

SENSOR FAULT DETECTION AND ISOLATION SYSTEM

A Dissertation

by

CHENG-KEN YANG

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

DOCTOR OF PHILOSOPHY

Chair of Committee,	Reza Langari
Committee Members,	Alexander Parlos
	Sivakumar Rathinam
	Stefan Hurlbaas
Head of Department,	Andreas A. Polycarpou

August 2014

Major Subject: Mechanical Engineering

Copyright 2014 Cheng-Ken Yang

## ABSTRACT

The purpose of this research is to develop a Fault Detection and Isolation (FDI) system which is capable to diagnosis multiple sensor faults in nonlinear cases. In order to lead this study closer to real world applications in oil industries, the system parameters of the applied system are assumed to be unknown. In the first step of the proposed method, phase space reconstruction techniques are used to reconstruct the phase space of the applied system. This step is aimed to infer the system property by the collected sensor measurements. The second step is to use the reconstructed phase space to predict future sensor measurements, and residual signals are generated by comparing the actually measured measurements to the predicted measurements. Since, in practice, residual signals will not perfectly equal to zero in the fault-free situation, Multiple Hypothesis Shiriyayev Sequential Probability Test (MHSSPT) is introduced to further process those residual signals, and the diagnostic results are presented in probability. In addition, the proposed method is extended to a non-stationary case by using the conservation/dissipation property in phase space.

The proposed method is examined by both of simulated data and real process data to support that it is capable of detecting and isolating multiple sensor faults in nonlinear cases. In the section of simulation results, a three tank model is introduced for generating simulated data. The three tank model is modeled according to a nonlinear laboratory setup DTS200. On the other hand, in the section of experimental results, the real process data collected from a sugar factory actuator system are used to examine the proposed method.

According to our results obtained from simulations and experiments, the proposed method is capable to indicate both of healthy and faulty situations. These results further confirm that the proposed method is able to deal with not only simulated data but also real process data.

DEDICATION

To My Dear Parents

and

To My Brother, Shao-Hui Yang

and

To My Lovely Wife, Meng-Ying Wu

## ACKNOWLEDGEMENTS

I would like to show my deepest appreciation to everyone who has helped me obtain my Ph.D. degree. Without their kind assistance, I would not have been able to achieve my goal. To those who are missing in this brief list, I sincerely apologize.

First, I would like to thank my advisor, Dr. Langari. His knowledge, experience, and precious advice are really helpful for this research.

Second, I would like to thank my committee members, Dr. Parlos, Dr. Rathinam, and Dr. Hurlebaus, for their guidance throughout this research.

Many thanks are also owed to my friends, colleagues, and the department faculty and staff for their assistance.

Finally, thank my parents for their encouragement and my wife for her patience and love.

## TABLE OF CONTENTS

	Page
ABSTRACT .....	ii
DEDICATION.....	iv
ACKNOWLEDGEMENTS .....	v
TABLE OF CONTENTS.....	vi
LIST OF FIGURES .....	viii
LIST OF TABLES.....	xvi
1. INTRODUCTION .....	1
1.1 Importance and Preliminary Background.....	1
1.2 Problem Statement.....	8
2. LITERATURE REVIEW.....	13
2.1 Model-based Methods.....	13
2.2 Process History-based (Data-driven) Methods.....	20
3. THE PROPOSED METHOD.....	53
3.1 Phase Space Reconstruction.....	54
3.2 Smooth Mechanism.....	64
3.3 Prediction Mechanism.....	68
3.4 Conservation/Dissipation Property in Phase Space.....	71
4. SIMULATION RESULTS.....	79
4.1 Generating Training and Testing Data.....	79
4.2 Results of the Blocks of Phase Space Reconstruction and Smooth Mechanism.....	88
4.3 Results of the Block of Prediction Mechanism.....	93
4.4 Residual Signal Processing.....	99
4.5 Results of the Block of Conservation in Phase Space.....	112
5. EXPERIMENTAL RESULTS .....	116
5.1 DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems).....	116

5.2 Results of Applying the Proposed Method to DAMADICS.....	126
5.3 Results of Applying the Proposed Method to THIBAR22C Unit.....	153
6. CONCLUSIONS .....	159
REFERENCES .....	161

## LIST OF FIGURES

	Page
Figure 1(a). The hardware redundancy based fault diagnosis scheme (Adapted from [4]).....	4
Figure 1(b). The unit operation of the product condenser in Tennessee Eastman process (Adapted from [13]).....	5
Figure 1(c). Implementation of hardware redundancy in the unit operation of the product condenser in Tennessee Eastman process (Adapted from [13]).....	5
Figure 2. The scheme of plausibility testing (Adapted from [4]).....	6
Figure 3. The signal processing based fault diagnosis scheme (Adapted from [4]).....	7
Figure 4. The software/analytical redundancy based fault diagnosis scheme (Adapted from [4]).....	8
Figure 5. The continuous stirred tank heater (CSTH) (Adapted from [14]).....	10
Figure 6. The architecture of autoassociative neural network (Adapted from [30]).....	31
Figure 7. Sensor monitoring module in Hines' method (Adapted from [34]).....	33
Figure 8. The illustration of the Euclidean partition and the projection partition (Adapted from [40]).....	46
Figure 9. The illustration of misclassification.....	48
Figure 10. The structure of the proposed method.....	54
Figure 11. Time series of $x$ , $y$ , and $z$ in Lorenz equations.....	61
Figure 12. The result of applying mutual information method to the time series of $x$ .....	62
Figure 13. The result of applying FNN method to the time series of $x$ .....	63
Figure 14. The reconstructed phase space of the time series of $x$ .....	63
Figure 15. The illustration of the prediction approach; arrow represents the direction of state flow (Adapted from [53]).....	70
Figure 16. The illustration of the setpoint changes problem.....	72



Figure 17. The phase space of a three states dynamic system (Adapted from [64]).....	73
Figure 18. An example for illustrating how the estimation mechanism works in the first case.....	77
Figure 19. DTS200 setup (Adapted from [4]).....	80
Figure 20. The block diagram of the three tank system.....	81
Figure 21. The illustration of modeling measurement noise.....	82
Figure 22. Training data - sensor 1 output with 10% measurement noise.....	83
Figure 23. Training data - sensor 1 output with 5% measurement noise.....	84
Figure 24. Training data - sensor 1 output with 1.5% measurement noise.....	84
Figure 25. The result of subtracting healthy testing data from training data in the case of 5% noise.....	86
Figure 26. The comparison of abrupt fault, incipient fault, and fault-free cases with 5% noise level.....	88
Figure 27. Reconstruct phase space by training data – Sensor 1 output with 5% noise - Mutual information.....	89
Figure 28. Reconstruct phase space by training data – Sensor 1 output with 5% noise - FNN.....	90
Figure 29. The reconstructed phase space of the sensor 1 in the three tank system.....	90
Figure 30. The smoothed trajectory by using MVU estimator.....	91
Figure 31. The magnitude of the generalized eigenvalues for the smooth orthogonal decomposition.....	92
Figure 32. The smoothed trajectory by using SOD.....	92
Figure 33. Raw residual signal in the fault-free situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.....	94
Figure 34. Raw residual signal in the abrupt fault situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.....	94

Figure 35. Raw residual signal in the incipient fault situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.....	95
Figure 36. Raw residual signal in the fault-free situation with 5% noise produced by using the SOD smoothed trajectory for prediction.....	97
Figure 37. Raw residual signal in the abrupt fault situation with 5% noise produced by using the SOD smoothed trajectory for prediction.....	98
Figure 38. Raw residual signal in the incipient fault situation with 5% noise produced by using the SOD smoothed trajectory for prediction.....	98
Figure 39. Residual signal generated by setting a threshold in the fault-free situation with 1.5% noise.....	100
Figure 40. Residual signal generated by setting a threshold in the abrupt fault situation with 1.5% noise.....	100
Figure 41. Residual signal generated by setting a threshold in the incipient fault situation with 1.5% noise.....	101
Figure 42. Residual signal generated by setting a threshold in the fault-free situation with 5% noise.....	102
Figure 43. Residual signal generated by setting a threshold in the abrupt fault situation with 5% noise.....	103
Figure 44. Residual signal generated by setting a threshold in the incipient fault situation with 5% noise.....	103
Figure 45. Residual signal generated by setting a threshold in the fault-free situation with 10% noise.....	104
Figure 46. Residual signal generated by setting a threshold in the abrupt fault situation with 10% noise.....	104
Figure 47. Residual signal generated by setting a threshold in the incipient fault situation with 10% noise.....	105
Figure 48. The illustration of using the second set of training data to choose parameters for the MHSSPT.....	107
Figure 49. The probability of the occurrence of fault in the fault-free situation with 5% noise.....	109

Figure 50. The probability of the occurrence of fault in the abrupt fault situation with 5% noise.....	109
Figure 51. The probability of the occurrence of fault in the incipient fault situation with 5% noise.....	110
Figure 52. The probability of the occurrence of fault in the fault-free situation with 10% noise.....	110
Figure 53. The probability of the occurrence of fault in the abrupt fault situation with 10% noise.....	111
Figure 54. The probability of the occurrence of fault in the incipient fault situation with 10% noise.....	111
Figure 55. The probability of the occurrence of fault in the fault-free situation with 5% noise - FDI by conservation/dissipation property.....	113
Figure 56. The probability of the occurrence of fault in the abrupt fault situation with 5% noise - FDI by conservation/dissipation property.....	113
Figure 57. The probability of the occurrence of fault in the incipient fault situation with 5% noise - FDI by conservation/dissipation property.....	114
Figure 58. The probability of the occurrence of fault in the fault-free situation with 10% noise - FDI by conservation/dissipation property.....	114
Figure 59. The probability of the occurrence of fault in the abrupt fault situation with 10% noise - FDI by conservation/dissipation property.....	115
Figure 60. The probability of the occurrence of fault in the incipient fault situation with 10% noise - FDI by conservation/dissipation property.....	115
Figure 61. The scheme of the actuator (Adapted from [66]).....	116
Figure 62. The block diagram of the actuator and its application (Adapted from [66]).....	117
Figure 63. Illustration of inputs and outputs of the ACT block (Adapted from [69]).....	120
Figure 64. Illustration of noise entries in the ACT block (Adapted from [69]).....	121
Figure 65. Control input of the ACT block for generating simulated process data.....	121
Figure 66. Subtracting healthy testing data from training data in 5% noise case.....	122

Figure 67. The flow rate sensor measurements in the cases of abrupt fault, incipient fault, and fault-free contaminated with 5% noise.....	124
Figure 68(a). The raw residual signal produced in the healthy situation with 1.5% noise.....	127
Figure 68(b). The probability of the occurrence of fault in the healthy situation with 1.5% noise.....	128
Figure 69(a). The raw residual signal produced in the abrupt fault situation with 1.5% noise.....	128
Figure 69(b). The probability of the occurrence of fault in the abrupt fault situation with 1.5% noise.....	129
Figure 69(c). The enlargement of Fig. 69(b).....	129
Figure 70(a). The raw residual signal produced in the incipient fault situation with 1.5% noise.....	130
Figure 70(b). The probability of the occurrence of fault in the incipient fault situation with 1.5% noise.....	130
Figure 70(c). The enlargement of Fig. 70(b).....	131
Figure 71(a). The raw residual signal produced in the healthy situation with 5% noise.....	132
Figure 71(b). The probability of the occurrence of fault in the healthy situation with 5% noise.....	132
Figure 72(a). The raw residual signal produced in the abrupt fault situation with 5% noise.....	133
Figure 72(b). The probability of the occurrence of fault in the abrupt fault situation with 5% noise.....	133
Figure 72(c). The enlargement of Fig. 72(b).....	134
Figure 73(a). The raw residual signal produced in the incipient fault situation with 5% noise.....	134
Figure 73(b). The probability of the occurrence of fault in the incipient fault situation with 5% noise.....	135
Figure 73(c). The enlargement of Fig. 73(b).....	135

Figure 74(a). The raw residual signal produced in the healthy situation with 10% noise.....	136
Figure 74(b). The probability of the occurrence of fault in the healthy situation with 10% noise.....	137
Figure 75(a). The raw residual signal produced in the abrupt fault situation with 10% noise.....	137
Figure 75(b). The probability of the occurrence of fault in the abrupt fault situation with 10% noise.....	138
Figure 75(c). The enlargement of Fig. 75(b).....	138
Figure 76(a). The raw residual signal produced in the incipient fault situation with 10% noise.....	139
Figure 76(b). The probability of the occurrence of fault in the incipient fault situation with 10% noise.....	139
Figure 76(c). The enlargement of Fig. 76(b).....	140
Figure 77. The probability of the occurrence of fault in the fault-free situation with 1.5% noise - FDI by conservation/dissipation property.....	141
Figure 78(a). The probability of the occurrence of fault in the abrupt fault situation with 1.5% noise - FDI by conservation/dissipation property.....	142
Figure 78(b). The enlargement of Fig. 78(a).....	142
Figure 79. The probability of the occurrence of fault in the incipient fault situation with 1.5% noise - FDI by conservation/dissipation property.....	143
Figure 80. The probability of the occurrence of fault in the fault-free situation with 5% noise - FDI by conservation/dissipation property.....	143
Figure 81(a). The probability of the occurrence of fault in the abrupt fault situation with 5% noise - FDI by conservation/dissipation property.....	144
Figure 81(b). The enlargement of Fig. 81(a).....	144
Figure 82. The probability of the occurrence of fault in the incipient fault situation with 5% noise - FDI by conservation/dissipation property.....	145
Figure 83. The probability of the occurrence of fault in the fault-free situation with 10% noise - FDI by conservation/dissipation property.....	145

Figure 84(a). The probability of the occurrence of fault in the abrupt fault situation with 10% noise - FDI by conservation/dissipation property.....	146
Figure 84(b). The enlargement of Fig. 84(a).....	146
Figure 85. The probability of the occurrence of fault in the incipient fault situation with 10% noise - FDI by conservation/dissipation property.....	147
Figure 86. The training data used for reconstructing phase space of the actuator system.....	149
Figure 87. The result of mutual information method applied to the training data.....	149
Figure 88. The result of FNNs applied to the training data.....	150
Figure 89. The result of the proposed method applied to real process data collected from the actuator system in the healthy situation.....	150
Figure 90. The result of the prediction mechanism block in the healthy situation.....	151
Figure 91(a). The result of the proposed method applied to real process data collected from the actuator system in the faulty situation.....	151
Figure 91(b). The enlargement of Fig. 91(a).....	152
Figure 92. The result of the prediction mechanism block in the faulty situation.....	152
Figure 93. The training data used for reconstructing phase space of the THIBAR22C unit.....	154
Figure 94. The result of applying mutual information method for reconstructing the phase space of THIBAR22C unit.....	155
Figure 95. The result of applying FNNs method for reconstructing the phase space of THIBAR22C unit.....	155
Figure 96(a). The raw residual signal produced by applying the proposed method to real process data collected from the THIBAR22C unit in the healthy situation.....	156
Figure 96(b). The raw residual signal produced by applying the proposed method to the second healthy testing data collected from the THIBAR22C unit.....	156
Figure 97. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the healthy situation.....	157

Figure 98. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the abrupt fault situation.....157

Figure 99. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the incipient fault situation.....158

## LIST OF TABLES

	Page
Table 1. Parameters of DTS200 (Adapted from [4]).....	81
Table 2. Calculated variance of the raw residual signals in fault-free situation. ....	97
Table 3. The dimension of smoothed trajectories with three different noise levels. ....	97
Table 4. The thresholds for three different noise levels.....	101
Table 5. Description of sensor outputs in the actuator system (Adapted from [66]).....	118
Table 6. Description of DABLib blocks (Adapted from [69]).....	119
Table 7(a). Description of inputs of the ACT block (Adapted from [69]).....	120
Table 7(b). Description of outputs of the ACT block (Adapted from [69]).....	120
Table 8. Description of the flow rate sensor fault (Adapted from [69]).....	123
Table 9. Description of actuator faults(Adapted from [70]).....	125
Table 10. Description of artificially added actuator faults in the 3 <sup>rd</sup> actuator (Adapted from [70]).....	126



# 1. INTRODUCTION

## **1.1 Importance and Preliminary Background**

Fault detection and diagnosis is the central component of abnormal event management (AEM) [1-3]. Because of the increasing needs for higher system performance, product quality, human safety, and cost efficiency, fault diagnosis systems are applied in diverse industrial fields, such as petrochemical and petroleum industries, robotics, and automotive/aerospace systems [4, 5]. According to the International Federation of Automatic Control (IFAC), a fault is defined as an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/usual/standard condition [6-8]. If the unpermitted deviation grows worse with time, a fault may result in abnormal events or accidents. According to statistics from Abnormal Situation Management Consortium (ASM) in 2007 and 2010, abnormal situations cost the process industries billions of dollars per year in the United States [9]. In addition to the monetary losses, there has been an increasing interest due to human safety. A well-known case is the explosion caused by a gas leak at the Kuwait Petrochemical's Mina Al-Ahmedi refinery in June of 2000 [1]. The Indian Express Newspaper reported that at least fifty workers were killed or injured in this explosion in July of 2000. Another explosion with the same reason was reported by the Kuwait National Petroleum Corporation (KNPC) in Oct of 2011, and four workers were killed in this accident. In United States, a well-known example is the Texas City Refinery explosion in March of 2005 killing fifteen workers and injuring near two hundred employees [10]. This explosion was occurred at an

isomerization process unit at BP's Texas City refinery. Moreover, according to the reports from both BP and Chemical Safety and Hazard Investigation Board (CSB), "Inoperative alarms and level sensors" is one of BP's technical failings which leads to this tragedy [10]. Although the catastrophic events are generally rare happened, minor accidents are very common [1]. These minor accidents result in many workplace injuries which is a serious issue for the society. Based on the latest statistics from the US Bureau of Labor Statistics, nearly 3 million nonfatal workplace injuries were reported in 2011<sup>1</sup>. Furthermore, these workplace injuries also cause an economic burden to our society; the National Safety Council reported that workplace injuries and illness cost U. S. government more than US\$39BN in 2010 due to 1.7MM injuries in low wage occupations<sup>2</sup>. The above issues can be relieved if one can diagnose a fault before it becomes an accident. Therefore, fault detection and diagnosis plays an important role for both reasons of economics and human safety.

Before proceeding with the discussion on fault detection and diagnosis, there are several basic concepts needed to be defined. These concepts were initially introduced by Willsky in 1976 and developed in both Ding's and Gertler's works [4, 11, 12]. Generally, there are three parts in a fault diagnosis system: detection, isolation, and identification [4, 12]. Detection is to indicate the occurrence of faults, that lead to undesired or intolerable behavior in the process; isolation is to determine the exact location of faults; identification is to determine the type, magnitude and cause of faults [4, 11]. Depending on the

---

<sup>1</sup> The occupational injuries and illness report is released in October 2012 by U.S. Department of Labor.

<sup>2</sup> The related statistics is released in December 2012 by the University of California at Davis.

performance of fault diagnosis systems, there are FD (for fault detection) or FDI (for fault detection and isolation) or FDIA (for fault detection, isolation and analysis) systems [4].

After defining the necessary functions in a fault detection and diagnosis system, this paragraph is to classify different types of FDI systems from a methodological perspective. These works were started from Willsky in 1976 [11], and continuously updated by Venkatasubramanian, Gertler, Ding, and Hwang et al. among others [1-4, 8, 12]. The following classifications are mainly based on Ding's work but combined with others. FDI methods can be roughly classified into four different schemes: hardware redundancy based fault diagnosis, signal processing based fault diagnosis, plausibility testing, and software/analytical redundancy based fault diagnosis<sup>3</sup> [4].

Hardware redundancy based fault diagnosis is implemented by constructing identical (redundant) hardware components. The main idea of hardware redundancy based fault diagnosis is shown in Fig. 1(a). A fault is detected by the difference of the output of the given process component and that of the identical hardware component. The advantages of this scheme are its high reliability and direct fault isolation. An example of hardware redundancy is the duplicate power supply for an online server. The duplicated power supply is used to generate hardware redundancy for fault detection and replace the original power supply when it is faulty. However, the main disadvantage is obviously due to its high costs. In addition, this scheme may complicate the original problem in some applications. To illustrate this point, here we use the unit operation of the product

---

<sup>3</sup> In Venkatasubramanian's and Gertler's works, they classify signal processing based schemes and plausibility tests to software/analytical redundancy based fault diagnosis.

condenser in Tennessee Eastman process as the example shown in Fig. 1(b). The purpose of the unit operation is aimed to cool down the temperature of the input product flow in order to transfer the status of the product from vapor to liquid [13]. In this unit operation, there are two inputs: product flow and cooling water flow. The input product flow is given from the previous unit operation, and the input cooling water flow rate is adjusted according to the temperature of product outflow. To implement hardware redundancy, the components used to adjust cooling water flow rate are duplicated shown as Fig. 1(c). Similar to the previous example, the duplicated components serve as the reference for fault detection and the replacement of the original components when it is faulty. However, in order to achieve these purposes, the boundary and initial conditions of the heat conduction problem between cooling water flow and product flow will become more complicated so that the original problem of adjusting cooling water flow rate will also become more difficult.

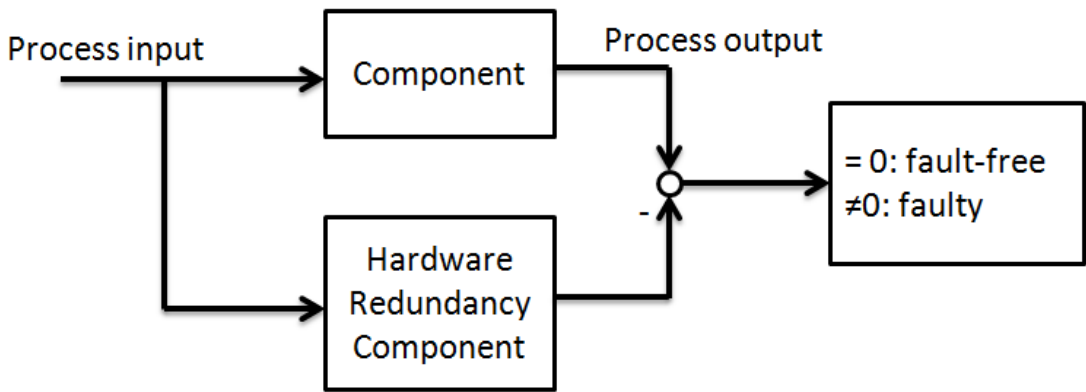


Figure 1(a). The hardware redundancy based fault diagnosis scheme. (Adapted from [4])

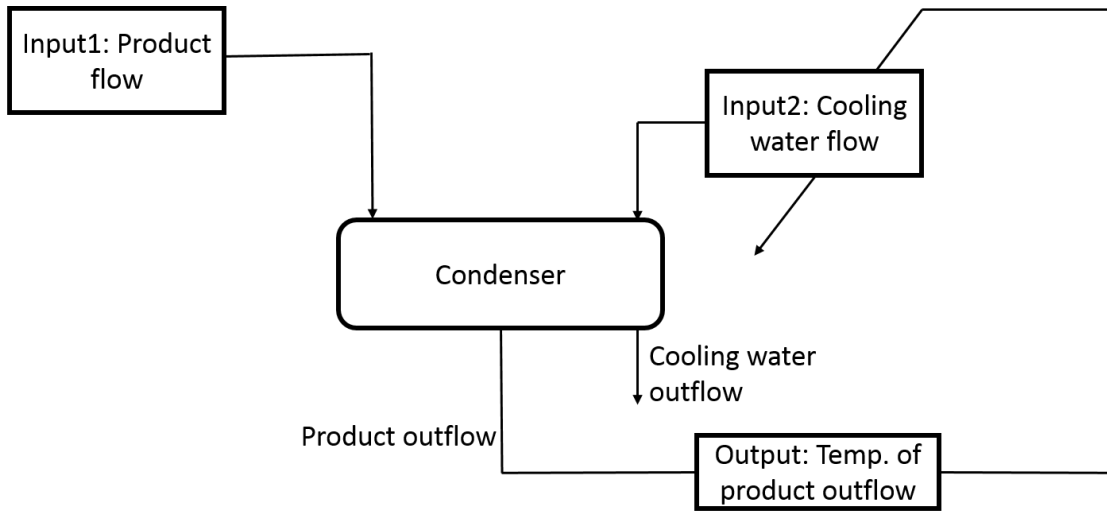


Figure 1(b). The unit operation of the product condenser in Tennessee Eastman process. (Adapted from [13])

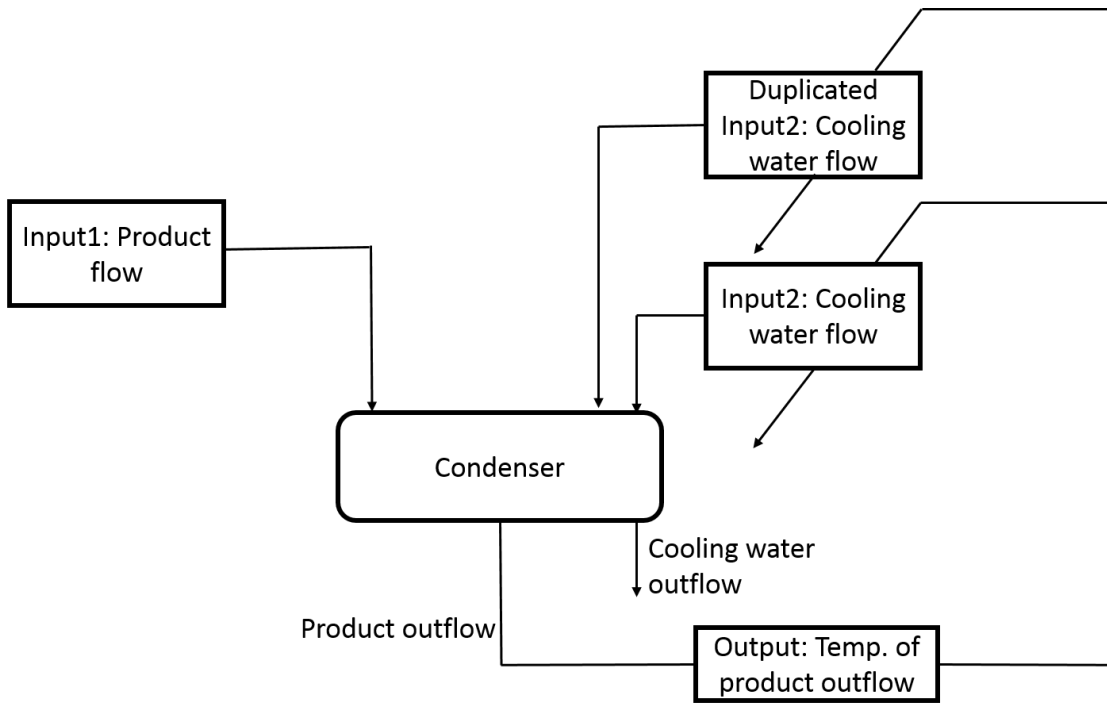


Figure 1(c). Implementation of hardware redundancy in the unit operation of the product condenser in Tennessee Eastman process. (Adapted from [13])

The second scheme is plausibility tests which are to check some simple physical laws such as the relations between mass, velocity, and acceleration in an automotive system. This scheme is based on the assumption that a fault will lead some components violating the underlying physical laws. The drawbacks of this scheme are that its efficiency is limited in a complex process [1-4]. The scheme of plausibility testing is shown in Fig. 2.

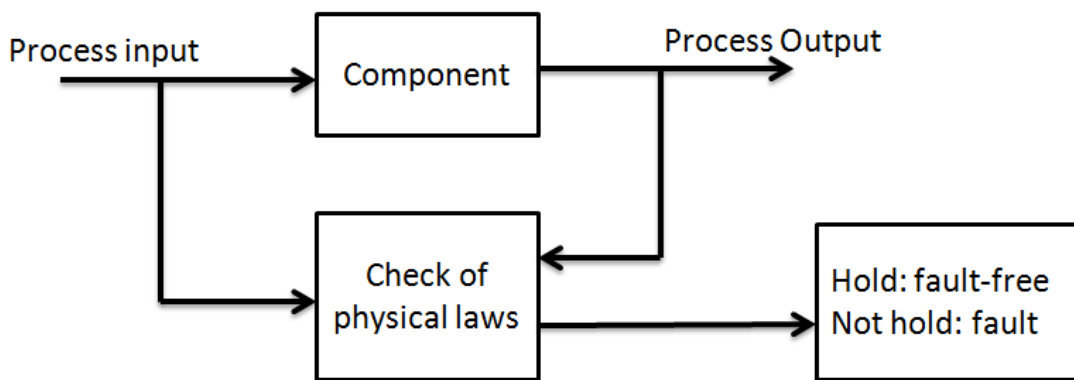


Figure 2. The scheme of plausibility testing. (Adapted from [4])

Signal processing based fault diagnosis is based on the assumption that the information of faults will be carried by some process outputs and presented in the form of symptoms. This scheme is to detect faults by checking those symptoms via time domain functions such as magnitude, mean values, and trends, or frequency domain functions like spectral power densities [4]. The efficiency of the signal processing based schemes is considerably limited for the processes with a wide range of operations due to the possible variation of input signals. The signal processing based scheme is shown in Fig. 3.

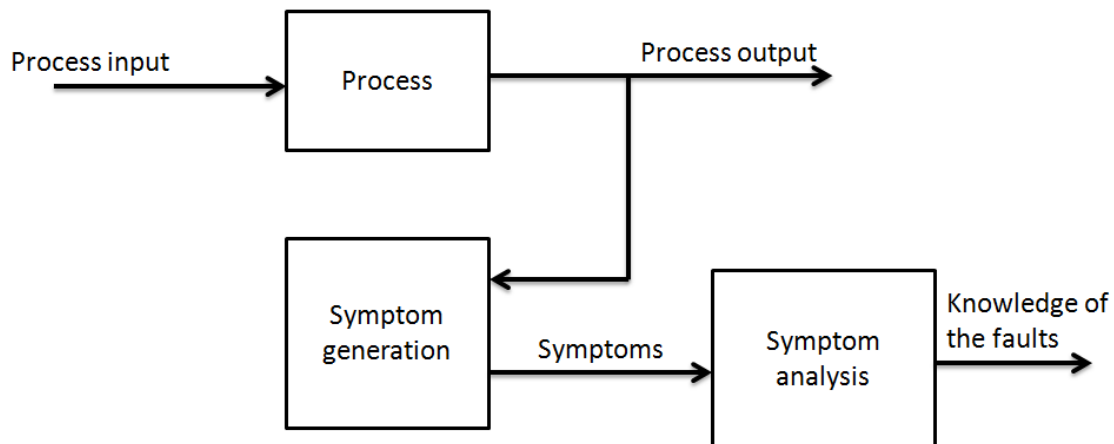


Figure 3. The signal processing based fault diagnosis scheme. (Adapted from [4])

The last type of FDI is software/analytical redundancy based fault diagnosis. The structure of this scheme is to replace the hardware redundancy component in Fig. 1(a) by a process model which is implemented in a computer. The structure of this scheme is shown in Fig. 4. The process model is a quantitative or qualitative description of the process dynamics of the respective component [1-4]. Venkatasubramanian et al. further classify those approaches which construct the process model with a priori knowledge into model-based methods and process history based (data-driven) methods, which are referred in the absence of a priori knowledge of the process [1-3]. After obtaining the process model, residual signals are generated online, and the knowledge of faults is obtained by analyzing the residual signals.

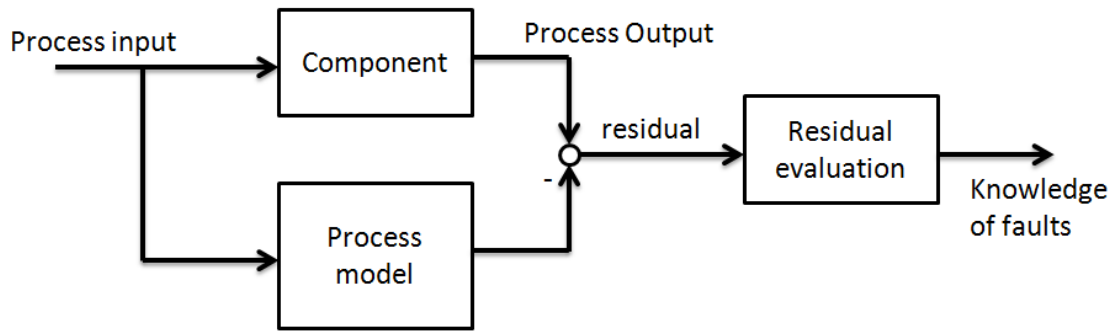


Figure 4. The software/analytical redundancy based fault diagnosis scheme. (Adapted from [4])

In the field of fault diagnosis, sensor faults, actuator faults, and system parameter faults are the three main direction for developing fault diagnosis systems [4]. This dissertation is focused in the direction of sensor faults and aimed to develop a novel FDI system to overcome nonlinear limitations of current methods. The rest of this dissertation is organized as follows. The rest of Section 1 is aimed to illustrate the problem statement of sensor fault detection and isolation. A comprehensive literature review is given in the Section 2, and the proposed approach is presented in the Section 3. In the Section 4, the simulation results are shown, and the experimental results are shown in the Section 5. In the end, conclusions are given in the Section 6.

## 1.2 Problem Statement

To further understand the problem of sensor fault detection and isolation, the effect of sensor faults and different fault types will be illustrated in this subsection.



### **1.2.1 Effect of Sensor Faults**

Consider the Continuous Stirred Tank Heater (CSTH) proposed by N. F. Thornhill et al. [14]. The configuration of the CSTH is redrawn and revised with two sensors, one actuator, and one controller in order to illustrate the following concepts shown in Fig. 5. The pilot plant is a stirred tank experimental rig to mix hot and cold water, and then heat the mixed water using steam through a couple of heating coils and drain the mixed water from the tank through a long pipe. The temperature in the tank is assumed to be the same as the outflow temperature, assuming a well mixed situation. There are three manipulating variables used to actuate the CSTH: 1. hot water flow; 2. hot water temperature (actuator); 3. cold water temperature. In addition, four variables are involved in the CSTH: 1. outflow water temperature (sensor 1); 2. cold water flow (sensor 2); 3. tank level; 4. the heat released by heater. The controller is added in order to control the outflow water temperature to be a target temperature.



*loop* sensor fault. In contrast, a Sensor 2 fault will not be feedback to the CSTH, so this situation is considered as an *open-loop* sensor fault.

In short, an open-loop sensor fault will only lead to a single abnormal event in one sensor. On the other hand, since a closed-loop sensor fault may mislead the applied control strategies to generate inappropriate control inputs, the applied system will be deviated from its desired operation, which will lead to multiple abnormal events occurring in a set of sensors. Furthermore, multiple abnormal events occurring in a set of sensors may also appear in the cases of multiple closed-loop sensor faults and multiple open-loop sensor faults. Therefore, in this dissertation, we define the cases of single closed-loop sensor fault, multiple closed-loop sensor faults, and multiple open-loop sensor fault to be the multiple sensor faults situation and the case of single open-loop sensor fault to be the single sensor fault situation. A sensor fault detection and isolation (SFDI) system is aimed to timely detect single or multiple abnormal events occurring in sensors and locate their locations in order to avoid catastrophic tragedies and reduce monetary losses [1].

### **1.2.2 Additive Faults and Multiplicative Faults**

Classified by the way faults affect the system dynamics except the effect of feedback, there are two types of faults: additive faults and multiplicative faults [4]. Consider the linear system in Eq. (1),

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx. \end{aligned} \tag{1}$$

Here,  $y$  is the system output;  $u$  is the system input and  $(A, B, C)$  are the system parameter matrices. The occurrence of an additive fault will not affect the system dynamics in the open-loop cases. The representation of an additive fault,  $f_s$ , is shown in Eq. (2),

$$y = Cx + f_s. \quad (2)$$

In contrast, a multiplicative sensor fault is caused by malfunctions in the process or in the sensors. This kind of faults will lead to changes in the system dynamics. The representation of a multiplicative fault,  $\Delta A$ , is shown in Eq. (3),

$$\begin{aligned} A' &= A + \Delta A \\ \dot{x} &= A'x + Bu \\ y &= Cx. \end{aligned} \quad (3)$$

In general, sensor faults and actuator faults are modeled as additive faults, and system parameter faults are modeled as multiplicative faults [4]. Therefore, in our simulation, all sensor faults are simulated to be additive.

## 2. LITERATURE REVIEW

Depending on methods, most FDI systems are classified into two categories: model-based and process history based [1-3]. Generally speaking, model-based methods are more systematic and efficient, but the main limitation is they require prior knowledge of the system. In the other words, an accurate model of the system is needed. Most model-based methods are based on analytical redundancy which is the difference between measured process variables and their estimates [4, 8, 15]. Usually, analytical redundancy is processed by a residual generator to enhance the effect of a sensor fault such that it can be recognized [15]. On the other hand, process history based methods do not require prior knowledge of the system. Instead, these methods require large amount of historical process data which include whole system behaviors.

### **2.1 Model-based Methods**

In 1984, Chow and Willsky introduced the concept of parity relations [15]. Based on parity relations, Chow and Willsky further developed the scheme of residual generation by parity equations [15]. Parity equations have been widely applied to fault diagnosis. The following equations, Eq. (4-10), are quoted from the original paper wrote by Chow and Willskys' [15] in order to explain the concept of parity equations. Consider the following deterministic model:

$$x(k+1) = Ax(k) + \sum_{j=1}^q b_j u_j(k)$$

$$y_j(k) = c_j x(k), j = 1, \dots, M. \quad (4)$$

Where  $x$  is the  $N$  dimensional state vector,  $A$  is a constant  $N \times N$  matrix,  $b_j$  is a constant column  $N$ -vector, and  $c_j$  is a constant row  $N$ -vector. Then, define

$$C_j(r) = \begin{bmatrix} c_j \\ c_j A \\ \vdots \\ c_j A^r \end{bmatrix}, r = 0, 1, \dots; j = 1, \dots, M. \quad (5)$$

According to the Cayley-Hamilton Theorem, there is an integer  $1 \leq n_j \leq N$  such that

$$\text{rank}(C_j(r)) = \begin{cases} n_j, & \text{if } r \geq n_j \\ r + 1, & \text{if } r < n_j \end{cases} \quad (6)$$

Also, define

$$Y_j(k) = \begin{bmatrix} y_j(k) \\ \vdots \\ y_j(k + n_j) \end{bmatrix}, j = 1, \dots, M. \quad (7)$$

Consider a nonzero row vector,  $w$ , of dimension,  $n = \sum_{j=1}^M (n_j + 1)$ , assume  $w$  exists and satisfies

$$[w^1, \dots, w^M] \begin{bmatrix} C_1(n_1) \\ \vdots \\ C_M(n_M) \end{bmatrix} x(k) = 0, \quad (8)$$

where  $w^j, j = 1, \dots, M$ , is a  $(n_j - 1)$ -dimensional row vector. Assume the system in Eq. (4) is observable, there are only  $n - N$  linearly independent  $w$ 's satisfying Eq. (8). Let  $\Omega$  be an  $(n - N) \times n$  matrix composed by those linearly independent  $w$ 's as its rows. Define

$$T = \begin{bmatrix} C_1(n_1) \\ \vdots \\ C_M(n_M) \end{bmatrix}, \quad (9)$$

the rows of  $\Omega$  span the orthogonal complement of the range space of  $T$ . Then, we can get the generalized parity vector

$$P(k) = \Omega \left\{ \begin{bmatrix} Y_1(k, n_1) \\ \vdots \\ Y_M(k, n_M) \end{bmatrix} - \begin{bmatrix} B_1(n_1) \\ \vdots \\ B_M(n_M) \end{bmatrix} U(k, n_0) \right\}, \quad (10)$$

where

$$B_j(n_j) = \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ c_j B & 0 & \ddots & & & \vdots \\ \vdots & c_j B & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ c_j A^{n_j-1} B & c_j A^{n_j-2} B & \cdots & c_j B & \cdots & 0 \end{bmatrix}$$

$$B = [b_1, b_2, \dots, b_q]$$

$$u(k) = [u_1(k), \dots, u_q(k)]'$$

$$n_0 = \max(n_1, \dots, n_M)$$

$$U(k, n_0) = [u'(k), \dots, u'(k + n_0)]'$$

In Eq. (10),  $P(k)$  is called the *generalized parity vector* [15], and it is nonzero when a sensor fault occurs. The  $(n - N)$  dimensional space of all parity vectors is called the *generalized parity space*<sup>11</sup> [15]. Any linear combination of the rows of the generalized parity vector is called a *parity relation* or a *parity equation*<sup>11</sup> [15], and any linear combination of the right hand side of Eq. (10) is called a *parity function*<sup>11</sup> [15]. The above concept of parity equations, developed by Chow and Willsky [15], contributes to nearly all FDI related researches.

In Chow and Willskys' subsequent works, they consider the robustness of designing residual generators. They handle the problem of robustness in two ways: 1. estimate the effect of uncertainties including noise disturbance and system parameter uncertainty, and then compensate the FDI system; 2. minimize the sensitivity of a FDI system to those uncertainties [15]. In the end, the problem of robustness is formulated as an optimization problem [15]. However, Chow and Willsky mainly focus on the detection part, and their method does not deal with the isolation issue.



Further research has been conducted by Gertler. To address the issue of isolation, he proposes several ways to design the structure of residuals in order to address their isolability. Gertler introduce three schemes for designing the structure of residuals: 1. diagonal; 2. directional; 3. structured [12, 16]. These schemes are briefly described following. Diagonal scheme is that each residual vector is corresponded to one and only one sensor fault shown as Eq. (11).

$$Z_F(q) = \text{Diag}[\sigma_1(q) \quad \cdots \quad \sigma_k(q)] \quad (11)$$

Where,  $\sigma_j(q)$ ,  $j = 1, \dots, k$ , are response dynamics in the shift operator  $q$ , and  $Z_F(q)$  is the diagonal residuals in the shift operator  $q$ . The diagonal scheme is ideal for isolating multiple sensor faults. However, this scheme is limited on the number of faults which can be handled in implementation [12, 16]. The second scheme is the directional scheme which is to restrict the response of a particular sensor fault to a straight line in the residual space at all times [12, 16]. The  $j$ -th directional residual is shown as Eq. (12).

$$r(t|p_{Fj}) = \psi_j \sigma_j(q) p_{Fj}(t) \quad (12)$$

Where,  $p_{Fj}(t)$  is the  $j$ -th additive unknown input at time  $t$ , and  $\psi_j$  is the  $j$ -th response direction. This scheme is capable of isolating a single fault. However, multiple faults can be isolated only when the residual directions are independent [12, 16]. Finally, the structured scheme is to design each residual element responds only to a subset of faults

[12, 16]. The following two examples are used to explain the structured scheme shown in Eq. (13).

$$\begin{array}{ccccccc}
 & f_1 & f_2 & f_3 & & f_1 & f_2 & f_3 \\
 r_1 & I & I & 0 & r_1 & I & I & 0 \\
 r_2 & I & 0 & I & r_2 & I & 0 & I \\
 r_3 & I & I & 0 & r_3 & 0 & I & 0
 \end{array} \tag{13}$$

Where,  $r_j, j = 1, 2, 3$ , is the  $j$ -th residual element, and  $f_j$  is the  $j$ -th sensor fault. The right example is a “strongly isolating” structure which is no response pattern can be obtained from other pattern by replacing “ $I$ ”s into “ $0$ ”s [12, 16]. Otherwise, it is a “weakly isolating” shown as the left example [12, 16]. Under the assumption that system parameters are known, Gertler transfers the residuals generated by Eq. (10) to his proposed structure, and then isolates faults by different residual patterns.

To enhance the isolability of Gertler’s method, Koscielny et al. proposed an efficient search for large scale systems [17]. This search method is an extension of the “weakly isolating” structure proposed by Gertler. Therefore, the uncertainties of residual elements are not considered [17]. They assume the set of residual elements<sup>4</sup> from different faults are known, and the large scale system can be separated into several subsystems. By reducing the residual elements of uncorrelated subsystems, their method greatly improve the search efficiency in the part of isolation [17].

---

<sup>4</sup> In fact, Kocielny uses “symptom” instead of “residual element” used in Gertler’s paper and book.

On the other hand, Ding provides a numerical way to design the residual generator in Eq.(9) [4]. Ding's method is based on the Luenberger type observer [4, 18]. Ding converts the requirement for the residual generator,  $w$ , in Eq. (8) to the Luenberger conditions, and provides a numerical solution for the Luenberger conditions and then for the residual generator. Ding's method will be briefly described in subsequent content. Consider the system in Eq. (1), the Luenberger type observer is described by

$$\dot{z} = Gz + Hu + Ly; \hat{y} = \bar{W}z + \bar{V}y + \bar{Q}u, \quad (14)$$

where  $z \in R^s$ ,  $s$  is the observer order. Assume  $G_{yu}(p) = C(pI - A)^{-1}B + D$ , matrices  $G, H, L, \bar{Q}, \bar{V}, \bar{W}$ , and a matrix  $T \in R^{s \times n}$  have to fulfill the Luenberger conditions: 1.  $G$  is stable; 2.  $TA - GT = LC$  and  $H = TB - LD$ ; 3.  $C = \bar{W}T + \bar{V}C$  and  $\bar{Q} = -\bar{V}D + DE$ . In this case,  $D$  equals to zero. Define  $e = Tx - z$ , and then it turns out the Eq. (15),

$$\dot{e} = Ge, y - \hat{y} = \bar{W}e. \quad (15)$$

Based on Eq. (15), define a residual vector shown as Eq. (16),

$$r = V^*(y - \hat{y}), V^* \neq 0. \quad (16)$$

Where,  $V^*$  is a residual generator. Ding proposes a numerical approach to solve for the residual generator,  $V^*$ , based on Luenberger conditions. The complete proof and details are referred to the chapter 5 in Ding's book [4].

The major limitation of above model based methods is that they all require the priori knowledge of system parameters. However, the system parameters are usually unknown and hard to be obtained in practice. A famous case is the Tennessee Eastman process [13]. The Tennessee Eastman process is a nonlinear chemical process with 41 measurements and 12 manipulating indices. Because of its complicated nonlinear property, model-based methods are not suitable to in this kind of process. On the other hand, process history based methods are built in a statistical way. Different from model-based methods, process history methods do not require prior knowledge of system parameters. Next section is a review for process history based methods.

## **2.2 Process History-based (Data-driven) Methods**

Process history based methods have been developed in various tracks: system identification, dimension reduction and its nonlinear extensions, and expert systems [3].

### **2.2.1 System Identification**

This track is focused on removing the major limitation of model-based methods. Ding and Qin propose methods to acquire system parameters from historical data by using system identification techniques such as prediction error methods (PEM) and subspace identification methods (SID) [19-21]. However, system identification comes with the

problem of modeling errors which will render the residual generator sensitive, and there is no related analysis in literatures [21]. To deal with this problem, Dong provides a strategy to enhance the robustness of identified parameter under the assumption of only the occurrence of additive faults [21]. Dong analyzes the effect of parameters identification errors and establishes the error bound in neglecting the bias term due to the initial states. In Dong's simulation, he applies his method on a linearized vertical take-off and landing (VTOL) and turns out good results [21]. However, the VTOL system is a linearized fourth order system, and the performance of Dong's method is undetermined for a large scale system with nonlinear properties such as the Tennessee Eastman process. In addition, another limitation of these methods is that they are application dependent which means there is no unified procedure for different application [3].

### 2.2.2 Dimension Reduction

Another track is via dimension reduction techniques to generate a residual generator from the left null space of the matrix  $T$  in Eq. (9). This is so called parity space analysis. In this track, historical process data are assumed to be stationary in the fault free case [22]. To explain the basic idea of applying dimension reduction techniques for fault diagnosis, consider the deterministic system in Eq. (1). Assume a sensor fault occurs such that

$$y = Cx + e_s, \tag{17}$$

where  $e_s$  is a bias due to the sensor fault. Define the residual signal,  $r$ , as

$$r = we_s = w(y - Cx), \quad (18)$$

where  $r$  is a residual generator. The Eq. (18) is a linear combination of Eq. (10), so it is a parity equation for the deterministic system in Eq. (1). Therefore, the problem of fault diagnosis becomes how to find a solution for  $w$  satisfying Eq. (8). To solve this problem by dimension reduction, consider  $y$  as a vector of sensor measurements in the original coordinates system, and denote those measurements in a new coordinates system by  $t$ , then:

$$t = Uy; \text{cov}(t_i, y_j) = 0; \forall i \neq j, \quad (19)$$

where  $U = [u_1 \ u_2 \ \dots \ u_n]^T$  is the set of unit vectors of the new coordinates system. By using dimension reduction techniques,  $t$  and  $U$  can be separated into two parts depending on the variance of those measurements [22]. In Eq. (20),  $x_y$  denotes the measurements with high variance, and  $v_y$  denotes low variance measurements. Similarly,  $P$  and  $Q$  denote the corresponding set of unit vectors to  $x_y$  and  $v_y$  respectively.

$$\begin{aligned} t &= \begin{bmatrix} x_y \\ v_y \end{bmatrix} \\ U &= \begin{bmatrix} P \\ Q \end{bmatrix} \end{aligned} \quad (20)$$

Where  $P = [u_1 \ u_2 \ \cdots \ u_k]^T$ ,  $Q = [u_{k+1} \ u_{k+2} \ \cdots \ u_n]^T$ ,  $u_i$ ,  $i = 1, \dots, n$  is the eigenvector of the covariance matrix of  $y_{stand}$  corresponding to the  $i$ -th largest eigenvalue, and  $y_{stand}$  is obtained by standardize  $y$  with its mean value and standard deviation [23]. In addition, when  $y$  is well approximated by  $x_y$  in a lower dimension,

$$v_y = Qy = QCx \approx \vec{0}. \quad (21)$$

Comparing Eq. (21) with Eq. (8), the condition in Eq. (8) is satisfied by  $Q$ . Hence,  $Q$  can be considered as an approximate solution of the residual generator,  $w$ , in Eq. (18). Therefore, when the applied system is faulty, the residual signal,  $r$ , in Eq. (18) will be nonzero. Otherwise,  $r$  will be zero.

There are various dimension reduction techniques. Basically, these techniques can be classified according to their function on linear data and nonlinear data [23]. In the category of linear dimension reduction, principal component analysis (PCA) is a well known method. PCA is the best linear dimension reduction technique in the mean-squared error sense [23-25]. In essence, PCA is used to find few orthogonal linear combinations or principal components (PCs) with largest variance which can approximate the original data. There are several methods that are known to be the same sense as PCA such as the singular value decomposition (SVD), the Karhunen-Loeve transform, the Hotelling transform, and the empirical orthogonal function (EOF) method [23].

However, PCA is not perfect. One of the major problems is that PCA is time invariant, but most of industrial processes is time varying [3]. Li et al.. proposed recursive PCA

(RPCA) in his work [26]. Li et al. concerns the following key points in his method: 1. recursive update of the mean value; 2. efficient sample-wise calculations for updating; 3. recursive determination of the sufficient number of principal components for representing the original data [26]. To briefly describe Li's method, consider the raw initial data block  $X_1^0 \in R^{n_1 \times m}$  of  $n_1$  samples and  $m$  variables. Then the mean of each variable,  $b_1$ , is

$$b_1 = 1/n_1 (X_1^0)^T \mathbf{1}_{n_1}, \quad (22)$$

where  $\mathbf{1}_{n_1} = [1, 1, \dots, 1]^T \in R^{n_1}$ . In addition, the data is normalized by Eq. (23) to zero mean and unit variance before processing by PCA.

$$X_1 = (X_1^0 - \mathbf{1}_{n_1} b_1^T) \Sigma_1^{-1} \quad (23)$$

Where,  $\Sigma_1 = \text{diag}(\sigma_{11}, \dots, \sigma_{1m})$  composed by the standard deviation of each variable. The correlation matrix of the variables,  $R_1$ , can be approximated by

$$R_1 = 1/n_1 - 1 X_1^T X_1. \quad (24)$$

Now, assume  $b_k$ ,  $X_k$ , and  $R_k$  are calculated when the  $k$ -th block has been collected. The recursive calculation of the covariance matrix can be obtained by



$$R_{k+1} = \frac{1}{N_{k+1} - 1} X_{k+1}^T X_{k+1} - \frac{N_k - 1}{N_{k+1} - 1} \Sigma_{k+1}^{-1} \Sigma_k R_k \Sigma_k \Sigma_{k+1}^{-1} + \frac{N_k}{N_{k+1} - 1} \Sigma_{k+1}^{-1} \Delta b_{k+1} \Delta b_{k+1}^T \Sigma_{k+1}^{-1} + \frac{1}{N_{k+1} - 1} X_{n_{k+1}}^T X_{n_{k+1}}. \quad (25)$$

Where,  $N_k = \sum_{i=1}^k n_i$  and  $\Delta b_{k+1} = b_{k+1} - b_k$  [26]. In addition to updating the covariance matrix, next step is to determine recursively the number of principal components [26]. Li considers several methods such as cumulative percent variance (CPV), average eigenvalue (AE), and imbedded error function (IEF) in this step [26, 27]. Here, we take the CPV method as an example. CPV is to measure the variance captured by the first  $l_k$  principal components associated to the  $l_k$  largest eigenvalues of the covariance matrix [26, 27]. CPV can be presented in the following form.

$$CPV(l_k) = \frac{\sum_{j=1}^{l_k} \lambda_j}{\sum_{i=1}^m \lambda_i} 100\% \quad (26)$$

Where,  $\lambda_i$  is the  $j$ -th large eigenvalues of the covariance matrix. By setting a predetermined limit, one can determine the needed number of principal components [26]. In the end, Li applies the recursive PCA to detect faults on a semiconductor process with slow process changes and turns out a significant improvement in reducing false alarms [26].

Although Li's research positively improves the performance of PCA, the issue that most methods lack of isolability still remains. In Sharifi and Langari's work, they introduce the concept of fault image vector in the residual space. They use PCA to

approximate the original sensor measurements and generate residual signals by Eq. (21) [22]. Then, they extend Gertler's directional residual structure to define the sensor fault index (SFI) which can be used to indicate the occurrence of a single fault [22]. The sensor fault index is briefly illustrated as following content. Consider Eq. (21), assume  $y \in R^n$  is a column vector composed by  $n$  healthy outputs, and there is a sensor fault,  $\delta_j$ , occurs in the  $j$ -th sensor. Therefore, the sensor measurements,  $y^*$ , becomes

$$y^* = y + \Delta, \quad (27)$$

where  $\Delta = [0, 0, \dots, \delta_j, \dots, 0]^T$ ,  $\Delta \in R^n$ . Moreover, consider  $Q$  is in the form of column vectors,  $Q = [q_1, q_2, \dots, q_n]$ , and substitute Eq. (27) into Eq. (21). Then,

$$v_{y^*} = Qy^* = Qy + Q\Delta \approx q_j\delta_j, \quad (28)$$

where  $q_j$  is called the fault image vector for the  $j$ -th sensor [22]. Let

$$n_i = q_i / \|q_i\|, i = 1, \dots, n, \quad (29)$$

and

$$n_{v_{y^*}} = v_{y^*} / \|v_{y^*}\|. \quad (30)$$

The sensor fault index for the  $j$ -th sensor is defined as

$$f_i = n_{v_{y^*}} \cdot n_j, \quad (31)$$

and the dot sign is inner product of those two vectors [22]. In absence of noise, disturbance, and sensor uncertainties, if there is a single fault which occurs in the  $j$ -th sensor,  $|f_j| \approx 1$ , and  $|f_i| \approx 0$  for  $i \neq j$ . Therefore, the  $j$ -th sensor fault is isolated by the sensor fault index.

The disadvantage of Sharif's method is the limited number of faults. The SFI only can indicate a single fault, and it will lose its performance in the situation of multiple faults. In Sharifi and Langari's following work, they propose a probabilistic process for detecting faults [28]. They propose an index to determine non-detectable sensors in a probabilistic way based on the concept of the sensor fault index, and then apply to a subpart of the Tennessee Eastman process. Their results show the capability of this method [28]. Now, we continue from their previous work in Eq. (27-31) to explain this method. Consider the effect of noise, the vector of sensor measurements,  $y^*$ , in Eq. (27) becomes

$$y^* = y + n + \Delta, \quad (32)$$

where  $n$  is the noise. Assume the noise has a stationary Gaussian distribution, and its parameters are known such as

$$n \propto N(0, \Sigma). \quad (33)$$

In this case, the noise is zero mean, and its standard deviation is  $\Sigma$ . Assume the  $j$ -th sensor is faulty, and there is only one sensor fault. Denote the residual  $r = Qy^*$ . According to Eq. (28), one can obtain

$$r \propto N(\delta_j q_j, \Sigma_r), \quad (34)$$

where  $\Sigma_r = Q\Sigma Q^T$ . Based on Eq. (34), one can obtain the following conditional distributions.

$$\begin{aligned} p(r|H) &= N(0, \Sigma_r) \\ p(r|S_j, \delta_j) &= N(\delta_j q_j, \Sigma_r) \end{aligned} \quad (35)$$

Where  $p(r|H)$  is the probability distribution of  $r$  in fault-free case, and  $p(r|S_j, \delta_j)$  is the probability distribution of  $r$  when there is a sensor fault in the  $j$ -th sensor with magnitude  $\delta_j$ . Assume  $\delta_j$  is uniform distributed, and marginalize  $p(r|S_j, \delta_j)$  by integrating over  $\delta_j$ ,

$$p(r|S_j) = \int p(r|S_j, \delta_j)p(\delta_j)d\delta_j = N(0, \Sigma^j). \quad (36)$$

Where,  $\Sigma^j = Q_j \Sigma_r Q_j^T$ , and  $Q_j$  is composed by the orthonormal vectors of  $q_j$  [28].

According to Bayes' theorem,

$$\begin{aligned} p(S_j|r) &= \frac{p(r|S_j)p(S_j)}{p(r)}, \\ p(H|r) &= \frac{p(r|H)p(H)}{p(r)}, \end{aligned} \quad (37)$$

where  $p(r) = p(r|H)p(H) + \sum_{i=1}^n p(r|S_i)p(S_i)$ . In addition,  $p(H)$  and  $p(S_j)$  are estimated by

$$\begin{aligned} p(H) &= p(H|r^{<old>}) \\ p(S_j) &= p(S_j|r^{<old>}), \end{aligned} \quad (38)$$

where  $r^{<old>}$  is the residual value calculated from previous measurements [28]. The level of fault detectability of sensors can be estimated by the probability density [28]. Therefore, the detection index for a sensor fault in the  $j$ -th sensor with unit magnitude is defined as

$$\theta_j = \frac{1}{1 + p(r = q_j|H)}. \quad (39)$$

Where,  $0 < \theta_j < 1$ , and the smaller  $\theta_j$  means the smaller probability to detect the  $j$ -th sensor fault [28].

Sharifi's work is focused on the detection part of a FDI system, and this new method can be applied to multiple faults situation. For the isolation part, this work has the same limitation of the number of faults. The other limitation comes with PCA. Since PCA is to search linear combination of principal components, modeling errors will be large when we apply PCA on a complex nonlinear system.

### **2.2.3 Nonlinear Extensions of Dimension Reduction Based Methods**

#### **2.2.3.1 Neural Networks**

To overcome the limitation of PCA, several nonlinear dimension reduction methods have been developed. Kramer proposes a method to do nonlinear PCA (NLPCA) based on auto-associative neural network (AANN) [29]. AANN is a neural network structure previously proposed by Kramer [30]. The structure of AANN is shown in Fig. 6.

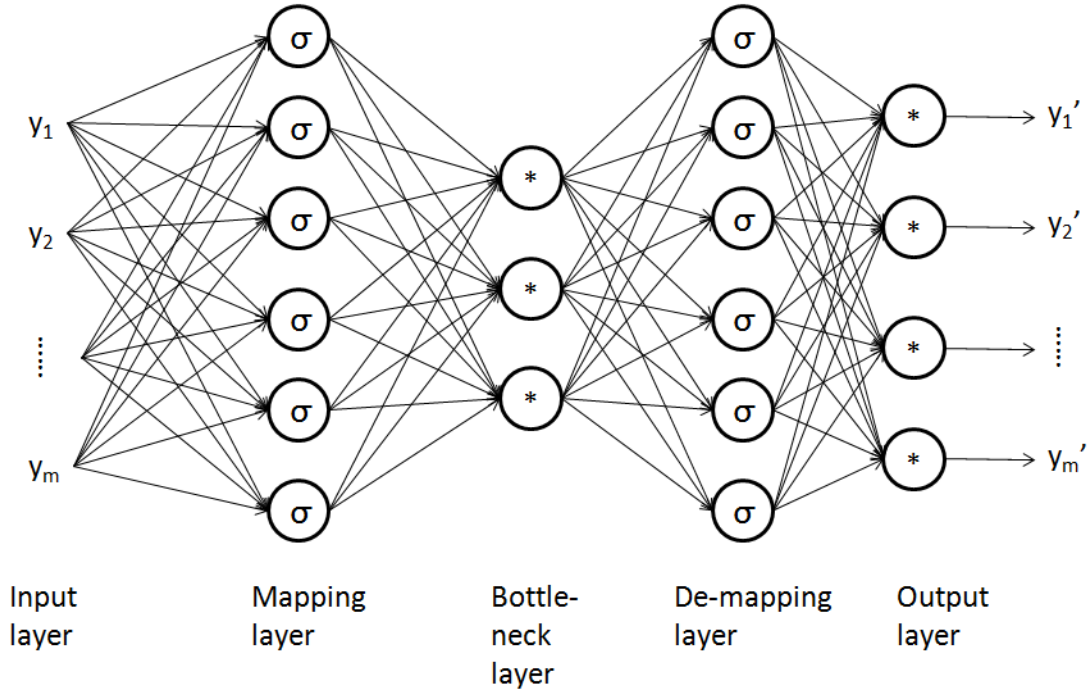


Figure 6. The architecture of autoassociative neural network. ( $\sigma$  indicates sigmoidal nodes, \* indicates sigmoidal or linear nodes). (Adapted from [30])

The key feature of an AANN is the bottleneck layer with fewer nodes between the input and output layers. This feature will force the input data to be approximated in a lower dimension [29]. Therefore, AANN structure has the property of noise filtering and robustness depending on different training strategies [29]. Kramer used the AANN feature of approximating the data in lower dimensions to do NLPCA [29, 31]. He trained the AANN by the backpropagation approach with the cost function in Eq. (40).

$$E = \sum_{p=1}^n \sum_{i=1}^m (Y_i - Y_i')^2, \quad (40)$$

where  $n$  is the number of measurements, and  $m$  is the number of input/output pairs. In addition, Kramer uses a basis function, shown in Eq. (41), whose capability of fitting any nonlinear function has been proved elsewhere [32].

$$v_k = \sum_{j=1}^{N_2} w_{jk2} \sigma(\sum_{i=1}^{N_1} w_{ij1} u_1 + \theta_{ji})$$

$$\sigma(x) = 1 / (1 + e^{-x}) \quad (41)$$

Eq. (41) is the description for a single hidden layer neural network (ANN) with  $N_1$  inputs, a hidden layer composed of  $N_2$  sigmoidal nodes, and a linear output node for each  $k$  [29]. In Eq. (40),  $w_{ijk}$  is the weight connecting from node  $i$  in layer  $k$  to node  $j$  in layer  $k + 1$ , and the  $\theta$  are adjustable nodal biases similar to weights [29].

Najafi developed an enhanced AANN (E-AANN) for diagnosing sensor faults [33]. E-AANN is based on the property of AANN to produce identical output signals as input signals. When faults occur, the outputs of AANN will not be exactly the same as its inputs. Further, Najafi assumes that the maximum number of faults occurrence at one time is one [33]. So he can identify a single fault by adjusting each input of AANN by a small stepsize from a predefined minimum to its maximum until the outputs of AANN are the same as its inputs [33]. The advantage of this method is that it is able to not only isolate a single fault, but also to identify the magnitude of the single fault [33]. Nevertheless, its isolability is limited to the number of occurrence of faults.

Hines et al. use AANN in a different way in their work [34, 35]. In their work, they take advantage of the AANN property of robustness. By a robust training procedure,



AANN is forced to rely on the inherent information in the signals corresponded to a specific sensor to estimate the specific sensor measurement [34]. In other words, Hines uses AANN as a filter which can filter noise/disturbance and sensor faults. AANN will produce uncorrupted and fault-free output signals if the magnitude of faults, noise and disturbance are in the acceptable range of the filtering ability of AANN. By subtracting the uncorrupted and fault-free outputs from faulty inputs to generate the residual signals, and then one can detect and isolate multiple faults [34, 35]. The configuration of Hines' method is shown in Fig. 7 [35].

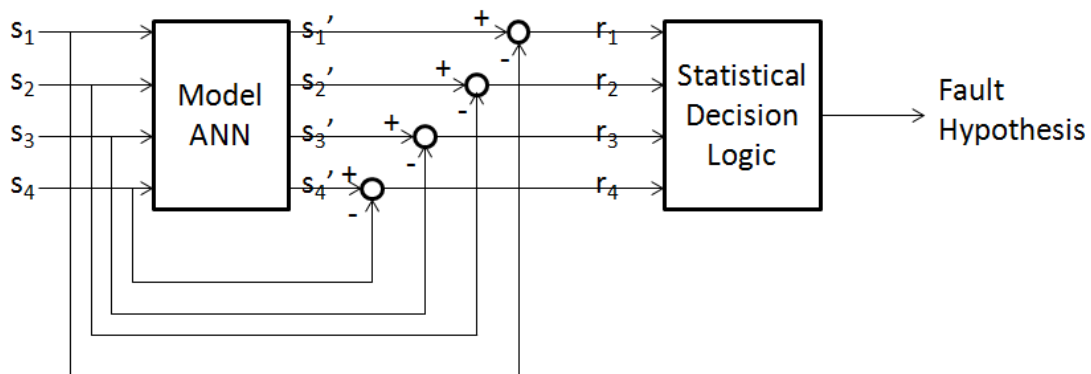


Figure 7. Sensor monitoring module in Hines' method. (Adapted from [34])

Hines adopts three strategies to increase the robustness of AANN. First, he sets a network training stopping criterion to avoid overfitting the training set [34, 35]. Second, he corrupts the input data of the training set instead of using the same data as target outputs [34, 35]. To detail the second strategy, assume the original training set is composed by

$[X, Y]$ , where  $Y$  is the target outputs and  $X = Y$  in this case. The original training inputs,  $X$ , is augmented by

$$X_{aug} = X + \delta_j I_j, j = 1, \dots, m, \quad (42)$$

where  $I_j$  is the  $j$ -th column of the identity matrix,  $m$  is the number of inputs, and  $\delta_j$  is randomly chosen from  $\pm 10$  percent of the  $j$ -th output [35]. Each sensor is corrupted several times by different  $\delta$  [35]. Third, he uses SVD method to solve the weights between the de-mapping layer and the linear output layer [34, 35]. Since the output layer of the AANN structure is linear, training is greatly accelerated [35]. To illustrate the third strategy, assume  $W$  is the weight matrix between the de-mapping layer and the output layer in Fig. 6, and  $X_d$  is the inputs of the output layer. Therefore, the target outputs,  $Y$ , can be obtained by

$$Y = WX_d. \quad (43)$$

Then, the weights can be solved by the general least squared solution [35].

$$W = (X_d'X_d)^{-1}X_d'Y \quad (44)$$

By using SVD to solve the linear output layer weights, only the most relevant information is retained to compute the weights [34, 35]. Therefore, SVD method not only

reduces the training time but also provides better initial weights for iterative methods applied to all weights later [35]. In addition to these three strategies, Hines also uses the sensitivity analysis as a tool for network parameter selection [35]. Hines chooses the final AANN parameters from a set of network parameters which were obtained by training AANN with different initial weights [35]. Sensitivity is defined as

$$S_{input}^{output} = \frac{\partial(output)}{\partial(input)}. \quad (45)$$

According to Eq. (45), the smaller sensitivity means that the change of the output due to the change of the corresponding input is smaller [35]. In the other words, a smaller sensitivity means that the trained AANN is more robust [35].

The limitation of Hines' method is originated from the filtering ability and the approximating ability of AANN related to the number of nodes in the bottleneck layer. In his results, the range of fault detection levels is between 0.3% and 3% of the maximum values of each sensor, which is relative narrow compared to aforementioned methods. In addition, the main issue of AANN-based methods is its lack of an analytic foundation. Currently, there are several problems still can't be solved analytically, such as how many nodes are needed for hidden layers, and training methods for AANN are seeking for a local minimum instead of a global minimum solution. Moreover, AANN structures are hard to be trained in practice.

### 2.2.3.2 Mixture PCA

In addition to neural networks, Choi and Sharifi et al. employ the mixture of probabilistic PCA (MPPCA) for sensor fault diagnosis for nonlinear systems in their current work [36, 37]. The probabilistic PCA (PPCA) is introduced by Tipping and Bishop in 1999 [38]. By PPCA, they overcome the issue which is the conventional PCA is unable to process data in different regions simultaneously [38]. Furthermore, they extend the PPCA to a nonlinear extension, MPPCA, by separating data into several locally linear regions, and each region is defined by a PPCA model. Then, they surmount the linear limitation of the conventional PCA [39]. The subsequent content is ranged to start from PPCA to MPPCA, and then illustrate Sharifi's current work. PPCA is to rewrite the conventional PCA as a probability density model shown in Eq. (46) [36, 38].

$$y = Px + \mu + m \quad (46)$$

Where  $y \in R^n$  is the sensor measurement,  $x \in R^r$  is the latent variable with fewer dimensions ( $r < n$ ),  $\mu$  is constant,  $m$  is the process noise, and  $P$  is the projection matrix [36, 38]. Moreover, let  $x \sim N(0, I)$ , and assume the process noise has a Gaussian distribution with zero mean and isotropic variance,  $m \sim N(0, \Phi)$ , where  $\Phi = \sigma^2 I$ , and  $\sigma$  is the standard deviation of the process noise [36, 38]. Therefore,

$$y \sim N(\mu, PP^T + \Phi). \quad (47)$$

Under these assumptions,  $\Phi$  is diagonal, so  $y$  are conditionally independent given  $x$  [36, 38]. Hence, one can obtain the probability model for PPCA by a combination of the conditional distribution [36], so

$$P(y|x) = 1/(2\pi\sigma^2)^{n/2} \exp(-\|y - Px - \mu\|^2/2\sigma^2), \quad (48)$$

and

$$p(x) = 1/(2\pi)^{r/2} \exp(-x^T x/2). \quad (49)$$

Then, integrating out the latent variable,  $x$ , in Eq. (48) to obtain the marginal distribution of  $y$  shown in Eq. (50).

$$p(y) = N(\mu, \Sigma) = 1/(2\pi\sigma^2)^{n/2} \Sigma^{1/2} \exp\left(-\frac{1}{2}(y - \mu)^T \Sigma^{-1}(y - \mu)\right), \quad (50)$$

where  $\Sigma = PP^T + \sigma^2 I$ . The corresponding log-likelihood for fitting the  $k$  set of measurements is

$$L = \sum_{i=1}^k -k/2 [n \log(2\pi) + \log|\Sigma| + \text{tr}(\Sigma^{-1}S)], \quad (51)$$

where

$$S = 1/k \sum_{i=1}^k (y_i - \mu)(y_i - \mu)^T, \quad (52)$$

and  $\mu$  is set to be the sample mean [36, 38]. In this case, there is an analytical solution of the  $P$  and  $\sigma^2$  for the maximum likelihood estimate shown in Eq. (53-54) [36, 38].

$$P = U_r (\Lambda_r - \sigma^2 I)^{1/2} \quad (53)$$

Where  $\Lambda_r$  is a diagonal matrix composed by the largest  $r$  eigenvalues of  $S$ , and  $U_r$  is a matrix composed by the  $r$  eigenvectors of  $S$  corresponding to those eigenvalues in  $\Lambda_r$  [38].  $\sigma^2$  is given by

$$\sigma^2 = 1/n - r \sum_{j=r+1}^n \lambda_j, \quad (54)$$

where  $\lambda_j$  is the  $j$ -th largest eigenvalue of  $S$ . In the other words, Eq. (54) can be considered as the averaged variance “lost” in projection for the lost dimensions [38].

Now, the subsequent content is to extend the PPCA to the MPPCA proposed by Tipping and Bishop [39]. As aforementioned, the MPPCA is to separate the sensor measurement space into several locally linear regions [36, 39]. Each region is defined by its own probability model obtained by the PPCA algorithm, and the whole sensor measurement space is defined by the mixture of those probability models corresponding

to each region [36, 39]. Assume there are totally  $Q$  regions, the mixture of those probability models can be written as

$$p(y) = \sum_{j=1}^Q \pi_j p(y|x_j) = \sum_{j=1}^Q \pi_j N(y|\mu_j, \Sigma_j), \quad (55)$$

where each component is a PPCA model,  $\pi_j$ ,  $x_i$ ,  $\mu_j$ , and  $\Sigma_j$  are the mixing coefficient, the latent variable, the mean value, and the covariance for the  $j$ -th probability model respectively [36, 39]. Furthermore,

$$L = \sum_{i=1}^k \ln \sum_{j=1}^Q \pi_j p(y^i|x_j), \quad (56)$$

where  $y^i$  is the  $i$ -th set of measurements [36, 39]. This problem of the maximum likelihood can be solved by the EM algorithm [36, 38, 39]. After obtaining the MPPCA, the responsibility function of  $j$ -th component in Eq. (55), with respect to the new set of measurements,  $y^{<new>}$ , is defined as

$$R_j(y^{<new>}) = p(x_j|y^{<new>}) = \frac{p(y^{<new>}|x_j)\pi_j}{p(y^{<new>})}. \quad (57)$$

In Sharifi's work, he chooses the component which has the largest responsibility for the new set of measurements, and then deal with the problem of sensor fault diagnosis in a local region corresponding to the chosen component discussed in his former work [36].

In this work, Sharifi et al. overcome the limitations of PCA, however, it comes out the issue of misclassification due to sensor faults<sup>5</sup>. When a sensor is faulty, the sensor measurement is contaminated with fault which may lead the classification techniques fail [36]. Therefore, a sensor fault might not be detected and isolated by this method when the misclassification occurs [36].

### 2.2.3.3 Manifold Methods

In order to deal with the nonlinear issue, manifold methods may be a potential solution. Manifold methods assume that the unprocessed data can be sampled from some smooth underlying manifolds in a lower dimension [40-43]. From the aspect of dimension reduction, the goal of manifold methods is to find a pair of maps  $g: R^D \rightarrow R^d$  with  $d < D$  and  $f: R^d \rightarrow R^D$  [40]. Currently, manifold methods are used in the field of image processing for the purpose of nonlinear dimension reduction such as locally linear embedding (LLE), Riemannian manifold learning (RML), and local PCA (LPCA) [40-43]. The general idea of manifold methods is to separate the data space into multiple local regions, and if these regions are small enough, they can be considered as linear and a specific linear model will be a good fit for its corresponding region [40]. Hence, the nonlinear issue can be simplified to a linear dimension reduction problem in a local region.

The LLE algorithm is introduced by Roweis and Saul. This method is based on geometric intuitions [41]. Suppose the data consist of  $N$  real-valued vectors,  $\vec{X}_i$ ,  $i =$

---

<sup>5</sup> The concept of misclassification will be clearly illustrated in the next subsection by the example of LPCA.



$1, \dots, N$ , and the dimension of the data space is  $D$ . The first step of the LLE algorithm follows from the general idea of manifold methods. Roweis et al. assume that each data point and its neighbors lie on a locally linear patch of the manifold in a lower dimension [41]. Then, linearly reconstruct each data point by its  $k$  nearest neighborhood, and the reconstruction error is measured by Eq. (58) [41].

$$\varepsilon(W) = \sum_i |\vec{X}_i - \sum_j W_{ij} \vec{X}_j|^2, \quad (58)$$

where  $W_{ij}$  is the weight which summarize the contribution of the  $j$ -th data point to the  $i$ -th reconstruction [41]. There are two constraints for solving Eq. (58): 1. enforcing  $W_{ij} = 0$ , if  $\vec{X}_j$  is not a neighbor of  $\vec{X}_i$ ; 2.  $\sum_j W_{ij} = 1$  [41]. The reason for these two constraints is to keep the weights invariant to rotations, rescaling, and translations of the data point,  $\vec{X}_i$ , and its neighbors [41]. The optimal solution for Eq. (58) has been solved by Roweis et al. [41].

After reconstructing each point, they employ the fact that the reconstruction weights reflect the intrinsic geometric properties of the data, and these geometric properties are invariant to rotations, rescaling, and translations [41]. Therefore, assume  $\vec{Y}_i$  is the representation of  $\vec{X}_i$  in a lower dimension  $d$ , and obtain the cost function based on the reconstruction error,

$$\Phi(Y) = \sum_i |\vec{Y}_i - \sum_j W_{ij} \vec{Y}_j|^2. \quad (59)$$

In Eq. (59), only  $\vec{Y}_i$  and its neighbors are unknown, because the weights are invariant in Eq. (58-59). Therefore, one can obtain the lower dimensional representation by minimizing the reconstruction error,  $\Phi(Y)$  [41]. The optimal solution has been solved by Roweis et al. too. However, from the aspect of sensor fault diagnosis, LLE is not quite suitable, because the projection matrix between  $R^D$  and  $R^d$  is not preserved. It means we have to reconstruct data and solve Eq. (58-59) every time while calculating the residuals.

Another manifold method is the Riemannian manifold learning (RML) algorithm proposed by Lin et al., which is a popular nonlinear reduction technique used in image processing currently [42, 43]. Similar to the LLE algorithm, the RML algorithm starts from reconstructing each data point by its neighbors as a simplex [42, 43]. The simplex can be considered as a linear relation between the data point and its neighbors in a local region [42, 43]. After reconstructing each data point, those simplexes are grouped, and then the shortest path between each point is obtained according to these simplexes [42, 43]. Then, the specific constraints for the image processing purpose are defined based on those shortest paths [42, 43]. In the end, each data point is projected to a locally lower dimensional coordinates, namely Riemannian normal coordinates (RNC), by the constraints defined in the last step [42, 43]. Compared with other manifold methods, the RML algorithm is capable of generating a smoother manifold, because some of the projected data are distorted in some regions due to their constraints [42, 43]. However, there are still several problems have not been well addressed in RML, such as how to choose an adequate dimension for the locally lower dimensional coordinates and how to choose an suitable number of points in each neighborhood [42, 43].

In addition to above methods, the LPCA algorithm proposed by Kambhatla et al. is a manifold method which is fast to compute [40]. In the same sense of other manifold methods, LPCA is to partition data into multiple locally linear regions [40]. Kambhatla et al. suggest two ways to separate data: the Euclidean partition and the projection partition [40]. The Euclidean partition is to classify data depending on the Euclidean distance to the center point of each local region [40]. On the other hand, the projection partition is to classify data based on their contribution of the reconstruction error for each local region [40]. These two partitions are briefly described below.

### ***Euclidean Partition***

The Euclidean partition is an easier way to construct the partition. In fact, this method is so called the generalized Lloyd algorithm (Gersho & Gray, 1992) [40]. Assume there are  $K$  reference points  $p^{(i)}$ ,  $i = 1, \dots, K$  and corresponding regions  $P^{(i)}$  in the measurement space  $y \in R^n$ . The Euclidean partition satisfies the Lloyd's optimality conditions [40].

1. Define  $d_E(a, b)$  is the Euclidean distance between  $a$  and  $b$ , and then

$$p^{(i)} = \{y | d_E(y, p^{(i)}) < d_E(y, p^{(j)}), \forall i \neq j\}. \quad (60)$$

2.  $p^{(i)}$  is the center of the  $i$ -th region,  $P^{(i)}$ . For the Euclidean partition,  $p^{(i)}$  is the mean value of the data points classified to the  $i$ -th region.

After partitioning data, implementing PCA to each local region, and reduce dimension from  $n$  to  $m$ . Assume the locally principal components related to the  $i$ -th region are  $e^{(j)}$ ,  $j = 1, \dots, m$  and  $y^{(i)}$  is the subset of data classified to the  $i$ -th region. The lower dimensional representation for  $\forall y \in y^{(i)}$  is given by Eq. (61) [40],

$$z = \left( e_1^{(i)} \cdot (y - p^{(i)}), \dots, e_m^{(i)} \cdot (y - p^{(i)}) \right)$$

$$\hat{y} = p^{(i)} + \sum_{j=1}^m z_j e_j^{(i)}. \quad (61)$$

The mean squared reconstruction error can be calculated by

$$\varepsilon_{recon} = E[\|y - \hat{y}\|^2]. \quad (62)$$

### ***Projection Partition***

However, the Euclidean partition is not the optimal solution, because the data does not be reconstructed according to the reconstruction error in Eq. (61) [40]. With this in mind, the projection partition is proposed to classify data based on the contribution of the reconstruction error for each region [40]. According to Eq. (61, 62), the reconstruction distance is defined as

$$d(y, p^{(i)}) = \left\| y - p^{(i)} - \sum_{j=1}^m z_j e_j^{(i)} \right\|^2 = (y - p^{(i)})^T G^{(i)T} G^{(i)} (y - p^{(i)}) = (y - p^{(i)})^T \Pi^{(i)} (y - p^{(i)}), \quad (63)$$

where  $G^{(i)}$  is the  $(n - m) \times n$  matrix whose rows are the normalized eigenvectors of the covariance matrix  $\Sigma^{(i)}$  corresponding to the smallest  $(n - m)$  eigenvalues. The covariance matrix can be obtained by Eq. (64).

$$\Sigma^{(i)} = E[(y - Ey)(y - Ey)^T | y \in R^{(i)}], i = 1, \dots, K \quad (64)$$

The projection partition is to replace the Euclidean distance in Eq. (60) by the reconstruction distance shown as Eq. (65).

$$p^{(i)} = \{y | d(y, p^{(i)}) < d(y, p^{(j)}), \forall i \neq j\} \quad (65)$$

Therefore, the reconstruction error is minimized by the projection partition. There is one thing needed to be noticed, unlike the Euclidean partition, the regions separated by the projection partition may be disjoint [40]. Figure 8 is to illustrate the difference between those two partitions in a two dimensional case. The left part is the Euclidean partition, and the right part is the projection partition. The  $e_1^{(1)}$  and  $e_1^{(2)}$  axes are the principal components found by PCA in their corresponding regions, and  $y'$  is an unclassified data. According to Fig. 8, the Euclidean partition is to classify  $y'$  depending on the distance from  $y'$  to the center of each region, and the projection partition is depending on the shortest distance from  $y'$  to the principal component corresponding to each region [40].

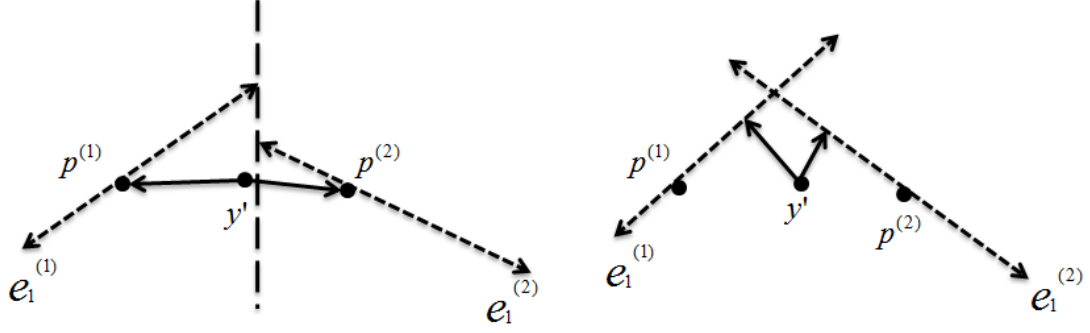


Figure 8. The illustration of the Euclidean partition and the projection partition. (Adapted from [40])

The procedure of the projection partition is summarized as follows [40]:

1. Assume the set of data,  $y$ , is partitioned into  $K$  regions. Initialize the  $K$  reference points,  $p^{(i)}$ ,  $i = 1, \dots, K$  by choosing randomly from the data set.
2. Initialize the covariance matrix,  $\Sigma^{(i)}$ ,  $i = 1, \dots, K$ , to the identity matrix. Use Eq. (63, 65) to classify data into their corresponding regions,  $P^{(i)}$ ,  $i = 1, \dots, K$ .
3. Update the center of each region by  $p^{(i)} = \bar{y}^{(i)}$ , where  $\bar{y}^{(i)}$  is the mean value of the data classified to the  $i$ -th region.
4. Assume the number of data in the  $i$ -th region is  $N_i$ . Update

$$\Sigma^{(i)} = \frac{1}{N_i} \sum_{y^{(i)} \in P^{(i)}} (y^{(i)} - p^{(i)})(y^{(i)} - p^{(i)})^T. \quad (66)$$

5. Repeat above steps until all data are classified.

After classifying each data point to its corresponding region, PCA is implemented to construct the specific linear model for each region [40]. Therefore, a lower dimensional representation is obtained. The advantage of LPCA is that the projection matrix for each local region is preserved, which is useful for sensor fault diagnosis. Another advantage is that data are partitioned by the projection partition based on reconstruction error [40]. Reconstruction error is the core concept of history process data based methods. Consider Eq. (21),  $Q$  matrix is composed by the eigenvectors corresponding to  $N - k$  smallest eigenvalues of the covariance matrix of data. Therefore, in Eq. (21),  $v_y \approx 0$  is based on assuming the reconstruction error is small enough to be neglected. The difference between LPCA and other manifold methods is that LPCA will not generate a smooth manifold [41]. This is a disadvantage from the aspect of image processing. However, generating an unsmooth manifold is not a drawback for the sensor fault diagnosis, because we only care about the reconstruction error instead of the smoothness of an image signal. Therefore, the LPCA algorithm is suitable to be applied on the nonlinear sensor fault diagnosis.

However, similar to applying MPPCA on the problem of sensor fault diagnosis, applying LPCA also comes out the issue of misclassification. This issue appears in both partition methods. This issue is illustrated by the projection partition in Fig. 9. Consider a two outputs nonlinear system, assume  $y'$  is a healthy data point,  $y_1$  and  $y_2$  are the measurements from the sensor 1 and sensor 2 represented in the two dimensional measurement space,  $p^{(1)}$  and  $p^{(2)}$  are the center of training data in the local region 1 and 2 respectively. From Fig. 9, we can understand that  $y'$  is classified to the region 2 when the system is healthy. However, when a sensor 1 drift fault,  $\Delta y_1$ , occurs, the data point,  $y'$ ,

will be placed at  $y^*$  and classified to the local region 1 by the projection partition algorithm. The reconstruction error is small due to the misclassification of  $y^*$ . Therefore, the sensor fault may not be detected and isolated in this situation, unless there is a better way to classify the faulty data.

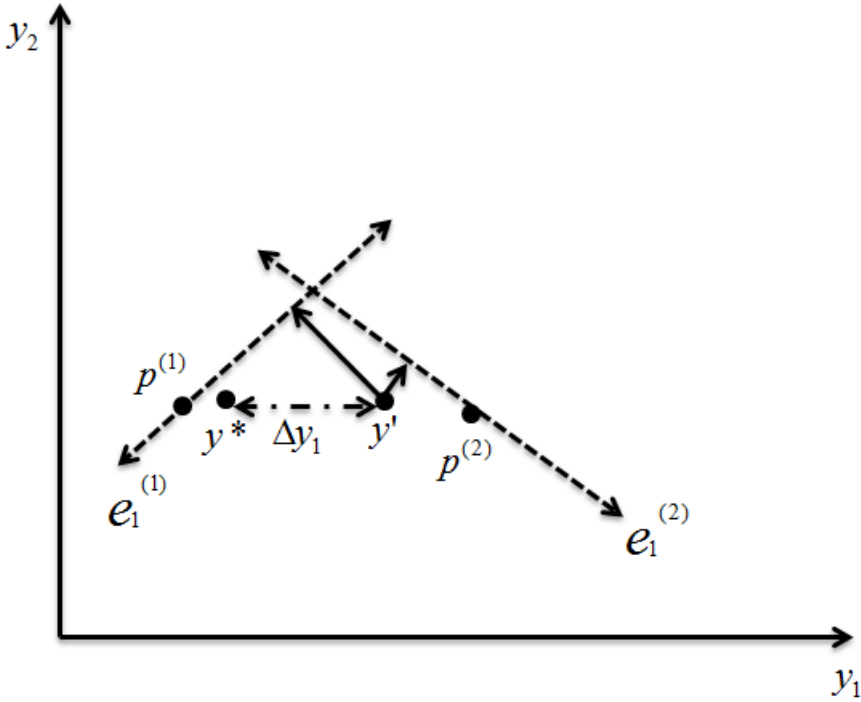


Figure 9. The illustration of misclassification.

**2.2.4 Expert Systems**

From previous discussion, dimension reduction based methods are limited in their isolability in both linear and nonlinear cases. Escobet et al.. and Zhang et al.. proposed expert system based approaches by using fuzzy logic for sensor fault detection and



isolation [44, 45]. The basic idea of their works is to construct a library which classifies every pattern contained in the training data. The biggest advantage of this kind of methods is its capability of transparent reasoning [3]. However, in order to isolate faults, all patterns of possible healthy/faulty scenarios have to be included in the training data. This is a general limitation of this kind of methods, since faulty data may not be available in some applications for safety issues. The other well-known limitation is that the constructed library developed from expert rules is very system-specific and difficult to update [3].

### **2.2.5 Phase Space Reconstruction**

In addition to the direction of dimension reduction, Chelidze et al. propose their methods to deal with the problem of fault diagnosis from a different perspective. In their works, they deal with the system parameter fault diagnosis problem in the *phase space* [46-52]. Those works are motivated by the needs to track slowly evolving hidden damage in the process [46]. Chelidze reconstructs the phase space by the system measurements, and predicts the future measurement by linearly and locally approximating the state trajectory in the reconstructed phase space [46]. Then, he successfully links the estimated error<sup>6</sup> to the slowly evolving hidden damage and examines his results both experimentally and numerically [46, 47, 49]. Moreover, Chelidze suggests compensate the fluctuation, which are not related to the changes of hidden parameter or damage, in the estimated error

---

<sup>6</sup> Here, the estimated error is the difference between the actual measurement and the predicted measurement.

by selecting a proper weighting function according to the reference data<sup>7</sup> probability density near the current measured data point [46]. However, this method can only track and identify single hidden damage. In order to address the multiple slowly evolving hidden damages problem, Chelidze induces the concept of multidimensional damage feature vector in his following work [48]. Multidimensional damage vector is formed by evaluating the estimated error in several disjoint regions of the reconstructed phase space [48]. Here, the number of elements of the multidimensional damage vector is the same as the number of those disjoint regions. After constructing the feature vector, Chelidze identifies the multiple slowly evolving hidden damages in the sense of dimension reduction [48]. He applies proper orthogonal decomposition (POD) and smooth orthogonal decomposition (SOD) to find the optimal coordinates (POCs) for approximating the estimated error in a lower dimension [48]. In essence, the optimal coordinates is composed with those coordinates with larger variance. Here, POD is also known as SVD in the discrete case, and SOD can be viewed as a different version of POD with an additional constraint requiring POCs to be smooth [48]. Then, Chelidze relates the optimal coordinates to each hidden damage based on his previous work, and he applies this method to a numerical model. According to his results, SOD has a significantly better signal-to-noise ratio than POD [48]. Therefore, SOD-based identification is recommended for the multiple damages situation [48]. Moreover, Chelidze further improves his method by partitioning the reconstructed phase space with a condition that every disjoint region

---

<sup>7</sup> Here, the reference data are the chosen nearest neighbors for the current measured data in the reconstructed phase space.

has the same data points [50]. The performance of this approach is examined by a two-dimensional damage accumulation process experimentally, and its results show that damages are identified correctly [50]. Chelidze's works provide a great start to further invest efforts to address the remain issues in the field of fault diagnosis, since the capability of tracking system states in the reconstructed phase space. However, although Chelidze et al... have done some good works in the field of fault diagnosis, their works are focused on the incipient system parameter faults. Therefore, a unified solution for isolating multiple sensor faults in both of linear and nonlinear cases is still unaddressed.

With this in mind, the proposed approach deals with the sensor fault diagnosis problem in the *phase space*, which offers several advantages. First of all, we have to emphasize especially on the isolability of multiple sensor faults or the occurrence of multiple abnormal events. Currently, there is no general solution for this problem. The existing methods are limited in the single sensor fault situation, requiring additional information of specific systems, or their performance cannot be guaranteed in nonlinear processes [17, 22, 28, 36, 37, 44]. By reconstructing the phase space from each sensor, the proposed approach can predict their future measurements, and then isolate multiple sensor faults in both of linear and nonlinear cases. Second is that the proposed approach has the capability of predicting future healthy sensor measurements. With the prediction capability, the proposed approach diagnoses sensor faults intuitively by comparing the predicted outputs with the actually measured outputs. Moreover, when the applied system is faulty, the predicted sensor measurement can replace the faulty measurement to avoid the potential damages of the applied system in the closed-loop sensor fault situation. Third

is associated with an assumption of PCA-related methods. Since PCA-related approaches consider approximation errors as residual signals, these approaches have to assume that the original data can be approximated well in a lower dimensional space. In the other words, the number of measurement channels has to be larger than the dimension of the applied system. The proposed approach deals with the fault diagnosis problem by reconstructing data in the phase space, so this assumption can be discarded. Fourth is that the proposed approach does not require specific information of the applied system. Therefore, the procedure of the proposed approach is unified for all applications.

### 3. THE PROPOSED METHOD

The structure of the proposed method is shown in Fig. 10. In this figure, blue blocks represent different functions/mechanisms for specific purposes. First, the training data are collected from the applied system. In the next step, we use the collected training data to reconstruct the phase space by the block of Phase Space Reconstruction. After reconstructing the phase space, one can obtain the reconstructed trajectory which can be viewed as the alternative of system state flow. In this stage, the reconstructed trajectory may contain a lot of fluctuations, which will lower the performance of the prediction mechanism. There are mainly two causes of the fluctuations. First is the effect of noise, and the second is the reconstructing fluctuations<sup>8</sup> [46]. In order to fix this issue, a smooth mechanism is adopted to reduce the fluctuations. Then, we use the smoothed trajectory to predict the future sensor measurements and compare to the actual measured measurements to detect and isolate sensor faults. In addition, we also apply the conservation/dissipation property in phase space to further extend the proposed method to deal with a slightly nonstationary situation due to setpoint changes.

---

<sup>8</sup> Here, the reconstructing fluctuations means that the fluctuations produced due to the phase space reconstruction techniques. In Chelidze's works, since he did not consider the effect of noise, the reconstructing fluctuations is the only cause of the fluctuations contained in the reconstructed trajectory.

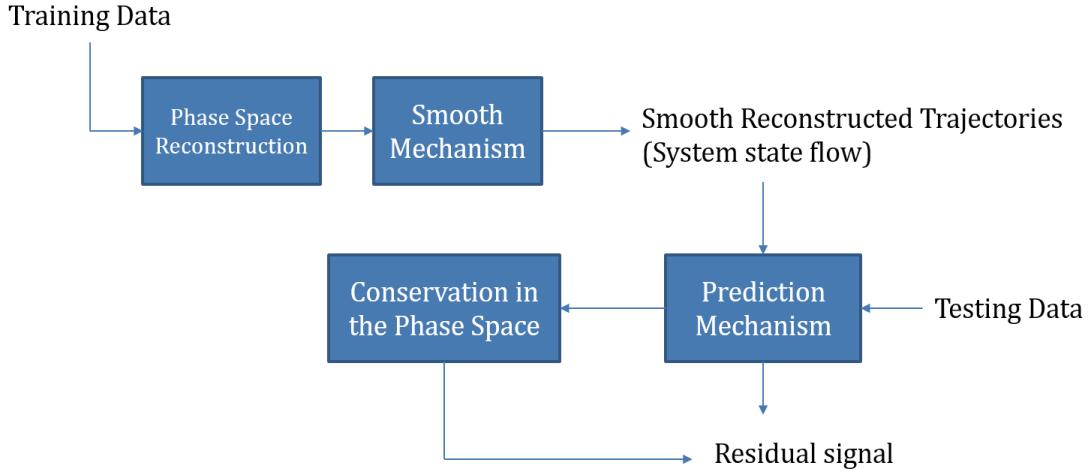


Figure 10. The structure of the proposed method.

### 3.1 Phase Space Reconstruction

The goal of phase space reconstruction is to reconstruct the state space of a dynamic system by observing its outputs [54]. Consider the discrete dynamical system,

$$\begin{aligned}
 x(k+1) &= f(x(k)) \\
 y(k) &= H(x(k)),
 \end{aligned} \tag{67}$$

where  $x \in R^n$  are the system states and  $y \in R^p$  are the system outputs, which can be observed in the measurement space. In the other words, the measurement function  $y = H(state)$  can be considered as a mapping from  $R^n$  to  $R^p$  [54]. According to Whitney's Embedding Theorem, assuming a  $d$ -dimensional state manifold  $M$  which approximates the trajectory of the system states in the state space,  $d < n$ , is sufficient to be reconstructed in the measurement space,  $R^p$ , by  $2d + 1$  simultaneous and independent measurements

[54, 55]. Although Whitney's theorem is in principle powerful, a large number of independent and simultaneous measurements may not be available in practice [54, 56]. Moreover, in the case of fault diagnosis, measurements of a single sensor are sampled as a time series. Therefore, a time series version of Whitney's theorem is used in this paper, namely Takens' time delay embedding theorem [56].

### 3.1.1 Takens' Time Delay Embedding Theorem

In mathematics, a flow is defined as a time dependent variable [54, 56]. Suppose  $\Phi$  is a state flow on a  $d$ -dimensional manifold  $M$ , and  $\Phi(X)$  is the flow map of the system states,  $x$ , in the phase space. In addition, assume  $T$  is a positive number (called the *time delay*), and  $h: M \rightarrow R$  is a smooth function. Define the  $p'$ -dimensional *delay-coordinate map*  $F(h, \Phi, T): M \rightarrow R^{p'}$  as:

$$F(h, \Phi, T)(x) = \left( h(\Phi(x)), h(\Phi_T(x)), h(\Phi_{2T}(x)), \dots, h(\Phi_{(p'-1)T}(x)) \right), \quad (68)$$

where  $\Phi_T(x)$  is to delay  $\Phi(x)$  with a delay time  $T$ . According to Takens' theorem, if  $p' > 2d$ , the delay-coordinate map in Eq. (68) will be an *embedding* of  $M$  [54, 56]. Here,  $F(h, \Phi, T)(x)$  is an *embedding* of  $M$  iff both  $F(h, \Phi, T)(x)$  and its derivative map  $DF(h, \Phi, T)(x)$  are one-to-one at every point  $x$  of  $M$  [54]. Assuming  $p' > 2d$ , Eq. (68) suggests that  $2d + 1$  simultaneous measurements in Whitney's theorem can be replaced by a  $p'$ -element delay-coordinate map. In other words, if  $p' > 2d$ , Eq. (68) is sufficient to

reconstruct the manifold  $M$  in the phase space [54, 56]. To extend Taken's theorem to the multiple output system described in Eq. (67), suppose  $y_k^{(i)}$ ,  $i = 1, \dots, p$ , is the  $i$ -th sensor measurement at time  $k$ , and choose  $T = 1$ . Then, we can obtain the general form of an embedding of the  $d$ -dimensional state manifold  $M$ ,

$$X(k) = \{y_k^{(1)}, y_{k+1}^{(1)}, \dots, y_{k+n_1-1}^{(1)}, y_k^{(2)}, \dots, y_{k+n_2-1}^{(2)}, y_k^{(p)}, \dots, y_{k+n_p-1}^{(p)}\}. \quad (69)$$

Here, the sufficient condition in Taken's theorem,  $p' > 2d$ , becomes  $\sum_{i=1}^p n_i > 2d$  [54]. Eq. (69) suggests that an embedding can be reconstructed by a series of lagged observations not only from multiple sensors, but also from a single sensor. For the convenience of illustrating the subsequent content, we constrain embeddings reconstructed from a single sensor. Let  $n_i = 2d + 1$ , and  $n_j = 0, \forall j \neq i$ . Then, Eq. (69) becomes

$$X(k) = \{y_k^{(i)}, y_{k+1}^{(i)}, \dots, y_{k+2d}^{(i)}\}. \quad (70)$$

In Eq. (70), the only parameter that needs to be determined is the dimension of the embedding. However, Takens' theorem, although correct in principle, is not precisely true in the real world because of the effect of noise [54]. Consider the  $2d + 1$  dimensional reconstruction described in Eq. (70) with a small  $d$ , each element will be indistinguishable in the presence of noise. Therefore, the reconstructed trajectory in the phase space will lie



on the line  $y_k^{(i)} = y_{k+1}^{(i)} = \dots y_{k+2d}^{(i)}$  [57]. In this case, the reconstructed trajectory will no longer be an embedding of  $M$  because it is no longer one-to-one at every point in  $M$ . In order to unfold the trajectory, an additional delay is added to make the elements distinguishable. Therefore, the chosen time delay should make the elements statistically independent, and Eq. (70) becomes

$$X_{\tau,d}(k) = \{y_k^{(i)}, y_{k+\tau}^{(i)}, y_{k+2\tau}^{(i)}, \dots, y_{k+2d\tau}^{(i)}\}. \quad (71)$$

According to Eq. (71), the time delay,  $\tau$ , and the dimension of the embedding need to be determined for reconstructing the phase space. Moreover,  $\tau$  has to be chosen such that the elements in the embedding are statistically independent, and the dimension of the embedding must be larger or equal to  $2d + 1$ . How to choose the proper time delay and optimal dimension of the embedding is a well-discussed issue in the field of phase space reconstruction [54]. This is discussed below.

### **3.1.2 Determination of Time Delay and Dimension of an Embedding in Phase Space Reconstruction**

Various methods have been used to determine the time delay and the dimension of an embedding separately such as autocorrelation and mutual information for estimating the time delay [57] and False Nearest Neighbor (FNN) algorithm for estimating the dimension of embeddings [58].

### 3.1.2.1 Time Delay

To determine the time delay,  $\tau$ , an easier way is to choose the smallest  $\tau$  which makes the autocorrelation function close or equal to zero [57]. However, the autocorrelation function can only measure *linear dependence* of the elements. To improve this, the mutual information is proposed to measure the general dependence of the elements [57]. To illustrate the notion of mutual information, let us consider a two-dimensional reconstruction,  $X_{\tau,d}(k) = \{y_k^{(i)}, y_{k+\tau}^{(i)}\}$ . Fraser suggests that the time delay should be chosen to make  $y_k^{(i)}$  and  $y_{k+\tau}^{(i)}$  independent from the standpoint of mutual information [57]. The notion of mutual information is to measure the information shared by these two variables, as defined by:

$$I(Y_k^{(i)}, Y_{k+\tau}^{(i)}) = \sum_{y_k^{(i)} \in Y_k^{(i)}} \sum_{y_{k+\tau}^{(i)} \in Y_{k+\tau}^{(i)}} p(y_k^{(i)}, y_{k+\tau}^{(i)}) \log \frac{p(y_k^{(i)}, y_{k+\tau}^{(i)})}{p(y_k^{(i)})p(y_{k+\tau}^{(i)})}, \quad (72)$$

where  $Y_k^{(i)}$  and  $Y_{k+\tau}^{(i)}$  are two time series of  $y_k^{(i)}$  and  $y_{k+\tau}^{(i)}$  respectively [57]. In practice, the difficulty in calculating mutual information from experimental data is in estimating the joint probability for  $y_k^{(i)}$  and  $y_{k+\tau}^{(i)}$ . Fraser estimates  $p(y_k^{(i)}, y_{k+\tau}^{(i)})$  by selecting a box in the  $(Y_k^{(i)}, Y_{k+\tau}^{(i)})$  plane of size  $\Delta$  around the point  $(y_k^{(i)}, y_{k+\tau}^{(i)})$ . Assume there are  $N_j$  points in the box, and then  $p(y_k^{(i)}, y_{k+\tau}^{(i)})$  is estimated to be  $N_j / N_{total} \Delta$ , where  $N_{total}$  is the number

of total points in the  $(Y_k^{(i)}, Y_{k+\tau}^{(i)})$  plane [57]. In addition, for choosing the time delay, Fraser calculates  $I(Y_k^{(i)}, Y_{k+\tau}^{(i)})$  for  $\tau = 1, \dots, N$ , and chooses the  $\tau$  at which the first local minimum of  $I(Y_k^{(i)}, Y_{k+\tau}^{(i)})$  occurs [57].

### 3.1.2.2 Dimension of Embedding

After choosing the time delay, the dimension of the embedding is chosen by the False Nearest Neighbor (FNN) algorithm [58]. The physics behind the FNN algorithm is that the data in the true phase space will be projected to a lower dimensional space if the chosen dimension of the embedding,  $D$ , is lower than the optimal dimension of the embedding,  $D_{opt}$ <sup>9</sup> [58]. In that situation, any two points which are not close to each other in the optimal dimension may become neighbors in the lower dimension,  $D$ . This phenomenon is called false nearest neighbor. Therefore, the FNN value is calculated according to the ratio of the distances of the two data points in the  $D$  and  $D + 1$  dimensional spaces shown in Eq. (73) [58].

$$FNN \text{ value} = 1 - \frac{\text{distance in dimension } D}{\text{distance in dimension } D + 1} \quad (73)$$

---

<sup>9</sup> According to Takens' theorem,  $D_{opt}$  equals to  $2d + 1$  without considering the effect of noise and disturbance.

Observing Eq. (73), when  $D < D_{opt}$ , the FNN value will be larger than a threshold. In contrast, the value will be close to zero when  $D > D_{opt}$ . The dimension of the embedding can be determined accordingly.

### 3.1.2.3 Applications of Phase Space Reconstruction

Phase space reconstruction techniques can be applied to several applications. One of them is to find the underlying deterministic pattern from a time series [53]. A famous example is the Lorenz equations shown in Eq. (74) [59].

$$\begin{aligned}x' &= \sigma(y - x) \\y' &= x(r - z) - y \\z' &= xy - bz\end{aligned}\tag{74}$$

The Lorenz equations are first proposed by Lorenz for atmospheric convection. In this example, we use the standard settings for their parameters,  $\sigma=10$ ,  $r=28$ ,  $b=8/3$ . In addition, the initial conditions are set to be  $(x_0, y_0, z_0) = (5, 5, 5)$ . For the first two thousands samples, the time series of  $x$ ,  $y$ , and  $z$  are shown in Fig. 11.

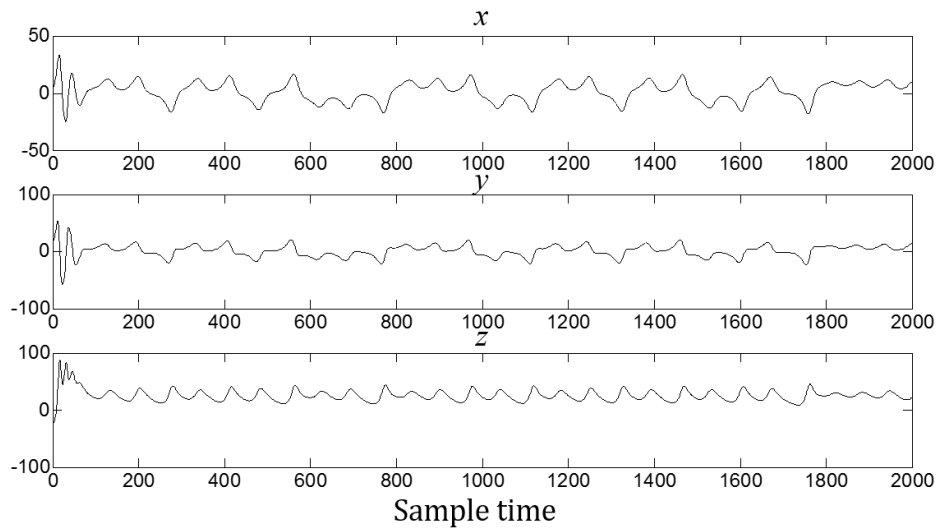


Figure 11. Time series of  $x$ ,  $y$ , and  $z$  in Lorenz equations.

According to the above figure, it is difficult to find the deterministic pattern by only observing those time series. However, by applying phase space reconstruction, one can find their deterministic pattern. Figure 12 is the result of applying mutual information method to the time series of  $x$ . According to this result, one can see that the first local minimum of mutual information occurs at sample time 19. Therefore, we choose the time delay to be 19 for reconstructing its phase space.

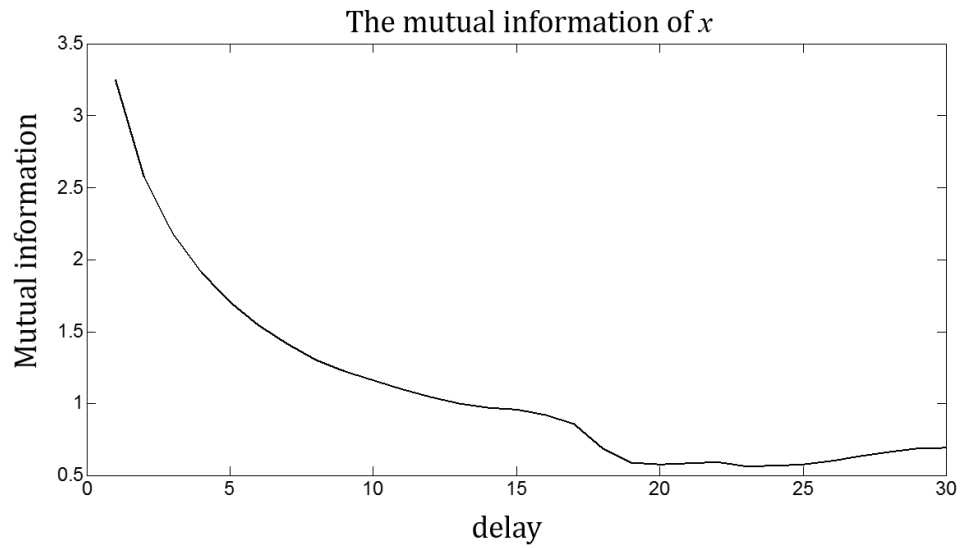


Figure 12. The result of applying mutual information method to the time series of  $x$ .

For choosing the dimension of embedding, Fig. 13 is the result of applying FNN method to the time series of  $x$ . According to this result, the FNN value is close to zero when the dimension is larger or equal to 3. Therefore, the dimension of embedding is chosen to be 3.

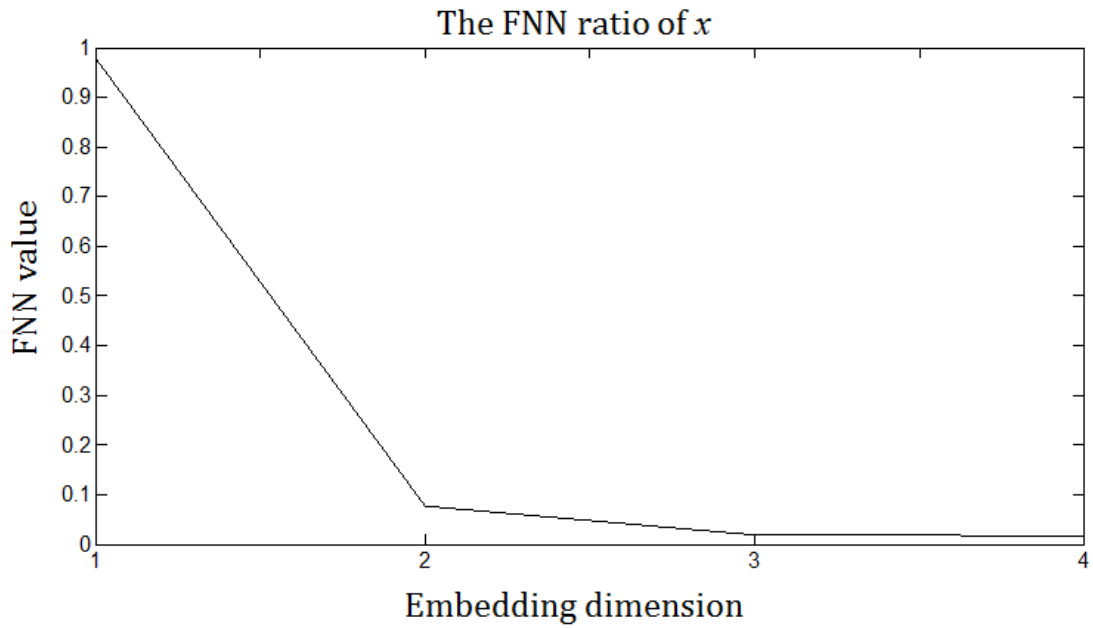


Figure 13. The result of applying FNN method to the time series of  $x$ .

Figure 14 is the reconstructed phase space of the time series of  $x$ . By applying the phase space reconstruction techniques, the deterministic pattern of the time series is explicitly seen.

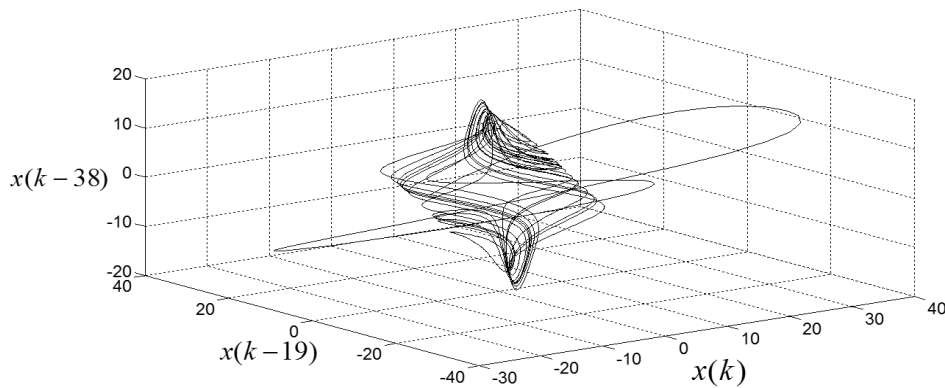


Figure 14. The reconstructed phase space of the time series of  $x$ .

In addition, the reconstructed phase space can be also considered as a nonlinear relation between current measurement and former measurements of a dynamic system or mathematical model [53]. For instance, the reconstructed trajectory in Fig. 14 can be viewed as a nonlinear relation between  $x(k)$ ,  $x(k - 19)$ , and  $x(k - 38)$ . Therefore, by phase space reconstruction techniques, it is possible to predict the future measurements by the former measurements of a dynamic system.

### **3.2 Smooth Mechanism**

The purpose of the smooth mechanism in Fig. 10 is aimed to reduce the fluctuations contained in the reconstructed trajectory and improve the performance of the prediction mechanism. As mentioned before, there are two reasons for the occurrence of fluctuations. First is due to the effect of noise, and second reason is the reconstructing fluctuations. The cause of the reconstructing fluctuations is that the chosen time delay may not be able to make the elements in the current embedding perfectly independent to each other [46]. In the other words, practically, it may not be able to find a time delay which makes the mutual information exactly equals to zero. Since the prediction mechanism is to estimate the future measurements based on the reconstructed trajectory, fluctuations will lead the prediction inaccurate and then lower the performance of fault detection. Therefore, two strategies are introduced here in order to fix this problem.



### 3.2.1 Minimum Variance Unbiased Estimator (MVU)

The first strategy is to consider this problem as an estimation problem, since our purpose is to estimate the true value<sup>10</sup> from the reconstructed trajectory contaminated by fluctuations. Therefore, in this direction, the smooth mechanism functions as an estimator. The estimation problem can be described as:

$$Y = A(k) + w. \quad (75)$$

Where  $Y$  is the reconstructed trajectory obtained from sensor measurements,  $A(k)$  is the true value of  $Y$  at the  $k^{\text{th}}$  instant, and  $w$  is the fluctuations. Assume  $w$  is Gaussian distributed, and MVU estimator exists. A  $L$ -sample averaging frame can be considered to be the minimum variance unbiased (MVU) estimator if  $L$  is small so that  $A(t)$  can be considered as a constant in the averaging frame [60]. The mathematical description of the MVU estimator is shown as Eq. (76). In Eq. (76),  $\widehat{A}_e(k)$  represents the estimated true value at the  $k^{\text{th}}$  instant.

$$\widehat{A}_e(k) = \sum_{i=k-L+1}^k y_e(i) / L \quad (76)$$

In the other words, averaging samples is the best way to estimate the true value of  $Y$  under above assumptions. More precisely, assume  $w$  is Gaussian with variance  $\sigma^2$ .

---

<sup>10</sup> Here, true value means that the data without noise and reconstructing fluctuations.

Because of the fluctuations,  $Y$  no longer equals to its true value, and  $Y \sim N(A(t), \sigma^2)$ . After averaging  $Y$  with its former data, the averaged value,  $\bar{Y} \sim N(A(t), \sigma^2/L)$ , has lower variance and is closer to its true value,  $A(t)$ . Therefore, the effect of fluctuations can be reduced.

### **3.2.2 Smooth Orthogonal Decomposition (SOD)**

The second strategy is to consider the problem of fluctuations as a noise reduction problem. On this track, we consider the fluctuations as noise contaminating the reconstructed trajectory. With this in mind, we applied dimension reduction techniques to this problem. Among all dimension reduction techniques, PCA-related methods have been considered the most efficient tool for reducing the effect of noise in field of signal processing [23]. Consider the measurement space of a dynamic system, PCA-related methods are aimed to find few orthogonal components with largest variance [23]. After finding those components with largest variance, the rest components will be considered as noise and can be neglected. However, PCA-related methods do not consider the tendency of the reconstructed trajectory. In order to keep the smoothness of the reconstructed trajectory, we introduce the smooth orthogonal decomposition (SOD) as a smooth mechanism. SOD can be viewed as a constrained version of singular value decomposition (SVD). This approach is first proposed by Chelidze et al. for extracting linear normal modes and natural frequencies of multi-degree-of-freedom and distributed-parameter vibration systems [61].

To illustrate the SOD algorithm, let  $y_e \in R^{n_e \times m_e}$  be the  $m_e$  dimension reconstructed trajectory with  $n_e$  points. In order to describe the smoothness of  $y_e$ , defining a  $(n_e - 1) \times n_e$  differential operator:

$$D_e = 1/\Delta t \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}. \quad (77)$$

Where  $\Delta t$  is the sample time between each sample. After defining the differential operator, the approximated velocity matrix can be obtained by multiplying  $y_e$ .

$$V_e = D_e y_e \quad (78)$$

Let  $\Phi$  be a set of linear orthogonal combinations. The goals of the noise reduction problem can be described as the following constrained maximum variance problem:

$$\max_{\Phi} \|y_e \Phi\|^2 \text{ subject to } \min_{\Phi} \|V_e \Phi\|^2. \quad (79)$$

By derivations, Eq. (79) will become a *generalized eigenvalue problem* shown as Eq. (80).

$$\Sigma_{y_e} \Phi_i = \lambda_i \Sigma_{V_e} \Phi_i, \quad i = 1, \dots, m_e \quad (80)$$

Where,  $\Sigma_{y_e}$  and  $\Sigma_{V_e}$  are the covariance matrix of  $y_e$  and  $V_e$  respectively,  $\lambda_i$  is the  $i^{\text{th}}$  largest generalized eigenvalue, and  $\Phi_i$  is the generalized eigenvector corresponding to  $\lambda_i$ . Composing  $\Phi$  by a set of generalized eigenvectors corresponding to larger generalized eigenvalues, the fluctuation-reduced trajectory,  $y_s$ , can be obtained in a lower dimension shown as

$$y_s = y_e \Phi. \quad (81)$$

Compared with MVU estimator, SOD can achieve a smoothed trajectory with lower dimension. Therefore, SOD is more memory efficient in practice.

### 3.3 Prediction Mechanism

After obtaining smoothed trajectory, a prediction mechanism is induced for predicting the future measurement of each sensor in the applied system. In principle, embedding is not unique [54-56]. Therefore, assume there are totally  $p$  sensors in the applied system, one can obtain multiple embeddings corresponding to each sensor such as

$$\begin{aligned} X_1(k) &= \{y_{i_1,2d+1}(k - \tau_{1,2d+1}), \dots, y_{i_1,2}(k - \tau_{1,2}), y_1(k)\} \\ X_2(k) &= \{y_{i_2,2d+1}(k - \tau_{2,2d+1}), \dots, y_{i_2,2}(k - \tau_{2,2}), y_2(k)\} \\ &\vdots \\ X_p(k) &= \{y_{i_p,2d+1}(k - \tau_{p,2d+1}), \dots, y_{i_p,2}(k - \tau_{p,2}), y_p(k)\}, \end{aligned} \quad (82)$$

where  $y_{i_{a,b}}, i_{a,b} \in \{1, \dots, p\}$ , is the  $i_{a,b}$ -th sensor measurement,  $k$  is sample time, and  $X_j, j = 1, \dots, p$ , is the embedding corresponding to the  $j^{\text{th}}$  sensor. Notice that elements in each embedding are not necessarily obtained from the corresponded sensor. Pecora has proven that an embedding can be composed by the measurements from different measurement channel [62]. The purpose of the prediction mechanism is to estimate the future measurement of each sensor,  $y(k + 1)$ .

$$y(k + 1) = (y_1(k + 1), y_2(k + 1), \dots, y_p(k + 1)) \quad (83)$$

According to our search of the relevant literature, very little application has been done with prediction by using phase space reconstruction. Farmer and Sidorowich proposed an approach to predict chaotic signals [53]. In their work, they assume that they have all trajectories of embeddings obtained from the training data, and then the prediction is done according to these trajectories [53]. Liu et al.. applied Farmer's approach to predict the daily streamflow in the United States [63]. In addition, Chelidze et al.. also applied Farmer's approach for detecting system parameter faults [46-52].

In order to illustrate the prediction mechanism, suppose we have all trajectories of  $X_i(k)$  for  $i = 1, \dots, p$  for a long time run, and these trajectories are smooth enough, at least  $C^2$ , to be approximated their geometric characteristics. Usually, local approximation, i.e. fits a given number of data points in their local region, produces better fits than global approximation [53]. Therefore, a proper local approximation is the key to predict the future measurement accurately. After obtaining a proper local approximation, the future

measurement can be predicted according to these trajectories [53]. The basic idea is shown in Fig. 15. In the figure,  $X_i(k)$  is the current state for the  $i$ -th embedding,  $X_i(k + 1)$  is the unknown future state, and the solid curve is the trajectory obtained from the training data. The black dots inside the dashed circles are the neighbors of  $X_i(k)$  and  $X_i(k + 1)$  respectively. Ideally, without considering the effect of fluctuation, the dashed curve will perfectly overlap with the solid curve.

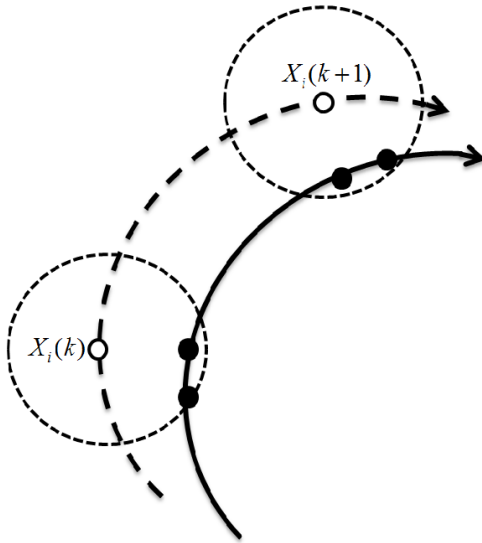


Figure 15. The illustration of the prediction approach; arrow represents the direction of state flow. (Adapted from [53])

Consider Fig. 15,  $X_i(k)$  is given by the current sensor measurement while its neighbors,  $n_j(k)$ ,  $j = 1, \dots, N_j$ , are chosen from the embedding trajectory, where  $N_j$  is the number of neighbors. The neighbors of  $X_i(k + 1)$  are determined by  $n_j(k + 1)$ . Therefore, by finding a local chart that maps  $n_j(k)$  into  $n_j(k + 1)$ , the future state,  $X_i(k + 1)$ , can

be predicted<sup>11</sup> [53]. After obtaining the future state, the predicted sensor measurement for the  $i$ -th sensor,  $\hat{y}_i(k + 1)$ , is the first element of  $X_i(k + 1)$  according to Eq. (82). Once  $\hat{y}_i(k + 1)$  is obtained, it will be used to compare with the real measured  $i$ -th sensor output,  $y_i(k + 1)$ , and the difference between  $\hat{y}_i(k + 1)$  and  $y_i(k + 1)$  can be further processed to generate the residual signal for the use of fault detection and isolation.

### 3.4 Conservation/Dissipation Property in Phase Space

The conservation/dissipation property in the phase space is introduced in order to improve the proposed method in a sub-problem. In order to illustrate the problem, consider Eq. (67), the applied system is assumed to be autonomous in the proposed method. However, in practice, the setpoint of the applied system may be changed during operating. This fact will lead the applied system becomes time varying and then violate the assumption.

In order to fix this problem, our strategy is to consider every change of setpoint as different autonomous systems. Figure 16 is the illustration of the setpoint changes problem. Assume the applied system has two setpoints. At sample time 500, its setpoint changes from 45 to 55. Our strategy is to consider the applied system as two autonomous systems. Before sample time 500, the applied system is represented by system 1. On the other hand, the applied system is represented by system 2 after sample time 500. In addition, the initial condition of system 2 is the same as the setpoint of system 1.

---

<sup>11</sup> In our simulation, we use the linear regression as the local approximation.

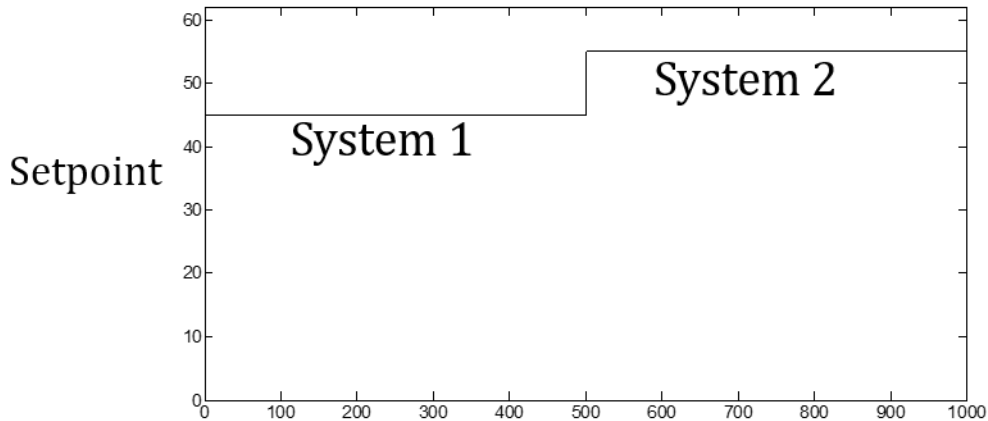


Figure 16. The illustration of the setpoint changes problem.

However, this strategy will lead another problem. According to the uniqueness and existence theorem proposed by Peano, in a dynamic system, every pair of setpoint and initial conditions is corresponded to a unique states flow in the phase space [64]. Now, assume our training data is comprehensive, i.e. including all possible pairs of setpoint and initial conditions. If the applied system has  $N_s$  different setpoints, there will be  $N_s^2$  different pairs. In the case, for each sensor,  $N_s^2$  trajectories have to be saved for fault detection and isolation. Since saving all  $N_s^2$  trajectories is impractical, the conservation/dissipation property is adopted to relate these trajectories by the inherent properties of the phase space and lower the number of trajectories needed to be saved.

To illustrate the conservation/dissipation property, consider a dynamic system with three states:



$$\begin{aligned}
\dot{h}_1 &= f_1(h_1, h_2, h_3) \\
\dot{h}_2 &= f_2(h_1, h_2, h_3) \\
\dot{h}_3 &= f_3(h_1, h_2, h_3) \text{ or } \frac{d\vec{h}}{dt} = \vec{f}(\vec{h}).
\end{aligned}
\tag{84}$$

Eq. (84) describes the system states flow in the phase space. Now, assume a set of initial conditions enclosed in a volume  $V$  flows to another position in the phase space, where it occupies a volume  $V'$  shown as Fig. 17.

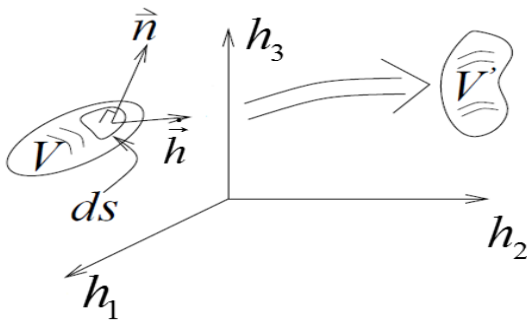


Figure 17. The phase space of a three states dynamic system. (Adapted from [64])

Define  $S$  to be the surface of the volume  $V$ ,  $\rho$  to be the density of initial conditions in  $V$ ,  $\rho\vec{f}$  to be the rate of flow of points, i.e. trajectories emanating from initial conditions through unit area perpendicular to the direction of flow [64]. In Fig. 17,  $dS$  represents a small region of  $S$ , and  $\vec{n}$  is the outward unit normal to  $dS$ . In order to illustrate the conservation/dissipation property and its proof, first step is to calculate the net flux of points out of  $S$ .

$$\int_V \frac{\partial \rho}{\partial t} dV = - \int_S (\rho \vec{f} \cdot \vec{n}) ds \quad (85)$$

Then, applying the divergence theorem to the right-hand part of Eq. (85), one can obtain

$$\int_V \frac{\partial \rho}{\partial t} dV = - \int_V [\vec{\nabla} \cdot (\rho \vec{f})] dV. \quad (86)$$

According to Eq. (86),

$$\frac{\partial \rho}{\partial t} = -\vec{\nabla} \cdot (\rho \vec{f}). \quad (87)$$

The second step is to calculate  $\frac{d\rho}{dt}$ , i.e. the rate of change of  $\rho$  as the volume moves, according to Eq. (87). The calculation is shown as

$$\begin{aligned} \frac{d\rho}{dt} &= \frac{\partial \rho}{\partial t} + \frac{\partial \rho}{\partial h_1} \frac{dh_1}{dt} + \frac{\partial \rho}{\partial h_2} \frac{dh_2}{dt} + \frac{\partial \rho}{\partial h_3} \frac{dh_3}{dt} = -\vec{\nabla} \cdot \\ &(\rho \vec{f}) + (\vec{\nabla} \rho) \cdot \vec{f} = -(\vec{\nabla} \rho) \cdot \vec{f} - \rho \vec{\nabla} \cdot \vec{f} + (\vec{\nabla} \rho) \cdot \vec{f} = -\rho \vec{\nabla} \cdot \vec{f}. \end{aligned} \quad (88)$$

In addition, although the boundaries of  $V$  and  $V'$  deform, they always contains the same points. Now, assume the number of points in  $V$  is  $N$ ,  $N = \rho V$ . Since the number of points is constant, we must have

$$dN/dt = V d\rho/dt + \rho dV/dt = 0. \quad (89)$$

Then, substitute Eq. (88) into Eq. (89), we obtain the *Lie derivative* shown in Eq. (90).

$$1/V dV/dt = \vec{\nabla} \cdot \vec{f} \quad (90)$$

The Lie derivative describes the behavior of volume  $V$  when the system states move. Consider the Lie derivative, there are two situations of the volume  $V$  when the system states move: conservative and dissipative. The case of conservative occurs when the right-hand part of Eq. (90) equals to zero. In this case, assume there are two state trajectories starting from different initial conditions, the volume  $V$  between them will keep the same when the system states move [64]. A famous example for this case is the frictionless pendulum. On the other hand, the case of dissipative occurs when the right-hand part of Eq. (90) is negative. In the case, the volume  $V$  between two system trajectories starting from different initial condition will contract when the system states move to their setpoint [64]. An example for this case is the pendulum with friction or damping.

Since the applied system will be controlled to achieve its setpoint, our application belongs to the case of dissipative. Therefore, according the dissipative property, assume we have two reference system states trajectories with same setpoint and different initial conditions, we can set the upper and lower boundaries for all other trajectories starting from other initial conditions shown as Eq. (91).

$$\begin{aligned}
& (\text{Volume between } E_1 \text{ and } E_t \text{ when } t = k+1 ) < (\text{Volume between } E_1 \text{ and } E_t \text{ when } t = \\
& k) + \delta_1 \\
& (\text{Volume between } E_2 \text{ and } E_t \text{ when } t = k+1 ) < (\text{Volume between } E_2 \text{ and } E_t \text{ when } t = \\
& k) + \delta_2 \tag{91}
\end{aligned}$$

Where  $E_1$  and  $E_2$  are the two reference trajectories,  $E_t$  is the unknown trajectory starting from other initial conditions,  $k$  is the time instant, and  $\delta_1$  and  $\delta_2$  are two small constants in order to tolerate the effect of fluctuations. In essence, Eq. (91) is aimed to timely examine whether the volume contracts or not. Furthermore, by shrinking the volume, we can calculate distance between two trajectories instead of volume, and then Eq. (91) will become

$$\begin{aligned}
& (\text{Distance between } E_1 \text{ and } E_t \text{ when } t = k+1 ) < (\text{Distance between } E_1 \text{ and } E_t \text{ when } t \\
& = k) + \delta_1 \\
& (\text{Distance between } E_2 \text{ and } E_t \text{ when } t = k+1 ) < (\text{Distance between } E_2 \text{ and } E_t \text{ when } t \\
& = k) + \delta_2. \tag{92}
\end{aligned}$$

Therefore, according the dissipative property, if  $E_t$  obeys Eq. (92), the applied system is considered as healthy. Otherwise, the applied system is faulty.

In addition, an estimation mechanism is needed when the applied system is faulty. This is because when the applied system is faulty, an estimated value of  $E_t$  is needed for the following diagnosis. There are three situations considered in this part: 1. No

trajectories has arrived the setpoint; 2. The unknown trajectory,  $E_t$ , has arrived the setpoint; 3. Only one of reference trajectories has arrived the setpoint. In the first case, all trajectories will keep moving in the phase space. Since we don't have any information about the unknown trajectory except its initial condition and setpoint. Therefore, the estimated value of  $E_t$  is produced according to the relative position of their initial conditions. Figure 18 is an example for illustrating how the estimation mechanism works in the first case. Assume there is a 3-D reconstructed phase space, the initial condition of the unknown trajectory is  $(10, 10, 10)$ , and the initial conditions of two reference trajectories are  $(20, 20, 20)$  and  $(0, 0, 0)$  respectively. In this example, according to the relative position of the initial conditions of these three trajectories, the estimated value of the unknown trajectory is produced by averaging the value of two reference trajectories.

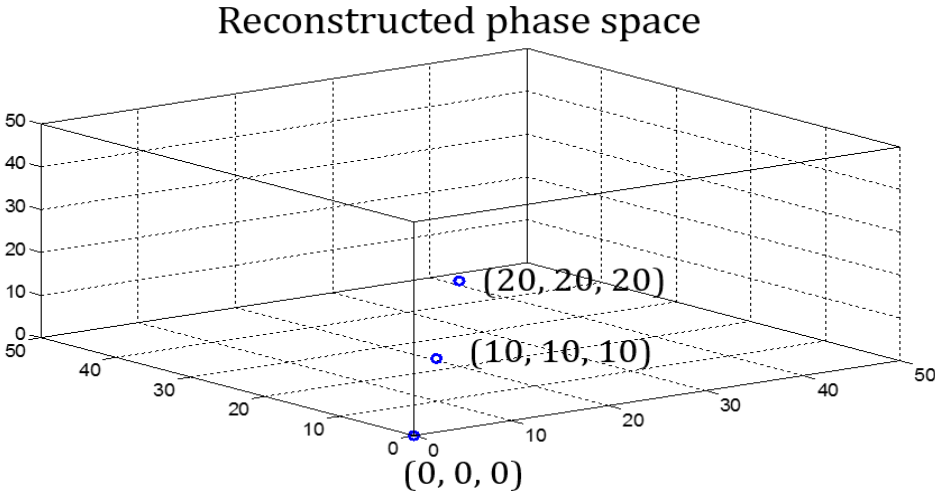


Figure 18. An example for illustrating how the estimation mechanism works in the first case.

In the second case, since the unknown trajectory,  $E_t$ , has arrived the setpoint,  $E_t$  will stay at a fixed point in the phase space. Therefore, in this case, the estimated value of  $E_t$  is set to be the same as the setpoint. In the third case, one of two reference trajectories will stay at a fixed point in the phase space. Similar to the first case, the estimated value of  $E_t$  is generated by the relative position of the previous estimated value of  $E_t$ , the setpoint, and the current value of the other reference trajectory which has not arrived the setpoint.

## 4. SIMULATION RESULTS

As mentioned in the section 3.3, since embedding is not unique, the proposed method can be applied on every sensor in parallel and viewed as a residual generator for the corresponding sensor. In the other words, each residual generator is supposed to indicate the sensor fault occurred in its corresponding sensor, and then isolation will be done simultaneously. Therefore, this simulation is aimed to test the performance of the proposed method served as a residual generator for a single sensor.

### **4.1 Generating Training and Testing Data**

#### **4.1.1 Three Tank System**

The proposed method is applied to a three tank system. This model is a nonlinear process, and modeled according to the typical characteristics of tanks, pipelines and pumps used in chemical industry [4]. The three tank model introduced in this paper is a laboratory setup DTS200 shown as Fig. 19 [4].

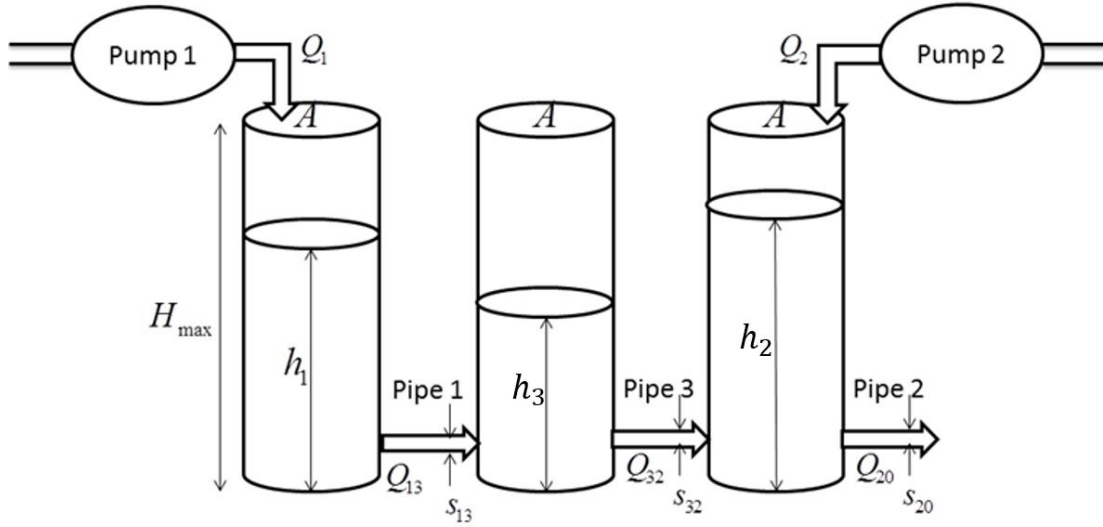


Figure 19. DTS200 setup. (Adapted from [4])

In the figure,  $Q_1$  and  $Q_2$  are incoming mass flow,  $Q_{20}$  is the outgoing mass flow, and  $Q_{ij}$  is the mass flow from the  $i$ -th tank to the  $j$ -th tank. The height of each tank,  $h_1$ ,  $h_2$ , and  $h_3$  are considered as system outputs and measured by sensors. The relative parameters are listed in **Table 1**, and the dynamics of DTS200 is modeled as Eq. (93) [4].

$$\begin{aligned}
 Ah_1 &= Q_1 - Q_{13} \\
 Ah_2 &= Q_2 + Q_{32} - Q_{20} \\
 Ah_3 &= Q_{13} - Q_{32} \\
 Q_{13} &= a_1 s_{13} \operatorname{sgn}(h_1 - h_3) \sqrt{2g|h_1 - h_3|} \\
 Q_{32} &= a_3 s_{32} \operatorname{sgn}(h_3 - h_2) \sqrt{2g|h_3 - h_2|} \\
 Q_{20} &= a_2 s_{20} \sqrt{2gh_2}
 \end{aligned} \tag{93}$$



Table 1. Parameters of DTS200 (Adapted from [4])

Parameters	Symbol	valve	Unit
Cross section area of tanks	$A$	154	$\text{cm}^2$
Cross section area of pipes	$s_{12}, s_{32}, s_{20}$	0.5	$\text{cm}^2$
Maximum height of tanks	$H_{max}$	62	cm
Maximum flow rate of pump 1	$Q_{1max}$	100	$\text{cm}^3/\text{sec}$
Maximum flow rate of pump 2	$Q_{2max}$	100	$\text{cm}^3/\text{sec}$
Coeff. Of flow for pipe 1	$a_1$	0.46	
Coeff. Of flow for pipe 2	$a_2$	0.6	
Coeff. Of flow for pipe 3	$a_3$	0.45	

In addition, a nonlinear controller is induced in DTS200 in order to control the height of tank 1 and 2 to their setpoint. The block diagram of whole system is shown as Fig. 20.

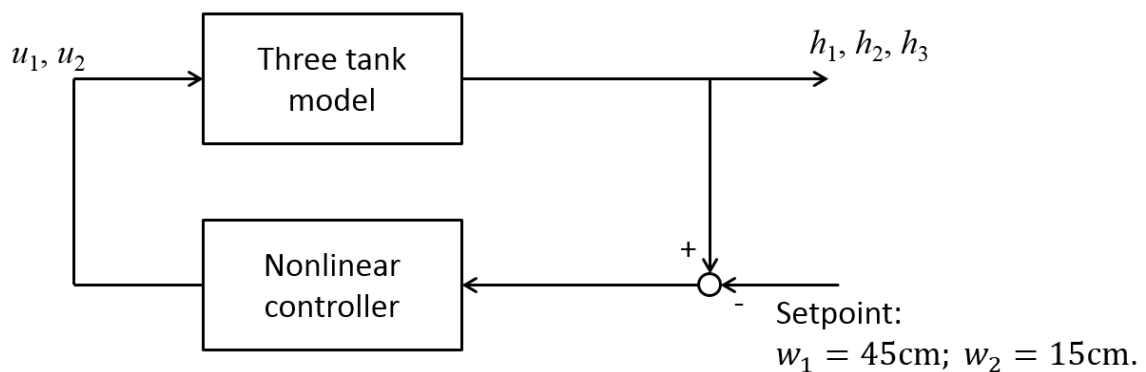


Figure 20. The block diagram of the three tank system.

The dynamic equations of the nonlinear controller are described as Eq. (94),

$$\begin{aligned} u_1 &= Q_1 = Q_{13} + A(a_{11}h_1 + v_1(w_1 - h_1)) \\ u_2 &= Q_2 = Q_{20} - Q_{32} + A(a_{22}h_2 + v_2(w_2 - h_2)). \end{aligned} \quad (94)$$

Where  $w_1$  and  $w_2$  are the desired height of tank 1 and tank 2 respectively, and we set  $a_{11} = a_{22} = 0$  and  $v_1 = v_2 = 1$  in our simulation. In our simulation, we set the setpoint to be  $w_1 = 45$  cm and  $w_2 = 15$  cm.

#### 4.1.2 Measurement Noise

In order to more accord with real world applications, measurement noise is considered in our simulation. The measurement noise is assumed to be zero-mean and Gaussian distributed with three different level of variance:  $5 \text{ cm}^2$  (10%),  $1 \text{ cm}^2$  (5%), and  $0.1 \text{ cm}^2$  (1.5%). Figure 21 is to illustrate where the measurement noise is added.

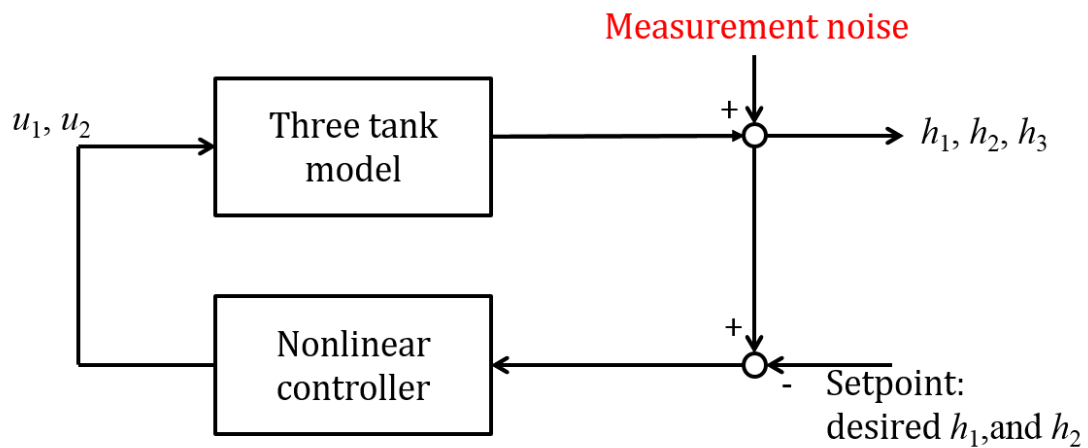


Figure 21. The illustration of modeling measurement noise.

### 4.1.3 Training Data

In our simulation, we use the first 300 samples of sensor 1 output in the fault-free situation to be our training data. Figure 22, 23, and 24 are the training data generated for fault detection and isolation.

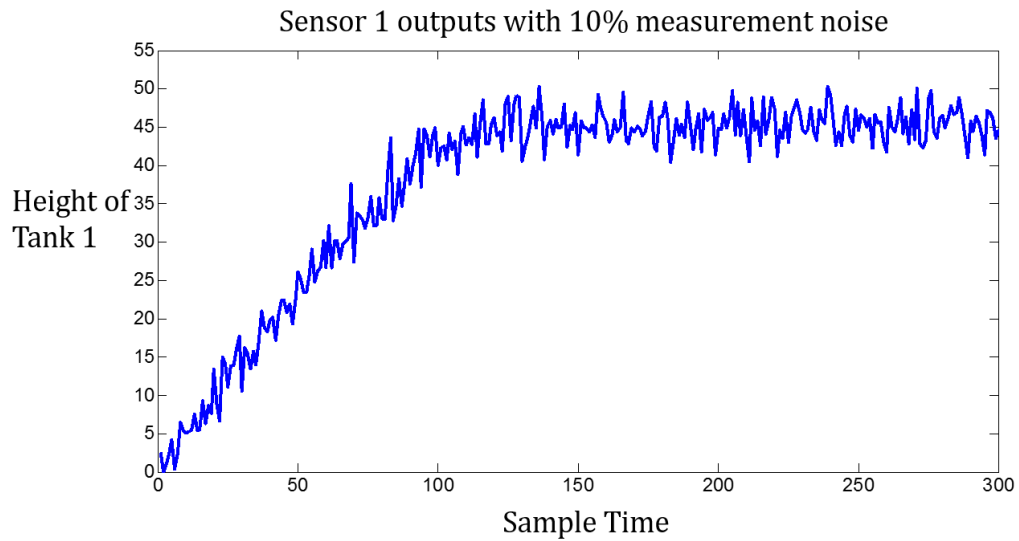


Figure 22. Training data - sensor 1 output with 10% measurement noise.

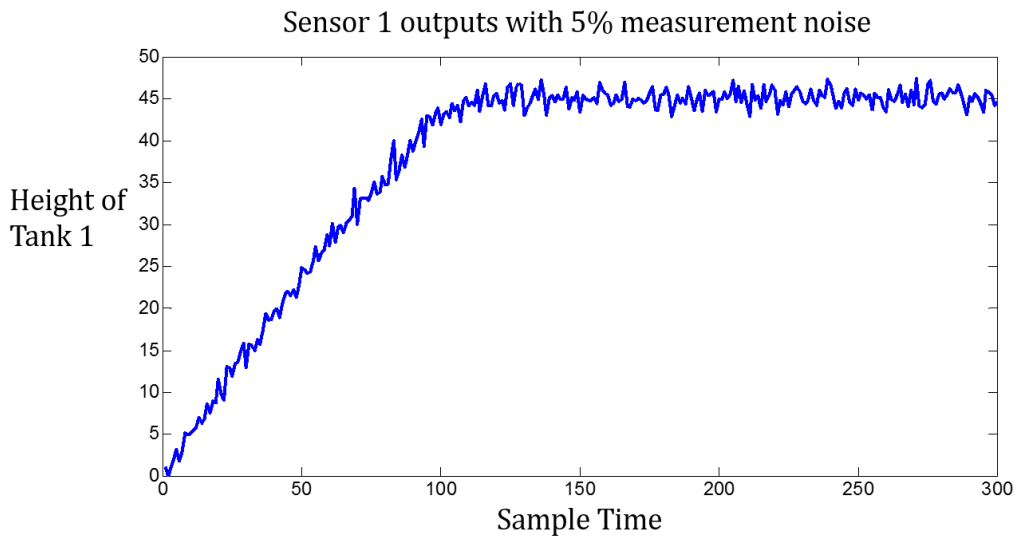


Figure 23. Training data - sensor 1 output with 5% measurement noise.

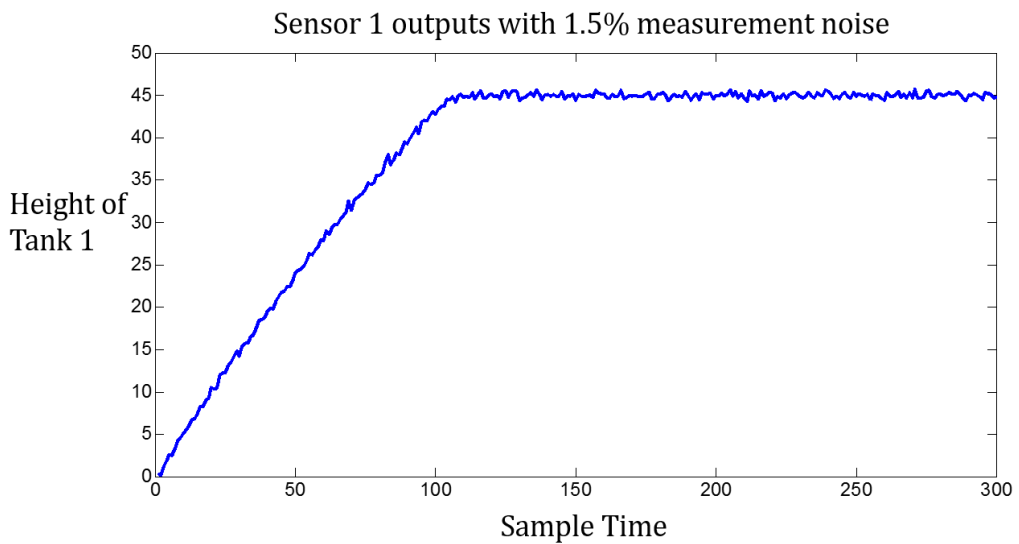


Figure 24. Training data - sensor 1 output with 1.5% measurement noise.

#### **4.1.4 Testing Data**

There are two situations considered for generating testing data: healthy situation and faulty situation. Here, healthy situation means that there is no fault occurring during the system operation. On the other hand, faulty situation means that at least one sensor fault occurs when we run the system model.

##### **4.1.4.1 Healthy (Fault-free) Situation**

In the absence of noise, the testing data should be the same as the training data in the healthy situation. Therefore, the healthy testing data is generated in the same way as the training data except their noise terms are started from different starting seed in Simulink. Accordingly, by subtracting one from another, the result is expected to be a noise signal shown in Fig. 25.

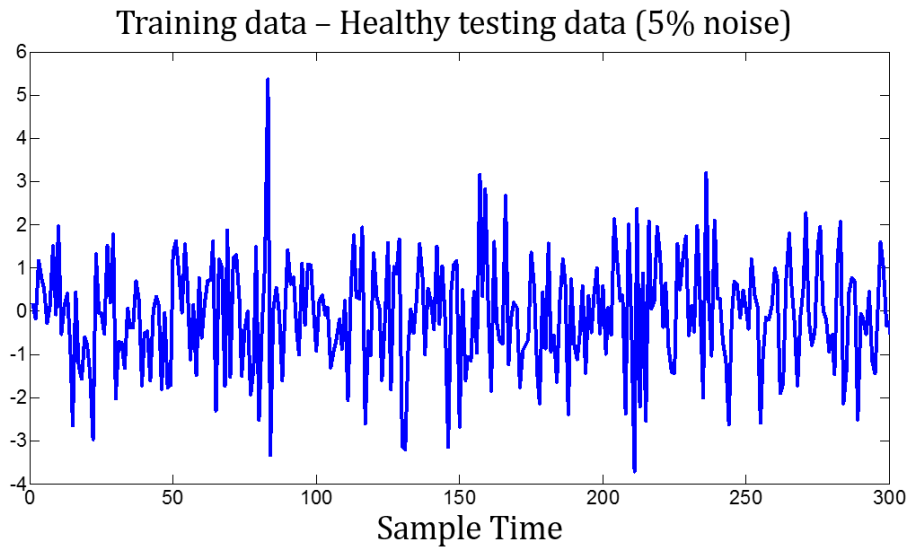


Figure 25. The result of subtracting healthy testing data from training data in the case of 5% noise.

#### 4.1.4.2 Faulty Situation

Generally, a sensor fault will lead the faulty sensor loses its accuracy. For modeling sensor faults, they are used to be modeled as additive [4]. In the three tank system, each tank is equipped with a piezo-resistive pressure transducer for measuring the level of liquid [65]. By using the piezo-resistive material, small potential difference will be generated when the material is deforming by pressure. Then, after amplifying by a sensor amplifier, a potentiometer is used for scaling the voltage coming from the sensor amplifier to generate the measurement output [65]. A sensor fault may occur due to the fatigue of the piezo-resistive material or the failure of any part in the output generating mechanism [65].

In our simulation, there are two scenarios, abrupt and incipient, simulated according to the benchmark for testing FDI systems proposed by Koscielny et al. [66]. An abrupt sensor fault means that the effect of the sensor fault develops rapidly. Here, we simulate an abrupt sensor fault as a bias, and the magnitude of the bias is set to be 20% of the maximum output of the sensor in the fault-free situation. On the other hand, an incipient sensor fault means that the effect of the sensor fault develops slowly. In our simulation, an incipient fault is assumed to develop uniformly and simulated by a bias with 100 samples developing time. The magnitude of the bias is also set to be 20% of the maximum output of the sensor in the fault-free situation. In addition, the occurring time of a sensor fault is set to be sample time 51 for both abrupt and incipient cases. Figure 26 is the sensor 1 measurements in the cases of abrupt fault, incipient fault, and fault-free with 5% noise level.

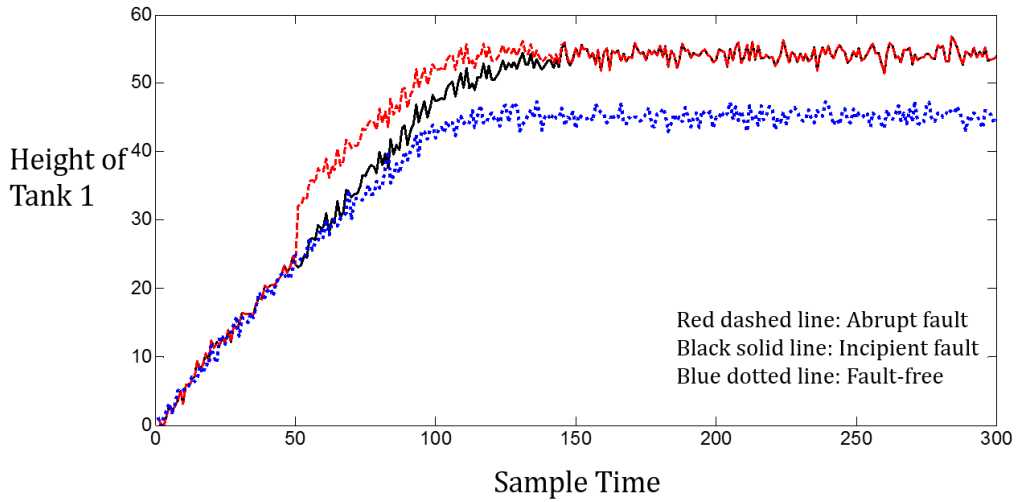


Figure 26. The comparison of abrupt fault, incipient fault, and fault-free cases with 5% noise level.

In Fig. 26, the red dashed line represents the case of abrupt fault, the black solid line represents the case of incipient fault, and the blue dotted line is the fault-free case. Observing the red dashed line, the height of tank 1 suddenly increases when the sensor fault occurs in the case of abrupt fault. On the other hand, in the case of incipient fault, the height of tank 1 does not have similar feature as in the case of abrupt fault. Instead, the magnitude of the incipient fault develops gradually.

#### 4.2 Results of the Blocks of Phase Space Reconstruction and Smooth Mechanism

In this section, the results phase space reconstruction and smooth mechanism will be presented. After collecting data, the first step is to reconstruct the phase space from the collected training data. Here, we follow the steps mentioned in the section 3.1. Figure 27



is the results of applying the mutual information method to the collected training data with 5% noise. According to the mutual information method, the time delay for reconstructing should be chosen to be the value which the first local minimum occurs. Therefore, the time delay is set to be 3.

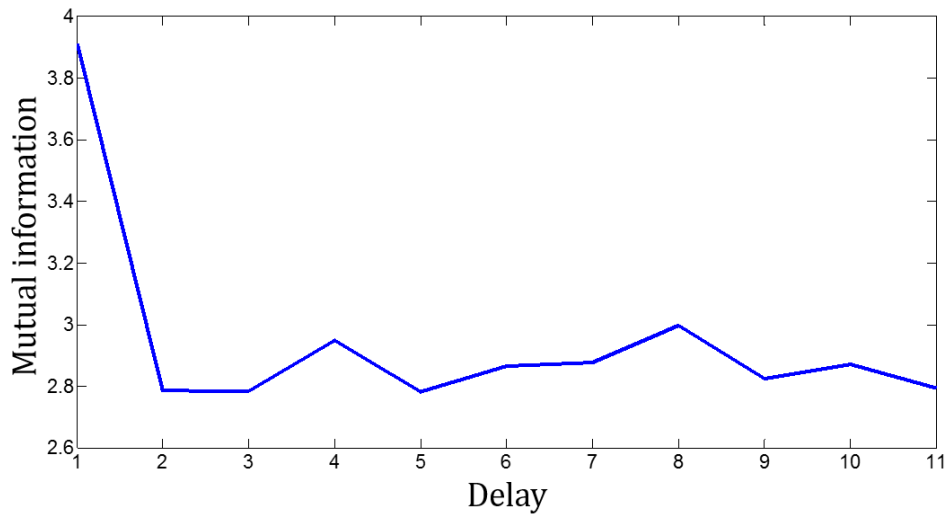


Figure 27. Reconstruct phase space by training data – Sensor 1 output with 5% noise - Mutual information.

For determining the dimension of embedding, the FNN method is applied to the collected training data with 5% noise, and its result is shown in Fig. 28. From this result, one can find that the FNN values is close or equal to zero when the dimension is larger or equal to 2. Therefore, we choose the dimension to be 2. Once the time delay and the dimension of embedding are determined, one can obtain the reconstructed phase space shown in Fig. 29.

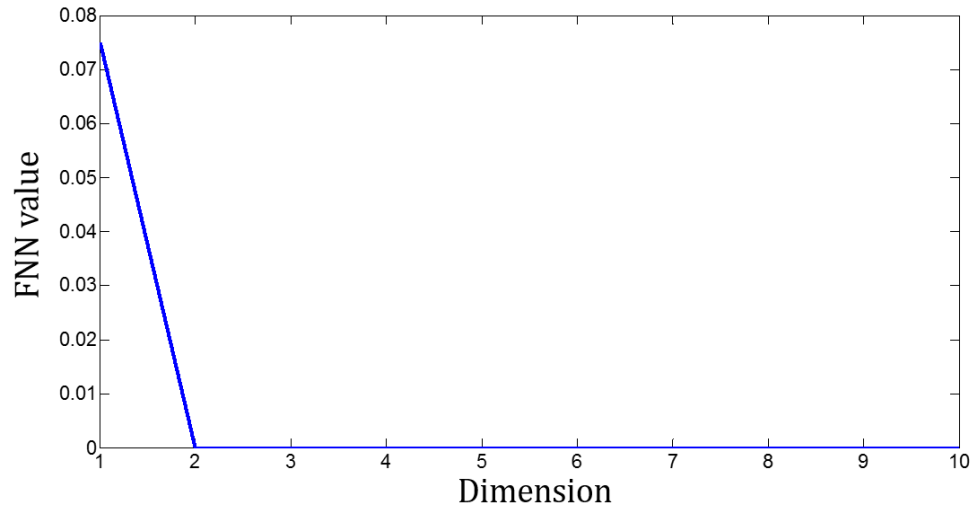


Figure 28. Reconstruct phase space by training data – Sensor 1 output with 5% noise - FNN.

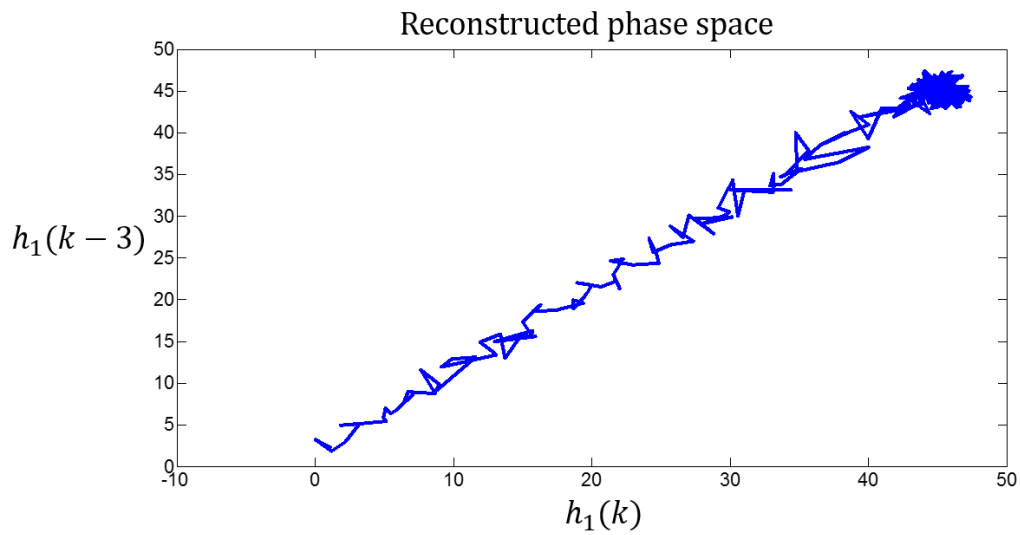


Figure 29. The reconstructed phase space of the sensor 1 in the three tank system.

Observing Fig. 29, the reconstructed trajectory is not smooth and contaminated with fluctuations. In order to obtain better performance in the later blocks, the next step is to smooth the trajectory by the smooth mechanism. Figure 30 is the result of applying MVU estimator to smooth the reconstructed trajectory.

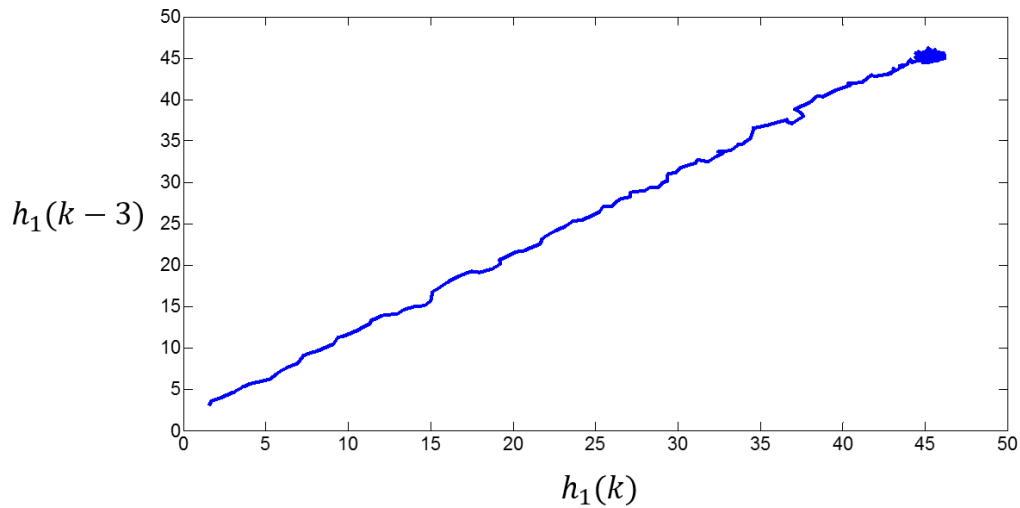


Figure 30. The smoothed trajectory by using MVU estimator.

The other way to smooth the reconstructed trajectory is by using SOD. Figure 31 is the magnitude of the generalized eigenvalues for the smooth orthogonal decomposition. In Fig. 31, the horizontal axis represents the  $i^{\text{th}}$  generalized eigenvalue,  $i = 1, 2$ , and the vertical axis represents the magnitude. According to the result, one can find that the magnitude of the second generalized eigenvalue is much larger than the first eigenvalue. Therefore, we compose the  $\Phi$  in Eq. (79) by the generalized eigenvector corresponding to the second generalized eigenvalue. The smoothed trajectory is shown in Fig. 30. Notice

that, unlike MVU estimator, the smoothed trajectory by using SOD is 1-D shown as Fig. 32. This is because smoothing by SOD deals with the problem of fluctuations in the manner of dimension reduction, and one of the original two dimensions is considered as trivial.

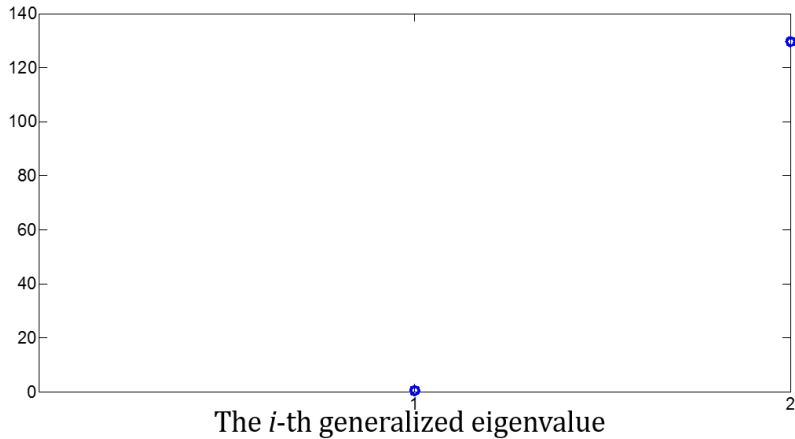


Figure 31. The magnitude of the generalized eigenvalues for the smooth orthogonal decomposition.

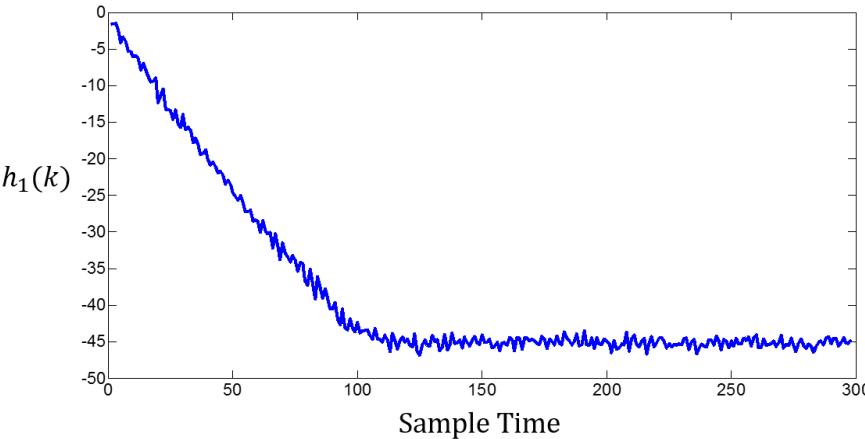


Figure 32. The smoothed trajectory by using SOD.

### **4.3 Results of the Block of Prediction Mechanism**

This block is aimed to generate the predicted future measurements by using the smoothed trajectories obtained from the block of smooth mechanism. The output of this block is called raw residual signals which is produced by taking the absolute value of the difference between the predicted measurements and actual measured measurements. Figure 33, 34, and 35 are the raw residual signals in the fault-free, abrupt fault, and incipient fault situations with 5% noise produced by using the MVU estimator smoothed trajectory for prediction. From these results, one can find that the magnitude of the raw residual signal in the fault-free situation keeps low. In the abrupt fault situation, the magnitude of the raw residual signal rapidly increases at the sample time 51. This phenomenon means that the proposed method indicates there is a sensor fault occurring at that moment. In the incipient fault situation, the raw residual signal increases slowly from the sample time 51. In this case, it is hard to claim the applied system is faulty or healthy. Therefore, further process is needed to make these raw residual signals distinguished to each other.

Fault-free - Raw residual signal:  $|measured\ value - predicted\ value|$

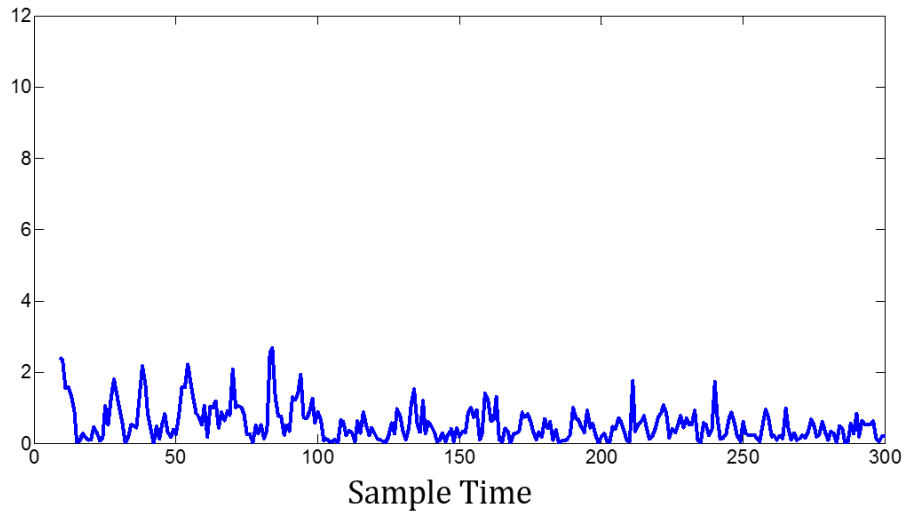


Figure 33. Raw residual signal in the fault-free situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.

Abrupt fault - Raw residual signal

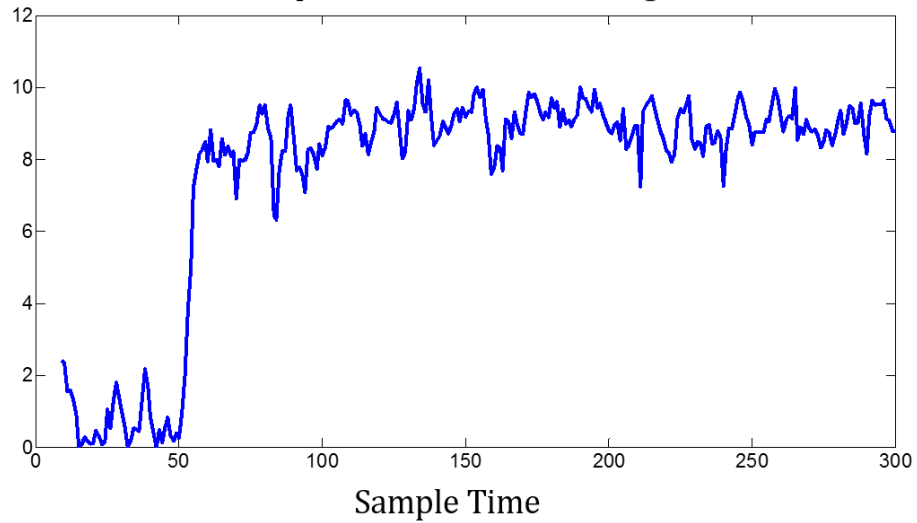


Figure 34. Raw residual signal in the abrupt fault situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.

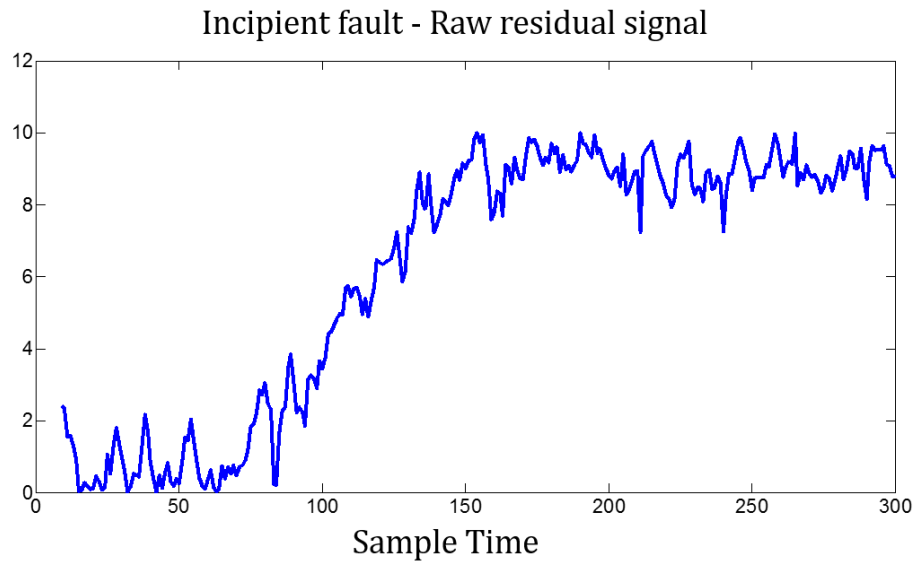


Figure 35. Raw residual signal in the incipient fault situation with 5% noise produced by using the MVU estimator smoothed trajectory for prediction.

On the other hand, Fig. 36, 37, and 38 are the results of using SOD smoothed trajectory for prediction. Observing these results, they have similar trend as former results. In order to compare the performance of MVU estimator and SOD, we calculate the variance of the raw residual signals produced by using MVU estimator and SOD in fault-free situation. We expect that the one with better performance in smoothing produces the raw residual signal with lower variance. The reason is that, ideally, if the effect of fluctuations can be totally reduced by the smooth mechanism, the predicted measurements will be the same as the measured measurements in the fault-free situation, and the raw residual signal will be all zero. Moreover, the smooth mechanism is aimed to improve the performance of the prediction mechanism, and the raw residual with lower variance means

it is closer to the ideal result. Therefore, the smooth mechanism which produces the raw residual signal with lower variance will be considered to be better. **Table 2** is the calculated variance of the raw residual signals in fault-free situation with three different noise levels. From this table, in the case of 1.5% noise level, the performance of the prediction mechanism is acceptable even though the reconstructed trajectory is not smoothed. This means 1.5% noise is small enough to be neglected while using the proposed method. When the noise level is high, i.e. 5% and 10% noise, smoothing the reconstructed trajectory obviously improves the performance of the prediction mechanism. In addition, according to the results, SOD is more robust for all noise level, and MVU estimator has better performance when the noise level is low. Although we can improve the performance of MVU estimator by enlarging the size of the averaging frame in Eq. (76), this action will violate the assumption about  $A(k)$  in Eq. (75)<sup>12</sup>. Therefore, SOD is more recommended when the noise level is unknown. In addition, as mentioned in section 4.2, using SOD for smoothing is more storage memory efficient. **Table 3** is the dimension of smoothed trajectories with three different noise levels. From the table, the smoothed trajectory is all 1-D for all noise levels by using SOD. This is the other reason to more recommend SOD than MVU estimator. Therefore, in following sections, the results are generated by using SOD as the smooth mechanism.

---

<sup>12</sup> Let the length of the averaging frame is  $L$ . We assume  $L$  is small enough and the true value,  $A(k)$ , can be considered as constant in the  $L$ -samples averaging frame.



Table 2. Calculated variance of the raw residual signals in fault-free situation.

	1.5%	5%	10%
No smooth	0.2828	2.1686	16.1487
MVU ( $L = 5$ )	0.2701	0.5564	6.964
SOD	1.0852	1.6226	2.9675

Table 3. The dimension of smoothed trajectories with three different noise levels.

	1.5%	5%	10%
MVU	2	2	3
SOD	1	1	1

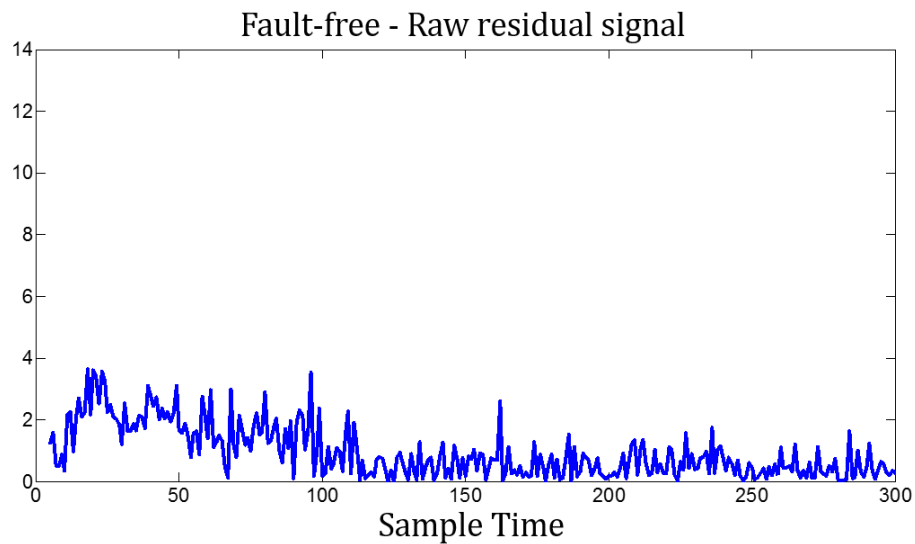


Figure 36. Raw residual signal in the fault-free situation with 5% noise produced by using the SOD smoothed trajectory for prediction.

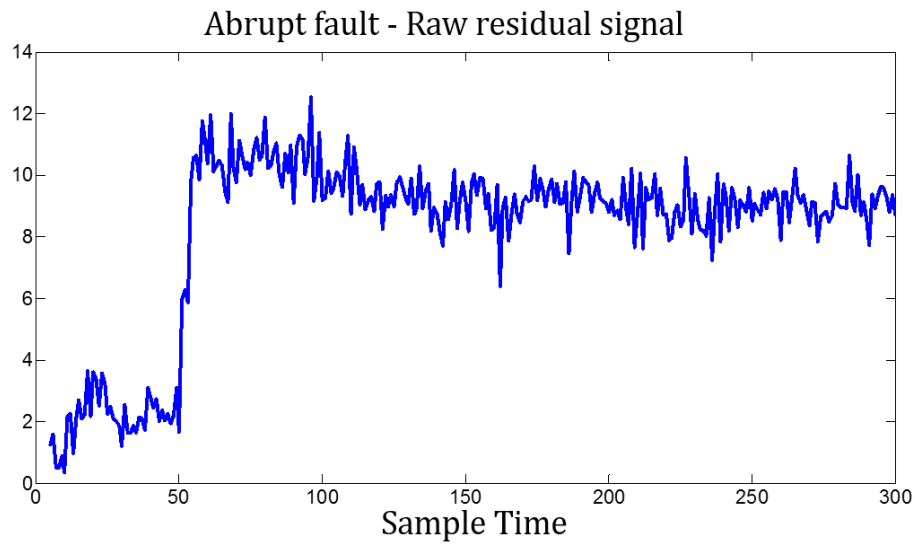


Figure 37. Raw residual signal in the abrupt fault situation with 5% noise produced by using the SOD smoothed trajectory for prediction.

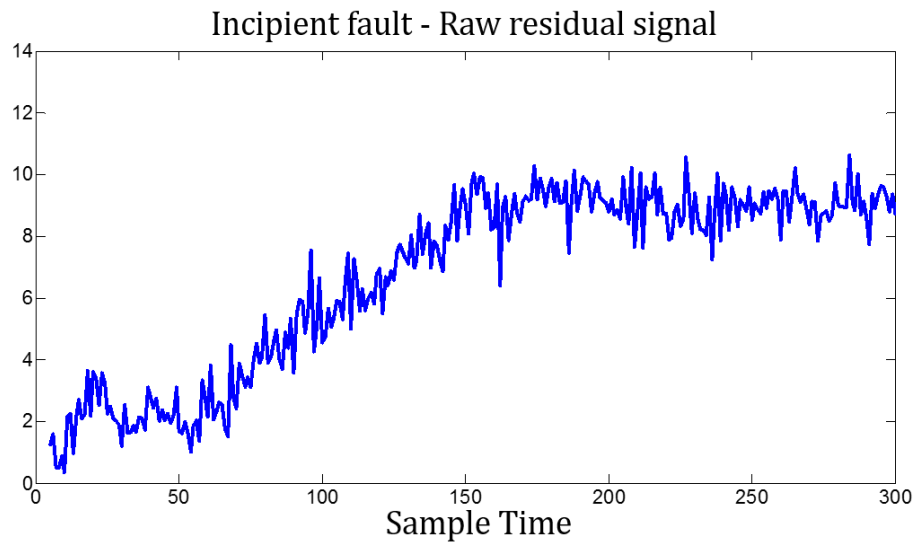


Figure 38. Raw residual signal in the incipient fault situation with 5% noise produced by using the SOD smoothed trajectory for prediction.

#### 4.4 Residual Signal Processing

In this section, the raw residual signals are further processed in order to distinguish the fault-free and faulty situations. Normally, a threshold is chosen for the purpose. When the magnitude of the raw residual signal is larger than the threshold, the applied system will be considered to be faulty, and the residual signal is set to be one. Otherwise, the applied system will be considered to be healthy, and the residual signal is set to be zero. Here, the threshold is set to be twice of the standard deviation of the raw residual signals in fault-free situation. **Table 4** is the thresholds for three different noise levels. Figure 39, 40, and 41 are the residual signals generated by setting a threshold in the fault-free, abrupt fault, and incipient fault situations with 1.5% noise. According to these results, the proposed method works well. The residual signal is all zero in the fault-free situation. In the abrupt fault situation, the residual signal correctly indicate the abrupt fault occurs at sample time 51. In the incipient fault situation, the magnitude of the incipient fault is considered to be large enough to claim the applied system is faulty around sample time 65. In addition, there is no false alarm or miss detection in all situations with 1.5% noise.

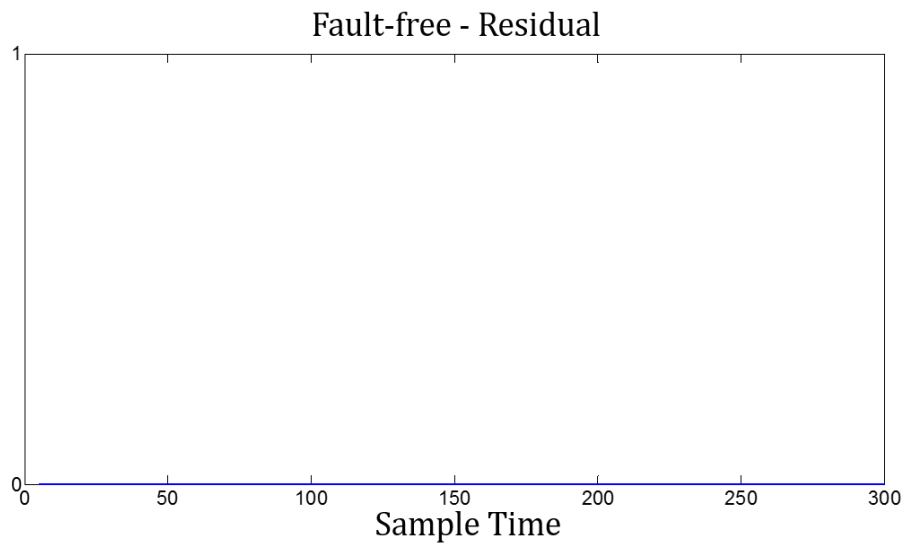


Figure 39. Residual signal generated by setting a threshold in the fault-free situation with 1.5% noise.

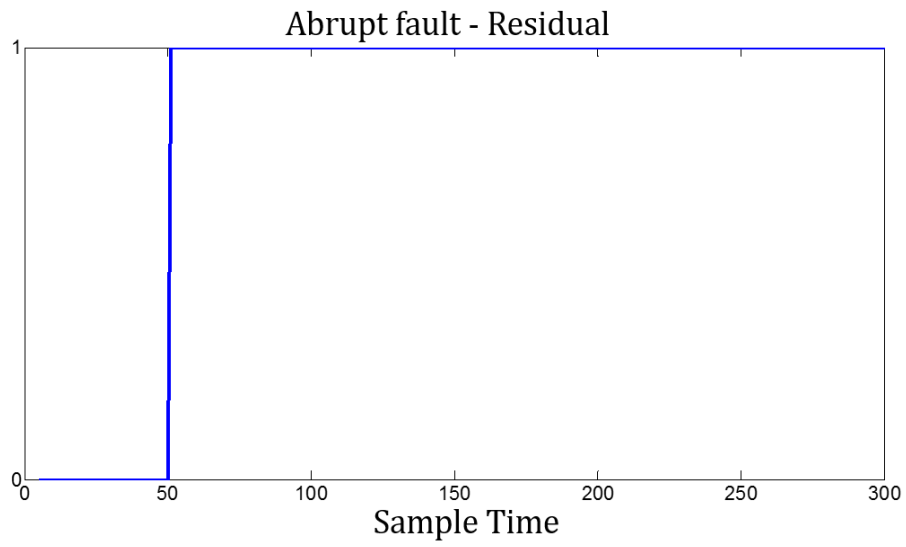


Figure 40. Residual signal generated by setting a threshold in the abrupt fault situation with 1.5% noise.

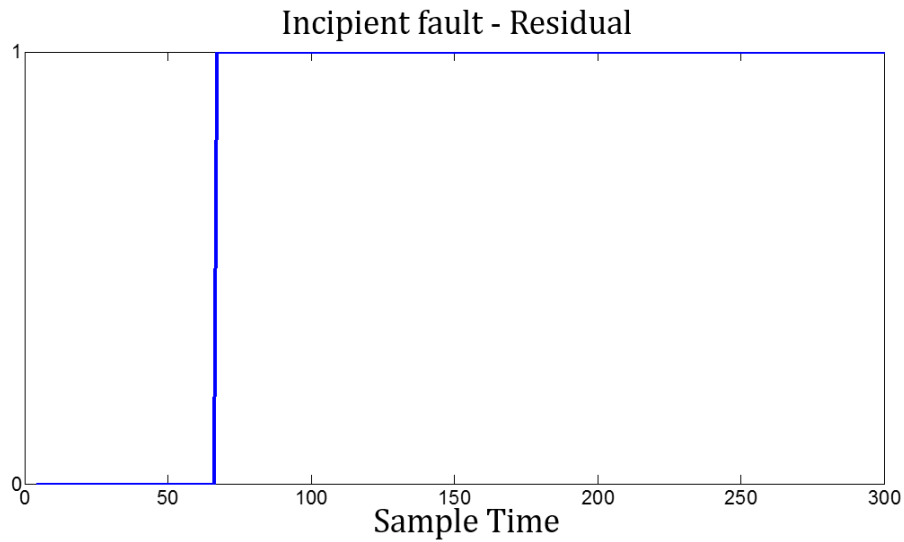


Figure 41. Residual signal generated by setting a threshold in the incipient fault situation with 1.5% noise.

Table 4. The thresholds for three different noise levels.

	1.5%	5%	10%
Threshold	2.5032	3.5734	3.9147

Until now, setting a threshold works well. However, the problem of false alarm and miss detection occurs when the noise level increases. Figure 42, 43, and 44 are the residual signals generated by setting a threshold in the fault-free, abrupt fault, and incipient fault situations with 5% noise. According to these results, in the fault-free and abrupt fault situation, there is no false alarm and miss detection, and the abrupt sensor fault is detected correctly. However, in the incipient fault situation, there are several peaks appears before the magnitude of the residual signal stays in one, and it is difficult to determine that those

peaks appear due to the sensor fault or false alarms. This problem is worse when the noise level increases from 5% to 10%. Figure 45, 46, and 47 are the results for 10% noise level. According to these results, false alarm and miss detection occur in all situations. These results are reasonable. Since the threshold is chosen in a statistical way, its reliability decreases when the variance of noise increases. Therefore, it may be more objective to show residual signals by probability instead of a binary decision, i.e. faulty/healthy, and the Multiple Hypothesis Shiriyayev Sequential Probability Test (MHSSPT) is introduced for this purpose.

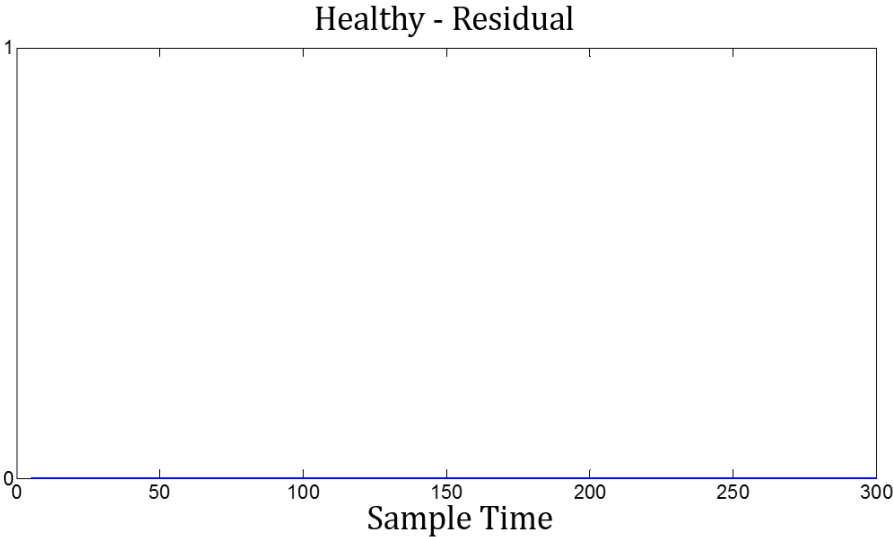


Figure 42. Residual signal generated by setting a threshold in the fault-free situation with 5% noise.

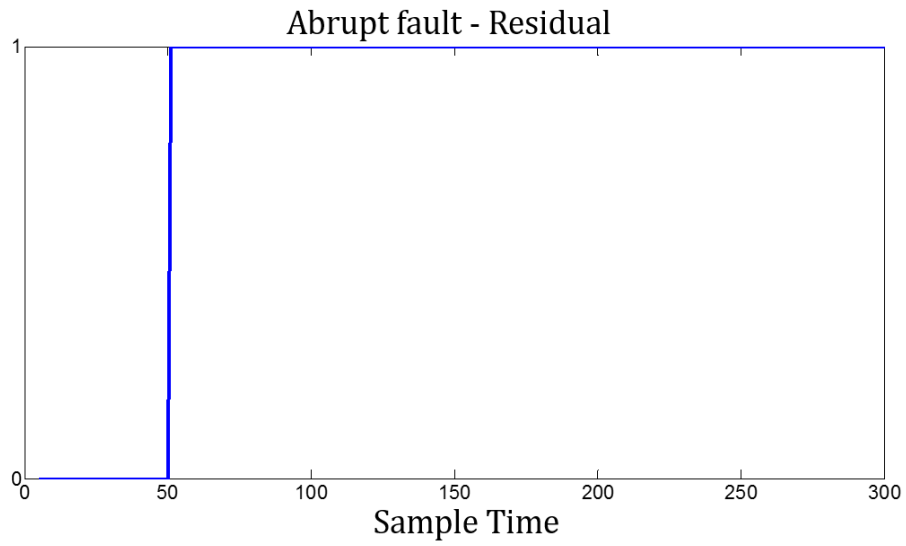


Figure 43. Residual signal generated by setting a threshold in the abrupt fault situation with 5% noise.

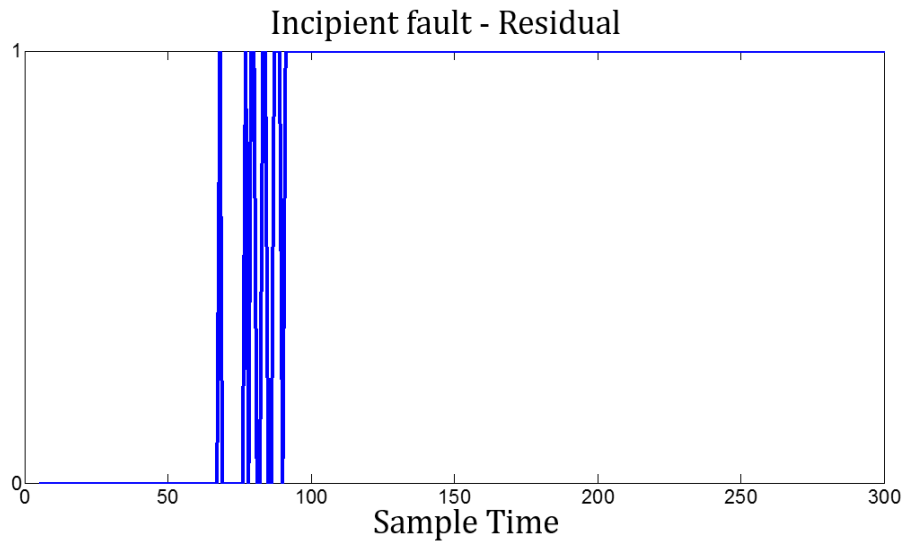


Figure 44. Residual signal generated by setting a threshold in the incipient fault situation with 5% noise.

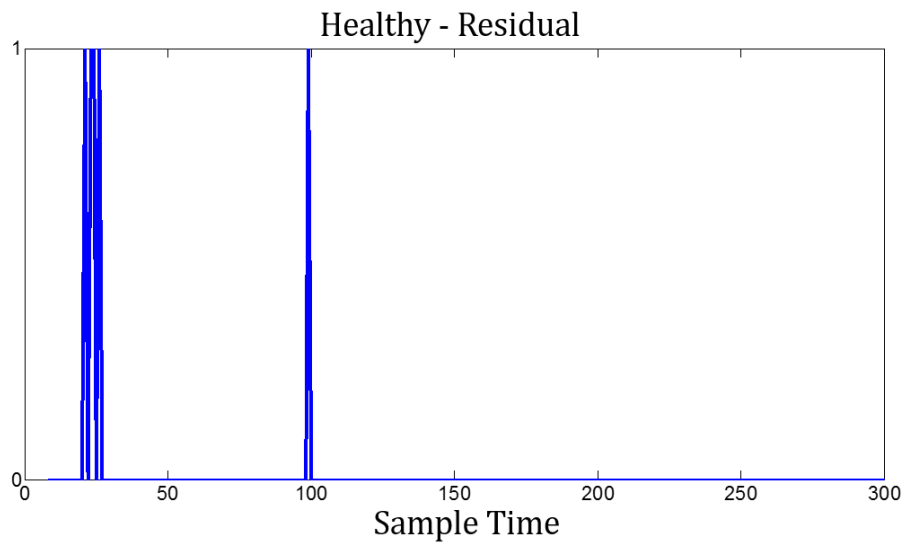


Figure 45. Residual signal generated by setting a threshold in the fault-free situation with 10% noise.

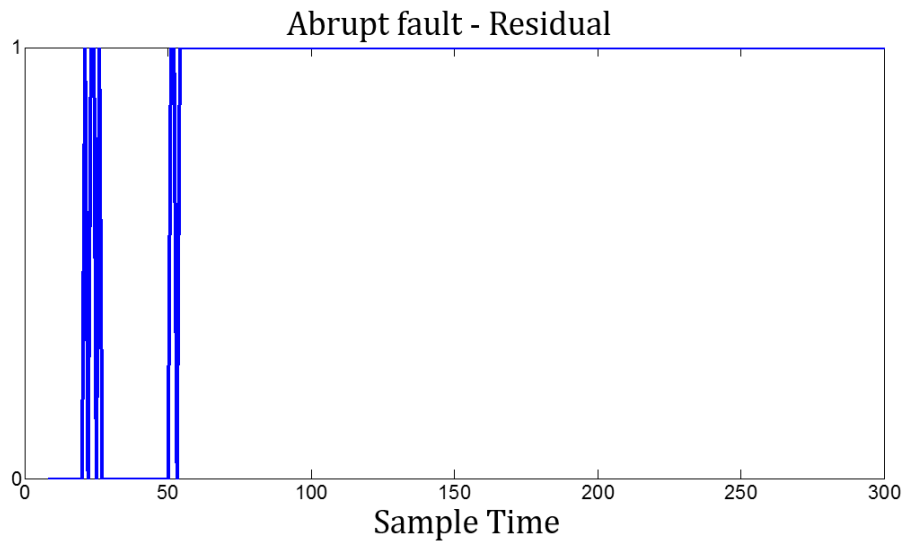


Figure 46. Residual signal generated by setting a threshold in the abrupt fault situation with 10% noise.



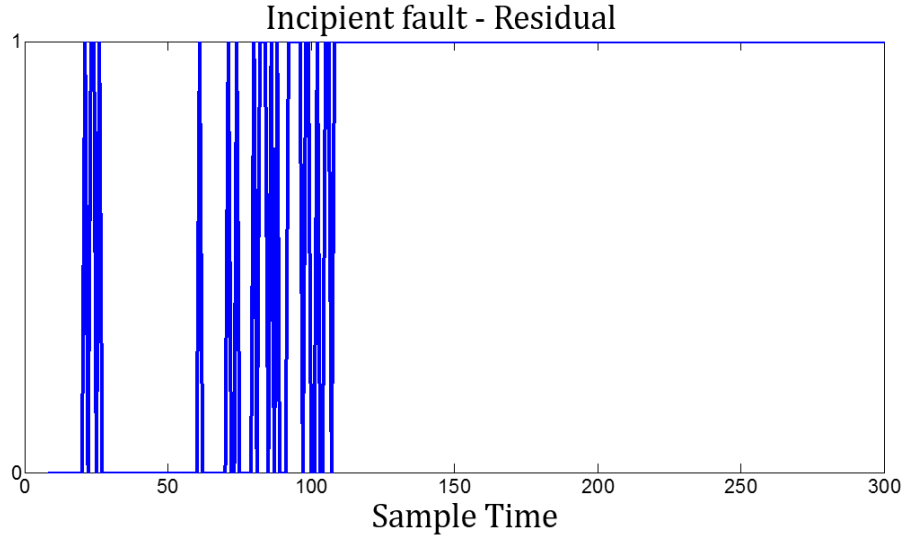


Figure 47. Residual signal generated by setting a threshold in the incipient fault situation with 10% noise.

#### 4.4.1 Multiple Hypothesis Shirayev Sequential Probability Test (MHSSPT)

Assume there are two hypotheses:  $\mathcal{H}_0$  means the situation is healthy, and  $\mathcal{H}_1$  means the applied system is faulty. In essence, MHSSPT is to calculate the posterior probability of  $\mathcal{H}_1$  conditioned on the measurement sequence [67, 68]. Eq. (95.a-c) are the propagation equations for the posterior probability of  $\mathcal{H}_1$  [67].

$$P(\theta_1 \leq t_{k+1} | X_{k+1}) = \frac{P(\theta_1 \leq t_{k+1} | X_k) f_1(x_{k+1})}{\sum_{j=0}^1 P(\theta_j \leq t_{k+1} | X_k) f_1(x_{k+1})} \quad (95.a)$$

$$P(\theta_1 \leq t_{k+1} | X_k) = P(\theta_1 \leq t_k | X_k) + \widetilde{p}_1 [1 - P(\theta_1 \leq t_k | X_k)] \quad (95.b)$$

$$P(\theta_0 \leq t_{k+1}|X_k) = 1 - P(\theta_1 \leq t_{k+1}|X_k) \quad (95.c)$$

Where  $\theta_i$ ,  $i = 0,1$ , is the time of transition to the hypothesis  $\mathcal{H}_i$ ,  $x_k$  is the residual obtained at time  $t_k$ ,  $X_k$  is the sequence of residual obtained up to time  $t_k$ ,  $f_i(x_k)$ ,  $i = 0,1$ , is the probability density function of  $x_k$  given the hypothesis  $\mathcal{H}_i$ , and  $\tilde{p}_1$  is the **a priori probability** of transition hypothesis  $\mathcal{H}_1$  from  $t_k$  to  $t_{k+1}$ . The propagation equation is worked under three assumptions [68]. First, the measurement sequence  $X_k$  is assumed to be conditionally independent, i.e.  $P(\theta_i \leq t_k|X_k) = P(\theta_i \leq t_k|x_k)P(\theta_i \leq t_k|x_{k-1}) \cdots P(\theta_i \leq t_k|x_1)$ . Second, the measurements under different hypothesis are assumed to be Gaussian distributed,  $x \sim N(m_i, \Lambda_i)$ ,  $i = 0,1$ , and its mean value is assumed to be uniform distributed,  $m_i \sim Unif[b_i, b_i + 2m_i^*]$ . Under the second assumption, Malladi et al. proves that the probability density function,  $f_i(x)$ , can be calculated by [68]

$$f_i(x) = \left(1/4m_i^*\right) \left[ erf \left\{ 1/\sqrt{2} \Lambda_i^{-1/2} (x - b_i) \right\} - erf \left\{ 1/\sqrt{2} \Lambda_i^{-1/2} (x - b_i - 2m_i^*) \right\} \right] . \quad (96)$$

Where the Gauss error function is defined as

$$erf(*) = 2/\sqrt{\pi} \int_0^* e^{-t^2} dt. \quad (97)$$

Third, the a priori probability of transition,  $\widetilde{p}_1$ , is assumed known and constant for all time. In our simulation,  $\widetilde{p}_1$  is set to be  $10^{-8}$  for 5% noise level and  $10^{-9}$  for 10% noise level.

In order to set the initial conditions for the propagation equation, we further assume that the applied system is healthy at the beginning, so the initial conditions are set to be  $P(\theta_0 \leq t_0 | X_0) = 1$  and  $P(\theta_1 \leq t_0 | X_0) = 0$ . Moreover, for the healthy hypothesis,  $m_0^*$ ,  $b_0$ , and  $\Lambda_0$  are obtained from the raw residual signal produced by generating another set of training data and using it as a healthy testing data shown as Fig. 48. In addition, for the faulty hypothesis,  $b_1$  is set to be  $b_0 + 2m_0^*$ ,  $m_1^*$  is obtained by guessing the maximum magnitude of fault, and  $\Lambda_1$  is set to be equal to  $\Lambda_0$  for the same noise level.

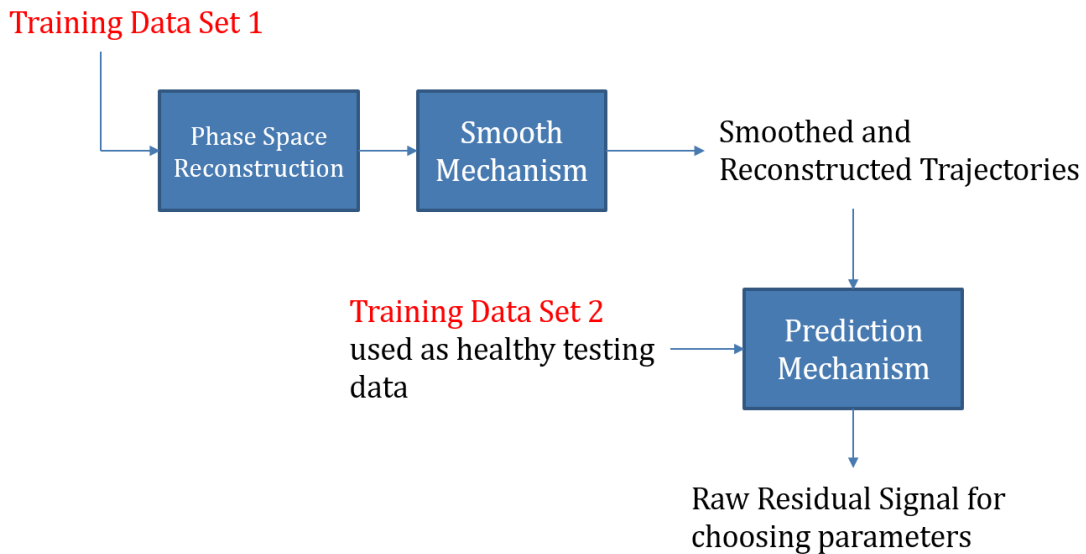


Figure 48. The illustration of using the second set of training data to choose parameters for the MHSSPT.

Figure 49, 50, and 51 are the probability of the occurrence of fault in the fault-free, abrupt fault, and incipient fault situations with 5% noise calculated by MHSSPT. Figure 52, 53, and 54 are the results for 10% noise level. Observing these results, the problem of false alarm and miss detection is well handled by showing the residual signals in a probabilistic way. In the abrupt fault situation, the residual signal increases rapidly almost right after the sensor fault occurring at sample time 51. In addition, for the incipient fault, the residual signal indicates the fault when its magnitude increases to certain level. Moreover, for the fault-free situation, the residual signal keeps low and closes to zero for all time. Therefore, our results suggests that the proposed method is capable to detect both of abrupt and incipient sensor fault correctly and well handle the problem of false alarm and miss detection.

The cost is that the MHSSPT algorithm will take few sample time to confirm the applied system is really faulty. In the other words, the MHSSPT algorithm needs few sample time to transfer the hypothesis from  $\mathcal{H}_0$  to  $\mathcal{H}_1$ . This fact can be explicitly seen in the abrupt fault situation. The probability of the occurrence of fault does not indicate immediately when the sensor fault occurs at sample time 51.

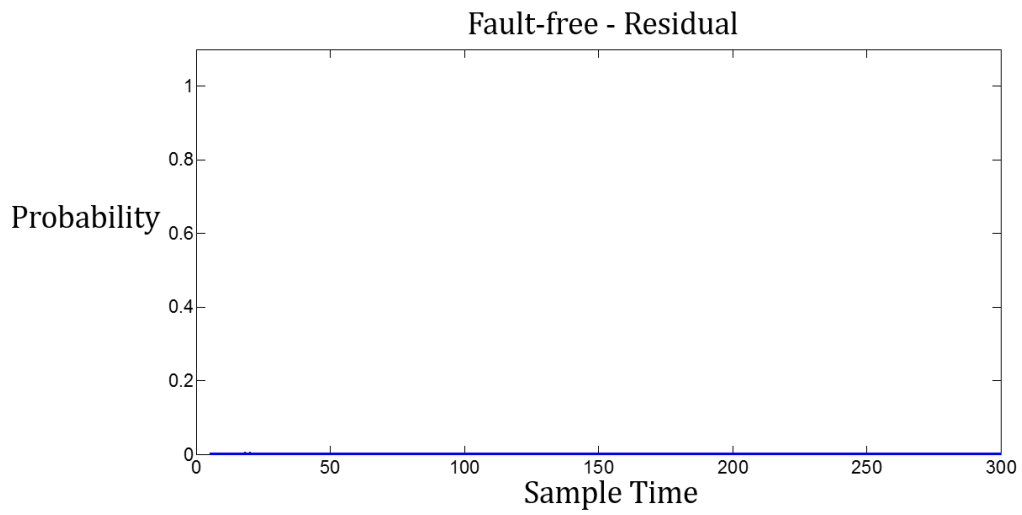


Figure 49. The probability of the occurrence of fault in the fault-free situation with 5% noise.

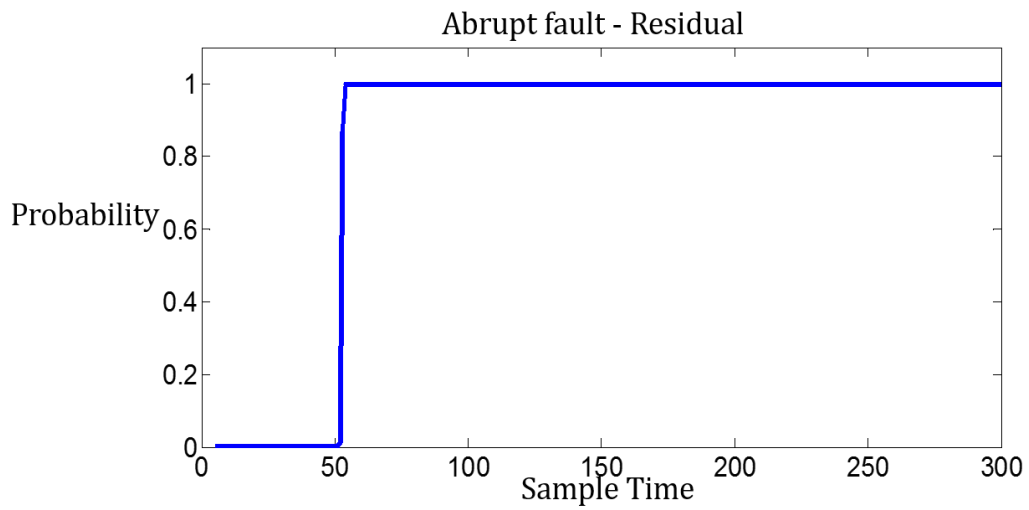


Figure 50. The probability of the occurrence of fault in the abrupt fault situation with 5% noise.

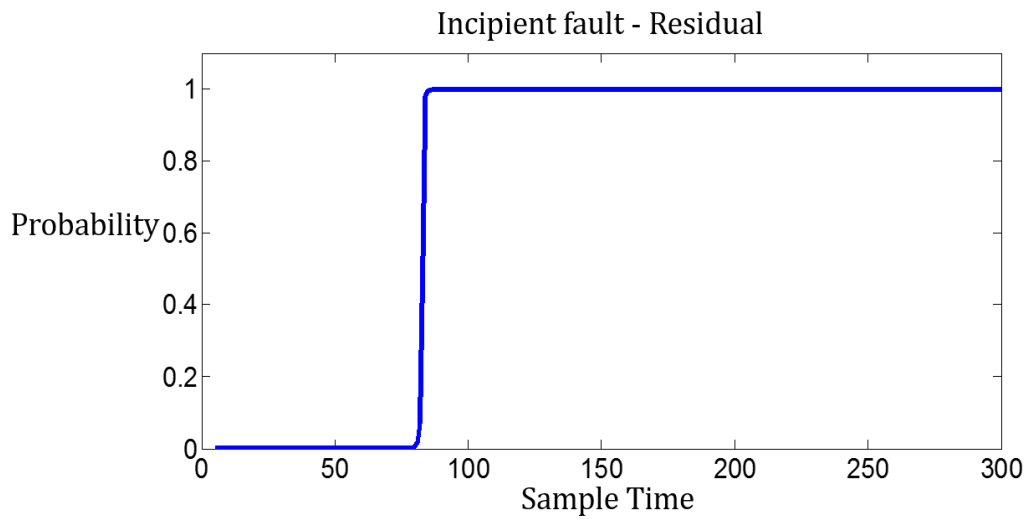


Figure 51. The probability of the occurrence of fault in the incipient fault situation with 5% noise.

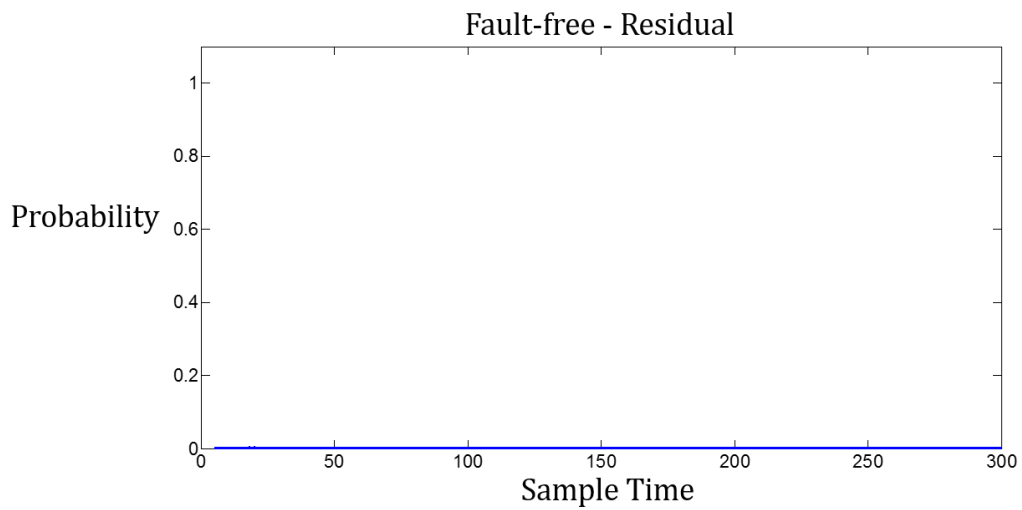


Figure 52. The probability of the occurrence of fault in the fault-free situation with 10% noise.

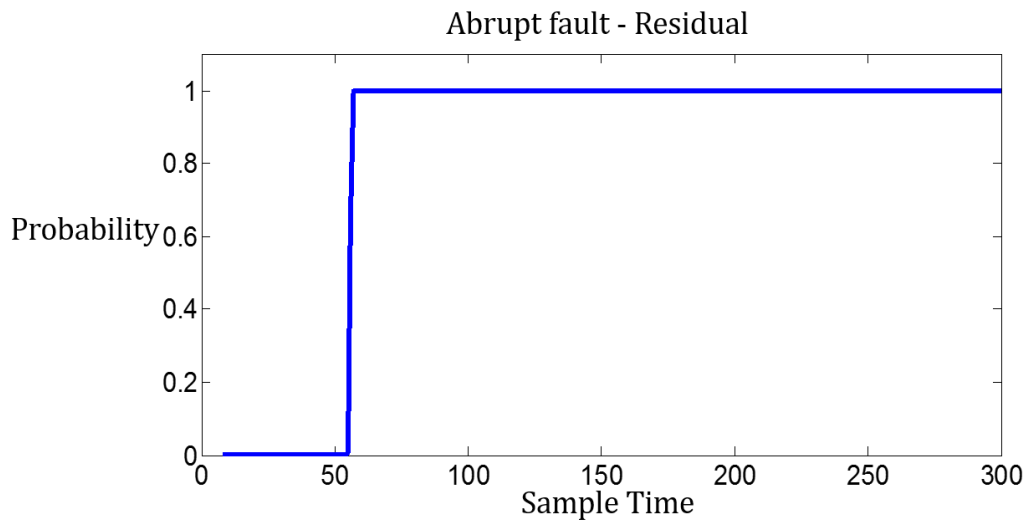


Figure 53. The probability of the occurrence of fault in the abrupt fault situation with 10% noise.

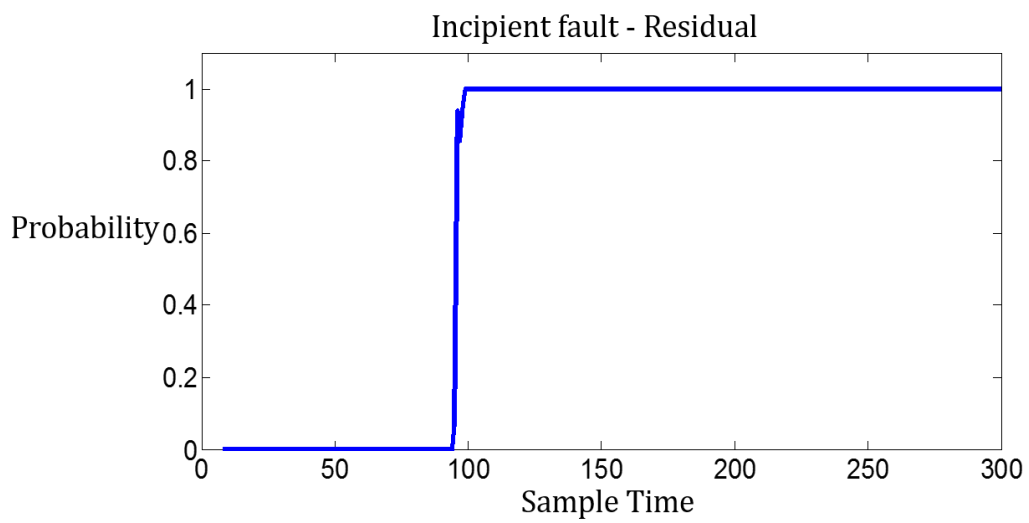


Figure 54. The probability of the occurrence of fault in the incipient fault situation with 10% noise.

#### 4.5 Results of the Block of Conservation in Phase Space

In this section, the results of the extension of the proposed method by the conservation/dissipation property in phase space will be presented. In our simulation, the initial conditions of two reference trajectories are set to be  $(0, 0, 0)$  and  $(20, 20, 20)$ , and the initial condition of the unknown trajectory is set to be  $(10, 10, 10)$ . Their setpoint is all set to be the same as in previous simulations,  $w_1 = 45\text{cm}$  and  $w_2 = 15\text{cm}$ . Figure 55, 56, and 57 are the results of applying the conservation/dissipation property for fault detection and isolation in 5% noise level. According to these results, in the fault-free situation, the generated residual signal indicates the probability of the occurrence of fault is very low and close to zero for all time. In the abrupt fault situation, the probability of the occurrence of fault increases rapidly right after the sensor fault occurs at sample time 51. Moreover, for the incipient sensor fault, the proposed method is also capable to detect the fault when its magnitude grows to a certain level. For 10% noise level, the results for the fault-free, abrupt fault, and incipient fault situations are shown in Fig. 58, 59, and 60 respectively. According to these results, the residual signals response correctly to their corresponding situation. However, the response time is longer in the abrupt fault situation than the same situation with 5% noise. The reason is that when the variance of noise is larger, the MHSSPT algorithm needs more time to confirm the applied system is faulty in order to avoid false alarm. According to our results, the proposed method is capable to detect both abrupt and incipient sensor fault without saving the trajectory corresponded to the initial condition,  $(10, 10, 10)$ .



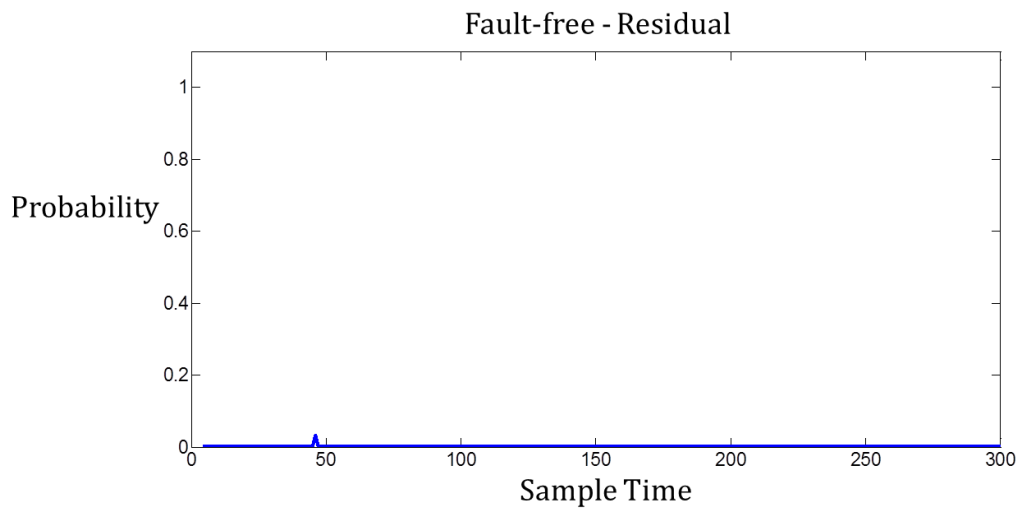


Figure 55. The probability of the occurrence of fault in the fault-free situation with 5% noise - FDI by conservation/dissipation property.

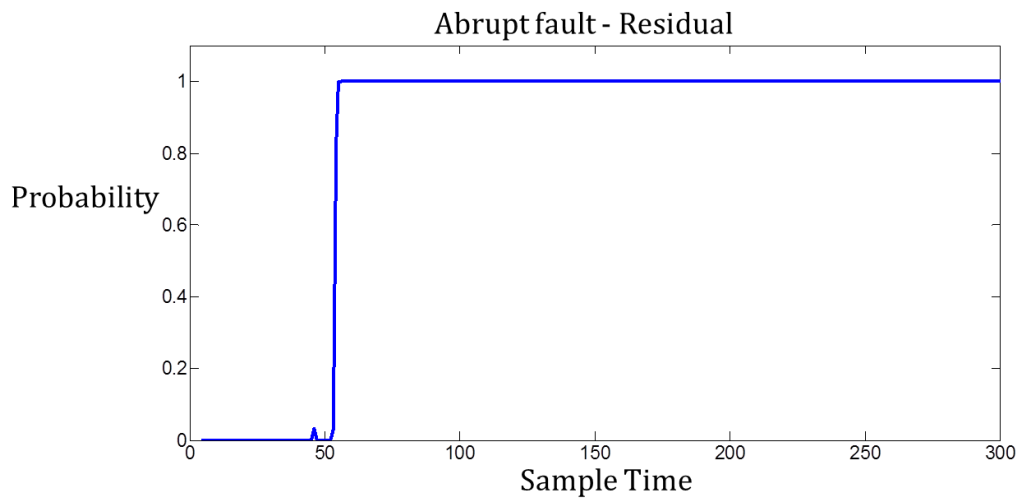


Figure 56. The probability of the occurrence of fault in the abrupt fault situation with 5% noise - FDI by conservation/dissipation property.

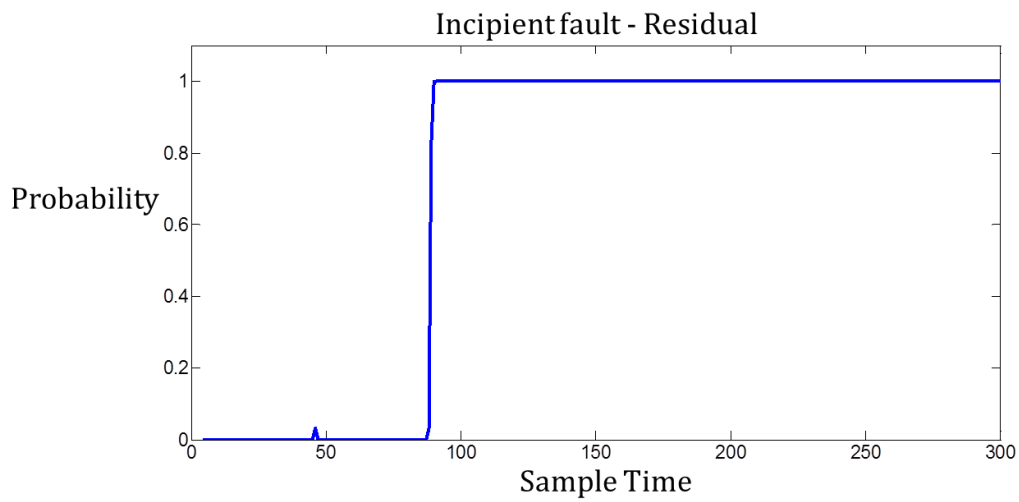


Figure 57. The probability of the occurrence of fault in the incipient fault situation with 5% noise - FDI by conservation/dissipation property.

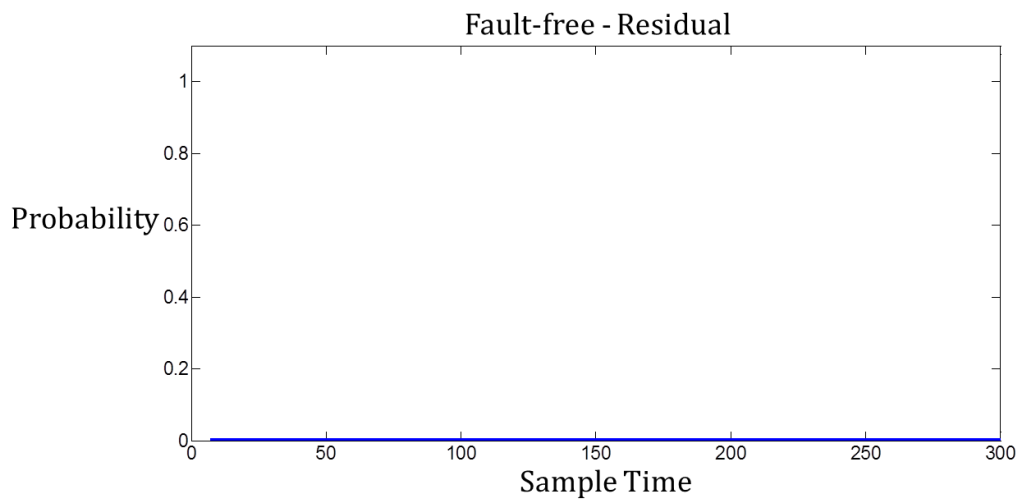


Figure 58. The probability of the occurrence of fault in the fault-free situation with 10% noise - FDI by conservation/dissipation property.

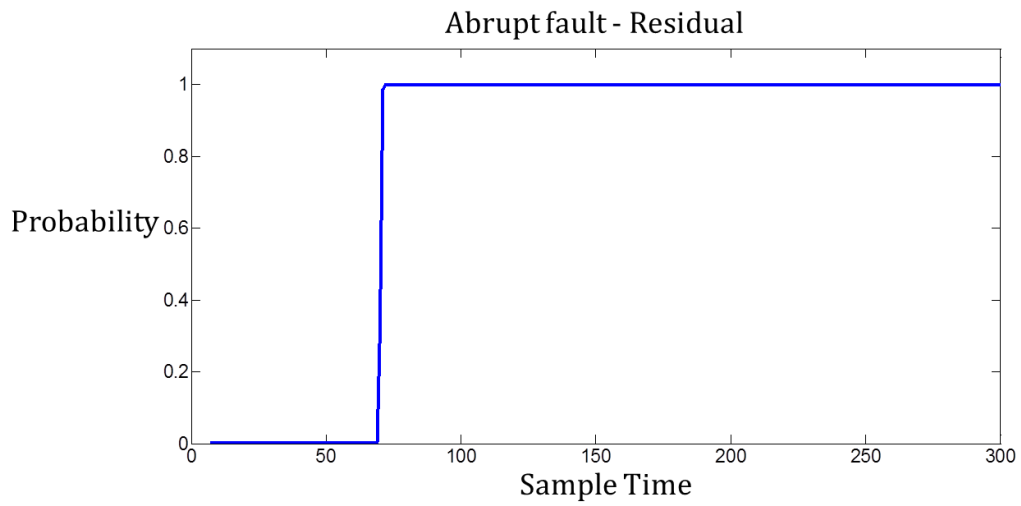


Figure 59. The probability of the occurrence of fault in the abrupt fault situation with 10% noise - FDI by conservation/dissipation property.

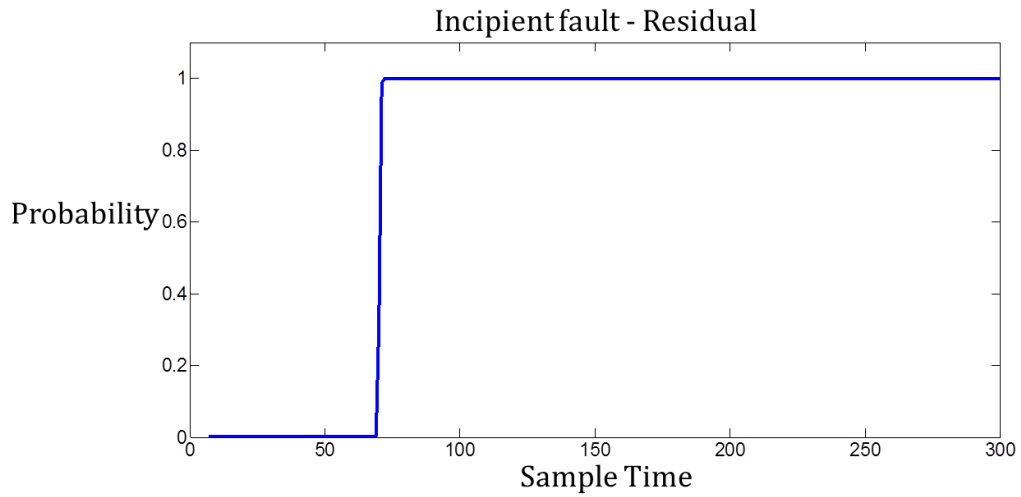


Figure 60. The probability of the occurrence of fault in the incipient fault situation with 10% noise - FDI by conservation/dissipation property.

## 5. EXPERIMENTAL RESULTS

In this section, we use a sugar factory actuator system and a computer controlled heat pump/air condition/refrigeration (THIBAR22C) unit to examine the proposed method.

### 5.1 DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems)

DAMADICS is a benchmark which concerns on an actuator proposed by Koscielny et al.... There are mainly three parts included in the actuator: control valve, pneumatic servo-motor, and positioner [66]. The scheme of the actuator is shown as Fig. 61.

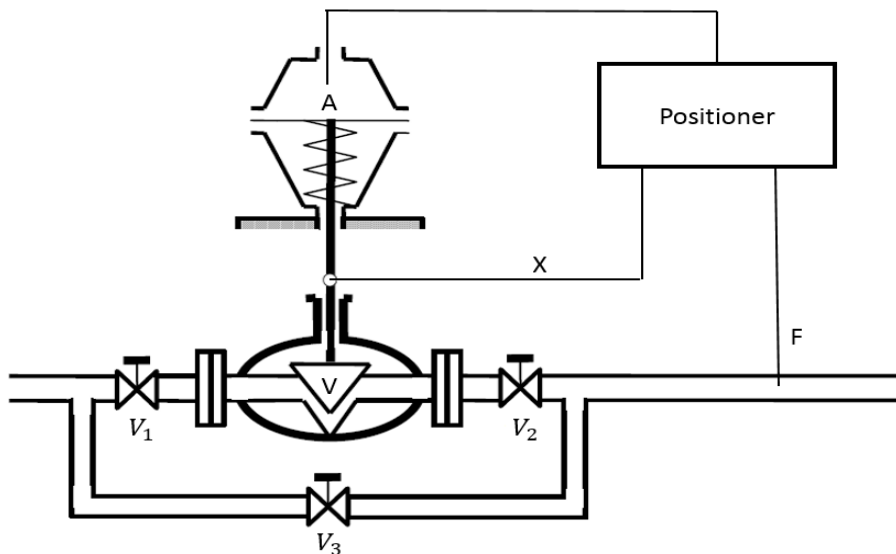


Figure 61. The scheme of the actuator. (Adapted from [66])

In Fig. 61, A is pneumatic servo-motor; V is control valve; X is measured rod displacement; F is measured medium flow rate;  $V_1$ ,  $V_2$ , and  $V_3$  are hand-driven valves. Here, the control valve is used to prevent and/or limit the flow of fluids; the pneumatic servo-motor is an air/fluid powered device used to provide linear motion of its stem; the positioner is used to eliminate the control-valve-stem miss-position due to friction or pressure unbalance [66]. A comprehensive block diagram of the actuator is shown in Fig. 62.

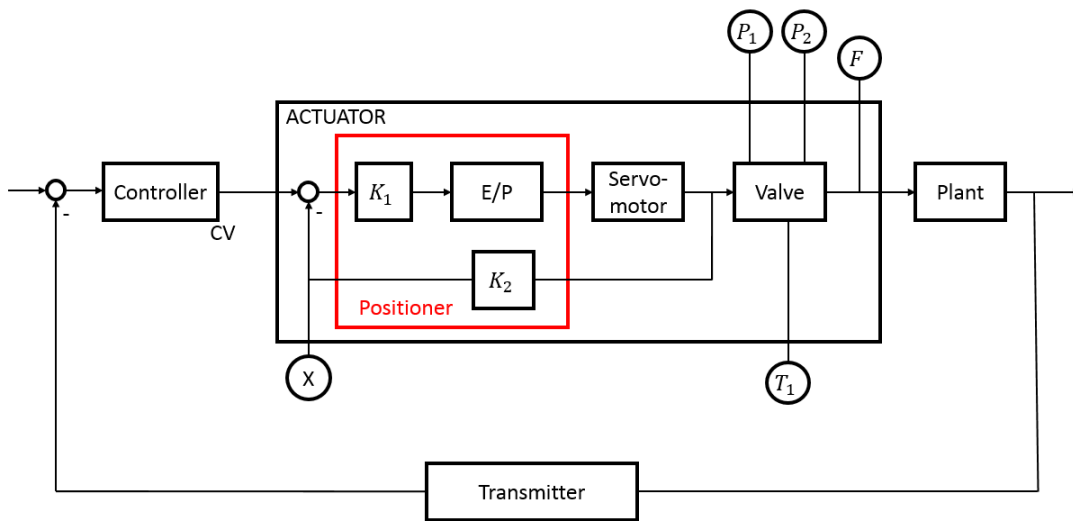


Figure 62. The block diagram of the actuator and its application. (Adapted from [66])

In Fig. 62, E/P represents electro-pneumatic transducer; blocks  $K_1$  and  $K_2$  are adaptors for transferring control input to corresponding input of the electro-pneumatic transducer and the unit of rod displacement to percentage respectively [66]. As shown in

Fig. 62, there are five sensors for monitoring the status of the actuator. The description of these sensors is listed in Table 5.

Table 5. Description of sensor outputs in the actuator system. (Adapted from [66])

Sensor Output	Range	Physical Unit	Transducer range	Description
P1	<0,1>	-	<0,4>[MPa]	The value of the pressure on the control valve inlet.
P2	<0,1>	-	<0,4>[MPa]	The value of the pressure on the control valve outlet.
T1	<0,1>	-	<0,200>[°C]	The medium temperature.
X	<0,1>	-	<0,100>[%]	The value of rod displacement.
F	<0,1>	-	<0,40>[t/h]	The medium flow.

### 5.1.1 Training Data and Testing Data

There are two types of data sources provided by DAMADICS: 1. Actuator Simulink model (DABLib); 2. Real process data with artificial faults. In addition, the analytic model of actuator is not available for both of these two data sources. Therefore, all necessary information of the actuator system has to be acquired from the generated/collected data [66].

#### 5.1.1.1 Actuator Simulink Model (DABLib)

On this track, all sensor measurements of the actuator system are generated by a Simulink model named DABLib. DABLib contains nine different blocks, and every block serves for different purpose. The description of these blocks is listed in Table 6.

Table 6. Description of DABLib blocks. (Adapted from [69])

Block name	Description
Fault generator	Setting fault strength
Fault selector	Selecting desired fault and assigning fault attributes
Data generator	Generating repeatable patterns of data sets for benchmark purposes
Coordinator data generator	Generating repeatable patterns of data sets with hidden faults for benchmark purposes
Performance indexes calculator	Automatically calculating performance indexes for benchmark purposes
<b>PROCESS COMPONENTS</b> block group	Containing components that are used only in case of process simulating
<b>DABLIB UTILITIES</b> block group	Containing the set of components that are used by other DABLib blocks
ACT	The model of actuator
ExtACT	An extension of ACT block. This block has additional three outputs.

Among these blocks, ACT block is the Simulink model of the actuator system and also the main block for generating simulated data. The illustration of inputs and outputs of the ACT block is shown as Fig. 63, and the description of these inputs and outputs is listed in Table 7(a) and 7(b). Every output of the actuator system are assumed to be disturbed by a Gaussian distributed noise, and this noise is considered as measurement noise [69]. The illustration of noise entries in the ACT block is shown as Figure 64. Observing Fig. 64, one can notice that not only sensor outputs of the actuator system are disturbed but also control input (CV) is disturbed before it enters the actuator system [69]. In addition, for generating simulated process data, the inputs of the ACT block are fixed as: 1. Initial value of CV: 0; 2. Final value of CV: 0.8; 3. Step time of CV from its initial value to final value: sample time 2000; 4.  $P_1 = 3.5$  MPa; 5.  $P_2 = 2.6$  MPa; 6.  $T_1 = 20$  °C. The control input (CV) signal used to generate simulated data is shown as Fig. 65.

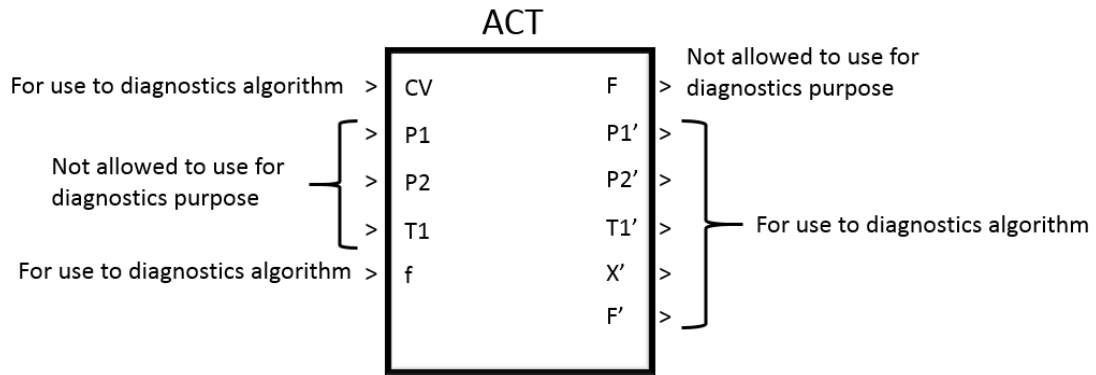


Figure 63. Illustration of inputs and outputs of the ACT block. (Adapted from [69])

Table 7(a). Description of inputs of the ACT block. (Adapted from [69])

Input	Range	Physical Unit	Description
CV	<0,1>	100%	Control variable is the output signal from the controller.
P1	-	Pa	The value of the pressure on the control valve inlet.
P2	-	Pa	The value of the pressure on the control valve outlet.
T1	-	°C	The temperature of medium.
f	<-1,1> or <0,1>	-	The vector of faults

Table 7(b). Description of outputs of the ACT block. (Adapted from [69])

Output	Range	Physical Unit	Transducer range	Description
F	-	t/h	-	The medium flow.
P1'	<0,1>	-	<0,4>[MPa]	The disturbed value of the pressure on the control valve inlet.
P2'	<0,1>	-	<0,4>[MPa]	The disturbed value of the pressure on the control valve outlet.
T1'	<0,1>	-	<0,200>[°C]	The disturbed medium temperature.
X'	<0,1>	-	<0,100>[%]	The disturbed value of rod displacement.
F'	<0,1>	-	<0,40>[t/h]	The disturbed medium flow.



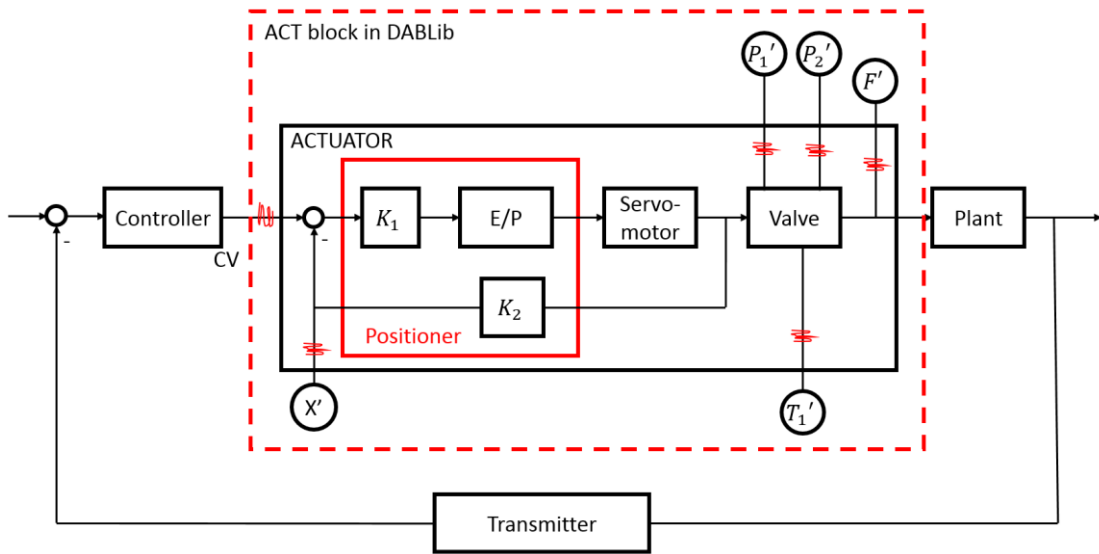


Figure 64. Illustration of noise entries in the ACT block. (Adapted from [69])

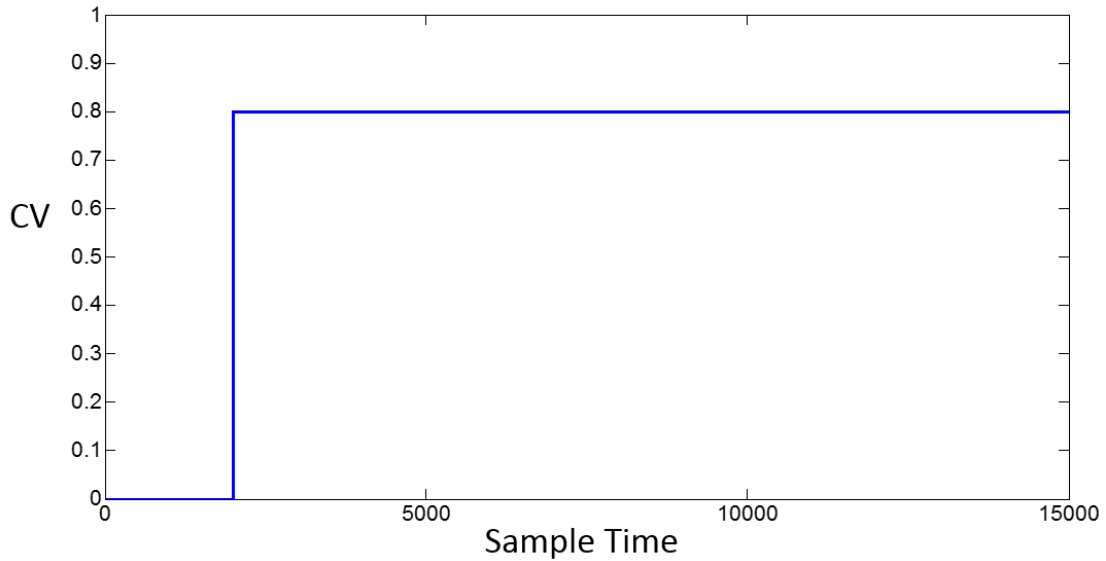


Figure 65. Control input of the ACT block for generating simulated process data.

To generate training data and healthy testing data, we select “no fault” in the fault selector block. The only difference between training data and healthy testing data is that

the sensor measurement noise is simulated from different starting seed. Therefore, we will obtain a noise signal by subtracting healthy from testing data training data shown as Fig. 66.

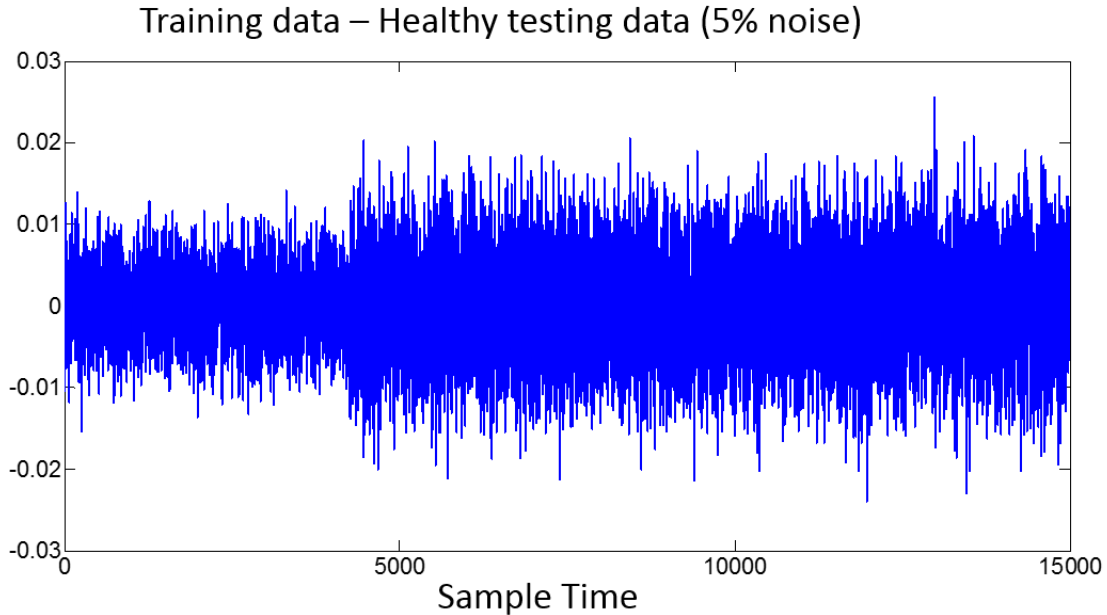


Figure 66. Subtracting healthy testing data from training data in 5% noise case.

On the other hand, we select “fault No.19 – flow rate sensor fault” to generate the faulty testing data. The description of the flow rate sensor fault is shown in Table 8. In our experiments, two types of fault are simulated: abrupt and incipient, and both of them are set to occur at sample time 5200. For incipient fault, its develop time is set to be 4000 samples. In the other words, assume an incipient fault occurs at sample time 5200, this fault will be fully developed at sample time 9200. Figure 67 is the flow rate sensor measurements in the cases of abrupt fault, incipient fault, and fault-free contaminated with

5% noise. According to Fig. 67, one can observe that the flow rate sensor will be immediately disabled when an abrupt fault occurs. On the other hand, in the incipient fault situation, the flow rate sensor will not be totally disabled at the beginning. Instead, the flow rate sensor will gradually lose its function and be totally disabled after the incipient fault is fully developed.

Table 8. Description of the flow rate sensor fault. (Adapted from [69])

Fault type	Flow rate sensor fault
Physical interpretation	Caused by electronics or wiring failure
Fault primary nature	Abrupt
Simulated action	Simulated by modifying sensor output characteristics
Range	<-1,1>
Fault strength interpretation	-1: flow measurement = 0; independent on real flow. 0: no fault. 1: flow measurement = 1; independent on real flow.
Fault simulation technique equation	$\begin{cases} F'_f = F'(1 + f_s) & F'_f - \text{relative flow measurement} \\ \text{if } F'_f < 0 \text{ then } F'_f = 0 & F' - \text{relative flow} \\ \text{if } F'_f > 1 \text{ then } F'_f = 1 & f_s - \text{fault strength} \end{cases}$

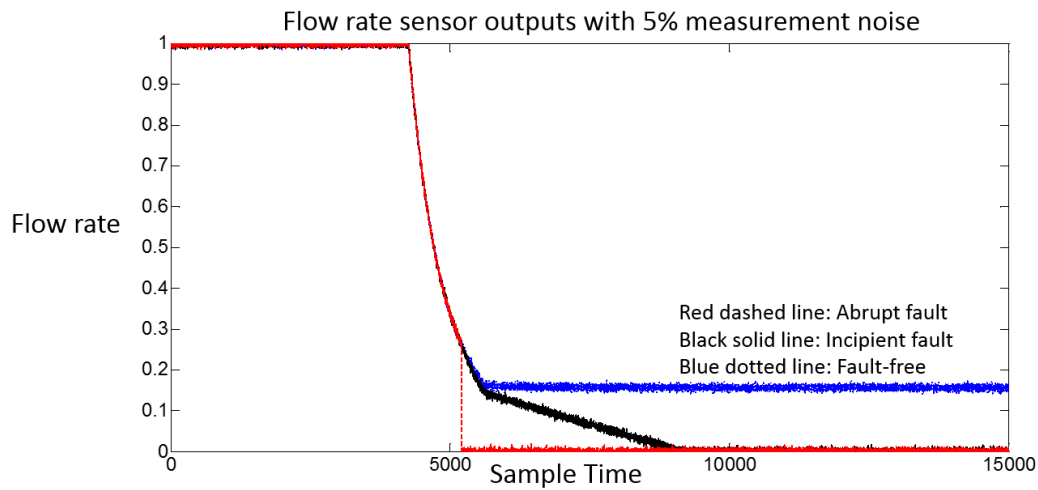


Figure 67. The flow rate sensor measurements in the cases of abrupt fault, incipient fault, and fault-free contaminated with 5% noise.

### 5.1.1.2 Real Process Data with Artificially Generated Faults

Koscielny et al... also provide real process data in their proposed benchmark for testing FDI methods. These real process data are collected from three actuators used in a sugar factory during 2001.10.29 to 2001.11.20, and various types of artificial faults are added in 2001.10.30, 2001.11.9, 2001.11.17, and 2001.11.20 [70]. Table 9 is the description of actuator faults added in the real process data. In our experiments, we choose the data file collected from the 3<sup>rd</sup> actuator on 2001.11.17 since several artificial faults were added in order to cause abnormal events in the flow rate sensor measurements on that day. Table 10 is the description of artificially added actuator faults for the chosen data file. According to the occurring time listed in Table 10, we choose 50001 ~ 60000<sup>th</sup> sample as the faulty testing data, 10001 ~ 20000<sup>th</sup> sample as the training data, and 30001 ~ 40000<sup>th</sup> sample as the healthy testing data.

Table 9. Description of actuator faults. (Adapted from [70])

Fault	Description
Control valve faults	
f1	Valve clogging
f2	Valve plug or valve seat sedimentation
f3	Valve plug or valve seat erosion
f4	Increased of valve or bushing friction
f5	External leakage (leaky bushing, covers, terminals)
f6	Internal leakage (valve tightness)
f7	Medium evaporation or critical flow
Pneumatic servo-motor faults	
f8	Twisted servo-motor's position rod
f9	Servo-motor's housing or terminals tightness
f10	Servo-motor's diaphragm perforation
f11	Servo-motor's spring fault
Positioner faults	
f12	Electro-pneumatic transducer fault
f13	Rod displacement sensor fault
f14	Pressure sensor fault
f15	Positioner feedback fault
General faults/External faults	
f16	Positioner supply pressure drop
f17	Unexpected pressure change across the valve
f18	Fully or partly opened bypass valves
f19	Flow rate sensor fault

Table 10. Description of artificially added actuator faults in the 3<sup>rd</sup> actuator. (Adapted from [70])

Fault	Sample	Date	description
f16	57475-57530	2001.11.17	Positioner supply pressure drop
f16	57675-57800	2001.11.17	Positioner supply pressure drop
f19	58150-58325	2001.11.17	Flow rate sensor fault

## 5.2 Results of Applying the Proposed Method to DAMADICS

### 5.2.1 Results of Data Generated by Actuator Simulink Model (DABLib)

In this section, the proposed method is examined by the actuator Simulink model generated data. There are three different level of noise in this experiment: 1.5%, 5%, and 10%, and three different situations are considered under each noise level: healthy, abrupt fault, and incipient fault. Figure 68, 69, and 70 are the results of 1.5% noise in healthy, abrupt fault, and incipient fault situation. The raw residual signal for this case is shown in Fig. 68(a). Observing Fig. 68(a), one can find that the magnitude of the raw residual signal does not equal zero and has fluctuations. As mentioned in Section 4, these fluctuations may cause false alarms in the healthy situation. In order to address this issue, the probability of the occurrence of fault is obtained by using MHSSPT to further process the raw residual signal shown in Fig. 68(b). According to Fig. 68(b), the probability keeps low and close to zero for all time. This result shows that the proposed method correctly indicate the actuator system is healthy. The result of abrupt fault situation is shown in Fig.69. Similar to Fig. 68, the raw residual signal in the abrupt fault situation is shown in Fig. 69(a). Comparing Fig. 69(a) with Fig. 68(a), since these two raw residual signals are already distinguished, ideally one can set an appropriate threshold to indicate healthy and

faulty situations instead of further processing by MHSSPT. However, without prior knowledge of faulty data, one can only set the threshold according to the raw residual signal produced by the second set of healthy testing data, and this way is not reliable enough to avoid all false alarms and miss detection due to the effect of noise. The probability of the occurrence of fault is shown in Fig. 69(b). In Fig. 69(b), the probability increases immediately right after the abrupt sensor fault occurs at sample time 5200. This result indicates that the proposed method correctly detects the abrupt fault when it occurs. The result of incipient fault situation is shown in Fig. 70. Observing Fig. 70, the probability of the occurrence of fault does not increase immediately right after sample time 5200. Instead, it increases around sample time 5500. This is reasonable because the magnitude of the incipient fault is not larger enough to be considered as an abnormal event at the beginning.

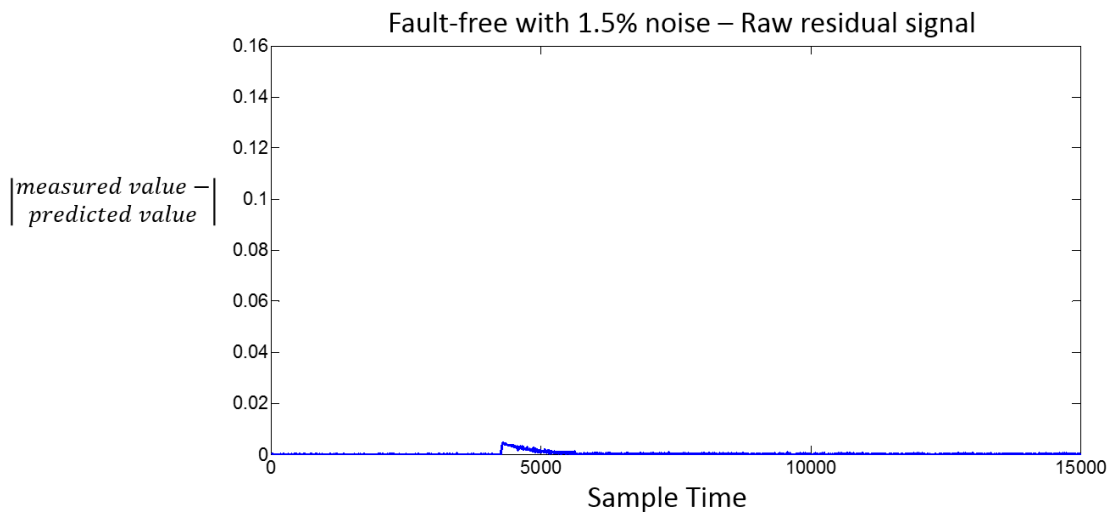


Figure 68(a). The raw residual signal produced in the healthy situation with 1.5% noise.

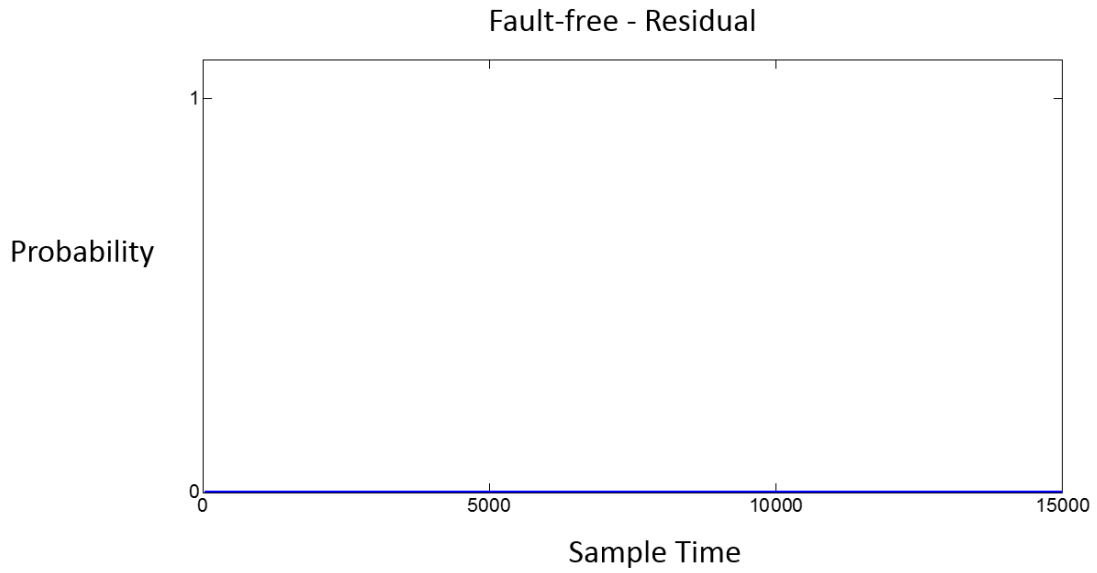


Figure 68(b). The probability of the occurrence of fault in the healthy situation with 1.5% noise.

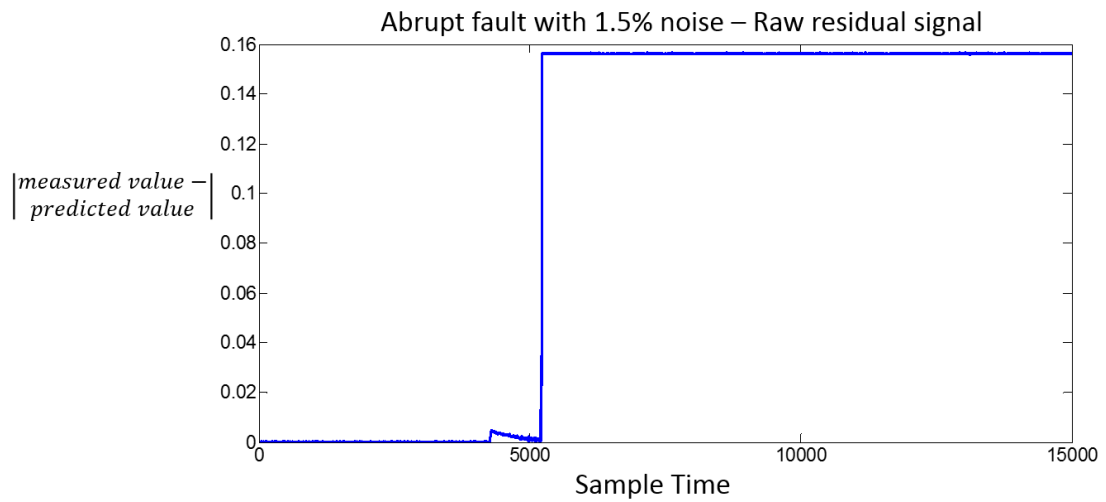


Figure 69(a). The raw residual signal produced in the abrupt fault situation with 1.5% noise.



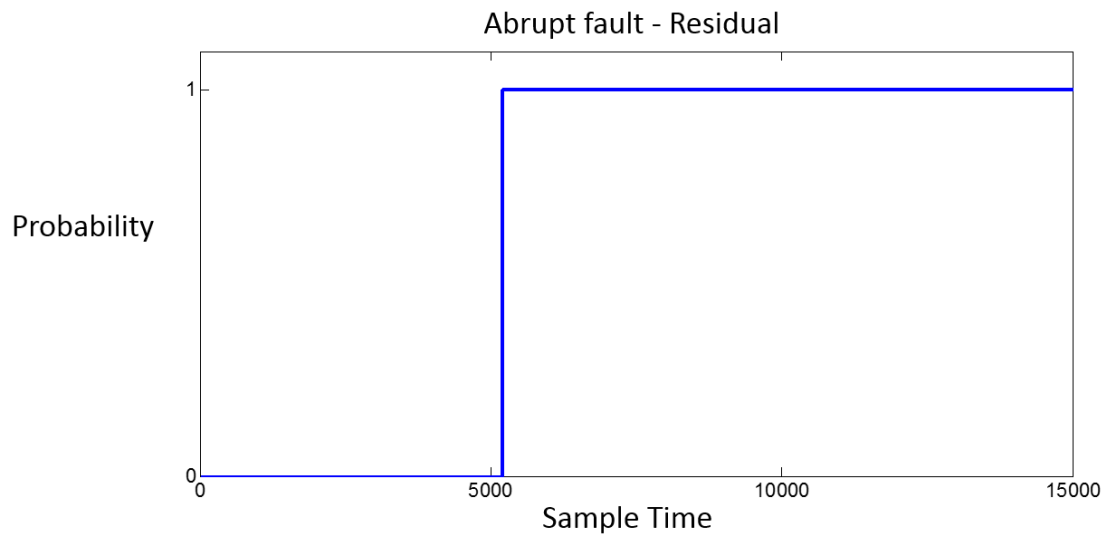


Figure 69(b). The probability of the occurrence of fault in the abrupt fault situation with 1.5% noise.

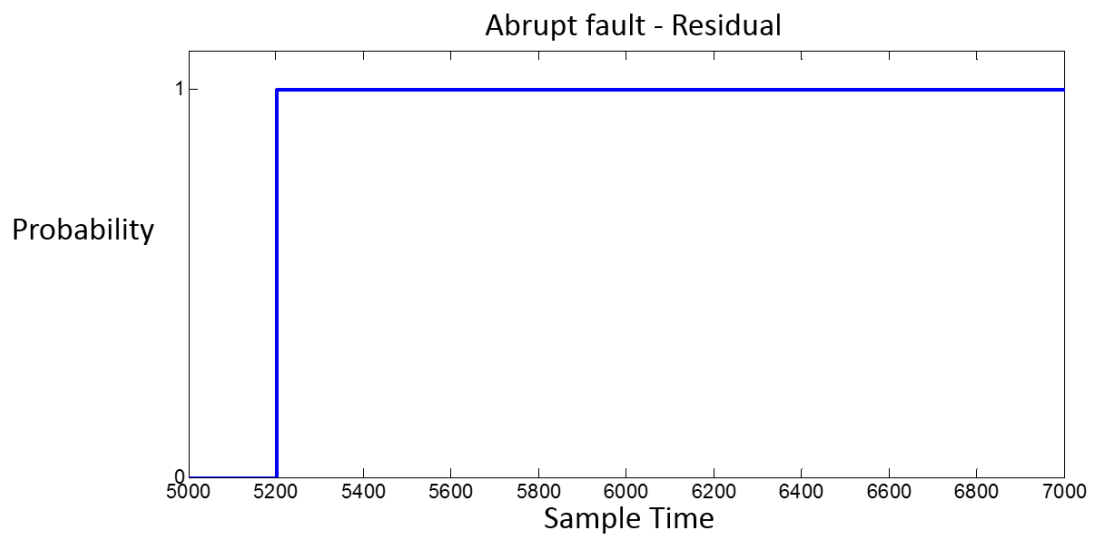


Figure 69(c). The enlargement of Fig. 69(b).

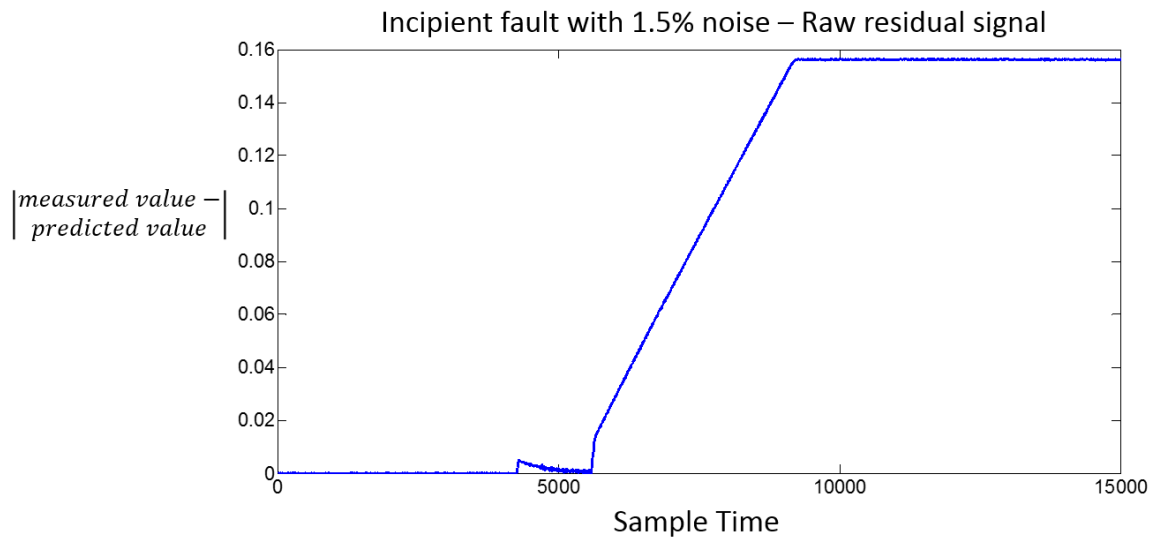


Figure 70(a). The raw residual signal produced in the incipient fault situation with 1.5% noise.

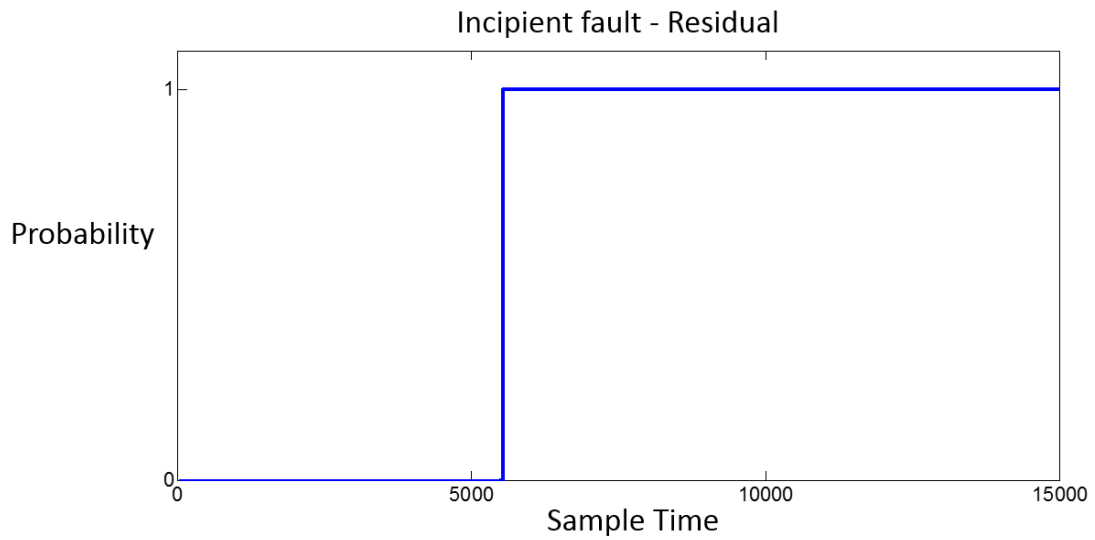


Figure 70(b). The probability of the occurrence of fault in the incipient fault situation with 1.5% noise.

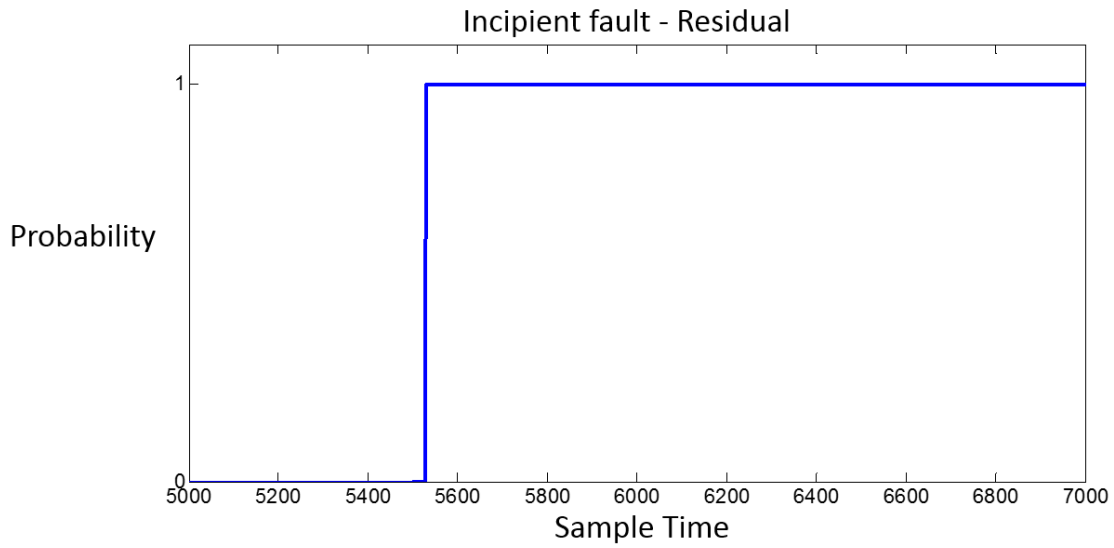


Figure 70(c). The enlargement of Fig. 70(b).

The results of 5% noise in healthy, abrupt fault, and incipient fault situation are shown in Fig. 71, 72, and 73. Observing these results, one can notice that the behavior of the residual signals in healthy, abrupt fault, and incipient fault situation is similar to the case of 1.5% noise. Therefore, the capability of the proposed method to detect faults in the case of 5% noise can be proved.

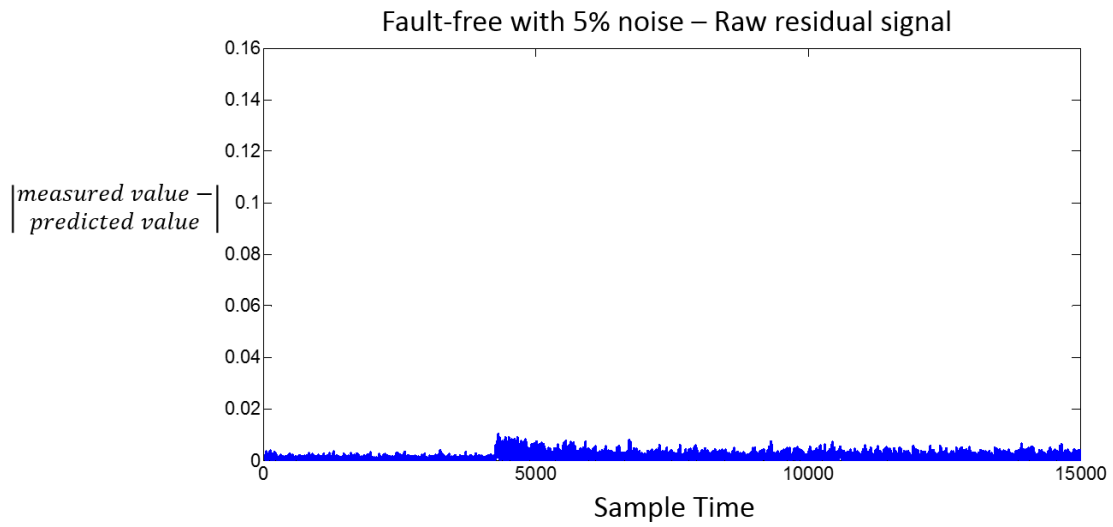


Figure 71(a). The raw residual signal produced in the healthy situation with 5% noise.

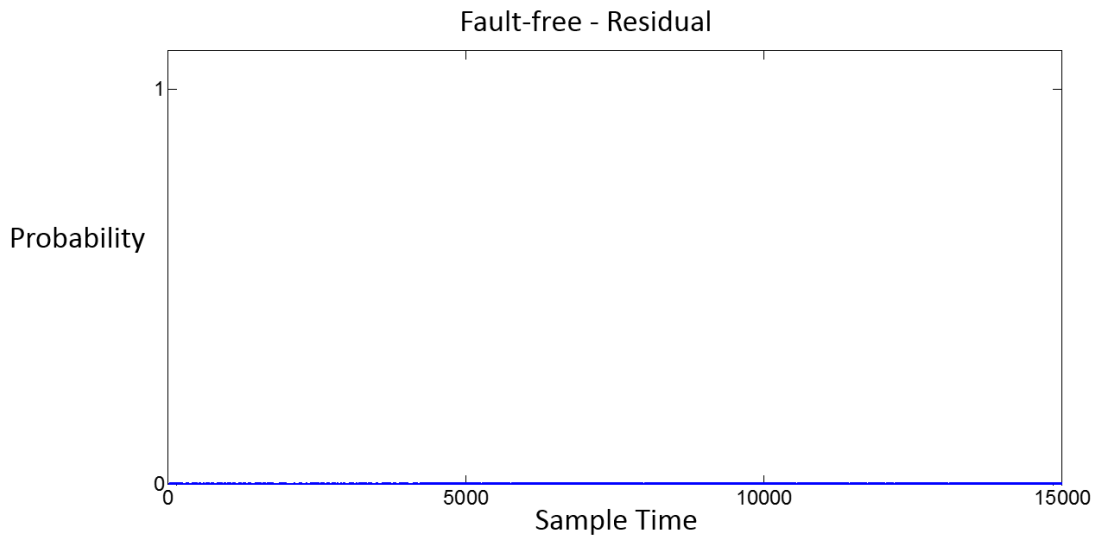


Figure 71(b). The probability of the occurrence of fault in the healthy situation with 5% noise.

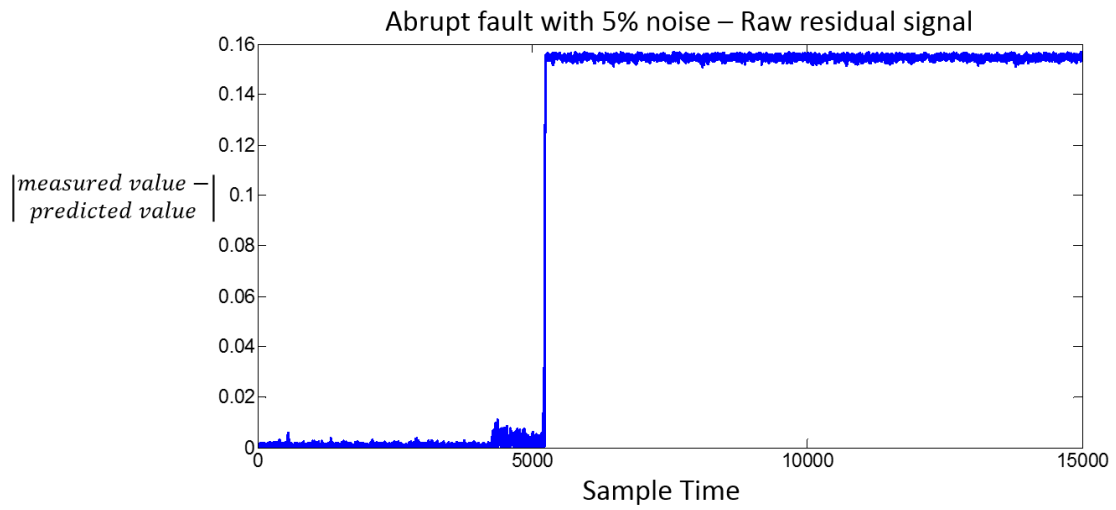


Figure 72(a). The raw residual signal produced in the abrupt fault situation with 5% noise.

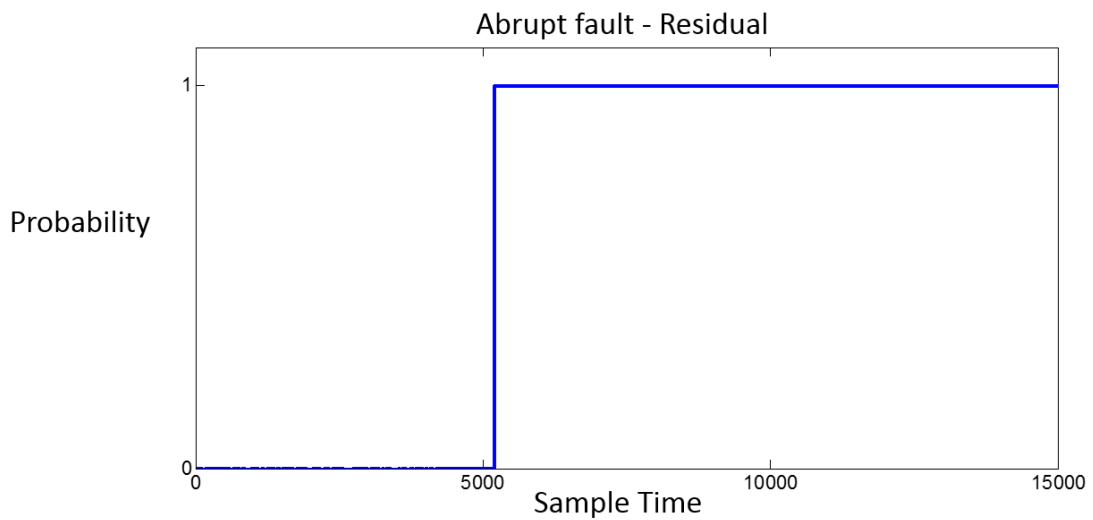


Figure 72(b). The probability of the occurrence of fault in the abrupt fault situation with 5% noise.

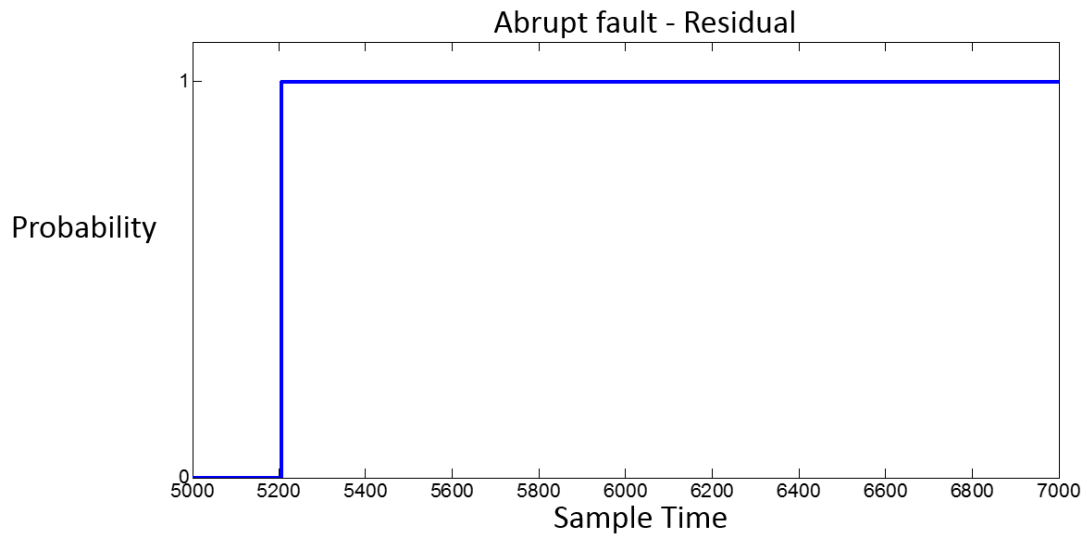


Figure 72(c). The enlargement of Fig. 72(b).

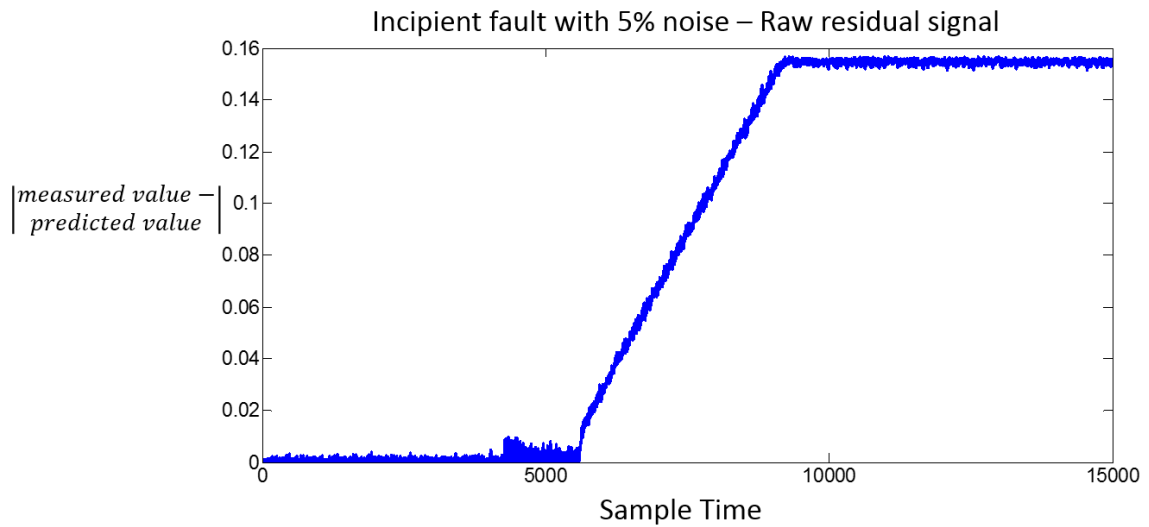


Figure 73(a). The raw residual signal produced in the incipient fault situation with 5% noise.

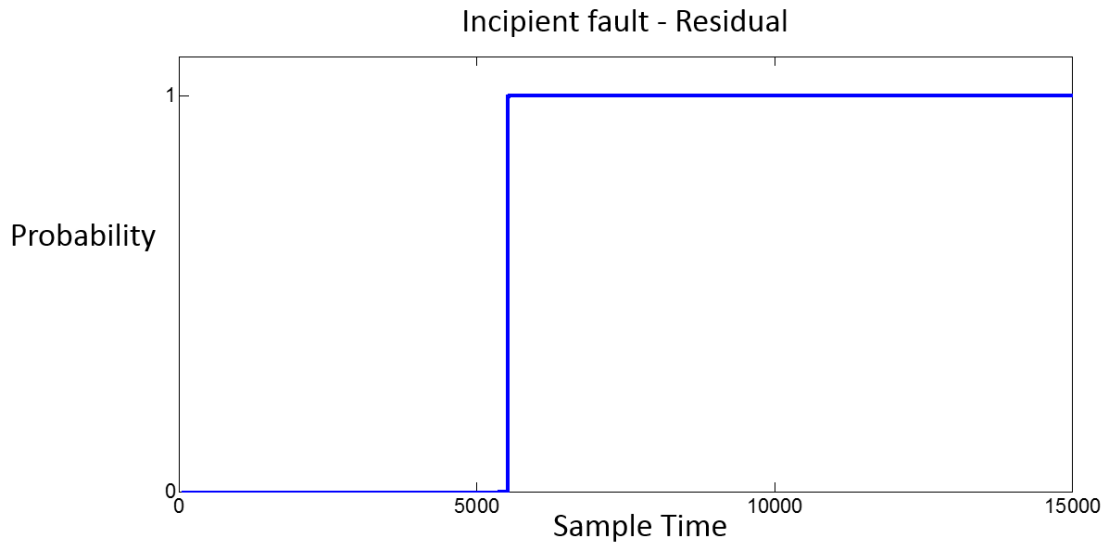


Figure 73(b). The probability of the occurrence of fault in the incipient fault situation with 5% noise.

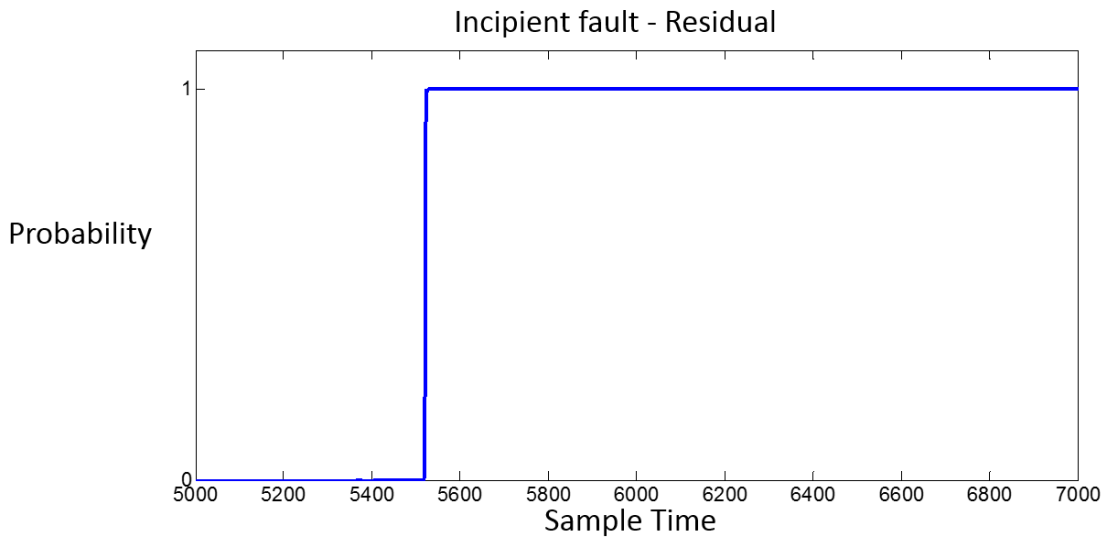


Figure 73(c). The enlargement of Fig. 73(b).

The results of 10% noise in healthy, abrupt fault, and incipient fault situation are shown in Fig. 74, 75, and 76. According to these results, the proposed method correctly indicate the actuator is healthy in the healthy situation shown in Fig. 74, and it also correctly and immediately detects the abrupt fault in the abrupt fault situation shown in Fig. 75. However, for the incipient fault situation, the proposed method takes longer time to confirm the incipient fault compared to 1.5% and 5% noise level. This result is reasonable since the required magnitude of an incipient fault to be detected is larger in the case of 10% noise level than in the case of 1.5% and 5% noise level.

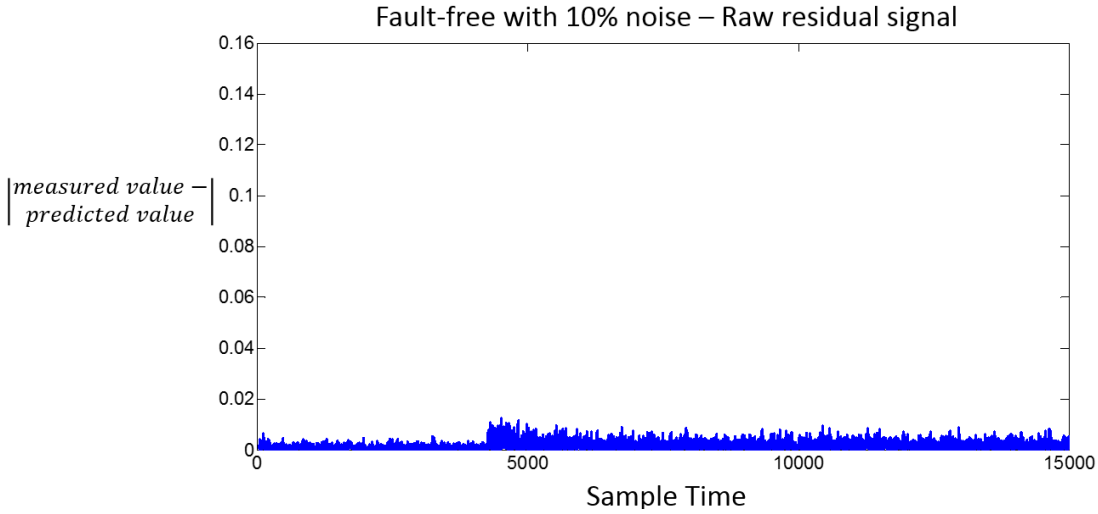


Figure 74(a). The raw residual signal produced in the healthy situation with 10% noise.



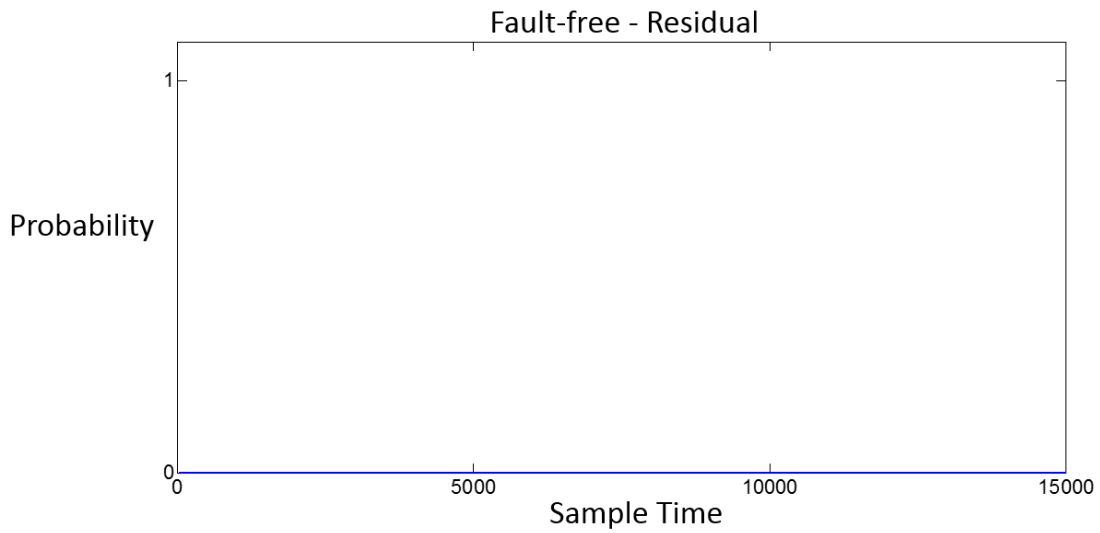


Figure 74(b). The probability of the occurrence of fault in the healthy situation with 10% noise.

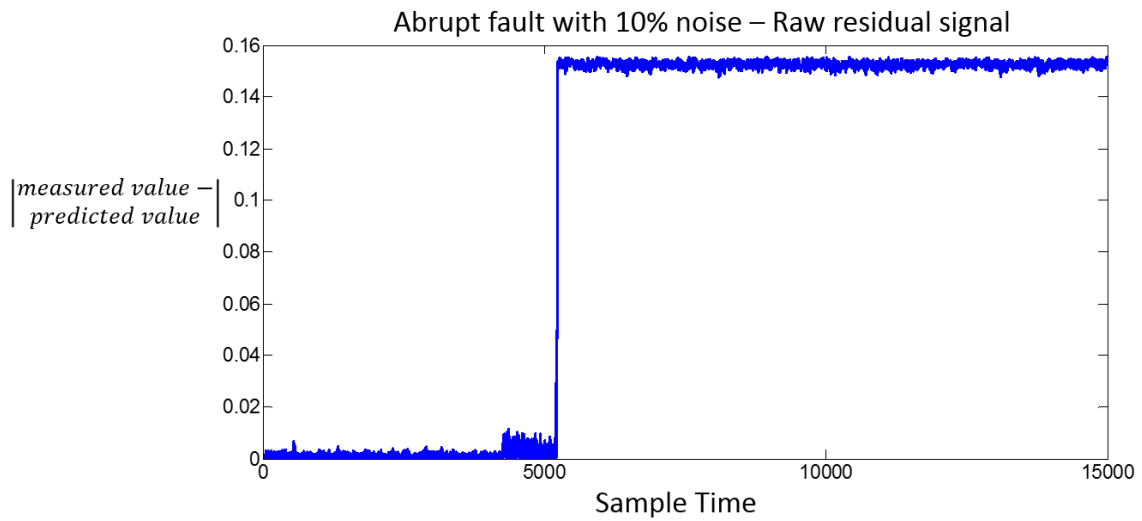


Figure 75(a). The raw residual signal produced in the abrupt fault situation with 10% noise.

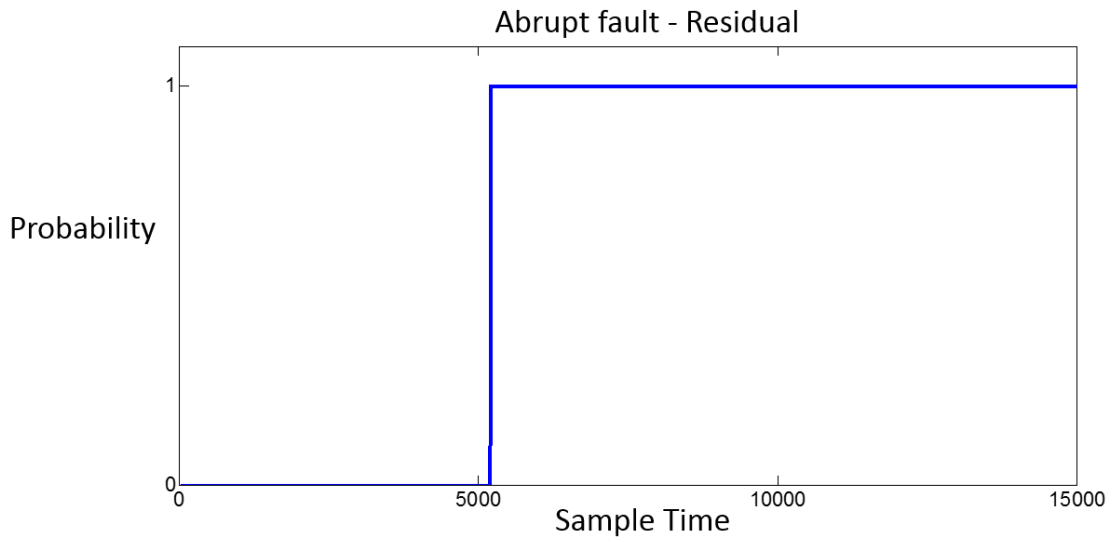


Figure 75(b). The probability of the occurrence of fault in the abrupt fault situation with 10% noise.

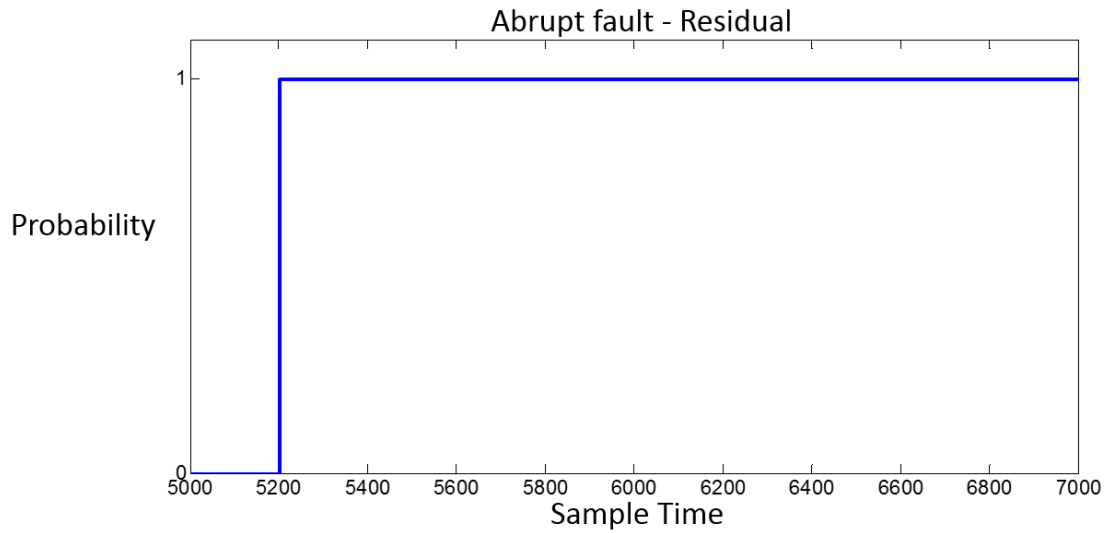


Figure 75(c). The enlargement of Fig. 75(b).

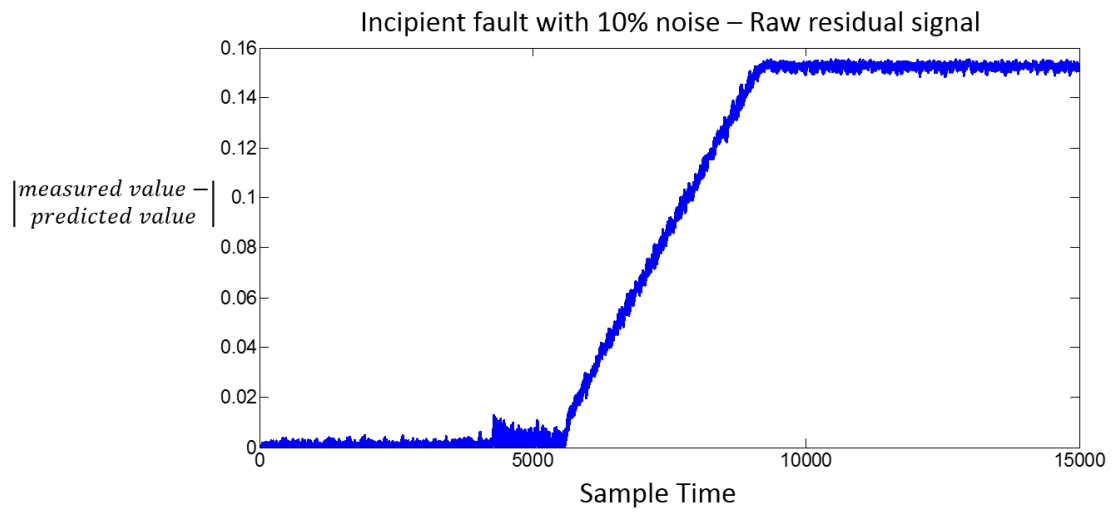


Figure 76(a). The raw residual signal produced in the incipient fault situation with 10% noise.

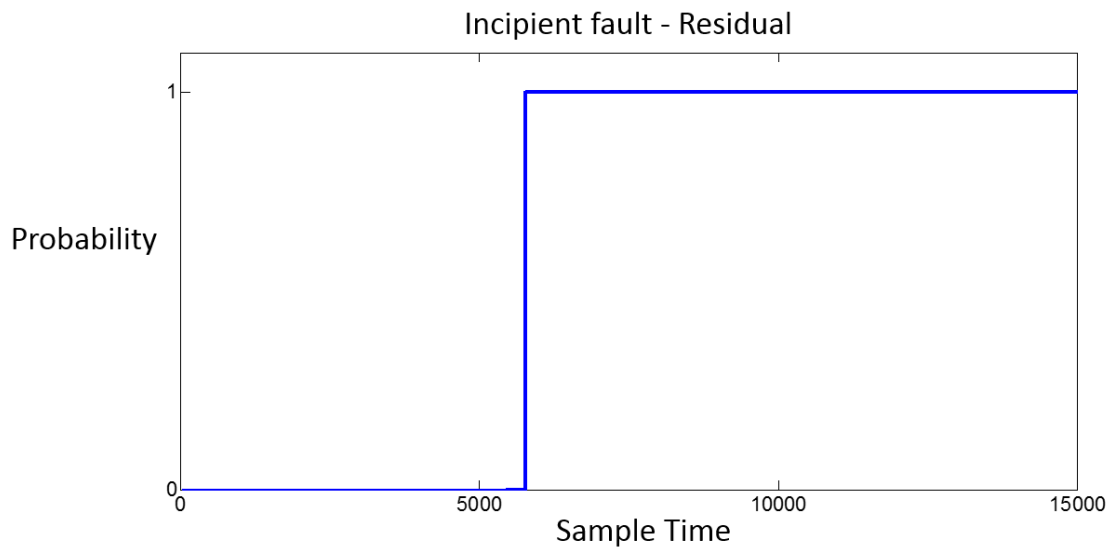


Figure 76(b). The probability of the occurrence of fault in the incipient fault situation with 10% noise.

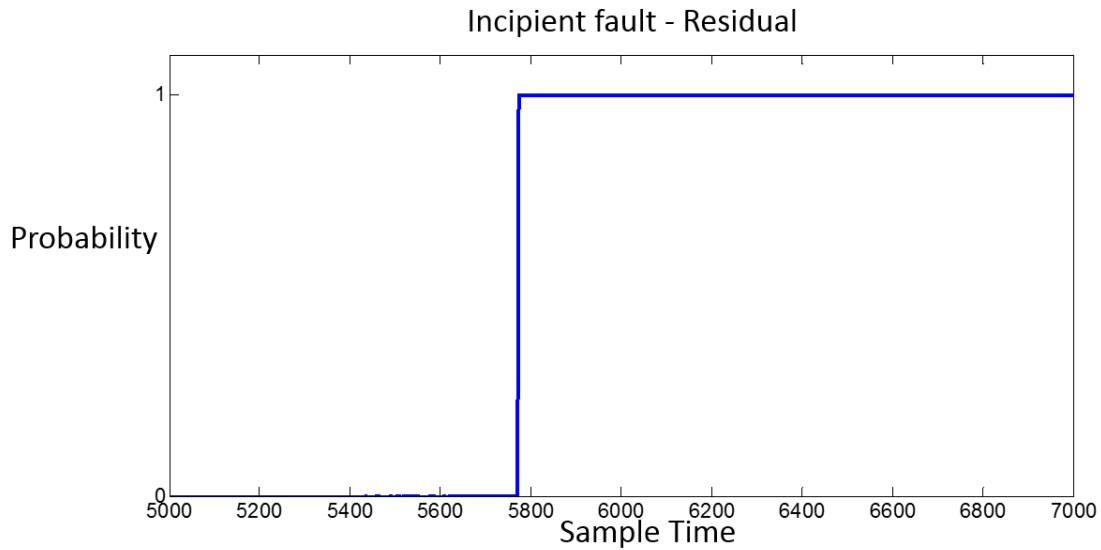


Figure 76(c). The enlargement of Fig. 76(b).

On the other hand, the results of the extension of the proposed method by the conservation/dissipation property in phase space will be presented below. In this experiment, the initial value of the control input (CV) for two reference trajectories are set to be 0.5 and 0.6, and the initial condition of the unknown trajectory is set to be 0.55. The final value of the control input are all set to be 0.8. All faults are set to occur at sample time 2800. Figure 77-85 are the results of applying the conservation/dissipation property for fault detection and isolation in 1.5%, 5%, and 10% noise level. According to these results, the probability of the occurrence of fault keeps low and closes to zero in the healthy situation for all noise levels. This phenomenon suggests that the proposed method correctly indicate the actuator system is healthy without any false alarm. In the abrupt fault situation, the probability of the occurrence of fault increases around sample time 2810 for all noise levels, and this indicates that the proposed method is able to detect the abrupt

fault occurs at sample time 2800 with an acceptable reacting time. Finally, in the incipient fault situation, the probability of the occurrence of fault increases around sample time 3050 for all noise levels. These results suggest that the proposed method is able to detect an incipient fault which has been developed to a certain level.

Therefore, according to those results shown in this section, the performance of the proposed method and its extension by using conservation/dissipation property can be proved in the cases of 1.5%, 5%, and 10% noise level.

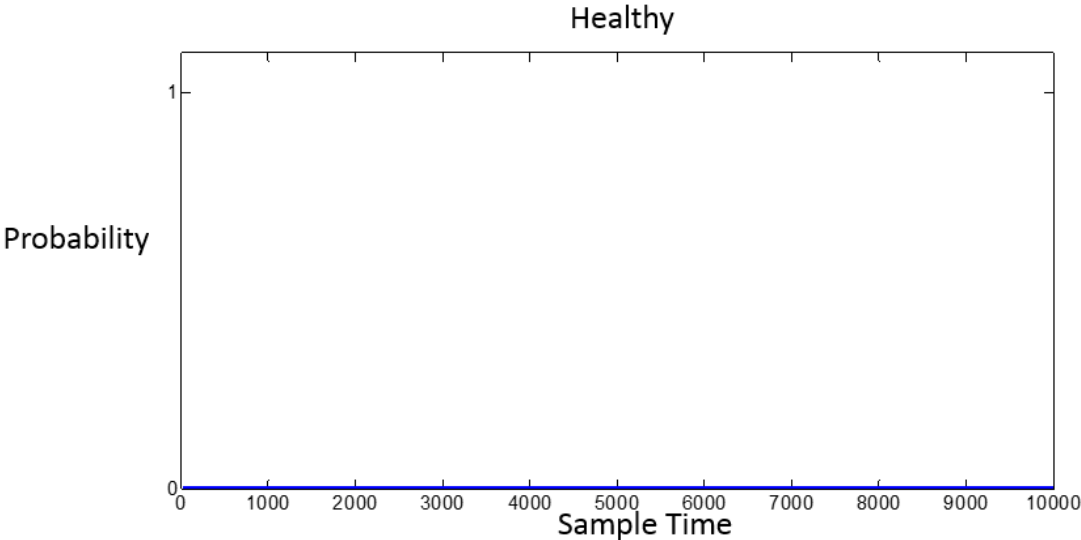


Figure 77. The probability of the occurrence of fault in the fault-free situation with 1.5% noise - FDI by conservation/dissipation property.

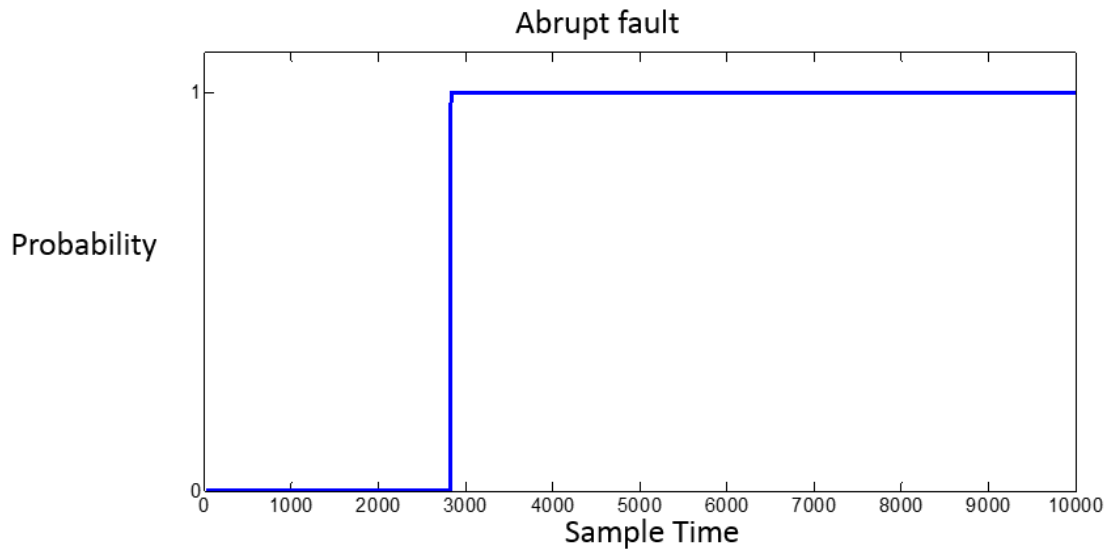


Figure 78(a). The probability of the occurrence of fault in the abrupt fault situation with 1.5% noise - FDI by conservation/dissipation property.

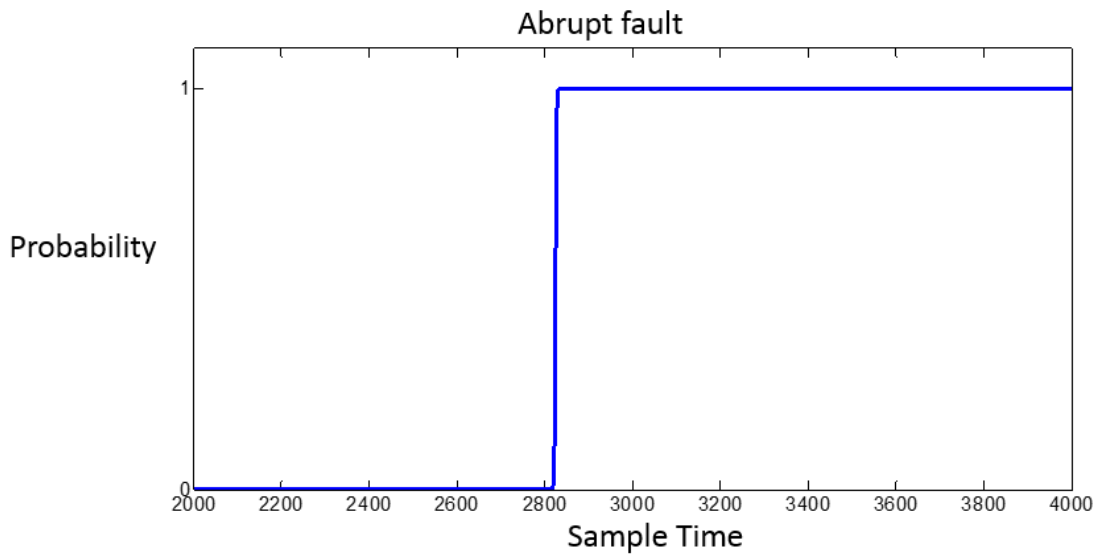


Figure 78(b). The enlargement of Fig. 78(a).

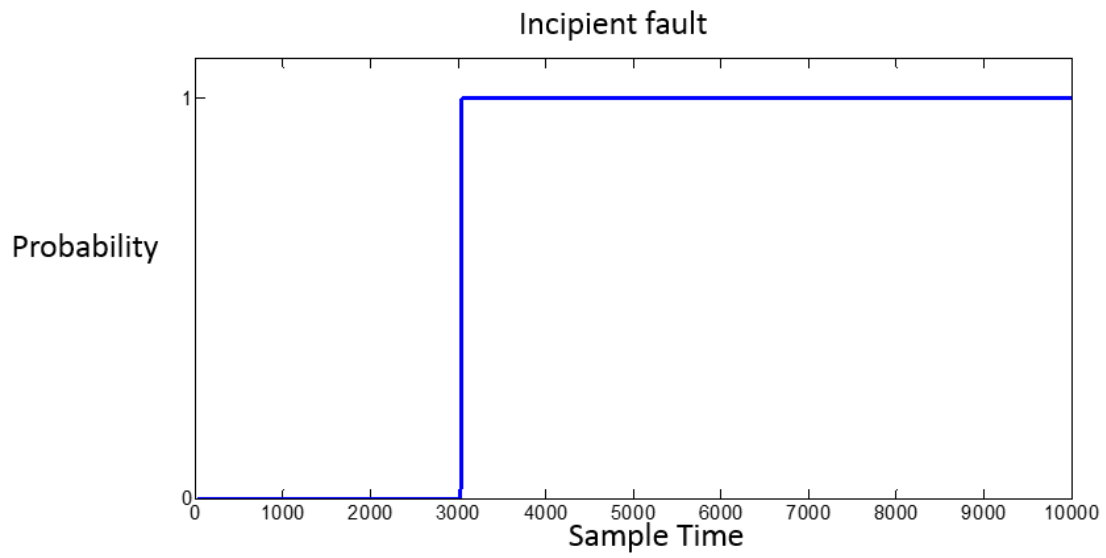


Figure 79. The probability of the occurrence of fault in the incipient fault situation with 1.5% noise - FDI by conservation/dissipation property.

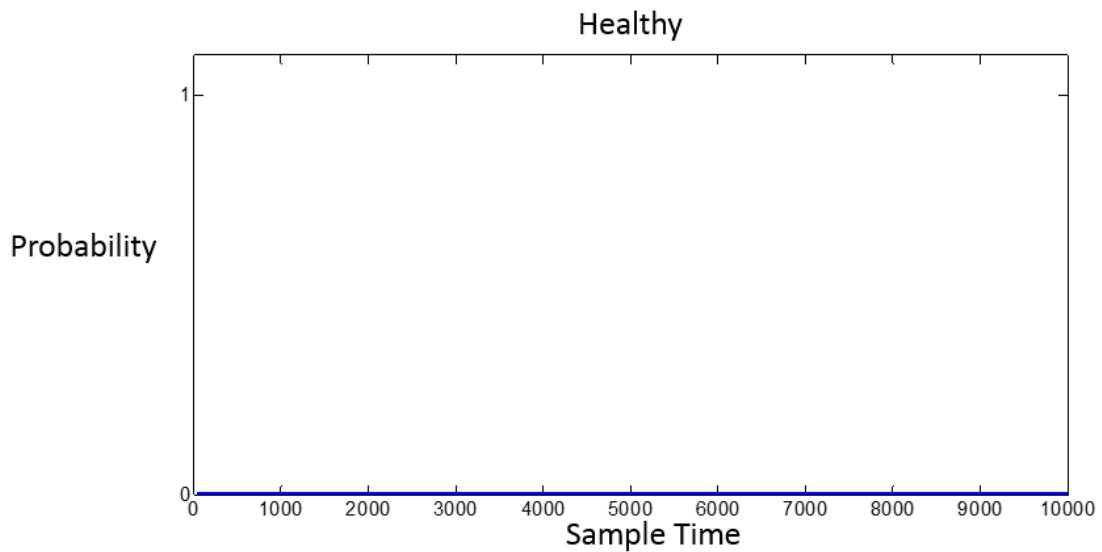


Figure 80. The probability of the occurrence of fault in the fault-free situation with 5% noise - FDI by conservation/dissipation property.

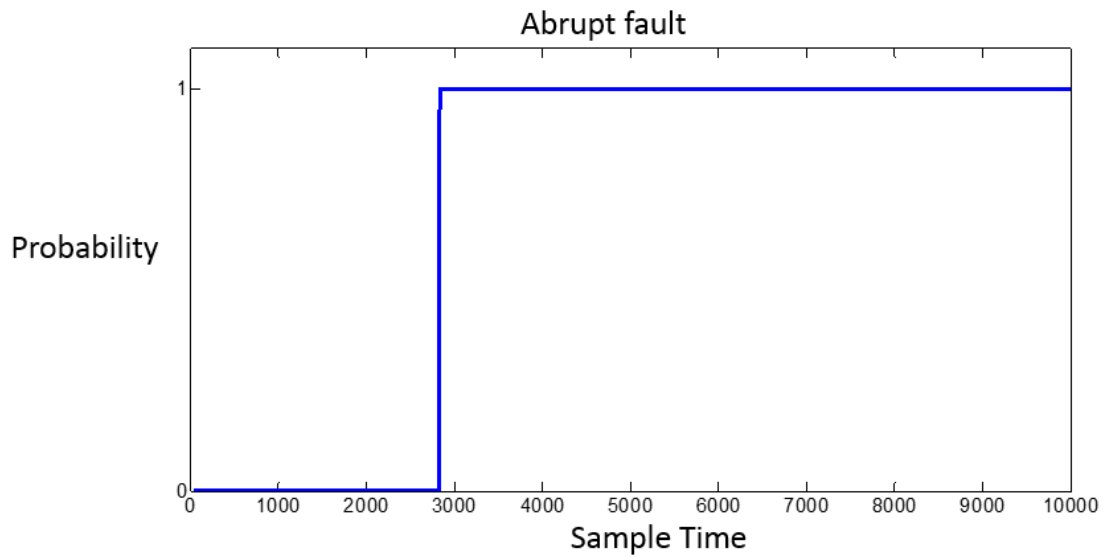


Figure 81(a). The probability of the occurrence of fault in the abrupt fault situation with 5% noise - FDI by conservation/dissipation property.

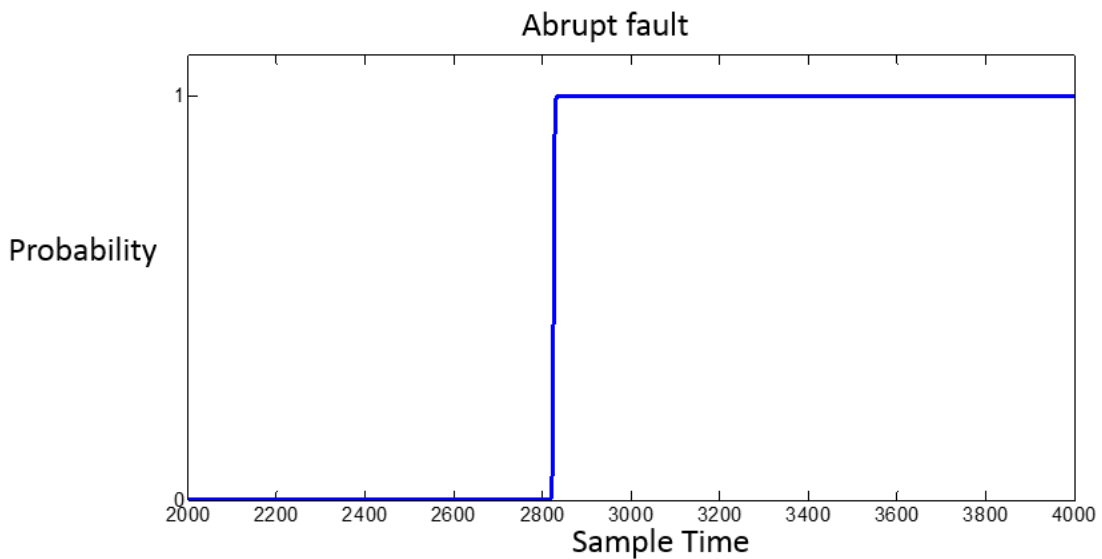


Figure 81(b). The enlargement of Fig. 81(a).



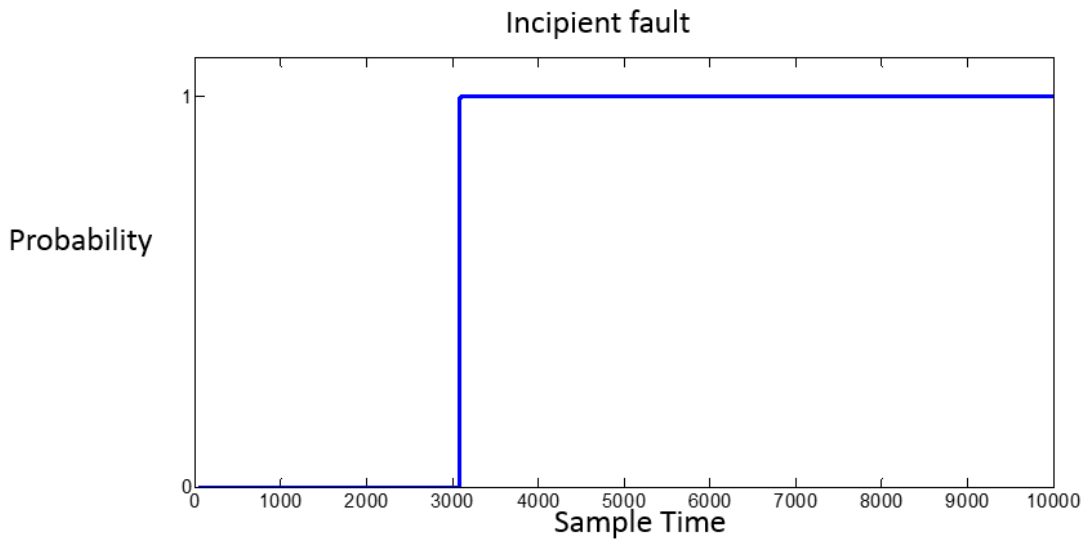


Figure 82. The probability of the occurrence of fault in the incipient fault situation with 5% noise - FDI by conservation/dissipation property.

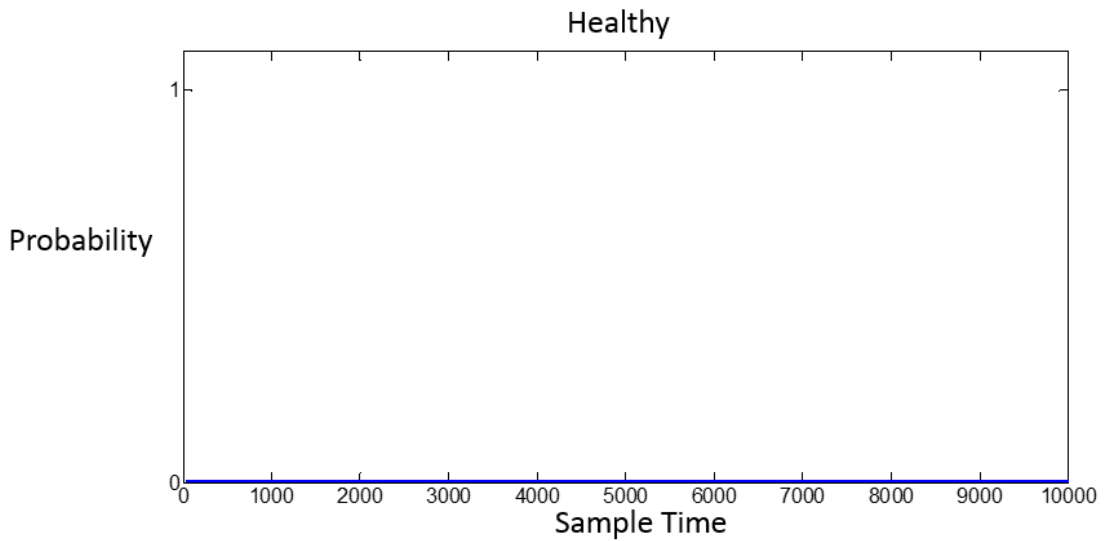


Figure 83. The probability of the occurrence of fault in the fault-free situation with 10% noise - FDI by conservation/dissipation property.

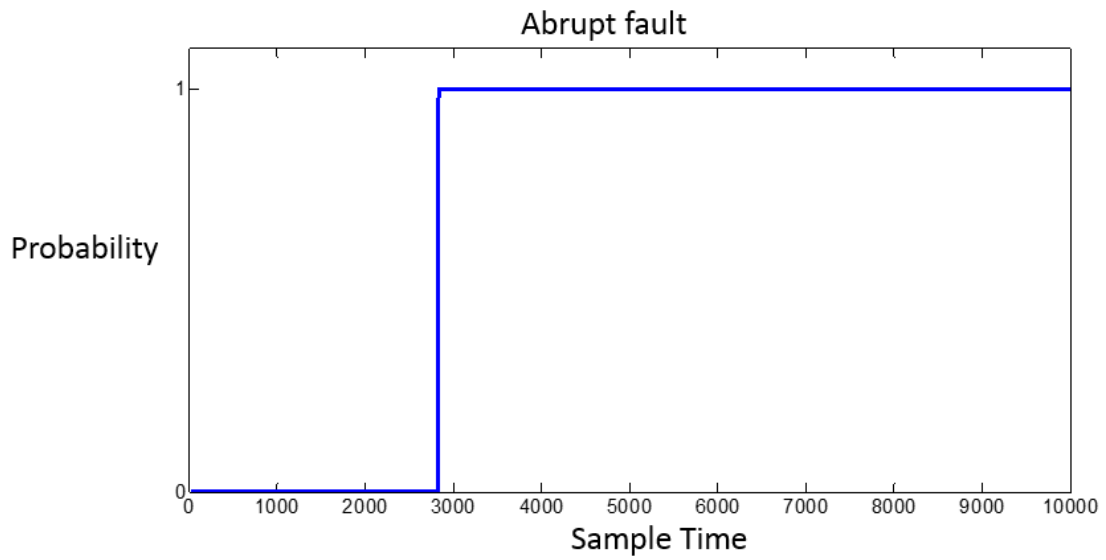


Figure 84(a). The probability of the occurrence of fault in the abrupt fault situation with 10% noise - FDI by conservation/dissipation property.

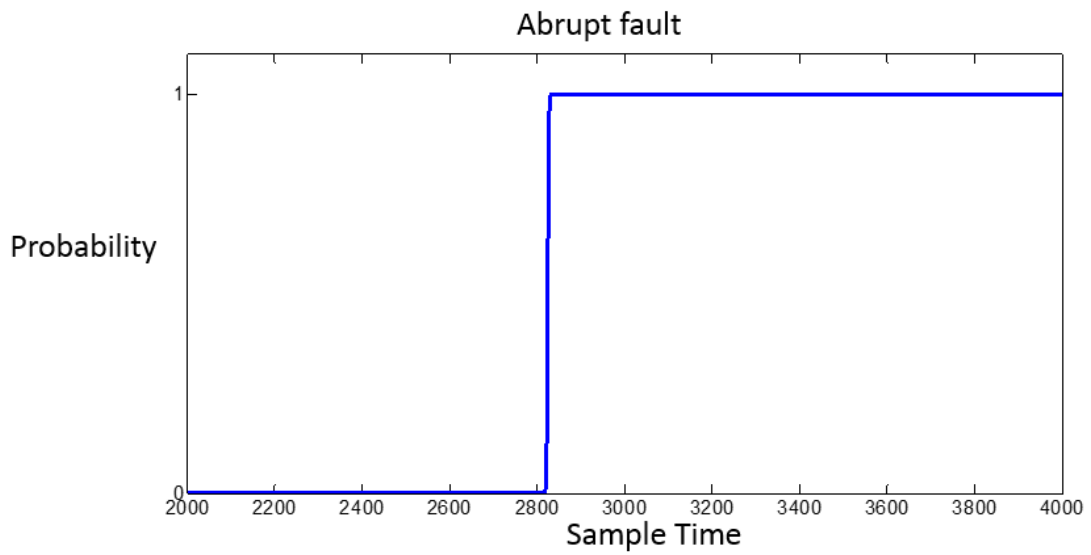


Figure 84(b). The enlargement of Fig. 84(a).

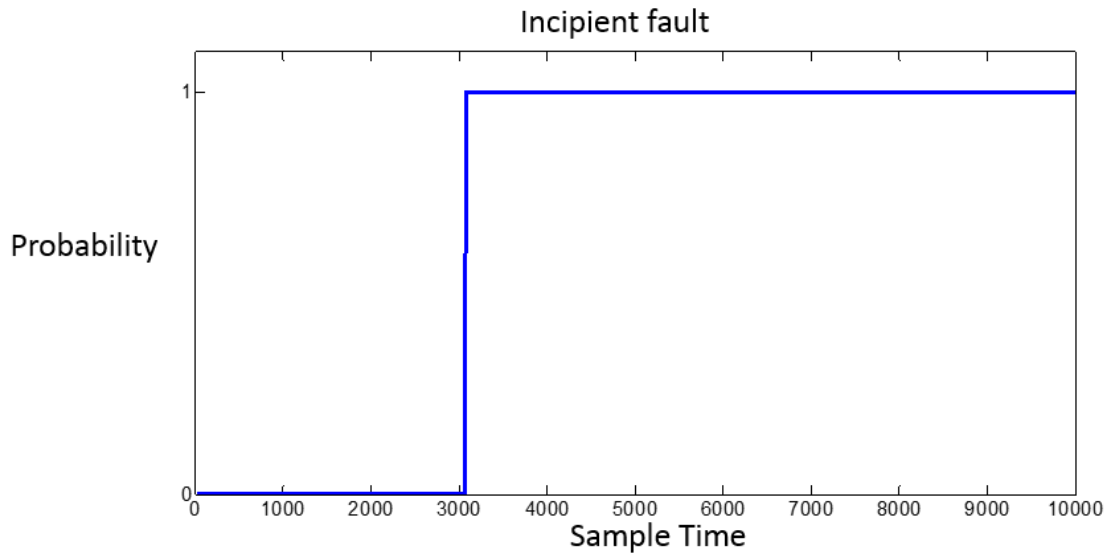


Figure 85. The probability of the occurrence of fault in the incipient fault situation with 10% noise - FDI by conservation/dissipation property.

### 5.2.2 Results of Real Process Data Collected from Actuators Used in a Sugar Factory

In this section, we examine the proposed method by the real process data collected from a sugar factory. Figure 86 shows the training data used for reconstructing phase space of the actuator system. As mentioned before, the training data is the 10001 ~ 20000<sup>th</sup> sample of the data file collected on 2001.11.17. The results of mutual information method and False Nearest Neighbors (FNNs) method are shown in Fig. 87 and 88. According to these results, the delay time and the dimension for reconstructing phase space are set to be 4 for both.

The result of the proposed method applied to real process data in the healthy situation is shown in Fig. 89. According to this figure, the probability of the occurrence of fault keeps low and closes to zero for all time. This suggests that the proposed method correctly indicates the situation is healthy. To further support this point, Fig. 90 shows the result of

the prediction mechanism block. Here, the red dotted line represents the predicted sensor measurements, and the blue solid line represents the actual measured sensor measurements. According to Fig. 90, one can notice that the predicted sensor measurements and the actually measured sensor measurements are overlapped to each other for all time. This suggests that the performance of the proposed method is reliable since the residual signal shown in Fig. 89 is generated based on the difference between the predicted measurements and the actually measured measurements.

On the other hand, the result of the proposed method in the faulty situation is shown in Fig. 91. Observing Fig. 91, the proposed method takes around 60 samples to react the sensor fault occurs at sample time 7475, and the probability of the occurrence of fault increases around sample time 7535. In addition, the probability does not go back to zero for those short-term healthy situations between each fault. This is because the proposed method needs few sample time to confirm the system is really healthy or faulty. Therefore, the proposed method will not response to a fault immediately, and it also will not response to those short-term healthy situations.

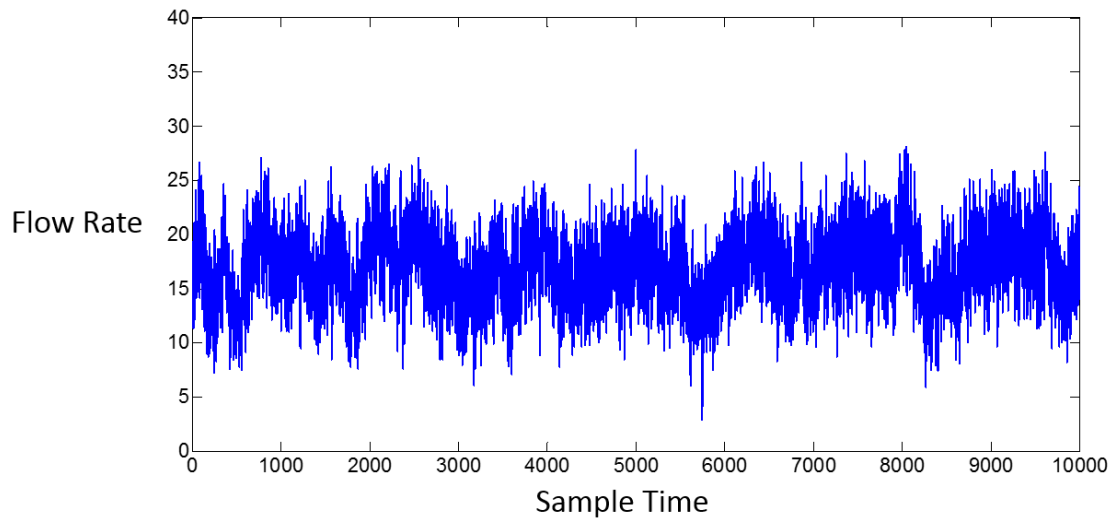


Figure 86. The training data used for reconstructing phase space of the actuator system.

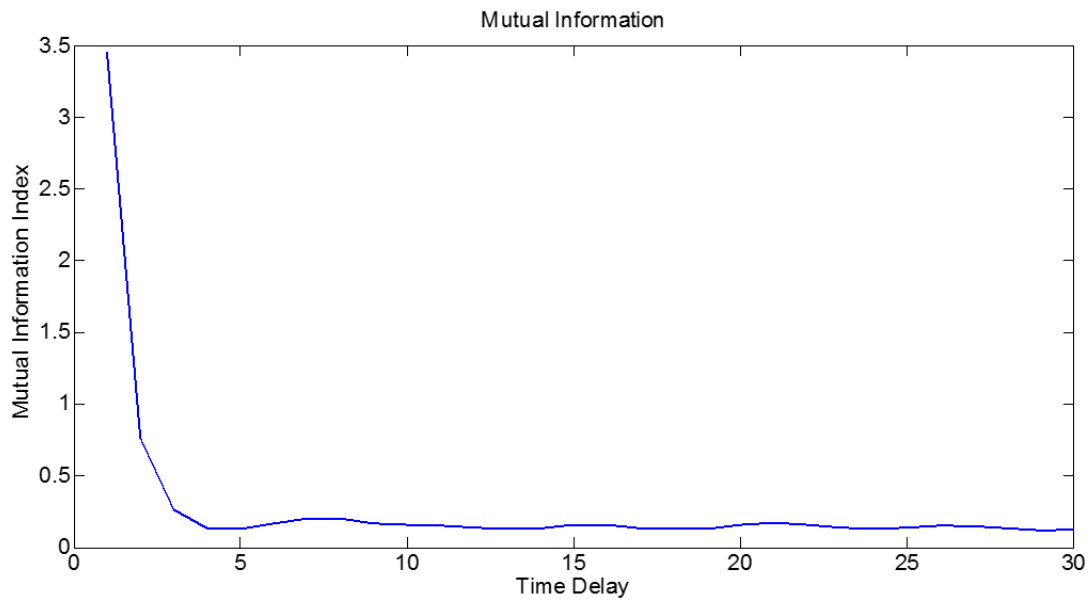


Figure 87. The result of mutual information method applied to the training data.

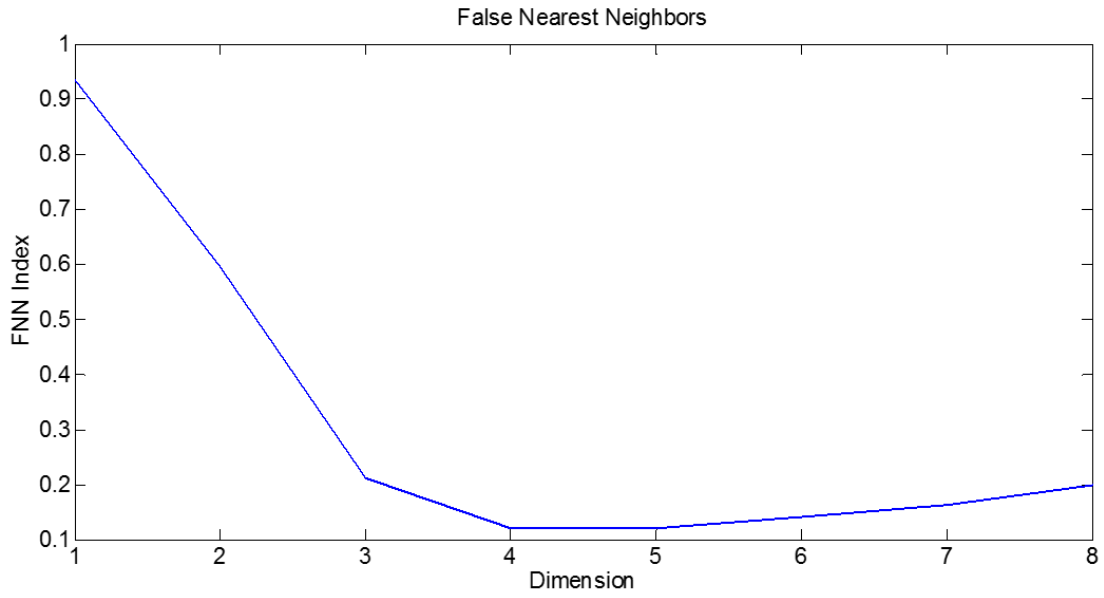


Figure 88. The result of FNNs applied to the training data.

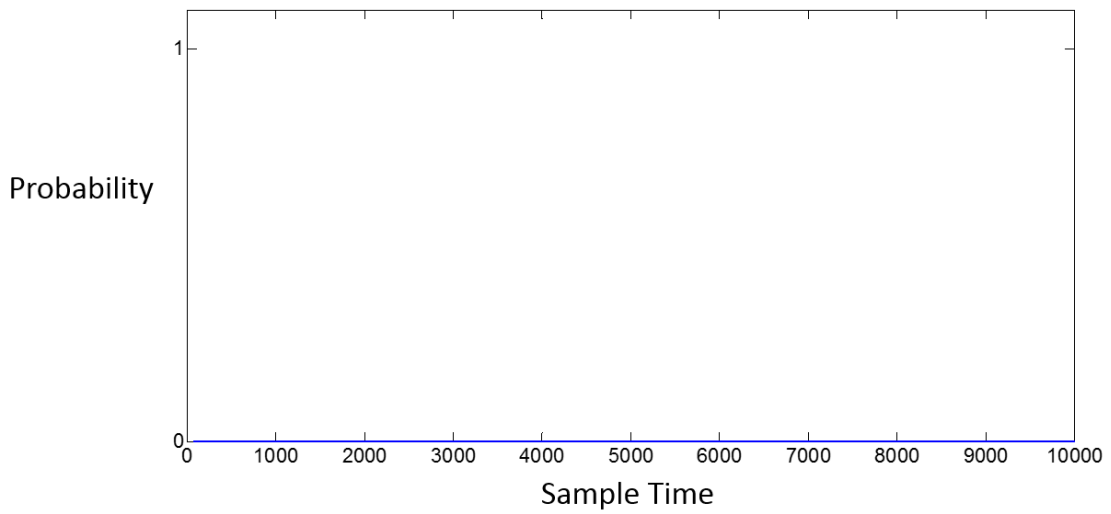


Figure 89. The result of the proposed method applied to real process data collected from the actuator system in the healthy situation.

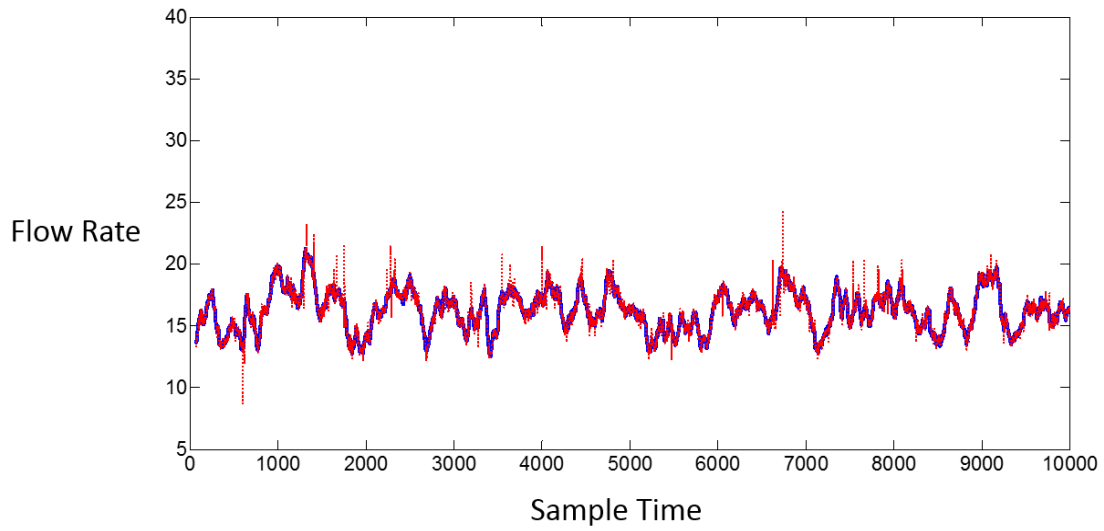


Figure 90. The result of the prediction mechanism block in the healthy situation (Red dotted line: the predicted sensor measurements; Blue solid line: the actually measured sensor measurements).

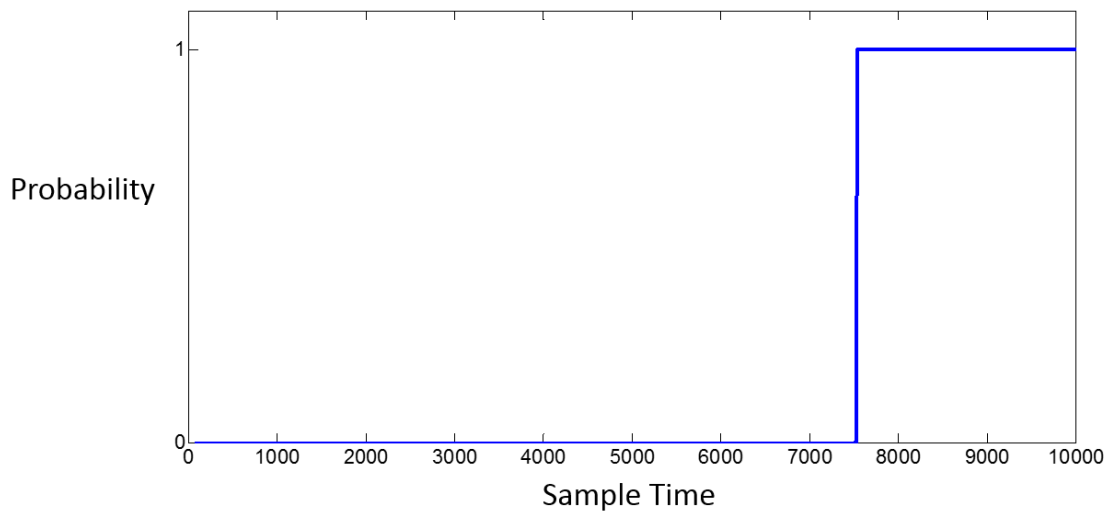


Figure 91(a). The result of the proposed method applied to real process data collected from the actuator system in the faulty situation.

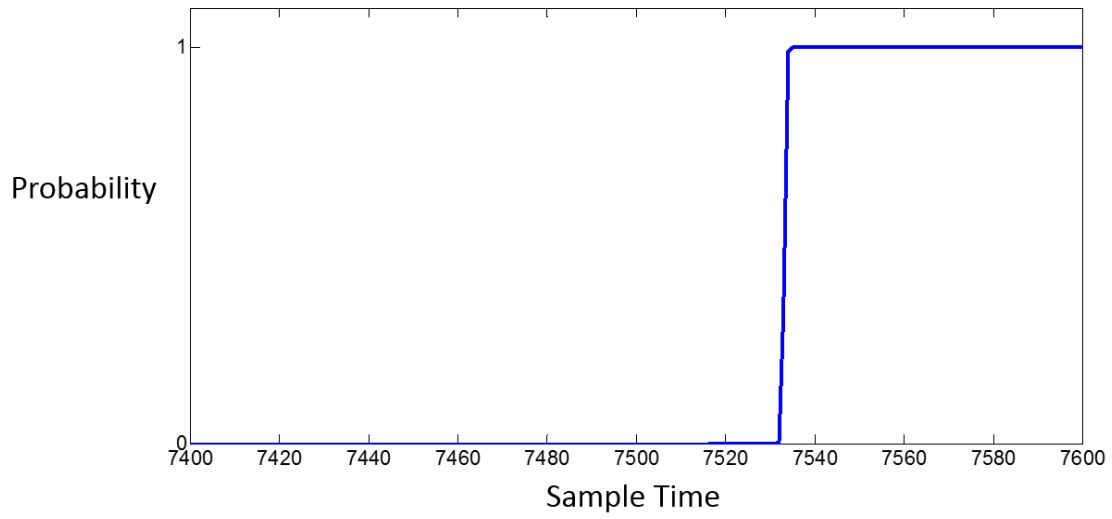


Figure 91(b). The enlargement of Fig. 91(a).

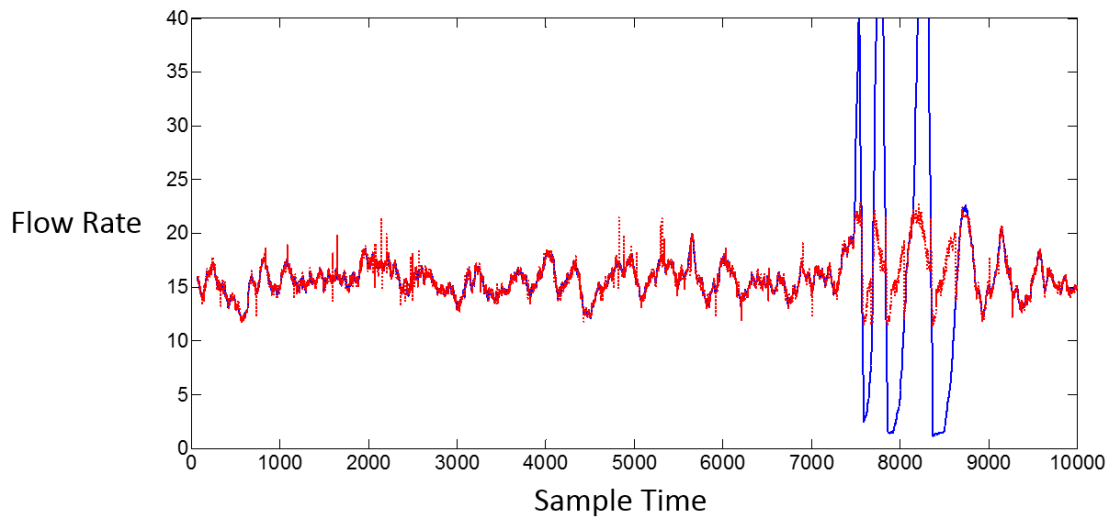


Figure 92. The result of the prediction mechanism block in the faulty situation (Red dotted line: the predicted sensor measurements; Blue solid line: the actually measured sensor measurements).



### **5.3 Results of Applying the Proposed Method to THIBAR22C Unit**

In this section, we examine the proposed method by applying it to a computer controlled heat pump/air condition/refrigeration (THIBAR22C) unit located in Qatar University. This unit is a nonlinear unit and built by EDIBON Technical Teaching Equipment Company for laboratory uses [71]. There are a total of fifteen sensors mounted in the unit for monitoring purpose. The unit is equipped with a Data Acquisition Board (DAB) for recording sensor measurements with adjustable sampling rate. Moreover, the detailed analytical model and dynamic equations of the THIBAR22C unit are not available. In our experiments, we use the data collected from a temperature sensor to examine the proposed method. Similar to previous steps, the abrupt fault situation is modeled by adding a bias, and its magnitude is set to be -40. On the other hand, the incipient fault situation is modeled by adding a uniformly developed bias with developing time 2000 samples, and the maximum magnitude of the incipient fault is set to be -40. The occurring time is set to be sample time 6001 for both abrupt and incipient fault situation.

Figure 93 shows the training data used in this experiment, and the results of applying mutual information method and FNNs method are shown in Fig. 94 and 95. According to these results, the time delay and the dimension of embedding are set to be 9 and 2 respectively. The raw residual signals produced by applying the proposed method to the healthy testing data and the second set of healthy testing data are shown in Fig. 96. In general, without prior knowledge of faulty data, a threshold is set according to the confidence level of the raw residual signal generated from the second set of healthy testing data. However, according to Fig. 96, even though the threshold is set to be 99% confidence

level, false alarms still occur since the maximum value in Fig. 96(a) is still higher than the threshold. Therefore, MHSSPT is also adopted in this case. The results of the healthy situation, the abrupt fault situation, and the incipient fault situation are shown in Fig. 96, 97, and 98. Observing these results, the probability of occurrence of faults stays low in the healthy situation in Fig. 96, which suggests that the proposed method indicates the healthy situation correctly. On the other hand, the probability suddenly increases around the occurring time of faults in both of abrupt and incipient fault situation shown in Fig. 97 and 98. This phenomenon proves the capability of the proposed method in detecting both of abrupt and incipient sensor faults.

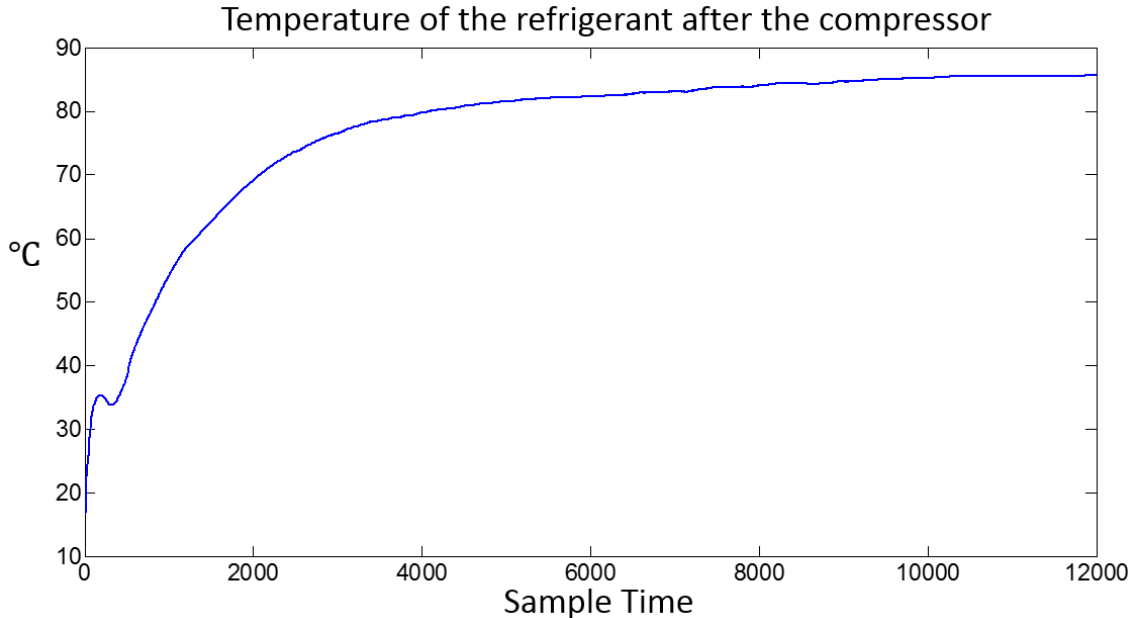


Figure 93. The training data used for reconstructing phase space of the THIBAR22C unit.

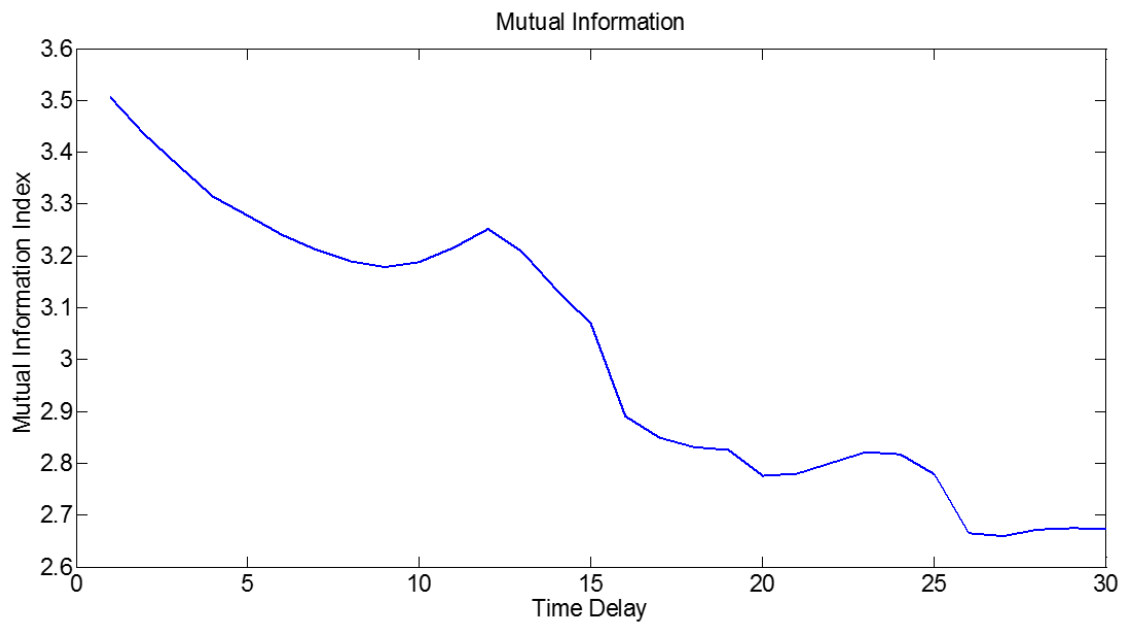


Figure 94. The result of applying mutual information method for reconstructing the phase space of THIBAR22C unit.

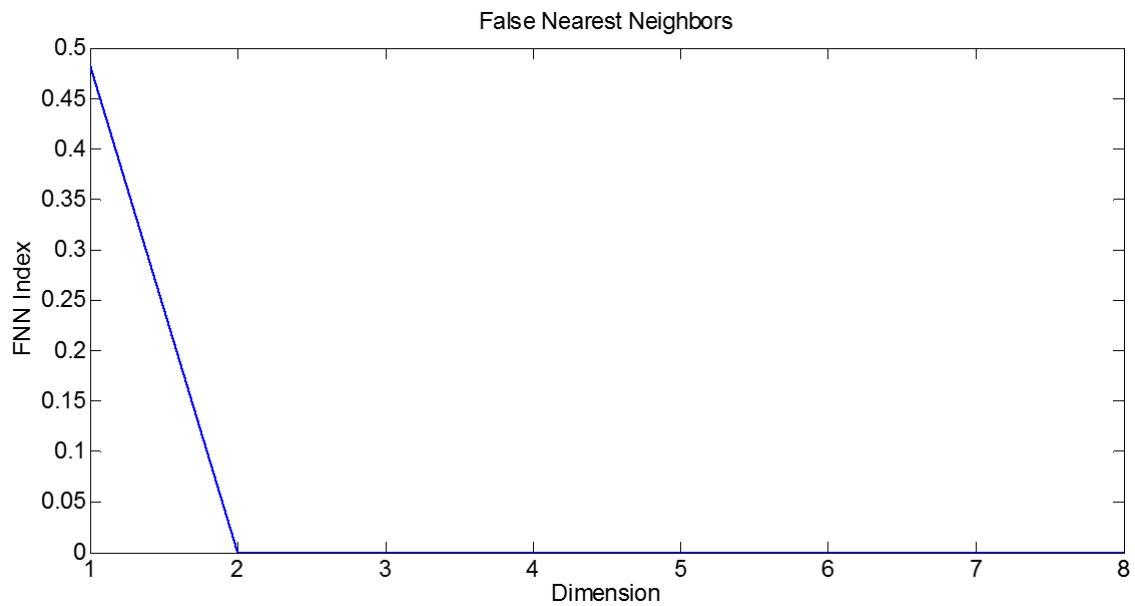


Figure 95. The result of applying FNNs method for reconstructing the phase space of THIBAR22C unit.

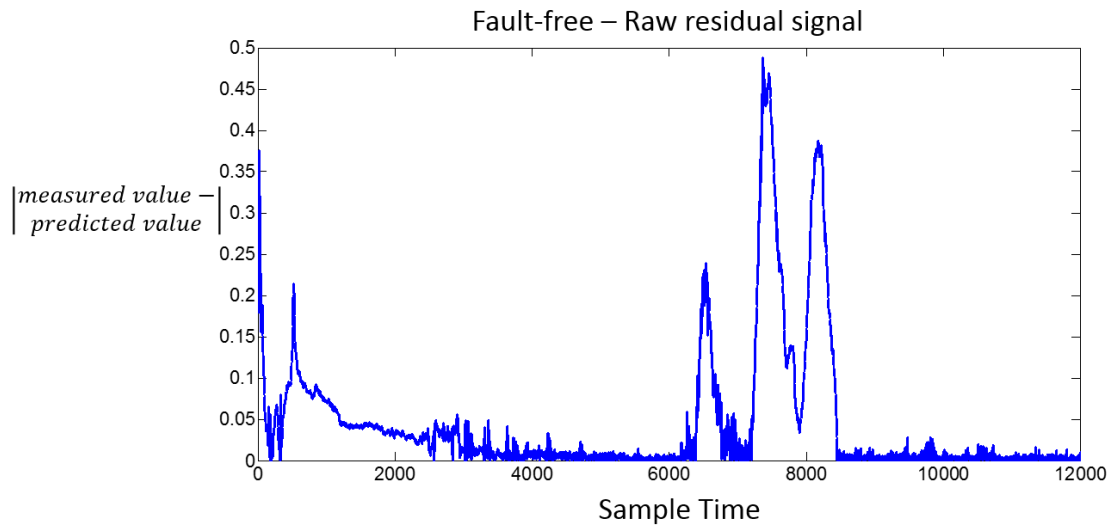


Figure 96(a). The raw residual signal produced by applying the proposed method to real process data collected from the THIBAR22C unit in the healthy situation.

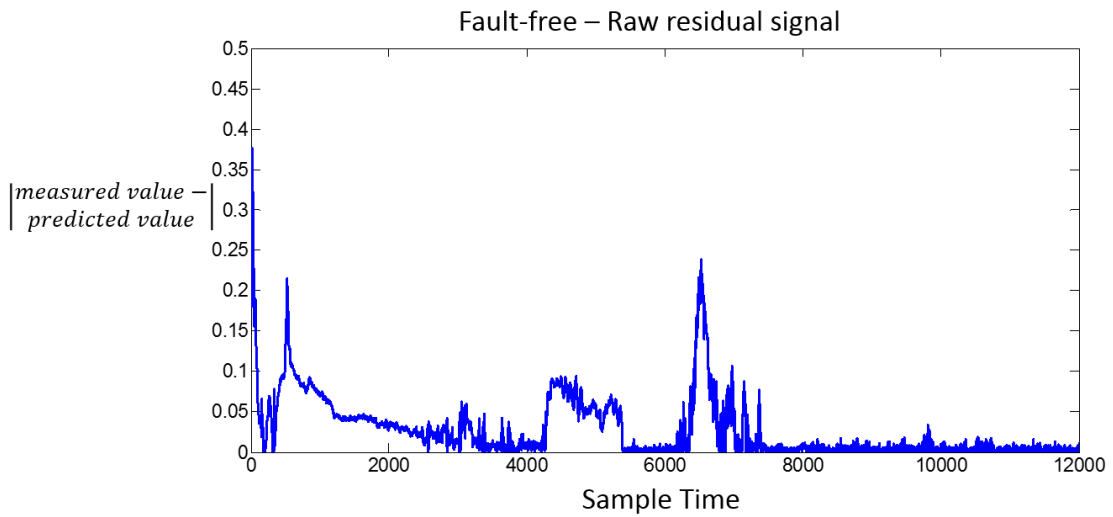


Figure 96(b). The raw residual signal produced by applying the proposed method to the second healthy testing data collected from the THIBAR22C unit.

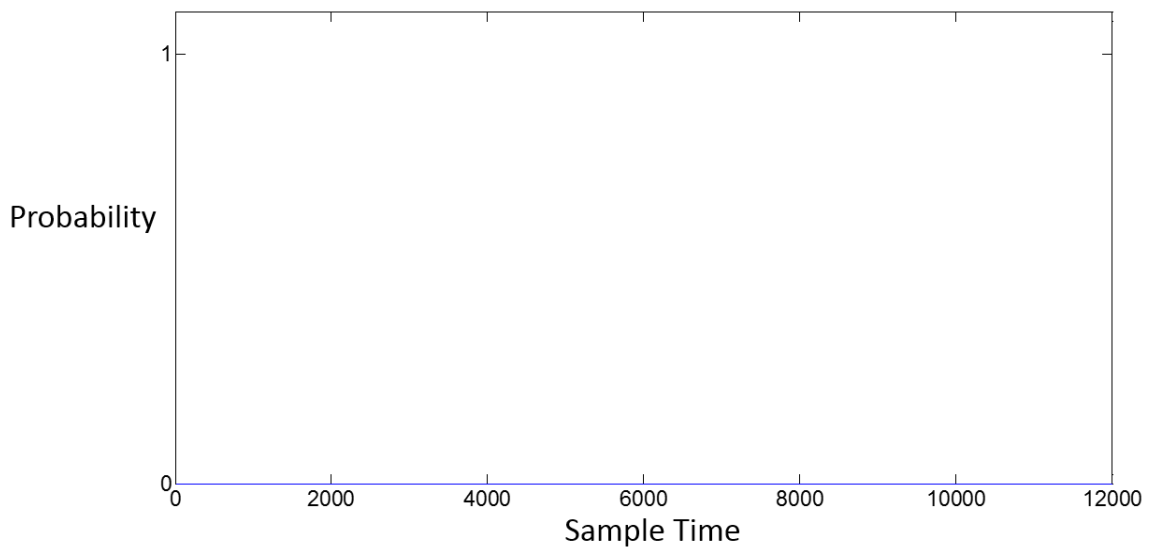


Figure 97. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the healthy situation.

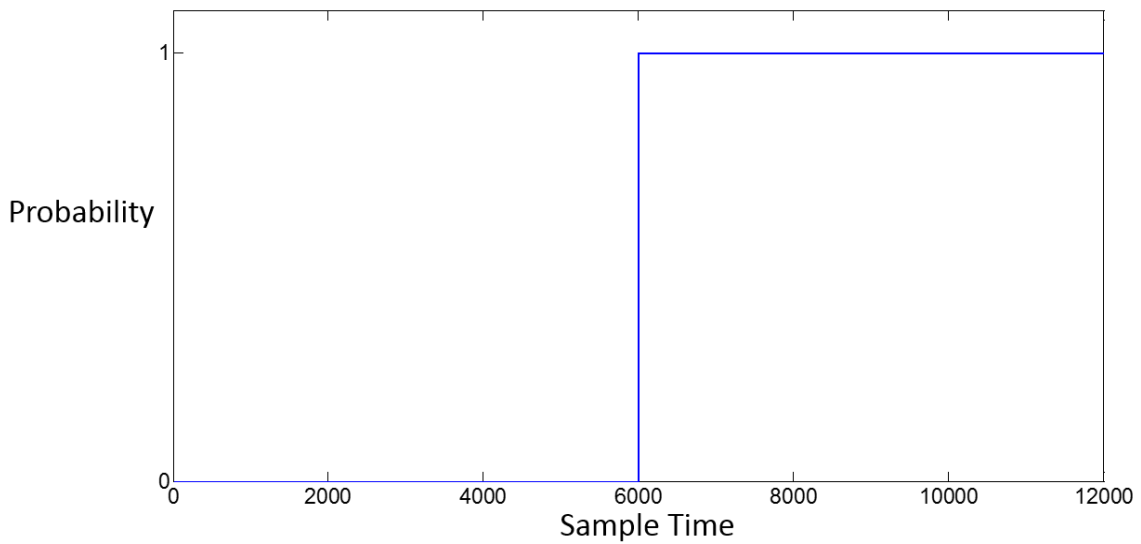


Figure 98. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the abrupt fault situation.

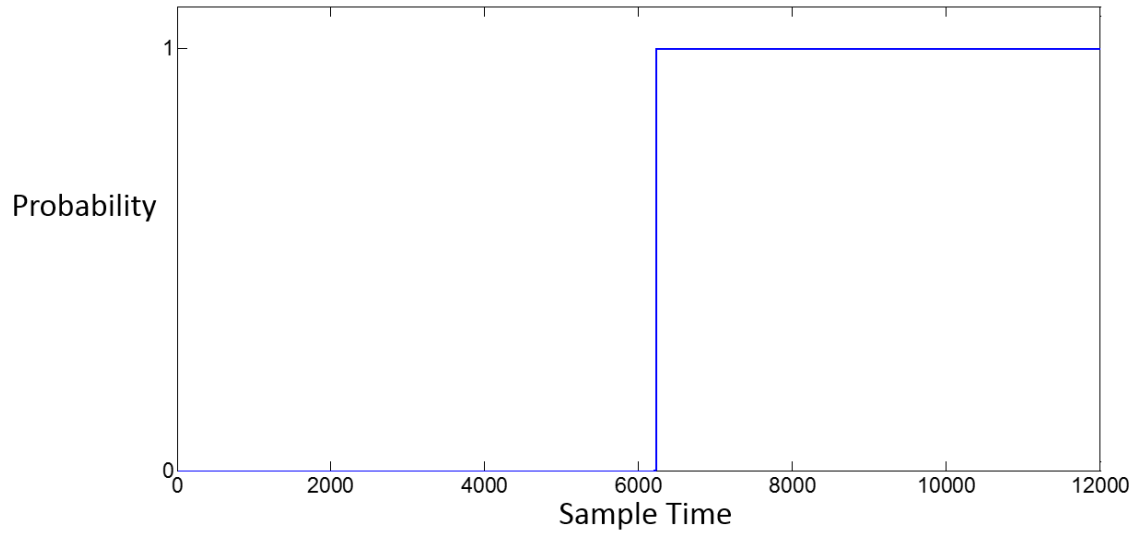


Figure 99. The result of the proposed method applied to real process data collected from the THIBAR22C unit in the incipient fault situation.

## 6. CONCLUSIONS

This dissertation provides a unified approach for detecting and isolating multiple sensor faults. The proposed method deals with the FDI problem by reconstructing the phase space by the measurements collected from each sensor. Then, the reconstructed phase space is used to construct the residual generator for detecting the sensor fault occurring in its corresponding sensor, and the isolation part is done simultaneously. The main contribution of the proposed method is its detectability and isolability of multiple sensor faults in nonlinear cases. In addition, unlike other dimension reduction based methods, the proposed method does not require that the dimension of the applied system must be smaller than the number of measurement channels. This advantage eliminates the limitation of the minimum number of sensors and greatly lowers the economic cost of setting sensors. Furthermore, the conservation/dissipation property in phase space is adopted to deal with the problem of setpoint changes and extend the proposed method from stationary to nonstationary applications.

In our simulation, a three tanks system is modeled for testing the proposed method. The three tank system is a nonlinear system with 2 inputs and 3 outputs used in chemical industry. There are two scenarios for the faulty situation: abrupt fault and incipient fault, and our results show that the proposed method is capable to indicate both of them when the sensor fault occurs. Moreover, unlike traditional way to present the residual signal by setting a threshold, we introduce the MHSSPT algorithm to further process and present the residual signals by probability in order to avoid the problem of false alarm and miss detection. In the other words, instead of faulty or healthy, our residual signals show the

probability of the occurrence of sensor fault. The proposed approach can be applied to process industries in order to enhance the product quality and the process safety such as the process of oil refining in the petroleum industry.

In our experiments, a sugar factory actuator system is used to examine the proposed method. This system is a nonlinear system with 4 inputs and 6 outputs proposed by Koscielny et al. for testing FDI systems [66]. Koscielny et al. releases two types of data sources for this system: Simulink-Matlab model and real process data with artificial generated faults. Moreover, the proposed method is also applied to the THIBAR22C unit located in Qatar University. The THIBAR22C unit is a computer controlled heat pump/air conditioning/refrigeration unit built by EDIBON Technical Teaching Equipment Company for laboratory use. According to our results, the proposed method is capable to indicate both of healthy and faulty situations. These results further confirm that the proposed method is able to deal with not only simulated data but also real process data.



## REFERENCES

- [1] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods," *Computers and Chemical Engineering*, vol. 27, pp. 293-311, 2003.
- [2] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri, "A review of process fault detection and diagnosis Part II: Qualitative modes and search strategies," *Computers and Chemical Engineering*, vol. 27, pp. 313-326, 2003.
- [3] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, K. Yin, "A review of process fault detection and diagnosis Part III: Process history based methods," *Computers and Chemical Engineering*, vol.27, pp. 327-346, 2003.
- [4] S.X. Ding, *Model-based Fault Diagnosis Techniques*. Leipzig: Springer, 2008.
- [5] M.N. Nashalji, M.A. Schoorehdeli, and M. Teshnehlab, "Fault detection of the Tennessee Eastman Process using improved PCA and neural classifier," *International Journal of Electrical & Computer Sciences*, vol. 9, pp. 481-486, 2009.
- [6] R. Isermann, "Trends in the application of model-based fault detection and diagnosis of technical processes," *Control Eng. Practice*, vol. 5, pp. 709-719, 1997.
- [7] D.V. Schrick, "Remarks on terminology in the field of supervision, fault detection and diagnosis," in *Proc. IFAC Symp. Fault Detection, Supervision Safety for Techn. Process. (SAFEPROCESS)*, pp. 959-964, 1997.

- [8] I. Hwang, S. Kim, Y. Kim, and C.E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Trans. on Control Systems Technology*, vol. 18, pp. 636-653, 2010.
- [9] P.D. Christofides, J.F. Davis, N. H. El-Farra, D. Clark, K.R.D. Harris, and J.N. Gipson, "Smart plant operations: Vision, progress and challenges," *AIChE Journal*, vol. 53, pp. 2734-2741, 2007.
- [10] U.S. Chemical Safety and Hazard Investigation Board, *Refinery Fire and Explosion and Fire. BP Texas City March 23, 2005, Final Investigation Report*, Report No. 2005-04-I-TX, March, 2007.
- [11] A.S. Willsky, "A survey of design methods for failure detection in dynamic systems," *Automatic*, vol. 12, pp. 601-611, 1976.
- [12] J.J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, Inc., 1998.
- [13] J.J. Downs and E.F. Vogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, pp. 245-255, 1993.
- [14] N.F. Thornhill, S.C. Patwardhan and S.L. Shah, "A continuous stirred tank heater simulation model with applications," *Journal of Process Control*, vol. 18, pp. 347-360, 2008.
- [15] E. Chow and A.S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. on Automatic Control*, vol. AC-29, pp. 603-614, 1984.

- [16] J.J. Gertler, "Fault detection and isolation using parity relations," *Control Eng. Practice*, vol. 5, pp. 653-661, 1997.
- [17] J.M. Koscielny, M. Bartys, and M. Syfert, "Method of multiple fault isolation in large scale systems," *IEEE Trans. on Control Systems Technology*, vol. 20, pp. 1302 - 1310, 2012.
- [18] D.G. Luenberger, "Canonical forms for linear multivariable systems," *IEEE Trans. on Automatic Control*, vol. AC-12, pp. 290-293, 1967.
- [19] S.X. Ding, P. Zhang, A. Naik, E.L. Ding, and B. Huang, "Subspace method aided data-driven design of fault detection and isolation systems," *Journal of Process Control*, vol. 19, pp. 1496-1510, 2009.
- [20] S. Qin and W. Li, "Detection and identification of faulty sensors in dynamic processes," *AIChE Journal*, vol. 47, pp. 1581-1593, 2001.
- [21] J. Dong, M. Verhaegen, and F. Gustafsson, "Robust fault detection with statistical uncertainty in identified parameters," *IEEE Trans. on Signal Processing*, vol. 60, pp. 5064-5076, 2012.
- [22] R. Sharifi and R. Langari, "Isolability of faults in sensor fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 25, pp. 2733-2744, 2011.
- [23] I.K. Fodor, "A survey of dimension reduction techniques," US Department of Energy, 2002. Available: <https://e-reports-ext.llnl.gov/pdf/240921.pdf>
- [24] J.E. Jackson, *A User's Guide to Principal Components*. New York: John Wiley and Sons, 1991.
- [25] I.T. Jolliffe, *Principal Component Analysis*. Berlin: Springer-Verlag, 1986.

- [26] W. Li, H. Yue, S. Valle-Cervantes, and S. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control*, vol. 10, pp. 471-486, 2000.
- [27] F.R. Malinowski, *Factor Analysis in Chemistry*. New York: Wiley-Interscience, 1991.
- [28] R. Sharifi and R. Langari, "Sensor fault diagnosis with a probabilistic decision process," *Mechanical Systems and Signal Processing*, vol. 34, pp. 146-155, 2012.
- [29] M.A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, pp. 233-243, 1991.
- [30] M.A. Kramer, "Autoassociative neural networks," *Computers and Chemical Engineering*, vol. 16, pp. 313-328, 1992.
- [31] D.H. Ballard, "Modular learning in neural networks," in *Proc. Conf. on AI (AAAI-87)*, vol. 1, pp. 279-284, 1987.
- [32] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303-314, 1989.
- [33] M. Najafi, *Use of Auto Associative Neural Networks for Sensor Diagnostic*, M.S. Thesis, Texas A&M University, College Station, Texas, 2003.
- [34] J.W. Hines, R.E. Uhrig, and D.J. Wrest, "Use of autoassociative neural networks for signal validation," *Journal of Intelligent and Robotic Systems*, vol. 21, pp. 143-154, 1998.
- [35] D.J. Wrest, J.W. Hines, and R.E. Uhrig, "Instrument surveillance and calibration verification through plant wide monitoring using autoassociative neural networks," in *Proceedings of the 1996 American Nuclear Society International*

*Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*, University Park, PA, pp. 6-9, 1996.

- [36] R. Sharifi, *Sensor Fault Diagnosis Using Principal Component Analysis*, Ph.D Dissertation, Texas A&M University, College Station, Texas, 2009.
- [37] S.W. Choi, E.B. Martin, and A.J. Morris, "Fault detection based on a maximum-likelihood principal component analysis (PCA) mixture," *Industrial and Engineering Chemistry Research*, vol. 44, pp. 2316-2327, 2005.
- [38] M.E. Tipping and C.M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 61, pp. 611-622, 1999.
- [39] M.E. Tipping and C.M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, pp. 443-482, 1999.
- [40] N. Kambhatla and T.K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, pp. 1493-1516, 1997.
- [41] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323-2326, 2000.
- [42] T. Lin, H. Zha, and S.U. Lee, "Riemannian manifold learning for nonlinear dimensionality reduction," *Lecture Notes in Computer Science*, vol. 3951, pp. 44-55, 2006.
- [43] T. Lin and H. Zha, "Riemannian manifold learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 796-808, 2008.

- [44] A. Escobet, A. Nebot, and F.E. Cellier, "Fault diagnosis system based on fuzzy logic: Application to a valve actuator benchmark," *Journal of Intelligent and Fuzzy Systems*, vol. 22, pp. 155-171, 2011.
- [45] L. Zhang, G. Xiong, H. Liu, H. Zou, and W. Guo, "Bearing fault diagnosis using multi-scale entropy and adaptive neuro-fuzzy interface," *Expert Systems with Applications*, vol. 37, pp. 6077-6085, 2010.
- [46] D. Chelidze, J.P. Cusumano, and A. Chatterjee, "A dynamical systems approach to damage evolution tracking, Part 1: Description and experimental application," *Journal of Vibration and Acoustics*, vol. 124, pp. 250-257, 2002.
- [47] J.P. Cusumano, D. Chelidze, and A. Chatterjee, "A dynamical systems approach to damage evolution tracking, Part 2: Model-based validation and physical interpretation," *Journal of Vibration and Acoustics*, vol. 124, pp. 258-264, 2002.
- [48] D. Chelidze, "Identifying multidimensional damage in a hierarchical dynamical system," *Nonlinear Dynamics*, vol. 37, pp. 307-322, 2004.
- [49] D. Chelidze and M. Liu, "Dynamical systems approach to fatigue damage identification," *Journal of Sound and Vibration*, vol. 281, pp. 887-904, 2005.
- [50] D. Chelidze and M. Liu, "Multidimensional damage identification based on phase space warping: an experimental study," *Nonlinear Dynamics*, vol. 46, pp. 61-72, 2006.
- [51] D. Chelidze and J.P. Cusumano, "Phase space warping: Nonlinear time series analysis for slowly drifting systems," *Philosophical Trans. of the Royal Society A*, vol. 364, pp. 2495-2513, 2006.

- [52] D. Chelidze and M. Liu, "Reconstructing slow-time dynamics from fast-time measurements," *Invited paper, Philosophical Trans. of the Royal Society A*, vol. 366, pp. 729-745, 2008.
- [53] J.D. Farmer and J.J. Sidorowich, "Exploiting chaos to predict future and reduce noise," *Physical Review Letters*, vol. 59, pp. 845-848, 1987.
- [54] T. Sauer, J.A. Yorke, and M. Casdagli, "Embedology," *Journal of Statistical Physics*, vol. 65, pp. 579-616, 1991.
- [55] H. Whitney, "Differentiable manifolds," *Annals of Mathematics*, vol. 37, pp. 645-680, 1936.
- [56] F. Takens, "Detecting strange attractors in turbulence," *Lecture Notes in Mathematics*, vol. 898, pp. 366-381, 1981.
- [57] A.M. Fraser and H.L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A*, vol. 33, pp. 1134-1140, 1986.
- [58] M.B. Kennel, R. Brown, and H.D.I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Phys. Rev. A*, vol. 45, pp. 3403-3411, 1992.
- [59] E.N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130-141, 1963.
- [60] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Westford, MA: Prentice Hall, 1993.
- [61] D. Chelidze and W. Zhou, "Smooth orthogonal decomposition-based vibration mode identification," *Journal of Sound and Vibration*, vol. 209, pp. 461-473, 2006.

- [62] L. Pecora, L. Moniz, J. Nichols, and T. Carroll, "A unified approach to attractor reconstruction," *Complex Dynamics in Physiological Systems: From Heart to Brain*, vol. 1, pp. 3-19, 2009.
- [63] Q. Liu, S. Islam, I. Rodriguez-Iturbe, and Y. Le, "Phase-space analysis of daily streamflow: characterization and prediction," *Advances in Water Resources*, vol. 21, pp. 463-475, 1998.
- [64] D. Rothman. *12.006J Nonlinear Dynamics I: Chaos, Fall 2006. (MIT OpenCourseWare: Massachusetts Institute of Technology)*. Available: <http://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-006j-nonlinear-dynamics-i-chaos-fall-2006>
- [65] Amira DTS200: "Laboratory setup three tank system", Amira Gmbh, Duisburgh, Germany, 1996.
- [66] J.M. Koscielny, M. Bartys, P. Wasiewicz, A. Ostasz, and M. Syfert, "Benchmark definition file," Warsaw University of Technology 2002. Available: <http://diag.mchtr.pw.edu.pl/damadics/>
- [67] R.H. Chen, H.K. Ng, J.L. Speyer, L.S. Guntur, and R. Carpenter, "Health monitoring of a satellite system," *Journal of Guidance, Control, and Dynamics*, vol. 29, pp. 593-605, 2006.
- [68] D.P. Malladi and J.L. Speyer, "A generalized Shiriyayev sequential probability ratio test," *IEEE Trans. on Automatic Control*, vol. AC-44, pp. 1522-1534, 1999.



- [69] J.M. Koscielny, M. Bartys, P. Wasiewicz, A. Ostasz, and M. Syfert, "Using DAMADICS actuator benchmark library (DABLib)," Warsaw University of Technology, 2002. Available: <http://diag.mchtr.pw.edu.pl/damadics/>
- [70] J.M. Koscielny, M. Bartys, P. Wasiewicz, A. Ostasz, and M. Syfert, "Lubin sugar factory data description file," Warsaw University of Technology, 2002. Available: <http://diag.mchtr.pw.edu.pl/damadics/>
- [71] EDIBON corp., *Computer Controlled Heat Pump+Air Conditioning+Refrigeration Unit with Cycle Iversion Valve, Manual*, 2013. Available: <http://www.edibon.com/products/?area=thermodynamicsthermotechnics&subarea=heatpumps>