Aalto University
School of Chemical
Technology

School of Chemical Technology
Degree Programme of Chemical Technology

Lanre Balogun

VISUALIZATION OF FAULT PROPAGATION PATHS IN PROCESS MONITORING SYSTEMS

Master's thesis for the degree of Master of Science in Technology submitted for inspection, Espoo, 07 January, 2015.

Supervisor **Professor Sirkka-Liisa Jämsä-Jounela**

Instructor **M.Sc. Jukka Kortela**

Aalto University
School of Chemical
Technology

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Lanre Balogun

Title of thesis Visualization of fault propagation paths in process monitoring systems

Department Biotechnology and Chemical Technology

Professorship Process Control **Code of professorship** KE-90

Thesis supervisor Professor Sirkka-Liisa Jämsä-Jounela

Thesis advisor(s) / Thesis examiner(s) M.Sc. Jukka Kortela

Date 07.01.2015 **Number of pages** 83+13 **Language** English

Abstract

Modern industrial processes typically have complex interconnections among process equipment and units. A fault can easily propagate far beyond its origin, making it difficult to locate the root cause and to identify its propagation paths. Thus, the latest approach for identifying fault propagation paths uses qualitative models, e.g., topology-based models, for improving the results acquired from the data-driven methods of causal analysis. However, the current approach includes no display of the fault propagation paths in process monitoring systems, which could enable the visualisation of the spread of fault effects.

In this thesis, the aim is to develop and to test a new technique which enables a process monitoring system to display fault propagation paths, based on the causality matrix obtained from a process. A causality matrix is created using the Granger causality method, which contains the details of the fault propagation path. Subsequently, the causality matrix is refined using a connectivity matrix, which was obtained from a piping and instrumentation diagram (P&ID). Furthermore, the process monitoring system and the refined causality matrix are linked together, to transfer the fault propagation paths details to the process monitoring interface. A number of algorithms in the automation system are developed and implemented to accomplish the link.

A flotation pilot process is the case study in this thesis, used for testing this technique. The experiment data acquired from the flotation pilot was analysed to generate a causality matrix, which is subsequently refined and stored in the database of the automation system. Hence, the fault propagation paths is visualised in the process monitoring system. Finally, this procedure confirmed that the propagation of the effects of a fault is better understood if the propagation path is visualised in the process monitoring system.

Keywords Fault propagation, Flotation process, Causal analysis, Causality matrix, Electronic P&ID, Connectivity matrix, ABB System 800xA

ACKNOWLEDGEMENTS

This thesis is written at the laboratory of the Process Control and Automation Research Group at Aalto University School of Chemical Technology. The hospitality of the group members during my working time in year 2014 is highly appreciated.

First of all, I would like to express my deep gratitude to my supervisor, Professor Sirkka-Liisa Jämsä-Jounela, for providing me the opportunity to study in Aalto University and for offering me this interesting thesis opportunity. The high expectations from her and her endless encouragement are a source of my motivation. My sincere appreciation also goes to my instructor, M.Sc. Jukka Kortela, for giving me sense of direction when needed, despite working on his P.hD. thesis in the same period. The contributions and feedbacks from him and my supervisor are very valuable indeed.

Furthermore, I would like to give special thanks to D.Sc. Alexey Zakharov, for the very valuable feedback and guidance I received from him. Special thanks also goes to M.Sc. Rinat Landman, for her valuable contribution and advice during this thesis. I am also grateful to all other members of the Research Group for their kindness and friendliness during my time there: Sasha, Palash, Octavio, Miao, Tushar, Aleksi, Moses and Jerri thanks to all of you.

Lastly, my appreciation goes to my dear family and friends, they never stopped believing in me. I would also like to express a deep appreciation to my late mother, she is responsible for any success I achieve. I thank you very much, the almighty God, you are the one I always look up to.

Espoo,

January 2015

Lanre Balogun

Table of Contents

1 INTRODUCTION....1

LITERATURE PART

2 FAULT PROPAGATION AND CAUSAL ANALYSIS....4

2.1 Data-driven causal analysis methods....5

2.1.1 Time-domain Granger Causality....7

2.1.2 Cross-correlation analysis....10

2.1.3 Transfer entropy....13

2.2 Topology-based causal models....17

2.2.1 Signed directed graphs....18

2.2.2 Electronic P&ID design in AutoCAD....21

2.2.2.1 Extraction of the process topology data from the AutoCAD P&ID....23

2.2.2.2 Connectivity matrix generated from the process topology data....24

3 APPLICATIONS OF FAULT DIAGNOSIS AND FAULT PROPAGATION ANALYSIS....26

3.1 Industrial condition monitoring and fault diagnosis applications....27

3.1.1 Siemens asset management and condition monitoring system....27

3.1.2 ELMAS logical relations software....30

3.2 Academic fault diagnosis and causal analysis applications....33

3.2.1 The Cause-and-Effect Analyzer tool....33

3.2.1.1 Search algorithm for verifying fault propagation paths....35

EXPERIMENTAL PART

4 AIM OF THE EXPERIMENTAL PART....38

5 FLOTATION CELL PROCESS AND ITS CAUSAL MODEL....41

5.1 Description of the flotation process....41

5.1.1 Flotation cell pilot plant in the Aalto University laboratory....46

5.1.1.1 Control strategy of the flotation pilot plant....47

5.2 Creation of a causal model from the process flowsheet....49

5.2.2 Representation of the flotation pilot process in AutoCAD P&ID....50

5.2.2.1 Conversion of the process topology data to a connectivity matrix....54

6 PROCESS MONITORING SYSTEM AND DATA ANALYSIS....57

6.1 Identification of a fault propagation path with the causal analysis....57

6.2 Process monitoring systems and data transfer scheme....60

6.2.1 Design of the process monitoring system....61
6.2.2 Data transfer to the system database and to the elements of the process graphics....65
6.2.2.1 Softpoint objects in System 800xA for data representation....65
6.2.2.2 Calculation aspects in System 800xA for data transfer....68
6.2.2.3 Data access from softpoints to the process graphics elements....72
6.3 Results....76
7 CONCLUSION....78
REFERENCES....79

LIST OF APPENDICES

Appendix A: Electronic P&ID drawing of the flotation pilot process

Appendix B: Monitoring interface for the flotation pilot process

Appendix C: Operator trend display for the flotation pilot process

Appendix D: Faceplates for controlling the flotation pilot process

Appendix E: Algorithm for accessing data from the database

Appendix F: Softpoint objects for storing the causality matrix

Appendix G: Design of the monitoring interface in System 800xA

Appendix H: Data transfer algorithms in VBScript language

Appendix I: Alarm and Event list for fault propagation path analysis

ABBREVIATIONS

AEM	Abnormal Event Management
CAEX	Computer Aided Engineering Exchange
CCF	Cross-Correlation Function
DCS	Distributed Control System
DIN	Deutsches Institut für Normung (German Institute for Standardization)
GC	Granger Causality
ISA	International Society of Automation
ISO	International Organization for Standardization
JIS	Japanese Industrial Standards
OOP	Object-oriented Programming
OPC	OLE for Process Control
PCA	Principal Component Analysis
PDF	Probability Density Function
PIP	Process Industry Practices
PLS	Partial Least Square Regression
P&ID	Piping and Instrumentation Diagram
QTA	Qualitative Trend Analysis
TE	Transfer Entropy
XML	Extensible Markup Language
XLS/XLSX	Microsoft Excel File Extension

1 INTRODUCTION

The tight global competition has forced the process industries to identify cost reduction strategies for operating industrial plants competitively. However, processes are prone to faults and disturbances, causing failure to achieve the optimal targets in their operations. A fault refers to any unpermitted deviation of at least one characteristic property of a process from the acceptable and normal condition. Faults and disturbances have been reported to have considerable negative effects on condition of processes (Horch et al., 2007), since production and throughput are forced to deviate from their optimum settings in order to accommodate "unallowable" variability in the process. Additional consequences of a faulty process system include increased variability of the process variables leading to poor control performance, inferior products and higher rejection rates (Jiang et al., 2009). It is important to identify the root causes of faults and disturbances in industrial processes. Furthermore, early verification of the root cause of a fault and its effects can help to avoid the potential harm and danger arising from abnormal operation in the plant. However, faults and disturbances propagate downstream in industrial processes due to complicated interlinking of equipment, as well as material and energy recycle, and control loop interactions. These factors make identifying the root causes of faults in processes a rather challenging task.

Consequently, several methods have been introduced over the years to enable the identification of root causes of plant-wide faults and disturbances. Thornhill and Horch (2007); Yang and Xiao (2012) presented a collection of data-based methods for analysing causal relationships of process variables, such as granger causality, Bayesian nets, cross-correlation analysis and transfer entropy methods. Jiang et al. (2009) presented a concept known as the Adjacency Matrix – a qualitative model-based approach, aimed at the root cause diagnosis of plant-wide faults; they further described the implementation of this method as well as provided an industrial case work. However, these approaches to root cause analysis exhibit some limitations: faults propagate through multiple paths within processes, thus causing these methods to occasionally produce spurious and unreliable results. Therefore, the process flowsheet containing the process topology details, i.e.,

in form of a P&ID, can be utilized to validate the results from the data-based methods to achieve improved results in root cause analysis (Thambirajah et al., 2009; Venkatasubramanian et al., 2003a; Yim et al., 2006).

A number of authors have introduced the approach which can preferably integrate the typical data-based methods, such as PCA and PLS, together with qualitative model-based methods, such as directed graphs (Chiang and Braatz, 2003; Lee et al., 2003). However, their proposed solution requires a manual approach to passing the results from a data-based method to the signed directed graph (SDG) method in order to verify the correct root cause from a set of candidate root causes suggested by the data-based method. Moreover, this approach is time consuming, requires deep knowledge of the process and is error prone. Hence, a comprehensive and automated approach for diagnosing process faults is indeed needed.

This thesis aims to utilise a method to firstly extract the topology data from a process using an electronic P&ID tool (i.e., AutoCAD P&ID) and to subsequently convert this data in order to obtain a connectivity matrix. The matrix will be utilized for validating another matrix, i.e., a causality matrix acquired from a data-based method, thus providing a non-spurious solution for determining and verifying fault propagation paths. This thesis develops a procedure to integrate a causality matrix with a process monitoring system, to enable the visualization of fault propagation paths. Thus, it contributes significantly to the efforts of identifying the root cause and the propagated effects of a fault in a process. Importantly, this is accomplished using a real DCS automation system, i.e., the ABB System 800xA, which is widely utilised in the process industries.

The remainder of this thesis is organized as follows. Chapter 2 presents the data-based causal analysis methods and the topology-based models, including electronic P&ID. Chapter 3 presents both the commercial and the academic applications for identifying the propagation of faults in fault diagnosis. Chapter 4 provides the motivation and states the tasks of the experimental part of the thesis. Chapter 5 presents a case study of a flotation pilot process, as well as presents the creation of a topology-based model (i.e., P&ID) for the flotation process and the generation of a connectivity matrix. Chapter 6 presents the generation of a causality matrix by implementing the Granger

causality method on the data acquired from the flotation process. The validated causality matrix is then accessed from the process monitoring system for visualizing the fault propagation paths in the flotation pilot process. Finally, Chapter 7 provides a conclusion for the entire thesis.

LITERATURE PART

2 FAULT PROPAGATION AND CAUSAL ANALYSIS

Industrial plants typically have well connected process units as well as complex interconnections among process equipment and instruments. Additionally, interactions among control loops are common in industrial processes. These observations are partially due to the attempt to consume material and energy more efficiently through the utilization of recycling streams and heat integration (Horch et al., 2007). Although, the past decades have seen significant advancements in the control and monitoring of industrial processes, certain faults and disturbances are nevertheless unavoidable in industrial process operations. A few examples include fouling of a heat exchanger, oscillations caused by valve stiction, degradation of catalyst and controller interactions; the effects of which can potentially influence the process performance in negative ways (Pozo Garcia et al., 2013; Thornhill and Horch, 2007). Thus, when a process is operating under an abnormal event, i.e., under the presence of a fault or disturbance, the effects of such upset typically propagate downstream to other equipment and units of the process.

Moreover, many of the existing approaches to locate root causes and to identify fault propagation paths are known to generate spurious solutions, since the candidate root causes are often interlinked and effects of a fault can propagate through multiple paths. Landman et al. (2014) reported that even in applications adopting multiple methods, each method might point to different potential root cause. Consequently, different researchers have introduced the “top-down” approach to combine complementary methods, in order to drastically reduce the amount of spurious results from the methods. Bauer and Thornhill (2008) indicated that methods for causal analysis can also be used for other purposes, e.g., root cause location and isolation. However, their use in the identification and verification of fault propagation paths requires also the addition of process knowledge either using a manual or an automated approach. The addition of process knowledge, for instance, using process flowsheet helps in validating the hypothesis generated by data-driven causal analysis.

This Chapter presents an introduction to causal analysis of faults, mainly through data-driven causal analysis and topology-based methods. The first section of the Chapter defines and provides a background to fault propagation paths analysis and its benefits in industrial processes. It further describes a number of data-driven causal analysis methods, which are beneficial to fault propagation paths analysis. The second section presents the topology-based qualitative causal models and their use in the domain of causal analysis research.

Causality among variables must be analysed and their cause-and-effect relationship subsequently mapped out, as the initial steps towards the diagnosis of a faulty system. Causality is a physical phenomenon based on the cause-and-effect relationship between variables (Pearl, 2009). Additionally, propagation of a fault can happen through multiple paths within a process. This can be either through physical (i.e., material and energy) or signal flow paths, hence contributing to the difficulty of identifying the root causes and the propagations paths of faults. Accordingly, this leads to the application of several methods used today for acquiring causal models, which are ultimately utilised for identifying the propagation paths and root causes of faults.

2.1 Data-driven causal analysis methods

Large quantities of historical data, typically available in industrial processes, account for one of the reasons that data-driven methods are beneficial to causal analysis. Over the years, a number of researchers have acknowledged the benefits of data-driven methods of causal analysis to fault propagation and root cause analysis problems, since they require no prior knowledge of the intrinsic system (Bauer et al., 2007; Duan et al., 2014; Landman et al., 2014; Yang and Xiao, 2012). The technique behind many of the data-driven causal analysis methods is the estimation of the degree at which the individual time series comprising the historical data affect one another. Yang and Xiao (2012) categorized data-based methods into the following three main classes: frequency domain methods, time domain methods and information theory methods.

A causal model in form of a causality matrix is the typical outcome derived from data-driven causal analysis. Specifically, a causality matrix identifies the causal relationship between a root cause and the propagated effects in a structured model (Bauer and Thornhill, 2008; Landman et al., 2014; Thambirajah et al., 2009). Furthermore, a causality matrix usually identifies the direction of propagation of the effects of a fault to the affected variables, in addition to the identification of the actual root cause (Bauer and Thornhill, 2008; Thambirajah et al., 2009). A causality matrix is occasionally referred to as a cause-and-effect matrix, since each element (*i,j*) of the matrix denotes a "cause and effect" relationship from variable *i* to variable *j* (Landman et al., 2014; Thambirajah et al., 2009). Additionally, the values of these matrix elements denote the strength of the interactions among the process variables which they represent. An example of a causality matrix, derived from a time series using the transfer entropy method for causal analysis, is seen in Table 1. Furthermore, the method has indicated three candidate root causes of the emerged fault. However the actual root cause must be ultimately identified. A number of data-driven methods for causal analysis are presented as follows.

Table 1: A causality matrix derived using transfer entropy (Thambirajah et al., 2009)

Cause elements	Effect elements							
	LevCont-001	PressCont-001	PressInd-001	PressInd-002	TempInd-001	TempInd-002	TempInd-003	TempCont-001
LevCont-001	-	0	0	0	0	0	1	0
PressCont-001	0	-	0	0	1	1	1	0
PressInd-002	0	0	-	0	1	1	1	0
PressInd-001	0	0	0	-	0	0	0	0
TempInd-001	0	0	0	0	-	0	0	0
TempInd-002	0	0	0	0	0	-	0	0
TempInd-003	0	0	0	0	0	0	-	0
TempCont-001	0	0	0	0	0	0	0	-

2.1.1 Time-domain Granger Causality

Time-domain Granger causality is a data-driven method of causal analysis, which is based on linear predictions of time series. Granger causality is based on the evaluation of the relationship between two variables. The linear prediction, based on the lagged values of one variable is compared with the combined linear prediction, based on the augmented lagged values of both variables. A conclusion that the first variable is Granger-caused by the second variable is reached if the augmentation produces a better regression outcome (Yang and Xiao, 2012). Specifically, a variable x_t Granger-causes (i.e. influences) another variable y_t, if the lagged values of x_t reduces the error of estimating y_t in a linear regression model of combined y_t and x_t. This is contrary to a regression model of only lagged values of y_t (Duan et al., 2014).

The causal influence of a time series by another time series can be estimated by initially using a univariate prediction model. For a univariate autoregression (AR) of a time series y_t, a restricted model is initially formulated as follows:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + ... + a_m y_{t-m} + \xi_{y,t} \quad (1)$$

where a_i, $i = 1 \ldots m$, coefficients of the AR model

ξ_y = prediction error (i.e. residual)

Likewise, a bivariate autoregression model for estimating the causal influence of the two multiple stationary time series over one another can also be formulated (Landman, 2013; Yuan and Qin, 2012). Therefore, an unrestricted model (i.e. augmented regression) is formulated by including the lagged values of x_t in the model as follows:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + ... + a_m y_{t-m} + b_1 x_{t-1} + ... + b_m x_{t-m} + \xi_{y|x,t} \quad (2)$$

where m = order of the AR model

$\xi_{y|x,t}$ = prediction error

A conclusion can be drawn that x_t Granger-caused y_t if the prediction of y_t is significantly better after the inclusion of x_t (i.e., $\mathrm{var}[\xi_{y|x}] < \mathrm{var}[\xi_y]$) in the augmented regression model.

Elements of complex systems are known to interact regularly among one another. Therefore, Ladroue et al. (2009) extended the pairwise Granger causality to a generalised Granger causality, which enables its application to multi-dimensional data input for causal analysis. This allows for the implementation of the method to test for Granger-causality among a group of signals. Hence, considering two multiple stationary time series y_t and x_t of length n; the generalised univariate AR model is formulated for each of the time series individually as follows:

$$y_t = \sum_{i=1}^{k} b_{y,i} y_{t-i} + \xi_{y,t} \tag{3}$$

$$x_t = \sum_{i=1}^{k} b_{x,i} x_{t-i} + \xi_{x,t} \tag{4}$$

where b's = coefficients of the AR model

ξ's = prediction errors (i.e. residuals)

k = model order, defines the number of the time lag terms in the model

Equations (3) and (4) are referred to as a restricted AR model of the time series y_t and x_t. Ladroue et al. (2009) represented the multi-dimensional time series as an unrestricted model or full model, which is jointly formulated in the following bivariate AR model:

$$y_t = \sum_{i=1}^{k} a_{y1,i} y_{t-i} + \sum_{i=1}^{k} a_{y2,i} x_{t-i} + \xi_{y|x,t} \tag{5}$$

$$x_t = \sum_{i=1}^{k} a_{x1,i} y_{t-i} + \sum_{i=1}^{k} a_{x2,i} x_{t-i} + \xi_{x|y,t} \tag{6}$$

where A's = coefficients of the AR model

ξ's = prediction errors (i.e. residuals)

k = model order, defines the number of the time lag terms in the model

A causal influence from x_t to y_t can be concluded if the variance of ξ_y is significantly larger than the variance of $\xi_{y|x}$, i.e., the prediction of y_t improves by including the time lagged values of x_t in the AR model. Thus, the degree of interaction (i.e. causal relationship) of the complex Granger causality is quantified by:

$$F_{x \to y} = \ln \frac{\mathrm{var}(\xi_y)}{\mathrm{var}(\xi_{y|x})} \tag{7}$$

The value of F is obviously $F_{x \to y} = 0$ in the absence of any causal influence from x_t to y_t; otherwise, $F_{x \to y} > 0$ (Ladroue et al., 2009). Furthermore, this bivariate Granger causality can be generalized to a multivariate case. Specifically, in a process with n observed variables ($x_1, x_2, x_3, \ldots, x_n$), x_j Granger causes x_i if the incorporation of x_j can help to predict x_i when all other variables are included in the regression model (Duan et al., 2014; Yuan and Qin, 2012). Hence, for a full model in Eq. (5) and (6), the residual covariance matrix can be formulated as follows:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \cdots & \Sigma_{1n} \\ \Sigma_{21} & \Sigma_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ \Sigma_{n1} & \cdots & \cdots & \Sigma_{nn} \end{bmatrix}_{n \times n} \tag{8}$$

Subsequently, an analysis begins with a full (unrestricted) model, where a predictor variable x_i is excluded at a time; thus, a new corresponding restricted model can be generated, and its (n-1) by (n-1) dimension residual matrix is defined as follows:

$$\rho = \begin{bmatrix} \rho_{11} & \cdots & \rho_{1(j-1)} & \rho_{1(j+1)} & \cdots & \rho_{1n} \\ \vdots & \ddots & & & & \vdots \\ \rho_{(j-1)1} & & & & & \rho_{(j-1)n} \\ \rho_{(j+1)1} & & & & & \rho_{(j+1)n} \\ \vdots & & & & \ddots & \vdots \\ \rho_{n1} & \cdots & \rho_{n(j-1)} & \rho_{n(j+1)} & \cdots & \rho_{nn} \end{bmatrix} \tag{9}$$

Hence, from the residual matrix, it is possible to quantify the Granger causality from a variable j to another variable i as follows:

$$F_{j \to i|n} = \ln \frac{\rho_{ii}}{\Sigma_{ii}} \tag{10}$$

The degree of interaction $F_{j \to i|n}$ is conditioned on other process variables. Finally, the residual covariance matrix in Eq. (8) and the residual matrix in Eq. (9) are defined above, since their individual elements are required in quantifying multi-dimensional Granger causality in Eq. (10).

2.1.2 Cross-correlation analysis

The causality (i.e., causal relationship) among variables can be captured through analysis of the correlation between them. Moreover, the effects of fault and process disturbances take some time to propagate; hence, the time delays between process measurements enable the method of cross-correlation function (CCF) to be beneficial in the analysis of causality among variables. Cross-correlation analysis can be used for identifying the cause-and-effect relationships between two variables of a process. The cross-correlation function measures the similarity of a bivariate stationary time series,

in order to describe their statistical properties. Bauer and Thornhill (2008) defined a cross-correlation function of two time series x_t and y_t, sampled at x_i and y_i respectively, where $i = 1...N$ (N is the total number of samples), as a function of the assumed lag k:

$$\phi_{xy}[k] = \begin{cases} \frac{1}{N-k} \sum_{i=1}^{N-k} \hat{x}_i \hat{y}_{i+k} & \text{if } k \geq 0 \\ \frac{1}{N+k} \sum_{i=1-k}^{N} \hat{x}_i \hat{y}_{i+k} & \text{if } k < 0 \end{cases} \tag{11}$$

where, $\hat{x}$ and $\hat{y}$ are obtained by mean centering x and y, and scaling to a unit standard deviation

Causal analysis with the method of cross-correlation function tests for the maximum (highest peak) and minimum (lowest valley) of the cross-correlation function, in order to estimate the time delay between the two time series. Thus, the relationship between these two time series is evaluated by finding the maximum absolute value of the cross-correlation function (Bauer and Thornhill, 2008; Landman, 2013). Hence, for a specific time index $k = k^{\max}$ where the absolute value of the CCF $\phi_{xy}[k]$ is at its maximum, a time series shifted by $k^{\max}$ is said to correlate to the other time series. The absolute value of the highest peak is often less than that of the lowest valley. In such cases, inverting and shifting the second series by $k^{\min}$ makes both series become most similar, given that $k^{\min}$ is the time index of the lowest valley. Therefore, the maximum and minimum of the cross-correlation function ϕ_{xy} are both recorded for the analysis as follows:

$$\begin{aligned} \phi^{\max} &= \left\{ \phi_{xy}[k^{\max}] \right\}, \\ \phi^{\min} &= \left\{ \phi_{xy}[k^{\min}] \right\}. \end{aligned} \tag{12}$$

The selection of either $k^{\max}$ or $k^{\min}$ as the detected time delay between the time series x and y is dependent on whether $\phi^{\max}$ or $\phi^{\min}$ has the greater absolute value. Thus, the detected time delay λ from x to y is defined as follows:

$$\lambda = \begin{cases} k^{\max} & \text{if } \phi^{\max} \geq -\phi^{\min} \\ k^{\min} & \text{if } \phi^{\max} < -\phi^{\min} \end{cases} \tag{13}$$

The time delay correlation between x_i and $y_{i+\lambda}$ must exceed the threshold of the statistical significance as described in Bauer and Thornhill (2008). Hence, the maximum time delayed correlation between these two time series is given as:

$$\rho = \max\left\{\phi^{\max}, |-\phi^{\min}\right\} \tag{14}$$

The obtained value for the maximum time delayed correlation ρ must be evaluated carefully; since ρ is a statistical estimate and unavoidably prone to some uncertainty due to noise, disturbances and the data windows size (Yang et al., 2012). Accordingly, the value of the cross-correlation function between two time series should be checked against a threshold. The effects of noise will dominate the results if the correlation between two series is quite weak. Hence, the CCF values that are considerably greater than a pre-specified threshold (e.g., ± 0.2) can be concluded to be results of true correlations.

Bauer and Thornhill (2008) introduced a measure known as a directionality index, the purpose of which is to quantify the difference between the minimum and maximum of ϕ_{xy}. Since even arbitrary series have both $\phi^{\max}$ and $\phi^{\min}$, a directionality must be confirmed by checking that the magnitude of one is considerably larger than the other. The directionality index for ϕ_{xy} is defined as follows:

$$\psi = 2\frac{|\phi^{\max} + \phi^{\min}|}{\phi^{\max} + |\phi^{\min}|}. \tag{15}$$

A small directionality index ψ signifies an ambiguous result; thus, no decision is made due to the similarity in the magnitude of $\phi^{\max}$ and $\phi^{\min}$.

Ultimately, the implementation of CCF in causal analysis for the identification of fault propagation paths requires a causal model to be presented in form of a causality matrix.

Hence, as an illustrative example, element $\lambda_{a,b}$ of the matrix indicates the estimated time delay of variable b relative to variable a. The approach to constructing a causality matrix from CCF is briefly described as follows. Firstly, the estimated time delays Eq. (13) among all possible pairs of variables are combined to create a delay matrix. Bauer and Thornhill (2008) described the tests of statistical significance of both maximum correlation ρ and a directionality index ψ, which is the basis for the threshold for the elements of the delay matrix. Secondly, the elements of the delay matrix which have exceeded the thresholds are organized in a causal matrix Λ. Additionally, the maximum number of entries in the matrix is $\frac{p(p-1)}{2}$, for a p x p matrix.

Lastly, the estimated delays must be arranged to locate a maximum number of entries above the main diagonal of the matrix. This rearrangement leads to the location of the most likely root cause variable in the first row of the matrix, since all other variables are time delayed relative to it. An algorithm suitable for performing this rearrangement is presented in Bauer et al. (2007). Ultimately, the non-negative causal matrix Λ is of the following form:

$$\Lambda = \begin{bmatrix} - & \lambda_{1,2} & \cdots & \lambda_{1,p} \\ & - & \ddots & \vdots \\ & & \ddots & \lambda_{p-1,p} \\ & & & - \end{bmatrix} \tag{16}$$

where entry $\lambda_{x,y}$ indicates the time delay between the variables x and y.

2.1.3 Transfer entropy

Multivariable systems, such as chemical processes, can benefit from the transfer entropy (TE) method, since essential structural information can be acquired by quantifying the extent at which each time series contributes to the information exchange among each other. The complex interactions among process variables contribute to the challenges in fault propagation paths and root cause analyses. Thus, the causal relation between the process variables can also mean information transfer

between them (Yang et al., 2012). Transfer entropy offers an advantage of giving a good sense of causality without the time delay information, and the ability to distinguish information which is exchanged from a common source. Schreiber (2000) first introduced the transfer entropy method as an information theoretic measure that quantifies the statistical coherence among systems developing in time. In information theory, Schreiber (2000) defined transfer entropy from time series x to time series y as follows:

$$t\left(y \mid x\right) = \sum_{y_{j+h}, \mathrm{y}_j, \mathrm{x}_j} p\left(y_{j+h}, \mathrm{y}_j, \mathrm{x}_j\right) \cdot \log \frac{p\left(y_{j+h} \mid \mathrm{y}_j, \mathrm{x}_j\right)}{p\left(y_{j+h} \mid \mathrm{y}_j\right)} \tag{17}$$

where p is complete or conditional probability density function (PDF)

$$\mathrm{x}_j = \left[x_j, x_{j-\tau}, ..., x_{j-(k-1)\tau} \right]$$

$$\mathrm{y}_j = \left[y_j, y_{j-\tau}, ..., y_{j-(l-1)\tau} \right]$$

h = prediction horizon

τ = sampling period

Transfer entropy is the difference between the information about y acquired from the simultaneous observation of the past values of both y and x, and the information about y acquired from the past values of y only. A range of possible values for parameters k and l, required for implementing the transfer entropy in Eq. (17) is provided in Bauer and Thornhill (2008). Thus, the separate outcomes obtained from the ranging values of h and τ can be tested and compared.

Bauer et al. (2007) first adopted the transfer entropy method as a data-driven analysis method to generate a qualitative causal model in form of a causal map, in order to identify the propagation paths of process disturbances. They described how the magnitude and direction of information transfer can be determined to measure the causality when the transfer entropies in two directions are considered. The influence

of y on x is compared with the influence of x on y, in order to determine the causal measure as follows.

$$t_{x\to y} = t(y|x) - t(x|y) \tag{18}$$

If the value of $t_{x\to y}$ is positive, then x causes y, while the value is negative if y causes x. However, if $t(x|y)$ and $t(y|x)$ both have a similar value, then no causality is detected, thus $t_{x\to y}$ is close to zero.

A histogram or nonparametric kernel methods that fit any shape of the distributions can be utilized to estimate the p (i.e., PDF) function in Eq. (17). A robust Gaussian kernel method is adopted and the function is defined as follows:

$$K(\upsilon) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\upsilon^2}. \tag{19}$$

Therefore, the PDF for a univariate time series can be estimated using the following function:

$$\hat{p}(x) = \frac{1}{Nh} \sum_{j=1}^{N} K\left(\frac{x - x_j}{h}\right) \tag{20}$$

where N = the number of samples

h = the bandwidth chosen to minimize the mean square error of the PDF estimation by obtained by $h = c \cdot \sigma \cdot N^{-1/5}$, for $c = (4/3)^{1/5} \approx 1.06$

The probability density function estimation presented above is utilized in the transfer entropy between two time series, which is used in a univariate measure of the causal relationship between the time series. However, multivariate analysis of causal relationship between multiple time series is highly beneficial, since the complex interconnectedness of variables in industrial processes can hereby be captured. Therefore, the estimation of PDF for an m-dimensional multivariate case is presented as follows:

$$\hat{p}(x_1, x_2, \ldots, x_m) = \frac{1}{N h_1 \cdots h_m} \sum_{j=1}^{N} K\left(\frac{x_1 - x_{j1}}{h_1}\right) \cdots K\left(\frac{x_m - x_{jm}}{h_m}\right) \tag{21}$$

where $h_s = c\dot{\sigma}(x_{is})_{j=1}^{N} \cdot N^{-\frac{1}{(4+q)}}$, for $s = 1, \ldots, m$

The computational burden is rather high in the application of this method. This is somehow due to several parameters that must be specified for each application, as well as the high dependence on the estimation of PDFs, often in Gaussian forms (Yang et al., 2012). Causal relationships among several variables of industrial processes, especially those processes with numerous recycle streams, can be analysed using the transfer entropy method. Furthermore, information transfer between two time series in both directions (i.e., $t(y|x)$ and $t(x|y)$) can be measured with transfer entropy.

Ultimately, the result of applying the information theoretic TE measure to numerous process measurements is to generate a causal model. The causality measure according to Eq. (18) is calculated for all combinations of variables $x^{(m)}$, such that $\frac{m(m-1)}{2}$ relationships are calculated and arranged in the causality matrix below, where $m = 1, \ldots, m$ measurements are considered.

$$\mathrm{t} = \begin{bmatrix} 0 & t_{x^{(1)} \to x^{(2)}} & \cdots & t_{x^{(1)} \to x^{(m)}} \\ -t_{x^{(1)} \to x^{(2)}} & 0 & \cdots & t_{x^{(2)} \to x^{(m)}} \\ \vdots & \vdots & \ddots & \vdots \\ -t_{x^{(1)} \to x^{(m)}} & -t_{x^{(2)} \to x^{(m)}} & \cdots & 0 \end{bmatrix} \tag{22}$$

Cause variables are represented in the rows of the matrix while effects variables are represented in the matrix columns. Because of symmetry, negative causality measures can be ignored while no information will be lost.

Bauer et al. (2007) adopted a technique to ignore (i.e., eliminate) those matrix entries (i.e., relationships) having a significance level below a certain limit. The technique then yields a matrix Λ comprised of only 1's and 0's, where entries 1's indicate that the column variable influences the row variable, and a 0 entry indicates that no

relationship exists between the variables. An algorithm described in Bauer et al. (2007) is then utilized for rearranging the order of the causality matrix entries. This rearranging maximizes the number of non-zero entries above the main diagonal. Furthermore, the causal model to be generated here is a causality matrix, which can subsequently be utilized to generate a causal map or a directed graph. Bauer et al. (2007) introduced an algorithm that can construct a causal map from the causality matrix. The causal model can thus be utilized in its different forms to identify fault propagation paths and a root cause of the fault.

2.2 Topology-based causal models

This section focuses on topology-based causal models which are useful in revealing the physical relationships among process equipment and instruments. These models provide details of connections in process units and equipment that are very crucial to fault diagnosis research. Several authors have emphasized the significance of utilizing process topology-based causal models. These qualitative topology-based causal models can be used for improving and verifying the results of data-driven causal analysis. A technique of capturing these connections is described and later demonstrated in the next subsection. The subsection following it further demonstrates the method of creating a causal model from the captured process topology data.

The qualitative model of a process must be created, for instance, based on a piping and instrumentation diagram (P&ID) representation of the process. P&ID displays the flowsheet of a process in a clear and concise manner, allowing for a good access to the topology details of the process during fault diagnosis research. A process topology-based causal model can be obtained either in form of a signed directed graph (SDG) or a connectivity matrix. Both causal model types offer a major benefit of revealing the relationship or physical connections among process elements.

It is important to consider a technique of retrieving the details contained in the P&ID of a process for further processing into causal models. Consequently, in recent years, some authors have devised methods of extracting topology details from a process

flowsheet, which is then used for obtaining causal models. Therefore, computer aided engineering exchange (CAEX) schema, which is specified in the IEC/PAS 62424 standard is widely used today as the standard for implementing data exchange, used for converting process topology data into topology-based causal models. The data exchange is commonly accomplished in the XML or XLS/XLSX data formats (Fedai and Drath, 2005). Thambirajah et al. (2009); Yim et al. (2006) described the implementation of process topology data extraction using the XML data format, which conforms to the CAEX data exchange standard.

2.2.1 Signed directed graphs

A process knowledge-based causal model typically contains much understanding of a process and further provides a description of the means by which fault propagates from a root cause to other positions (Thornhill et al., 2003). Over the years, the benefits of graphical qualitative models showing the relationships between process variables have been acknowledged by a number of authors. First proposed by Iri et al. (1979) for an algorithm used in fault diagnosis, signed directed graphs or signed digraphs (SDG) have been the most extensively used form of causal knowledge for fault diagnosis in industrial processes. Their uses have extended over the years from single-fault diagnosis to multiple-fault diagnosis (Vedam and Venkatasubramanian, 1997).

An SDG can be constructed by representing the process variables and parameters as graph nodes and representing causal relations as directed arcs. The following values are utilized in the nodes of the SDG: (+), (-) and (0) for representing the higher than steady-state, lower than steady-state and nominal steady-state values respectively (Vedam and Venkatasubramanian, 1997). The directed arcs in an SDG points from a cause node to an effect node; an arrow at the pointing end of an arc indicates the direction of causality. Furthermore, a positive arc (solid line) or a negative arc (dotted line) points from a cause node to an effect node, in order to represent change in the cause node and the effect node either in the same direction or in the opposite direction, respectively (Gao et al., 2010).

A signed digraph for a process can be constructed from the mathematical models representing the process. Occasionally, it is necessary to use another manual approach to SDG construction by utilizing process knowledge, such as process flowsheets and operators' knowledge (Yang and Xiao, 2012).

For example, Yang and Xiao (2012) presented a bitank process shown in Figure 1, from which an SDG model in Figure 2 was subsequently built. The bitank consists of two identical tanks placed on the same level and connected together by a pipe. The following set of algebraic and differential equations describe the process:

$$C_1 \frac{\mathrm{d}l_2}{\mathrm{d}t} = f_1 - f_3 - f_5,$$

$$C_2 \frac{\mathrm{d}l_7}{\mathrm{d}t} = f_5 - f_8,$$

$$f_3 = \frac{1}{R_{b1}} \sqrt{l_2}, \tag{21}$$

$$f_5 = \frac{1}{R_{12}} \left(\sqrt{l_2} - \sqrt{l_7} \right),$$

$$f_8 = \frac{1}{R_{b2}} \sqrt{l_7}.$$

where f_1, f_3, f_5 and f_8 are the flowrates

l_2 and l_7 are the levels in Tank 1 and Tank 2 respectively

C_1 and C_2 are cross-sectional areas of the Tank 1 and Tank 2 respectively

R_{b1}, R_{b2} and R_{12} are the resistances of the outlet pipes and the connecting pipe respectively

The SDG model in Figure 2 is hereby built by converting Eq. (21) into the nodes and arcs that constitute the SDG model, where negative relationships are denoted by dotted arcs and positive relationships denoted by solid arcs.

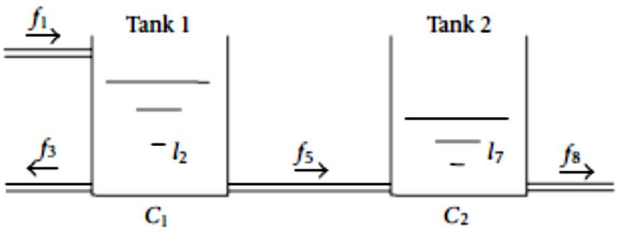


Figure 1: The process flowsheet of the bitank process (Yang and Xiao, 2012)

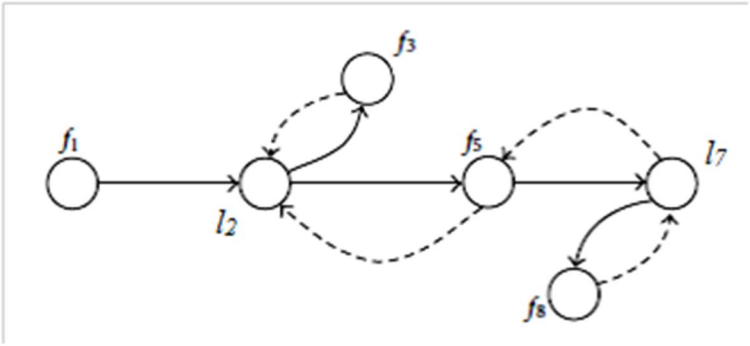


Figure 2: An SDG model derived from model equations (Yang and Xiao, 2012)

Vedam and Venkatasubramanian (1997) indicated that for several processes, both mathematical model equations and process knowledge must inevitably be combined in order to develop an SDG for the entire process. Nevertheless, over the past decades, researchers have introduced the interesting approach to an automated procedure for developing SDGs based mainly on mathematical model equations (Vedam and Venkatasubramanian, 1997). This enabled the widespread use of SDGs today for industrial process application, since SDGs can be constructed for even more process units where an automated development is available.

Maurya et al. (2007) utilized an SDG model in their attempt to minimise spurious results generated by a data-driven analysis for fault diagnosis. Gao et al. (2010) also presented an approach to combine qualitative trend analysis (QTA) with SDG to provide a more effective fault diagnosis for an atmospheric distillation tower unit in

an atmospheric and vacuum distillation process. Nevertheless, there are some limitations to the application of SDG. One of the limitations of SDG is their lack of resolution, which is due to qualitative ambiguities in some applications. Moreover, the accuracy of an SDG model has an effect on the correctness of the results generated from overall causal analysis.

2.2.2 Electronic P&ID design in AutoCAD

Electronic P&ID software applications are often referred to as intelligent P&ID software. The commonly used intelligent P&ID software today includes, AutoCAD P&ID from Autodesk, PDMS™ from AVEVA, Comos P&ID from Siemens and SmartPlant P&ID from Intergraph. They enable users to create electronic P&ID drawings and to store essential drawings data in the database of the applications. Furthermore, they also conform to the CAEX standard since they enable the storage and export of drawings data in either XML or XLS/XLSX file formats. In addition to compliance to the CAEX standard, the AutoCAD P&ID drawing application also supports the following different standards of P&ID: PIP, ISO, ISA, DIN, and JIS (Autodesk, 2013).

The AutoCAD P&ID program starts by displaying the user interface as the main window as shown in Figure 3 below. P&ID PIP is selected from the dropdown menu at the top panel as the chosen drawing type among other AutoCAD based types of drawing included in the software. A menu bar, toolbar, ribbon and tool palette that are relevant to a chosen P&ID standard are available in the workspace right after selection, i.e. each workspace represents a different standard of P&ID. Thus, when P&ID PIP is selected, the menu bar, toolbar, ribbon and tool palette which are required for a P&ID PIP drawing are displayed in the workspace. All the functions that are available for operating and managing P&ID projects and drawing are organized respectively in the menu, toolbar and ribbon. In addition, the tool palette as shown in the upper right side of Figure 3, allows for easy and quick access to the entire elements needed for creating a P&ID drawing. This means that the user needs to only select a process component from the tool palette and to select a spot in the drawing space to place the component.

Both pipelines and signal lines which are used for connecting the components as well as all inline elements (e.g. valve) of the process are available in the palette. Furthermore, all the available process elements included in the tool palettes conform to their corresponding industry standards.

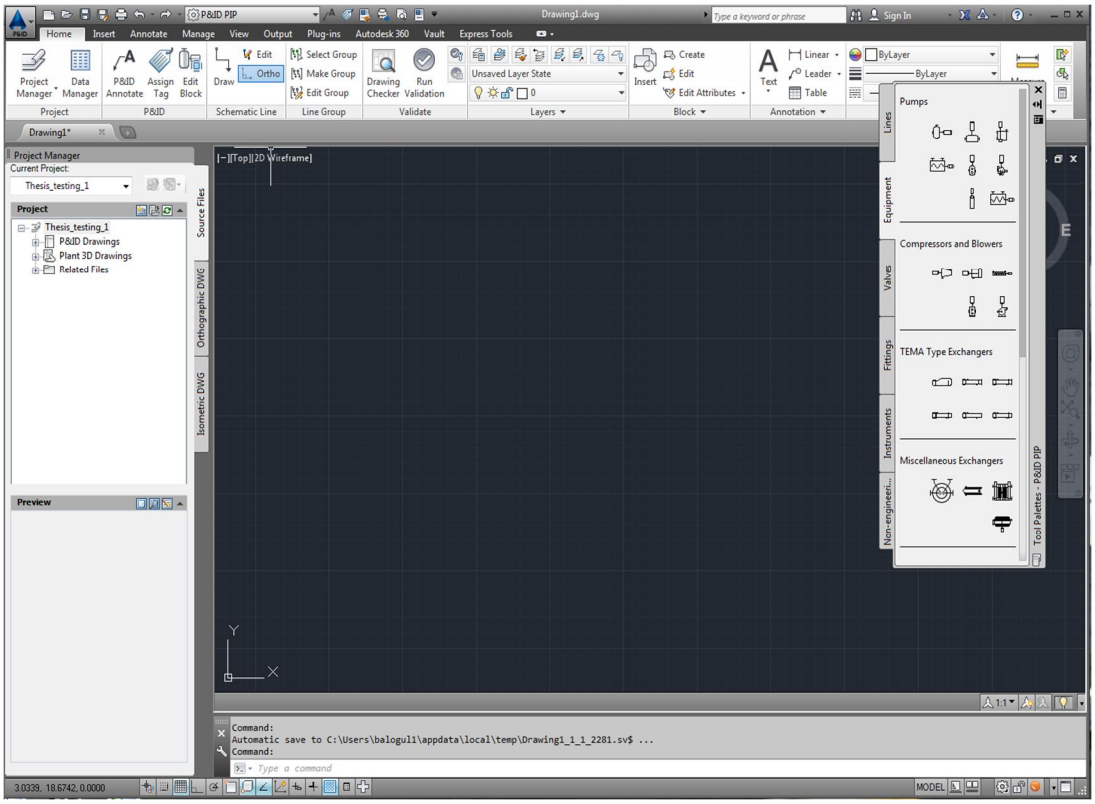


Figure 3: AutoCAD P&ID user interface displaying an empty PIP workspace

One of the major benefits offered by the AutoCAD P&ID application is the storage of the P&ID drawing components and lines related data in the Data Manager. This enables viewing, editing, manipulation of the data as well as import of the data which is modified in Microsoft Excel, back into the P&ID drawing (Autodesk, 2013). Creating a dynamic process flowsheet in electronic P&ID representation using AutoCAD P&ID typically requires a series of particular steps. Figure 4 below illustrates a sequence of tasks to be performed in creating a typical dynamic flowsheet of a process in AutoCAD P&ID using the PIP standard.

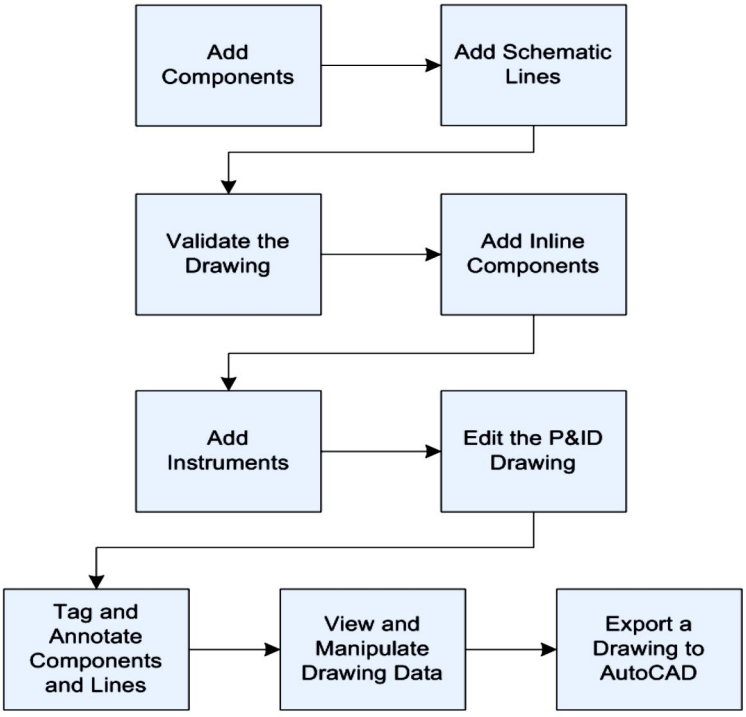


Figure 4: Illustration of dynamic flowsheet design for a process in AutoCAD P&ID (Autodesk, 2013)

2.2.2.1 Extraction of the process topology data from the AutoCAD P&ID

Intelligent P&ID applications such as AutoCAD P&ID are considered "intelligent" partly due to their rich libraries of process items, high functionality and their compliance to certain standards, e.g., the ISO 15926 XML scheme for data exchange. Additionally, they offer also the possibility to store, manage and export P&ID drawing data, thus making them highly beneficial to fault diagnosis research. Drawing data refers to all the data entries contained in all the elements of the P&ID drawing such as names, types, coordinates and connections. Figure 5 shows the structure of the data needed to construct a connectivity matrix from the extracted topology data of a process.

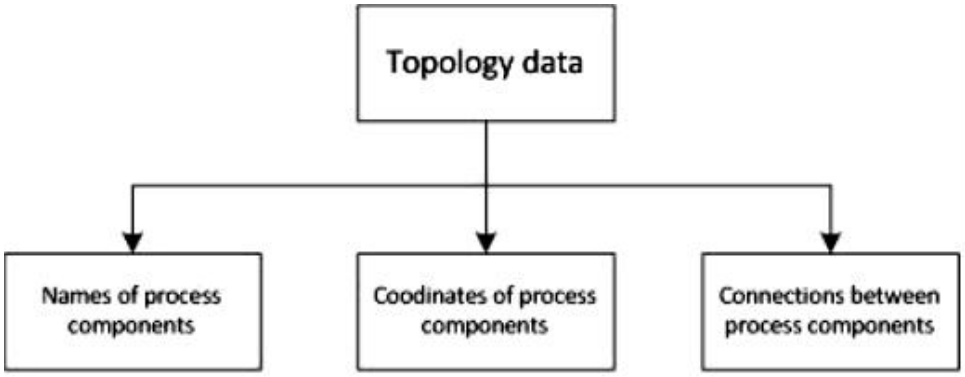


Figure 5: Structure of the topology data for constructing a connectivity matrix (Sun, 2013)

After a process flowsheet is completely drafted in the AutoCAD P&ID drawing application, the next task is to perform a validation of the P&ID drawing within the application. The purpose of this is to verify the consistency of the process equipment in the drawing as well as the pipe connections between them. The validated P&ID of the process is ready to be utilized; thus, the Data Manager is accessed from the application menu. Afterwards, the AutoCAD P&ID Data Manager window opens up with all the available functions including the export function to external environment. The appropriate destination file path on the computer is specified in the export dialogue box; a Microsoft Excel spreadsheet file is subsequently created, which contain an abundant amount of drawing data in multiple sheets within the file. The drawing data is mainly a read-only data in the Data Manager; however, the data can be manipulated when exported to a Microsoft Excel file.

2.2.2.2 Connectivity matrix generated from the process topology data

The term connectivity matrix refers to an array of numbers structured in a square matrix of order *N*, i.e. matrix size *N* x *N* where *N* denotes the number of variables. All the elements of a connectivity matrix can only have values either “0” or “1”. This depends on the existence of a forward physical directional connection between items represented by the intersection of a row header and a column header (Thambirajah et

al., 2009). Specifically, the connections between process items which constitute a connectivity matrix can be either an equipment-to-instrument connection or an equipment-to-equipment connection. Therefore, a process flowsheet or P&ID can be viewed and used as the source from which a connectivity matrix is constructed.

A connectivity matrix is utilised in this thesis because it can be obtained relatively easily from the topology data extracted from an electronic P&ID of a process. The numerical data extracted from the P&ID as described earlier can be transformed into a connectivity matrix which is also in numeric form (Sun, 2013; Thambirajah et al., 2009; Yim et al., 2006). Additionally, it is relatively easier to utilize the connectivity matrix for further analysis since it is a numerical form of the causal model. A connectivity matrix offers the benefit to view and to verify the existence of forward physical directional connections between the components of an industrial process. Thus, it offers the benefit of being easily utilized for refining a causality matrix acquired from a data-driven causal analysis, in order to achieve an enhanced fault propagation results. Specifically, the aforementioned concept is one the goals of this thesis.

3 APPLICATIONS OF FAULT DIAGNOSIS AND FAULT PROPAGATION ANALYSIS

Data-driven or purely data-based multivariate methods are widely utilized for causal analysis in fault diagnosis, since they require no process models and thus they are relatively easier to establish for industrial processes. These measurement-based methods generate good hypothesis about the root cause and propagation paths of a fault, in order to identify the most likely candidate root cause and its propagation paths. Likewise, some model-based causal methods e.g., SDG, which contains qualitative information on equipment connectivity, can generate hypothesis on fault causality. However, these qualitative model-based causal methods are known to occasionally produce spurious solutions when utilized in fault propagation paths analysis. Similarly, data-driven methods have a drawback, as their results can generate multiple candidate root causes and propagation paths for a single fault or disturbance.

Conventionally, qualitative information about the process from model-based causal methods have been implicitly used in the process industry; this happens when engineers manually use information e.g., from SDG or P&ID, to validate results of data-driven methods (Thornhill and Horch, 2007; Yim et al., 2006). Therefore, some researchers addressed the challenge by developing an automated procedure of extracting qualitative models for improving the results from data-driven causal analysis (Chiang and Braatz, 2003; Thambirajah et al., 2009; Yim et al., 2006). A vital component in this approach is the depth-first search algorithm, which enables the automated identification of propagation paths and location of the root cause of fault. Although, there are different methods of fault diagnosis in the process industry today; nevertheless, both data-driven methods and model-based causality are somehow used in fault diagnosis applications.

This Chapter describes some commercial applications used in industrial processes for visualizing faulty equipment and propagated effects of fault, as well as an academic application of a search algorithm used in the automated approach of fault propagation paths identification and validation. The first section of the Chapter describes a number of commercial applications adopted by process plants. This is followed by the second

section, where the cause-and-effect analyzer tool for identifying and verifying fault propagation paths is described.

3.1 Industrial condition monitoring and fault diagnosis applications

A large number of applications exist today used for condition monitoring and fault diagnosis in industrial processes. Thus, this section aims to identify the main trends in the industry by exploring two different kinds of industrial diagnostic applications in the following subsections.

3.1.1 Siemens asset management and condition monitoring system

Siemens AG is one of the leading manufacturers of equipment and automation solutions for the process industries. Asset management and condition monitoring applications are among the solutions offered with their automation solution packages to Siemens clients. Since Siemens offers no generalised software applicable to monitor and visualize fault propagation for an entire industrial process, these modular asset management and condition monitoring applications are applied for specific elements of a process. Siemens (2014b) has introduced the Simatic PCS 7 plant asset management software, it was initially developed for monitoring individual pumps in a process. A number of additional modules have been introduced afterwards for monitoring other mechanical components in the plant, including a heat exchanger, valves, turbo-compressors and positioners. The status of the plant mechanical equipment is achievable through a series of specific diagnosis modules, interfacing its specific mechanical equipment directly. This is made possible through advanced sensor signal processing already available in the plant control system, without the requirement of auxiliary sensors such as structure-bone or acceleration sensors.

The Simatic PCS 7 condition monitoring software can be set up individually for monitoring mechanical equipment and reporting faulty plant components to maintenance personnel. A number of different windows are customized for the plant

equipment being monitored. Figure 6 is an example of a dedicated monitoring faceplate for a heat exchanger in a process. Siemens branded the modular applications with commercial names for the specific modules; for instance, HeatXchMon is offered specifically for monitoring the condition of heat exchangers.

Figure 6: Monitoring of a heat exchanger with the Siemens Simatic PCS 7 software (Siemens, 2014b)

A number of other modules available with the Simatic PCS 7 software include CompMon for turbo-compressors, ValveMon for valves and PumpMon for pumps (Siemens, 2014b). Siemens have paid more emphasis on the condition monitoring of pumps in comparison to other mechanical equipment types in process plants. Accordingly, several types of faceplates are available in the PumpMon monitoring module, enabling as much as possible visualisation of different performance and safety conditions of individual pumps in a process. Figure 7 and 8 are examples of faceplates specifically dedicated to condition monitoring and reporting for the net positive suction head (NPSH) characteristic, and power and efficiency characteristic respectively of an individual pump in Simatic PCS 7 software.

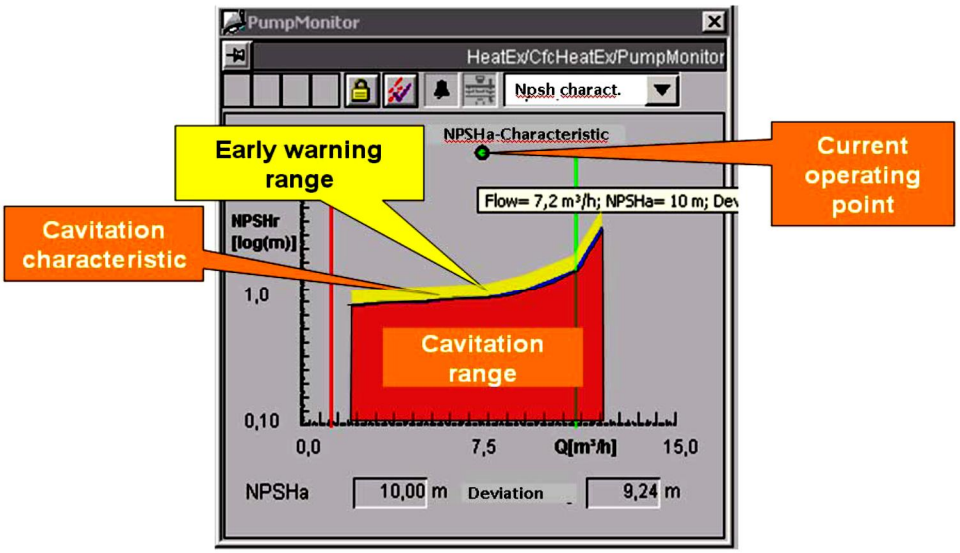


Figure 7: A dedicated monitoring of NPSH for a pump in Simatic PCS 7 software (Siemens, 2014a)

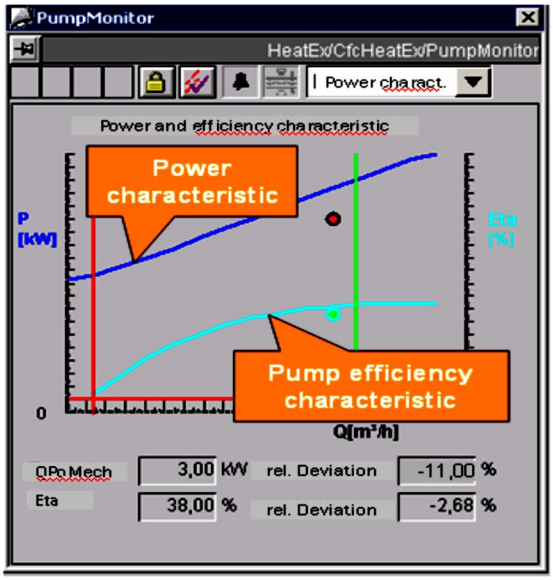


Figure 8: A dedicated monitoring of power and efficiency for a pump in Simatic PCS 7 software (Siemens, 2014a)

In addition to the power and efficiency, NPSH characteristics monitoring, the Simatic PCS 7 software also offers a possibility to monitor the condition of other

characteristics of a pump including power characteristics, delivery height characteristics, operating states (flow rate) and power values (mechanical, electrical and hydraulic). Thus, for fault diagnosis in a pump, the PumpMon diagnosis module establishes the threshold violations of the rated operating range for the pump, any deviations from the established characteristic curve, and finally generates reports on these to the personnel and stores them for further processing. This way, process alarms can be generated to warn operating personnel of abnormal conditions of any characteristic in the pump; this diagnostic logic is also used for other mechanical assets in a process plant (Siemens, 2014b).

3.1.2 ELMAS logical relations software

A commercial software which has been implemented together with other automation system packages is the Event Logic Modelling and Analysis Software (ELMAS). It comprises three main tools for analysing causes and consequences of fault in a system (Ramentor, 2014). Metso and Pöyry corporations are among other partners who have implemented ELMAS in their solutions. The entire system functionality must be taken into account in order to set up the software for a process, as well as identifying the most critical elements of the process that leads to a fault. ELMAS can also be an effective tool in creating collaboration between process operational and maintenance personnel (Ramentor, 2014). Fault Trees and Reliability Block Diagrams are very beneficial in the attempt to locate the root cause and manner of spread of fault in a system.

ELMAS Fault Tree offers a visual possibility to examine the relationships between elements comprising a system as a means of analysing the reliability of the system. The tool functions as an implementation of fault tree analysis, i.e., a logic tree composed of layers and nodes which maps out primary faults or events to the top level event or hazard (Venkatasubramanian et al., 2003b). The root node of the fault tree represents the abnormal event in the system. Figure 9 is an illustration of an abnormal event diagnosis in the ELMAS software. The nodes represent elements of a system;

hence, they are labelled with text that describes the node and provide the identity of the element.

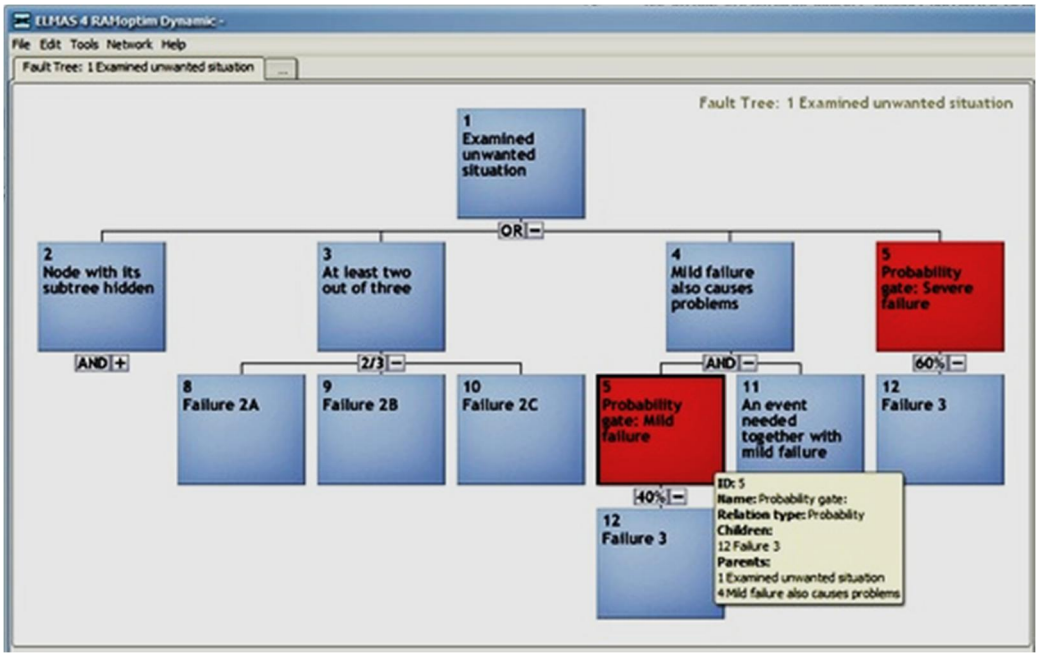


Figure 9: A system reliability analysis with the fault tree tool in ELMAS software (Ramentor, 2014)

ELMAS fault tree uses logic gates between the nodes of the model, defining the rules of the occurrence of an event. The typical logic gates used in the implementation are AND, OR, XOR as well as probability gates. The tool also provides the visualization of the relationship between the elements in terms of the logic connecting them. The logic gates between the nodes of the fault tree in Figure 9 are described as follows.

XOR - Node 1 occurs if one of node 2, node 3, node 4 or node 5 is true.

Probability - Node 5 occurs (severe failure) with 60% probability if node 12 is true.

AND - Node 4 occurs if node 5 and node 11 are simultaneously true.

OR - A Node x occurs if node y or node z is true.

ELMAS Reliability Block Diagram is a tool which enables the use of block diagrams for modelling the behaviour of a system, thereby greatly increasing the visibility of the entire system. A process operator can better understand the behaviour of the system, especially in applications where reliability block diagrams are complementarily used together with ELMAS fault tree (Ramentor, 2014).

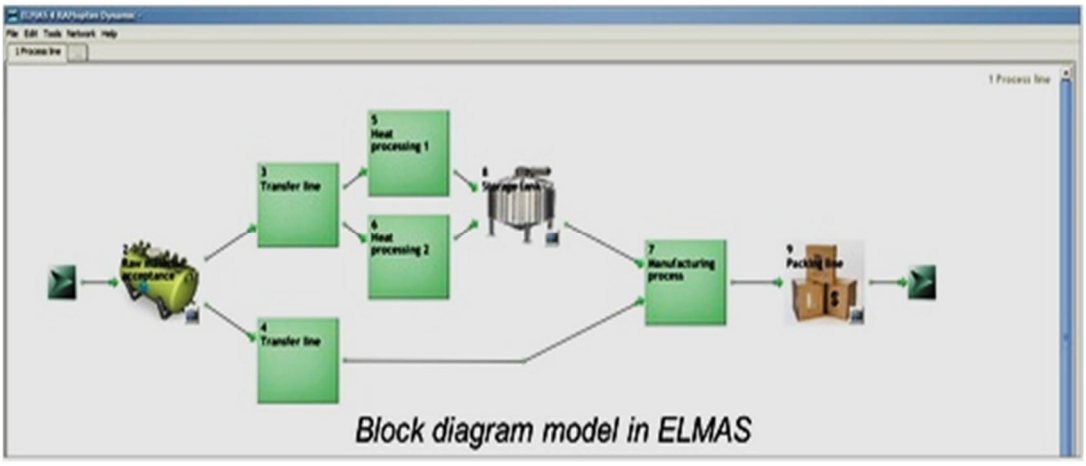


Figure 10: Reliability block diagram model, build in a tool from the ELMAS software (Ramentor, 2014)

The reliability block diagram is typically used in ELMAS to model a system in normal functioning state. Specifically, the reliability block diagram contrarily describes the same process as fault trees, where the system is instead modelled for abnormal situation. The blocks that constitute the model is similar to the nodes that constitute fault trees. Thus, a single node in the fault tree can be examined for a failure, e.g., disconnection or leakage, meanwhile the system connectivity in the reliability block diagram will be used to supply system topology detail during the analysis (Ramentor, 2014). Finally, in a reliability block diagram, the number of input and output connections in a single block can be defined, similar to the specific logical gates defined in fault trees. According to (Ramentor, 2014), ELMAS has been implemented in a countless number of applications.

3.2 Academic fault diagnosis and causal analysis applications

Chemical processes today constitute complex connections among process units and equipment; thus, locating the root cause or propagations paths of fault becomes even more challenging, especially in large-scale processes. Accordingly, several authors have acknowledged the benefits of utilizing the qualitative model of a process, e.g., a topology-based model to drastically improve the results generated from data-driven methods of causal analysis (Thornhill and Horch, 2007). The following subsection presents an academic application, where qualitative process models along with corresponding dedicated search algorithms are utilised for generating improved results from data-driven causal analysis.

3.2.1 The Cause-and-Effect Analyzer tool

Thambirajah et al. (2009) developed the "cause-and-effect analyzer", a set of tools which generates a connectivity matrix containing process topology details and a causality matrix representing the empirical relationships between variables affected by disturbance. The candidate root cause variables of a process disturbance and probable propagation paths are already identified in the causality matrix. Thus, the search of the connectivity matrix can either confirm the feasibility of a propagation hypothesis, or it can identify the most probable hypothesis for root cause and propagation paths if multiple paths exist. The cause-and-effect analyzer builds on the CAEX Plant Analyzer which a is prototype presented in Yim et al. (2006); where a rule-base or diagnosis facts are used for inference, enabled in a reasoning engine constructed with the help of prolog, a logic programming language.

The cause-and-effect analyzer tool set depended very much on the Extensible Markup Language (XML) for representing the process flowsheet of industrial processes. XML language is easily comprehensible by humans and can be easily interpreted by computers (Thambirajah et al., 2009). The tool set adopted XML technology primarily because a process flowsheet created using an electronic P&ID can export and exchange process drawings data in an open XML text file. The XML text file generated complies

with the Computer Aided Engineering Exchange (CAEX) schema; a standard which specifies the representation of process diagrams as specified in the IEC/PAS 62424 standards (Fedai and Drath, 2005). Hence, any electronic P&ID software utilised for building a process flowsheet must conform to the CAEX standard, thus enabling a proper transfer of the drawing data for further use without losing information. Additionally, a schema described under the CAEX standard the accepted structure for a CAEX-compliant XML file. Specifically, CAEX enables the representation of the process flowsheet as a sequence of nodes containing information, such as standard names, descriptions of the element represented and the directional connections among the process elements (Thambirajah et al., 2009).

In attempt to test the cause-and-effect analyzer, Thambirajah et al. (2009) made reference to the causality matrix derived in the work of Bauer et al. (2007), where three possible root causes were indicated from transfer entropy method instead of only one. False positive results which are due to statistical chance must be eliminated from the analysis, since Transfer entropy method is statistical in nature. Although, Bauer et al. (2007) utilized a high threshold to prevent spurious relationships among variables; however, some weaker causal relationships are ignored as a repercussion. As a demonstration of this fact, Thambirajah et al. (2009) confirmed in their case study that the Transfer entropy method failed to detect the causal relationships which could have resolved the uncertainty between the three hypothesised root causes. Thus, the causal links between the measurement points must be assumed and checked. The benefits of the process connectivity matrix is very evident here, since it will be utilized to check the feasibility of propagation paths from each of the hypothesised root causes to all the measurement points affected by a root cause. The most likely paths of propagation and root cause of a disturbance are confirmed by identifying the hypothesis which can best describe the propagation to all other measurement points affected by the disturbance (Thambirajah et al., 2009).

3.2.1.1 Search algorithm for verifying fault propagation paths

A search algorithm can be utilized in order to exploit the connectivity matrix in identifying and confirming feasible propagation paths of fault, using the hypothesised root causes and affects variables from the causality matrix. It finds all paths beginning at each postulated root cause and checks the presence of a path to other elements where effects of the disturbance have been detected (Thambirajah et al., 2009). This search algorithm is based on graph traversal, used for searching a sequence of nodes (i.e. process elements), where a node can only be visited once to prevent the search from leading to an endless loop. Specifically, this technique uses a depth-first search method, where a search begins from a specified node and proceeds to explore a specific set of nodes until termination, thus the algorithm retracts and advances the search afterwards. Before the search procedure, the causality matrix and the connectivity matrix must be linked, this is achieved by identifying the elements which represent indicators and controllers in the process (Thambirajah et al., 2009).

The search procedure is described as follows. The algorithm begins at the row of the connectivity matrix which represents the given hypothesised root cause; it searches for a matrix input "1", indicating the column element to which this row element is physically connected. Afterwards, the algorithm continues to the row of the connectivity matrix representing the column element, in order to identify the element it is connected to; meanwhile, this procedure continues exhaustively as illustrated in Figure 11. Thus, this depth-first search compiles a list of all elements in a forward path from the start element (i.e. a root cause). Although, the algorithm systematically searches the connectivity matrix of the process; however, the search must terminate with either of the two conditions indicates as follows. The first condition requires that the element next to the point of termination has already been previously visited, thus indicating that the physical paths from this point are already identified. The other termination condition requires that no forward physical connections exist from a point where the search terminates (Thambirajah et al., 2009). At this point, the algorithm retracts to follow other branches after the termination of one path, using the same termination conditions systematically. Figure 11 illustrates the described procedure, where a search begins with Element-006 as a root cause and a compiled list of forward

path elements is as follows: Element-006, Element-005, Element-007 and Element-001. Here, the search terminates at Element-001 since there is no forward physical connection from that point. Hence, the algorithm retracted partially, where it then follows another branch beginning from Element-007 and terminating at Element-004.

The search algorithm establishes the fault propagation path by checking if each element affected by fault belongs to the list of paths compiled by the algorithm. Thus, a propagation path from a hypothesised root cause to an affected element can be established if the affected element belongs to the compiled list of paths. Contrarily, a root cause hypothesis is not valid if the affected element does not belong to the list of paths compiled by the algorithm; this is because a fault at the hypothesised root cause could not have reached the affected element. Meanwhile, the existence of a fault propagation path is thereby established, however the specific path of propagation is not confirmed here. (Thambirajah et al., 2009)

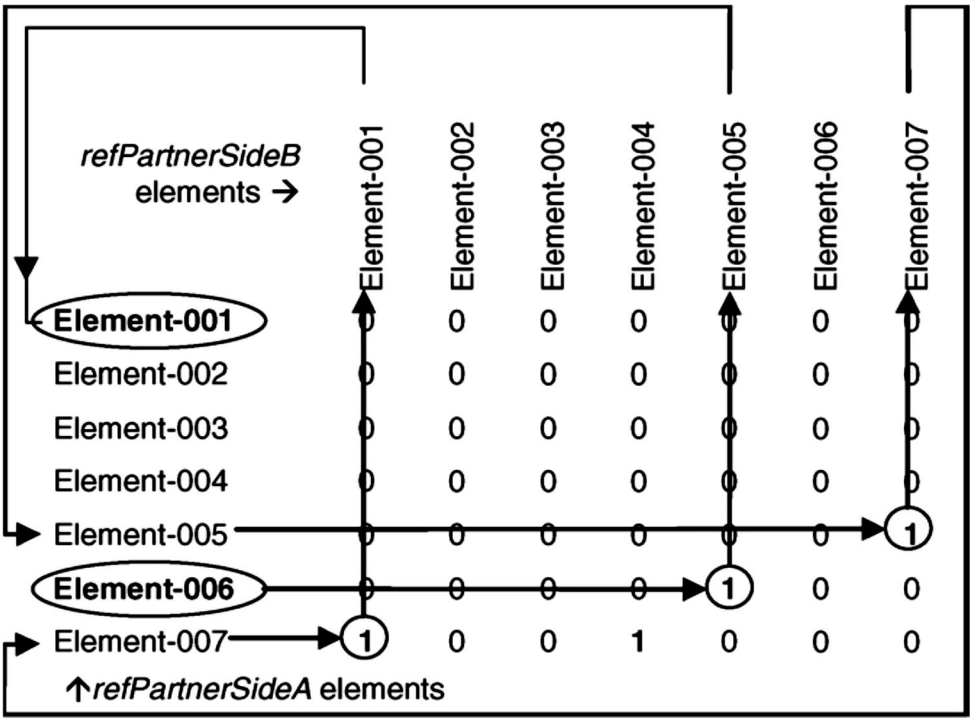


Figure 11: Depth-first search traversal, used in the propagation path search algorithm (Thambirajah et al., 2009)

Identification of the precise path of propagation follows the search procedure described above. The search procedure generated a compiled list of all elements in any forward path, a similar technique is hereby utilised for determining the precise propagation paths; although in this approach, the search procedure begins from an affected element. Furthermore, this search procedure begins from the column of the connectivity matrix representing an affected element, since the affected element represents an end point (Thambirajah et al., 2009). The procedure starts by searching for a matrix input "1", indicating that this column element is physically connected to the row element at the intersection. This is then followed by checking to verify that the row element belongs to the list of forward path elements previously compiled using the earlier depth-first search. Hence, the search is thereby traversing backwards along a propagation path if the row element is verified to belong to the list of forward path elements (Thambirajah et al., 2009). After this verification, the search continues to the column of the connectivity matrix which represents that element, thereby advancing the search in the same way until it arrives at the hypothesised root cause. A precise propagation path is hereby identified if this backwards search successfully arrives at the hypothesised root cause (Thambirajah et al., 2009). Finally, the cause-and-effect analyser tool records the total number of feasible propagation paths, including the shortest path alongside its length for further analysis. This is possible since the algorithm searches for the feasible paths from all the affected elements to all the hypothesised root cause elements within the connectivity matrix.

EXPERIMENTAL PART

4 AIM OF THE EXPERIMENTAL PART

A major challenge often to be encountered in fault diagnosis is generation of spurious results when attempting to locate the actual root cause of a fault and/or a disturbance within a process. Although, several enhanced diagnostic methods have been implemented over the recent decades, the improvement of the performance of the diagnostic systems has not increased remarkably. The impacts of the drawbacks of the current fault diagnosis systems (e.g. diagnosis during unplanned shutdowns) are reported extensively in recent publications (Jämsä-Jounela et al., 2013); also considering the typical adoption of multiple complementary fault diagnosis methods in industrial processes. Also, Jiang et al. (2009) reported that the adjacency matrix method was combined with the spectral envelope method to locate the root cause of a fault; nevertheless, the results produced from such practices were occasionally imprecise and uncertain. Therefore, a new approach that can verify the results acquired from the data-driven causal analysis methods, using the equipment connectivity information extracted from topology details of the process could offer better diagnosis results. Specifically, a bottom-up approach, preferably one which can provide a visualization of the propagation paths of a fault displayed on the process monitoring interface is needed.

Hence, the experimental part of this thesis aims at establishing a technique which extracts the connectivity data of a process from the electronic P&ID flowsheet, for verification of resulting data acquired from a data-driven fault analysis method. Specifically, the topology data of the process in the form of a connectivity matrix is used for validation of the obtained causality matrix from data-driven causal analysis. Furthermore, the validated causality matrix must be transferred into the database of the automation system; subsequently, it will be used as input data in the process monitoring system, where visualization of the fault propagation paths will be enabled. AutoCAD P&ID will be used to create and capture process topology details; the generated topology data was then used to obtain causal models by instantiating the methods of a predefined class in Matlab OOP (Sun, 2013).

The goal of the experimental part of this thesis is to develop a system that generates a connectivity matrix from the electronic P&ID flowsheet of a process, further utilizing it for validating a causality matrix, which is transferred as input to enable visualization of the paths of fault propagation in the process monitoring interface. Thus, a process flowsheet will be created in P&ID PIP standard using AutoCAD P&ID software in order to ultimately obtain a causal model. There are two types of causal model that can be obtained from a process flowsheet, namely a connectivity matrix and causal digraphs (Sun, 2013). They are considered a numerical representation and a graphical representation of a process flowsheet respectively. Henceforth, in this thesis, the connectivity matrix is the causal model type that will be generated and further utilized.

The following sequential steps describe the action plan for the experimental part of this thesis as illustrated in Figure 12. Firstly, the flotation cell process is chosen as an industrial process to be experimented; a description of the flotation process and the flotation pilot plant will be presented as well. Secondly, the AutoCAD P&ID drawing software shall be selected as the choice of electronic P&ID for creating a process flowsheet; additionally, the drawing data of the process flowsheet is extracted for further processing. Thus, a topology-based causal model of the flotation pilot process in the form of a connectivity matrix will be generated, using the extracted data and the procedure presented in Sun (2013). Thirdly, this thesis implements the Time domain Granger causality method, from which a causality matrix is acquired and further utilized in identifying fault propagation paths. Hence, the connectivity matrix is used to refine the causality matrix, which is then transferred to the database of the ABB automation system. Next, the process monitoring system for the flotation pilot process is designed using the ABB graphics builder tool. Therefore, the causality matrix previously transferred into the automation system is used for the visualisation of fault propagation paths within the process monitoring system. Finally, the system will be tested by performing experiments at the flotation pilot plant.

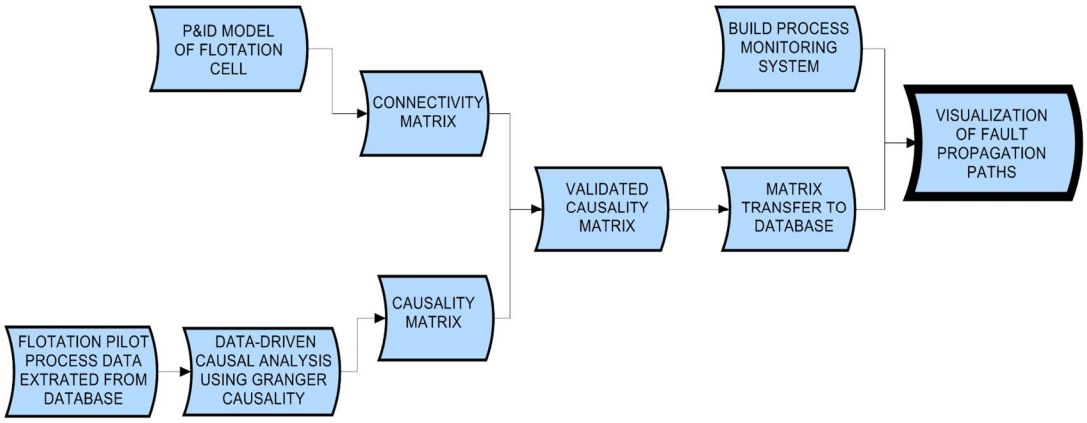


Figure 12: Overall plan of the experimental part of the thesis

5 FLOTATION CELL PROCESS AND ITS CAUSAL MODEL

Complex connections among process equipment and units as well as interactions between the control loops enable and amplify the propagation of faults in industrial processes. Several fault diagnosis tools are used today in the process industry as an attempt to identify the correct root cause of a fault from many candidate root causes. However, they are often reported to produce spurious results in the fault diagnosis efforts. Moreover, there has been little research on methods which can verify the results produced by these diagnostic tools. Therefore, a better approach which utilizes a topology-based causal model for an enhancement of a causality matrix obtained from a data-driven diagnosis method is greatly needed. The topology-based causal model can help to improve the hypothesis of the causal relationship obtained from the causality matrix. These topology details can help confirm and reduce the number of candidate root causes suggested from the causality matrix. Furthermore, this approach can verify the propagation path of the fault and disturbance in a process, since the connectivity matrix checks the physical connections between the faulty elements against the hypothesis from the causality matrix. Finally, the correct root cause can be verified from among all root cause candidates (Schleburg et al., 2013; Thambirajah et al., 2009).

This Chapter presents the approach of creating a causal model from the flowsheet of a process in the form of a connectivity matrix, and further using it to refine a causality matrix for verified fault propagation paths identification. The first section of this Chapter presents a description of the flotation cell process as a case study in this thesis. The second section describes the creation of an electronic P&ID from the flotation process using the AutoCAD P&ID software. It further describes the extraction of process topology data from the P&ID and the creation of the causal model in the form of the connectivity matrix.

5.1 Description of the flotation process

In this section, the flotation cell process description is presented as a case study in this thesis. The case study is presented in order to properly apply the proposed method.

That is, combination of the causal model of a process with the outcome of the data-driven analysis, for the verification of the fault propagation paths. The control strategy of the process is presented as well; the aim of controlling some of the key variables in the process is further explained. Firstly, a general introduction to the flotation cell as well as its characteristics and benefits to the mineral industry is presented. Secondly, the following subsection provides a description of the floatation cell pilot plant, located at the process control laboratory of Aalto University. Finally, this thesis describes the requirements of applying the specific tools and methods upon the flotation pilot process for fault diagnosis research.

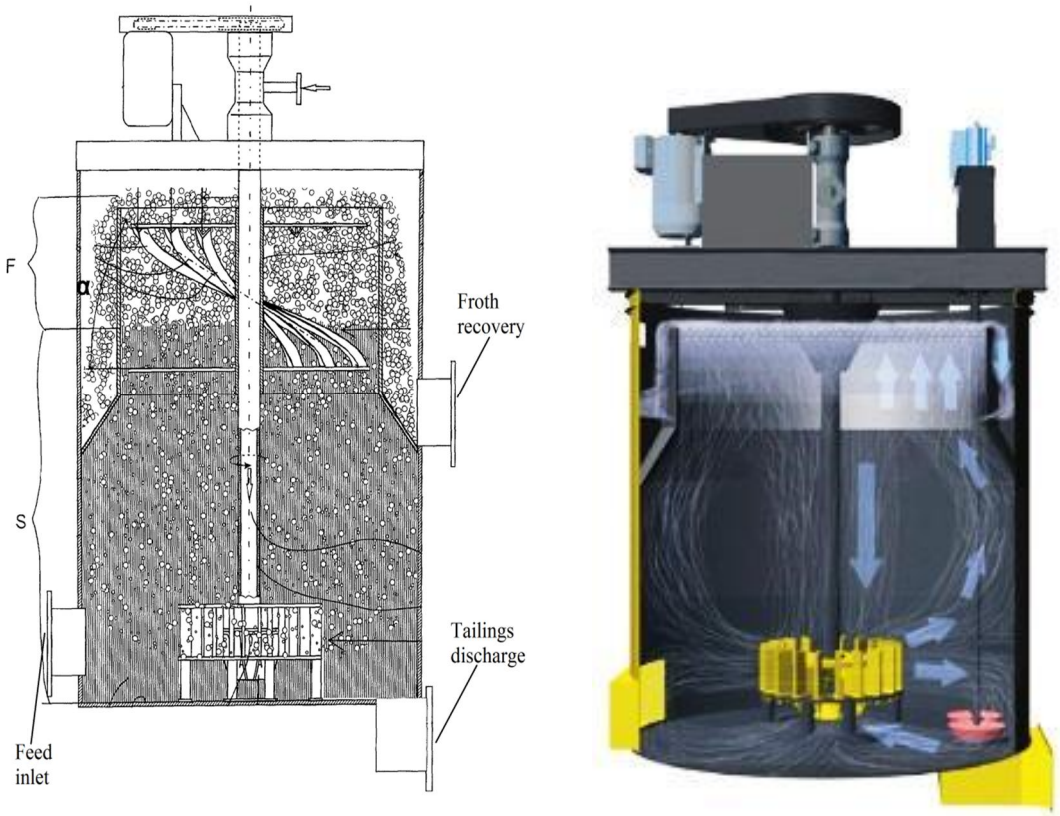


Figure 13: A flotation cell; (a) equipment sketch (Kujawa, 2012), (b) equipment picture (Outotec, 2014b)

A flotation cell is one of the processes which is used to extract the valuable minerals contained within metal ores. Additionally, the flotation process is a physico-chemical

separation process which exploits the contrast in the surface properties of the valuable minerals and the unwanted gangue minerals. This is accomplished by using the froth flotation as a means of collecting the valuable minerals from the pulp upwards, as it can be seen in Figure 14. Wills and Napier-Munn (2006) reported the theory of froth flotation to be complex and involving three phases (water, solids and froth), including several subprocesses and interactions. The attachment of the valuable minerals to air bubbles is the most important mechanism representing the bulk of the collected particles to the concentrate. However, most minerals are hydrophilic in nature, meaning that the attachment to air is impossible. Therefore, some chemical substances known as flotation reagents, such as collectors are added to the pulp, to enable the attachment of the fine particles to the air bubbles. Likewise, some other reagents called frothers are added in order to decrease the surface tension, while also making the froth coating on the surface strong enough to bear the mineral particles and to prevent the burst of the air bubbles (Stenlund and Medvedev, 2002; Wills and Napier-Munn, 2006). Additionally, pressurized air is typically blown into the tank through the shaft of the mixing agitator. The action of the agitator provides sufficient turbulence in the pulp to promote the collision of the particles and the bubbles. This causes the attachment of valuable mineral particles to air bubbles, and leading to their movement into the froth phase (F region) for extraction, as seen in Figure 13a. Figure 14 illustrates the upward drifting of the particles-attached air bubbles to form the mineralized froth at the top of the flotation cell.

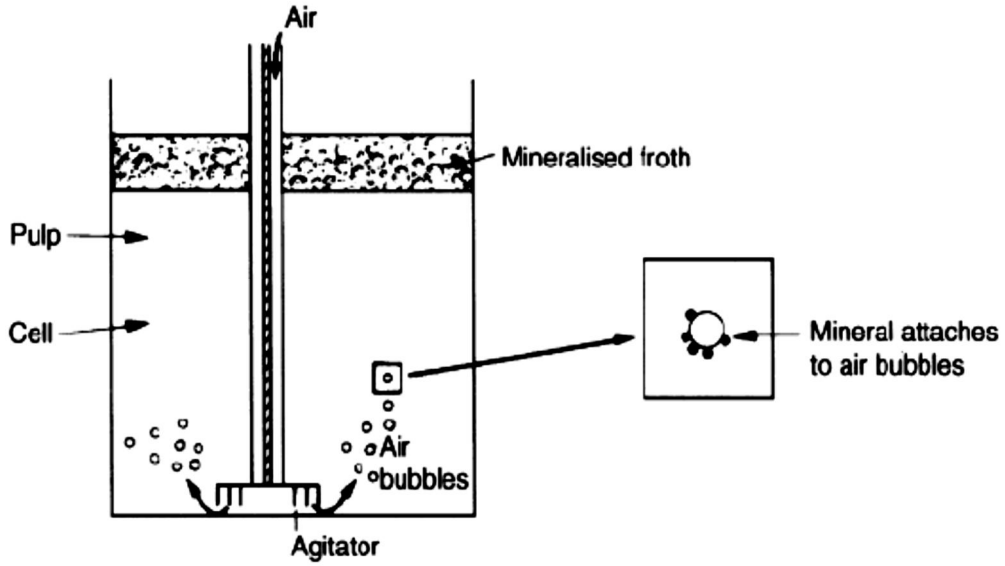


Figure 14: Mineral separation mechanism in a flotation cell tank (Wills and Napier-Munn, 2006)

The most typical flotation tanks in the mining industry have volume ranging from 120 m^3 to 150 m^3 and they are normally set up in series, forming a bank or a farm in a plant. However, a much larger size family of tanks have been developed for the much higher throughputs and economic benefit. Recently, Outotec Corporation has introduced the e500 flotation cell, with the active volume of 500 m^3 as the largest member of its TankCell® product series, which is arguably the world's largest commercial single flotation tank (Outotec, 2014a). Table 2 shows an example of the number of banks selection in a plant as a function of the cell size and the total pulp volume. The cell size of 75 m^3 was reported to produce the best balance between costs (energy and capital) and operational flexibility. Hence, a series of the flotation tanks constitute a bank in commercial flotation operation; banks of the flotation cells are operated as continuous processes. The configuration of the single cells series allows the flow of the pulp downstream between cells to be controlled by the connecting valves, since each next cell is installed at a slightly lower position (i.e. lower elevation), as seen in Figure 15. This differs from the double cells series where two tanks are installed on the same elevation, and a pressure difference between pulps in the two cells is the only driving force of pulp flow through their connecting flanges (Jämsä-Jounela et al., 2003).

Table 2: Factors that determines flotation cell size selection, adapted from (Meech, 2014)

Cell size (m^3)	Residence time/cell (min.)	Number of banks	Number of cells per bank	Total volume (m^3)	Total residence time (min.)
25	0.98	6	12	1,800	11.8
50	0.98	3	12	1,800	11.8
75	0.98	2	12	1,800	11.8
100	1.31	2	9	1,800	11.8
150	0.98	1	12	1,800	11.8
200	1.31	1	9	2,000	13.1
250	1.64	1	8	2,000	13.1
300	1.96	1	6	1,800	11.8

A flotation operation begins with the entry of pulp (water-mineral slurry) through the inlet flange as the feed into the first cell of the bank. A fraction of the valuable minerals are collected from the froth in the first tank as an overflow. Afterwards, the tailings (i.e. depleted pulp) flow to the next cell where more mineralised froth is further collected, and continues in the same approach downstream the bank. Outflow to the next tanks are controlled by discharge valves that are located between the two flotation cells, which partially control the level of material in the preceding tank (Kämpjärvi and Jämsä-Jounela, 2003). Additionally, the froth columns in the first few cells of the bank are kept high, since the proportion of the hydrophobic mineral particles is high at this stage. Henceforth, as the floatable minerals in the pulp diminishes, the pulp level raises from one cell to the next. Consequently, relatively low-grade froth, containing poor aerophilic particles remains in the last few cells of the flotation bank. They typically contain middling particles sometimes referred to as scavengers, some of which normally gets recirculated to the head of the bank as a form of recycling (Wills and Napier-Munn, 2006). Finally, this separation phase completes as the valuable minerals collected in the froth layer of the tanks across the bank are transferred for further processing.

Figure 15: A typical flotation cell tank series forming a bank (Outotec, 2014a)

5.1.1 Flotation cell pilot plant in the Aalto University laboratory

The industry hall of the Chemical technology research laboratory hosts the flotation cell equipment for the laboratory of Process control and automation. The pilot flotation plant offers the research possibilities for monitoring and control of the flotation process. Thus, it comprises the necessary equipment, instrumentation and a distributed control system (DCS). Control of the pilot plant is achieved using the ABB System 800xA industrial IT automation information system, and 800xA PM 856 PLC controller. The configuration of this pilot plant differs from typical flotation cell banks in the mineral industry as described earlier, primarily based on the presence of only two tanks in this pilot plant. Initially, pulp (slurry) feed into the flotation tank comes from the feeding tank installed beside and below the flotation tank. Hence, a sufficient supply of pressurized air is transferred from a blower to the flotation tank through the air-feed pipe and the mixing equipment. In this pilot plant, slurry is pumped upwards to the flotation tank where constant mixing of slurry with compressed air and reagents

takes place. The overflow (froth) is collected from the upper edges of the flotation tank through the overflow pipe, back into the feeding tank below it. Additionally, underflow (i.e. equivalent to tailings) is allowed to flow back into the feeding tank when the level of the flotation tank becomes high. Either of the two exit valves located at the bottom of the flotation tank controls the rate of the outflow to the feeding tank. A picture of the flotation pilot plant is visible in Figure 16b.

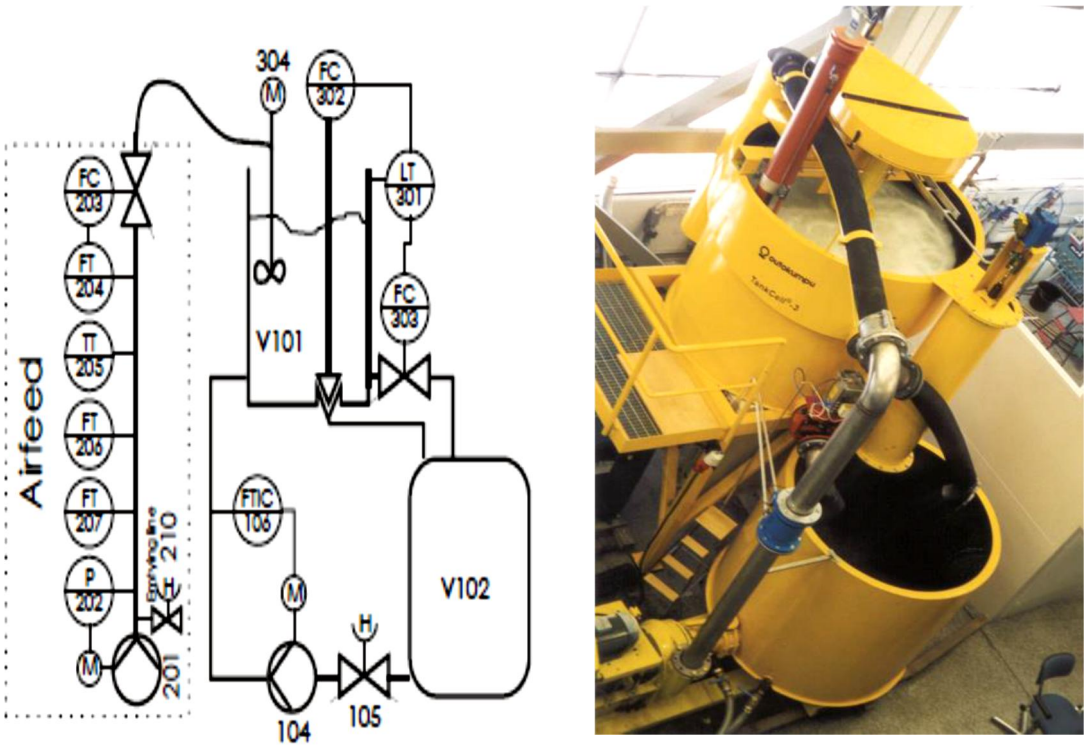


Figure 16: Flotation cell pilot plant at Aalto; (a) P&ID sketch, (b) picture of the process equipment

5.1.1.1 Control strategy of the flotation pilot plant

The pilot-size flotation cell system is equipped with all the necessary instruments, actuators, equipment, and controllers. The main equipment are the flotation tank V-101 and the feeding tank V-102; the flotation tank being located right above the feeding tank as seen in Figure 16b. Both have 3 m^3 of active volume. An agitator shaft

is located right in the middle of the flotation tank with a driving electric motor M-304, which drives the actuator (mixer) and also provides the channel for introducing pressurized air into the slurry. Compressed air is fed into the slurry in the flotation tank from the Aerzen blower (compressor) M-201, which can be speed-controlled. A number of instruments are installed along the air-feed line, from the Aerzen blower to the flotation tank. A speed-controlled electric centrifugal pump M-104 enables transfer of slurry along the primary line from the feeding tank to the flotation tank. The instrument installed on the primary slurry feed line is a flow transmitter and control element FTIC-106, meanwhile a hand valve H-105 is installed between the feeding tank and the slurry pump M-104, as seen in Figure 16a. Level of material in the flotation tank is measured using a level sensor/transmitter LT-301; hence, its output signal is further used for controlling the level of the tank. That is, the tank level is controlled using either of the positioner FC-302 or FC-303 of the Dart or the Larox exit valves respectively. These exit valves provide the possibility for the underflow of slurry from the flotation tank back to the feeding tank below it, either of the Dart and Larox valves can be used for this purpose.

The air-feed line between the Aerzen compressor and the flotation tank hosts a few instruments, such as the flow transmitter FT-207, the temperature transmitter TT-205 and the control valve CV-203, as well as a pressure relief hand valve H-210. These instruments enable a suitable supply of compressed air into the slurry in the flotation tank through the shaft and the mixer. Finally, although some instruments connected in the pilot plant such as the flow-meter FT-206 and the pressure probe P-202 are visible in Figure 16a; however, these instruments have remained unused in operating the process. A key factor during operation of the flotation process presented earlier in this Chapter is: the level control of a flotation cell. This is achieved by adjusting the underflow, i.e., through the manipulation of either of the Dart or Larox control valves from the flotation tank to the feeding tank.

The monitoring and control of the flotation pilot process just as other processes possess some challenges. Occurrence of fault and disturbance limits the ability to effectively control the flotation pilot process. The spread of a fault and disturbance from its source to other locations within the process further complicates the challenges in controlling

the process. Thus, this thesis seeks to implement the tools and methods proposed in earlier Chapters for achieving an improved diagnosis of faults in the flotation process. A classification of variables in the flotation pilot process based on their roles in monitoring and control of the process is presented in Table 3 below.

Table 3: A classification of the flotation pilot plant process variables

Variable	Measuring/monitoring	Manipulated	Controlled
LT-301	*		
FC-302		*	
FC-303		*	
M-104			
FTIC-106	*	*	
M-304		*	
M-201			
FT-207	*		
TT-205	*		
FC-203		*	

5.2 Creation of a causal model from the process flowsheet

In this thesis, process topology details are used to obtain a connectivity matrix, which is later utilized in the graphical display of possible fault propagation paths in a process monitoring system. Therefore, an electronic P&ID of a process is created, from which the process topology data is subsequently generated, in order to create a causal model. The data to be extracted from the AutoCAD P&ID will be exported in the Microsoft Excel XLS/XLSX file format, which contains the tag names of the process variables and the physical relationship among them in a numeric form, i.e., a connectivity matrix.

5.2.2 Representation of the flotation pilot process in AutoCAD P&ID

The tasks to be performed in the creation of the P&ID of the flotation process in the AutoCAD P&ID software are as follows: insert the components (i.e. process equipment), add the connecting lines, add tags, validate the drawing and export the drawing data. Firstly, the main equipment of the process, a feeding tank TK-102 and a flotation tank TK-101 are added to the empty drawing area one at a time; by selecting a dome-roof tank from the equipment tab of the tool palette and clicking on the preferred location in the drawing area where the equipment is positioned. Secondly, a slurry pump P-104 is placed in a position next to the feeding tank and below the flotation tank and the pump is further rotated to the correct orientation. Thirdly, a centrifugal compressor is placed in the drawing to represent the speed-controlled air compressor, which is the source of air into the flotation cell process; its position is on the far left side of the pump. Then, the last process equipment to be added to the drawing is the flat-blade paddle agitator, which represents the motorized mixer and is positioned in the centre of the flotation tank. Additionally, the sizes of all the added equipment are adjusted to their respective scales.

The next phase of the P&ID design is to add flowsheet lines which are the connecting pipes between the process equipment as presented in Figure 4. In the flotation cell process, the slurry pipes enable material flow between the two tanks; additionally, the air pipe enables the flow of air from the compressor into the flotation tank. Hence, the first pipe line is created in the P&ID drawing by selecting the primary line symbol from the lines tab of the tool palette and first clicking on the flotation tank TK-101, then clicking on the feeding tank TK-102 to complete the connection. AutoCAD P&ID automatically adds nozzles to the points where the line is connected to two tanks in order to indicate and verify the presence of connections at those spots. The second drain pipe line is created in the drawing also by selecting the primary line symbol from the tool palette and first clicking on the flotation tank TK-101, then clicking on the feeding tank TK-102 to complete the connection. The second exit line is an alternative drain pipe line that leads material from the flotation tank to the feeding tank. Next, a primary line is added to the drawing that connects the feeding tank TK-102 to the entry point of the slurry pump P-104. Then, another primary line is added to the drawing

that connects the exit point of the pump to the flotation tank TK-101. Nozzles are automatically created also at the connecting points between the pipes and the tanks. Finally, the last primary line to be created in the drawing is the air flow line from the compressor C-201 to the flotation tank TK-101. Furthermore, the drawing is validated as suggested in Figure 4, in order to highlight any possible errors at this stage. The validation process detects and highlights property mismatches among components and lines in the drawing and identifies any items that do not conform to the chosen standards (Autodesk, 2013).

A few components of the process that are located along the pipe lines, referred to as inline components are added to the drawing as suggested by the design workflow. Firstly, a hand valve is selected from the valves tab of the tool palette and placed along the primary line that connects the feeding tank TK-102 and the slurry pump P-104. The valve automatically takes the same diameter of the pipe line and requires no nozzles to acquire a physical connection along the pipe line. Secondly, a three-way valve is selected from the tool palette and placed on the primary line that connects the compressor C-201 and the flotation tank TK-101. This three-way valve is necessary here since it is the required logic for creating a pipe branch from a pipe line. Thus, a primary line is created from the second outlet of the valve to make a complete connection. Finally, a hand valve is placed at the end of the newly created pipe to complete this stage of the P&ID drawing.

Instruments of a process play a key role in the fault propagation path analysis. The connectivity matrix to be generated must not only contain connections among the components of the process, but also among the instruments and other components. Faults can propagate also through the instruments and along the information pathways, i.e. not only along the material pathways. Hence, the first instrument is created by selecting FIC from the instruments tab of the tool palette and clicking on a space between the slurry pump P-104 and the slurry pipe line. Accordingly, the instrument is tagged as FIC-106, since it is a FIC type instrument and it is located in the area 100. It is worth mentioning at this stage that as each instrument is created in the drawing, signal lines (dotted lines) are created right away to connect the newly created instruments to their respective equipment and/or other instruments. Additionally, the

tag (unique identification) for each instrument is also included as each instrument is being added to the drawing, since it is a part of the symbol (sign) of each individual instrument in the drawing. Likewise, other instruments including TT-205, FT-207, LT-301, FC-302 and FC-303, are selected from the tool palette, placed in the respective locations, tagged for unique identification and connected to their respective equipment via signal lines. Then, control valves are the last instruments to be added to the drawing; the first one is selected from the instrument tab of the tool palette, placed on the air flow primary line and tagged 2-CV-209 immediately. Finally, two more control valves are likewise added separately to the two exit pipe lines that connect the flotation tank TK-101 to the feeding tank TK-102; they are 3-CV-305 and 3-CV-306 respectively.

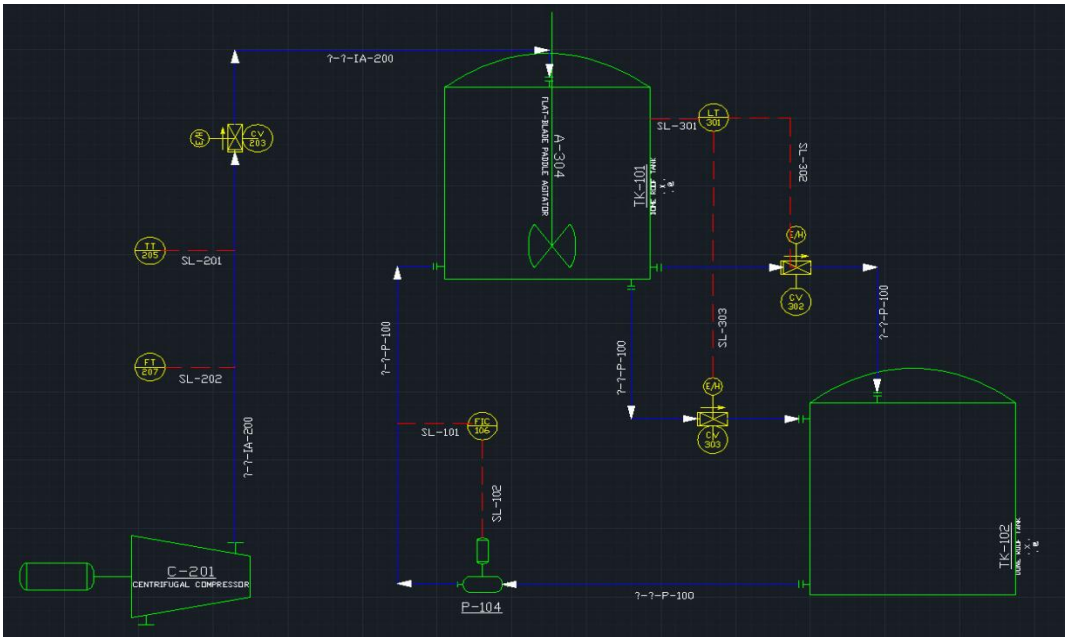


Figure 17: The P&ID model of the flotation pilot process in AutoCAD P&ID

According to the suggested workflow, tags and annotations are added to the elements of the P&ID drawing. There are different alternatives for creating and placing tags and annotations on the elements of a drawing in AutoCAD P&ID; however, clicking on the “Assign Tag” button in the P&ID ribbon is an obvious alternative. Hence, by

clicking the “Assign Tag” button, the program allows the user to select a P&ID element (component or line) to be tagged, thus a dialogue box opens up which allows a unique name and an annotation style to be allocated to the element being tagged. Therefore, the untagged elements at this stage are given their respective tags as follows. The slurry pipe line group is tagged “?-?-P-100”, while the air flow pipe line is tagged “?-?-IA-200”. The two question marks are present in the pipe line tags above because AutoCAD P&ID requires the material type and diameter to be included in the Assign Tag dialogue box, since this information is not available question mark sign is automatically generated into the tags. Next, the flat-blade paddle agitator is tagged A-304 and the centrifugal compressor is tagged C-201. Then, the hand valve between the feeding tank and the slurry pump is tagged HA-105, while the three-way valve on the air flow line is tagged HA-210 and the valve located at the end of its outlet is tagged HA-208. By opening the Data Manager from the project ribbon, it is possible to view and manipulate the drawing data. Thus, by using the export function in the Data Manager shown in Figure 18, the drawing data is exported to Microsoft Excel where the tags for all the signal lines in the drawing are created and saved with the following tag names: SL-101, SL-201, SL-202, SL-301, SL-302 and SL-303. Subsequently, the import function of the Data Manager is used to transfer the newly created tag names into the P&ID drawing. Other drawing related data can be manipulated within the Microsoft Excel environment; however, in this case, only the tag names for the signal lines were created there. Finally, a complete model of the flotation cell process is now ready and can be seen in Figure 17. A paper-view of the model in larger scale is also available in Appendix A.

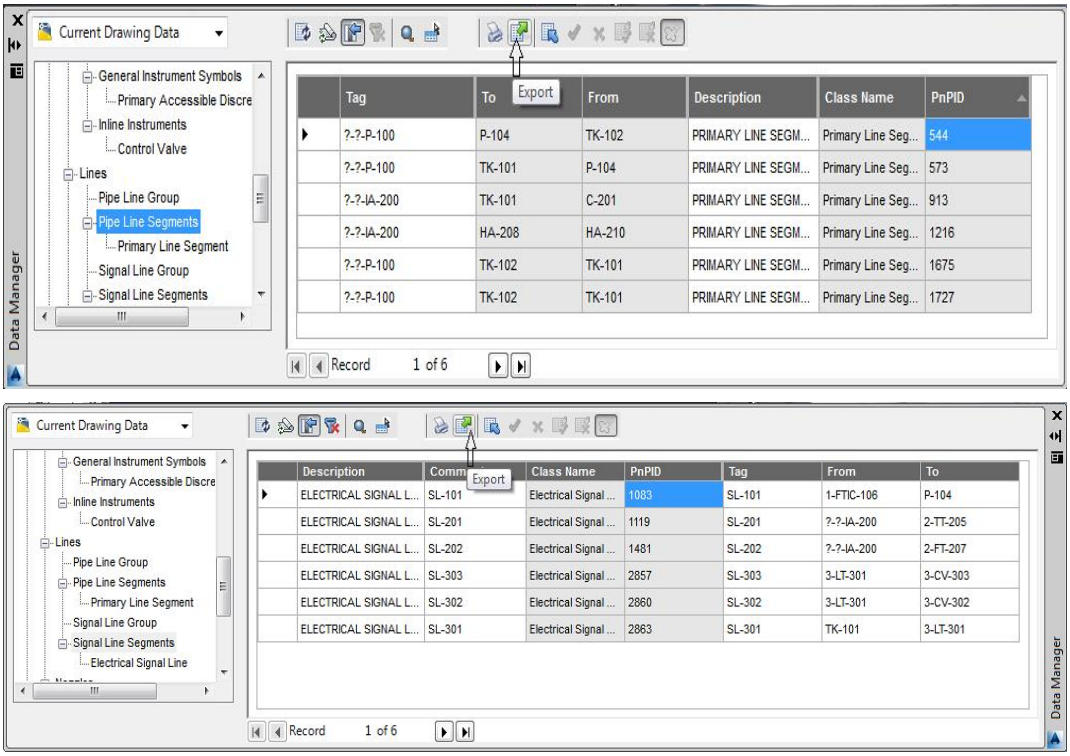


Figure 18: Data Manager Window in AutoCAD P&ID, (top) pipe lines data, (bottom) signal lines data

5.2.2.1 Conversion of the process topology data to a connectivity matrix

The P&ID of a typical industrial process is large; consequently, the size of the topology data acquired from such P&ID is as well large. Therefore, an automated method of converting an exported drawing data from a P&ID into a connectivity matrix is required. Sun (2013) developed a method that automatically creates two types of causal model from the drawing data extracted from a process flowsheet. The method included a Matlab OOP program which takes a Microsoft Excel data file as an input in order to generate a connectivity matrix among other outputs of that program. Henceforth, this thesis will benefit from that method by adopting the already existing program for the purpose of obtaining the connectivity matrix.

The object-oriented programming OOP feature of Matlab software is used to implement the causal model generation program in (Sun, 2013). The program consists

of a class file and an instantiation file; the structure of the class comprises properties and methods, where methods are the functions that can be performed on each instance of the class. Thus, this conforms to the characteristics of object-oriented programming. Additionally, the instantiation file contains all the Matlab OOP commands which are required for instantiating the class, i.e. define an object of that class; further, the methods of the class are invoked (e.g. get connectivity matrix) in the instantiation file. Therefore, all the three necessary files: the class file, instantiation file and the input file (topology data in Excel) are saved in a one folder. The following tasks describe the procedure for obtaining a connectivity matrix from the causal model program of (Sun, 2013): Firstly, the class file is modified such that the appropriate sheet is automatically selected from the Excel file, i.e. the sheet that contains the "From" and "To" data of all pipe and signal lines in the process flowsheet. Secondly, the precise file name of the input file (topology data in Excel) is inserted as the argument of the ".getLabel" method, for both pipe and signal lines respectively inside the instantiation file. The sheet position numbers of the pipe lines and signal lines data inside the Excel file are also inserted as second arguments while invoking their respective methods. An additional command (.getMatrix()) is subsequently included in the instantiation file to extract the connectivity matrix of the object. Thirdly, execution of the instantiation file follows next; in the absence of any run-time errors, the newly created connectivity matrix of the flotation cell process is available in the Workspace of Matlab command window. Finally, the connectivity matrix is contained in a spreadsheet array as seen in Table 4a, which contains zeroes and ones representing the physical directional connections among the equipment and instruments of the flotation process.

Table 4(a): Connectivity matrix of the flotation pilot process

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	[]	'P-104'	'2-CV-203'	'2-FT-207'	'2-TT-205'	'3-LT-301'	'1-FIC-106'	'3-CV-302'	'3-CV-303'	'C-201'	'TK-102'	'TK-101'	'?-?-IA-200'	'?-?-P-100'
2	'P-104'	0	0	0	0	0	0	0	0	0	0	1	0	0
3	'2-CV-203'	0	0	0	0	0	0	0	0	0	0	1	0	0
4	'2-FT-207'	0	0	0	0	0	0	0	0	0	0	0	0	0
5	'2-TT-205'	0	0	0	0	0	0	0	0	0	0	0	0	0
6	'3-LT-301'	0	0	0	0	0	0	1	1	0	0	0	0	0
7	'1-FIC-106'	1	0	0	0	0	0	0	0	0	0	0	0	0
8	'3-CV-302'	0	0	0	0	0	0	0	0	0	1	0	0	0
9	'3-CV-303'	0	0	0	0	0	0	0	0	0	1	0	0	0
10	'C-201'	0	1	0	0	0	0	0	0	0	0	0	0	0
11	'TK-102'	1	0	0	0	0	0	0	0	0	0	0	0	0
12	'TK-101'	0	0	0	0	1	0	1	1	0	0	0	0	0
13	'?-?-IA-200'	0	0	1	1	0	0	0	0	0	0	0	0	0
14	'?-?-P-100'	0	0	0	0	0	1	0	0	0	0	0	0	0

Table 4(b): The list of all connecting line segments in the P&ID

	1	2
1	TK-101	3-CV-303
2	3-CV-303	TK-102
3	TK-101	3-CV-302
4	3-CV-302	TK-102
5	C-201	2-CV-203
6	2-CV-203	TK-101
7	P-104	TK-101
8	TK-102	P-104
9	?-?-IA-200	2-FT-207
10	TK-101	3-LT-301
11	3-LT-301	3-CV-303
12	3-LT-301	3-CV-302
13	?-?-IA-200	2-TT-205
14	?-?-P-100	1-FIC-106
15	1-FIC-106	P-104

This connectivity matrix generated here will be beneficial for analysis in a fault diagnosis problem of the flotation process. Specifically, it will be utilised for the refinement of a causality matrix which will be generated from the analysis of data acquired from the flotation process.

6 PROCESS MONITORING SYSTEM AND DATA ANALYSIS

Process monitoring systems for industrial processes play an essential role in achieving desirable supervisory control during normal and abnormal situations. A process operator must have a proper visualization of process conditions through the monitoring systems especially during an abnormal event. However, the effects of a fault are normally observable far from the origin; thus, making it more challenging for plant operators to verify the paths of fault propagation and to take corrective measures. This is further complicated by complex interconnections of equipment, instruments and control loops in typical industrial processes. Although, most of the process monitoring systems used today by operators provide good view of process conditions, and some of the downstream effects of a fault can also be observed through the monitoring interfaces. These monitoring systems are not designed to include visualization of paths and the manner by which faults propagate from an origin to other locations in industrial processes. Thus, providing plant operators the visualization tool of fault propagation paths from any origin within a process is needed.

This Chapter describes the creation of the process monitoring system and the display of fault propagation paths for the flotation pilot process. In the first section, data acquired from the flotation process are used to generate a causality matrix, which is further refined in order to verify the propagation path and a root cause of the fault in the process. The second section, describes the design of the process monitoring system and the scheme designed to enable transfer and display of the fault propagation path in the interface of the process monitoring system.

6.1 Identification of a fault propagation path with the causal analysis

The flotation pilot process at Aalto University laboratory is the case study in this thesis; henceforth, experiments will be carried out in the pilot process to demonstrate the benefits of enhanced fault propagation paths identification. To display the fault propagation paths in the process monitoring interface, data from the process must be analysed to obtain a causality matrix. Therefore, the pilot process must be operated for

a number of hours in order to produce a sufficient quantity of measurements for causal analysis.

A successful experiment was performed on the 18th of September 2014 where water was used in place of slurry and air was supplied into the flotation tank. The operation was monitored using the process monitoring system (see appendix B), while using a trends aspect in System 800xA available in appendix C to view and extract measurements from the pilot process operation. Furthermore, ten variables were measured from which data was extracted for analysis. The key parameters in the pilot process operation are the flotation tank level setpoint at 475 mm using the Dart valve and the liquid flowrate setpoint at 1.50 m^3/min using the pump M104. Although, the operation lasted about five hours, the data eventually analysed was selected from about four hours of operation. A level sensor fault was then introduced halfway into the process for a few minutes. The upset and eventual restoration can be seen in the trends window (see appendix C). Finally, the data from process measurements was logged and extracted from the database of the System 800xA for the data analysis purpose.

The experiment aims to identify the propagation paths of the fault introduced into the process operation described above. A causality matrix containing the candidate root causes and the propagation path of the fault must be generated from the process data. Hence, the Time domain Granger causality is selected as a causal analysis method of creating the causality matrix from the process data, where seven out of ten variables are used in the analysis. Furthermore, since a connectivity matrix of the flotation pilot process was created in this thesis, it is thus used for refining the causality matrix generated here. Landman et al. (2014) described a procedure for creating a causality matrix using Time domain Granger causality and further refining it using the connectivity matrix of the process. Therefore, this procedure is implemented in this thesis to generate the causality matrix in Table 5a, and to ultimately obtain the refined causality matrix in Table 5b.

Table 5(a): Causality matrix generated from the Granger causality method

Initial GC matrix							
	LT301	FTIC106	TT205	FT207	M104	M201	Dart_Back
LT301	0.080659	0.335898	0	0	0.296704	0	0.564525
FTIC106	0.2034	0.767593	0	0.066254	0.730609	0	0.13356
TT205	0	0	0.191514	0	0	0	0
FT207	0	0	0	0.458454	0	0	0
M104	0.520659	0.644326	0	0	0.727407	0	0.335642
M201	0	0	0	0	0	0.393561	0
Dart_Back	0	0.55429	0	0	0.345964	0	0.079553

Table 5(b): Causality matrix refined using the connectivity matrix

Refined causality matrix							
	LT301	FTIC106	TT205	FT207	M104	M201	Dart_Back
LT301		0	0	0	0.296704	0	0.564525
FTIC106	0.2034		0	0	0.730609	0	0.13356
TT205	0	0		0	0	0	0
FT207	0	0	0		0	0	0
M104	0.520659	0.644326	0	0		0	0.335642
M201	0	0	0	0	0		0
Dart_Back	0	0	0	0	0.345964	0	

Although, it is known that LT301 (representing the faulty level sensor) is the root cause of the fault; however, the propagation of that fault is not accurately represented in the initial causality matrix. Therefore, after refining this matrix using the connectivity matrix, it can be seen in Table 5b that the affected elements are correctly and realistically revealed in the resulting refined causality matrix. Finally, the refined causality matrix obtain here is numerical, thus enabling it to be easily transferred and stored in the database of System 800xA, where it can be used for displaying the propagation paths of the fault in the process monitoring system.

6.2 Process monitoring systems and data transfer scheme

The design of a process monitoring system for the flotation pilot process and the creation of a technique to transfer data from external sources into the database of the automation system constitute a part of the goals in this thesis. Specifically, the refined causality matrix acquired from the data analysis must be transferred from an external source to the database of the automation system. Furthermore, a framework will be created in the database to enable the retrieval of the data (i.e. the refined causality matrix), as well as to update the data with a newly acquired refined causality matrix. In addition to designing a process monitoring interface, essential program codes must be written inside the elements of the monitoring interface; these elements are the individual components that constitute the entire monitoring interface. The programs codes for the individual elements are composed to enable a dynamic transformation of elements colours, leading to the actual visualization of fault propagation paths in the monitoring interface. The tasks in this section will be accomplished within the environment of ABB Industrial IT System 800xA. ABB offers System 800xA as its flagship collaborative process automation system (CPAS), thus extending this Industrial IT framework well beyond the scope of traditional DCS. Accordingly, the "xA" in the name System800xA refers to "extended automation", which enables a highly integrated automation system. System800xA is based on "aspect object" technology, where every plant data (i.e. aspects) are linked to their respective plant assets (i.e. objects), leading to a seamless integration of information plant-wide irrespective of its source (ABB, 2014).

This section describes the steps in creating the process monitoring interface within the environment of System 800xA. It also describes the approach used to implement a strategy of transferring data from external source into the database of System 800xA. Thus, the following subsection describes the creation of the process monitoring interface; further describing the required program codes written to enable the visualization of the propagation paths. Subsequently, the succeeding subsection explains the approach used in this thesis to transfer and access data coming from the external source to the database of System 800xA.

6.2.1 Design of the process monitoring system

System 800xA automation solution is composed of two sections: the project explorer in the Control Builder M section and the Plant Explorer section. Most of the tasks involved in creating a process monitoring interface take place in the plant explorer section, which is structured in a hierarchical arrangement, i.e., in a tree structure with subdivisions. “Functional structure” is one of the structures contained in the plant explorer; multiple plants (objects) are created and explored in this structure. Hence, a new site object is created; then, a new “Liquid Processing Cell” is created inside the plant, named “Flotation Cell”. This newly created object represents the flotation pilot process for which several aspects can be created afterwards. Under the functional structure, the Flotation Cell object is selected and using the context menu (i.e. mouse right-click), a new aspect for that object can be created. Inside the “New Aspect” window, the aspect type is chosen to be a “Graphics Display” aspect and named “Flota_cell_disp_thesis”. Hence, this newly created aspect appears in the list of aspects associated to the Flotation Cell object in the functional structure. Building the graphics display begins by selecting “Edit” from the context menu of this new aspect; it hereby opens the Process Graphics Editor application of System 800xA, where a process monitoring interface is built.

The Process Graphics Editor application starts with a blank editing space, with the menu bar at the top of the window and the toolboxes panel on the left side of the window. The properties panel is docked on the right side of the window; all the graphics elements to be added to this monitoring interface have properties, which are to be modified and programmed through the properties panel. Hence, the first graphic element is added by opening the “special” toolbox from the toolboxes panel; then, selecting “Symbol Factory Symbol” from the list of tools, thus an equipment symbol appears by clicking on the empty editor space. A basic pump, belonging to the pumps category of equipment is the default “Symbol Factory Symbol” tool; hence, it appears on the editor space by clicking once. At this point, the equipment is modified to a preferable equipment symbol by selecting “Edit” from the context menu of the symbol. This opens up the symbols factory properties window; thus, by selecting “Tanks” from the list of equipment on the left side of this window, a wide variety of tanks used in

the process industry appears in the window. Tank type “Tank-1” is selected to represent the flotation tank in the flotation pilot process. This step completes the task of creating the first graphics element of the “Flota_cell_disp_thesis” process operator interface. Therefore, the above described steps are followed in order to create the remaining equipment and instruments of the flotation pilot process, which will be represented by their respective graphics elements. Specifically, the other equipment and instruments to be created using this same procedure are: feeding tank, slurry pump, air-flow valve (203), flotation side column (right), Larox control valve (303), flotation side column (left), Dart control valve (302), flow meter/controller (106), compressed-air blower, mixer and air-flow transmitter (207) respectively. Once the equipment and instruments are created using the previously described procedure, they are moved to their respective positions, to represent the topology of the actual flotation pilot process. An exception to the above-described procedure is the creation of the level sensor (LT-301), which is created using a different approach described as follows. The “Special” toolbox is selected from toolboxes panel, “Symbol Factory bar” is then selected from this toolbox by clicking once. Thus, by clicking on the editor space, a level sensor/indicator symbol appears, which is resizable and movable as necessary.

At this stage, the equipment and instruments are arranged in a suitable order inside the process graphics editor, it is now required to connect them together using pipes. Pipes are created by opening the “Shapes” toolbox from the toolboxes panel; thus, selecting the “Flexible Pipe” from the list, then by clicking on a source element and double-clicking on a destination element, a pipe now appears to connect two graphics elements. The above described procedure is utilized several times to create all of the necessary connections among the equipment and instruments. Furthermore, the pipes in the air-flow pipeline are given eight units of width and “cornflowerblue” inner-shade colour; whereas, the pipes in the slurry pipeline are given sixteen units of width and silver inner-shade colour. This ensures clear distinction between the air-flow pipeline and slurry pipeline. Electric signal transmission cables connecting instruments and equipment are created using “Polylines”, which can also be selected from the “Shapes” toolbox. The Polylines are drawn to make connections between the instruments and equipment using a similar procedure described earlier for the flexible

pipes. The default values for the polylines are used in these signal transmission cables, i.e., the black colour and one unit of width.

A process monitoring interface must not only graphically display the state of a process during operation and show values of key variables, control actions over the process must also be possible through this monitoring interface. Therefore, addition graphics elements, referred to as faceplates must be created (at relevant positions) within this monitoring interface, to enable a good monitoring and supervisory control of the process. These graphics elements are created using a procedure described as follows. Three individual faceplates are firstly created to enable the on/off control of the compressed-air blower, slurry pump and mixer, using "function block" variables M-201Run, M-104Run and M-304Run respectively, which have been created in the control programs (i.e. control builder application) section. On the contrary to the creation of other graphics elements, faceplates are created by opening the element explorer tab (i.e. next to toolboxes tab) or by accessing the element explorer through the "view" tab in the menu bar. Thus, several "structures", including the control structure, appear in the upper half of the panel. Furthermore, the appropriate control project is selected through the control network from the control structure, while the appropriate object (i.e. the function block variable) is selected from either the programs or the control modules. Once the required object is selected, different graphics element types (i.e. alternatives) appear in the lower half of the panel. Thus, the first faceplate is created for the compressed-air blower by selecting M-201Run from the programs (in the upper panel) and selecting "Display Element Reduced Icon" from the list of alternatives (in the lower panel). Then, the graphics element is drawn by clicking on an appropriate location in the graphics builder editor. The second and third faceplates (on/off control) are created for the slurry pump and mixer using similar steps as in the first faceplate.

Since this process monitoring interface must enable the visualization of fault propagation paths, small text windows that appear by clicking faulty elements of the process will be beneficial to diagnosis of faults. The purpose of displaying these small text windows is to provide further information about the specific process element for the operator while the fault propagation path is being displayed. Thus, a minimised

window with specified helpful text is designed for all the pipe and signal line segments, by creating a “Graphics Display” object for every line segment in the generic softpoint under the control structure of system 800xA. Afterwards, following the same procedure for creating the faceplates as described earlier, the Graphics Display objects are then used similarly to the faceplates to add to the property of a pipe segment. Therefore, by writing appropriate scripts in the property editor of a line segment, the small text boxes can appear on a screen by clicking on a pipe or signal line segment when propagation paths are being visualised.

To achieve an adequately functioning monitoring interface, other faceplates must be added to the drawing into their relevant locations. Therefore, this procedure described above is repeated several times to create the necessary types of faceplates for other equipment and instruments. Specifically, in addition to the three prior created faceplates, thirteen additional faceplates are created for different purposes, including control setpoint, motor-fault alarm, measurements display and control mode. These faceplates are created using their corresponding variables from control modules or control programs as follows: M-104, M-201, DartPID (302), FC203 AO, Larox_CC_SP, Dart_MA, M104F, FTIC_106, FC-203, TT-205, FT-207 and LT-301. Hence, a fully functioning monitoring interface can be achieved by creating a few additional necessary items, such as push-buttons, an aspect-view wrapper and aspect-view buttons. Four push-buttons are created in the drawing, the “Start” button and “Stop” button are used for initiating and ending the operation of the pilot process respectively; while the “Start FPPD” and “Reset FPPD” buttons are used for initiating and resetting the function for fault propagation path visualization respectively. An aspect-view wrapper is created through the “Special” toolbox, enabling a minimized display of an aspect; thus, an alarms aspect is created into the bottom part of the monitoring interface. Finally, two aspect-view buttons are created to enable access to other aspects; the first one enables access to the trends aspect, while the second enables access to the alarms aspect.

6.2.2 Data transfer to the system database and to the elements of the process graphics

Causal models, such as a connectivity matrix, can be utilized to refine the results of fault propagation analysis (e.g. causality matrix) in order to achieve non-spurious and more reliable solutions. This thesis seeks to utilize a refined causality matrix acquired from data analysis for the visualization of fault propagation paths in the process monitoring interface. Thus, the remainder of this subsection describes the implemented procedure which enables the elements of the process monitoring interface to extract data (i.e. the elements of the causality matrix) contained in an Excel file. Figure 19 shows the form and path which data is transferred from the source (i.e. Excel file) to the elements of the process monitoring interface.

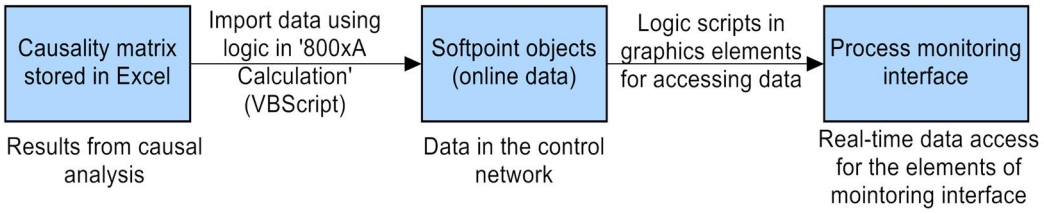


Figure 19: Overview of data transfer from analysis to visualization

6.2.2.1 Softpoint objects in System 800xA for data representation

System 800xA supports “softpoint” objects which can be utilized to configure and use internal process variables that are not connected to an external physical process signal (ABB, 2008). All physical process signals (including measurements and control signals) are available within the control network and can be accessed, for instance, through the “control structure”. An instantiated softpoint object can represent or is similar to an actual element of a process, while the signals contained in such softpoint object are similar to tags (i.e. variables) of an actual process element. The benefits of “softpoint” service in the context of this thesis is enabling the structuring and storing data transferred from an external source (e.g. Microsoft Excel) into System 800xA

control network, which can then be utilized in similarity to other process variables. A softpoint object type must first be configured (i.e. pre-specified) in the "object type structure" of System 800xA; hence, this object type becomes a template from which actual softpoint objects can be instantiated. Once the softpoint objects are instantiated and the signals are configured, they can be accessed by other applications in 800xA system in similarity to actual process points, i.e. process signals. Larger-scale industrial processes can as well benefit from the potential of softpoint services, since the single softpoint server in System 800xA can host up to 25,000 signals within 2,500 individual softpoint objects (ABB, 2008). Furthermore, they support a number of different data types, including integer, Boolean, floating point and string. The following three paragraphs describe the procedure of firstly creating a softpoint object-type in the object-type structure, secondly instantiating softpoint objects in the control structure and lastly adjusting some of the individual signals properties.

The plant explorer is navigated to the "object type structure", from which "softpoint object types" is selected. A new object type is created by selecting "new object" in the context menu (i.e. mouse right-click) and it is named "PropagationCellData" in the "New object" dialogue window. The PropagationCellData object type is configured by selecting the "process object configuration" aspect, where new signals are added to the new softpoint in the "Add signal" menu. Hence, the following five different signals are added to the PropagationCellData softpoint by selecting their appropriate data types: "from", "isendpoint", "isrootcause", "to" and "value". Subsequently, the properties of these signals must be configured to the suitable format. This is achieved by first selecting an individual signal from the softpoint structure, then selecting "signal configuration" aspect, where a few properties of the signals are individually configured according to the specification seen in Table 6 below. Henceforth, the PropagationCellData object type created here in the "object type structure" is available as a template, numerous softpoint objects can thereby be created in the control structure using this template.

Table 6: Configuration properties of signals in PropagationCellData softpoint

PROPERTY / SIGNAL	Is controllable	Range	Precision
from			
isendpoint	*		
isrootcause	*		
to			
value	*	0.00 – 100.00	single (32 bit)

The actual softpoint objects are instantiated (i.e. created) in the control structure, where most parameters and properties are inherited from the associated object type (ABB, 2008). A little configuration is required here in the instantiation phase; thus, initially specified parameters and properties can be further configured and adjusted according to specific requirements. New softpoint objects can be created using two different approaches: the one-object-at-a-time dialogue box can create a single unique object, whereas the bulk instantiation dialogue box allows the creation of numerous unique softpoint objects at once. Accordingly, all the softpoint objects to be utilized in this thesis are created using the bulk instantiation approach, which begins by navigating to the control structure and selecting the "softpoint generic control network" node. Selecting the "generic control network configuration" aspect on the right side causes the configuration window to appear. Subsequently, the "new object" icon is selected to open the bulk instantiation dialogue box; thus, PropagationCellData is selected as the type to base the new objects on, while name of new object is "CausalMatrixData" and number of new objects is fourteen. Hence, fourteen new softpoint objects are created simultaneously with unique names (i.e. CausalMatrixData1 – CausalMatrixData14).

At this stage, certain properties of some individual signals must be adjusted or further configured. Consequently, the values of only two signals (i.e. "from" and "to") are

specified for each individual CausalMatrixData objects, since each one of these fourteen softpoint objects represent a pipe-segment in the flotation pilot process. Thus, names of the two equipment (or instruments) which each pipe-segment links together are specified in the "from" and "to" signals of the corresponding softpoint object. The other three signals are not specified, since these fourteen softpoint objects are needed in this thesis to store data points in the control network and will be frequently modified by other applications as required. Furthermore, one additional softpoint object is created using similar procedure as earlier, to enable a complete reset for the "value", "isendpoint" and "isrootcause" signals of all the fourteen softpoint objects created earlier. This softpoint object is named "Reset_F_Prop" and only two of its five signals ("start_signal" and "reset_signal") will be functional. Finally, before the new configurations take effect and become accessible to other applications, all softpoint objects must be "deployed" once created or after modification. Therefore, these softpoint objects are deployed by selecting the "softpoint generic control network" node; thereafter, the "Deploy" aspect is selected from the list of aspects, subsequently the "Deploy" function (i.e. through a push-button) completes the deployment task in the lower windowpane.

6.2.2.2 Calculation aspects in System 800xA for data transfer

Calculation aspects in System 800xA provide an environment for transfer, manipulation and computation of real-time data objects (i.e., actual process points and softpoints or OLE for Process Control (OPC)). They can also be applied to object types, enabling the calculation to be re-applied every time a new object is created from such object type. A calculation aspect may be applied to any aspect object, such as a unit, tank, pump, valve or softpoint object. Specifically, the signal of any aspect object can be read (i.e. input direction) from such object into a variable of a calculation aspect. Likewise, the signal of any aspect object can be written (i.e. output direction) into a variable of a calculation aspect. The ease of configuring, scheduling and logic programming of calculation aspects is enhanced by a dedicated editing interface. (ABB, 2008)

Since calculations are created as aspects of OPC objects, a new calculation is thus created by navigating the plant explorer to control structure and selecting the "softpoint generic control network" node. Once the node is expanded, "softpoints type" which is a container of all the softpoint objects is selected and using its context menu, the "new aspect" creation dialogue box is opened. "Calculation" is selected in the dialogue box as aspect type and the aspect is named "Calculation2_FCell_Thesis" to complete the aspect creation step. At this point, the new calculation aspect appears on the right side, in the list of aspects associated with "softpoint type". The calculation editor interface opens up by selecting a calculation aspect (i.e., Calculation2_FCell_Thesis) as shown in Figure 20; the following paragraphs describe the configuration and logic programming tasks in this calculation aspect.

Fourteen variables are initially added to the variable grid, each of these will be referencing some specific signals from the fourteen softpoint objects created earlier. A new row for placing a new variable is inserted using the "insert line" function, which is selected from the menu bar of the editor view. The new variable is named "value1", given in the variable column of the variable grid. While the signal of the softpoint object to which it is referencing (i.e. target signal) is specified in the object column. Furthermore, this softpoint object signal must be specified using a full name path i.e., "[Control Structure]SoftPoints.CausalMatrixData1.value". The next column of the variable grid is the property column, where the property of the signal type is specified as "REAL PCA:VALUE". The log column is left empty, while the column after it is the direction column, used for specifying the output/input direction of the variable to its target signal. Thus, the output is chosen as the direction here, since the signal of the softpoint object takes the value of variable "value1" during the execution of this calculation aspect. The online-value column is left empty, since it will be updated online; whereas, the state column is specified as "online", in order to allow the actual transfer of data between the variable and the softpoint object signal in real-time. Further, the next column is the offline-value and this is left empty, since it will be updated during offline tests. The last column is the event column, which is specified as false, this is because changes in the values of this variable do not trigger the execution of this calculation aspect. The steps described above are repeated many times to create and specify the remaining variables with unique names, each of these

variables will reference its corresponding softpoint object signal in the control network. In addition to these fourteen variables, two variables named “Start_FP” and “Reset_FP” are created in order to reference the “Reset_F_Prop” softpoint object signals i.e. “start_signal” and “reset_signal” respectively. The property column of these variables is specified as “BINARY PCA:VALUE”, while the remaining columns are specified mostly similar to other fourteen variables. Furthermore, seventeen additional variables are added to the variable grid, the property column of these variables is specified as “BINARY PCA:VALUE”, thus any of these can be utilized for specifying a root cause or an endpoint of fault propagation. It is important to provide unique names to all of the thirty-three individual variables that constitute this calculation aspect. Lastly, all of the modifications to the calculation aspect must be saved before leaving the editor interface, in order to confirm the recent modifications.

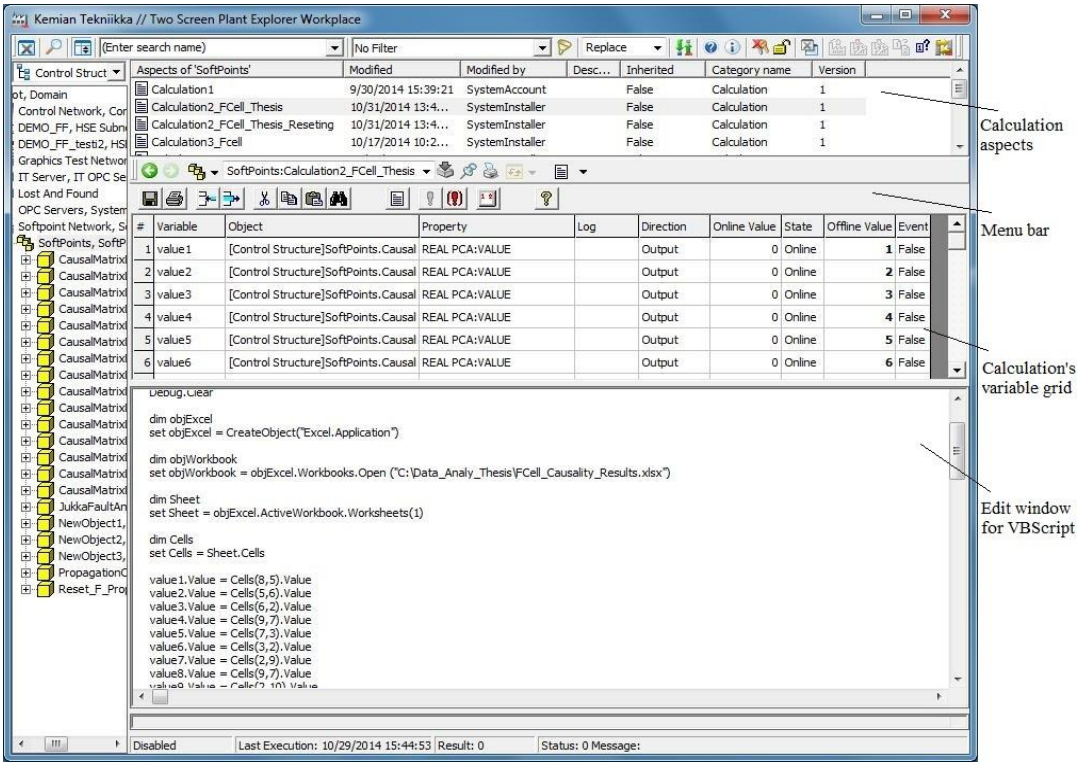


Figure 20: The calculation aspect editor interface in System 800xA

All calculation logics are programmed within the edit windowpane of a calculation editing interface, where Visual Basic language scripting edition i.e., Visual Basic script (VBScript) is the accepted language for programming the logic in calculations. Since calculations are defined using VBScript, data can be manipulated using all of the programming capabilities within VBScript language (ABB, 2008); therefore, data is imported and transferred to the necessary variables within this calculation edit windowpane. Thus, program code is written in the edit windowpane using VBScript to accomplish the following tasks: locate and open an Excel file (causality matrix file), open the first Excel sheet, get data from the Excel cells (i.e. assign to calculation variables) and close the Excel file. The above mentioned tasks constitute the logic which must be programmed in this calculation aspect to achieve the objectives of data transfer from Excel to the softpoint objects. This VBScript program code is available (see Appendix H). In addition, a new calculation aspect named "Calculation2_FCell_Thesis_Reseting" is created using a similar approach. Since the aim of the first calculation aspect is to import and transfer data from external source to the process graphics elements through the softpoint objects, the property values of the process graphics elements must be returned to their original values, thus resetting the animated display. Consequently, this second calculation aspect is created, configured and programmed particularly for this purpose. A list of calculation aspects and the calculation editor interface in System 800xA is shown in Figure 20. Meanwhile, two push-buttons (i.e. "Start FPPD" and "Reset FPPD") in the process monitoring interface are used for controlling the execution of these two calculation aspects through the two binary signals of the "Reset_F_Prop" softpoint object. Specifically, the binary signals "start_signal" and "reset_signal" of softpoint object Reset_F_Prop are used for controlling the start and reset actions respectively. Finally, in order to run the calculation aspect, it must be enabled using the "Enable" function in the menu bar of the calculation editor interface. In the absence of error messages, the calculation aspects are executed automatically using either the "Start_FP" or the "Reset_FP" push-button in the process monitoring interface.

6.2.2.3 Data access from softpoints to the process graphics elements

Process monitoring interfaces provide visual and numerical information about the state of a process as well as the enabling supervisory control of a process. To achieve these functions, the elements of a monitoring interface must have access to the relevant sources of data, which are real-time process data and accessed through the connectivity server of the automation network. Since the method of fault propagation paths visualization being developed in this thesis works as an offline-tool, thus real-time process data are not utilized for controlling the visualization of fault propagation paths. Contrarily, data acquired from causal analysis methods (i.e. causality matrix) are transferred and stored into the database of System 800xA using softpoint objects. These softpoint objects store the most recent causality matrix data; since softpoint objects are in the control network of System 800xA, applications such as the graphics builder thus have real-time access to the softpoint objects. Therefore, the elements of the monitoring interface are programmed in System 800xA graphics builder to transform based on the individual softpoint object data they access. This is achievable through program codes written for the properties of the graphics elements; Figure 21 is an example of a program code (i.e., graphics builder script language) which is written for controlling the colour property of a graphics element. The properties panel is visible on the right side of the window, while the overlapping window is the "Expression Editor" where relevant logic (i.e., program codes) is implemented. This thesis implements a dynamic colour transformation not only for the pipes and signal lines; likewise, equipment and instruments of the flotation pilot process are programmed as well to transform dynamically, for specifying the root cause and end-point of propagation of faults. Accordingly, the following paragraphs describe the implementation of fault propagation paths in the flotation process, using the dynamic transformation in the colours of the process elements within the process monitoring interface.

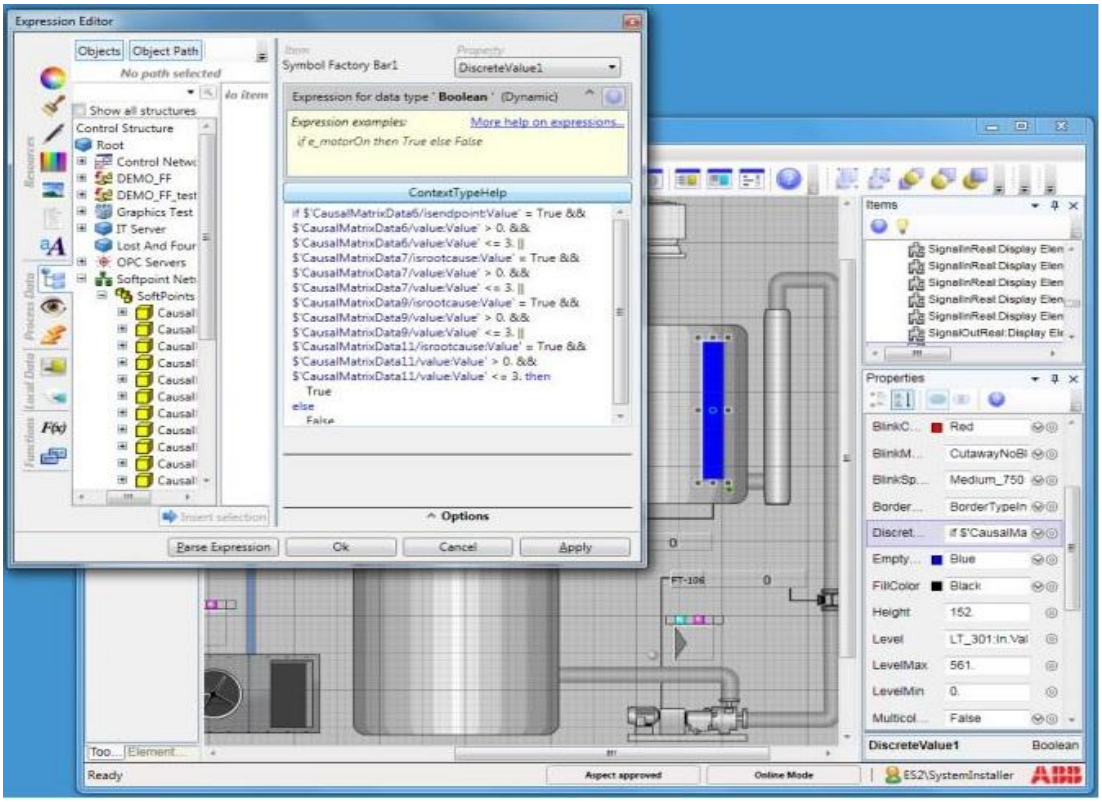


Figure 21: Expression editor of an element in the graphics builder application

Most of the elements in the process monitoring interface have silver-grey colour during normal operation of the process. However, the colour of the certain elements must transform dynamically, as an implementation of the visualization of fault propagation paths. This helps the process operator to visualize the spread and effects of faults as part of the monitoring and supervisory control of the process. That is, the colour of the pipe and signal lines located between (i.e. within the path) the root cause of a fault and the end point transforms for as long as the fault is present. Therefore, this thesis implements a colour transformation for the root cause, paths of spread and the end point of a fault, by altering the colour from the original colour to orange colour. The colour transformation occurs only in the event of a fault; this is enabled since the elements of the process have access to the softpoint objects, which represents the causality matrix acquired from causal analysis. Additionally, orange colour is selected as the colour for visualizing fault propagation paths, since process elements fault states are presented in orange colour, as recommended by guidelines and conventions from

the Joint Controls Project (JCOP) under The European Organization for Nuclear Research (CERN) (JCOP, 2007).

To implement the visualization of fault propagation paths function for the process monitoring interface, the graphics builder application in System 800xA is used to accomplish most of the tasks involved in the design. Since this implementation is integrated with the earlier created monitoring interface, the "Flota_cell_disp_thesis" aspect is opened in the graphics builder application automatically by selecting "Edit" from its context menu. This implementation intends only to be supplementary to the "Flota_cell_disp_thesis" aspect; thus the aim is not to modify the functionality of the existing monitoring system, but to add new functionalities to it. Since the process monitoring interface is composed of equipment, instruments, connecting pipes and signal lines; each of these components can be individually selected and programmed to access specific data points (softpoint signals), as well as implement the logic for dynamic transformations. Therefore, as the first implementation instance, the connecting pipe between the compressed-air blower and the air-flow meter is selected, thus its properties are visible in the properties panel. The properties panel displays all the possible adjustable properties of the element in a list; hence, the "InnerShadeColour" property is selected from this list. Each property has a button, through which an "Expression Editor" can be accessed; furthermore, this expression editor is an overlapping window in which data access and logic programming are implemented. Figure 21 is a similar illustration of an overlapping expression editor for one of the properties of the level sensor. Accordingly, the expression editor for the "InnerShadeColour" property of this connecting pipe is opened, enabling the implementation of dynamic colour transformation of the connecting pipe. Signals from the control network (e.g., the softpoint objects signals) can be access through this expression editor; further, selecting the "process data" tab and navigating to the appropriate signals within the control structure branch. When the appropriate signal has been selected, "Insert selection" is used for selecting the signal into the script-editing space for composing the logic program. The logic for manipulating the dynamic transformation of the colour of an element is completely programmed here in the script-editing space of the expression editor. Figure 22 shows the data access and the manipulation logic in the expression editor for the first connecting pipe. Lastly,

"Apply" and "Ok" buttons are pushed consecutively to complete this instance of the implementation.

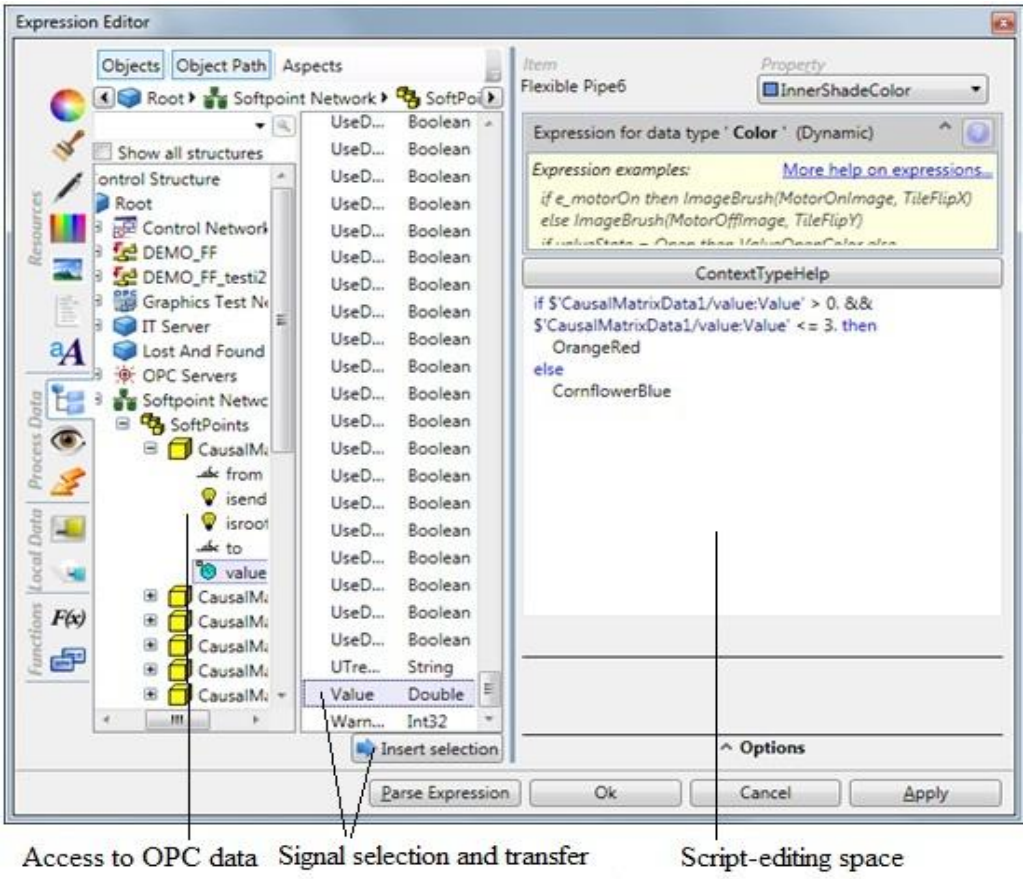


Figure 22: Graphic element access to data and logic programming

The dynamic transformation of elements of the monitoring interface based on the real-time values of the causality matrix in the database (softpoint objects) implies that: the colour of an element will transform from the original to orange when its associated softpoint signal receives a value larger than zero. Accordingly, the script in Figure 22 implements this logic by asserting that the original colour of "CornflowerBlue" changes to "OrangeRed" when the value of the associated softpoint signal is larger than zero. Henceforth, the above described procedure is repeated several times; specifically, sixteen times for the pipe and signal lines, as well as twelve times for the

equipment and instruments. This hereby completes the implementation of the dynamic transformation of colours for the monitoring interface elements. Additionally, this procedure is mainly implemented similarly for all elements; however, it is implemented slightly differently the for equipment and instruments, specifically in the selection of the element property being adjusted. Therefore, the procedure is implemented for the equipment and instruments as follows. Firstly, the "AnimationMode" and "BandCount" properties are specified to "DiscreteColorFill" and "1" values respectively in the properties panel of the selected element. Subsequently, "band1Color" is selected to "OrangeRed" colour, thus its associated property i.e. "DiscreteValue1" is selected and the expression editor is thereby opened. Finally, all the necessary data access and logic programming tasks are implemented in the expression editor, thereby completing the implementation of the method for fault propagation paths visualization.

6.3 Results

A flowsheet of the flotation pilot process was created in AutoCAD P&ID and further converted into a connectivity matrix, using a procedure described in Sun (2013). The connectivity matrix of the flotation pilot process is presented in section 5.2. Afterwards, the experiment performed from the operation of the flotation pilot process produced sufficient data, which was later analysed using the Time-domain Granger causality, thereby generating a causality matrix. Using the procedure described in Landman et al. (2014), the connectivity matrix was exploited for refining the initial causality matrix, in order to subsequently obtain a refined causality matrix. Both the initial causality matrix and the refined causality matrix are presented in section 6.1. This refined causality matrix provides the details of the propagation paths of the fault introduced into the flotation process experiments.

The process monitoring system for the flotation pilot process was built using the graphics builder application in System 800xA. Furthermore, a scheme that will enable the transfer of the causality matrix data into the database of System 800xA was developed, thus enabling the process monitoring system to access the refined causality

matrix. Since the design of the process monitoring interface is complete and the refined causality matrix is available in the database of System 800xA, the overall procedure of visualizing fault propagation paths in the monitoring interface was then tested offline. It was known that the level sensor LT301 is the root cause, it can be seen in Figure 23, indicated with the light-green circle. Therefore, it was specified as the fault origin inside the Microsoft Excel file containing the refined causality matrix before initializing the test. The test was initiated with the proper push-button, thus displaying the propagation path of the fault in the interface of the process monitoring system as seen in Figure 23. Also, the small text window, seen at the lower-left corner opened up when a faulty or an affected element is clicked. While the fault propagation path was being displayed, the root cause and the end point of the fault blink constantly, thus indicating the scope of the fault propagation.

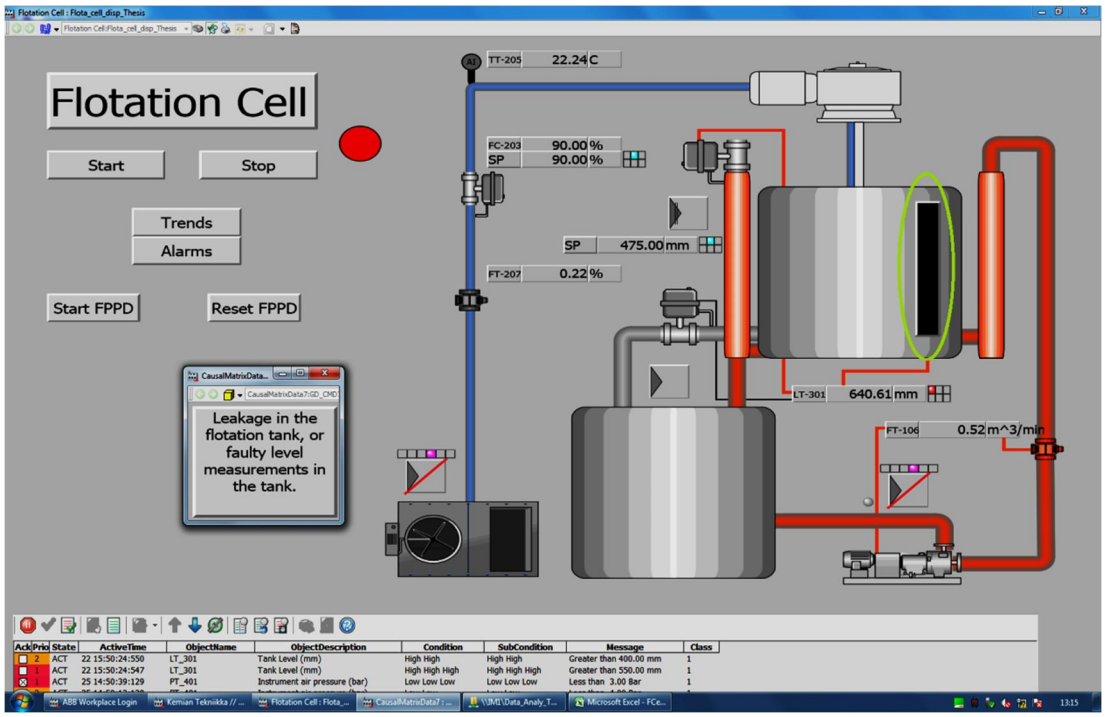


Figure 23: The visualization of fault propagation path in the flotation pilot process

7 CONCLUSION

The methods used today for diagnosing faults in industrial processes are reported to often generate spurious results. Accordingly, several researchers introduced an approach combining data-based and topology-based methods, in order to significantly improve the results generated by the methods. In this thesis, a connectivity matrix acquired from the P&ID of the flotation process represents a topology-based model, which was used for validating a causality matrix generated from a Granger causality – a data-based method, thus providing a refined matrix. This hereby addresses the challenges of spurious results reported in the past, also providing reliable details of fault propagation paths for a process.

This thesis proposed that displaying fault propagation paths in a process monitoring system can improve the understanding of spread of the effects of a fault throughout the process. Thus, the procedure to access the refined causality matrix and visualise it in the process monitoring system was successfully developed, and further tested using the case study – flotation pilot process. The tests revealed that visualising the propagation of faults with specified colours in the monitoring interface enables the operator of the process clearly see the directions and effects of fault downstream the process. Further studies can provide a means to visualise the fault propagation paths online and in real-time. This will drastically mitigate the effects of potential harms or danger, which are reportedly not rare in the process industry. Lastly, the procedure developed in this thesis can be implemented regardless of the vendor of the software and tools utilised.

REFERENCES

ABB Corporation. ABB Industrial IT System 800xA, Information Management configuration document. *http://www.aotewell.com/pdf/manual/ABB/Information%20Management/3BUF001092_IM_Configuration.pdf* (accessed 14 November, 2014).

ABB Corporation. ABB corporation's website - System 800xA. *http://www05.abb.com/global/scot/scot343.nsf/veritydisplay/e2c8177d884cef5bc1257b4e004c577f/$file/3BSE069330_C_en_System_800xA_Solutions_Handbook.pdf* (accessed 12 November, 2014).

Autodesk Inc. AutoCAD P&ID 2014 getting started guide. *Autodesk, Inc.* **2013**, 114p.

Bauer, M.; Cox, J. W.; Caveness, M. H.; Downs, J. J.; Thornhill, N. F. Finding the Direction of Disturbance Propagation in a Chemical Process Using Transfer Entropy. *Control Systems Technology, IEEE Transactions on* **2007***, 15*, 12-21.

Bauer, M.; Thornhill, N. F. A practical method for identifying the propagation path of plant-wide disturbances. *J. Process Control* **2008***, 18*, 707-719.

Chiang, L. H.; Braatz, R. D. Process monitoring using causal map and multivariate statistics: fault detection and identification. *Chemometrics Intellig. Lab. Syst.* **2003***, 65*, 159-178.

Duan, P.; Chen, T.; Shah, S. L.; Yang, F. Methods for root cause diagnosis of plant-wide oscillations. *AIChE J.* **2014**.

Fedai, M.; Drath, R. CAEX-a neutral data exchange format for engineering data. *ATP International Automation Technology* **2005***, 1*, 3.

Gao, D.; Wu, C.; Zhang, B.; Ma, X. Signed Directed Graph and Qualitative Trend Analysis Based Fault Diagnosis in Chemical Industry. *Chin. J. Chem. Eng.* **2010***, 18*, 265-276.

Horch, A.; Cox, J. W.; Bonavita, N. Peak performance. *ABB Rev.* **2007**, 24-29.

Iri, M.; Aoki, K.; O'Shima, E.; Matsuyama, H. An algorithm for diagnosis of system failures in the chemical process. *Comput. Chem. Eng.* **1979***, 3*, 489-493.

Isermann, R. *Fault-diagnosis applications 1st edition;* Springer-Verlag, 2011.

Jämsä-Jounela, S. -L.; Dietrich, M.; Halmevaara, K.; Tiili, O. Control of pulp levels in flotation cells. *Control Eng. Pract.* **2003***, 11*, 73-81.

Jämsä-Jounela, S. -L.; Tikkala, V. -M.; Zakharov, A.; Pozo Garcia, O.; Laavi, H.; Myller, T.; Kulomaa, T.; Hämäläinen, V. Outline of a fault diagnosis system for a large-scale board machine. *Int J Adv Manuf Technol* **2013***, 65*, 1741-1755.

JCOP, F. T. Joint Controls Project (JCOP) Framework Sub-Project Guidelines and Conventions. *CERN, Geneva, CERNJCOP-2000-008, Draft* **2007***, 6*.

Jiang, H.; Patwardhan, R.; Shah, S. L. Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix. *J. Process Control* **2009***, 19*, 1347-1354.

Kämpjärvi, P.; Jämsä-Jounela, S. -L. Level control strategies for flotation cells. *Minerals Eng* **2003***, 16*, 1061-1068.

Kujawa, C. (Outotec Oyj). Froth flotation method and apparatus, a froth flotation method and apparatus for extracting bitumen from a slurry of water and oil sand, and use of the apparatus. U.S. Patent 8,276,761 B2. October 2, 2012. Google patents (accessed 09.07.2014). **2012**.

Ladroue, C.; Guo, S.; Kendrick, K.; Feng, J. Beyond element-wise interactions: identifying complex interactions in biological processes. *PLoS One* **2009***, 4*, e6899.

Landman, R. Data-driven causal analysis and itsapplication on a large-scale board machine. Master's thesis, Aalto University, Degree programme for Master of Science in Chemical Technology, Espoo, Aalto University, Espoo, 2013.

Landman, R.; Kortela, J.; Sun, Q.; Jämsä-Jounela, S. -L. Fault propagation analysis of oscillations in control loops using data-driven causality and plant connectivity. *Comput. Chem. Eng.* **2014***, 71*, 446-456.

Lee, G.; Song, S.; Yoon, E. S. Multiple-Fault Diagnosis Based on System Decomposition and Dynamic PLS. *Ind Eng Chem Res* **2003***, 42*, 6145-6154.

Maurya, M. R.; Rengaswamy, R.; Venkatasubramanian, V. A Signed Directed Graph and Qualitative Trend Analysis-Based Framework for Incipient Fault Diagnosis. *Chem. Eng. Res. Design* **2007***, 85*, 1407-1422.

Meech, J. A. Flotation machines. *http://www.jmeech.mining.ubc.ca/MINE290/MINE292-Lecture17-Flotation%20Machines-2014.pdf* (accessed 08 July, 2014).

Outotec Corporation. Outotec corporation's website. *http://www.outotec.com/en/Products--services/Process-equipment/Flotation-cells/Tank-cells/* (accessed 04 July, 2014a).

Outotec Corporation. Outotec corporation's website. *http://www.outotec.com/en/Products--services/Process-equipment/Flotation-cells/Tank-cells/#tabid-5* (accessed 07 July, 2014b).

Pearl, J. *Causality: Models, Reasoning, and Inference: 2nd edition;* Cambridge University Press, 2009 .

Pozo Garcia, O.; Tikkala, V. -M.; Zakharov, A.; Jämsä-Jounela, S. -L. Integrated FDD system for valve stiction in a paperboard machine. *Control Eng. Pract.* **2013***, 21*, 818-828.

Ramentor Limited. ELMAS - Event Logic Modeling and Analysis Software. *http://www.ramentor.com/products/elmas/* (accessed December 16, 2014).

Schleburg, M.; Christiansen, L.; Thornhill, N. F.; Fay, A. A combined analysis of plant connectivity and alarm logs to reduce the number of alerts in an automation system. *J. Process Control* **2013***, 23*, 839-851.

Schreiber, T. Measuring information transfer. *Phys. Rev. Lett.* **2000***, 85*, 461.

Siemens Corporation. Siemens AG - Asset Management and Condition Monitoring. *http://w3.siemens.com/mcms/process-control-systems/SiteCollectionDocuments/efiles/pcs7/pdf/76/Paper_PumpMon_AM_CM_EN.pdf* (accessed December 16, 2014).

Siemens Corporation. Siemens process condition monitoring modular software. *http://www.industry.siemens.com/topics/global/en/magazines/process-news/process-automation/condition-monitoring-savvy-maintenance-doubles-savings/pages/default.aspx* (accessed December 16, 2014).

Stenlund, B.; Medvedev, A. Level control of cascade coupled flotation tanks. *Control Eng. Pract.* **2002***, 10*, 443-448.

Sun, Q. A method for generating process topology-based causal models. Master's thesis, Aalto University, Degree programme for Master of Science in Chemical Technology, Espoo. 2013.

Thambirajah, J.; Benabbas, L.; Bauer, M.; Thornhill, N. F. Cause-and-effect analysis in chemical processes utilizing XML, plant connectivity and quantitative process history. *Comput. Chem. Eng.* **2009***, 33*, 503-512.

Thornhill, N. F.; Horch, A. Advances and new directions in plant-wide disturbance detection and diagnosis. *Control Eng. Pract.* **2007***, 15*, 1196-1206.

Thornhill, N. F.; Cox, J. W.; Paulonis, M. A. Diagnosis of plant-wide oscillation through data-driven analysis and process understanding. *Control Eng. Pract.* **2003***, 11*, 1481-1490.

Vedam, H.; Venkatasubramanian, V. Signed digraph based multiple fault diagnosis. *Comput. Chem. Eng.* **1997***, 21, Supplement*, S655-S660.

Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N. A review of process fault detection and diagnosis part II: Qualitative models and search strategies. *Computers and Chemical Engineering* **2003b***, 27*, 313-326.

Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A review of process fault detection and diagnosis part I: Quantitative model-based methods. *Computers and Chemical Engineering* **2003a***, 27*, 293-311.

Wills, B.; Napier-Munn, T. *Mineral Processing Technology: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery;* Butterworth-Heinemann: Jordan Hill, 2006.

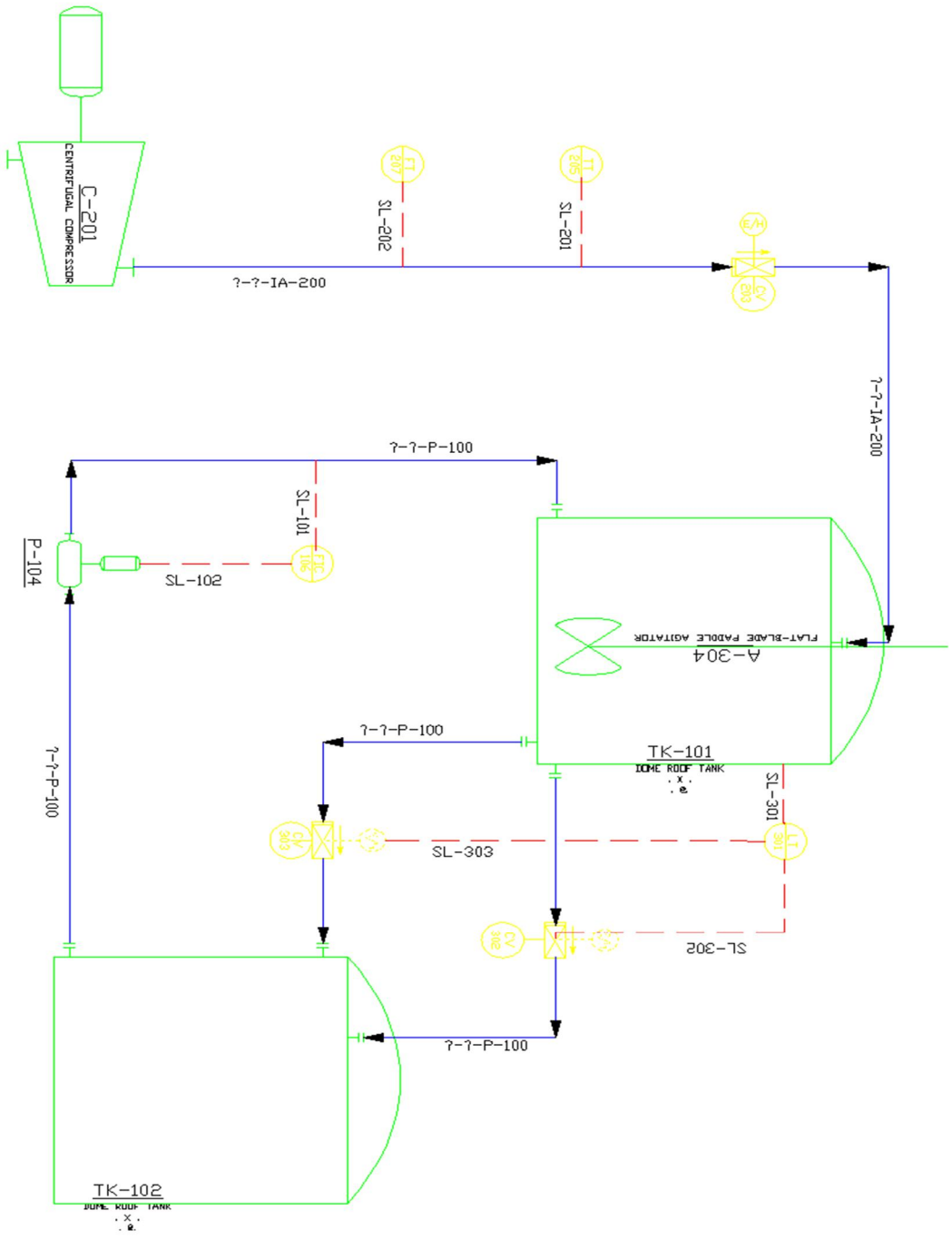
Yang, F.; Shah, S.; Xiao, D. Signed directed graph based modeling and its validation from process knowledge and process data. *Int. J. Appl. Math. Comput. Sci.* **2012***, 22*, 41-53.

Yang, F.; Xiao, D. Progress in Root Cause and Fault Propagation Analysis of Large-Scale Industrial Processes. *Journal of Control Science and Engineering* **2012**, 10.

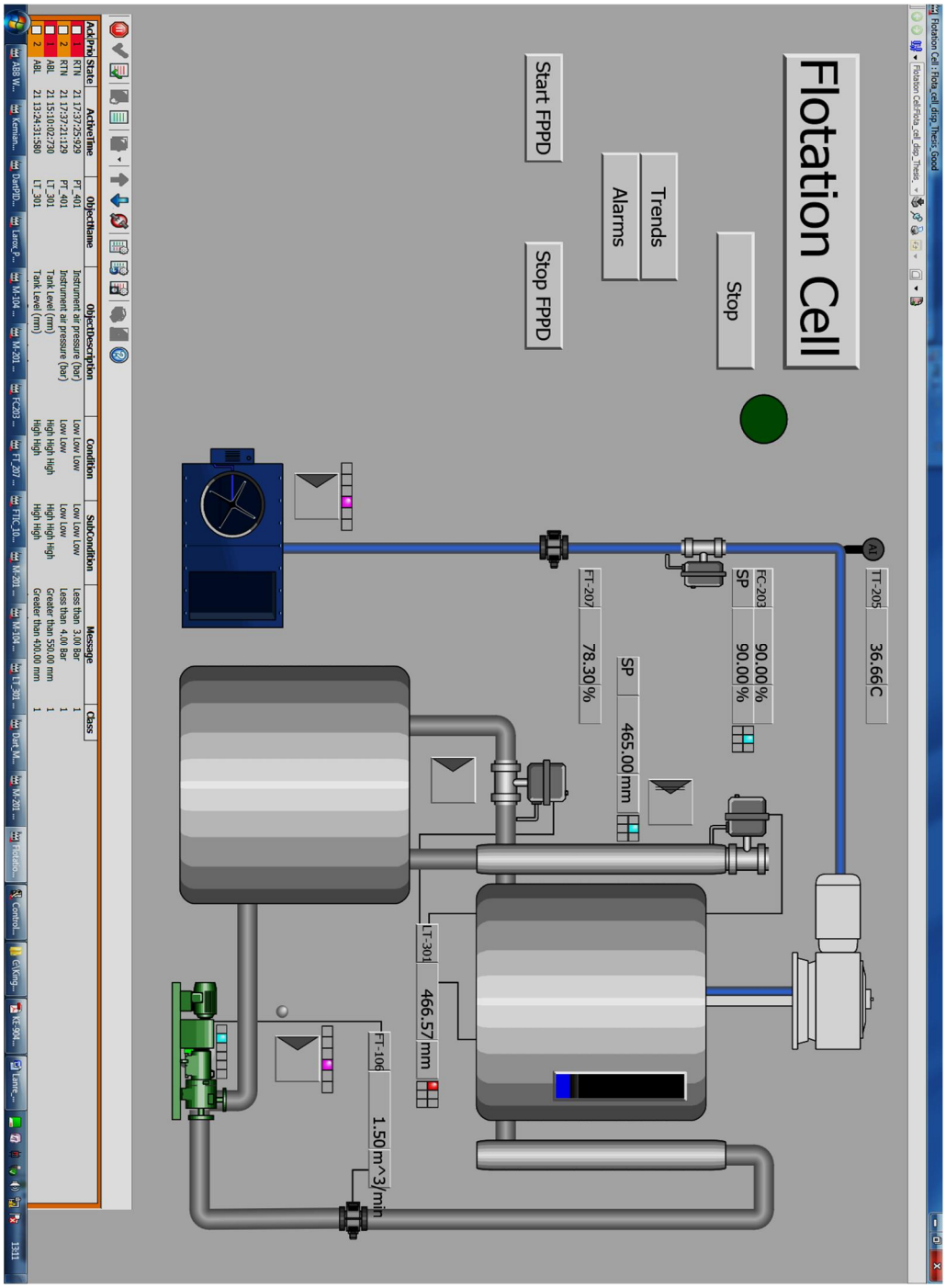
Yim, S. Y.; Ananthakumar, H. G.; Benabbas, L.; Horch, A.; Drath, R.; Thornhill, N. F. Using process topology in plant-wide control loop performance assessment. *Comput. Chem. Eng.* **2006***, 31*, 86-99.

Yuan, T.; Qin, S. J. In *In Root cause diagnosis of plant-wide oscillations using Granger causality;* Proceedings of 8th IFAC international symposium on advanced control of chemical processes, Singapore; 2012; , pp 160-165.

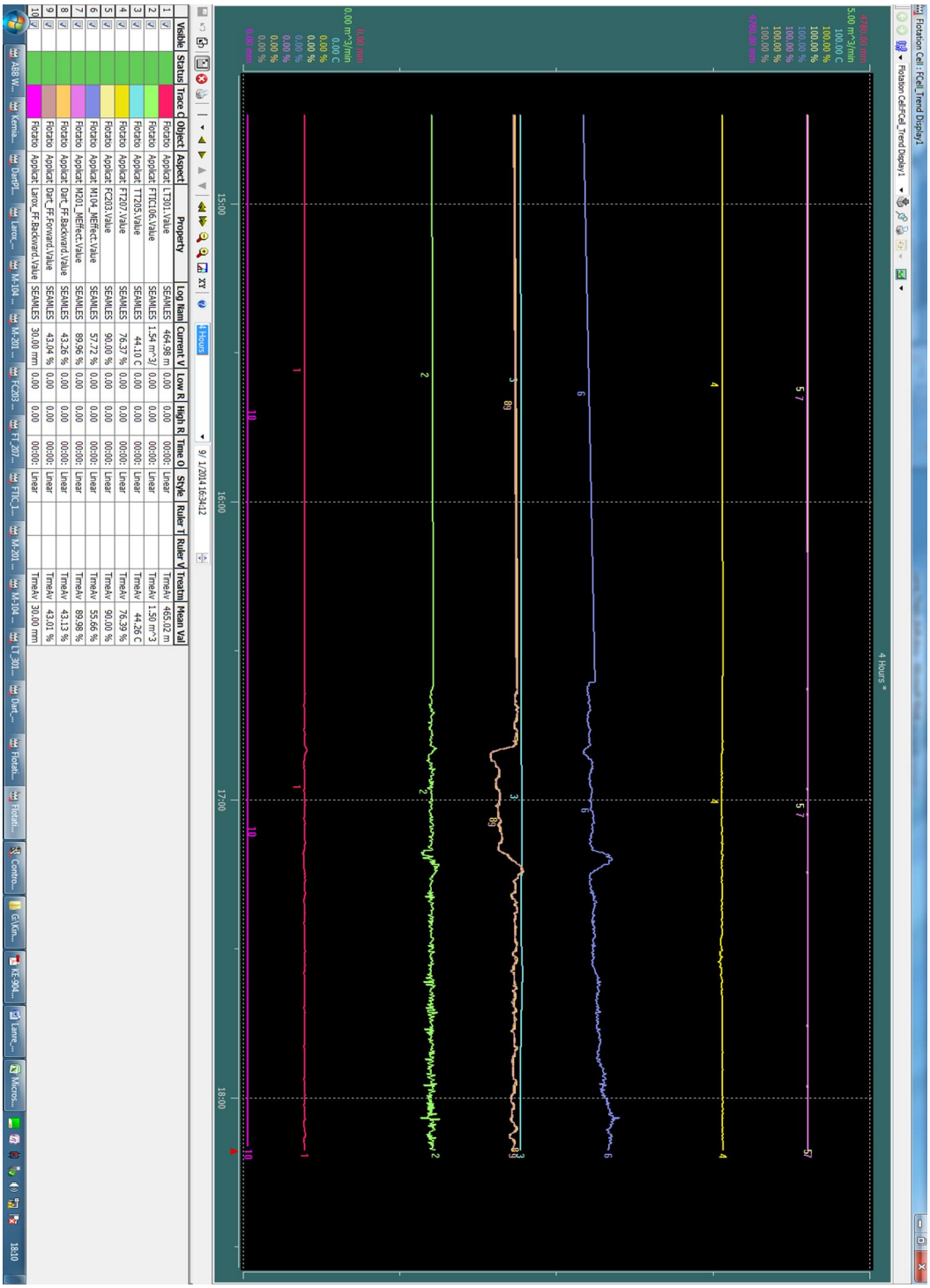
Appendix A: Electronic P&ID drawing of the flotation pilot process



Appendix B: Monitoring interface for the flotation pilot process

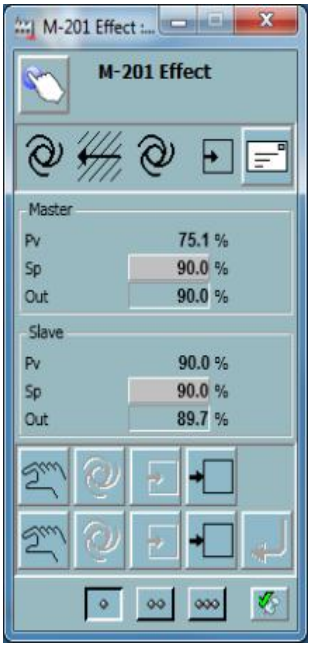


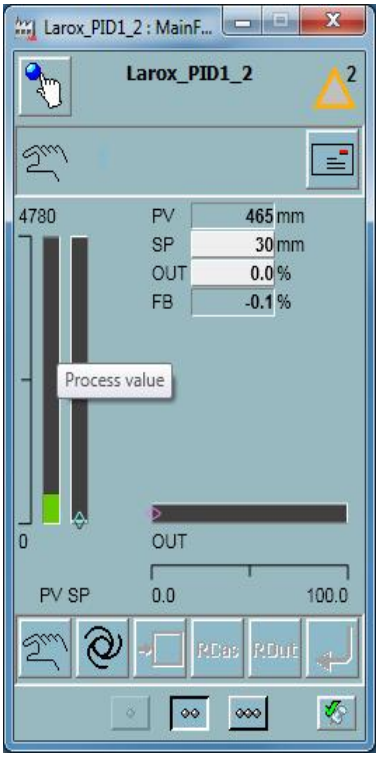
Appendix C: Operator trend display for the flotation pilot process

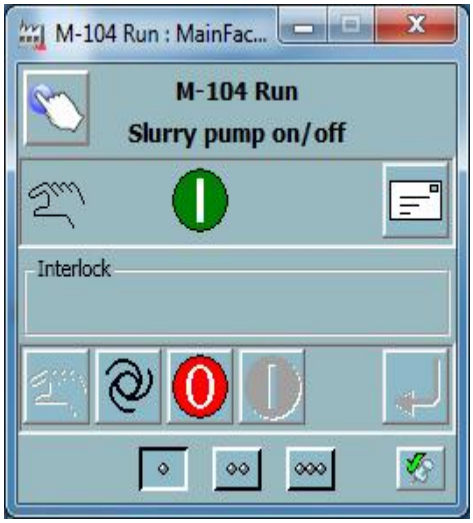


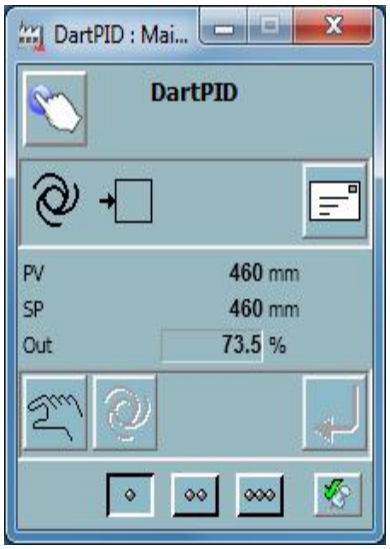
	Visible	Status	Trace Cl	Object	Aspect	Property	Log Ram	Current V	Low R	High R	Time O	Style	Ruler T	Ruler V	Treatm	Mean Val
1	✓			Flotato	Applicat	LT301.Value	SEAMLES	464.98 m	0.00	0.00	00:00:	Linear			TimeAv	465.02 m
2	✓			Flotato	Applicat	FTIC106.Value	SEAMLES	1.54 m^3/	0.00	0.00	00:00:	Linear			TimeAv	1.50 m^3
3	✓			Flotato	Applicat	TT205.Value	SEAMLES	44.10 C	0.00	0.00	00:00:	Linear			TimeAv	44.26 C
4	✓			Flotato	Applicat	FT207.Value	SEAMLES	76.37 %	0.00	0.00	00:00:	Linear			TimeAv	76.39 %
5	✓			Flotato	Applicat	FCI203.Value	SEAMLES	90.00 %	0.00	0.00	00:00:	Linear			TimeAv	90.00 %
6	✓			Flotato	Applicat	M104_MEffect.Value	SEAMLES	57.72 %	0.00	0.00	00:00:	Linear			TimeAv	55.66 %
7	✓			Flotato	Applicat	M201_MEffect.Value	SEAMLES	89.96 %	0.00	0.00	00:00:	Linear			TimeAv	89.98 %
8	✓			Flotato	Applicat	Dart_FF.Backward.Value	SEAMLES	43.26 %	0.00	0.00	00:00:	Linear			TimeAv	43.13 %
9	✓			Flotato	Applicat	Dart_FF.Forward.Value	SEAMLES	43.04 %	0.00	0.00	00:00:	Linear			TimeAv	43.01 %
10	✓			Flotato	Applicat	Larox_FF.Backward.Value	SEAMLES	30.00 mm	0.00	0.00					TimeAv	30.00 mm

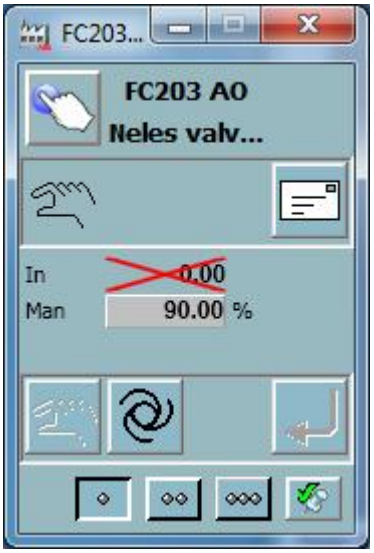
Appendix D: Faceplates for controlling the flotation pilot process





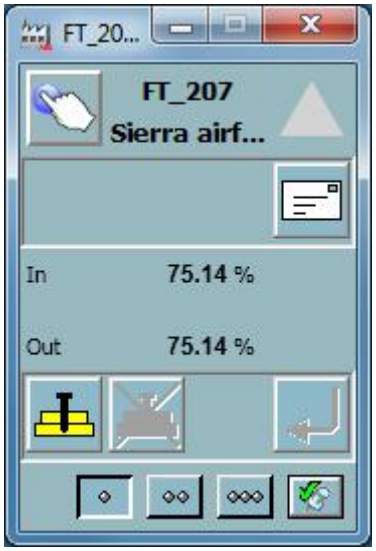



DartPID : Mai...
DartPID
PV 460 mm
SP 460 mm
Out 73.5 %

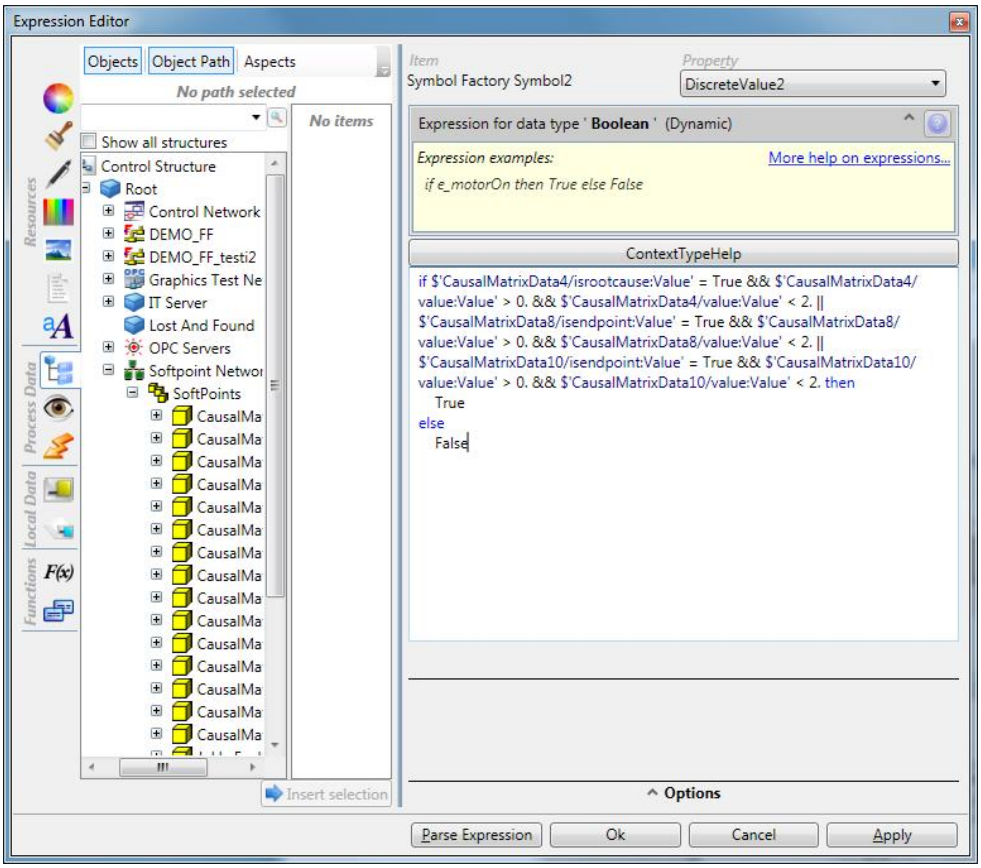

FC203...
FC203 AO
Neles valv...
In 0.00
Man 90.00 %


FTIC_1...
FTIC_106
Slurry flo...
In 2.51 m^3/min
Out 2.51 m^3/min

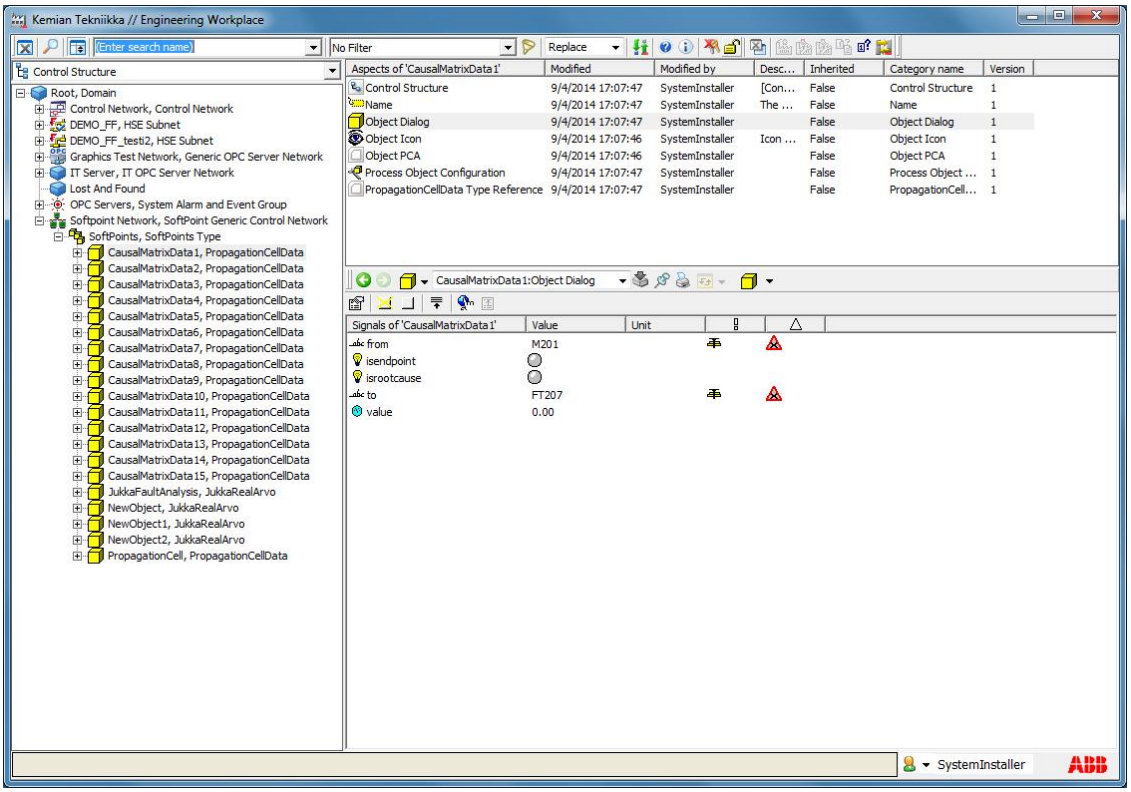

LT_30...
LT_301
Tank Leve...
In 464.76 mm
Out 463.02 mm


FT_20...
FT_207
Sierra airf...
In 75.14 %
Out 75.14 %

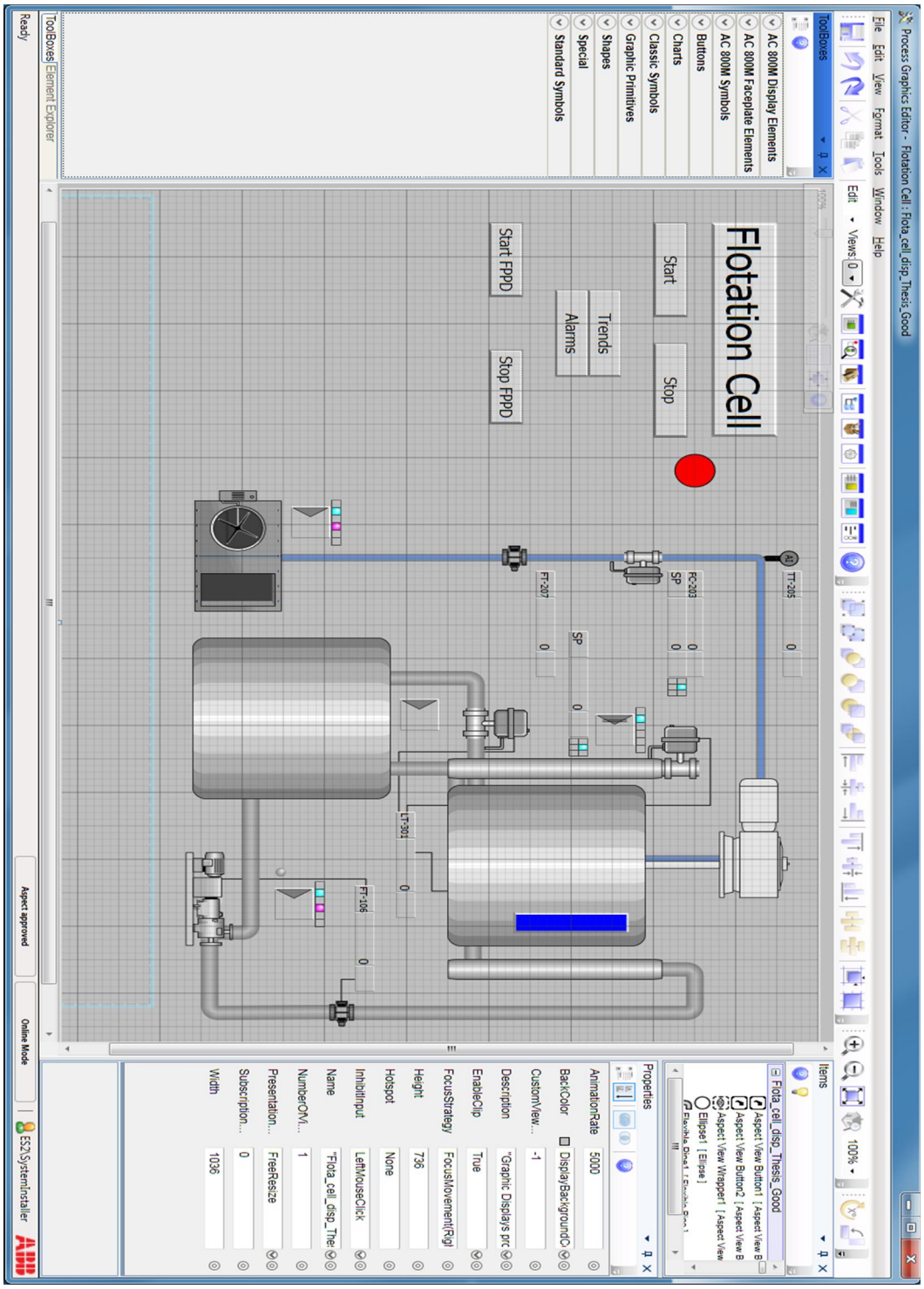
Appendix E: Algorithm for accessing data from the database



Appendix F: Softpoint objects for storing the causality matrix



Appendix G: Design of the monitoring interface in System 800xA



Appendix H: Data transfer algorithms in VBScript language

```
Option Explicit
If Start_FP = True Then
    Reset_FP = False
    Debug.Clear
    dim objExcel
    set objExcel = CreateObject("Excel.Application")
    dim objWorkbook
    set objWorkbook = objExcel.Workbooks.Open
("C:\Data_Analy_Thesis\FCell_Causality_Results.xlsx")
    dim Sheet
    set Sheet = objExcel.ActiveWorkbook.Worksheets(1)
    dim Cells
    set Cells = Sheet.Cells
    value1.Value = Cells(8,5).Value
    value2.Value = Cells(5,6).Value
    value3.Value = Cells(6,2).Value
    value4.Value = Cells(9,7).Value
    value5.Value = Cells(7,3).Value
    value6.Value = Cells(3,2).Value
    value7.Value = Cells(2,9).Value
    value8.Value = Cells(9,7).Value
    value9.Value = Cells(2,10).Value
    value10.Value = Cells(10,7).Value
    value11.Value = Cells(3,2).Value
    value12.Value = Cells(2,9).Value
```

```
value13.Value = Cells(2,10).Value
value14.Value = Cells(3,7).Value

dim rootcause
dim endpoint
rootcause = Cells(12,2).Value
endpoint  = Cells(12,10).Value
If rootcause = "LT301" Then
    rc301.Value = True
ElseIf rootcause = "FTIC106" Then
    rc106.Value = True
ElseIf rootcause = "TT205" Then
    rc205.Value = True
ElseIf rootcause = "FT207" Then
    rc207.Value = True
ElseIf rootcause = "FC203" Then
    rc203.Value = True
ElseIf rootcause = "M104" Then
    rc104.Value = True
ElseIf rootcause = "M201" Then
    rc201.Value = True
ElseIf rootcause = "Dart_valve" Then
    rcDart.Value = True
ElseIf rootcause = "Larox_valve" Then
    rcLarox.Value = True
Else
    Debug.Trace "Please specify the Root Cause correctly!"
```

```
End If

If endpoint = "LT301" Then
    ep301.Value = True
ElseIf endpoint = "FTIC106" Then
    ep106.Value = True
ElseIf endpoint = "TT205" Then
    ep205.Value = True
ElseIf endpoint = "FT207" Then
    ep207.Value = True
ElseIf endpoint = "FC203" Then
    ep203.Value = True
ElseIf endpoint = "M104" Then
    ep104.Value = True
ElseIf endpoint = "Dart_valve" Then
    epDart.Value = True
ElseIf endpoint = "Larox_valve" Then
    epLarox.Value = True
Else
    Debug.Trace "Please specify the End Point correctly!"
End If

objExcel.Workbooks(1).Close
objExcel.Quit

set Sheet = Nothing
set objWorkbook = Nothing
```

```
    set objExcel = Nothing
    dim i
    For i = 1 to 4
        Debug.Trace "Closing the current Excel-spreadsheet and quiting Excel application...."
    Next
    'Debug.Clear
End If
```

Appendix I: Alarm and Event list for fault propagation path analysis