Summer 2022

Automated Contingency Management for Water Recycling System

Shijie Tang

Automated Contingency Management for Water Recycling System

by

Shijie Tang

Bachelor of Science
Yanshan University, 2016

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2022

Accepted by:

Bin Zhang, Major Professor

Xiaofeng Wang, Committee Member

Guoan Wang, Committee Member

Lingyu Yu, Committee Member

Tracey L. Weldon, Vice Provost and Dean of the Graduate School

Dedication

This dissertation dedicates to my parents, for their love and support all the time.

Acknowledgments

Four years of Ph.D. study is a very valuable experience in my life. I want to express thanks to all the people I met and helped me during my study.

Foremost, I want to express my sincere thanks to my advisor Dr. Bin Zhang. His innovative ideas in research and cautious attitude in writing inspire me most. He always read the documents that I write patiently and make very detailed comments. And he guides me to make a precise main research objective and keep me on the right track when I feel distracted at other objectives. The time and effort he put into my Ph.D. study not only benefits this dissertation but also for the success of my future career.

Second, I would also like to acknowledge my committee members Dr. Xiaofeng Wang, Dr. Guoan Wang, Dr. Lingyu Yu, for their time and efforts as my Ph.D. committee and their comments and suggestions in the proposal defense.

Thanks to all my group members: Wuzhao Yan who helped me understand the Lebesgue sampling theory and Bayesian-based diagnosis. Zhichao Liu who helped me a lot in the control system model design, Guangxing Niu who helped me in understanding the data-driven based method. Enhui Liu and Xuan Wang who helped me validate the WRS model.

Thanks to all my office mates: Zheqing Zhou, Tengxing Wang, Wei Jiang, Lixing Yang, Jinqun Ge for the friendly working environment, and meaningful discussions that we had.

Thanks to all professors who had taught me and helped me: Dr. Bin Zhang, Dr. Xiaofeng Wang, Dr. Guoan Wang, Dr. Herbert L. Ginn, Dr. Enrico Santi. Thanks to all the department staff for their assistance during my study.

Thanks to all the friends I met here for cheering me up when I was struggling with research and future career. I can never forget the encouragement you give me and the fun activities we had together.

Finally, I would like to express my sincere gratitude to my family for their love and support.

Abstract

NASA's exploration program envisions the utilization of a Deep Space Habitat (DSH) for human exploration of the space environment in the vicinity of Mars and beyond. Communication latency and extreme limitations of power and life-supporting resources make it imperative to operate the DSH systems in a highly autonomous fashion. One such system is the Environmental Control and Life Support System (ECLSS) which needs to be monitored and optimized to support its designated missions.

Integrated System Health Management (ISHM) technologies have been developed in the past decades to provide a real-time assessment of system health and use this information to improve system safety and reduce operation cost. The goal of this study is to apply the ISHM principle to ECLSS, and bring in innovative solutions to address several key elements that NASA has been soliciting in its recent research announcements. To achieve this goal, Water Recycling System (WRS) deployed at NASA Ames Research Center's Sustainability Base is selected as a testbed. Efforts are then made to develop a physics-based model with high fidelity fault simulations and data-driven model enhancements. With the system in place, an Automatic Contingency Management (ACM) system concept is introduced to embrace the fault diagnosis, prognosis, and automated fault accommodation through control reconfiguration and optimization, thus contingencies in the system are managed automatically without human in the loop. The ACM system architecture and key enabling techniques are developed. The fault diagnosis and prognosis task are accomplished by a novel method combining Lebesgue sampling (LS) and

unscented Kalman filter (UKF) that allows computing resources to be allocated efficiently based on system health conditions without sacrificing performance. Multi-Stage PID and Time-Varying Weight Model Predictive Control (MPC) based fault mitigation strategies are designed for automatic fault accommodation which allows the integration of diagnostics and prognostics in the control strategy. Special consideration is given to distinguish sensor faults from system component faults since faulty sensor information could confuse the ACM system if not addressed sufficiently. Overall, the development and enhancements of WRS modeling, the ACM system architecture for the WRS, as well as the novel fault diagnosis and control algorithms constitute the major contributions from this study.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The Deep Space Habitat (DSH) is a series of concepts by NASA that would be used to support crewed exploration missions to the Moon, asteroids, and eventually Mars. Early preliminary concepts are composed of International Space Station-derived hardware, the Orion crew capsule, and various support craft. Developing a deep space habitat would allow a crew to live and work safely in space for about one year on missions to explore the space. The environmental control and life support system (ECLSS) supply air, water, and food, as well as to provide shielding against harmful external influences such as radiation and micro-meteorites.

Since the ECLSS system and its components are life-critical, they are designed and constructed using safety engineering techniques. Moreover, the system needs to be monitored closely to handle unexpected contingencies to maintain its functionality in the presence of faults [1–5].

Traditionally, planning and health management of space operations are conducted locally on the earth. This is subject to a large communication latency and limits real-time inputs from crew members during space missions. By migrating operational support from the Earth to the habitat in space, crew members can be allowed to manage, plan, and operate much of the purpose themselves in real-time with first-hand information, furthermore, if the operational support system can be made intelligent and autonomous, the crew member can be alleviated from such

maintenance burden and focus their precious time on scientific research work. This motivates the Automatic Contingency Management concept which is further detailed in this dissertation.

Due to the criticality of the ECLSS for space exploration missions, NASA has solicited, in several recent Research Announcements, Model-based Systems Engineering approaches for failure mode analysis, fault diagnosis, prognosis, and automated fault accommodation strategies. The development of these technologies can certainly leverage a great deal from the Integrated Vehicle Health Management (IVHM) techniques that were developed in the past three decades, as well as from the broader PHM community across industries. However, these approaches have to be improved to address the particular challenges in harsh DSH environments with extreme resource limitations. This motivates the development of innovative fault diagnosis, prognosis, and accommodation methods that are optimized for deployment in deep space habitat environments.

1.2 Literature Review

The development of the Automatic Contingency Management system (ACM) for ECLSS requires innovative technologies for fault diagnosis, prognosis, and fault-tolerant control methods that are optimized for deep space applications. A literature review is conducted to understand the state of the art in these areas and to identify candidate algorithms for this study. Previous work done on ECLSS modeling, controls, and operation optimization is also studied and included in this dissertation.

1.2.1 Introduction to ACM

The goal of ACM is to enhance the reliability of the system during harsh conditions and component fault. ACM diagnosis fault information and utilize fault information to optimal control strategy to mitigate the effect of fault. ACM is a software

framework, which main modules are anomaly detection, diagnosis/prognosis, and control. ACM has been adapted to Simulink models like the propulsion system and unmanned aerial vehicle (UAV)., etc. In [6], a Simulink/Stateflow based ACM isolate valve stuck the fault in the propulsion system, the Stateflow model automated choose optimal strategy based on the fault type, In [7], a proactive prognostic enhanced ACM is proposed to mitigate the effect of actuator faults in UAV based on both current health state and future health state. The proposed ACM framework will be introduced in Chapter 2.

1.2.2 Current Research on ECLSS

1.2.2.1 Current Research on ECLSS Modeling

The ECLSS's reliability and survivability are critical to NASA especially in long-term human space exploration missions. The health management of life support systems has to cover several critical components such as the power unit, waste processing system, water recovery system, and biomass processing system, etc.. Under the recent rapid growth of space commercialization efforts, many environment control and life support simulation models have been developed.

The investigated models include BioSim [8] developed by TracLabs and commissioned by NASA Johnson Space Center (JSC). BioSim simulates an advanced life support system in which many resources are reused or regenerated. The model integrates several major components, such as air, biomass, food, thermal, waste, and water. This model is designed for long-duration human missions where resupply systems are tightly integrated and computation resources are very limited. The model can be used as a platform for the development of control methods. V-HAB, led by Space Exploration Lab in the Technical University of Munich, simulates a system-level dynamic of the mission and crew for the Life Support System (LSS). The innovation of the V-HAB, compared with BioSim, is that it simulates crew

activities using the human model and crew controller. The human model predicts the physiological response from the human body, which can be associate with the crew member's schedule and task assignment.

HabNet [9] is another simulation platform that was developed in the MIT Strategic Engineering lab. It is a MATLAB simulation model based on BioSim. HabNet quantitatively evaluates many technology options for a proposed mission architecture regarding their functional performance, failure modes, supportability requirements, initial deployment, and operational cost. The innovation of this model, compared with BioSim and V-HAB, is that HabNet not only focuses on the LSS and crew model but also includes many technology modules such as Water Processor Assembly (WPA), Urine Processor Assembly (UPA), Oxygen Generation Assembly (OGA), and Carbon Reduction Assembly (CRA). All of these models have achieved a certain level of fidelity (from medium to high) and have been validated by real data from ISS! Live! website [https://isslive.com/displays]. However, most of these models are system-level models focusing on the producer-costumer relationship instead of the dynamics of the system. To address these limitations, one of the work in this project is to develop a high fidelity model that focuses on the system dynamics for diagnosis and control. Since the scope of ECLSS covers a wide range of subsystems, this project focuses on the modeling of the water recycling system (WRS) and uses this model as a platform for diagnosis, control, and automated contingency management development.

1.2.2.2 Current Studies on ECLSS Health Management

There are reported works on using data-driven approaches to analyze data from NASA Ames Sustainability Base for health monitoring purposes. Studies have shown that weather effects have a significant effect on the power production of the photovoltaic (PV), to monitor the power production for different weather conditions,

an artificial neural network is trained by using the on-site sensor data from the PV system [10]. Instead of only monitoring the power generation based on weather conditions, it is also important to monitor the system operation in the presence of faults. To detect abnormal events in the NASA Ames Sustainability Base building, a MATLAB adverse condition detection algorithm called ACCEPT [11] is developed. ACCEPT integrates dimensionality reduction and Bayesian network to detect the thermal discomforts of occupants. In the area of Fault identification and parameter identification, a distance-based anomaly detection method [12] is used to monitor parameter values in the space applications including International Space Station flight control, satellite vehicle system health management, launch vehicle ground operations. Besides data-driven based methods, there are also several model-based diagnoses and prognosis approaches designed for the ECLSS. To simulate the WRS and study approaches for fault diagnosis, a hydraulic system model and diagnosis and prognosis methods are developed for the WRS deployed at NASA Ames Research Center's Sustainability Base in [13]. However, fault accommodation strategies are not studied in the reported work.

1.2.3 Fault Diagnosis

1.2.3.1 Fault Diagnosis Concepts

Undetected faults can cause severe system performance degradation that could jeopardize the designated mission. A fault is different from a failure in that the affected system does not totally lose its functionality, i.e. system can still operate at a deteriorated condition. In many cases, it indicates an unexpected change of system dynamics or continuous degradation of the component performance. Examples are clogging of filter, degradation of motor insulation [14], etc. If these faults are not detected and maintained in a timely manner, they could lead to severe system

degradation resulting in loss of critical function, failed mission and even injuries and casualties.

Fault diagnosis consists of fault detection, fault isolation, and fault identification. Fault detection refers to the detection of an anomaly in system functionality or performance by observing measured system outputs. This process often involves comparing measured parameters to the prediction/estimation from a nominal model to determine noticeable deviations. Other methods include combining empirical fault dynamic models constructed from data-driven or machine learning methods with real-time measurement. Fault detection only recognizes if a fault occurs, but does not locate the root cause. Once a fault is detected based on observed anomalous system behavior, it is important to isolate the faulty component/subsystem in order to fix and accommodate it. This is a fault isolation problem. In a complex system like the WRS, the effect of a fault in a certain component may propagate in the system, causing confusion and ambiguity in fault isolation. Isolability of faults usually depends on the observability of the system, which means more sensors are needed to isolate the fault modes. When a fault is detected and isolated, it is often useful to determine the severity (or magnitude) of the fault in order to accommodate it with proper control reconfiguration. The estimation of fault magnitude is often referred to a fault identification problem. For example, as water flows through the filter, certain particles will be trapped, causing filter deterioration and eventually clogging. Fault identification aims to determine the severity of filter clogging.

In general, the methods of fault diagnosis can be divided into three categories: model-based, data-driven, and hybrid approaches [15]. These approaches have their own advantages and disadvantages. In these approaches, model-based approaches require extensive domain knowledge to build the system model, and it is often time-consuming. On the other hand, data-driven approaches requires statistically

sufficient data under different operating conditions and different fault modes. Detailed reviews of all three categories follow.

1.2.3.2 Data Preparation and Data Driven Diagnosis

Data-driven approaches can be further divided into supervised learning methods (classification and regression) and unsupervised learning methods (clustering).

Since the data collected from system often contain noise, outliers, or other forms of errors and duplications, they typically need to be preprocessed. Some widely used data preprocessing methods include de-noising, signal enhancement, missing value imputation [16], categorical feature encoding [17], dimension reduction [18]. In order to extract hidden factors in the dataset, dimension reduction methods such as principal component analysis (PCA) [19] and linear discriminant analysis (LDA) [20] can be used. PCA converts observations of possibly correlated variables into a set of linearly uncorrelated variables. LDA uses a linear combination of features to characterizes two or more classes. Multi-scale principal component analysis (MSPCA) extracts the correlation by decomposing the individual variables into wavelet approximation, which is able to detect and identify faults earlier [21].

There are a variety of data-driven models for fault diagnosis. An artificial neural network (ANN) [22] is an information processing algorithm inspired by human nervous systems that can be used in pattern recognition and classification problems. The process of ANN can be divided into the training phase and testing phase. The training phase uses labeled data to train weights of the network while the testing phase uses testing dataset to verify the accuracy of the network. There have been a lot of applications of ANN for fault diagnosis. For example, in [23], a neural work model is applied together with a discrete wavelet transform method to analyze gearbox vibration data and diagnose different fault conditions accordingly.

Deep Learning is a subfield of machine learning algorithms inspired by the structure of artificial neural networks [24]. It provides efficient learning algorithms for large neural networks with numerous (hundreds or thousands of) hidden layers. Deep learning can be applied to several different network architectures such as deep neural networks, deep belief networks [25], recurrent neural networks [26], and convolutional neural networks [27], long short-term memory network [28]. Deep learning approaches are capable of modeling complex systems without much domain knowledge if sufficient data is provided. However, in the case of the deep space habitat applications where data (particularly failure data) is rather limited, deep learning's applicability is rather limited at the early phase of the development.

1.2.3.3 Model Based Diagnosis

Model based methods compare observed behavior (through measurements) with modeled behavior to detect and isolate faults. In such approaches, the dynamics of the system and its components are often described using dynamic models.

Causal model based methodologies considers the system as a combination of correlated dependent variables [29]. Those approaches have been used to diagnosis intermittent faults (IF) in the system, such as pump or valve stuck. In [30], a simple first-order system to construct the system and linear functions is used to show the relationship between components. The fault mode is the component stuck. Based on it, discrepancies between the observed and expected system behaviors are used to identify conflicts first (conflict recognition). When there are more components in the system, there are more conflicts. Then system-level reasoning is proposed to reduce the number of conflicts by comparing the failure probability of each candidate.

1.2.3.4 Bayesian Based Diagnosis

In order to combine the benefits of domain knowledge (as represented by the models) and the information from data (measurements), Bayesian-based diagnosis methods are introduced in many model-based applications. Bayesian approach is a recursive algorithm that estimates the true state of a system based on noisy measurement. It involves two steps, namely prediction and correction. The prediction step estimates the state value based on the dynamic model and the previous state estimate. This step generates a priori distribution of the state variable which is then adjusted by the correction step based on the actual measurement [31]. This step generates a posterior distribution of the fault state. Many Bayesian estimation methods have been investigated for diagnosis and prognosis, such as the Kalman Filter (KF) [32], Extended Kalman Filter (EKF) [33], Unscented Kalman Filter (UKF) [34], and Particle Filter (PF) [35]. The comparison between these approaches is shown in Table 1.1.

Table 1.1 Evaluation between different state estimation methods

State estimation	Model	Noise distribution	Computational effort
KF	Linear	Gaussian	Low
EKF	Locally Linear	Gaussian	Medium
UKF	Nonlinear	Gaussian	Medium
PF	Nonlinear	Non-Gaussian	High

KF produces an optimal solution for a linear system with Gaussian noise. Although KF provides optimal state estimates for linear systems, its real world applications are rather limited since most systems are essentially nonlinear. To address nonlinear systems, EKF is introduced, which linearizes the state model using Taylor expansion [36]. EKF has a lot of application in diagnosis. For example, In [37], EKF and autoregressive exogenous (ARX) models were used for the diagnosis of faults in heating, ventilating, and air-conditioning (HVAC) systems. However, EKF

algorithms are usually computational expensive since they try to linearize the system equations by taking the Jacobians. It is also difficult to apply machine learning models such as Neural Networks due to the difficulties associated with the Jacobian calculation for black-box models. To solve this problem, the UKF algorithm is introduced. Unlike the KF algorithm, which only evolves the mean and variance of state variables, the UKF uses a set of sigma points to approximate the distribution of state variables. The UKF method does not require taking Jacobians which makes it a suitable solution to combine with the data-driven, black-box models such as Neural Networks. For highly nonlinear fault dynamics with non-Gaussian noises, particle filter (PF) is introduced [38]. However, due to the use of a large amount of "particles" (often hundreds or thousands), the PF algorithms typically have a high demand on computation which is not suitable for deep space applications where the computation resources are very limited.

1.2.4 Fault Tolerant Control

Fault-Tolerant Control System (FTCS) can be classified into two types: passive (PFTCS) and active (AFTCS) [31]. In PFTCS, the controller is designed in a way that it can tolerate presumed faults by sacrificing performance. An example is robust control design. In such a design, control parameters are set such that, when a fault happens, the robust control can accommodate the change of system dynamics caused by the fault. In other words, it can maintain system stability and performance. However, since the robust control needs to work in a wide range (with and without faults), the control will not work in the optimal state in this wide range. On the contrary, AFTCS designs the controller in a way that it can monitor the fault state and adjust the control strategy accordingly. In such a scheme, the controller is first optimally designed for the healthy system. When a fault happens, the fault is detected and fault severity is evaluated. This information is then used to reconfigure

the control structure or parameters so that the control still maintains the optimal operation even with the presence of faults. Although AFTCS has obvious advantages over PFTCS, it needs sensing systems and reliable diagnostic algorithm to provide real-time fault information. Examples of AFTCS include Linear Quadratic (LQ) [39], Model Predictive Control (MPC) [40], Adaptive Control [41], etc.

A proportional–integral–derivative controller (PID controller or three-term controller) is a classic control mechanism, which has been used in many industrial applications. A PID controller uses the error value between the desired setpoint (SP) and a measured process variable as a correction based on proportional, integral, and derivative gains (denoted as P, I, and D respectively). While a PID controller is capable of tracking the system reference signal, from a system health management perspective, it lacks a systematic means to handle system constraints imposed by not only hardware limits, but also system faulty behaviors.

Dynamic Programming (DP) is an optimization method that breaks the problem into subproblems, which are locally optimized, then the optimization results are combined with a global optimization solution [42]. In [43], a boundary line DP that has better efficiency is proposed. The author uses the model of Lotka-Volterra fish population and hybrid electric vehicle problem to justify the efficiency of this approach. The limitation of DP is that it seeks a global optimization solution, which will request a lot of computation efforts.

In order to realize real time implementation on deep space habitat, MPC that seeks a local optimization is introduced. MPC is an advanced control method that can be used to control a process while satisfying specified system constraints. It takes the current state of the plant as the initial states and defines a finite horizon open-loop optimization problem. Optimization produces an optimal control sequence by minimizing the cost function, in which the first value in the sequence is used as the input to the system and the rest are discarded. An essential advantage of MPC

is its ability to generate optimal solutions with constraints on control inputs and states. In some applications with crucial constraints, MPC has been widely used for its efficiency reaching operating points or working under predefined state and input constraints. MPC is based on iterative a finite-horizon optimization of a plant model [44]. In [45], Moving Horizon Estimation (MHE) and Unscented Kalman Filter (UKF) are utilized for parameter estimation in a Quad-Rotor Helicopter application. The estimated fault information is then used to construct an AFTCS control strategy to accommodate the fault.

For the control of the deep space habitat systems where the components are costly and replacing them are labor-intensive, strategies that could optimize the usage of components during the mission based on health information are highly sought after. The MPC framework, with the potential to incorporate diagnostics and prognostics in its objective function and/or constraints, lends itself a candidate approach to deep space habitat applications.

1.3 Research Objectives and Contributions

This work aims to support NASA's deep space habitat development by providing innovative solutions for managing and optimizing the operation of Environmental Control and Life Support Systems (ECLSS). The ECLSS recycles air and water to the International Space Station (ISS) crew by using major components like the Oxygen Generation System (OGS) and WRS. To ensure the normal operation of the ECLSS in harsh deep space environments, such as on Mars, a highly resilient health management system is needed to quickly detect/isolate system faults and promptly reconfigure/react to accommodate the faults. For this purpose, the concept of Automatic Contingency Management (ACM) is introduced. The ACM system consists of a suite of techniques for fault detection, isolation, and prognostics, as well as the fault accommodating control strategies. Similar concepts and methods

have been developed for other applications, such as the concept of Integrated Vehicle Health Management systems for aircraft and related technologies. However, innovations are needed to adjust and improve these methods to tackle the particular challenges in the deep space environments, where the resource is extremely limited when it comes to electrical power, computing power, and materials.

Responding to these challenges, the following research contributions are established:

1. Focus on the Water Recycling System (WRS), further develop a physics-based model and enhance it with high fidelity fault simulations. The WRS is a critical system of the ECLSS that NASA is particularly interested in. In-depth knowledge of the physical interactions among its components and the introduction of high fidelity failure models are highly desired in the research community. Efforts on these models also help to lay the foundation for the rest of the algorithm development.

2. Develop the concept and architecture for the Automatic Contingency Management system and apply it to the WRS. The ACM is an autonomous system that automatically detects and reacts to contingencies in the system with human intervention. Autonomy is a pivotal feature since the communication latency can delay decisions that need to be made promptly in the presence of certain faults.

3. Develop system health-aware, computationally-efficient fault diagnosis algorithms for the WRS. Due to the limited computing power available in the DSH, all the algorithms (software) need to be designed with consideration of computing power. This leads to the idea of combing Lebesgue Sampling techniques with traditional fault estimation algorithms to save computing time.

4. Introduce diagnostics into control objectives; this is the key to implementing proactive fault accommodation strategies. The Model Predictive Control algorithm is enhanced with time-varying weights to incorporate diagnostics information into the control loop.

5. Address sensor faults that could otherwise mislead the ACM system; a method is put in place to distinguish sensor faults from component faults.

1.4 Dissertation Organization

Following these objectives, research work is carried out in each area. The rest of the dissertation presents the results on each topic and is organized as follows: Chapter 2 discusses the framework and main modules of ACM. Chapter 3 discusses the modeling and simulation of the WRS based on hydraulic equations and high-fidelity filter clogging model based on particle accumulation. Chapter 4 addressed the system health-aware, computationally efficient fault diagnosis algorithm which introduces Lebesgue Sampling to the well-known Unscented Kalman Filter algorithm (aka LS-UKF). the LS-UKF algorithm is then applied to the filter clogging fault scenario as a proof of concept. Chapter 5 presents the implementation of the ACM for the WRS to mitigate the performance degradation caused by filter clogging. The fault diagnosis information is used in an MPC framework to optimize the control objective.

Chapter 2

ACM Framework

To effectively optimize the operation and safety of WRS, an ACM framework that enables the mitigation of the fault condition using fault information is proposed, which shown in Figure 2.1. The main modules of ACM include a high fidelity model which represents the physical dynamic of the system, fault diagnosis which provides fault severity information, and fault-tolerant control which utilizes fault information and mitigates the effect of fault. The main modules are discussed in the following chapters.

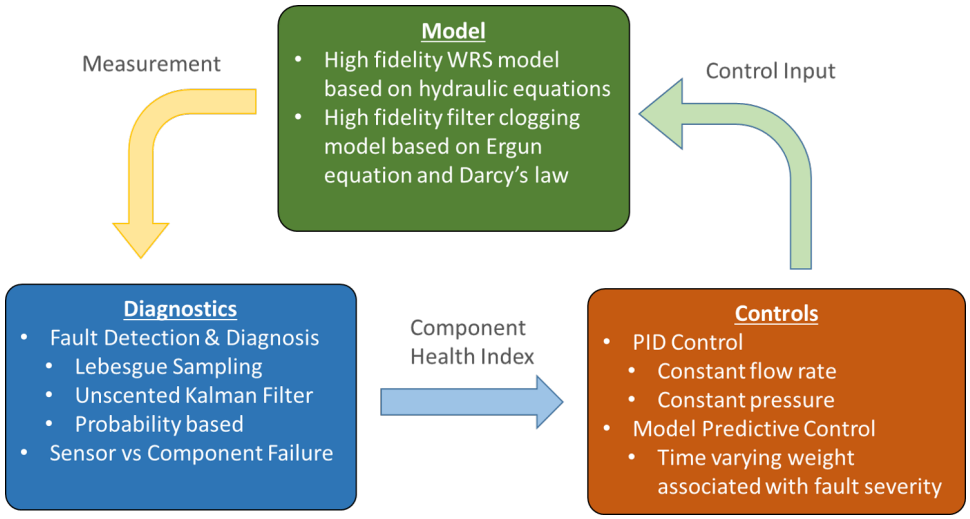


Figure 2.1 ACM Main Modules

ACM health management strategies include two features. The first feature is, to make ACM achieve safe and optimal solutions, it is essential that the fault diagnosis algorithm not only detects and identifies a fault component but also determines the severity of the fault, i.e. its potential impact on the health and stability of the

overall WRS. Some faults may be considered as minor and not affecting major mission objectives, while others may contribute to an eventual catastrophic failure or mission failure. The first category of faults may be accommodated but the second most certainly will require a reconfiguration strategy if major objectives are to be achieved.

The second feature of the ACM is considering the capabilities and constraints of the components and systems, which are indicated by the FDP information. Since the violation of any of the constraints is detrimental to the ACM optimization solution validity. The impact of any constraint violation on the cost function must be negative. The ACM collects all the diagnosis information and capability at components, systems, and mission to determine the optimal solution.

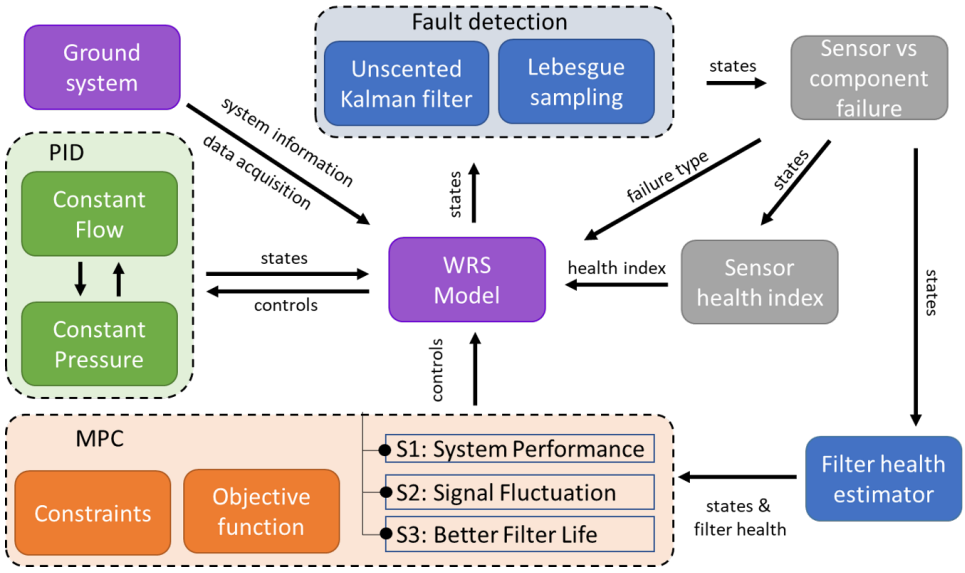


Figure 2.2 ACM with WRS

This dissertation uses WRS simulation model as a testbed to validate the proposed ACM framework. The integration of WRS and ACM is introduced in Figure 2.2. For deep space habitat, the computation resource is limited, Lebesgue Sampling-based unscented Kalman filter (LS-UKF) based fault diagnosis is introduced to reduce the utilization of the computational resource. The fault diagnosis function provides fault information that an ACM system can use to reconfigure system control strategy to

accommodate current and future faulty conditions, which based on the PID controller and MPC separately.

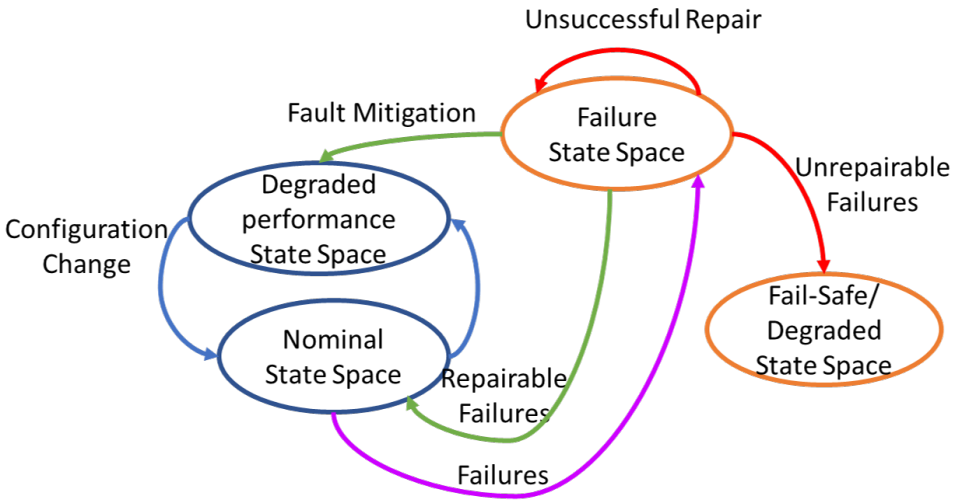


Figure 2.3 Finite State Machine (FSM) of ACM

When faults are detected, the ACM is triggered and moves the system to a failure state. Repairable failures allow the system to eventually come back to the normal state, whereas irreparable failures may force the system to a fail-safe state to avoid further catastrophes and buy some extra time before external help can be sent, if possible. However, in case of faults that may not be completely repairable, ACM tries to find alternatives that will still let the system perform within acceptable limits but with degraded performance. The transition between states is described in Figure 2.3.

Chapter 3

Water Recycling System modeling and filter clogging component fault scenario modeling

This chapter introduces the functions and modeling of WRS based on hydraulic equations. A high fidelity filter clogging fault scenario based on Ergun equation and Darcy's law is injected in the Water Recycling System (WRS) model.

3.1 Water Recycling System Modeling

A WRS plays a critical role in life support system in human habitats. The WRS collects waste water from sinks and showers and recycles them into clean water, which reduces the water consumption and extends the duration of NASA missions [46]. In this research, the WRS deployed at NASA Ames Research Center's Sustainability Base is used as a reference for modeling, fault injection, diagnosis, and automated contingency management. Figure 3.1 shows the diagram of sustainability base grey WRS.

This WRS consists of some pumps, filters, valves, tanks, pipes and is a complex hydraulic system. In the service of WRS, these components degrade, such as clogging of filter, corrosion of pumps, fatigue, fraction, and cracking of pipes, etc. These degradation result in performance degradation (clogging of the filter), and if not detected or maintained, they eventually lead to system failure (breakage of pipe or pump). Therefore, diagnosis of WRS is of vital importance to the efficiency, reliability, and safety of life support systems. In the past, the results from monitoring have often

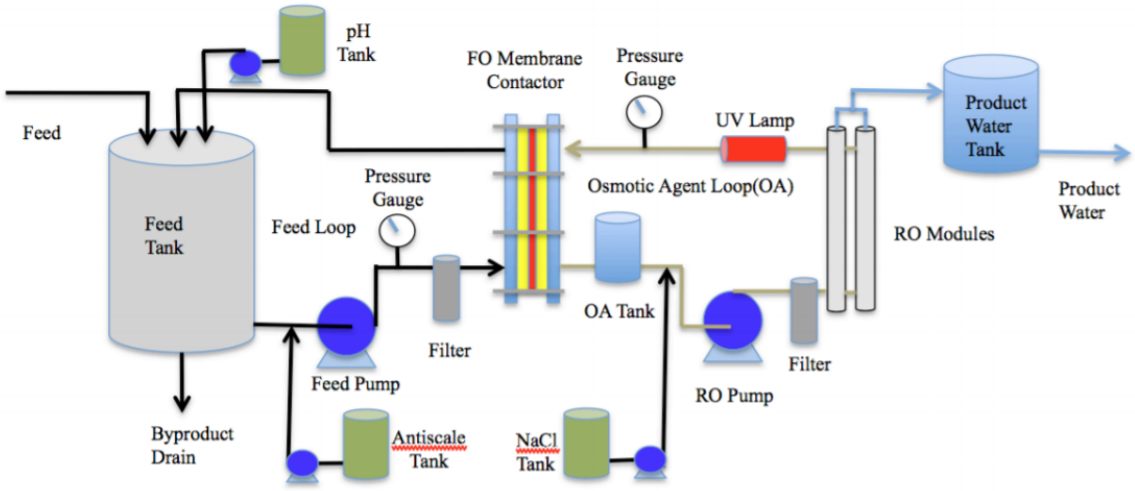


Figure 3.1 Sustainability base grey water recycling system. [46]

been used in maintenance. With development of diagnosis and prognosis, when a fault is detected, it is desirable that the system can accommodate the fault to maintain the system performance or extend the life of the system. For NASA's outer space missions, when the maintenance is not available, such capability is significant to the safety of the system and crew. To enable such capabilities in WRS, this section builds a model of the WRS with fault injection for simulation of different fault modes in WRS.

3.1.1 Forward Osmosis Module

The forward osmosis (FO) module and a reverse osmosis (RO) module are major components in the WRS with their operating mechanisms being shown in Figure 3.2. Figure 3.2A shows the FO module, in which it separates the wastewater through a semi-permeable membrane and moves the water from a region of higher water chemical potential (greywater) to a region of lower water chemical potential (saltwater). Eventually, when the liquid level pressure difference (Δp) between the two sides is the same as the osmotic pressure difference ($\Delta\pi$), the liquid level of the

saline water stays constant. The FO module has two parts FO1 and FO2 which are separated by a semi-permeable. The RO module has a similar structure.

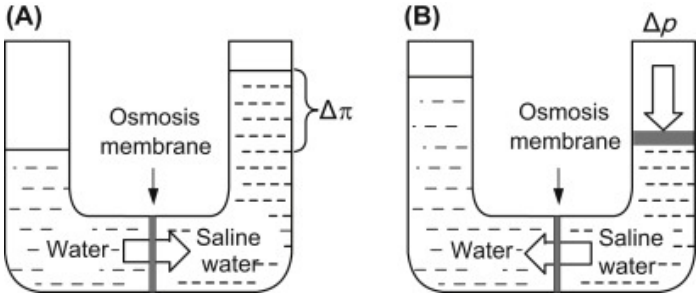


Figure 3.2 FO and RO module [47]

3.1.2 Osmotic Agent

The WRS in Figure 3.1 shows that the grey water is re-circulated through the FO module for filtration purposes. The grey water is first pre-processed by the FO module and is then processed by osmotic agent (OA), which is a 10% NaCl salt solution together with a semi-permeable membrane.

3.1.3 Reverse Osmosis Module

Then the water is transferred from OA to RO module. The RO module uses another set of semipermeable membranes and RO Pump provides hydraulic pressure to the lower water chemical potential (saltwater), as shown in Figure 3.2B. The RO pump moves the water from a region of lower water chemical potential (saltwater) to an area of higher water chemical potential (clean water) [47]. With outer pressure (Δp) on the saline water side as shown in Figure 3.2B, the water moves from saline water side to the pure water side through the semipermeable membrane due to the outer pressure ($\Delta p > \Delta\pi$) in the RO process.

Figure 3.3 shows a schematic diagram of the WRS described in Figure 3.1. The Waste Water Tank contains the grey water to be processed. The pumps provide force

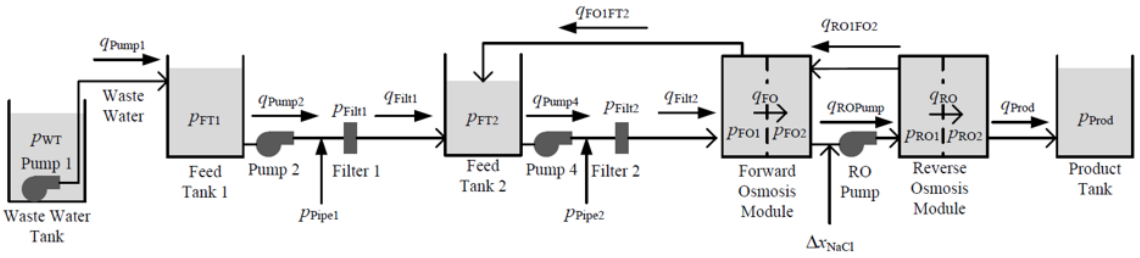


Figure 3.3 WRS layout [13]

to make sure the water recycles in the WRS. The filters are mainly used to filter the particles in the water physically. The filtered water is then transferred to FO and RO modules. The water cleaning materials in FO and RO modules are mainly used to clean water. Clean water is stored in the Product Tank. In normal operation, Pump 1 transfers water from the Waste Water Tank into Feed Tank 1. Then Pump 2 transfers water into Feed Tank 2 with suspending solids being trapped in the feed solution by Filter 1 and anti-scale chemical being added to adjust the pH value of the water. Pump 4 recirculates the water through Filter 2 and FO 1. The RO module maintains the flow of water through RO membranes by applying external pressure from RO Pump. Eventually, the clean water in Product Tank is then used as the flush water which reduces the water consumption in the building. In [13], a model of the WRS system has been developed in Eq. (3.1). White Gaussian noise is added to the model to make the model more realistic. This model is further developed and used in this research. The parameters in the model are shown in Table 3.1. The parameters are selected based on the physical limitation of the components, such as tank capacity, filter pressure, and flow rate range.

$$
\begin{aligned}
\dot{p_{\mathrm{WT}}} &= \frac{1}{C_{\mathrm{WT}}}\left(-q_{\mathrm{Pump1}}\right)\\
\dot{p_{\mathrm{FT1}}} &= \frac{1}{C_{\mathrm{FT1}}}\left(q_{\mathrm{Pump1}} - q_{\mathrm{Pump2}}\right)\\
\dot{p_{\mathrm{Pipe1}}} &= \frac{1}{C_{\mathrm{Filt1}}}\left(q_{\mathrm{Pump2}} - q_{\mathrm{Filt1}}\right)\\
\dot{p_{\mathrm{FT2}}} &= \frac{1}{C_{\mathrm{FT2}}}\left(q_{\mathrm{Filt1}} + q_{\mathrm{FO1FT2}} - q_{\mathrm{Pump4}}\right)\\
\dot{p_{\mathrm{Pipe2}}} &= \frac{1}{C_{\mathrm{Filt2}}}\left(q_{\mathrm{Pump4}} - q_{\mathrm{Filt2}}\right)\\
\dot{p_{\mathrm{FO1}}} &= \frac{1}{C_{\mathrm{FO1}}}\left(q_{\mathrm{Filt2}} - q_{\mathrm{FO1FT2}} - q_{\mathrm{FO}}\right)\\
\dot{p_{\mathrm{FO2}}} &= \frac{1}{C_{\mathrm{FO2}}}\left(q_{\mathrm{FO}} + q_{\mathrm{RO1FO2}} - q_{\mathrm{ROPump}}\right)\\
\dot{p_{\mathrm{RO1}}} &= \frac{1}{C_{\mathrm{RO1}}}\left(q_{\mathrm{ROPump}} - q_{\mathrm{RO1FO2}} - q_{\mathrm{RO}}\right)\\
\dot{p_{\mathrm{RO2}}} &= \frac{1}{C_{\mathrm{RO2}}}\left(q_{\mathrm{RO}} - q_{\mathrm{Prod}}\right)\\
\dot{p_{\mathrm{Prod}}} &= \frac{1}{C_{\mathrm{Prod}}}\left(q_{\mathrm{Prod}}\right)\\
\dot{x_{\mathrm{NaCl}}} &= -1.1111 \times 10^{-5}\\
\Delta_{x_{\mathrm{Nacl}}} &= \min\left(20, \frac{155 \times 2.78 \times 10^{-8}}{0.841 \times 10^{5} \times A_{\mathrm{FO}}} - x_{\mathrm{NaCl}}\right)\\
q_{\mathrm{Pump1}} &= u_{\mathrm{Pump1}}\left(R_{\mathrm{Pump1}}\sqrt{|p_{\mathrm{WT}} + p_{\mathrm{Pump1}} - p_{\mathrm{FT1}}|}\,\mathrm{sign}\left(p_{\mathrm{WT}} + p_{\mathrm{Pump1}} - p_{\mathrm{FT1}}\right)\right)\\
q_{\mathrm{Pump2}} &= u_{\mathrm{Pump2}}\left(R_{\mathrm{Pump2}}\sqrt{|p_{\mathrm{FT1}} + p_{\mathrm{Pump2}} - p_{\mathrm{Filt1}}|}\,\mathrm{sign}\left(p_{\mathrm{FT1}} + p_{\mathrm{Pump2}} - p_{\mathrm{Filt1}}\right)\right)\\
q_{\mathrm{Filt1}} &= R_{\mathrm{Filt1}}\sqrt{|p_{\mathrm{Pipe1}} - p_{\mathrm{FT2}}|}\,\mathrm{sign}\left(p_{\mathrm{Pipe1}} - p_{\mathrm{FT2}}\right)\\
q_{\mathrm{Pump4}} &= u_{\mathrm{Pump4}}\left(R_{\mathrm{Pump4}}\sqrt{|p_{\mathrm{FT2}} + p_{\mathrm{Pump4}} - p_{\mathrm{Filt2}}|}\,\mathrm{sign}\left(p_{\mathrm{FT2}} + p_{\mathrm{Pump4}} - p_{\mathrm{Filt2}}\right)\right)\\
q_{\mathrm{FO1FT2}} &= R_{\mathrm{FO1FT2}}\sqrt{|p_{\mathrm{FO1}}|}\,\mathrm{sign}\left(p_{\mathrm{FO1}}\right)\\
q_{\mathrm{Filt2}} &= R_{\mathrm{Filt2}}\sqrt{|p_{\mathrm{Pipe2}} - p_{\mathrm{FO1}}|}\\
q_{\mathrm{FO}} &= u_{\mathrm{FO}} \cdot Area_{\mathrm{FO}} \cdot A_{\mathrm{FO}} \cdot \left(x_{\mathrm{NaCl}} + \Delta_{x_{\mathrm{NaCl}}}\right) \times 0.841 \times 10^{5}\\
q_{\mathrm{RO1FO2}} &= R_{\mathrm{RO1FO2}}\sqrt{|p_{\mathrm{RO1}} - p_{\mathrm{FO2}}|}\,\mathrm{sign}\left(p_{\mathrm{RO1}} - p_{\mathrm{FO2}}\right)\\
q_{\mathrm{ROPump}} &= u_{\mathrm{ROPump}}\left(R_{\mathrm{ROPump}}\sqrt{|p_{\mathrm{FO2}} + p_{\mathrm{ROPump}} - p_{\mathrm{RO1}}|}\right)\mathrm{sign}\left(p_{\mathrm{FO2}} + p_{\mathrm{ROPump}} - p_{\mathrm{RO1}}\right)
\end{aligned}
\tag{3.1}
$$

$$
\begin{aligned}
q_{\mathrm{RO}} &= u_{\mathrm{RO}} \cdot Area_{\mathrm{RO}} \cdot A_{\mathrm{RO}} \left(20 \times 0.841 \times 10^5 - (x_{\mathrm{NaCl}} + \Delta_{x_{\mathrm{Nacl}}}) \times 0.841 \times 10^5\right) \\
q_{\mathrm{Prod}} &= R_{\mathrm{Prod}} \sqrt{|p_{\mathrm{RO2}} - p_{\mathrm{Prod}}|}\, \mathrm{sign}\,(p_{\mathrm{RO2}} - p_{\mathrm{Prod}}) \\
p_{\mathrm{Filt1}} &= p_{\mathrm{Pipe1}} - p_{\mathrm{FT2}} \\
p_{\mathrm{Filt2}} &= p_{\mathrm{Pipe2}} - p_{\mathrm{FO1}}
\end{aligned}
\tag{3.2}
$$

Table 3.1 Parameters for WRS model

Parameter	Values
C_{WT}	1.8614×10^{-4}
C_{FT1}	9.1207×10^{-5}
C_{FT2}	5.1207×10^{-5}
C_{Filt1}	2.0682×10^{-7}
C_{Filt2}	2.0682×10^{-7}
C_{FO1}	2.0682×10^{-6}
C_{FO2}	2.0682×10^{-7}
C_{RO1}	2.0682×10^{-6}
C_{RO2}	2.0682×10^{-6}
C_{Prod}	3.4952×10^{-5}
R_{Pump1}	1.0000×10^{-7}
R_{Pump2}	1.5000×10^{-7}
R_{Pump4}	1.2000×10^{-7}
R_{Filt1}	1.1000×10^{-7}
R_{Filt2}	5.0000×10^{-7}
R_{FO1FT2}	5.0000×10^{-7}
R_{RO1FO2}	2.6800×10^{-7}
R_{Prod}	1.0000×10^{-7}
R_{ROPump}	8.2000×10^{-8}
$Area_{\mathrm{FO}}$	1.0000×10^{-5}
A_{FO}	5.1196×10^{-6}
$Area_{\mathrm{RO}}$	1.0000×10^{-5}
A_{RO}	5.1196×10^{-6}

3.2 Simulation Results for Normal System

The WRS mission usually takes 6 months to one year to replace filters or membrane and the data sampling rate is given by 1 sample/minute. In this thesis, a mission given in 1000 seconds is used and the data sampling rate is given by 1 sample/second. This accelerated 1000 second simulation with an increased sampling rate of 1 sample/second is accelerate fault scenarios and develop fault diagnosis and ACM.

In the simulation, the case study uses an operating profile sheet shown in Table 3.2 and Figure 3.4 to simulate the real-time operational condition in the system and validate the WRS model. The operating profile sheet is adjusted from the WRS data file from NASA. The initial condition of the model is shown in Table 3.3. This system operating profile is to be used for all the simulations in this thesis.

Table 3.2 Pump open time sheet

Pump Input	Time Range(s)
u_{Pump1}	0$\sim$ 248
u_{Pump2}	109$\sim$ 387
u_{Pump4}	317$\sim$ 1000
u_{ROPump}	817$\sim$ 1000

Table 3.3 Initial value of state

Initial Value of State	Values
p_{WT}	5(psi)
p_{FT1}	2(psi)
p_{Pipe1}	0(psi)
p_{FT2}	0(psi)
p_{Pipe2}	0(psi)
p_{FO1}	0(psi)
p_{FO2}	0(psi)
p_{RO1}	0(psi)
p_{RO2}	0(psi)
p_{Prod}	0(psi)
x_{Nacl}	10(gL-1)

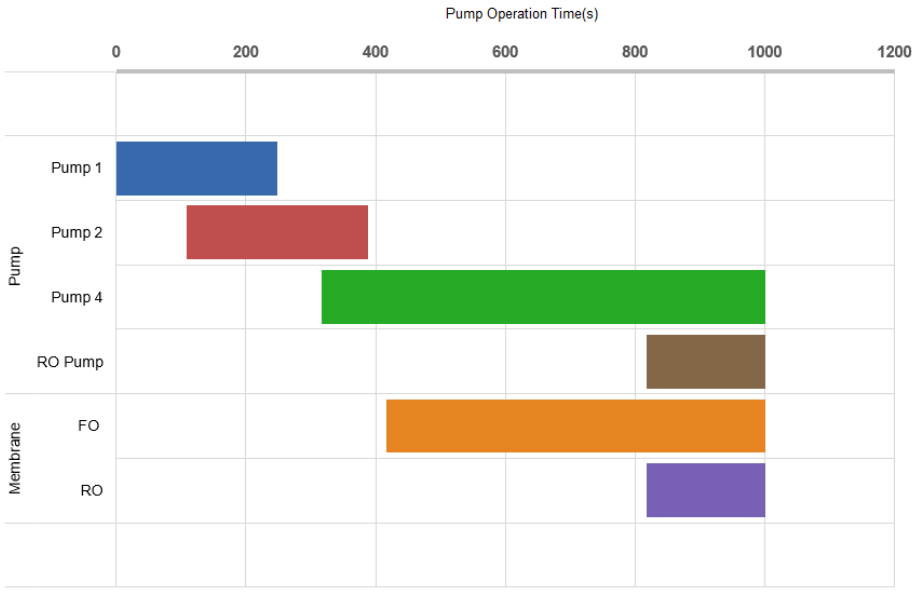


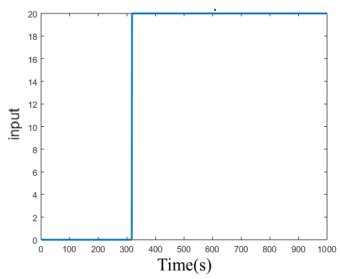
Figure 3.4 The operation time for pumps and membranes

At the beginning of the simulation, the Waste Water Tank has an initial water pressure of 5 psi. Then, Pump 1 starts working from 0 second to 248 seconds. At this time, water pressure in the Waste Water Tank decreases, because wastewater is transferred from the Waste Water Tank to the Feed Tank 1, as shown in Figure 3.5f.

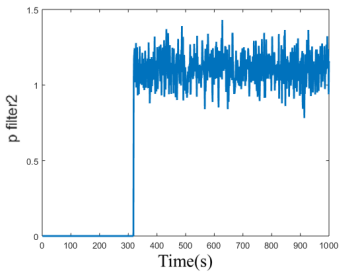
Feed Tank 1 has a water pressure of 2 psi at the beginning. As Pump 1 transfers water to the Feed Tank 1 from the beginning to the 248th second, the water pressure in the Feed Tank 1 increases. In the 109th second to the 387th second, the Pump 2 works and transfer water from Feed Tank 1 to the Feed Tank 2. As a result, the water level in the Feed Tank 1 decreases as shown in Figure 3.5g.

At the 109th second, the Pump 2 starts to transfer water and the pressure in Feed Tank 2 starts to increase, and at the 317th second, the Pump 4 turns on and it starts to transfer water from Feed Tank 2 to the FO 1 and the pressure in Feed Tank 2 decrease and eventually kept at a steady state as shown in Figure 3.5h.

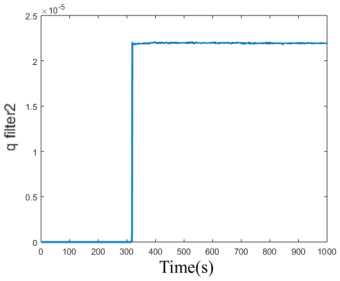
At the 317th second, the Pump 4 starts to transfer water through the filter. Figure 3.5a shows that the voltage of the Pump 4 changes from 0 V to 20 V, which indicates the motor is turned on. In the normal operation, 20 V is considered as



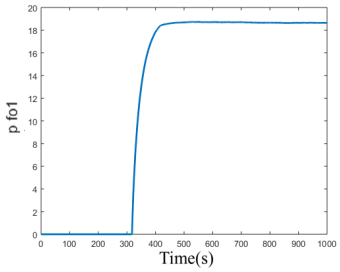
(a) Voltage of Pump 4



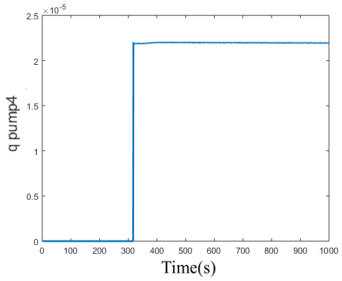
(b) Pressure of Filter 2



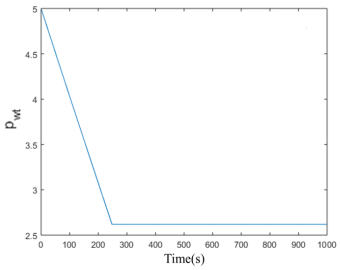
(c) Outflow rate of Filter 2



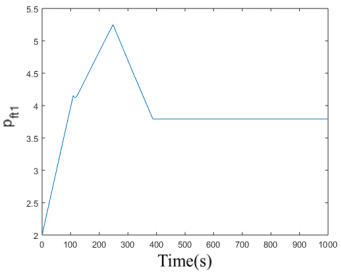
(d) Pressure of FO Module 1



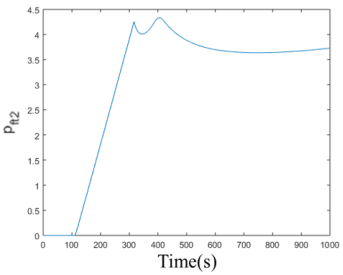
(e) Outflow rate of Pump 4



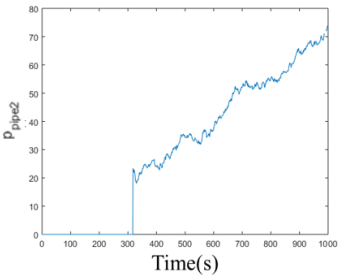
(f) Pressure of Waste Tank



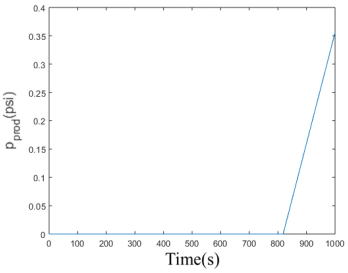
(g) Pressure of Feed Tank 1



(h) Pressure of Feed Tank 2



(i) Pressure of Pipe 2



(j) Pressure of Product Tank

Figure 3.5 WRS normal system simulation result

normal operational voltage. According to the motor setting, the voltage range of the motor is from 0 V to 24 V.

In this simulation, the water is assumed clean. When Pump 4 turns on, the clean water with a very small number of particles passes through the filter. The

degradation rate of Filter 2 is very small with clean water and can be neglected. The flow rate keeps at a constant of 2.2 e-5 cu. m/s, as shown in Figure 3.5c. The increase in pressure of Filter 2 is very slow. Therefore, the pressure is caused by the filter hydraulic resistance when the filter is clean is 1 psi as shown in Figure 3.5b. The pressure of FO 1 represents how much water in FO 1. In the normal condition, the pressure of FO 1 is stable which means there is enough water for the following modules in the system as shown in Figure 3.5d. In the end, the pressure in FO 1 remained at a steady level for the reason that the water transferring into FO 1 and the water transferring out from the FO 1 are the same. A similar analysis can be obtained from the RO module.

At the 817th second, with constant water flow into the Product Tank, the water level in the Product Tank begins to increase, which is shown in Figure 3.5j.

3.2.1 Filter Clogging Fault Scenario Based on the Hydraulic Resistance

In the previous study [48], a degradation model based on the decreasing of the hydraulic resistance R is used to describe the degradation of filter.

As an example, when a clogging fault happens in Filter 2, the hydraulic resistance related to the outflow rate of Filter 2, denoted by R_{Filt2}, decreases. The clogging of the Filter 2, reflected by R_{Filt2}, can be represented as:

$$R_{\text{Filt2}} = \begin{cases} 5 \times 10^{-7}, t \leq \text{t}_\text{f} \\ 5 \times 10^{-7} - 5 \times 10^{-14} * (t - \text{t}_\text{f})^2, \text{ otherwise} \end{cases} \tag{3.3}$$

where t_f is the time when the clogging fault occurs. In this example, $t_\text{f} = 317$ second and R_{Filt2} is the hydraulic resistance with initial value of $R_{\text{Filt2}} = 5 \times 10^{-7}$, and the sampling time is 1 second. Figure 3.6 shows the filter resistance degradation curve defined in Eq. (3.3).

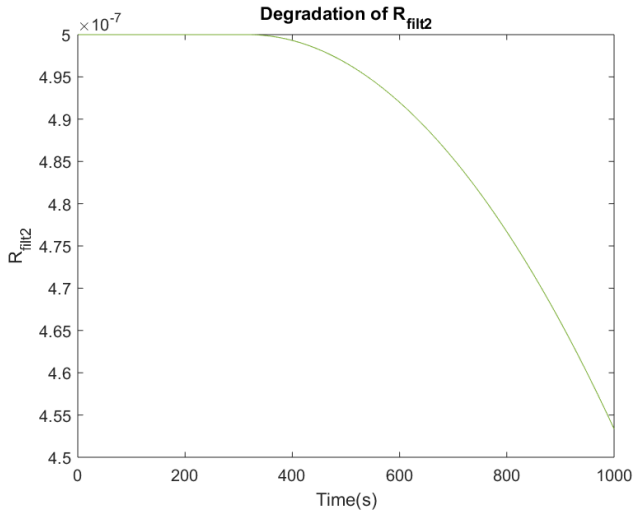
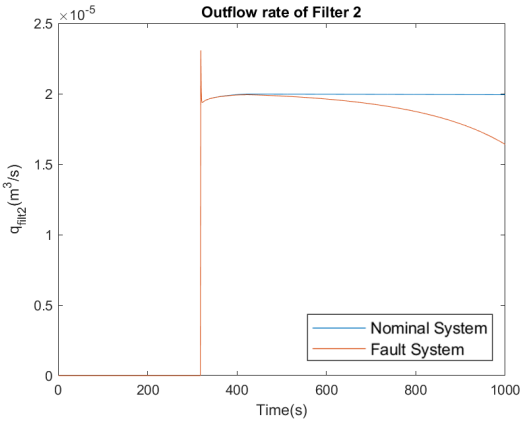
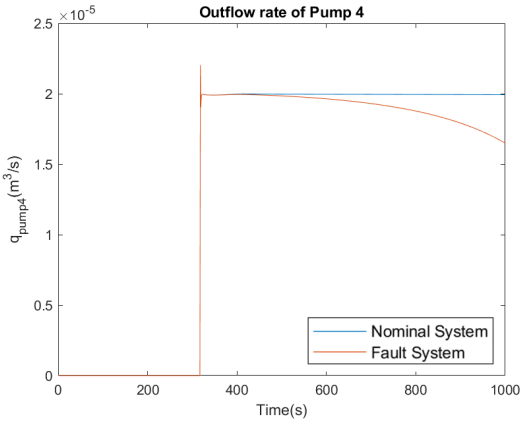


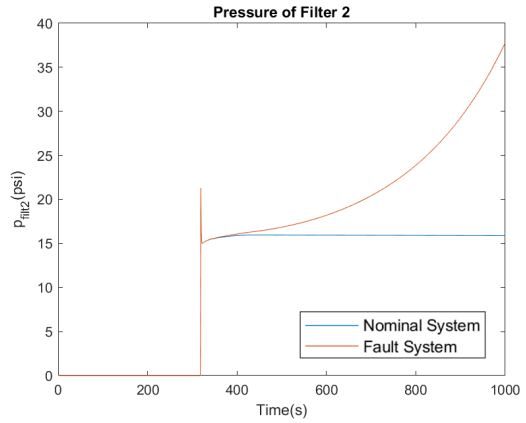
Figure 3.6 The degradation of hydraulic resistance R



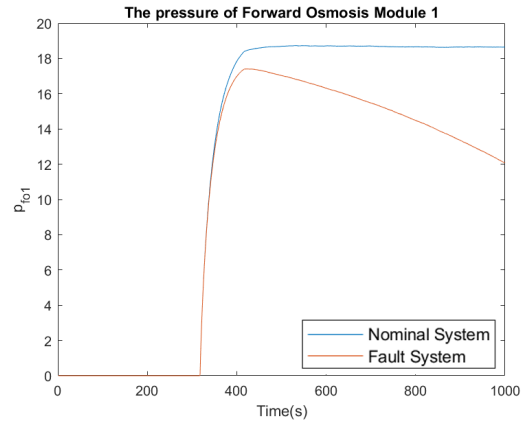
(a) Outflow rate of Filter 2



(b) Outflow rate of Pump 4



(c) Pressure of Filter 2



(d) Pressure of Forward Osmosis 1

Figure 3.7 WRS filter clogging simulation result

The filter clogging fault scenario is realized by reducing the hydraulic resistance R. When fault happens, the value of R decreases to indicate the decrease of the permeability of filter. Therefore, flow rate of Filter 2 decreases (shown as Figure 3.7a) and pressure of Filter 2 increases (shown as Figure 3.7c). As a result, the water in the FO 1 decreases, shown as Figure 3.7d.

3.3 High Fidelity Filter Clogging Component Fault Model

3.3.1 Filter Physical Configuration

Although the model shown in Eq. (3.3) is simple to analyze, the hydraulic resistance R does not illustrate the physical structure of filter as well as the degradation mechanism. For fault diagnosis and ACM development, it is desirable to develop a high-fidelity model.

To increase the fidelity of the clogging modeling in WRS, Ergun equation and Darcy's law are used to develop a more realistic filter model. Figure 3.8 shows the physical configuration of the filter. When water with high suspension rate pass through the filter, the filter filtrate water, particles remain in the filter and clean water with low suspension rate come out. The particles remained in the filter cause the decrease of porosity ϵ and increase of cake thickness L.

The high fidelity modeling of this configuration is discussed as follows.

3.3.2 Ergun Equation

Ergun equation represents the relationship between pressure drop across a filter and a number of factors of filter including viscosity, velocity of fluid, cake thickness, and porosity of cake, particle diameter and particle roundness. It has the form of:

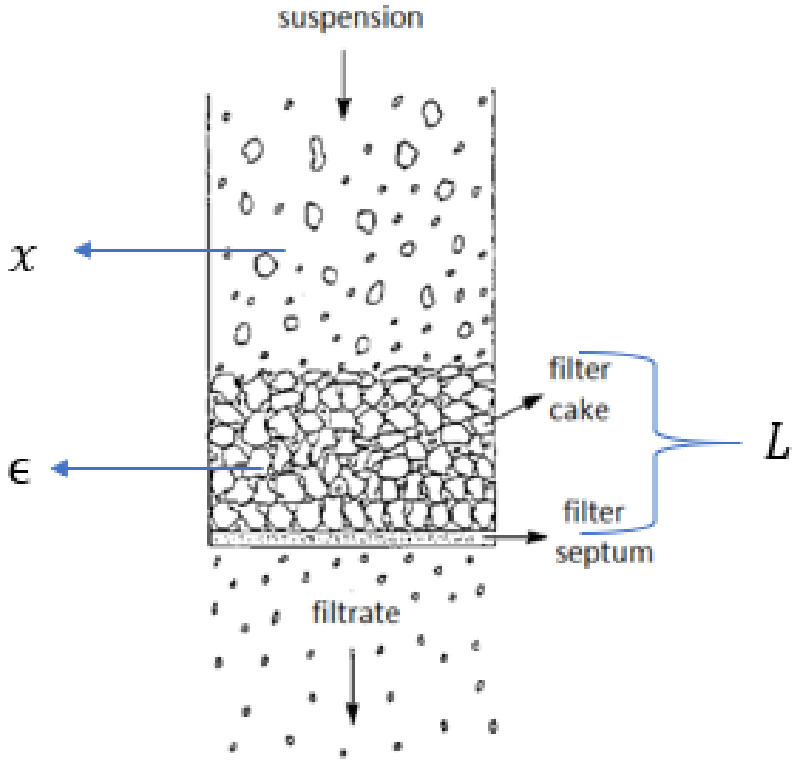


Figure 3.8 The physical configuration of the filter

$$\Delta p = \frac{150 v_s \mu (1-\epsilon)^2 L}{D_p^2 \epsilon^3} + \frac{1.75(1-\epsilon)\rho v_s^2 L}{\epsilon^3 D_p} \tag{3.4}$$

- Δp is the pressure drop across the bed
- L is the length of the bed
- D_p is the equivalent spherical diameter of the packing
- ρ is the density of the fluid
- μ is the dynamic viscosity of the fluid
- v_s is the superficial velocity
- ϵ is the void fraction of the bed (bed porosity at any time)

3.3.3 Darcy's Law

Darcy's law shows a proportional relationship between the instantaneous flow rate through a porous medium of permeability, the dynamic viscosity of the fluid and the pressure drop over a given distance in a homogeneously permeable medium. The permeability is used to measure filter's capability to filter the solid particles and organisms in the water. The Darcy's law has the form of:

$$Q = \frac{KA}{\mu L}\Delta p \tag{3.5}$$

- K is the permeability
- A is the cross-sectional area to flow
- μ is the dynamic viscosity of the fluid
- L is the length of the bed

3.3.4 Parameters in Filter Clogging Model

Besides filter pressure and flow rate, the length of the bed L, porosity ϵ, and permeability K of the filter need to be modeled as well. They are quantified for each particular filter. The cake thickness is related to the water quality and physical configurations of the filter, which can be described as follows:

$$L = \frac{\ln\left(l_1 \frac{\sum Qx}{A_f}\right)}{l_2} \tag{3.6}$$

where $\sum Qx$ stands for cumulative particle volume retained in the filter chamber, Q is the flow rate, x is the solid fraction of the suspension, l_1 and l_2 are fixed parameters, and A_f stands for filtration area.

The porosity of the filter measures the void spaces in the filter, which is represented by a fraction of the volume of voids over the total volume, between 0 and 1. It can be described as:

$$\epsilon = 1 - \frac{e^{\left(p_1 \frac{\sum Qx}{V_f}\right)}}{p_2} \tag{3.7}$$

where $\frac{\sum Qx}{V_f}$ term gives the solid fraction of the cake, p_1 and p_2 are fixed parameters, V_f is the maximum filtration volume that can be filled in the filter chamber.

The permeability of the filter can be described as:

$$K = \frac{1}{CS^2} \frac{\epsilon^3}{(1-\epsilon)^2} \tag{3.8}$$

where C is Kozeny constant, which is a function of the shape and size distribution of the cross-sectional areas, and S is the mean surface of the particles.

Filter Clogging Component Fault Scenario under Constant Flow Rate

In this case study, we want to study the filtering model response when the flow rate from the pump is constant. The flow rate is set as 600 ml/min and the percentage of suspension x is 0.14%. Other parameters of the simulation are shown in Table 3.4.

Table 3.4 Parameters for filter clogging fault scenario

Parameters	Value
V_f	6.7×10^{-6}
A_f	0.001344
ρ	1000
μ	8.9×10^{-4}
D_p	5.8×10^{-5}
l_1	4×10^5
l_2	899
p_1	2.2346
p_2	9.1464

Figure 3.9 shows that when the pump starts to transfer water through the filter, particles accumulate in the filter and the fault of clogging is initiated. This leads

to an increase of cake thickness of L, a decrease of porosity ϵ, and a decrease of permeability K. According to the Ergun equation, the two parameters L and ϵ cause an increase of pressure across the filter.

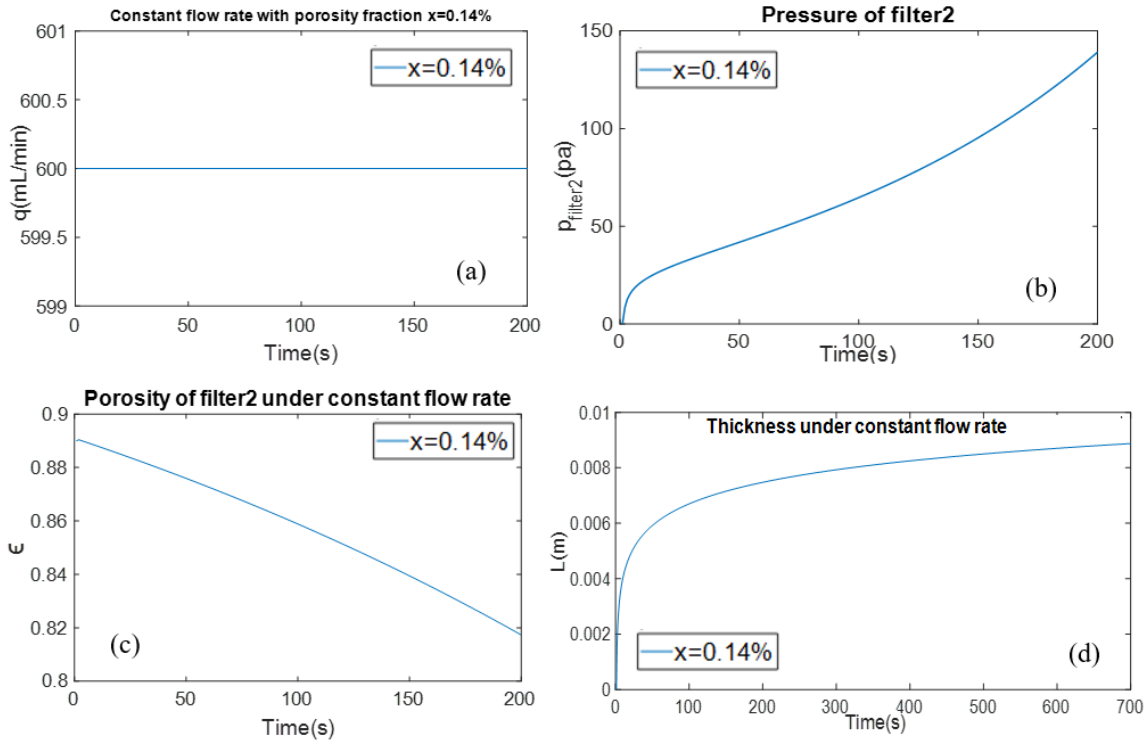


Figure 3.9 Filter clogging fault scenario simulation result under constant flow rate

Filter Clogging under Constant Flow with Different Particle Volume Fractions x

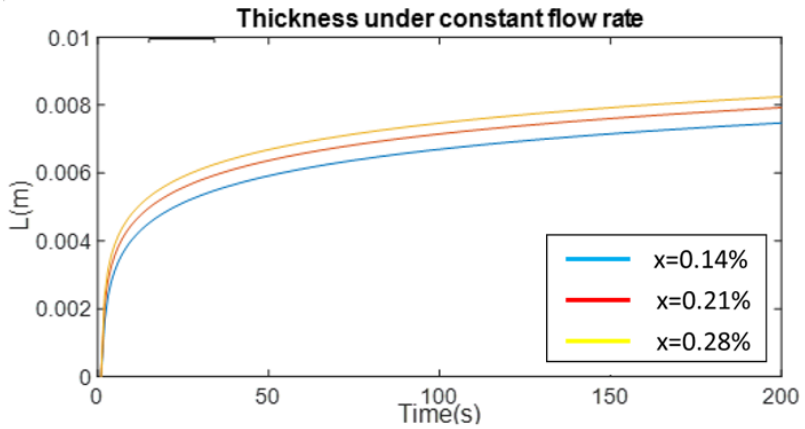
Different volume fractions of particles x in the water represent how many particles are suspended in the incoming gray water transferred through the filter. The simulation is conducted at constant flow rate at $1 \times 10^{-5} m^3/s$ which is 600 ml/min, and x is set at 0.14%, 0.21% and 0.28% to represent different levels of the water quality. A larger number represents more particles in the water stream. Figure 3.10 shows the changes of length of bed and porosity under different percentages of suspension. It can be seen that more particles in the water leads to faster decrease of porosity and increase of cake thickness. Moreover, based on the Ergun equation, the pressure increase faster.

Filter Response Scenario Under Impulse Flow Rate

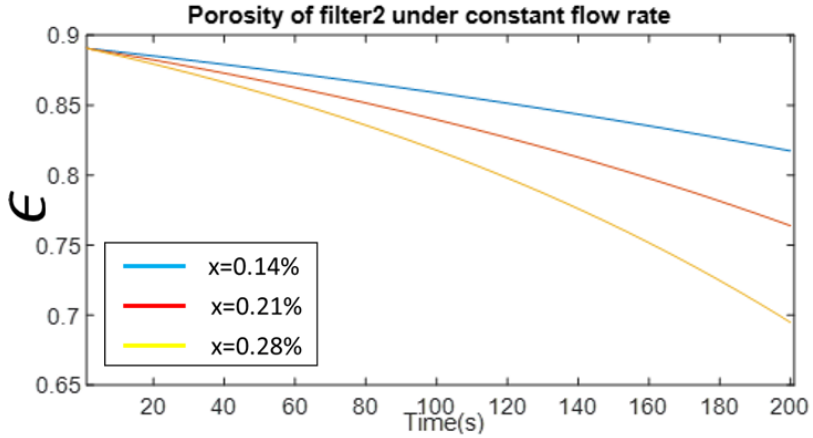
In this scenario, Pump 4 provides water at $1 \times 10^{-5} m^3/s$, for only 1 second as an impulse input, and the percentage of suspension x is 0.14%. Figure 3.11 shows the filter clogging fault scenario simulation result under the impulse flow rate. The flow rate continues to decrease because water flow in the filter is for a very short time. The pressure increases at first because water passes through the filter at first. When water is gone, the pressure remains as a very small constant because there are particles accumulated in the filter. Cake thickness L increases at the very beginning but stay as a constant because particles stop being accumulated in the filter. A similar analysis can be observed for the porosity ϵ.

3.3.5 Integration of Ergun Equation and Darcy's Law with WRS

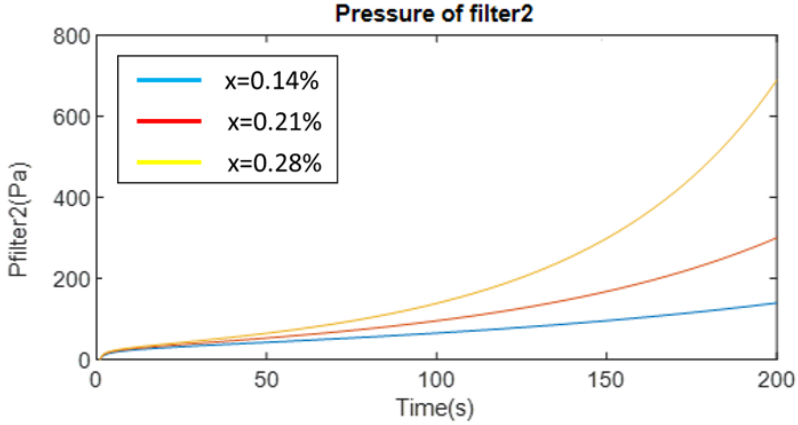
To integrate the Ergun Equation and Darcy's Law with WRS and generate a more realistic filter clogging fault scenario, the pressure, and flow rate equations for Filter



(a) Cake thickness L



(b) Porosity



(c) Pressure

Figure 3.10 Filter clogging fault scenario simulation result under constant flow rate with different percentages of suspension x

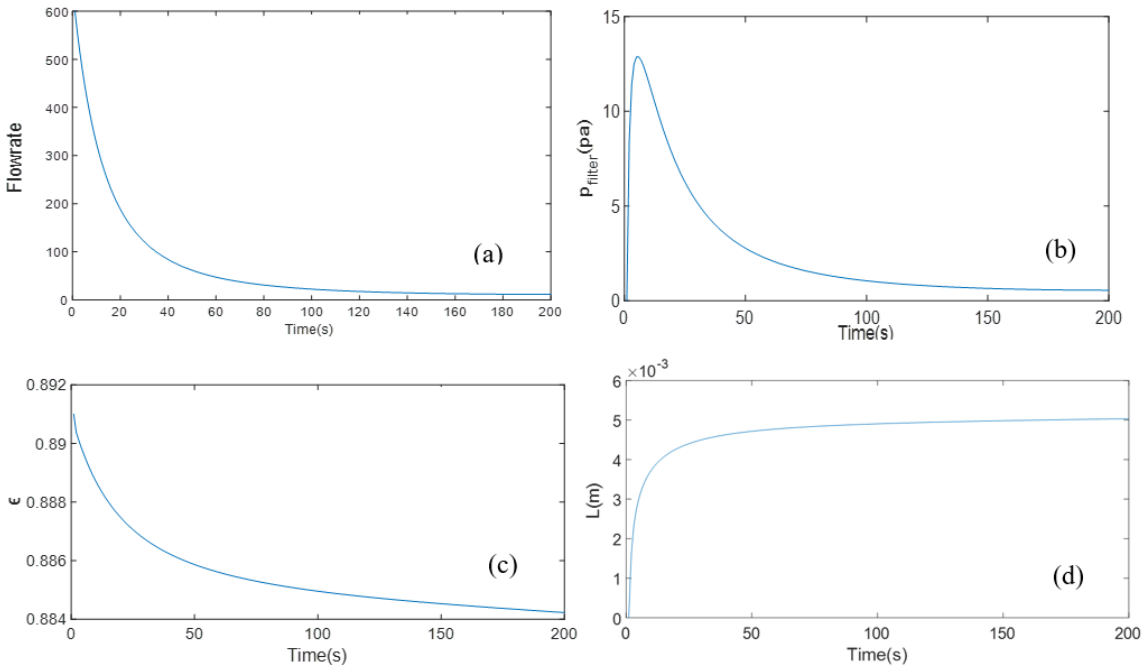


Figure 3.11 Filter clogging fault scenario simulation result under impulse flow rate

2 are replaced by the Ergun equation and Darcy's law respectively, as shown in Table 3.5.

Table 3.5 Compare previous filter clogging model with current model

Measurement	Previous Model	Current Model
Pressure	$p_{\text{Filt2}} = p_{\text{Pipe2}} - p_{\text{FO1}}$	$p_{\text{Filt2}} = \frac{150 v_s \mu (1-\epsilon)^2 L}{D_p^2 \epsilon^3} + \frac{1.75(1-\epsilon)\rho v_s^2 L}{\epsilon^3 D_p}$
Flow Rate	$q_{\text{Filt2}} = R_{\text{Filt2}} \sqrt{\lvert p_{\text{Pipe2}} - p_{\text{FO1}} \rvert}$	$q_{\text{Filt2}} = \frac{KA}{\mu L} p_{\text{Filt2}}$

The Ergun equation and Darcy's law-based filter clogging fault with WRS is shown in Figure 3.12. In this fault scenario, the filter begins to clog when the Pump 4 starts to transfer water through the filter. When water flows through Filter 2, the particles in the water accumulate in the filter as $x \sum Q$, which leads to the decrease of porosity ϵ shown in Eq. (3.7) and increase of the cake thickness L as in Eq. (3.6). As mentioned earlier, the change of these two parameters L and ϵ cause the increase of filter pressure based on the Ergun equation, Eq. (3.4). When ϵ decreases, the permeability decrease as shown in Eq. (3.8). Eventually, the flow rate decreases,

because K has a significant influence on flow rate Q based on Darcy's law as shown in Eq. (3.5).

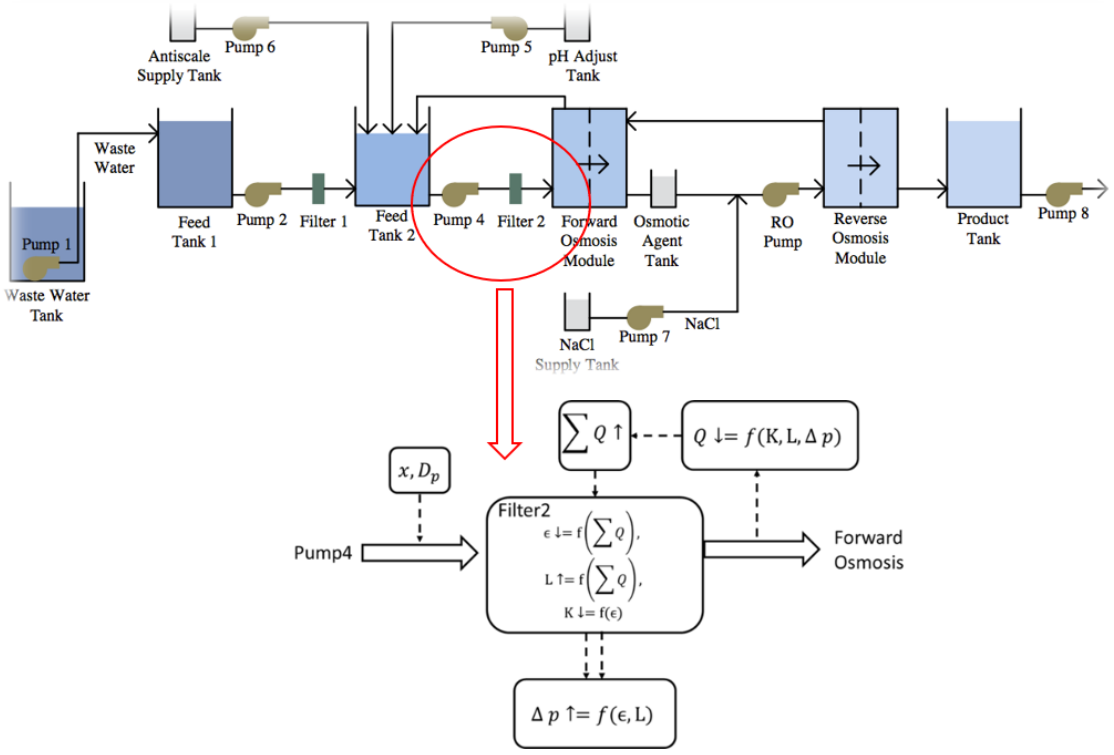


Figure 3.12 Injecting Darcy's law and Ergun equation-based filter clogging fault scenario in the WRS

3.3.6 Simulation Results for Filter Clogging Component Fault

The simulation is performed corresponding to the profile given in Table 3.2. In the fault condition, it is assumed that the water has many particles. The particles begin to accumulate in the filter when the water pass through the filter, which leads to the increase of pressure of Filter 2 p_{Filt2} (as shown in Figure 4.3a) and the decrease of flow rate of Filter 2 q_{Filt2} (as shown in Figure 4.3b). The variation of these two measurements can be used to estimate the value of ϵ, which are introduced in the next section. When the pressure of Filter 2 p_{Filt2} increases, the pressure difference between

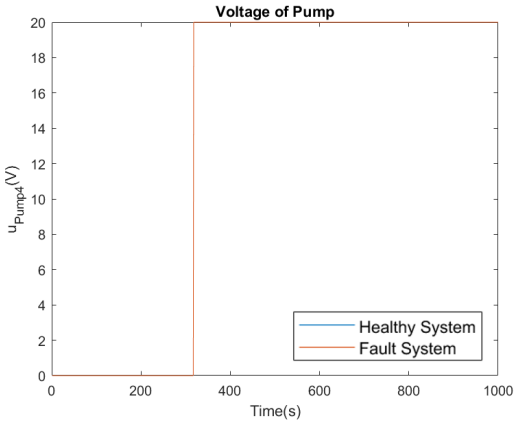
p_{Pump4} and p_{Filt2} decreases. Therefore, the flow rate of Pump 4, q_{Pump4} decreases, as shown in Figure 3.13e.

Note that, in these results, no control is added and therefore the flow rate and pressure of Filter 2 change as the physical degradation of filter. When control is added, the filter should be able to maintain the desired working condition even under clogging conditions.

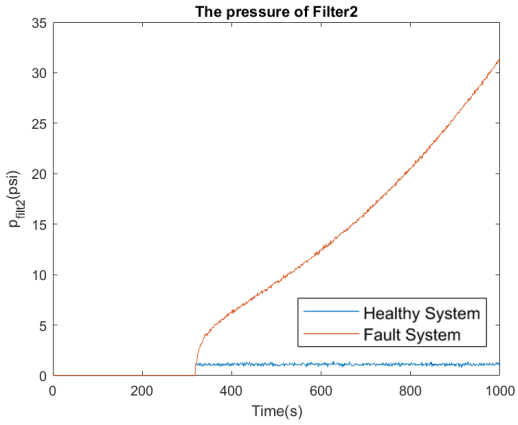
In previous work, the fault severity is introduced by the coefficient of filter resistance, which does not demonstrate the mechanism of filter clogging. To improve the fidelity of the filter clogging model, Ergun equation and Darcy's law are introduced. The effectiveness of the improved model is demonstrated by a series of simulation results. In the following sections, Ergun equation and Darcy's law are combined to achieve more realistic filter clogging fault model in WRS model for diagnosis and ACM.

3.4 Conclusion

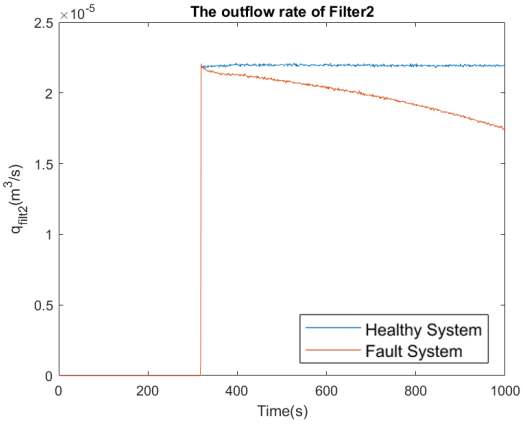
In this section, the physical dynamic model for WRS is introduced. In previous work, the fault severity is introduced by using the coefficient of filter resistance, which does not demonstrate the mechanism of filter clogging. To improve the fidelity of the filter clogging model, Ergun equation and Darcy's law are introduced. The effectiveness of the improved model is demonstrated by a series of simulation results. The main work is combining Ergun equation and Darcy's law to achieve more realistic filter clogging fault model in WRS model for diagnosis and ACM.



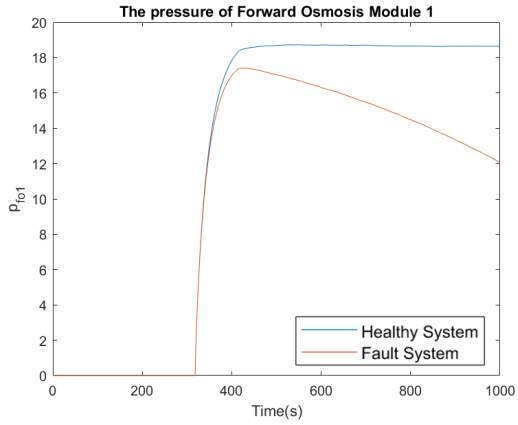
(a) Voltage of Pump 4



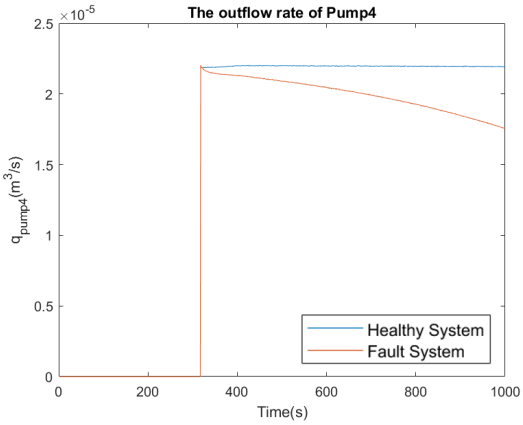
(b) Pressure of Filter 2



(c) Outflow rate of Filter 2



(d) Pressure of Forward Osmosis Module 1



(e) Outflow rate of Pump 4

Figure 3.13 WRS filter clogging simulation results

Chapter 4

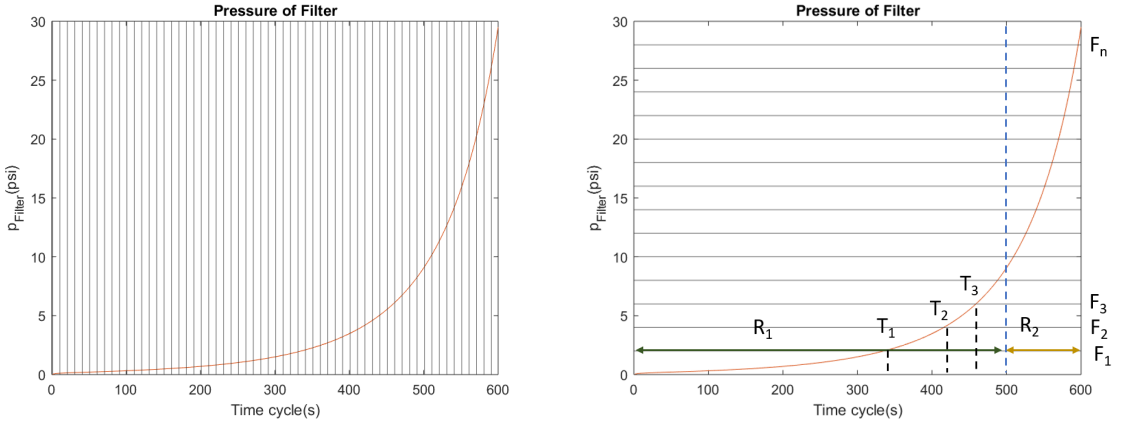
Unscented Kalman filter for fault diagnosis and prognosis

This section introduces Lebesgue Sampling and Unscented Kalman Filter (LS-UKF) based fault diagnosis and prognosis. The fault diagnosis function provides fault information that an ACM system can use to reconfigure system control strategy to accommodate current and future faulty conditions. The fault information includes fault mode, fault severity level, fault detection time and time to failure. Since computational resources are limited in space operations, Lebesgue Sampling is introduced to reduce the computational resource requirements.

4.1 Lebesgue Sampling based Fault Diagnosis

In a remote environment like deep space habitat, computational resource is extremely limited. A fault diagnostics system has to share processor time and power with many other mission critical systems. Therefore, the diagnostic algorithms have to be as computationally efficient as possible. To this end, Lebesgue sampling (LS) is introduced in diagnosis and prognosis to reduce the computation and enable real-time fault diagnosis [35]. While most diagnostic algorithms run at a fix interval (step size), Lebesgue sampling method allows "event driven" execution of the diagnostic algorithm at an uneven step size. In the LS-based diagnosis algorithm, the range of the state $x(t)$ is partitioned into Lebesgue states $\{F_1, F_2, \cdots, F_n\}$, with which the diagnostic model is discrete. The event is defined as the state $x(t)$, which is reflected by the

fault indicator extracted from the raw measurement, changes from one Lebesgue state to another. The time instant when an event is generated is called the "event stamp". The sequence of the event stamps is denoted as $\{T_1, T_2, \cdots, T_n\}$. If an event happens when a new measurement becomes available, it indicates the fault state has significant change and the diagnostic algorithm is executed. The output of the diagnostic algorithm is an estimate of the current states and the probability that a fault has been detected. On the other hand, if a new measurement does not trigger an event, it indicates the fault state does not have much change and the diagnostic algorithm will not be executed. In this case, the fault state estimated from a previous time stamp is used as the fault state at the current time instant.



(a) RS with fixed time interval (b) LS with fixed Lebesgue state length

Figure 4.1 Illustration of LS

To illustrate the advantages of LS, suppose filter pressure is used as a feature to estimate the filter clogging. Note that, filter pressure is not used as feature in this thesis, and it is only used as illustration purpose.

Figure 4.1a shows that the filter pressure p_{Filter} increases at the constant flow rate operating condition when filter clogging happens. The red curve is the pressure of filter p_{Filter}, the black vertical lines represent the RS time interval. The pressure of filter in the range $R_1 = [0, 500]$ second is slower than that in the range $R_2 = [500, 600]$ second. Using the Riemann sampling (RS) method with fixed time interval, as

shown in Figure 4.1a, the diagnosis algorithm is executed at each time instance no matter whether the measurement shows the fault condition has changes or not. This requires a lot of computation resources and is not suitable for applications where the computation resource is limited, like the deep space habitat.

To address this problem, LS is introduced and integrated with diagnosis. For example, Figure 4.1b shows that only 5 Lebesgue states are defined in 500 seconds when the pressure increases with a slow rate. On the contrary, 10 Lebesgue states are defined in 100 seconds in R_2 when the pressure increases with a fast rate. This indicates that the filter pressure causes 5 events in the range of R_1 and the diagnostic algorithm only executes 5 times. Similarly, the filter pressure causes 10 events in the range of R_2 and the diagnostic algorithm executes 10 times. Compared with the RS scheme, Figure 4.1a, in which the diagnostic algorithm executes every time when a filter pressure measurement becomes available, the LS scheme shown in Figure 4.1b can significantly reduce the computation. It is worth noting that during R_2, LS scheme allows more computing resource to be allocated to the algorithm due to the more frequent transitions between Lebesgue states. This indicates that the LS-based scheme can adaptively configure the computation resources, which saves computation when the fault dynamic is slow and assigns more computation power to diagnosis when the faulty dynamic is fast to get more accurate diagnosis results.

4.2 Unscented Kalman Filter

Traditional Bayesian estimation methods such as Kalman filter (KF), extended Kalman filer (EKF) and unscented Kalman filter (UKF) have shown many successes in diagnosis and prognosis. In the diagnosis of filter clogging, UKF is selected due to its advantages of highly efficiency and better linearization. Moreover, UKF does not need to calculate the Jacobian of process and measurement equations, which is suitable for data-driven modeling.

Fault diagnosis in the Bayesian estimation framework can be formulated as a state estimation problem. In this framework, unobserved faulty conditions is estimated from system observations that are related to the unobserved faulty condition and a dynamic faulty degradation model. Mathematically, LS-based diagnosis is defined as follows: Assume the unobserved fault process X to be a Markov process characterized by initial distribution $p(x_0)$ and the transition probability $p(x_k|x_{k-1})$, which is defined by $x_k = f(x_{k-1}, \omega_k)$ with ω_k being the process noise, subscript k represents the k^{th} event stamp caused by the transition of Lebesgue states. The observations Z are assumed to be conditionally independent of all other variables given X. The distribution of $Z_k|X_k$ is defined by $z_k = h(x_k, v_k)$ with v_k being observation noise. Let $x_{0:k} = \{x_0, \cdots, x_k\}$ and $z_{1:k} = \{y_1, \cdots, z_k\}$ denote the state and the observation up to the k^{th} event. It is of interest to estimate the posterior distribution $p(x_{0:k}|z_{1:k})$.

Consider the following nonlinear system, described by the process and the observation model with additive noise:

$$x_k = f(x_{k-1}) + w_{k-1}$$
$$z_k = h(x_k) + v_k$$

where w_{k-1} and v_k are process noise and measurement noise that are subject to Gaussian distributions. In filter clogging case, state x is the filter porosity ϵ and z is the feature, which is extracted from filter measurements of flow rate and filter pressure. The implementation of UKF involves selection of sigma points, forecast, and data assimilation [49], which are discussed as follows:

4.2.1 Selection of Sigma Points

Let X_{k-1} be a set of $2n+1$ sigma points (where n is the dimension of the state space) and W as their associated weights:

$$x_{k-1} = \left\{ \left(x_{k-1}^j, W^j \right) | j = 0 \dots 2n \right\} \tag{4.1}$$

Let P_{k-1} be the co-variance and W^j for all $j = 1 \dots 2n$ be the weights on the sigma points $j = 1, \dots, 2n$.

Consider the following selection of sigma points that incorporates higher order information in the selected points[50]:

$$x_{k-1}^0 = x_{k-1}^a, \text{the first sigma point is the mean} \tag{4.2}$$

$$x_{k-1}^i = x_{k-1}^a + \left(\sqrt{(n+\lambda) * P_{k-1}}\right)_i \text{ for all } i = 1 \dots n \tag{4.3}$$

$$x_{k-1}^{i+n} = x_{k-1}^a - \left(\sqrt{(n+\lambda) * P_{k-1}}\right)_i \text{ for all } i = n+1 \dots 2 \tag{4.4}$$

$$W_m^0 = \frac{\lambda}{n+\lambda} \tag{4.5}$$

$$W_c^0 = W_m^0 + \left(1 - \alpha^2 + \beta\right) \tag{4.6}$$

$$W_m^j = W_c^j = \frac{\lambda}{2(n+\lambda)} \text{for all} j = 1 \dots 2n \tag{4.7}$$

where λ is given as $\lambda = \alpha^2(n+k) - n$. Table 4.1 shows the parameters that determine the sigma points. Moreover, the sum of weights must be 1, i.e.,

$$\sum_{j=0}^{2n} W_m^j = 1 \tag{4.8}$$

Table 4.1 Parameters in UKF

Parameters	Values
α	1e-3
k	0
β	2

4.2.2 Prediction

In the prediction stage, each sigma point propagates through the nonlinear process model:

$$x_k^{f,j} = f\left(x_{k-1}^j\right) \tag{4.9}$$

where $x_k^{f,j}$ are the sigma points after propagated through the nonlinear process model.

The transformed points are used to compute the mean and covariance of the forecast value of x_k^f:

$$x_k^f = \sum_{j=0}^{2n} W_m^j x_k^{f,j} \tag{4.10}$$

$$P_k^f = \sum_{j=0}^{2n} W_c^j \left(x_k^{f,j} - x_k^f\right)\left(x_k^{f,j} - x_k^f\right)^T + Q_{k-1} \tag{4.11}$$

where Q_{k-1} is the process noise covariance.

The sigma points of measurement $z_{k-1}^{f,j}$ propagate through the nonlinear observation model:

$$z_{k-1}^{f,j} = h\left(x_{k-1}^j\right) \tag{4.12}$$

With the transformed observations, the mean z_{k-1}^f and covariance $\mathrm{Cov}\left(\tilde{z}_{k-1}^f\right)$ (innovation covariance) are computed as:

$$z_{k-1}^f = \sum_{j=0}^{2n} W_m^j z_{k-1}^{f,j} \tag{4.13}$$

$$\mathrm{Cov}\left(\tilde{z}_{k-1}^f\right) = \sum_{j=0}^{2n} W_c^j \left(z_{k-1}^{f,j} - z_{k-1}^f\right)\left(z_{k-1}^{f,j} - z_{k-1}^f\right)^T + R_k \tag{4.14}$$

where R_k is the measurement noise covariance.

The cross-covariance between $\tilde{x}_k^f$ and $\tilde{z}_{k-1}^f$ is:

$$\mathrm{Cov}\left(\tilde{x}_k^f, \tilde{z}_{k-1}^f\right) = \sum_{j=0}^{2n} W_c^j \left(x_k^{fi} - x_k^f\right)\left(z_{k-1}^{fi} - z_{k-1}^f\right)^T \tag{4.15}$$

4.2.3 Data Assimilation

The information obtained in the prediction stage needs to be updated with the new observation z_k (the feature extracted from filter measurements) to achieve the posterior estimation of states. The mean value is updated as:

$$x_k^a = x_k^f + k_k\left(z_k - z_{k-1}^f\right) \tag{4.16}$$

where gain K_k is given by:

$$K_k = \mathrm{Cov}\left(\tilde{x}_k^f, \tilde{z}_{k-1}^f\right)\mathrm{Cov}^{-1}\left(\tilde{z}_{k-1}^f\right) \tag{4.17}$$

The posterior covariance is updated as:

$$P_k = P_k^f - K_k\,\mathrm{Cov}\left(\tilde{z}_{k-1}^f\right)K_k^T \tag{4.18}$$

4.3 LS-UKF Diagnosis Methodology

When a fault is formulated as a state, the diagnosis problem becomes estimating fault magnitude (severity) based on system measurements (observation of feature). In LS-UKF, the algorithm first checks if the measurement (observation of a feature) triggers an event, i.e., if the feature value changes from one Lebesgue state to another one. If yes, it indicates the fault state has a significant change. Then, the LS-UKF takes the measurement to conduct fault diagnosis and estimates the fault state. If not, it indicates the fault state does not have significant change. In this case, fault diagnosis is not needed, and the fault state estimated from diagnosis at the previous timestamp is used as the fault stat at the current time instant. When

LS-UKF based fault diagnosis is implemented, the sigma points drawn from the state probability density function are propagated through the state equation (process model) to predict the state at the next time instant. Then the predicted state is updated based on systematic observation of the feature by applying a filtering process to get the posterior estimation of fault state and covariance. Figure 4.2 shows the implementation flow chart of the above-discussed LS-UKF based fault diagnosis.

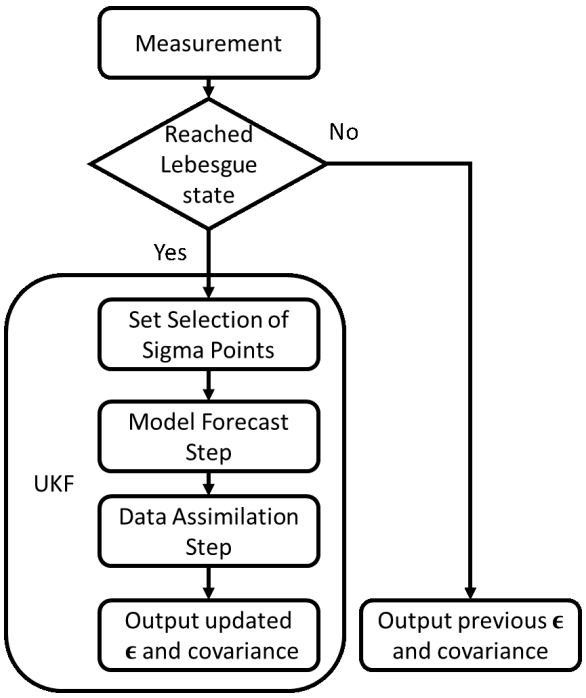


Figure 4.2 LS-UKF based fault diagnosis

4.4 LS-UKF Diagnosis with WRS

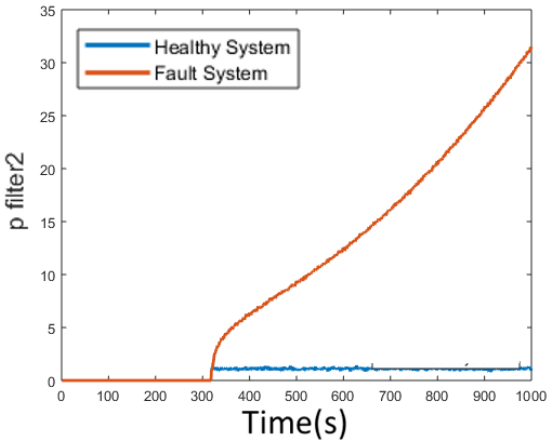
In this thesis, the fault scenario under study is filter clogging. This section provides details on integrating LS-UKF with filter clogging.

4.4.1 Process Model and Measurement Model of LS-UKF

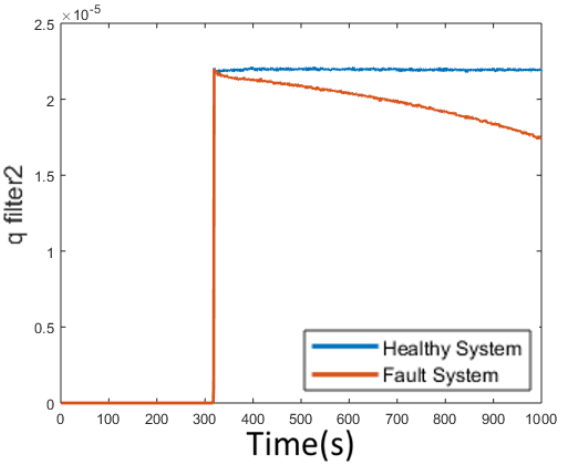
Feature Extraction

As motioned in previous section, when the filter starts clogging, more and more particles are gathered in the filter, which results in decrease of the porosity ϵ. Figure 4.3 shows the simulation results of filter porosity ϵ, pressure and flow rate when filter clogging happens without control method. The results are based on the same operation profile in section 2. The figure shows that the decrease of the porosity ϵ results in the deterioration of the filtering performance and, therefore, can be used to indicate the clogging state of the filter. For the filter model with fault being injected, the decrease of filter porosity ϵ leads to the increase of p_{Filt2} and decrease of q_{Filt2} because less water can pass through the filter, shown in Figure 4.3. This leads to increase of ratio of $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$.

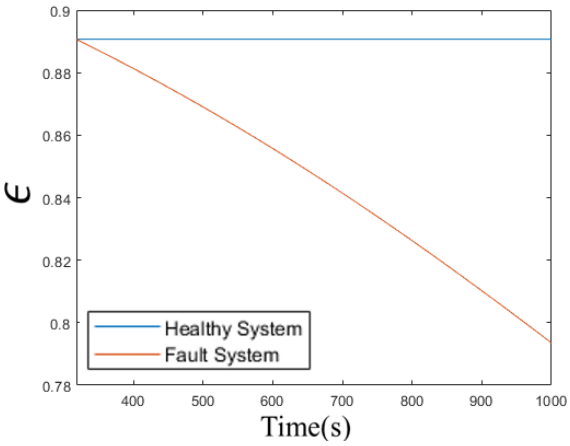
Note that flow rate or pressure alone cannot be used as measurement in diagnosis to effectively diagnosis clogging. It may work under one particular operating condition. When the operating condition changes, diagnostic algorithm based on flow rate or pressure alone cannot accurately estimate the fault state of porosity ϵ. For example, Figure 4.4 shows the estimation results when only flow rate is used as feature under three different working conditions: (i) constant flow rate without control; (ii) constant flow rate with PID-based fault-tolerant control; and (iii) reconfigured PID (in which the system has constant flow rate at the beginning of the clogging and switches to constant pressure control when the clogging becomes severe and the filter pressure reaches the safety critical threshold). Figure 4.4a shows the estimation result under operating condition of (i) constant flow rate when fault happens, which shows the estimation result is close to the ground truth. However, Figure 4.4b and Figure 4.4c, which correspond to operating conditions (ii) and (iii), respectively, indicate that the estimation results cannot accurately estimate the fault condition, especially at the



(a) Pressure of Filter 2



(b) Outflow rate of Filter 2

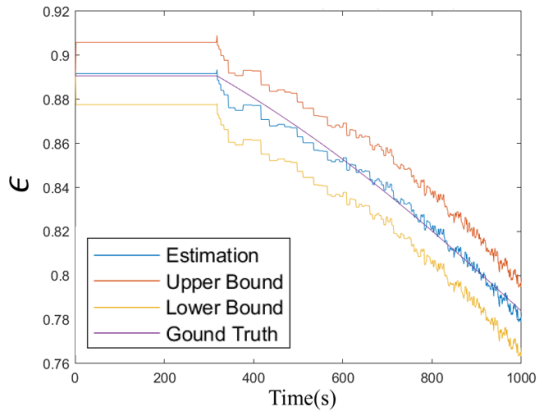


(c) Porosity ϵ of Filter 2

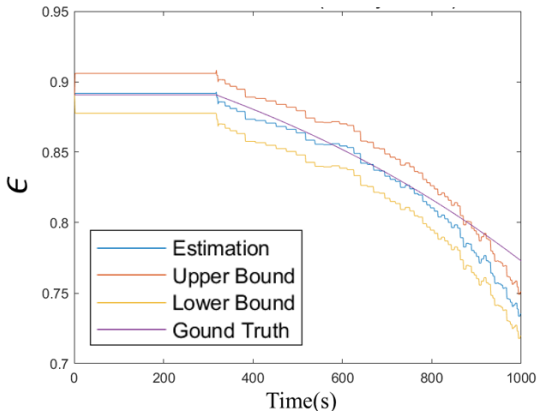
Figure 4.3 WRS filter clogging simulation result

later stage of clogging. The reason is that, when the WRS is working under constant flow rate condition for operating conditions (ii) and (iii), although the porosity ϵ decreases as more particles are trapped, the controller will maintain the constant flow rate and this constant flow rate cannot indicate the decrease of the porosity ϵ. Similar analysis applied to the filter pressure p_{Filt2}.

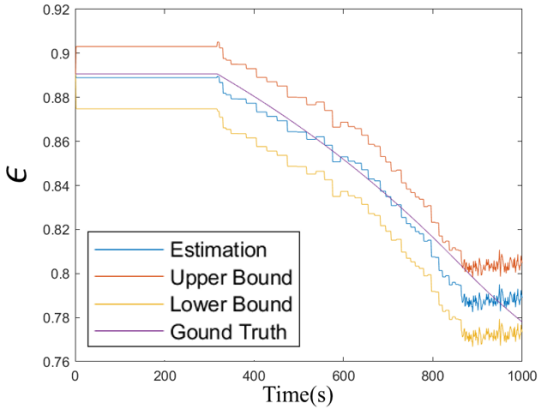
Based on the results of Figure 4.3, it is clear that the filter condition, described by porosity ϵ, can be indicated by combining flow rate and filter pressure. Figure 4.5 shows that, when the filter clogging condition increases, the porosity of the filter



(a) Constant flow rate without control



(b) Constant flow rate based PID



(c) Reconfigured PID

Figure 4.4 Porosity ϵ estimation under different operating conditions

decreases, along with the increase of $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$. That is, $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ is inversely proportion to ϵ, which can be used as a feature to indicate the decrease of porosity ϵ.

When p_{Filt2} is divided at both sides of Darcy's law, we have:

$$\frac{q_{\text{Filt2}}}{p_{\text{Filt2}}} = \frac{KA}{\mu L} = \frac{A}{\mu} \times \frac{K}{L} \tag{4.19}$$

Then we have:

$$\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}} = \frac{\mu}{A} \times \frac{L}{K} \tag{4.20}$$

- K is the permeability
- A is the cross-sectional area to flow

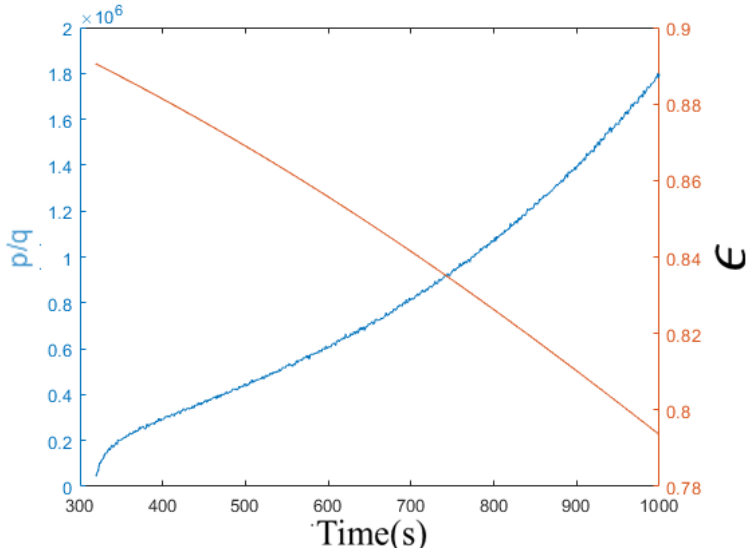
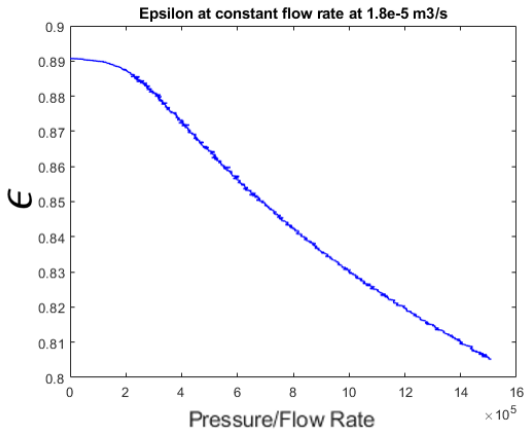


Figure 4.5 Feature $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and ϵ variation when filter clogging

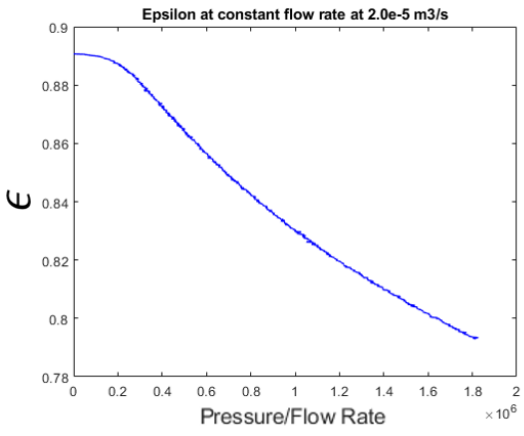
- μ is the dynamic viscosity of the fluid
- L is the cake thickness

where the cross-sectional area A and the viscosity of the fluid μ can be considered as constant, permeability K and cake thickness L can be approximated as function of porosity ϵ. When the filter is in operation and water pass through the filter, particles accumulate in the filter, which causes increase of cake thickness L, decrease of permeability K. As a result, $\frac{L}{K}$ in Eq.(4.20) increases. Since $\frac{\mu}{A}$ is considered as a constant, $\frac{\mu}{A} \times \frac{L}{K}$ increases, which shows that $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ increases. This analysis is consistent with the results shown in Figure 4.5. Both analysis of Eq.(4.20) and the result in Figure 4.5 show that $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ can be used as a feature to indicate the clogging of the filter.

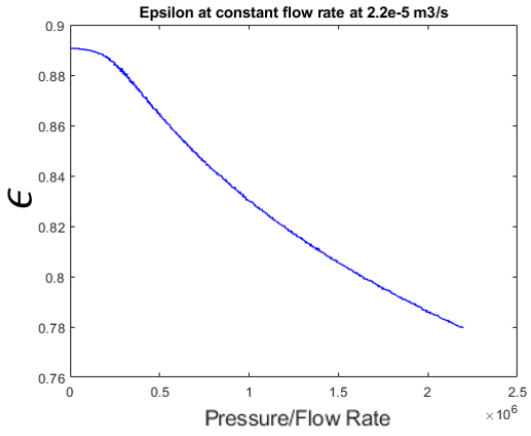
Figure 4.6 shows the results to validate this feature. The x-axis is the value of $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$. The y-axis is the value of ϵ. Figure 4.6a-c shows the ϵ variation when the constant flow rate set at 1.8×10^{-5} m^3/s, 2.0×10^{-5} m^3/s, 2.2×10^{-5} m^3/s, respectively. Figure 4.6d shows the curve at different flow rates are all comparable. In the view of analysis



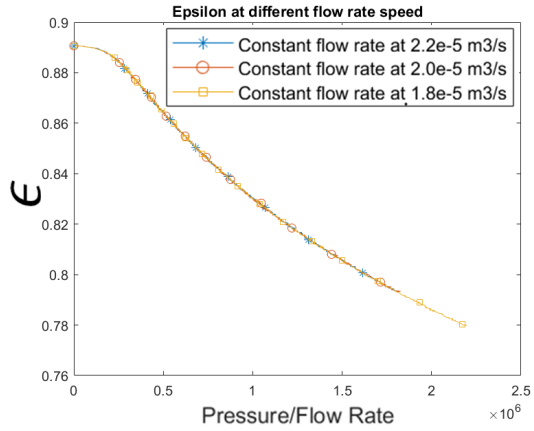
(a) Flow rate at 1.8e-5 m^3/s



(b) Flow rate at 2.0e-5 m^3/s



(c) Flow rate at 2.2e-5 m^3/s



(d) Comparison at different flow rates

Figure 4.6 Porosity ϵ versus feature when WRS at different constant flow rates

above, with the clogging of the filter, the ratio $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ is inversely proportional to the porosity ϵ curve.

Since the task is to detect clogging and estimate the clogging state indicated by ϵ, these results show that $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ is a good feature. Moreover, Figure 4.6d shows that, when the flow rate does not have significant changes, its influence on porosity degradation is very small and can be ignored. In other words, Figure 4.6d indicates that the porosity vs feature (the ratio of flow rate and pressure) curve is similar under different flow rates. This indicates that the feature extracted from filter flow rate and pressure is good for diagnosis and prognosis under different flow rates. This greatly facilities the modeling and design of fault diagnosis algorithms.

In this study, the ratio is normalized as $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}} \times 2.0 \times 10^{-5}$ and is used as the feature vector for fault identification. The normalized feature is shown in Figure 4.7. To implement diagnosis in LS framework, 100 Lebesgue states are used to evenly divides the feature vector ranging from 0 to the maximum of the feature value shown as the red horizontal lines. The maximum of the feature value can be different based on the filter parameters and consumer requirements. In this case study, the maximum of feature value is set as 50. The number and the range of Lebesgue states are chosen based on the dynamics and the range of the feature vector, respectively. The LS-based approach takes fewer samples when the fault is not severe (P1), but takes more samples when the fault becomes severe (P2), which will save the computational time, shown in Figure 4.7.

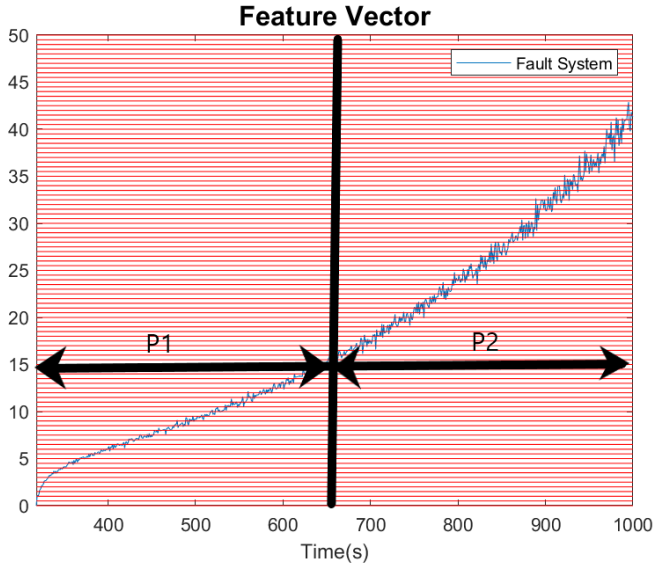


Figure 4.7 Feature vector(z_k) with Lebesgue length

Above analysis shows that the filter clogging is reflected by ϵ and the ratio of flow rate and pressure can be used as feature in fault diagnosis and prognosis to estimate the fault state. The task of fault diagnosis is to estimate the parameter ϵ from the feature given by $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$. The estimated porosity $\hat{\epsilon}$ is then used in ACM for fault mitigation.

From the above analysis, the process model and measurement model were developed as follows:

Process Model in UKF

$$x(k) = x(k-1) - C \times (q_{\text{Filt2}}) \times LBG + w_k \sim N(0, Q_k) \tag{4.21}$$

where $x(k)$ is the state ϵ, which is the porosity of Filter 2, q_{Filt2} is the flow rate of Filter 2, C is a constant, LBG is the Lebesgue length and w_k is the process noise. The process model is derived based on the relationship between flow rate q_{Filt2} and filter porosity ϵ, shown in Figure 4.8. When water pass through the filter, the particles will remain in the filter and cause the decrease of filter porosity to decrease.

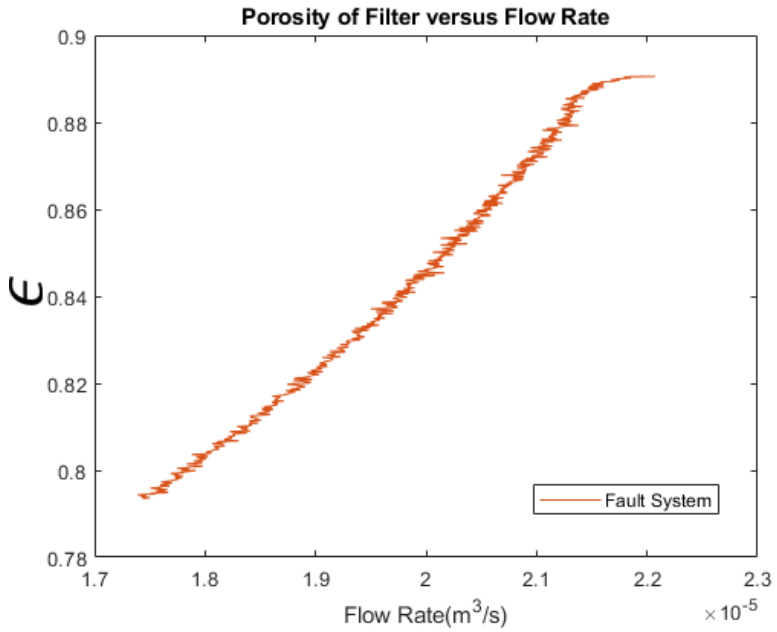


Figure 4.8 Porosity ϵ versus flow rate when filter clogging

Measurement Model in UKF

The measurement model shows the relationship between fault state, porosity of filter ϵ, and feature, the ratio of filter pressure and filter flow rate $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$. The model is given as follows:

$$z_k = h\left(x_k\right) + v_k \quad v_k \sim N\left(0, R_k\right) \tag{4.22}$$

where $z_k = \frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ is the measurement (feature) and v_k is measurement noise, $h\left(x_k\right)$ is the nonlinear function that describes the relationship between measurements and filter state. Since the relationship is nonlinear, a machine learning approach based on neural networks is used to get this model.

The structure of the neural network implemented has 1 hidden layer with 10 neurons. The input is porosity ϵ and the output is $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$. In the simulation, the fault is injected at the 317th second, and the mission ends at the 1000th second. Next, we randomly selected 70% samples as training data, 15% samples as testing data, and 15% samples as validation data. Levenberg-Marquardt algorithm in MATLAB neural network fitting toolbox is used for training. This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Figure 6.12a shows the neural network fitting result, in which the x-axis represents the input ϵ and the y-axis represents the predicted $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ from the model. Ground truth (target value) and output of the model from training dataset, testing dataset and validation dataset are represented by the blue, red and green dots. Figure 6.12b shows the errors between targets and outputs of the training dataset, testing dataset and validation dataset.

Figure 6.13 shows the modeling error histogram. The blue, green, and red bars represent the errors of model on the training data, validation data, and testing data, respectively. The x-axis represents the errors between the predicted $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and ground truth $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$, while the y-axis is the number of instances (samples) fall into these error ranges.

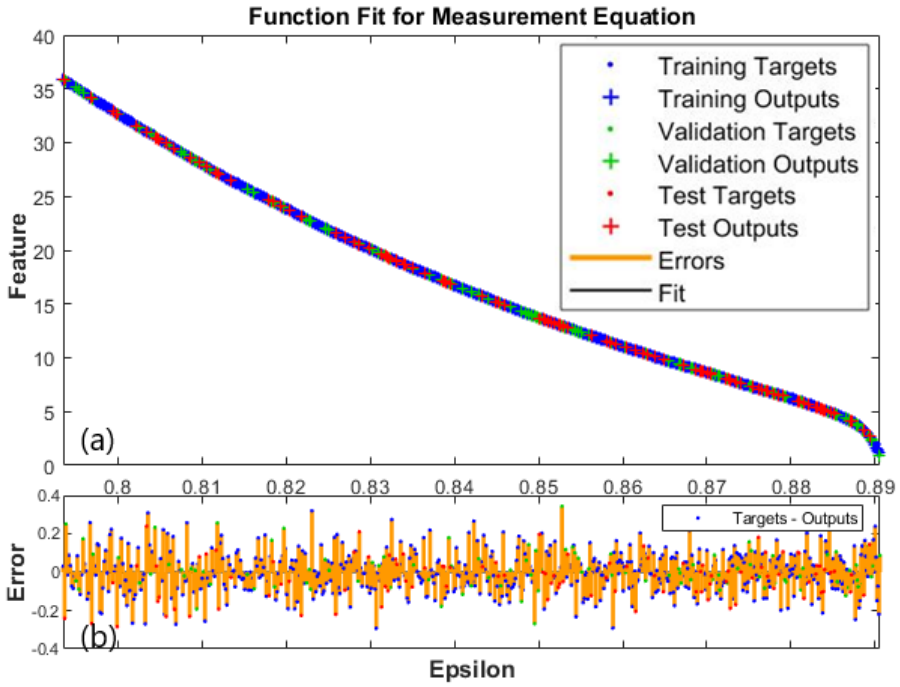


Figure 4.9 Neural network fitting result

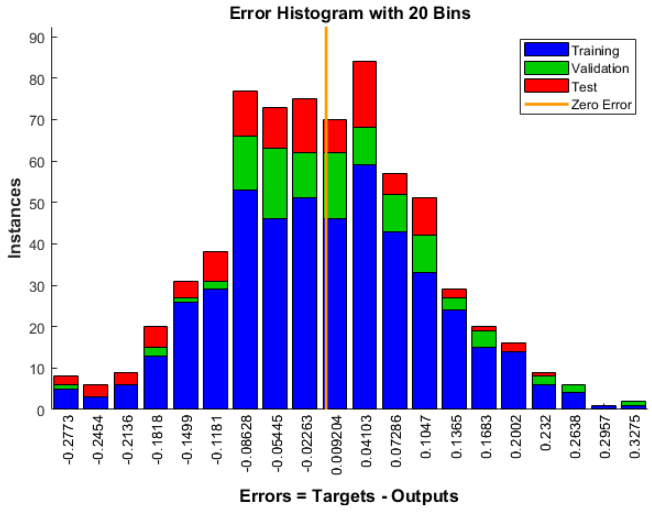


Figure 4.10 Neural network error histogram

4.5 Results Analysis

4.5.1 Fault Detection

Fault detection is conducted by comparing the baseline ϵ distribution against the real-time $\hat{\epsilon}$ distribution. Figure 4.11 shows the comparison of baseline distribution

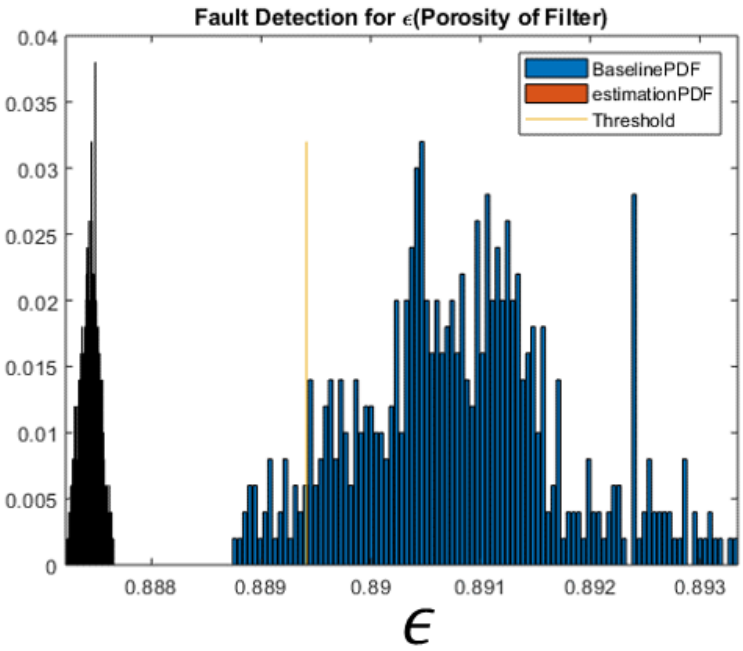


Figure 4.11 The fault detection results

and real-time distribution when the fault of clogging was detected. The baseline distribution is established from the ϵ values without clogging. In WRS, before fault occurs (the filter is in clogging-free working condition), the ϵ is assumed to follows a Gaussian distribution with mean of 0.8907 and standard deviation of 1×10^{-3}, as shown in blue bars in Figure 4.11. For detection, the false alarm rate and probability of missing detection are both set to 5%. The fault detection threshold can then be determined from the baseline distribution and the 5% false alarm rate, which is 0.9984, indicated by the yellow vertical line. The estimation PDF shows the diagnostic results at the 334th second when the fault was detected. The mean value of the real-time distribution at the 334th second is 0.8876 with standard deviation at 1×10^{-4} and its 95% confidence interval is [0.9695, 1.0009].

4.5.2 Comparison between RS and LS

In RS, the estimation of porosity ϵ is calculated every time when a new feature $(\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}})$ becomes available. However, in LS-based algorithm, the Lebesgue states are

defined as 100 states with even length ranging from 0 to 50. In the operation of the system, when a new feature becomes available, the feature is compared with the predefined Lebesgue states and the diagnosis algorithm is executed only when an event happens, i.e., the feature changes from one Lebesgue state to another one. The results of filter clogging estimation from LS-UKF are shown in Figure 4.12. Since the output of LS-UKF is a distribution, the figure shows the mean and upper/lower bounds of the estimation of porosity ϵ. In this case, the fault is detected at the 334th second. For a mission that ends at the 1000th second, there are only 44 executions to the end of the estimation process, which is very small compared with RS-UKF (1000-317=683 executions). The reduced execution times for the diagnosis algorithm benefits applications have limited computation resource, like deep space habitat.

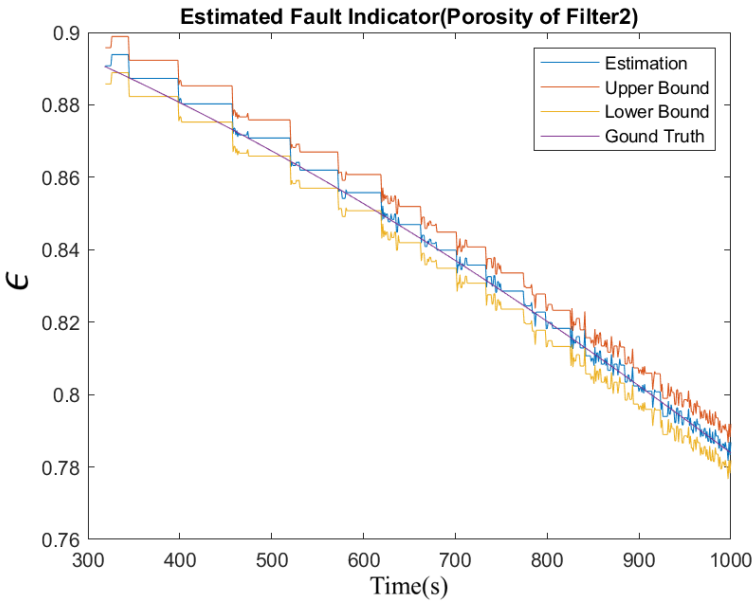


Figure 4.12 Lebesgue sampling-based fault identification simulation result

4.5.3 Fault Identification Under Different Water Quality

Due to its high modeling capability, neural network model can describe the faults under different operating conditions. To demonstrate this, the filter is simulated with

different flow rates in this case study and the neural networks model is implemented to validate the model.

When the flow rate is high, the accumulation rate of particles in the filter becomes fast and lead to fast degradation of porosity ϵ. We want to test if LS-UFK is capable of estimating ϵ under different flow rates. Figure 4.13 shows the simulation result with flow rates of 2.3×10^{-5} m^3/s, 2.0×10^{-5} m^3/s, 1.8×10^{-5} m^3/s. For this test, the flow rate is set as a constant and the PID controller (discussed in the next section) is used to maintain the flow rate as the filter condition degrades, i.e., the clogging condition becomes severe as particles accumulate.

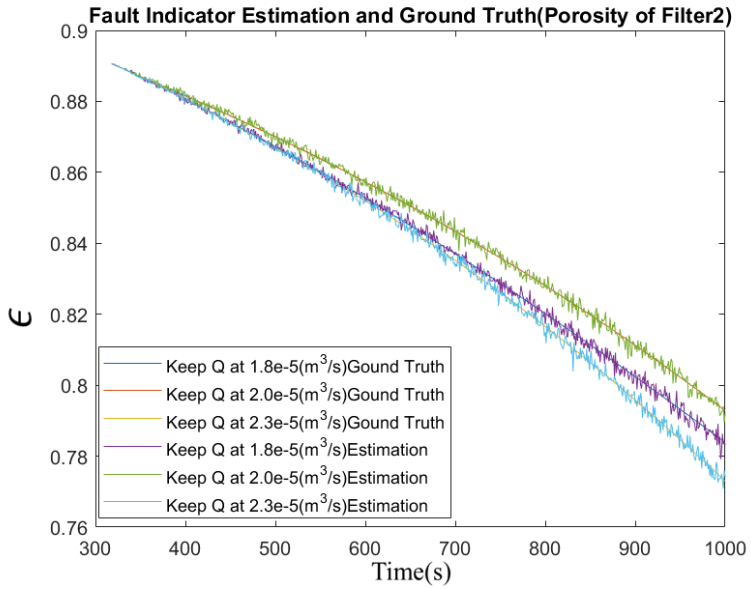


Figure 4.13 LS-UKF based fault identification tested under different flow rates

The accuracy of LS-UKF based fault identification under different flow rates is evaluated by using the Root Mean Square Error (RMSE), which is defined as:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{4.23}$$

where $\hat{y}_i$ and y_i, $i = 1, 2, \cdots, n$ are predicted value and ground truth value respectively, n is the number of observations.

In this case study, RMSE is used to evaluate the residual between the predicted $\hat{\epsilon}$ and the ground truth ϵ. Table 4.2 shows the results, which indicate that the RMSE under different flow rates are very close to 0. The simulation results indicate that the developed LS-UKF has high accuracy for the prediction of the ϵ in different operating conditions.

Table 4.2 Estimation RMSE for fault with PID at constant flow rate

Constant flow rate	RMSE
2.3×10^{-5} m^3/s	0.0012
2.0×10^{-5} m^3/s	0.0012
1.8×10^{-5} m^3/s	0.0013

4.6 Lebesgue Sampling based Fault Prognosis

4.6.1 Prognosis methodology

The diagnosis process only provides information about current system health status, i.e., does a fault exist in the system? It does not necessarily predict future health state, the propagation of a fault if exists. The latter is addressed by a prognosis module.

In the process of prognosis, the Lebesgue time model (LTM) is developed to predict the operation time reaching each predefined Lebesgue state. The estimated state PDF can not be used for prognosis directly in the developed LTM, which needs to be converted to the corresponding time distribution PDF and is used as the initial time state for prognosis once the prognosis algorithm is triggered, as shown in Figure.4.14. With the developed LTM and the initial converted time state, the operation time PDF reaching each predefined Lebesgue state is predicted, and the operation time PDF that reaches the predefined failure threshold is the remaining useful life (RUL). In the absence of future measurements, the propagation of future

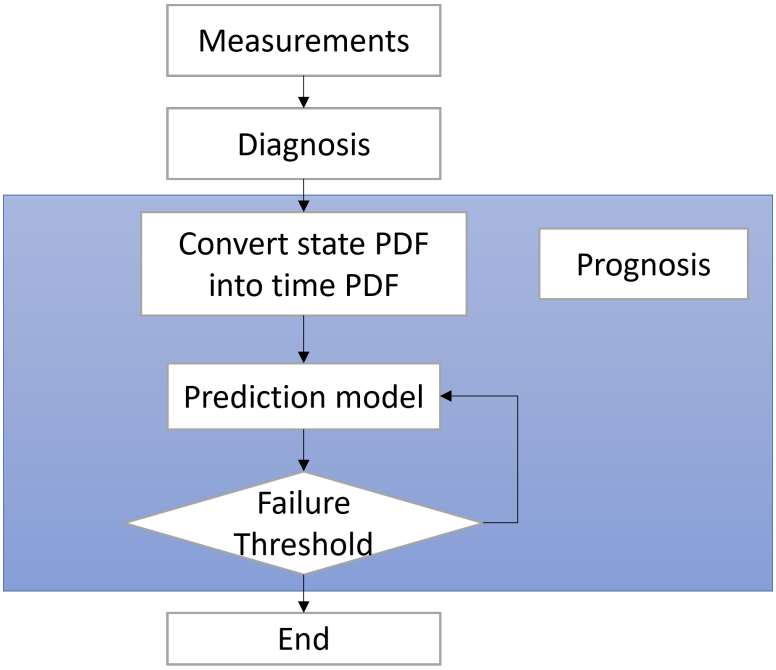


Figure 4.14 Prognosis methodology framework

states does not involve the updates from measurement as applied in a filtering framework. Compared to the RS-based prognosis, the uncertainty of LS-based prognosis caused by noise accumulation is significantly reduced since the prognosis horizon of the LS-based method has been reduced considerably. The performance of prognosis is evaluated by the α-λ metrics, defined as follows:

$$[1-\alpha]\cdot r^*(t_\lambda) \leq r(t_\lambda) \leq [1+\alpha]\cdot r^*(t_\lambda) \tag{4.24}$$

where r^* is the predicted RUL at t_k, r_t is the ground truth RUL, α is the accuracy modifier, shown in Figure 4.15.

4.6.2 Prognosis result

The Filter clogging fault scenario is illustrated as a case study for the prognosis function. After the filter clogging fault is detected, the estimated filter porosity ϵ PDF from diagnosis function is converted to the corresponding time distribution

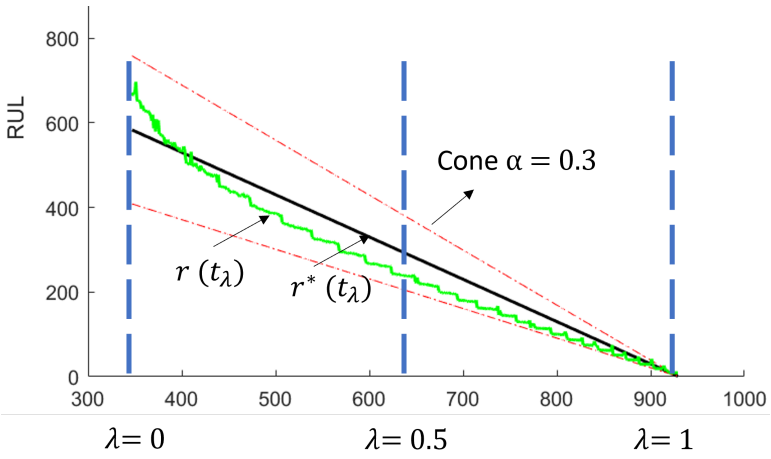


Figure 4.15 α-λ metrics

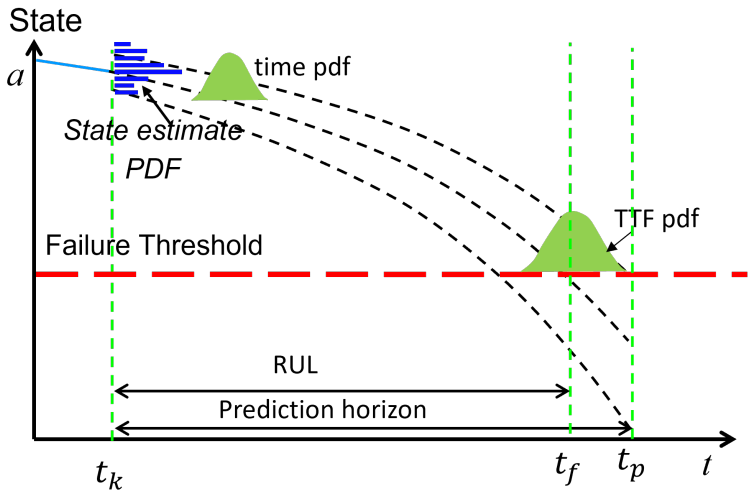


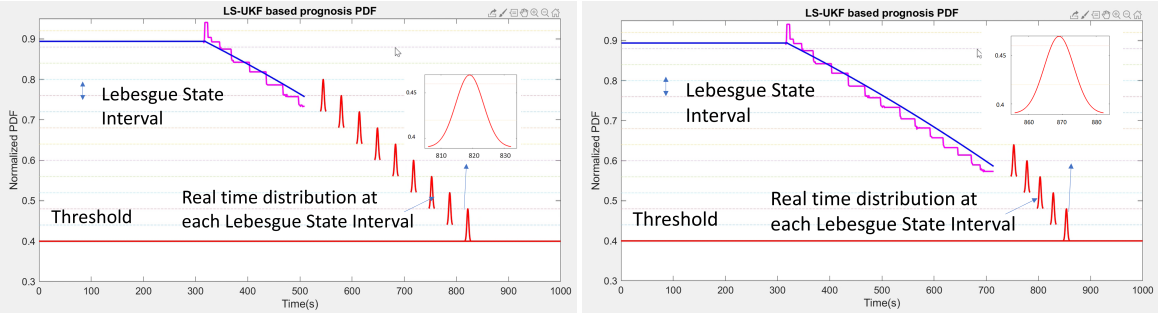
Figure 4.16 Prognosis process

PDF and is used as the initial time state for prognosis once the prognosis algorithm is triggered, as shown in Figure 4.16.

When the filter clogging happens, flow rate decreases, and pressure increases. A combination of those two measurements $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ has been used as a feature in the filter clogging fault diagnosis section. In the prognosis section, we still use $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ as a feature in the prognosis model, shown as Eq. (4.25):

$$t(k) = t(k-1) + A * \frac{p_{\text{Filt2}}}{q_{\text{Filt2}}} + w_k \quad w_k \sim N(0, Q_k) \tag{4.25}$$

The prognosis module takes the initial time distribution PDF and the feature $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ as inputs to the prognosis model to calculate the prognosis result iteratively when the predicted time distribution PDF reaches the next predefined Lebesgue state, until the predicted time distribution PDF reaches the failure threshold. The final time distribution PDF is time to failure (TTF) PDF.



(a) Prognosis result at Time 1 (b) Prognosis result at Time 2

Figure 4.17 Prognosis result

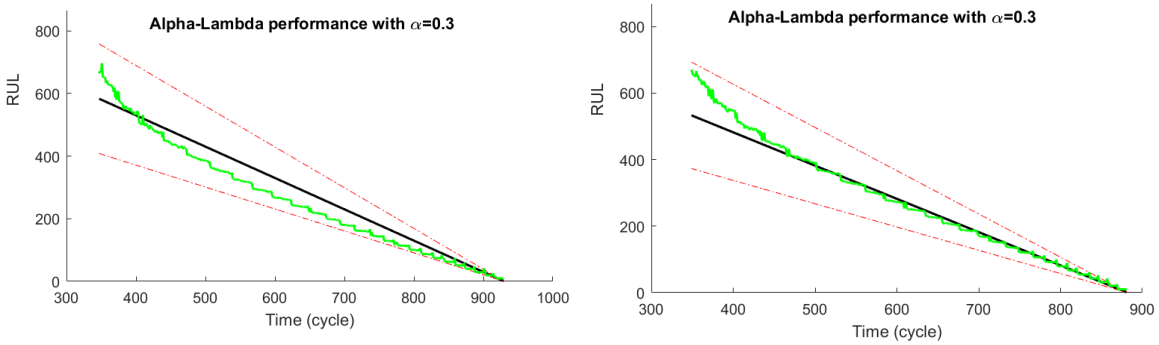
Figure 4.17 also shows the prognosis result at two different timestamps when the Filter 2 clogging happens. The number of Lebesgue states in the prognosis is calculated by the difference between the estimated filter porosity ϵ and filter porosity ϵ threshold divided by the predefined Lebesgue state length. At the 508th second, the difference between current time and TTF (921) is 413 seconds, or 413 sampling points in Riemann sampling, However, there are only 28 Lebesgue states in Lebesgue sampling. Therefore, Lebesgue sampling based prognosis save 93.23% computation. Similarly, at the 723rd second, there are only 18 Lebesgue states, compared to 198 Riemann sampling points, it saved 90.91% computation.

As for prediction accuracy, at the 508th second, TTF PDF has a mean value of 819 seconds and 95% CI (confidence interval) of [811 827]. At the 723rd second, TTF PDF has a mean value of 868 seconds and 95% CI of [856 881]. Compared with the ground truth TTF 921 seconds, the differences between the ground truth and the predictions are 82 and 53 seconds at the 508th and 723rd second, respectively. The

RDT prediction deviation from the ground truth TTF 921 seconds is 82/921×100% = 8.9% and 53/921×100% = 5.7%, respectively.

Given the above analysis, the proposed LS-based prognosis has better performance than RS-based methods in terms of computation cost and uncertainty management, which lays a solid foundation for decision-making and safety system operation.

4.6.3 Prognosis result evaluation



(a) Prognosis result evaluation under α-λ metrics

(b) Prognosis result evaluation under α-λ metrics under constant flow rate

Figure 4.18 Alpha lambda metrics

The prognosis result is evaluated by the α-λ metrics. From Figure 4.18a, we can see the prognosis result located within the α-λ metrics lower bound and upper bound, showing satisfactory results, which indicates that the prognosis algorithm can be used to predict RUL of the component.

Figure 4.18a shows that when the filter clogging fault happens and no control method is applied, the TTF is 930 seconds. However, if the flow rate is maintained as constant by a controller, which means there will be more water being transferred through the filter, more particles will accumulate in the filter. Therefore, the clogging of the filter will be faster, and the TTF is 880 seconds, as shown in Figure 4.18b. The prognostic algorithm is able to predict both scenarios accurately, as shown in Figure 4.18.

4.7 Conclusion

This section develops an LS-UKF based diagnostic and prognostic algorithms for filter clogging. The results under different operating conditions show the effectiveness of the proposed method. Since our objective is to develop ACM strategies based on diagnostic information, fault-tolerant control for ACM will be developed in the next section.

Chapter 5

Control based ACM Strategies

This chapter introduces health management approach after component fault is detected and identified. After fault diagnosis, the component fault can be mitigated in order to maintain mission success.

5.1 PID Based ACM Strategies

5.1.1 PID Controller Introduction

PID control is a classic control mechanism, which has been used in many industrial applications. A PID controller uses the error value between the desired reference signal and a measured process variable to adjust the control variable to the system and eliminate the error.

The block diagram in Figure 5.1 shows the implementation of PID controller. The error signal $e(t)$ is the difference between the reference signal $r(t)$ and the measured process variable $y(t)$. The control output is calculated by the gains of proportional K_p, integral K_i, and derivative K_d as shown in Eq. (5.1):

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(t)\,dt + K_\mathrm{d} \frac{de(t)}{dt} \tag{5.1}$$

Since the WRS is a discrete time system, the continuous time signals, $r(t)$, $y(t)$, $e(t)$, and $u(t)$ need to be discretized. The discretized signals are denoted as $r(k)$, $z(k)$, $e(k)$, and $u(k)$, where k is the index of the sampling time. Then, the discrete error signal becomes $e(k) = r(k) - z(k)$. A discrete PID-controller is obtained

by substituting the integral of the error by the sum of discrete error signal and substituting the derivative by the difference of error signal. Then the discrete PID controller can be written as:

$$u(k) = K_{\mathrm{p}} e(k) + K_{\mathrm{i}} \sum e(k) + K_{\mathrm{d}}(e(k) - e(k-1)) \tag{5.2}$$

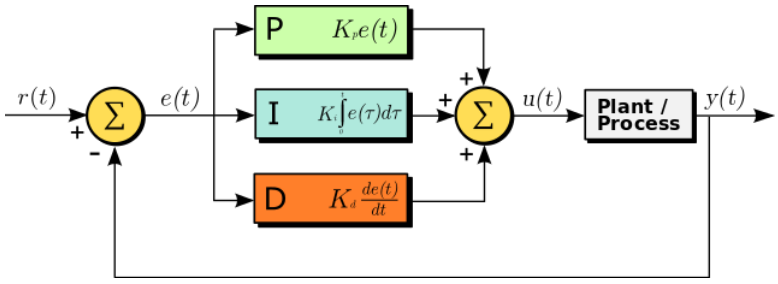


Figure 5.1 PID controller diagram

From the above equations, the function of PID terms can be summarized as follows:

Term P, is proportional to the error. In a proportional controller, steady state error tends to depend inversely upon the proportional gain K_{p}. For $K_{\mathrm{p}} > 0$, the controller output $u(k)$ increases as its input signal $e(k)$ decreases. When $K_{\mathrm{p}} < 0$, the controller is said to be direct acting because the controller output increases as the output increases. A high proportional gain K_{p} results in a large change in the output for a given change in the error. If the proportional gain K_{p} is very high, the system can become unstable. In contrast, a small proportional gain K_{p} results in a small output response to a large input error. If the proportional gain K_{p} is very low, the control action may be too small when responding to system disturbances.

Term I, is the integration of past values of error, which is proportional to both the magnitude of the error and the duration of the error. The integral is the sum of the instantaneous error over time and gives the accumulated offset that should have been corrected previously. Therefore, it has the effect of eliminating the steady-state error, but it may make the transient response worse.

Term D, takes the derivative of the process error and determines the slope of the error over time. The derivative term slows the rate of change of the controller output. It has the effect of increasing the stability of the system and improving the transient response.

The previous WRS model was built without a controller. The WRS model was an open-control system. As the first step to introduce ACM to WRS system, the PID control was added to the WRS to accommodate filter clogging fault in the system.

5.1.2 Reconfigured PID based fault tolerant control

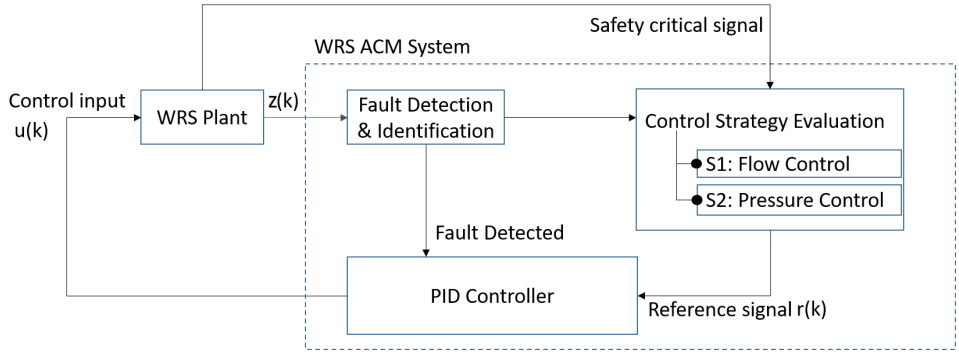


Figure 5.2 Framework of PID based fault tolerant control

The thesis considers two operating modes, constant flow rate, and constant pressure. Here, it is assumed that the water requirement is constant, which requires a constant water flow rate. In PID-based ACM, the main objective of the WRS system is to maintain sufficient water, i.e., maintain a constant flow rate of water. When the fault of filter clogging happens, the water flow rate will decrease under open control mode (before the PID-based ACM is activated). When the fault is detected, the PID-based ACM is activated to recover the constant flow rate to guarantee enough water is generated. Meanwhile, the system keeps monitoring the severity of filter clogging. When filter clogging becomes severer, the filter pressure will increase, when the filter pressure increase to a critical value (a predefined safety threshold), keeping the constant flow rate will endanger the system safety. Under this situation, the main

objective of the WRS system is to keep the system safety but work under a degraded but acceptable performance, i.e. a reduced but acceptable water generation. With this consideration, the operating mode switches to constant pressure mode. The constant pressure model under severe fault conditions will lead to the decrease of flow rate. However, it guarantees the filter pressure is under the safety threshold such that the system is not endangered.

Figure 5.2 shows the framework of PID based fault mitigation. At each discrete time step k, the fault detection and identification algorithm takes the measurement $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ ($z(k)$ in the figure) from the WRS plant to estimate the filter porosity $\hat{\epsilon}$ to detect if clogging happens in the system. Once the fault is detected, the PID based fault mitigation is activated to control the WRS. In this case, the PID controller controls the input voltage to the pump, i.e., $u(k)$ is the pump voltage. The fault detection and identification module uses the LS-UKF developed in the previous section to estimate the porosity of the filter. When the fault is detected, the PID controller starts working. As mentioned early, at the early stage of filter clogging, when the clogging is not severe, the PID is a constant flow control. During the operation of the filter, when the clogging becomes severe, the pressure is monitored as a switch signal of PID control mode. When the safety-critical signal p_{Filt2} becomes larger than a threshold, the PID control switches to constant pressure mode. Although the switch of control model from constant flow rate to contestant pressure causes the change of reference signal $r(k)$, the PID controller always works on the pump input voltage.

5.1.3 PID Simulation Case Studies

This section shows two PID simulation case studies of constant flow rate and reconfigured PID based fault tolerant control. The objective of those two case studies is to mitigate the effect of the filter clogging based on different control priorities. The operating profile of the WRS follows the one described in WRS modeling and

simulation. In the simulation, the parameters are set as follows: the particle's suspension rate is 0.14% and the normal flow rate is set as $2.2 \times 10^{-5}\text{m}^3/\text{s}$. The filter pressure limit is different for different filters. In this case study, the limit is 32 psi. The nominal voltage of the Pump is 20 V, and the maximum voltage is 24 V.

In the simulation, the parameters for the PID parameters are shown in Table 5.1. The PID parameters show in this table were tuned to achieve good tracking performance of the reference signal and stability under both constant flow rate and constant pressure control modes.

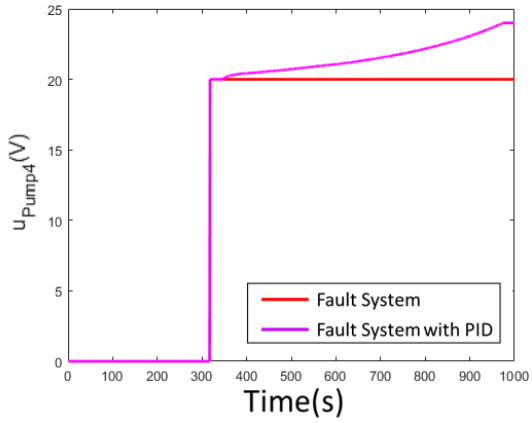
Table 5.1 Parameters for the PID controller

Parameters	Values
K_p	100000
K_i	40000
K_d	10

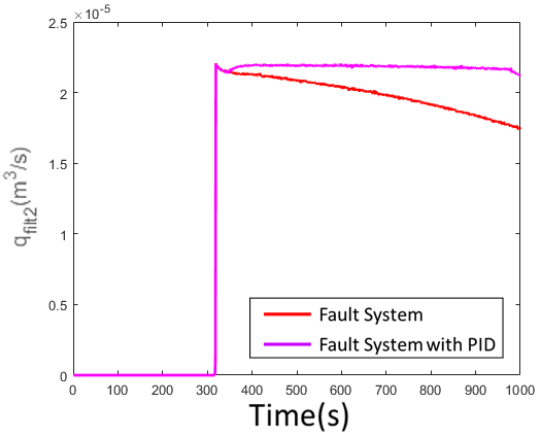
5.1.3.1 Constant Flow Rate PID based Fault Tolerant Control

The first case study is for a filter working under constant flow rate mode. As mentioned early, this control mode is activated when the fault of clogging is detected at its early stage and the fault severity is low. The main objective at this moment is to maintain the quantity of the water. To demonstrate the effects of PID based ACM, the results are shown and compared with fault WRS without PID controller. Figure 5.3 shows the simulation results of WRS filter clogging under constant flow rate PID based fault tolerant control (given by magenta curves), which are compared with the WRS filter clogging without fault tolerant control (given by red curves). Figure 5.3a shows the input signal of the Pump 4 in which the voltage at the 317th second indicates that the pump-filter system starts to operate. The operation ends at the 1000th second. Figure 5.3b-d shows measurements of the WRS of Filter 2 flow rate q_{Filt2}, pump flow rate, and Filter 2 pressure p_{Filt2} by red curves). Figure 5.3a shows the input signal of the Pump 4 q_{Pump4} in which the voltage at the 317th

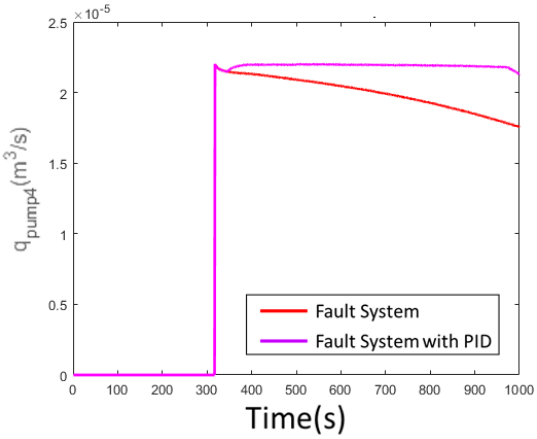
second. Figure 5.3e-g shows the parameters of Filter 2 porosity ϵ, cake thickness L, and permeability K.



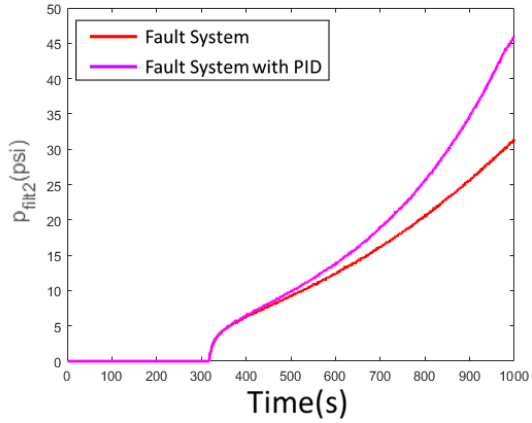
(a) Voltage of Pump 4



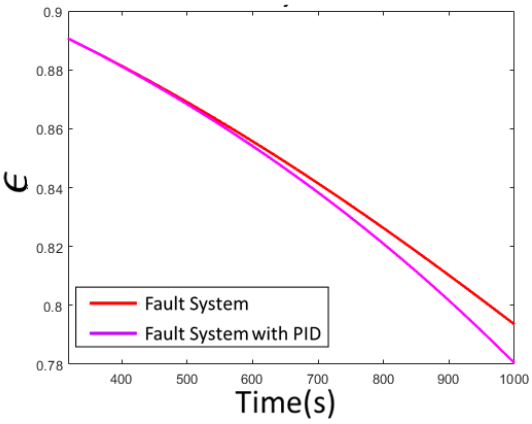
(b) Outflow rate of Filter 2



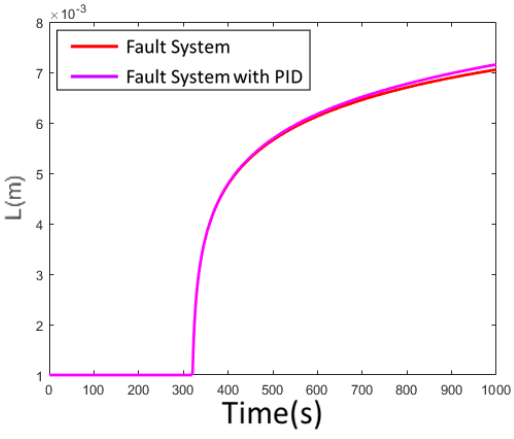
(c) Outflow rate of Pump 4



(d) Pressure of Filter 2

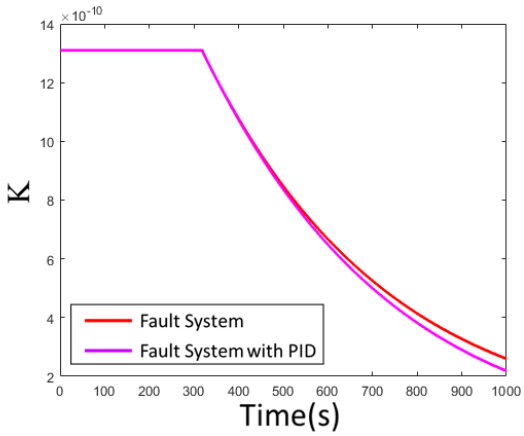


(e) Porosity of Filter 2



(f) Cake thickness of Filter 2

Figure 5.3 WRS filter clogging with PID simulation result 1



(g) Permeability of Filter 2

Figure 5.3 WRS filter clogging with PID simulation result 1

At the 317th second, the system is put into operation and the pump starts to transfer water through the filter. The input voltage of the pump is shown in Figure 5.3a. When water passing through the filter, the filter begins to clog because the particles in the water accumulate in the filter. For the original WRS without PID control to mitigate the fault, the input voltage of Pump 4 is a constant as there is no control to adjust it, Figure 5.3a. With the presence of clogging, the flow rate of Pump 4 and Filter 2 decrease as shown in Figure 5.3b and 5.3c. The clogging also leads to the pressure increase on the flow-in side and pressure decrease on the flow-out side of Filter 2, which results in the increase of pressure of Filter 2 (defined as the pressure difference on flow-in and flow-out side) as shown in Figure 5.3b. The simulation also shows the change of filter parameters with clogging. With particles trapped in the filter, the porosity and permeability of the filter decrease, as shown in Figure 5.3e and Figure 5.3f. Meanwhile, the cake thickness of the filter increases with the accumulation of particles, as shown in Figure 5.3f. The change of filter parameters consistent with the change of filter variables. For instance, according to Ergun's law, when cake thickness increases, the filter pressure increases. According to Darcy's law, the decrease of permeability and the increase of cake thickness are dominant, as shown in Figure 5.3f and Figure 5.3g. Although the filter pressure increase, due to

the dominant influence of permeability and cake thickness, the flow rate of Filter 2 decreases as shown in Figure 5.3d.

For the same simulation scenario, when the PID-based ACM is included, the PID controller is activated when the fault is detected. At the 317th second, the system is put into operation and the pump starts to transfer water through the filter. Particles start to accumulate on the bed of the filter, and the fault was detected at the 347th second. At this time, the PID controller starts to work. Based on the fault diagnosis and check if safety critical threshold reaches 32 psi, the filter clogging is not severe at this moment. Therefore, the main objective of the system is still to maintain the quantity of water. Since the clogging causing decrease of flow rate (as shown in the [317, 347] second in Figure 5.3b and Figure 5.3c), the PID adjusted the input voltage of the pump. This leads to an increase of pump voltage shown in Figure 5.3a to compensate for the decrease of flow rate. With the increase of pump voltage, the flow rate of Pump 4 increases, which also leads to an increase of the flow rate of Filter 2. This happens after the PID controller starts to work. From Figure 5.3b and Figure 5.3c, it is clear that after 347 second, the decreased flow rate of Pump 4 and Filter 2 gets back to their reference values in 349 seconds. The response under the selected PID gains are good. Moreover, with the increase of clogging severity, the destined PID control can maintain the flow rate and, therefore, meet the mission needs.

However, with the clogging of the filter, maintaining a constant flow rate leads to an accelerated increase of filter pressure, as shown in Figure 5.3d. At the end of the simulation, the pressure of Filter 2 increases to 45 psi, compared with 31 psi without PID. The maintaining of the constant also accelerate the degradation of the filter condition. In Figure 5.3e-g, the porosity, cake thickness, and permeability of the filter degrade faster with PID. At the end of the simulation, the values of porosity, cake thickness, and permeability degrade to 0.780, 0.0072, 2.183×10^{-10} compared to 0.793, 0.0071, 2.597×10^{-10}. The conclusion is that, at the early stage of the fault, the

ACM strategy focuses on maintaining water quantity but not on the system remaining useful life.

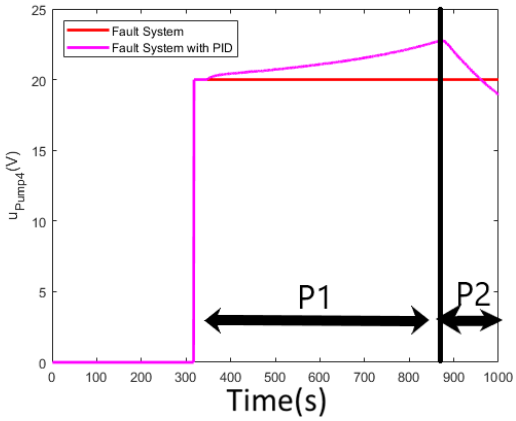
Limitation of the Approach:

This case study only considers the flow rate to maintain water quantity. As a result, at the end of the operation, the filter pressure increases significantly. Suppose that the filter pressure increases exceeds the safety threshold, it may lead to damage to the filter or the WRS. Therefore, in the operation of the WRS, the filter pressure must be regulated to keep the safety of WRS. This leads to the PID-based ACM with switching mode from constant flow rate to constant pressure. That is, the PID control is reconfigured in the operation of the system. Details are discussed below

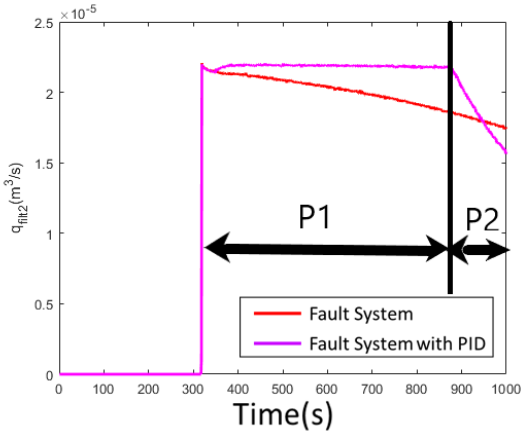
5.1.3.2 Reconfigured PID based Fault Tolerant Control

This section presents the result for reconfigured PID based fault tolerant control. Previous case study shows that when the fault is not severe, the PID-based ACM keeps constant flow rate. Although it maintains the water quantity, the filter pressure increase significantly. For the safety of the WRS, the pressure needs to be regulated. In this case study, the pressure critical threshold of 32 psi is introduced. That is, the filter pressure cannot be higher than this threshold. With this design, when the filter pressure reaches 32 psi, the PID control strategy is reconfigured from constant flow rate to constant pressure. In other word, the main objective changes from guaranteeing water quantity to maintaining system safety. Figure 5.4 shows the simulation results (given by magenta) and its comparison against the scenario without PID based ACM (given by red).

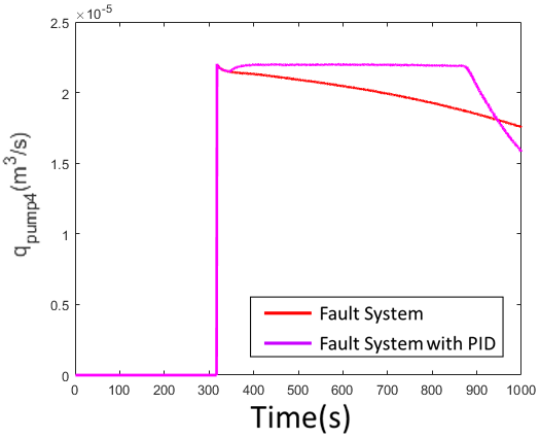
Again, the case study adopts the same operating mode. The system is put into use at the 317th second, as shown in Figure 5.4a. The P1 stage shown in the figure is the constant flow rate. In this stage, the system works exactly as described in previous case study. With the clogging of filter, the PID controller can keep the flow



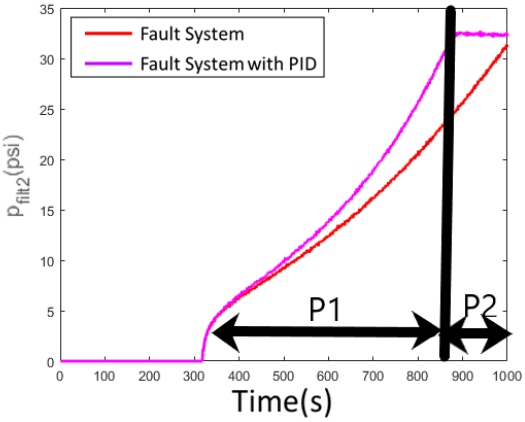
(a) Voltage of Pump 4



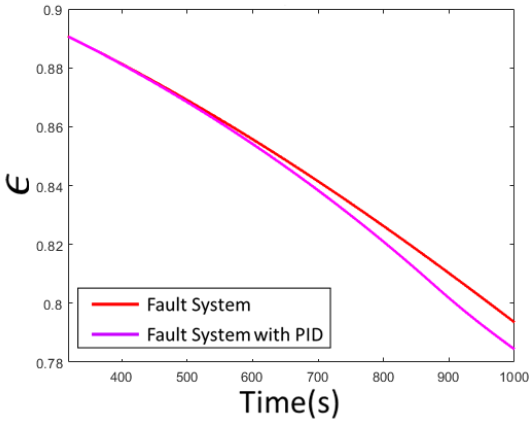
(b) Outflow rate of Filter 2



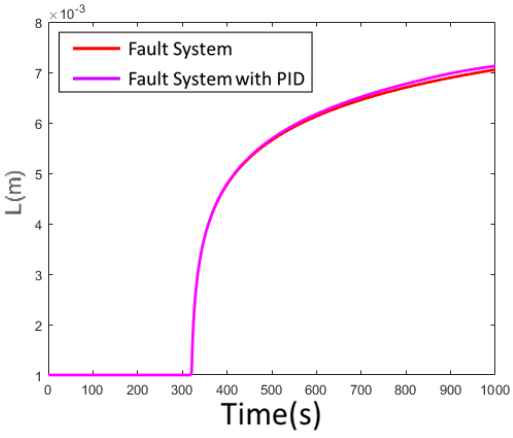
(c) Outflow rate of Pump 4



(d) Pressure of Filter 2



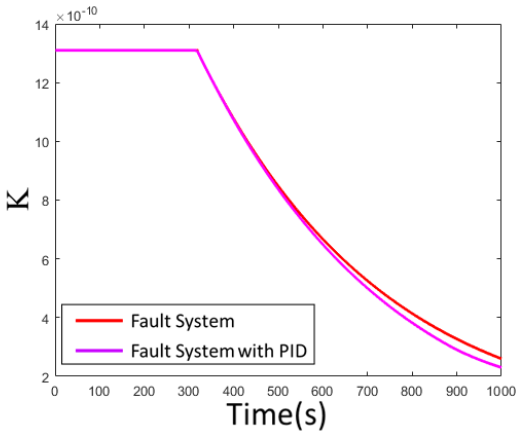
(e) Porosity of Filter 2



(f) Cake thickness of Filter 2

Figure 5.4 WRS filter clogging with PID simulation result 2

rate of Pump 4 and Filter 2, as shown in Figure 5.4b and Figure 5.4c. Meanwhile, the pressure of Filter 2 increases as shown in Figure 5.4d. The degradation of the filter



(g) Permeability of Filter 2

Figure 5.4 WRS filter clogging with PID simulation result 2

parameters, porosity, cake thickness, and permeability are shown in Figure 5.4e-g. At the 871st second, the pressure of Filter 2 reaches the safety critical threshold of 32 psi. Then, the control strategy switches from constant flow rate to constant pressure, the P2 stage. When the operation moves into the P2 constant pressure mode, the Filter 2 pressure keeps as a constant, as shown in Figure 5.4d, which keeps the pressure under threshold to guarantee the system safety. With this objective, the input voltage of Pump 4 is decreased. The reason is that, with the clogging becomes severe, the pump should decrease work load to reduce the pressure. Consequently, the flow rates of Pump 4 and Filter 2 decrease. This is acceptable to the system because, at this fault severity, the water quantity has to be reduced to maintain the system safety. Note that even with the decrease of water flow, the filter keeps degrading, which can be observed from filter parameters of porosity, cake thickness, and permeability. It is easy to understand because the filter is still in operation and more particles are trapped in the filter.

With this reconfigured control, the system works more efficient and has a longer life. This can be observed by comparing the results from the two case studies. The comparison can be observed from Figure 5.4c-d, and are summarized in Table 5.2.

Table 5.2 PID simulation results evaluation at the end of simulation

	Fault only	Constant flow based PID	Reconfigured PID
ϵ	0.7935	0.78	0.785
ϵ decrease percentage	11.37%	12.43%	11.87%
Average flow rate (m^3/s)	1.9869×10^{-5}	2.179×10^{-5}	2.179×10^{-5}
water produced (m^3)	0.0136	0.0149	0.0145
Pump voltage(V)	20	21.615	21.1122
Energy cost(J)	2.7320×10^4	2.9526×10^4	2.8839×10^4

At the end of the mission at 1000 second, the porosity of filter is 0.78 in constant flow rate, (decreased by 12.43% compared with the value under normal condition), which is also smaller than that in reconfigure) PID at 0.785 (decreased by 11.87% compared with the value under normal condition. The reason is that, during the constant pressure period (P2), the flow rate decrease, and fewer particles accumulate in the filter. Therefore, filter porosity ϵ degradation rate decreases. Similar analysis for the cake thickness L and permeability K. That is, the filter life is extended with this reconfigured PID. However, the water quantity from the reconfigured PID is reduced. The constant flow rate case generates water average flow rate at 2.179×10^{-5}m^3/s (0.0149 m^3 water produced during mission) in this mission, while the reconfigured PID only generates water average flow rate at 2.1219×10^{-5}m^3/s (0.0145 m^3 water produced during mission). Moreover, the energy consumption is also different, the constant flow rate used average voltage at 21.6150 V (2.9526×10^4J energy usage during the mission), while the reconfigured PID used average voltage at 21.1122 V (2.8839×10^4J energy usage during the mission).

The previous two case studies demonstrate the effectiveness of the PID-based ACM. However, the control strategy is only reactive. That is, use fault information as a trigger. The ACM control strategy itself does not make use of the diagnostic information. For example, the detection of a clogging fault is only used to activate the PID-based ACM. The control strategy in this design is presumed. With diagnostic information about fault severity being available, it is desirable to integrate this

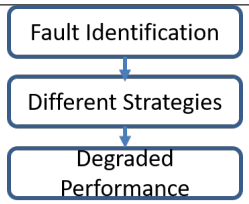
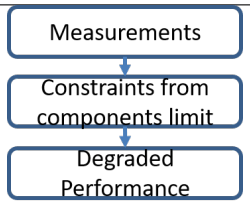
information in ACM strategy to achieve optimal fault mitigation results. To this end, it is desirable to integrate an MPC-based ACM strategy in WRS and this will be discussed in the next section.

5.2 MPC based Fault Mitigation

In the last section, a PID-based fault-tolerant control was introduced in the WRS for fault mitigation. However, PID-based control has a few limitations: First, it only makes the system follow a reference signal, which makes it not suitable for multi-objective optimization. It may be possible but very challenging to use more than one closed-loop PID controller to satisfy the output performance when the multi-input multi-output (MIMO) system have interactions between inputs and outputs. For example, a change in the first output also affects the second output in a MIMO system, which is a challenging task to design the two PID control loops because the two control loops are coupled. It is more challenging to design large systems, as they require tuning too many controller gains. In our application, it is desirable to include filter life, energy consumption, control signal variation in the optimization. This requires a multi-objective optimization approach. Second, PID-based control is a reactive approach, in which the fault detection is only used as a trigger signal to activate different mitigation strategies at different fault severity levels. It is more desirable to develop a proactive mitigation strategy in which the fault diagnosis information can be used directly to obtain the optimal mitigation strategy. To address this problem, this section develops a model predictive control (MPC)-based fault-tolerant control with prediction capability to follow the reference trajectory. Different from PID-based approach, MPC-based approach is able to integrate different objectives for multi-objective optimization and at the same time, consider different constraints in the control and the constraints are incorporated explicitly. MPC formulates the control problem as a constrained optimization problem [51]. For these

reasons, MPC approach is often easy to implement and maintain. MPC has the ability to anticipate future events and can take control actions accordingly. This makes MPC an excellent control strategy for a diagnostics and prognostics enhanced ACM system. The comparison between the reconfigured PID controller versus the MPC controller is shown in Table 5.3.

Table 5.3 Comparison between reconfigured PID controller and MPC

	Reconfigured PID based	MPC based
Diagram		
Pros	Reference signal based on the required of crew or system itself	Constraints based on the physical limitation of the system
Cons	Human involved	Required more computation effort

MPC is a control method that takes the current state of the plant as the initial states and defines a finite horizon open-loop optimization problem. Optimization produces an optimal control sequence by minimizing the cost function, in which the first value in the sequence is used as the input to the system and the rest are discarded. An essential advantage of MPC is its ability to generate optimal solutions with constraints on control inputs and states. In some applications with crucial constraints, MPC has been widely used for its efficiency reaching operating points or working under predefined state and input constraints. As for the deep space habitat, the components are also precious and the replacement of the component requires a significant amount of resource. Therefore, MPC can be adopted into deep space habitat applications.

Figure 5.5 shows an MPC process. In this figure, the measured past output are the measurements of WRS subsystem (q_{Pump4}, q_{Filt2}, q_{FO1FT2}), the predicted future outputs are obtained from the prediction model, and the target output or set-point

are the desired value of the output. In this case study, the target output is the reference pressure of FO Module 1. The manipulated variable is the control output, which is the voltage of the Pump 4 u_{Pump4} in this case study. Prediction horizon p is the number of future control intervals during which the MPC controller is evaluated by the prediction model and optimizer. Prediction horizon p defines how far MPC looks into the future. The control horizon, m, is the number of control sequence to be optimized in the optimization process. The control horizon is often set as a value between 1 and the prediction horizon p.

The objective of MPC is to minimize the cost function (objective function) process. At time k, the controller signal is calculated by optimizing the cost function in a relatively short time horizon $[k, k+p]$ based on current system states and the predicted system trajectory (from a dynamic model). When the control signal is applied to the system, only the first step of the control strategy is implemented, while the remaining ones are discarded. At the next time step $k+1$, system state estimates are updated with measurements from the system at $k+1$ and the calculations are repeated. This yields a new control strategy and a new predicted state path. The prediction horizon keeps being shifted forward and, for this reason, MPC is also called receding horizon control.

5.2.1 Implementation of MPC in WRS

In a control problem, the goal of the controller is to calculate the input to the plant such that the plant output follows a desired reference. MPC computes the input by predicting the future. MPC mainly includes two parts: prediction model and optimizer. MPC uses prediction model (also known as plant model) to make a prediction about the future plant output behavior. The optimizer is used to ensure that the predicted future plant output tracks the desired references. Figure 5.6 shows the block diagram of WRS with MPC. In the diagram of MPC, the inputs of the

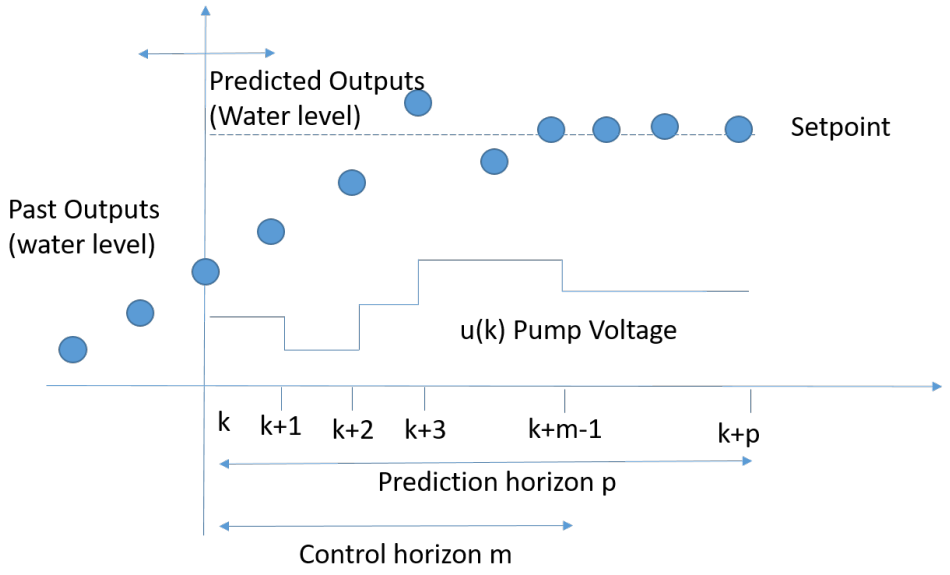


Figure 5.5 MPC process [44]

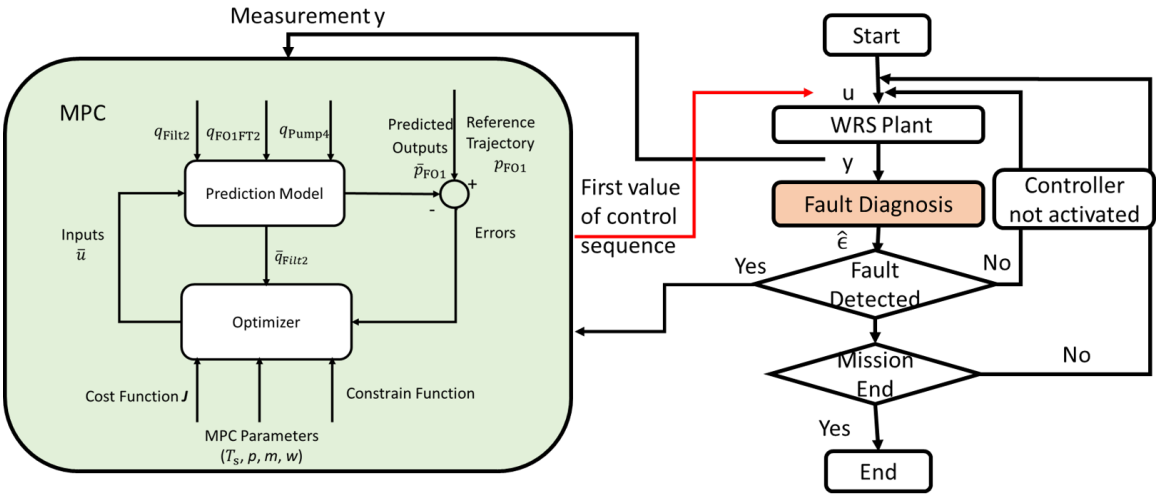


Figure 5.6 MPC with ACM framework

prediction model are the measurements of WRS (q_{Pump4}, q_{Filt2}, q_{FO1FT2}). The output of the prediction model is the predicted outputs during the prediction horizon p. The MPC controller uses the prediction model to find the best predicted path that is the closest to the reference. So it simulates multiple scenarios in the prediction horizon p systematically by using the optimizer of MPC. The inputs of the optimizer are predicted variables from the prediction model and the cost function J. The predicted variables include $\bar{q}_{\text{Filt2}}$, the error between $\bar{p}_{\text{FO1}}$ and p_{FO1}. Cost function J includes

MPC parameters include control horizon m, prediction horizon p, sampling time T_{s}, and weights w. By solving an optimization problem, the optimizer minimizes the error between the reference p_{FO1} and the predicted outputs $\bar{p}_{\mathrm{FO1}}$. It also minimizes Filter 2 degradation and control signal variation from one time step to the next. If Filter 2 degradation is fast, the filter life can be enhanced by increasing the weights on filter life. If the control signal variation is too large, it can be reduced by increasing the weight for fluctuation of the input signal. The cost function J of this optimization problem includes all these terms and is represented as a weighted squared sum of the predicted errors, control signal variation, and Filter 2 degradation. While minimizing the cost function J, the optimizer of MPC makes sure that the input constraints and state constraints are within the safety thresholds. The output of MPC is the input voltage to the pump u_{Pump4}.

At each discrete time step k, the fault detection and identification algorithm takes the measurement $\frac{p_{\mathrm{Filt2}}}{q_{\mathrm{Filt2}}}$ from the WRS plant to estimate the filter porosity $\hat{\epsilon}$ to detect if clogging happens in the system. If no fault happens, MPC is not activated. Once the fault is detected, the MPC based fault-tolerant control is activated to control the system based on the constraints, objective function, and prediction model. When the output of the MPC, which is the voltage to the Pump 4 u_{Pump4}, increases, the pressure of Pump 4 increases, which aims to compensate the flow rate decrease caused by the Filter 2 degradation. This process continues until the mission ends. By including fault diagnosis information in the cost function, MPC is able to take advantage of its preview ability to balance the water production and filter life.

5.2.1.1 Objective function

By solving an optimization problem, the optimizer minimizes the cost function J, which represents the sum of error between the reference trajectory and the predicted outputs in the prediction horizon p while all the parameters and variables satisfying

their constraints. The objective function J is the most important part of the optimizer since it considers the health condition of the filter, and system performance (water production and the acceleration rate of the pump input voltage). The influence of these factors is adjusted by changing their corresponding weights in the objective function.

The optimal control problem is to find an admissible control sequence $u_k, k = 0, 1, \ldots, m$ (m indicates the control horizon) such that the cost function is minimized, and the constraints are satisfied. The problem is defined as below:

Consider the stabilization of the discrete time-invariant nonlinear system of the form:

$$x(k+1) = x(k) + f(x(k), u(k)), \quad x(0) = x_0 \tag{5.3}$$

where $x(k)$ are the states of the pump filter subsystem and x_0 are initial states, $u(k)$ are the inputs of the subsystem, $u(k)$ and $x(k)$ are subject to input and state constraints:

$$u(k) \in U, \quad \forall k \geq 0 \tag{5.4}$$

$$x(k) \in X, \quad \forall k \geq 0 \tag{5.5}$$

where U and X are input constraints and state constraints respectively.

Find

$$\min_{\bar{u}(\cdot)} \mathrm{J}(x(k), \bar{u}(\cdot)) \tag{5.6}$$

$$\text{subject to:} \quad \bar{x}(h+1) = \bar{x}(h) + f(\bar{x}(h), \bar{u}(h)), \bar{x}(h) = x(k) \tag{5.7}$$

$$\overline{u}(h) \in U, \forall h \in [k, k+m] \tag{5.8}$$

$$\overline{u}(h) = \overline{u}\,(k+m)\,, \forall h \in [k+m, k+p] \tag{5.9}$$

with the cost function of

$$J(x(k), \bar{u}(\cdot)) := \sum_{k}^{k+p} F(\bar{x}(h), \bar{u}(h)) \tag{5.10}$$

where p and m are the prediction and the control horizon with $m \leqslant p$, $\bar{u}(h)$ represents internal controller sequences obtained from the optimizer by minimizing the cost function J, and $\bar{x}(h)$ is the internal state sequences acquired from Eq. (5.6) driven by the input signal $\bar{u}(h)$ sequences under the initial condition $x(k)$. The internal input sequences $\bar{u}(h)$ and the internal state sequences $\bar{x}(h)$ are used to calculate the prediction trajectory simultaneously in the time range $[k,\ k+p]$. The function $F(\bar{x}(h), \bar{u}(h))$ is defined as:

$$\begin{aligned} F(\bar{x}(h), \bar{u}(h)) = {} & (\bar{p}_{\mathrm{FO1}}(h) - p_{\mathrm{FO1}})^T Q\,(\bar{p}_{\mathrm{FO1}}(h) - p_{\mathrm{FO1}}) + (\bar{q}_{\mathrm{Filt2}}(h))^T T\,(\bar{q}_{\mathrm{Filt2}}(h)) + \\ & (\bar{u}(h+1) - \bar{u}(h))^T R(\bar{u}(h+1) - \bar{u}(h)) \end{aligned} \tag{5.11}$$

where $(\bar{p}_{\mathrm{FO1}}(h) - p_{\mathrm{FO1}})^T Q\,(\bar{p}_{\mathrm{FO1}}(h) - p_{\mathrm{FO1}})$ represents state tracking cost on pressure of FO module 1, $(\bar{q}_{\mathrm{Filt2}}(h))^T T\,(\bar{q}_{\mathrm{Filt2}}(h))$ represents the cost on the filter life degradation because higher flow rate leads to faster filter degradation, $(\bar{u}(h+1) - \bar{u}(h))^T R\,(\bar{u}(h+1) - \bar{u}(h))$ represents accumulated rate of change cost from $\bar{u}(h+1)$ to $\bar{u}(h)$, $\bar{p}_{\mathrm{FO1}}(h)$ and p_{FO1} represent the estimated and the reference values of the pressure of FO Module 1, respectively, $\bar{q}_{\mathrm{Filt2}}(h)$ represents the estimated flow rate q_{Filt2} from MPC prediction model, $\bar{u}(h)$ and $\bar{u}(h+1)$ represent the control

signals at the current and the next time instants, respectively. The control signal variation $\bar{u}(h+1) - \bar{u}(h)$ is used to reduce the fluctuation in the control signal.

These factors are weighted by positive definite matrices Q, R, and T.

- For a higher value of Q, the cost function puts more effort into keeping the pressure of FO Module 1 stable, which means the system provides the required water.
- For a higher value of R, the cost function puts more effort into reducing the fluctuation in the control signal, which makes the system operating smoothly.
- For a higher value of T, the cost function puts more effort into extending the life of the filter.

5.2.1.2 Prediction Model

The prediction model represents the dynamic and static interactions between input and output of the WRS. The case studies in this section are simulation results based on short time accelerated experiments. In these simulations, the filter clogging does not affect its following modules too much, and the optimization of the MPC takes more computation time if the whole WRS model is taken into consideration. Therefore, the pump-filter subsystem as shown in Figure 5.7 is used in the MPC, in which the filter clogging can be injected as discussed in the previous section. The hydraulic equations are used to build the prediction model, which describes the dynamics of the components.

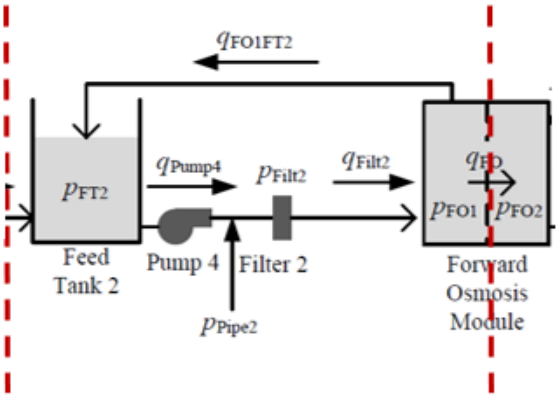


Figure 5.7 Simplified diagram of subsystem of WRS [13]

p_{FT2} = Hydraulic pressure in Feed Tank 2

p_{Pipe2} = Hydraulic pressure in Pipe 2

p_{Filt2} = Hydraulic pressure in the Filter 2

p_{FO1} = Hydraulic pressure in the FO Module 1

q_{Pump4} = Outflow rate of Pump 4

q_{FO1FT2} = Flow of water from the FO Module 1 to Feed Tank 2

q_{Filt2} = Flow rate of Filter 2

When clogging fault occurs in Filter 2, the pressure of Filter 2 (p_{Filt2}), defined as the pressure difference before and after Filter 2 increases. As a result, the outflow rate at Pump 4 (q_{Pump4}) and the flow rate of Filter 2 (q_{Filt2}) decreases. The water transferred from FO Module 1 to Feed Tank 2 is denoted by (q_{FO1FT2}). Therefore, based on these measurements and states, the filtering subsystem has three states (p_{Pipe2}, p_{FO1}, p_{FT2}) and three measurements (q_{Pump4}, q_{Filt2}, q_{FO1FT2}). Here, q_{FO} is a constant based on the assumption that the water flow through the FO module 1 is stable.

For the pump-filtering subsystem shown in Figure 5.7, the following equations describe its dynamics:

$$\dot{p_{\mathrm{FT2}}} = \frac{1}{C_{\mathrm{FT2}}} \left(q_{\mathrm{Filt1}} + q_{\mathrm{FOIFT2}} - q_{\mathrm{Pump4}} \right) \tag{5.12}$$

$$\dot{p_{\mathrm{Pipe2}}} = \frac{1}{C_{\mathrm{Filt2}}} \left(q_{\mathrm{Pump4}} - q_{\mathrm{Filt2}} \right) \tag{5.13}$$

$$\dot{p_{\mathrm{FO1}}} = \frac{1}{C_{\mathrm{FO1}}} \left(q_{\mathrm{Filt2}} - q_{\mathrm{FOIFT2}} - q_{\mathrm{FO}} \right) \tag{5.14}$$

$$q_{\mathrm{Pump4}} = u_{\mathrm{Pump4}} \left(R_{\mathrm{Pump4}} \sqrt{|p_{\mathrm{FT2}} + p_{\mathrm{Pump4}} - p_{\mathrm{Filt2}}|} \operatorname{sign} \left(p_{\mathrm{FT2}} + p_{\mathrm{Pump4}} - p_{\mathrm{Filt2}} \right) \right) \tag{5.15}$$

$$q_{\mathrm{FO1FT2}} = R_{\mathrm{FO1FT2}} \sqrt{|p_{\mathrm{FO1}}|} \operatorname{sign} \left(p_{\mathrm{FO1}} \right) \tag{5.16}$$

$$p_{\mathrm{Filt2}} = \frac{150 v_{\mathrm{s}} \mu (1-\epsilon)^2 L}{D_p^2 \epsilon^3} + \frac{1.75(1-\epsilon) \rho v_s^2 L}{\epsilon^3 D_p} \tag{5.17}$$

$$q_{\mathrm{Filt2}} = \frac{KA}{\mu L} p_{\mathrm{Filt2}} \tag{5.18}$$

Neural Network Model

When filter clogging happens, the porosity ϵ decreases and cake thickness L increases. In order to make MPC prediction model adaptive to the degradation of the component. The prediction models for ϵ and L need to be established based on the measurements $\frac{p_{\mathrm{Filt2}}}{q_{\mathrm{Filt2}}}$. To make the simulation close to real application, models need to be developed to estimate the value of porosity ϵ and cake thickness L.

Since the relationship between measurement and porosity ϵ and cake thickness L are nonlinear, neural networks are used to estimate them based on measurements

q_{Filt2} and p_{Filt2}. The same neural network structure, which has 1 hidden layer with 10 neurons, is used for estimating both porosity and cake thickness. For the neural network models, the input is $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and the output is either porosity ϵ or cake thickness L. The operation profile is the same as the one used in the previous section, in which the fault happens at the 317th second, and the mission ends at the 1000th second. For data processing, porosity ϵ and $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ after fault happens are interpolated 10 times its original samples, which provide $(1000 - 317) \times 10 = 6830$ samples. We randomly select 70% samples as training data, 15% samples as testing data, and 15% samples as validation data. Levenberg-Marquardt algorithm in MATLAB neural network fitting toolbox is used for training. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Neural Network Curve Fitting Porosity ϵ

In this case, the output is porosity ϵ. Figure 5.8 shows the neural network fitting result. In this figure, x-axis represents the neural network input $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and y-axis represents neural network output of porosity ϵ, which is compared against the ground truth (target value). The figure shows the model output from training dataset, testing dataset and validation dataset. The bottom subfigure shows the error from target to the output for each input $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$.

Figure 5.9 represents the neural network fitting error histogram. The x-axis represents the errors between the predicted porosity $\hat{\epsilon}$ and ground truth porosity ϵ. The y-axis represents how much instances (samples) located in these error areas.

Figure 5.10 shows the regression plots that display the network outputs Y (estimated porosity $\hat{\epsilon}$) to targets T (ground truth porosity ϵ) for training, validation, and test sets. By looking at the regression plot, the correlation R values are larger than 0.999. Therefore, the neural network model provides good fitting. **Neural**

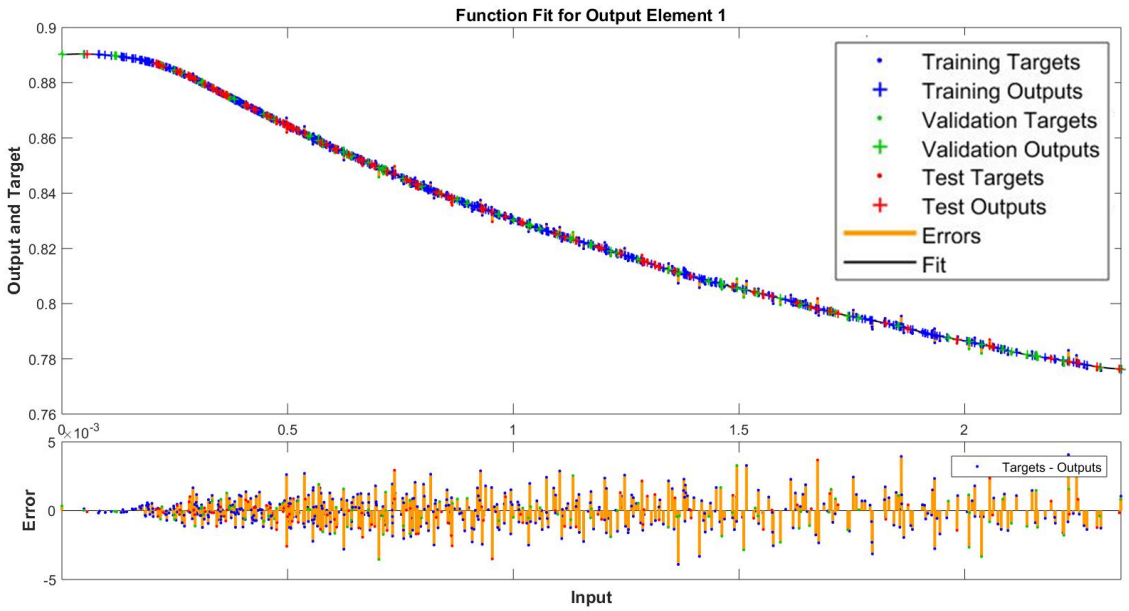


Figure 5.8 Neural network fitting plot of porosity ϵ

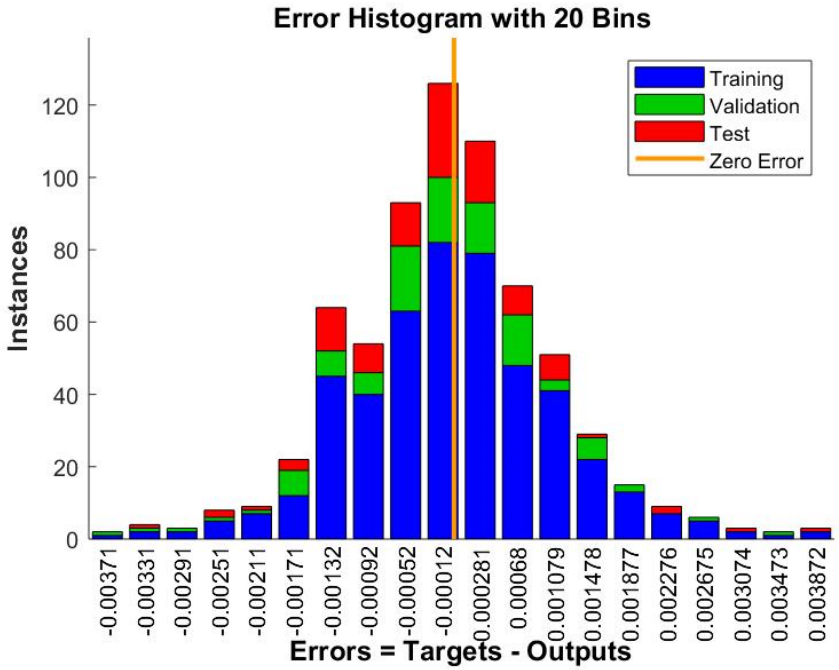


Figure 5.9 Neural network error histogram of porosity ϵ

Network Curve Fitting Cake Thickness L

In this case, the output target is cake thickness L.

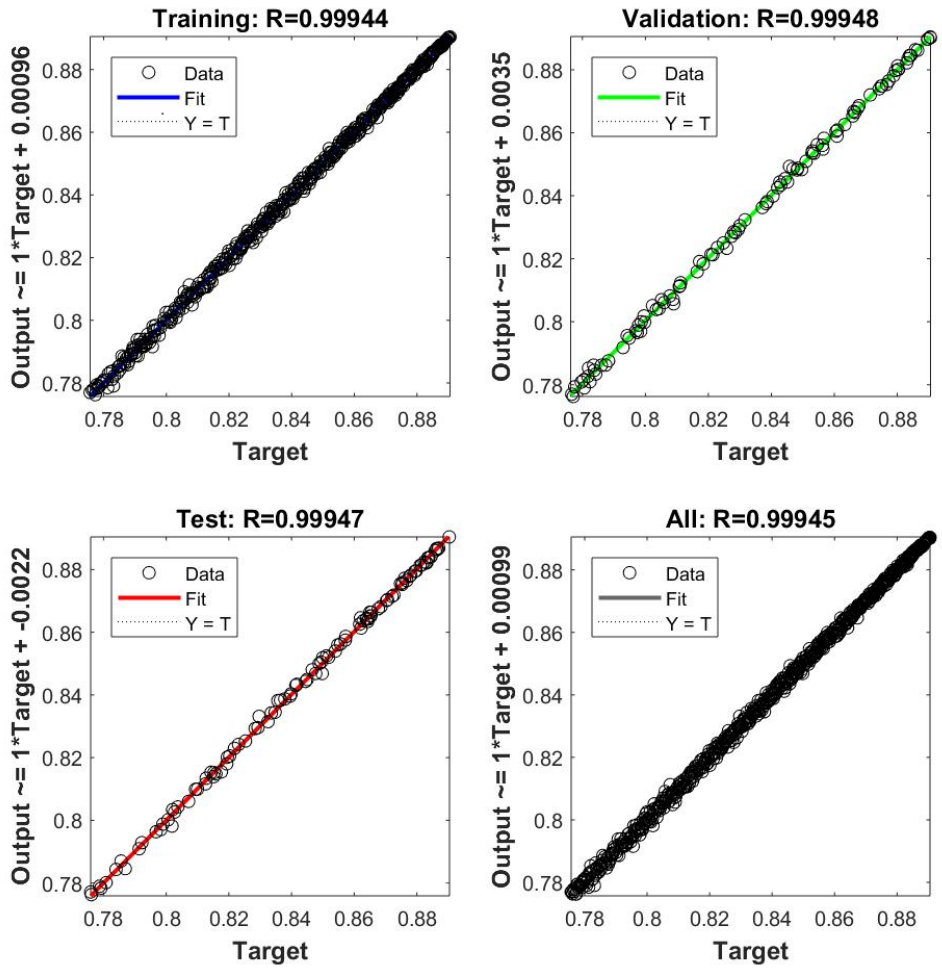


Figure 5.10 Neural network regression plot of porosity ϵ

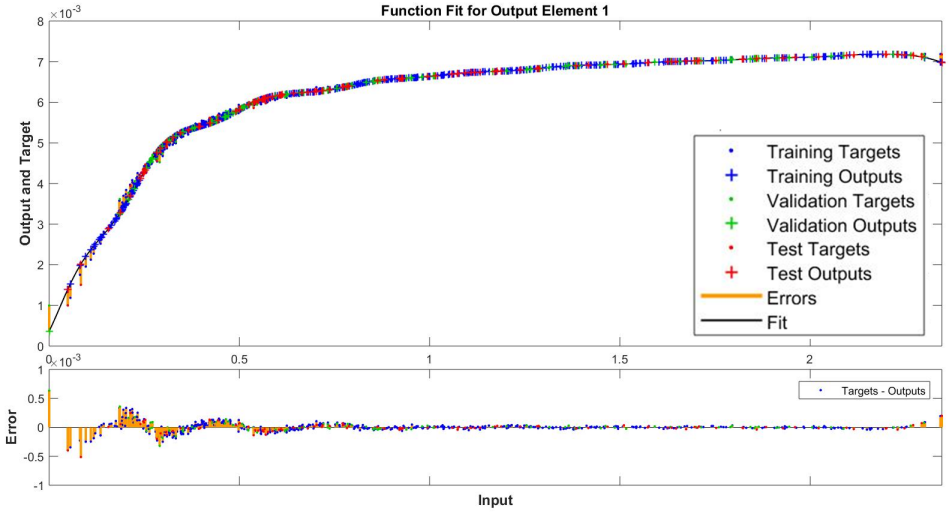


Figure 5.11 Neural network fitting plot of cake thickness L

Figure 5.11 shows the neural network fitting result. In this figure, the x-axis represents the input $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and y-axis represents neural network output of cake thickness L, which is compared against the ground truth (target value). The figure shows the model output from training dataset, testing dataset and validation dataset. The bottom subfigure shows the error from target to the output for each input.

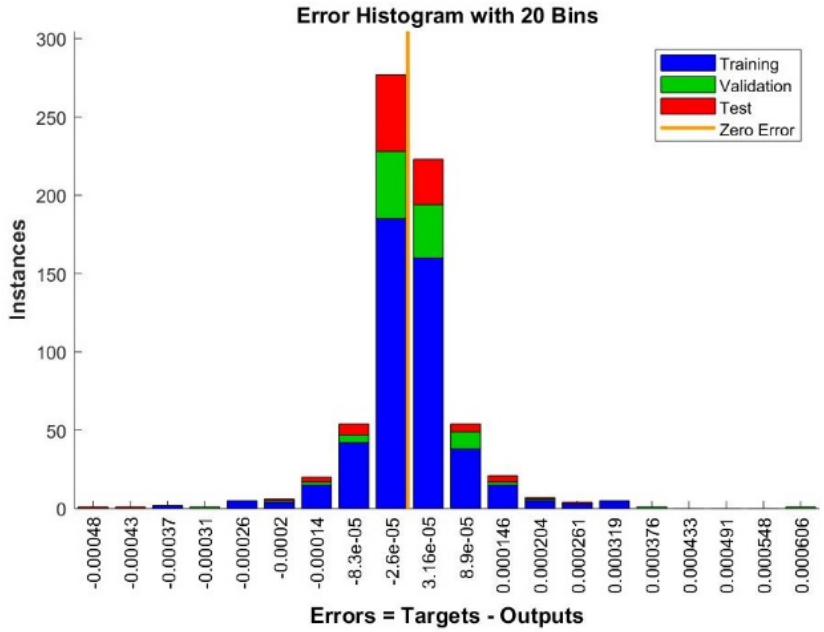


Figure 5.12 Neural network error histogram of cake thickness L

Figure 5.12 represents the neural network histogram. The x-axis represents the errors between the predicted L and ground truth L. The y-axis represents how many instances (samples) located in these error areas.

Figure 5.13 display the network outputs Y (estimated cake thickness L) with respect to targets T (ground truth cake thickness L) for training, validation, and test sets. By looking at the regression plot, the correlation R values are larger than 0.99. Therefore, we can claim this is a good fit.

Since the framework of MPC involves the WRS plant, fault diagnosis, and prediction model, it is necessary to explicitly compare the models used in each part. Table 5.4 shows the model used in each part.

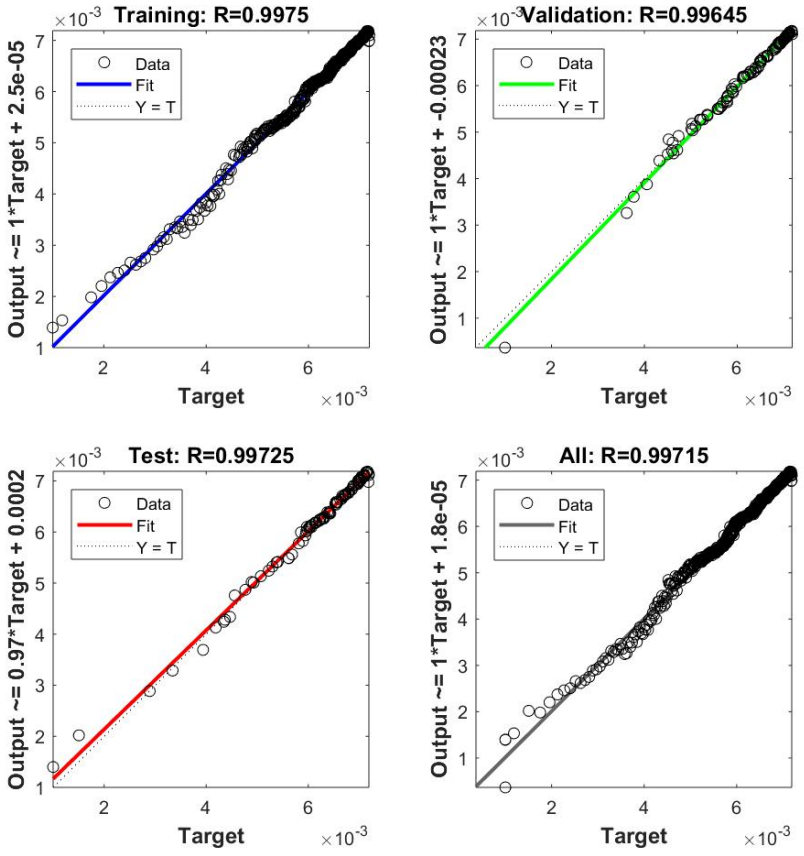


Figure 5.13 Neural network regression plot of cake thickness L

Table 5.4 Comparison of models in WRS plant and prediction model in MPC

	WRS Plant	Prediction Model
Cake thickness L	$L = \frac{\ln\left(l_1 \frac{\sum Qx}{A_f}\right)}{l_2}$	Neural network curve fitting
Porosity ϵ	$\epsilon = 1 - \frac{e^{(p_1 \frac{\sum Qx}{V_f})}}{p_2}$	Neural network curve fitting
Filter flow rate q_{Filt2}	Darcy's law	Darcy's law
Filter pressure p_{Filt2}	Ergun equation	Ergun equation
Other components in WRS	WRS model in Section 2	Eq. (5.12) - Eq. (5.16)

5.2.1.3 Constraints Function

The constraints function considers the crew requirements and the physical limitation of the components. Note that the constraint on the pressure of FO Module 1 (p_{FO1})

is based on the assumption that, at the range of [15,19] (psi), the water flow into the FO Module 1 is sufficient for the crew and other components. The constraint on the voltage of Pump 4 (u_{Pump4}) is based on the assumption that the motor voltage range is [20,24] (V).

5.2.2 MPC Simulation Results

This section shows three MPC simulation case studies of constant weight MPC based fault tolerant control. The objective of those case studies is to mitigate the effect of the filter clogging based on different control priorities. The operating profile of the WRS follows the one described in WRS modeling and simulation. In the simulation, the parameters are set as follows: the particles' suspension rate is 0.14% and the normal flow rate is set as 2.2×10^{-5} m^3/s. The nominal voltage of the Pump 4 is 20 V, and the maximum voltage is 24 V, p_{FO1} is set as 17.8 psi and initial porosity ϵ is set as 0.89 (initial condition when no degradation happens in the filter). For prediction horizon and control horizon, large values involve more computation resources. In this simulation, prediction horizon p and control horizon m are both given as 10 [52]. The values are chosen based on the simulation time of 1000 second and computational time for real time implementation.

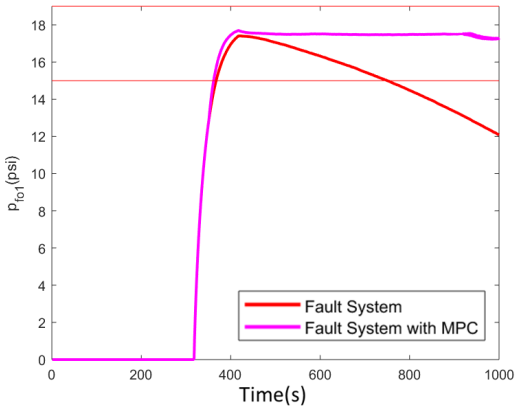
5.2.2.1 MPC Case Study I: Keeping the Pressure of FO Module 1 Stable

In this case study, the optimization function is primarily focusing on guaranteeing that enough water is generated by the WRS. This is realized by keeping the pressure in the FO Module 1 stable. Figure 5.14 shows the simulation results with constant weight MPC (given by magenta) and its comparison against the results without MPC (given by red). In the simulation, the pressure of the FO Module 1 is set as 17.8 psi to make sure the following modules have enough water to use, shown in Figure 5.14a.

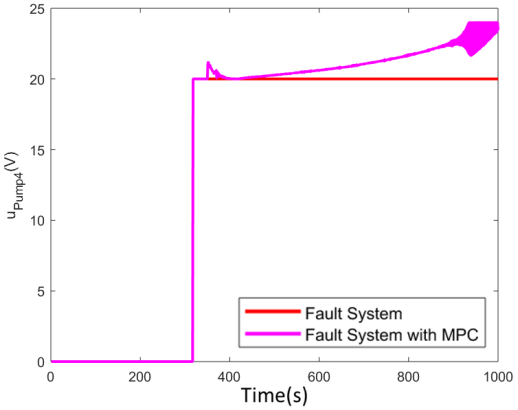
To maintain enough water in the nominal condition, the water flow in and flow out of the FO Module 1 should be the same to maintain the FO Module 1 at a steady state. The parameters in cost function are set as $Q = 0.1$, $R = 0.005$, $T = 0$. The Q is the weight of pressure of the FO Module 1 to make sure enough water can be produced.

At the 317th second, the system starts operating, and the pump starts to transfer water through the filter. The input voltage of the pump is shown in Figure 5.14b. When water passing through the filter, the filter begins to clog because the particles in the water accumulate in the filter. For the original WRS without MPC control to mitigate the fault, the input voltage of Pump 4 is a constant as there is no control to adjust it. With the presence of clogging, the flow rate of Filter 2 decreases as shown in Figure 5.14c. The clogging also leads to the pressure increase on the flow-in side and pressure decrease on the flow-out side of Filter 2, which results in the increase of pressure of Filter 2 (defined as the pressure difference on flow-in and flow-out side) as shown in Figure 5.14d. The simulation also shows the change of filter parameters with clogging. With particles trapped in the filter, the porosity decreases, as shown in Figure 5.14e. When porosity decreases, the permeability decreases and cake thickness increases. The change of filter parameter (porosity, permeability, cake thickness) are consistent with the change of filter measurements (flow rate, pressure). For instance, according to Ergun's law, when porosity decreases, the filter pressure increases. According to Darcy's law, the decrease of permeability and the increase of cake thickness leads to the decrease of flow rate of Filter 2 as shown in Figure 5.14c.

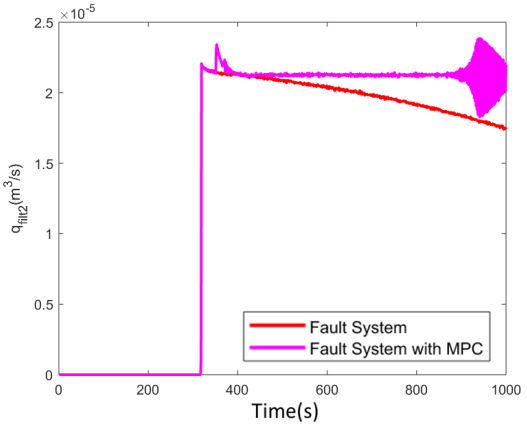
For the same simulation scenario, when the MPC-based ACM is integrated, the MPC-based ACM is activated when the fault is detected. In this simulation, the fault was detected at the 331 second. At this time, the MPC-based ACM starts to work. In Figure 5.14a, the two horizontal lines represent the constraints of the pressure in the FO Module 1, which means the pressure must stay in 15 psi to 19 psi to make



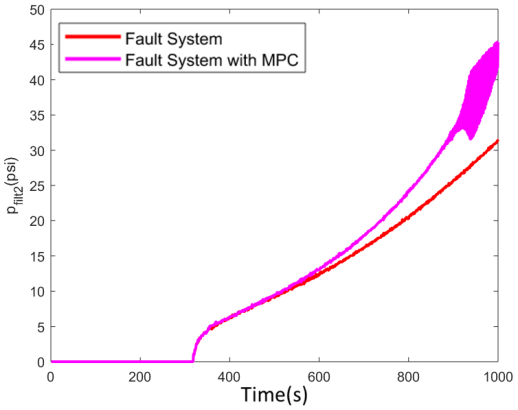
(a) Pressure of FO Module 1



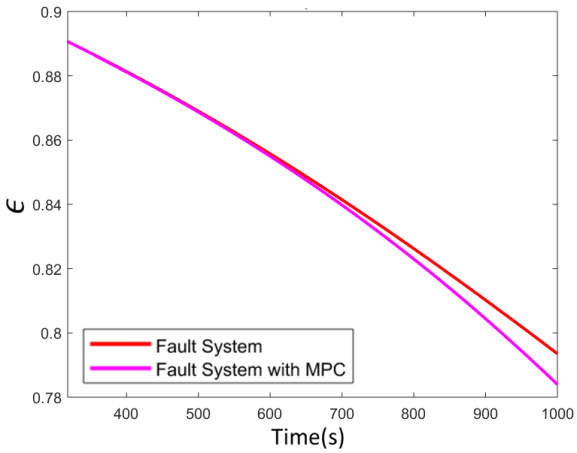
(b) Voltage of Pump 4



(c) The outflow rate of Filter 2



(d) The pressure of Filter 2



(e) Estimated porosity ϵ of Filter 2

Figure 5.14 MPC case study I simulation result

sure the following system has enough water to use. When the fault happened, the pressure in the FO Module 1 decreases to lower than 15 psi shown as the red line.

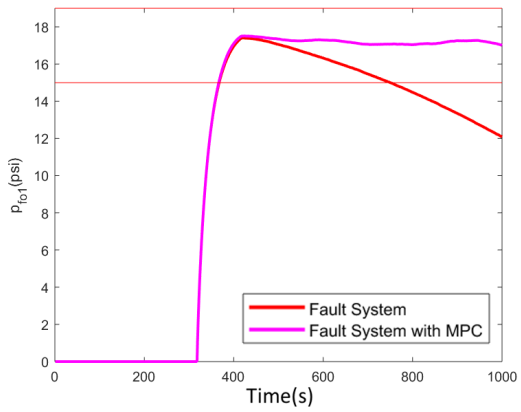
With MPC based mitigation, the pressure of the FO Module 1 maintains at the reference pressure.

Since the clogging cause decrease of pressure of FO 1 in the [317, 331] second (as shown in Figure 5.14a), after fault was detected at the 331st second, the MPC adjusted the input voltage of the pump. This leads to an increase of pump voltage, shown in Figure 5.14b, to compensate for the decrease of pressure of FO 1. With the increase of pump voltage, the flow rate of Pump 4 increases, which also leads to an increase of the flow rate of Filter 2. From Figure 5.14c, it is clear that after 361 second, the decreased flow rate of Filter 2 gets back to their reference values. The response under the selected MPC parameters shows that it can manage the contingency caused by clogging. Moreover, with the increase of clogging severity, the MPC control is able to maintain the pressure of FO1 and, therefore, meet the mission needs.

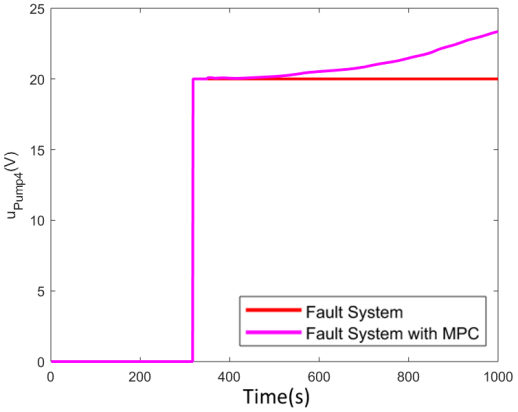
However, the simulation results in Figure 5.14b shows that the control signal has significant fluctuation. In the real situation, this fluctuation can be harmful to the pump motor. Therefore, the value of R needs to be increased to suppress the control signal oscillation. This will be studied in the next case study to reduce the effect of the fluctuation of the control signal.

5.2.2.2 MPC Case study II: Reducing the Fluctuation of the Control Signal

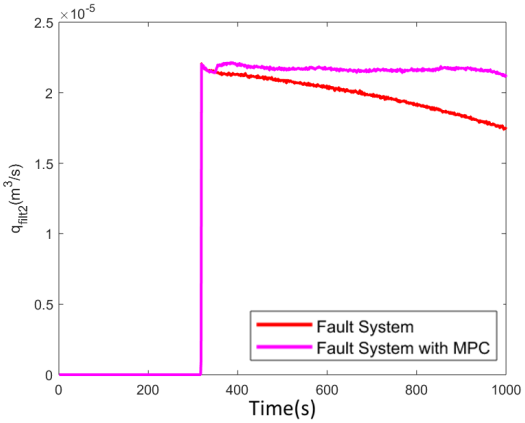
In this case study, parameters in the cost function are set as $Q = 0.1$, $R = 10$, $T = 0$. With the increase of R, the objective function also considers reducing the fluctuation of the input signal, instead of only focusing on keeping enough water production. In this case study, clogging is detected at the 331st second, which is the same as case study I as all the simulation condition in system operation and fault diagnosis are the same.



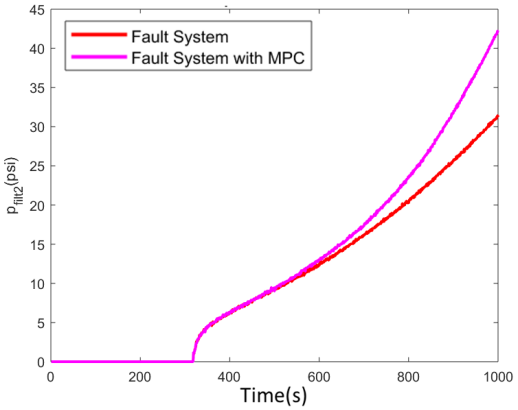
(a) Pressure of FO Module 1



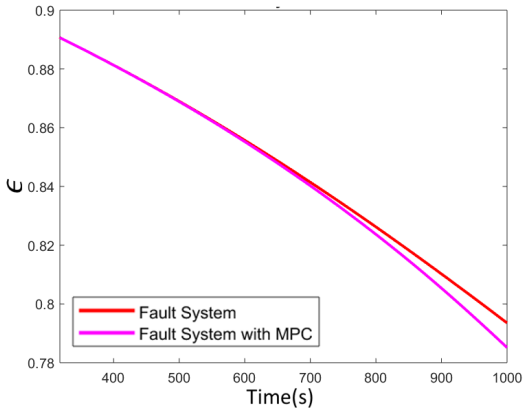
(b) Voltage of Pump 4



(c) The outflow rate of Filter 2



(d) The pressure of Filter 2



(e) Estimated porosity ϵ of Filter 2

Figure 5.15 MPC case study II simulation result

Figure 5.15 shows the simulation results of scenario with constant weight MPC (given by magenta) and its comparison against the scenario without MPC (given by red). By comparing the control signal in case study I shown in Figure 5.14b,

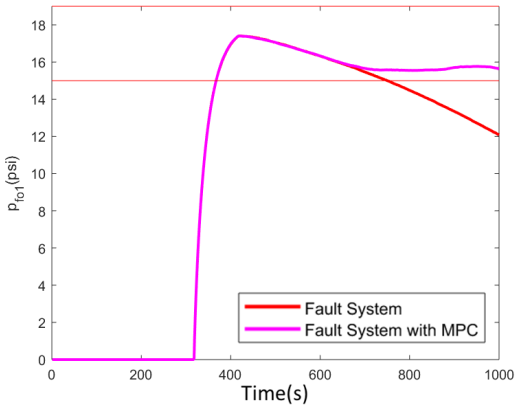
the fluctuation of the control signal in this case study has been significantly reduced, shown in Figure 5.15b, which is beneficial to the motor of the pump and improves the system reliability. Note that the increase of R does not affect the system performance as the other signals are similar to those in Case study 1, but without oscillations.

This case study shows that by including a proper R in the cost function, the system runs smoothly. Apart from water production and system smooth operation, another objective of ACM is to extend the remaining useful life of the filter. The next case study will integrate the filter life into the cost function.

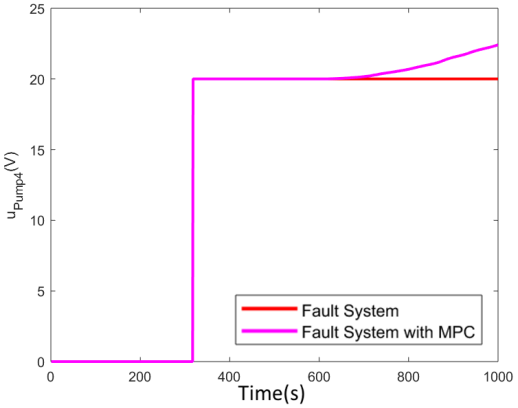
5.2.2.3 MPC Case Study III: Better Filter Life

In this case study, parameters in the cost function are set as $Q = 0.1$, $R = 10$, $T = 100$. With the increase of T, the objective function also considers keeping the balancing between reducing the fluctuation of the input signal, keeping enough water production and better filter life. Same as the operation and fault diagnosis in the previous two case studies, clogging is detected at the 331st second.

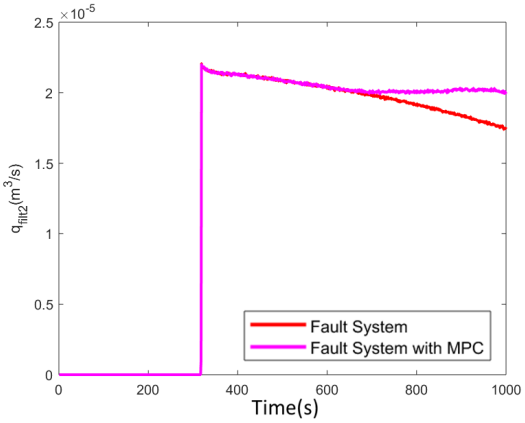
Figure 5.16 shows the simulation results of scenario with constant weight MPC (given by magenta) and its comparison against the scenario without MPC (given by red). In this case study, since the objective function focuses on enhancing filter life, the water production represented by the pressure of FO Module 1 is not considered as a priority. Note that the filter life is indicated by porosity ϵ. Therefore, the voltage of the Pump 4 decreases (shown in Figure 5.16b), the water pass through the Filter 2 decrease (shown in Figure 5.16c), and the water transferred to the FO Module 1 decreases. As a result, the pressure of FO Module 1 is kept around the lower safety constraints, shown in Figure 5.16a. Eventually, the porosity of the filter decrease slowly since the water pass through the filter decreases, shown in 5.16e.



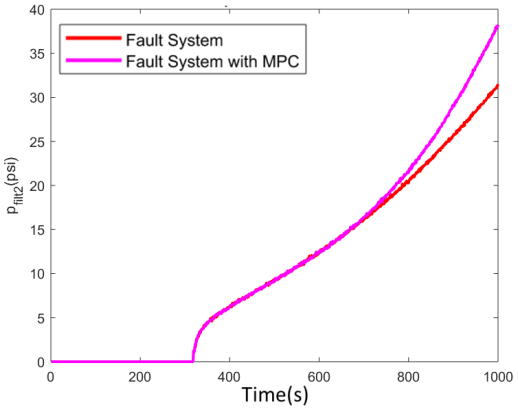
(a) Pressure of FO Module 1



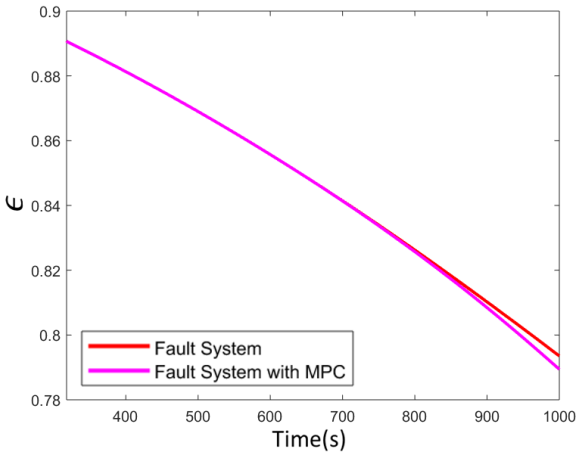
(b) Voltage of Pump 4



(c) The outflow rate of Filter 2



(d) The pressure of Filter 2

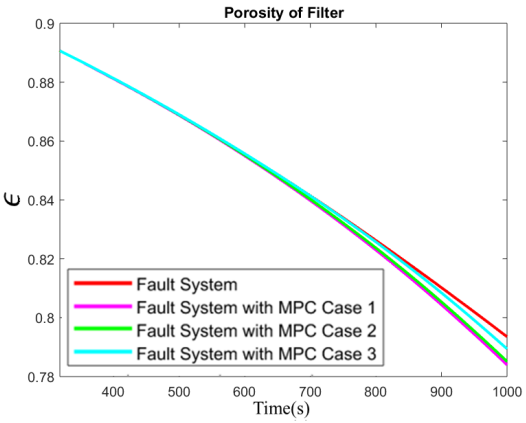


(e) Estimated porosity ϵ of Filter 2

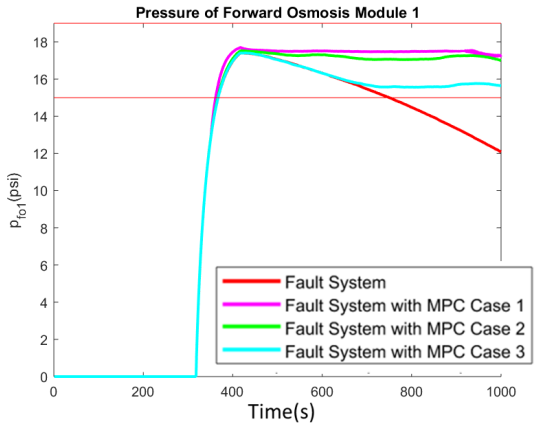
Figure 5.16 MPC case study III simulation result

5.2.3 Comparison of Different MPC Case Studies

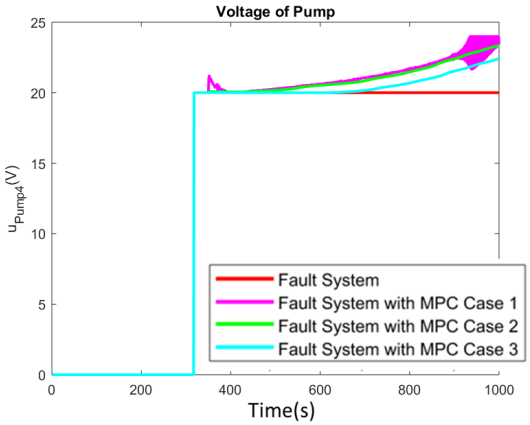
The simulation result under fault condition without and with different MPC case studies are compared in Figure 5.17.



(a) Estimated porosity ϵ of Filter 2



(b) Pressure of FO Module 1



(c) Voltage of Pump 4

Figure 5.17 Comparison of different MPC case studies

In Figure 5.17a, when the fault happens, the particles begin to accumulate in the filter and the porosity ϵ decreases. With the increased control input from MPC, the flow rate increases. Therefore, the particle's accumulation speed increases, which makes the clogging more severe.

In Figure 5.17b, when the fault happens, less water transferred into the FO Module 1 and the pressure of FO Module 1 decreases. Eventually, the pressure level becomes lower than the threshold. With MPC, case studies I and II keep the pressure of FO Module 1 at the reference value of 17.8 psi to maintain enough water production, while case study III keeps the filter life as long as possible, which maintains the pressure of

FO Module 1 at the lower state constraint of 15 psi to generate just sufficient water production.

Figure 5.17c shows that the fluctuation of the input signal is significantly reduced when the cost function considers the fluctuation of the input signal.

Quantitative Evaluation Between Different MPC Case Studies

Table 5.5 shows quantitative evaluation of these case studies. In case study I, when the priority for the cost function is to make sure that the following modules have enough water, the average flow rate is maintained at the nominal condition. In case study III, when the priority for the cost function is to keep the filter life as long as possible, the filter life, indicated by porosity ϵ, is improved compared with that of case study I, but with less water production.

Table 5.5 Quantitative evaluation of different MPC case studies

	Without MPC	Case study I	Case study II	Case study III
ϵ	0.7935	0.7840	0.7851	0.7894
ϵ decrease percentage	10.91%	11.98%	11.86%	11.37%
Average flow rate (m^3/s)	1.9869×10^{-5}	2.1274×10^{-5}	2.1102×10^{-5}	2.0473×10^{-5}
Water produced (m^3)	0.0136	0.0145	0.0144	0.0140
Pump voltage(V)	20	21.1811	21.0400	20.5168
Energy cost(J)	2.7320×10^4	2.8933×10^4	2.8741×10^4	2.8026×10^4

At the end of the mission at 1000 second, the porosity of filter is 0.7840 in MPC case study I (decreased by 11.98% compared with the value under normal condition), which is smaller than that in case study III at 0.7894 (decreased by 11.377% compared with the value under normal). The reason is that, in case study III, the flow rate decrease and fewer particles accumulate in the filter. Therefore, filter porosity ϵ degradation rate decreases. Similar analysis for the cake thickness L and permeability K. That is the filter life is extended with this case study III.

However, the water quantity from the case study III is reduced. The case study I generate water average flow rate at $2.1274 \times 10^{-5} m^3/s$ (0.0145 m^3 water produced during the mission) in this mission, while the case study III only generates water

average flow rate at $2.0473 \times 10^{-5} m^3/s$ (0.0140 m^3 water produced during the mission). Moreover, the energy consumption is also different, the case study I used average voltage at 21.1811 V (2.8933×10^4J energy usage during the mission) while the case study III used average voltage at 20.5168 V (2.8026×10^4J energy usage during the mission).

Therefore, tuning weights on objective function enable MPC to operate under different priorities. However, tuning these weights requires human involvement. Moreover, in a deep space habitat, the crew activities are limited, and tuning these weights requires domain knowledge and expertise. Therefore, an automated time-varying weight based on the fault severity and fault detection time is developed next to accommodate the fault conditions.

5.3 MPC with Variant Weights based on the Fault Severity

When the fault severity and fault detected time are different for different case studies, the optimization for the MPC controller should be different. For example, when the fault happens at the beginning of the mission, the priority for the optimization should be extending the filter life, so that the mission can be accomplished in degraded performance. However, when the fault happens at the end of the mission, the priority for the optimization should be enhancing the system performance, which means generating more water. Furthermore, the optimization should also be related to fault severity. Fault severity can be represented by the measurements (pressure, flow rate) or the parameters (porosity, cake thickness, and permeability). As discussed earlier, pressure or flow rate alone cannot be used to represent fault severity. The reason is that, when the WRS is working under constant flow rate condition, although the porosity ϵ decreases as more particles are trapped, the controller will maintain

the constant flow rate and this constant flow rate cannot indicate the decrease of the porosity ϵ. Similar analysis applied to the filter pressure p_{Filt2}.

For filter parameters, LS-UKF was used to estimate the porosity. The estimated porosity is used to represent fault severity. When the fault is not severe and the porosity of the filter is large, the weight of filter life in the objective function can be small such that the objective function focuses more on enhancing the system performance. When the fault become severe and the porosity of the filter is small, the weight of filter life in the objective function should be significant such that the objective function focuses more on extending the filter life by reducing the filter performance.

5.3.1 Variant Weight MPC

As discussed in the previous section, the objective function of MPC has three factors Q, T, and R, which represent the water production, fluctuation of the input signal, and the filter life. When the weight coefficient for a particular factor is more significant, the factor in the objective function is more substantial. The objective function focuses more on this particular factor.

To connect the weight coefficient T with the fault detection time and fault severity, the modified Sigmoid function is introduced, shown in Eq. (5.19). Note that two factors of fault detection time and estimated porosity are chosen in the function of T. The reason is that: 1) the estimated porosity represents the capability of water passing through the filter and is directly used as the health condition of the filter; 2) the fault detection time is essential in mission scenarios. In other words, the remaining useful life of the filter has different levels of critically for fault at different periods of the mission. In the cost function, T is the weight for filter life. A higher value of T indicates the optimization function focuses more on enhancing filter life. Therefore, the optimization in MPC does not consider the filter life factor when T is 0.

In this simulation, $T = 50$ is set as a benchmark for the optimization. Different fault detection time and estimated porosity will lead to different weight T and, therefore, different ACM strategies. With this consideration, the weight T is given as follows:

$$T = \frac{(100/\epsilon)}{(1 + e^{(20(t_x - 0.5))})} \tag{5.19}$$

where $t_x = \frac{T_d}{T_f}$ in which T_f is the mission time and T_d is the fault detection time. Figure 5.18 shows how T changes with t_x. The value of t_x is determined when the fault is detected. The estimation of porosity ϵ changes during the simulation because when the water passes through the filter, more and more particles are trapped in the filter. Therefore, T is time-varying in the simulation. Since the porosity decreases in the operation, the value of T increase with the operation time.

To study the effect of variant weight MPC, three simulation studies are provided in the next section. In these three studies, the fault is detected at different time, as indicated by vertical lines in Figure 5.18. The setting of three simulation studies are as follows:

1. Fault happens at the beginning of the mission: In this study, the fault is injected at the 317th second and detected at the early stage of the mission at the 331st second. If the estimated porosity at this time is 1, according to the T function, the initial value of T is 95, and it increases in the simulation. With this setting, the optimization focuses more on enhancing filter life.

2. Fault happens in the middle of the mission: In this study, the fault is injected at the 500th second and detected at the middle stage of the mission at the 520th second. If the estimated porosity at this time is 1, according to the T function, the T function generate an initial weight of 48 so that the optimization consider the filter life, but it is no longer dominant over other two factors of water production and extending filter life.

3. Fault happens in the middle of the mission: In this study, the fault is injected at the 700th second and detected at the middle stage of the mission at the 721st second. If the estimated porosity at this time is 1, according to the T function, the T function generate an initial weight of 3 so that the optimization considers the filter life, but it is no longer a major factor of optimization.

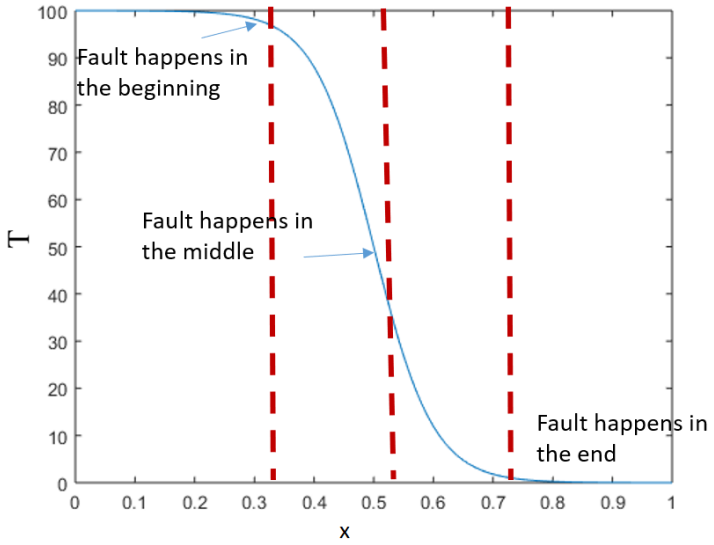


Figure 5.18 T variation versus $t_x = T_d/T_f$

5.3.2 Variant Weight MPC Simulation Results

This section shows three MPC simulation case studies of variant weight MPC based fault-tolerant control. The objective of these case studies is to mitigate the effect of the filter clogging based on the fault detection time and fault severity. The operating profile of the WRS follows the one described in WRS modeling and simulation. In the simulation, the parameters are set as follows: the particle's suspension rate is 0.14% and the normal flow rate is set as 2.2×10^{-5} m^3/s. The nominal voltage of the Pump 4 is 20 V, and the maximum voltage is 24 V. The Pump 4 starts to transfer water from the 317th second.

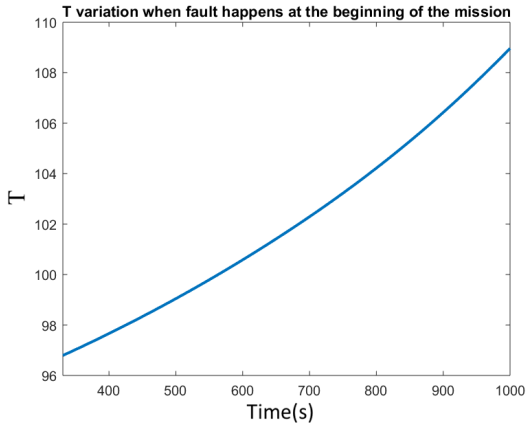
5.3.2.1 Fault Happens at the Beginning of the Mission

In a deep space habitat, the replacements of components are limited since the transport of those components requires a significant number of resources. Therefore, when the fault happens at the early stage of the mission, the objective of the ACM is to make sufficient use of each component. The component's remaining useful life is very significant, and the priority now is to keep mission success and let the system operate in degraded performance.

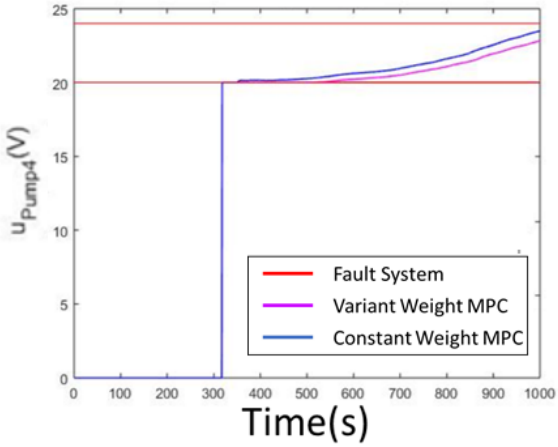
Figure 5.19 shows the simulation results of scenario with variant weight MPC (given by blue) and its comparison against the scenario without MPC (given by red) and scenario with constant weight MPC (given by magenta). In this case study, the T value for variant weight MPC increasing from 97 when the fault happens at the beginning of the mission to 109 at the end of the mission. The increasing of T is caused by the decrease of porosity when fault becomes severer. Figure 5.19a shows the change of T in the mission. The results show that variant weight MPC generates lower voltage of Pump 4 (Figure 5.19b). As a result, the pressure of the Filter 2 decreases (Figure 5.19c). This leads to a lower average flow rate cross the filter (Figure 5.19d), which represents the water production is lower (Figure 5.19e). When less water pass through the Filter 2, fewer particles are trapped in Filter 2. Therefore, the degradation of Filter 2, indicated by the decrease of porosity ϵ, is slower (Figure 5.19f). This case study proves that, when fault happens at the beginning of the mission, the value of T is higher, which represents the system focuses more on enhancing filter life instead of generating water.

Table 5.6 summarizes the performance of three cases in terms of filter life (given by porosity ϵ at the end of the mission), water production (given by average flow rate), and energy consumption (given by pump voltage).

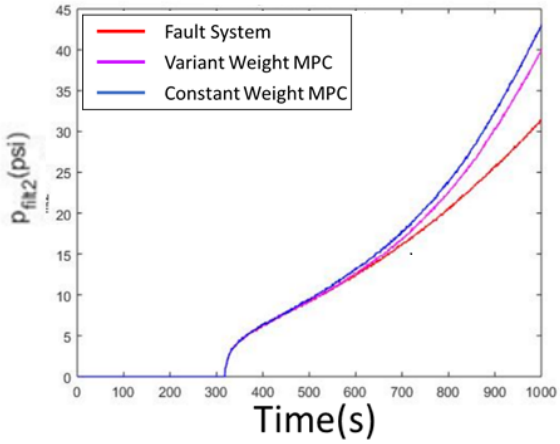
At the end of the mission at 1000 second, the porosity of filter for MPC with constant weight is 0.7843 (decreased by 11.95% compared with the value under normal



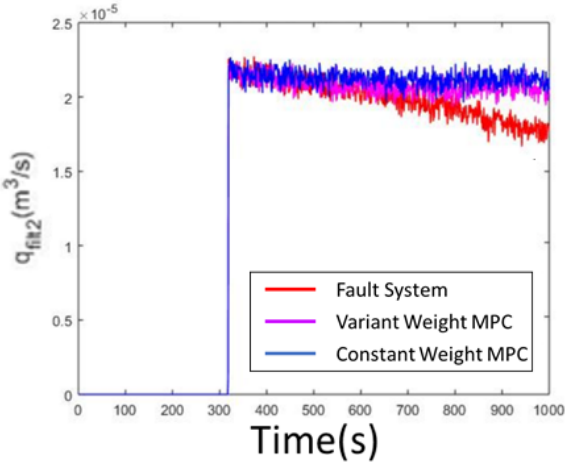
(a) T variation



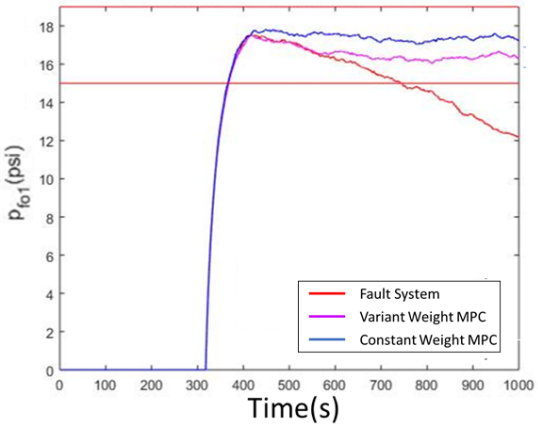
(b) Voltage of Pump 4



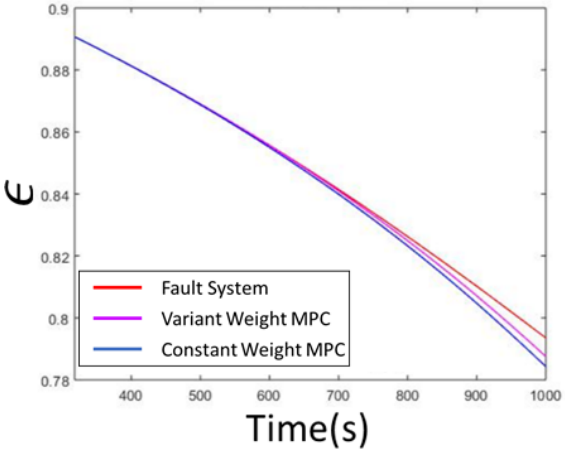
(c) The pressure of Filter 2



(d) The outflow rate of Filter 2



(e) Pressure of FO 1



(f) Porosity of Filter 2

Figure 5.19 MPC simulation when fault happens at the beginning

condition), which is smaller than that MPC with variant weight at 0.7869 (decreased by 11.65% compared with the value under normal).

However, the water quantity from the MPC with variant weight is reduced. The MPC with constant weight generates water average flow rate at 2.1212×10^{-5}m^3/s (0.0145 m^3 water produced during the mission) in this mission, while the MPC with variant weight only generates water average flow rate at 2.0844×10^{-5}m^3/s (0.0142 m^3 water produced during the mission). Moreover, the energy consumption is also different, the MPC with constant weight used average voltage at 21.1380 V (2.8875×10^4J energy usage during the mission), while the MPC with variant weight used average voltage at 20.7469 V (2.8340×10^4J energy usage during the mission).

The quantitative results in this table show that MPC with variant weight has better filter life and less energy consumption than MPC with constant weight. However, it also generates less water than MPC with constant weight. The MPC with variant weight achieves the ACM objective better than other cases.

Table 5.6 Comparison of results from case studies when fault happens at the beginning of the mission

	Without MPC	Constant weight MPC	Variant weight MPC
ϵ	0.7935	0.7843	0.7869
ϵ decrease percentage	10.91%	11.95%	11.65%
Average flow rate (m^3/s)	1.9869×10^{-5}	2.1212×10^{-5}	2.0844×10^{-5}
Water produced (m^3)	0.0136	0.0145	0.0142
Pump voltage(V)	20	21.1380	20.7469
Energy cost(J)	2.7320×10^4	2.8875×10^4	2.8340×10^4

5.3.2.2 Fault Happens in the Middle of the Mission

In this study, the fault is injected at the middle stage of the mission. The fault is injected at the 500th second and detected at the 528th second. The T value for variant weight MPC increases from 41 when the fault is injected to 44.5 at the end of the mission.

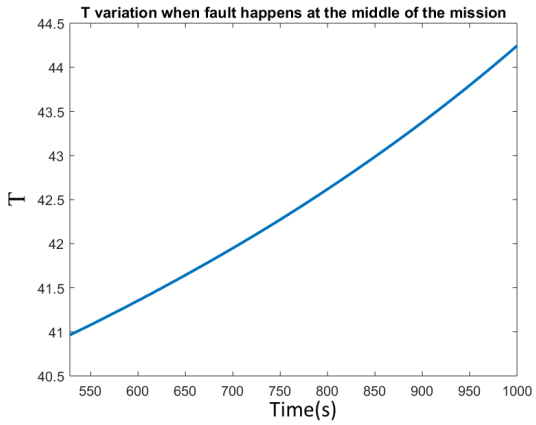
Figure 5.20a shows the change of T. Same as the previous case, the increase of T is caused by the decrease of porosity when fault becomes severer. Figure 5.20 shows the simulation results of scenario with variant weight MPC (given by blue) and its comparison against the scenario without MPC (given by red) and scenario with constant weight MPC (given by magenta). The increase of T is caused by the decrease of porosity when the fault becomes severer. Since T changes from 41 to 44.5 in variant weight MPC, which is very close to the benchmark value of $T = 50$ in the constant weight MPC, their simulation results are very similar.

Compared with constant weight MPC, variant weight MPC generates a similar increase of voltage of Pump 4 (Figure 5.20b). Therefore, the pressure of the Filter 2 increases (Figure 5.20c). This leads to a higher average flow rate cross the filter (Figure 5.20d). This indicates that the water production is maintained (Figure 5.20e). When more water pass through the Filter 2, more particles are trapped in Filter 2 and the degradation of Filter 2 becomes faster (Figure 5.20f).

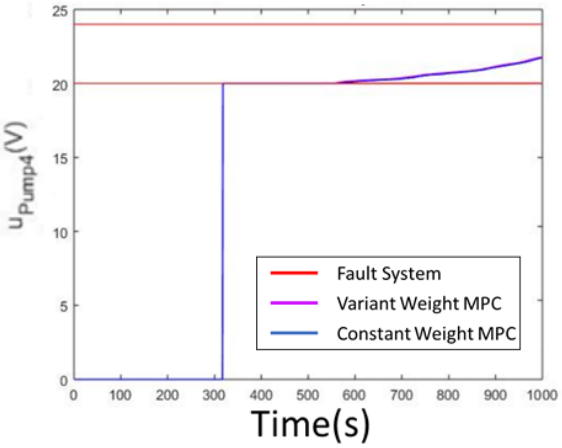
Table 5.7 summarizes the performance of three cases in terms of filter life (given by porosity ϵ at the end of the mission), water production (given by average flow rate), and energy consumption (given by pump voltage). The quantitative results in this table show that MPC with variant weight has comparable performance with MPC with constant weight.

Table 5.7 Comparison of case studies when fault happens at the middle of the mission

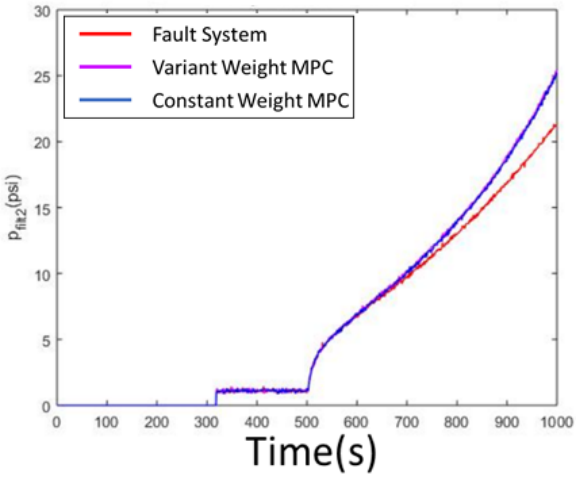
	Without MPC	Constant weight MPC	Variant weight MPC
ϵ	0.8234	0.8201	0.8196
ϵ decrease percentage	7.56%	7.93%	8.09%
Average flow rate (m^3/s)	2.0823×10^{-5}	2.1432×10^{-5}	2.1487×10^{-5}
Water produced (m^3)	0.0104	0.0107	0.0107
Pump voltage(V)	20	20.4372	20.4972
Energy cost(J)	2.0×10^4	2.0437×10^4	2.0497×10^4



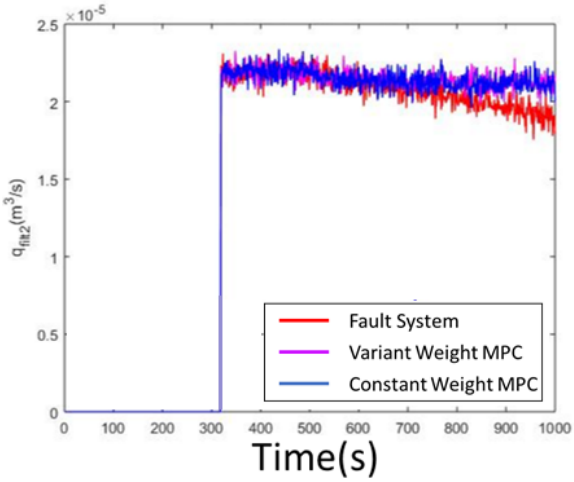
(a) T variation



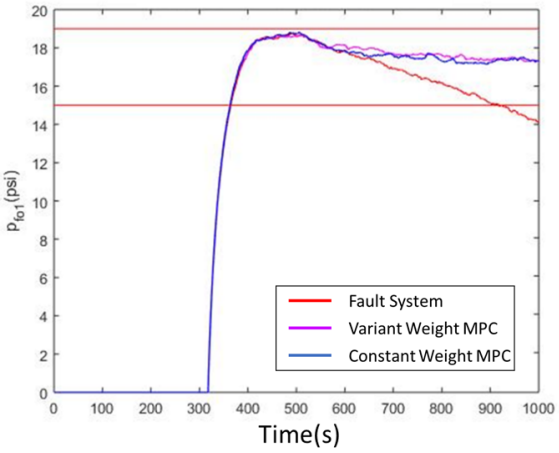
(b) Voltage of Pump 4



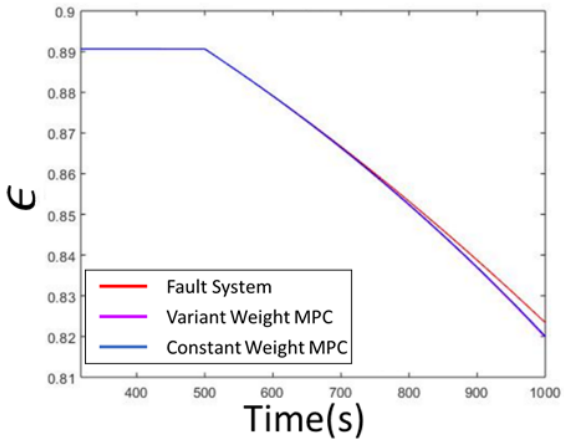
(c) The pressure of Filter 2



(d) The outflow rate of Filter 2



(e) Pressure of FO 1



(f) Porosity of Filter 2

Figure 5.20 MPC simulation when fault happens at the middle

5.3.2.3 Fault Happens Late in the Mission

In this study, the fault is injected at the later stages of the mission. The fault is injected at the 700th second and detected at the 730th second. Since the mission is approaching the end, MPC aims to enhance the system performance, i.e., maintain the water production in WRS. The reason is that, at the end of the mission, as long as the faulty component can hold until the mission end, then the remaining useful life for this particular component is no longer significant. Figure 5.21 shows the simulation results of scenario with variant weight MPC based ACM (given by blue) and its comparison against the scenario without MPC (given by red) and scenario with constant weight MPC (given by magenta).

In this case study, the T value for variant weight MPC increases from 1.12 when the fault happens to 1.17 at the end of the mission, which is shown in Figure 5.21a. The increase of T is caused by the decrease of porosity when the fault becomes severer. Compared with constant weight MPC, variant weight MPC generates higher voltage of Pump 4 (Figure 5.21b). Therefore, the pressure applied to the Filter 2 increases (Figure 5.21c). This leads to a higher average flow rate cross the filter (Figure 5.21d). This indicates that the water production is maintained (Figure 5.21e). When more water pass through the Filter 2, more particles are trapped in Filter 2 and the degradation of Filter 2 is faster (Figure 5.21f). This case study proves that, at the end of the mission, a low value of T is reasonable and makes the system focuses more on improving water production instead of enhancing filter life.

Table 5.8 summarizes the performance of three cases in terms of filter life (given by porosity ϵ at the end of the mission), water production (given by average flow rate), and energy consumption (given by pump voltage). The quantitative results in this table show that MPC with variant weight generates the most water production. Meanwhile, this ACM strategy consumes the most energy consumption and filter life.

At the end of the mission at 1000 second, the porosity of filter for MPC with constant weight is 0.8524 (decreased by 4.3% compared with the value under normal condition), which is larger than that MPC with variant weight at 0.8516 (decreased by 4.4% compared with the value under normal).

Table 5.8 Comparison of case studies when fault happens at the end of the mission

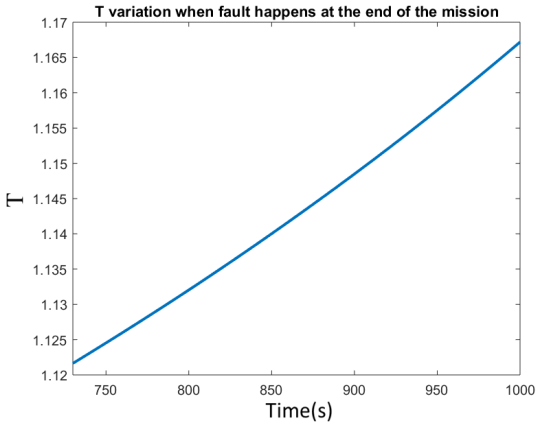
	Without MPC	Constant weight MPC	Variant weight MPC
ϵ	0.8532	0.8524	0.8516
ϵ decrease percentage	4.21%	4.3%	4.4%
Average flow rate (m^3/s)	2.1493×10^{-5}	2.1657×10^{-5}	2.1830×10^{-5}
Water produced (m^3)	0.0064	0.0065	0.0065
Pump voltage(V)	20	20.1170	20.2307
Energy cost(J)	1.2×10^4	1.2070×10^4	1.2138×10^4

When a fault is detected in the system, the fault mitigation uses this fault information to reduce the failure impact and prevent the mission from failure. The constraints consider the physical limitation of the components and the crew requirement. The cost function in MPC considers the water production, fluctuation of the input signal and degradation of filter. The variant weight MPC integrates the fault information at different fault detection times and fault severity in the objective function to achieve more flexible and robust ACM strategies.

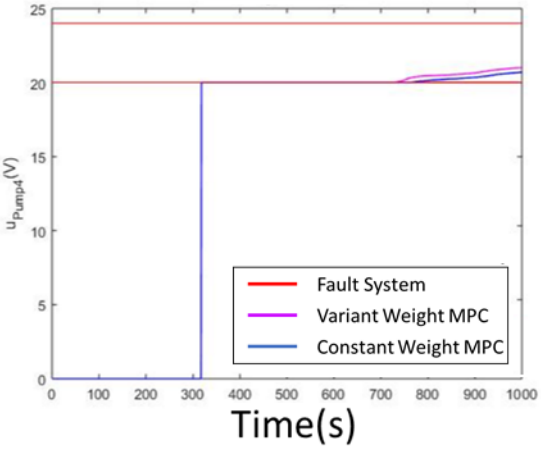
5.4 Conclusion

In the previous chapter, the diagnosis function detects the fault and identifies the fault severity. After the fault is detected, the fault mitigation uses this fault information to reduce the failure impact and prevent the mission from failure. The cost function in MPC considers the water production, fluctuation of the input signal and degradation of filter. The constraints consider the safety constraint for the component. The main contribution of this work is in traditional MPC, the weight in MPC objective function

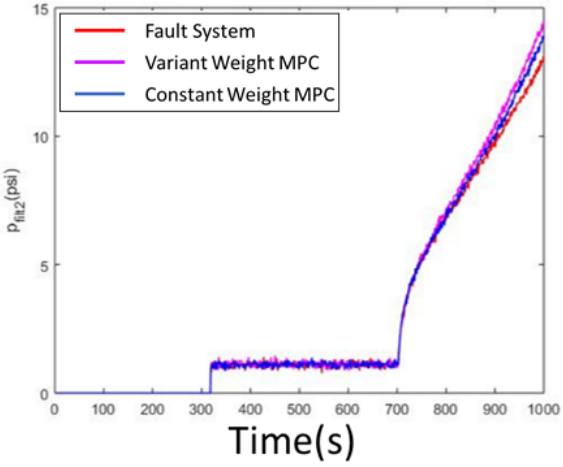
is fixed as constant. However, in the variant weight MPC use the information of fault severity to adjust the weight in the MPC.



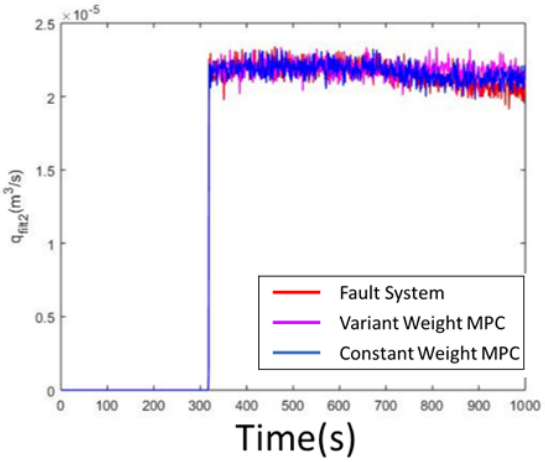
(a) T variation



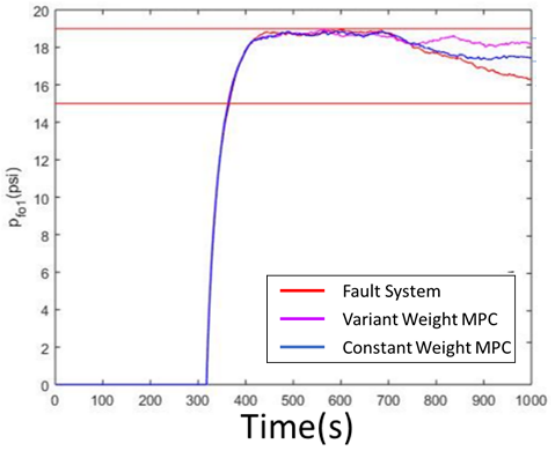
(b) Voltage of Pump 4



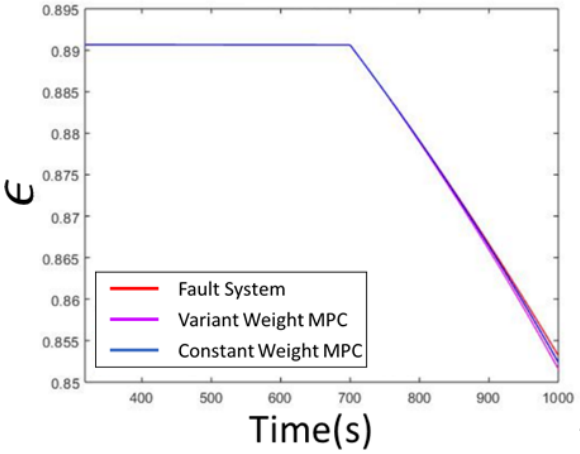
(c) The pressure of Filter 2



(d) The outflow rate of Filter 2



(e) Pressure of FO 1



(f) Porosity of Filter 2

Figure 5.21 MPC simulation when fault happens at the end

Chapter 6

Sensor and Component Faults Diagnosis

Sensors are important components of all modern machines and vehicles, from the ground, to oceans, air and space, both for manned and unmanned missions. Sensors provide critical information for the control, operation and monitoring of such systems. Health monitoring systems, particularly for diagnostics and prognostics of faults and failures, rely on sensor information to make PHM decisions. Most of the algorithms presented in the previous chapters have implicitly assumed the sensors are healthy, i.e., giving accurate measurements without significant bias or drifting. In this chapter, sensor health is considered as a fault mode of the system. A systematic method is developed to distinguish sensor faults from component faults, and to ensure a faulty sensor would not mislead the PHM system to a wrong diagnosis. The method is demonstrated with the WRS application on a single sensor. However, the method can be extended to cover the case of multiple faulty sensors with some future development.

6.1 Sensor Faults

6.1.1 Effects of Sensor Faults on System Performance

Sensors are essential components for monitoring the health condition of a complex system like deep space habitat. However, sensors may generate inaccurate or poor quality data due to long-term operation in adverse conditions.

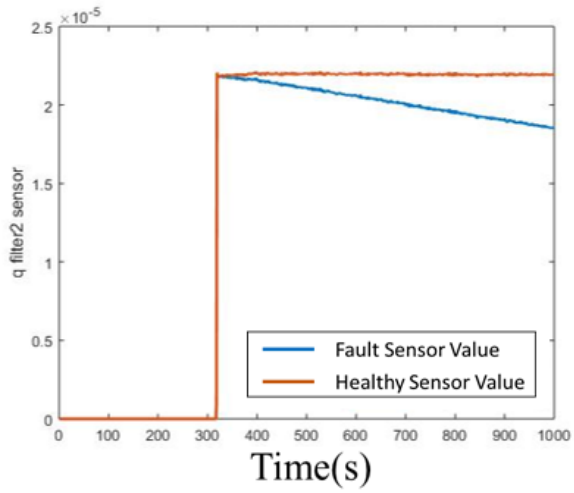
To demonstrate the influence of sensor fault, three different fault modes, namely, sensor drifting, precision degradation(i.e., noisy sensor), and sensor bias, are studied

in the WRS system. The target sensor in this example is the flow sensor in the pump-filtering subsystem discussed in the previous chapter. To make a fair comparison with the results from previous chapters, simulations are conducted based on the same operating profile discussed early, i.e., a simulated 1000-second mission in which the system is put in use from the 317th second. The simulation results (without incorporating the ACM strategy described in the previous chapter) with Filter 2 flow rate sensor drifting fault, sensor precision degradation and sensor bias fault are shown in Figures 6.1a, 6.1b, and 6.1c, respectively. The Pump 4 starts to transfer water at the 317th second. If the water is free of particles, the Filter 2 keeps in healthy condition when the water passes through the Filter 2. The true flow rate pass through Filter 2 is a constant (2.2×10^{-5} m^3/s), as shown by the orange line in the simulation. With sensor drifting fault, the sensor reading decreased over time, as displayed in the blue line in Figure 6.1a. With sensor precision degradation, the sensor measurement becomes rather noisy, as displayed in the blue line in Figure 6.1b. With sensor bias fault, the sensor reading increases or decreases by a certain constant, as displayed in the blue line in Figure 6.1c.

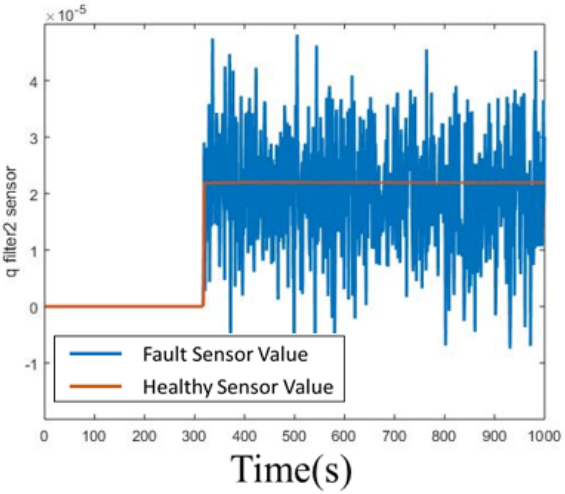
Apparently these sensor faults can lead to false alarms, missed detections or incorrect diagnosis. If misdiagnosed, the fault accommodation logic, such as the MPC optimization algorithm presented in the previous chapter can potentially act on a wrong decision and bring the system to undesired states. Therefore, it is important to be able to detect sensor faults, and further to distinguish sensor faults from component faults, such that they can be addressed accordingly.

6.1.2 Sensor Drifting Fault

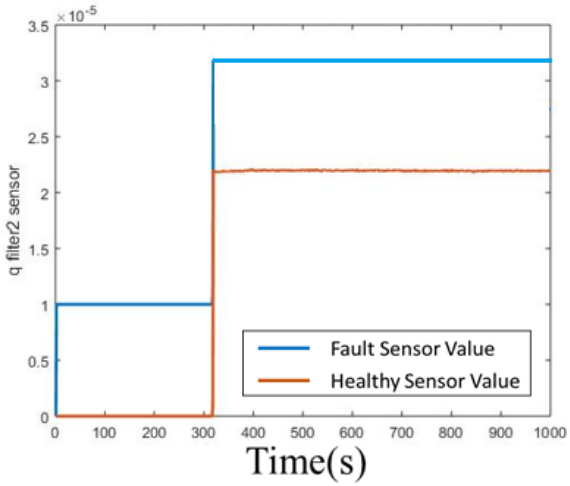
The sensor drifting fault scenario in this simulation study is configured based on the assumption that sensor reading decreases over time while the actual value remains a



(a) Sensor drifting



(b) Sensor precision degradation



(c) Sensor bias

Figure 6.1 Sensor fault types

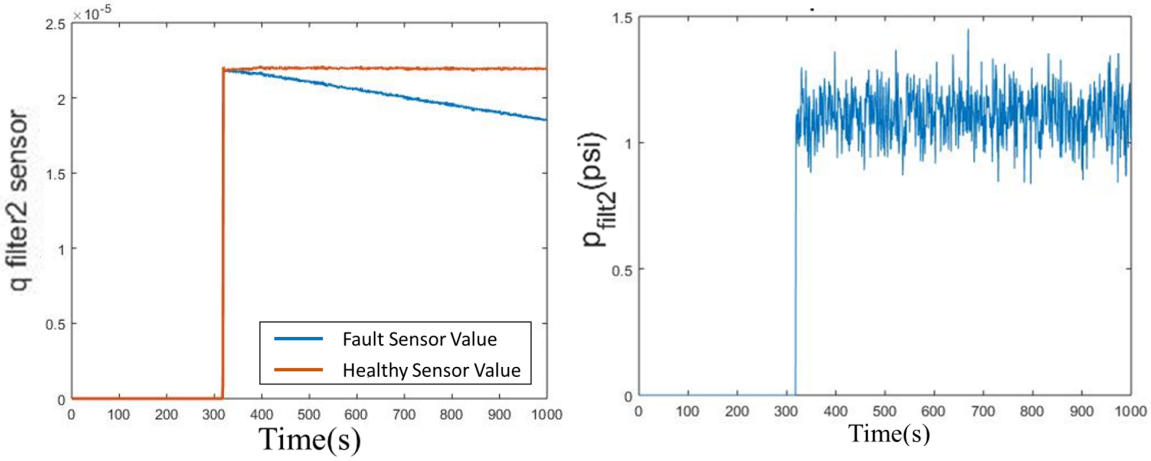
constant and no component fault is present. The sensor drifting fault of Filter 2 in the simulation is defined in Eq. (6.1).

$$s_{\text{Filt2}} = \begin{cases} q_{\text{Filt2}}, & t \leq \text{t}_\text{f} \\ q_{\text{Filt2}} - 5 \times 10^{-9} \times (t - \text{t}_\text{f}), & \text{otherwise} \end{cases} \tag{6.1}$$

where t_f is the time when the sensor fault occurs. In this example, $t_\text{f} = 317$ second and the sampling time is 1 second. In the simulation, drifting rate and fault injection time can be adjusted by the parameters in this equation.

Sensor Drifting Fault Simulation Results

The operating profile of the WRS follows the one described in the WRS modeling and simulation chapter. In the simulation, the parameters are set as follows: the particles' suspension rate is 0.14% and the normal flow rate is set as 2.2×10^{-5} m^3/s. The nominal voltage of the Pump 4 is 20 V, and the maximum voltage is 24 V, p_{FO1} is set as 17.8 psi and initial porosity ϵ is set as 0.89 (initial condition without filter degradation).

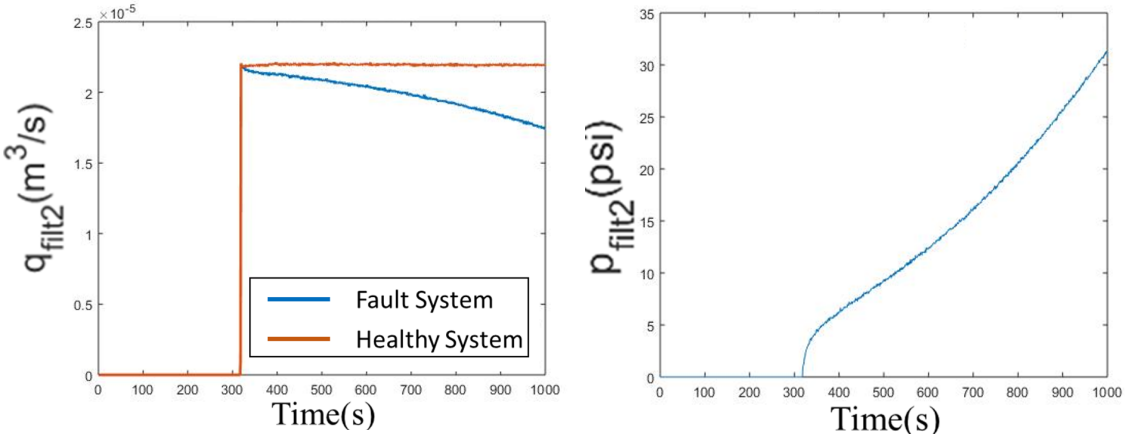


(a) Sensor drifting fault flow rate sensor reading

(b) Sensor drifting fault pressure sensor reading

Figure 6.2 Sensor fault simulation results

Here is a comparison between sensor fault (as shown in Figure 6.2) and component fault(as shown in Figure 6.3). When the flow rate sensor-drifting fault happens while the component (Filter 2) remains healthy, the pressure of Filter 2 keeps at around 1.15 psi (as shown in Figure 6.2b) because the component (Filter 2) is not clogged. Note that while the faulty flow rate sensor reading is drifting low, the actual flow rate remains constant (as shown in Figure 6.2a). However, when the component (Filter 2) fault happens, the flow rate sensor reading will decrease as the actual flow rate decreases (as shown in Figure 6.3a), and at the same time, the pressure sensor reading will increase (as shown in Figure 6.3b) because of the filter clogging. This indicates



(a) Filter clogging component fault flow rate sensor reading

(b) Filter clogging component fault pressure sensor reading

Figure 6.3 Component fault simulation result

that the sensor fault can be detected by leveraging the Filter 2 pressure measurement, since the flow rate sensor fault does not affect the downstream pressure measurement at Filter 2.

Sensor	Flow sensor drift	Clogged filter	Healthy System
Flow sensor	↓	↓	—
Pressure sensor	—	↑	—

Figure 6.4 Comparison of sensor fault and component fault

By analyzing the physics of fault propagation in the system, shown as Figure 6.4, the consequence of sensor faults, as compared with component faults, can be modeled and used for correct fault isolation. In general, a component fault affects multiple sensor measurements while a sensor fault only affect itself unless the sensor is included in the control loop.

6.2 Sensor Fault Diagnosis Case Study I

6.2.1 Sensor Fault Diagnosis

Figure 6.5 shows a sensor fault accommodation scheme leveraging analytical measurement redundancy. Suppose n measurements are used to estimate a parameter. With analytical sensor redundancy in the system, it is often feasible that the parameter can be estimated using multiple, different subsets of the measurement combinations. For example, one estimate could be based on (m_1, m_2), while another based on (m_2, m_3, m_4). If all sensors are healthy, the two estimates will be rather close. Otherwise, the sensor suite configuration used in the estimation provides insight into which sensor might have contributed to the discrepancy. The differences in the estimates can be further utilized to generate a health index for the sensor of interest. Note that the Bayesian Network in Figure 6.5 is just a representative estimator that can be replaced, in general, by other form of estimators.

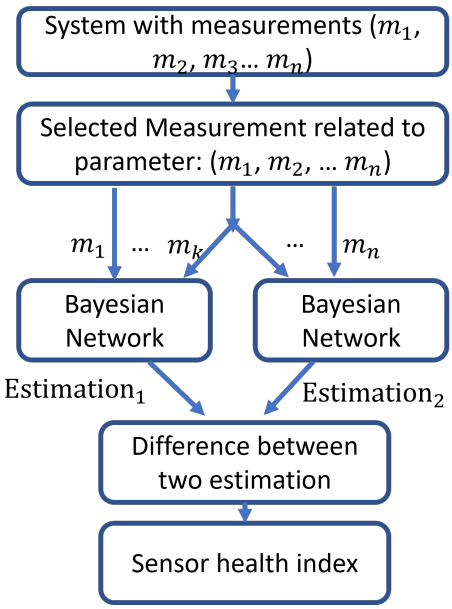


Figure 6.5 Sensor fault accommodation diagram

In case of the flow rate sensor drifting fault mentioned above, the flow rate reading from the sensor will gradually increase/decrease while the filter pressure response to fault is different. To diagnose this fault, a model is built to describe the relationship between flow rate and filter pressure for healthy nominal condition. This model will generate an estimated pressure values. A residual pressure Δp will be calculated using the estimated pressure and measured pressure. LS-UKF is then applied to estimate the sensor health index based on the residual. When the fault is correctly classified, the appropriate fault mitigations can be carried out. This fault accommodation scheme is described in Figure 6.6 below.

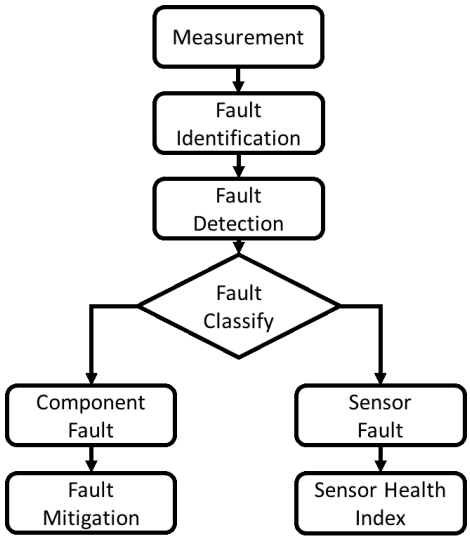


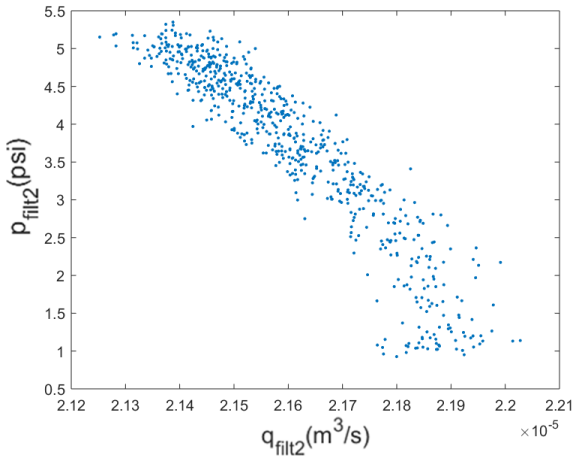
Figure 6.6 Sensor fault classification diagram

Feature Extraction (Predicted Pressure From Flow Rate Sensor)

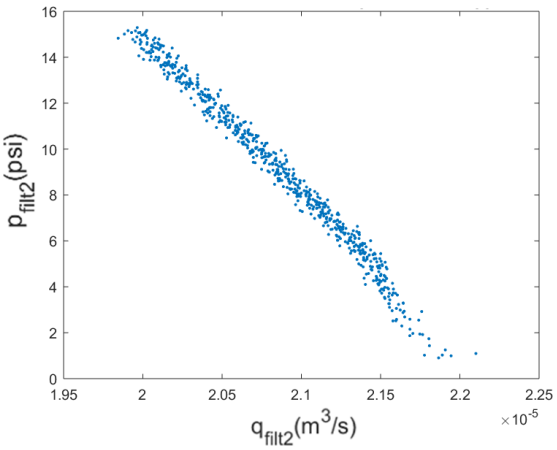
As mentioned earlier, we use the pressure residual to detect the sensor fault and identify the health index of the sensor.

We assume the sensor-drifting fault and component fault do not happen at the same time. In order to estimate filter pressure, we need to build a simple model.

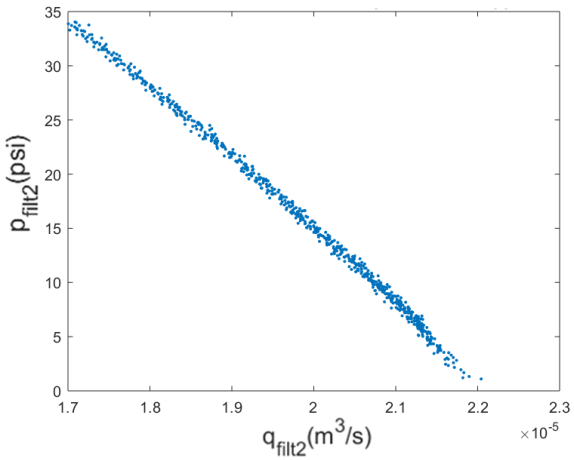
Figure 6.7a, Figure 6.7b, Figure 6.7c shows the relationship between pressure and flow rate when the water particle percentage are 0.01%, 0.07%, 0.14%, respectively,



(a) Particle suspension rate at 0.01%



(b) Particle suspension rate at 0.07%



(c) Particle suspension rate at 0.14%

Figure 6.7 Pressure versus flow rate for different levels of particle suspension rates

in which the x-axis represents the flow rate through Filter 2, q_{Filt2}, and the y-axis represents the pressure of Filter 2, p_{Filt2}.

Since the relationship between Filter 2 pressure and flow rate is linear for all operating conditions, a linear model is used to represent the relationship between p_{Filt2} (output y) and q_{Filt2} (input x) when Pump 4 operates at a constant voltage. The linear model is shown in Eq. (6.2). The dataset in Figure 6.7a, Figure 6.7b, Figure 6.7c are used to validate the accuracy of this model. The Root Mean Square Error (RMSE) between the predicted pressure and the actual pressure are 0.4115, 0.6529, and 0.8913 respectively.

$$y = -6.5171 \times 10^6 \times x + 144.5 \tag{6.2}$$

6.2.2 LS-UKF based Sensor Fault Diagnosis

The fault detection is based on the LS-UKF method developed in the previous section. To take advantage of the Lebesgue sampling method, the state space (flow rate) is divided into several Lebesgue states, and the LS-UKF is implemented as follows:

At first, the LS-UKF checks if the feature (Filter 2 pressure residual) reaches a new Lebesgue state. Consequently, the sigma points of the UKF are initialized based on the dimension of the state, and sigma points are propagated through the process model to obtain the prediction results. Next, the prediction obtained in the forecast step is combined with the measurements and the state co-variance is updated if new Lebesgue state is reached, as well as the state estimation. The final output of the algorithm is the estimated sensor health index and co-variance or the previously estimated sensor health index if the feature did not reach the next Lebesgue state.

The LS-UKF diagnosis framework requires a process model and a measurement model. The process model and measurement model were developed as follows:

Process Model in UKF

The process model describes the degradation of the sensor health index. The sensor health index is related to the residual pressure of Δp, which is the difference between the predicted pressure from the linear model given in Eq. (6.2) and the sensor reading. Since the model given in Eq. (6.2) describes the pressure with the actual flow rate (no matter clogging happens or not), it predicts pressure based on the actual flow rate. The pressure reading, however, behaves differently under sensor fault or component fault. When the component fault happens, the pressure sensor reading will be close to the predicted pressure. On the other hand, when the flow rate sensor fault happens, the pressure sensor reading will deviate from the predicted

pressure. Therefore, the difference between the predicted pressure and pressure sensor reading can be used to distinguish sensor fault and component fault.

To quantify sensor fault, a sensor health index is introduced. The sensor health index has a range from 1 to 0. When the sensor is in good condition, the health index is 1, the sensor health index decreases when the Δp increases as the sensor starts to drift from its true value. When the value of sensor health index reaches 0, it indicates a complete sensor failure. With this consideration, the process model for sensor fault index can be given as follows:

$$x(k) = x(k-1) - C1 \times (\Delta p) \times LBG1 + w_k \quad w_k \sim N(0, Q_k) \tag{6.3}$$

where Δp is the residual of predicted pressure and sensor reading, $x(k)$ is the state of sensor health index of Filter 2, $C1$ is a constant, $LBG1$ is the Lebesgue length and w_k is the process noise.

Measurement Model in UKF

The measurement model shows the relationship between fault state x_k (sensor health index) and feature z_k (Δp). The model is given as follows:

$$z_k = 20 - 20 \times x_k + v_k \quad v_k \sim N(0, R_k) \tag{6.4}$$

where $z_k = \Delta p$ is the measurement (feature) and v_k is measurement noise.

Simulation Results

Figure 6.8 shows the sensor fault detection simulation result. The baseline distribution has a mean value of 1 (healthy condition) and a standard deviation of 0.1. The detection threshold is established by threshold the intersection of healthy state distribution and estimated faulty state distribution, as shown in Figure 6.8. The thresholds reflect the trade-off between false alert rate and miss-detection rate.

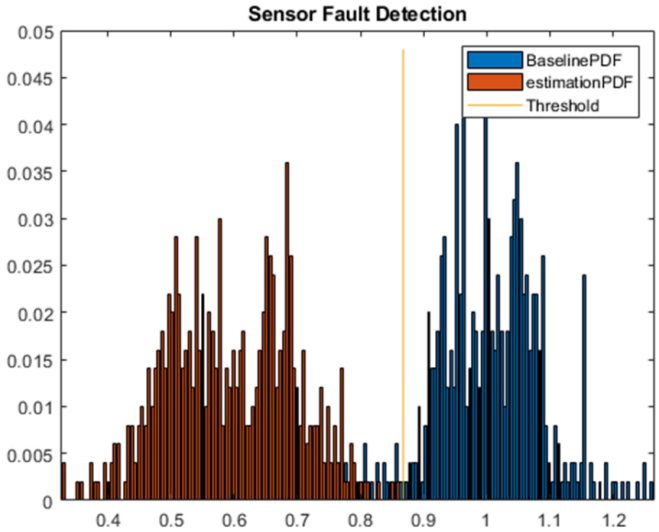


Figure 6.8 Sensor fault detection simulation result

In this case study, the sensor fault is injected at the 317th second, and the fault is detected at the 386th second, and then at the 409th second the fault is classified as a sensor fault.

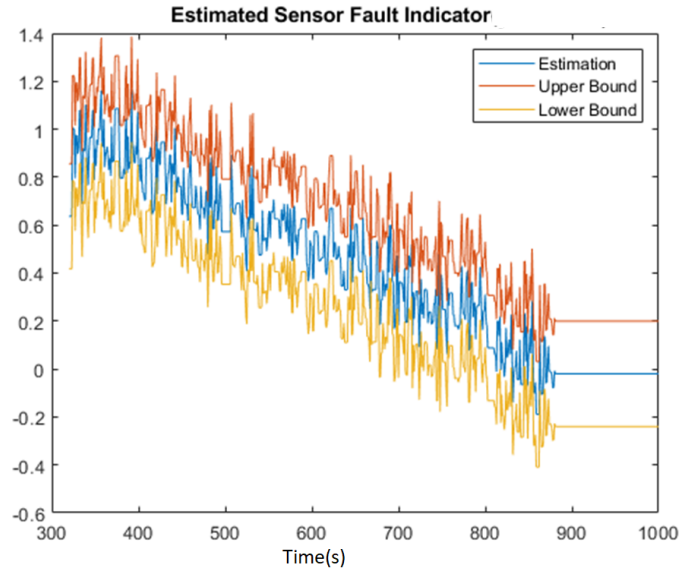


Figure 6.9 Estimated sensor fault index

Figure 6.9 shows the estimation of the sensor health index, in which mean value, lower and upper bounds of 95% confidence interval are shown by blue, yellow and red lines, respectively. At the beginning, the sensor drifting happens, the sensor health

index decreases from 1 to 0 as the residual pressure Δp increases. At the 852nd second, the residual pressure is larger than 20 psi, the sensor health index is 0, the sensor is in complete failure.

Limitation of This Approach

This case study only applies when Pump 4 operates at constant voltage. However, in real applications, WRS may work under different conditions, such as constant pressure, constant flow, water quality variation, etc. To make the sensor fault diagnosis more complete, a more advanced sensor fault diagnosis approach by involving another measurement is studied in case study II.

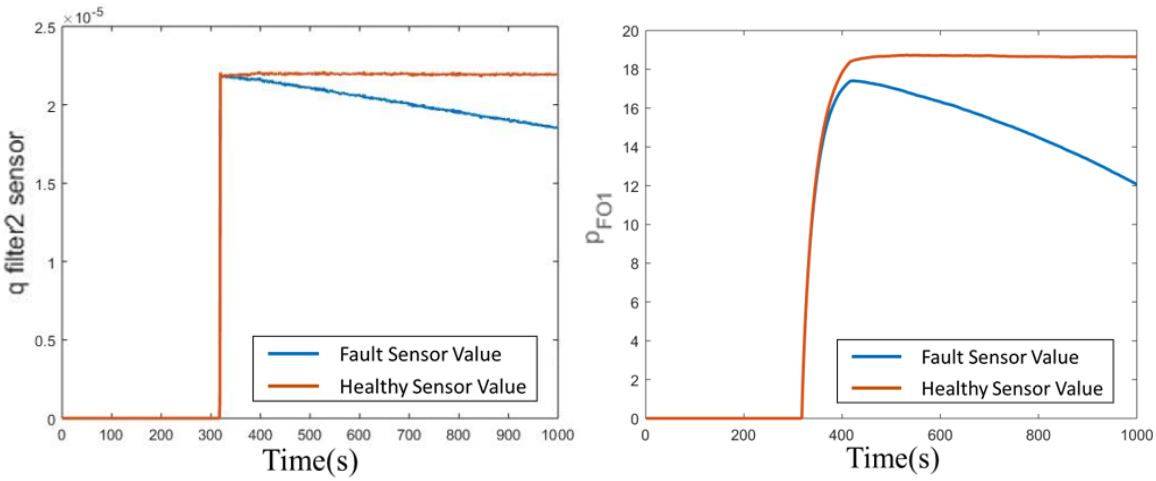
6.3 Sensor Fault Diagnosis Case Study II

In the previous case study, a sensor fault detection method was developed based on the relationship between p_{Filt2} and q_{Filt2} when Filter 2. It only works for certain working conditions. When the system operating condition changes as mentioned above, its detection accuracy may decrease. To address this problem, and make the sensor fault diagnosis more robust, another example of exploring analytical sensor redundancy is developed.

When p_{Filt2} and q_{Filt2} are used to estimate the porosity of Filter 2 ϵ, the estimation may be affected by a faulty flow rate sensor. To leverage this knowledge, another measurement p_{FO1} is employed to estimate the porosity of Filter 2 ϵ in parallel. The reason we choose p_{FO1} is that q_{Filt2} and p_{FO1} have similar responses when a component fault of filter clogging occurs, as shown in Figure 6.10. Therefore, q_{Filt2} p_{FO1} can be combined with q_{Filt2} to estimate porosity at the same time.

Sensor Fault Diagnosis

When the flow rate sensor has a drifting fault, the flow rate reading from the sensor will consistently increase or decrease. This case study focuses on the decrease scenario. $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ is used as a feature to estimate porosity, the estimation result is $\hat{\epsilon}_1$.



(a) q_{Filt2} variation with Filter 2 clogging

(b) p_{FO1} variation with Filter 2 clogging

Figure 6.10 Comparing q_{Filt2} and p_{FO1} variation with Filter 2 clogging

Meanwhile, $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ is also used to estimate porosity, the estimation result is $\hat{\epsilon}_2$. The error between the two estimations is used to generate the sensor health index.

$$\text{Error} = |\hat{\epsilon}_2 - \hat{\epsilon}_1| \tag{6.5}$$

Eq. (6.6) is used to convert the Error into sensor health index (SensorHI) as shown in Figure 6.11.

$$\text{SensorHI} = 1 - 20 \times \text{Error} \tag{6.6}$$

LS-UKF Diagnosis for Feature $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$

Process Model

The process model is derived based on the relationship between the pressure of FO Module 1 p_{FO1} and the filter porosity ϵ. When water pass through the filter, particles will remain in the filter and cause the decrease of filter porosity.

$$x(k) = x(k-1) - C2 \times (p_{\text{FO1}}) \times LBG2 + w_k \quad w_k \sim N(0, Q_k) \tag{6.7}$$

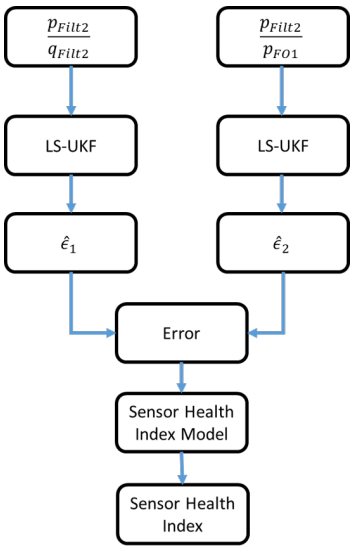


Figure 6.11 Sensor vs component failures: diagnosis framework

where $x(k)$ is the state, which is the porosity of Filter 2 ϵ, p_{FO1} is the pressure of FO Module 1, $C2$ is a constant, $LBG2$ is the Lebesgue length and w_k is the process noise.

Measurement Model in UKF

The measurement model shows the relationship between fault state (porosity of filter ϵ) and feature (the ratio of filter pressure and pressure of FO Module 1 $\frac{p_{\mathrm{Filt2}}}{p_{\mathrm{FO1}}}$). The model is given as follows:

$$z_k = h(x_k) + v_k \quad v_k \sim N(0, R_k) \tag{6.8}$$

where $z_k = \frac{p_{\mathrm{Filt2}}}{p_{\mathrm{FO1}}}$ is the measurement (feature) and v_k is measurement noise, $h(x_k)$ is a nonlinear function that describes the relationship between measurements z_k and filter state x_k, in this case, a Neural Network model trained with operation data.

The neural network has 1 hidden layer with 10 neurons. The input is ϵ and the output is $\frac{p_{\mathrm{Filt2}}}{p_{\mathrm{FO1}}}$. We randomly selected 70% samples as training data, 15% samples as testing data, and 15% samples as validation data. Levenberg-Marquardt algorithm in MATLAB neural network fitting toolbox is used for training. Training automatically

stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

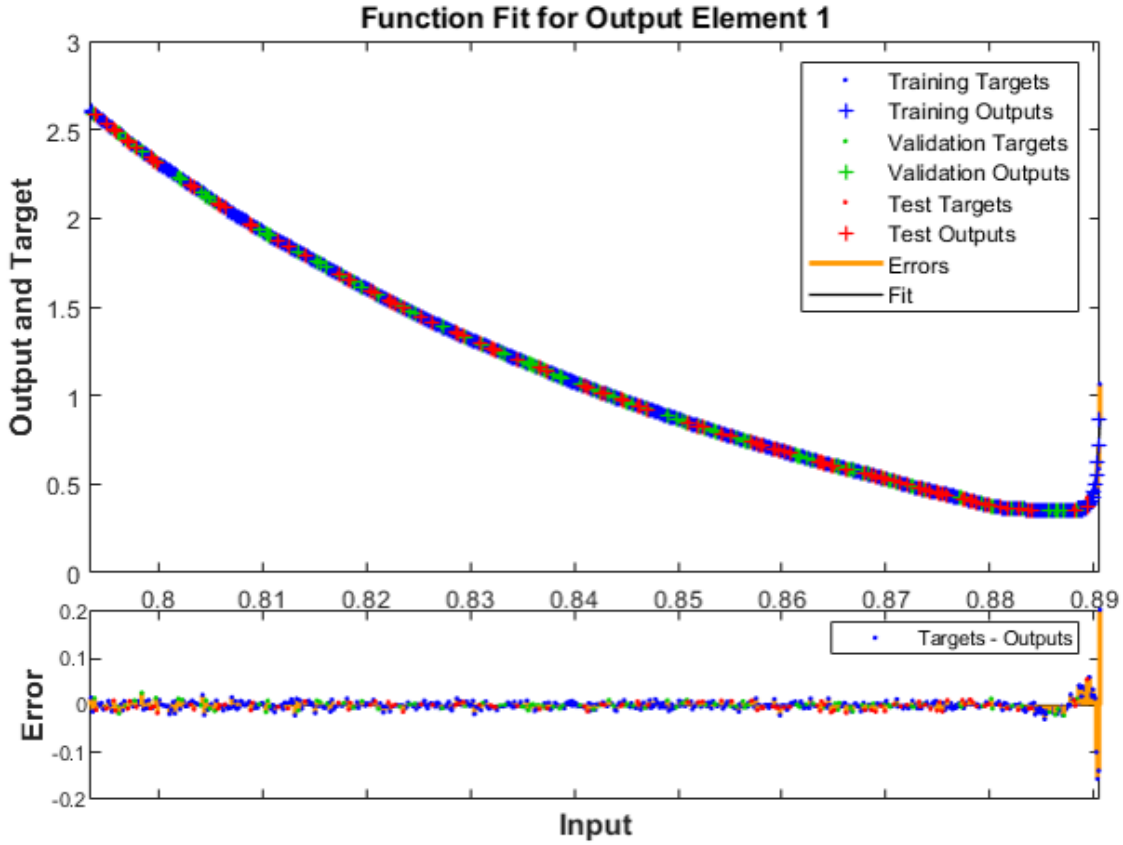


Figure 6.12 Neural network fitting result

Figure 6.12a shows the neural network fitting result, in which the x-axis represents the input ϵ and the y-axis represents the predicted $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ from the model. Ground truth (target value) and output of the model from training dataset, testing dataset and validation dataset are represented by the blue, red and green dots. The errors between targets and outputs of the training dataset, testing dataset and validation dataset are also shown in the figure.

Figure 6.13 shows the modeling error histogram. The blue, green, and red bars represent the errors of model on the training data, validation data, and testing data, respectively. The x-axis represents the errors between the predicted $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ and ground

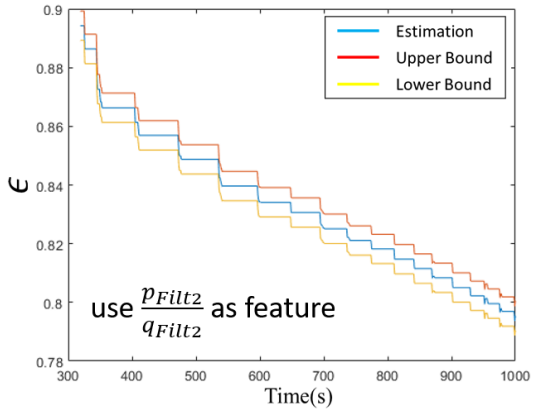
Figure 6.13 Neural network error histogram

truth $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$, while the y-axis is the number of instances (samples) fall into these error ranges.

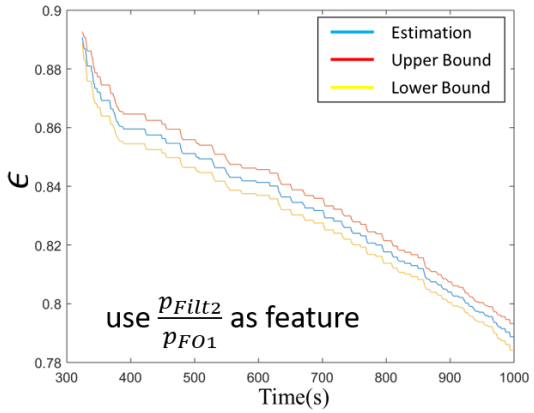
6.4 Simulation Results Comparison

When filter clogging happens without sensor fault, the two features $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ are used to estimate ϵ_1 and ϵ_2 in parallel. The error between the two estimates is calculated by Eq. (6.5). The two features have very similar estimation result when component fault happens, and the error between the two estimates are shown in Figure 6.14c.

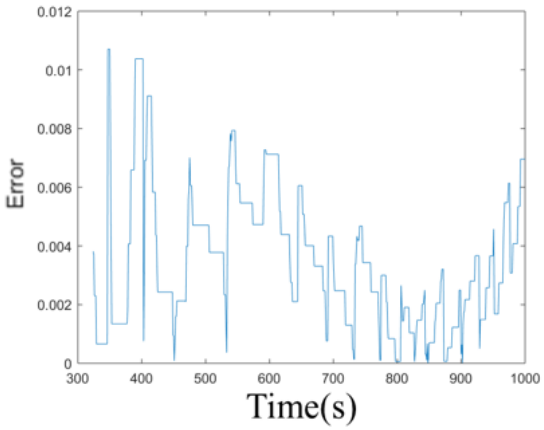
When filter clogging and sensor fault happens at the same time, the two features $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ and $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ are used to estimate ϵ_1 and ϵ_2 in parallel. The error between two estimation is calculated by Eq. (6.5). The two features have different estimates when



(a) Use $\frac{p_{\text{Filt2}}}{q_{\text{Filt2}}}$ to estimate porosity $\hat{\epsilon}_1$



(b) Use $\frac{p_{\text{Filt2}}}{p_{\text{FO1}}}$ to estimate porosity $\hat{\epsilon}_2$



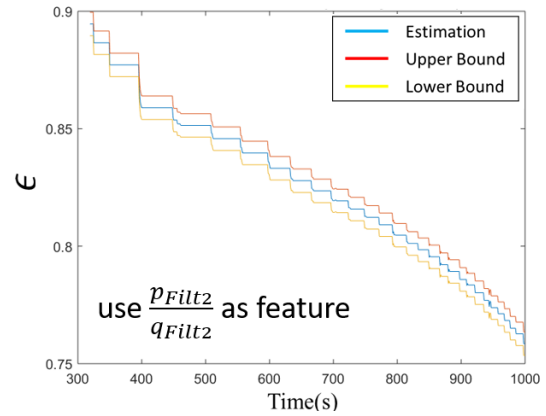
(c) Error between two estimations

Figure 6.14 Error between two estimations when Filter 2 clogging

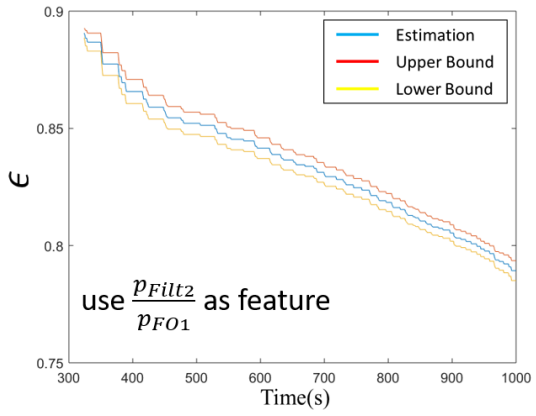
sensor fault happens and the error between the two estimates, shown in Figure 6.15c increases as the sensor drifting fault happens.

Consequently, the error is converted to the sensor health index model to generate a sensor health index by Eq. (6.6). Figure 6.16 shows the comparison between two health indexes from the errors mentioned above. When there is no sensor fault, the error between the two estimates remains small, which generates a high sensor health index, as shown in Figure 6.16a.

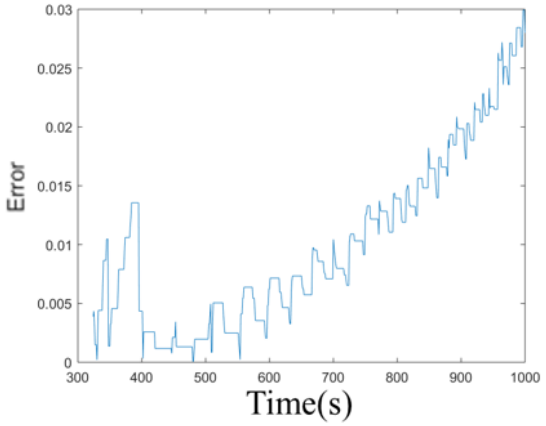
When there is a sensor fault, the error between the two estimates starts to deviate, which generates a low sensor health index, as shown in Figure 6.16b. The sensor fault detected threshold of 0.77 on the health index is established to distinguish the sensor



(a) Use $\frac{p_{\mathrm{Filt2}}}{q_{\mathrm{Filt2}}}$ as feature to estimate porosity $\hat{\epsilon}_1$



(b) Use $\frac{p_{\mathrm{Filt2}}}{p_{\mathrm{FO1}}}$ as feature to estimate porosity $\hat{\epsilon}_2$

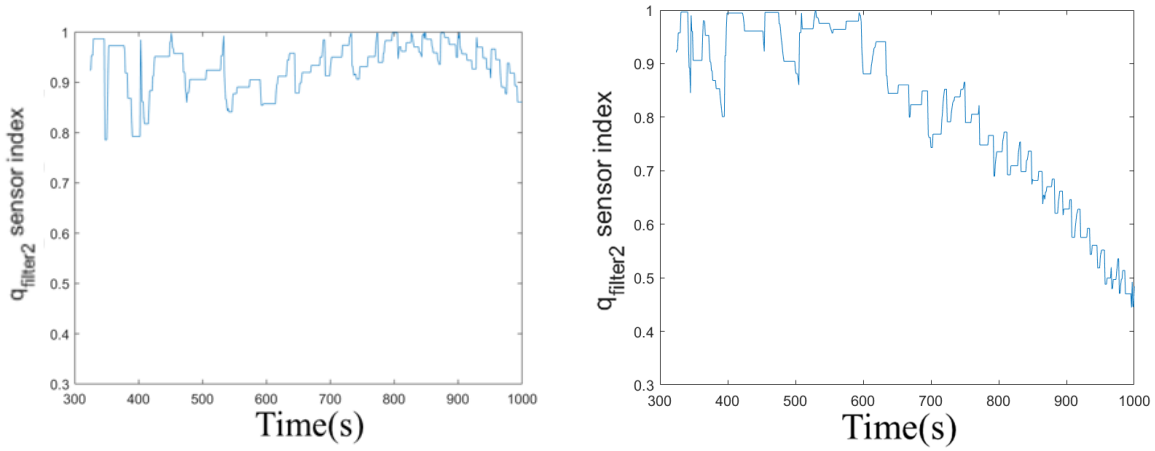


(c) Error between two estimations

Figure 6.15 Error between two estimates with Filter 2 sensor drifting

fault and component fault. In this case study, at the 621st second, the sensor fault is detected.

The simulation results above show that the sensor health index can be obtained by comparing the two estimates of filter parameter from different sensors for sensor fault detection. When the sensor is in good condition, the health index defined based on the difference of two estimates of filter parameter should be close to 1. When the sensor drifting happens, the health index decreases as the error between the two estimates increases. When the health index drops below the threshold, a sensor fault is detected.



(a) Sensor Health Index when Filter 2 clogging (no sensor fault)

(b) Sensor Health Index when Filter 2 clogging (with sensor fault)

Figure 6.16 Comparing sensor health index

Compared with case study 1 that focuses on the flow rate q_{Filt2} and pressure p_{Filt2} relationship under Pump 4 constant voltage condition, case study 2 estimates ϵ from different sensors and uses the error of the two estimates to generate the health index. Therefore, the approach in case study 2 is more generic and can be applied to different operating conditions.

6.5 Conclusion

In this chapter, a systematic sensor fault diagnosis method is developed to distinguish sensor fault and component fault. Two cases studies have been discussed in this chapter using two different method. The first case study is developed based on the measurements' variation difference when component and sensor fault happens. However, this method only apply to certain condition. The second case study is developed by bring extra estimator using existing sensor measurements.The second case study is more generic and more practical to be used in complex physical system.

Chapter 7

Conclusions and Future Works

7.1 Conclusions

To enable effective management, planning, and operations for future space exploration missions that involve a crewed space habitat, operational support must be migrated from Earth to the habitat and be operated at a higher level of autonomy and resilience. In recent years, the developments of Intelligent System Health Management technologies (ISHM) across the industries have brought promising solutions to help ensure the normal operations of future deep space habitats (DSH) in harsh deep space environments while minimizing operational risks and costs. In this thesis, the concepts of ISHM are applied and implemented as an Automatic Contingency Management (ACM) system. The ACM system provides an architecture that embraces a suite of enabling techniques from modeling, simulations to fault diagnosis, prognosis and accommodation. More specifically, a physics-based, discrete component-oriented model is developed for the Grey Water Recycling System (WRS), which represents one of the principal subsystems of the Environmental Control and Life Support Systems (ECLSS). This dynamic model is based on an existing operational WRS platform deployed at NASA Ames Research Center's Sustainability Base. Diagnosis and automated contingency management strategies are developed to enhance the resilience and optimal operations of the system to ensure mission success.

The main contributions of this dissertation are:

1. Developed a physics-based model for the WRS with a high-fidelity filter clogging fault scenario. Unlike most conventional ECLSS models, which are based on a conventional "producer-costumer" relationship, a physics-based dynamic model for WRS is developed. In previous works, filter fault severity is mostly modeled using the coefficient of filter resistance, which does not demonstrate the mechanism of filter clogging. To improve the fidelity of the filter clogging model, the Ergun equation and Darcy's law are introduced. The effectiveness of the improved model is demonstrated by a series of simulation results.

2. Implemented a computationally efficient "Lebesgue sampling – Unscented Kalman Filter" (LS-UKF) algorithm for fault diagnosis and prognosis, applied it to the WRS. Since the computational resource is extremely limited for space operations, the LS-UKF algorithm is introduced to reduce the utilization of the computational resource. In several previous studies, the LS method has been integrated with KF, Particle Filter (PF), and Extended Kalman Filter (EKF) algorithms and applied to other application domains (such as battery diagnosis and prognosis). However, these forms of Kalman Filters are not optimal for DSH applications because PF is computationally expensive and EKF requires closed-form derivatives of system dynamic equations, which is prohibitive. UKF, on the other hand, is a perfect fit due to its performance advantages and relaxed requirements for calculating derivatives. The integration of LS and UKF and the application of LS-UKF algorithm to the WRS are the first attempt in this research area.

3. Developed diagnostics enhanced automatic fault accommodation control strategies. A variant weight MPC algorithm is proposed to mitigate the effect of component faults. The fault diagnosis function provides fault information that an ACM system uses to reconfigure system control strategy to accommodate

current and future faulty conditions. As proof of the concept, a reconfigurable PID-based fault-tolerant control algorithm is implemented to switch the WRS control schedule depending on system health condition, e.g. between constant flow rate mode and constant pressure mode based on the fault severity. To further utilize fault information in system reconfiguration and optimization, a varying weight model predictive control (MPC)-based fault-tolerant control algorithm with prediction capability is developed to allow better tracking of a reference trajectory that optimizes overall system performance throughout the entire mission. Unlike the traditional MPC algorithm which only considers tracking performance from controls perspective, a new algorithm in this work essentially takes into consideration system health information. For example, the proposed time-varying weight MPC algorithm in this study considers two factors: estimated filter porosity and fault detection time in the optimization process.

4. Developed a systematic method that distinguishes sensor fault and component fault in order to improve the reliability of automated systems. Sensor faults in engineering systems are often challenging to avoid. In complex systems, any fault possesses the potential to impact the entire system's behavior. In a manufacturing process, a simple sensor fault may result in off-specification products, higher operation costs, a shutdown of production lines, and environmental damage, etc. In a continuously operating system, ignoring a small fault can lead to disastrous consequences. For instance, a position sensor fault in an automated guided vehicle (AGV) may bring the vehicle towards hazardous directions or cause a collision. Without considering these faulty conditions, fully automated systems are not reliable. However, when component fault happens, there might be multiple measurements related to this component variant, which is different from single sensor fault that

only have one measurement variant. A systematic method utilizes LS-UKF estimation and analytical redundancy in system parameter relationships is developed based on this observation to distinguish sensor fault and component fault.

With the highlights of the aforementioned contributions, as well as many productive ongoing research conducted by NASA and collaborators, it is fair to conclude that the success of future DSH projects can leverage greatly from the advances of ISHM techniques and the development of an Automatic Contingency Management System. The key is to fully understand the unique challenges in harsh deep space environments and the extreme limitation on materials and resources, such that innovative algorithms are developed to address these challenges as attempted in this dissertation.

7.2 Future Work

Based on the work done in this dissertation, a significant amount of effort is still needed in order to improve the Technological Readiness Levels of the proposed methods before they can be deployed in actual systems. At a high level, the following topics need to be addressed thoroughly.

1. The proposed sensor fault detection method focuses on single sensor fault but not simultaneous multiple sensor faults in the whole WRS system. The chance for multiple sensor faults to occur at the same time is minimal. When they occur at different time, they can be tackled separately, leveraging the approach developed in this dissertation.

2. Integration of Remaining Useful Life (RUL) and mission resources to realize mission level optimization. Current research only considers the current health state in the optimization process. However, the future health state could also

be considered. Therefore, the future health state, (e.g., RUL) can be considered as an importation factor in a mission level optimization as well as other factors like system performance and resources.

3. Extend the ISHM and ACM study to cover more sub-systems of the DSH. The current work only covers the WRS system, which is only one of the sub-system of DSH. The idea and algorithms developed in this work can be applied to a broader scope of subsystems in DSH.

Chapter 8

PUBLICATIONS

1. **Shijie Tang**, Bin Zhang, Wuzhao Yan, Ash Thakker, Santiago Vivanco, Rodney Martin, and Craig Moore. "Operation-Aware ISHM for Environmental Control and Life Support in Deep Space Habitants." In 2018 AIAA Information Systems-AIAA Infotech@ Aerospace, p. 1365. 2018.

2. **Shijie Tang**, Guangxing Niu, Bin Zhang, Ash Thakker, Santiago Vivanco, Rodney Martin, and Craig Moore. "Automated Contingency Management for Water Recycling System." In Proceedings of the Annual Conference of the PHM Society, vol. 10, no. 1. 2018.

3. Yan, Wuzhao, Bin Zhang, Guangquan Zhao, **Shijie Tang**, Guangxing Niu, and Xiaofeng Wang. "A battery management system with a Lebesgue-sampling-based extended Kalman filter." IEEE transactions on industrial electronics 66, no. 4 (2018): 3227-3236.

4. Yan, Wuzhao, Guangxing Niu, **Shijie Tang**, and Bin Zhang. "State-of-charge estimation of Lithium-ion batteries by Lebesgue sampling-based EKF method." In IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, pp. 3233-3238. IEEE, 2017.

5. Niu, Guangxing, **Shijie Tang**, and Bin Zhang. "Machine Condition Prediction Based on Long Short Term Memory and Particle Filtering." In IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society, pp. 5942-5947. IEEE, 2018.

6. Liu, Enhui, Guangxing Niu, **Shijie Tang**, Bin Zhang, Jesse Williams, Rodney Martin, and Craig Moore. "Permanent Magnet Synchronous Motor Winding Fault Simulation and Diagnosis." In Proceedings of the Annual Conference of the PHM Society, vol. 11, no. 1. 2019.

Bibliography

[1] R. L. Carrasquillo, J. L. Reuter, and C. L. Philistine. *Summary of resources for the international space station environmental control and life support system.* Tech. rep. SAE Technical Paper, 1997.

[2] R. G. Schunk and W. R. Humphries. "Environmental Control and Life Support Testing at the Marshall Space Flight Center". In: *SAE Transactions* (1987), pp. 548–558.

[3] B. R. Dunaway and D. M. Cudd. *Impact of Shuttle Orbiter Booster Fan Bypass on Integrated Environmental Control and Life Support Systems.* Tech. rep. SAE Technical Paper, 2006.

[4] R. M. Bagdigian, R. L. Carrasquillo, J. Metcalf, and L. Peterson. "National Aeronautics and Space Administration (NASA) Environmental Control and Life Support (ECLS) Capability Roadmap Development for Exploration". In: (2012).

[5] W. Mitsch. "Ecological engineering a cooperative role with the planetary life-support system". In: *Environmental Science & Technology* 27.3 (1993), pp. 438–445.

[6] A. Saxena, M. E. Orchard, B. Zhang, G. Vachtsevanos, L. Tang, Y. Lee, and Y. Wardi. "Automated Contingency Management for Propulsion Systems". In: *2007 European Control Conference (ECC)*. IEEE. 2007, pp. 3515–3522.

[7] L. Tang, G. J. Kacprzynski, K. Goebel, A. Saxena, B. Saha, and G. Vachtsevanos. "Prognostics-enhanced automated contingency management

for advanced autonomous systems". In: *2008 International Conference on Prognostics and Health Management*. IEEE. 2008, pp. 1–9.

[8] D. Kortenkamp and S. Bell. "BioSim: An integrated simulation of an advanced life support system for intelligent control research". In: *Proc. of the 7th Intl. Symposium on Artificial Intelligence, Robotics and Automation in Space*. 2003.

[9] S. Do, A. Owens, and O. d. Weck. "HabNet–An Integrated Habitation and Supportability Architecting and Analysis Environment". In: 45th International Conference on Environmental Systems. 2015.

[10] C. Poolla, A. Ishihara, S. Rosenberg, R. Martin, A. Fong, S. Ray, and C. Basu. "Neural network forecasting of solar power for NASA Ames sustainability base". In: *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG)*. IEEE. 2014, pp. 1–8.

[11] A. Basak, O. Mengshoel, S. Hosein, and R. Martin. "Scalable causal learning for predicting adverse events in smart buildings". In: *workshops at the Thirtieth AAAI Conference on Artificial Intelligence*. 2016.

[12] P. Elbert, S. Ebbesen, and L. Guzzella. "Implementation of dynamic programming for n-dimensional optimal control problems with final state constraints". In: *IEEE Transactions on Control Systems Technology* 21.3 (2012), pp. 924–931.

[13] I. Roychoudhury, V. Hafiychuk, and K. Goebel. "Model-based diagnosis and prognosis of a water recycling system". In: *2013 IEEE Aerospace Conference*. IEEE. 2013, pp. 1–9.

[14] E. Liu, G. Niu, S. Tang, B. Zhang, J. Williams, R. Martin, and C. Moore. "Permanent Magnet Synchronous Motor Winding Fault Simulation and Diagnosis". In: *Proceedings of the Annual Conference of the PHM Society*. Vol. 11. 1. 2019.

[15] J. Zaytoon and S. Lafortune. “Overview of fault diagnosis methods for discrete event systems”. In: *Annual Reviews in Control* 37.2 (2013), pp. 308–320.

[16] S. Moritz and T. Bartz-Beielstein. “imputeTS: time series missing value imputation in R”. In: *The R Journal* 9.1 (2017), pp. 207–218.

[17] K. Potdar, T. S. Pardawala, and C. D. Pai. “A comparative study of categorical variable encoding techniques for neural network classifiers”. In: *International journal of computer applications* 175.4 (2017), pp. 7–9.

[18] W. Zhao and S. Du. “Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.8 (2016), pp. 4544–4554.

[19] W. Sun, J. Chen, and J. Li. “Decision tree and PCA-based fault diagnosis of rotating machinery”. In: *Mechanical Systems and Signal Processing* 21.3 (2007), pp. 1300–1317.

[20] L. Xu, J. Huang, H. Wang, and B. Long. “A novel method for the diagnosis of the incipient faults in analog circuits based on LDA and HMM”. In: *Circuits, Systems and Signal Processing* 29.4 (2010), pp. 577–600.

[21] M. Misra, H. H. Yue, S. J. Qin, and C. Ling. “Multivariate process monitoring and fault diagnosis by multi-scale PCA”. In: *Computers & Chemical Engineering* 26.9 (2002), pp. 1281–1293.

[22] Z. Wang, Y. Liu, and P. J. Griffin. “A combined ANN and expert system tool for transformer fault diagnosis”. In: *2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 00CH37077)*. Vol. 2. IEEE. 2000, pp. 1261–1269.

[23] N Saravanan and K. Ramachandran. “Incipient gear box fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using

artificial neural network (ANN)". In: *Expert Systems with Applications* 37.6 (2010), pp. 4168–4181.

[24] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[25] G. Zhao, X. Liu, B. Zhang, Y. Liu, G. Niu, and C. Hu. "A novel approach for analog circuit fault diagnosis based on deep belief network". In: *Measurement* 121 (2018), pp. 170–178.

[26] T. De Bruin, K. Verbert, and R. Babuška. "Railway track circuit fault diagnosis using recurrent neural networks". In: *IEEE transactions on neural networks and learning systems* 28.3 (2016), pp. 523–533.

[27] S. Guo, B. Zhang, T. Yang, D. Lyu, and W. Gao. "Multi-Task Convolutional Neural Network with Information Fusion for Bearing Fault Diagnosis and Localization". In: *IEEE Transactions on Industrial Electronics* (2019).

[28] G. Niu, S. Tang, and B. Zhang. "Machine Condition Prediction Based on Long Short Term Memory and Particle Filtering". In: *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2018, pp. 5942–5947.

[29] D. Zhou, Y. Zhao, Z. Wang, X. He, and M. Gao. "Review on diagnosis techniques for intermittent faults in dynamic systems". In: *IEEE Transactions on Industrial Electronics* 67.3 (2019), pp. 2337–2347.

[30] J. De Kleer and J. Kurien. "Fundamentals of model-based diagnosis". In: *IFAC Proceedings Volumes* 36.5 (2003), pp. 25–36.

[31] Y. Zhang and J. Jiang. "Bibliographical review on reconfigurable fault-tolerant control systems". In: *Annual reviews in control* 32.2 (2008), pp. 229–252.

[32] S. Huang, K. K. Tan, and T. H. Lee. "Fault diagnosis and fault-tolerant control in linear drives using the Kalman filter". In: *IEEE Transactions on Industrial Electronics* 59.11 (2012), pp. 4285–4292.

[33] W. Yan, B. Zhang, G. Zhao, S. Tang, G. Niu, and X. Wang. "A battery management system with a Lebesgue-sampling-based extended Kalman filter". In: *IEEE Transactions on Industrial Electronics* 66.4 (2018), pp. 3227–3236.

[34] M. Sepasi and F. Sassani. "On-line fault diagnosis of hydraulic systems using unscented Kalman filter". In: *International Journal of Control, Automation and Systems* 8.1 (2010), pp. 149–156.

[35] W. Yan, B. Zhang, X. Wang, W. Dou, and J. Wang. "Lebesgue-sampling-based diagnosis and prognosis for lithium-ion batteries". In: *IEEE Transactions on Industrial Electronics* 63.3 (2015), pp. 1804–1812.

[36] W. Yan, G. Niu, S. Tang, and B. Zhang. "State-of-charge estimation of Lithium-ion batteries by Lebesgue sampling-based EKF method". In: *IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2017, pp. 3233–3238.

[37] H. Yoshida, T. Iwami, H. Yuzawa, and M. Suzuki. *Typical faults of air conditioning systems and fault detection by ARX model and extended Kalman filter*. Tech. rep. American Society of Heating, Refrigerating and Air-Conditioning Engineers ..., 1996.

[38] J. Carpenter, P. Clifford, and P. Fearnhead. "Improved particle filter for nonlinear problems". In: *IEE Proceedings-Radar, Sonar and Navigation* 146.1 (1999), pp. 2–7.

[39] R. Zhang, R. Lu, A. Xue, and F. Gao. "New minmax linear quadratic fault-tolerant tracking control for batch processes". In: *IEEE Transactions on Automatic Control* 61.10 (2015), pp. 3045–3051.

[40] D. M. Raimondo, G. R. Marseglia, R. D. Braatz, and J. K. Scott. "Fault-tolerant model predictive control with active fault isolation". In: *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE. 2013, pp. 444–449.

[41] X. Wang and N. Hovakimyan. "L1 adaptive controller for nonlinear time-varying reference systems". In: *Systems & Control Letters* 61.4 (2012), pp. 455–463.

[42] K. van Berkel, B. de Jager, T. Hofman, and M. Steinbuch. "Implementation of dynamic programming for optimal control problems with continuous states". In: *IEEE Transactions on Control Systems Technology* 23.3 (2014), pp. 1172–1179.

[43] O. Sundström, D. Ambühl, and L. Guzzella. "On implementation of dynamic programming for optimal control problems with final state constraints". In: *Oil & Gas Science and Technology–Revue de l'Institut Français du Pétrole* 65.1 (2010), pp. 91–102.

[44] R. Findeisen, L. Imsland, F. Allgower, and B. A. Foss. "State and output feedback nonlinear model predictive control: An overview". In: *European journal of control* 9.2-3 (2003), pp. 190–206.

[45] H. A. Izadi, Y. Zhang, and B. W. Gordon. "Fault tolerant model predictive control of quad-rotor helicopters with actuator fault estimation". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 6343–6348.

[46] M. Flynn, M. Hightower, L. Delzeit, S. Gormly, M. Hammoudeh, H. Shaw, T Richardson, and K. Howard. "Sustainability base graywater recycling system". In: *41st International Conference on Environmental Systems*. 2011.

[47] H. Zheng. "Chapter 2 - Solar Energy Utilization and Its Collection Devices". In: *Solar Energy Desalination Technology*. Ed. by H. Zheng. Amsterdam: Elsevier, 2017, pp. 47 –171. ISBN: 978-0-12-805411-6. DOI: https://doi.org/10.1016/B978-0-12-805411-6.00002-6. URL: http://www.sciencedirect.com/science/article/pii/B9780128054116000026.

[48] I. Roychoudhury, V. Hafiychuk, and K. Goebel. "Model-based diagnosis and prognosis of a water recycling system". In: *2013 IEEE Aerospace Conference*. IEEE, 2013, pp. 1–9. DOI: 10.1109/AERO.2013.6496976.

[49] E. A. Wan and R. Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee. 2000, pp. 153–158.

[50] G. A. Terejanu. "Unscented kalman filter tutorial". In: *University at Buffalo, Buffalo* (2011).

[51] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[52] F. Allgower, R. Findeisen, Z. K. Nagy, et al. "Nonlinear model predictive control: From theory to application". In: *Journal-Chinese Institute Of Chemical Engineers* 35.3 (2004), pp. 299–316.