

University of Alberta

**DEVELOPING MATLAB TOOLS FOR DATA BASED ALARM MANAGEMENT
AND CAUSALITY ANALYSIS**

by

Md Shahedul Amin

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

in

Control Systems

Department of Electrical and Computer Engineering

©Md Shahedul Amin

Fall 2012

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

To my parents and my wife Farzana

Abstract

In this thesis, two Graphical User Interfaces (GUIs) are designed in MATLAB to perform causality analysis and alarm management. Finding out the root-cause of a fault scenario or an abnormality in a large industrial process typically requires one to logically analyze cause and effect relationships between variables. Causality analysis can play a vital role to capture process connectivity and topology and to identify relationships among the variables in a process. The availability of large volumes of industrial process data has now opened the way to develop data-driven methods for causality detection. In the first tool, different techniques of data visualization along with three data-driven methods of causality analysis, namely, cross-correlation, transfer entropy, and Granger causality, have been implemented. Case studies are provided to illustrate the capture of process connectivity using both transfer entropy and Granger causality methods.

Recent studies have shown that the number of alarms in process industries is far more than the approved standards because of a very high number of false and nuisance alarms. The large number of alarms distracts the operator from safe and regulatory monitoring of the processes, which leads to plantwide upset and affects overall productivity of the system. Therefore root cause identification of faults and alarm management have become very important for process industries. The second tool for alarm management is proposed where historic alarm data can be used to find out the top bad actors in the system. Also functions for correlated alarms and similarity between different alarm flood analysis have been implemented in the GUI for easier root cause identification.

Acknowledgements

I would like to use this opportunity to extend my sincere thanks to all the individuals without whom this thesis would not have been possible.

- First of all, I would like to express my heartiest gratitude to my supervisor Dr. Tongwen Chen (Professor, Department of Electrical and Computer Engineering, University of Alberta), for his guidance during my study and research at the University of Alberta. I am highly obliged to him for the time he spent in leading, supporting, and encouraging me, which helped me to shape my thoughts in the right direction not only on my research, but also for my future endeavors. Dr. Chen's continuous support made the momentum of the research going. I would like to thank him for always finding time for me in his busy schedule.
- I am highly grateful to my co-supervisor Dr. Sirish L. Shah (Professor, Department of Chemical and Materials Engineering, University of Alberta), for his valuable comments and guidance for the advancement of the research.
- I am deeply thankful to Dr. Fan Yang (Lecturer, Department of Automation, Tsinghua University, Beijing, China), for his unbounded support. His constant and precious feedback, comments, motivation, and guidelines played a significant role for this research.
- My earnest thanks to Dr. Iman Izadi (Solution Specialist, Alarm Management, Honeywell Process Solutions, Edmonton, Canada), for his valuable suggestions, support, and motivation on the research and help in conducting the industrial case study.
- I acknowledge and thank the National Sciences and Engineering Research Council of Canada, Strategic Project Grant program, for their financial support in this work. I also wish to thank Suncor Energy Inc. (Extraction Plant, Fort McMurray and Pipelines, Sherwood Park), Imperial Oil, and Matrikon Inc. (now part of Honeywell Advanced Solutions) for their collaboration in this research.
- My earnest thanks to my thesis examination committee members, for their careful reading of this thesis and perceptive suggestions.
- Last but not the least, my heartfelt gratefulness to my family and my colleagues. My special thanks to Sandeep R. Kondaveeti, Kabir Ahmed, Ping Duan, and Salman Ahmed for their help in preparing this thesis.

My sincere apology for omissions of any contribution, which may have occurred unintentionally.

Md Shahedul Amin

Contents

1	Introduction	1
1.1	Process Industry and Causality Analysis	1
1.2	Alarm Management	2
1.3	Graphical User Interface (GUI)	3
1.4	Literature Review	4
1.5	Scope and Organization	5
1.5.1	Scope	5
1.5.2	Organization	6
2	Causality Analysis based on Process Data	7
2.1	Transfer Entropy	7
2.2	Granger Causality	8
2.3	Summary	10
3	Graphical User Interface (GUI) for Causality Analysis	12
3.1	Data Loading	14
3.2	Data Preprocessing	15
3.2.1	Remove Outliers	16
3.2.2	Detrend and Demean Data	17
3.2.3	Filter Data	17
3.2.4	Normalize Data	17
3.3	Time Trend	17
3.4	Frequency Response	19
3.5	Power Spectral Density	20

3.6	Spectrogram	21
3.7	High Density Plot	21
3.7.1	High Density Time Trend	22
3.7.2	High Density Frequency Trend	23
3.7.3	High Density Spectral Plot	23
3.8	Correlation Color Map	24
3.8.1	Considering No Lag	24
3.8.2	Considering Lag	25
3.9	Parallel Coordinate Plot	27
3.9.1	Linear Parallel Coordinate Plot	28
3.9.2	Polar Parallel Coordinate Plot	30
3.10	Transfer Entropy	30
3.11	Granger Causality	32
3.12	Summary	33
4	Alarm Management	36
4.1	High Density Alarm Plot	37
4.2	Alarm Similarity Color Map	39
4.3	Run Length and Run Length Distribution	40
4.4	Chattering Index	42
4.5	Alarm Burst Plot	43
4.6	Alarm Flood Similarity Analysis	45
4.7	Summary	47
5	Case Studies	48
5.1	Experimental Case Study	48
5.1.1	Exp # 1	49
5.1.2	Exp # 2	52
5.1.3	Exp # 3	54
5.1.4	Exp # 4	55
5.2	Industrial Case Study	56

5.2.1 Alarm Management	57
5.2.2 Causality Analysis	63
5.3 Summary	63
6 Conclusions	66
6.1 Contributions	66
6.2 Future Scope	66
Bibliography	68

List of Tables

4.1	Run Length for tag <i>Tag_1</i>	41
4.2	Similar flood sequences of real industrial data	46
5.1	Summary of alarm system of a pipeline industry	57
5.2	An example of similar alarm flood sequences in the industrial case study . .	62

List of Figures

3.1	Outline of the MDAtool GUI	13
3.2	Experimental setup of the 4 - tank system	13
3.3	Dialogue box for selecting files to load process data	14
3.4	Dialog box appearing after loading the data when the file contains some empty spaces which might produce wrong results and the actions performed by the corresponding buttons	15
3.5	Data preprocessing interface of the MDAtool which comes with the options for outlier removal, detrending, demeaning, filtering, and/or normalizing the data	16
3.6	Dialogue box that appears upon clicking on “ <i>Time Trend</i> ” where the user can choose to plot any number of interested tags or all of them present in the data	18
3.7	Time trend of 3 selected tags from Figure 3.6	18
3.8	Frequency response of 3 selected tags	19
3.9	Power Spectral Density (PSD) of a selected tag, <i>Right Pump Flow</i>	20
3.10	Spectrogram of a selected tag, <i>Right Pump Flow</i>	21
3.11	High density time trends	22
3.12	High density frequency response	23
3.13	High density spectral plot	24
3.14	Correlation colormap among the variables considering no lag	25
3.15	Lag adjusted correlation colormap between the variables	26
3.16	Options provided for parallel coordinate plot (PCP)	27
3.17	Simple linear PCP: (a) using real values, (b) using normalized values	28

3.18	Dialogue box where the user can choose a variable and its normal operating region	28
3.19	PCP with diferent colors showing the normal data, abnormal low data, and abnormal high data based on the <i>Right Pump Flow</i>	29
3.20	Dialogue box providing option to plot two tags side-by-side in PCP and see the impact of one variable on the other by selecting a range of data from the scatter plot	30
3.21	Polar PCP where darker color represent the initial values and lighter color represent the final values	31
3.22	Results of transfer entropy method in the form of: (a) colormap, (b) connectivity diagram	32
3.23	Dialogue box asking the user to give the parameter values to calculate Granger causality	33
3.24	Results of Granger causality method for the 4-tank system setup in: (a) time domain analysis, (b)frequency domain analysis	34
4.1	Outline of the <i>alarm_tool</i>	37
4.2	HDAP of 20 tags with number of alarms ranging from 537 to 9196 for individual tags	38
4.3	Alarm similarity color map shown in the tool	39
4.4	Run Length Distribution of <i>Tag-1</i> : a) On to On, b) On to Off, c) Off to On	42
4.5	Run Length Distribution in the tool with options to select any tag and from different kinds of RLDs	43
4.6	Run Length Distributions for: a) Non-chattering tag, b) Chattering tag . .	44
4.7	Sorted chattering index of the tags	45
4.8	Alarm burst plot of real industrial data showing the instants of alarm flood after removal of chattering alarms	46
4.9	Similar flood analysis of real industrial alarm data	47
5.1	High Density Time Trend Plot of different signals obtained in Exp # 1 . . .	49
5.2	Water levels in meters of the three related tanks in Exp # 1	50

5.3	Correlation colormap (considering lags) of the related variables of Exp # 1	51
5.4	Causality results obtained from transfer entropy for Exp # 1: (a) Colormap of transfer entropies among the variables, (b) Corresponding connectivity diagram	51
5.5	Causality results obtained from time domain Granger causality for Exp # 1: (a) Granger causality colormap, (b) Corresponding connectivity diagram . .	51
5.6	Results of frequency domain Granger causality for Exp # 1	52
5.7	(a): High Density Time Trend Plot of different tags obtained in Exp # 2, (b): Water levels in meter of the three related tanks of Exp # 2	53
5.8	Connectivity diagram obtained for Exp # 2 using: (a) transfer entropy, (b) Granger causality	54
5.9	Connectivity diagram obtained for Exp # 3 using: (a) transfer entropy, (b) Granger causality	55
5.10	(a): High Density Time Trend Plot of different tags obtained in Exp # 4, (b): Water levels in meters of the three related tanks of Exp # 4	56
5.11	Connectivity diagrams obtained for Exp # 4 using: (a) transfer entropy, (b) Granger causality	56
5.12	High density alarm plot of one month industrial alarm data	58
5.13	Run length distributions of top 4 bad actors in the system	59
5.14	Chattering indices of the bad actors of the system	59
5.15	Correlation color map of the correlated alarm tags	60
5.16	Alarm burst plot of one week	61
5.17	Similarity color map of similar floods in one month alarm data	63
5.18	Causality analysis results of one month process data	64

Chapter 1

Introduction

1.1 Process Industry and Causality Analysis

The economical development of an industrial plant depends on efficient operation of the whole system. Recent research has led towards regulating more and more process variables. As a result, a typical industrial process consists of thousands of sensors, actuators, and control loops with wired or wireless communication networks. The process is itself is highly interconnected via material flow paths. The complexity of the processes is increasing exponentially along with the advancement of control systems as well as with the ever increasing number of components of the system. A large industrial process consists of a number of small units, interconnected together. As a result, faults may start from a small unit, propagate through out the system, and eventually cause catastrophic consequences [1]-[2]. In order to have increased productivity plus cost benefit and safety of the personnel and plant, it is very important that fault detection schemes are prompt and efficient to identify faults and that appropriate corrective action is taken as soon as possible when necessary. This is very important to prevent plantwise upset and loss and especially to improve safety and security of the personnel involved in the system. This is an important motivation for capturing process connectivity and process topology to relate to the dynamics between variables of the system which eventually allows one to find out the root cause of faults in the system.

The historical experience will help the operator to identify the exact location where special attention is required and be ready should such a scenario be encountered again. Most often, direct access or knowledge is not easily available on the dynamics of the physical process underlying the system. It is even more complicated to capture the process connectivity

from the Piping and Instrumentation Diagrams (P&IDs) for a large and complex process, having a large number of process variables. Moreover, the captured topology from P&IDs needs to be validated from the process data [1]. Causality analysis shows the information transfer pathways in the system. Therefore, process connectivity is not equal to causality [1]. The abundance of the industrial process data paves the way for data-driven methods for causality analysis as there is plenty of historical data from industrial facilities [3]. Therefore, causality analysis by data driven methods can play a very important role to identify the root cause of a fault. There are some existing data-driven methods for causality analysis with some advantages and disadvantages [3]. In this thesis a tool is designed in MATLAB for data visualization of the process and to capture the process connectivity to identify the root cause of faults in the system as well as data propagation paths. The tool provides options for 3 kinds of data driven methods: 1) correlation analysis, 2) transfer entropy, and 3) Granger causality; which can be readily applied to the available loaded process data.

1.2 Alarm Management

Research over the last few decades has created a new generation of fault detection methods and they are well applied in industry [4], [5], [6]. Recent introduction of *Distributed Control Systems* (DCS) in the process industry has led the way to monitor thousands of process tags and set alarms for many of the critical tags. A fault is detected in the system when a process variable goes beyond its normal operating region. As soon as a fault is detected in the system an alarm is sent to the operator for notification about the abnormal situation. Different priorities are set for different variables depending on the impact of the variable on the overall performance of the system. In the ideal case there should be only one alarm for each fault in the system [7]. However, the practical scenario is completely opposite. Operators receive far more alarms than they can effectively manage. Out of the total alarms that an operator receives, many of them are either *False* or *Nuisance* alarms. For every single alarm raised, the operator has to acknowledge, find out the significance of the alarm, and take appropriate action. Too many alarms distracts the operator from the main focus point, which may lead to a poor operating condition of the system [8]. The annunciation of a large sequence of alarms due to a single event is known as an *Alarm Flood*. Generally speaking,

an alarm flood is the condition when an operator receives more alarms than he/she can effectively manage. The quantitative number varies from industry to industry as can be found in [9]. Therefore, to reduce the total number of alarms as well as to ensure that the operator is not getting more than necessary alarm per fault, alarm management is very important. Several steps are required to design and obtain an efficient alarm management scheme. High density alarm plots can play an important role in visualizing the number of alarms in the system from historical alarm data [10]. Identification of correlated alarms and proper analysis can help to identify the root cause of faults in the system [10]. The chattering index plays an important role and proper rationalization of chattering tags can reduce the number of alarms in the system significantly [11]. Detection of alarm floods and identifying similar floods plays a vital role to find out the top bad actors and disturbance propagation paths [12]. In this thesis a tool is proposed where historical alarm data can be loaded and the necessary analysis for alarm management can be performed.

1.3 Graphical User Interface (GUI)

A Graphical User Interface (GUI) uses graphics object and interface components to create a tool that can be used interactively. GUIs can help reduce the necessary effort to do a repetitive task. From an industrial view point GUIs deliver a great deal of comfort and ease of use of the tools without compromising the analysis. GUIs support data loading and analysis by creating a suitable program for the necessary analytical task. GUIs can be developed using many types of softwares, e.g., MATLAB, Visual Basic, C, C++ and so on. In this thesis two GUIs are proposed: one for multivariate process data visualization and causality analysis named *MDAtool* and another one for industrial alarm management from alarm data named *alarm_tool*. Both tools are designed using MATLAB. There are certain advantages of using MATLAB for designing GUIs [13]:

- High level script based development,
- Independent of operating system,
- User interactivity and real time measurements,
- Tremendous computational capability with the power of MATLAB.

MATLAB GUIs provide the following architectural components, function, and tools:

- Handle graphics objects,
- Figure, axes, and uicontrol,
- Mouse/keyboard input,
- Use of global variables in all functions,
- Standalone executable GUI outside MATLAB.

In this thesis all the above mentioned features are used extensively.

1.4 Literature Review

Causality analysis and alarm management for process industries are relatively new area of research. The idea of causality analysis was first introduced by Granger in 1969 in the field of Economics [14]. According to Granger, a variable ‘ X ’ Granger causes the other variable ‘ Y ’ if incorporating the past values of ‘ X ’ helps better predict the future of ‘ Y ’ than incorporating the past values of ‘ Y ’ alone [14]. Causality analysis using Granger causality has been used extensively in the field of Economics and Neuroscience. Since Granger causality is based on cross-spectral methods, it assumes that data is linear which is not the case in reality [3]. Further research on Granger causality has led to the method which can handle non linear data and it can identify causal structure in the neurons [15]. A tool has been developed in the Neuroscience area using Granger causality to capture the connectivity in the neurons [16]. Direct and indirect connectivity of neurons using spectral based methods have also been investigated [17].

Research on root cause identification of faults for process industries started a long time ago. Initially signed directed graph based methods were used to identify the direction of fault propagation [18], [1]. Spectral based methods have not been used to capture the connectivity of process industries. Research on model free method to capture the connectivity other than spectral based methods has paved the way for a new method known as transfer entropy. Transfer entropy is a model free method [19] and it has been proven to be very effective in capturing connectivity from process data [20], [21]. This method is also used for capturing

the connectivity of neurons [22]. Calculation of transfer entropy is based on probability density functions and hence the computational burden increases along with the size of data [1]. So far, to the best of the author's knowledge no tool has been developed for process industries to capture the process connectivity and identify the root cause of faults.

The Abnormal Situation Management (ASM) Consortium [23] reports that, once every three years petrochemical plants on average suffer from a major accident. The ISA 18.2 standard says that, there should be no more than 10 alarms over a 10 minutes time slot [24]. But according to [9], currently process industries have a much higher number of alarms than they can effectively manage. Extensive research is in progress to efficiently manage the alarm system of process industry. [7] and [8] discuss different performance measures of the alarm system, e.g., false alarm rate, missed alarm rate, detection delay and deadband. A very effective and efficient method on industrial alarm data representation is used in [10] using the high density alarm plot. Jaccard similarity coefficient is also used in [10] to identify the correlated alarms. A chattering index to identify the nuisance alarms in the system based on run length distribution is proposed in [11]. Plant wide upsets and alarm floods are discussed in [12]. Alarm burst plots are used to show the instants of alarm floods in the system. Also two methods are proposed in [12] to show similar alarm flood sequences. But so far no tool has been developed for the analysis of historical alarm data to prevent future upset in the plant. Efficient design of alarm management depends on proper identification of bad actors in the system. This thesis proposes a new tool designed to use historical alarm data to detect the top bad actors in the system and show correlated alarms in the system as well as the similar flood sequences.

1.5 Scope and Organization

1.5.1 Scope

Causality analysis and alarm management for process industries are very new areas of research. So far no tool has developed either for causality analysis to show process topology based on historical process data or for alarm management to identify and reduce the number of nuisance and false alarms.

This thesis proposes a new tool for industries to visualize multivariate process data and

perform causality analysis. Several methods for causality analysis are incorporated together to get a better understanding on the underlying process topology and process connectivity based on historic process data. The core goal is to identify root cause of faults in the system.

This thesis also combines methodology to identify correlated alarms on the system, detect the top bad actors, and show similar flood events in the system. Combining these two tools will help the researcher and/or practitioner to sort the most important tags of the system that require utmost attention. Combining process knowledge and results of alarm data analysis can also be useful in identifying the root cause of faults in the system. Once the important tags are identified, later when alarm flood occurs on the system the operators can focus on those tags so that plantwise upsets do not occur and productivity is not hampered.

1.5.2 Organization

This thesis first describes the data based methods of causality analysis. Different aspects of such different methods are discussed in Chapter 2. The tool for causality analysis is presented in Chapter 3. The tool for alarm management or alarm analysis is described in Chapter 4. Chapter 5 shows industrial case studies to justify the importance of the tools followed by concluding remarks and future scopes in Chapter 6.

Chapter 2

Causality Analysis based on Process Data

Causality analysis from historical process data is discussed in this chapter. Correlation analysis can find out the correlated tags in the system but it suffers from the problem that data has to be linear for successful analysis [1]. Transfer entropy, a model free method can handle non linear data and can be very effective for causality analysis [19]. Cross spectral based methods such as Granger causality have been very effective for causality analysis in the field of Neuroscience [15]. For causality analysis all these methods are implemented in the tool with numerous functionality. The cross correlation method is described in Chapter 3. In this chapter the theoretical aspects of transfer entropy and Granger causality method are discussed in more detail.

2.1 Transfer Entropy

For two process variables with sampling interval of τ , $\mathbf{x}_i = [x_i, x_{i-\tau}, \dots, x_{i-(k-1)\tau}]$ and $\mathbf{y}_i = [y_i, y_{i-\tau}, \dots, y_{i-(l-1)\tau}]$, transfer entropy from y to x is defined as follows [25]:

$$t(x|y) = \sum_{x_{i+h}, \mathbf{x}_i, \mathbf{y}_i} p(x_{i+h}, \mathbf{x}_i, \mathbf{y}_i) \cdot \log \frac{p(x_{i+h} | \mathbf{x}_i, \mathbf{y}_i)}{p(x_{i+h} | \mathbf{x}_i)}, \quad (2.1)$$

where p denotes the probability density function (PDF) and h is the prediction horizon. Non parametric methods, e.g., kernel method can be used to estimate the PDF [26]. The Gaussian kernel function is used to estimate the PDF [1] which is defined as follows:

$$K(v) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}v^2}. \quad (2.2)$$

Therefore, a univariate PDF can be estimated by,

$$\hat{p}(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right), \quad (2.3)$$

where N is the number of samples, h is the bandwidth chosen to minimize the error of estimated PDF. h is calculated by $h = c \cdot \sigma \cdot N^{-\frac{1}{5}}$ where $c = (4/3)^{1/5} \approx 1.06$ according to “normal reference rule-of-thumb” approach [27]. For a q -dimensional multivariate case the estimated PDF is given by [1]:

$$\hat{p}(x_1, x_2, \dots, x_q) = \frac{1}{Nh_1 \dots h_q} \sum_{i=1}^N K\left(\frac{x_1 - x_{i1}}{h_1}\right) \dots K\left(\frac{x_q - x_{iq}}{h_q}\right), \quad (2.4)$$

where $h_s = c \cdot \sigma(x_{is})_{i=1}^N \cdot N^{-1/(4+q)}$ for $s = 1, \dots, q$.

Transfer entropy represents the measure of information transfer from y to x by measuring the reduction of uncertainty while assuming predictability [19]. It is defined as the difference between the information about a future observation of x obtained from the simultaneous observation of past values of both x and y , and the information about the future of x obtained from the past of x alone. In [1] it is shown that the parameter values can be chosen as: $\tau = h \leq 4$, $k = 0$, and $l = 1$. In the tool $\tau = h = 2$ is used.

Using the above definitions, direction and amount of information transfer from x to y is as follows:

$$t(x \rightarrow y) = t(y|x) - t(x|y). \quad (2.5)$$

If $t(x \rightarrow y)$ is negative then information is transferred from y to x .

The advantage of using transfer entropy is that it is a model free method and can be applied to non linear data. It has already been proved to be very effective in capturing process topology and process connectivity. But it suffers from a large computational burden due to the calculation of the PDFs. In this thesis while implementing transfer entropy for process data, a pairwise analysis is performed. The results of transfer entropy method are shown in Chapter 3 while discussing the tool as well as in the case studies.

2.2 Granger Causality

The concept of Granger causality was first introduced by C.W.J. Granger in 1969 [14]. According to Granger, x_2 causes x_1 if the inclusion of past observations of x_2 reduces the

prediction error of x_1 in a linear regression model of x_1 and x_2 , as compared to a model which includes only previous observations of x_1 . Two covariance stationary time series signals $x_1(t)$ and $x_2(t)$ of length T can be described by an autoregressive (AR) model as follows:

$$x_1(t) = \sum_{j=1}^p A_{11,j}x_1(t-j) + \sum_{j=1}^p A_{12,j}x_2(t-j) + \xi_1(t) \quad (2.6)$$

$$x_2(t) = \sum_{j=1}^p A_{21,j}x_1(t-j) + \sum_{j=1}^p A_{22,j}x_2(t-j) + \xi_2(t), \quad (2.7)$$

where p is the amount of lag considered and $p < T$, A 's are the coefficients, and ξ 's represent the prediction error. If the inclusion of x_2 reduces ξ_1 then x_2 Granger causes x_1 and vice versa. The magnitude of interaction is measured as follows:

$$F_{2 \rightarrow 1} = \ln \frac{\text{var}\xi_{1R(12)}}{\text{var}(\xi_{1U})}, \quad (2.8)$$

where $\xi_{1R(12)}$ is obtained by omitting $A_{12,j}$ (for all j) and ξ_{1U} is obtained from the complete model. It is assumed that data can be represented by an AR model. The A parameters are calculated using least square method. One important parameter while estimating the AR model is the model order p . A small p can lead to poor estimation while a large p can lead to the problem of model estimation [15]. Two criteria are used to determine the model order namely, Akaike Information criterion (AIC) [28] and Bayesian Information criterion (BIC) [29]. For n variables model orders, AIC and BIC are given as follows:

$$\text{AIC}(p) = \ln(\det(\Sigma)) + \frac{2pn^2}{T}, \quad (2.9)$$

$$\text{BIC}(p) = \ln(\det(\Sigma)) + \frac{\ln(T)pn^2}{T}, \quad (2.10)$$

where Σ represents the noise covariance matrix. A time domain Granger causality is significant if A_{ij} 's are jointly significant or large relative to 0. This is done using an F -test on the null hypothesis that A_{ij} 's are 0 [14]. The tests are corrected using Bonferroni correction where a threshold of $\alpha/n(n-1)$ is used. The value of α is taken from the user in tool. A lower value of α yields better results. However, the computational time increases with decreasing value of α . Typically α is chosen as 0.01 or 0.005.

In the tool frequency domain Granger causality is also implemented. The Fourier transform of 2.6 and 2.7 is as follows:

$$\begin{pmatrix} A_{11}(f) & A_{12}(f) \\ A_{21}(f) & A_{22}(f) \end{pmatrix} \begin{pmatrix} X_1(f) \\ X_2(f) \end{pmatrix} = \begin{pmatrix} E_1(f) \\ E_2(f) \end{pmatrix}, \quad (2.11)$$

where components of A are $A_{lm}(f) = \delta_{lm} - \sum_{j=1}^p A_{lm}(j)e^{(-i2\pi fj)}$ and $\delta_{lm} = \begin{cases} 0, & \text{when } l = m \\ 1, & \text{when } l \neq m \end{cases}$. From 2.11 it can be shown that,

$$\begin{pmatrix} X_1(f) \\ X_2(f) \end{pmatrix} = \begin{pmatrix} H_{11}(f) & H_{12}(f) \\ H_{21}(f) & H_{22}(f) \end{pmatrix} \begin{pmatrix} E_1(f) \\ E_2(f) \end{pmatrix}, \quad (2.12)$$

where $\begin{pmatrix} H_{11}(f) & H_{12}(f) \\ H_{21}(f) & H_{22}(f) \end{pmatrix} = \begin{pmatrix} A_{11}(f) & A_{12}(f) \\ A_{21}(f) & A_{22}(f) \end{pmatrix}^{-1}$ is called the transfer matrix. The spectral matrix is given as follows [15]:

$$S(f) = \langle X(f)X^*(f) \rangle = \langle H(f)\Sigma H^*(f) \rangle, \quad (2.13)$$

where $*$ represents the complex conjugate transpose. If $S_{ii}(f)$ is the power spectrum variable i at frequency f then spectral Granger causality from j to i is given as following:

$$I_{j \rightarrow i}(f) = -\ln \left(1 - \frac{\left(\Sigma_{jj} - \left(\Sigma_{ij}^2 / \Sigma_{ii} \right) \right) |H_{ij}(f)|^2}{S_{ii}(f)} \right). \quad (2.14)$$

While calculating frequency domain Granger causality, two parameters are taken from the user for the calculation of Fourier coefficients: sampling frequency and maximum frequency to analyze.

Both time domain and frequency domain Granger causality methods are implemented in the tool. Usage of this method is shown in Chapter 3 and in Chapter 5 via application on case studies. This method has been proved to be very effective in capturing process connectivity and process topology. Since this method is based on AR models, for highly nonlinear data sometimes it cannot identify the causal direction and in addition to this the estimated model order may be much higher than the original system.

2.3 Summary

In this chapter theoretical aspects of transfer entropy method and Granger causality method are discussed. Both these methods can be very effective to find the causal pathways and to

identify the root cause of faults from the obtained process connectivity. Both the methods have certain advantages and disadvantages. However, it is not very easy to say which method is better. The obtained process connectivity from both methods needs to be validated from the Piping and Instrumentation Diagrams (P&IDs) [1], [16].

Chapter 3

Graphical User Interface (GUI) for Causality Analysis

This chapter discusses the Multivariate Data Analysis Tool (MDAtool) designed for data visualization and causality analysis. The MDAtool is designed in MATLAB. It can be run from MATLAB or run as a standalone application outside MATLAB. The tool is designed using MATLAB 7.9.0 (R2009b) and it is compatible with version 7 and later. The MDAtool runs on all operating systems. If the user wants to run the tool from MATLAB, the user has to make sure that the folder named “MDAtool” is located in the current directory of MATLAB. To avoid the hassle of changing the current directory every time the user runs MATLAB, one can save the path of the tool as follows: `File -> Set Path -> Add Folder -> Select ‘‘MDAtool’’ -> Ok -> Save -> Close`. Once the directory is able to locate the file, the tool will start by entering `MDAtool` in the command window of MATLAB. If the user decides to run the tool outside MATLAB, one can simply run the application “*MDAtool.exe*”. But it is mandatory that MATLAB has to be installed in the system.

The main interface of the MDAtool is shown in Figure 3.1. The user can get the necessary information about running the tool just by clicking on the “*Info*” button located in the toolbar as shown in the red box of Figure 3.1. A sample file can be observed by clicking on the “*Show Sample Data*” button as shown in the green box of Figure 3.1. This sample file provides information about the format of the data saved in MS Excel. Necessary information about the data saved in other formats are provided in section 3.1. It is worth mentioning that all the plots come with a user friendly feature that whenever the user clicks on any plot, it would pop up and open in a new figure window. Moreover the tool provides

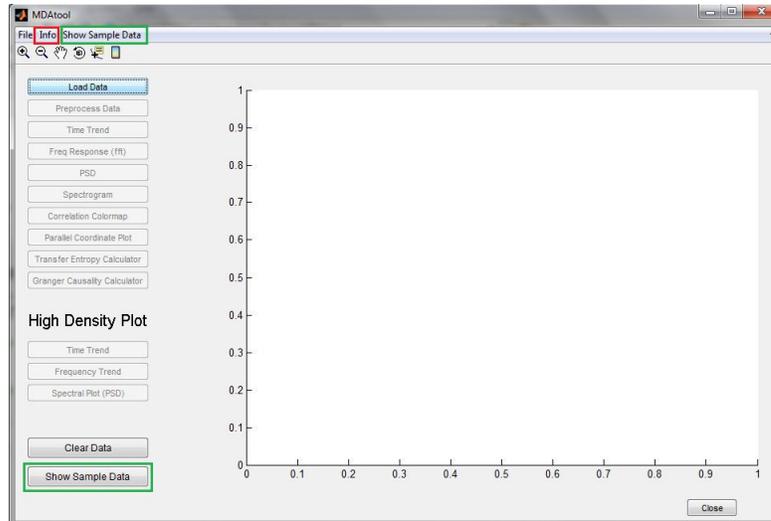


Figure 3.1: Outline of the MDAtool GUI

standard supports for the figures, e.g., Zoom-in, Zoom-out, Pan, Rotate 3D, Data cursor, and Colorbar.

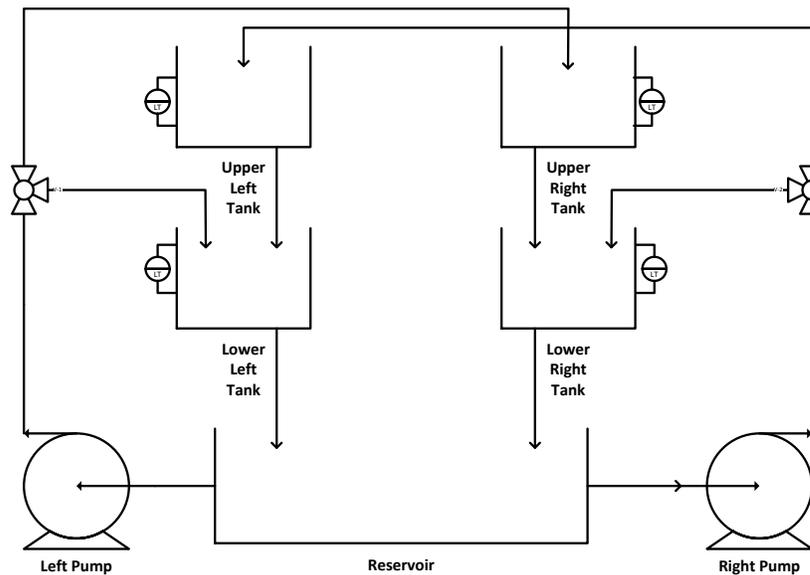


Figure 3.2: Experimental setup of the 4 - tank system

Features of the MDAtool are described based on the data obtained from the experiments performed on a 4 - tank system. The schematic of the 4-tank system is shown in Figure 3.2 where the arrows represent the direction of water flow. The left pump supplies water to the lower left tank and the upper right tank. On the other hand the right pump delivers water

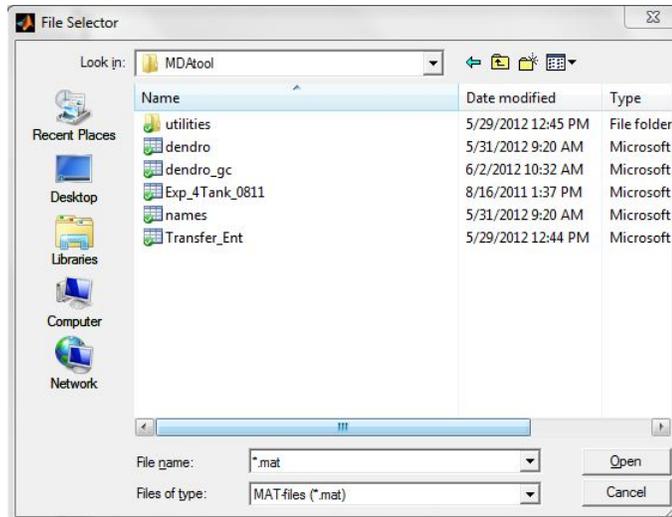


Figure 3.3: Dialogue box for selecting files to load process data

to the lower right tank and the upper left tank. In this experiment the goal is to control the levels of the lower two tanks using the right pump only. The flow from both pumps and levels of all the four tanks are the variables of interest and they are recorded for 1 hour at a sampling rate of 1 second. The data obtained from the experiment is loaded in the tool. The detailed functionalities of all the functions implemented in the tool are described below.

3.1 Data Loading

Process data saved in *.xls*, *.xlsx*, *.mat*, and *.txt* format can be loaded in the tool by clicking on the “**Load Data**” button or selecting *File -> Load Data* or pressing *Ctrl + L*. This will bring up the *File Selector* as shown in Figure 3.3 which is a standard operating system file selector dialogue box. The file the user wants to load can be located in any directory of the computer and browsed from the dialogue box. The tool does not require that the file must be saved in the current directory. Necessary file formats can be chosen from *Files of Type* located at the bottom the dialogue box.

If the process data is saved in either *.xls* or *.txt* format then the tag names must be in the first row and corresponding data needs to be arranged columnwise. The date/time information is optional but if it is there then the tool requires that they are placed in the first column. Even if the date/time is not there, the $\{1,1\}$ position must not be empty.

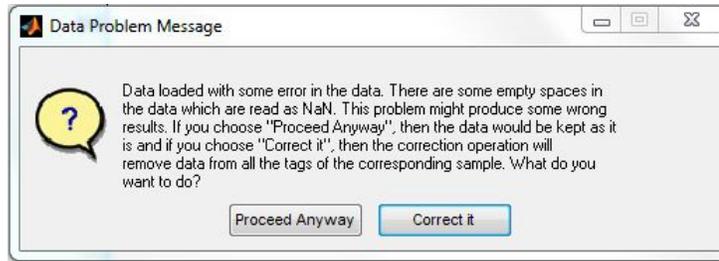


Figure 3.4: Dialog box appearing after loading the data when the file contains some empty spaces which might produce wrong results and the actions performed by the corresponding buttons

If the user wants to load *.mat* data then the process data must be arranged columnwise as done for *.xls* files. The tag names and date/time information are not needed. The tool uses the tag names given in the loaded process data. But the user must know the variable name which was used to save the *.mat* file. When the tool asks for it the user has to provide the name of the variable.

Sometimes it happens that there are some empty spaces in the process data which are read as “NaN” (Not a Number) in MATLAB and they might produce wrong results. In order to avoid this problem while loading data, the toolbox checks for such empty spaces and a message box as shown in Figure 3.4 pops up if there are any empty spaces in the data. If the user selects “*Proceed Anyway*” then the data would be kept as it is and if the user selects “*Correct it*” then the corresponding samples of all the variables would be removed.

3.2 Data Preprocessing

By clicking on the “**Preprocess Data**” button in the main interface the user has the option to remove outliers from the data, detrend and zero-mean centre the data, filter the data, and normalize the data. The user can select more than one option at a time. Figure 3.5 shows the corresponding dialogue box for data preprocessing. It is worth mentioning that when one option is selected, it is applied to all the tags present in the data. The functionalities of each function is described below:

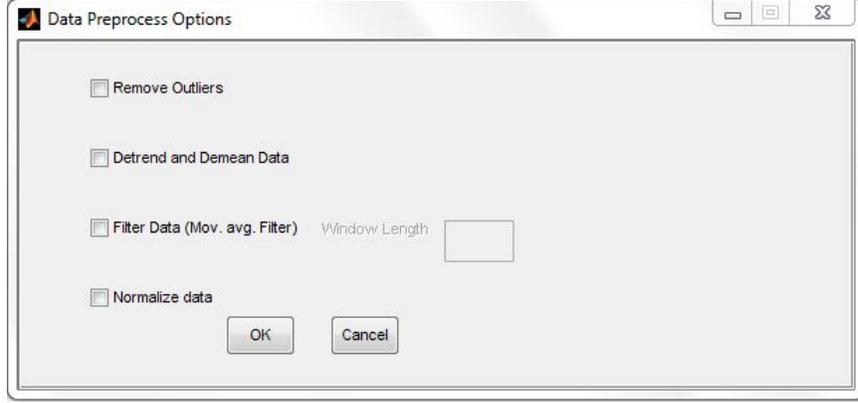


Figure 3.5: Data preprocessing interface of the MDAtool which comes with the options for outlier removal, detrending, demeaning, filtering, and/or normalizing the data

3.2.1 Remove Outliers

An outlier is an observation in a series of data which is numerically at a far distance from the rest of the data [30]. According to Grubbs, “*An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs*” [31]. Outliers can be present in the data because of some unwanted disturbances, noise or any other factor and most importantly it is not a part of the normal data. It has also been observed that sometimes because of the presence of outlier in the data, some misleading results are obtained. Therefore, the option for outlier removal is kept in the tool and if the user needs to remove the outliers, that can be done simply by checking on this function. In this tool, Grubbs’ Test [31] for detecting the outliers is performed and they are replaced by the mean value of the normal part of the data. For a variable x_i ; $i = 1, 2, \dots, N$; where N is the number of samples, the Grubbs’ Test statistic is defined as follows:

$$G = \frac{\max |x_i - \bar{x}|}{s}, \quad (3.1)$$

where, \bar{x} and s represent the mean and standard deviation of x . For the test of detecting outliers, the hypothesis of no outliers is rejected if

$$G > \frac{(N-1)}{\sqrt{N}} \sqrt{\frac{1}{N-2 + (t_{\alpha/(2N), N-2})^2}}, \quad (3.2)$$

where $(t_{\alpha/(2N), N-2})^2$ denotes the critical value of the t -distribution with $(N-2)$ degrees of freedom and a significance level of $\alpha/(2N)$.

3.2.2 Detrend and Demean Data

This optional function provides the option to the user to remove any linear trend present in the data (*detrend*) and remove the mean value of the data (*demean*) so that the user has a dataset which has zero mean and no linear trend. It has been observed that presence of any trend in the data might produce wrong results and that data scaling is very important especially for methods which require that the data can be modeled by linear processes e.g. Granger causality [17].

3.2.3 Filter Data

In the tool an option is provided for filtering the data using a moving average filter which is a low pass filter or a finite impulse response filter. While filtering the data the user needs to set the window length which is required for the filtering operation. Filtering the data will remove the high frequency components present in the data or in other words it will remove a portion of the high frequency noise from the data.

3.2.4 Normalize Data

Some causality detection methods, e.g., Granger causality [16], require that data should be normalized. This function normalizes all the variables present in the loaded file between 0 and 1. It may be mentioned that even if the user does not normalize the data using this feature, before performing Granger causality analysis default data normalization is carried out so that accurate results are obtained. For a tag x , the normalized version $x_{normalized}$ is as follows:

$$x_{normalized} = \frac{x - \min(x)}{|\max(x) - \min(x)|}. \quad (3.3)$$

3.3 Time Trend

The MDAtool comes with several options for data visualization and among them the first one is time trend. When the user clicks the “**Time Trend**” button, then a dialogue box “*Time Options*” pops up and provides the user all the tag names as can be seen in Figure 3.6. From there, one can select as many tags as possible and plot them by clicking on “*Plot Selected*” or all the tags by clicking on “*Plot All*” to be plotted. While plotting the tags,

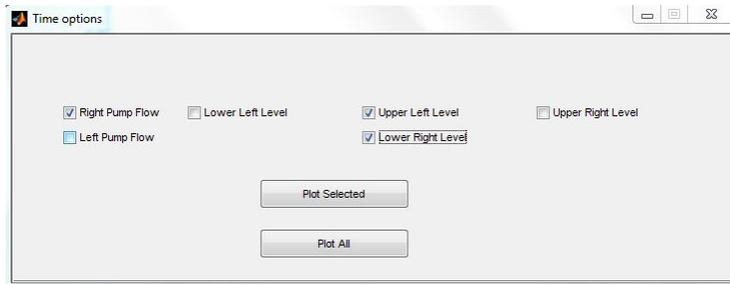


Figure 3.6: Dialogue box that appears upon clicking on “*Time Trend*” where the user can choose to plot any number of interested tags or all of them present in the data

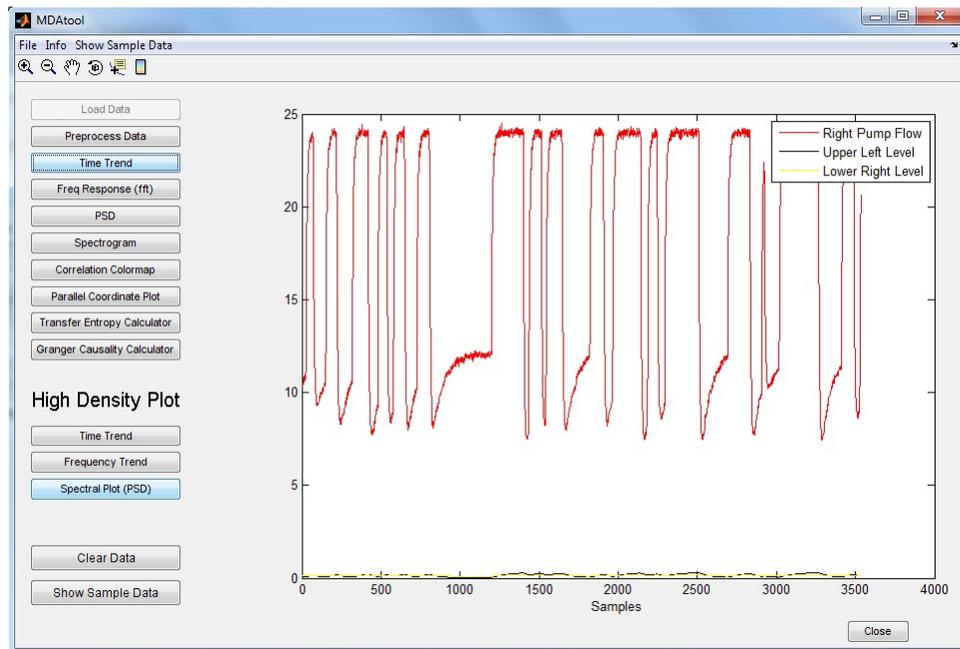


Figure 3.7: Time trend of 3 selected tags from Figure 3.6

the real values of the selected tag(s) are shown in terms of the number of samples. Figure 3.7 shows the time trends of 3 selected tags namely *Right Pump Flow*, *Upper Left Level*, and *Lower Right Level*. Legends are used for the purpose of better visualization. It may be mentioned that the toolbox can handle effectively at most 50 tags at a time. If the loaded process data contains more than 50 tags then all the tag names would not be displayed. But it is still possible to see the time trend of all the tags by clicking on “*Plot All*”.

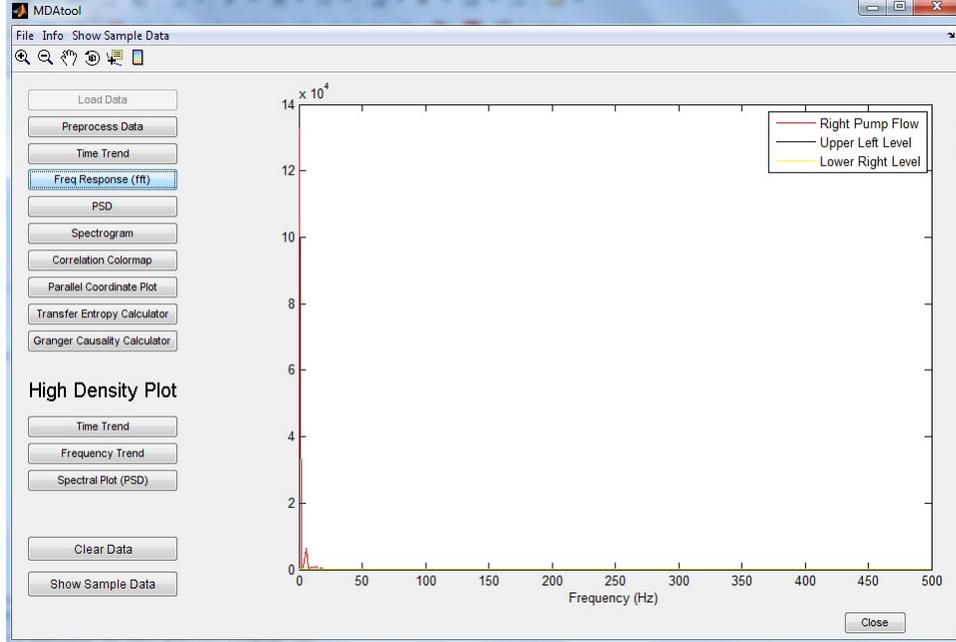


Figure 3.8: Frequency response of 3 selected tags

3.4 Frequency Response

The next available option for data visualization with the MDAtool is in frequency domain to show the user the frequency response of the tags of interest(s). When the “**Frequency Response (fft)**” button is clicked a dialogue box named “*Freq Options*”, similar to the “*Time Options*” appears where all the tag names are provided. The user has the liberty to plot frequency response of the tag(s) of interest by clicking on “*Plot Selected*” after checking on the tag names. The user can also select “*Plot All*” to see the frequency response of all the tags. The frequency response of the corresponding tag is calculated by performing Fourier Transform according to

$$X(f) = \sum_{k=1}^n x(k)\omega_N^{(k-1)(f-1)}, \quad (3.4)$$

where $x(t)$ is a time-domain signal, $X(f)$ is its corresponding frequency-domain signal, N is the total number of samples, and $\omega_N = e^{-2\pi i/N}$ is an N^{th} root of unity. Figure 3.8 shows the frequency response of 3 selected tags.

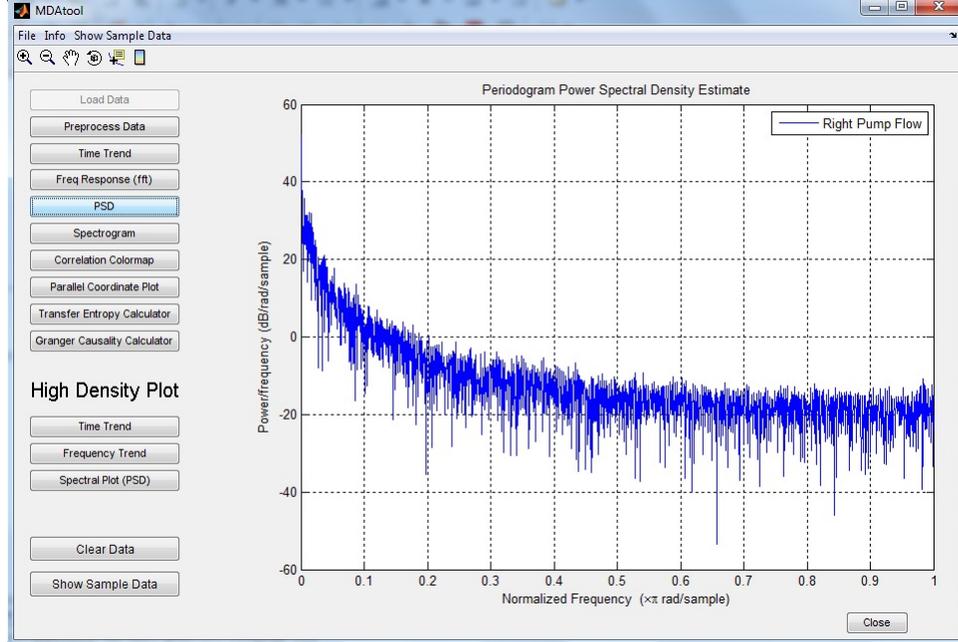


Figure 3.9: Power Spectral Density (PSD) of a selected tag, *Right Pump Flow*

3.5 Power Spectral Density

Another data visualization option in the frequency domain is provided with the button named “**Power Spectral Density (PSD)**”. Same as the previous data visualization options, the user can choose to plot the PSD of all the tags or a chosen number of tag(s). The PSD of a tag x_k ; $k = 1, 2, \dots, N$; where N is the number of samples, at any frequency ω is calculated as follows:

$$S(e^{j\omega}) = \frac{1}{N} \left| \sum_{k=1}^N x(k) e^{-j\omega k} \right|^2. \quad (3.5)$$

The PSD provides the information on how much power in dB a selected tag has in the normalized frequency. The options are similar to the options shown in Section 3.3. Figure 3.9 shows the PSD of the selected tag “*Right Pump Flow*”. If the user selects multiple tags or all the tags then the corresponding PSD of individual tags overlap with each other and it is not easy to infer information from there. Therefore, it is recommended to plot the PSD of individual tags to avoid confusion and better understanding of the results.

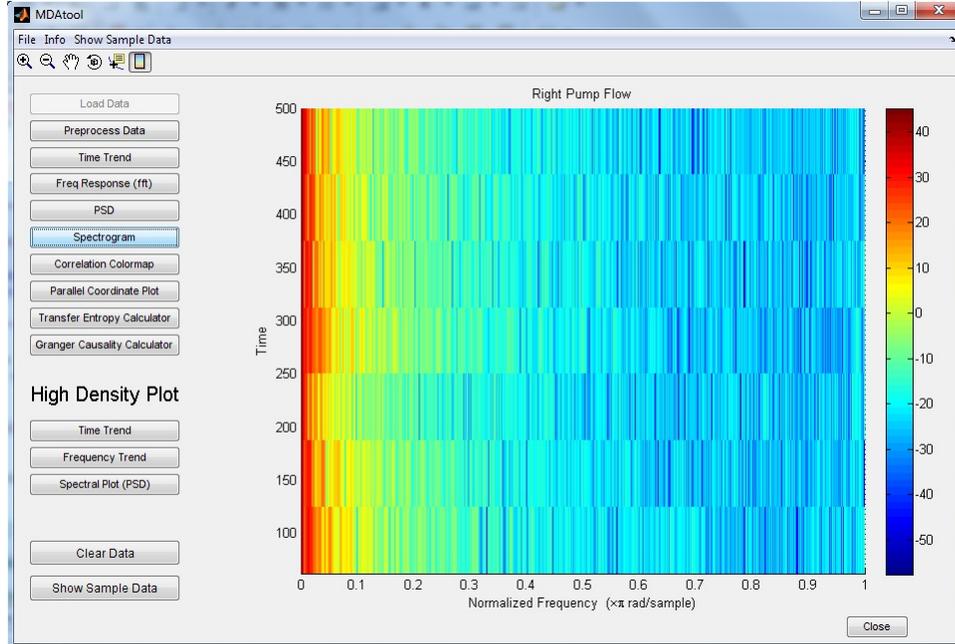


Figure 3.10: Spectrogram of a selected tag, *Right Pump Flow*

3.6 Spectrogram

The last option for data visualization in the frequency domain is the spectrogram. When the user clicks on the “**Spectrogram**” button, the tool shows the variation in the spectral components of a tag with respect to time as calculated using the following:

$$X(m, \omega) = \sum_{k=-\infty}^{\infty} x(k)w(k - m)e^{-j\omega k}, \quad (3.6)$$

where w represents the window size. The user has the option to plot the spectrogram of all the tags or tags of interest similar to 3.3. Figure 3.10 shows the spectrogram of the tag “*Right Pump Flow*”. If the user selects multiple tags or all the tags then the corresponding spectrogram of individual tags overlap with each other and it is next to impossible to get the results from there. Therefore, it is recommended to plot the spectrogram of individual tags to avoid confusion and better understanding of the results.

3.7 High Density Plot

High density plots are used to show the behavior of the tags present in the process data at a glance. These plots are highly useful to show a huge amount of data in a small area and

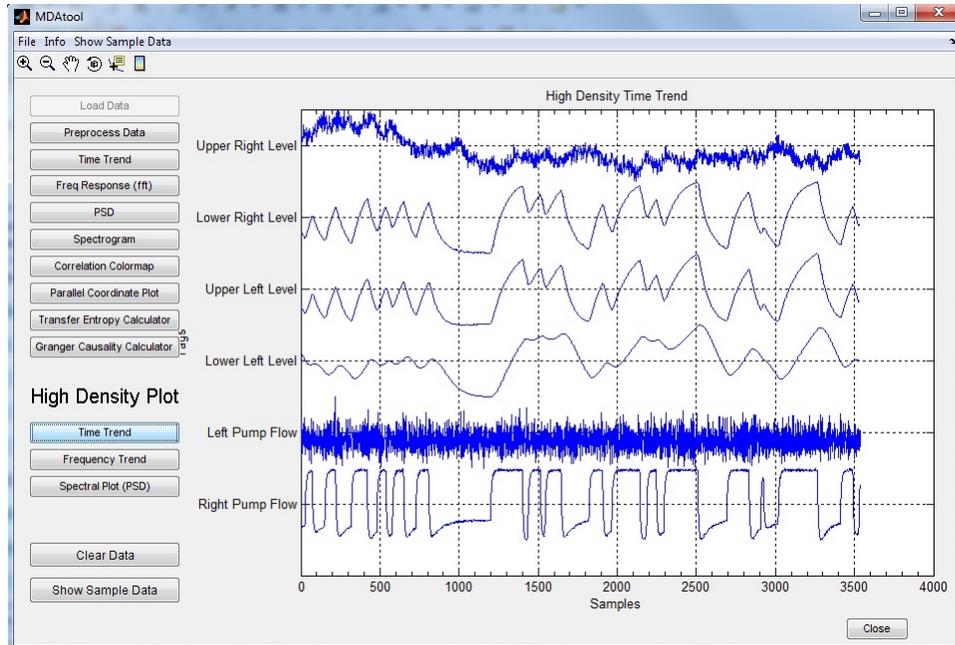


Figure 3.11: High density time trends

contain a good amount information about the overall system or the tags in the process data. From this plot the user can have an idea about the trend of the variables present in the file and how they are changing. Moreover a rough idea of the similarities and dissimilarities of one variable with other(s) can be obtained from the high density plot. Each row in the high density plot represents a tag and the results are normalized before plotting. Three types of high density plots are implemented in the MDAtool. Out of these three kinds, one is in the time domain and the other two are in the frequency domain. They are discussed below:

3.7.1 High Density Time Trend

When the user clicks on the **“Time Trend”** button in high density plot screen, then the time trend of all normalized tags is plotted. Individual tags are normalized using the formula given in 3.3. So the normalized values range from -1 to +1. All the normalized tags are placed along the rows and then they are plotted so that there is no overlap with each other. The user gets a complete idea of the behavior of all the tags for the whole duration as well as a rough knowledge of relationship regarding the causal relationships among the tags. Figure 3.11 shows the high density time trend of all the tags of interest for the experiment performed on the 4-tank system. Now the variation in the tank levels with pump flows are

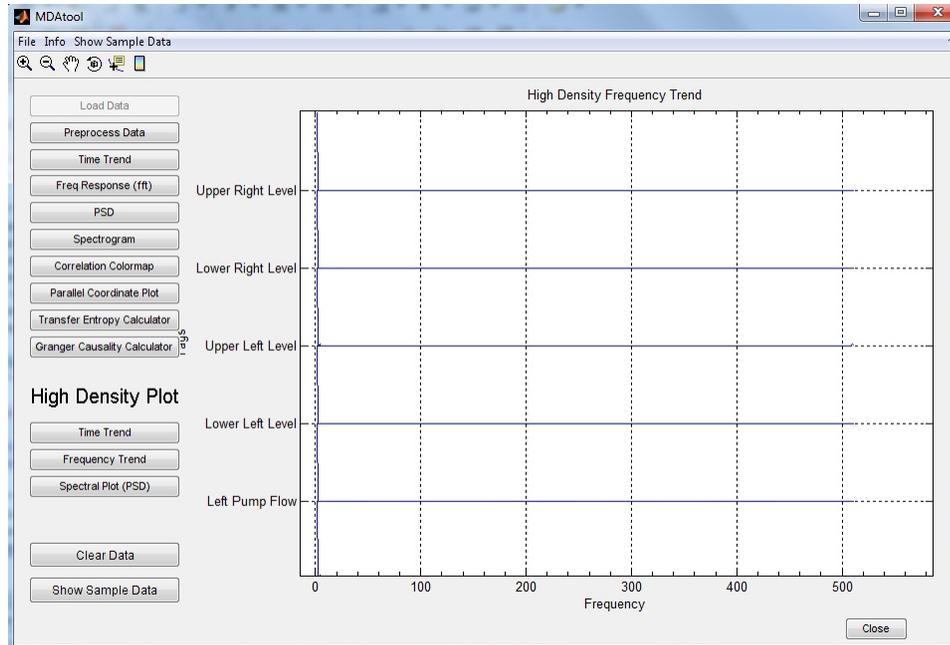


Figure 3.12: High density frequency response

much more evident than it was in Figure 3.7. This is due to fact that in “*Time Trend*” the original values are used to plot the time response where the units may be different and one tag may have a higher range or value compared to another and vice versa. On the other hand for “*High Density Time Trend*” normalized values are used and the values of all the tags have the same range which provides a better visualization.

3.7.2 High Density Frequency Trend

“**Frequency Trend**” under high density plots shows the normalized frequency response of all the tags. This function first performs the FFT using the similar formulation as in Section 3.4 and then normalizes the Fourier coefficients using the maximum coefficient. Therefore they are ranged between 0 and 1. The plotting principle is same as subsection 3.7.1. Figure 3.12 shows the high density frequency response of all the tags.

3.7.3 High Density Spectral Plot

The last option under high density plot is “**Spectral Plot (PSD)**” which shows the normalized PSD calculated in Section 3.5. The plotting principal is same as subsection 3.7.1. Figure 3.13 shows the high density spectral plot of all the tags.

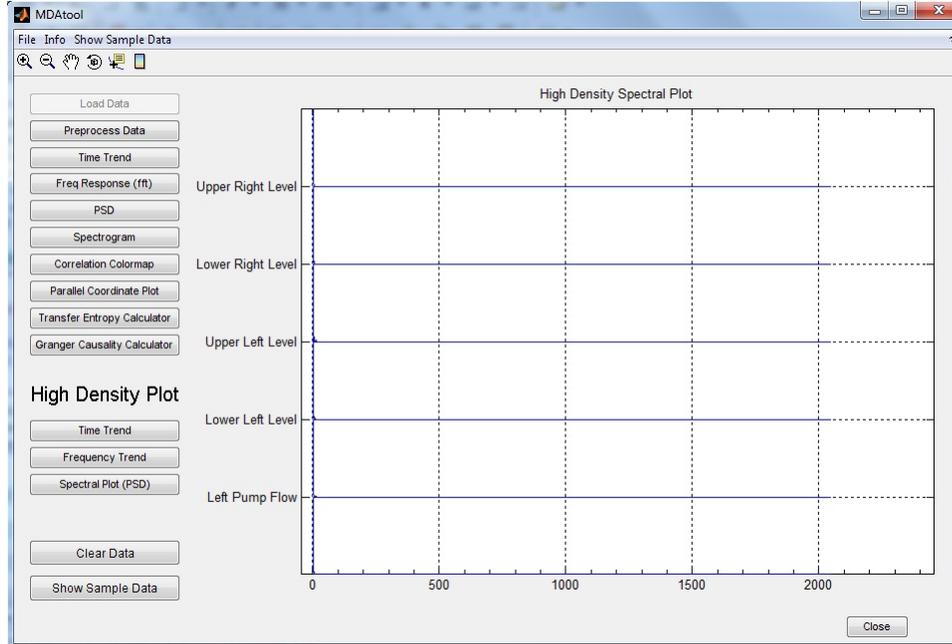


Figure 3.13: High density spectral plot

3.8 Correlation Color Map

Cross-correlation finds the similarity or dissimilarity between two time-series signals. In this tool, the “**Correlation Colormap**” button provides the user two options to graphically display the correlation among all the variables present in the input file in terms of a colormap and colorbar: 1) Considering no lag, and 2) Considering lag. It is important to note over here that while calculating the correlation coefficient pairwise calculation is performed.

3.8.1 Considering No Lag

“*Considering No Lag*” calculates the cross-correlation coefficients between two time series signals considering the fact that there is no time delay between them. For two time series signals x and y , the cross-correlation coefficient is defined as follows:

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x\sigma_y} = \frac{E[x - \mu_x]E[y - \mu_y]}{\sigma_x\sigma_y}, \quad (3.7)$$

where μ_x, μ_y are the means and σ_x, σ_y are the standard deviations of x and y respectively. Figure 3.14 shows the cross-correlation among the variables considering no time lag between them. Evidently the colormap is symmetric due to the fact that $\rho_{x,y} = \rho_{y,x}$. Along the diagonal the correlation coefficient is always going to be one since it is the correlation of

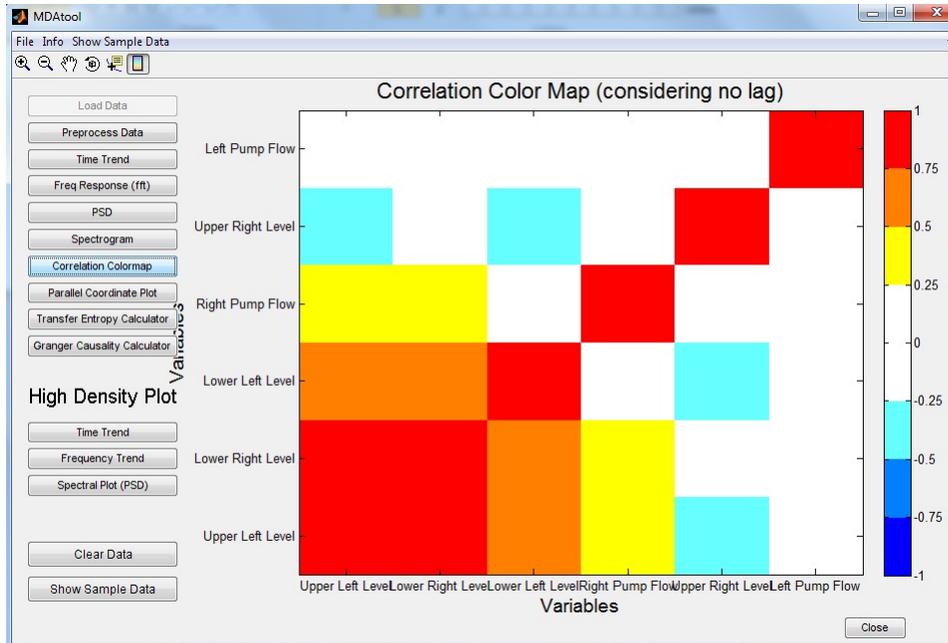


Figure 3.14: Correlation colormap among the variables considering no lag

a variable with itself. Moreover, while plotting the colormap discrete colors are used to distinguish the level of correlation. It is evident from Figure 3.14 that Lower Right and Upper Left Levels are highly correlated with each other. Moreover, they are also correlated with Lower Left Level to a lesser extent.

3.8.2 Considering Lag

On the other hand, “*Considering Lag*” assumes that there is a lag between two time-series signals. While performing correlation analysis it is very important to consider the effect of lag between two time series signals. It is because of the fact that the affected variable can be far away from the affecting variable or there might be some intermediate variables present in between them. Most often there is always some time lag between two variables in real process data. To eliminate the effect of time delay different time lags are assumed and maximum correlation coefficient is computed and it is known as the real coefficient [1], [2], [20].

If x and y are two process variables with mean μ_x , μ_y and standard deviation σ_x , σ_y ,

then with a lag of k on y the correlation coefficient is defined as follows:

$$\phi_{xy}(k) = \frac{E[(x_i - \mu_x)(y_{i+k} - \mu_y)]}{\sigma_x \sigma_y}, k = -n + 1, \dots, n - 1. \quad (3.8)$$

The expectation can be estimated as follows:

$$\hat{\phi}_{xy}(k) = \begin{cases} \frac{1}{n-k} \sum_{i=1}^{n-k} (x_i - \mu_x)(y_{i+k} - \mu_y) / \sigma_x \sigma_y, & \text{if } k \geq 0, \\ \frac{1}{n-k} \sum_{i=1-k}^n (x_i - \mu_x)(y_{i+k} - \mu_y) / \sigma_x \sigma_y, & \text{if } k < 0. \end{cases} \quad (3.9)$$

The correlation coefficient is obtained by considering a fixed time lag for one variable compared with the other one. Therefore it may be considered that the absolute maximum value is the real correlation coefficient and the corresponding lag is the time difference between the two variables. If $\phi^{max} = \max_k \{\phi_{xy}(k), 0\} \geq 0$, $\phi^{min} = \min_k \{\phi_{xy}(k), 0\} \leq 0$, and k^{max} and k^{min} are the corresponding arguments then the estimated time delay from x to y is as follows:

$$\lambda = \begin{cases} k^{max}, & \text{if } \phi^{max} \geq -\phi^{min}, \\ k^{min}, & \text{if } \phi^{max} < -\phi^{min}. \end{cases} \quad (3.10)$$

Therefore the actual correlation is $\rho = \phi_{xy}(\lambda)$. The sign of ρ indicates whether the corre-

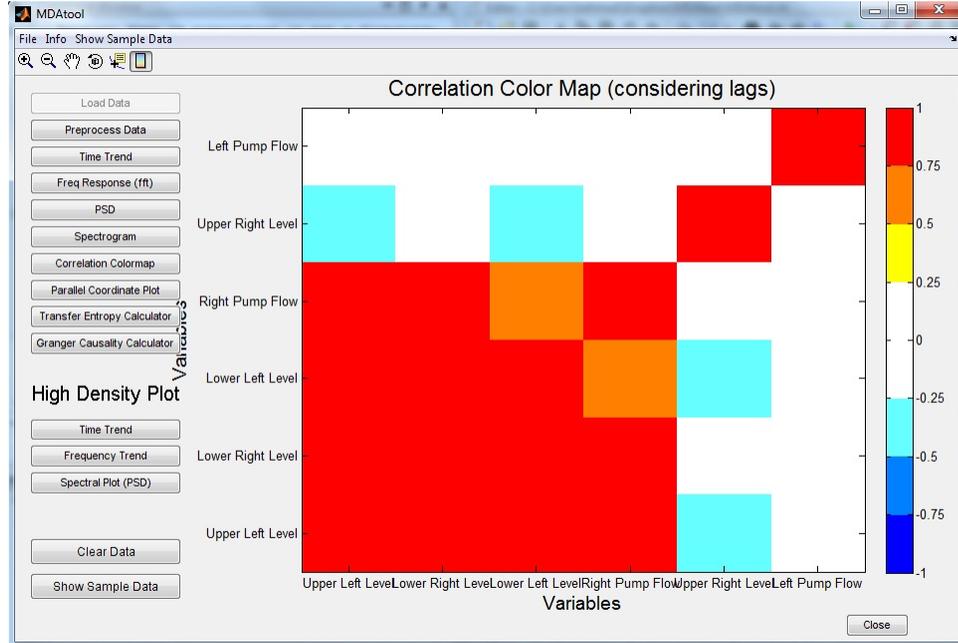


Figure 3.15: Lag adjusted correlation colormap between the variables

lation is positive or negative. If the sign of λ is negative then the delay is from y to x . It is important to note over here that this directionality does not necessarily mean causality since there might be other causes which led to this correlation [1].

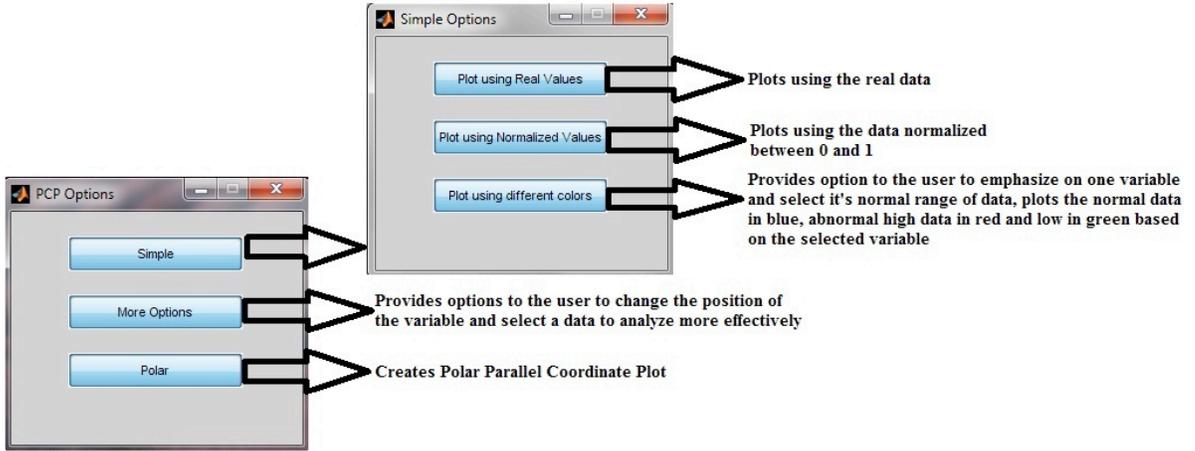


Figure 3.16: Options provided for parallel coordinate plot (PCP)

Figure 3.15 shows the correlation colormap among the variable after taking the effect of lag into consideration. As can be seen in Figure 3.15 compared to Figure 3.14 that Lower Right Level, Upper Left Level, Lower Left Level, and Right Pump Flow are highly correlated with each other. From the experimental setup it is also known that the 4-tank system is a highly correlated system and according to the control view point the results of lag adjusted correlation analysis is more accurate than that of considering no lag. In the real process industries it is impossible to have two tags with no time delay between them as it takes a certain amount of time for information transfer.

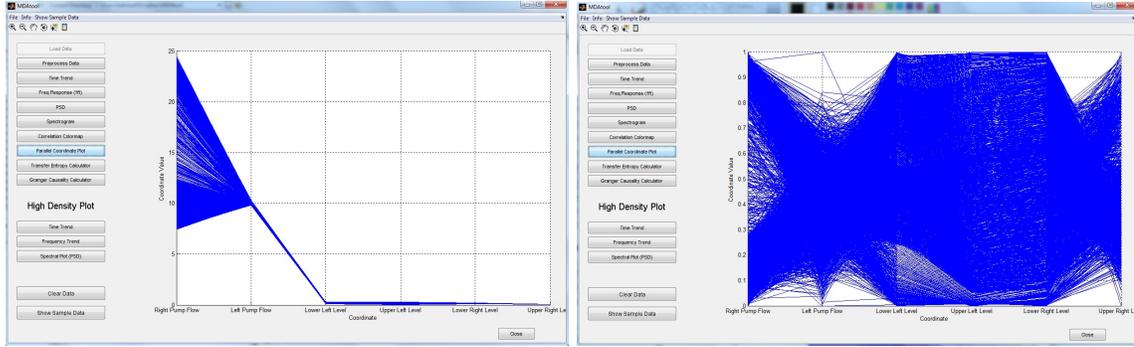
It is noteworthy to mention that since cross-correlation is a statistical measure, it is prone to noise, disturbance, and data length. Moreover, non-linear causal relationships do not necessarily show up using this method [1].

3.9 Parallel Coordinate Plot

The parallel coordinate plot (PCP) is a graphical method of plotting multivariate data obtained without any calculation and is being used extensively for causality analysis [32], [33], [34]. In the parallel coordinate plot the variables are placed horizontally and the values of the variables are placed vertically in the corresponding axis. The GUI tool comes with several options of visualizing the PCP of a multivariate process data, as shown in Figure 3.16. They are discussed in details below.

3.9.1 Linear Parallel Coordinate Plot

The linear parallel coordinate plot plots the multivariate process data in the linear plane. The plots are explained below.



(a)

(b)

Figure 3.17: Simple linear PCP: (a) using real values, (b) using normalized values

Figure 3.17(a) shows a simple PCP of the experiment performed on the 4-tank system which can be obtained by clicking on “Plot using real values”. While plotting the simple PCP original values of the variables are used. Since the flow rates are much higher than the tank levels, it is not very easy to infer much information for the tank levels from there. That is why another option named “Plot using normalized values” is implemented in the tool so that the user can have a better understanding of the dataset. Figure 3.17(b) shows the PCP plotted using the normalized values of all the variables. Since the data is now normalized all the variables now have the same scale and provides a better understanding. On the other hand, if a dataset is formed using a large of number of samples for all the variables, it is very hard to identify the impact of one variable on other since only one color is used for Figure 3.17. To have much more information from the PCP, another option

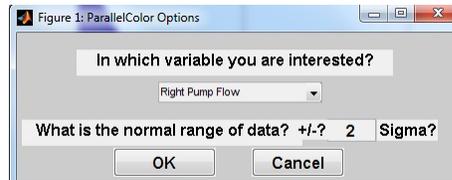


Figure 3.18: Dialogue box where the user can choose a variable and its normal operating region

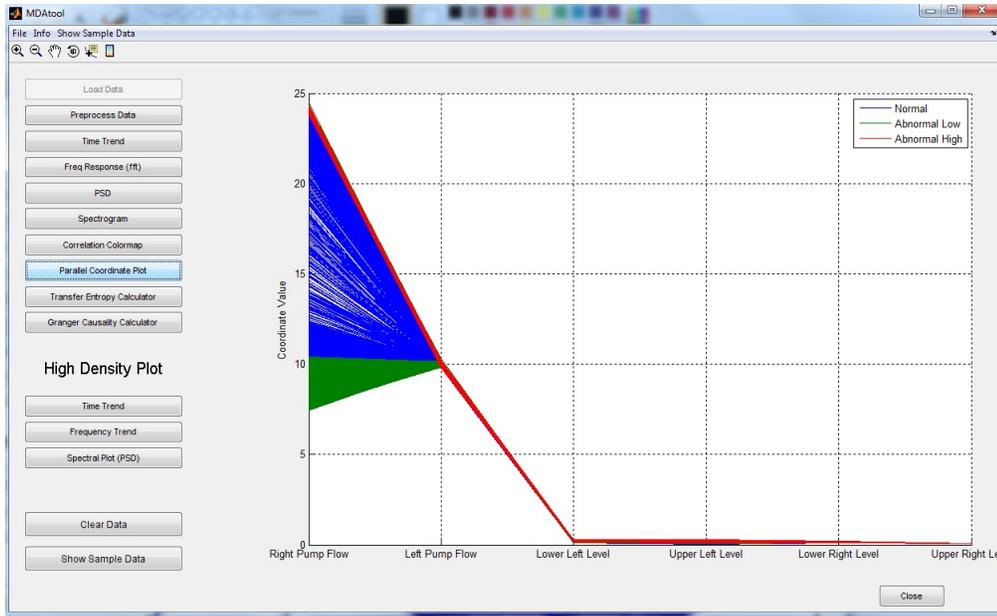


Figure 3.19: PCP with different colors showing the normal data, abnormal low data, and abnormal high data based on the *Right Pump Flow*

“Plot using different colors” is provided. For PCP with different colors one dialogue box is provided as shown in Figure 3.18 where the user has to select the variable and see the impact of that variable on others by defining its normal region. The normal data is defined by the user as $mean \pm c \times standarddeviation$ of the selected tag and the value of c must be provided. In Figure 3.19 the chosen variable is “*Right pump flow*” and c is taken as 2. Abnormal high values shown in red represent the values higher than the defined normal range whereas Abnormal low values shown in green represent the opposite. As can be seen the user can now have an idea of the effects of higher values or lower values of the selected tag on the other tags, e.g., the higher values of flow rates increases the tank levels as can be seen from Figure 3.19. It should be mentioned that the user has the freedom to choose any tag of the input file and can choose the normal range of data based on the process knowledge and original values of the variables are used.

So far the user did not get any option to place two tags of interest side-by-side. When the user clicks on “*More Options*” Figure 3.20 appears. Now any tag(s) can be chosen from the drop down menu to place them side-by-side. Along with PCP on the top a scatter plot on the bottom of the chosen tag(s) are shown from which the user can select a range of

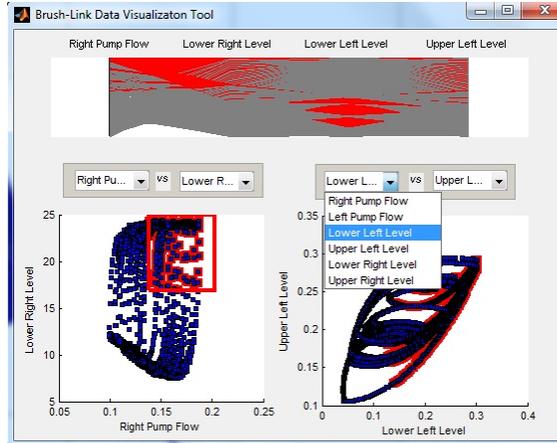


Figure 3.20: Dialogue box providing option to plot two tags side-by-side in PCP and see the impact of one variable on the other by selecting a range of data from the scatter plot

data to see the impact of that particular region of data on the other tag(s) and vice versa.

3.9.2 Polar Parallel Coordinate Plot

The last but not the least option for PCP is the polar PCP where data is plotted in polar domain. The user can infer the timing information from here as well, since the older values are plotted using the darker color (black) and the newer values are plotted using the lighter color (white) as can be seen in Figure 3.21. The Polar PCP shows the user a normal range of data defined as $mean \pm 2 \times \sigma$ in the *Grey* zone in the middle where σ is the standard deviation. If the data lies beyond 2σ but below 3σ then it is plotted in the *Yellow* region. And if the data goes beyond 3σ then the *Red* region is used. Also using PCP the user can see the transition of the data which gives a better understanding of the process. It is worth mentioning that Polar PCP takes a lot of time to generate. A warning message is given to user before starting of the application. Upon approval the process starts and can be interrupted anytime simply by pressing *Ctrl+C* on the keyboard.

3.10 Transfer Entropy

The theoretical background behind transfer entropy is discussed in Section 2.1. In this section only the part related with the MDAtool will be discussed. The parameter values for calculating transfer entropy are chosen as, $k = 0$, $l = 2$, $h = 1$, and $\tau = 1$. Results of transfer entropy can be obtained simply by clicking on “**Transfer Entropy Calculator**”.

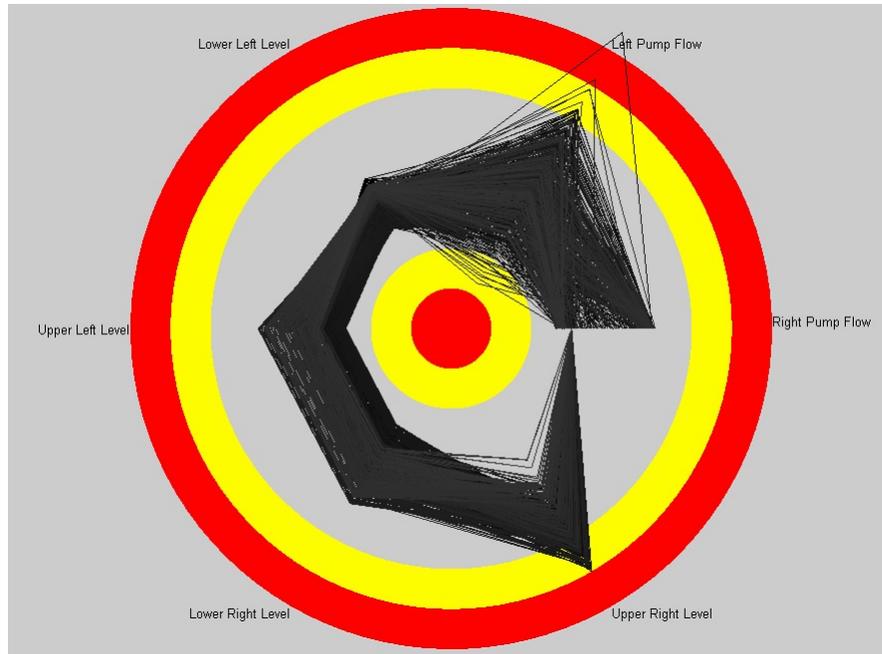


Figure 3.21: Polar PCP where darker color represent the initial values and lighter color represent the final values

Calculation of transfer entropy may take a long period of time to process. That is why a warning message is given to the user before processing. Upon approval from the user the process begins and the results are shown in the form of colormap and connectivity diagram. In the colormap a colorbar is used to show the strength of the connectivity between the two variables. Directed connections are shown using a *Arrowed Green Line* to show the direction of causal flow. *Red Lines* are used to show the feedback or bidirectional connectivity between the variables. It is noteworthy to mention that pairwise analysis is performed. Direct or indirect causality is not considered in this thesis.

The results of transfer entropy method for the experiment on the 4-tank system using a colormap is shown in Figure 3.22(a) and the corresponding connectivity diagram is shown in Figure 3.22(b). As can be inferred from Figure 3.22, there is no relationship between the flow rates of the pump and it is true from the experimental setup. The upper right tank receives water only from the left pump. The upper right and lower right tank heights change simultaneously as can be seen in Figure 3.11 and this is the reason for the bidirectional connection between them. There must be a bidirectional connectivity between the lower right level and the right pump flow because of the control strategy of the system and it is

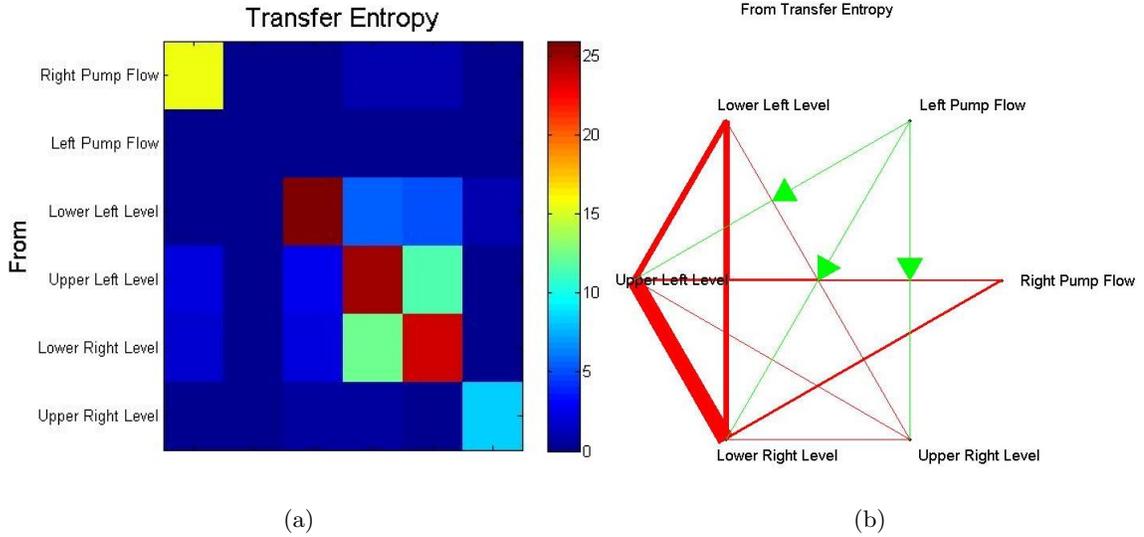


Figure 3.22: Results of transfer entropy method in the form of: (a) colormap, (b) connectivity diagram

obtained from the results of transfer entropy. On the other hand, the right pump controls the lower left level through upper left level and it is also confirmed from the results of transfer entropy. There is a bidirectional connectivity path between the lower levels because they change simultaneously. Very small amount of flow from the left pump goes to the lower left tank. Moreover, the lower left tank is controlled by the right pump. Therefore, the causal path from the left pump to the lower left level is missing. It is worth mentioning that, the thickness of the connectivity paths are proportional to the measure of interactions.

3.11 Granger Causality

Theoretical aspects of Granger causality have already been discussed in Section 2.2. When the user clicks on “**Granger Causality Calculator**” a dialogue box as shown in Figure 3.23 appears and the user has to provide the parameter values. The user has the liberty to choose either time domain Granger causality or frequency domain Granger causality or both. For time domain Granger causality calculation only the probability threshold is required. On the other hand for the calculation in frequency domain both the sampling frequency and maximum frequency to be analyzed must be given by the user. A connectivity diagram appears after the calculation of time domain Granger Causality as shown in Figure 3.24(a)

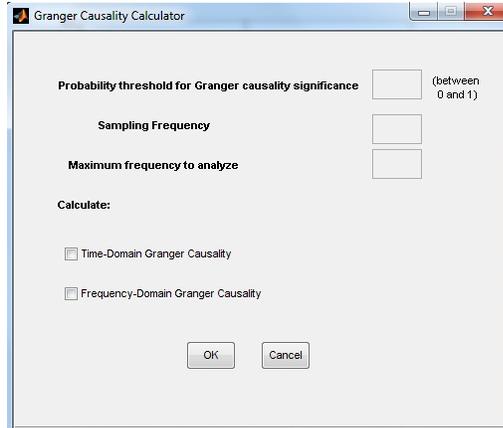


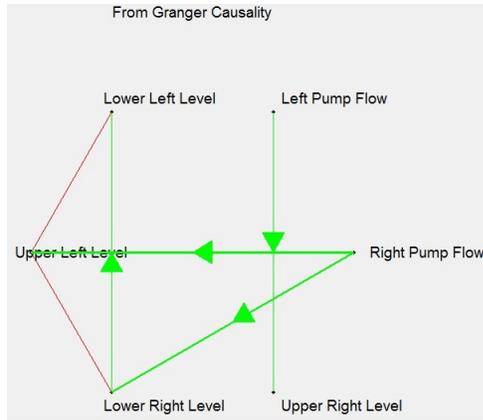
Figure 3.23: Dialogue box asking the user to give the parameter values to calculate Granger causality

and the results of frequency domain Granger causality are shown in 3.24(b). Both these figures have the same interpretation and the only difference being the domain in which this analysis is performed.

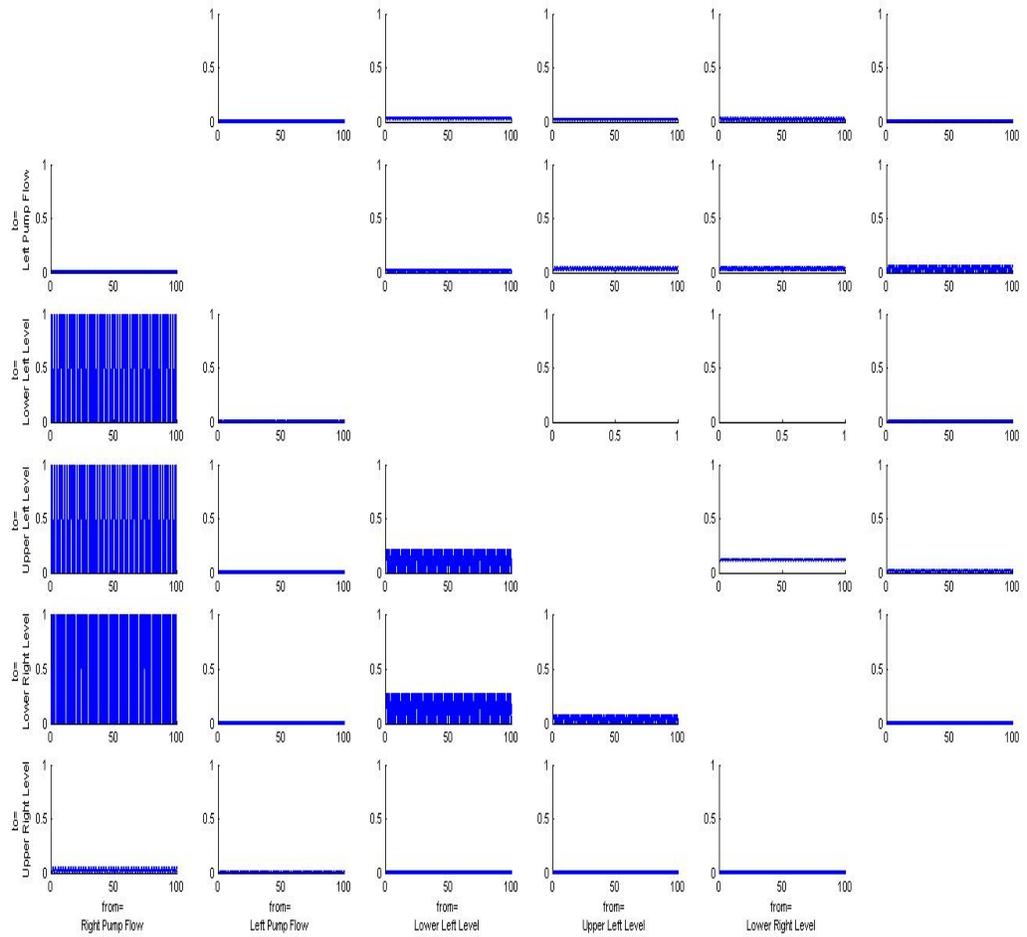
The probability threshold is set as 0.01, sampling frequency is set to 1, and 100 Hz is chosen as the maximum frequency to be analyzed. From the connectivity diagram of Figure 3.24(a) it is evident that flow of left pump has a causal direction to the upper right level which is true. Right pump affects the lower right level and upper left level which is also true from the experimental setup. Though there is no physical connection between the lower levels, the possible reason for the connection between them might be the fact that they change simultaneously. Obviously the flow from the right pump is connected to the lower left level via the upper left level.

3.12 Summary

In this chapter the tool for causality analysis and visualization of multivariate process data *MDAtool* is discussed. Users are referred to this chapter for detailed understanding and usage of the tool. This tool provides the freedom of using different kinds of data and also gives different data preprocessing options before doing any analysis. The tool provides a number of options for data visualization in both time domain and frequency domain. Different kinds of causality analysis can be done by the tool simply by loading the data. It is noteworthy to mention that, both transfer entropy and Granger causality are data



(a)



(b)

Figure 3.24: Results of Granger causality method for the 4-tank system setup in: (a) time domain analysis, (b) frequency domain analysis

based methods. The results obtained from causality analysis must be validated with the process or Piping & Instrumentation Diagrams (P&IDs) of the system before reaching any conclusion [1].

Chapter 4

Alarm Management

This chapter describes a tool designed for alarm management with a view to identifying the top bad actors in the system so that the user of the tool gets a detailed idea on which tags need more attention. The algorithms implemented in the tool are formulated by Sandeep R. Kondaveeti, *PhD* Candidate, Department of Chemical and Materials Engineering, University of Alberta. The tool is developed using the algorithms with permission from Sandeep. The tool is named as *alarm_tool*. The outline of the tool is shown in Figure 4.1. The user can get the necessary information about running the tool just by clicking on the “*Info*” button located in the toolbar. The *alarm_tool* is designed in MATLAB. It can be run from MATLAB or run as a standalone application outside MATLAB. The tool is designed using MATLAB 7.9.0 (R2009b) and it is compatible with version 7 and later.

The *alarm_tool* runs on all operating systems. If the user wants to run the tool from MATLAB, the user has to make sure that the folder named “*alarm_tool*” is located in the current directory of MATLAB. To avoid the hassle of changing the current directory every time the user runs MATLAB, one can save the path of the tool as follows: `File -> Set Path -> Add Folder -> Select ‘‘alarm_tool’’ -> Ok -> Save -> Close`. In either of the way, the tool will start by entering `alarm_tool` in the command window of MATLAB. If the user decides to run the tool outside MATLAB, he/she can simply run the application “*alarm_tool.exe*”. But it is mandatory that MATLAB has to be installed in the system.

It is worth mentioning that all the plots come with a user friendly feature that whenever the user clicks on any plot, it would pop up and open in a new figure window. Moreover the tool provides standard supports for the figures, e.g., Zoom-in, Zoom-out, Pan, Rotate 3D,

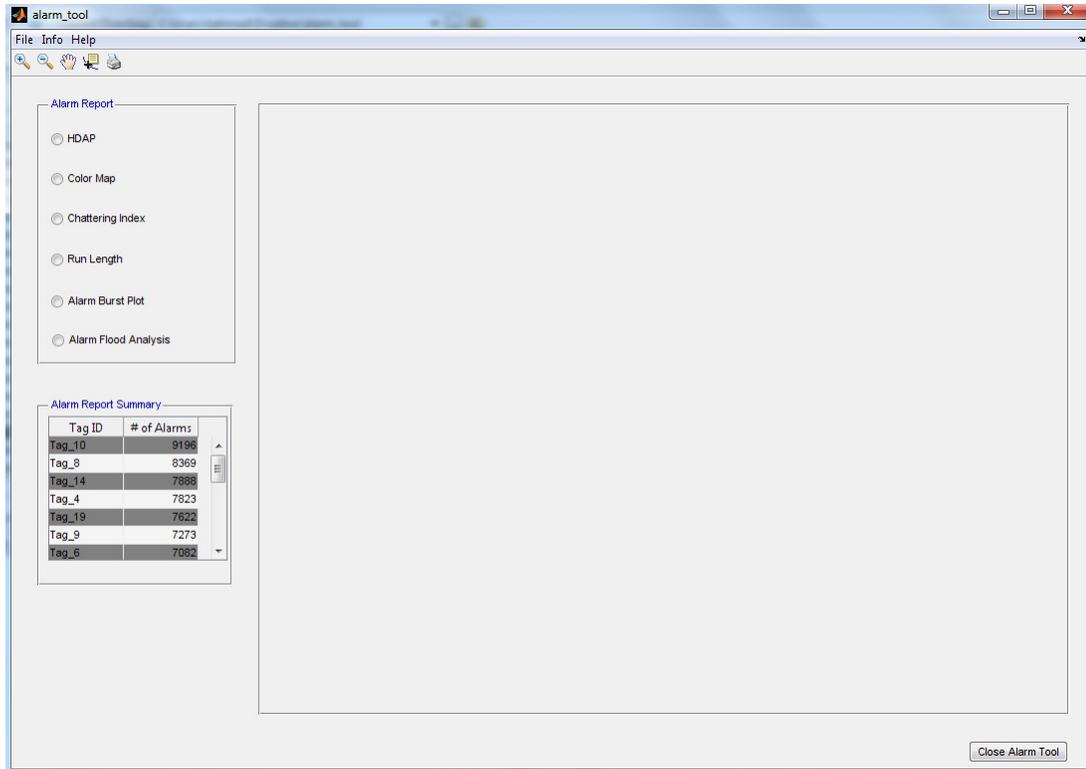


Figure 4.1: Outline of the *alarm_tool*

Data cursor, and Colorbar. Also when the tool starts, it shows the total number of alarms for individual tags present in the alarm data at the lower left corner of the tool. The user has to scroll up and down to see all the tags if there are a lot of tags involved.

This tool is described with computer generated alarm data of 20 tags. This tool only deals with alarm data where the necessary information is: time of alarm rise, tag name, tag identifier (id) of alarm rise, alarm message, return to normal (RTN) time, and RTN id. The detailed functionalities of all the functions implemented in the tool are described below.

4.1 High Density Alarm Plot

The High Density Alarm Plot (HDAP) is very useful for the purpose of visualization of large amount of alarm data in the system. The user of the tool can simply click on the radiobutton **HDAP** to get the HDAP of the loaded alarm data. Visual identification of system instability is the main advantage of using HDAP. Also it is possible to get the information on chattering tags and redundant alarms [10]. Each row of the HDAP corresponds

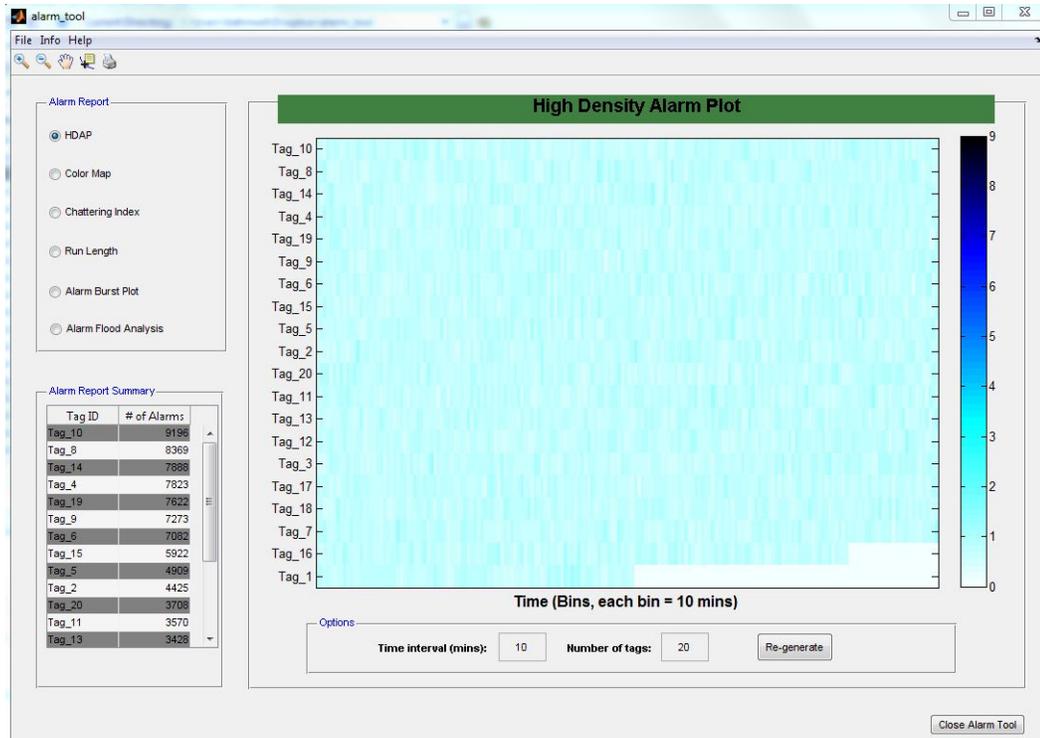


Figure 4.2: HDAP of 20 tags with number of alarms ranging from 537 to 9196 for individual tags

to a unique tag and it helps the user to find out for which tags alarms were raised during the whole period. As a result the user can look back and identify what was the reason for the plant instability and what could have been done to reduce the total number of alarms. While creating the HDAP by default a bin size of 10 minutes is used for all the tags, i.e., number of alarms in a 10 minutes time slot is recorded for all the tags. The user can choose any bin size from the default size of 10 to plot the HDAP. Initially all the tags present in the alarm data are used to plot the HDAP. Also the HDAP is designed in such a way that the tags are sorted according to the number of alarms. User of tool has the option to change the number of tags to be displayed. Under any condition the top bad actors are shown in the HDAP. Color coding is used to plot the HDAP to easily understand the number of alarms at a certain time. This figure is called high density alarm plot due to the reason that it contains the alarm information of all the tags irrespective of duration of data.

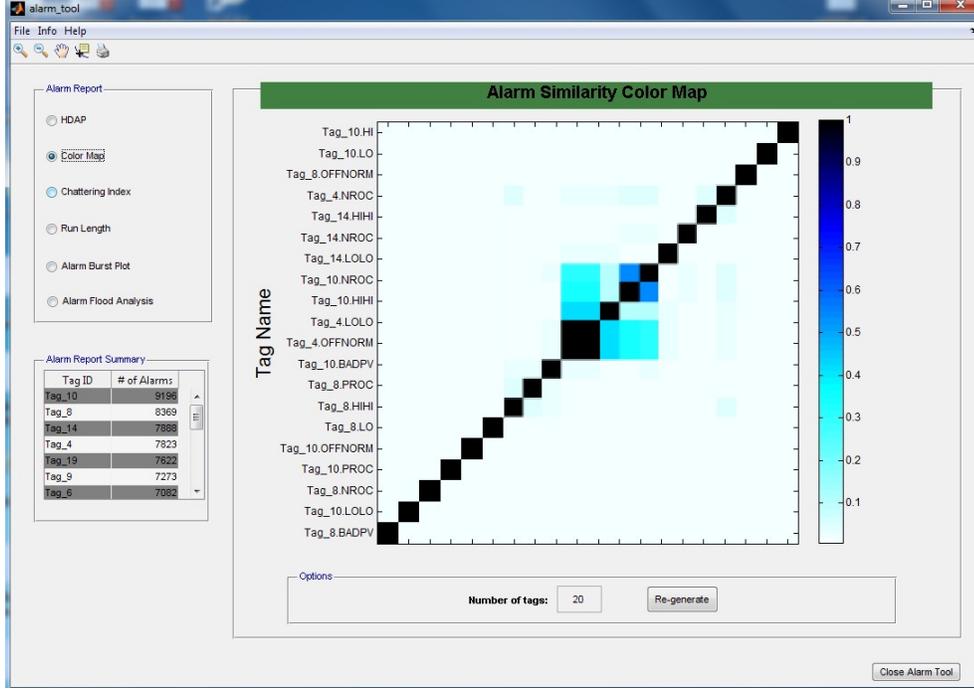


Figure 4.3: Alarm similarity color map shown in the tool

4.2 Alarm Similarity Color Map

Alarm Similarity Color Map (ASCM) shows the correlated alarms in terms of a colormap. Generally alarms are represented in the form of binary data, i.e., when there is an alarm it is represented as “1” and “0” is used to represent that there is no alarm. While calculating ASCM tag ids are combined with tag names to provide better understanding of the system, e.g., which tag id of one tag is triggered related to the id of another tag. In the very beginning all the tags are enriched with extra 1’s. This is done to take into account the possible time lag between tags. Whenever there is an alarm five 1’s are added on both sides, totalling eleven 1’s corresponding to one alarm [10]. To quantify similar alarms, the Jaccard similarity coefficient is used [35]. For two alarm sequences of length N , $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_N)$ the Jaccard similarity coefficient is defined as:

$$S_{jac}(X, Y) = \max_{l \in L} \left(\frac{a(l)}{a(l) + b(l) + c(l)} \right) \quad (4.1)$$

where $a(l)$ is the number of matches $(x_i = 1, y_i = 1) \forall i$

$b(l)$ is the number of mismatches $(x_i = 1, y_i = 0) \forall i$

$c(l)$ is the number of mismatches $(x_i = 0, y_i = 1) \forall i$

l is the time lag between the sequences and $L = \{-240, \dots, 0, \dots, 240\}$.

$S_{jac}(X, Y)$ lies between 0 and 1 [35]; and the higher the value, the more the two tags are correlated. After pairwise calculation of the similarity coefficient the tags are arranged and clustered together for the purpose of better visualization. Like HDAP, ASCM is also color coded to show the value of the coefficient.

When the user clicks on the radiobutton **Color Map**, ASCM for the loaded data is calculated and results are shown in the tool in terms of colormap and colorbar, as can be seen in Figure 4.3. The darker the color, the higher the correlation index. Every tag is highly correlated with itself and that is why along the diagonal it is always 1. As mentioned earlier tag id along with tag names are also considered for the calculation of ASCM and it is evident in Figure 4.3. It can be inferred from Figure 4.3 that *Tag-4.LOLO* and *Tag-4.OFFNORM* are highly correlated with each other. Also *Tag-14.LOLO* and *Tag-10.NROC* are correlated with each other. Moreover these 4 tags are correlated with *Tag-10.HIHI* since they form a large cluster in the ASCM. The user of the tool has the option to change the number of tags at the bottom on which analysis needs to be performed. In that case the top tags are considered for analysis and the number is given by the user.

4.3 Run Length and Run Length Distribution

As mentioned earlier alarm data can be represented by binary data in the form of 1's and 0's. In the context of alarm data *Run* is defined as the sequence of a 1 followed by uninterrupted 0's before another 1 comes. The length of the sequence is defined as the *Run Length* [11]. Therefore run length can be viewed as the time difference in seconds between two consecutive alarms of the same tag. Operator action is ignored over here. 3 kinds of run lengths are implemented in the tool: On to On, On to Off, and Off to On. *On to On* considers the time difference of two consecutive alarm rise times. Time taken to clear the alarm or time difference between alarm rise time and return to normal time (RTN) is considered as *On to Off*. Finally the time taken for the tag to raise another alarm after clearing the previous one is considered as the *Off to On*. An example is used to illustrate the above mentioned definitions.

Table 4.1: Run Length for tag *Tag_1*

Sl. No.	Alarm Rise Time	RTN Time	Time Dif- ference (Run Length, <i>r</i>) On to On	Time Dif- ference (Run Length, <i>r</i>) On to Off	Time Dif- ference (Run Length, <i>r</i>) Off to On
1	07/06/2012 8:40:00	07/06/2012 8:40:05	-	5	-
2	07/06/2012 8:40:10	07/06/2012 8:40:17	10	7	5
3	07/06/2012 8:40:25	07/06/2012 8:40:28	15	3	8
4	07/06/2012 8:40:35	07/06/2012 8:40:45	10	10	7
5	07/06/2012 8:40:50	07/06/2012 8:40:55	15	5	5

Example: For the example, *Tag_1* is considered and its alarm rise time return to normal time (RTN) are shown in Table 4.1. The corresponding *Run Lengths* are given in columns 2, 3, and 4. For this tag the 2nd alarm was raised within 10 seconds after the 1st alarm was raised. It takes from 3 seconds to 10 seconds for the operator to clear the alarm. And after clearing alarm, an alarm is raised within 5 seconds. This is a typical example of the industries today where the operators have to remain busy in responding to alarms of more than thousands of tags a day.

Once the run lengths are calculated, *Run Length Distribution* (RLD) is plotted. RLD is the sum and group of the various run lengths namely, the histogram. Figure 4.4 shows the RLD of the example shown above. Therefore if RLD is right skewed then it can be understood that alarms for one tag is appearing after a certain amount of time. On the other hand if the RLD is left skewed then the operator is not having sufficient time to take action which leads to chattering alarms and consequently an alarm flood.

In the tool when the user clicks on **Run Length** then the RLD of type *On to On* of top 4 bad actors are plotted as shown in Figure 4.5. The user can select any tag from the drop down menu in any of the figures. Moreover the user has the option to choose from different kinds of RLDs. The RLDs which are left skewed require more attention for rationalization and to reduce the number of alarms in the system.

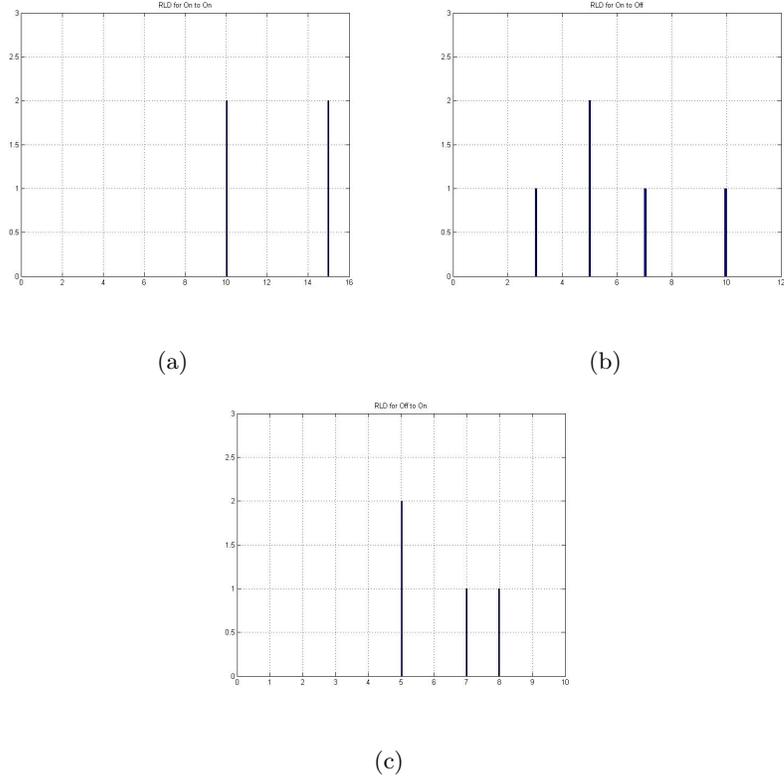


Figure 4.4: Run Length Distribution of *Tag_1*: a) On to On, b) On to Off, c) Off to On

4.4 Chattering Index

Figure 4.6(a) shows the RLD of a non-chattering tag where the RLD is not left-skewed, meaning that the operator gets alarm for after a certain amount of time and has sufficient time to respond. On the other hand, Figure 4.6(b) shows the RLD of a chattering tag where the RLD is highly left-skewed, meaning that the alarm for that tag is repeating within a very short span of time. The *Chattering Index* is calculated based on RLD first by calculating the Probability Density Function (PDF). PDF is obtained by normalizing RLD with a factor of $\sum_{r \in \mathbb{N}} AC_r$ which is 1 less than the total number of alarms. The PDF P_r for any run length r is calculated as follows [11]:

$$P_r = \frac{AC_r}{\sum_{r \in \mathbb{N}} AC_r}, \forall r \in \mathbb{N}, \quad (4.2)$$

where AC_r represents the alarm count for any run length r . A weighting function, inversely proportional to the run length is used to define the *Chattering Index* to penalize the short

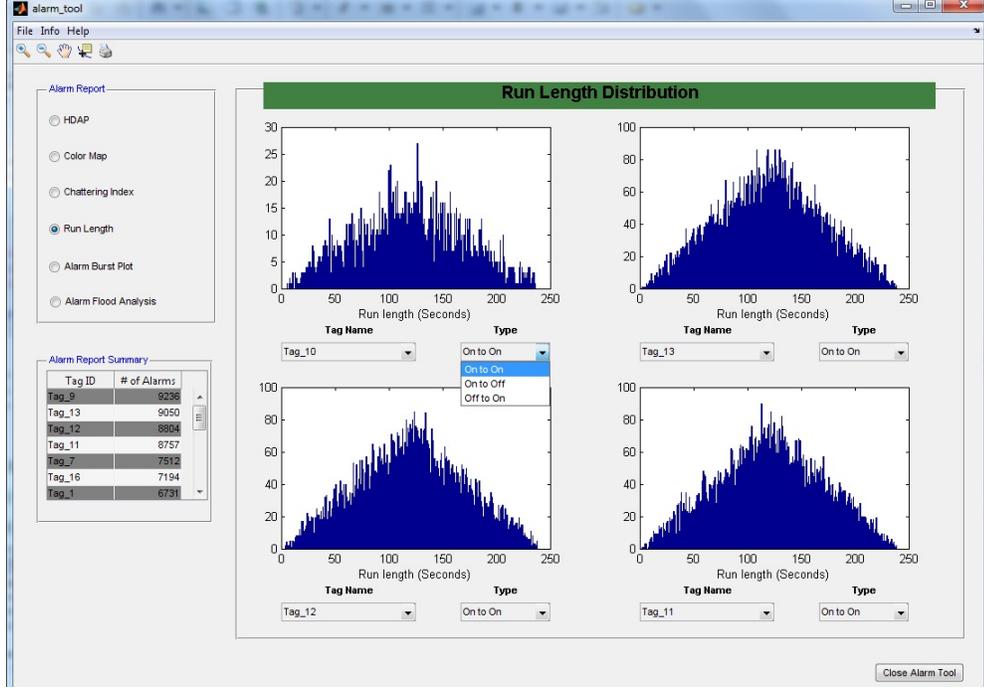


Figure 4.5: Run Length Distribution in the tool with options to select any tag and from different kinds of RLDs

run lengths more than the longer ones. The chattering index, Ψ , is defined as follows:

$$\Psi = \sum_{r \in \mathbb{N}} P_r \frac{1}{r}. \quad (4.3)$$

Ψ lies between 0 and 1. The higher the value, the more the chattering referring to the moment that the tag needs more attention to reduce the number of alarms [11].

When the user clicks on the radiobutton **Chattering Index** the tool calculates the index for all the tags from the RLDs. Then the indices are sorted to show the user the most chattering tags as shown in Figure 4.7. The user has the option to select the number of tags he/she is interested in at the bottom of the tool. There is an option to truncate the chattering index which is also provided in the tool. If the user selects to truncate then for calculation of the chattering index, run lengths beyond 600 seconds are ignored for all the tags. This is also known as modified chattering index [11].

4.5 Alarm Burst Plot

Alarm burst plot shows the number of alarms per 10 minutes time slot in the system for the whole duration of the alarm data. Using the burst plot, one can easily show the instants

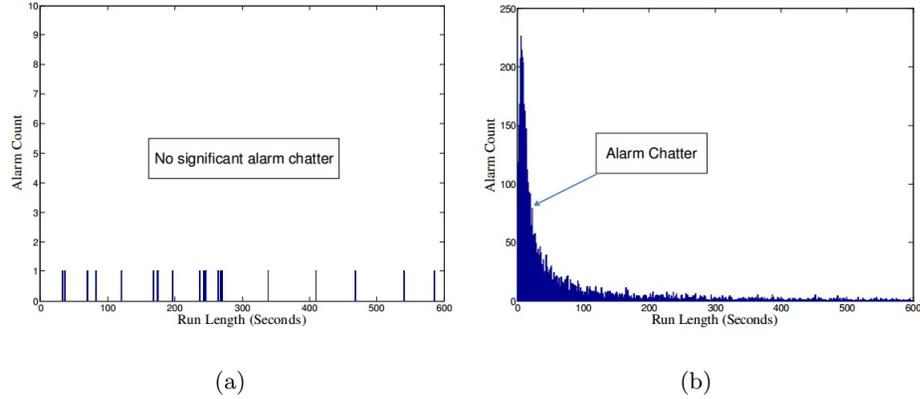


Figure 4.6: Run Length Distributions for: a) Non-chattering tag, b) Chattering tag of alarm floods in the system. Alarm flood is defined according to International Society of Automation (ISA) 18.2 as [24]:

“A condition during which the alarm rate is greater than the operator can effectively manage (e.g., more than 10 alarms per 10 minutes).”

According to the above definition, when the number of alarms in the system crosses 10 alarms per 10 minutes, it is considered as the beginning of an alarm flood. When the number of alarms goes below 5 alarms per 10 minutes, it is considered as the end of the alarm flood. Also, the chattering alarms as discussed in Section 4.4 are removed while considering alarm floods. If there are a number of alarms for one tag within a short period time, then they are considered as one single alarm for that duration to remove the effect of chattering alarm in detecting alarm floods in the system. The idea of alarm burst plot to represent alarm floods was introduced in [36]. The removal of chattering alarms reduces the total number of alarm count in the system and eases the further alarm analysis [12]. Bad alarm setting to response these phenomenon is the reason for chattering alarms.

In the “*alarm_tool*”, when the user clicks on **Alarm Burst Plot** of the main interface, weekly alarm burst plots are shown in the tool. Figure 4.8 shows the alarm burst plot of a real industrial alarm data where the *dotted* lines represent the chattering alarms. As discussed, while considering the detection of alarm floods, the chattering alarms are removed. The instants of alarm floods are shown using the *vertical green line*. The user of the tool can obtain a detailed timing information on the beginning of alarm floods and can look

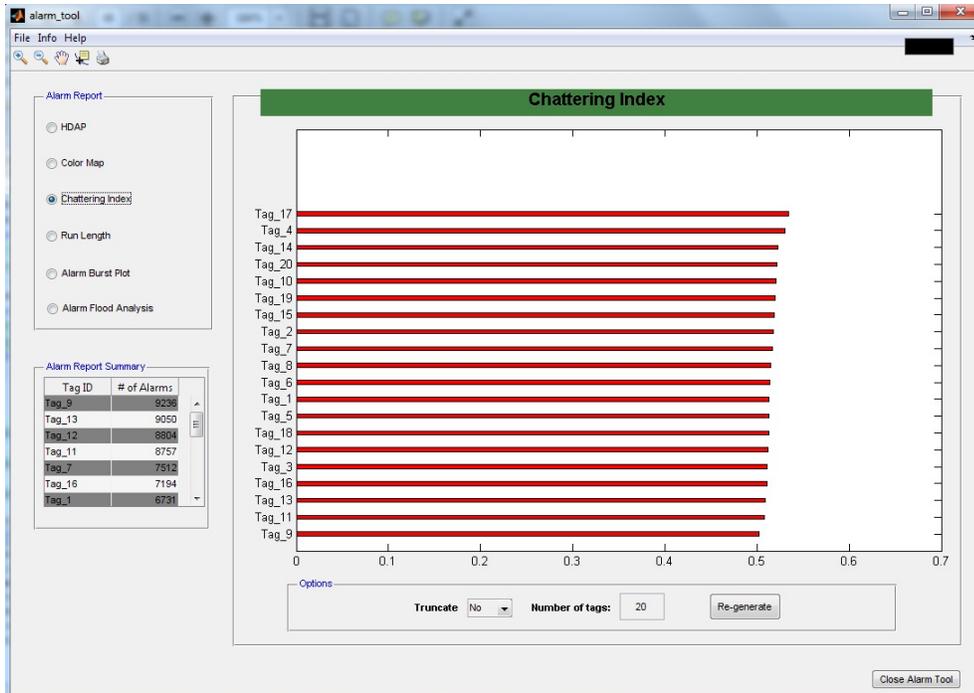


Figure 4.7: Sorted chattering index of the tags

further to identify the cause of the flood to take preventive measures.

4.6 Alarm Flood Similarity Analysis

Once alarm floods are identified, it is the right time to do flood similarity analysis. Typically, an alarm flood consists of a number of alarms. By doing the flood similarity analysis, one can see similar alarm sequences in alarm floods. After proper analysis the user can identify the fault propagation pathway as well as the root cause of fault. When the user clicks on the radiobutton **Alarm Flood Analysis**, the tool displays similar alarm floods in terms of a colormap and colorbar. For the similar flood analysis, Dynamic Time Warpring (DTW) is used. DTW is a nonlinear time alignment method used in aligning time dependent sequences [35], [12], [37].

Figure 4.9 shows similar flood analysis of a real industrial alarm data. Similar flood sequences are clustered together for better visualization. As can be seen in Figure 4.9 that flood sequences 9, 21 and 52 are similar. The tags present in the flood sequences are given in Table 4.2. The real tag names are masked because of confidentiality. It is evident from Table 4.2 that flood sequences 9 and 21 are identical. Sequence 52 follows them with a

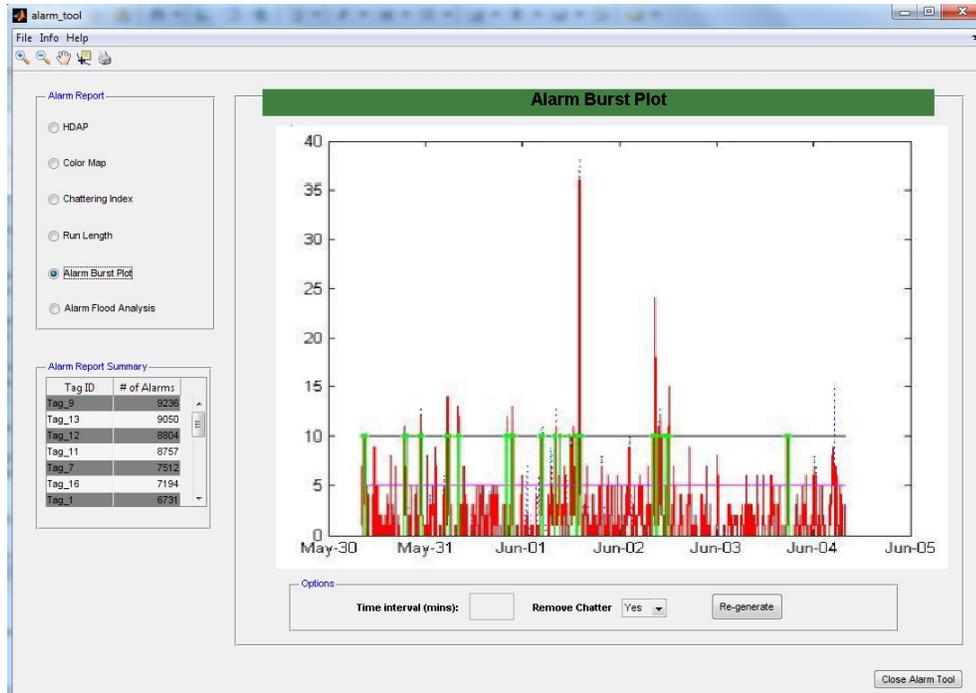


Figure 4.8: Alarm burst plot of real industrial data showing the instants of alarm flood after removal of chattering alarms

Table 4.2: Similar flood sequences of real industrial data

Sequence 9	Sequence 21	Sequence 52
TAG_1.HI	TAG_1.HI	TAG_4.LO
TAG_1.HIHI	TAG_1.HIHI	TAG_1.HI
TAG_1.HIHIHI	TAG_1.HIHIHI	TAG_1.HIHI
TAG_2.HI	TAG_2.HI	TAG_1.HIHIHI
TAG_2.HIHI	TAG_2.HIHI	TAG_4.LO
TAG_2.HIHIHI	TAG_2.HIHIHI	TAG_2.HI
TAG_3.HI	TAG_3.HI	TAG_2.HIHIHI
TAG_3.HIHI	TAG_3.HIHI	TAG_2.HIHIHI
TAG_3.HIHIHI	TAG_3.HIHIHI	TAG_4.LO
-	-	TAG_3.HI
-	-	TAG_3.HIHI
-	-	TAG_3.HIHIHI

different tag in between.

Therefore, from the results of similar flood analysis, the user of the tool gets a detailed idea of the floods and also gets information on similar flood sequences. The advantage is that once the user knows the similar patterns from the historic data, it becomes a lot easier

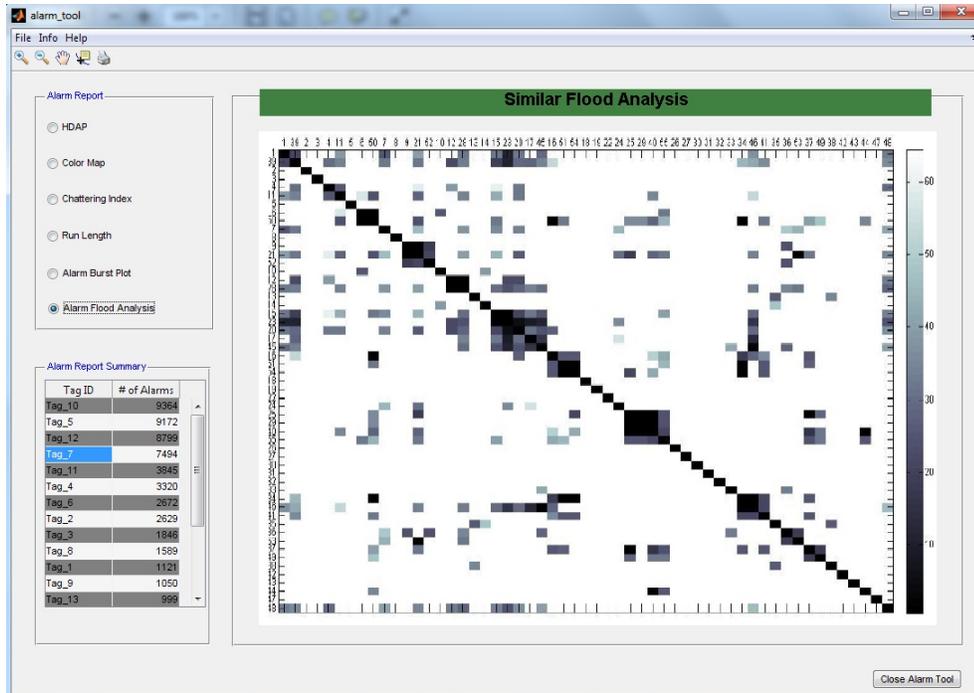


Figure 4.9: Similar flood analysis of real industrial alarm data

to detect the root cause of faults and preventive measures can be taken instantly to reduce the number of alarms in the alarm flood.

4.7 Summary

In this chapter, a tool designed for alarm management is described. By using this tool the user gets a detailed picture of the alarm system performance of the plant. Also he/she determines the time instants when there are significant numbers of alarms and thus can investigate further to identify the cause. The user also collects information on the run lengths of individual tags in the system along with the chattering information. Finally, by doing flood similarity analysis, the user can identify the similar flood sequences, which helps in looking for common root causes of similar floods to improve the alarm system.

Chapter 5

Case Studies

In Chapters 3 and 4, experimental and computer generated data were used to show the utility of the *MDAtool* and *alarm_tool* respectively. In this chapter, results of experimental as well as industrial case studies are presented. *MDAtool* is used for the analysis of process data collected from experiments and industrial operations. On the other hand, *alarm_tool* is used for industrial alarm data analysis. The original tag names of the industrial data are masked due to confidentiality.

5.1 Experimental Case Study

A number of experiments are performed on a 4-tank system. The purpose is to visualize the performance of the *MDAtool* to capture process connectivity and topology. The experimental setup is located at the Computer Process Control Laboratory in the Department of Chemical and Materials Engineering at the University of Alberta. An Emerson Delta-V facility is used to control the system as well as to collect the process data. Figure 3.2 shows the schematic of the system where the lower left and upper right tanks get water from the left pump, while the lower right and upper left tanks receive water from the right pump. Moreover, the lower tanks receive downstreams from the upper tanks. The water levels of all tanks and the water flows from both pumps are considered as the variables of interest in all the experiments performed with this setup. Their roles of being either the controlled or manipulated variables changed based on the nature and requirements of the experiments. It is worth mentioning that, in all the experiments, an enough time is given to the 4-tank system to reach steady state before data collection. A detailed discussion on the experi-

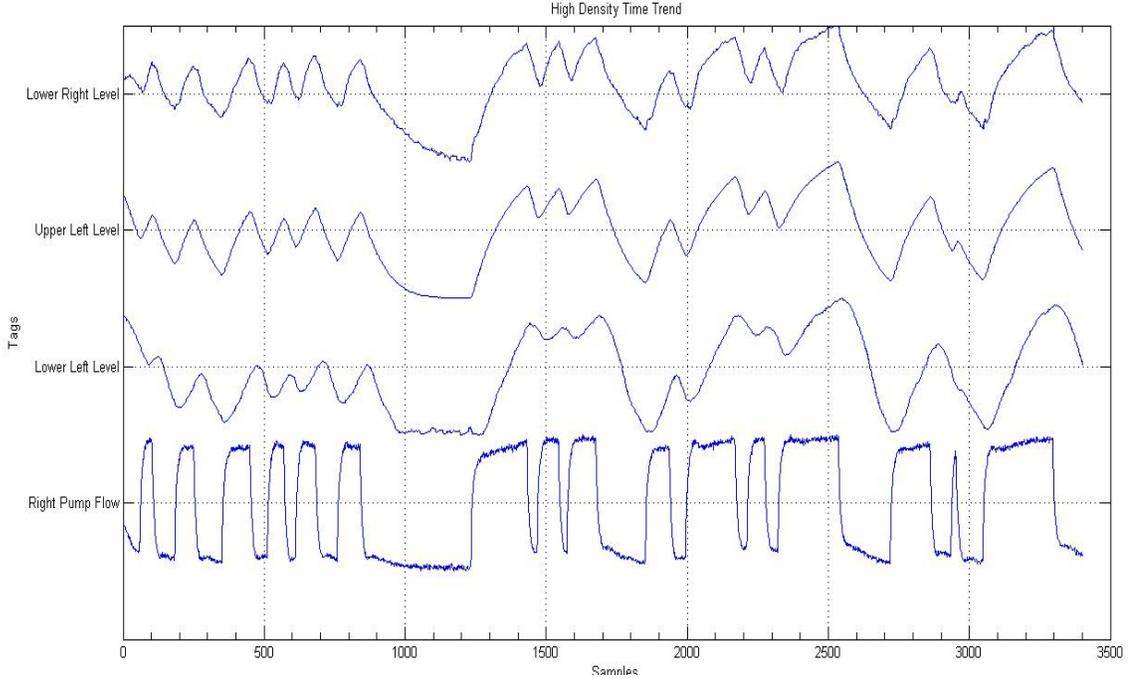


Figure 5.1: High Density Time Trend Plot of different signals obtained in Exp # 1

ments and results is given in the next section. While discussing the results, only relevant variables are considered.

5.1.1 Exp # 1

In Exp # 1, the left pump is turned off and only the right pump is in operation to control the desired level of the lower right tank. In steady state, the lower right tank has a level of 5 cm, the lower left tank has 4 cm, and the upper left tank has 19 cm to maintain a reasonable water level in the lower left tank. The flow rate of the right pump is 22.1 L/min. The setpoint of the lower right tank level is set by a Random Binary Sequence (RBS) of $5 \text{ cm} \pm 2.5 \text{ cm}$. The control strategy is to maintain the desired level of the lower right tank using the right pump only. Thus the system is running under closed-loop operation. The water level on the lower right tank is the controlled variable, and the water flow from the right pump is the manipulated variable. Figure 5.1 shows the High Density Time Trend Plot of the experiment. A total of 3400 sample values of all the variables are recorded. Figure 5.2 shows the real values of water levels of the three related tanks during the experiment. During the experiment, the upper left tank is always given a reasonable amount of water

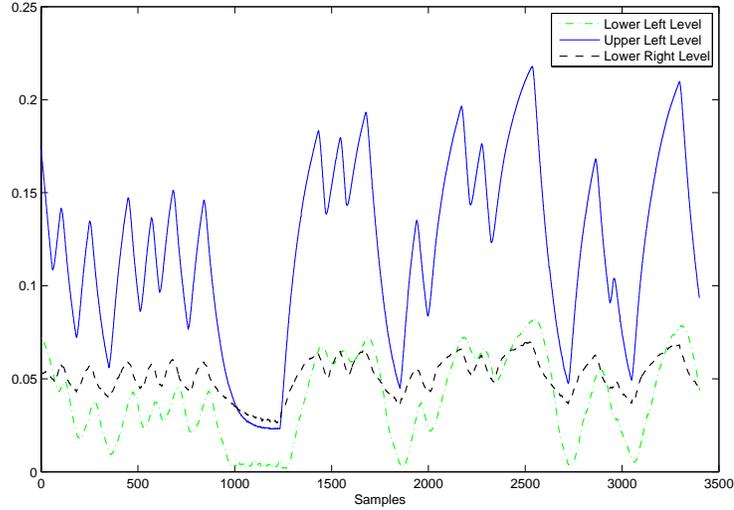


Figure 5.2: Water levels in meters of the three related tanks in Exp # 1

flow from the pump so that the lower left tank has a significant amount of water level, as can be seen in Figure 5.2.

From Figure 5.3 it is evident that the variables in the system are highly correlated with each other. The correlation coefficients are very high and positive among all the variables. Figure 5.4(a) shows the colormap of transfer entropies calculated among the variables of the system. Figure 5.4(b) shows the process connectivity obtained from transfer entropy and it is developed from the values found in Figure 5.4(a). The green lines with an arrow represent the direction of causality and the red lines symbolize two-way connectivity or feedback. Also thickness of the lines shows the strength of the connectivity between the related variables. For this particular experiment, the parameter values for transfer entropy method are chosen as $k = 0, l = 2, h = 1$, and $\tau = 1$. Figure 5.5(a) shows the colormap of Granger causality among the variables for Exp # 1 with a probability threshold of 0.01. Figure 5.5(b) shows the connectivity diagram of the system obtained from Granger causality and it is derived from Figure 5.5(a). Frequency domain Granger causality analysis is also performed with 150 Hz as the maximum frequency to be analyzed and the results are shown in Figure 5.6. From Figure 5.6 one can have a picture of which frequencies are propagating from one variable to the other. It is worth mentioning here that Figure 5.5(b) and Figure 5.6 provide the same result with the only difference being the fact that one is in the time domain and the other is in the frequency domain.

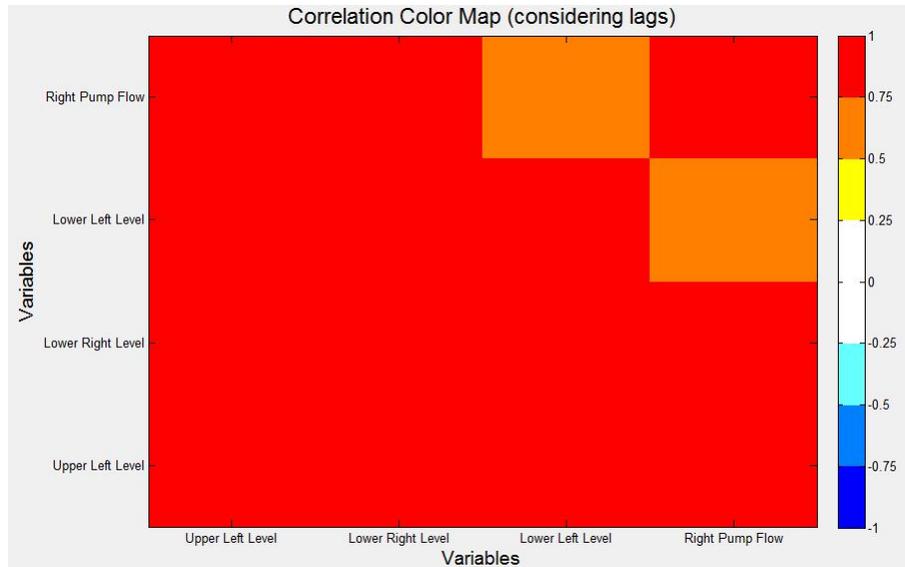


Figure 5.3: Correlation colormap (considering lags) of the related variables of Exp # 1

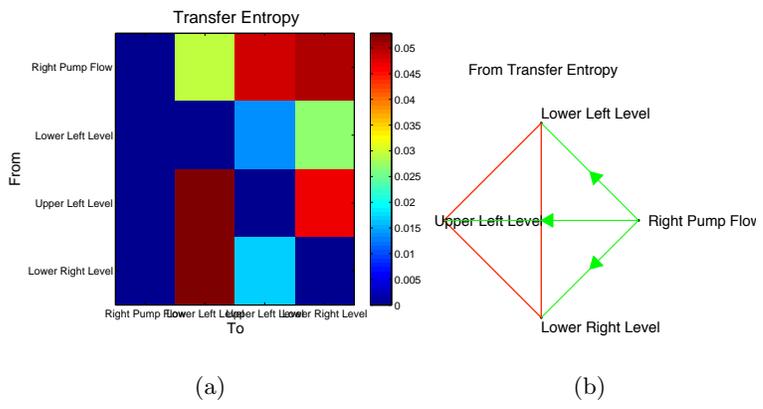


Figure 5.4: Causality results obtained from transfer entropy for Exp # 1: (a) Colormap of transfer entropies among the variables, (b) Corresponding connectivity diagram

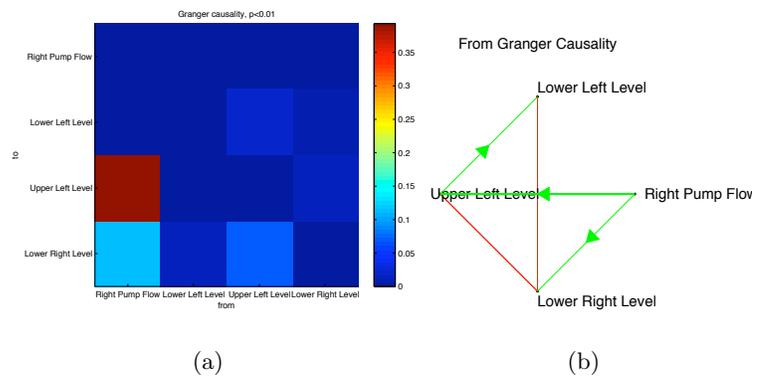


Figure 5.5: Causality results obtained from time domain Granger causality for Exp # 1: (a) Granger causality colormap, (b) Corresponding connectivity diagram

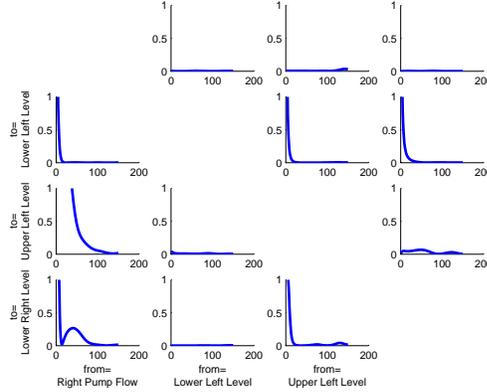


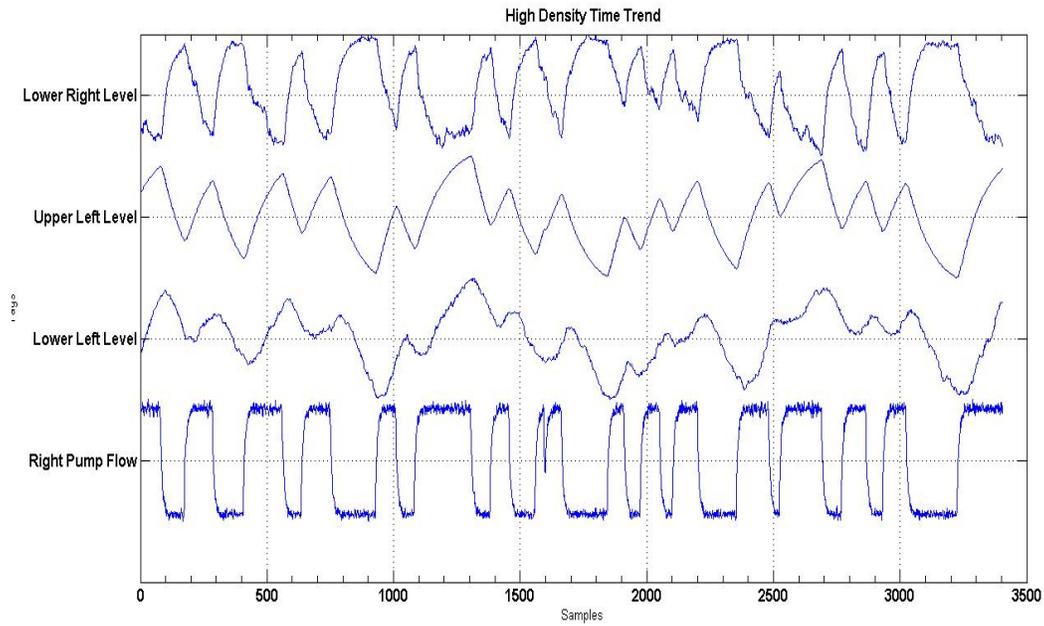
Figure 5.6: Results of frequency domain Granger causality for Exp # 1

The connectivity diagrams for both transfer entropy (Figure 5.4(b)) and Granger causality (Figure 5.5(b)) show that, the right pump affects the level of water in the lower right and upper left tank which is indeed true as can be inferred from Figure 3.2. Because of the control action, however, there should have been a path from the lower right tank to the flow of the right pump, which is absent in both methods. The upper left level has a causal effect on the lower left level, and it is obtained in both cases. In transfer entropy, there is also a path in the reverse direction, which is not true. There is no direct effect of the pump on the lower left tank, yet there is an indirect effect through the upper left tank, which is captured in transfer entropy. Although physically there is no connection between the upper left and lower right tanks, from both methods it is observed that there is a feedback between them. The possible reason might be the fact that they receive flow from the same source and their levels change at the same time. From the above discussion it can be concluded that, both transfer entropy and Granger causality capture the process connectivity very effectively.

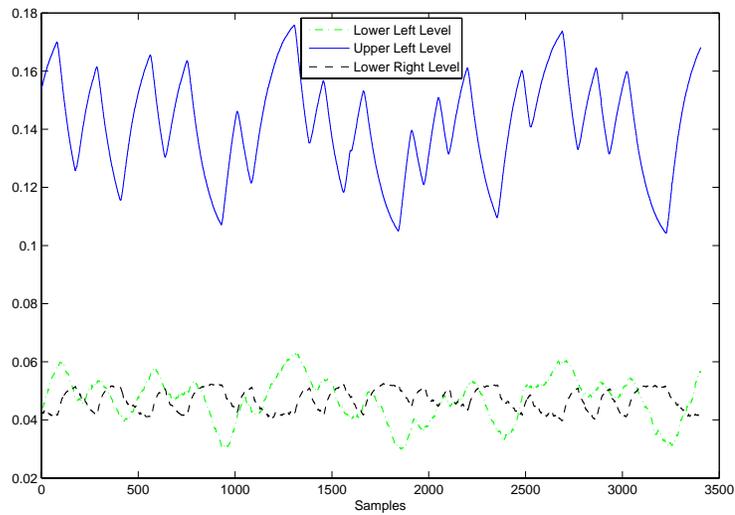
5.1.2 Exp # 2

This experiment is almost the same as Exp # 1 but it is run under open loop, i.e., with the same initial conditions, the output of the right pump is 20 ± 2 L/min and the levels of the lower two tanks and upper left tank are recorded. Figure 5.7(a) shows the high density plot of the variables in the experiment and Figure 5.7(b) shows the levels of the related tanks in meters.

Figure 5.8(a) and Figure 5.8(b) show the connectivity diagrams obtained from transfer



(a)



(b)

Figure 5.7: (a): High Density Time Trend Plot of different tags obtained in Exp # 2, (b): Water levels in meter of the three related tanks of Exp # 2

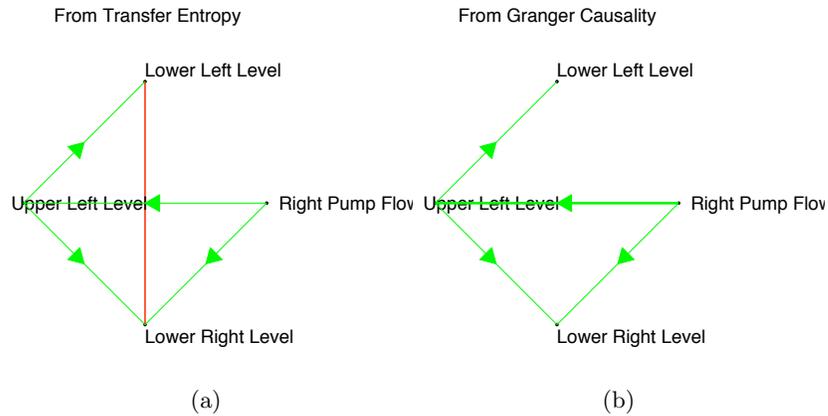


Figure 5.8: Connectivity diagram obtained for Exp # 2 using: (a) transfer entropy, (b) Granger causality

entropy and Granger causality methods respectively. Since there is no control action, there should not be any feedback and Granger causality shows that. The results are same as that of Exp #1 with only one difference that transfer entropy shows a feedback path between the levels of lower left and lower right tanks; the reason for this path is that the levels of the lower two tanks change simultaneously.

5.1.3 Exp # 3

In this experiment, the left pump is turned on with a constant flow of 10 L/min. The system is run under closed-loop to control the Random Binary Sequence (RBS) level of lower right tank by the right pump. The water levels of all the tanks and flow of the two pumps are used to capture the process connectivity. Figure 5.9 shows the connectivity diagrams obtained from transfer entropy and Granger causality respectively. The level of the upper right tank is affected by the left pump which is captured in Granger causality. Yet the effect of the left pump on the lower left tank is not captured in both the methods. The possible reason might be the fact that, more water is coming from the upper left tank to the lower left tank than from the left pump. The flow rates of the pumps are independent of each other and it is captured in both the methods. Although transfer entropy shows a path from the left pump to the upper left tank, physically there is no connection. The remaining paths are the same as Exp # 1 as expected. Thus, both transfer entropy and Granger causality provides satisfactory results to capture the process connectivity.

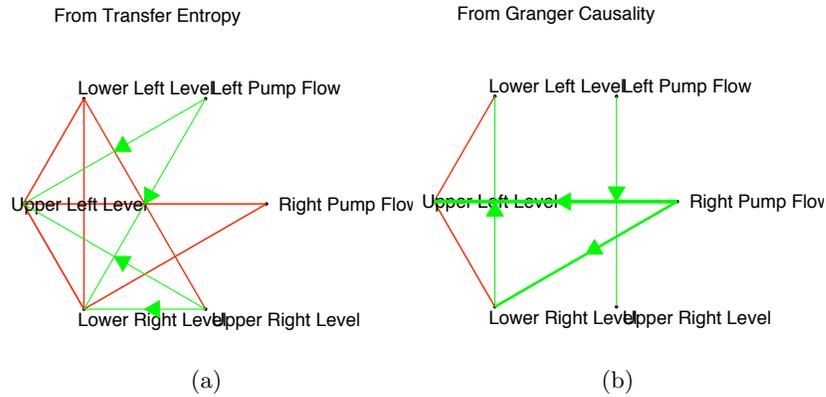


Figure 5.9: Connectivity diagram obtained for Exp # 3 using: (a) transfer entropy, (b) Granger causality

5.1.4 Exp # 4

This experiment is run under closed loop configuration. The control strategy is to control the desired RBS level of both the lower tanks using both the pumps. Initially both the pumps are set to provide 10 L/min flow of water. The RBS set point of the lower left tank is $6.5 \text{ cm} \pm 2.5 \text{ cm}$ and it is $11 \text{ cm} \pm 2.5 \text{ cm}$ for the lower right tank. The experiment is designed to run for 56 minutes and 30 seconds. Figure 5.10(a) shows the high density plot of the data and Figure 5.10(b) shows the actual values of the water levels of all the tanks. From Figure 5.10 it is evident that, the system is now highly oscillating because of the control strategy and the system is not getting enough time to settle down during the whole experiment. The connectivity diagrams obtained from transfer entropy and the one from Granger causality are shown in Figure 5.11. As can be seen from Figure 5.11(a), transfer entropy provides very good result. Both pumps affect all the four tanks and it is expected as levels are maintained from both pumps. But the feedback from both lower levels to both pumps are missing and they should have been over there as the system is run under closed loop operation. The pumps affect each other because of the control strategy as the flow from them are changing to maintain the water level of both the lower tanks. The lower right level cannot affect the upper right level and it is indeed true. And the presence of feedback among the other levels is due to the reason that the levels are changing at the same time. On the other hand Granger causality can detect only one path as can be seen

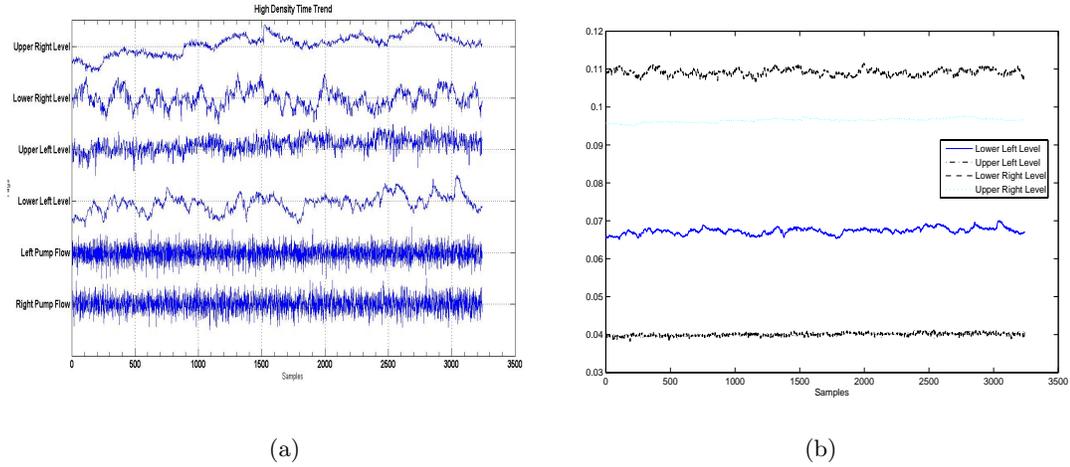


Figure 5.10: (a): High Density Time Trend Plot of different tags obtained in Exp # 4, (b): Water levels in meters of the three related tanks of Exp # 4

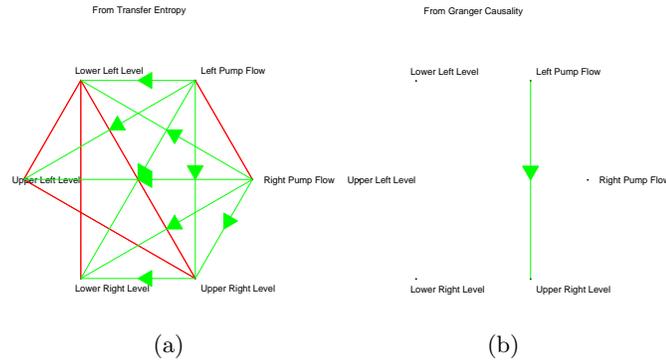


Figure 5.11: Connectivity diagrams obtained for Exp # 4 using: (a) transfer entropy, (b) Granger causality

in Figure 5.11(b) and it is from the left pump to the upper right levels which is indeed true. The reason is because the system is oscillating very fast and it is not getting enough time to settle down. Therefore, the AR model cannot represent the system effectively.

5.2 Industrial Case Study

The industrial case study is performed on an intrastate pipeline system. Weekly alarm data of one year is collected and 4 weeks data are combined to create monthly data. Top bad actors are identified and process data of those tags of almost 13 months is collected. Monthly analysis is performed for causality analysis as well. Analysis results of one month

Table 5.1: Summary of alarm system of a pipeline industry

Month	No. of alarms	Alarms / hr	HI	HIHI	HIHIHI	LO	LOLO	LOLOLO	No. of tags
1	8847	13.14	2553	1698	1394	1665	593	245	297
2	8425	12.54	2405	1598	1278	1581	571	222	266
3	8962	13.33	2612	1735	1428	1516	647	253	258
4	8677	12.91	2649	1380	1008	1740	624	202	319
5	7094	10.55	2187	1114	817	1375	514	164	293
6	3170	4.72	518	179	124	544	240	241	462
7	7117	14.27	1639	925	678	2002	508	358	286
8	7381	8.78	2153	1232	943	1656	481	291	275
9	7089	10.56	1825	1145	880	1834	464	229	275
10	8606	12.8	2668	1422	1132	2025	642	255	254
11	9402	13.99	3149	1553	1158	1869	600	423	272
12	9769	14.54	3553	1732	1337	1702	531	318	245
13	6045	12	2164	1023	816	1118	306	177	221
Total	100584	11.85	30075	12993	12993	20627	6721	3378	-

for both process data and alarm data is discussed below.

5.2.1 Alarm Management

Table 5.1 shows the total number of alarms along with the number of tags in the system for one year. On a monthly basis, the number of tags ranged from 221 to 462, and the number of alarms ranged from 3170 to 9769. The number of alarms per hour is from 4.72 to 14.54; and according to Engineering Equipment and Materials Users Association (EEMUA) [9], it should be no more than 6 alarms per hour. In this case, study results of one month alarm data is presented.

Figure 5.12 shows the high density alarm plot of one month alarm data of the system, where each bin contains the number of alarms in a 10-minute time slot. As can be seen in Figure 5.12 that *Loc3.M4ACC.HI* has the highest number of alarms among all the tags combining the identifiers.

Figure 5.13 shows the run length distributions of the top 4 bad actors of the system. Figure 5.14 shows the top chattering indices. Even though *Loc3.M4ACC.HI* has the highest number of alarms, *Loc4.M1GRV.LO* has the highest chattering index. Figure 5.15 shows the

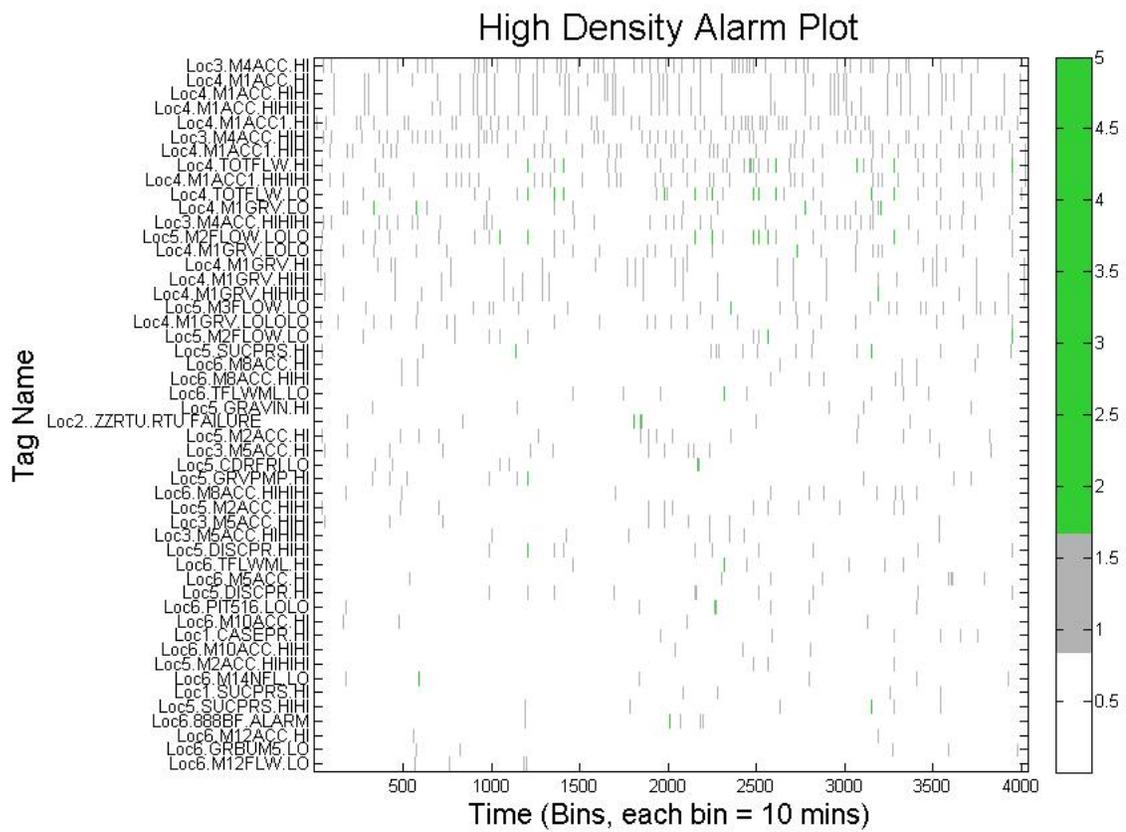


Figure 5.12: High density alarm plot of one month industrial alarm data

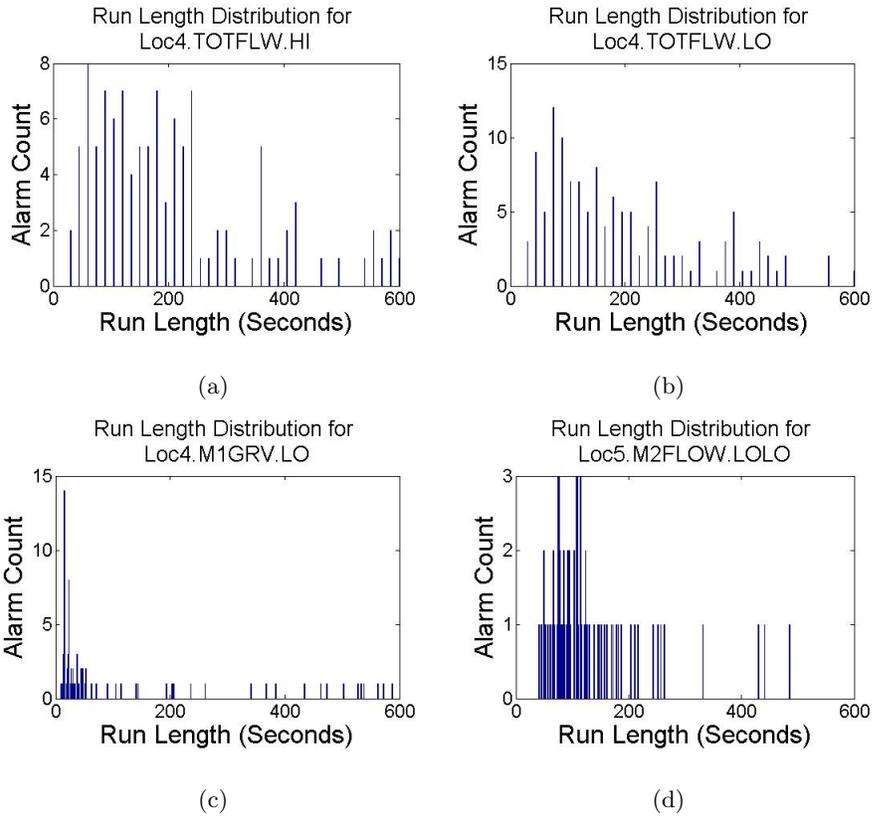


Figure 5.13: Run length distributions of top 4 bad actors in the system

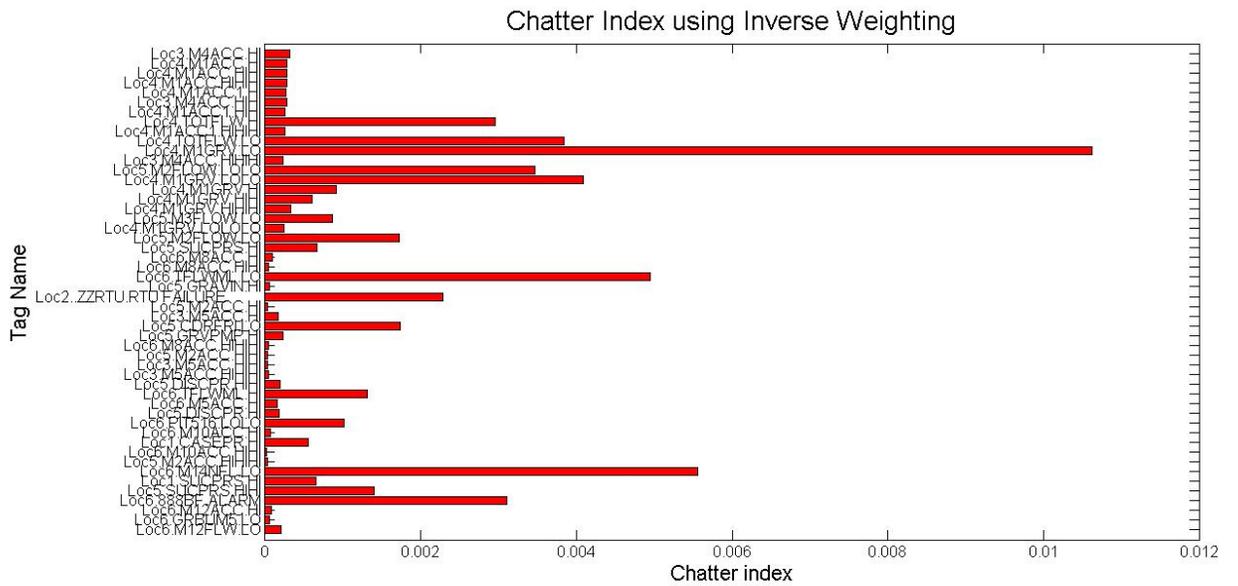


Figure 5.14: Chattering indices of the bad actors of the system

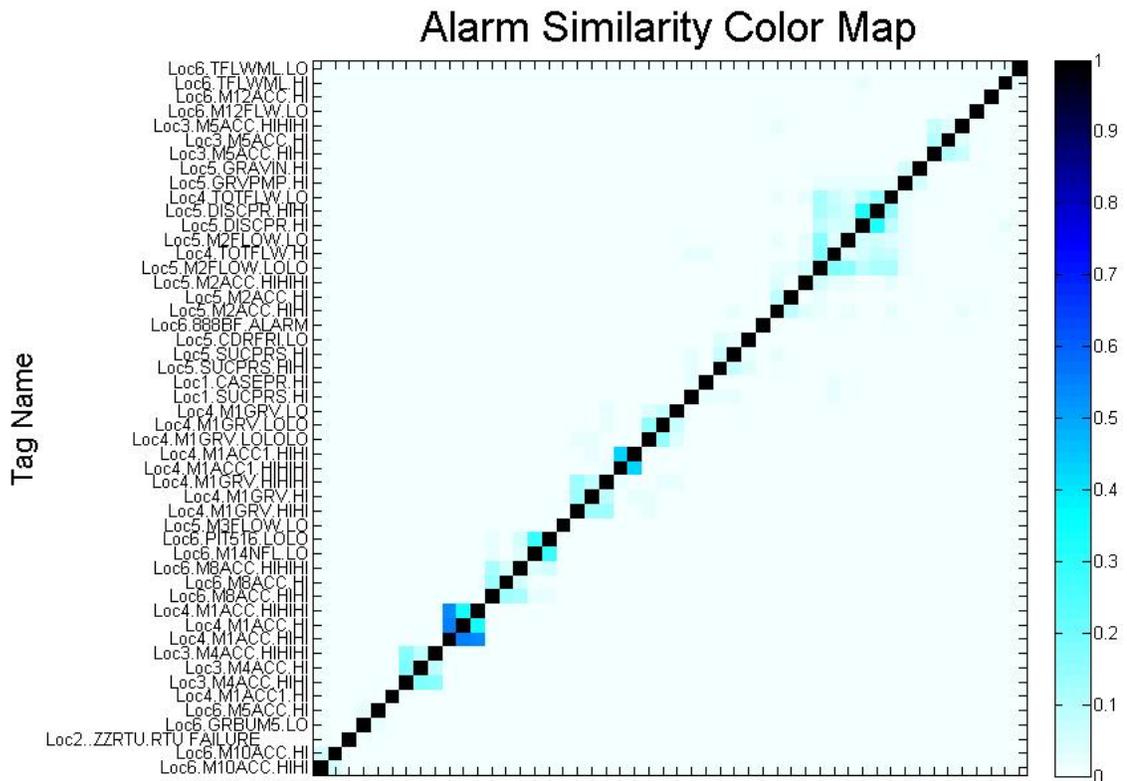


Figure 5.15: Correlation color map of the correlated alarm tags

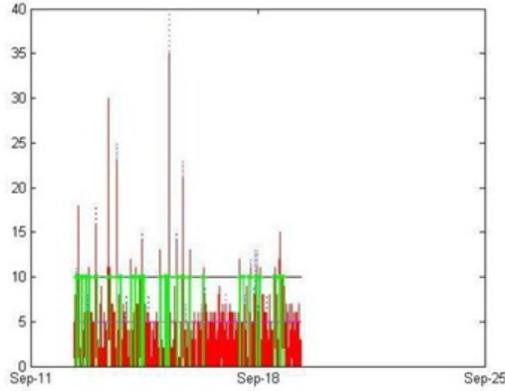


Figure 5.16: Alarm burst plot of one week

correlated tags of the system, where it is evident that *Loc3.M4ACC.HI*, *Loc3.M4ACC.HIHI*, and *Loc3.M4ACC.HIHIHI* are correlated with each other. Also *Loc4.M1ACC.HI*, *Loc4.M1ACC.HIHI*, and *Loc4.M1ACC.HIHIHI* are correlated with each other. It can also be inferred from Figure 5.15 that, tags *M2FLOW.LOLO*, *M2FLOW.LO*, *DISCPR.HI*, *DISCPR.HIHI*, and *GRAVPMP.HI* of *Loc5* are correlated with each other. By observing the data again, it is verified that these correlated tags trigger each other a number of times during the period considered.

After this, alarm flood similarity analysis is performed on the alarm data. While plotting the burst plots, weekly data is used to show the floods distinctively. Figure 5.16 shows the burst plot of one week data where dotted lines represent the chattering alarms and green lines represent the existence of alarm floods in the system. Alarm flood similarity analysis is performed using the dynamic time warping method. Figure 5.17 shows the similar flood sequences in terms of a colormap and colorbar.

Table 5.2 shows two similar flood sequences. Though the lengths of the alarm flood sequences are different, both the sequences contain similar alarms.

All the above mentioned steps are repeated for the rest of the 12 months. All the similar flood sequences were showed in details to the company for all the 13 months alarm data. A report is created and presented to the industrial partner. Based on the analysis of alarm data, process data of important 28 tags are requested. Process data of 25 tags are collected and causality analysis to find out the direction of propagation of fault is performed using

Table 5.2: An example of similar alarm flood sequences in the industrial case study

Seq 6	Seq 36
Loc1.SUCPRS.HI	Loc3.TOTFLOW.LO
Loc3.M1GRV.LO	Loc3.M1GRV.LO
Loc3.M1GRV.LOLO	Loc3.TOTFLW.LO
Loc3.M1GRV.LOLOLO	Loc1.SUCPRS.HI
Loc3.M1GRV.HIHI	Loc3.M1GRV.LOLO
Loc3.M1GRV.HI	Loc3.M1GRV.LOLOLO
Loc3.M1ACC.HI	Loc3.M1ACC.HI
Loc3.M1ACC.HIHI	Loc3.M1ACC.HIHI
Loc3.M1ACC.HIHIHI	Loc3.M1ACC.HIHIHI
Loc3.M1ACC.HI	Loc3.M1ACC.HI
Loc3.M1ACC.HIHI	Loc3.M1ACC.HIHI
Loc3.M1ACC.HIHIHI	Loc3.M1ACC.HIHIHI
Loc3.M1GRV.HI	Loc3.M1GRV.HI
Loc3.M1GRV.HIHI	Loc3.M1GRV.HIHI
Loc3.M1ACC.HI	Loc3.M1ACC.HI
Loc3.M1ACC.HIHI	Loc3.M1ACC.HIHI
Loc3.M1ACC.HIHIHI	Loc3.M1ACC.HIHIHI
Loc3.M1ACC1.HI	Loc3.M1ACC1.HI
Loc3.M1ACC1.HIHI	Loc3.M1ACC1.HIHI
Loc3.M1ACC1.HIHIHI	Loc3.M1ACC1.HIHIHI
Loc1.SUCPRS.HI	Loc2.M4ACC.HI
Loc3.TOTFLW.HI	Loc2.M4ACC.HIHI
Loc2.M4ACC.HI	Loc2.M4ACC.HIHIHI
Loc2.M4ACC.HIHI	
Loc2.M4ACC.HIHIHI	
Loc3.TOTFLW.LO	

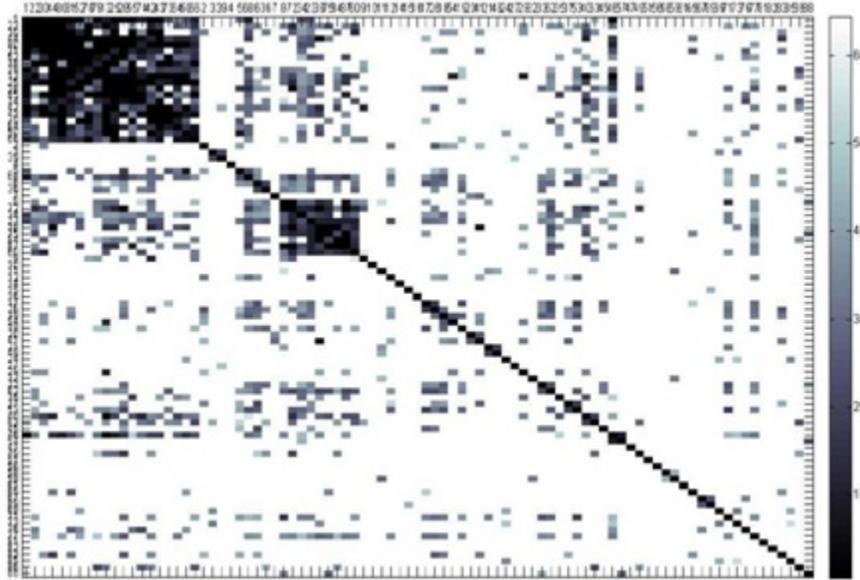


Figure 5.17: Similarity color map of similar floods in one month alarm data

Granger causality.

5.2.2 Causality Analysis

Figure 5.18 shows the captured connectivity of the process, obtained from the process data of one month using Granger causality. The results of causality analysis are validated using the P&ID diagrams of the system. Combining the results of both causality analysis and alarm data analysis, root cause of faults is detected as *Loc3.M1GRV* and it is validated with the system engineers.

5.3 Summary

Both transfer entropy and Granger causality methods are very effective to capture the causality. In terms of computational complexity, calculation of Granger causality is much faster. Moreover, it has also been observed that by changing the parameter values, the process connectivity obtained from transfer entropy changes. Thus, choosing the right parameter for the right process is very important. On the other hand, transfer entropy outperforms Granger causality clearly in Exp # 3 and Exp # 4. When a system is highly oscillating, then model based methods cannot capture the desired process topology; but model free methods can provide very good results for such kind of processes. This is evident

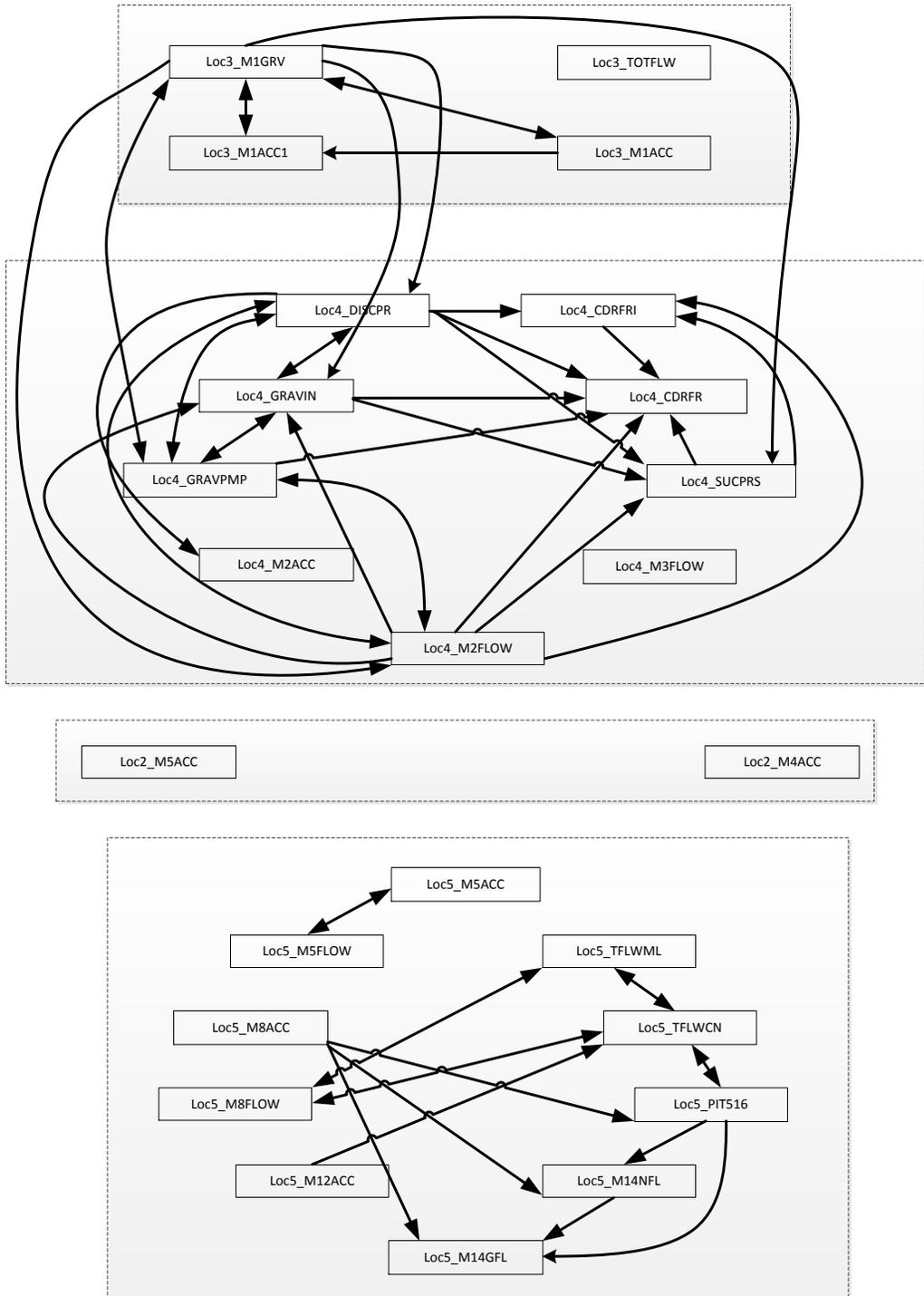


Figure 5.18: Causality analysis results of one month process data

from the results of Exp #4. On the other hand, if the data could be better predicted from the model estimated, then Granger causality outperforms transfer entropy. Therefore, method of analysis depends on the data set available and quality of the data. Industrial case studies show the successful usage of the tools for alarm management, root cause identification, and finding plantwide disturbance propagation pathways.

Chapter 6

Conclusions

6.1 Contributions

Two graphical tools are designed in this thesis. The first tool, *MDAtool* uses historic process data for data visualization and performs causality analysis to capture process connectivity and topology. This tool incorporates three data driven causality detection methods. The second tool, namely, *alarm_tool*, shows the alarms in the loaded alarm data. This tool not only pictures the alarms in the system but also identifies the top bad actors by visualizing the chattering indices. This tool also exhibits the correlated alarm tags and similar alarm flood sequences, which ultimately pinpoints the tags which require further attention. Both tools are successfully used to represent system topology and identification of root causes of faults in the system. Both experimental and industrial case studies are performed using the tools. Obtained results are validated using the system P&ID and knowledge of system engineers. Successful use of the tools decreases the effort required for causality analysis and alarm management. They also can save a lot of time, for the analysis and visual way of representing the results provide an easy way to interpret the results.

6.2 Future Scope

Combining the tools in the future can be proved very effective which will handle both historical alarm data and process data. This will pave the way of using a single graphical tool for alarm management, as well as detecting the system topology and identifying root causes of faults. Also further applications of the tools for industrial alarm rationalization will help the Canadian industry in a significant way. Inclusion of user defined functions

can pave the way for incorporating new methods of analysis in the tool. Also direct and indirect causal relationships are not considered in this thesis. The inclusion of methods for identifying direct and indirect causal pathways can be very effective for root-cause detection of faults in systems.

Bibliography

- [1] F. Yang, S. L. Shah, and D. Xiao, “Signed directed graph based modeling and its validation from process knowledge and process data,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 1, pp. 41–53, 2012.
- [2] M. Bauer and N. F. Thornhill, “A practical method for identifying the propagation path of plant-wide disturbances,” *Journal of Process Control*, vol. 18, no. 7, pp. 707–719, 2008.
- [3] M. Lungarella, K. Ishiguro, Y. Kuniyoshi, and N. Otsu, “Methods for quantifying the causal structure of bivariate time series,” *International Journal of Bifurcation and Chaos*, vol. 17, no. 3, pp. 903–921, 2007.
- [4] J. Chen and R. Patton, *Robust Model-based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [5] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.
- [6] R. Isermann, *Fault Diagnosis Systems: an Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.
- [7] I. Izadi, S. Shah, D. Shook, and T. Chen, “An introduction to alarm analysis and design,” *Proc. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pp. 645–650, 2009.
- [8] I. Izadi, S. Shah, D. Shook, S. Kondaveeti, and T. Chen, “A framework for optimal design of alarm systems,” *Proc. 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pp. 651–656, 2009.

- [9] Engineering Equipment and Materials Users Association (EEMUA), *Alarm systems: a guide to design, management and procurement*, 2007.
- [10] S. Kondaveeti, I. Izadi, S. Shah, and T. Black, “Graphical representation of industrial alarm data,” *In Proc. of the 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Human-Machine Systems*, vol. 11, no. 1, pp. 181–186, 2010.
- [11] S. Kondaveeti, I. Izadi, S. Shah, D. Shook, and R. Kadali, “Quantification of alarm chatter based on run length distributions,” *Proc. of the 49th IEEE Conference on Decision and Control*, pp. 6809–6814, 2010.
- [12] K. Ahmed, I. Izadi, T. Chen, D. Joe, and T. Burton, “Similarity analysis of industrial alarm flood data,” *revised and under review by IEEE Transactions on Automation Science and Engineering*, 2011.
- [13] S. T. Smith, *MATLAB: Advanced GUI Development*. Dog Ear Publishing, 1st Edition, 2006.
- [14] C. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.
- [15] A. Seth and G. Edelman, “Distinguishing causal interactions in neural populations,” *Neural Computation*, vol. 19, no. 4, pp. 910–933, 2007.
- [16] A. Seth, “A Matlab toolbox for Granger causal connectivity analysis,” *Journal of Neuroscience Methods*, vol. 186, pp. 262–273, 2010.
- [17] S. Gigi and A. Tangirala, “Quantitative analysis of directional strengths in jointly stationary linear multivariate processes,” *Biological Cybernetics*, vol. 103, pp. 119–133, 2010.
- [18] M. Iri, K. Aoki, E. O’shima, and H. Matsuyama, “An algorithm for diagnosis of system failures in the chemical process,” *Computers and Chemical Engineering*, vol. 3, no. 1-4, pp. 489–493, 1979.

- [19] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, “Transfer entropy: a model-free measure of effective connectivity for the neurosciences,” *Journal of Computer Neuroscience*, vol. 30, no. 1, pp. 45–67, 2011.
- [20] M. Bauer, J. W. Cox, M. H. Caveness, J. J. Downs, and N. F. Thornhill, “Finding the direction of disturbance propagation in a chemical process using transfer entropy,” *IEEE Transaction on Control Systems Technology*, vol. 15, no. 1, pp. 12–21, 2007.
- [21] M. Bauer, N. F. Thornhill, and A. Meaburn, “Specifying the directionality of fault propagation paths using transfer entropy,” *7th International Symposium on Dynamics and Control of Process Systems*, no. 62, pp. 1–6, 2004.
- [22] M. Wibral, B. Rahm, M. Rieder, M. Lindner, R. Vicente, and J. Kaiser, “Transfer entropy in magnetoencephalographic data: Quantifying information flow in cortical and cerebellar networks,” *Progress in Biophysics and Molecular Biology*, vol. 105, no. 1, pp. 80–97, 2011.
- [23] “Abnormal Situations Management Consortium.” <http://www.asmconsortium.com>.
- [24] The International Society of Automation (ISA), *ANSI/ISA-18.2-2009, Management of alarm systems for the process industries*, 2009.
- [25] T. Schreiber, “Measuring information transfer,” *Physical Review Letters*, vol. 85, no. 2, pp. 461–464, 2000.
- [26] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London; New York, 1986.
- [27] Q. Li and J. Racine, *Nonparametric Econometrics: Theory and Practice*. Princeton University Press, 2007.
- [28] H. Akaike, “A newlook at the statistical model identification,” *IEEE Transaction on Automatic Control*, vol. 19, pp. 716–723, 1974.
- [29] G. Schwartz, “Estimating the dimension of a model,” *Annals of Statistics*, vol. 5, no. 2, pp. 461–464, 1978.

- [30] V. Barnett and T. Lewis, *Outliers in Statistical Data*. John Wiley & Sons., 1994.
- [31] F. Grubbs, “Procedures for detecting outlying observations in samples,” *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.
- [32] E. J. Wegman, “Hyperdimensional data analysis using parallel coordinates,” *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 664–675, 1990.
- [33] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications*. Springer, 2009.
- [34] R. Brooks, J. Wilson, and R. Thorpe, “Geometry unifies process control, production control and alarm management,” *IEE Computing & Control Engineering*, vol. 15, no. 1, pp. 22–27, 2004.
- [35] M. Lesot, M. Rifqi, and H. benhadda, “Similarity measures for binary and numerical data: a survey,” *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 1, no. 1, pp. 63–84, 2009.
- [36] B. R. Hollifield and E. Habibi, *Alarm Management - Seven Effective Methods for Optimum Performance*. ISA, 2007.
- [37] R. Bellman, *Dynamic Programming*. Dover Publications, 2003.