The study of fault diagnosis on automotive engine systems has been an interesting and ongoing topic for many years. Numerous research projects were conducted by automakers and research institutions to discover new and more advanced methods to perform diagnosis for better fault isolation (FI). Some of the research in this field has been reported in [1]–[5].

In most automotive systems today, the diagnostic systems monitor multiple components in the engine and are independent of each other. However,
In most automotive systems today, the diagnostic systems monitor multiple components in the engine and are independent of each other.

some faults have a tendency to manifest and simultaneously or subsequently trigger several other monitors [6]. For instance, a disconnected intake system hose has a high potential to result in flow and pressure faults further along in the engine system. To overcome this, residuals from several monitors (coupled with an intelligent algorithm) are needed to enhance the FI location accuracy and identify the root cause of the problem. The ability to identify the cause of the fault and pinpoint its exact location is crucial for taking proper measures and avoiding the replacement of misdiagnosed engine components [7].

One of the main disadvantages of existing diagnostic systems is that the faults are not detected in a chronological order. As a result (depending on the location of the monitors in the engine and the propagation time of the electrical signals), the manifested fault(s) may trigger the monitors much sooner than the root cause of the problem. Also, should a monitor break or not operate well, it may take more time to detect the root fault, or worse, the fault may not be detected at all. This will, in turn, lead to incorrect onboard diagnostic reconfiguration efforts or the erroneous replacement of the so-called faulty components [8], [9]. These misdiagnosis and robustness issues are especially critical in autonomous vehicular systems, where it is essential for the onboard computers to know the health of the system so that they can take corrective measures to protect the lives of the occupants as well as others on the road. Depending on the severity of the fault, the vehicle can be reconfigured to operate at a reduced performance level (to ensure safety until the car is taken to a repair shop for repairs and maintenance) or 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 the recoverability of autonomous systems can be found in [10]–[12].

The failure to detect and isolate a fault (or the incorrect identification of one) may cause the reconfiguration of the system to be unoptimized. This affects the health and lifespan of individual components and the engine as a whole. Therefore, an improved fault diagnosis method is crucial not only to identify the root cause of the problem but immediately and correctly reconfigure the engine before the condition worsens. While many software and simulation packages provide interesting studies of the dynamics of the engine system, very few have explored the design and analysis of fault diagnosis schemes (see [13]–[17]).

This article presents a simulation testbed of the engine system, whereby its operation can be realistically modeled using industrial-standard driving cycles, such as the Worldwide Harmonized Light Vehicle Test Procedure (WLTP), New European Driving Cycle (NEDC), Extra-Urban Driving Cycle (EUDC), and U.S. Environmental Protection Agency Federal Test Procedure (FTP-75) [18] (see “Summary”). A graphical user interface (GUI) enables users 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 physically inducing the faults, thus minimizing the risk of shortening the engine’s lifespan or causing permanent damage. The simulation environment (available as open-source software at https://github.com/nykmark/TCSISimTestbed) is intended to enhance the development of theoretical and physical applications of engine system fault diagnosis, with the hope that researchers in the field will collaborate or use the tool as a virtual laboratory for teaching purposes.

**Summary**

Research on fault diagnosis on highly nonlinear dynamic systems, such as the engine of a vehicle, has attracted much interest in recent years, especially as the automotive industry heads toward self-driving technologies. This article presents a novel open-source simulation testbed of a turbocharged spark-ignited gasoline engine system for testing and evaluating residuals generation and fault diagnosis methods. Designed and developed using Matlab/Simulink, the user interacts with the testbed through a graphical interface where the engine can be realistically simulated using industrial-standard driving cycles, including the Worldwide Harmonized Light Vehicle Test Procedure, New European Driving Cycle, Extra-Urban Driving Cycle, and U.S. Environmental Protection Agency Federal Test Procedure. The engine is modeled by using the mean value engine model and controlled through a proportional-integral-based boost controller. The graphical interface also enables the user to induce one of the 11 faults of interest and gain a better understanding of the fault’s effects on the performance of the engine. This minimizes the risk of causing permanent damage to the engine and shortening its lifespan. The simulation testbed will serve as an excellent platform where researchers can generate critical data to develop and compare current and future research methods for the fault diagnosis of automotive engine systems.
MODELING THE ENGINE

The simulation environment uses a four-cylinder single-turbocharged spark-ignited (TCSI) gasoline engine as the testbed to design and verify the performance of fault diagnosis schemes. Figure 1 shows the engine test bench used for data collection in the lab, while Figure 2 presents 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 matter from reaching the engine block.

» **Compressor**: This component is modeled on a radial compressor and driven by the turbine. The filtered air is compressed to increase its volumetric flow, pressure, and temperature.

» **Intercooler**: Air from the compressor is cooled while its mass flow velocity is maintained.

» **Throttle**: This controls the intake manifold pressure, 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 four cylinders in the engine.

» **Engine block**: The combustion mixture is ignited to generate the torque for the mechanical work.

» **Exhaust manifold**: Gases produced by the combustion reactions in the engine are directed to the turbine and the wastegate.

» **Turbine**: This device harvests energy from the gases in the exhaust manifold to generate power to drive the compressor.

» **Wastegate**: This valve bypasses the turbine and controls the power delivered by the turbocharger.

» **Exhaust system**: Gases from the engine system exit to the ambient environment.

The engine is modeled by using differential equations that describe the airflow through the subsystems. The equations are derived from the mean value engine model for a TCSI engine, as reported in [19] and [20]. The key parameters of the vehicle and engine used for this testbed (as well as the model equations of the engine with a list of engine variables) can be found in “Vehicle Parameters” and...
“The Differential Equations of the Turbocharged Spark-Ignited Engine System and Corresponding Engine Parameters.” The system has 13 states \((T_{th}, p_{th}, T_i, p_i, T_w, p_w, T_{int}, p_{int}, T_{emf}, p_{emf}, T_r, p_r, \text{ and } \omega)\) that represent the temperature (K) and pressure (Pa) in the air filter, compressor, intercooler, intake manifold, exhaust manifold, and turbine as well as the turbine speed (rad/s), respectively. Six actuators \((A_{th}, u_{weg}, \omega_{REF}, \lambda, p_{amb}, \text{ and } T_{amb})\) represent the throttle position area (m²), wastegate input \((0 \ldots 1)\), reference engine speed (rad/s), air–fuel ratio, ambient pressure (Pa), and temperature (K), respectively. The system has nine sensors \((T_s, p_s, T_{ic}, p_{ic}, T_{im}, p_{im}, T_{emf}, p_{emf}, T_{w}, p_{w})\), where \(W_{st}\) and \(T_{q_f}\) are the mass flow in the air filter (kg/s) and engine torque (N·m), respectively (see Table 1).

**GENERATION OF REFERENCE INPUTS AND CONTROLLER DESIGN**

This section discusses the design of the proportional-integral-based boost controller with antiwindup to generate the control inputs for the throttle effective area \(A_{th}\) and wastegate actuator for the turbocharger \(u_{weg}\). Figure 3 displays the closed-loop engine control system with the boost controller. See Tables S3 and S4 in “The Turbocharged Spark-Ignited 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 r/min in eighth gear is 62.9 km/h. From the key vehicle parameters in Table S1, the vehicle speed per 1000 r/min (km/h), \(v_{e,1000/r\text{min}}\) for each gear is

\[
v_{e,1000/r\text{min}} = \frac{120 \times v_{e,\text{1000/rmin}}}{\text{final gear ratio} \times \text{current gear ratio}},
\]

where \(r_w\) is the wheel radius (m).

The results are then tabulated in Table S2, which also shows the gearbox’s estimated gear shift points. The data in Tables S1 and S2 (with the vehicle speed versus time information from the driving cycle profile) provide the reference engine speed (rad/s), \(\omega_{REF}\), and reference engine torque (N·m) \(T_{q,REF}\) for the boost controller. The reference engine speed is

\[
\omega_{REF} = \frac{V_i \cdot g(V)}{r_w},
\]

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 \(T_{q,REF}\), the vehicle’s force equation is first expressed using

\[
m_v \cdot V = F_w - F_d - F_r,
\]

where \(F_w\), \(F_d\), and \(F_r\) are the forces (N) at the wheel, drag resistance force, and roll resistance force, respectively, and
The Differential Equations of the Turbocharged Spark-Ignited Engine System and Corresponding Parameters

The testbed is modeled on the engine test bench in the lab, as shown in Figure 1. The model equations for the turbocharged spark-ignited (TCIS) engine are stated here. The first 47 of them represent the TCIS engine, as derived in [19] and [20]. The remaining 15 (\(e_{48}, \ldots, e_{62}\)) describe the considered sensors and actuators, as shown in Table 1. The faults introduced in Table 2 can be found in equations \(e_{15}, e_{17}, e_{23}, e_{25}, e_{29}, e_{30}, e_{56}, e_{57}, e_{59}\), and \(e_{60}\), respectively. Table S3 presents the engine model variables, and Table S4 conveys the corresponding key engine parameters.

\[
\begin{align*}
\psi_c &= \frac{4\rho_c^2 (R_s + c \omega) T_d}{R_c^2 \omega_t^2}, \\
\psi_{cmax} &= \frac{2\pi W_c R_s T_d}{R_c^2 \omega_t^2 p_c}, \\
\eta_c &= \frac{\psi_c}{\psi_{cmax}}, \\
\Phi_c &= \frac{W_c (R_s + c \omega) T_d}{\eta_c \omega_t^2}, \\
\eta_{cmax} &= \frac{2\pi W_c R_s T_d}{\Omega_c \omega_t^2 p_c - \phi_c}, \\
\Omega_c &= \frac{W_c (R_s + c \omega) T_d}{\eta_c \omega_t^2}, \\
\Omega_{cmax} &= \frac{2\pi W_c R_s T_d}{\Omega_c \omega_t^2 p_c - \phi_c}, \\
\Psi_c &= \frac{4\rho_c^2 (R_s + c \omega) T_d}{R_c^2 \omega_t^2} (1 - \frac{\Omega_{cmax}^2 - 1}{\Omega_c^2}), \\
\Psi_{cmax} &= \frac{2\pi W_c R_s T_d}{R_c^2 \omega_t^2 p_c}, \\
\eta_{cmax} &= \frac{2\pi W_c R_s T_d}{\Omega_c \omega_t^2 p_c - \phi_c}.
\end{align*}
\]
\( \theta_{41}: \) \( T_{\text{out}} = T_{\text{am}} (1 - \frac{\rho_{\text{amb}}}{\rho_{\text{in}}}) \eta_{c} \)
\( \theta_{42}: \) \( W_{i} = \frac{k_{1} p_{\text{am}}}{\sqrt{T_{\text{am}}}} \sqrt{1 - \frac{P_{c}^{\text{crit}}}{P_{c}}}, \)
\( \theta_{43}: \) \( T_{Q_{t}} = \frac{W_{c} C_{p} q_{Wc}}{\eta_{t}} \)
\( \theta_{44}: \) \( W_{\text{turbo}} = -(W_{i} + W_{w}), \)
\( \theta_{45}: \) \( T_{\text{turbo}} = \frac{W_{c} C_{p} q_{Wc} + W_{w} C_{p} q_{w}}{W_{c} C_{p} q_{Wc}} \)
\( \theta_{46}: \) \( T_{\text{en}} = \begin{cases} T_{\text{amb}}, & \text{if } \rho_{\text{amb}} > \rho_{t} \\ T_{t}, & \text{if } \rho_{t} > \rho_{\text{amb}}. \end{cases} \)
\( \theta_{47}: \) \( W_{\text{en}} = \sqrt{\max(p_{t}, \rho_{\text{amb}})} \sqrt{\max(p_{t}, \rho_{\text{amb}}) - \min(p_{t}, \rho_{\text{amb}})}, \)
\( \theta_{48}: \) \( u_{p,\text{amb}} = \rho_{\text{amb}}, \)
\( \theta_{49}: \) \( u_{T,\text{en}} = T_{\text{amb}}. \)
\( \theta_{50}: \) \( u_{x}= = A_{\text{th}} + f_{\text{int}}, \)
\( \theta_{51}: \) \( u_{\text{swt}} = \omega_{\text{REER}}, \)
\( \theta_{52}: \) \( u_{\text{sw}} = u_{\text{w}}, \)
\( \theta_{53}: \) \( u_{h} = \lambda_{i}, \)
\( \theta_{54}: \) \( y_{t} = T_{e}, \)
\( \theta_{55}: \) \( y_{p} = \rho_{c}, \)
\( \theta_{56}: \) \( y_{r} = T_{c} + f_{\text{turbo}}, \)
\( \theta_{57}: \) \( y_{u} = p_{c} + f_{\text{turbo}}, \)
\( \theta_{58}: \) \( y_{\text{th}} = T_{\text{th}}, \)
\( \theta_{59}: \) \( y_{\text{pf}} = p_{\text{in}} + f_{\text{turbo}}, \)
\( \theta_{60}: \) \( y_{\text{w}} = W_{\text{af}} + f_{\text{turbo}}, \)
\( \theta_{61}: \) \( y_{\text{out}} = T_{\text{out}}. \)

**TABLE S3** The engine model variables.

| Variable | Description | Unit |
|----------|-------------|------|
| \( p_{\text{amb}} \) | Ambient pressure | Pa |
| \( p_{\text{af}} \) | Air filter pressure | Pa |
| \( p_{c} \) | Compressor pressure | Pa |
| \( p_{\text{ic}} \) | Intercooler pressure | Pa |
| \( p_{\text{im}} \) | Intake manifold pressure | Pa |
| \( p_{\text{en}} \) | Exhaust manifold pressure | Pa |
| \( p_{t} \) | Turbine pressure | Pa |
| \( T_{\text{amb}} \) | Ambient temperature | K |
| \( T_{\text{af}} \) | Air filter temperature | K |
| \( T_{\text{af,in}} \) | Air filter-in temperature | K |
| \( T_{c} \) | Compressor temperature | K |
| \( T_{\text{ic}} \) | Compressor-in temperature | K |
| \( T_{\text{ic,in}} \) | Intercooler-in temperature | K |
| \( T_{\text{im}} \) | Intake manifold temperature | K |
| \( T_{\text{im,in}} \) | Intake manifold-in temperature | K |
| \( T_{\text{en}} \) | Exhaust manifold temperature | K |
| \( T_{t} \) | Turbine temperature | K |
| \( T_{\text{t,in}} \) | Turbine-in temperature | K |
| \( T_{w} \) | Wastegate temperature | K |
| \( T_{\text{out}} \) | Temperature difference over turbine | K |
| \( T_{e} \) | Engine-out temperature | K |
| \( T_{\text{turbo}} \) | Turbine and wastegate mixture temperature | K |
| \( T_{\text{exh}} \) | Exhaust temperature | K |
| \( W_{\text{af}} \) | Mass flow through air filter | kg/s |
| \( W_{c} \) | Mass flow through compressor | kg/s |
| \( W_{\text{ic}} \) | Mass flow through intercooler | kg/s |
| \( W_{\text{th}} \) | Mass flow through throttle | kg/s |
| \( W_{\text{as}} \) | Mass flow into engine | kg/s |
| \( W_{f} \) | Fuel mass flow | kg/s |
| \( W_{\text{vo}} \) | Mass flow out from engine | kg/s |
| \( W_{\text{wg}} \) | Mass flow through wastegate | kg/s |
| \( W_{\text{turbo}} \) | Mass flow of turbine and wastegate mixture | kg/s |
| \( W_{\text{exh}} \) | Mass flow through exhaust | kg/s |
| \( \omega_{t} \) | Turbine speed | rad/s |
| \( \omega_{e} \) | Engine speed | rad/s |
| \( T_{\text{Q}} \) | Turbine torque | N m |
| \( T_{\text{Q c}} \) | Compressor torque | N m |
| \( \Pi_{c} \) | Pressure ratio in compressor | — |
| \( \Pi_{\text{th}} \) | Pressure ratio in throttle | — |
| \( \Pi_{\text{CRIT}} \) | Critical pressure ratio in throttle | — |
| \( \Pi_{c} \) | Pressure ratio in turbine | — |
| \( \Pi_{\text{CRIT}} \) | Critical pressure ratio in turbine | — |
| \( \phi_{c} \) | Energy transfer coefficient | — |
| \( \Psi_{c} \) | Compressor flow coefficient | % |
| \( \Psi_{\text{th}} \) | Throttle flow coefficient | % |
| \( \Psi_{t} \) | Turbine flow coefficient | % |
| \( \eta_{c} \) | Compressor efficiency | % |
| \( \eta_{t} \) | Turbine efficiency | % |
| BSR | Blade speed ratio | — |

(Continued)
The Differential Equations of the Turbocharged Spark-Ignited Engine System and Corresponding Parameters (Continued)

| Description                  | Value  | Unit     |
|------------------------------|--------|----------|
| Ambient-air data             |        |          |
| Ratio of specific heats, $\kappa_{ic}$ | 1.4    | [—]      |
| Gas constant, $R_s$          | 287.2  | J/(kg K) |
| Engine block                 |        |          |
| Bore, $B$                    | 0.0831 | m        |
| Displacement volume, $V_d$  | 0.0018 | m³       |
| Number of cylinders, $n_{cyl}$ | 4      | [—]      |
| Number of revolutions/power stroke, $n_r$ | 2      | [—]      |
| Compression ratio, $r_c$     | 9.5    | [—]      |
| Boost layout, $\Pi_B$        | 2      | [—]      |
| Factor for auxiliary devices, $\xi_{aux}$ | 1      | [—]      |
| Gross efficiency, $\eta_{ig}$ | 0.4    | [—]      |
| Stoichiometric factor air-to-fuel, ($A/F$)$_s$ | 15.1   | [—]      |
| Air-to-fuel ratio, $\lambda$ | 1      | [—]      |
| Volumetric efficiency constant, $C_{n,sl}$ | 0.8    | [—]      |
| Ratio of specific heats, $\kappa_{ei}$ | 1.3    | [—]      |
| Gas constant, $R_{em}$       | 290    | J/(kg K) |
| Intake manifold volume, $V_{in}$ | 0.0018 | m³       |
| Exhaust manifold volume, $V_{em}$ | 0.0025 | m³       |
| BMEP parameter 1, $C_{\tau_{l1}}$ | 0.2 × 10$^6$ | Pa     |
| BMEP parameter 2, $C_{\tau_{l2}}$ | 1.2 × 10$^6$ | Pa     |
| Temperature at zero mass flow, $T_0$ | 1100   | K        |
| Temperature change with mass flow, $C_{ao}$ | 3000   | K/s/kg   |
| Fuel lower heating value, $q_{HV}$ | 4.4 × 10$^7$ | J/kg    |
| Measurement constant, $a_0$  | 1.1647 | [—]      |
| Measurement constant, $a_1$  | 3.0718 | [—]      |
| Measurement constant, $a_2$  | 0.0029 | [—]      |
| Air filter                   |        |          |
| Volume, $V_{af}$             | 0.01   | m³       |
| Flow resistance, $H_{af}$    | 2 × 10$^8$ | Pa$^2$s$^2$/kg$^2$K |
| Linearization limit, $\rho_{ln,af}$ | 2000   | Pa       |
| Compressor                   |        |          |
| Volume, $V_c$                | 0.005  | m³       |
| Diameter, $D_c$              | 0.06   | m        |
| Maximum efficiency, $\eta_{\text{MAX}}$ | 0.8    | [—]      |
| Minimum efficiency, $\eta_{\text{MIN}}$ | 0.3    | [—]      |
| Maximum flow coefficient, $\Phi_{\text{MAX}}$ | 0.12   | [—]      |
| Head parameter, $\Psi_{\text{MAX}}$ | 2.3    | [—]      |

**TABLE S4** The key parameters of the engine in this testbed. BMEP: brake mean effective pressure; BSR: blade speed ratio.

| Description                  | Value  | Unit     |
|------------------------------|--------|----------|
| Throttle                     |        |          |
| Ratio of specific heats, $\kappa_{th}$ | 2      | [—]      |
| Maximum pressure ratio, $\Pi_{th\text{MAX}}$ | 0.9    | [—]      |
| Intercooler                  |        |          |
| Volume, $V_x$                | 0.005  | m³       |
| Flow resistance, $H_x$       | 4 × 10$^8$ | Pa$^2$s$^2$/kg$^2$K |
| Linearization limit, $\rho_{ln,x}$ | 500    | Pa       |
| Heat transfer coefficient, $h_x$ | 0.8    | W/(m$^2$K) |
| Regulated pressure drop across throttle, $\Delta P_{\text{REF}}$ | 10,000 | Pa |
| Exhaust and turbine inlet    |        |          |
| Volume, $V_{ex}$             | 0.02   | m³       |
| Ratio of specific heats, $\kappa_{em}$ | 1.3    | [—]      |
| Gas constant, $R_{em}$       | 290    | J/(kg K) |
| Pipe diameter, $d_{pipe}$    | 0.045  | m        |
| Pipe length, $l_{pipe}$      | 0.45   | m        |
| Number of parallel pipes, $n_{pipe}$ | 4      | [—]      |
| External heat transfer coefficient, $h_{ext}$ | 95    | W/(m$^2$K) |
| Dynamic viscosity, $\mu_{em}$ | 4 × 10$^{-5}$ | kg/(m s) |
| Thermal conductivity, $k_{em}$ | 0.07   | W/(m K)  |
| Flow resistance, $H_{ex}$    | 3 × 10$^8$ | Pa$^2$s$^2$/kg$^2$K |
| Linearization limit, $\rho_{ln,ex}$ | 300    | Pa       |
| Turbocharger                  |        |          |
| Friction coefficient, $\omega_f$ | 1 × 10$^{-6}$ | [—]      |
| Inertia of turbocharger, $J_f$ | 3 × 10$^{-5}$ | kg m$^2$ |
| Initial speed, $\omega_{\text{INIT}}$ | 3000   | rad/s    |
| Minimum speed, $\omega_{\text{MIN}}$ | 2000   | rad/s    |
| Maximum speed, $\omega_{\text{MAX}}$ | 2.4 × 10$^4$ | rad/s    |
| Turbine and wastegate         |        |          |
| Turbine diameter, $d_t$      | 0.05   | m        |
| Specific heat of gas, $c_{p,ag}$ | 1200   | J/(kg K) |
| Ratio of specific heats, $\kappa_{em}$ | 1.3    | [—]      |
| Maximum turbine efficiency, $\eta_{\text{MAX}}$ | 0.75  | [—]      |
| Minimum turbine efficiency, $\eta_{\text{MIN}}$ | 0.3    | [—]      |
| BSR at maximum turbine efficiency, $\text{BSR}_{\text{MAX}}$ | 0.7    | [—]      |
| Mass flow constant 1, $k_{1,1}$ | 0.017  | [—]      |
| Mass flow constant 2, $k_{2,2}$ | 1.4    | [—]      |
| Discharge coefficient, $c_{D,\text{wg}}$ | 0.9    | [—]      |
| Maximum wastegate area, $A_{\text{wgMAX}}$ | 3.5 × 10$^{-4}$ | m$^2$ |
$m_v$ is the mass of the vehicle (kg). The forces $F_d$ and $F_r$ can be further determined using

$$F_d = \frac{1}{2} \rho_s c_d A_f V^2,$$ \hspace{1cm} (4)

where $\rho_s = 1.29$ kg/m$^3$ is the air density, $g$ is gravity (m/s$^2$), $c_d$ is the drag coefficient, and $A_f$ represents the frontal area of the vehicle (m$^2$). If the torque produced at the wheel is written as $T_{q_w} = F_w r_w$, the reference engine torque can be finally expressed as

$$T_{q, \text{REF}} = \frac{T_{q_w}}{I_{\text{gear}}}.$$ \hspace{1cm} (6)

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

$$\text{BMEP}_{\text{REF}} = \frac{2 \pi n_r T_{q, \text{REF}}}{V_d},$$ \hspace{1cm} (7)

where $T_{q, \text{REF}}$ is the reference engine torque (N·m), $V_d$ is the displacement volume of the engine (m$^3$), and $n_r$ is the number of engine revolutions per power stroke (for a four-cylinder engine, $n_r = 2$). As a result, the reference intake manifold and intercooler pressures ($p_{\text{im}_{\text{REF}}}$ and $p_{\text{ic}_{\text{REF}}}$, respectively) are obtained as

$$p_{\text{im}_{\text{REF}}} = \text{BMEP}_{\text{REF}} + C_{\text{P0}},$$

$$p_{\text{ic}_{\text{REF}}} = p_{\text{im}_{\text{REF}}} + \Delta p_{\text{ic}_{\text{REF}}},$$ \hspace{1cm} (9)

where $\Delta p_{\text{ic}_{\text{REF}}}$ is the regulated pressure drop across the throttle (Pa). The constants $C_{\text{P0}}$ and $C_{\text{P1}}$ are computed as

![TABLE 1 The system states, actuators, and sensor measurements of the engine system in Figure 2.](image)

![FIGURE 3 The closed-loop engine control system with the boost controller, actuators, and sensors. Mux: multiplexer.](image)
where $T_{Qe}$ is the measured engine torque (N·m) and $p_{im}$ is the measured intake manifold pressure (Pa).

The reference throttle effective area (m$^2$) $A_{thREF}$ is then computed as

$$ A_{thREF} = W_{thREF} \sqrt{\frac{R_a}{T_{amb}}} \Psi_{thREF}, \quad (11) $$

where $R_a$ is the gas constant (J/(kg·K)), $T_{amb}$ is the ambient temperature, and $\Psi_{thREF}$ is the reference throttle flow coefficient (%). The reference mass flow into the engine (kg/s), $W_{thREF}$, and $\Psi_{thREF}$ are computed as

$$ W_{thREF} = \frac{C_{th REF} \cdot V_d \cdot \omega_{th REF} \cdot p_{th REF}}{4\pi R_a \cdot (\gamma - 1) \cdot T_{im}} \cdot (r_c - \left(\frac{p_{im} \cdot \kappa_{em}}{p_{th REF}}\right)^{\gamma_{em}}), \quad (12) $$

$$ \Psi_{thREF} = \frac{2 \cdot p_{th REF}}{\sqrt{\frac{r_c}{\kappa_{th REF} - 1} \left(\Pi_{th REF}^{r_c} - \Pi_{th REF}^{r_c+1}\right)}}, \quad (13) $$

where $C_{th REF}$ is the volumetric efficiency constant, $r_c$ is the compression ratio, $T_{im}$ is the intake manifold temperature (K), $p_{im}$ is the exhaust manifold pressure (Pa), $\kappa_{em}$ is the ratio of specific heats at the exhaust, and $\kappa_{th}$ is the ratio of specific heats at the throttle. The pressure ratio in the throttle $\Pi_{th REF}$ is obtained as

$$ \Pi_{th REF} = \frac{p_{th REF}}{p_{im}}, \quad (14) $$

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

To design the controller with antiwindup for the throttle, the reference throttle position $\alpha_{thREF}$ is computed as

$$ \alpha_{thREF} = \alpha_{thFF} + \alpha_{thFB}, \quad (15) $$

where $\alpha_{thFF}$ and $\alpha_{thFB}$ are the feedforward and feedback components of the throttle position controller, respectively. Using the solution from (11), $\alpha_{thFF}$ is expressed as

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

where the constants $a_0$, $a_1$, and $a_2$ are parameters obtained from measurements in the engine lab. The controller’s feedforward component enables a quick response to changes such as a rapid acceleration when the accelerator pedal is depressed fully onto the floor.

The feedback component of the controller $\alpha_{thFB}$ is obtained as

$$ \alpha_{thFB} = K_{p,th} \cdot \hat{e}_{im} + K_{i,th} \cdot \int \hat{e}_{im} + K_{d,th} (\alpha_{thREF}_{SAT} - \alpha_{thREF}) \, dt, \quad (17) $$

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

$$ \alpha_{thREF}_{SAT} = \begin{cases} \alpha_{thMAX}, & \text{if } \alpha_{thREF} > \alpha_{thMAX} \\ \alpha_{thREF}, & \text{if } \alpha_{thMIN} < \alpha_{thREF} < \alpha_{thMAX} \\ \alpha_{thMIN}, & \text{if } \alpha_{thREF} < \alpha_{thMIN} \end{cases} \quad (18) $$

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

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

$$ u_{wgFB} = K_{p,wg} \cdot \hat{e}_{ic} + K_{i,wg} \cdot \int \hat{e}_{ic} + K_{d,wg} (u_{wgREF}_{SAT} - u_{wgREF}) \, dt, \quad (19) $$

where $K_{p,wg}$ and $K_{i,wg}$ are the proportional and integral gains of the feedback controller, respectively, and $\hat{e}_{ic} = p_{ic} - p_{wgREF}$. The saturation of the reference wastegate input $u_{wgREF}_{SAT}$ is defined as the static nonlinearity

$$ u_{wgREF}_{SAT} = \begin{cases} u_{wgMAX}, & \text{if } u_{wgREF} > u_{wgMAX} \\ u_{wgREF}, & \text{if } u_{wgMIN} < u_{wgREF} < u_{wgMAX} \\ u_{wgMIN}, & \text{if } u_{wgREF} < u_{wgMIN} \end{cases} \quad (20) $$

The design of the controller for the engine is then verified in simulations using the reference engine torque $T_{Qe,REF}$ and speed $\omega_{REF}$ generated from a selected driving cycle. Figure 4 provides 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 highly nonlinear (with many interconnected subsystems where the airflow 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 are compared with those of the actual test bench system in the lab, which are fully controlled in real time and have sensor measurements that are visualized and recorded by using a dSPACE MicroAutoBox + RapidPro

Structural analysis is a useful tool for the early determination of fault isolability.

formatting corrections:
system as well as integrated calibration and application tools. The sensor measurements that are recorded for comparison between the model and test bench are the air filter mass flow \( y_{Waf} \), intercooler temperature \( y_{Tic} \), intake manifold pressure \( y_{pim} \) and intercooler pressure \( y_{pic} \). During the EUDC run, the engine model produces sensor measurements that are close to the actual test bench system. These results show that the model is realistic and accurately represents the actual engine system (see Figure 5).

### FAULT SCENARIOS

The simulation testbed considers 11 sensor, actuator, and variable faults of different degrees of severity in different parts of the engine system: six variable faults \( f_{Waf}, f_{Cvol}, f_{Waf}, f_{Wth}, f_{Wc} \) and \( f_{Wic} \), one actuator measurement fault \( f_{xth} \), and four sensor measurement faults \( f_{yWaf}, f_{ypim}, f_{ypic} \) and \( f_{yTic} \). Some faults are less severe, and the engine can be reconfigured to a reduced performance operation mode to accommodate them until the vehicle is sent for repairs. Other faults are severe enough that they might cause permanent damage if they are not detected and isolated promptly, which, in turn, could endanger the vehicle occupants as well as other road users.

### Fault Types and Classification

The faults included in this simulation testbed can be categorized into three types: sensor, actuator, and variable.

#### Sensor Faults

This research considers four sensor measurement faults: \( f_{Waf}, f_{Cvol}, f_{Waf} \) and \( f_{Wth} \). They occur because of electrical and mechanical errors that lead to an offset or deviation in the sensor measurements.

The \( f_{Waf} \) 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 to be replaced as soon as the fault is detected.

The remaining sensor faults are pressure and temperature measurement errors in the engine’s intercooler (\( f_{ypic} \)) and intake manifold (\( f_{ypim} \)). The pressure measurement...
errors ($f_{in}$ and $f_{ex}$) produce a 20% deviation in the measured values. The $f_{in}$ is modeled by using a long-term incipient fault to indicate a drift in the sensor signal through time, while the $f_{ex}$ is modeled through repeating abrupt pulses. The $f_{in}$ indicates an offset in the sensor that measures the temperature in the intercooler. This fault is also modeled by using repeating abrupt pulses.

**Actuator Faults**

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

### Modeling a Mass Flow Fault Caused by a Leak

The intensity of a mass flow fault caused by a leak is determined by the area of the orifice, which is usually measured in square millimeters. Larger leak areas increase the mass flow and pressure difference between both sides of the orifice (hence, the higher intensity of the fault). Conventionally, a mass flow fault is induced by drilling a hole in a specific component of the engine test bench system. Tests are performed by running the engine through driving cycles, and the fault's effects are analyzed. The leak orifice is then sealed by a screw plug to disable the fault and return the engine to its nominal operation mode. Simulating the mass flow fault through a testbed in Matlab/Simulink removes the need for the engine to be physically modified (which could lead to irreversible damage).

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

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

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

$$\Psi \left( \frac{p_{\text{low}}}{p_{\text{high}}} \right) = \begin{cases} \frac{2k}{k - 1} \left( \frac{p_{\text{low}}}{p_{\text{high}}} \right)^{2k} - 1, & \text{if } \frac{p_{\text{low}}}{p_{\text{high}}} \leq \frac{2}{k + 1}, \\ \frac{1}{k} \left( \frac{p_{\text{low}}}{p_{\text{high}}} \right)^{k} - 1, & \text{otherwise,} \end{cases}$$

where $k$ is the specific heat ratio in the affected part of the engine.

### TABLE 2

| Fault    | Description                                      | Fault Threshold         | Nature of Fault (Active Period)          | Severity |
|----------|--------------------------------------------------|-------------------------|------------------------------------------|----------|
| $f_{in}$ | Loss of pressure in the air filter               | 20-kPa pressure drop    | Abrupt (from 200 s until $T_{DC}$)       | Medium   |
| $f_{out}$ | Intake-valve timing stuck at an arbitrary position | Stuck at end or middle position | Abrupt pulses (active for 30 s every 150 s) | High     |
| $f_{W}$  | Air leakage between the air filter and the compressor | 20% of flow through leakage | Incipient (from 200 s until $T_{DC}$)     | Medium   |
| $f_{W}$  | Air leakage between the compressor and the intercooler | 20% of flow through leakage | Abrupt (from 0.4$T_{DC}$ until $T_{DC}$) | High     |
| $f_{W}$  | Air leakage between the intercooler and the throttle | 20% of flow through leakage | Abrupt (from 0.4$T_{DC}$ until 0.8$T_{DC}$) | High     |
| $f_{W}$  | Air leakage after the throttle in the intake manifold | 20% of flow through leakage | Abrupt pulses (active for 40 s every 200 s) | High     |
| $f_{X}$  | Throttle position actuator error                 | Fault leading to 20% flow error | Abrupt (from 0.4$T_{DC}$ until $T_{DC}$) | Medium   |
| $f_{in}$ | Air filter flow sensor fault                     | 20% flow error          | Abrupt pulses (active for 30 s every 150 s) | Low      |
| $f_{in}$ | Intake manifold pressure sensor fault            | 20% pressure deviation   | Incipient (from 200 s until $T_{DC}$)    | Low      |
| $f_{in}$ | Intake manifold pressure sensor fault            | 20% pressure deviation   | Abrupt pulses (active for 40 s every 200 s) | Low      |
| $f_{in}$ | Intake manifold pressure sensor fault            | 20% pressure deviation   | Abrupt pulses (active for 30 s every 150 s) | Low      |
| $f_{in}$ | Intake manifold temperature sensor fault         | 20-K offset             | Abrupt pulses (active for 30 s every 150 s) | Low      |
Variable Faults

This research considers six variable faults: \( f_{\text{paf}} \), \( f_{\text{Cvol}} \), \( f_{\text{Waf}} \), \( f_{\text{Wth}} \), \( f_{\text{Wc}} \), and \( f_{\text{Wic}} \). These faults are due to physical and mechanical damage that produce pressure drops, leaks, and degraded performance. As a result, most of these faults are of a high severity.

The \( f_{\text{paf}} \) fault indicates a pressure drop in the air filter due to a flow restriction. It is modeled by using a long-term abrupt fault to simulate a constant restrictive flow in the air filter. The fault \( f_{\text{Cvol}} \) indicates that the intake-valve timing actuator is stuck at an arbitrary position. This affects the engine’s volumetric efficiency and thus its overall performance, 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 must be quickly detected and isolated. It is modeled by repeating abrupt pulses.

The remaining variable faults are leaks that could occur in different parts of the engine system, including the air filter \( (f_{\text{Waf}}) \), compressor \( (f_{\text{Wc}}) \), intercooler \( (f_{\text{Wic}}) \), and throttle

![Figure 6](image)

**Figure 6** The fault isolation matrix based on the structural model of the engine system. The figure shows that there are two pairs of faults that cannot be isolated: \((f_{\text{paf}}, f_{\text{th}})\) and \((f_{\text{th}}, f_{\text{th}})\). This is an ideal result for fault isolation, assuming that there are no limits to the magnitudes and shapes of the faults.

![Figure 7](image)

**Figure 7** A 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 from Figure 3, while the subsystem within the red dashed box is the residuals generator. The locations where the faults are induced are also shown.

**TABLE 3** The default residuals (“Original 9”) for fault detection given the sensor setup in Table 1.

| Residual | Description                                      |
|----------|--------------------------------------------------|
| \( r_{\text{Tc}} \) | Residual for the compressor temperature sensor  |
| \( r_{\text{pc}} \) | Residual for the compressor pressure sensor      |
| \( r_{\text{Tic}} \) | Residual for the intercooler temperature sensor  |
| \( r_{\text{pinc}} \) | Residual for the intercooler pressure sensor     |
| \( r_{\text{Tm}} \) | Residual for the intake manifold temperature sensor |
| \( r_{\text{pm}} \) | Residual for the intake manifold pressure sensor  |
| \( r_{\text{Waf}} \) | Residual for the air filter mass flow sensor     |
| \( r_{\text{Te}} \) | Residual for the engine torque sensor            |
| \( r_{\text{pem}} \) | Residual for the exhaust manifold pressure sensor |
These leaks, in the form of varying-diameter orifices, lead to a change in the mass flow. Other than the $f_{\text{MC}}$ (which is of medium severity), the mass flow faults are of high severity, as they occur after the compressor and closer to the engine block (where the pressure is higher). Unattended mass flow faults in these components could cause a degradation in the engine performance (such as overpressure and increased emissions) as well as damage to the components themselves, especially if external abrasive particulate matters manage to enter the engine. This could lead to faults in other parts of the engine system, thus making efforts to isolate the original leak fault more difficult (see “Modeling a Mass Flow Fault Caused by a Leak”). The faults and their characteristics are summarized in Table 2.

### Fault Isolation Analysis From a Model

Using the differential equations in “The Differential Equations of the Turbocharged Spark-Ignited Engine System and Corresponding Engine Parameters,” a structural model of the engine system with the faults defined in Table 2 is constructed. The structural model shows the relationships among the engine system’s unknown variables, known variables (actuators and sensors), and measurable outputs $i, \omega, \Delta T$, and $f_i$. It is assumed that there are no faults in these measurements.

| Fault | $V = i(R + f_i) + L \frac{di}{dt} + K_a \omega_i$ | $T_m = K_a \omega_i$ | $\Delta T = T_m - T_i$ | $\frac{d\omega}{dt} = \alpha_i$ | $y_i = i + f_i$ |
|-------|-------------------------------------------------|------------------|-----------------|---------------------|--------------|
| $e_1$ | $e_2$ | $e_3$ | $e_4$ | $e_5$ | $e_6$ | $e_7$ | $e_8$ | $e_9$ |
| X | X | X | X | X | X | X | X | X |

Where $V$ is the input voltage, $R$ is the resistance, $L$ is the inductance, and $i$ is the current in the armature circuit. On the mechanical side of the system, $T_m$ is the motor torque, $T_i$ is the load’s torque, $J$ is the rotor’s moment of inertia, $K_a$ is the motor-torque constant, and $K_b$ is the back electromotive-force constant. The rotational displacement of the motor is $\theta$, while $\omega$ and $\alpha$ are the rotational velocity and acceleration, respectively. The states of the system are $i, \theta, \omega, \alpha, T_m, T_i, \Delta T$, and the measurable outputs are $i, \omega, \Delta T$. It is assumed that there are no faults in these measurements.

### The Structural Model and Fault Isolation Matrix: A General Tutorial

Let’s consider the dc motor system in Figure S1. It is modeled by using the following equations:

1. $e_1: V = i(R + f_i) + L \frac{di}{dt} + K_a \omega_i$ (S3)
2. $e_2: T_m = K_a \omega_i$ (S4)
3. $e_3: J \frac{d\omega}{dt} = \Delta T - K_b \omega_i$ (S5)
4. $e_4: \Delta T = T_m - T_i$ (S6)
5. $e_5: \frac{d\omega}{dt} = \alpha_i$ (S7)
6. $e_6: \frac{d\omega}{dt} = \alpha_i$ (S8)
7. $e_7: y_i = i + f_i$ (S9)

Where $V$ is the input voltage, $R$ is the resistance, $L$ is the inductance, and $i$ is the current in the armature circuit. On the mechanical side of the system, $T_m$ is the motor torque, $T_i$ is the load’s torque, $J$ is the rotor’s moment of inertia, $K_a$ is the motor-torque constant, and $K_b$ is the back electromotive-force constant. The rotational displacement of the motor is $\theta$, while $\omega$ and $\alpha$ are the rotational velocity and acceleration, respectively. The states of the system are $i, \theta, \omega, \alpha, T_m, T_i, \Delta T$, and the measurable outputs are $i, \omega, \Delta T$. It is assumed that there are no faults in these measurements.

### Table S5

The structural model of the dc-motor system as created using (S3)–(S11). 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.
faults. Structural analysis is a useful tool for the early determination of fault isolability, which incorporates different levels of knowledge and results in various conclusions [21].

Using the structural model, the FI matrix (FIM) is generated for an initial FI analysis. The FIM is a square matrix where each row and column corresponds to a fault. A dot is placed at a 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. Two pairs of faults that cannot be isolated can be observed; \(f_{pa}\) is not isolable from \(f_W\), and \(f_{Wd}\) is not isolable from \(f_{xth}\). However, this is a best-case performance of FI 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 FI capability. The simulation testbed (with the bounded magnitudes of the faults and consideration of sensor noise) provides a more realistic outlook on the fault isolability for the engine system. See “The Structural Model and Fault Isolation Matrix: A General Tutorial” for more about the structural model and FIM. Further information on the studies and development of the structural model and FIM can be found in [21].

**DESIGN AND GENERATION OF RESIDUALS**

**Introduction**

Initially, nine residuals are generated based on the sensor setup described in Table 1. The simulation testbed is by default distributed with a state estimator/observer, which is constructed by using the differential equations that describe the engine system. As such, the estimator/observer provides an estimate of the internal states of the engine system. The design of the observer can be replaced by other types of observers that are found in the literature, such as the sliding-mode observer [22], [23], Kalman filter [24], [25], and reduced-order observer [26], [27]. Therefore, is a system fault, \(f_n\), representing inconsistency in the value of the resistance, \(R\). It is also assumed that all outputs are potentially faulty through \(f_i\), \(f_{\omega}\), and \(f_n\), respectively, as shown in (S9)–(S11).

Using a 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 \(\omega\) (as well as the fault \(f_n\)) are used in equation \(e_5\) in (S3) to express the input voltage, \(V\).

By performing canonical decomposition onto the unknown variables in Table S5, the structural model can be remodeled as Table S6. The fault isolation matrix is obtained by extracting the bottom-right section of the structural model in Table S6. It can be seen that for the dc-motor system, the pair \((h, l)\) is not isolable. See Table S7.
this simulation testbed also enables researchers to design, develop, and compare strategies for designing residuals generators for state estimation and fault diagnosis applications in automotive engine systems. The residuals are generated by computing the difference between the sensor outputs of the model in Figure 3 and 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 gives 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:

\[
 r_{(N)} = \frac{r_i - \mu_r}{\sigma_{\text{NOMINAL}}}, \tag{21}
\]

where \( r_{(N)} \) is the normalized residual, \( \mu_r \) is the mean of the residual, and \( \sigma_{\text{NOMINAL}} \) is the standard deviation of the corresponding residual during a fault-free scenario. These normalized residuals are called the “Original 9” and listed in Table 3.

In a nominal fault-free scenario, all of the residuals have zero-mean values. This indicates that the model and

![FIGURE 8](image_url)  
**FIGURE 8** The normalized plots of the “Original 9” for a fault-free scenario. All residuals have zero-mean values, which indicates that 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 represent the fault detection thresholds to determine if the residuals have triggered and a fault found.
engine estimator produce similar actual and estimated outputs, respectively, while being excited by the same control inputs. Figure 8 conveys 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, \(|r_{\text{in}}| > J\) ) 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 tradeoff between the 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.

The engine control system and residuals generator are then simulated with the faults in Table 2. The residuals sensitive to the corresponding faults will trigger and produce nonzero mean values. Figures 9–19 show the simulation results for the “Original 9” residuals during the WLTP driving cycle profile when the faults are introduced into the engine system. Only single-fault scenarios are currently considered. The figures show that the dynamics of the engine system and nature of the faults (that is, if they are of the actuator, sensor,
or variable type and induced as an abrupt or incipient occurrence) influence the corresponding residuals, causing them to exceed the threshold and trigger. For example, all of the sensor faults \( f_{\text{yp}} \), \( f_{\text{ypim}} \), \( f_{\text{ync}} \), and \( f_{\text{ypa}} \) triggered only one residual each, as they did not directly affect the states of the engine system (see Figures 16–19). However, they could still affect the system indirectly if they were used as feedback signals. Since the actuator and variable faults directly affect the dynamics of the engine system, more residuals are sensitive to them. Therefore, if they are detected, it is usually easier to isolate sensor faults than variable faults. By collectively identifying which residuals were triggered for the faults induced, FI analysis can be performed to locate the fault in the engine system.

During simulations of faults in real-world conditions (especially for nonrepetitive 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 and permanent faults (such as a gradual restricted-pressure increase in the air filter, \( f_{\text{paf}} \)), the residuals might exhibit occasional spikes. The 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 affect the outlook for the residuals that were generated, and, hence,

**FIGURE 10** The normalized plots of the “Original 9” for a stuck intake valve timing, \( f_{\text{Cvol}} \). This fault triggered all of the residuals. The shaded regions show the duration for which the faults were active.
they must be considered 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 to perform fault detection is based on the amplitude of the residuals (that is, if it exceeds the threshold $J$) as well as the duration for which it remains above the threshold. For the simulation testbed, the fault should be detected if the residuals exceed the threshold for the duration of $t_f > 3\, \text{s}$.

- **Missed detections**: This testbed is designed so that the amplitudes of the faults are large enough that they should be detected.

Fault Isolation Analysis From Simulations
The fault sensitivity matrix (FSM) in Table 4 can be constructed from the simulation results shown in Figures 9–19. It is tabulated by placing a value of one if the residual is triggered by the specific fault and zero otherwise. Using the FSM in Table 4, the FIM of the system for the current residuals design can be constructed. Figure 21 shows the FIM with a more realistic FI performance when the magnitudes and shapes of the faults acting on the engine system...
are also considered. See Figure 6 for a 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 create and perform model-in-the-loop tests of fault diagnosis schemes, with an application to actual automotive engine systems.

THE SIMULATION ENVIRONMENT
Figure 22 shows the GUI of the simulation testbed in Matlab. Through this interface, users can set their preferences for the simulation settings, design, and testing of their residuals generation and fault diagnosis schemes. They can also view the simulation results.

Establishing Simulation Settings
In the left section of the GUI, there are pop-up menus for users to set key simulation settings, which include the following:

- **Fault mode**: Users can induce any of the 11 faults defined in Table 2. A fault-free scenario is also available and selected by default. Currently, only single-fault scenarios are available.
- **Driving cycle**: A selection of four industrial-standard driving cycles (WLTP, NEDC, EUDC, and FTP-75) is available.
- **Simulation mode**: There is a choice of two modes, which simulate only the engine for the chosen driving cycle or extend the simulation to include residuals generation and the execution of the fault diagnosis algorithm.

FIGURE 12 The normalized plots of the “Original 9” for an air leakage between the compressor and intercooler $f_{w}$. This fault triggered all of the residuals. The shaded regions show the duration for which the faults were active.
The latter choice would require design and coding inputs from the user.

**Design and Testing of Residuals Generation and Fault Diagnosis Schemes**

A block-diagram representation of the engine control system, residuals generator, and fault diagnosis scheme can be found in the top-right section of the GUI. Users can select 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” button starts the simulation, while the “EXIT” button closes the simulation environment and GUI.

**Simulation Results**

The simulation results are displayed in the bottom-right section of the GUI. They include the reference and actual engine torques as well as the normalized plot of the induced fault. A “Simulation Log” is also available in the bottom-left section of the GUI to show a summary of the settings and provide a progress update in real time. The plots and “Simulation Log” are automatically saved to the folder/Results/DrivingCycle_FaultMode_Date, which is located in the same directory as the simulation files. A Matlab MAT-file containing key variables and

![Normalized plots](image-url)
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 in the “Simulation Log” using the `PrintLog()` command.

**The Simulation Kit**
The simulation kit is available as open-source software and can be downloaded from https://github.com/nkymark/TCSiSimTestbed. It contains the following key files:

» `main.m`: This is the main execution file. Run it to open the GUI.

» `Engine.mdl`: This shows the Simulink model of the closed-loop nonlinear engine system. Open the model from the GUI using the “Boost Controller (Simulink)” or “Engine System (Simulink)” blocks.

» `GenerateResiduals.m`: The codes for the residuals generation algorithm are placed here. Open the file from the GUI using the “Residuals Generator Design (m-file)” button.

» `ResidualsGen.mdl`: This is the Simulink model of the residuals generator. The model is called and run from `GenerateResiduals.m`. The default residuals that are generated are also filtered and normalized and with added signal noise. Open the model

![Normalized plots of the “Original 9” for an air leakage after the throttle in the intake manifold, \( f_{\text{W}} \). The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are residuals not sensitive to the fault. This fault triggered all of the residuals except \( r_{\text{Tc}} \), \( r_{\text{Tic}} \), and \( r_{\text{pem}} \). The shaded regions show the duration for which the faults were active.](image)

**FIGURE 14** The normalized plots of the “Original 9” for an air leakage after the throttle in the intake manifold, \( f_{\text{W}} \). The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are residuals not sensitive to the fault. This fault triggered all of the residuals except \( r_{\text{Tc}} \), \( r_{\text{Tic}} \), and \( r_{\text{pem}} \). The shaded regions show the duration for which the faults were active.
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: The algorithm for the fault diagnosis is placed here. Open the file from the GUI using the “Fault Isolation Scheme Design (m-file)” block.

CONCLUSION
This article presented a simulation testbed for evaluating residuals generation and fault diagnosis schemes in a TCSI gasoline engine system. Key features of the simulation testbed were emphasized.

- It includes a realistic nonlinear model of the engine system compared with the physical test bench.
- The testbed enables researchers to simulate actuator, sensor, and variable faults in various components of the engine system without having to physically modify the test bench.
- Researchers are able to compare the performance of their fault diagnosis schemes against the presented structural model and FIM benchmark.
- General simulation and fault settings can be easily configured using the GUI, and the testbed can be customized to accommodate different residuals generation as well as fault diagnosis schemes.

FIGURE 15 The normalized plots of the “Original 9” for throttle position actuator error $f_{th}$. The plots in red are the residuals sensitive to the fault and triggered, while plots in blue are the residuals not sensitive to the fault. This fault triggered only the $r_{Wd}$ residual. The shaded regions show the duration for which the fault was active.
The simulation kit is available as open-source software and can be downloaded for research and/or teaching purposes.

The data generated from the simulation testbed are suitable for the study of model-based and data-driven fault diagnosis methods. This testbed will serve as an excellent platform to demonstrate the effectiveness of 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 will include the activation of intermittent residuals to mimic applications where some residuals are turned off during certain driving conditions (such as rough terrain and extreme weather) so that they do not trigger false alarms. The addition of faults in other parts of the engine and new simulation options, including weather, will also be considered.

Some of the interesting research challenges in this field of study include 1) the issue of robustness (which will always be one of the critical problems for any control system) and since most automakers sell their vehicles all across the world, it is very difficult for one fault diagnosis method to remain resilient against a variety of terrains, weather, driving styles, and traffic conditions; 2) with the

![Figure 16](image.png)

**Figure 16** The normalized plots of the “Original 9” for an intercooler pressure sensor fault $f_{pco}$. The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are the residuals not sensitive to the fault. This fault triggered only the $r_{pco}$ residual. The shaded regions show the duration for which the fault was active.
ever-increasing development of autonomous vehicles, it is important for systems to be aware of their health and perform self-diagnosis and self-healing to ensure that occupants’ lives are protected at all times; and 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 contributes data to a general pool in the cloud.

ACKNOWLEDGMENT
This research was supported by Volvo Cars, Gothenburg, Sweden.

AUTHOR INFORMATION
Kok Yew Ng (mark.ng@ulster.ac.uk) is a lecturer in the School of Engineering, Ulster University, United Kingdom, where he conducts research at the Nanotechnology and Integrated Bioengineering Centre. He is also an adjunct senior research fellow in the School of Engineering, Monash University, Malaysia. He received the B.E. degree (with honors) in electrical and computer systems engineering and the Ph.D. degree from Monash University in 2006 and 2009, respectively. From 2014 to 2015, he was a postdoctoral researcher at the Division of Vehicular Systems, Linköping University, Sweden, where he focused

FIGURE 17 The normalized plots of the “Original 9” for an intake manifold pressure sensor fault $f_{\text{pm}}$. The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are the residuals not sensitive to the fault. This fault triggered only the $f_{\text{pm}}$ residual. The shaded regions show the duration for which the fault was active.
on the design and development of advanced fault diagnosis schemes using model-based and data-driven methods on vehicular engines in conjunction with Volvo Cars. His research interests include fault diagnosis, sliding-mode observation, vehicular systems, and data analytics for anomaly detection and classification.

Erik Frisk received the Ph.D. degree from Linköping University, Sweden, in 2001. He is a professor in the Department of Electrical Engineering, Linköping University. His research interests include fault diagnosis and isolation, prognostics using model-based techniques and data-driven approaches, and the control of autonomous vehicles.

Mattias Krysander is an associate professor in the Department of Electrical Engineering, Linköping University,

Table 4: The fault sensitivity matrix of the “Original 9” residuals.

| Residual | $f_{p_{i,t}}$ | $f_{c_{i,t}}$ | $f_{w_{i,t}}$ | $f_{p_{i,t}}$ | $f_{w_{i,t}}$ | $f_{p_{i,t}}$ | $f_{w_{i,t}}$ | $f_{p_{i,t}}$ | $f_{w_{i,t}}$ |
|----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| $r_{T_{c}}$ | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| $r_{p_{w}}$ | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| $r_{w_{c}}$ | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| $r_{p_{m}}$ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| $r_{w_{m}}$ | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| $r_{p_{v_{w}}}$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| $r_{w_{v_{w}}}$ | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| $r_{p_{w_{w}}}$ | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |

Figure 18: The normalized plots of the “Original 9” for an intercooler temperature sensor fault, $r_{T_{c}}$. The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are the residuals not sensitive to the fault. This fault triggered only the $r_{T_{c}}$ residual. The shaded regions show the duration for which the fault was active.
FIGURE 19 The normalized plots of the “Original 9” for an air filter flow sensor fault, \( r_{Waf} \). The plots in red are the residuals sensitive to the fault and triggered, while the plots in blue are the residuals not sensitive to the fault. This fault triggered only the \( r_{Waf} \) residual. The shaded regions show the duration for which the fault was active.

### TABLE 5
The variables saved into the MAT-file after each simulation run. A user could employ these data for further processing and analysis toward the design of the fault diagnosis scheme.

| Saved Variable    | Description                                                                 |
|-------------------|-----------------------------------------------------------------------------|
| \( \text{omega}_\text{eREF}_\text{sync} \) | Reference engine speed, \( \omega_{\text{eREF}} \)                           |
| \( \text{Tq}_\text{eREF}_\text{sync} \)    | Reference engine torque, \( Tq_{\text{eREF}} \)                           |
| \( \text{inputSig}_\text{sync} \)         | Five actuator measurements of the engine \( (A_{\text{th}}, u_{\text{eq}}, \omega_{\text{eREF}}, p_{\text{amb}}, \text{and } T_{\text{amb}}) \) |
| \( \text{outputSig}_\text{sync} \)        | Nine sensor measurements from the engine \( (T_c, p_c, T_c, p_c, T_m, p_m, \text{and } T_{\text{qs}}) \) |
| \( \text{statesSig}_\text{sync} \)        | Thirteen states of the engine \( (T_{\text{ds}}, p_{\text{ds}}, T_{\text{c}}, p_{\text{c}}, T_{\text{cs}}, p_{\text{cs}}, T_{\text{ms}}, p_{\text{ms}}) \) |
| \( \text{faultSig}_\text{sync} \)         | Normalized data of the faults (the selected induced fault would have nonzero data, except when the “fault-free” scenario is selected, where all faults would have a data of value zero) |
| \( \text{residualSig}_\text{sync} \)       | Data for all of the “Original 9” residuals based on the current sensors setup \( (r_{Tc}, r_{pc}, r_{Tic}, r_{pim}, r_{Waf}, r_{Tqe}, r_{pem}, r_{Tqm}, \text{and } r_{pew}) \). Note that these data are only generated if simulation mode 2 is selected. |
FIGURE 20 The $r_{\text{fp}}$ residual signal generated (amber line) for an $f_{\text{fp}}$ fault (red line), with the engine torque (blue line), during a Worldwide Harmonized Light Vehicle Test Procedure driving cycle profile. All of the signals were normalized to an interval of [0, 1].

FIGURE 21 The fault isolation matrix based on the fault sensitivity matrix in Table 4. This is a more realistic representation of the fault isolation analysis, as it considers the magnitudes and shapes of the faults.

FIGURE 22 The main graphical user interface of the simulation testbed in Matlab. 1. Set the fault mode for simulation. 2. Set the driving cycle. 3. Set the simulation mode. 4. Run the simulation. 5. Exit and close the testbed graphical user interface. 6. Show the simulation progress and log. 7. Select the reference generator Simulink model. 8 and 9. Select the boost controller and engine Simulink model. 10. Select the residuals generator Simulink model. 11. Select the fault diagnosis design scheme m-file. 12. Select the residuals generator design scheme m-file. 13. Display the residuals that were generated. 14. Display the reference torque and actual torque of the engine. 15. Display the fault signal that was induced (normalized).
Sweden. His research interests include model-based and data-driven diagnosis and prognosis. To address the complexity and size of industrial systems (mainly vehicle systems), he has used structural representations of models and developed graph theoretical methods for assisting the design of diagnosis systems and fault isolation and sensor placement analysis.

**Lars Eriksson** is a professor of vehicular systems in the Department of Vehicular Systems, Linköping University, Sweden, where he manages the engine laboratory. He received the M.S. degree in electrical engineering and the Ph.D. degree in vehicular systems from Linköping University in 1995 and 1999, respectively. From 2000 to 2001, he was a postdoctoral researcher in the Measurement and Control Group at the Swiss Federal Institute of Technology, Zurich, Switzerland. His research interests include the modeling, simulation, and control of internal combustion engines for vehicle propulsion, with a focus on downsizing and supercharging concepts for improved fuel economy. His foremost contributions were to engine control and the control-oriented modeling of combustion engines; he was the first to demonstrate the real-time control of combustion fidelity, engine-in-vehicle simulation of the International Class VI truck, and simulation package for electric and hybrid electric vehicle design.

**REFERENCES**

[1] J. Gerler, M. Costin, X. Fang, Z. Kowalczyk, M. Kunwer, and R. Monajemy, “Model based diagnosis for automotive engines-algorithm development and testing on a production vehicle,” *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 1, pp. 16–69, Mar. 2005. doi: 10.1109/TCST.2004.817873.

[2] S. Kher, P. K. Chande, and P. C. Sharma, “Automobile engine fault diagnosis using neural network,” in *IEEE Intelligent Transportation Systems Proc. (ITSC 2001)*, Oakland, CA, Aug. 2001, pp. 492–495. doi: 10.1109/ITSC.2001.948707.

[3] M. Nyberg, “Model-based diagnosis of an automotive engine using several types of fault models,” *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 5, pp. 679–689, Sept. 2002. doi: 10.1109/TCST.2002.803873.

[4] Y. L. Murphy, J. A. Crossman, Z. Chen, and J. Cardillo, “Automotive fault diagnosis—Part II: A distributed agent diagnostic system,” *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 1076–1098, July 2003. doi: 10.1109/TVT.2003.814236.

[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 Langley Research Center, Hampton, VA, Tech. Rep. NASA/NR-2010-216724, 2010.

[7] A. Scacchioli, G. Rizzoni, and P. Pisu, “Model-based fault detection and isolation in automotive electrical systems,” in *Proc. ASME Int. Mechanical Engineering Congr. and Exposition*. Chicago, IL: ASME, Nov. 2006, pp. 315–324. doi: 10.1115/IMECE2006-14504.

[8] P. Weber, S. Gentil, P. Ripoll, and L. Foulloy, “Multiple fault detection and isolation,” in *Proc. 14th IFAC World Congr.*, Beijing, 1999. pp. 223–228. doi: 10.1016/S1474-6670(17)57348-6.

[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 Syst. Appl.*, vol. 39, no. 12, pp. 10,977–10,989, 2012. doi: 10.1016/j.eswa.2012.03.026.

[10] J. Tang, G. J. Kaczprzynski, K. Goebel, A. Saxena, B. Saha, and G. Vachtsevans, “Prognostics-enhanced automated contingency management for advanced autonomous systems,” in *Proc. 2008 Int. Conf. Prognostics and Health Management*, Denver, CO, Oct. 2008, pp. 1–9. doi: 10.1109/PHM.2008.471448.

[11] R. Loureiro, R. Merzouki, and B. Ould-Bouamaama, “Bond graph model based on structural diagnosability and recoverability analysis: Application to intelligent autonomous vehicles,” *IEEE Trans. Veh. Technol.*, vol. 61, no. 3, pp. 986–997, Mar. 2012. doi: 10.1109/TVT.2012.2186472.

[12] R. Loureiro, S. Benmoussa, Y. Touati, R. Merzouki, and B. Ould-Bouamaama, “Integration of fault diagnosis and fault-tolerant control for health monitoring of a class of MIMO intelligent autonomous vehicles,” *IEEE Trans. Veh. Technol.*, vol. 63, no. 1, pp. 30–39, Jan. 2014. doi: 10.1109/TVT.2013.2274289.

[13] R. Isermann, Schaffitin, and S. Sinsel, “Hardware-in-the-loop simulation for the design and testing of engine-control systems,” *Control Eng. Pract.*, vol. 7, no. 5, pp. 643–653, 1999. doi: 10.1016/S0967-0661(98)00205-6.

[14] K. L. Butler, M. Ehsani, and P. Kamath, “A Matlab-based modeling and simulation package for electric and hybrid electric vehicle design,” *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1770–1778, Nov. 1999. doi: 10.1109/TVT.1999.790679.

[15] D. Assanis et al., “Validation and use of SIMULINK integrated, high fidelity, engine-in-vehicle simulation of the International Class VI truck,” *SAE Trans.*, vol. 109, pp. 384–399, Jan. 2000. doi: 10.4271/2000-01-0288.

[16] W. Lee, M. Yoon, and M. Sunwoo, “A cost- and time-effective hardware-in-the-loop simulation platform for automotive engine control systems,” *Proc. Inst. Mech. Eng. D, J. Automob. Eng.*, vol. 217, no. 1, pp. 41–52, 2003. doi: 10.1243/095440703762702969.

[17] M. Yoon, W. Lee, and M. Sunwoo, “Development and implementation of distributed hardware-in-the-loop simulator for automotive engine control systems,” *Int. J. Automot. Technol.*, 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 CO₂ emission standards,” International Council on Clean Transportation, Washington, D.C., White Paper, Sept. 2014.

[19] L. Eriksson, “Modeling and control of turbocharged SI and DI engines,” *Oil Gas Sci. Technol. - Rev. IFP*, vol. 62, no. 4, pp. 523–538, 2007. doi: 10.2516/osx/o07042.

[20] L. Eriksson and L. Nielsen, *Modeling and Control of Engines and Drivelines*. Hoboken, NJ: Wiley, 2014.

[21] D. Deger, E. Frisk, V. Coquempot, M. Kryssander, and M. Staroswiecki, “Structured analysis of fault isolability in the DAMADICS benchmark,” *Control Eng. Pract.*, vol. 14, no. 6, pp. 597–608, 2006. doi: 10.1016/j.conengprac.2005.04.008.

[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. doi: 10.1016/S0005-1098(99)00177-6.

[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. doi: 10.1016/j.automatica.2012.02.005.

[24] S. Simani, C. Fantuzzi, and S. Beghelli, “Diagnosis techniques for sensor faults of industrial processes,” *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 5, pp. 848–855, Sept. 2000. doi: 10.1109/87.865858.

[25] T. Kobayashi and D. L. Simon, “Application of a bank of Kalman filters for aircraft engine fault diagnostics,” in *Proc. ASME Turbo Expo 2003*, Atlanta, GA: ASME, pp. 461–470, 2003. doi: 10.1115/GT2003-38580.

[26] M. Darouch, M. Zasadzinski, and M. Hayar, “Reduced-order observer design for descriptor systems with unknown inputs,” *IEEE Trans. Autom. Control*, vol. 41, no. 7, pp. 1068–1072, July 1996. doi: 10.1109/9.590891.

[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. doi: 10.1016/S0005-1098(97)00175-1.