DATA-DRIVEN APPROACH TO PREVENT PROCESS INCIDENTS IN COMPLEX SYSTEMS – IDENTIFY FUNCTION INTERACTIONS AND WEAK SIGNALS LEADING TO HAZARDS

A Dissertation

by

MENGXI YU

Submitted to the Graduate and Professional School of Texas
A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Costas Kravaris
Committee Members,	Mahmoud El-Halwagi
	S. Camille Peres
	Qingsheng Wang
Head of Department,	Arul Jayaraman

August 2021

Major Subject: Chemical Engineering

ABSTRACT

Most incidents in complex systems such as process plants follow incubation periods where weak signals exist for a long time. It is necessary to identify and resolve weak signals to prevent incidents proactively. Since “weak signal” was not precisely defined, the study first proposed its definition. Weak signals were defined as performance variabilities of functions whose interactions combine clues or signs giving rise to early prediction of a future incident. However, identifying weak signals based on individuals’ knowledge tends to be intellectually unmanageable due to complex function interactions and noise within the abundance of data in plants.

To recognize weak signals by their interaction effects, it is a prerequisite to systematically understand function interactions that lead to emerging hazards. The study developed a novel framework for process plants. The framework started from simulating function interactions in process plants through the integration of a human performance model, an equipment performance model and a chemical first-principal model based on Functional Resonance Analysis Method (FRAM). It was followed by a data-driven approach to quantify function couplings and identify the interactions leading to emerging hazards based on lift confidence intervals of association rules. The case study of a batch process showed the identified interactions could be graphically represented in FRAM with quantified function couplings, guiding people to understand how emerging hazards occur and take preventive measures.

Given abundance of data, challenges exist from selecting appropriate information for observing weak signals, evaluating their relevance, to responding. Therefore, the study developed a data-driven framework which involves FRAM and machine learning techniques to address the challenges. The case study of the same batch process showed great potentials for applying the framework in real operations. Given information of the potential weak signals that were extracted based on FRAM, probabilities of a selected hazard were predicted with high accuracy by Random Forest (RF) to indicate relevance of underlying weak signals. For interpretability, a Decision Tree (DT) that approximated the RF was developed with high fidelity, unfolding weak signals and corresponding corrective actions.

DEDICATION

To my family for their unconditional love and supports.

To the late Dr. M. Sam Mannan for inspiring, encouraging, and mentoring me.

ACKNOWLEDGEMENTS

The journey at Texas A&M University will be a treasure of my life. It is unforgettable because of all the challenges I have met, all the milestones I have achieved, and more importantly, all the great people I have known.

First, I would like to thank my committee chair Dr. Costas Kravaris. He accepted me as his student during the toughest time during my doctoral program. I greatly appreciate his guidance and care. Additionally, I would like to thank my committee members, Dr. Mahmoud El-Halwagi, Dr. Qingsheng Wang, and Dr. S. Camille Peres for their advice and supports throughout the course of this research.

I would like to extend my great appreciation to the late Dr. Sam Mannan. He inspired me to work on the novel research topic with his vision of the Process Safety discipline. Besides his technical guidance, his leadership and professional personality made great impacts on me.

My thanks also go to all my colleagues and friends at Mary Kay O'Connor Process Safety Center. Especially thank Dr. Noor Quddus and Dr. Hans Pasman for taking their time to discuss my research ideas and providing me valuable advice during my entire doctoral program. Thanks to Dr. Steward Behie, Mike O'Connor, Henry Goyette, and all administrative staff for their supports. I would also extend my thanks to Dr. Zohra Halim, Dr. Lin Zhao, Guanyang Liu, and Pratik Krishnan. I always enjoyed discussing my research topic with them.

Many thanks to Dr. Madhav Erraguntla from the Department of Industrial Engineering at Texas A&M University for his guidance about machine learning.

Thank High Performance Research Computing (HPRC) at Texas A&M University for providing advanced computing resources for this study. I gratefully acknowledge the consulting support provided by the HPRC staff, especially Marinus Pennings.

I also acknowledge my colleagues and friends at Department of Engineering. Especially thank Ashely Henley for her administrative support, and thank all the members at Dr. Kravaris's research group for their suggestions for my research.

Finally, my deepest appreciation goes to my family for their unconditional love and support. Special thanks go to my parents for their encouragement, and go to my husband, Yibo Zhu, for always believing in me and trying his best to help me when I struggled. I would not make it without the beautiful family.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professors Costas Kravaris, Mahmoud El-Halwagi, Qingsheng Wang of the Department of Chemical Engineering and Professor S. Camille Peres of the School of Public Health. This work was also supervised by Dr. Noor Quddus, a research scientist of Mary Kay O’Connor Process Safety Center at Texas A&M University.

All work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a fellowship from Texas A&M University and a dissertation research fellowship from Mary Kay O’Connor Process Safety Center.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION [*]

1.1 Background and Motivation

Process safety management (PSM) programs have been implemented over years since OSHA developed the PSM standard in 1992 (OSHA, 1992), however incidents still occur in process industries such as chemical plants and refineries (Halim & Mannan, 2018; Kannan et al., 2016). Today, the fast pace of technological changes brings new hazards and increases the complexity and coupling of processes. Therefore, the processes should be considered as socio-technical systems rather than solely technical systems, since the operations need to be supported by the interactions among technological, human, and organizational factors (Leveson, 2004; Perrow, 1999; Rasmussen, 1997). The higher complexity arises because of the interactions among different components, which may follow simple cause and effect relationships in a small group and aggregate into nonlinear and circular relationships leading to emerging failures (Cameron *et al.*, 2017). It is important to ask how we can further prevent incidents and to do that we need to understand incident mechanism more distinctly.

[*] Part of this chapter is reprinted with permissions from "Development of a FRAM-based framework to identify hazards in a complex system" by Mengxi Yu, Noor Quddus, Costas Kravaris, M. Sam Mannan, 2020. *Journal of Loss Prevention in the Process Industries*, 63, Pages 103994, Copyright 2019 by Elsevier Ltd. and from "A data-driven approach of quantifying function couplings and identifying paths towards emerging hazards in complex systems" by Mengxi Yu, Madhav Erraguntla, Noor Quddus, Costas Kravaris, 2021. *Process Safety and Environmental Protection*, 150, 2021, Pages 464-477, Copyright 2021 Institution of Chemical Engineers. Published by Elsevier B.V.

Although randomness may play a role, incidents are often not chance-events that suddenly occur out of nowhere. Incidents follow incubation periods when chains of discrepant events develop and accumulate without notices (Turner & Pidgeon, 1997). If early warnings or weak signals are recognized and managed in time, the incidents can be prevented (Øien *et al.*, 2011). The disastrous vapor cloud explosion at BP Texas City in 2005 was a tragic example that resulted from ignorance of weak signals (Hopkins, 2008; Le Coze, 2008). The investigation of the incident showed multiple weak signals existed in the plant before the incident occurred (Hopkins, 2008). The sight glass for verifying the tower level was not functional for years, and the malfunction of the level transmitter was identified before the start-up. With these failures, the overfill of the raffinate splitter distillation tower could not be noticed easily. Additionally, other weak signals such as overtime shifts, insufficient supervision and staffing made the severity even worse and the hydrocarbon overfill was not recognized until the pressure raised. However, all these weak signals had existed at the plant before the incident but were not recognized and resolved. On the other hand, Pasman (2020) recently gave a series of examples in which signals were recognized on the work floor but leadership/management did not act upon them. In order to prevent incidents, weak signals need to be recognized proactively and resolved as early as possible (Drupsteen & Wybo, 2015).

To recognize weak signals for proactive incident preventions, it is important to understand the definition and characteristics of weak signals. The concept of weak signals was proposed by Ansoff & McDonnell for strategic planning and management, and weak signals were defined as “imprecise early indications about impending, impactful events”

(Ansoff & McDonnell, 1990). Based on the definition, researchers emphasized that weak signals have crucial meanings for the future and provide a threat or opportunity for a business, but they are not mature at the time when they appear (Coffman, 1997; Holopainen & Toivonen, 2012). The concept was applied in multiple domains for anticipating the future, such as technology foresight (Tabatabaei, 2011), defense (Koivisto *et al.*, 2016), and natural disasters (Shelly *et al.*, 2007). In domain of safety, "Weak signal" is not a common terminology and is not distinguished with other terminologies such as precursors and leading indicators. Therefore, the study was aimed to thoroughly understand weak signals in the domain of safety and provide solutions to address challenges of recognizing weak signals.

1.2 Dissertation Organization

The dissertation includes six chapters:

Chapter I introduces the background, motivation, and organization of this dissertation.

Chapter II contains the literature review and summarizes understandings of weak signals in the domain of safety. The literature review shows weak signals have not been defined precisely and formally, therefore the definition of "weak signal" is proposed and presented in the chapter. Lastly, the chapter summarizes the challenges of identifying weak signals and proposes research questions of this study to overcome the challenges.

Chapter III presents a novel framework to model function interactions in process plants. The framework integrates a chemical first-principle model, an equipment performance model and a human performance model based on the system-based technique, i.e., Functional Resonance Analysis Method (FRAM) to develop a hybrid simulator for simulating possible function interactions. The chapter starts with the introduction of FRAM, summarizes the literature review of the models, and presents the methodology to develop the hybrid simulator. The application of the framework to a hypothetical batch polymerization process is presented in the chapter as a case study. Lastly, since the hybrid simulator is utilized as a data generator for the rest of this study in Chapters IV and V, the process of the data generation and the data description is also provided.

Chapter IV focuses on developing a data-driven approach to identify function interactions that lead to emerging hazards. First, function couplings are quantified based on the lift of association rules. Lift confidence intervals are derived from the bootstrap and provide information about uncertainties. The interactions leading to emerging hazards are identified by merging association rules. Lastly, the interactions are graphically represented in FRAM with quantified couplings, guiding people to understand how emerging hazards occur and take preventive measures.

Chapter V presents the developed framework that addresses the challenges from identifying weak signals to responding. FRAM is utilized to determine the information to be collected for observing weak signals, and machine learning techniques, i.e. Random Forest (RF) and Decision Tree (DT) classification models, are developed to evaluate the relevance of underlying weak signals, unfold weak signals and corresponding corrective actions.

Chapter VI summarizes the conclusions of this study and discusses directions for future work.

CHAPTER II

WEAK SIGNALS [*]

2.1 Literature Review

In the domain of safety, Vaughan (1997) defined weak signals as ambiguous information not showing clear threats to safety in the study of the Challenger Shuttle Space incident. The definition was expanded later; a weak signal was defined as "an anomaly that at the time has no clear and direct connection to a potential danger, or an anomaly that only occurs once and does not seem likely to occur again" (Vaughan, 2002). The weak signals were referred to anomalies at technical levels, such as past in-flight anomalies on the Solid Rocket Boosters which warned the flawed design, and the unprecedented cold temperature on the day of launching the Challenger which degraded the resilience of the O-ring (Vaughan, 2002, 2004). The weak signals were not perceived as a threat or managed properly before the incident. Similarly, Weick et al. (1999) defined weak signals as anomalies that are observed during operations, only referring to technical issues. Instead of only treating weak signals as technical issues, Guillaume (2011) pointed out weak signals could be re-occurring technical failures or deficiencies at upper management levels, such as work collaboration and risk perception. Guillaume (2011) defined weak

[*] Part of this chapter is reprinted with permission from "Development of a FRAM-based framework to identify hazards in a complex system" by Mengxi Yu, Noor Quddus, Costas Kravaris, M. Sam Mannan, 2020. *Journal of Loss Prevention in the Process Industries*, 63, Pages 103994, Copyright 2019 by Elsevier Ltd. and from "A data-driven approach of quantifying function couplings and identifying paths towards emerging hazards in complex systems" by Mengxi Yu, Madhav Erraguntla, Noor Quddus, Costas Kravaris, 2021. Process Safety and Environmental Protection, 150, 2021, Pages 464-477, Copyright 2021 Institution of Chemical Engineers. Published by Elsevier B.V.

signals as the information that can anticipate events but are ambiguous for interpretations. Additionally, Hollnagel (2004) proposed a definition of weak signals and revealed insights into why weak signals are ambiguous. Within a complex system, multiple functions are involved ranging from technological functions, human functions, to organization functions. Each function has its own performance variabilities. A weak signal is ambiguous since it does not cause detectable effects until it combines with noise and its constitution to a hazard is only amplified by the combination (Hollnagel, 2004).

"Weak signal" is not common terminology in the domain of safety. Besides the definitions mentioned above, there are multiple terminologies resembling a similar spirit, such as precursors, early warning signals/signs and leading indicators. The terminologies bring more confusions to understand what weak signals are. For example, precursors have various definitions (Carroll, 2004; Körvers, 2004; Kunreuther *et al.*, 2004; Saleh *et al.*, 2013), and one of them saying precursors are "re-occurring deviations in operational process" (Körvers, 2004), which is the same as the technical weak signals that were defined by Guillaume (2011). "Early warning signal" is treated as an interchangeable terminology of "weak signal", which is an early warning of a precursor (Luyk, 2011). Additionally, a leading indicator is another form of an early warning to evaluate overall safety or risk in a system, as it is a symptom of a deficiency in the safety management system, e.g., lack of training of new employees, that in due time could contribute to accident. However, it is not interchangeable with a weak signal (Øien *et al.*, 2011), but monitoring indicators over time can be used as a trend measure of safety culture and organizational health. To utilize leading indicators directly as signals for incident prediction is not feasible, since the correlation between most leading indicators and event realization is unknown, vague and unpredictable (Körvers, 2004; Luyk, 2011; Øien *et al.*, 2011). Instead, the number of weak signals noticed could be a leading indicator of an organization (Øien *et al.*, 2011).

Even though weak signals were defined differently, the characteristics of weak signals that were addressed in literature are consistent. First, weak signals emerge a long time before incidents and exist as early indications of potential risks of impactful events. They can be too early to be precise, but they provide more opportunities to respond (Guillaume, 2011; Luyk, 2011). Second, it is difficult to recognize weak signals since they cannot be interpreted in isolation. A weak signal seems irrelevant and uncertain to defeat a safety system, while its impact becomes noticeable when it combines with other signals (Brizon & Wybo, 2009; Hollnagel, 2004; Luyk, 2011). Besides, noise brings challenges to connect multiple weak signals. The weak signals only can be interpreted by appropriate filtering and processing (Brizon & Wybo, 2009; Guillaume, 2011).

2.2 Proposed Definition of Weak Signals

Since weak signals have not been defined precisely in the domain of safety, a formal definition of weak signals needs to be proposed. Among the existing definitions that were mentioned in the previous section, the definition proposed by Hollnagel (2004) provided the most clarity through emphasizing multiple sources of weak signals and indicating that a weak signal alone could not cause impactful events. However, Hollnagel (2004) defined a weak signal and noise as relative terms depending on focuses and stated detectable hazard was caused by the interaction between a weak signal and its noise. According to the Cambridge dictionary, a signal is something that “gives a message or a warning”. It is more sensible by treating anything representing a sign of a hazard as a signal, instead of noise. In view of the definition by Hollnagel (2004) and the understandings of weak signals in other literature, this study proposed new definitions of weak signals and noise:

- Weak signals are performance variabilities of technological, organizational, or human functions whose interactions combine clues or signs giving rise to early prediction of a future unexpected event/incident.
- Noise is performance variabilities of the functions, which have no or negligible impacts on a future event and do not provide information about the future event.

Compared to the definition by Hollnagel (2004), the proposed definition differentiates signals and noise depending on whether they contribute to predicting future events. Additionally, the definition of weak signals addresses three aspects. First, weak signals are not restricted to be at technological level, and it can be performance variabilities of human or organization functions. Second, weak signals exist as combinations. An individual weak signal does not cause noticeable impacts until it interacts with others. Otherwise, it should be treated as a strong signal. Lastly, weak signals are early predictors. Even though “early” is a characteristic that has been well recognized in existing literature, the scope of weak signals in terms of how early is early was rarely clarified. Only Luyk (2011) explicitly defined weak signals needed to be early enough to predict incident precursors, which were precisely defined as “a chain of adverse events flowing an initiating off-nominal event and that can lead to an accident” (Saleh et al., 2013). According to the concepts of weak signals and precursors, the study further defined the characteristic “early” of weak signals precisely. Since precursors are adverse events leading to an incident, in order to be early enough to indicate the precursors, weak signals are restricted to be conditions that can warn for adverse events. For example, relief valve that fails is a precursor event, but unqualified maintenance of the relief valve is a condition which can be a weak signal of the failure.

2.3 Challenges of Identifying and Responding to Weak Signals

2.3.1 Complex Interactions

The first challenge of identifying and responding to weak signals is related to the characteristic that weak signals exist as combinations. Due to complex interactions between functions in a socio-technical system, recognizing weak signals based on individuals’ knowledge about how they interact can be intellectually unmanageable. Therefore, systematic modeling of function interactions that lead to a hazardous scenario is a prerequisite.

To address interactions among a complex system, system-based incident analysis techniques such as accident causation model (Accimap) (Rasmussen, 1997), Systems Theoretic Accident Modeling and Processes (STAMP) (Leveson, 2011) were developed to investigate the emergent failures. A few of them were further extended for proactive hazard identifications and eventually overcame some limitations of the conventional hazard identification techniques. Based on the original accident model Accimap, Rasmussen *et al.* developed a proactive strategy for risk management using “generic Accimaps” to understand the interactions leading to different critical events (Rasmussen, 1997; Rasmussen & Suedung, 2000). Another system-based hazard identification technique, Systems Theoretic Process Analysis (STPA), was developed by Leveson based on STAMP (Cameron *et al.*, 2017; Leveson, 2011). With STPA, hazards and the corresponding interactions among the hierarchy levels can be identified. Instead of modeling interactions based on a hierarchical structure, Functional Resonance Accident Model (FRAM) models interactions by decomposing a complex system into functions (Hollnagel, 2017; Patriarca *et al.*, 2020). FRAM is widely used to describe complex systems with three categories of functions, i.e., technological, organization, and human functions. The rationale behind FRAM is that each function in the system has its performance variabilities, and the emerging effect of the performance variabilities can lead to a resonance phenomenon, which may cause negative impacts on the system. FRAM has flexibilities to model and describe an event that has occurred or an event that may occur in the future. However, these techniques have low degree of automation to describe how the interactions in a system lead to hazards, and lack of quantification to describe performances of components in a system and how their interactions impact the system behavior. Among the system-based techniques, FRAM stands out with evolutions over time to improve its automated and quantitative features. Thus, FRAM and its limitations are reviewed and presented in subchapter 2.3.1.1. The evolutions of FRAM to address the limitations are presented in subchapter 2.3.1.2.

2.3.1.1 FRAM and Its Limitations

Behaviors of a complex system are emerging outcomes that are resulted from the aggregated performance of individual functions, and usually cannot be explained by simple causal-effect logic. These functions (either technological or organization or human) have their performance variabilities, which means they can adjust their behaviors to reach a specific objective of the system. The interactions of their performance variabilities could cause unexpected events in the system. Steps to identify the functional resonance through FRAM has been detailed out by Hollnagel (2017) thus they are briefly described as follows, along with the associated limitations:

Step 1 Identify and describe functions: At this step, all involved functions are identified and the system is decomposed into the functions. Each function is represented by a hexagon and has six aspects describing the functions (i.e. Input, Precondition, Resource, Time, Control, and Output). The interactions among the functions are represented by linking the aspects among different functions. With the function decomposition method, functions and their associated interactions can be identified step-by-step in a complex system. As the example of FRAM shown in Figure II-1, the system involves five functions. A connection between two functions shows their interaction, which is called “upstream-downstream coupling” in FRAM. For example, Function 1 controls the performance of Function 2. Those functions in shades and with output aspects only are called background functions. These functions define the boundary of the system.

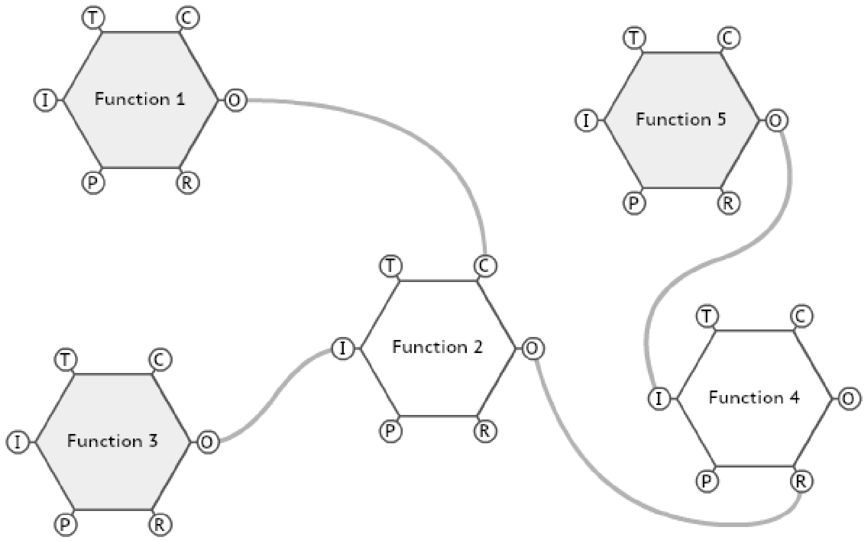


Figure II-1 An example of FRAM.

Step 2 Identification of performance variabilities: Performance variabilities of every single function are identified and described by time and precision of the function output. Variabilities in terms of both time and precision are described by rough linguistic levels for qualitative understanding. For example, the time variability can be described as “too early” and the variabilities of the precision can be “precise”. FRAM does have an elaborate method to characterize the variabilities by more aspects, but the aspects are still described by linguistic levels. There has not been any limitation of using the qualitative way to define performance variabilities so far, but it brings the challenges to the next step.

Step 3 Aggregation of performance variabilities: Output performance variabilities of upstream functions can serve as any aspect except the outcomes of the downstream functions. They can either damp, increase, or not impact the outcome performance variabilities of the downstream functions. The limitation of qualitatively defining performance variables in the previous step makes the understanding of the aggregated effects challenging when a downstream function is affected by multiple upstream functions thus requires the quantifications of the damping or amplifying effects from its upstream functions.

Step 4 Consequence and analysis: After performance variabilities are aggregated, emerging hazards and their corresponding uncontrollable performance variabilities can be identified. The measures to eliminate or mitigate the hazards can be carried out accordingly.

In summary, three main limitations of FRAM are 1) lack of quantitative way to describe performance variabilities and function couplings, 2) each instantiation of FRAM only can represent an individual scenario, 3) the identification of interactions leading to the hazards highly depends on expert judgements and the completeness cannot be guaranteed (Frost & Mo, 2014).

2.3.1.2 Evolutions of FRAM

To address the first limitation regarding lack of quantitation, Rosa *et al.* (2015, 2017) assigned weight factors based on questionnaires to identify the upstream function which most impacts a downstream function. Patriarca *et al.* developed a semi-quantitative approach to use FRAM for hazard analysis in a sinter plant and an aviation system (Patriarca *et al.*, 2017a; Patriarca *et al.*, 2017b). Ranking scores and probabilistic distributions were assigned to the levels of timing and precision for functions, and the Monte Carlo simulation was used to obtain indices for functions in different scenarios. A similar approach was also used to develop a matrix representation of function couplings by Patriarca *et al.* (2018) and Köpke *et al.* (2020). The semi-quantitative method provided relative understanding of coupling effects, but it was still based on the linguistic description of time and precision. Besides, the studies only focused on quantifying the coupling between a single upstream function and its downstream function, rather than aggregated couplings from multiple upstream functions. Due to interactions in a complex system, an aggregated effect of multiple upstream functions can be stronger than that of any of them. Meanwhile, it is also possible that the deviation of an upstream function can be dampened by other functions, thus the impact of the upstream function is weakened. Appropriate quantification of performance variabilities and upstream-downstream couplings is essential to understand interactions leading to emerging hazards, especially in process industries where productions involve nonlinear and complex kinetics.

Additionally, to overcome the second and the third limitations, FRAM has been integrated with model checking (Duan *et al.*, 2015; Yang & Tian, 2015; Zheng & Tian, 2015; Zheng *et al.*, 2016). Model checking, a formal verification technique, can identify multiple potential paths leading to hazards, and it can be used to understand interactions systemically and automatedly through computing, rather than sole reliance on expert judgements. However, the studies have limitations in terms of qualitatively defining states as finite states. The combination of FRAM and model checking are applicable in the domains that are driven by discrete events, such as the aviation and manufacturing processes, but they are limited to be applied in process industries. Besides, researchers conducted studies of simulating the whole FRAM to identify possible interactions in complex systems. Smoczyński *et al.* (2018) simulated function interactions in a transportation system to speed up the boarding process and Asadzadeh and Azadeh (2014) simulated the interactions between human and organization functions to understand how maintenance activities were impacted. However, the studies are only applicable to the systems with discrete events.

Operations in process industries are usually hybrid systems involving continuous production and discrete events such as process and operator behaviors. Precise descriptions of such operations need to be supported by kinetics and thermodynamics, rather than by simply using qualitative finite states. FRAM is an outstanding technique for understanding complex systems, but to take advantage of it to improve process safety, a novel methodology based on FRAM is needed.

2.3.1.3 Research Question 1

As discussed above, applications of FRAM in process industries still have limitations even though researchers have integrated the original FRAM with other techniques to improve its automation and quantification. Thus, the first research question to be answered in this study is:

- How can the system-based technique, FRAM, describe function interactions leading to potential hazards in process industry quantitatively and automatically?

2.3.2 Existence with Noise

Another challenge of identifying weak signals is related to its existence with noise. Given the rapid development of computing technologies in the past decades, digitized process plants have the capability to collect a tremendous amount of various types of data from process operations, control rooms, business and information systems (Pasman, 2020; Qin, 2014; Xu *et al.*, 2015). The abundance of data creates values for industries such as tailoring products and services, but also brings a lot of noise making weak signals hardly be picked up and connected.

2.3.2.1 Life cycle of weak signals

The life cycle from identifying to acting on weak signals consists of three stages (Brizon & Wybo, 2009; Holopainen & Toivonen, 2012; Rossel, 2009).

1. First, weak signals need to be observed in the system. This stage requires an organization to collect data of weak signals so they can be monitored for further evaluations.
2. The second stage is evaluating the relevance of weak signals. At this stage, weak signals need to be interpreted based on knowledge and context to recognize their indications of future events. In most cases, one who interprets weak signals is not the one who has power to act. Therefore, the relevance needs to be evaluated so that one can transmit the existence of weak signals to decision-makers when one thinks the relevance is strong enough or it meets any criterion in organizational standards.
3. Once the existence of weak signals is transmitted to decision-makers, decision-makers can determine whether to act on the weak signals and decide how to prioritize actions based on the significance of weak signals.

Unfortunately, coexistence of weak signals and noise in the large pool of information in today’s digitized plants brings challenges throughout the entire life cycle.

First, information to be collected in a system depends on the purpose of uses and knowledge. Selecting appropriate information to be collected and observed for identifying weak signals can screen out large portion of noise, as well as ensure all the potential sources of weak signals that are relevant to a selected hazard become observable in the first stage of the life cycle. Therefore, a technique is necessary to guide organizations to decide what information to be collected and observed for identifying weak signals. As discussed earlier, FRAM is a system-based technique to model function interactions, thus it could be a solution to address the challenge in the first stage by identifying those functions that are relevant to a selected hazard as the potential sources of weak signals.

Next, even though potential sources of weak signals are observable, it is still subjective to identify actual weak signals and evaluate their relevance based on awareness of individuals and their knowledge about function interactions in the system. Only a few studies (Brizon & Wybo, 2009; Guillaume, 2011; Körvers, 2004; Luyk, 2011) were conducted aiming to help industries improve abilities to identify weak signals, but the studies mainly focused on improving organizational management qualitatively, instead of developing techniques to identify weak signals and quantify their relevance to hazards proactively.

Without solutions to overcome the challenges for the first two stages, it cannot guarantee weak signals are identified and transmitted to decision-makers with an explanation of significance. Furthermore, it is also doubtful whether responses to weak signals are effective without appropriate interpretations. Thus, it requires advanced techniques to help industries identify weak signals from abundance of data and provide guidance about effective responses to prevent incidents proactively.

2.3.2.1 Learning from historical data

Learning from past anomalies and incidents plays a critical role in identifying and interpreting weak signals (Brizon & Wybo, 2009; Guillaume, 2011), but advanced techniques are necessary to extract the knowledge from massive historical data. Machine learning techniques have caught increasing attention in the past decades for item classification and pattern recognition. They learn patterns from existing data then make predictions on future events (Ge *et al.*, 2017; Han *et al.*, 2011; Xu *et al.*, 2015). In process industry, machine learning techniques are widely applied using process data for process and quality monitoring, fault identification and as soft sensor (Qin, 2014). However, other valuable information to identify weak signals such as equipment failures, quality-related issues, and performance measurement (Haji-Kazemi & Andersen, 2013; Körvers, 2004) was seldom utilized to predict process incidents. Instead, such information was commonly used for preventing occupational incidents. Table II-1 summarizes the studies which applied classification algorithms to extract knowledge from historical data. Most studies in the table utilized only historical incident data for understanding causes and consequences of past incidents, rather than predicting occurrences of potential incidents. On the other hand, data of both safe operations and incidents was used by (Goh & Chua, 2013; Goh *et al.*, 2018; Poh *et al.*, 2018; Sarkar *et al.*, 2018) and the studies showed promising results of predicting incident occurrences relying on potential weak signals such as safety management elements. Therefore, applications of machine learning techniques in process industries should consider a wider range of data beyond process data to recognize and respond to existing weak signals in plants.

2.3.2.2 Research Question 2

According to the discussions above, FRAM could be a solution to identify potential sources of weak signals and preliminarily screen out noise. Additionally, past studies of applying machine learning techniques for occupational safety showed great potentials of machine learning techniques to identify and respond to weak signals in process industries. Therefore, the second research question to be answered in this study is:

- How FRAM and machine learning techniques can be utilized together to address challenges throughout the life cycle of weak signals?

Table II-1 Applications of machine learning techniques for preventing occupational incidents

Literature	Domain	Data	Outcome	Algorithms
Bevilacqua *et al.* (2008)	Petrochemical	Incident data only	Predict incident categories, identify underlying causal factors	DT
Rivas *et al.* (2011)	Mining and construction	Incident data only	Predict event types, identify underlying causal factors	DT, SVM, LR, BN
Goh and Chua (2013)	Construction	Inspection data, Incident data	Predict incident occurrences and severities, identify critical safety management elements	NN
Mistikoglu *et al.* (2015)	Construction	Incident data only	Predict incident severities, identify underlying causal factors	DT
Sarkar *et al.* (2016)	Steel plant	Incident data only	Predict incident categories, identify underlying factors	DT
Tixier *et al.* (2016)	Construction	Incident data only	Predict incident consequences	ET
Goh *et al.* (2018)	Construction	Survey and observation	Predict safety class of workers, identify underlying causal factors	DT, NN, KNN, SVM, LR, NB, ET
Poh *et al.* (2018)	Construction	Inspection data, Project-related data, Incident data	Predict incident occurrences	DT, KNN, SVM, LR, ET
Sarkar *et al.* (2018)	Construction	Inspection data, Incident data	Predict incident occurrences, identify underlying causal factors	ET
Kang and Ryu (2019)	Construction	Incident data only	Predict incident categories, identify underlying causal factors	ET
Kakhki *et al.* (2019)	Agribusiness	Incident data only	Predict incident categories	SVM, NB, ET

DT: decision tree, ET: ensemble model based on decision trees, KNN: k-nearest neighbors, SVM: support vector machine, LR: logistic regression, NB: Naïve Bayes, BN: Bayesian networks, NN: neural network

CHAPTER III

MODELING FUNCTION INTERACTIONS BASED ON FRAM [*]

3.1 Introduction

As indicated by Research Question 1, function interactions in process plants need to be automated and quantified, which requires simulations using mathematical models. This chapter presents the development of a framework to show how a human performance model, an equipment performance model and a first-principal model of a chemical process can be integrated into a hybrid-simulator to simulate possible interactions between functions. This chapter contains reviews of existing human performance models and equipment performance models and the selection of appropriate models to be utilized for simulations. First-principle models depend on processes thus it is selected for the specific process of a case study. These models are theoretical and only for demonstrating the framework, but they can be enhanced based on real-world information.

On the other hand, as indicated by Research Question 2, machine learning techniques need to be developed for identifying weak signals and a data source is needed for extracting such patterns. When historical data collected from plants is unavailable or limited, synthetic data generated from the hybrid simulator can be a time- and cost-

[*] Part of this chapter is reprinted with permission from "Development of a FRAM-based framework to identify hazards in a complex system" by Mengxi Yu, Noor Quddus, Costas Kravaris, M. Sam Mannan, 2020. *Journal of Loss Prevention in the Process Industries*, 63, Pages 103994, Copyright 2019 by Elsevier Ltd. and from "A data-driven approach of quantifying function couplings and identifying paths towards emerging hazards in complex systems" by Mengxi Yu, Madhav Erraguntla, Noor Quddus, Costas Kravaris, 2021. Process Safety and Environmental Protection, 150, 2021, Pages 464-477, Copyright 2021 Institution of Chemical Engineers. Published by Elsevier B.V.

effective alternate. Therefore, without data from real plants in this study, synthetic data of a case study process was generated from the developed hybrid-simulator.

The implementation of the hybrid simulator for data generation is explained in details in this chapter and the description of the synthetic data is provided. The synthetic data will be further used as the data source for the work of Chapter IV and Chapter V.

3.2 Framework Overview

Figure III-1 overviews the framework to model function interactions in a complex system based on FRAM. The functions that are involved in the system and their performance variabilities were first identified. Due to limitations of the FRAM application in process industry, only Step 1 and Step 2 of FRAM were used. Additionally, instead of solely using linguistic levels to describe performance variabilities in the Step 2, quantitative descriptions were allowed if they were applicable. After functions and their performance variabilities were identified, an equipment performance model, a human performance model and a chemical first-principle model were integrated to describe interactions between the functions. Then, a hybrid simulator that integrates both discrete events and continuous process, was built to simulate the interactions within the complex system. The stochastic performance of functions was simulated based on sampling strategies. Therefore, the hybrid simulator could simulate interactions between functions and generate synthetic data for further analysis.

Besides, the overview provides the flowchart of model integration. In a complex system such as a chemical plant, organization functions impact human functions, which further affect the physical chemical process. The human performance model namely Cognitive Reliability and Error Analysis Method (CREAM) was used to describe such impacts. Inputs to the CREAM model are levels of performance shaping factors (PSFs), which describe performances of organization functions. Based on the CREAM model, occurrences of generic failures and their probabilities can be obtained. Additionally, performance variabilities of human functions corresponding to a specific generic failure can be identified. Probabilities of generic failure occurrences and corresponding performance variabilities are used during simulation to sample human performances, which directly impact process variables in the chemical process. Similarly, organization functions such as Preventive Maintenance (PM) impact equipment performances, which further impact the physical chemical process. In the study, a PM model namely age-reduction based imperfect model was applied to describe the impact of PM on equipment performance. Given a PM schedule of equipment as the input, the equipment performance model was used to obtain corresponding cumulative failure probabilities during the operating life of the equipment. The cumulative failure probabilities were used in simulations to sample equipment performance, which would directly impact process variables in the chemical process. Finally, given process variables depending on human and equipment function, first principle model was used to model interactions of process variables.

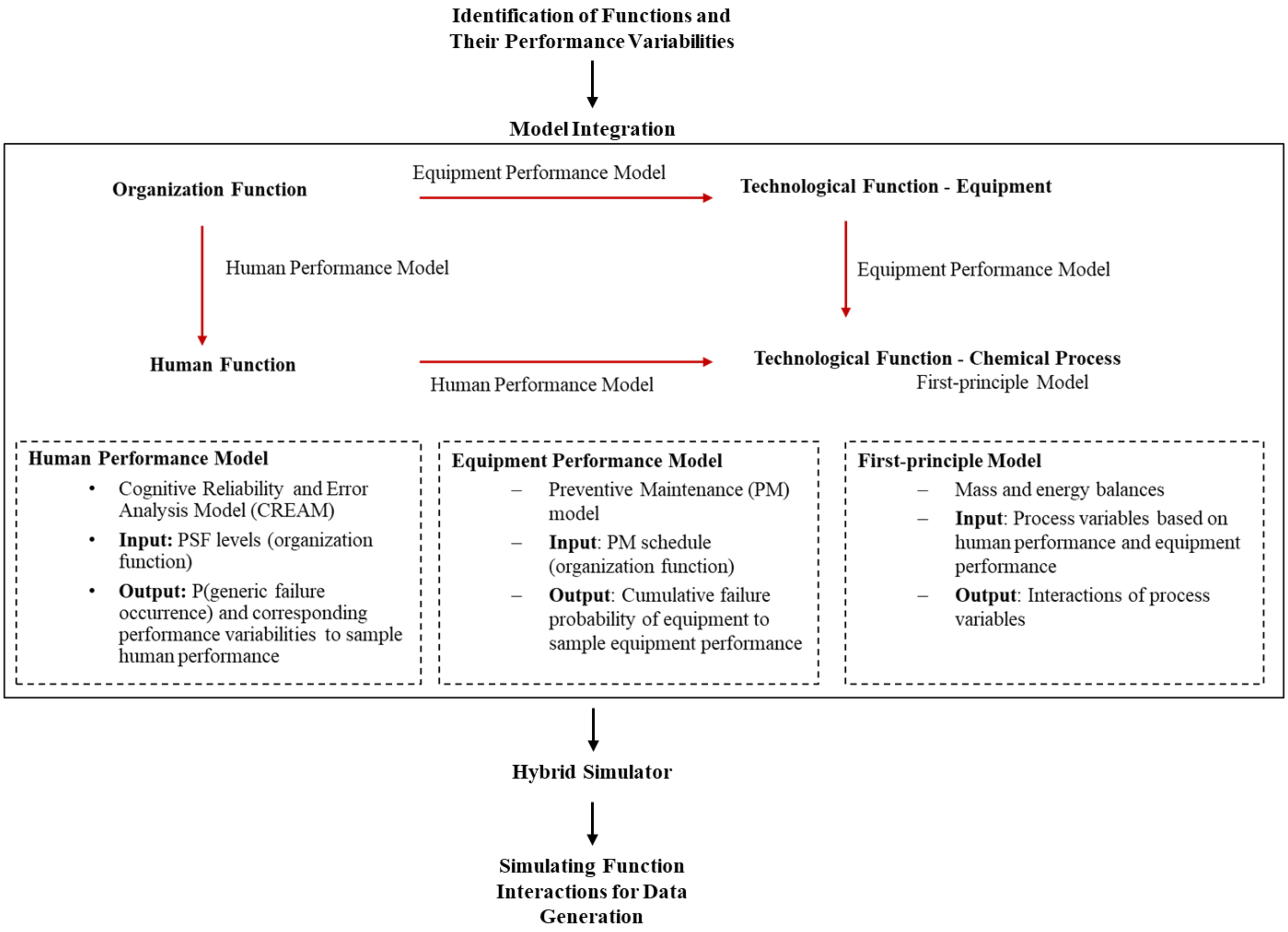

Identification of Functions and Their Performance Variabilities
Model Integration
Organization Function
Equipment Performance Model
Technological Function - Equipment
Human Performance Model
Equipment Performance Model
Human Function
Human Performance Model
Technological Function - Chemical Process
First-principle Model
Human Performance Model
• Cognitive Reliability and Error Analysis Model (CREAM)
• Input: PSF levels (organization function)
• Output: P(generic failure occurrence) and corresponding performance variabilities to sample human performance
Equipment Performance Model
– Preventive Maintenance (PM) model
– Input: PM schedule (organization function)
– Output: Cumulative failure probability of equipment to sample equipment performance
First-principle Model
– Mass and energy balances
– Input: Process variables based on human performance and equipment performance
– Output: Interactions of process variables
Hybrid Simulator
Simulating Function Interactions for Data Generation

Figure III-1 Framework to model function interactions in a process plant

3.3 Human Performance Model

Humans have critical functions in the process industries such as starting-up a process, monitoring and responding to an abnormal situation. Failures of these functions may lead to disastrous incidents. Statistics shows that most of the incidents in the process industries involve human failures (Kariuki & Löwe, 2007). Many techniques on human reliability analysis (HRA) have been developed for years to analyze human failures to define failure scenarios and their probabilities. However, the HRA is always conducted separately with conventional process hazard analysis, and only a few studies (Ávila *et al.*, 2013; Kariuki & Löwe, 2007; Kennedy & Kirwan, 1998; Schurman & Fleger, 1994) integrated them. In the study, existing HRA techniques (Bell & Holroyd, 2009; CCPS, 1994; Embrey *et al.*, 1984; Gertman *et al.*, 2005; Hollnagel, 1998; Jahangiri *et al.*, 2016; Kirwan, 1992a, 1992b, 1996; Massaiu & Paltrinieri, 2016; Yang *et al.*, 1997) were reviewed.

Cognitive Reliability and Error Analysis Method (CREAM) was selected for this study to model human performance based on four criteria: 1) Performance shaping factors (PSFs): Generally, an HRA starts with a nominal human failure probability of a (sub-) task, then the nominal probability would be adjusted according to PSFs. PSFs may be named in different ways. For example, they are called as error-producing conditions (EPCs) in Human Error Assessment and Reduction Technique (HEART), or Common Performance Conditions (CPCs) in CREAM. However, they are conceptually similar in terms of adjusting probabilities of human failures. PSFs that are covered in different techniques vary. To select a technique that is more practical to be used, the technique with a concise set of PSFs to describe organization performance is desired; 2) Dependencies among PSFs are considered; 3) Flexibility: an HRA technique with rigorous instructions and low flexibility is desirable to ensure consistent analysis; 4) Application in process industries: many HRA techniques are developed for nuclear plants specifically such as Technique for Human Error-Rate Prediction (THERP) and Justification of Human Error Data Information (JHEDI). Thus, the desirable technique is expected to have extended applications in process industry. The characteristics of some common techniques based on the four criteria are summarized in Table III-1.

CREAM allows detailed analysis of human failure types based on cognitive functions instead of only categorizing human behaviors by binary success and failure. The CPCs correspond to organization functions and their levels will lead to human failures with different probabilities. The steps to use CREAM are identifying cognitive demands of the task to be analyzed, identifying most likely cognitive failures, and determine the corresponding failure probabilities based on CPC levels. Potential failure types with their nominal probabilities by different cognitive functions are provided in Table III-2.

After appropriate failure types are identified, the corresponding nominal probabilities can be adjusted by multiplying weight factors of the CPC levels. The weight factors, which are greater than 1, increase nominal probabilities, while those which are less than 1 reduce nominal probabilities, and those which are equal to 1 indicate that nominal probabilities will not be changed. The weight factors are provided in Table III-3. The probability of a generic failure type is calculated by Equation III-1. Inversely, the probability of not encountering a failure type is calculated by Equation III-2.

Probability (failure type j) = $\prod_{\mathrm{i}=1}^{9} \mathrm{WF_i} * \mathrm{NP_j}$ (Equation III-1)

Probability (no failure type j) = 1 – Probability (failure type j) (Equation III-2)

where $\mathrm{WF_i}$ is weight factor of the applicable level of CPC i and $\mathrm{NP_j}$ is the nominal probability of failure type j

Table III-1 Comparisons among HRA Techniques

HRA Technique	PSF	PSF Dependency	Flexibility	Application in Process Industries	Reference
SLIM-MAUD	No defined set of PSFs	Considered	High	Yes	(CCPS, 1994; Embrey *et al.*, 1984; Kirwan, 1996)
THERP	Over 60 PSFs	Considered	High	Yes	(Kirwan, 1992a, 1992b, 1996)
JHEDI	PSF determined by questions	Not considered	Low	No	(Bell & Holroyd, 2009; Kirwan, 1996)
HEART	32 EPCs	Not considered	High	Yes	(Bell & Holroyd, 2009; Kirwan, 1996; Massaiu & Paltrinieri, 2016)
SPAR-H	8 PSFs	Not considered	Low	No	(Gertman *et al.*, 2005; Jahangiri *et al.*, 2016)
CREAM	9 CPCs	Considered	Low	Yes	(Hollnagel, 1998)
HCR	No defined set of PSF	Not considered	High	No	(Yang *et al.*, 1997)

Table III-2 Nominal probabilities of generic failure types by cognitive functions (Hollnagel, 1998)

Cognitive Function	Generic Failure Type	Nominal Probability
Observation	O1 Wrong object observed	0.003
	O2 Wrong identification	0.070
	O3 Observation not made	0.070
Interpretation	I1 Faulty diagnosis	0.200
	I2 Decision error	0.010
	I3 Delayed interpretation	0.010
Planning	P1 Priority Error	0.010
	P2 Inadequate Plan	0.010
Execution	E1 Action of wrong type	0.003
	E2 Action of wrong time	0.003
	E3 Action on wrong object	0.0005
	E4 Action out of sequence	0.003
	E5 Missed action	0.030

Table III-3 Weight factors of CPC levels by cognitive functions (Hollnagel, 1998)

CPC	Level	O	I	P	E
Adequacy of organization	Very efficient (1)	1	1	0.8	0.8
	Efficient (2)	1	1	1	1
	Inefficient (3)	1	1	1.2	1.2
	Deficient (4)	1	1	2	2
Working conditions	Advantageous (1)	0.8	0.8	1	0.8
	Compatible (2)	1	1	1	1
	Incompatible (3)	2	2	1	2
Adequacy of MMI and operational support	Supportive (1)	0.5	1	1	0.5
	Adequate (2)	1	1	1	1
	Tolerable (3)	1	1	1	1
	Inappropriate (4)	5	1	1	5
Availability of procedure/plans	Appropriate (1)	0.8	1	0.5	0.8
	Acceptable (2)	1	1	1	1
	Inappropriate (3)	2	1	5	2
Number of simultaneous goals	Fewer than capacity (1)	1	1	1	1
	Matching current capacity (2)	1	1	1	1
	More than capacity (3)	2	2	5	2

*O: observation, I- interpretation, P- plan, E- execution

Table III-3 Continued

CPC	Level	O	I	P	E
Available time	Adequate (1)	0.5	0.5	0.5	0.5
	Temporarily inadequate (2)	1	1	1	1
	Continuously inadequate (3)	5	5	5	5
Time of day	Day-time (adjusted) (1)	1	1	1	1
	Night-time (unadjusted) (2)	1.2	1.2	1.2	1.2
Adequacy of training	Adequate, high experience (1)	0.8	0.5	0.5	0.8
	Adequate, low experience (2)	1	1	1	1
	Inadequate (3)	2	5	5	2
Crew collaboration	Very efficient (1)	0.5	0.5	0.5	0.5
	Efficient (2)	1	1	1	1
	Inefficient (3)	1	1	1	1
	Deficient (4)	2	2	2	5

*O: observation, I- interpretation, P- plan, E- execution

One important point of discussion is that whether CREAM is obsolete since the original developer disclaimed that CREAM was outdated in 2012. The reason was that it only considers one component, human, in a complex system, and only considers human failures, instead of the performance variabilities of humans. However, it was still selected in the current study because of the way it was used. The study was based on FRAM to understand the system qualitatively from the functional viewpoint. CREAM was only used as a part of the framework. Interactions among different functions in a complex system were modeled by integrating CREAM with an equipment performance model and a chemical process model based on the first principles. Additionally, even though the CREAM only considers generic failure types, it provides enough guidance for people to understand how generic failures occur. Thus, the study first identified the generic failures based on the original CREAM, then analyzed them further to identify the performance variabilities in terms of failure as shown in the case study, and the successes of humans were also considered as indicated by Eqn. III-2. It should be noted that CREAM is still commonly recognized in different fields since 2012 (Akyuz & Celik, 2015; Chen *et al.*, 2019; Shokria, 2017; Zhou *et al.*, 2018).

3.4 Equipment Performance Model

Malfunctions or failures of critical equipment in chemical processes can lead to hazardous loss of containment. The organizations in plants are responsible for planning maintenance activities to maintain equipment performance and prevent malfunctions and failures. Most of the incidents in plants are contributed by improper maintenance (Nguyen *et al.*, 2008) and less than 80% of scheduled preventive maintenances (PMs) for safety-critical equipment are carried out as planned in process industries according to the insights from the industry professionals (Brewer, 2016). Thus, one part of the study was to model equipment performance variabilities with different schedules to carry out PMs.

To study how equipment performance is impacted by PMs in practice, the study first reviewed existing PM models. Based on the physical conditions of equipment that could be restored by a PM, PMs typically can be categorized to perfect maintenance (as good as new), imperfect maintenance, minimal maintenance (as bad as old), worse maintenance (worse than old), and worst maintenance (make equipment fail) (Carlo & Arleo, 2017; Pham & Wang, 1996). Among the categories, imperfect maintenance is the most realistic one since it restores the equipment to the state between as good as new and as bad as old. Many studies of modeling imperfect maintenance have been developed (Carlo & Arleo, 2017; Pham & Wang, 1996; Valdez‐Flores & Feldman, 1989). The models are categorized into hazard-rate based, age-reduction based, hybrid model, and damage level reduction-based model, based on different theoretical assumptions. Due to inadequate knowledge of the underlying physics of different equipment, models need to be validated based on failure data before being applied in reality (Liu *et al.*, 2012). Thus, age-reduction based imperfect PM model was used in the study as an example to illustrate the framework.

Age-reduction factor is used in age-reduction based imperfect model to indicate how PM improves equipment performance by reducing its effective age. Each time when i_{th} PM is conducted, the effective age of the equipment y_i is reduced to $\xi_i y_i$ where ξ_i is the age-reduction factor of the i_{th} PM, in the range of [0,1] (Lin *et al.*, 2015). Even though the original purpose of the imperfect PM models is to develop optimal PM schedules, it can be used to study the impact of not complying optimal PM schedules on equipment performance by adjusting the age-reduction factor accordingly.

In the study, equipment performance under two extreme cases was studied. The two cases were 1) all the PMs being carried out as scheduled, and 2) no PM being carried out. In the first case, the age-reduction factor is assumed as a constant ξ to obtain the optimal PM schedule, while in the second case, the age-reduction factor is assumed as 1 to indicate that there is no effective age reduction since no PM is conducted. Additionally, common assumptions of the model are 1) providing minimal repair with negligible repair time if a failure occurs after a PM but before the next one. 2) assuming the equipment is deteriorating and the failure rate is monotonically increasing with time. With the assumptions, the failure rate of equipment is expressed as $\lambda(\mathrm{t}) = \alpha\beta t^{\beta-1}$, where α is the scale parameter of the Weibull power law, and β is the deterioration rate. β is assumed to be greater than 1 to model the deteriorating stage of equipment. The relevant formulas to simulate the two extreme cases are illustrated as follows.

Case 1 PMs being carried out as the optimal PM schedule (Lin et al., 2015) : The optimal PM schedule ensures that the conditional reliability of the equipment is greater than its predetermined critical reliability threshold r_c. Given r_c, the effective age y_i that is right before the i_{th} PM is carried out is calculated by Equation III-3. Equation III-4 is used to calculate the time interval x_i between the $(i-1)_{th}$and the i_{th} PM.

$$y_i = [-\frac{\ln(r_c)}{\alpha}]^{1/\beta} \qquad \text{(Equation III-3)}$$

$$x_i = (1-\xi)[-\frac{\ln(r_c)}{\alpha}]^{1/\beta} \qquad \text{(Equation III-4)}$$

The cumulative failure probability (F) after $(i-1)_{th}$ PM in terms of effective age t is

$$F = 1 - R = 1 - \exp[-\alpha(t^{\beta} + (i-1)(1-\xi^{\beta}){y_i}^{\beta})] \qquad \text{(Equation III-5)}$$

for $\xi y_{i-1} \leq t \leq y_i$, where $i = 1,2,3,\ldots,N_{pm}$, and N_{pm} is the number of the PMs that are scheduled during the operating time, R is the cumulative reliability.

Expected number of equipment failures during the operating time is

$$N(t) = \alpha \sum_{i=1}^{Npm}[{y_i}^{\beta} - \xi\, {y_i}^{\beta}] \qquad \text{(Equation III-6)}$$

Effective age (t) is converted to the operating time (T) by the piecewise function in Equation III-7.

$$T = (i-1) * x_i + t - \xi y_i,\ for\ \xi y_{i-1} \leq t \leq y_i \qquad \text{(Equation III-7)}$$

Case 2 no PM being carried out: When no PM is being carried out, the age-reduction factor ξ is equal to 1 indicating no age reduction, thus the effective age t is the same as the real operating time. The general formulas in Case 1 can be simplified as follows.

$$F = 1 - R = \exp[-\alpha t^{\beta}] \qquad \text{(Equation III-8)}$$

$$N(t) = \int_0^t \lambda(t)dt = \alpha t^{\beta} \qquad \text{(Equation III-9)}$$

3.5 First-principle Model of Polymerization Process

A first-principle model describes interactions of the process variables in a chemical process. In the study, a hypothetical industrial-scale batch process of poly-methyl methacrylate (PMMA) was selected as a case study due to its runaway hazard during the free radical polymerization process (Chiu *et al.*, 1983; Soroush & Kravaris, 1993; Wright & Kravaris, 1997). This subchapter presents the first-principle model of the polymerization process. Figure III-2 shows the Process Flow Diagram (PFD) of the process, which was adopted based on (Soroush & Kravaris, 1993; Wright & Kravaris, 1997).

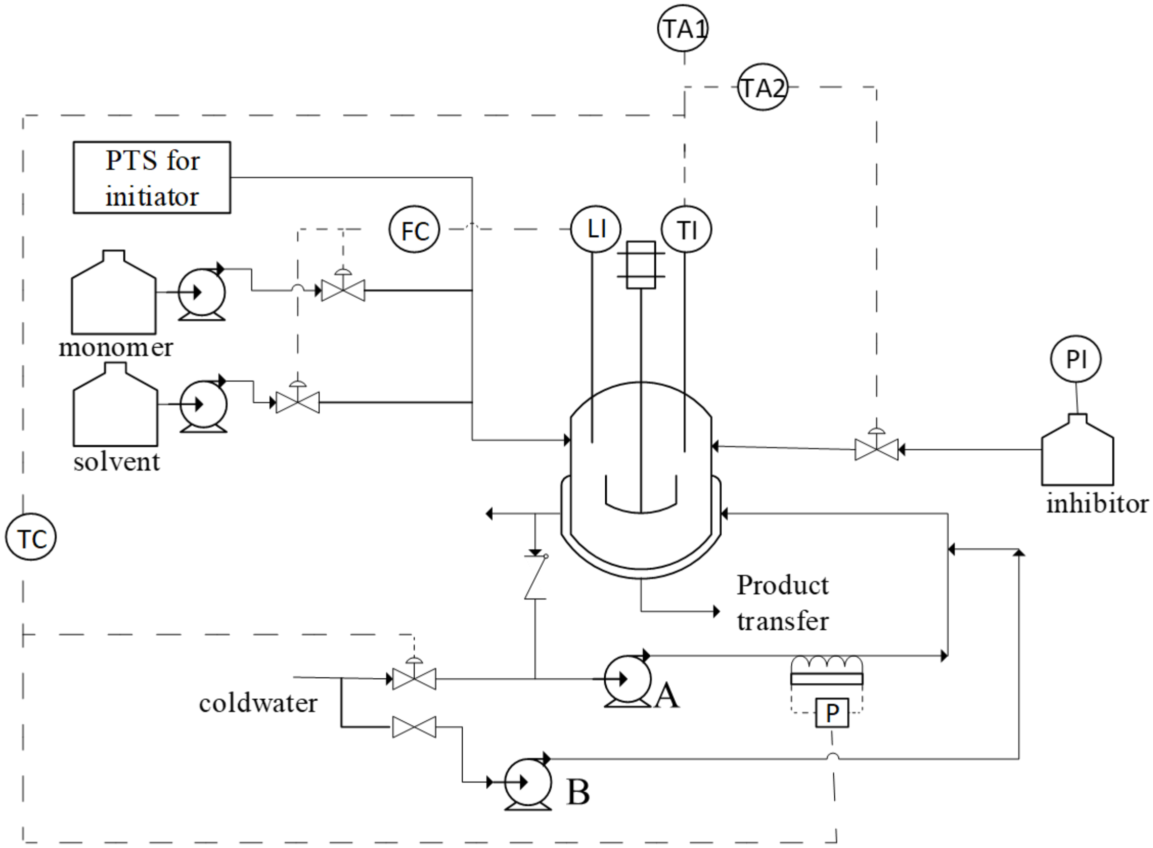


Figure III-2 Process Flow Diagram of a hypothetical PMMA batch process

PMMA is produced through the solution polymerization with methyl methacrylate (MMA) as the monomer, toluene as the solvent, and azobis(isobutyronitrile) (AIBN) as the initiator. Since the polymerization mechanism is highly complex and is detailed out in (Soroush & Kravaris, 1993), only mass balances of the process are provided as follows.

$$\frac{\mathrm{dC_m}}{dt} = \left(1 + \varepsilon * \frac{\mathrm{C_m}}{\mathrm{C_{m0}}}\right) * \left[-\mathrm{C_m} * \xi_0 * \left(\mathrm{k_p} + \mathrm{k_{cm}}\right)\right] \qquad \text{(Equation III-10)}$$

$$\frac{\mathrm{dC_i}}{dt} = -\mathrm{k_i C_i} + \varepsilon * \frac{\mathrm{C_m}}{\mathrm{C_{m0}}} * \left[-\mathrm{C_m} * \xi_0 * \left(\mathrm{k_p} + \mathrm{k_{cm}}\right)\right] \qquad \text{(Equation III-11)}$$

where $\mathrm{C_m}$ is monomer concentration, $\mathrm{C_{m0}}$ is initial monomer concentration, ε is volume change factor, k_p is rate constant of propagation, k_i is rate constant of imitation, k_{cm} is the rate constant of chain transfer to monomer, and ξ_0 is total concentration of live polymer chains.

The temperature of the reactant (T_r) is controlled by a Proportional Integral (PI) controller with the back-calculation anti-windup technique (see Figure III-3).

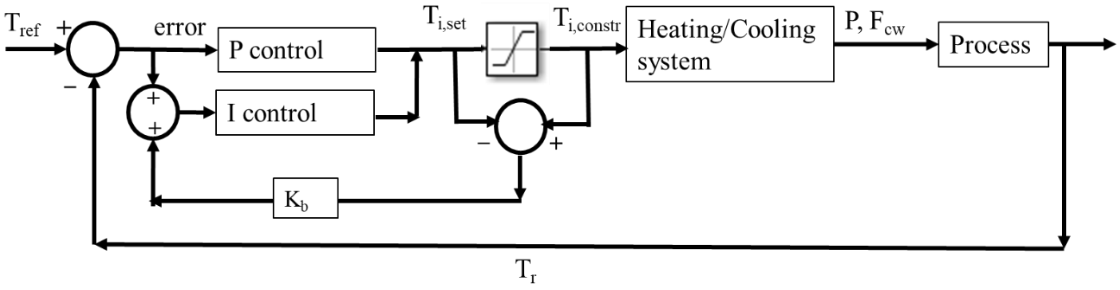


Figure III-3 Heating/Cooling Control System

The controller is with gain k_c =28.4, time constant τ_I = 268, and back-calculation gain k_b = 0.04 (Bohn & Atherton, 1995; Wright & Kravaris, 1997). The controller measures T_r and outputs the corresponding jacket temperature that is needed ($T_{i,set}$) based on the deviations from the reference reactant temperature profile (T_{ref}). In the case that the $T_{i,set}$ exceeds the feasible temperature range of the utility water, $T_{i,const}$ is the final reference temperature to take the feasible range into account. $T_{i,const}$ is used to adjust the cooling water flow rate (F_{cw}) and the power of the heater (P) to ensure the reactor is operated following the predetermined temperature profile. The optimal temperature profile of the 8-hour batch process in Wright and Kravaris (1997) was used as the T_{ref}. Energy balances of the reactor are shown as Equation III-12 and III-13 (Soroush & Kravaris, 1992).

$$\frac{dTr}{dt} = \frac{-\Delta H_p k_p \xi_0 C_m V}{c*m} + \frac{U_0(a+(1-a)*\exp{(-b{x_m}^c)}*A_0(1+\varepsilon x_m)}{c*m}(T_j - Tr) \qquad \text{(Equation III-12)}$$

$$\frac{dT_j}{dt} = \frac{U_0(a+(1-a)*\exp{(-b{x_m}^c)}*A_0(1+\varepsilon x_m)*(Tr-T_j)}{V_j \rho_j C_{pj}} + \frac{P-F_{cw}(T_j-T_{cw})\rho_j C_{pj}}{V_j \rho_j C_{pj}} \qquad \text{(Equation III-13)}$$

where T_j is the temperature of reactor jacket, ΔH_p is the heat of propagation reaction, V is volume of the reactor, c is the heat capacity of the reactant, m is the mass of the reactant, U_0 is the initial overall heat transfer coefficient, x_m is monomer conversion rate, A_0 is the initial heat transfer area, V_j is the volume of the reactor jacket, ρ_j is the density of water, C_{pj} is the heat capacity of water, T_{cw} is the temperature of the utility cooling water, and a, b, c are empirical coefficients to calculate the overall heat transfer coefficient. According to the design of the heating/cooling control system, the energy to be generated/removed is

$$u = C_{pj} * \rho_j * F_{cw\_max} * (T_{i,const} - T_j) \qquad \text{(Equation III-14)}$$

and it can be applied to the coordination rules in (Soroush & Kravaris, 1992) to set the F_{cw} and P of the Heating/Cooling control system. The formulas and values to derive the kinetic parameters are available in (Soroush & Kravaris, 1993). Other physical parameters that are involved in the process are provided in Table III-4.

Table III-4 Values of Physical Parameters for the Polymerization Process

Parameter	Value	Parameter	Value
Volume of the reactor, V	37.85 m^3	Initial heat transfer area, A_0	52.95 m^2
Volume of the reactor jacket, V_j	3.405 m^3	Heat capacity of water, C_{pj}	4.19 kJ/kg/K
Density of water, ρ_j	10^3 kg/m^3	Empirical coefficient, a	0.2
Initial overall heat transfer coefficient, U_0	0.4543 kJ/Km^2s	Empirical coefficient, b	7
		Empirical coefficient, c	3

3.6 Hybrid Simulator

Interactions among functions can be described qualitatively by the FRAM, but to understand the interactions quantitatively, a simulator needs to be developed to integrate the models that were mentioned above. The process involves discrete events such as operator actions and equipment failures, as well as continuous events such as the reaction mechanism. The hybrid simulator was built by using Simulink and Stateflow, two modules in MATLAB. They can model continuous and discrete processes accordingly and be integrated in the uniform MATLAB environment (Bonabeau, 2002; Papakonstantinou *et al.*, 2011, 2012; Pascal & Sahbani, 2000; Zhang *et al.*, 2013; Zhang *et al.*, 2007).

Additionally, the process involves both stochastic and deterministic events. Human performance and equipment performance are stochastic depending on the probabilistic human and equipment performance models, while the reaction itself is a deterministic evolution of the process parameters (Zhang *et al.*, 2013). By simulating the process along the operating time, thousands of batches can be produced in years. Human behaviors can be sampled during each batch, leading to a large sample size. However, the sample size of the equipment failures is relatively small since equipment failures occur rarely based on the expected number of failures. Given different sample sizes, random sampling was used to sample human behaviors, while Latin Hypercube Sampling (LHS) was used to sample equipment failures. Random sampling can reflect the real distribution of the sampled variable when the sample size is large, but it is not when the sample size is small. Instead, LHS is a constrained sampling method following the stratified manner, thus it ensures the samples are generated from different portions under the distribution (McKay *et al.*, 1979). Equipment failures can be sampled by sampling the timestamps when equipment failures occur based on the cumulative failure distribution. The sample size is the same as the expected number of failures N. The cumulative failure probability for n_{th} failure P_n can be obtained by Equation III-15 (Wyss & Jorgensen, 1998) and the corresponding timestamp of n_{th} equipment failure can be obtained by taking the inverse of the sampled cumulative failure probability.

$$\mathrm{P_n} = \frac{1}{\mathrm{N}} * \mathrm{U_n} + \frac{\mathrm{n-1}}{\mathrm{N}} \qquad \text{(Equation III-15)}$$

where $n = 1,2,3 \ldots N,$ U_n is the n_{th} sample that is randomly sampled from the uniform distribution in [0,1].

3.7 Case Study: Batch Polymerization Process

The subchapter provides the case study of the hypothetical PMMA batch process to illustrate how models can be integrated into a hybrid simulator for modeling function interactions.

The process in Figure III-2 is operated by one field operator (FO) and one control room operator (CRO) in each batch. The operating procedure is below:

1. The FO loads 742-kg initiator to the powder transfer system (PTS)
2. The control room operator enters the initial monomer concentration, which is 3.66 kgmol/m^3 on the control panel. The batch size V_0 = 37.09 m^3 is not allowed to be customized by the operator thus the corresponding initial solvent concentration is automatically calculated by the program.
3. The CRO authorizes the automatic solvent charging process.
4. The solvent is added to the reactor by the automatic solvent charging process.
5. The CRO starts the agitator.
6. The CRO authorizes the automatic monomer charging process.
7. The monomer is added to the reactor by the automatic monomer charging process.
8. After the monomer is added to the reactor, the CRO checks the remote level display. If the amount of the reactant that is added into the reactor is not the same as the amount that the recipe requires, the CRO adjusts the reactant amount in the reactor.
9. The CRO circulates the utility cooling water to the reactor jacket.
10. Heating/Cooling system controls the reactant temperature based on the preset reactor temperature profile.
11. The CRO monitors the reactant temperature and remotely adds the initiator from the PTS to the reactor at the optimal initial temperature, 295K.
12. After the batch runs for 8 hours, the reaction mass is transferred for further treatment.

3.7.1 Functions and Performance Variabilities

FRAM was applied first to understand the interactions among the functions qualitatively. Since the full-scale FRAM of the batch process could involve many functions, only essential functions to illustrate the framework are discussed in this study. The FRAM of the system is shown in Figure III-4, with organizational functions colored in blue, human functions colored in purple, and technological functions colored in green. The FRAM was constructed using the FRAM Model Visualizer software (Hill & Hollnagel, 2016). The function descriptions and their performance variabilities are summarized in Tables III-5, III-6, and III-7 by function categories. Instead of describing performance variabilities in the FRAM solely using linguistic levels, the performance variabilities of functions were described quantitatively if they were applicable. In the original FRAM, performance variabilities can be described in terms of precision and time. As a case study for demonstration, performance variabilities in this study were only described in terms of precision, and performances of some functions were assumed to be invariant. In the tables, the functions with invariant performance are labeled with *.

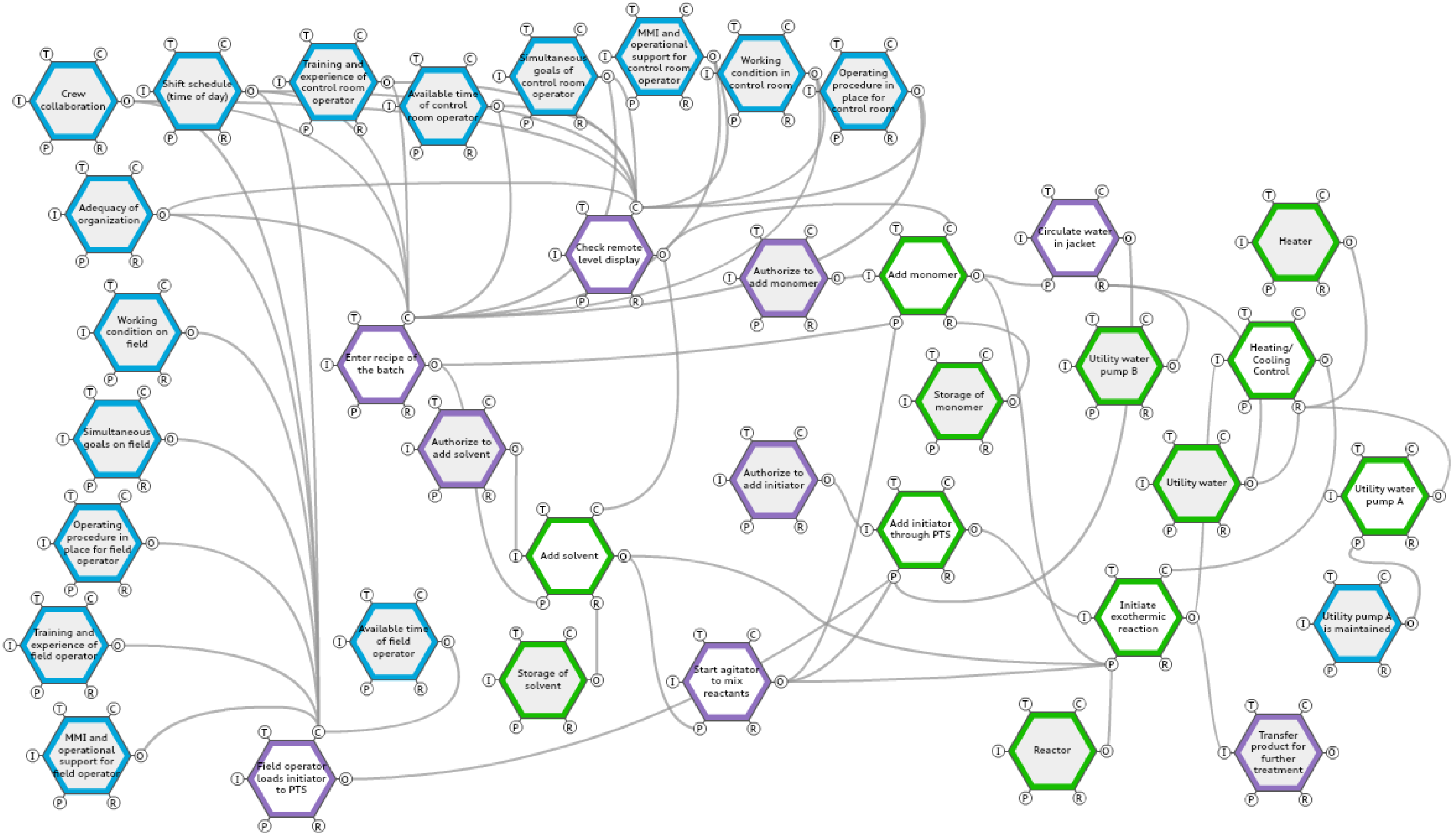

Crew collaboration
Shift schedule (time of day)
Training and experience of control room operator
Available time of control room operator
Simultaneous goals of control room operator
MMI and operational support for control room operator
Working condition in control room
Operating procedure in place for control room
Adequacy of organization
Check remote level display
Authorize to add monomer
Add monomer
Circulate water in jacket
Heater
Working condition on field
Enter recipe of the batch
Utility water pump B
Heating/ Cooling Control
Storage of monomer
Simultaneous goals on field
Authorize to add solvent
Authorize to add initiator
Utility water
Utility water pump A
Operating procedure in place for field operator
Add solvent
Add initiator through PTS
Initiate exothermic reaction
Utility pump A is maintained
Training and experience of field operator
Available time of field operator
Storage of solvent
Start agitator to mix reactants
Reactor
Transfer product for further treatment
MMI and operational support for field operator
Field operator loads initiator to PTS
T
C
I
O
P
R

Figure III-4 FRAM analysis of MMA polymerization process. Organization functions are in blue. Human functions are in purple. Technological functions are in green. Shaded functions in grey represent the boundary of the system.

Organization functions in FRAM and their performance variabilities are summarized in Table III-5. The functions are CPCs in CREAM and their performance variabilities are consistent with levels of CPCs in CREAM. The performance variabilities of the functions “Adequacy of organization”, and “Operating procedure in place for control room operator/field operator” were neglected and assumed to be at constant levels since they could vary infrequently compared to other CPCs. The variabilities of “Crew collaboration” and “MMI and operational support for control room operator/field operator” were neglected for the purpose of the simplification since the impacts of their performance on their downstream human functions would require detailed analysis of information processing.

Table III-5 Descriptions of organization functions and performance variabilities

Organization Function	Description	Performance Variabilities
Adequacy of organization*	Quality of supports and resources provided by the organization	Efficient
Available time of field operator	Time pressure for field operator	Adequate, Temporarily inadequate, Continuously inadequate
Available time of control room operator	Time pressure for control room operator	Adequate, Temporarily inadequate, Continuously inadequate
Crew Collaboration*	Quality of collaboration between crew members	Efficient
MMI and operational support for control room operator*	Quality of man-machine interface and supports provided for control room operator	Adequate
MMI and operational support for field operator*	Quality of man-machine interface and supports provided for field operator	Adequate
Operating procedure in place for control room*	Availability of operating procedure and guidance for control room operator's tasks	Acceptable
Operating procedure in place for field operator*	Availability of operating procedure and guidance for field operator's tasks	Acceptable
Shift schedule (time of day)	Shift time when the task is carried out	Day-time, Night-time
Simultaneous goals of control room operator	Number of tasks that control room operator must attend to	Fewer than capacity, Matching current capacity, More than capacity
Simultaneous goals on field	Number of tasks that field operator must attend to	Fewer than capacity, Matching current capacity, More than capacity
Training and experience of control room operator	Training and experience levels of control room operator	Adequate training and high experience, Adequate training and low experience, Inadequate training

*Represents functions whose performance are assumed to be invariant

Table III-5 Continued

Organization Function	Description	Performance Variabilities
Training and experience of field operator	Training and experience levels of field operator	Adequate training and high experience, Adequate training and low experience, Inadequate training
Working condition in control room	Working conditions in control room such as noise and lighting	Advantageous, Compatible, Incompatible
Working condition on field	Working conditions on field such as noise and lighting	Advantageous, Compatible, Incompatible
Utility pump A is maintained	Preventive maintenance (PM) of utility pump A	"PMs being carried out as optimal PM schedule", "no PM being carried out"

*Represents functions whose performance are assumed to be invariant

Human functions and their performance variabilities are summarized in Table III-6. The human functions that were assumed to be carried out as the procedure required in this study did not have variable performance. The performance variabilities of the rest of human functions were identified based on the generic failure types in CREAM. According to the procedure, the control room operator only needed to enter the initial monomer concentration on the control panel manually. Typing error could occur if the operator pressed the keys out of sequence. The control panel was designed to allow 3-digit inputs and only allowed the inputs to vary in the range of [3.20, 4.11] for safety reasons. When the control operator checked the remote level display after raw materials were added into the reactor, if the typing error did occur, the likely failures could be faulty diagnosis or not making an observation, which made the process start with the wrong amount of raw materials. Otherwise, the wrong amount could be adjusted before the process started. If no typing error occurred, it was likely the control room operator did not verify the remote level display, but it would not cause any hazardous situation fortunately. On the other hand, the field operator could load a wrong amount of the initiator, and based on the expert judgements, the possible range of the loading could be in the range of 742 (± 5%) kg. It was likely the field operator forgot to load the initiator before a batch. Since the polymerization process could not occur if no initiator was loaded and could not lead to any hazardous situation, the scenario of missed action was neglected.

Table III-6 Descriptions of human functions and performance variabilities

Human Function	Description	Performance Variabilities
Enter recipe of the batch	Control room operator enters three-digit monomer concentration on the control panel	Monomer concentration is range from 3.20 to 4.11 kgmol /m^3
Authorize to add solvent*	Control room operator authorizes to add solvent through the automatic feeding system	Carried out as the procedure requires
Check remote level display	Control room operator checks remote level displays to ensure the amount of solvent and monomer added is the same as the procedure requires	Faulty diagnosis, Observation not made, No failure
Authorize to add monomer*	Control room operator authorizes to add monomer through the automatic feeding system	Carried out as the procedure requires
Authorize to add initiator*	Control room operator authorizes to add initiator through the automatic feeding system	Carried out as the procedure requires
Field operator loads initiator to PTS	Field operator loads the initiator to PTS	The mass of the initiator is range from 705 to 780 kg
Start agitator to mix reactants*	Control room operator starts the agitator remotely	Carried out as the procedure requires
Circulate water in jacket*	Control room operator circulates water into the reactor jacket remotely	Carried out as the procedure requires
Transfer product for further treatment*	Field operator transfers products for further treatment	Carried out as the procedure requires

* Represents functions whose performance are assumed to be invariant

Regarding technological functions, their performance variabilities did not require identifications by expert judgements since they are deterministic based on first principles, the controller design, and capabilities of equipment. The technological functions and their performance variabilities are summarized in Table III-7. The performance variabilities of the functions with * were neglected in the current scope of the study, which means they were assumed to be functional and carried out as how they were designed.

Table III-7 Descriptions of technological functions and performance variabilities

Technological Function	Description	Performance Variabilities
Initiate exothermic reaction	Reaction is initiated to synthesis polymer product	Reactant temperature is decided by mass and energy balances
Utility water pump A	Pump utility water to the reactor jacket to cool down reactant temperature when necessary	Pump is functional, Pump breaks down
Heating/Cooling control	Heating/cooling control system to control reactant temperature	Performance variabilities are decided by the design of the control system
Heater*	Provide heat to the reactor when necessary	Heater is functional
Reactor	Condition of the reactor	The reactor is under clean condition, The reactor has fouling which reduces overall heat transfer coefficient by 20%
Storage of solvent*	Store enough solvent at an appropriate temperature to be fed into the reactor	Enough solvent is stored at the optimal temperature to start a batch reaction
Storage of monomer*	Store enough monomer at an appropriate temperature to be fed into the reactor	Enough monomer is stored at the optimal temperature to start a batch reaction
Add initiator through PTS	PTS releases initiator to the reactor	PTS is functional to add initiator that is loaded by field operator to the reactor (i.e. mass of initiator added ranges from 705 to 780 kg)

*Represents functions whose performance are assumed to be invariant

Table III-7 Continued

Technological Function	**Description**	**Performance Variabilities**
Add solvent	Add solvent to the reactor by the automatic feeding system	The feeding system is functional to add solvent into reactor based on the recipe that entered (i.e. solvent concentration ranges from 5.03 to 5.94 kgmol/m^3)
Add monomer	Add monomer to the reactor by the automatic feeding system	The feeding system is functional to add monomer into reactor based on the recipe that entered (i.e. Monomer concentration ranges from 3.20 to 4.11 kgmol/m^3)
Utility water pump B*	Pump utility water to reactor jacket before a reaction starts	Pump B is functional
Utility water*	Availability of utility water	Utility water is available at a constant temperature

*Represents functions whose performance are assumed to be invariant

3.7.2 Implementation of Hybrid Simulator

After the possible performance variabilities were identified, the hybrid simulator was built to model the interactions among the functions and understand the corresponding aggregated effects when functions have various performances. The schematic diagram of the simulator is shown in Figure III-5. The simulator was divided into five sections:

Section A was to simulate the performance of the human functions “Enter recipe of the batch”, and “Check remote level display” that were carried out by the control room operator. The section started from the organizational CPC levels that would impact the control room operator’s behavior and output the initial monomer concentration (C_{m0}) to start the process.

Section B was to simulate the performance of the human function “Field operator loads the initiator to the PTS”. The section started from the CPC levels that would influence the field operator’s performance and output the mass of the initiator (m_{i0}) to start the process.

Section C was to simulate the polymerization process based on the first-principles given C_{m0} and m_{i0} from section A and B, and the performance of utility pump A, which was simulated in section D.

Section D was to simulate the performance of the utility pump A, given the performance of the function “Utility pump A is maintained”, which could be “PMs being carried out as optimal PM schedule” and “no PM being carried out”.

To improve the computational efficiency of data generation, Section E was to disable Section C for skipping the simulations of the 8-hr process under the most common conditions, i.e., when the initial reactant composition was the same as the procedure requires and utility pump failure did not occur within a batch.

Regarding the time sequence of simulating the sections in a batch, section A and B was executed and provided their outputs at the simulation time = 0s, then the outputs were held as constants until the simulation of the polymerization process was completed. Section E took the outputs from sections A and B as its inputs, and was executed at simulation time = 1s. Then, the output condition from section E determined whether section C was to be executed. When the execution of section C was enabled based on the output conditions of section E, the execution started from the simulation time at 2s and lasts for 28800s (8 hours), i.e., ends at 28802s. Section D was executed at t = 0 to provide information of equipment performance to section E then ran simultaneously with section C.

Since the Simulink and Stateflow modules only simulate the process batch by batch with assigned parameters, besides the sections in the schematic diagram, there was a separate MATLAB script which pseudo-randomly assigned the values for CPC levels to initiate the process, sampled the timestamps of pump A failure, simulated the process with a 5-year operating time and output interested variables. The MATLAB script is provided in Appendix A.

The detailed explanations of the simulator are provided as follows and codes of Matlab functions for all the sections are provided in Appendix B.

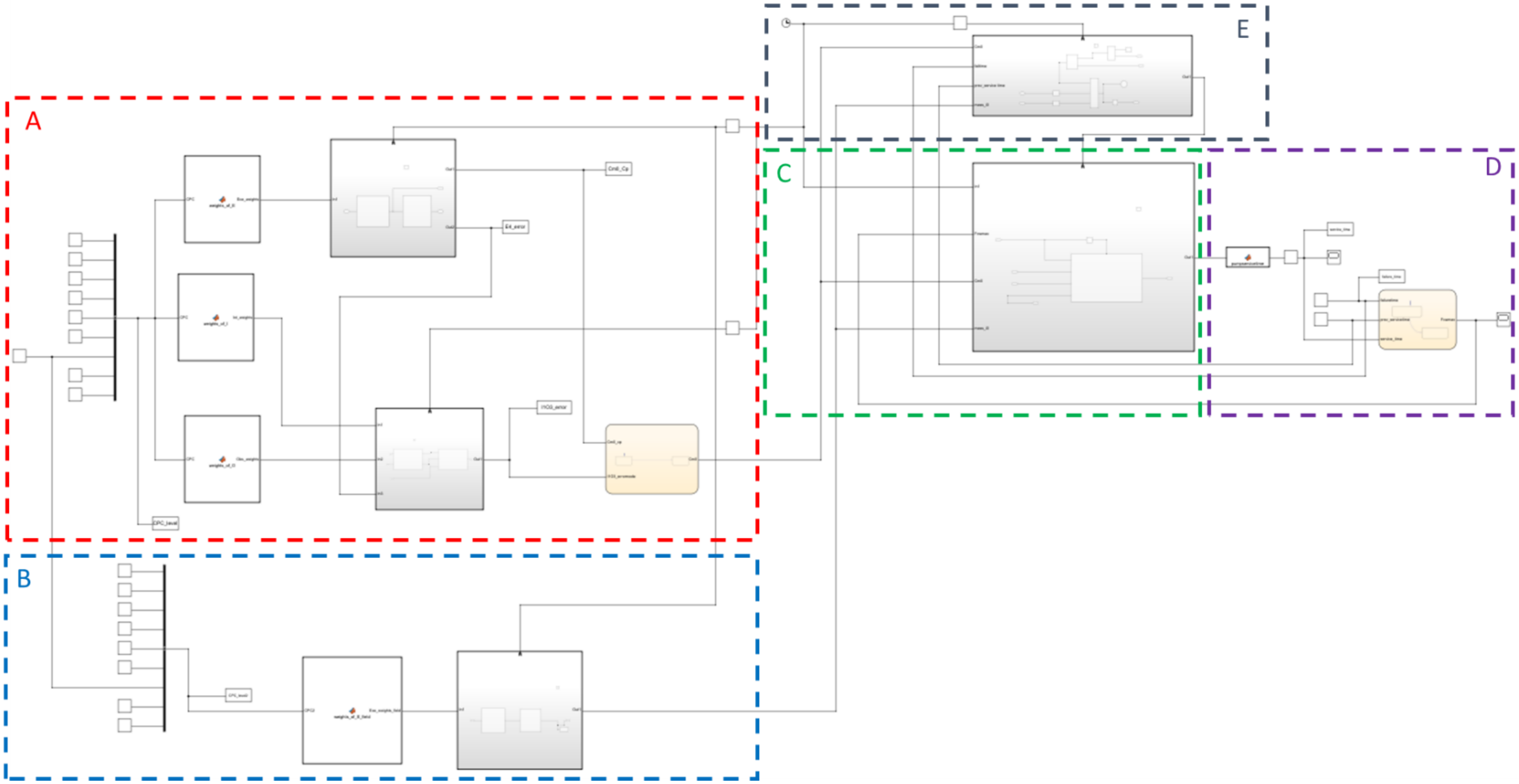


Figure III-5 Schematic diagram of the hybrid simulator

In section A, the simulation started from the CPC levels that would impact the behaviors of the control room operator. According to the performance variabilities that were identified above, the CPC levels were coded to ordinal variables as shown in Table III-3 and randomly sampled from uniform distributions for each batch to represent the performance variabilities of the organization functions. Once the CPC levels were randomly sampled, they were sent to the "Weight Calculation" block to calculate the weights for updating the nominal probabilities of feasible errors. According to "Generic Failure Occurrence" in Table III-2, three types of feasible errors were execution, interpretation, and observation. Within the block "Enter recipe of the batch", the nominal probability of the execution error was updated by the execution weights, and the occurrence of E4 was randomly sampled based on the updated probability. Based on the occurrence of E4, a simulated initial monomer concentration that was entered on the control panel (C_{m0}_cp) was generated according to the performance variabilities that were identified. The simulated occurrence of E4 was then taken as an input to simulate the performance variabilities of the function "Check remote level display". Depending on whether E4 occurred, the error modes of the failure occurrence were simulated by sampling the modified probabilities of interpretation and observation errors. The possible failure occurrences in Table III-2 were coded as shown in Table III-8.

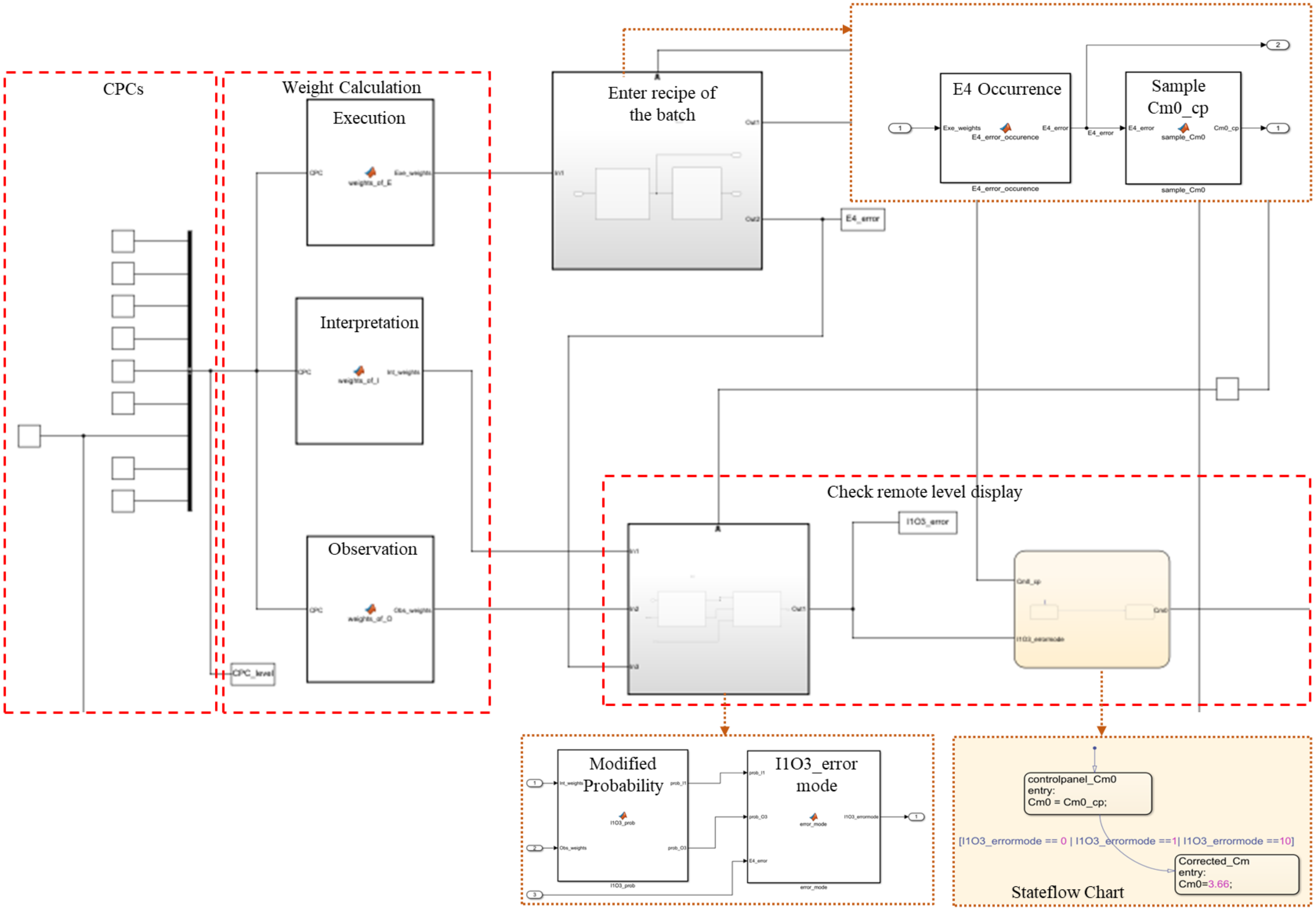


Figure III-6 Detailed schematic of section A

Table III-8 Codes of failure occurrences for the human function "Check remote level display"

Human Function	Failure Occurrence	Code of Failure Occurrence (I1O3_errormode)
Check remote level display	If typing error occurs	
	Faulty diagnosis (I1)	I1O3_errormode = 11
	Observation not made (O3)	I1O3_errormode = 13
	No failure	I1O3_errormode = 10
	If no typing error occurs	
	Observation not made (O3)	I1O3_errormode = 1
	No failure	I1O3_errormode = 0

The monomer with the initial monomer concentration to start the batch (C_{m0}) was added to the reactor by the function "Add monomer", which was impacted by its upstream functions "Enter recipe of the batch" and "Check remote level display". Thus, the upstream-downstream coupling was simulated by the Stateflow chart in Figure III-6, which had C_{m0}_cp and Interpretation/Observation failure occurrence (I1O3_errormode) as inputs and C_{m0} as the output. Given the C_{m0}_cp, if the I1O3_errormode=0, 1 or 10, the batch would start with C_{m0} as 3.66 kgmol/m^3. Otherwise, the batch would start with the C_{m0} that was originally entered on the control panel. The output of the Stateflow chart, $C_{m0,}$ was sent to section E as a variable to determine whether Section C was enabled and was also sent section C as an input variable of the batch process.

Similarly, as shown in Figure III-7, section B started from the randomly sampled CPC levels that would impact the performance of the field operator. Since both the field operator and the control room operator worked in the same organization and during the same shift, the simulated CPC levels of "Crew collaboration", "Time of day" and

“Adequacy of organization” for both operators were the same. The CPC levels were used to sample the failure occurrences based on Table III-2. Finally, the corresponding m_{i0} was simulated as the output of section B. The output was sent to section E as a variable to determine whether Section C was enabled and was sent to section C as an input variable of the batch process.

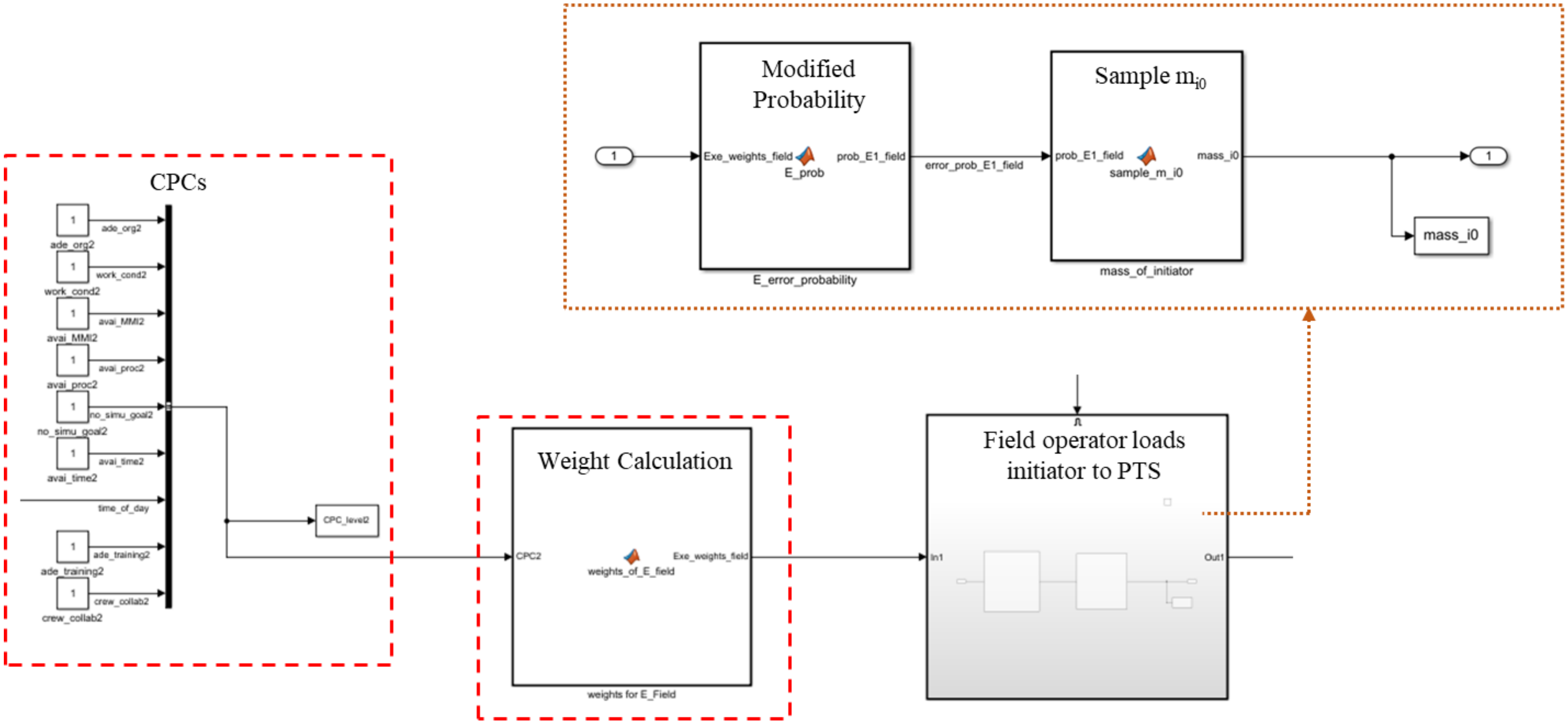


Figure III-7 Detailed schematic of section B

Section C simulated the temperature profile of a batch process based on the first-principle model. This section was built as an embedded system containing two subsystems. Figure III-8 shows the outer subsystem and Figure III-9 shows the inner subsystem. The main purpose of the outer system was to provide the time constraint through the input port ‘clock_time’ and ensured section C was executed at the simulation time = 2s when it was enabled by section E, as explained earlier. Additionally, the outer subsystem took outputs from other sections and transferred them to the inner subsystem to simulate the batch process.

As shown in Figure III-8, the input of the section required the initial concentration of monomer (C_{m0}) and the initial mass of initiator (m_{i0}), which were outputs from section A and B accordingly and represented by the input ports ‘Cm0’, and ‘mass_i0’ in the figure. Additionally, the reactant temperature of the batch process depended on the performance of the utility pump A, whose maximum utility water flow rate (F_{cw_max}) was the output of Section D and was represented by the input port ‘Fcwmax’. These inputs were sent to the inner system that was shown in Figure III-9 and became the process variables of the polymerization process. The inner system was constructed based on the first-principle model in subchapter 3.4.

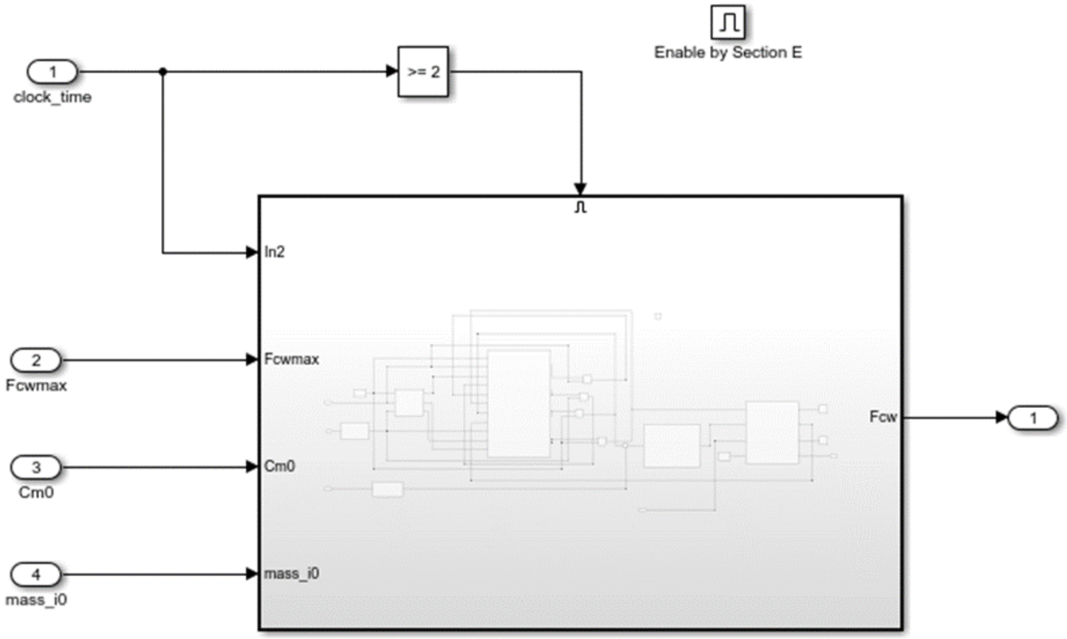


Figure III-8 Detailed schematic of the outer subsystem in Section C

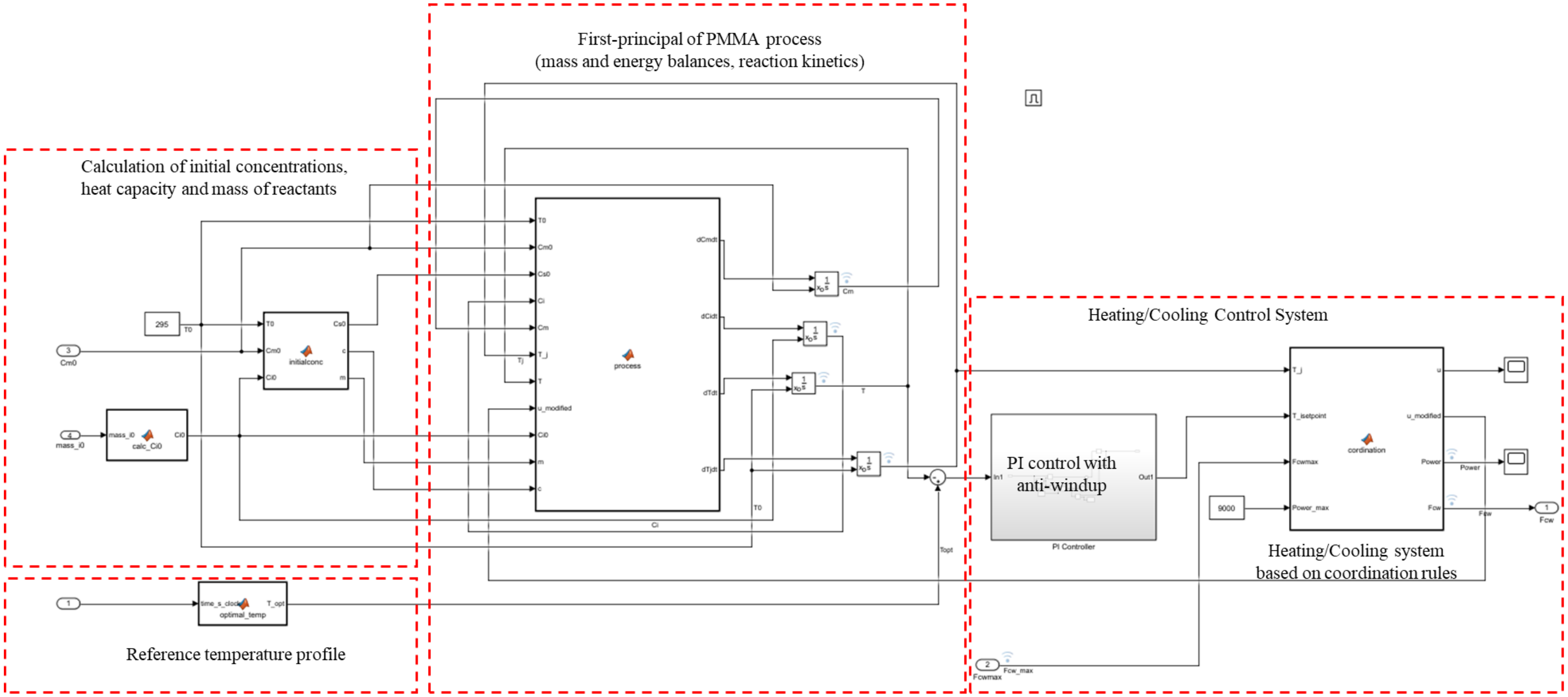


Figure III-9 Detailed schematic of the inner subsystem in Section C

Figure III-10 shows the detailed schematic of section D. The performance of the utility pump depended on PM plans. Given a pre-determined PM plan, the timestamps of pump failures were simulated by LHS. In the study, the lifetime of the pump was assumed to be 43800 hours (5 years), with $\beta = 1.2$, the characteristic life = 10000 hours, and the corresponding $\alpha = 1.58*10^{-5}$. The optimal PM plan was scheduled based on $r_c = 0.7$ and $\xi = 0.7$. Therefore, to study the impact of PMs on the pump performance, the batch process was simulated with 5-year scale for both case 1 “PMs being carried out as the optimal PM schedule”, and case 2 “PM not being carried out”. In the section, the simulated timestamps of pump failures for each case were fed into the block “failuretime”. The block was programmed to be updated as the earliest timestamp of the pump failure which was later than the service time of the previous batch (block “prev_servicetime”). During the first batch in a 5-year scale, the earliest failure time was first fed into the block “failuretime” and the block “prev_servicetime” was initiated from 0. As the batch ran, the service time of the pump was accumulated by integrating the time when the utility water was pumped to control the reactant temperature. When the first batch was completed, the total service time of the first batch would be set as the value of block “prev_servicetime” for the next batch. By comparing the timestamps of the pump failures and the service times, the pump performances were simulated in the Stateflow chart. The pump was capable of pumping water at $F_{cw\_max} = 0.04413$ m^3/s when it worked. When the total service time of the pump equaled a timestamp of the pump failure, the pump broke down and the F_{cw_max} became 0 m^3/s. With the simulation of the pump performance, F_{cw_max} was sent to section C as an input variable of the process simulation. Besides, before each batch process started, “failuretime” and “prev_servicetime” were sent to Section E to determine whether a pump failure would occur during the batch.

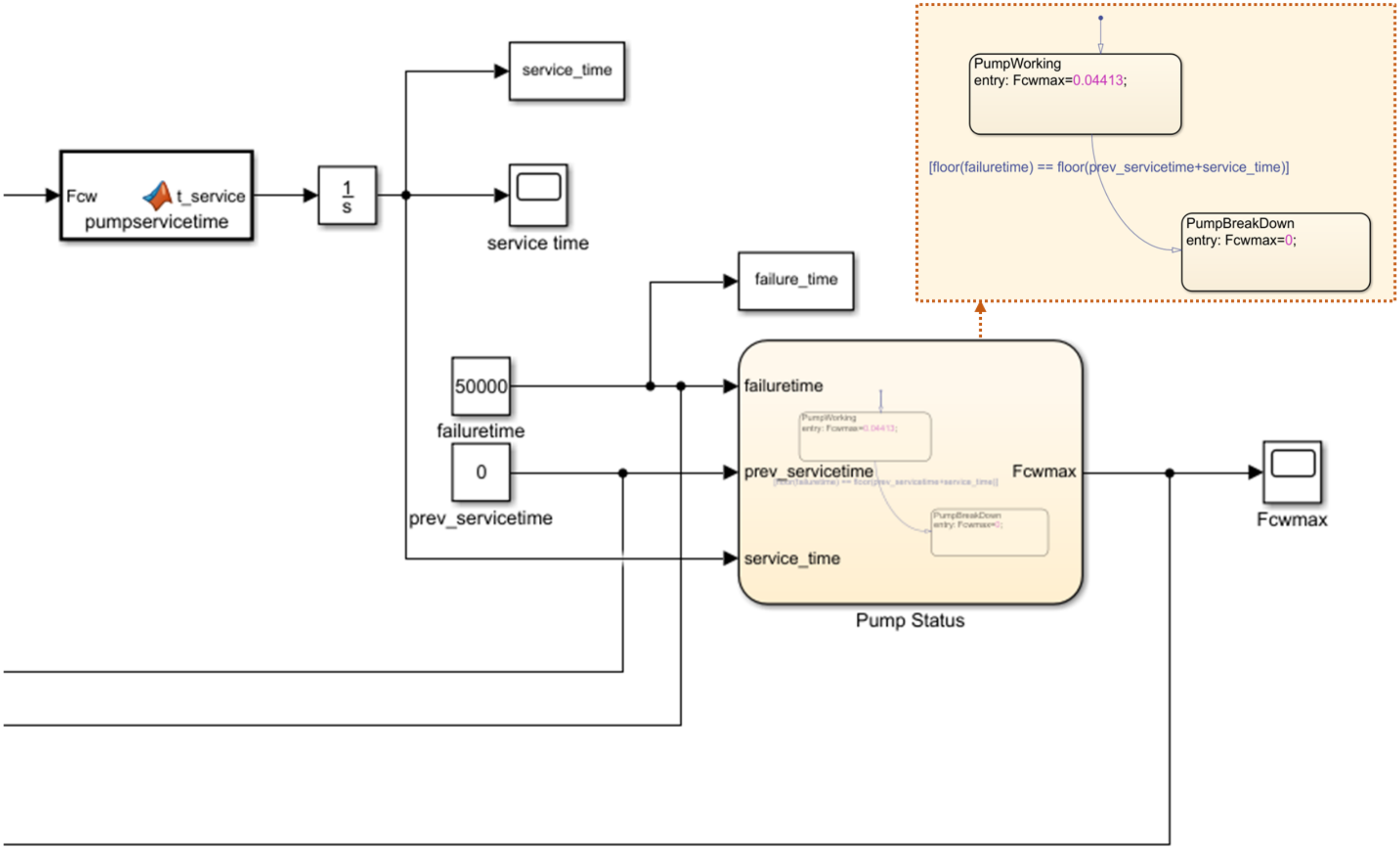


Figure III-10 Detailed schematic of Section D

Figure III-11 shows the detailed schematic of Section E. As mentioned, Section E was to skip the process simulation in Section C under the most common conditions, i.e., when the initial reactant composition was the same as the procedure required and utility pump failure did not occur within a batch. As shown in Figure III-11, when

- duration of the batch system (‘batch_duration”) + the service time of the previous batch (“prev_servicetime”) < the simulated timestamp of pump failure AND,
- initial concentration of the monomer was the same as the recipe required, i.e., Cm0 = 3.66 AND,
- initial mass of the initiator was the same as the recipe required, i.e., mass_i0 =742,

the simulation of the batch would be terminated by the block “STOP”. Otherwise, when the condition was false, section C was enabled and the polymerization process was simulated. The system was simulated by pre-defining the performance of “Reactor” (i.e., pre-defining the overall heat transfer coefficient in the first principle model) so that simulation could be skipped under two conditions: 1) initial reactant composition was same as the procedure required, no utility pump failure, and the reactor was under a clean condition and 2) initial reactant composition was same as the procedure required, no utility pump failure, and the reactor had fouling that reduced the overall heat transfer coefficient by 20%. Datasets for the two conditions had been generated in advance so they could be loaded into the output data when the simulations of the batches were terminated by section E.

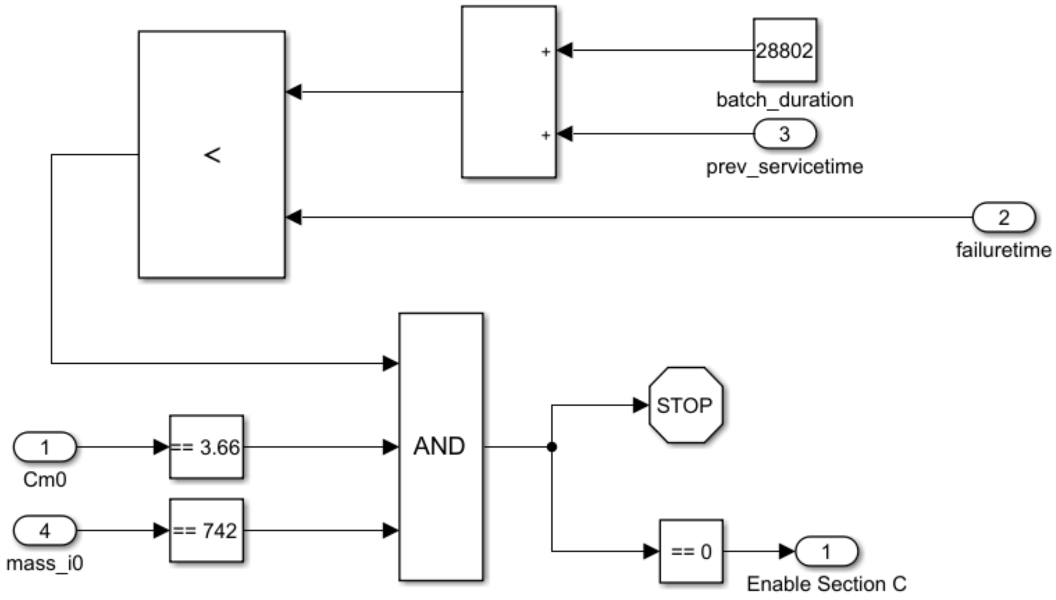


Figure III-11 Detailed schematic of Section E

3.7.3 Synthetic Data Generation

As mentioned previously, since the hybrid simulator only simulated the process batch by batch with assigned parameters, the separate MATLAB script which was created to initialize the hybrid simulator was also used to export the output data. The parameters to be assigned in the script corresponded to the performances of the background functions in the FRAM in Figure III-4, which only have output aspects. However, performances of the background functions could vary with different time scales. For example, the performance of the function "Utility pump A is maintained", representing PM schedule of the pump, depended on the pump's operating time in years, while performances of the functions such as "Training and experience of control room operator" depended on shift personnel thus could vary every batch. In order to simulate interactions between the functions which impacted the system with various time scales, the system was simulated in a time-series manner up to the longest time scale. In the study, the system was simulated with a 5-year time span, which was the utility pump's operating life, assuming there were 3650 batches in every 5-year time span.

Additionally, performances of the function "Utility pump A is maintained" and the function "Reactor" needed to be pre-defined for a 5-year simulation in the separate MATLAB script and Section C in the hybrid simulator accordingly. Thus, the simulations were conducted under four cases and 219,000 batch operations were simulated for each case, which was 876,000 batch operations in total. The simulation matrix is provided in Table III-12. The simulations were conducted by using clusters of High-Performance Research Computing (HPRC) at Texas A&M University, College Station, TX.

Case #	**Performance of the function "Utility pump A is maintained"**	**Performance of the function "Reactor"**	**# of batches**
1	"PMs being carried out as optimal PM schedule"	The reactor is under clean condition	219,000
2	"PMs being carried out as optimal PM schedule"	The reactor has fouling which reduces overall heat transfer coefficient by 20%	219,000
3	"no PM being carried out"	The reactor is under clean condition	219,000
4	"no PM being carried out"	The reactor has fouling which reduces overall heat transfer coefficient by 20%	219,000

Figure III-12 Simulation matrix

The output data of the simulation contained the variables that represented function performances. As shown in Tables III-5, 6 and 7, some functions were considered to have invariant performance as the system was designed. These functions did not impact performance variabilities of its downstream function and did not impact the aggregated couplings on the downstream function. Figure III-13 shows the simplified FRAM by removing these functions. Thus, the variables which represented the performances of the functions in the simplified FRAM were exported in the output data. Table III-9 shows output variables that were exported from the simulations, as well as variable types and descriptions corresponding to the functions in the simplified FRAM. Performances of the functions “Initiate exothermic reaction”, “Heating/Cooling Control”, and “Utility water pump A” varied after the polymerization process started, thus performances of these functions were exported as time-series data of an 8-hr batch operation. On the other hand, performances of the rest of the functions only varied batch by batch, thus they were exported as single values for a batch. The output data will be further processed and used as data sources of the work in Chapters IV and V.

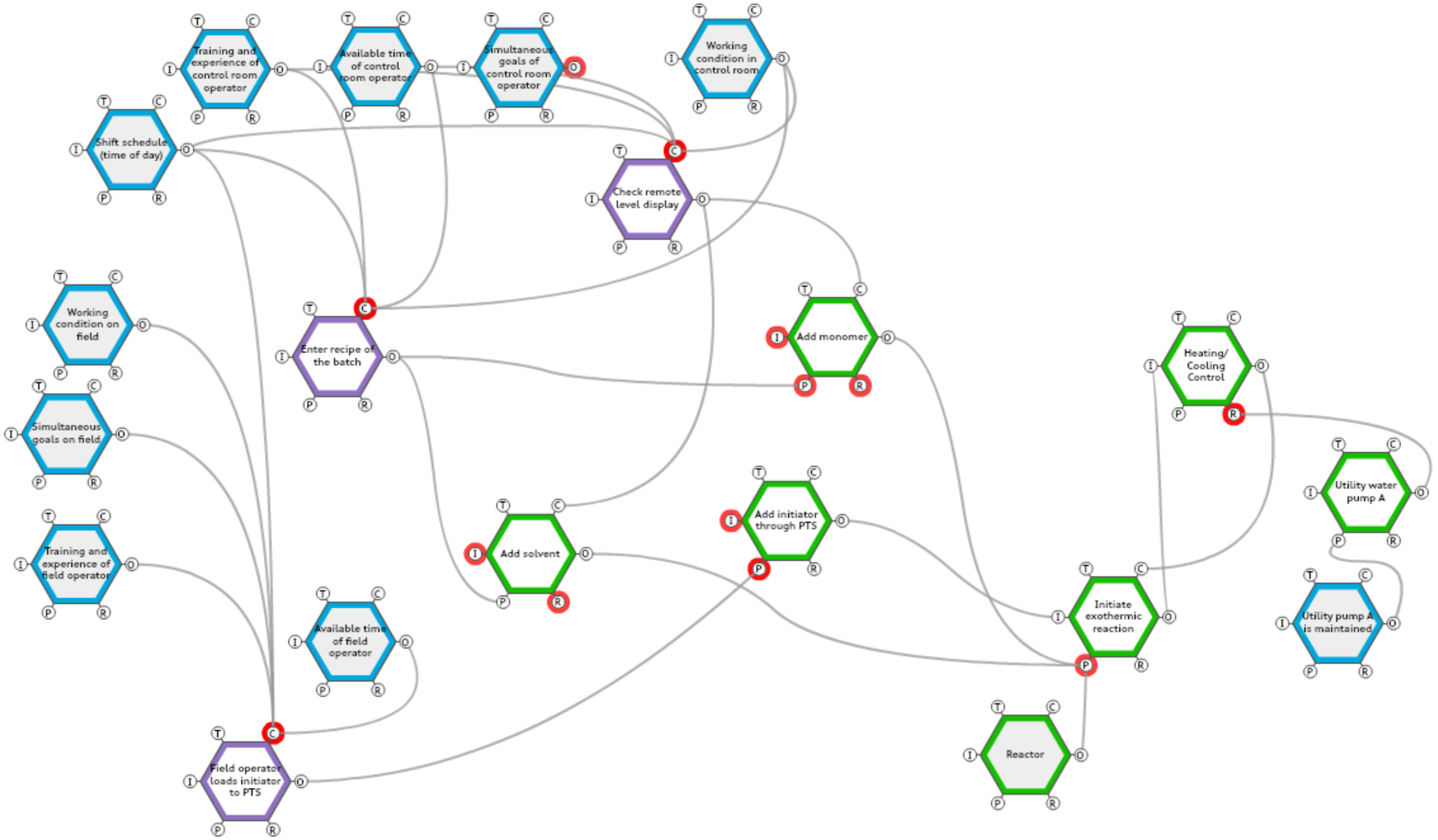

Training and experience of control room operator
Available time of control room operator
Simultaneous goals of control room operator
Working condition in control room
Shift schedule (time of day)
Check remote level display
Working condition on field
Enter recipe of the batch
Add monomer
Heating/ Cooling Control
Simultaneous goals on field
Utility water pump A
Add initiator through PTS
Training and experience of field operator
Add solvent
Initiate exothermic reaction
Utility pump A is maintained
Available time of field operator
Reactor
Field operator loads initiator to PTS

Figure III-13 Simplified FRAM with 20 functions after functions with invariant performance were removed. Red circles indicate where an upstream function was removed. Organization functions are in blue. Human functions are in purple. Technological functions are in green. Shaded functions in grey represent the boundary of the system.

Table III-9 Output variables from simulations representing performances of functions

Function	Variables	Variable type	Variable Description
Enter recipe of the batch	Cm0_cp	Numerical	Monomer concentration that is entered on the control panel
Check remote level display	I1O3_error	Categorical	Human error occurrences and error modes
Field operator loads initiator to PTS	mass_i0	Numerical	Mass of initiator that is loaded into the PTS
Available time of control room operator	avai_time_1	Categorical	Levels of available time
Available time of field operator	avai_time_2	Categorical	
Shift schedule (time of day)	time_of_day	Categorical	Time to carry out tasks
Simultaneous goals of control room operator	no_simu_goal_1	Categorical	Levels of simultaneous goals
Simultaneous goals of field operator	no_simu_goal_2	Categorical	
Training and experience of control room operator	ade_training_1	Categorical	Levels of training and experience
Training and experience of field operator	ade_training_2	Categorical	
Working condition in control room	work_cond_1	Categorical	Levels of working condition
Working condition on field	work_cond_2	Categorical	
Utility pump A is maintained	pm	Boolean	Whether PM is carried out as scheduled
Initiate exothermic reaction	T	Numerical	Reactant temperature (time-series)
Utility water pump A	Fcw_max	Numerical	Maximum utility water flow rate (time-series)

Table III-9 Continued

Function	**Variables**	**Variable type**	**Variable Description**
Heating/Cooling control	Power	Numerical	Power of heater (time-series)
	Fcw	Numerical	Utility water flow rate (time-series)
Reactor	fouling	Boolean	Whether the reactor has a fouling condition
Add initiator through PTS	mass_i0	Numerical	Mass of initiator that starts a reaction
Add solvent	Cs0	Numerical	Concentration of solvent that starts a reaction
Add monomer	Cm0	Numerical	Concentration of monomer that starts a reaction

3.8 Summary

It has been discussed in subchapter 2.3.1 that FRAM still has limitations for applications in process industries in terms of quantification and automation. Without quantification of performance variabilities, upstream-downstream couplings cannot be aggregated appropriately, resulting in misunderstandings of the interactions that lead to emerging hazards. Additionally, in order to improve the automation of FRAM applications in process industries, modeling function interactions in a hybrid system is necessary.

This chapter presents the framework which allows quantifying performance variabilities and model function interactions through integrating a human performance model, an equipment performance model, and a first-principle model. A hybrid simulator has been established to simulate possible function interactions thus manual efforts to analyze function interactions can be significantly reduced. With the data generated from the hybrid simulator, the framework provides fundamentals to further quantify upstream-downstream couplings and identify the interactions leading to emerging hazards automatically through a data-driven approach.

CHAPTER IV

IDENTIFY FUNCTION INTERACTIONS LEADING TO EMERGING HAZARDS [*]

4.1 Introduction

In a complex system, each function has performance variabilities. In traditional methods of hazard analysis such as domino and swiss cheese models, focuses are those performance variabilities that contribute to negative outcomes (Hollnagel, 2017). However, it is also important to realize some functions perform in a way to achieve the success of the system, even though the outcome is negative. Compared to the traditional analysis methods, the rationale of the FRAM is to treat behaviors of a complex system as resonance phenomena. The resonance results from interactions of the functions that can perform in either a positive or negative way (Hollnagel, 2017). Instead of only emphasizing malfunctions and failure modes, the FRAM also identifies couplings of performance variabilities which potentially lead to hazards when a system is functioning. The identified couplings are helpful to understand how the resonance occurs by looking at the underlying causation relationships. But when no root causes can be addressed, the ways of managing the couplings could be beyond looking at the cause-effect links, including isolating the dependencies or weakening resonance effects through dampening

[*] Part of this chapter is reprinted with permission from "Development of a FRAM-based framework to identify hazards in a complex system" by Mengxi Yu, Noor Quddus, Costas Kravaris, M. Sam Mannan, 2020. *Journal of Loss Prevention in the Process Industries*, 63, Pages 103994, Copyright 2019 by Elsevier Ltd. and from "A data-driven approach of quantifying function couplings and identifying paths towards emerging hazards in complex systems" by Mengxi Yu, Madhav Erraguntla, Noor Quddus, Costas Kravaris, 2021. Process Safety and Environmental Protection, 150, 2021, Pages 464-477, Copyright 2021 Institution of Chemical Engineers. Published by Elsevier B.V.

the performance variability which has negative outcomes or amplifying the one which has positive outcomes (Hollnagel, 2017; Patriarca *et al.*, 2018).

However, FRAM is originated as a qualitative technique. In order to understand how functions interact in process plants, the work presented in previous chapter has provided solutions to quantify performance variabilities of functions and simulate function interactions through a hybrid simulator, which brings opportunities to further understand how functions interact leading to potential hazard scenarios. However, simulations can generate a large amount of data, which is challenging to interpret manually. The number of possible interactions among the functions could exponentially increases as more functions are involved in the system. Extraction of useful information and knowledge from the simulated data requires data mining techniques. Therefore, this chapter aims to identify function interactions leading to emerging hazards based on FRAM by providing a data-driven solution.

4.2 Framework Overview

Figure IV-1 shows the overview of the framework to identify the interactions leading to hazard scenarios in a complex system. First, information of possible function interactions in the system were collected. Functions involved in the selected system and their performance variabilities were identified based on FRAM, then the data of possible function interactions could be synthesized from simulations through the framework developed in chapter III or collected from fields. Next, upstream-downstream function couplings were quantified by association rule mining and the bootstrap. Lastly, to interpret and manage the performance variabilities leading to emerging hazards, paths leading to the hazards were identified by post-processing association rules. The paths would show how performance variabilities of functions interact and provide guidance for people to prevent emerging hazards.

Chapter III has presented the details about generating data through modeling functions in a hybrid simulator, therefore the chapter focuses on quantifying function couplings and identifying paths leading to hazard scenarios. The work of this chapter is demonstrated via the same case study of the batch polymerization process in Chapter III and by utilizing the synthesized data from Chapter III.

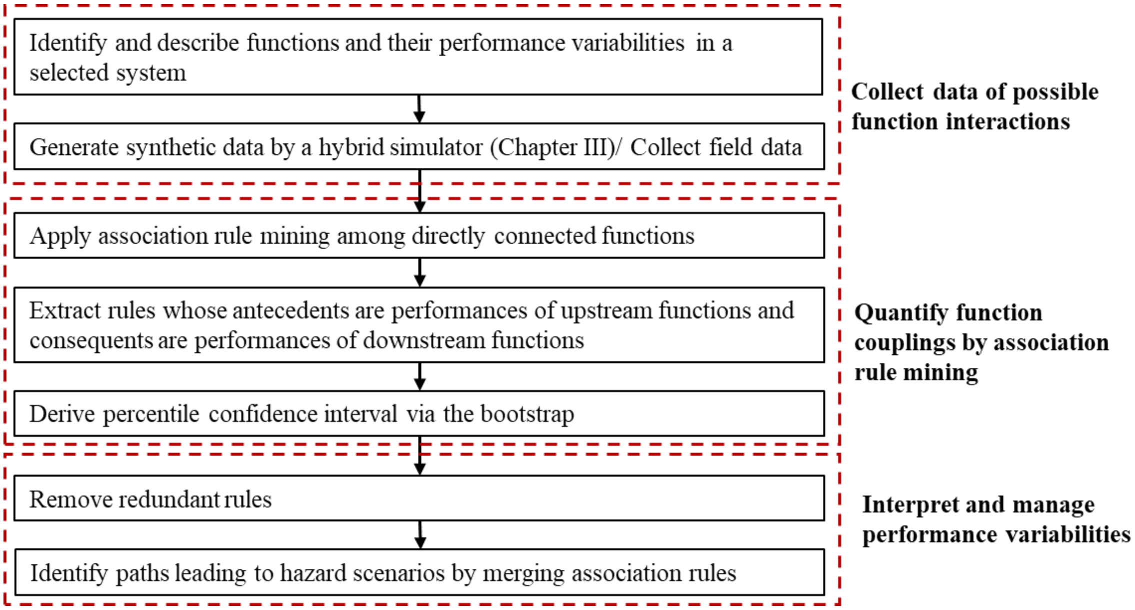

Identify and describe functions and their performance variabilities in a selected system
Generate synthetic data by a hybrid simulator (Chapter III)/ Collect field data
Collect data of possible function interactions
Apply association rule mining among directly connected functions
Extract rules whose antecedents are performances of upstream functions and consequents are performances of downstream functions
Derive percentile confidence interval via the bootstrap
Quantify function couplings by association rule mining
Remove redundant rules
Identify paths leading to hazard scenarios by merging association rules
Interpret and manage performance variabilities

Figure IV-1 Framework overview

4.3 Quantify Function Couplings in FRAM

4.3.1 Association Rule Mining

In order to understand the interactions that are relevant to hazard scenarios, association rule mining was applied to identify the couplings among upstream and downstream functions. Association rule mining has been widely applied by researchers to identify causation relationships or patterns from incident records (Anand *et al.*, 2006; Bevilacqua & Ciarapica, 2018; Cheng *et al.*, 2010; Hu & Guo, 2016; Keren *et al.*, 2006; Verma *et al.*, 2014). It finds the associations by exhaustively searching combinations of attribute values, which are called as item sets. The attribute values in the item sets are split to antecedents and consequents of rules. The rules that are generated from the association rule mining are formatted as A=>C, where A is the antecedent set and C is the consequent set. In the context of the study, the item sets to be searched were combinations of function performances.

The association between A and C is commonly described by three metrics: support, confidence, and lift (Han *et al.*, 2011). Their formulars are shown as follows.

$$\text{Support (A=> C)} = \text{Probability (A} \cap \text{C)} \qquad \text{(Equation IV-1)}$$

$$\text{Confidence (A => C)} = \text{Probability (C | A)}$$

$$= \frac{\text{Probability (A} \cap \text{C)}}{\text{Probability (A)}} \qquad \text{(Equation IV-2)}$$

$$\text{Lift (A=> C)} = \frac{\text{Probability (A} \cap \text{C)}}{\text{(Probability(A) Probability (C))}} \qquad \text{(Equation IV-3)}$$

Support and confidence are the most common metrics that are used to find strong association rules based on their minimum thresholds. When exploring frequent patterns in the contexts such as market analysis, large values of support and confidence indicate strong associations. However, they are not proper metrics given the context of the study to study hazardous situations. Hazardous situations occur more rarely compared to normal situations, but the interactions that lead to hazardous situations are more meaningful for the purpose of hazard identification. With the consequent set being a rare hazardous situation, whose frequency is extremely low, the values of the two metrics are expected to be extremely low as well. Additionally, when the consequent set is a normal situation, the frequency of the consequent set is high leading to high values of support and confidence, regardless of what interactions are shown in the antecedent set. Thus, support and confidence do not help with identifying the interactions leading to normal situations either. Instead of relying on support and confidence, lift value was used to study the interactions between an antecedent set and a consequent set since the lift value is scaled by the frequency of the consequent set. When lift (A=> C) =1, the occurrences of A and C are independent. When lift (A=> C) > 1, the occurrences of A is positively associated with occurrences of C, and C occurs more frequently when A is present. When lift (A=> C) < 1, the occurrences of A and C are negatively associated, and C occurs less frequently when A is present (Han *et al.*, 2011).

The MLxtend library in python programming was applied to find frequent item sets based on Apriori algorithm and extract association rules (Raschka, 2018). Apriori algorithm extracts frequent item sets based on a minimum support threshold. If the frequency of any subset of a frequent itemset is smaller than the minimum support threshold, the frequent itemset will be neglected to further generate association rules (Han *et al.*, 2011). There is no explicit rule of thumb to set the minimum threshold for rule extractions, since the threshold setting depends on contexts of studies, available amount of data and different perceptions of meaningful rules (Verma *et al.*, 2014).

In the study, to extract the interesting rules which were related to the rare abnormal situations, the minimum support threshold was set as the inverse of the data size, meaning that the co-occurrence frequency of an antecedent and consequent was at least 1. The minimum lift threshold was set as 1 to find all the positive associations. In order to quantify upstream-downstream couplings, association rule mining was only applied between a downstream function and its directly connected upstream functions. Taken the FRAM in Figure II-1 as an example, association rule mining would be only conducted (1) between functions 1, 2, and 3, (2) between Function 2 and 4, and (3) between function 4 and 5. Additionally, since the rule mining process does not distinguish upstream and downstream functions, antecedents of rules could contain performances of downstream functions while consequents could contain performances of upstream functions. Therefore, another rule extraction step was conducted to obtain the rules whose antecedents were upstream functions and consequents were downstream functions. Lastly, lift values of such association rules were used to indicate the couplings between upstream and downstream functions.

4.3.2 Confidence Intervals of Lift by the Bootstrap

Lift is a descriptive metric to represent the associations between antecedents and consequents in a given dataset. The calculation of lift solely depends on the statistics in the given data and does not address uncertainties related to randomness in the dataset. To develop lift CIs, the bootstrap was applied. The bootstrap is a widely applicable statistical method to estimate the sampling distribution of a metric, such as a statistic or a model parameter (Bruce & Bruce, 2017; James *et al.*, 2013; Wehrens *et al.*, 2000). It does not involve any assumption about data or metrics such as being normally distributed (Bruce & Bruce, 2017). The bootstrap is a process of empirical sampling with replacements from the original dataset for multiple times. The interested metric is obtained from each empirical sampling. Thus, with sufficient iterations of sampling, the distribution of the metric can be obtained and CI of the metric can be further calculated (Wehrens *et al.*, 2000).

Figure IV-2 shows the process of the bootstrap embedded with association rule mining. Starting with the categorized synthetic data with *m* records, the bootstrap process was replicated for *n* times to obtain *m*-size sample data each time. *R* represents the set of association rules which were extracted from the original dataset, and R_i represents a rule in the *R*. By calculating the lift of R_i from each bootstrapped sample, an empirical distribution consisted of *n* lifts was obtained for the R_i, which would be used to obtain the lift CI for the R_i. Exceptions occurred when the combination of the antecedent and consequent of R_i only occurred a few times in the original dataset, thus the rule would not show up in some bootstrapped replicates. In such cases, the lift CI from *n* bootstrap replicates did not exist. The nonexistent CI indicates such rules are not supported by enough data in the given dataset thus become unstable (Waitman *et al.*, 2006). Such rules were not considered for further analysis in the study to illustrate the methodology but reliable lift CIs for such rules can be easily obtained by expanding the size of the synthetic data through more simulations. Lastly, for the rules which had n lifts from the bootstrap replicates, percentile CIs of their lifts were derived through Equation IV-4 (Efron & Tibshirani, 1994). $\hat{l}^*$ represents the distribution of the lift that derived from the bootstrap replicates for an association rule. With the confidence level at 1-2α, the lower limit of confidence interval ($\hat{l}_{\%,low}$) is 100* αth empirical percentile, and the upper limit ($\hat{l}_{\%,up}$) is 100*(1-α)th empirical percentile.

$$[\hat{\boldsymbol{l}}_{\%,\boldsymbol{low}}, \hat{\boldsymbol{l}}_{\%,\boldsymbol{up}}] = [\hat{\boldsymbol{l}}^{*(\boldsymbol{\alpha})}, \hat{\boldsymbol{l}}^{*(\boldsymbol{1-\alpha})}] \qquad \text{(Equation IV-4)}$$

In the study, the percentile confidence interval of lift value was derived from 1,000 bootstrap replicates at 99% CI.

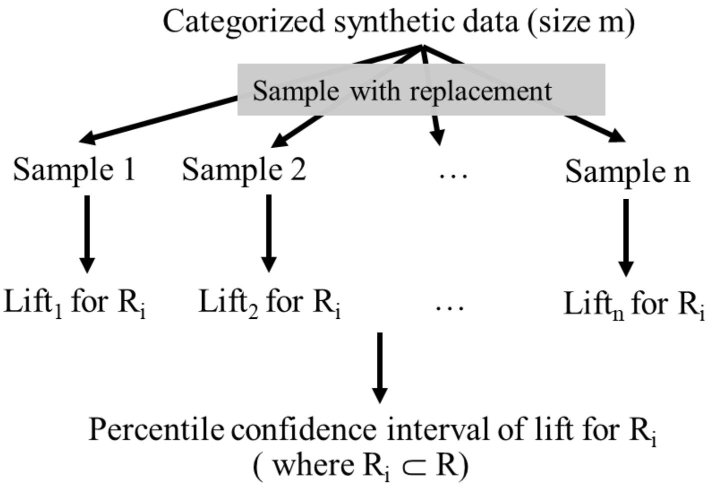


Figure IV-2 The bootstrap process to obtain the percentile confidence interval of lift

4.4 Interpret and Manage Performance Variabilities

To ease the process of interpreting and managing the performance variabilities, rules were postprocessed through two steps: removing redundant rules and identifying the paths leading to hazard scenarios. For instance, a rule was redundant when it contained more elements in the antecedent than another rule but did not provide additional information (Batbarai & Naidu, 2014; Jiawei & Yongjian, 1999). The metrics to remove redundant rules could be different depending on the context. Given the context of the study, a rule was removed when the rule had additional upstream function performances in the antecedent than another rule had but did not show increased couplings with the same downstream function performance. Assuming there were two rules containing the same downstream function performance:

Rule 1: A1, A2 → C, lift CI = [Lift_low_1, Lift_high_1]

Rule 2: A1, A2, A3 → C, lift CI = [Lift_low_2, Lift_high_2]

Rule 2 was redundant if

(Lift_high_2 < Lift_low_1) or; (Equation IV-5)

(Lift_high_2 = Lift_high_1) and (Lift_low_2 = Lift_low_1) (Equation IV-6)

However, the removal process would not remove all redundant rules. For example, it was possible to derive Rule 2 whose Lift_low_2 > Lift_high_1. In this case, Rule 1 could be redundant depending on how much its lift CI was increased by adding the element A3 to its antecedent. For example,

Rule 1 was redundant if Lift_low_2 was much greater than Lift_high_1, Rule 2 was redundant if Lift_low_2 and Lift_high_2 were slightly greater than Lift_high_1.

A quantitative criterion was needed to define “much” and “slightly” for refining the removal process but has not been defined at the current stage of the study. The only purpose of removing redundant rules was to reduce the number of interactions that were identified thus reduced the manual efforts for reviewing. Since the study focused on how interactions could be identified rather than minimizing redundant interactions, rules were removed conservatively based on the conditions in Equations IV-5 and IV-6.

After removing such redundant rules, rules were merged to form hazard paths. For instance, C was the hazardous performance of the target function, and performances of two upstream functions, A1 and A2, impacted C positively. To identify the paths leading to C, rules whose consequents were A1 or A2 would be extracted and merged with A1, A2 → C. The merging process was stopped until the furthest upstream functions in the system were reached, or the functions which performed as desired in a normal operation were reached. After a few iterations, paths leading to the hazardous scenario C were identified. Such paths with quantified couplings are essential to understand how the hazardous scenario occurs, as well as guide people to determine appropriate measures to prevent the paths from emerging in operations.

4.5 Coupling and Causation

As mentioned earlier, FRAM is a technique to identify function couplings leading to a resonance scenario. Even though further analysis of managing resonance effects could include identification of underlying causation relationships, it is critical to understand causality inferences from couplings need expert knowledge or contextual information. Pearl and Mackenzie (2018) proposed a three-step ladder of causation theory, including association at the bottom ladder, intervention at the middle, and counterfactual analysis at the top. Association is from observations, which permits inferences and predictions of future events. Association could be observed between X and Y when they have a cause-effect relationship or they share a common cause. Explicit causal analysis requires further steps to investigate the association between X and Y in a changing context (Pearl, 2010; Pearl & Mackenzie, 2018), such as how Y is changed along with changes of X (i.e. intervention), and whether Y can occur due to other factors when X does not occur (i.e. counterfactuals). In a complex system, a hazard scenario is a result of function interactions. The scenarios could occur even when some functions work as desired to dampen the undesired resonance rather than contribute to the hazard. The scope of the study was to be consistent with the FRAM and to identify the function couplings potentially leading to hazard scenarios. The couplings would show significant beliefs about occurrences of hazard scenarios and provide guidance for further causal analysis.

4.6 Case Study: Batch Polymerization Process

Uncontrollable temperature excursions in batch polymerization process could develop to fires and explosions very quickly. This section is to illustrate how the interactions leading to hazardous temperature excursions were identified through a data-driven approach based on association rule mining. The batch polymerization process which has been used as the case study in Chapter III was also used for the illustration in this chapter. The synthetic data of 879,000 batch operations was processed and utilized for the analysis in this chapter.

4.6.1 Data Description

Table III-9 provided descriptions of raw data that was generated from simulations. As the table showed, performances of functions were collected in different formats. For example, performance of the function "Initiate exothermic reaction" was collected as time-series temperature during an 8-hour batch, while performance of the function "Enter recipe of the batch" was collected as a single value in a batch. The raw data needed to be processed to apply association rule mining. Features could be extracted from the time-series data depending on study interests. For example, the maximum temperature of a batch operation was extracted as the interested performance of the function "Initiate exothermic reaction" in the study. Additionally, since the variables that were extracted from the simulated data were a mix of numerical and categorical variables, the numerical variables were discretized to categorical values to apply the association rule mining algorithm. The output variables in Table III-9 has been processed and the corresponding new variables are provided in Table IV-1. Temperature excursions were defined when the maximum temperature of a batch operation was 10% higher than that during a normal operation.

Table IV-1 Processed output variables representing performances of functions

Function	Variables	Variable Description	Values and Descriptions of Variables
Enter recipe of the batch	Cm0_cp	Monomer concentration that is entered on the control panel	0: recipe amount 1: less than recipe amount 2: more than recipe amount
Check remote level display	I1O3_error	Human error occurrences and error modes	0: no human error 1: faulty diagnosis 2: observation missed
Field operator loads initiator to PTS	mass_i0	Mass of initiator that is loaded into the PTS	0: recipe amount 1: less than recipe amount 2: more than recipe amount
Available time of control room operator	avai_time_1	Levels of available time	1: adequate 2: temporarily inadequate 3: continuously inadequate
Available time of field operator	avai_time_2		
Shift schedule (time of day)	time_of_day	Time to carry out tasks	1: day-time 2: night-time
Simultaneous goals of control room operator	no_simu_goal_1	Levels of simultaneous goals	1: fewer than capacity 2: matching current capacity 3: more than capacity
Simultaneous goals of field operator	no_simu_goal_2		
Training and experience of control	ade_training_1	Levels of training and experience	1: adequate training and high experience 2: adequate training and low experience 3: inadequate training
Training and experience of field	ade_training_2		
Working condition in control room	work_cond_1	Levels of working condition	1: advantageous 2: compatible 3: incompatible
Working condition on field	work_cond_2		

Table IV-1 Continued

Function	Variables	Variable Description	Values and Descriptions of Variables
Utility pump A is maintained	pm	Whether PM is carried out as scheduled	0: no pm carried out 1: pm was carried out as scheduled
Initiate exothermic reaction	max_temp	Maximum temperature of reaction	0: no temperature excursion 1: temperature excursion occurs
Utility water pump A	pumpfail	Whether the pump fails	0: pump does not fail 1: pump fails
Heating/Cooling control	heat_kJ	Total heat provided in kJ	0: same as the heat provided during normal operation 1: less than the heat provided during normal operation 2: more than the heat provided during normal operation
	cooling_water_m3	Total cooling water supplied in m^3	0: same as the cooling water provided during normal operation 1: less than the cooling water provided during normal operation 2: more than the cooling water provided during normal operation
Reactor	fouling	Whether the reactor has a fouling condition	0: no fouling 1: fouling exists
Add initiator through PTS	mass_i0	Mass of initiator that starts a reaction	0: recipe amount 1: less than recipe amount 2: more than recipe amount
Add solvent	Cs0	Concentration of solvent that starts a reaction	0: recipe amount 1: less than recipe amount 2: more than recipe amount
Add monomer	Cm0	Concentration of monomer that starts a reaction	0: recipe amount 1: less than recipe amount 2: more than recipe amount

4.6.2 Quantification of function couplings by lift CI

Effects of function couplings are difficult to be manually estimated, especially when multiple functions are involved and impact downstream functions in different ways. Therefore, it is necessary to find a quantitative measure of function couplings. In order to investigate whether lift can be used as the quantitative measure, the relationship between lift and function couplings needs to be analyzed through the comparison with ground truths or theoretically known couplings. Human performance modeling method that was used by the simulator is CREAM, which models the interactions between organization and human functions (Hollnagel, 1998). In CREAM, a weight is assigned to the performance of an organization to describe its coupling with a specific human performance. An aggregated coupling of multiple organization functions on a human performance is defined as the product of the weights as shown in Equation III-1. The product of weights is the theoretical coupling thus it was compared with lift.

As an example, couplings between the human function "Enter recipe of the batch" and its upstream organization functions were analyzed to compare lifts with the theoretical couplings. In CREAM, weights are assigned for the execution error, which covers the cases when the monomer concentration that entered on the control panel was less or more than the recipe amount. Thus, an additional performance of "Enter recipe of the batch" was categorized temporarily as "incorrect amount". Lift CIs were obtained from the rules whose consequents were "incorrect amount" and antecedents were the performances of its upstream organization functions. Since multiple rules could have antecedents corresponding to the same product of weights, the rules were grouped by the product of weights, and the maximum upper limit and the minimum lower limit of lift CIs were extracted from each group. Figure IV-3 plots the lift CI limits with the products of weights and shows strong positive correlations.

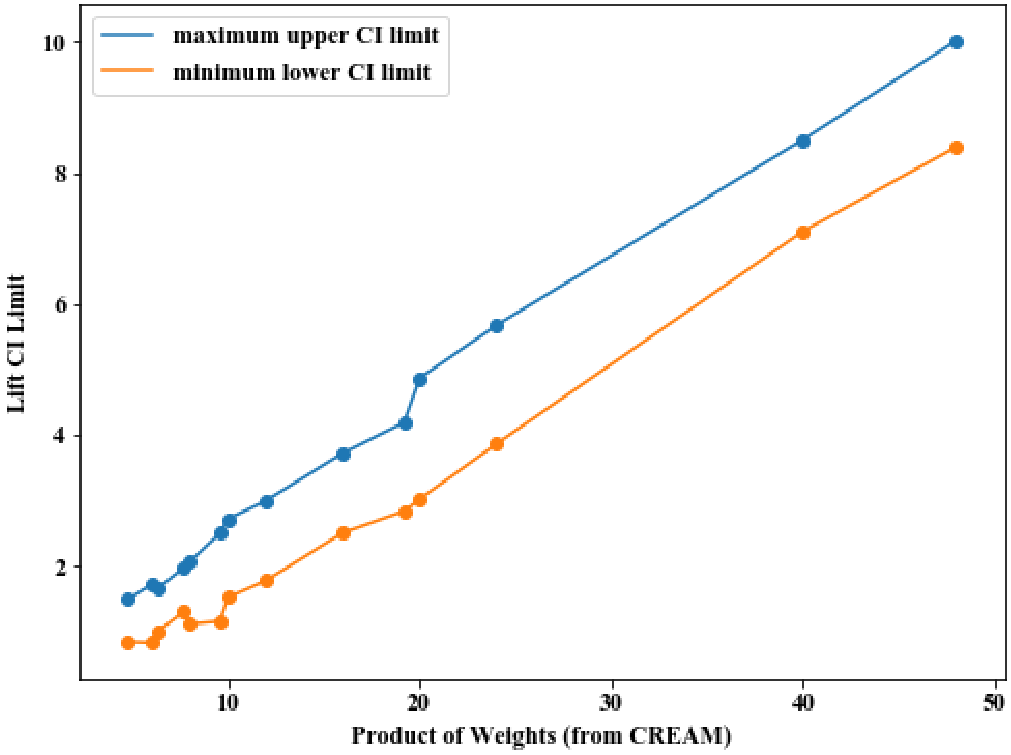


Figure IV-3 Comparison between lift CI limits and theoretical couplings (i.e., product of weights derived from the CREAM)

Additionally, in the cases that multiple rules were with the same product of weights, lift CIs of such rules were plotted by their products of weights as comparisons. As examples, Figure IV-4 a and b show lift CIs of the rules whose product of weights were

8 and 20 accordingly. For the rules with the same product of weights, lifts obtained from the entire dataset (represented by dots) fluctuated, but with the CIs, a common range of lifts could be found corresponding to a product of weights.

Summarizing, the patterns in Figure IV-3 and IV-4 showed lift CIs can quantitively represent the couplings between upstream and downstream functions.

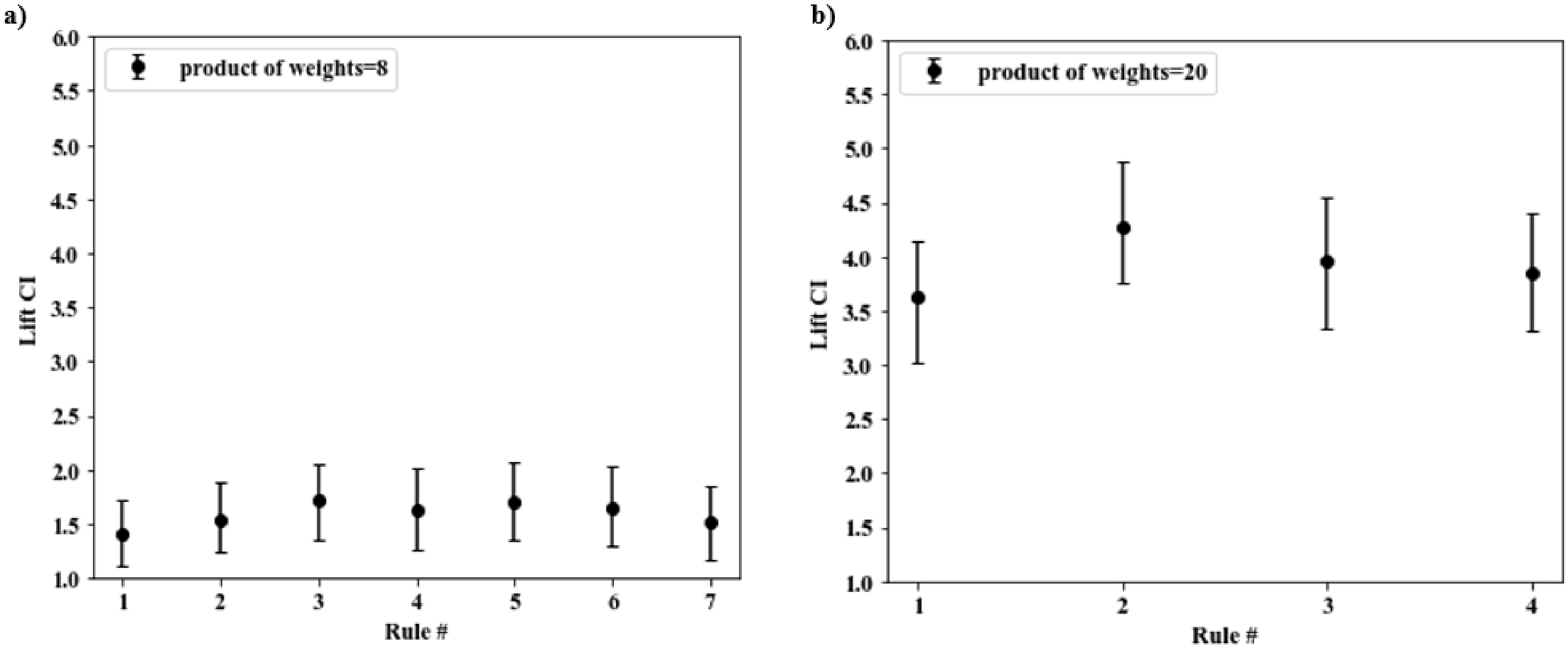


Figure IV-4 Comparison between lift CIs of association rules with the same theoretical couplings (i.e. the same product of weights derived from the CREAM)

In the studied system, theoretical couplings between functions were unknown except for the ones that were modeled by the CREAM. It was similar to realities where theoretical couplings are unknown and a data-driven approach to quantify couplings is essential. Given the positive correlations between lift CIs and the theoretical couplings modeled by the CREAM, association rule mining was conducted between the directly connected upstream-downstream functions to demonstrate the interpretations of lift CIs. Since the performance "temperature excursion occurs" ("max_temp = 1") of the function "Initiate exothermic reaction" was the hazard scenario of interest, rules for the consequent "max_temp =1" were first extracted. Table IV-2 provides all the rules, which contained individual upstream function performances and showed positive associations, to understand which upstream function performance had the strongest coupling with temperature excursions. Descriptions of upstream functions or antecedents have been provided in Table IV-1. Some important interpretations of the rules are discussed below:

- Table IV-2 shows "Cm0 = 2" and "Cs0 = 1" had the strongest coupling with temperature excursions (i.e. rule 1 and rule 2). Their lift CI were exactly the same indicating they had the same coupling strengths with temperature excursions, which was consistent with the fact that the initial monomer and solvent concentrations were dependent in the system.
- On the other hand, rule 7 shows when the function "Add initiator through PTS" added the initiator with the recipe amount, the performance very weakly coupled with temperature excursions according to the narrow lift CI around 1. However, when the antecedent shows the fewer amount of initiator started the reaction

("mass_i0 = 1") in rule 8, lift at 1.36 indicates the antecedent had positive couplings with temperature excursions, which was contrary to the reaction kinetics. This was an example that explained well why uncertainty information provided by CI is critical for interpreting rules. Even though the lift of rule 8 is greater than 1, its CI is wide-ranging from 0.59 to 2.2, indicating the dataset could not provide enough evidence to conclude how the fewer mass of the initiator was coupled with temperature excursion. More data will be needed to quantify the coupling more precisely.

- Rule 5 with the antecedent "fouling = 1" shows fouling in the reactor was coupled with temperature excursions, but the coupling was much weaker compared to the couplings with "Cm0 = 2" and "Cs0 = 1".
- The rules mentioned so far were straightforward since there was no complex relationship and the couplings could be easily explained based on the knowledge of the system. However, regarding the rest of the rules (i.e., rule 3, 4, and 6) in Table IV-2, whose antecedents are performances of the function "Heat/Cooling Control", the closed loop between the functions "Heat/Cooling Control" and "Initiate exothermic reaction" made the interpretation of the rules less intuitive. For example, rule 3 indicates less heat provided by the control system was strongly coupled with temperature excursions. It would be spurious to simply interpret the coupling as a causation relationship.

Table IV-2 Association rules between temperature excursion ("max_temp = 1") and its upstream functions

Rule #	Antecedent	Consequent	Lift	Lift CI
1	Cm0 = 2	max_temp = 1	75.4	[71.23, 79.62]
2	Cs0 = 1	max_temp = 1	75.4	[71.23, 79.62]
3	heat_kJ=1	max_temp = 1	68.04	[66.04, 70.16]
4	coolingwater_m3 = 1	max_temp = 1	42.5	[39.48, 45.23]
5	fouling =1	max_temp = 1	1.54	[1.49, 1.59]
6	coolingwater_m3 = 2	max_temp = 1	1.14	[1.09, 1.2]
7	mass_i0 = 0	max_temp = 1	1(rounded)	[0.99, 1.01]
8	mass_i0 = 1	max_temp = 1	1.36	[0.59,2.2]

To better understand the interactions between the functions "Heat/Cooling Control" and "Initiate exothermic reaction" within the loop, association rule mining was conducted to extract the rules whose consequents were the performances of the function "Heat/Cooling Control". As Figure III-13 shows, upstream functions of the function "Heating/Cooling Control" were "Initiate exothermic reaction" and "Utility water pump A". Thus, the rules, which contained the performances of the two upstream functions as their antecedents, were extracted and provided in Table IV-3.

Since lift measures couplings instead of causation, expert knowledge is needed to interpret the rules based on the knowledge of the system. Especially when functions are mutually coupled in a loop, symmetric rules can be generated since lift only measures mutual dependencies and does not change when the antecedent and consequent simply switch orders. Expert inputs are needed to interpret the causations hidden in the symmetric rules. For example, rule 3 in both Table IV-2 and Table IV-3 is a pair of symmetric rules. According to the design of the heating/cooling control system, less heat will be provided by the control system when the reactant temperature is higher than the normal temperature. Thus, rule 3 in Table IV-3 with temperature excursions "max_temp = 1" as the antecedent is in the format which is consistent with the underlying causality. Similarly, rule 4 in Table IV-2 and rule 6 in Table IV-3 are symmetric, showing temperature excursions were coupled with less cooling water provided by the heat/cooling control system ("cooling water_m3_1"). The rules indicate temperature excursions were very likely to occur if more cooling could not be provided by the control system when reactant temperature exceeded the normal temperature. Another pair of symmetric rules is rule 6 in Table IV-2 and rule 8 in Table IV-3, showing the associations between "max_temp = 1" and "cooling water_m3_2". The heat/cooling control system was designed to provide more cooling water when the temperature in the reactor was higher than the normal temperature. Thus, the associations between the two function performances were consistent with the design. However, given the lifts are slightly greater than 1, it indicates providing more cooling water could not completely prevent temperature excursions. The couplings highlight even though the performance of "cooling water_m3_2" had positive impacts on the temperature control, it could not always dampen the performance variability of "initiate exothermic reaction" to avoid temperature excursions.

Besides, expert knowledge is needed to identify spurious associations within the loop. Spurious associations refer to the relationships between components which are associated without direct causal relationships. Rule 1 and Rule 2 in Table IV-3 shows the coupling between pump failure ("pumpfail =1") and less heat provided by the heat/cooling control system ("heat_kJ_1"). However, the performance of the function "Utility water pump A" did not impact the total heat provided by the control system based on the system design. The two performances were associated since pump failures could cause "cooling_water_m3_1" (i.e., rule 5 in Table IV-3) which further could cause "max_temp =1" (i.e., rule 4 in Table IV-2 and rule 6 in Table IV-3), meanwhile, "max_temp =1" could result in "heat_kJ_1" (i.e., rule 3s in Table IV-2 and IV-3) regardless of whether "max_temp=1" was caused by "pumpfail=1".

Table IV-3 Association rules whose consequents are performances of the function "Heat/Cooling Control"

Rule #	Antecedent	Consequents	Lift	Lift CI

1	pumpfail = 1,
	max_temp =1
		heat_kJ_1	79.05	[77.26, 81.21]

2	pumpfail = 1		76.22	[74.01, 78.54]
3	max_temp=1		68.04	[66.04, 70.16]

Table IV-3 Continued

Rule #	Antecedent	Consequents	Lift	Lift CI

4	pumpfail = 1,
	max_temp =1
		cooling water_m3_1	89.09	[85.47, 92.67]

5	pumpfail = 1		82.51	[79.02, 86.04]
6	max_temp=1		42.5	[39.48, 45.23]

7	pumpfail = 0,
	max_temp = 1
		cooling water_m3_2	1.98	[1.98, 1.99]

8	max_temp = 1		1.14	[1.09, 1.2]

As discussed above, even though expert inputs are needed to ensure rules are interpreted appropriately, quantified couplings by association rule mining provide guidance to identify contributing factors of the hazard and bring to light the emerging hazardous scenario which is resulted from both positive and negative performances of upstream functions. In addition, another benefit from quantified couplings is to identify the most important couplings. In Table IV-3, lift of rule 4 is about twice the lift of rule 6, meaning rule 4 is much more important since it simply contains an extra function performance in the antecedent and shows a much stronger association with the consequent. On the other hand, compared to rule 5, rule 4 contains an extra function performance in the antecedent but results in a very similar association which is only about 8% higher. Comparing these three rules, rule 5 is the most important one reflecting the strong association between pump failures and "cooling water_m3_1". To minimize manual efforts in identifying important couplings, the automatic process of reducing rule redundancy need to be finetuned in the future.

4.6.3 Example paths leading to potential temperature excursions

In order to understand how performance variabilities of functions could spread within the system and lead to the hazard scenario, association rules that were extracted between the performances of directly connected functions were further merged to identify the paths of such spread. As discussed previously, there were some redundant rules such as rule 4 and rule 6 in Table IV-3 existing in the current study. Merging of such redundant rules led to redundant paths, which requires further removals in the future. This subchapter provides example paths that were identified to explain how they can be interpreted and used for managing the hazard, as well as shows examples to discuss the existence of redundant paths.

The first example of the paths, as shown in Figure IV-5, consists of 7 functions, indicating temperature excursions are likely to occur once the performances of the 7 functions are known. The path in Figure IV-5 is a part of Figure III-13, only showing necessary connections for representing an instantiation of the temperature excursion. In the figure, performances of the functions are labeled next to the connections that are originated from the "output" aspects. Descriptions of the performances are available in Table IV-1. The process of merging rules for path identification started from identifying the rules that showed positive associations with temperature excursions. In Figure IV-5, such rule had the antecedent containing the performances "Cm0=0", "mass_i0=0", "fouling = 0" and "cooling water_m3=1". Next, the rule was merged with the rules which had the individual performances as consequents. For instance, the rule which had “cooling water_m3 = 1” as the consequent and “pumpfail = 1” as the antecedent was involved in the path. The step-by-step rule merging could be a few iterations, till the furthest upstream functions in the system or the functions that perform as desired in normal operation were reached. The functions at the boundary of a path were shaded in grey. Besides, for the functions which were connected with upstream functions, lift CIs were shown in brackets to show the couplings with their upstream functions quantitively.

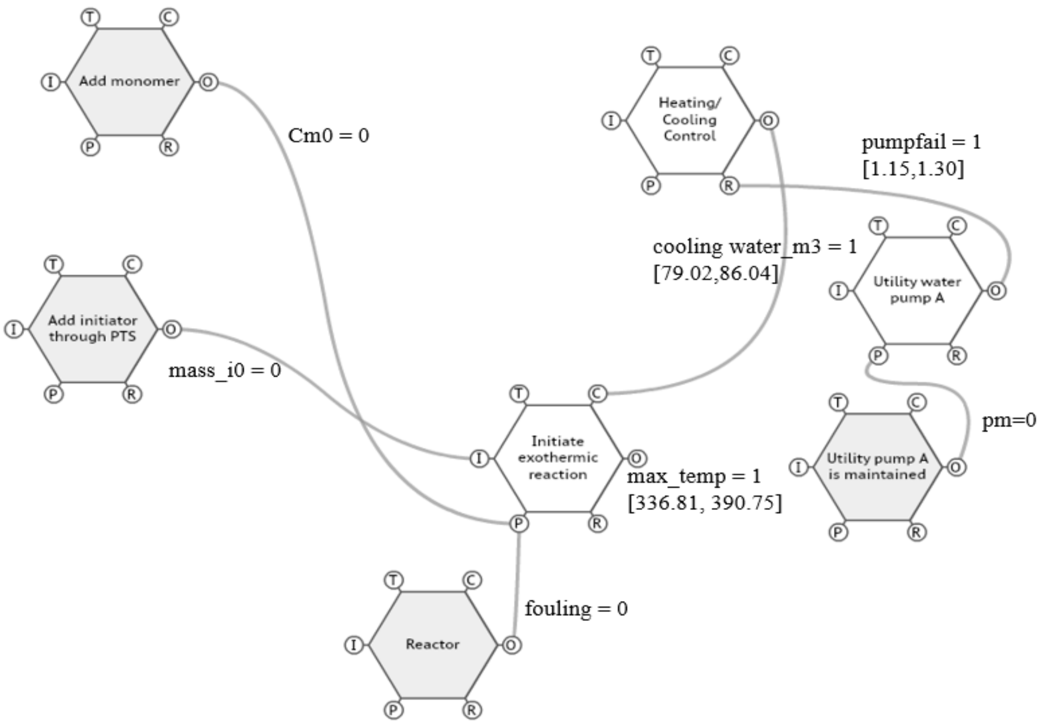


Figure IV-5 First path leading to the temperature excursion “max_temp = 1”. Function performances are provided on the connections that are originated from the output (O) aspects. Lift CI which quantifies the couplings between the directly connected functions are shown in brackets. Functions shaded in grey represent the boundary of the path.

In Figure IV-5, a strong coupling between temperature excursions with the combination of its upstream functions' performances was indicated by the lift CI at [336.81, 390.75]. The combination shows when the functions “Add monomer”, “Add initiator through PTS” and “Reactor” performed as desired, temperature excursions were likely to occur when the function “Heating/Cooling Control” provided less cooling water (“cooling water_m3 =1”). The performance “cooling water_m3 = 1” was strongly coupled with pump failures with lift CI at [79.02, 86.04]. Additionally, no PM being carried out (“pm = 0”) was coupled with pump failures but the lift CI at [1.15, 1.30] shows the coupling was relatively weak. It indicates no PM being carried out was not a dominant factor leading to pump failures. As the path shows, temperature excursions were resonant results of the interactions of the functions including those performing as desired. According to the path identified, several measures could be considered to manage the performance of “Initiate exothermic reaction”, such as designing the reactor with more heat transfer efficiency, adjusting the recipe to make the process inherently safer, integrating barriers other than PM to prevent pump failures or providing a back-up pump for redundancy.

Figure IV-6 shows another example path, which involves more complicated function couplings than the previous one. The example highlights the contribution of association rule mining in identifying the paths whose complexity could be beyond direct observations. Among the upstream functions connecting to the function “Initiate exothermic reaction” in the path, the function “Add monomer” overcharged the monomer to the reactor (“Cm0 = 2”) and the function “Reactor” had fouling (“fouling = 1”). Both performance variabilities increased the likelihood of temperature excursions. On the other hand, the function "Add initiator through PTS" added the initiator as the recipe required ("mass_i0 = 0") therefore it did not have a coupling effect on temperature excursions when it was considered alone. Additionally, since the function "Heating/Cooling Control" controlled the reactant temperature through a feedback loop, it performed as both upstream and downstream function of the function "Initiate exothermic reactions". When the function "Heating/Cooling Control" was treated as the downstream function, "cooling water m3 = 2" was coupled with temperature excursion with the lift CI [1.09,1.20]. It indicates that even when more cooling water was provided by "Heating/Cooling Control", a few temperature excursions cases still occurred. On the other hand, when the function "Heating/Cooling Control" was treated as the upstream function, the lift CI [144.41, 161.0] revealed one of the cases when "cooling water_m3_2" failed to control the reactant temperature. The case occurred when upstream functions had the performances "Cm0=2", "fouling = 1" and "mass_i0=0" and led to an aggregated effect which could not be dampened by "cooling water_m3_2". Simply the small group of function couplings suggested that the process can be improved not only by avoiding the performance such as the overcharge of the monomer, but by increasing cooling efficiency from the control system.

Additionally, the path covers how "Cm0 =2" emerged from the performance variabilities of its upstream functions. In a batch operation, the CRO entered the initial monomer concentration on the control panel then verified the recipe that was entered through the function of "Check remote level display". Ideally, if a human error occurred and ended up with a larger monomer concentration (“Cm0_cp = 2”) being entered on the control panel, the verification function was expected to fix the error then started the batch operation with the required monomer concentration through the function “Add monomer”. However, the identified path shows the upstream function couplings, which could cause the performance “Cm0_cp = 2”, could also bring variabilities to the function “Check remote level display”. In the identified path, five organization functions impacted the functions “Enter recipe of the batch” and “Check remote level display”. Two of organization functions (i.e., “Shift schedule” and “Working condition in control room”) had the performances at the most competent levels, while the other three functions had the performances at the poorest levels. The aggregated coupling of the five functions showed a positive association with both “Cm0_cp =2” and missed observation (“I1O3_error = 2”), with lift CI at [2.64, 4.02] and [3.30,3.35] accordingly. It indicates the aggregated effect of the five organization functions led to the undesired performances of both downstream functions. The undesired performances were strongly coupled with “Cm0 =2” of the function “Add monomer” further, which could start the reaction with the overcharged monomer. The identified path shows the function “Check remote level display” was not always an effective control aspect of the function “Add monomer”. In order to manage the performance of the function “Add monomer”, it is important to ensure the coupling of the five organization functions positively impact the performance of function “Check remote level display”. Otherwise, additional functions should be integrated to control the performance of “Add monomer”.

As discussed earlier, association rule mining generated redundant rules which resulted in redundant paths leading to the hazard scenario. Figure IV-7 shows a redundant path with respect to the path in Figure IV-6. Compared to the path in Figure IV-6, the redundant path does not involve the function “Enter recipe of the batch”. Without knowing the performance of “Enter recipe of the batch”, valuable information was lost to understand how the performance “Cm0 =2” of the function “Add monomer” occurred. Correspondingly, two rules were found as follows by comparing Figure IV-7 to IV-6:

Rule 1 (Figure IV-7):

I1O3_error=2 => Cm0=2, lift CI = [1.37,1.48]

Rule 2 (Figure IV-6):

I1O3_error=2, Cm0_cp=2 => Cm0 =2, lift CI = [138.98,147.40]

As discussed in subchapter 4.3, Rule 1 is redundant. With the loss of information, the lift CI between the performance “Cm0 =2” and its upstream function performance was reduced significantly from [138.98, 147.40] to [1.37, 1.48]. Therefore, the path in Figure IV-6 is a more complete path to show how the hazard emerged and the path in Figure IV-7 is redundant. The limitation regarding the redundancies of rules and paths needs to be resolved in the future to minimize manual efforts in reviewing the identified paths.

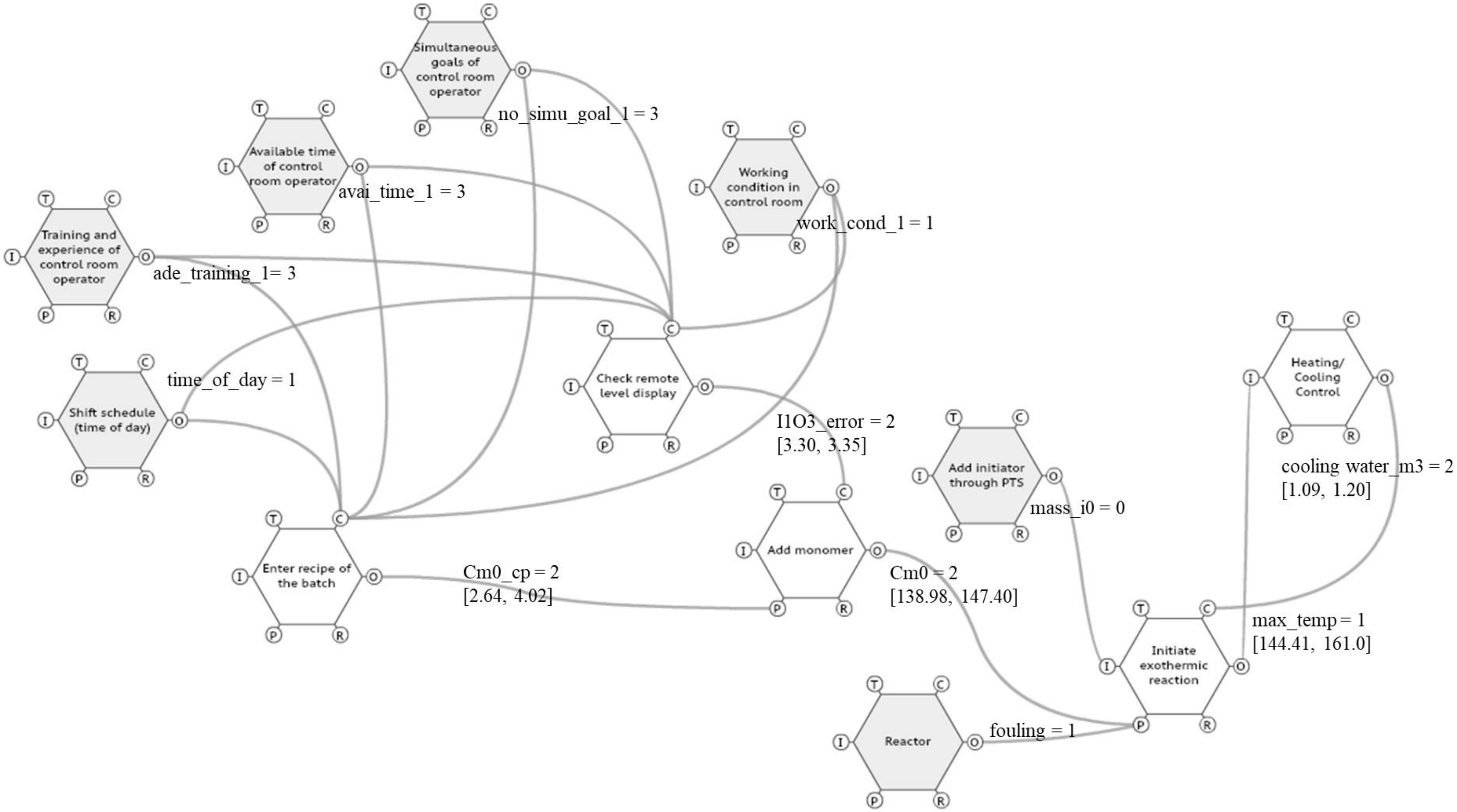

Simultaneous goals of control room operator
no_simu_goal_1 = 3
Available time of control room operator
avai_time_1 = 3
Working condition in control room
work_cond_1 = 1
Training and experience of control room operator
ade_training_1= 3
Shift schedule (time of day)
time_of_day = 1
Check remote level display
I1O3_error = 2
[3.30, 3.35]
Heating/ Cooling Control
cooling water_m3 = 2
[1.09, 1.20]
Add initiator through PTS
mass_i0 = 0
Enter recipe of the batch
Cm0_cp = 2
[2.64, 4.02]
Add monomer
Cm0 = 2
[138.98, 147.40]
Initiate exothermic reaction
max_temp = 1
[144.41, 161.0]
Reactor
fouling = 1
T
C
I
O
P
R

Figure IV-6 Second path leading to the temperature excursion "max_temp =1". Function performances are provided on the connections that are originated from the output (O) aspects. Lift CI which quantifies the couplings between the directly connected functions are shown in brackets. Functions shaded in grey represent the boundary of the path.

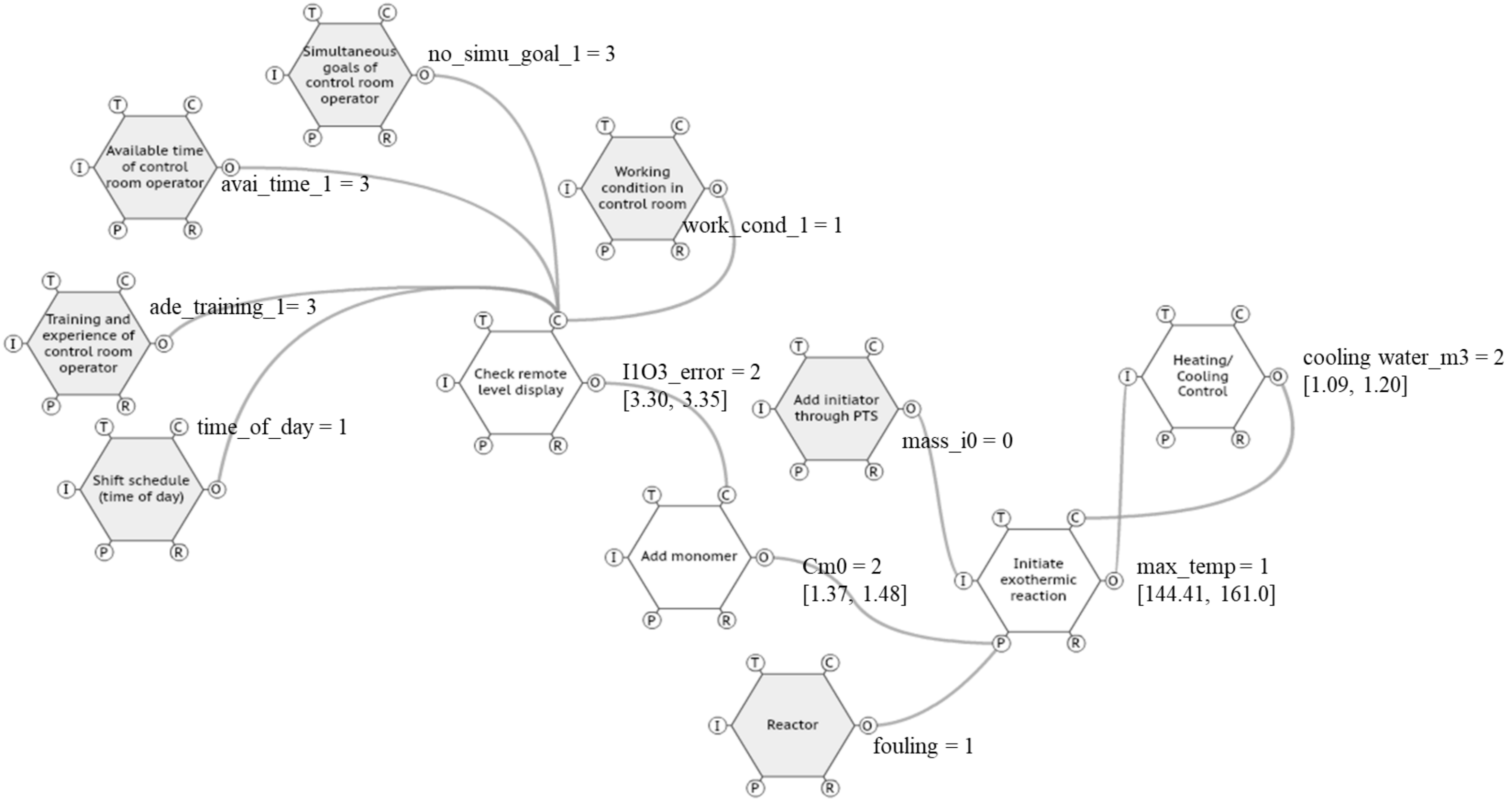

Simultaneous goals of control room operator
no_simu_goal_1 = 3
Available time of control room operator
avai_time_1 = 3
Working condition in control room
work_cond_1 = 1
Training and experience of control room operator
ade_training_1= 3
Shift schedule (time of day)
time_of_day = 1
Check remote level display
I1O3_error = 2
[3.30, 3.35]
Add initiator through PTS
mass_i0 = 0
Heating/ Cooling Control
cooling water_m3 = 2
[1.09, 1.20]
Add monomer
Cm0 = 2
[1.37, 1.48]
Initiate exothermic reaction
max_temp = 1
[144.41, 161.0]
Reactor
fouling = 1
T
C
I
O
P
R

Figure IV-7 Redundant path leading to the temperature excursion “max_temp = 1” with respect to the path in Figure IV-6. Function performances are provided on the connections that are originated from the output (O) aspects. Lift CI which quantifies the couplings between the directly connected functions are shown in brackets. Functions shaded in grey represent the boundary of the path.

4.7 Summary

FRAM allows qualitative analysis of function interactions in a complex system by function decompositions, but it is still difficult to estimate aggregated effects of functions especially when upstream functions impact a downstream function in different ways. With the development of the framework to simulate possible function interactions in Chapter III, function couplings can be quantified through data-driven approaches to identify the interactions leading to potential hazards.

The work in this chapter showed how the lift of association rules quantified upstream-downstream function couplings. Lift CI was derived through the bootstrap process to provide uncertainty information of the quantification process. The comparison between lift CIs and the corresponding theoretical couplings showed strong positive correlations thus indicated lift CI is a quantitative metric of function couplings. More importantly, paths leading to hazard scenarios were identified by merging association rules. Graphical representations of the paths with quantified couplings showed how hazard scenarios emerged thus guided people to take measures to manage undesired outcomes. The work also indicated the need for future work to improve its feasibility in the real world such as finetuning the process of removing redundant rules and paths.

CHAPTER V

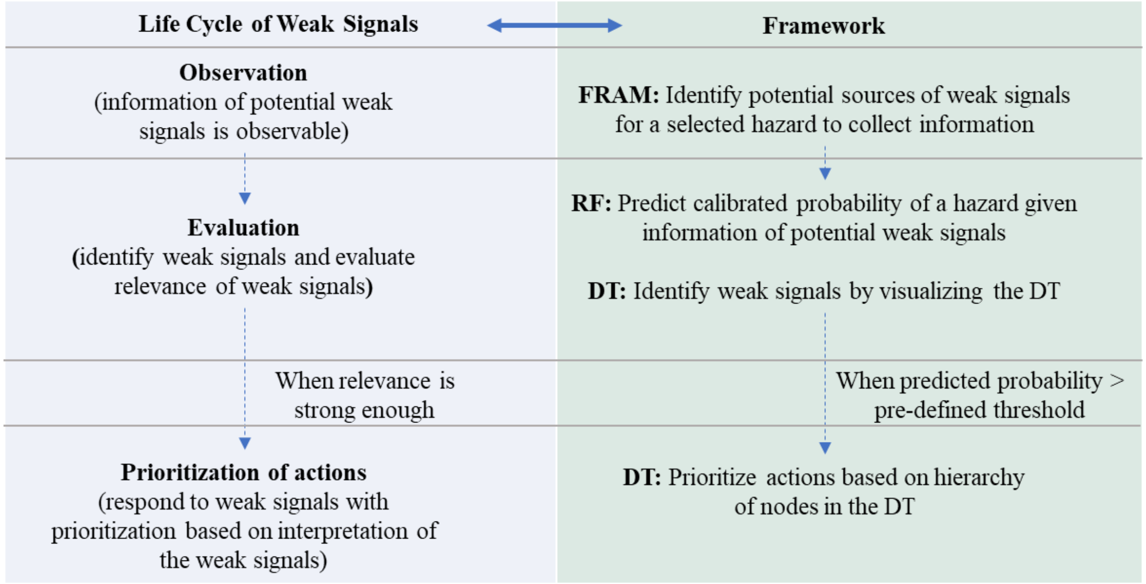
IDENTIFICATION AND RESPONSE TO WEAK SIGNALS

5.1 Introduction

As discussed in subchapter 2.3.2, due to the abundance of information collected in today's digitalized plants, the coexistence of weak signals and noise brings challenges throughout the entire life cycle from observing, evaluating, and responding to weak signals. It is intellectually unmanageable to identify and respond to weak signals only based on individuals' knowledge and experience. The work in this chapter is to develop a novel framework to provide techniques that address challenges during the life cycle.

5.2 Framework Overview

This subchapter provides an overview of the framework. Figure V-1 compares the life cycle of weak signals and the developed framework side by side. The techniques that address challenges during each stage of the life cycle are listed correspondingly. To ensure the information regarding potential sources of weak signals were collected by an organization, the framework is started from utilizing a system-based technique, *i.e.*, FRAM, to identify potential sources of weak signals. FRAM is a graphical representation of a complex system and allows identifying functions and their interactions in the system. With a constructed FRAM for a system, it is straightforward to visualize the upstream functions that set the boundary of the system. Such functions are the ones that could bring the initial performance variabilities to the system. Therefore, their performances are potential weak signals. Information of the potential weak signals can be collected from actual operations or synthesized from simulations. Once performances of these potential sources of weak signals are collected, machine learning techniques are utilized for evaluation. A Random Forest (RF) model is developed and calibrated to estimate the probability of the occurrence of a selected hazard scenario. The estimated probability is a quantitative indication of the relevance of underlying weak signals and provides quantitative information for people to decide whether the weak signals are necessary to respond to. In the cases when the probability exceeds a pre-defined acceptable threshold, it indicates underlying weak signals are relevant enough to call for actions. However, RF is not interpretable. To identify the weak signals contributing to occurrences of a hazardous scenario, a Decision Tree (DT) is further developed by mimicking the knowledge learned by the RF. The hierarchy of nodes in a DT can reflect relative significances of the weak signals, thus the DT is used as a guideline to prioritize responses to the weak signals.



Life Cycle of Weak Signals	Framework
Observation (information of potential weak signals is observable)	**FRAM:** Identify potential sources of weak signals for a selected hazard to collect information
Evaluation **(**identify weak signals and evaluate relevance of weak signals**)**	**RF:** Predict calibrated probability of a hazard given information of potential weak signals **DT:** Identify weak signals by visualizing the DT
When relevance is strong enough	When predicted probability > pre-defined threshold
Prioritization of actions (respond to weak signals with prioritization based on interpretation of the weak signals)	**DT:** Prioritize actions based on hierarchy of nodes in the DT

Figure V-1 Framework corresponding to the life cycle of weak signals

5.3 Identify potential sources of weak signals based on FRAM

Understanding function interactions in a complex system is a prerequisite to systemically identify potential sources of weak signals and to ensure useful information is fed into machine learning techniques. FRAM is a system-based technique of modeling function interactions, such as cause-effect, control of one by another, passing information on, etc., and has been increasingly adopted in industrial environments in recent years (Patriarca *et al.*, 2020). The standard application of FRAM has four steps: 1. Identify and describe functions 2. Identify performance variabilities of functions 3. Aggregate performance variabilities to evaluate function couplings 4. Manage uncontrolled performance variabilities leading to undesired outcomes. As a recent review of FRAM applications shows, which step of FRAM can be up to during an application could vary depending on the user's purposes (Patriarca *et al.*, 2020).

In the context of the study, only the first two steps are needed to decide potential sources of weak signals of a selected hazard. In the first step, technological, human and organization functions that are involved in the system are identified. As Figure II-1 shows, those functions which only have active <u>O</u>utput aspects, are called background functions that define the boundary of a system. These functions could bring initial performance variabilities to the system. The background functions form a preliminary set to identify potential sources of weak signals. The first round of refining the preliminary set is to remove the background functions which are not sources of weak signals based on the proposed definition in subchapter 2.2. Depending on the analysis scope of a system, it is possible to have background functions whose performances are actual events instead of conditions. As mentioned earlier in subchapter 2.2, weak signals are conditions which are early enough to indicate precursor events. Therefore, performances of these background functions are not early enough to be called weak signals and these functions should be excluded from the potential sources of weak signals.

In the second step of FRAM, performance variabilities of functions in the system need to be identified. Given the context of the study, only performance variabilities of background functions need to be identified instead of all the functions in the system. Identifying performance variabilities provides the second round of refinement to finalize potential sources of weak signals. If performance of a background function does not vary, it only provides constant context information and does not have impacts on predicting occurrences of a hazard scenario. Therefore, such functions can be excluded from potential sources of weak signals. After the two rounds of refinement, potential sources of weak signals can be identified, and their information should be collected and utilized for identifying weak signals.

5.4 Random Forest (RF)

Various classification algorithms exist. Performances of algorithms highly depend on nature of data thus no strict procedure exists to decide which is the best classifier before supervisory experimenting and comparing performances on test data. Past studies in Table II-1 covered most classification algorithms. Since an algorithm that performs better than another one in a study can perform worse in a different study, the objective of this study was to explore how a classification algorithm can identify weak signals with promising performances, instead of finding the best classifiers through exhaustive experimenting. The top commonly used algorithms in the domain of safety were the non-parametric classification models: Decision Trees (DT) and ensemble models based on multiple decision trees (ET). DT consists of decision nodes with branches and leaf nodes as the outputs. The number of decision nodes in the longest branch is the depth of the decision tree. DT is popularly applied since it is easily interpretable, which is very critical to understand the underlying factors of incidents. However, it has a few limitations regarding its performance in predicting classes and probabilities. First, DT is unstable meaning that a small variation in the training dataset may result in a different tree. Second, when classes are unbalanced, a typical condition when classifying normal operations versus incidents, DT could overfit as it is highly biased towards a dominant class (Pedregosa *et al.*, 2011). Additionally, DT is grown to derive homogeneous leaves, thus the probability of a class that is observed in a leaf is shifted towards 0 or 1 (Zadrozny & Elkan, 2001). Hence, DT is not good for estimating probabilities. Instead, compared to using a single DT, ET is more robust since classifications are predicted by combining the results of multiple DTs. RF is a typical ensemble model of randomly chosen sub-spaces, hence independent trees using the randomly re-sampled bootstrap bagging strategy. To deal with an extreme imbalance issue, extended versions of RF, *i.e.*, Balanced Random Forest (BRF) and Weighted Random Forest (WRF) have been developed to apply class balancing strategies during the training process (Chen *et al.*, 2004). Additionally, RF improves probability estimation by ensemble and results in a smoother probability estimation (Chawla & Cieslak, 2006). Thus, both BRF and WRF were trained and evaluated in the study for class and probability predictions. BRF was implemented by using the imbalanced-learn toolbox and WRF was implemented by using the scikit-learn library (Lemaître *et al.*, 2017; Pedregosa *et al.*, 2011).

Since BRF and WRF work similarly to a standard RF except for built-in class balancing strategies, Figure V-2 shows a generalized training process of RF and Figure V-3 shows a generalized process of predicting a new instance (Breiman, 2001; Pedregosa *et al.*, 2011). During the training process of a standard RF, a bootstrap data sample is sampled from training data to train each individual DT. An individual DT is trained through the training induction process where the input feature space is recursively partitioned to find the most homogeneous splits. The quality of a node split is determined by an impurity function, which is commonly measured based on Gini Index (Pedregosa *et al.*, 2011). Equation V-1 shows the formula of Gini Index in node m, which contains samples represented by Q_m. The proportion of class k in node m is represented by p_{mk}. A smaller Gini Index indicates a better separation of different classes. Equation V-2 shows the impurity function based on the Gini Index when a feature value θ is used as a split candidate. N_m is the number of samples in node m, and N_m^{left} is the number in the left split generated from node m, and N_m^{right} the same in the right split. $G(Q_m^{left}(\theta))$ is the Gini Index of the left split, and $G(Q_m^{right}(\theta))$ is the Gini Index of the right split. The optimal θ for splitting node m is the one that minimizes the impurity function.

$$G(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \qquad \text{(Equation V-1)}$$

$$G(Q_m, \theta) = \frac{N_m^{left}}{N_m} * G\left(Q_m^{left}(\theta)\right) + \frac{N_m^{right}}{N_m} * G\left(Q_m^{right}(\theta)\right) \qquad \text{(Equation V-2)}$$

During a prediction process, probabilities of classes are estimated at terminal nodes in individual DTs. The probability of a class in a single tree is defined as the fraction of the class in the training samples in its terminal node. Lastly, class probabilities estimated from all the trees are averaged and the class with the highest average probability is voted as the final class.

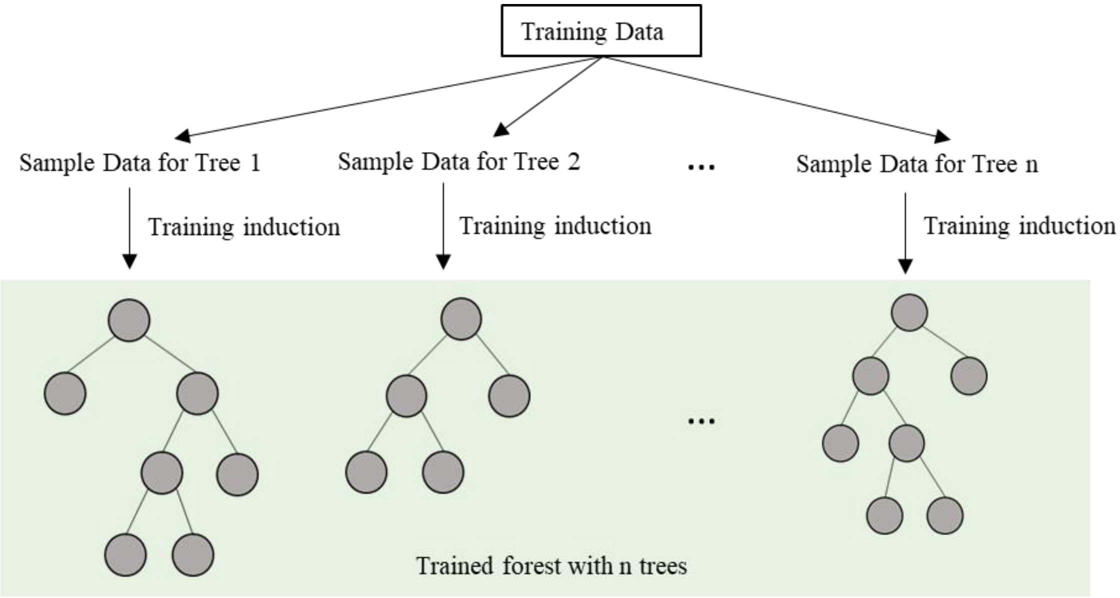


Figure V-2 Training process of a Random Forest with n trees

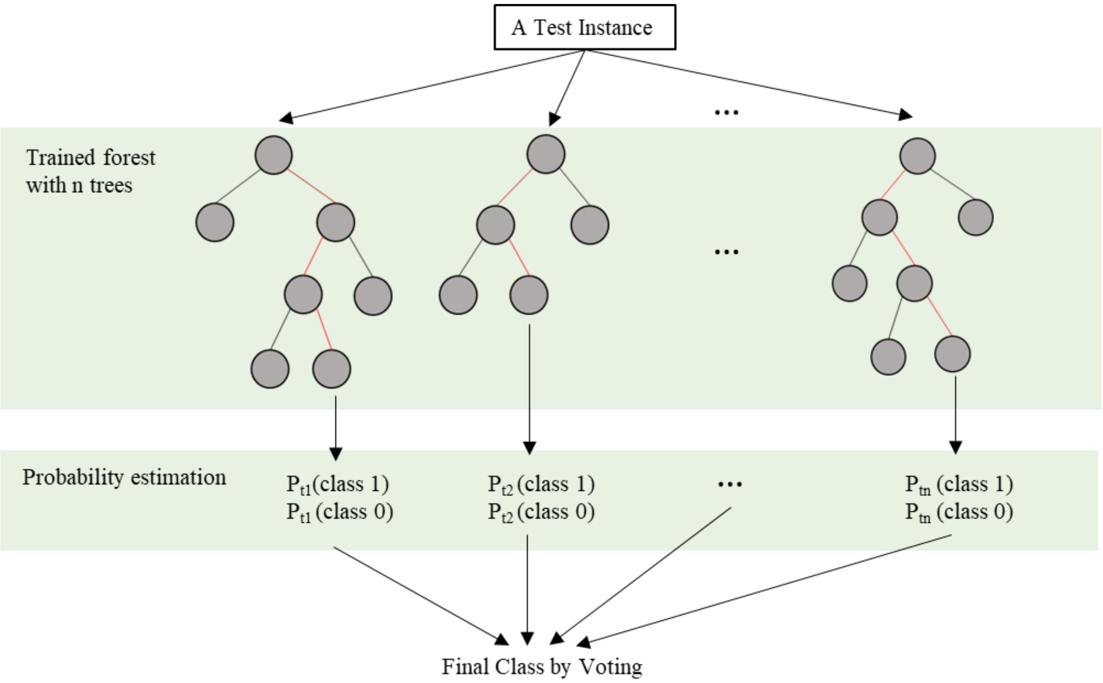


Figure V-3 Process of predicting a test instance by a trained Random Forest with n trees

BRF and WRF take into account balancing strategies at different stages during the training and prediction process. Table V-1 compares the algorithms with a standard RF (Chen *et al.*, 2004; Lemaître *et al.*, 2017; Pedregosa *et al.*, 2011). WRF assigns a higher weight to the minority class to make the weighted number of the samples balanced with the number of majority class samples. The weight is applied to the step of training induction by plugging weighted numbers of samples into the impurity function to find optimal splits. Besides, it is also applied to the step of probability estimation. Regarding BRF, the classes are balanced during the step of sampling data for individual trees. A bootstrap sample of training data is first drawn then samples of the majority class are under-sampled to achieve the class ratio at 1:1. Therefore, training samples for individual trees contain balanced classes. The steps of training induction and prediction work in the same way as the standard RF.

Basic metrics to evaluate classification performance are accuracy, misclassification rate, recall, specificity and precision (Saito & Rehmsmeier, 2015). However, the choice of metric depends on class distributions in datasets and objectives to achieve. Accuracy and misclassification rate are commonly used to evaluate classification performance. They work well when classes in the dataset are approximately balanced. However, when the class distribution is extremely skewed, the metrics have biases toward the majority class and are not appropriate to evaluate predictions on the minority classes. To evaluate predictions for a specific class when the positive class is the minority and the negative class is the majority, specificity measures accurate predictions among the majority class, while recall, which is also called as sensitivity and true positive rate, measures accurate predictions among the minority class (Douzas *et al.*, 2018). Precision measures how many predicted positive classes are true positive classes. According to the context of the study to predict low frequency but high consequence events, false negative prediction is much more costly than false positive ones. Therefore, recall was used as the metric to evaluate the performance of RF.

Table V-1 Comparison between basic RF, Weighed Random Forest (WRF), and Balanced Random Forest (BRF)

	Basic RF	**WRF**	**BRF**
Sample data for individual trees	Draw a bootstrap sample from training data	Draw a bootstrap sample from training data	Draw a bootstrap sample from training data then under-sample the majority class to achieve the class ratio of 1:1
Training induction	Split a node based on the impurity function in Equation V-2	Split a node based on the impurity function in Equation V-2 but the numbers of samples in Equation V-2 are replaced with weighted sums	Split a node based on the impurity function in Equation V-2
Probability estimation	Fraction of classes at a leaf node	Weighted fraction of classes at a leaf node	Fraction of classes at a leaf node

5.5 Probability calibration

Besides class prediction, probability estimation is crucial for making decisions and evaluating conditional risk (Zadrozny & Elkan, 2001; Zhong & Kwok, 2013), especially for those rare events with large uncertainties and asymmetric, hence high costs (Wallace & Dahabreh, 2012). Classification algorithms predict classes based on the ranking of class probabilities. However, it was recognized that the probabilities that are estimated by most classification algorithms are not accurate conditional probabilities (Dal Pozzolo et al., 2015; Niculescu-Mizil & Caruana, 2005; Wallace & Dahabreh, 2012; Zadrozny & Elkan, 2001). The probabilities that are estimated from classification algorithms are based on raw training samples. For example, as the previous section explained, the probability of a class that is estimated from a DT is the fraction of the class within the training samples in its terminal node. Especially with imbalanced classes, balancing techniques are used to improve classification performance by modifying the skewed distribution of the training dataset, which results in a biased probability estimation since training and testing samples are not drawn from the same distribution any more (Wallace & Dahabreh, 2012). Thus, calibration is needed to map the probability values predicted by classifiers to corrected posterior class probabilities. Platt scaling and isotonic regression are two commonly used calibration methods for binary classification problems. Empirical studies showed these calibration methods improve probability estimations of different classification algorithms including RFs (Caruana et al., 2008; Caruana & Niculescu-Mizil, 2006). There is no absolute winner between the two techniques (Caruana et al., 2008). Platt scaling is parametric assuming the calibration function follows a sigmoid-shape, while non-parametric isotonic regression can be more generally applied (Platt, 1999; Zadrozny & Elkan, 2001, 2002). The study used isotonic regression since it outperforms Platt scaling on most problems (Caruana et al., 2008). To avoid overfitting isotonic regression, the dataset for calibration needs to be independent of the dataset for training the classifier, while the data size for calibration needs to be large enough, at least 1000 samples (Caruana *et al.*, 2008; Niculescu-Mizil & Caruana, 2005).

In reality, the true class probabilities are usually unknown, therefore the performance of calibration is evaluated by comparing calibrated probabilities with empirical ones from observations via a reliability curve (Niculescu-Mizil & Caruana, 2005). A hypothetical calibration curve is shown in Figure V-4 for demonstration. The x-axis represents the mean value of calibrated probabilities of a class in a bin, while y-axis represents the fraction of the class in a bin. The calibration curve is derived by first discretizing the calibrated probabilities into bins. The demonstration shows the bins with the bin width = 0.1. The observations with the calibrated probabilities between 0 and 0.1 fall in the first bin, between 0.1 and 0.2 fall in the second bin, and so on. For each bin, the mean value of the calibrated probabilities that are fallen in a single bin is calculated, as well as the fraction of the class among the observations in the bin. The mean value and the fraction in a bin are plotted against each other as a dot. For an ideally calibrated model, all the points are expected to be on the dash diagonal line, meaning the fraction of the class from observations is equal to its calibrated probability from predictions. To evaluate the calibration performance quantitively, coefficient of determination (R^2) was used to describe how well the calibration curve is fitted to the ideal calibration curve.

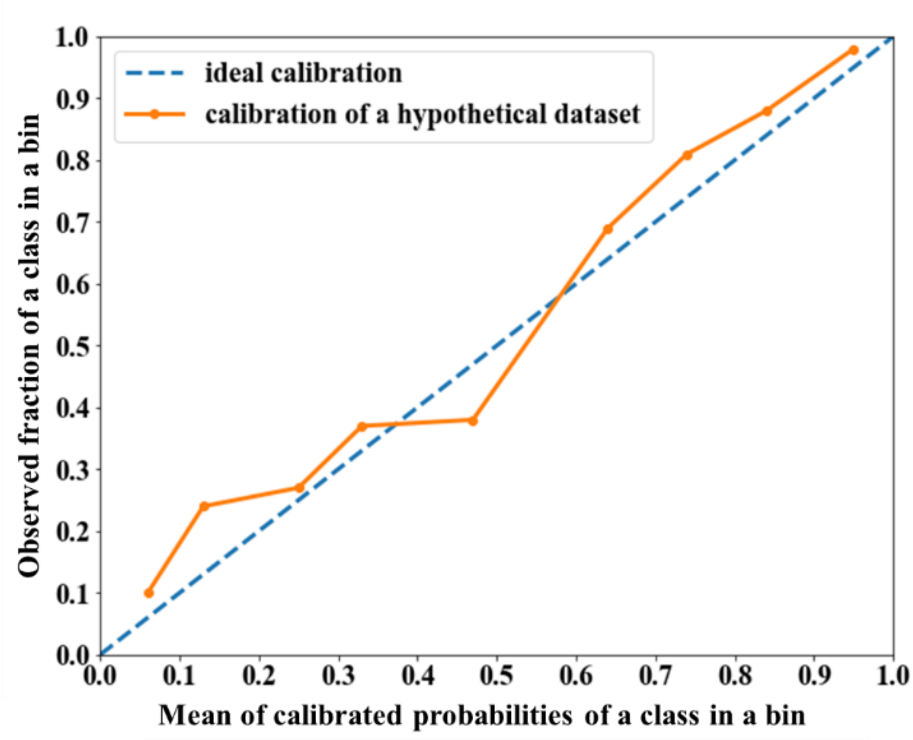


Figure V-4 Demonstration of a hypothetical calibration curve

5.6 Decision Tree (DT) to interpret weak signals

DT is weak for predicting classes and probabilities, but it is an effective method to interpret underlying causations of a prediction. For incident prevention, predicting occurrences of incidents is not enough since people need guidance to identify and interpret the underlying weak signals to take effective measures. DT has been used as a rule extraction tool of black-box models (Augasta & Kathirvalavakumar, 2012; Barakat & Bradley, 2010; Bastani *et al.*, 2017; Han *et al.*, 2014; He *et al.*, 2006), so that powerful predictability of black-box models such as RF can be utilized together with the interpretability of DT. Even though a DT cannot be an explainer to reveal what a black-box model actually does, it can be an approximation of a trained black-box model (Rudin, 2019). The DT that approximates the performance of a black-box model combines the predictability of the black-box model and how a prediction is made. Thus, DT was developed to extract rules from a trained RF model in the study.

There are mainly three types of rule extraction methods using DT, namely pedagogical, decompositional and eclectic approaches (Barakat & Bradley, 2010). A pedagogical approach mimics the learning of a black-box model and is independent of black-box algorithms. It has been extensively utilized to extract rules from RF, NN and SVM. In the pedagogical approach, the classes in the original training data are replaced with the classes predicted by a black-box model. The input features in the original dataset and the predicted classes become an artificial dataset to develop the DT, thus the DT infers the knowledge that has been learned by the black-box model. Studies showed DT successfully maintains similar classification performance of black-box models through mimic-learning (Baesens *et al.*, 2003; Bastani *et al.*, 2017; He *et al.*, 2006). Unlike a pedagogical approach, decompositional and eclectic approaches need partial information from black-box models such as support vectors of a SVM and weights in a NN. Since a pedagogical approach outperformed decompositional and eclectic approaches from multiple aspects such as computational costs and classification performances (Augasta & Kathirvalavakumar, 2012), it was utilized in the study to extract rules from RF. The process of developing the DT are explained in chapter 5.7.

To evaluate the extent to which the rules in a DT mimic the learning of a RF, fidelity was used as the evaluation metric (Barakat & Bradley, 2010; Bastani *et al.*, 2017;

Guidotti *et al.*, 2018). For a classification problem, fidelity is the percentage of the samples whose classifications by the black-box model agree with the classifications by the DT.

Given potential sources of weak signals as input features to develop a DT, branches in the DT unfold the interactions between weak signals that lead to the predictions of incidents. Since DT is a global model which extracts most impacting rules from entire training data (Du *et al.*, 2019; Yang *et al.*, 2018), the weak signals in such a branch could be a subset of weak signals depending on the context under which the DT is developed. The interaction between the subset of weak signals is significant enough to indicate an occurrence of incident. Those input features which are not involved in a branch could be weak signals whose interactions are relatively less important than the ones involved in the branch, or could be noise which has negligible impact on incident predictions in any context. Additionally, relative significances of weak signals can be indicated by the hierarchy of the DT. Typically, the feature at the top node is the most influencing (Bevilacqua *et al.*, 2008; James *et al.*, 2013). The hierarchical structure provides guidance for people to prioritize responses to prevent weak signals from impacting safety performance.

5.7 Development of machine learning models

5.7.1 Development of RF and probability calibration

Figure V-5 shows the flowchart of developing BRF and WRF and calibrating their class probabilities. Given the full dataset for identifying weak signals, the dataset was split into training data and test data. The training data was utilized to train and validate a RF through 3-fold cross-validation. In each iteration out of the three, 2/3 of the training data was used for training and the rest 1/3 was used for validation. To obtain an optimal model, hyperparameters need to be tuned. The hyperparameters of BRF and WRF that were tuned in the study are listed in Table V-3. Before the training process started, ranges of the hyperparameters were specified constituting a hyperparameter space. The strategy of hyperparameter tuning was to randomly search the hyperparameter space. The randomized searching strategy was proved to be more effective to find an optimal model compared to the exhaustive grid search (Bergstra & Bengio, 2012; Mantovani *et al.*, 2015). Every time when a new set of hyperparameters was searched from the space, a corresponding model was trained and validated through the 3-fold cross-validation. After a specified number of searches, the hyperparameters which provided the best classification performance on a hold-out dataset were selected as the optimal hyperparameters. Finally, all training data was used to train the RF with the optimal hyperparameters to derive the optimal model for predicting new instances. As mentioned in subchapter 5.4, the recall was the performance metric to select the optimal model. RandomizedSearchCV in the scikit-learn library was used in the study for the process of hyperparameter tuning and model selection. Classes of the test data were predicted by the optimal model to evaluate the model performance on unseen data.

Table V-2 Hyperparameters in WRF and BRF (Pedregosa *et al.*, 2011)

Hyperparameters	**Description**
n_estimators	Number of trees in the forest
Max_features	Number of features to be considered when looking for the best split
Max_depths	Maximum depth of a tree in the forest
Min_samples_split	Minimum number of samples that is required to split an intermediate node
Min_samples_leaf	Minimum number of samples that is required at a leaf node

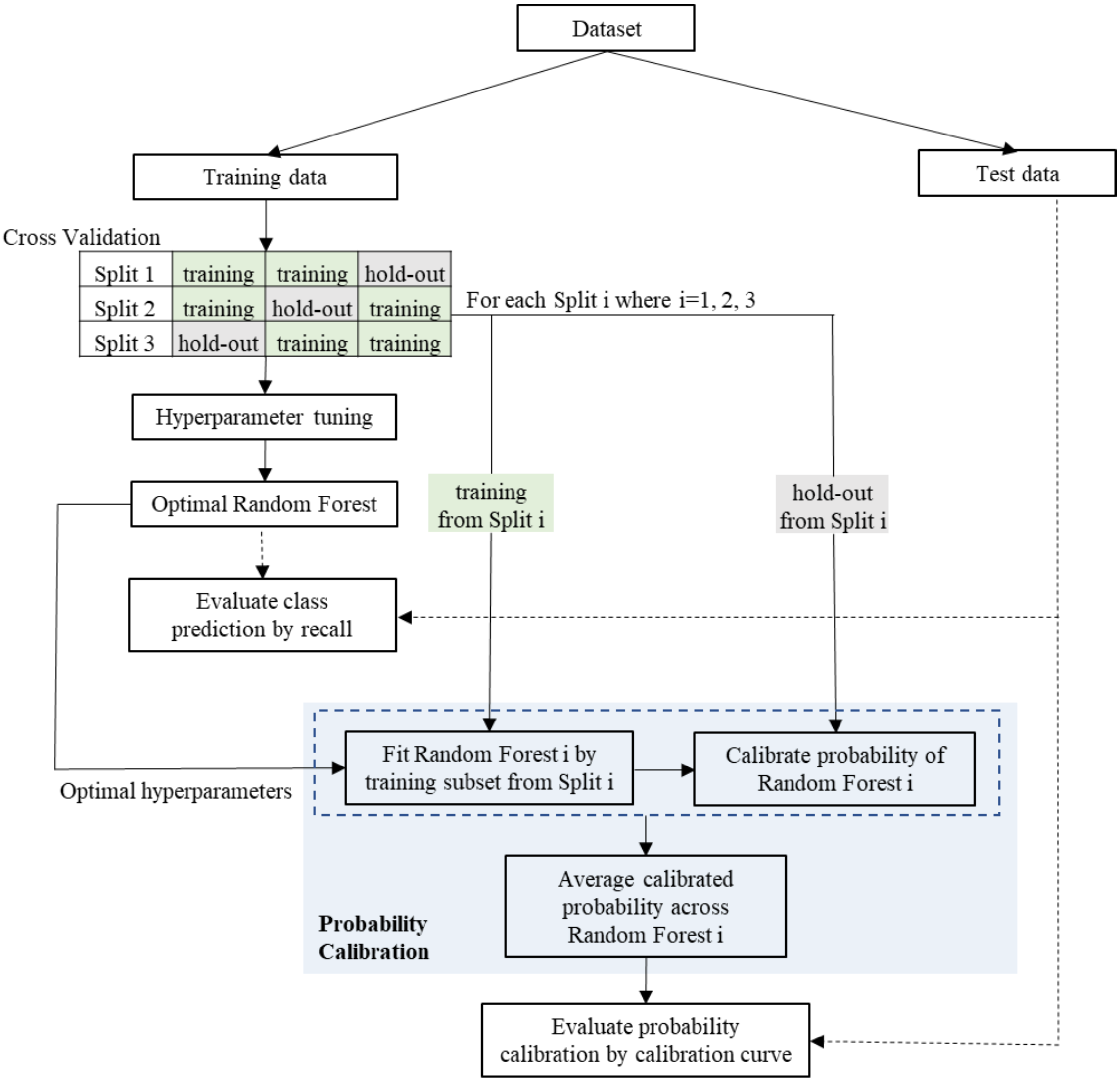

Dataset
Training data
Test data
Cross Validation
Split 1 | training | training | hold-out
Split 2 | training | hold-out | training
Split 3 | hold-out | training | training
For each Split i where i=1, 2, 3
Hyperparameter tuning
Optimal Random Forest
training from Split i
hold-out from Split i
Evaluate class prediction by recall
Fit Random Forest i by training subset from Split i
Calibrate probability of Random Forest i
Optimal hyperparameters
Average calibrated probability across Random Forest i
Probability Calibration
Evaluate probability calibration by calibration curve

Figure V-5 Training and evaluation process of a Random Forest and probability calibration

For probability calibration, a dataset that is independent of the training data is needed. However, in the case when there is only a limited number of samples, holding out a large number of samples only for the calibration reduces the size of training data and may impact classification performance. An alternative approach to utilize the training data more efficiently is to utilize k-fold cross-validation. During each iteration, all the training samples except the hold-out dataset are used to train the classifier, then the hold-out dataset can be utilized as an independent dataset to calibrate the probability (Boström, 2008; Pedregosa *et al.*, 2011). Since all the training data had already been used to develop the optimal RF, instead of using the model directly, only the hyperparameters of the optimal RF were used to further obtain calibrated probabilities. Same as the cross-validation for training the optimal RF, training samples were split into 3 folds. For each split i, where i=1,2,3, the two folds for training fitted a RF i with the optimal hyperparameters, and the hold-out fold was used to calibrate probabilities that had been predicted by the RF i. The final calibrated probability of an instance was the average of calibrated probabilities that were predicted by the three RFs. The process of probability calibration was implemented through CalibratedClassifierCV in the scikit-learn library (Pedregosa *et al.*, 2011). Lastly, the test data was used again to evaluate the calibration performance through the coefficient of determination (R^2) between the actual calibration curve and the ideal one.

5.7.2 Development of DT to extract rules from a trained RF

In order to develop a DT to approximate the prediction performance of a trained RF, the DT needs to be trained and evaluated with an artificial training dataset whose input features are the same as the original training data, but the classes are predictions by the trained RF. Thus, the training data that had been used for developing the RF was kept the same for developing the DT. As Figure V-6 shows, the same training dataset was first treated as an unlabeled dataset and their classes were predicted by the trained RF. The actual classes in the training data were replaced by the predicted classes to become the artificial training dataset for developing the DT. Following a similar method of training and validating as the RF, the process of hyperparameter tuning and model selection was conducted by the randomized search and 3-fold cross-validation using RandomizedSearchCV in the scikit-learn library. The hyperparameters to be tuned for a DT were the same as the hyperparameters in Table V-3 except that “n_estimators” was not applicable since DT only contains a single tree. To evaluate the performance of the DT that approximated the trained RF, fidelity was calculated by comparing the predicted classes by the DT with those by the trained RF.

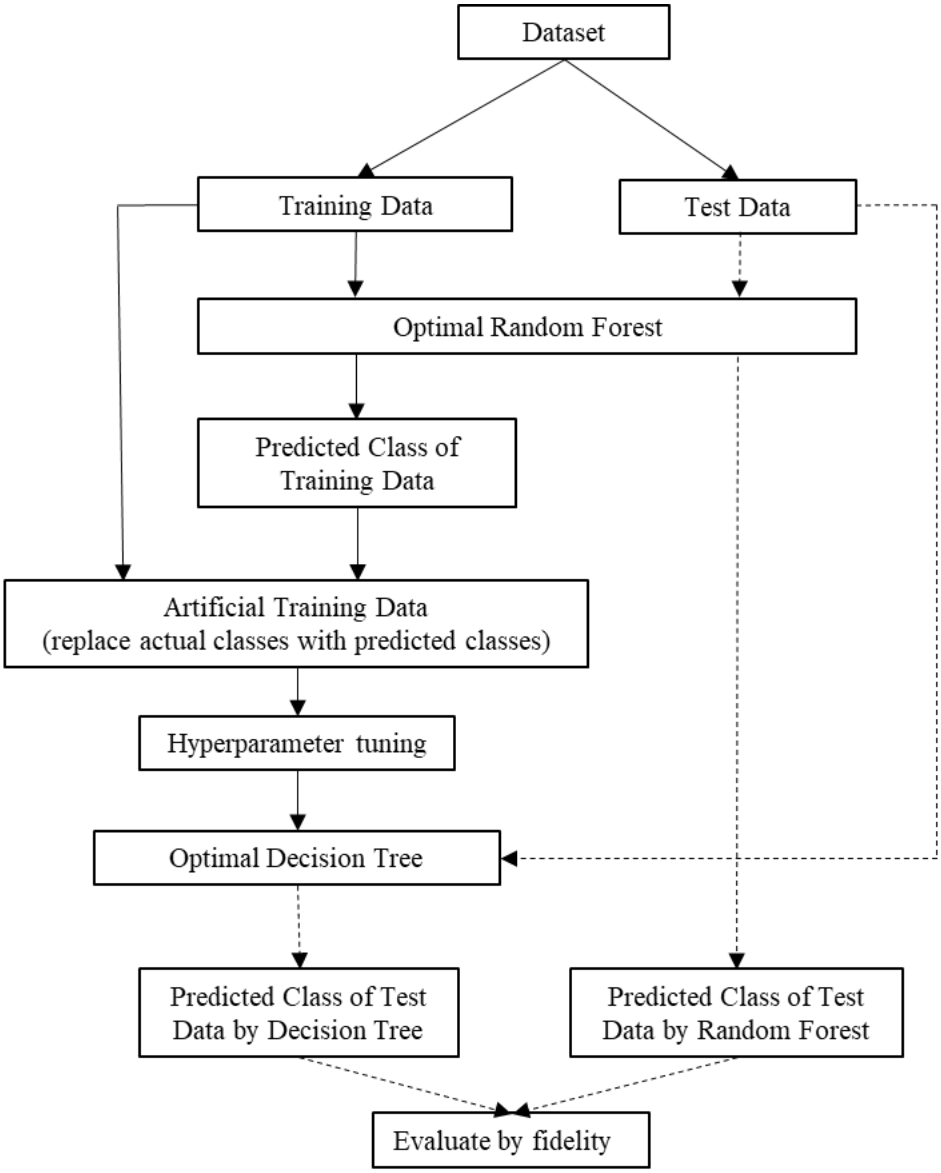

Dataset
Training Data
Test Data
Optimal Random Forest
Predicted Class of Training Data
Artificial Training Data (replace actual classes with predicted classes)
Hyperparameter tuning
Optimal Decision Tree
Predicted Class of Test Data by Decision Tree
Predicted Class of Test Data by Random Forest
Evaluate by fidelity

Figure V-6 Training and evaluation process of a Decision Tree to extract rules from a trained Random Forest

5.8 Case Study: Batch Polymerization Process

The batch polymerization process studied in previous chapters was used as the case study for the work. Identification of weak signals that could lead to potential temperature excursions was demonstrated, and corrective actions for responses were suggested to illustrate the application of the framework.

5.8.1 Potential sources of weak signals

The system of the case study was analyzed by FRAM to identify the involved functions and their performance variabilities. Details of the FRAM analysis were provided in subchapter 3.7.1. As explained in subchapter 2.2, potential sources of weak signals are the functions that have various performances and at the boundary of the FRAM. Figure III-13 shows the simplified FRAM after the functions with invariant performance were removed. Red circles indicate where an upstream function had been removed. The shaded functions were at the boundary of the FRAM thus they were identified as the potential sources of weak signals. The information of their performances was collected to ensure actual weak signals were observable. The potential sources of weak signals and their possible performance variabilities were extracted from the FRAM analysis in subchapter 3.7.1 and provided in Table V-3.

Table V-3 Potential sources of weak signals

Potential Sources of Weak Signals	**Performance Variabilities**
Available time of field operator	Adequate, Temporarily inadequate, Continuously inadequate
Available time of control room operator	Adequate, Temporarily inadequate, Continuously inadequate
Shift Schedule (time of day)	Day-Time, Night-time
Simultaneous goals of control room operator	Fewer than capacity, Matching current capacity, More than capacity

Table V-3 Continued

Potential Sources of Weak Signals	**Performance Variabilities**
Simultaneous goals on field	Fewer than capacity, Matching current capacity, More than capacity
Training and experience of control room operator	Adequate training and high experience, Adequate training and low experience, Inadequate training
Training and experience of field operator	Adequate training and high experience, Adequate training and low experience, Inadequate training
Working condition in control room	Advantageous, Compatible, Incompatible
Working condition on field	Advantageous, Compatible, Incompatible
Utility pump A is maintained	“PMs being carried out as optimal PM schedule”, “no PM being carried out”
Reactor	Reactor is under clean condition, Reactor has fouling which reduces overall heat transfer coefficient by 20%

5.8.2 Data Description

To identify weak signals that are related to potential temperature excursions in the batch process, information of the potential sources of weak signals was collected as input features for machine learning models. Data of the reactant temperature, which is the performance of the function “Initiate exothermic reaction”, was collected to describe the target classes. The data that had been synthesized through the hybrid simulator in subchapter 3.7.3 was processed for the analysis in this chapter. First, since the reactant temperature was collected as time-series data in a batch. The maximum reactant temperature was extracted from batch reactions to assign the target classes. Additionally, instead of directly predicting hazardous temperature excursions, a broader class "temperature deviation" was predicted. The rationale was the way temperature deviations occur is similar to that of temperature excursions, but occurrences of former are more frequent, so that for classification modeling a smaller dataset suffices. Lastly, strong signals may exist in the potential sources of weak signals, which needed to be excluded from the input features. A strong signal is a performance that can indicate occurrences of temperature deviations by itself instead of by interacting with others. If such strong signals were used as input features to develop a classification model, the classification model would treat the strong signals as important features while impacts of weak signals would be overshadowed. As Table V-3 shows, the function "Reactor" is a potential source of weak signals, but its performance "Reactor has fouling which reduces overall heat transfer coefficient by 20%" is a strong signal. According to the domain knowledge, it is well recognized that fouling is a typical fault that results in temperature deviations (Chylla & Haase, 1993; Crowley & Choi, 1996; Kim *et al.*, 1993; Wieme *et al.*, 2007). Thus, the data of the batches under the reactor fouling conditions was excluded for developing classification models.

The final data for developing classification models was from 438,000 batch operations where the reactor was under a clean condition. Data of a batch was represented as a row/sample in the dataset. The samples of no temperature deviation and of temperature deviations were at the ratio of 32.6:1. The dataset was split into two datasets – 21,900 samples for training and 416,100 samples for testing. The input features and classes are described in Table V-4. The classes regarding temperature deviations were assigned based on the maximum reactant temperature in a batch. Since no process noise was considered during the simulation, class 1 (temperature deviation occurs) was assigned when the maximum reactant temperature of a batch was not exactly the same as that of a batch under a normal operation. Otherwise, class 0 (no temperature deviation occurs) was assigned.

Table V-4 Input features and classes that were used to develop a Random Forest and a Decision Tree

Input Feature (Variable)	Values
Available time of control room operator (avai_time_1)	1: Adequate 2: Temporarily inadequate 3: Continuously inadequate
Available time of field operator (avai_time_2)	1: Adequate 2: Temporarily inadequate 3: Continuously inadequate
Shift Schedule (time_of_day)	1: Day-time 2: Night-time
Simultaneous goals of control room operator (no_simu_goal_1)	1: Fewer than capacity 2: Matching current capacity 3: More than capacity
Simultaneous goals of field operator (no_simu_goal_2)	1: Fewer than capacity 2: Matching current capacity 3: More than capacity
Training and experience of control room operator (ade_training_1)	1: Adequate training and high experience 2: Adequate training and low experience 3: Inadequate training
Training and experience of field operator (ade_training_2)	1: Adequate training and high experience 2: Adequate training and low experience 3: Inadequate training
Working condition in control room (work_cond_1)	1: Advantageous 2: Compatible 3: Incompatible

Table V-4 Continued

Input Feature (Variable)	Values
Working condition on field (work_cond_2)	1: Advantageous 2: Compatible 3: Incompatible
Maintenance of Utility Pump A (pm)	0: no pm carried out 1: pm was carried out as scheduled
Classes	**Values**
no temperature deviation occurs	0
temperature deviation occurs	1

5.8.3 Classification

BRF and WRF for predicting occurrences of temperature deviations were trained and evaluated. Hyperparameters were tuned through the randomized search with 3-fold cross-validations. Ranges of the hyperparameters for the randomized search were provided in Table V-5. The optimal set of the hyperparameters was derived from 200 searches within the feature space.

Table V-5 Range of hyperparameters that were tuned for a RF

Hyperparameters	Ranges
n_estimators	Integers in [100,500]
Max_features	[3, 5, 7, 9]
Max_depths	Integers in [3,11]
Min_samples split	26 evenly spaced numbers over [102,202]
Min_samples_leaf	26 evenly spaced numbers over [50,100]

Due to randomness during a training process such as random searching the hyperparameter space and random selection of features to find best split nodes in an individual tree, repeating the same training process multiple times could lead to various performances of BRF and WRF. To compare performances of BRF and WRF on predicting temperature deviations, the training process was repeated three times for both BRF and WRF. Table V-6 shows the three trained BRFs and WRFs, with corresponding hyperparameters. Their performances on the test data were provided in terms of recall and precision of class 1. According to the table, BRFs have higher recalls than WRFs indicating BRFs predicted class 1 cases more successfully. However, precisions of class 1 of all the six models are 5%, indicating only 5% of predicted class 1 are true class 1, and the rest are false positive alarms. The low precision was expected since predictions were made based on weak signals, whose characteristic is predicting early but always not precisely enough. Due to the characteristic of weak signals, class predictions are not reliable enough. Probability estimation is needed to tell how likely a temperature deviation would occur and inform the relevance of the underlying weak signals.

Table V-6 Performances of Balanced Random Forests (BRFs) and Weighted Random Forests (WRFs)

	Recall of Class 1	**Precision of Class**	**n_estimators**	**Max_features**	**Max_depths**	**Min_samples_split**	**Min_samples_leaf**
BRF 1	0.83	0.05	367	9	7	114	92
BRF 2	0.85	0.05	459	9	6	166	98
BRF 3*	0.85*	0.05	297	9	4	118	100
WRF 1	0.78	0.05	440	7	3	102	88
WRF 2	0.79	0.05	273	9	3	138	54
WRF 3	0.78	0.05	472	9	3	158	76

* The model used for further analysis

5.8.4 Probability calibration

Since all BRFs performed similarly and better than WRFs, any BRF model of the three could be used for further probability estimation. In the study, BRF 3 was used. Figure V-7 shows the calibration curve to compare the mean of calibrated probabilities of class 1 in bins with the empirical probabilities which were the observed fraction of class 1 in the bins. The performance was evaluated using the test data. The bin width was 0.02 which was determined based on the number of samples in bins to ensure there were not too few samples for estimating an empirical probability in a bin. Correspondingly, the histogram in Figure V-8 shows the number of samples in the same bins. Figure V-7 shows that the calibrated probabilities were near the ideal calibration with $R^2 = 0.96$. The calibrated probabilities deviated from the ideal calibration in those bins which corresponded to larger probabilities. Correspondingly, Figure V-8 shows the bins for the larger probabilities contained much fewer samples. Since the instances with higher probabilities of class 1 were rarer than those with lower probabilities, the calibrated probabilities based on observed fractions can be less precise.

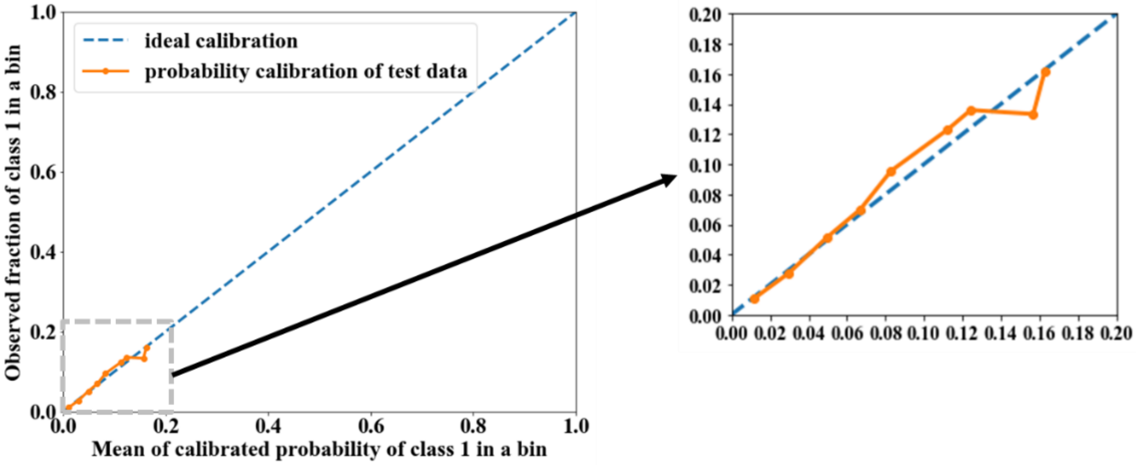


Figure V-7 Calibration curve of the model BRF 3

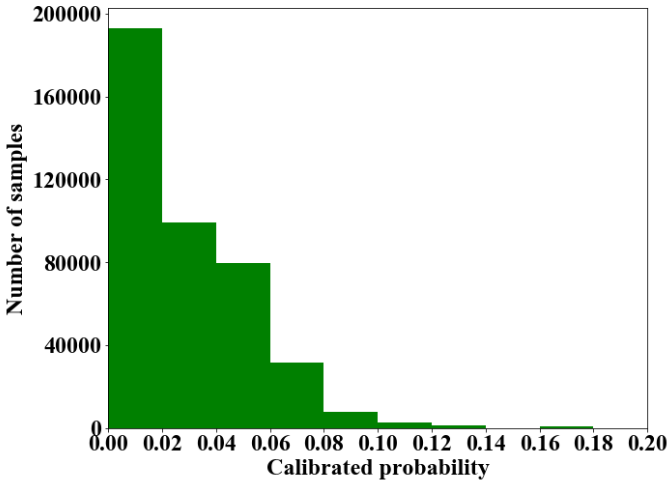


Figure V-8 Number of samples within the bins for plotting the calibration curve

5.8.5 Identification of weak signals via DT

DT was developed to approximate the performance of BRF 3 then identify actual weak signals leading to potential temperature deviations. The training dataset for developing the DT was the training dataset that had been used for developing the BRF, whose original classes were replaced with the predicted classes by the BRF. The ratio of class 0 to class 1 in the artificial training dataset was 1:1.2, which is nearly balanced, therefore the performance metric for tuning hyperparameters of DT was changed from recall of class 1 to accuracy, which is the same as fidelity. Ranges of the hyperparameters for the randomized search are provided in Table V-7 and the optimal hyperparameters were derived from 200 searches. Since the same training process may lead to different DTs due to randomness in the training process, DTs were developed by repeating the same training process three times to understand how different DTs would affect identifying and responding to weak signals.

Table V-7 Range of hyperparameters that were tuned for a DT

Hyperparameters	**Ranges**
Max_features	[3, 5, 7, 9]
Max_depths	Integers in [3,11]
Min_samples_split	26 evenly spaced numbers over [102,202]
Min_samples_leaf	26 evenly spaced numbers over [50,100]

Developed DTs are listed in Table V-8. The training process resulted in DTs with consistent results. All the DTs had the same performance with 99.91% fidelity on the test data, indicating the DTs approximated the performance of the BRF very well. The three DTs had two different structures as shown in Figure V-9. However, the structures contained redundant intermediate nodes whose splits led to the same classes thus such nodes were pruned. Figure V-10 shows the pruned tree structures. Given the same/similar model performance, a smaller tree is desirable since it provides more compact descriptions about how a prediction is made (Mehta *et al.*, 1995; Narodytska *et al.*, 2018) thus could guide people to find more effective responses. Therefore, the DT in Figure V-10 b was used as the DT to identify actual weak signals. A detailed investigation regarding how the two different tree structures affected identifying and responding to weak signals will be discussed in the next section. In the DT in Figure V-10 b, the blue paths led to predicted class 1 and showed interactions between actual weak signals that indicate potential temperature deviations. Additionally, since the strong signal “Reactor is under fouling condition” had been excluded to avoid overshadowing impacts of weak signals, the DT was developed under the context when “Reactor is under clean condition”. “Reactor is under clean condition” was an additional weak signal which interacted with the performance variabilities in each path. For example, the top of the DT indicates occurrences of temperature deviation were predicted when “avai_time_1 <= 2.5” was false (i.e., “Available time of control room operator is continuously inadequate”). The performance was relatively strong under the context of “Reactor is under clean condition”, but “Available time of control room operator is continuously inadequate” and “Reactor is under clean condition” were together considered as weak signals since temperature deviations only could be predicted when they interacted. According to the DT, two groups of weak signals leading to the predicted temperature deviation were identified in the case study:

Group 1: Available time of control room operator is continuously inadequate (*i.e.*, “avai_time_1” <=2.5 is false), Reactor is under clean condition

Group 2: Available time of control room operator is adequate or temporarily inadequate *(i.e.*, “avai_time_1” <=2.5 is True), Available time of field operator is continuously inadequate (*i.e.*, “avai_time_2” <=2.5 is false), Reactor is under clean condition

Table V-8 Performances of DTs

	Fidelity	**Max_features**	**Max_depths**	**Min_samples_split**	**Min_samples_leaf**
DT 1	99.91%	7	7	166	62
DT 2	99.91%	9	9	174	76
DT 3	99.91%	9	5	154	68

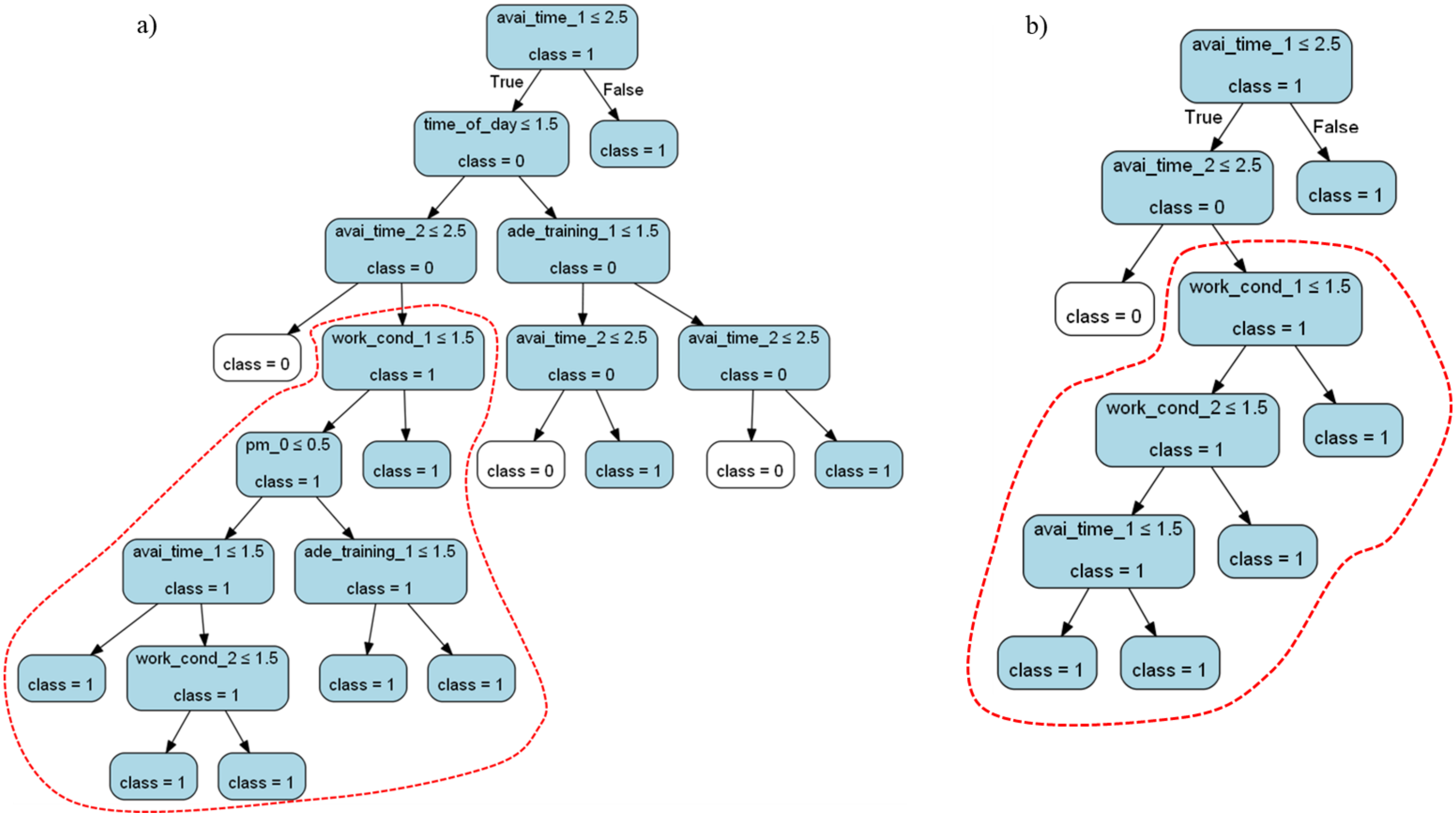


Figure V-9 DTs that are developed from the same training process. Figure a is the structure of DT1 in Table V-8. Figure b is the structure of DT2 and DT3 in Table V-8. Nodes in red circles are redundant nodes whose splits lead to the same classes

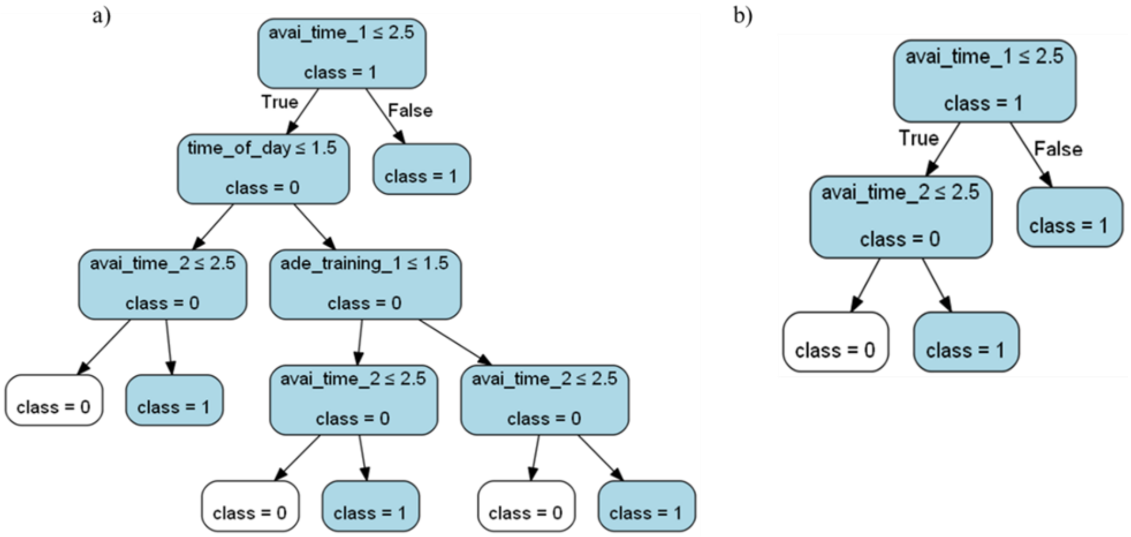
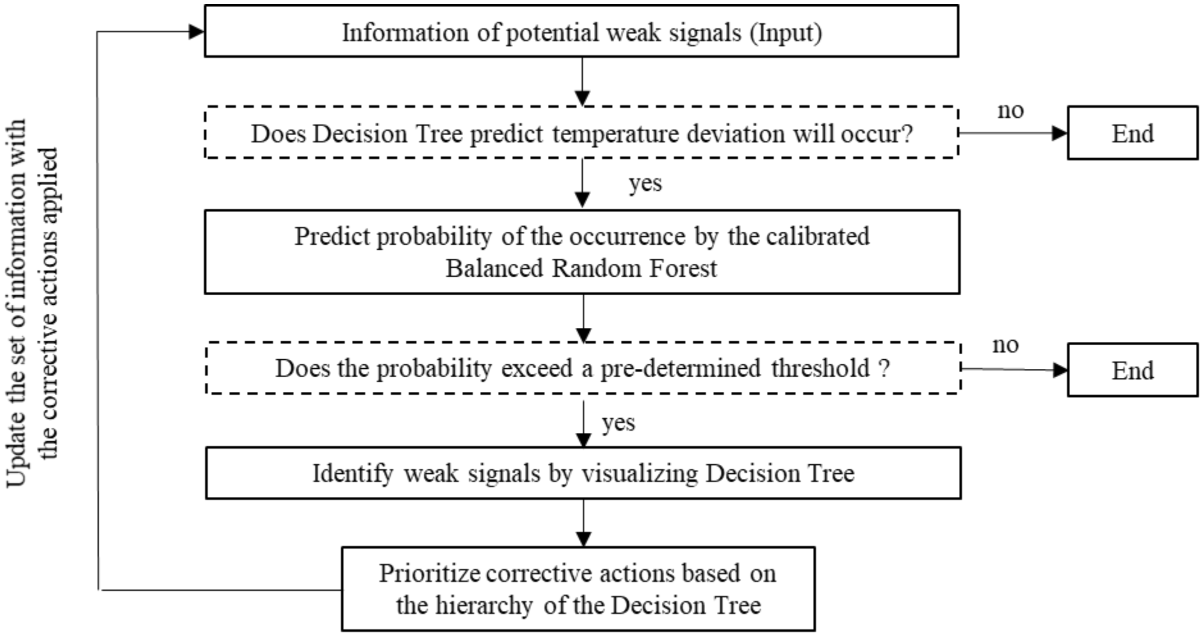


Figure V-10 DTs after pruning redundant nodes. Figure a is the pruned structure of DT1 in Table V-8. Figure b is the pruned structure of DT2 and DT3 in Table V-8.

5.8.6 Demonstration of the framework using a test instance

The chapter is to utilize a test instance for demonstrating how the BRF and the DT are integrated together, as well as for investigating how different DT structures affect the decision-making process. The flowchart in Figure V-11 summarizes the steps of the framework application. Once the information of potential weak signals is collected, it becomes the input information of the developed models. First, the DT is used to predict the occurrence of the temperature deviation to alert users about the existence of weak signals. If the prediction is that temperature deviation occurs, the probability of the occurrence is further predicted by the BRF to evaluate the relevance of underlying weak signals. If the probability exceeds a pre-determined threshold, weak signals need to be identified to take corrective actions. The underlying weak signals can be identified by visualizing the DT and corrective actions can be made and prioritized based on the hierarchy of the DT. As explained in subchapter 5.6, the weak signals that are identified could be a subset of weak signals whose interactions are significant enough to predict the occurrence of the temperature deviation. After corrective actions are made, the input information can be updated and the process is repeated till the temperature deviation is not predicted or the probability of the temperature deviation does not exceed the pre-determined threshold.



Information of potential weak signals (Input)
Does Decision Tree predict temperature deviation will occur?
no
End
yes
Predict probability of the occurrence by the calibrated Balanced Random Forest
Does the probability exceed a pre-determined threshold ?
no
End
yes
Identify weak signals by visualizing Decision Tree
Prioritize corrective actions based on the hierarchy of the Decision Tree
Update the set of information with the corrective actions applied

Figure V-11 Flowchart of applying the framework for decision-making

Table V-9 shows an example test instance. The collected information of potential weak signals is listed as input features, and the corresponding values and descriptions are also listed.

Table V-9 Input features of a test instance for demonstration

Input features	**Value**	**Description**
work_cond_1	3	Working condition in control room is incompatible
no_simu_goal_1	3	Simultaneous goals of control room operator are more than capacity
avai_time_1	3	Available time of control room operator is continuously inadequate
time_of_day	2	Night-time
ade_training_1	3	Training and experience of control room operator is inadequate
work_cond_2	2	Working condition on field is compatible
no_simu_goal_2	2	Simultaneous goals of control room operator match current capacity
avai_time_2	3	Available time of field operator is continuously inadequate
ade_training_2	3	Training and experience of field operator is inadequate
pm	0	No Preventive Maintenance of utility pump is carried out

The framework is demonstrated based on the DT that was shown in Figure V-10 b. The weak signals that were identified from the DT and the corresponding probability of the temperature deviation for the test instance are summarized in Table V-10 as “Iteration 0”. According to the DT in Figure V-10 b, corrective actions can be made through multiple iterations till the probability is reduced to a pre-determined threshold. Table V-10 summarizes two iterations of the corrective actions. The column “corrective actions” shows the corrective actions that can be made, relative to the original test instance. Details of utilizing the framework for the instance is described below.

- The prediction by the DT is “temperature deviation occurs” (Class 1).
- The probability of the temperature deviation predicted by the BRF is 0.163, indicating relatively high relevance between the underlying weak signals and the prediction.
- Figure V-12 shows the path in red with the label “Iteration 0”. As the path shows, the weak signals of the temperature deviation for the specific test instance are “avai_time_1 =3” (*i.e.,* “Available time of control room operator is continuously inadequate”), as well as “Reactor is under clean condition” which is involved in the interaction as a context.
- A pre-determined threshold is not given in the study since it needs a detailed risk assessment. Assuming the probability exceeds a pre-determined threshold, corrective actions can be made to reduce the probability till it becomes lower than the threshold.
- As the path shows, “avai_time_1” is at the top of the DT indicating it is the most impacting feature. It is at a poor performance level which satisfies the condition of the right split of the node and leads to the terminal node “class = 1”. The first iteration of

corrective actions is to improve the performance level of “avai_time_1” from level 3 to 2, which significantly reduces the probability to 0.038. The corresponding decision path after the corrective action is made is shown in yellow in Figure V-12 and labeled as “Iteration 1”. The weak signals corresponding to the prediction “class 1” are “avai_time_1 =2”, “avai_time_2 = 3” and “Reactor is under clean condition”.

- To further reduce the probability, the next corrective actions can be made through improving “avai_time_2” since it is the second top node in the DT. In the second iteration, besides the corrective actions that has been made to “avai_time_1”, “avai_time_2” is improved from level 3 to level 2. The corresponding path after the corrective action is made is shown in green and is labeled as “Iteration 2” in the figure. The prediction is class 0, which ends the further iteration of the flow chart in Figure V-12.
- To provide more insights on the prediction of class 0, the probability of temperature deviation is estimated and is as low as 0.0142. The weak signals leading to the prediction of class 0 are “avai_time_1 =2”, “avai_time_2 =2”, and “Reactor is under clean condition”.
- In case when an organization still needs to reduce the probability even though class 0 is predicted at the iteration 2, RF and DT models can be developed under the context when “Reactor is under clean condition”, “avai_time_1 ⩽ 2.5”, and “avai_time_2 ⩽ 2.5” thus the weak signals which have been overshadowed in the current context can be further revealed and acted upon.

Similarly, the same flowchart is walked through using the DT in Figure V-10 a to investigate how different DTs impact the identification of weak signals and responses. Table V-11 shows iterations of corrective actions, along with corresponding predicted class, probability, and weak signals of temperature deviations. Corresponding paths are colored in Figure V-13.

Table V-10 Iterations of corrective actions based on the decision tree in Figure V-10 b, with corresponding predicted class, probability, and weak signals of temperature deviations (class 1)

Iteration	Corrective actions	Predicted class	Probability	Weak signals
0	NA	1	0.163	avai_time_1 =3, reactor is under clean condition
1	avai_time_1 improved from level 3 to 2	1	0.038	avai_time_1 =2, avai_time_2= 3, reactor is under clean condition
2	avai_time_1 improve from level 3 to 2 avai_time_2 improve from level 3 to 2	0	0.0142	avai_time_1 =2, avai_time_2 =2, reactor is under clean condition

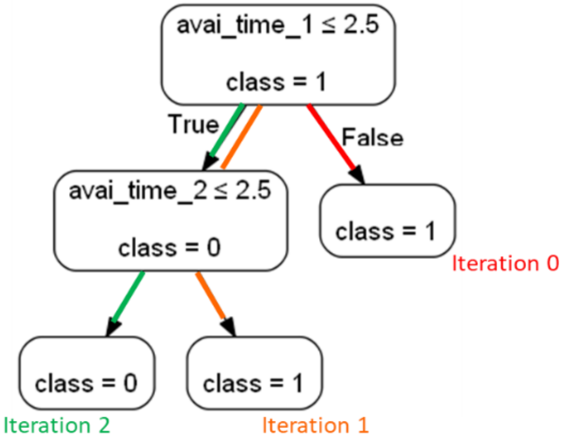


Figure V-12 DT in Figure V-10 b with prediction paths corresponding to iterative corrective actions

Table V-11 Iterations of corrective actions based on the decision tree in Figure V-10 a, with corresponding predicted class, probability, and weak signals of temperature deviations (class 1)

Iteration	Corrective actions	Predicted class	Probability	Weak signals
0	NA	1	0.163	avai_time_1 =3, reactor is under clean condition
1	avai_time_1 improved from level 3 to 2	1	0.038	avai_time_1 =2, time_of_day = 2, ade_training_1 =3, avai_time_2= 3, reactor is under clean condition
2	avai_time_1 improve from level 3 to 2 time_of_day improve from level 2 to 1	1	0.030	avai_time_1 =2, time_of_day = 1, avai_time_2 =3, reactor is under clean condition
3	avai_time_1 improve from level 3 to 2 time_of_day improve from level 2 to 1 avai_time_2 improve from level 3 to 2	0	0.0138	avai_time_1 =2, time_of_day = 1, avai_time_2 =2, reactor is under clean condition

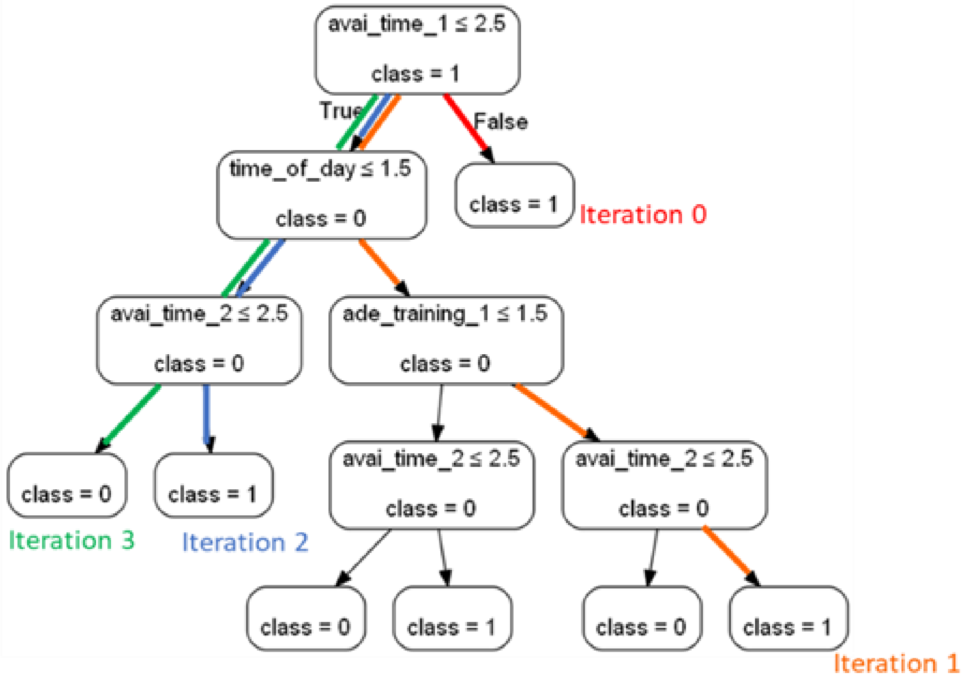


Figure V-13 Decision tree in Figure V-10 a with prediction paths of iterative corrective actions.

Compared last iterations of both DTs, the probabilities of temperature deviations are reduced to the approximately same level after iterative corrective actions. However, based on the hierarchy of the larger decision tree, an additional iteration (*i.e.*, Iteration 2 in Table V-11) to improve the performance level of "time_of_day" is indicated but the iteration only reduces the probability by 0.008 compared to its previous iteration. In the larger decision tree, the impact of "time_of_day" is actually less critical than "avai_time_2", which is at the same depth of the smaller decision tree. The difference between the tree structures is due to the limitation of the DT training process which has unstable node selection. For example, one of the hyperparameters that was tuned is “max_feature”, which determines the number of features to be considered during each node splitting. The node split candidates were randomly selected from all input features based on “max_feature” to find the one which could separate the classes best, thus different node split candidates could lead to different tree structures even though the same training process was used. The comparison between the two DTs agrees with the statement that a smaller tree is desirable when performances of trees are similar (Mehta *et al.*, 1995; Narodytska *et al.*, 2018). The demonstration also shows a smaller tree can guide people to find more effective corrective actions.

5.9 Summary

Proactive incident prevention requires the identification of weak signals and appropriate responses to the weak signals. This chapter presented the development of a framework based on FRAM and machine learning techniques, which provided explicit guidance for people throughout the entire life cycle of weak signals from identifying to responding. The case study of a hypothetical batch process showed great potentials for applying the framework in real operations. Given the information of the potential weak signals that were extracted based on FRAM, probabilities of a selected hazard were predicted with high accuracy by the RF. Underlying weak signals and the corresponding corrective actions were identified from the DT. In order to improve the feasibility of the framework to be applied in real operations, several directions of future work can be explored such as investigating the stability of DT that is related to changes in training samples and expanding the scope of the framework from studying a single hazard to multiple hazards. Detailed discussions about the future work will be presented in Chapter VI.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK [*]

6.1 Conclusions

Most incidents in complex systems such as process plants are not-chance events and weak signals emerge a long time before incidents occur. To prevent incidents proactively, the dissertation aimed to understand the challenges of identifying and responding to weak signals and address them by developing data-driven solutions. The contributions and conclusions are summarized as follows:

Definition and characteristics of weak signals in the domain of safety were critically reviewed and formal definitions of weak signals and noise were proposed in Chapter II due to the imprecise definitions in the existing literatures. The proposed definition of weak signals emphasizes three aspects of weak signals:

- Weak signals can be performance variabilities of technological, human, or organization functions.
- Weak signals exist as combinations to indicate noticeable consequences.
- Weak signals give rise to early prediction of future incidents, and "early" are precisely defined to be early enough to indicate precursors.

[*] Part of this chapter is reprinted with permission from "Development of a FRAM-based framework to identify hazards in a complex system" by Mengxi Yu, Noor Quddus, Costas Kravaris, M. Sam Mannan, 2020. Journal of Loss Prevention in the Process Industries, 63, Pages 103994, Copyright 2019 by Elsevier Ltd. and from "A data-driven approach of quantifying function couplings and identifying paths towards emerging hazards in complex systems" by Mengxi Yu, Madhav Erraguntla, Noor Quddus, Costas Kravaris, 2021. Process Safety and Environmental Protection, 150, 2021, Pages 464-477, Copyright 2021 Institution of Chemical Engineers. Published by Elsevier B.V.

Due to the complex interactions in process industries, it could be intellectually unmanageable to recognize how weak signals interact and lead to emerging hazards. The system-based technique FRAM is a promising technique for understanding interactions in a complex system but has low degrees of automation and quantification for applications in process plants. Therefore, the work in Chapter III and Chapter IV answered the first research question regarding the limitations of FRAM to understand interactions in process plants.

The work in Chapter III developed a framework based on FRAM to simulate possible function interactions in process plants. The conclusions of the work in Chapter III are:

- Function interactions in process plants can be modeled based on FRAM, by integrating a human performance model, an equipment performance model, and a first-principle model into a hybrid simulator.
- The hybrid simulator is capable to describe performance variabilities of a function quantitatively and generate data of possible interactions for further analysis.

With the data of possible function interactions that were generated from simulations, Chapter VI presented the framework to identify those interactions leading to potential hazards. The conclusions of the work in Chapter VI were justified by the case study of a batch polymerization process and are summarized below:

- Lifts of association rules quantify upstream-downstream couplings.

- Lift confidence intervals provide uncertainties which address the randomness in data and provide references to evaluate whether the quantifications are supported by sufficient data.
- Interactions leading to hazard scenarios can be identified by merging association rules and their graphical representations with quantified couplings guide people to understand how emerging hazards occur and take preventive measures.

Besides, another challenge of recognizing weak signals is due to the existence of noise. Given the abundance of information in plants, weak signals are hard to be picked up. The work in Chapter V developed a data-driven framework that involves techniques from identifying, evaluating, and responding to weak signals. The conclusions of the work were justified by the case study of a batch polymerization process and are summarized below:

- FRAM can be used to identify potential sources of weak signals, from which machine learning techniques can further identify actual weak signals.
- BRF can predict occurrences of a rare hazard scenario with high recall and predict probabilities of the occurrences with high accuracy to indicate relevance of underlying weak signals.
- DT approximates the performance of BRF with high fidelity to predict occurrences of the hazard scenario. The interpretable DT shows weak signals and can provide guidance to respond to weak signals based on its hierarchy.

6.2 Future Work

In order to improve the feasibility of the framework to be applied in real operations, several directions of future work are discussed in this chapter.

6.2.1 Modeling function interactions

In Chapter III, the models that were integrated into the framework are theoretical models. To apply the framework in the industry, expert judgments or appropriate data are required to make necessary adjustments. For example, the equipment performance model and its parameters could be adjusted based on failure data. The nominal probabilities of generic failure types and weights of the CPCs that are used in CREAM could be modified based on real operations. Ranges of the probabilities could be integrated to capture more uncertainties of both equipment and human performance variabilities.

6.2.2 Function interactions leading to hazards

In Chapter IV, redundant rules were only removed conservatively in the study thus resulted in redundant paths which still requires manual efforts to identify the critical ones. The criteria for the automatic removal of redundant rules need to be further finetuned to achieve a concise set of the paths. Additionally, for the purpose of demonstration, numerical performance variabilities were categorized based on deviations from the desired performance. In the future, the numerical performance variabilities could be further discretized to provide deeper insights into safety constraints of the system. Lastly, computational time to extract association rules increases exponentially as a system expands. Future work can be done to improve efficiency of rule extractions. For example, an algorithm can be developed to only generate the association rules whose antecedents are upstream functions and consequents are downstream functions, instead of extracting such rules through additional filtering steps.

Regarding the depth of understanding emerging hazard, the observed data alone cannot identify the causal relationship in the function coupling and the domain knowledge was used to make the causal inference from the function coupling to establish the path to a hazard scenario. As Pearl explained in his ladder of causation, a causal analysis is required to investigate the complete causal relationship among the functions in a changing context, requiring a model at the intervention or counterfactual levels (Pearl, 2010; Pearl & Mackenzie, 2018). The transition from association to causal inferences by utilizing data-driven approach can be further investigated in future work.

6.2.3 Identification of weak signals

DT is widely applied since it is easily interpretable, but it could be unstable. There are two perspectives of the instability. First, given the same set of training samples, DT could have different structures due to randomness in the training process. Such perspective has been discussed and investigated in the study concluding a smaller tree provides more compact decision paths and more effective corrective actions. The other perspective of the instability is related to changes in training samples. Small changes in training samples may lead to a different tree. If some candidate split nodes have similar importance to distinguish classes, the resultant tree structure could be sensitive to small changes where a slightly inferior node split in one tree becomes slightly superior in another tree (Li & Belford, 2002). The instability is highly dependent on training samples (Aluja-Banet & Nafria, 2003), therefore a systematic investigation is recommended for future work, regarding how much difference in a specific training data that the Decision Tree algorithm can tolerate.

Additionally, since the study aimed to develop a novel framework by utilizing machine learning techniques to identify and respond to weak signals, the focus was to address how different techniques can be integrated together to overcome challenges throughout the life cycle of weak signals, instead of optimizing a specific machine learning algorithm. A DT algorithm based on a more stable node splitting method could be an alternative to the current one. The study utilized the most classic DT algorithm, which measures impurity in a node by Gini Index and recursively partitions a feature space, while a few researchers proposed DT algorithms that have the potentials to improve the stability of a DT but have not been extensively studied and commonly applied (Dwyer & Holte, 2007; Guggari *et al.*, 2018; Last *et al.*, 2002; Mirzamomen & Kangavari, 2017).

The probability that is predicted by RF is a quantitative measure to alert people about the relevance of underlying weak signals. An acceptable threshold should be pre-defined by an organization to call for responses to the weak signals, therefore individuals explicitly know whether they need to act on the weak signals or report to decision-makers at upper management levels. Determination of the threshold depends on the consequences of selected hazards, as well as the commitment of an organization to improve safety proactively such as expenses for responding to weak signals. Given predicted probabilities, a risk-based and/or cost-based measure could be further developed to determine whether responses to weak signals are needed.

Finally, the framework was developed based on binary classification problems which only addressed a single hazard each time. Extending the framework for multi-classification problems to deal with multiple hazards could improve its generalizations in operations.

REFERENCES

Akyuz, E.,Celik, M. 2015. Application of CREAM human reliability model to cargo loading process of LPG tankers. *Journal of Loss Prevention in the Process Industries, 34*(Supplement C), 39-48.

Aluja-Banet, T.,Nafria, E. 2003. Stability and scalability in decision trees. *Computational Statistics, 18*(3), 505-520. doi:10.1007/BF03354613

Anand, S., Keren, N., Tretter, M. J., Wang, Y., O'Connor, T. M.,Mannan, M. S. 2006. *Harnessing data mining to explore incident databases*.

Ansoff, I.,McDonnell, E. 1990. Implanting corporate strategy. *Hemel Hempstead*.

Asadzadeh, S. M.,Azadeh, A. 2014. An integrated systemic model for optimization of condition-based maintenance with human error. *Reliability Engineering & System Safety, 124*, 117-131.

Augasta, M. G.,Kathirvalavakumar, T. 2012. *Rule extraction from neural networks—A comparative study*. Paper presented at the International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012).

Ávila, S., Pessoa, F.,Andrade, J. C. S. 2013. Social HAZOP at an oil refinery. *Process Safety Progress, 32*(1), 17-21.

Baesens, B., Setiono, R., Mues, C.,Vanthienen, J. 2003. Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312-329.

Baillagou, P. E.,Soong, D. S. 1985a. MAJOR FACTORS CONTRIBUTING TO THE NONLINEAR KINETICS OF FREE-RADICAL POLYMERIZATION. *Chemical Engineering Science, 40*(1), 75-86. doi:10.1016/0009-2509(85)85048-X

Baillagou, P. E.,Soong, D. S. 1985b. Molecular weight distribution of products of free radical nonisothermal polymerization with gel effect. Simulation for polymerization of poly(methyl methacrylate). *Chemical Engineering Science, 40*(1), 87-104.

Barakat, N.,Bradley, A. P. 2010. Rule extraction from support vector machines: a review. *Neurocomputing, 74*(1-3), 178-190.

Bastani, O., Kim, C.,Bastani, H. 2017. Interpreting blackbox models via model extraction. *arXiv preprint arXiv:1705.08504*.

Batbarai, A.,Naidu, D. 2014. Approach for rule pruning in association rule mining for removing redundancy. *Int. J. Innov. Res. Comput. Commun. Eng, 2*(5), 4207-4213.

Bell, J.,Holroyd, J. 2009. Review of human reliability assessment methods. *Health and Safety Laboratory, United Kingdom*.

Bergstra, J.,Bengio, Y. 2012. Random search for hyper-parameter optimization. *the Journal of machine Learning research, 13*(1), 281-305.

Bevilacqua, M.,Ciarapica, F. E. 2018. Human factor risk management in the process industry: A case study. *Reliability Engineering & System Safety, 169*, 149-159.

Bevilacqua, M., Ciarapica, F. E.,Giacchetta, G. 2008. Industrial and occupational ergonomics in the petrochemical process industry: A regression trees approach. *Accident Analysis & Prevention, 40*(4), 1468-1479.

Bohn, C.,Atherton, D. 1995. An analysis package comparing PID anti-windup strategies. *IEEE Control Systems Magazine, 15*(2), 34-40.

Bonabeau, E. 2002. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences, 99*(suppl 3), 7280-7287.

Boström, H. 2008. *Calibrating random forests.* Paper presented at the 2008 Seventh International Conference on Machine Learning and Applications.

Breiman, L. 2001. Random Forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/A:1010933404324

Brewer, C. 2016. Improving Process Safety Management in Hazardous Industries.

Brizon, A.,Wybo, J.-L. 2009. The life cycle of weak signals related to safety. *International Journal of Emergency Management, 6*(2), 117-135.

Bruce, P.,Bruce, A. 2017. *Practical statistics for data scientists: 50 essential concepts*: " O'Reilly Media, Inc.".

Cameron, I., Mannan, S., Németh, E., Park, S., Pasman, H., Rogers, W.,Seligmann, B. 2017. Process hazard analysis, hazard identification and scenario definition: Are the conventional tools sufficient, or should and can we do much better? *Process Safety and Environmental Protection, 110*, 53-70. doi:10.1016/j.psep.2017.01.025

Carlo, F. D.,Arleo, M. A. (2017). Imperfect Maintenance Models, from Theory to Practice. In C. Volosencu (Ed.), *System Reliability* (pp. Ch. 18). Rijeka: InTech.

Carroll, J. S. 2004. Knowledge management in high-hazard industries. *Accident precursor analysis and management: Reducing technological risk through diligence,* 127-136.

Caruana, R., Karampatziakis, N.,Yessenalina, A. 2008. *An empirical evaluation of supervised learning in high dimensions.* Paper presented at the Proceedings of the 25th international conference on Machine learning.

Caruana, R.,Niculescu-Mizil, A. 2006. *An empirical comparison of supervised learning algorithms.* Paper presented at the Proceedings of the 23rd international conference on Machine learning.

CCPS. 1994. *Guidelines for preventing human error in process safety*: New York, NY : Center for Chemical Process Safety of the American Institute of Chemical Engineers, [1994].

Chawla, N. V.,Cieslak, D. A. 2006. *Evaluating probability estimates from decision trees.* Paper presented at the American Association for Artificial Intelligence.

Chen, C., Liaw, A.,Breiman, L. 2004. Using random forest to learn imbalanced data. *University of California, Berkeley, 110*(1-12), 24.

Chen, D., Fan, Y., Li, W., Wang, Y.,Zhang, S. 2019. Human reliability prediction in deep-sea sampling process of the manned submersible. *Safety Science, 112,* 1-8. doi:https://doi.org/10.1016/j.ssci.2018.10.001

Cheng, C.-W., Lin, C.-C.,Leu, S.-S. 2010. Use of association rules to explore cause–effect relationships in occupational accidents in the Taiwan construction industry. *Safety Science, 48*(4), 436-444. doi: https://doi.org/10.1016/j.ssci.2009.12.005

Chiu, W. Y., Carratt, G. M.,Soong, D. S. 1983. A computer model for the gel effect in free-radical polymerization. *Macromolecules, 16*(3), 348-357.

Chylla, R.,Haase, D. R. 1993. Temperature control of semibatch polymerization reactors. *Computers & Chemical Engineering, 17*(3), 257-264.

Coffman, B. 1997. Weak signal research, part I: Introduction. *Journal of Transition Management, 2*(1).

Crowley, T. J.,Choi, K.-Y. 1996. On-line monitoring and control of a batch polymerization reactor. *Journal of Process Control, 6*(2), 119-127. doi:https://doi.org/10.1016/0959-1524(95)00054-2

Dal Pozzolo, A., Caelen, O., Johnson, R. A.,Bontempi, G. 2015. *Calibrating probability with undersampling for unbalanced classification.* Paper presented at the 2015 IEEE Symposium Series on Computational Intelligence.

Douzas, G., Bacao, F.,Last, F. 2018. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences, 465*, 1-20. doi:https://doi.org/10.1016/j.ins.2018.06.056

Drupsteen, L.,Wybo, J. L. 2015. Assessing propensity to learn from safety-related events. *Safety Science, 71*, 28-38. doi:10.1016/j.ssci.2014.02.024

Du, M., Liu, N.,Hu, X. 2019. Techniques for interpretable machine learning. *Communications of the ACM, 63*(1), 68-77.

Duan, G., Tian, J.,Wu, J. 2015. Extended FRAM by integrating with model checking to effectively explore hazard evolution. *Mathematical Problems in Engineering, 2015*.

Dwyer, K.,Holte, R. 2007. *Decision tree instability and active learning.* Paper presented at the European conference on machine learning.

Efron, B.,Tibshirani, R. J. 1994. *An introduction to the bootstrap*: CRC press.

Embrey, D., Humphreys, P., Rosa, E., Kirwan, B.,Rea, K. (1984). *SLIM-MAUD: an approach to assessing human error probabilities using structured expert judgment. Volume II. Detailed analysis of the technical issues*. Retrieved from

Frost, B.,Mo, J. P. 2014. *System hazard analysis of a complex socio-technical system: the functional resonance analysis method in hazard identification.* Paper presented at the Proc. of Australian System Safety Conference, Melbourne Australia.

Ge, Z., Song, Z., Ding, S. X.,Huang, B. 2017. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access, 5*, 20590-20616. doi:10.1109/ACCESS.2017.2756872

Gertman, D., Blackman, H., Marble, J., Byers, J.,Smith, C. 2005. The SPAR-H human reliability analysis method. *US Nuclear Regulatory Commission*.

Goh, Y. M.,Chua, D. 2013. Neural network analysis of construction safety management systems: a case study in Singapore. *Construction Management and Economics, 31*(5), 460-470.

Goh, Y. M., Ubeynarayana, C. U., Wong, K. L. X.,Guo, B. H. 2018. Factors influencing unsafe behaviors: A supervised learning approach. *Accident Analysis & Prevention, 118*, 77-85.

Guggari, S., Kadappa, V.,Umadevi, V. 2018. Non-sequential partitioning approaches to decision tree classifier. *Future Computing and Informatics Journal, 3*(2), 275-285. doi:https://doi.org/10.1016/j.fcij.2018.06.003

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F.,Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR), 51*(5), 1-42.

Guillaume, E. (2011). *Identifying and responding to weak signals to improve learning from experiences in high-risk industry*. TU Delft, Delft University of Technology.

Haji-Kazemi, S.,Andersen, B. 2013. Application of performance measurement as an early warning system. *International Journal of Managing Projects in Business*.

Halim, S. Z.,Mannan, M. S. 2018. A journey to excellence in process safety management. *Journal of Loss Prevention in the Process Industries, 55*, 71-79. doi:https://doi.org/10.1016/j.jlp.2018.06.002

Han, J., Pei, J.,Kamber, M. 2011. *Data mining: concepts and techniques*: Elsevier.

Han, L., Luo, S., Yu, J., Pan, L.,Chen, S. 2014. Rule extraction from support vector machines using ensemble learning approach: an application for diagnosis of diabetes. *IEEE journal of biomedical and health informatics, 19*(2), 728-734.

He, J., Hu, H.-J., Harrison, R., Tai, P. C.,Pan, Y. 2006. Rule generation for protein secondary structure prediction with support vector machines and decision tree. *IEEE Transactions on nanobioscience, 5*(1), 46-53.

Hill, R.,Hollnagel, E. 2016. FRAM Model visualizer. Retrieved from https://functionalresonance.com/FMV/index.html (accessed 10-01-2020).

Hollnagel, E. 1998. *Cognitive reliability and error analysis method (CREAM)*: Elsevier.

Hollnagel, E. 2004. Barriers and accident prevention: or how to improve safety by understanding the nature of accidents rather than finding their causes. *Hampshire (United Kingdom)*.

Hollnagel, E. 2017. *FRAM: the functional resonance analysis method: modelling complex socio-technical systems*: CRC Press.

Holopainen, M.,Toivonen, M. 2012. Weak signals: Ansoff today. *Futures, 44*(3), 198-205. doi:https://doi.org/10.1016/j.futures.2011.10.002

Hopkins, A. 2008. *Failure to learn : the BP Texas City refinery disaster. Andrew Hopkins*: Sydney, N.S.W. : CCH Australia, 2008.

Hu, J.,Guo, F. 2016. *Application of data mining technology in selecting key parameters of chemical processes.* Paper presented at the 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD 2016, August 13, 2016 - August 15, 2016, Changsha, China.

Jahangiri, M., Hoboubi, N., Rostamabadi, A., Keshavarzi, S.,Hosseini, A. A. 2016. Human Error Analysis in a Permit to Work System: A Case Study in a Chemical

Plant. *Safety and Health at Work, 7*(1), 6-11. doi:https://doi.org/10.1016/j.shaw.2015.06.002

James, G., Witten, D., Hastie, T.,Tibshirani, R. 2013. *An introduction to statistical learning* (Vol. 112): Springer.

Jiawei, H.,Yongjian, F. 1999. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering, 11*(5), 798-805. doi:10.1109/69.806937

Kakhki, F. D., Freeman, S. A.,Mosher, G. A. 2019. Evaluating machine learning performance in predicting injury severity in agribusiness industries. *Safety Science, 117*, 257-262. doi:https://doi.org/10.1016/j.ssci.2019.04.026

Kang, K.,Ryu, H. 2019. Predicting types of occupational accidents at construction sites in Korea using random forest model. *Safety Science, 120*, 226-236.

Kannan, P., Flechas, T., Mendez, E., Angarita, L., Chaudhari, P., Hong, Y.,Mannan, M. S. 2016. A web-based collection and analysis of process safety incidents. *Journal of Loss Prevention in the Process Industries, 44*, 171-192.

Kariuki, S. G.,Löwe, K. 2007. Integrating human factors into process hazard analysis. *Reliability Engineering & System Safety, 92*(12), 1764-1773. doi:https://doi.org/10.1016/j.ress.2007.01.002

Kennedy, R.,Kirwan, B. 1998. Development of a hazard and operability-based method for identifying safety management vulnerabilities in high risk systems. *Safety Science, 30*(3), 249-274.

Keren, N., Anand, S.,Sam Mannan, M. 2006. Calibrate failure-based risk assessments to take into account the type of chemical processed in equipment. *Journal of Loss Prevention in the Process Industries, 19*(6), 714-718. doi:10.1016/j.jlp.2006.05.005

Kim, K., Crowley, T.,Choi, K. (1993). On-line estimation and control of polymerization reactors *Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes* (pp. 161-166): Elsevier.

Kirwan, B. 1992a. Human error identification in human reliability assessment. Part 1: Overview of approaches. *Applied Ergonomics, 23*(5), 299-318. doi:https://doi.org/10.1016/0003-6870(92)90292-4

Kirwan, B. 1992b. Human error identification in human reliability assessment. Part 2: Detailed comparison of techniques. *Applied Ergonomics, 23*(6), 371-381. doi:https://doi.org/10.1016/0003-6870(92)90368-6

Kirwan, B. 1996. The validation of three human reliability quantification techniques — THERP, HEART and JHEDI: Part 1 — technique descriptions and validation issues. *Applied Ergonomics, 27*(6), 359-373. doi:https://doi.org/10.1016/S0003-6870(96)00044-0

Koivisto, R., Kulmala, I.,Gotcheva, N. 2016. Weak signals and damage scenarios—Systematics to identify weak signals and their sources related to mass transport attacks. *Technological Forecasting and Social Change, 104,* 180-190.

Köpke, C., Schäfer-Frey, J., Engler, E., Wrede, C. P.,Mielniczek, J. 2020. *A joint approach to safety, security and resilience using the Functional Resonance*

Analysis Method. Paper presented at the 8th REA Symposium on Resilience Engineering: Scaling up and Speeding up.

Körvers, P. M. W. (2004). *Accident precursors: pro-active identification of safety risks in the chemical process industry.* Technische Universiteit Eindhoven.

Kunreuther, H. C., Bier, V. M.,Phimister, J. R. 2004. *Accident precursor analysis and management: reducing technological risk through diligence*: National Academies Press.

Last, M., Maimon, O.,Minkov, E. 2002. Improving stability of decision trees. *International Journal of Pattern Recognition and Artificial Intelligence, 16*(02), 145-159.

Le Coze, J.-C. 2008. *BP Texas city accident: weak signal or sheer power?*

Lemaître, G., Nogueira, F.,Aridas, C. K. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *the Journal of machine Learning research, 18*(1), 559-563.

Leveson, N. 2004. A new accident model for engineering safer systems. *Safety Science, 42*(4), 237-270.

Leveson, N. 2011. *Engineering a safer world: Systems thinking applied to safety*: MIT press.

Li, R.-H.,Belford, G. G. 2002. *Instability of decision tree classification algorithms.* Paper presented at the Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.

Lin, Z., Huang, Y.,Fang, C. 2015. Non-periodic preventive maintenance with reliability thresholds for complex repairable systems. *Reliability Engineering & System Safety, 136*, 145-156. doi:https://doi.org/10.1016/j.ress.2014.12.010

Liu, Y., Huang, H. Z.,Zhang, X. 2012. A Data-Driven Approach to Selecting Imperfect Maintenance Models. *IEEE Transactions on Reliability, 61*(1), 101-112. doi:10.1109/TR.2011.2170252

Luyk, J. 2011. Towards improving detection of early warning signals within organizations: an approach to the identification and utilization of underlying factors from an organizational perspective.

Mantovani, R. G., Rossi, A. L., Vanschoren, J., Bischl, B.,De Carvalho, A. C. 2015. *Effectiveness of random search in SVM hyper-parameter tuning.* Paper presented at the 2015 International Joint Conference on Neural Networks (IJCNN).

Massaiu, S.,Paltrinieri, N. (2016). Chapter 14 - Human Reliability Analysis: From the Nuclear to the Petroleum Sector *Dynamic Risk Analysis in the Chemical and Petroleum Industry* (pp. 171-179): Butterworth-Heinemann.

McKay, M. D., Beckman, R. J.,Conover, W. J. 1979. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 21*(2), 239-245.

Mehta, M., Rissanen, J.,Agrawal, R. 1995. *MDL-Based Decision Tree Pruning.* Paper presented at the KDD.

Mirzamomen, Z.,Kangavari, M. R. 2017. A framework to induce more stable decision trees for pattern classification. *Pattern Analysis and Applications, 20*(4), 991-1004. doi:10.1007/s10044-016-0542-2

Mistikoglu, G., Gerek, I. H., Erdis, E., Mumtaz Usmen, P. E., Cakan, H.,Kazan, E. E. 2015. Decision tree analysis of construction fall accidents involving roofers. *Expert Systems with Applications, 42*(4), 2256-2263. doi:https://doi.org/10.1016/j.eswa.2014.10.009

Narodytska, N., Ignatiev, A., Pereira, F., Marques-Silva, J.,RAS, I. 2018. *Learning Optimal Decision Trees with SAT*. Paper presented at the IJCAI.

Nguyen, D. Q., Brammer, C.,Bagajewicz, M. 2008. New tool for the evaluation of the scheduling of preventive maintenance for chemical process plants. *Industrial & engineering chemistry research, 47*(6), 1910-1924.

Niculescu-Mizil, A.,Caruana, R. 2005. *Predicting good probabilities with supervised learning*. Paper presented at the Proceedings of the 22nd international conference on Machine learning.

Øien, K., Utne, I. B.,Herrera, I. A. 2011. Building safety indicators: Part 1–theoretical foundation. *Safety Science, 49*(2), 148-161.

OSHA, P. 1992. Rule "Process Safety Management of Highly Hazardous Chemicals; Explosives and Blasting Agents," 1910.0119; 1920.0109, Final Rules. *Federal Register, 57*, 6356.

Papakonstantinou, N., Sierla, S., Jensen, D. C.,Tumer, I. Y. 2011. Capturing Interactions and Emergent Failure Behavior in Complex Engineered Systems at Multiple Scales. (54792), 1045-1054. doi:10.1115/DETC2011-47767

Papakonstantinou, N., Sierla, S., Jensen, D. C.,Tumer, I. Y. 2012. Simulation of Interactions and Emergent Failure Behavior During Complex System Design. *Journal of Computing and Information Science in Engineering, 12*(3), 031007-031007-031010. doi:10.1115/1.4007309

Pascal, J.,Sahbani, A. 2000. *Simulation of hybrid systems using stateflow*. Paper presented at the 14th European Simulation Multiconference (ESM'2000).

Pasman, H. J. 2020. Early warning signals noticed, but management doesn't act adequately or not at all: a brief analysis and direction of possible improvement. *Journal of Loss Prevention in the Process Industries*, 104272. doi:https://doi.org/10.1016/j.jlp.2020.104272

Patriarca, R., Del Pinto, G., Di Gravio, G.,Costantino, F. 2018. FRAM for systemic accident analysis: a matrix representation of functional resonance. *International Journal of reliability, Quality and safety Engineering, 25*(01), 1850001.

Patriarca, R., Di Gravio, G.,Costantino, F. 2017a. A Monte Carlo evolution of the Functional Resonance Analysis Method (FRAM) to assess performance variability in complex systems. *Safety Science, 91*, 49-60.

Patriarca, R., Di Gravio, G., Costantino, F.,Tronci, M. 2017b. The Functional Resonance Analysis Method for a systemic risk based environmental auditing in a sinter plant:

A semi-quantitative approach. *Environmental Impact Assessment Review, 63*, 72-86.

Patriarca, R., Di Gravio, G., Woltjer, R., Costantino, F., Praetorius, G., Ferreira, P.,Hollnagel, E. 2020. Framing the FRAM: A literature review on the functional resonance analysis method. *Safety Science*, *129*, 104827. doi:https://doi.org/10.1016/j.ssci.2020.104827

Pearl, J. 2010. *Causal inference.* Paper presented at the Causality: Objectives and Assessment.

Pearl, J.,Mackenzie, D. 2018. *The book of why: the new science of cause and effect*: Basic Books.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Perrow, C. 1999. *Normal Accidents : Living with High Risk Technologies - Updated Edition*. Princeton, N.J.: Princeton University Press.

Pham, H.,Wang, H. 1996. Imperfect maintenance. *European Journal of Operational Research*, *94*(3), 425-438. doi:https://doi.org/10.1016/S0377-2217(96)00099-9

Platt, J. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers, 10*(3), 61-74.

Poh, C. Q. X., Ubeynarayana, C. U.,Goh, Y. M. 2018. Safety leading indicators for construction sites: A machine learning approach. *Automation in Construction, 93*, 375-386. doi:https://doi.org/10.1016/j.autcon.2018.03.022

Qin, S. J. 2014. Process data analytics in the era of big data. *AIChE Journal, 60*(9), 3092-3100.

Raschka, S. 2018. MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *J. Open Source Software, 3*(24), 638.

Rasmussen, J. 1997. Risk management in a dynamic society: a modelling problem. *Safety Science, 27*(2), 183-213. doi:https://doi.org/10.1016/S0925-7535(97)00052-0

Rasmussen, J.,Suedung, I. 2000. *Proactive risk management in a dynamic society*: Swedish Rescue Services Agency.

Rivas, T., Paz, M., Martín, J., Matías, J. M., García, J. F.,Taboada, J. 2011. Explaining and predicting workplace accidents using data-mining techniques. *Reliability Engineering & System Safety, 96*(7), 739-747.

Rosa, L. V., França, J. E., Haddad, A. N.,Carvalho, P. V. 2017. A resilience engineering approach for sustainable safety in green construction. *Journal of Sustainable Development of Energy, Water and Environment Systems, 5*(4), 480-495.

Rosa, L. V., Haddad, A. N.,Carvalho, P. V. 2015. Assessing risk in sustainable construction using the Functional Resonance Analysis Method (FRAM). *Cognition, Technology & Work, 17*(4), 559-573.

Rossel, P. 2009. Weak signals as a flexible framing space for enhanced management and decision-making. *Technology Analysis & Strategic Management, 21*(3), 307-320.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence, 1*(5), 206-215.

Saito, T.,Rehmsmeier, M. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *Plos One, 10*(3).

Saleh, J. H., Saltmarsh, E. A., Favarò, F. M.,Brevault, L. 2013. Accident precursors, near misses, and warning signs: Critical review and formal definitions within the framework of Discrete Event Systems. *Reliability Engineering & System Safety, 114*, 148-154.

Sarkar, S., Patel, A., Madaan, S.,Maiti, J. 2016. *Prediction of occupational accidents using decision tree approach.* Paper presented at the 2016 IEEE Annual India Conference (INDICON).

Sarkar, S., Verma, A.,Maiti, J. (2018). Prediction of occupational incidents using proactive and reactive data: a data mining approach *Industrial Safety Management* (pp. 65-79): Springer.

Schurman, D. L.,Fleger, S. A. 1994. Human factors in HAZOPs: guide words and parameters. *Professional Safety, 39*(12), 32.

Shelly, D. R., Beroza, G. C.,Ide, S. 2007. Non-volcanic tremor and low-frequency earthquake swarms. *Nature, 446*(7133), 305-307.

Shokria, S. 2017. A Cognitive Human Error Analysis with CREAM in Control Room of Petrochemical Industry. *Biotechnology and Health Sciences*(1), 13-21.

Smoczyński, P., Kadziński, A.,Gill, A. (2018). Simulating the world described with the functional resonance analysis method *Safety and Reliability–Safe Societies in a Changing World* (pp. 1247-1252): CRC Press.

Soroush, M.,Kravaris, C. 1992. Nonlinear control of a batch polymerization reactor: An experimental study. *AIChE Journal, 38*(9), 1429-1448. doi:10.1002/aic.690380914

Soroush, M.,Kravaris, C. 1993. Optimal design and operation of batch reactors. 2. A case study. *Industrial & engineering chemistry research, 32*(5), 882-893.

Tabatabaei, N. (2011). *Detecting weak signals by internet-based environmental scanning.* University of Waterloo.

Tixier, A. J.-P., Hallowell, M. R., Rajagopalan, B.,Bowman, D. 2016. Application of machine learning to construction injury prediction. *Automation in Construction, 69*, 102-114.

Turner, B. A.,Pidgeon, N. F. 1997. *Man-made disasters*: Butterworth-Heinemann.

Valdez‐Flores, C.,Feldman, R. M. 1989. A survey of preventive maintenance models for stochastically deteriorating single‐unit systems. *Naval Research Logistics (NRL), 36*(4), 419-446.

Vaughan, D. 1997. *The Challenger launch decision: Risky technology, culture, and deviance at NASA*: University of Chicago Press.

Vaughan, D. 2002. Signals and interpretive work: The role of culture in a theory of practical action. *Culture in mind: Toward a sociology of culture and cognition*, 28-54.

Vaughan, D. 2004. Theorizing disaster: Analogy, historical ethnography, and the Challenger accident. *Ethnography, 5*(3), 315-347.

Verma, A., Khan, S. D., Maiti, J.,Krishna, O. B. 2014. Identifying patterns of safety related incidents in a steel plant using association rule mining of incident investigation reports. *Safety Science, 70*, 89-98. doi:https://doi.org/10.1016/j.ssci.2014.05.007

Waitman, L. R., Fisher, D. H.,King, P. H. 2006. Bootstrapping rule induction to achieve rule stability and reduction. *Journal of Intelligent Information Systems, 27*(1), 49-77.

Wallace, B. C.,Dahabreh, I. J. 2012, 10-13 Dec. 2012. *Class Probability Estimates are Unreliable for Imbalanced Data (and How to Fix Them).* Paper presented at the 2012 IEEE 12th International Conference on Data Mining.

Wehrens, R., Putter, H.,Buydens, L. M. 2000. The bootstrap: a tutorial. *Chemometrics and intelligent laboratory systems, 54*(1), 35-52.

Weick, K., Sutcliffe, K.,Obstfeld, D. (1999). Organizing for high reliability. Processes of collective mindfulness. JAI: Stanford.

Wieme, J., De Roo, T., Marin, G. B.,Heynderickx, G. J. 2007. Simulation of pilot- and industrial-scale vinyl chloride batch suspension polymerization reactors. *Industrial and Engineering Chemistry Research, 46*(4), 1179-1196. doi:10.1021/ie0602355

Wright, R. A.,Kravaris, C. 1997. *Practical development of a nonlinear controller for an industrial-scale batch polymerization reactor*. Paper presented at the American Control Conference, 1997. Proceedings of the 1997.

Wyss, G. D.,Jorgensen, K. H. (1998). *A users guide to LHS: Sandias Latin hypercube sampling software*. Retrieved from

Xu, S., Lu, B., Baldea, M., Edgar, T. F., Wojsznis, W., Blevins, T.,Nixon, M. 2015. Data cleaning in the process industries. *Reviews in Chemical Engineering, 31*(5), 453-490. doi:10.1515/revce-2015-0022

Yang, C., Rangarajan, A.,Ranka, S. 2018. *Global model interpretation via recursive partitioning*. Paper presented at the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS).

Yang, M., Gao, J., Zhao, B., Huang, X.,Shen, Z. 1997. Study of operator reliability in nuclear power plants. *Chinese Science Bulletin, 42*(19), 1585-1590. doi:10.1007/bf02882562

Yang, Q.,Tian, J. 2015. *Model-Based Safety Assessment using FRAM for complex systems*. In Proceedings of the 25th European Safety and Reliability Conference.

Zadrozny, B.,Elkan, C. 2001. *Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers*. Paper presented at the Icml.

Zadrozny, B.,Elkan, C. 2002. *Transforming classifier scores into accurate multiclass probability estimates.* Paper presented at the Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.

Zhang, H., De Saporta, B., Dufour, F.,Deleuze, G. 2013. *Dynamic reliability by using simulink and stateflow*.

Zhang, J., Li, Q., Guo, Q.,Wang, Z. 2007. *A simulation method of controlled hybrid Petri nets based on Matlab Simulink/Stateflow*. Paper presented at the 2007 IEEE International Conference on Automation and Logistics.

Zheng, Z.,Tian, J. 2015, 21-23 Oct. 2015. *Bridging the gap between FRAM and safety practice by applying FSM and model checking.* Paper presented at the 2015 First International Conference on Reliability Systems Engineering (ICRSE).

Zheng, Z., Tian, J.,Zhao, T. 2016. Refining operation guidelines with model-checking-aided FRAM to improve manufacturing processes: a case study for aeroengine blade forging. *Cognition, Technology & Work, 18*(4), 777-791. doi:10.1007/s10111-016-0391-1

Zhong, W.,Kwok, J. T. 2013. *Accurate probability calibration for multiple classifiers.* Paper presented at the Twenty-Third International Joint Conference on Artificial Intelligence.

Zhou, Q., Wong, Y. D., Loh, H. S.,Yuen, K. F. 2018. A fuzzy and Bayesian network CREAM model for human reliability analysis – The case of tanker shipping. *Safety Science, 105*, 149-157. doi:https://doi.org/10.1016/j.ssci.2018.02.011

APPENDIX A

MATLAB CODE FOR SIMULATION

A.1 Simulate Batches During a 5-year Operating Time

```
%% set seed number
seed = %set an integer
s = RandStream('mt19937ar','Seed', seed);
RandStream.setGlobalStream(s);

%% sample failure time of utility water pump based on preventive
maintenance schedule
%%%%%%%% use code in A.2 and A.3 here accordingly%%%%%%%%%%%%%%%

%% initialize matrices to save data
n=3650; % 3650 batches in a 5-year operating time
servicetimematrix = 10^20*ones(1,n);
variables = 10^20*ones(28801,8,n);%8 logged time-series signals:
timestamp, Fcw_max, Cm, Tj, T, Ci, Fcw, Power
ini_c = 10^20*ones(n,15); % 15 variables to initiate a batch operation

%% run simulation
for i =1:n % %run simulator for n times, based on simulation input
objects
      in(i) = Simulink.SimulationInput('hybridsimulator');

% assign pump failure timestamp to the block 'failuretime'
if i==1

      in(i)=in(i).setBlockParameter('hybridsimulator/prev_servicetime',
      'Value','0'); % when 1st batch starts, assign service time of
      previous batches 0 to the block 'prev_servicetime' ;
      index (i) = find(floor(failtimelist_sort)>0,1); % find the index
      of the smallest failure timestamp which is greater than 0
      in(i)=
      in(i).setBlockParameter('hybridsimulator/failuretime','Value',num
      2str(failtimelist_sort(index(i)))); % assign the failure
      timestamp which is corresponding to the index to the block
      'failuretime'
elseif i>1 &&
~isempty(find(floor(failtimelist_sort)>sum(servicetimematrix(1:i-
1)),1)) %if a sampled failure timestamp that is greater the sum of
service time of previous batches can be found

      in(i)=in(i).setBlockParameter('hybridsimulator/prev_servicetime',
      'Value',num2str(sum(servicetimematrix(1:i-1)))); % assign service
      time of previous batches to the block 'prev_servicetime'
      index(i) =
      find(floor(failtimelist_sort)>sum(servicetimematrix(1:i-1)),1); %
      find the index of the smallest failure timestamp which is greater
      than sum of service time of previous batches
        in(i) =
      in(i).setBlockParameter('hybridsimulator/failuretime','Value',num
      2str(failtimelist_sort(index(i)))); % assign the failure
      timestamp which is corresponding to the index to the block
      'failuretime'
else %%if a sampled failure timestamp that is greater the sum of
service time of previous batches cannot be found

      in(i)=in(i).setBlockParameter('hybridsimulator/prev_servicetime',
      'Value',num2str(sum(servicetimematrix(1:i-1)))); % assign service
      time of previous batches to the block 'prev_servicetime'
      index(i) = 0; % assign 0 to the index
      in(i) =
      in(i).setBlockParameter('hybridsimulator/failuretime','Value','0'
      ); %assign 0 to the block of 'failuretime'
end

%sample performance levels of organization functions and assign the
values to corresponding blocks
%control room operator
in(i)
=in(i).setBlockParameter('hybridsimulator/ade_org','Value',num2str('2')
);
in(i)
=in(i).setBlockParameter('hybridsimulator/work_cond','Value',num2str(ra
ndi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_MMI','Value',num2str('2'
));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_proc','Value',num2str('2
'));
in(i)
=in(i).setBlockParameter('hybridsimulator/no_simu_goal','Value',num2str
(randi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_time','Value',num2str(ra
ndi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/time_of_day','Value',num2str(
randi(2)));
in(i)
=in(i).setBlockParameter('hybridsimulator/ade_training','Value',num2str
(randi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/crew_collab','Value',num2str(
'2'));
%field_operator
in(i)
=in(i).setBlockParameter('hybridsimulator/ade_org2','Value',num2str('2'
));
```

```
in(i)
=in(i).setBlockParameter('hybridsimulator/work_cond2','Value',num2str(r
andi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_MMI2','Value',num2str('2
'));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_proc2','Value',num2str('
2'));
in(i)
=in(i).setBlockParameter('hybridsimulator/no_simu_goal2','Value',num2st
r(randi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/avai_time2','Value',num2str(r
andi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/ade_training2','Value',num2st
r(randi(3)));
in(i)
=in(i).setBlockParameter('hybridsimulator/crew_collab2','Value',num2str
('2'));

%start simulation and write data to workspace
simout(i)=sim(in(i));

% when simulation of a normal batch is skipped (as section E designed
in the hybrid simulator),
% load variables during a normal condition which are extracted in
advance
normal= load('variables_normal.mat');
if isempty(simout(1,i).logsout{1}.Values.Time)
    servicetimematrix(:,i)= 2.419305555548399e+04; % service time of
    utility pump during a normal condition
    variables(:,:,i)= normal.variables;
else
    % if the simulation of a batch is not skipped
    result(i)= simout(1,i).get('service_time'); %extract time-series
    cumulative service time that is saved in workspace
    servicetime(i)=max(result(i).Data); % extract the maximum service
    time as the total service time of a batch
    servicetimematrix(:,i)= servicetime(i); % write the maximum service
    time to matrix
    variables(:,1,i)=simout(1,i).logsout{1}.Values.Time; % extract
    timestamps from workspace
    for ind=1:7 % extract other time-series variables from workspace
          variables(:,ind+1,i)=simout(1,i).logsout{ind}.Values.Data;
    end
end
end

%% export data from the hybrid simulator
% store variables which initiate a batch and are not changed along the
process
```

```
for i =1:n
      ini_c(i,:)=[i,...
          simout(1,i).Cm0_Cp.Data,...
          simout(1,i).E4_error.Data,...
          simout(1,i).I1O3_error.Data,...
          simout(1,i).CPC_level.work_cond.Data,...
          simout(1,i).CPC_level.no_simu_goal.Data,...
          simout(1,i).CPC_level.avai_time.Data,...
          simout(1,i).CPC_level.time_of_day.Data,...
          simout(1,i).CPC_level.ade_training.Data,...
          simout(1,i).failure_time.Data,...
          simout(1,i).mass_i0.Data,...
          simout(1,i).CPC_level2.work_cond2.Data,...
          simout(1,i).CPC_level2.no_simu_goal2.Data,...
          simout(1,i).CPC_level2.avai_time2.Data,...
          simout(1,i).CPC_level2.ade_training2.Data];
end
save(sprintf('ini_c_s%d.mat',seed),'ini_c','-v7.3')
%store time-series variables
save(sprintf('variables_s%d.mat',seed),'variables','-v7.3')
```

A.2 Sample Timestamps of Utility Pump Failures (No PM)

```
t_life= 0:1/3600:43800; % discretize 43800-hr(5-year) operating time of
pump in seconds
beta = 1.2; %deterioration rate
eta=10000; %characteristic life in hours
alpha = 1/(eta^beta);% scale parameter
F_nopm=1-exp(-alpha*(t_life.^beta)); % cumulative failure rate
(Equation III-8)
Nr_nopm=round(alpha*(43800)^beta); % expected number of failures
(Equation III-9)

%% use LHS to sample failure timestamps based on Nr_nopm
for j=1:Nr_nopm
      P_LHS(j)=F_nopm(1)+(1-F_nopm(1))/Nr_nopm*(rand+(j-1)); % sampled
      cumulative failure probability for n_th failure (Equation III-15)
end
t_life_inv=interp1(F_nopm,t_life,P_LHS,'linear');   % calculate
timestamps of failures based on sampled cumulative failure probability
t_life_inv(isnan(t_life_inv))=4.6*10^4; % represent all the failure
timestamps beyond the 5-year range
failtimelist_sort= t_life_inv.*3600; %convert unit of timestamps of
failures from hours to seconds
```

A.3 Sample Timestamps of Utility Pump Failures (Optimal PM)

```
t_life= 0:1/3600:43800; %discretize 43800-hr(5-year) operating time of
pump
beta = 1.2; % deterioration rate
eta=10000; % characteristic life in hours
alpha = 1/(eta^beta); %scale parameter
rc=0.7; %critical reliability
ksi=0.7; %improvement factor

yi=(-log(rc)/alpha)^(1/beta); %effective age right before a PM(Equation
III-3)
x1=yi; %time interval between 0 and 1st PM
xi=(1-ksi)*yi; % time interval between two consecutive PMs after 1st PM

% calculate cumulative reliability based on Equation III-5
% As the Equation III-5 shown, the cumulative reliability is a periodic
% function, the for loop below is to derive cumulative reliability
during the operating time between two consecutive PMs.
% To calculate # of PMs during 43800 hrs
    % n is number of PMs
    % x1+(n-1)*xi=43800 -> n = 32.1381,
    % thus 32 PMs can be conducted until the 5-year operating time ends
for i=1:length(t_life)
      if t_life(i)>=0 && t_life(i) < x1 % before 1st PM
          t(i) = t_life(i);
          R(i) = exp(-alpha*((t(i))^beta));
      elseif t_life(i)>= x1 && t_life(i) < x1+xi % between 1st and 2nd
      PMs
          t(i) = t_life(i)-x1+ksi*yi;
          R(i)= exp (-alpha*((t(i))^beta+(1-ksi^beta)*yi^beta));
      elseif  t_life(i)>= x1+xi && t_life(i) < x1+2*xi % between 2nd
      and 3rd PMs
          t(i) = t_life(i)-x1-(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+2*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+2*xi && t_life(i) < x1+3*xi
          t(i) = t_life(i)-x1-2*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+3*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+3*xi && t_life(i) < x1+4*xi
          t(i) = t_life(i)-x1-3*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+4*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+4*xi && t_life(i) < x1+5*xi
          t(i) = t_life(i)-x1-4*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+5*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+5*xi && t_life(i) < x1+6*xi
          t(i) = t_life(i)-x1-5*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+6*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+6*xi && t_life(i) < x1+7*xi
          t(i) = t_life(i)-x1-6*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+7*(1-ksi^beta)*yi^beta));
      elseif t_life(i)>= x1+7*xi && t_life(i) < x1+8*xi
          t(i) = t_life(i)-x1-7*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+8*(1-ksi^beta)*yi^beta));
```

```
elseif t_life(i)>= x1+8*xi && t_life(i) < x1+9*xi
    t(i) = t_life(i)-x1-8*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+9*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+9*xi && t_life(i) < x1+10*xi
    t(i) = t_life(i)-x1-9*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+10*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+10*xi && t_life(i) < x1+11*xi
    t(i) = t_life(i)-x1-10*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+11*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+11*xi && t_life(i) < x1+12*xi
    t(i) = t_life(i)-x1-11*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+12*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+12*xi && t_life(i) < x1+13*xi
    t(i) = t_life(i)-x1-12*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+13*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+13*xi && t_life(i) < x1+14*xi
    t(i) = t_life(i)-x1-13*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+14*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+14*xi && t_life(i) < x1+15*xi
    t(i) = t_life(i)-x1-14*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+15*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+15*xi && t_life(i) < x1+16*xi
    t(i) = t_life(i)-x1-15*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+16*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+16*xi && t_life(i) < x1+17*xi
    t(i) = t_life(i)-x1-16*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+17*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+17*xi && t_life(i) < x1+18*xi
    t(i) = t_life(i)-x1-17*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+18*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+18*xi && t_life(i) < x1+19*xi
    t(i) = t_life(i)-x1-18*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+19*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+19*xi && t_life(i) < x1+20*xi
    t(i) = t_life(i)-x1-19*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+20*(1-ksi^beta)*yi^beta));

elseif t_life(i)>= x1+20*xi && t_life(i) < x1+21*xi
    t(i) = t_life(i)-x1-20*(1-ksi)*yi+ksi*yi;
    R(i)=exp(-alpha*((t(i))^beta+21*(1-ksi^beta)*yi^beta));
```

```
      elseif t_life(i)>= x1+21*xi && t_life(i) < x1+22*xi
          t(i) = t_life(i)-x1-21*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+22*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+22*xi && t_life(i) < x1+23*xi
          t(i) = t_life(i)-x1-22*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+23*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+23*xi && t_life(i) < x1+24*xi
          t(i) = t_life(i)-x1-23*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+24*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+24*xi && t_life(i) < x1+25*xi
          t(i) = t_life(i)-x1-24*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+25*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+25*xi && t_life(i) < x1+26*xi
          t(i) = t_life(i)-x1-25*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+26*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+26*xi && t_life(i) < x1+27*xi
          t(i) = t_life(i)-x1-26*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+27*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+27*xi && t_life(i) < x1+28*xi
          t(i) = t_life(i)-x1-27*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+28*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+28*xi && t_life(i) < x1+29*xi
          t(i) = t_life(i)-x1-28*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+29*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+29*xi && t_life(i) < x1+30*xi
          t(i) = t_life(i)-x1-29*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+30*(1-ksi^beta)*yi^beta));

      elseif t_life(i)>= x1+30*xi && t_life(i) < x1+31*xi  % between 31
      and 32 PMs
          t(i) = t_life(i)-x1-30*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+31*(1-ksi^beta)*yi^beta));

      else % after 32th PM is conducted
          t(i) = t_life(i)-x1-31*(1-ksi)*yi+ksi*yi;
          R(i)=exp(-alpha*((t(i))^beta+32*(1-ksi^beta)*yi^beta));
end
F_withpm(i) = 1-R(i); %cumulative failure rate after (i-1)th PM
(Equation III-5)
end
t_end=43800-x1-31*(1-ksi)*yi+ksi*yi; %the effective age when 43800 hrs
is reached (Equation III-7)
```

```
Nr_withpm = round(alpha*((yi^beta-0^beta)+31*(yi^beta-
(ksi*yi)^beta)+(t_end^beta-(ksi*yi)^beta))); %% expected number of
failures (Equation III-6)

%% use LHS to sample failure timestamps based Nr_withpm
for j=1:Nr_withpm
      P_LHS(j)=F_withpm(1)+(1-F_withpm(1))/Nr_withpm*(rand+(j-1));%
      sampled cumulative failure probability for n_th failure (Equation
      III-15)
end
t_life_inv=interp1(F_withpm,t_life,P_LHS,'linear');% calculate
timestamps of failures based on sampled cumulative failure probability
t_life_inv(isnan(t_life_inv))=4.6*10^4;% represent all the failure
timestamps beyond the 5-year range
failtimelist_sort= t_life_inv.*3600; %convert unit of timestamps of
failures from hours to seconds
```

APPENDIX B

MATLAB FUNCTIONS IN THE HYBRID SIMULATOR

B.1 Matlab Functions in Figure III-6

1. Matlab function in the block “Execution”

```
function Exe_weights = weights_of_E(CPC)
% weigthts from CREAM model
e_weights = ...
    [0.8 1.0 1.2 2.0;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 5.0;...
     0.8 1.0 2.0 NaN;...
     1.0 1.0 2.0 NaN;...
     0.5 1.0 5.0 NaN;...
     1.0 1.2 NaN NaN;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 5.0];
% calculate product of weights
Exe_weights = e_weights(1,CPC(1))*...
                   e_weights(2,CPC(2))*...
                   e_weights(3,CPC(3))*...
                   e_weights(4,CPC(4))*...
                   e_weights(5,CPC(5))*...
                   e_weights(6,CPC(6))*...
                   e_weights(7,CPC(7))*...
                   e_weights(8,CPC(8))*...
                   e_weights(9,CPC(9));
```

2. Matlab function in the block “Interpretation”

```
function Int_weights = weights_of_I(CPC)
% weigthts from CREAM model
i_weights = ...
    [1.0 1.0 1.0 1.0;...
     0.8 1.0 2.0 NaN;...
     1.0 1.0 1.0 1.0;...
     1.0 1.0 1.0 NaN;...
     1.0 1.0 2.0 NaN;...
     0.5 1.0 5.0 NaN;...
     1.0 1.2 NaN NaN;...
     0.5 1.0 5.0 NaN;...
     0.5 1.0 1.0 2.0];
% calculate product of weights
Int_weights = i_weights(1,CPC(1))*...
                   i_weights(2,CPC(2))*...
                   i_weights(3,CPC(3))*...
                   i_weights(4,CPC(4))*...
                   i_weights(5,CPC(5))*...
                   i_weights(6,CPC(6))*...
                   i_weights(7,CPC(7))*...
                   i_weights(8,CPC(8))*...
                   i_weights(9,CPC(9));
```

3. Matlab function in the block “Observation”

```
function Obs_weights = weights_of_O(CPC)
% weigthts from CREAM model
o_weights = ...
    [1.0 1.0 1.0 1.0;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 5.0;...
     0.8 1.0 2.0 NaN;...
     1.0 1.0 2.0 NaN;...
     0.5 1.0 5.0 NaN;...
     1.0 1.2 NaN NaN;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 2.0];
% calculate product of weights
Obs_weights = o_weights(1,CPC(1))*...
                   o_weights(2,CPC(2))*...
                   o_weights(3,CPC(3))*...
                   o_weights(4,CPC(4))*...
                   o_weights(5,CPC(5))*...
                   o_weights(6,CPC(6))*...
                   o_weights(7,CPC(7))*...
                   o_weights(8,CPC(8))*...
                   o_weights(9,CPC(9));
```

4. Matlab function in the block “E4 Occurrence”

```
function E4_error = E4_error_occurence(Exe_weights)
coder.extrinsic('E4_error_occur') %E4_error_occur is an extrinsic
function
E4_error=0; %to initialize
E4_error = E4_error_occur(Exe_weights);
end
```

5. Extrinsic function “E4_error_occur”

```
function error = E4_error_occur(Exe_weights)
NP_E4=0.003; % nominal probability of error mode E4
error_prob_E4 = Exe_weights*NP_E4;
if error_prob_E4 >0 && error_prob_E4<1
   error_prob_E4 = error_prob_E4;
elseif error_prob_E4 <=0
       error_prob_E4 =0.000001;
else
    error_prob_E4=0.999999;
```

```
end
error = randsample((0:1),1,true,[1-error_prob_E4, error_prob_E4]);
end
```

6. Matlab function in the block “Sample Cm0_cp”

```
function Cm0_cp= sample_Cm0(E4_error)
coder.extrinsic('sampleCm0') %sampleCm0 is an extrinsic function
Cm0_cp=0; %to initialize
Cm0_cp=sampleCm0(E4_error);
end
```

7. Extrinsic function “sampleCm0”

```
function Cm0 = sampleCm0(error)
if error == 0
    Cm0=3.66;
else
    %sample possible initial monomer concentration with uniform
    %distribution
    possibleCm=[3.20:0.01:3.65,3.67:0.01:4.11];
    Cm0=randsample(possibleCm,1,true);
 End
```

8. Matlab function in the block “Modified Probability”

```
function [prob_I1,prob_O3] = I1O3_prob(Int_weights,Obs_weights)
NP_I1=0.2; %nominal probability of error mode I1
NP_O3=0.07; %nominal probability of error mode O3
prob_I1 = Int_weights*NP_I1; % modified probability of I1
prob_O3 = Obs_weights*NP_O3; % modified probability of O3
    if prob_I1 >0 && prob_I1<1
       prob_I1 = prob_I1;
    elseif prob_I1<=0
        prob_I1=0.000001;
    else
        prob_I1=0.999999;
    end

    if prob_O3 >0 && prob_O3<1
       prob_O3 = prob_O3;
    elseif prob_O3<=0
        prob_O3 = 0.000001;
    else
       prob_O3=0.999999;
    end
end
```

9. Matlab function in the block “I1O3_error mode”

```
function I1O3_errormode = error_mode(prob_I1,prob_O3,E1_error)
coder.extrinsic('I1O3_error') %I1O3_error is an extrinsic function
coder.extrinsic('O3_error') %O3_error is an extrinsic function
I1O3_errormode = 0; %to initialize
if E1_error==1
   I1O3_errormode = I1O3_error(prob_I1,prob_O3);
else
   I1O3_errormode = O3_error(prob_O3);
end
end
```

10. Extrinsic function “I1O3_error”

```
function I1O3_errormode = I1O3_error(prob_I1, prob_O3)
% prob_I1 represents occurrence probability of error mode I1
% prob_O3 represents occurrence probability of error mode O3
% prob_I1O3 represents the probability when none of I1 and O3 occurs
probI1O3=1-prob_I1-prob_O3;
if probI1O3 > 0 && probI1O3 < 1
    probI1O3 = probI1O3;
elseif probI1O3<=0
    probI1O3=0.000001;
else
    probI1O3=0.999999;
end
probvector = [prob_I1, prob_O3, probI1O3];
I1O3_errormode = randsample([11,13,10],1, true, probvector);
```

11. Extrinsic function “O3_error”

```
function O3_error = O3_error(prob_O3)
O3_error = randsample((0:1),1,true,[1-prob_O3 , prob_O3 ]);
End
```

B.2 Matlab Functions in Figure III-7

1. Matlab function in the block “Weight calculation”

```
function Exe_weights_field = weights_of_E_field(CPC2)
e_weights_field = ...
    [0.8 1.0 1.2 2.0;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 5.0;...
     0.8 1.0 2.0 NaN;...
     1.0 1.0 2.0 NaN;...
     0.5 1.0 5.0 NaN;...
     1.0 1.2 NaN NaN;...
     0.8 1.0 2.0 NaN;...
     0.5 1.0 1.0 5.0];
Exe_weights_field = e_weights_field(1,CPC2(1))*...
                    e_weights_field(2,CPC2(2))*...
                    e_weights_field(3,CPC2(3))*...
                    e_weights_field(4,CPC2(4))*...
                    e_weights_field(5,CPC2(5))*...
                    e_weights_field(6,CPC2(6))*...
                    e_weights_field(7,CPC2(7))*...
                    e_weights_field(8,CPC2(8))*...
                    e_weights_field(9,CPC2(9));
```

2. Matlab function in the block “Modified Probability”

```
function prob_E1_field = E_prob(Exe_weights_field)
E1_error=0; %to initialize
NP_E1=0.003; % nominal probability of error mode E1
prob_E1_field = Exe_weights_field*NP_E1;
    if prob_E1_field >0 && prob_E1_field<1
       prob_E1_field = prob_E1_field;
    elseif prob_E1_field <=0
        prob_E1_field = 0.000001;
    else
       prob_E1_field = 0.999999;
    end
end
```

3. Matlab function in the block “Sample m$_{i0}$”

```
function mass_i0 = sample_m_i0(prob_E1_field)
coder.extrinsic('sample_m_i0') % sample_m_i0 is an extrinsic function
mass_i0=0; %to initialize
mass_i0=sample_m_i0(prob_E1_field);
end
```

4. Extrinsic function “sample_m_i0”

```
function mass_i0 = sample_m_i0(E1_field)
probvector = [E1_field,(1-E1_field)];
field_E_error = randsample([1,0],1,true,probvector);
if field_E_error ==1
   possible_mass_i0= [705:1:741,743:1:780]; %sample from uniform
distribution
   mass_i0 = randsample(possible_mass_i0,1,true);
else
    mass_i0=742;
end
```

B.3 Matlab Functions in Figure III-9

1. Matlab function in the block “calc_Ci0”

```
function Ci0 = calc_Ci0(mass_i0,MW_i, V0)
% calculate initial concentration of initiator
Ci0=mass_i0/V0/MW_i;
```

2. Matlab function in the block “initial_conc”

```
function [Cs0,c,m]= initialconc(T0,Cm0,Ci0,cp_m,cp_s, MW_i, MW_m, MW_s,
V0)
rou_m=915.1;
rou_s=842;
rou_i=915;
% calculate initial concentration of solvent
Cs0=(V0-Cm0*V0*MW_m/rou_m)*rou_s/MW_s/V0;
% calculate mass of reactants
mass_s=Cs0*V0*MW_s;
mass_i=Ci0*V0*MW_i;
mass_m=Cm0*V0*MW_m;
m=mass_s+mass_i+mass_m;
% calculate heat capacity of reactants
cp_i=1.449;
cp_m = 1.6736;
cp_s = 2.2384;
c=(cp_m*mass_m+cp_s*mass_s+cp_i*mass_i)/(mass_i+mass_m+mass_s);
```

3. Matlab function in the block “optimal_temp”

```
function T_opt= optimal_temp(time_s_clock)
time_s=time_s_clock-2; %time_s is how long has process been started
%fitted profile of reactant temperature(Wright & Kravaris, 1997)
if time_s>=0 && time_s<=0.4914*3600
   T_opt = 71.616*(time_s/3600)+295.36;
 else
   T_opt= 0.0033*(time_s/3600)^6 - 0.0755*(time_s/3600)^5 +
0.6633*(time_s/3600)^4 - 2.8532*(time_s/3600)^3 +
6.7167*(time_s/3600)^2 - 10.108*(time_s/3600) + 334.2;
end
end
```

4. Matlab function in the block “process” (Soroush & Kravaris, 1992, 1993)

```
function [dCmdt,dCidt,dTdt,dTjdt] = process(T0,Cm0,Cs0,Ci,Cm,T_j,MW_m,
MW_s,R,Z_p0, Z_theta_p,Z_fm, E_p0, E_fm, Z_i, E_i,Z_t0, E_t0,rou_p,
Z_theta_t, f, E_theta_t, E_theta_p, A_area0, delta_Hp,a_coeff, b_coeff,
c_coeff, T,U0,Vj,c_water,rou_j, T_gp, B, T_cw, u_modified, Ci0, rou_i,
MW_i, V0, m, c)
    rou_m=9.151*10^2; % density of monomer
    rou_s=8.420*10^2; % density of solvent
    epslon= Cm0*MW_m/rou_m*(rou_m/rou_p-1); %volumn expansion factor
    A=0.168-(8.21*10^(-6))*(T-T_gp)^2; % temperature-dependent
parameter in gel effect
    V=(1+epslon)*Cm0*V0/(Cm0+epslon*Cm); % volumn of reacting mixture
    Cs=Cs0*V0/V; % concentration of solvent
    k_i=Z_i*exp(-E_i/(R*T));% reaction rate constant for initiation
reaction
    k_t0= Z_t0*exp(-E_t0/(R*T)); % reaction rate constant for
termination at zero conversion
    k_theta_t=Ci0*Z_theta_t*exp(-E_theta_t/(R*T)); %temperature and
initiator loading concentration dependent parameter
    k_p0= Z_p0*exp(-E_p0/(R*T)); %  reaction rate constant for
propagation at zero conversion
    k_theta_p= Z_theta_p*exp(-E_theta_p/(R*T)); %temperature-dependent
parameter in gel effect model
    k_fm= Z_fm*exp(-E_fm/(R*T)); % reaction rate constant for chain
transfer to monomer
    phi_p=(MW_m/(1+epslon)*(Cm0-Cm)/rou_p)/[MW_m/(1+epslon)*(Cm0-
Cm)/rou_p+Cm*MW_m/rou_m+Cs*MW_s/rou_s];%volume fraction of dead polymer
    D=exp(2.3*(1-phi_p)/(A+B*(1-phi_p))); % dead polymer chain

ksi_0=((2*f*k_i*Ci*k_t0)/(D*k_theta_t)+sqrt(abs((2*f*k_i*Ci*k_t0/D/k_th
eta_t)^2+4*k_t0*2*f*k_i*Ci)))/(2*k_t0);% molar concentrations of live
polymer chains
    k_t=k_t0/(1+k_t0*ksi_0/k_theta_t/D); % reaction rate constant for
termination
    x_m=(Cm0*V0-Cm*V)/(Cm0*V0); % monomer fraction
    A_area=A_area0*(1+epslon*x_m); % effective heat-transfer area of
reactor
    k_p=k_p0/(1+k_p0*ksi_0/k_theta_p/D);% reaction rate constant for
propagation
    % mass balance
    dCmdt=(1+epslon*Cm/Cm0)*(-Cm*ksi_0)*(k_p+k_fm);
    dCidt=-1*k_i*Ci+epslon*Ci/Cm0*(-Cm*ksi_0)*(k_p+k_fm);
    % heat transfer coefficient
    U=U0*[a_coeff+(1-a_coeff)*exp(-b_coeff*((x_m)^c_coeff))]; % no
fouling
    % U=0.8* U0*[a_coeff+(1-a_coeff)*exp(-b_coeff*((x_m)^c_coeff))];
with fouling
    % energy balance
    dTdt=(-delta_Hp)*k_p*ksi_0*Cm*V/(c*m)+U*A_area*(T_j-T)/(c*m);
    dTjdt=U*A_area*(T-
T_j)/(Vj*rou_j*c_water)+u_modified/(c_water*Vj*rou_j);
end
```

5. Matlab function in the block “coordination”

```
function [u,u_modified,Power,Fcw]  =
cordination(T_j,T_isetpoint,c_water, rou_j,T_cw, Fcwmax, Power_max)
% coordination rules of heating/cooling control system (Soroush &
Kravaris, 1992)
u=(T_isetpoint-T_j)*c_water*44.13; % energy that is need to be
generated/removed by heating/cooling control system
if u>=0 && u< Power_max
    Power=u;
elseif u>=Power_max
    Power=Power_max;
else
    Power=0;
end

if u<0 && u>=-Fcwmax*c_water*rou_j*(T_j-T_cw)
    Fcw= -u/[c_water*rou_j*(T_j-T_cw)];
elseif  u<-Fcwmax*c_water*rou_j*(T_j-T_cw)
    Fcw= Fcwmax;
 else
    Fcw=0;
end
u_modified=Power-Fcw*c_water*rou_j*(T_j-T_cw);% actual energy
generated/removed by heating/cooling control system
```

APPENDIX C

INPUT PARAMTERS FOR SIMULATION

Parameter values are from (Baillagou & Soong, 1985a, 1985b; Soroush & Kravaris, 1992, 1993; Wright & Kravaris, 1997)

```
A_area0 = 52.95; % reactor volumn (m^3)
B = 0.03; %constant parameter in gel effect
E_fm = 74479; % activation energy for chain transfer to monomer
(kJ/kgmol)
E_i = 128770; % activation energy for initiation (kJ/kgmol)
E_p0 = 18283; % activation energy for propagation at zero conversion
(kJ/kgmol)
E_t0 = 2944.2; % activation energy for termination at zero conversion
(kJ/kgmol)
E_theta_p = 117000; % activation energy for the parameter k_theta_p
(kJ/kgmol)
E_theta_t = 145840; % activation energy for the parameter k_theta_t
(kJ/kgmol)
MW_i = 164.21; % molecular weight of initiator (kg/kgmol)
MW_m = 100.12; % molecular weight of monomer (kg/kgmol)
MW_s = 92.14; % molecular weight of solvent (kg/kgmol)
R = 8.314; % ideal gas constant (kJ/K/kgmol)
T_cw = 288; % cooling water temperature (K)
T_gp = 387.2; %glass transition temperature of PMMA (K)
U0 = 0.4543; % initial overall heat transfer coefficient (kJ/m^2/s/K)
V0 = 37.09; % volume of initial reactant mixture (m^3)
Vj = 3.405; % volume of reactor jacket (m^3)
```

```
Z_fm = 4.661E+9; % frequency factors for chain transfer to monomer reactions (m^3/kgmol/s)
Z_i = 1.0532999999999999E+15; %frequency factor for initiation reactions (1/s)
Z_p0 = 491669.99999999994;% frequency factor for the reaction rate constant k_theta_p0 (m^3/kgmol/s)
Z_t0 = 9.8E+7; % frequency factor for the reaction rate constant k_t0 (m^3/kgmol/s)
Z_theta_p = 3.0233E+13; % frequency factor for the reaction rate constant k_theta_p (m^3/kgmol/s)
Z_theta_t = 1.454E+20; % frequency factor for the reaction rate constant k_theta_t (m^3/kgmol/s)
delta_Hp = - 57800; % heat of propagation reaction (kJ/kgmol)
f = 0.58; % initiator efficiency
m0 = 3405; % mass in jacket (kg)
% coefficients to calculate overall heat-transfer coefficient
a_coeff = 0.2;
b_coeff = 7;
c_coeff = 3;
% heat capacity (kJ/kg/K)
c_water = 4.19;
cp_m = 1.6736;
cp_s = 2.2384;
cp_i = 1.449
%density (kg/m^3)
rou_m= 915.1;
rou_s= 842;
```

```
rou_i = 915;

rou_j = 1000;

rou_p = 1200;
```