



Digital twin-enabled autonomous fault mitigation in diesel engines: An experimental validation

Raj Pradip Khawale^a, Dhrubajit Chowdhury^b, Raman Goyal^b, Shubhendu Kumar Singh^a, Ankur Bhatt^a, Brian Gainey^a, Benjamin Lawler^a, Lara Crawford^b, Rahul Rai^{a,c,*}

^a Department of Automotive Engineering, Clemson University, Greenville, 29607, SC, USA

^b PARC, part of SRI International, Palo Alto, 94304, CA, USA

^c Department of Computer Science, Clemson University, Greenville, 29631, SC, USA

ARTICLE INFO

Keywords:

Fault tolerant control
Diesel engine digital twin
Neural networks
Hierarchical control
Hardware-in-loop
Autonomous system

ABSTRACT

Due to the growing demand for robust autonomous systems, automating maintenance and fault mitigation activities has become essential. If an unexpected fault occurs during the travel, the system should be able to manage that fault autonomously and continue its mission. Thus, a robust fault mitigation system is needed that can quickly reconfigure itself in an optimal way. This paper presents a novel digital twin-based fault mitigation strategy that uses hierarchical control architecture. Here, a computationally efficient high-fidelity hybrid engine model is developed to simulate actual engine behavior. This hybrid engine model includes a neural network model representing the cylinder combustion process and well-studied physics-based analytical equations describing the remaining subsystems. This architecture uses a feedback controller on top of the control calibration map, generated offline using the hybrid model, to mitigate faults and modeling errors. The fault mitigation strategies are calibrated and validated through model-in-loop (MIL) and hardware-in-loop (HIL) simulations for various operating points using the Navistar 7.6 liters six-cylinder engine. The effectiveness of the proposed architecture in handling injector nozzle clogging, intake manifold leaks, and pressure shift faults is illustrated. The results demonstrate that the proposed architecture can completely overcome faults and maintain the desired torque in a few seconds. Moreover, the average accuracy of 96% is observed for the engine model compared to experimental data. It is anticipated that the proposed end-to-end architecture will be easily deployable on unmanned marine vessels and can be extended to accommodate other component faults.

1. Introduction

The definition of complete autonomy in autonomous systems should also consider their ability to handle faults, thereby eliminating the need for rescue in case of mishaps. The inability to mitigate faults can cause severe damage to components and delays in accomplishing the mission. Particularly in the case of unmanned marine vessels, autonomous fault mitigation becomes essential, as faults can occur anywhere during travel and are difficult to rectify. Therefore, creating self-reliant systems capable of managing faults autonomously and continuing their mission holds significant potential. This paper specifically focuses on creating an online fault mitigation framework for marine diesel engines to achieve comprehensive autonomy in unmanned surface vessels.

Fault correction plays a critical role in enhancing the safety and reliability of the system. Rectification of faults in complex systems, such as engines, is almost impossible without pre-built fault management

systems. Researchers have tackled fault mitigation in engines primarily in three ways. First, diagnose the faults and then isolate or correct them manually. There are numerous works that have been conducted on diagnosing and isolating the faults in the engines, Goyal, Chowdhury, Hazarika, Khawale, Singh, Crawford, and Rai (2023), Song and He (2023) and Yuan, Liu, Ding, and Pan (2017). Second, for slight deviations from the nominal behavior, there are built-in capabilities in the recently developed engines where the Engine Control Unit (ECU) rectifies minor deviations by modifying the actuators/control parameters (Jeffrey, Cutajar, Prosser, Lickess, Richardson, & Riches, 2005). Lastly, reconfiguring the system on its own according to the faults information. This reconfiguration is generally carried out using the Fault Tolerant Control (FTC) systems. FTC is typically a design approach in control systems to ensure the continued operation of a system even in the presence of faults and failures (Amin & Hasan, 2019). The primary goal of FTC is to detect, isolate, and accommodate faults to

* Corresponding author at: Department of Automotive Engineering, Clemson University, Greenville, 29607, SC, USA.

E-mail address: rrai@clemson.edu (R. Rai).

<https://doi.org/10.1016/j.conengprac.2024.106045>

Received 31 January 2024; Received in revised form 6 August 2024; Accepted 8 August 2024

Available online 17 August 2024

0967-0661/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

maintain the desired system performance. Due to the advancements in computational capabilities, control theories, and artificial intelligence models, FTC is widely used in many applications (Li, Zhang, Luo, & Li, 2023; Saffar, Ghanbari, Ebrahimi, & Jannati, 2023).

There are two main approaches to FTC: Passive Fault Tolerant Control (PFTC) and Active Fault Tolerant Control (AFTC). The PFTC relies on the inherent design and characteristics of the system to tolerate faults without active intervention (Jiang & Yu, 2012). Amin and Mahmood-ul Hasan (2023) introduced the PFTC system for air-fuel ratio control of internal combustion engines. Additionally, Souami, Mechbal, and Ecoutin (2015) proposed an efficient PFTC system using Takagi–Sugeno formalism to control fuel flow rate while Wang and Wang (2012) utilized grouping theory to control actuator faults. Generally, PFTC systems are considered faster in fault correction, but they face significant issues in the case of adaptability, as they are designed to operate within certain predefined limits, and their response to faults is often predetermined. On the other hand, AFTC systems involve actively compensating for faults when they occur and stabilizing the operation according to the fault information (Abbaspour, Mokhtari, Sargolzaei, & Yen, 2020). Various AFTC systems for air-fuel ratio control are proposed by AA Amin and his group (Shahbaz & Amin, 2023) using Artificial Neural Networks (ANNs), fuzzy logic controllers, genetic algorithms, and regression-based observer models. Zhang, Amodio, Li, Aksun-Güvenç, and Rizzoni (2018) presented AFTC using a fault diagnosis and mitigation technique for maintaining the torque level, focusing on pedal mechanical stiction and pedal sensor faults. Gutiérrez León, García-Morales, Escobar-Jiménez, Gómez-Aguilar, López-López, and Torres (2018) introduce FTC for internal combustion engine's MAF sensor based on adaptive observers. Sakthivel, Joby, Wang, and Kaviarasan (2018) proposed a nonlinear fault-tolerant controller method that addresses time-varying delays, actuator saturation, and actuator faults, ensuring the finite-time stability of the closed-loop system. Kaviarasan, Sakthivel, and Kwon (2016) designed a fault-tolerant state feedback controller using a linear matrix inequality-based optimization algorithm. Their method ensures the robust stochastic stability of the entire system. While designing these controllers, accurate fault information is essential. However, many of the aforementioned FTCs operate under the assumption of a flawless fault diagnosis scheme. Without taking into account an accurate fault diagnosis system, the AFTC may not perform as anticipated.

Apart from the PFTC and AFTC, a Hybrid Fault Tolerant Controller (HFTC) is developed from the amalgamation of these two controllers (Abbaspour et al., 2020). Within the HFTC, the system swiftly responds to faults through the PFTC functionality in safety-critical scenarios and subsequently optimizes operations using the AFTC characteristic in post-fault conditions (Khatibi & Haeri, 2019; Wang, Yao, & Li, 2016). Alsuwian, Tayyeb, Amin, Qadir, Almasabi, and Jalalah (2022) designed an HFTC system for air-fuel ratio control of internal combustion engines using a genetic algorithm and higher-order sliding mode control. Shahbaz and Amin (2023) developed HFTC using ANNs and sliding mode control for sensor faults. These architectures are effective for non-linear systems, uncertainties and disturbances. However, their primary drawback lies in their intricate structure and slow response due to the excessive computational load. Furthermore, the knowledge-based reconfiguration system is another type of mitigation system that researchers have explored. These architectures do not rely on predefined system models. Instead, they need a large amount of historical data about the system (Ding, 2021). Generally, this strategy utilizes Principal component analysis (Wu, Zhou, & Chen, 2021), ANNs (Ebenezer, khurshid, Devi, Sandeep, Manipal, Vijay, & Dhanesh, 2021), support vector machine (Lv, Yang, Li, Liu, & Li, 2024), and fuzzy systems (Mu, Zhang, Xi, Wang, & Sun, 2021) to build a fault correction system. The effectiveness of these architectures relies on a wide range of faulty data, and generating precise data for faulty behavior and component failure is the biggest challenge.

To sum up, existing fault-tolerant architectures typically depend on predetermined scenarios and fault severity, an accurate fault diagnosis system, and a wide range of faulty data for data-driven models. Some controllers are unstable for non-linear systems, cannot handle uncertainties and disturbances due to sensor noise, and are computationally heavy, making them unsuitable for deployment in physical systems. Therefore, while these architectures may work for specific faults, conditions, or situations, robust and generalizable models for autonomous systems are needed. Specifically, strategies are needed that are not restricted to certain conditions and can handle highly non-linear systems, such as engines, while being computationally efficient and lightweight enough for deployment in autonomous vehicles. In this article, a digital twin-enabled fault mitigation system is proposed. The control calibration map, generated from a high-fidelity hybrid engine model, provides initial parameter values; later, feedback control compensates for minor deviations. This strategy enables us to handle non-linearities in the system and makes it computationally faster due to the use of an accurate control calibration map. Moreover, a digital twin-generated control calibration map allows us to cover a broader and more unusual range of conditions, making the proposed architecture robust enough to manage unprecedented faulty scenarios.

Multiple diesel engine models are available in the literature. For instance, 0D (Payri, Olmeda, Martín, & García, 2011), 1D (Cornolti, Onorati, Cerri, Montenegro, & Piscaglia, 2013), 2D (Lyubarsky & Bartel, 2016), and 3D Computational Fluid Dynamics (CFD) (Hong, Wooldridge, Im, Assanis, & Pitsch, 2005). 0D models are a function of only one variable, time, and it does not have any spatial dependence. These models consist of ordinary differential and algebraic equations. On the contrary, 1D, 2D, and 3D models have one, two, and three spatial dimensions of a considered coordinate system. A well-known 0D physics-based diesel engine model developed by Wahlström and Eriksen (2011) is widely used for the development of engine controllers and diagnosis systems. This is typically an air-path model and one of the computationally efficient models which can be used for multiple purposes. In recent years, Bondarenko and Fukuda (2020) have built a 0D mean value engine model that combines the time domain cycle with the crank-angle resolved combustion model. Furthermore, a high-fidelity 1D model was proposed by Bo, Wu, Che, Zhang, Li, and Myagkov (2024) for the purpose of fault diagnosis; this model demonstrates accurate results for fault detection. A computationally faster 1D engine model is designed by Hautala, Mikulski, Söderäng, Storm, and Niemi (2023) and Söderäng, Hautala, Mikulski, Storm, and Niemi (2022). It is a control-oriented model developed by incorporating model reduction strategies. Overall, 1D models serve well for precise fault detection and control but are generally considered computationally slower than 0D models. This is because they involve multiple differential equations, which increase the computational effort (Tang, Zhu, & Men, 2022). Additionally, 1D models are less flexible to integrate into other engines due to the numerous parameters required, and adding component faults to 1D models can be challenging. On the contrary, 0D models, while suffering from accuracy issues, require fewer computational resources and are more robust and computationally faster. Therefore, this paper introduces a novel 0D diesel engine model that substitutes the cylinder combustion process with an ANN and uses physics-based equations to describe the remaining subsystems. This approach helps us preserve accuracy while achieving good computational efficiency.

In this paper, a Bayesian Optimization (BO) (Frazier, 2018) based feedback control is proposed to minimize fuel consumption while maintaining the desired torque for both nominal and faulty scenarios. This feedback control system utilizes a Proportional–Integral (PI) controller (Chowdhury, Goyal, Khawale, Crawford, & Rai, 2023; Wang, Sun, Zhang, Huang, Yu, & Dong, 2023) to adjust parameters such as injection pressure, injection duration, and the Start of Injection (SOI). The PI controllers achieve this by using the difference between the setpoint torque and the current engine torque as the feedback signal. To address errors arising from non-calibrated faulty operating points, models, and

external disturbances, an additional feedback controller is proposed. The accuracy of the proposed hierarchical control architecture is highly dependent on the fidelity of the hybrid engine model. Consequently, a high-fidelity and computationally efficient hybrid engine model for the diesel engine is introduced. Finally, the procedure to extend this architecture for hardware implementation is demonstrated and tested in model-in-loop (MIL) and hardware-in-loop (HIL) environments for injector nozzle clogging, intake manifold leaks, and pressure shift faults.

The main contributions of this paper are as follows:

- New fuel injector and neural network-based multi-cylinder models are introduced. These models show suitability for controller modeling along with an error of less than 4% with experimental data.
- This paper provides new fault models for fuel injector nozzle erosion/clogging and manifold air leakages in diesel engines and demonstrates the fault mitigation strategies for those faults in the MIL environment.
- The complete engine model is calibrated and validated over the experimental data for various operating points.
- This paper discusses a novel fault mitigation strategy based on the hierarchical control architecture. The observation shows nearly a complete recovery of faults in less than a second, with minimal overshoot.
- A Python-based framework was developed for direct communication with the data acquisition system, enabling HIL functionality. The code is available on GitHub at <https://github.com/Rajk-hawale/HIL-FaultMitigation>, offering benefits to fellow researchers in establishing their HIL setup.

The remainder of the paper is organized as follows: Section 2 presents information on the experimental setup and measurements. Section 3 discusses the hybrid engine model and fault modeling, while Section 4 describes the controller architecture. Section 5 details the HIL setup. Results and discussion are provided in Section 6, including the validation of the hybrid engine model and the HIL and MIL simulations. Section 7 discusses the comparison with existing literature, as well as the assumptions and limitations of the proposed work. Finally, Section 8 concludes the article.

2. Experimental setup and measurements

2.1. Experimental setup description

Inline 6-cylinder Navistar production DT diesel engine is used for experimental validation, as shown in Fig. 1. The specifications of the engine are detailed in Table 1. It is equipped with a turbocharger, electro-hydraulic-based fuel injectors, and overhead valves. Sensors have been integrated into the engine to monitor various parameters, including air-path pressure, fuel rail pressure, fuel flow rate, airflow rate, as well as temperature for intake and exhaust. The air-path pressure and temperature are measured using low-speed pressure transducers and a K-type thermocouple. The airflow rate is measured using a laminar flow element, while the AVL fuel balance is employed for measuring the fuel flow rate. The active Dynamometer operated through the test bed automation system AVL PUMA, regulates the engine speed. Two data acquisition systems, INCA and PUMA, control and read the engine signals. Using INCA, a Controller Area Network (CAN) calibration protocol is established to connect the ECU to the user hardware. INCA software is used to control the injection parameters through an open ECU interface. Apart from the INCA CAN bus, a separate CAN bus (called PUMA CAN bus) communication was established using KVASER hardware to transmit torque, engine speed, and other engine air-path measurements like pressure, temperature, and flow rate. ECU data collection is synchronized once per engine cycle, while the air-path measurements and flow rates are sampled at 5 Hz. The details of the HIL system are discussed in the last section, along with HIL testing results.

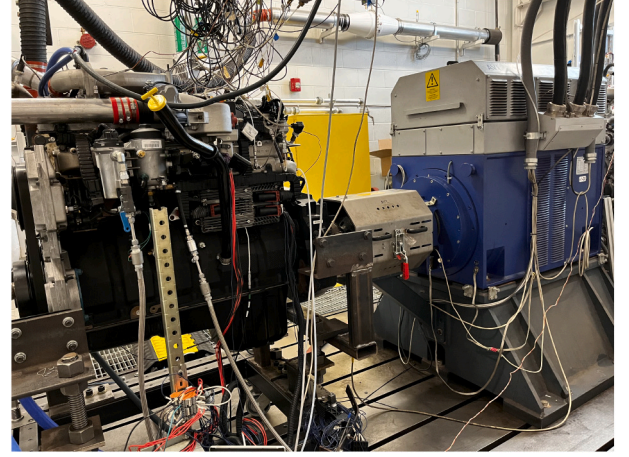


Fig. 1. Engine test cell setup.

Table 1
Engine specifications.

Parameters	Units	Values
Numbers of cylinders	[-]	6
Displacements	[L]	7.6
Bore	[mm]	116.6
Stroke	[mm]	118.9
Compression ratio	[-]	16.9
Fuel injector type	[-]	Hydraulically actuated electronic unit injection

Table 2
Design of experiments for experimental data.

Parameters	Operating points (baseline)			
Ne (RPM)	1000	1200	1400	1600
Dinj (ms)	1.2	1.2	1.2	1.2
SOI (CAD)	0.64	1.95	2.84	2.60
Pinj (MPa)	6	8	8	8
Control parameter change from baseline				
Dinj (ms)			-0.4, +0.4, +0.8	
SOI (CAD)	-5, +5, +8	-7, -12, +9	-7, -11, +5	-5, +5, +8
Pinj (MPa)			-2, +2, +4	

2.2. Experimental dataset

The experimental data is collected for the engine model calibration and validation. The Design of Experiments is truncated between 1000–1600 Revolutions per Minute (RPM) at four different engine speeds and restricts the brake torque between 100–500 N m. Table 2 mentions injection parameter values at different speeds. For more operating points and to cover the entire brake torque region, the control parameters are changed one by one from the baseline data.

2.3. Data preprocessing

The experimental data is preprocessed in two steps to address issues such as arbitrary missing data and mismatched sampling frequency rates between INCA and PUMA. First, to deal with missing data entries in the data files, the neural network-based regression model is utilized to fill in the missing values. This issue is primarily observed in fuel flow rate sensor readings. Separate neural network models are trained for each engine speed using data points where sensor readings are available. Second, to have a consistent sampling frequency for INCA and PUMA data, a simple bilinear interpolation scheme is applied to match the data timestamp for all data points.

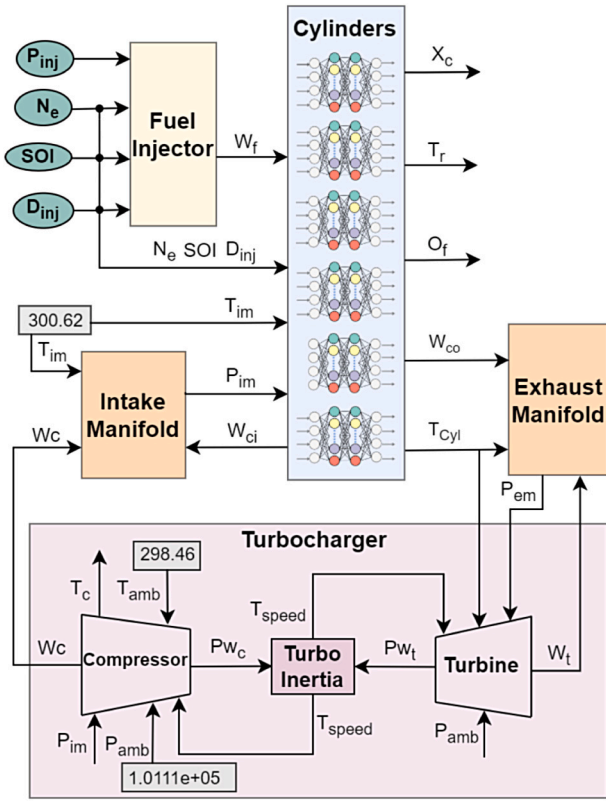


Fig. 2. Structural diagram for the hybrid engine model.

3. Open-loop diesel engine modeling

3.1. Model introduction and layout

The proposed open-loop model is a mean-value hybrid diesel engine model that uses filling and emptying approaches for the manifolds (Wahlström & Eriksson, 2011). Typically, Mean Value Engine Models (MVEM) disregard crank angle-based information and instead employ an effective value obtained from the combustion cycle. MVEM models are primarily utilized for faster operations (Singh, Khawale, Hazarika, Bhatt, Gainey, Lawler, & Rai, 2024). The proposed model is characterized by its subsystem-based structure, which utilizes parameterized functions to describe the performance of each subsystem. This approach reduces the reliance on maps and their associated performance, making the model more versatile and adaptable. Fig. 2 presents an outline of the model, which obtains the net torque, oxygen-to-fuel ratio, and oxygen concentration from the cylinder subsystem, as well as mass flow, temperature, and pressure from most of the subsystems, utilizing sixteen parameters. The notations for the parameters in Fig. 2 are provided in the nomenclature. The control input parameters for the engine model are as follows:

1. N_e - Rotational engine speed (r/min)
2. SOI - Start of injection (Crank angle degree - CAD)
3. D_{inj} - Injection duration (ms)
4. P_{inj} - Injection pressure (MPa)

The engine model comprises the fuel injector, cylinder, turbocharger, and intake and exhaust manifold subsystems. Individual subsystems are detailed in the following subsections.

3.2. Fuel injection subsystem

Effective fuel injection timing, precise delivery, and controlled injection rate are critical factors that significantly impact combustion and emission control (Lino, Maione, & Rizzo, 2005). Considering those factors, the high-pressure common-rail solenoid-operated fuel injection system is modeled. Generally, modern fuel injection systems have four primary components: fuel tank, high-pressure pump, common rail, and fuel injector. The fuel injector model is fundamentally based on the principle of conservation of mass (Vortmann, Schnerr, & Seelecke, 2003). The injector flow rate is associated with the pressure difference, density of fuel, area of the nozzle, and discharge coefficient, as illustrated in the following equation:

$$W_f = \sum_{\zeta=0}^{720} \alpha(\zeta) \cdot \text{sign}(\Delta P) \cdot c_d \cdot A_{flow} \sqrt{\frac{2}{\rho} |P_{inj}(\zeta) - P_{cyl}(\zeta)|} \quad (1)$$

$$\alpha(\zeta) = \begin{cases} 1, & \text{if } SOI \leq \zeta \leq EOI \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Here, ζ defines the crank angle in each combustion cycle, which ranges from 0 – 720 CAD. ΔP is the pressure difference between the injector and cylinder pressure, and all other notations from Eqs. (1) and (2) are mentioned in the nomenclature. The injection nozzle's flow area and discharge coefficient are tuned using experimental data. α from Eq. (1) controls the opening and closing of valve movement.

Based on the system characteristics, certain assumptions have been made. Firstly, the injector valve is assumed to be either completely open or completely closed. This results in a sudden movement of fuel flow, as opposed to a gradual increase or decrease. Secondly, the pressure is used to represent the system state during operations without considering changes in fuel temperature. Thirdly, The fluid dynamics associated with the flow through pipes is being disregarded. Discussing the accuracy of the fuel injection model, an error of less than 3% is observed with respect to experimental data. A comparison between experimental and model data for 1600 RPM is shown in the first plot (FI mass flow) of Fig. 10.

3.3. Cylinder subsystem

The proposed approach employs a multi-cylinder model for the cylinder subsystem, which consists of six separate cylinders and fuel injectors (Garcia, Monsalve-Serrano, Villalta, & Fogué-Robles, 2022). This enables us to control the input parameters and induce faults for individual cylinders. Although most MVEM uses a combined cylinder approach to facilitate faster computation for control purposes (Wahlström, 2006), there exist a few multi-cylinder models (Souder, Mehresh, Hedrick, & Dibble, 2004), Yin, Turesson, Tunestål, and Johansson (2020) that prioritize computational speed over accuracy. In contrast, CFD-based multi-cylinder engine models (Adhikary, Reitz, & Ciatti, 2013), Bhawe, Kraft, Montorsi, and Mauss (2004) offer higher accuracy but are slower than real engine cycle time. Therefore, the proposed model offers a balance between speed and precision by leveraging a detailed thermodynamic cylinder model for accuracy and then replacing it with a neural network model for faster computation.

3.3.1. Thermodynamic cylinder model

A high fidelity 0D thermodynamic cylinder model, described in Gainey and Lawler (2021), was used to determine work output from each cylinder and exhaust temperature, pressure, mass flow rate, and composition based on intake and fueling conditions. This model consists of a single zone initialized at the top dead center (TDC) gas exchange (Tang, Zhang, Gan, Jia, & Xia, 2017). At each time step (0.1 crank angle degrees), the specific volume and specific energy are computed using an Euler forward method, defining the thermodynamic state. The specific volume is calculated by dividing the volume, determined by the equations of crank slider motion, by the mass in the

cylinder, determined by the conservation of mass. Mass flows through the intake and exhaust valves are calculated using the equations of compressible flow through an orifice:

$$\frac{dm}{dt} = \frac{C_d A_r P_0}{\sqrt{RT_0}} \left(\frac{P_r}{P_0} \right)^{\frac{1}{\gamma}} \sqrt{\frac{2\gamma}{\gamma-1} \left[1 - \left(\frac{P_r}{P_0} \right)^{\frac{\gamma-1}{\gamma}} \right]} \quad (3)$$

where $\frac{dm}{dt}$ is the mass flow, C_d is the discharge coefficient, A_r is the orifice area, P_0 is the pressure for upstream stagnation, R is the universal gas constant divided by the molecular weight of the gas, T_0 is the temperature for upstream stagnation, P_r is the throat pressure (considered to be equivalent to the downstream pressure), and γ is the ratio of the specific heat.

The specific energy is computed using the first law of thermodynamics for an open system including expansion and compression work, injection of fuel, exhaust and intake flow, convective heat transfer, and crevice flow. The Woschni heat transfer correlation is used to calculate the convective heat transfer coefficient. Once the thermodynamic state is defined, other mixture properties are determined using the NASA polynomials (McBride, 2002), considering the fluid as a six-species mixture of dodecane (fuel), O_2 , N_2 , CO_2 , H_2O , and Ar. The mixing-controlled combustion process is modeled using a Wiebe function calculated using the methodology described by Miyamoto, Chikahisa, Murayama, and Sawyer (1985) based on injection timing, injection duration, and ignition delay. In this thermodynamic model, the combustion of dodecane immediately results in the formation of CO_2 and H_2O .

3.3.2. Neural network-based cylinder model

Using a thermodynamic cylinder model for multi-cylinder simulation in every engine cycle makes the entire model computationally inefficient. However, replacing the thermodynamic cylinder model with a shallow neural network (Singh, Rai, Khawale, Patel, Bielecki, Nguyen, Wang, & Zhang, 2023) helps in improving the speed significantly without compromising accuracy. Thus, this hybrid engine model, which comprises six neural networks for six different cylinders, becomes faster than a wall clock time with better fidelity. To train this neural network, a dataset was created from the thermodynamic cylinder model. Considering the maximum possible range for all the cylinder inputs, around 100,000 input data points are generated using the Latin hypercube sampling (LHS) based design of the experimental technique (Shields & Zhang, 2016). LHS is a statistical method to generate near-random samples that can cover the entire design space of cylinder inputs. Training and testing data are randomly segregated with 90% and 10% of data points.

With respect to the network architecture, this regression neural network is relatively shallow, comprising three layers. Each layer consists of neurons ranging from 6 to 12. The network has six inputs and six outputs, as illustrated in the cylinder subsystem of Fig. 2. Both of these inputs and outputs are normalized using the z-score normalization strategy (Singh & Singh, 2020).

3.4. Manifolds and turbocharger subsystems

The intake and exhaust manifold subsystems are modeled with two states: intake manifold pressure and exhaust manifold pressure. The ideal gas law and the principles of mass conservation are utilized as the fundamental basis for modeling the pressure of manifolds. Equations to compute the pressure in both the intake and exhaust manifolds are as follows:

$$\frac{d}{dt} P_{im} = \frac{R_a T_{im}}{V_{im}} (W_c - W_{ei}) \quad (4)$$

$$\frac{d}{dt} P_{em} = \frac{R_e T_{em}}{V_{em}} (W_{eo} - W_i) \quad (5)$$

All the parameters from the above equations are described in the nomenclature. The turbocharger consists of a compressor, turbine, and

turbo inertia models, as depicted in the turbocharger subsystem of Fig. 2. The work of Wahlström and Eriksson (2011) is used for the turbocharger subsystem.

3.5. Fault modeling

The capability of the proposed hybrid engine model is showcased by extending it to model a couple of important failure modes. The modeling of fuel injector nozzle erosion/clogging and intake manifold leakage is briefly described in this subsection. These failure modes are difficult to incorporate experimentally since they are non-recoverable, and it is challenging to control or change the magnitude of these faults. Therefore, fault mitigation strategies for these failure modes are demonstrated in the MIL environment.

3.5.1. Fuel injector area erosion/clogging model

Fuel injector nozzle erosion and clogging can occur due to cavitation and poor fuel quality. These failure modes are simulated by increasing and decreasing the area of the injector nozzle in the following manner.

$$Q_{inj} = \text{sign}(P_{inj} - P_{cyl}) \alpha c_d A_{ff} A_{flow} \sqrt{\frac{2}{\rho} |P_{inj} - P_{cyl}|} \quad (6)$$

The fuel injector erosion and clogging are modeled using the area fault factor parameter A_{ff} , where $A_{ff} > 1$ corresponds to erosion and $A_{ff} < 1$ corresponds to clogging. All other notations from the above equations are mentioned in the nomenclature.

3.5.2. Intake manifold leak model

The connection between the cylinder head and the intake manifold is established by a seal that ensures an airtight arrangement. Nevertheless, with the passage of time, these seals may deteriorate, leading to a leakage in the intake manifold, which adversely affects the efficiency of the engine. Gustafsson, Nyberg, and Omstedt (1998) modeled the leak as a flow through a restriction and validated it with good accuracy. The leakage mass flow rate from the intake manifold is:

$$W_{leak} = \frac{A_{leak} P_{im}}{\sqrt{R_a T_{im}}} \psi_{\kappa_s} \left(\frac{P_{atm}}{P_{im}} \right) \quad (7)$$

A_{leak} is the area of the hole given by $\pi(d/2)^2$ with d as leak diameter, and the function ψ_{κ_s} is a complex function which is defined by

$$\psi_{\kappa_s} \left(\frac{P_{atm}}{P_{im}} \right) = \begin{cases} \sqrt{\frac{2\kappa_s}{\kappa_s - 1} \left\{ \left(\frac{P_{atm}}{P_{im}} \right)^{\frac{2}{\kappa_s}} - \left(\frac{P_{atm}}{P_{im}} \right)^{\frac{\kappa_s + 1}{\kappa_s}} \right\}} & \text{if } \left(\frac{P_{atm}}{P_{im}} \right) \geq \left(\frac{2}{\kappa_s + 1} \right)^{\frac{\kappa_s}{\kappa_s - 1}} \\ \sqrt{\kappa_s \left(\frac{2}{\kappa_s + 1} \right)^{\frac{\kappa_s + 1}{\kappa_s - 1}}} & \text{otherwise} \end{cases} \quad (8)$$

where κ_s is the ratio of specific heat capacity (C_p/C_v) and all remaining notations from Eqs. (7) and (8) are mentioned in the nomenclature. In the presence of a leak, W_{leak} is subtracted from W_c in Eq. (4).

3.6. Model calibration

Calibrating the model as closely as possible to the experimental setup is crucial in its development. To achieve this, the engine model is calibrated in two stages. Firstly, the tuning parameters are initialized by calibrating the fuel injector, cylinder, and turbocharger subsystems individually using the non-linear least square optimization method (Markovsky & Van Huffel, 2007). In this stage, the mean value from each operating condition is used instead of the entire dataset. This approach, which considers mean values and individually calibrates

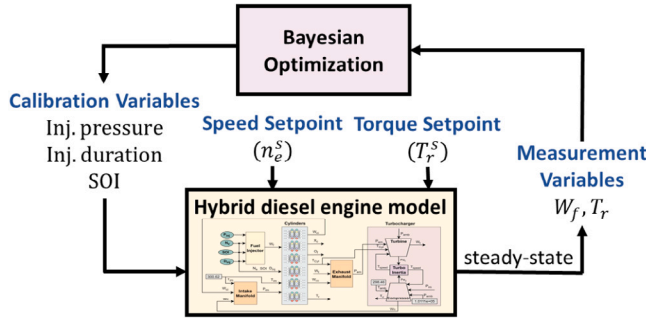


Fig. 3. Architecture for engine calibration.

subsystems, helps to speed up the calibration process, avoid measurement/sensor noise, and improve validation results. The parameters considered for tuning in this stage are C_d , A_{flow} , V_{im} , V_{em} from Eqs. (1), (4), and (5). Additionally, since the turbocharger model developed by Wahlström and Eriksson (2011) is utilized, the tuning parameters for the turbocharger subsystem can be found in their description of the turbine and compressor model.

In the second stage, all subsystems are combined, and the model is recalibrated using the complete dataset. By doing so, the effect of one system over another is addressed, showing a significant improvement in transient behavior. One or two important state variables from each subsystem are used, and the parameters for those state variables are re-tuned exclusively. The problem formulation for parameter tuning is as follows:

$$\begin{aligned} \text{minimize}_{\theta} \quad & F(\theta) = \sum_{v=1}^V \sum_{d=1}^D \frac{(\beta_v^{exp}[d] - \beta_v^{mod}[d])^2}{N_d} \\ \text{subject to} \quad & \theta_{min} \leq \theta \leq \theta_{max} \end{aligned} \quad (9)$$

The objective function minimizes the error between experimental data (β_v^{exp}) and model data (β_v^{mod}), both of which are normalized. The tuning parameters are represented by θ , with lower and upper bounds of θ_{min} and θ_{max} , respectively. v is the number of state variables, with ($V = 6$), one from each subsystem. d represents the number of data points used for calibration, and N_d is the total number of data points. This two-stage calibration process has proven effective in enhancing model accuracy and capturing transient behavior more comprehensively.

4. Controller architecture

4.1. Control calibration map

The hybrid diesel engine model, calibrated with experimental data, is utilized to employ the offline Bayesian optimization (BO) algorithm. The BO algorithm creates the control calibration map. Typically, BO is employed to identify the global maximum or minimum of an unknown objective function (Shahriari, Swersky, Wang, Adams, & De Freitas, 2015). The function might not have a closed form but can be evaluated at a query point.

The map produced by the BO algorithm optimizes the fuel flow rate efficiency W_f for different engine operating points. The map contains the mapping from the injection parameters to the engine speed and torque. The engine speeds ranged from 1000 to 1600 RPM with a step of 100 RPM, and torque values ranged from 100 to 500 N m with a step of 50 N m in the map. The calibration map is generated at a steady-state operating condition with fixed speed (N_e) and torque (T_r). The engine parameters which can be controlled are denoted as:

$$u_1 = P_{inj}, \quad u_2 = D_{inj}, \quad u_3 = SOI.$$

Table 3

Simulation setup for engine calibration.

	Parameters	Range
Calibration variables	P_{inj}	[4–25] MPa
	D_{inj}	[0.3–3] ms
	SOI	[345–380] CAD
Constraint	T_r	$[T_r^s - \delta, T_r^s + \delta]$
Objective	W_f	–

The fuel flow rate is a function of the injection parameters and operating conditions and is defined as $W_f(\mathbf{u}, N_e, T_r)$, where $\mathbf{u} = [u_1, u_2, u_3]$. In addition to the constraints on the inputs \mathbf{u} , the steady-state torque is set as an additional constraint. The engine calibration optimization problem can be outlined as follows:

$$\begin{aligned} & \text{Minimize } W_f(\mathbf{u}, N_e, T_r) \\ & \text{subject to: } T_r^s - \delta \leq T_r(\mathbf{u}, N_e, T_r) \leq T_r^s + \delta, \\ & \quad N_e = N_e^s, \\ & \quad P_{inj_{min}} \leq u_1 \leq P_{inj_{max}}, \\ & \quad D_{inj_{min}} \leq u_2 \leq D_{inj_{max}}, \\ & \quad SOI_{min} \leq u_3 \leq SOI_{max}. \end{aligned} \quad (10)$$

The variables N_e^s and T_r^s represent the setpoint speed and torque. In the problem formulation described by Eq. (10), the torque constraint is relaxed into inequality constraints, with a small parameter denoted as δ . For this particular problem, the chosen value of δ is 0.3 N m. The primary objective of the optimization Eq. (10) is to minimize the fuel consumption rate W_f for a specific engine torque T_r^s and engine speed N_e^s . Table 3 provides a comprehensive overview of the simulation setup, including all engine variables and their associated constraints used in the optimization problem outlined in Eq. (10). Fig. 3 illustrates the architecture implemented for the engine calibration process.

4.2. System model

The engine torque is directly dependent on the injection parameters and can be described by the relation (Heywood, 1988).

$$\dot{T}_r = f(T_r, \mathbf{u}) \quad (11)$$

where f is a global Lipschitz function. By using the calibration map, feedforward control inputs are generated $\mathbf{u}^* = [u_1^*, u_2^*, u_3^*]$ which makes the system operate around the setpoints (N_e^s, T_r^s). Therefore, the torque at steady-state using the feedforward control inputs can be defined by

$$0 = f(T_r^s, \mathbf{u}^*)$$

The change of variables $\tilde{T}_r = T_r^s - T_r$, $\mathbf{u}^f = \mathbf{u} - \mathbf{u}^*$ results in

$$\dot{\tilde{T}}_r = -f(T_r^s - \tilde{T}_r, \mathbf{u}^* + \mathbf{u}^f) = \tilde{f}(\tilde{T}_r, \mathbf{u}^f) \quad (12)$$

where \mathbf{u}^f is designed as the feedback control of \tilde{T}_r which is explained in the next section. Since the steady-state system operates in a neighborhood of the setpoint (N_e^s, T_r^s), the perturbed system (12) can be defined by a linear time-invariant system

$$\dot{\tilde{T}}_r = a\tilde{T}_r + B\mathbf{u}^f \quad (13)$$

where

$$a = \left. \frac{\partial \tilde{f}}{\partial \tilde{T}_r}(\tilde{T}_r, \mathbf{u}^f) \right|_{\tilde{T}_r=0, \mathbf{u}^f=0}, \quad (14)$$

$$B = \left. \frac{\partial \tilde{f}}{\partial \mathbf{u}^f}(\tilde{T}_r, \mathbf{u}^f) \right|_{\tilde{T}_r=0, \mathbf{u}^f=0} \quad (15)$$

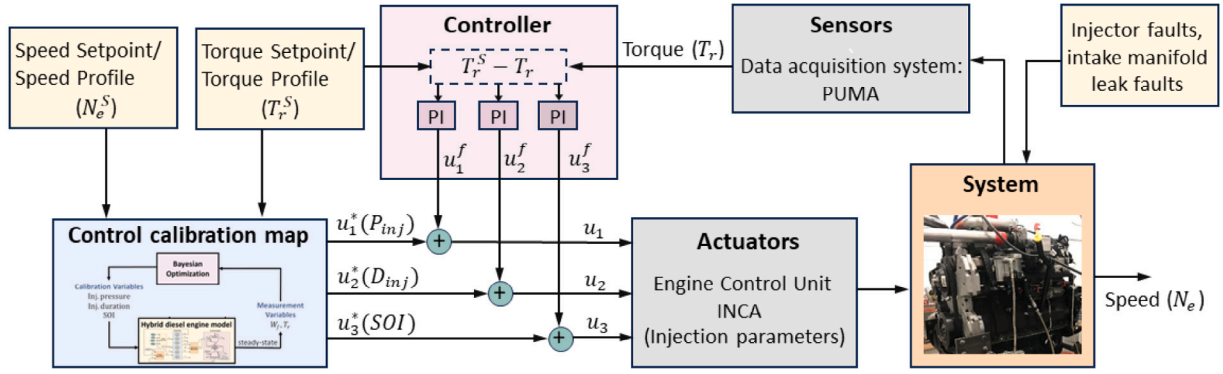


Fig. 4. System architecture consisting of sensors, actuators, and controllers components.

4.3. Feedback control using PI controller

Proportional-integral (PI) feedback controllers are used to sustain the torque at the specified setpoint (Schiavo, Padula, Latronico, Pal-tenghi, & Visioli, 2023). These PI controllers utilize the tracking error $\hat{T}_r = T_r^s - T_r$ to adjust parameters such as injection duration, injection pressure, and SOI. The PI controllers are outlined as follows:

$$u_1^f = k_p^1(T_r^s - T_r) + k_i^1 \int (T_r^s - T_r) dt, \quad (16a)$$

$$u_2^f = k_p^2(T_r^s - T_r) + k_i^2 \int (T_r^s - T_r) dt. \quad (16b)$$

$$u_3^f = k_p^3(T_r^s - T_r) + k_i^3 \int (T_r^s - T_r) dt. \quad (16c)$$

The controller parameters k_p^1 , k_p^2 , k_p^3 denote the proportional gains and k_i^1 , k_i^2 , k_i^3 denote the integral gains.

4.4. Hierarchical engine control

The overall control architecture is depicted in Fig. 4. The control calibration map generates the optimal injection parameters tailored to a specific setpoint of speed (N_e^s) and torque (T_r^s). Nevertheless, owing to modeling inaccuracies and potential engine faults, the generated optimal injection parameters may not attain the intended torque setpoint T_r^s . The PI controllers compensate for these faults and modeling discrepancies, ensuring that the torque remains at the desired setpoint. The applied injection parameters to the engine are outlined below:

$$u_1 = u_1^f + u_1^*, \quad u_2 = u_2^f + u_2^*, \quad u_3 = u_3^f + u_3^*,$$

Where u_1^* , u_2^* , and u_3^* represent the optimal injection parameters generated from the control calibration map.

5. Hardware-in-loop setup details

HIL system enables the data measurement and changing of the control signals in real time, autonomously, from the Python environment. A schematic of the HIL is detailed in Fig. 5. The Python API block in Fig. 5 contains the master Python script. This script synchronously reads data from both data acquisition systems, utilizes the acquired data for filter and controller operations, and transmits the control signals obtained from the controller to the ECU. The Python API code can be accessed using the following link (<https://github.com/Rajkhawale/HIL-FaultMitigation>). Complete software packages and hardware connections must be installed to run this code. The main components of the Python API are discussed below:

- **Torque Profile:** This is a Comma-Separated Value (CSV) containing the torque profile the engine torque should track. To demonstrate fault mitigation in the HIL environment, the profile illustrated in Fig. 6 is considered, containing torque setpoints for a total experiment duration of thirteen minutes, including the time when the pressure shift faults are introduced.

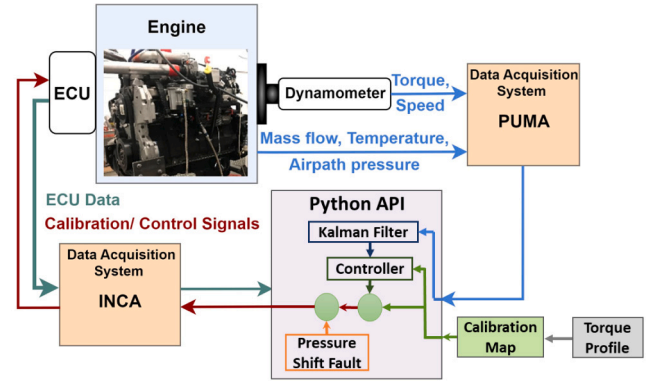


Fig. 5. Different components and their connections in the HIL system.

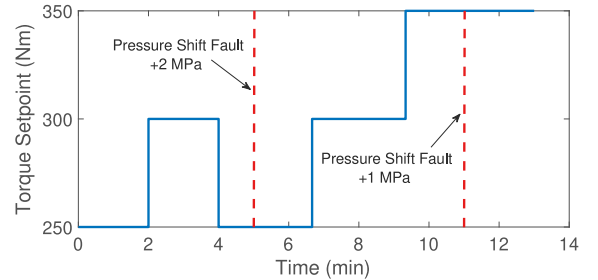


Fig. 6. Torque profile for the HIL simulation.

- **Calibration Map:** This CSV file contains a mapping from the injection parameters to the desired torque for a desired engine speed. This map has been offline utilizing a high-fidelity hybrid diesel engine model, optimizing the fuel efficiency across various engine operating points.
- **Kalman Filter:** The measured engine torque is very noisy and, when given to the PI controller without filtering, causes frequent changes in the injection parameters. A batch-wise Kalman filter is used (Khodarahmi & Maihami, 2023; Ou, Wang, Tang, Huang, & Liu, 2023) with a window size of 10 samples to smooth the measured torque. The filtered torque is then sent to the PI controller.
- **Controller:** The PI controller is used to change the injection pressure, injection duration, and start of injection (CAD) so that the engine torque is able to track the given torque profile. The PI controller uses the tracking error as the difference between the setpoint torque and the engine torque to change the injection parameters.

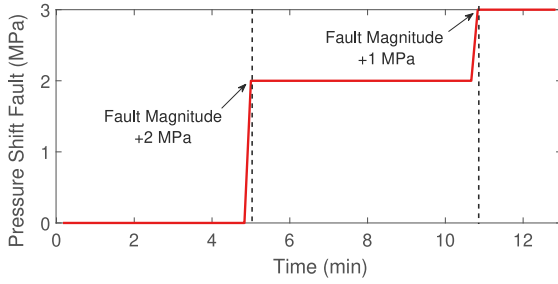




Fig. 7. Pressure shift fault profile for the HIL simulation.

Table 4

Hardware and API details for INCA and PUMA.

	INCA	PUMA
Function	Read ECU data and write the control signals	Read time-based sensor data (such as mass flow, temperature, pressure, torque, and speed)
GUI name	INCA V7.2.17	Kvaser CANKing
Adapter name	ES581.3	Kvaser Leaf Light V2
Adapter image		
Python library	INCA-python	CANlib and cantools
Sampling rate	Synchronous with engine speed	$\frac{1}{2}$, 1, 2, 4, 5, 10 Hertz

- **Pressure Shift Fault:** A step function is used to introduce the pressure shift faults. The faults are added to the injection pressure control input and then sent to the engine. Fig. 7 illustrates the pressure shift fault profile.

Appropriate hardware, drivers, and Python libraries have been installed to establish communication with both data acquisition systems. The INCA-Python and Kvaser Application Programming Interface (API) are used to interface with INCA and PUMA. Detailed information about the hardware and API for both data acquisition systems is provided in Table 4. The INCA-Python API has built-in Python dynamic libraries that can load and visualize the workspace environment from the INCA GUI, change booleans for control inputs, and write new calibration values for actuators in real time. Apart from the INCA CAN bus, a separate CAN bus (referred to as PUMA CAN bus) communication was established using Kvaser hardware. A custom dbc format file is created for the PUMA CAN bus to categorize signals into CAN frames identified by different arbitration IDs. These arbitration IDs determined the hierarchy of messages, and the frame with higher importance was imparted with the priority to transmit the message first in case of CAN bus congestion. CAN frames are transmitted to the Python API at fixed time intervals based on the transmission rate of the bus. The run-time for the Python loop consisting of read/write commands was iteratively tweaked and synchronized with the CAN bus transmission rate to avoid any data loss/congestion in the CAN bus.

6. Results and discussion

This section discusses the validation of the hybrid engine model using experimental data. Furthermore, the responses of the proposed fault mitigation architecture to multiple faults are demonstrated in MIL and HIL simulations.

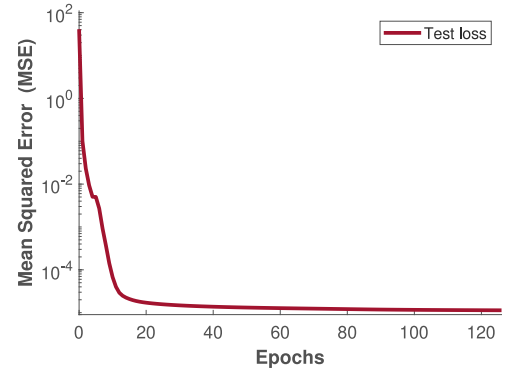


Fig. 8. Neural network training result for the cylinder model.

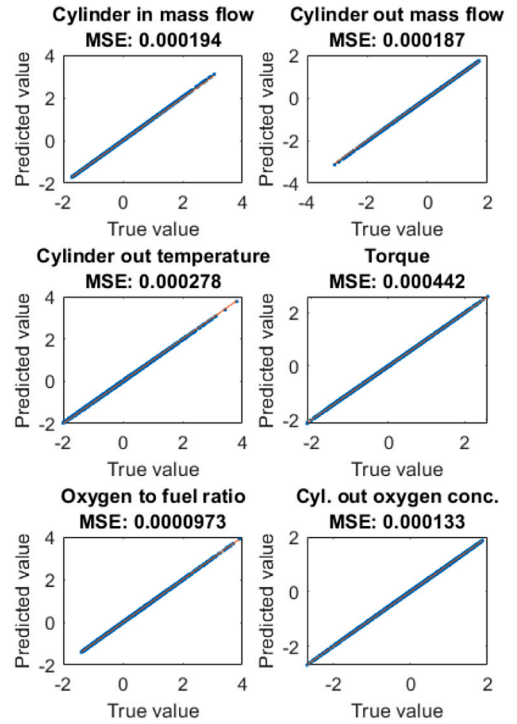


Fig. 9. True value vs. predicted value plot for neural network-based cylinder model.

6.1. Engine model validation

Here, the validation of the engine model is presented in two parts. First, the accuracy of the neural network-based cylinder model is described in terms of Mean Squared Error (MSE). Later, the overall accuracy of the engine model is detailed in Mean Percent Error (MPE) (Khair, Fahmi, Al Hakim, & Rahim, 2017).

As described in the previous section, the cylinder model is represented by a shallow neural network. The training result for the test loss is shown in Fig. 8. The performance of the trained network is evaluated using the predicted vs. true value plot, as presented in Fig. 9. The performance is evaluated on the test dataset, with true values from the thermodynamic cylinder model and predicted values from the trained network. Regarding the accuracy of the trained network, the overlapping behavior observed between the true and predicted values in the plots shows good prediction capability. Furthermore, as indicated in the title of Fig. 9, the MSE values are also in the acceptable range. The average MSE for the trained cylinder model is 2.21×10^{-4} .

The hybrid engine model is validated using the experimentally collected dataset described in Table 2. The model is validated in two

Table 5

The validation results of the hybrid engine model with experimental data are shown in the form of MPE over the mean values and the entire dataset for each output of the model.

System outputs	Over mean value data	Over entire data
	Mean percent error (%)	Mean percent error (%)
Cylinder exhaust temperature	2.14	2.07
Exhaust manifold pressure	3.57	3.78
Cylinder mass flow in	3.24	3.75
Cylinder mass flow out	4.09	4.78
Intake manifold pressure	3.88	4.34
Torque	4.6	6.2
Compressor mass flow	3.25	3.65
Turbine mass flow	4.1	4.9
Fuel injector mass flow	1.9	2.8

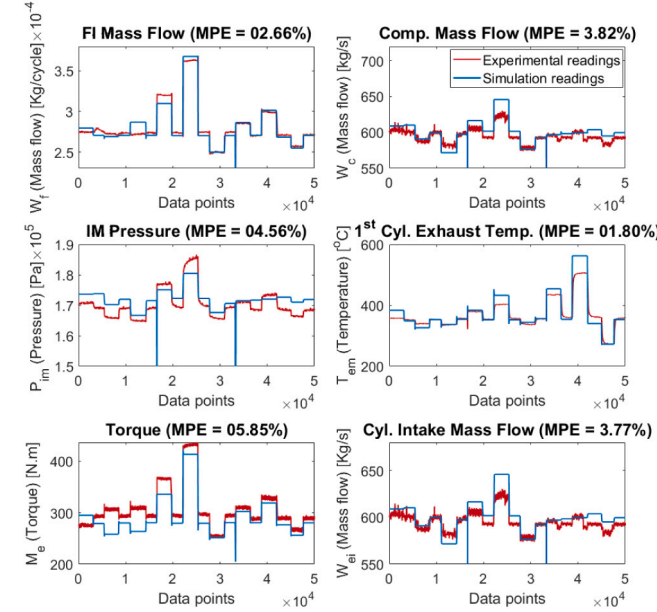


Fig. 10. Hybrid engine model validation plots for the engine speed of 1600 RPM with different operating points. The simulation and experimental readings are represented by the blue and red lines, respectively. The title of each plot includes the MPE value. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

different ways. Firstly, the mean values for each operating condition are identified. Instead of comparing all data points from a particular operating condition, the mean value is used and compared with the mean value obtained from the model's data. Validating over mean values helps in understanding how accurately the model reproduces the experimental behavior, particularly in the steady state. Secondly, validation is performed over the entire dataset. Validating the model over the entire (continuous) data provides insights into the model's response to transient behavior and the effect of sensor noise on the accuracy of the model.

Table 5 displays the results of the model validation, revealing that the MPE is approximately below 4% for mean value data and 6% for the entire data. The comparative plot between the experimental and hybrid engine model data for the engine speed of 1600 RPM is presented in Fig. 10. The MPE values mentioned above each plot in Fig. 10 are calculated for the dataset considered in the plots. The plots from Fig. 10 and MPE values obtained over entire data from Table 5 show that the proposed hybrid engine model is capable of giving decent accuracy in the presence of transient behavior and sensor noise. Furthermore, the controller requires faster computational models for real-time scenarios. Specifically, for the proposed fault mitigation framework, high-fidelity and resource-efficient models are required for effective fault correction.

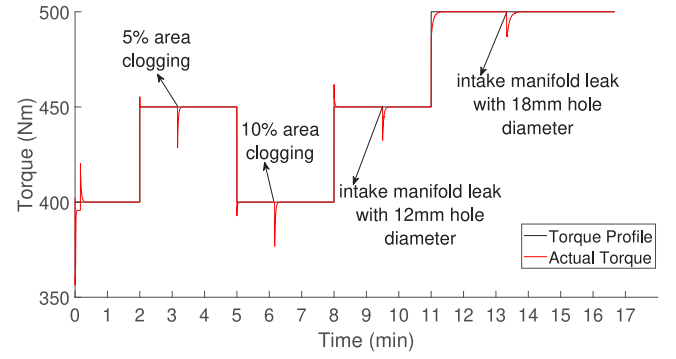


Fig. 11. Actual torque and torque profile in the presence of faults for MIL simulation.

Thus, the computationally faster hybrid engine model, with an accuracy of more than 95%, plays a critical role in the proposed fault mitigation strategy.

6.2. Fault mitigation results

This subsection presents the results of MIL and HIL simulations for real-time fault mitigation scenarios. Fault mitigation strategies are demonstrated in the MIL and HIL environments for a specific torque profile at 1400 RPM. Injector nozzle clogging and intake manifold leak faults are induced in the MIL environment, and a pressure shift fault is induced a couple of times in the HIL environment during the profile for mitigation. The effectiveness of the proposed method is showcased on injector nozzle clogging, intake manifold leak, and pressure shift faults, with the technique readily applicable to address malfunctions or failures in any other component.

6.2.1. Model-in-loop simulation

MIL simulations demonstrate the effectiveness of fault mitigation architecture in addressing various component faults and maintaining the torque profile in the modeling environment. A 5% and 10% injector nozzle area clogging fault is introduced at 180 and 370 s, respectively, and an intake manifold leak fault with a 12 mm and 18 mm hole diameter at 570 and 810 s, respectively. The chosen clogging and intake manifold leak scenarios illustrate progressively increasing fault severity.

At every torque setpoint change, a new set of injection parameters is loaded from the calibration map. The speed setpoint is 1400 RPM, and the PI controller parameters are given by $k_p^1 = 10^3$, $k_i^1 = 10^5$, $k_p^2 = 1e-3$, $k_i^2 = 1e-3$, $k_p^3 = 1e-4$, $k_i^3 = 1e-4$. At the beginning of the simulation, the system is given time to reach a steady state, typically during the first two seconds. Following this initial time period, the PI controllers are activated. Fig. 11 shows that the proposed architecture can completely mitigate faults and track the torque profile. The fault is also recovered in a couple of seconds with a minimal overshoot. Fig. 12 shows that whenever a fault is introduced or changes in the torque setpoint, the PI controller changes the injection pressure and duration to either compensate for the faults or track the new setpoint. Moreover, Fig. 12 shows that injection pressure and injection duration have the greatest impact on torque, while SOI has the least impact.

6.2.2. Hardware-in-loop simulation

The HIL test is demonstrated for the planner shown in Fig. 6. Engine torque tracking performance in the presence of faults is presented in Fig. 14, and the injection parameters that the controllers change to track the torque profiles are shown in Fig. 13. The HIL simulation results show that the proposed fault mitigation strategy can completely recover faults in a few seconds. When the torque profile from the planner changes, a new set of optimized injection parameters is loaded from

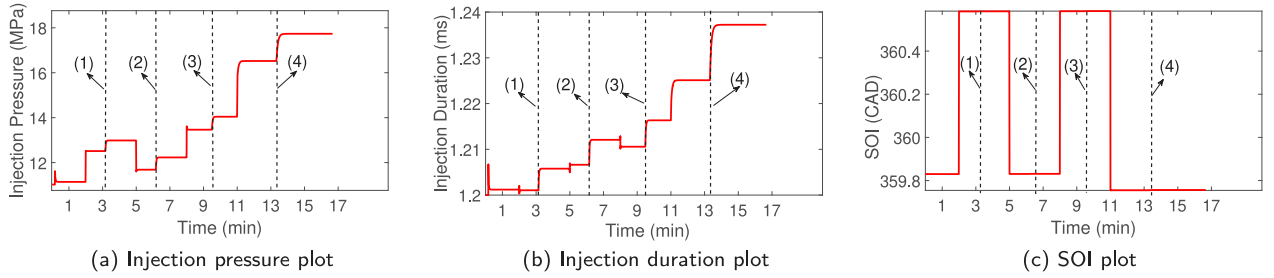


Fig. 12. Control/input parameters for MIL simulation. (1), (2), (3), and (4) from the above three plots define 5% area clogging, 10% area clogging, intake manifold leak with 12 mm hole diameter, and intake manifold leak with 18 mm hole diameter, respectively.

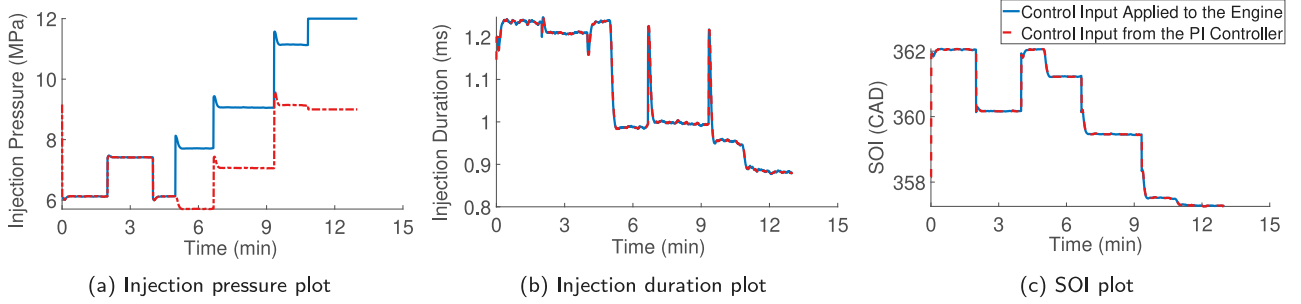


Fig. 13. Control/input parameters for the HIL simulation.

the calibration map corresponding to the setpoint torque. However, due to modeling errors, the engine torque does not reach the setpoint when the optimized injection parameters are applied to the engine. It can be seen from Fig. 14 that the engine torque does not reach the setpoint immediately; the PI controller compensates for the modeling error and tracks the torque profile with an overshoot of less than 15%. Once the fault is introduced, the mitigation architecture compensates for it, but the overshoot reaches up to 31%. Furthermore, the overshoot is observed to be higher when the torque setpoint changes after the fault occurs. The obtained engine torque is very noisy, as seen in the measured torque profile in Fig. 14. Directly feeding this noisy signal to the PI controller causes frequent changes in the injection parameters, ultimately making the entire system unstable. Thus, the Kalman filter plays a critical role. The filtered torque can be seen in Fig. 14 with reduced noise.

Regarding the observations on the injection parameters, since there is no shift fault introduced in the injection time and start of injection (SOI), the control input generated from the controller and the control input applied to the engine are the same, as seen in Figs. 13(b) and 13(c). Before introducing the first pressure shift fault, the injection pressure generated from the controller and the injection pressure applied to the engine are the same. After the first fault is introduced, the injection pressures differ by 2 MPa, and after introducing the second fault, the injection pressures differ by 3 MPa. After introducing the first pressure shift fault, the injection pressure for the PI controller jumps and maintains a high steady-state injection pressure. Since the injection pressures for the PI controller have higher steady-state values, the injection time considerably reduces to maintain the engine torque tracking the setpoint torque. This can be seen from Fig. 13(b). The same trend can also be seen from the SOI in Fig. 13(c).

Further discussing the results of the proposed architecture, the MIL and HIL simulations demonstrate that the strategy can effectively handle highly non-linear systems. This effectiveness is primarily due to the precise control calibration map, which provides parameters that bring the system close to the desired conditions. By bringing the system closer to these conditions, non-linearities are avoided, allowing the feedback controller to compensate for minor deviations. Faster fault

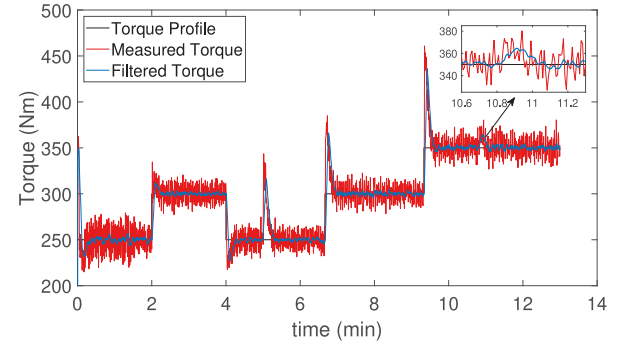


Fig. 14. Engine torque tracking performance in the presence of faults for HIL simulation.

recovery was observed in the MIL simulations compared to the HIL simulations, mainly due to noise in the torque sensor, which makes fault mitigation in HIL slightly slower. Moreover, based on the MIL and HIL results, it can be concluded that the proposed strategy is capable of overcoming multiple categories of faults with higher severity.

Here, the effectiveness of the proposed fault mitigation strategy is demonstrated on the heavy-duty Navistar engine. However, this approach can be easily extended to other engines by retuning the parameters from the hybrid engine model and regenerating an appropriate control calibration map.

7. Discussion

This section compares the proposed digital twin model and fault mitigation architecture with recent literature. Additionally, the assumptions and limitations of the proposed work are discussed.

7.1. Comparison with literature

To compare the accuracy of the digital twin model, a recently published marine diesel engine model by Hautala et al. (2023) is

Table 6

Comparison of the proposed fault mitigation architecture with existing literature.

	Alsuwian et al. (2022)	Escobar-Jiménez, Ocampo-Rodríguez, Osorio-Gordillo, and García-Beltrán (2024)	Riaz, Amin, and Tayyeb (2022)	Present work
FTC system	Hybrid FTC	Dual extended Kalman filter based FTC	Fuzzy logic controller based AFTC	Digital-twin based AFTC
Considered faults	Air-fuel ratio monitoring	Sensor faults	Air-fuel ratio monitoring	Injector nozzle clogging, intake manifold leaks, and pressure shift faults
Accuracy	Complete recovery	Complete recovery	85% of recovery	Complete recovery
Time	Less than 5 s	–	Less than a s	5–10 s
Overshoot	70%–80% and takes Time for stability	–	25% and quickly stabilizes	10%–30% and quickly stabilizes
Validation environment	MIL	Validated over experimental data	MIL	MIL and HIL
Limitations	Mitigate the faults only above threshold limit. Slightly unstable.	Highly dependent on the accuracy of sensors data	Mitigate the faults only above threshold limit	Reliant on the accuracy of the engine model

considered. This is a 1D engine model developed for a mid-speed marine engine. While 1D models are generally more accurate, they often come with a computational speed trade-off compared to 0D models. To address this, Hautala et al. incorporated model reduction strategies to create a fast-running engine model. The proposed hybrid engine model performs slightly better in terms of accuracy, demonstrating a 5% tolerance of error.

To the best of our knowledge, no FTC architectures have demonstrated the types of faulty scenarios discussed in this article for internal combustion engines. Most engine FTC architectures in the literature have focused on mitigating actuator and sensor faults. The detailed comparison with existing fault mitigation architectures is presented in Table 6. The last column of Table 6 highlights the key merits of the present work. As such, the proposed approach effectively and rapidly mitigates a wide variety of faults (e.g., different component failures, actuator faults, and sensor faults) with minimal fluctuations. Furthermore, validation in MIL and HIL environments demonstrates its applicability in real-time scenarios.

7.2. Assumptions and limitations of the proposed work

The architecture is developed on a heavy-duty engine specifically for autonomous ships. Ships generally operate under steady-state conditions for long durations; thus, the models are developed based on these assumptions and are not as effective for continuous variations in torque and speed profiles. The proposed controllers heavily rely on the accuracy of the engine model. The control calibration map initially guides fault mitigation and new operating conditions, while the feedback controller compensates for minor deviations due to faults and modeling errors. Therefore, calibration maps generated from high-fidelity models are crucial in effective operation. Finally, the models are developed for low-speed and low-torque scenarios to emulate the usual operating range for ships and ensure the safety of engine components during testing. Incorporating faults could make physical testing at higher torque dangerous and may lead to unsafe conditions for the engine.

8. Conclusions and future scope

This article proposes a computationally efficient, high-fidelity hybrid diesel engine model and a hierarchical control architecture for online fault mitigation. The hybrid engine model was utilized to develop a control calibration map and feedback controller for the fault mitigation architecture. The effectiveness of the fault mitigation strategies was validated in both the MIL and HIL environments for fuel injector and intake manifold faults.

The hybrid engine model showed an average accuracy of 96% compared to the experimental data. This high-fidelity and computationally efficient engine model demonstrated the suitability for controller modeling and can be extended further for designing advanced controllers. Novel mathematical models were provided for fuel injector erosion/clogging and manifold air leakages in diesel engines. The experimental validation of the proposed fault mitigation strategies showed a complete fault recovery in a few seconds, with minimal overshoot. Furthermore, a Python framework was designed to autonomously enable direct communication with the engine data acquisition system from the Python environment. The complete code for this Python framework is available on the GitHub.

Future scope: The proposed fault mitigation architecture is demonstrated for injector and manifold faults using a steady-state planner profile. However, this framework can be extended to address other components and sensor faults, incremental faults, and dynamic planner profiles. Another area for future research could involve expanding the hybrid engine model to develop a fault diagnosis system. This developed fault diagnosis component could then be integrated with the fault mitigation architecture. Simultaneously diagnosing and correcting the fault may facilitate smoother fault mitigation.

CRediT authorship contribution statement

Raj Pradip Khawale: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Dhrubajit Chowdhury:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Software, Writing – original draft. **Raman Goyal:** Conceptualization, Investigation, Methodology, Writing – original draft. **Shubhendu Kumar Singh:** Conceptualization, Investigation, Writing – review & editing. **Ankur Bhatt:** Data curation, Investigation, Methodology, Software, Writing – original draft. **Brian Gainey:** Conceptualization, Data curation, Writing – original draft. **Benjamin Lawler:** Conceptualization, Investigation, Data curation, Supervision, Writing – review & editing. **Lara Crawford:** Conceptualization, Funding acquisition, Investigation, Project administration, Supervision. **Rahul Rai:** Conceptualization, Funding acquisition, Investigation, Project administration, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing

Nomenclature

P_{atm}	Atmospheric pressure (Pa)
EOI	End of injection (Crank angle degree - CAD)
ρ	Fuel density (Kg/m ³)
R_e	Ideal-gas constant exhaust (J/Kg K)
R_a	Ideal-gas constant intake (J/Kg K)
C_d	Injector discharge coefficient (mm)
A_{flow}	Injector nozzle area (m ²)
W_c	Mass flow compressor (kg/s)
W_{ci}	Mass flow cylinder in (kg/s)
W_{co}	Mass flow cylinder out (kg/s)
W_f	Mass flow fuel (Kg/cycle/cylinder)
W_t	Mass flow turbine (kg/s)
T_r	Net torque (N m)
X_c	Oxygen concentration cylinder out (%)
O_f	Oxygen to fuel ratio
P_{cyl}	Pressure cylinder exhaust (Pa)
P_{W_t}	Power delivered by turbine (W)
P_{em}	Pressure exhaust manifold (Pa)
P_{im}	Pressure intake manifold (Pa)
P_{W_c}	Power required to drive compressor (W)
T_{amb}	Temperature ambient (K)
T_c	Temperature compressor (K)
T_{cyl}	Temperature cylinder exhaust (K)
T_{im}	Temperature cylinder intake (K)
T_{speed}	Turbo speed (rad/s)
V_{em}	Volume Exhaust manifold (m ³)
V_{im}	Volume intake manifold (m ³)

interests: Rahul Rai reports financial support was provided by Office of Naval Research. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Office of Naval Research Science of Artificial Intelligence program under contract N00014-20-C-1065.

References

- Abbaspour, Alireza, Mokhtari, Sohrab, Sargolzaei, Arman, & Yen, Kang K (2020). A survey on active fault-tolerant control systems. *Electronics*, 9(9), 1513.
- Adhikary, Bishwadipa Das, Reitz, Rolf D., & Ciatti, Stephen (2013). *Study of in-cylinder combustion and multi-cylinder light duty compression ignition engine performance using different RON fuels at light load conditions: Technical report*, SAE Technical Paper.
- Alsuiwan, Turki, Tayyeb, Muhammad, Amin, Arslan Ahmed, Qadir, Muhammad Bilal, Almasabi, Saleh, & Jalalah, Mohammed (2022). Design of a hybrid fault-tolerant control system for air-fuel ratio control of internal combustion engines using genetic algorithm and higher-order sliding mode control. *Energies*, 15(15), 5666.
- Amin, Arslan Ahmed, & Hasan, Khalid Mahmood (2019). A review of fault tolerant control systems: advancements and applications. *Measurement*, 143, 58–68.
- Amin, Arslan Ahmed, & Mahmood-ul Hasan, Khalid (2023). Robust passive fault tolerant control for air fuel ratio control of internal combustion gasoline engine for sensor and actuator faults. *IETE Journal of Research*, 69(5), 2846–2861.
- Bhave, Amit, Kraft, Markus, Montorsi, Luca, & Mauss, Fabian (2004). Modelling a dual-fuelled multi-cylinder HCCI engine using a PDF based engine cycle simulator. *SAE SP*, 87–100.
- Bo, Yaqing, Wu, Han, Che, Weifan, Zhang, Zeyu, Li, Xiangrong, & Myagkov, Leonid (2024). Methodology and application of digital twin-driven diesel engine fault diagnosis and virtual fault model acquisition. *Engineering Applications of Artificial Intelligence*, 131, Article 107853.
- Bondarenko, Oleksiy, & Fukuda, Tetsugo (2020). Development of a diesel engine's digital twin for predicting propulsion system dynamics. *Energy*, 196, Article 117126.
- Chowdhury, Dhruvajit, Goyal, Raman, Khawale, Raj Pradip, Crawford, Lara, & Rai, Rahul (2023). Adaptive online fault mitigation using hierarchical engine control. In *2023 American control conference* (pp. 2371–2378). IEEE.

- Cornolti, Luca, Onorati, Angelo, Cerri, Tarcisio, Montenegro, Gianluca, & Piscaglia, Federico (2013). 1D simulation of a turbocharged diesel engine with comparison of short and long EGR route solutions. *Applied Energy*, 111, 1–15.
- Ding, Steven X. (2021). *Advanced methods for fault diagnosis and fault-tolerant control*. Springer.
- Ebenezer, Nitla Stanley, khurshid, Abdul, Devi, K Anjani, Sandeep, Chodiseti Naga, Manipal, Penke Pragnana, Vijay, Gorthi Siva, et al. (2021). Self-automated fault diagnosis system for internal combustion engines. In *Computer communication, networking and IoT: proceedings of ICICC 2020* (pp. 329–337).
- Escobar-Jiménez, RF, Ocampo-Rodríguez, LE, Osorio-Gordillo, GL, & García-Beltrán, CD (2024). Sensor fault-tolerant control for an internal combustion engine. *International Journal of Adaptive Control and Signal Processing*, 38(2), 475–491.
- Frazier, Peter I. (2018). A tutorial on Bayesian optimization. arXiv preprint arXiv: 1807.02811.
- Gainey, Brian, & Lawler, Benjamin (2021). A fuel cell free piston gas turbine hybrid architecture for high-efficiency, load-flexible power generation. *Applied Energy*, 283, Article 116242.
- Garcia, Antonio, Monsalve-Serrano, Javier, Villalta, David, & Fogué-Robles, Álvaro (2022). Detailed analysis of particulate emissions of a multi-cylinder dual-mode dual-fuel engine operating with diesel and gasoline. *Fuel*, 330, Article 125578.
- Goyal, Raman, Chowdhury, Dhruvajit, Hazarika, Subhashis, Khawale, Raj Pradip, Singh, Shubhendu Kumar, Crawford, Lara, et al. (2023). Hybrid machine learning and autonomous control assisted framework for fault diagnostics and mitigation in diesel engines. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 325–339). Springer.
- Gustafsson, Nils, Nyberg, Leif, & Omstedt, Anders (1998). Coupling of a high-resolution atmospheric model and an ocean model for the Baltic sea. *Monthly Weather Review*, 126(11), 2822–2846.
- Gutiérrez León, P, García-Morales, J, Escobar-Jiménez, RF, Gómez-Aguilar, JF, López-López, G, & Torres, L (2018). Implementation of a fault tolerant system for the internal combustion engine's MAF sensor. *Measurement*, 122, 91–99.
- Hautala, Saana, Mikulski, Maciej, Söderäng, Emma, Storm, Xiaoguo, & Niemi, Seppo (2023). Toward a digital twin of a mid-speed marine engine: From detailed 1D engine model to real-time implementation on a target platform. *International Journal of Engine Research*, 24(12), 4553–4571.
- Heywood, J. B. (1988). *Internal combustion engine fundamentals*. McGraw-hill.
- Hong, Sangjin, Wooldridge, Margaret S, Im, Hong G, Assanis, Dennis N, & Pitsch, Heinz (2005). Development and application of a comprehensive soot model for 3D CFD reacting flow studies in a diesel engine. *Combustion and Flame*, 143(1–2), 11–26.
- Jeffrey, Carl, Cutajar, Reuben, Prosser, Stephen, Lickess, M, Richardson, Andrew, & Riches, Stephen (2005). The integration of on-line monitoring and reconfiguration functions using IEEE1149. 4 into a safety critical automotive electronic control unit. In *Design, automation and test in europe* (pp. 153–158). IEEE.
- Jiang, Jin, & Yu, Xiang (2012). Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in Control*, 36(1), 60–72.
- Kaviarasan, B., Sakthivel, R., & Kwon, O. M. (2016). Robust fault-tolerant control for power systems against mixed actuator failures. *Nonlinear Analysis. Hybrid Systems*, 22, 249–261.
- Khair, Ummul, Fahmi, Hasanul, Al Hakim, Sarudin, & Rahim, Robbi (2017). Forecasting error calculation with mean absolute deviation and mean absolute percentage error. vol. 930, In *Journal of physics: conference series*. IOP Publishing, Article 012002.
- Khatibi, Mahmood, & Haeri, Mohammad (2019). A unified framework for passive-active fault-tolerant control systems considering actuator saturation and I infinity disturbances. *International Journal of Control*, 92(3), 653–663.
- Khodarahmi, Masoud, & Maihami, Vafa (2023). A review on Kalman filter models. *Archives of Computational Methods in Engineering*, 30(1), 727–747.
- Li, Juan, Zhang, Luyao, Luo, Lin, & Li, Shengquan (2023). Extended state observer based current-constrained controller for a PMSM system in presence of disturbances: Design, analysis and experiments. *Control Engineering Practice*, 132, Article 105412.
- Lino, Paolo, Maione, Bruno, & Rizzo, Alessandro (2005). A control-oriented model of a common rail injection system for diesel engines. vol. 1, In *2005 IEEE conference on emerging technologies and factory automation* (p. 7). IEEE.
- Lv, Yaqiong, Yang, Xueting, Li, Yifan, Liu, Jialun, & Li, Shijie (2024). Fault detection and diagnosis of marine diesel engines: A systematic review. *Ocean Engineering*, 294, Article 116798.
- Lyubarsky, Pavlo, & Bartel, Dirk (2016). 2D CFD-model of the piston assembly in a diesel engine for the analysis of piston ring dynamics, mass transport and friction. *Tribology International*, 104, 352–368.
- Markovsky, Ivan, & Van Huffel, Sabine (2007). Overview of total least-squares methods. *Signal Processing*, 87(10), 2283–2302.
- McBride, B. J. (2002). NASA glenn coefficients for calculating thermodynamic properties of individual species. *NASA/TP*, 2002, Article 211556.
- Miyamoto, Noboru, Chikahisa, Takemi, Murayama, Tadashi, & Sawyer, Robert (1985). Description and analysis of diesel engine rate of combustion and performance using Wiebe's functions. *SAE Transactions*, 622–633.
- Mu, Yunfei, Zhang, Huaguang, Xi, Ruipeng, Wang, Zhiliang, & Sun, Jiayue (2021). Fault-tolerant control of nonlinear systems with actuator and sensor faults based on T-S fuzzy model and fuzzy observer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(9), 5795–5804.

- Ou, Wenquan, Wang, Chun, Tang, Aihua, Huang, Bo, & Liu, Kang (2023). Multistate joint estimation of ultracapacitor based on trans-scale dual extended Kalman filter. *Control Engineering Practice*, 137, Article 105555.
- Payri, Francisco, Olmeda, Pablo, Martín, Jaime, & García, Antonio (2011). A complete 0D thermodynamic predictive model for direct injection diesel engines. *Applied Energy*, 88(12), 4632–4641.
- Riaz, Umar, Amin, Arslan Ahmed, & Tayyeb, Muhammad (2022). Design of active fault-tolerant control system for air-fuel ratio control of internal combustion engines using fuzzy logic controller. *Science Progress*, 105(2), Article 00368504221094723.
- Saffar, Esmail, Ghanbari, Mahmood, Ebrahimi, Reza, & Jannati, Mohammad (2023). A simple fault-tolerant control method for open-phase three-phase induction motor drives. *Control Engineering Practice*, 136, Article 105525.
- Sakthivel, Rathinasamy, Joby, Maya, Wang, Chao, & Kaviarasan, B (2018). Finite-time fault-tolerant control of neutral systems against actuator saturation and nonlinear actuator faults. *Applied Mathematics and Computation*, 332, 425–436.
- Schiavo, Michele, Padula, Fabrizio, Latronico, Nicola, Paltenghi, Massimiliano, & Visioli, Antonio (2023). Experimental results of an event-based PID control system for propofol and remifentanyl coadministration. *Control Engineering Practice*, 131, Article 105384.
- Shahbaz, Muhammad Hamza, & Amin, Arslan Ahmed (2023). Design of hybrid fault-tolerant control system for air-fuel ratio control of internal combustion engines using artificial neural network and sliding mode control against sensor faults. *Advances in Mechanical Engineering*, 15(3), Article 16878132231160729.
- Shahriari, Bobak, Swersky, Kevin, Wang, Ziyu, Adams, Ryan P, & De Freitas, Nando (2015). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.
- Shields, Michael D., & Zhang, Jiaxin (2016). The generalization of Latin hypercube sampling. *Reliability Engineering & System Safety*, 148, 96–108.
- Singh, Shubhendu Kumar, Khawale, Raj Pradip, Hazarika, Subhashis, Bhatt, Ankur, Gainey, Brian, Lawler, Benjamin, et al. (2024). Hybrid physics-infused 1D-CNN based deep learning framework for diesel engine fault diagnostics. *Neural Computing and Applications*, 1–29.
- Singh, Shubhendu Kumar, Rai, Rahul, Khawale, Raj Pradip, Patel, Darshil, Bielecki, Dustin, Nguyen, Ryan, et al. (2023). Deep learning in computational design synthesis: A comprehensive review. *Journal of Computing and Information Science in Engineering*, 1–39.
- Singh, Dalwinder, & Singh, Birmohan (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97, Article 105524.
- Söderäng, Emma, Hautala, Saana, Mikulski, Maciej, Storm, Xiaoguo, & Niemi, Seppo (2022). Development of a digital twin for real-time simulation of a combustion engine-based power plant with battery storage and grid coupling. *Energy Conversion and Management*, 266, Article 115793.
- Song, Jiahao, & He, Xiao (2023). Robust state estimation and fault detection for autonomous underwater vehicles considering hydrodynamic effects. *Control Engineering Practice*, 135, Article 105497.
- Souami, Yani, Mechbal, Nazih, & Ecoutin, Stephane (2015). Robust passive fault tolerant control applied to a fuel metering valve of a jet engine. In *2015 IEEE aerospace conference* (pp. 1–12). IEEE.
- Souder, Jason S, Mehresh, Parag, Hedrick, J Karl, & Dibble, Robert W (2004). A multi-cylinder HCCI engine model for control. vol. 47063, In *ASME international mechanical engineering congress and exposition* (pp. 307–316).
- Tang, Yuanyuan, Zhang, Jundong, Gan, Huibing, Jia, Baozhu, & Xia, Yu (2017). Development of a real-time two-stroke marine diesel engine model with in-cylinder pressure prediction capability. *Applied Energy*, 194, 55–70.
- Tang, Jian, Zhu, Guoming G., & Men, Yifan (2022). Review of engine control-oriented combustion models. *International Journal of Engine Research*, 23(3), 347–368.
- Vortmann, C., Schnerr, G. H., & Seelecke, S. (2003). Thermodynamic modeling and simulation of cavitating nozzle flow. *International Journal of Heat and Fluid Flow*, 24(5), 774–783.
- Wahlström, Johan (2006). *Control of EGR and VGT for emission control and pumping work minimization in diesel engines* (Ph.D. thesis), Institutionen för systemteknik.
- Wahlström, Johan, & Eriksson, Lars (2011). Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proceedings of the Institution of Mechanical Engineers, Part D (Journal of Automobile Engineering)*, 225(7), 960–986.
- Wang, Qing-Guo, Sun, Jitao, Zhang, Jin-Xi, Huang, Jiangshuai, Yu, Jinpeng, & Dong, Huiwen (2023). Survey of transient performance control. *Control Engineering Practice*, 138, Article 105559.
- Wang, Rongrong, & Wang, Junmin (2012). Passive actuator fault-tolerant control for a class of overactuated nonlinear systems and applications to electric vehicles. *IEEE Transactions on Vehicular Technology*, 62(3), 972–985.
- Wang, Jun, Yao, Xiaowan, & Li, Wei (2016). Hybrid active-passive fault-tolerant control of nonlinear NCS based on event-triggered communication scheme. In *2nd international conference on electronics, network and computer engineering (ICENCE 2016)* (pp. 327–335). Atlantis Press.
- Wu, Dehao, Zhou, Donghua, & Chen, Maoyin (2021). Performance-driven component selection in the framework of PCA for process monitoring: A dynamic selection approach. *IEEE Transactions on Control Systems Technology*, 30(3), 1171–1185.
- Yin, Lianhao, Turesson, Gabriel, Tunestål, Per, & Johansson, Rolf (2020). Model predictive control of an advanced multiple cylinder engine with partially premixed combustion concept. *IEEE/ASME Transactions on Mechatronics*, 25(2), 804–814.
- Yuan, Ye, Liu, Xiaofeng, Ding, Shuiting, & Pan, Bochao (2017). Fault detection and location system for diagnosis of multiple faults in aeroengines. *IEEE Access*, 5, 17671–17677.
- Zhang, Jiyu, Amodio, Alessandro, Li, Tianpei, Aksun-Güvenç, Bilin, & Rizzoni, Giorgio (2018). Fault diagnosis and fault mitigation for torque safety of drive-by-wire systems. *IEEE Transactions on Vehicular Technology*, 67(9), 8041–8054.