A Dependable Actuator and Sensor Health Monitoring System

by

Mehdi Zabihi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Name: Guangjun Liu Title: Professor Department: Aerospace Engineering University: Toronto Metropolitan University
Supervisor(s):	Name: Amir Khajepour Title: Professor Department: Mechanical and Mechatronics Engineering
Internal Member:	Name: William Melek Title: Professor Department: Mechanical and Mechatronics Engineering
Internal Member:	Name: Armaghan Salehian Title: Associate Professor Department: Mechanical and Mechatronics Engineering
Internal-External Member:	Name: Shoja'eddin Chenouri Title: Professor Department: Statistics and Actuarial Science

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Impressive progress in vehicle control technologies has equipped modern vehicles with advanced driver assistance and stability control systems to help the driver handle unfavorable driving conditions. These control systems rely on sensor measurements and/or information estimated based on these measurements to generate control commands for vehicle actuators. Therefore, any failure in sensors and actuators can degrade the performance of these control systems and cause instability in vehicle operation. Sensor failures make the control systems generate undesired control commands and actuator failures prevent desired control commands from being applied. To achieve safe and satisfactory vehicle operation, a real-time health monitoring system is crucial for vehicle sensors and actuators. The reliability of health monitoring is determined by its sensitivity to faults and robustness to disturbances. A reliable health monitoring system is responsible for timely detecting the occurrence of a fault in the target vehicle, accurately identifying the source of the fault, and properly determining the type and magnitude of the fault. This information is crucial for reconstructing sensor data or scaling actuator output, which ultimately could result in a fault-tolerant vehicle system.

This thesis proposes a hybrid model/data fault detection and diagnosis system to monitor the health status of any sensor or actuator in a vehicle. The proposed approach works based on residuals generated by comparing sensor measurements or control inputs with their estimations. The estimations are obtained by a hybrid estimator that is developed based on the integration of model-based and data-driven estimators to leverage their strengths. Due to the poor performance of data-driven estimators in unknown conditions, a self-updating dataset is proposed to learn new cases. Estimation based on updated datasets necessitates the use of data-driven estimators that do not require any pre-training. As case studies, the proposed hybrid fault detection and diagnosis system is applied to a vehicle's lateral acceleration sensor and traction motor. The experimental results show that the hybrid estimator outperforms the model-based and data-driven estimators used individually. These results also confirm that the proposed hybrid fault detection and diagnosis system can detect the faults in the target sensor or actuator, and then reconstruct the healthy value of the faulty sensor or find the failure level of the faulty actuator.

For cases where a set of sensors and actuators should be monitored to evaluate their health status, this thesis develops a general data-driven health monitoring system to detect, isolate, and quantify faults in these components. This method checks the coherency among the target vehicle's variables by using the vehicle's data. The coherency among the vehicle variables means that the variables reflect the physical principles governing the target vehicle's motion and the causality between its states. Each variable corresponds to a component in the vehicle. When a fault occurs in one of the vehicle's components, the coherency among the variables is no longer valid. This idea is incorporated to detect faults. After fault detection, to isolate the faulty component, the developed system explores the coherency in the subsets of the vehicle's variables to find which variable is not coherent with others. Once the faulty component is determined, the health monitoring system uses the remaining healthy components to reconstruct the true value of the faulty sensor or find the failure level of the faulty actuator. The experimental results show that the developed health monitoring system appropriately detects, isolates, and quantifies faults in the test vehicle's sensors and actuators.

The developed data-driven health monitoring system requires a pre-collected dataset for each vehicle to monitor the health status of its sensors and actuators. To relax this requirement, this thesis proposes a universal health monitoring system for the vehicle IMU sensor (measuring the longitudinal/lateral accelerations and yaw rate) by involving vehicle parameters in the health monitoring process. The proposed universal health monitoring system is able to monitor the health status of the target vehicle's IMU sensor using the other vehicles' data. The performance of the universal health monitoring system is evaluated by simulations. The simulation results are in line with what is expected.

Acknowledgements

Foremost, I would like to express my sincerest gratitude to my supervisor, Prof. Amir Khajepour for his exceptional support, guidance, and encouragement throughout my PhD program at the University of Waterloo. I am very grateful to have had the privileged opportunity to work under his supervision, and it is to him that I owe my present academic skills, vision of the profession, and insights into engineering ethics.

I would also like to further thank the members of my thesis committee, Prof. Guangjun Liu, Prof. William Melek, Prof. Armaghan Salehian, and Prof. Shoja'eddin Chenouri for agreeing to be in my PhD committee. Their insightful feedback and constructive criticism that helped me to shape and refine my research. I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC), Ontario Research Fund (ORF), and General Motors for their financial support. Special thanks to Dr. Bakhtiar Litkouhi, and Dr. Alireza Kasaizadeh at the GM Research and Development Center in Warren, MI, USA, and Dr. Hojjat Izadi at GM Canadian Technical Center for their technical support and valuable comments in improving my research.

I would like to express my profound gratitude to Dr. Mohammad Pirani for the spiritual and scientific collaborations that we have shared over the years, which have greatly inspired my research. His wealth of knowledge and expertise has truly inspired and motivated me to achieve my goals. Additionally, I am grateful to Dr. Ehsan Hashemi and Dr. Reza Valiollahimehrizi for their valuable technical support and advice during my PhD research. I would like to thank the technician in the MVS laboratory, Jeff Graansma, for helping me in the road experiments. I have had a great time at Waterloo because of some amazing people who have been a valuable source of assistance, affection, and friendship. I highly appreciate my friends including Amin Habibnejad Korayem, Reza Hajiloo, Amir Fathazam, Ali Shahidi, Ehsan Mohammadbagher, and Mobin Khamooshi.

Most importantly, none of this would have been possible without the endless love and patience of my wife, Fatemeh Motaghedi. Her encouragement gave me the strength to persevere through the most challenging times, and her sacrifices and understanding allowed me to focus on my work. I am forever indebted to her for the sacrifices she made for me. I would also like to express my heartfelt appreciation to my parents, Adolmohammad Zabihi

and Soheila Soltani, and my sister, Zahra Zabihi, for supporting me spiritually throughout my life and for their encouragement during this endeavor. Their unwavering belief in me gave me the courage to pursue my dreams.

Dedication

Dedicated to my beloved spouse, parents, and sister.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

As the complexity of vehicles increases, they are more susceptible to various faults, some of which may have life-threatening consequences [1]. Therefore, health monitoring is crucial to prevent these disastrous consequences and provide added safety to vehicle operations. Vehicles, similar to any other mechatronic system, work based on two main processes: sensing and acting. In the sensing process, different sensors are applied to measure some of vehicle states, and in the acting process, actuators apply the control actions to vehicles [2]. Sensors and actuators are two of the main sources of faults in vehicles. If a sensor becomes faulty, controllers generate control actions based on wrong measurements, and if an actuator becomes faulty, desired control actions are not applied properly. In autonomous vehicles, since there is no human to monitor the system, the importance of health monitoring becomes more obvious.

Fault Detection and Diagnosis (FDD) generally works based on information redundancy. One method to achieve information redundancy is analytical redundancy utilizing the correlations among system variables that can be either expressed by a mathematical model in an explicit form or implicitly through input-output data. Model-based and data-driven approaches are two of the main classes of FDD methods. Each class has its own

advantages and disadvantages. In model-based methods, since only limited information is needed to detect and diagnose a fault, their computational cost is low. This fact makes them applicable for real-time applications. In addition, these methods can detect and diagnose unknown faults. However, model-based FDD methods require an explicit model for the target system and the accuracy of the model affects the performance of such methods. Unlike model-based methods, data-driven approaches do not require an explicit model of the system. Instead, they need a significant amount of high-fidelity data in all operational states of the vehicle making them costly. The performance of data-based methods is dependent on the training data, and they do not have the ability to detect unknown faults [3, 4]. To tackle the disadvantages and to gain the advantages of both methods, hybrid FDD approaches are being developed. These approaches, made by the integration of two or more fault diagnosis methods, can outperform each working individually.

1.2 Objectives

The main objective of this research is to develop a reliable health monitoring system for vehicle sensors and actuators. This objective entails the development of fault detection, fault isolation, and fault reconstruction tasks. The fault detection system is responsible for determining whether there is any fault in the vehicle. After the fault is detected, the fault isolation system localizes the source of the fault. Finally, the fault quantification system determines the type and magnitude of the fault that can be used to reconstruct sensor data or find the level of actuator failure.

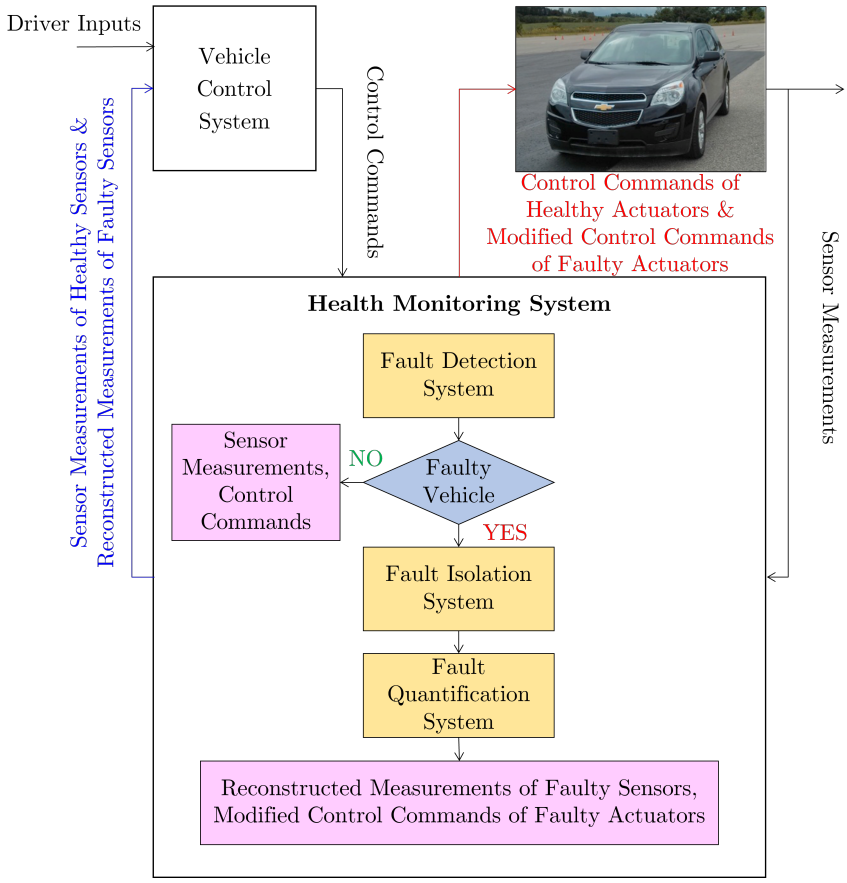

Figure 1.1: The general scheme of a health monitoring system's contribution to a vehicle system.

Fig. 1.1 illustrates how a reliable health monitoring system works with a vehicle control system. The ultimate goal of the health monitoring system is to provide fault-free sensory information for the vehicle control system in the event of sensor failure and/or adjust control commands to achieve desired actuator output in case of actuator failure. The proposed health monitoring system is made of fault detection, isolation, and quantification systems. Using analytical redundancy provided by sensor measurements and control commands, the fault detection system always searches for anomalies in the target vehicle. If there is no fault in the system, the vehicle control system receives the sensor measurements and the vehicle receives control commands without any changes. If the fault detection system

notifies that a fault occurs in the vehicle, the fault isolation system specifies the faulty component. If the faulty component is a sensor, the fault quantification system quantifies the fault and reconstructs the healthy value of that sensor. In this condition, the vehicle control system receives the reconstructed measurements of the faulty sensor instead of its faulty measurements. If the faulty component is an actuator, the fault quantification system determines the type and magnitude of the fault to find the failure level of the faulty actuator. In partial failures when the faulty actuator provides a portion of control commands, the desired actuator output may be achieved by scaling the corresponding control commands.

Furthermore, this research attempts to accomplish its objective using common sensors and actuators in commercial vehicles without the availability of prior knowledge of parameters such as road friction coefficient and road bank/grade angles. Hence, the health monitoring systems developed in this research can be applied to real-world applications.

1.3 Thesis Outline

A literature review on relevant fault detection and diagnosis methods with a focus on sensor and actuator fault detection is presented in the second chapter. These methods are categorized into three classes: model-based, data-driven, and hybrid methods.

In the third chapter, a reliable hybrid model/data fault detection and diagnosis system is developed for any vehicle sensor and actuator. This system works based on information redundancy provided by a hybrid estimator operating by the integration of a model-based observer with a data-driven estimator. For sensor fault detection, the fault detection and diagnosis system compares the sensor measurements with their estimations provided by the hybrid estimator. For actuator fault detection, this system compares the desired control actions determined by vehicle controllers or a driver with the estimations of actuator output obtained by the hybrid estimator. After fault detection, the estimations of the hybrid estimator are used to reconstruct sensor data or find the level of actuator failure. To evaluate the performance of the proposed hybrid fault detection and diagnosis system, it is applied to a vehicle's lateral acceleration sensor and traction motor. The results

of experimental tests conducted on an all-wheel-drive vehicle validate that the proposed system can successfully detect and quantify faults in the target component.

Chapter four proposes a general data-driven health monitoring system for a set of vehicles' sensors and actuators. The proposed health monitoring system generally works based on the coherency among a vehicle's variables including its sensors' measurements and actuators' actions. The performance of this system is evaluated through experimental tests conducted on an electric all-wheel-drive vehicle. The results of these tests validate that the proposed health monitoring system can successfully detect, isolate, and quantify faults in vehicles' sensors and actuators.

Chapter five focuses on adding universality to the data-driven health monitoring system developed in Chapter four. Universality in the health monitoring application provides an opportunity to monitor the health status of a vehicle by using the other vehicles' data. Therefore, vehicle parameters should participate in the health monitoring process, and the governing equations of motion at the vehicle CG are used to find how it should be done. The performance of the proposed system is evaluated by CarSim simulations with six high-fidelity vehicle models. In these simulations, the universal health monitoring system monitors the health status of the IMU sensor in a vehicle by using the pre-collected data of the five other vehicles. The simulation results show that the universal health monitoring system for an IMU sensor is achieved with some limitations.

Chapter 2

Literature Review

2.1 Introduction

Automotive systems work based on measurements provided by sensors and control actions generated by actuators to achieve control goals. Any failure in sensors and actuators adversely affects system performance and may cause system degradation or instability [1]. A sensor fault provides incorrect measurements that no longer correspond to a target physical variable [5]. A sensor health monitoring system is responsible for determining when a fault occurs, which sensor is faulty, and whether it can be reconstructed through other measurements. An indication of an actuator fault is its deviation from the desired control actions. In a complete failure, the actuator produces no action, whereas, in a partial failure, the actuator provides a portion of the commanded action [6]. A vehicle actuator health monitoring system identifies the faulty actuator and the deviation from the expected value.

Fault detection and diagnosis (FDD) generally works based on information redundancy. A traditional method to achieve information redundancy is physical redundancy that can be provided by duplication of devices [7]. Not only is this duplication very costly, but also it is not always practical [8–10]. Duplication of hardware itself can be another source of fault because it makes the system more complex and more susceptible to faults. Another approach

for information redundancy is analytical redundancy that utilizes the correlation among system variables, which are either expressed by a mathematical model in an explicit form or implicitly hidden behind a large amount of data [3]. Regarding analytical redundancy, FDD methods can be generally categorized into model-based, and data-driven approaches. Since each of these approaches has advantages and disadvantages, hybrid approaches are introduced to take advantage of their strengths and tackle their weaknesses [11].

2.2 Model-based FDD Methods

Model-based FDD methods require an explicit model of the target system for the estimation of the system's variables. These methods work based on residuals determined by comparing the measured or expected values of the system's variables with their prediction [12–18]. Model-based FDD methods have their own advantages and disadvantages. One of the advantages of these techniques is that they need limited information to monitor the health status of the system, so their computational cost is low which makes them easily applicable for real-time applications. Another advantage of these approaches is their ability to detect and diagnose unknown faults. However, the disadvantages of model-based methods are that they need an explicit model for the target system and the model's accuracy remarkably affects these methods' performance [3, 19, 20]. Model-based FDD methods can be categorized into different categories including [21]:

- System Identification and Parameter Estimation Methods
- Parity Space Methods
- Observer-based (state estimation) Methods

2.2.1 System Identification and Parameter Estimation Methods for FDD

System identification and parameter estimation methods are applied to provide information redundancy regarding the physical parameters of a system by using its input and output

signals. If the estimated physical parameters deviate from their actual values, it indicates that a fault occurs in the system [22]. These approaches are usually applied for plant fault detection. Sensor and actuator FDD with these techniques may be difficult because the faults in sensors and actuators affect the system's input and output in the same as plant faults do [3]. There are some studies conducted on FDD using parameter estimation methods. In [23], a parameter estimation-based method is proposed to detect, isolate, and reconstruct faults in nonlinear satellite models. In [24], a recursive least squares parameter estimation approach is applied to detect and localize a fault that occurs in the electrical power system of an aircraft. In [25], a multi-parametric programming method is proposed to estimate the parameters of nonlinear process systems for fault detection.

2.2.2 Parity Space Methods for FDD

In the application of fault detection, parity space methods work based on comparing the output of an actual system and its model. In these methods, the model output is obtained by using an input-output transfer matrix without applying any observers [26]. Some research has been carried out in the field of FDD by applying parity space approaches. For instance, in [26], a parity space method is developed to detect actuator faults in linearized systems. This method is applied to an unmanned aerial vehicle and its effectiveness is validated by simulation and practical tests. In [27], the parity space method applied for fault detection in linear systems is generalized to nonlinear systems by using Takagi-Sugeno fuzzy models. In [28], A fault detection and quantification method is developed by the integration of a parity space technique and a recursive least squares algorithm. The performance of this method is evaluated by numerical simulation on a quadrotor unmanned aerial vehicle.

2.2.3 Observer-based Methods for FDD

Observer-based FDD methods rely on residuals, which are obtained by comparing the actual system states and inputs with their estimations provided by observers to detect faults in sensors and actuators. If the target component is a sensor, the residual is determined

by comparing the sensor measurement with its estimation [29]. If the target component is an actuator the residual is obtained by comparing the desired control actions generated by controllers with the estimation of the actuator's actions. For deterministic cases, Luenberger observers and ellipsoidal set-membership estimators [30], and for stochastic cases, Kalman filters, extended Kalman filters, and unscented Kalman filters can be used to estimate the system states [31]. It is essential for FDD techniques to be robust to disturbances and the model uncertainties which can be modeled as unknown inputs [3, 12]. Unknown input observers without a need to know the system inputs are able to estimate the states and inputs of the system [15, 32, 33]. The aforementioned tools are model-based estimators that can be used to detect faults in the system sensors and actuators.

Some research focuses on observer-based approaches for the FDD application in vehicles. In [34], a Luenberger observer is designed for FDD in lane-keeping control. The loop-shaping methodology is applied to retune the observer gain. The robustness required for fault detection is modeled as constraints on the frequency domain of residual signals. In [35], a higher-order sliding mode observer is proposed to estimate the vehicle motor speed. Sensor faults can be detected by comparing the measurement of the motor speed with its estimation. A reconfiguration section works based on switching from measured speed to its estimation in presence of the sensor faults. According to this reconfiguration section, a PI controller can provide suitable performance in the speed tracking control task in presence of faults such that it can act as a fault-tolerant controller. In [16], a fault detection and reconstruction approach is proposed for the longitudinal motion of autonomous vehicles. The longitudinal acceleration of the host vehicle is provided by an acceleration sensor and the relative states of its front object are obtained by a radar sensor. A sliding mode observer is applied to estimate the longitudinal acceleration. Based on the residuals obtained by comparing the sensor measurement and its estimation, the acceleration sensor faults can be detected. In [36], an observer-based FDD method is provided for some important vehicle sensors and actuators. Kalman filters and extended Kalman filters are applied to construct a bank of observers. Since each sensor or actuator fault affects a unique subset of residuals obtained by the bank of observers, it is possible to localize the source of faults. A sensor and actuator FDD approach for a small autonomous helicopter is presented in [32]. This approach works based on residuals generated by a

bank of observers. In [37], a bank of adaptive unknown input observers is constructed to detect and identify the actuators' faults in aircraft, and the effectiveness of this approach is evaluated by applying it to the linear model of the F-16 aircraft.

Several observer-based FDD approaches are developed for different types of dynamic systems. In [38], an observer-based method is provided to generate residuals for FDD in linear discrete time-varying systems. In this study, the performance of this method is improved by increasing its sensitivity to the fault and its robustness to disturbance. In [39,40], a model-based estimator is presented for state estimation and hence fault detection in time-varying nonlinear systems. An unscented Kalman filter is applied for FDD in nonlinear systems with unknown input in [41]. In [42], an unknown input observer is designed to detect and diagnose faults in the sensors and actuators of Lipschitz nonlinear systems.

2.3 Data-driven FDD Methods

Data-driven FDD methods, unlike model-based ones, do not need an explicit model. However, they require a significant amount of data to discover knowledge implicitly hidden behind them [43]. This knowledge can provide information redundancy required for fault detection and diagnosis. The main advantage of these approaches is that they do not require a complete model of the system in contrast to model-based methods. This is quite essential, especially for complex systems, because deriving an accurate explicit model of these systems may be difficult or impractical. The performance of data-driven methods is heavily dependent on the quality of training data. Hence, these methods are not suitable to be applied to detect unknown faults [3, 4].

Data-driven fault detection and diagnosis methods are generally categorized into quantitative and qualitative methods. The quantitative approaches include supervised and unsupervised learning methods [3]. The main difference between supervised and unsupervised learning methods is that in the supervised method, training data need to be labeled, but in unsupervised methods, the data are labeled by themselves.

2.3.1 Unsupervised Methods for FDD

Unsupervised learning techniques usually attempt to cluster data based on their similarity. In FDD applications, each cluster may be an indication of a special type of fault [3]. Several studies have been conducted on FDD by applying unsupervised learning methods. In [44], an autoencoder, one of the unsupervised learning methods, is applied to detect faults in unmanned aerial vehicles. The autoencoder is initially trained by safe flight data and then based on the magnitude of reconstruction loss produced by the well-trained model, the healthy and faulty states of the vehicle can be determined. The low reconstruction loss is an indication of healthy states and the large one is a sign of fault. K means and fuzzy C means algorithms that are well-known clustering methods are applied for FDD in [45]. In [46], an unsupervised learning method is proposed for online FDD in autonomous robots. The effectiveness of the method is evaluated by conducting experimental tests on the electrical power system of a spacecraft, unmanned aerial vehicles, a vacuum-cleaning robot, and a flight simulator.

2.3.2 Supervised Methods for FDD

Supervised learning methods generally work based on training a mapping function from input to output and can be further classified into classification and regression techniques. Neural networks, principal component analysis (PCA), partial least squares (PLS), and fuzzy logic are supervised approaches applied for FDD in the literature. Neural networks are well-known machine learning methods for FDD applications since they are powerful tools to approximate heavily nonlinear functions and have great adaptive learning capabilities. In FDD tasks, the input of neural networks is data history and their output is normally the healthy or faulty system status [3]. Unlike neural networks that implicitly describe knowledge through a network of connections, fuzzy logic can explicitly express human knowledge [47]. Although supervised techniques show impressive performance in FDD applications, they are highly dependent on balanced datasets containing a substantial number of healthy and faulty data. In reality, collecting sufficient amounts of faulty data to create balanced datasets can be extremely difficult [48].

In the literature, there are several studies that focus on classification techniques for FDD applications. In [49], a general FDD method is proposed for autonomous vehicles. This method works based on learning faulty patterns by using deep learning algorithms, and its performance is examined by applying it to a multi-wheeled vehicle. It can appropriately identify faults that occur in the vehicle. In [50], a data-driven method is proposed for sensor FDD in unmanned aerial vehicles. In this method, an adaptive neuro fuzzy inference system (ANFIS)-based algorithm uses the residual determined by the Kalman filter for FDD. This model-free method can be considered one of the robust FDD methods and its adaptive rules can be extracted by updating the training dataset. A data-driven fault detection and identification method for multirotor aircraft is expressed in [51]. This method uses a combination of a statistical time series model and a neural network to detect and classify faults. In [52], another data-driven method is presented for sensor FDD in aircraft engines. In this method, training data are classified with different labels based on various engine health conditions, and feature extraction and pattern classification are applied for FDD.

The regression techniques can be considered data-driven estimators that provide the analytical redundancy required for fault detection. The data-driven estimators usually act as black boxes and by satisfying the observability conditions, they can estimate any system variables whether these variables are the inputs or states of the system. Hence, these estimators can be applied to estimate both sensor measurements and actuator commands. Kernel regression, one of the most well-known regression techniques, is used for FDD applications [53]. In [54], a kernel-based method as a nonlinear regression technique is applied to predict the output of nonlinear systems. By comparing sensor measurements to the predicted values, the residual required for FDD is generated. In [55], a fault detection approach that works based on auto-associative kernel regression is introduced, and some modifications are applied to it to increase its robustness.

2.4 Hybrid FDD Methods

As previously mentioned, each FDD method has its own advantages and disadvantages. To tackle the disadvantages and to gain the advantages of each method, hybrid FDD

approaches are developed. Hybrid approaches made by the integration of two or more FDD methods can outperform each of them when used individually.

Some studies focus on hybrid approaches for FDD in vehicles. A hybrid FDD method for autonomous vehicles is proposed in [56]. In this method, a support vector machine is applied to determine safe and unsafe domains, and a Kalman filter is adopted to predict the position of vehicles. Finally, the Jarque-Bera test is used to detect faults by checking the obtained residuals, and a fuzzy system is designed to diagnose the types of faults. Another hybrid FDD method is developed based on a convolutional neural network supplemented by the Kalman filter for automated vehicles [57]. The convolutional neural network is applied to generate images from time-series data and classify them into healthy and faulty categories. The results show that the hybrid approach can outperform a convolutional neural network and a Kalman filter when used individually. A hybrid fault detection and isolation algorithm is proposed for unmanned aerial vehicles [58]. This hybrid method is developed at low and high levels. The low-level method is able to appropriately detect faults and the high-level one can isolate the faults by the reconstruction of the low-level method. In [59], another hybrid fault detection and isolation method is introduced for a network of unmanned vehicles. In this method, a set of residuals is determined by a bank of residual-generator modules and then a discrete-event system fault detector can detect and isolate faults by using the residuals and their sequential features.

In addition to vehicles, hybrid FDD approaches are developed and applied to a variety of systems. For instance, in [60, 61], a hybrid approach is proposed to detect and isolate faults in power transmission systems. The proposed approach combines several techniques, including principal component analysis, multilevel wavelet transform, support vector machines, and adaptive structure neural networks, to enhance its robustness against noise and compatibility with regard to learning new environments. A hybrid FDD method composed of two data-driven techniques is presented in [53]. The first one is an auto-associative kernel regression technique that is applied to generate residuals. The second approach is a dynamic independent component analysis that can detect faults by using the obtained residuals. The efficiency of this method is verified by some experimental tests. To deal with unclassified data, hybrid FDD approaches are developed in [62, 63] for robotics systems. These methods are made of the integration of unsupervised and supervised meth-

ods. The unsupervised approach is applied to classify data into healthy and faulty, and then the supervised method is used to detect faults. The results show the superiority of the hybrid method over the original unsupervised one. In [64], a hybrid FDD method for high performance operational systems is presented. This method works based on the fuzzy c-means clustering technique and artificial immune system. The simulation results verify the strength of this method for real-time applications. A hybrid approach proposed in [65] works based on integrating data-driven and model-based approaches to select a set of residuals for fault detection in systems where model uncertainties and measurement noises are considerable.

Chapter 3

A Hybrid Model-Data Vehicle Sensor & Actuator Fault Detection & Diagnosis System

3.1 Introduction

The main goal of this chapter is to develop a reliable hybrid model/data fault detection and diagnosis system for any vehicle sensor and actuator. The contributions of this chapter are as follows:

- Developing a hybrid framework simultaneously using vehicle models and data to improve the capability of detecting and quantifying faults.
- Proposing a self-updating rule for the dataset in confronting new environments. The developed fault detection approach can use the updated dataset to learn these environments.

This chapter is organized as follows: Section 3.2 presents the developed hybrid fault detection and diagnosis methodology. A test vehicle and its experimental setup are presented in Section 3.3. In Sections 3.4 and 3.5, the proposed approach is applied to the

vehicle's lateral acceleration sensor and traction motor, respectively, and its performance is evaluated by experimental tests. Finally, the chapter is summarized in Section 3.6.

3.2 Hybrid Model/Data Fault Detection & Diagnosis System

The general scheme of the proposed fault detection and diagnosis system is provided in Fig. 3.1. The authentication module is a decision center that can determine which of the model-based or data-driven estimators provides more reliable estimation. In the ideal case, the dataset of a data-driven estimator should cover the whole feasible space; however, in this research, a self-updating dataset is proposed that relaxes this requirement. Using the self-updating dataset, new data will be added to the dataset when the vehicle faces new environments. The fault-detector module is able to detect the fault in the vehicle by using the residuals obtained by comparing the measurements (provided by sensors) or desired control actions (obtained by controllers or a driver) with their estimations (determined by the hybrid estimator). If the residuals exceed the threshold and are persistently placed outside the prediction interval, the target sensor or actuator becomes faulty. After detecting a fault, if the target component is a sensor, its healthy value is reconstructed by using the estimation provided by the hybrid estimator, and if the target component is an actuator, its actual action is obtained by the hybrid estimation, which can be utilized to find its level of failure. All modules and blocks shown in Fig. 3.1 are explained in the following.

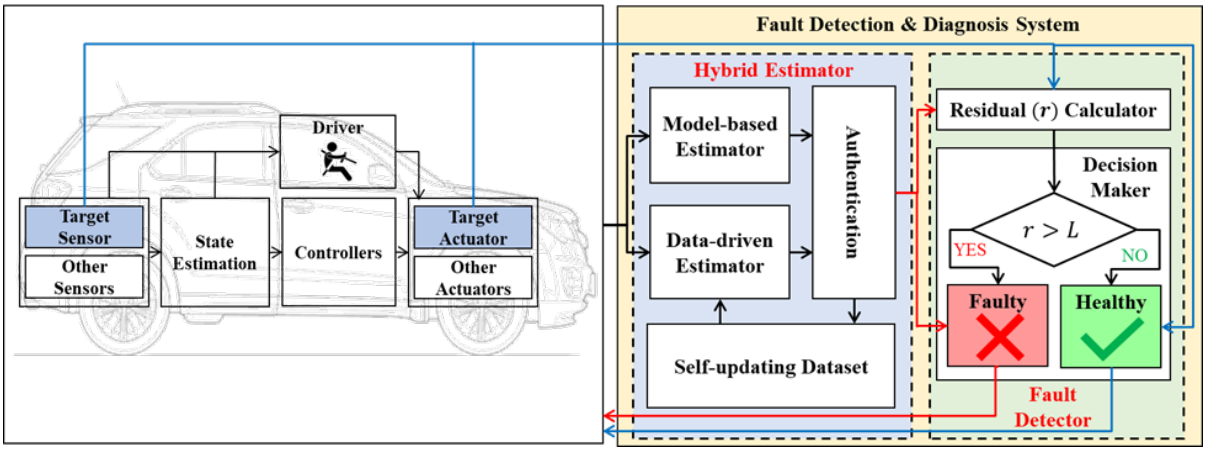

Figure 3.1: The general scheme of the hybrid fault detection and diagnosis system.

3.2.1 Data-driven Estimator

There are various data-driven techniques that can be used for estimation applications such as deep learning neural networks, multi-layer perceptron, and adaptive neuro-fuzzy inference systems [3]. Most of these methods not only require a significant amount of data to obtain suitable performance but also have numerous parameters that must be trained [46]. As new data is needed for the training of unseen conditions that often happen in vehicles, the dataset must be gradually updated. Estimation based on the updated dataset brings about a need of using instant-based learning methods as the data-driven estimator for this research.

Kernel regression, one of the most well-known statistical techniques, is widely used as a function approximator [66]. This method is applied for estimation by finding a nonlinear relationship between inputs and output and provides a smooth approximation based on an appropriate distance measure. Kernel regression as an instant-based learning method does not need pre-training, which makes it more applicable for real-time fault detection. By using such kernel regressions, the dataset can be updated in real-time and the estimation can be conducted based on the updated dataset.

3.2.1.1 Nadaraya-Watson Kernel Regression

Nadaraya-Watson Kernel Regression (NWKR), one of the well-known kernel regression approaches, works based on weighted averaging [66]. The kernel function, $K(\,,\,)$, applied in NWKR should satisfy the following conditions:

$$\begin{aligned} &\text{Positive Semi-definiteness: } K(\mathbf{x}, \mathbf{x}_j) \geq 0 \\ &\text{Symmetry: } \int \mathbf{x} K(\mathbf{x}, 0)\, d\mathbf{x} = 0 \end{aligned} \tag{3.1}$$

where $\mathbf{x}$ is a test point that is required to be predicted, and $\mathbf{x}_j$ is one of the points in the dataset. The Gaussian kernel function:

$$K_\sigma(\mathbf{x}, \mathbf{x}_j) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(\mathbf{x} - \mathbf{x}_j)^2}{2\sigma^2}\right) \tag{3.2}$$

is a well-known and commonly used kernel function in the literature. In this equation, σ is the standard deviation of the Gaussian function, which is also called the bandwidth of the kernel function and determines the width of this function. The feature vector $\mathbf{x}_j$ and test point $\mathbf{x}$ with m individual variables are defined as

$$\mathbf{x}_j = \left(x_j^1, \ldots, x_j^m\right)^\top, \qquad \mathbf{x} = \left(x^1, \ldots, x^m\right)^\top. \tag{3.3}$$

Additionally, their respective outputs are denoted by y and y_j. NWKR approach estimates the output of the test point y using

$$\widehat{y}(\mathbf{x}) = \frac{\sum_{j=1}^{n} y_j \left(\prod_{i=1}^{m} K_\sigma\left(x^i, x_j^i\right)\right)}{\sum_{j=1}^{n} \left(\prod_{i=1}^{m} K_\sigma\left(x^i, x_j^i\right)\right)} = \sum_{j=1}^{n} w_j y_j = \mathbf{w}^\top \mathbf{y} \tag{3.4}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^\top$ and $\mathbf{w} = (w_1, \ldots, w_n)^\top$.

In the next subsections, the following questions are addressed:

- How is the performance of NWKR evaluated? (Subsection 3.2.1.2)
- How are the features that are essential in the estimation of a target variable selected or in other words, how are the inputs of NWKR determined? (Subsection 3.2.1.3)

- How σ, the only parameter of NWKR, is tuned? (Subsection 3.2.1.4)
- What are the effects of distant points in the dataset on the estimation? (Subsection 3.2.1.5)
- How is the dataset used in NWKR managed? (Subsection 3.2.1.6)

3.2.1.2 Evaluation of NWKR's Performance

To evaluate the performance of NWKR during a test maneuver, Root Mean Squared Error (RMSE) is applied. RMSE shows the spread of prediction errors called residuals, and it is determined by

$$\text{RMSE} = \frac{\sum_{i=1}^{n'} \left(y(i) - \widehat{y}(i)\right)^2}{n'} \tag{3.5}$$

where, $y(i)$ and $\widehat{y}(i)$ are respectively the i^{th} observed and predicted point in the test maneuver, and n' is the number of points in the maneuver. Based on the definition of RMSE, a lower RMSE value indicates a more precise prediction.

3.2.1.3 Feature Selection by Sensitivity Analysis (Pearson Correlation Coefficient)

Sensitivity analysis is applied for feature selection, which is an important part of data-driven methods. In a system, just some of the features (variables) have a strong correlation with a target feature. In fact, these features play the main role in the estimation of the target variable. If all features are applied for the estimation, it may cause an overfitting problem. Overfitting makes the trained model too fitted to the training dataset which has an adverse effect on the fitting of the trained model to test data.

By considering the vehicle dynamics equations, it is possible to find the features that play a meaningful role in the estimation of the target variable in vehicles. This approach may not be applicable to all vehicle variables due to the system's complexity, and unmodeled dynamics. Therefore, a systematic method is required to select the features that are essential in the estimation of the target variable.

Pearson Correlation Coefficient (PCC) is a statistical method that is applied to determine the strength and direction of the linear relationship between two variables (u_1 and u_2) by calculating the linear correlation between them. The value of PCC is always between -1 and 1. If the absolute value of PCC between two variables is equal to 1, it shows that these two variables are completely linearly correlated. By decreasing the strength of linear correlation between variables, the absolute value of PCC decreases. When PCC is equal to zero, it indicates that the variables are linearly independent of each other [67]. For two feature vectors $\mathbf{u}_1 = (u_1^1, \ldots, u_1^n)$ and $\mathbf{u}_2 = (u_2^1, \ldots, u_2^n)$, PCC is calculated using

$$\text{PCC} = \frac{\sum_{i=1}^{n} \left(u_1^i u_2^i\right) - n\bar{u}_1\bar{u}_2}{\sqrt{\sum_{i=1}^{n} \left(u_1^i\right)^2 - n\bar{u}_1^2}\sqrt{\sum_{i=1}^{n} \left(u_2^i\right)^2 - n\bar{u}_2^2}} \tag{3.6}$$

where $\bar{u}_1$ and $\bar{u}_2$ are the means of $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively.

3.2.1.4 Bandwidth Selection

In NWKR, it is essential to obtain a rule to tune the bandwidth of the Gaussian kernel in Eq. 3.2. According to the literature, the bandwidth is selected based on the variance of output [66]. In this study, since the variance of the output is varying, it may be difficult to formulate the bandwidth as a function of the output variance. In this case, to tune the bandwidth, the RMSE of estimations with different values of bandwidth are calculated. The bandwidth that corresponds to the minimum RMSE is the optimal bandwidth [68].

3.2.1.5 Estimation Based on Neighbors

In this section, it is shown that the nearest points in the dataset to the test point play the main role in its estimation, so the estimation can be conducted only based on these points. To show it, the Gaussian function, Eq. (3.2), used in NWKR is considered. In Eq. (3.2), if the distance between x_j^i (the i^{th} feature of the j^{th} point in the dataset) and x^i (the i^{th} feature of the test point) becomes greater than 3σ, $K_\sigma(x^i, x_j^i) < 0.005$, and it is negligible. According to Eq. (3.4), if $K_\sigma(x^i, x_j^i)$ has a very small value, the j^{th} point can be omitted to participate in the estimation of the test point's output. It is inferred that the

nearest neighbors to each test point play the main role in its output estimation. Therefore, to reduce the computational time, it is possible to use only points in the dataset that are placed in the neighborhood of each test point to estimate its output. In this research, an ε-neighborhood is defined as a multi-dimensional cube that is considered around each test point and its edge is 2ε. By considering the ε-neighborhood, for each test point, only a portion of the dataset participates in the estimation. Hence, the computational cost of the estimation decreases significantly.

Note that without loss of generality, since the scales of features are different from each other, each feature is normalized by dividing it by the maximum of its absolute value. Therefore, all data lie between -1 and 1.

3.2.1.6 Dataset Management

To ensure that a dataset does not include repeated points or closely clustered points, it is important to analyze and manage it during data collection. A dataset is ideal when it uniformly covers all the feasible space, but in reality, it is possible that some points in the dataset are too close to each other. Hence, a systematic method is required to manage and remove excess and unnecessary points. To achieve this goal, a data management approach is proposed in this study. This approach, presented in Algorithm 1, works based on the number of neighbors in the ε_1-neighborhood of each point in the dataset. If this number is greater than or equal to a threshold denoted by k_1, the point should be removed from the dataset.

Algorithm 1 Data Management.

Input: Dataset: $\mathbf{x}$,
ε in ε-neighborhood: ε_1,
Threshold of the number of neighbors: k_1.
$\mathbf{z} \leftarrow$ Normalize $\mathbf{x}$
$n \leftarrow$ Size of $\mathbf{x}$
for $j = 1, ..., n$ **do**
  if the number of neighbors of $\mathbf{z}_j$ in its ε_1-neighborhood $> k_1$ **then**
    remove $\mathbf{x}_j$ from $\mathbf{x}$
  end if
end for
Output: Reduced dataset: $\mathbf{x}$.

This algorithm is applied to the dataset with different values of k_1 to provide different reduced datasets. Each new dataset is applied for the estimation, and its RMSE and average computational time are determined. Given the obtained RMSE and average computational time, the optimal dataset is determined. By using the optimal dataset, the quality of estimation is preserved while its computational time decreases.

3.2.2 Model-based Estimator

There are several studies conducted to estimate vehicle states and inputs in the literature. Some studies regarding vehicle state estimation are summarized in [69, 70], and for each one, the estimation methodology, vehicle model, required measurements, and estimated states are provided. This research presents the model-based estimators used for vehicle state estimation, including different versions of the Kalman filter, Luenberger observer, nonlinear observer, and sliding mode observer. These techniques can be used as model-based estimators to detect faults in sensors. Unknown input observers can estimate the states and inputs of a system without a need to know its inputs. Some studies conducted on automotive systems apply these observers to estimate the vehicle engine torque [71],

driver torque [72, 73], and vehicle steering angle [74]. Hence, for fault detection of vehicle actuators, these approaches can be applied as model-based estimators in the proposed fault detection system.

3.2.3 Authentication & Self-updating Dataset

The hybrid estimator works based on the integration of the model-based and data-driven estimators. Based on the availability of data in the self-updating dataset around the operating points (test points), the authentication module is responsible for determining which estimator should be used. As previously mentioned, model-based estimators are able to work in unknown conditions and environments, unlike data-driven ones. Therefore, in these conditions, the authentication module chooses the estimation provided by the model-based estimator. If the dataset has enough data around the test points, the data-driven estimator can provide accurate estimation, so the authentication module chooses the estimation obtained by the data-driven estimator. The hybrid estimation, $\widehat{y}_H$, provided by the authentication module can be modeled as:

$$\widehat{y}_H = f\left(\widehat{y}_M, \widehat{y}_D, n_\varepsilon\right) \;=\; \begin{cases} \widehat{y}_M & n_\varepsilon < k_2 \\ \widehat{y}_D & n_\varepsilon \geq k_2 \end{cases} \tag{3.7}$$

in which $\widehat{y}_M$ and $\widehat{y}_D$ are the estimations of the model-based and data-driven estimators, respectively. Parameter n_ε is the number of neighbors in the ε-neighborhood of each test point, and k_2 is a threshold for the number of neighbors.

The self-updating module adds new data to the dataset when the vehicle faces unknown conditions or the dataset is not rich enough around test points. Therefore, the data-driven estimator can use the new data if the vehicle encounters these conditions or similar ones again. In the proposed fault detection system, when $n_\varepsilon < k_2$, and the fault detector confirms that the target component is healthy, the authentication module issues a command to update the self-updating dataset by adding test points to it.

3.2.4 Fault Detector

The fault detector module shown in Fig. 3.1 notifies the occurrence of faults in the target sensor or actuator. This module contains the residual-calculator and decision-maker modules. The residual-calculator module obtains the residual, r, by comparing the measurements (provided by sensors) or desired control actions (obtained by controllers or a driver) with their estimations (determined by the hybrid estimator) as follows:

$$r = y - \widehat{y}_{H} \tag{3.8}$$

where y is the true value of the target variable (sensor measurement or desired control action). If the residual lies persistently outside of the prediction interval $(-L_{H}(i), +L_{H}(i))$, the decision-maker module notifies that there is a fault in the target component. Otherwise, the target component is healthy. The hybrid threshold L_{H} is equivalent to the model-based threshold L_{M} when the model-based estimator is utilized and it is the same as the data-driven threshold L_{D} when the data-driven estimator is employed. The threshold L_{D} can be derived using NWKR estimation is Eq. (3.9) as follows,

$$L_{D} = t_{\alpha/2,df} \sqrt{\widehat{\sigma}_{D}^{2} \left(1 + \mathbf{w}^{\top}\mathbf{w}\right)} \tag{3.9}$$

where $t_{\alpha/2,df}$ is the $\frac{\alpha}{2}$-th quantile of the t-distribution with df degrees of freedom and α as the significance level [75]. The quantity $\widehat{\sigma}_{D}^{2}$ is the *Mean Square Error (MSE)* defined as:

$$\widehat{\sigma}_{D}^{2} = \frac{1}{n_{\varepsilon} - 1} \sum_{j=1}^{n_{\varepsilon}} \left(y_{j} - \widehat{y}_{j}\right)^{2} \tag{3.10}$$

where $\widehat{y}_{j}$ is the estimation of y_{j}. For the model-based estimator, the threshold L_{M} for the prediction of the current test point is obtained by,

$$L_{M} = t_{\alpha/2,df} \sqrt{\widehat{\sigma}_{M}^{2} \left(1 + \frac{1}{n'}\right)} \tag{3.11}$$

in which $\widehat{\sigma}_{M}^{2}$ is the standard deviation of predictions obtained for the previous test points placed in the determined window of time, and the number of these previous test points is denoted by n'. L_{M0} is considered a minimum value for L_{M}, and this value is determined

based on the target component's specifications or nominal accuracy. If L_M obtained by Eq. (3.11) is less than L_{M0}, then $L_M = L_{M0}$.

Once a fault has been detected, the hybrid estimator can be used to reconstruct the healthy value of the faulty sensor. This reconstructed data can then be utilized by the vehicle's control systems instead of the faulty sensor measurements. In the event that the faulty component is an actuator, the hybrid estimator can be utilized to determine the level of failure. Based on this information, it is possible to scale the faulty actuator's output to achieve the desired results.

3.3 Test Vehicle & Experimental Setup

In this research, To experimentally evaluate the performance of the developed approaches, an electric all-wheel-drive test vehicle shown in Fig. 3.2 is utilized. The specifications of this vehicle are listed in Table 3.1. This vehicle has an independent motor for each four wheels as its powertrain system and a robotic steering system to perform repeatable maneuvers. The vehicle collects the required data for the health monitoring system using a 6-axis IMU sensor, a steering wheel sensor, and wheel speed and torque sensors.

The desired control actions of the steering actuator and the traction motors are exported from the MATLAB/Simulink environment and are sent directly to the vehicle actuators (drive motors, brake, etc.) via a CAN-bus network. After applying the desired control actions, all the sensor measurements are read through the CAN-bus in the real time MATLAB/Simulink environment. The experimental setup diagram of the test vehicle is illustrated in 3.3.

Figure 3.2: An electric all-wheel-drive test vehicle.

Table 3.1: The specifications of the test vehicle.

Parameter	**value**	**Description**
M	2271 kg	Vehicle mass
I_z	4600 kgm^2	Vehicle moment of inertia about normal direction
a, b	1.42,1.43 m	Distance from front/rear axle to the CG
R_w	0.347 m	Tire effective radius
C_α	83700 N	Tire cornering stiffness
I_w	1.7 kgm^2	Wheel moment of inertia about its axis
SR	18 : 1	Ratio between steering wheel angle & front wheel angle

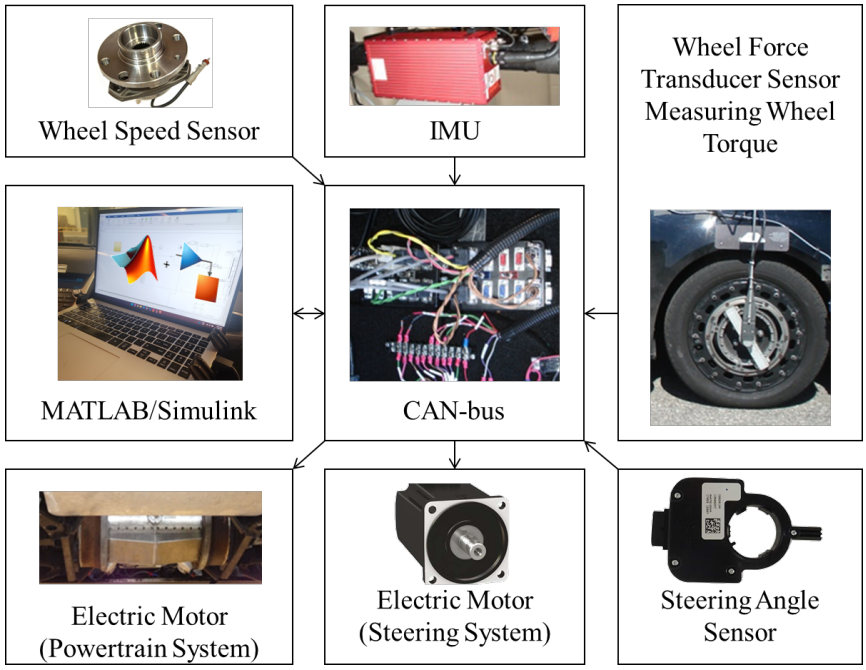

Figure 3.3: The experimental setup diagram of the test vehicle.

3.4 Hybrid Fault Detection and Diagnosis System Applied to Lateral Acceleration Sensor

In this section, the hybrid fault detection and diagnosis system is applied to detect faults in a lateral acceleration sensor and reconstruct its healthy value. In the following subsections, all modules shown in Fig. 3.1 and explained in Section 3.2 are developed for this objective. The performance of this system is experimentally evaluated with the test vehicle.

3.4.1 Data-driven Estimator for Lateral Acceleration

NWKR is applied as the data-driven estimator to estimate the vehicle's lateral acceleration. The data-driven estimator requires a dataset to operate and needs test data to evaluate

its performance. The dataset and test data are provided by conducting experimental tests with the test vehicle. Without loss of generality, some acceleration-in-turn maneuvers are used to create the dataset. Two experimental tests are also considered as test data. Like the maneuvers in the dataset, one of these two tests is an acceleration-in-turn maneuver, and another one is a double-lane-change (DLC) maneuver that is new for the data-driven estimator based on its primary dataset. The specifications of all these maneuvers are provided in Table 3.2. In this table, the subscripts $ij \in \{fL, fR, rL, rR\}$ represent the front-left, front-right, rear-left, and rear-right wheels, respectively.

Table 3.2: The specifications of the maneuvers used for collecting the dataset and test data

	Torque of each wheel $T_{ij} = A \times$ Ramp Function $T_{\max} = 500\ N.m$	Steering wheel angle $\delta = B \times$ Step function (deg)
Dataset	$A \in \{50, 70, 80\}$	$B \in \{140, 160, 200, 220\}$
Test Data #1	$A = 60$	$B = 180$
Test Data #2	$A = 60$	δ for DLC maneuver

3.4.1.1 Feature Selection by PCC

All variables that play a role in the vehicle planar motion are listed in Table 3.3. These variables are either measurements provided by sensors or control actions obtained by actuators. Since the reliable estimation of longitudinal velocity can be available by the GPS sensor, it is also considered in the list of variables.

Table 3.3: List of the vehicle variables in the proposed string of data

Variable	**Description**
δ	Steering Wheel Angle
δ_w	Front Wheel Steering Angle
T	Total torque applied to the wheels
a_x	Longitudinal Acceleration
a_y	Lateral Acceleration
r,$\dot{r}$	Yaw Rate Acceleration
ω_{ij}	Wheel Angular Velocity
$\dot{\omega}_{ij}$	Wheel Angular Acceleration

PCC is applied to select the essential features for the estimation of the vehicle lateral acceleration. Using the collected data from the maneuvers in Table 3.2, PCCs between all the variables presented in Table 3.3 and the vehicle lateral acceleration are calculated and shown in Fig. 3.4.

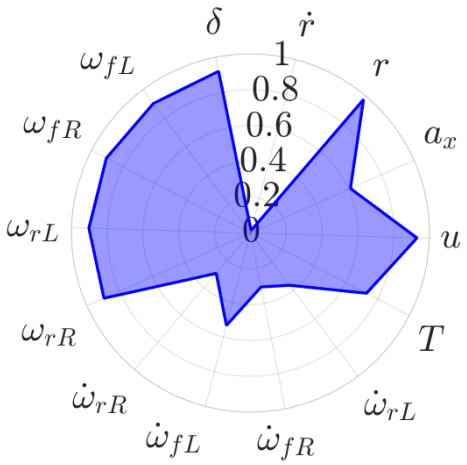

Figure 3.4: PCCs between the vehicle lateral acceleration and other variables.

This figure shows that a_y has strong linear correlations with the yaw rate, steering wheel angle, vehicle longitudinal velocity, and angular velocity of all wheels.

To show the importance of feature selection, two NWKR models shown in Fig. 3.5 are considered to estimate the vehicle lateral acceleration. In the first NWKR, the features selected by using the PCC test are considered as the NWKR's inputs, and in the second one, all features presented in Fig. 3.4 are assigned as its inputs. Given these two NWKR models, the estimations with different values of bandwidth (σ) are provided to reconstruct the dataset and estimate the first test data (Table 3.2). The RMESs of these estimations are calculated and shown in Fig. 3.6. This figure shows that in the small values of the bandwidth, the 1st NWKR model provides more accurate estimations than those obtained by the 2nd NWKR model for the first test data. Therefore, the missed variables not only are not essential features in the estimation of a_y but also decrease the estimation accuracy. Fig. 3.6 also illustrates that the excessive reduction of σ causes decreasing the RMSEs of the dataset reconstruction while increasing the RMSEs of the test data estimation. This condition is known as overfitting which makes the dataset reconstruction too fitted to the dataset and adversely affects the prediction of the test set. Fig. 3.7 illustrates the sensor measurement and the estimations of a_y provided by the two NWKR models with $\sigma = 0.05$ for the first test data.

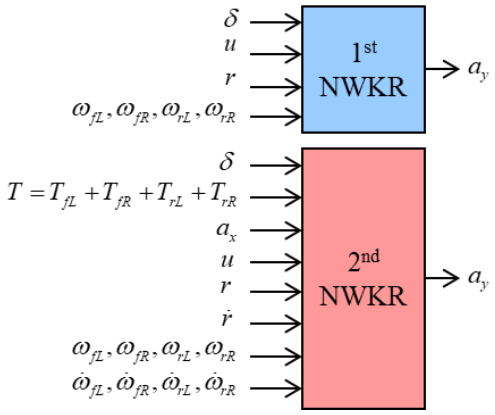

Figure 3.5: Two NWKR models are applied for the vehicle lateral acceleration estimation.

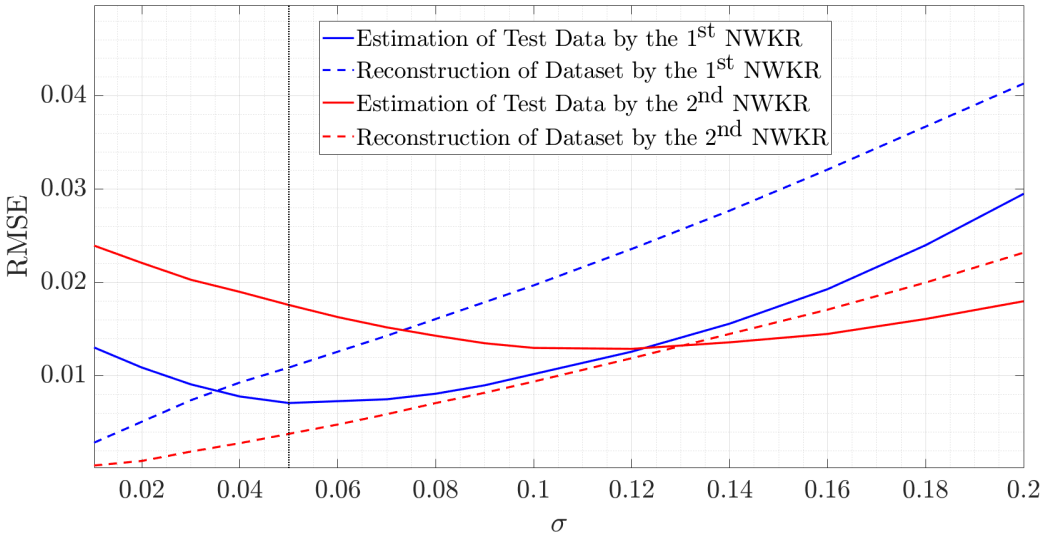

Figure 3.6: RMSE of the estimations provided by the two NWKR models with different values of bandwidth.

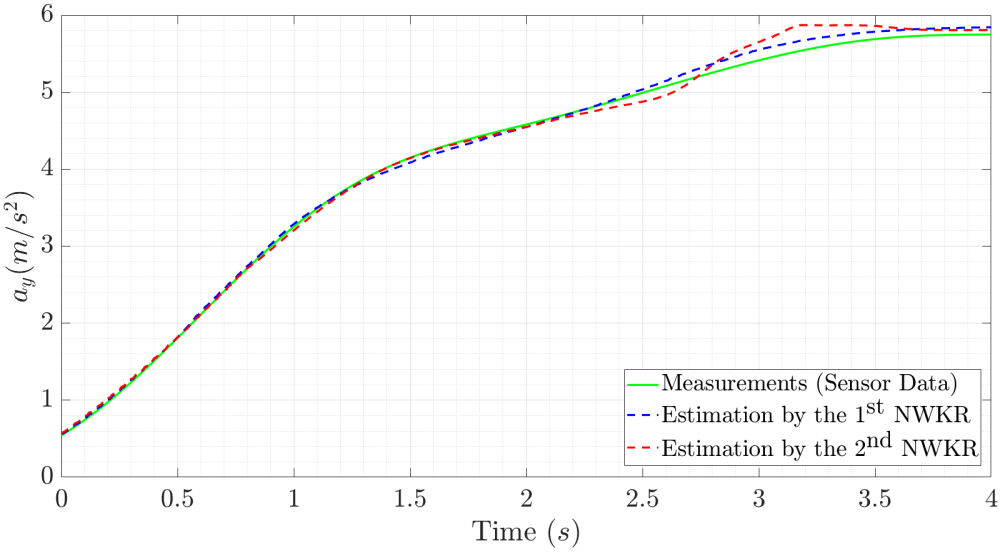

Figure 3.7: The sensor measurement and the estimation of a_y provided by the 1st and 2nd NWKR models for $\sigma = 0.05$.

3.4.1.2 Bandwidth Selection

As previously mentioned, to tune the bandwidth, at first, NWKR with different values of the bandwidth is applied for the estimation of test data, and then, the RMSEs of these estimations are calculated. Finally, based on the results, the bandwidth that corresponds to the minimum RMSE is considered the desired bandwidth. Fig. 3.6 (the solid blue line) shows that when $\sigma = 0.05$, the minimum RMSE of the estimations provided by the chosen NWKR is achieved.

3.4.1.3 Estimation Based on Neighbors

In Section 3.2.1.5, it is mentioned that considering a proper ε-neighborhood for the test data not only does not decrease the quality of estimations but also causes a significant reduction in the computational time. This point can also be evaluated by using the first test data. To find an appropriate ε-neighborhood, for different values of ε, the RMSEs of the estimations and the average computational time required for the estimation of each test point are determined. These values are presented in Fig. 3.8. This figure illustrates that for the NWKR with $\sigma = 0.05$, the RMSE remains constant by reducing ε to 3σ. This point has conformity with the negligibility of $K_\sigma(x^i, x^i_j)$ (Eq. (3.2)) when $|x^i - x^i_j| > 3\sigma$. Fig. 3.8 also shows that the average computational time required for the estimation of each test point significantly decreases by shrinking the ε-neighborhood. By considering $\varepsilon = 0.15$, the computational time drops to almost $1/3$ of the computational time of the estimation in which the ε-neighborhood is not considered.

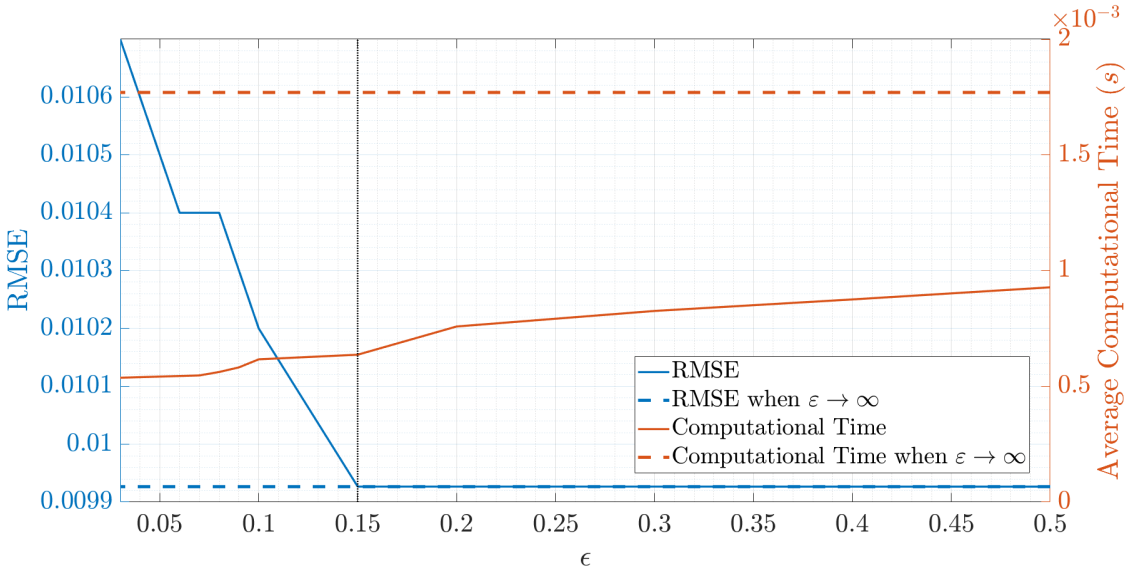

Figure 3.8: The effect of considering the ε-neighborhood on the accuracy of estimation and its computational time.

3.4.1.4 Dataset Management

To remove the redundant data from the dataset, the data management technique proposed in Algorithm 1 is used. This algorithm is applied with different values of k_1 and $\varepsilon_1 = 0.02$ to provide different reduced datasets, and these extracted datasets are applied for estimation. The RMSE and average computational time of these estimations are provided in Fig 3.9. This figure shows that the desired k_1 is 40 since by using its corresponding dataset, the average computational time remarkably decreases, while the RMSE remains constant. It indicates that the accuracy of estimation is preserved although the size of the database denoted by n on the x-axis of Fig 3.9 decreases.

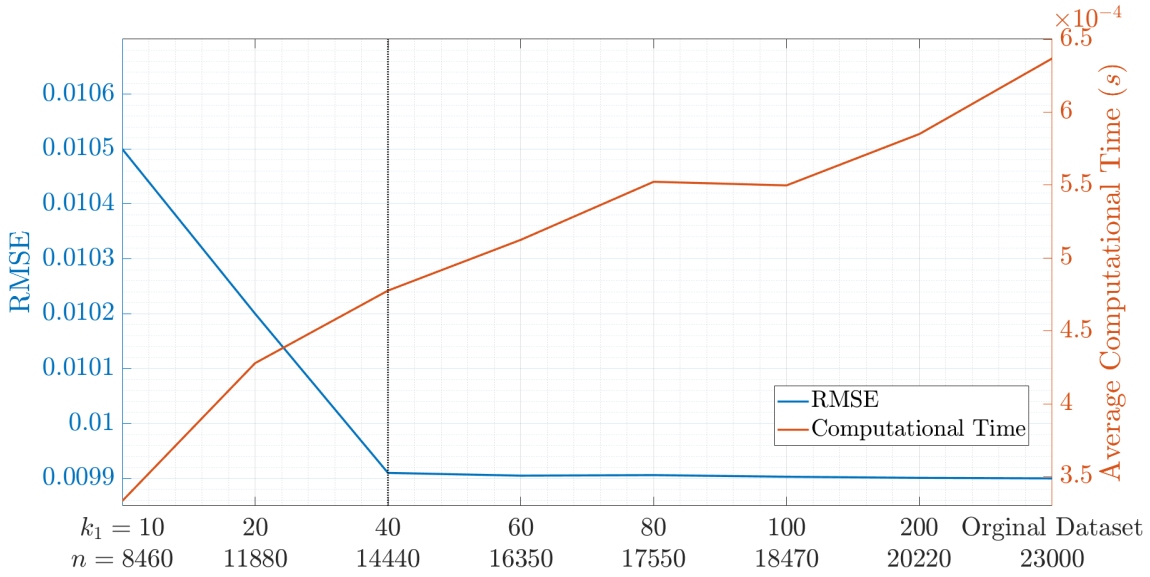

Figure 3.9: The RMSE and average computational time of estimations provided by the reduced datasets.

3.4.2 Model-based Estimator for Lateral Acceleration

To provide the model-based observer, the vehicle is modeled by using a planar single-track model with the following equations of motion:

$$\dot{\mathbf{x}}_s(t) = \begin{bmatrix} -\dfrac{C_{\alpha f}+C_{\alpha r}}{Mu(t)} & -\dfrac{aC_{\alpha f}-bC_{\alpha r}}{Mu(t)} - u(t) \\ -\dfrac{aC_{\alpha f}-bC_{\alpha r}}{I_z u(t)} & -\dfrac{a^2C_{\alpha f}+b^2C_{\alpha r}}{I_z u(t)} \end{bmatrix} \mathbf{x}_s(t) + \begin{bmatrix} \dfrac{C_{\alpha f}}{M} \\ a\dfrac{C_{\alpha f}}{I_z} \end{bmatrix} \delta_w(t) \tag{3.12}$$

$$\dot{\mathbf{x}}_s(t) = \begin{bmatrix} \dot{v}(t) \\ \dot{r}(t) \end{bmatrix} \quad \mathbf{x}_s(t) = \begin{bmatrix} v(t) \\ r(t) \end{bmatrix}$$

in which v is the vehicle's lateral velocity, and $C_{\alpha f}$ and $C_{\alpha f}$ are the front and rear tire cornering stiffness, respectively. The descriptions of other variables and parameters are provided in Tables 3.1 and 3.3. Note that the time variable is denoted by $t \in \mathbb{R}$ in the

continuous form and $k \in \mathbb{Z}$ in the discrete form. For the tires, a piecewise linear model is considered, so based on that:

$$C_{\alpha i} = \begin{cases} C_\alpha & \alpha_i < \frac{4\pi}{180} \\ \frac{4\pi}{180} \frac{C_\alpha}{|\alpha_i|} & \alpha_i \geq \frac{4\pi}{180} \end{cases} \tag{3.13}$$

where the subscripts $i \in f, r$ represent the front and rear wheels, and their slide-slip angles are respectively α_f and α_r that can be determined by:

$$\begin{aligned} \alpha_f &= \tan^{-1}\left(\delta_w(t) - \frac{v(t) + ar(t)}{u(t)}\right) \\ \alpha_r &= \tan^{-1}\left(-\frac{v(t) - br(t)}{u(t)}\right) \end{aligned} \tag{3.14}$$

The forward Euler approach with the time step, Δt, is utilized to discretize the vehicle model. The discretized vehicle model with $\Delta t = 0.01$ is presented in:

$$\begin{aligned} \mathbf{x}_s(k+1) &= \mathbf{A}_d \mathbf{x}_s(k) + \mathbf{b}_d \delta(k) + \Omega_p \\ y(k) = r(k) &= \mathbf{c}_d^\top \mathbf{x}_s(k) + \Omega_m \\ &= (0,\ 1)\, \mathbf{x}_s(k) + \Omega_m \end{aligned} \tag{3.15}$$

in which, yaw rate is considered the available measurement, and Ω_p and Ω_m are process and measurement noises, respectively.

The Kalman filter expressed in [76] is

$$\boxed{\begin{array}{l}\widehat{\mathbf{x}}_s^{-}(k+1)=\mathbf{A}_d(k)\,\widehat{\mathbf{x}}_s(k)+\mathbf{b}_d(k)\,\delta(k)\\ \mathbf{P}^{-}(k+1)=\mathbf{A}_d(k)\,\mathbf{P}(k)\,\mathbf{A}_d{}^{\top}(k)+\mathbf{Q}\end{array}}$$

$$\begin{array}{cc}\text{Time} & \widehat{\mathbf{x}}_s(k+1)\\ \text{Update} & \mathbf{P}(k+1)\\ \text{(Prediction)} & -\rightarrow+\\ k\rightarrow k+1 & \text{Measurement}\\ \widehat{\mathbf{x}}_s^{-}(k+1) & \text{Update}\\ \mathbf{P}^{-}(k+1) & \text{(Correction)}\end{array} \tag{3.16}$$

$$\boxed{\begin{array}{l}\mathbf{k}(k+1)=\mathbf{P}^{-}(k+1)\,\mathbf{c}_d\big(\mathbf{c}_d^{\top}\mathbf{P}^{-}(k+1)\,\mathbf{c}_d+R\big)^{-1}\\ \widehat{\mathbf{x}}_s(k+1)=\widehat{\mathbf{x}}_s^{-}(k+1)+\mathbf{k}(k+1)\left(y(k+1)-\mathbf{c}_d{}^{\top}\widehat{\mathbf{x}}_s^{-}(k+1)\right)\\ \mathbf{P}(k+1)=\left(\mathbf{I}-\mathbf{k}(k+1)\,\mathbf{c}_d{}^{\top}\right)\mathbf{P}^{-}(k+1)\end{array}}$$

is applied to the discretized vehicle model to estimate its states. Then, based on the state estimation, the vehicle lateral acceleration can be obtained by

$$\begin{aligned}\widehat{\mathbf{x}}_s(k+1)&=\begin{bmatrix}\widehat{v}(k+1)\\ \widehat{r}(k+1)\end{bmatrix},\widehat{\mathbf{x}}_s(k)=\begin{bmatrix}\widehat{v}(k)\\ \widehat{r}(k)\end{bmatrix}\\ \widehat{a}_y(k)&=\frac{\widehat{v}(k+1)-\widehat{v}(k)}{\Delta t}+r(k)\,u(k)\end{aligned} \tag{3.17}$$

In Eq. (3.16), $\mathbf{I}$ is an identity matrix.

3.4.3 Authentication & Self-updating Dataset for Lateral Acceleration

According to Eq. (3.7), k_2 is a threshold that determines which estimator should be used. When the model-based observer is used, the authentication module issues a command to update the self-updating dataset by adding the operating point to it. Note that updating

the dataset happens when the fault detection system confirms that the lateral acceleration sensor is healthy.

The estimations provided by the model-based, data-driven, and hybrid estimators for the first test data when $k_2 = 100$ are illustrated in Fig. 3.10. On the horizontal axis of this figure, the red and green colors show when the hybrid estimator uses the data-driven and model-based estimators, respectively. In this estimation, the hybrid estimator uses the model-based observer in 25 points from the 400 points of the test data, and the dataset is updated by adding these 25 points to itself. The RMSEs of the model-based, data-driven and hybrid estimations are 0.0208, 0.0099, and 0.0097, respectively. This result shows that the hybrid estimator outperforms the model-based observer.

The superiority of the hybrid estimator over the data-driven one can be shown when the vehicle encounters unknown environments. The second test data in Table 3.2 is used to simulate one of these conditions. Fig. 3.11 shows that in some test points, the data-driven estimator is not able to estimate since there is no neighbor in their ε-neighborhood. In some other points of this test, the estimation provided by the data-driven estimator is not as accurate as the one obtained by the model-based observer because the dataset is not rich enough around these points. Therefore, the hybrid estimator uses the model-based observer in these points, and it indicates the superiority of the hybrid estimator over the data-driven one. In this test, the hybrid estimator uses the prediction of the model-based observer in almost 280 points from the 400 points of this maneuver. The dataset is updated in these 280 points, so the data-driven estimator can use these new points if the vehicle encounters this environment or similar ones again. Since the data-driven estimator does not have the ability to estimate the entire test data, the RMSE is not determined for that. The RMSEs of the model-based and hybrid estimations are 0.0501 and 0.0497, respectively.

Based on the results it can be concluded that the hybrid estimator can outperform the model-based and data-driven estimators.

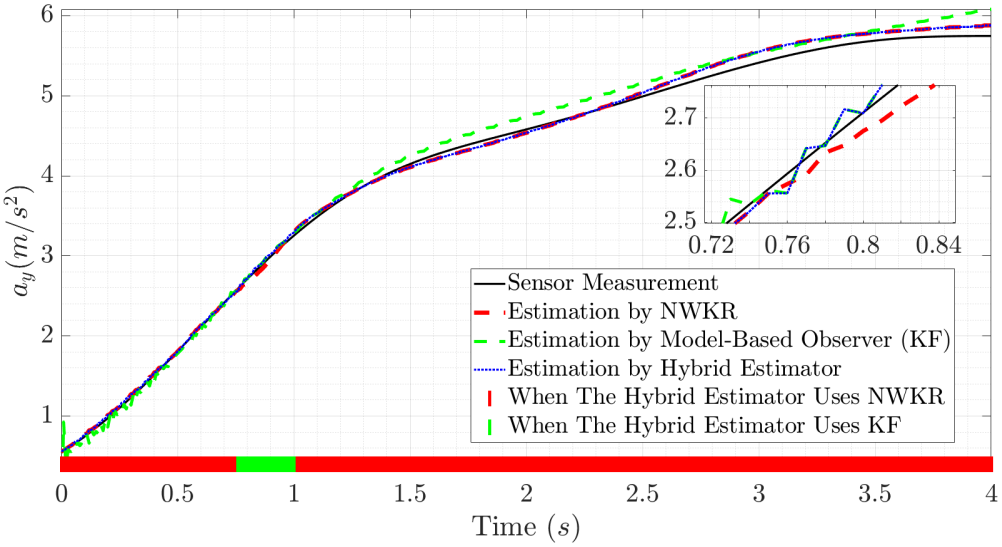

Figure 3.10: Estimation of a_y by the hybrid, model-based, and data-driven estimators for the 1st test data.

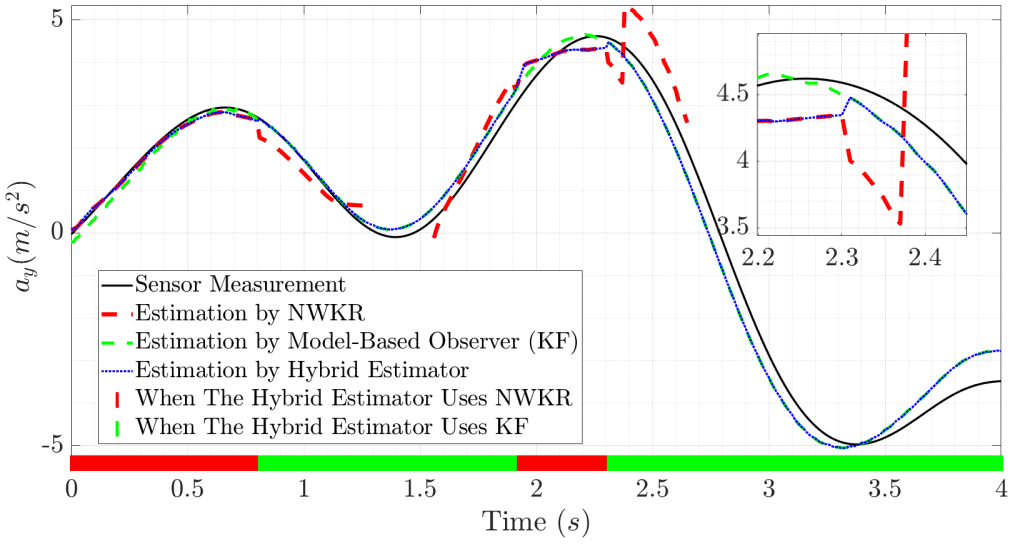

Figure 3.11: Estimation of a_y by the hybrid, model-based, and data-driven estimators for the 2nd test data.

3.4.4 Fault Detector for Lateral Acceleration

In this part, the fault detector module is applied to generate residuals by using the estimations obtained by the hybrid estimator and the measurements provided by the lateral acceleration sensor. If the residuals exceed the threshold L_H and persistently lie outside of the prediction interval, it is an indication of occurring a fault in the lateral acceleration sensor, and if the residuals remain within the prediction interval, it shows that the sensor is healthy.

To simulate a scenario with the faulty lateral acceleration sensor, the measurements of this sensor are multiplied by a factor. To evaluate the performance of the proposed fault detection approach, the first test data with a 20% fault applied to the lateral acceleration sensor from the 3^{rd} second is used. Fig. 3.12 illustrates the residuals of a_y and their prediction intervals for the first test data in the presence of the fault. In this figure, the green and red regions are the prediction intervals for the residuals obtained by using the model-based and data-driven estimators, respectively. Fig. 3.12 shows that the residuals exceed the threshold and persistently lie outside the prediction interval after the 3^{rd} second. It indicates that there is a fault in the lateral acceleration sensor and the proposed fault detection system successfully detects this fault. After fault detection, the estimation of a_y obtained by the hybrid estimator is considered as the reconstruction of the healthy value of the lateral acceleration sensor.

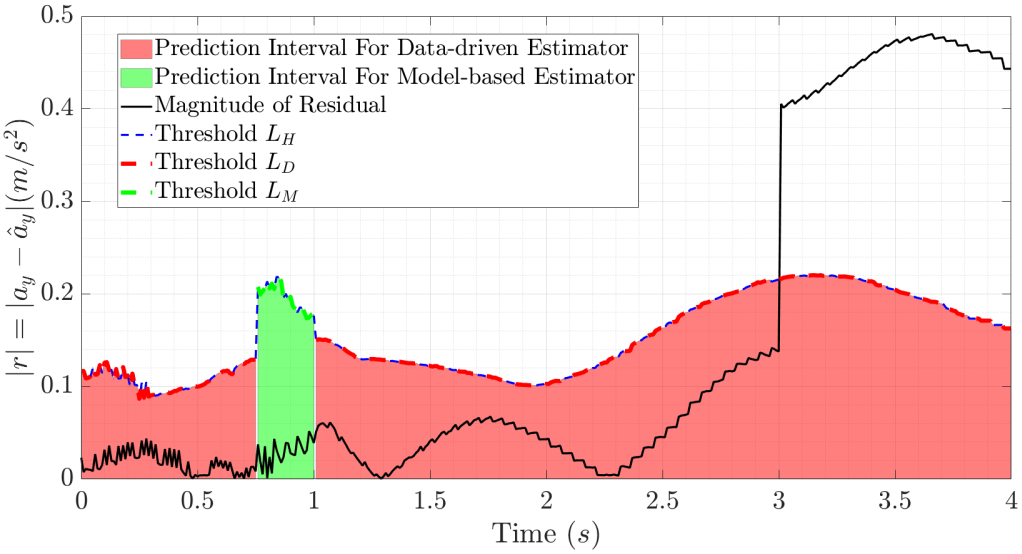

Figure 3.12: The residuals of a_y and their prediction intervals for the 1st test data in the presence of the fault.

3.5 Hybrid Fault Detection and Diagnosis System Applied to Vehicle Traction Motor

In this section, the hybrid fault detection and diagnosis system is applied to detect faults in the vehicle traction motor and find its level of failure. The system's performance is experimentally evaluated with the test vehicle. All required modules for this system are developed in the following subsections.

3.5.1 Data-driven Estimator for Total Traction Torque

NWKR is the data-driven estimator used to estimate the wheels' total torque. To evaluate its performance, the dataset and test data provided in Table 3.2 are utilized. Like the previous section, the same procedure is used to develop the data-driven estimator.

3.5.1.1 Feature Selection by PCC

To select the important features for the estimation of the wheels' total torque, PCCs between all the variables presented in Table 3.3 and the total torque are calculated and shown in Fig. 3.13. This figure illustrates that T has strong linear correlations with a_x and the angular acceleration of all wheels. Based on our physical knowledge, an aerodynamic force can affect the vehicle traction torque especially, in high-speed maneuvers. Since the aerodynamic force can be defined as a function of the vehicle longitudinal velocity, this velocity is also considered another essential feature to estimate the total torque.

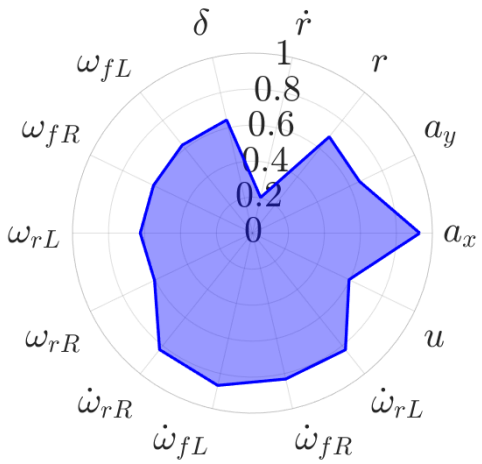

Figure 3.13: PCCs between the wheels' total torque and other variables

The importance of feature selection can be validated by using two NWKR models shown in Fig. 3.14. In the first NWKR, only the selected features are considered as its inputs, and in the second one, all features presented in Table 3.3 are assigned as the inputs of NWKR. Given these two NWKR models, the estimations with different values of bandwidth (σ) are provided to reconstruct the dataset and estimate the first test data (Table 3.2). The RMESs of these estimations are illustrated in Fig. 3.15. This figure illustrates that for the small values of the bandwidth, the 1st NWKR model provides more precise estimations than those obtained by the 2nd NWKR model for the first test data. It shows that the missed variables in the 1st NWKR model not only are not important in the estimation of

T but also adversely affect the estimation accuracy. Fig. 3.15 also shows that especially in the 2nd NWKR model, the excessive reduction of σ causes overfitting. Fig. 3.16 illustrates the desired control action and the estimations of T provided by the two NWKR models with $\sigma = 0.02$ for the first test data.

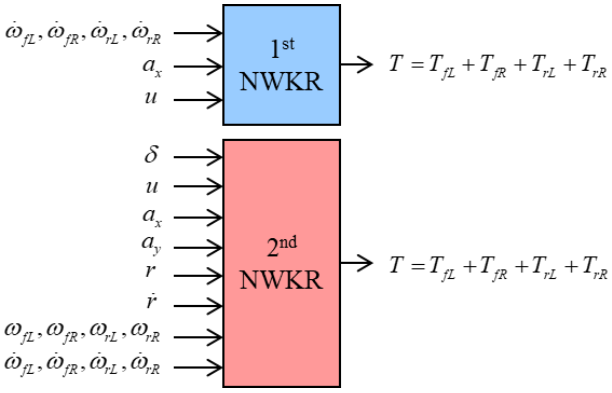

Figure 3.14: Two NWKR models are applied for the wheels' total torque estimation.

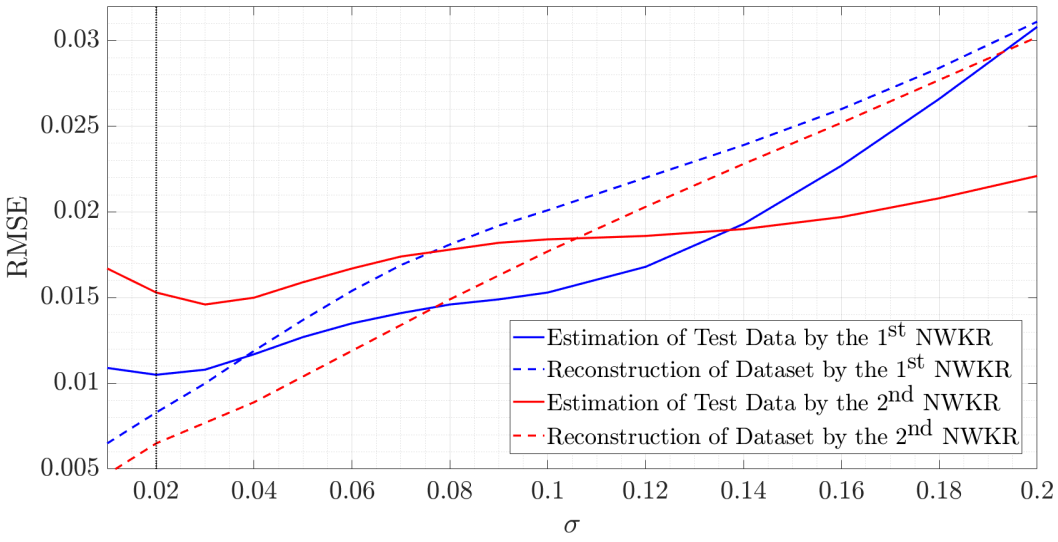

Figure 3.15: RMSE of the estimations provided by the two NWKR models with different values of bandwidth.

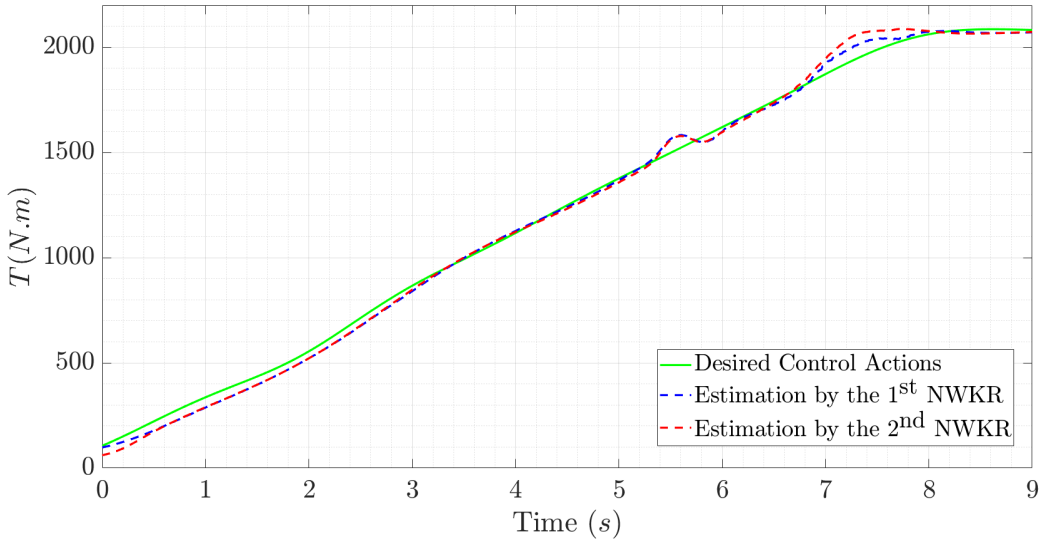

Figure 3.16: The desired control action and the estimations of T provided by the 1st and 2nd NWKR models for $\sigma = 0.02$.

3.5.1.2 Bandwidth Selection

As previously mentioned, the optimal bandwidth is the one that corresponds to the minimum RMSE. Fig. 3.15 (the solid blue line) shows that in the estimation of test data, the minimum RMSE occurs when $\sigma = 0.02$.

3.5.1.3 Estimation Based on Neighbors

To find an appropriate ε-neighborhood, for different values of ε, the RMSEs of the estimations and their computational time are determined. Given these values shown in Fig. 3.17, the RMSE remains constant by reducing ε to 3σ. Fig. 3.17 also illustrates that the computational time of the estimation reduces by shrinking the ε-neighborhood. By considering $\varepsilon = 0.06$, while the accuracy of estimation is preserved, the computational time decreases to almost $1/3$ of the computational time of the estimation in which the ε-neighborhood is not considered.

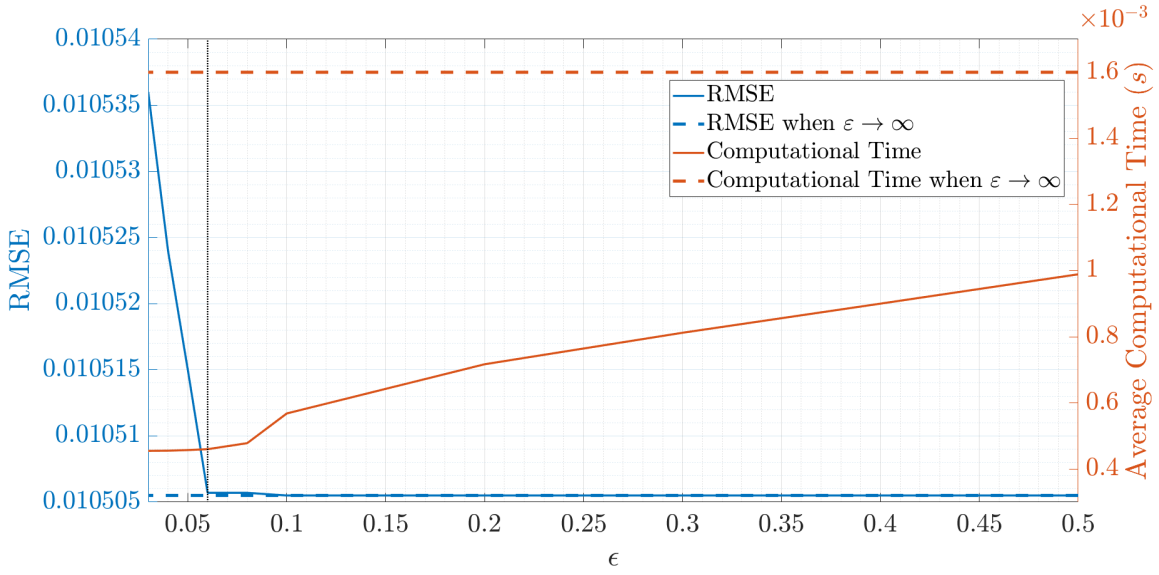

Figure 3.17: The effect of considering the ε-neighborhood on the accuracy of estimation and its computational time.

3.5.1.4 Dataset Management

Like what has been done in Section 3.4.1.4, Algorithm 1 is applied with different values of k_1 and $\varepsilon_1 = 0.02$ to remove the redundant data from the dataset and provide different reduced datasets. The extracted datasets are utilized to estimate the first test data, and the RMSE and average computational time of these estimations are provided in Fig 3.18. This figure illustrates that the desired k_1 is 80 since by using its corresponding dataset, the average computational time decreases while the accuracy of estimation is preserved.

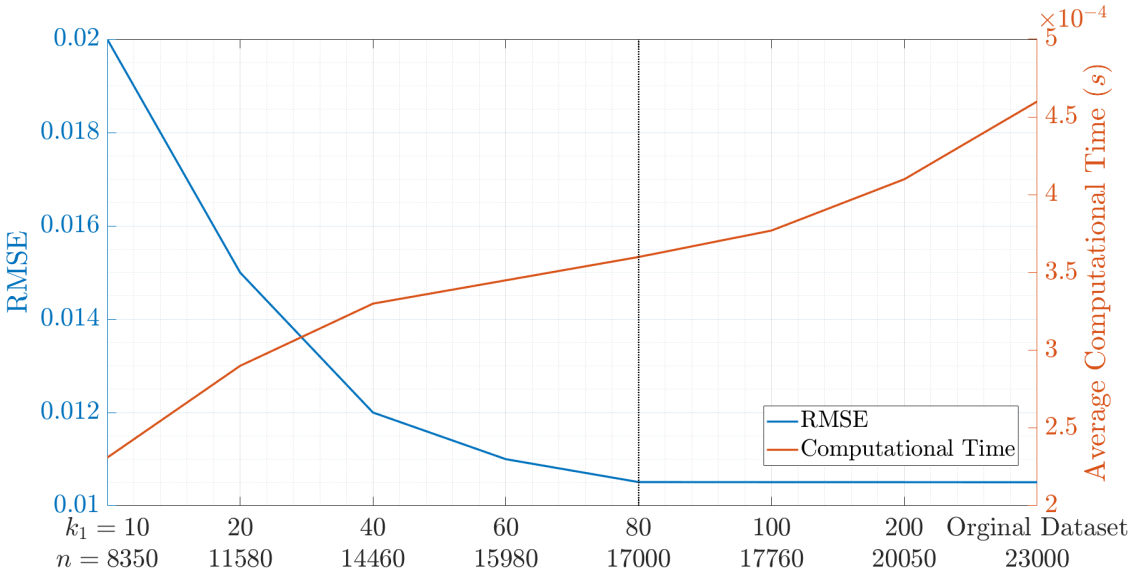

Figure 3.18: The RMSE and average computational time of estimations provided by the reduced datasets.

3.5.2 Model-based Estimator for Total Traction Torque

To design the model-based estimator for the total traction torque, the vehicle longitudinal model and wheel rotational dynamics are used. The wheel rotational dynamics for each wheel is

$$T_{ij} - R_w F_{xij} = I_w \dot{\omega}_{ij} \tag{3.18}$$

in which the subscripts $j \in \{fL,\ fR,\ rL,\ rR\}$ represent the front-left, front-right, rear-left, and rear-right wheels, respectively. By taking the summation of Eq. (3.18) for all four wheels, the following equation is achieved:

$$\underbrace{\sum T_{ij}}_{T} - R_w \underbrace{\sum F_{xij}}_{F_x} = I_w \sum \dot{\omega}_{ij} \tag{3.19}$$

in which F_x is the total traction force of all wheels.

The general form of the vehicle longitudinal dynamics is expressed as:

$$F_x - F_R = Ma_x \tag{3.20}$$

in which F_R is the vehicle's total resistance force. Since this force is considered the summation of aerodynamic and rolling resistance forces, it can be modeled as follows:

$$F_R = A_1 u + A_2 u^2 \tag{3.21}$$

in which A_1 and A_2 are unknown parameters that should be determined. Therefore, based on Eqs. (3.19), (3.20), and (3.21), the following parametric model is considered:

$$\underbrace{I_w \sum \dot{\omega}_{ij} + R_w M a_x}_{z} = \underbrace{\begin{pmatrix} T & A_1 & A_2 \end{pmatrix}}_{\boldsymbol{\theta}^\top} \underbrace{\begin{pmatrix} 1 \\ -u \\ -u^2 \end{pmatrix}}_{\boldsymbol{\varphi}} \Rightarrow z = \boldsymbol{\theta}^\top \boldsymbol{\varphi} \tag{3.22}$$

in which $\boldsymbol{\varphi}$ and z are the regression and output signals, and $\boldsymbol{\theta}$ is a vector of unknown estimated by the following recursive least squares approach [77]:

$$\begin{aligned} &\widehat{\boldsymbol{\theta}}^\top (k) = \begin{pmatrix} \widehat{T}(k) & \widehat{A}_1(k) & \widehat{A}_2 & (k) \end{pmatrix} \\ &\widehat{\boldsymbol{\theta}}(k+1) = \\ &\widehat{\boldsymbol{\theta}}(k) + \mathbf{k}(k+1) \left(z(k+1) - \boldsymbol{\varphi}^\top (k+1) \widehat{\boldsymbol{\theta}}(k) \right) \\ &\mathbf{k}(k+1) = \frac{\mathbf{P}(k) \boldsymbol{\varphi}(k+1)}{(\lambda + \boldsymbol{\varphi}^\top (k+1) \mathbf{P}(k) \boldsymbol{\varphi}(k+1))} \\ &\mathbf{P}(k+1) = \left(\mathbf{I} - \mathbf{k}(k+1) \boldsymbol{\varphi}^\top (k+1) \right) \mathbf{P}(k) / \lambda \end{aligned} \tag{3.23}$$

in which λ is the forgetting factor, and $\mathbf{I}$ is an identity matrix. The recursive least squares approach guarantees that $\boldsymbol{\theta} \to \boldsymbol{\theta}^*$ when $t \to \infty$ if $\boldsymbol{\varphi}$ is persistently exciting [78]. $\boldsymbol{\theta}^*$ is the actual value of $\boldsymbol{\theta}$, and the first element of $\boldsymbol{\theta}$ shows the total traction torque estimation.

3.5.3 Authentication & Self-updating Dataset for Total Traction Torque

Given k_2 in Eq. (3.7), the authentication module decides to choose between the model-based and data-driven estimators and also determines when the self-updating dataset

should be updated. The estimations provided by the model-based, data-driven, and hybrid estimators for the first test data when $k_2 = 100$ are shown in Fig. 3.19. The green and red colors on the horizontal axis of this figure illustrate when the hybrid estimator uses the model-based and data-driven estimators, respectively. In this test, the hybrid estimator uses the model-based estimator in 99 points from the 1000 points, and the dataset is updated in these 99 points. The RMSEs of the model-based, data-driven and hybrid estimations are 0.0150, 0.0167, and 0.0139, respectively. This result shows that the hybrid estimator outperforms the others. To evaluate the performance of the hybrid estimator when the vehicle faces new environments, the second test data in Table 3.2 is used, and its result is shown in Fig. 3.20. This figure illustrates that in some test points, the data-driven estimator is not able to estimate or its estimation is not as precise as the one obtained by the model-based estimator due to the lack of neighbors in the dataset. Therefore, in these points, the hybrid estimator uses the model-based estimator, and the dataset is updated by adding them to itself. Fig. 3.20 shows that in 580 points from the 900 points of this maneuver, the hybrid estimator uses the estimation obtained by the model-based estimator. Since the data-driven estimator cannot estimate the entire test data, the RMSE is not determined for that. The RMSEs of the model-based and hybrid estimations are 0.0258 and 0.0251, respectively. The result shows the superiority of the hybrid estimator over the data-driven one when the vehicle encounters unknown environments.

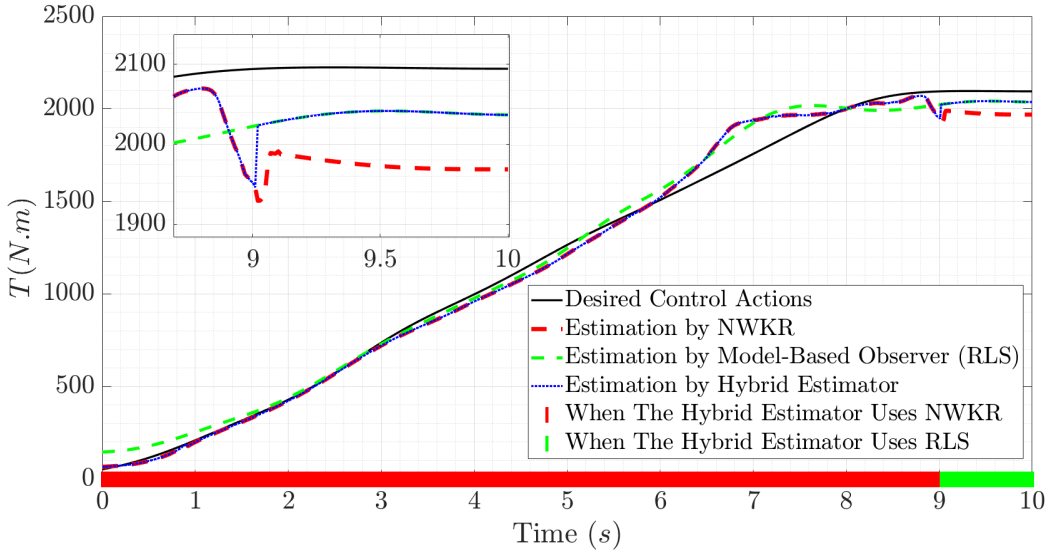

Figure 3.19: Estimation of T by the hybrid, model-based, and data-driven estimators for the 1st test data.

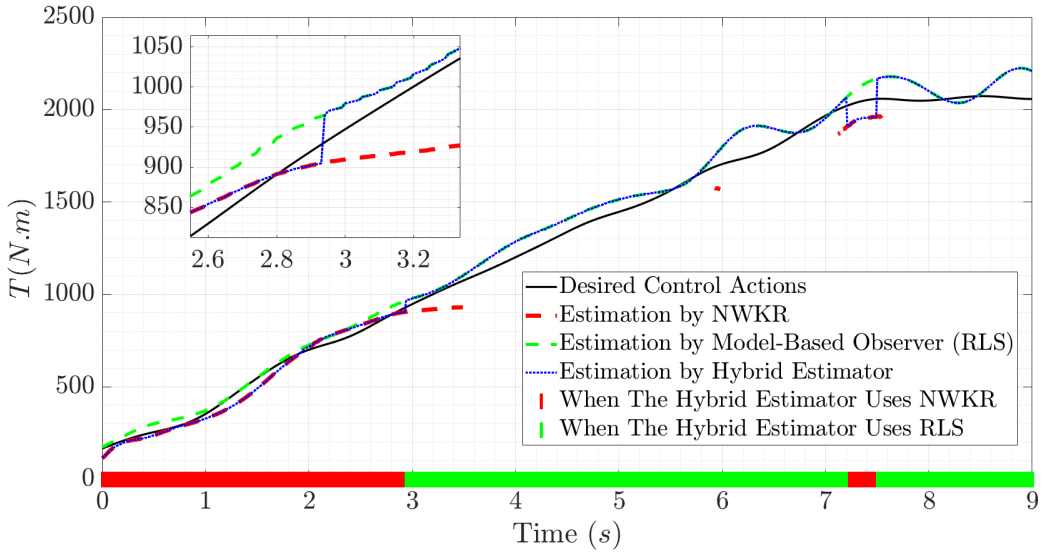

Figure 3.20: Estimation of T by the hybrid, model-based, and data-driven estimators for the 2nd test data.

3.5.4 Fault Detector for Total Traction Torque

In the fault detector module, the residuals are obtained by comparing the estimations provided by the hybrid estimator and the desired control command determined by controllers or a driver. The traction motor is healthy while the residuals are placed within the prediction interval. If the residuals exceed the threshold L_H and persistently lie outside of the prediction interval, it indicates that a fault occurs in the vehicle traction motor.

To simulate a fault in the vehicle traction motor, the desired control actions are multiplied by a factor, then these new control actions are applied to the vehicle. To evaluate the performance of the proposed fault detection approach, a 35% fault is applied to the traction motor after the 10^{th} second of the first test data. Fig. 3.21 illustrates the residuals of T and their prediction intervals for the first test data in the presence of the fault. Fig. 3.21 illustrates that the residuals exceed the threshold and persistently lie outside the prediction interval after the 10.18^{th} second. It represents that there is a fault in the vehicle traction motor and the proposed fault detection system appropriately detects this fault. Fig. 3.22 shows the estimation of the actual actuator commands obtained by the hybrid estimator. This figure illustrates that after occurring the fault, the estimation of the actual actuator command is almost placed between 60% to 70% of the desired control actions. Therefore, the failure level of the actuator is estimated between 30% and 40%. It shows that the failure level of the traction motor is successfully determined since its actual failure level is 35%.

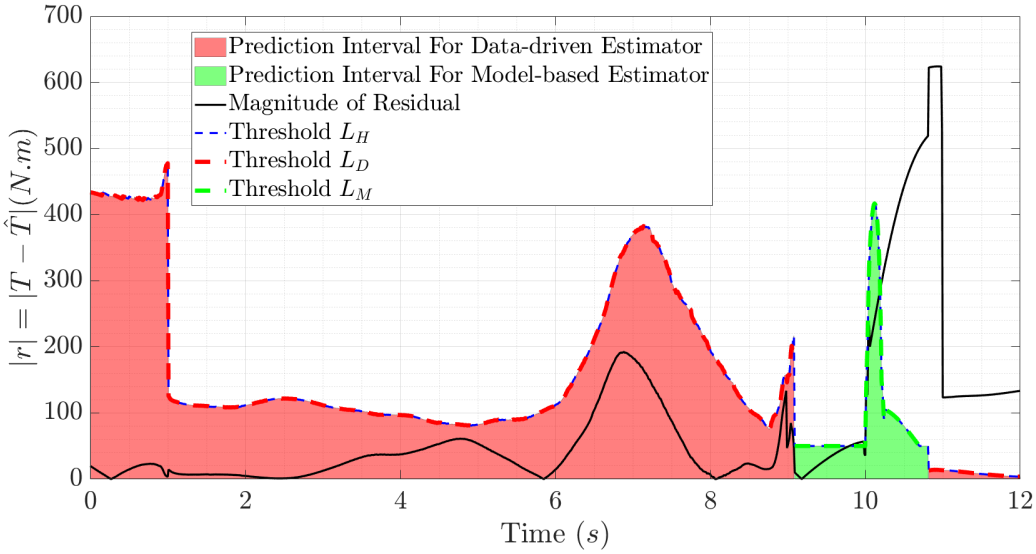

Figure 3.21: The residuals of T and their prediction intervals for the 1st test data in the presence of the fault.

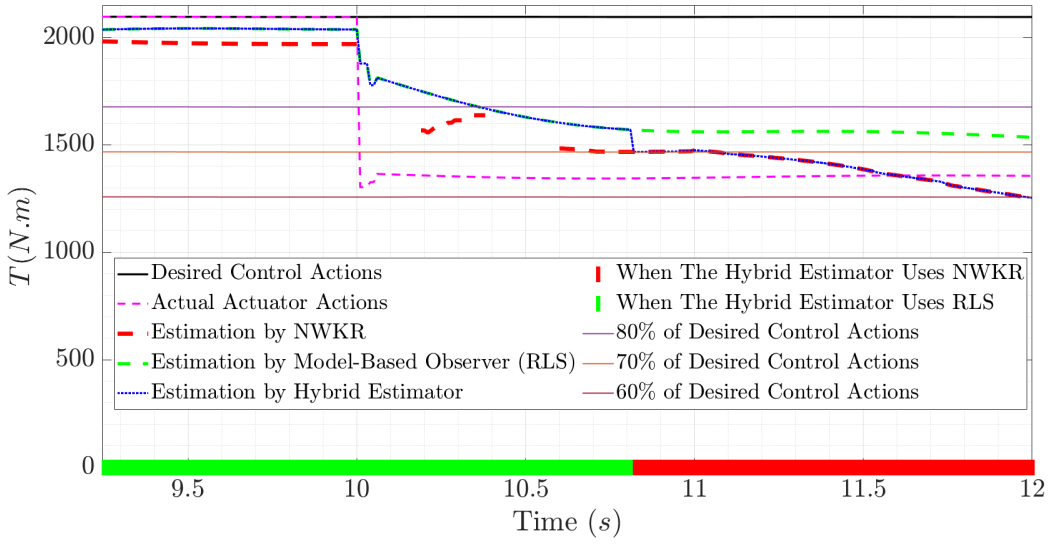

Figure 3.22: The desired, actual, and estimated value of T after the vehicle traction motor becomes faulty.

3.6 Summary

In this chapter, a hybrid model/data fault detection and diagnosis methodology was developed for any vehicle sensor and actuator. This system generally works based on the information redundancy obtained by a hybrid estimator. In the hybrid estimator, the authentication module decides how to choose the model-based and data-driven estimators based on the availability of data around the operating points. The hybrid estimator uses the model-based and data-driven estimators in unknown and known conditions, respectively. The authentication module also issues a command to update the self-updating dataset when the vehicle faces unknown conditions or environments. In this study, Nadaraya-Watson kernel regression (NWKR) was used as a data-driven estimator. Since NWKR is an instant-based learning method, it does not need any pre-training, and new data can be used immediately upon collection. Pearson Correlation Coefficient is used to select the essential features required for the data-driven estimator. Regarding the nature of the Gaussian kernel function in NWKR, it is shown that the nearest neighbors of each test point play the main role in estimation. Therefore, an ε-neighborhood is defined so that the prediction is conducted based on only the points in this neighborhood. In addition, a data management approach was proposed to remove redundant points from the database. Using points placed in the ε-neighborhood for estimation and applying the proposed data management cause a significant reduction in the computational time of estimation, which is essential for real-time fault detection. The performance of this fault detection system was evaluated by applying it to the vehicle's lateral acceleration sensor and traction motor. For the lateral acceleration sensor, a Kalman filter applied to a vehicle single-track model was used as the model-based estimator, and for the traction motor, the recursive least squares approach applied to the vehicle longitudinal model is proposed as the model-based estimator. NWKR was used as the data-driven estimator in these two cases. The results of the experimental tests conducted on the all-wheel-drive test vehicle confirm that the hybrid estimator outperforms the model-based and data-driven estimators used individually. These results also justify that the proposed fault detection and diagnosis system is able to successfully detect the faults in the target sensor or actuator, then reconstruct the healthy value of the faulty sensor or find the failure level of the faulty actuator.

The hybrid fault detection and diagnosis approach proposed in this chapter is applicable to one target component (a sensor or an actuator) in the vehicle. Therefore, since the target component is known, there is no need for fault isolation. The need for fault isolation arises when a set of sensors and actuators is considered. In the next chapter, a data-driven proposed is proposed to monitor the health status of a set of sensors and actuators. This approach is able to detect, isolate and quantify a fault occurring in these components.

Chapter 4

Data-Driven Sensor and Actuator Health Monitoring System

4.1 Inroduction

The main goal of this chapter is to develop a reliable data-driven health monitoring system for a set of vehicle sensors and actuators. The algorithm is built upon the coherency of the vehicle data at any given time. The vehicle data must be coherent, meaning that it must reflect the physics of the vehicle motion and the causality between states. The contribution of this chapter is threefold:

- A fault detection algorithm is proposed that works based on the coherency among the target vehicle's variables. If a fault occurs in the vehicle, the coherency is violated.
- The proposed fault isolation algorithm checks the coherency among the subsets of the target vehicle's variables. This technique finds the subset of the data in which the coherency among the data holds. In this case, the faulty components are those not included in that subset.
- Fault quantification determines the type and magnitude of the fault that can be used to reconstruct sensor data or find the level of actuator failure. The proposed

fault quantification system applies Nadaraya-Watson Kernel Regression (NWKR) to reconstruct the healthy value of the faulty sensor and estimate the actual action of the faulty actuator by using the remaining healthy components.

This chapter is organized as follows: Section 4.2 explains the developed data-driven health monitoring system in detail. In Section 4.3, this health monitoring system is applied vehicle and in Section 4.4 its performance is evaluated by experimental results. Finally, the chapter is summarized in Section 4.5.

4.2 Health Monitoring System

A vehicle's motion follows physics laws and its variables at each time instance should not violate these physical causalities. These variables include measurements provided by the vehicle's sensors and desired control actions obtained by its controllers or driver. The proposed health monitoring system checks whether the string of these variables belongs to the target vehicle or not. If this string of data does not belong to the vehicle, this system notifies that a fault occurs in the vehicle's sensors or actuators. After detecting a fault, to isolate the fault, the health monitoring system searches to find the data coherency in the subsets of the string of data and determine which of these subsets belongs to the target vehicle. The faulty component is the variable that is absent from the determined subset. In this study, it is assumed that at most one component becomes faulty at a time although the proposed health monitoring system can be easily extended for multiple-fault conditions. After fault localization, the magnitude and type of the fault are determined by using the healthy components to reconstruct sensor data or find the level of actuator failure. The reconstruction of sensor data is essential since the vehicle's controllers may use the sensor data to obtain the control actions. When a sensor becomes faulty, if the controllers use the measurement of the faulty sensor, they will generate inappropriate control actions that may cause the vehicle's instability. Therefore, when a sensor becomes faulty, the vehicle's controllers should use its reconstructed healthy value instead of its faulty measurement to generate appropriate control actions. Determining the actuator failure level is also beneficial, since desired control actions may be obtained by scaling the actuator output

in partial failures. Therefore, the proposed health monitoring system can be broken down into three subsystems, including fault detection, isolation, and quantification systems as will be explained in detail in the following sections.

4.2.1 Fault Detection System

The proposed fault detection system is responsible to notify the fault when it occurs in the vehicle's sensors or actuators. This system works based on checking the coherency among the data received from the vehicle by using the pre-collected data. Since a vehicle is a deterministic system, by considering all variables presented in the vehicle's governing equations of motion, it is possible to check whether these variables' values belong to the target vehicle and specify the health status of the target vehicle at the current time. These variables, also called *features*, are provided by sensor measurements and desired control actions and can be expressed in the form of a data string. Throughout this chapter, vectors and matrices are shown in bold lowercase and bold capitals, respectively. Let $\mathbf{x} \in \mathbb{R}^n$ denote an n-dimensional data string, i.e.,

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top, \tag{4.1}$$

where x_i, $i = 1, \ldots, n$ corresponds to a feature. Similarly, an operating point, $\mathbf{x}(i) = (x_1(i), x_2(i), \ldots x_n(i))^\top$, is defined as a point at which the vehicle is currently working. Additionally, let

$$\mathscr{D} = \left\{ \mathbf{x}^j \in \mathbb{R}^n | j = 1, \ldots, m \right\}. \tag{4.2}$$

denote the pre-collected dataset.

The main idea of the proposed fault detection technique is as follows. First, the vehicle is locally modeled around the operating point by a multiple linear model. This is in fact a multi-dimensional linear surface estimated using data points around the operating point. Then using the distance of the operating point from this linear surface, the health status of the operating point is specified. To estimate a multi-dimensional linear surface, in the dataset, the data points placed in an ε-neighborhood are used. An ε-neighborhood is a multi-dimensional cube with edge length 2ε around the operating point. These data points

placed in an ε-neighborhood, also called *neighbors*, are defined as

$$\left\{ \mathbf{x}^j \in \mathscr{D} \middle| \left\| \mathbf{x}\left(i\right) - \mathbf{x}^j \right\|_\infty < \varepsilon,\ j = 1, ..., m \right\}. \tag{4.3}$$

To create an accurate linear surface estimation, a rich dataset is required. Hence, if the created neighborhood is not rich enough, meaning that the number of neighbors is less than a threshold denoted by k, previous operating points placed within a pre-specified window of time are considered. In other words, if the neighborhood is rich, the selected data points are $\left(\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^N\right)$, otherwise, are $\left(\mathbf{x}\left(i-h\right), \ldots, \mathbf{x}\left(i-1\right)\right)$, where h is the time window size. The selected data points, either neighbors or history points, are used to form the input matrix $\mathbf{X} \in \mathbb{R}^{N \times n}$. The least square method is a form of statistical regression analysis used to determine the linear surface of best fit for a set of data points. Each point of data represents the relationship between input variables and an output variable. This method creates a model that minimizes the sum of the squared residuals. In general, for a given input matrix $\mathbf{X}^* \in \mathbb{R}^{N \times n}$ and an output vector $\mathbf{y}^* \in \mathbb{R}^N$, the least square method aims to solve the quadratic problem [79]

$$\widehat{\boldsymbol{\beta}}^* = \underset{\boldsymbol{\beta}^* \in \mathbb{R}^n}{\arg\min} \left\| \mathbf{y}^* - \mathbf{X}^* \boldsymbol{\beta}^* \right\|_2^2. \tag{4.4}$$

The estimated weight vector $\widehat{\boldsymbol{\beta}}^*$ is used for estimations and predictions. Importantly, for a new input vector, $\mathbf{x}^*_{\text{new}} \in \mathbb{R}^N$ and an output y^*_{new}, the prediction value and its corresponding residual are defined as $\widehat{y}^*_{\text{new}} = \mathbf{x}^\top_{\text{new}} \widehat{\beta}^*$ and $r_{\text{new}} = \widehat{y}^*_{\text{new}} - y^*_{\text{new}}$, respectively. In order to align the aforementioned least square method with the proposed fault detection algorithm, the following notations are defined. In the string of data $\mathbf{x}$, for an arbitrary feature x_q, let $y = x_q$, and $\mathbf{x}_{-q} = (1, x_1, \ldots, x_{q-1}, x_{q+1}, \ldots, x_n)^\top$. Similarly, given the matrix $\mathbf{X}$, $\mathbf{X}_{-q} = \left(\mathbf{x}^1_{-q}, \ldots, \mathbf{x}^N_{-q}\right)^\top$ and $\mathbf{y} = \left(x^1_q, \ldots, x^N_q\right)^\top$. With the above notations, the least square function for the proposed fault detection system is given by

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\arg\min} \left\| \mathbf{y} - \mathbf{X}_{-\mathbf{q}} \boldsymbol{\beta} \right\|_2^2. \tag{4.5}$$

The solution of Eq. (4.5) is used to calculate the predicted value of $x_q(i)$, the q^{th} element of the operating point, which is given by

$$\widehat{x}_q(i) = x^\top_{-q} \widehat{\boldsymbol{\beta}} = x^\top_{-q} \left(\mathbf{X}^\top_{-q} \mathbf{X}_{-q}\right)^{-1} \mathbf{X}^\top_{-q} \mathbf{y}. \tag{4.6}$$

The next step is to determine the health status of the operating point. This step is executed using the residual $r(i) = \widehat{x}_q(i) - x_q(i)$ for the corresponding operating point. If this residual lies in a prediction confidence interval $(-L_r(i), +L_r(i))$, then the operating point belongs to the target vehicle and the vehicle is healthy, otherwise, faulty. The boundary (threshold) of the prediction confidence interval, $L_r(i)$, is given by

$$L_r(i) = t_{\frac{\alpha}{2},df} \sqrt{\widehat{\sigma}^2 \left(1 + \mathbf{x}_{-q}^{\top}(i) \left(\mathbf{X}_{-q}^{\top} \mathbf{X}_{-q}\right)^{-1} \mathbf{x}_{-q}(i)\right)}, \tag{4.7}$$

where $100 \times \frac{\alpha}{2}$-th quantile of t-distribution with nominal level α, and degrees of freedom $df = N - n - 1$ [79]. Note that the $\widehat{\sigma}^2$ known as *Mean Square Error (MSE)* is the estimated value of noise variance. This estimation is given by

$$\widehat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} r_i^2, \tag{4.8}$$

where $r_i = \widehat{y}_i - y_i$, and $\widehat{y}_i = \mathbf{x}_i^{\top} \widehat{\beta}$, $i = 1, \ldots, N$ are the residuals and fitted values for the modeled linear surface, respectively. The procedure of the proposed residual analysis is presented in Algorithm 2.

Algorithm 2 Residual Analysis

Input: Dataset $\mathbf{X}$, operating point $\mathbf{x}(i)$.

Output: Determine the vehicle's health status.

Step 1: Fit a linear model by solving the lease square problem

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^n}{\arg\min} \|\mathbf{y} - \mathbf{X}_{-q}\boldsymbol{\beta}\|_2^2$$

and predict $x_q(i)$ using

$$\widehat{x}_q(i) = x_{-q}^\top \left(\mathbf{X}_{-q}^\top \mathbf{X}_{-q}\right)^{-1} \mathbf{X}_{-q}^\top \mathbf{y}$$

Step 2: Compute the residual $r(i)$ for the operating point by

$$r(i) = \widehat{x}_q(i) - x_q(i)$$

Step 3: Compute the prediction interval $\left(-L_r\left(i\right), +L_r\left(i\right)\right)$ for $r(i)$, where

$$L_r\left(i\right) = t_{\alpha/2,df}\sqrt{\widehat{\sigma}^2\left(1 + \mathbf{x}_{-q}^\top\left(i\right)\left(\mathbf{X}_{-q}^\top \mathbf{X}_{-q}\right)^{-1}\mathbf{x}_{-q}\left(i\right)\right)}$$

Step 4: Determine the health status of the operating point. It is healthy if $|r\left(i\right)| < L_r\left(i\right)$; Otherwise, it is faulty .

It should be mentioned that a residual is a vertical distance (r) between the actual point and the estimated or predicted point in a linear regression model. Moreover, the orthogonal distance (d) is the length of the line from the actual point that is perpendicular to the estimated linear surface. In fact, the orthogonal distance is the main criterion that shows the deviation of the actual point from the estimated surface. The residual determined in Algorithm 2 can be easily converted to the orthogonal distance using the equation $d\left(i\right) = |r\left(i\right) \times \cos\theta|$, where θ is the angle between the normal vector of the linear surface and the vertical direction. This equation along with Eq.(4.7), results in the threshold of the prediction interval based on the orthogonal distance, $L_d(i)$, which

is determined by $L_d(i) = L_r(i)|\cos\theta|$. For clarification, the schematics of the residual, (orthogonal) distance, and their corresponding thresholds are shown in Fig. 4.1

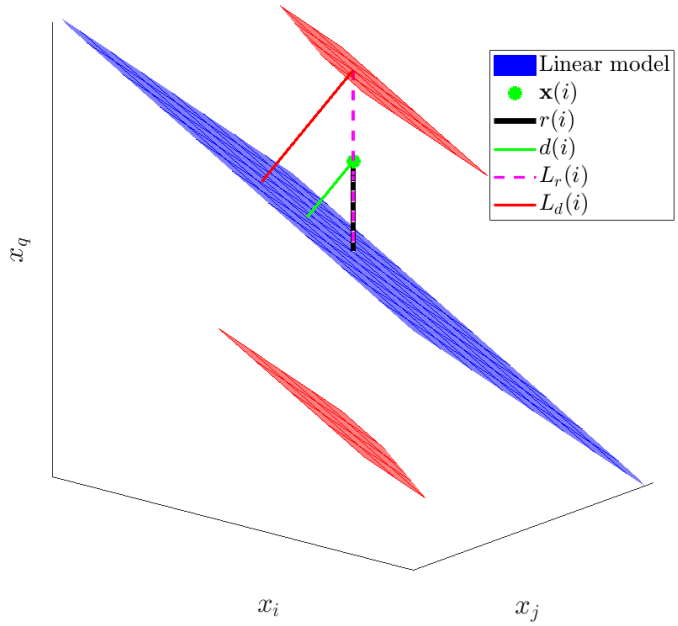

Figure 4.1: The schematics of the residual, orthogonal distance, and their corresponding thresholds.

The diagram in Fig. 4.2 visually illustrates how the proposed fault detection system performs. As previously mentioned, when the history of operating points is used in the fault detection algorithm, it is shown that the pre-collected dataset is not rich enough around the current operating point. On this condition, if this algorithm confirms that the current operating point is healthy, this healthy point is called *missing data* and it is added to the dataset to update it. This kind of dataset is called *Self-updating dataset*.

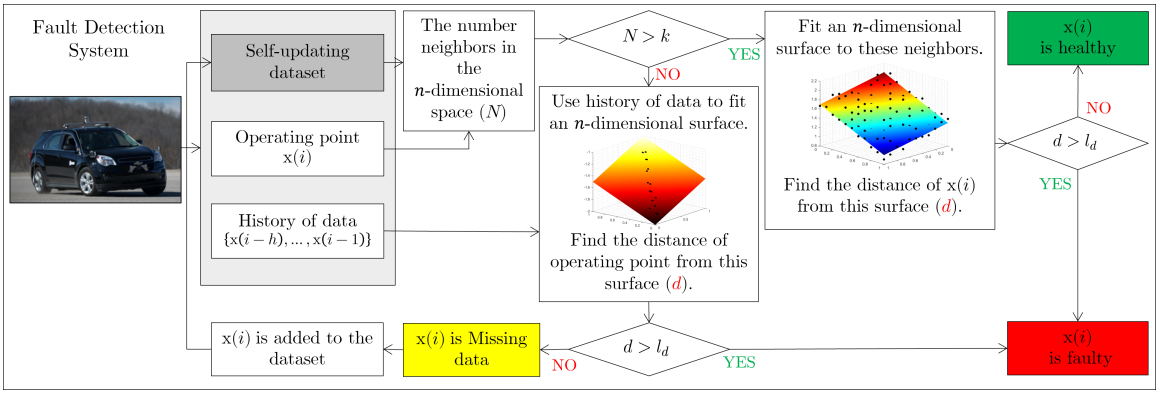

Figure 4.2: The overview of the proposed fault detection system.

Figures 4.3 and 4.4 visualize the orthogonal distances for faulty and healthy operating points when the string of data has two features. Fig. 4.3 shows the cases when the dataset is rich enough around the operating point.

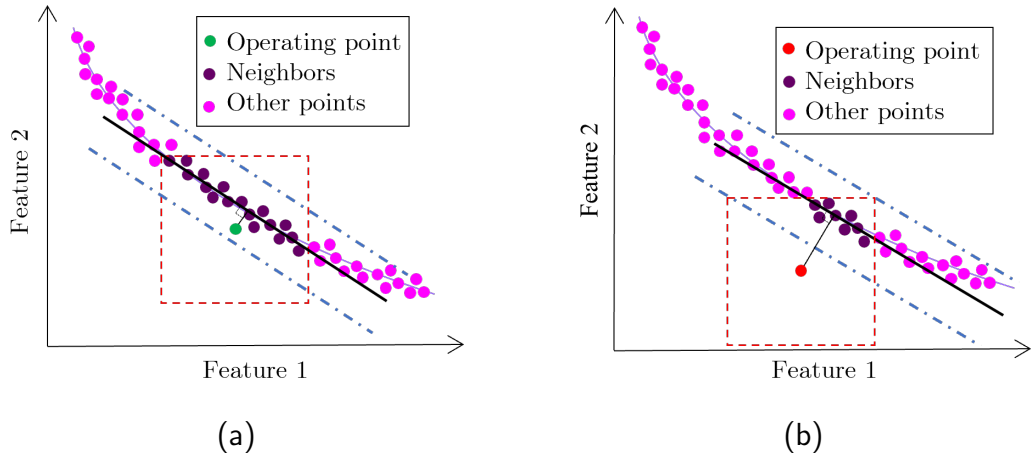

Figure 4.3: Visualization of the concept of the proposed fault detection system when data are available around the operating point. (a) Healthy Case. (b) Faulty Case.

In Fig.4.3a, first, by considering an ε-neighborhood around the operating point (green point), the operating point's neighbors are determined, and then a two-dimensional surface

is fitted to these neighbors. By considering a prediction interval (blue lines), it can be shown that the operating point is placed within this prediction interval. Therefore, it shows that the operating point belongs to the target system and the system is healthy. This process is also performed in Fig. 4.3b, but this time, the distance of the operating point (red point) from the fitted surface exceeds the threshold and the operating point lies outside the prediction interval. This point shows that the operating point does not belong to the target system, so there is a fault in the target system.

Fig. 4.4 visualizes the cases when the dataset is not rich enough around the operating point. On this condition, a set of previous operating points is used to form a linear surface, and the distance of the current operating point from the fitted surface can indicate whether the current operating point belongs to the target system or not. Fig. 4.4a shows that if this distance is less than the determined threshold (yellow lines) and the current operating point is placed within the prediction interval, this point belongs to the target vehicle, but it is missing from the dataset. Since there are not enough neighbors around the operating point, this point is added to the dataset to update it. Fig. 4.4b illustrates that if this distance exceeds the threshold and the current operating point is placed outside the prediction interval, it shows that the operating point does not belong to the target system and there is a fault occurring in the target system.

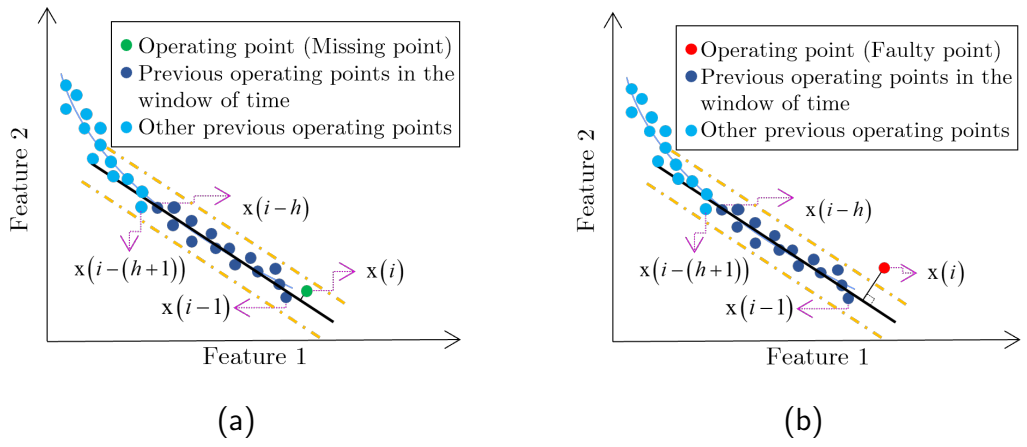

Figure 4.4: Visualization of the concept of the proposed fault detection system when data are not available around the operating point. (a) Healthy Case. (b) Faulty Case.

4.2.2 Fault Isolation System

Algorithm 2 proposed in the previous subsection can specify the status of the operating point. If the operating point is determined as faulty, the next step is to isolate the faulty feature. As previously mentioned, it is assumed that at most one feature (sensor or actuator) becomes faulty at a time in this study. For fault isolation, based on the string of data, all subspaces in each of which one feature is absent are explored. To this end, in each subspace, first, an ε-neighborhood is considered around the projected operating point to find neighbors, and then Algorithm 2 is applied to check the health status of the projected operating point. Algorithm 2 shows that the projected operating point is healthy in only one subspace, and the faulty feature is the feature that is absent from this subspace. Only in this subspace, the projected operating point is placed within the prediction interval and its distance from the fitted surface does not exceed the corresponding threshold. This algorithm can be extended to isolate multiple faults by exploring lower-dimensional subspaces.

Fig. 4.5 visualizes this isolation process in three-dimensional space. In Fig. 4.5a, the blue points show the points in the dataset and the red point is an operating point whose Feature 3 is made faulty. Figs. 4.5b, 4.5c, and 4.5d illustrate these points in the subspaces from which Features 1, 2, and 3 are respectively removed. In each of these subspaces, an ε-neighborhood is plotted around the operating point, and then a two-dimensional surface is fitted to neighbors placed in this ε-neighborhood. It can be shown that only in the subspace from which Feature 3 is absent, the projected operating point is placed within the prediction interval. Based on the proposed approach for fault isolation, the faulty feature is Feature 3, so the existence of a fault in Feature 3 is verified. The diagram in Fig. 4.6 displays the general overview of this fault isolation process.

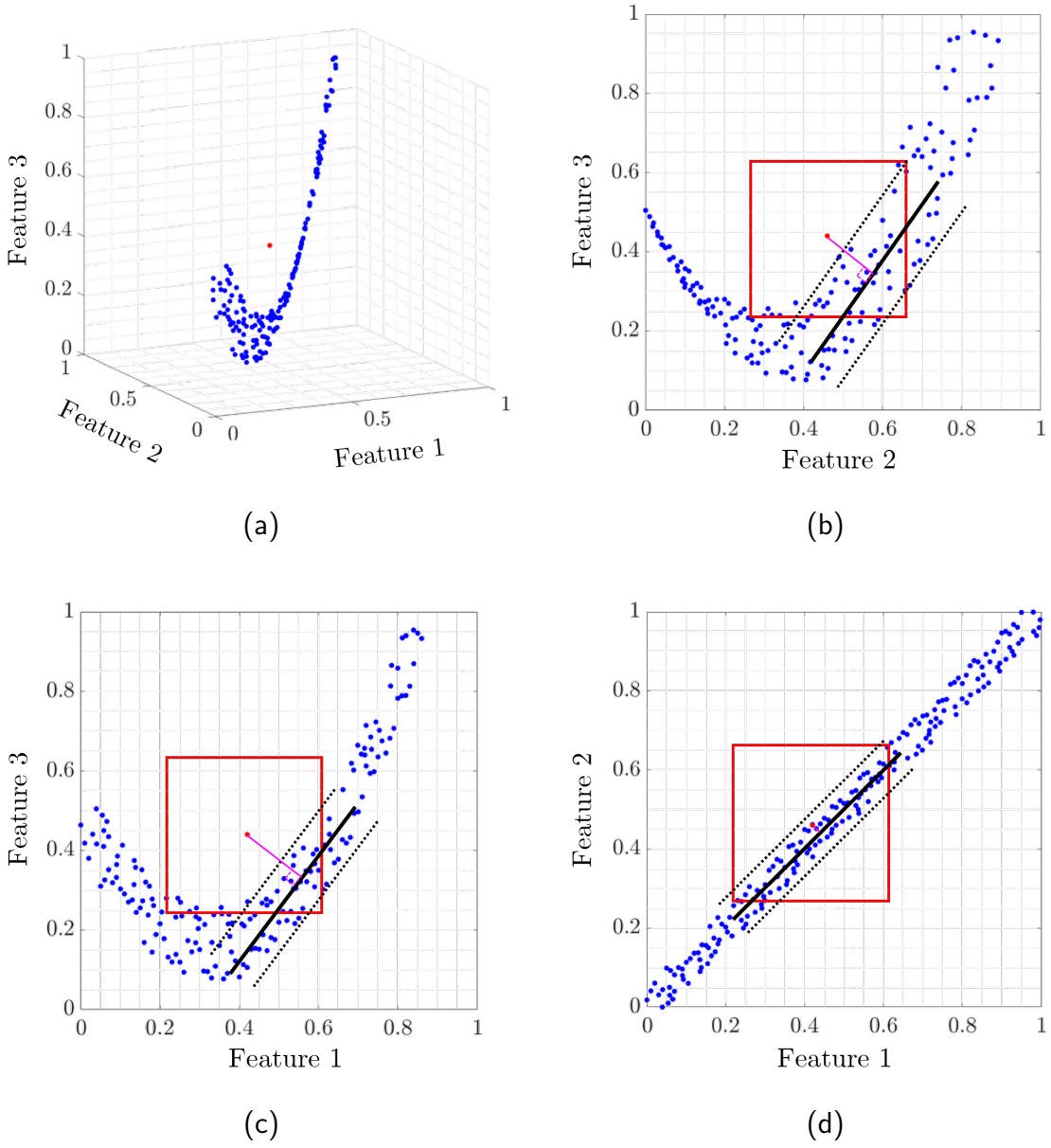

Figure 4.5: Visualization of the concept of the proposed fault isolation system. (a) The red point is the operating point whose faulty feature is Feature 3 and the blue points are data in the dataset in three-dimensional space. Projection of all these points into the plane whose absent feature is Feature 1 (b)/Feature 2 (c)/Feature 3 (d).

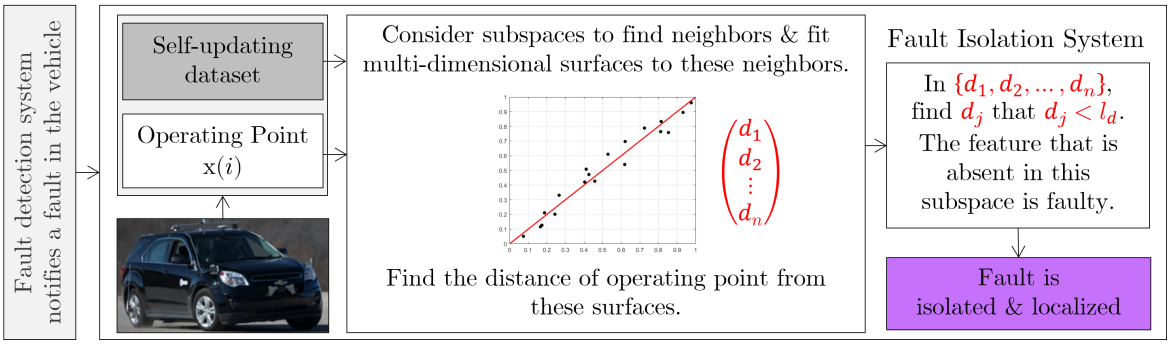

Figure 4.6: The overview of the proposed fault isolation system.

4.2.3 Fault Quantification System

After detecting and isolating the fault, it is required to quantify the fault and reconstruct the actual value of the faulty feature. If an actuator becomes faulty, the fault quantification system can determine its failure level by using the estimation of its actual output. If a sensor becomes faulty, by using its reconstructed healthy value, appropriate control actions can be obtained to achieve safe and satisfactory vehicle performance. After localizing the source of the fault, by removing the faulty feature from the string of data and by using the remaining healthy features, the proposed fault quantification system applies Nadaraya-Watson Kernel Regression (NWKR) to reconstruct the healthy value of the faulty sensor or estimate the actual action of the faulty actuator. As previously mentioned in the proposed fault detection system, a self-updating dataset is proposed to be updated by adding missing data to itself. Estimation based on the updated dataset raises a need of using instant-based learning methods (e.g., kernel regression methods) for the data-driven estimator. NWKR is one of the well-known kernel regression techniques that works based on the extraction of a nonlinear relationship that is the best fit to the given dataset, and it can estimate the unseen data with weighted averaging [66].

Based on the definition of $\mathbf{x}$ in Eq. (4.1) and without loss of generality, it is assumed that the faulty feature is $x_p(i)$ (the p^{th} feature of the operating point). By eliminating this feature from the string of data and using the dataset, NWKR can estimate the healthy

value of this faulty feature as follows:

$$\widehat{x}_p(i) = \frac{\sum_{j=1}^{m} x_p^j \left(\prod_{k=1,k\neq p}^{k=n} K_\sigma\left(x_k(i), x_k^j\right) \right)}{\sum_{j=1}^{m} \left(\prod_{k=1,k\neq p}^{k=n} K_\sigma\left(x_k(i), x_k^j\right) \right)} \tag{4.9}$$

where $x_k(i)$ is the k^{th} feature of the operating point, and x_k^j is the k^{th} feature of the j^{th} point in the dataset. In addition, $K_\sigma(\,,\,)$ is the Gaussian kernel function defined as,

$$K_\sigma\left(x_k(i), x_k^j\right) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-\left(x_k(i) - x_k^j\right)^2}{2\sigma^2} \right) \tag{4.10}$$

in which σ is the standard deviation of the Gaussian function, also known as the bandwidth of the kernel function.

In the proposed reconstruction approach, the faulty feature, $x_p(i)$, is considered as the output of NWKR, and all remaining healthy features are considered as NWKR's inputs.

Based on the nature of the Gaussian kernel function, if $\left|x_k(i) - x_k^j\right| > 3\sigma$, $K_\sigma\left(x_k(i), x_k^j\right)$ has a negligible value, according to the NWKR formula, x^j can be omitted to participate in the estimation of $\widehat{x}_p(i)$. Therefore, in the subspace from which the faulty feature is absent, an ε-neighborhood with $\varepsilon = 3\sigma$ is considered around the operating point to find its neighbors, and only these neighbors participate in the estimation process. Based on this idea, in Eq. (4.9), m will be the number of these neighbors, and x_k^j will be the k^{th} feature of the j^{th} neighbor placed in this ε-neighborhood.Considering the proper ε-neighborhood not only does not decrease the quality of estimation but also causes a significant reduction in its computational time. Fig. 4.7 shows the overview of the proposed fault quantification system.

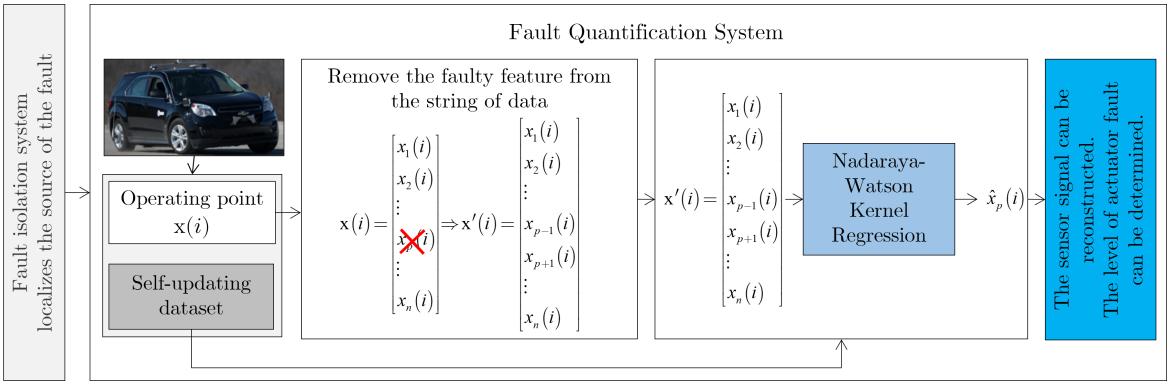

Figure 4.7: The overview of the proposed fault quantification system.

4.3 Application of The Health Monitoring System to Vehicles

To apply the proposed health monitoring system to a vehicle, its planar motion is considered. Based on this motion, the available vehicle sensors are the IMU sensor and the wheel speed sensors, and the available desired control actions are the vehicle traction torque, brake torque, and steering wheel angle. Given the vehicle's equations of motion, the desired control actions and sensor measurements along with some of their derivatives are essential to determine the vehicle's state. Therefore, based on the aforementioned points, the string of data that can express the vehicle's state is:

$$\mathbf{x} = (a_x,\, a_y,\, r,\, \dot{r},\, \omega_{fL},\, \dot{\omega}_{fL},\, \omega_{fR},\, \dot{\omega}_{fR},\, \omega_{rL},\, \dot{\omega}_{rL},\, \omega_{rR},\, \dot{\omega}_{rR},\, \delta,\, T)^\top \tag{4.11}$$

in which the descriptions of all these variables are provided in Table 4.1. In this table, subscripts $ij \in \{fL, fR, rL, rR\}$ represent the front-left, front-right rear-left, and rear-right wheels, respectively.

Table 4.1: List of the vehicle variables in the proposed string of data

Variable Notation	**Description**
δ	Steering Angle
T	Total torque applied to the wheels
a_x	Longitudinal Acceleration
a_y	Lateral Acceleration
r	Yaw Rate
$\dot{r}$	Yaw Acceleration
ω_{ij}	Wheel Angular Velocity
$\dot{\omega}_{ij}$	Wheel Angular Acceleration

As previously mentioned, the proposed health monitoring approach generally works based on the distance of the operating point from the multi-dimensional surface fitted to the operating point's neighbors. In the fault detection task, this subject is studied in the full-dimensional space, but in the fault isolation task, it is investigated in all subspaces. Based on Eq. (4.11), some features along with their derivatives are in the proposed string of data. If each of these features is absent from subspaces, its derivative should not be considered either, since if a feature is not valid due to being faulty, its derivative is not valid either. Given Eq. (4.11), there are 9 subspaces, and Table 4.2 illustrates the removed feature from the string of data in each of these subspaces.

Table 4.2: Feature(s) removed from the proposed string of data in all 9 subspaces

	Feature(s) removed from x (Eq. (4.11))
Subspace 1	a_x
Subspace 2	a_y
Subspace 3	$r, \dot{r}$
Subspace 4	$\omega_{fL}, \dot{\omega}_{fL}$
Subspace 5	$\omega_{fR}, \dot{\omega}_{fR}$
Subspace 6	$\omega_{rL}, \dot{\omega}_{rL}$
Subspace 7	$\omega_{rR}, \dot{\omega}_{rR}$
Subspace 8	δ
Subspace 9	T

Note that, in reality, due to different disturbance sources, the sensor measurements are noisy. Therefore, persistency in the health status of operating points should be considered in the health monitoring process. It means that if the distance of the operating point from the corresponding surface is persistently placed within the prediction interval, it indicates that the target system is healthy. If this distance exceeds the threshold and persistently lies outside the prediction interval, it shows that there is a fault in the target system.

4.4 Vehicle Health Monitoring System Evaluation with Experimental Results

The performance of the proposed health monitoring system is experimentally evaluated with the electric all-wheel-drive test vehicle shown in Fig. 3.2.

To generate a dataset for the algorithm evaluation and without loss of generality, the acceleration-in-turn maneuver is also considered in this study. The specifications of the acceleration-in-turn maneuvers used for collecting the dataset and test data are presented in Table 4.3. To test the proposed algorithm, the collected data are used with different sensor/actuator faults to evaluate the fault detection, isolation, and quantification systems.

Table 4.3: The specifications of the acceleration-in-turn maneuvers used for collecting the dataset and test data

	Dataset	**Test Data**
Torque applied to each wheel $T_{ij} = A \times$ Ramp Function, $T_{\max} = 500\, N.m$	$A \in \{50, 70, 80\}$	$A = 60$
Steering Wheel Angle $\delta = B \times$ Step function	$B \in \{140, 160, 200, 220\}$	$B = 180$ deg

To evaluate the performance of the proposed health monitoring system, the procedure is applied to the following cases:

1. the healthy and known test data
2. the healthy and unknown test data
3. the faulty test data (Faulty sensor)
4. the faulty test data (Faulty Actuator)

Note that since the scales of features are different from each other, they are normalized between -1 and 1. The design parameters used in the proposed health monitoring system are provided in Table 4.4. If the number of neighbors in ε-neighborhood is greater than 500, only 500 of them are randomly selected and used in the health monitoring process.

Table 4.4: The design parameters of the proposed health monitoring system

Design Parameter	**Value**
ε in the ε-neighborhood	0.1
σ in the Gaussian kernel function	0.025
k in Fig. 4.2	20
The confidence level of the prediction interval $(1 - \alpha)$	99%

4.4.1 Healthy & Known Test Data

The healthy and known test data is the test data when the vehicle operates under conditions that are approximately the same as those in its dataset and the vehicle's sensors and actuators are completely healthy. In each operating point of the healthy and known test data, it is expected that there are enough neighbors to fit a full-dimensional surface, and the distance of the operating point from the fitted surface is placed within the prediction interval. Having enough neighbors to fit a surface means that the number of neighbors is greater than a determined threshold, k. Fig. 4.8 shows the distance of the operating point from the associated surface in the healthy and known test data, and the result confirms what is expected.

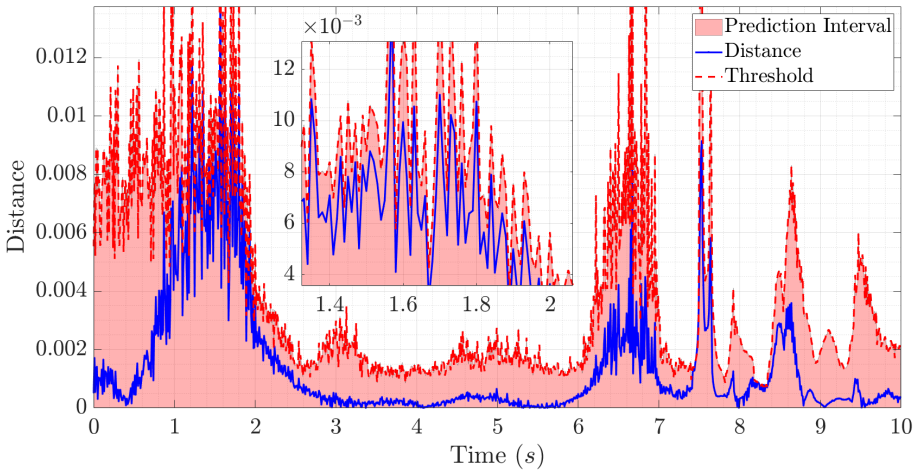

Figure 4.8: The distance of the operating point from the associated surface in the healthy and known test data.

4.4.2 Healthy & Unknown Test Data

The healthy and unknown test data is the test data when the vehicle encounters unknown environments and the vehicle's sensors and actuators are completely healthy. Since all data in the dataset are for acceleration-in-turn maneuvers, to simulate an unknown environment,

an acceleration maneuver with sinusoidal steering is considered as test data in this part. Fig. 4.9 shows the distance of the operating point from its associated surface in this maneuver. On the horizontal axis of this figure, the pink/yellow color shows when the dataset has/does not have enough neighbors around the operating point to fit the full-dimensional surface. For each operating point that has enough neighbors, the fault detection system fits a surface to its neighbors and then finds its distance from this surface. This distance and its corresponding threshold are shown with blue and dashed red lines in Fig. 4.9. For each operating point that does not have enough neighbors, the fault detection system fits a surface to its history of operating points and then finds its distance from this surface. This distance and its threshold are also shown with black and dashed green lines in Fig. 4.9. This figure confirms that the distance of the operating point from its associated surface in this test data does not exceed the threshold and is placed within the prediction interval. This point shows that all vehicle sensors and actuators are healthy. Moreover, since the dataset is not rich enough around the healthy operating points that do not have enough neighbors (missing data), these operating points are added to the self-updating dataset to update it.

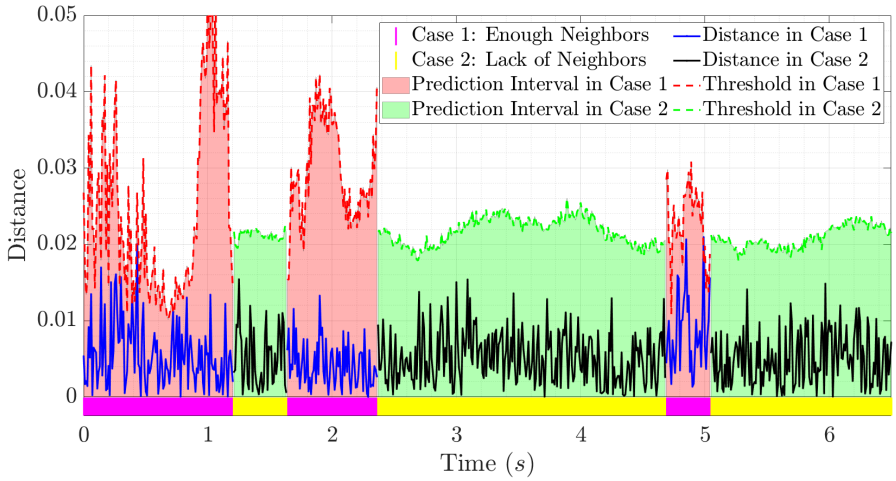

Figure 4.9: The distance of the operating point from the associated surface in the healthy and unknown test data. The pink/yellow color on the horizontal axis shows the availability/lack of data around the operating points.

4.4.3 Faulty Test Data (Faulty Sensor)

In this part, the proposed health monitoring system is applied to the test data in which one of the vehicle's sensors is faulty. To evaluate the performance of this system, the following cases are studied in this section:

- Faulty longitudinal acceleration sensor
- Faulty lateral acceleration sensor
- Faulty rear-right wheel speed sensor
- Faulty front-left wheel speed sensor

It is expected that the health monitoring system is able to detect and isolate the fault and then reconstruct the healthy value of the faulty sensor.

4.4.3.1 Faulty longitudinal acceleration sensor

In the first case, a 30% fault is applied to the vehicle's IMU sensor measuring the longitudinal acceleration starting at the 5^{th} second of the test maneuver. First, the fault detection system is applied to find when the fault occurs. Based on its results shown in Fig. 4.10, after the 5^{th} second, the distance of the operating point from the fitted surface exceeds its threshold and persistently lies outside the prediction interval. Therefore, the fault detection system notifies that a fault occurs in the vehicle. After that, the fault isolation system checks the distance of the operating point from the fitted surface in all nine subspaces. Fig. 4.11 shows that only in Subspace 1, the distance is persistently placed within the prediction interval, so the faulty feature is the one that is absent from this subspace. Based on Table 4.2, it can be found that the faulty feature is the longitudinal acceleration sensor, and this point has conformity with what is expected. Finally, all remaining healthy features (features in Subspace 1) are used to reconstruct the healthy value of the longitudinal acceleration by applying NWKR. Fig. 4.12 illustrates that the fault quantification system can appropriately reconstruct the healthy value of the longitudinal acceleration sensor.

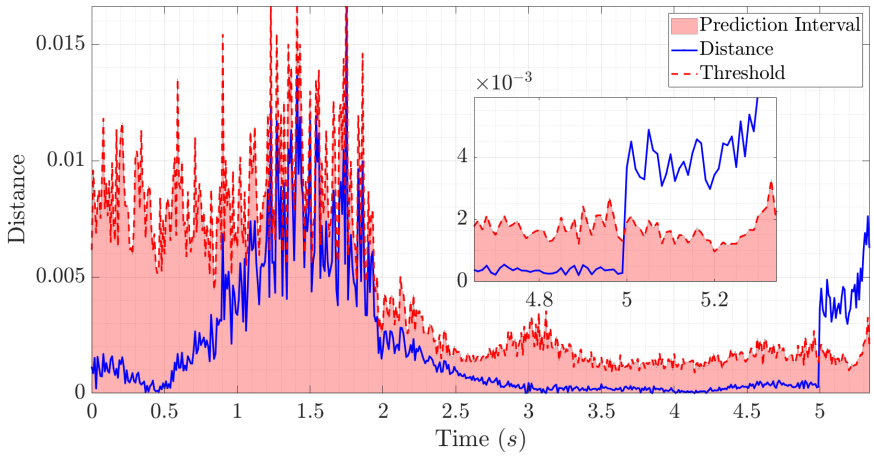

Figure 4.10: The distance of the operating point from the associated surface in the faulty test data in which the longitudinal acceleration sensor becomes faulty.

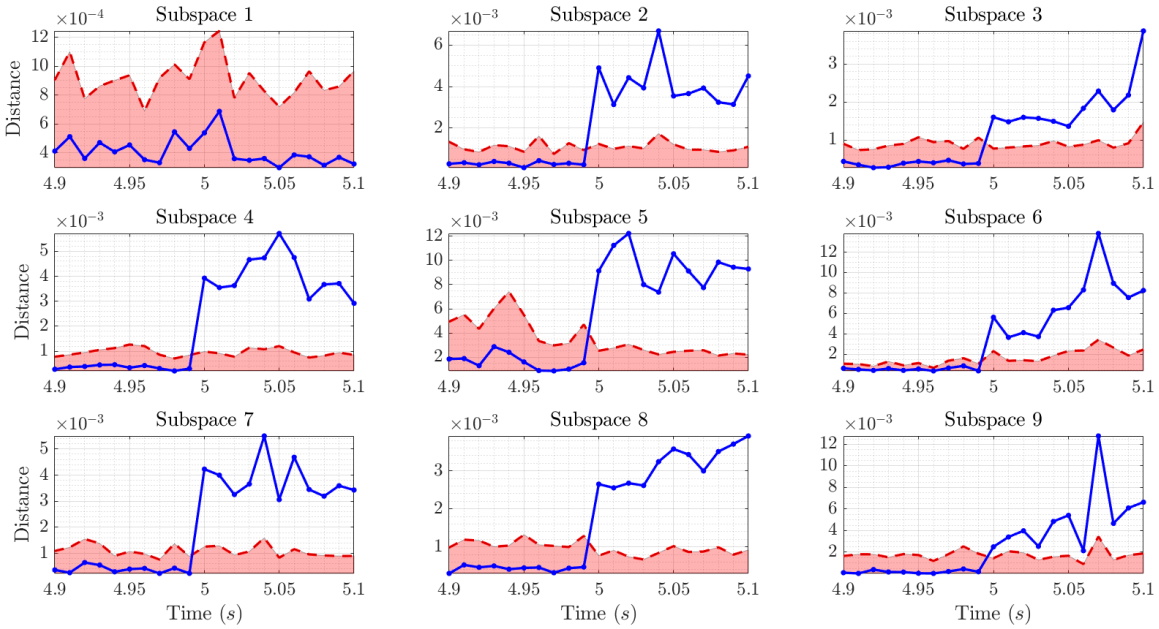

Figure 4.11: The distance of the operating point from the associated surface in all subspaces when the fault occurs in the longitudinal acceleration sensor.

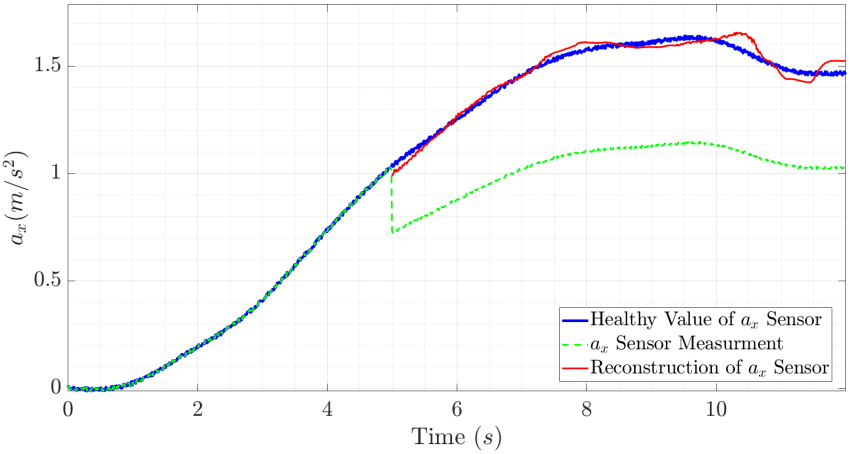

Figure 4.12: Reconstruction of the longitudinal acceleration sensor after it has become faulty.

4.4.3.2 Faulty lateral acceleration sensor

In the second case, a bias fault of $0.5m/s^2$ is applied to the lateral acceleration sensor starting at the 6^{th} second of the test maneuver. First, Fig. 4.13 shows that after the 6^{th} second, the distance of the operating point from the fitted surface persistently lies outside the prediction interval, so it indicates that there is a fault in the vehicle and the fault detection system successfully detects it. Then, based on Fig. 4.14, the fault isolation system shows that only in Subspace 2 in which the lateral acceleration sensor is removed, the distance of the projected operating point from the corresponding surface is persistently placed within the prediction interval. Therefore, it verifies that the faulty feature is the lateral acceleration sensor. Finally, Fig. 4.15 illustrates that the fault quantification system by using the all features in Subspace 2 can properly reconstruct the healthy value of the faulty sensor.

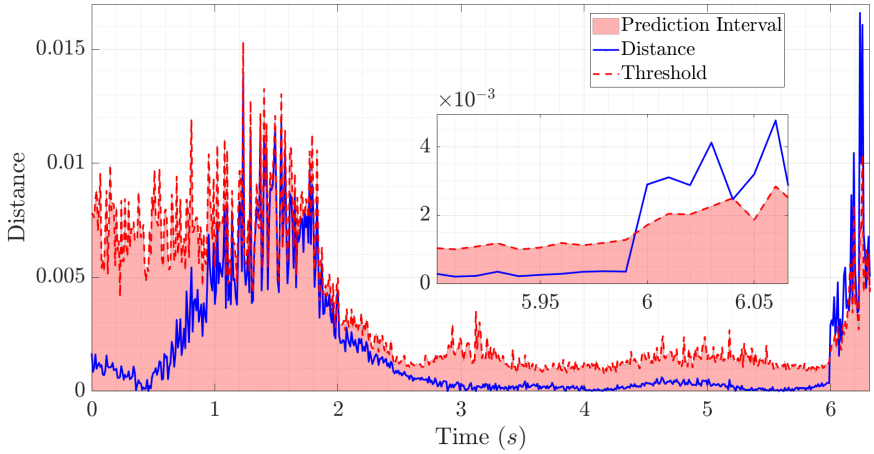

Figure 4.13: The distance of the operating point from the associated surface in the faulty test data in which the lateral acceleration sensor becomes faulty.

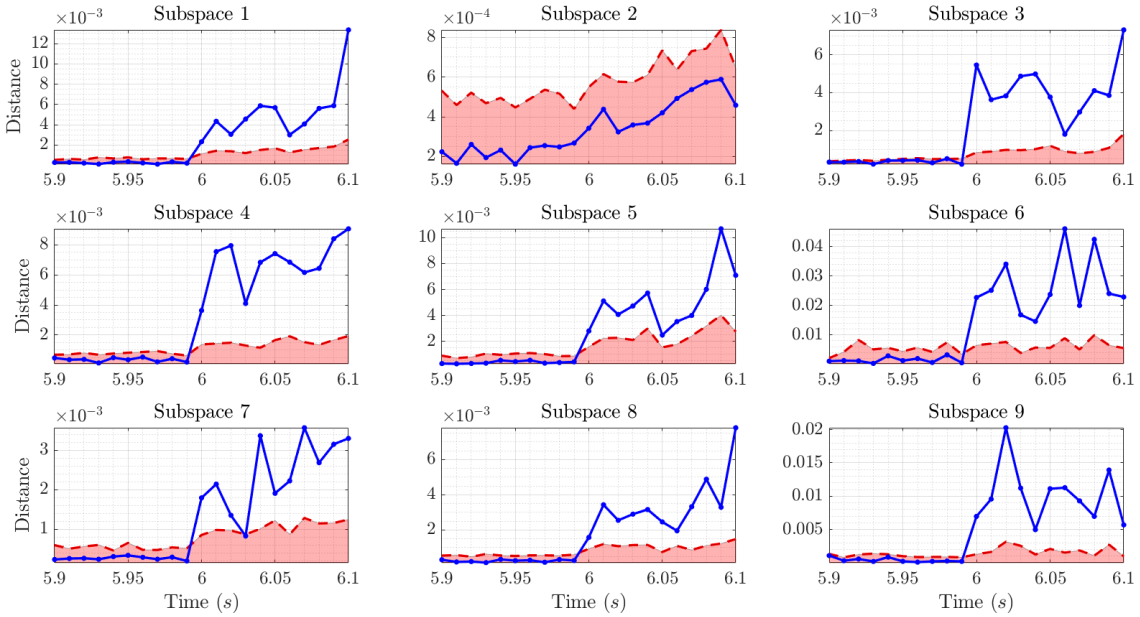

Figure 4.14: The distance of the operating point from the associated surface in all subspaces when the fault occurs in the lateral acceleration sensor.

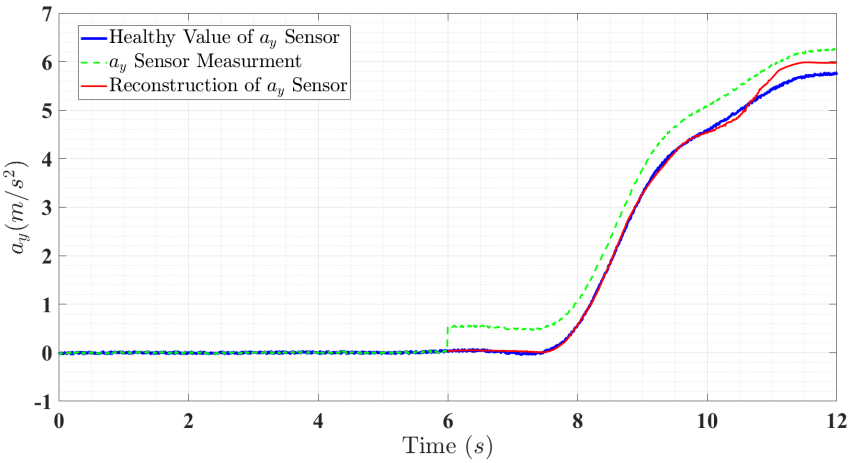

Figure 4.15: Reconstruction of the lateral acceleration sensor after it has become faulty.

4.4.3.3 Faulty rear-right wheel speed sensor

In the third example, a 30% fault is applied to the rear-left wheel speed sensor starting at the 6^{th} second of the test maneuver. Fig. 4.16 shows that after the 6^{th} second, the distance of the operating point from the fitted surface in the full-dimensional space is persistently placed outside the prediction interval, which is an indication occurring a fault in the vehicle. After fault detection, in Fig. 4.17, the fault isolation system shows that only the distance of the operating point from the fitted surface in Subspace 7 persistently lies within the prediction interval. Based on Table 4.2, Subspace 7 is the subspace from which the rear-right wheel angular velocity and its derivative are absent, so this point confirms that the faulty feature is the rear-right wheel speed sensor. By using the all features in Subspace 7, NWKR can successfully reconstruct the healthy value of the rear-right wheel angular velocity shown in Fig. 4.18.

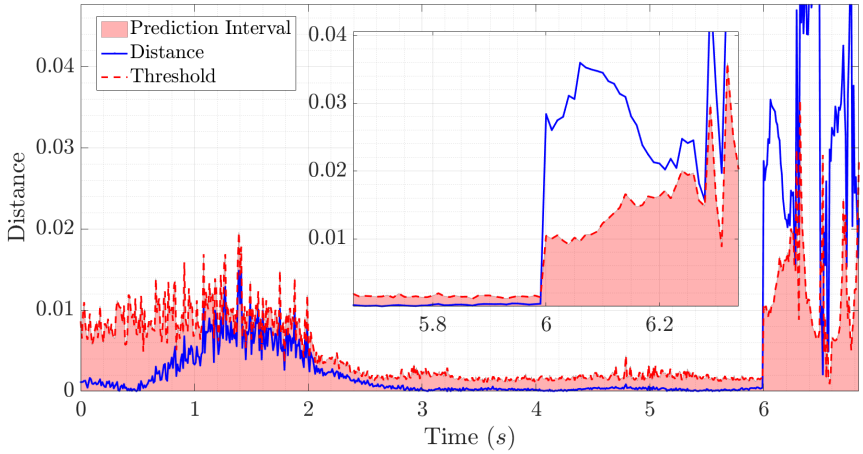

Figure 4.16: The distance of the operating point from the associated surface in the faulty test data in which the rear-right wheel speed sensor becomes faulty.

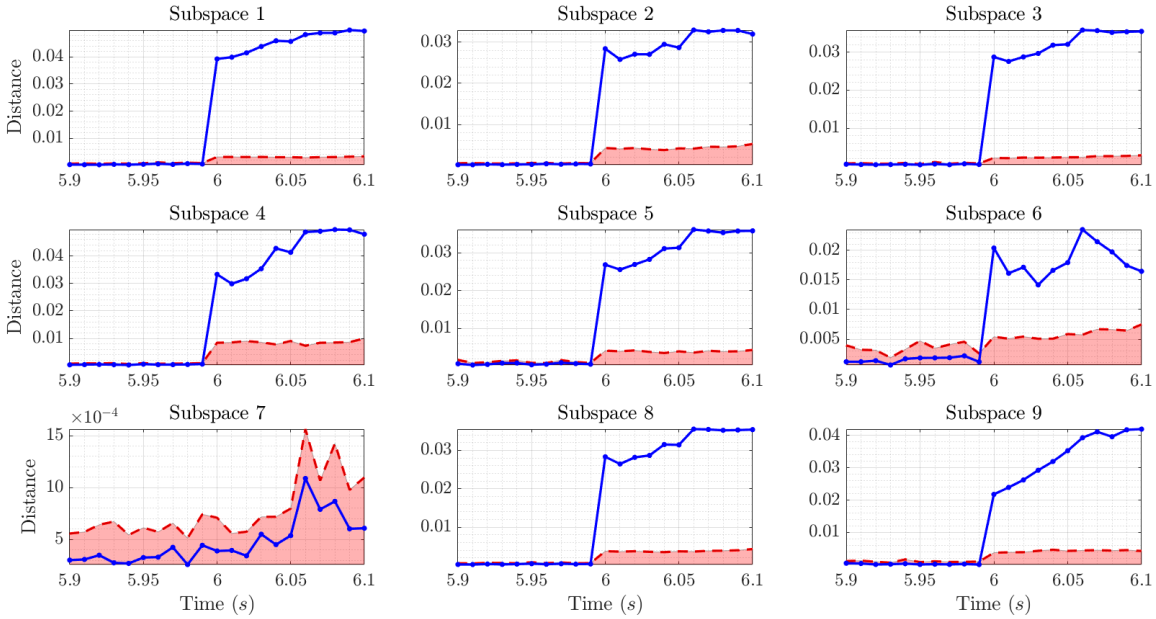

Figure 4.17: The distance of the operating point from the associated surface in all subspaces when the fault occurs in the rear-right wheel speed sensor.

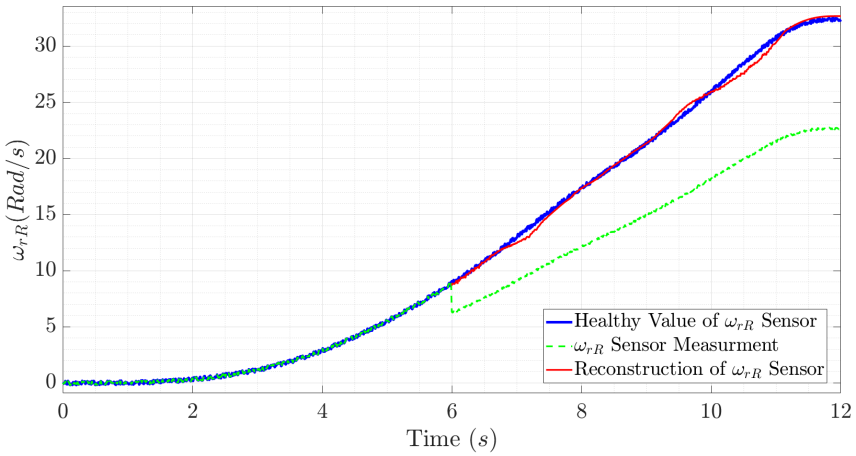

Figure 4.18: Reconstruction of the rear-right wheel speed sensor after it has become faulty.

4.4.3.4 Faulty front-left wheel speed sensor

In the last case, a gradual multiplicative fault with the pattern shown in Fig. 4.19 is applied to the front-right wheel speed sensor. Fig. 4.20 illustrates that after the 4.1^{th} second, the distance of the operating point from the fitted surface exceeds the threshold and persistently lies outside the prediction interval. Therefore, the fault detection system notifies the existence of the fault at this time when the fault's magnitude is 2% (based on Fig. 4.19). Then, based on Fig. 4.21, the fault isolation system shows that only in Subspace 4 in which the front-left wheel angular velocity and its derivative are removed, the distance of the projected operating point from the corresponding surface is persistently placed within the prediction interval. Therefore, it verifies that the faulty feature is the front-left wheel speed sensor. Lastly, Fig. 4.22 shows that the fault quantification system by using the all features in Subspace 4 can appropriately reconstruct the healthy value of the faulty sensor.

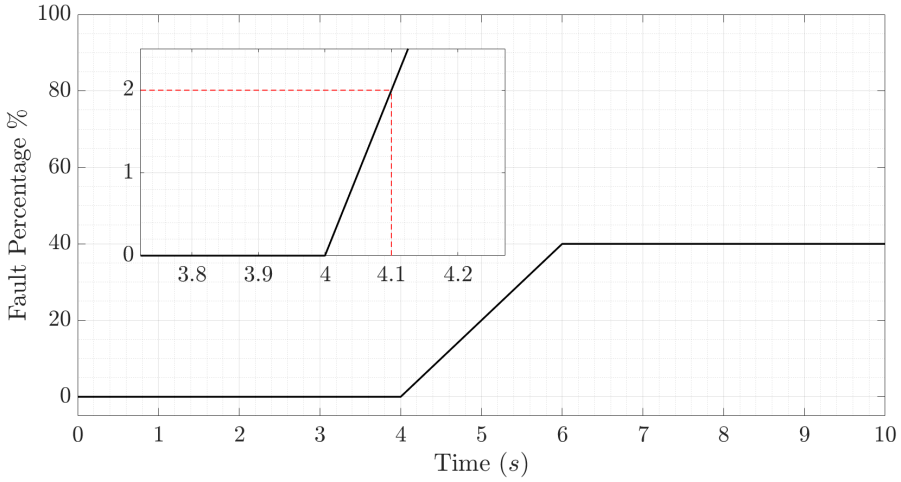

Figure 4.19: The fault pattern applied to the front-left wheel speed sensor.

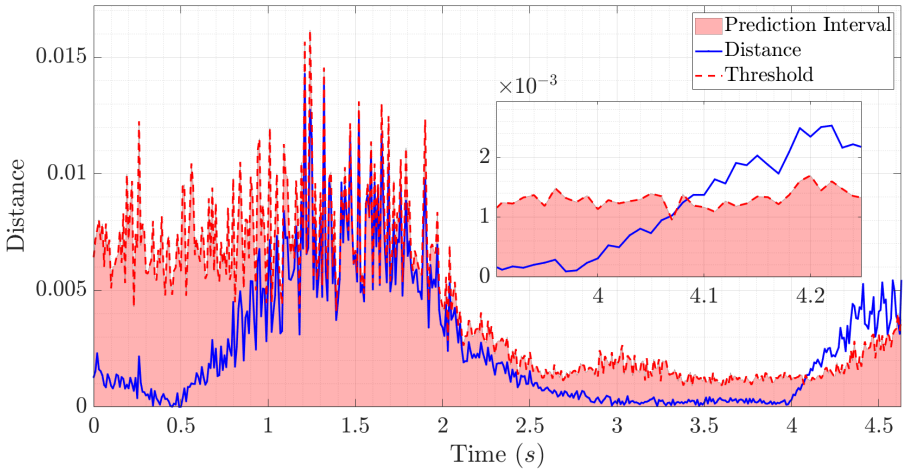

Figure 4.20: The distance of the operating point from the associated surface in the faulty test data in which the front-left wheel speed sensor becomes faulty.

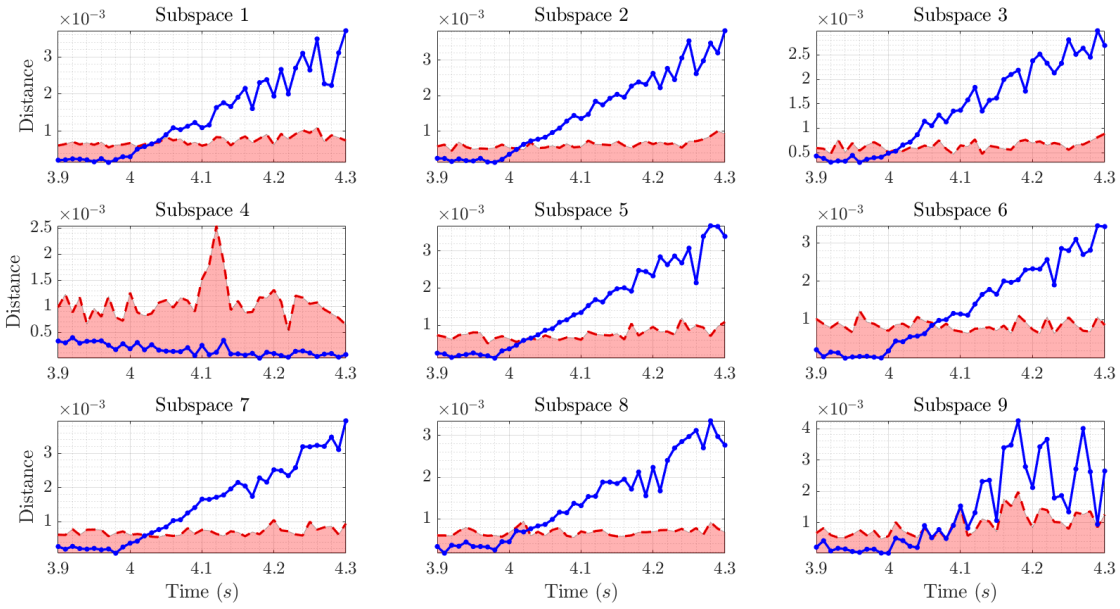

Figure 4.21: The distance of the operating point from the associated surface in all subspaces when the fault occurs in the front-left wheel speed sensor.

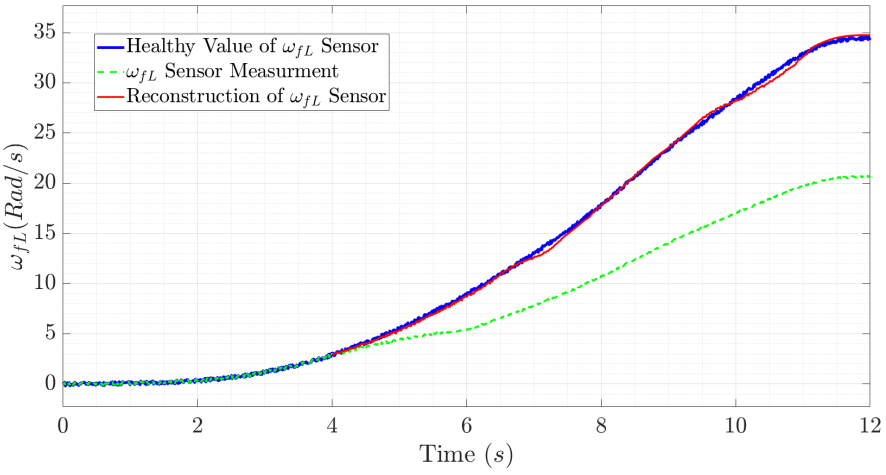

Figure 4.22: Reconstruction of the front-left wheel speed sensor after it has become faulty.

4.4.4 Faulty Test Data (Faulty Actuator)

In this section, the proposed health monitoring system is applied to the test data in which one of the vehicle's actuators is faulty. It is expected that the health monitoring system is able to detect and isolate the fault and then find the level of actuator failure. To study the performance of this system in health monitoring of the vehicle actuators, a 25% fault is applied to the vehicle traction motor or brake actuator starting at the 11^{th} second of the test maneuver. Fig. 4.23 illustrates that after the 11^{th} second, the distance of the operating point from the fitted surface exceeds its threshold and it is persistently placed outside the prediction interval. Based on this result, the fault detection system notifies occurring a fault in the vehicle. After fault detection, the distance of the operating point from the fitted surface in all nine subspaces shown in Fig. 4.24 is determined by the fault isolation system. Based on Fig. 4.24, the distance of the operating point from the fitted surface in Subspace 9 lies within the prediction interval. According to Table 4.2, Subspace 9 is the subspace in which the wheels' total torque is excluded, so the fault isolation system determines that the faulty feature is the vehicle traction motor or brake actuator. After localizing the source of the fault, NWKR uses all features in Subspace 9 to estimate the wheels' actual torque, and Fig. 4.25 shows that its actual value is successfully estimated. The sign of its estimated value can specify whether the faulty feature is the vehicle's traction motor or brake system. If the estimated value of the wheels' actual torque is positive/negative, it shows that the vehicle is in the acceleration/brake mode, so the faulty feature is the vehicle's traction motor/brake system. By comparing the desired value of the wheels' total torque (T) provided by the vehicle's controllers or driver to the estimation of its actual value obtained by the fault quantification system, the level of actuator failure can be determined. Fig. 4.25 also shows that the estimated value of the wheels' total torque (T) can be approximately fitted to 75% of its desired value, so the level of actuator failure is determined as 25%.

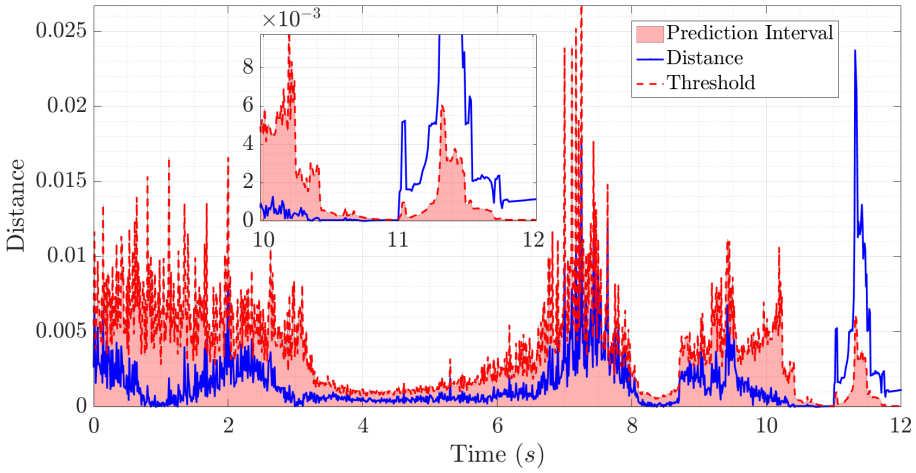

Figure 4.23: The distance of the operating point from the associated surface in the faulty test data in which the vehicle traction motor becomes faulty.

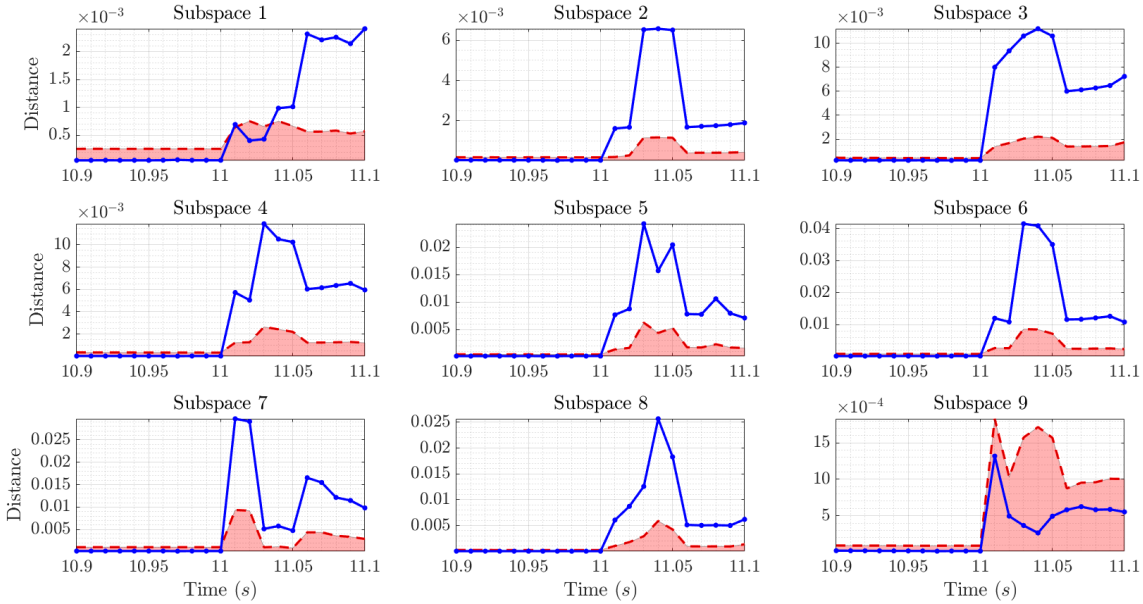

Figure 4.24: The distance of the operating point from the associated surface in all subspaces when the fault occurs in the vehicle traction motor.

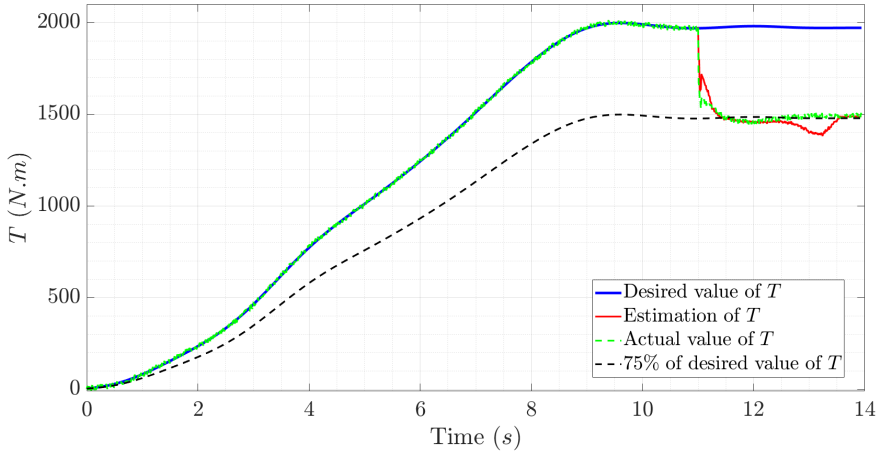

Figure 4.25: Estimation of the actual action of the traction motor after it has become faulty.

4.5 Summary

In this chapter, a general data-driven health monitoring system was developed to evaluate the health status of vehicles' sensors and actuators. This system involves three tasks: fault detection, isolation, and quantification, which are respectively responsible for determining the occurrence, location, and magnitude of faults. The developed system requires data provided by the sensors' measurements and desired control actions of the target vehicle. The vehicle's current state expressed by a string of data with n features is called the vehicle's operating point. By using the target vehicle's data around its operating point, the proposed health monitoring system checks whether the operating point belongs to the target vehicle. If the operating point does not belong to the vehicle, the health monitoring system notifies that there is a fault in the vehicle. After fault detection, to isolate the fault, the health status of the operating point is investigated on subspaces in each of which one feature is absent. In the subspace from which the faulty feature is absent, the projected operating point onto this subspace belongs to the target vehicle. Based on this idea, the faulty component was localized. If a sensor becomes faulty, the fault quantification sys-

tem reconstructs its healthy value by using the remaining healthy features. Therefore, the vehicle's controllers can use the reconstructed value instead of the faulty measurements to generate appropriate control actions. If an actuator becomes faulty, by using the remaining healthy features, the fault quantification system estimates its actual action and finds its level of failure. This information can be used to generate desired control actions by scaling the actuator output. The proposed health monitoring system can also detect abrupt faults when the vehicle encounters unknown environments or when there is a lack of data around its operating point. In these conditions, the proposed self-updating dataset also collects data directly from the vehicle to enrich the dataset, and the health monitoring system is able to use the new data when the vehicle faces similar environments again. The performance of the proposed health monitoring system is evaluated using experimental tests. The results show that the health monitoring system appropriately detects, isolates, and quantifies faults in the vehicle's sensors and actuators.

There are two issues in the developed data-driven health monitoring system. The first one is that although this health monitoring system is able to detect abrupt faults in a lack of data, it is challenging for this system to deal with gradual faults in these conditions. In [80], a model-based health monitoring system is developed for vehicle sensors, which can detect sensor faults, isolate the faulty sensor and provide the fault-tolerant estimation of vehicle states. When the vehicle faces unknown environments, this model-based health monitoring system can be used to improve the performance of the data-driven one developed in this chapter. Integrating these two methodologies provides a reliable hybrid health monitoring system that can outperform the model-based and data-driven ones used individually. The next issue is that for each vehicle, the data-driven health system needs an individual dataset. The next chapter is provided to address this issue.

Chapter 5

Universal Health Monitoring System for IMU Sensor

5.1 Introduction

In the previous chapter, a data-driven health monitoring system is developed to monitor the health status of a set of sensors and actuators in a vehicle. The main requirement of the proposed algorithm is that it needs an individual dataset for each vehicle. This chapter attempts to investigate whether it is possible to provide a universal health monitoring system for vehicles. Universality in the health monitoring application means that the health monitoring system is able to monitor the health status of a vehicle using other vehicles' data.

5.2 Universal Health Monitoring System in The Vehicle Planar Motion

In universality, the vehicle inertial and geometrical parameters must be combined with vehicle variables so that the resulting features are invariant from one vehicle to another.

To develop a universal health monitoring system, it is required to propose a string of data with these features, and then apply the proposed data-driven health monitoring system operating based on this data string. To this end, the planar double-track vehicle model shown in Fig. 5.1 is considered. The model parameters with their descriptions are listed in Table 5.1.

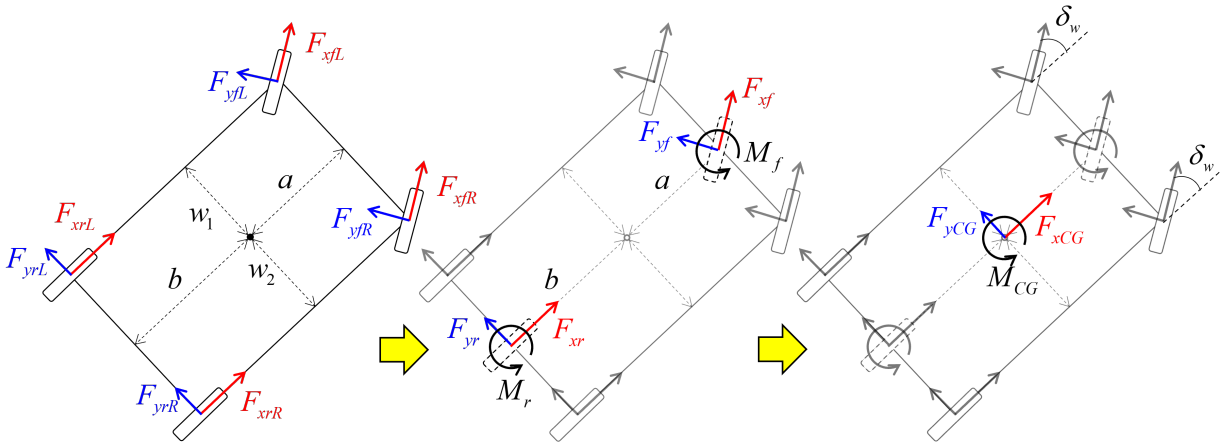

Figure 5.1: a planar double-track vehicle model

Table 5.1: The parameters of the double-track vehicle model with their descriptions

Parameter	**Description**
M	Vehicle mass
I_z	Vehicle moment of inertia about normal direction
a, b	Distance from the vehicle CG to front/rear axle
w_1, w_2	Distance from the vehicle CG to left/right wheels
R_w	Tire effective radius
I_w	Wheel moment of inertia about its axis
SR	Ratio between steering wheel angle & front wheel angle

Based on Fig. 5.1, an external force applied to each wheel can be projected onto the wheel's longitudinal and lateral axes, and these forces are respectively denoted by F_{xij} and

F_{yij}, $ij \in \{fL, fR, rL, rR\}$. According to Fig. 5.1, the equivalent external longitudinal force, lateral force, and moment applied to an axle are respectively denoted by F_{xi}, F_{yi}, and M_i. The subscripts $i \in \{f, r\}$ represent the front and rear axles, respectively. These forces and moments for the front and rear axles are as follows:

$$\begin{aligned} F_{xf} &= F_{xfL} + F_{xfR} \\ F_{yf} &= F_{yfL} + F_{yfR} \\ M_f &= F_{xfR} w_2 \cos\delta_w - F_{yfR} w_2 \sin\delta_w - F_{xfL} w_1 \cos\delta_w + F_{yfL} w_1 \sin\delta_w \end{aligned} \tag{5.1}$$

$$\begin{aligned} F_{xr} &= F_{xrL} + F_{xrR} \\ F_{yr} &= F_{yrL} + F_{yrR} \\ M_r &= F_{xrR} w_2 - F_{xrL} w_1 \end{aligned} \tag{5.2}$$

where δ_w is the front wheel steering angle. By transferring the axles' forces to the vehicle CG with their corresponding moments. The governing equations of motion at the CG are determined as

$$\begin{aligned} F_{xCG} &= F_{xf}\cos\delta_w - F_{yf}\sin\delta_w + F_{xr} \Rightarrow F_{xCG} = M a_x \\ F_{yCG} &= F_{xf}\sin\delta_w + F_{yr}\cos\delta_w + F_{yr} \Rightarrow F_{yCG} = M a_y \\ M_{CG} &= M_f + M_r + (F_{xf}\sin\delta_w + F_{yr}\cos\delta_w)\, a - F_{yr} b \Rightarrow M_{CG} = I_z \dot{r} \end{aligned} \tag{5.3}$$

where F_{xCG}, F_{yCG}, and M_{CG} are respectively the total longitudinal force, lateral force, and moment applied to the CG. The variables a_x, a_y, $\dot{r}$ are respectively the vehicle longitudinal, lateral, and yaw accelerations. Due to the uncertainties and unmodeled dynamics, the equality in Eq. (5.3) is not always valid, so it may be not possible to detect a fault in the vehicle by exploring inconsistency in these equations. However, these equations provide an opportunity to find how the vehicles' parameters should participate in the health monitoring process to achieve universality. Based on Eq. (5.3), the following string of data is proposed.

$$\mathbf{x} = \begin{pmatrix} a_x & a_y & \dot{r} & \dfrac{F_{xCG}}{M} & \dfrac{F_{yCG}}{M} & \dfrac{M_{CG}}{I_z} \end{pmatrix}^\top \tag{5.4}$$

The first three features in the proposed string of data can be obtained by the IMU sensor, and the last three features can be determined if the longitudinal and lateral forces of each wheel are available. These forces can be measured by wheel force/moment sensors,

but these sensors may not be available in commercial vehicles. Hence, reliable estimation approaches in the literature can be utilized to obtain them. The longitudinal force of each wheel can be estimated independently of the IMU sensor measurements by using the wheel dynamics as follows:

$$\hat{F}_{xij} = \frac{T_{ij} - I_w \dot{\omega}_{ij}}{R_w} \tag{5.5}$$

In the literature, several approaches are developed to estimate the tire lateral force [80–84]. By applying the developed data-driven health monitoring system to the proposed string of data (Eq. (5.4)), the universal health monitoring system is provided. The universal health monitoring system can monitor the health status of the target vehicle's IMU sensor (measuring the longitudinal/lateral accelerations and yaw rate) by using the data of other vehicles. Despite some limitations in the isolation and reconstruction tasks, the universal health monitoring system is capable of the following:

- **Longitudinal Acceleration Sensor:** The universal health monitoring system can detect and isolate faults in the target vehicle's longitudinal sensor and also reconstruct its healthy value by using other vehicles' data.
- **Lateral Acceleration Sensor:** The proposed universal health monitoring system is able to only detect faults in the target vehicle's lateral sensor. The reason behind it is that approaches developed in the literature to estimate tire lateral force require lateral acceleration as their input [80–84]. Therefore, if the lateral acceleration sensor becomes faulty, the three last features in the proposed string of data are not valid anymore. Based on the proposed data-driven health monitoring approach, to isolate the fault in the lateral acceleration sensor, the lateral acceleration feature and all features affected by it should be removed from the string of data. On this condition, the remaining features are longitudinal and yaw accelerations which are insufficient to represent a vehicle model. Therefore, it is not possible to check the vehicle health status in this subspace and hence isolate the fault in the lateral acceleration sensor.
- **Yaw Rate Sensor:** The universal health monitoring system can detect and isolate non-bias faults in the yaw rate sensor. Since the yaw acceleration feature is in the proposed string of data and a bias fault in the yaw rate sensor does not affect its

derivative, it is not possible to detect a bias fault in the yaw rate sensor. Moreover, after detecting and isolating non-bias faults in the yaw rate sensor, the health monitoring system based on the proposed string of data is able to reconstruct the healthy value of the yaw acceleration signal not the yaw rate sensor.

Since it is only possible to isolate faults when the longitudinal acceleration sensor or yaw rate sensor becomes faulty, for fault isolation, only two subspaces are considered. Table 5.2 illustrates the available and removed features in each of these subspaces.

Table 5.2: The available and removed features in subspaces

	Removed Feature	**Available Features**
Subspace 1	a_x	$\left(a_y,\ \dot{r},\ \frac{F_{xCG}}{M},\ \frac{F_{yCG}}{M},\ \frac{M_{CG}}{I_z}\right)$
Subspace 2	$\dot{r}$	$\left(a_x,\ a_y,\ \frac{F_{xCG}}{M},\ \frac{F_{yCG}}{M},\ \frac{M_{CG}}{I_z}\right)$

5.3 Universal Health Monitoring System Evaluation with Simulation Results

The performance of the universal health monitoring system is evaluated by software simulations with high-fidelity CarSim models.

5.3.1 Training and Test Vehicles/Dataset

Six vehicle models in the CarSim software are considered and their specifications are listed in Table 5.3. To investigate the effect of universality, one of the vehicles is considered a test vehicle, and others are considered training vehicles.

Table 5.3: The specifications of the six CarSim vehicle models

	Test Vehicle	**Training Vehicles**				
Vehicle Parameters	D Class Sedan	E Class SUV	E Class Sedan	C Class Hatchback	B Class Hatchback	Minivan
$M(Kg)$	1525	3397	1839	1675	1236	1936
$I_z(Kgm^2)$	2315	2687	3234	1413	1343	3528
$a(m)$	1.1	1.18	1.4	1.015	1.04	1.35
$b(m)$	$1,76$	1.77	1.65	1.895	1.56	1.65
$w_1(m)$	0.775	0.788	0.8	0.838	0.740	0.82
$w_2(m)$	0.775	0.788	0.8	0.838	0.740	0.82
$R_w(m)$	0.325	0.393	0.353	0.325	0.298	0.346
$I_w(Kgm^2)$	1.5	2.8	2	1.5	1	2
SR	17.41 : 1	20.05 : 1	17.61 : 1	17.97 : 1	17 : 1	17 : 1

The acceleration-in-turn maneuver comprised of steering and pushing the accelerator pedal is considered in this study. Note that the position of the accelerator pedal can be represented by the percentage of the throttle opening. Therefore, to generate a dataset, the simulation runs for the training vehicles with different steering wheel angles, different percentages of throttle opening, and different road frictions. Test data is obtained by running another acceleration-in-turn maneuver for the test vehicle. The specifications of the acceleration-in-turn maneuvers used for collecting the dataset and test data are presented in Table 5.4.

Table 5.4: The specifications of the acceleration-in-turn maneuvers used for collecting the dataset and test data

	Dataset	**Test Data**
Throttle Opening (THRL) %	THRL $\in \{0\%, 10\%, ..., 100\%\}$	THRL $= 65\%$
$\delta = B \times$ Step function (deg)	$B \in \{-270, -225, ..., 225, 270\}$	$B = 100$
Road Friction μ	$\mu \in \{0.5, 0.75, 0.85, 1\}$	$\mu = 0.8$

5.3.2 Simulation Results

In this section, the universal health monitoring system is utilized on the test data from the test vehicle using the dataset collected from training vehicles. The following cases are studied to evaluate the performance of this system:

- Faulty longitudinal acceleration sensor
- Faulty Yaw rate sensor
- Faulty lateral acceleration sensor

5.3.2.1 Faulty longitudinal acceleration sensor

In the first case, a 50% fault is applied to the vehicle's IMU sensor measuring the yaw rate starting at the 3.5^{th} second of the test maneuver. First, the fault detection system is applied to find when the fault occurs. Fig. 5.2 shows that after the 3.5^{th} second, the distance of the operating point from the fitted surface exceeds its threshold and persistently lies outside the prediction interval. Hence, the fault detection system notifies that a fault occurs in the vehicle. After that, the fault isolation system checks the distance of the projected operating point from the fitted surface in two subspaces. Fig. 5.3 shows that in Subspace 1, the distance is persistently placed within the prediction interval, so the faulty feature is the longitudinal acceleration sensor that is absent from this subspace. Finally, the healthy features in Subspace 1 are used to reconstruct the healthy value of the longitudinal acceleration shown in Fig. 5.4. The simulation results show that the universal health monitoring system can successfully detect and isolate the fault in the a_x sensor of the test vehicle and this system can also appropriately reconstruct the healthy value of a_x although the other vehicles' data are used.

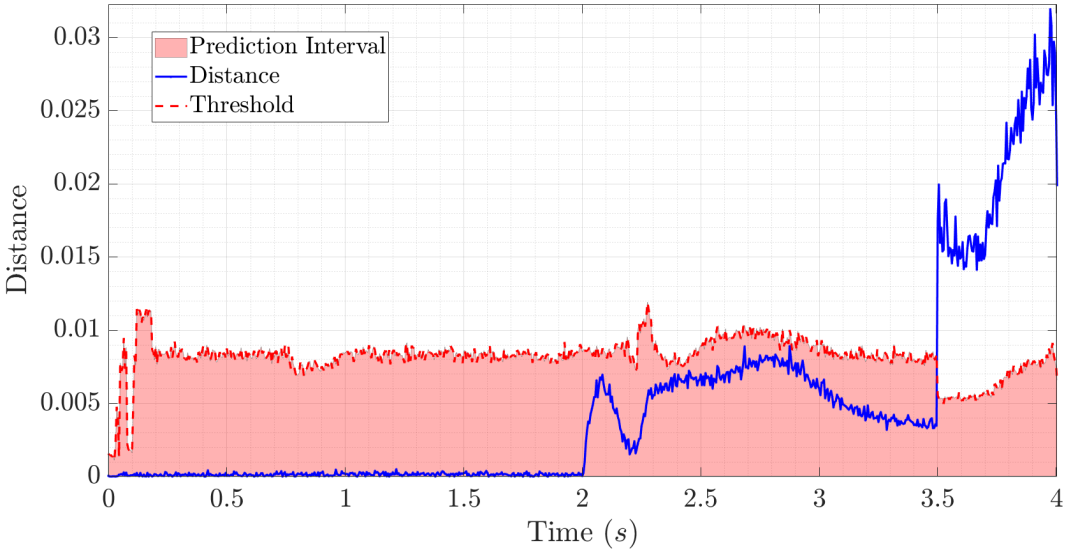

Figure 5.2: The distance of the operating point from the associated surface in the faulty test data in which the longitudinal acceleration sensor becomes faulty.

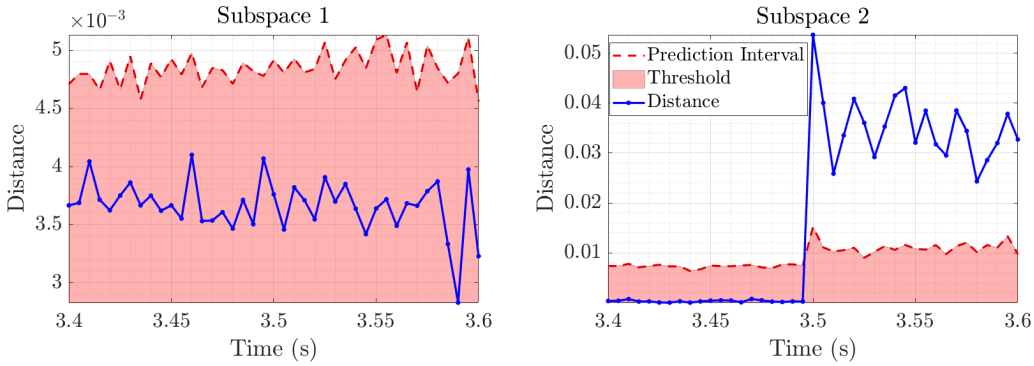

Figure 5.3: The distance of the operating point from the associated surface in two determined subspaces when the fault occurs in the longitudinal acceleration sensor.

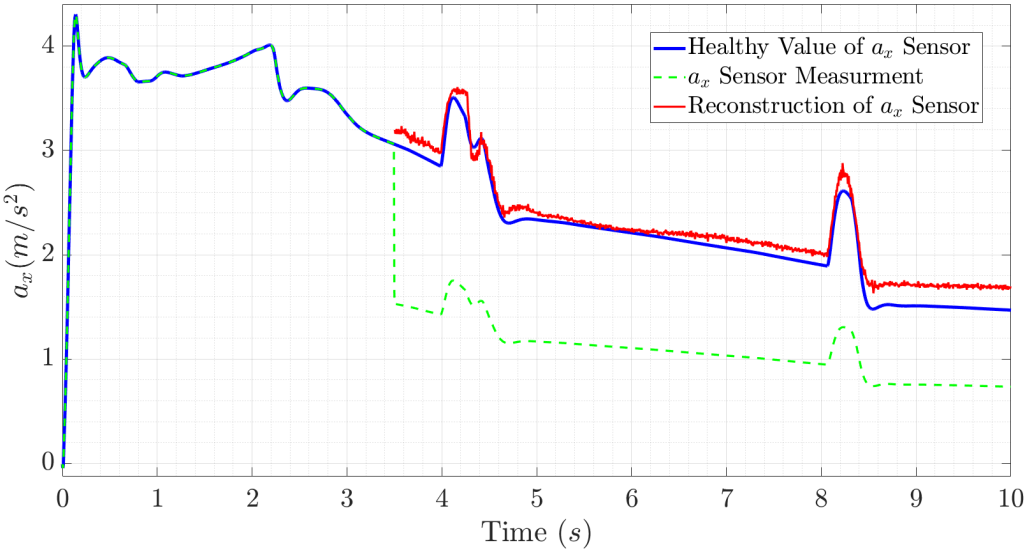

Figure 5.4: Reconstruction of the longitudinal acceleration sensor after it has become faulty by using the other vehicles' data.

5.3.2.2 Faulty yaw rate sensor

In the second case, a 60% fault is applied to the yaw rate sensor starting at the 4^{th} second of the test maneuver. When the yaw rate sensor becomes faulty, the yaw acceleration is also 40% of its healthy value. Fig. 5.5 shows that after the 4^{th} second, the distance of the operating point from the fitted surface in the full-dimensional space is persistently placed outside the prediction interval, so it indicates that there is a fault in the system. After fault detection, in Fig. 5.6, the fault isolation system shows that the distance of the operating point from the fitted surface in Subspace 2 persistently lies within the prediction interval. Therefore, the faulty feature is the yaw acceleration that is absent from this subspace. Since yaw acceleration is obtained by using the measurements of the yaw rate sensor, the actual faulty component is the yaw rate sensor. By using all features in Subspace 2, NWKR can reconstruct the healthy value of the yaw acceleration, but it is not able to reconstruct the healthy value of the yaw rate sensor.

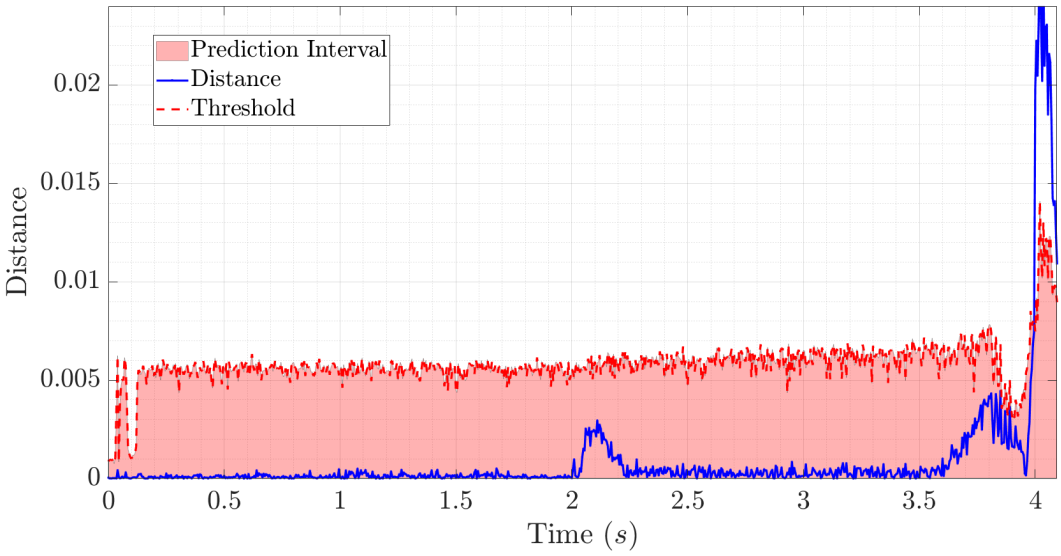

Figure 5.5: The distance of the operating point from the associated surface in the faulty test data in which the yaw rate sensor becomes faulty.

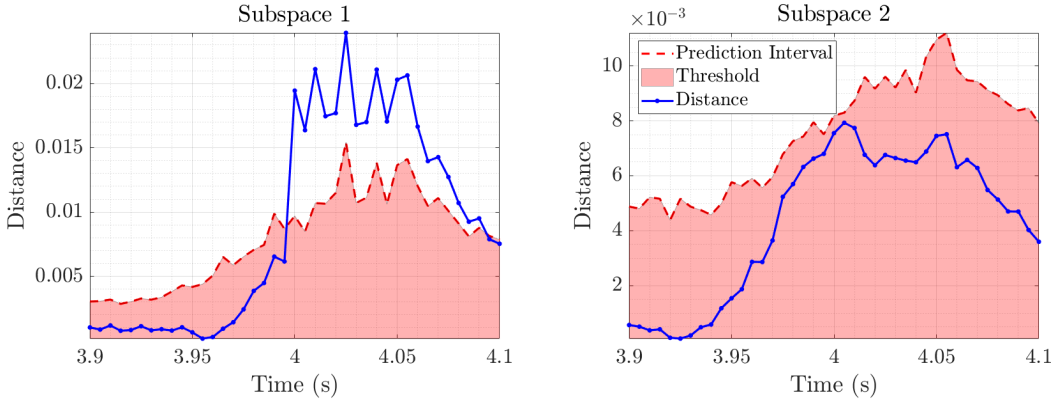

Figure 5.6: The distance of the operating point from the associated surface in two determined subspaces when the fault occurs in the yaw rate sensor.

5.3.2.3 Faulty lateral acceleration sensor

In the last case, a 20% fault is applied to the lateral acceleration sensor starting at the 5^{th} second of the test maneuver. Based on points mentioned in Section 5.2, by using the other vehicles' data, it is expected that the universal health monitoring system is able only to notify the occurrence of a fault in the test vehicle when the lateral acceleration sensor becomes faulty. Fig. 5.7 illustrates that after the 5^{th} second, the distance of the operating point from the fitted surface persistently lies outside the prediction interval, so it is an indication of occurring a fault in the test vehicle. Based on this result, the expectation has been successfully met.

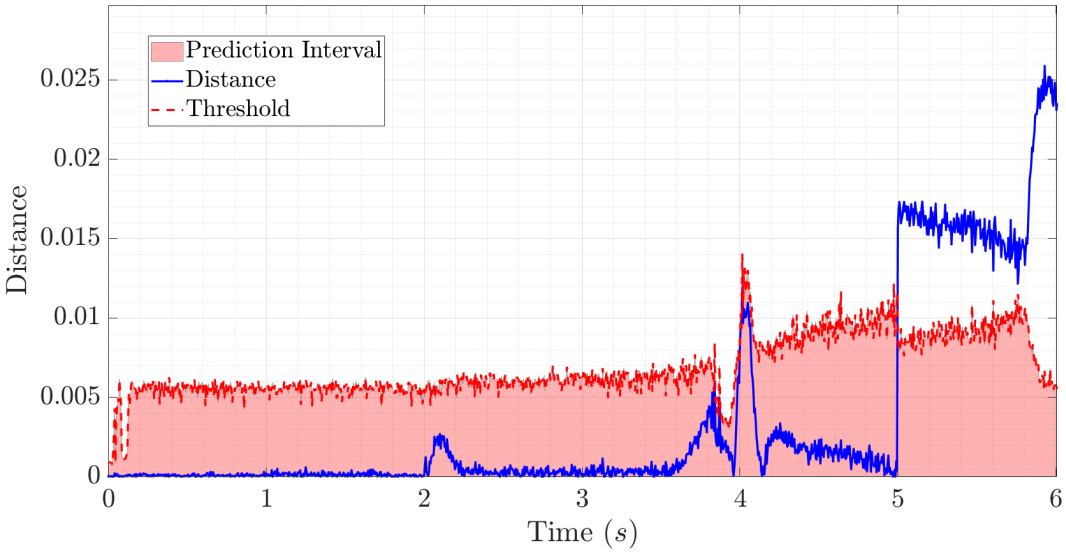

Figure 5.7: The distance of the operating point from the associated surface in the faulty test data in which the lateral acceleration sensor becomes faulty.

5.3.3 Summary

This chapter attempts to develop a universal data-driven health monitoring system for vehicles. Universality in the vehicle health monitoring application means monitoring the

health status of a vehicle by using other vehicles' data. To develop a universal health monitoring system, it is required to define a universal string of data including vehicle parameters along with vehicle variables. The governing equations of motion at the vehicle CG shows how vehicle parameters can combine with vehicle variables to provide a universal string of data. Based on the proposed string of data, universality with some restrictions is obtained for the vehicle IMU sensor. To develop a more comprehensive framework, further study is required. The performance of the proposed universal health monitoring system is evaluated by CarSim simulations. These simulations are carried out on six vehicle high-fidelity models. One of these vehicles is considered a test vehicle to provide the test data and the others are considered training vehicles to build the universal dataset. The simulation results show by using the training vehicles' data, the universal health monitoring system is able to:

- detect and isolate a fault in the test vehicle's longitudinal acceleration sensor and reconstruct its healthy value.
- detect and isolate a non-bias fault in the test vehicle's yaw rate sensor.
- detect a fault in the test vehicle's lateral acceleration sensor.

Chapter 6

Conclusions & Future Work

6.1 Conclusions

The main objective of this thesis was the development of a dependable health monitoring system for vehicles' sensors and actuators. Health monitoring is generally comprised of three tasks: fault detection, isolation, and quantification, which respectively determine the occurrence, location, and magnitude of faults.

For cases where the target sensor or actuator is known in advance, a hybrid model/data fault detection and diagnosis methodology was developed. This fault detection system worked based on residuals generated by comparing the measurements of a target sensor or control inputs of a target actuator with their estimations. These estimations were provided by a hybrid estimator developed based on the integration of model-based and data-driven estimators to take advantage of their strengths and tackle their weaknesses. The proposed self-updating dataset showed that collecting data directly from the vehicle when it confronts new environments could enrich the dataset. The data-driven estimator can use the new data for prediction if the vehicle confronts these environments again without any pre-training, thanks to Nadaraya-Watson kernel regression (NWKR) as an instant-based learning method. The Gaussian nature of the kernel function used in NWKR highlighted the remarkable role of each point's neighbors in the estimation of its output.

It was shown that considering an ε-neighborhood around each point and using only data in this neighborhood for prediction could drastically reduce the computational time of estimation while maintaining estimation quality. The proposed data management system can significantly reduce the size of the dataset and hence computational time by removing redundant points that are highly clustered. The performance of this fault detection system was evaluated by applying it to the vehicle's lateral acceleration sensor and traction motor. The results of the experimental tests conducted on the all-wheel-drive test vehicle confirmed that the hybrid estimator can outperform the model-based and data-driven estimators used individually. These results also validated that the proposed fault detection and diagnosis system can successfully detect the faults in the target sensor or actuator, then reconstruct the healthy value of the faulty sensor or find the failure level of the faulty actuator.

For cases where it is required to monitor the health status of a set of sensors and actuators rather than an individual component, in addition to fault detection, the fault isolation task comes into play. Therefore, a general data-driven health monitoring system was developed to detect, isolate, and quantify faults in a set of vehicle sensors and actuators. The fault detection algorithm works based on exploring the coherency among the target vehicle's variables, and it was shown that when a fault occurs in the system, the coherency is violated. The fault isolation approach works based on identifying a subset of the vehicle variables that are coherent with each other. It was argued that the sensor/actuator corresponding to the absent variable(s) from this subset is faulty. For cases where the faulty component was a sensor, it was demonstrated that the fault quantification algorithm can reconstruct the healthy value of the faulty sensor by using the remaining healthy variables, so the vehicle's controllers can use the reconstructed value instead of the faulty measurements to generate appropriate control actions. For cases where the faulty component was an actuator, it was shown that the fault quantification approach can estimate the actual actions of the faulty actuator and determine its level of failure. This information can be used to generate desired control actions by scaling the actuator output. It was discussed that the proposed health monitoring system can detect abrupt faults in confronting new environments. In these environments, the proposed self-updating dataset can collect data directly from the vehicle to enrich the dataset. The performance of the proposed health monitoring system was evaluated using experimental tests. The results

showed that the health monitoring system can appropriately detect, isolate, and quantify faults in the vehicle's sensors and actuators.

One of the main concerns about the developed data-driven health monitoring system is that each vehicle needs its own dataset. To relax this requirement, a universal health monitoring system was proposed to monitor the health status of the target vehicle by using the other vehicles' data. In this universal system, new features provided by the combination of vehicle parameters and variables were involved in the health monitoring process. The performance of the universal health monitoring system was evaluated by CarSim simulations. The simulation results showed that the universal health monitoring system for an IMU sensor could be achieved with some limitations.

6.2 Future Work

As potential future works, a few suggestions are mentioned in this section to enhance the accuracy of the developed health monitoring system and amplify its capabilities:

- **Extend the developed data-driven health monitoring system to other vehicle sensors and actuators:** In this research, the developed data-driven health monitoring system was applied to monitor the health status of a set of sensors (wheel speed sensors and IMU sensor measuring longitudinal/lateral accelerations and yaw rate) and actuators (steering actuator and traction motor) that plays a role in vehicle planar (yaw) dynamics and appears in its governing equations of motion. However, the application of the proposed health monitoring system can be extended to vehicle variables participating in vehicle pitch and roll dynamics. These variables including height sensors and the IMU sensor measuring vertical acceleration and pitch/roll rates can form a set of features that is representative of vehicle pitch and roll motions. Therefore, the developed data-driven health monitoring can be applied to monitor the health status of these components. As previously mentioned, the application of the proposed system can exceed the automotive systems and it can be applied to all dynamic systems. For any deterministic dynamic system, if a set of variables that

represents this system is available, the proposed health monitoring methodology can be applied to detect, isolate, and quantify faults in this system.

- **Enhance the performance of the developed data-driven health monitoring system in data scarcity by using the model-based approaches:** The developed data-driven health monitoring system is capable of monitoring abrupt faults in data scarcity. However, dealing with gradual faults is challenging for this system in a lack of data. In [80], a comprehensive model-based health monitoring methodology is proposed for vehicle sensors. This approach is able to detect sensors' faults, isolate the faulty sensor and provide fault-tolerant estimation of the faulty sensor. By extending this approach for monitoring the health status of vehicle actuators, it is possible to use this approach when the vehicle faces unknown environments. Integrating these two methodologies provides a reliable hybrid model/data health monitoring system for a set of vehicle sensors and actuators. As previously mentioned, the advantage of data-driven health monitoring approaches is no need to deal with model uncertainties and unmodeled dynamics, so they can provide accurate performance in the presence of sufficient data. The advantage of model-based approaches is their ability in confronting new environments and faults. Therefore, by taking the advantage of the model-based and data-driven health monitoring systems, the hybrid system can outperform each of them used individually.

- **Extend the developed data-driven health monitoring system for multiple-fault conditions:** In this research, it is assumed that at most one sensor or actuator becomes faulty at a time. This assumption is reasonable since it is unlikely that multiple faults occur simultaneously. Given this assumption, to isolate the fault, the proposed health monitoring system checks the coherency in subsets of the vehicle variables in which only one feature is omitted. The faulty feature is the feature that is absent from the subset in which the coherency exists. When multiple faults simultaneously occur in the target vehicle, the fault detection process is exactly the same as when a mono fault occurs. However, to isolate faulty features, the fault isolation system should explore all subsets of the vehicle variables in which more than one feature is omitted and find the subset in which the vehicle variables are

coherent with each other. Faulty features are the ones removed from this subset. This idea can be utilized for fault isolation as far as the features in each subspace can be representative of the target vehicle and can satisfy observability conditions. As long as these conditions are satisfied, by using the remaining healthy features, the fault quantification system can also reconstruct the healthy value of faulty sensors and find the failure level of faulty actuators. In future studies, it is possible to investigate what combination of simultaneous faults can be monitored.

References

[1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches," *IEEE transactions on industrial electronics*, vol. 62, no. 6, pp. 3757–3767, 2015.

[2] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–24, 2018.

[3] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[4] J. Zhou, D. Zhang, C. Ooi, M. Luo, S. Mao, and D. Wang, "Comparative study of data driven and model based approaches of rotary machines fault detection and diagnosis," *SIMTech Tech. Rep.*, vol. 11, no. 4, pp. 195–201, 2010.

[5] N. Mehranbod, M. Soroush, and C. Panjapornpon, "A method of sensor fault detection and identification," *Journal of Process Control*, vol. 15, no. 3, pp. 321–339, 2005.

[6] R. Hallouzi, V. Verdult, R. BabuŜka, and M. Verhaegen, "Fault detection and identification of actuator faults using linear parameter varying models," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 119–124, 2005.

[7] A. Abid, M. T. Khan, and J. Iqbal, "A review on fault detection and diagnosis techniques: basics and beyond," *Artificial Intelligence Review*, vol. 54, pp. 3639–3664, 2021.

[8] S.-K. Chen, A. Khajepour, W. Melek, R. Zarringhalam, and E. Hashemi, "Detection and reconstruction of suspension height sensor faults," Oct. 19 2017, uS Patent App. 15/097,570.

[9] S.-K. Chen, B. B. Litkouhi, A. Khajepour, R. Zarringhalam, and W. Melek, "Detection and reconstruction of pitch rate sensor fault," Jun. 19 2018, uS Patent 10,000,215.

[10] S.-K. Chen, N. K. Moshchuk, B. B. Litkouhi, R. Zarringhalam, A. Khajepour, and W. Melek, "Detection and reconstruction of sensor faults," Feb. 12 2019, uS Patent 10,204,461.

[11] Y. Wilhelm, P. Reimann, W. Gauchel, and B. Mitschang, "Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models," *Procedia Cirp*, vol. 99, pp. 278–283, 2021.

[12] 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 & chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[13] H. Chen and B. Jiang, "A review of fault detection and diagnosis for the traction system in high-speed trains," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 450–465, 2019.

[14] R. Zarringhalam, A. Rezaeian, S. Fallah, A. Khajepour, W. Melek, S.-K. Chen, and B. Litkouhi, "Optimal sensor configuration and fault-tolerant estimation of vehicle states," *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, vol. 6, pp. 83–92, 05 2013.

[15] F. Xu, J. Tan, X. Wang, V. Puig, B. Liang, and B. Yuan, "Mixed active/passive robust fault detection and isolation using set-theoretic unknown input observers," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 863–871, 2017.

[16] S. Park, K. Oh, Y. Jeong, and K. Yi, "Model predictive control-based fault detection and reconstruction algorithm for longitudinal control of autonomous driving vehicle

using multi-sliding mode observer," *Microsystem Technologies*, vol. 26, no. 1, pp. 239–264, 2020.

[17] M. Pirani, M. Hosseinzadeh, J. A. Taylor, and B. Sinopoli, "Optimal active fault detection in inverter-based grids," *IEEE Transactions on Control Systems Technology*, 2022.

[18] R. Wang and J. Wang, "Fault-tolerant control with active fault diagnosis for four-wheel independently driven electric ground vehicles," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 9, pp. 4276–4287, 2011.

[19] M. Luo, D. Wang, M. Pham, C. Low, J. Zhang, D. Zhang, and Y. Zhao, "Model-based fault diagnosis/prognosis for wheeled mobile robots: a review," in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.* IEEE, 2005, pp. 6–pp.

[20] R. Petrone, Z. Zheng, D. Hissel, M.-C. Péra, C. Pianese, M. Sorrentino, M. Béchérif, and N. Yousfi-Steiner, "A review on model-based diagnosis methodologies for pemfcs," *International Journal of Hydrogen Energy*, vol. 38, no. 17, pp. 7077–7091, 2013.

[21] Z. Zhao, P. X. Liu, and J. Gao, "Model-based fault diagnosis methods for systems with stochastic process–a survey," *Neurocomputing*, 2022.

[22] R. Isermann, "Model-based fault-detection and diagnosis–status and applications," *Annual Reviews in control*, vol. 29, no. 1, pp. 71–85, 2005.

[23] T. Jiang, K. Khorasani, and S. Tafazoli, "Parameter estimation-based fault detection, isolation and recovery for nonlinear satellite models," *IEEE Transactions on control systems technology*, vol. 16, no. 4, pp. 799–808, 2008.

[24] K. O'Shea, B.-H. Tsao, L. Herrera, and C. Miller, "Recursive least squares parameter estimation for dc fault detection and localization," in *2019 IEEE National Aerospace and Electronics Conference (NAECON).* IEEE, 2019, pp. 7–10.

[25] E. Che Mid and V. Dua, "Model-based parameter estimation for fault detection using multiparametric programming," *Industrial & Engineering Chemistry Research*, vol. 56, no. 28, pp. 8000–8015, 2017.

[26] H. M. Odendaal and T. Jones, "Actuator fault detection and isolation: An optimised parity space approach," *Control Engineering Practice*, vol. 26, pp. 222–232, 2014.

[27] S. K. Nguang, P. Zhang, and S. X. Ding, "Parity relation based fault estimation for nonlinear systems: An lmi approach," *International Journal of Automation and Computing*, vol. 4, no. 2, pp. 164–168, 2007.

[28] W. Han, Z. Wang, and Y. Shen, "Fault estimation for a quadrotor unmanned aerial vehicle by integrating the parity space approach with recursive least squares," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 232, no. 4, pp. 783–796, 2018.

[29] M. Mansouri, M.-F. Harkat, H. Nounou, and M. N. Nounou, *Data-driven and model-based methods for fault detection and diagnosis*. Elsevier, 2020.

[30] E. Mousavinejad, X. Ge, Q.-L. Han, T. J. Lim, and L. Vlacic, "An ellipsoidal set-membership approach to distributed joint state and sensor fault estimation of autonomous ground vehicles," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 6, pp. 1107–1118, 2021.

[31] G. Heredia, A. Ollero, M. Bejar, and R. Mahtani, "Sensor and actuator fault detection in small autonomous helicopters," *Mechatronics*, vol. 18, no. 2, pp. 90–99, 2008.

[32] G.-R. Duan and R. J. Patton, "Robust fault detection using luenberger-type unknown input observers-a parametric approach," *International Journal of Systems Science*, vol. 32, no. 4, pp. 533–540, 2001.

[33] C. Edwards and C. P. Tan, "A comparison of sliding mode and unknown input observers for fault reconstruction," *European Journal of control*, vol. 12, no. 3, pp. 245–260, 2006.

[34] S. Ibaraki, S. Suryanarayanan, and M. Tomizuka, "Design of luenberger state observers using fixed-structure h/sub/spl infin//optimization and its application to fault detection in lane-keeping control of automated vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 10, no. 1, pp. 34–42, 2005.

[35] S. K. Kommuri, M. Defoort, H. R. Karimi, and K. C. Veluvolu, "A robust observer-based sensor fault-tolerant control for pmsm in electric vehicles," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 12, pp. 7671–7681, 2016.

[36] Y. Jeong, K. Kim, B. Kim, J. Yoon, H. Chong, B. Ko, and K. Yi, "Vehicle sensor and actuator fault detection algorithm for automated vehicles," in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 927–932.

[37] D. Wang and K.-Y. Lum, "Adaptive unknown input observer approach for aircraft actuator fault detection and isolation," *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 1, pp. 31–48, 2007.

[38] Y. Li, H. R. Karimi, Q. Zhang, D. Zhao, and Y. Li, "Fault detection for linear discrete time-varying systems subject to random sensor delay: A riccati equation approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 5, pp. 1707–1716, 2017.

[39] J. Hu, Z. Wang, and H. Gao, "Joint state and fault estimation for time-varying nonlinear systems with randomly occurring faults and sensor saturations," *Automatica*, vol. 97, pp. 150–160, 2018.

[40] A. Forrai, "System identification and fault diagnosis of an electromagnetic actuator," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 1028–1035, 2016.

[41] Z. Luo and H. Fang, "Fault detection for nonlinear systems with unknown input," *Asian Journal of Control*, vol. 15, no. 5, pp. 1503–1509, 2013.

[42] W. Chen and M. Saif, "Fault detection and isolation based on novel unknown input observer design," in *2006 American Control Conference*. IEEE, 2006, pp. 6–pp.

[43] G. Zhiwei, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—part ii: Fault diagnosis with knowledge-based and hybrid/active approaches," 2015.

[44] K. H. Park, E. Park, and H. K. Kim, "Unsupervised fault detection on unmanned aerial vehicles: Encoding and thresholding approach," *Sensors*, vol. 21, no. 6, p. 2208, 2021.

[45] K. Jyoti and S. Singh, "Data clustering approach to industrial process monitoring, fault detection and isolation," *International Journal of Computer Applications*, vol. 17, no. 2, pp. 41–45, 2011.

[46] E. Khalastchi, M. Kalech, G. A. Kaminka, and R. Lin, "Online data-driven anomaly detection in autonomous robots," *Knowledge and Information Systems*, vol. 43, no. 3, pp. 657–688, 2015.

[47] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]," *IEEE Transactions on automatic control*, vol. 42, no. 10, pp. 1482–1484, 1997.

[48] K. Yan, J. Huang, W. Shen, and Z. Ji, "Unsupervised learning for fault detection and diagnosis of air handling units," *Energy and Buildings*, vol. 210, p. 109689, 2020.

[49] J. Ren, R. Ren, M. Green, and X. Huang, "A deep learning method for fault detection of autonomous vehicles," in *2019 14th International Conference on Computer Science & Education (ICCSE)*. IEEE, 2019, pp. 749–754.

[50] R. Sun, Q. Cheng, G. Wang, and W. Y. Ochieng, "A novel online data-driven algorithm for detecting uav navigation sensor faults," *Sensors*, vol. 17, no. 10, p. 2243, 2017.

[51] A. Dutta, M. E. McKay, F. Kopsaftopoulos, and F. Gandhi, "Fault detection and identification for multirotor aircraft by data-driven and statistical learning methods," in *2019 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*. IEEE, 2019, pp. 1–18.

[52] S. Sarkar, X. Jin, and A. Ray, "Data-driven fault detection in aircraft engines with noisy sensor measurements," *Journal of Engineering for Gas Turbines and Power*, vol. 133, no. 8, 2011.

[53] J. Yu, J. Yoo, J. Jang, J. H. Park, and S. Kim, "A novel hybrid of auto-associative kernel regression and dynamic independent component analysis for fault detection in nonlinear multimode processes," *Journal of Process Control*, vol. 68, pp. 129–144, 2018.

[54] M. Kallas, G. Mourot, D. Maquin, and J. Ragot, "Data-driven approach for fault detection and isolation in nonlinear system," *International Journal of Adaptive Control and Signal Processing*, vol. 32, no. 11, pp. 1569–1590, 2018.

[55] P. Baraldi, F. Di Maio, P. Turati, and E. Zio, "A modified auto associative kernel regression method for robust signal reconstruction in nuclear power plant components," in *Proceedings of the European Safety and Reliability Conference, ESREL 2014*, 2014.

[56] Y. Fang, H. Min, W. Wang, Z. Xu, and X. Zhao, "A fault detection and diagnosis system for autonomous vehicles based on hybrid approaches," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9359–9371, 2020.

[57] F. Van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1264–1276, 2019.

[58] M. M. Tousi, A. G. Aghdam, and K. Khorasani, "A hybrid fault diagnosis and recovery for a team of unmanned vehicles," in *2008 IEEE International Conference on System of Systems Engineering*. IEEE, 2008, pp. 1–6.

[59] N. Meskin, K. Khorasani, and C. A. Rabbath, "A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 6, pp. 1422–1429, 2010.

[60] J.-A. Jiang, C.-L. Chuang, Y.-C. Wang, C.-H. Hung, J.-Y. Wang, C.-H. Lee, and Y.-T. Hsiao, "A hybrid framework for fault detection, classification, and location—part i: Concept, structure, and methodology," *IEEE Transactions on Power Delivery*, vol. 26, no. 3, pp. 1988–1998, 2011.

[61] ——, "A hybrid framework for fault detection, classification, and location—part ii: implementation and test results," *IEEE transactions on power delivery*, vol. 26, no. 3, pp. 1999–2008, 2011.

[62] E. Khalastchi, M. Kalech, and L. Rokach, "A hybrid approach for fault detection in autonomous physical agents," BEN-GURION UNIV OF THE NEGEV BEERSHEBA (ISRAEL), Tech. Rep., 2014.

[63] ——, "A hybrid approach for improving unsupervised fault detection for robotic systems," *Expert Systems with Applications*, vol. 81, pp. 372–383, 2017.

[64] M. A. K. Jaradat and R. Langari, "A hybrid intelligent system for fault detection and sensor fusion," *Applied Soft Computing*, vol. 9, no. 1, pp. 415–422, 2009.

[65] D. Jung and C. Sundström, "A combined data-driven and model-based residual selection algorithm for fault detection and isolation," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 616–630, 2017.

[66] M. I. Shapiai, Z. Ibrahim, M. Khalid, L. W. Jau, and V. Pavlovich, "A non-linear function approximation from small samples based on nadaraya-watson kernel regression," in *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE, 2010, pp. 28–32.

[67] A. E. Hassanien *et al.*, *Machine learning paradigms: Theory and application*. Springer, 2019.

[68] L. Fengping, Z. Yuqing, and X. Wei, "Parameter optimization for nadaraya–watson kernel regression method with small samples," *Int. J. Adv. Res. Arfic. Intell*, vol. 5, pp. 1–6, 2016.

[69] H. Guo, D. Cao, H. Chen, C. Lv, H. Wang, and S. Yang, "Vehicle dynamic state estimation: State of the art schemes and perspectives," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 2, pp. 418–431, 2018.

[70] K. B. Singh, M. A. Arat, and S. Taheri, "Literature review and fundamental approaches for vehicle and tire state estimation," *Vehicle system dynamics*, vol. 57, no. 11, pp. 1643–1665, 2019.

[71] J. Na, A. S. Chen, G. Herrmann, R. Burke, and C. Brace, "Vehicle engine torque estimation via unknown input observer and adaptive parameter estimation," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 409–422, 2017.

[72] A.-T. Nguyen, T.-M. Guerra, C. Sentouh, and H. Zhang, "Unknown input observers for simultaneous estimation of vehicle dynamics and driver torque: Theoretical design and hardware experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 6, pp. 2508–2518, 2019.

[73] B. Soualmi, C. Sentouh, and J.-C. Popieul, "Both vehicle state and driver's torque estimation using unknown input proportional multi-integral ts observer," in *2014 European Control Conference (ECC)*. IEEE, 2014, pp. 2957–2962.

[74] B. Zhang, H. Du, J. Lam, N. Zhang, and W. Li, "A novel observer design for simultaneous estimation of vehicle steering angle and sideslip angle," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 7, pp. 4357–4366, 2016.

[75] T. Strutz, *Data fitting and uncertainty: A practical introduction to weighted least squares and beyond*. Springer, 2011.

[76] R. Isermann, *Fault-diagnosis applications: model-based condition monitoring: actuators, drives, machinery, plants, sensors, and fault-tolerant systems*. Springer Science & Business Media, 2011.

[77] I. D. Landau, R. Lozano, M. M'Saad, and A. Karimi, *Adaptive control: algorithms, analysis and applications*. Springer Science & Business Media, 2011.

[78] C. R. Johnson, "Lectures on adaptive parameter estimation," *Prentice Hall Advanced Reference Series*, pp. 160–163, 1988.

[79] G. A. Seber and A. J. Lee, *Linear regression analysis*. John Wiley & Sons, 2012.

[80] R. Zarringhalam, "Sensor fault detection and fault-tolerant estimation of vehicle states," 2023.

[81] A. Rezaeian, R. Zarringhalam, S. Fallah, W. Melek, A. Khajepour, S.-K. Chen, N. Moshchuck, and B. Litkouhi, "Novel tire force estimation strategy for real-time implementation on vehicle applications," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2231–2241, 2014.

[82] E. Hashemi, "Full vehicle state estimation using a holistic corner-based approach," 2017.

[83] E. Hashemi, M. Pirani, A. Khajepour, A. Kasaiezadeh, S.-K. Chen, and B. Litkouhi, "Corner-based estimation of tire forces and vehicle velocities robust to road conditions," *Control Engineering Practice*, vol. 61, pp. 28–40, 2017.

[84] A. Rezaeian, "Development of an integrated estimation method for vehicle states, parameters and tire forces," 2015.