults and disturbances is that disturbances are always present, while faults may be present. Furthermore, a control system is designed to attenuate the influence of known disturbances. Faults, which tend to not be known, are not taken into consideration during control design, such that the controller will fail to attenuate its influence.

The performance regions that are considered in the context of fault diagnosis are shown in Figure 1.2. Assume that the condition of a system can be described

by the two process variables x_1 and x_2. The system is considered to exhibit nominal operating conditions, i.e., satisfy its innate function, if it remains in the region of required performance. Movements in the (x_1, x_2) space should only be the result of control actions that attenuate disturbances and reference changes. Faults bring the system into the region of degraded performance. A diagnostic system is to detect this fault-induced shift and diagnose its causal origins. A recovery control action is then initiated to prevent further degradation of performance and return the system back to the region of required performance. This sequential process of diagnosis and recovery constitutes the AEM procedure, and is invoked when the system crosses the border of the required and degraded performance regions. AEM is critical to the survival of industrial process systems; if left untreated, a fault may progress to failure, which is "the inability of an engineering process, product, service or system to meet the design team's goals for which it has been developed" [31], and may lead to the loss of system functions such that recovery is not possible. In principle, AEM is the means by which the development of a fault to failure is prevented.

It is noteworthy to point out that systems remaining in the region of required performance are not necessarily free of faults. For example, even if a corrective action moves a system from the region of degraded performance back to the region

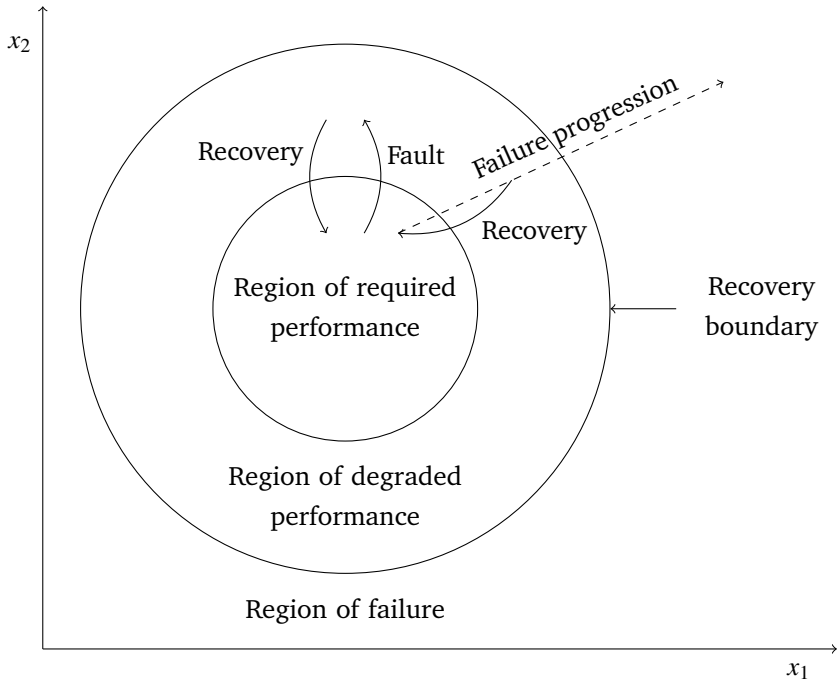


Figure 1.2: Regions of required and degraded performance.

of required performance, the fault still persists; the corrective action simply "hides" the effects of the fault. For instance, switching from a faulty primary power supply to a secondary power supply will retain operational performance, but the primary power supply remains faulty. A closed loop controller may even keep the system within the region of required performance if certain faults occur. For instance, a leakage in a tank can be negated by pumping more water into the tank; the leakage persist, but the amount of water contained remains the same.

Figure 1.3. depicts the integration of an AEM architecture with a closed loop control system, where the measured states of the plant are actuated via a combination of (a) open loop control signals and (b) closed loop control signals. Closed loop control signals regulate the system in terms of reference tracking and disturbance rejection. Faults are classified as follows:

- Plant faults: Faults that affect the dynamic properties of the system;
- Actuator faults: Plant properties are unaffected, but actuators provide undesired/degraded performance.
- Sensor faults: Plant properties are unaffected, but sensors produce erroneous readings.

The blocks 'diagnostic system' and 'recovery decision' comprise the AEM architecture. The diagnostic system processes the control input signals $\mathbf{u}$ and measured output signals $\mathbf{y}$ to detect an abnormal event and then characterize information on actuator, plant, and sensor faults $\mathbf{f}_u$, $\mathbf{f}_p$, and $\mathbf{f}_s$, respectively. The recovery decision consolidates/uses the fault information to produce an appropriate control decision $\mathbf{u}_r$ that recovers the process back to a nominal state. This research project focused on the diagnosis of abnormal events, which corresponds to the diagnostic system block.

1.2 Diagnosis of abnormal events

Abnormal events negatively impact several aspects of production. For instance, a fault-induced deterioration in process performance can reduce overall product quality, as well as increase the operational cost required to maintain stable productivity, resulting in diminished economic profit. Abnormal events can be the source of accidents that result in occupational injury, illness, and death, which have negative impacts on society. A fault progressing to the failure of a single system function may render an entire process non-operational. Failures can be the cause of major catastrophes that have serious environmental ramifications, such as the Deepwater Horizon oil spill. Though the context of abnormal events so far has been

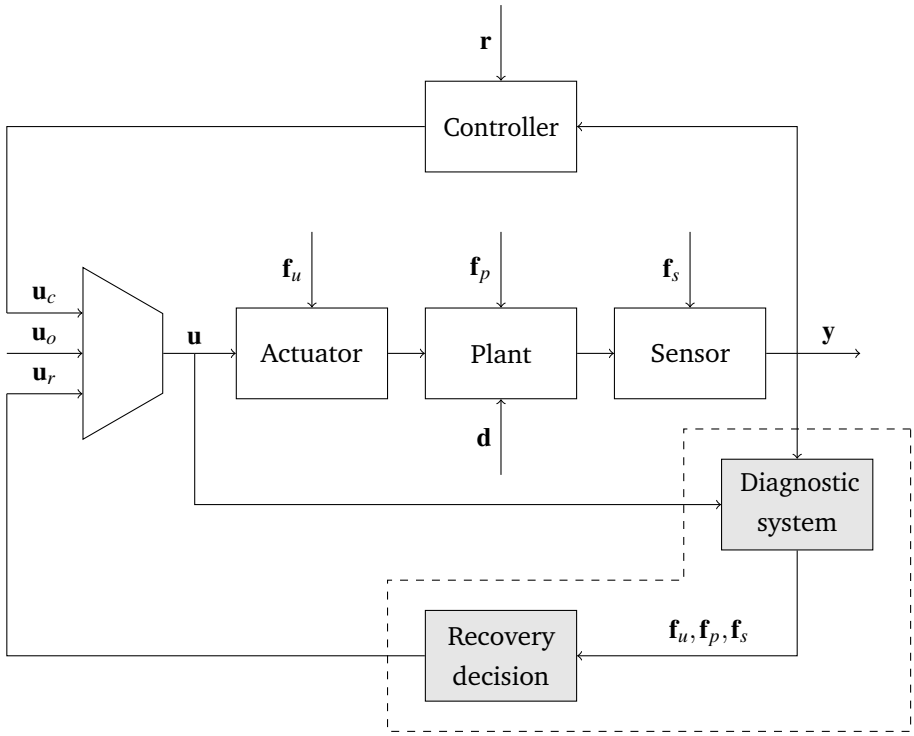


Figure 1.3: General AEM architecture. Notation: $\mathbf{y}$ - Output; $\mathbf{r}$ - Reference; $\mathbf{u}_c$ - Closed loop control input; $\mathbf{u}_o$ - Open loop control input; $\mathbf{u}_r$ - Recovery control input; $\mathbf{u} = [\mathbf{u}_c, \mathbf{u}_o, \mathbf{u}_r]$ - Input; $\mathbf{d}$ - Plant disturbance; $\mathbf{f}_u$ - Actuator fault; $\mathbf{f}_p$ - Plant fault; $\mathbf{f}_s$ - Sensor fault.

on industrial processes, abnormal events also occur in other systems. For example, the crashing of Lion Air Flight 610 and Ethiopian Airlines Flight 302 stemmed from a malfunction in the angle of attack measurements, resulting in loss of life.

AEM is essential for maximizing productivity, maintaining stable plant operation, and ensuring the safety of human operators. Timely diagnosis creates a larger time window for initiating recovery control actions that restore the process back to a nominal state and prevent further progression to failure. Various methods have been developed for abnormal event diagnosis over the years. Each method requires prior knowledge about the nominal behavior of a process, but the source and representation of this knowledge differs between the methods. Approaches are divided into knowledge-based methods, where knowledge stems from an understanding of the process using first principles, and evidential-based methods, where knowledge is gathered from past experiences and a more abstract understanding of the process. Each approach is subcategorized into a quantitative or qualitative method.

1.2.1 Knowledge-Based Quantitative Methods

Knowledge-based quantitative methods are based on analytically redundancy [131]. An explicit mathematical model is used to estimate the output measurements given the computed input control signals. A residual computes the difference between the output measurements and their estimates. The residual represents the consistency between the physical plant and its analytical model. The residual will be non-zero due to faults, disturbances, process and measurement stochastisity, and/or model uncertainties. Incorporating more knowledge increases the residual's sensitivity to faults and decreases it to the other variations. Consider the depiction of a single tank system in Figure 1.4. The nonlinear differential equation describing the nonstochastic evolution of the liquid level in the tank is derived by applying mass balances and Bernouilli's law:

$$\frac{dh}{dt} = -\frac{a}{A}\sqrt{2gh(t)} + \frac{k}{A}u(t) \tag{1.1}$$

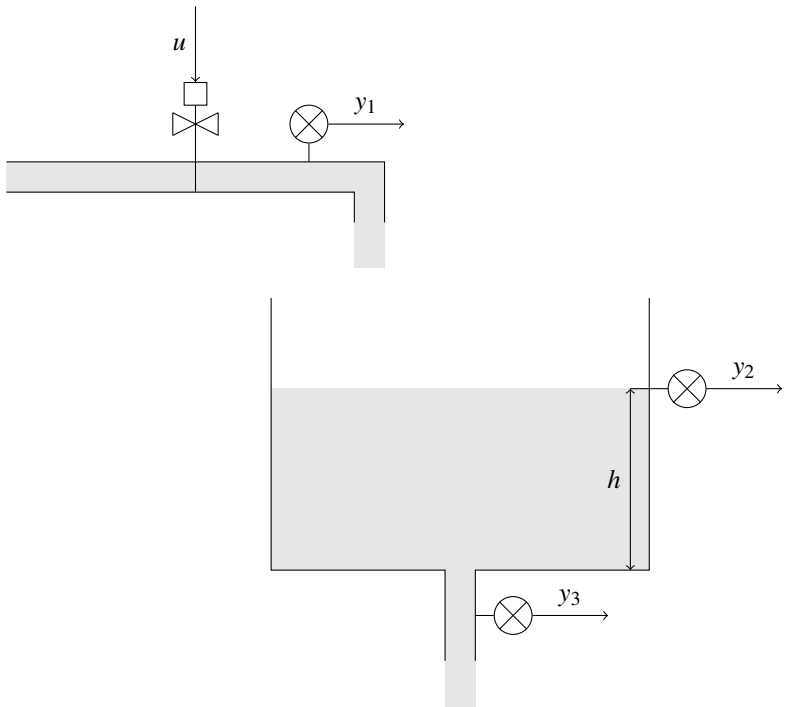


Figure 1.4: Illustration of an open loop single tank system. Notation: h - liquid level; y_1 - measurement of inlet flow; y_2 - measurement of liquid level; y_3 - measurement of outlet flow; u - control input signal for valve.

where A is the cross section of the tank, a is the cross section of its outlet hole, and $ku(t)$ is the flow into the tank. The measurements of the inlet flow, liquid level, and outlet flow are:

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) & y_2(t) & y_3(t) \end{bmatrix}^\mathsf{T} = \begin{bmatrix} ku(t) & h(t) & \sqrt{2gh(t)} \end{bmatrix}^\mathsf{T} \tag{1.2}$$

The consistency of the single tank system with the model is reflected by the residual:

$$\mathbf{r}(t) = \mathbf{y}(t) - \hat{\mathbf{y}}(t) \tag{1.3}$$

where $\hat{\mathbf{y}}(t)$ is evaluated by solving for $\mathbf{y}(t)$ in Eqs. (1.1)-(1.2) given the control signal $v(t)$ computed during the system's operation. The residual $\mathbf{r}(t)$ is then evaluated in order to detect, isolate, and identify faults that occur in the system.

1.2.2 Knowledge-Based Qualitative Methods

Knowledge-based qualitative methods are based on qualitative causal models - models that describe the influence relationship between process components, process functions, and control architecture [132]. Diagnosis is a combination of causal modeling of the system and fault-symptom analysis. It consists of a knowledge base (a large set of if-then-else rules that detail a system's causal nature) and an inference engine that searches though the knowledge base to derive conclusions, i.e., causes, from given evidence of abnormal process behavior. Unlike knowledge-based quantitative models, qualitative models do not require knowledge of the underlying physics governing a process but rather a fundamental understanding of its behavior. For example, a qualitative model that summarizes the single tank system in Figure 1.4 is:

$$\frac{dh}{dt} \propto F_1(t) - F_2(t) \tag{1.4}$$

where F_1 and F_2 correspond to the inlet and outlet flows, respectively. Rather than describing the inherent dynamics of the liquid level, Eq. (1.4) describes its fundamental behavior: the rate of change in liquid level is proportional to the difference of the inlet and outlet flows. In principle, Eq. (1.4) describes a cause-and-effect relationship. The remainder of the equations are:

$$u(t) \propto F_1(t) \tag{1.5}$$

$$\mathbf{y}(t) = \begin{bmatrix} y_1(t) & y_2(t) & y_3(t) \end{bmatrix}^\mathsf{T} \propto \begin{bmatrix} F_1(t) & h(t) & F_2(t) \end{bmatrix}^\mathsf{T} \tag{1.6}$$

A signed directed graph (SDG) is a qualitative causal model that incorporates the cause-and-effect relations among process components and functions with abnormal deviations observed in process variables. The graph consists of nodes that represent

the condition of process functions and directed edges that represent the causal relationship between them. A node takes the value of 0 when its corresponding process variable satisfies nominal operating conditions, + when its variable exhibits an abnormal positive deviation, and − when its variable exhibits an abnormal negative deviation. Arcs take values of + and − representing the propagation of a cause-and-effect chain in the same or opposite deviation, respectively. A SDG of Eqs. (1.4)-(1.5) is shown in Figure 1.5. The expected symptoms of three faults known to occur in the system are shown from the SDGs: a blockage in the inlet flow (Figure 1.5(a)), a blockage in the outlet flow (Figure 1.5(b)), and a leakage in the tank (Figure 1.5(c)). Diagnosis consists of comparing the symptoms observed in a SDG with symptoms expected from known faults.

1.2.3 Evidential-Based Quantitative Methods

Evidential-based quantitative methods establish a diagnostic system with a combination of: (a) historical process data - a collection of samples for process variables gathered during past operations of the process; and (b) knowledge of whether data was sampled from when the process exhibiting nominal or abnormal behavior [133]. Diagnosis consists of characterizing new observations as being sampled from either a nominal or abnormal process.

Evidential-based quantitative methods are divided into two different approaches, namely, supervised anomaly detection and unsupervised anomaly detection. Su-

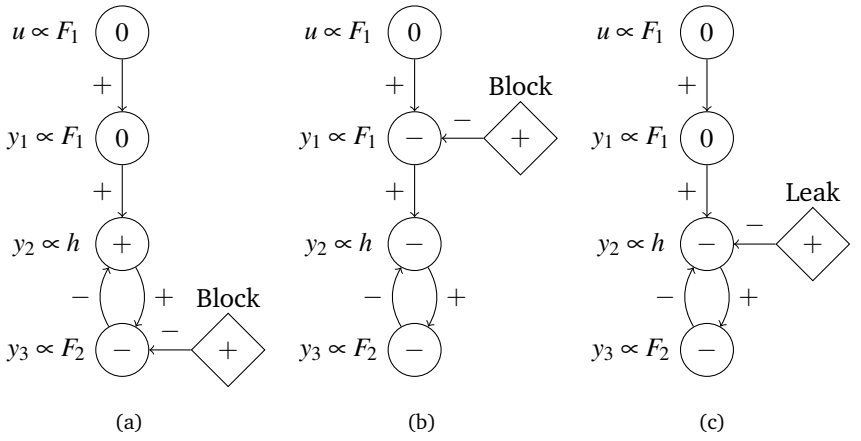


Figure 1.5: SDGs for the single-tank system with symptoms of (a) a blockage in the inlet flow, (b) a blockage in the outlet flow, and (c) a leakage in the tank. Circular nodes represents qualitative variables, and diamond-shaped nodes represent faults.

pervised anomaly detection formulates diagnosis as a classification problem. The method requires knowledge of the process's condition for every sample such that each sample is labelled as "nominal", "abnormal 1", "abnormal 2", etc. A classifier is trained to identify which of a set of faults a new observation belongs to. Unsupervised anomaly detection formulates diagnosis as a feature extraction problem. The method requires historical data strictly sampled from a process consistent with nominal operational conditions. A feature extractor is trained to extract features from the nominal data. Fault detection consists of comparing the features of new observations against those learned from the nominal data; a fault is deemed to have occurred if the disparity is significantly large. Unlike supervised anomaly detection, unsupervised anomaly detection offers no information of which fault has occurred.

1.2.4 Evidential-Based Qualitative Methods

Evidential-based qualitative methods are grounded in qualitative trend analysis - abnormal deviations in process variables are referred to in terms of qualitative elements, such as: normal, high, low, increasing, decreasing, etc [133]. A trend is a change in a process variable that develops with time. Diagnosis via trend analysis involves a hierarchical representation of evident variable trends, comparison of current trends with trends of previous faults, and the ability to distinguish between control input-induced transients and abnormal deviations.

1.3 Motivation, goals, and scope of the project

This research project was a part of larger research venture on "Water-Management" funded by the Danish Hydrocarbon Research and Technology Center (DHRTC) at the Technical University of Denmark. The goal of the "Water-Management" project is to improve monitoring and operations of oil and gas production platforms. Under development is a methodology that generates a knowledge-representative qualitative causal model from plant documentation that facilitates abnormal event diagnosis and recovery action planning. The goal of this work was to propose a method for detecting and evaluating evidence of abnormal process behavior that serves as evidence for initiating the causal model's inference engine.

The size and complexity of oil and gas production platforms - or any modern process plant, for that matter - is an important factor in the design of a method for abnormal event diagnosis. For example, there may be as many as 1500 observable process variables in a large process plant [9]. Size can render a knowledge-based quantitative approach to diagnosis cumbersome, as a large number of process variables requires a mathematical model that is represented by a large set of differential

equations that relate the influence of control inputs on the measurements outputs. Most achievements in knowledge-based quantitative methods are dedicated for linear time-invariant systems, despite the fact that most practical systems are time-variant and nonlinear [151]. For example, oil and gas production plants involve the flow of compressible fluids, which requires competence in thermodynamics and computational fluid mechanics to model. Models available in the literature may require certain model parameters that are unknown or difficult to determine experimentally.

Rather than being based on first principles, process knowledge is regularly represented in the form of plant documents and operator experience. Fault diagnosis is performed in modern process plants with a combination of alarm systems and causal reasoning [139]:

- **Alarm systems:** Alarm systems provide operators with information on abnormal changes in the characteristics of process variables. According to the industrial standard ANSI/ISA-18.2 [55], "an alarm system is the collection of hardware and software that detects an alarm state, communicates the indication of that state to operators, and records changes in the alarm state". The alarm state characterizes an abnormal change in a variable's characteristics with a qualitative label such as high, low, increasing, decreasing, etc. The most common approach in detecting an alarm state $x_a(t)$ of a process variable $x(t)$ is to compare its value to a constant high and low trippoints x_{low} and x_{high} [135]:

$$x_a(t) = \begin{cases} 0, & \text{if } x_{low} \leq x(t) \leq x_{high} \\ -, & \text{if } x(t) < x_{low} \\ +, & \text{if } x(t) > x_{high} \end{cases} \tag{1.7}$$

  In principle, alarm systems are a synthesis of (a) an evidential-based quantitative approach for detecting abnormal events; and (b) an evidential-based qualitative approach for characterizing abnormal changes in process variables.

- **Causal reasoning**: Upon receiving an alarm state/states, an operator applies their qualitative-based process knowledge to (a) deduce its causal origins and (b) deduce the corrective control actions needed to return the variable to within normal operating ranges. Though this approach is a knowledge-based qualitative method, operators tend to perform this manually on the fly and without any explicit qualitative model such as a SDG.

This approach to AEM has become difficult for three reasons:

- **Selection of alarm limits:** The selection of alarm limits is an important factor in the process of characterizing abnormal changes in process variables. An

abnormal change may remain undetected if the limits are specified incorrectly, which can lead to an incorrect diagnosis from causal reasoning. Consider again the single tank system in Figure 1.4. Figure 1.6(a) displays a time series of control signal u and the measurements y_1, y_2, and y_3. A leakage in the tank occurs at sample T_f and causes the measured tank level y_2 and measured outflow y_3 to decrease such that y_2 and y_3 cross their lower control limits. Consequently, the nodes of x and F_2 are given a negative label in the SDG in Figure 1.6(b). Figure 1.6(c) displays a time series of the same signals. A blockage occurs at the inlet flow at sample T_f and causes a decrease in the measured inlet flow rate y_1, measured tank level y_2, and measured outflow y_3. Variables y_2 and y_3 cross their lower limit, but the decrease in y_1 is undetected due to the improper selection of its alarm limits. Consequently, the nodes of x and F_2 are given a negative label in Figure 1.6(d), while node F_1 is given a nominal label. The SDG of Figure 1.6(d) is exactly the same as in Figure 1.6(b); hence the blockage would be improperly reasoned as a leakage fault.

- **Univariate approach to multivariate diagnosis:** Causal reasoning is a multivariate approach to diagnosis: the collection of alarm states one receives are assessed collectively when determining their causal origin. Eq. (1.7) indicates that the method for characterizing an alarm state is univariate: the process variable $x(t)$ is assumed to be independent from other variables when evaluating the alarm state $x_a(t)$. The assumption of independence between abnormal variables is rarely met in large industrial process systems. In practice, the qualitative state of a process variable may only be assessed by considering the qualitative state of the remaining variables in a process [58], [91]. Consequently, a multivariate approach to diagnosis that is dependent on information that is assessed with a univariate method is liable to produce incorrect results.

- **Information overload:** Advances in electronics and sensor technologies have made monitoring systems and devices cheaper, easier to implement, and capable of monitoring previously unobservable process states. As a result, modern industrial process systems can monitor a large number of process states, which increases the amount of process information available as data. A system comprising a deep integration of computer systems with physical processes is termed a cyber-physical system [79], [80]. Cyber-physical systems are one of the leading innovative fronts in Industry 4.0 [142]. Coupled with the fact that industrial systems increase in size over time to increase production, the number of sensors and computers connected to every modern industrial process has increased significantly over the past decades. Consequently, not

only has the complexity of systems increased but so has the amount of process information that is provided to an operator in real-time. An overload of information may prove difficult to comprehend and act upon at the onset of an abnormal event. This slows down diagnosis and increases the risk of incorrect diagnosis [41]. It is also the case that an operator is exposed to more alarms during nominal operation that occur due to poorly selected alarm limits. In such situations, alarms become useless distractions, and poses the risk that an operator may doubt the information provided to them. Several incident reports have highlighted scenarios of information overload and ignored alarms as one of the root causes for the incidents [29] [48]. Industrial statistics show that 70% of industrial accidents are caused by human errors, some of which are attributed to poorly managed alarm system [131].

Venkatasubramanian et al. [133] postulate that a successful diagnostic system for a large industrial process system is a hybrid of three diagnostic components: (a) an evidential-based quantitative method for detection; (b) an evidential-based qualitative method for explicitly assessing abnormal process trends; and (c) a knowledge-based qualitative method for root-cause analysis. Alarm systems address the first and second diagnostic component, whereas causal reason address the third diagnostic component. However, as discussed above, the integration of alarm systems with causal reasoning can lead to incorrect diagnoses. The main objective of this work is develop a method that meets the first and second diagnostic components and simultaneously addresses the limitation of alarm systems. The goals of the project were defined as follows:

1. Develop an evidential-based quantitative method for establishing a multivariate diagnostic model that detects abnormal events occurring in complex processes;

2. Develop an evidential-based qualitative method that evaluates the trends of abnormal process variables with an established diagnostic model.

Based on these goals, the project scope includes:

1. The diagnostic model is to detect abnormal events in an unsupervised manner: the model is established by learning the features of historical process data, with detection consisting of comparing the features of new observations against those learned from nominal data. The motive for choosing this approach over supervised anomaly detection is that the latter requires a sufficient amount of data for every possible abnormal event, which is difficult to acquire by because (a) repeating occurrences of certain faults are rare; and (b) a single fault may have varying effects that depend on the current state of the process.

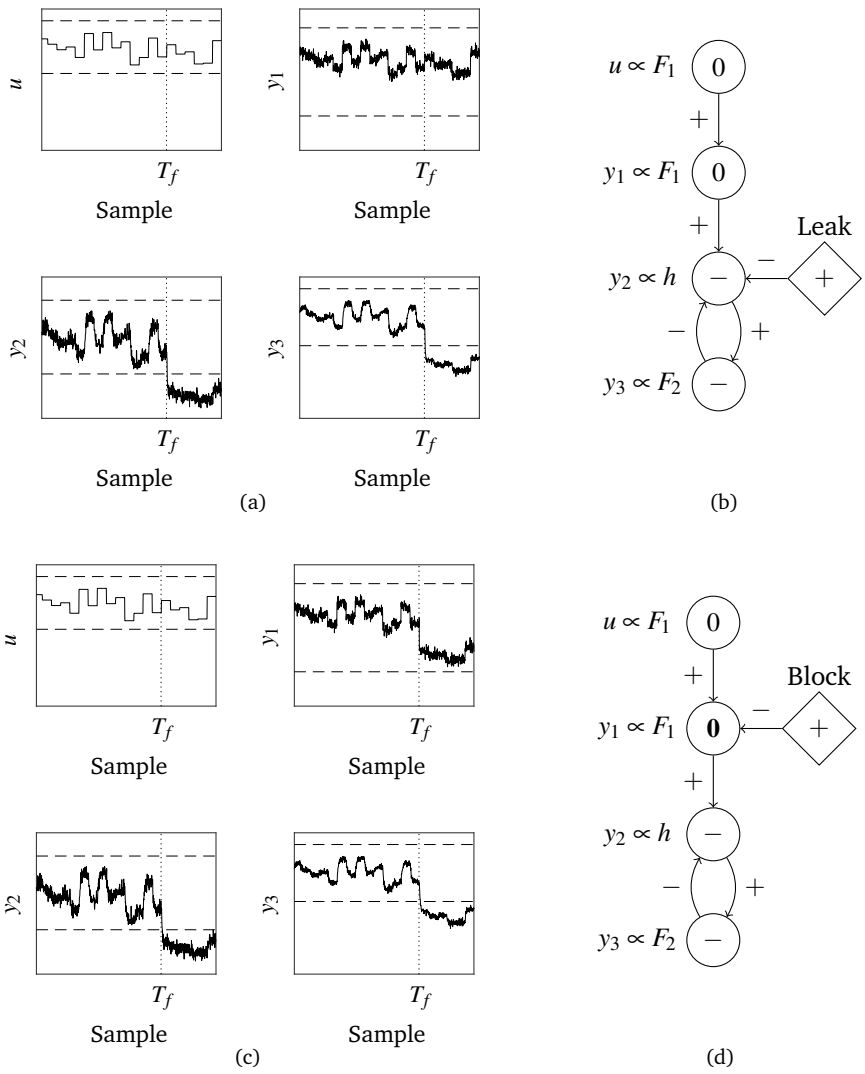


Figure 1.6: Diagnosis for single tank system with (a)-(b) leakage in the tank and (c)-(d) blockage in the inlet flow. Evidence is not obtained for F_1 in (c)-(d) due to incorrect limits for y_1, causing to the symptomatic information in (d) to be equivalent to (b).

2. To facilitate the diagnosis of abnormal events that have not occurred before, abnormal changes in process variables are evaluated in an unsupervised manner: variable trends are assessed with the diagnostic model rather than comparing them with the trends of previous cases of abnormal events.

3. The proposed methods must be suited for nonlinear and dynamic processes.

1.4 Thesis outline

This thesis is written as a collection of publications. Main results were submitted to journals and peer-reviewed conference proceedings over the course of the Ph.D. study. The main body of this dissertation offers a comprehensive summary of the state-of-the-art in evidential-based quantitative diagnosis. The chapters are:

Chapter 2 introduces the concepts behind a multivariate approach to evidential-based quantitative diagnosis. The chapter explains that the approach consists of two parts: (a) the discovery of a low-dimensional principal manifold that provides a joint summary for the distribution of a high-dimensional process variable space; and (b) detecting abnormal abnormal process variable observations by referring them against the principal manifold.

Chapter 3 provides a summary of the most common state-of-the-art methods for finding principal manifolds, detecting abnormal events, and isolating abnormal process variables. The chapter highlights some their limitations. The chapter motivates the method that was adopted in the research project.

Chapter 4 provides a summary of all main contributions that are provided in **Appendices A-E**

Chapter 5 concludes the thesis and provides a summary of its overall contribution to AEM, and proposes research areas for future study.

Chapter 2

Multivariate evidential-based abnormal event diagnosis

This chapter introduces a multivariate approach to evidential-based abnormal event diagnosis. The chapter begins with an introduction to principal manifolds, which are low-dimensional representations that provide a summary of the joint behavior of high-dimensional variable distributions. The discovery of principal manifolds serve as a prelude to feature extraction - a numerical process for deriving a new set of low-dimensional principal variables that retain informative properties of an original, high-dimensional variable space. The chapter ends with the application of feature extraction for abnormal event diagnosis.

2.1 Principal manifolds

Consider a data set consisting of 30 observations of two variables x_1 and x_2 shown in a scatter plot in Figure 2.1(a). It is sometimes the case that one wishes to summarize the joint pattern exhibited by the samples. One approach is to treat one of the variables as a response variable and the other as an explanatory variable. The task is to construct a linear regression model that predicts the response from the explanatory variable. Figure 2.1(a) shows the prediction of x_2 modeled as a linear function of x_1, estimated by least squares. This estimation approach is equivalent to finding the sum of squared deviations along the response variable x_2.

One may not always have a preference for which variable is treated as either a response or explanatory, but would still like to summarize the joint behaviour between the variables. Figure 2.1(b) shows the prediction of x_1 modeled as a linear function of x_2, as well as a dashed line that corresponds to the linear regression model in Figure 2.1(a). The figure shows that assigning a different variable as

the response leads to a noticeably different joint summary. Figure 2.1(c) shows a straight line that treats the variables symmetrically; it is found by minimizing the sum of squared orthogonal deviations. The dashed lines correspond to the regression models in Figures 2.1(a) and 2.1(b). It can be seen that the line that treats x_1 and x_2 symmetrically sets a compromise between the two regression models.

A straight line provides an accurate symmetric summary provided that the variables are linearly related. The line must be generalized as a nonlinear function if it is to provide an accurate summary of nonlinearly related variables. Instead of

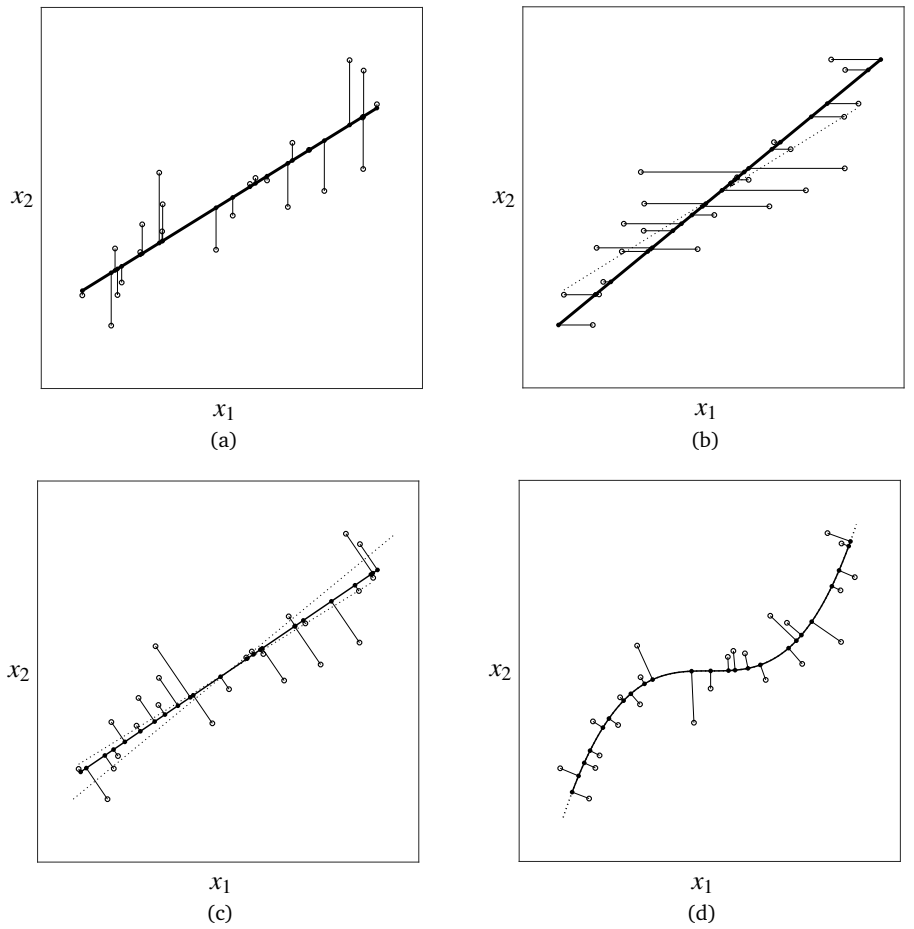


Figure 2.1: Linear regression that minimizes the sum of squared deviations in the response variable (a) y_2 and (b) y_1. (c) A straight line that minimizes the sum of squared deviation in variables y_1 and y_2. (d) A smooth curve that minimizes the sum of squared deviations in variables y_1 and y_2.

summarizing the data with a straight line, a smooth curve is used. Such a curve passes pass through the *middle* of the data in a smooth way, whether or not the data is linearly related. Figure 2.1(d) shows a nonlinear curve that summarizes the nonlinear joint behaviour between x_1 and x_2. The curve treats the variables symmetrically. Such a curve, called a **principal curve**, minimizes the orthogonal distance between itself and the samples.

A principal curve is a smooth, one-dimensional curve that passes through the center of a m dimensional data set [47]. Its shape minimizes the squared deviations in all variables to the curve and provides a nonlinear summary of the data. The curve is a vector $\mathbf{f}(\lambda)$ of m functions of a single variable λ. These functions are called coordinate functions, with λ parameterizing the curve and providing an ordering along it. Let $\mathbf{x} \in \mathbb{R}^m$ be continuous random vector. The curve $\mathbf{f}$ is called a principal curve of $\mathbf{x}$ if:

$$E(\mathbf{x}|\lambda_{\mathbf{f}}(\mathbf{x}) = \lambda) = \mathbf{f}(\lambda) \tag{2.1}$$

where the projection index $\lambda_{\mathbf{f}} : \mathbb{R}^m \to \mathbb{R}^1$ is defined as

$$\lambda_{\mathbf{f}} = \sup_{\lambda}\{\lambda : ||\mathbf{x} - \mathbf{f}(\lambda)|| = \inf_{\mu}||\mathbf{x} - \mathbf{f}(\mu)||\} \tag{2.2}$$

The projection index $\lambda_{\mathbf{f}}(\mathbf{x})$ is the value of λ for which $\mathbf{f}(\lambda)$ is closest to $\mathbf{x}$. The definition of a principal curve in two dimensions is illustrated in Figure 2.2. The smooth curve $\mathbf{f}(\lambda)$ is a principal curve if it traverses through a series of projection points that minimize the sum of squared deviations of samples that project orthogonally to $\mathbf{f}(\lambda)$.

It is noteworthy to point out that the definition of a principal curve does not imply that each sample $\mathbf{x}$ has a unique projection index $\lambda_{\mathbf{f}}(\mathbf{x})$. In other words, a single projection point along $\mathbf{f}(\lambda)$ may be shared by multiple samples. Figure 2.2 shows 96 observations summarized by a principal curve consisting of 26 projections, where each point is shared by four observations. Multiple samples having a scaled form of a common projection vector is attributed to common cause multivariate variations occurring in the direction of the projection vector.

Figure 2.3(a) shows a three dimensional data set summarized by the principal curve $\mathbf{f}(\lambda)$. The same concepts apply as for the two-dimensional example - the principal curve passes through a series of projection points that minimize the sum of squared deviations of the three-dimensional samples that project there orthogonally.

The concept of a principal curve can be generalized to higher-dimensional manifolds. Consider the case of a principal surface - a smooth, two-dimensional surface that passes through the center of $\mathbf{x} \in \mathbb{R}^m$ where $m \geq 3$. The surface is a vector $\mathbf{f}(\boldsymbol{\lambda})$

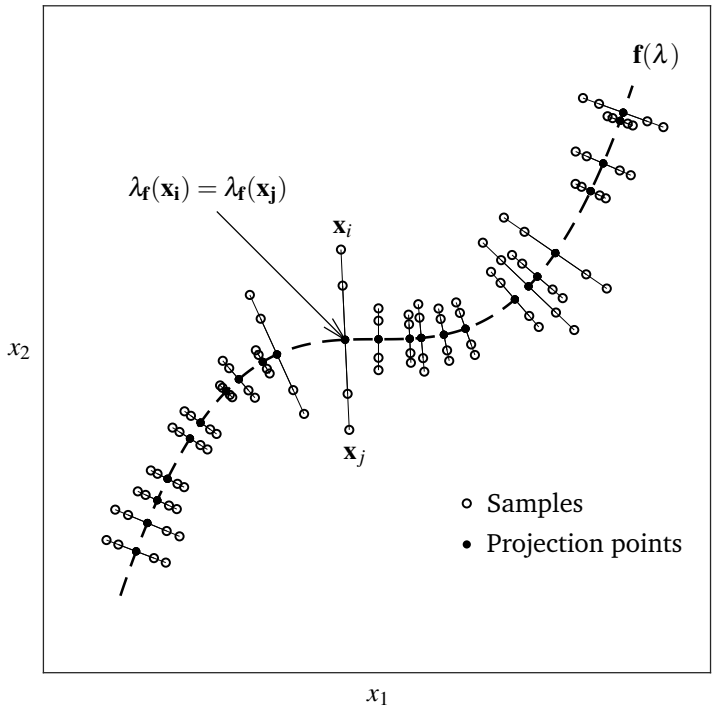


Figure 2.2: Principal curve $\mathbf{f}(\lambda)$ given a set of samples for $\mathbf{x}$. The points $\mathbf{x}_i$ and $\mathbf{x}_j$ share a projection point, and their projection vectors are a scalable version of each other.

of m continuous function of two variables λ_1 and λ_2:

$$\mathbf{f}(\boldsymbol{\lambda}) = \begin{bmatrix} f_1(\lambda_1, \lambda_2) \\ f_2(\lambda_1, \lambda_2) \\ \vdots \\ f_m(\lambda_1, \lambda_2) \end{bmatrix} \tag{2.3}$$

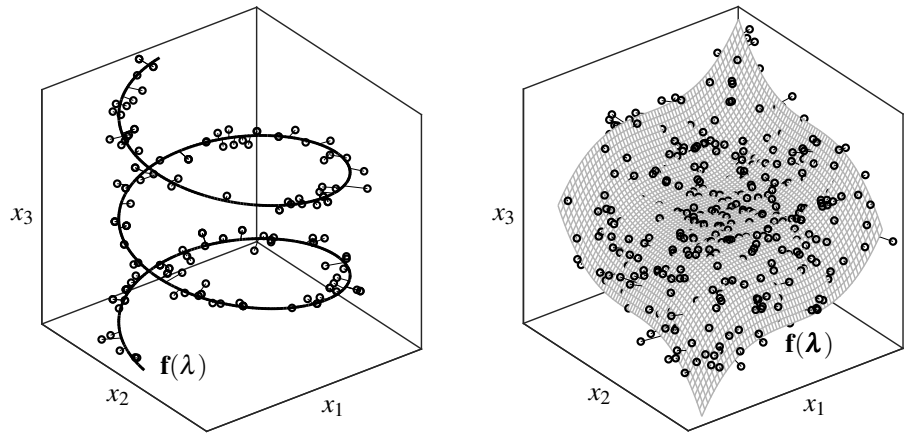
As before, let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random vector. The surface $\mathbf{f}$ is a principal surface of $\mathbf{x}$ if:

$$E(\mathbf{x}|\boldsymbol{\lambda}_{\mathbf{f}}(\mathbf{x}) = \boldsymbol{\lambda}) = \mathbf{f}(\boldsymbol{\lambda}) \tag{2.4}$$

Here the projection index $\boldsymbol{\lambda}_{\mathbf{f}}(\mathbf{x})$ is the value of $\boldsymbol{\lambda}$ for which the point on the surface $\mathbf{f}(\boldsymbol{\lambda})$ is closest to $\mathbf{x}$. Figure 2.3(b) illustrates a principal surface $\mathbf{f}(\boldsymbol{\lambda})$ summarizing a three-dimensional data set.

A **principal manifold** is defined as q-dimensional manifold that jointly summarizes the continuous random vector $\mathbf{x} \in \mathbb{R}^m$, where $q < m$. Principal manifolds summarize the *principal axis of variance* of a m-dimensional variable distribution. The manifold $\mathbf{f}$ is called a principal manifold of $\mathbf{x}$ if:

$$E(\mathbf{X}|\boldsymbol{\lambda}_{\mathbf{f}}(\mathbf{X}) = \boldsymbol{\lambda}) = \mathbf{f}(\boldsymbol{\lambda}) \tag{2.5}$$



(a) (b)

Figure 2.3: (a) Principal curve $\mathbf{f}(\lambda)$ given a set of samples for a three dimensional variable distribution and (b) principal surface $\mathbf{f}(\boldsymbol{\lambda})$ given a set of samples for a three dimensional variable distribution. Projection vectors are included.

where the projection index $\boldsymbol{\lambda}_\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^q$ is defined as

$$\boldsymbol{\lambda}_\mathbf{f} = \sup_{\boldsymbol{\lambda}} \{\lambda : ||\mathbf{x} - \mathbf{f}(\boldsymbol{\lambda})|| = \inf_{\boldsymbol{\mu}} ||\mathbf{x} - \mathbf{f}(\boldsymbol{\mu})||\} \tag{2.6}$$

The manifold is a a vector $\mathbf{f}(\boldsymbol{\lambda})$ of m continuous function of q variables:

$$\mathbf{f}(\boldsymbol{\lambda}) = \begin{bmatrix} f_1(\lambda_1, \lambda_2, \ldots, \lambda_q) \\ f_2(\lambda_1, \lambda_2, \ldots, \lambda_q) \\ \vdots \\ f_m(\lambda_1, \lambda_2, \ldots, \lambda_q) \end{bmatrix} \tag{2.7}$$

As was the case for principal curves, the definition of a principal manifold does not imply that each sample $\mathbf{x}$ has a unique projection index $\boldsymbol{\lambda}_\mathbf{f}(\mathbf{x})$ - a single projection point on the manifold $\mathbf{f}(\boldsymbol{\lambda})$ may be shared by multiple samples.

Given a specified dimension q, the task at hand is to construct a numerical model that estimates a q-dimensional principal manifold given a set of n observations of a m-dimensional variable vector $\mathbf{x}$. The original approach proposed by Hastie and Stuetzle [47] is to (a) initialize the principal manifold $\mathbf{f}(\boldsymbol{\lambda})$ as a linear function of an initialized distribution $\boldsymbol{\lambda}$ and then (b) iteratively improve the projection index $\boldsymbol{\lambda}_\mathbf{f}$ and redefine the distribution $\boldsymbol{\lambda}$ until the condition in Eq. (2.5) is satisfied. Estimation accuracy is largely determined by the complexity of the model $\mathbf{f}(\boldsymbol{\lambda})$, such as the number of available parameters in the coordinate functions in $\mathbf{f}(\lambda)$. However,

obtaining an estimate for principal manifold may also be formulated as a feature extraction problem.

2.2 Dimensionality reduction

Dimensionality reduction is a numerical technique for transforming a high-dimensional variable space into a low-dimensional feature space with the constraint that the reduced dimension retains informative properties of the original space. It is typically employed as an initial data-processing step to identify salient properties of data that ultimately improve the performance of subsequent tasks, such as regression and classification [109]. Approaches are divided into (a) feature selection: the process of selecting a subset of relevant variables; and (b) feature extraction: the process of deriving a new set of principal variables [107]. Consider the continuous random vector $\mathbf{x} \in \mathbb{R}^6$. An example of feature selection is the following isolation of variables:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \rightarrow \begin{bmatrix} x_2 \\ x_4 \\ x_5 \end{bmatrix} \tag{2.8}$$

Here the features x_2, x_4, and x_5 are simply a subset of the variable vector $\mathbf{x}$. The central motivation for using feature selection is that the original space contains variables that are either redundant due to variable correlations, i.e., $x_i \approx x_j$ for $i \neq j$ or $x_i = x_j + e_{ij}$ for $i \neq j$, or irrelevant for the required analysis, and can thus be dismissed without inducing significant loss of information. Feature extraction attempts to retain the entire original variable space to derive new features:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} \rightarrow \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ g_3(\mathbf{x}) \end{bmatrix} \tag{2.9}$$

Here, each feature z_i is a function g_i of the original variable vector $\mathbf{x}$.

If the continuous random vector $\mathbf{x} \in \mathbb{R}^m$ comprises variables that are treated neither as a response nor explanatory, then the feature extraction $\mathbf{z} = \mathbf{g}(\mathbf{x})$ is the process of finding a q-dimensional principal manifold $\mathbf{f}(\boldsymbol{\lambda})$ that summarizes $\mathbf{x}$, where

$\mathbf{z} \triangleq \boldsymbol{\lambda}$. Let $\mathbf{x} \in \mathbb{R}^{m\times 1}$ be a vector of continuous random variables. An optimal mapping to the feature space $\mathbf{z} \in \mathbb{R}^{q\times 1}$ is sought in the form:

$$\mathbf{z} = \underline{E}(\mathbf{x}) \tag{2.10}$$

where $\underline{E}$ is a vector function, composed of q individual functions; $\underline{E} = [E_1, E_2, \ldots, E_q]^\intercal$ such that if z_i represents the ith element of $\mathbf{z}$,

$$z_i = E_i(\mathbf{x}) \tag{2.11}$$

The optimal mapping $\mathbf{z} = \underline{E}(\mathbf{x})$ is difficult to derive if one lacks a set of algebraic equations that describe the relationship between the variables in $\mathbf{x}$. One solution is to augment feature extraction with a subsequent feature expansion task where the objective is to reconstruct, i.e., estimate, the original variable space $\hat{\mathbf{x}} \in \mathbb{R}^{m\times 1}$ from a transformation of $\mathbf{z}$. The inverse transformation that reconstructs the original variable space is implemented by a second vector function $\underline{D} = [D_1, D_2, \ldots, D_m]^\intercal$:

$$\hat{\mathbf{x}} = \underline{D}(\mathbf{z}) \tag{2.12}$$

The objective is for $\mathbf{z}$ to retain sufficient information about $\mathbf{x}$ that permits an accurate reconstruction $\hat{\mathbf{x}}$. The vector functions $\underline{E}$ and $\underline{D}$ are selected to minimize the loss of information represented by the following loss function:

$$\mathscr{L}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{2.13}$$

Minimization of Eq. (2.13) consists of minimizing the sum of squared deviations between the original variables $\mathbf{x}$ and reconstructions $\hat{\mathbf{x}}$. With respect to Figure 2.2, this corresponds to minimizing the squared distance between samples and projections. In principle, the process of deriving the transformations in Eqs. (2.10) and (2.12) constitues the estimation of a q-dimensional manifold described in Eq. (2.5) where (a) feature extraction $\underline{E}(\mathbf{x})$ corresponds to the projection index $\boldsymbol{\lambda}_\mathbf{f}$; and (b) feature expansion $\underline{D}(\mathbf{z})$ corresponds to the function $\mathbf{f}(\boldsymbol{\lambda})$.

2.3 Multivariate Quality Control Methods

Most industrial alarm systems monitor the qualitative state of process variables with univariate control charts such as Schewart, CUSUM, and EWMA. The charts examine the variables independently, and it is assumed that abnormal event that affects multiple process variables is reflected in the individual charts. However, the assumption of independence between abnormal process variables is rarely met. It is often the case that the qualitative state of an abnormal variable is only inferrable by considering the qualitative state of the remaining variables in the system [58], [91].

The difficulty with using univariate control charts to diagnose faults occurring in multivariate systems is illustrated by reference to Figure 2.4. Two process variables (x_1, x_2) are considered for ease of illustration. The joint plot of (x_1, x_2) shows that samples for x_1 and x_2 follow a multivariate Normal distribution and are negatively correlated when the process is operating under nominal conditions. The ellipse encapsulates 99.73% of the nominal samples. Unexplained common cause variation is attributed to process and measurement stochasticity. The nominal samples are also plotted as individual time series charts with their corresponding 99.73% upper (UCL) and lower (LCL) control limits. Abnormal samples gathered from when the process operated under an abnormal condition that terminated the correlation between x_1 and x_2 are included in the individual charts and the joint plot. By inspection of each of the univariate charts, each variable appears to be consistent with nominal operating conditions since the abnormal samples are within the control limits. On the other hand, the multivariate joint plot shows that some of the abnormal samples are outside of the joint control region. Therefore, an indication of the abnormal process condition is only revealed in the joint plot and not in the univariate charts, where abnormal samples are incorrectly considered nominal. This improvement in detectability is understood by considering the effect the correlation between x_1 and

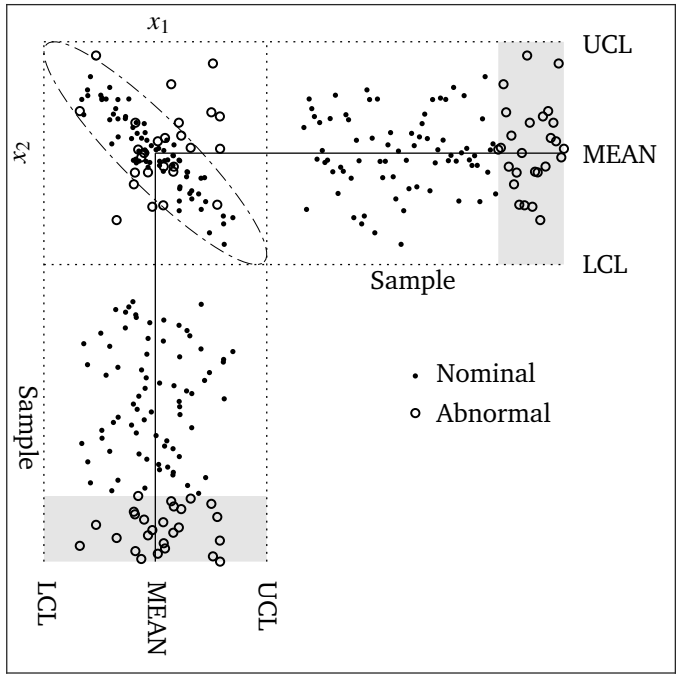


Figure 2.4: Monitoring of two variables depicting the misleading nature of univariate charts.

x_2 has on forming a joint control region. When the correlation is not considered, the control region would have otherwise been the square shaped contour parameterized by the UCLs and LCLs of x_1 and x_2, wherein the abnormal samples reside and remain undetected. Incorporating the correlation between x_1 and x_2 compresses the square shaped region into an ellipse, thereby constricting the limit that classifies a sample as abnormal. Evaluating the qualitative trends of abnormal variables thus consists of assessing the movement of samples that reside outside the joint control region.

Detecting and evaluating abnormal events with a multivariate correlation approach is rooted in the discovery of principal manifolds. A principal curve is the only possible manifold for the example in Figure 2.4. Figure 2.5 displays the joint plot of nominal samples from Figure 2.4. The principal curve is a straight line that traverses through the middle of the samples and points in the direction of joint maximal variance. Included in the plot are two abnormal samples that reside outside the 99.73% joint control region. It is necessary to point out that a suitable choice for a joint control region that summarizes the joint behavior of nominal process variables is strictly dependent on the distribution of samples. In the case of the samples in Figure 2.5, an elliptical control region is suitable because the samples follow a multivariate Normal distribution with a correlation index of $\rho_{y_1,y_2} = -0.91$. However, not all data distributions follow the assumption of normality, and so an elliptical control region may not always be viable.

Fault detection is the first step in multivariate process monitoring. The Hotelling T^2 statistic and the squared prediction error (SPE) statistic (also known as the Q statistic) are used for assessing the condition of a process. The SPE and T^2 statistics are illustrated in Figure 2.5. These multivariate indices are preferable to univariate indices because the correlation between variables is taken into account. Although both the Hotelling T^2 and SPE statistics are used for detecting abnormal events, it is necessary to point out that they measure different statistical properties of variable distributions, and their roles in process monitoring are not symmetric. The SPE statistic is representative for the Euclidean distance between a sample and its projection on the principal manifold, and is used as an indicator for variability that breaks nominal process variable correlations. The T^2 statistic is representative for the distance between a sample and the origin of the principal manifold, and is used as an indicator for process variables that lie outside of previously seen operating limits. The control limits CL_{SPE} and CL_{T^2} define the bounds of nominality, and a process is considered nominal if:

$$\begin{aligned} SPE &\leq CL_{SPE} \\ T^2 &\leq CL_{T^2} \end{aligned} \tag{2.14}$$

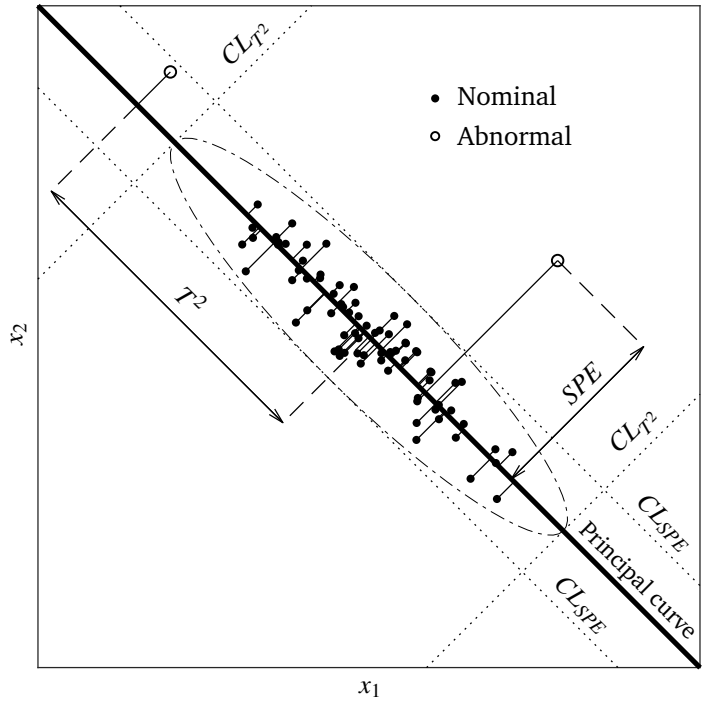


Figure 2.5: Illustration of monitoring statistics and their respective control limits.

Note that CL_{SPE} and CL_{T^2} correspond to the edges defined by an elipsoid encapsulating Normally distributed samples (Figure 2.5). Although the statistics appear frequently in the literature, SPE is generally favored over T^2 for three reasons [63]:

1. The region defined by CL_{T^2} is associated with explained (nominal) variation occurring in the process, whereas the region defined by CL_{SPE} is associated with unexplained common cause variation. It is normally the case that the variance of process variables is more associated with explained variation. Consequently, the region defined by CL_{T^2} is usually much larger than that of CL_{SPE}, and it usually takes a more severe fault for the T^2 statistic to cross its limit. Additionally, a sample that exceeds the CL_{T^2} limit but not the CL_{SPE} limit implies that the nominal correlation structure among variables is retained but that the sample has shifted significantly far away from the origin of the principal manifold. This could be an indicator of an abnormal event, but could also be a attributed to a nominal sample that was not available when establishing the CL_{T^2} control region. Furthermore, the region defined by CL_{T^2} tends to coincide with the UCL and LCL of traditional univariate charts; thus the T^2 statistic offers little improvement in fault detectability compared to the SPE statistic which detects alterations in nominal process variable correlations.

2. Abnormal event detection with the T^2 statistic requires for the principal manifold to be defined beyond the CL_{T^2} limit. This is an easy task for linearly correlated variables such as with the example in Figure 2.5, where the principal curve (a straight line) is simply extended beyond the CL_{T^2} limit. However, this task can be difficult for nonlinearly correlated variables such as the example illustrated in Figure 2.6(a). It is often the case that few to no nominal samples exist past the CL_{T^2} limit; hence defining a principal manifold may be ill-posed.

3. A CL_{T^2} limit may be undefinable for certain data distributions, such as for the nominal samples shown in Figure 2.6(b). The cyclical nature of the data distribution denies the extension of the principal curve and, to the same extent, the existence of a CL_{T^2} limit. However, a CL_{SPE} limit is definable.

Multivariate process monitoring consists of verifying that a principal manifold is representable for new observations. New observations $\mathbf{x}_{new}$ are transformed via Eqs. (2.10) and (2.12) to generate the features $\mathbf{z}_{new}$ and reconstructions $\hat{\mathbf{x}}_{new}$. The Hotelling T^2 statistic is given by:

$$T^2 = \sum_{i=1}^{q} \frac{z_{new,i}^2}{\bar{\sigma}_i^2} \tag{2.15}$$

where $\bar{\sigma}_i$ is the sample standard deviation of feature variable z_i. The SPE statistic is:

$$\begin{aligned} SPE &= ||\mathbf{x}||^2 \\ &= \sum_{i=1}^{m} (x_{new,i} - \hat{x}_{new,i})^2 \end{aligned} \tag{2.16}$$

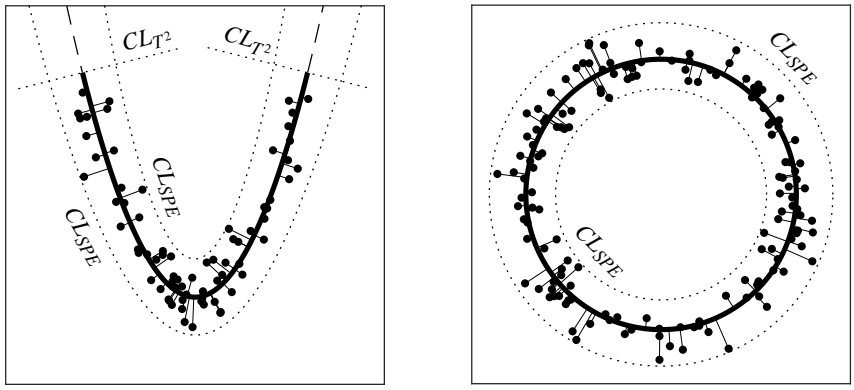


Figure 2.6: Principal curve for (a) a quadratic variable distribution and (b) a cyclical variable distribution. Monitoring statistics and respective control limits are included.

An abnormal event is detected if either statistic no longer satisfies the inequalities in Eq. (2.14). The qualitative changes in abnormal process variables are then evaluated by assessing the multivariate trends in the anomaly-suggestive statistic(s).

2.4 Relation to analytical redundancy

Feature extraction as an evidential-based quantitative approach to fault diagnosis is comparable to (though not equivalent to) the knowledge-based quantitative approach given by analytical redundancy. Consider the following system of linear equations describing the flow of fluid through a liquid cooling unit:

$$\begin{aligned} h &= N(0,1) \\ y_1 &= h + e_1 \\ y_2 &= h + e_2 \end{aligned} \tag{2.17}$$

where h is the flow through the cooling unit, y_1 is the measured inlet flow rate, y_2 is the measured outlet flow rate, e_i denotes a noise term for measurement y_i, and $N(\mu,\sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. The following process variable vector is defined:

$$\mathbf{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{2.18}$$

A residual based on analytical redundancy is the difference between x_1 and x_2:

$$z_1 = x_1 - x_2 \tag{2.19}$$

Substituting Eqs. (2.18) and (2.17) into Eq. (2.19) yields:

$$\begin{aligned} z_1 &= (h + e_1) - (h + e_2) \\ &= e_1 - e_2 \end{aligned} \tag{2.20}$$

In principle, z_1 models the difference between the noise terms e_1 and e_2 when the cooling system is operating nominally. In other words, the variance of z_1 is attributed to stochasticity present in the measurements. Coincidentally, z_1 does not permit a reconstruction of the original variables, as any information about h is lost in z_1. Still, z_1 is sensitive to faults that alter the balance between the two flow rates; a significant shift away from the mean of z_1, i.e., zero, would imply the onset of a fault.

A feature based on feature extraction is the following weighted sum of x_1 and x_2:

$$\begin{aligned} z_2 &= \frac{x_1}{\sqrt{2}} + \frac{x_2}{\sqrt{2}} \\ &= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{2.21}$$

Substituting Eqs. (2.18) and (2.17) into Eq. (2.21) yields:

$$\begin{aligned} z_2 &= \frac{h+e_1}{\sqrt{2}} + \frac{h+e_2}{\sqrt{2}} \\ &= \frac{2h}{\sqrt{2}} + \frac{e_1+e_2}{\sqrt{2}} \end{aligned} \tag{2.22}$$

Unlike the residual z_1, the variance of z_2 is attributed to nominal variations in the flow rate h as well as the noise terms e_1 and e_2. This means that it is possible to reconstruct the original variables from z_2 since the feature retains information about the process state h. Consider the following reconstruction for $\hat{x}_1$ and $\hat{x}_2$:

$$\begin{aligned} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} &= \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} z \\ &= \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{aligned} \tag{2.23}$$

The reconstruction mapping in Eq. (2.23) is given by the transpose of the feature mapping in Eq. (2.21). The rationale behind Eq. (2.23) is self-evident: since x_1 and x_2 are positively correlated, one can reconstruct either variable by taking the average of x_1 and x_2. Substituting Eqs. (2.18) and (2.17) into Eq. (2.23) yields:

$$\begin{aligned} \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} &= \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} h+e_1 \\ h+e_2 \end{bmatrix} \\ &= \begin{bmatrix} h+0.5(e_1+e_2) \\ h+0.5(e_1+e_2) \end{bmatrix} \end{aligned} \tag{2.24}$$

The reconstructions retain the flow rate h, and correspond to the original variables. They only difference between the original variables and the reconstructions is that noise terms e_1 and e_2 are uncorrelated in the former and correlated in the latter.

The example above demonstrates the key difference between abnormal event diagnosis with analytical redundancy and feature extraction: the former produces a residual that corresponds to the mean and variance of unexplained process and measurement variations, while the latter produces a feature that corresponds to the mean and variations of explained *and* unexplained process and measurement variations. Correspondingly, features, which, unlike residuals, retain information about nominal process variation, permit a reconstruction of the original variables.

2.5 The unsupervised learning problem

In the absence of *a priori* knowledge on the relations among variables in the continuous random vector **x**, feature extraction is formulated as an unsupervised machine learning problem: the objective is to find the vector functions $\underline{E}$ and $\underline{D}$ that minimize the loss function in Eq. (2.13), where $\underline{E}$ and $\underline{D}$ are parameterized by a mathematical model. Unsupervised learning is used for other purposes besides feature extraction, such as data compression, cluster analysis, and factor analysis. Unsupervised learning can identify the underlying structure of data that facilitates its use in a subsequent task such as regression or classification [33]. Figure 2.7(a) illustrates a traditional approach for building and evaluating a mathematical model trained to compress handwritten digits in an unsupervised manner. The model is initially defined for the learning task, and is then trained to fit its model parameters to a *training data set* - a set of training examples sampled from the data distribution (handwritten digits). The model, now fitted, then makes predictions for the observations in a *validation data set* - a set of validation examples that provide an unbiased evaluation of the model's performance; if the evaluation is not satisfactory, then the model is retrained with new model constraints. If model evaluation is satisfactory, then the model makes predictions for the observations in a *test data set* - a set of test examples that provide an unbiased estimate of the final model performance.

The use of unsupervised learning for abnormal event diagnosis is significantly distinctive from other applications. Figure 2.7(b) illustrates the approach for building and evaluating a mathematical model trained to extract features from process variables. The figure shows that the training data set and validation data set consists of samples gathered from a process exhibiting nominal operating conditions. Coincidentally, the model is evaluated based on its performance at extracting features from nominal data. However, the figure indicates that the test data set consists of samples gathered from when the process exhibits both nominal and abnormal operating conditions. In other words, once the model is evaluated to satisfactorily extract feature from nominal samples, its final model performance is unbiasedly estimated with both nominal and abnormal samples. This significant contrast between the test and training/validation data sets is not traditional in machine learning. It is akin to, for example, training an image classifier to classify between several species of animals and then testing it with images of a new specie that is unlike the prior investigated species; the classifier would fail at classifying this new specie. Likewise, a feature extraction model will fail at extracting representative features and be unable to reconstruct the original variables for abnormal samples. However, it is this failure that permits abnormal event diagnosis; upon detecting

that the feature extraction model is not representative for new observations, it is concluded that the new observations must have been sampled from an abnormal system, and can thus be analyzed to reveal the nature of this abnormality.

Feature extraction-based anomaly diagnosis consists of three steps:

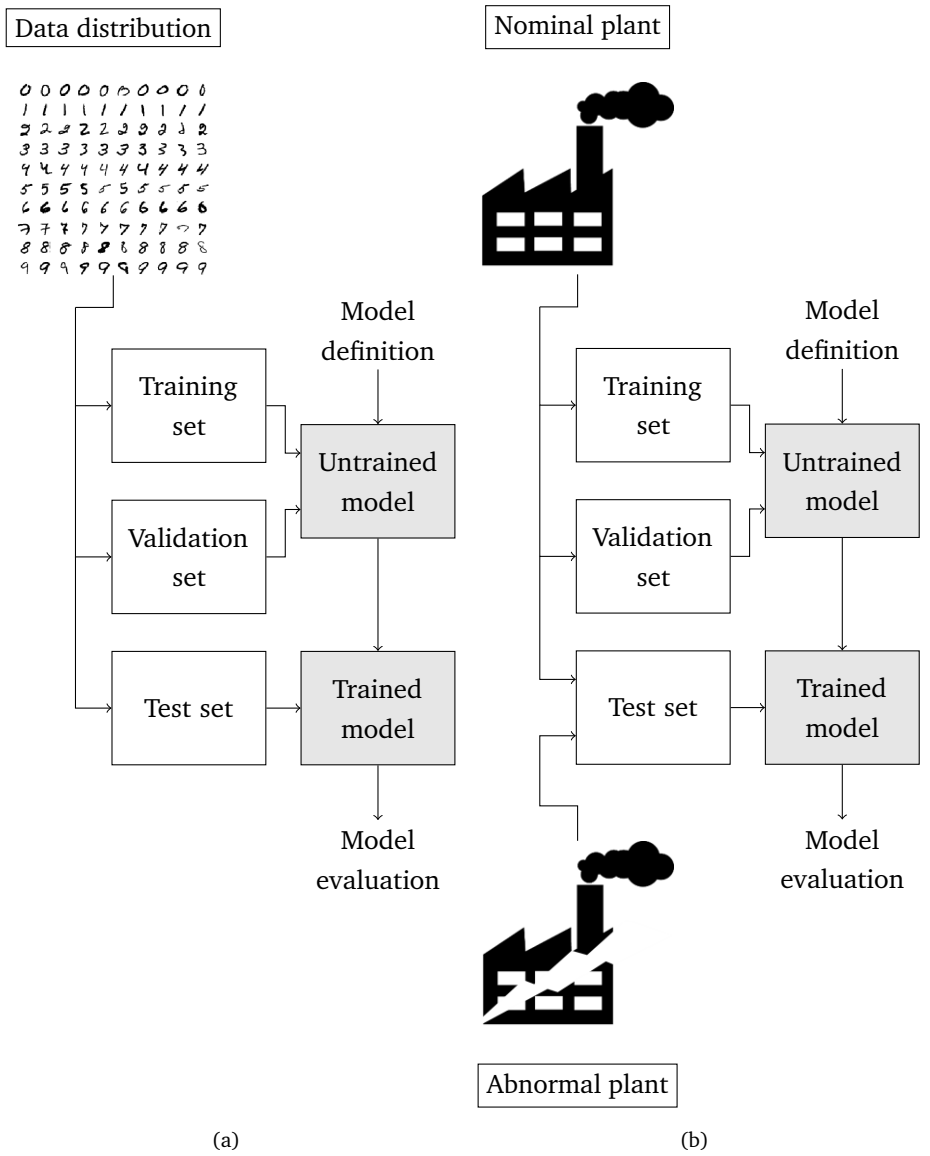


Figure 2.7: Comparison between (a) a traditional unsupervised learning problem and (b) an unsupervised learning approach to anomaly detection. Note that the test sets between (a) and (b) are different.

1. **Model creation**: A feature extraction model is optimized with a training data set $\mathbf{X}_t \in \mathbb{R}^{m \times n}$. The data set consists of historical process data, which is the collection of past n samples for m continuous process variables. These samples must be collected from a process exhibiting nominal operating conditions. By obtaining the transformations $\underline{E}$ and $\underline{D}$, the model learns redundancies and correlations present in nominal process variables.

2. **Fault detection**: A new observation for process variables $\mathbf{x}_{new}$ is propagated through the optimized model to generate the principal variables $\mathbf{z}_{new}$ and reconstructions $\hat{\mathbf{x}}_{new}$. The condition of the process is assessed by motoring the SPE and Hotelling T^2 statistics. A fault is deemed to have occurred if the control limits in Equations 2.14 are crossed.

3. **Fault isolation**: Once a fault is detected, the signal characteristics of the SPE and Hotelling T^2 statistics are analyzed to determine abnormal changes in process variables.

Chapter 3

State of the art

This chapter provides a summary of the fundamental theories and recent advances in feature extraction and its application to abnormal event diagnosis. The chapter begins with a comparative overview of unsupervised methods for feature extraction that are commonly referenced in the literature. The chapter details the importance of selecting an appropriate dimension for the feature space and its influence on multivariate monitoring statistics. A short summary is provided for the method of extracting features from dynamic, i.e., auto-correlated and cross-correlated, process variables. Approaches for defining multivariate control limits for different feature extraction methods are presented. Lastly, the chapter provides an overview of the state-of-the-art method for assessing abnormal trends in process variables.

3.1 Latent projection

Latent projection (LP) is a statistical approach to feature extraction. LP assumes that prior knowledge on the relations among variables in the continuous random vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ is unavailable. The task of LP is to obtain the vector functions $\underline{E}$ and $\underline{D}$ from a statistical analysis of samples for $\mathbf{x}$ such that, given the dimension q for the feature (latent) space, the squared difference $||\mathbf{x} - \underline{E}(\underline{D}(\mathbf{x}))||^2$ is minimized. The vector functions $\underline{E}$ and $\underline{D}$ parameterize a LP model. In the context of abnormal event diagnosis, the first step is to sample the process whilst it is consistent with nominal operating conditions. The n samples are collected in the *reference*, i.e., training, data matrix $\mathbf{X}_t \in \mathbb{R}^{m\times n}$:

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[n] \end{bmatrix} = \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[n] \\ x_2[1] & x_2[2] & \cdots & x_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ x_m[1] & x_m[2] & \cdots & x_m[n] \end{bmatrix} \tag{3.1}$$

The vector functions $\underline{E}$ and $\underline{D}$ are then given from a LP of $\mathbf{X}_t$. From Eq. (2.10), the training samples are transformed to the feature matrix $\mathbf{Z}_t \in \mathbb{R}^{q\times n}$:

$$\mathbf{Z}_t = \underline{E}(\mathbf{X}_t) \tag{3.2}$$

From Eq. (2.12), the feature matrix is transformed to the reconstruction matrix $\hat{\mathbf{X}}_t \in \mathbb{R}^{m\times n}$:

$$\hat{\mathbf{X}}_t = \underline{D}(\mathbf{Z}_t) \tag{3.3}$$

The validation data matrix $\mathbf{X}_v$ and test (fault) data matrix $\mathbf{X}_f$, as well as their respective feature matrices $(\mathbf{Z}_v, \mathbf{Z}_f)$ and reconstruction matrices $(\hat{\mathbf{X}}_v, \hat{\mathbf{X}}_f)$, are defined similarly. Note that $\mathbf{X}_f$ consists of nominal and abnormal samples.

3.1.1 Principal component analysis

Principal component analysis (PCA) is a widely used approach for establishing a linear LP model [114]. Within the class of linear methods, PCA delivers a model with the least loss of information [51], [104]. Many state-of-the-art LP methods are an extension of PCA. PCA consists of finding a set of principal component scores $t_i = \mathbf{p}_i^\intercal \mathbf{x}$ for $i \in \mathbb{Z} : i \in [1, m]$, where $\mathbf{x}$ is assumed to have mean of zero; if this is not true, then $\mathbf{x}$ may be mean-centered. The column vectors $\mathbf{p}_i$ form the orthonormal principal component loading matrix $\mathbf{P} \in \mathbb{R}^{m\times m}$. The first principal component score t_1 has maximum variance, the second principal component score t_2 has the next greatest variance, with additional scores up to m similarly defined. The scores t_i form the score vector $\mathbf{t} \in \mathbb{R}^{m\times 1}$, which is given by the transformation $\mathbf{t} = \mathbf{P}^\intercal \mathbf{x}$. Because the orthonormal matrix $\mathbf{P}$ is orthogonal, i.e., $\mathbf{P}\mathbf{P}^\intercal = \mathbf{P}\mathbf{P}^{-1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, the original variable vector $\mathbf{x}$ is reproducible via $\mathbf{x} = \mathbf{P}\mathbf{t}$. The loading matrix $\mathbf{P}$ is given by the eigenvectors of the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{x}$:

$$\mathbf{\Sigma} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\intercal \tag{3.4}$$

where $\mathbf{\Lambda}$ is a non-negative real diagonal $m \times m$ matrix whose diagonal elements λ_i are the corresponding eigenvalues of the loading vector $\mathbf{p}_i$. The eigenvalues λ_i are the variances of the principal component scores t_i, i.e., $\mathrm{Var}(t_i) = \lambda_i$. If $\mathbf{\Sigma}$ is not known, then it may be estimated with:

$$\mathbf{\Sigma} = \frac{\mathbf{X}_t^\intercal \mathbf{X}_t}{n-1} \tag{3.5}$$

LP with a model given by PCA involves identifying q principal components that explain most of the predictable variation in $\mathbf{x}$. The remaining $m-q$ principal components are attributed to unpredictable common cause variation. For that purpose, the loading matrix is partitioned as follows:

$$\mathbf{P} = \begin{bmatrix} \hat{\mathbf{P}} & \tilde{\mathbf{P}} \end{bmatrix}, \quad \hat{\mathbf{P}} \in \mathbb{R}^{m\times q}, \quad \tilde{\mathbf{P}} \in \mathbb{R}^{m\times(m-q)} \tag{3.6}$$

The variable vector $\mathbf{x}$ is then decomposed into the reconstruction vector $\hat{\mathbf{x}}$ and the residual vector $\tilde{\mathbf{x}}$:

$$\begin{aligned}\mathbf{x} &= \hat{\mathbf{x}} + \tilde{\mathbf{x}} \\ &= \hat{\mathbf{P}}\hat{\mathbf{t}} + \tilde{\mathbf{P}}\tilde{\mathbf{t}} \\ &= \hat{\mathbf{P}}\hat{\mathbf{P}}^{\intercal}\mathbf{x} + \tilde{\mathbf{P}}\tilde{\mathbf{P}}^{\intercal}\mathbf{x}\end{aligned} \tag{3.7}$$

The mapping to the latent vector $\mathbf{z}$ is:

$$\mathbf{z} = \hat{\mathbf{t}} = \hat{\mathbf{P}}^{\intercal}\mathbf{x} \tag{3.8}$$

The mapping to the reconstructions $\hat{\mathbf{x}}$ is:

$$\hat{\mathbf{x}} = \hat{\mathbf{P}}\mathbf{z} \tag{3.9}$$

3.1.2 Independent component analysis

The PCA of $\mathbf{x}$ computes a collection of m unit vectors $\mathbf{p}_i$ that constitute an orthonormal basis. The change of basis $\mathbf{t} = \mathbf{P}^{\intercal}\mathbf{x}$ warrants that the scores t_i are linearly uncorrelated with (orthogonal to) one another as long as the original variables $\mathbf{x}$ follow the assumption of multivariate normality [114]. Likewise, the scores t_i are correlated with one another if $\mathbf{x}$ does not follow the assumption of multivariate normality. From a geometric perspective, PCA can be thought of as fitting a m-dimensional ellipsoid to $\mathbf{x}$, where each axis i of the ellipsoid is represented by the direction $\mathbf{p}_i$ and length corresponding to λ_i. The ellipsoid is representable for $\mathbf{x}$ as long as $\mathbf{x}$ follows a multivariate Gaussian distribution. Coincidentally, a LP model given by PCA provides a suitable estimation for the principal manifold $\mathbf{z} = \hat{\mathbf{t}} \subseteq \mathbf{t}$ of $\mathbf{x}$ as long as $\mathbf{x}$ follows the assumption of multivariate normality.

The independent component analysis (ICA) of $\mathbf{x}$ consists of searching for a linear transformation that minimizes the statistical independence between its components [28], [53]. ICA is preferred over PCA if the variables in $\mathbf{x}$ do not follow a multivariate Gaussian distribution but remain linearly related. ICA was originally developed for blind source separation applications: the objective is to recover a set of independent source signals after that have been mixed by an unknown linear transformation [126]. The concept of ICA may be seen as an extension of PCA: PCA defines an orthogonal basis that aligns along the direction that best explains the variability of $\mathbf{x}$, whereas ICA defines a linear span (not necessarily orthogonal) that orients along the direction that best explains the hidden sources comprising $\mathbf{x}$. ICA proves useful if the variance of the process variable vector $\mathbf{x}$ is explained by a source, i.e., disturbance, that is not measured directly but has considerable influence on process variations. Coincidentally, if the variation in $\mathbf{x}$ is explained by an immeasurable source variable,

then an ICA-based LP model will generate features $\mathbf{z}$ that are unobtainable with PCA and are more suitable for estimating the principal manifold of $\mathbf{x}$.

To illustrate ICA, consider two independent source variables $s_1 = U\left(-\sqrt{3}, \sqrt{3}\right)$ and $s_2 = U\left(-\sqrt{3}, \sqrt{3}\right)$ that have the uniform distribution shown in Figure 3.1(a). The range of values for the uniform distributions are chosen to make the mean zero and the variance equal to one. The independent components IC_1 and IC_2 correspond to the source variables s_1 and s_2. The vectors represented by IC_1 and IC_2 form an orthogonal basis that summarizes the independence between s_1 and s_2. The sources are mixed as follows:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{3.10}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{3.11}$$

Figure 3.1(b) shows a scatter plot of the mixtures. The mixed data has a uniform distribution on a parallelogram with a correlation index of $\rho_{x_1,x_2} \approx 0.89$; the random variables x_1 and x_2 are therefore not independent. The independent components IC_1 and IC_2 are represented in the mixed space as $\mathbf{A}\mathrm{IC}_1$ and $\mathbf{A}\mathrm{IC}_2$: the two vectors define a linear span that explains the non-Gaussian variation comprising the mixed data. The principal components PC_1 and PC_2 are given by a PCA of the mixed data. PC_1 and PC_2 are represented as $\mathbf{A}^{-1}\mathrm{PC}_1$ and $\mathbf{A}^{-1}\mathrm{PC}_2$ in the original source space in Figure 3.1(a). Clearly, the vectors represented by $\mathbf{A}^{-1}\mathrm{PC}_1$ and $\mathbf{A}^{-1}\mathrm{PC}_2$ do not summarize the statistical independence between s_1 and s_2. The objective of the ICA

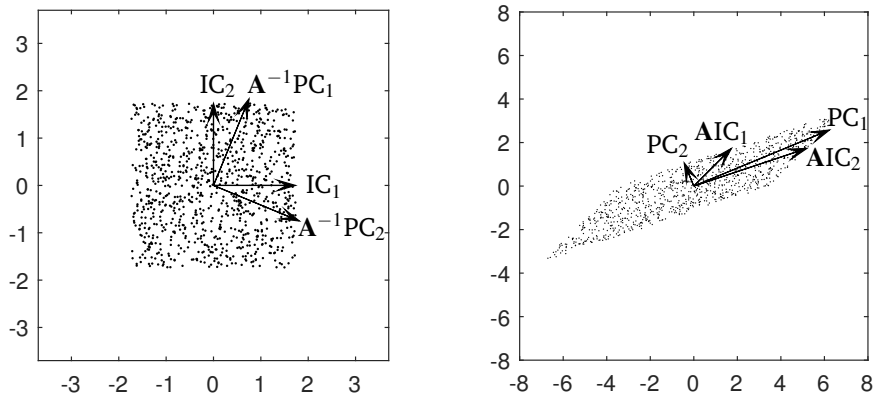


Figure 3.1: Scatter plots of (a) the source variables and (b) the mixed variables. 1000 samples are plotted.

of $\mathbf{x}$ is to determine $\mathbf{A}^{-1}$ that transforms the mixed space $\mathbf{x}$ to the source space $\mathbf{s}$, where the uniform variation in each source variable s_i corresponds to the vector represented by the independent component IC_i.

The example given above demonstrates that the independent components IC_1 and IC_2 are an appropriate representation of the statistical independence observed by the source signals, whereas the principal components PC_1 and PC_2 are not.

When solving for independent components, it is assumed that the mixed variable vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ can be expressed as a linear combination of the unobservable source variable vector $\mathbf{s} \in \mathbb{R}^{d\times 1}$, where $d \leq m$ (for simplicity, it is often assumed that $d = m$ [53]). The source and mixed variables are assumed to have means of zero; if this is not true, then $\mathbf{x}$ may be mean-centered. Given the reference data matrix $\mathbf{X}_t$, the relationship between the mixed and source variables is given by:

$$\mathbf{X}_t = \mathbf{A}\mathbf{S}_t + \mathbf{E}_t \tag{3.12}$$

where $\mathbf{A} \in \mathbb{R}^{m\times d}$ is the unknown mixing matrix, $\mathbf{S}_t \in \mathbb{R}^{d\times n}$ is the source matrix, and $\mathbf{E}_t \in \mathbb{R}^{m\times n}$ is the residual matrix. The task of ICA is to estimate the mixing matrix $\mathbf{A}$ and the original source matrix $\mathbf{S}$ from solely the observations $\mathbf{X}$. Without further constraints, it is impossible to identify both $\mathbf{A}$ and $\mathbf{S}$ from $\mathbf{X}$. Hence, one major assumption is that the source variables s_i are statistically independent with each other and that their variances are equal to one. The demixing matrix $\mathbf{W} \in \mathbb{R}^{d\times m}$ provides an estimate of the inverse of the mixing matrix $\mathbf{A}$, i.e., $\mathbf{W} \approx \mathbf{A}^{-1}$, such that:

$$\mathbf{S}_t \approx \mathbf{W}\mathbf{X}_t \tag{3.13}$$

The transformation $\mathbf{W}$ projects the mixed signals $\mathbf{X}$ onto the independent components that summarize the uniform distribution of $\mathbf{S}$, where the rows of $\mathbf{W}$ project the mixed data onto a single independent component.

Similar to PCA, ICA-based LP requires the selection of q dominant components from the d available independent components, i.e., q rows in demixing matrix $\mathbf{W}$. The remaining $d-q$ rows are attributed to common cause variation. For that purpose, the demixing matrix is partitioned as follows [84]:

$$\mathbf{W} = \begin{bmatrix} \hat{\mathbf{W}} \\ \tilde{\mathbf{W}} \end{bmatrix}, \quad \hat{\mathbf{W}} \in \mathbb{R}^{q\times m}, \quad \tilde{\mathbf{W}} \in \mathbb{R}^{(m-q)\times m} \tag{3.14}$$

The mixed variable vector $\mathbf{x}$ is decomposed into the reconstruction vector $\hat{\mathbf{x}}$ and residual vector $\tilde{\mathbf{x}}$:

$$\begin{aligned} \mathbf{x} &= \hat{\mathbf{x}} + \tilde{\mathbf{x}} \\ &= \hat{\mathbf{W}}^{-1}\hat{\mathbf{s}} + \tilde{\mathbf{W}}^{-1}\tilde{\mathbf{s}} \\ &= \hat{\mathbf{W}}^{-1}\hat{\mathbf{W}}\mathbf{x} + \tilde{\mathbf{W}}^{-1}\tilde{\mathbf{W}}\mathbf{x} \end{aligned} \tag{3.15}$$

The mapping to the latent vector $\mathbf{z}$ is:

$$\mathbf{z} = \hat{\mathbf{s}} = \hat{\mathbf{W}}\mathbf{x} \tag{3.16}$$

The mapping to the reconstruction vector $\hat{\mathbf{x}}$ is:

$$\hat{\mathbf{x}} = \hat{\mathbf{W}}^{-1}\mathbf{z} \tag{3.17}$$

ICA-based process monitoring is applicable when the source of significant variance in process variables is unknown and cannot be attributed to common cause variation. For example, wind speed has a significant effect on the variance a wind mill dynamics and control system; if unmeasured, it can be considered as hidden source factor. Another example is the presence of control changes that are not registered as computer process variables signals, such as the manual opening of a gas valve.

3.1.3 Kernel component analysis

The LP methods reviewed so far are only viable in a linear setting; PCA is viable if $\mathbf{x}$ follows the assumption of multivariate normality and ICA is viable if $\mathbf{x}$ is a linear combination of an independent uniformly distributed source vector $\mathbf{s}$. It is sometimes the case that variables in $\mathbf{x}$ observe nonlinear characteristics that result in nonlinear correlations, such as those shown in Figures 2.6(a) and 2.6(b). In such cases, variables exist on a nonlinear principal manifold that cannot be modeled with a linear LP method.

Nonlinear ICA is an extension of ICA addressed for nonlinear mixing models [54], [113]. If $\mathbf{x}$ is assumed to be a nonlinear mixture of the independent source vector $\mathbf{s}$:

$$\mathbf{x} = \mathbf{f}(\mathbf{s}) + \mathbf{e} \tag{3.18}$$

where $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^m$ is an unknown nonlinear function, then the task of nonlinear ICA is to estimate $\mathbf{s}$ by finding an inverse mapping $\mathbf{g} : \mathbb{R}^m \to \mathbb{R}^d$ such that:

$$\mathbf{s} \approx \mathbf{g}(\mathbf{x}) \tag{3.19}$$

Kernel principal component analysis (KPCA) augments PCA with a kernel method [119]. The basis of kernel methods is to map the predictors of a machine learning model to a higher dimensional space prior to fitting the model. Kernel methods are a class of algorithms that transform a linear model into a nonlinear model by mapping its predictors with a nonlinear kernel function [120]. Expanding the original variable space makes it possible for variable relations, that were impossible to model in the original variable space, to be modeled in the higher dimensional space. KPCA maps

the variable vector $\mathbf{x}$ to a higher dimensional space and then computes the principal components in that feature space that describe the nonlinear variations in the data.

The concept behind KPCA is too extensive for a short summary and is thus left out in this thesis. A detailed review of KPCA and its application in multivariate process monitoring is provided in [40], [82], [96], [100], [118], [119],. Multivariate process monitoring with KPCA is analogues to that of PCA and ICA; q kernel principal components that explain most of the predictable nominal variation in $\mathbf{x}$ are identified in the higher dimensional space to decompose the original variables $\mathbf{x}$ into the reconstruction vector $\hat{\mathbf{x}}$ and residual vector $\bar{\mathbf{x}}$.

The kernel method is also used as a means for obtaining the inverse (demixing) function from nonlinear ICA [81], [136], [150].

3.1.4 Autoencoders

An autoencoder (AE) is a type of an artificial neural network (ANN) configured for LP. The theoretical base for ANNs is inspired by studies of the brain and nervous system in biological organisms [69]. McCulloch and Pitts [94] introduced the concept behind ANNs by creating a computational model for neural networks. Neural activity was based on an "all-or-none" process that is comparable to high threshold logic. The objective was to convert continuous input information into a discrete decision output. Rosenblatt [110] developed the concept further by introducing the perceptron - an algorithm for learning a binary classifier that maps the real-valued input vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ to a single binary output value $f(\mathbf{x})$:

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{w}\mathbf{x} + b > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

where $\mathbf{w} \in \mathbb{R}^{1\times m}$ is a vector of real-valued weights and b is the bias that shifts the decision boundary, i.e., threshold, away from the origin. Figure 3.2(a) illustrates the perceptron. Multiple perceptrons arranged into an interconnecting structure such as in Figure 3.2(b) forms a complex logical network that computes a logical function. In fact, the perceptrons comprising a logical network can be configured to resemble logic gates contained in logic circuits [99].

The subject of ANNs has developed considerably since its conception. Their use extends beyond that of computing logical functions: ANNs are a stable in the field of machine learning and are used for statistical tasks such as regression, classification, clustering, feature extraction, probabilistic modeling, and prediction [42], [99], [112]. ANNs are regularly applied in disciplinary fields such as speech recognition, image analysis, and automated decision making [43], [74], [97].

Of interest in this thesis are feedforward neural networks (FNNs) [115]. FNNs are composed of neurons that are a functional alteration of the perceptron in Eq. (3.20), namely, they map the input vector $\mathbf{x}$ to a real-valued output $f(\mathbf{x})$. This change permits continuous information to propagate through the FNN. Figure 3.3 shows four interconnecting feedforward network layers that occupy a deep FNN. Each network layer i is composed of k_i neurons. For layer l, the real valued activation a_l^i of the i^{th} neuron is related to the activations in the $(l-1)^{th}$ layer by the equation:

$$a_l^i = \sigma_l^i \left(\sum_{j=1}^{k_{l-1}} w_l^{i,j} \cdot a_{l-1}^j + b_l^i \right) \tag{3.21}$$

where the weight $w_l^{i,j}$ is a real-valued modeling parameter for the activation a_{l-1}^j of the j^{th} neuron in the $(l-1)^{th}$ layer, b_l^i is the bias, and the activation function σ_l^i is a real-valued function. The expression in Eq. (3.21) can be rewritten in a vector form:

$$a_l^i = \sigma_l^i \left(\mathbf{w}_l^i \mathbf{a}_{l-1} + b_l^i \right) \tag{3.22}$$

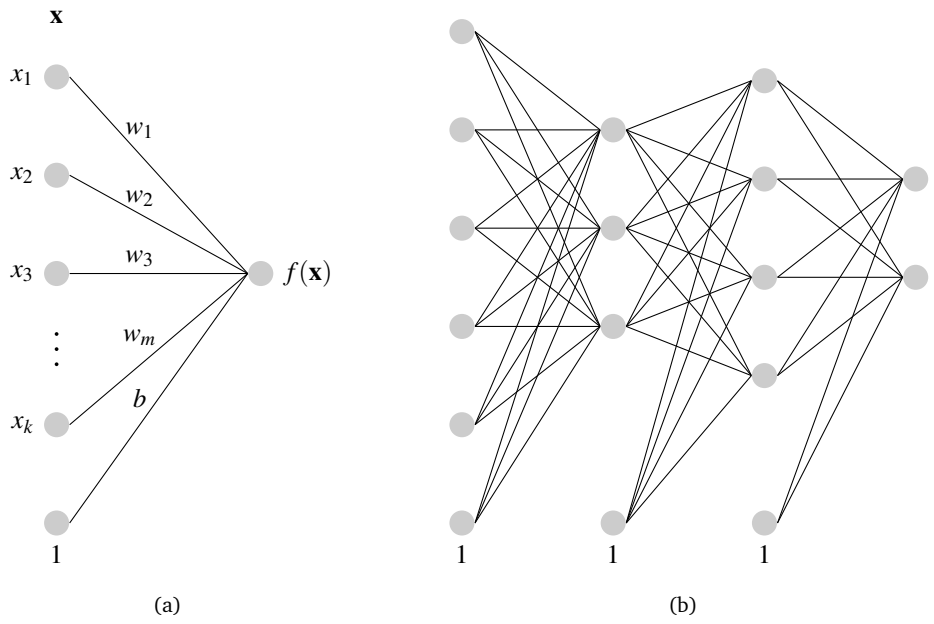


Figure 3.2: (a) The perceptron. (b) A deep ANN composed of interconnecting perceptrons.

where

$$\mathbf{w}_l^i = \begin{bmatrix} w_1^i & w_2^i & \dots & w_{k_{l-1}}^i \end{bmatrix}, \qquad \mathbf{a}_{l-1} = \begin{bmatrix} a_{l-1}^1 \\ a_{l-1}^2 \\ \dots \\ a_{l-1}^{k_{l-1}} \end{bmatrix} \tag{3.23}$$

The vector of activations $\mathbf{a}_l = \begin{bmatrix} a_l^1 & a_l^2 & \dots & a_l^{k_l} \end{bmatrix}^\intercal$ is computed by rewriting the expression in Eq. 3.22 in a matrix form:

$$\mathbf{a}_l = \sigma_l \left(\mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l \right) \tag{3.24}$$

where

$$\mathbf{W}_l = \begin{bmatrix} \mathbf{w}_l^1 \\ \mathbf{w}_l^2 \\ \vdots \\ \mathbf{w}_l^{k_l} \end{bmatrix}, \qquad \mathbf{b}_l = \begin{bmatrix} b_l^1 \\ b_l^2 \\ \dots \\ b_l^{k_l} \end{bmatrix} \tag{3.25}$$

and $\sigma_l = \begin{bmatrix} \sigma_l^1 & \sigma_l^2 & \dots & \sigma_l^{k_l} \end{bmatrix}^\intercal$ is a vector function where each element σ_l^i is the activation function for the i^{th} neuron in the l^{th} layer. Given n instances (observations) for $\mathbf{a}_{l-1}$, n instances for $\mathbf{a}_l$ are computed via an extension of Eq. 3.24:

$$\mathbf{A}_l = \sigma_l \left(\mathbf{W}_l \mathbf{A}_{l-1} + \mathbf{b}_l \right) \tag{3.26}$$

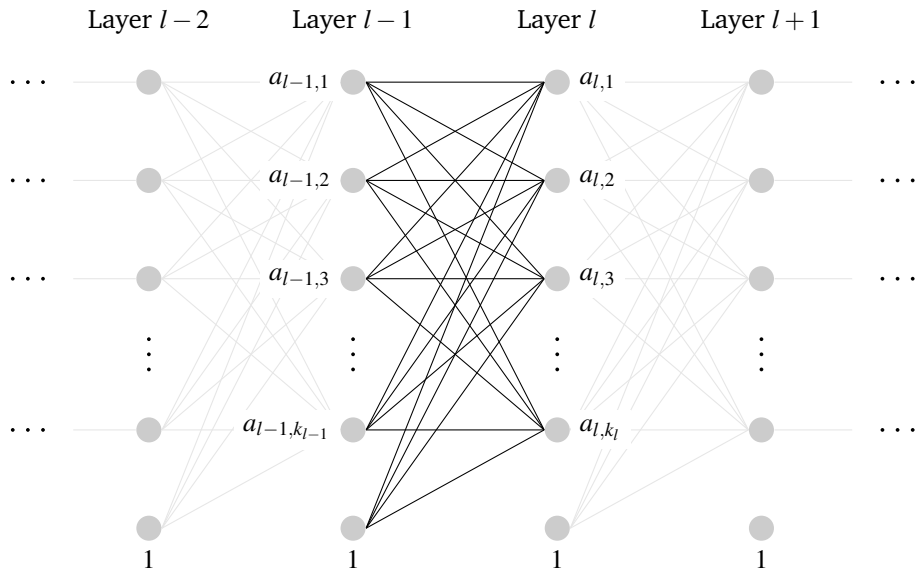


Figure 3.3: Illustration of four interconnecting network layers residing in a deep FNN.

where

$$\mathbf{A}_{l-1} = \begin{bmatrix} \mathbf{a}_{l-1,1} & \mathbf{a}_{l-1,2} & \dots & \mathbf{a}_{l-1,n} \end{bmatrix}, \qquad \mathbf{A}_{l} = \begin{bmatrix} \mathbf{a}_{l,1} & \mathbf{a}_{l,2} & \dots & \mathbf{a}_{l,n} \end{bmatrix} \tag{3.27}$$

Activation functions define the final output of a neuron given the weighted sum of its inputs. Several activation functions are defined in the literature, each with its own useful properties. Figure 3.3 shows six activation functions that are commonly used in the literature. A neuron employing the binary activation function in Figure 3.4(a) corresponds to the perception in Eq. (3.20). The logistic sigmoid activation function in Figure 3.4(b) is inspired by probability theory. It can be regarded as smoothed version of the binary activation function, such that the output ranges continuously between 0 and 1. The contrast between the binary and sigmoid activation functions is equivalent to the distiction fuzzy logic and Boolean logic. The tangent hyperbolic function is similar to the logistic sigmoid function, except the output ranges continuously between -1 and 1. It is typically used when processing mean-centered data. Neurons may employ the identity function to permit unbounded activity. The rectifier activation function may be regarded as a combination of the binary and identity functions: a network composed of rectifier neurons will exhibit a sparse activation where only some neurons are active, i.e., have non-zero output, with, unlike a binary neuron, a continuous activation. In fact, the rectifier is, as of 2015, the most popular nonlinear activation function [78]. Numerous other activation functions, such as the exponential linear unit, are provided in the literature [108]. Activation functions affect the modeling performance of FNNs; given a machine learning task, one activation function may be favorable over another given the nature of the data. In fact, a neural network employing nonlinear functions can be proven to satisfy the Universal Approximation Theorem, whereby it can model any nonlinear function [30]. This property is not met by the identify function.

Given the input vector $\mathbf{x} \in \mathbb{R}^{k_x \times 1}$, a FNN models the relationship between the label vector $\mathbf{y} \in \mathbb{R}^{k_y \times 1}$ and $\mathbf{x}$:

$$\begin{aligned} \mathbf{y} &= f_{\text{FNN}}(\mathbf{x}) + \boldsymbol{\varepsilon} \\ &= \hat{\mathbf{y}} + \boldsymbol{\varepsilon} \end{aligned} \tag{3.28}$$

where f_{FNN} is a function modelled by the FNN, $\hat{\mathbf{y}}$ is an estimate for $\mathbf{y}$, and $\boldsymbol{\varepsilon}$ is the model error. Figure 3.5 shows an illustration of a FNN with L hidden network layers that connect the input layer to the output layer. The equation for $\hat{\mathbf{y}}$ given $\mathbf{x}$ is:

$$\begin{aligned} \hat{\mathbf{y}} &= \sigma_y(\mathbf{W}_y \mathbf{a}_L + \mathbf{b}_y) \\ &= \sigma_y(\mathbf{W}_y \times \sigma_L(\mathbf{W}_L \mathbf{a}_{L-1} + \mathbf{b}_L) + \mathbf{b}_y) \\ &= \sigma_y(\mathbf{W}_y \times \sigma_L(\mathbf{W}_L \times \sigma_{L-1}(\mathbf{W}_{L-1} \mathbf{a}_{L-2} + \mathbf{b}_{L-1}) + \mathbf{b}_L) + \mathbf{b}_y) \\ &= \sigma_y(\mathbf{W}_y \times \dots \sigma_1(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \cdots + \mathbf{b}_y) \end{aligned} \tag{3.29}$$

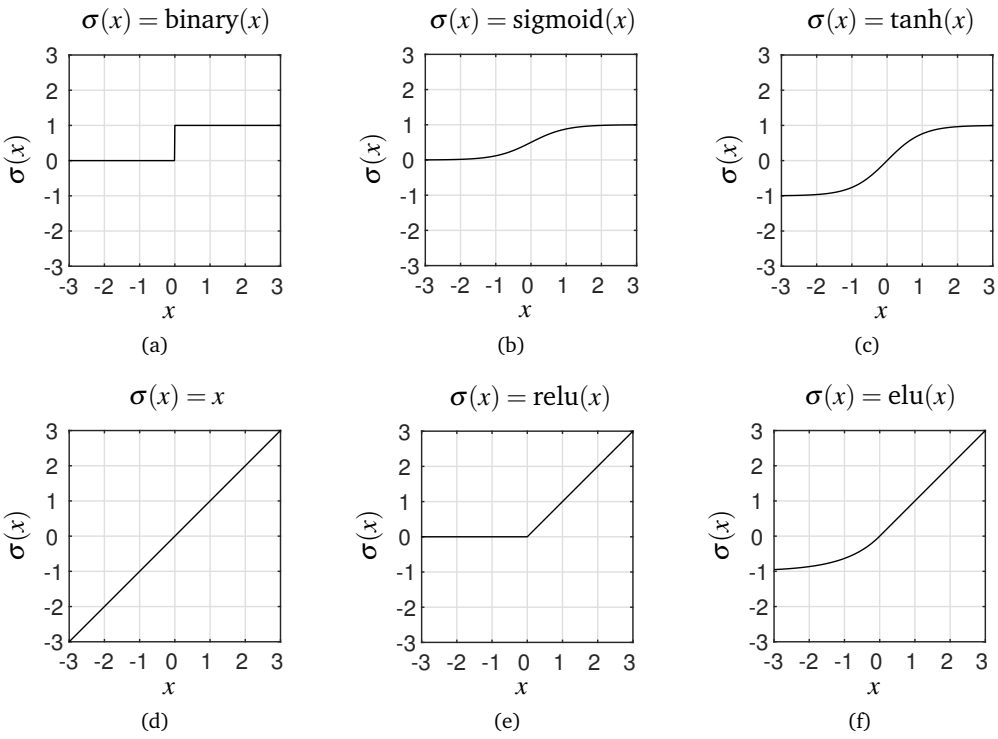


Figure 3.4: Neuron activation functions: (a) the binary function, (b) the logistic sigmoid function, (c) the tangent hyperbolic function, (d) the identity function, (e) the rectifier function, and (f) the exponential linear unit function.

If n instances (observations) for $\mathbf{x}$ and $\mathbf{y}$ are provided in the form of $\mathbf{X} \in \mathbb{R}^{k_x \times n}$ and $\mathbf{Y} \in \mathbb{R}^{k_y \times n}$, respectively, then n instances for $\hat{\mathbf{y}}$ in the form of $\hat{\mathbf{Y}} \in \mathbb{R}^{k_y \times n}$ are obtained via a combination of Eq. (3.26) and Eq. (3.29):

$$\begin{aligned} \hat{\mathbf{Y}} &= \sigma_y \left(\mathbf{W}_y \mathbf{A}_L + \mathbf{b}_y \right) \\ &= \sigma_y \left(\mathbf{W}_y \times \ldots \sigma_1 \left(\mathbf{W}_1 \mathbf{X} + \mathbf{b}_1 \right) \cdots + \mathbf{b}_y \right) \end{aligned} \tag{3.30}$$

Neural networks are trained, i.e., optimized, with a reference input set $\mathbf{X}_r$ and reference label set $\mathbf{Y}_r$. The machine learning task is to optimize the weight matrices $\mathbf{W}_i$ and biases $\mathbf{b}_i$ where $i \in \mathbb{Z} : j \in [1, L]$ with the intent of minimizing a loss function:

$$\text{Loss} = \mathscr{L}(\mathbf{Y}_r, \hat{\mathbf{Y}}_r) \tag{3.31}$$

where $\hat{\mathbf{Y}}_r$ is computed from $\mathbf{X}_r$ via Eq. (3.30). There are many forms for the error function $\mathscr{L}$, but the most common the quadratic cost function:

$$\mathscr{L}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{n} \left|\left| \mathbf{Y} - \hat{\mathbf{Y}} \right|\right|^2 \tag{3.32}$$

Weights are optimized with backpropagation via stochastic gradient descent [99].

An AE is a FNN configured for LP. An AE consists of two parts, an encoder and a decoder, where the former transforms the high-dimensional variable vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ into the lower-dimensional feature vector $\mathbf{z} \in \mathbb{R}^{q\times 1}$ and the latter reconstructs the original variable vector $\hat{\mathbf{x}} \in \mathbb{R}^{m\times 1}$ with a transformation of the features. With respect to Eq. (3.29), this is equivalent to the substitution $\hat{\mathbf{y}} \triangleq \hat{\mathbf{x}}$. The encoder maps the input $\mathbf{x} \in \mathbb{R}^{m\times n}$ to the latent variables $\mathbf{z} \in \mathbb{R}^{q\times n}$:

$$\mathbf{e}_i = \begin{cases} \sigma_1^e\left(\mathbf{W}_1^e\mathbf{x} + \mathbf{b}_1^e\right), & \text{for } i = 1 \\ \sigma_i^e\left(\mathbf{W}_i^e\mathbf{e}_{i-1} + \mathbf{b}_i^e\right), & \text{else} \end{cases} \tag{3.33}$$

$$\mathbf{z} = \sigma^z\left(\mathbf{W}^z\mathbf{e}_N + \mathbf{b}^z\right)$$

where $i \in \mathbb{Z} : i \in [1, N]$ and $q < m$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer. The decoder maps the latent

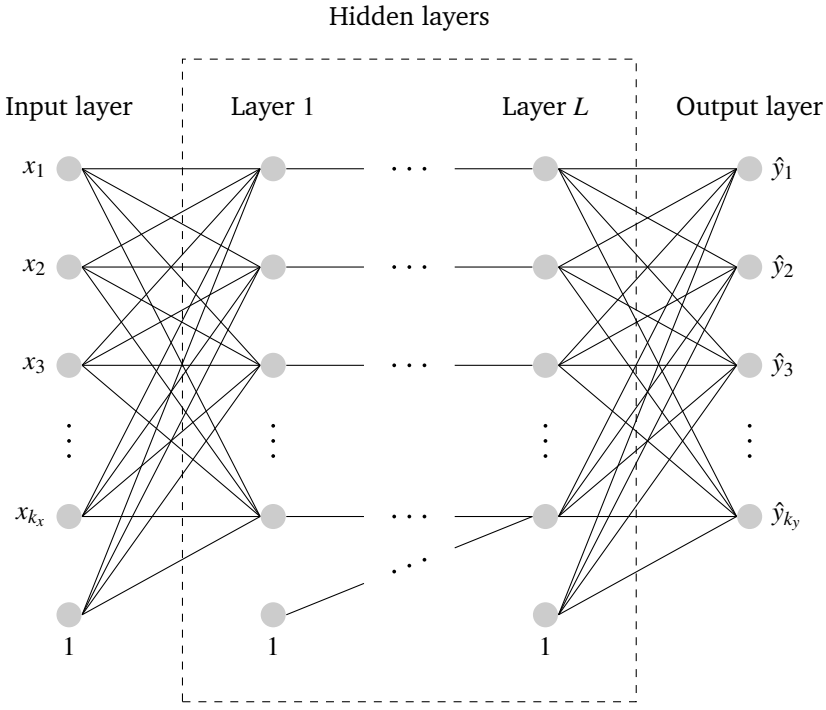


Figure 3.5: Illustration of a FNN with L hidden layers.

variables $\mathbf{z} \in \mathbb{R}^{q \times n}$ to the input reconstruction $\hat{\mathbf{x}} \in \mathbb{R}^{m \times n}$:

$$\mathbf{d}_i = \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{z} + \mathbf{b}_1^d \right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{d}_{j-1} + \mathbf{b}_j^d \right), & \text{else} \end{cases} \tag{3.34}$$

$$\hat{\mathbf{x}} = \sigma^{\hat{\mathbf{x}}} \left(\mathbf{W}^{\hat{\mathbf{x}}} \mathbf{d}_M + \mathbf{b}^{\hat{\mathbf{x}}} \right)$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{\mathbf{x}}}$, $\mathbf{b}^{\hat{\mathbf{x}}}$, and $\sigma^{\hat{\mathbf{x}}}$ are defined similarly for the output layer. Given the reference set $\mathbf{X}_t \in \mathbb{R}^{m \times n}$, the model parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{\mathbf{x}}}$, and $\mathbf{b}^{\hat{\mathbf{x}}}$ are optimized with respect to minimizing the following reconstruction loss function:

$$\mathscr{L}(\mathbf{X}_t, \hat{\mathbf{X}}_\mathbf{t}) = \frac{1}{n} \left|\left| \mathbf{X}_t - \hat{\mathbf{X}}_t \right|\right|^2 \tag{3.35}$$

Figure 3.6 shows an AE configured to extract a two-dimensional feature vector $\mathbf{z}$ from a six dimensional variable vector $\mathbf{x}$. The AE gradually condenses the input to the feature space before gradually reconstructing it; such a configuration, i.e., one where the dimension of every encoder/decoder layer is less than the original variable space, is typically presented in the literature. The figure shows that dimensions of the encoder and decoder layers are mirrored, but this it not a necessary requirement.

3.1.4.1 Relation to PCA, ICA, and KPCA

Under certain modelling constraints, AEs can uncover the mappings $\underline{E}(\mathbf{x})$ and $\underline{D}(\mathbf{x})$ given by other LP methods. For instance, an AE consisting exclusively of neurons using the identity activation function will learn the transformations of Eqs. (3.8) and (3.9) that are given by PCA [7], [106]. The transformations are learned even if the encoder and decoder part comprise several layers of linear units. The mappings given by linear ICA are learned by (a) having an AE consist exclusively of linear neurons; and (b) augmenting the cost function in Eq. (3.35) with the constraint that the latent activations in $\mathbf{z}$ are as independent as possible [68], [67]. Nonlinear ICA is facilitated by employing nonlinear activation functions. Wasserman [137] presented the radial basis function neural network, where kernel functions are learned with hidden neurons computing the radial basis functions of the inputs. One of the challenges with kernel methods is the need to select a proper kernel function and optimization its parameters prior to solving the method. The works of Le et al. [76], [77] present a FNN that learns an optimal kernel function from the data. Their experiments demonstrate superior feature extraction performance

for a kernel-based AE than for a model given by KPCA. It is important to note that despite the flexibility of an AE, the network must solve a nonlinear optimization problem with the possibility of getting trapped in local minima, whereas PCA and KPCA requires only the solution of an eigenvalue problem. For AE components need to be specified in advance.

3.1.5 Determining the latent dimension q

The dimension q of the feature vector $\mathbf{z} \in \mathbb{R}^q$ is an integral part of a LP model. Section 2.1 details that q specifies the dimension of the principal manifold $\mathbf{f}(\boldsymbol{\lambda})$ that jointly summarizes $\mathbf{x} \in \mathbb{R}^m$. Given a low dimension m, the dimension q may be determined by, for example, a visual inspection of $\mathbf{x}$ as in Figure 2.3, where it is shown that a principal curve summarizes the data in Figure 2.3(a) and a principal

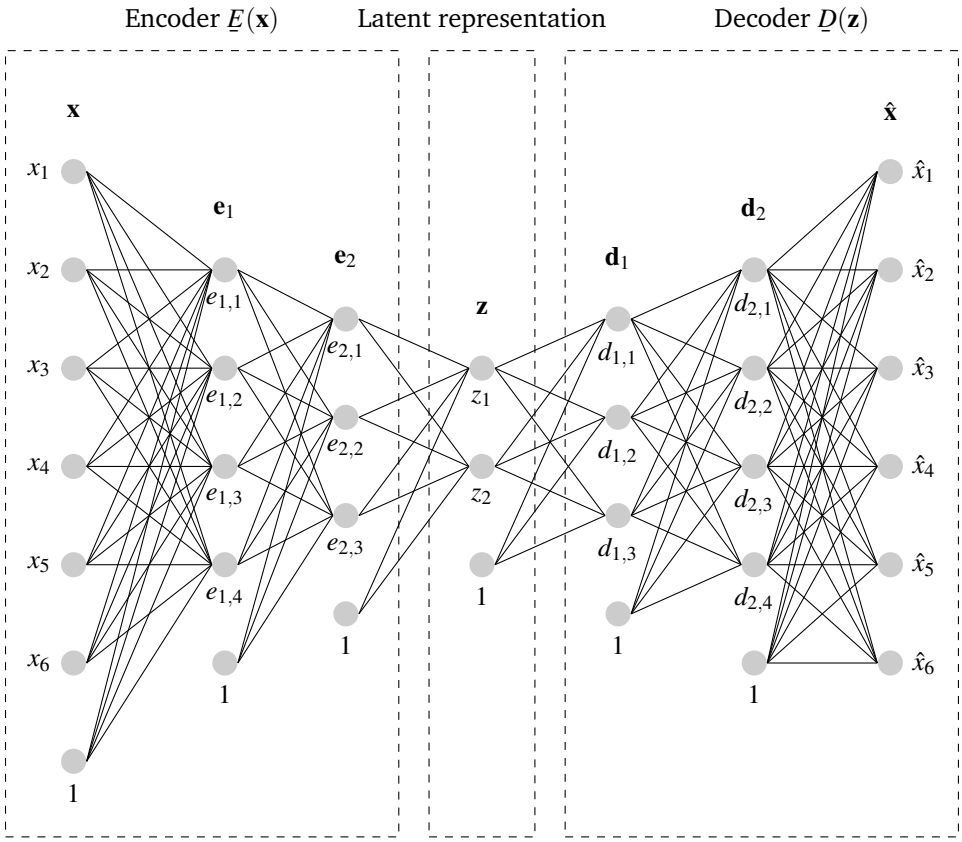


Figure 3.6: Illustration of a 6-4-3-1-3-4-6 AE.

surface summarizes the data in Figure 2.3(b). However, determining q via a visual inspection of $\mathbf{x}$ is difficult when m is large, as is often the case for large process systems where there may be as many as 1500 observable process variables [9].

In the context of process monitoring, the principal manifold $\mathbf{f}(\boldsymbol{\lambda})$ is to provide a joint summary of the nominal process variance exhibited by the process variable vector $\mathbf{x}$; hence the dimension q of the LP model's latent vector $\mathbf{z}$ should coincide with nominal process variance while the remaining $m-q$ dimensions should be attributed to unexplained process and measurement stochasticity. A low q results with the latent vector $\mathbf{z}$ not sufficiently retaining nominal process variance. Consequently, the LP model is not be representative of nominal process behavior and provides an inaccurate reconstruction $\hat{\mathbf{x}}$ by mistaking nominal variance as common cause variance, thereby likely to generate incorrect results for abnormal event diagnosis. A large q results with a LP model that models unexplained stochasticity by retaining common cause variation in its latent vector $\mathbf{z}$. Consider the following static process:

$$\begin{aligned} v &= N(0,1) \\ h &= v + \mathcal{N}(0,0.1) + f_1 \\ y_1 &= h + \mathcal{N}(0,0.1) + f_2 \\ y_2 &= h + \mathcal{N}(0,0.1) + f_3 \end{aligned} \tag{3.36}$$

where v is an input signal, h is a process variable, y_1 and y_2 are output signals, f_1 is a process fault, and f_2 and f_3 are measurement faults. For the purpose of feature extraction, the following variable vector is defined:

$$\mathbf{x} = \begin{bmatrix} y_1 \\ y_2 \\ v \end{bmatrix} \tag{3.37}$$

Vector $\mathbf{x}$ is sampled to produce the training set $\mathbf{X}_t$, validation set $\mathbf{X}_v$, and fault set $\mathbf{X}_f$. Each set consists of 300 samples. For the fault set $\mathbf{X}_f$, the system operated under nominal conditions for samples 1-75, $(f_1, f_2, f_3) = (1,0,0)$ for samples 76-150, $(f_1, f_2, f_3) = (0,1,0)$ for samples 151-225, and $(f_1, f_2, f_3) = (0,0,1)$ for samples 226-300.

An LP model is given by PCA since (a) the variables in $\mathbf{x}$ represent a linear system, (b) the variables follow the assumption of normality, and (c) the source variable, i.e., input v, is known, rendering ICA unnecessary. The eigenvectors $\mathbf{P}$ and eigenvalues $\boldsymbol{\Lambda}$ are obtained from an eigendecomposition of the estimated covariance matrix $\hat{\boldsymbol{\Sigma}}(\mathbf{X}_t)$:

$$\mathbf{P} = \begin{bmatrix} 0.5872 & 0.7473 & 0.3111 \\ 0.5886 & -0.6580 & 0.4697 \\ 0.5557 & -0.0927 & -0.8262 \end{bmatrix}, \qquad \boldsymbol{\Lambda} = \begin{bmatrix} 3.3000 & 0 & 0 \\ 0 & 0.0446 & 0 \\ 0 & 0 & 0.0347 \end{bmatrix} \tag{3.38}$$

It is evident from Eq. (3.36) that variables y_1, y_2, and v are strongly correlated with one another. Therefore, the variable vector $\mathbf{x}$ may be adequately summarized with a principal curve, i.e., a principal manifold with $q = 1$. Since the intent of this example is to demonstrate the issue with setting q as too large, two latent vectors $\mathbf{z}_1 = \hat{\mathbf{P}}_1^\intercal \mathbf{x}$ (see Eq. (3.8)) and $\mathbf{z}_2 = \hat{\mathbf{P}}_2^\intercal \mathbf{x}$ are defined:

$$\hat{\mathbf{P}}_1 = \begin{bmatrix} 0.5872 \\ 0.5886 \\ 0.5557 \end{bmatrix}, \qquad \hat{\mathbf{P}}_2 = \begin{bmatrix} 0.5872 & 0.7473 \\ 0.5886 & -0.6580 \\ 0.5557 & -0.0927 \end{bmatrix}, \tag{3.39}$$

where $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$ are obtained via a decomposition of the principal loading matrix $\mathbf{P}$ (Eq. (3.7)):

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \hat{\mathbf{P}}_1 & \tilde{\mathbf{P}}_1 \end{bmatrix}, \quad \tilde{\mathbf{P}}_1 = \begin{bmatrix} 0.7473 & 0.3111 \\ -0.6580 & 0.4697 \\ -0.0927 & -0.8262 \end{bmatrix}, \\ \mathbf{P} &= \begin{bmatrix} \hat{\mathbf{P}}_2 & \tilde{\mathbf{P}}_2 \end{bmatrix}, \quad \tilde{\mathbf{P}}_2 = \begin{bmatrix} 0.3111 \\ 0.4697 \\ -0.8262 \end{bmatrix} \end{aligned} \tag{3.40}$$

From Eq. (3.39), $\mathbf{z}_1$ is a one dimensional vector and $\mathbf{z}_2$ is a two dimensional vector. Consequently, the reconstruction vector $\hat{\mathbf{x}}_1 = \mathbf{P}\mathbf{z}_1$ (see Eq. (3.9)) lies on a principal curve, whereas $\hat{\mathbf{x}}_2 = \mathbf{P}\mathbf{z}_2$ lies on a principal surface. By combining Eqs. (3.8) and (3.9), the reconstruction sets $\hat{\mathbf{X}}_{t,1} = \hat{\mathbf{P}}_1\hat{\mathbf{P}}_1^\intercal\mathbf{X}_t$ and $\hat{\mathbf{X}}_{t,2} = \hat{\mathbf{P}}_2\hat{\mathbf{P}}_2^\intercal\mathbf{X}_t$ are generated. Figures 3.7(a) and 3.7(b) display a three dimensional scatter plot of the data contained in $\hat{\mathbf{X}}_{t,1}$ and $\hat{\mathbf{X}}_{t,2}$, respectively. It is evident that the reconstructions $\hat{\mathbf{X}}_{t,1}$ lie on a principal curve and that the projections $\hat{\mathbf{X}}_{t,2}$ lie on a principal surface.

A plot of the training set $\mathbf{X}_t$ is included in Figure 3.7(a). The figure demonstrates that the principal component $\hat{\mathbf{P}}_1$ is associated with nominal process variation in $\mathbf{x}$ since the principal curve provides an adequate joint summary of the training set $\mathbf{X}_t$; hence the remaining principal components $\tilde{\mathbf{P}}_1$ are associated with common cause variation in $\mathbf{x}$, represented by the spread of $\mathbf{X}_t$ around the principal curve. Note that the first column of $\hat{\mathbf{P}}_2$ is equivalent to $\hat{\mathbf{P}}_1$ and that the second column of $\hat{\mathbf{P}}_2$ is equivalent to the first column of $\tilde{\mathbf{P}}_1$. Therefore, the principal surface described by $\mathbf{z}_2$ is associated with both nominal process variation and common cause variation in $\mathbf{x}$ retained along the primary (long) and secondary (short) axis, respectively, of the ellipsoid in Figure 3.7(b). Table 3.1 presents the standard deviation of the SPE (Eq. (2.13)) of reconstructing the training set $\mathbf{X}_t$ and validation set $\mathbf{X}_v$ with both LP models. The table indicates that $\hat{\mathbf{x}}_2$ retains more common cause variation than $\hat{\mathbf{x}}_1$ since the standard deviation of the SPE of $\mathbf{x}_2 - \hat{\mathbf{x}}_2$ is less than $\mathbf{x}_1 - \hat{\mathbf{x}}_1$ for $\mathbf{X}_t$ and $\mathbf{X}_v$.

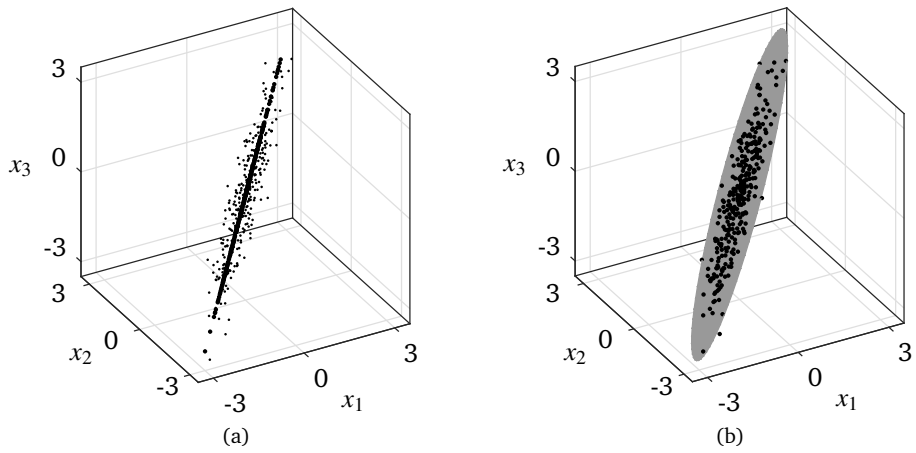


Figure 3.7: (a) Scatter plot of original training samples $\mathbf{X}_t$ and reconstructions $\hat{\mathbf{X}}_{t,1}$, (b) scatter plot of reconstructions $\hat{\mathbf{X}}_{t,2}$ that lie on a principal surface illustrated by the ellipsoid.

Retaining common cause variation in a LP model creates erroneous results for detecting abnormal events. Figures 3.8(a) and 3.8(b) show the SPE of reconstructing each sample of $\mathbf{X}_f$ with $\hat{\mathbf{P}}_1$ and $\hat{\mathbf{P}}_2$, respectively. Figure 3.8(a) shows that the SPE is equally sensitive to each fault f_i. Meanwhile, Figure 3.8(b) shows that the sensitivity of the SPE varies depending on the fault f_i, with f_1 begin the least sensitive and f_3 begin the most sensitive. Choosing a PCA-based LP model where $q = 2$ instead of $q = 1$ would thus produce worse performance for fault detectability.

The issue with a large q is made clearer if one considers the fact that the principal loading matrix $\mathbf{P}$ is obtained from an eigendecomposition of the estimated covariance matrix $\hat{\mathbf{\Sigma}}(\mathbf{X}_t)$; hence $\mathbf{P}$ is solely dependent on $\mathbf{X}_t$. Consider the case that a different training set $\underline{\mathbf{X}}_t$ is sampled. The eigenvectors $\underline{\mathbf{P}}$ and eigenvalues $\underline{\mathbf{\Lambda}}$ obtained from an eigendecomposition of the estimated covariance matrix $\hat{\mathbf{\Sigma}}(\underline{\mathbf{X}}_t)$ are now:

$$\underline{\mathbf{P}} = \begin{bmatrix} 0.5841 & 0.6378 & 0.5020 \\ 0.5922 & 0.0881 & -0.8009 \\ 0.5551 & -0.7651 & 0.3263 \end{bmatrix}, \qquad \underline{\mathbf{\Lambda}} = \begin{bmatrix} 3.1545 & 0 & 0 \\ 0 & 0.0422 & 0 \\ 0 & 0 & 0.0360 \end{bmatrix} \tag{3.41}$$

	$\text{std}\left\{\frac{1}{n}\left\|\left\|\mathbf{X}_i - \hat{\mathbf{P}}_1\hat{\mathbf{P}}_1^\intercal\mathbf{X}_i\right\|\right\|^2\right\}$	$\text{std}\left\{\frac{1}{n}\left\|\left\|\mathbf{X}_i - \hat{\mathbf{P}}_2\hat{\mathbf{P}}_2^\intercal\mathbf{X}_i\right\|\right\|^2\right\}$
$\mathbf{X}_t$	0.0254	0.0173
$\mathbf{X}_v$	0.0249	0.0161

Table 3.1: Standard deviation of SPE of reconstructions.

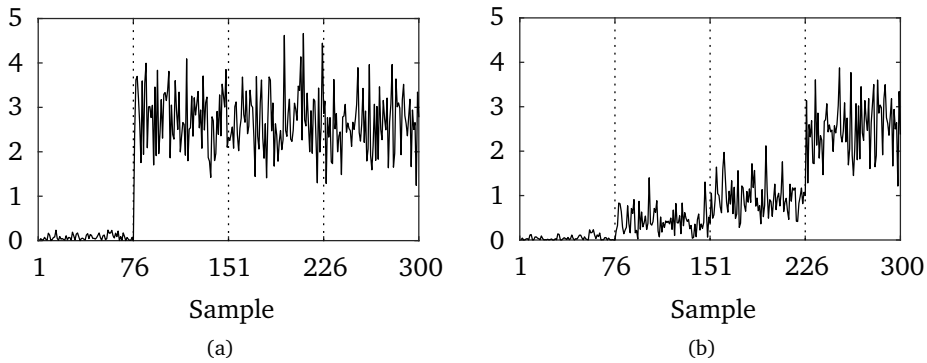


Figure 3.8: SPE of reconstructing each sample of $\mathbf{X}_f$ with (a) $\hat{\mathbf{P}}_1$ and (b) $\hat{\mathbf{P}}_2$.

As before, the latent vectors $\underline{\mathbf{z}}_1 = \hat{\underline{\mathbf{P}}}_1^\intercal \mathbf{x}$ and $\underline{\mathbf{z}}_2 = \hat{\underline{\mathbf{P}}}_2^\intercal \mathbf{x}$ are defined where:

$$\hat{\underline{\mathbf{P}}}_1 = \begin{bmatrix} 0.5841 \\ 0.5922 \\ 0.5551 \end{bmatrix}, \qquad \hat{\underline{\mathbf{P}}}_2 = \begin{bmatrix} 0.5841 & 0.6378 \\ 0.5922 & -0.0881 \\ 0.5551 & -0.7651 \end{bmatrix}, \tag{3.42}$$

are obtained from a decomposition of $\underline{\mathbf{P}}$. Note that $\hat{\mathbf{P}}_1$ and $\hat{\underline{\mathbf{P}}}_1$ are similar while the second columns of $\hat{\mathbf{P}}_2$ and $\hat{\underline{\mathbf{P}}}_2$ are not. Figures 3.9(a) and 3.9(b) show the SPE of reconstructing each sample of $\mathbf{X}_f$ with $\hat{\underline{\mathbf{P}}}_1$ and $\hat{\underline{\mathbf{P}}}_2$, respectively. Note that the fault matrix is $\mathbf{X}_f$ remains the same as the one used to generate the SPEs in Figures 3.8(a) and Figures 3.8(b). Similar to Figure 3.8(a), Figure 3.9(a) shows that the SPE is equally sensitive to each fault f_i. Similar to Figure 3.8(b), Figure 3.9(b) shows that the sensitivity of the SPE varies depending on the fault f_i, but with f_3 being the least sensitive and f_2 begin the most sensitive. This change in fault sensitivity stems from

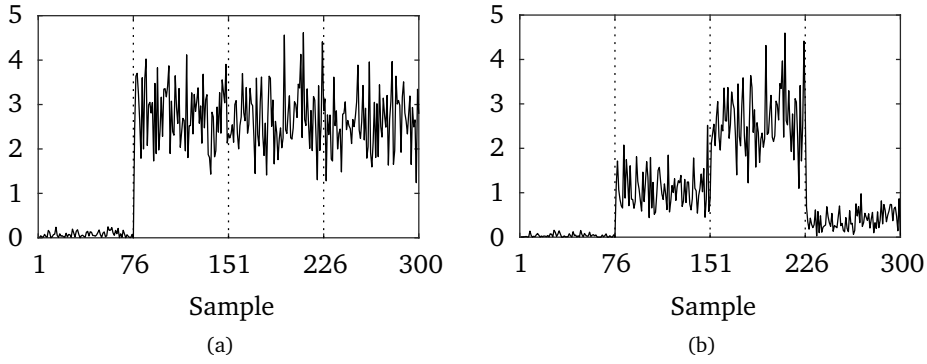


Figure 3.9: SPE of reconstructing each sample of $\mathbf{X}_f$ with (a) $\hat{\mathbf{P}}_1$ and (b) $\hat{\mathbf{P}}_2$.

the difference in the second column of $\hat{\mathbf{P}}_2$ and $\underline{\hat{\mathbf{P}}}_2$. Therefore, a large q makes a LP model sensitive to common cause variation in training data; hence, two LP models with consistent q trained yet on different data will generate different results for fault detection, even if data is sampled from the same process.

Although the example above was demonstrated with a LP model given by PCA, it is reasonable to assume that complications in abnormal event detection due to a large dimension q of the latent vector $\mathbf{z}$. will be present for a LP model given by ICAand KPCA, as well as a LP model parameterized by an AE.

Several methods exist for estimating q for a LP model given by PCA [125]. Within the process monitoring literature, q is typically selected by calculating the cumulative percent variance (CPV) [1], [2], [10], [37], [39]. The CPV is a measure of the percent variance captured by the first q principal components:

$$\text{CPV}(q) = \frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \times 100\% \tag{3.43}$$

The remaining percent variance is explained by the remaining $m-q$ principal components that are attributed to common cause variance. With this criterion, one determines q by selecting a desired CPV, e.g., 80%, 85%, 90%, etc., which is subjective. For instance, one should select a large CPV if the process observes minimal stochasticity because the variance of the process variables is attributed to nominal process variance. On the other hand, a process exhibiting significant stochasticity requires a low CPV, since setting q too high would model the stochasticity and introduce difficulties in detecting abnormal events.

An LP model given by ICA requires the selection of q dominant components from the d available independent components. In PCA, the order of the row vectors $\mathbf{t}_i$ of the score matrix $\mathbf{T}$ is determined by their corresponding variance λ_i. However, the ordering of row vectors $\mathbf{w}_i$ of the demixing matrix $\mathbf{W}$ poses no statistical significance. The problem becomes more complex if a nonlinear extension of ICA is used. A number of methods have been suggested to determine a component order [6], [23], [24], [52]. For example, in the linear setting, Cardoso and Souloumiac [17] sorted the rows of $\mathbf{W}$ according to the L_2 norm of each individual $\mathbf{w}_i$. LP is performed by selecting the q dominant rows of the sorted demixing matrix $\mathbf{W}$ based on the assumption that the rows with the largest L_2 norm have the greatest effect on the variation of the source variable vector $\mathbf{s}$.

Methods for estimating q for a LP model given by KPCA tend to be a kernel extension of those applied for PCA-based LP models, such as the CPV approach and parallel analysis [35], [36], [65], [124]. With these methods, the objective is to not

only estimate q but to also determine an optimal value for kernel hyperparameters, such as the smoothing factor of the Gaussian kernel.

One advantage with a PCA/ICA/KPCA-based approach to establishing a LP model is that the dimension q is chosen as a subset of the components obtained from solving the eigendecomposition/demixing problem; hence, retaining additional components does not require the optimization problem to be re-solved. AEs, on the other hand, require the dimension q to be specified prior to optimizing the loss function in Eq. (3.35). Therefore, changing the dimension q requires the optimization problem to be re-solved. One approach is to generalize the Akaike's information criterion to be applicable to neural networks, thereby estimating the amount of information lost by an AE when assessing its quality [98]. However, this approach still requires a neural network to be continuously redefined. Incremental learning offers an alternative approach [5], [111], [147], [152],. The principal idea is that neural network structure progressively evolves over the optimization period in two ways: neurons are added to aid in minimizing the objective function, and redundant neurons are merged to obtain a compact representation that prevents overfitting.

3.1.6 Dynamic systems

Process variables of a dynamic process system generally exhibit correlations that are time dependent. Consider the following dynamic process:

$$y[t] = u[t-1] + e[t] \tag{3.44}$$

where $y[t]$ is the process output, $u[t]$ is the process input, and $e[t]$ is an unknown noise parameter. It is evident from Eq. (3.44) that $y[t]$ and $u[t-\tau]$ are cross-correlated for the lag parameter $\tau = 1$. If, for the purpose of LP, the variable vector were:

$$\mathbf{x} = \begin{bmatrix} y[t] & u[t] \end{bmatrix}^\mathsf{T} \tag{3.45}$$

then a LP model built for $\mathbf{x}$ in Eq. (3.45) would not model the cross-correlation between y and u. Rather, it would provide a static approximation. An appropriate variable vector would be:

$$\mathbf{x} = \begin{bmatrix} y[t] & u[t-1] \end{bmatrix}^\mathsf{T} \tag{3.46}$$

A LP model built for $\mathbf{x}$ in Eq. (3.46) would model the cross-correlation between y and u. It is generally difficult to determine which process variable x_i needs to be displaced by a lag of l when defining the variable vector $\mathbf{x}$. A generalized version for the variable vector is:

$$\mathbf{x} = \begin{bmatrix} y[t] & y[t-1] & u[t] & u[t-1] \end{bmatrix}^\mathsf{T} \tag{3.47}$$

An appropriate LP model built for $\mathbf{x}$ in Eq. (3.47) would model the correlation between $y[t]$ and $y[t-1]$, as well as retain $y[t]$ and $u[t]$. In the general case of establishing a LP model for a dynamic process system comprising m process variables, Ku et. al [75] propose a modified version that defines a dynamic process variable vector. The following historical vector for process variable $i \in \mathbb{Z} : i \in [1,m]$ is defined:

$$\mathbf{x}_i^l[k] = \begin{bmatrix} x_i[k] & x_i[k-1] & \dots & x_i[k-l] \end{bmatrix}^\mathsf{T} \tag{3.48}$$

The dynamic process variable vector is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^l[k] & \mathbf{x}_2^l[k] & \dots & \mathbf{x}_m^l[k] \end{bmatrix}^\mathsf{T} \tag{3.49}$$

Extending the variable vector $\mathbf{x}$ to include past l samples provides a *dynamic* formulation of a PCA/ICA/KPCA/AE-based approach to LP intended for motoring of dynamic process systems [26], [75], [83], [149].

3.2 Detection of abnormal events

Section 2.3 details that multivariate statistical process monitoring consists of referring a new observation $\mathbf{x}_{new}$ against a LP model that is established with nominal historical process data. The observation $\mathbf{x}_{new}$ is propagated through the model to generate the features $\mathbf{z}_{new}$ and reconstructions $\hat{\mathbf{x}}_{new}$, whereafter the Hotelling T^2 and SPE statistics are calculated with Eqs. (2.15) and (2.16), respectively. A process is considered nominal if the inequalities $SPE \leq CL_{SPE}$ and $T^2 \leq CL_{T^2}$ in Eq. (2.14) are satisfied. Assuming that the SPE follows a chi-squared distribution, the control limit CL_{SPE} is computed with the following approximate value [12]:

$$CL_{SPE} = \frac{\bar{\sigma}^2}{2\bar{\mu}} \chi^2_{(2\bar{\mu}^2/\bar{\sigma}^2,\alpha)} \tag{3.50}$$

where $\bar{\mu}$ and $\bar{\sigma}$ are, respectively, the sample mean and sample standard deviation of the SPE of a validation set $\mathbf{X}_v$ and α is the false alarm rate. For LP models given by PCA, KPCA, or an AE, and assuming that the latent vector $\mathbf{z}$ follows a multivariate normal distribution, the control limit CL_{AE} is computed with the following approximate value [123]:

$$CL_{T^2} = \frac{q(n+1)(n-1)}{n(n-q)} F_{(\alpha,q,n-q)} \tag{3.51}$$

where α is the false alarm rate. Applications of Eqs. (3.50) and (3.51) for are found in [26], [38], [85], [58], [63], [82], [91], [101]. Since SPE is generally favored over T^2 (see section 2.3), SPE is used as the sole monitoring statistic in this thesis.

The I^2 monitoring statistic is a measure for the variation of a new observation in the latent space of a LP model given by ICA [84]:

$$I^2 = \sum_{i=1}^{q} z_{new,i}^2 \tag{3.52}$$

However, if the independent components have unit variance, as is usually the case, then the I^2 monitoring statistic is equivalent to the Hotelling T^2 statistic in Eq. (2.15). Since the latent variables in ICA follow a uniform distribution, a kernel density estimation is used in calculating the control limit of I^2/T^2 and SPE [93], [21], [146], [66], [87].

3.3 Evaluation of abnormal events

Monitoring of the SPE and T^2 statistic via Eq. (2.14) only provides information on whether or not a process is nominal; by themselves, the statistics do not provide information on which process variables contain an abnormal alteration in their signal characteristics that are the cause of the detected abnormal event. Miller et al. [95] propose analyzing the individual contribution of a process variable x_i to the SPE statistic for assessing abnormal trends in process variables. The contribution C_i^{SPE} of process variable i to the SPE is:

$$C_i^{\text{SPE}} = (x_{new,i} - \hat{x}_{new,i})^2 \tag{3.53}$$

Variables showing large contributions are concluded to no longer be consistent with nominal operating conditions. Note that there is no definition of what constitutes a large contribution. This is left up to the judgment of the analyst. One diagnostic approach is to rank the abnormality of a variable in accordance to its contribution to the SPE, such that the variable with the largest contribution is considered the most abnormal, the variable with the second largest contribution is considered the second most abnormal, and so on for remaining variables [90].

It is noted that contribution analysis does not unambiguously reveal the cause of an abnormal event. Rather, it will expose the group of process variables that are no longer consistent with nominal operating conditions [90]. Operators can then focus their attention on fewer variables, but must still apply their process knowledge to infer a potential cause. Contribution analysis becomes a valuable diagnostic tool as the process becomes more complex and the number of process variable grows.

LP models suffer a fault-smearing effect - an effect where an abnormal drift in an abnormal variable generates contributions from variables consistent with normal operating conditions. The fault-smearing effect hampers contribution analysis since nominal variables are highlighted and abnormal variables are obscured [128].

Furthermore, there is no guarantee that abnormal variables have the largest contributions [4], [60]. In extreme cases, the fault-smearing effect leads to incorrect diagnosis. The fault-smearing effect occurs because the compression of variables to a smaller latent space and subsequent expansion to the original variable space enables nominal and abnormal variables to interact. Westerhuis et al. [138] offer a second interpretation: the LP model, having been trained on nominal process data, is not valid for abnormal process data and will produce model residuals, i.e., contributions, that cannot be trusted. The fault-smearing effect is illustrated with an AE in Figure 3.10. A fault is induced in a process that causes an abnormal drift in the variable x_1. This abnormality propagates through the AE and manifests itself onto the reconstructions, thereby generating a contribution for nominal process variables. The works of Van den Kerkhof et al. [127], [128] show that the fault-smearing effect is an unavoidable complication experienced by LP models. Yoon and MacGrecor [148] propose a method for isolating abnormal variables that consists of comparing the contributions for a newly detected abnormal event with the contributions for a previously diagnosed abnormal event. However, the method is only applicable for abnormal events that have occurred before; hence, the fault-smearing effect remains an issue when evaluating previously unseen abnormal events.

Figure 3.11 visualizes the fault-smearing effect with a scatter plot of samples

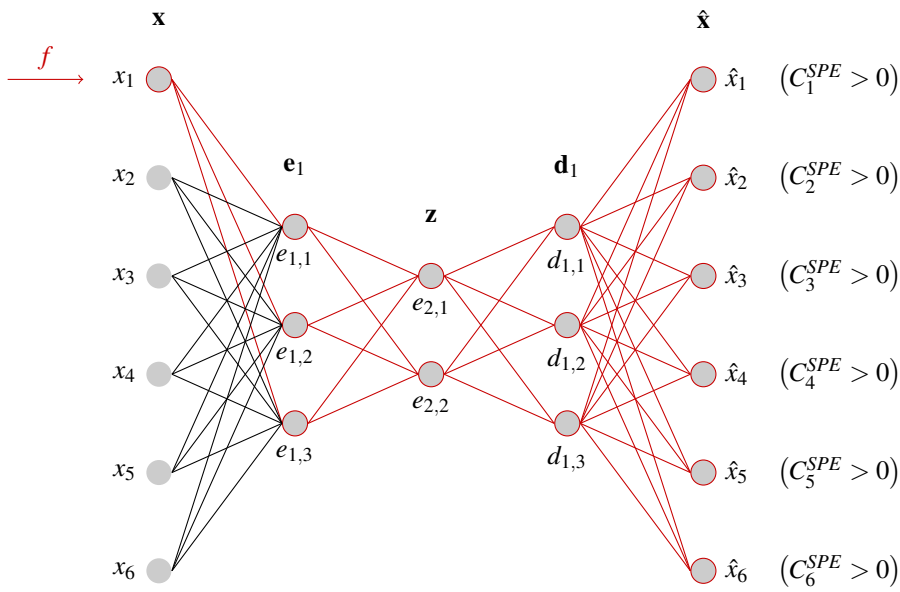


Figure 3.10: Illustration of the fault-smearing effect for an AE. Biases are not included in order to improve visibility. Red edges indicate the propagation of the fault f through the AE.

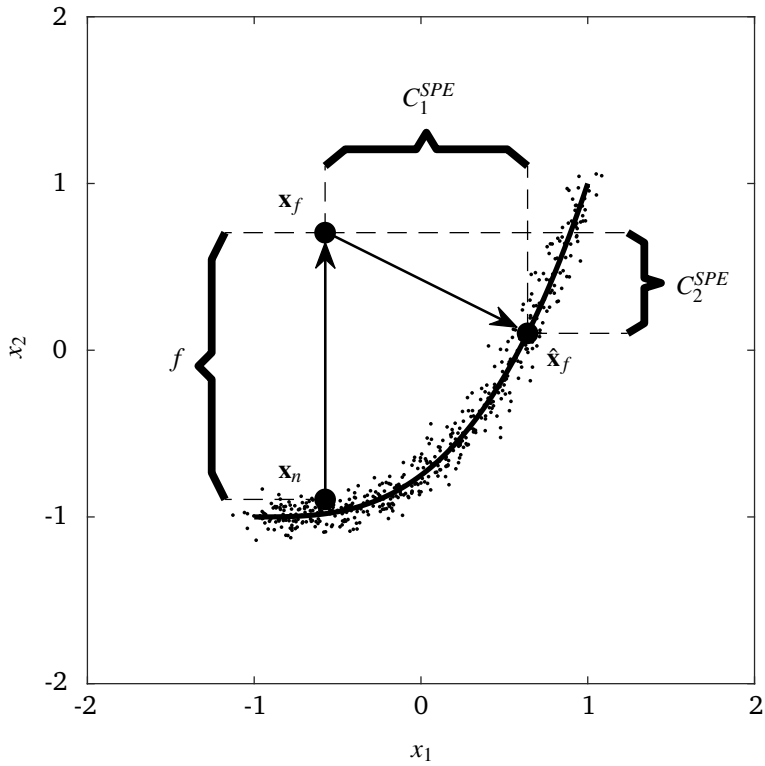


Figure 3.11: Illustration of the fault-smearing effect with a scatter plot. The fault f causes a positive shift from sample $\mathbf{x}_n$ to sample $\mathbf{x}_f$. Nonlinear LP generates the reconstruction at $\hat{\mathbf{x}}_f$.

for the process variables of a nonlinear static process. Two process variables are considered for ease of illustration. The figure shows that nominal samples are summarized by a nonlinear principal curve. The sample pair $\mathbf{x}_n$ is the last indication that the process is operating under nominal conditions. An abnormal event causes a positive shift in x_2 by a magnitude of f. The sample pair $\mathbf{x}_f$ is sampled following the onset of the abnormal event. LP towards the principal curve generates the reconstruction $\hat{\mathbf{x}}_f$. The contributions C_1^{SPE} and C_2^{SPE} are determined from the SPE between $\hat{\mathbf{x}}_f$ and $\mathbf{x}_f$. The figure shows that the fault-smearing effect generates contributions that are unrepresentative of the abnormal event: though the fault f causes an abnormal shift in x_2, the contributions C_1^{SPE} and C_2^{SPE} indicate that both x_1 and x_2 suffer from an abnormal change. The erroneous results from contribution analysis are that (a) the contribution C_2^{SPE} does not correspond to the magnitude of the fault f; and (b) the contribution C_1^{SPE} incorrectly suggests that the fault causes an abnormal shift in x_1.

3.4 Effect of standardization

It is usually the case that the scalings between process variables differ. Consider the vector $\mathbf{x}$ of two process variables x_1 and x_2, where the former is a measurement of pressure with a scale in pascals and the latter is a measurement of pressure with a scale of kilopascals. Both variables measure the pressure of gas inside a tank; hence, x_1 and x_2 are linearly correlated. Assume that the pressure of gas varies. Due to differences in scaling between x_1 and x_2, a variance in gas pressure induces a larger variance in x_1 compared to x_2. As a result, x_1 is a dominant contributor to the total variance in $\mathbf{x}$. Differences in scaling among multiple process variables can affect the estimation for the dimension q of a LP model's latent vector $\mathbf{z}$. Consequently, $\mathbf{x}$ is typically standardized with zero mean and unit variance to ensure that the variance of each variable contributes equally.

Figure 3.12 illustrates the effect of standardization on contribution analysis. Two process variables are considered for ease of illustration. Figure 3.12(a) shows that the variance of x_1 is greater than that of x_2 for unstandardized samples. An abnormal event causes a positive shift in x_2 by a magnitude of f. The sample pair $\mathbf{x}_f$ is sampled following the onset of the abnormal event. LP towards the principal curve generates the reconstruction $\hat{\mathbf{x}}_f$. The contributions C_1^{SPE} and C_2^{SPE} are determined from the SPE between $\hat{\mathbf{x}}_f$ and $\mathbf{x}_f$. Note that C_2^{SPE} is larger than C_1^{SPE}. Figure 3.12(b) plots the same data after standardization. Standardization ensures that the variance of x_1 is equal the variance of x_2. However, standardization changes the perceived influence of the fault such that the abnormal shift in standardized x_2 is larger than the shift in unstandardised x_2. Standardization also affects the results from contribution analysis by changing the magnitude of contributions such that C_1^{SPE} is equal to C_2^{SPE}

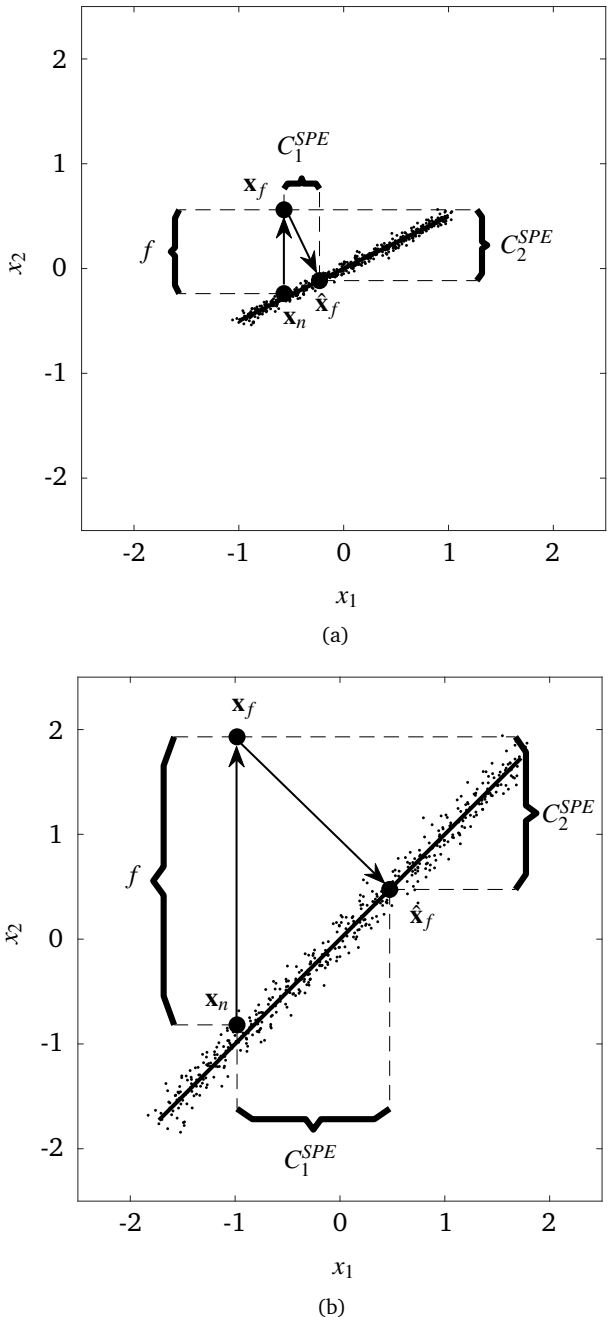


Figure 3.12: Scatter plots of (a) unstandardized samples and (b) standardized samples. Plots illustrate the influence of standardization on contribution analysis. The fault f causes a positive shift from sample $\mathbf{x}_n$ to $\mathbf{x}_f$. Linear LP generates the reconstruction at $\hat{\mathbf{x}}_f$.

Chapter 4

Summary of Main Contributions

The contributions of the research presented in this thesis cover three topics. The first relates to the modeling of nonlinear correlations among process variables with LP methods, as well as the detection of abnormal events. The second topic concerns the fault-smearing effect and the means to reduce it, as well the task of isolating abnormal process variables via trend analysis. The third topic addresses the LP-based modeling of process variables that exhibit nonlinear cross-correlations. The contributions are reported in three journal articles (each undergoing a peer-review process at the time of thesis submission) and two peer reviewed conference proceedings. The contributions are presented in separate appendices A, B, C, D, and E. The appendices are organized by the date the contributions were submitted to their respective journal or conference proceedings. Their summaries are detailed in this chapter by the order of the three aforementioned topics.

Modeling of nonlinear process variable correlations

(D) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Modeling Correlations of Nonlinear Process Variables with Expanding Autoencoders. *Journal of Process Control*. To be submitted for review.

This study examines the model complexity required for an AE to model nonlinearly correlated process variables. Within the process monitoring literature, common approaches are to configure an AE with (a) hidden encoder layers that gradually reduce the dimension of original variables to the latent space; and (b) hidden decoder layers that gradually restore the dimension of original variables from the latent space. This paper demonstrates that such configurations are not sufficient for modeling certain nonlinearly correlated variables, and it is shown that the AE provides a linear approximation of the principal manifold

that summarizes the nonlinear joint behavior among variables. This paper proposes an AE that includes hidden encoder/decoder layers with dimensions larger than the original variables space. Such an AE, termed *expanding* AE, is trained to identify the principal curves of three distinct two-dimensional variable distributions. The results show that an expanding AE's accuracy at identifying principal curves is attributed to its expanding hidden layers.

(A) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2019). Autoencoder based residual generation for fault detection of quadruple tank system. *IEEE Conference on Control Technology and Applications, p:994-999*.

This study examines the residual generation performance of an AE used for monitoring a quadruple tank process (QTP). The QTP employs a switching controller design: the design switches between two different controller configurations depending on the combined state of several process variables. This induces a bipartite (nonlinear) correlation structure among process variables: process variables are positively correlated for one controller configuration and negatively correlated for the other. Switching controllers are regularly employed in multiobjective process systems. The AE's residual generation performance is compared against a LP model given by PCA. The results show that the AE models the bipartite correlation among process variables while the PCA-based model does not. A fault is induced in the QTP, and the results show that the AE performed better than the PCA-based model at generating a fault-sensitive residual.

The fault-smearing effect and isolation of abnormal process variables

(B) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Improved process diagnosis using fault contribution plots from sparse autoencoders. *21st IFAC World Congress*.

This study examines the fault-smearing effect that is observable when an AE is used for abnormal event detection. Fault-smearing occurs when the constriction of original variables to the latent space permits abnormal variables to interact with nominal variables. Fault-smearing poses a problem for fault contribution analysis. One active area of research within the process and chemical engineering disciplines is the extraction and interpretation of valuable process information from historical process data. The fundamental idea behind proposed methods is to present a model employed for a certain data-based task

(such as LP or system identification) and enlist a sparsity constraint that strips away redundant model coefficients, thereby revealing the simplified structure that underlies the data. The removal of model coefficients ultimately reduces variable interactivity. This paper proposes a method for inducing sparsity in an AE that is used for monitoring a nonlinear TTP - a process consisting of two sub-systems. It is demonstrated that the sparsity constraint produces an interpretable variable grouping effect that reveals the nonlinear correlations among process variables. More specifically, the AE (a) groups together variables that are coupled within the same sub-system; and (b) removes any interactivity between variables that are decoupled between the different sub-systems. A fault is induced in the TTP, and results from performing fault contribution analysis show that sparsity (a) reduces the fault-smearing effect; and (b) makes it possible to determine in which sub-system the fault resides.

(C) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Unsupervised Isolation of Abnormal Process Variables using Sparse Autoencoders. *Journal of Process Control*. Submitted paper under review.

This study proposes a method for isolating abnormal process variables with a sparse AE. The paper explains that fault-induced movements in the process variable space propagate through an AE to cause abnormal movements in the reconstruction space. Previous works have shown that a sparse AE reduces the interactivity of variables by removing redundant network connections, thereby exposing the simplified structure that underlies the data. It is shown that a simplified network structure makes it possible to propagate the detected fault-induced movements in the reconstruction space backwards through the sparse AE to isolate the abnormal movements in the original variable space. The method is demonstrated with two distinct faults occurring in a simulated TTP. One of the faults causes abnormal shifts in multiple process variables. The results show that the proposed method can isolate the abnormal process variables that are affected by a fault, even if the fault affects multiple variables.

Modeling of nonlinear cross-correlations

(E) Hallgrímsson, Á.H., Niemann, H.H., and Lind. M (2020). Fault detection with recurrent autoencoders. *Journal of Process Control*. To be submitted for review.

This study investigates the fault detection performance of a recurrent autoencoder (RAE) used for monitoring of a dynamic system that exhibits cross-correlated process variables. A RAE is an AE that includes internal states

(memory). This alows the RAE to exhibit temporal dynamic behaviour when processing its input. The paper explains that methods such as dynamic PCA, i.e., methods that perform dynamic LP by including past observations in its original variable space, are not dynamic in a traditional sense since they do not include any internal states. Rather, these methods mimic dynamic behavior by performing simultaneous computation on current and past observations. In this paper, a comparison in terms of fault detection performance is done between a linear RAE and a model derived via dynamic PCA. Both models are used for monitoring a linear process. The paper shows that RAEs are more sensitive to a fault and thus offer better performance at fault detection. The paper also demonstrates the nonlinear capabilities of a nonlinear RAE used for monitoring a nonlinear process.

Chapter 5

Conclusions and Future Research

5.1 Conclusion

Motivated by the challenges of detecting and evaluating abnormal events in complex industrial systems, the purpose of this research project was to contribute to the area of multivariate statistical process monitoring. Advantage was taken from recent developments in neural network-based machine learning methods proposed in other research fields. This allows for the modeling and monitoring of nonlinear dynamic processes, as well as the isolation of abnormal process variables.

The first contribution of this research project relates to the development of an evidential-based quantitative method for detecting abnormal events in an industrial process. The goal was to propose a method that established a diagnostic system capable of distinguishing between nominal and abnormal process variable observations. It was assumed that the process was nonlinear and dynamic: consequently, process variables would be nonlinearly cross-correlated. The proposed method is facilitated with AEs: neural networks that are configured for LP - a numerical method for performing feature extraction. Case studies showing cases of nonlinearly correlated variables showed that accurate modeling of nonlinear correlations required hidden encoder/decoder layers with dimensions larger than the original variable space. It was shown that such an AE identifies a nonlinear principal manifold that summarizes the nonlinear joint behavior between variables. Training an AE to accurately model the correlations among nominal process variables facilitated abnormal event detection. Case studies show that an AE generates a monitoring residual that (a) is robust to nonlinear correlations present among process variables; and (b) is sensitive to previously unseen faults. A RAE, which provides a dynamic formulation of an AE, was proposed for the monitoring of dynamic processes. The results show that RAEs provide better fault detection performance than standard dynamic approaches to

LP-based process monitoring approaches.

The second contribution of this research project relates to the development of an evidential-based qualitative method that isolates abnormal process variables. The goal was to propose a method that determines the trends (movements) of abnormal process variables with an AE that has been trained on nominal process data. The proposed method is facilitated by augmenting an AE's original optimization function with a sparsity constraint. This promoted for a small number of high importance network connections as redundant connections were pruned away. The result was a sparse AE. Probing into a sparse AE provided insight into the process knowledge the AE had captured. More specifically, the sparse AE reveals the nominal correlations among process variables. It was shown that sparsity reduced the interactivity among process variables, which consequently reduced the fault-smearing effect that is present in fault contribution analysis; process variables unaffected by a fault produced significantly less contributions, while affected variables produced larger contributions. When combined with a simplified network structure, the increased fault contribution disparity between process variables makes it possible to propagate the detected fault-induced movements in the reconstruction space backwards through the sparse AE to isolate the abnormal movements in the original variable space. The method is entirely dependent on the availability of nominal historical process data. In other words, prior instances of faults are not required when isolating abnormal process variables.

As a whole, the research project contributes to the area of AEM. Recent advances in information and process monitoring technologies have made modern industrial systems a cesspool of sensory information, and it has become increasing difficult for operators to comprehend and act upon an abnormal event due to an overload of diagnostic information. This area of AEM, i.e., the processing of process information to (a) detect and abnormal event; and (b) assessing abnormal process variable trends, requires innovation if operators are to perform qualitative root-causal analysis, particularly when under time constraints. The results from the proposed methods show that such innovation in the context of nonlinear and dynamic industrial processes is a possible reality.

5.2 Future research

The detection and isolation of abnormal process variables by means of AE-based anomaly diagnosis is an open field for future research. Key extensions of the research presented in this thesis are summarized in the following:

Monitoring of large processes: The methods proposed in this project were

demonstrated on simulated processes that comprised a handful of process variables. In fact, the quadruple tank process was the "largest" system that was composed of eight process variables. A natural direction in continuing this research project would be to demonstrate the proposed methods on much larger systems that are physically available, i.e., not simulated. This would test their scalability for larger system, as well as provide an indication of their suitability for real-world applications.

Application of different network activation functions: The AEs trained in this project employed primarily the tangent hyperbolic activation function. The reason for this design choice was based on the smoothness and monotonicity of the tangent hyperbolic function. These two properties produced AEs with interpretable model structure: smoothness ensured that there were no functional discontinuities in the modeling of the reconstructions from the original variables, while monotonicity preserved the influence order of neurons in layer l on neurons in layer $l+1$. The smoothness and monotonicity properties were essential when isolating abnormal process variables with a sparse AE, as the nonlinear influence of the tangent hyperbolic activation function could be omitted. There are, however, circumstances where other activation functions may be more practical. For example, the linear rectifier function should be employed when process variables exhibit piecewise linear correlations since the rectifier is a piecewise activation function. The feasibility and diagnostic performance of an AE using different activation functions given certain process variable distributions should be considered for further research.

Non-unique isolation of abnormal process variables: The AEs optimized in this project always provided a unique (non-ambiguous) isolation of abnormal process variables. It is possible, however, for an AE to provide a non-unique (ambiguous) isolation of abnormal process variables. The concept of uniqueness is illustrated by reference to Figure 5.1. The figure shows a sparse AE trained for monitoring a nonlinear process. Three correlated process variables are considered for ease of illustration. The AE is used to detect a fault for the observation $\mathbf{x}_{new}$, and one of the observed movements in the reconstruction space was a positive shift in $\hat{x}_1$. This shift is propagated backwards through the AE in Figure 5.1(a). Propagating this shift backwards through the AE demonstrates the non-uniqueness property of the network: inferring a shift in x_2 produces an ambiguous result because the shift is either (a) positive due to the positive causal relation between x_2 and $e_{1,1}$; or (b) negative due to the negative causal relation between x_2 and $e_{1,2}$. Figure 5.1(b) shows the result of backpropagating a second observed movement in the reconstruction space, namely, a negative shift in $\hat{x}_2$. The figure shows that inferring a shift in z_1 produces an ambiguous result due to the conflicting influences from $d_{1,2}$ and $d_{1,3}$. The ambiguous shift in z_1 generates further ambiguous shifts backwards through the network such

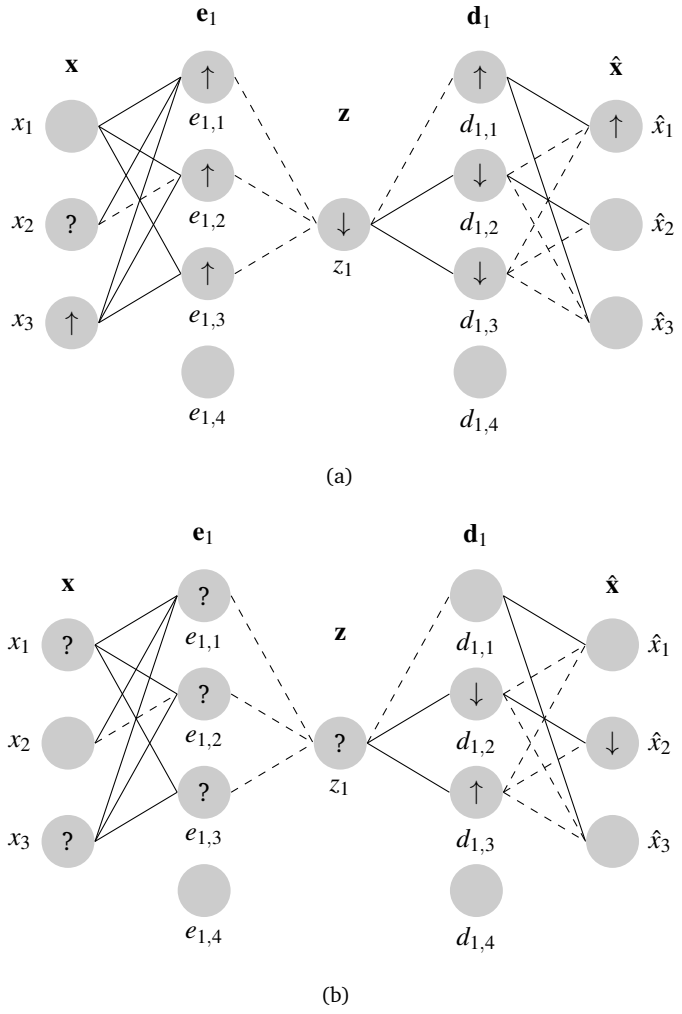


Figure 5.1: Causal inference for a positive shift in $\hat{x}_1$. The result is a positive shift in x_3 and an ambiguous shift in x_2. Biases have been removed for improved visibility.

that inferring a shift in x_1 and x_3 is difficult. A resolution to the issue of non-unique isolation of abnormal process variables is vital if the proposed isolation method is to be used for variable distributions that require an AE with the non-unique property to perform LP. One direction the project had proposed was to evaluate the sign of the derivative of a neuron in layer $l+2$ with respect to a change in the activation of an ambiguous neuron in layer l, given the observation $\mathbf{x}_{new}$. This would determine the path that a change in the ambiguous neuron propagates through layer $l+1$. For the example in Figure 5.1(b), the proposed method corresponds to determining the

sign of the derivative of $\hat{x}_2$ with respect to z_1. If, for instance, the sign is negative (given the observation $\mathbf{x}_{new}$), i.e., a positive change in z_1 induces a negative change in $\hat{x}_2$, then the influence of a variation in z_1 experienced by $\hat{x}_2$ is propagating through the neuron represented by $d_{1,3}$. Following this logic, the shift in z_1 is determined to be a positive shift since the shift in $\hat{x}_2$ is negative. Figure 5.2 shows the result of propagating the resolved shift in z_1 backwards through the AE. The size differences of the arrows in the neurons represented by $d_{1,2}$ and $d_{1,3}$ illustrates the path a shift in z_1 propagates forwards through the AE: the negative change in $\hat{x}_2$ could only have been explained by a positive change in z_1 that propagated through $d_{1,3}$, since the sign of the derivative of $\hat{x}_2$ with respect to z_1 is negative. Unfortunately, the validity of the proposed method for resolving the non-uniqueness property had not been tested at the time of thesis submission.

Isolation of abnormal process variables for dynamic systems: The proposed method for isolating abnormal process variables was demonstrated with static AEs in this project. However, static AEs were shown to produce false positives for fault detection when the monitored process variables observed dynamic transients that were induced by nominal reference changes. This meant that it was necessary to wait for the process to reach steady-state to confirm that a false positive had occurred. RAEs were proposed to alleviate this issue. The results show that RAEs (a) capture static and dynamic nominal process variable transients in its latent representations; and (b) are capable of distinguishing between a transient induced by nominal reference changes and a transient induced by an abnormal event. The natural direction of the research project was to demonstrate the proposed method for isolating abnormal

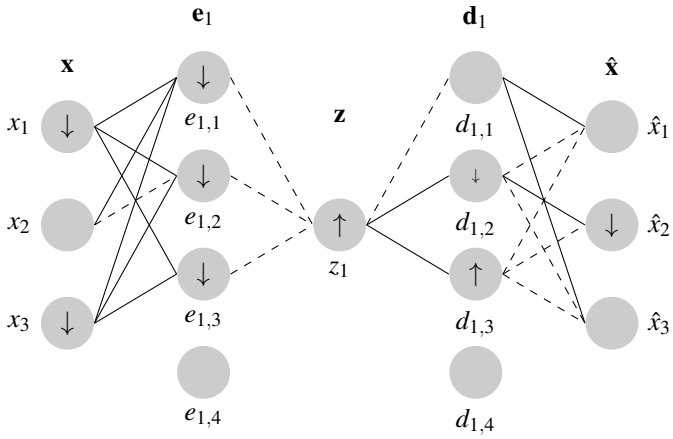


Figure 5.2: Causal inference for a positive shift in $\hat{x}_1$. The result is a positive shift in x_3 and an ambiguous shift in x_2. Biases have been removed for improved visibility.

process variables with a RAE. However, the effectiveness of the proposed method relies on the structural interpretability of a sparse static/RAE: propagating observed abnormal movements in the reconstructions backwards through the AE becomes difficult if its structure is too complex. Demonstrating the method with a sparse RAE proved difficult since the sparse structure consisted of weight connections that represented both static and time dependent variable correlations. A method for isolating abnormal process variables in a dynamic setting is vital if RAEs are to be used to diagnose abnormal events. One direction the project had proposed is to train a static AE and RAE in parallel. Figure 5.3 illustrates this concept. Each AE is provided with the original variables $\mathbf{x}$ as its input. The static AE is to generate the vector $\hat{\mathbf{x}}^s$ that contains the static latent information about $\mathbf{x}$, while the RAE is to generate the vector $\hat{\mathbf{x}}^r$ that contains the dynamic latent information about $\mathbf{x}$. The two AEs are optimized so that the sums of $\hat{\mathbf{x}}^s$ and $\hat{\mathbf{x}}^r$ produces the reconstruction vector $\hat{\mathbf{x}}$ that contains both the static and dynamic latent information about $\mathbf{x}$. The proposed architecture would decouple the static-correlations and cross-correlations among variables into two separate AEs. This could enhance model interpretability sufficiently enough so that the isolation of abnormal process variables is feasible.

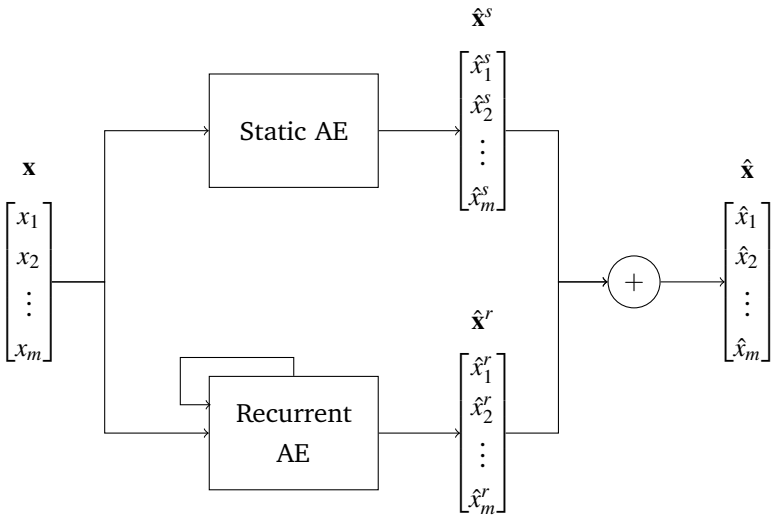


Figure 5.3: Proposed architecture.

Paper A

Autoencoder Based Residual Generation for Fault Detection of Quadruple Tank System

Ásgeir Daniel Hallgrímsson[1,*], Hans Henrik Niemann[1], Morten Lind[1]

[1]Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

Abstract:
Increasing complexity of industrial processes has made statistical methods for process monitoring and diagnosis a more attractive alternative to model-based methods. A primary reason is that statistical approaches can be formulated to rely less on process knowledge. Since multivariable processes can exhibit complex, nonlinear dynamics, there is a need for methods capable of diagnosing nonlinear process data. A Monte Carlo simulation was conducted on a numerical model of the quadruple tank process (QTP) - a novel multivariate nonlinear process. The simulation was designed so that the QTP exhibited bipartite nonlinear behavior. Reference data obtained from the simulation was used to obtain principal component analysis PCA and autoencoder AE models. The models generated residuals that were used to monitor the condition of the process. The results showed that AEs, which have nonlinear functionalities, performed better than PCA models at generating residuals.

[*]Corresponding author. E-mail: asdah@elektro.dtu.dk

A.1 Introduction

The development of model-based fault detection methods for large-scale processes, such as complex industrial systems, can require a considerable high effort. Quantitative descriptions may be difficult to formulate from first principles due to lack of *a priori* knowledge of the process. A developers lack of experience in the the *ab initio* approach further hinders development. Data driven approaches and statistical methods, which can be formulated to not rely on knowledge of the process, offer an alternative way. This approach to quality control is more formally known as statistical process control (SPC). The objective is to evaluate the performance of a process with a quality variable that signifies whether the process is remaining in a state of statistical control [91]. Most SPC methods applied in industry are based on prioritizing a small number of process variables, i.e., measurement and control signals, and examining them independently. Operators typically monitor the quality of a process by observing traditional univariate control charts such as Schewart, CUSUM, and EWMA [15], [91]. Despite their popular use, their successful performance is hampered by the assumption that process variables are independent of one another and that a single quality variable can verify that the collective process is out of statistical control. Instead, the statistical condition of a process should be defined as a multivariate property that takes into consideration the simultaneous quality of all process variables. Abnormal events, such as faults and incorrect control decisions, can cause unexpected changes in the associated variation between variables, which should be identified in order to diagnose the abnormality. A second limitation of the univariate approach is the difficulty in selecting which variables to monitor. Improper selection may yield incorrect diagnosis since events may be reflected in unmonitored variables. Lastly, the univariate approach is challenged by the increasing prevalence of industrial big data. Industries across different areas of production are shifting towards generating data with higher complexity by increasing the number of sensors and computers connected to every industrial process [142]. Industrial big data yields data sets that are too large for univariate methods. Consequently, prioritization on which variables, some of which may not be well explained nor understood, to monitor becomes difficult.

Dimensionality reduction techniques of multivariate processes for abnormal event detection have gained increasing interest over the past decade [105]. Latent projection (LP) methods, which transform data to a latent space of fewer dimensions, are of particular interest. In principle, LP methods reduce the number of original process variables by obtaining a set of principal variables that guarantee minimal information loss. An LP model is built with historical data, i.e., a reference set,

collected from a process existing in an "in-control" state. The quality of a process is then monitored with a residual that reflects the amount of information loss for new data observations. If an abnormal event occurs, the relations between process variables may change such that the latent projection no longer applies to the new observations. As a result, the principal variables cannot retain the information in the original process data, causing the residual to increase.

The most well known LP method is principal component analysis (PCA). It is a linear statistical procedure that determines a set of orthogonal vectors called principal components (PCs) that point in the direction of maximal variance of reference data [114]. A process is monitored by comparing the direction of variance of new data samples with the reference PCs. Abnormal events typically cause unexpected changes in the covariance structure of the process variables, which then differ from the obtained PCs [75].

Since PCA is a linear transformation, it is ineffective at obtaining principal directions for nonlinear data. This can be illustrated by reference to Fig. A.1, which depicts the distribution of process variables x_1 and x_2 contained in $\mathbf{X}_a$:

$$\mathbf{X}_a = \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[n] \\ x_2[1] & x_2[2] & \cdots & x_2[n] \end{bmatrix} \in \mathbb{R}^{2\times n} \tag{A.1}$$

The variables were generated from a simulated bi-modal process in which the association between x_1 and x_2 depended on the mode the process operated in. The two modes are illustrated by $\mathbf{X}_b$ and $\mathbf{X}_c$ in Fig. A.1. The plot depicts the nonlinear nature between x_1 and x_2 and how different PCs are obtained based on which partition, i.e., $\mathbf{X}_a$, $\mathbf{X}_b$, or $\mathbf{X}_c$, is used as reference data. The first PC for $\mathbf{X}_a$ depicts that, on average, x_1 and x_2 are positively correlated. Even though it points in the direction of maximal variance, it does not provide a meaningful description for the nonlinear dependency. Quality control with this PC model would generate false positives for abnormal event detection at the extremes of x_1 and x_2. On the other hand, the first PCs for $\mathbf{X}_b$ and $\mathbf{X}_c$ provide a more precise description for the bi-modal behavior observed by the process. In essence, the PCs between x_1 and x_2 are contextual, and depend on which mode the process is in.

If a process exhibits nonlinearly correlated data, quality control with PCA is achievable by monitoring individual PC models build on reference data obtained by sufficiently partitioning the data set into modal parts and other sets that depict known nonlinear behavior. A drawback of such an approach is that it requires knowledge about the nonlinear nature of the process. Furthermore, processes with a large number of process variables may have complex relationships and variables with smooth dependencies can be difficult to partition. The increasing prevalence of nonlinearly correlated data has given rise to new algorithms that perform nonlinear

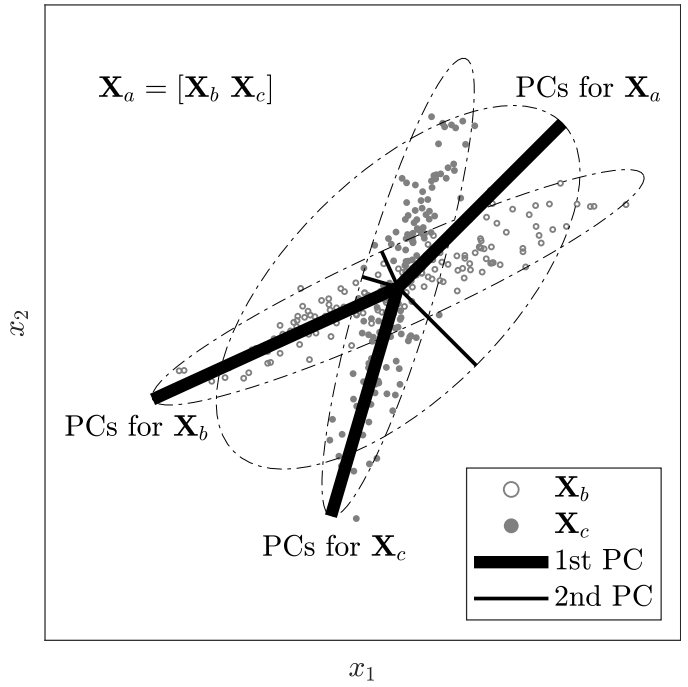


Figure A.1: Scatter plot for x_1 and x_2 contained in data matrix $\mathbf{X}_a$, where $\mathbf{X}_a$ has been further partitioned into $\mathbf{X}_b$ and $\mathbf{X}_c$. The two PCs of each of the three reference sets are included. A primary PC has maximum variance and a secondary PC has minimum variance. The 99% confidence intervals of the PCs are included.

dimensionality reduction. Several nonlinear LP methods perform a nonlinear form of PCA, most notably kernel-based PCA [119]. Principal components are obtained in high-dimensional, implicit feature spaces generated via a nonlinear mapping of the original variable space. The kernel trick is used so that the method can operate in the feature space without performing the nonlinear mapping, reducing the computational complexity of the method. One drawback is that kernel operations on one sample involves computations of the entire data set used to obtain kernel-based PCs. Therefore, if years of process data is used to generate a kernel-based PCA model, the computational complexity of online process monitoring will be high.

LP machine learning techniques such as autoencoders (AEs) offer an alternative approach [130]. An AE is a type of artificial neural network trained to learn optimal nonlinear transformations via backpropagation of its reconstruction error. AEs offer a computationally efficient approach to working with large data sets since computation of LPs is independent on the sample size of the reference set.

In this paper, a comparison study between PCA and AE residual generation

models is performed on a numerical model of the quadruple tank process (QTP). The QTP is a multivariate, nonlinear process that depicts a bipartite nonlinear relationship between its process variables that, in the context of decentralized control, is introduced via a change in its input-output control pairing [64]. The key result in this study was that the AE was significantly better at generating residual signals for the purpose of fault detection.

This paper presents the mathematical model of the QTP in section II. Section III describes the PCA method and the AE is presented in section IV. Section V presents how the dynamic behaviour of process variables can to be taken into consideration when applying a LP method. Section V also describes how LP methods are used to generate residuals that monitor the QTP. The effectiveness of the LP methods at process monitoring are presented in section VI.

A.2 The Quadruple Tank Process

A schematic drawing of the QTP is given in Fig. A.2. The four tanks are supplied with liquid that is transported from a large sump by the means of two gear pumps. Liquid flows from the upper tanks into the lower tanks, which sequentially flows into the sump. The relative supply of liquid across the four tanks is determined by the configuration of two dual valves. The objective is to control the liquid levels in the lower two tanks, which are monitored with two voltage-based level measurement devices. A nonlinear numerical model of the QTP is derived by applying mass balances and Bernouilli's law to yield a set of differential equations that describes the evolution of the liquid level of each tank. They are:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1} v_1(1+\eta_1) \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2} v_2(1+\eta_2) \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3} v_2(1+\eta_2) \\
\frac{dh_4}{dt} &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4} v_1(1+\eta_1)
\end{aligned} \tag{A.2}$$

where A_i is the cross-section of tank i and a_i is the cross-section of its outlet hole. The liquid level of tank i is h_i and g is acceleration due to gravity. The voltage applied to pump i is v_i and the corresponding flow is $k_i v_i(1+\eta_i)$, where $\eta_i \in \mathbb{R}$ is zero mean Gaussian noise emitted from pump i. An interesting aspect of the quadruple-tank system is the physical interpretation for the process in terms of how the valves γ_1 and γ_2 are set. In particular, the nonlinear system is non-minimum phase for

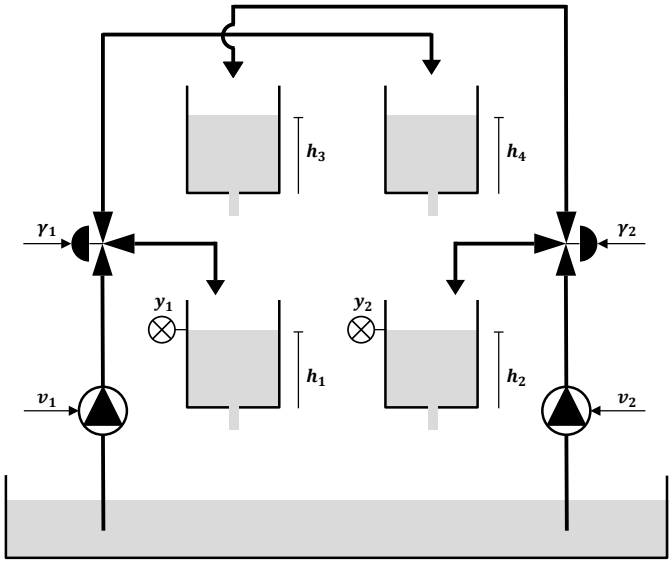


Figure A.2: A schematic of the QTP illustrating the connectivity of the tanks and location of the pumps, dual valves, and the level measurement devices.

$0 < \gamma_1 + \gamma_2 < 1$ and minimum phase for $1 < \gamma_1 + \gamma_2 < 2$. It follows that, through the application of relative gain array (RGA), the following input-output paring rule for decentralized control is derived [64]:

$$\begin{aligned} R_1 &: 0 < \gamma_1 + \gamma_2 < 1 \rightarrow v_1 \text{ controls } y_2,\ v_2 \text{ controls } y_1 \\ R_2 &: 1 < \gamma_1 + \gamma_2 < 2 \rightarrow v_1 \text{ controls } y_1,\ v_2 \text{ controls } y_2 \end{aligned} \tag{A.3}$$

The pairing rule in (A.3) introduces a bipartite nonlinearity in the QTP, i.e., (v_1, v_2) is correlated to (y_2, y_1) for $0 < \gamma_1 + \gamma_2 < 1$ and correlated to (y_1, y_2) for $1 < \gamma_1 + \gamma_2 < 2$. The system is measured and actuated discretely with a sample time of T_s. The measured level signals at sample k are:

$$\begin{aligned} y_1[k] &= k_c h_1[k] + w_1[k] \\ y_2[k] &= k_c h_2[k] + w_2[k] \end{aligned} \tag{A.4}$$

where $w_i[k] \in \mathbb{R}$ is zero mean measurement noise with Gaussian distribution for level signal i. For decentralized control, the error terms are:

$$\begin{aligned} e_1[k] &= r_1[k] - y_1[k] \\ e_2[k] &= r_2[k] - y_2[k] \end{aligned} \tag{A.5}$$

where $r_1[k]$ and $r_2[k]$ are reference signals for level signals $y_1[k]$ and $y_2[k]$, respectively. The error terms are minimized by a discrete PI controller configured with the

proportional gain K_P and integral gain K_I. The controller adequately sets the voltages v_1 and v_2 by taking into consideration the pairing rule in (A.3). Monte Carlo simulations were performed on the QTP in order to generate reference data that exhibited the bipartite nonlinear behavior imposed by the pairing rule in (A.3). The reference data was subsequently used to obtain PCA and AE models. The uncertain parameters were configurations of the two dual valves γ_1 and γ_2. Values for γ_1 and γ_2 were sampled from two independent uniform distributions. Process, controller, noise, and Monte Carlo parameters are listed Table A.1 [64].

A.3 Principal Component Analysis

It is a common occurrence that process variables in large-scale processes are highly correlated with one another. It is then of practical interest to reduce the dimensions of the original variable space to a lower dimension to reveal the simplified structure that underlie it. A well known approach for dimensionality reduction is PCA. Given a $m \times 1$ vector of process variables $\mathbf{x}$, the $m \times n$ reference data matrix consisting of n standardized observations is:

$$\mathbf{X} = \begin{array}{c} \begin{matrix} \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[n] \end{matrix} \\ \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[n] \\ x_2[1] & x_2[2] & \cdots & x_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ x_m[1] & x_m[2] & \cdots & x_m[n] \end{bmatrix} \end{array} \in \mathbb{R}^{m \times n} \tag{A.6}$$

The first PC of $\mathbf{x}$ is the linear transformation $t_1 = \mathbf{x}^\intercal \mathbf{p}_1$ that has maximum variance subject to $|\mathbf{p}_1| = 1$. The second PC is the linear transformation $t_2 = \mathbf{x}^\intercal \mathbf{p}_2$ that has the second greatest variance subject to $|\mathbf{p}_2| = 1$, and subject to the condition that it be orthogonal to the first PC. Additional PCs up to m are similarly defined. The PCs

Table A.1: List of Parameters

Process param.		**Noise param.**		**Controller param.**	
A_1, A_3	28 cm^2	η_1	$\mathcal{N}(0, 10^{-4})$	T_s	10
A_2, A_4	32 cm^2	η_2	$\mathcal{N}(0, 10^{-4})$	K_P	20
a_1, a_3	0.071 cm^2	w_1	$\mathcal{N}(0, 10^{-3})$	K_I	0.1
a_2, a_4	0.057 cm^2	w_2	$\mathcal{N}(0, 10^{-3})$		
k_c	1 V/cm				
k_1	3.33 cm^3/Vs	**Monte Carlo param.**			
k_2	3.35 cm^3/Vs	γ_1	$\mathcal{U}(a_1, b_1)$		
g	981 cm/s^2	γ_2	$\mathcal{U}(a_2, b_2)$		

form the orthonormal principal component loading matrix $\mathbf{P}$ obtained by solving for the eigenvectors of the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{X}$:

$$\mathbf{\Sigma} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\intercal} \tag{A.7}$$

where $\mathbf{\Lambda}$ is a non-negative real diagonal $m \times m$ matrix whose diagonal elements are the corresponding eigenvalues. The diagonal entries λ_i of $\mathbf{\Lambda}$ are the variances of the PCs. The principal component scores are defined as the observed values of the PCs for each of the n observation vectors:

$$\mathbf{t}_i = \mathbf{X}^{\intercal}\mathbf{p}_i, \quad i = 1, 2, ..., m \tag{A.8}$$

Essentially, PCA decomposes the process matrix $\mathbf{X}$ as:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\intercal} = \sum_{i=1}^{m} \mathbf{t}_i \mathbf{p}_i^{\intercal} \tag{A.9}$$

It is often the case that a small number of PCs is sufficient to account for most of the variability in the data. The first q PCs are determined with the cumulative percent variance (CPV) approach to capture at least 85% of total variance:

$$\frac{\sum_{i=1}^{q} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \times 100\% \geq 85\% \tag{A.10}$$

In this manner, dimensionality reduction is achieved by identifying q PCs that explain most of the predictable variations in the data. The remaining $q - m$ PCs are typically associated with random noise present in the data. The $\mathbf{X}$ matrix is thus approximated by:

$$\hat{\mathbf{X}} = \sum_{i=1}^{q} \mathbf{t}_i \mathbf{p}_i^{\intercal} \tag{A.11}$$

A.4 Autoencoders

An AE is an artificial neural network used for dimensionality reduction. An AE consists of two parts, an encoder and a decoder. The encoder transforms high-dimensional input into lower-dimensional features. The decoder then reconstructs the original data with a transformation of the features [130]. Modifiable parameters are introduced in the AE such that it learns in an unsupervised manner to minimize the difference between its input and its reconstruction. The AE essentially learns to compress data into a lower-dimensional representation that captures its essential information. The compressed data, being sufficiently representative of the original data, allows for accurate reconstruction of the input data.

The simplest form of an AE is a multilayered, feedforward, non-recurrent neural network. Nonlinear transformations occur at the layers of the network, allowing for

processing of data that has inherent nonlinear properties. An illustration of an AE is given in Fig. A.3. The encoder maps the input $\mathbf{X} \in \mathbb{R}^m$ to the latent variables $\mathbf{Z} \in \mathbb{R}^q$:

$$\mathbf{E}_i = \begin{cases} \sigma_1^e \left(\mathbf{W}_1^e \mathbf{X} + \mathbf{b}_1^e\right), & \text{for } i = 1 \\ \sigma_i^e \left(\mathbf{W}_i^e \mathbf{E}_{i-1} + \mathbf{b}_i^e\right), & \text{else} \end{cases} \tag{A.12}$$

$$\mathbf{Z} = \sigma^z \left(\mathbf{W}^z \mathbf{E}_N + \mathbf{b}^z\right)$$

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer. The decoder maps the latent variables $\mathbf{Z} \in \mathbb{R}^q$ to the input reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^m$:

$$\mathbf{D}_i = \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{Z} + \mathbf{b}_1^d\right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{D}_{j-1} + \mathbf{b}_j^d\right), & \text{else} \end{cases} \tag{A.13}$$

$$\hat{\mathbf{X}} = \sigma^{\hat{x}} \left(\mathbf{W}^{\hat{x}} \mathbf{D}_M + \mathbf{b}^{\hat{x}}\right)$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the activation function at layer j. $\mathbf{W}^{\hat{x}}$, $\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer. The modifiable parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with respect to minimizing the following loss function via stochastic gradient descent [99]:

$$\mathscr{L}(\mathbf{X}, \hat{\mathbf{X}}) = \left|\left|\mathbf{X} - \hat{\mathbf{X}}\right|\right|^2 \tag{A.14}$$

A.5 Dynamic latent projections

Directly applying a LP method on the reference data matrix $\mathbf{X}$ will construct a static model. When the data contains dynamic information and the correlation of variables is time dependent, projecting it to a latent space will not reveal the exact relations between the variables but rather a static approximation[75]. Furthermore, the transformed variables will be auto-correlated and possibly cross-correlated.

The dynamic data matrix $\bar{\mathbf{X}}$ is generated by introducing a properly chosen 'time lag shift' to the data matrix $\mathbf{X}$. Applying a LP method on $\bar{\mathbf{X}}$ will construct latent

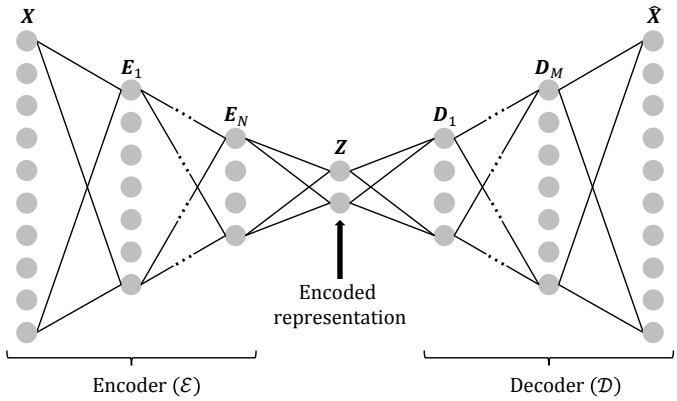


Figure A.3: Illustration of an AE. Labels for the encoder and decoder of the network are included.

variables that retain the cross-correlation between variables. Defining the time shifted process variable vector:

$$\bar{\mathbf{x}}_i[k] = \begin{bmatrix} x_i[k] & x_i[k-1] & \cdots & x_i[k-l] \end{bmatrix}^\mathsf{T} \tag{A.15}$$

where $i \in \mathbb{Z} : i \in [1,m]$, l is the time lag parameter, and $k \in \mathbb{Z} : k \in [l+1,n]$, the dynamic data matrix is:

$$\bar{\mathbf{X}} = \begin{array}{c} \begin{matrix} \bar{\mathbf{x}}[1] & \bar{\mathbf{x}}[2] & \cdots & \bar{\mathbf{x}}[n] \end{matrix} \\ \begin{bmatrix} \bar{\mathbf{x}}_1[l+1] & \bar{\mathbf{x}}_1[l+2] & \cdots & \bar{\mathbf{x}}_1[n] \\ \bar{\mathbf{x}}_2[l+1] & \bar{\mathbf{x}}_2[l+2] & \cdots & \bar{\mathbf{x}}_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_m[l+1] & \bar{\mathbf{x}}_m[l+2] & \cdots & \bar{\mathbf{x}}_m[n] \end{bmatrix} \end{array} \tag{A.16}$$

where $\bar{\mathbf{X}} \in \mathbb{R}^{(m\cdot(l+1))\times(n-l)}$. With regards to $\mathbf{X}$, the columns of $\bar{\mathbf{X}}$, i.e., $\bar{\mathbf{x}}[k]$, are composed of the sequence of columns $(\mathbf{x}[k], \mathbf{x}[k-1], \ldots, \mathbf{x}[k-l])$. Matrix $\bar{\mathbf{X}}$ can be interpenetrated as describing the evolution of the past l samples in $\mathbf{X}$.

Having built a dynamic LP model based on historical data collected when only common cause variation was present, future behavior can be referenced against this "in-control" model. New observations can be reconstructed from their projection onto the latent space to obtain the residuals $\mathbf{e}_{new} = \bar{\mathbf{x}}_{new} - \hat{\bar{\mathbf{x}}}_{new}$. An abnormal event, whose response is not present in the reference data used to establish the LP model, can be detected by computing the squared prediction error (SPE) of the residuals of new observations [73]:

$$SPE = \sum_{i=1}^{m} \left(\bar{x}_{new,i} - \hat{\bar{x}}_{new,i} \right) \tag{A.17}$$

A.6 Results and Discussion

A.6.1 Reference data generation from QTP simulation

Reference data was generated by performing a Monte Carlo simulation of the QTP. Two simulation scenarios, noted as S_1 and S_2, were designed. Each scenario specified the upper and lower bounds for the uniform distributions for γ_1 and γ_2. In S_1, $\gamma_1 \sim \mathscr{U}(0.1, 0.2)$ and $\gamma_2 \sim \mathscr{U}(0.1, 0.2)$. In S_2, $\gamma_1 \sim \mathscr{U}(0.8, 0.9)$ and $\gamma_2 \sim \mathscr{U}(0.8, 0.9)$. The scenarios were designed to generate data that demonstrated the influence of the input-output paring rule in (A.3) on the dynamics of the system. From the paring rule in (A.3), rule R_1 was followed in simulation scenario S_1 and rule R_2 was followed in scenario S_2. Table A.2 presents the sampled values for γ_1 and γ_2. Each scenario generated 3 unique configurations for γ_1 and γ_2, with each configuration simulated for 6000 time steps. This resulted with a reference set of $n = 36000$ observations. During each configuration, the process was excited by performing 30 step changes to reference signals r_1 and r_2. The simulated control signals v_1 and v_2, measurement signals y_1 and y_2, and dual valve configurations γ_1 and γ_2 were collected into the following data matrix:

$$\mathbf{X} = \begin{array}{c} \begin{array}{cccc} \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[n] \end{array} \\ \begin{bmatrix} v_1[1] & v_1[2] & \cdots & v_1[n] \\ v_2[1] & v_2[2] & \cdots & v_2[n] \\ y_1[1] & y_1[2] & \cdots & y_1[n] \\ y_2[1] & y_2[2] & \cdots & y_2[n] \\ \gamma_1[1] & \gamma_1[2] & \cdots & \gamma_1[n] \\ \gamma_2[1] & \gamma_2[2] & \cdots & \gamma_2[n] \end{bmatrix} \end{array} \tag{A.18}$$

A.6.2 Fault data generation from QTP simulation

The QTP was simulated with a decrease in the gain of pump 1, introduced via the multiplicative fault:

$$k_{1,f} = k_1 \cdot (1 - f) \tag{A.19}$$

with $f = 0.1$. Reference signals were generated with the same simulation procedure detailed in the previous subsection. The fault was introduced after 200 time steps. This way, the first 200 samples of the simulation data described a process operating under faultless conditions, whereas the remainder depicted a faulty process. Values for γ_1 and γ_2 were sampled from scenario S_2 and are presented in Table A.2. Due to the selection of γ_1, γ_2, and reference signals, the LP models were tested with previously unseen observations. Since a portion of the data described an "in-control" process, it was possible to test the robustness of the generated residuals to new,

Table A.2: Sampling of γ_1 and γ_2

Set type	γ_1	γ_2	Pairing rule
Reference	0.1134	0.1426	R_1
	0.1107	0.1351	R_1
	0.1234	0.1043	R_1
	0.8127	0.8913	R_2
	0.8441	0.8350	R_2
	0.8347	0.8381	R_2
Fault	0.8000	0.8302	R_2

normal data. The same signals in the reference data matrix $\mathbf{X}$ were collected in the fault data matrix $\mathbf{X}_f$.

A.6.3 Latent projection model generation and testing

A dynamic PCA model, named model M_1, and a dynamic AE model, named model M_2, were generated with the results from simulating the QTP. Since the pairing rule in (A.3) introduced a bipartite nonlinearity in the QTP, it was expected that model M_1 would be ineffective at retaining the nonlinear variance of the variables. Consequently, the model would be less effective at generating residual signals.

The time lag parameter was selected as $l = 2$ by following the selection procedure in *Ku et al.* [75]. The dynamic reference data matrix $\bar{\mathbf{X}}$ was subsequently generated from $\mathbf{X}$ with (A.15) and (A.16), resulting with 18 row vectors. The dynamic fault data matrix $\bar{\mathbf{X}}_f$ was similarly obtained from $\mathbf{X}_f$. The matrices $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}_f$ were standardized with the mean and standard deviation of $\bar{\mathbf{X}}$. The CPV approach in (A.10) was applied on $\bar{\mathbf{X}}$, resulting with $q = 3$ PCs being required to capture at least 85% of total variance. To ensure consistency, M_1 was generated with $q = 3$ PCs and the dimension of the latent layer in M_2 was set to $q = 3$. The dimensions and activation function of each layer in the encoder and decoder were chosen as:

$$\begin{bmatrix} \dim_{\mathrm{L}}(\mathscr{E}) \\ \sigma_i^e \end{bmatrix} = \begin{bmatrix} \mathbf{X} & \mathbf{E}_1 & \mathbf{E}_2 \\ 18 & 54 & 54 \\ & \tanh & \mathrm{ReLU} \end{bmatrix}, \qquad \begin{bmatrix} \dim_{\mathrm{L}}(\mathscr{D}) \\ \sigma_i^d \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 & \hat{\mathbf{X}} \\ 54 & 54 & 18 \\ \mathrm{ReLU} & \tanh & I \end{bmatrix} \tag{A.20}$$

where tanh is the tangent hyperbolic function, ReLU is the rectifier function, and I is

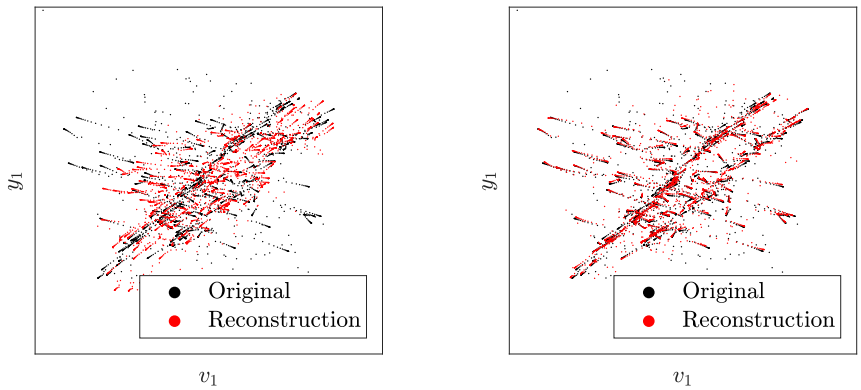


Figure A.4: Scatter plot of original and reconstruction of unshifted, i.e. $x_i[k]$ in (A.15), standardized samples for v_1 and y_1 for models (a) M_1 and (b) M_2.

the identity function. Setting the dimension of the encoder and decoder layers as triple the size of the input dimension, as well as utilizing two different activation functions, allowed the AE to generate complex, higher dimensional features [102]. The tangent hyperbolic function was implemented at the latent layer.

Fig. A.4 illustrates the effectiveness of M_2 at information retention for the reference set. The deterministic part of the process variable pair v_1 and y_1 is plotted, which depicts the bipartite nonlinearity introduced in the QTP; the variables observed both positive and negative correlations, depending on which pairing rule in (A.3) was being followed. This nonlinearity hampered the effectiveness of M_1 at reconstructing the variable pair. On the other hand, M_2 performed much better at data reconstruction.

Fig. A.5 displays the SPE of propagating the deterministic part of dynamic fault data matrix $\bar{\mathbf{X}}_f$ through models M_1 and M_2. The figure exhibits how the signals vary with time as reference changes occurred and when the fault f was introduced. The fault specified in (A.19) was introduced along with a reference change at time t_f. In comparison to M_2, model M_1 generated large values for SPE when the QTP operated nominally. This is because the PCA model performed much worse at reconstructing the data, as is depicted in Fig. A.4. Meanwhile, the SPE of M_2 increased only briefly during reference changes and observed a DC gain increase when the fault was introduced. This demonstrates the sensitivity of the AE model to the simulated fault and its ability to detect it. On the other hand, the PCA model made no clear distinction that a fault had occurred, since its SPE signal was already of a large

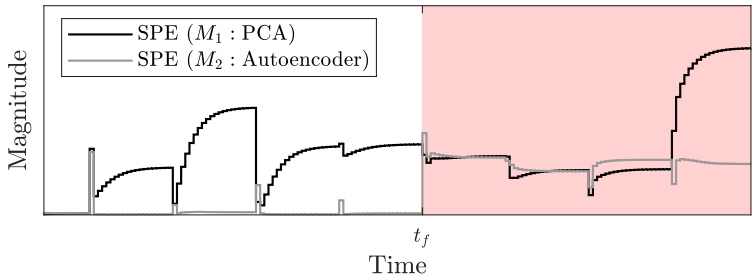


Figure A.5: SPE of propagating the deterministic part of the dynamic fault data matrix $\bar{\mathbf{X}}_f$ through models M_1 and M_2. A fault occurs at time t_f.

magnitude prior to the fault.

Fig. A.6 displays the SPE of each model when the stochastic part of $\bar{\mathbf{X}}_f$ was included, with process and measurement noise now causing osculations in the signals. Compared to M_1, model M_2 was less sensitive to random noise when the QTP operated nominally. The amplitude of the random variations of the SPE of each model increased when the fault was introduced, but M_2 still depicted a clear DC gain increase in its signal, indicating its sensitivity to the fault.

A.7 Conclusion

This paper introduces the application of LP based process monitoring models that verify that a process is remaining in a state of statistical control. Future behavior is referenced against the "in-control" model to detect abnormal events in the process. Two LP methods were presented. The former established a linear PCA model acquired by solving for the eigenvectors of the covariance matrix of sampled reference data. The latter trained an AE, a type of an artificial neural network, to

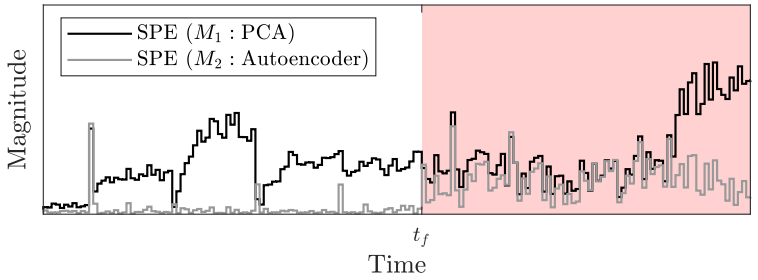


Figure A.6: SPE of propagating the dynamic fault data matrix $\bar{\mathbf{X}}_f$ through models M_1 and M_2. A fault occurs at time t_f.

optimize its learning parameters so as to minimize the information loss resulting from reducing the dimensions of the reference data. This allowed for a comparison of effectiveness at detecting faults in the QTP.

The results demonstrated that the AE based model method was robust to new, unseen observations that described an "in control" process, with large magnitudes in the residuals only occurring during reference changes. The PCA basd model was not robust, as reference changes caused lasting negative influences on its diagnosing capabilities. The results also indicated that the inherent nonlinear nature of the QTP reduced the effectiveness of residual generation with the linear PCA model. On the other hand, due to its nonlinear functional capabilities, the AE showed better performance at generating residuals.

Paper B

Improved Process Diagnosis Using Fault Contribution Plots from Sparse Autoencoders

Ásgeir Daniel Hallgrímsson[1,*], Hans Henrik Niemann[1], Morten Lind[1]

[1]Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

Abstract:
Development of model-based fault diagnosis methods is a challenge when industrial systems are large and exhibit complex process behavior. Latent projection (LP), a statistical method that extract features of data via dimensionality reduction, is an alternative approach to diagnosis as it can be formulated to not rely on process knowledge. However, LP methods may perform poorly at identifying abnormal process variables due a "fault smearing" effect - variables unaffected by a fault are unintentionally characterized as being abnormal. The effect occurs because data compression permits faulty and non-faulty variables to interact. This paper presents an autoencoder (AE), a nonlinear LP method based on neural networks, as a monitoring method of a simulated nonlinear triple tank process (TTP). Simulated process data was used to train the AE to generate a monitoring statistic representing the condition of the TTP. Sparsity was introduced in the AE to reduce variable interactivity. The AE's ability to detect a fault was demonstrated. The individual contributions of process variables to the AE's monitoring statistic were analyzed to reveal the process variables that were no longer consistent with normal operating conditions. The key result in this study was that sparsity reduced fault smearing onto unaffected variables and increased the contributions of actual faulty variables.

[*]Corresponding author. E-mail: asdah@elektro.dtu.dk

B.1 Introduction

Effective online monitoring of process performance is integral for maintaining stable plant operation, maximizing production, and ensuring the survivability of industrial systems. In fact, abnormal events that disrupt plant performance can cause up to 8% annual loss in production profit [16]. Due to the increasing complexity of large-scale industrial processes, statistical methods - which can be formulated to not rely on process knowledge - are a practical alternative to more traditional and rigorous model-based fault detection methods. The relevance of statistical monitoring schemes is further supported by the current trend of industries to generate industrial big data thanks to the integration of additional sensors, computers, and other technological artifacts connected to every industrial process [142].

This approach to quality control is known as statistical process control (SPC) [91]. An important component of SPC is diagnosis of a detected abnormal event and determining its cause. Once an unintended plant upset is identified, it is typically up to the operators to decipher which statistical quality variables contain signal characteristics that help diagnose the problem. Unfortunately, industrial application of SPC-based event diagnosis is ineffective since the most common practice for monitoring the quality of a process is to observe traditional univariate control charts such as Schewart, CUSUM, and EWMA [15]. Their application inherently assumes that process variables are independent of one another, potentially making their use ineffective at diagnosing events that affect multiple process variables.

Multivariate quality control (MQC) methods - which produce quality variables that summarize the condition of several process variables - are a better alternative to univariate approaches for monitoring of multivariable processes [105]. Essentially, the Hotelling's T^2 and Q statistics are paired with latent projection (LP) - dimensionality reduction methods such as principal component analysis (PCA) that uncover the correlation structure of data - to detect out-of-control situations. A process is monitored by comparing current plant behavior with an LP model representing its "in-control" behavior. An abnormal event that changes the correlation between process variables is detected when the monitored deviation between the current process state and that predicted by the model exceeds a threshold.

Industrial applications of LP-based process monitoring tend to use linear methods, such as PCA, due to their ease of implementation. Unsurprisingly, linear methods result in high Type I and Type II error rates if the process is nonlinear [45], [144]. Nonlinear extensions of PCA have emerged to uncover both linear and nonlinear correlations between variables. The focus of this paper is on autoencoders (AEs); a type of artificial neural network that learns salient, encoded representations via

nonlinear transformations of an original data set. Dong et al. [32] show that AEs can discover principal curves, i.e., a one-dimensional curve whose shape provides a nonlinear summary of the nonlinear structure of the complex data set it passes through. Kramer [72] demonstrates significant improvement in nonlinear feature extraction by using a multi-layered AE as opposed to a single-layered AE, assuming that the dimension of latent layers were consistent.

Recent advances in AE-based process monitoring have been made by including developments from other applications of neural networks. Yan et al. [144] observed improved fault detectability of the Tennessee Eastman process over PCA-based process monitoring by using novel variants of AEs; denoising AEs, which reconstruct the uncorrupted version of corrupted input data, and contractive AEs, which penalize the sensitivity of hidden representations to small (noisy) perturbations around the input. Lee et al. [85] used a variational autoencoder (VAE) to enforce the monitored data to follow a multivariate normal distribution in the latent space to facilitate appropriate use of Hotelling's T^2 monitoring charts for nonlinear and non-normal processes, resulting in a reduction of Type I and Type II error rates. Osmani et al. [103] monitored the condition of a turbo-compressor using a recurrent neural network (RNN) that captured temporal dependency of process variables with the additional regularization constraint that activations in the reduced space followed a Bernoulli distribution. Cheng et al. [22] combined VAEs and RNNs to produce a variational recurrent neural network for fault detection of the Tennessee Eastman process.

Contributions in the AE-based SPC literature tend to prioritize fault detection over fault isolation. Much of the subject matter focuses on reducing Type I and Type II error detection rates by: (a) increasing model sensitivity to faults; (b) obtaining more robust and complex monitoring statistics; and (c) reducing hampering effects from nominal process changes. Though AEs have been used as a pre-training step for fault-classification networks when labeled fault data is scarce [121], few methods exist where fault isolation is performed exclusively with an AE. However, rudimentary diagnosis with PCA models can be carried out with the analysis of fault contribution plots [63], [95]. The plots indicate the contributions of process variables to an observed increase in the T^2 or Q statistic, with variables showing large contributions concluded as no longer following nominal operating conditions. Operators can then apply process knowledge to determine an appropriate cause.

There are reports of fault contribution plots suffering from a property called "fault smearing" - variables unaffected by the fault demonstrate a contribution and actual faulty variables are obscured [138]. Smearing occurs because the compression of the input to a smaller number of latent variables and subsequent expansion to the

original space permits faulty and non-faulty variables to interact [128].

Gao et al. [39] imposed an elastic net constraint to obtain a sparse PCA model for the Tennessee Eastman process. The result was a reduction in interactivity between variables in the latent space. It subsequently lead to the discovery of process knowledge, specifically the relationships among process variables.

The objective of this paper is to extend the analysis of fault contribution plots to AEs and investigate the effect reduced latent variable interactivity has on process variable contribution. Two AEs - a dense one and a sparse one - are generated to monitor a numerical simulation of a nonlinear triple tank process (TTP) - a variant of the quadruple tank process (QTP) [64]. Their ability to detect a fault is demonstrated by inducing an abnormal bias in one of the TTP's sensors. Individual contributions of process variables to the AEs' monitoring statistics are then analyzed. The key result in this study was that sparsity reduced fault smearing onto non-faulty variables and increased the contributions of faulty variables.

This paper presents the mathematical model of the TTP in section II. Section III describes how a sparse AE is obtained and subsequently used in process monitoring. The effectiveness of the sparse AE method at process monitoring and improved generation of fault contribution plots is presented in section IV.

B.2 The Triple Tank Process

A schematic drawing of the TTP is given in Fig. B.1. The upper tanks are supplied with liquid that is transported from a large sump by the means of two gear pumps. Liquid flows from the upper left tank into the sump. The liquid from the upper right tank flows into the lower tank, which sequentially flows into the sump. The objective is to control the liquid levels in the upper left and lower right tanks, which are monitored with two voltage-based level measurement devices. A level measurement device is also fixed to the upper right tank. A nonlinear numerical model of the TTP is derived by applying mass balances and Bernouilli's law to yield a set of differential equations that describes the evolution of the liquid level of each tank. They are:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{1}{2}\frac{k_1}{A_1}v_1(1+\eta_1) \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{1}{2}\frac{k_1}{A_2}v_1(1+\eta_1) + \frac{k_2}{A_2}v_2(1+\eta_2) \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{a_2}{A_3}\sqrt{2gh_2}
\end{aligned} \tag{B.1}$$

where A_i is the cross-section of tank i and a_i is the cross-section of its outlet hole. The liquid level of tank i is h_i and g is acceleration due to gravity. The voltage applied

to pump i is v_i and the corresponding flow is $k_i v_i(1+\eta_i)$, where $\eta_i \in \mathbb{R}$ is zero mean Gaussian noise emitted from pump i. The system is measured and actuated discretely with a sample time of T_s. The measured level signals at sample k are:

$$\begin{aligned} y_1[k] &= k_c h_1[k] + w_1[k] \\ y_2[k] &= k_c h_2[k] + w_2[k] \\ y_3[k] &= k_c h_3[k] + w_3[k] \end{aligned} \tag{B.2}$$

where $w_i[k] \in \mathbb{R}$ is zero mean measurement noise with Gaussian distribution for level signal i. For decentralized control, the error terms are:

$$\begin{aligned} e_1[k] &= r_1[k] - y_1[k] \\ e_2[k] &= r_2[k] - y_3[k] \end{aligned} \tag{B.3}$$

where $r_1[k]$ and $r_2[k]$ are reference signals for level signals $y_1[k]$ and $y_3[k]$, respectively. The error terms are minimized by a discrete PI controller. The closed loop control laws for the process inputs are:

$$\begin{aligned} K_1: \; v_1[k] &= K_P e_1[k] + K_I \sum_{i=1}^{k} e_1[i] T_s \\ K_2: \; v_2[k] &= K_P e_2[k] + K_I \sum_{i=1}^{k} e_2[i] T_s \end{aligned} \tag{B.4}$$

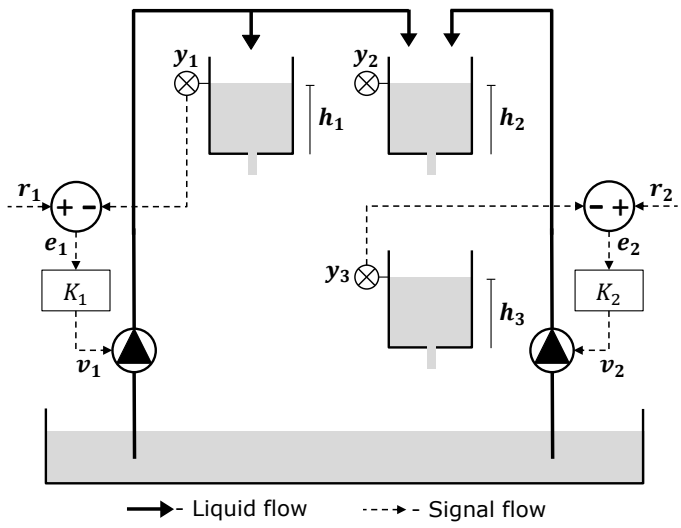


Figure B.1: A schematic of the TTP showing the connectivity of the tanks and location of the pumps, dual valves, and the level measurement devices. Included are the decentralized feedback loops.

Here K_P and K_I denote the proportional and integral gains, respectively, of the PI controller. Monte Carlo simulations were performed on the TTP to generate data sets that exhibited nonlinear correlations between the process variables. The data sets were used to train, validate, and test an AE model that monitored the process. The uncertain parameters were the reference signals r_1 and r_2. Values for r_1 and r_2 were sampled from two independent uniform distributions. Process, controller, and noise parameters were based on the QTP from [64] and are listed in Table B.1.

B.3 Autoencoders

Process variables tend to be highly correlated with one another due to the presence of physical laws and control loops in process plants. Feature extraction can be performed on the original variable space to reveal the simplified structure that underlies it. An AE - an artificial neural network used for learning encoded representations for a set of data - is applicable when variables exhibit nonlinear correlations. Given a $m \times 1$ vector of process variables $\mathbf{x}$, the $m \times n$ reference data matrix consisting of n standardized observations is:

$$\mathbf{X} = \begin{array}{c} \begin{matrix} \mathbf{x}[1] & \mathbf{x}[2] & \cdots & \mathbf{x}[n] \end{matrix} \\ \begin{bmatrix} x_1[1] & x_1[2] & \cdots & x_1[n] \\ x_2[1] & x_2[2] & \cdots & x_2[n] \\ \vdots & \vdots & \ddots & \vdots \\ x_m[1] & x_m[2] & \cdots & x_m[n] \end{bmatrix} \end{array} \in \mathbb{R}^{m \times n} \tag{B.5}$$

An AE consists of two parts - an encoder and a decoder. The encoder transforms its input $\mathbf{X}$ into new, higher-level representative features $\mathbf{Z} \in \mathbb{R}^{q \times n}$. The decoder then reconstructs the original data as $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$ with a transformation of the features [50]. Modifiable interconnecting weights are introduced in the AE such that it learns in an unsupervised manner to minimize the difference between its input and its reconstruction.

The simplest form of an AE is a multilayered, feedforward, non-recurrent neural network. Nonlinear transformations occur at the layers of the network, allowing for processing of data that has inherent nonlinear properties. The encoder maps the input $\mathbf{X} \in \mathbb{R}^{m \times n}$ to the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$:

$$\mathbf{E}_i = \begin{cases} \sigma_1^e \left(\mathbf{W}_1^e \mathbf{X} + \mathbf{b}_1^e \right), & \text{for } i = 1 \\ \sigma_i^e \left(\mathbf{W}_i^e \mathbf{E}_{i-1} + \mathbf{b}_i^e \right), & \text{else} \end{cases} \tag{B.6}$$

$$\mathbf{Z} = \sigma^z \left(\mathbf{W}^z \mathbf{E}_N + \mathbf{b}^z \right)$$

Table B.1: List of Parameters

Process param.		**Noise param.**	
A_1, A_3	28 cm^2	η_i	$\mathcal{N}(0, 0.1)$
A_2	32 cm^2	w_i	$\mathcal{N}(0, 0.0005)$
a_1, a_3	0.071 cm^2		
a_2	0.057 cm^2		
k_c	1 V/cm	**Controller param.**	
k_1	3.33 cm^3/Vs	T_s	10
k_2	3.35 cm^3/Vs	K_P	20
g	981 cm/s^2	K_I	0.25

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer.

The decoder maps the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$ to the input reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$:

$$\mathbf{D}_i = \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{Z} + \mathbf{b}_1^d \right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{D}_{j-1} + \mathbf{b}_j^d \right), & \text{else} \end{cases} \tag{B.7}$$

$$\hat{\mathbf{X}} = \sigma^{\hat{x}} \left(\mathbf{W}^{\hat{x}} \mathbf{D}_M + \mathbf{b}^{\hat{x}} \right)$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{x}}$, $\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer. The modifiable parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with respect to minimizing the following reconstruction loss function via stochastic gradient descent [99]:

$$\mathscr{L}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n} \left|\left| \mathbf{X} - \hat{\mathbf{X}} \right|\right|^2 \tag{B.8}$$

Fig. B.2 illustrates a typical autoencoder that gradually condenses the input to the latent space and then gradually reconstructs it. The dimension q of the latent layer plays a significant role in the discovery of informative representations of the input. The traditional approach is to create a bottleneck by setting $q < m$, thereby forming an under-complete representation. In this case, the network pursues an effective compression that retains information about input $\mathbf{X}$. The compressed data, being sufficiently representative of the original data, allows for accurate reconstruction of

the input data, albeit with a non-zero reconstruction error. An AE using linear nodal activation functions will uncover latent projection that correspond to the projection onto the subspace obtained from PCA of the covariance matrix of $\mathbf{X}$ [7]. This occurs even if the network is composed of several layers of linear units. However, Bourlard and Kamp [13] show that PCA-like projections can be obtained even if nonlinear functions are used since it is possible for activations to remain in the linear regions of functions such as the sigmoid or tangent hyperbolic. This becomes unlikely if the AE is composed of several hidden layers with varying activation functions [59].

B.3.1 Invoking network sparsity

Further optimization constraints are introduced to obtain latent representations that generalize better and prevent over-fitting. One approach is to include the naïve elastic net weight decay penalty - a regularized regression method that linearly combines the L_1 and L_2 weight decay penalties of the LASSO and ridge methods [154]. The loss function in (B.8) becomes:

$$\mathscr{L}(\mathbf{X},\hat{\mathbf{X}},\mathbf{W}) = \frac{1}{n}\left|\left|\mathbf{X}-\hat{\mathbf{X}}\right|\right|^2 + \lambda_1 ||\mathbf{W}||_1 + \lambda_2 ||\mathbf{W}||_2^2 \tag{B.9}$$

where λ_1 and λ_2 control the importance of the LASSO and ridge regressions, respectively, and $\mathbf{W}$ is the collection of weight matrices in (B.6) and (B.7). Biases $\mathbf{b}$ in (B.6) and (B.7) are not included in the naïve elastic net penalty. Minimization of (B.9) yields an optimized AE consisting of shrunk weights that minimize its reconstruction loss. The individual contribution of each regularization term is: (a) L_1 regularization shrinks weights at a constant rate towards zero, thereby establishing a small number

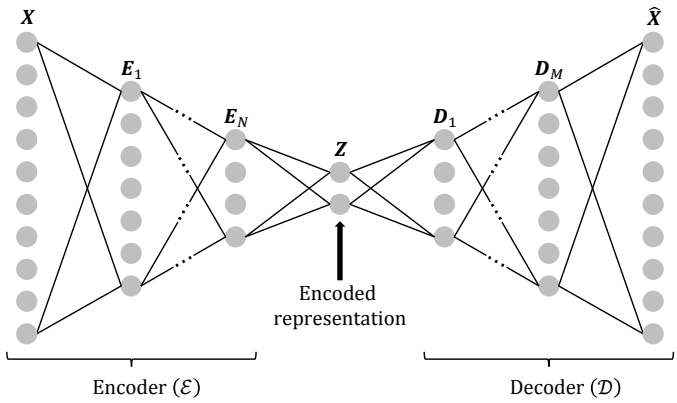


Figure B.2: Illustration of an under-complete AE. Labels for the encoder and decoder of the network are included. Biases are excluded from the illustration.

of high-magnitude, i.e., high-importance, connections by driving redundant weights to zero; and (b) L_2 regularization shrinks weights by an amount proportional to their magnitude, thus penalizing larger weights more than smaller weights. The net result is an interpretable grouping of correlated variables; L_2 regularization opposes the tendency of L_1 regularization to prioritize one variable from a group correlated variables and ignore the others. Grouping of process variables is relevant for identification of control systems; Gao et al. [39] demonstrate that a sparse principal component model can uncover the underlying process variable relations.

Weight connections deemed redundant can be removed to clarify the interconnectivity of a neural network. Magnitude-based weight pruning is a technique that reduces the number of non-zero weight parameters to invoke network sparsity. Zhu and Gupta [153] introduce a pruning algorithm that progressively trims away redundant weight connections. Weight connections are removed according to a pruning function that sets the current sparsity percentage, i.e., the ratio of the number zero magnitude weights to the total number of weights, of a network:

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t - t_0}{n\Delta t}\right)^3 \text{ for } t \in \{t_0, t_0 + \Delta t, \ldots, t_0 + n\Delta t\} \tag{B.10}$$

The network is first trained for t_0 time steps to permit the weights to converge to an acceptable solution. Thereafter the initial sparsity of the network is set to s_i (usually zero). Weights are then pruned every Δt steps to gradually increase the network's sparsity while allowing it to recover from any pruning-induced loss in accuracy. The intuition behind the order of (B.10) is to rapidly prune the network in the beginning phase when redundant connections are plentiful before slowing down once fewer connections remain (Fig. B.3). The algorithm operates continuously over n sparsity updates until the final sparsity value s_f is reached. Zhu and Gupta [153] discovered that large-sparse models consistently outperformed small-dense models when the

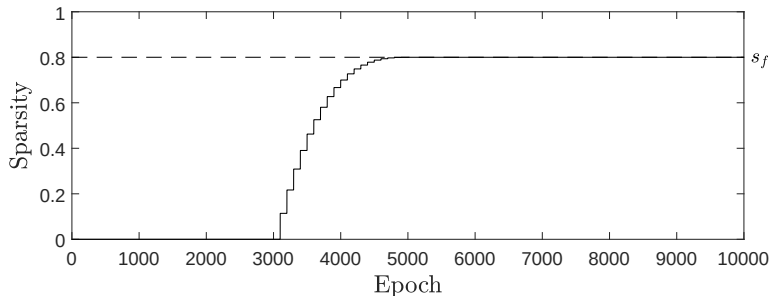


Figure B.3: Example sparsity function used for gradual pruning with $s_f = 0.8$, $s_i = 0.0$, $t_0 = 3000$, $\Delta t = 100$, $n = 20$.

number of parameters was kept the same.

The pruning algorithm presented by Zhu and Gupta [153] is extended upon in this paper. At every sparsity update s_t, each weight matrix $\mathbf{W}_i \in \mathbf{W}$ is divided by the largest absolute value of $\mathbf{W}_i$. This normalization step is done to prevent severe pruning of weight matrices whose largest absolute value is much smaller compared to the other matrices. The normalized matrices are then flattened and concatenated. The smallest weights are then masked to zero until the desired sparsity level s_t is reached. Furthermore, the pruning algorithm is stopped prematurely if the validation loss experiences a 5%-10% increase.

Once pruning ends, the näive elastic net weight penalty is removed from the training session. This is to relax the constraints on the AE and permit the remaining weights to maximize their capacity to reduce the loss function in (B.8) without the concern of any additional loss penalties.

B.3.2 Process Monitoring

Process monitoring consists of comparing current plant behaviour with that predicted by an "in-control" AE trained with historical data collected when the process exhibited nominal behaviour. New observations are propagated through the AE to generate the residuals $\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new}$. The quality of new observations is assessed by computing the squared prediction error (SPE) (more formally known as the Q statistic) of the residuals of new observations [91]:

$$SPE = \sum_{i=1}^{m} (x_{new,i} - \hat{x}_{new,i})^2 \tag{B.11}$$

An abnormal event that changes the correlation between process variables will cause the SPE to increase. Assuming that the SPE follows a chi-squared distribution, the control limit can be computed with the following approximate value [12]:

$$CL_{SPE_{AE}} = \frac{\bar{\sigma}^2}{2\bar{\mu}} \chi^2_{(2\bar{\mu}^2/\bar{\sigma}^2,\alpha)} \tag{B.12}$$

where $\bar{\mu}$ and $\bar{\sigma}$, respectively, are the sample mean and sample standard deviation of the SPE and α is the false alarm rate. An abnormal event is deemed to have occurred if the SPE crosses the control limit. Abnormal process variables are isolated by analysing the contribution of each variable i to the SPE in (B.11) [95]:

$$C_i = (x_{new,i} - \hat{x}_{new,i})^2 \tag{B.13}$$

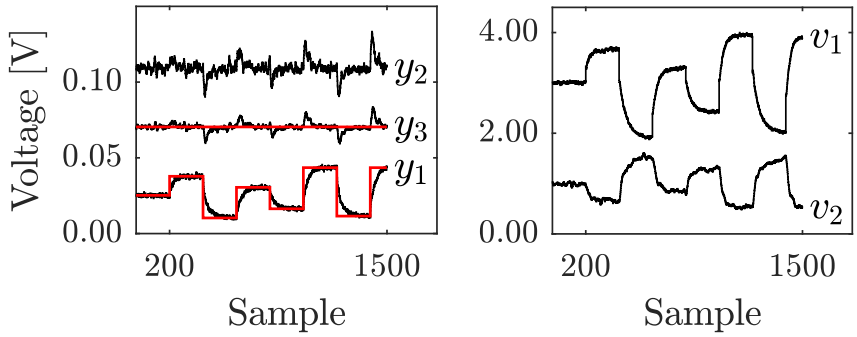
Variables with large contributions are said to no longer be consistent with normal operating conditions. It is noted that analysis of (B.13) does not determine the

underlying cause of a fault. Rather, it will highlight the process variables containing signal characteristics of a fault. The results of (B.13) must be integrated with a qualitative model of the process that takes into account the causal nature of system components to decipher the actual cause.

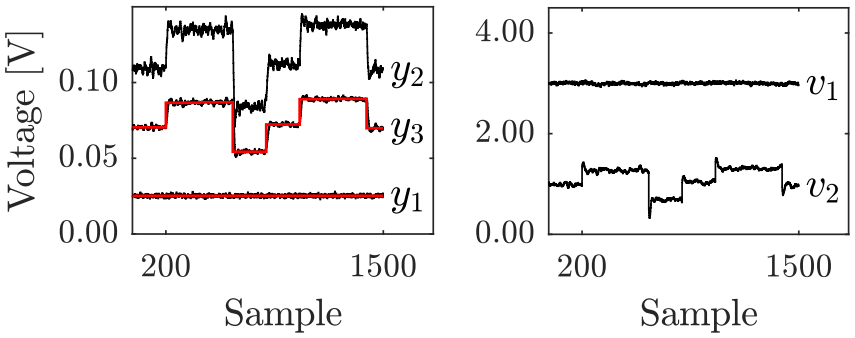
B.4 Results and Discussion

B.4.1 Derivation of influence rules

It was of practical interest to determine the influence of reference variables r_1 and r_2 on the control and measurement variables; steady-state correlations uncovered by the AE can then be validated to what is implied by the data. Fig. B.4 displays the time series plots obtained from inducing random step changes in a single reference



(a)



(b)

Figure B.4: Time series of simulated process variables where (a) r_1 is changed whilst r_2 is held fixed and (b) r_2 is changed whilst r_1 is held fixed. Red lines indicate the references for the measurement.

variable while keeping the other constant. The plots demonstrate that: (a) a step change in r_1 causes a transient change in the steady state values of y_1, v_1, and v_2, while variables y_2 and y_3 experience a transient change that has no affect on their steady-state values; and (b) a step change in r_2 has no influence on y_1 and v_1 yet generates a transient change in the steady state values of y_2, y_3 and v_2. The correlation sets $C_1 = (r_1, y_1, v_1, v_2)$ and $C_2 = (r_2, y_2, y_3, v_2)$ are determined from the plots. They indicate which process variables observe a permanent change in their steady state value caused by a change in a reference signal.

B.4.2 Data generation from TTP simulation

The TTP was simulated with random step changes in reference signals r_1 and r_2 occurring every 200 time steps. The training set $\mathbf{X}_t$ (consisting of 300,000 samples) and validation set $\mathbf{X}_v$ (consisting of 30,000 samples) were generated to train and validate, respectively, an AE. Fig. B.5 displays the distribution of standardized samples of variables in $\mathbf{X}_t$ in the form of scatter and histogram plots. The scatter plots indicate the existence of nonlinear correlations between variable pairs (r_1, v_1),

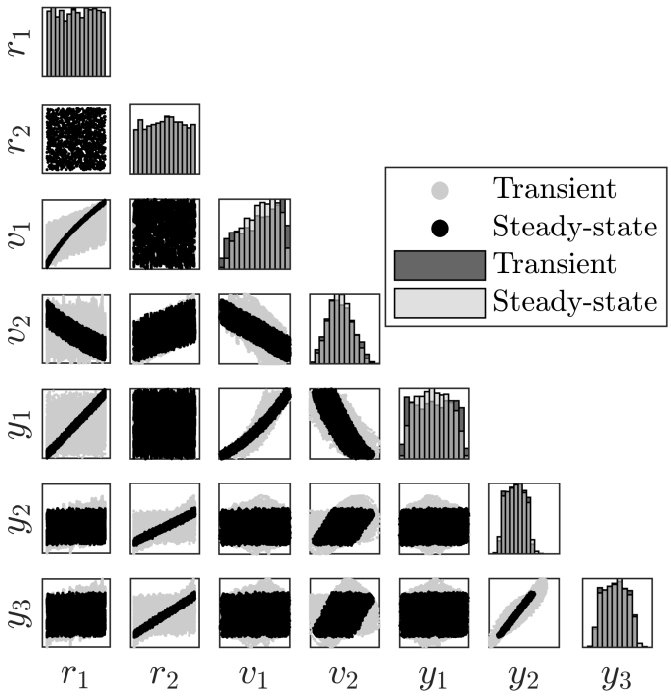


Figure B.5: Scatter plot of standardized process variables, including a histogram along the diagonal.

(v_1, y_1), and (v_2, y_1). The histogram plots reveal that several variables do not follow the assumption of normality with v_1 in particular.

The fault set $\mathbf{X}_f$ (consisting of 300 samples) was generated by simulating the TTP with a bias in sensor 1, introduced with the additive fault $y_1[k] = k_c h_1[k] + w_1[k] + f$ with $f = -0.01$. The fault was introduced after 100 time steps. No reference changes occurred in r_1 and r_2. Fig. B.6 presents time series plots of the first 200 samples of $\mathbf{X}_f$ and shows the fault's effect on the process variables. Deterministic results (in grey) from the same simulation case (obtained by setting η_i and w_i in (B.1), (B.2) to zero) are included to aid in interoperability. The plots demonstrate that: (a) the fault has no influence on r_1 and r_2; (b) the fault induces temporary changes in y_1, y_2, and y_3 that have no influence on their steady state values; and (c) the fault induces a permanent change in v_1 and v_2 and thus carry steady-state signatures that explain the presence of the fault.

B.4.3 AE model generation and testing

Two AEs, denoted AE_1 and AE_2, were trained with the training set $\mathbf{X}_t$. Both networks were inherently the same, i.e., same number and dimension of layers, same number of latent variables, same initialization of the weights, and so on, except AE_2 included the näive elastic net weight penalty with $\lambda_1 = \lambda_2 = 0.001$ and was pruned. The pruning parameters in (B.10) of AE_2 were $s_f = 0.9$, $s_i = 0.7$, $t_0 = 5000$, $\Delta t = 100$, $n = 200$, but early-stopping resulted with a sparsity of 80.86%. Both AEs were trained for 12000 epochs using the Adam gradient-based optimization with a learning rate of 0.001 for stochastic gradient descent [71]. The matrices $\mathbf{X}_t$, $\mathbf{X}_v$, and $\mathbf{X}_f$ were standardized with the mean and standard deviation of $\mathbf{X}_t$. The dimension

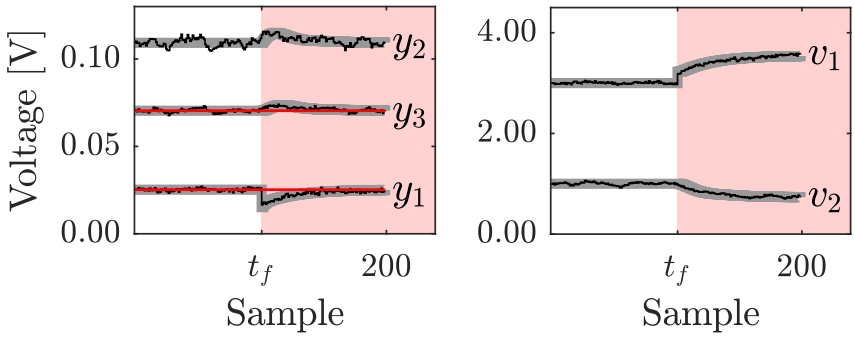


Figure B.6: Influence of fault f_1 at sample t_f on (left) measurements and (right) control inputs. Red lines indicate the references for the measurements.

of the latent layer in each model was set to $q = 2$. This was to see if the sparse AE_2 would expose the correlation sets C_1 and C_2. The dimensions and activation function of each layer were specified as:

$$\begin{bmatrix} \dim_{\mathrm{L}}(\mathscr{E}) \\ \sigma_i^e \end{bmatrix} \Bigg| \begin{bmatrix} \dim_{\mathrm{L}}(\mathscr{D}) \\ \sigma_i^d \end{bmatrix} = \begin{array}{c} \begin{matrix} \mathbf{X} & \mathbf{E}_1 & \mathbf{E}_2 & \mathbf{D}_1 & \mathbf{D}_2 & \hat{\mathbf{X}} \end{matrix} \\ \left[\begin{array}{ccc|ccc} 7 & 9 & 9 & 9 & 9 & 7 \\ & \tanh & \tanh & \tanh & \tanh & I \end{array}\right] \end{array} \tag{B.14}$$

where tanh is the tangent hyperbolic function and I is the identity function. The tangent hyperbolic transfer function was primarily used since the data is mean-centered. The design of the AE is essentially an expanded under-represented AE; setting the dimension of the encoder and decoder layers larger than the size of the input dimension allowed the AE models to generate complex, higher dimensional features before information retaining compression occurred [102]. The tangent hyperbolic function was implemented at the latent layer.

The training loss (TL) and validation loss (VL) from training AE_1 and AE_2 are plotted in Fig. B.7. It can be seen that the TL and VL of AE_2 observe a significant difference that recedes when pruning ends. This is because the TL includes the näive elastic net weight penalty in (B.9) that is then removed once pruning stops. The figure shows that the VL of AE_2 is similar to the VL of AE_1 at the end of training. In fact, the VL of AE_2 is only 7.2% larger despite AE_2 having 80.86% fewer weight parameters than AE_1.

Fig. B.8 portrays the connectivity between network layers of AE_2 and shows the propagation of original variables $\mathbf{x}$ to the reconstructions $\hat{\mathbf{x}}$. It can be seen that the network has identified the correlations between process variables, thus eliminating potential fault smearing between uncorrelated variables. The activation of the

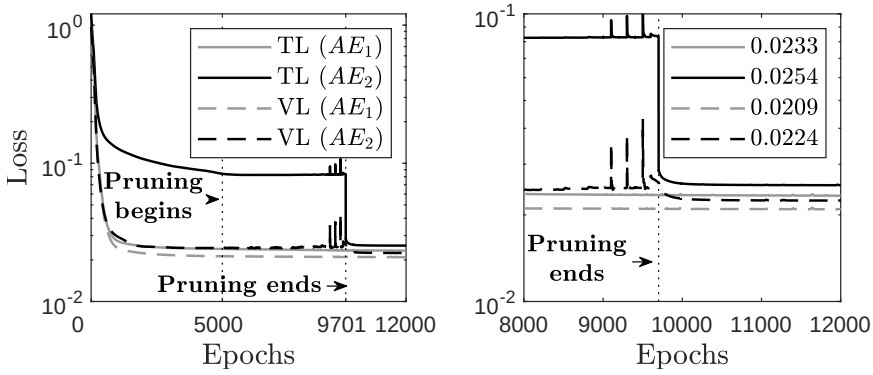


Figure B.7: Training and validation losses during training. Right figure zooms in on epoch interval [8000,12000] and includes final losses in its legend.

second node in the latent layer is computed by the process variables of correlation set C_1. In addition, the activation is solely responsible for the reconstruction of the same variables. The activation of the first latent node is determined by the variables of correlation set C_2 with the exception of v_2. However, the latent node's activation reconstructs all of the variables of C_2. From this it follows that fault signatures contained in v_2 cannot not smear onto $\hat{r}_2$, $\hat{y}_2$, and $\hat{y}_3$. Although it provides a partial explanation for the loss in validation accuracy in comparison to AE_1 (Fig. B.7), AE_2 has discovered a form of reconstruction redundancy: although v_2 appears both in C_1 and C_2, it is sufficient to reconstruct it from a partial subset of process variables. It is noted that the interconnectivity of AE_2 is heavily influenced by the chosen hyperparameters for the learning rate, regression coefficients λ_1 and λ_2, and pruning parameters; a different selection is bound to result with a different connectivity.

The contribution plots obtained from propagating $\mathbf{X}_f$ through AE_1 and AE_2 are displayed in Fig. B.9. Plots from the deterministic equivalent of $\mathbf{X}_f$ are included to ease the analysis of the effect of network pruning on mean contributions. The fault is detected by both AEs as their SPEs cross their control limit at sample t_f, i.e., the onset of the fault. Despite the model complexity of AE_1 being greater than that of AE_2, their SPEs are nearly identical over the fault set. This indicates that a more complex model is not necessarily more sensitive to faults. Smearing onto unaffected variables r_2, y_2, and y_3 is less for AE_2, indicated by a reduction in the variance (Fig B.9(a)) and mean (Fig B.9(b)) of their contributions. In fact, their mean contributions are zero once steady-state is reached because the steady state fault signatures retained in v_1 and v_2 cannot propagate to $\hat{r}_2$, $\hat{y}_2$, and $\hat{y}_3$ (Fig. B.8). Even though smearing occurs onto non fault-carrying variables r_1 and y_1, invoking network sparsity guarantees that the steady state signal characteristics of faulty variables v_1 and v_2 stay within the variables of correlation set C_1. In fact, network AE_2 generates larger contributions for fault-carrying variables v_1 and v_2 and reduces

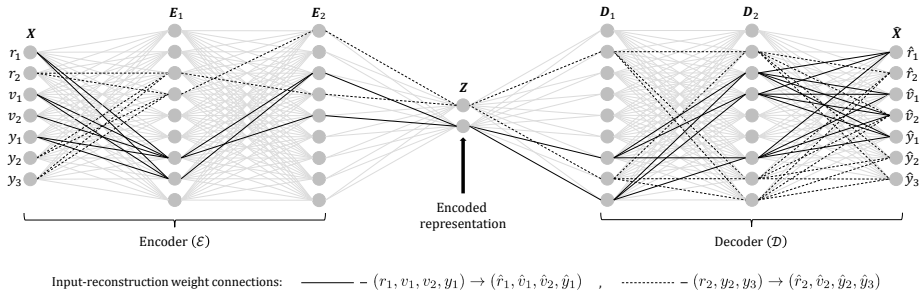


Figure B.8: Illustration of trained AE_2, showing pruned weight connections (grey) and remaining connections (black). Biases have been excluded from the illustration.

the contributions for non-fault-carrying variables r_1 and y_1, indicating that network sparsity makes faulty variables more highlighted.

It is reiterated that analysis of fault contribution plots does not determine the cause of a fault. Instead, process variables containing steady-state fault signatures are inferred. An additional "causal reasoning" step must be performed that takes into consideration the causal nature of the monitored process, e.g, qualitative modeling of relations between different components of a system, to determine the root cause of fault-contaminated process variables. The presented method makes qualitative diagnosis more effective, since the reduction of fault smearing ensures that more precise qualitative information is provided.

B.5 Conclusion

This study introduces the combined application of a sparsity constraint and a pruning strategy to produce a sparse AE with the purpose of diagnosing a sensor fault occurring in the TTP. The obtained AE lead to the discovery of process knowledge, specifically the relationships among process variables. The solution demonstrated

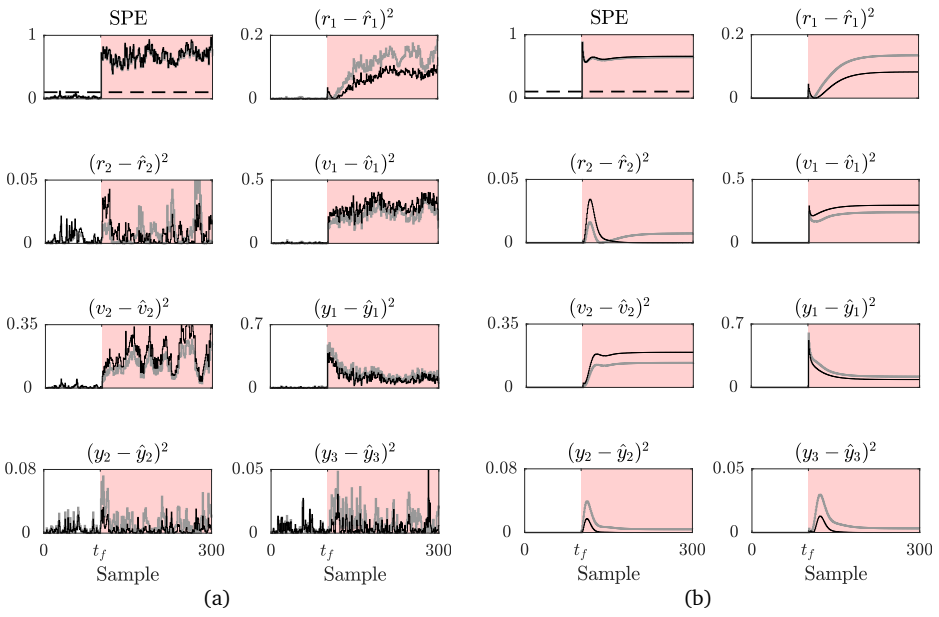


Figure B.9: Magnitudes of contributions to the SPE from AE_1 (grey) and AE_2 (black) via (a) stochastic simulations for $\mathbf{X}_f$ and (b) deterministic simulations for $\mathbf{X}_f$. Dashed line in SPE plot indicates the control limit obtained from $\mathbf{X}_v$.

that a sparse AE, which inherently has fewer parameters than a fully connected AE, suffered little in its validation performance.

The results show that the proposed method improved the performance of fault contribution plots; process variables unaffected by the fault produced significant less contributions due a reduction of fault smearing. The results also demonstrated that variables carrying no fault signatures, but were strongly correlated with the faulty variables, observed reduced contributions. Finally, variables that contained fault signatures produced larger contributions, providing further fault isolation capabilities.

Acknowledgments

The authors would like to acknowledge the support of the Danish Hydrocarbon Research and Technology Center(DHRTC) at the Technical University of Denmark.

Paper C

Unsupervised Isolation of Abnormal Process Variables Using Sparse Autoencoders

Ásgeir Daniel Hallgrímsson[1,*], Hans Henrik Niemann[1], Morten Lind[1]

[1]Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

Abstract:
Statistical approaches to fault diagnosis formulated on historical process data are well-suited for complex processes where detailed process knowledge is missing. However, methods for isolating abnormal process variables are prone to produce ambiguous results because they are established on nominal data. Literature suggests that diagnostic models are to include labeled abnormal data to improve fault isolation, yet inconclusive results may remain for previously unseen faults. This paper presents a method that isolates abnormal variables with an autoencoder (AE), a type of neural network that performs latent projection, without requiring prior knowledge of faults. The AE is trained on nominal process data with an additional sparsity constraint to produce a sparse network. The network is then probed to extract information regarding the correlation between process variables. Movements in the AEs residual space are interrogated alongside the acquired knowledge to isolate the variables that explain the observed movements. The method is demonstrated with a simulation of a nonlinear triple tank process, and is shown to isolate both simple and complex faults.

[*]Corresponding author. E-mail: asdah@elektro.dtu.dk

C.1 Introduction

Process operators are regularly confronted with the problem of proposing an appropriate cause to an abnormal event. In most scenarios, operators diagnose a plant fault by isolating abnormal changes in process variable signals. A probable cause is then assigned given a series of changes. As process plants become larger and more complex, complete reliance on operators for signal evaluation becomes more difficult. An increasing number of observable process variables will lead to information overload, slowing down analysis and risking incorrect diagnosis. With recent advances in data storage technologies, industry is more inclined to archive historical process data [62]. The growing availability of large sets of process data, coupled with increasing process complexity, has lead to an increase in research on statistical monitoring methods formulated to rely exclusively on data and not on process knowledge.

Data-driven process monitoring can be divided into two different approaches, namely, statistical fault classification and feature extraction. Fault classification is the problem of identifying to which of a set of faults a new observation belongs. Unfortunately, developing an effective classifier requires an abundant number of training observations for every possible fault. Obtaining sufficient training data proves difficult when faults, regardless of their severity, rarely occur. Feature extraction is the process of deriving numerical quantities intended to be informative about a data set. It is applied to process monitoring by comparing the features of new observations with features of a reference data set describing the nominal behavior of a plant. A fault is detected when the disparity, usually represented by a monitoring statistic, exceeds a certain threshold. Since it is often the case that historical process data contains disproportionately more samples explaining nominal plant behavior, monitoring based on feature extraction is generally favored over fault-classification.

The focus of this paper is latent projection (LP) - a method related to feature extraction. The method reduces the dimension of the original process variable space to produce features that retain information in the original data. In practice, LP uncovers the nominal correlation structure of process variables as correlated variables are summarized with a single principal variable. An LP model detects fault-induced abnormal changes in the correlation structure among process variable.

Venkatasubramanian et al. [133] propose that a successful diagnostic system is a hybrid of three diagnostic components: (a) a data-driven method for quick detection; (b) a trend-based method for assessing abnormal changes in process variables; and (c) an expert system that proposes a probable cause given the result from trend analysis. In the context of LP, much of the available literature addresses the first

diagnostic component. Within the class of linear methods, Principal Component Analysis (PCA) and Parial Least Squares (PLS) have been successfully applied to linear systems where process data follows the assumption of normality [91], [63], [90]. For nonlinear systems where the normality assumption is not met, independent component analysis, kernel PCA, and neural networks demonstrate superior performance [84], [82], [45]. Ku et al. [75] propose dynamic LP, where the process variable vector is extended with past samples to include dynamic behavior in the LP model. An overview of these methods is provided in [49].

Component wise residual analysis in the form of contribution plots are the primarily used diagnostic tool for assessing abnormal trends in process variables [95], [138]. The plots indicate the contribution of each variable to the monitoring statistic. If the statistic exceeds its control limit, the variables exhibiting the largest contributions are investigated. However, LP produces a fault smearing effect wherein signal characteristics of abnormal variables smear onto nominal variables [3]. Identifying a probable cause becomes difficult since the results from trend analysis are ambiguous. Yoon and MacGregor [148] propose a workaround where normalized contributions are compared with previously diagnosed contributions in a fault library to isolate the faulty variables. However, the method only applies to abnormal events that have occurred before, and thus fault smearing remains an issue for unfamiliar faults.

The objective of this paper is to develop a method that can isolate abnormal variables with an LP model without requiring prior knowledge of faults. It is assumed that the process being monitored is nonlinear and that its variables do not obey the assumption of normality. Autoencoders, a type of neural network designed for LP, are chosen since neural networks are capable of fitting any nonlinear function [30]. However, it is the author's opinion that the proposed method scales onto other LP methods as well. The AE model is optimized with an additional sparsity constraint to produce a sparse network, permitting one to probe into it to extract information regarding the interconnecting structure between process variables. Once a fault is detected, the LP model is interrogated to determine the changes in the process variables that would have produced the obtained contribution plots.

The organization of this paper is as follows. Section 2 reviews the method of LP in the context of PCA and AEs. Section 3 describes how AEs optimized with a sparsity constraint can expose process variable structure. Process monitoring with AEs and the isolation of abnormal process variables is discussed in section 4. Section 5 presents the results from diagnosing two different faults occurring in a triple tank process. The last two sections provide a discussion and conclusion, respectively, of the results.

C.2 Latent Projection

Latent projection, also known as dimensionality reduction, is a numerical technique that transforms high-dimensional data to a smaller set of latent, principal variables that retain essential information about the original data. The technique sets a compromise between the degree of dimensionality reduction and loss of information. Latent projection methods have seen increased application in process monitoring as large processes with many observable variables can be monitored with a smaller number of principal variables. Let $\mathbf{X} \in \mathbb{R}^{m \times n}$ represent a reference data matrix consisting of n standardized observations of an $m \times 1$ vector of process variables $\mathbf{x}$. An optimal mapping to the latent space $\mathbf{Z} \in \mathbb{R}^{q \times n}$ is sought in the form:

$$\mathbf{Z} = \underline{\mathscr{E}}(\mathbf{X}) \tag{C.1}$$

where $\underline{\mathscr{E}}$ is a vector function, composed of q individual functions; $\underline{\mathscr{E}} = [\mathscr{E}_1, \mathscr{E}_2, \ldots, \mathscr{E}_q]^\intercal$ such that if $\mathbf{z}_i$ represents the ith row vector of $\mathbf{Z}$,

$$\mathbf{z}_i = \mathscr{E}_i(\mathbf{X}) \tag{C.2}$$

The inverse transformation that reconstructs the original dimensionality of the data is implemented by a second vector function $\underline{\mathscr{D}} = [\mathscr{D}_1, \mathscr{D}_2, \ldots, \mathscr{D}_m]^\intercal$:

$$\hat{\mathbf{X}} = \underline{\mathscr{D}}(\mathbf{Z}) \tag{C.3}$$

The vector functions $\underline{\mathscr{E}}$ and $\underline{\mathscr{D}}$ are selected to minimize the loss of information represented by the average squared prediction error (SPE):

$$\mathscr{L}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{n} \left|\left| \mathbf{X} - \hat{\mathbf{X}} \right|\right|^2 \tag{C.4}$$

C.2.1 Principal Component Analysis

Within the class of linear methods, the transformation with the least information loss is obtained via PCA [51], [104]. PCA is a procedure that performs an orthogonal transformation on $\mathbf{X}$ to produce the scores matrix $\mathbf{T} \in \mathbb{R}^{m \times n}$:

$$\mathbf{T} = \mathbf{P}^\intercal \mathbf{X} \tag{C.5}$$

where $\mathbf{P} \in \mathbb{R}^{m \times m}$ is the principal component loading matrix. The score vector $\mathbf{t}_i$ is the ith row vector of $\mathbf{T}$. The procedure has the following properties: (a) the loading matrix $\mathbf{P}$ is orthonormal, i.e., $\mathbf{P}\mathbf{P}^\intercal = \mathbf{P}\mathbf{P}^{-1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, and thus $\mathbf{X}$ is reproducible via $\mathbf{X} = \mathbf{P}\mathbf{T}$; and (b) the first principal component score $\mathbf{t}_1 = \mathbf{p}_1^\intercal \mathbf{X}$ has maximum variance, the second principal component score $\mathbf{t}_2 = \mathbf{p}_2^\intercal \mathbf{X}$ has the second greatest variance, with additional scores up to m similarly defined. The

loading matrix $\mathbf{P}$ is obtained by solving for the eigenvectors of the covariance matrix $\mathbf{\Sigma} = \mathbf{X}^\intercal\mathbf{X}/(n-1)$:

$$\mathbf{\Sigma} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^\intercal \tag{C.6}$$

where $\mathbf{\Lambda}$ is a non-negative real diagonal $m \times m$ matrix whose diagonal elements are the corresponding eigenvalues. The diagonal entries λ_i of $\mathbf{\Lambda}$ are the variances of the principal component scores t_i.

Latent projection is achieved by identifying q principal components that explain most of the predictable variation in the data. The remaining $m-q$ principal components are associated with common cause variations. For that purpose, the loading matrix is partitioned as follows:

$$\mathbf{P} = \begin{bmatrix} \hat{\mathbf{P}} & \tilde{\mathbf{P}} \end{bmatrix}, \quad \hat{\mathbf{P}} \in \mathbb{R}^{m\times q}, \quad \tilde{\mathbf{P}} \in \mathbb{R}^{m\times(m-q)} \tag{C.7}$$

Matrix $\mathbf{X}$ is then decomposed into the reconstructed data matrix $\hat{\mathbf{X}}$ and residual data matrix $\tilde{\mathbf{X}}$:

$$\begin{aligned} \mathbf{X} &= \hat{\mathbf{X}} + \tilde{\mathbf{X}} \\ &= \hat{\mathbf{P}}\hat{\mathbf{T}} + \tilde{\mathbf{P}}\tilde{\mathbf{T}} \\ &= \hat{\mathbf{P}}\hat{\mathbf{P}}^\intercal\mathbf{X} + \tilde{\mathbf{P}}\tilde{\mathbf{P}}^\intercal\mathbf{X} \end{aligned} \tag{C.8}$$

Analogies can be drawn between the mapping functions in Eqs. (C.1) and (C.3) and principal component decomposition in Eq. (C.8), namely: (a) The latent space $\mathbf{Z} = \underset{\sim}{\mathscr{E}}(\mathbf{X})$ is equivalent to the score matrix $\hat{\mathbf{T}} = \hat{\mathbf{P}}^\intercal\mathbf{X}$, thus $\underset{\sim}{\mathscr{E}}(\mathbf{X}) \triangleq \hat{\mathbf{P}}^\intercal\mathbf{X}$; and (b) The reconstruction space $\hat{\mathbf{X}} = \underset{\sim}{\mathscr{D}}(\mathbf{Z})$ is equivalent to $\hat{\mathbf{X}} = \hat{\mathbf{P}}\hat{\mathbf{T}}$, thus $\underset{\sim}{\mathscr{D}}(\mathbf{Z}) \triangleq \hat{\mathbf{P}}\hat{\mathbf{T}}$.

C.2.2 Autoencoders

PCA is unsuitable for dimensionality reduction of process variables that exhibit nonlinear correlations. Doing so resembles to fitting a first order linear regression model to a polynomial data set; the model retains the *average* trend of the data but fails in describing its nonlinearity. Similarly, loading vectors in $\mathbf{P}$ will describe the *average* linear correlations present in nonlinear data.

The existence of nonlinearly correlated variables requires for extensions of PCA that perform nonlinear mappings between the original and reduced dimension spaces. Such models describe the data with better accuracy than PCA for the same number of latent variables. Nonlinear latent projection is achieved with AEs - a type of artificial neural network designed for dimensionality reduction. The simplest form of an AE is a multilayered, feedforward, non-recurrent neural network, illustrated in Fig. C.1. The vector functions $\underset{\sim}{\mathscr{E}}$ and $\underset{\sim}{\mathscr{D}}$ are generated with a nonlinear basis function approach. The network is composed of several vectors of nodes, known as

network layers. With the exception of the input layer, each layer is a component wise nonlinear function of a linear transformation of its previous layer. The encoder maps the input $\mathbf{X} \in \mathbb{R}^{m \times n}$ to the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$:

$$\mathbf{E}_i = \begin{cases} \sigma_1^e \left(\mathbf{W}_1^e \mathbf{X} + \mathbf{b}_1^e \right), & \text{for } i = 1 \\ \sigma_i^e \left(\mathbf{W}_i^e \mathbf{E}_{i-1} + \mathbf{b}_i^e \right), & \text{else} \end{cases} \tag{C.9}$$

$$\mathbf{Z} = \sigma^z \left(\mathbf{W}^z \mathbf{E}_N + \mathbf{b}^z \right)$$

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer. The decoder maps the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$ to the input reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$:

$$\mathbf{D}_i = \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{Z} + \mathbf{b}_1^d \right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{D}_{j-1} + \mathbf{b}_j^d \right), & \text{else} \end{cases} \tag{C.10}$$

$$\hat{\mathbf{X}} = \sigma^{\hat{x}} \left(\mathbf{W}^{\hat{x}} \mathbf{D}_M + \mathbf{b}^{\hat{x}} \right)$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{x}}$, $\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer. The modifiable parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with respect to minimizing the loss function in Eq. (C.4) via stochastic gradient descent [99].

Fig. C.1 shows that the dimensions of the encoder and decoder layers are larger than the size of the input dimension. This permits the AE to generate complex,

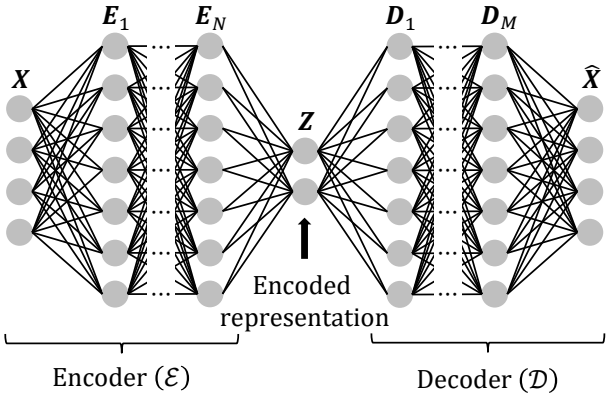


Figure C.1: Illustration of an autoencoder.

higher dimenional features before and after data compression [102]. Since the process data in $\mathbf{X}$ is usually mean-centered and real valued, the tangent hyperbolic activation function is used at the hidden layers of $\mathbf{E}_i$, $\mathbf{Z}$, and $\mathbf{D}_j$, and the linear identity function is used at the output layer of $\hat{\mathbf{X}}$.

C.3 Discovery of Process Knowledge

It is known that the performance of neural networks is largely determined by their level of complexity, with maximum modeling accuracy achieved by increasing the number and size of network layers. However, proceeding in such a direction has a tendency of overfitting a model to the training data that then performs poorly on validation and test data. This is undesirable in statistical fault diagnosis since due to the disparity between the training and test set; both are sampled from the same process, but the former describes its nominal behavior while the latter may contain samples describing abnormal behavior. An overfitted AE corresponding too close to nominal data does not necessarily capture the overall behavior of the process and may perform poorly at diagnosing abnormal data.

It is favourable to use a network with minimum complexity that retains acceptable levels of accuracy in order to prevent overfitting. A simple method is to have an algorithm by which a large network (whose complexity is greater than is justified by the data) is progressively trimmed down in size by identifying and deleting redundant weights in the network. Redundant weights are defined as those that are small in absolute magnitude. The optimization function is augmented with the naïve elastic net penalty to promote for a small number of high-importance weight connections, thereby shrinking the remaining weights to zero. The penalty is a regularized regression method that linearly combines the L_1 and L_2 weight decay penalties of the LASSO and ridge methods [154]. The dual objective function becomes:

$$\mathscr{L}(\mathbf{X},\hat{\mathbf{X}},\mathbf{W}) = \frac{1}{n}\left|\left|\mathbf{X}-\hat{\mathbf{X}}\right|\right|^2 + \lambda_1 ||\mathbf{W}||_1 + \lambda_2 ||\mathbf{W}||_2^2 \tag{C.11}$$

where $\mathbf{W}$ is the collection of weights in the AE and λ_1 and λ_2 control the importance of the LASSO and ridge regressions, respectively.

Zhu and Gupta [153] introduce a magnitude-based weight pruning algorithm that periodically removes redundant weights. Trimming occurs according to a pruning function that sets the current sparsity percentage, i.e., the number of zero magnitude weights divided by the total number of weights, of a network:

$$s_t = s_f + (s_i - s_f)\left(1 - \frac{t-t_0}{n\Delta t}\right)^3 \text{ for } t \in \{t_0, t_0+\Delta t, \ldots, t_0+n\Delta t\} \tag{C.12}$$

The algorithm begins by setting the initial sparsity of the network to s_i after t_0 training steps. The network is then gradually pruned every Δt steps to increase its sparsity while permitting it to recover from any pruning-induced loss in accuracy. The order of the pruning function enforces rapid pruning in the initial phase when redundant weights are abundant before slowing down once fewer connections remain. Weights are continuously pruned over n sparsity updates until the final sparsity value s_f is reached or if modeling accuracy deteriorates significantly.

Minimization of the weight-redundancy inducing optimization function in Eq. (C.11) in combination with the pruning function in Eq. (C.12) produces a sparse network. Consequently, the establishment of a small number of high-importance connections encourages an interpretable variable grouping effect. It is then possible to probe into a sparse AE to extract information from it regarding the inherent relationships between process variables. For example, Gao et al. [39] demonstrate the discovery of feedback control loops and downstream process variable relations by constructing a sparse PCA model of the Tennessee Eastman process. Similarly, Bhat and McAvoy [11] used a pruning strategy to discover correlations between process variables existing in the context of time series prediction.

The pruning strategy is used to uncover the relationships between the process variables of the system illustrated in Fig. C.2. In this example, the open loop control signal v_1 and measurement signal y_1 are correlated with one another through open loop subsystem A. Similarly, the open loop reference signal r_1, closed loop control signal v_2, and measurement signal y_2 are correlated with one another through closed loop subsystem B.

The system is excited with random variations in v_1 and r_1, and the resulting signals are sampled to generate a reference data set. The result of training an AE to project the variables down to two dimensions along with the pruning strategy

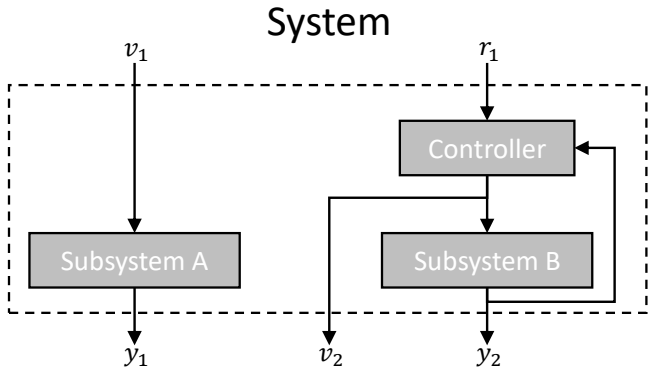


Figure C.2: Example system.

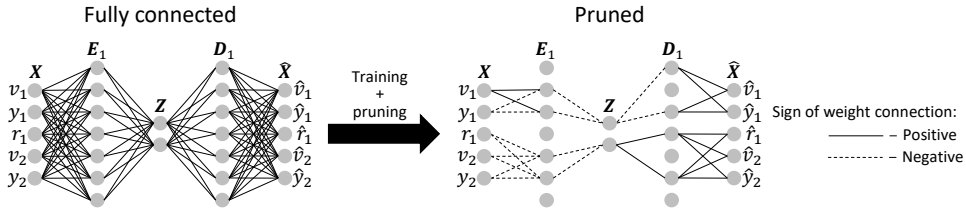


Figure C.3: Included are the magnitude of the weights within the network.

is illustrated in Fig. C.3. The structure of the pruned AE exposes the relationships between the variables; v_1 and y_1 are projected to one of the available latent variables whilst r_1, v_2, and y_2 are projected to the other. This indicates that the interaction between each group of variables can be summarized by a single variable. In effect, the AE learns the underlying structure of the process system. Furthermore, each group of variables is subsequently reconstructed with their corresponding latent variable and remain independent from one another.

C.4 Online process monitoring and fault contribution analysis

Online process monitoring consists of referring new variable samples against an "in-control" AE trained with historical data collected when only common cause variation was present in the process. New observations are reconstructed by propogating them through the AE to obtain the residuals $\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new}$. Previously unseen changes in signal characteristics caused by an abnormal event can be detected by computing the SPE (otherwise known as the Q monitoring statistic) of the residuals [91]:

$$SPE = \sum_{i=1}^{m} (x_{new,i} - \hat{x}_{new,i})^2 \tag{C.13}$$

Assuming that the SPE follows a Chi-squared distribution, a control limit is computed with the approximate value [12]:

$$CL_{SPE_{AE}} = \frac{\bar{\sigma}^2}{2\bar{\mu}} \chi^2_{(2\bar{\mu}^2/\bar{\sigma}^2,\alpha)} \tag{C.14}$$

where $\bar{\mu}$ and $\bar{\sigma}$ are the sample mean and sample standard deviation of the SPE and α is the false alarm rate, respectively. An abnormal event is said to have occurred if the SPE statistic crosses the control limit, signifying that the trained AE no longer applies to the new observations.

The common approach to contribution analysis is determining the individual contributions of process variable i to the SPE in Eq. (C.13) [95]:

$$C_i = (x_{new,i} - \hat{x}_{new,i})^2 \tag{C.15}$$

The analysis is magnitude-based, and variables showing large contributions are concluded to no longer be consistent with normal operating conditions. Operators can focus their attention on few variables in the plant and apply their process knowledge to infer potential causes.

The problem with magnitude-based contribution analysis is that a fault smearing effect will generate contributions from variables that are operating within normal conditions. This is an inherent property of latent projection methods; the compression to a smaller latent space and subsequent expansion to the original space enables faulty and non-faulty variables to interact. The fault-smearing effect hampers fault diagnosis since non-faulty variables are highlighted and faulty variables are concealed. The effect primarily occurs within groups of correlated process variables that are entangled in control loops and coupled across different process units. However, erroneous contributions are also observable from variables decoupled from a faulty subsystem if couplings exist within the latent projection model. These couplings are removed by applying the pruning strategy in the previous section or by explicitly building separate LP models for each decoupled system as in [90]. However, contributions from normal variables coupled with abnormal variables are unavoidable [127].

The contribution analysis proposed in this paper is the examination of linear prediction errors, i.e. the unsquared contributions of Eq. (C.15):

$$C_i = x_{new,i} - \hat{x}_{new,i} \tag{C.16}$$

In addition to providing a magnitude for the relative contribution of each variable to the SPE, the sign of linear contributions indicate the direction the reconstruction $\hat{x}_{new,i}$ has shifted from $x_{new,i}$. For example, if C_i is positive then $\hat{x}_{new,i}$ has shifted negatively from $x_{new,i}$. Previous reports imply that linear contributions can determine whether a process variable is too low or too high, but its application is explored fully in this paper [90], [138]. More specifically, the trained AE model is interrogated with observed output drifts in $\hat{\mathbf{x}}_{new}$ to determine changes in signal characteristics of original process variables at the input $\mathbf{x}_{new}$.

Process faults typically cause abnormal drifts in a subset of process variables. Identifying the direction of these drifts permits determining a probable cause. Since abnormal variables form a subset of $\mathbf{x}_{new}$, propagating them through the pruned AE will affect a subset of $\hat{\mathbf{x}}_{new}$. A list of observed shifts in $\hat{\mathbf{x}}_{new}$ is obtained by computing

the linear contributions in Eq. (C.16) for each $x_{new,i}$. Drifts in $\mathbf{x}_{new}$ that explain the observed shifts are determined by interrogating the list of observed shifts in $\hat{\mathbf{x}}_{new}$ with the underlying AE; each entry in the list is propagated backwards through the AE, forming a connection between potential shifts at its input that explain the observed output shift in question. Doing this for every observed shift produces a list of *causal paths*. Causal paths that explain multiple observed output shifts with the same derived input shifts are pooled together into *reasoning paths*. Finally, a list of possible explanations for the detected fault is obtained by invalidating potential reasoning paths by applying *a priori* knowledge about process variables that cannot contain faulty signatures. For example, the signal characteristics of open loop reference and open loop control signals cannot be influenced by a fault; they are external signals whose values are uninfluenced by the process plant.

The diagnostic strategy is demonstrated on the system depicted in Fig. C.2. Fig. C.4 shows the linear contribution plots produced by the AE in Fig. C.3 when an abnormal positive shift occurs in v_2 at sample t_f. The abnormality is detected as the SPE crosses its threshold. From the linear contribution plots $x_i - \hat{x}_i$, the most noticeable result is that no contributions are observed for $\hat{v}_1$ and $\hat{y}_1$ while the remaining variables display noticeable contributions. This is due to the pruning strategy; the fault carried by v_2 cannot influence $\hat{v}_1$ and $\hat{y}_1$ since there is no connection established by the AE (Fig. C.3). If, on the other hand, the AE remained unpruned and a connection existed, the fault would smear from v_2 onto $\hat{v}_1$ and $\hat{y}_1$, regardless if v_2 is explicitly decoupled from v_1 and y_1 (Fig. C.2). This would generate a contribution for $\hat{v}_1$ and $\hat{y}_1$, hampering variable isolation by providing too much information. The pruning strategy guarantees that faulty variable v_2 smears only onto $\hat{r}_1$, $\hat{v}_2$, and $\hat{y}_2$, at least making it easier to know in which control loop the fault resides [46].

The contribution plots demonstrate the smearing of the fault residing in v_2 onto r_1 and y_2; even though r_1 and y_1 are known to be nominal, their contribution plots indicate otherwise. Although the plots suggest that the fault is contained within the

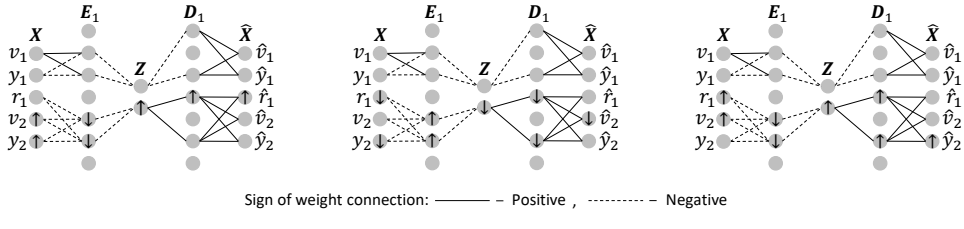


Figure C.5: Illustration of causal paths.

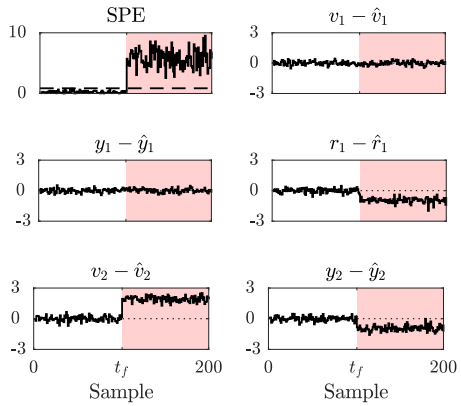


Figure C.4: Contribution plots. Fault occurs at sample t_f.

variables of subsystem B, it is impossible to know which variables are inconsistent with normal operating conditions by relying on the plots alone. Despite v_2 providing the largest contribution, one cannot conclude that it is the variable that explains the fault; for example, r_1, v_2, and y_2 could all be abnormal with v_2 being the most abnormal. An expert system would be required to interpret the observations which could require additional effort if this is the first recorded occurrence of the fault.

The concealed abnormal variable v_2 is unveiled by interrogating the causal paths of the observed shifts in the reconstructed variables. The contribution plots indicate a positive shift in $\hat{r}_1$, a negative shift in $\hat{v}_2$, and a positive shift in $\hat{y}_2$. Causal paths are derived by individually backpropagating the shifts through the sparse network in Fig. C.3 whilst considering the sign of the weight connections and the monotonicity of the tangent hyperbolic activation function. The causal paths are illustrated in Fig C.5 and summarized in Table C.1. The crosses along the diagonal of the original variables indicate the presence of a confliction rule that is discussed later; for now, the rule is that a valid causal path cannot be derived from x_i to $\hat{x}_j$ if $i = j$.

The causal paths are pooled into valid reasoning paths by examining compliments and conflicts between the paths. Table C.1 reveals that: (a) causal paths 1 and 2 conflict with another with respect to y_2, i.e., y_2 cannot simultaneously experience a positive and negative shift; (b) observations 2 and 3 conflict with another with respect to r_1; and (c) causal paths 1 and 3 agree with another with respect to v_2. The final results are compiled in Table C.2 and are: (a) a combined positive shift in $\hat{r}_1$ and a positive shift in $\hat{y}_2$ could only have been induced by positive shift in v_2; and (b) a negative shift in $\hat{v}_2$ could only have been induced by combined negative shift in r_1 and a negative shift in y_2. An illustration of the derived reasoning paths is given in

Table C.1: Causal paths.

Causal path	$\hat{r}_1$	$\hat{v}_2$	$\hat{y}_2$	r_1	v_2	y_2
1	↑			×	↑	↑
2		↓		↓	×	↓
3			↑	↑	↑	×

Table C.2: Reasoning paths.

Reasoning path	$\hat{r}_1$	$\hat{v}_2$	$\hat{y}_2$	r_1	v_2	y_2
1	↑		↑		↑	
2		↓		↓		↓

Fig. C.6. The correct diagnosis, i.e., that v_2 experienced an abnormal positive drift, is obtained by considering that it is known *a priori* that r_1 is an open loop signal and thus unable to carry any indication of a fault. This leaves the latter results as the only viable and correct conclusion; that the fault detected via statistical means could only have been induced by a positive drift in the signal of v_2.

The aforementioned confliction rule, namely, a causal path from x_i to $\hat{x}_j$ is invalid if $i = j$, is illustrated by reference to Fig. C.7. Two process variables are considered for ease of illustration. It is known *a priori* that x_1 is an open loop signal, and thus its signal characteristic is insensitive to a fault. x_1 and x_2 are positively correlated

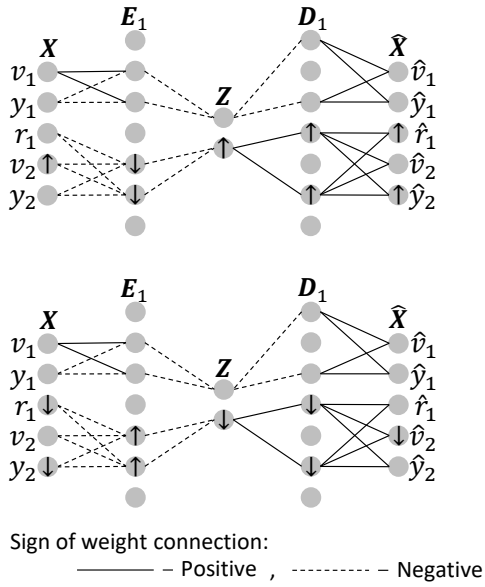


Figure C.6: Illustration of reasoning paths.

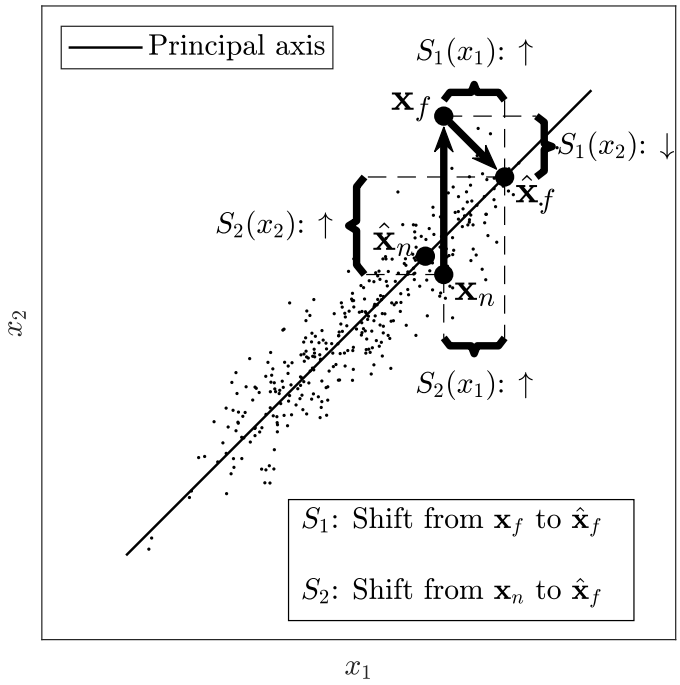


Figure C.7: Monitoring of two variables.

when the process exists in a state of statistical control, as depicted by the principal axis of variance. The process is monitored via the SPE between the original variables $\mathbf{x} = (x_1, x_2)$ and the reconstructed variables $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2)$ residing on the principal axis. The variable pair $\mathbf{x}_n$ is the last indication that the process is in control. The process unexpectedly experiences a fault that causes a positive shift in x_2 by a magnitude of f, driving the process to $\mathbf{x}_f$. Latent projection along the principal axis generates the reconstruction $\hat{\mathbf{x}}_f$ and, as a result of fault smearing, generates the positive shift $S_1(x_1)$ in $\hat{x}_1$ and negative shift $S_1(x_2)$ in $\hat{x}_2$. Fig. C.8 illustrates the causal paths derived from backpropagating the observed reconstruction shifts. Here, the confliction rule is neglected, e.g., a viable causal path for a negative shift in $\hat{x}_2$ is a negative shift in x_2. Based on the prior knowledge that x_1 cannot explain the presence of the fault,

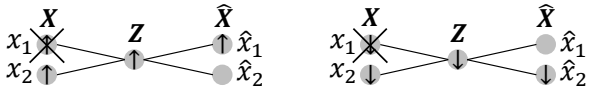


Figure C.8: Causal paths obtained from not abiding to the confliction rule. Crosses indicate path invalidation due to *a priori* knowledge. Weight connections are positive.

one is left with two plausible explanations: the fault was caused either by a positive or negative shift in x_2. This result is ambiguous as the two explanations contradict each other.

The contradiction is resolved by considering the displacement between the reconstruction $\hat{\mathbf{x}}_f$ and the last indicated nominal pair $\mathbf{x}_n$. What is actually being interrogated for when backpropagating the observed reconstruction shifts is the effect an abnormal shift in $\mathbf{x}_n$ will have on its displacement from its reconstruction $\hat{\mathbf{x}}_n$. In other words, one desires the component wise shifts in $\mathbf{x}_n$ that explains the shift of $\hat{\mathbf{x}}_n$ to $\hat{\mathbf{x}}_f$. Hence the causal path in the right causal path of Fig. C.8 is logical; a negative shift in $\mathbf{x}_n$ will cause $\hat{\mathbf{x}}_n$ to move towards the bottom-left end of the principal axis, coinciding to a negative shift in $\hat{x}_2$.

The ambiguous causal explanation obtained for x_2 stems from it being the faulty variable. The correct explanation, i.e., x_2 has increased, is derived if the true shift $S_2(x_2)$ is backpropogated through the AE instead of the observed shift $S_1(x_2)$. Unfortunately, $S_2(x_1)$, which relies on the sample prior to the fault, is unattainable since contributions in Eq. (C.16) consider new samples. However, Fig. C.7 demonstrates that latent projection of $\mathbf{x}_f$ produces correct observed shifts for $\hat{x}_j$ from a faulty variable x_i as long as $i \neq j$; the observed shift $S_1(x_1)$ corresponds, both in direction and absolute magnitude, to the true shift $S_2(x_1)$. This is the incentive for the confliction rule, i.e., that the faulty variable x_i will generate a correct shift for $\hat{x}_j$ if $i \neq j$ and an incorrect shift if $i = j$. Enforcing the confliction rule before applying *a priori* knowledge produces the causal paths in Fig. C.9. After applying the *a priori* knowledge about x_1 the correct conclusion is derived, namely that the observed positive shift in $\hat{x}_1$ could only have been caused by a positive shift in x_2.

C.5 Case Study: The Triple Tank Process

The methods presented in the previous sections are applied on the triple tank process (TTP) in this section. The TTP - a multivariate, nonlinear process - is a variant of the quadruple tank process [64]. A schematic drawing of the TTP is given in Fig. C.10. The liquid supplying the upper tanks is transported from a large sump by the means of two gear pumps. Liquid flows out from the bottom of each

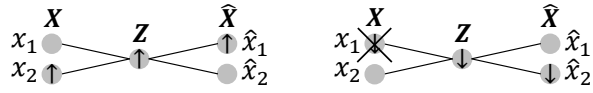


Figure C.9: Causal paths obtained from abiding to the confliction rule. Crosses indicate path invalidation due to *a priori* knowledge. Weight connections are positive.

tank, with the liquid from the upper right tank first supplying the lower tank before sequentially flowing returning to the sump. The level of each tank is monitored with a voltage-based level measurement device. The objective is to control the liquid levels in the upper left and lower right tanks. The set of nonlinear differential equations describing the evolution of the liquid level of each tank are derived by applying mass balances and Bernouilli's law:

$$\begin{aligned}
\frac{dh_1}{dt} &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{1}{2}\frac{k_1}{A_1}v_1(1+\eta_1) \\
\frac{dh_2}{dt} &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{1}{2}\frac{k_1}{A_2}v_1(1+\eta_1) + \frac{k_2}{A_2}v_2(1+\eta_2) \\
\frac{dh_3}{dt} &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{a_2}{A_3}\sqrt{2gh_2}
\end{aligned} \tag{C.17}$$

where A_i is the cross-section of tank i and a_i is the cross-section of its outlet hole. The liquid level of tank i is h_i and g is acceleration due to gravity. The voltage applied to pump i is v_i and the corresponding flow is $k_i v_i(1+\eta_i)$, where $\eta_i \in \mathbb{R}$ is zero mean Gaussian noise emitted from pump i. The system is measured and actuated discretely with a sample time of T_s. The measured level signals at sample k are:

$$y_i[k] = k_c h_i[k] + w_i[k] \tag{C.18}$$

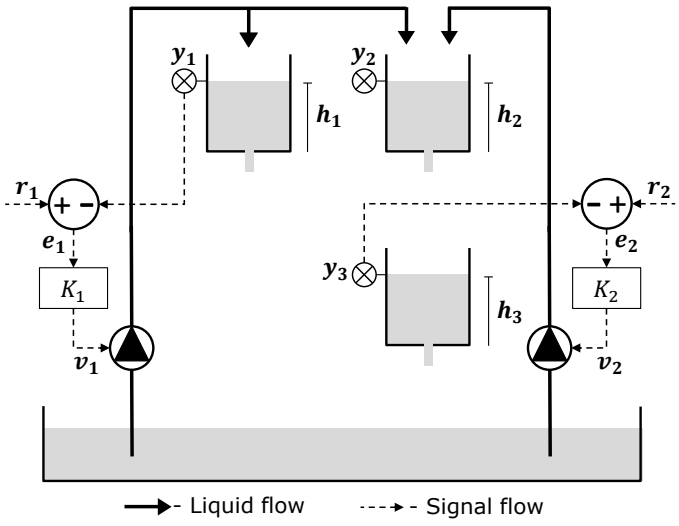


Figure C.10: A schematic of the TTP illustrating the connectivity of the tanks and location of the pumps, dual valves, and the level measurement devices. Included are the decentralized feedback loops.

where $w_i[k] \in \mathbb{R}$ is zero mean measurement noise with Gaussian distribution for level signal i. For decentralized control, the error terms are:

$$\begin{aligned} e_1[k] &= r_1[k] - y_1[k] \\ e_2[k] &= r_2[k] - y_3[k] \end{aligned} \tag{C.19}$$

where $r_1[k]$ and $r_2[k]$ are reference signals for level signals $y_1[k]$ and $y_3[k]$, respectively. The error terms are minimized by a discrete PI controller. The closed loop control laws for the process inputs are:

$$K_i: \; v_i[k] = K_P e_i[k] + K_I \sum_{j=1}^{k} e_i[j] T_s \tag{C.20}$$

Here K_P and K_I denote the proportional and integral gains, respectively, of the PI controller. Monte Carlo simulations were performed on the TTP in order to generate data sets that exhibited nonlinear correlations between its process variables. The data sets were used to train, validate, and test an AE model that monitored the process. The uncertain parameters were the reference signals r_1 and r_2. Values for r_1 and r_2 were sampled from two independent uniform distributions. Process, controller, and noise parameters are listed Table C.3 [64].

Since probing a sparse AE may reveal the steady state interaction between process variables, it is relevant to determine them *a priori* for validation purposes. Fig. C.11 displays a time series plot from inducing independent step changes in the reference variables r_1 and r_2. The plots demonstrate that: (a) a step change in r_1 induces a dynamic change in the steady state values of y_1, v_1, and v_2, while the steady state values of y_2 and y_3 remain unaffected except for a dynamic change that dies out; and (b) a step change in r_2 induces a transient change in the steady state values of y_2, y_3, and v_2 whilst having no influence on y_1 and v_1. The correlation sets $C_1 = \{r_1, y_1, v_1, v_2\}$ and $C_2 = \{r_2, y_2, y_3, v_2\}$ are determined from the plots.

Table C.3: List of Parameters

Process param.		**Noise param.**	
A_1, A_3	28 cm^2	η_i	$\mathcal{N}(0, 0.1)$
A_2	32 cm^2	w_i	$\mathcal{N}(0, 0.0005)$
a_1, a_3	0.071 cm^2		
a_2	0.057 cm^2		
k_c	1 V/cm	**Controller param.**	
k_1	3.33 cm^3/Vs	T_s	10
k_2	3.35 cm^3/Vs	K_P	20
g	981 cm/s^2	K_I	0.25

C.5.1 AE Training and process discovery

The TTP is simulated with random step changes occurring every 200 time steps in reference signals r_1 and r_2. The training set $\mathbf{X}_t$ (consisting of 300,000 time steps) and the validation set $\mathbf{X}_v$ (consisting of 30,000 time steps) are sampled to train and validate an AE. Fig. C.12 illustrates the AE obtained from minimizing the loss function in Eq. (C.11) and following the aforementioned pruning strategy. The pruning parameters were $s_f = 0.9$, $s_i = 0.7$, $t_0 = 5000$, $\Delta t = 400$, and $n = 300$. The tangent hyperbolic function was applied at the hidden layers of the AE. The network is partitioned into two separate subnetworks in Fig. C.13 to clarify the interconnectivity of the AE; each subnetwork shows the propagation of the inputs to the reconstructions through a single latent variable in $\mathbf{Z}$. The subnetworks reveal the process variable relationships, with subnetwork A comprising variables of

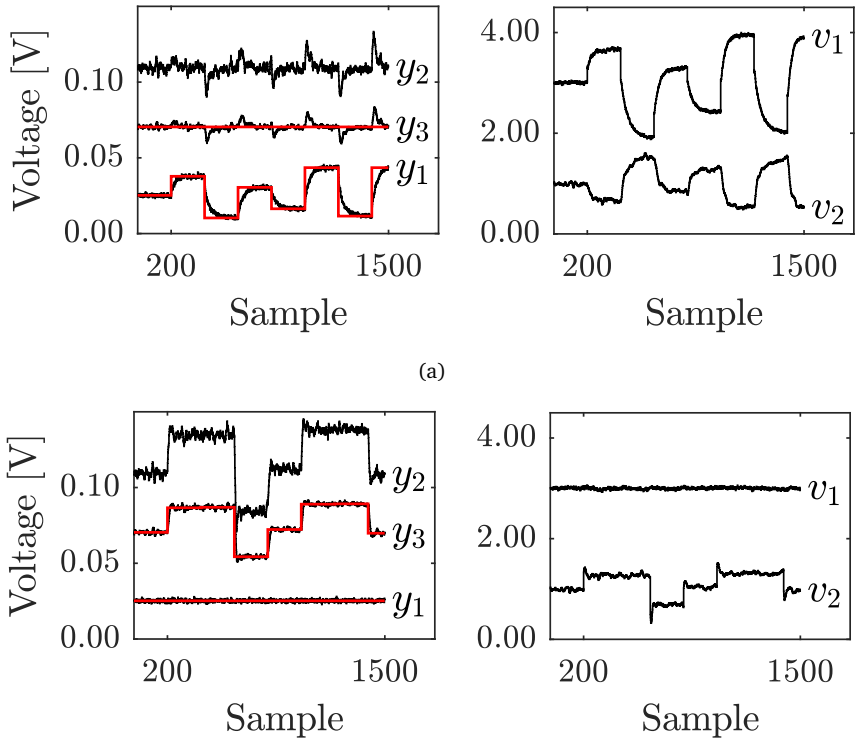


Figure C.11: Time series of simulated process variables where (a) r_2 is changed whilst r_1 is held fixed and (b) r_1 is changed whilst r_2 is held fixed. Red lines indicate the references for the measurement.

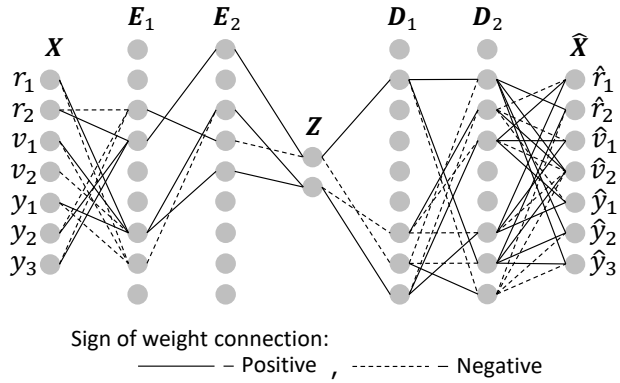


Figure C.12: Illustration of trained *AE*. Biases have been excluded from the illustration.

correlation set C_1 and subnetwork B comprising variables of set C_2 with the exception of v_2. These results indicate that, given the training parameters, information of variable v_2 is not required to perform an accurate reconstruction of $\hat{r}_2$, $\hat{y}_1$, and $\hat{y}_2$ but that information about variables r_2, y_2, and y_3 is required to perform an accurate reconstruction of $\hat{v}_2$. This has the property that changes in the signal characteristics of v_2 due to a fault will not smear to $\hat{r}_2$, $\hat{y}_1$, and $\hat{y}_2$, despite them being in the same control loop.

A monitoring control limit is determined from the SPE of validation set $\mathbf{X}_v$ and setting the false alarm rate $\alpha = 0.01$ in Eq. (C.14), implying that a false alarm occurs every 100 samples. Fig. C.14 displays the SPE from propagating the first 800 samples of $\mathbf{X}_v$ through the AE. The SPE increases immediately when a reference change occurs before returning underneath the control limit. The sharp increases occur because the AE, being a static LP model, only learns about the steady-state nature of the TTP. Reference changes induce temporary dynamics in the TTP which the AE has difficulty processing.

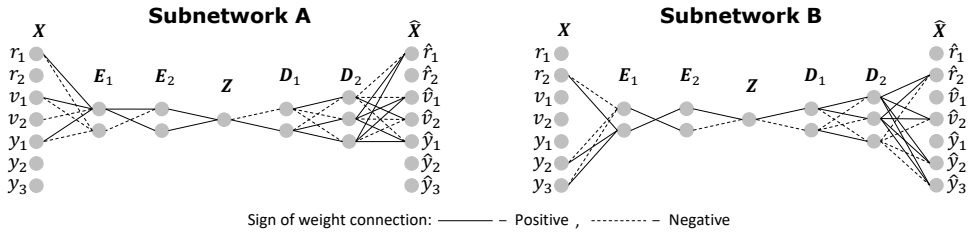


Figure C.13: Partitioning of trained AE, showing variable interaction. Biases have been excluded from the illustration.

C.5.2 Fault diagnosis

Two fault sets $\mathbf{X}_{f,1}$ and $\mathbf{X}_{f,2}$ were generated by simulating the TTP with a bias in sensors 2 and 1, respectively. Each bias was introduced after 100 time steps via an additive fault; $y_2[k] = k_c h_2[k] + w_2[k] + f_1$ for $\mathbf{X}_{f,1}$ and $y_1[k] = k_c h_1[k] + w_1[k] + f_2$ for $\mathbf{X}_{f,2}$. Like with $\mathbf{X}_t$ and $\mathbf{X}_v$, random step changes in r_1 and r_2 occurred every 200 time steps. The influence of faults on the signal characteristics of TTP variables are determined prior to diagnosis so that results from fault propagation analysis can be verified. Fig. C.15 displays the time series of the first 200 samples of $\mathbf{X}_{f,1}$, and $\mathbf{X}_{f,2}$. Table C.4 summarizes the shifts of each signal after a fault is introduced. Since the AE performs steady state analysis, the results expected from analysis of $\mathbf{X}_{f,1}$ and $\mathbf{X}_{f,2}$ are: (a) Fault f_1 induces a negative shift in y_3; and (b) fault f_2 induces a positive shift in v_1 and a negative shift in v_2. The faults also demonstrate different complexity [148]; f_1 is a *simple fault* that occurs at sensor 2 and its effect is not propagated into other variables, whereas f_2 is a *complex fault* that occurs at sensor 1 but its effects propagates to the two gear pumps.

Contribution plots from propagating $\mathbf{X}_{f,1}$ through the AE are displayed in Fig. C.16. The fault f_1 is detected after its onset at sample t_f when the SPE crosses its control limit. Simultaneously, the contribution plots indicate a negative shift in $\hat{r}_2$, a negative shift in $\hat{v}_2$, a positive shift in $\hat{y}_2$, and a negative shift in $\hat{y}_3$. These variables encompass subnetwork A in Fig. C.13. The causal paths derived from individually backpropagating each of the four observations through subnetwork A are summarized in Table C.5. Causal path 3 conflicts with causal paths 1, 2, and 4, since there is a dispute about the drifts of r_2 and y_3, and is thus considered a valid reasoning path. Causal paths 1, 2, and 4 compliment each other with respect to the drifts of r_2, y_2, and y_3 and are thus combined into a valid reasoning path. The derived reasoning paths are presented in Table C.6 and visualized in Fig. C.17. Given the *a priori* knowledge that the characteristics of open loop reference signal r_2 cannot be influenced by a fault, the implications of reasoning path 1 , i.e., that the observed positive shift in $\hat{y}_2$ is caused by a positive drift in r_2 and y_3, is invalid. By

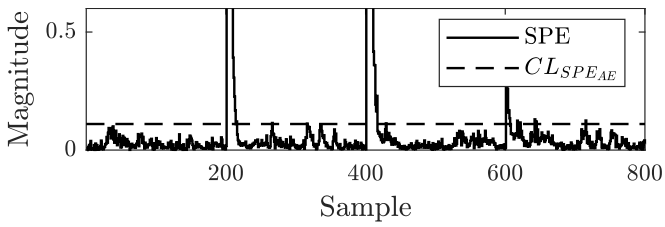


Figure C.14: SPE of validation set and the control limit.

the process of elimination, reasoning path 2 is the only viable explanation for the detected fault, i.e., that it could only have been induced by a negative process drift in y_2. This conclusion coincides with the observed steady-state influences of fault f_1 presented in Table C.4.

The same approach to diagnosis is applied to the contribution plots in Fig. C.18 obtained by propagating $\mathbf{X}_{f,2}$ through the AE. The fault is detected when the SPE crosses its control limit at the onset of fault f_2 at time t_f, and the individual linear contribution plots reveal a positive shift in $\hat{r}_1$, a negative shift in $\hat{v}_1$, a positive shift in $\hat{v}_2$, and a positive shift in $\hat{y}_1$. These variables are encompassed by subnetwork B (Fig. C.13). Causal paths derived by backpropagating each of the four observations through subnetwork B are presented in Table C.7. By considering causal path compliments and conflicts, two reasoning paths are derived, with the former comprising causal paths 1 and 4 and the latter paths 2 and 3. The reasoning paths are presented in Table C.8 and their propogation is visualized in Fig. C.19. Since it is known *a*

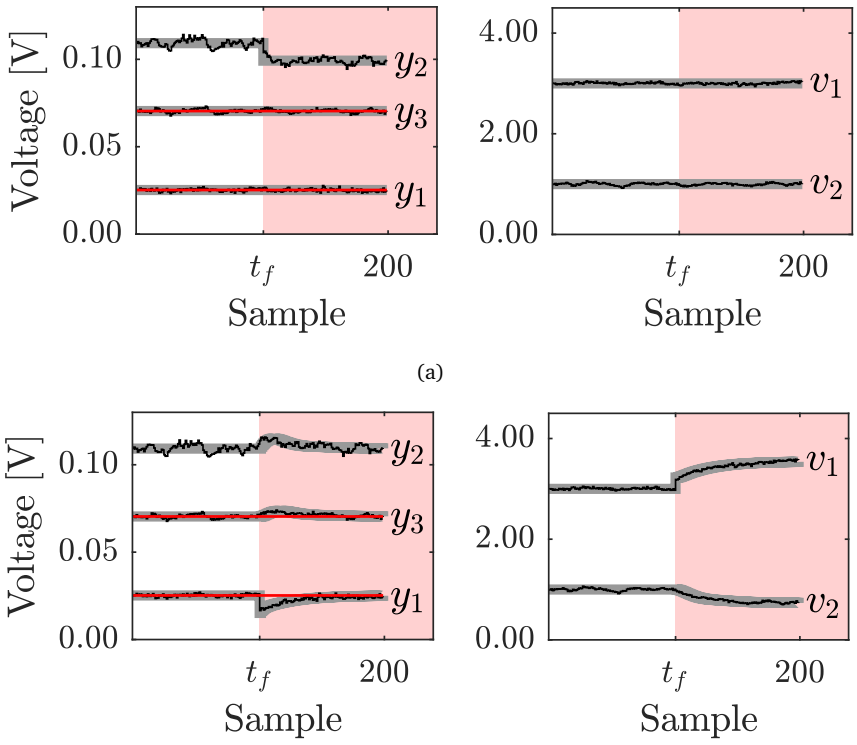


Figure C.15: Influence of fault at time t_f on (left) measurements and (right) control inputs. Red lines indicate the references for the measurements.

Table C.4: Influence of faults on variables with - denoting no influence, Δ denoting a temporary change that dies out, ↑ denoting a permanent positive drift in the steady state value, and ↓ denoting a negative drift.

Fault	r_1	r_2	v_1	v_2	y_1	y_2	y_3
Fault 1	-	-	-	-	-	↓	-
Fault 2	-	-	↑	↓	Δ	Δ	Δ

priori that the characteristic of open loop reference signal r_1 is insensitive to faults, reasoning path 2 is an invalid explanation of the fault. Reasoning path 1 is left as the only viable explanation for the detected fault, i.e., that it was caused by a positive drift in v_1 and a negative drift in v_2. This conclusion correspond with the observed influences of fault f_3 presented in Table C.4.

C.6 Discussion

Since the LP model in this paper is a static AE, diagnosis performs poorly when process variable samples contain temporal information. Reference changes (which induce dynamic process behavior) cause the SPE signal of Fig. C.14 to exceed its control limit, generating false alarms and hampering diagnosis. The process must reach steady-state to confirm that a false alarm has occurred, visualized by the

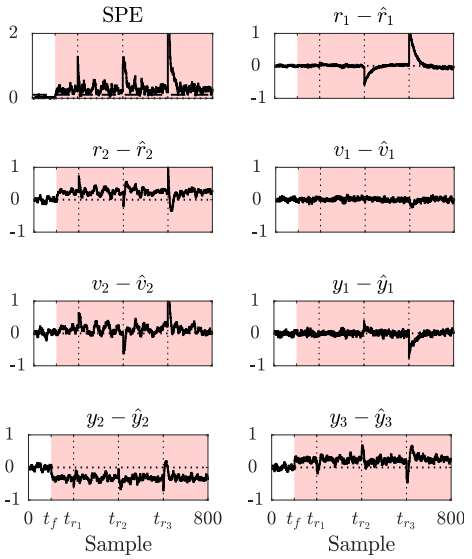


Figure C.16: Contribution plots of $\mathbf{X}_{f,1}$.

Table C.5: Causal paths of f_1.

Causal path	$\hat{r}_2$	$\hat{v}_2$	$\hat{y}_2$	$\hat{y}_3$	r_2	y_2	y_3
1	↓				×	↓	↓
2		↓			↓	↓	↓
3			↑		↑	×	↑
4				↓	↓	↓	×

Table C.6: Reasoning paths of f_1.

Reasoning path	$\hat{r}_2$	$\hat{v}_2$	$\hat{y}_2$	$\hat{y}_3$	r_2	y_2	y_3
1			↑		↑		↑
2	↓	↓		↓		↓	

SPE returning to below the control limit. This behavior is explained by reference to Fig. C.20. Transient samples, i.e., samples whose variance are primarily a result of reference changes, and steady-state samples, i.e., samples whose variance are explained by noise, of original variables (y_1, v_1) from validation set $\mathbf{X}_v$ are displayed in a scatter plot. Of interest are the reconstructed variables $(\hat{y}_1, \hat{v}_1)$ residing along the principal axis of variance from the AE in Fig. C.12; the principal axis of a conventional PCA model is included to demonstrate the superiority of an AE at capturing the nonlinear steady-state correlation between the data. Data

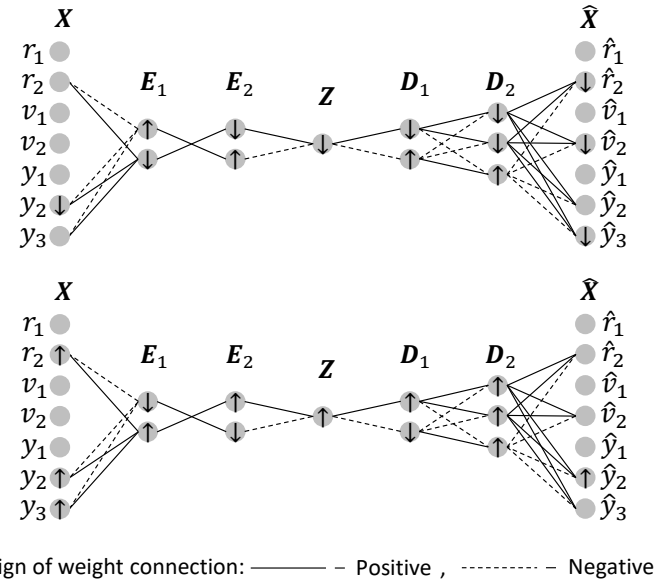


Figure C.17: Illustration of reasoning paths of f_1.

sampled during a transient phase is further away from the principal axis, i.e., its reconstruction, thereby generating a large SPE per Eq. (C.13).

The effect of reference changes is also observable in Figs. C.16 and C.18; contributions, with $r_1 - \hat{r}_1$ and $y_1 - \hat{y}_1$ in particular, abruptly fluctuate about the origin at the onset of a reference change such their interpretation, i.e., whether variable $\hat{x}_{new,i}$ has shifted positively or negatively, changes. Like with Fig. C.14, the process must reach steady-state before the plots are evaluated for shifts in $\hat{x}_{new,i}$. This is especially undesirable since the process exists in a faulty state and a quick diagnosis is required.

Figs. C.16 and C.18 indicate that the magnitude of steady-state contributions vary as a result of reference changes. This is explained by reference to Fig. C.20. Included are select samples for (y_1, v_1) and $(\hat{y}_1, \hat{v}_1)$ from $\mathbf{X}_{f,2}$. Here, the mean contribution $C_{y_1,1}$ is larger in magnitude than $C_{y_1,2}$. This is observable in the plot of $y_1 - \hat{y}_1$ in Fig. C.18, as the average contribution shifts towards the origin from samples [501-600] to samples [701-800]. Varying contributions occur because the nonlinear steady-state trend between y_1 and v_1 is explained by a nonlinear principal axis of variance. Since the AE minimizes the loss of information between (y_1, v_1) and $(\hat{y}_1, \hat{v}_1)$, it learns a projection whose direction varies with the nonlinear correlation of the nominal data, causing the magnitude of the contributions to vary as well. This exemplifies the limitation of magnitude-based contribution analysis, as varying contributions may

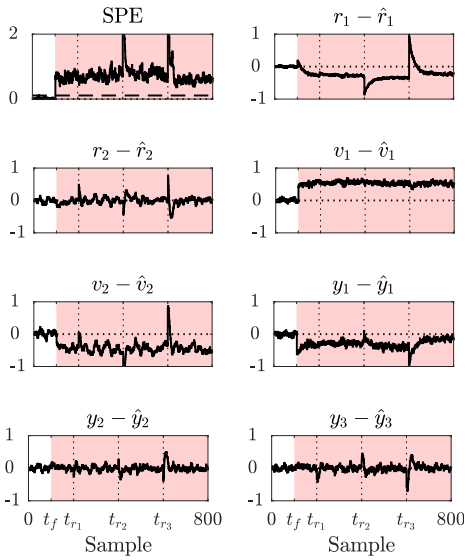


Figure C.18: Contribution plots of $\mathbf{X}_{f,2}$.

Table C.7: Causal paths of f_2.

Causal path	$\hat{r}_1$	$\hat{v}_1$	$\hat{v}_2$	$\hat{y}_1$	r_1	v_1	v_2	y_1
1	↑				×	↑	↓	↑
2		↓			↓	×	↑	↓
3			↑		↓	↓	×	↓
4				↑	↑	↑	↓	×

Table C.8: Reasoning paths of f_2.

Reasoning path	$\hat{r}_1$	$\hat{v}_1$	$\hat{v}_2$	$\hat{y}_1$	r_1	v_1	v_2	y_1
1	↑			↑		↑	↓	
2		↓	↑		↓			↓

yield different diagnosis for the same fault.

The limitation of this paper is that the examples are simple. Each produced two reasoning paths with one invalidated with *a priori* knowledge, resulting with a correct diagnosis. It is suspected that one may obtain more than two reasoning paths for larger systems and that it may be impossible invalidate some of them. However, one will still obtain a list of viable process variable drifts that explain the fault. This is an improvement over relying solely on magnitude-based contribution analysis.

It is important to remember that results from statistical fault diagnosis do not

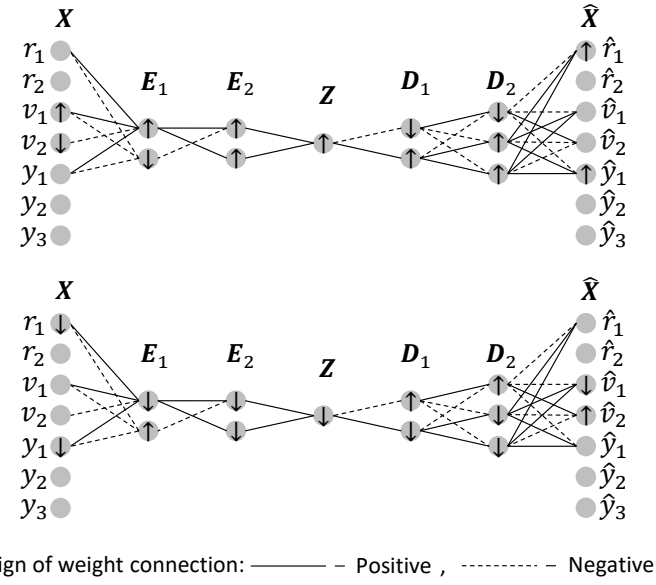


Figure C.19: Illustration of reasoning paths of f_2.

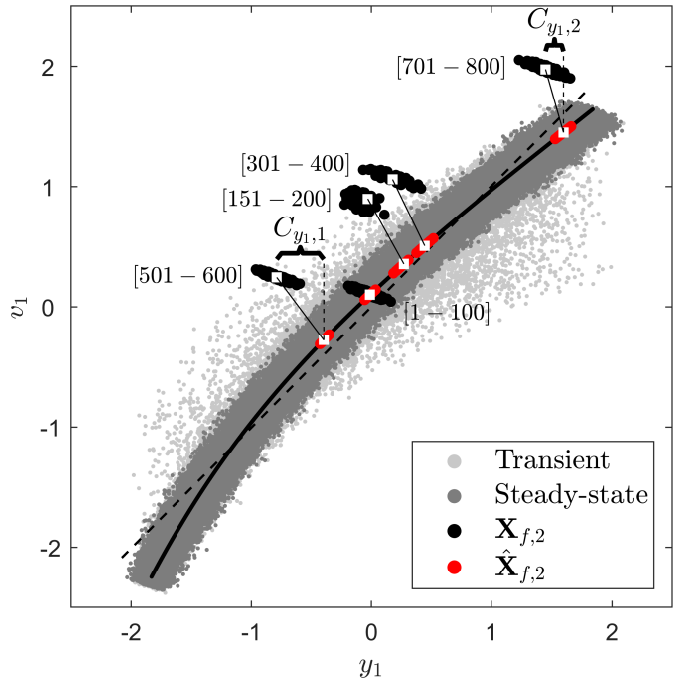


Figure C.20: Scatter plot of v_1 and y_1 of the validation set $\mathbf{X}_v$ as well as select samples from $\mathbf{X}_{f,2}$.

provide a cause of a fault, especially for complex faults. They can, however, form a basis for proposing a probable cause by an expert system. For example, the complex fault f_2 originates at sensor y_1 but its influence propagates to pump signals v_1 and v_2. However, this does not mean that there is something wrong with the pumps. An expert system is needed to decipher the cause of the isolated shifts of signals v_1 and v_2.

C.7 Conclusion

In this paper, a sparse AE for detection and isolation of abnormal process variable drifts is presented. In the proposed method, a static LP model is trained to perform data compression of dynamic process data. The naïve elastic net regularization penalty is introduced in the training scheme to shrink redundant interconnecting weights. A pruning algorithm then periodically removes (forcing to zero) less salient connections, ultimately producing a sparse network. Zeroing out irrelevant connections provides better interpretability of the network; correlated process variables are grouped together when projected to the latent space, leading to the discovery of

process knowledge.

Graph properties of the sparse AE are derived to give insight into its fault isolation capabilities. Since the activation function and final interconnectivity of hidden nodes is known, it is possible to predict the influence of abnormal process drifts in a subset of process variables on the drift in the reconstruction of remaining variables. When a fault is detected, matching between observed reconstruction drifts with prior predictions is performed to isolate the process variables that contain fault signatures.

Acknowledgments

The authors would like to acknowledge the support of the Danish Hydrocarbon Research and Technology Center(DHRTC) at the Technical University of Denmark.

Paper D

Modelling Nonlinearly Correlated Process Variables with Expanding Autoencoders

Ásgeir Daniel Hallgrímsson[1,*], Hans Henrik Niemann[1], Morten Lind[1]

[1]Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

Abstract:
An autoencoder (AE) is considered a nonlinear generalization of principal component analysis. An AE is a type of feedforward neural network configured for feature extraction that identifies nonlinear correlations among variables. From this, AEs are used for condition monitoring of nonlinear processes: a process is considered to be operating under abnormal conditions if the features of new process variable observations are inconsistent with the features of nominal observations. Reports in the process monitoring literature are unclear in regards to the selection of model complexity for identifying nonlinear correlations of a complex process variable space. The common approach is to include several hidden layers that gradually reduce the original variable space, while other reports propose AEs without any hidden layers. These configurations render an AE unable to perform feature extraction, thus reducing its performance at process monitoring. In this paper, an AE is trained to perform feature extraction for nonlinearly correlated data sets. The objective is to seek a principal curve, which is a smooth one-dimensional curve that passes through the middle of an m-dimensional variable space, providing a nonlinear joint summary of the variables. Three examples are provided to illustrate the use of AEs to identify principal curves. Results show that an AE successfully reduces dimensionality and provides a descriptive principal curve only if the AE includes additional hidden layers with dimensions larger than the original variable space.

[*]Corresponding author. E-mail: asdah@elektro.dtu.dk

D.1 Introduction

The task of extracting statistical information from process variables is an integral component of statistical process monitoring (SPM). Several techniques for obtaining salient information from variables have been proposed in the literature [129], but of interest in this paper is feature extraction: a mathematical procedure for dimensionality reduction that consists of deriving low-dimensional quantities intended to be informative about a high-dimensional variable vector. Feature extraction facilitates anomaly detection [19]. By comparing the features of new observations to those of a reference data set, abnormal observations are identified if their features deviate from the reference set's features. This technique translates to SPM where the reference set consists of historical process data sampled from a process operating under nominal conditions [91].

It is a common occurrence that process variables are correlated with one another. For example, control and measurements signals are correlated via an input-output relationship between actuators and process states. Several sensors may monitor the same physical property, and different measured properties may be related to one another via a physical law. As pointed out by Kramer: "The *superficial dimensionality* of process data, or the number of individual observations constituting one measurement vector, is often greater than its *intrinsic dimensionality*, the number of independent variables underlying the significant non-random variations in the observations" [72]. Fundamentally, feature extraction exploits variable correlations to derive informative features. SPM based on feature extraction thus consists of querying whether correlations of new process variable observations coincide with those established by a feature extraction model.

In principle, feature extraction methods obtain a set of principal variables with a dimensionality-reducing transformation of the original variables. Ideally, the dimensionality of the reduced representation should correspond to the intrinsic dimensionality of the data. The optimal linear transformation is given by principal component analysis (PCA) [63]. Features embedded in the linear subspace are derived via an orthogonal vector transformation of the original variables. The transformation given by PCA preserves information if variables are sampled from a multivariate distribution that satisfies the assumption of normality. PCA has demonstrated success in the monitoring of linear processes, primarily due to its simplistic nature. Independent component analysis (ICA) may be applied if the normality assumption is not met [53]. ICA was mainly designed for blind-source separation, which is the task of determining independent source features from a set of multivariate non-normal distributed variables that are a linear mixture of the

sources [126]. ICA thus finds a transformation that linearly decomposes the original variables into a set of mutually independent feature components. ICA sees great success to process monitoring when the source of significant variation in process variables is unknown and cannot be attributed to common cause variation [83], [84].

Nonlinear feature extraction methods have been proposed for addressing nonlinear variable correlations. Kernel PCA (KPCA) maps the original variable space to a higher-dimensional space with a nonlinear kernel function [82], [119], [100]. Feature extraction is achieved by performing PCA in the higher-dimensional space. Autoencoders (AEs) are neural networks configured for feature extraction [50], [45], [46]. An AE employs a basis function approach to solve a nonlinear optimization function; it learns a transformation of the original variables to a reduced space with the constraint that the original variables can be reconstructed from the reduced space with minimal loss of information.

A compelling trait of AEs is the flexibility in varying their model complexity. An AE may comprise multiple hidden layers with varying dimensions, as well as employ different nonlinear activation functions at its network neurons. As a matter of fact, an AE given certain optimization constraints can uncover the transformations of other feature extraction techniques frequently employed for process monitoring. For instance, an AE composed of linear hidden layers will yield a feature extraction mapping that corresponds to the subspace projection obtained from PCA [7]. An AE augmented with the additional constraint that the latent variables within its feature space are as independent as possible will optimize its transformations that are then equivalent to ICA [67]. Le at al. [76] show than an AE can learn an optimal kernel function that offers superior feature extraction performance than KPCA.

Reports in the literature are unclear in regards to the number and dimension of hidden layers needed for monitoring nonlinear processes. Kramer [72] reported in 1991 that AEs may require hidden layers with dimensions larger than the original variable space, i.e., an *expanding* layer, when performing feature extraction for complex nonlinear variable distributions. Despite Kramer's findings, reports in the process monitoring literature tend to propose AEs configured to constrict the original variable space without higher-dimensional hidden layers [86], [61], [141]. Some reports propose AEs lacking hidden layers altogether, such as in the works of Cheng et al. [22]. In their work, an AE underperformed at extracting features from a parabolic data distribution. To improve modelling performance, the AE was augmented with additional functionalities present in variational AEs and recurrent neural networks, which significantly increased its implementation complexity. It begs the question whether the original problem, that is, extracting features from a

parabolic data set, could be solved by simply including a hidden layer that expanded upon the original variable space.

An important component to AE-based anomaly detection is training the network to learn the correlations of nominal process variables. In this paper, a study is performed on the significance of the hidden layers of an AE. More specifically, the number and size of hidden layers were varied to analyze their effects on an AE's performance at extracting features from nonlinear data sets. Three two-dimensional data sets are provided for the analysis. The key result in this study was that complex data sets can be reduced to lower dimensions as long as the AE includes hidden layers that expand the dimension of the original variables space. Consequently, an AE employed for process monitoring will observe improved performance in anomaly detection since it will have learned an appropriate representation of nominal process variable correlations.

The organization of the paper is as follows. Principal curves, which are one-dimensional curves that summarize the joint distribution of multivariate variables, are discussed in section 2. Section 3 provides a description of latent projection and AEs. The concept behind transforming an original variable vector to a higher dimensional space in order to improve the fitting of numerical models is presented in section 4. Section 5 demonstrates the effects of AE model complexity on its ability to perform feature extraction. More specifically, the effects of including hidden network layers that expand the original variable space are investigated. The last two sections provide a discussion and conclusion of the results.

D.2 Principal Curves

Consider a data set consisting of n observations of m variables. It is sometimes the case that one wishes to summarize the joint statistical characteristics exhibited by the data. For example, linear regression models the relationship between dependent and independent variables. In certain situations, one may not have a preference for which variables are labelled as dependent and independent (such as in image analysis) but would still like to summarize their joint characteristics. An alternative is to seek a q-dimensional principal manifold that summarizes an m-dimensional variable space, where $q < m$ [47]. The chosen dimension q depends on the desired summary.

This paper directs its focus on one-dimensional principal manifolds called principal curves. A principal curve is a smooth, one-dimensional curve that traverses through the center of an m dimensional variable space. Its shape provides a nonlinear summary of the data by minimizing the sum of squared deviations in all of the

variables to the curve. The curve is a vector $\mathbf{f}(\lambda)$ of m functions of a single variable λ. These functions are called the coordinate functions. λ parameterizes the curve and provides a total ordering along it. Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random vector. The curve $\mathbf{f}$ is called a principal curve of $\mathbf{x}$ if

$$E(\mathbf{x}|\lambda_{\mathbf{f}}(\mathbf{x}) = \lambda) = \mathbf{f}(\lambda) \tag{D.1}$$

where the projection index $\lambda_{\mathbf{f}} : \mathbb{R}^m \to \mathbb{R}^1$ is defined as

$$\lambda_{\mathbf{f}} = \sup_{\lambda}\{\lambda : ||\mathbf{x} - \mathbf{f}(\lambda)|| = \inf_{\mu}||\mathbf{x} - \mathbf{f}(\mu)||\} \tag{D.2}$$

The projection index $\lambda_{\mathbf{f}}(\mathbf{x})$ is the value of λ for which $\mathbf{f}(\lambda)$ is closest to $\mathbf{x}$. Fig. D.1 illustrates the definition of a principal curve in two dimensions. Here, $\mathbf{f}(\lambda)$ is a principal curve if it passes through a series of projections that minimize the sum of squared deviations of samples that project there orthogonally to $\mathbf{f}(\lambda)$. Naturally, $\mathbf{f}(\lambda)$ is purely an estimation of the principal curve that summarizes the distribution of $\mathbf{x}$. Its approximation will further resemble its theoretical counterpart as more samples are drawn from $\mathbf{x}$.

The definition of a principal curve does not imply that each sample $\mathbf{x}$ has a unique projection index $\lambda_{\mathbf{f}}(\mathbf{x})$. It is plausible that a single point $\mathbf{f}(\lambda)$ is shared by multiple

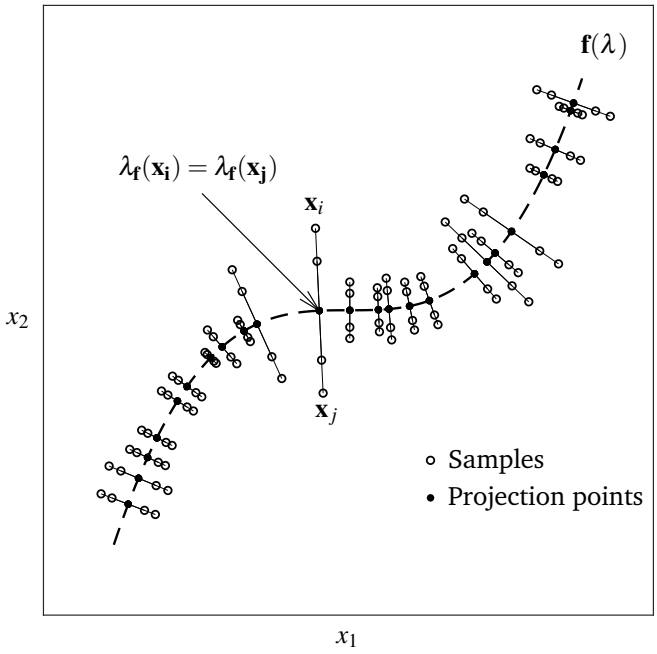


Figure D.1: Principal curve f that summarizes a given set of samples. The points $\mathbf{x}_i$ and $\mathbf{x}_j$ share a projection point. Their projection vectors are a scaled version of the other.

observations. This is illustrated in Fig. D.1 where 96 observations are summarized by a principal curve consisting of 24 projections. Each projection point is the average of, and thus shared by, four observations. Multiple samples residing on the same projection vector is attributed to noise and other common cause variations occurring in the direction of the projection vector.

D.3 Latent Projection and Autoencoders

Latent projection techniques reduce the dimension of a variable space to a smaller set of latent variables that retain principal information about the original variables. The procedure is formulated as a machine learning problem. Initially, n observations of the variable vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ are sampled to produce the reference data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$. An optimal data reduction transformation to the latent space $\mathbf{Z} \in \mathbb{R}^{n\times q}$ is sought for. The learning criteria is that the latent representation $\mathbf{Z}$ retains essential information about the original set $\mathbf{X}$.

Formulating the requirement for $\mathbf{Z}$ to retain information about $\mathbf{X}$ is difficult if no *a priori* knowledge about the statistical properties of the original variable space exists. Therefore, the latent projection technique is augmented with a data expansion procedure; the reduced space $\mathbf{Z}$ is used to reconstruct the original variable space $\hat{\mathbf{X}} \in \mathbb{R}^{n\times m}$. Combining the data reduction and expansion learning problems promotes an LP model to learn a transformation for $\mathbf{Z}$ that retains essential information required to reproduce the original variable space [50]. Data compression and decompression are selected to minimize the loss of information represented by the average squared prediction error:

$$\mathscr{L}(\mathbf{X},\hat{\mathbf{X}}) = \frac{1}{n}\left|\left|\mathbf{X}-\hat{\mathbf{X}}\right|\right|^2 \tag{D.3}$$

An autoencoder is a type of artificial neural network that learns latent representations for a data set [18]. The simplest form of an AE is a multilayered, feedforward, non-recurrent neural network, illustrated in Fig. D.2. The encoder part corresponds to the compression of $\mathbf{X}$ to $\mathbf{Z}$ and the decoder part corresponds to the subsequent expansion of $\mathbf{Z}$ to $\hat{\mathbf{X}}$. The encoder and decoder transformations are generated with a nonlinear basis function approach. The network is composed of several vectors of nodes, known as network layers. With the exception of the input layer, each layer is a component wise nonlinear function of a linear transformation of its previous layer.

The encoder maps the input $\mathbf{X} \in \mathbb{R}^{m \times n}$ to the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$:

$$\mathbf{E}_i = \begin{cases} \sigma_1^e \left(\mathbf{W}_1^e \mathbf{X} + \mathbf{b}_1^e \right), & \text{for } i = 1 \\ \sigma_i^e \left(\mathbf{W}_i^e \mathbf{E}_{i-1} + \mathbf{b}_i^e \right), & \text{else} \end{cases} \tag{D.4}$$

$$\mathbf{Z} = \sigma^z \left(\mathbf{W}^z \mathbf{E}_N + \mathbf{b}^z \right)$$

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the weight matrix between layers $i-1$ and i, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer. The decoder maps the latent variables $\mathbf{Z} \in \mathbb{R}^{q \times n}$ to the input reconstruction $\hat{\mathbf{X}} \in \mathbb{R}^{m \times n}$:

$$\mathbf{D}_j = \begin{cases} \sigma_1^d \left(\mathbf{W}_1^d \mathbf{Z} + \mathbf{b}_1^d \right), & \text{for } j = 1 \\ \sigma_j^d \left(\mathbf{W}_j^d \mathbf{D}_{j-1} + \mathbf{b}_j^d \right), & \text{else} \end{cases} \tag{D.5}$$

$$\hat{\mathbf{X}} = \sigma^{\hat{x}} \left(\mathbf{W}^{\hat{x}} \mathbf{D}_M + \mathbf{b}^{\hat{x}} \right)$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the weight matrix between layers $j-1$ and j, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{x}}$, $\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer. The modifiable parameters $\mathbf{W}_i^e, \mathbf{b}_i^e, \mathbf{W}^z, \mathbf{b}^z, \mathbf{W}_j^d, \mathbf{b}_j^d, \mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with respect to minimizing the loss function in Eq. (D.3) via stochastic gradient descent [99].

The AE estimates a principal curve of a data set by setting the dimension of its latent layer $q = 1$ (thus $\mathbf{Z} \triangleq z_1$) [32], [92]. Minimization of Eq. (D.3) generates the projection points $\hat{\mathbf{x}}$ that minimize the sum of squared deviations of observations

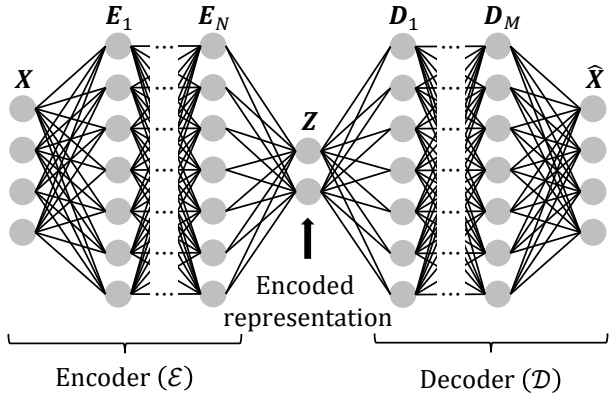


Figure D.2: Illustration of an autoencoder.

for $\mathbf{x}$ that project there. The projection points are summarized by a projection curve parameterized by $\mathbf{f}(z_1)$ that is an estimate for the theoretical principal curve $\mathbf{f}(\lambda)$. The requirement for orthagonal projections is met implicitly by Eq. (D.3) as orthagonal projections tend to minimize Eq. (D.3). However, as will be shown later, this depends on the complexity of the LP model.

D.4 Transforming variables to a higher dimensional space

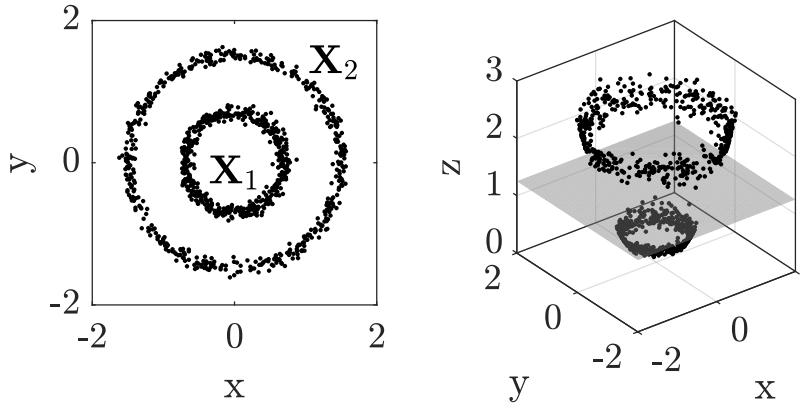
It is sometimes the case that a model is not suitable for modeling a particular data set. This occurs when the basis of the method does not satisfy the complexity of the data. Consider the data sets $\mathbf{X}_1$ and $\mathbf{X}_2$ shown in Fig. D.3(a). The discrepancy between the two sets is explained by the radius of the circles they form centred at the origin. The task is to find a hyperplane, i.e., a one-dimensional straight line, that linearly classifies the data points. It is evident that a linear basis for the classifier is insufficient - the nonlinear complexity of the data prohibits the existence of of a straight line that separates it. Therefore, in the data's current form, a nonlinear decision boundary is required to partition the data. In such a situation, mapping the data to a higher dimensional space produces new features that enhance model performance. Fig. D.3(b) displays the data mapped to a three-dimensional space. The z axis corresponds to the distance of data points to the origin in the xy plane, i.e., the radius of the circles formed by the data. The disparity between $\mathbf{X}_1$ and $\mathbf{X}_2$ is revealed by the z axis, such that a hyperplane that separates $\mathbf{X}_1$ and $\mathbf{X}_2$ is easily found. In the example above, it is impossible for a linear classifier to separate $\mathbf{X}_1$ and $\mathbf{X}_2$ in the original space. However, mapping the data to a three-dimensional space produces clusters in the data that can be better separated than in the original space - the data has been augmented but the approach to classification remains unchanged. It is reiterated that although the data is augmented with a third variable z, the basis of the classifier remains linear.

Mapping the predictors of a model to a higher dimensional space prior to fitting the model forms the basis of Kernel methods [120]. They are a class of algorithms that turn a linear model into a nonlinear model by mapping its predictors with a nonlinear kernel function. In the example in Fig. D.3(b), a kernel function is used to compute z. The motive for using kernel methods is that expanding the original variable space makes it possible for variable relations, that were impossible to model in the original variable space, to be modeled in the higher dimensional space. A similar functionality is attained with FNNs by including hidden layers whose dimensions are larger than the input layer. In such a configuration, the *expanding* FNN can learn a transformation to a higher dimensional space that aids

in its ultimate modeling task. It is noted that the Kernel method is considered as a separate method that augments the original machine learning task, while a FNN employing a hidden layer that expands the original variable space does not change its formulation.

An expanding AE is an AE with a hidden layer in its encoder and decoder parts whose dimension is larger than the original variable space. An example of an expanding AE with a 7-9-3-2-3-9-7 configuration is shown in Fig. D.4(b). Kramer [72] reported that a condensing AE, that is, an AE that lacks an expanding layer, is incapable of modeling certain data distributions describing complex, nonlinear one dimensional manifolds such as a circle. Kramer showed that accptable model performance required the inclusion of expanding hidden layers. The works of Scholz et al. [117], [116] further demonstrate the need for expanding AEs for one dimensional manifolds existing in higher order spaces, such as a circle or a helix. Regardless of their depth, AEs, and possibly all forms of neural networks, often require layers wider than the input space to handle data with certain topological properties, such as links, intersections, or sampled from a multivalued function [102], [112].

Although the need for AEs to include expanding layers was proclaimed in the 90s and early 2000s, reports in the process monitoring literature are unclear regarding this requirement. One approach is to include no additional hidden layers, thereby configuring an AE to consist of an original variable layer, a latent layer, and a reconstruction layer [86], [61], [141]. Fig. D.4(c) shows such an AE. Although



(a) (b)

Figure D.3: Scatter plot of (a) data sets $\mathbf{X}_1$ and $\mathbf{X}_2$ and (b) mapping of the data to a three-dimensional space with $z = x^2 + y^2$.

this configuration is reportedly sufficient for feature extraction of nonlinear process variables, it begs the question of whether the process is ultimately nonlinear since Bourlard and Kamp [13] show that AEs lacking hidden layers learn linear PCA-like projections. The cause for this is that neuron activations can remain in the linear regions of nonlinear activation functions such as the sigmoid or tangent hyperbolic. A second approach is include hidden network layers to assemble a deep AE, also referred to as a stacked AE, that resembles the AE in Fig. D.4(a). In this configuration, the original variable space is gradually compressed to the feature space, and then gradually expanded to the reconstruction space. This effectively increases the modelling capabilities of the AE by introducing additional modelling parameters. However, Kramer [72] shows that, for certain nonlinear data distributions, adding additional layers may not increase modelling capabilities unless the hidden layers expand the original variable space.

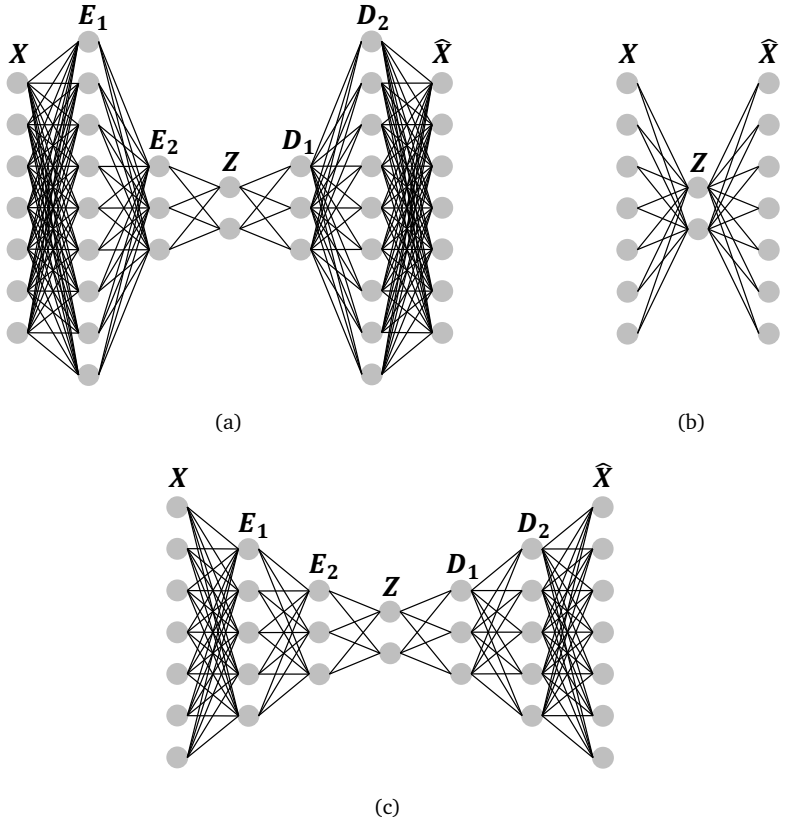


Figure D.4: Illustration of (a) an expanding AE, (b) a single-layer AE, and (c) a condensing AE.

D.5 Result

The effects of hidden layers on an AE's performance at fitting a principal curve for a two-dimensional variable distribution are analysed in this section.

D.5.1 Example 1: Parabolic distribution

Consider the following nonlinear process [134]:

$$\begin{aligned} t &= \mathscr{U}[-0.5, 0.5] \\ q_1 &= t \\ q_2 &= t^2 \\ x_1 &= x_1 + \mathscr{N}(0, 0.02) \\ x_2 &= x_2 + \mathscr{N}(0, 0.02) \end{aligned} \tag{D.6}$$

where t is the underlying variable of the process and x_1 and x_2 are the measurements of the state variables q_1 and q_2. Furthermore, $\mathscr{U}[a,b]$ denotes the uniform distribution in the range (a,b) and $\mathscr{N}(\mu,\sigma^2)$ the normal distribution with mean μ and variance σ^2. The process is sampled to produce the training set $\mathbf{X}_t$ (consisting of 10,000 samples) and validation set $\mathbf{X}_v$ (consisting of 5,000 samples). Fig. D.5 shows the validation set $\mathbf{X}_v$. The data may be summarized with a parabolic principal curve. The implications of the process's characterization in Eq. (D.6) are that (a) the principal curve describes a nonlinear relationship between x_1 and x_2; and (b) x_1 is a multivalued variable along the curve since two values of x_1 are associated with a single value of x_2. An AE was trained to perform feature extraction on $\mathbf{X}_t$ to approximate the principal curve.

One approach to AE-based latent projection is to reduce the original variable space to a smaller dimension without additional hidden layers. Fig. D.6(a) illustrates the connectivity of a 2-1-2 AE produced when proceeding in this direction. Included are the magnitudes of weights and biases obtained after optimizing the AE with the training set $\mathbf{X}_t$. The result is that the activation of latent variable z_1 depends solely on x_1 since the weight between x_2 and z_1 is zero. Fig. D.6(b) plots the tangent hyperbolic $\tanh(w \cdot x)$ with $w = -0.45$, i.e., the weight between x_1 and z_1. The interval $x \in [-0.5, 0.5]$ encloses the random set $t = \mathscr{U}[-0.5, 0.5]$. Given the bounds of t and the value of w, it can be seen that the activation of z_1 remains relatively within the linear region (dashed line) of the tangent hyperbolic. In fact, linear approximation of $\tanh(w \cdot x)$ along the interval $x \in [-0.5, 0.5]$ has a mean squared error (MSE) of $3.34e-7$. Nonlinear capabilities of the network are not fully utilized, and the AE reduces to a linear model since the layer between $\mathbf{Z}$ and $\hat{\mathbf{X}}$ is linear.

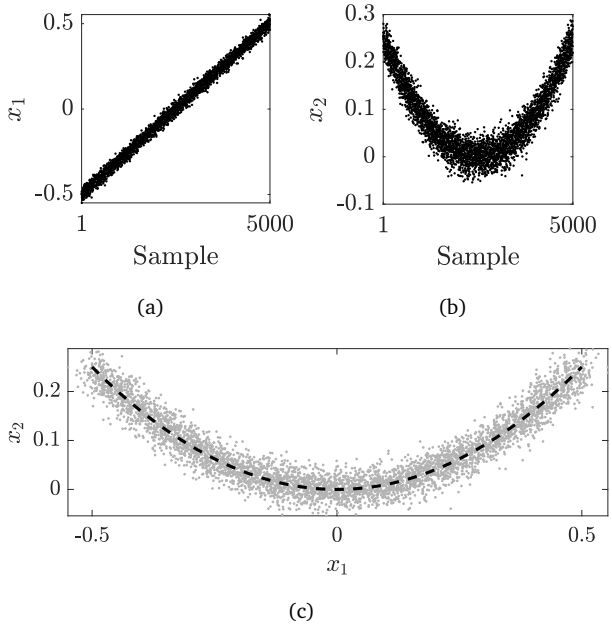


Figure D.5: Samples of (a) x_1 and (b) x_2 over the validation set $\mathbf{X}_v$, as well as a (c) scatter plot of the samples.

Fig. D.7 displays the sample-wise result of propagating $\mathbf{X}_v$ through the 2-1-2 AE. Kramer [72] reports that to successfully reconstruct nonstochastic data sets from a single factor, an AE must produce a latent variable z_1 that is analogues to the underlying variable t. This proposition is extended to stochastic data sets in this paper, namely, that the mean activation of z_1 must be analogues to the underlying

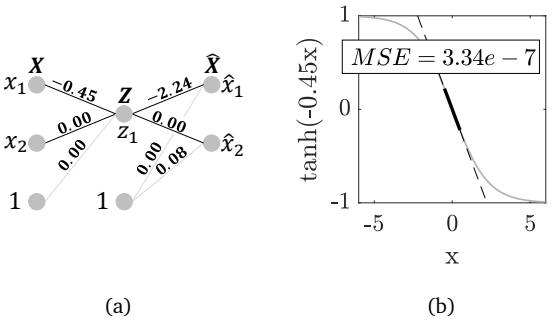


Figure D.6: (a) Illustration of the 2-1-2 AE and (b) plot of the tangent hyperbolic over (grey) $x \in [-6, 6]$ and (black) $x \in [-0.5, 0.5]$.

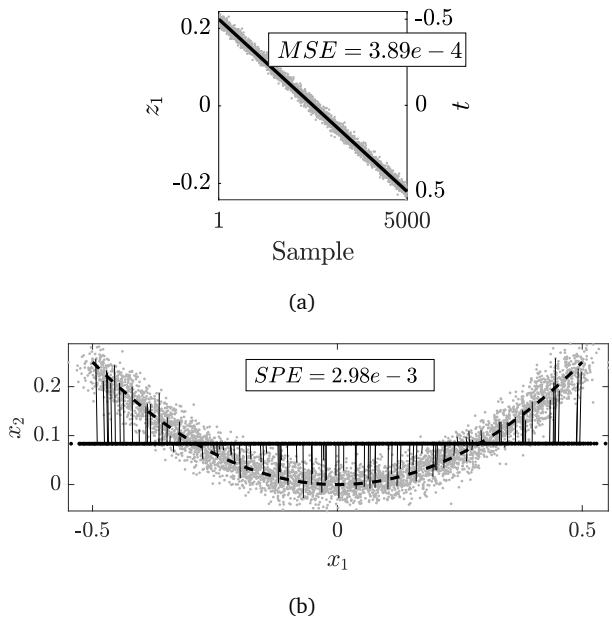


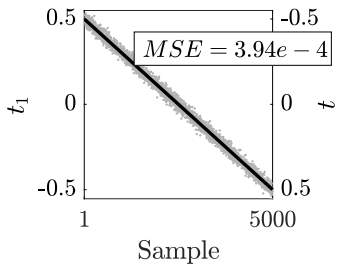
Figure D.7: From propagating validation set $\mathbf{X}_v$ through the 2-1-2 AE: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

variable t. However, this remark is not supported by Fig. D.7. Despite the similarity between z_1 and t (estimation error MSE $= 3.89e-4$ is caused by stochasticity in the original variables x_1 and x_2), the AE performs an inaccurate reconstruction by approximating the principal curve with a linear projection curve. The cause for this is interpretable via the weight connections in Fig. D.6(a). The activation of z_1 is similar to t because the latent variable is solely dependent on the measurement x_1 which, in turn, is equivalent to t with additional noise. The magnitude of the weight connection between the unbiased reconstruction $\hat{x}_1$ and z_1 is close to the inverse of the weight between x_1 and z_1 ($1/-0.45 \approx -2.23$). Because z_1 remains relatively in the linear region of the tangent hyperbolic (Fig. D.6(b)), z_1 and $\hat{x}_1$ are essentially a copy of x_1. The reconstruction $\hat{x}_2$ is independent of z_1 (since the weight between the two nodes is zero) and is determined entirely by its bias. Consequently, $\hat{x}_2$ is constant over the validation set. In fact, the magnitude of its bias (0.08) is equivalent to the expected value $E(x_2)$; the AE reconstructs $\hat{x}_2$ with the learned the mean of x_2. The results indicate that the AE priorities reconstructing the variable with the greatest variance, namely x_1, and reconstructs x_2 with its mean.

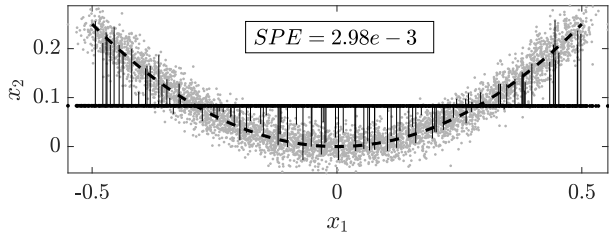
As previously stated, Fig. D.6(b) indicates that the 2-1-2 AE reduces to a linear

model. In fact, it is possible to show that the AE is an approximation of PCA. PCA is a linear procedure with the following properties: (a) its projection curve is a straight line when reducing data to a single factor t_1; and (b) its projections are orthanormal. Fig. D.8 shows the result of propagating the validation set $\mathbf{X}_v$ through a PCA model built on the training set $\mathbf{X}_t$. The activation of t_1 is similar to that of z_1. The projection curve of the PCA model is similar to that of the 2-1-2 AE. The two models also have the same SPE. These results support the findings of Baldi and Hornik [7], namely, that an AE with insufficient hidden layers may reduce to a PCA model.

It is favorable that an AE's nonlinear potential is utilized when reducing the dimension of a nonlinear data set. This may be achieved by including hidden layers that do not reduce the dimension of the original variable space but rather increase or, at the very least, keep it constant. Fig. D.9 shows a 2-2-1-2-2 AE trained on the training set $\mathbf{X}_t$. This network has increased nonlinear processing capacity since it employs additional nonlinear layers. All of the optimized weights are non-zero in this configuration. Contrary to the AE in Fig. D.6, original variables are not ignored which ensures that z_1 is a function of both x_1 and x_2. Figs. D.10(a) and D.10(b) show the activation of encoder nodes $e_{1,1}$ and $e_{1,2}$ over $\mathbf{X}_v$. The plots indicate



(a)



(b)

Figure D.8: From propagating validation set $\mathbf{X}_v$ through the PCA model: (a) Activation of t_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

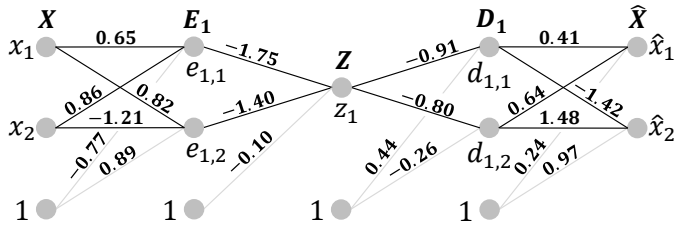


Figure D.9: Illustration of the 2-2-1-2-2 AE.

that the encoder splits the parabola into two; node $e_{1,1}$ becomes more active, i.e., absolute magnitude closer to 1 and further away from 0, as $t \to -0.5$ whereas node $e_{1,2}$ becomes more active as $t \to 0.5$. When the activation of these two nodes are combined in the latent layer z_1 (Fig. D.10(c)), the network performs well at fitting the principal variable t (MSE $= 3.75e-4$). Note that in this AE configuration, the MSE is a little larger than the MSE of the 2-1-2 configuration in Fig. D.7(a). This is explained by the fact that, contrary to the 2-1-2 AE, the 2-2-1-2-2 AE retains the nonlinear variable x_2 along with x_1 when processing z_1, which (a) induces more noise in z_1 and (b) makes compression more challenging than taking a copy of x_1.

Figs. D.10(d) and D.10(e) show the activation of decoder nodes $d_{1,1}$ and $d_{1,2}$ obtained by a nonlinear transformation of z_1. Both nodes indicate a linear activation with slight curvature. Node $d_{1,1}$ becomes more active as $t \to 0.5$ and $d_{1,2}$ becomes more active as $t \to -0.5$. Figs. D.10(f) and D.10(g) show the activation of reconstructions $\hat{x}_1$ and $\hat{x}_2$, obtained from a linear transformation of the decoder layer. The reconstructions are similar to the original variables x_1 and x_2 in Figs. D.5(a) and D.5(b). It is possible to infer the shape of the reconstructions by looking at the weights between layers $\mathbf{D}_1$ and $\hat{\mathbf{X}}$. Activation of $\hat{x}_1$ is obtained by adding the activations of $d_{1,1}$ and $d_{1,2}$, which negates the observed curvatures to produce a line. Activation of $\hat{x}_2$ is obtained by subtracting the activation of $d_{1,2}$ from $d_{1,1}$, equivalent to flipping the curve of $d_{1,1}$ and adding it to $d_{1,2}$, to produce the parabolic shape of $\hat{x}_2$.

Fig. D.10(h) shows a scatter plot of the original variables x_1 and x_2, reconstructions $\hat{x}_1$ and $\hat{x}_2$, as well as a few projection lines. The reconstruction provided by the 2-2-1-2-2 AE is more accurate compared to the 2-1-2 AE; the SPE has been reduced by a factor of 14.1. The projection curve resembles the principal curve $\mathbf{f}(\lambda)$, although the projections are not perfectly orthagonal. In fact, they are less orthogonal compared to projections of the 2-1-2 AE (Fig. D.7(b)). In this case however, it is apparent that the loss in orthogonality is well justified by the increased goodness of fit for the principal curve.

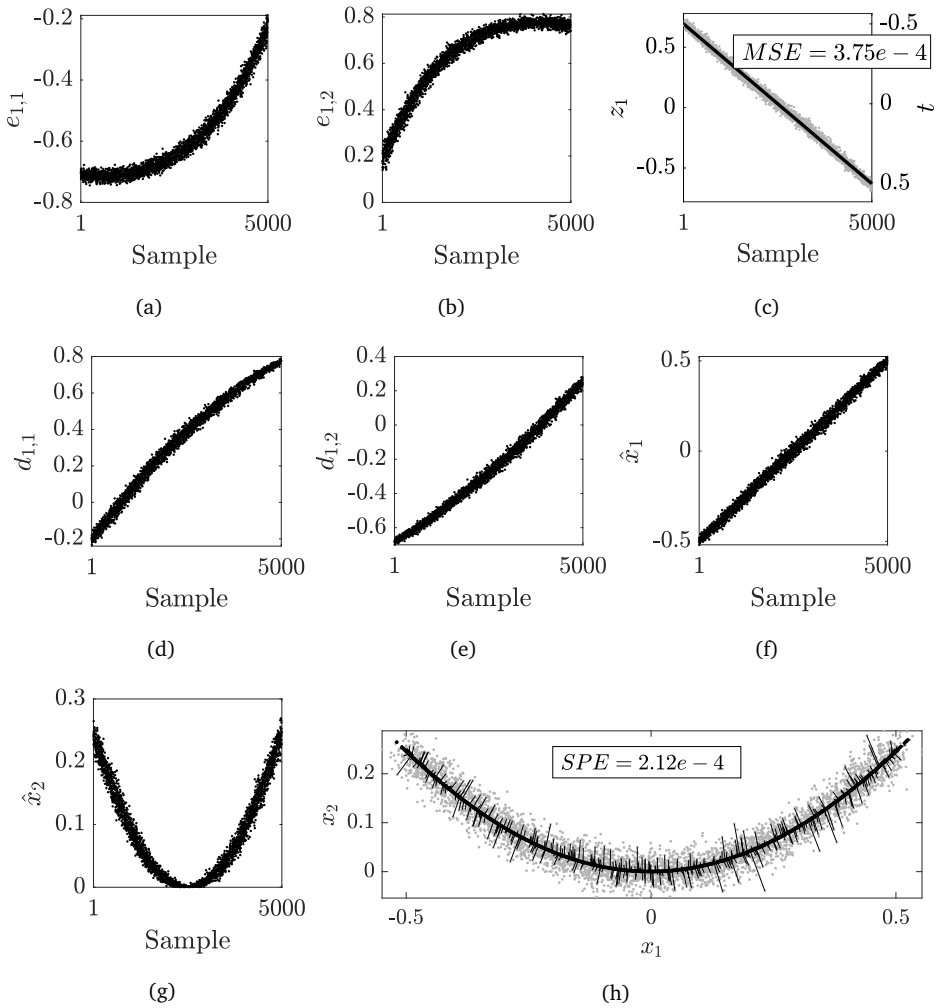


Figure D.10: From propagating validation set $\mathbf{X}_v$ through the 2-2-1-2-2 AE: (a)-(g) Activation of network neurons over samples and (h) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

D.5.2 Example 2: Circular distribution

Consider the following nonlinear process:

$$\begin{aligned} t &= \mathscr{U}[1.0\pi, 2.0\pi] \\ q_1 &= \sin(t) \\ q_2 &= \cos(t) \\ x_1 &= x_1 + \mathscr{N}(0, 0.05) \\ x_2 &= x_2 + \mathscr{N}(0, 0.05) \end{aligned} \tag{D.7}$$

The process is sampled to produce the training set $\mathbf{X}_t$ and validation set $\mathbf{X}_v$, consisting of 10,000 and 5,000 samples, respectively. The validation set $\mathbf{X}_v$ is shown in Fig. D.11. The data is summarized by a principal curve taking the shape of a semi circle. Similar to the principal curve of Eq. (D.6), it (a) describes a nonlinear relationship between x_1 and x_2; and (b) depicts x_1 as a multivalued variable. The principal curve in Fig. D.11(c) exhibits greater curvature at the edges of the data set, adding complexity. This is in part due to the nonlinear component contained in x_1.

Fig. D.12 illustrates the weights of a 2-2-1-2-2 AE trained on the $\mathbf{X}_t$. Figs. D.13(a)-D.13(g) show the activation of each node over $\mathbf{X}_v$. It can be seen that the activations are very similar to that of Figs. D.10(a)-D.10(g) but with some differences. For example, $e_{1,1}$ and $e_{1,2}$ both depict a parabolic activation but containing two-thirds of a parabola instead of a half. The activation for $d_{1,1}$ and $d_{1,2}$ indicate a linear activation that has additional curvature at its ends. The activation of reconstructions $\hat{x}_1$ and $\hat{x}_2$ show that the AE provides a decent reconstruction with some discrepancies; the ends of $\hat{x}_1$ are slightly curved compared to x_1 and the ends of $\hat{x}_2$ have additional curvature that is not present in x_2. The net result is seen in Fig. D.13(h), namely, that the AE approximates the half circle with a parabolic projection curve.

Model misfitting with small AEs becomes more apparent as the data sets become more complex. Consider the following nonlinear process:

$$\begin{aligned} t &= \mathscr{U}[0.5\pi, 2.0\pi] \\ q_1 &= \sin(t) \\ q_2 &= \cos(t) \\ x_1 &= x_1 + \mathscr{N}(0, 0.05) \\ x_2 &= x_2 + \mathscr{N}(0, 0.05) \end{aligned} \tag{D.8}$$

The process is sampled to produce training set $\mathbf{X}_t$ and validation set $\mathbf{X}_v$, consisting of 15,000 and 7,500 samples, respectively. The validation set $\mathbf{X}_v$ is shown in Fig. D.14. The data is summarized by a principal curve taking the shape of a three-quarter of a circle centered at the origin. Essentially, the data in Fig. D.11 has been extended by

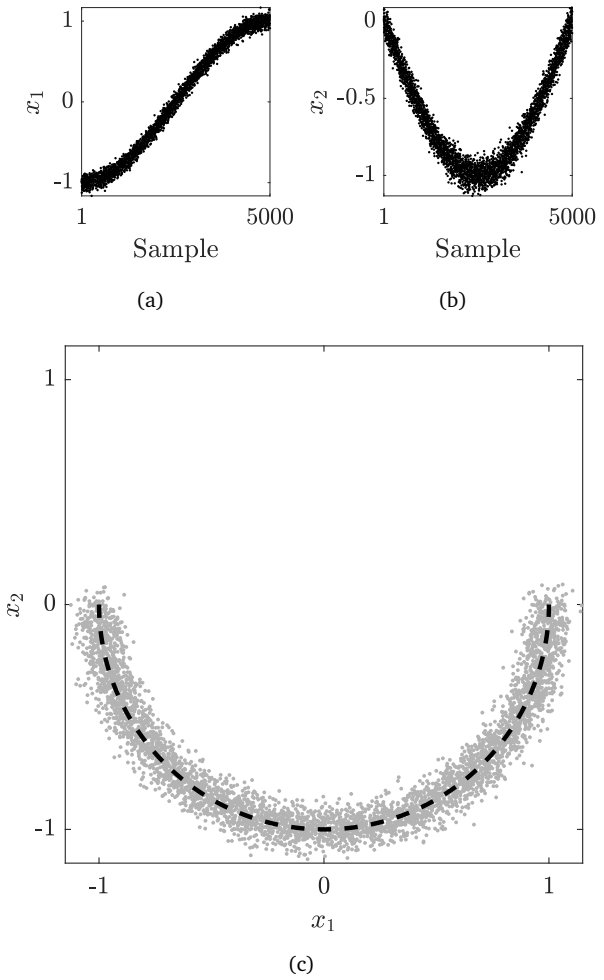


Figure D.11: Samples of (a) x_1 and (b) x_2 over the validation set $\mathbf{X}_v$, as well as a (c) scatter plot of the samples.

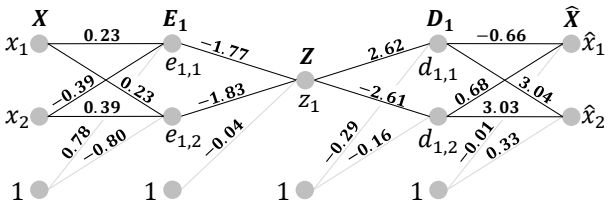


Figure D.12: Illustration of the 2-2-1-2-2 AE.

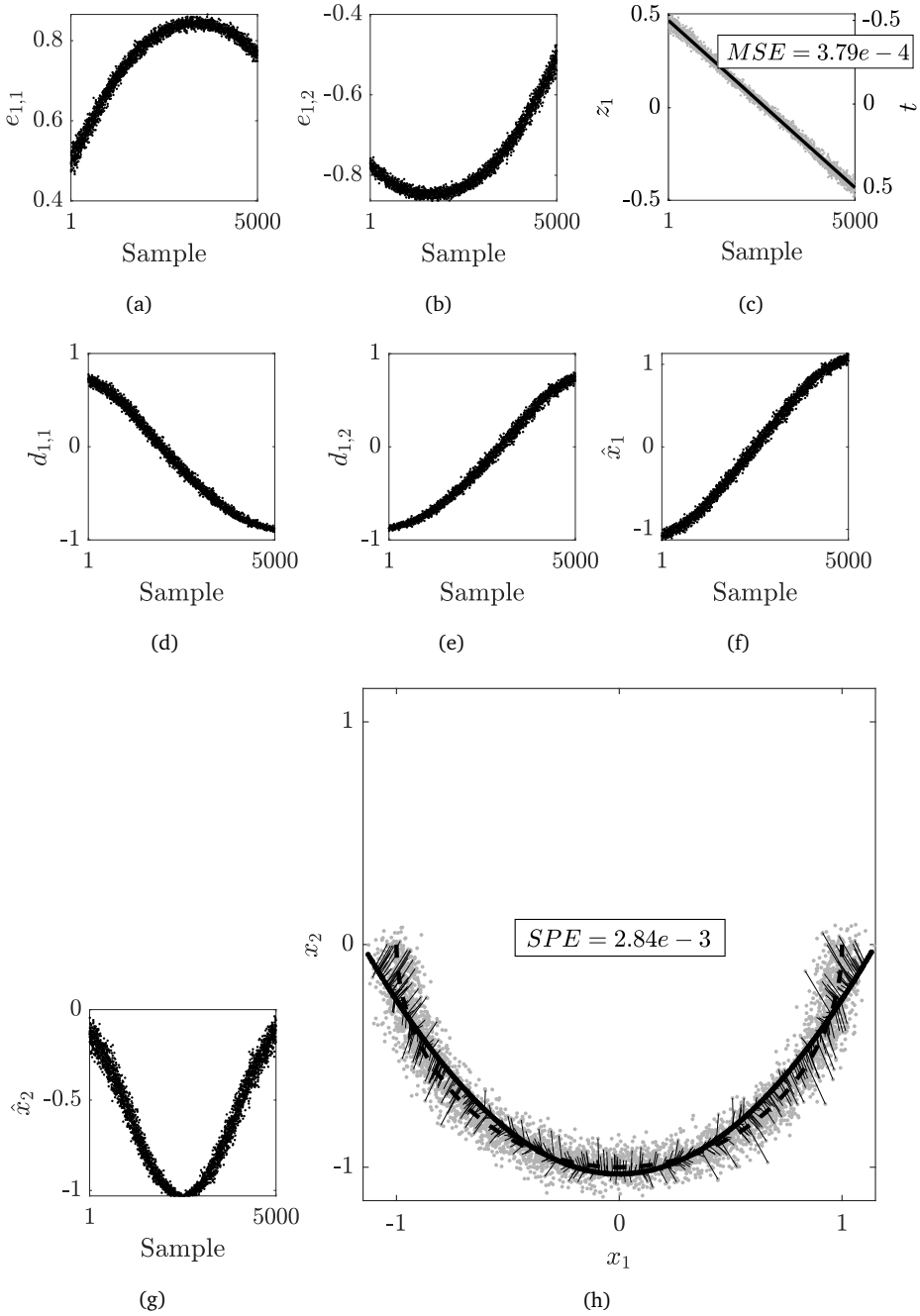


Figure D.13: From propagating validation set $\mathbf{X}_v$ through the 2-2-1-2-2 AE: (a)-(g) Activation of network neurons over samples and (h) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

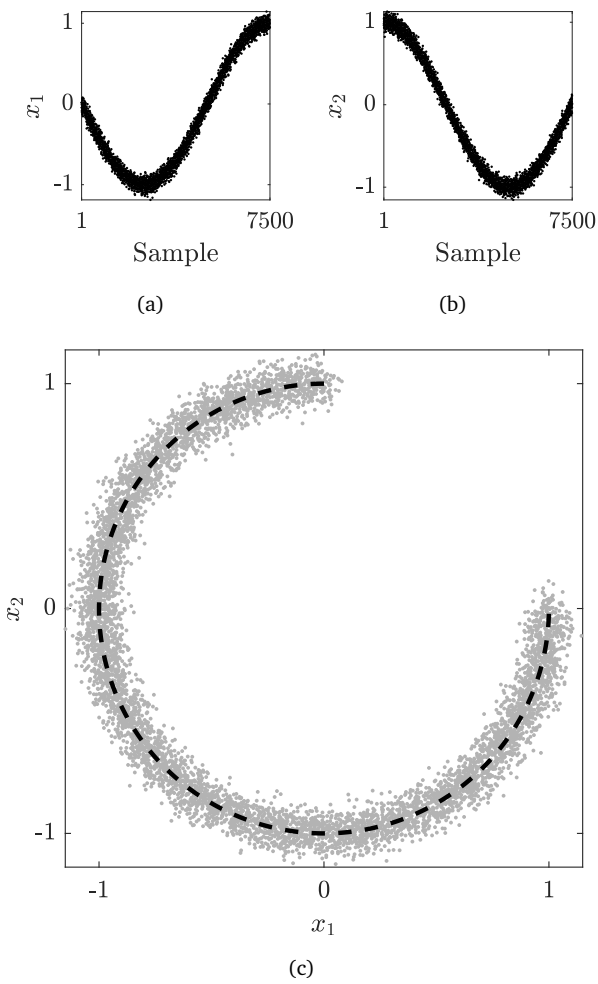


Figure D.14: Samples of (a) x_1 and (b) x_2 over the validation set $\mathbf{X}_v$, as well as a (c) scatter plot of the samples.

one-quarter of a circle. This augmentation adds additional complexity, as x_2 is now a multivalued variable for $x_1 < 0$.

Weights of a 2-2-1-2-2 AE optimized with the training set $\mathbf{X}_t$ are illustrated in Fig. D.15. Weight connections between $\mathbf{X}$ and $\mathbf{E}_1$ indicate a level of weight disparity in the encoding layer. The activation of layer $\mathbf{E}_1$ is dominated by the value of x_1, evident from the relatively large magnitude of connection between x_1 and $e_{1,1}$ compared to the remaining connections. This hints at a similar occurrence as in Fig D.6(a), where the complexity of the network does not meet the complexity of the data, so the AE compromises its functionality to satisfy the objective function.

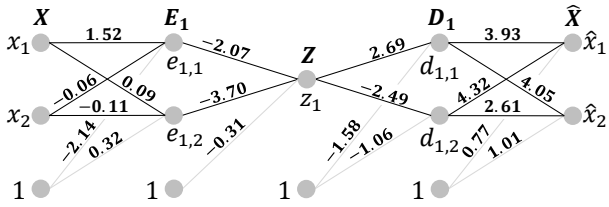


Figure D.15: Illustration of the 2-2-1-2-2 AE.

The effects of this are evident in the latent layer, illustrated by Fig. D.16(a). The latent activation z_1 is not summarized by a line over the validation set as the ends of z_1 curve away from the line. Also, z_1 becomes more stochastic at the end of the validation set. Fig. D.16(b) displays the scatter plot of the output of the AE. It can be seen that, similar to Fig. D.13(h), the best the AE can do is approximate the principal curve by a parabolla, generating many projections that are not orthagonal.

The dimension of the encoder and decoder hidden layers are now augmented with an additional node. Fig. D.17 depicts a 2-3-1-3-2 AE trained on $\mathbf{X}_t$. The figure

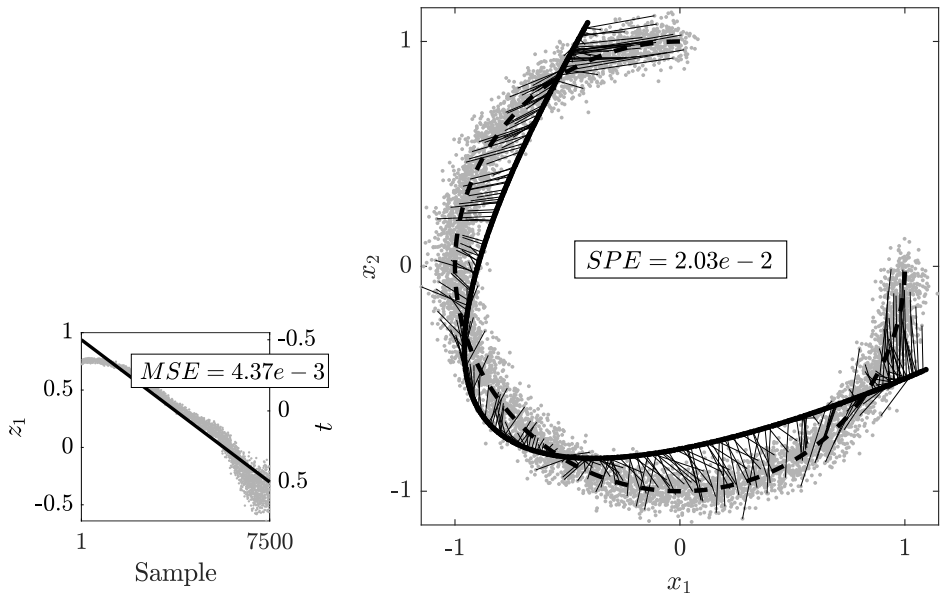


Figure D.16: From propagating validation set $\mathbf{X}_v$ through the 2-2-1-2-2 AE: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

shows that the weights disparity between $\mathbf{X}$ and $\mathbf{E}_1$ has been reduced from Fig. D.15 by adding an extra node. However, minor weight disparity remains. For example, the activation of $e_{1,1}$ will be lesser than $e_{1,2}$ and $e_{1,3}$ due to smaller weights. Furthermore, the activation of $e_{1,1}$ is dominated by x_1 since the weight between x_2 and $e_{1,1}$ is minuscule (-0.02) compared to the weight between x_1 and $e_{1,1}$ (0.18). Regardless, Fig. D.18(a) shows that the latent variable z_1 of the 2-3-1-3-2 AE resembles the underlying variable t more than z_1 of the 2-2-1-2-2 AE (Fig. D.16(a)), evident by a decrease in MSE. The output of the AE is visualized in Fig. D.18(b). Increasing the dimension of the AE's hidden layers produces a more accurate approximation for the principal curve.

D.6 Discussion

Fig. D.7 indicates that a correspondence between the latent variable z_1 and the underlying parameter t is not an indicator for reconstruction accuracy, which counters the claim made by Kramer [72]. In this paper, it is proposed that the similarity between z_1 and t_1 is an indicator for (a) how similar the distribution of observations is to the distribution of projections; and (b) the quality of consistent ordering along the projection curve.

The measure for how similar the distribution of observations is to the distribution of projections is demonstrated with Fig. D.19. This indicator corresponds to how well the mean of z_1 tracks t. Here, observations in $\mathbf{X}_v$ (the original validation set sampled from Eq. (D.6)) are propagated through the 2-1-2 AE in Fig. D.19(a). The weight between x_1 and z_1 is larger compared to the same weight in Fig. D.6(a), causing the latent variable to exist in the nonlinear region of the tangent hyperbolic. Consequently, the latent variable's activation is less distributed at the edges of the sample space and more distributed at its center. This effect propagates onto the reconstruction space as the distribution of reconstructions $\hat{x}_1$ and $\hat{x}_2$ is dense at the

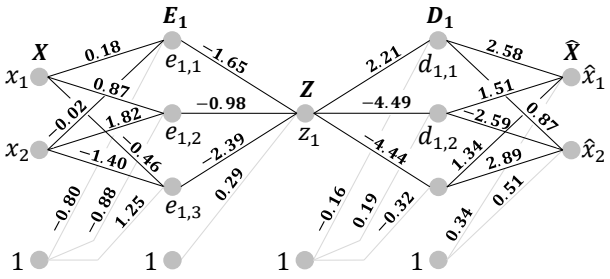
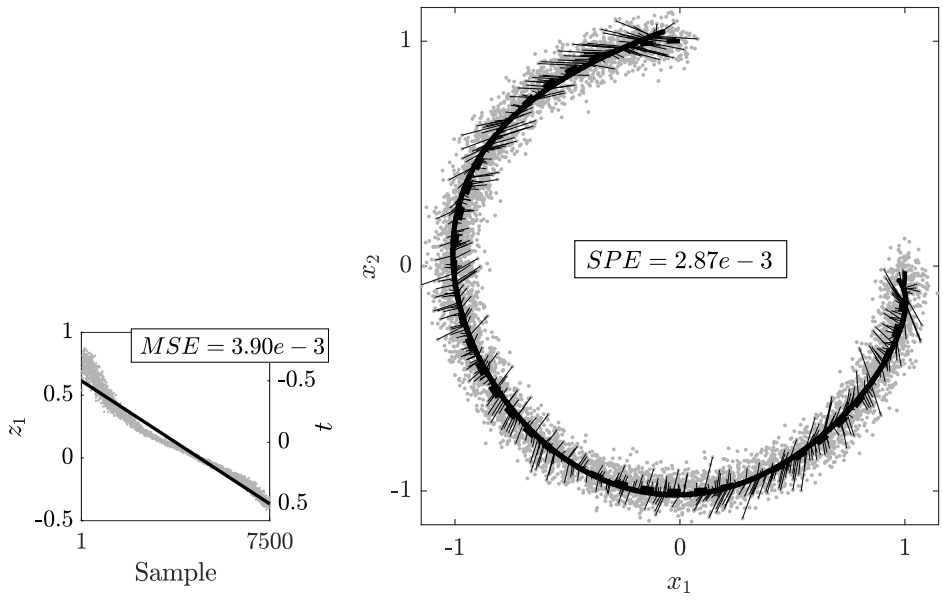


Figure D.17: Illustration of the 2-3-1-3-2 AE.



(a) (b)

Figure D.18: From propagating validation set $\mathbf{X}_v$ through the 2-3-1-3-2 AE: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

edges of the sample space and sparse at its center (Fig. D.19(b)).

The presence of a varying distribution in z_1 is observable in Fig. D.16. Here, the activation of z_1 remains relatively constant for the first few hundred samples. This indicates that no variation in z_1 is caused by the original variables x_1 and x_2. However, Fig. D.16(b) shows that there is a significant variance in x_1 for first few hundred samples. Because z_1 is relatively constant, the projections fall onto the same region of the projection curve.

The quality of consistent ordering along a projection curve corresponds to fluctuation in the latent variable z_1 that is not explained by stochasticity in the original variables x_1 and x_2. For instance, Fig. D.8(a) shows that the activation of z_1 is noisy over the validation set, yet Fig. D.8(b) shows that the projection points are ordered along the projection curve. Hence the fluctuation in z_1 is explained by stochasticity in x_1 and x_2. Fig. D.20 shows the result of propagating $\mathbf{X}_v$ through the 2-1-2 AE in Fig. D.20(a). The AE is equivalent to the AE in Fig. D.6(a) with the exception that z_1 includes additional noise introduced by the random variable $u = \mathscr{U}[-1,1]$. Naturally, an AE with a stochastic neuron is not proposed in practice, but the purpose is to analyse the effect of noise in z_1 on projections. The mean activation of z_1 is consistent

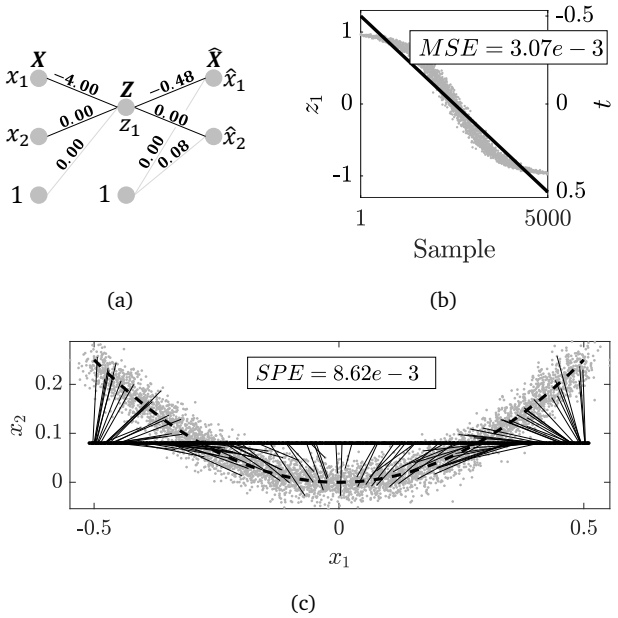


Figure D.19: From propagating validation set $\mathbf{X}_v$ through the 2-1-2 network: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

to that of Fig. D.7(a) but includes more noise. Because the distribution of z_1 remains uniform, the distribution of the reconstructions $\hat{x}_1$ and $\hat{x}_2$ remain uniform along the projection curve. However, the unexplained variation in z_1 induces a stochasticity in the reconstructions such that $\hat{x}_1$ and $\hat{x}_2$ are less orderly along the curve. Hence for this example, the quality of consistent ordering along the projection curve is low.

The presence of varying stochasticity in z_1 is visible in Fig. D.16. Fig. D.16(a) shows that the activation of z_1 becomes more stochastic towards the end of the validation set despite the level of noise in original variables x_1 and x_2 remaining constant. Consequently, projections become less orderly and more inconsistent in Fig. D.16(b) over the validation set. This occurs due to the optimized weights of the 2-2-1-2-2 AE in Fig. D.15.

It is proposed that these two measures, namely, how well the mean of z_1 corresponds to t and how much the variance in z_1 corresponds to the variance in the original variables, provides an indication for the quality of orthogonal projections to a projection curve.

The examples provided consisted of finding a principal curve for a two-dimensional data set. The same concepts conveyed in this paper apply for finding a principal

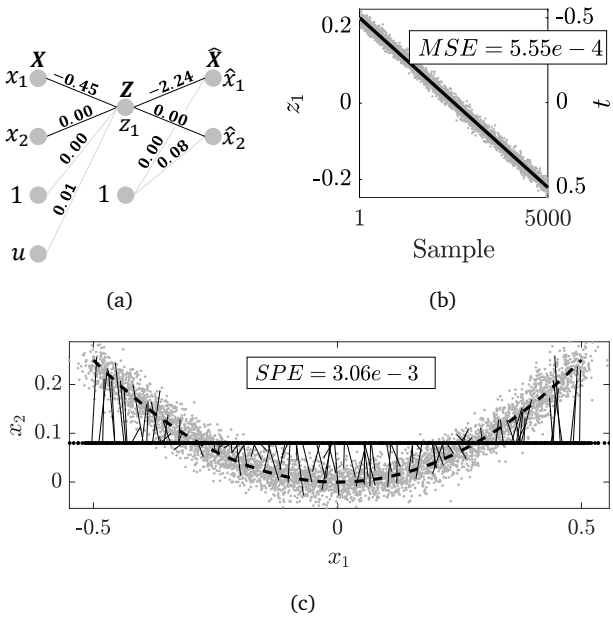


Figure D.20: From propagating validation set $\mathbf{X}_v$ through the 2-1-2 network: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

curve for any m-dimensional data set where $m > 2$. Furthermore, the definition of a principal curve can be extended to two-dimensional surfaces and other q-dimensional manifolds where $q > 2$ [47]. Estimating a higher-dimensional manifold requires configuring an AE with q nodes in its latent layer. Regardless of the choice for m and q, it is attested that an AE may require hidden layers with dimension larger than m for certain nonlinear variable spaces.

Figs. D.10(h) and D.18(b) show that the projection curves do not provide a precise estimate for the principal curves since the projection are not orthogonal. The imprecision may be reduced by a combination of (a) including more hidden encoder and decoder layers and (b) setting the dimension of hidden layers much larger than the original variable space. This gives an AE increased model complexity, thereby providing it with additional model parameters to precisely estimate a principal curve. Fig D.21 show the result of propagating the validation set through an 2-8-8-8-1-8-8-8-2 AE trained on the three-quarters training set; the large AE produces a more precise estimate of a principal curve.

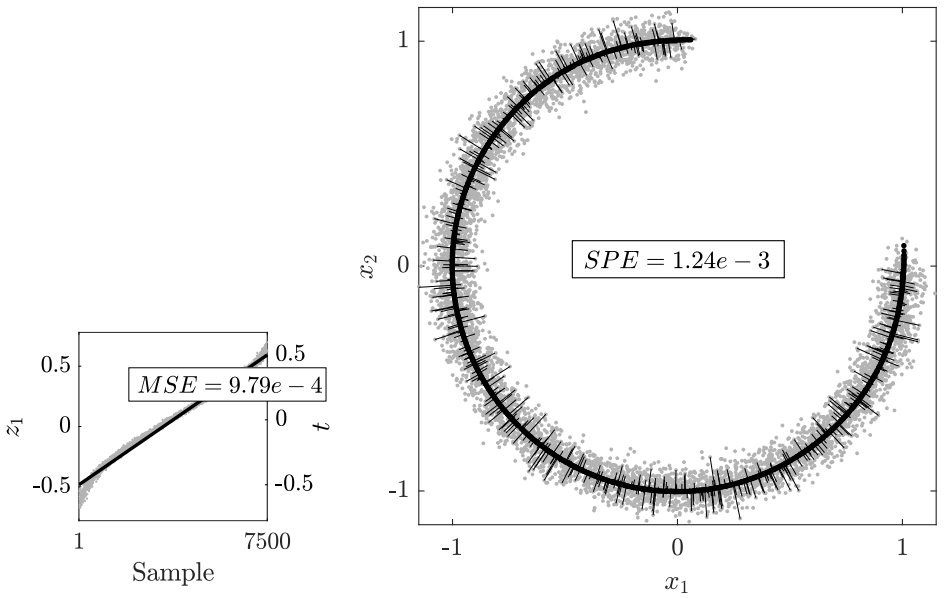


Figure D.21: From propagating validation set $\mathbf{X}_v$ through a 2-8-8-8-1-8-8-8-2 AE: (a) Activation of z_1 over samples and (b) scatter plot of (grey) original data and (black) reconstructions for x_1 and x_2.

D.7 Conclusion

This paper investigates the effects of model complexity on the performance of autoencoders. An analysis was conducted in terms of (a) an AEs ability to reproduce a latent variable analogues to the underlying variable of a data set; and (b) an AEs ability to estimate a principal curve that traverses through the center of a data set. Three two-dimensional data sets were provided for the analysis. When displayed on a two dimensional scatter plot, the variables of each data set exhibited a parabola, a half circle, and three quarters of a circle, respectively.

The results demonstrate that the common approach to AE-based dimensionality reduction, whereby no additional hidden layers are included, is insufficient for accurate reconstruction of the provided data sets. An AE employing no additional hidden layers is liable to reduce to a linear PCA model and ignore input variables. In the context of the parabolic data set, it was necessary to include an additional encoder and decoder layer with the same dimension as the original variable space to allow for an accurate reconstruction. In the context of the half circle and three quarters of a circle data sets, the results show that, although the AE included

additional hidden layers with the same dimension as the original variable space, the AE reconstructed each data set with a parabola. It was necessary to increase the two-dimensional hidden layers to three dimensions to accurately reconstruct the data. Collectively, the results show that it is possible to perform AE-based feature extraction for complex nonlinear data sets without major reformulation of the provided method but rather by simply including hidden layers with dimensions larger than the original variable space. Consequently, an AE employed for process monitoring will perform better at detecting process anomalies since it retains a suitable depiction of nominal process variables correlation.

The results demonstrate that an AEs ability produce a latent variable analogues to the underlying variable of a data set is not an indicator for reconstruction accuracy. Instead, it is proposed that the similarity provides an indication for the quality of orthogonal reconstructions.

Acknowledgments

The authors would like to acknowledge the support of the Danish Hydrocarbon Research and Technology Center (DHRTC) at the Technical University of Denmark.

Paper E

Detection of Abnormal Events in Dynamic Processes Using Recurrent Autoencoders

Ásgeir Daniel Hallgrímsson[1,*], Hans Henrik Niemann[1], Morten Lind[1]

[1]Automation and Control Group, Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark

Abstract:
Feature extraction methods employed for multivariate statistical process monitoring (MSPM) detect abnormal events (faults) in multivariate processes by comparing the correlation among new observations for process variables against the correlation among historical observations; a fault is concluded to have occurred if the disparity is significantly large. However, state-of-the-art feature extraction methods assume that variables are not cross-correlated; hence a static approximation that fails to summarize the dynamic variation among variables is provided for dynamic processes. In this paper, a recurrent autoencoder (RAE) is proposed for dynamic feature extraction. A regularized RAE is trained on historical process data sampled from a dynamic process and subsequently employed to detect faults from new observations. Regularization promotes a dynamic model structure that provides insight into the cross-correlation among variables. The proposed method for fault detection is demonstrated with a simulation of a linear and a nonlinear process. Performance of fault detection is compared against that of a conventional dynamic principal components analysis method.

*Corresponding author. E-mail: asdah@elektro.dtu.dk

E.1 Introduction

Multivariate statistical process monitoring (MSPM) is the act of monitoring a process with a statistical method that is independent of prior knowledge about the process [91]. Instead, the condition of the process is monitored by evaluating the multivariate statistical properties of new process observations and comparing them against available historical process data [63]. MSPM is found applicable when a large system consists of numerous, correlated process variables that may be summarized by a few, intrinsic variables [72]. Fundamentally, methods for MSPM are based on feature extraction: a numerical technique that exploits the correlation of a high-dimensional variable space to derive low-dimensional quantities intended to be informative about the original variables. Features are first extracted from a historical process data set sampled from when a process operated under nominal conditions. Process monitoring consists of comparing the features of new observations against the features of the historical data set, usually represented by a monitoring statistic. An anomaly, i.e., a fault, is concluded to have occurred if this monitoring statistic exceeds a predefined threshold.

Several feature extraction-based MSPM methods have been proposed in the literature. Principal component analysis (PCA) is a method for obtaining a linear transformation that projects data into a low-dimensional space [104]. The coefficients of the transformation are such that an inverse transformation of the projected data will reconstruct the original data with a minimum sum of squares difference (residual). Process monitoring consists of evaluating the variance of the projections and residual information of new process observations [63]. However, acceptable performance of fault detection with PCA requires that variables follows the assumption of multivariate normality. Independent component analysis (ICA) is employed if the normality assumption is not met [126]. ICA attempts to find a linear transformation that decomposes the non-Gaussian distributed variables into a set of mutually independent features. Similar to PCA, the transformation is inverted to reconstruct the original data for facilitating MSPM [84].

Both PCA and ICA are linear methods that perform poorly for nonlinear processes where process variables are nonlinearly correlated. Kernel PCA (KPCA) was proposed as a nonlinear extension of PCA [82], [119]. KPCA maps the original variable space to a high-dimensional space with a nonlinear kernel function, and then performs PCA in this high-dimensional space. Compared to other nonlinear methods, KPCA does not involve a nonlinear optimization; it essentially requires only linear algebra, making it relatively as simple as PCA. An example of a method requiring nonlinear optimization are autoencoders (AEs) [50]. They are a class of neural

networks configured for feature extraction. An AE contains a bottleneck layer that constricts the original variable space to a lower dimension. Optimization consists of learning (a) a nonlinear transformation to the bottleneck layer; and (b) a nonlinear transformation back to the original high-dimensional space that reconstructs the variables [18]. Although nonlinear optimization and the number of hyperparameters involved make AEs harder to implement, reports in the literature show that AEs can learn the transformations of PCA, ICA, and KPCA given certain optimization constraints, which makes them highly versatile [7], [67], [76].

A deficiency among common feature extraction methods for MSPM is the lack of focus on time dependence, i.e., the methods do not exploit any cross-correlation existing between process variables. It is not uncommon for process variables to be cross-correlated, and thus the inability of a static feature extraction method to exploit it poses several problems. Since static feature extraction cannot unveil the dynamic relationships existing between variables, the method may produce features that are auto-correlated and possible cross-correlated, making it difficult to infer the time-dependent relationship between variables from the model. Misleading results such as false alarms might be generated due to variable transients induced by disturbances and control input changes, since transients contain dynamic variations that are nominal but may be interpreted as abnormal by a static model.

Ku et. al [75] propose a dynamic PCA method that augments the original process variable vector with lagged versions of itself. Static PCA extract features from the augmented (dynamic) variable vector that are a function of cross-correlated process variables. The practice of constructing a dynamic variable vector may also be performed with ICA, KPCA, and AEs to obtain a dynamic formulation of said methods. However, this approach to dynamic feature extraction is not dynamic in a traditional sense, since the resulting models do not contain any internal states (memory). Rather, they mimic dynamic behavior by performing a simultaneous computation on current and past observations. An issue one might also consider is that the number of parameters required for a transformation of the dynamic variable vector increases as the number of lags increase, which could make model interpretation of the underlying cross-correlation structure difficult.

A recurrent neural network (RNN) is a neural network that includes internal states. This allows it to exhibit temporal dynamic behaviour. RNNs are used for sequential aplications such as time series prediction [27], speech recognition [43], and natural language processing [89]. An RNN is configured for feature extraction by including a bottleneck layer. Such an RNN is referred to as a recurrent AE (RAE). RAEs are applied for encoding tasks such as speech spectrogram compression [145], video game song compression [34], phrase representations for translation tasks

[25], [122], and video compression [88]. More recently, RAEs have been applied for abnormal event detection from videos [143] as well as nonlinear processes [22].

In this paper, an RAE is proposed for MSPM. The RAE is trained with a näive elastic net regularization constraint and a denoising criterion to promote an interpretable model structure. A comparison in terms of fault detection performance is performed against DPCA-based MSPM of a linear system. Furthermore, a nonlinear augmentation is introduced to the system to test the nonlinear capabilities of an RAE. The key result of this study was that RAEs offers superior fault detection performance compared to DPCA, and are capable of monitoring nonlinear processes.

The organization of the paper is as follows. Latent projection, which is a numerical method for performing feature extraction, as well as DPCA and RAEs are introduced in section 2. Section 3 presents the results from performing fault detection of a linear process and a nonlinear process. Section 4 provides a conclusion of the study.

E.2 Latent projection

Latent projection is a numerical method for performing feature extraction. The procedures consists of transforming a high-dimensional variable vector $\mathbf{x} \in \mathbb{R}^{m\times 1}$ to a smaller set of latent variables $\mathbf{z} \in \mathbb{R}^{q\times 1}$ (with $q < m$) that retain principal information about the original variables. The transformation is obtained via a machine learning approach. Initially, n observations of $\mathbf{x}$ are sampled to produce the reference data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$. The objective is to seek a transformation to the latent space $\mathbf{Z} \in \mathbb{R}^{q\times n}$ such that $\mathbf{Z}$ retains information about $\mathbf{X}$.

E.2.1 Principal component analysis

PCA is a linear method for obtaining a transformation with the least information loss. The procedure consists of finding a set of principal component scores $t_i = \mathbf{p}_i^\intercal \mathbf{x}$ for $i \in \mathbb{Z} : i \in [1, m]$ where (a) the column vectors $\mathbf{p}_i$ form the orthonormal principal component loading matrix $\mathbf{P} \in \mathbb{R}^{m\times m}$; and (b) the first principal component score t_1 has maximum variance, the second principal component score t_2 has the next greatest variance, with additional scores up to m similarly defined. Provided that $\mathbf{x}$ follows the multivariate normality assumption, the constraint that $\mathbf{P}$ is orthonormal ensures that the scores t_i are uncorrelated with (orthogonal to) one another. The scores t_i form the score vector $\mathbf{t} \in \mathbb{R}^{m\times 1}$, obtained via the transformation $\mathbf{t} = \mathbf{P}^\intercal \mathbf{x}$. Because $\mathbf{P}$ is orthonormal, i.e., $\mathbf{P}\mathbf{P}^\intercal = \mathbf{P}\mathbf{P}^{-1} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{x}$ is reproducible via the transformation $\mathbf{x} = \mathbf{P}\mathbf{t}$. The loading matrix $\mathbf{P}$ is obtained by

solving for the eigenvectors of the covariance matrix $\mathbf{\Sigma}$ of $\mathbf{x}$:

$$\mathbf{\Sigma} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{\intercal} \tag{E.1}$$

where $\mathbf{\Lambda}$ is a non-negative real diagonal $m \times m$ matrix whose diagonal elements are the corresponding eigenvalues. The diagonal entries λ_i of $\mathbf{\Lambda}$ are the variances of the scores t_i. If $\mathbf{\Sigma}$ is not known, then it may be estimated via $\mathbf{\Sigma} = \mathbf{X}^{\intercal}\mathbf{X}/(n-1)$.

Latent projection is performed by identifying q PCs that explain most of the variation in $\mathbf{x}$. The remaining $q-m$ PCs are associated with common cause variation. To that effect, $\mathbf{P}$ is partitioned as follows:

$$\mathbf{P} = \begin{bmatrix} \hat{\mathbf{P}} & \tilde{\mathbf{P}} \end{bmatrix}, \quad \hat{\mathbf{P}} \in \mathbb{R}^{m \times q}, \quad \tilde{\mathbf{P}} \in \mathbb{R}^{m \times (m-q)} \tag{E.2}$$

The variable vector $\mathbf{x}$ is then decomposed into the reconstruction vector $\hat{x}$ and residual vector $\tilde{x}$:

$$\begin{aligned} \mathbf{X} &= \hat{\mathbf{x}} + \tilde{\mathbf{x}} \\ &= \hat{\mathbf{P}}\mathbf{z} + \tilde{\mathbf{P}}\tilde{\mathbf{z}} \end{aligned} \tag{E.3}$$

where $\mathbf{z} = \hat{\mathbf{P}}^{\intercal}\mathbf{x}$ are the latent variables and $\tilde{\mathbf{z}} = \tilde{\mathbf{P}}^{\intercal}\mathbf{x}$ are the residual latent variables.

E.2.2 Dynamic principal component analysis

Consider the case that $\mathbf{x}$ consists of process variables sampled from a dynamic system. For a dynamic system, it is reasonable to assume that the values of $\mathbf{x}[k]$ at sample k depend on past values, i.e., they exhibit some degree of cross-correlation. Performing PCA on $\mathbf{x}$ will construct a linear static model that will not reveal the exact time-dependent relations between variables but rather a linear static approximation. Ku et al. [75] propose a dynamic formulation of PCA called dynamic PCA (DPCA). In a general case, the following historical variable vector for process variable $i \in \mathbb{Z} : i \in [1,m]$ is defined:

$$\mathbf{x}_i^l[k] = \begin{bmatrix} x_i[k] & x_i[k-1] & \dots & x_i[k-l] \end{bmatrix}^{\intercal} \tag{E.4}$$

The dynamic process variable vector is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^l[k] & \mathbf{x}_2^l[k] & \dots & \mathbf{x}_m^l[k] \end{bmatrix}^{\intercal} \tag{E.5}$$

Performing latent projection follows the same procedure as performing latent projection with PCA.

E.2.3 Recurrent autoencdoers

A recurrent autoencoder (RAE) is a type of artificial neural network configured for latent projection. An RAE consists of two parts, an encoder and a decoder.

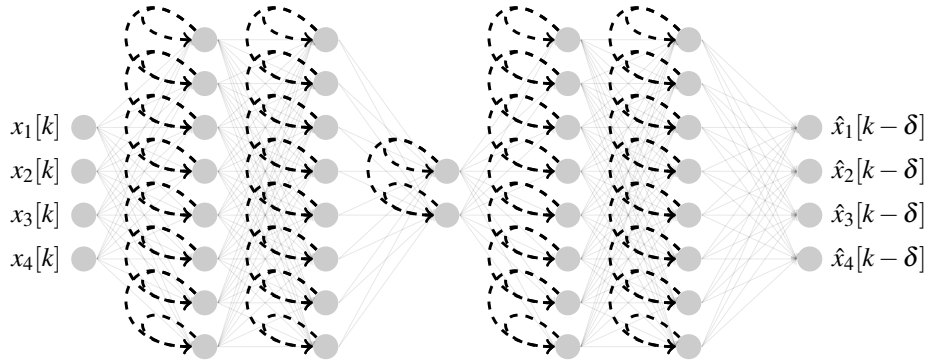


Figure E.1: Illustration of a 4-8-8-2-8-8-4 RAE. The solid edges represent feedforward weight connections, while dashed edges represent recurrent weight connections. Biases as well as some recurrent weight connections are omitted to enhance image clarity.

The encoder maps the original variables $\mathbf{x}$ to the latent variables $\mathbf{z}$. The decoder reconstructs, i.e., estimates, the reconstructions $\hat{\mathbf{x}}$ with a transformation of $\mathbf{z}$. The RAE's model parameters are optimized such that the difference between $\mathbf{x}$ and $\hat{\mathbf{x}}$ is minimized. This ensures that $\mathbf{z}$ retains the salient information about $\mathbf{x}$ required to reconstruct it.

A RAE contains an internal state (memory). This allows it to exhibit temporal dynamic behavior when processing its input. This makes a RAE applicable for performing latent projection when $\mathbf{x}$ consists of dynamic process variables. Figure E.1 shows an illustration of a RAE, along with labels to the encoder and decoder parts. The RAE is composed of several vectors of nodes, known as network layers. Each node represents a time-varying real-valued activation. With the exception of the input layer, each layer at sample k is a component wise nonlinear function of the sum of (a) a linear transformation of its previous layer; and (b) a linear transformation of its own layer at the previous sample $k-1$. The encoder maps the input $\mathbf{x}$ to the latent variables $\mathbf{z}$:

$$\mathbf{e}_i[k] = \begin{cases} \sigma_1^e\left(\mathbf{W}_1^e\mathbf{x}[k] + \mathbf{R}_1^e\mathbf{e}_1[k-1] + \mathbf{b}_1^e\right), & \text{for } i = 1 \\ \sigma_i^e\left(\mathbf{W}_i^e\mathbf{e}_{i-1}[k] + \mathbf{R}_i^e\mathbf{e}_i[k-1] + \mathbf{b}_i^e\right), & \text{else} \end{cases} \tag{E.6}$$

$$\mathbf{z}[k] = \sigma^z\left(\mathbf{W}^z\mathbf{e}_N[k] + \mathbf{R}^z\mathbf{z}[k-1] + \mathbf{b}^z\right)$$

where $i \in \mathbb{Z} : i \in [1, N]$. $\mathbf{W}_1^e$ is the feedforward weight matrix between the input layer and the first encoder layer. $\mathbf{W}_i^e$ is the feedforward weight matrix between layers $i-1$

and i, $\mathbf{R}_i^e$ is the recurrent weight matrix for layer i between samples k and $k-1$, $\mathbf{b}^e$ is the bias at layer i, and σ_i^e is the component wise activation function at layer i. $\mathbf{W}^z$, $\mathbf{R}^z$, $\mathbf{b}^z$, and σ^z are defined similarly for the latent layer. The decoder maps the latent variables $\mathbf{z}$ to the reconstructions $\hat{\mathbf{x}}$:

$$\begin{aligned}\mathbf{d}_j[k] &= \begin{cases}\sigma_1^d\left(\mathbf{W}_1^d\mathbf{z}[k]+\mathbf{R}_1^d\mathbf{d}_1[k-1]+\mathbf{b}_1^d\right), & \text{for } j=1\\ \sigma_j^d\left(\mathbf{W}_j^d\mathbf{d}_{j-1}[k]+\mathbf{R}_j^d\mathbf{d}_j[k-1]+\mathbf{b}_j^d\right), & \text{else}\end{cases}\\ \hat{\mathbf{x}}[k-\delta] &= \sigma^{\hat{x}}\left(\mathbf{W}^{\hat{x}}\mathbf{d}_M[k]+\mathbf{b}^{\hat{x}}\right)\end{aligned} \tag{E.7}$$

where $j \in \mathbb{Z} : j \in [1, M]$. $\mathbf{W}_1^d$ is the feedforward weight matrix between the latent layer and the first decoder layer. $\mathbf{W}_j^d$ is the feedforward weight matrix between layers $j-1$ and j, $\mathbf{R}_j^d$ is the recurrent weight matrix for layer j between samples k and $k-1$, $\mathbf{b}^d$ is the bias at layer j, and σ_j^d is the component wise activation function at layer j. $\mathbf{W}^{\hat{x}}$,$\mathbf{b}^{\hat{x}}$, and $\sigma^{\hat{x}}$ are defined similarly for the output layer.

Note that unlike previous network layers, the reconstructions $\hat{\mathbf{x}}$ do not form any recurrent weight connections with themselves. In addition, the RAE is configured to reconstruct $\hat{\mathbf{x}}$ with a delay of δ time steps. In other words, the RAE reconstructs the original variables at sample $k-\delta$ from the projection of the kth sample of $\mathbf{x}$. The motivation for this design is explained in section 2.5.

The modifiable parameters $\mathbf{W}_i^e$, $\mathbf{R}_i^e$, $\mathbf{b}_i^e$, $\mathbf{W}^z$, $\mathbf{b}^z$, $\mathbf{W}_j^d$, $\mathbf{R}_j^d$, $\mathbf{b}_j^d$, $\mathbf{W}^{\hat{x}}$, and $\mathbf{b}^{\hat{x}}$ are optimized with a machine learning task. The parameters are initially randomized. For a single training update, each sample $\mathbf{x}$ in the reference matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ is corrupted by adding an uncorrelated noise vector $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$. The motive for this is explained in section 2.6. The corrupted sample $\tilde{\mathbf{x}}[k]$ is propagated through the RAE to produce the latent variable $\mathbf{z}[k]$ and δ-shifted reconstruction $\hat{\mathbf{x}}[k-\delta]$, which are gathered in the latent matrix $\mathbf{Z} \in \mathbb{R}^{q\times n}$ and the δ-shifted reconstruction matrix $\hat{\mathbf{X}}_\delta \in \mathbb{R}^{m\times n}$, respectively. The following loss function is defined:

$$\mathscr{L}(\mathbf{X}_\delta, \hat{\mathbf{X}}_\delta) = \frac{1}{n}\left|\left|\mathbf{X}_\delta - \hat{\mathbf{X}}_\delta\right|\right|^2 \tag{E.8}$$

where the δ-shifted reference matrix $\mathbf{X}_\delta$ consists of the original, i.e., uncorrupted, samples $\mathbf{x}_\delta[k] = \mathbf{x}[k-\delta]$ that have been shifted by δ time steps. This is to ensure that the rows of $\hat{\mathbf{X}}_\delta$ and $\mathbf{X}_\delta$ match. The RAE's model parameters are optimized by minimizing the loss calculated with Eq. (E.8) via stochastic gradient descent [99], thereby finishing the training update. The process is repeated until Eq. (E.8) is minimized satisfactorily.

E.2.3.1 Regularization and network sparsity

Regularization prevents over-fitting of a neural network. One approach is to augment the network's optimization function with the näive elastic net weight decay penalty. It is a regularization method that linearly combines the L_1 and L_2 penalties of the LASSO and ridge methods [154]. The loss function in Eq. (E.8) becomes:

$$\mathscr{L}(\mathbf{X},\hat{\mathbf{X}},\mathbf{W}) = \frac{1}{n}\left|\left|\mathbf{X}_\delta - \hat{\mathbf{X}}_\delta\right|\right|^2 + \lambda_1 ||\mathbf{W}||_1 + \lambda_2 ||\mathbf{W}||_2^2 \tag{E.9}$$

where $\mathbf{W} = \{\mathbf{W}_i^e, \mathbf{R}_i^e, \mathbf{W}_j^d, \mathbf{R}_j^d\}$ is the collection of feed-forward and recurrent weight matrices in the RAE, and λ_1 and λ_2 control the importance of the LASSO and ridge regressions, respectively. Note that the biases in Eqs. (E.6) and (E.7) are not included in regularization.

In addition to learning an optimal encoding/decoding transformation, minimization of Eq. (E.9) will shrink redundant weights in the RAE to form a small number of high-importance weight connections. Low-importance connections are pruned away to produce a sparse network, thereby producing an interpretable variable grouping effect.

E.2.4 Online process monitoring

Having built either a DPCA or RAE model on historical data $\mathbf{X}$ collected from a process operating under nominal conditions, new observations $\mathbf{x}_{new}$ can be referenced against this "in-control" model. The reconstructions $\hat{x}_{new}$ are referenced against the original observations to produce the residuals $\mathbf{e}_{new} = \mathbf{x}_{new} - \hat{\mathbf{x}}_{new}$. The squared prediction error (SPE), otherwise known as the Q statistic, is computed to monitor the quality of the residuals $\mathbf{e}_{new}$:

$$SPE = \sum_{i=1}^{m} (x_{new,i} - \hat{x}_{new,i})^2 \tag{E.10}$$

An abnormal event whose response was not present in $\mathbf{X}$ will generate an increase in the SPE. Assuming that the SPE follows a Chi-squared distribution, a control limit is computed with the approximate value [12]:

$$CL_{SPE_{AE}} = \frac{\bar{\sigma}^2}{2\bar{\mu}} \chi^2_{(2\bar{\mu}^2/\bar{\sigma}^2,\alpha)} \tag{E.11}$$

where $\bar{\mu}$ and $\bar{\sigma}$ are the sample mean and sample standard deviation of the SPE and α is the false alarm rate, respectively. $\bar{\mu}$ and $\bar{\sigma}$ may be estimated from the SPE time series for a new historical data matrix $\mathbf{X}_v$ sampled from a nominal process. An abnormal event is concluded to have occurred if the SPE exceeds its control limit, signifying that the "in-control" model does not apply for the new observations.

E.2.5 Motive for reconstruction delays

It is necessary to delay the sample-wise RAE reconstructions $\hat{x}_i$ by δ time steps to model cross-correlations between variables. Consider the following linear process:

$$\begin{aligned} t[k] &\in \mathscr{U}(0,1) \\ x_1[k] &= t[k] + \mathscr{N}(0,0.1) \\ x_2[k] &= t[k] + \mathscr{N}(0,0.1) \\ x_3[k] &= t[k-\tau] + \mathscr{N}(0,0.1) \end{aligned} \tag{E.12}$$

where $\mathscr{U}(a,b)$ denotes the uniform distribution in the range (a,b) and $\mathscr{N}(\mu,\sigma^2)$ denotes the normal distribution with mean μ and variance σ^2. Eq. (E.12) indicates that $x_1[k]$ and $x_2[k]$ are correlated, while $x_1[k]$ and $x_2[k]$ are cross-correlated with $x_3[k-\tau]$ for the lag perimeter τ.

Consider the 3-4-1-4-3 RAE shown in Fig. E.2(a), where the reconstructions are not delayed, i.e., $\delta = 0$. The RAE is trained on data sampled from Eq. (E.12) with $\tau = 1$. The sparse structure of the RAE is a result of the pruning strategy. The figure shows that the latent variable $z_1[k]$ is a function of $x_1[k]$ and $x_2[k]$ but not of $x_3[k]$. This suggests that the encoder of the RAE does not identify the cross-correlation between the variables when reducing their dimensionality. In fact, the RAE has learned to estimate $\hat{x}_1[k]$ based on the correlation between $x_1[k]$ and $x_2[k]$, since $\hat{x}_3[k]$ is a single-lagged function of $z_1[k]$. The RAE does not perform its intended feature extraction task and instead performs a regression on $x_3[k]$.

Fig. E.2(b) shows a 3-4-1-4-3 RAE where the reconstructions are delayed with $\delta = 1$. The network is trained with the same data as the RAE in Fig. E.2(a) along with the pruning strategy. The recurrent weight connection in the encoder part of the RAE indicates that $z_1[k]$ is a function of $x_1[k-1]$, $x_2[k-1]$, and $x_3[k]$; the RAE has learned to delay $x_1[k]$ and $x_2[k]$ by a single sample in order to identify the cross-correlation between the variables. The figure shows that $\hat{x}_1[k-1]$ and $\hat{x}_2[k-1]$ are a zero-lagged function of $z_1[k]$, while $\hat{x}_3[k-1]$ is a single-lagged function of $z_1[k]$. In effect, delaying the reconstructions ensures that the RAE retains all of its original variables.

It is not a necessary requirement that the reconstruction delay δ is equivalent to the lag parameter τ. Figure E.2(c) shows a 3-4-1-4-3 RAE where the reconstructions are delayed with $\delta = 2$. The network is trained with the same data as the RAEs in figures E.2(a) and E.2(b). The RAE in figure E.2(c) has an identical structure as the RAE in figure E.2(b) with the exception that the decoder in figure E.2(c) employs an additional delay before reconstructing the variables. In other words, choosing a larger δ will require more recurrent weight connections in order to perform the reconstructions, though this will not alter the RAE structure required to model variable cross-correlations.

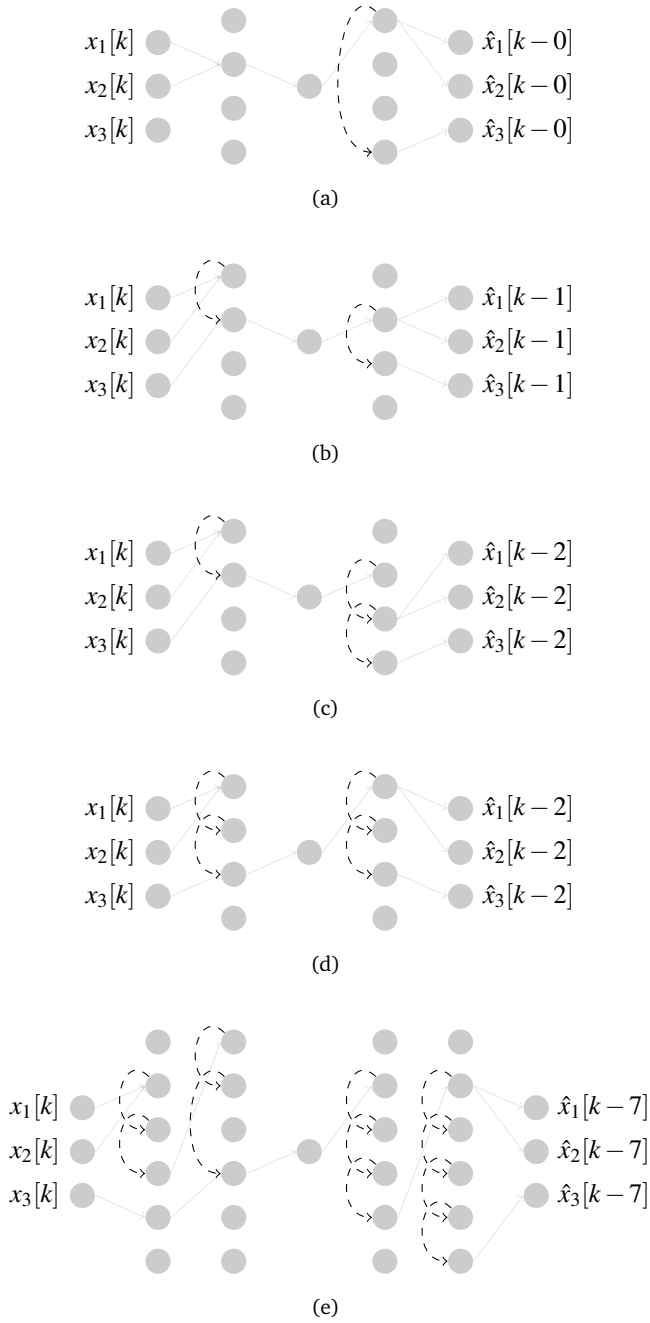


Figure E.2: Sparse connections of RAEs trained on data sampled from Eq. (E.12) where (a) $\tau = 1$ and $\delta = 0$, (b) $\tau = 1$ and $\delta = 1$, (c) $\tau = 1$ and $\delta = 1$, (d) $\tau = 2$ and $\delta = 2$, (e) $\tau = 4$ and $\delta = 7$. Solid edges indicate a feed-forward weight connection. Dashed edges indicate a recurrent weight connection. All network nodes employ linear activation functions.

Figure E.2(d) shows a 3-4-1-4-3 RAE trained on data sampled from Eq. (E.12) with $\tau = 2$. The reconstructions are delayed with $\delta = 2$. The figure shows that (a) the encoder models the cross-correlation between variables by setting $z_1[k]$ as a function of $x_1[k-2]$, $x_2[k-2]$, and $x_3[k]$; and (b) the decoder reconstructs $\hat{x}_1[k-2]$ and $\hat{x}_2[k-2]$ as a zero-lagged function of $z_1[k]$ and reconstructs $\hat{x}_3[k-2]$ as a double-lagged function of $z_1[k]$. From a structural perspective, it can be seen that the RAE in figure E.2(d) is very similar to the RAE in figure E.2(c), with differences between the two being a result of the difference in τ.

The examples presented so far demonstrate that τ and δ play an important role in the optimized weight structure of an RAE. For example, a large τ requires more recurrent weight connections in the encoder in order to identify the cross-correlation between variables. Similarly, a large δ requires more recurrent weight connections in the decoder in order to appropriately delay the reconstructions. Ultimately, a large RAE may be required for a system of few variables if the selection for τ and δ is large. Fig. E.2(e) shows a 3-6-6-1-6-6-3 RAE with $\delta = 7$ trained on data sampled from Eq. (E.12) with $\tau = 4$. The figure shows that increasing the dimension and number of hidden layers is well justified by the number of recurrent weight connections required to appropriately compute the latent variable $z_1[k]$ and reconstruct the variables.

What constitutes as a "sufficiently large" RAE is up to the judgment of the analyst. Fortunately, what the examples show is that one may propose an "abundantly large" RAE. For instance, a 3-100-100-1-100-100-3 RAE with $\delta = 30$ could instead be proposed for the data used to train the RAE in Fig. E.2(e). Both RAEs would require the same number of recurrent weight connections to compute $z_1[k]$, but the final structural difference would be that the larger RAE will employ more recurrent weight connections in order to delay the reconstructions. Training and analysis will be more computationally expensive, but one will be certain that the RAE performs its intended task and not run into the problems exemplified with Fig. E.2(a). However, one drawback of a large δ is that it takes more samples for input variations to show up at the reconstructions. This is unideal from a fault detection perspective, as it will take δ time steps for the fault to be observable in the reconstruction.

E.2.6 Motive for sample corruption

It is necessary to corrupt the training samples in the reference matrix $\mathbf{X}$ with uncorrelated normally distributed noise in order to prevent a regularized RAE from learning to reconstruct $\hat{\mathbf{x}}$ from a subset of the original variables $\mathbf{x}$. Consider the

following nonstochastic linear process:

$$\begin{aligned} t[k] &\in \mathscr{U}(0,1) \\ x_1[k] &= t[k] \\ x_2[k] &= x_1[k-2] \end{aligned} \tag{E.13}$$

Eq. (E.13) states that $x_1[k]$ and $x_2[k-\tau]$ are perfectly cross-correlated for the lag parameter $\tau = 2$. That is, knowing the value of one variable exactly predicts the sample-lagged value of the other variable.

Consider the 2-4-1-4-2 RAE shown in Fig. E.3(a) where the reconstructions are delayed with $\delta = 2$. The RAE is trained on data sampled from Eq. (E.12). Its sparse structure is a result of regularization. The figure shows that the latent variable $z_1[k]$ is a function of $x_2[k]$ but not of $x_1[k]$. This suggests that the encoder does not identify the cross-correlation between the variables when performing feature extraction. As a whole, the RAE has learned to reconstruct $\hat{x}_1[k-2]$ from $x_2[k]$. The cause for this is weight regularization in Eq. (E.9); since x_1 and x_2 are perfectly cross-correlated, an optimized RAE will shrink any redundant weights connecting x_1 to z_1 and simply reconstruct $\hat{x}_1[k-2]$ from $x_2[k]$.

Consider the following corrupted version $\tilde{\mathbf{x}}$ of the original variables $\mathbf{x}$:

$$\begin{aligned} \tilde{x}_1[k] &= x_1[k] + \mathscr{N}(0, 0.2) \\ \tilde{x}_2[k] &= x_2[k] + \mathscr{N}(0, 0.2) \end{aligned} \tag{E.14}$$

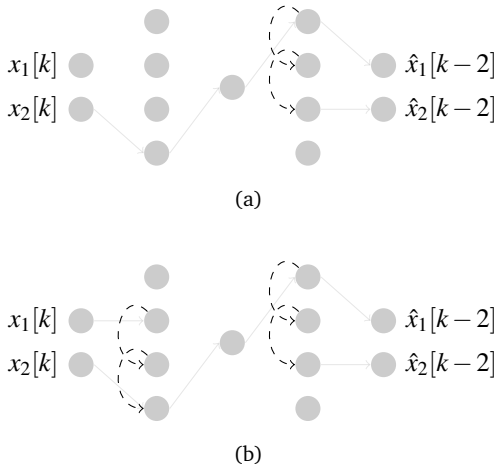


Figure E.3: Sparse connections of RAEs trained on data sampled from Eq. (E.13) where (a) inputs are uncorrupted and (b) inputs are corrupted. Dashed edges indicate a recurrent weight connection. All network nodes employ linear activation functions.

Eq. (E.14) reduces the correlation between the two variables by corrupting them with uncorrelated noise. The act of corrupting the input of an RAE is referred to as augmenting the RAE with a denoising criterion [130]. Vincent et. al [130] refer the act of corrupting the input of an RAE as *augmenting the RAE with a denoising criterion*. The denoising criterion is shown to guide the learning of useful higher level representations. Consider the 2-4-1-4-2 RAE shown in Fig. E.3(b). The RAE is trained on data sampled from Eq. (E.12) and (E.14), where the input is the corrupted $\tilde{\mathbf{x}}[k]$ and the labels for the reconstructions in Eq. (E.9) are the original variables $\mathbf{x}[k-2]$. The figure shows that z_1 is a function of both x_1 and x_2. This suggests that the denoising criterion guides the encoder to identify the cross-correlation between the variables when reducing their dimensionality.

E.3 Case studies

Two case studies are presented in this section. The first case study is a linear process from the literature. The second case study is a nonlinear augmentation of the linear process.

E.3.1 Comparison between DPCA and RAE

Consider the following linear process presented by Ku et al. [75]:

$$\begin{aligned}\mathbf{z}[k] &= \begin{bmatrix}0.118 & -0.191\\ 0.847 & 0.264\end{bmatrix}\mathbf{z}[k-1]\\ &\quad+\begin{bmatrix}1 & 2\\ 3 & -4\end{bmatrix}\left(\mathbf{u}[k-1]+\begin{bmatrix}f\\ 0\end{bmatrix}\right)\\ \mathbf{y}[k] &= \mathbf{z}[k]+\mathbf{v}[k]\end{aligned} \tag{E.15}$$

where $\mathbf{u}$ is the correlated control input:

$$\begin{aligned}\mathbf{u}[k] &= \begin{bmatrix}0.811 & -0.226\\ 0.477 & 0.415\end{bmatrix}\mathbf{u}[k-1]\\ &\quad+\begin{bmatrix}0.193 & 0.689\\ -0.320 & -0.749\end{bmatrix}\mathbf{w}[k-1]\end{aligned} \tag{E.16}$$

Variable f is a process fault that induces a bias in the control input. The source variable $\mathbf{w}$ and measurement noise $\mathbf{v}$ follow the following Gaussian distributions:

$$\mathbf{v} \in \mathcal{N}(0,0.1), \quad \mathbf{w} \in \mathcal{N}(0,1) \tag{E.17}$$

Both input $\mathbf{u}$ and output $\mathbf{y}$ are observable process variables but $\mathbf{z}$ and $\mathbf{w}$ are not. Following the DPCA method, Ku et al. [75] define the following dynamic process

variable vector:

$$\mathbf{x}_1 = \begin{bmatrix} y_1[k-0] \\ y_1[k-1] \\ y_2[k-0] \\ y_2[k-1] \\ u_1[k-0] \\ u_1[k-1] \\ u_2[k-0] \\ u_2[k-1] \end{bmatrix} \tag{E.18}$$

Vector $\mathbf{x}_1$ is sampled to produce the training set $\mathbf{X}_1^t$ (consisting of 3000 samples), validation set $\mathbf{X}_1^v$ (consisting of 1000 samples), and fault set $\mathbf{X}_1^f$ (consisting of 1000 samples with a step change in f introduced at sample 50). The matrices $\mathbf{X}_1^t$, $\mathbf{X}_1^v$, $\mathbf{X}_1^f$ are standardized with the mean and standard deviation of $\mathbf{X}_1^t$. A DPCA model is built for $\mathbf{X}_1^t$. Five out of the eight available principal components are chosen as instructed in [75].

Following the RAE method proposed in this paper, the following process variable vector:

$$\mathbf{x}_2 = \begin{bmatrix} y_1[k-0] \\ y_2[k-0] \\ u_1[k-0] \\ u_2[k-0] \end{bmatrix} \tag{E.19}$$

is defined. Vector $\mathbf{x}_2$ is sampled to produce the training set $\mathbf{X}_2^t$ (consisting of 3000 samples), validation set $\mathbf{X}_2^v$ (consisting of 1000 samples), and fault set $\mathbf{X}_2^f$ (consisting of 1000 samples with a step change in f introduced at sample 50). The matrices $\mathbf{X}_2^t$, $\mathbf{X}_2^v$, $\mathbf{X}_2^f$ are standardized with the mean and standard deviation of $\mathbf{X}_2^t$. A 4-10-1-10-4 RAE is trained with $\mathbf{X}_2^t$ with $\delta = 3$. Linear activation functions are employed at each layer.

Figure E.4(a) shows the structurally pruned 4-10-1-10-4 RAE that resulted from optimizing it with $\mathbf{X}_2^t$. Figure E.4(b) plots the training loss (TL) and validation loss (VL) observed over the RAE's training period. It is evident that the TL and VL observe a difference that dissipates once the network is pruned and the näive elastic net weight penalty is removed from the TL.

Figures E.5(a) and E.5(b) show a time series plot and histogram plot, respectively, of the SPE obtained from propagating $\mathbf{X}_1^v$ and $\mathbf{X}_1^f$ through the DPCA model. Figures E.5(c) and E.5(d) show equivalent plots of the SPE obtained from propagating $\mathbf{X}_2^v$ and $\mathbf{X}_2^f$ through the linear RAE model. The figures indicate that there is a larger difference between nominal and abnormal samples for the RAE's SPE compared to the DPCA's SPE. This indicates that the RAE's SPE is more sensitive to the fault f.

In other words, the RAE is more suitable for distinguishing between nominal and abnormal observations and thus provides better fault detectability. Furthermore, some abnormal samples when propagated through the DPCA produced an SPE with a magnitude less than the 95% control limit (figs. E.5(a) and E.5(b)). This did not occur for the RAE (figs. E.5(c) and E.5(d)). This indicates that the DPCA model occasionally incorrectly classifies abnormal samples as nominal.

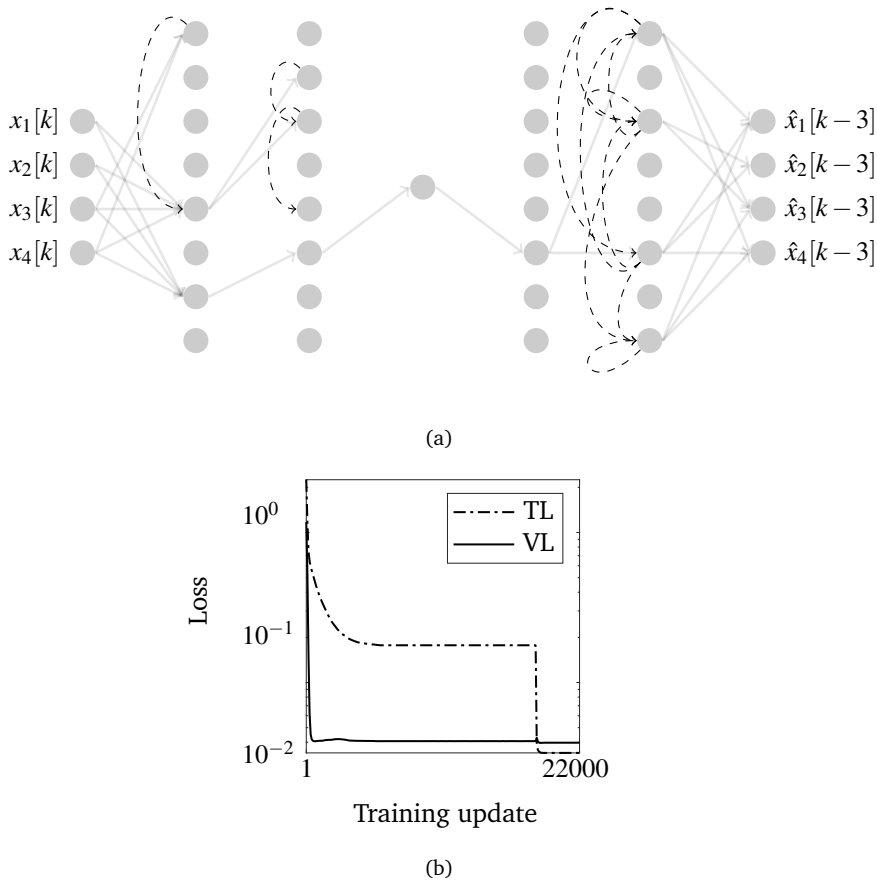


Figure E.4: (a) Resulting RAE, showing feedforward and recurrent weight connections. (b) Evolution of loss terms.

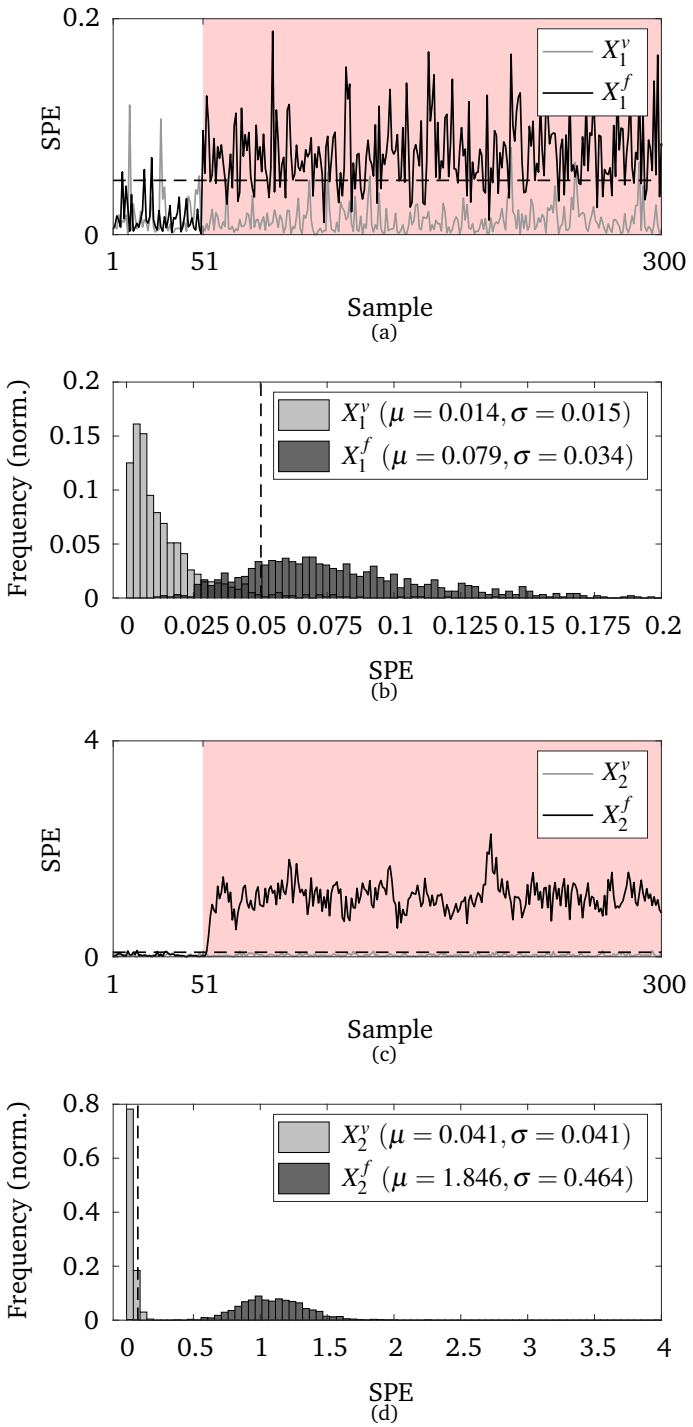


Figure E.5: SPE for (a),(b) the DPCA model and (c),(d) the linear RAE model. All plots include a 95% control limit (dashed line) determined from their respective validation set.

E.3.2 Comparison between linear RAE and nonlinear RAE

The process in Eq. (E.15) was made nonlinear by augmenting the influence of the process inputs $\mathbf{u}$ on the process states $\mathbf{z}$:

$$\begin{aligned}\mathbf{z}[k] &= \begin{bmatrix}0.118 & -0.191\\ 0.847 & 0.264\end{bmatrix}\mathbf{z}[k-1] \\ &\quad + \begin{bmatrix}1 & 2\\ 3 & -4\end{bmatrix}\left(\mathbf{u}[k-1] + \begin{bmatrix}f_1\\ 0\end{bmatrix}\right)^2 \\ \mathbf{y}[k] &= \mathbf{z}[k] + \mathbf{v}[k]\end{aligned} \tag{E.20}$$

where $\mathbf{u}$ remains as the correlated control input defined in Eq. (E.16). The measurement noise $\mathbf{v}$ and source variable $\mathbf{w}$ maintain their definitions in Eq. (E.17). Following the RAE method proposed in this paper, the following process variable vector:

$$\mathbf{x}_3 = \begin{bmatrix}y_1[k-0]\\ y_2[k-0]\\ u_1[k-0]\\ u_2[k-0]\end{bmatrix} \tag{E.21}$$

is defined. Vector $\mathbf{x}_3$ is sampled to produce the training set $\mathbf{X}_3^t$ (consisting of 3000 samples), validation set $\mathbf{X}_3^v$ (consisting of 1000 samples), and fault set $\mathbf{X}_3^f$ (consisting of 1000 samples with a step change in f introduced at sample 50). The matrices $\mathbf{X}_3^t$, $\mathbf{X}_3^v$, $\mathbf{X}_3^f$ are standardized with the mean and standard deviation of $\mathbf{X}_3^t$. Two 4-10-1-10-4 RAEs with $\delta = 3$ are trained with $\mathbf{X}_3^t$: a linear RAE employing the linear activation function and a nonlinear RAE employing the tangent hyperbolic activation function.

Figure E.6(a) shows a scatter plot of the samples $u_1[k-1]$ and $y_2[k]$ from the validation set $\mathbf{X}_3^v$. It is evident from the figure that the cross-correlation between $u_1[k-\tau]$ and $y_2[k]$ is nonlinear for the lag parameter $\tau = 1$. Figures E.6(b) and Figures E.6(c) shows a scatter plot of the reconstructions $\hat{u}_1[k-1]$ and $\hat{y}_2[k]$ from propagating $\mathbf{X}_3^v$ through the linear RAE and nonlinear RAE, respectively. The figures show that the linear RAE does not capture the nonlinear cross-correlation between the variables while the nonlinear RAE does capture the nonlinear cross-correlation.

Figures E.7(a) and E.7(b) show a time series plot and histogram plot, respectively, of the SPE obtained from propagating $\mathbf{X}_3^v$ and $\mathbf{X}_3^f$ through the linear RAE model. Figures E.7(c) and E.7(d) show equivalent plots of the SPE obtained from propagating $\mathbf{X}_3^v$ and $\mathbf{X}_3^f$ through the nonlinear RAE model. The figures indicate that the linear RAE's SPE has greater variance compared to the nonlinear RAE's SPE when nominal samples are provided. This suggests that the linear RAE performs worse at modeling

the nonlinear correlations exhibited by the training data, which consequently leads to a larger 95% control limit. The linear RAE also performs poorly at processing abnormal samples, evident from the fact that many abnormal samples produce an SPE below the 95% control limit.

E.4 Conclusion

This paper introduces a RAE-based method for MSPM of dynamic processes. In the proposed method, an RAE is trained to perform dynamic feature extraction of process variables sampled from a dynamic process exhibiting nominal operating conditions. The extracted features are then decoded to produce a sample-delayed reconstruction of the original process variables. New observations are referenced

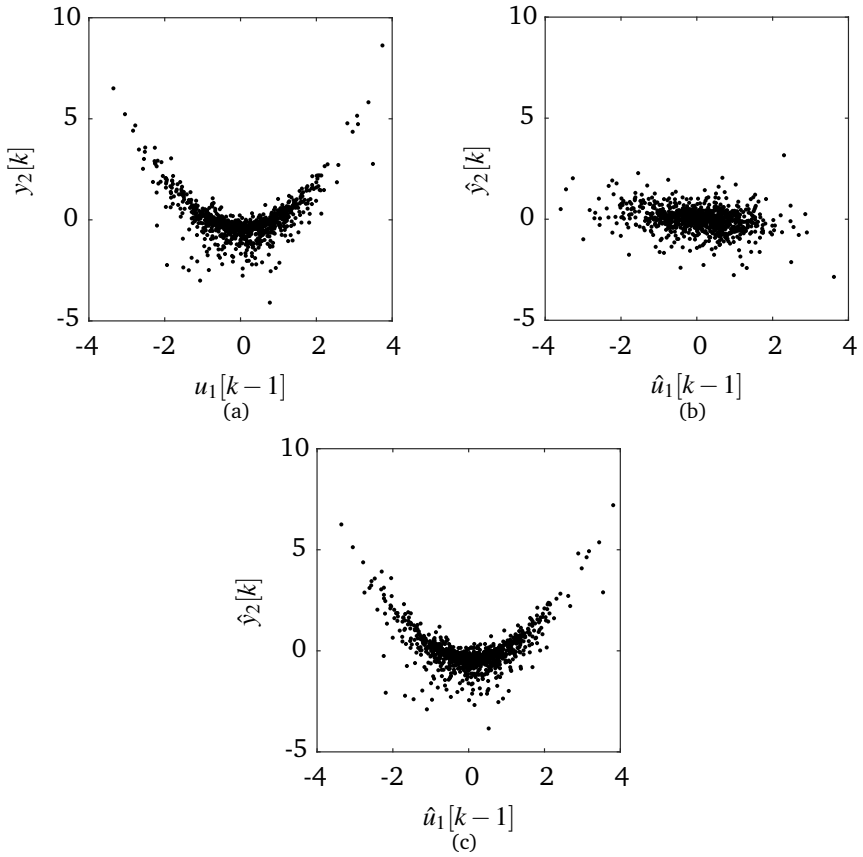


Figure E.6: For the validation set $\mathbf{X}_3^v$: (a) scatter plot of original variables, (b) scatter plot of reconstructions from linear RAE, and (c) scatter plot of reconstructions for nonlinear RAE.

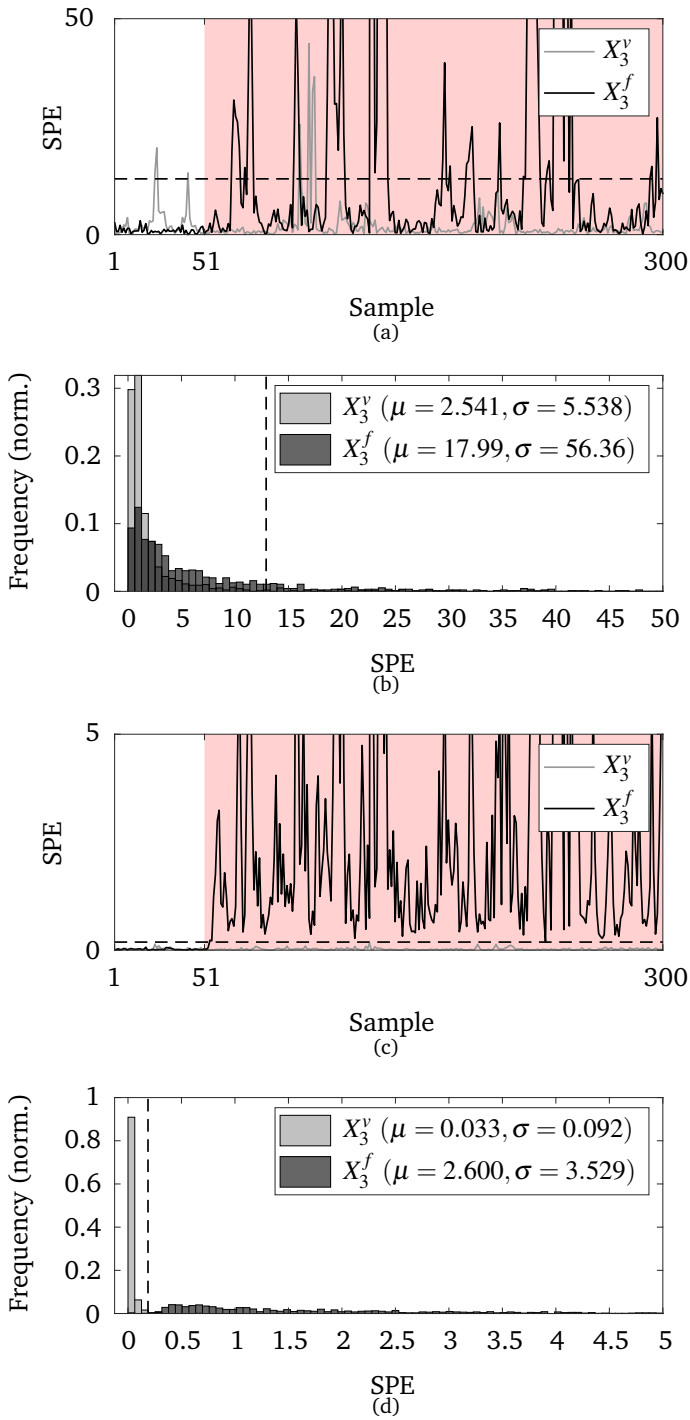


Figure E.7: SPE for (a),(b) the linear RAE model and (c),(d) the nonlinear RAE model. All plots include a 95% control limit (dashed line) determined from their respective validation set.

against the “in-control” model to detect abnormal events in the process. Recurrent hidden network layers facilitate the discovery of cross-correlations between process variables, permitting an RAE to distinguish well between input-induced and fault-induced process variations.

The fault detection performance of a linear RAE was compared against that of a linear model given by dynamic PCA. Both models were used to monitored a linear process. DPCA extends PCA to account for cross-correlation by augmenting the variable vector with time-lagged versions of itself. The results show that the RAE-based model was more sensitive to a process fault compared to the DPCA-based model. Furthermore, the DPCA-based model incorrectly inferred some abnormal samples as nominal. This did not occur for the RAE-based model. Collectively, the results indicate that an RAE-based approach to fault detection provides a definite indication that an abnormal event has occurred in a dynamic process.

The performance of a nonlinear RAE was compared against that of a linear RAE for the monitoring of a nonlinear process. The results show that the nonlinear RAE performed better than the linear RAE at modeling the nonlinear correlations exhibited by the process variables. Furthermore, the nonlinear RAE performed well at distinguishing between nominal and abnormal samples.

Acknowledgments

The authors would like to acknowledge the support of the Danish Hydrocarbon Research and Technology Center (DHRTC) at the Technical University of Denmark.

Bibliography

[1] Abdi, H., and Williams, L.J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics, 2(4): 433-459*.

[2] Ahmed, M., Baqqar, M., Gu, F., and Ball, A. D. (2012). Fault detection and diagnosis using Principal Component Analysis of vibration data from a reciprocating compressor. *Proceedings of 2012 UKACC international conference on control (pp. 461-466). IEEE*.

[3] Alcala, C.F. and Joe Qin, S.(2009). Reconstruction-based contribution for process monitoring. *Automatica, 45(7), 1593-1600*.

[4] Alcala, C.F., and Qin, S.J. (2011). Analysis and generalization of fault diagnosis methods for process monitoring. *Journal of Process Control, 21(3), 322-330*.

[5] Ashfahani, A., Pratama, M., Lughofer, E., and Ong, Y.S. (2020). DEVDAN: Deep evolving denoising autoencoder. *Neurocomputing, 390: 297-314*.

[6] Back, A.D., and Weigend, A.S. (1997). A first application of independent component analysis to extracting structure from stock returns. *International Journal of Neural Systems 8(04): 473-484*.

[7] P. Baldi and K. Hornik. Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks, 2(1):53-58,1989*.

[8] Bailey Peterson (September 10, 2020). Car Ownership Statistics (2020 Report). Last accessed 06 October 2020. Value Penguin. *url:https://www.valuepenguin.com/auto-insurance/car-ownership-statistics*.

[9] Bailey, S.J. (1984). From desktop to plant floor, a CRT is the control operators window on the process. *Control Engineering 31(6): 86-90*.

[10] Bakdi, A., and Kouadri, A. (2017). A new adaptive PCA based thresholding scheme for fault detection in complex systems. *Chemometrics and Intelligent Laboratory Systems, 162: 83-93.*.

[11] Bhat ,N.V. and McAvoy, T.J. (1992). Determining model structure for neural models by network stripping. *Computers & Chemical Engineering 16(4):271-281*.

[12] G.E. Box (1954). Some theorems on quadratic forms applied in the study of analysis of variance problems: I, Effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics, 25(2):290-302*.

[13] H. Bourlard and Y. Kamp (1988). Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics, 59(4-5): 291-294*.

[14] (July 19, 2020). What is industry value added?. Last accessed 20 August 2020. Bureau of Economic Analysis. *url:https://www.bea.gov/help/faq/184*.

[15] Brooks, E.B., et al. newblock On-the-Fly Massively Multitemporal Change Detection Using Statistical Quality Control Charts and Landsat Data. *IEEE Transactions on Geoscience and Remote Sensing, 52(6):3316-3332*.

[16] Bullemer, P., Vernon Reising, D., Burns, C., Hajdukiewicz, J., and Andrzejewski, J. (2008). ASM consortium guidelines-effective operator display design. *Houston, Honeywell International Inc./ASM Consortium*.

[17] Cardoso, J.F., and Souloumiac, A. (1993). Blind beamforming for non-Gaussian signals. *In IEE proceedings F (radar and signal processing) (Vol. 140, No. 6: 362-370). IET Digital Library*.

[18] D. Charte, F. Charte, S. García, M.J. del Jesus, and F. Herrera (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion, 44: 78-96*.

[19] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR), 41(3): 1-58*.

[20] Chen, J., and Patton, R.J. (2012). Robust model-based fault diagnosis for dynamic systems (Vol. 3). *Springer Science & Business Media*.

[21] Chen, Q., Wynne, R.J., Goulding, P., and Sandoz, D. (2000). The application of principal component analysis and kernel density estimation to enhance process monitoring. *Control Engineering Practice, 8(5): 531-543*.

[22] Cheng, F., He, Q.P., and Zhao, J. (2019). A novel process monitoring approach based on variational recurrent autoencoder. *Computers & Chemical Engineering, 129:106515*.

[23] Cheung, Y.M., and Xu, L. (1999). An empirical method to select dominant independent components in ICA for time series analysis. *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339) (Vol. 6: 3883-3887). IEEE*.

[24] Cheung, Y.M., and Xu, L. (2000). Independent component ordering in ICA time series analysis. *Neurocomputing 41(1-4): 145-152*.

[25] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv: 1406.1078*.

[26] Choi, S.W., and Lee, I.B. (2004). Nonlinear dynamic process monitoring based on dynamic kernel PCA. *Chemical engineering science, 59(24): 5897-5908.*.

[27] Connor, J.T., Martin, R.D., and Atlas, L.E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks, 5(2): 240-254*.

[28] Comon, P. (1994). Independent component analysis, a new concept?. *Signal processing 36(3): 287-314*.

[29] CSB15 (2007). Investigation report, Refinery explosion and fire. *BP-Texas City, Texas, March 23, 2005*.

[30] G. Cybenko (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems, 2(4):303-314*.

[31] Del Frate, L. (2014). Failure: analysis of an engineering concept. *Ph.D. Thesis*.

[32] Dong, D. and McAvoy, T.J. (1996). Nonlinear principal component analysis-based on principal curves and neural networks. *Computers & Chemical Engineering 20(1):65-78*.

[33] Duda, R.O., Hart, P.E., and Stork, D.G. (1973). Pattern classification and scene analysis. *New York: Wiley*.

[34] Fabius, O., and van Amersfoort, J.R.(2014). Variational recurrent auto-encoders. *arXiv preprint arXiv: 1412.6581*.

[35] Fauvel, M., Chanussot, J., and Benediktsson, J. A. (2009). Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. *EURASIP Journal on Advances in Signal Processing 2009: 1-14*.

[36] Fezai, R., Mansouri, M., Taouali, O., Harkat, M.F., and Bouguila, N. (2018). Online reduced kernel principal component analysis for process monitoring. *Journal of Process Control, 61: 1-11*.

[37] Gajjar, S., Kulahci, M., and Palazoglu, A. (2018). Real-time fault detection and diagnosis using sparse principal component analysis. *Journal of Process Control, 67: 112-128*.

[38] Gajjar, S., Kulahci, M., and Palazoglu, A. (2018). Use of sparse principal component analysis (SPCA) for fault detection. *IFAC-PapersOnLine, 49(7): 693-698*.

[39] Gao, H., Gajjar, S., Kulahci, M., Zhu, Q., and Palazoglu, A. (2016). Process knowledge discovery using sparse principal component analysis. *Industrial & Engineering Chemistry Research, 55(46): 12046-12059*.

[40] Ge, Z., Yang, C., and Song, Z. (2009). Improved kernel PCA-based monitoring approach for nonlinear processes. *Chemical Engineering Science, 64(9): 2245-2255*.

[41] Goel, P., Datta, A., and Mannan, M.S (2007). Industrial alarm systems: Challenges and opportunities. *Journal of Loss Prevention in the Process Industries, 50:23-36*.

[42] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep learning. *MIT press*.

[43] Graves, A., Mohamed, A.R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE*.

[44] Groover, M. P. (2010). Fundamentals of modern manufacturing: materials, processes, and systems. 4th edition. *John Wiley & Sons*.

[45] Autoencoder based residual generation for fault detection of quadruple tank system. *IEEE Conference on Control Technology and Applications (CCTA), p:994-999*.

[46] Hallgrímsson, Á.D., Niemann, H.H., and Lind, M. (2020) Improved process diagnosis using fault contribution plots from sparse autoencoders. *21st IFAC World Congress*.

[47] Hastie, T., and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association, 84(406): 502-516*.

[48] Health and Safety Executive (1997). The explosion and fires at the Texaco Refinery, Milford Haven, 24 July 1994. *A report of the investigation by the Health and Safety Executive into the explosion and fires on the Pembroke Cracking Company Plant at the Texaco Refinery, Milford Haven on 24 July 1994.*.

[49] He, Q.P. and Wang, J. (2018). Statistical process monitoring as a big data analytics tool for smart manufacturing. *Journal of Process Control, 67:35-43.*

[50] Hinton G.E. (2006). Reducing the dimensionality of data with neural networks. *Science, 313(5786):504-507*.

[51] Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology, 24(6):417-441*.

[52] Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys 2: 94-128*.

[53] Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks, 13(4-5), 411-430*.

[54] Hyvärinen, A. and Pajunen, P. (1999). Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks 12(3): 429-439*.

[55] ISA. A (2009). ISA-18.2: Management of Alarm Systems for the Process Industries. *International Society of Automation. Durham, NC, USA*.

[56] ISO 9000:2005, Clause 3.2.11.

[57] ISO 9000:2005, Clause 3.2.10.

[58] Jackson, J.E.(1991). A User's Guide to Principal Components. *Wiley-Interscience: New York, 1991*.

[59] Japkowicz, N., Hanson, S.J., and Gluck, M.A. (2000). Nonlinear autoassociation is not equivalent to PCA. *Neural Computation, 12(3): 531-545*.

[60] Ji, H., He, X., and Zhou, D. (2016). On the use of reconstruction-based contribution for fault diagnosis. *Journal of Process Control, 40, 24-34*.

[61] G. Jiang, P. Xie, H. He, and J. Yan (2017). Wind turbine fault detection using a denoising autoencoder with temporal information. *IEEE/Asme Transactions on Mechatronics, 23(1): 89-100*.

[62] Joe Qin, S. (2014). Process data analytics in the era of big data. *AIChE Journal, 60(9), 3092-3100*.

[63] Joe Qin, S. (2003). Statistical process monitoring: basics and beyond. *Journal of Chemometrics: A Journal of the Chemometrics Society, 17(8-9): 480-502*.

[64] Johansson, K.H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *JIEEE Transactions on control systems technology, 8(3), 456-465*.

[65] Jorgensen, K.W., and Hansen, L.K. (2011). Model selection for Gaussian kernel PCA denoising. *IEEE transactions on neural networks and learning systems, 23(1): 163-168*.

[66] Hsu, C.C., Chen, M.C., and Chen, L.S. (2010). A novel process monitoring approach with dynamic independent component analysis. *Control Engineering Practice, 18(3): 242-253.*.

[67] Karhunen, J., Oja, E., Wang, L., Vigario, R., and Joutsensalo, J. (1997). A class of neural networks for independent component analysis. *IEEE Transactions on neural networks, 8(3): 486-504*.

[68] Karhunen, J. (1996). Neural approaches to independent component analysis and source separation. *ESANN, vol. 96, pp. 249-266*.

[69] Karunanithi, N., Whitley, D., and Malaiya, Y.K. (1992). Using neural networks in reliability prediction. *IEEE Software 9.4 (1992): 53-59*.

[70] Kenton, W. (Jul 19, 2020). Value-Added Definition. Last accessed 20 August 2020. Investopedia. *url:https://www.investopedia.com/terms/v/valueadded.asp*.

[71] Kingma ,D.P. and Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint, arXiv:1412.6980*.

[72] Kramer, M.A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal, 37(2): 233-243*.

[73] Kresta, J.V., MacGregor, J.F., and Marlin, T.E. (1991). Multivariate statistical monitoring of process operating performance. *The Canadian journal of chemical engineering, 69(1):35-47*.

[74] Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems: 1097-1105*.

[75] Ku, W., Storer, R.H., and Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and intelligent laboratory systems, 30(1): 179-196*.

[76] Le, L., Hao, J., Xie, Y., and Priestley, J. (2016). Deep kernel: Learning kernel function from data using deep neural network. *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies: 1-7*.

[77] Le, L., and Xie, Y. (2019). Deep embedding kernel. *Neurocomputing, 339: 292-302.*.

[78] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature, 521(7553): 436-444*.

[79] Lee, E.A. (2008). Cyber physical systems: Design challenges. *In 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC):363-369*.

[80] Lee, J., Bagheri, B., and Kao, H. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters, 3:18-25*.

[81] Lee, J.M., Qin, S.J., and Lee, I.B. (2007). Fault detection of non-linear processes using kernel independent component analysis. *The Canadian Journal of Chemical Engineering, 85(4): 526-536*.

[82] Lee, J.M., Yoo, C., and Choi, S.W., Vanrolleghem, P.A., and Lee, I.B. (2004). Nonlinear process monitoring using kernel principal component analysis. *Chemical Engineering Science, 59(1): 223-234*.

[83] Lee, J.M., Yoo, C., and Lee, I.B. (2004) Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chemical engineering science, 59(14): 2995-3006.*.

[84] Lee, J.M., Yoo, C., and Lee, I.B. (2004) Statistical process monitoring with independent component analysis. *Journal of Process Control, 14(5): 467-485*.

[85] Lee, S., Kwak, M., Tsui, K.L., and Kim, S.B. (2019). Process monitoring using variational autoencoder for high-dimensional nonlinear processes. *Engineering Applications of Artificial Intelligence, 83: 13-27*.

[86] J. Li, and X. Yan (2020). Process monitoring using principal component analysis and stacked autoencoder for linear and nonlinear coexisting industrial processes. *Journal of the Taiwan Institute of Chemical Engineers, 112: 322-329*.

[87] Li, Z., and Yan, X. (2018). Adaptive selective ensemble-independent component analysis models for process monitoring. *Industrial & Engineering Chemistry Research, 57(24): 8240-8252*.

[88] Li, Y., and Mandt, S. (2018). Disentangled sequential autoencoder. *arXiv preprint arXiv:1803.02991*.

[89] Liu, P., Qiu, X., and Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

[90] MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock PLS methods. *AIChE Journal, 40(5): 826-838*.

[91] MacGregor, J.F. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice, 3(3):403-414*.

[92] Malthouse, E.C. (1998). Limitations of nonlinear PCA as performed with generic neural networks. *IEEE Transactions on neural networks, 9(1): 165-173*.

[93] Martin, E.B., and Morris, A.J. (1996). Non-parametric confidence bounds for process performance monitoring charts. *Journal of Process Control, 6(6): 349-358*.

[94] McCulloch, W.S., and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics, 5(4): 115-133*.

[95] Miller, P., Swanson, R.E., and Heckler, C.E. (1998). Contribution plots: a missing link in multivariate quality control. *Applied Mathematics and Computer Science, 8(4):775-792*.

[96] Mika, S., Schölkopf, B., Smola, A.J., Müller, K.R., Scholz, M., and Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. *Advances in neural information processing systems: 536-542*.

[97] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[98] Murata, N., Yoshizawa, S., and Amari, S.I. (1994). Network information criterion-determining the number of hidden units for an artificial neural network model. *IEEE transactions on neural networks, 5(6): 865-872*.

[99] Nielsen, M.A. (2015). Neural networks and deep learning. *Determination press*.

[100] Nguyen, V.H., and Golinval, J.C. (2010). Fault detection based on kernel principal component analysis. *Engineering Structures, 32(11): 3683-3691*.

[101] Nomikos, P., and MacGregor, J.F. (1995). Multivariate SPC charts for monitoring batch processes. *Technometrics, 37(1): 41-59*.

[102] Olah, C. (2014). Neural networks, manifolds, and topology, last accessed 07 June 2020. *url:http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/*.

[103] Osmani, A., Hamidi, M., and Bouhouche, S. (2019). Monitoring of a dynamic system based on autoencoders. *Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press:1836-1843*.

[104] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series B 2, 559-572*.

[105] Peres, F.A.P., and Fogliatto, F. S. (2018). Variable Selection Methods in Multivariate Statistical Process Control: A Systematic Literature Review. *Computers & Industrial Engineering, 115:603-619*.

[106] Plaut, E. (2018). From principal subspaces to principal components with linear autoencoders. *arXiv preprint arXiv:1804.10253*.

[107] Pudil, P., Novovičová, J. (1998). Novel methods for feature subset selection with respect to problem knowledge. *Feature extraction, construction and selection. Springer, Boston, MA., 101-116*.

[108] Ramachandran, P., Zoph, B., and Le, Q.V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

[109] Rico-Sulayes, A. (2017). Reducing vector space dimensionality in automatic classification for authorship attribution. *Revista Científica de Ingeniería Electrónica, Automática y Comunicaciones 38(3): 26-35*.

[110] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review, 65(6): 386-408*.

[111] Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R. and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

[112] Sarle, W. (1994). Neural networks and statistical models. *Proceedings Of The 19th Annual SAS Users Group International Conference (pp. 1538-1550). SAS Institute*.

[113] Shi, X. (2011). Blind signal processing. *Springer Berlin Heidelberg, Chapter 5*.

[114] Shelns, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

[115] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61, 85-117*.

[116] M. Scholz (2012). Validation of nonlinear PCA. *Neural Processing Letters 36(1): 21-30*.

[117] M. Scholz, F. Kaplan, C.L. Guy, J. Kopka, and J. Selbig. Non-linear PCA: a missing data approach. *Bioinformatics, 21(20): 3887-3895*.

[118] Schölkopf, B., Mika, S., Burges, C.J., Knirsch, P., Müller, K.R., Ratsch, G., and Smola, A.J (1999). Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks, 10(5): 1000-1017*.

[119] Schölkopf, B., Smola, A., and Müller, K.R. (1998) Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation, 10(5): 1299-1319*.

[120] Schölkopf, B. (2001). The kernel trick for distances. *Advances in Neural Information Processing Systems: 301-307*.

[121] Sun, W., Shao, S., Zhao, R., Yan, R., Zhang, X., and Chen, X. (2016). A sparse auto-encoder-based deep neural network approach for induction motor faults classification. *Measurement 89: 171-178*.

[122] Sutskever, I., Vinyals, O., and Le, Q.V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems: 3104-3112*.

[123] Tracy, N. D., Young, J. C., and Mason, R. L. (1992). Multivariate control charts for individual observations. *Journal of quality technology, 24(2): 88-95*.

[124] Varon, C., Alzate, C., and Suykens, J.A. (2015). Noise level estimation for model selection in kernel PCA denoising. *IEEE transactions on neural networks and learning systems, 26(11): 2650-2663.*.

[125] Valle, S., Li, W., and Qin, S.J. (1999). Selection of the number of principal components: the variance of the reconstruction error criterion with a comparison to other methods. *Industrial & Engineering Chemistry Research, 38(11): 4389-4401*.

[126] Vigário, R.N. (1997). Extraction of ocular artefacts from EEG using independent component analysis. *Electroencephalography and clinical neurophysiology, 103(3): 395-404*.

[127] Van den Kerkhof, P., Vanlaer, J., Gins, G., and Van Impe, J.F. (2013). Analysis of smearing-out in contribution plot based fault isolation for statistical process control. *Chemical Engineering Science, 104, 285-293*.

[128] Van den Kerkhof, P., Vanlaer, J., Gins, G., and Van Impe, J.F. (2013). Contribution plots for statistical process control: Analysis of the smearing-out effect. *In 2013 European Control Conference (ECC). IEEE:428-433*.

[129] Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative review. *Journal of Machine Learning Research, 10(66-71): 13*.

[130] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A., and Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research, 11(12):3371-3408*.

[131] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K. (2003). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering, 27(3):293-311*.

[132] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K. (2003). A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies. *Computers & Chemical Engineering, 27(3):313-326*.

[133] Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N. and Yin, K. (2003). A review of process fault detection and diagnosis: Part III: Process history based methods. *Computers & Chemical Engineering, 27(3):327-346*.

[134] Wang, J. and He, Q.P. (2010). Multivariate statistical process monitoring based on statistics pattern analysis. *Industrial & Engineering Chemistry Research, 49(17):7858-7869*.

[135] Wang, J., Yang, F., Chen, T., and Shah, S.L. (2015) An overview of industrial alarm systems: Main causes for alarm overloading, research status, and open problems. *IEEE Transactions on Automation Science and Engineering, 13(2): 1045-1061*.

[136] Wang, L., and Shi, H. (2010). Multivariate statistical process monitoring using an improved independent component analysis. *Chemical engineering research and design, 88(4): 403-414*.

[137] Wasserman, P.D. (1993). Advanced methods in neural computing. *John Wiley & Sons, Inc.*.

[138] Westerhuis, J.A., Gurden, S.P., and Smilde, A.K. (2000). Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and intelligent laboratory systems, 51(1): 95-114*.

[139] A.S. Willsky (1976). A survey of design methods for failure detection in dynamic systems. *Automatica, 12(6):601-611*.

[140] World Bank Open Data. Manufacturing, value added (% of GDP). *url:https://data.worldbank.org/*.

[141] X. Wu, G. Jiang, X. Wang, P. Xie, and X. Li (2019). A multi-level-denoising autoencoder approach for wind turbine fault detection. *IEEE Access, 7: 59376-59387*.

[142] Yan, J., Meng, Y., Lu, L., and Li., L. (2017). Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance. *IEEE Access 5, 23484-23491*.

[143] Yan, S., Smith, J.S., Lu, W., and Zhang, B. (2018). Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Transactions on Cognitive and Developmental Systems, 12(1):30-42*.

[144] Yan, W., Guo, P., and Li, Z. (2016). Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems 158:31-40*.

[145] Yang, Y., Sautière, G., Ryu, J.J., and Cohen, T.S. (2020). Feedback Recurrent AutoEncoder. *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP): 3347-3351*.

[146] Yoo, C.K., Lee, J.M., Vanrolleghem, P.A., and Lee, I.B. (2004). On-line monitoring of batch processes using multiway independent component analysis. *Chemometrics and intelligent laboratory systems, 71(2): 151-163*.

[147] Yoon, J., Yang, E., Lee, J., and Hwang, S.J. (2017). Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*.

[148] Yoon, S. and MacGregor, J.F. (2001). Fault diagnosis with multivariate statistical models part I: using steady state fault signatures. *Journal of Process Control, 11(4), 387-400*.

[149] Zhang, N., Tian, X.M., and Cai, L.F. (2013). Nonlinear dynamic fault dignosis method based on dautoencoder. *2013 Fifth International Conference on Measuring Technology and Mechatronics Automation (pp. 729-732)*. *IEEE*.

[150] Zhang, Y., and Qin, S.J. (2007). Fault detection of nonlinear processes using multiway kernel independent component analysis. *Industrial & engineering chemistry research, 46(23): 7780-7787*.

[151] H. Zhong, T. Xue, and S.X. Ding (2018). A survey on model-based fault diagnosis for linear discrete time-varying systems. *Neurocomputing, 306:51-60*.

[152] Zhou, G., Sohn, K., and Lee, H. (2012). Online incremental feature learning with denoising autoencoders. *Artificial intelligence and statistics:1453-1461*.

[153] Zhu ,M. and Gupta, S. (2017). To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*.

[154] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301-320*.

Technical University of Denmark
Automation and Control (AUT)
Elektrovej Building 326
DK-2800, Kgs. Lyngby
Denmark
Phone: (+45) 45 25 38 00
Email: elektro@elektro.dtu.dk
www.elektro.dtu.dk