A Realistic Simulation Testbed of A Turbocharged SI Engine System

A Platform for The Evaluation of Fault Diagnosis Algorithms & Strategies

Kok Yew Ng, Erik Frisk, Mattias Krysander, and Lars Eriksson
POC: K. Y. Ng (mark.ng@ulster.ac.uk)

September 17, 2019

The study of fault diagnosis on automotive engine systems has been an interesting and ongoing topic for many years. Numerous research activities have been carried out by both the automakers and research institutions to discover new and more advanced methods to perform diagnosis for better fault isolation (FI). Some of the research reported in this field have been reported in [1], [2], [3], [4], [5].

In most of the automotive systems today, the diagnostic systems monitor multiple components in the engine and are quite independent of each other. However, some faults have a tendency to manifest and trigger several other monitors either simultaneously or subsequently [6]. For instance, a disconnected intake system hose has a high potential to result in both flow and pressure faults down the line in the engine system. To overcome this, residuals from several monitors coupled with an intelligent algorithm are needed to enhance the accuracy in isolating the faults, both in locations as well as to identify the root cause of the problem. The ability to identify the root cause of the fault and to pinpoint its exact location to the correct component is crucial towards making the right corrective measures as well as to avoid the replacements of engine components that are otherwise misdiagnosed [7].

One of the main disadvantages in these existing diagnostic systems is that the faults are not detected in a chronological order. As a result, depending on the locations of the monitors in the engine as well as the propagation time of electrical signals, the manifested fault(s) may trigger the monitors much sooner than the root cause of the problem. Also, should a monitor break down or not running well, it may take more time to detect the root fault or worse, the fault may not get detected at all. This will in turn lead to wrong diagnostic reconfiguration efforts being taken on-board or an incorrect replacement of the so-called 'faulty' components off-board by the technician at the workshop [8], [9]. And these issues with misdiagnosis and robustness are especially critical in autonomous vehicular systems, where it is essential for the computers onboard the vehicles to know the health of the system such that corrective measures can be taken to protect the lives of occupants in the vehicles as well as those of other road users. Depending on the severity of the fault, the vehicle can either be reconfigured to operate at a reduced performance level to ensure safety until the vehicle is brought into the workshop for repair and maintenance works, or that the vehicle can be safely brought to a halt at a suitable location as soon as possible. Reports on similar concepts of reconfigurations for the purpose of fault-tolerant control, self-healing, and recoverability of autonomous systems can be found in [10], [11], [12]. The failure to detect and to isolate a fault, or to wrongly identify one, may cause the reconfiguration of the system to not be optimized and thus the health and lifespan of other components or the engine as a whole is affected. Therefore, an improved fault diagnosis method is crucial to not only identify the root cause of the problem, but also to reconfigure the engine immediately and correctly before the condition worsens. While many existing software and simulation packages provide interesting simulated studies on the dynamics of the engine system, very few have delved to include the design and analysis of fault diagnosis schemes. See [13], [14], [15], [16], [17].

This article presents a simulation testbed of the engine system whereby its operation can be realistically simulated using industrial standard driving cycles such as the Worldwide harmonized Light vehicles Test Procedures (WLTP), the New European Driving Cycle (NEDC), the Extra-Urban Driving Cycle (EUDC), as well as the EPA Federal Test Procedure (FTP-75) [18]. A GUI interface is available to enable the user to set simulation preferences such as the desired driving cycle as well as one of the 11 faults of interest. The performance of the developed fault diagnosis scheme can be analyzed without having to physically induce the faults on the physical engine, thus minimizing the risk of shortening its lifespan or causing permanent damages to the engine. The simulation environment, which is available as an open-source through the URL address https://github.com/nkymark/SimEngineBenchmark, is intended to enhance the development of theoretical and application of fault diagnosis of engine systems, with hope that researchers in the field can benefit from this to encourage research collaborations or used as a virtual laboratory for teaching purposes.

Modeling The Engine

The simulation environment uses a 4-cylinder single turbocharged spark ignited (TCSI) petrol engine as the testbed to design and to verify the performance of fault diagnosis schemes. Figure 1 shows the actual engine test bench used for data collection in the lab while Figure 2 shows the schematic diagram of the engine system, which consists of the following subsystems:

- Air filter: Ambient air enters the engine system and the filter prevents abrasive particulate matters from entering the engine block.
- Compressor: Modeled based on the radial compressor and driven by the turbine, air from the air filter is compressed, thus increasing the air mass flow, pressure, and temperature.
- Intercooler: Air from the compressor is cooled down while maintaining most of the air mass flow velocity.
- Throttle: This is used to control the pressure in the intake manifold, thus regulating the amount of fuel that goes into the engine.
- Intake manifold: The combustion mixture of air and fuel is distributed evenly to the 4 cylinders in the engine.
- Engine block: The combustion mixture is ignited to generate the torque for mechanical work.
- Exhaust manifold: Directs the gases produced by the combustion reactions in the engine to the turbine and the wastegate.
- Turbine: Harvests the energy of gases from the exhaust manifold to generate power to drive the compressor.
- Wastegate: A valve that bypasses the turbine and controls the power delivered by the turbocharger.
- Exhaust system: Gases in the engine system exits to the ambient environment.

The engine system is modeled using differential equations that describe the air flow through the subsystems in the engine. These equations are derived based on the mean value engine model (MVEM) for a TCSI engine as reported in [19], [20]. The key parameters of the vehicle and the engine used for this testbed, as well as the total model equations of the engine with a list of engine variables can be found in “Vehicle Parameters” and “Differential Equations of The TCSI Engine System and Corresponding Engine Parameters”. The system has 13 states; $\{T_{af}, p_{af}, T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}, T_{em}, p_{em}, T_t, p_t, \omega_t\}$, which are the temperatures (K) and pressures (Pa) in the air filter, compressor, intercooler, intake manifold, exhaust manifold, and turbine, as well as the turbine speed (rad/s), respectively. There are 6 actuators; $\{A_{th}, u_{wg}, \omega_{eREF}, \lambda_c, p_{amb}, T_{amb}\}$, which represent the throttle position area (m^2), the wastegate input ([0...1]), reference engine speed (rad/s), air-fuel ratio, and the ambient pressure (Pa) and temperature (K), respectively. The system has 9 sensors; $\{T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}, p_{em}, W_{af}, Tq_e\}$, where W_{af} and Tq_e are the mass flow in the air filter and the engine torque, respectively. See Table 1.

Generation of Reference Inputs and Controller Design

This section will discuss on the design of the PI-based boost controller with anti-windup to generate the control inputs for the throttle effective area, A_{th}, and the wastegate actuator for the turbocharger, u_{wg}. Figure 3 shows the closed-loop engine control system with the boost controller. See Tables S3 and S4 in “Differential Equations of The TCSI Engine System and Corresponding Engine Parameters” for the descriptions and values of the engine variables and parameters used in the following equations.

To estimate the gear shift points, it is assumed that the speed of the moving vehicle per 1000 rpm in 8th gear is 62.9 km/h. From the key vehicle parameters in Table S1, the vehicle speed per 1000 rpm (km/h), $v_{g,1000rpm}$ for each gear can be computed using

$$v_{g,1000rpm} = \frac{120\pi r_w}{\text{final gear ratio} \times \text{current gear ratio}}, \tag{1}$$

where r_w is the wheel radius (m).

The results are then tabulated in Table S2, which also shows the estimated gear shift points of the gearbox. The data in Tables S1 and S2, together with the information of vehicle speed versus time from the driving cycle profile, would provide the reference engine speed (rad/s), ω_{eREF} and reference engine torque (N·m), Tq_{eREF} for the boost controller. The reference engine speed can be computed using

$$\omega_{eREF} = \frac{V i_{gear}(V)}{r_w}, \tag{2}$$

where V is the vehicle speed (m/s) obtained from the driving cycle profile and the function $i_{gear}(V)$ is the gear shifting vector developed from V and the gear ratios in Table S1.

To obtain the reference engine torque, Tq_{eREF}, the force equation of the vehicle is first expressed using

$$m_v \dot{V} = F_w - F_d - F_r, \tag{3}$$

where F_w, F_d, and F_r are the forces (N) at the wheel, drag resistance force, and roll resistance force, respectively, and m_v is the mass of the vehicle (kg). The forces F_d and F_r can then be further written using

$$F_d = \frac{1}{2}\rho_a c_d A_f V^2, \tag{4}$$

$$F_r = m_v c_r g, \tag{5}$$

where $\rho_a = 1.29$ kg/m^3 is the air density, g is gravity (m/s^2), c_d the drag coefficient, and A_f represents the frontal area of the vehicle. If the torque produced at the wheel is written as

$Tq_w = F_w r_w$, then the reference engine torque can be finally expressed as

$$Tq_{eREF} = \frac{Tq_w}{i_{gear}(V)}. \tag{6}$$

To model the driver accelerator pedal interpretation, the reference brake mean effective pressure (BMEP) can first be expressed using

$$BMEP_{REF} = \frac{2\pi n_r Tq_{eREF}}{V_d}, \tag{7}$$

where Tq_{eREF} is the reference engine torque (N·m), V_d the displacement volume of the engine (m^3), and n_r represents the number of revolutions per power stroke of the engine (for a 4-cylinder engine, $n_r = 2$). As a result, the reference intake manifold and intercooler pressures, p_{imREF} and p_{icREF}, respectively, can be obtained using

$$\begin{aligned} p_{imREF} &= \frac{BMEP_{REF} + C_{P0}}{C_{P1}}, && (8) \\ p_{icREF} &= p_{imREF} + \Delta p_{thrREF}, && (9) \end{aligned}$$

where Δp_{thrREF} is the regulated pressure drop across the throttle (Pa). The constants C_{P0} and C_{P1} can be computed using

$$\begin{bmatrix} C_{P0} \\ C_{P1} \end{bmatrix} = \frac{2\pi n_r Tq_e}{V_d} \begin{bmatrix} -1 & p_{im} \end{bmatrix}^{+}, \tag{10}$$

where Tq_e is the measured engine torque (N·m) and p_{im} is the measured intake manifold pressure (Pa).

The reference throttle effective area (m^2), A_{thREF}, can then be computed using

$$A_{thREF} = W_{eiREF} \frac{\sqrt{R_a T_{amb}}}{\Psi_{thREF}}, \tag{11}$$

where R_a is the gas constant (J/(kg·K)), T_{amb} the ambient temperature, and Ψ_{thREF} is the reference throttle flow coefficient (%). The reference mass flow into the engine (kg/s), W_{eiREF}, and Ψ_{thREF} can be computed using

$$\begin{aligned} W_{eiREF} &= \frac{C_{\eta_{vol}} V_d \; \omega_{eREF} p_{imREF}}{4\pi R_a (r_c - 1) T_{im}} \left(r_c - \left(\frac{p_{em}}{p_{imREF}} \right)^{\kappa_{em}} \right), && (12) \\ \Psi_{thREF} &= \Pi_{thREF} \sqrt{\frac{2\kappa_{th}}{\kappa_{th} - 1} \left(\Pi_{thREF}^{\frac{2}{\kappa_{th}}} - \Pi_{thREF}^{\frac{\kappa_{th}+1}{\kappa_{th}}} \right)}, && (13) \end{aligned}$$

where $C_{\eta_{vol}}$ is the volumetric efficiency constant, r_c the compression ratio, T_{im} the intake manifold temperature (K), p_{em} the exhaust manifold pressure (Pa), κ_{em} the ratio of specific heats at the exhaust, κ_{th} the ratio of specific heats at the throttle, and the pressure ratio in the throttle, Π_{thREF}, can be obtained using

$$\Pi_{thREF} = \frac{p_{imREF}}{max(p_{imREF}, p_{ic})}, \tag{14}$$

where p_{ic} is the intercooler pressure (Pa).

To design the controller with anti-windup for the throttle, the reference throttle position, α_{thREF}, can be computed using

$$\alpha_{thREF} = \alpha_{thFF} + \alpha_{thFB} \tag{15}$$

where α_{thFF} and α_{thFB} are the feedforward and feedback components of the controller for the throttle position, respectively. Using the solution from (11), α_{thFF} can be expressed using

$$\alpha_{thFF} = -\frac{a_0}{2a_2} \pm \sqrt{\frac{A_{thREF} - a_0}{a_2} + \left(\frac{a_1}{a_2}\right)^2}, \tag{16}$$

where the constants a_0, a_1, and a_2 are parameters obtained from measurements in the engine lab. The feedforward component of the controller enables it to respond quickly to changes made to the engine, for example a rapid acceleration when the accelerator pedal is depressed fully onto the floor.

The feedback component of the controller, α_{thFB} can be obtained using

$$\alpha_{thFB} = K_{p,th} e_{im} + \frac{K_{p,th}}{Ti_{th}} \int e_{im} + K_{p,th}(\alpha_{thREF,SAT} - \alpha_{thREF})\ dt, \tag{17}$$

where $K_{p,th}$ and $T_{i,th}$ are the proportional and integral gains of the feedback controller, respectively, and $e_{im} = p_{imREF} - p_{im}$. The saturation of the reference throttle position, $\alpha_{thREF,SAT}$, can be defined as the static nonlinearity

$$\alpha_{thREF,SAT} = \begin{cases} \alpha_{th,MAX}, & \text{if } \alpha_{thREF} > \alpha_{th,MAX} \\ \alpha_{thREF}, & \text{if } \alpha_{th,MIN} < \alpha_{thREF} < \alpha_{th,MAX} \\ \alpha_{th,MIN}, & \text{if } \alpha_{thREF} < \alpha_{th,MIN} \end{cases}, \tag{18}$$

where $\alpha_{th,MAX}$ and $\alpha_{th,MIN}$ are the maximum and minimum allowed actuation signals for the throttle position, respectively. The feedback component of the controller ensures that the engine system is able to follow its references during operation.

As for the controller for the wastegate input, it consists of only a feedback component, and it is expressed using

$$u_{wgFB} = K_{p,wg} e_{ic} + \frac{K_{p,wg}}{Ti_{wg}} \int e_{ic} + K_{p,wg}(u_{wgREF,SAT} - u_{wgREF})\ dt, \tag{19}$$

where $K_{p,wg}$ and $T_{i,wg}$ are the proportional and integral gains of the feedback controller, respectively, and $e_{ic} = p_{ic} - p_{icREF}$. The saturation of the reference wastegate input, $u_{wgREF,SAT}$, can be defined as the static nonlinearity

$$u_{wgREF,SAT} = \begin{cases} u_{wg,MAX}, & \text{if} \;\; u_{wgREF} > u_{wg,MAX} \\ u_{wgREF}, & \text{if} \;\; u_{wg,MIN} < u_{wgREF} < u_{wg,MAX} \\ u_{wg,MIN}, & \text{if} \;\; u_{wgREF} < u_{wg,MIN} \end{cases}, \tag{20}$$

The design of the controller for the engine is then verified in simulations using the reference engine torque, Tq_{eREF}, and speed, ω_{eREF}, generated from a selected driving cycle. Figure 4 shows the reference and actual torque of the engine during the WLTP driving cycle. It can be seen that the actual engine torque is able to follow its reference well.

The engine model is also verified against the actual test bench system to ensure that the model is realistic and viable for simulations of real-world operations. Considering that the engine is a system with 13 states and is highly-nonlinear with many interconnected subsystems where air-flow can travel upstream and downstream depending on the pressure difference, it has always been a challenge to accurately model an entire engine system. The dynamics of the engine model is compared with that of the actual test bench system in the lab, which is fully controlled in real-time and have its sensor measurements visualized and recorded using a dSPACE MicroAutoBox + RapidPro system as well as INCA tools. The sensor measurements recorded for comparison between the model and the test bench are the air filter mass flow, $y_{W_{af}}$, intercooler temperature, $y_{T_{ic}}$, intake manifold pressure, $y_{p_{im}}$, and intercooler pressure, $y_{p_{ic}}$. During the EUDC run, the engine model produces sensor measurements that are close to the actual test bench system. These results show that the engine model is realistic and that it represents the actual engine system accurately. See Figure 5.

Fault Scenarios

The simulation testbed considers 11 sensor, actuator, and variable faults in different parts of the engine system; 6 variable faults ($f_{p_{af}}, f_{C_{vol}}, f_{W_{af}}, f_{W_{th}}, f_{W_c}, f_{W_{ic}}$), 1 actuator measurement fault ($f_{x_{th}}$), and 4 sensor measurement faults ($f_{y_{W_{af}}}, f_{y_{p_{im}}}, f_{y_{p_{ic}}}, f_{y_{T_{ic}}}$). The faults are of different degrees of severity. Some faults are less severe and the engine can be reconfigured to a reduced performance operation mode to accommodate the faults until the vehicle is sent into the workshop for repair and maintenance. Some other faults are more severe that if not detected and isolated promptly, might cause permanent and serious damages to the engine system, which in turn will endanger the occupants in the vehicle as well as other road users.

Fault Types and Classification

The faults included in this simulation testbed can be categorized into 3 types; sensor fault, actuator fault, and variable fault.

Sensor Faults

This research considers 4 sensor measurement faults. They are $f_{y_{W_{af}}}$, $f_{y_{p_{im}}}$, $f_{y_{p_{ic}}}$, and $f_{y_{T_{ic}}}$. The nature of these faults is due to electrical or mechanical errors that lead to either an offset or a deviation in the sensor measurements.

The $f_{y_{W_{af}}}$ fault indicates a sensor measurement error in the air filter flow. The air filter flow sensor measures the amount of air that goes into the engine. As such, it is critical for this fault to be fixed and the necessary parts replaced as soon as it is detected.

The remaining of the sensor faults are pressure and temperature measurement errors in the intercooler ($f_{y_{p_{ic}}}$, $f_{y_{T_{ic}}}$) and intake manifold ($f_{y_{p_{im}}}$) of the engine. The pressure measurement errors, $f_{y_{p_{ic}}}$ and $f_{y_{p_{im}}}$, produce a deviation of 20% in the measured values. The $f_{y_{p_{im}}}$ is modeled using a long term incipient fault to indicate a drift in the sensor signal over time, while the $f_{y_{p_{ic}}}$ is modeled using repeating abrupt pulses. As for the $f_{y_{T_{ic}}}$, it indicates an offset in the sensor that measures the temperature in the intercooler. This fault is also modeled using repeating abrupt pulses.

Actuator Faults

The actuator fault considered in this research is the $f_{x_{th}}$, which indicates a throttle position actuator error, where an angular fault in the actuator leads to a flow error. This fault is modeled using repeating abrupt pulses. As this fault directly affects the throttle and subsequently the amount of fuel used for combustion, its severity level is medium.

Variable Faults

This research considers 6 variable faults. They are $f_{p_{af}}$, $f_{C_{vol}}$, $f_{W_{af}}$, $f_{W_{th}}$, f_{W_c}, and $f_{W_{ic}}$. The nature of these faults is due to physical and mechanical damages to certain parts of the engine system, leading to either a pressure drop, leakages, or a performance degradation of the engine itself. As a result, most of these faults are of high severity.

The $f_{p_{af}}$ fault indicates a pressure drop in the air filter due to a restriction in the flow. The fault is modeled using a long term abrupt fault to simulate a constant restrictive flow in the air filter.

The fault $f_{C_{vol}}$ indicates that the intake valve timing actuator is stuck at an arbitrary position. As a result, this affects the volumetric efficiency of the engine, which subsequently affects the overall performance of the engine system such as power output, emission control, and fuel consumption. The volumetric efficiency is modeled as a function of the intake valve timing actuator position. Therefore, this is a serious fault and has to be detected and isolated quickly. This fault is modeled using repeating abrupt pulses.

The remaining of the variable faults are leakages that could happen in different parts of the engine system, such as the air filter ($f_{W_{af}}$), compressor (f_{W_c}), intercooler ($f_{W_{ic}}$), and throttle ($f_{W_{th}}$). These leakages in the form of varying diameter orifices would lead to a change in the mass flow. Other than $f_{W_{af}}$ which is of medium severity, the other mass flow faults are of high severity as they happen after the compressor and closer to the engine block, where the pressure is higher. Unattended mass flow faults in these engine components could cause a degradation in the engine performance such as overpressure and increased emissions as well as damages to the engine components themselves, especially if external abrasive particulate matters manage to enter the engine through the leakage orifices. As a result, this could lead to faults in other parts of the engine system, thus making efforts to isolate the original leakage fault more difficult.

The faults and their characteristics are summarized in Table 2.

Fault Isolation Analysis from Model

Using the differential equations in "Differential Equations of The TCSI Engine System and Corresponding Engine Parameters", a Structural Model of the engine system with the faults defined in Table 2 can be constructed. The Structural Model shows the relationships amongst the unknown variables of the engine system, the known variables (actuators and sensors), and the faults. The Structural Analysis is a useful tool for early determination of fault isolability. It shows how different levels of knowledge about faults can be incorporated in a Structural Fault Isolability Analysis and their results in different fault isolability conclusions [21].

Using the Structural Model, the Fault Isolation Matrix (FIM) can then be generated for initial fault isolation analysis. The FIM is a square matrix where each row and column corresponds to a fault. A dot is placed at position (i, j) to indicate that fault f_i is not isolable from fault f_j. Figure 6 shows the FIM for the engine system given the current sensors setup. In the figure, 2 pairs of non-isolable faults can be observed; $f_{p_{af}}$ is not isolable from $f_{W_{af}}$, and $f_{W_{th}}$ is not isolable from $f_{x_{th}}$. However, this is a best case performance of fault isolation in theory as this method does not consider the magnitudes and shapes of the faults, model uncertainties, and disturbances. Thus, this method is not able to provide an accurate representation of the actual fault isolation capability. As such, this simulation testbed with bounded magnitudes of the faults and with the consideration for sensor noise, will be able to provide a more realistic outlook on the fault isolability for the engine system. See "Structural Model and FIM: A General Tutorial" for a general tutorial on Structural Model and FIM. Further information on the studies and development of the Structural Model and the FIM can be found in [21].

Design and Generation of Residuals

Introduction

Initially, 9 residuals are generated based on the sensors setup described in Table 1. The simulation testbed is by default distributed with a state observer/estimator, which is constructed using the differential equations that describe the engine system. As such, the observer/estimator is able to provide an estimate of the internal states of the engine system. The design of the observer can be replaced by other types of observers in the literature such as the sliding mode observer (SMO) [22], [23], Kalman Filter [24], [25], reduced order observer [26], [27], etc. according to the designer's wish. Therefore, this simulation testbed also enables researchers to design, develop, and compare strategies for residuals generator design for applications of states estimation and fault diagnosis of automotive engine systems. The residuals are generated by computing the difference between the sensor outputs of the model in Figure 3 and the estimated outputs of an estimator/observer of the engine system: $r_i = \hat{y}_i - y_i$, where $\hat{y}_i$ and y_i represent the i^{th} estimated and actual sensor outputs of the model, respectively. Figure 7 shows the overall block diagram representation of the closed-loop engine control system in Figure 3 with the residuals generator. The residuals are then normalized using the standard deviation of the fault-free data as the measure of scale,

$$r_{i,N} = \frac{r_i - \mu_{r_i}}{\sigma_{r_{i,FF}}} \tag{21}$$

where $r_{i,N}$ is the normalized residual, μ_{r_i} the mean of the residual, and $\sigma_{r_{i,FF}}$ represents the standard deviation of the corresponding residual during a fault-free scenario. These normalized residuals are called the 'Original 9' and they are listed in Table 3.

In a nominal fault-free scenario, all residuals have zero mean values. This indicates that both the model and the estimator of the engine produce similar actual and estimated outputs, respectively, while being excited by the same control inputs. Figure 8 shows the results of the residuals generated for a simulated fault-free scenario during the WLTP driving cycle profile. The dashed lines in Figure 8 represent the default fault detection threshold, J, which determines if the residuals have triggered, that is $|r_{i,N}| > J$, and hence, indicates that a fault has been detected. For this simulation testbed, the threshold is tuned based on the nominal fault-free data to achieve a trade-off between false-detection and missed detection rates. As such, the value of the threshold is initially set to $J = 5$. Of course, the value of the threshold can be easily changed according to the designer's wish.

The engine control system and the residuals generator are then simulated with the faults in Table 2. Residuals that are sensitive to the corresponding faults would trigger and produce non-zero mean values. Figures 9–19 show the simulation results for the 'Original 9' residuals during the WLTP driving cycle profile when the engine system is induced with the faults. Only single-fault scenarios are considered for now. The figures show that due to the dynamics of the engine system and the nature of the faults, that is if they are actuator, sensor, or variable fault type, or that if they are induced as an abrupt fault or an incipient fault, would influence the corresponding residuals that are sensitive to the individual faults to exceed the threshold and to trigger. For example, all sensor faults $\{f_{y_{p_{ic}}}, f_{y_{p_{im}}}, f_{y_{T_{ic}}}, f_{y_{W_{af}}}\}$ triggered only one residual each as they do not directly affect the states of the engine system. See Figures 16–19. However, they could still affect the system indirectly if they are used as feedback signals. As for the actuator and variable faults, since they directly affect the dynamics of the engine system, more residuals are sensitive to such faults. Therefore, if detected, it is usually easier to isolate sensor faults compared to variable faults. By collectively identifying which residuals have triggered for the faults induced, fault isolation analysis to locate the fault in the engine system can then be performed.

During simulations of faults in real-world conditions, especially for non-repetitive driving cycles such as the WLTP, it is interesting to visualize the effects of the faults on the residuals. Figure 20 shows that for long term or permanent faults such as a gradual increase of restricted pressure in the air filter, $f_{p_{af}}$, the residuals might exhibit occasional spikes. These spikes are influenced by the engine dynamics, such as an increase or decrease in the engine torque. This indicates that the amplitude of the faults and the engine dynamics would affect the outlook of the residuals generated and hence, this has to be taken into consideration during the design of the fault diagnosis scheme.

Fault Detection Requirements

The suggested requirements for fault detection are as follows:

- *Time for fault detection*: The decision for fault detection is made based on the amplitude of the residuals, that is if the residuals exceed the threshold J, as well as the duration that they stay above the threshold. For this simulation testbed, the fault should be detected if the duration that the residuals exceed the threshold, $t_f > 3$ s.

- *Missed detections*: This testbed is designed such that the amplitudes of the faults are of sizes that they all should be detected.

Fault Isolation Analysis from Simulations

From the simulation results shown in Figures 9–19, the Fault Sensitivity Matrix (FSM) in Table 4 can be constructed. The FSM is tabulated by placing a value of '1' if the residual is triggered by the specific fault and '0' otherwise. Using the FSM in Table 4, the FIM of the system for the current residuals design can then be constructed. Figure 21 shows the FIM with a more realistic fault isolation performance of the system now that the magnitudes and shapes of the faults acting onto the engine system are also considered. See Figure 6 for comparison. However, the results are not exciting as many faults are not isolable from each other. Therefore, this model would serve as an excellent platform for designers and researchers to design and to perform Model-In-The-Loop tests of fault diagnosis schemes with application to actual automotive engine systems.

The Simulation Environment

Figure 22 shows the GUI of the simulation testbed in Matlab. Through this interface, the user can set the preferences for simulation settings, design, and test their residuals generation and fault diagnosis schemes, as well as view simulation results.

Establishing Simulation Settings

In the left section of the GUI are pop-up menus for the user to establish some key simulation settings. The simulation settings available include

- *Fault Mode*: To induce any of the 11 faults defined in Table 2. A fault-free scenario is also available and is selected by default. As of current development, only single fault scenarios are available.
- *Driving Cycle*: A selection of 4 industrial-standard driving cycles, which are Worldwide harmonized Light vehicles Test Procedure (WLTP), New European Driving Cycle (NEDC), Extra-Urban Driving Cycle (EUDC), and EPA Federal Test Procedure (FTP-75).
- *Simulation Mode*: A choice of 2 simulation modes, which is to either simulate only the engine for the chosen driving cycle, or to extend the simulation to also include generation of the residuals and the execution of the fault diagnosis algorithm. The latter choice would require design and coding inputs from the user.

Design and Testing of Residuals Generation and Fault Diagnosis Schemes

In the top right section of the GUI, a block diagram representation of the engine control system, residuals generator, and fault diagnosis scheme can be found. The user can click on each block to access the corresponding Simulink model or m-file. For example, the user could use the ‘Residuals Generator (Simulink)’, ‘Residuals Generator Design (m-file)’, and ‘Fault Isolation Scheme Design (m-file)’ components to edit their design and codes for the residuals generation and fault diagnosis algorithms. The ‘RUN SIMULATION’ pushbutton starts the simulation while the ‘EXIT’ pushbutton exits the simulation environment and closes the GUI.

Simulation Results

The results obtained from the simulation are displayed in the bottom right section of the GUI. The results displayed are the reference vs actual engine torque, and the normalized plot of the fault induced. A ‘Simulation Log’ is also available in the bottom left section of the GUI to show a summary of the simulation settings and to provide an update in real-time on the progress of the simulation. The plots and the ‘Simulation Log’ are automatically saved into the folder ‘`/Results/`*`DrivingCycle_FaultMode_Date`*’ that is located in the same directory as the simulation files. A Matlab MAT-file containing key variables and data from the simulation is also saved. See Table 5. Depending on the user’s requirements, additional plots can be generated and saved into the same folder using the ‘`SavePlot()`’ command and additional messages can be displayed onto the ‘Simulation Log’ using the ‘`PrintLog()`’ command.

The Simulation Kit

The simulation kit is available as an open-source and can be downloaded through the URL address `https://github.com/nkymark/SimEngineBenchmark`. The simulation kit contains the following key files:

- `main.m` — Main execution file. Run this file to start the GUI.
- `Engine.mdl` — Simulink model of the closed-loop nonlinear engine system shown. Open the model from the GUI using either the ‘Boost Controller (Simulink)’ or ‘Engine System (Simulink)’ blocks.
- `GenerateResiduals.m` — Codes for the residuals generation algorithm to be placed here. Open the file from the GUI using the ‘Residuals Generator Design (m-file)’ button.
- `ResidualsGen.mdl` — Simulink model of the residuals generator. The model is called and run from `GenerateResiduals.m`. The default residuals generated are also filtered and normalized, and with added signal noise. Open the model from the GUI using the ‘Residuals Generator (Simulink)’ block. Replace the ‘Residuals Generator’ in the Simulink

model as desired to accommodate other methods for residuals generation.

- `RunFI.m` — Algorithm for fault diagnosis to be placed here. Open the file from the GUI using the 'Fault Isolation Scheme Design (m-file)' block.

Conclusion

This article has presented a simulation testbed for testing and evaluation of residuals generation and fault diagnosis schemes in a TCSI petrol engine system. It emphasized the key features of the simulation testbed, including 1) a realistic nonlinear model of the engine system compared with the actual physical test bench, 2) the testbed enables researchers to simulate actuator, sensor, and variable faults in various components of the engine system without having to physically modify the engine test bench, 3) researchers are able to compare the performance of their fault diagnosis schemes against the presented Structural Model and FIM benchmark, 4) general simulation and fault settings can be easily configured using the GUI interface, and the testbed can be modified and is customizable to accommodate the different residuals generation as well as fault diagnosis schemes, 5) the simulation kit is available as an open-source and can be downloaded for research and/or teaching purposes. The data generated from the simulation testbed are suitable for the study of both model-based and data-driven fault diagnosis methods. This testbed will serve as an excellent platform to demonstrate the effectiveness in designing, simulating, and analyzing fault diagnostic schemes on automotive systems for the development and comparison of current and future research methods as well as for teaching initiatives.

Future developments of the simulation testbed include the activation of intermittent residuals in order to mimic actual applications where some residuals are turned off during certain driving conditions such as rough terrains or extreme weather so that they do not trigger and cause a false alarm. The addition of faults in other parts of the engine as well as new simulation options such as weather will also be considered.

Some of the interesting research challenges in this field of study include, but certainly not limited to 1) the issue of robustness will always be one of the main and critical problems of any control systems, and since most automakers sell their vehicles all over the world, it is very difficult for one fault diagnosis method to remain robust against a variety of terrains, weather, driving styles, and traffic conditions, 2) with an ever increasing development of the autonomous vehicles, it is important for the systems to be aware of their health and to perform self-diagnosis or self-healing to ensure occupants' lives are protected at all times, 3) a combination of model-based and data-driven methods could enhance fault diagnosis performance, especially when combined with cloud-based technologies where a fleet of vehicles can contribute data to a general pool in the cloud.

Acknowledgment

This research was supported by Volvo Car Corporation in Gothenburg, Sweden.

References

[1] J. Gerler, M. Costin, Xiaowen Fang, Z. Kowalczuk, M. Kunwer, and R. Monajemy, “Model based diagnosis for automotive engines-algorithm development and testing on a production vehicle,” *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 61–69, March 1995.

[2] S. Kher, P. K. Chande, and P. C. Sharma, “Automobile engine fault diagnosis using neural network,” in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, Aug. 2001, pp. 492–495.

[3] M. Nyberg, “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.

[4] Y. L. Murphey, J. A. Crossman, ZhiHang Chen, and J. Cardillo, “Automotive fault diagnosis - part II: a distributed agent diagnostic system,” *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 1076–1098, July 2003.

[5] T. Denton, *Advanced Automotive Fault Diagnosis*. London: Routledge, 2017.

[6] A. E. Goodloe and L. Pike, “Monitoring Distributed Real-Time Systems: A Survey and Future Directions,” *NASA Technical Reports Server*, 2010.

[7] A. Scacchioli, G. Rizzoni, and P. Pisu, “Model-Based Fault Detection and Isolation in Automotive Electrical Systems,” in *Proceedings of the ASME 2006 International Mechanical Engineering Congress and Exposition. Dynamic Systems and Control, Parts A and B. Chicago, Illinois, USA.* ASME, Nov. 2006, pp. 315–324.

[8] P. Weber, S. Gentil, P. Ripoll, and L. Foulloy, “Multiple fault detection and isolation,” in *14th IFAC World Congress*, Beijing, China, 1999, pp. 223–228.

[9] J. C. da Silva, A. Saxena, E. Balaban, and K. Goebel, “A knowledge-based system approach for sensor fault modeling, detection and mitigation,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 10 977–10 989, 2012.

[10] L. Tang, G. J. Kacprzynski, K. Goebel, A. Saxena, B. Saha, and G. Vachtsevanos, “Prognostics-enhanced Automated Contingency Management for advanced autonomous systems,” in *2008 International Conference on Prognostics and Health Management*, Oct. 2008, pp. 1–9.

[11] R. Loureiro, R. Merzouki, and B. O. Bouamama, “Bond Graph Model Based on Structural Diagnosability and Recoverability Analysis: Application to Intelligent Autonomous Vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 986–997, March 2012.

[12] R. Loureiro, S. Benmoussa, Y. Touati, R. Merzouki, and B. Ould Bouamama, “Integration of Fault Diagnosis and Fault-Tolerant Control for Health Monitoring of a Class of MIMO

Intelligent Autonomous Vehicles," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 30–39, Jan. 2014.

[13] R. Isermann, J. Schaffnit, and S. Sinsel, "Hardware-in-the-loop simulation for the design and testing of engine-control systems," *Control Engineering Practice*, vol. 7, no. 5, pp. 643–653, 1999.

[14] K. L. Butler, M. Ehsani, and P. Kamath, "A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1770–1778, Nov. 1999.

[15] D. Assanis, Z. Filipi, S. Gravante, D. Grohnke, X. Gui, L. Louca, G. Rideout, J. Stein, and Y. Wang, "Validation and Use of SIMULINK Integrated, High Fidelity, Engine-ln-Vehicle Simulation of the International Class VI Truck," *SAE Transactions*, vol. 109, pp. 384–399, 2000.

[16] W. Lee, M. Yoon, and M. Sunwoo, "A cost- and time-effective hardware-in-the-loop simulation platform for automotive engine control systems," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 217, no. 1, pp. 41–52, 2003.

[17] M. Yoon, W. Lee, and M. Sunwoo, "Development and implementation of distributed hardware-in-the-loop simulator for automotive engine control systems," *International Journal of Automotive Technology*, vol. 6, no. 2, pp. 107–117, 2005.

[18] J. Kühlwein, J. German, and A. Bandivadekar, "Development of test cycle conversion factors among worldwide light-duty vehicle CO_2 emission standards," *The International Council on Clean Transportation*, Sept. 2014.

[19] Eriksson, L., "Modeling and Control of Turbocharged SI and DI Engines," *Oil & Gas Science and Technology - Rev. IFP*, vol. 62, no. 4, pp. 523–538, 2007.

[20] L. Eriksson and L. Nielsen, *Modeling and Control of Engines and Drivelines*. John Wiley & Sons, 2014.

[21] D. Düştegör, E. Frisk, V. Cocquempot, M. Krysander, and M. Staroswiecki, "Structural analysis of fault isolability in the DAMADICS benchmark," *Control Engineering Practice*, vol. 14, no. 6, pp. 597–608, 2006.

[22] C. Edwards, S. K. Spurgeon, and R. J. Patton, "Sliding mode observers for fault detection and isolation," *Automatica*, vol. 36, no. 4, pp. 541–553, 2000.

[23] K. Y. Ng, C. P. Tan, and D. Oetomo, "Disturbance decoupled fault reconstruction using cascaded sliding mode observers," *Automatica*, vol. 48, no. 5, pp. 794–799, 2012.

[24] S. Simani, C. Fantuzzi, and S. Beghelli, "Diagnosis techniques for sensor faults of industrial processes," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 5, pp. 848–855, Sep. 2000.

[25] T. Kobayashi and D. L. Simon, "Application of a Bank of Kalman Filters for Aircraft

Engine Fault Diagnostics," in *Proceedings of the ASME Turbo Expo 2003, collocated with the 2003 International Joint Power Generation Conference. Volume 1: Turbo Expo 2003. Atlanta, Georgia, USA*. ASME, 2003, pp. 461–470.

[26] M. Darouach, M. Zasadzinski, and M. Hayar, "Reduced-order observer design for descriptor systems with unknown inputs," *IEEE Transactions on Automatic Control*, vol. 41, no. 7, pp. 1068–1072, July 1996.

[27] H. Yang and M. Saif, "Observer design and fault diagnosis for state-retarded dynamical systems," *Automatica*, vol. 34, no. 2, pp. 217–227, 1998.

TABLE 1: The system states, actuators, and sensor measurements of the engine system in Figure 2.

Description	Unit
System states:	
Air filter temperature, T_{af}	K
Air filter pressure, p_{af}	Pa
Compressor temperature, T_c	K
Compressor pressure, p_c	Pa
Intercooler temperature, T_{ic}	K
Intercooler pressure, p_{ic}	Pa
Intake manifold temperature, T_{im}	K
Intake manifold pressure, p_{im}	Pa
Exhaust manifold temperature, T_{em}	K
Exhaust manifold pressure, p_{em}	Pa
Turbine temperature, T_t	K
Turbine pressure, p_t	Pa
Turbine speed, ω_t	rad/s
Actuators:	
Reference engine speed, ω_{eREF}	rad/s
Control input for throttle position area, A_{th}	m^2
Control input for wastegate, u_{wg}	$[0...1]$
Air-fuel ratio, λ_c	$[-]$
Ambient pressure, p_{amb}	Pa
Ambient temperature, T_{amb}	K
Sensors:	
Compressor temperature, T_c	K
Compressor pressure, p_c	Pa
Intercooler temperature, T_{ic}	K
Intercooler pressure, p_{ic}	Pa
Intake manifold temperature, T_{im}	K
Intake manifold pressure, p_{im}	Pa
Air filter mass flow, W_{af}	kg/s
Engine torque, Tq_e	N·m
Exhaust manifold pressure, p_{em}	Pa

TABLE 2: Faults, their descriptions, and types.

Fault	Description	Fault Threshold	Nature of Fault (Active Period)	Severity
$f_{p_{af}}$	Loss of pressure in the air filter	20 kPa pressure drop	Abrupt (From 200 s till T_{DC})	Medium
$f_{C_{vol}}$	Intake valve timing stuck at arbitrary position	Stuck at end or middle positions	Abrupt pulses (Active for 30 s every 150 s)	High
$f_{W_{af}}$	Air leakage between the air filter and the compressor	20% of flow through leakage	Incipient (From 200 s till T_{DC})	Medium
f_{W_c}	Air leakage between the compressor and the intercooler	20% of flow through leakage	Abrupt (From 0.4T_{DC} till T_{DC})	High
$f_{W_{ic}}$	Air leakage between the intercooler and the throttle	20% of flow through leakage	Abrupt (From 0.4T_{DC} till 0.8T_{DC})	High
$f_{W_{th}}$	Air leakage after the throttle in the intake manifold	20% of flow through leakage	Abrupt pulses (Active for 40 s every 200 s)	High
$f_{x_{th}}$	Throttle position actuator error	Fault leading to 20% flow error	Abrupt (From 0.4T_{DC} till T_{DC})	Medium
$f_{y_{W_{af}}}$	Air filter flow sensor fault	20% flow error	Abrupt pulses (Active for 30 s every 150 s)	Low
$f_{y_{p_{im}}}$	Intake manifold pressure sensor fault	20% pressure deviation	Incipient (From 200 s till T_{DC})	Low
$f_{y_{p_{ic}}}$	Intercooler pressure sensor fault	20% pressure deviation	Abrupt pulses (Active for 40 s every 200 s)	Low
$f_{y_{T_{ic}}}$	Intercooler temperature sensor fault	20 K offset	Abrupt pulses (Active for 30 s every 150 s)	Low

*T_{DC} represents the duration of the driving cycle: WLTP (1800 s), NEDC (1220 s), EUDC (400 s), FTP-75 (1874 s).

TABLE 3: Default Residuals (‘Original 9’) for fault detection given the sensors setup in Table 1.

Residual	Description
r_{T_c}	Residual for Compressor Temperature Sensor
r_{p_c}	Residual for Compressor Pressure Sensor
$r_{T_{ic}}$	Residual for Intercooler Temperature Sensor
$r_{p_{ic}}$	Residual for Intercooler Pressure Sensor
$r_{T_{im}}$	Residual for Intake Manifold Temperature Sensor
$r_{p_{im}}$	Residual for Intake Manifold Pressure Sensor
$r_{W_{af}}$	Residual for Air Filter Mass Flow Sensor
r_{Tq_e}	Residual for Engine Torque Sensor
$r_{p_{em}}$	Residual for Exhaust Manifold Pressure Sensor

TABLE 4: The Fault Sensitivity Matrix (FSM) of the ‘Original 9’ residuals.

Residual	$f_{p_{af}}$	$f_{C_{vol}}$	$f_{W_{af}}$	f_{W_c}	$f_{W_{ic}}$	$f_{W_{th}}$	$f_{x_{th}}$	$f_{y_{p_{ic}}}$	$f_{y_{p_{im}}}$	$f_{y_{T_{ic}}}$	$f_{y_{W_{af}}}$
r_{T_c}	1	1	0	1	0	0	0	0	0	0	0
r_{p_c}	1	1	1	1	1	1	0	0	0	0	0
$r_{T_{ic}}$	0	1	0	1	0	0	0	0	0	1	0
$r_{p_{ic}}$	1	1	1	1	1	1	0	1	0	0	0
$r_{T_{im}}$	0	1	0	1	0	1	0	0	0	0	0
$r_{p_{im}}$	1	1	1	1	1	1	0	0	1	0	0
$r_{W_{af}}$	1	1	1	1	1	1	1	0	0	0	1
r_{Tq_e}	1	1	1	1	1	1	0	0	0	0	0
$r_{p_{em}}$	1	1	0	1	0	0	0	0	0	0	0

TABLE 5: The variables saved into the MAT-file after each simulation run. The user could then use these data for further processing and analysis towards the design of the fault diagnosis scheme.

Saved Variable	Description
`omega_eREF_sync`	Reference engine speed, ω_{eREF}
`Tq_eREF_sync`	Reference engine torque, Tq_{eREF}
`inputSig_sync`	5 actuator measurements of the engine $(A_{th}, u_{wg}, \omega_{eREF}, p_{amb}, T_{amb})$
`outputSig_sync`	9 sensor measurements from the engine $(T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}, p_{em}, W_{af}, Tq_e)$
`statesSig_sync`	13 states of the engine $(T_{af}, p_{af}, T_c, p_c, T_{ic}, p_{ic}, T_{im}, p_{im}, T_{em}, p_{em}, T_t, p_t, \omega_t)$
`faultSig_sync`	Normalized data of the faults (the selected induced fault would have non-zero data, except when 'Fault-free' scenario is selected where all faults would have data of value zero)
`residualSig_sync`	Data for all 'Original 9' residuals based on the current sensors setup $(r_{T_c}, r_{p_c}, r_{T_{ic}}, r_{p_{ic}}, r_{T_{im}}, r_{p_{im}}, r_{W_{af}}, r_{Tq_e}, r_{p_{em}})$. Note that these data would only be generated if Simulation Mode 2 is selected.

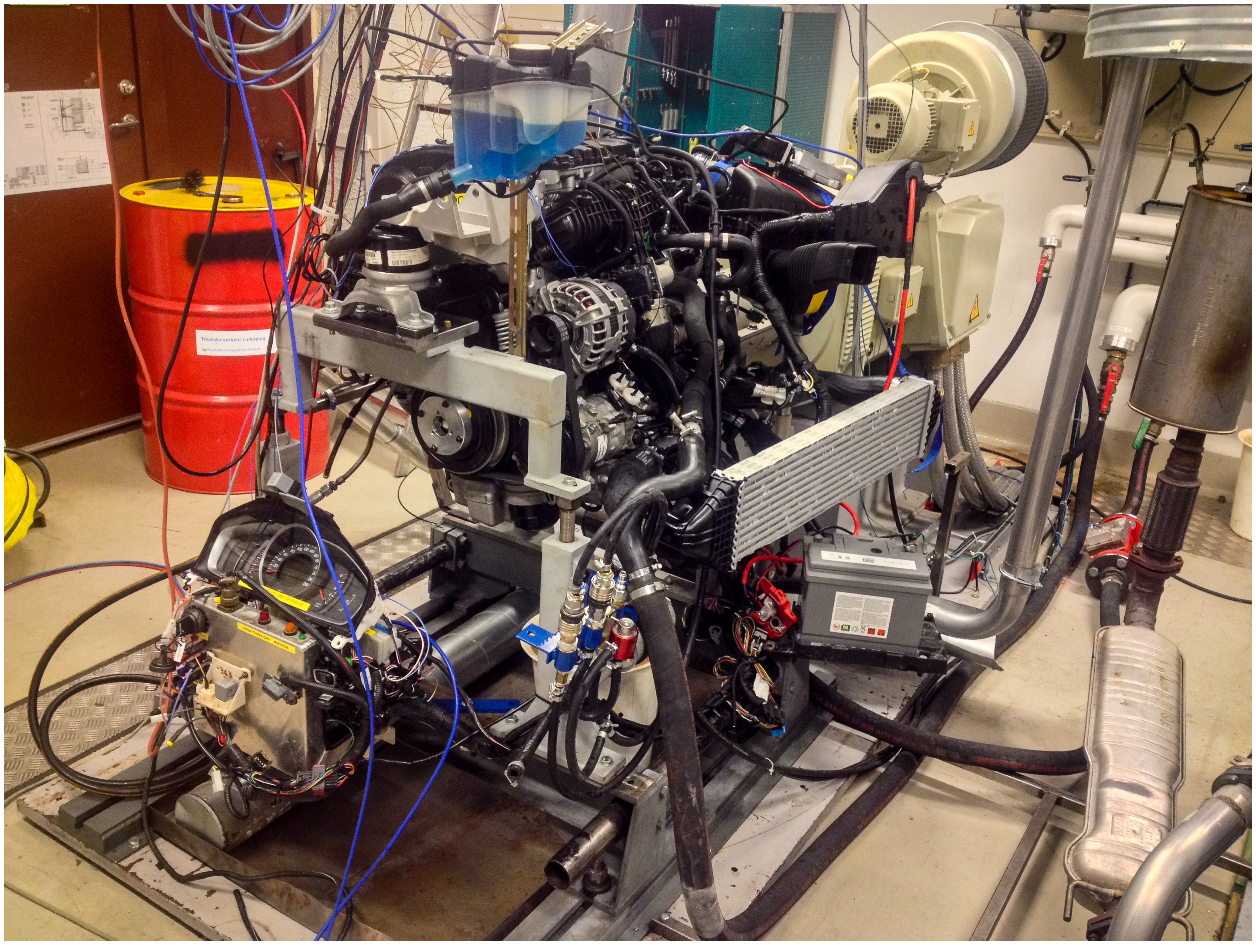


Figure 1: The actual engine test bench used for data collection in the lab.

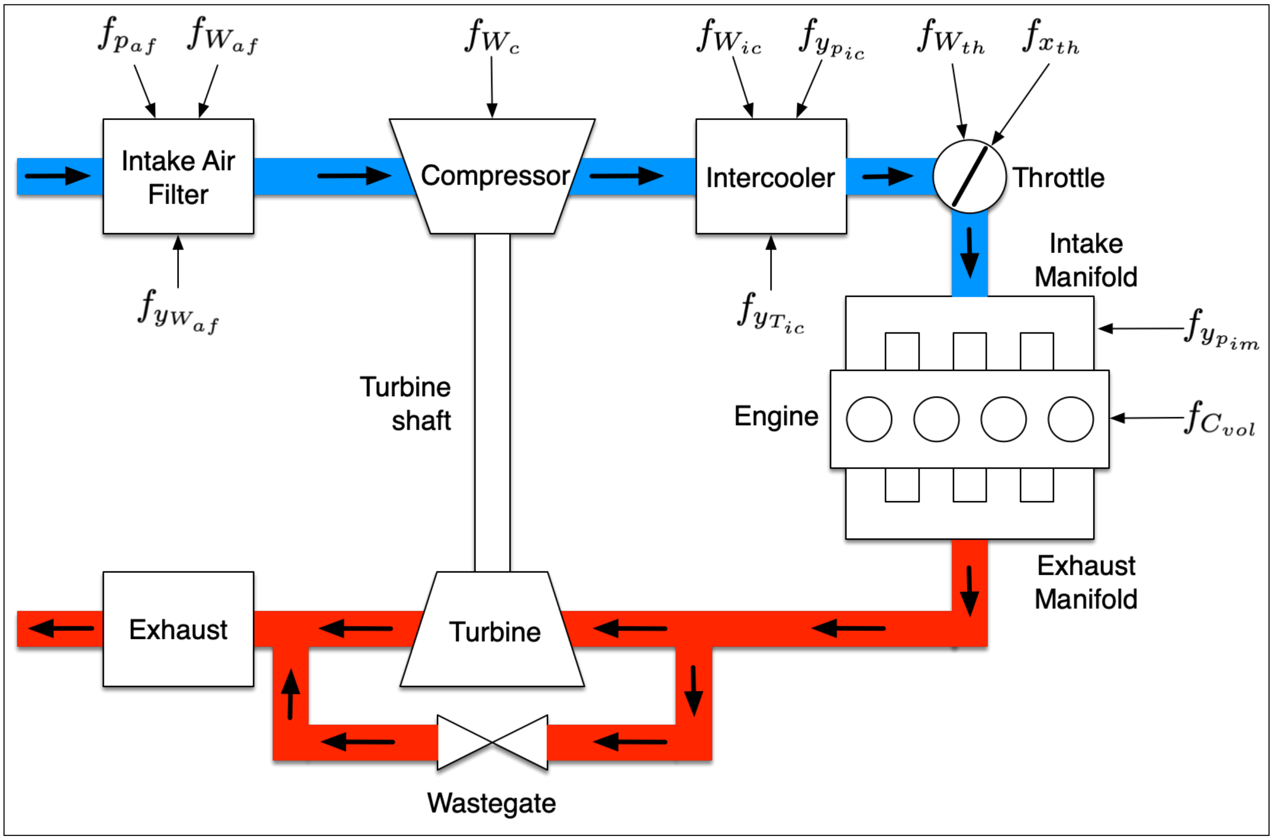


Figure 2: Schematic diagram of the engine system with its subsystems. The blue paths indicate the mass flow of air before entering the engine block while the red paths show the usually much hotter gases produced as a result of the combustion reactions in the engine block. The figure also shows the locations in the engine system where each fault is induced.

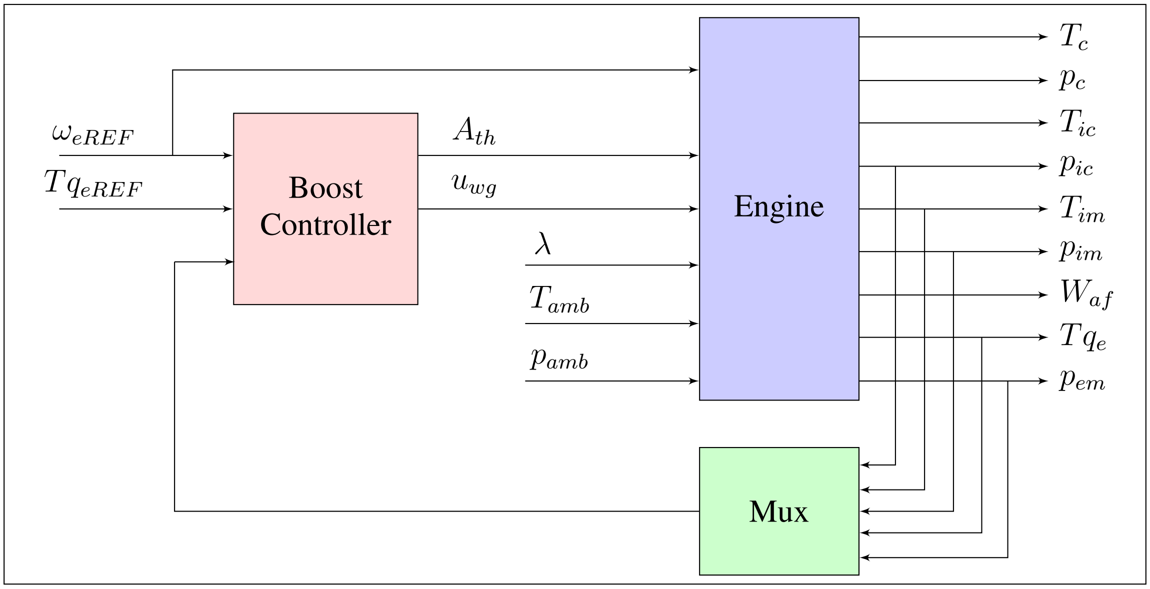


Figure 3: The closed-loop engine control system with the boost controller, actuators, and sensors.

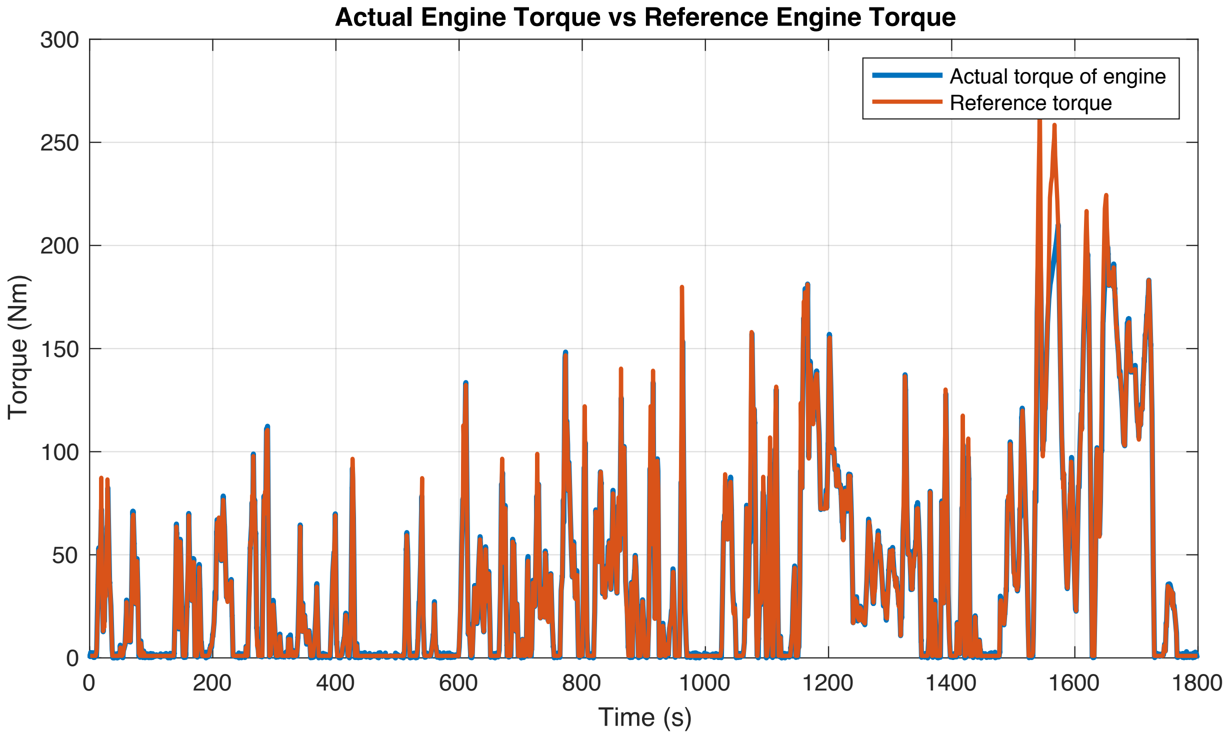


Figure 4: Simulation result showing the engine torque from the model is able to follow its reference during the WLTP driving cycle profile.

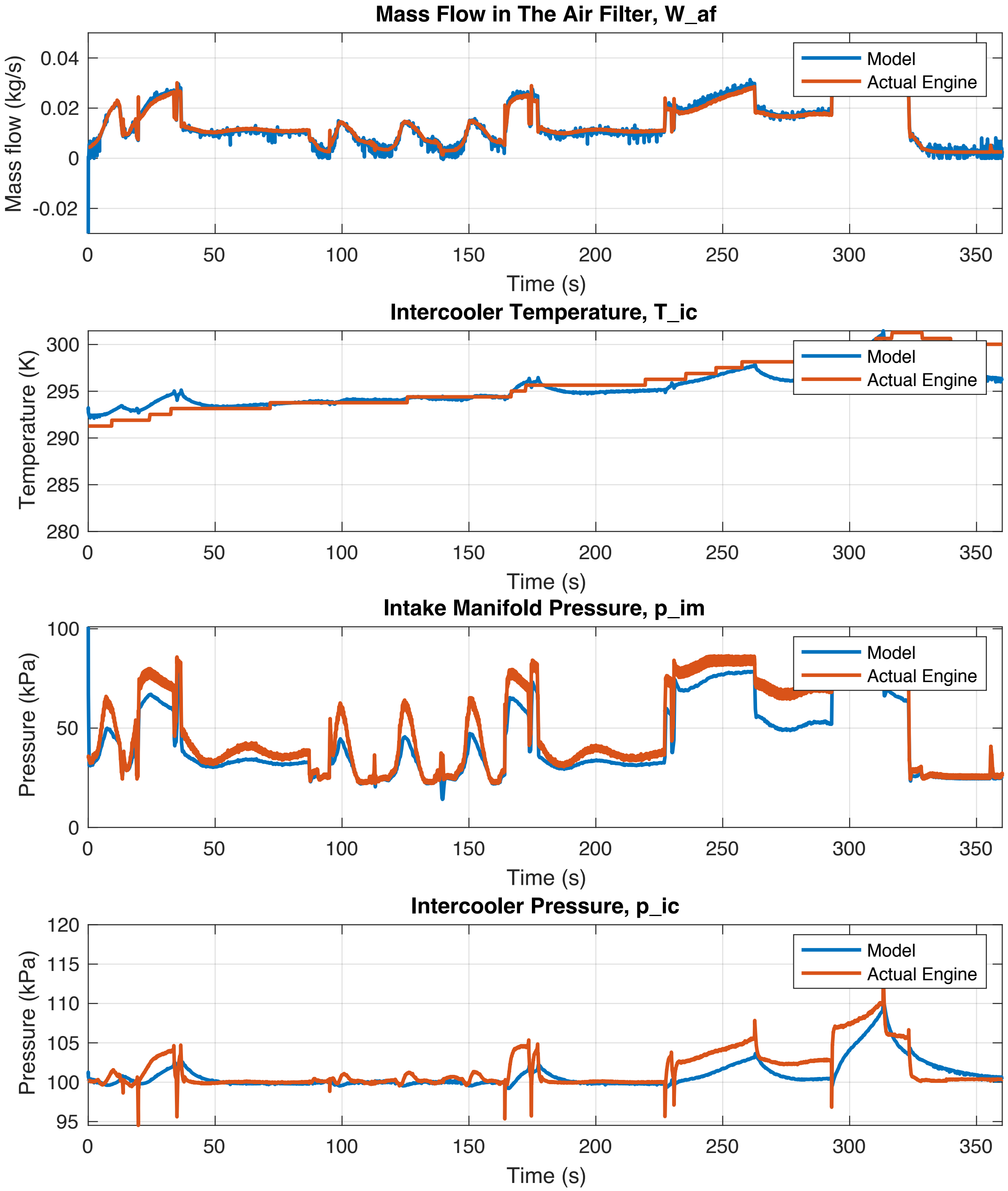


Figure 5: Plots showing sensor signals of the model (blue lines) against the actual sensors of the engine in the lab (red lines) during the EUDC driving cycle profile. The sensor measurements are (from top to bottom) $y_{W_{af}}, y_{T_{ic}}, y_{p_{im}}$, and $y_{p_{ic}}$. The figures show that the engine model is realistic and that it represents the actual engine system accurately.

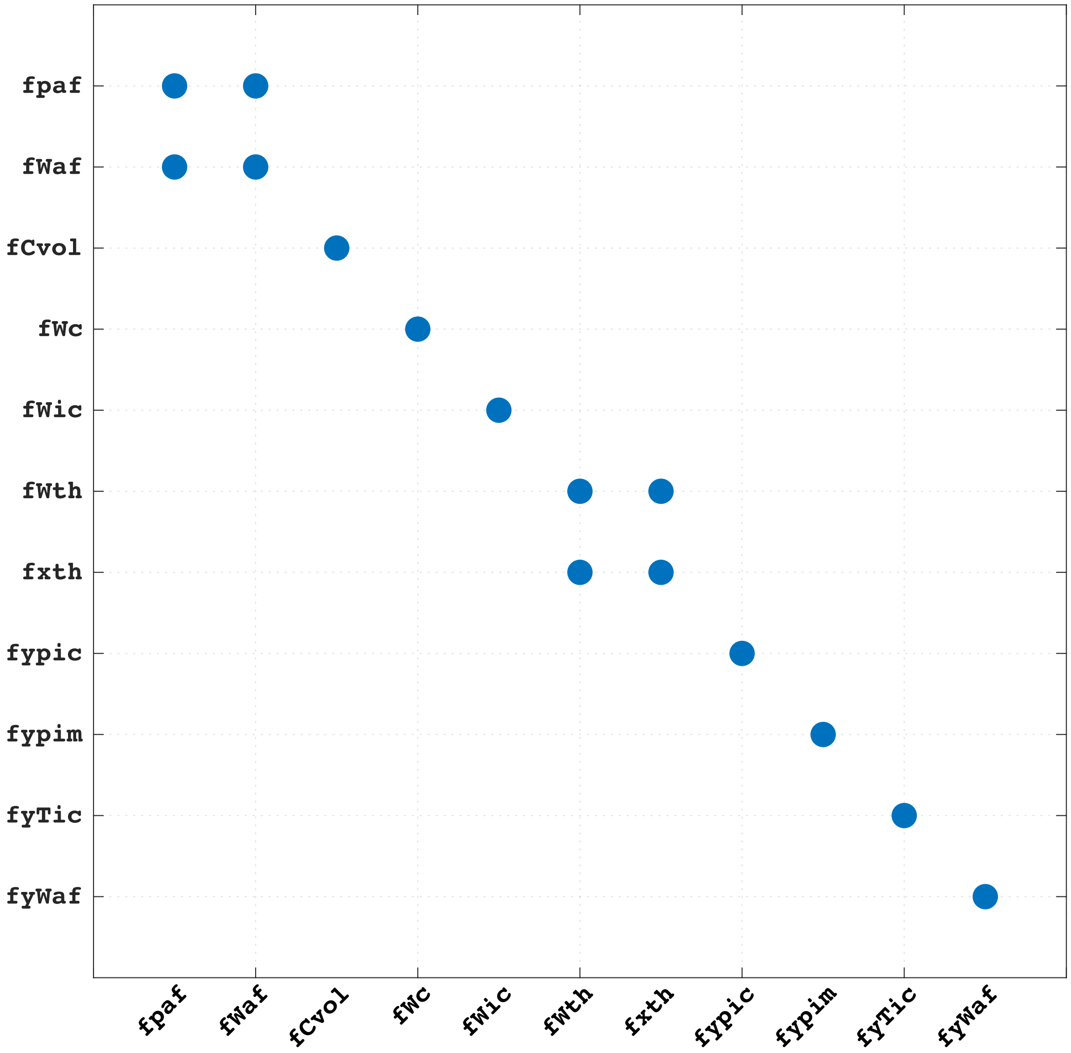


Figure 6: Fault Isolation Matrix (FIM) constructed based on the Structural Model of the engine system. The figure shows that there are 2 pairs of non-isolable faults; $\{f_{p_{af}}, f_{W_{af}}\}$ and $\{f_{W_{th}}, f_{x_{th}}\}$. This is an ideal result for fault isolation assuming no limits to the magnitudes and shapes of the faults.

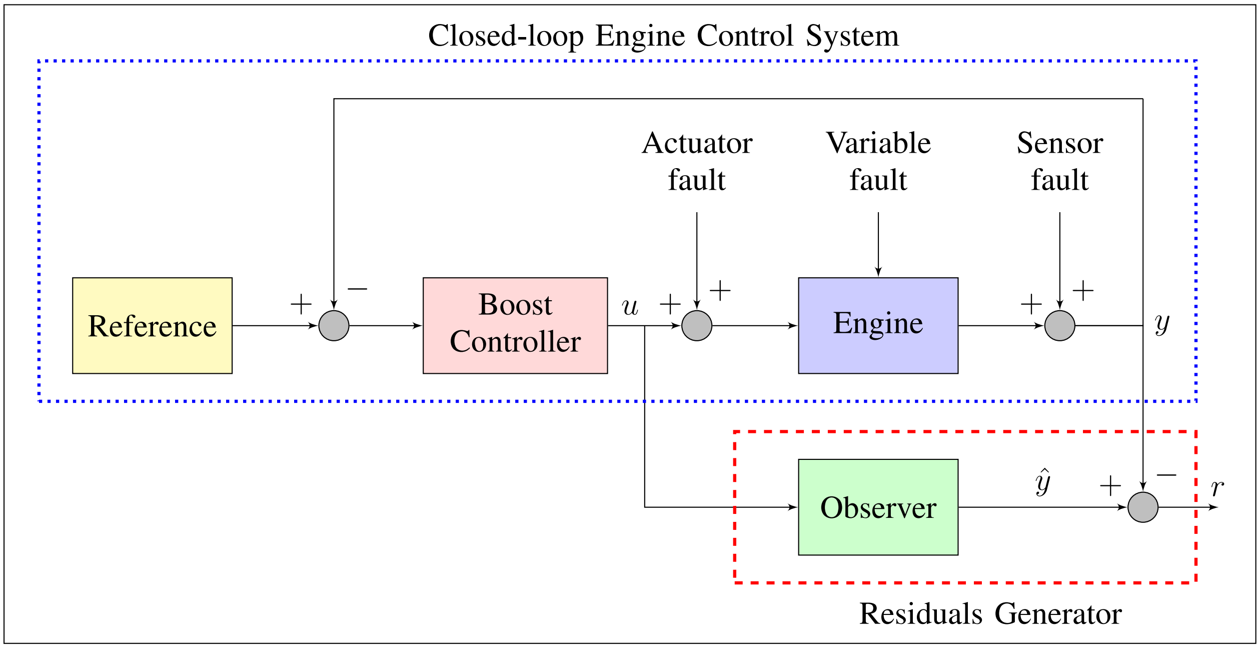


Figure 7: Block diagram representation of the closed-loop engine control system and the residuals generator. The subsystem within the blue dotted box is the closed-loop engine control system shown in Figure 3 while the subsystem within the red dashed box is the residuals generator. This figure also shows the locations where the faults are induced.

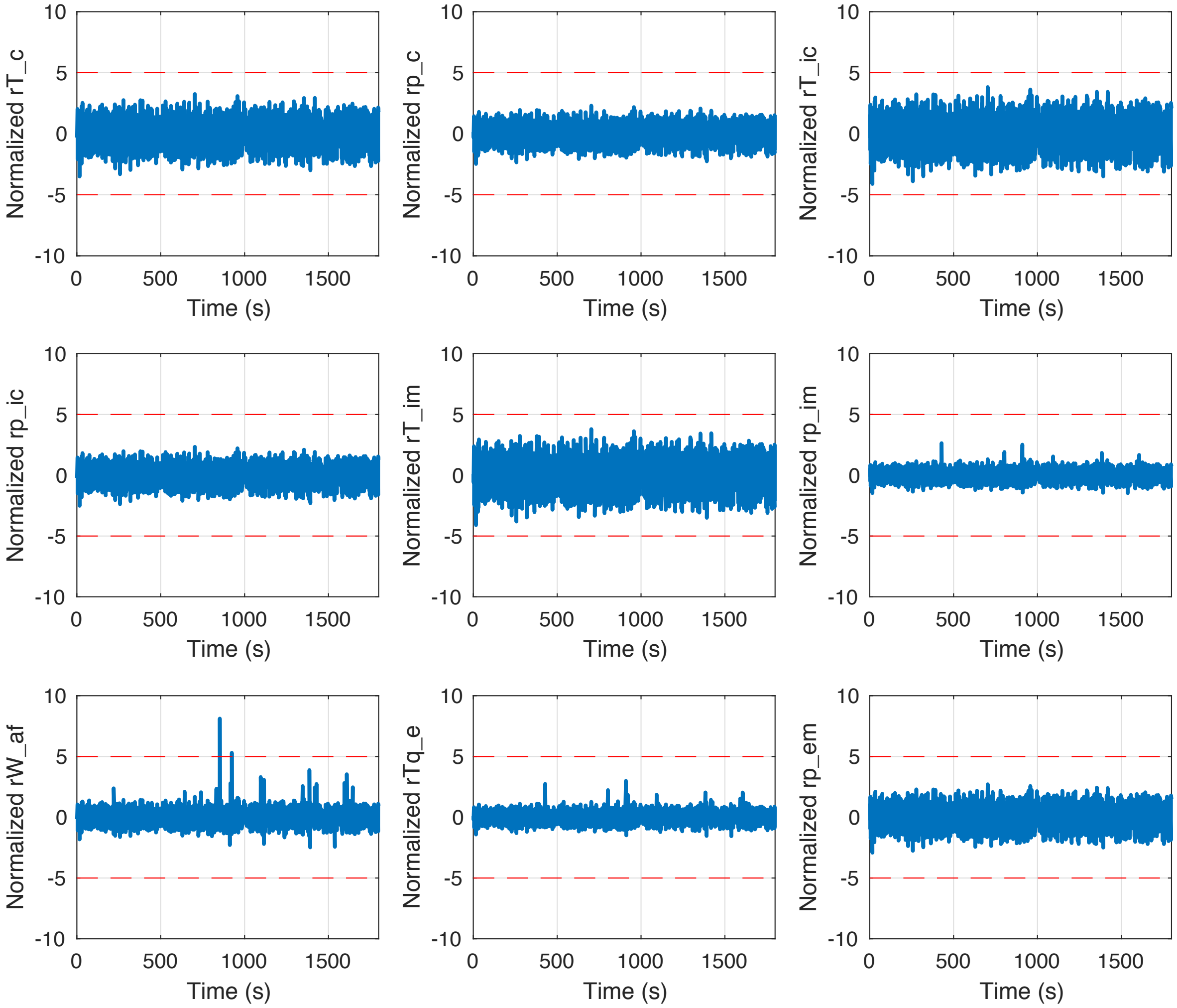


Figure 8: Normalized plots of the ‘Original 9’ for a fault-free scenario. All residuals have zero mean values, which indicate that both the model of the engine and the estimator produce almost identical outputs while being excited by the same control inputs when there are no faults in the engine system. The dashed lines are the fault detection thresholds to determine if the residuals have triggered and that a fault has been detected.

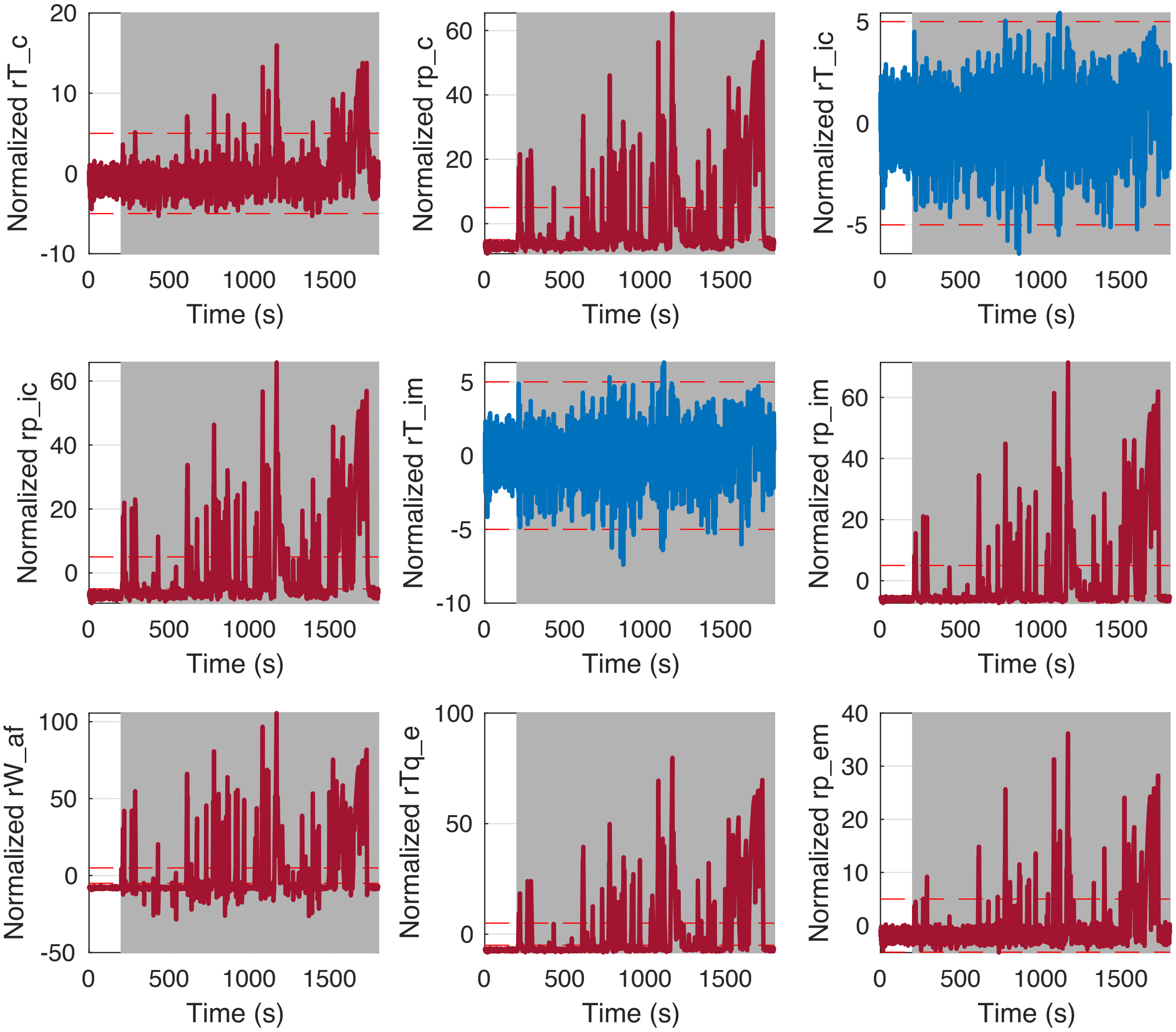


Figure 9: Normalized plots of the 'Original 9' for a loss of pressure in the air filter, $f_{p_{af}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault triggers all residuals except for $r_{T_{ic}}$ and $r_{T_{ic=m}}$. The shaded regions show the duration of which the fault is active.

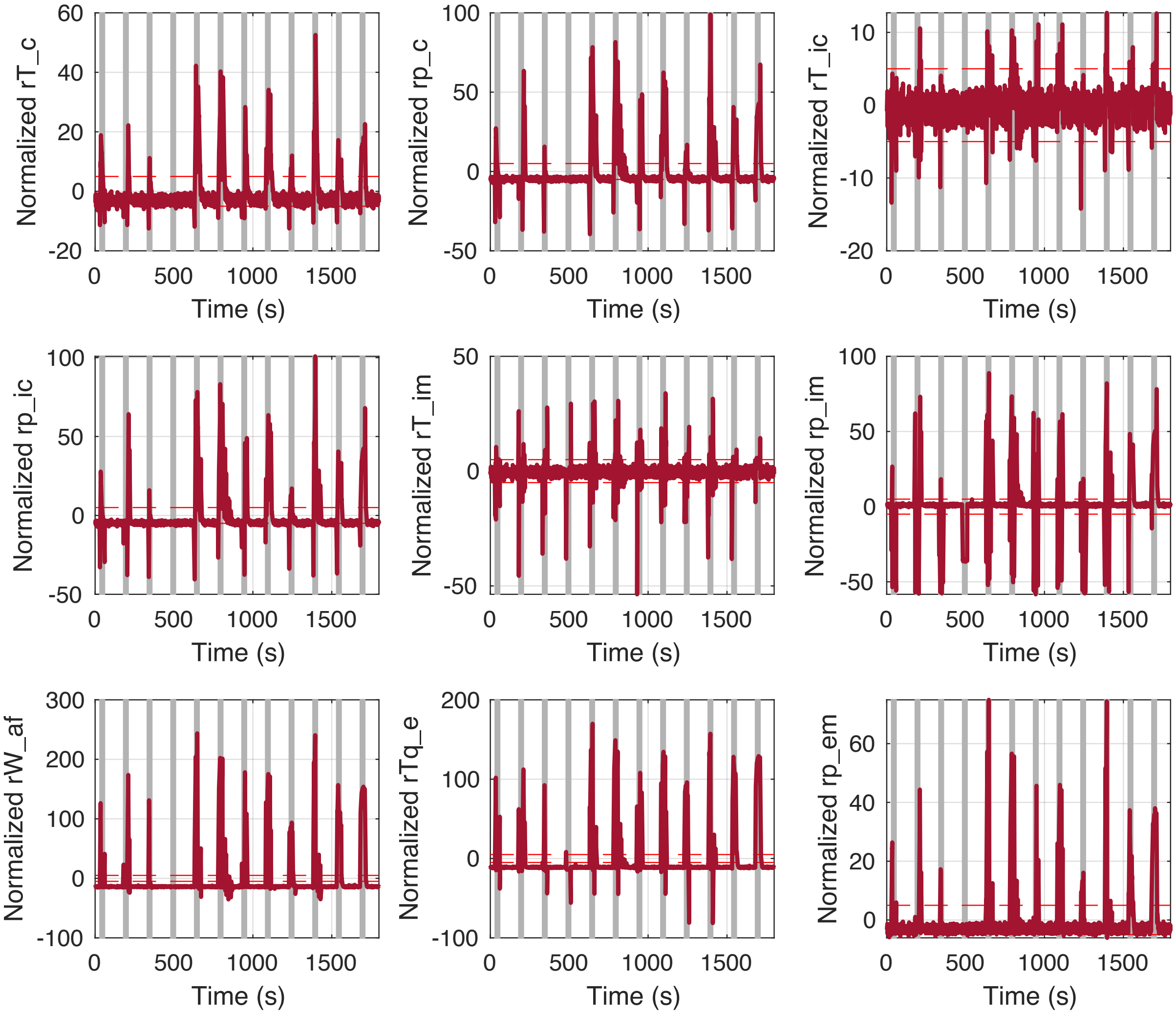


Figure 10: Normalized plots of the ‘Original 9’ for a stuck intake valve timing, $f_{C_{vol}}$. This fault triggered all residuals. The shaded regions show the duration of which the fault is active.

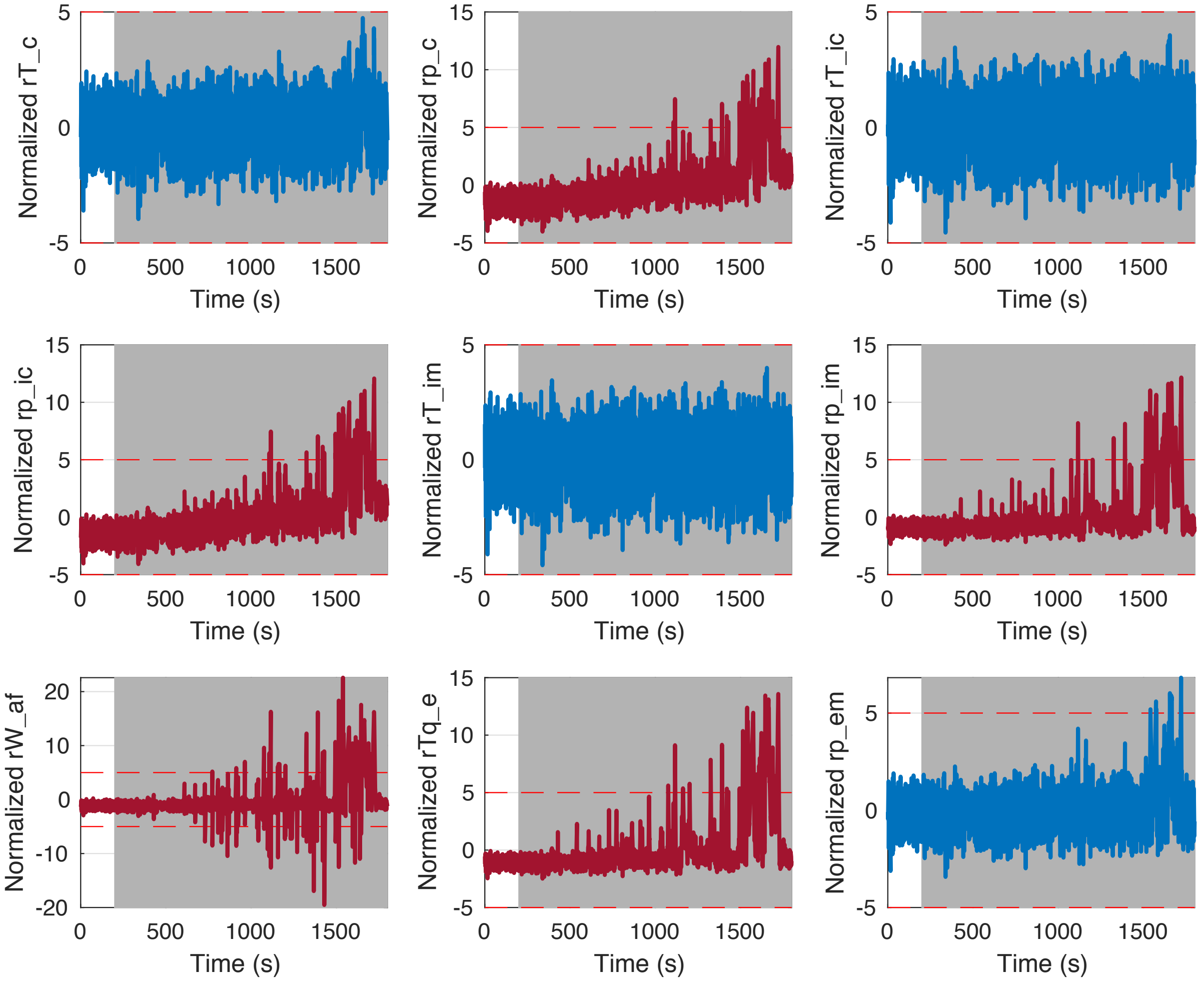


Figure 11: Normalized plots of the ‘Original 9’ for an air leakage between the air filter and the compressor, $f_{W_{af}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault triggered all residuals except for those related to temperature and $r_{p_{em}}$. The shaded regions show the duration of which the fault is active.

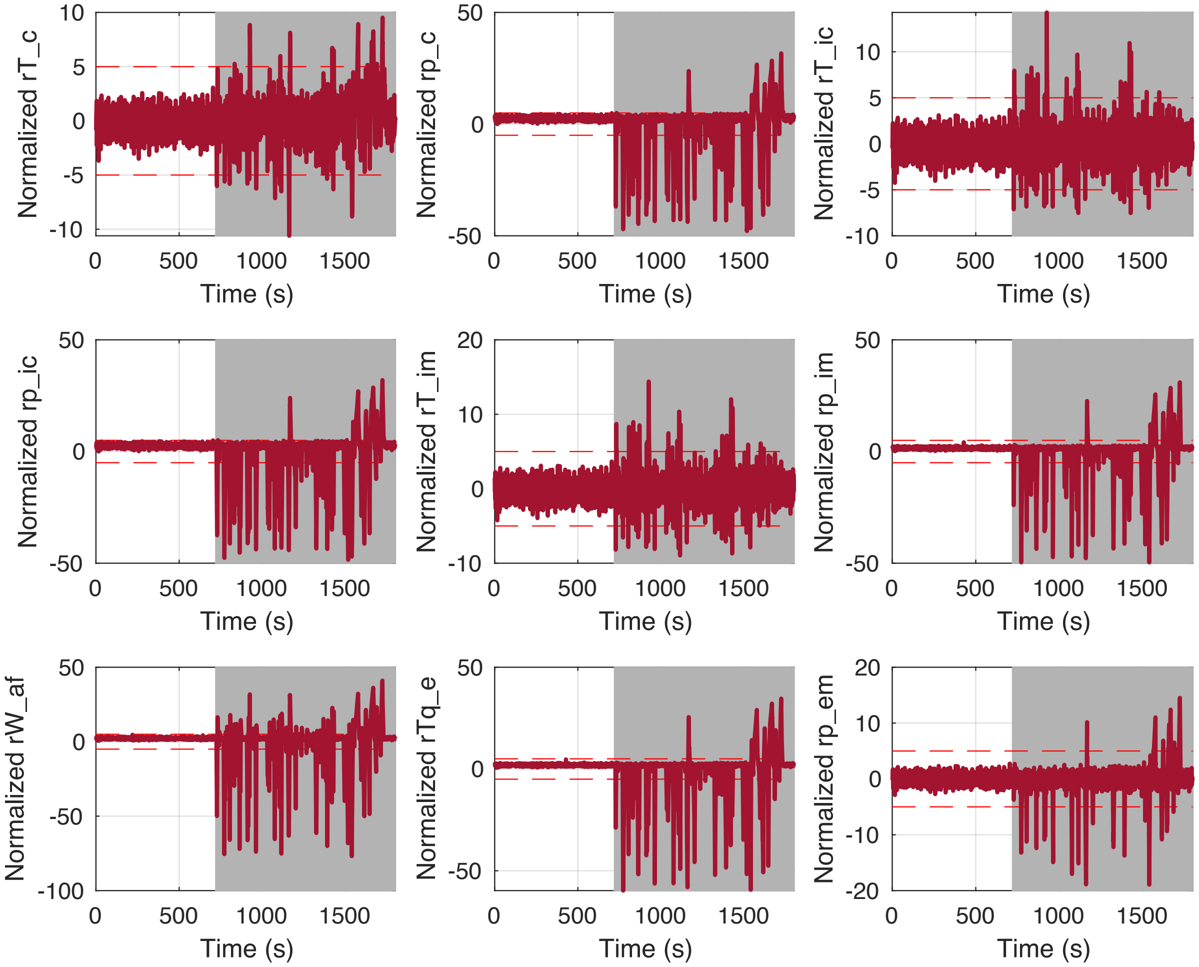


Figure 12: Normalized plots of the ‘Original 9’ for an air leakage between the compressor and the intercooler, f_{W_c}. This fault triggered all residuals. The shaded regions show the duration of which the fault is active.

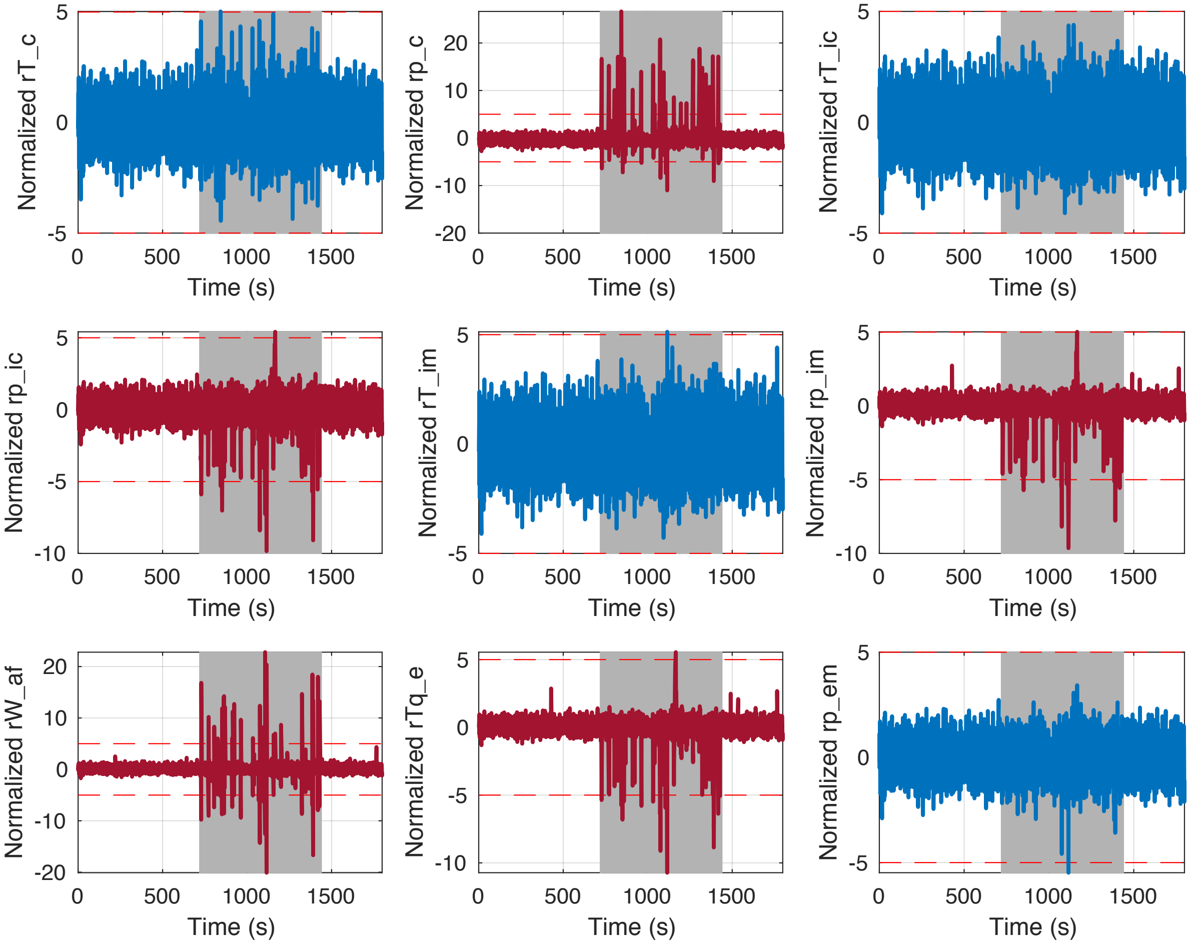


Figure 13: Normalized plots of the ‘Original 9’ for an air leakage between the intercooler and the throttle, $f_{W_{ic}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault triggered all residuals except for those related to temperature and $r_{p_{em}}$. The shaded regions show the duration of which the fault is active.

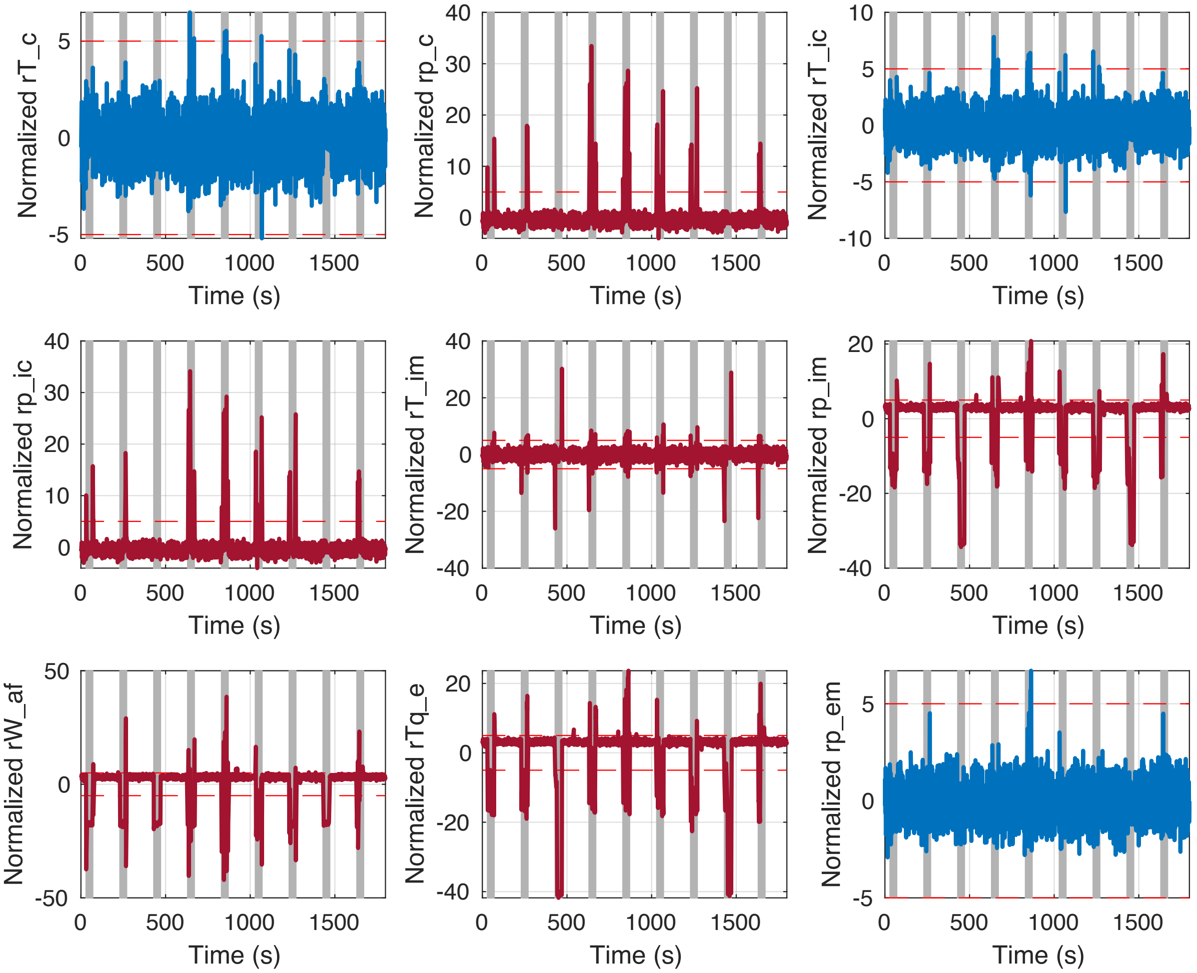


Figure 14: Normalized plots of the 'Original 9' for an air leakage after the throttle in the intake manifold, $f_{W_{th}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault triggered all residuals except for $r_{T_c}, r_{T_{ic}}$, and $r_{p_{em}}$. The shaded regions show the duration of which the fault is active.

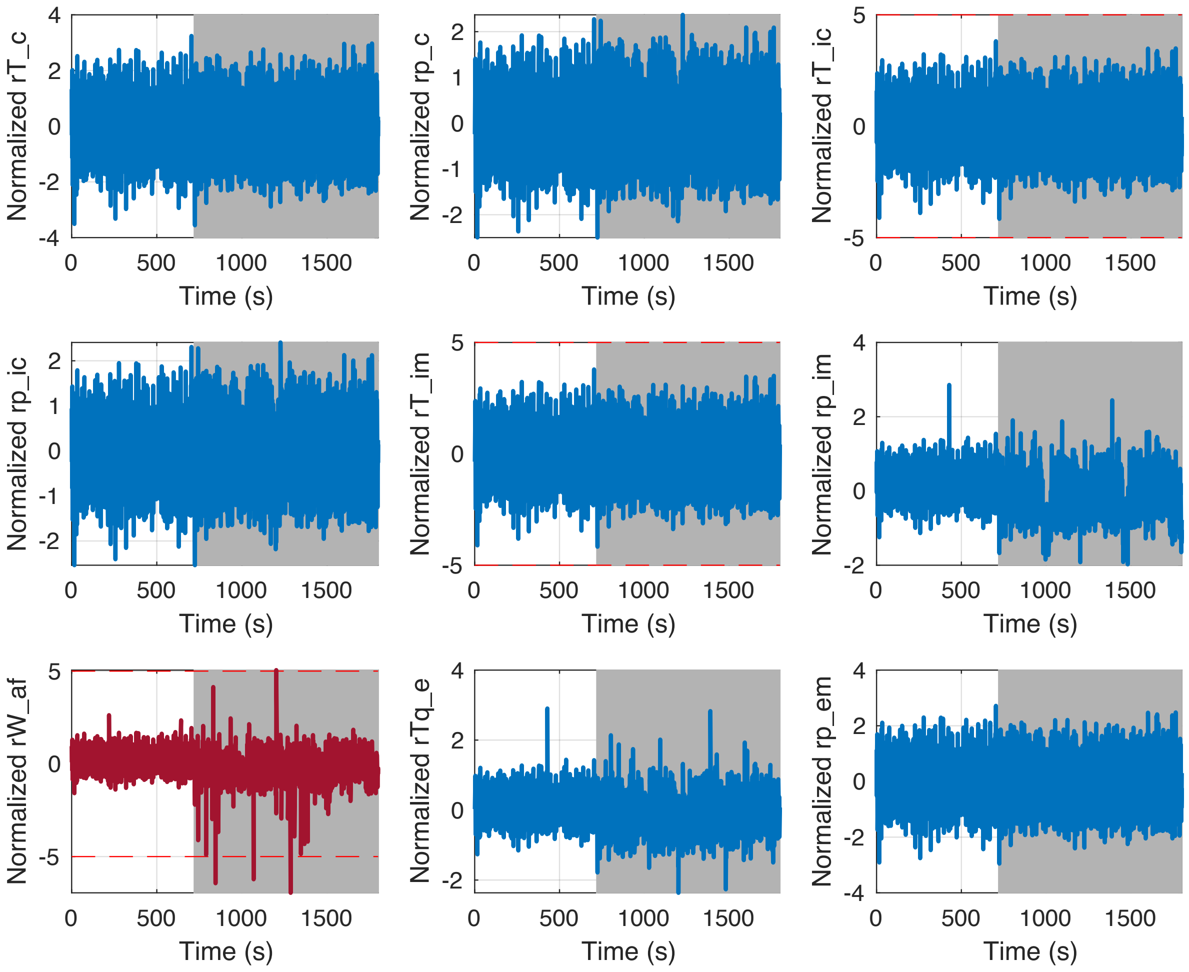


Figure 15: Normalized plots of the ‘Original 9’ for throttle position actuator error, $f_{x_{th}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault only triggered the $r_{W_{af}}$ residual. The shaded regions show the duration of which the fault is active.

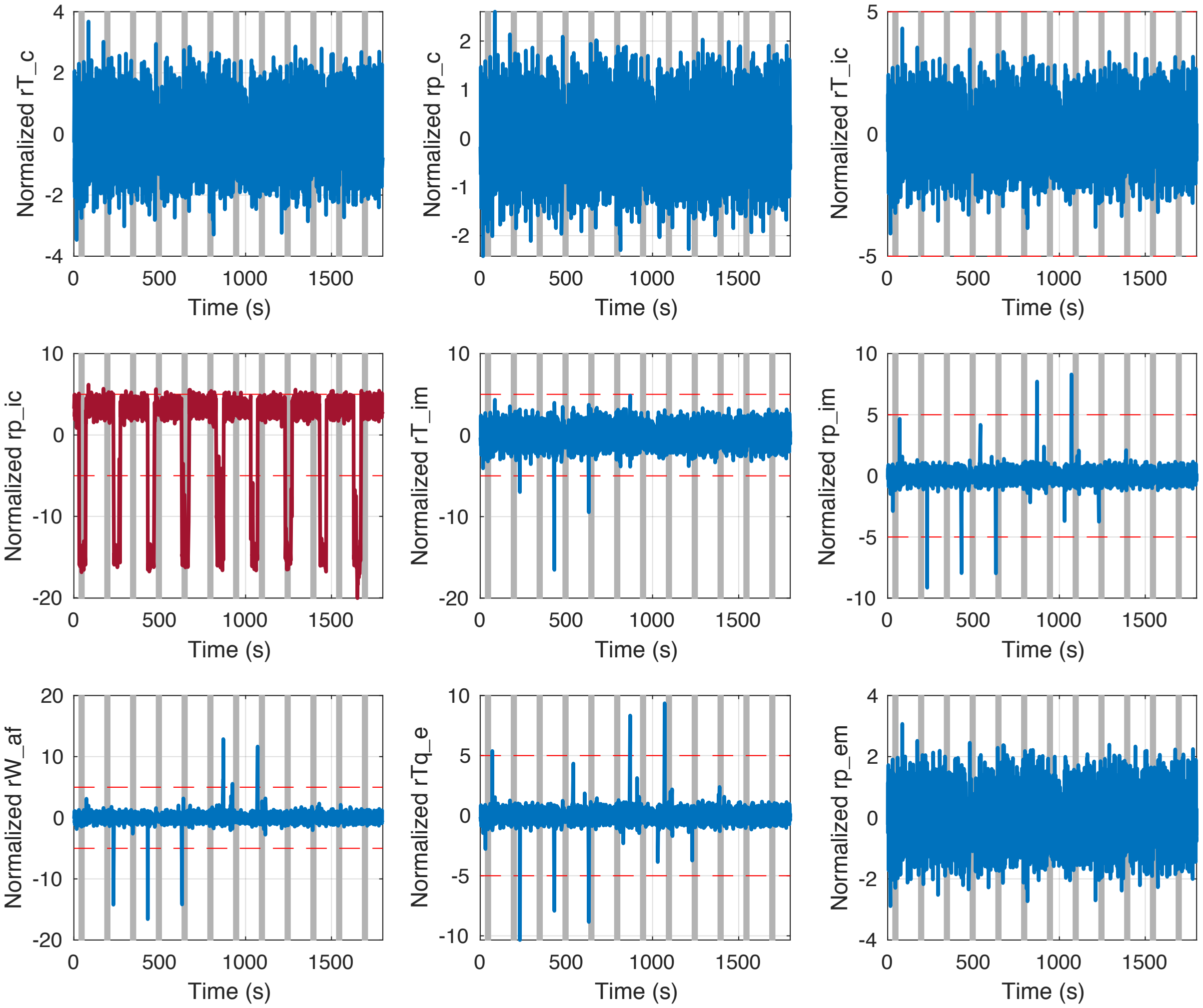


Figure 16: Normalized plots of the ‘Original 9’ for an Intercooler pressure sensor fault, $f_{y_{p_{ic}}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault only triggered the $r_{p_{ic}}$ residual. The shaded regions show the duration of which the fault is active.

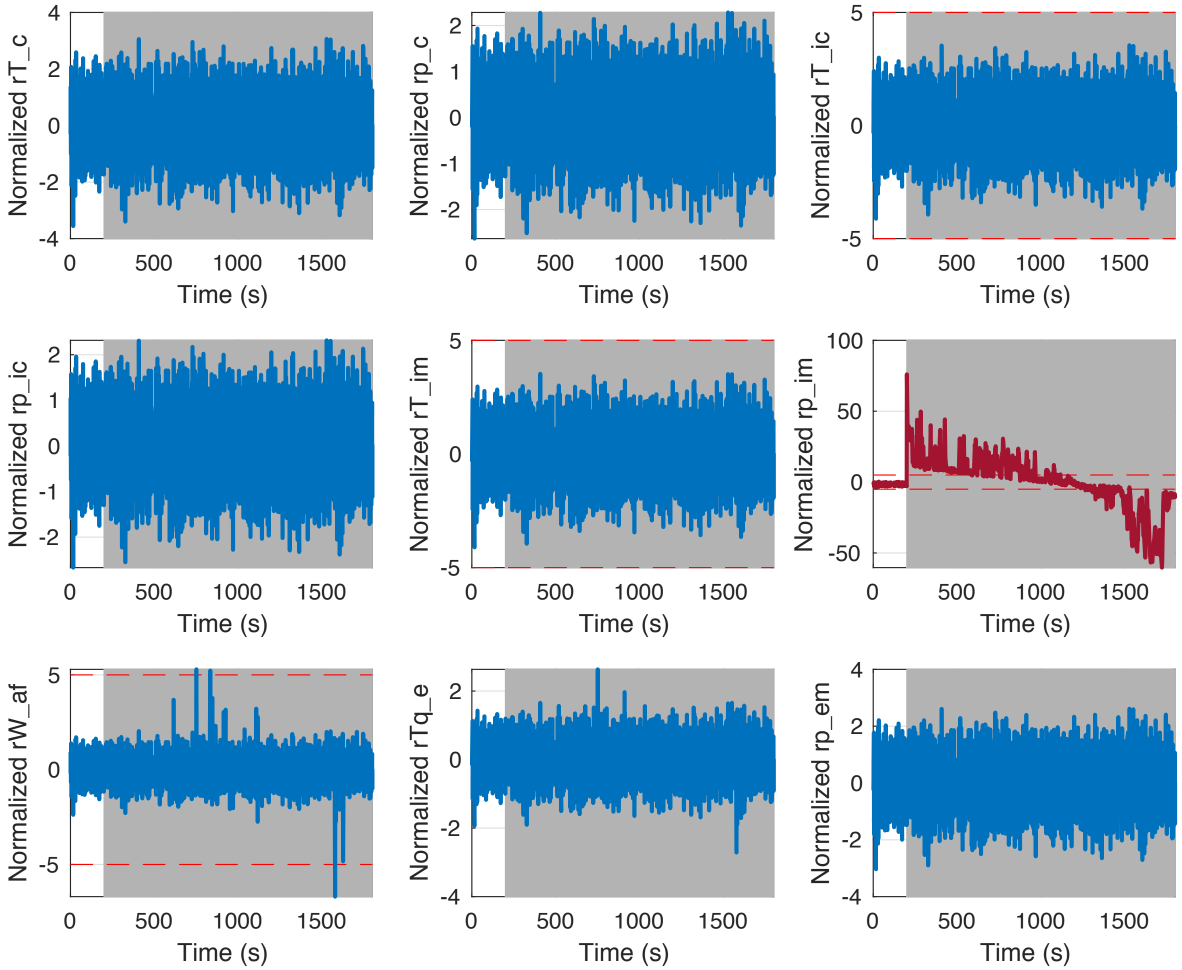


Figure 17: Normalized plots of the 'Original 9' for an intake manifold pressure sensor fault, $f_{y_{p_{im}}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault only triggered the $r_{p_{im}}$ residual. The shaded regions show the duration of which the fault is active.

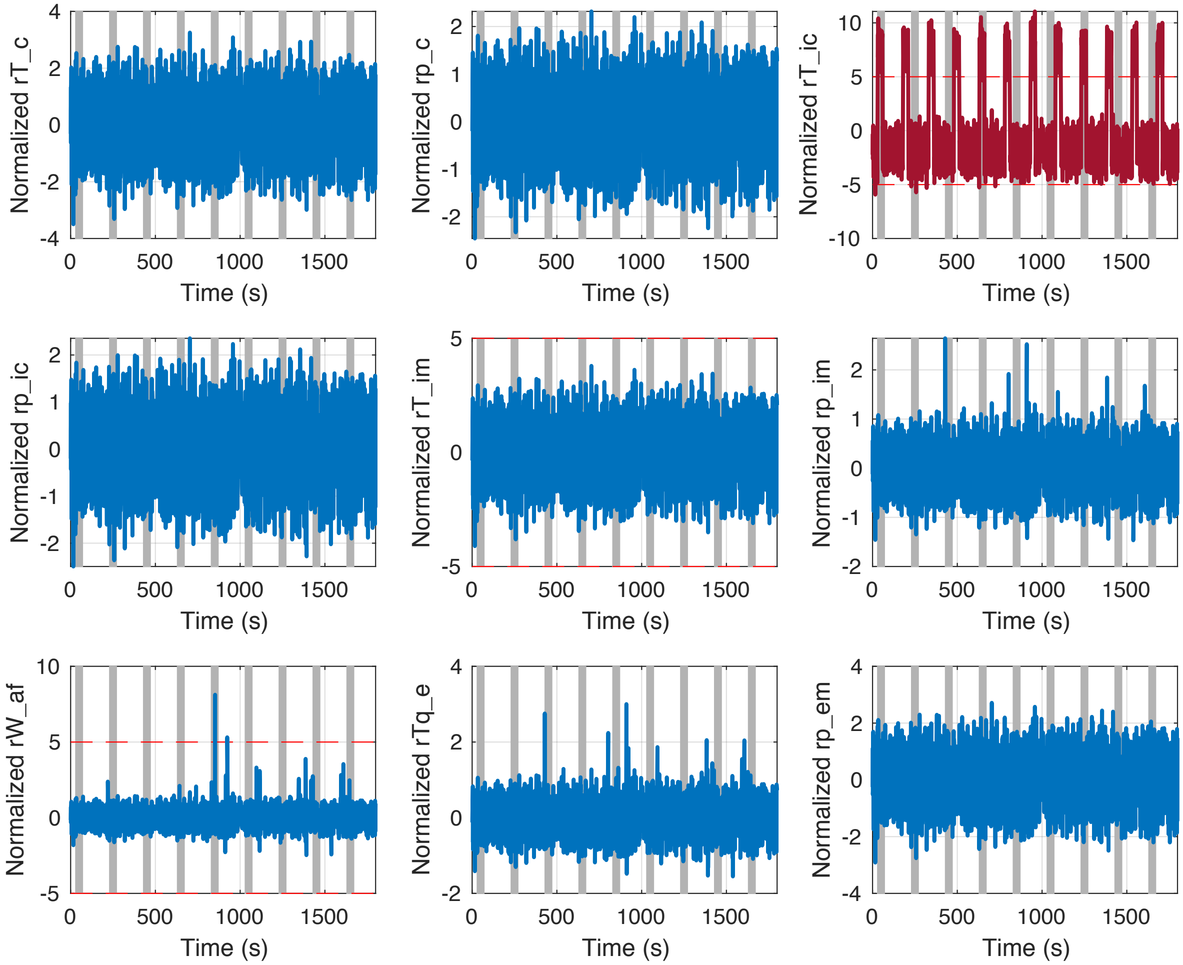


Figure 18: Normalized plots of the 'Original 9' for an intercooler temperature sensor fault, $f_{y_{T_{ic}}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault only triggered the $r_{T_{ic}}$ residual. The shaded regions show the duration of which the fault is active.

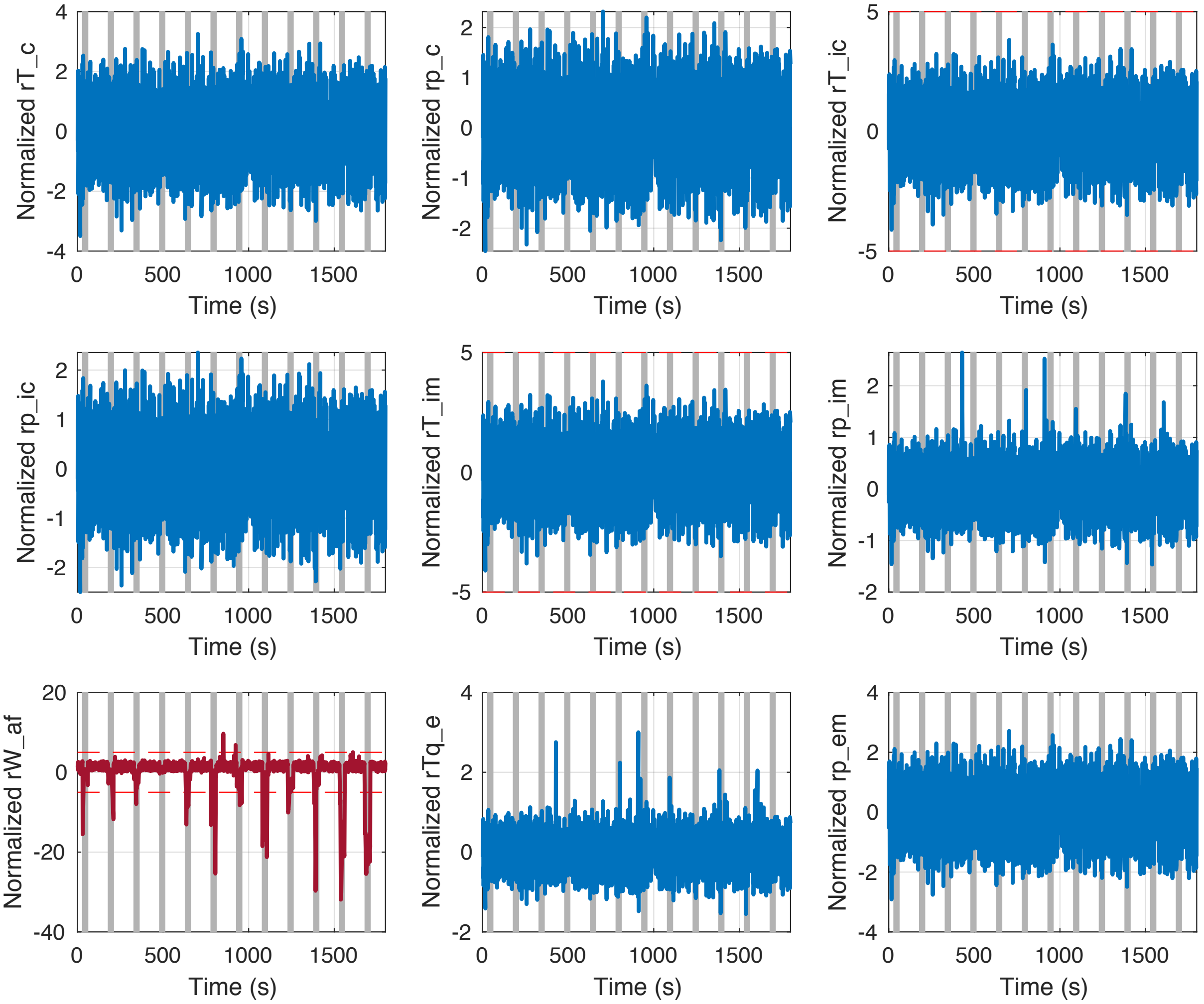


Figure 19: Normalized plots of the 'Original 9' for an air filter flow sensor fault, $f_{y_{W_{af}}}$. Plots in red are the residuals sensitive to the fault and hence triggered, while plots in blue are residuals that are not sensitive to the fault. This fault only triggered the $r_{W_{af}}$ residual. The shaded regions show the duration of which the fault is active.

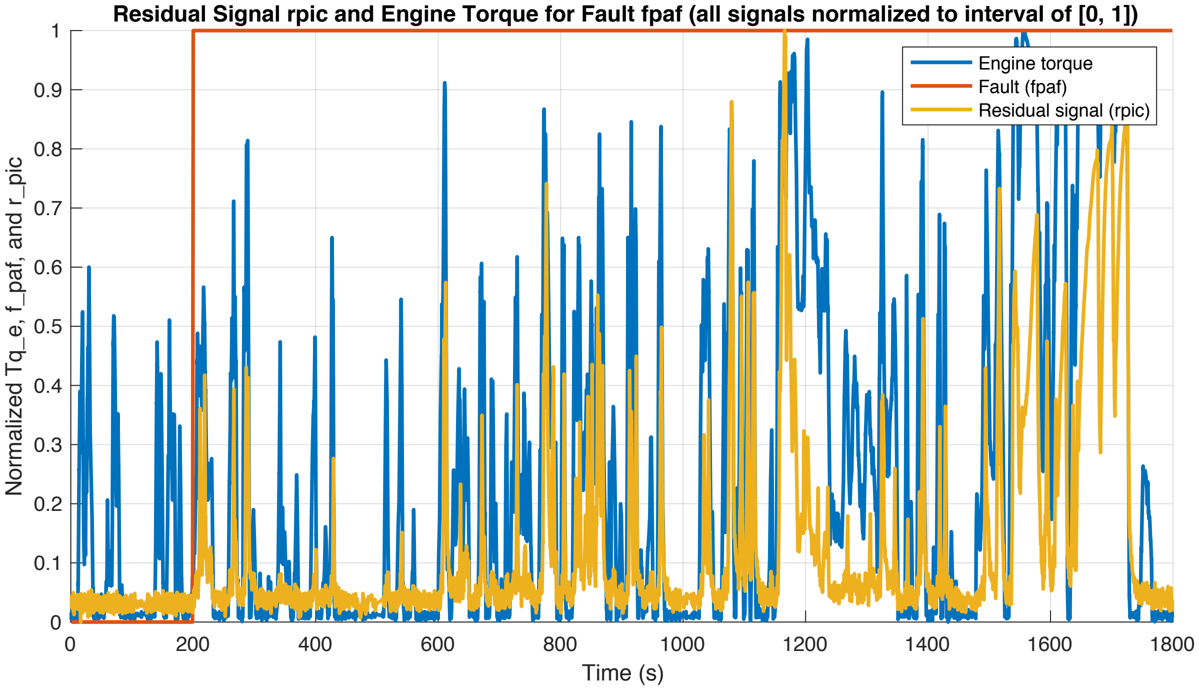


Figure 20: The $r_{p_{ic}}$ residual signal generated (amber line) for a $f_{p_{af}}$ fault (red line) with the engine torque (blue line) during a WLTP driving cycle profile.

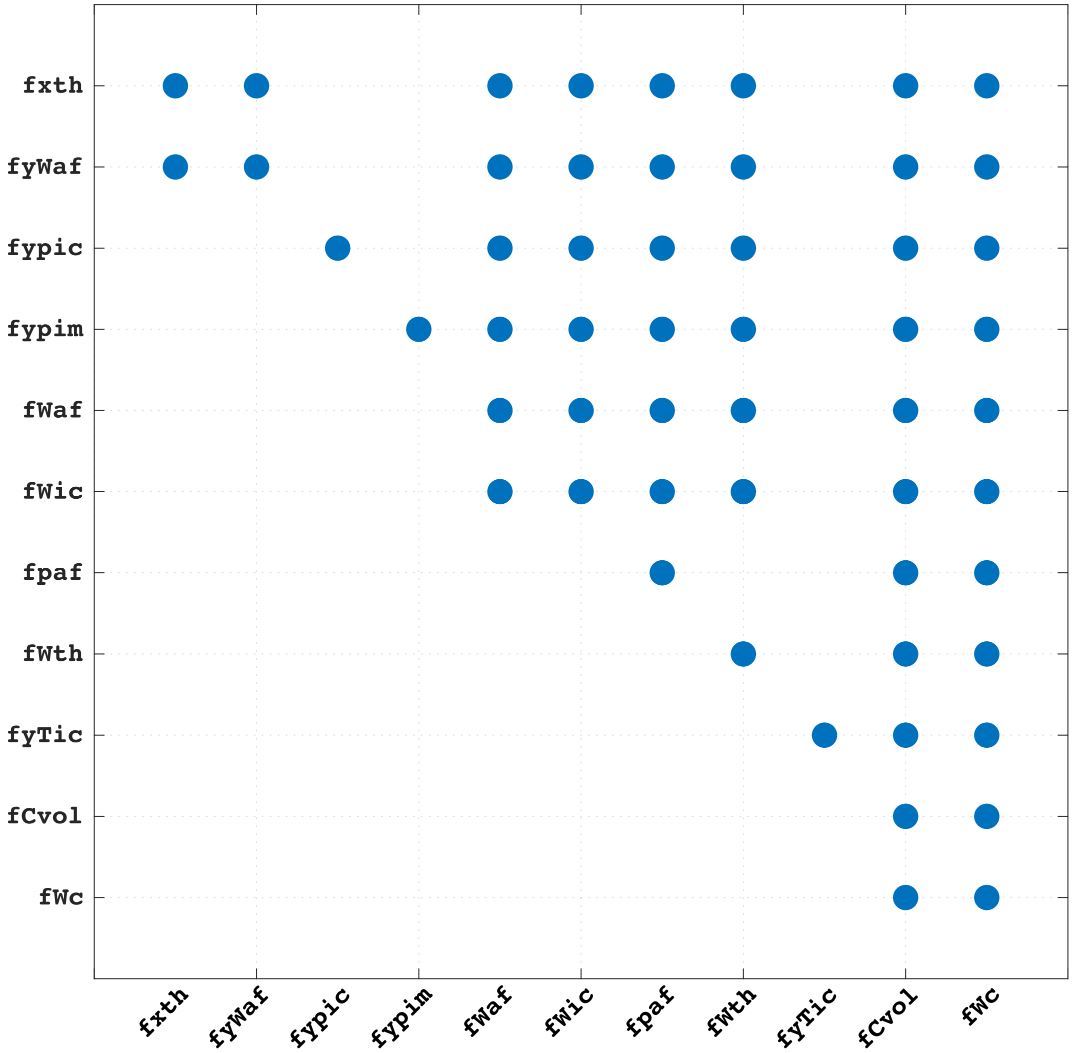
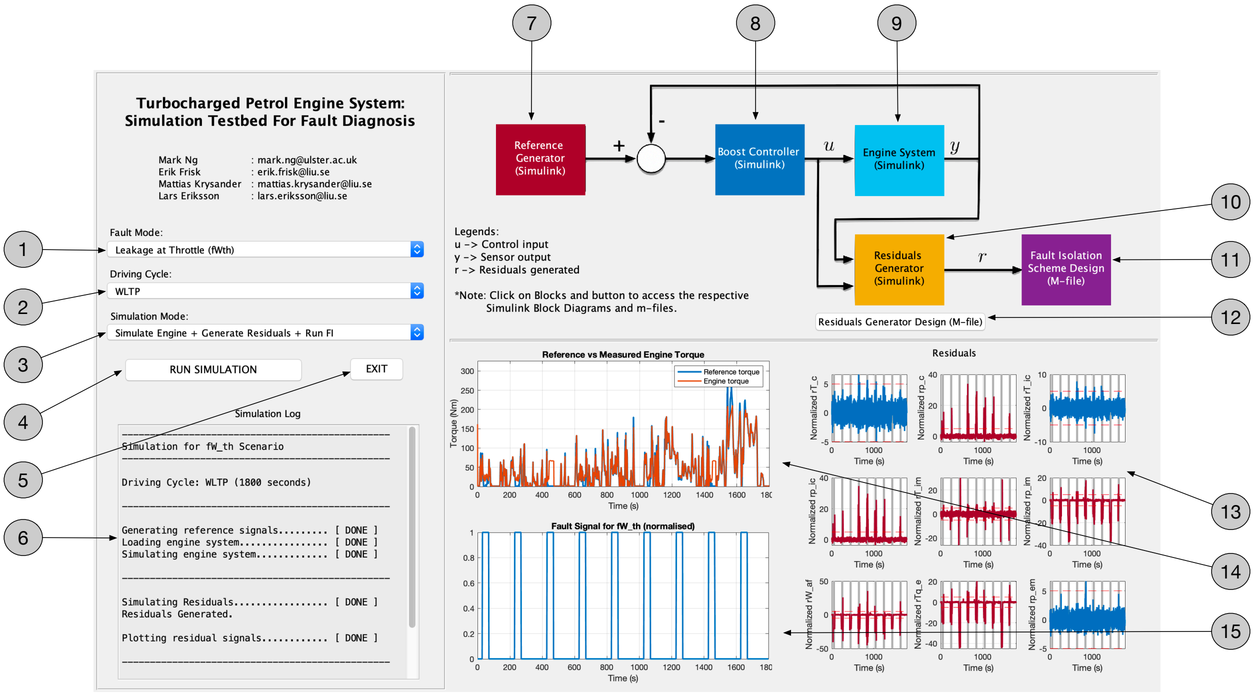


Figure 21: Fault Isolation Matrix (FIM) constructed based on the Fault Sensitivity Matrix (FSM) in Table 4. This is a more realistic representation on the fault isolation analysis as it considers the magnitudes and shapes of the faults.



Turbocharged Petrol Engine System:
Simulation Testbed For Fault Diagnosis

Mark Ng : mark.ng@ulster.ac.uk
Erik Frisk : erik.frisk@liu.se
Mattias Krysander : mattias.krysander@liu.se
Lars Eriksson : lars.eriksson@liu.se

Fault Mode: Leakage at Throttle (fWth)

Driving Cycle: WLTP

Simulation Mode: Simulate Engine + Generate Residuals + Run FI

Legends:
u -> Control input
y -> Sensor output
r -> Residuals generated

*Note: Click on Blocks and button to access the respective
Simulink Block Diagrams and m-files.

Simulation Log

```
--------------------------------------------
Simulation for fW_th Scenario
--------------------------------------------

Driving Cycle: WLTP (1800 seconds)

--------------------------------------------

Generating reference signals......... [ DONE ]
Loading engine system................ [ DONE ]
Simulating engine system............. [ DONE ]

--------------------------------------------

Simulating Residuals................. [ DONE ]
Residuals Generated.

Plotting residual signals............ [ DONE ]

--------------------------------------------
```

Figure 22: The main GUI of the simulation testbed in Matlab; 1) Sets the fault mode for simulation. 2) Sets the driving cycle. 3) Sets the simulation mode. 4) Runs the simulation. 5) Exits and closes the testbed GUI. 6) Shows the simulation progress and log. 7) Click to open the reference generator Simulink model. 8–9) Click to open the boost controller and engine Simulink model. 10) Click to open the residuals generator Simulink model. 11) Click to open the fault diagnosis design scheme m-file. 12) Click to open the residuals generator design scheme m-file. 13) Displays the residuals generated. 14) Displays the reference torque vs actual torque of the engine. 15) Displays the fault signal induced (normalized).

Sidebar: Summary

Research on fault diagnosis on highly nonlinear dynamic systems such as the engine of a vehicle have garnered huge interest in recent years, especially with the automotive industry heading towards driverless technologies and big data. This article presents a novel simulation testbed of a single turbocharged spark ignited (TCSI) petrol engine system for testing and evaluation of residuals generation and fault diagnosis methods.

Designed and developed using Matlab/Simulink, the user interacts with the testbed using a GUI interface where the engine can be realistically simulated using industrial-standard driving cycles such as the WLTP, NEDC, EUDC, and FTP-75. The engine is modeled using mean value engine model (MVEM) and is controlled using a PI-based boost controller. The GUI interface also allows the user to inject one of the 11 faults of interest into the engine so that their effects onto the performance of the engine could be understood better without having to physically induce the faults on the actual engine. Hence, this minimizes the risk of shortening its lifespan or causing permanent damages to the engine.

This simulation testbed will serve as an excellent platform where researchers can generate critical data to develop and to compare current and future research methods for fault diagnosis of automotive engine systems. We also hope that academics in the field are able to benefit from this simulation testbed to encourage scholarly collaborations or used as a virtual laboratory for teaching purposes.

The simulation kit is available as an open-source and can be obtained from https://github.com/nkymark/SimEngineBenchmark.

Sidebar: Vehicle Parameters

Tables S1 and S2 show the key vehicle parameters as well as the estimated gear shift points of the gearbox used to generate the reference inputs for the engine speed, ω_{eREF}, and the engine torque, Tq_{eREF} in “Generation of Reference Inputs and Controller Design”.

TABLE S1: Key vehicle parameters.

Description	Value	Unit
General vehicle parameters:		
Mass of vehicle, m_v	1700	kg
Drag coefficient, c_d	0.29	[–]
Roll coefficient, c_r	0.013	[–]
Frontal area, A_f	2.28	m^2
Wheel radius, r_w	0.3234	m
(assuming tires rated 215/50R17)		
Gear ratios:		
1st	5.250	[–]
2nd	3.029	[–]
3rd	1.950	[–]
4th	1.457	[–]
5th	1.221	[–]
6th	1.000	[–]
7th	0.809	[–]
8th	0.673	[–]
Reverse	4.015	[–]
Final Drive	2.774	[–]
*Speed per 1000 rpm in 8th gear is 62.9 km/h		

TABLE S2: Estimated gear shift points of the gearbox.

Gear	Vehicle Speed [km/h] per 1000 rpm	Shifting Engine Speed [rpm]
1st	8.070	2800
2nd	14.00	2700
3rd	21.70	2600
4th	29.00	2400
5th	34.70	2200
6th	42.30	2000
7th	52.34	1800
8th	62.90	1600

Sidebar: Differential Equations of The TCSI Engine System and Corresponding Engine Parameters

The testbed is modeled based on the actual engine test bench in the lab as shown in Figure 1. The total model equations of the TCSI engine are stated below. The first 47 equations are the TCSI engine as derived in [19], [20]. The remaining 15 equations; $e_{48} \ldots e_{61}$, describe the considered sensors and actuators as shown in Table 1. The faults introduced in Table 2 can be found in equations $e_{15}, e_{17}, e_{23}, e_{25}, e_{29}, e_{50}, e_{55}, e_{56}, e_{58}$, and e_{59}, respectively. Table S3 shows the variables in the engine model and Table S4 shows the corresponding key engine parameters.

$$
\begin{aligned}
&e_1 : \dot{T}_{af} = \frac{R_a T_{af}}{p_{af} V_{af} c_{vi}}((R_a + c_{vi}) W_{af} T_{af,in} - (R_a + c_{vi}) W_c T_{af} - (W_{af} - W_c) c_{vi} T_{af}),\\
&e_2 : \dot{p}_{af} = \frac{R_a T_{af}}{V_{af}}(W_{af} - W_c) + \frac{p_{af}}{T_{af}}\dot{T}_{af},\\
&e_3 : \dot{T}_c = \frac{R_a T_c}{p_c V_{ic} c_{vi}}((R_a + c_{vi}) W_c T_{c,in} - (R_a + c_{vi}) W_{ic} T_c - (W_c - W_{ic}) c_{vi} T_c),\\
&e_4 : \dot{p}_c = \frac{R_a T_c}{V_{ic}}(W_c - W_{ic}) + \frac{p_c}{T_c}\dot{T}_c,\\
&e_5 : \dot{T}_{ic} = \frac{R_a T_{ic}}{p_{ic} V_{ic} c_{vi}}((R_a + c_{vi}) W_{ic} T_{ic,in} - (R_a + c_{vi}) W_{th} T_{ic} - (W_{ic} - W_{th}) c_{vi} T_{ic}),\\
&e_6 : \dot{p}_{ic} = \frac{R_a T_{ic}}{V_{ic}}(W_{ic} - W_{th}) + \frac{p_{ic}}{T_{ic}}\dot{T}_{ic},\\
&e_7 : \dot{T}_{im} = \frac{R_a T_{im}}{p_{im} V_{im} c_{vi}}((R_a + c_{vi}) W_{th} T_{im,in} - (R_a + c_{vi}) W_{ei} T_{im} - (W_{th} - W_{ei}) c_{vi} T_{im}),\\
&e_8 : \dot{p}_{im} = \frac{R_a T_{im}}{V_{im}}(W_{th} - W_{ei}) + \frac{p_{im}}{T_{im}}\dot{T}_{im},\\
&e_9 : \dot{T}_{em} = \frac{R_{em} T_{em}}{p_{em} V_{em} c_{ve}}((R_{em} + c_{ve}) W_{turbo} T_{em} - (R_{em} + c_{ve})(-W_{eo}) T_{t,in} - (W_{turbo} - (-W_{eo})) c_{ve} T_{em}),\\
&e_{10} : \dot{p}_{em} = \frac{R_{em} T_{em}}{V_{em}}(W_{turbo} - (-W_{eo})) + \frac{p_{em}}{T_{em}}\dot{T}_{em},\\
&e_{11} : \dot{T}_t = \frac{R_{em} T_t}{p_t V_{ex} c_{ve}}((R_{em} + c_{ve}) W_{exh} T_{exh} - (R_{em} + c_{ve}) W_{turbo} T_{turbo} - (W_{exh} - W_{turbo}) c_{ve} T_t),\\
&e_{12} : \dot{p}_t = \frac{R_{em} T_t}{V_{ex}}(W_{exh} - W_{turbo}) + \frac{p_t}{T_t}\dot{T}_t,\\
&e_{13} : \dot{\omega}_t = \frac{1}{J_t}((Tq_t - Tq_c) - \omega_f \omega_t),\\
&e_{14} : T_{af,in} = \begin{cases} T_{amb}, & \text{if } p_{amb} > p_{af} \\ T_{af}, & \text{if } p_{af} > p_{amb} \end{cases},
\end{aligned}
$$

$$e_{15} : W_{af} = \sqrt{\frac{max(p_{amb}, p_{af})}{H_{af}T_{af,in}}}\sqrt{max(p_{amb}, p_{af}) - min(p_{amb}, p_{af})} + f_{Waf} + f_{paf},$$

$$e_{16} : \Pi_c = \frac{p_c}{p_{af}},$$

$$e_{17} : W_c = \frac{\sqrt{1 - \frac{\Psi_c}{\Psi_{c,max}^2}}\Phi_{c,max}R_c^3\omega_t p_{af}}{2\pi T_{af}R_a} + f_{Wc},$$

$$e_{18} : \Psi_c = \frac{4\pi^2(R_a + c_{vi})T_{af}}{R_c^2\omega_t^2}\left(\Pi_c^{\frac{\kappa_{ic}-1}{\kappa_{ic}}} - 1\right),$$

$$e_{19} : \Phi_c = \frac{2\pi W_c R_a T_{af}}{R_c^3\omega_t p_{af}},$$

$$e_{20} : \eta_c = \frac{\Phi_c\eta_{c,max}}{\Phi_{c,max}^2}(2\Phi_{c,max} - \Phi_c),$$

$$e_{21} : Tq_c = \frac{W_c(R_a + c_{vi})T_{af}}{\eta_c\omega_t}\left(\Pi_c^{\frac{\kappa_{ic}-1}{\kappa_{ic}}} - 1\right),$$

$$e_{22} : T_{ic,in} = \begin{cases} T_c, & \text{if } \ p_c > p_{ic} \\ T_{ic}, & \text{if } \ p_{ic} > p_c \end{cases},$$

$$e_{23} : W_{ic} = \sqrt{\frac{max(p_{ic}, p_c)}{H_{ic}T_{ic,in}}}\sqrt{max(p_{ic}, p_c) - min(p_{ic}, p_c)} + f_{Wic},$$

$$e_{24} : T_{th} = \begin{cases} T_{ic}, & \text{if } \ p_{ic} > p_{im} \\ T_{im}, & \text{if } \ p_{im} > p_{ic} \end{cases},$$

$$e_{25} : W_{th} = \frac{p_{ic}A_{th}}{\sqrt{T_{ic}R_a}}\Psi_{th}(\Pi_{th}) + f_{Wth},$$

$$e_{26} : \Pi_{th} = \frac{p_{im}}{p_{ic}},$$

$$e_{27} : \Pi_{th,crit} = \left(\frac{2}{\kappa_{th} + 1}\right)^{\frac{\kappa_{th}}{\kappa_{th}-1}},$$

$$e_{28} : \Psi_{th} = \begin{cases} \sqrt{\kappa_{th}}\left(\frac{2}{\kappa_{th}+1}\right)^{\frac{\kappa_{th}+1}{2(\kappa_{th}-1)}}, & \text{if } \ \Pi_{th} \leq \Pi_{th,crit} \\ \sqrt{\frac{2\kappa_{th}}{\kappa_{th}-1}\left(\Pi_{th}^{\frac{2}{\kappa_{th}}} - \Pi_{th}^{\frac{\kappa_{th}+1}{\kappa_{th}}}\right)}, & \text{if } \ \Pi_{th} > \Pi_{th,crit} \end{cases},$$

$$e_{29} : W_{ei} = C_{\eta_{vol}}\frac{r_c - \left(\frac{p_{em}}{p_{im}}\right)^{\frac{1}{\kappa_{ei}}}}{r_c - 1}\frac{V_{im}\omega_e p_{im}}{4\pi R_a T_{im}} + f_{C_{vol}},$$

$$e_{30} : W_f = \frac{W_{ei}}{(A/F)_s\lambda_c},$$

$$e_{31} : W_{eo} = W_{ei} + W_f,$$

$$e_{32} : T_{eo} = (W_{eo}C_{eo}) + T_0,$$

$$e_{33} : T_{t,in} = T_{amb} + (T_{eo} - T_{amb})e^{-\frac{h_{tot}\pi d_{pipe} l_{pipe} n_{pipe}}{W_{eo}\frac{R_{em}\kappa_{em}}{\kappa_{em}-1}}},$$

$$e_{34} : T_{wg} = \begin{cases} T_{em}, & \text{if} \;\; p_{em} > p_t \\ T_t, & \text{if} \;\; p_t > p_{em} \end{cases},$$

$$e_{35} : W_{wg} = \frac{p_{em} u_{wg} c_{D,wg}}{\sqrt{T_{em} R_{em}}} \Psi_t(\Pi_t),$$

$$e_{36} : \Pi_t = \frac{p_t}{p_{em}},$$

$$e_{37} : \Pi_{t,crit} = \left(\frac{2}{\kappa_{em}+1}\right)^{\frac{\kappa_{em}}{\kappa_{em}-1}},$$

$$e_{38} : \Psi_t = \begin{cases} \sqrt{\kappa_{em}} \left(\frac{2}{\kappa_{em}+1}\right)^{\frac{\kappa_{em}+1}{2(\kappa_{em}-1)}}, & \text{if} \;\; \Pi_t \leq \Pi_{t,crit} \\ \sqrt{\frac{2\kappa_{em}}{\kappa_{em}-1}\left(\Pi_t^{\frac{2}{\kappa_{em}}} - \Pi_t^{\frac{\kappa_{em}+1}{\kappa_{em}}}\right)}, & \text{if} \;\; \Pi_t > \Pi_{t,crit} \end{cases},$$

$$e_{39} : BSR = \frac{d_t \omega_t}{2\sqrt{2c_{p,eg} T_{em}\left(1 - \Pi_t^{\frac{1-\kappa_{em}}{\kappa_{em}}}\right)}},$$

$$e_{40} : \eta_t = \eta_{t,max}\left(1 - \left(\frac{BSR - BSR_{eff,max}}{BSR_{eff,max}}\right)^2\right),$$

$$e_{41} : T_{t,out} = T_{em}\left(1 - \Pi_t^{\frac{\kappa_{em}-1}{\kappa_{em}}}\right)\eta_t,$$

$$e_{42} : W_t = \frac{k_{1,t} p_{em}}{\sqrt{T_{em}}}\sqrt{1 - \Pi_t^{k_{2,t}}},$$

$$e_{43} : Tq_t = \frac{W_t c_{p,eg} T_{t,out}}{\omega_t},$$

$$e_{44} : W_{turbo} = -(W_t + W_{wg}),$$

$$e_{45} : T_{turbo} = \frac{W_t c_{p,eg} T_{t,out} + W_{wg} c_{p,eg} T_{wg}}{W_t c_{p,eg} + W_{wg} c_{p,eg}},$$

$$e_{46} : T_{exh} = \begin{cases} T_{amb}, & \text{if} \;\; p_{amb} > p_t \\ T_t, & \text{if} \;\; p_t > p_{amb} \end{cases},$$

$$e_{47} : W_{exh} = \sqrt{\frac{max(p_t, p_{amb})}{H_{ex} T_{exh}}}\sqrt{max(p_t, p_{amb}) - min(p_t, p_{amb})},$$

$$e_{48} : u_{p_{amb}} = p_{amb},$$

$$e_{49} : u_{T_{amb}} = T_{amb},$$

$$e_{50} : u_{x_{th}} = A_{th} + f_{x_{th}},$$

$e_{51}: u_{\omega_{eREF}} = \omega_{eREF},$

$e_{52}: u_{x_{wg}} = u_{wg},$

$e_{53}: u_{\lambda_c} = \lambda_c,$

$e_{54}: y_{T_c} = T_c,$

$e_{55}: y_{p_c} = p_c,$

$e_{56}: y_{T_{ic}} = T_{ic} + f_{y_{T_{ic}}},$

$e_{57}: y_{p_{ic}} = p_{ic} + f_{y_{p_{ic}}},$

$e_{58}: y_{T_{im}} = T_{im},$

$e_{59}: y_{p_{im}} = p_{im} + f_{y_{p_{im}}},$

$e_{60}: y_{W_{af}} = W_{af} + f_{y_{W_{af}}},$

$e_{61}: y_{p_{em}} = p_{em},$

$e_{62}: y_{Tq_e} = Tq_e.$

TABLE S3: Variables in the engine model.

Variable	Description	Unit
p_{amb}	Ambient pressure	Pa
p_{af}	Air filter pressure	Pa
p_c	Compressor pressure	Pa
p_{ic}	Intercooler pressure	Pa
p_{im}	Intake manifold pressure	Pa
p_{em}	Exhaust manifold pressure	Pa
p_t	Turbine pressure	Pa
T_{amb}	Ambient temperature	K
T_{af}	Air filter temperature	K
$T_{af,in}$	Air filter in temperature	K
T_c	Compressor temperature	K
$T_{c,in}$	Compressor in temperature	K
T_{ic}	Intercooler temperature	K
$T_{ic,in}$	Intercooler in temperature	K
T_{im}	Intake manifold temperature	K
$T_{im,in}$	Intake manifold in temperature	K
T_{em}	Exhaust manifold temperature	K

TABLE S3: Variables in the engine model.

Variable	Description	Unit
T_t	Turbine temperature	K
$T_{t,in}$	Turbine in temperature	K
T_{wg}	Wastegate temperature	K
$T_{t,out}$	Temperature difference over turbine	K
T_{eo}	Engine out temperature	K
T_{turbo}	Turbine and wastegate mixture temperature	K
T_{exh}	Exhaust temperature	K
W_{af}	Mass flow through air filter	kg/s
W_c	Mass flow through compressor	kg/s
W_{ic}	Mass flow through intercooler	kg/s
W_{th}	Mass flow through throttle	kg/s
W_{ei}	Mass flow into engine	kg/s
W_f	Fuel mass flow	kg/s
W_{eo}	Mass flow out from engine	kg/s
W_{wg}	Mass flow through wastegate	kg/s
W_{turbo}	Mass flow of turbine and wastegate mixture	kg/s
W_{exh}	Mass flow through exhaust	kg/s
ω_t	Turbine speed	rad/s
ω_e	Engine speed	rad/s
Tq_t	Turbine torque	N·m
Tq_c	Compressor torque	N·m
Π_c	Pressure ratio in compressor	[–]
Π_{th}	Pressure ratio in throttle	[–]
$\Pi_{th,crit}$	Critical pressure ratio in throttle	[–]
Π_t	Pressure ratio in turbine	[–]
$\Pi_{t,crit}$	Critical pressure ratio in turbine	[–]
Φ_c	Energy transfer coefficient	[–]
Ψ_c	Compressor flow coefficient	%
Ψ_{th}	Throttle flow coefficient	%
Ψ_t	Turbine flow coefficient	%
η_c	Compressor efficiency	%
η_t	Turbine efficiency	%
BSR	Blade speed ratio	[–]

TABLE S4: Key parameters of the engine in this testbed.

Description	Value	Unit
Ambient Air Data:		
Ratio of specific heats, κ_{ic}	1.4	[–]
Gas constant, R_a	287.2	J/(kg·K)
Engine Block:		
Bore, B	0.0831	m
Displacement volume, V_d	0.0018	m^3
Number of cylinders, n_{cyl}	4	[–]
Number of revolutions per power stroke, n_r	2	[-]
Compression ratio, r_c	9.5	[–]
Boost layout, Π_{bl}	2	[–]
Factor for auxiliary devices, ξ_{aux}	1	[–]
Gross efficiency, η_{ig}	0.4	[–]
Stoichiometric factor air to fuel, $(A/F)_s$	15.1	[–]
Air to fuel ratio, λ_c	1	[–]
Volumetric efficiency constant, $C_{\eta_{vol}}$	0.8	[–]
Ratio of specific heats, κ_{ei}	1.3	[–]
Gas constant, R_{em}	290	J/(kg·K)
Intake manifold volume, V_{im}	0.0018	m^3
Exhaust manifold volume, V_{em}	0.0025	m^3
BMEP parameter 1, C_{Tq1}	$0.2{\times}10^6$	Pa
BMEP parameter 2, C_{Tq2}	$1.2{\times}10^6$	Pa
Temperature at 0 mass flow, T_0	1100	K
Temperature change with mass flow, C_{eo}	3000	K·s/kg
Fuel lower heating value, q_{HV}	$4.4{\times}10^7$	J/kg
Measurement constant, a_0	1.1647^{-5}	[-]
Measurement constant, a_1	3.0718^{-5}	[-]
Measurement constant, a_2	0.0029	[-]
Air Filter:		
Volume, V_{af}	0.01	m^3
Flow resistance, H_{af}	$2{\times}10^8$	[–]
Linearization limit, $p_{lin,af}$	2000	Pa
Compressor:		
Volume, V_c	0.005	m^3

TABLE S4: Key parameters of the engine in this testbed.

Description	Value	Unit
Diameter, D_c	0.06	m
Maximum efficiency, $\eta_{c,max}$	0.8	[–]
Minimum efficiency, $\eta_{c,min}$	0.3	[–]
Maximum flow coefficient, $\Phi_{c,max}$	0.12	[–]
Head parameter, $\Psi_{c,max}$	2.3	[–]
Throttle:		
Ratio of specific heats, κ_{th}	2	[–]
Maximum pressure ratio, $\Pi_{th,max}$	0.9	[–]
Intercooler:		
Volume, V_{ic}	0.005	m^3
Flow resistance, H_{ic}	4×10^8	[–]
Linearization limit, $p_{lin,ic}$	500	Pa
Heat transfer coefficient, h_{ic}	0.8	[–]
Regulated pressure drop across throttle, Δp_{thrREF}	10000	Pa
Exhaust and Turbine Inlet:		
Volume , V_{ex}	0.02	m^3
Ratio of specific heats, κ_{em}	1.3	[–]
Gas constant, R_{em}	290	J/(kg·K)
Pipe diameter, d_{pipe}	0.04	m
Pipe length, l_{pipe}	0.45	m
Number of parallel pipes, n_{pipe}	4	[–]
External heat transfer coefficient, h_{ext}	95	$\text{W/(m}^2\text{K)}$
Dynamic viscosity, μ_{em}	4×10^{-5}	kg/(m·s)
Thermal conductivity, k_{em}	0.07	W/(m·K)
Flow resistance, H_{ex}	3×10^8	[–]
Linearization limit, $p_{lin,ex}$	300	Pa
Turbocharger:		
Friction coefficient, ω_f	1×10^{-6}	[–]
Inertia of turbocharger, J_t	3×10^{-5}	kg·m^2
Initial speed, $\omega_{t,init}$	3000	rad/s
Minimum speed, $\omega_{t,min}$	2000	rad/s
Maximum speed, $\omega_{t,max}$	2.4×10^4	rad/s

TABLE S4: Key parameters of the engine in this testbed.

Description	Value	Unit
Turbine and Wastegate:		
Turbine diameter, d_t	0.05	m
Specific heat of gas, $c_{p,eg}$	1200	J/(kg·K)
Ratio of specific heats, κ_{em}	1.3	[–]
Maximum turbine efficiency, $\eta_{t,max}$	0.75	[–]
Minimum turbine efficiency, $\eta_{t,min}$	0.3	[–]
BSR at maximum turbine efficiency, $BSR_{eff,max}$	0.7	[–]
Mass flow constant 1, $k_{1,t}$	0.017	[–]
Mass flow constant 2, $k_{2,t}$	1.4	[–]
Discharge coefficient, $c_{D,wg}$	0.9	[–]
Maximum wastegate area, $A_{wg,max}$	3.5×10^{-4}	m^2

Sidebar: Modeling Mass Flow Fault Caused By A Leakage

The intensity of a mass flow fault caused by a leakage is determined by the area of the leakage orifice, which is usually measured in mm^2. The bigger the leakage area, the higher the mass flow through the leakage and the higher pressure difference between both sides of the orifice, hence the higher intensity of the fault. Conventionally, a mass flow fault is physically induced by drilling a hole on the specific component of the engine test bench system. Tests are then performed by running the engine through driving cycles and the effects of the fault on the performance of the engine are analyzed. The leakage orifice is then sealed using a screw plug to 'disable' the fault and to return the engine to its nominal operation mode. Therefore, by simulating the mass flow fault using a simulation testbed in Matlab/Simulink, it removes the need for the engine system to be physically modified to accommodate such fault, which could lead to irreversible damages.

The mathematical modeling of the mass flow fault for compressible flows was briefly discussed in [3], where the flow through the leakage can be described using

$$W_{leak} = k_{leak} \frac{p_{high}}{\sqrt{T_{amb}}} \Psi \left(\frac{p_{low}}{p_{high}} \right), \tag{S1}$$

where k_{leak} is the area of the leakage orifice (mm^2), T_{amb} the ambient temperature (K), while p_{high} and p_{low} are the higher and lower pressures (Pa) on either side of the leakage. The function $\Psi \left(\frac{p_{low}}{p_{high}} \right)$ is defined as

$$\Psi \left(\tfrac{p_{low}}{p_{high}} \right) = \begin{cases} \sqrt{\tfrac{2\kappa}{\kappa-1} \left\{ \left(\tfrac{p_{low}}{p_{high}} \right)^{2/\kappa} - \left(\tfrac{p_{high}}{p_{low}} \right)^{(\kappa+1)/\kappa} \right\}}, & \text{if } \left(\tfrac{p_{low}}{p_{high}} \right) \geq \left(\tfrac{2}{\kappa+1} \right)^{\kappa/(\kappa-1)} \\ \sqrt{\kappa \left(\tfrac{2}{\kappa+1} \right)^{(\kappa+1)/(\kappa-1)}}, & \text{otherwise} \end{cases}, \tag{S2}$$

where κ is the specific heat ratio in the affected part of the engine system.

Sidebar: Structural Model and FIM: A General Tutorial

Let's consider a DC motor system shown in Figure S1, which can be modeled using the following equations:

$$e_1 : V = i(R + f_R) + L\frac{di}{dt} + K_a i\omega, \tag{S3}$$

$$e_2 : T_m = K_a i^2, \tag{S4}$$

$$e_3 : J\frac{d\omega}{dt} = \Delta T - K_b\omega, \tag{S5}$$

$$e_4 : \Delta T = T_m - T_L, \tag{S6}$$

$$e_5 : \frac{d\theta}{dt} = \omega, \tag{S7}$$

$$e_6 : \frac{d\omega}{dt} = \alpha, \tag{S8}$$

$$e_7 : y_i = i + f_i, \tag{S9}$$

$$e_8 : y_\omega = \omega + f_\omega, \tag{S10}$$

$$e_9 : y_\Delta = \Delta T + f_\Delta, \tag{S11}$$

where V is the input voltage, R the resistance, L the inductance, and i the current in the armature circuit. On the mechanical side of the system, T_m is the motor torque, T_L the torque of the load, J the moment of inertia of the rotor, K_a the motor torque constant, and K_b the back EMF constant. The rotational displacement of the motor is θ while ω and α are the rotational velocity and acceleration, respectively. The states of the system are $\{i, \theta, \omega, \alpha, T_m, T_L, \Delta T\}$ and the measurable outputs are $\{i, \omega, \Delta T\}$. It is assumed that there is a system fault f_R representing inconsistency in the value of the resistance R. It is also assumed that all outputs are potentially faulty through f_i, f_ω, and f_Δ, respectively, as shown in (S9)–(S11).

Using Structural Model, the relationships among the unknown variables (states), known variables (inputs and outputs), and faults in the system can be explained using Table S5. In the table, an 'X' is placed in the corresponding columns where the variables or faults are used to explain each equation in (S3)–(S11). For example, the states i and ω, as well as the fault f_R are used in equation e_1 in (S3) to express the input voltage V.

By performing canonical decomposition onto the unknown variables in Table S5, the Structural Model can then be remodeled as Table S6. Then, the Fault Isolation Matrix (FIM) can be obtained by extracting the bottom-right section of the the Structural Model in Table S6. It can be seen that for the DC motor system, the pair $\{f_R, f_i\}$ are not isolable from each other. See Table S7.

TABLE S5: Structural Model of the DC motor system as modeled using (S3)–(S11). Here, the relationships among the unknown variables (states), known variables (inputs and outputs), and faults in the system can be explained by placing an 'X' in the corresponding columns where the variables or faults are used to explain each equation.

	i	θ	ω	α	T_m	T_L	ΔT	V	y_i	y_ω	y_Δ	f_R	f_i	f_ω	f_Δ
e_1	X		X					X				X			
e_2	X				X										
e_3			X				X								
e_4					X	X	X								
e_5		X	X												
e_6			X	X											
e_7	X								X				X		
e_8			X							X				X	
e_9							X				X				X

TABLE S6: A remodel of the Structural Model in Table S5 by performing canonical decomposition onto the unknown variables.

	θ	α	T_L	T_m	i	ΔT	ω
e_5	X						X
e_6		X					X
e_4			X	X		X	
e_2				X	X		
e_1			f_R	$\longrightarrow$	X		X
e_7			f_i	$\longrightarrow$	X		
e_3						X	X
e_9			f_Δ	$\longrightarrow$		X	
e_8			f_ω	$\longrightarrow$			X

TABLE S7: The Fault Isolation Matrix (FIM) of the DC motor system in (S3)–(S11). It can be seen that the pair $\{f_R, f_i\}$ are not isolable from each other.

	f_R	f_i	f_Δ	f_ω
f_R	X	X		
f_i	X	X		
f_Δ			X	
f_ω				X

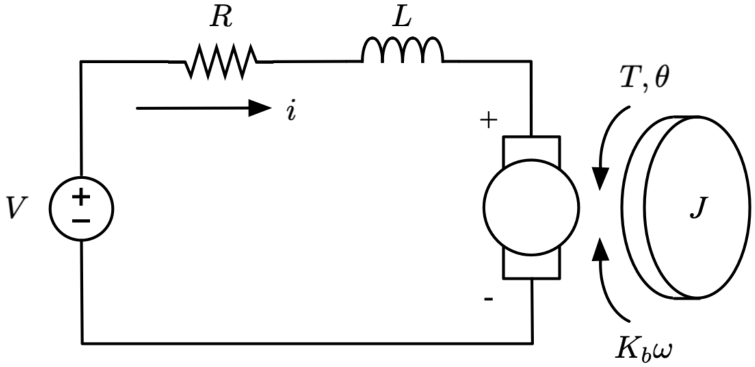


Figure S1: A general DC motor system commonly used in control engineering studies.

Author Biography

Kok Yew Ng (mark.ng@ulster.ac.uk) is currently a Lecturer with the School of Engineering, Ulster University, UK, where he also conducts research at the Nanotechnology and Integrated Bioengineering Centre (NIBEC). He received the B.Eng degree (Hons) in Electrical & Computer Systems Engineering and the PhD degree in fault diagnosis from Monash University in 2006 and 2009, respectively. From 2014 to 2015, he was a post-doc with the Division of Vehicular Systems, Linköping University, Sweden, where he focused on the design and development of advanced fault diagnosis schemes using model-based and data-driven methods on vehicular engines with Volvo Car Corporation. He is also an Adjunct Senior Research Fellow with the School of Engineering, Monash University Malaysia. His research interests are fault diagnosis, sliding mode observer, vehicular systems, and data analytics for anomaly detection and classification.

Erik Frisk is a Professor at the Department of Electrical Engineering at Linköping University, Sweden. Current research interests include issues in model-based fault diagnosis and fault isolation, prognostics, and autonomous functions.

Mattias Krysander is an Associate Professor at the Department of Electrical Engineering, Linköping University, Sweden. His research interests include model-based diagnosis and prognosis. As a way to cope with the complexity and size of industrial systems, mainly vehicle systems, he has used structural representations of models and developed graph theoretical methods for assisting design of diagnosis systems and for fault isolation and sensor placement analysis.

Lars Eriksson is Full Professor in Vehicular Systems at Linköping University where he also manages the engine laboratory. He received the M.Sc. degree in Electrical Engineering in 1995 and the PhD degree in Vehicular Systems in 1999, both from Linköping University. During 2000 and 2001, he was a post-doc in the Measurement and Control group at Swiss Federal Institute of Technology (ETH) in Zurich. His research interests are modeling, simulation, and control of internal combustion engines for vehicle propulsion in general, but with a focus on downsizing and supercharging concepts for improved fuel economy. His research was the first to demonstrate real-time control of the combustion timing using information obtained from the ion current. He has published one book, three book chapters, and more than 140 international peer-reviewed articles. He is a member of the IFAC Technical Board as Chair for Coordinating Committee CC7 on Transportation and Vehicle Systems. He is also Associate Editor for the Elsevier journal Control Engineering Practice, and has served as Adjoint Technical Editor for several conferences such as the IFAC World Congresses, Advances in Automotive Control, and E-COSM.