LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Energy Systems
Master's Programme in Electrical Engineering

Pekka Räsänen

EVENT-DRIVEN APPROACH FOR TENNESSEE EASTMAN PROCESS

Examiners: Assistant Professor, D.Sc. (Tech.) Pedro Nardelli
Junior Researcher, M.Sc. Daniel Gutiérrez

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Energy Systems
Degree programme in Electrical Engineering

Pekka Räsänen

Event-driven approach for Tennessee Eastman process

Master's Thesis

2020

60 pages, 26 figures, 23 tables, 4 appendices

Examiners: Assistant Professor, D.Sc. (Tech.) Pedro Nardelli
Junior Researcher, M.Sc. Daniel Gutiérrez

Keywords: Tennessee Eastman process, event-driven filtering, principal component analysis, threshold method, delta method

As the Industrial Internet of Things is becoming more and more common, the amount of sensor data transferred inside systems is getting bigger and bigger. The amount of data comes easily so large that it effects to the performance of the operating system, so there is an obvious need to reduce this data. Data reduction for the system shouldn't be done at the expense of the reliability and the effectiveness of failure detection of the system and in this study, Threshold and Delta methods of event-driven filtering in Tennessee Eastman process were studied to explore the possibility to prevent the quality drop in failure detection. Event-driven filtering methods were used to resample data sets from Tennessee Eastman process, data sets which were originally made for testing of failure detection. To verify the results of compressing the data without causing problems to the fault detection, a research which was using principal component analysis was included to the process. Those original fault data sets from Tennessee Eastman process were used in that research and the regenerated fault data sets of this study were generated over and over again with multiple different compression ratios to achieve reliable results on the effectiveness of event-driven filtering in Tennessee Eastman process. These tested methods were confirmed to work properly to a certain, fairly high degree of compression ratios.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Energy Systems
Sähkötekniikka

Pekka Räsänen

Tapahtumapohjainen lähestymistapa Tennessee Eastman prosessiin

Diplomityö

2020

60 sivua, 26 kuvaa, 23 taulukkoa, 4 liitettä

Työn tarkastajat: Apulaisprofessori, TkT. Pedro Nardelli
Nuorempi tutkija, DI. Daniel Gutiérrez

Avainsanat: Tennessee Eastman prosessi, tapahtumapohjainen suodatus, pääkomponenttianalyysi, raja-arvo-menetelmä, delta-menetelmä

Teollisen esineiden internetin tullessa yhä yleisemmäksi, järjestelmien sisäisesti lähettämän anturidatan määrä kasvaa yhä suuremmaksi ja suuremmaksi. Datan määrä kasvaa helposti niin suureksi että se vaikuttaa operoivan järjestelmän suorituskykyyn jolloin tarve vähentää tätä määrää on ilmeinen. Järjestelmän lähettämän datan vähentämistä ei ole kuitenkaan syytä tehdä virheentunnistusmenetelmän luotettavuuden ja suorituskyvyn kustannuksella ja tässä tutkimuksessa keskityttiinkin raja-arvo- ja delta-menetelmien tapahtumapohjaiseen suodatukseen Tennessee Eastman prosessissa, tavoitteena estää virheentunnistuskyvyn laadun alentuminen. Tapahtumapohjaisen suodatuksen menetelmiä käytettiin muokkaamaan alun perin Tennessee Eastman prosessissa virheentunnistusmenetelmien testaukseen luotuja datajoukkoja. Jotta voitiin todentaa ettei datajoukon muokkaaminen pakkaamalla aiheuta ongelmia virheentunnistusmenetelmälle, työhön sisällytettiin tutkimus jossa käytettiin pääkomponenttianalyysia. Sisällytetyssä tutkimuksessa käytettiin Tennessee Eastman prosessin alkuperäisiä datajoukkoja ja tässä työssä muokatut datajoukot luotiin uudelleen ja uudelleen eritasoisilla pakkaussuhteilla jotta saavutettaisiin luotettavat tulokset Tennessee Eastman prosessin tapahtumapohjaisen suodatuksen testauksessa. Tutkitut menetelmät todettiin toimiviksi eräillä, melko korkeankin asteen pakkaussuhteilla.

ACKNOWLEDGEMENTS

This thesis was accomplished for the School of Energy Systems, Lappeenranta-Lahti University of Technology LUT in Finland, during spring and summer of 2020.

I would like to express my honest gratitude to Assistant Professor Pedro Nardelli and Junior Researcher Daniel Gutierrez, supervisors of this master's thesis and also to professor Gustavo De Almeida for their invaluable help throughout the research. Without your encouragement and support, this thesis would have been impossible to complete. I’m also grateful that I got to know your FIREMAN project, excited to see the final results of the project.

Last but not least, I would like to express my special gratitude to my dear family and my dearest friends for their support and understanding.

Pekka Räsänen

Veikkola, July 2020

TABLE OF CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

Symbols

A, C, D, E	Tennessee Eastman reactants
B	Tennessee Eastman inert
Deg C	degree, Celsius
F	Tennessee Eastman by-product
G, H	Tennessee Eastman products
hr	hour
kg	kilogram
kPa	kilo-Pascal
kscmh	kilo standard cubic meters per hour
kW	kilowatt
m^3	cubic-meter
Q	statistic for fault detection (also known as SPE, explained in abbreviations)
T^2	statistic for fault detection

Abbreviations

CVA	Canonical Variate Analysis
DSP	Digital Signal Processing
DPCA	Dynamic Principal Component Analysis
FA	Factor Analysis
FDA	Fisher Discriminant Analysis
IoT	Internet of Things
MDR	Missed Detection Rate
PCA	Principal Component Analysis
PLS	Partial Least Squares
SPE	Squared Prediction Error
TE	Tennessee Eastman
TEP	Tennessee Eastman process
XMV	Manipulated Variable in Tennessee Eastman process
XMEAS	Measurement in Tennessee Eastman process

1 INTRODUCTION

1.1 Background

Industrial IoT, which refers to Industrial Internet of Things, has become a standard for the systems and technology of modern society. It is about devices and systems communicating to each other, creating their own network for transferring data and information between each other. These networks between devices are growing day by day, with more and more devices included. Even normal everyday devices like refrigerators are joining this groups of devices, which are able to report about themselves to the network via their automation systems. (D. Serpanos et al., 2018)

Even though the idea of IoT technology seems to be quite clear, there is no easy clarification what this term really means and which parts can be categorized under IoT -network. D. Serpanos and M. Wolf are identifying some example possibilities for them, and those are next:

- "Internet-enabled physical devices, although many devices don't use the Internet Protocol" (D. Serpanos et al., 2018)
- "Soft real-time sensor networks" (D. Serpanos et al., 2018)
- "Dynamic and evolving networks of embedded computing devices" (D. Serpanos et al., 2018)

Behind the subject of this thesis, associated with the industrial IoT and the sensor network -part of it, is a well-known process which became familiar already in 1993. That process is called Tennessee Eastman process, TEP, and it was invented in Eastman Chemical factory in Tennessee. TEP is a simulated model of a process in a chemical factory with 21 made up faults and it was used for improving a fault detection in a system. At the same time, the transferred sensor data inside the IoT -network is reduced dramatically to ease up the workload of the system to improve the fault detection.

For the IoT, one of the most important viewpoints is aperiodic sampling, also known as event-driven. Normally it is presumable to have periodic sampling in time-series data for digital signal processing, DSP and especially for controlling it. The problem though is that this consumes too much energy at the nodes, and it uses too much the bandwidth of the network. For some situations, this is not acceptable and for any situation, this is load that should be eliminated from the system. Event-driven, which uses edge processing as its operating method, eliminates unnecessary data from the system which eases up the use of the bandwidth. And because wireless communication uses quite a lot of energy, it helps especially the solutions with battery-powered sensors.

D. Serpanos and M. Wolf are relying that for the IoT-systems, events are pivotal data types and that event-driven is an important structuring technique and for the use of the results of this thesis, it may have an important effect and it might model an useful solution. (D. Serpanos et al., 2018)

1.2 Goals and delimitations

This thesis is concentrating only on the effects of Threshold and Delta methods of event-driven approach in TEP for testing data sets which will be used with a project which is using Principal Component Analysis (PCA) -method. Other fault detection methods like Dynamic Principal Component Analysis (DPCA), Partial Least Squares (PLS), Fisher Discriminant Analysis (FDA) etc. are delimited out of this research. Also, the decision of the final actions, about of will these studied methods be taken in use with the co-operation project, will be made by other association. The idea was to study about the consequences of TEP -procedure with simulated data sets to provide the answers through comparable results, which will inform should this procedure be used in co-operation project or is there a better solution available.

1.3 Research tools

Tools for making the research were mostly limited to Jupyter Notebook and Python as the coding language. Original code, what for the resampled data was supposed to be made, was also written in Python. Though this didn’t really force to use the same coding language for resampling the data sets, but it made the whole procedure way simple and it made the testing of those resampled data sets possible in the same environment they were made. Testing was in a huge role to get a certainty of a success in resampling the data sets. Otherwise the comparison of the used technique wouldn’t have been reliable enough.

The data of the thesis was managed by using GitHub. Codes and figures of this thesis are available at https://github.com/pedrohjn/event-tep. The project where this thesis was involved is available in https://github.com/5uperpalo/FIREMAN-project.

1.4 Structure of the thesis

This thesis is structured in two main parts and the first part of it, is about the Tennessee Eastman process. This part starts on section 2 and it contains the literature review of Tennessee Eastman process and the story behind the process, all the way to early 90s when the process was invented. This also contains an overview of Russell's results which are used for a comparison later in the thesis. Section 3 starts the second part of the thesis and it is about the event-driven approaches. Event-driven filtering was a part of the Tennessee Eastman process which was used in this study and two different methods were under study. Threshold method and Delta method were compared together and also compared against the results without original TEP method. The results of the study are available in section 4, and the discussion of the results and the conclusions of the study are presented in section 5 including the summary of the thesis.

2 TENNESSEE EASTMAN PROCESS

2.1 Overview

Tennessee Eastman process (TEP) has become known in 1993 when it was created by the Eastman Chemical Company in Tennessee. The idea behind the process was to able process control evaluations and monitoring methods. Real factory data for these kind of needs was normally not available but now TEP provided simulation data which was good enough and it was made from available real industrial process so it became widely used as a source data in process monitoring community. (J.J. Downs et al., 1993; H.B. Aradhye et al., 1999)

2.2 Process

P.R. Lyman has introduced a flowsheet for the industrial plant, Figure 2.1, which explains the parts of the TEP (P.R. Lyman, 1992). Parts of the process are explained next and these consist mainly of a process flowsheet, process variables and process faults.

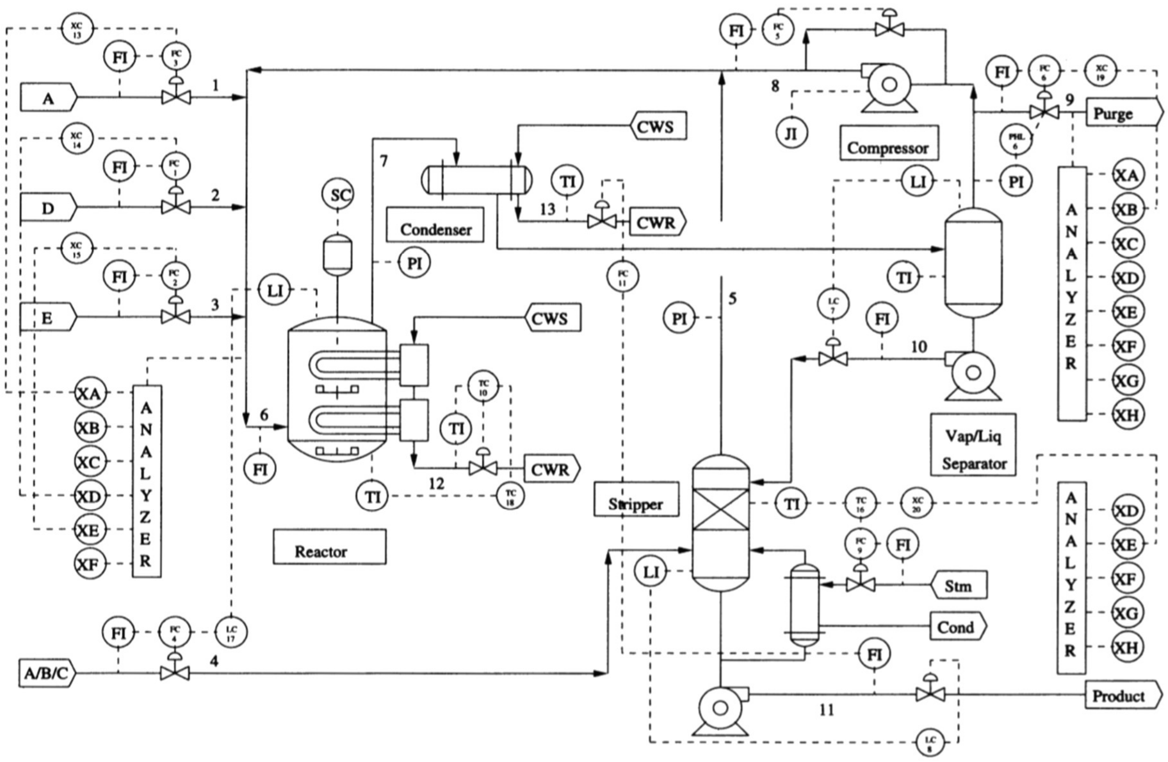


Figure 2.1 TEP -process flowsheet (P.R. Lyman, 1992)

The process flowsheet indicates that the process has five main units, which are from left to right: Reactor, Condenser, Stripper, Compressor and Vapour-Liquid Separator. Components in the process are marked as A, B, C, D, E, F, G and H and they have next relations in this chemical process:

$$A(g) + C(g) + D(g) \rightarrow G(liq)$$
$$A(g) + C(g) + E(g) \rightarrow H(liq)$$
$$A(g) + E(g) \rightarrow F(liq)$$
$$3D(g) \rightarrow 2F(liq)$$

Components A, C, D and E are gaseous reactants which are forming liquid products G and H in the process. Component B though is an inert which has no reaction and for that it has no relation in the process as can be seen on the formulas. Product F is a liquid by-product which also forms in the process. (L.H. Chiang et al., 2001)

In addition of components, TEP contains altogether 53 variables. Some of them are manipulated variables in Table 2.1, some are process measurements in Table 2.2 and the rest of them are composition measurement in Table 2.3. In these Tables, explanations for the variables are next:

XMV	Manipulated Variable
XMEAS	Measurement

Table 2.1 Manipulated variables (L.H. Chiang et al., 2001)

Variable	Description
XMV(1)	D Feed Flow (Stream 2)
XMV(2)	E Feed Flow (Stream 3)
XMV(3)	A Feed Flow (Stream 1)
XMV(4)	Total Feed Flow (Stream 4)
XMV(5)	Compressor Recycle Valve
XMV(6)	Purge Valve (Stream 9)
XMV(7)	Separator Pot Liquid Flow (Stream 10)
XMV(8)	Stripper Liquid Product Flow (Stream 11)
XMV(9)	Stripper Steam Valve
XMV(10)	Reactor Cooling Water Flow
XMV(11)	Condenser Cooling Water Flow
XMV(12)	Agitator Speed

Table 2.2 Process measurements (3-minute sampling interval) (L.H. Chiang et al., 2001)

Variable	Description	Units
XMEAS(1)	A Feed (Stream 1)	kscmh
XMEAS(2)	D Feed (Stream 2)	kg/hr
XMEAS(3)	E Feed (Stream 3)	kg/hr
XMEAS(4)	Total Feed (Stream 4)	kscmh
XMEAS(5)	Recycle Flow (Stream 8)	kscmh
XMEAS(6)	Reactor Feed Rate (Stream 6)	kscmh
XMEAS(7)	Reactor Pressure	kPa gauge
XMEAS(8)	Reactor Level	%
XMEAS(9)	Reactor Temperature	Deg C
XMEAS(10)	Purge Rate (Stream 9)	kscmh
XMEAS(11)	Product Sep Temp	Deg C
XMEAS(12)	Product Sep Level	%
XMEAS(13)	Prod Sep Pressure	kPa gauge
XMEAS(14)	Prod Sep Underflow (Stream 10)	m^3/hr
XMEAS(15)	Stripper Level	%
XMEAS(16)	Stripper Pressure	kPa gauge
XMEAS(17)	Stripper Underflow (Stream 11)	m^3/hr
XMEAS(18)	Stripper Temperature	Deg C
XMEAS(19)	Stripper Steam Flow	kg/hr
XMEAS(20)	Compressor Work	kW
XMEAS(21)	Reactor Cooling Water Outlet Temp	Deg C
XMEAS(22)	Separator Cooling Water Outlet Temp	Deg C

Table 2.3 Composition measurements (L.H. Chiang et al., 2001)

Variable	Description	Stream	Sampling Interval (min.)
XMEAS(23)	Component A	6	6
XMEAS(24)	Component B	6	6
XMEAS(25)	Component C	6	6
XMEAS(26)	Component D	6	6
XMEAS(27)	Component E	6	6
XMEAS(28)	Component F	6	6
XMEAS(29)	Component A	9	6
XMEAS(30)	Component B	9	6
XMEAS(31)	Component C	9	6
XMEAS(32)	Component D	9	6
XMEAS(33)	Component E	9	6
XMEAS(34)	Component F	9	6
XMEAS(35)	Component G	9	6
XMEAS(36)	Component H	9	6
XMEAS(37)	Component D	11	15
XMEAS(38)	Component E	11	15
XMEAS(39)	Component F	11	15
XMEAS(40)	Component G	11	15
XMEAS(41)	Component H	11	15

2.3 Simulation and control

The codes for simulating the process has been made available for everyone, normally this kind of real data is not available which makes the simulation data the best option for studies. Originally these codes were announced to be available in Fortran but at least currently they are available also in GitHub (R.D. Braatz, 2017). Professor Richard D. Braatz has shared there a directory which includes all 77 of those Fortran -codes for open-loop and closed-loop simulations, including training -files and testing data files. These training and testing data -files are meant to be used for evaluating the data-driven methods like Principal Component Analysis (PCA), Partial Least Squares (PLS), Fisher Discriminant Analysis (FDA) and Canonical Variate Analysis (CVA) and for this thesis, these same data files were used to test event-driven method with TEP for a code which was using PCA -methods. (J.J. Downs et al., 1993).

There is available 21 process faults and all of these were used in this study for testing the effects of compressing the data by event-driven TEP. Faults are explained on Table 2.4

The simulation in TEP has been made with open-loop process and the control structure of the process can be seen in Figure 2.1. Open-loop process is an unstable process and therefore normally industrial processes are operated in closed loop, which makes a need for a plant-wide control scheme while applying methods for process monitoring. (L.H. Chiang et al., 2001)

2.4 Data sets

Available simulated data for TEP includes those variables and measurements, which were separated in Tables 2.1, 2.2 and 2.3 but in this thesis, the concentration was on the faults which were mentioned in Table 2.4. The simulated data of TEP has been made by using a 3-minute intervals in sampling and the data of the faults includes 22 different simulation runs. (L.H. Chiang et al., 2001)

Fault simulation runs differ from each other by a random seed which was changed for every run. Fault 0 was a comparison run which was generated without faults and all the other 21 of them has a certain fault in them. Five of the faults were unknown and the explanations for all TEP Faults can be found on Table 2.4. (L.H. Chiang et al., 2001)

These faulty data sets were in the lead role for this thesis. These were used in the other project which was using Principal Component Analysis for a method to test does the system recognize the faults how well. These sets were resampled with Threshold and Delta methods in event-driven filtering and the compression ratio for the new data sets were 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 95%. Original fault data sets were then compared against these 11 new series of fault data sets to research the limits for the compression ratio. Objective was that the performance of the fault detection shouldn’t get worse with new lighter data.

Table 2.4 Process faults (L.H. Chiang et al., 2001)

Variable	Description	Type
IDV(1)	A/C Feed Ratio, B Composition Constant (Stream 4)	Step
IDV(2)	B Composition, A/C Ratio Constant (Stream 4)	Step
IDV(3)	D Feed Temperature (Stream 2)	Step
IDV(4)	Reactor Cooling Water Inlet Temperature	Step
IDV(5)	Condenser Cooling Water Inlet Temperature	Step
IDV(6)	A Feed Loss (Stream 1)	Step
IDV(7)	C Header Pressure Loss - Reduced Availability (Stream 4)	Step
IDV(8)	A, B, C Feed Composition (Stream 4)	Random Variation
IDV(9)	D Feed Temperature (Stream 2)	Random Variation
IDV(10)	C Feed Temperature (Stream 4)	Random Variation
IDV(11)	Reactor Cooling Water Inlet Temperature	Random Variation
IDV(12)	Condenser Cooling Water Inlet Temperature	Random Variation
IDV(13)	Reaction Kinetics	Slow Drift
IDV(14)	Reactor Cooling Water Valve	Sticking
IDV(15)	Condenser Cooling Water Valve	Sticking
IDV(16)	Unknown	
IDV(17)	Unknown	
IDV(18)	Unknown	
IDV(19)	Unknown	
IDV(20)	Unknown	
IDV(21)	The valve for Stream 4 was fixed at the steady state position	Constant Position

2.5 Fault detection

In Tennessee Eastman process, fault detection can be made for example by using PCA-based T^2 -statistic. This example is chosen for the PCA -part of the other project which relates to this thesis. PCA-based T^2 -statistic requires the estimated parameter M_a to match the number of independent degrees of freedom of the matrix product of $P\sum_a{}^{-2}P^T$. Estimated parameter M_a is calculated from Equation 1. (L.H. Chiang et al., 2001)

$$M_a = \frac{a+2am- \ ^2}{2}, \qquad (1)$$

For example, when parameter $a = 51$, the formula gives a value of 1377 for the number of independent parameter M_a .

Fault detection can be described also with a Shewhart chart. This kind of method was also used in a demonstration of TEP event-driven in Appendix 1, which can be found in the end of this thesis. There was an example of the effects of Threshold method in TEP for a single variable with a 90% compression rate where the upper and lower limits were marked for the diagrams to describe the moments of faults. Crossing the limits means a fault on that demonstration and same kind of example is also available on Figure 2.2 which is a more specific description of Shewhart chart. (N. Doganaksoy et al., 2001)

Upper and lower limits of control on the chart must be chosen very precise to minimize the amount of false alarms and to prevent missing detections. False alarm shows up as a indication of a fault even though there is no actual fault occurred and if a detection is missed, there is no information of an actual fault which can be a huge problem in a process of a chemical plant for example. For a decent fault detection, there is a compromise to balance between the amount of false alarms and the amount of missed detection rates. If the threshold limits have been estimated as too strict, the result will be high false alarm rate and low missed detection rate and if the limits have been estimated as too wide from each other, it causes low false alarm rate and a high missed detection rate. (L.H. Chiang et al., 2001)

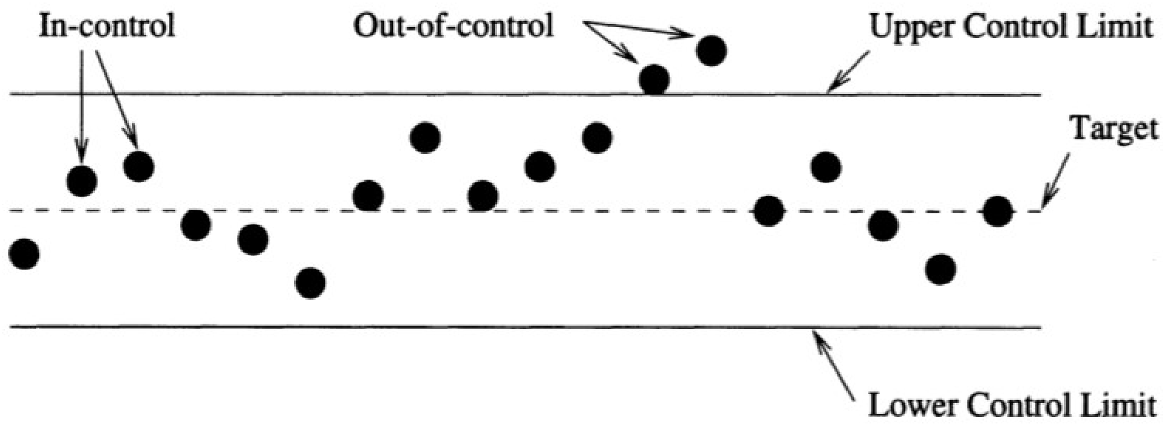


Figure 2.2 An example of the Shewhart chart. (L.H. Chiang et al., 2001)

2.6 E.L. Russell's results

Evan L. Russell has made a study of TEP with Richard D Braatz and Leo Chiang and the idea of this thesis was to make a comparison of the results which they have achieved compared to the results of this thesis. The difference was that in this study, event-driven - methods were taken alongside of TEP to add the data compression to the data sets. The event-driven methods have proven to improve the data handling of the system and the effects of that improvement with TEP were interesting to research.

In this section, the thesis was concentrating on the faults and their detection and these results are called Russell's results in this study. Russell's results contain PCA, DPCA and CVA - methods and variants which has been build up by these methods but the primary concentration in this study was on PCA -method. From the Russell's results, case study was chosen to be Fault 5 and the comparable figures for Russel's results can be found under chapter 4.

As an example, Fault 5 is about a step change of the temperature in the condenser cooling water inlet. The difference between the normal operation and the operation when the fault is on can be seen on Figure 2.3. Effect that the fault causes is that it induces a step change in the flow rate of cooling water in the condense. The fault also affects by increasing the flow rate of the outlet stream from the condenser to the vapor/liquid separator and this causes the temperature in the vapor/liquid separator to increase. Control loops compensate the change and if helps the separator to returns to its set-point temperature. It takes 10 hours from the system to reach the steady-state and the other 32 monitored variables out of 50, have the similar transients which takes the same 10 hours to settle. For the fault detection this kind of fault is not a hard task to recognize and the results of PCA, DPCA and CVA-based statistics can be seen in Figures 2.4 and 2.5

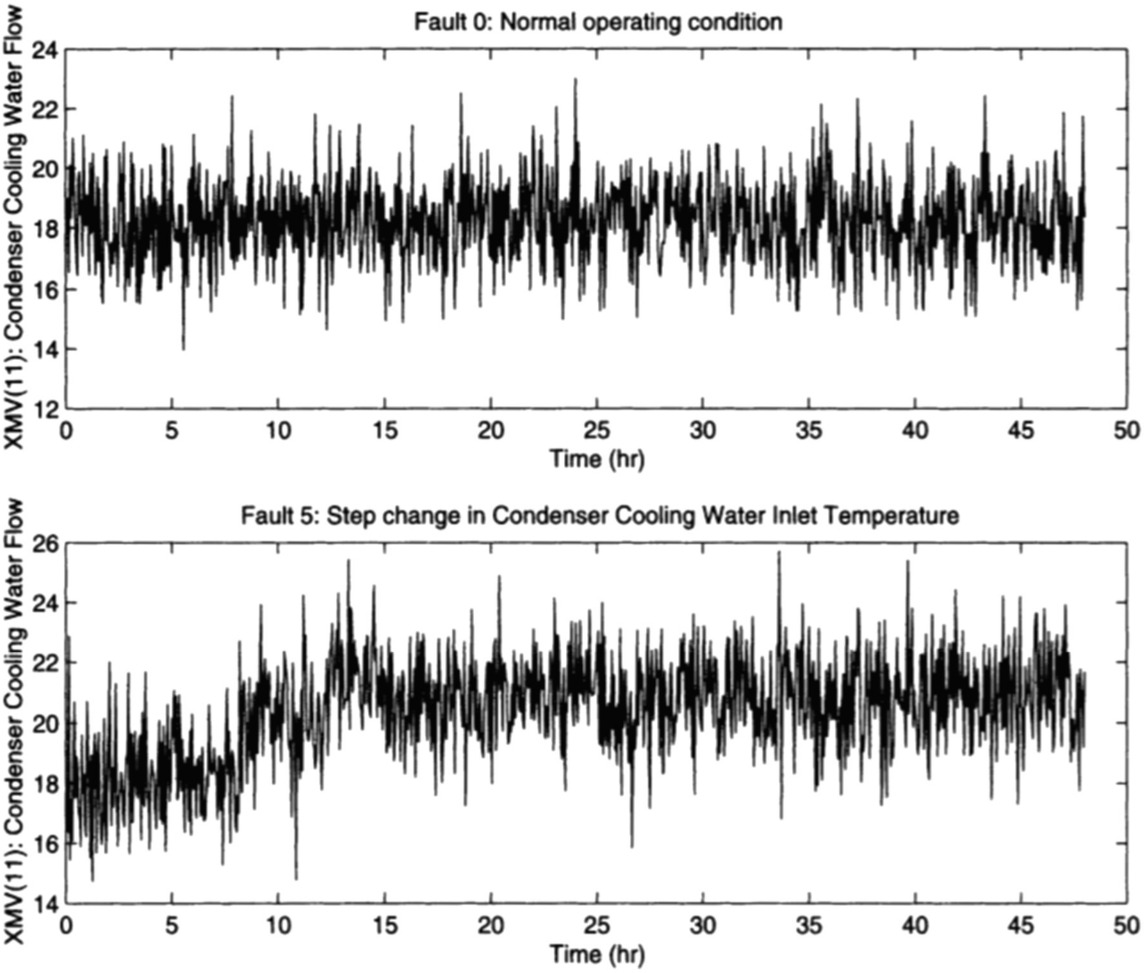


Figure 2.3 Comparison of XMV(11) for Faults 0 and 5 (E.L. Russell et al., 2001)

Table 2.5 Missed detection rates for Faults 1, 4, 5, and 1 (E.L. Russell et al., 2001)

Method	Fault Basis	1	4	5	11
PCA	T^2	0.008	0.956	0.775	0.794
PCA	Q	0.003	0.038	0.746	0.356
DPCA	T^2	0.006	0.939	0.756	0.801
DPCA	Q	0.005	$\boxed{0}$	0.748	$\boxed{0.193}$
CVA	T_s^2	0.001	0.688	$\boxed{0}$	0.515
CVA	T_r^2	$\boxed{0}$	$\boxed{0}$	$\boxed{0}$	0.195
CVA	Q	0.003	0.975	$\boxed{0}$	0.669

Table 2.5 shows some differences between different methods Russell's study was investigating and there can be seen that PCA- and DPCA -based statistics had a high missed detection rate. This differs totally from all of the CVA statistics which didn't have any missed detections.

The reason for this poor result compared to CVA can be clearly seen from the plotted figures of the observation variables over time. After 10 hours of the fault occurs, the PCA- and DPCA -based statistics failed to indicate a fault again which can be seen on Figure 2.4. And for the CVA statistics, they stayed above their thresholds which can be seen on Figure 2.5. (E.L. Russell et al., 2001)

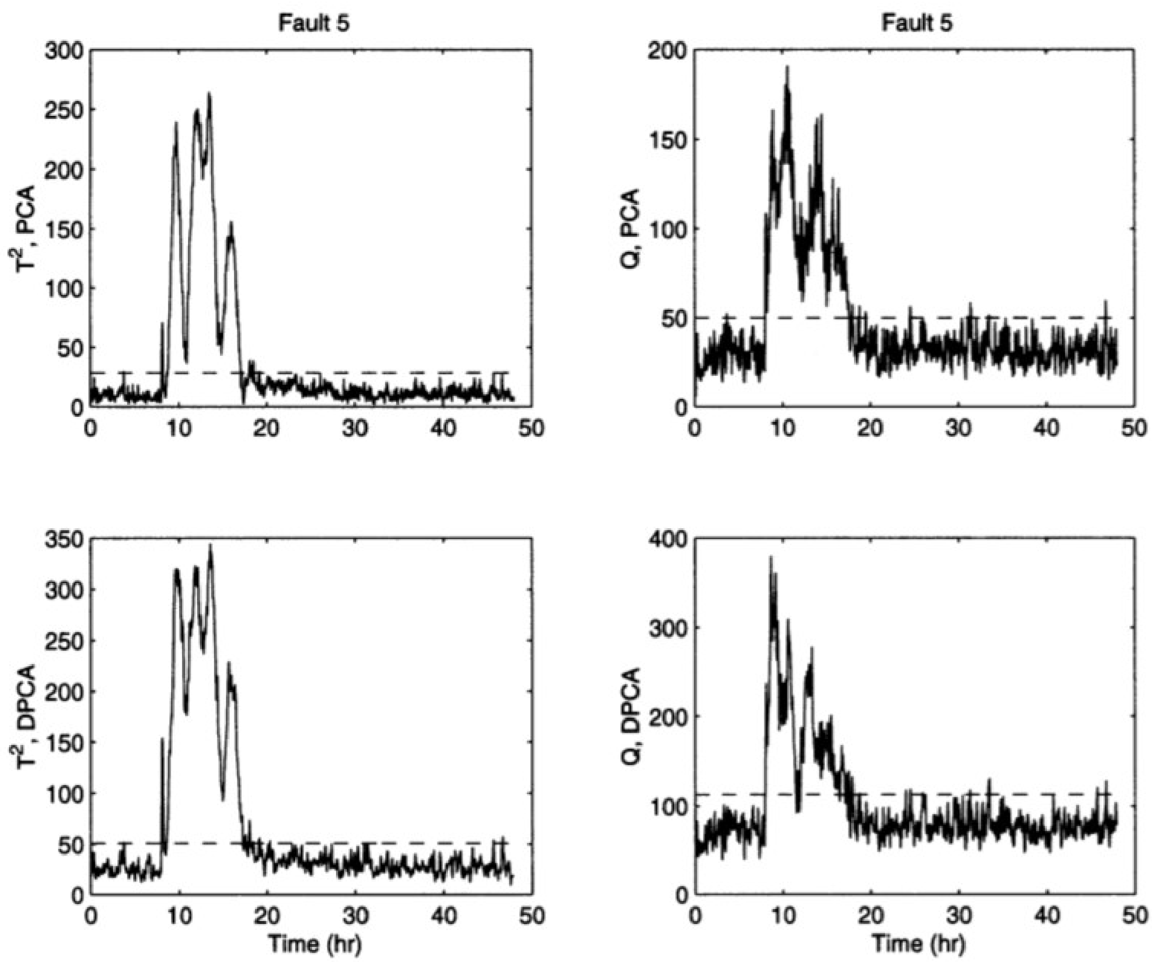


Figure 2.4 The PCA and DPCA multivariate statistics for fault detection for Fault 5 (E.L. Russell et al., 2001)

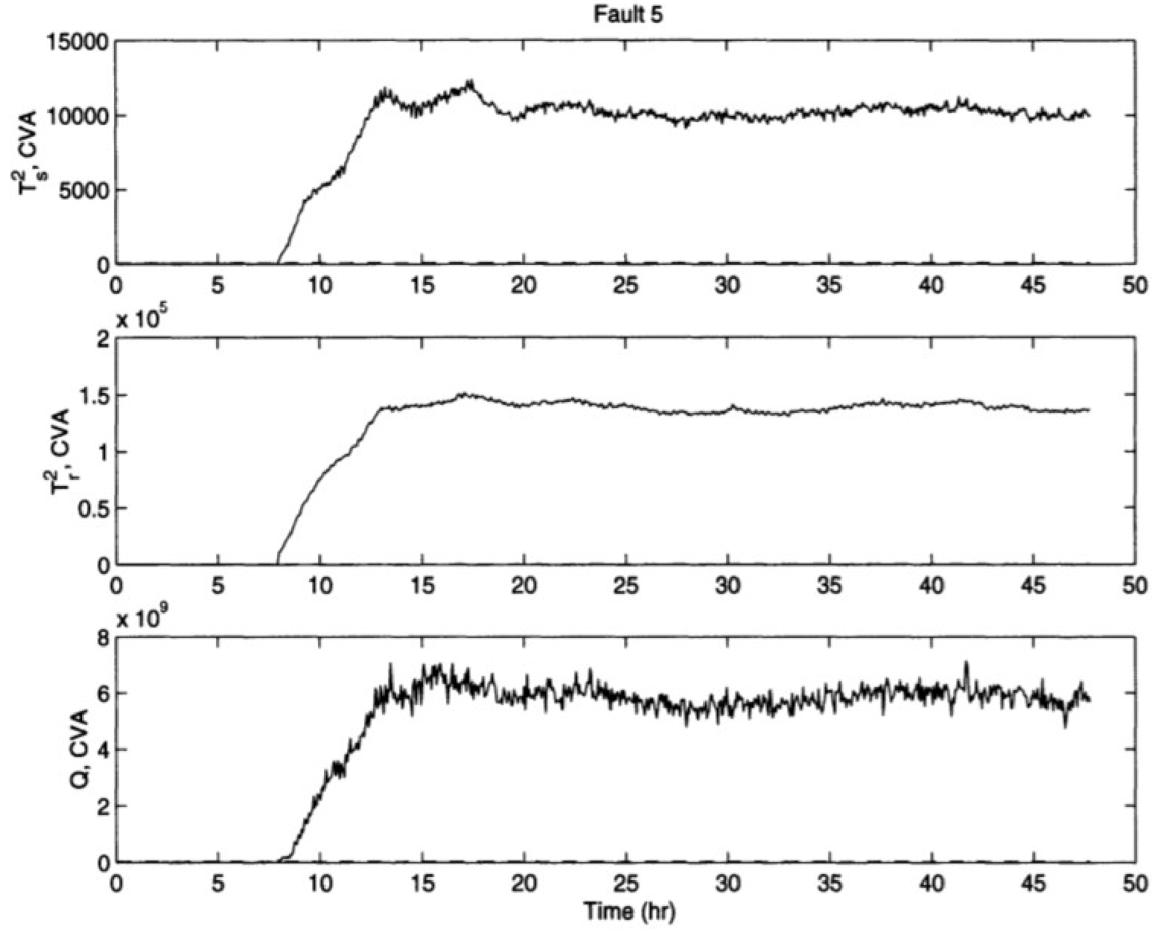


Figure 2.5 The CVA multivariate statistics for fault detection for Fault 5 (E.L. Russell et al., 2001)

Missed detection rates and detection delays are shown on Tables 2.6 and 2.7 and PCA - values of these Tables are the values, which has been recreated on the other study which went side-by-side with this thesis. There was a code created which ran through the faults and the similar results with these were able to be regenerated. This was important part for the success of this study and this made it possible to compare the results against the Russell's results reliably. Chapter 4 on this thesis contains the results with event-driven approach for TEP and the comparable results against these are shown there.

Table 2.6 Missed detection rates for the testing set (E.L. Russell et al., 2001)

Fault	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q
1	**0.008**	**0.003**	**0.006**	**0.005**	**0.001**	**0**	**0.003**
2	0.020	0.014	0.019	0.015	0.011	0.010	0.026
3	0.998	0.991	0.991	0.990	0.981	0.986	0.985
4	**0.956**	**0.038**	**0.939**	**0**	**0.688**	**0**	**0.975**
5	**0.775**	**0.746**	**0.758**	**0.748**	**0**	**0**	**0**
6	0.011	0	0.013	0	0	0	0
7	0.085	0	0.159	0	0.386	0	0.486
8	0.034	0.024	0.028	0.025	0.021	0.016	0.486
9	0.994	0.981	0.995	0.994	0.986	0.993	0.993
10	0.666	0.659	0.580	0.665	0.166	0.099	0.599
11	**0.794**	**0.356**	**0.801**	**0.193**	**0.515**	**0.195**	**0.669**
12	0.029	0.025	0.010	0.024	0	0	0.021
13	0.060	0.045	0.049	0.049	0.047	0.040	0.055
14	0.158	0	0.061	0	0	0	0.122
15	0.988	0.973	0.964	0.976	0.928	0.903	0.979
16	0.834	0.755	0.783	0.708	0.166	0.084	0.429
17	0.259	0.108	0.240	0.053	0.104	0.024	0.138
18	0.113	0.101	0.111	0.100	0.094	0.092	0.102
19	0.996	0.873	0.993	0.735	0.849	0.019	0.923
20	0.701	0.550	0.644	0.490	0.248	0.087	0.354
21	0.736	0.570	0.644	0.558	0.440	0.342	0.547

Table 2.7 Detection delays (minutes) for the testing set (E.L. Russell et al., 2001)

Fault	PCA T^2	PCA Q	DPCA T^2	DPCA Q	CVA T_s^2	CVA T_r^2	CVA Q
1	**21**	**9**	**18**	**15**	**6**	**9**	**6**
2	51	36	48	39	39	45	75
3	–	–	–	–	–	–	–
4	–	**9**	**453**	**3**	**1386**	**3**	–
5	**48**	**3**	**6**	**6**	**3**	**3**	**0**
6	30	3	33	3	3	3	0
7	3	3	3	3	3	3	0
8	69	60	69	63	60	60	63
9	–	–	–	–	–	–	–
10	288	147	303	150	75	69	132
11	**912**	**33**	**585**	**21**	**876**	**33**	**81**
12	66	24	9	24	6	6	0
13	147	111	135	120	126	117	129
14	12	3	18	3	6	3	3
15	–	2220	–	–	2031	–	–
16	936	591	597	588	42	27	33
17	87	75	84	72	81	60	69
18	279	252	279	252	249	237	252
19	–	–	–	246	–	33	–
20	261	261	267	252	246	198	216
21	1689	855	1566	858	819	1533	906

3 EVENT-DRIVEN APPROACHES

3.1 Introduction to Event-driven filtering

Event-driven filtering is an important part of data processing in a wide field of engineering systems and it was one of the main parts of this thesis. Term “event”, generally means a change in the state of a variable in this category of a subject. And an event model for an IoT system can be produced by the repetition of an event value. Events are also used to model sampled data and time-series data.

The effects of event-driven filtering can be seen clearly in Figure 3.3 where the original data has been filtered to a form, which includes only the significant data. The range where the values of the data are between upper and lower limit means, that there is no fault in the system and from the system and fault detection point of view, this data is not meaningful. There is no point to transfer that part of the data in the system and the event-driven filtering is focusing only to the events which do matter. This way the amount of data can be highly reduced. (R.J. Doyle et al., 1993; D. Serpanos et al., 2018)

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a data reduction technique which was used in the process, where the results of this thesis were meant to offer resampled data sets. The idea behind dimensionality reduction techniques, is to improve process monitoring procedures and characterizing the state of the process is to be made by projecting the data into lower-dimensional space.

According to L.H. Chiang, E.L. Russell and R.D. Braatz, PCA has three major qualities that helps to automate the process monitoring procedures:

- “PCA can produce lower-dimensional representations of the data which better generalize to data independent of the training set than that using the entire dimensionality of the observation space, and therefore, improve the proficiency of detecting and diagnosing faults.” (L.H. Chiang et al., 2001)
- “The structure abstracted by PCA can be useful in identifying either the variables responsible for the fault and/or the variables most affected by the fault.” (L.H. Chiang et al., 2001)

- “PCA can separate the observation space into a subspace capturing the systematic trends of the process and a subspace containing essentially the random noise.” (L.H. Chiang et al., 2001)

Fault detection of the PCA method can be implemented with examination of statistics T^2 and Q. Both statistics were used in this thesis to compare the behaviour of the original data sets and the resampled compressed data sets. In some situations, T^2 -statistics tends to represent in-control behaviour of the process inaccurately and for those situations, Q -statistics suits better.

The T^2 -statistic, presented in Equation 2, can be calculated from the PCA representation. There is a downside with T^2 -statistic if the quantity of the available data is relatively small and the number of observation variables is large though. This appears to effect the in-control process behaviour by making the representation of it inaccurate. Because of the fact that the square of the singular values is inverted in, these inaccuracies with smaller singular values have a big effect on the calculated T^2 -statistic. (L.H. Chiang et al., 2001)

Equation for the T^2 -statistic is next:

$$T^2 = x^T V (\sum\text{^}T\sum)^{-1} V^T x \ , \tag{2}$$

For the detection of the deviations in the TEP training set, the threshold though can be derived from the Equation 3.

$$T_\alpha^2 = \frac{(n-1)^2 \frac{a}{(n-a-1)} F_\alpha(a,n-a-1)}{n(1+\frac{a}{n-a-1}) F_\alpha(a,n-a-1)} \ , \tag{3}$$

The Q -statistic, presented in Equation 4, is also known as Squared Prediction Error, SPE. It is a squared 2-norm which measures the deviation of the observations to the lower-dimensional PCA representation and it is more robust way than the T^2 -statistic to monitor smaller singular values. This is a result of a feature which prevents it to suffer from an oversensitivity to inaccuracies. On the equation, r is the residual vector which is a projection of the observation x into the residual space. (L.H. Chiang et al., 2001)

$$Q = r^T r \quad \text{and} \quad r = (I - PP^T)x\,, \tag{4}$$

The distribution for the Q statistic, Equation 5, has been approximated by J.E. Jackson and G.S. Mudholkar in their study of control procedures for residuals. (J.E. Jackson et al., 1979)

$$Q_\alpha = \theta_1 \left[\frac{h_0 c_\alpha \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2}\right]^{\frac{1}{h_0}}, \tag{5}$$

On the Equation 5, c_α is a the normal deviate which corresponds to the $(1 - \alpha)$ percentile. Other parts of the equation can be seen on Equation 6:

$$\theta_i = \sum_{j=a+1}^{n} \sigma_j^{2i} \quad \text{and} \quad h_0 = 1 - \frac{2\theta_1\theta_3}{3\theta_2^2}\,, \tag{6}$$

The threshold α for the Q statistic can be computed by using Equation 5. and it is used for fault detection. (J.E. Jackson et al., 1979)

The detection of different types of faults is an advantage of the combination of T^2 and Q statistics along with their appropriate thresholds used together as side-by-side measures. The utilizing of these statistics T^2 and Q, along with their respective thresholds, can be described as a cylindrical in-control region which is presented in Figure 3.1 (L.H. Chiang et al., 2001)

In presentation of fault detection in Figure 3.1, symbols have next meanings:

- x -data = were collected during in-control operations
- o -data = T^2 statistic violation
- + -data = Q statistic violation

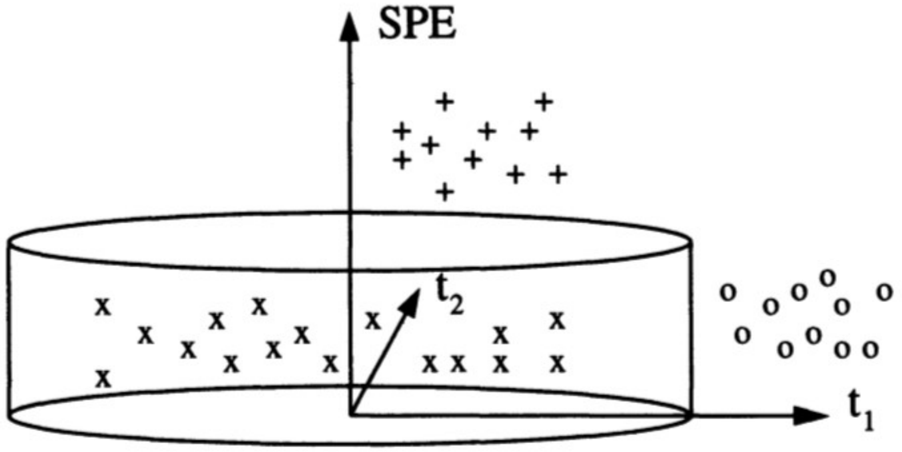


Figure 3.1 Presentation of fault detection using Q and T^2 -statistics (L.H. Chiang et al., 2001)

3.3 Resampling data sets

3.3.1 Threshold based event-driven method

Resampling of the data sets with Threshold based event-driven TEP -process was made with Python -coding language and the actual code for this procedure can be found on Appendix 2.; Code for resampling data sets. It is also available for download in the management locations of the study in GitHub, which were mentioned in the Introduction, Section 1.3 Research tools.

Figure 3.2 represents the resampling process with Threshold method as a flow chart and the process is described more precisely on next sentences:

- Normal operation data without a failure was loaded to the code
- Sample values, the number of samples and variables were named for the code
- Parameters were extracted from the data: max, min and mean -values were determined from the data until all of the 52 variables were gone through
- fault operation data was loaded from TEP and the threshold percentages were defined (data compression of 5%, 10%, 20%, 30% … 90% and 95%)
- Each sample value from the fault operation data was compared to the max and min values from the normal operation data
- If the sample value was bigger than the max value from the normal operation data multiplied by percentage or if the sample value was smaller than the min value from the normal operation data multiplied by percentage, there happened a transmit and the value from the original fault file without percentage resampling was maintained
- Otherwise there was no transmit and the value was resampled by the mean -value
- Finally, there was obtained a resampled fault operation data and it was named by the percentage there was used
- After that, the code moved to next fault operation data file and this was repeated until there was a pair for every original fault operation data file and with every compression rate

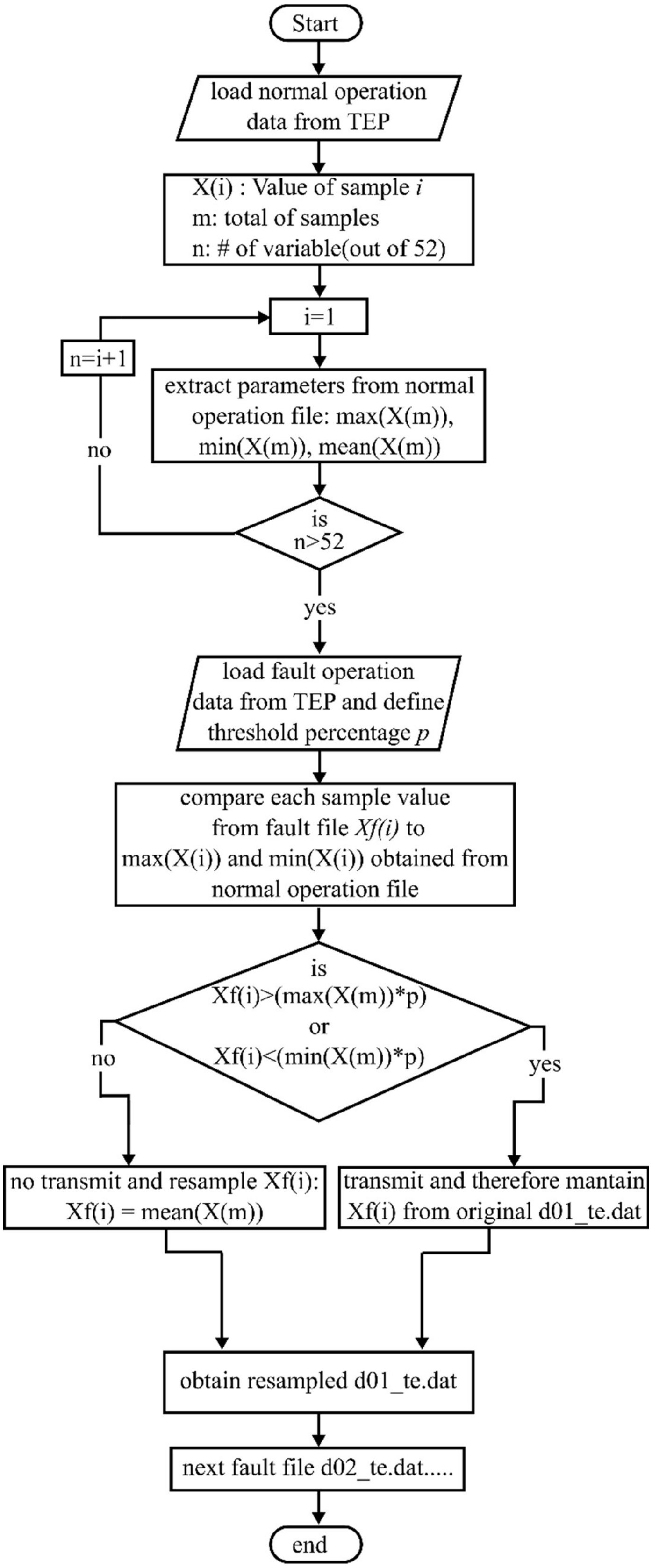


Figure 3.2 Flow chart of resampling the data sets with Threshold method

The end-result of the amount of the fault data files which were after this tested and studied in the PCA -method study was:

- 21pcs. of original fault data files
- 21pcs. of resampled fault data files with 5% compression ratio
- 21pcs. of resampled fault data files with 10% compression ratio
- 21pcs. of resampled fault data files with 20% compression ratio
- 21pcs. of resampled fault data files with 30% compression ratio
- 21pcs. of resampled fault data files with 40% compression ratio
- 21pcs. of resampled fault data files with 50% compression ratio
- 21pcs. of resampled fault data files with 60% compression ratio
- 21pcs. of resampled fault data files with 70% compression ratio
- 21pcs. of resampled fault data files with 80% compression ratio
- 21pcs. of resampled fault data files with 90% compression ratio
- 21pcs. of resampled fault data files with 95% compression ratio

Total amount of the original fault data files and the resampled fault data files with compression rate -changes created for the PCA -study

- Original data: 21 fault data files
- Resampled data: 231 fault data files

Original and resampled data is available in GitHub in https://github.com/pedrohjn/event-tep.

Graphical example of the original fault data file and the resampled fault data files of Threshold method with few different compression ratios is available on Figure 3.3 and for this example, there was selected Fault 1.

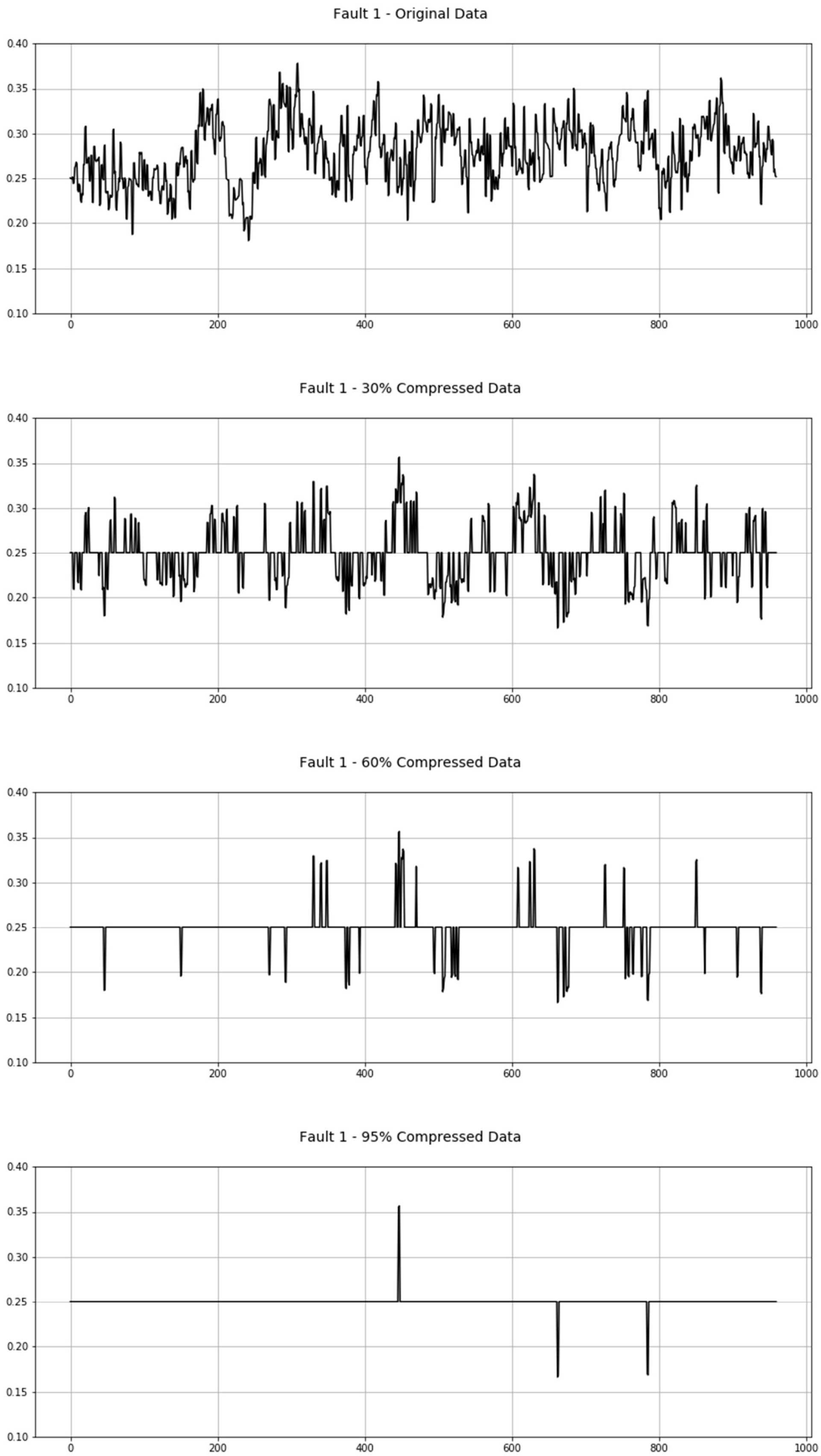


Figure 3.3 Threshold method resampled data sets of Fault 1 – Original / 30% / 60% / 95%

More specific example of resampling the data by the upper and lower limits is shown in Figure 3.4. From the Figure 3.5 can be seen how the resampling ignores the data which shows that there is no fault. Whenever the sample value passes one of the limits, the resampled data keeps the values of those moments.

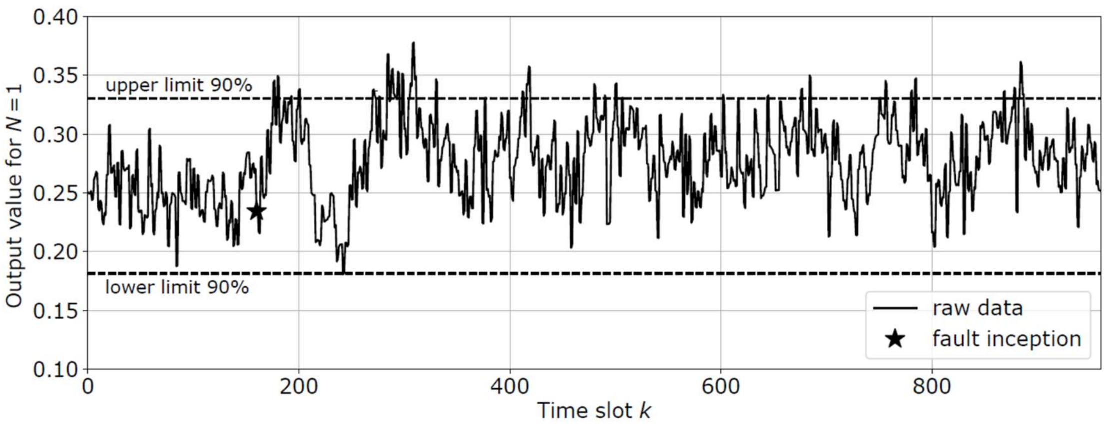


Figure 3.4 Original data set with upper and lower limits

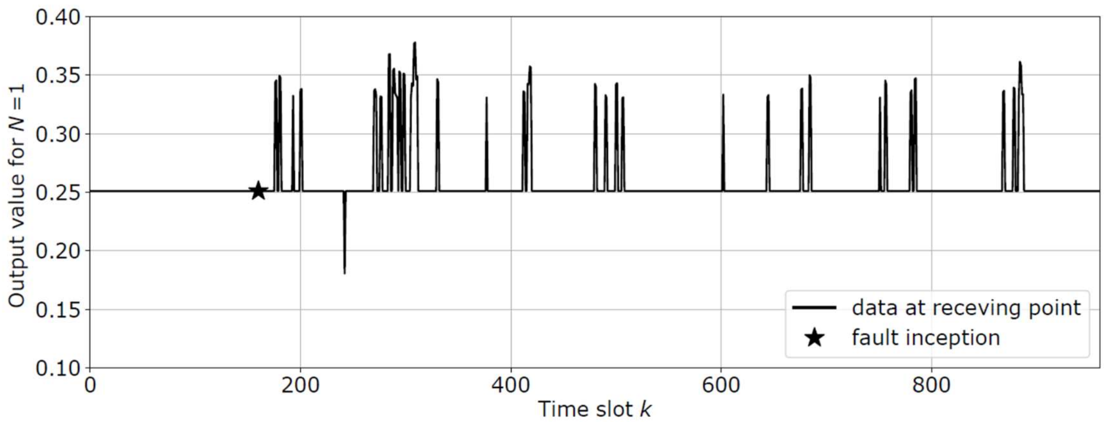


Figure 3.5 Resampled data set where data under 90% limit (normal behaviour) is filtered

3.3.2 Delta based event-driven method

This study was started with the Threshold method and during the study became an idea for testing also other event-driven method called Delta. Delta method is one of the most natural sampling methods and like it is typical for event-driven methods, data collection if triggered only in certain values. In Delta method the triggering values occurs when the signal deviation is more than delta and the delta is defined from the most recent sample by the change of its value. This method has been chosen for many smart sensor devices, especially wireless sensors with LonWork -technology which uses event detection. The idea behind the LonWorks -technology is to keep the node sleeping to prevent unnecessary energy consumption and only if the values of the sensor data crosses the limits, the node wakes up. While sleeping, the sensor can also be programmed to listen or to ignore incoming messages by the value and target of the messages. (M. Miskowicz et al., 2006; M. Miskowicz, 2015)

This method differs quite a lot from Threshold method and in this approach, the file of normal operation is taken as a base for generating the data set and the biggest change from sample to sample is being calculated. This can be also described as the difference or Delta between the sample t and the sample $t + 1$. The maximum difference is being stored and it is used for generating new data files by the following rule: If the difference between a sample $t - (t + 1)$ is greater than the maximum Delta which was earlier calculated, the data or the sample is being transmitted forward. Otherwise the previous value of t for $t + 1$ is being maintained as well.

Resampling of the data sets with Delta based event-driven TEP -process was also made with Python -coding language and the actual code for this procedure can be found on Appendix 3.; Code for resampling data sets. It is also available for download in the management locations of the study in GitHub, which were mentioned in the Introduction, Section 1.3 Research tools. Also the same kind of graphical example of the original fault data file and the resampled fault data files with few different compression ratios was created for Delta method like it was created for Threshold method. Figure 3.6 shows Delta method -version as a comparison and also for this example, Fault 1 was selected.

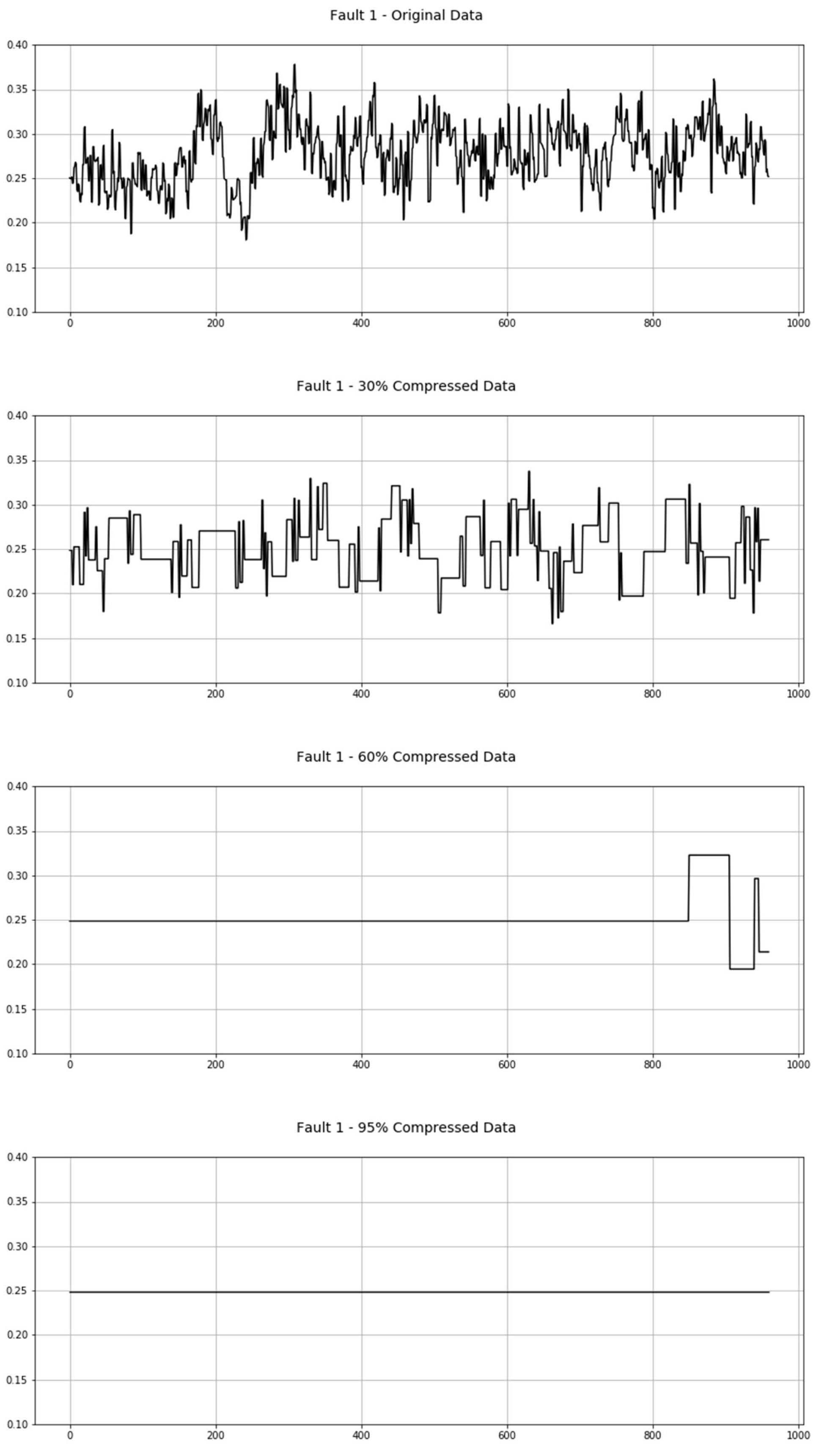


Figure 3.6 Delta method resampled data sets of Fault 1 – Original / 30% / 60% / 95%

Figure 3.7 represents the resampling process with Delta method as a flow chart and the process is described more precisely on next sentences:

- Normal operation data without a failure was loaded to the code
- Sample values, the number of samples and variables were named for the code
- Consecutive variation of each sample was extracted from the normal operation data; Delta variable was taken from the difference of the sample t and the sample $t + 1$ and this was continued until the value of sample has reached the value 960
- After that the max variation was calculated, the value of the variable was added by 1. These two functions were continued until the value of the variable has been risen up to 52
- Next consecutive variation of each sample was extracted from the fault operation data 01 and each of them was compared to Delta max from the fault file
- If the Delta max of the normal data was smaller than the Delta max of the fault data, no transmission was occurred and the sample was resample by the earlier sample value, otherwise there happened a transmit and the value from the original fault file without percentage resampling was maintained
- Finally, there was obtained a resampled fault operation data and it was named by the percentage there was used
- After that, the code moved to next fault operation data file and this was repeated until there was a pair for every original fault operation data file and with every compression rate

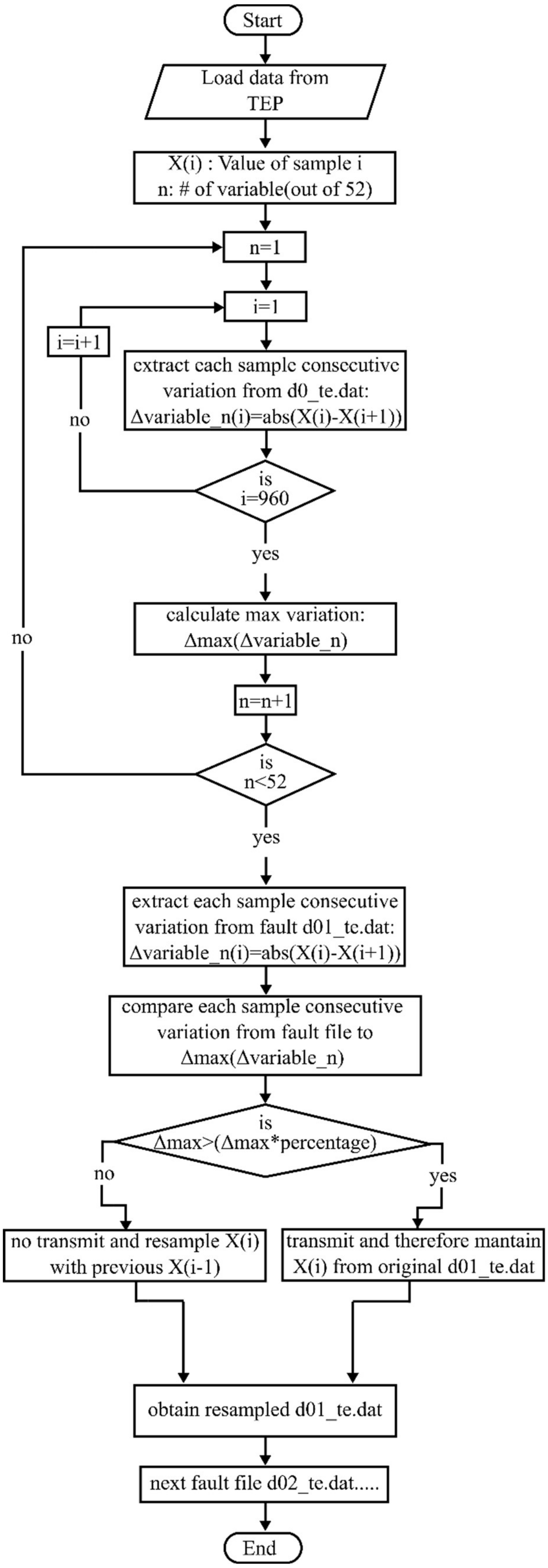


Figure 3.7 Flow chart of resampling the data sets with Delta method

3.4 Event-driven filtering in Tennessee Eastman process

In this thesis, as a method of event-driven filtering with Threshold and Delta methods, the idea of the method was the same as in any other case. Here the target for the procedure was fault data sets, more specific 21 of them and the procedure was repeated multiple times with different compression ratios. Figures 3.3 and 3.6 shows actual figures from the resampled data sets of the thesis and these were made for Fault 1 and the comparison example shows the data figures between original data set, 30% compressed, 60% compressed and 95% compressed data sets.

Next figures shows the average number of samples that needs to be transmitted via the communication network with the data sets made by Threshold method in Figure 3.8, considering the event-driven acquisition using a percentage threshold of the max/min values in normal operation and the compression ratio for all the variables in this baseline case in Figure 3.9 with different percentages of thresholds. These figures were created from the data which was used in this thesis.

Without the event-driven method, there will always be 52 samples to be transmitted in parallel. On the other hand, if the proposed event-driven method is based on a very restrictive threshold, like filtering 95% of the training set, then less than 10 samples need to be transmitted in average. This can be considered to cover all fault data sets. The greater the threshold, the lower the number of samples to be acquired, and the lower the traffic to be offered to the network. It must be noted that this relation is not linear though.

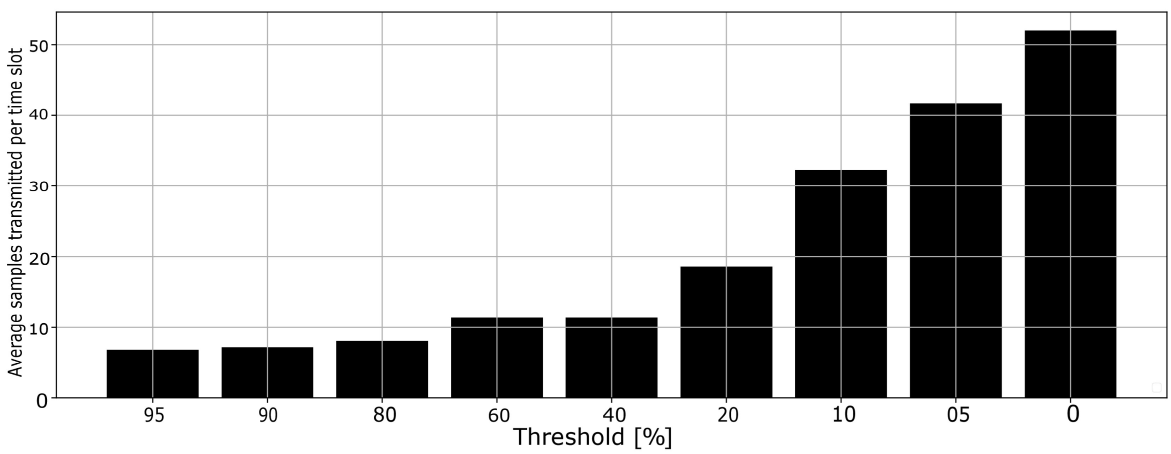


Figure 3.8 Threshold method: Average number of samples that needs to be transmitted

The compression ratio increases as the threshold gets closer to the max value of each variable. It can be seen from that the compression ratio is a non-linear function of the threshold that is selected, therefore the selection of the most suitable threshold is not straightforward and will depend on the end-application to be defined.

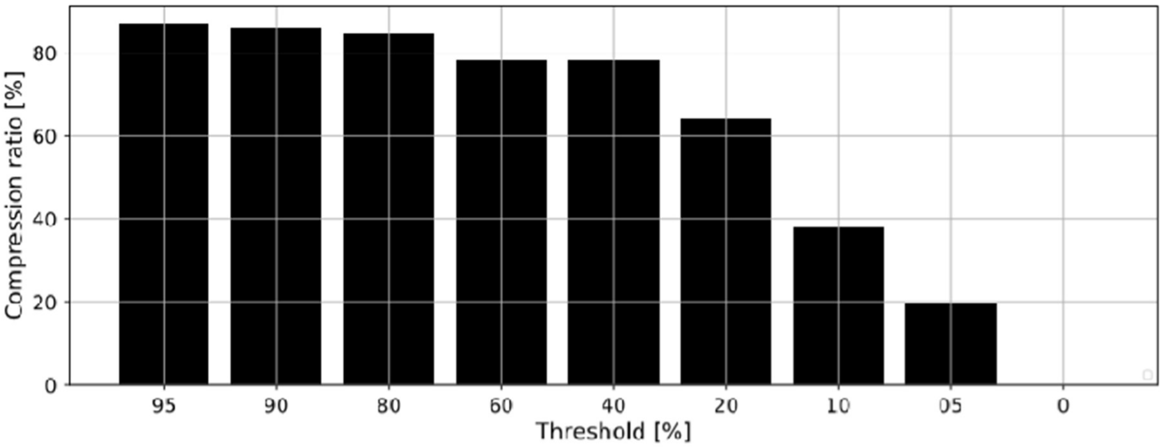


Figure 3.9 Threshold method: Compression ratio with different percentages of thresholds

Next figures shows the average number of samples that needs to be transmitted via the communication network with the data sets made by Delta method in Figure 3.10, considering the event-driven acquisition using a Delta max percentage in normal operation and the compression ratio for all Delta max values in this baseline case in Figure 3.11. These figures were created from the data which was used in this thesis.

Without the event-driven method, there will always be 52 samples to be transmitted in parallel. On the other hand, if the Delta based event-driven method is based on a very high percent, like filtering 95% of the training set, then less than 10 samples need to be transmitted in average. This can be considered to cover all fault data sets. The greater the Delta max, the lower the number of samples to be acquired, and the lower the traffic to be offered to the network. It must be noted that this relation is not linear though.

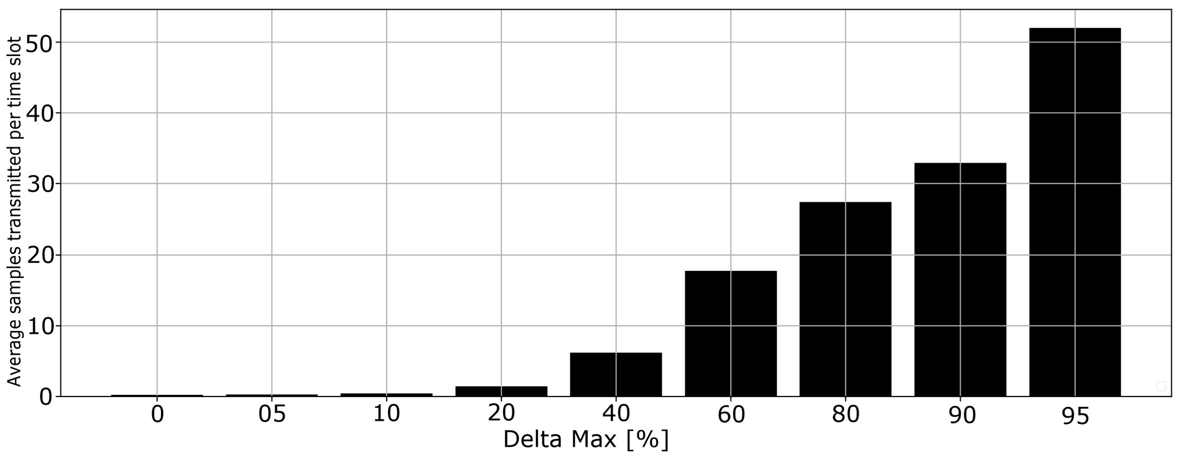


Figure 3.10 Delta method: Average number of samples that needs to be transmitted

The compression ratio increases as the Delta max gets closer to the max value of each variable. It can be seen from that the compression ratio is a non-linear function of the Delta that is selected; therefore the selection of the most suitable Delta max is not straightforward and will depend on the end-application to be defined.

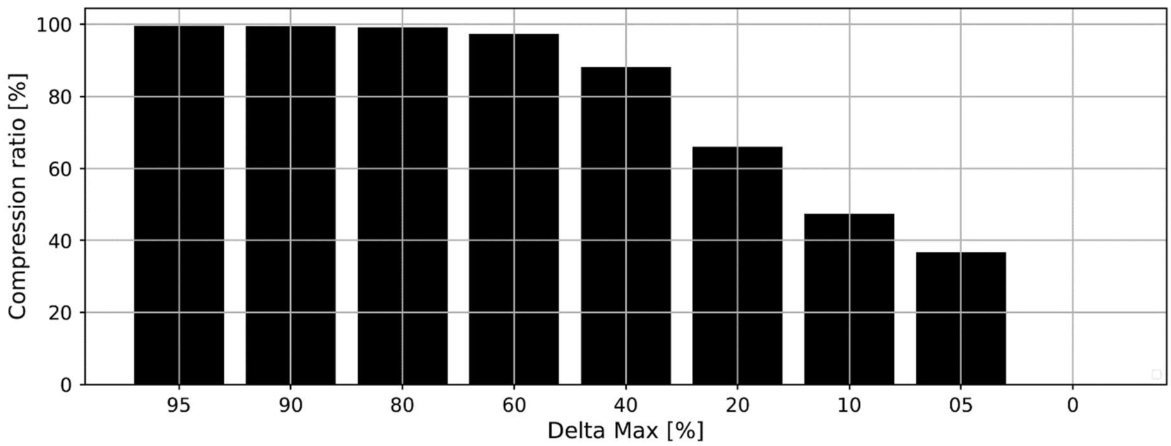


Figure 3.11 Delta method: Compression ratio with different percentages of Delta Max %

4 RESULTS

4.1 Results on Threshold method with event-driven approach in TEP

Event-driven approach with Threshold method in TEP didn't bring any revolutionary results in this study or in the testing part of it with the target study and PCA -method but it did prove that if the compression rate of the resampled data was kept on moderate level, event-driven approach in TEP didn't make the fault detection worse. It didn't make the fault detection better either but the known fact of the impact on the system by reducing the load and the transmitted data, study was a success.

As graphical results, Figures 4.1, 4.2, 4.3 and 4.4 describes the control charts of Faults 5 and 21 for T^2 and Q -statistics, including original data files and resampled data files for 30%, 70% and 95% compression ratios. As the figures clearly show, the effects of the event-driven approach in TEP do not have a major role in control charts which means that there is no major effects on the fault detection.

Control charts for T^2 and Q -statistics are available as figures in GitHub. These charts are generated for every fault with the original data and with the resampled data with every compression ratio.

The naming for the figures and data sets of Threshold method in GitHub, as an example, goes by the information of original / resampled data & compression ratio & fault file & chart type:

- d05_te.dat
    - Original data, Fault 5
- NEW_70_d05_te.dat
    - Resampled data, Compression ratio 70%, Fault 5
- Failure_05 - Q.png
    - Original data, Fault 5, Control chart for Q
- NEW_70_Failure_05 - T2.png
    - Resampled data, Compression ratio 70%, Fault 5, Control chart for T^2

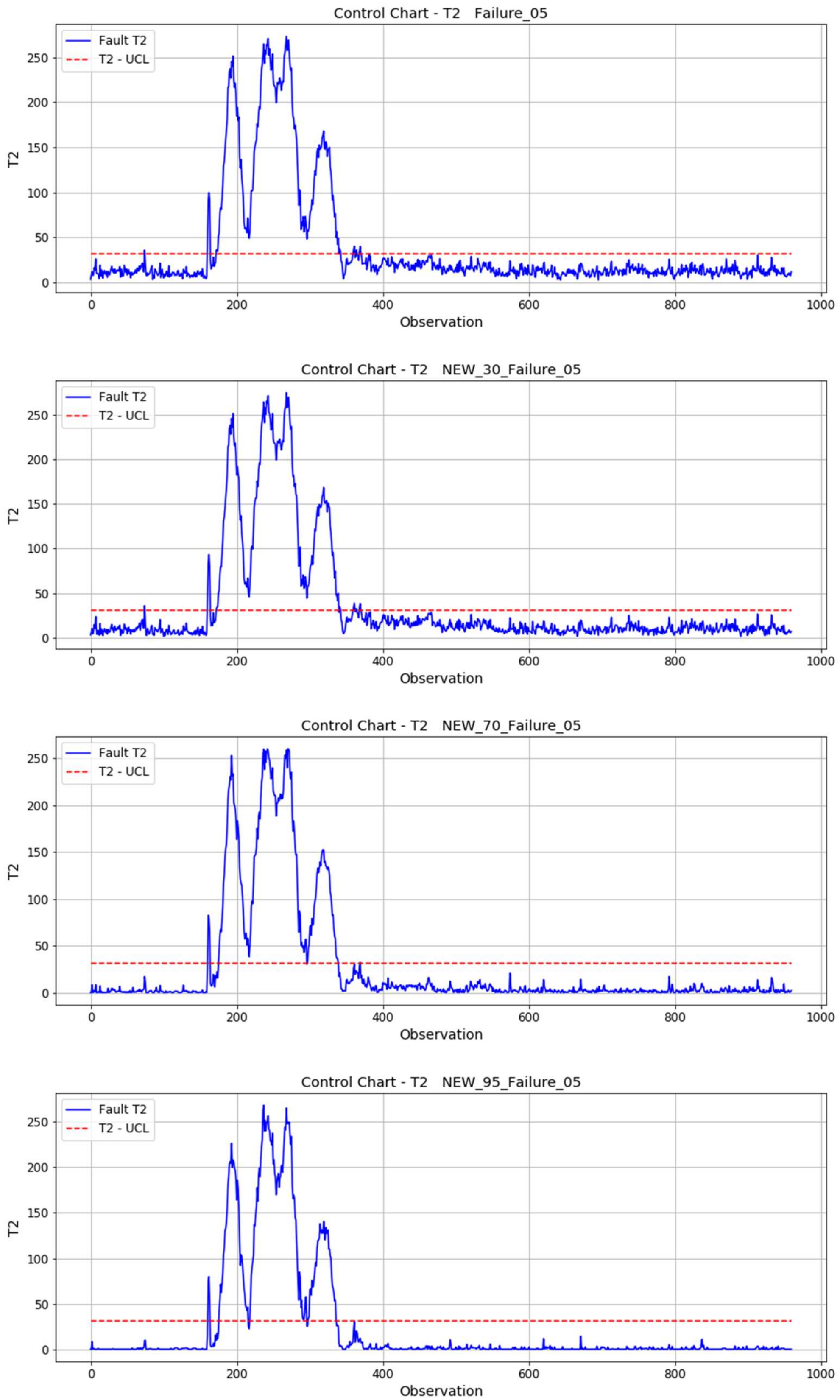


Figure 4.1 Threshold method: Control chart of T^2 for the Fault 5 – Orig. / 30 / 70 / 95 %

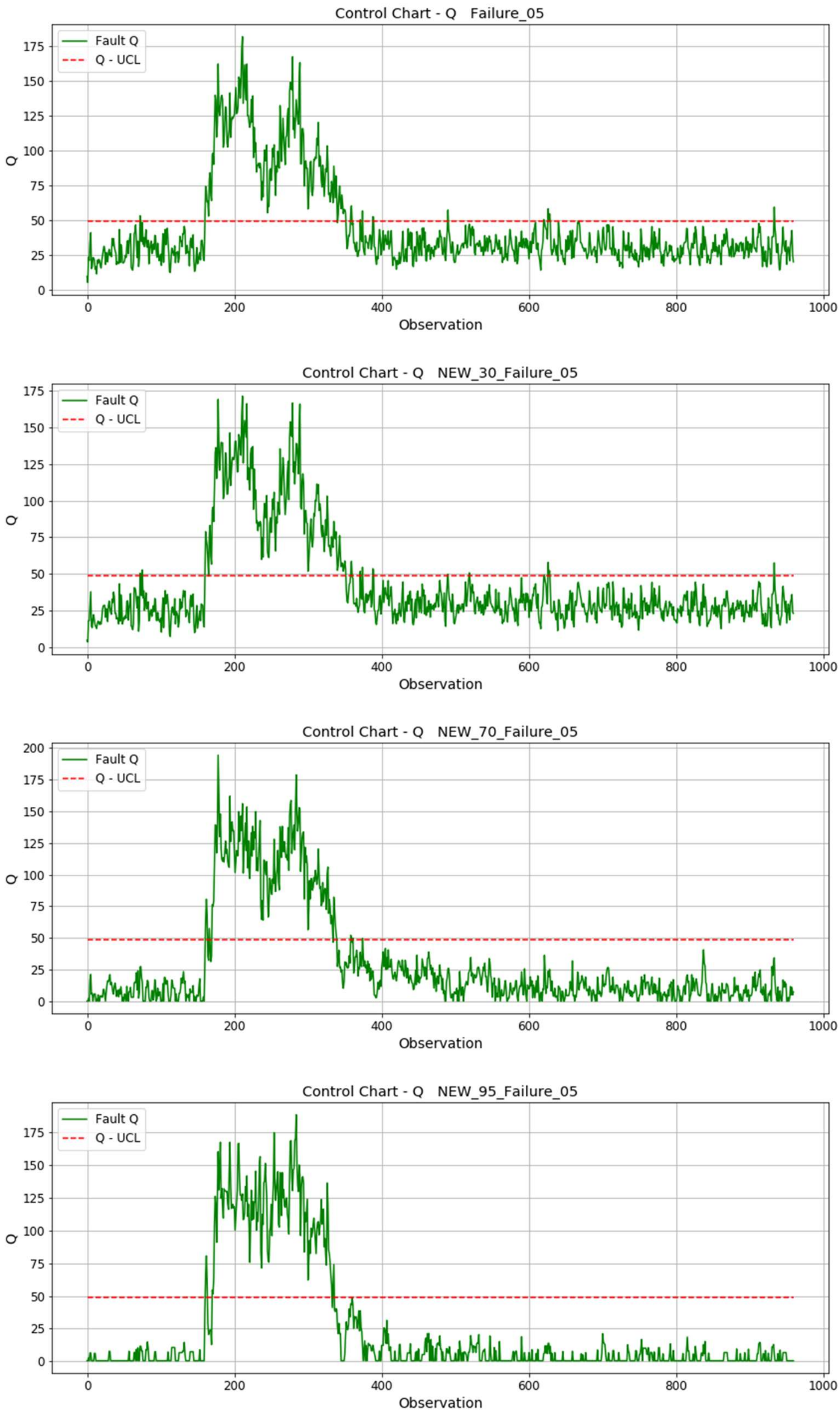


Figure 4.2 Threshold method: Control chart of Q for the Fault 5 – Orig. / 30 / 70 / 95 %

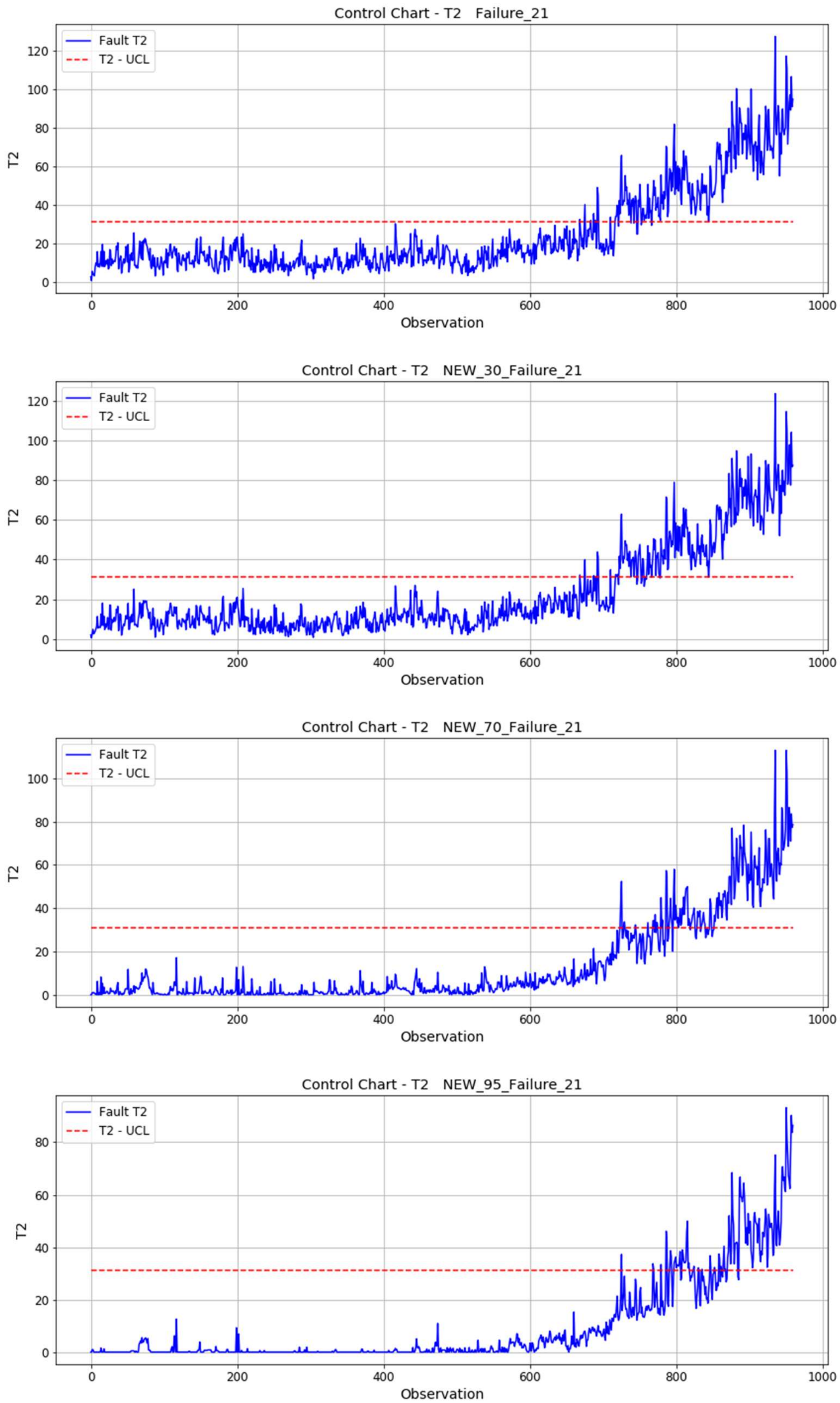


Figure 4.3 Threshold method: Control chart of T^2 for the Fault 21 – Orig. / 30 / 70 / 95 %

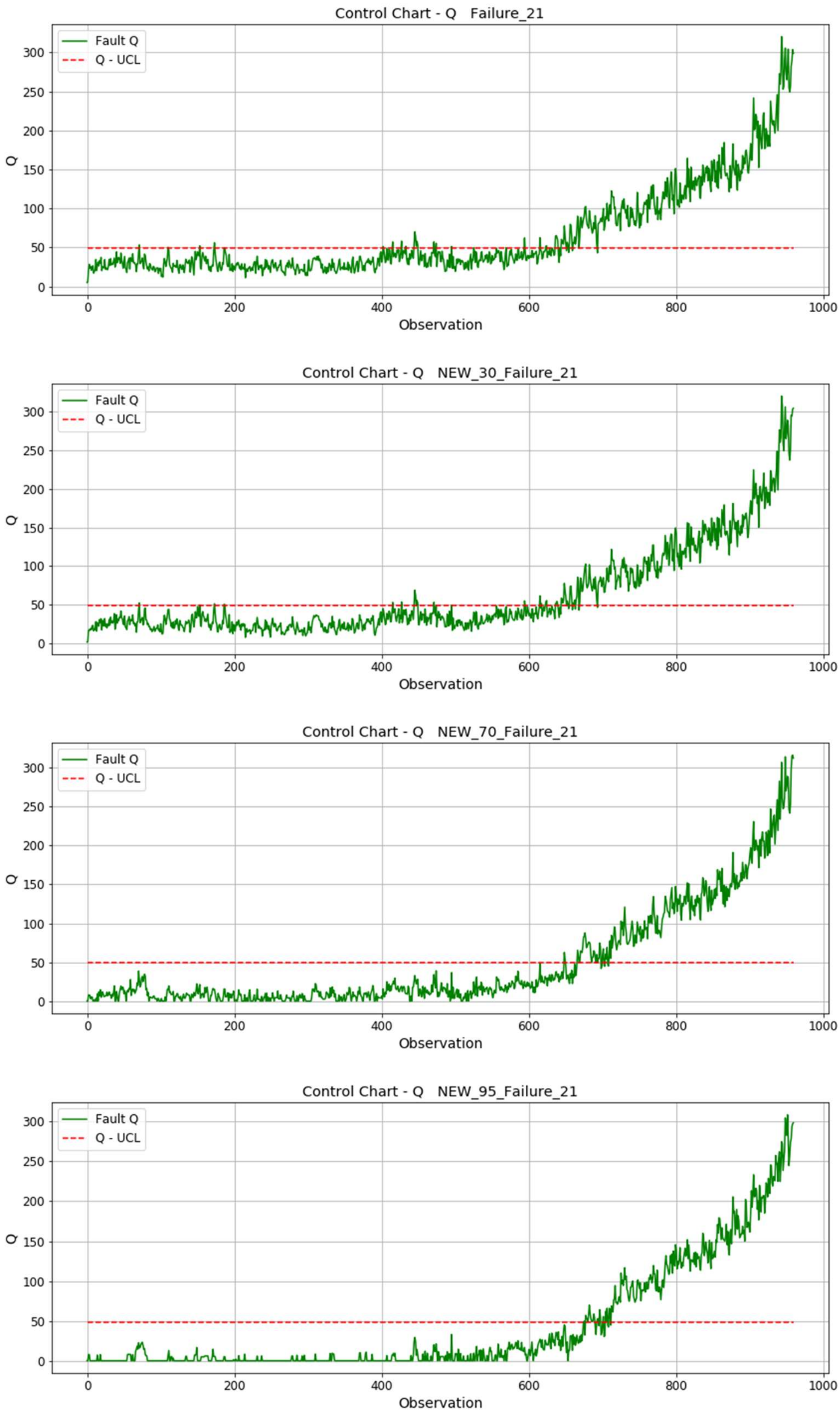


Figure 4.4 Threshold method: Control chart of Q for the Fault 21 – Orig. / 30 / 70 / 95 %

4.2 Results on Delta method with event-driven approach in TEP

Event-driven approach with Delta method in TEP had quite a huge difference, even when compared to Threshold method. It did bring some interesting results in this study and in the testing part of it with the target study and PCA -method. It had a remarkable effect on the fault detection and with best Delta percentages, there we only couple of undetected faults in the system and at the same time, amount of data was still lowered greatly.

As graphical results, Figures 4.5, 4.6, 4.7 and 4.8 describes the control charts of Faults 5 and 21 for T^2 and Q -statistics, including original data files and resampled data files for 30%, 70% and 95% compression ratios. It shows clearly on the figures that the effects of the event-driven approach in TEP do not have a major role in control charts but it did make a huge difference on fault detection.

Control charts for T^2 and Q -statistics are available as figures in GitHub. These charts are generated for every fault with the original data and with the resampled data with every compression ratio.

The naming for the figures and data sets of Delta method in GitHub, as an example, goes by the information of original / resampled data & compression ratio & fault file & chart type:

- d05_te.dat
  - Original data, Fault 5
- DEL_70_d05_te.dat
  - Resampled data, Compression ratio 70%, Fault 5
- Failure_05 - Q.png
  - Original data, Fault 5, Control chart for Q
- DEL_70_Failure_05 - T2.png
  - Resampled data, Compression ratio 70%, Fault 5, Control chart for T^2

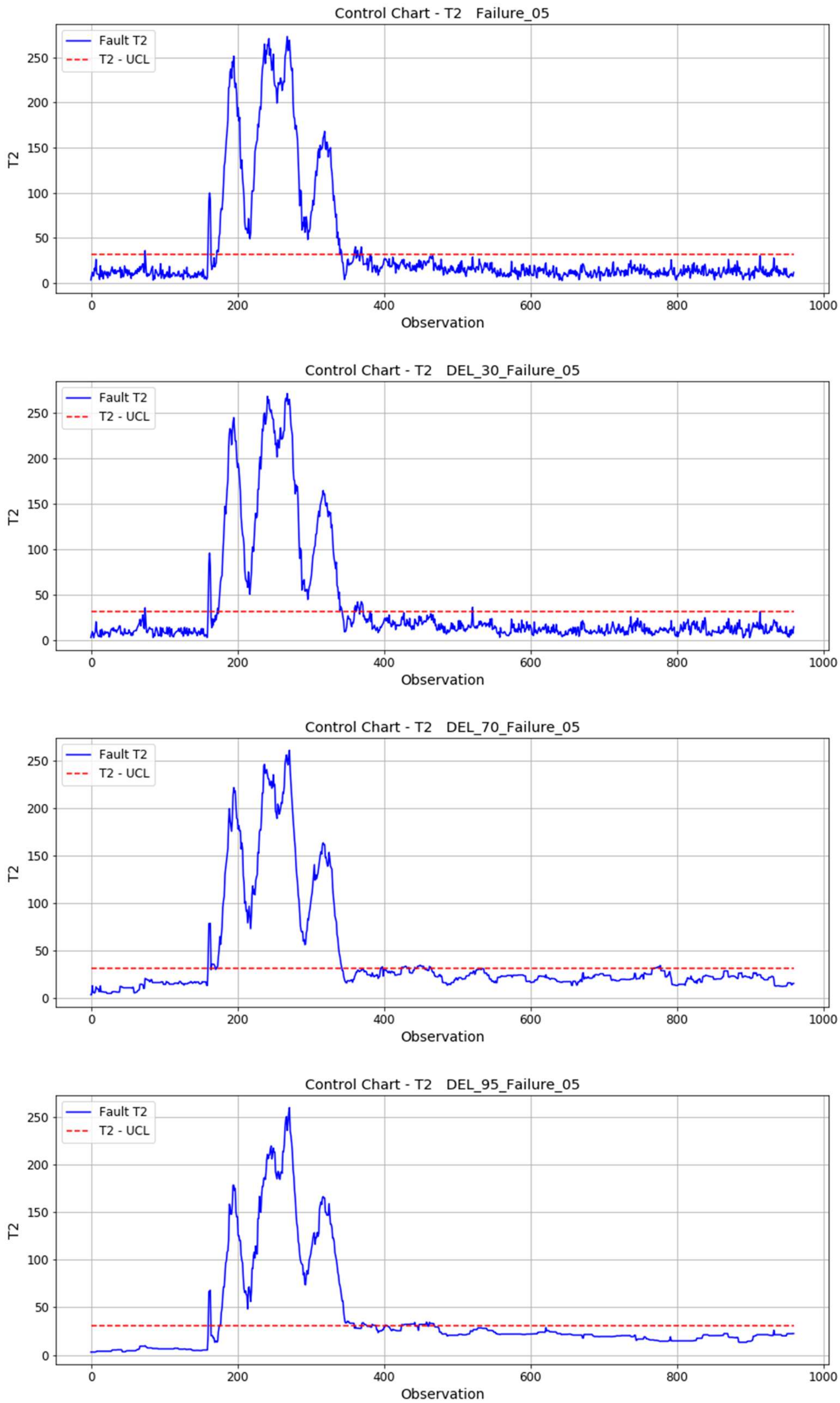


Figure 4.5 Delta method: Control chart of T^2 for the Fault 5 – Orig. / 30 / 70 / 95 %

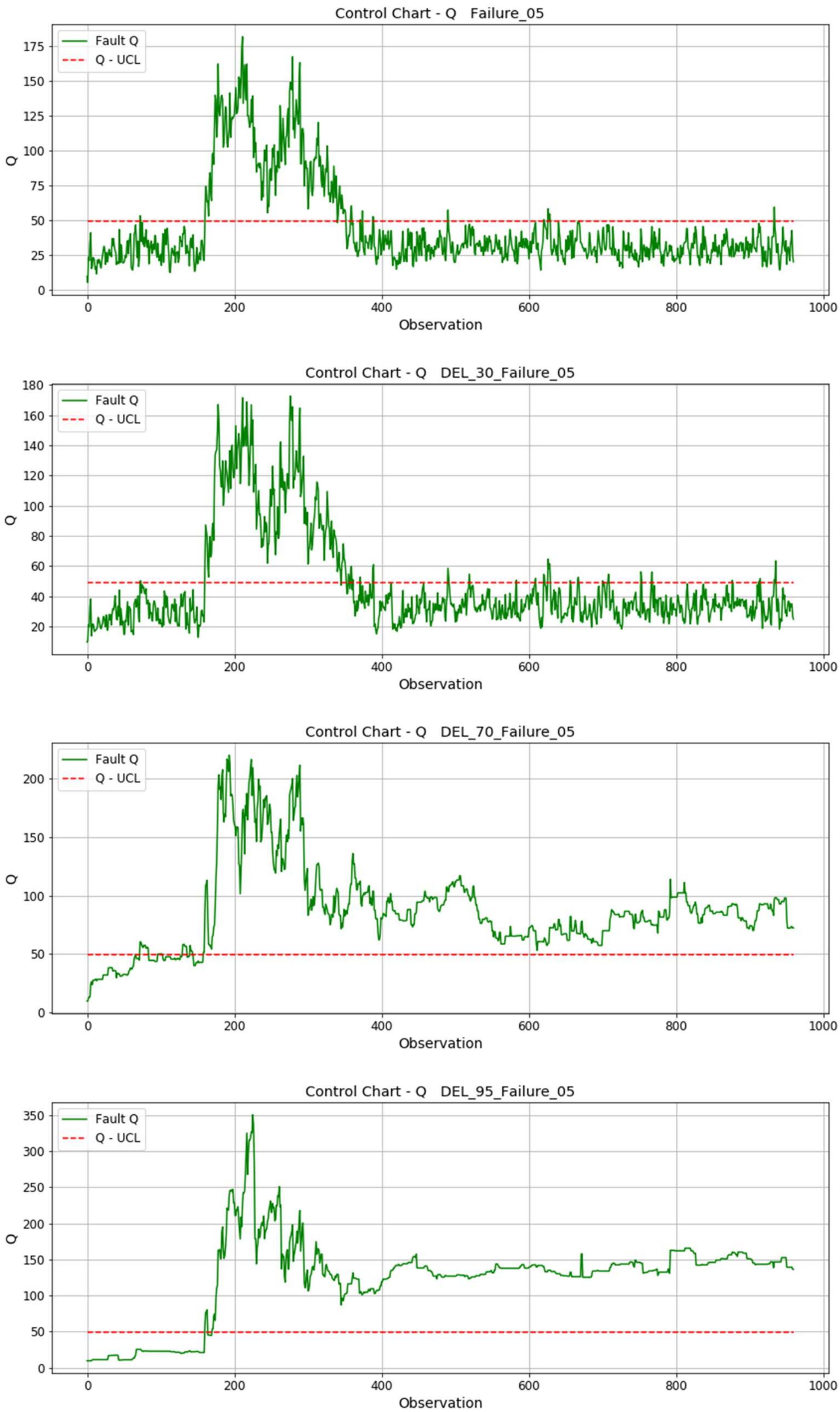


Figure 4.6 Delta method: Control chart of Q for the Fault 5 – Orig. / 30 / 70 / 95 %

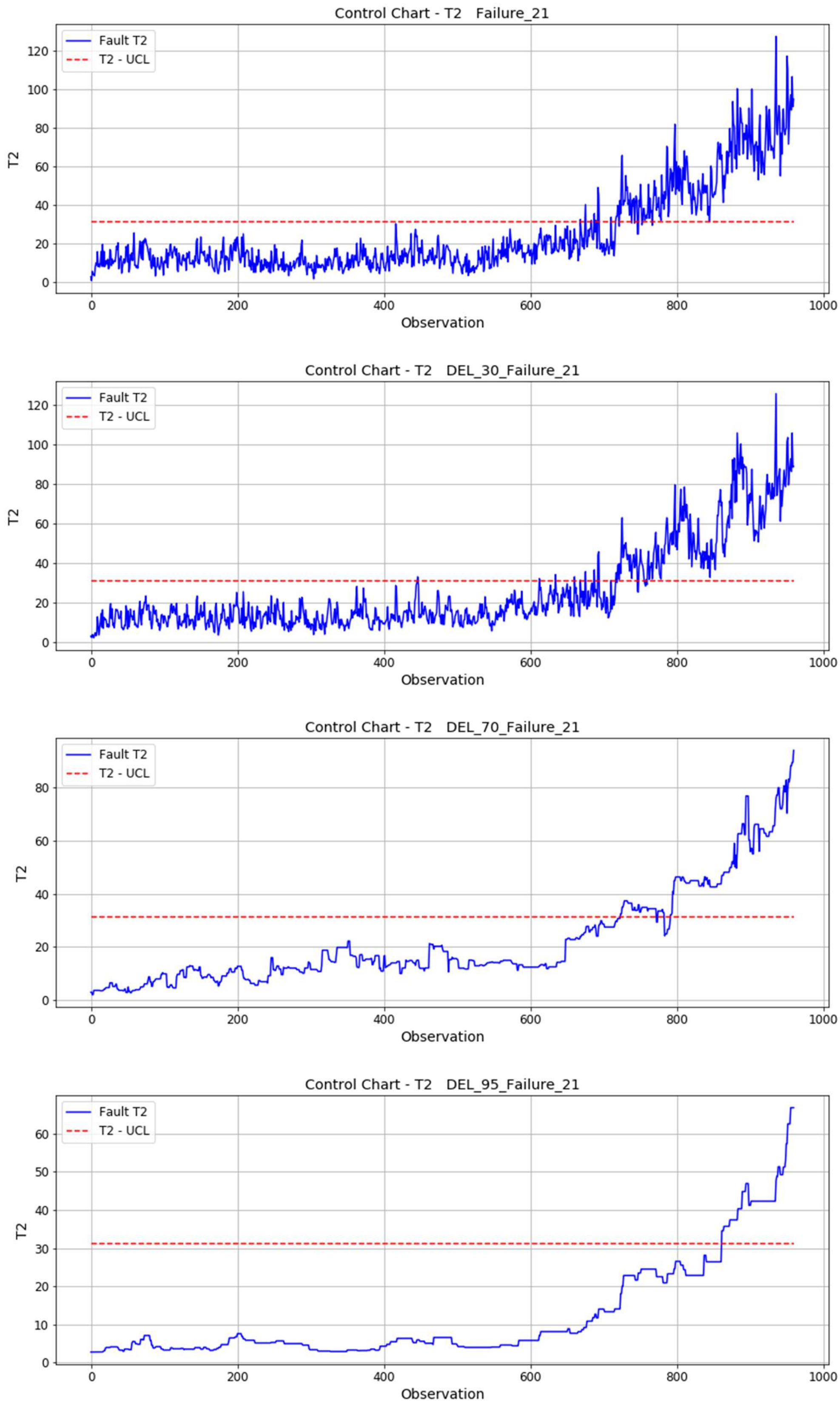


Figure 4.7 Delta method: Control chart of T^2 for the Fault 21 – Orig. / 30 / 70 / 95 %

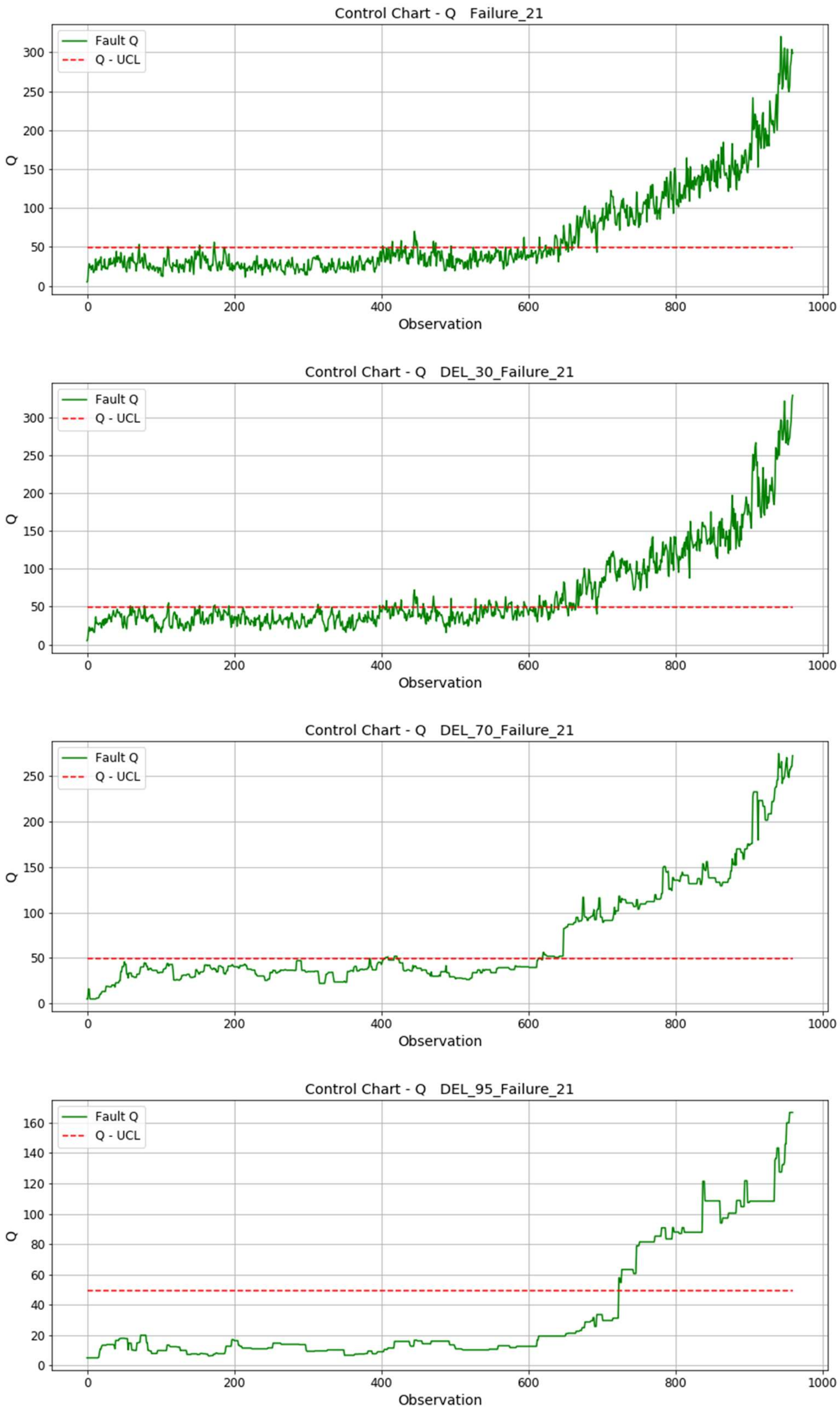


Figure 4.8 Delta method: Control chart of Q for the Fault 21 – Orig. / 30 / 70 / 95 %

4.3 Fault detection analysis

4.3.1 Detection analysis with Threshold method

Russell's original data and all of the new resampled data were analysed and compared against each other. The code for the tests with principal component analysis can be found as an Appendix 4 in the end of this thesis. The analysis shows clearly that those new compressed data sets are almost as great for detecting faults as the original, but they are using a lot of less data. This smaller amount of data was already at the beginning of the thesis described as valuable and necessary feature and improvement and the fact that this doesn't make the effectivity of the fault detection worse, is a success.

Although, there is a limit for the data compression and if the compression goes for too high percent's, the effectiveness starts to suffer. In this analysis, it was shown that with 80% of compression, there rise a new fault which the system was not able to detect. The speed and everything else stayed fairly similar but 70% compression ratio was the highest one to maintain the same number of detected faults. From the Tables of detection analysis can be read, that Faults 3, 4, 9, 15 and 19 are the ones which the system didn't detect in any cases, even from the original data. A rough figure for the shapes of those faults can be seen on below in Figure 4.9.

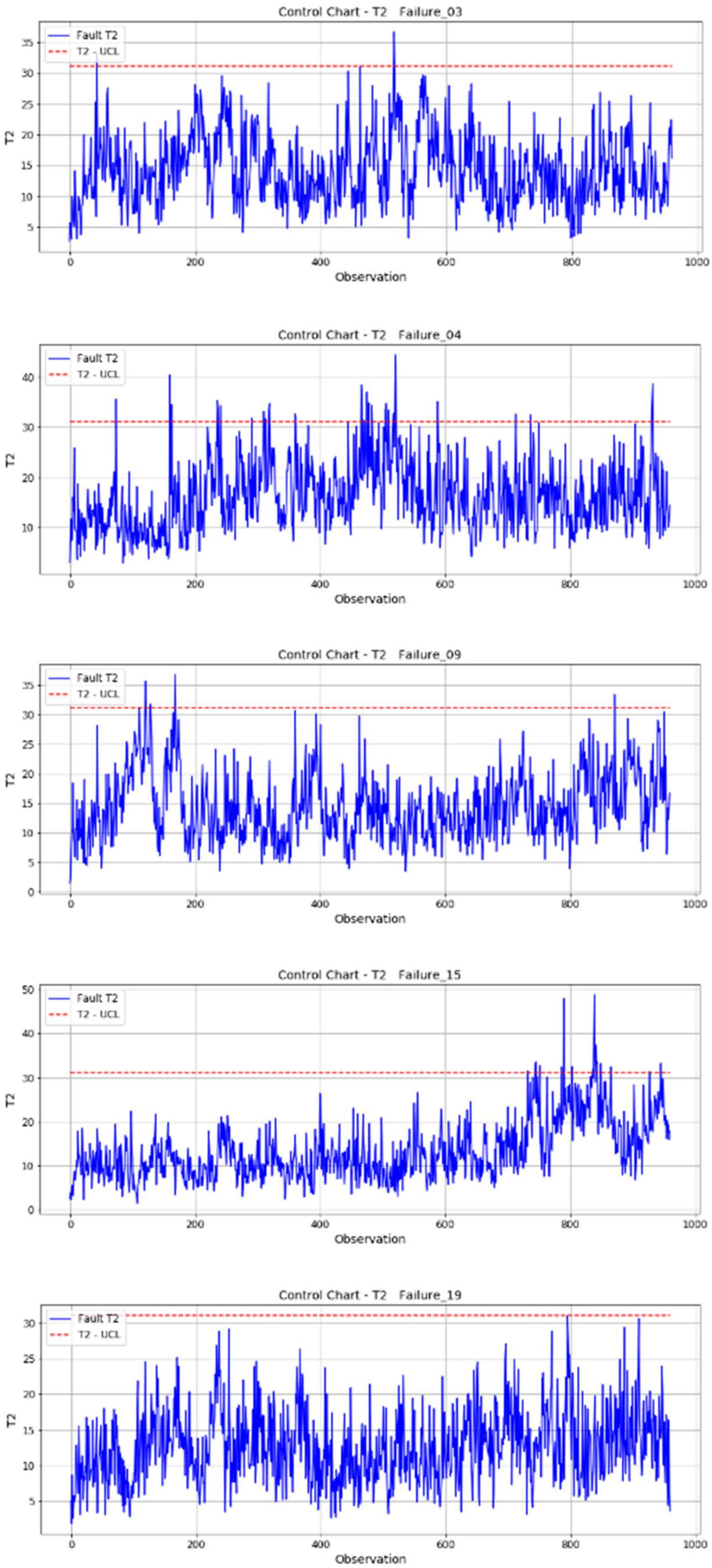


Figure 4.9 Undetected faults of Threshold method: Faults 3, 4, 9, 15 and 19

4.3.2 Detection analysis with Delta method

Russell's original data and all of the new resampled data which was made on Delta method were also analysed and compared against each other. The analysis shows clearly that those new compressed data sets are better for detecting faults as the original and they are using a lot of less data. Although, there is a limit for the data compression and if the compression goes for too high percent's, the effectiveness starts to suffer.

In this analysis, it was shown that with under 50% and over 70% of compression ratio there were faults which couldn't be detected. The speed and everything else stayed fairly similar but 50%, 60% and 70% of the compression ratios were the highest ones to maintain the same number of detected faults. From the Tables of detection analysis can be read, that mainly Faults 3 and 19 are the ones which the system didn't detect in any cases, except on one. That can be seen on Table 4.2 and it was the Fault 19 and the compression ratio of 70% from the original data. A rough figure for the shapes of those undetected faults can be seen on below in Figure 4.10.

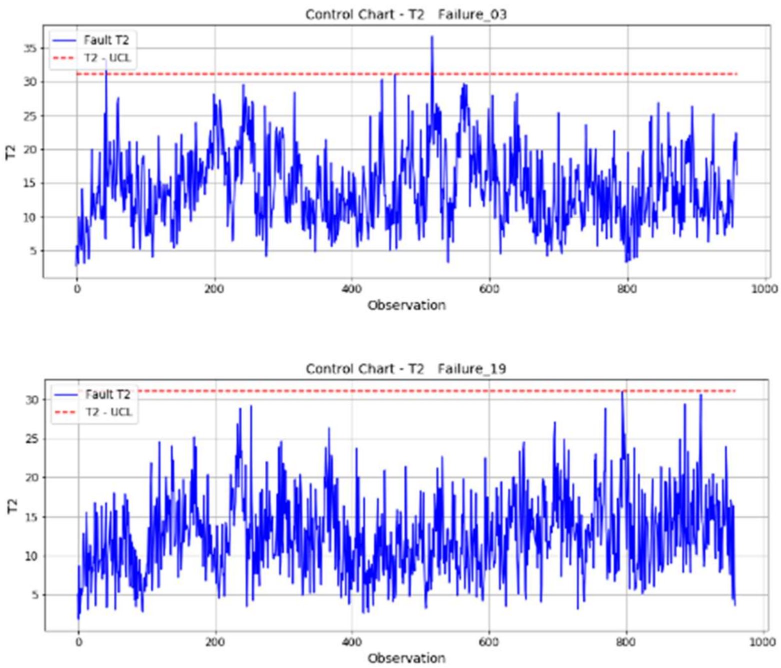


Figure 4.10 Undetected faults of Threshold method: Faults 3 and 19

4.3.3 Detection analysis comparison of Threshold and Delta methods

Comparison between detection analysis of Threshold method and Delta method is described with the following Tables. Explanations for the used terms in these Tables are next:

- The event detection is defined as six subsequent samples, which are considered faulted. The fault is considered detected at the index of the first sample of the sequence.
- All faults are inserted at a time-slot of 160 and the process stops at a time-slot of 960. This means that a value of 960 in the Index-column is a sign, that the fault was not recognized.
- **Index*** refers to time slots with a length of 3 minutes from where the fault is detected.
- **Delay**** refers to how many time slots will it take from the methods to detect the fault.
- **MDR***** defines how many samples are not classified as faulted after the 160 time-slot, divided by 800.
- **FA****** defines how many samples are classified as faulted before the 160 time-slot, divided by 160.
- **Red colour** informs if the fault is not detected

Table 4.1 Threshold method and T² -statistics: Comparison of Index*

Fault	Index* - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	167	167	167	167	167	167	167	167	167	167	167	167
2	177	177	177	177	177	177	177	178	177	177	179	179
3	960	960	960	960	960	960	960	960	960	960	960	960
4	960	960	960	960	960	960	960	960	960	960	960	960
5	173	173	173	173	173	174	176	177	177	177	177	177
6	170	170	170	170	170	170	170	170	171	171	171	171
7	161	161	161	161	161	161	161	161	161	161	161	161
8	183	183	183	183	183	183	183	183	185	185	185	187
9	960	960	960	960	960	960	960	960	960	960	960	960
10	256	256	256	256	256	256	258	261	266	266	266	266
11	464	464	464	464	463	464	545	545	545	960	960	960
12	182	182	182	182	182	182	182	182	182	182	188	188
13	209	209	209	209	209	209	209	209	215	217	217	217
14	164	164	164	164	164	164	164	164	164	165	165	165
15	960	960	960	960	960	960	960	960	960	960	960	960
16	472	472	472	472	472	472	472	472	472	472	472	472
17	189	189	189	191	191	191	191	191	191	191	191	191
18	253	253	253	253	255	255	254	254	254	255	255	255
19	960	960	960	960	960	960	960	960	960	960	960	960
20	247	247	247	247	247	247	247	249	249	249	250	250
21	723	723	723	723	723	723	740	724	797	797	798	798

Table 4.2 Delta method and T² -statistics: Comparison of Index*

Fault	Index* - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	167	167	167	167	167	167	167	167	167	167	169	169
2	177	177	177	177	177	179	177	178	179	177	179	179
3	960	960	960	960	960	960	960	960	960	960	960	960
4	960	960	960	960	960	960	225	630	607	960	960	960
5	173	173	173	173	176	175	174	161	173	161	175	179
6	170	170	170	170	170	170	170	167	168	168	168	168
7	161	161	161	161	161	161	161	161	161	161	161	161
8	183	183	183	183	183	185	183	185	185	185	187	187
9	960	960	960	960	960	960	941	942	894	960	960	960
10	256	256	256	256	233	234	256	255	256	259	261	262
11	464	464	464	464	357	357	357	256	485	306	306	306
12	182	182	182	182	182	182	182	182	182	182	190	182
13	209	209	209	209	209	209	209	211	210	211	218	219
14	164	164	164	164	164	165	165	166	166	166	166	166
15	960	960	960	960	960	960	806	941	960	960	960	960
16	472	472	472	473	472	470	468	465	446	470	470	471
17	189	189	189	190	189	189	189	191	189	191	191	191
18	253	253	253	253	249	253	254	259	259	254	259	259
19	960	960	960	960	960	960	960	960	604	960	960	960
20	247	247	247	247	247	247	250	247	250	250	251	251
21	723	723	723	718	720	718	706	674	724	726	856	862

Table 4.3 Threshold method and Q -statistics: Comparison of Index*

Fault	Index* - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	163	163	163	163	163	164	164	164	165	165	165	166
2	172	172	172	172	173	173	173	175	175	175	175	175
3	960	960	960	960	960	960	960	960	960	960	960	960
4	165	165	165	165	165	165	163	167	168	220	229	319
5	162	162	162	167	167	167	171	171	171	171	171	171
6	161	161	161	161	161	161	161	161	161	161	161	161
7	161	161	161	161	161	161	161	161	161	161	161	161
8	182	182	182	182	182	182	182	183	183	183	183	183
9	960	960	960	960	960	960	960	960	960	960	960	960
10	209	209	209	209	209	261	259	259	259	261	261	261
11	171	171	171	171	171	171	171	211	211	211	211	256
12	168	168	168	168	168	168	167	167	168	168	168	168
13	197	197	197	197	197	197	202	206	206	206	207	207
14	161	161	161	161	161	161	161	162	162	162	162	162
15	960	960	960	960	960	960	960	960	960	960	960	960
16	356	356	356	356	357	356	356	356	357	472	472	472
17	185	185	185	185	185	185	185	185	185	185	185	285
18	244	244	244	244	244	245	245	245	247	247	247	255
19	960	960	960	960	960	960	960	960	960	960	960	960
20	247	247	247	247	247	247	244	245	247	247	247	250
21	445	445	445	445	445	645	645	670	667	672	678	798

Table 4.4 Delta method and Q -statistics: Comparison of Index*

Fault	Index* - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	163	163	163	163	163	162	162	163	163	166	164	164
2	172	172	172	173	171	171	97	175	101	179	179	179
3	960	960	960	960	960	246	33	87	99	467	480	960
4	165	165	163	163	163	163	126	175	73	161	83	190
5	162	162	162	162	161	161	126	161	73	161	83	171
6	161	161	161	161	161	161	161	161	161	161	161	161
7	161	161	161	161	161	161	161	161	161	161	161	161
8	182	182	182	179	180	176	181	115	181	181	182	182
9	960	960	960	960	960	845	88	546	131	394	960	960
10	209	209	209	209	208	205	207	184	187	203	235	235
11	171	171	171	170	170	166	73	167	208	185	185	172
12	168	168	168	168	167	167	167	125	168	182	182	182
13	197	197	197	197	201	195	198	204	203	205	205	209
14	161	161	161	161	161	161	161	162	162	162	162	162
15	960	960	960	960	960	733	259	151	155	900	906	746
16	356	356	356	356	356	108	196	82	105	355	418	407
17	185	185	185	185	184	184	169	184	184	184	184	184
18	244	245	244	245	245	175	147	168	172	177	252	252
19	960	960	960	960	960	175	178	377	270	710	638	717
20	247	247	247	247	245	242	235	175	242	247	247	247
21	445	445	445	445	445	443	84	558	405	649	652	724

Table 4.5 Threshold method and T² -statistics: Comparison of Delay**

Fault	Delay** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	7	7	7	7	7	7	7	7	7	7	7	7
2	17	17	17	17	17	17	17	18	17	17	19	19
3	800	800	800	800	800	800	800	800	800	800	800	800
4	800	800	800	800	800	800	800	800	800	800	800	800
5	13	13	13	13	13	14	16	17	17	17	17	17
6	10	10	10	10	10	10	10	10	11	11	11	11
7	1	1	1	1	1	1	1	1	1	1	1	1
8	23	23	23	23	23	23	23	23	25	25	25	27
9	800	800	800	800	800	800	800	800	800	800	800	800
10	96	96	96	96	96	96	98	101	106	106	106	106
11	304	304	304	304	303	304	385	385	385	800	800	800
12	22	22	22	22	22	22	22	22	22	22	28	28
13	49	49	49	49	49	49	49	49	55	57	57	57
14	4	4	4	4	4	4	4	4	4	5	5	5
15	800	800	800	800	800	800	800	800	800	800	800	800
16	312	312	312	312	312	312	312	312	312	312	312	312
17	29	29	29	31	31	31	31	31	31	31	31	31
18	93	93	93	93	95	95	94	94	94	95	95	95
19	800	800	800	800	800	800	800	800	800	800	800	800
20	87	87	87	87	87	87	87	89	89	89	90	90
21	563	563	563	563	563	563	580	564	637	637	638	638

Table 4.6 Delta method and T² -statistics: Comparison of Delay**

Fault	Delay** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	7	7	7	7	7	7	7	7	7	7	9	9
2	17	17	17	17	17	19	17	18	19	17	19	19
3	800	800	800	800	800	800	800	800	800	800	800	800
4	800	800	800	800	800	800	65	470	447	800	800	800
5	13	13	13	13	16	15	14	1	13	1	15	19
6	10	10	10	10	10	10	10	7	8	8	8	8
7	1	1	1	1	1	1	1	1	1	1	1	1
8	23	23	23	23	23	25	23	25	25	25	27	27
9	800	800	800	800	800	800	781	782	734	800	800	800
10	96	96	96	96	73	74	96	95	96	99	101	102
11	304	304	304	304	197	197	197	96	325	146	146	146
12	22	22	22	22	22	22	22	22	22	22	30	22
13	49	49	49	49	49	49	49	51	50	51	58	59
14	4	4	4	4	4	5	5	6	6	6	6	6
15	800	800	800	800	800	800	646	781	800	800	800	800
16	312	312	312	313	312	310	308	305	286	310	310	311
17	29	29	29	30	29	29	29	31	29	31	31	31
18	93	93	93	93	89	93	94	99	99	94	99	99
19	800	800	800	800	800	800	800	800	444	800	800	800
20	87	87	87	87	87	87	90	87	90	90	91	91
21	563	563	563	558	560	558	546	514	564	566	696	702

Table 4.7 Threshold method and Q -statistics: Comparison of Delay**

Fault	Delay** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)										
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	3	3	3	3	3	4	4	4	5	5	5	6
2	12	12	12	12	13	13	13	15	15	15	15	15
3	800	800	800	800	800	800	800	800	800	800	800	800
4	5	5	5	5	5	5	3	7	8	60	69	159
5	2	2	2	7	7	7	11	11	11	11	11	11
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	22	22	22	22	22	22	22	23	23	23	23	23
9	800	800	800	800	800	800	800	800	800	800	800	800
10	49	49	49	49	49	101	99	99	99	101	101	101
11	11	11	11	11	11	11	11	51	51	51	51	96
12	8	8	8	8	8	8	7	7	8	8	8	8
13	37	37	37	37	37	37	42	46	46	46	47	47
14	1	1	1	1	1	1	1	2	2	2	2	2
15	800	800	800	800	800	800	800	800	800	800	800	800
16	196	196	196	196	197	196	196	196	197	312	312	312
17	25	25	25	25	25	25	25	25	25	25	25	125
18	84	84	84	84	84	85	85	85	87	87	87	95
19	800	800	800	800	800	800	800	800	800	800	800	800
20	87	87	87	87	87	87	84	85	87	87	87	90
21	285	285	285	285	285	485	485	510	507	512	518	638

Table 4.8 Delta method and Q -statistics: Comparison of Delay**

Fault	Delay** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)										
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	3	3	3	3	3	2	2	3	3	6	4	4
2	12	12	12	13	11	11	-63	15	-59	19	19	19
3	800	800	800	800	800	86	-127	-73	-61	307	320	800
4	5	5	3	3	3	3	-34	15	-87	1	-77	30
5	2	2	2	2	1	1	-34	1	-87	1	-77	11
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	22	22	22	19	20	16	21	-45	21	21	22	22
9	800	800	800	800	800	685	-72	386	-29	234	800	800
10	49	49	49	49	48	45	47	24	27	43	75	75
11	11	11	11	10	10	6	-87	7	48	25	25	12
12	8	8	8	8	7	7	7	-35	8	22	22	22
13	37	37	37	37	41	35	38	44	43	45	45	49
14	1	1	1	1	1	1	1	2	2	2	2	2
15	800	800	800	800	800	573	99	-9	-5	740	746	586
16	196	196	196	196	196	-52	36	-78	-55	195	258	247
17	25	25	25	25	24	24	9	24	24	24	24	24
18	84	85	84	85	85	15	-13	8	12	17	92	92
19	800	800	800	800	800	15	18	217	110	550	478	557
20	87	87	87	87	85	82	75	15	82	87	87	87
21	285	285	285	285	285	283	-76	398	245	489	492	564

Table 4.9 Threshold method and T² -statistics: Comparison of MDR***

Fault	MDR*** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750
2	0.02000	0.02000	0.02000	0.01875	0.02000	0.01875	0.02000	0.02125	0.02000	0.02000	0.02250	0.02250
3	0.99875	0.99875	0.99875	0.99875	0.99875	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
4	0.96125	0.96125	0.96125	0.96875	0.98125	0.99375	0.99750	0.99875	0.99875	0.99875	0.99875	0.99875
5	0.77375	0.77375	0.77500	0.77500	0.77500	0.77875	0.78375	0.79000	0.79125	0.79500	0.80000	0.80000
6	0.01125	0.01125	0.01125	0.01125	0.01125	0.01125	0.01125	0.01125	0.01250	0.01250	0.01250	0.01250
7	0.07750	0.08125	0.07750	0.08000	0.09000	0.11625	0.13500	0.16125	0.18750	0.19375	0.21125	0.21625
8	0.03250	0.03250	0.03250	0.03250	0.03250	0.03250	0.03500	0.03625	0.04000	0.04000	0.04375	0.04875
9	0.99750	0.99750	0.99750	0.99875	0.99875	0.99875	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
10	0.65000	0.64750	0.64750	0.66375	0.68375	0.71000	0.73500	0.76250	0.79375	0.81125	0.85000	0.86125
11	0.78750	0.78750	0.78750	0.79250	0.80250	0.82375	0.85375	0.88000	0.89000	0.90625	0.91750	0.92000
12	0.02125	0.02125	0.02125	0.02125	0.02125	0.02375	0.02875	0.03250	0.03875	0.04250	0.06125	0.06875
13	0.06000	0.06000	0.06000	0.06000	0.06000	0.06000	0.06000	0.06000	0.06375	0.07125	0.07500	0.07750
14	0.12125	0.12000	0.12125	0.12125	0.13750	0.14875	0.16875	0.17625	0.19250	0.20625	0.21750	0.21750
15	0.97875	0.98125	0.98000	0.98375	0.99375	0.99625	0.99625	0.99625	0.99875	1.00000	1.00000	1.00000
16	0.82875	0.83125	0.83125	0.83875	0.84875	0.87375	0.89000	0.92125	0.93000	0.93375	0.93875	0.95000
17	0.25250	0.25250	0.25125	0.26000	0.26875	0.28250	0.30625	0.32500	0.34625	0.36625	0.37500	0.38000
18	0.11250	0.11250	0.11250	0.11250	0.11500	0.11500	0.11500	0.11625	0.11625	0.11750	0.11750	0.11750
19	1.00000	1.00000	0.99875	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
20	0.70125	0.70250	0.70500	0.70750	0.73000	0.74500	0.76250	0.78625	0.79875	0.81250	0.83375	0.83750
21	0.69375	0.69375	0.69500	0.69500	0.70250	0.71250	0.74000	0.75375	0.78750	0.81000	0.82500	0.83750

Table 4.10 Delta method and T² -statistics: Comparison of MDR***

Fault	MDR*** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.00750	0.01000	0.01000
2	0.02000	0.02000	0.02000	0.02000	0.02000	0.02250	0.02000	0.02000	0.02250	0.02250	0.02250	0.02250
3	0.99875	0.99875	0.99625	0.99625	1.00000	0.99250	0.99500	1.00000	0.99625	1.00000	1.00000	1.00000
4	0.96125	0.96125	0.95875	0.96000	0.97250	0.96875	0.93875	0.93375	0.90500	0.99875	0.99875	0.99875
5	0.77375	0.77375	0.77125	0.77000	0.76625	0.76375	0.76625	0.77250	0.72875	0.68250	0.60375	0.70125
6	0.01125	0.01125	0.01125	0.01125	0.01125	0.01125	0.01125	0.00750	0.00875	0.00875	0.00875	0.00875
7	0.07750	0.07875	0.07625	0.07875	0.06500	0.04500	0.05125	0.16125	0.00000	0.00000	0.00000	0.00000
8	0.03250	0.03250	0.03125	0.03375	0.03125	0.03250	0.03125	0.03125	0.03125	0.04250	0.05375	0.04500
9	0.99750	0.99750	0.99625	0.99125	0.99250	0.99750	0.98375	0.98375	0.97875	1.00000	1.00000	1.00000
10	0.65000	0.65250	0.65250	0.64625	0.65500	0.67125	0.67750	0.67875	0.69375	0.74375	0.80625	0.82625
11	0.78750	0.78875	0.78625	0.79625	0.79250	0.79375	0.76125	0.68750	0.87625	0.88625	0.89875	0.92625
12	0.02125	0.02125	0.02125	0.02250	0.01875	0.02375	0.02625	0.02125	0.03500	0.03375	0.05875	0.05625
13	0.06000	0.06000	0.06000	0.06125	0.06125	0.06125	0.06125	0.06250	0.06125	0.06250	0.07625	0.07750
14	0.12125	0.12000	0.12500	0.11125	0.11375	0.09625	0.09375	0.10625	0.09500	0.15375	0.14875	0.15375
15	0.97875	0.97875	0.97750	0.98375	0.98250	0.99250	0.94000	0.97375	0.98375	0.99625	1.00000	1.00000
16	0.82875	0.83250	0.83250	0.85125	0.84375	0.83375	0.86000	0.80875	0.73250	0.92000	0.93375	0.94625
17	0.25250	0.25375	0.25125	0.25750	0.26000	0.24875	0.26125	0.25500	0.27250	0.33750	0.26125	0.32375
18	0.11250	0.11250	0.11250	0.11125	0.11000	0.11250	0.11625	0.11750	0.11250	0.11250	0.12125	0.12125
19	1.00000	0.99875	0.99875	1.00000	0.99875	0.99625	0.99250	0.99750	0.88500	0.98750	0.99625	1.00000
20	0.70125	0.70375	0.71000	0.70875	0.71750	0.72500	0.68375	0.73125	0.65000	0.76625	0.75375	0.80500
21	0.69375	0.69375	0.69250	0.69125	0.69125	0.69625	0.69000	0.65125	0.71625	0.78625	0.86500	0.87625

Table 4.11 Threshold method and Q -statistics: Comparison of MDR***

Fault	MDR*** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.00250	0.00250	0.00250	0.00250	0.00250	0.00375	0.00375	0.00375	0.00500	0.00500	0.00500	0.00625
2	0.01375	0.01375	0.01375	0.01375	0.01500	0.01500	0.01500	0.01750	0.01750	0.01750	0.01750	0.01750
3	0.99000	0.99000	0.99000	0.98750	0.99375	0.99750	0.99875	1.00000	1.00000	1.00000	1.00000	1.00000
4	0.03625	0.03625	0.03625	0.04750	0.05875	0.10125	0.16125	0.31000	0.45250	0.55500	0.63000	0.65625
5	0.74500	0.74625	0.74625	0.74625	0.74750	0.75500	0.76875	0.77750	0.78000	0.78500	0.78750	0.79000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.02500	0.02500	0.02500	0.02500	0.02500	0.02500	0.02500	0.02750	0.02750	0.02875	0.03000	0.03375
9	0.98000	0.98000	0.97750	0.98250	0.98500	0.99000	0.99750	1.00000	0.99875	1.00000	1.00000	1.00000
10	0.64625	0.64750	0.65125	0.64375	0.67500	0.68625	0.70125	0.73375	0.75000	0.75750	0.77625	0.78500
11	0.34250	0.34250	0.34125	0.35625	0.37375	0.41250	0.45500	0.51250	0.55250	0.57000	0.59625	0.60875
12	0.02500	0.02500	0.02500	0.02500	0.02500	0.02625	0.02250	0.02000	0.02625	0.02875	0.03125	0.02875
13	0.04500	0.04500	0.04500	0.04500	0.04500	0.04500	0.04750	0.05875	0.06625	0.07125	0.07500	0.07125
14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00125	0.00125	0.00125	0.00125	0.00125
15	0.97000	0.97000	0.97125	0.97375	0.98250	0.98875	0.99000	0.99125	0.98875	0.99375	0.99750	0.99750
16	0.74750	0.74625	0.74375	0.74375	0.75875	0.78625	0.83625	0.87375	0.90000	0.90875	0.92125	0.92125
17	0.09875	0.10125	0.10125	0.10125	0.11125	0.12375	0.13625	0.14875	0.16250	0.17500	0.18250	0.18625
18	0.10000	0.10000	0.10000	0.09875	0.10000	0.10500	0.10500	0.10500	0.10750	0.10750	0.10750	0.10750
19	0.86125	0.86000	0.86250	0.88625	0.91375	0.94125	0.96625	0.98250	0.99750	1.00000	1.00000	1.00000
20	0.54750	0.54875	0.55000	0.55000	0.54875	0.56375	0.57875	0.60125	0.61625	0.63375	0.65625	0.65750
21	0.57750	0.57625	0.57875	0.57875	0.58500	0.58875	0.60625	0.62500	0.63750	0.64250	0.65250	0.66000

Table 4.12 Delta method and Q -statistics: Comparison of MDR***

Fault	MDR*** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.00250	0.00250	0.00250	0.00250	0.00250	0.00125	0.00125	0.00250	0.00250	0.00625	0.00375	0.00375
2	0.01375	0.01375	0.01375	0.01500	0.01125	0.01125	0.00625	0.01750	0.00000	0.02250	0.02250	0.02250
3	0.99000	0.99000	0.99000	0.98500	0.97500	0.83000	0.54125	0.16000	0.46125	0.87625	0.92625	0.99500
4	0.03625	0.03500	0.03125	0.04000	0.03750	0.03125	0.03250	0.04625	0.00625	0.00625	0.00000	0.47000
5	0.74500	0.74500	0.74250	0.74250	0.72000	0.64375	0.30250	0.16375	0.00000	0.00125	0.00000	0.00750
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.02500	0.02500	0.02500	0.02125	0.02250	0.01875	0.02500	0.00000	0.02500	0.02500	0.02625	0.02625
9	0.98000	0.97875	0.98250	0.97875	0.96625	0.91500	0.67500	0.71125	0.48375	0.50125	1.00000	1.00000
10	0.64625	0.65000	0.65250	0.64750	0.57750	0.46500	0.25500	0.06125	0.06500	0.22625	0.45375	0.46375
11	0.34250	0.34000	0.33500	0.33500	0.32875	0.29000	0.19750	0.09875	0.32000	0.31250	0.51875	0.63500
12	0.02500	0.02500	0.02500	0.02875	0.03125	0.01750	0.00875	0.00500	0.01375	0.01625	0.01625	0.01625
13	0.04500	0.04500	0.04500	0.04500	0.04750	0.04125	0.04375	0.05375	0.05250	0.05500	0.05500	0.06000
14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00125	0.00125	0.00125	0.00125	0.00125
15	0.97000	0.97000	0.97125	0.96750	0.94750	0.85500	0.75625	0.25125	0.27875	0.95375	0.95625	0.88375
16	0.74750	0.74375	0.74000	0.72750	0.66125	0.57375	0.29000	0.21000	0.17375	0.53625	0.63250	0.54625
17	0.09875	0.09750	0.10125	0.10000	0.10000	0.06375	0.03375	0.05375	0.04250	0.08250	0.08625	0.09375
18	0.10000	0.10250	0.10000	0.10125	0.09625	0.07625	0.03125	0.04750	0.03375	0.06250	0.11375	0.11375
19	0.86125	0.86250	0.85625	0.86875	0.82125	0.66625	0.53875	0.52625	0.62750	0.84500	0.71125	0.74000
20	0.54750	0.54625	0.53750	0.52875	0.50875	0.44750	0.28625	0.16875	0.13500	0.13750	0.37625	0.35375
21	0.57750	0.57750	0.57125	0.57000	0.53500	0.45625	0.30875	0.50500	0.55500	0.61000	0.61375	0.70375

Table 4.13 Threshold method and T² -statistics: Comparison of FA****

Fault	FA**** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250
2	0.90000	0.90000	0.90000	0.90625	0.90000	0.90625	0.90000	0.89375	0.90000	0.90000	0.88750	0.88750
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.06250	0.06250	0.06250	0.04375	0.02500	0.01250	0.01250	0.00625	0.00625	0.00625	0.00625	0.00625
5	0.95000	0.95000	0.95000	0.95000	0.95000	0.94375	0.93125	0.91875	0.91250	0.90625	0.89375	0.89375
6	0.94375	0.94375	0.94375	0.94375	0.94375	0.94375	0.94375	0.94375	0.93750	0.93750	0.93750	0.93750
7	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
8	0.86250	0.86250	0.86250	0.86250	0.86250	0.86250	0.86250	0.86250	0.85000	0.85000	0.85000	0.83750
9	0.00615	0.00625	0.00625	0.00625	0.00625	0.00625	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.41785	0.42500	0.41875	0.37500	0.35625	0.28750	0.24375	0.19375	0.16250	0.15625	0.14375	0.13750
11	0.27500	0.27500	0.27500	0.26875	0.24375	0.24375	0.19375	0.17500	0.16875	0.15625	0.16875	0.16875
12	0.91250	0.91250	0.91250	0.91250	0.91250	0.90625	0.88750	0.88125	0.86250	0.83750	0.78750	0.76875
13	0.70000	0.70000	0.70000	0.70000	0.70000	0.70000	0.70000	0.70000	0.68750	0.66875	0.65625	0.65000
14	0.87500	0.87500	0.88125	0.88125	0.86250	0.84375	0.81875	0.81875	0.80000	0.78125	0.78125	0.78125
15	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
16	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
17	0.68750	0.68750	0.68750	0.68125	0.68125	0.65625	0.62500	0.60625	0.58750	0.57500	0.57500	0.57500
18	0.43750	0.43750	0.43750	0.43750	0.42500	0.42500	0.42500	0.41875	0.41875	0.41250	0.41250	0.41250
19	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
20	0.22500	0.22500	0.22500	0.23125	0.21875	0.21250	0.20000	0.16250	0.15625	0.15000	0.13750	0.13750
21	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 4.14 Delta method and T² -statistics: Comparison of FA****

Fault	FA**** - T2: Original data sets compared to compressed data sets (red color = fault is NOT detected)											
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.96250	0.95000	0.95000
2	0.90000	0.90000	0.90000	0.90625	0.90000	0.88750	0.90000	0.90000	0.88750	0.90000	0.88750	0.88750
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.03125	0.00625	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.06250	0.06250	0.06250	0.05625	0.05625	0.02500	0.01250	0.01250	0.00625	0.00625	0.00625	0.00625
5	0.95000	0.95000	0.95000	0.95000	0.93750	0.93750	0.94375	0.98750	0.98125	1.00000	0.93750	0.91250
6	0.94375	0.94375	0.94375	0.94375	0.94375	0.94375	0.94375	0.96250	0.95625	0.95625	0.95625	0.95625
7	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
8	0.86250	0.86250	0.86875	0.86250	0.86250	0.85625	0.86250	0.85000	0.85000	0.85000	0.83750	0.83750
9	0.00615	0.00625	0.00625	0.02500	0.02500	0.01250	0.00625	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.41785	0.41875	0.40625	0.41250	0.41250	0.35625	0.36250	0.30000	0.30625	0.23750	0.18750	0.16250
11	0.27500	0.27500	0.26875	0.26250	0.25000	0.26250	0.26250	0.26250	0.17500	0.16875	0.13750	0.13125
12	0.91250	0.91250	0.91250	0.91250	0.92500	0.91250	0.88750	0.90625	0.86250	0.85625	0.81250	0.83750
13	0.70000	0.70000	0.70000	0.69375	0.69375	0.69375	0.69375	0.68750	0.69375	0.68750	0.64375	0.63750
14	0.87500	0.87500	0.87500	0.87500	0.87500	0.88750	0.87500	0.83750	0.85625	0.83125	0.82500	0.83125
15	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
16	0.00000	0.00000	0.00000	0.00000	0.00625	0.00625	0.00000	0.00625	0.02500	0.00000	0.00000	0.00000
17	0.68750	0.68750	0.68750	0.68750	0.68750	0.65625	0.68750	0.66250	0.63750	0.65000	0.61250	0.61875
18	0.43750	0.43750	0.43750	0.44375	0.45000	0.43750	0.41875	0.41250	0.43750	0.43750	0.39375	0.39375
19	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
20	0.22500	0.21875	0.21875	0.22500	0.21875	0.23750	0.26250	0.18750	0.17500	0.13125	0.12500	0.12500
21	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 4.15 Threshold method and Q -statistics: Comparison of FA****

Fault	FA**** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)										
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.98750	0.98750	0.98750	0.98750	0.98750	0.98125	0.98125	0.98125	0.97500	0.97500	0.97500	0.96875
2	0.93125	0.93125	0.93125	0.93125	0.92500	0.92500	0.92500	0.91250	0.91250	0.91250	0.91250	0.91250
3	0.03125	0.03125	0.03125	0.04375	0.03125	0.00625	0.00625	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.96875	0.96875	0.96875	0.96250	0.96250	0.93125	0.86250	0.72500	0.55625	0.45625	0.38750	0.33750
5	0.99375	0.99375	0.99375	0.98750	0.98750	0.98750	0.97500	0.96875	0.96250	0.95625	0.95625	0.95625
6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
7	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
8	0.87500	0.87500	0.87500	0.87500	0.87500	0.87500	0.87500	0.86250	0.86250	0.86250	0.86250	0.86250
9	0.03125	0.03125	0.02500	0.01875	0.01250	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.43750	0.43750	0.43125	0.44375	0.40000	0.40000	0.41250	0.36250	0.33125	0.28125	0.24375	0.24375
11	0.71250	0.71250	0.71250	0.71250	0.68750	0.65000	0.60625	0.51250	0.46250	0.45625	0.44375	0.43125
12	0.89375	0.89375	0.89375	0.90625	0.89375	0.88125	0.91250	0.91875	0.90000	0.90625	0.89375	0.89375
13	0.77500	0.77500	0.77500	0.77500	0.77500	0.77500	0.76250	0.71875	0.68750	0.66250	0.65000	0.67500
14	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	0.99375	0.99375	0.99375	0.99375	0.99375
15	0.01875	0.01875	0.01875	0.01875	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
16	0.05000	0.05625	0.05000	0.05000	0.03750	0.01250	0.00625	0.00625	0.00000	0.00000	0.00000	0.00000
17	0.81250	0.81250	0.80625	0.81250	0.81875	0.81250	0.81250	0.81250	0.77500	0.76250	0.76875	0.76250
18	0.50000	0.50000	0.50000	0.50625	0.50000	0.47500	0.47500	0.47500	0.46250	0.46250	0.46250	0.46250
19	0.15000	0.15000	0.15000	0.11875	0.10000	0.06250	0.04375	0.03750	0.00000	0.00000	0.00000	0.00000
20	0.29375	0.29375	0.30000	0.29375	0.29375	0.28750	0.30625	0.26875	0.26250	0.26250	0.23750	0.23125
21	0.01250	0.00625	0.01250	0.01875	0.01875	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

Table 4.16 Delta method and Q -statistics: Comparison of FA****

Fault	FA**** - Q: Original data sets compared to compressed data sets (red color = fault is NOT detected)										
No.	Orig.	5 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %
1	0.98750	0.98750	0.98750	0.98750	0.98750	0.99375	0.99375	0.98750	0.98750	0.96875	0.98125	0.98125
2	0.93125	0.93125	0.93125	0.92500	0.94375	0.94375	0.96875	0.91250	1.00000	0.88750	0.88750	0.88750
3	0.03125	0.03125	0.03125	0.03750	0.05000	0.24375	0.51875	0.52500	0.19375	0.00000	0.00000	0.00000
4	0.96875	0.97500	0.97500	0.98125	0.98750	0.99375	0.98750	0.93750	0.96875	0.71250	1.00000	0.50000
5	0.99375	0.99375	0.99375	0.99375	1.00000	1.00000	1.00000	1.00000	1.00000	0.99375	1.00000	0.96250
6	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
7	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
8	0.87500	0.87500	0.87500	0.89375	0.88750	0.90625	0.87500	1.00000	0.87500	0.87500	0.86875	0.86875
9	0.03125	0.03125	0.01250	0.01250	0.01875	0.06250	0.62500	0.03125	0.00625	0.00000	0.00000	0.00000
10	0.43750	0.43750	0.44375	0.43125	0.50000	0.54375	0.66875	0.79375	0.82500	0.72500	0.54375	0.49375
11	0.71250	0.71875	0.72500	0.73750	0.73750	0.74375	0.73750	0.71250	0.55000	0.60000	0.49375	0.46250
12	0.89375	0.89375	0.89375	0.88750	0.89375	0.93750	0.95625	0.97500	0.93125	0.91875	0.91875	0.91875
13	0.77500	0.77500	0.77500	0.77500	0.76250	0.79375	0.78125	0.73125	0.73750	0.72500	0.72500	0.70000
14	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	0.99375	0.99375	0.99375	0.99375	0.99375
15	0.01875	0.01875	0.01875	0.01250	0.02500	0.07500	0.11875	0.71875	0.43750	0.00000	0.00000	0.00000
16	0.05000	0.05000	0.06875	0.05625	0.12500	0.13750	0.49375	0.25000	0.41875	0.00000	0.00625	0.00000
17	0.81250	0.81250	0.81250	0.81250	0.81875	0.85625	0.90625	0.84375	0.84375	0.84375	0.84375	0.84375
18	0.50000	0.48750	0.50000	0.49375	0.51875	0.61875	0.84375	0.76250	0.83125	0.68750	0.43125	0.43125
19	0.15000	0.14375	0.16250	0.14375	0.16875	0.42500	0.35000	0.12500	0.16875	0.00625	0.00625	0.01250
20	0.29375	0.30000	0.30625	0.30000	0.30625	0.38750	0.45000	0.66875	0.40000	0.42500	0.36875	0.22500
21	0.01250	0.00625	0.01875	0.00625	0.03125	0.00625	0.53750	0.00000	0.00000	0.00000	0.00000	0.00000

5 SUMMARY AND CONCLUSIONS

The main objective of this thesis was to find out the effects and the efficiency of event-driven approach on Tennessee Eastman process. The study was made with two different methods, Threshold method and Delta method and the idea was to prove that these methods don't lower the fault detection rate of the principal component analysis and this comparison was meant to be done by comparing the results to Evan L. Russell's results, which he has made with Richard D. Braatz and Leo Chiang.

The procedure for the event-driven data compression with Threshold method was confirmed to work properly to a certain, fairly high degree of compression ratio without lowering the fault detection rate. Tested method didn't improve the fault detection capabilities though, but it did reduce the amount of data being handled in the process greatly. The results of Threshold method by this point of view may improve the fault detection indirectly in certain cases on the system level.

The second comparison for the event-driven data compression with Delta method though was confirmed to work great. Low compression ratio didn't make that much change, it wasn't enough to make a change on the behaviour of the fault detection and too high compression ratio made the situation worse again. Right amount of compression ratio, 50-70%, had excellent results on fault detection and only couple of faults were missed while checking the results of T^2 -statistics and not a single fault was being left unnoticed by the system while checking the results of Q -statistics. Tested method did improve the fault detection capabilities and it did reduce the amount of data being handled in the process fairly good. The results of Delta method do improve the fault detection directly in certain cases on the system level and it can be seen as more effective method from studied methods. Improvement of fault detection is undeniable.

Achievements from this thesis were codes for resampling fault data sets with Threshold and Delta methods which were made for Tennessee Eastman process and the resampling was made possible with a code which handles various compression rates. Resampled codes, resampled data sets, original and resampled fault detection figures and codes for testing the methods were uploaded for available in GitHub.

Also the results of the made tests with the principal component analysis were uploaded available there, including the control charts of T^2 and Q -statistics for every fault data set with every compression rate which was tested, both Threshold and Delta methods.

Further possible studies would be testing the event-driven approach for Tennessee Eastman process with also other fault detection methods like partial least squares (PLS), Fisher discriminant analysis (FDA) and canonical variate analysis (CVA) to obtain an understanding that will the results be on the same level that they were with the principal component analysis (PCA) which was tested here.

REFERENCES

1. H.B. Aradhye, B.R. Bakshi, and R. Strauss. Process monitoring by PCA, dynamic PCA, and multiscale PCA - Theoretical analysis and disturbance detection in the Tennessee Eastman process. In AIChE Annual Meeting, 1999.

2. R.D. Braatz: The Fortran 77 codes, 2017. https://github.com/camaramm/tennessee-eastman-profBraatz

3. L.H. Chiang, E.L. Russell, and R.D. Braatz. Fault Detection and Diagnosis in Industrial Systems, Springer-Verlag, London, 2001.

4. L.H. Chiang, E.L. Russell, and R.D. Braatz. "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis," Chemometrics and Intelligent Laboratory Systems, 2000.

5. N. Doganaksoy, F.W. Faltin, and W.T. Tucker. Identification of out of control quality characteristics in a multivariate manufacturing environment. *Commun. Stat.- Theory Methods*, 20:2775, 1991.

6. J.J. Downs and E.F. Vogel, "A plant-wide industrial process control problem," Computers and Chemical Engineering, 1993

7. J.J. Downs and E.F. Vogel, "A plant-wide industrial process control problem." Presented at the AIChE 1990 Annual Meeting, Session on Industrial Challenge Problems in Process Control, Paper #24a Chicago, Illinois, November 14, 1990.

8. R. J. Doyle, L. Charest, N. Rouquette, J. Wyatt, and C. Robertson. Causal modeling and event-driven simulation for monitoring of continuous systems. *Computers in Aerospace*, 1993.

9. J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 1979.

10. J.V. Kresta, T.E. Marlin, and J.F. MacGregor. Multivariable statistical monitoring of process operating performance. Can. J. of Chem. Eng., 69:35-47, 1991.

11. P.R. Lyman. Plant-wide Control Structures for the Tennessee Eastman process, M.S. thesis, Lehigh University, 1992.

12. M.J. Piovoso, K.A. Kosanovich, and R.K. Pearson. Monitoring process performance in real time. In Proc. of the American Control Con/. , pages 2359-2363, Piscataway, New Jersey, 1992. IEEE Press

13. E.L. Russell, L.H. Chiang, and R.D. Braatz. Data-driven Techniques for Fault Detection and Diagnosis in Chemical Processes, Springer-Verlag, London, 2000.

14. E.L. Russell, L.H. Chiang, and R.D. Braatz. "Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis," Chemometrics and Intelligent Laboratory Systems, 51:81-93, 2000.

15. D. Serpanos and M. Wolf, Internet-of-Things (IoT) Systems, *Architectures, Algorithms, Methodologies,* Cham, Switzerland, 2018

16. M. Miskowicz, *Event-based control and signal processing*, CRC Press, 2015.

17. M. Miskowicz; R. Golanski. *LON Technology in Wireless Sensor Networking Applications*. Sensors, 2006.

APPENDIX 1. TEP event-driven: Demonstration

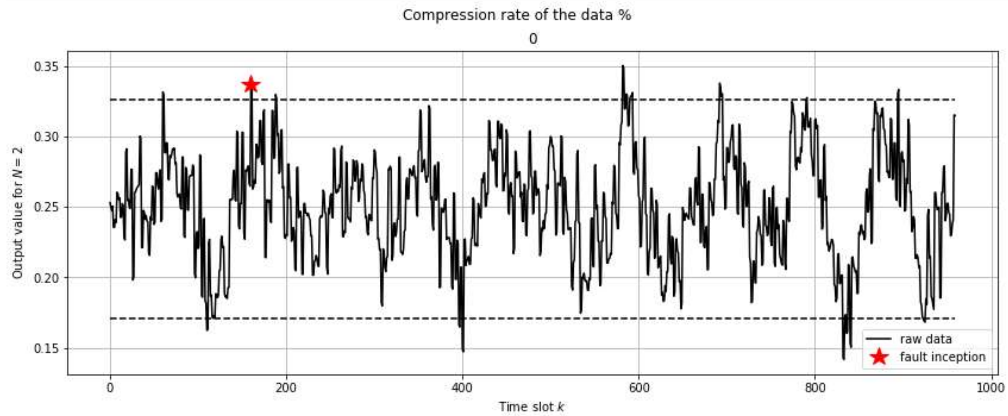
Example of the effects of TEP in a single variable with a 90% compression rate.

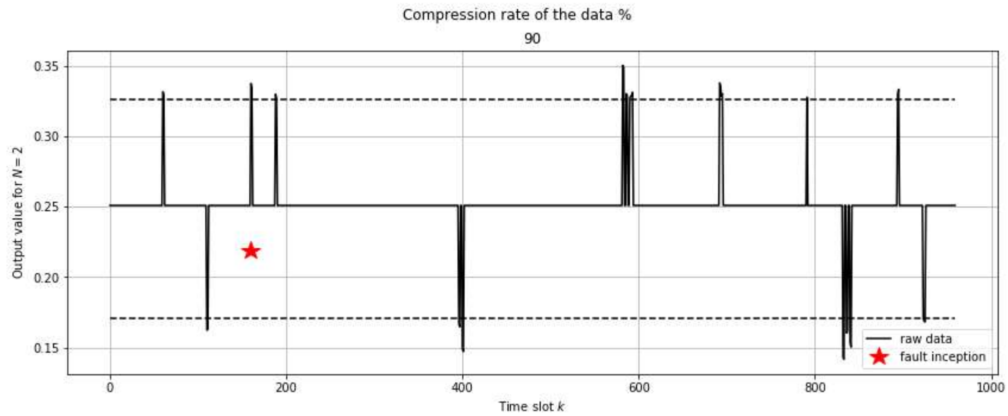
Comparison demonstration between original data and compressed data

```
# Picking up a variable from 0 < v < 53 and defining a compression rate of the data
variable = 1
fault = 1

plt.figure(figsize=(14,5))
percentage = 0
F1,index_0,plot_x0,plot_yu0,plot_yd0 = tennessee_resampling('d*_te.dat',variable,fault,percentage)
plt.plot(F1[variable],'k',label='raw data')
plt.plot(plot_x0,plot_yu0,'k--')
plt.plot(plot_x0,plot_yd0,'k--')
plt.plot(160,F1.iloc[160,variable-1], 'k*',markersize=16,label='fault inception', color='r')
plt.suptitle('Compression rate of the data %')
plt.title(percentage)
plt.legend(loc='lower right')
plt.xlabel('Time slot $k$')
plt.ylabel('Output value for $N = 2$')
plt.grid()
plt.show()

plt.figure(figsize=(14,5))
percentage = 90
F1,index_0,plot_x0,plot_yu0,plot_yd0 = tennessee_resampling('d*_te.dat',variable,fault,percentage)
plt.plot(F1[variable],'k',label='raw data')
plt.plot(plot_x0,plot_yu0,'k--')
plt.plot(plot_x0,plot_yd0,'k--')
plt.plot(160,F2.iloc[160,variable-1], 'k*',markersize=16,label='fault inception', color='r')
plt.suptitle('Compression rate of the data %')
plt.title(percentage)
plt.legend(loc='lower right')
plt.xlabel('Time slot $k$')
plt.ylabel('Output value for $N = 2$')
plt.grid()
plt.show()
```





APPENDIX 2. Code for resampling the data sets

The code which was used to resample the data sets with Threshold method

TEP - Resampling datasets - Threshold method

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import math
from glob import glob
import datetime
```

```
def print_new_samples(fault):
    percentages = [5,10,20,30,40,50,60,70,80,90,95]
    paths = glob('d*_te.dat')
    dfs = {}
    columns_name = ['Variable_' + str(x) for x in range(1,53)]
    for mov in range(0,len(paths)):
        data = np.genfromtxt(paths[mov])
        dfs[mov] = pd.DataFrame(data=data,columns=columns_name)

    for p in percentages:
        for v in range(1,53):

                max_v1 = dfs[0].iloc[:,v-1].max()
                mean_v1 = dfs[0].iloc[:,v-1].mean()
                min_v1 = dfs[0].iloc[:,v-1].min()

                        # Percentage UP AND DOWN THE MEAN

                mean_up = mean_v1+((p/100)*abs(max_v1-mean_v1))
                mean_down = mean_v1-((p/100)*abs(min_v1-mean_v1))



                index1 = dfs[fault][dfs[fault]['Variable_' + str(v)].between(mean_down,mean_up)].index


                dfs[fault].loc[index1,'Variable_' + str(v)] = mean_v1
        dfs[fault].to_csv('NEW_'+str(p)+'_d0'+str(fault)+'_te.dat',sep=" ",index=False, header=False)
```

```
for x in range(1,22):

    print_new_samples(x)
```

APPENDIX 3. Code for resampling the data sets

The code which was used to resample the data sets with Delta method.

TEP - Resampling datasets - Delta method

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import math
from glob import glob
import datetime
```

```
def print_new_samples_delta(fault):
    #percentages = [5,10,20,30,40,50,60,70,80,90,95]
    percentages = [5,10,20,40,60,80,90,95]
    paths = glob('d*_te.dat')
    dfs = {}
    columns_name = ['Variable_' + str(x) for x in range(1,53)]
    for mov in range(0,len(paths)):
        data = np.genfromtxt(paths[mov])
        dfs[mov] = pd.DataFrame(data=data,columns=columns_name)

    for p in percentages:

        delta = np.zeros((960,52))                          # Initiate delta sample per sample, 960 total samples
        m_delta = np.zeros((1,52))                          # Initiate max delta for normal operation, 52 total variables
        compress = np.zeros((1,52))                        # Initiate compression rate

        for v in range(0,52):                               # "For" all 52 variables of normal operation
            for d in range(0,dfs[0].shape[0]-1):            # "For" all samples

                delta[d,v] = abs(dfs[0].iloc[d,v]-dfs[0].iloc[d+1,v])  # Calculate all deltas sample per sample
            m_delta[0,v] = max(delta[:,v])                #Calculate the biggest delta in normal operation and store

        for v in range(0,52):                               # "For" all 52 variables of each fault
            count = 0
            for d in range(0,dfs[0].shape[0]-1):            # "For" all samples

                if abs(dfs[fault].iloc[d,v]-dfs[fault].iloc[d+1,v]) < ((p/100)*m_delta[0,v]): #Calculate per sample fault
                    dfs[fault].iloc[d+1,v] = dfs[fault].iloc[d,v]
                    count += 1
            compress[0,v] =  count/960                       # Compression rate for 'v' variable

        dfs[fault].to_csv('DEL_'+str(p)+'_d0'+str(fault)+'_te.dat',sep=" ",index=False, header=False)
    #return(compress)        #Uncomment this if you want to vizualize compression rate
```

APPENDIX 4. PCA: Failure testing

The code which was used to test failure recognition.

Failure Testing

```
for f in range(0,21):

    path2 = files[f]
    fault_file = np.loadtxt(path2)

    if np.shape(fault_file)[0] < np.shape(fault_file)[1]:
        fault_file = np.transpose(fault_file)

    #Normalizing Data
    fault_file_norm = fault_file.copy()

    for i in pd.DataFrame(fault_file_norm):
        pd.DataFrame(fault_file_norm)[i] = (pd.DataFrame(fault_file_norm)[i] - averages[i])/desv_pad[i]

    #Building the Main Components
    fault_pca = np.dot(fault_file_norm , auto_vet_matx)

    ##Calculations of T2
    #Calculation of T2
    fault_T2_matx = pd.DataFrame(fault_pca).apply(el2)

    for i in fault_T2_matx :
        fault_T2_matx[i] = fault_T2_matx[i]/auto_val[i]

    #Calculations of T2 on a Failure
    fault_T2 = np.array(fault_T2_matx.sum(axis=1))
    fault_Hotlling = pd.DataFrame(fault_T2, columns=['Fault T2'])
    fault_Hotlling['T2 - UCL'] = T2_UCL
    data_base_T2 += [fault_T2]

    ##Detection Speed Analysis
    #Detection Criteria => 6 consecutive points above the control limit

    for i in range(0,np.shape(fault_T2)[0]):
        if i > (np.shape(fault_T2)[0] - 6):
            break

        if ((fault_T2[i] >  T2_UCL)
            and (fault_T2[i+1] >  T2_UCL) and (fault_T2[i+2] >  T2_UCL) and (fault_T2[i+3] >  T2_UCL)
            and (fault_T2[i+4] >  T2_UCL) and (fault_T2[i+5] >  T2_UCL)):
            detect_T2 += [[i+1 , failures[f]]]
            break

    ##Calculations of Q
    #Calculation of the estimated X* Matrix of the failure
    fault_X_est = np.dot(fault_pca, np.transpose(auto_vet_matx))

    #Failure R Residues
    fault_R = fault_file_norm - fault_X_est

    #Calculations of Q on a Failure
    Q = np.diag(np.dot(fault_R,np.transpose(fault_R)))
    fault_Q_Chart = pd.DataFrame(Q, columns=['Fault Q'])
    fault_Q_Chart['Q - UCL'] = Q_UCL
    data_base_Q += [Q]

    ##Detection Speed Analysis
    #Detection Criteria => 6 consecutive points above the control limit

    for i in range(0,np.shape(Q)[0]):
        if i > (np.shape(Q)[0] - 6):
            break

        if ((Q[i] > Q_UCL)
            and (Q[i+1] > Q_UCL) and (Q[i+2] > Q_UCL) and (Q[i+3] > Q_UCL)
            and (Q[i+4] > Q_UCL) and (Q[i+5] > Q_UCL)):
            detect_Q += [[i+1 , failures[f]]]
            break
```