2010

Real-time implementation of a sensor validation scheme for a heavy-duty diesel engine

Manoharan Thiagarajan
West Virginia University

Recommended Citation

Thiagarajan, Manoharan, "Real-time implementation of a sensor validation scheme for a heavy-duty diesel engine" (2010). *Graduate Theses, Dissertations, and Problem Reports*. 2977.
https://researchrepository.wvu.edu/etd/2977

Real-time Implementation of a Sensor Validation Scheme for a Heavy-duty Diesel Engine

Manoharan Thiagarajan

Dissertation submitted to the College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Mridul Gautam, Ph.D., Chair
Donald Ferguson, Ph.D.
Hailin Li, Ph.D.
Natalia Schmid, Ph.D.
Benjamin C. Shade, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia
2010

Keywords: Fault Diagnosis, SFDIA, Sensor Validation, Neural Networks, Emissions

ABSTRACT

Real-time Implementation of a Sensor Validation Scheme for a Heavy-duty Diesel Engine

Manoharan Thiagarajan

With ultra-low exhaust emissions standards, heavy-duty diesel engines (HDDEs) are dependent upon a myriad of sensors to optimize power output and exhaust emissions. Apart from acquiring and processing sensor signals, engine control modules should also have capabilities to report and compensate for sensors that have failed. The global objective of this research was to develop strategies to enable HDDEs to maintain nominal in-use performance during periods of sensor failures. Specifically, the work explored the creation of a sensor validation scheme to detect, isolate, and accommodate sensor failures in HDDEs. The scheme not only offers onboard diagnostic (OBD) capabilities, but also control of engine performance in the event of sensor failures. The scheme, known as Sensor Failure Detection Isolation and Accommodation (SFDIA), depends on mathematical models for its functionality. Neural approximators served as the modeling tool featuring online adaptive capabilities. The significance of the SFDIA is that it can enhance an engine management system (EMS) capability to control performance under any operating conditions when sensors fail. The SFDIA scheme updates models during the lifetime of an engine under real world, in-use conditions. The central hypothesis for the work was that the SFDIA scheme would allow continuous normal operation of HDDEs under conditions of sensor failures. The SFDIA was tested using the boost pressure, coolant temperature, and fuel pressure sensors to evaluate its performance. The test engine was a 2004 Mack® MP7-355E (11 L, 355 hp). Experimental work was conducted at the Engine and Emissions Research Laboratory (EERL) at West Virginia University (WVU). Failure modes modeled were abrupt, long-term drift and intermittent failures. During the accommodation phase, the SFDIA restored engine power up to 0.64% to nominal. In addition, oxides of nitrogen (NOx) emissions were maintained at up to 1.41% to nominal.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADALINE	Adaptive Linear Network
ADI	Analog Devices$^{\circledR}$
AE	Accommodation Enable
ASTM	American Society for Testing and Materials
AT	Inlet-air Temperature
BP	Boost Pressure
BT	Boost Temperature
CAN	Controller Area Network
CARB	California Air Resources Board
CFR	Code of Federal Regulations
CO	Carbon Monoxide
CO_2	Carbon Dioxide
COV	Coefficient of Variation
CT	Coolant Temperature
CVS	Constant Volume Sampler
DAQ	Data Acquisition
DC	Direct Current
DOC	Diesel Oxidation Catalyst
DPF	Diesel Particulate Filter
EERL	Engine and Emissions Research Laboratory
EGR	Exhaust Gas Recirculation
EMRAN	Extended Minimal Resource Allocating Network
EMS	Engine Management System
EPA	Environmental Protection Agency
ET	EGR Temperature
EUI	Electronic Unit Injector
FADEC	Full Authority Digital Engine Control
FM1	Failure Mode 1
FM2	Failure Mode 2

FM3	Failure Mode 3
FP	Fuel Pressure
FTP	Federal Test Procedure
HC	Total Hydrocarbon
HDDE	Heavy-duty Diesel Engine
HFID	Heated Flame Ionization Detector
IEEE	Institute of Electrical and Electronics Engineers
LE	Learning Enable
MARS	Multiple Adaptive Regression Splines®
MOGA	Multi-objective Genetic Algorithms
MRAN	Minimal Resource Allocating Network
NDIR	Non-dispersive Infrared Detector
NI	National Instruments®
NLPCA	Nonlinear Principal Component Analysis
NN	Artificial Neural Networks
NOx	Oxides of Nitrogen
OBD	Onboard Diagnostic
OP	Oil Pressure
OT	Oil Temperature
PCA	Principal Component Analysis
PM	Particulate Matter
QC-QA	Quality Control - Quality Assurance
RAN	Resource Allocating Network
RBF	Radial Basis Function
RMS	Root Mean Square
SAE	Society of Automotive Engineers
SCR	Selective Catalytic Reduction
SFA	Failure Accommodation
SFDI	Failure Detection and Isolation
SFDIA	Sensor Failure Detection Isolation and Accommodation
SRA	Smart Remote Actuator

TDC	Top Dead Center
UGV	Unmanned Ground Vehicle
UVA	Unmanned Aerial Vehicle
VGT	Variable Geometry Turbocharger
V-MAC IV	Vehicle Management and Control System IV®
VTS	VGT Turbocharger Speed
WVU	West Virginia University

CHAPTER 1 INTRODUCTION

1.1 Overview

Modern heavy-duty diesel engines (HDDEs) are highly dependent upon the available sensors for emissions control and operation at optimum power. Such dependency becomes especially important, as engines are required to maintain or show improvement in durability, reliability, performance, and fuel economy with ever more stringent emissions regulations. All the engine sensors work in conjunction with the engine management system (EMS), a computer module with the purpose of engine and emissions control. Within it are algorithms programmed to accept sensor data and in turn output control signals to the various actuators around an engine.

To operate HDDEs according to the Environmental Protection Agency (EPA) emissions specifications, engine manufacturers developed technologies that include the variable geometry turbocharger (VGT), exhaust gas recirculation (EGR) system and the electronic unit injector (EUI). EUIs provide advanced control of fuel injection pressure and timing. Each of these devices contains multiple sensors that communicate with the EMS to maximize performance and keep emissions at the required levels. Owing to the new 2010 EPA standards for HDDEs, engine manufacturers have begun implementing advanced exhaust aftertreatment devices. These devices include the Selective Catalytic Reduction (SCR) system and Diesel Particulate Filter (DPF) along with their associated sensors [1]. Beginning in 2010, the EPA and California Air Resources Board (CARB) require engine manufacturers install a standardized onboard diagnostic (OBD) system on heavy-duty engines [2-3]. The EPA requires full compliance to the OBD regulations by 2013. It replaces the current Engine Manufacturers Diagnostic (EMD) that has no standardization requirements. Similarly, this new heavy-duty OBD (HD OBD) means heavy-duty vehicle manufacturers have to equip their vehicles with new sensors in addition to the previous ones already in use. The HD OBD primary function is to inform the vehicle operator if the pollution control system is failing. This could lead to repairs or complete replacement of parts, if necessary. All these additional sensors give an EMS added degrees of freedom and at the same time add to the challenges of controlling the engine. With all the extra complexity of HDDEs, an EMS does not include features for compensating sensor failures.

1.2 Hypothesis and Objectives

The ***central hypothesis*** for this research is that a sensor validation scheme can detect, isolate and correct (accommodate) sensor failures in HDDEs to operate normally in all conditions. The scheme called Sensor Failure Detection Isolation and Accommodation (SFDIA) [4] has been successfully applied to fault tolerant flight control systems. Malfunctioning sensors can lead to inconsistent engine operation that consequently disrupt engine control or simply shut it down. Such malfunctions could also trigger the "limp-home mode", which is undesirable for trucks hauling heavy cargo on highways. Limp-home mode is a condition set by the EMS when it detects a serious problem within the vehicle and proceeds to limit engine power but just enough to run to the next available service station. Correcting the sensor data becomes important in such instances especially when the component associated with the faulty sensor still functions properly.

The ***global objective*** of this research was to enhance the in-use performance of HDDEs during periods of sensor failures. The ***specific objective*** was to validate the SFDIA functionality in real-time inline with a test engine using the boost pressure, coolant temperature and fuel pressure sensors. Artificial neural networks are the modeling tool of choice.

1.3 Significance

Sensors are mechanical devices and are not immune to failures. Depending on the application, each sensor construction is different, thus, having varying levels of robustness. In HDDEs, sensors must remain reliable without the need to verify calibrations. Pulling into a garage frequently just to check sensor calibrations is cost prohibitive. The SFDIA provides a solution to this issue by permitting the EMS to control performance under several instances of sensor-based failures. As an example, engine manufacturers use crankshaft (engine) speed and position, and boost pressure sensors to provide primary control data to determine fuel injection timing. The second category of sensors is coolant temperature, boost temperature and throttle (pedal) position [5]. If any of these sensors were to fail and send erroneous information to the EMS, it can easily degrade its emission control capabilities while also affecting power. The proposed SFDIA will rectify the situation in real-time by detecting the failure, isolating the faults and accommodating the failed sensors with best approximate signals.

CHAPTER 2 LITERATURE REVIEW

This chapter presents an overview of the differences between analytical and "black box" modeling methods. Due to the focus of this research, several sections are dedicated to the topic of system fault prevention, neural networks (NNs) and sensors. The type of NN used in this study is a hybrid network. It combines two well-known network algorithms, namely the Adaptive Linear Network (ADALINE) and the Radial Basis Function (RBF) network featuring the Extended Minimal Resource Allocating Network (EMRAN) algorithm, to provide the best possible approximation.

2.1 Overview of Mathematical Models for Engines

There are several approaches available to model an HDDE along with its various components. One approach involves the development of analytical models using physics-based principles such as thermodynamics, fluids mechanics, and combustion kinetics [6-10]. In theory, these methods can provide great accuracy while giving a clear analytical representation of the system under investigation. They are usually very specific to the processes occurring within the modeled system. However, there are disadvantages associated with such modeling techniques. A rather long and expensive effort has to be expended within the initial model development phase. Within the maintenance phase, the model requires adaptation to the progressive changes made to an engine over the model lifecycle. These changes include the addition of new aftertreatment technologies that will unavoidably incur additional costs associated with system modeling. In addition, it is not expected that a model developed for one engine design will work on another.

A suitable approach for modeling engine systems involves linear and nonlinear "black box" tools such as neural networks, linear and nonlinear regression, system identification, and fuzzy logic [11-13]. The main advantage provided by these tools is that they have automatic model-building features. This is useful for systems too difficult to develop using analytical methods and accomplished through the application of training algorithms. Although such models are not transparent, these techniques can achieve good accuracies at a fraction of the developmental costs. In fact, these tools are normally easy to set up and are available within several standard mathematics and engineering software packages. Neural networks are particularly suitable for HDDE research as it can include automatic model updating features to account for changes to the engine physical condition during its lifetime. Such updating can happen under real world, in-use conditions. The following sections provide more details regarding modeling techniques and fault tolerant solutions.

2.2 Fault Tolerant Technologies

2.2.1 Concept Overview

Any system that operates using sensors and actuators will require features to protect its functionality, and thus improve reliability. Such features will involve constant monitoring of the system operation resembling an OBD system. In the event an error occurs, it must detect, identify and preferably correct the error without shutting a system down to restore normal operation. The concept of diagnosis here involves detecting and identifying faults in a given system. To accommodate implies substituting or correcting a failed sensor signal with an estimated one. A properly functioning fault diagnosis and correcting method can have various advantages. This consists of preventing severe damage from occurring, extending its system operation lifespan and providing an operator with an accurate location of the fault. Taking a system offline frequently for inspection is cost prohibitive and labor intensive, however it is one way of ensuring proper functionality. To resolve this issue, various methods researched in the topic of fault prevention involved model-based approaches. Model-based in this study refers to methods that use mathematical models created with software tools. Such an approach is capable of handling a failure when it happens or programmed to diagnose a problem ahead of time followed by alerting the operator before the problem worsens. A model-based approach would have built in thresholds that when exceeded will broadcast a warning and accommodate the error if it is correctable.

One such model-based scheme that diagnoses and accommodates errors is a sensor validation scheme. The idea behind this scheme is to provide a system with sensor fault tolerant abilities. It does this by detecting any abnormalities in the signals and if it exists, corrects the signals by approximating it from the other sensors. This would ultimately mean terminating the signal originating from the failed sensor. Correcting a faulty sensor is critical to prevent the system from reacting to a process fault, which is the assumption that the component the sensor is monitoring has failed. It is noted that any sensor signal accommodated will feedback into the scheme. Therefore, sensor validation will have limitations as to the number of sensors it can accommodate before rendering the built-in models ineffective. The figure below summarizes the logic behind a sensor validation scheme:

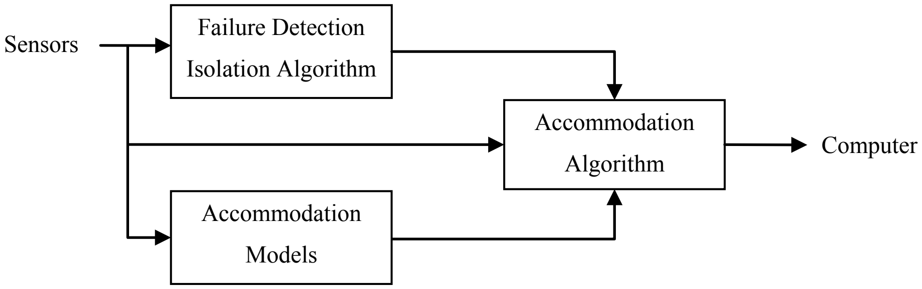


Figure 1: Sensor validation scheme.

A fault detection algorithm involves a process of monitoring the difference between a set of models and the sensor signals. This difference is called a residual. When this residual crosses a specified threshold, it is indicative of a warning or a failure has taken place. The models used to determine the residuals are separate from the accommodation models. Depending on the nature of the scheme setup, the accommodation algorithm will decide whether to initiate the accommodation mode.

In general, sensor errors are due to the various effects mentioned below [14]:

a) Faulty sensor (manufacturing flaw, wear, calibration error, loose wires)
b) Communication channel error (signal interference, cross-talking, jitter, scattering)
c) Data acquisition and data processing component faults (memory, power failure)
d) Computer hardware/software-related programming errors (software errors or "bugs", design error)
e) Transient errors (nonlinearity imperfections)

It is easy to detect a sensor that fails catastrophically, however detecting a malfunction that appears intermittent or as drifts can be challenging. A sensor validation scheme has the advantage of detecting both forms of sensor issues. The robustness of a sensor validation scheme will depend on its ability to avoid false alarms and when in the accommodation mode, provide an accurate representation of the faulty signal in real-time. To correct failures automatically is an indication of analytical redundancy [15]. Analytical redundancy essentially implies taking advantage of the functional relationship existing between the inputs, states, and outputs of the system. Adding more sensors to measure the same parameter as implied by physical redundancy, along with the voting scheme used by such systems, means extra cost while introducing increased complexity to the system. The extra hardware can also be space consuming and impose a weight penalty.

2.2.2 Types of Systems and Fault Prevention Studies

The types of systems whose operations are dependent on sensors and actuators include, but not limited to, the following:

a) Power plants
b) Chemical plants
c) Robotic systems
d) Aircraft
e) On-road vehicles

All systems mentioned above stand to benefit from a sensor validation scheme. Even civil engineering structures such as bridges that employ smart technology can deploy a sensor validation scheme [16]. In the following paragraphs, various fault prevention techniques that are in use or studied are presented with an emphasis on sensor fault prevention.

Both power and chemical plants have complex industrial processes running continuously while maintaining output quality and overall plant safety. Sensors located around a plant are monitoring various processes and have to be reliable at all times to avoid potential disasters. In coal power plants for instance, inaccurate sensor readings can cause combustion inefficiencies resulting in increased pollution. In a study that applied a sensor validation scheme to a boiling water nuclear reactor, it was necessary to monitor the sensors in specific groups instead of using a single, large scheme due to the number of sensors involved [17]. The method employed multi-objective genetic algorithms (MOGA) to search for optimal sensor groupings with similar characteristics. Genetic algorithm operations exhibit similar biological and behavioral traits as inherent in living beings. In general, it performs probabilistic optimization to one or more functions, known as objective functions, which are both linear and nonlinear. Data from the sensor signals contribute to the development of the objective functions. Below is a summary of methods that can reconstruct sensor signals obtained from a sensor validation study on a coal-fired power plant boiler [18]:

Linear Models: Linear models are popular in the field of modeling for systems exhibiting linear operations, especially in power and chemical plants where its operation depends on parameters that do not have significant variations. Principle component analysis (PCA) is an example of a method to develop linear models. It creates a model by reducing the dimensionality of a data set without sacrificing the signal variation [17, 19]. State space model is another form of a linear model derived from first-order differential equations:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1}$$

where A is the state matrix, B is the input matrix, C is the output matrix and D is the feed-forward matrix which in most cases is zero in value. In addition, $x(t)$ are the state variables, $u(t)$ is the input vector and $y(t)$ being the model output. State variables are the parameters that decide the state of the system and contain all the information to predict the future condition of the system. Most systems are nonlinear in nature making linear models unable to cope with high signal variations. For PCA methods, there are solutions researched to cope with nonlinear situations through the development of kernel PCA and nonlinear PCA (NLPCA) [20-23]. In the case of the kernel PCA, the linear model still performs the detection, but a kernel function that is similar to a weighting function, transforms the data into a high dimensional feature space for nonlinear mapping.

Conservation laws and data reconciliation: When using conservation laws to monitor the sensor signals, any malfunction would reveal itself as a violation. An example of a conservation law is the conservation of energy principle in thermodynamics. To prevent errors from affecting the conservation laws that lead to false alarms, especially in chemical processes, a method called data reconciliation showed better results than traditional techniques such as moving averages and wavelet filtering [24]. Data reconciliation optimally adjusts the measured data to increase the robustness of a model to obey the conservation laws and other constraints. Data reconciliation is also applicable to other modeling techniques.

Bayesian-Belief Network: The concept of a Bayesian-Belief network is to represent dependencies in the dataset graphically for probabilistic reasoning of a given system. A probabilistic reasoning network can determine the system behavior simply by examining the input variables and reason how they interact. The network assigns a value to the input variables and propagates their effects through the network to determine an output. Various researchers have suggested utilizing the Bayesian-belief network for a sensor validation scheme [25-27]. By assigning probabilities to a dataset consisting of faulty and nominal signals, it is possible to calculate the likelihood of a specific fault to occur.

Data Mining: Data mining is an approach that extracts information from a given dataset. One popular method for data mining is the Multiple Adaptive Regression Splines (MARS) [28-29]. MARS is a regression-modeling tool developed by Jerome Friedman in the early 1990s [30]. It automatically constructs the model from a set of coefficients and basis functions that are entirely "driven" from the regression data. Specifically, using a “divide and conquer” strategy, MARS partitions the input space into several regions, each with its own regression equation. This technique excels at finding optimal variable transformations and interactions within high-dimensional data of complex structures. In fact, this approach often reveals important data patterns and relationships that other methods may struggle to uncover [31]. A study performed by Chew [32] showed that MARS is capable of sensor validation tasks.

An aircraft system containing fault prevention technology has various benefits apart from safety reasons. With modern aircraft using digital fly-by-wire technology, the entire system very much relies on the sensors for control. Any type of failure to the flight control system will have catastrophic consequences. Aircraft sensors typically have up to four levels of physical redundancy [33]. A sensor validation scheme has the potential to increase sensor reliability and eliminate the need of having four independent sensors measuring the same parameter, hence the various studies conducted into this concept [11, 34-37]. The Controls and Dynamics Technology Branch at the NASA Glenn Research Center investigated an advanced adaptive flight control system that utilized model-based approaches similar to sensor validation [38-39]. It can allow the aircraft engine to operate beyond its normal operation domain. The work proposed methods for improving aircraft survivability during adverse conditions through better controlling of thrust. To accomplish this requires further enhancing the current Full Authority Digital Engine Control (FADEC) system for better power management, faster thrust response and producing more thrust for short periods when required. One of the solutions proposed is to validate pilot throttle command with a model that estimates thrust to compensate for any adverse conditions encountered by the engine control. The model receives inputs directly from the onboard sensors. The modeling tool used was a standard Kalman filter. The Kalman filter is state space model based and frequently applied to aircraft studies. Another area used for Kalman filter work is in robotic systems such as unmanned ground vehicles (UGV). An array of sensors is present on UGVs that measures robot orientation, angular rate, linear acceleration, velocity and position, thus making it ideal for applying sensor fault diagnosis technology [40].

The need of a sensor validation scheme for on-road vehicles stem from the task to better control emissions due to lower EPA and CARB regulations. Other concerns include maximizing fuel economy and the implementation of modern electronic safety features. The regulations led to the development of an electronic OBD system that monitors malfunctions, pinpoints a specific problem, and broadcasts a fault code computed by its unique software. The software performs this task using signals from various sensors. A device meant to scan the codes plug into a Society of Automotive Engineers (SAE) standardized interface located within the vehicle, as shown in Figures 2 and 3. Such scan tools can transmit the data to a laptop computer via a USB connection and allow a mechanic to determine what needs repaired or replacement. A typical location of the 16-pin OBD connector is under the driver side dashboard. The OBD II, the current version for diagnosing faults associated primarily with the engine emissions system, applies to vehicles model year 1996 and onwards [41]. The types of vehicles involved are passenger cars, and light vans and trucks powered by an internal combustion engine. Modern OBD II systems also monitor conditions on the vehicle body and chassis. However, the OBD II lacks the ability to correct faults caused by malfunctioning sensors. The HD OBD will be similar to the OBD II specifications but with an emphasis on regulating oxides of nitrogen (*NOx*) and particulate matter (*PM*) that are major issues in HDDEs [42]. The primary areas monitored by HD OBD are the fuel system, cylinder misfires, exhaust aftertreatment system and boost pressure components that include the turbocharger and charge air cooler. Emissions monitoring in HD OBD will use specified thresholds to determine when emissions are higher that normal.

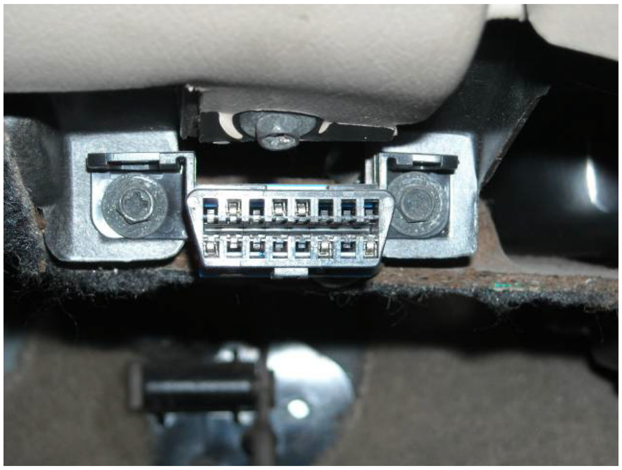


Figure 2: SAE J1962 connector.

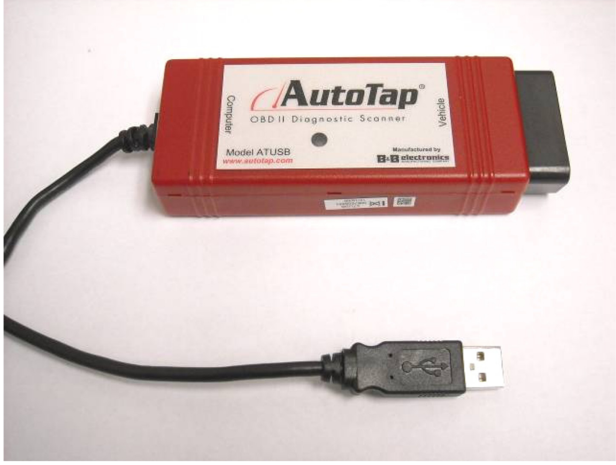


Figure 3: OBD II scan tool by B&B Electronics.

Studies have showed a variety of model-based fault diagnosis work with diesel engines. Some areas investigated were through modeling engine vibration characteristics or the intake manifold parameters and comparing their outputs with sensor readings for the purpose of fault diagnosis [23, 43-47]. The combination of an OBD with a sensor validation scheme can provide the OBD with engine control capabilities when correcting sensor errors. Unlike systems such as an aircraft, on-road engines do not have physical sensor redundancies. Research into on-road engine sensor validation does exist, but are few to report on and none tested inline with an EMS for direct control [48-51]. However, all these other studies did demonstrate unique ideas to perform sensor validation. Most of the work referenced here on model-based diesel engine fault prevention chose neural networks as the modeling tool. Section 2.3 provides details about neural networks and introduces the aforementioned hybrid network.

Applications such as environmental monitoring could deploy a technology called a wireless sensor network [52]. It consists of a number of sensors placed over a large area in the form of nodes on a grid measuring everything from temperature, light, sound and humidity. Each node on the network must have their own power supply to operate its sensing elements and transmit data in a wireless format. To maintain its power longevity means having very limited computational and storage capabilities. The military also relies on wireless sensor networks to transmit critical information on enemy location or weapons targeting. Nodes for military applications could also be mobile when placed on robotic vehicles [53]. In cases where node placements are dense within an area, it is possible to program the nodes to collaborate prior to releasing the data to a processing station. As such, densely placed nodes are ideal candidates for a model-based approximation in the event one of the nodes fails, as it is critical for a wireless sensor network to be fault-tolerant [54].

To improve the reliability of a sensor network, the Institute of Electrical and Electronics Engineers (IEEE) implemented the IEEE 1451 standard for defining the type of interface within a given wireless network. The main idea of the standard is to help sensor manufacturers develop smart technology and interface them to the networks, systems, and instruments already in current or emerging networks [55]. One of the features of these smart sensors is the Transducer Electronic Data Sheet (TEDS) that contains information such as sensor identification, measurement range, location, calibration, user information, and more that transmits across the interface [56].

2.3 Artificial Neural Networks

Artificial NNs are mathematical models based on neural biology. They are capable of providing solutions to various problems associated with function approximation, pattern recognition, optimization, signal classification, adaptive control, and system identification. A study by Gaura and Kraft [57] showed that NNs are suitable for modeling and diagnosing sensor faults. NNs are data driven and thus require training. Training here means evaluating or estimating the model parameters to link the inputs and outputs of a physical system. When the operation characteristics of the system alter, then it will require retraining. The entire training procedure can be time consuming [58]. However, NNs with real-time adaptive capabilities can address the issues of training duration and process changes.

Similar to biological neural networks, NNs process information at many simple interconnecting elements called neurons. Signal transmission between neurons occurs over connection links. Each connection link has an associated weight. In typical NNs the weights contain information used by the network to solve a given problem. The neurons apply an activation function to their input to determine their output signals [59].

In general, NNs are defined by the following characteristics:

- Type of connection links between the neurons, also known as its architecture
- Type of activation function, Φ, within the neurons
- Training algorithm used to determine the weights and other parameters

In the NN, the weights are specific objective functions used to derive an optimal solution.

With respect to the different architectures, a common way to classify NNs is single-layer or multi-layer. Single-layer architectures consist of a single output layer of neurons that processes the input data. Multi-layer architectures have more “hidden” layers of neurons between the input and the output layers. Furthermore, a NN has a “feed-forward” architecture if no signal from any layer is fed back as input into a previous layer [59]. In this study, all the NN architectures are of the feed-forward type. As mentioned earlier, there are sensor validation studies on diesel engines using NNs and some presented adaptive type NNs [49]. However, this research focuses on a new class of NNs, which is a major novelty of the SFDIA. It is discussed below along with its adaptive capabilities. It is the same hybrid NN used by Campa et al. [48] for a sensor validation scheme.

2.4 ADALINE + EMRAN RBF Networks

The Radial Basis Function Neural Networks (RBF NNs) are extensively used for non-linear system identification and online sequential learning (adaptive learning) due to their local and global generalization capabilities [60-62]. For this type of NNs, the estimation of a nonlinear engine parameter typically occurs through the weighted superposition of a number of *m* basis functions. In the case of Gaussian basis functions, a RBF NN is expressed as:

$$\hat{y}_{ran}(t) = \theta) f(t, = w\sum_{i=1}^{m} e_{i} \cdot {}^{\left(-\frac{\left\|x(t)\text{-}\mu_{i}(t)\right\|^{2}}{2\sigma_{i}^{2}(t)}\right)} \qquad (2)$$

$$\theta = \left[w_{1},\ldots,w_{m},\mu_{1},\ldots,\mu_{m},\sigma_{1}{}^{2},\ldots,\sigma_{m}{}^{2} \right]$$

where $x(t)$ is the vector of n regressing variables (input variables) at time instant t, and θ represents the set of parameters tuned by the training or adaptive algorithm. Specifically, w_i are the basis function weights, while μ_{i} and $\sigma_i{}^2$ are the center and variance of the i-th of m basis functions respectively. The center positions and variances are equally spaced in a uniform grid. The exponential term in (2) is the activation function. The NN has two layers, as Figure 4 illustrates:

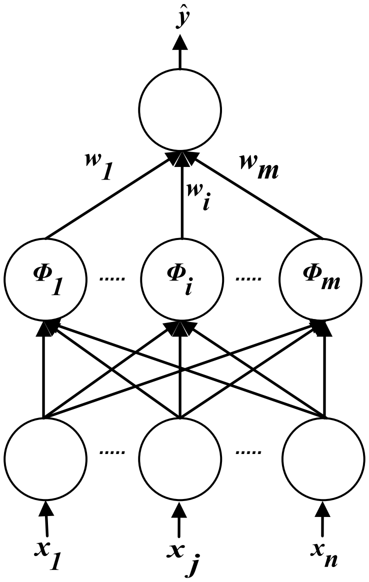


Figure 4: RBF NN architecture.

Various well-known results, based for example on the Stone-Weierstrass theorem, show that this kind of network can uniformly approximate any continuous function to an arbitrary degree of accuracy if there are sufficient neurons [63]. During the model development phase, the total number of neurons necessary to ensure a specified accuracy grows considerably with the dimension of the inputs. This is especially true as the number of weights, *m* increases linearly with the measurement. This problem is known in the technical literature as the "curse of dimensionality" [64]. Several variations of RBF NNs have been developed to deal with this issue. One of these variations is a new class of RBF NNs known as Resource Allocating Network (RAN). The main advantage of RAN NNs is that the topology of the NN is not fixed, but grows according to the perceived complexity of the model. The addition of neurons occurs only in the region of the input domain where the mapping accuracy is low. The three criteria listed below govern the addition of new neurons as the input data ($x(t)$,$y(t)$) are sequentially sampled at time instant t:

Current estimation error criteria: The estimation error must be bigger than a threshold E_1:

$$e(t) = y(t) - \hat{y}(t) > E_1 \tag{3}$$

Novelty criteria: The nearest center distance must be bigger than a threshold E_2:

$$\inf_{i=1}^{m} \left\| x(t) - \mu_i(t) \right\| \geq E_2 \tag{4}$$

Windowed mean error criteria: The windowed error must be bigger than a threshold E_3:

$$\frac{1}{N} \sum_{i=0}^{N} \left[y(t-N+i) - \hat{y}(t-N+i) \right] \geq E_3 \tag{5}$$

where $e(t)$ is the prediction error and N is the number of samples in the current sampling window. The initialization of each neuron is as follows:

$$w_{m+1}(t) = e(t) = y(t) - \hat{y}(t) \tag{6}$$

$$\mu_{m+1}(t) = x(t) \tag{7}$$

$$\sigma_{m+1} = \lambda \inf_{i=1}^{m} \left\| x(t) - \mu_i(t) \right\| \tag{8}$$

where λ is the overlapping factor. All three criteria in Equation (3) through Equation (5) must satisfy to facilitate the addition of a neuron. In the event one or more of the above criteria are not satisfied, the tuning parameter θ undergoes an updating process as described by the gradient-based algorithm below:

$$\theta(t+1) = \theta(t) - \eta \left. \frac{\partial \hat{y}(t)}{\partial \theta(t)} \right|_{(t)} \cdot e(t) \tag{9}$$

where η is the learning rate. If necessary, a pruning strategy implemented within the network can avoid an excessive increase of the NN size. In this case, the resulting algorithm is called the

Minimal RAN (MRAN) [65]. The Extended MRAN (EMRAN) is a powerful variation of the MRAN with the capability for online adaptation using limited computing resources [66]. Only the most activated neurons undergo updating while the others remain unchanged. This implies a reduction in the number of parameters to be updated, thus making it suitable for online applications. It is important to point out that the bulk of the learning must occur during the offline training phase.

A single-layer ADALINE NN can efficiently provide a mapping of the linear portion of the system dynamics. The following equation describes the linear network:

$$\hat{y}_{lin}(t) = \sum_{j=1}^{n} p_j \cdot x_j(t) \tag{10}$$

where *x*(*t*) is the activation function and also the input variables. The architecture for the ADALINE NN is:

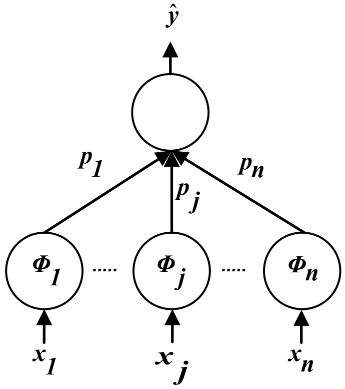


Figure 5: ADALINE NN architecture.

The weights of this network p_j are evaluated offline by applying the Least Square (LS) method to the input data at nominal conditions.

The hybrid structure comprising the ADALINE + EMRAN RBF NN may be represented as:

$$\begin{aligned} \hat{y}(t,p,\theta) &= \hat{y}_{ran}(t,\theta) + \hat{y}_{lin}(t,p) \\ &= \sum_{i=1}^{m} w_i(t) \cdot e^{\left(-\frac{\|x(t)-\mu_i(t)\|^2}{2\sigma_i^2(t)}\right)} + \sum_{j=1}^{n} p_j \cdot x_j(t) \end{aligned} \tag{11}$$

The weights, p_j is sufficient to estimate the importance of each input variable for a given parameter that Equation (11) models. Although the ADALINE + EMRAN RBF NN structure seems over complicated, the simple and computationally efficient ADALINE NN provides substantial approximation capabilities at linear operating conditions. The RAN algorithm allocates RBF neurons only where the system dynamics exhibit significant nonlinearities. This breakdown allowed for a considerable reduction in the number of basis functions. In addition, it has online adaptive capabilities to shorten the duration of the offline training phase with the idea that model refining can occur once implemented into a system.

2.5 Vehicle Sensors

In modern vehicles, sensors are the devices that provide information to various computer modules about a specific parameter for the purpose of onboard diagnostics and control. Actuators then receive the output signals from the computer to perform the required control. The sensing technology for each sensor differs based on the parameter it is monitoring. However, the sensor signals transmitted to the computers are usually in the form of analog electrical voltages. As vehicle technology continues to advance with the implementation of better safety features and pollution control, it is critical that sensors perform reliably under any operating conditions.

A common computer-controlled mechanism is the antilock braking system (ABS). Wheel speed sensors work in conjunction with the ABS computer to control various hydraulic valves and a modulator pump to prevent wheel lockup in the event of emergencies. Traction and stability control also depend on the ABS system for their operation but with additional hardware primarily in the form of extra hydraulic valves. In the case of engine control, heavy-duty engines use a variety of sensors that transmit data to the EMS. Primary actuators on modern HDDEs include EUI control valves, VGT sliding nozzle ring, SCR urea doser and EGR valve. The following sections present information about the technology behind common sensors used by HDDEs, with various concepts provided by Bonnick [41].

2.5.1 Speed and Position Sensor

Crankshaft speed and position are critical parameters used for optimizing fuel timing. Typically, these sensors operate based on variable reluctance technology. The term reluctance implies a resistance to magnetism. The concept of air having greater resistance to magnetism than metal forms the basis of the sensor operation. The reluctor for the sensor is a rotor with notches on its perimeter, with a narrow air gap between the sensor and rotor for optimal signal strength. These sensors are also known as magnetic (mag) pickup sensors. For engines, the rotor is the flywheel and the sensor fixed to the flywheel housing. These sensors consist of a magnet and a wire coiled around the magnet pole piece. This wire exiting the sensor housing is twisted in order to eliminate electrical interference. The figure below illustrates this:

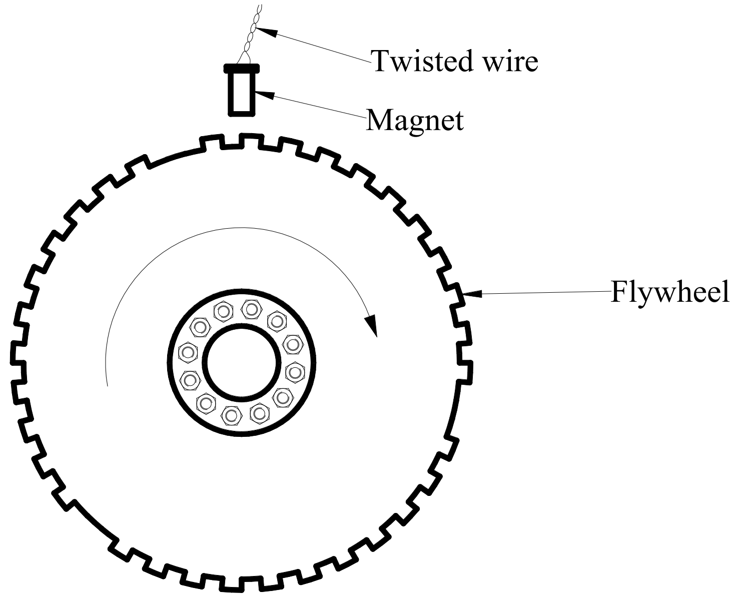


Figure 6: Speed sensor positioned above flywheel.

The notches for the sensor are not the gear teeth used by the starter motor. A typical shape of a single speed sensor notch is short and flat, as opposed to the pointed gear teeth for the starter motor. As each notch passes under the sensor, it induces a sinusoidal voltage signal in the wire coil. The frequency of this signal determines the engine speed. Missing notches on the flywheel, thus producing no signal pulse, allow the EMS to know the location of top dead center (TDC) for all pistons. Therefore, this sensor also provides the EMS crankshaft position information. The number of missing notches varies based on manufacturer specifications.

Another sensor operating with the same technology is the camshaft position sensor. For heavy-duty engines, the reluctor for this sensor can be the camshaft gear or the camshaft vibration dampener with notches machined onto it. Camshaft position is critical for engine ignition and fueling control. For a six-cylinder engine, one possible configuration is there would be seven notches. Six notches indicate TDC of the pistons and the seventh providing a reference point for cylinder No.1 position [67]. This sensor also measures indirectly crankshaft speed and position. The camshaft spins at half the crankshaft speed via idler gears. As a result, this sensor serves as a backup in the event the crankshaft speed sensor fails. In some cases, it serves as the primary speed sensor due to better accuracy as a function of time. The higher accuracy is due to its slower speed.

2.5.2 Pressure Transducer

An EMS monitors pressure at multiple locations around an engine. The pressure parameters include boost, crankcase, fuel and oil. Boost pressure is an important parameter for fuel timing and EGR control. The sensor that measures boost pressure, called the manifold absolute pressure (MAP) sensor, sends information to the EMS to calculate the amount of engine intake air. The form of technology this sensor uses commonly is of the resistive type. The main component of this sensor is a small silicon diaphragm containing four resistors arranged as a bridge circuit. Silicon is the preferred material in the automotive industry due to their effectiveness is sensing pressure. In addition, current silicon micromachining technology can manufacture very thin silicon diaphragms at low cost while not compromising reliability. The bridge circuit requires a power supply from the EMS. Figure 7 shows this:

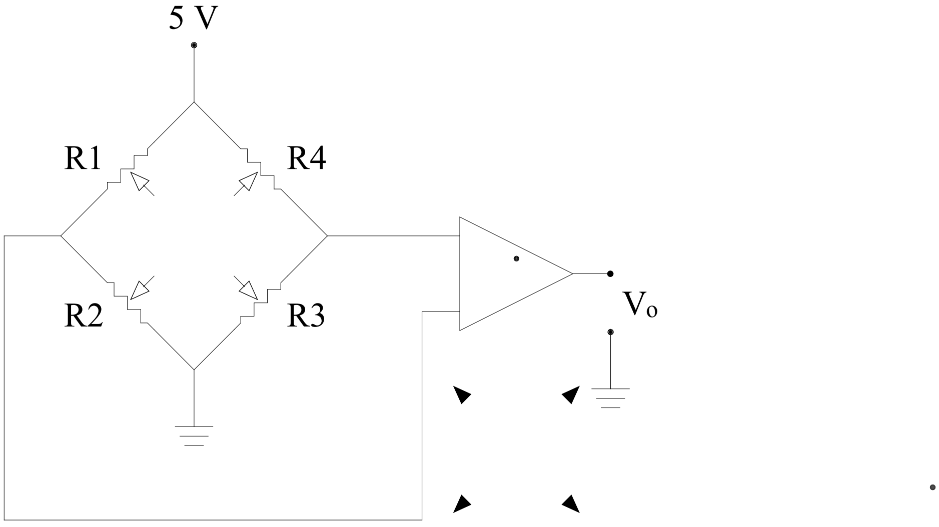


Figure 7: Bridge circuit in pressure transducer.

When pressure bends the diaphragm, the resistance in *R1* and *R3* varies equally, with an opposite variation in *R2* and *R4*. The change of voltage across the bridge flows to an amplifier, which produces a sensor output V_o that is proportional to the pressure. In general, V_o increases linearly as pressure rises.

2.5.3 Temperature Sensor

Similar to pressure, an EMS measures temperature at various locations in, on, or around an engine. An important parameter is coolant temperature to control fuel timing especially during cold starts and monitoring engine temperature in general. Boost temperature measured at the intake manifold helps in the overall determination of intake mass airflow rate. A popular type of temperature sensor used in HDDEs is the thermistor. In comparison to thermocouples, thermistors have high sensitivity when operating at typical automotive type temperature ranges. These sensors work according to the concept of negative temperature coefficient. That means as temperature rises, the resistance of the thermistor sensing element decreases. This is due to the element made out of semiconductor materials. Since an EMS has a readily available voltage source, the thermistor is placed in a voltage divider configuration. The circuit diagram below indicates the positioning of a thermistor relative to the EMS voltage source:

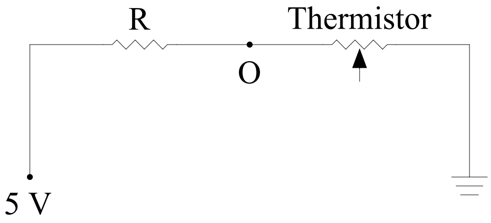


Figure 8: Thermistor with voltage source.

In Figure 8, resistor *R* has permanent resistance, with its value depending on the application.

As an example, when the temperature increases, the resistance in the thermistor falls and the EMS reads the voltage at node *O*. A non-linear relationship exists between the resistance and temperature, thus producing a non-linear voltage output with respect to temperature.

2.6 Summary

Initially, this chapter discussed the differences between analytical models and "black box" approximation methods. The discussion showed that for complex systems that are difficult to model using physics-based principles, it is better to use "black box" tools to create the models. They can achieve good accuracy without spending a lengthy amount of time in the model development phase. Although such "black box" methods can be as simple as linear regression, for most cases it is required to perform multidimensional type modeling to best approximate a system. This is especially true when attempting to model the operation of modern HDDEs mathematically. HDDEs tend to have very complex operations to maximize power output while maintaining low emissions and fuel consumption. One such multidimensional method is through NN modeling. The NN model used in this study is a combination of two popularly used networks but with the addition of online model adaptation. This adaptation capability meant reducing the amount of time it takes to train the networks offline as further training could occur during the implementation phase. The purpose of introducing this adaptive NN was the possibility of using it in a sensor validation scheme. A sensor validation scheme is a fault tolerant technology designed to ensure a system always receive the best sensor values for the purpose of diagnostic and control. The scheme monitors the health of the sensors by continuously comparing its signals to estimated values. When a sensor failure occurs, the scheme terminates the particular sensor producing the erroneous output and replaces it with a best approximation of it. The embedded models provide all the required approximations. Any system operating with sensors and actuators can use a sensor validation scheme to protect itself from incorrectly reacting to a process fault when in actuality the sensor measuring the process failed. Sensors are mechanical devices that are not immune to failures. Common sensors used on HDDEs are of the resistive type that output signals in the form of analog voltages. Sensors measuring speed on the other hand are magnetic in design.

CHAPTER 3 SENSOR FAILURE DETECTION ISOLATION AND ACCOMMODATION (SFDIA)

3.1 Background

The origin of the SFDIA concept has to do with early work on unmanned aerial vehicles (UAV) at West Virginia University (WVU) [4]. The motivation was to curb UAV flight control degradation, lower mission abort rate and lower aircraft loss rate induced by sensor failures. This led to the study detailed by Campa et al. [48] introducing a novel scheme to apply the SFDIA to HDDEs. It focused on proving its functionality using real engine data with one failure mode. In this research, the development went further through an update for handling various failure modes and setting it up for evaluation inline with a test engine. The setup allowed the SFDIA to send and receive signals with the EMS. The test engine was different from the one in the aforementioned study. Performance data were collected to check the SFDIA ability to restore normal operation.

The SFDIA scheme development was conducted within Simulink® [68]. The advantages offered by this high-level system-modeling tool are:

- integration with Matlab®, useful in the signal analysis phase
- availability of xPC Target™ and Real Time Workshop® for automatic synthesis of C code or custom executables from a Simulink diagram
- reduction in computational overhead using features such as the Simulink Accelerator™

Being already in a Simulink environment means quick interfacing with data acquisition hardware for testing in real-time without having to rely on the Matlab interpreter. The availability of all these features to a Simulink model is also one of the novelties of this work.

The upper level of the SFDIA scheme is shown in Figure 9. It consists of three main blocks. The first is the Failure block for introducing failures to the signals at user specified times. Next, the SFDIA Logic block performs the detection, isolation and accommodation tasks. The Approximation block contains the NN models used in the accommodation phase.

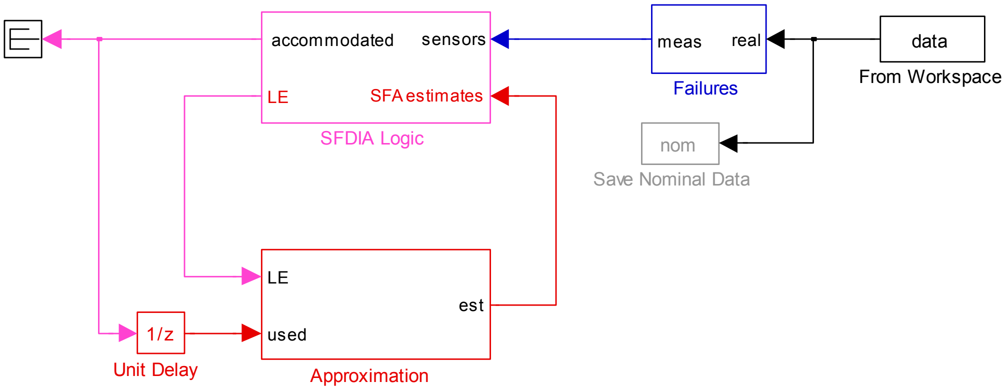


Figure 9: SFDIA Upper level.

3.2 Failure Block

In this proof of concept study, the SFDIA setup assumes failure of only three sensors out of the myriad number of sensors available. The SFDIA is applicable to failures for a larger number of sensors without losing any generality. Specifically, the same general SFDIA structure can handle failures on every sensor. However, a more complex residual generation block is necessary with slight differences to the failure identification logic. The SFDIA scheme studied here can cope with multiple failures. However, a certain amount of performance degradation is expected, depending on the models and type of failures.

A sensor signal can consist of three general components. It is represented as follows:

$$S_n(t) = Y_n(t) + v_n(t) + F_n(t - t_f) \qquad (12)$$

where S is the signal from a particular sensor indicated by n and Y is the true value of the parameter being measured. The noise, v is an additive component inevitably introduced by the measurement. The failure in this effort is a signal F added to the nominal signal and occurs at time t_f.

To test the ability of the SFDIA to accommodate failures, three different modes were selected. The selection is representative of situations such as abrupt and incipient faults (bias or drift). The sensor failure modes below are represented mathematically and as a Simulink model:

a) *Failure Mode 1 (FM1), Step Failure*:

$$F(t - t_f) = \begin{cases} 0 & t < t_f \\ H & t \geq t_f \end{cases} \qquad (13)$$

where H is the amplitude or final value of the failure.

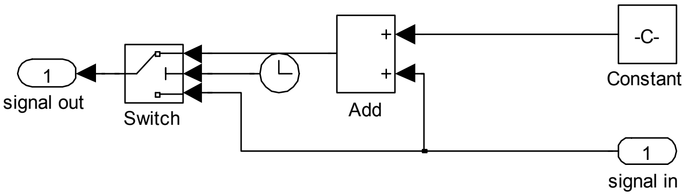


Figure 10: Step failure model.

b) *Failure Mode 2 (FM2), Ramp failure* [69]:

$$F(t-t_f)=\begin{cases}0 & t<t_f\\ H\cdot(t-t_f)/T_R & t_f\le t<t_f+T_R\\ H & t\ge t_f+T_R\end{cases} \tag{14}$$

where T_R is the duration of the ramp.

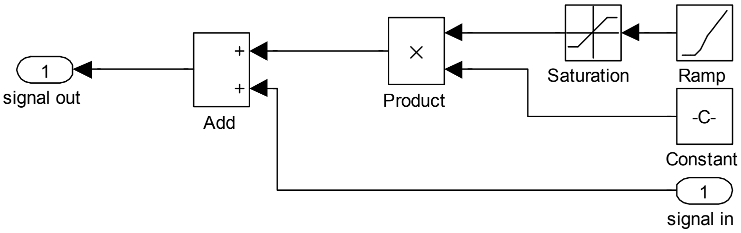


Figure 11: Ramp failure model.

c) *Failure Mode 3 (FM3), Square Wave failure*: The square wave equation is:

$$F(t-t_f)=\begin{cases}0 & t<t_i\\ H & t_i\le t\le t_w\end{cases} \tag{15}$$

where t_i is the initial time of one pulse and t_w is the pulse end. The number of pulses selected was three for purposes of simplicity.

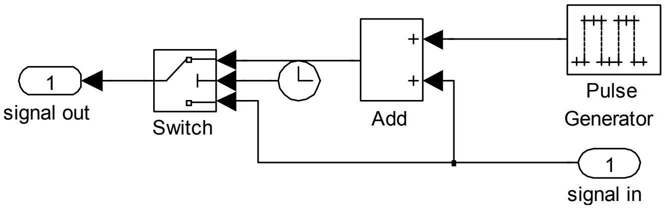


Figure 12: Square wave generator.

Although this mode uses only square waves, it was meant to demonstrate a detection algorithm that is based on a specific condition. The condition is that permanent accommodation triggers after a certain number of pulses. The idea behind this is if the signal appears to fail a few times, it could just be a temporary situation. However, if it happens repeatedly after a specified number of occurrences, then a failure has taken place and thus accommodated. These occurrences can be either close, fairly spaced in time or a combination of both.

All Simulink models in Figure 10 through Figure 12 are located in the subsystems seen in Appendix A. Consistent with Equation (12), Figure 10 through Figure 12 and Figure 75 shows failures can be artificially added to the sensor measurements at user-specified times.

3.3 SFDIA Logic Block

The SFDIA Logic block as seen in Figure 13 has two inputs. The first input consists of signals directly from the sensors. The signals from the faulty sensors are also in this input. The second input originates from the Approximation block. For explaining the SFDIA Logic block, the designation for the signals are as follows:

a) Sensors allocated for failure are S_1, S_2 and S_3; all others are collectively called S_n

b) Signals from the Approximation block are A_1, A_2 and A_3 (approximations of S_1, S_2 and S_3)

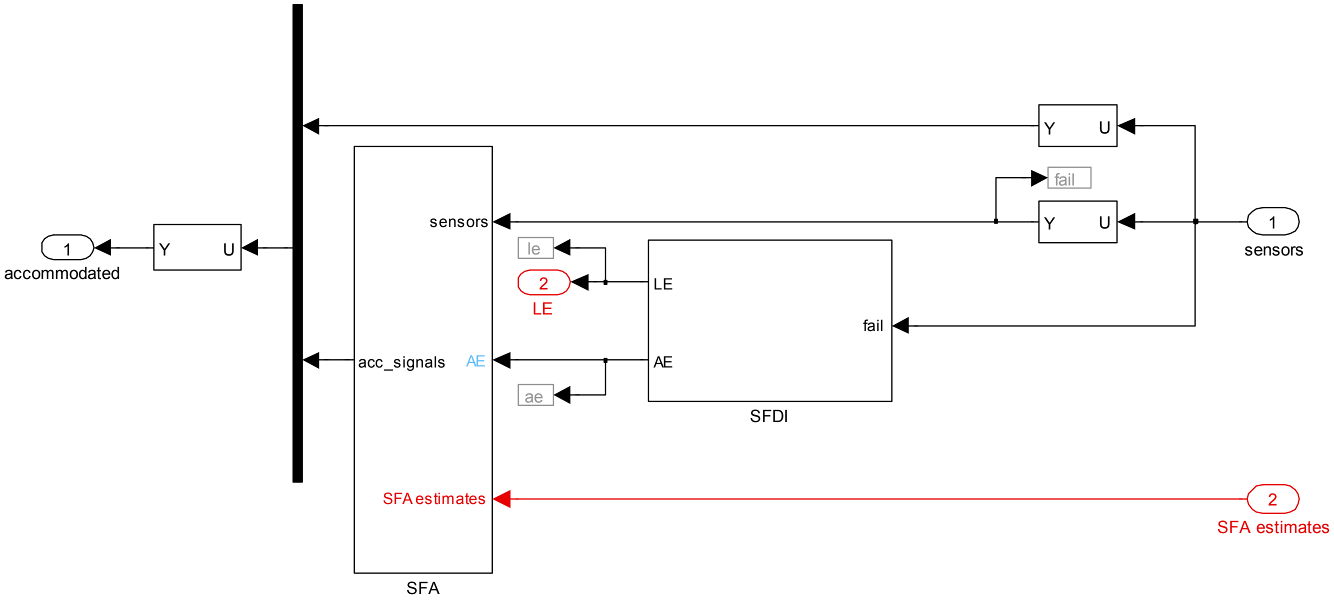


Figure 13: SFDIA Logic block.

The SFDI block in Figure 13 contains the fault detection and isolation logic. It takes as input the signals coming from the sensors and outputs two variables. Each variable is a vector with three values. These variables are called Learning Enable (*LE*) and Accommodation Enable (*AE*). The transition of an *LE* component from one to zero means the scheme has detected a first level of alarm. This is equivalent to declaring a sensor "suspect" and the SFDI block switches off the learning of the NNs to prevent the learning of a possibly corrupt signal. The transition of an

AE component from one to zero is instead the equivalent of declaring the detection and isolation of a fault. The switch block SFA uses the *AE* vector to replace the signal coming from the faulty sensor with its respective estimate provided by the Approximation block. The first variable accommodated will permanently turn off the online learning of all models.

The output of the SFDIA Logic block ensures that an EMS always receives the "values most likely to be correct" of the measured variables. Initially, when there are no faults present, the output vector of the SFDIA is:

$$[S_1(t), S_2(t), S_3(t), S_n(t)] \tag{16}$$

Only signals from sensors and nothing from the Approximation block gets past the SFDIA Logic block. Upon a successful fault detection and isolation, which means S_1, S_2 and S_3 are now the accommodated signals, the output vector of the SFDIA is:

$$[A_1(t), A_2(t), A_3(t), S_n(t)] \tag{17}$$

Therefore, the output of the SFDIA Logic block will contain the accommodated signals, while the other entries are signals coming from healthy sensors.

3.3.1 Detection and Isolation (SFDI) Logic

The key to the detection and isolation logic is through the generation of residual signals. Residuals are an implicit measure of the correlation between measurements and the model outputs for the system under observation. Within this effort, the residual signals generated follow the methods shown in the equations below, where for simplicity purposes the dependence from time has been dropped:

$$\begin{aligned} R_1 &= S_1 - g_1(S_n) \\ R_2 &= S_2 - g_2(S_n) \\ R_3 &= S_3 - g_3(S_n) \end{aligned} \tag{18}$$

The functions g_1, g_2, and g_3 produce NN-based estimates of the S_1, S_2 and S_3 variables. Assuming that these estimates are accurate, then by substituting Equation (12) into Equation (18), each of the three residuals will take on limited values under nominal (no fault) conditions. The magnitudes of R_1, R_2 and R_3 are mainly determined by the maximum deviations of the estimated values from the actual values, and by the intensity of the signal noise. In other words, the better g_1, g_2, and g_3 can estimate the measured values with low noise levels, then the closer the residuals will stay to zero.

Whenever a failure occurs in one of the sensors, only its respective residual will experience an increase of magnitude. The other two residuals will remain within a limited distance from zero. Each sensor failure will generate a clear signature through its residual in Equation (18). Appendix A contains the SFDI block for all the failure modes.

In more detail, the residual vectors are input signals to a couple of low pass filters; a "fast" and a "slow" one. The difference between the two lies in the selection of the bandwidth, as discussed in Chapter 5, thus producing two unique filtered residual vectors. The two bandwidths should not be the same value. The fast filter always has a higher bandwidth value than the one for the slow filter. The output of the fast filter is called "fast residual". Likewise, the output of the slow filter is called "slow residual". Whenever the absolute value of a fast residual entry crosses a predefined threshold, its corresponding value in the *LE* vector is set to zero. This preventively stops the learning of all NNs. Similarly, whenever the absolute value of a slow residual entry crosses another predefined threshold, then the corresponding entry in the *AE* vector is set to zero. When this happens, the switching logic in the Failure Accommodation (SFA) block replaces the signal coming from the faulty sensor with the corresponding estimate in the Approximation block. Figure 14 shows the switching logic:

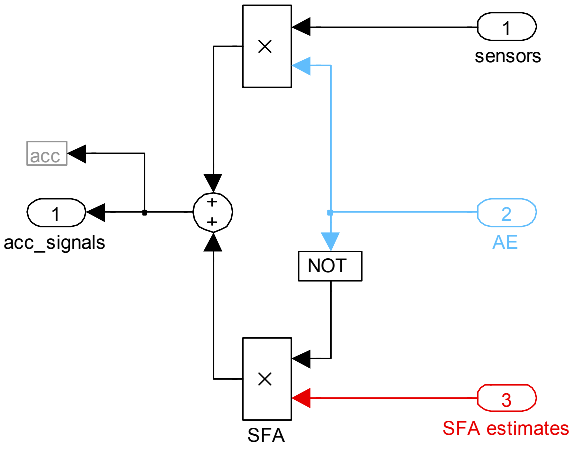


Figure 14: SFA block.

3.3.2 Impact of the Simplifying Assumptions on the SFDI Logic

The SFDI logic outlined above relies on the assumption that only sensors measuring S_1, S_2 and S_3 can fail. In addition, the implicit assumption is that g_1, g_2, and g_3 can adequately approximate the signals using the other sensors not subjected to failures. While those assumptions allow for a great simplification of the SFDI logic, there is no losing of generality because of this. In cases where any sensor could fail, there are ways of generating residual vectors where each component is independent from a specific sensor [70-71]. If that were the case, only the structure of the SFDI block specific to the residual generation would be different.

3.4 Approximation Block

The purpose of the Approximation block is to provide NN approximations for the variables subjected to failure. Inputs to this block are the *LE* vector discussed earlier and the output vector of the SFDIA Logic block. The estimates for S_1, S_2 and S_3 are outputted as a vector shown below:

$$\begin{aligned} A_1 &= f(S_2, S_3, S_n) \\ A_2 &= f(S_1, S_3, S_n) \\ A_3 &= f(S_1, S_2, S_n) \end{aligned} \tag{19}$$

It should be emphasized that the scheme performance is only marginally affected by the NNs online learning capabilities. This adapting feature is important in order to refine their approximations. The ability to refine a model during the online phase means spending a lesser amount of time for offline training. In all cases the training of the NNs largely occurs offline. If there is some confidence that the training data covered all regions of possible behaviors, then the NN learning capabilities could be safely turned off. The same thing applies to the online adaptation where it can be halted once there is nothing left to be learned. Figure 15 shows the contents of the Approximation block.

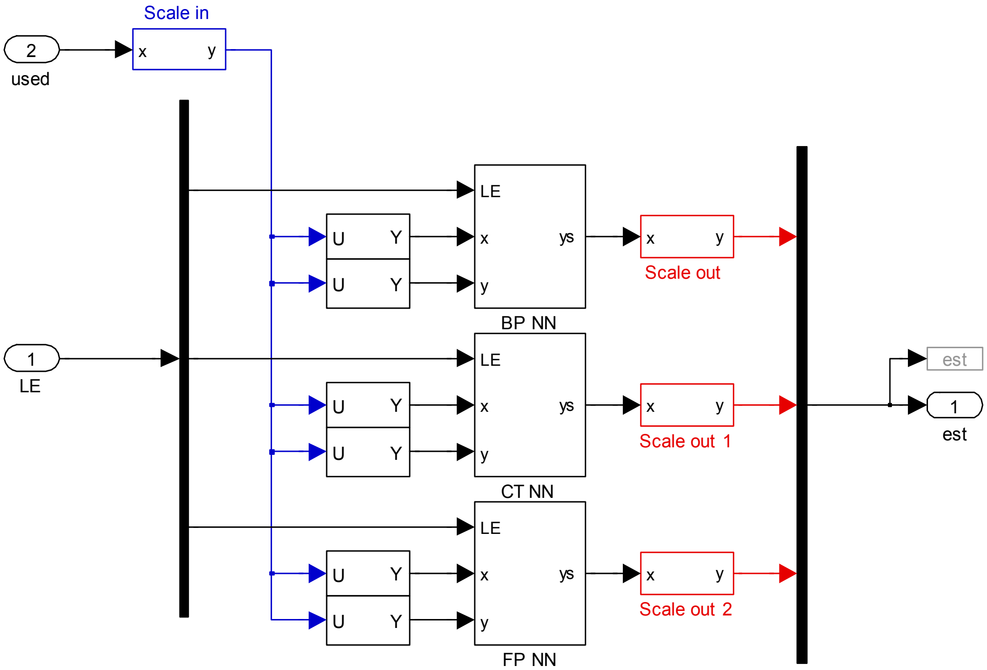


Figure 15: Approximation block.

The figure above is the same design for the NN bank in the Residual Generation block of Figure 76 and Figure 77.

Within the subsystems seen above are the blocks for the ADALINE + EMRAN RBF NN. Figure 16 shows how the blocks receive the inputs of SFDIA Logic block. The switches seen on the left hand side of Figure 16 can turn the NN learning both on and off. Although the ADALINE NN has its own switch to learn online, it remains in the 'off' position following the determination of the weights through the LS method. The nonlinear NN weights account for any variation that might occur. The GRBF block is the EMRAN RBF NN, renamed due to a recent update to the NN library [72].

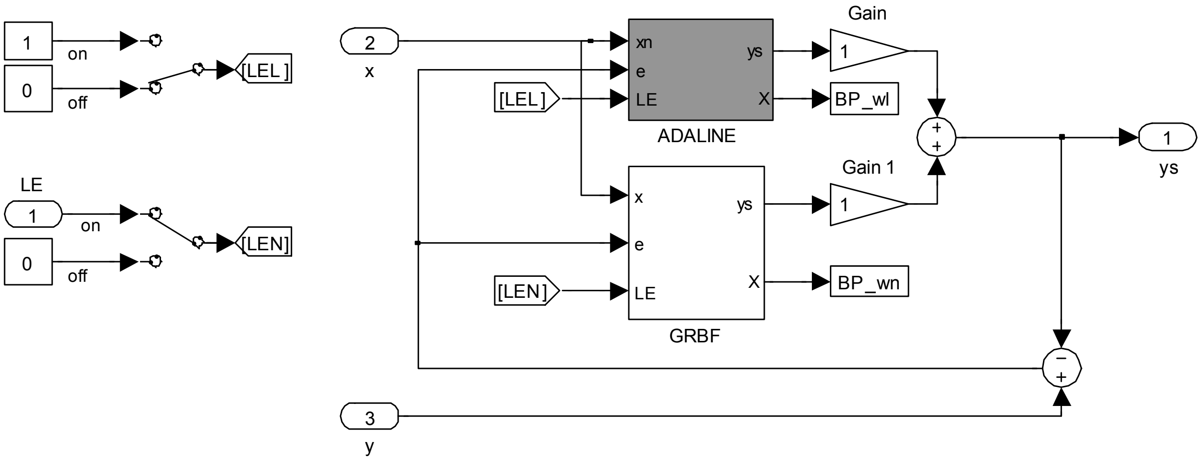


Figure 16: ADALINE + EMRAN RBF NN model for the boost pressure sensor.

The switch related to the GRBF block stays in the 'on' position with the learning controlled by the values of *LE*. The summation of the two NN blocks creates the prediction as seen in Equation (11).

3.5 Approach Limitations

Within the framework of this study, the assumption that the system "behaves correctly" is equivalent to saying that the data for training the NNs represents an accurate system behavior when no failure is present. If the system suddenly "misbehaves" in an abrupt manner, but too soon to be handled by the SFDI algorithm, the NNs cannot provide accurate estimates of the measured variables. This in turn, will result in increased sensitivity to false alarms leading to incorrect failure detection. For example, if a parameter measured by one of the sensors within S_n is not within its nominal value and the alteration is abrupt, it could eventually result in a failure attributed to sensors S_1, S_2 and S_3. Performing failure detection and isolation on every signal might present a solution. Such a scheme might attribute the failure to that specific S_n sensor even if it is correctly working and the offset caused by other unknown factors.

It is important to understand that limitations such as the one discussed above are common to any SFDIA type schemes that somehow rely on an internal system model. For instance, a scheme relying on an analytical model represented by explicit equations will also suffer similar limitations. Alternatively, due to the specific NN and learning algorithm presented in this effort, the proposed scheme offers the very peculiar possibility of adapting itself to gradual changes in the system behavior.

In the case of HDDEs, the EMS contains protection features such that when a fault occurs, it triggers a flag to send the engine into a limp-home mode or shuts it down. For example, in the case of oil temperature, the EMS monitors it to make sure it is within a specified range. When it begins to overheat, it triggers a protection flag, and the EMS reacts accordingly to protect the engine while broadcasting a fault code. As a result, the coolant can also overheat and will affect fuel timing along with the EMS start of combustion algorithm. The start of combustion algorithm is a strategy used to prevent misfires and white smoke during cold starts. This in turn affects the turbocharger to maintain proper boost for given load demand. A reduction in power is likely. These effects can happen very quickly prior to engine shutdown depending on how critical the fault is. In situations such as this, when no sensor failed, the SFDIA will not impede with the EMS' ability for engine protection. This is due to the functionality captured by the NN models where the predicted variable depends on the values of the other parameters. Coolant temperature has a functional dependence on oil temperature. This dependence is higher for HDDEs with an oil cooler in the form of a flat-plate liquid-to-liquid heat exchanger placed within the coolant flow path [73]. This means oil temperature will carry a significant weight within a coolant temperature basis function. Therefore, the residual signal for coolant temperature will remain normal when it is physically overheating and not interfere with the engine protection. On the other hand, if the sensor were to fail, the SFDIA will activate to maintain proper engine operation and prevent the aforementioned effects from taking place.

3.6 Summary

This chapter focused on the concept of the SFDIA and its functionality. It was created entirely in Simulink, thus making it suitable for experimental testing. Various features in Simulink include easy interfacing with data acquisition hardware. Further discussion went into describing its various components. The most important block in the scheme is the SFDIA Logic block that contained all the algorithms to carry out its operation. The current setup enabled the study of up to three sensor failures. However, the general structure of the scheme can handle any number of sensors. To create a scheme to handle any number of sensors would mean altering the method for calculating residuals only as opposed to the one used here. The residual generation served as inputs to the identification and isolation algorithms. When a single failure is identified, the NN learning is all collectively terminated. This prevents the models from acquiring any unwanted characteristics of the faulty signal. If the failure persists, then the isolation algorithm activates and sends a signal to the accommodation logic to provide the EMS with a best approximation of the isolated sensor. The approximations originate from a block containing the NN models. Since the NNs predict a parameter using the other available sensors, it will not affect the engine protection feature within the EMS to take action against a process fault such high coolant temperature or loss of oil pressure. This means the residual related to those parameters will remain within nominal values.

CHAPTER 4 EXPERIMENTAL EQUIPMENT AND PROCEDURES

4.1 Test Engine

A suitable, and readily available, platform to perform the experimental component of this study was the 2004 Mack® MP7-355E engine that complied with 2004 EPA emissions standards. The MP7-355E engine shown in Figure 17 was earmarked for refuse collection and hauling vehicles. It powered the Mack, Volvo®, and Renault® heavy-duty vehicles produced in late 2005. The engine has a high-pressure loop EGR system for NOx reduction. A high-pressure loop meant the EGR gas extraction occurred before the VGT. The EGR system features a cooler that is an air-to-liquid heat exchanger and an oil-actuated EGR valve [67]. The VGT uses a water-cooled sliding nozzle ring actuator to control both intake and exhaust manifold pressures.

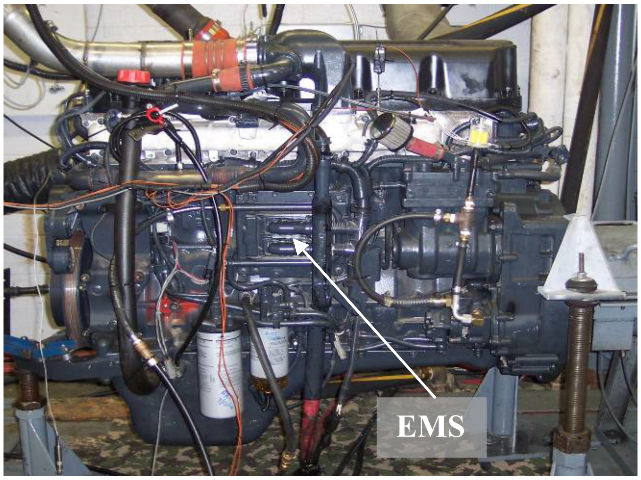


Figure 17: Mack MP7-355E, 2004 Emissions Standard.

This engine has all the necessary features suitable to demonstrate the SFDIA. It is equipped with sensors that are processed by the EMS shown in Figure 18. This EMS is part of the Vehicle Management and Control System IV® (V-MAC IV) [67]. The V-MAC IV continuously monitors itself and the entire vehicle. In the event of a fault, its built-in OBD will activate fault codes that pinpoint the source of the problem. The list of engine sensors used by the V-MAC IV is [73]:

a) Inlet-Air Temperature
b) Inlet-Air Humidity
c) Boost Pressure
d) Boost Temperature

e) Fuel Pressure

f) Water-in-fuel

g) Coolant Temperature

h) Coolant Level

i) Oil Pressure

j) Oil Temperature

k) Oil Level

l) Engine (Crankshaft) Speed/Position

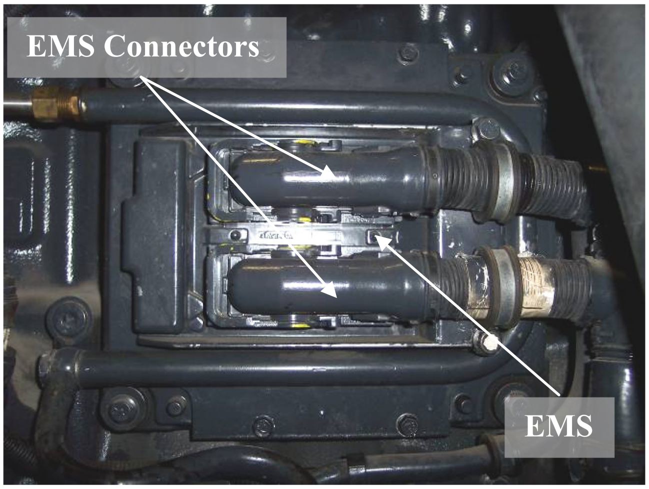
m) VGT Speed

n) VGT Smart Remote Actuator (SRA) Temperature

o) Camshaft Speed/Position

p) Crankcase Pressure

q) EGR MASS Flow Sensor System (separate flow rate and temperature sensors)



`

Figure 18: EMS of the MP7-355E.

Malfunctioning sensors that monitor critical parameters will affect engine performance due to inaccurate signals received by V-MAC IV, even if the components they are monitoring function adequately. For a few mission critical sensors, V-MAC IV has in-built features to handle them in the event they failed. The Engine Protection Warning and Shutdown activates during instances such as low oil pressure, high crankcase pressure, high oil temperature, low coolant level and high coolant temperature. It also has limp-home mode capability that allows continued operation of the engine in the event the vehicle speed, engine speed, camshaft position, pedal position or coolant temperature sensors fail. Depending on the type of failures, the limp-home feature of V-MAC IV will cut the engine power significantly in order to protect the engine. For sensors selected for SFDIA failure analysis, V-MAC IV sets fault codes based on the following thresholds that the signal crosses during in-use conditions [74]:

- Boost Pressure: Voltage less than 0.1 V or more than 4.55 V
- Coolant Temperature: Voltage less than 0.17 V or more then 4.92 V
- Fuel Pressure: Voltage less than 0.3 V or more then 4.75 V

With the SFDIA in place, it is possible to restore the sensor signals to within the above thresholds during occurrences of sensor failures. Thus, maintaining nominal power and emissions by preventing a shutdown or limp-home mode activation. Additional specifications about this engine are available in Appendix B.

4.2 Engine Test Cell

The engine was connected to a General Electric® (GE) motoring/absorbing dynamometer capable of 800 hp for engine speed and load control. The test cell for this facility is located at the Engine and Emissions Research Laboratory (EERL) at WVU. A total-exhaust double-dilution constant volume sampler (CVS) system that can handle engines manufactured to EPA 2007 emissions or later measured exhaust emissions [75]. Gaseous sampling for *NOx* and total hydrocarbons (*HC*) used the analyzer station designed for raw sampling, as seen in Figure 19. As for carbon monoxide (*CO*) and carbon dioxide (*CO_2*), the Horiba® MEXA 7200D in Figure 20 provided these data. Appendix C contains additional information regarding the EERL.

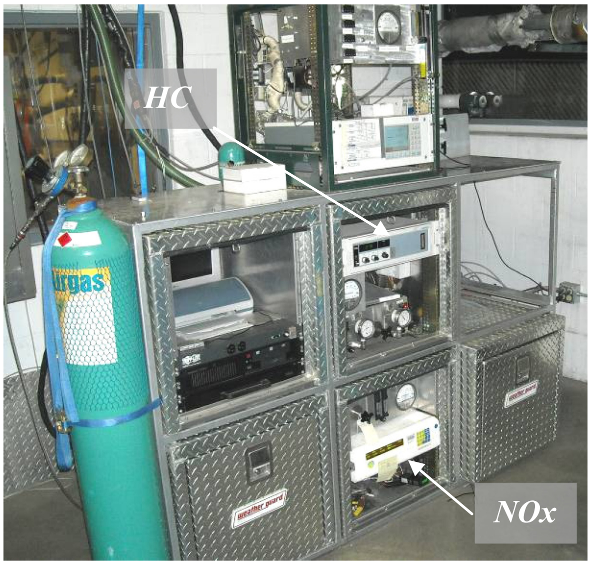


Figure 19: Gaseous analyzers for *NOx* and *HC*.

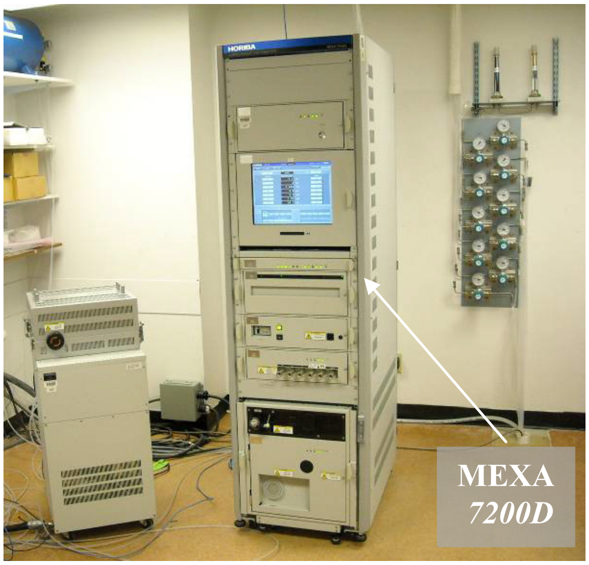


Figure 20: MEXA 7200D.

In Figure 19, the *NOx* analyzer is an Eco Physics chemiluminescent *NOx* analyzer, while a Horiba heated flame ionization detector (HFID) analyzer measured the *HC*. When necessary, the raw exhaust sampling system shown in Figure 19 can hold a separate Horiba non-dispersive infrared (NDIR) CO/CO_2 analyzer. This was the case when acquiring CO/CO_2 data for the analysis in Section 5.4.1 and Section 5.4.2.

4.3 Tasks

4.3.1 Task 1: Sensor Selection and NN Training

The available parameters, and the corresponding engineering units used in this study were:

a) Inlet-Air Temperature (*AT*), ºC
b) Boost Pressure (*BP*), kPa
c) Oil Pressure (*OP*), kPa
d) Oil Temperature (*OT*), ºC
e) Coolant Temperature (*CT*), ºC
f) Fuel Pressure (*FP*), kPa
g) Boost Temperature (*BT*), ºC
h) Engine Speed, Revolutions per Minute (*RPM*)
i) VGT speed (*VTS*), Revolutions per Second (RPS)

Although the selection was based on critical parameters related to fueling and EGR control, the limitation in variables was also due to availability of signals for breakout directly from the EMS. It was not possible to acquire all signals since some are only available through the SAE J1939 Controller Area Network (CAN) bus [76]. Communicating with the CAN bus using Simulink in real-time was possible but would require programming capabilities beyond the scope of this study. In addition, manufacturer participation would be necessary for providing the database to decrypt and translate the specific CAN messages into engineering units, especially the EGR and VGT parameters. The turbocharger speed signal goes directly to the EMS; thus, making the signal accessible without the CAN bus. A Type-T thermocouple provided inlet-air temperature data although it was possible to breakout the signal from the EMS. A magnetic pickup sensor placed at the bottom of the flywheel bell housing gave engine speed data since the buffered *RPM* signal is only available within the CAN bus. Figure 21 shows the engine speed sensor used for the study. The fuel pressure sensor measures pressure from the supply pump. The crankshaft drives the pump via an idler gear that is part of the flywheel gear train.

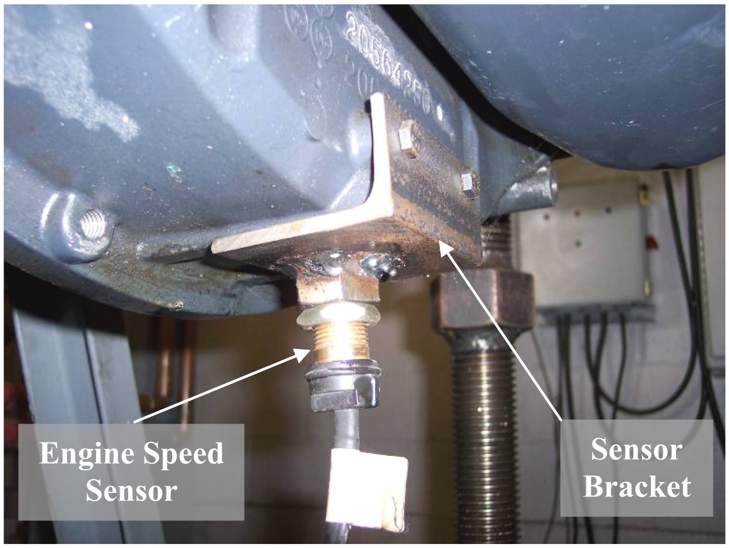


Figure 21: Engine speed sensor for study.

All test engine data for training and validating the NNs were similar to that used by Chew [32], who also provided detailed descriptions of all the duty cycles. The duty cycles simulated both urban and highway-driving conditions that include the cycles developed during the WVU Consent Decree tasks [77]. The assumption was that the data is fault free and covered a wide range of operating conditions. This would provide the scheme with good response time to diagnose and correct failures. As mentioned earlier, variables selected for performing failures are the *BP*, *CT* and *FP* sensors. Appendix D shows the composite duty cycle setpoints and the data of the three variables. Using Equation (19), these sensors are functionally related as follows:

$$A_{BP}(t) = f(AT(t), BT(t), FP(t), CT(t), OT(t), OP(t), RPM(t), VTS(t)) \qquad (20)$$

$$A_{CT}(t) = f(AT(t), BT(t), BP(t), FP(t), OT(t), OP(t), RPM(t), VTS(t)) \qquad (21)$$

$$A_{FP}(t) = f(AT(t), BT(t), BP(t), CT(t), OT(t), OP(t), RPM(t), VTS(t)) \qquad (22)$$

The above models are NN approximations trained for the Approximation block. The first two variables are important for injection timing determination. Proper fuel pressure values received by the EMS ensure normal operation of the engine.

The SFDI block thus monitors the residuals according to Equation (18) as shown below:

$$R_{BP}(t) = BP(t) - g_{BP}(AT(t), BT(t), OT(t), OP(t), RPM(t), VTS(t)) \qquad (23)$$

$$R_{CT}(t) = CT(t) - g_{CT}(AT(t), BT(t), OT(t), OP(t), RPM(t), VTS(t)) \qquad (24)$$

$$R_{FP}(t) = FP(t) - g_{FP}(AT(t), BT(t), OT(t), OP(t), RPM(t), VTS(t)) \quad (25)$$

where $g(t)$ are the NN estimates of *BP*, *CT* and *FP* using the other six sensors. Regardless of how good the estimates are, the values of *R* will retain some initial deviation. When a failure is introduced, the residual of a failed signal will begin to change. When this variation crosses a specified threshold that activates the accommodation logic, the SFDIA will replace the failed sensor values with the ones calculated by Equation (20) through Equation (22). The next chapter shows the model performances and the thresholds selected based on offline simulation of the SFDIA.

4.3.2 Task 2: Wiring Setup for Data Acquisition and Breakout Panel Verification

To access the sensors on the EMS involved making breakout panels for use with the EMS. Each panel breaks out the signals on one connector on the EMS. Figure 22 and Figure 23 show the breakout panels along with their placement with respect to the engine. The interface between the panels and the EMS used genuine Volvo EMS parts to ensure a robust connection. In addition, all wires were soldered to their respective terminals, both on panels and EMS parts.

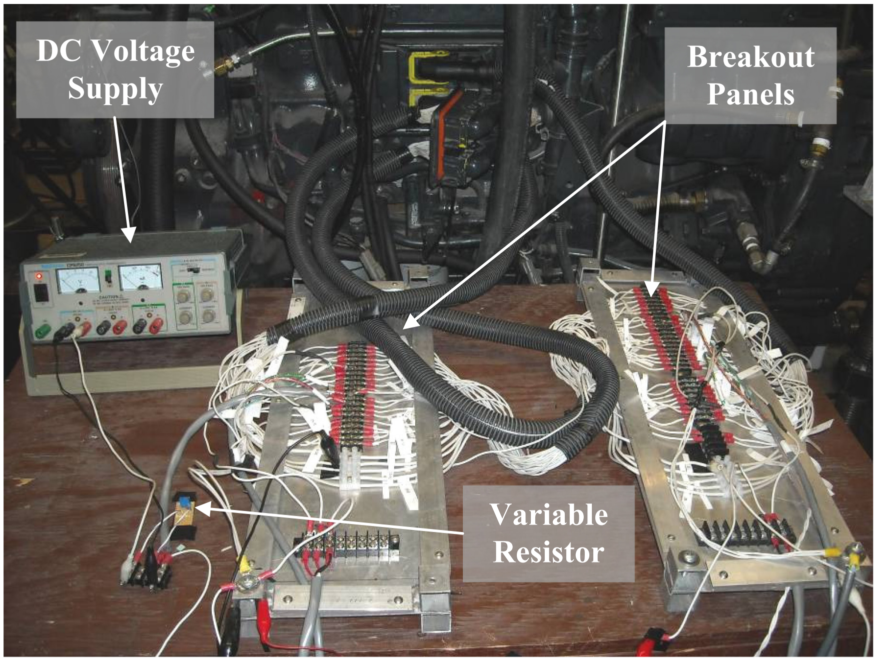


Figure 22: Breakout panels for the MP7-355E.

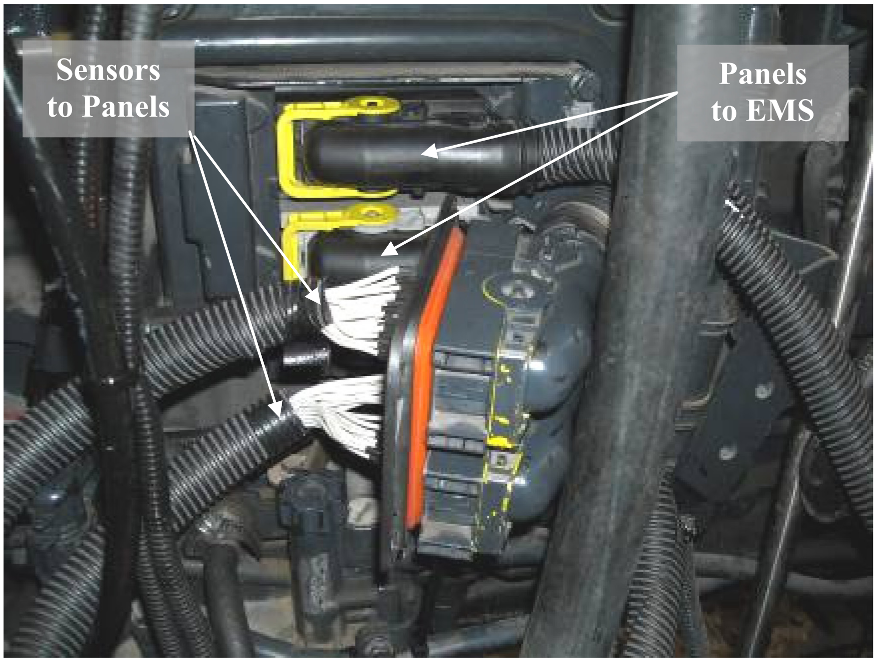


Figure 23: Interface between EMS and breakout panels.

Before continuing work on the DAQ system, it was critical to ensure the panels do not interfere with normal engine operation. Engine test data compared with and without the panels could determine whether there were any differences in performance. The duty cycle chosen was the Federal Test Procedure (FTP) cycle for heavy-duty diesel engines, which is a transient cycle described in Title 40 of the Code of Federal Regulations (CFR) Part 86, Subpart N [75, 78]. All results from the verifications are in Chapter 5.

With the panel verification completed, the DAQ system was assembled. The power supply unit in Figure 22 provided DC voltage power to the *CT* sensor via a variable resistor. The use of the breakout panel method involved cutting off the voltage to the thermistor. Reactivating the sensor meant using an external power supply and, as Figure 8 illustrates, determining the value of the permanent resistor *R* located inside the EMS. The variable resistor installed in the setup helped in determining this resistance through a tuning process. The reference points used to set the resistance value were determined when the EMS read the ambient room temperature prior to a cold start and it correctly indicated the thermostat opening at 82ºC [79].

A 3B Signal Conditioner manufactured by Analog Devices® (ADI) then received all the sensor signals used by the SFDIA. It conditioned signals from the panels, the intake-air thermocouple, engine speed sensor and VGT speed sensor with the use of 3B modules featuring noise filtering and magnetic signal isolation [80]. The output of the conditioner then went to two

National Instruments™ (NI) breakout boards. The use of two data acquisition (DAQ) cards - PCI-6024E and PCI-MIO-16E-1, both made by NI - required the two NI breakout boards. The cards were placed in the data acquisition computer manufactured by Dell™. The connection between the breakout boards and DAQ cards used NI 68-pin shielded cables.

The signals leaving the conditioner are all single-ended, analog voltages. Only the *BP*, *CT* and *FP* sensors went through the DAQ system before reaching the EMS. The rest were wired for reading the sensor voltages only. The purpose for using two DAQ cards was to have three analog output channels. The output channels also underwent conditioning using 7B output modules by ADI. The isolation modules in essence compensated for any ground loop effects caused by the EMS and ensured robust output signals. Therefore, grounding for the 3B input and 7B output signals was connected directly to the EMS ground channels. On the output side of the 3B and inputs to the 7B modules, their respective grounding was to their power supply ground. Figure 24 shows a schematic of the entire test setup in relation to the breakout panels. Appendix E contains information about the various devices used as part of the DAQ system.

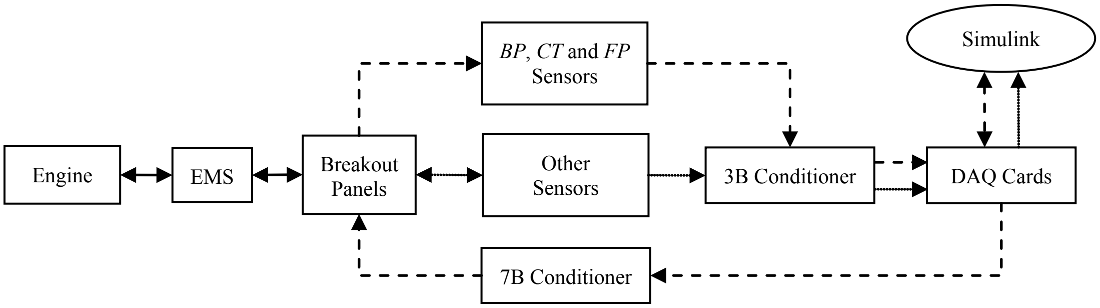


Figure 24: Setup to receive and output the three sensor signals for failure analysis.

4.3.3 Task 3: Configuring SFDIA for Data Acquisition

The signals from the 3B conditioner were all analog voltages. The NN models required data in engineering units. For the six engine sensors (not including *VTS*), the steps taken to obtain the calibration equations were as follows:

a) Disconnect the sensor output wires on the breakout panels, leaving the wires that go the EMS intact

b) For any one of the sensors disconnected in the previous step, connect a DAQ analog output channel in its place

c) Output a voltage signal from the DAQ card and record parameter values from VAT2000 [81], the Volvo software that reads the EMS values via the CAN bus

d) Repeat the above steps for the other five sensors

A DS345 Synthesized Function Generator made by Stanford Research Systems meant to replicate the output of a magnetic pickup sensor was used to calibrate both the speed sensor 3B modules. The device for calibrating the inlet-air temperature 3B module was the Fluke® 714 Thermocouple Calibrator. With the calibrations done, it was possible to restore the wiring connections as discussed in Section 4.3.2 and the equations placed in the SFDIA block called Input Conversion. In the block, it shows the engine sensor inputs divided in half. That was due to the 3B module multiplying the sensor signals by two. With the same data, it was also possible to convert *BP*, *CT* and the *FP* values into voltages with the block called Output Conversion.

With the DAQ system completed, the SFDIA needed configuring to communicate with the DAQ cards for receiving and outputting signals. The Data Acquisition Toolbox™ was a useful tool since it has analog input/output blocks that already recognize the DAQ cards in the computer. Considering that the operating system is Windows® XP, it was not possible for hard real-time testing, as it is a multitasking environment. To achieve soft real-time testing, first Windows needed to be configured to run at a bare minimum. This meant shutting down as many processes as possible leaving only the essential programs for this work. The next solution involved using a custom Simulink block created by Daga [82] called RTBlock. It prioritizes Simulink within Windows for real-time data acquisition. The concept behind this block is if Simulink is to run with real-time temporization, the cycle time should be lower than the desired simulation step. Cycle time means the time Simulink needs for calculating a simulation step, which is a function of the hardware and the operating system running Simulink. Performance comparisons of the block with respect to xPC Target and Real Time Workshop showed similar qualities in providing real-time data acquisition [82]. During testing, the VAT2000 software collected engine data on a Dell laptop computer to confirm the signals seen by the EMS. Appendix F shows the SFDIA scheme configured for data acquisition along with the correct method for using the RTBlock. In addition, Appendix F shows all calibration results along with the Input and Output Conversion blocks. By having the SFDIA as a subsystem shown in Figure 25, it allows the RTBlock to control the execution of the SFDIA.

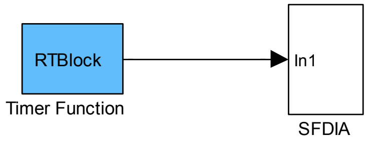


Figure 25: RTBlock with the SFDIA subsystem.

4.3.4 Task 4:Real-time SFDIA Testing

At this point, the NN models for the accommodation and residual determination were ready for implementation. The duty cycle in this task focused on the FTP cycle. All data collected were from Simulink, the engine test cell and VAT2000. The task consisted of a few stages mentioned below:

1. *Check DAQ system and SFDIA tuning*: The computer was checked to make sure it was communicating with the DAQ cards and functioning properly within Simulink. After that, it was necessary to inspect the breakout panels for any loose wiring. Figure 26 shows the completed test setup. During this time, the value of *H* in Equation (13) through Equation (15), including the thresholds for *LE* and *AE*, are now determined based on offline simulation observations. The basis for the selections was to show the SFDIA functionality and not specifically follow the thresholds mentioned in Section 4.1.

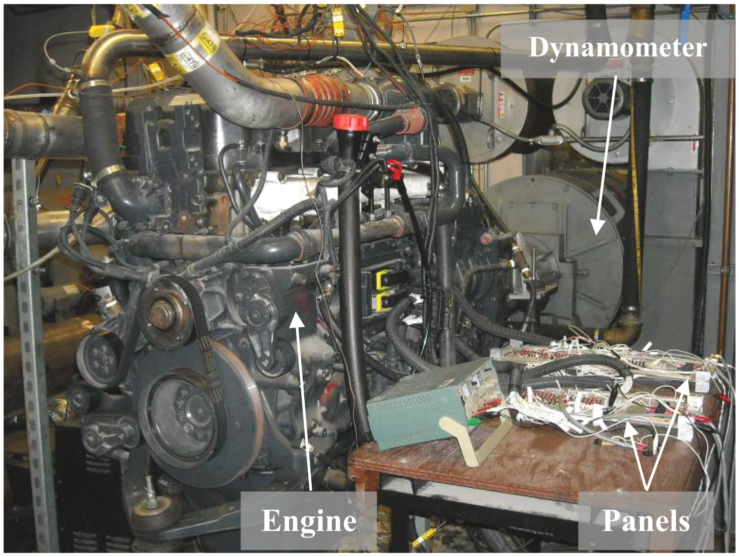


Figure 26: Test engine with dynamometer and breakout panels.

2. *Testing SFDIA without accommodation mode*: Engine map cycles were carried out to create the FTP setpoints. Then, to assess the effects of a failed sensor on the test engine, the *BP* sensor signal was subjected to FM1 without accommodation and performance data recorded. Next was to obtain baseline data, meaning test the SFDIA without inducing failures.
3. *Real-time SFDIA testing with accommodation mode*: With the availability of a baseline data set, the SFDIA test with accommodation enable could commence. All failure modes were tested with the failures occurring at specified points in time. Data collected were those from Simulink, VAT2000 and the laboratory.

CHAPTER 5 RESULTS AND DISCUSSION

Initially, this chapter focuses on models trained for the study, both for accommodations and residuals. Next, the SFDIA offline simulation is discussed, which helped in the selection of suitable thresholds and other constants mentioned in the previous chapter. Finally, results are presented from the real-time testing including the effects on the engine when the *BP* sensor fails.

5.1 NN models

5.1.1 Approximation NNs

Two sets of non-overlapping data provided the training and validating datasets for model creation. The datasets originated from VAT2000 through previous testing discussed in Section 4.3.1. The duration of the training data was 3.6 hours, while the validation data was 1.17 hours long, both sampled at 5 Hz. As a first step, weights for the ADALINE NN were obtained. The weights for the *BP*, *CT* and *FP* sensors calculated using the LS method is listed in Table 1.

Table 1: ADALINE weights for Approximation block NNs.

Parameter	*BP* weights	*CT* weights	*FP* weights
AT	0.003	-0.058	0.025
BP		-0.073	-0.057
OP	0.016	-0.119	0.427
OT	-0.130	0.411	0.221
CT	-0.019		-0.017
FP	-0.030	-0.033	
BT	0.247	0.339	-0.215
ES	-0.304	0.070	0.326
VTS	1.165	0.242	0.136

The table above also indicated the importance of a given variable to the overall prediction. The larger the magnitude, the more influence it has on the model. With the linear weights obtained, the focus moved to training the nonlinear NN portion. For this task, it was not necessary to train the EMRAN RBF NN within the SFDIA scheme itself, although it is possible to do so. Therefore, the training of the subsystem seen in Figure 16 took place within the following Simulink model to obtain all predictions:

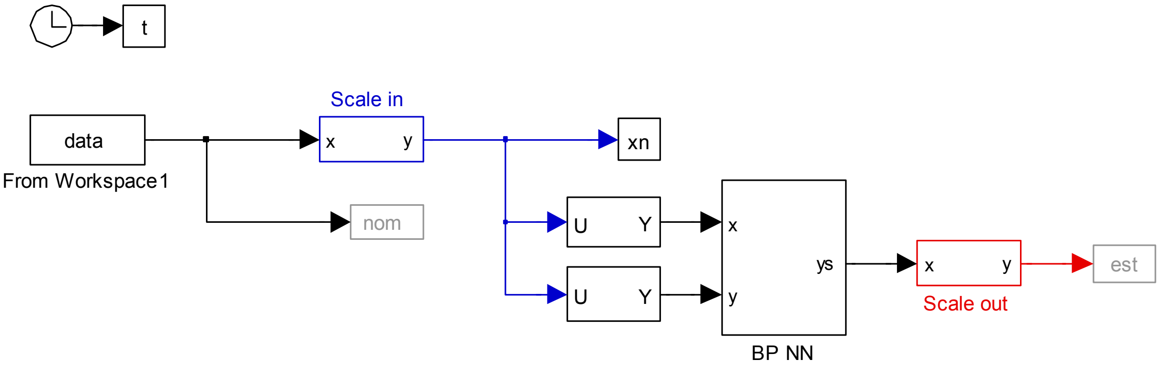


Figure 27: Simulink model for training and validating the *BP* prediction.

Using the above model, the Switch block for the GRBF block now resembles the one in Figure 28 . It was set to one during training and to zero during validation.

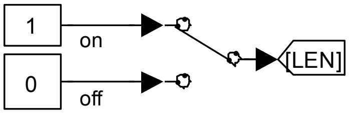


Figure 28: Simulink Switch block used during EMRAN RBF NN training.

Training the RAN network involved varying the parameters E_1, E_2 and λ, while running the scheme several times for a given combination. For each run, the scheme creates a set of values for θ from Equation (2). The values for E_3 and η were set to 0.01 and 1x10^{-6}, respectively. A small learning rate is critical to ensure the model learns as much as possible about the system, but enough to avoid learning measurements from slow drifting fault. The RMS percentage error, seen below normalized for the given data set, was calculated at each run until it stopped varying noticeably.

$$RMS\ \ percentage\ error = 100 \cdot \frac{\sqrt{\frac{1}{N} \cdot \sum_{t=1}^{N} (y(t) - \hat{y}(t))^2}}{max_t(y(t)) - min_t(y(t))} \qquad (26)$$

where $y(t)$ are nominal values. When the training for a set of E_1, E_2 and λ concluded, then the RMS error of the model was calculated with the validation (nominal) data set. This error value provided an indication of the model performance. The training process continued until the RMS error from the validation was at a minimum and did not significantly vary. The error analysis also served as a Quality Control-Quality Assurance (QC-QA) check for the SFDIA scheme. In general, variables that varied were within the following ranges:

Table 2: Range of possible values for the EMRAN RBF NN constants.

Parameter	Range
E_1	0.1 - 0.3
E_2	0.2 - 0.8
λ	0.1 - 1

Models selected for this study performed as seen in Figure 29 to Figure 31.

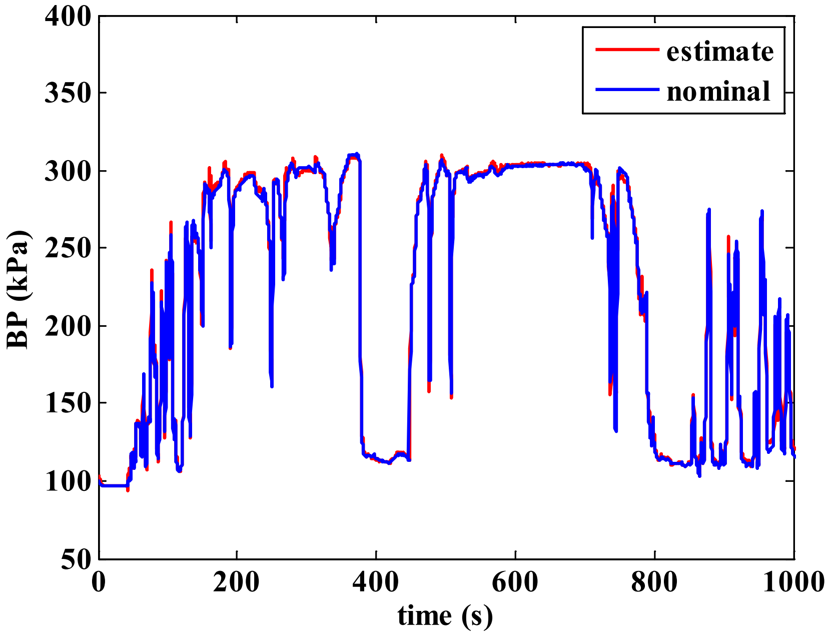


Figure 29: Performance of NN in estimating *BP*.

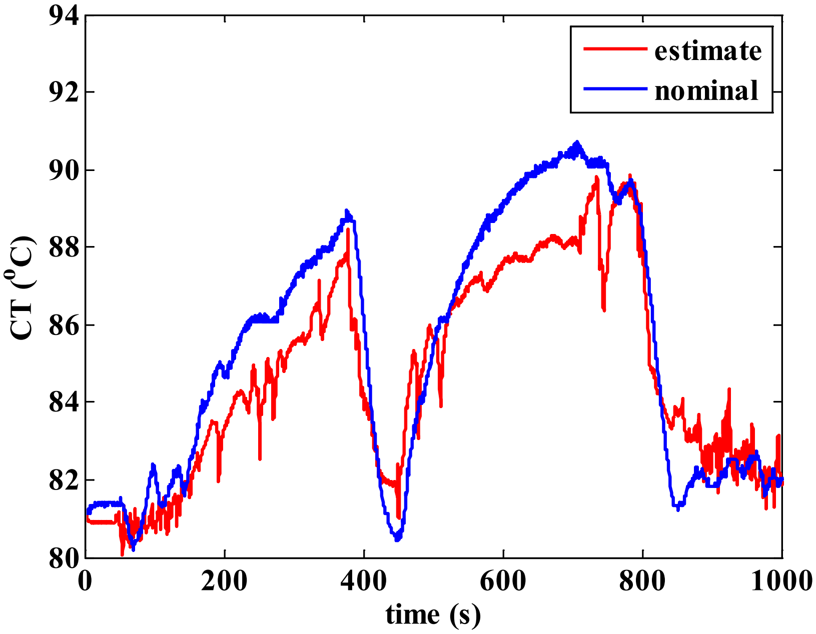


Figure 30: Performance of NN in estimating *CT*.

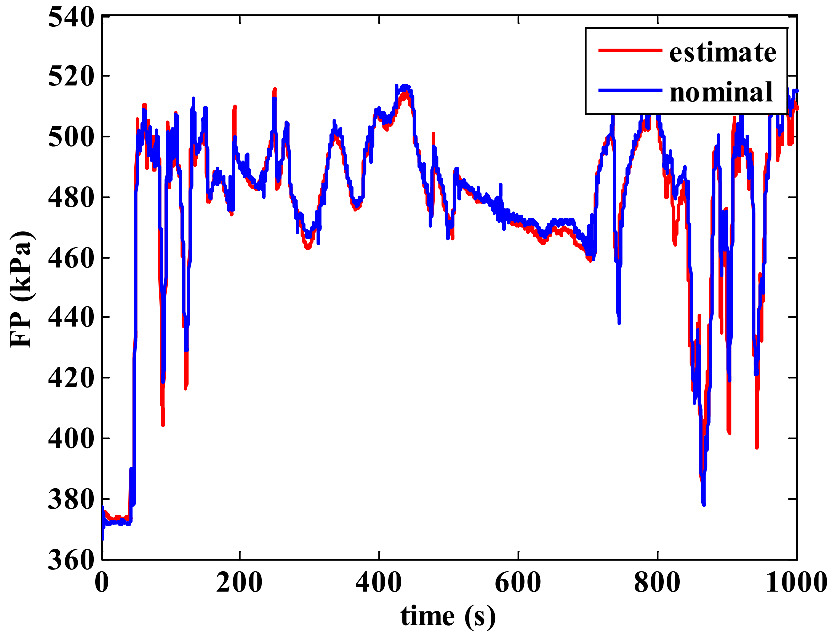


Figure 31: Performance of NN in estimating *FP*.

Appendix G indicates the values of the nonlinear NN constants E_1, E_2 and λ for each model along with the number of neurons. Visually, both pressure models that produced Figure 29 and Figure 31 had the best fits. However, the *CT* model performance seen in Figure 30 showed that more input variables were necessary to provide a better estimate. As an example, if EGR gas temperature (*ET*) available only through the inaccessible CAN bus was included, then the linear NN weight distribution is:

Table 3: ADALINE weights for *CT* model with EGR temperature included.

Parameter	*CT* weights
AT	0.003
BP	-0.171
OP	-0.229
OT	0.250
FP	-0.145
BT	0.132
ES	0.211
ET	0.487
VTS	0.206

The variable *ET* would carry a high weight since the EGR cooler on the MP7-355E requires a high volume of coolant flow to maintain the gas temperature. Weights in Table 3 also show that the addition of *ET* changes the weight distribution as some variables carry a higher weight and could improve the overall model accuracy by adding more neurons. The oscillations in the nominal *CT* data meant there might be other unknown factors such as signal noise that influenced the NN performance. The RMS errors for all the models after validation were:

Table 4: Summary of model performances.

Parameter	RMS Error (%)
BP	1.46
CT	8.13
FP	1.84

5.1.2 Residuals NNs

The training and validating procedures to obtain the residual NNs were as discussed in the previous section, except the models used fewer input variables. The input variables selected are as described in Equation (23) to Equation (25). Initially, the ADALINE weights were determined and listed in Table 5.

Table 5: ADALINE weights for the residual estimation.

Parameter	*BP* weights	*CT* weights	*FP* weights
AT	0.003	-0.059	0.025
OP	0.006	-0.134	0.429
OT	-0.145	0.414	0.222
BT	0.248	0.329	-0.235
ES	-0.316	0.082	0.342
VTS	1.160	0.154	0.067

With the linear NN weights known, the EMRAN RBF NN training then took place with the resulting model performances shown in Figure 32 to Figure 34:

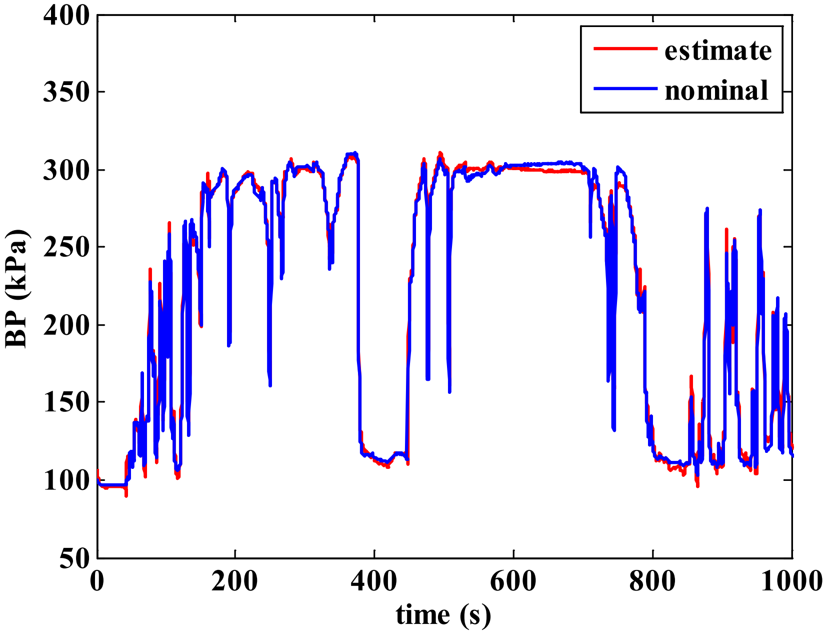


Figure 32: Performance of NN in estimating *BP* for the residuals

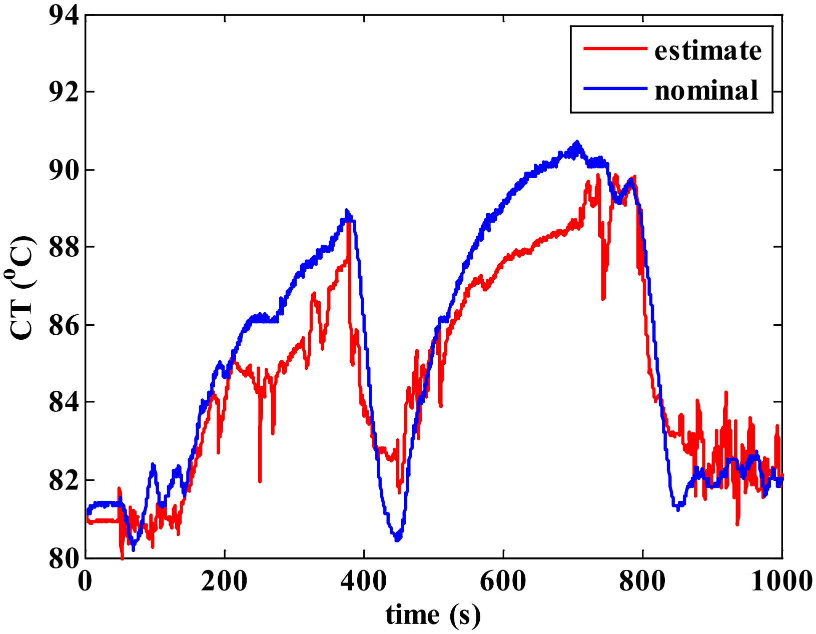


Figure 33: Performance of NN in estimating *CT* for the residuals

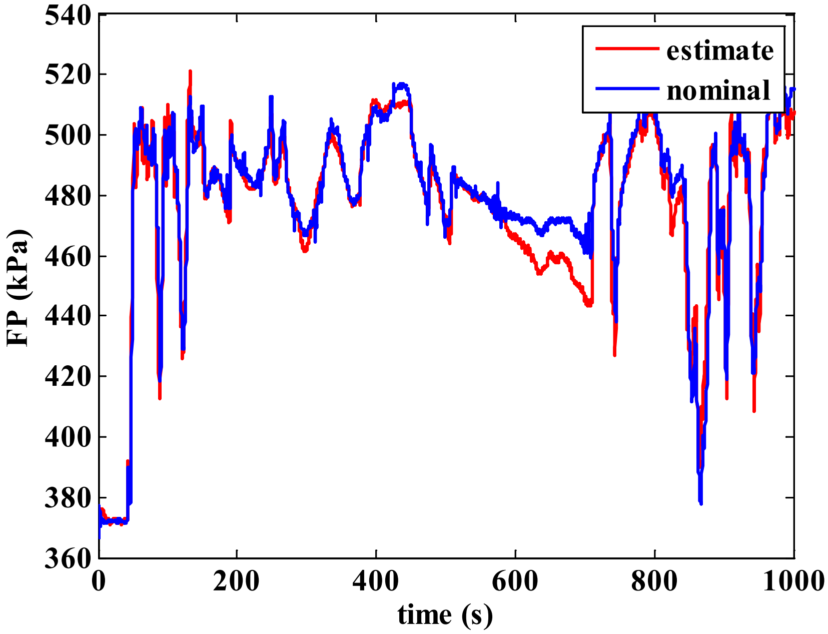


Figure 34: Performance of NN in estimating *FP* for the residuals.

The EMRAN RBF NN parameters that created Figure 32 to Figure 34 are in Appendix G. The model performances in Figure 32 and Figure 34 showed a reduction in accuracy from a visual observation in comparison to their Approximation block counterparts. In addition, both pressure models showed better predictions than the one obtained for estimating *CT*. The RMS errors calculated for the residual NNs were:

Table 6: Summary of model performances used for residual determination.

Parameter	RMS Error (%)
BP	2.11
CT	7.93
FP	2.46

The aforementioned reduction in accuracy of the *BP* and *FP* models may be due to the use of fewer input variables. However, a reduction in input variables gave the *CT* model a 2.65% improvement over the Approximation block *CT* model. It is noted that the residual *CT* model has 65 additional neurons that likely contributed to the result in Table 6. Although this is a minor difference in RMS errors, it is possible that reducing the number of variables can improve the overall accuracy. Another important consideration is that increasing the number of neurons can also amplify an estimation error [48]. The *BP* and *FP* models, in addition to the reduction in input variables, did have a higher number of neurons than the ones for the Approximation block.

It is understood that ADALINE weights provide just an estimate of how well each variable contributes to the overall models. The variables omitted from all three residual NN models had small weight values in Table 1. This would imply removing them would not significantly affect the model performances. A comparison of the residual NN weights in Table 5 with their equivalent Approximation block NN weights listed in Table 1 did show the following differences in Table 7.

Table 7: Residual NN weights in comparison to the Approximation block weights.

Parameter	Difference (%)		
	BP weights	*CT* weights	*FP* weights
AT	0.00	1.69	0.00
OP	-166.67	11.19	0.47
OT	10.34	0.72	0.45
BT	0.40	-3.04	8.51
ES	3.80	14.63	4.68
VTS	-0.43	-57.14	-102.99

In addition, the EMRAN RBF NN structure had a unique effect on all the models. Being a "black box" method, that overall structure is not quite apparent. This sometimes leads to difficulties in making conclusions on overall model performance, as seen in the pervious paragraph. Therefore, the differences in the ADALINE NN weights, and the EMRAN RBF NN configuration collectively had a unique effect on the RMS errors indicated in Table 4 and Table 6.

5.2 Tuning the SFDIA for the MP7-355E Test Engine

With models acquired, the final assembly of the SFDIA took place. All the NN weights were in files that the SFDIA accessed each time the scheme initialized. The SFDIA scheme was now run a few times using engine data to ascertain the appropriate bandwidths for the low pass filters that computed the fast and slow residuals. At the same time, thresholds that control values of *LE* and *AE* were determined. For this tuning, the learning switch for the GRBF block (see Figure 16) was set to 'off'.

The starting values for determining the low pass filter bandwidths were the ones used in the previous study [48]. To trigger the accommodation mode, the chosen value of *H* used by the failure models was 50% of the data range shown in Figure 84 to Figure 86 (see Appendix D), respectively. The choice of this selection was due to simplicity. The selection of *H* could take on any values but with appropriate consideration of the engine manufacturer thresholds assigned to the sensors. For instance, failure artificially induced on the *CT* sensor should not exceed 219^{o}F. Anything at this temperature or higher triggered the engine protection mode, resulting in a shutdown. This happens, as the SFDIA is not part of the current EMS programming. Having the engine shutdown for such a situation during experimental testing was undesirable. All failure amplitudes were positive, meaning the failures caused the signals to increase in value. The presence of Absolute Value blocks in the path of the residual signals seen in Figure 76 and Figure 77 (see Appendix A) meant that it does not matter whether failures are positive or negative in value.

As mentioned earlier, the tuning involved simulating the SFDIA repeatedly and the engine data used was the NN validation dataset. The reason for tuning the SFDIA scheme was to allow it to function with the test engine sufficiently. The failure mode used for this tuning process was FM1. The process that was followed for the tuning is listed below:

a) Set values for the low pass filter bandwidths; the fast filter bandwidth variation was between 0.5 and 1 rad/s, while the slow filter bandwidth was between 0.1 and 0.5 rad/s
b) Run the SFDIA without failures and record the fast and slow residuals
c) Obtain the maximum values of the fast residual vectors
d) Multiply the above values by a factor between one and two to obtain thresholds for both the *LE* and *AE* vectors; initial values being the ones used by Campa et al. [48]
e) Run SFDIA with FM1 applied to *BP*, *CT* and *FP* signals at specified times and visually observe results; for this study, selected failure times were 200s, 300s and 400s respectively
f) Repeat steps a) through d) until SFDIA responds adequately

Table 8 summarizes the results from the tuning process. It was possible to use one factor value mentioned in step (d) on all three fast residual vectors. The fast filter bandwidth from the previous study remained unchanged. Selection of the slow filter bandwidth was critical since its value affected the time it took for the accommodation mode to respond to a failure. The smaller the value, the quicker it accommodates but it could be susceptible to false alarms. A bigger number means the accommodation might take longer to happen. Appendix G shows additional data such as the factors determined in step (d) that led to calculating *H* and the thresholds.

Table 8: SFDIA settings.

Parameter	Filter Bandwidth (rad/s)		*H*	Thresholds	
	Fast	Slow		Fast	Slow
BP	1	0.3	107.085 kPa	34.851 kPa	38.020 kPa
CT			5.630ºC	3.432ºC	3.588ºC
FP			87.534 kPa	53.370 kPa	55.796 kPa

All parameters associated with the NN models and SFDIA scheme up to this point are unique to MP7-355E engine under study. Another engine or system might produce a completely different set of values, while still applying the same ranges in Table 2 and the tuning process mentioned in the previous paragraph. With the configuration for the test engine completed, it was now ready for an offline assessment of the SFDIA.

5.3 SFDIA Simulation Results

This section presents the SFDIA scheme results acquired through offline simulation using all failures modes of this study. Performing a simulation study of the SFDIA is an important step prior to the real-time test as it gave an overview of the scheme functionality. In summary, performance of the NN estimators in the accommodation and residual generation blocks are summarized in Tables 4 and 6. All thresholds and failure amplitudes are as described in Table 8 and updated within the SFDIA. Although the *LE* vectors change values, the Switch block associated with the GRBF block was set to the 'off' position (see Figure 16) to preserve the number of neuron and prevent the θ variables from changing values. All the induced failures occurred at user specified times. For each of the failure modes, the failure points were:

Table 9: SFDIA failure points.

Parameter	Failure Time (s)		
	FM1	FM2	FM3
BP	200	200	200
CT	300	400	300
FP	400	600	400

The intention was to have failures occur early on due to the length of the FTP cycle. Being a 1200-second cycle, it was the desire to have the SFDIA operation in the accommodation mode as much as possible to acquire sufficient data to determine the scheme performance. For all the failure modes, *BP* was always the first sensor to induce a failure. The *CT* sensor was next, followed by *FP*.

5.3.1 SFDIA with FM1

This section discusses the SFDIA performance when subjected to a step failure mode from an offline standpoint. The failures occurred at the time instants indicated in Table 9. Figure 35 to Figure 37 shows the SFDIA performance subjected to FM1 mode:

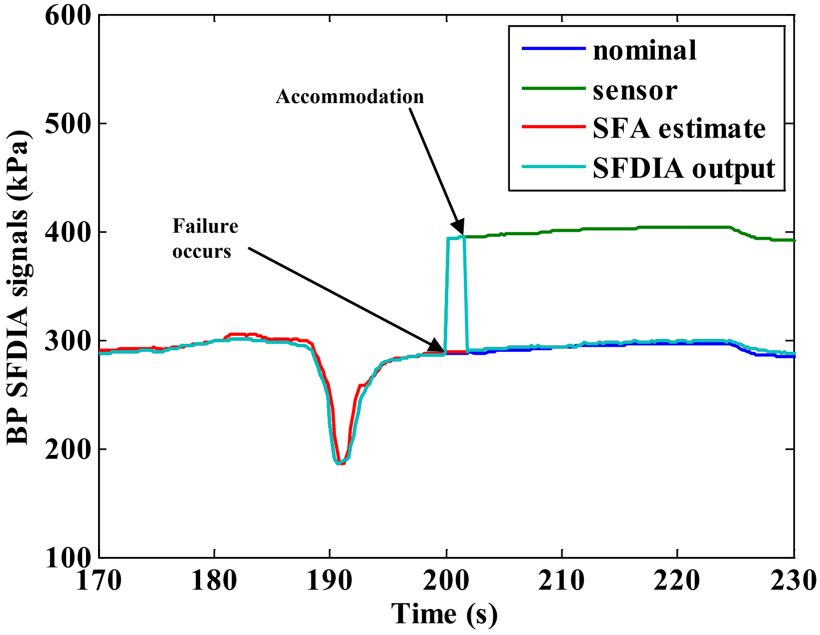


Figure 35: *BP* signal subjected to FM1 at 200s.

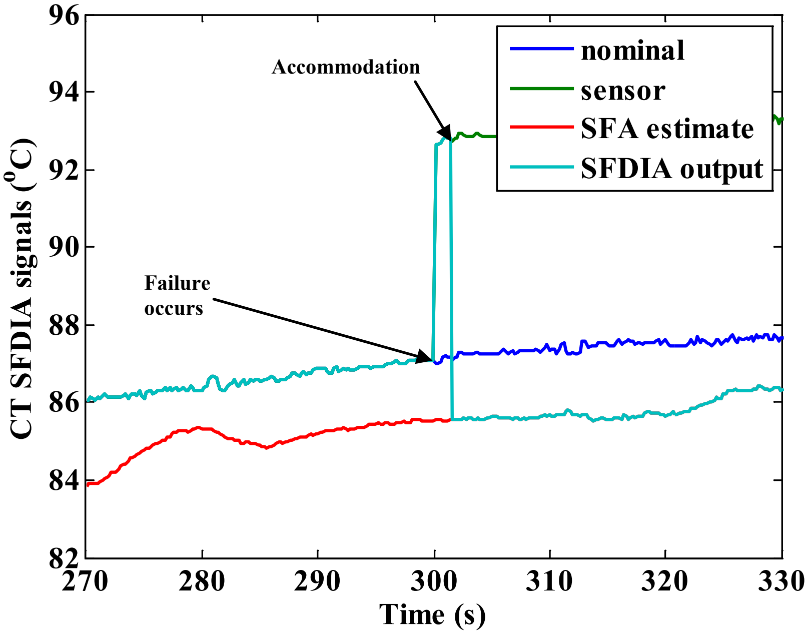


Figure 36: *CT* signal subjected to FM1 at 300s.

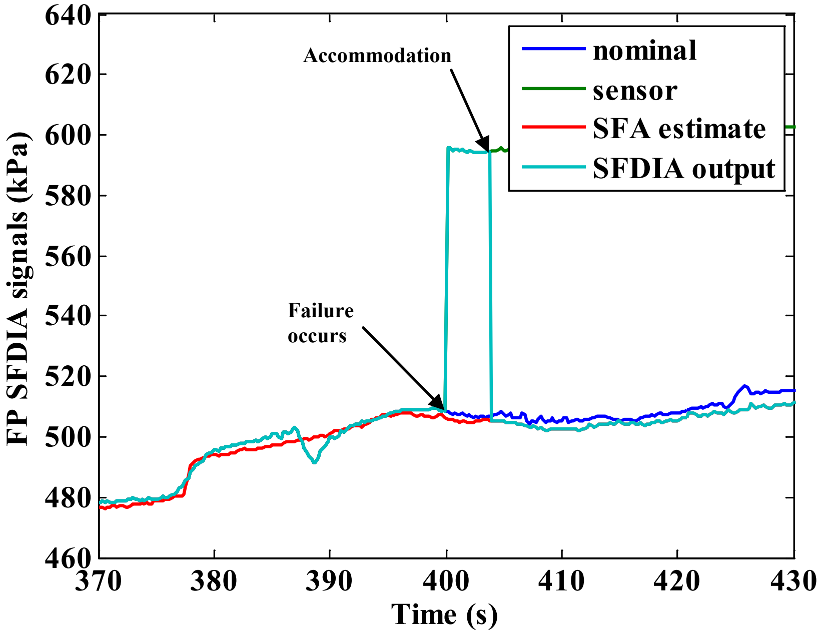


Figure 37: ***FP*** **signal subjected to FM1 at 400s.**

Prior to the failure, the SFDIA output is the ‘sensor’ signal. The ‘sensor’ signal originated from the block called ‘fail’ seen in Figure 13. The signal recorded by this block would reflect the addition of the failure as described in Equation (12). ‘Nominal’ is the signal saved before the Failure block. The ‘SFA estimate’ is the NN estimate from the Approximation block that the SFA block uses when a sensor fails. This description of the signals applied to all figures showing the SFDIA signal output.

When the *BP* sensor failed, the fast residual first reacted to stop the online learning by crossing the threshold in 0.2s, as seen in Figure 38. This changed the *LE* vector of all parameters from one to zero, effectively halting the online learning of all NNs. This prevented the NNs from adapting to faulty sensor characteristics that can affect model accuracy. When the residual for the *BP* sensor increases, the other residuals remained below their respective thresholds. Figure 110 and Figure 111 (see Appendix G) shows the residuals for *CT* and *FP* sensors. As the failure progressed, the *AE* vector for the *BP* signal changed from one to zero thus isolating and accommodating the sensor. This happened as the slow residual crossed its corresponding threshold.

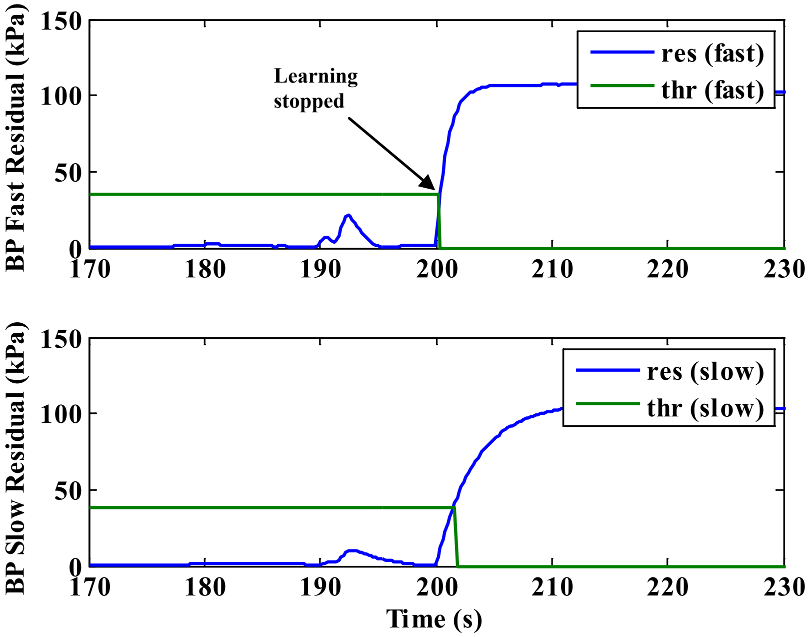


Figure 38: *BP* residuals using FM1.

The same processes occurred when the *CT* signal failed at the specified time, followed by *FP*. The SFDIA signals shown in Figure 35 to Figure 37 confirm the residual activities for each sensor. Table 10 lists the length of time it took for the accommodation modes to trigger:

Table 10: Duration of accommodation to occur with FM1 from offline simulation.

Parameter	Length of time (s)
BP	1.6
CT	1.4
FP	3.8

5.3.2 SFDIA with FM2

The focus of this section was to perform an offline demonstration of the SFDIA to respond to failures resembling a long-term drift. This means the SFDIA would not correct the failures immediately following its occurrence as it did with the FM1 mode. Each sensor selected for failure had the signal ramp begin at the time instants indicated in Table 9. Figure 39 to Figure 41 shows the SFDIA performance subjected to the FM2 mode:

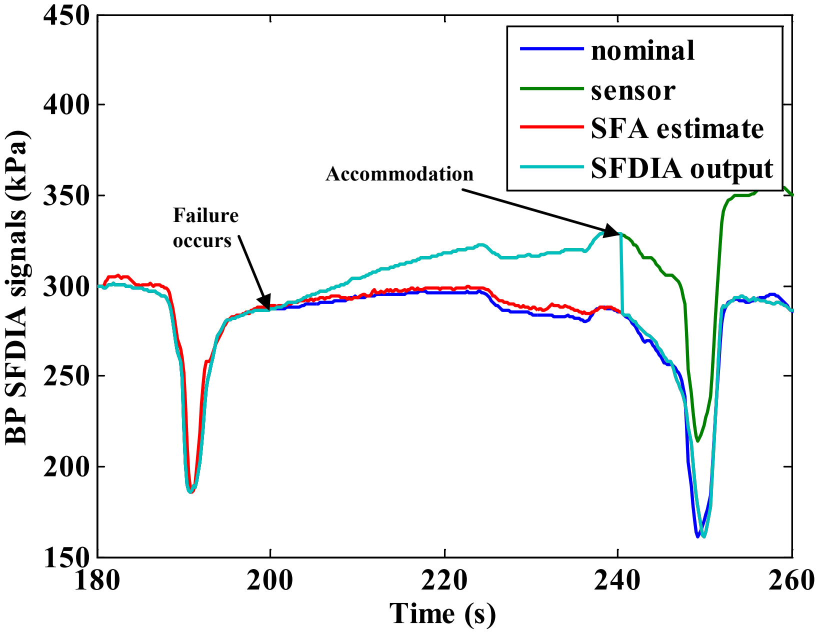


Figure 39: *BP* signal subjected to FM2 at 200s.

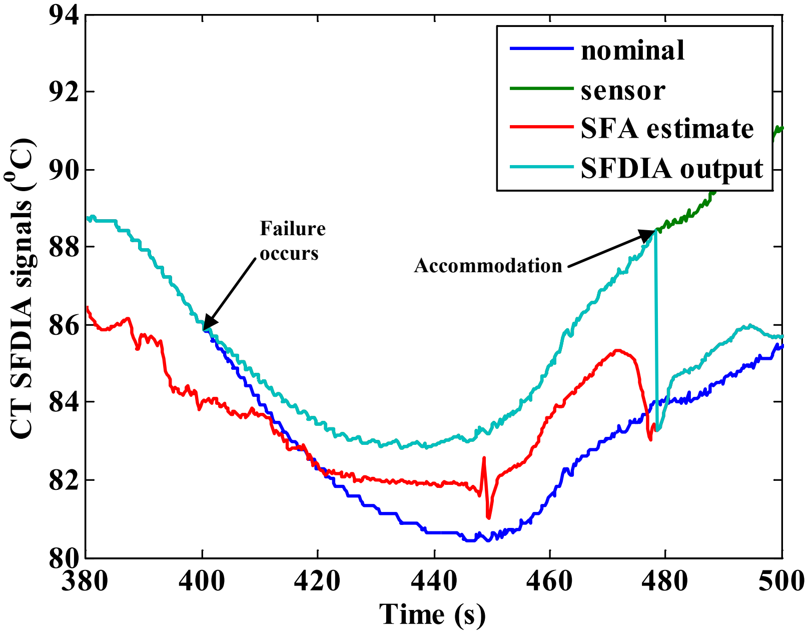


Figure 40: *CT* signal subjected to FM2 at 400s.

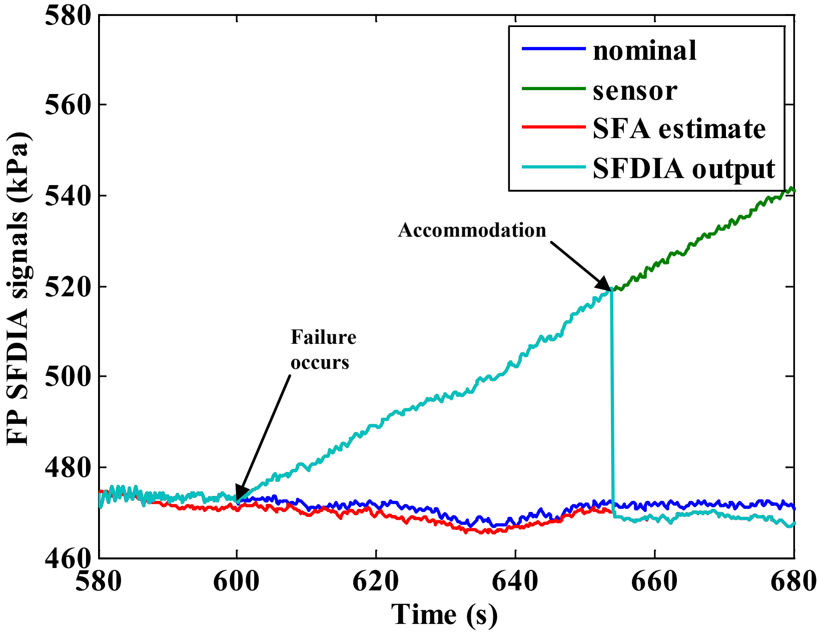


Figure 41: ***FP*** **signal subjected to FM2 at 600s.**

The specified slope of the Ramp block (see Figure 11) was 0.01. The slope chosen slowly creates the drift but without taking too long just to prove the SFDIA functionality. This takes into consideration the fact that it was not possible to run the engine in the test cell for a significant amount of time.

Each failure point was spaced further apart in comparison to FM1 due to the uncertainty of when the accommodation would trigger. In comparison, the step failure used in the previous study showed that the incidence of FM1 on a signal would cause a failure accommodation within a few seconds. Thus, spacing the FM1 failures 100s apart for each parameter was feasible. However, based on the results of Figure 39 to Figure 41, the 200s spacing of the FM2 failures were sufficient. The selected thresholds triggered all accommodations at under 100s.

According to Figure 42, all online learning capabilities ceased at 36.8s after the *BP* sensor failed. The residuals behaved in the same manner as discussed in the previous section. The difference was the type of failure under study in this section. Figure 112 and Figure 113 (see Appendix G) shows the residuals for *CT* and *FP* sensors.

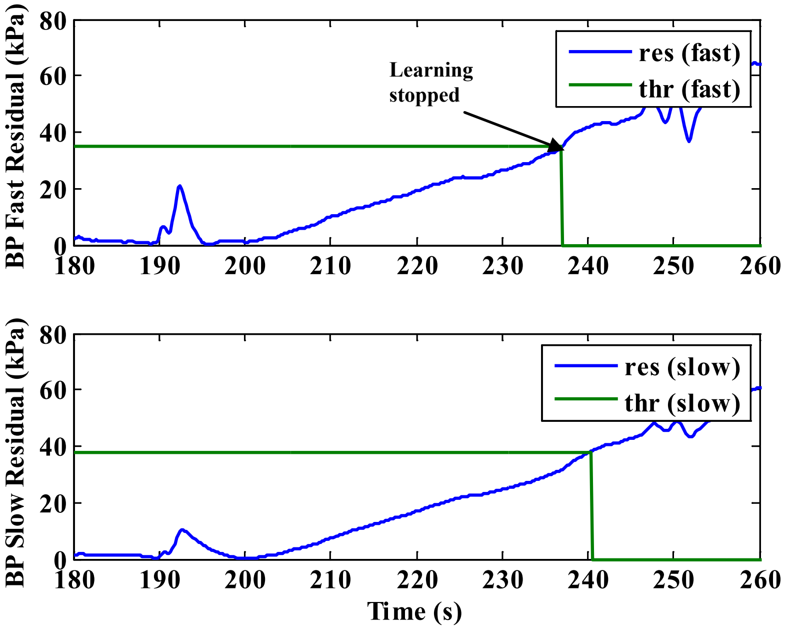


Figure 42: *BP* residuals using FM2.

Both the *CT* and *FP* sensors failed at the specified time and later both accommodated. The SFDIA signals shown in Figure 39 to Figure 41 correlate with all residual activities recorded during FM2 mode simulation.

Table 11 summarizes the time it took for each parameter to reach the accommodation mode when the slow residuals crossed their respective thresholds.

Table 11: Duration of accommodation to occur with FM2 from offline simulation.

Parameter	Length of time (s)
BP	40.4
CT	78.4
FP	53.8

Lower failure amplitudes for FM2 would take a longer time to accommodate for the given thresholds. Therefore, reducing thresholds in such instances would be necessary to best detect these failures. However, this could lead to false alarms associated with failures resembling the other modes [48].

5.3.3 SFDIA with FM3

It was important to see how the SFDIA works during instances of intermittent failures and to validate the logic developed, seen in Figure 77 (see Appendix A), to accommodate such failures. This failure mode consisted of multiple step failures that oscillated through the thresholds. As mentioned in Section 3.2, the number of pulses was three and the accommodation expected to hold after the third pulse. The setting of the Pulse Generator block (see Figure 12) specified a pulse width, t_w of 20s with the same duration between each pulse. It is noted that when Simulink started, so did the Pulse Generator blocks and their outputs added to the sensor signals at the failure times indicated in Table 9. Figure 43 to Figure 45 shows the SFDIA performance subjected to the FM3 mode:

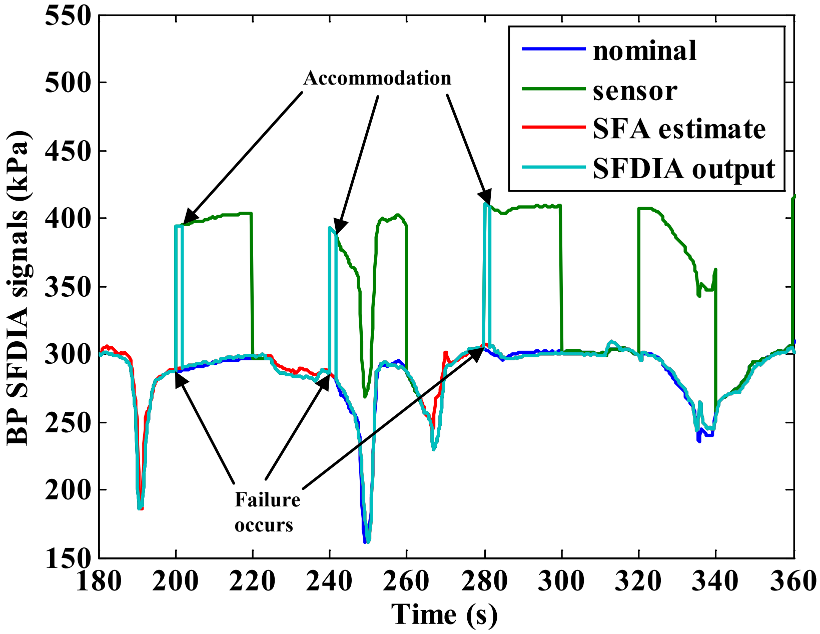


Figure 43: *BP* signal subjected to FM3 at 200s.

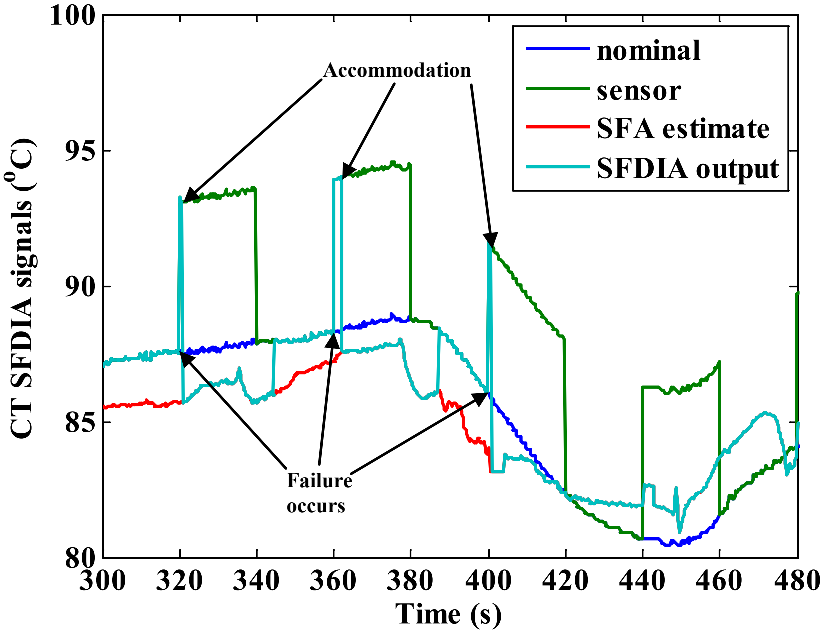


Figure 44: *CT* signal subjected to FM3 at 300s (pulse started at 320s).

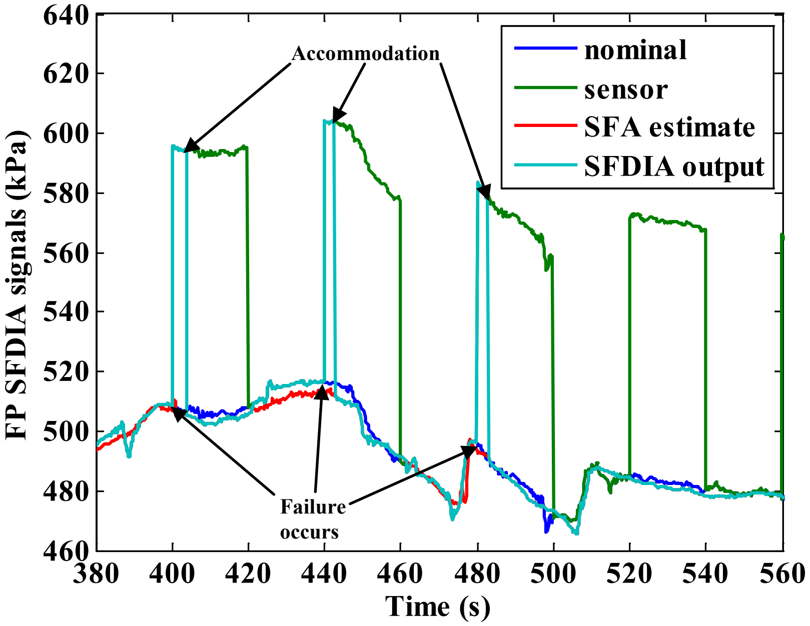


Figure 45: *FP* signal subjected to FM3 at 400s.

When the first pulse occurred on the *BP* sensor, shown in Figure 43 and Figure 46, all NN learning switched off and failure accommodation kicked in moments later. At the end of the first pulse, the failure amplitude returned to zero. This caused the residuals to drop below their thresholds and thus releasing the accommodation mode while re-enabling NN learning. When the second pulse occurred, the aforementioned effects took place again. After the third pulse, the accommodation permanently held as the SFDI logic ensured the scheme output was the NN estimate of the *BP* sensor.

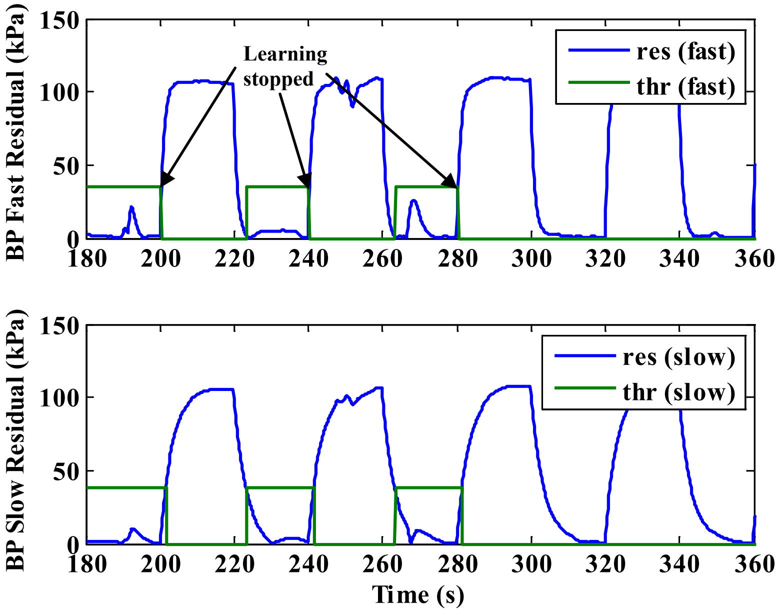


Figure 46: *BP* residuals using FM3.

The same processes took place to correct the *CT* and *FP* failures. Figure 114 and Figure 115 (see Appendix G) shows the residuals of the *CT* and *FP* sensors as they reacted to the square wave pulses that simulated the failures. All NN leaning ceased after the *BP* sensor underwent permanent accommodation. Table 12 indicates the time it took for all NN learning to disable during the *BP* failure events.

Table 12: Duration of *LE* vector to switch values with FM3 from offline simulation.

BP Pulse Number	Length of time (s)
1	0.2
2	0.2
3	0.4

For all sensors, Table 13 summarizes the length of time for the accommodation modes to enable.

Table 13: Duration of accommodation to occur with FM3 from offline simulation.

Parameter	Length of time (s)		
	Pulse 1	Pulse 2	Pulse 3
BP	1.6	1.6	1.6
CT	0.8	2	0.8
FP	3.8	2.8	3

It was noticeable that after the first two pulses ended in Figure 44 there were delays in restoring the signal to nominal. In fact, it also occurred in Figure 43 and Figure 45 for the pressure sensors. The delays there were shorter and not easily detectable in the figures. When the pulses ended, the *CT* slow residual did not fall as vertically as the ones for *BP* and *FP*. The solution for reducing this delay was to improve the accuracy of the NN models. Another possibility was to tune the SFDIA with FM3 type failures and adjust the slow filter bandwidth to reduce the delay. However, the SFDIA signals generated for this failure mode showed that the SFDI logic performed as intended.

5.4 Breakout Panel Verification and Effects of Sensor Failure

Offline simulation showed the functionality of the SFDIA, which led to testing the concept in real-time. First set of results in this section indicated whether the breakout panels interfered with EMS operation. This was followed by a test to observe the effects on the engine when a sensor critical to fuel timing failed with no accommodation. The failure was artificially added through the Failure block of the SFDIA scheme.

5.4.1 Engine Test Setup and Breakout Panel Verification

As a first step, laboratory calibrations and verifications for all thermocouples, pressure transducers, dynamometer and gaseous analyzers associated with the test setup were conducted. Next, engine map cycles were run for creating FTP setpoints without using the breakout panels. An engine map cycle involved running the engine at 100% load while varying engine speed between curb idle and high idle. The blower flow rate of the exhaust sampling tunnel was set to 2100 scfm. Test cell data acquisition for all sensors and gaseous analyzers was set to 10 Hz. Fuel for the test engine was Ultra-Low Sulfur Diesel (less than 15ppm sulfur) supplied by the Guttman Oil Company. Prior to running the map cycles, the engine inlet depression and exhaust backpressure were set to manufacturer specifications. Figure 78 (see Appendix B) shows the lug curve from the chosen map cycle. The EERL laboratory software is designed to create the FTP cycle using engine map data with normalized setpoints defined in 40 CFR. The data extracted from the map is shown in Table 28 (see Appendix B). A view of the generated FTP duty cycle is provided in Figure 116 (see Appendix G).

All FTP tests conformed to 40 CFR procedures, where each FTP test observed a 20-minute soak period between each run. The soak period entails shutting down the engine and allowing it to cool. This ensured each test starts at a common reference condition. The data of interest and the ones reported in this study are only from the ‘Hot Start’ FTPs. In other words, the engine first went through a warm-up phase prior to acquiring performance data. This consisted of initially running the engine manually until the engine thermostat opened. Immediately after this was a warm-up FTP test, followed by a 20-minute soak period. Thus, all FTP tests from this point onwards were Hot Start FTPs. If a test did not immediately start after the soak period and the engine had not completely cooled down, then the next FTP was considered a warm-up cycle. A repeat of the manual warm-up was sometimes necessary depending on the length of time the engine was shutdown.

All continuous emissions data measured by the analyzers were from the exhaust stream, which was diluted with conditioned ambient air. A Teflon® bag collected background (ambient) air during each test for sampling through the analyzers at the conclusion of a test. All dilute exhaust emissions measurements were corrected for background emissions levels. The ambient air and fuel conditioning all conformed to 40 CFR standards.

The verification of the breakout panels consisted of two parts. First part was to run baseline tests, meaning running the engine without the panels. After that, the breakout panels were connected as shown in Figure 22 but without the DAQ equipment attached. All engine signals had a direct connection to the EMS via breakout panels. The data of interest are the integrated results from each test. Each parameter had its repeatability checked by calculating the Coefficient of Variation (*COV*) percentage as follows:

$$COV\ percentage = \frac{standard\ deviation}{average} \cdot 100 \tag{27}$$

The *COV* gave an indication to the degree of dispersion within the dataset. In addition, it was also necessary to determine the occurrence of outliers using the ASTM E178 standard [83] based on a 1% significance level. Significance levels of 1% or 5% were typical in the usage of this standard. The method outlines a standardized procedure for determining data outlier limits. Any values outside these limits for a given significance level are deviations in the dataset. For a 1% significance level, values within the limits meant there is a 99% chance to believe they occurred as measured. Results from the verification tests are shown in Table 14 and Table 15:

Table 14: Emissions data without breakout panels from verification test.

Run		Total work (bhp-hr)	*HC* (g/bhp-hr)	*NOx* (g/bhp-hr)	*CO* (g/bhp-hr)	CO_2 (g/bhp-hr)
1		24.35	0.118	2.50	1.12	559.65
2		24.33	0.118	2.51	1.09	560.21
3		24.32	0.119	2.56	1.08	560.63
Average		24.33	0.118	2.53	1.10	560.16
Standard Deviation		0.015	0.001	0.033	0.017	0.495
COV		0.06%	0.62%	1.32%	1.54%	0.09%
Outlier Limits	Maximum	24.35	0.119	2.56	1.12	560.73
	Minimum	24.31	0.118	2.49	1.08	559.59

Table 15: Emissions data without breakout panels from verification test.

Run		Total work (bhp-hr)	*HC* (g/bhp-hr)	*NOx* (g/bhp-hr)	*CO* (g/bhp-hr)	CO_2 (g/bhp-hr)
1		24.33	0.120	2.52	1.08	559.87
2		24.32	0.118	2.55	1.07	560.50
3		24.33	0.117	2.49	1.08	559.45
Average		24.32	0.118	2.52	1.08	559.94
Standard Deviation		5.92×10^{-3}	1.15×10^{-3}	0.030	0.010	0.53
COV		0.02%	0.98%	1.19%	0.97%	0.09%
Outlier Limits	Maximum	24.33	0.120	2.56	1.09	560.55
	Minimum	24.32	0.117	2.49	1.07	559.33

In Run 1 without panels, a corrected *HC* value is presented. It used an adjusted background *HC* value averaged from the other test runs mentioned in Table 14 and Table 15. The original background *HC* level was an erroneous value recorded at the time of testing. All *COV* percentages indicated there was low dispersion of values within the dataset. In addition, all data acquired were within the outlier limits. Therefore, the results in the above tables verified that the breakout panels did not interfere with the operation of the engine. The DAQ system connection then took place as described in Section 4.3.2.

5.4.2 Simulink Settings and Failure without Accommodation Study

Engine testing for this portion of the study occurred 50 days from the one carried out for Section 5.4.1. The reason for the duration was laboratory scheduling. This required new laboratory checks and rerunning an engine map, with the map data in Table 36 (see Appendix G). The new FTP setpoints are shown in Figure 117 (see Appendix G). At a glance, there are no noticeable differences between this and the previous cycle setpoints. Between the two different tests, it was natural to expect variations in laboratory and ambient conditions.

Recalling that the offline simulation results in Section 5.3 had the NN learning capabilities turned off. The *LE* vectors were still varying to demonstrate the SFDIA functionality. In performing the real-time test, it was necessary to turn on all GRBF block-learning capabilities (see Figure 16). Therefore, the *LE* vectors now controlled all the EMRAN RBF NN learning. The learning rate selected was 1×10^{-4}. The purpose for using a small learning rate was that data for training the NNs originated from VAT2000 using its own DAQ hardware.

Now, the SFDIA signals received were from the DAQ system assembled for this study. This also means from this point onwards, the *BP*, *CT* and *FP* sensor signals passed through the DAQ system prior to the EMS. It was not expected that the manual calibrations shown in Appendix F would accurately replicate the calibrations used by VAT2000. In addition, there might be other unknown factors affecting the estimations from using this DAQ hardware. Therefore, there must be adaptation capabilities applied to the models to compensate for any differences present in the DAQ signals with respect to the ones from VAT2000. The sampling rate of the Data Acquisition Toolbox blocks in Figure 95 (see Appendix F) was 200 Hz. However, all data recording occurred at 10 Hz.

With the DAQ system connected to the breakout panels, it was necessary to observe the effects of a sensor failure on the engine. As mentioned earlier, only six sensors originated from the panels. The other three sensors were placed separately around the engine. The sensor selected to perform this observation was the *BP* sensor. This was because the sensor is critical for determining fuel timing, intake air mass flow rate and EGR control. To prevent accommodation, all thresholds had its values increased substantially such that the residuals could never cross them. The failure mode tested here was FM1 occurring at 200s.

In Simulink, the *BP* sensor variation occurred as shown in Figure 47. The SFDIA sent the erroneous signal to the EMS and continued until the end of the test. There appeared to be minimal noise levels in the signals. The SFA estimate trace indicated that its accuracy reduced from the point of failure. The primary reason for this was the NN model was learning from the error. The *LE* variable did not switch off the learning, as the fast residual threshold was too high.

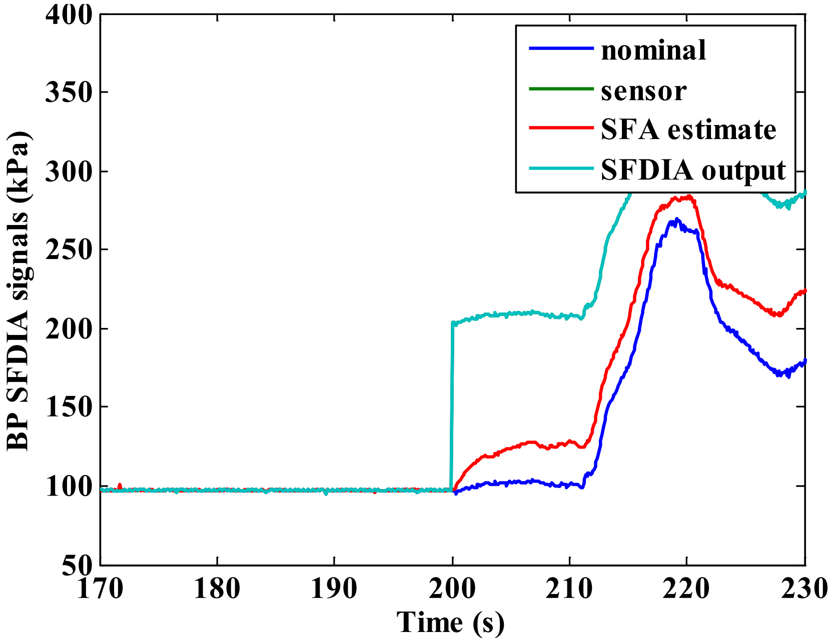


Figure 47: *BP* sensor failure at 200s with no accommodation.

Similar effects occurred in Figure 48 and Figure 49 for the *CT* and *FP* signals. This time the step increase in the *BP* signal affected the *CT* and *FP* models, as *BP* is an input parameter for those models. Since no accommodations took place, the model activities remained isolated from the EMS. The residuals from this test with the modified threshold are shown in Figure 118 to Figure 120 (see Appendix G). It was noticeable that there were slight noise oscillations in the SFDIA output signals associated with the *CT* and *FP* sensors. The *CT* model also had noise effects in its output, which was expected since the training *CT* data had noise levels in them. However, all the NN outputs tracked closely to their respective nominal signals prior to the failure. This was due to the learning rates now active within the NNs.

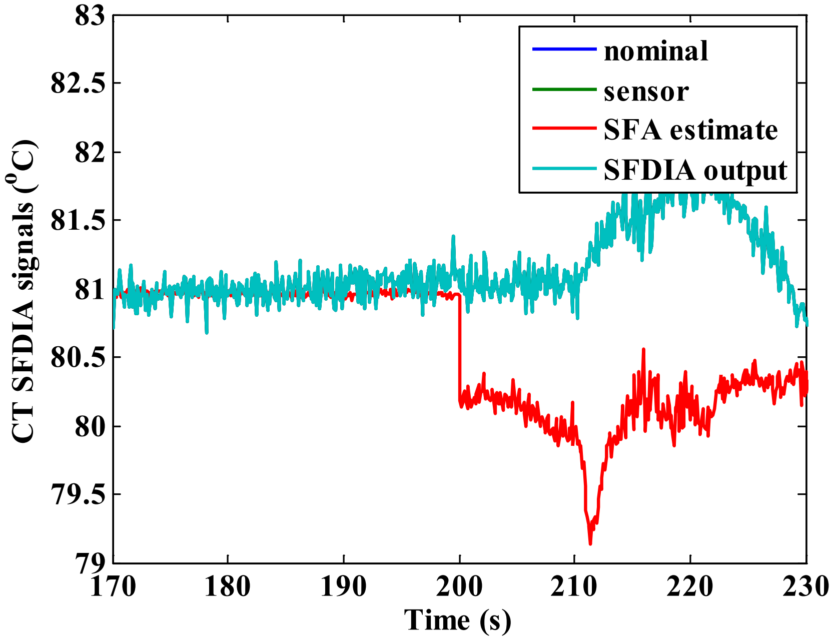


Figure 48: ***CT*** **signal when** ***BP*** **sensor failed at 200s with no accommodation.**

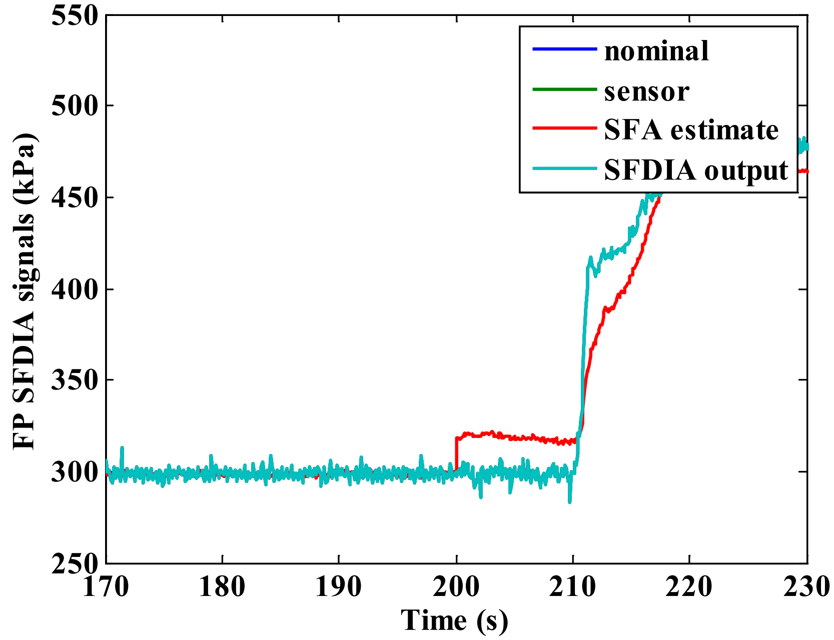


Figure 49: ***FP*** **signal when** ***BP*** **sensor failed at 200s with no accommodation.**

New FTP setpoints meant this portion of the study required new baseline data. The purpose was to compare the results above with data without any sensor failures. Baseline testing occurred with the breakout panels attached for accurate representation. Data acquired from the baseline tests are shown in Table 16. The most noticeable differences were in the *HC* averages reported in Table 14 and Table 16. The main cause was the usage of a different fuel batch; hence, different fuel properties, such as Cetane numbers and aromatic hydrocarbon content. Other contributions can include variations in background emissions levels.

Table 16: Baseline results for comparison purposes (failure without accommodation).

Run		Total work (bhp-hr)	*HC* (g/bhp-hr)	*NOx* (g/bhp-hr)	*CO* (g/bhp-hr)	CO_2 (g/bhp-hr)
1		24.23	0.140	2.54	1.09	560.37
2		24.24	0.140	2.54	1.09	560.35
3		24.24	0.140	2.54	1.07	560.78
Average		24.23	0.140	2.54	1.09	560.50
Standard Deviation		$2.59\text{x}10^{-3}$	$3.61\text{x}10^{-4}$	$2.70\text{x}10^{-3}$	0.011	0.24
COV		0.01%	0.26%	0.11%	1.01%	0.04%
Outlier Limits	Maximum	24.24	0.141	2.54	1.10	560.78
	Minimum	24.23	0.140	2.54	1.07	560.22

Therefore, Table 17 presents the differences between the emissions from a failed *BP* signal to the baseline. Baseline results are Run 1 in Table 16.

Table 17: Comparison of selected test results when *BP* failed at 200s.

Parameters	Baseline	With FM1	Difference (%)
Total Work (bhp-hr)	24.23	24.26	0.13%
HC (g/bhp-hr)	0.14	0.26	85.63%
NOx (g/bhp-hr)	2.54	1.89	-25.63%
CO (g/bhp-hr)	1.09	3.65	233.54%
CO_2 (g/bhp-hr)	560.37	557.81	-0.46%
Fuel Consumption (g/bhp-hr)	176.72	178.87	1.22%

The percentage differences calculated in Table 17 were in comparison to baseline. The results showed that failing of the *BP* sensor effected the *HC*, *NOx* and *CO* emissions significantly. The erroneous signal effectively compromised the emission control strategy of the engine. The figures below show how fuel injection timing measured in degrees from TDC along with EGR mass flow rate differed from baseline, viewed from a portion of the FTP cycle:

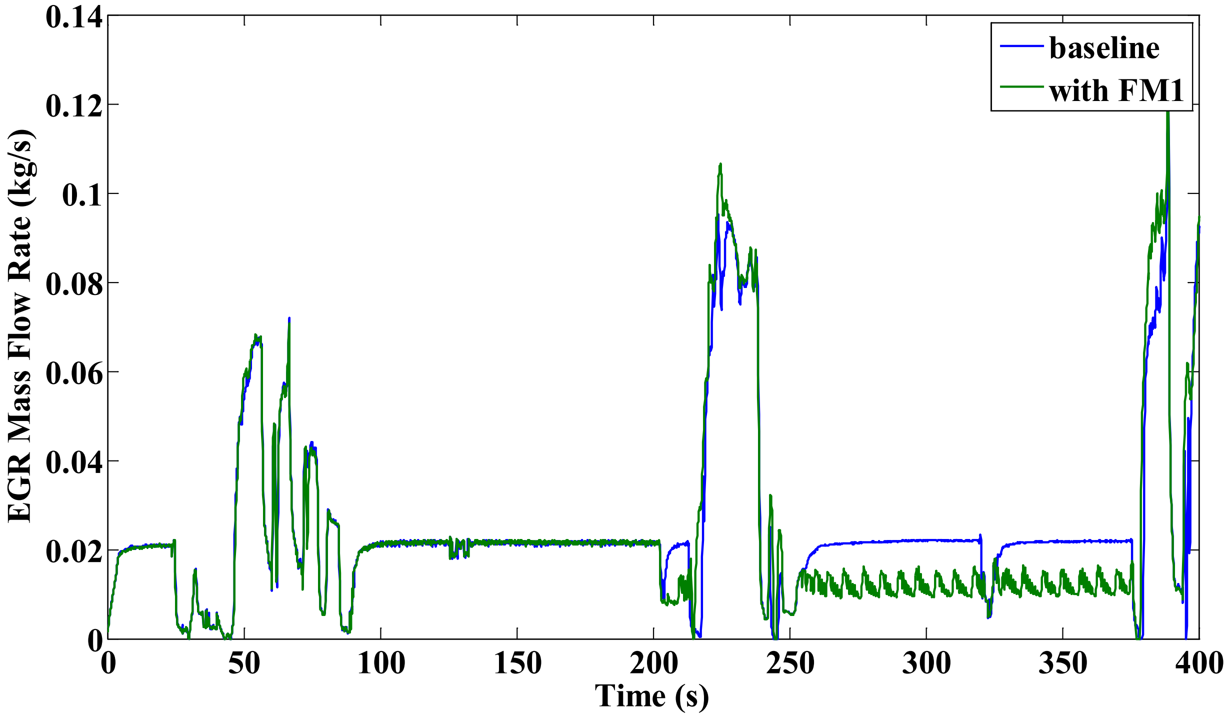


Figure 50: EGR control from FTP cycle when *BP* failed at 200s.

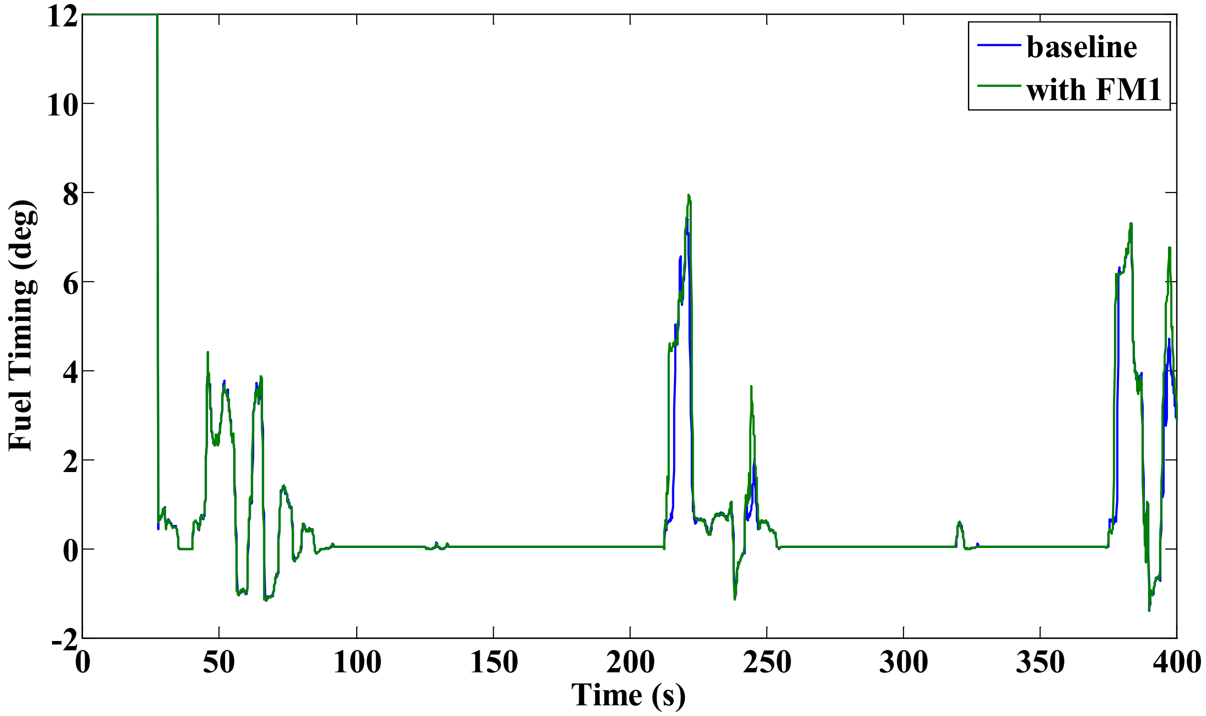


Figure 51: Fuel timing from FTP cycle when *BP* failed at 200s.

When the failure occurred, the EGR mass flow rate was first to show signs of disruption. Engine timing on the other hand did not show any deviations until the engine throttle ramped up. That is because at 200s, the FTP setpoints was in an engine idling region. However, there was EGR activity at idle.

Table 17 showed that there was only a small variation in total power with respect to baseline, implying that power was not affected. Using the same conclusion meant CO_2 variation would also be negligible, as evident in Table 17. In addition, the FTP cycle ran using the erroneous sensor signal conformed to all 40 CFR cycle regression limits, as shown in Table 37 (see Appendix G). However, the continuous trace of engine torque pointed to noticeable variations, especially in high load regions of the FTP cycle as illustrated in Figure 52.

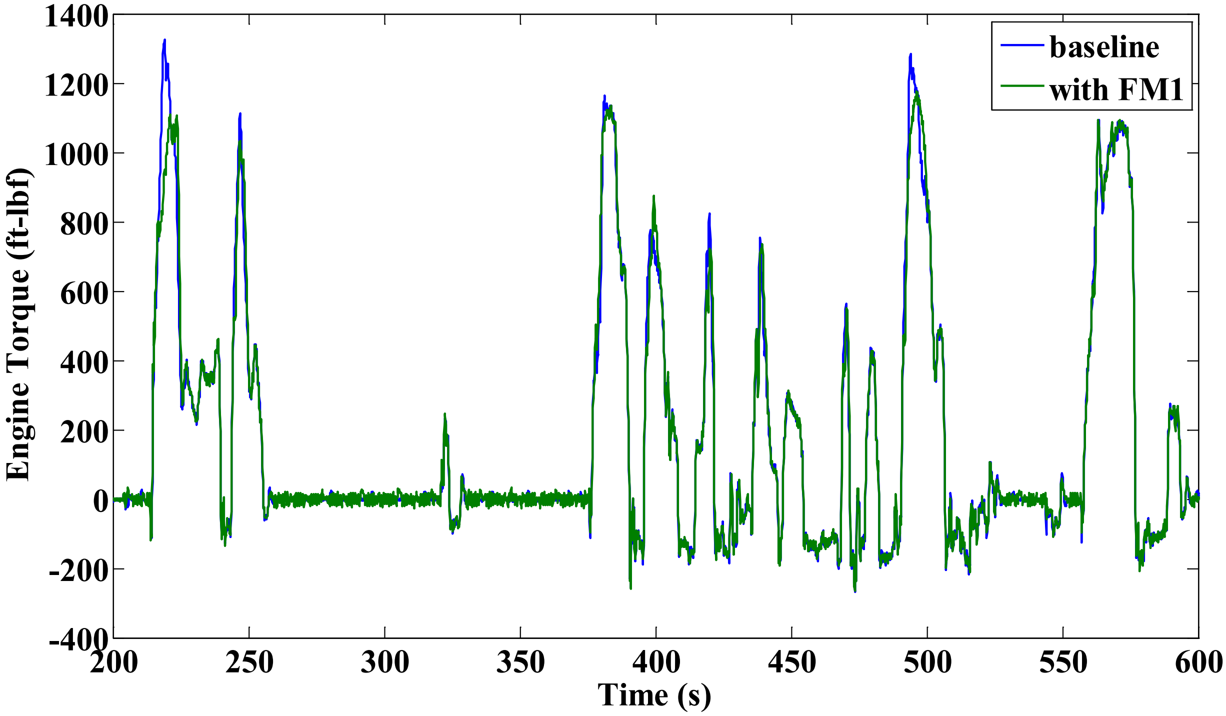


Figure 52: A segment of engine torque data from the FTP (*BP* failure at 200s).

Although Figure 51 shows a segment from the FTP cycle, fuel-timing angles were generally higher in comparison to baseline. Artificially sending the EMS higher *BP* values made the engine control strategy retard the injection timing. In other words, the erroneous *BP* signal delayed fuel injection more often with respect to baseline, as indicated by the higher injection timing angle values at high load points. This led to a decrease in combustion efficiency. Due to the load demand from the cycle, more fuel was injected to compensate for the resulting combustion inefficiency. Table 17 data does indicate higher fuel consumption. Higher fuel consumption entails an increase in CO_2 emissions, but that was not the case as there was a small reduction in brake-specific CO_2 emissions. Decreasing combustion efficiency produced higher *HC* and *CO* emissions, possibly resulting in lesser carbon available for CO_2 formation.

5.5 Real-time SFDIA Validation

As observed in the previous section, a sensor failure left uncorrected can disrupt the emission control capabilities of an engine. Therefore, this section discusses the real-time results of the SFDIA scheme. Similar to the previous section, the SFDIA testing occurred inline with the engine EMS. All residual thresholds were as defined in Table 8 to obtain the real-time data. Before beginning a test, the original NN weights were uploaded to maintain a common starting point for comparison purposes. Due to laboratory time constraints on the day of testing, it was possible to run one test for each failure mode and baseline. The data of interest here are the continuous data collected during each FTP cycle. The FTP setpoints is the same as the one used for acquiring data in the previous section.

5.5.1 SFDIA with FM1

This section discusses the real-time SFDIA performance subjected to step failures. The failures occurred at the time instants indicated in Table 9. Figure 53 to Figure 55 shows the SFDIA performance subjected to FM1 mode:

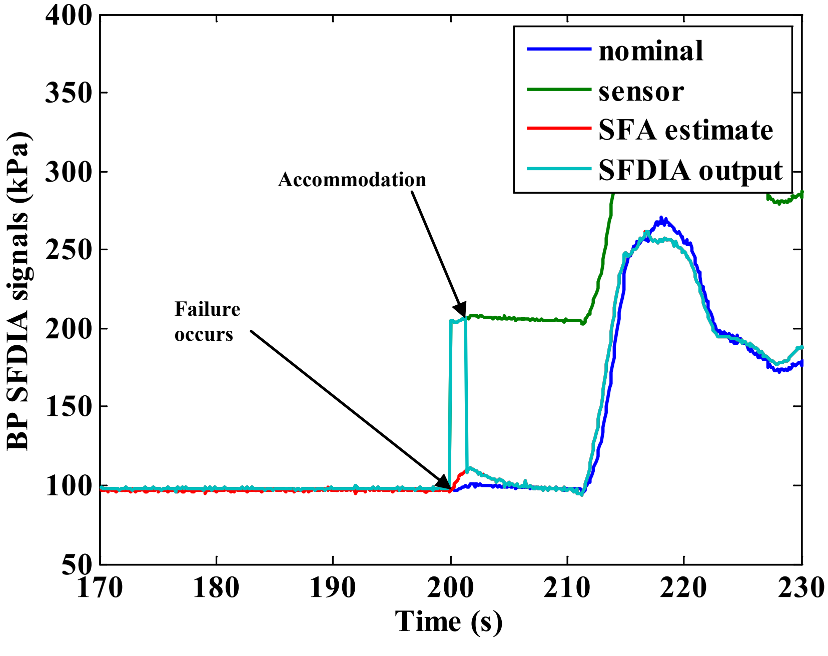


Figure 53: *BP* signal subjected to FM1 in real-time at 200s.

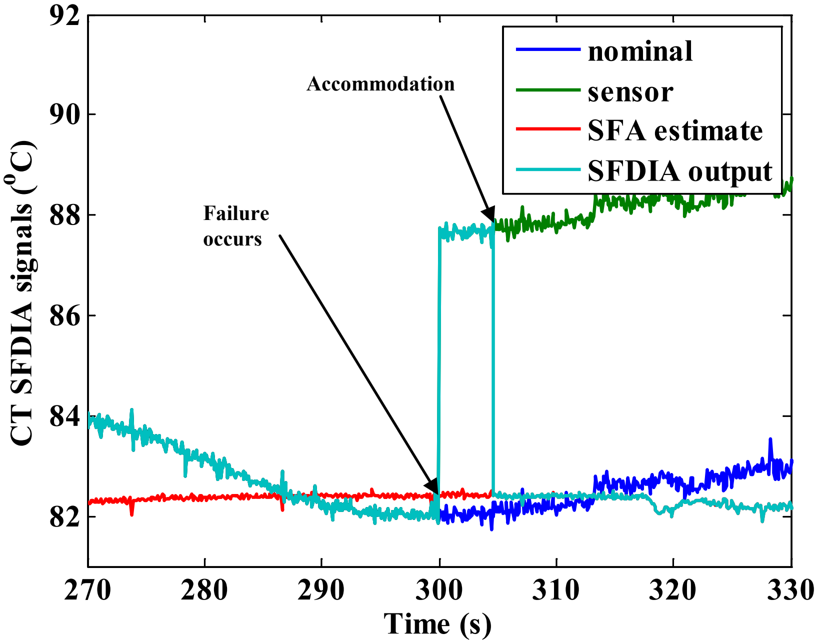


Figure 54: ***CT*** **signal subjected to FM1 in real-time at 300s.**

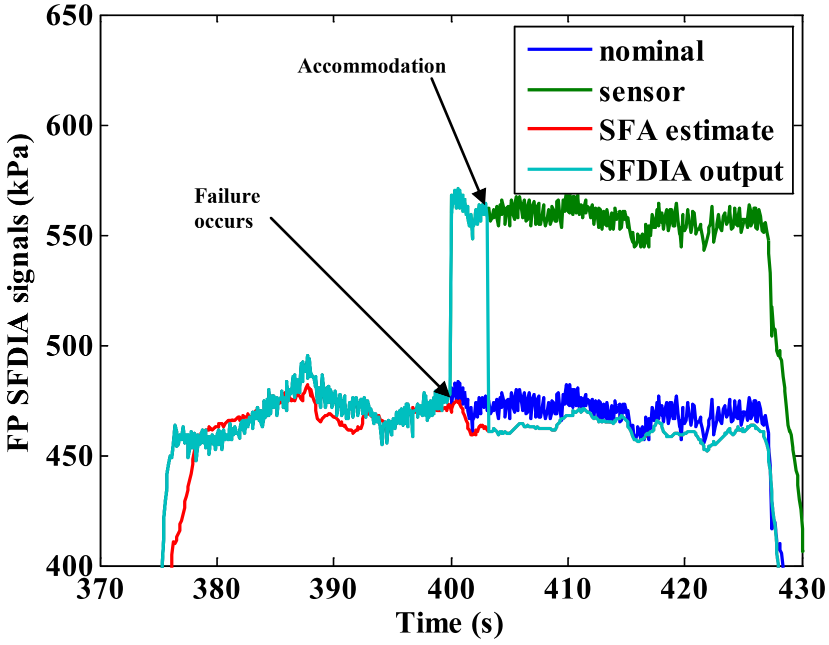


Figure 55: ***FP*** **signal subjected to FM1 in real-time at 400s.**

The fast residual associated with the *BP* sensor crossed its threshold in 0.3s, as seen in Figure 56. This halted the online learning of all NNs. As expected, the other fast residuals remained below their respective thresholds during this time. Figure 121 and Figure 122 in Appendix G shows the residuals for *CT* and *FP* sensors. The *AE* vector for the *BP* signal eventually switched values as the slow residual crossed its threshold. This process isolated the failed signal and corrected it with values from the NNs. At the point of the *BP* accommodation shown in Figure 53, the slow residual of the healthy sensors remained below their thresholds. Later when the *CT* and *FP* sensors both failed at their specified times, their signals were isolated and accommodated. The SFDIA signals shown in Figure 58 and Figure 59 confirm the residual activities associated with the *CT* and *FP* sensors. Both *BP* and *CT* failures occurred during engine idle periods.

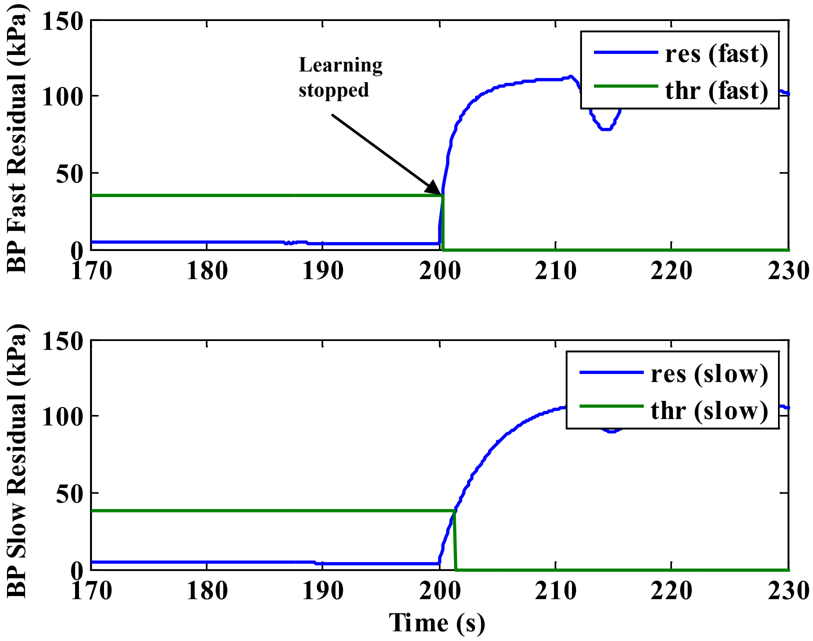


Figure 56: *BP* residuals with FM1 from real-time SFDIA tests.

Table 18 indicates the length of time it took for the accommodation modes to trigger:

Table 18: Duration of accommodation to occur in real-time with FM1.

Parameters	Length of time (s)
BP	1.4
CT	4.7
FP	3.2

Table 19 shows the performance of the models during the accommodation phase. Any changes to the original models occurred in the θ parameters of Equation (2), as the number of neurons remained the same.

Table 19: Model performance during accommodation phase (FM1failure).

Parameters	RMS Error (%)
BP	2.35
CT	14.89
FP	3.27

The error values in Table 19 implied that the *CT* model required a higher learning rate to compensate for the noise levels. Maintaining the learning rate mentioned in this study meant spending a longer amount of time for NN adaptation.

Figure 57 to Figure 59 shows the data recorded by VAT2000 as it received them from the EMS CAN bus during the failure events. It was not possible to record fault-free signals in VAT2000 simultaneously due to the usage of the breakout panels. Therefore, baseline data functioned as the fault-free dataset for comparison purposes. Although the *BP* and *CT* failures appear to follow the same trend of the Simulink results, the *FP* signal recorded in VAT2000 showed a different profile. In fact, it was expected that the VAT2000 plots would show similarities to the ones seen in Figure 53 to Figure 55 as the signals originated from Simulink. However, that was not the case. The *FP* sensor had noticeable noise levels in the signal but Figure 59 showed the noise removed. It is not known how VAT2000 or the EMS performs signal processing. The differences in the plots indicate that for each signal, there are varying levels of signal filtering or shaping involved. The solution is to involve manufacturer participation to implement the same signal processing into the scheme for more accurate representation.

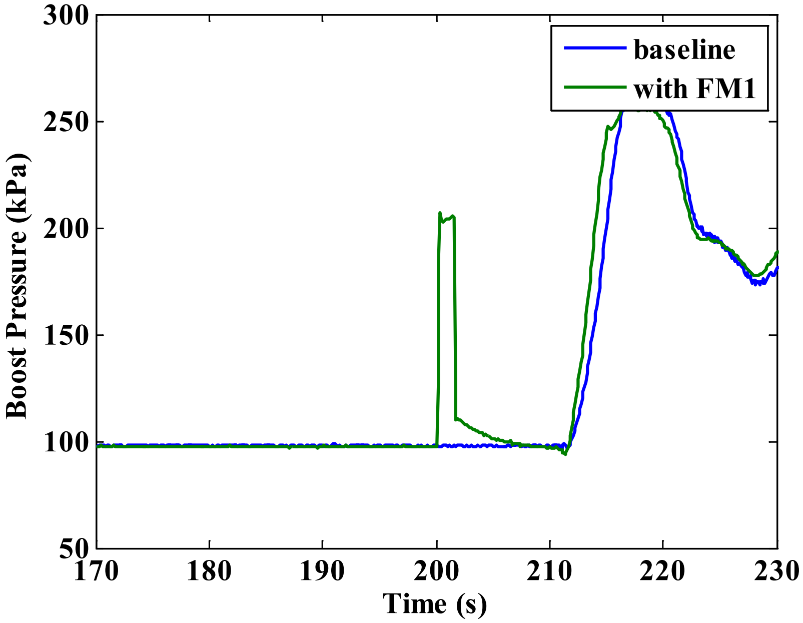


Figure 57: VAT2000 *BP* signal subjected to FM1 failure at 200s.

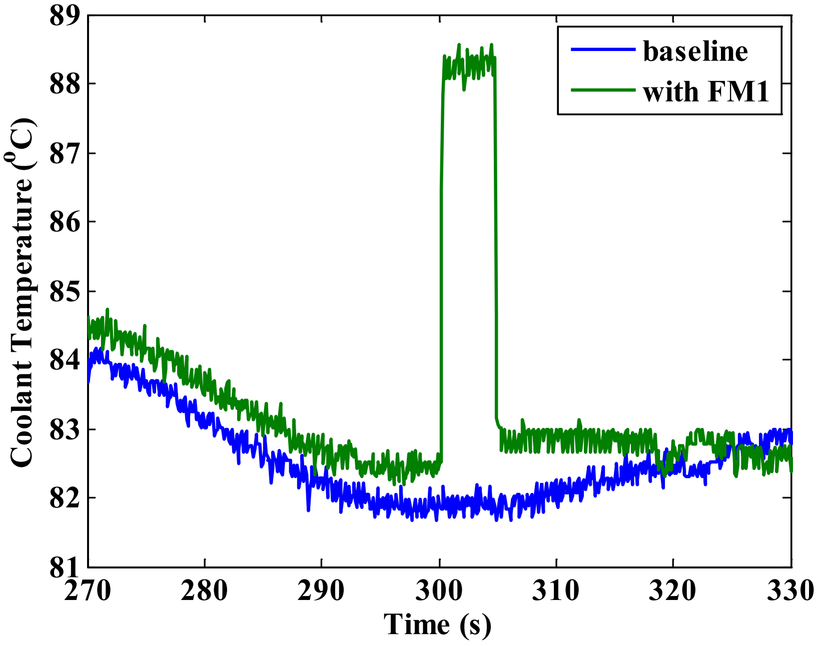


Figure 58: VAT2000 *CT* signal subjected to FM1 failure at 300s.

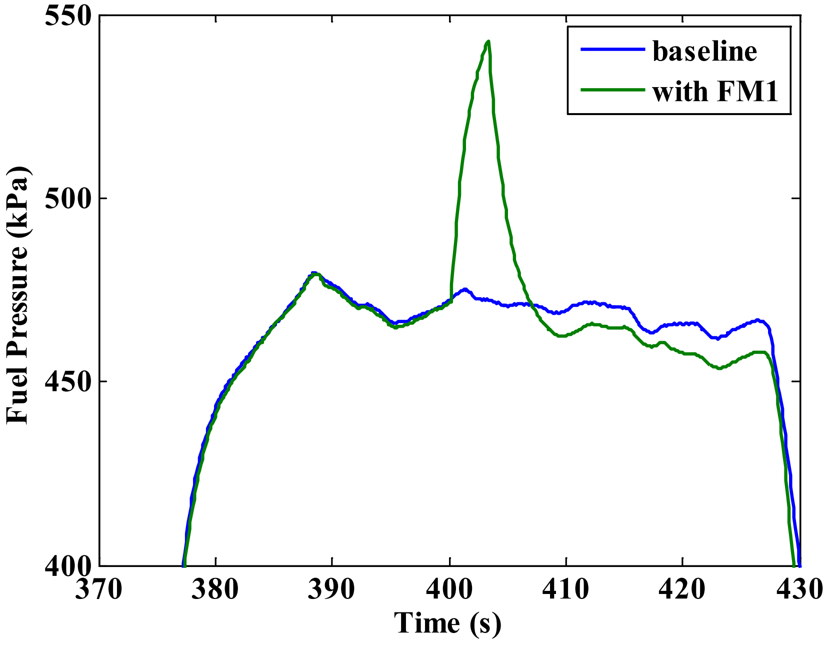


Figure 59: VAT2000 *FP* signal subjected to FM1 failure at 400s.

5.5.2 SFDIA with FM2

This section focuses on presenting the real-time SFDIA performance as it responded to long-term drift type failures. This implied the SFDIA did not correct the failures immediately following its occurrence. Each sensor selected for failure had the signal ramp begin at the time instants indicated in Table 9. Figure 39 to Figure 41 shows the SFDIA performance subjected to the FM2 mode:

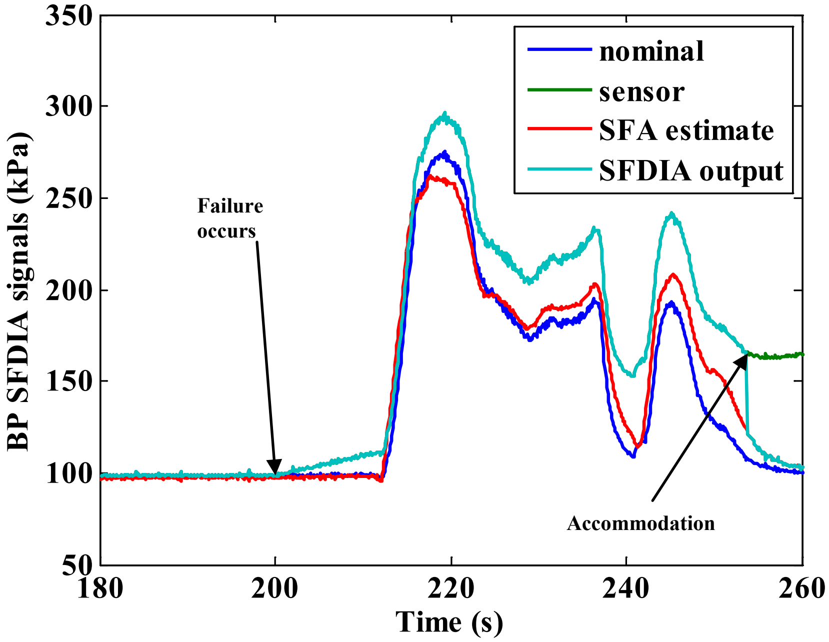


Figure 60: *BP* signal subjected to FM2 in real-time at 200s.

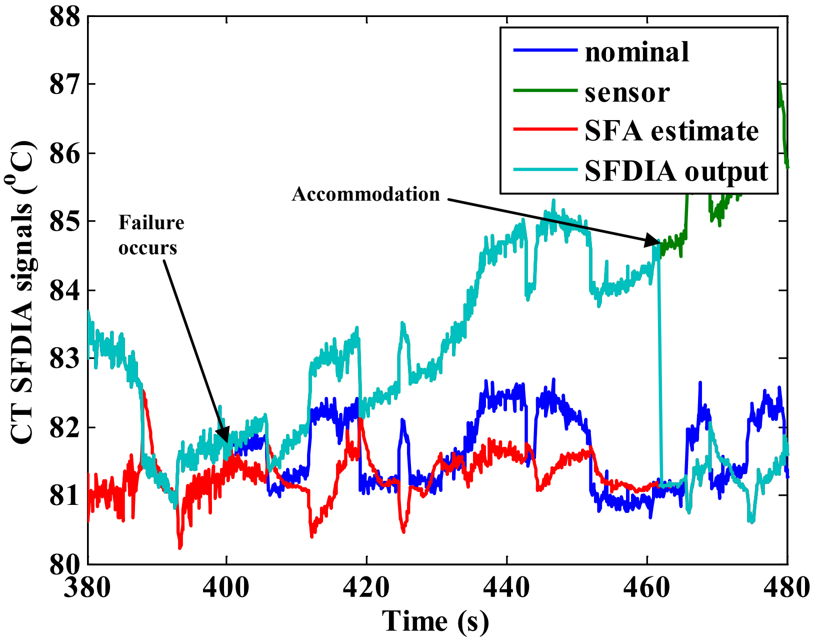


Figure 61: *CT* signal subjected to FM2 in real-time at 400s.

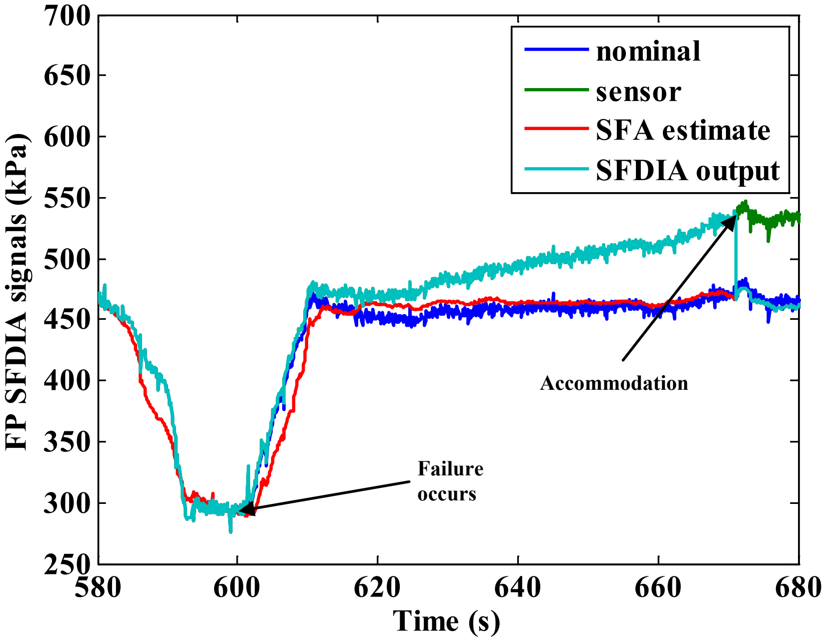


Figure 62: *FP* signal subjected to FM2 in real-time at 600s.

Figure 60 to Figure 62 indicated that the spacing between the failures was sufficient to prevent them from overlapping. The value of the Ramp block (see Figure 11) slope remained at 0.01 from the offline simulation. The Ramp block functioned to increase the sensor value gradually to create the effect of long-term drift. The selected thresholds triggered all failure isolation and accommodation at under 100s. A different slope value would certainly influence the SFDIA to correct the failure for the given thresholds.

According to Figure 63, all online learning capabilities ceased at 51.8s. The *LE* vector did switch values twice before permanently stopping the adaptation. The algorithm in the SFDI block is such that prior to an accommodation phase, the fast residual can move above and below the thresholds. Once an accommodation happens and holds, the adaptation cannot reactivate. Overall, the residuals reacted similarly as discussed in the previous section for the step failure. Figure 123 and Figure 124 (see Appendix G) shows the residuals as they occurred for the *CT* and *FP* sensors. The *CT* and *FP* sensors failed at the specified time and later both accommodated. The SFDIA signals shown in Figure 60 to Figure 62 agreed with all residual activities recorded for the FM2 mode.

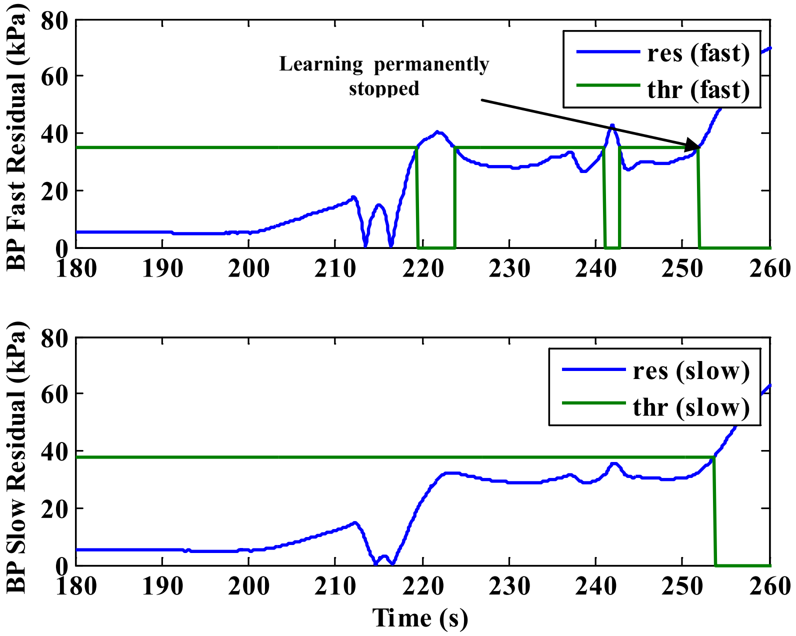


Figure 63: *BP* residuals with FM2 from real-time SFDIA tests.

Table 11 summarizes the duration it took for each parameter to reach the accommodation mode.

Table 20: Duration of accommodation to occur in real-time with FM2.

Parameters	Length of time (s)
BP	53.7
CT	61.7
FP	71.1

Table 21 lists the performance of the NN models from the Approximation block as they provided the EMS with replacement signals during the accommodation phase. Similar to the step failure study, any changes to the original models during periods of adaptation occurred only in the θ parameters of Equation (2). The number of EMRAN RBF neurons remained the same.

Table 21: Model performance during accommodation phase (FM2 failure).

Parameters	RMS Error (%)
BP	4.39
CT	13.79
FP	2.98

In comparison to Table 19, the *BP* NN performance showed a reduction in accuracy. The fast residual of Figure 63 indicated that all NNs learned from the failed *BP* signal for 45.6s. The *BP* model may have been more sensitive to this duration and thus affected its performance.

The VAT2000 data in Figure 64 to Figure 66 shows that the engine did receive the SFDIA signals as it recorded the ramp increase in values of the failed sensors. Visually, the *FP* signal during the ramp increase did not show much variation in comparison to its Simulink signal. This implies that the effects of the EMS signal processing on this signal are more prominent if they are abrupt.

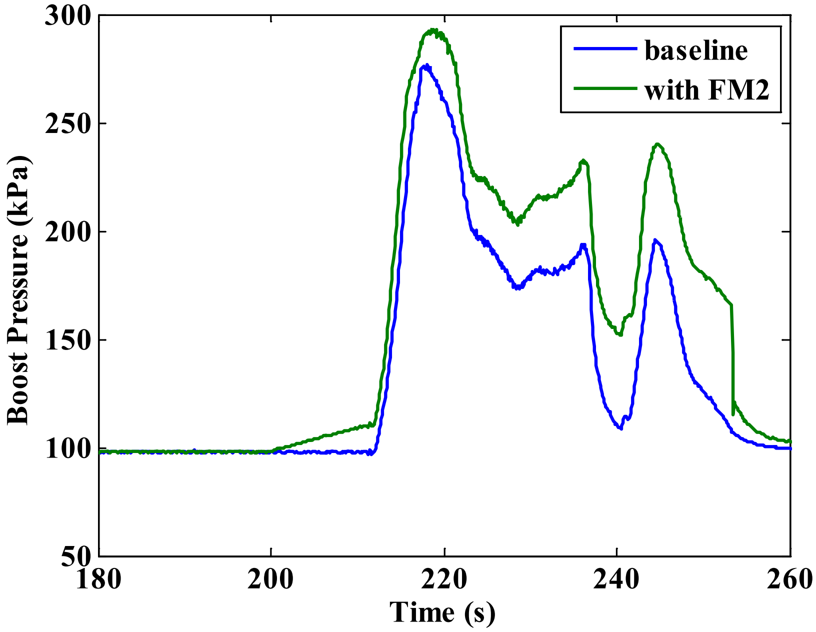


Figure 64: VAT2000 *BP* signal subjected to FM2 failure at 200s.

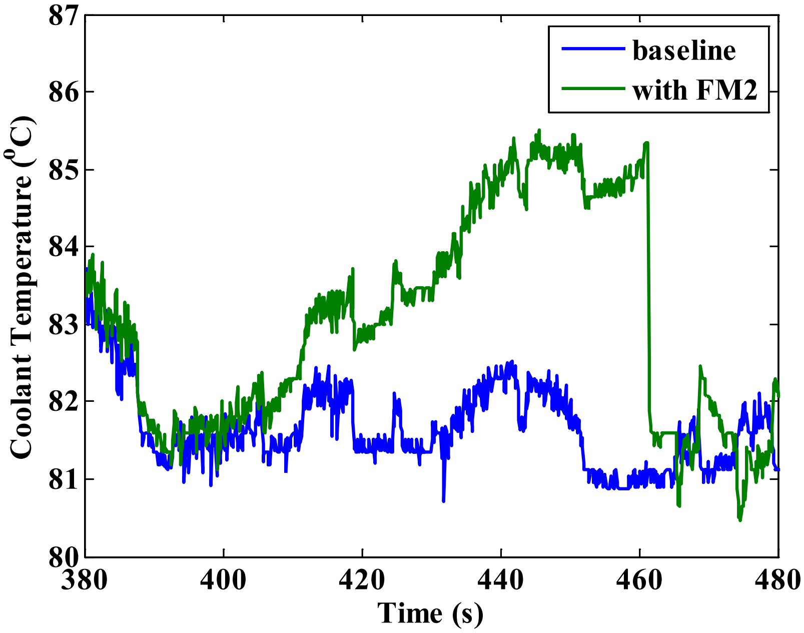


Figure 65: VAT2000 *CT* signal subjected to FM2 failure at 400s.

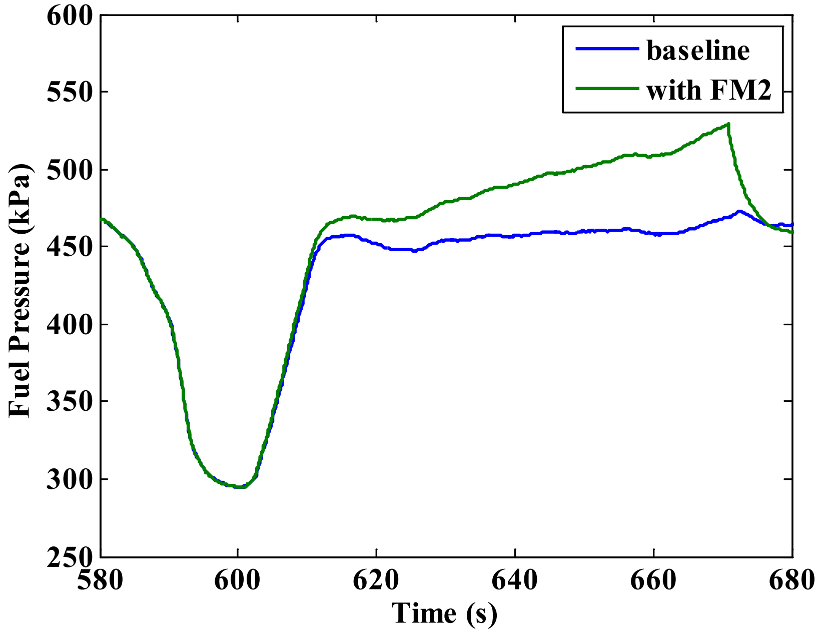


Figure 66: VAT2000 *FP* signal subjected to FM2 failure at 600s.

5.5.3 SFDIA with FM3

It was now necessary to show the SFDIA functionality during instances of intermittent failures. As discussed in Section 5.3.3, this failure mode consisted of pulses resembling a step function that moved back and forth through the thresholds. In addition, the number of pulses was three with the accommodation expected to hold following the third pulse. The setting of the Pulse Generator block (see Figure 12) specified a pulse width, t_w of 20s with the same duration between each pulse. The Pulse Generator blocks added its signals to the sensors at the failure times indicated in Table 9. Figure 67 to Figure 69 below shows the SFDIA performance subjected to the FM3 mode in real-time. All the SFDIA settings were maintained from the offline simulation.

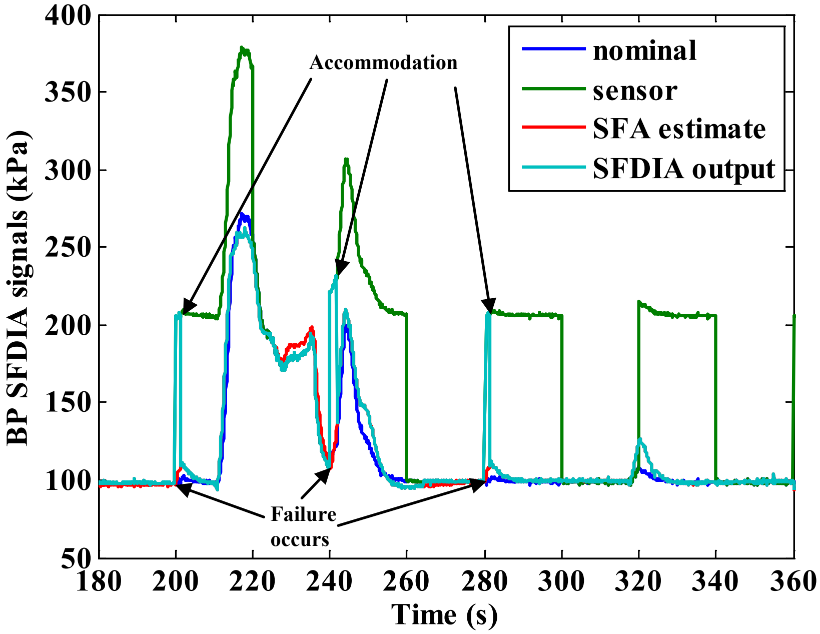


Figure 67: *BP* signal subjected to FM3 in real-time at 200s.

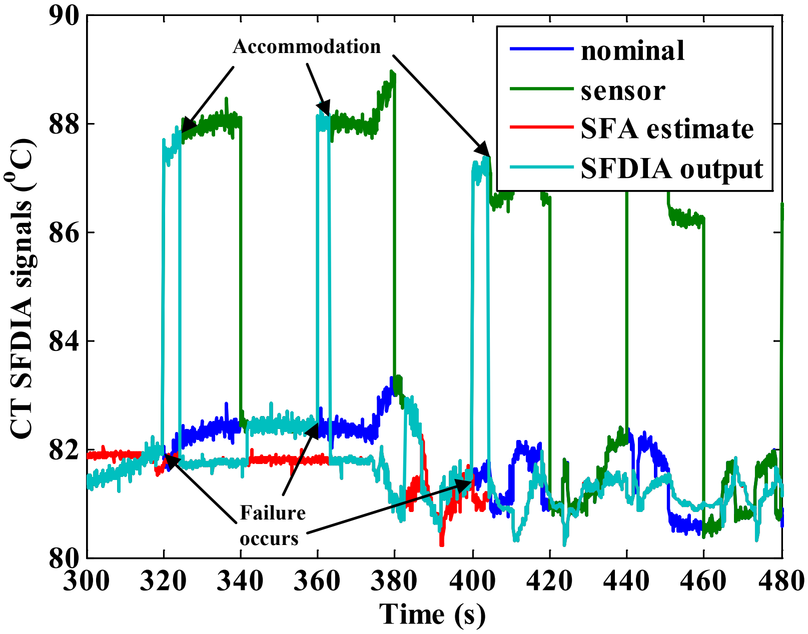


Figure 68: *CT* signal subjected to FM3 in real-time triggered at 300s (started at 320s).

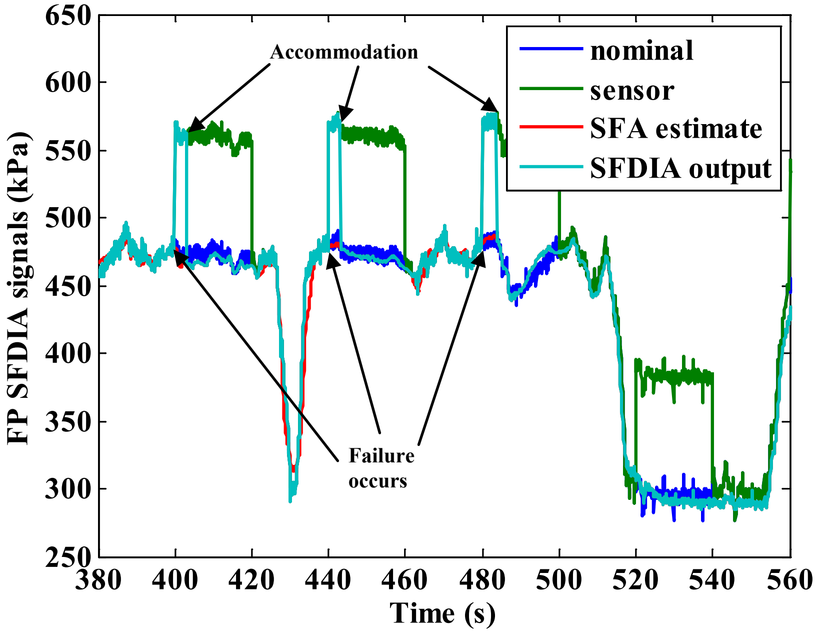


Figure 69: *FP* signal subjected to FM3 in real-time at 400s.

As the first pulse on the *BP* sensor shown in Figure 67 and Figure 70 made the fast residual cross its threshold, all NN learning simultaneously turned off. Moments later, the isolation and accommodation modes kicked in to correct the failures. The second failure pulse also produced similar residual characteristics. When the *BP* sensor accommodation took place at the third pulse, all NN learning permanently ceased. Each of the *BP* failure pulses caused the online learning of the NNs to stop in 0.3 seconds.

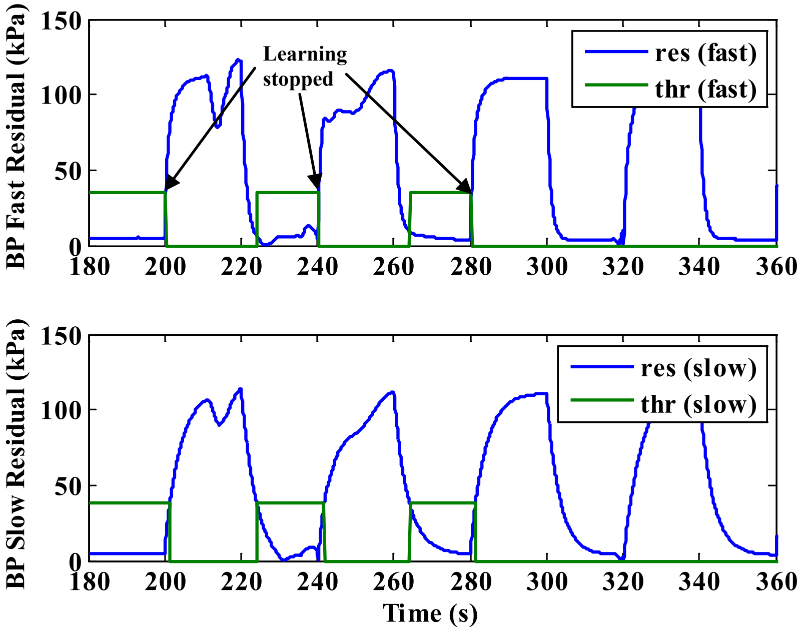


Figure 70: *BP* residuals with FM3 from real-time SFDIA tests.

The same processes took place for the *CT* and *FP* sensors as they failed and their SFDIA signals illustrated in Figure 68 and Figure 69. The residuals for the *CT* and *FP* sensors are shown Figure 125 and Figure 126 (see Appendix G). The residuals indicated that there were delays in restoring the signals to the nominal when the first two pulses ended for all models. This was also an effect observed in the offline simulation, thus expected to happen here. Adjusting the slow residual filter bandwidth and improving the quality of the model predictions can solve this problem. Table 22 summarizes the length of time the *AE* vectors took to enable the accommodations.

Table 22: Duration of accommodation to occur in real-time with FM3.

Parameter	Length of time (s)		
	Pulse 1	Pulse 2	Pulse 3
BP	1.4	1.8	1.4
CT	4.3	3.1	4.2
FP	3.2	3.1	3.7

Table 23 shows the performance of the models during the final accommodation phase. Similar to the previous failures, there was no change in the EMRAN RBF neurons. The *CT* model performed better in the FM3 accommodation phase compared to the other *CT* models from the other failure modes. The fact that the *CT* model spent more time in the learning phase during this failure mode test helped improve the model.

Table 23: Model performance during final accommodation phase (FM3 failure).

Parameters	RMS Error (%)
BP	2.68
CT	12.90
FP	3.70

Likewise, Figure 71 to Figure 73 created using VAT2000 data confirmed that it received the SFDIA output with the FM3 failure pulses present in the sensor signals. As in the step failure case, the EMS processed the *FP* signals to smoothen it and produce triangular shaped pulses at the failure points as opposed to the step variations seen in Figure 69.

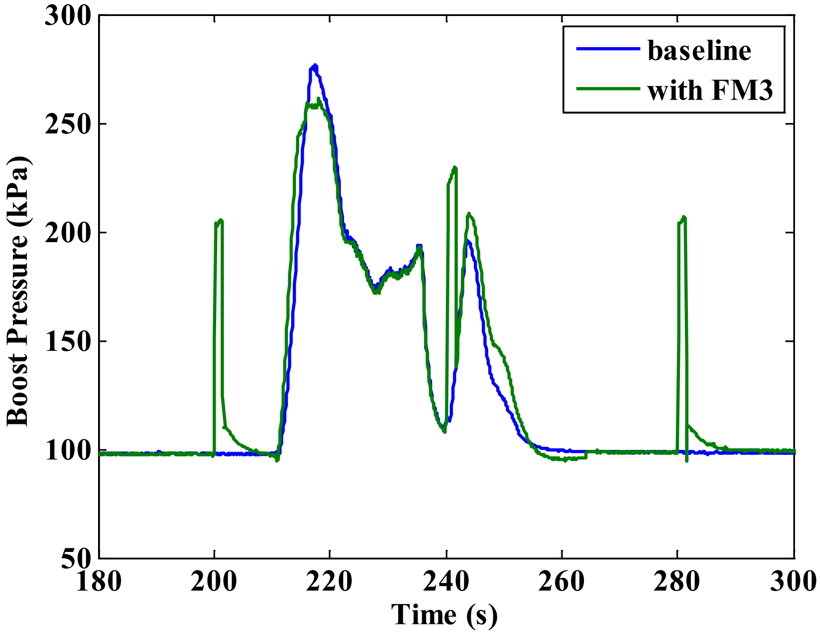


Figure 71: VAT2000 *BP* signal subjected to FM3 failure at 200s.

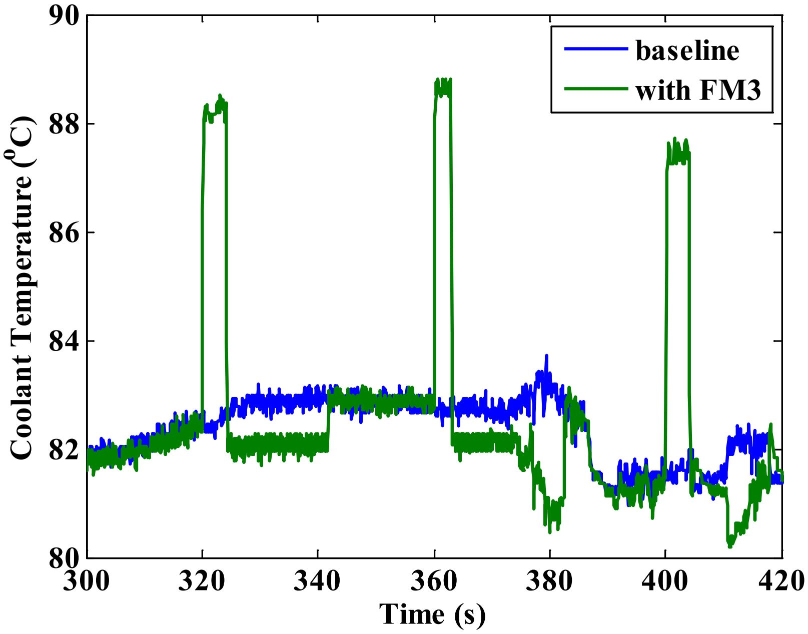


Figure 72: VAT2000 *CT* signal subjected to FM3 failure at 320s.

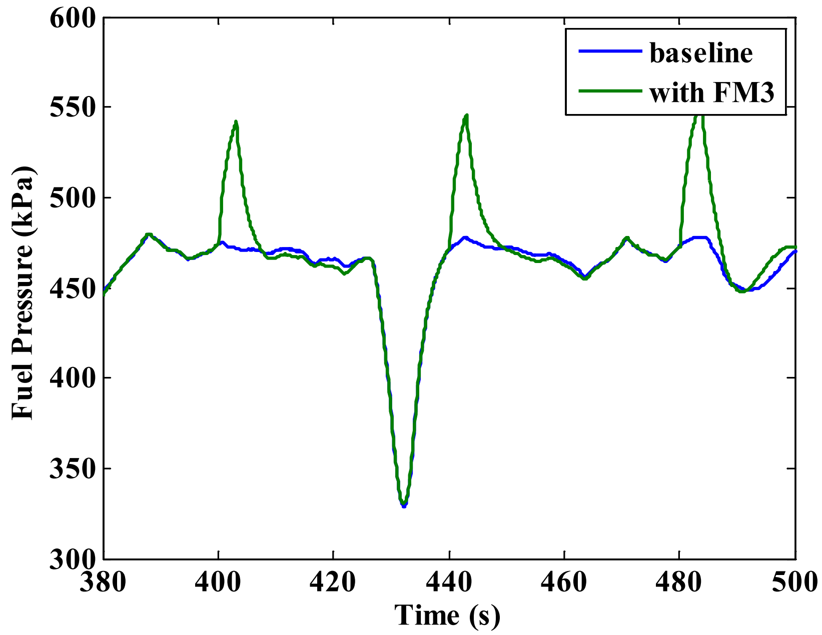


Figure 73: VAT2000 *FP* signal subjected to FM3 failure at 400s.

5.5.4 Effect on Engine Performance

It was the goal that the SFDIA would maintain engine power and emissions during periods of sensor failures. A summary of selected integrated data from the real-time SFDIA tests are provided in Table 24.

Table 24: Integrated data from real-time SFDIA tests.

Run	Total work (bhp-hr)	*HC* (g/bhp-hr)	*NOx* (g/bhp-hr)	*CO* (g/bhp-hr)	CO_2 (g/bhp-hr)
Baseline	24.24	0.139	2.55	0.98	557.41
With FM1	24.29	0.143	2.47	1.22	556.90
With FM2	24.30	0.134	2.31	1.37	557.62
With FM3	24.28	0.137	2.45	1.09	556.73

In comparison to the previous baseline tests, the baseline *CO* data in Table 24 showed a noticeable deviation. Although background variation was a factor, it is not possible to know if this occurrence was an isolated event due only to running a single baseline test. In fact, *NOx* and *CO* showed the most sensitivity to the SFDIA functionality, especially during the ramp failure mode. The duration that each of the three sensors spent learning from the ramp failure signal before the adaptation terminated was enough to influence the integrated data. Therefore, engine performance comparisons were done using the continuous recorded data, in particular during the accommodation phase. For FM3, the comparisons were performed against data collected during the final accommodation phase for simplicity. Baseline data served as the fault-free dataset.

To analyze the continuous emissions data first required estimating the de-skew time for each emissions component. The de-skew time is the length of time the gaseous analyzers take to detect the initial effects in the emissions from the point of inducing a failure. Exhaust gas exiting the turbocharger reached the analyzers via the exhaust-sampling tunnel, heated sampling lines for conditioning purposes and other various lengths of tubing. Estimating this time was possible using data collected from the real-time FM1 test. In particular, this involved locating the initial change in slope within the emissions data caused by the *BP* step failure at 200s. Figure 127 to Figure 130 (see Appendix G) provided a means to perform this estimation and the analyzer de-skew times summarized in Table 25:

Table 25: Gaseous analyzer de-skew time.

Parameters	Time
NOx	6
HC	8.6
CO	6.6
CO_2	6

Calculating the RMS error on the continuous emissions data now involved starting where the *FP* accommodation phase began plus the de-skew time to account for gaseous analyzer delays. In addition, determining the RMS error on continuous engine power indicated how well the engine maintained its performance. Engine power did not require correction for signal delays as the location of the load cell measuring engine torque was on the dynamometer housing. Table 26 shows the results from the error analysis:

Table 26: Emissions comparison from real-time test.

Parameters	RMS Error (%)		
	FM1	FM2	FM3
NOx	1.41	1.36	1.38
HC	4.13	2.13	1.23
CO	1.75	3.37	1.72
CO_2	0.56	0.52	0.56
Engine Power	0.64	0.61	0.64

The RMS errors calculated above used the data seen in Figures 131 to 145 (see Appendix G). The *NOx*, *CO_2* and engine power remained consistent for each accommodation phase. *HC* emissions got better for each consecutive failure mode. From the continuous trace, NOx emissions consistently peaked below the baseline at high load events. Proportionally, *CO* emissions at the same high load points were higher than baseline values. The direct reason for all the variation between baseline and the other tests relate to the performance of the NN models. Another possible cause was that the signals here originated from the DAQ hardware. Two of these sensors are critical to fuel and aftertreatment strategies. Sensor signals sent directly to the EMS will have different characteristics in comparison to the ones output by the DAQ hardware. Therefore, the EMS accordingly processed the signals and this affected the overall fueling and EGR strategies.

5.6 Summary

The first part of this chapter discussed the training and validating strategy for the ADALINE+EMRAN RBF NN. Six NN models created assisted in SFDIA residual generation and for completing the Approximation block. With the models obtained, an offline SFDIA simulation showed that it could perform as designed and could move to the real-time experimental phase. Prior to the real-time test, the breakout panels designed to acquire the sensor signals was verified to ensure it did not interfere with normal engine operation. In addition, data from the engine was acquired to observe the effects of a failed sensor critical to fuel and exhaust aftertreatment strategies. Finally, real-time SFDIA tests were performed with all the selected failure modes. The results showed that it did perform as expected by replacing the failed sensors with signals from the NNs in real-time. Therefore, engine performance was maintained during events of sensor failures by the SFDIA. If a sensor failed but it was still within the current thresholds specified in the SFDI logic, then it will not be detected. Therefore, the thresholds require a more detailed calibration to determine their optimal values to detect as many failure levels as possible.

CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

To improve the reliability of sensor signals in HDDEs, this study demonstrated the SFDIA concept and showed its capability to function in real-time. The study showed that the SFDIA could manage the engine control to maintain nominal performance during instances of sensor failures. It was understood that all critical sensors related to emissions control and engine protection must always be reliable. If the sensors are compromised and not the engine parameter being monitored, there must be a technology available to correct them under all operating conditions.

The sensors chosen to fail in order to prove the SFDIA functionality included boost pressure, coolant temperature and fuel pressure. The failures focused on three different modes and all added into the signals using Simulink. The following list summarizes the failures modes and their description:

a) Step failure - instantaneous bias in the signal such as an open or short in the electrical circuit
b) Ramp failure - representing long-term signal drifts
c) Square wave failure - to model intermittent failures

The SFDIA contained built-in NN models that mathematically represented the three sensors. Model training and validating was possible using data from the test engine collected prior to this study. The NN used had online adaptation capabilities. The choice of the test engine was due to the myriad of sensors available on it. In addition, it had an EMS that was suitable for signal breakout.

Initially, an offline simulation confirmed the SFDIA functionality for testing it in real-time. The offline results showed the NN models closely following the nominal values. For the experimental part, a DAQ system was assembled for signal breakout and ensuring robust signals from the test engine. Prior to the real-time SFDIA test, a study was done to show the consequences of a sensor failure on the test engine control strategy.

The final portion of the work involved obtaining real-time performance data of the SFDIA scheme. When a failure occurred, the SFDIA detected it and proceeded to terminate any adaptation activities. As the failure continued, the scheme logic isolated the fault and replaced the failed signals with the NN outputs. The replacement of the failed signal with those from the NN is called accommodation. In the case of intermittent failures, the logic designed to detect it functioned as expected and permanently accommodated the failure after a specified number of pulses. For the entire experimental portion, test data were collected from Simulink, the laboratory system and software that read the EMS signals. There was indication of degradation in model accuracy when transferring it to the experimental part. This happened since the data for training the NNs originated from VAT2000, while the models used signals from the DAQ system put together for this study. It was the goal to show the functionality of the SFDIA as opposed to trying to generate the best models. The NN adaptation capabilities prevented the degradation from rendering the models ineffective.

The main novelty of this proof-of-concept study was the development and setup of the SFDIA, a concept originating from an aerospace application, for testing in real-time, inline with an HDDE. Based on the real-time results, it showed that the SFDIA could function as designed on HDDEs. It can enhance an OBD system of a vehicle to correct sensor failures when they happen to maintain normal performance.

6.2 Recommendations

With this proof-of-concept study completed, it is possible to further this study to address various issues revealed during the course of this research. They are, but not limited to, the following:

a) Improvement in model accuracy and determine the optimal input parameters for a given model. Performing this optimization can reduce any computational overhead that occurred in this research for a more efficient scheme. Having models that best replicate the physical sensors is desirable to provide similar engine control.

b) Relative to the previous item, it will be necessary to find out the appropriate learning rate and how long a duration should the adaptation be active for a given engine model. The specified learning rates should allow the model to learn as much as possible before safely deactivating within the shortest amount of time. It is noted that the adaptation process does take up additional computational resources. Thus, terminating the learning after the models have sufficiently adapted to all current operation conditions is recommended. However, at some point it should reactivate to adapt to any alterations due to engine aging.

c) For any failure modes, determine the best thresholds to control the model adaptations and trigger the accommodations without the risk of false alarms. In the case of intermittent failures, improve the time it takes for the scheme to restore the nominal signals if the failure ends. In addition, a unified SFDI logic should preferably detect any failure mode, as opposed to the separate logics presented here.

d) Involve manufacturer participation to provide the SFDIA with CAN bus signals. An advantage in using the CAN protocol is the use of a single DAQ device, known as a CAN adapter. Such devices, as seen in Figure 74, would interface to a computer via a standard USB port. This eliminates any unwanted influence from DAQ hardware such as the ones used in this work.

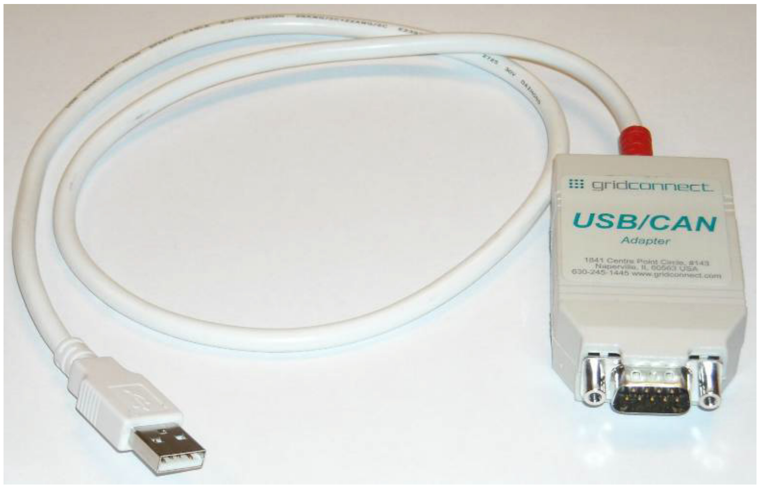


Figure 74: USB/CAN adapter by Grid Connect™ Inc.

In addition, the signals within the protocol are already time synchronized relative to the engine. The Data Acquisition Toolbox does have time synchronization capabilities. However, it is relative to the signals as the DAQ cards received them. Manufacturer participation also means the possibility of embedding the scheme directly into the EMS for testing.

e) Perform durability study to observe the robustness of the SFDIA to function on a long-term basis.

REFERENCES

1. "2010 Emissions Choosing The Right Technology", Cummins Inc., Jan 2009.
2. "Final Regulation: Order On-Board Diagnostic System Requirements--2010 and Subsequent Model-Year Heavy-Duty Engines", Title 13, California Code of Regulations, Section 1971.1, May 2010.
3. "EPA Finalizes Regulations Requiring Onboard Diagnostic Systems on 2010 and Later Heavy-Duty Engines Used in Highway Applications Over 14,000 Pounds; Revisions to Onboard Diagnostic Requirements for Diesel Highway Heavy-duty Applications Under 14,000 Pounds", Office of Transportation and Air Quality, EPA-420-F-08-032, Dec 2008.
4. Napolitano, M. R., Molinaro, G., Innocenti, M., Seanor, B., Martinelli, D., "A Complete Hardware Package for a Fault Tolerant Flight Control System Using Online Learning Neural Networks", Proceedings of the 1999 American Control Conference, Vol. 4, pp. 2615-2619, 1999.
5. Schihl, P., "Control Strategies for Heavy-Duty Diesel Engine Emissions", IEEE Instrumentation & Measurement Magazine, Vol. 4, No. 2, pp 11-15, Jun 2001.
6. Heywood, J. B., "Internal Combustion Engine Fundamentals", McGraw-Hill Series in Mechanical Engineering, New York, 1992.
7. Guzzella, L., Amstutz, A., "Control of Diesel Engines", IEEE Control Systems, Vol. 18, No. 5, pp. 53-71, 1998.
8. Chen L., Li G. X., Liu, Y., Lu, J., Wang G., Zhang X., "A Developed Model for Emissions Prediction of a DI Diesel Engine", SAE 1999-01-0233, Warrendale, PA, 1999.
9. Kee, R. J., Rupley, F. M., Miller, J. A., "CHEMKIN-II: A FORTRAN Chemical Kinetics Package for the Analysis of Gas-Phase Chemical Kinetics," Sandia National Laboratories Report, SAND 89-8009, 1989.
10. Schilling, A., Amstutz, A., Guzzella, L., "Model-Based Detection and Isolation of Faults Due to Ageing in the Air and Fuel Paths of Common-rail Direct Injection Diesel Engines Equipped with a λ and a Nitrogen Oxides Sensor", Proceedings of the IMechE Part D: Journal of Automobile Engineering, Vol. 222, No. 1, pp. 101-117, 2008.

11. Napolitano, M. R., Kincheloe, M., "On-line Learning Neural Network Controllers for Autopilot Systems", Journal of Guidance, Control, and Dynamics, Vol. 18, No. 5, pp. 1008-1015, Sep-Oct 1995.
12. Dinca, L., Aldemir, T., Rizzoni, G., "A Model-Based Probabilistic Approach for Fault Detection and Identification with Application to the Diagnosis of Automotive Engines", IEEE Transactions on Automatic Control, Vol. 44, No. 11, pp. 2200-2205, Nov 1999.
13. Laukonen, E. G., Passino, K. M., Krishnaswami, V., Luh, G. -C., Rizzoni, G., "Fault Detection and Isolation for an Experimental Internal Combustion Engine via Fuzzy Identification", IEEE Transactions on Control Systems Technology, Vol. 3, No. 3, pp. 347-355, Sep 1995.
14. Bartzoudis, N., McDonald-Maier, K., "An Adaptive Processing Node Architecture for Validating Sensors Reliability in a Wind Farm", 2007 ECSIS Symposium on Bio-inspired, Learning, and Intelligent Systems for Security, IEEE, pp. 83-86, 2007.
15. Patton R. J., Frank P. M., Clark R. N., "Fault Diagnosis in Dynamic Systems, Theory and Applications", Englewood Cliff, NJ, Prentice-Hall, 1989.
16. Abdelghani, M., Friswell, M. I., "Sensor Validation for Structural Systems with Multiplicative Sensor Faults", Mechanical Systems and Signal Processing, Vol. 21, No. 1, pp. 270-279, Jan 2007.
17. Baraldi, P., Zio1, E., Gola, G., Roverso, D., Hoffmann, M., "Genetic Algorithms for Signal Grouping in Sensor Validation: A Comparison of the Filter and Wrapper Approaches", Proceedings of the IMechE Part O: Journal of Risk and Reliability, Vol. 222, No. 2, pp. 189-205, 2008.
18. Kusiak, A., Song, Z., "Sensor Fault Detection in Power Plants", Journal of Energy Engineering, Vol. 135, No. 4, pp. 127-137, Dec 2009.
19. Qin, S. J., Yue, H., Dunia, R., "A Self-validating Inferential Sensor for Emission Monitoring", Proceedings of the 1997 American Control Conference, Vol. 1, pp. 473-477, 1997.
20. Cho, J. H., Lee, J. M., Choi, S. W., Lee, D. K., Lee, I. B., "Sensor Fault Identification Based on Kernel Principal Component Analysis" Proceedings of the 2004 IEEE International Conference on Control Applications, Vol. 2, pp. p 1223-1228, 2004.

21. Lee, J. M., Yoo, C. K., Choi, S. W., Vanrolleghem, P. A., Lee, I. B. "Nonlinear Process Monitoring Using Kernel Principal Component Analysis" Chemical Engineering Science, Vol. 59, No. 1, pp. 223-234, 2004.

22. Dong, D., McAvoy, T. J., Chang, L. J., "Emission Monitoring Using Multivariate Soft Sensors", Proceedings of the American Control Conference, Vol. 1, pp. 761-765, 1995.

23. Wang, X., Kruger, U., Irwin, G. W., McCullough, G., McDowell, N., "Nonlinear PCA With the Local Approach for Diesel Engine Fault Detection and Diagnosis", IEEE Transactions on Control Systems Technology, Vol. 16, No. 1, pp. 122-129, Jan 2008.

24. Crowe, C. M., "Data Reconciliation - Progress and Challenges", Journal of Process Control, Vol. 6, No. 2-3, pp. 89-98, Apr-Jun 1996.

25. Ibargüengoytia, P. H., Sucar, L. E., Vadera, S., "Real Time Intelligent Sensor Validation", IEEE Transactions on Power Systems, Vol. 16, No. 4, pp. 770-775, Nov 2001.

26. Mehranbod, N., Soroush, M., "Probabilistic Model for Sensor Fault Detection and Identification", AIChE Journal, Vol. 49, No. 7, pp. 1787-1802, Jul 2003.

27. Mehranbod, N., Soroush, M., Panjapornpon, C., "A method of sensor fault detection and identification", Journal of Process Control, Vol. 15, No. 3, pp. 321-339, Apr 2005.

28. Chen, X., Schmid, N. A., Wang, L., Clark, N. N., "Regression-Based Oxides of Nitrogen Predictors for Three Diesel Engine Technologies", The Journal of the Air & Waste Management Association, Vol. 60, No. 1, pp. 72-90, Jan 2010.

29. Zareipour, H., Bhattacharya, K., Cañizares, C. A., "Forecasting the Hourly Ontario Energy Price by Multivariate Adaptive Regression Splines", 2006 IEEE Power Engineering Society General Meeting, Montreal, QC, 2006.

30. Steinberg, D., "MARS User Guide", San Diego, CA: Salford Systems, 2001.

31. Hastie, T., Tibshirani, R., Friedman, J., "The Elements of Statistical Learning: Data Mining, Inference and Prediction," Springer-Verlag, San Diego, CA, 2001.

32. Chew, V., "Sensor Validation Scheme with Virtual NOx Sensing for Heavy Duty Diesel Engines", MS Thesis, Department of Mechanical and Aerospace Engineering, West Virginia University, Morgantown, WV, 2007.

33. Collinson, R. P. G., "Fly-by-Wire Flight Control", Computing & Control Engineering Journal, Vol. 10, No. 4, pp. 141-152, Aug 1999.

34. Napolitano, M. R., Chen, C. I., Naylor, S., "Aircraft Failure Detection and Identification using Neural Networks", Journal of Guidance, Control, and Dynamics, Vol. 16, No. 6, pp. 999-1009, Nov-Dec1993.

35. Campa, G., Fravolini, M. L., Napolitano, M. R., "A Neural Network Based Tool for Aircraft SFDIA Modeling and Simulation", Proceedings of the IASTED Conference, Pittsburgh, PA, May 2001.

36. Wei, T., Huang, Y., Chen, C. L. P., "Adaptive Sensor Fault Detection and Identification Using Particle Filter Algorithms", IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews), Vol. 39, No. 2, pp. 201-213, Mar 2009.

37. Yerramalla, S., Fuller, E., Cukic, B., "A Validation Approach for Neural Network-Based Online Adaptive Systems", Software - Practice and Experience, Vol. 36, No. 11-12, pp. 1209-1225, Sep-Oct 2006.

38. Guo, O. A., Litt, J. S., "Integrated Flight and Propulsion Control Research for the NASA Integrated Resilient Aircraft Control (IRAC)", Workshop at Ohio Aerospace Institute, Cleveland, OH, Nov 6-7, 2007.

39. Litt, J. S., "Engine Performance Deterioration Mitigating Control", Workshop at Ohio Aerospace Institute, Cleveland, OH, Nov 6-7, 2007.

40. Monteriù, A., Asthana, P., Valavanis, K. P., Longhi, S., "Real-Time Model-Based Fault Detection and Isolation for UGVs", Journal of Intelligent and Robotic Systems: Theory and Applications, Vol. 56, No. 4, pp. 425-439, 2009.

41. Bonnick, A. W. M., "Automotive Computer Controlled Systems: Diagnostic tools and techniques", Oxford, Butterworth-Heinemann, 2001.

42. Wong, J., "CARB Heavy-Duty OBD Update", SAE OBD TOPTEC, Pasadena, CA, Sep 2005.

43. Wang, C., Zhang, Y., Zhong, Z., "Fault Diagnosis for Diesel Valve Trains Based on Time-Frequency Images", Mechanical Systems and Signal Processing, Vol. 22, No. 8, pp. 1981-1993, Nov 2008.

44. Xiaochun, S., Hongying, H., "Diesel Engine Fault Diagnosis and Classification", 8th International Conference on Signal Processing, Vol. 1, 2007.

45. Isermann, R., Schwarte, A., "Fault Diagnosis of a Turbocharged Diesel Engine with Dynamic Neural Networks and Parity Methods", Proceedings of the IASTED International Conference on Intelligent Systems and Control, pp. 283-288, 2004.
46. Nyberg, M., "Model-Based Diagnosis of an Automotive Engine Using Several Types of Fault Models", IEEE Transactions on Control Systems Technology, Vol. 10, No. 5, pp. 679-689, Sep 2002.
47. Antory, D., "Fault Diagnosis Application in an Automotive Diesel Engine Using Auto-Associative Neural Networks", Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, Vol. 2, pp. 109-116, 2005.
48. Campa, G., Thiagarajan, M., Krishnamurty, M., Napolitano, M. R., Gautam, M., "A Neural Network Based Sensor Validation Scheme for Heavy-Duty Diesel Engines", Journal of Dynamic Systems, Measurement and Control, Vol. 130, No. 2, 021008 (10 pp.), Mar 2008.
49. Mesbahi, E., "An Intelligent Sensor Validation and Fault Diagnostic Technique for Diesel Engines", Journal of Dynamic Systems, Measurement and Control, Vol. 123, No. 1, pp. 141-144, Mar 2001.
50. Capriglione, D., Liguori, C., Pianese, C., Pietrosanto, A., "On-Line Sensor Fault Detection, Isolation, and Accommodation in Automotive Engines", IEEE Transactions on Instrumentation and Measurement, Vol. 52, No. 4, pp. 1182-1189, Aug 2003.
51. Capriglione, D., Liguori, C., Pietrosanto, A., "Real-Time Implementation of IFDIA Scheme in Automotive Systems", IEEE Transactions on Instrumentation and Measurement, Vol. 56, No. 3, pp. 824-830, Jun 2007.
52. Martincic, F., Schwiebert, L., "Introduction to Wireless Sensor Networking", Handbook of Sensors: Algorithms and Architectures, Stojmenovic, I. (Ed), Wiley, New Jersey, 2005.
53. Kosut, O., Turovsky, A., Sun, J., Ezovski, M., Tong, L., Whipps, G., "Integrated mobile and static sensing for target tracking", Proceedings - IEEE Military Communications Conference, Orlando, FL, 2007.
54. Hou, J. C., Li, N., Stojmenovic, I., "Topology Construction and Maintenance in Wireless Sensor Networks", Handbook of Sensors: Algorithms and Architectures, Stojmenovic, I. (Ed), Wiley, New Jersey, 2005.

55. Culley, D., “Distributed Engine Control”, Workshop at Ohio Aerospace Institute, Cleveland, OH, Nov 6-7, 2007.

56. Song, E. Y., Lee, K. B., “Sensor Network Based on IEEE 1451.0 and IEEE p1451.2-RS232”, IEEE Instrumentation and Measurement Technology Conference, pp. 1728-1733, 2008.

57. Gaura, E., Kraft, M., “Are Neural Network Techniques the Solution to Measurement Validation, Monitoring and Automatic Diagnosis of Sensor Faults?”, Proceedings of the 41st SICE Annual Conference, Vol. 3, pp. 2052-2057, 2002.

58. Fatemi, M., Roopaei, M., Shabaninia, F., “New Training Methods for RBF Neural Networks”, Proceedings of the 2005 International Conference on Neural Networks and Brain, Vol. 3, pp. 1322-1327, 2005.

59. Fausett, L., “Fundamentals of Neural Networks: Architecture, Algorithms and Application”, Prentice Hall, New Jersey, 1994.

60. Eberhart, R., Simpson, P., Dobbins, R. “Computational Intelligence PC Tools”, AP Professional, Academic Press, Inc., 1996.

61. Sanner R. M., Slotine J. J. E., “Gaussian Networks for Direct Adaptive Control”, IEEE Transactions on Neural Networks, Vol. 3, No. 6, pp. 837-863, 1992.

62. Fabbri S, Kadirkamanathan V., “Dynamic Structure Neural Networks for Stable Adaptive Control of Nonlinear Systems”, IEEE Transactions on Neural Networks, Vol. 7, No. 5, pp. 1151-1167, 1996.

63. Park, J., Sandberg, I. W., “Universal Approximation Using Radial-Basis-Function Networks”, Neural Computation, Vol. 3, pp. 246-257, 1991.

64. Aggarwal, C. C., “On the Effects of Dimensionality Reduction on High Dimensional Similarity Search”, Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 256-266, 2001.

65. Lu Y., Sundararajan N., Saratchandran P., “Analysis of Minimal Radial Basis Function Network Algorithm for Real-Time Identification of Non-Linear Dynamic Systems”, IEEE Proceedings on Control Theory and Application, Vol. 4, No. 147, pp. 476, 2000.

66. Campa, G., Fravolini, M. L., Napolitano, M. R., “A Library of Adaptive Neural Networks for Control Purposes”, IEEE International Symposium on Computer Aided Control System Design Proceedings, Glasgow, Scotland, U.K., Sep 2002.

67. “Vendor Application Guide (V-MAC IV), 2004 Emissions Std.”, Publication 8-359, Mack Trucks Inc., 2007.

68. “Using Simulink”, The Mathworks, 2005 (http://www.mathworks.com).

69. Fravolini, M. L., Campa, G., Napolitano, M. R., Perhinschi, M. G., “A Learning Based Sensor Validation Scheme Within Flight Control Laws”, Journal of Guidance, Control, and Dynamics, Vol. 27, No. 2, pp. 307-310, Mar-Apr 2004.

70. Blanke, M., Frei, C., Kraus, K., Patton, R. J., Staroswiecki, M., “Fault-Tolerant Control of Complex Systems,” Control of Complex Systems, Aström, K. J., Albertos, P., Blanke, M., Isidori, A., Schaufelberger, W., Sanz, R. (Eds.), Springer, New York, 2000.

71. Frank, P. M., Ding, X., “Survey of Robust Residual Generation and Evaluation Methods in Observer-Based Fault Detection Systems”, Journal of Process Control, Vol. 7, No. 6, pp. 403-424, Dec 1997.

72. Campa, G., “ANN,” 2008, (http://www.mathworks.com/matlabcentral/fileexchange/976-ann).

73. “Mack MP7 Diesel Engine Service Manual (’04 Emissions Regulations)”, Publication 5-112, Mack Trucks Inc., 2007.

74. “V-MAC IV Diagnostic Service Manual, 2004 Emissions Standard”, Publication 8-213, Mack Trucks Inc., 2007.

75. Code of Federal Regulations, Title 40, Part 1065 - Engine Testing Procedures, 2008.

76. Dickey, S. R., Lu, X-Y., “Control Actuation of Fully Automated Heavy-Duty Vehicles Using SAE J1939”, IEEE IV2003 Proceedings of the Intelligent Vehicles Symposium, pp. 400-405, 2003.

77. Gautam, M., Clark, N. N., Thompson, G. J., Carder, D. K., Lyons, D. W., “Evaluation of Mobile Monitoring Technologies for Heavy-Duty Diesel-Powered Vehicle Emissions,” Phase I Report, Submitted to the Settling Heavy-Duty Diesel Engine Manufacturers by West Virginia University, Morgantown, WV, Mar 2000.

78. Code of Federal Regulations, Title 40, Part 86, Subpart N - Emission Regulations for New Otto - Cycle and Diesel Heavy-Duty Engines; Gaseous and Particulate Exhaust Test Procedures, 2001.

79. “Mack Engine MP7-355E”, A Sales Engineering Publication, Mack Trucks Inc., Oct 2005.

80. "User's Manual: The 3B Series Signal Conditioning I/O Subsystem", Analog Devices Inc., 1986.

81. "VAT2000 Volvo Application Tool, Version 1.0: User Handbook", Volvo Car Corporation, Mar 1996.

82. Daga, L, "RT Blockset", 2008 (http://leonardodaga.insyde.it/Simulink/RTBlockset.htm).

83. ASTM Standard E178, 2008, "Standard Practice for Dealing With Outlying Observations", ASTM International, West Conshohocken, PA, 2008, DOI: 10.1520/E0178-08, www.astm.org.

APPENDIX A FAILURE AND SFDI BLOCKS

A.1 Failure Block

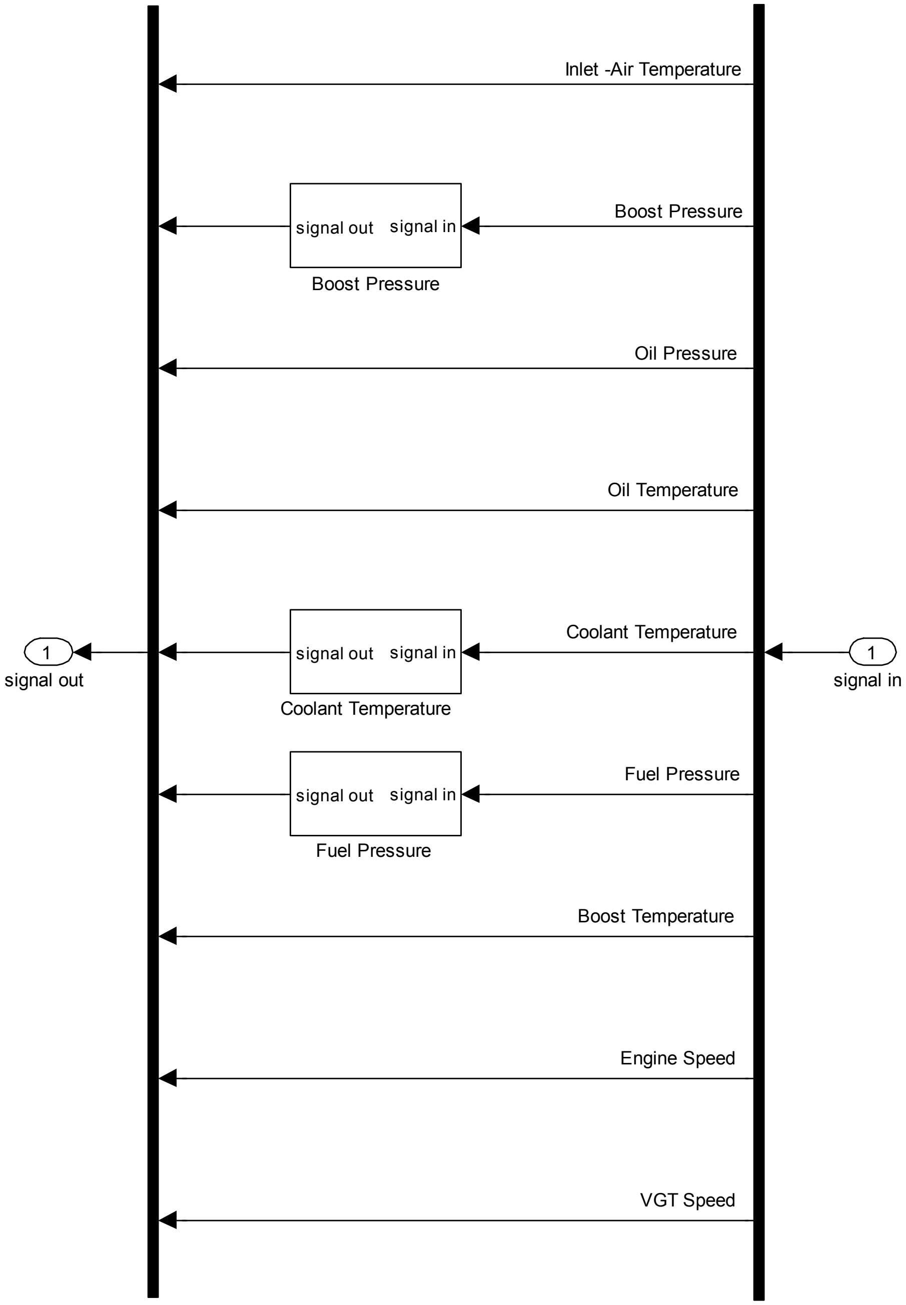


Figure 75: Failure model.

A.2 SFDI Blocks

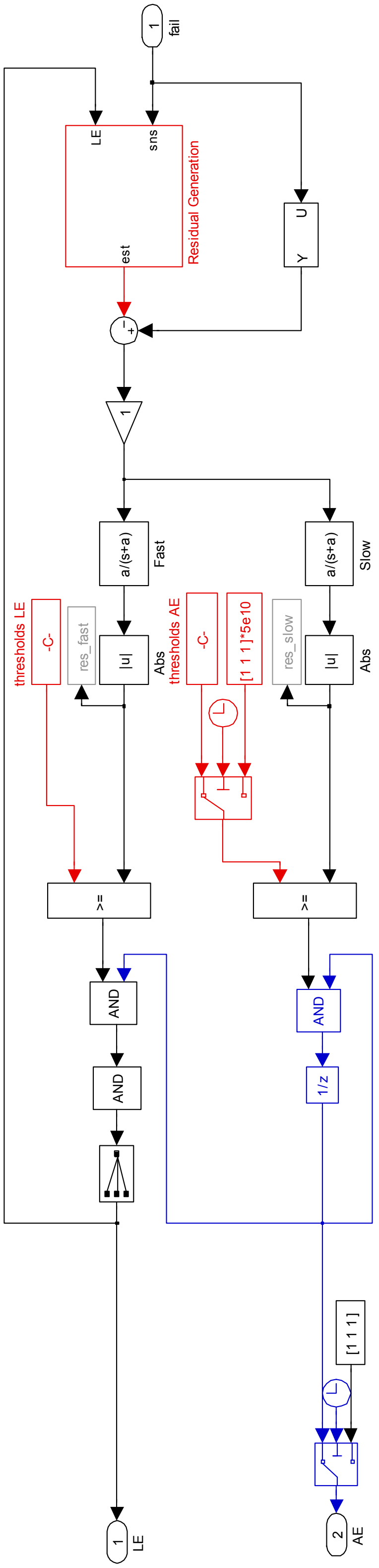


Figure 76: SFDI Logic for the FM1 and FM2.

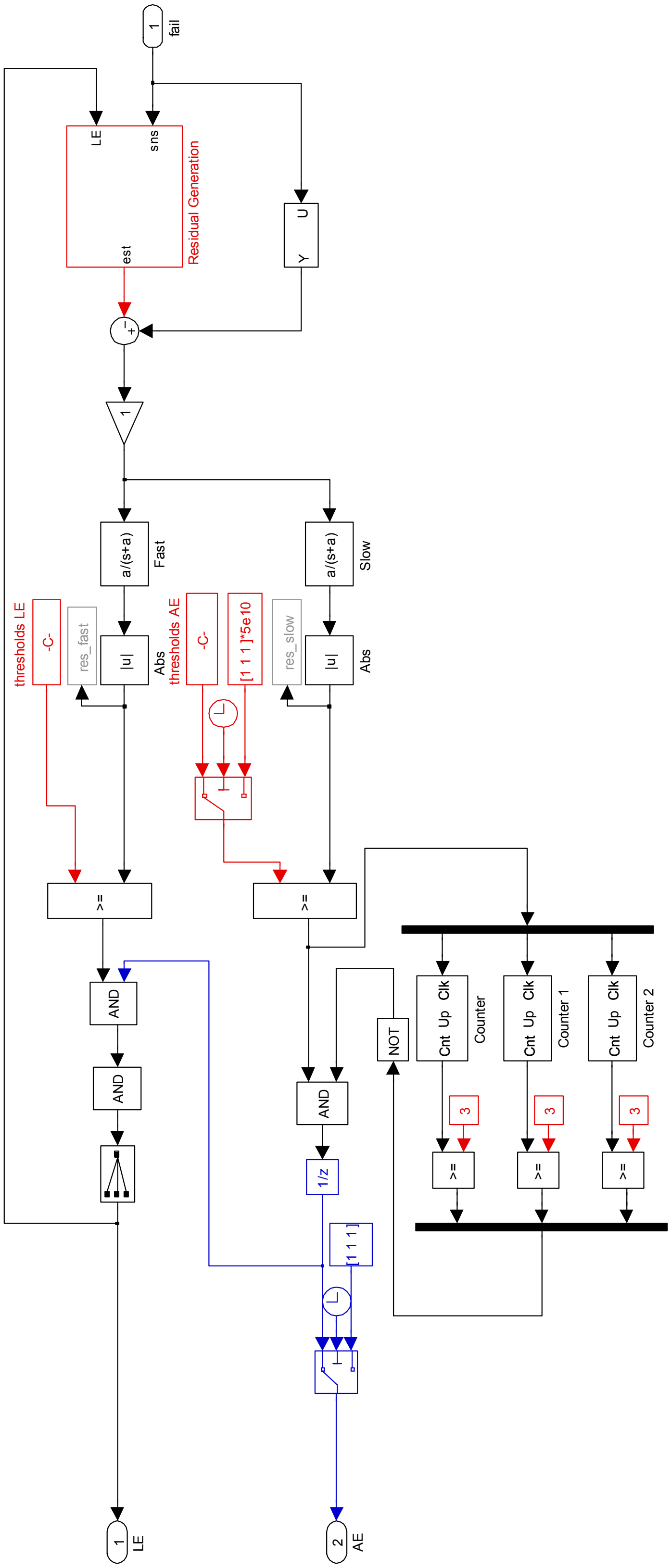


Figure 77: SFDI Logic for FM3.

APPENDIX B 2004 MACK® MP7-355E

Table 27: Manufacturer specifications.

Engine Type	Direct-Injection Diesel
Configuration	In-Line 6 Cylinder
Engine Output	355 hp @ 1800 RPM / 1360 ft-lbf @ 1200 RPM
Displacement	659 in^3 (11 L)
Bore x Stroke	4.84 in x 5.98 in
Fuel Injection	Dual Solenoid EUI
Inlet Depression	9.6 inH_2O @ Rated Power
Exhaust Backpressure	27.5 inH_2O @ Rated Power

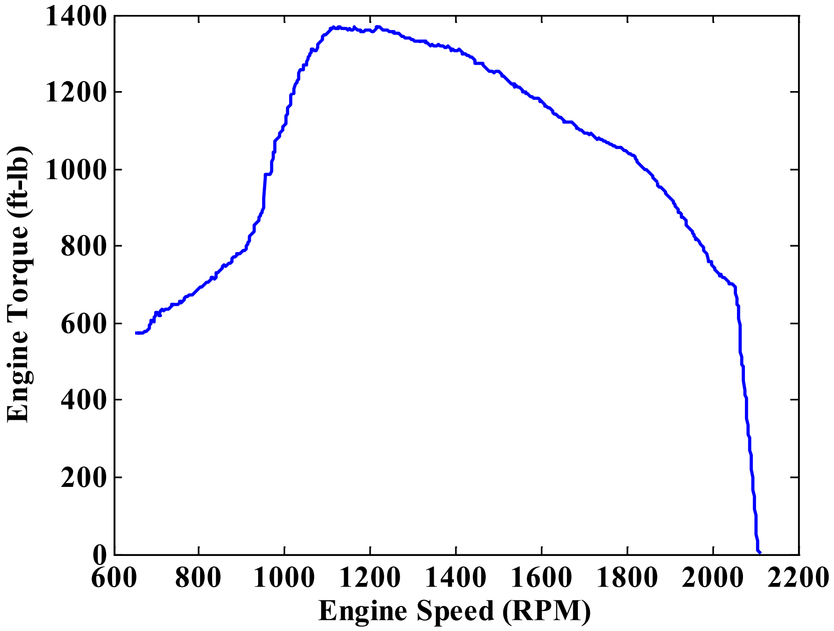


Figure 78: Performance curve of test engine obtained without breakout panels.

Table 28: Results from engine map cycle.

Curb Idle Speed	652 RPM
High Idle Speed	2100 RPM
Rated Speed	1806 RPM
Power at Rated Speed	360 hp
Peak Torque	1379 ft-lb

APPENDIX C ENGINE AND EMISSIONS RESEARCH LABORATORY (EERL)

The Engine and Emissions Research Laboratory (EERL) is located at West Virginia University in the town of Morgantown, WV. It is part of the Center for Alternative Fuels, Engines and Emissions (CAFEE). The testing capabilities at the EERL include light, medium and heavy-duty engines using both steady state and transient duty cycles. The laboratory has various types of dynamometers that can test a broad range of internal combustion engines regardless of manufacturer and size. Full-scale exhaust dilution tunnels allow the EERL to perform emissions certification testing that conforms to the 40 CFR Parts 86, 89, 92, and 1065. Gaseous samples from the tunnels flow through microprocessor-controlled heated probes and sampling lines to various gas analyzers in order to measure the emissions. *PM* sample collection is possible using Teflo® or fluorocarbon coated glass fiber filters. The EERL has a class 1000 clean room for proper conditioning and performing gravimetric analysis on *PM* filters with a high precision microbalance based on Part 1065. Data sampling rate for transient testing on the Part 1065 based tunnel is 10 Hz. The laboratory also has high-speed data acquisition equipment for in-cylinder pressure measurement to perform combustion analysis. Apart from certification testing, the laboratory can serve as a facility for developing and validating new engine and aftertreatment technology. Recent work in new aftertreatment technology used both an SCR and DPF with built-in diesel oxidation catalysts (DOC) to determine an emissions reduction strategy to meet EPA 2010 emissions. Below is a partial list of equipment available at the EERL:

1. General Electric 800 hp motoring/absorbing direct current (DC) dynamometer
2. General Electric 550 hp motoring/absorbing direct current (DC) dynamometer
3. Eaton® 300 hp motoring/absorbing alternating current (AC) dynamometer (rebuilt by Medsker Electric Inc.®)
4. Mustang® 400 hp, 300 hp, 100 hp eddy-current dynamometers
5. Go-Power® 800 hp, 2000 ft-lb torque capacity water-brake dynamometers
6. Two tunnel systems:
    - Exhaust dilution tunnel with variable speed blower and subsonic CVS venturi (based on 40 CFR Part 1065)

- Total exhaust double-dilution tunnel based on critical flow venture - CVS (CFV-CVS) concept (40 CFR Part 86)

7. MEXA-7200D
8. Horiba and Rosemount® 402 HFID *HC* analyzers, Eco Physics and Rosemount 955 chemiluminescent *NOx*/ammonia (NH_3) analyzers, Horiba NDIR *CO*/*CO2* analyzers, and ABB Limas 11 HW non-dispersive ultraviolet (NDUV) *NOx*/NH_3 analyzer
9. Proportional sampling systems for diesel, gasoline and alternative fuel exhaust gas speciation (methane (CH_4), non-methane hydrocarbons (*NMHC*), 1,3-Butadiene, benzene, toluene, ethylbenzene and xylene, volatile organic hydrocarbons, *PAH*, nitro-PAH, carbonyls, polychlorodibenzo-p-dioxins, polychlorodibenzo-p-furans, inorganic ions)
10. Scanning Mobility Particle Sizer (SMPS™)
11. Micro-Orifice Uniform Deposit Impactor (MOUDI™) particle sizer
12. Engine Exhaust Particle Sizer (EEPS™)
13. Mini- and micro-dilution tunnels for particulate sampling and sizing
14. PM_{10}, $PM_{2.5}$, $PM_{1.0}$ cyclone samplers
15. Tapered Element Oscillating Microbalance (TEOM™)
16. Cambustion™ DMS-500 Fast Particulate Spectrometer
17. Cambustion CLD-500 Fast Time Response NOx analyzer
18. Chemical characterization - gas chromatographs (GC), gas-chromatograph/mass spectrometers (GC-MS), thermogravimetric analyzers (TGA), TGA mass spectrometer (TGA-MS), ion chromatography
19. In-line fuel flow metering (MAX Fuel Flow Meter, AVL 735S Fuel Mass Flow Meter, AVL 753C Fuel Temperature Control)
20. Exhaust particulate matter bioassay sampling capability (Ames test, sister chromatid exchange, unscheduled DNA synthesis, micronucleus induction, HGPRT mutation)
21. High-speed data acquisition, control and archival systems

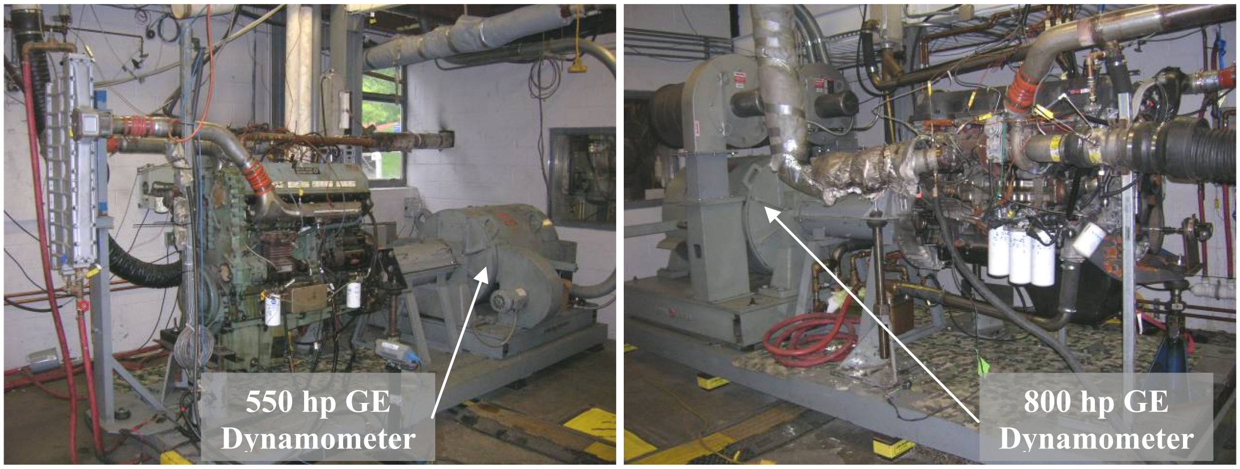


Figure 79: A view of the EERL test cell.

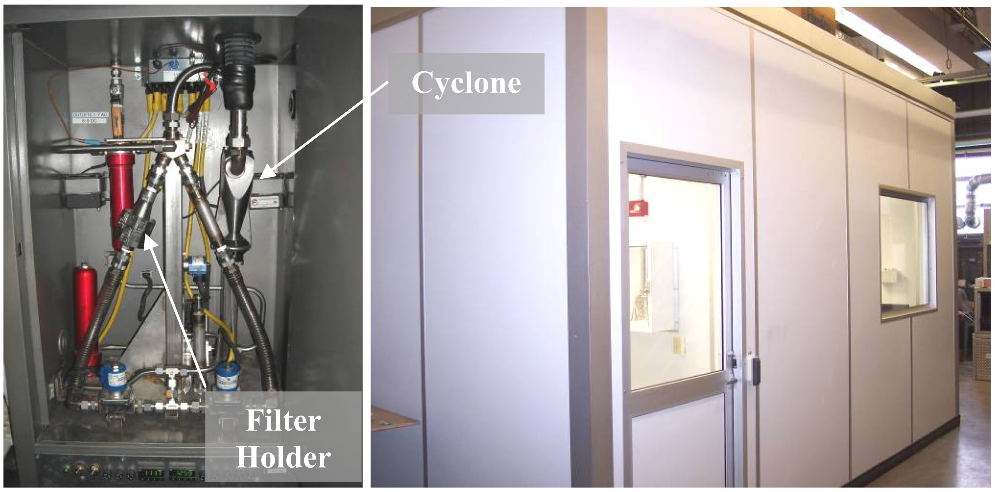


Figure 80: *PM* filter holder box and the clean room for gravimetric analysis.

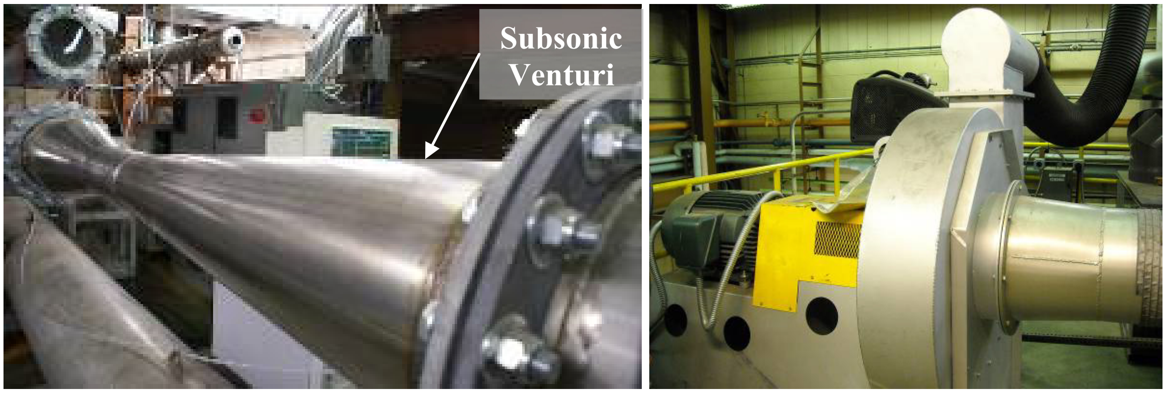


Figure 81: Part 1065 tunnel subsonic venturi used in this study along with the variable speed blower.

APPENDIX D SELECTED TRAINING AND VALIDATING DATA

D.1 Setpoints of the Duty Cycles

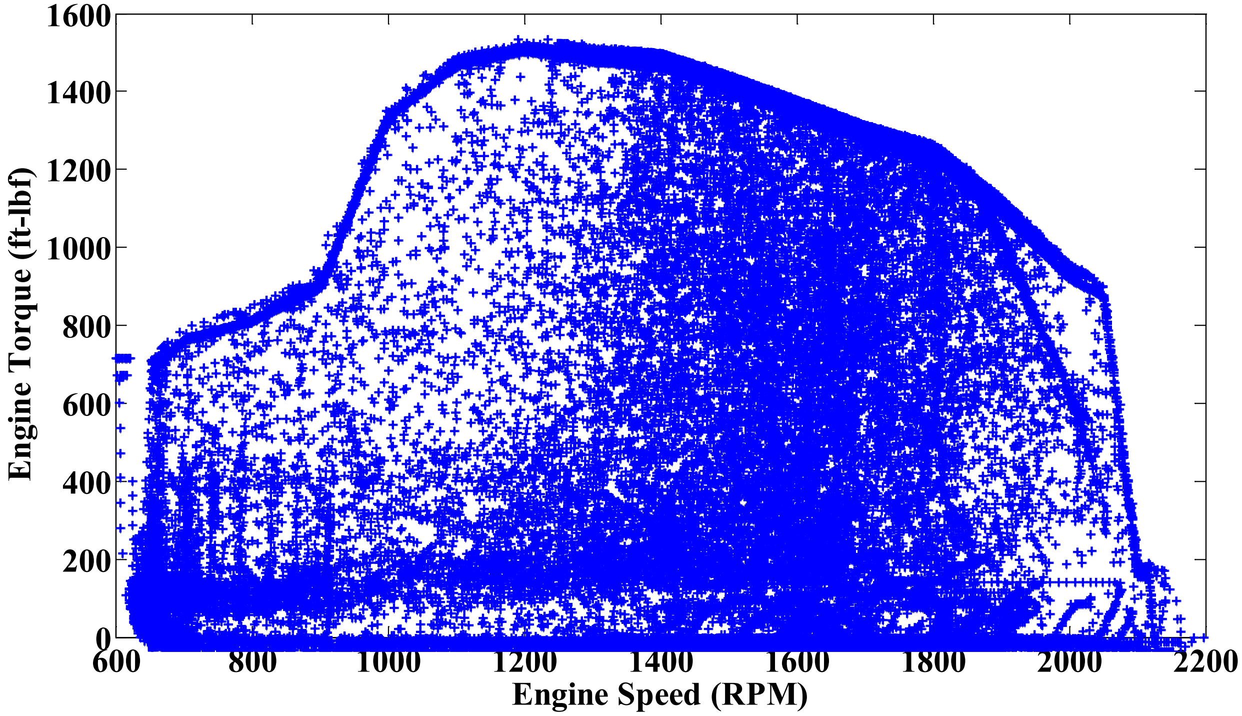


Figure 82: Training data setpoints.

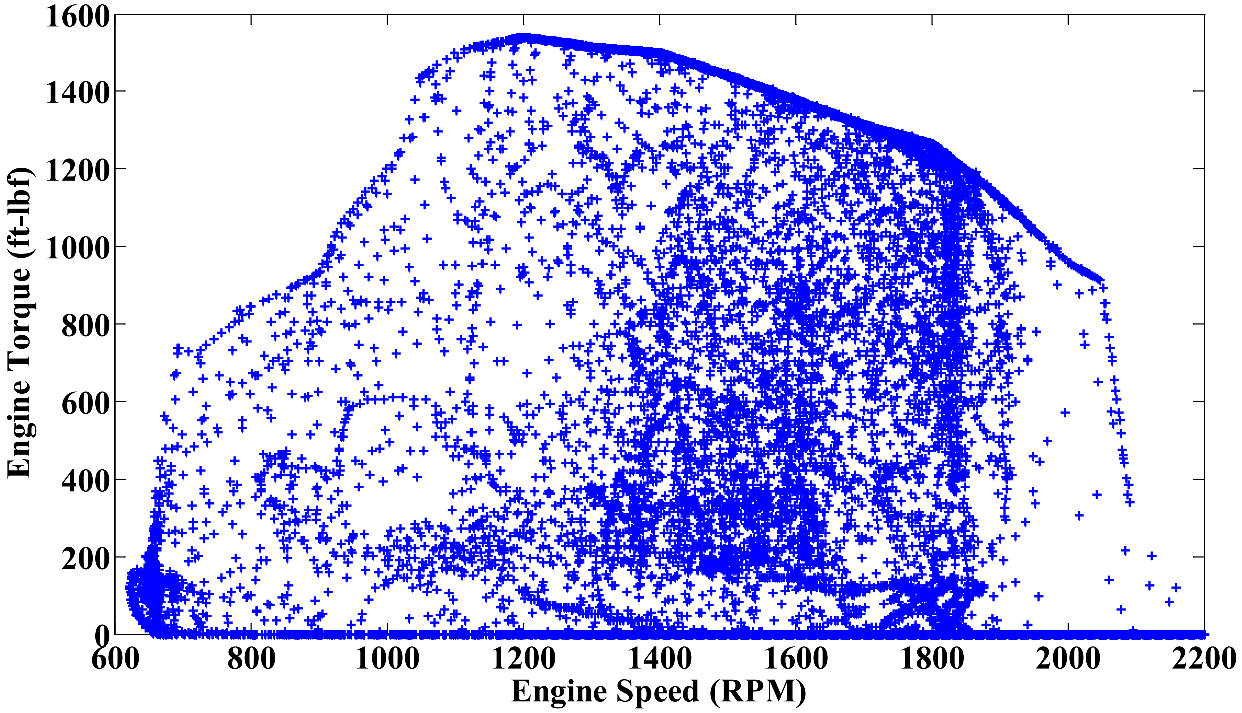


Figure 83: Validation data setpoints.

D.2 Data from the Sensors Selected for Failure in Training Dataset

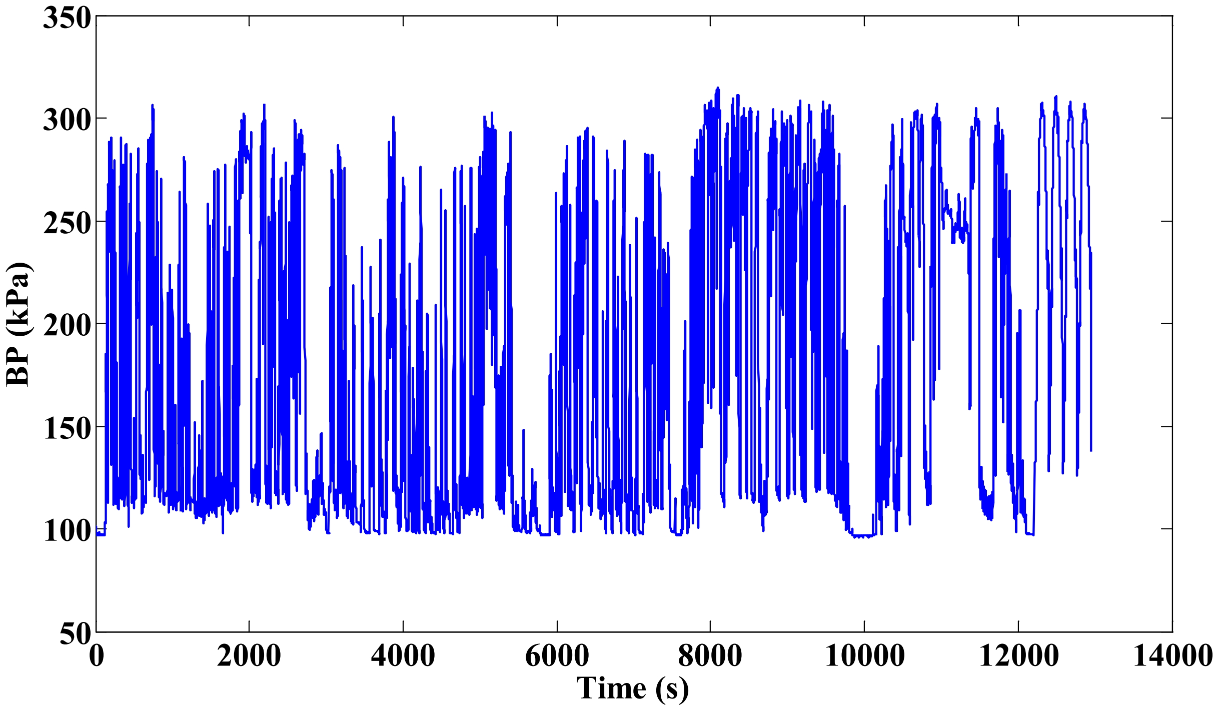


Figure 84: Boost pressure data used for NN training.

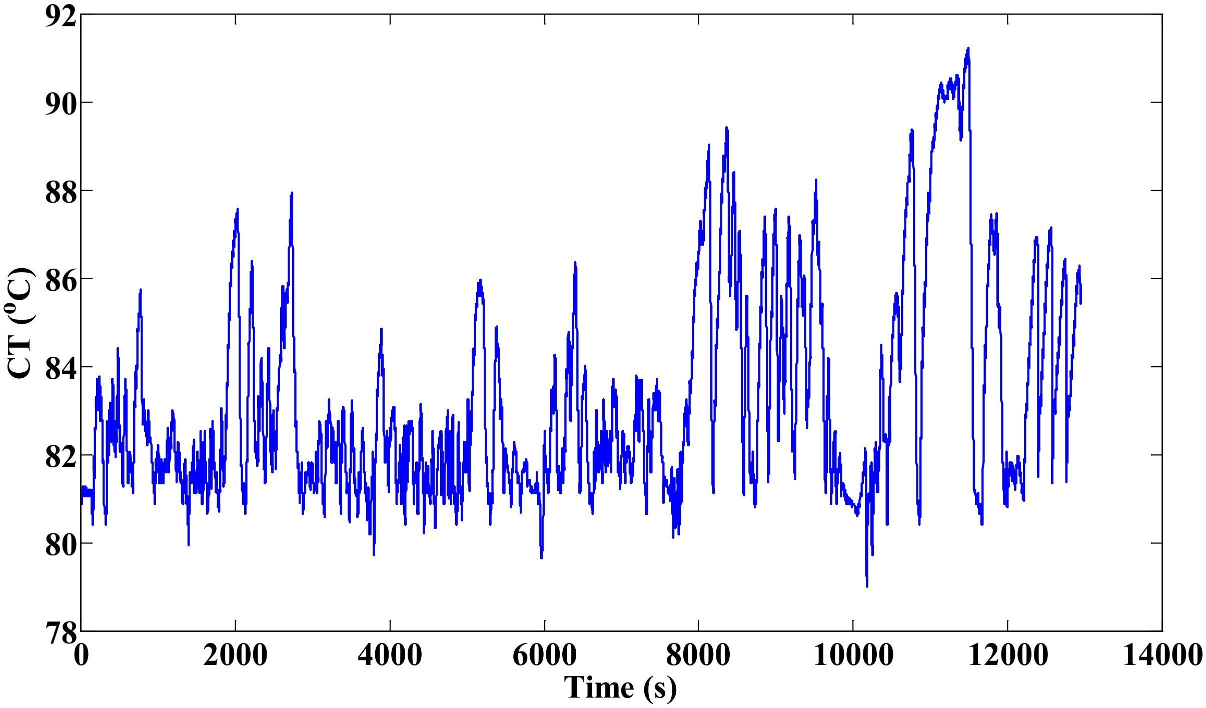


Figure 85: Coolant temperature data used for NN training.

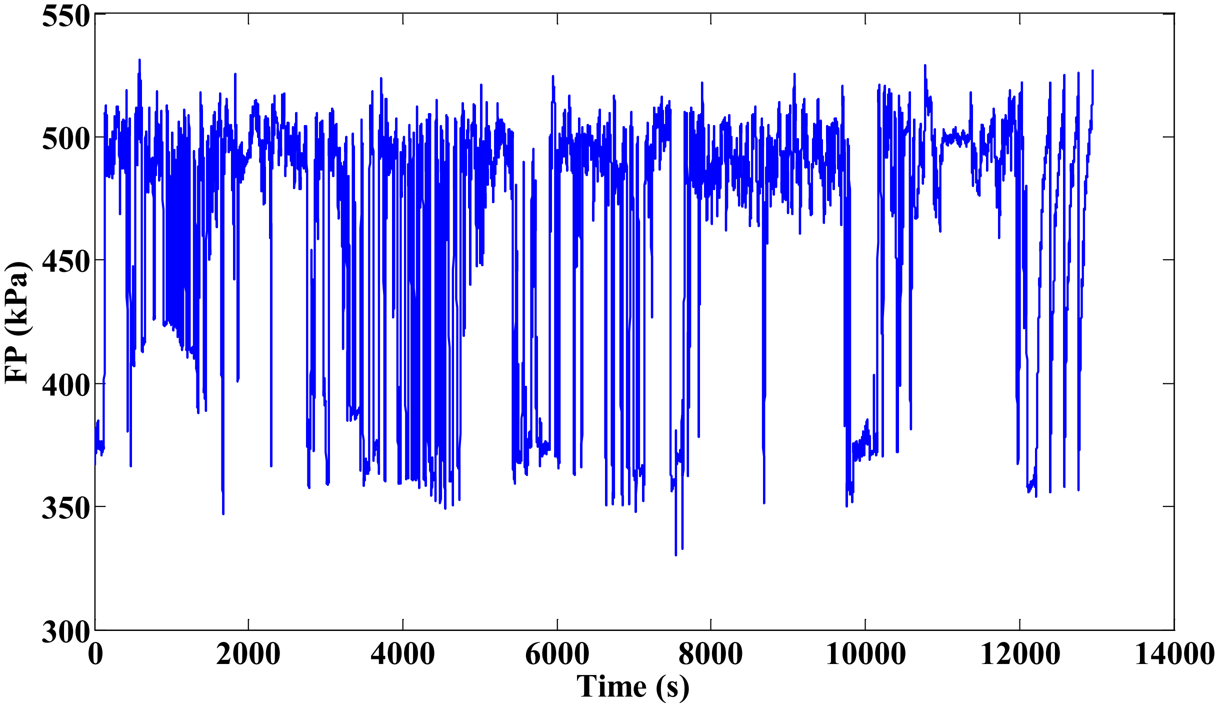


Figure 86: Fuel pressure data used for NN training.

D.3 Data from the Sensors Selected for Failure in Validation Dataset

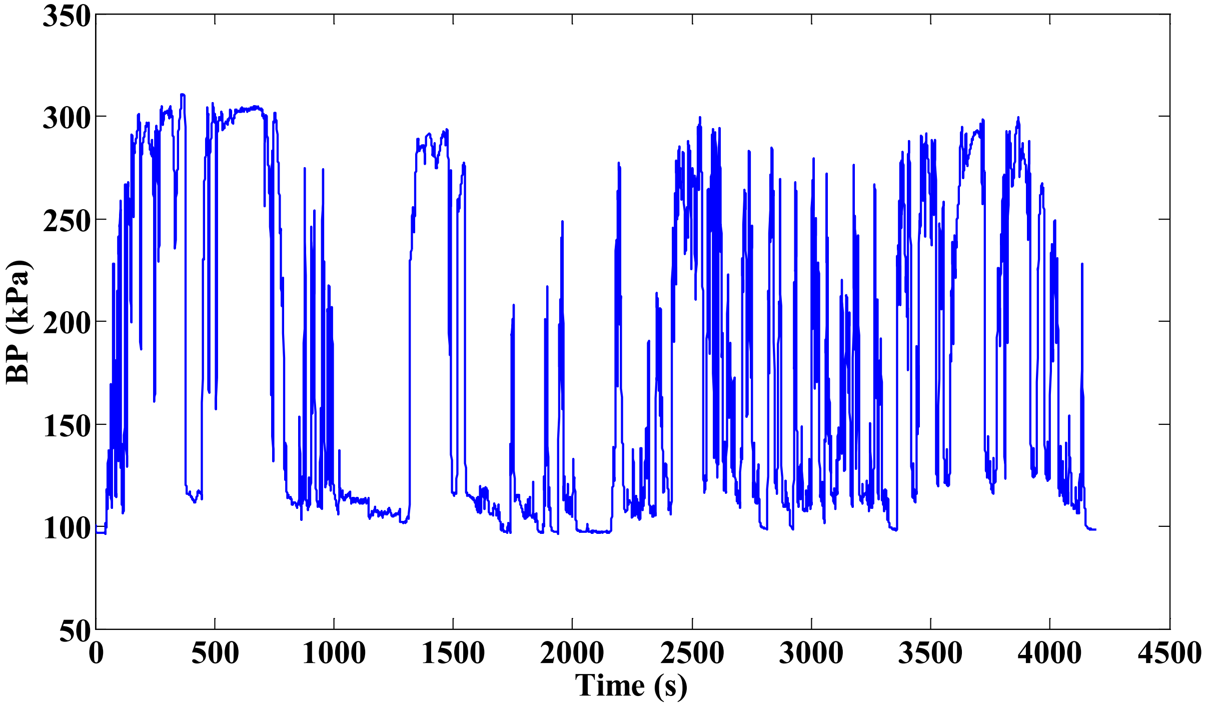


Figure 87: Boost pressure data for NN validation.

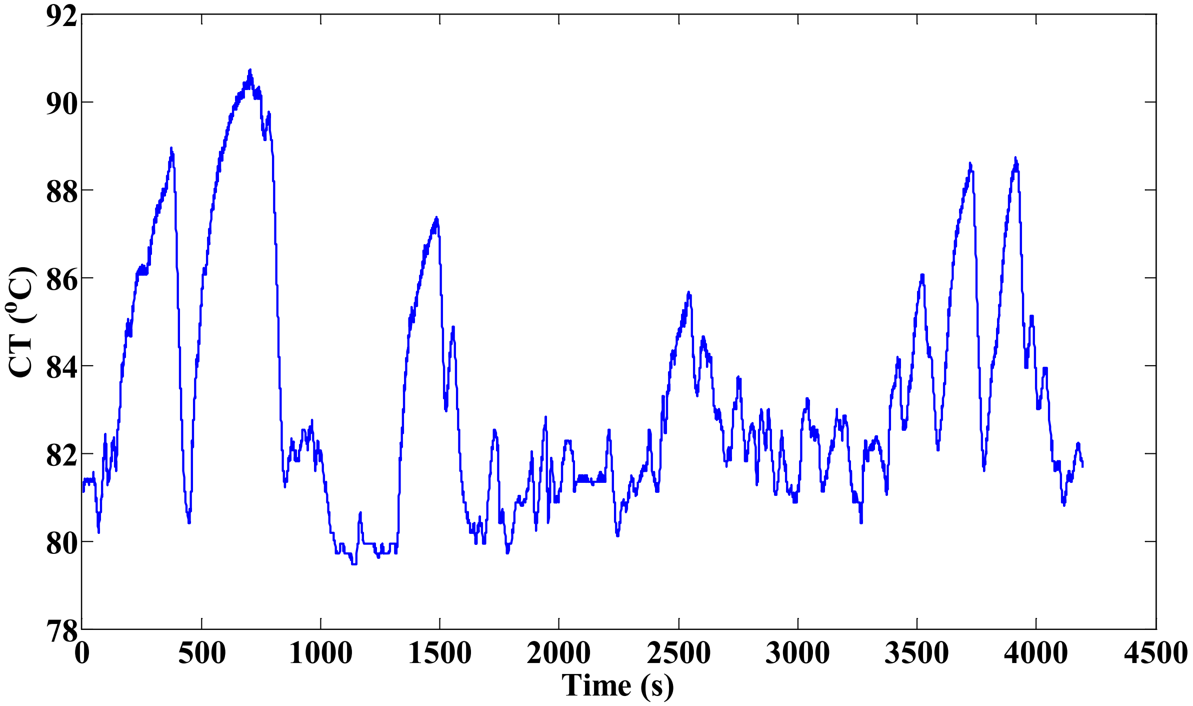


Figure 88: Coolant temperature data for NN validation.

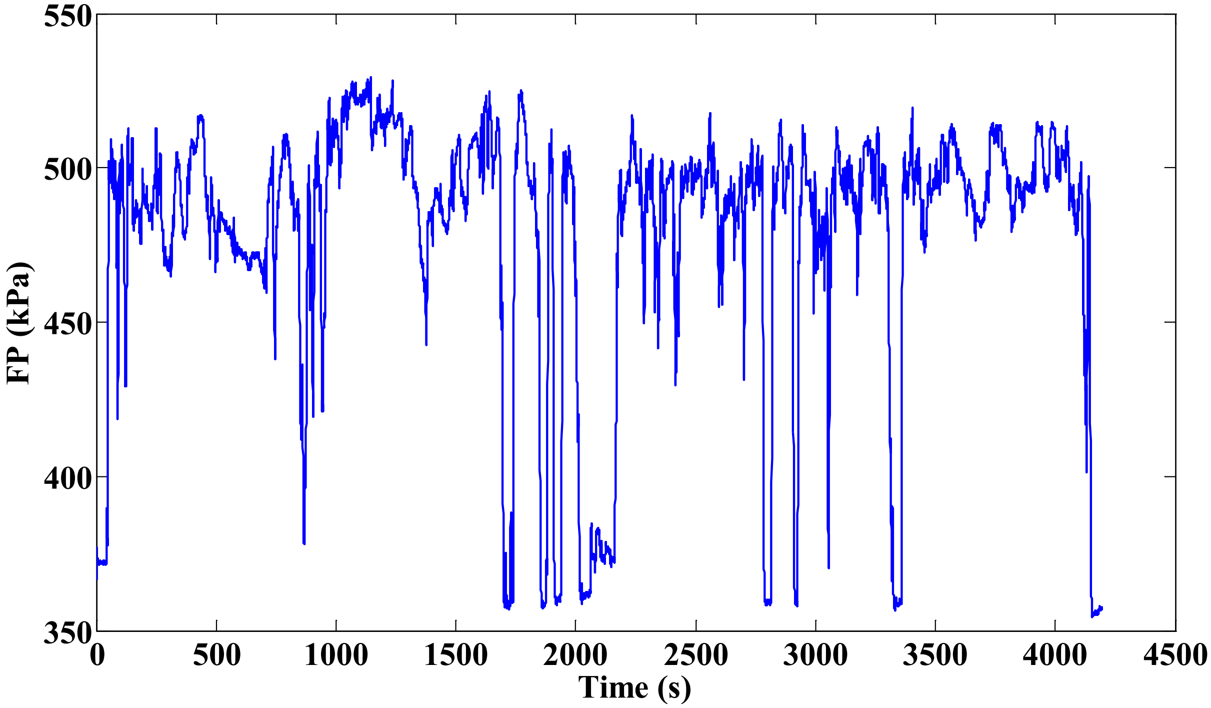


Figure 89: Fuel pressure data for NN validation.

APPENDIX E DAQ SYSTEM EQUIPMENT

E.1 3B01 Signal Conditioning Subsystem

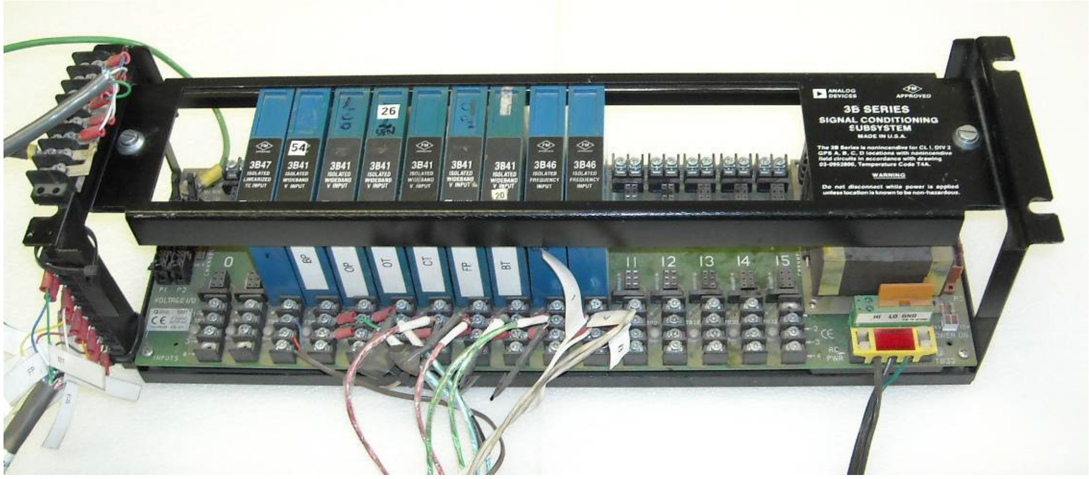

ANALOG DEVICES
FM APPROVED
3B SERIES
SIGNAL CONDITIONING SUBSYSTEM
MADE IN U.S.A.
The 3B Series is nonincendive for CL I, DIV 2 GPS A, B, C, D locations with nonincendive field circuits in accordance with drawing 03-0952800, Temperature Code T4A.
WARNING
Do not disconnect while power is applied unless location is known to be non-hazardous.
3B47 ISOLATED LINEARIZED TC INPUT
54
3B41 ISOLATED WIDEBAND V INPUT
3B41 ISOLATED WIDEBAND V INPUT
26
3B41 ISOLATED WIDEBAND V INPUT
3B41 ISOLATED WIDEBAND V INPUT
3B41 ISOLATED WIDEBAND V INPUT
3B41 ISOLATED WIDEBAND V INPUT
20
3B46 ISOLATED FREQUENCY INPUT
3B46 ISOLATED FREQUENCY INPUT
0
BP
OP
OT
CT
FP
BT
11
12
13
14
15

Figure 90: 3B01 Signal Conditioner.

Table 29: Specifications of the 3B01 Conditioner.

Channels	16
Backplane	19” universal mounting relay rack
AC Power Supply	115/240 V
Modules	Single Channel
	Magnetic Isolation up to 1500V Common-Mode Voltage
	±10 V DC or 4-20 mA Current Outputs
	Wide Zero Suppression
Size	17.4 in x 5.2 in x 4.37 in

E.2 NI Breakout Boards

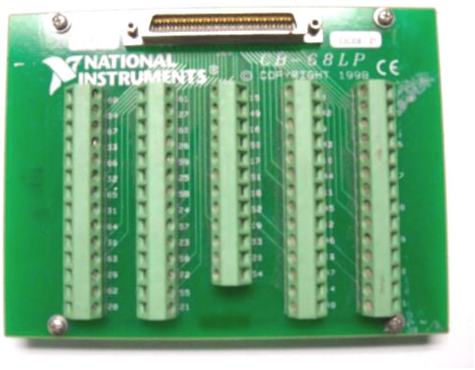


Figure 91: NI CB-68LP breakout board used with NI PCI-6024E.

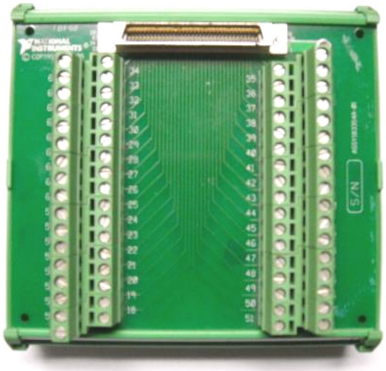


Figure 92: NI TBX-68 breakout board used with NI PCI-MIO-16E-1.

E.3 DAQ Cards

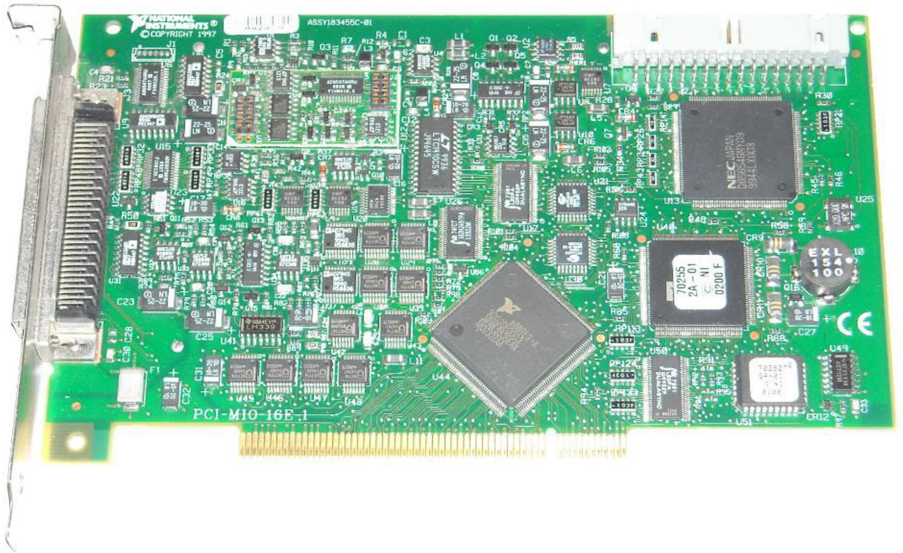


Figure 93: NI PCI-MIO-16E-1, one of two DAQ cards utilized.

Table 30: Specifications of the DAQ Cards.

	Cards	PCI-6024E	PCI-MIO-16E-1
Analog Channels	Input	16	
	Output	2	
Resolution		12 bits	
Maximum Sampling Rate		200 kHz	1.25 MHz

E.4 7B Output Modules

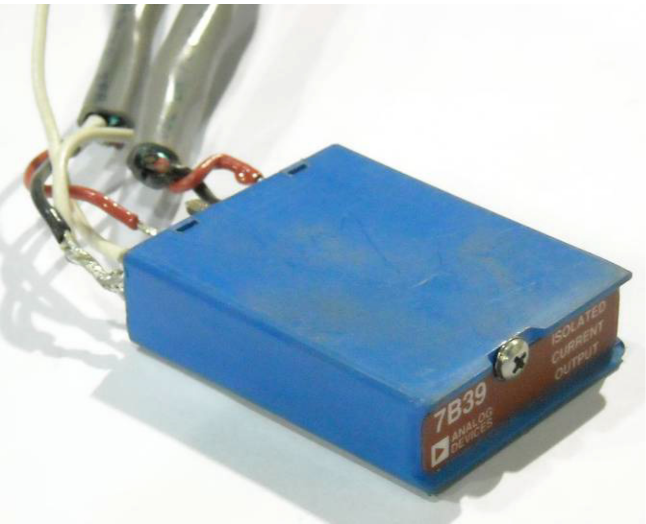


Figure 94: 7B39 Current Output Module used for the *BP* and *FP* sensors.

Table 31: Specifications of the 7B39.

DC Power Supply	24V
Signal Isolation	Up to 1500V Common-Mode Voltage
Input	0-10 V DC
Output	0-20 mA
Size	17.4 in x 5.2 in x 4.37 in

Table 32: 7B22 Voltage Output module specifications used for the *CT* sensor.

DC Power Supply	24V
Signal Isolation	Up to 1500V Common-Mode Voltage
Input	±10 V DC
Output	±10 V DC
Size	17.4 in x 5.2 in x 4.37 in

The *BP* and *FP* sensor channels on the EMS require voltage signals. A resistor placed across the 7B39 output pins converted the current output into the appropriate voltages. Knowing the maximum voltage, V_{max} for the channels are 5V and module I_{max} is 20mA, the resistor value was:

$$V_{max} = I_{max} R_{7B39}$$

$$R_{7B39} = \frac{V_{max}}{I_{max}} \Omega = \frac{5V}{20mA} = 250$$

APPENDIX F SFDIA CONFIGURATION FOR DATA ACQUISITION

F.1 SFDIA with Data Acquisition Configuration

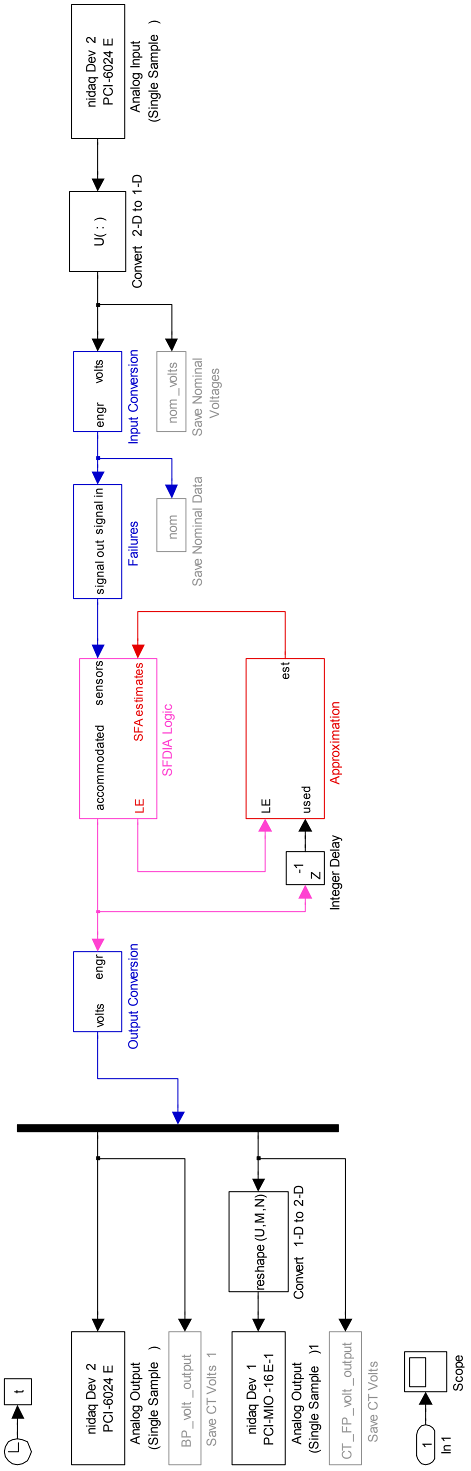


Figure 95: SFDIA subsystem configured for data acquisition.

F.2 Sensor Input Calibration Results

a) Inlet-air temperature

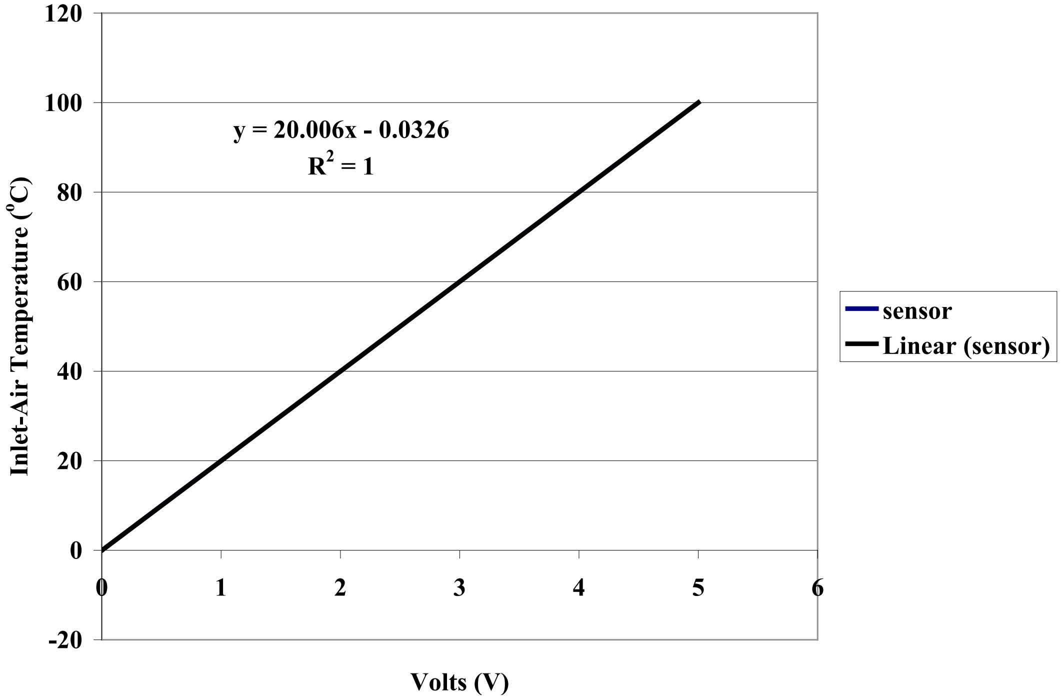


Figure 96: AT calibration plot.

b) Boost pressure

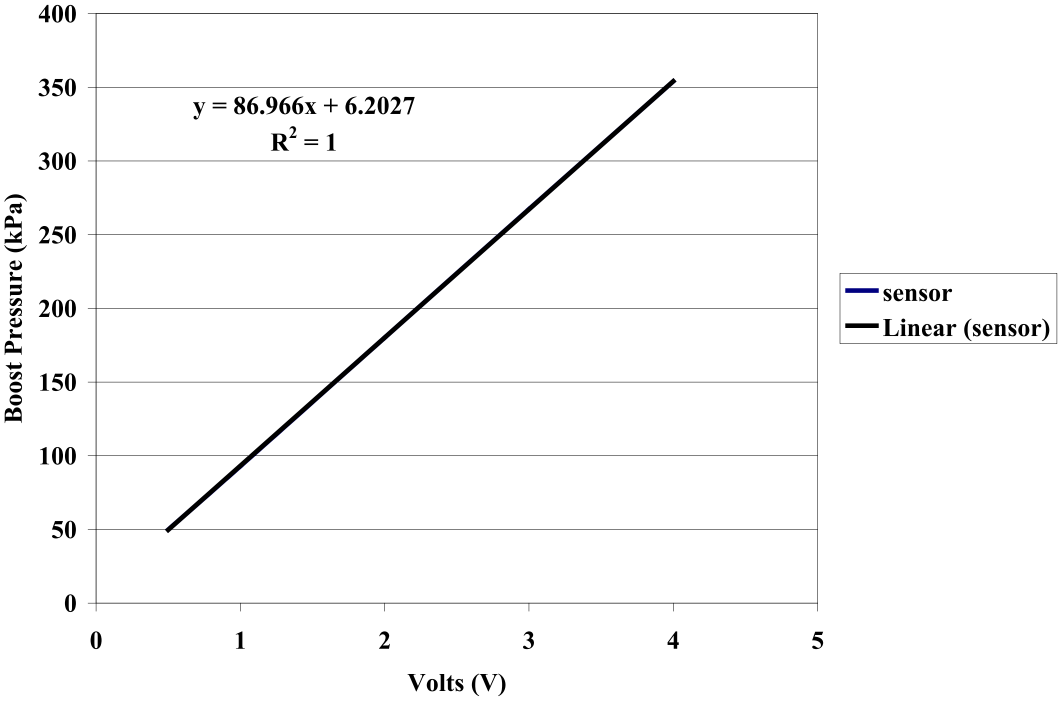


Figure 97: BP calibration plot.

c) Oil pressure:

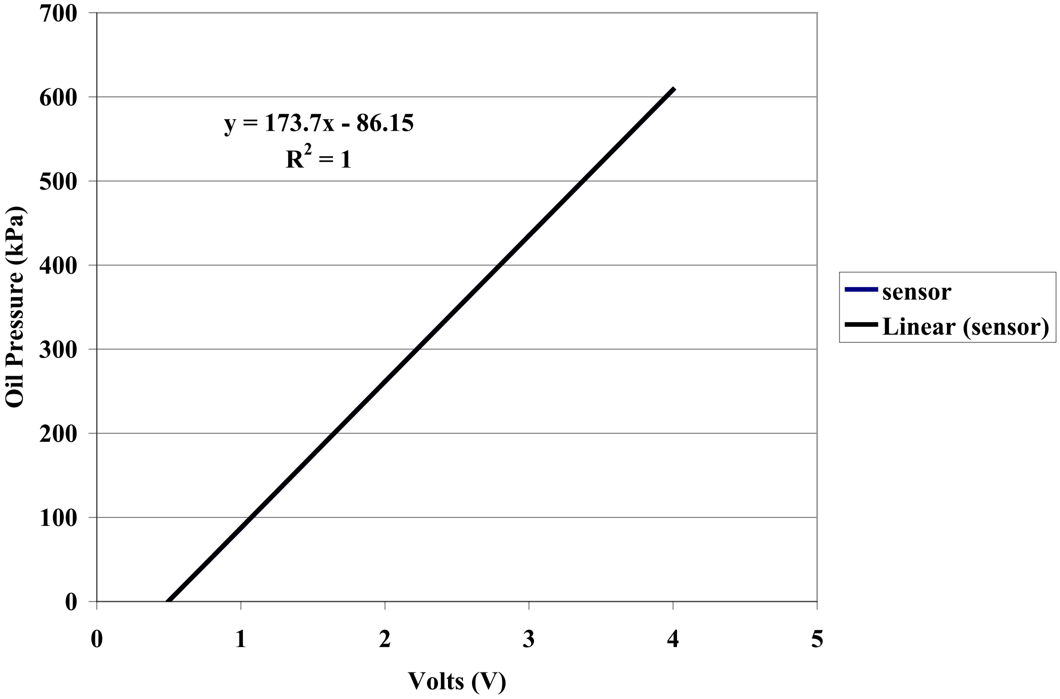


Figure 98: OP calibration plot.

d) Oil temperature:

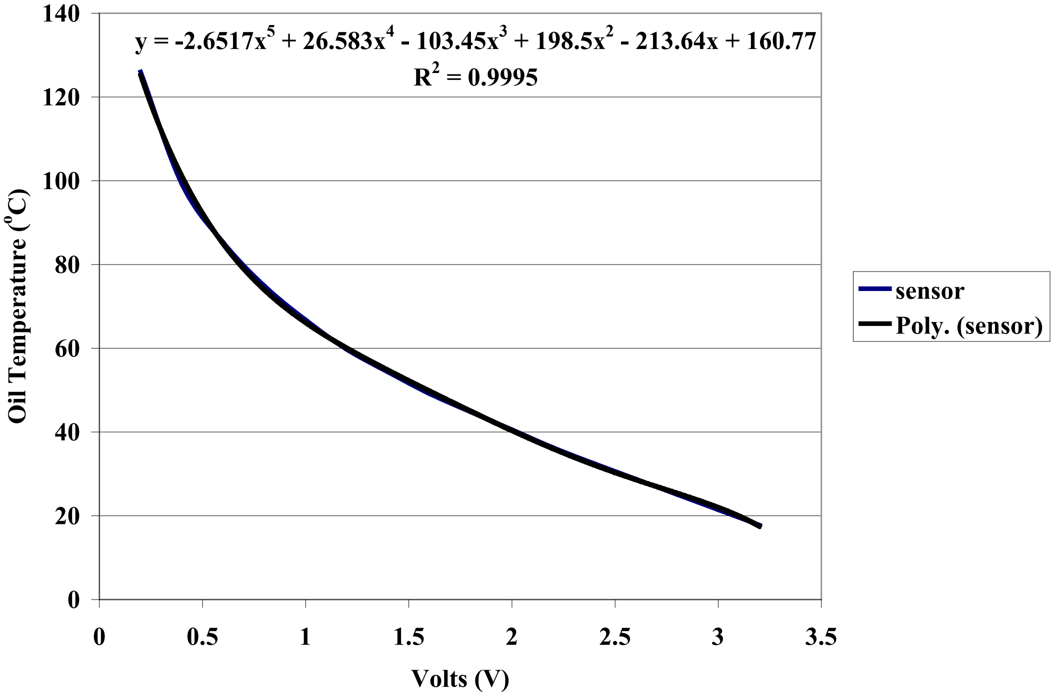


Figure 99: OT calibration plot.

e) Coolant temperature:

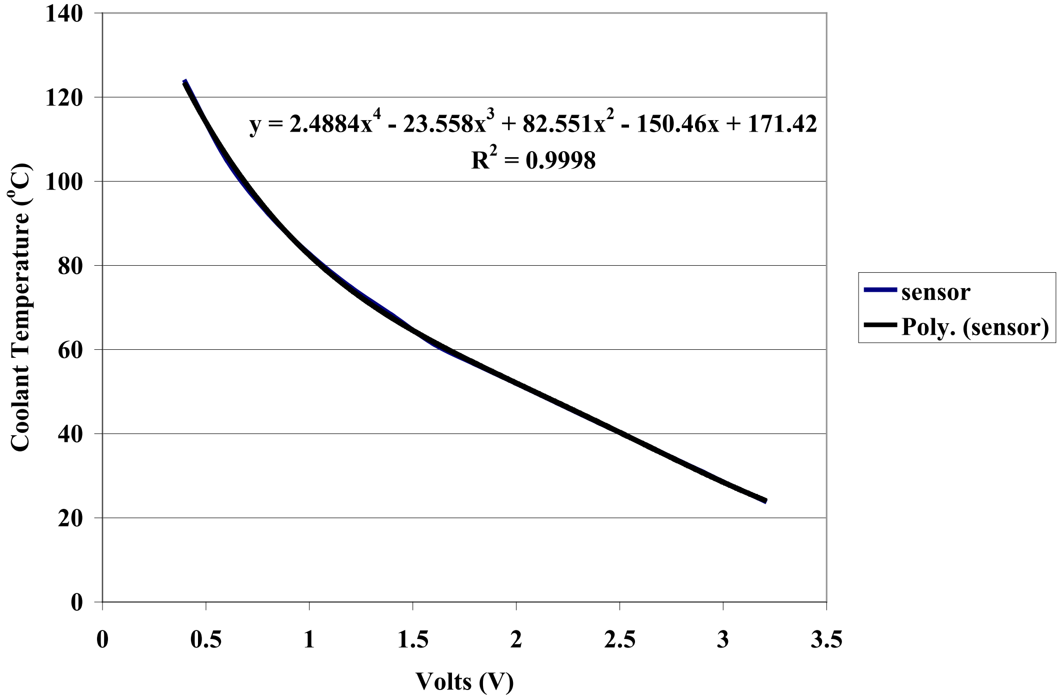


Figure 100: CT calibration plot.

f) Fuel pressure:

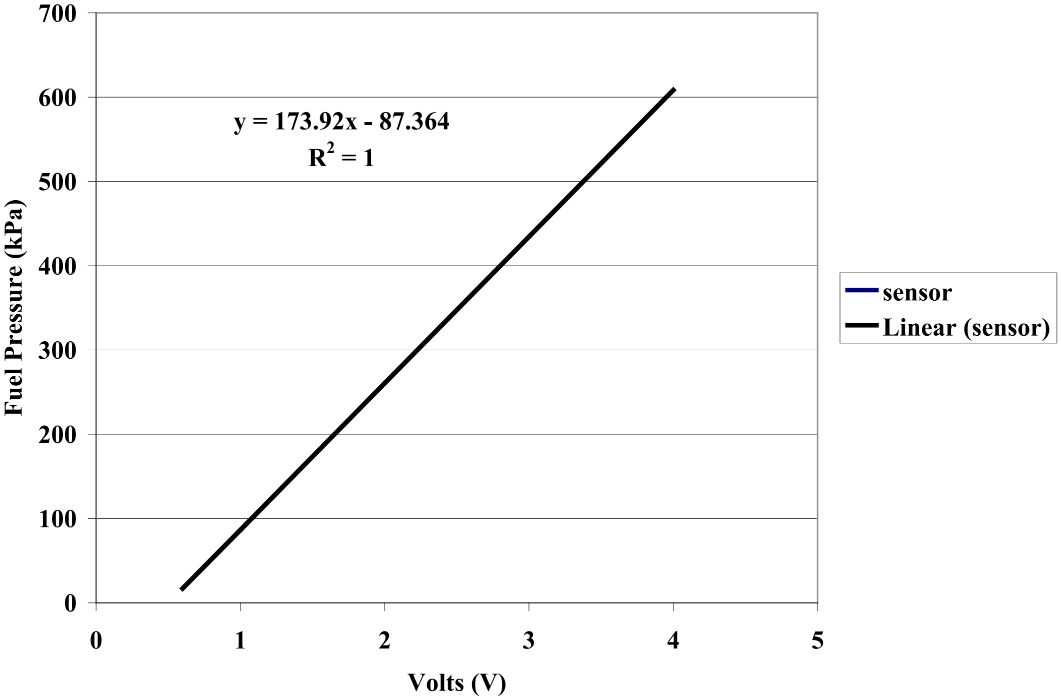


Figure 101: FP calibration plot.

g) Boost temperature:

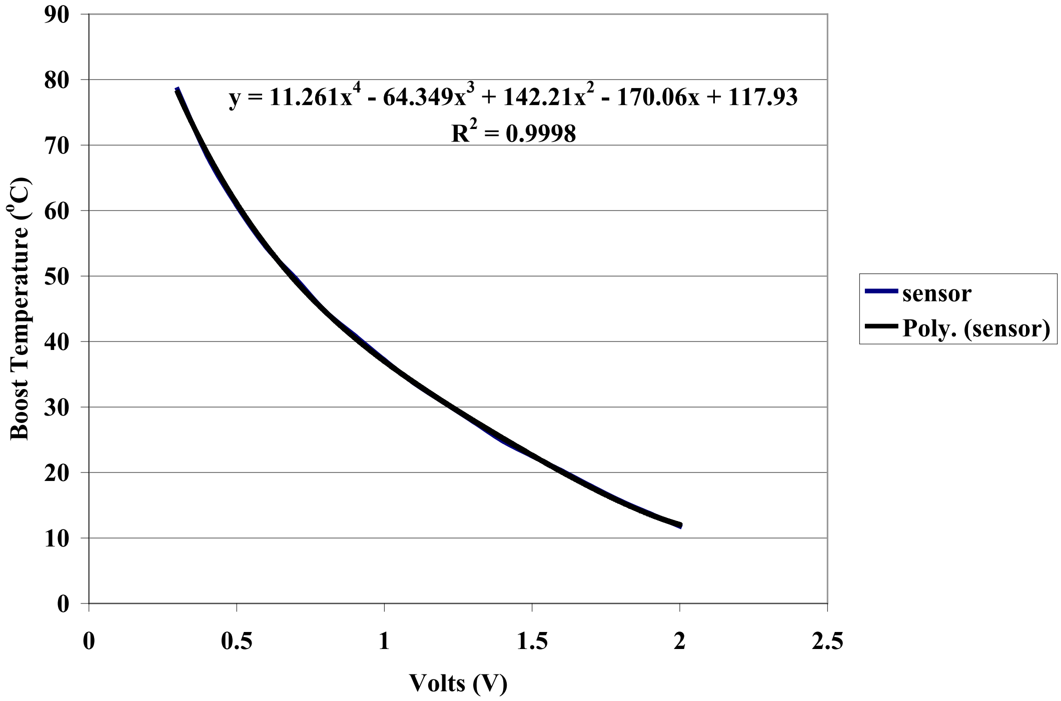


Figure 102: BT calibration plot.

h) Engine speed:

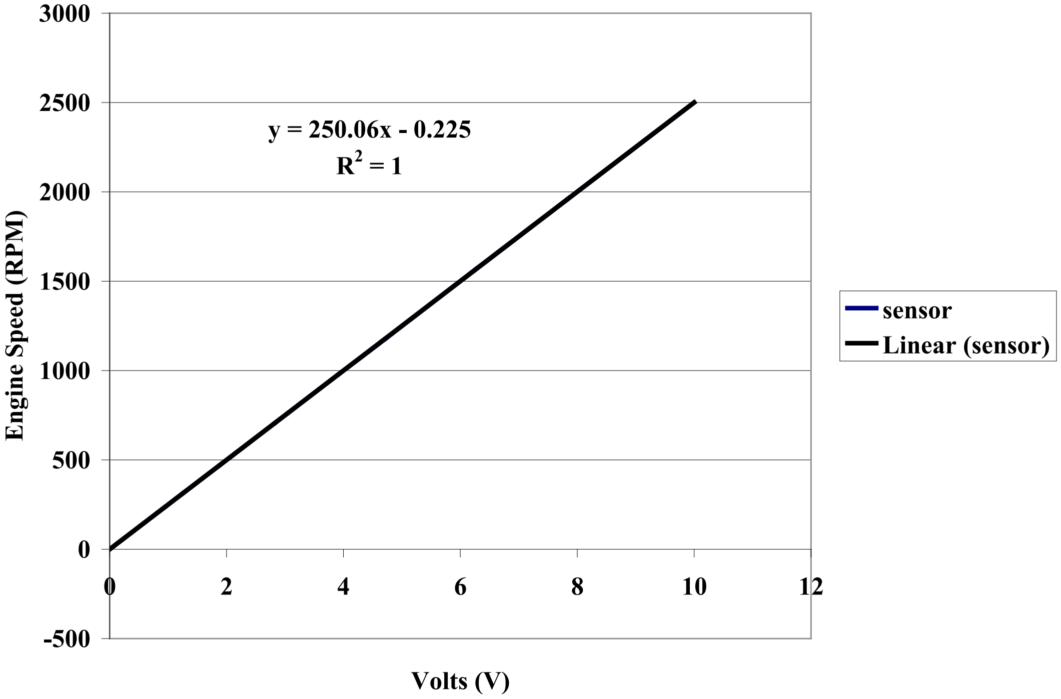


Figure 103: ES calibration plot.

i) VGT speed:

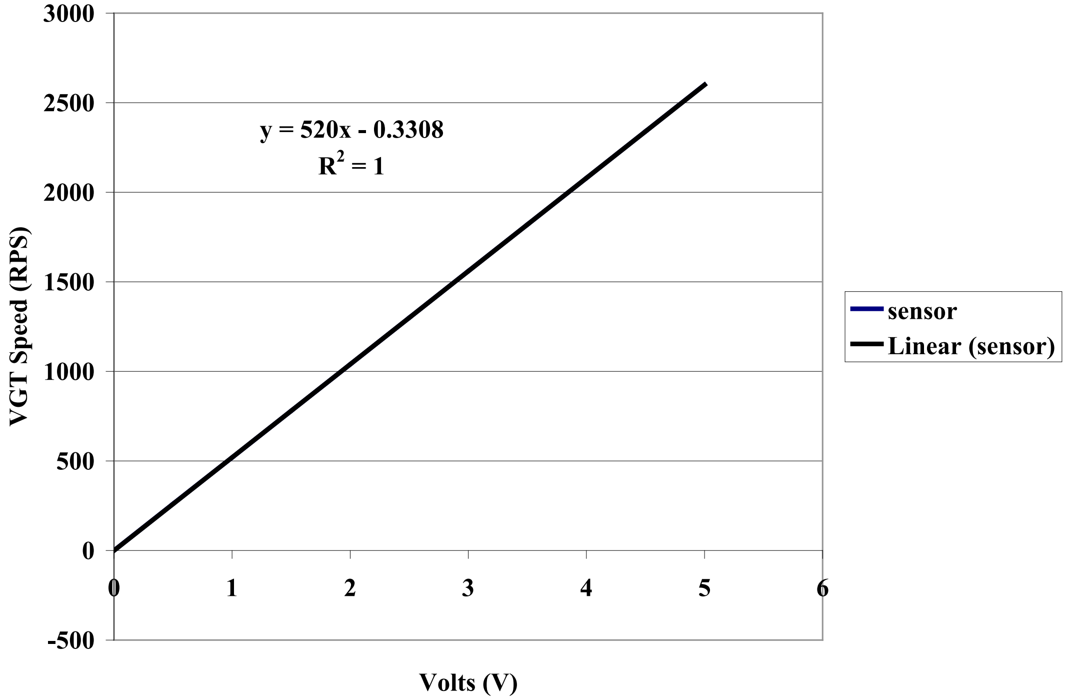


Figure 104: VGTS calibration plot.

F.3 Sensor Output Profile

a) Boost pressure:

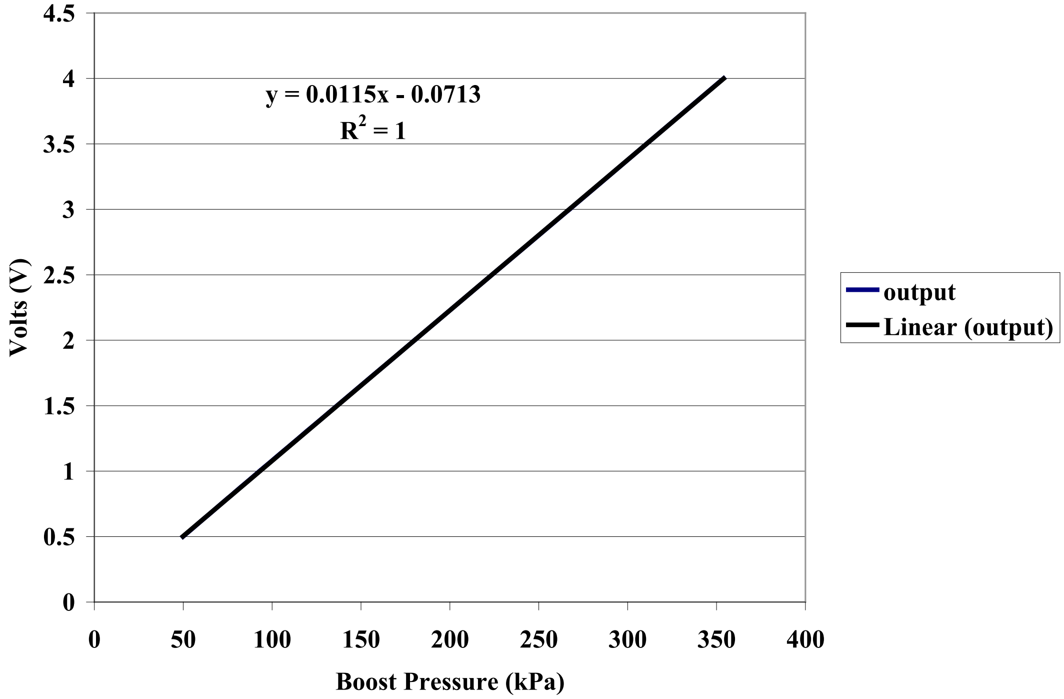


Figure 105: BP output calibration.

b) Coolant temperature:

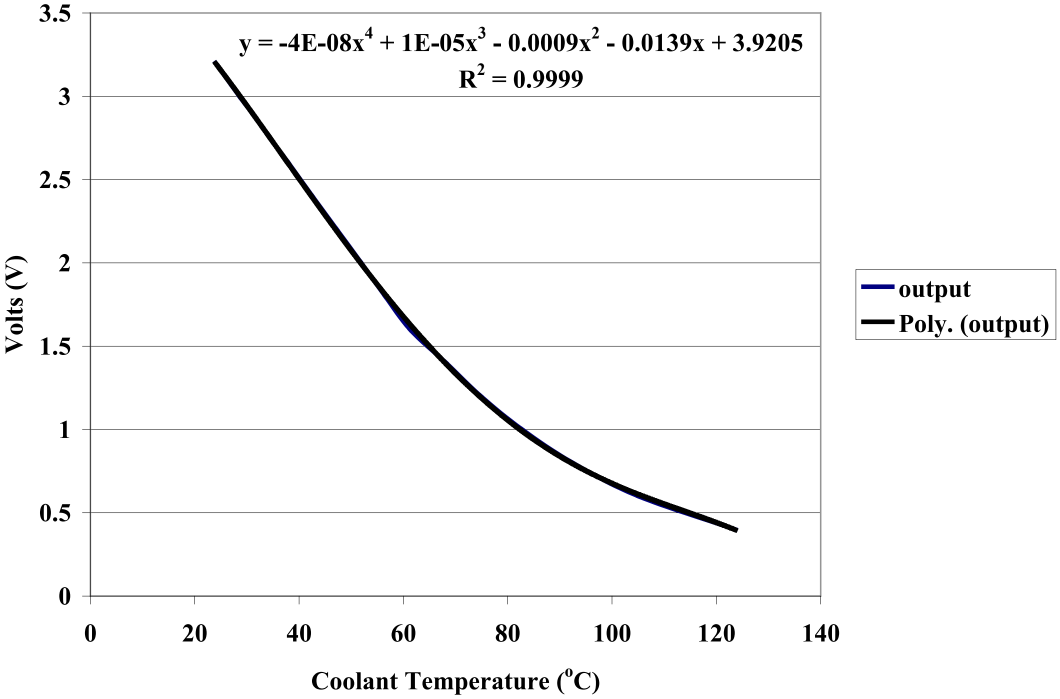


Figure 106: CT output calibration.

c) Fuel pressure:

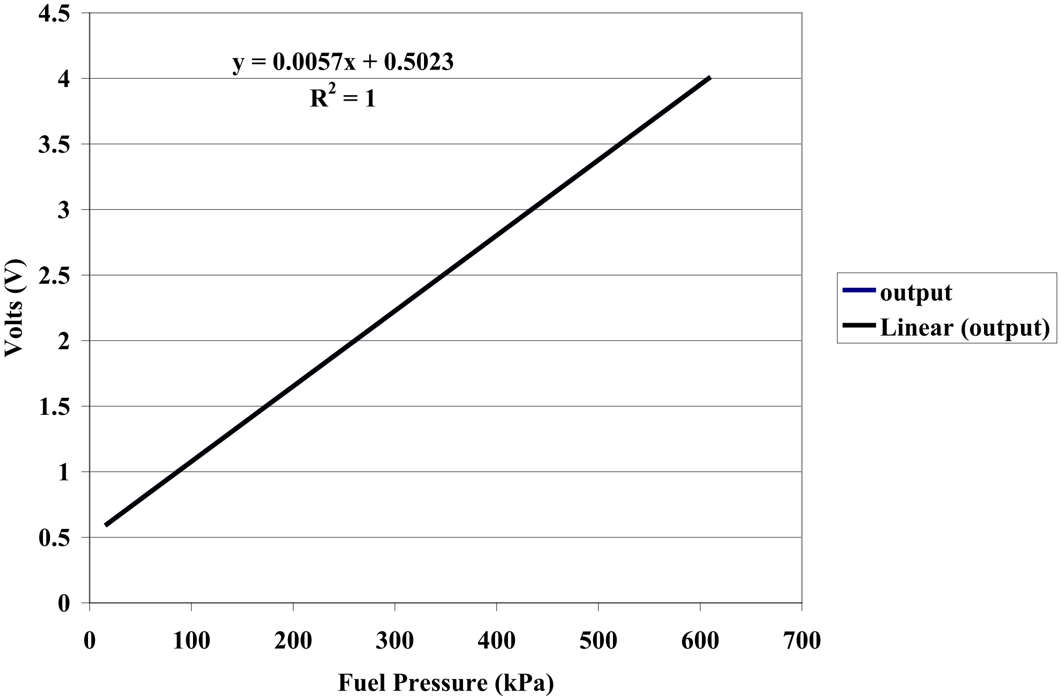


Figure 107: FP output calibration.

F.4 Input Conversion Block

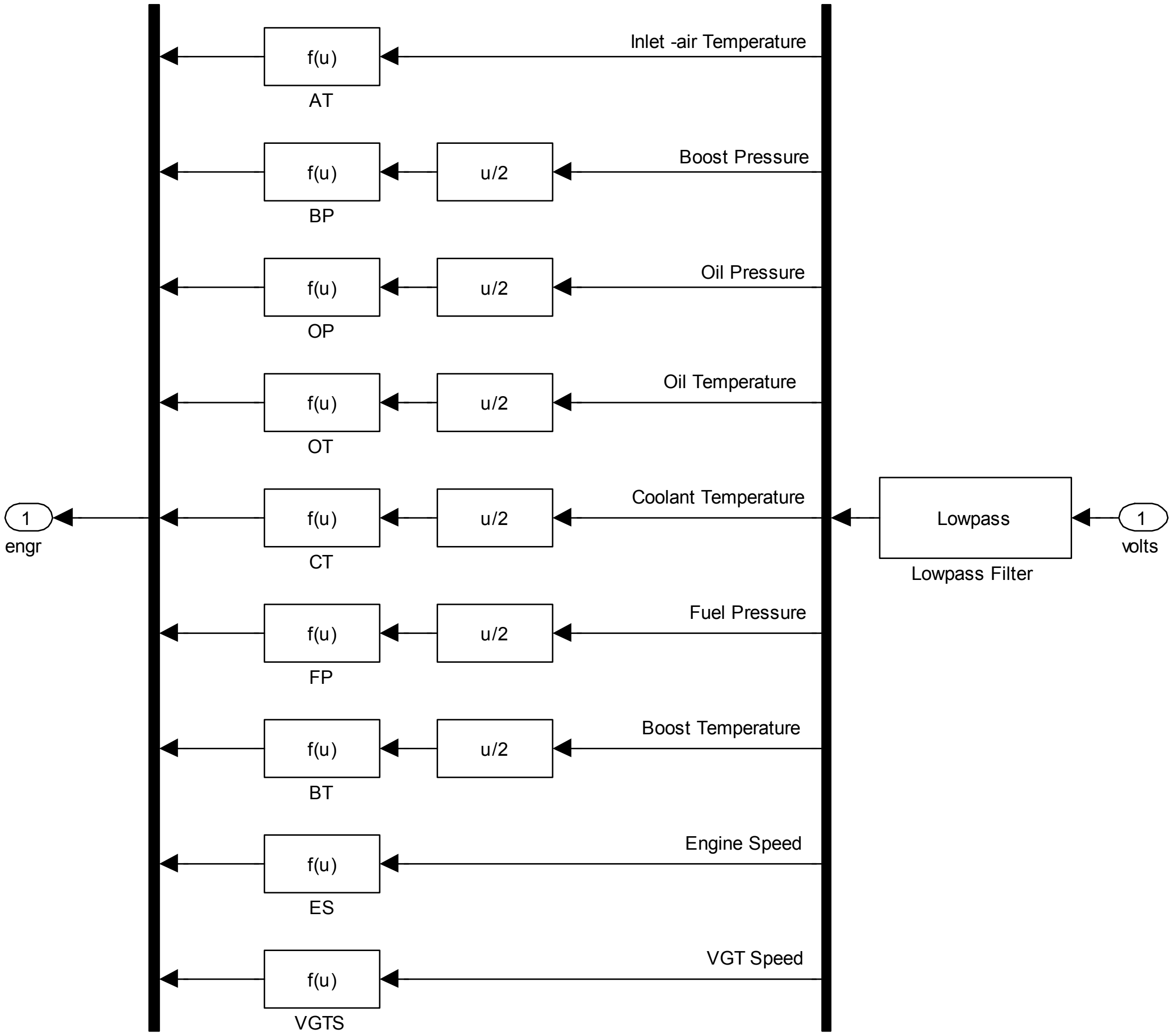


Figure 108: Input Conversion model.

F.5 Output Conversion Block

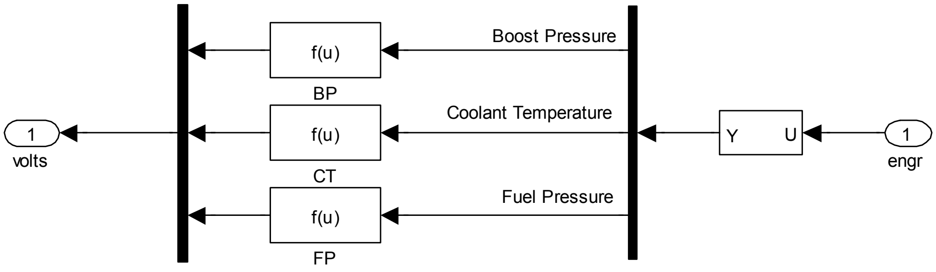


Figure 109: Output Conversion model.

APPENDIX G SFDIA SETTINGS AND TEST RESULTS SUMMARY

G.1 EMRAN RBF NN Parameters

Table 33: Nonlinear NN parameters for the Approximation block.

Parameters	*BP* model	*CT* model	*FP* model
Neurons, *m*	145	38	187
E_1	0.3	0.25	0.2
E_2	0.3	0.9	0.35
λ	0.3	0.5	0.3

Table 34: Nonlinear NN parameters for the Residuals Generation block.

Parameters	*BP* model	*CT* model	*FP* model
Neurons, *m*	208	103	190
E_1	0.2	0.25	0.2
E_2	0.25	0.5	0.35
λ	0.3	0.5	0.3

G.2 SFDIA Tuning Results

Table 35: Summary of SFDIA tuning to obtain *H* and the thresholds.

	Validation Data Range		*H*	Maximum Fast Residual Values	Factor		Thresholds	
	max	min			fast	Slow	fast	slow
BP	310.68 kPa	96.51 kPa	107.09 kPa	31.68 kPa	1.1	1.2	34.85 kPa	38.02 kPa
CT	90.73ºC	79.47ºC	5.63ºC	3.12ºC		1.15	3.43ºC	3.59ºC
FP	529.52 kPa	354.46 kPa	87.53 kPa	48.52 kPa		1.15	53.37 kPa	55.8 kPa

G.3 Offline SFDIA Residuals

a) FM1 residuals:

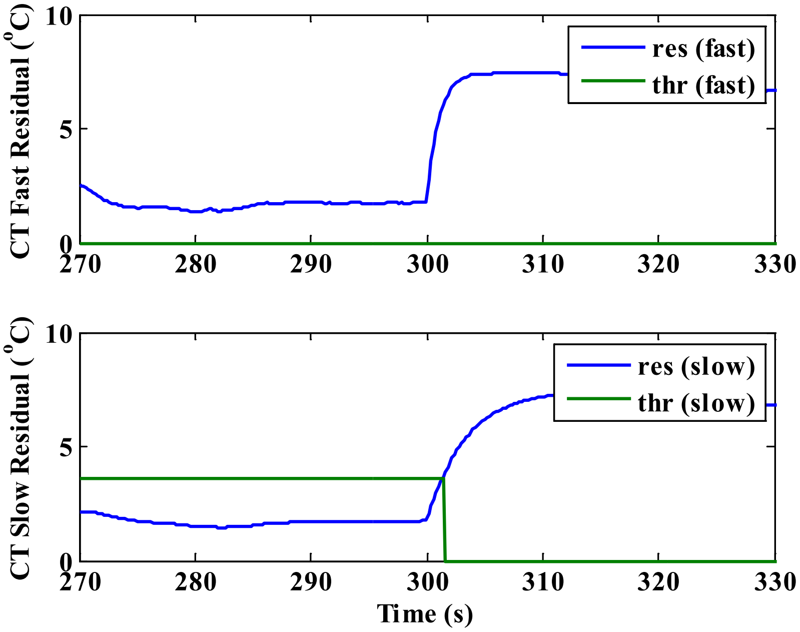


Figure 110: *CT* residuals using FM1.

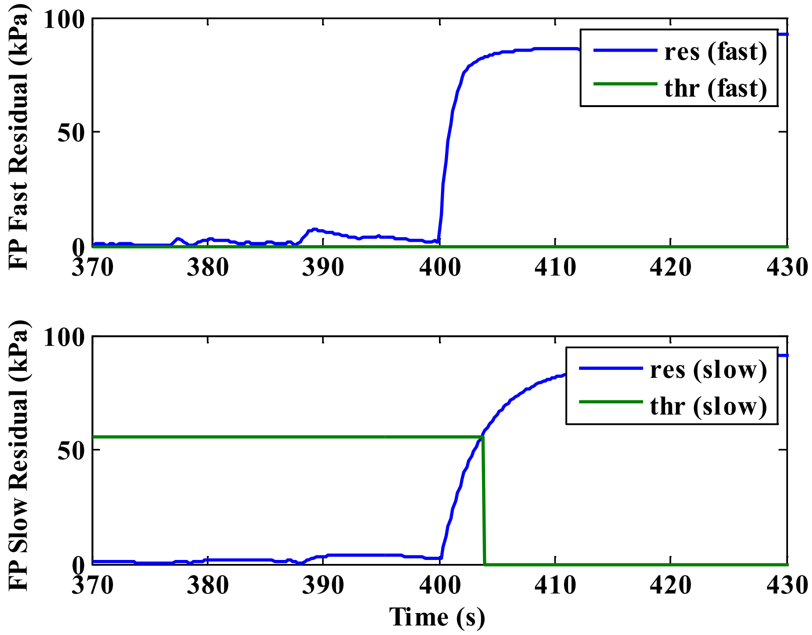


Figure 111: *FP* residuals using FM1.

b) FM2 residuals:

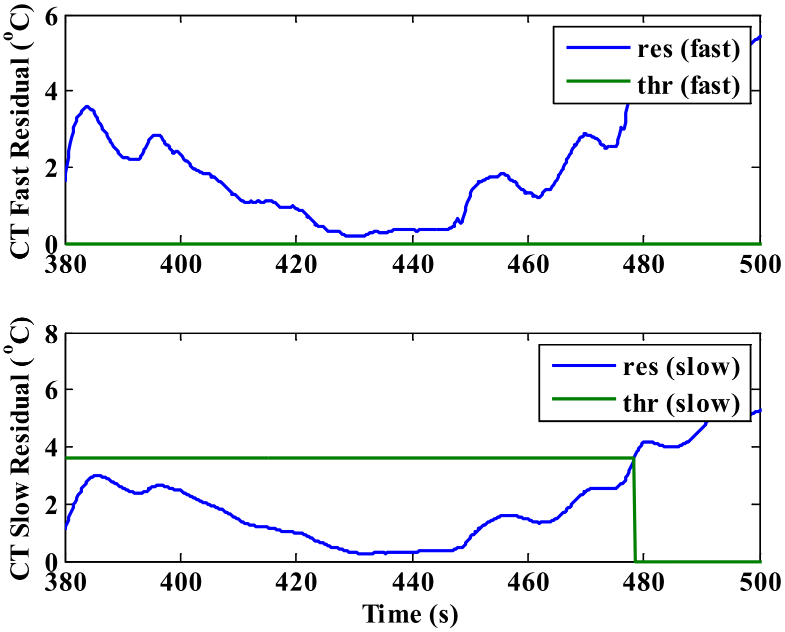


Figure 112: ***CT*** **residuals using FM2.**

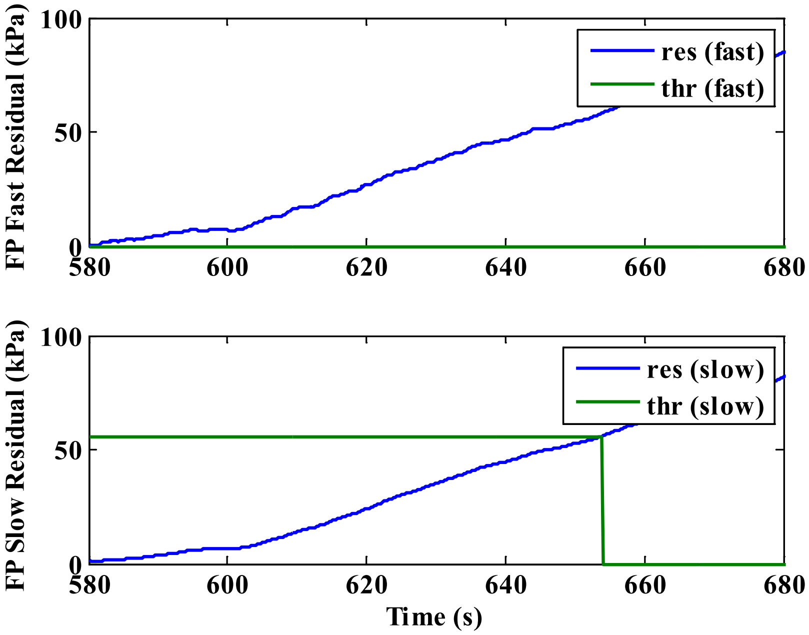


Figure 113: ***FP*** **residuals using FM2.**

c) FM3 residuals:

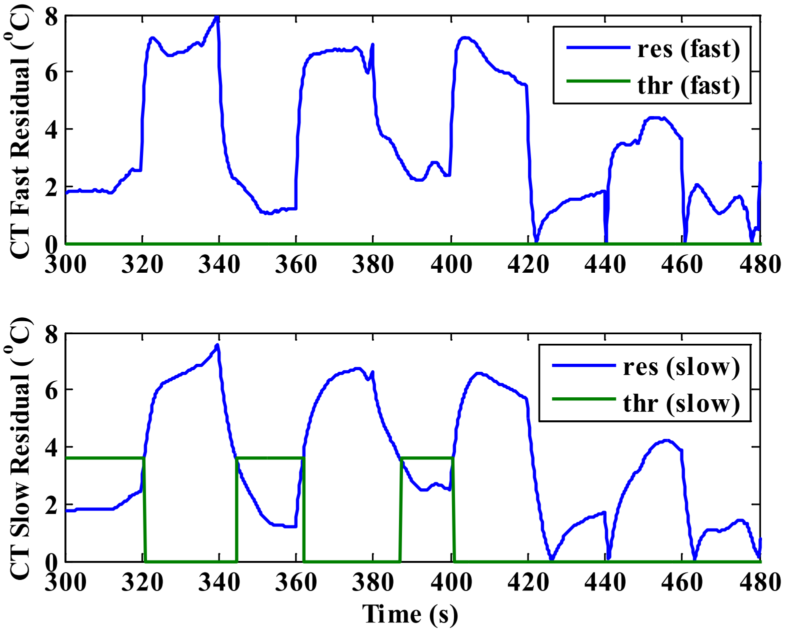


Figure 114: *CT* residuals using FM3.

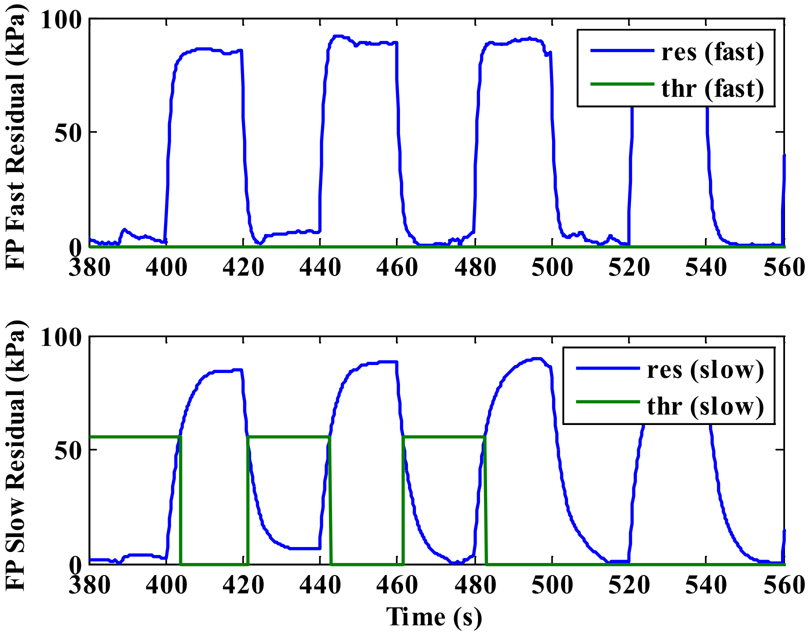


Figure 115: *FP* residuals using FM3.

G.4 FTP Cycle for Breakout Panel Verification

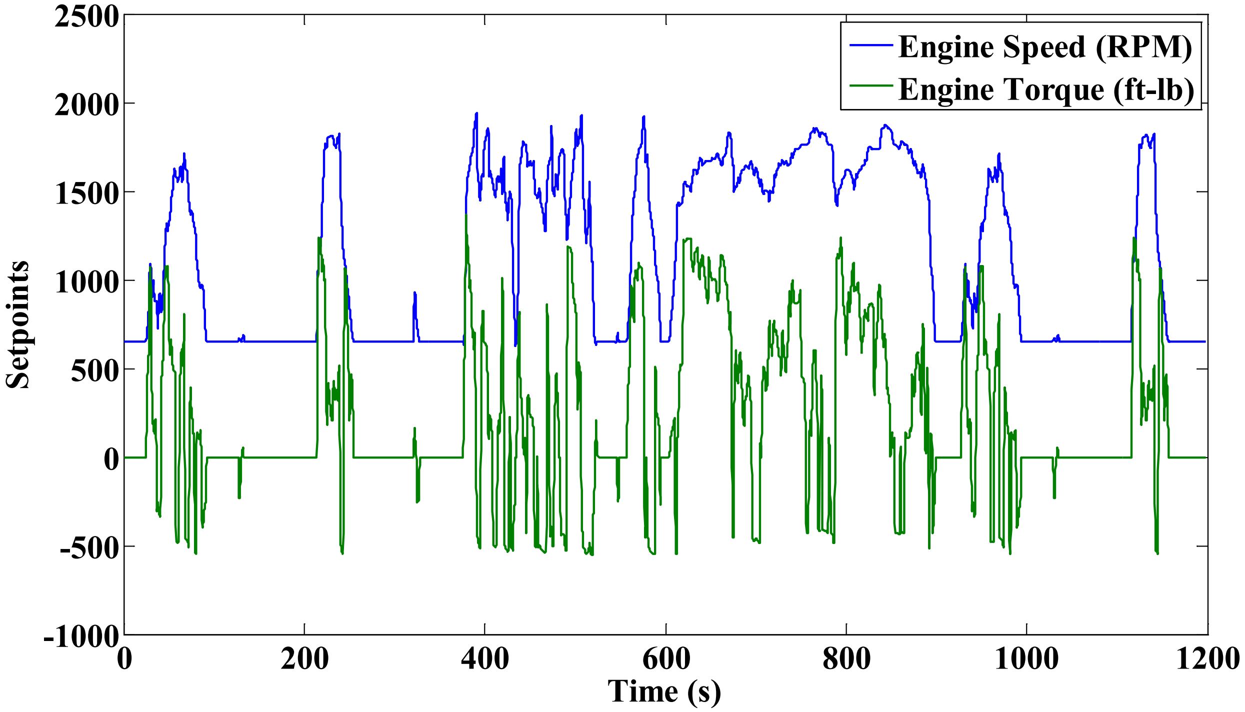


Figure 116: FTP duty cycle from breakout panel verification.

G.5 Data for Sensor Failure without Accommodation Study

a) Engine map data

Table 36: Engine map data (failure without accommodation study).

Curb Idle Speed	652 RPM
High Idle Speed	2099 RPM
Rated Speed	1806 RPM
Power at Rated Speed	358 hp
Peak Torque	1370 ft-lb

b) FTP setpoints:

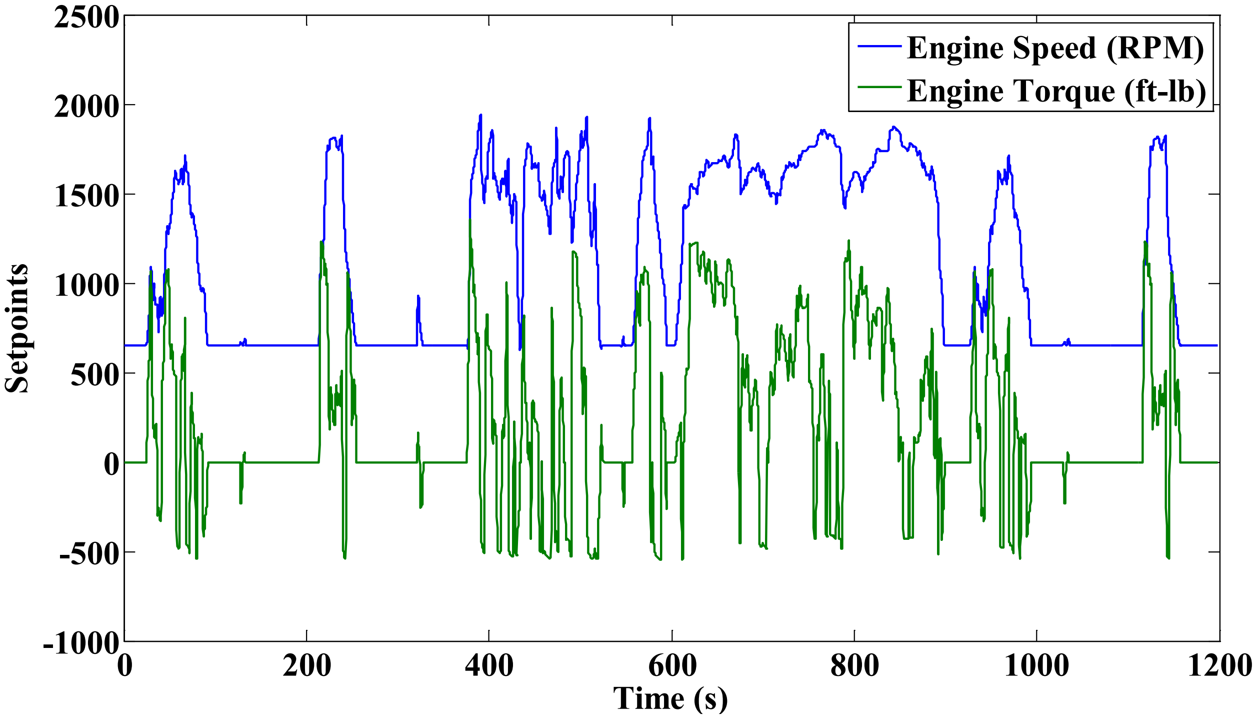


Figure 117: FTP duty cycle after breakout panel verification.

c) Residuals

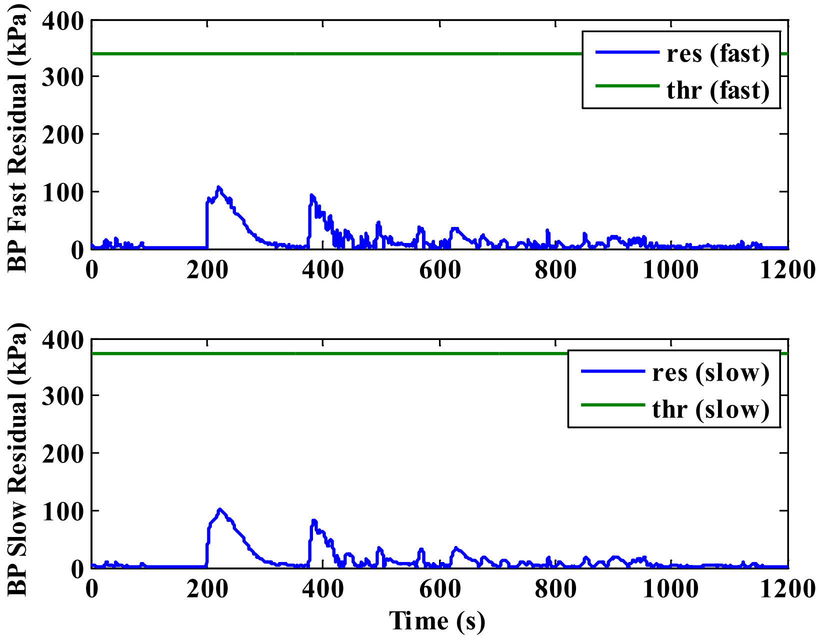


Figure 118: *BP* residuals with modified thresholds preventing accommodation (failure at 200s).

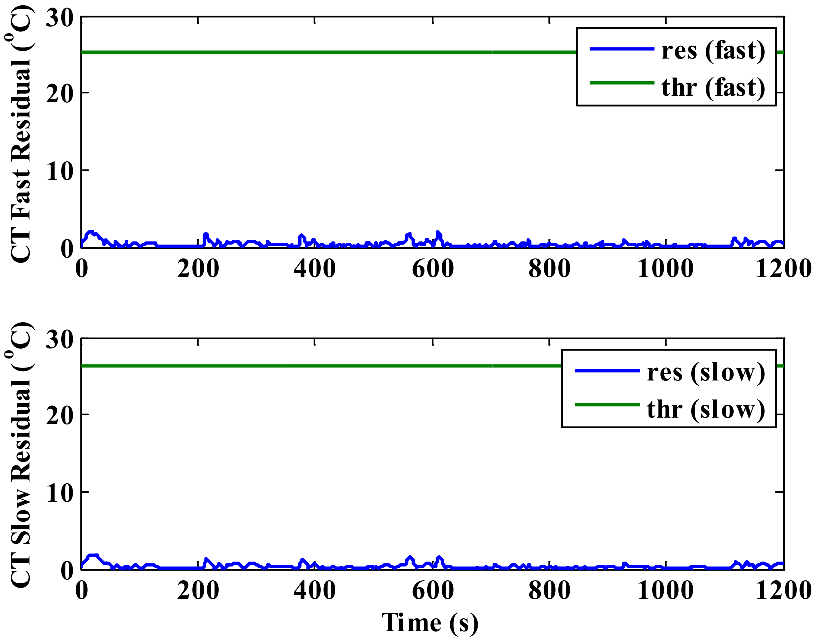


Figure 119: *CT* residuals with modified thresholds (*BP* failure at 200s).

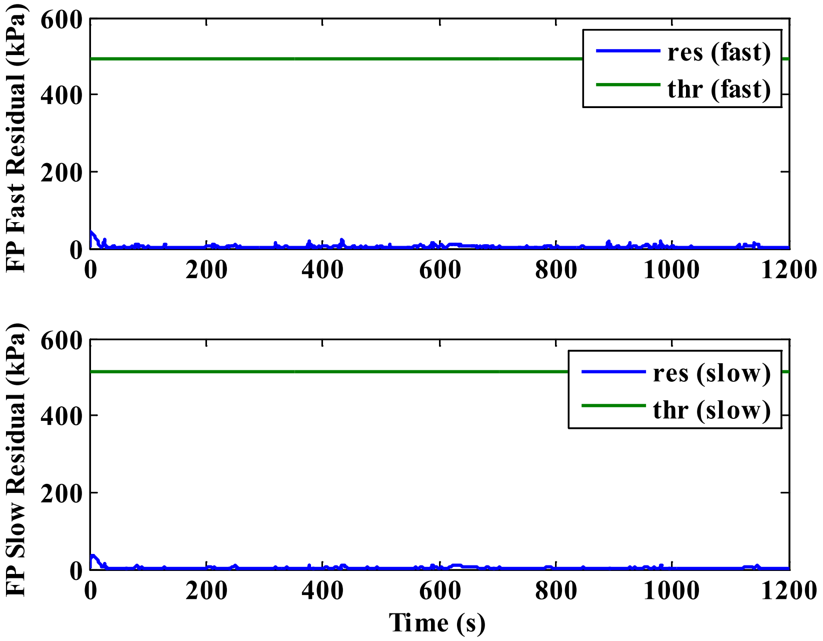


Figure 120: *FP* residuals with modified thresholds (*BP* failure at 200s).

d) FTP regression analyses:

Table 37: 40 CFR regression analyses of the FTP cycle (*BP* failure at 200s).

Parameters		Minimum	Cycle Value	Maximum
Engine Speed	Intercept (RPM)	-65.200	13.301	65.200
	Slope	0.950	0.992	1.030
	R^2	0.970	0.999	1.000
	Standard Estimate of Error (*SEE*)	0.000	9.251	105.000
Torque	Intercept (ft-lb)	-28.338	13.051	28.338
	Slope	0.830	0.935	1.030
	R^2	0.850	0.914	1.000
	SEE	0.000	116.818	141.690
Power	Intercept (bhp)	-7.532	4.144	7.532
	Slope	0.830	0.955	1.030
	R^2	0.850	0.917	1.000
	SEE	0.000	33.554	37.660
Work	Actual (bhp-hr)	23.679	24.267	26.171
	Reference (bhp-hr)	24.925		

G.6 Real-time SFDIA Data

a) FM1 residuals:

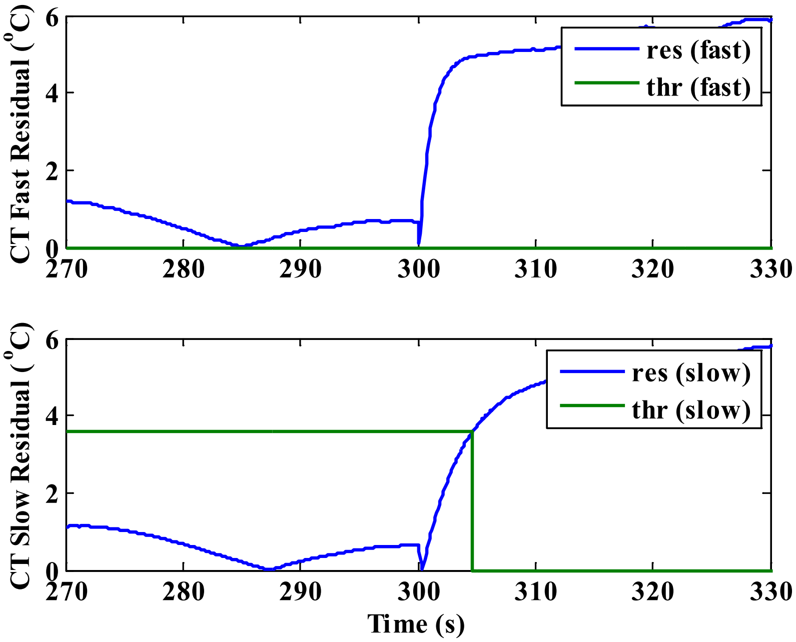


Figure 121: *CT* residuals with FM1 from real-time SFDIA tests.

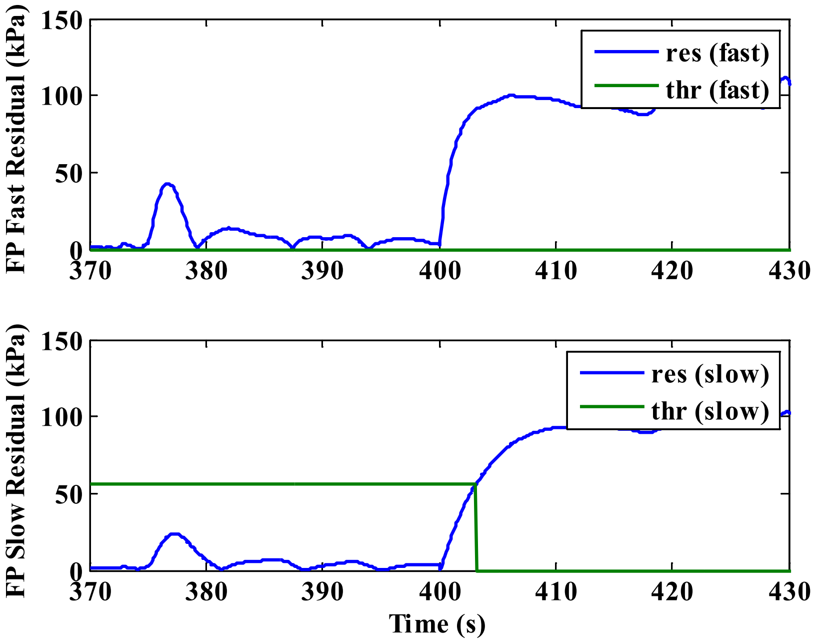


Figure 122: *FP* residuals with FM1 from real-time SFDIA tests.

b) FM2 residuals:

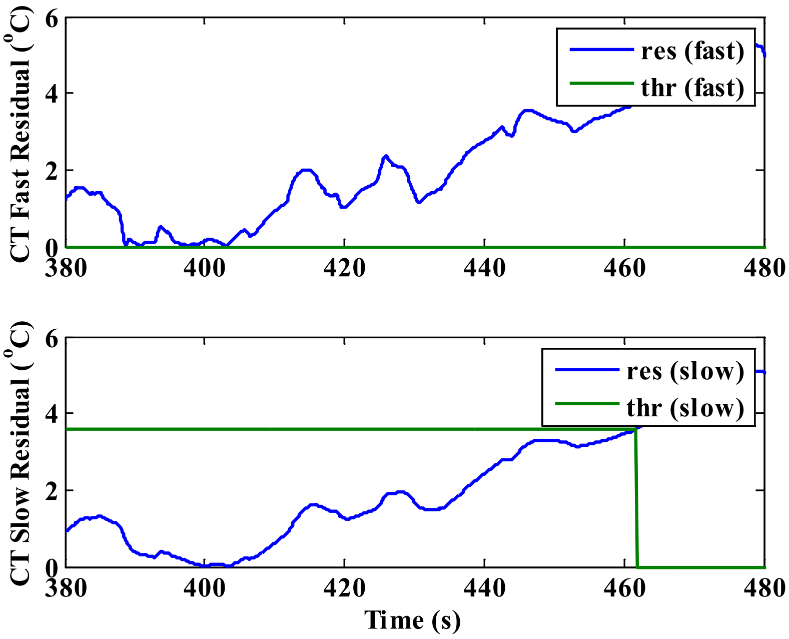


Figure 123: *CT* residuals with FM2 from real-time SFDIA tests.

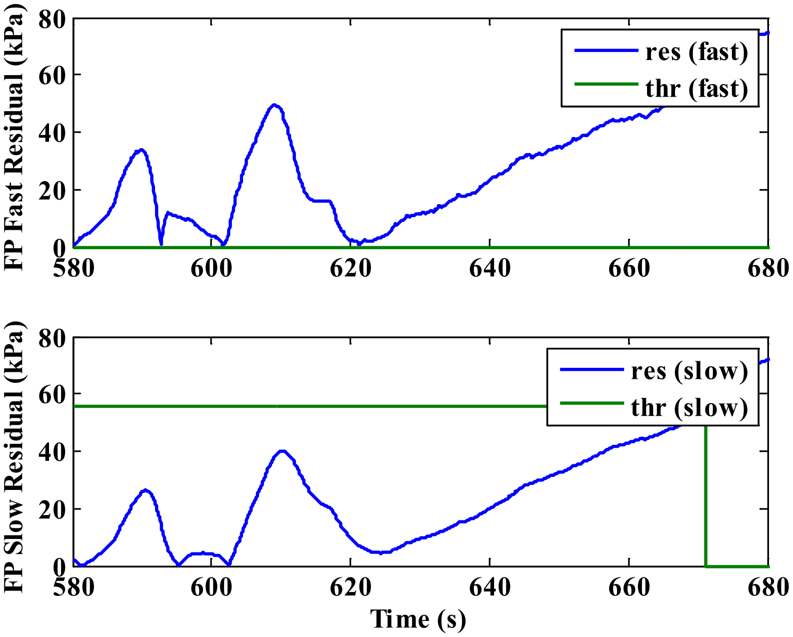


Figure 124: ***FP*** **residuals with FM2 from real-time SFDIA tests.**

c) FM3 residuals:

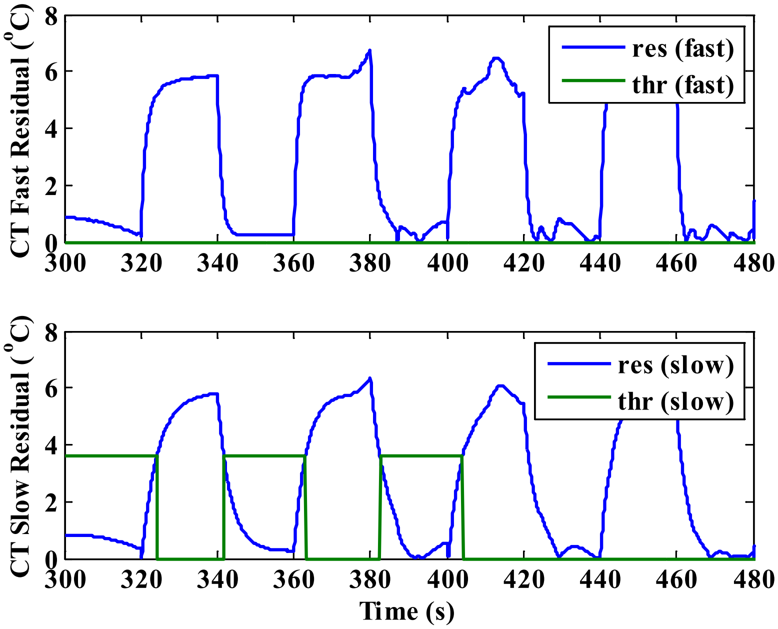


Figure 125: ***CT*** **residuals with FM3 from real-time SFDIA tests.**

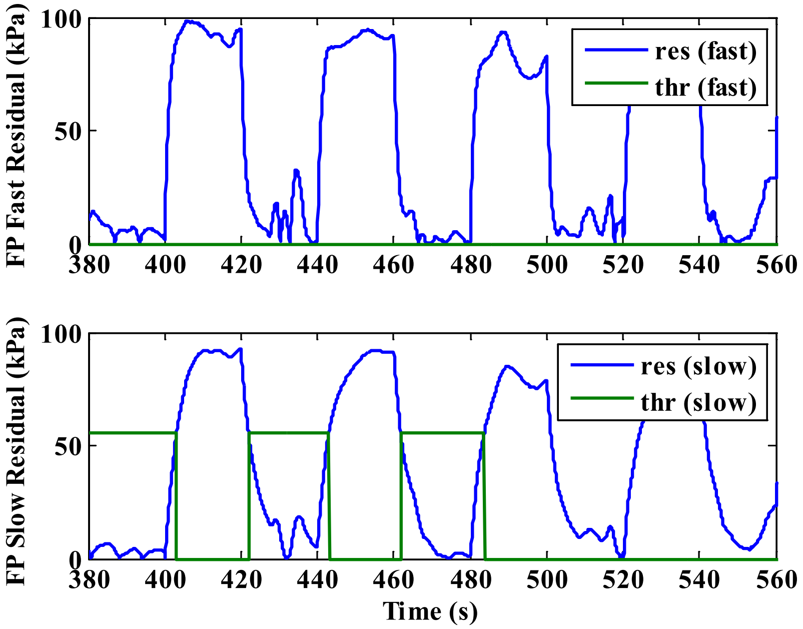


Figure 126: ***FP*** **residuals with FM3 from real-time SFDIA tests.**

G.7 Engine Performance Data from Real-time SFDIA Test

a) To estimate gaseous analyzer de-skew time using real-time SFDIA with FM1 data:

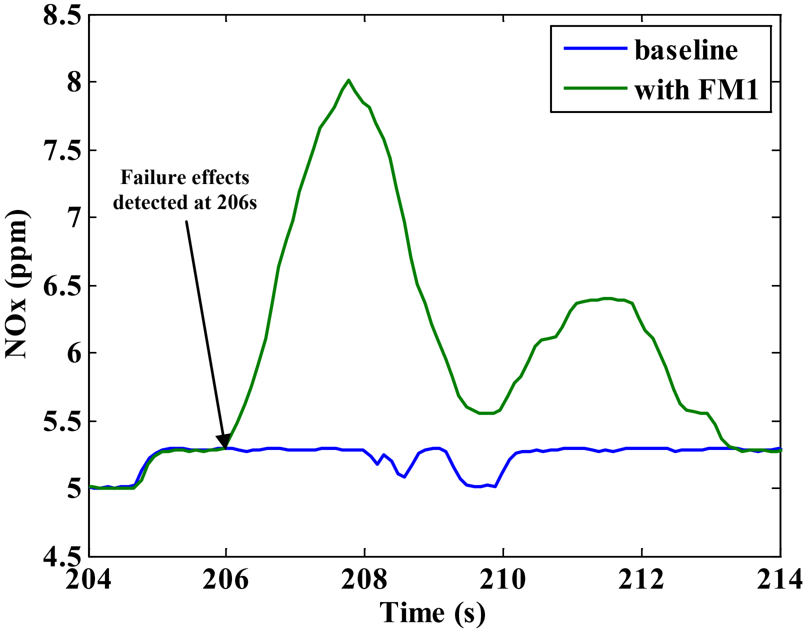


Figure 127: ***NOx*** **analyzer de-skew time estimation.**

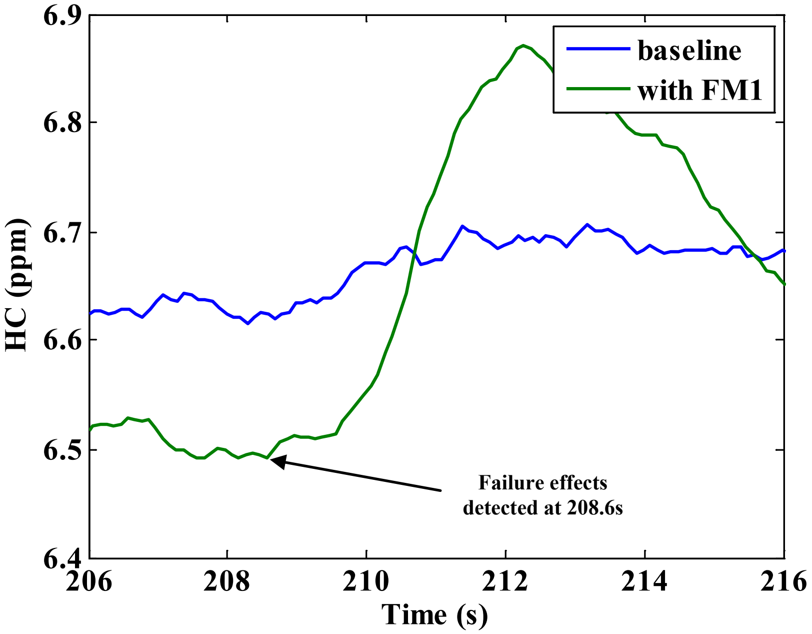


Figure 128: *HC* analyzer de-skew time estimation.

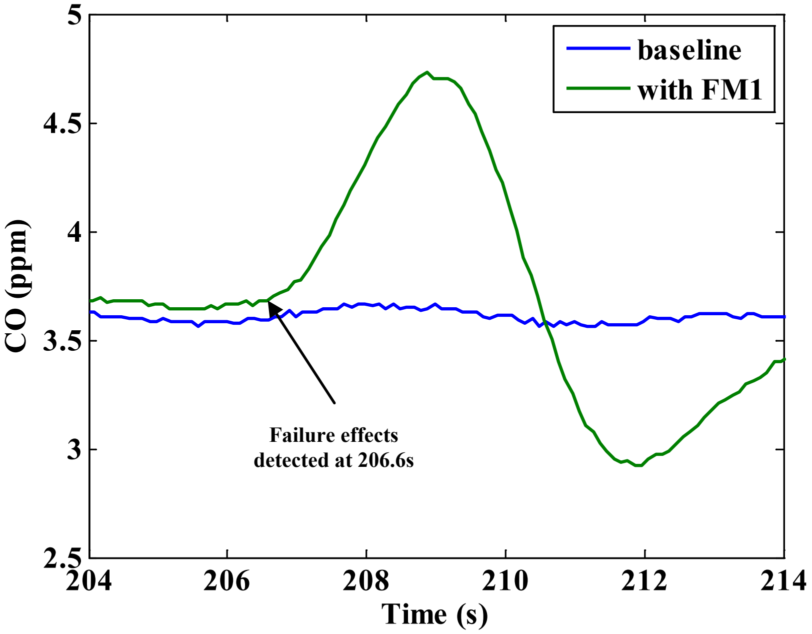


Figure 129: *CO* analyzer de-skew time estimation.

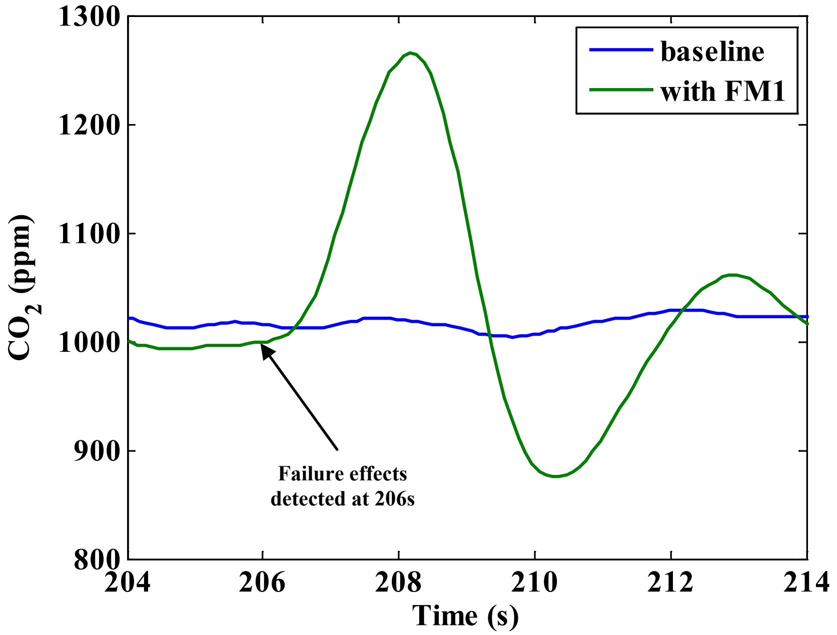


Figure 130: ***CO_2*** **analyzer de-skew time estimation.**

b) Selected continuous data from FM1 failure mode test:

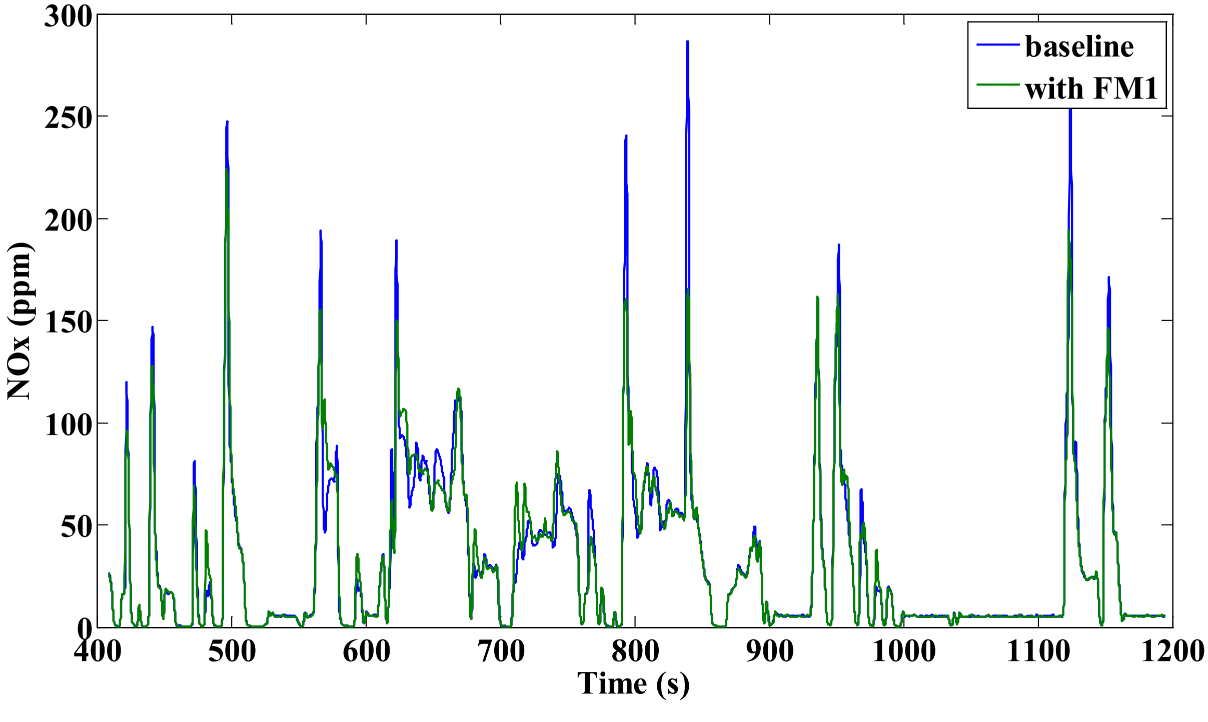


Figure 131: ***NOx*** **data during accommodation phase (FM1).**

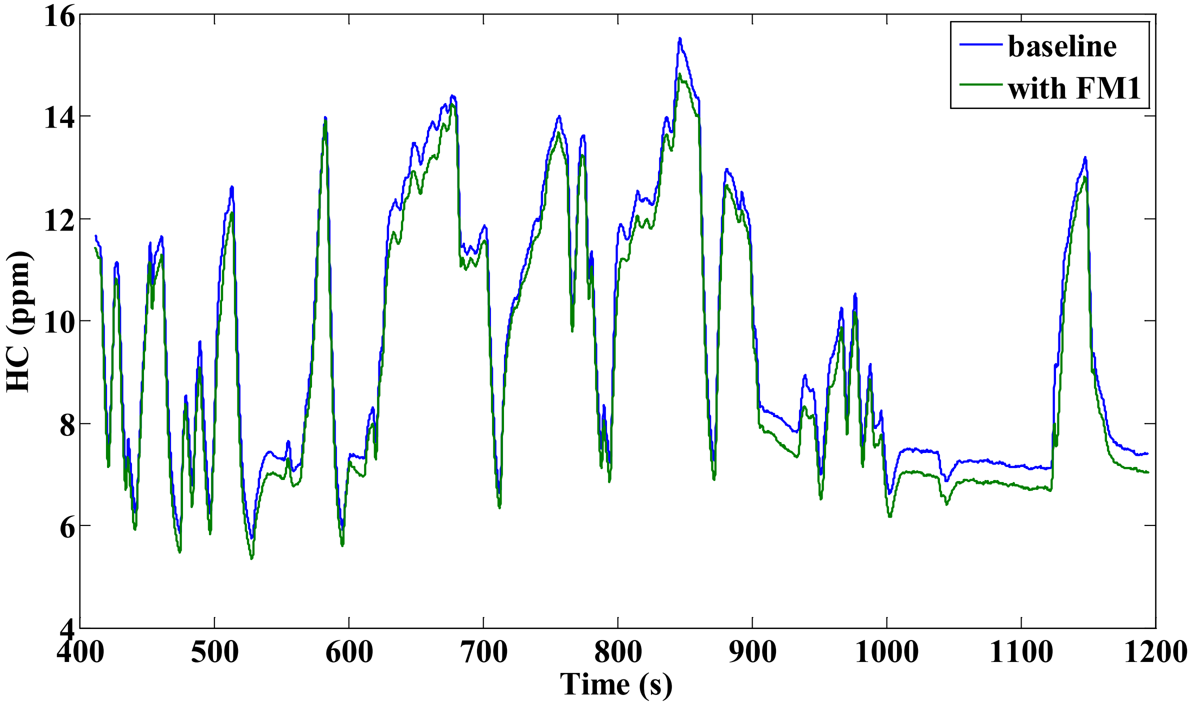


Figure 132: ***HC*** **data during accommodation phase (FM1).**

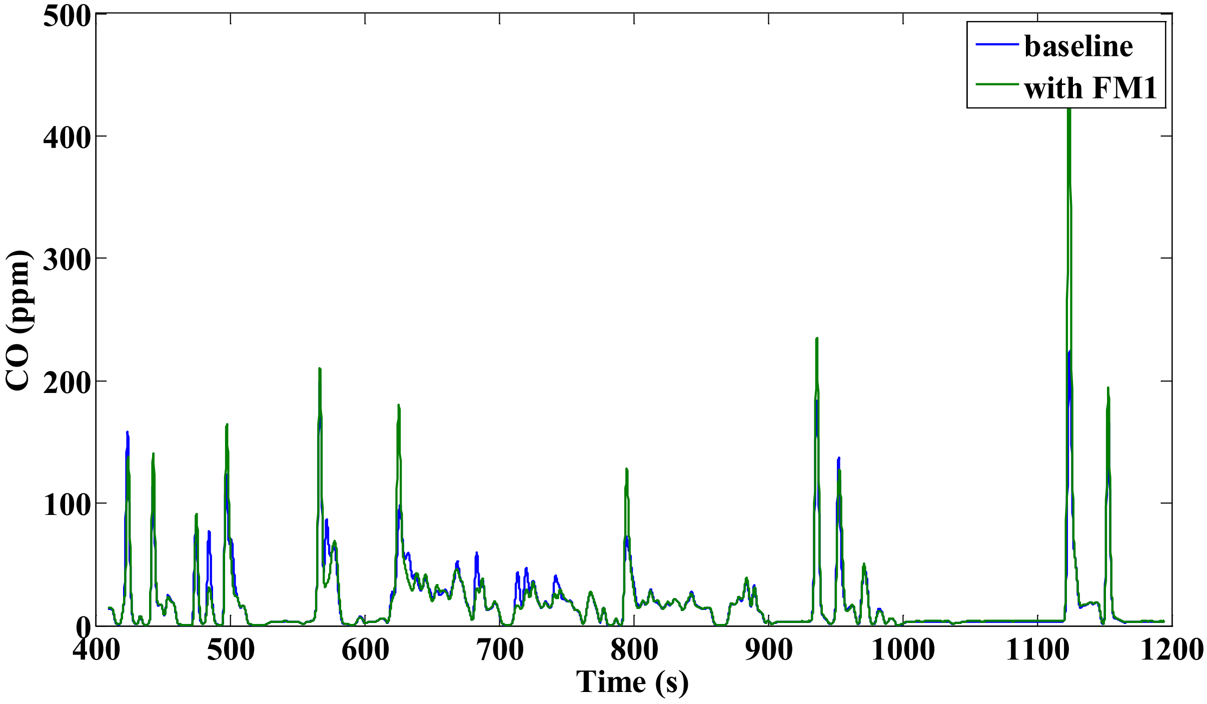


Figure 133: ***CO*** **data during accommodation phase (FM1).**

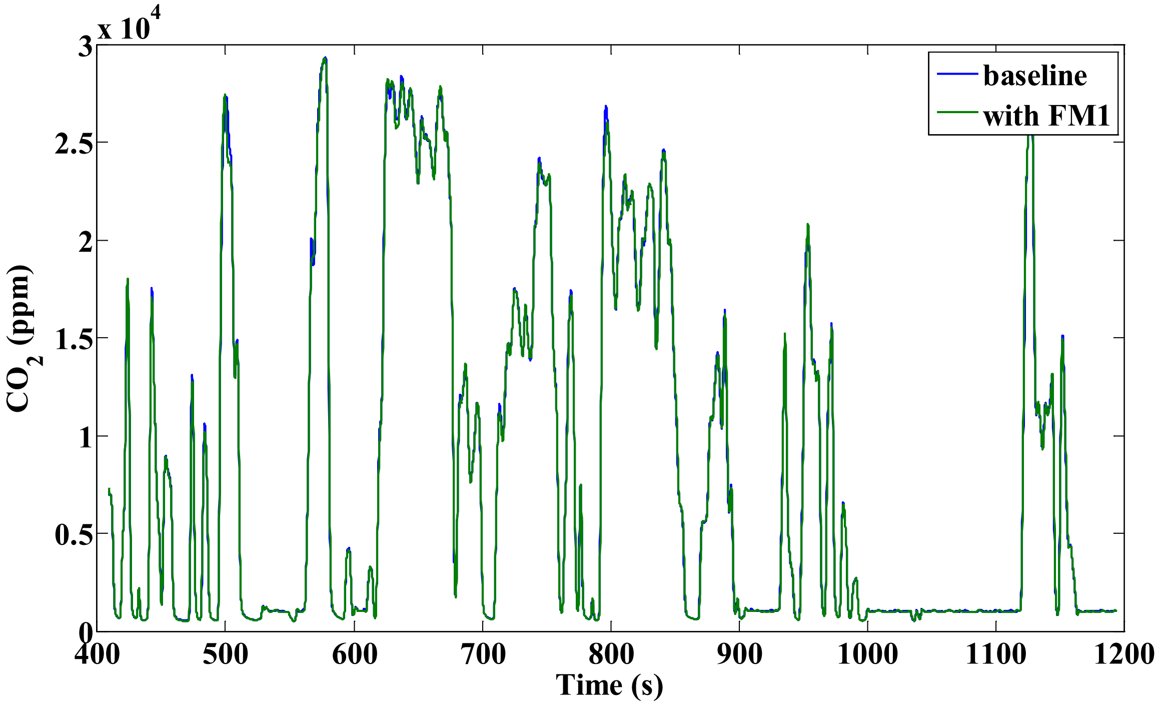


Figure 134: *CO_2* data during accommodation phase (FM1).

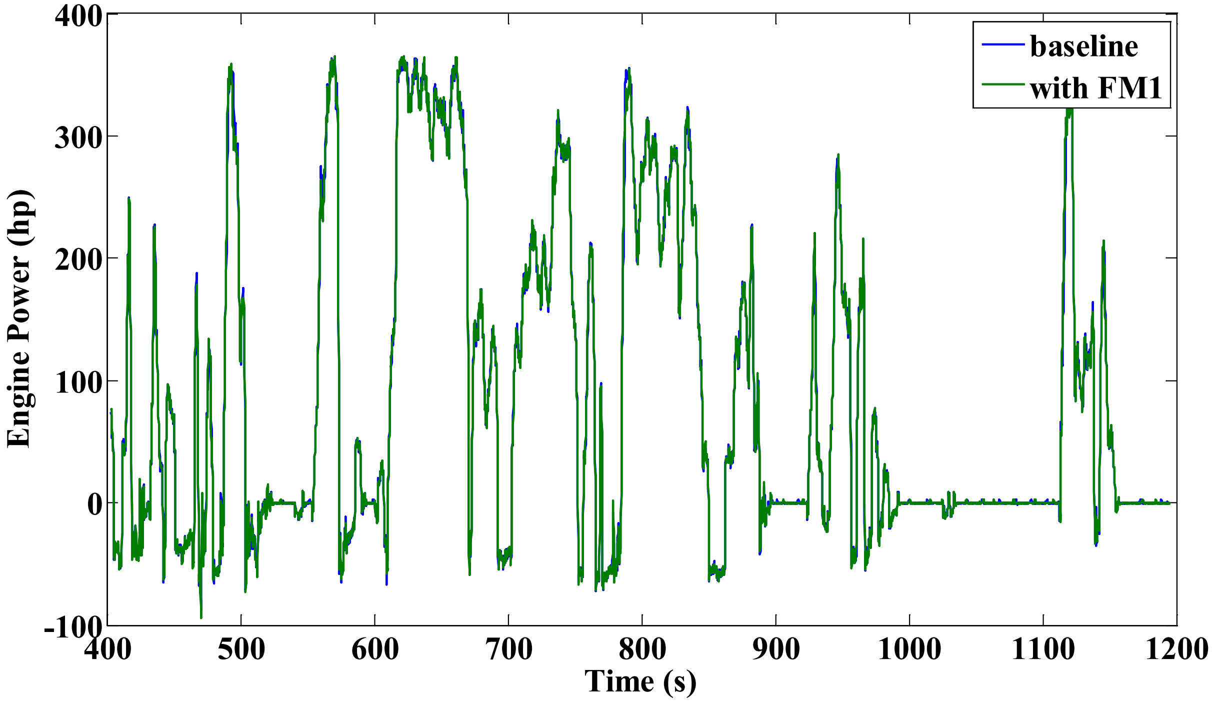


Figure 135: Engine power during accommodation phase (FM1).

c) Selected continuous data from FM2 failure mode test:

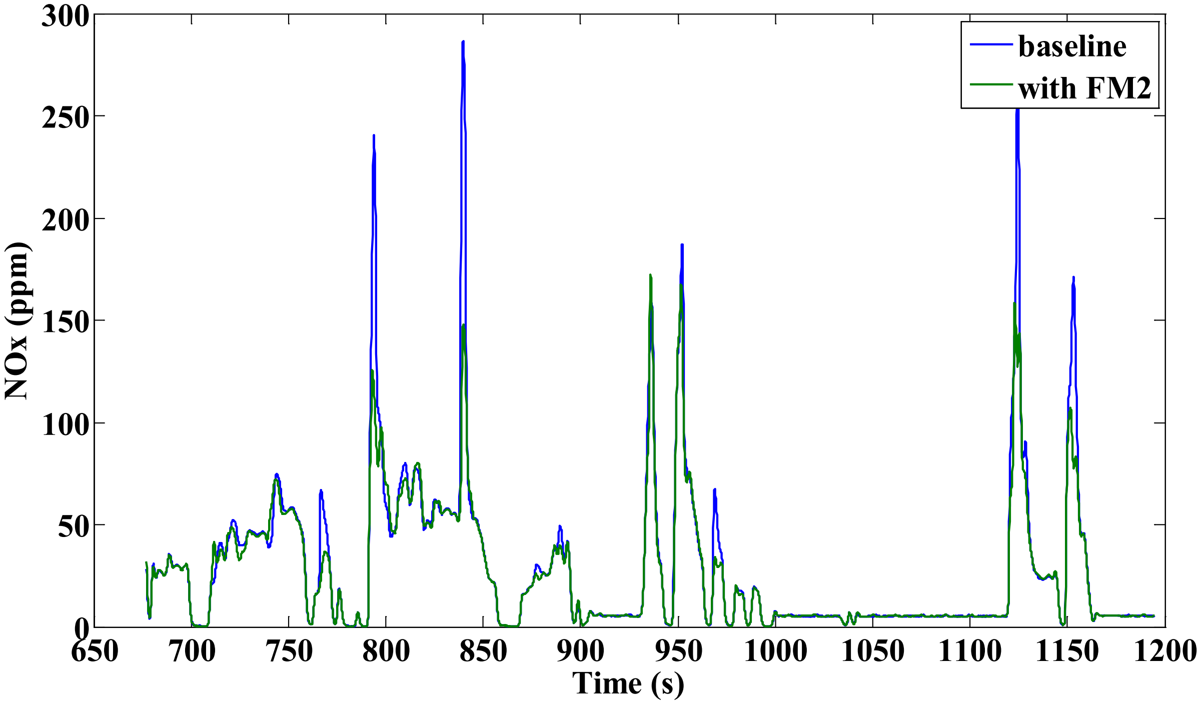


Figure 136: ***NOx*** **data during accommodation phase (FM2).**

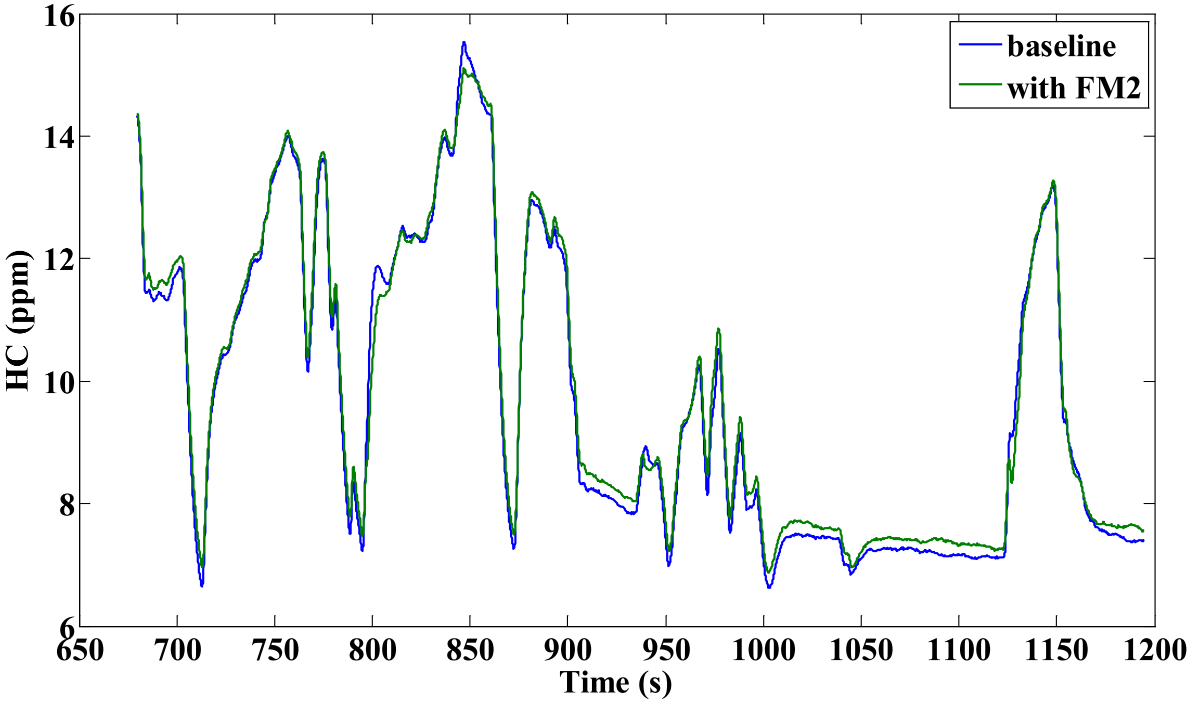


Figure 137: ***HC*** **data during accommodation phase (FM2).**

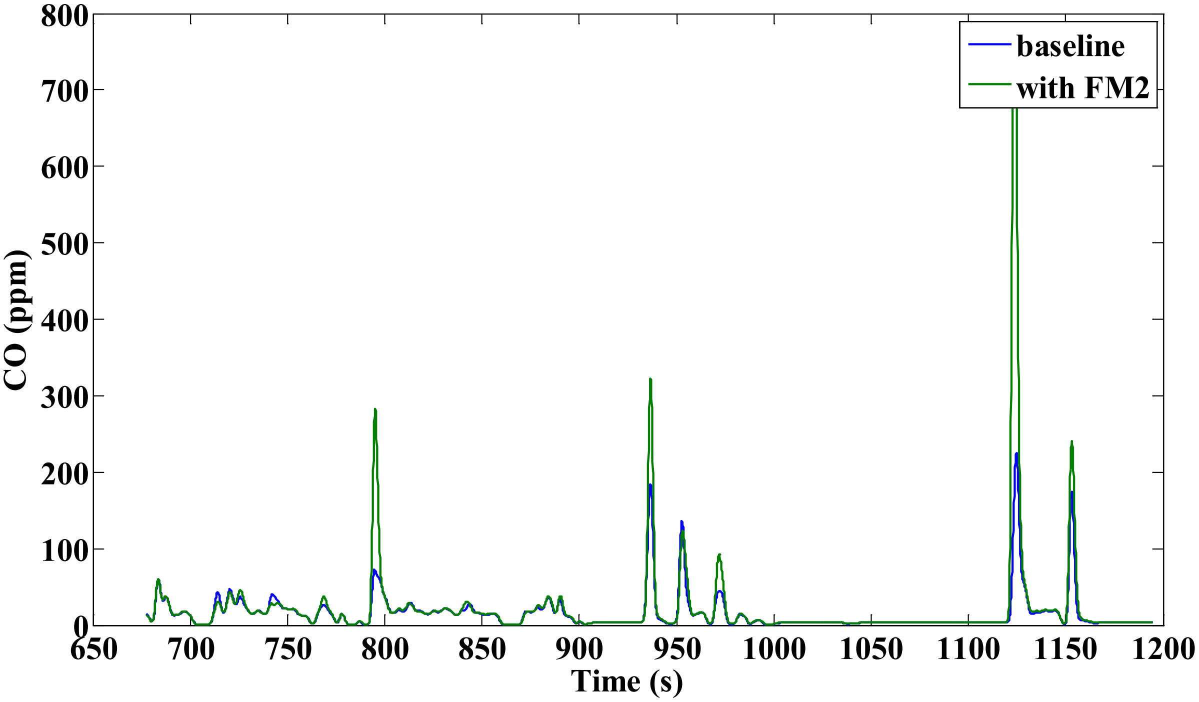


Figure 138: ***CO*** **data during accommodation phase (FM2).**

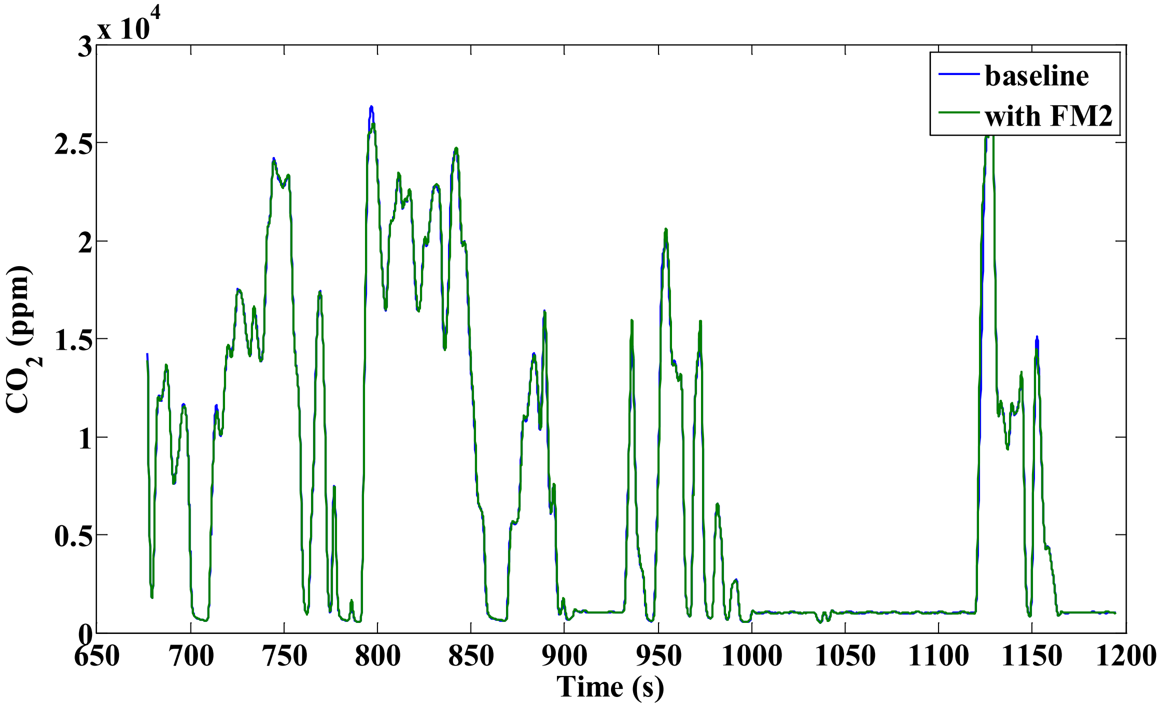


Figure 139: ***CO_2*** **data during accommodation phase (FM2).**

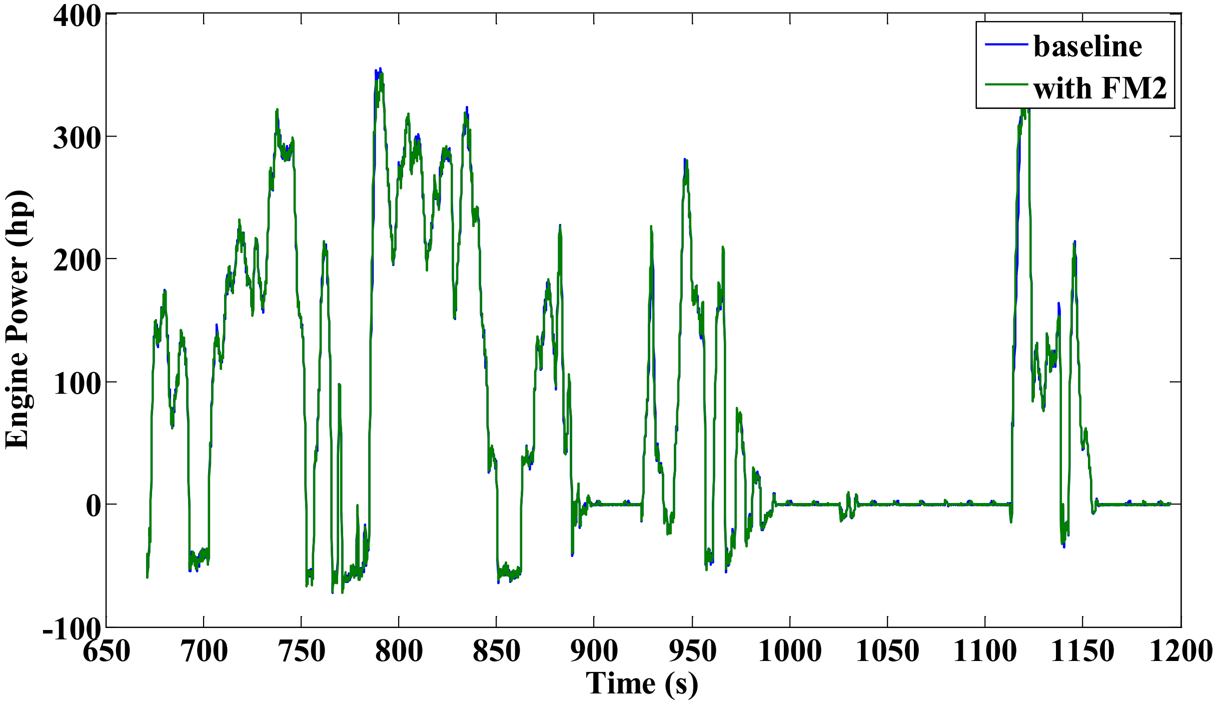


Figure 140: Engine power during accommodation phase (FM2).

d) Selected continuous data from FM3 failure mode test:

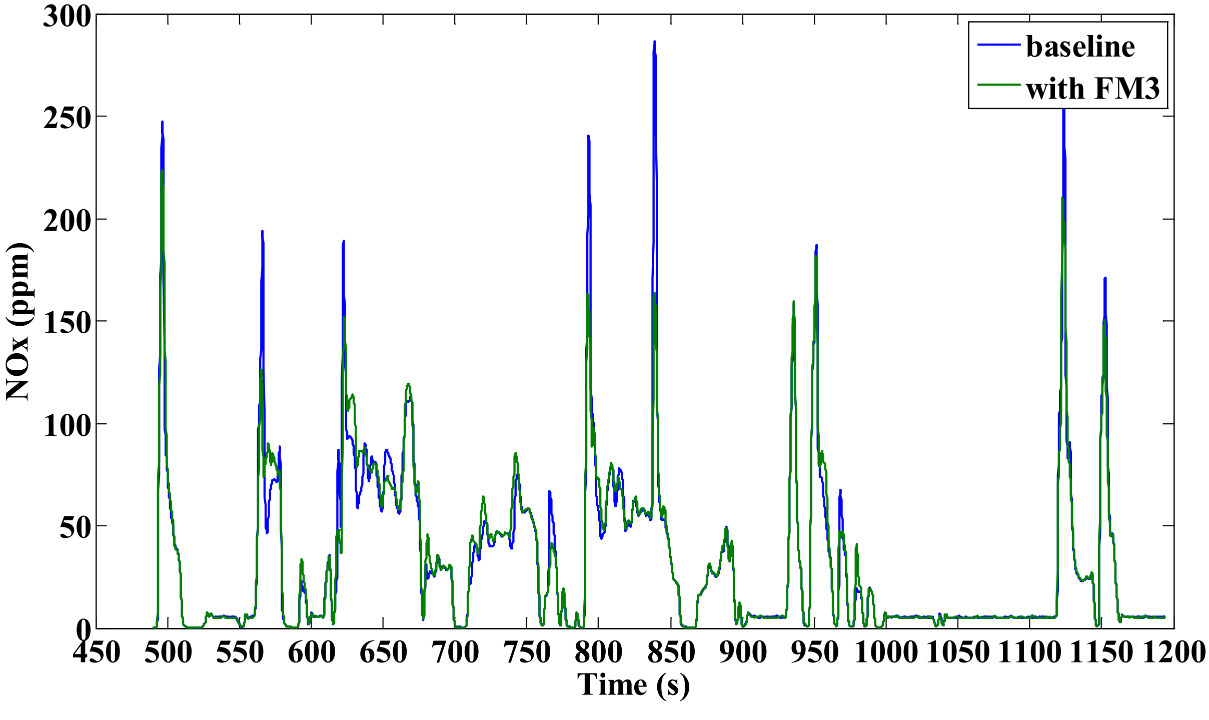


Figure 141: *NOx* data during final accommodation phase (FM3).

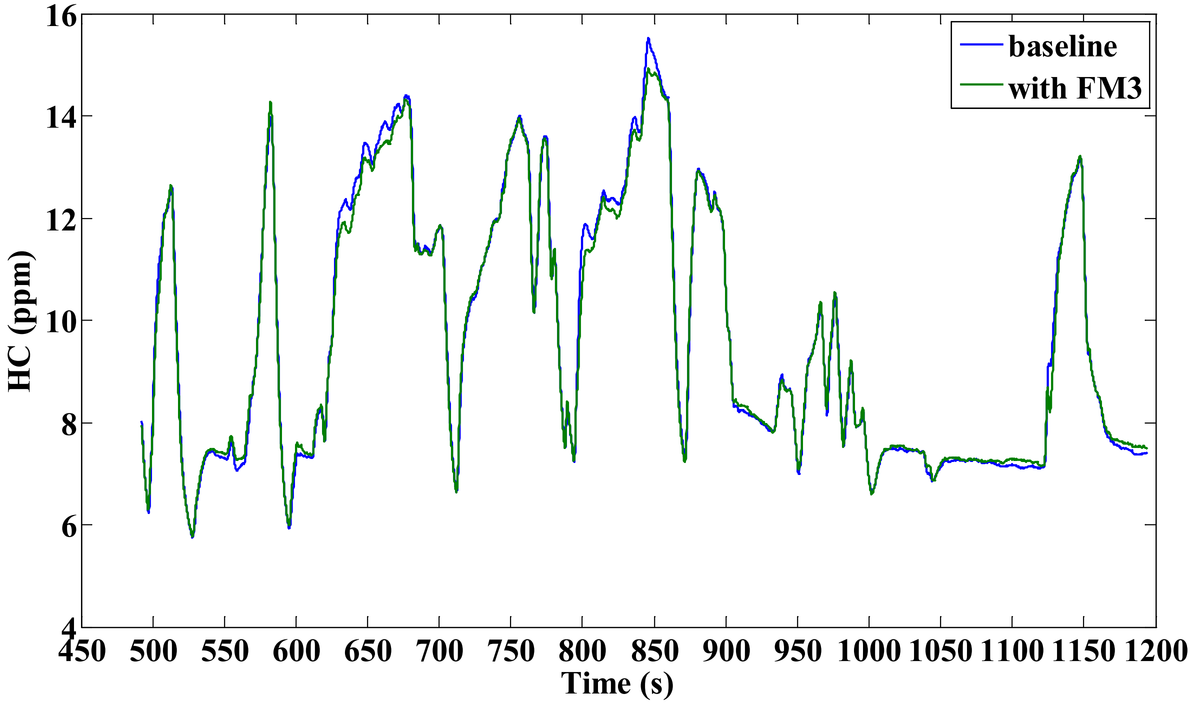


Figure 142: ***HC*** **data during final accommodation phase (FM3).**

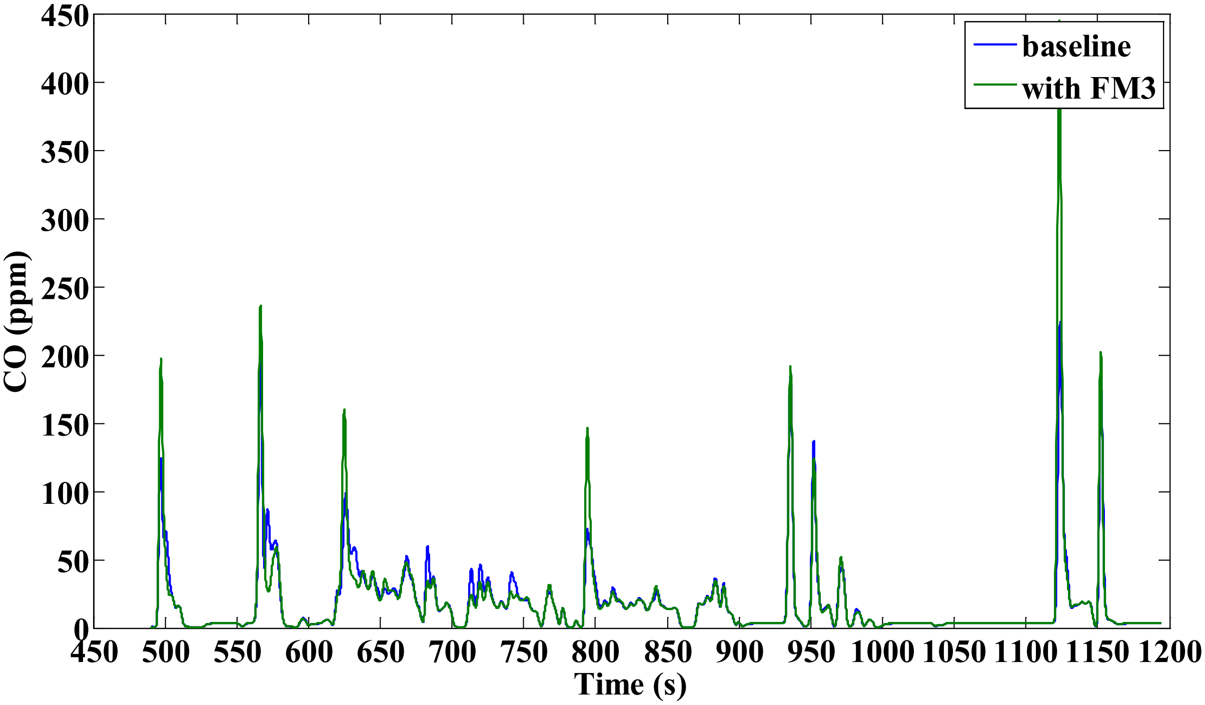


Figure 143: ***CO*** **data during final accommodation phase (FM3).**

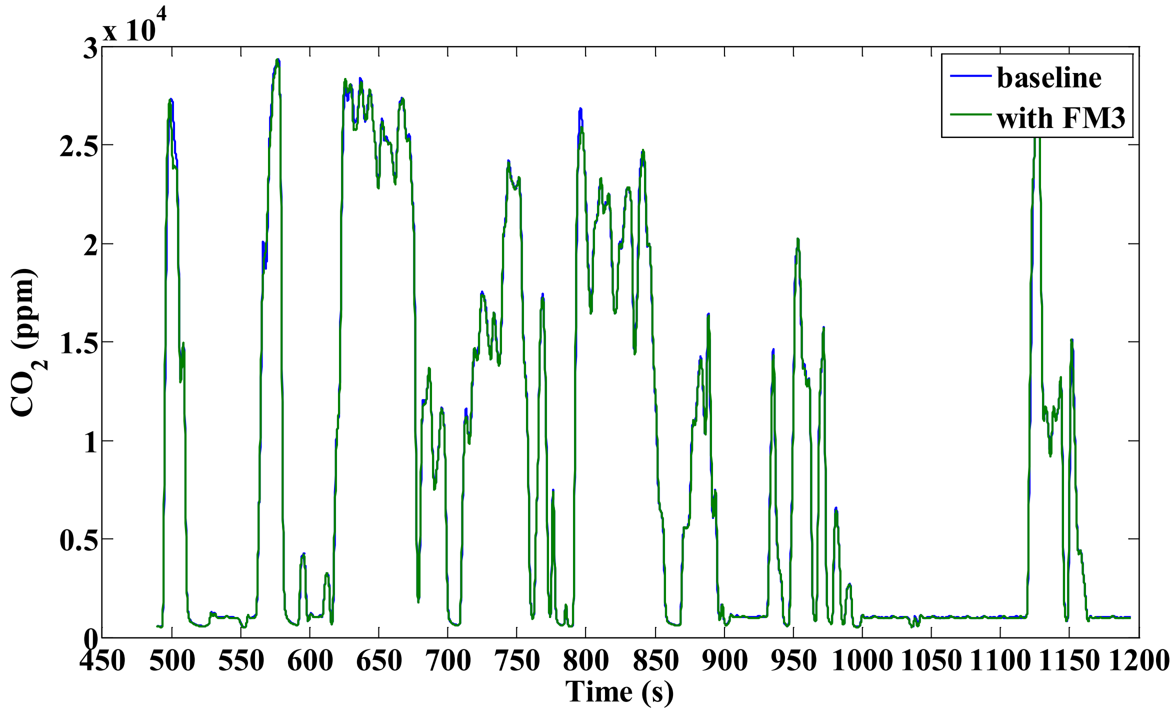


Figure 144: *CO_2* data during accommodation phase (FM3).

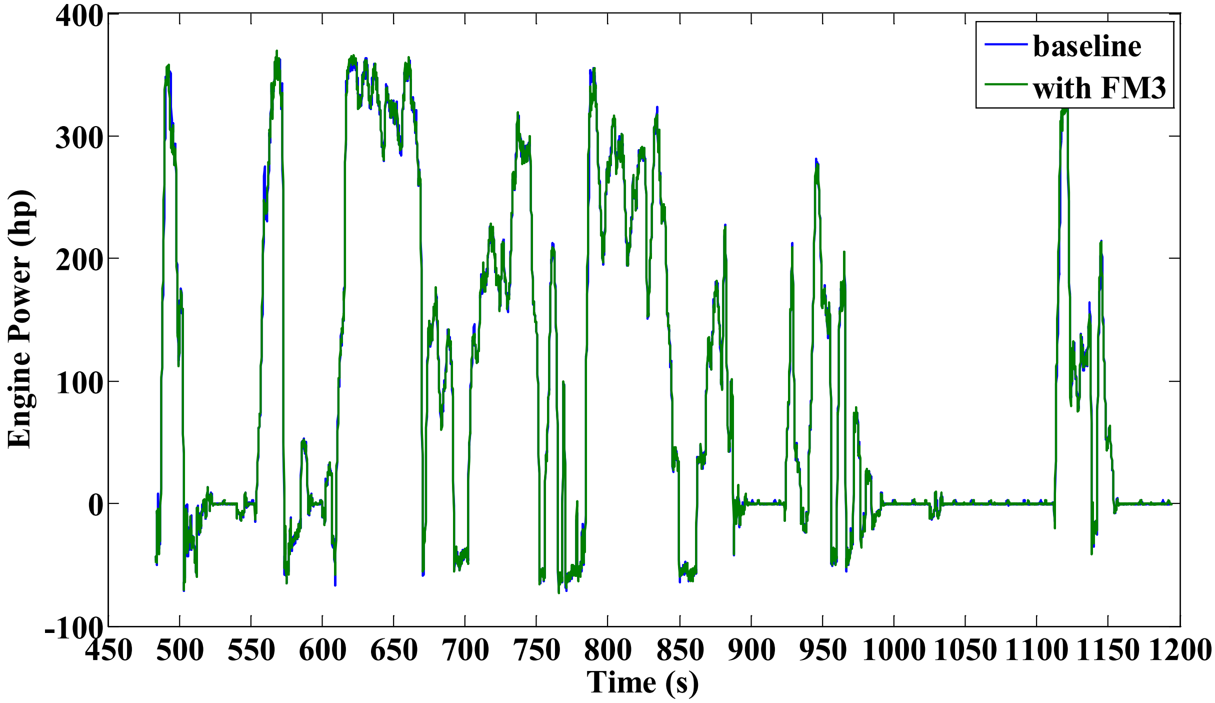


Figure 145: Engine power during accommodation phase (FM3).