



Hybrid Machine Learning and Autonomous Control Assisted Framework for Fault Diagnostics and Mitigation in Diesel Engines

Raman Goyal¹(✉), Dhrubajit Chowdhury¹, Subhashis Hazarika¹,
Raj Pradip Khawale², Shubhendu Kumar Singh², Lara Crawford¹,
and Rahul Rai²

¹ Palo Alto Research Center, Palo Alto, CA, USA

{rgoyal, chowdhury, shazarika, lcrawford}@parc.com

² Department of Automotive Engineering, Clemson University, Clemson, SC, USA

{rkhwawal, shubhes, rrai}@clemson.edu

Abstract. The paper proposes a hybrid machine learning framework along with a hierarchical control module for fault diagnosis, isolation, and mitigation to develop a resilient diesel engine system. The hybrid diagnostics system combines experimental data with physics-based simulation data to improve fault diagnosis, isolation, and severity prediction. The hybrid architecture consists of a denoising autoencoder to transform the engine data to a fixed lower-dimension latent space representation. The combined data is then passed to a Twin-Deep Neural Network (DNN) framework to detect and predict fault severity. The hierarchical control module consists of control calibration maps generated offline using Bayesian optimization to maintain the desired engine torque while minimizing fuel consumption. The module also uses proportional-integral (PI) and extremum seeking (ES) controllers on top of the offline map to compensate for engine faults and modeling errors. The simulation results show the efficacy of the proposed architecture to maintain the desired performance for different fault scenarios.

Keywords: Machine Learning · Neural Network · Engine Diagnostics · Fault mitigation · Bayesian Optimization · Autoencoder · Engine Control

1 Introduction

The concept of full autonomy is still a major challenge, especially in the realm of autonomous vessels. The US Navy aspires to create self-sufficient ships that are capable of resolving faults without human intervention. This requirement extends to other autonomous vehicles and systems that must make decisions on their own to achieve specific goals. However, there still exists a gap in the ability

of autonomous systems to respond and adapt to faults, as they lack the necessary models and decision-making processes. The driving force behind this research is the necessity to autonomously address engine faults to ensure mission completion [15]. The paper aims to develop a comprehensive online fault detection and mitigation framework for marine engines. The comprehensive online framework would help in the successful completion of the mission in adverse environments.

Despite diesel engines being a popular choice for maritime vessels because of their dependability and efficiency, they are still prone to faults resulting in reduced performance, increased emissions, and even catastrophic failure. Fault diagnosis is critical to ensure the safe and efficient operation of diesel engines. Traditional diagnostic methods rely on expert knowledge and manual inspection, which are time-consuming, costly, and limited in their accuracy. Various advanced fault diagnosis techniques, such as model-based and data-driven approaches, have emerged in recent years. Model-based approaches rely on mathematical models that describe the physical behavior of the diesel engine under normal and faulty conditions. These models are based on fundamental principles and are used to simulate the engine's response to different operating conditions and detect faults by analyzing the deviation of residuals from a pre-defined threshold [1]. Nohra et al. [8] suggested a linearized linear time-invariant model based on mu-analysis control theory, which effectively detects and isolates faults in turbocharged diesel engines, even in the presence of noise and uncertainties. These models are developed using analytical or numerical methods and can provide insights into the underlying mechanisms of fault propagation. However, they require detailed knowledge of the engine's structure and parameters and may only be accurate under some operating conditions and fault types.

Contrary to the model-based approach, data-driven models are based on the statistical analysis of the data collected from the engine under normal and faulty conditions. These models can capture the complex and nonlinear relationships between the diagnostic signals and the fault. Rahimi et al. [11] devised an SVM and RBF-based approach to explore the impact of oil metal contamination on the operational conditions of diesel engines using extensive datasets. Guoqiang et al. [2] introduced a deep belief network technique for intelligent fault diagnosis of marine diesel engines. However, data-driven approaches usually need more interpretability and generalizability and require a large amount of training data.

To overcome the limitations faced by the traditional model-based and data-driven approaches, researchers are resorting to a new class of methods termed Physics-Informed Machine Learning (PIML) [5]. PIML aims to enhance ML models' accuracy, interpretability, and generalizability by incorporating domain-specific knowledge and constraints into the learning process. In the context of diesel engine fault diagnosis, PIML can leverage the physical laws governing the engine's operation to guide the feature extraction, model selection, and decision-making steps of the ML pipeline. PIML can also provide insight into the underlying mechanisms of fault propagation and enable the design of more effective diagnostic strategies. In this research, we propose a hybrid physics-informed twin neural network framework for fault detection and severity prediction in a diesel engine system.

Once the faults are identified, the engine needs to autonomously mitigate the effect of faults. One aspect of these faults can be decreased generation of torque that can be compensated by controlling the fuel quantity and timing in the combustion phase. The fuel injection system uses electronic control to change the fuel quantity and timing based on a calibrated lookup table. The lookup table stores the values of engine control parameters, which are optimized to achieve maximum fuel efficiency for different engine operating conditions. This control structure is open-loop in nature as the engine control parameters are optimized offline [14]. This structure suffers in practice due to differences in engine modeling errors, non-calibrated operating points, and faults, which necessitates the need for online feedback control [14]. Extremum seeking (ES) controllers have been used for online calibration of the engine control parameters due to their model-free nature [7]. Some of the implementations of ES controller for engine optimization include [3, 6, 10], and a comprehensive review of the optimization algorithms used for engine calibration can be found in [17]. This paper will develop a hierarchical control framework that will pool offline and online controllers to mitigate the effect of faults and modeling errors while optimizing for fuel consumption.

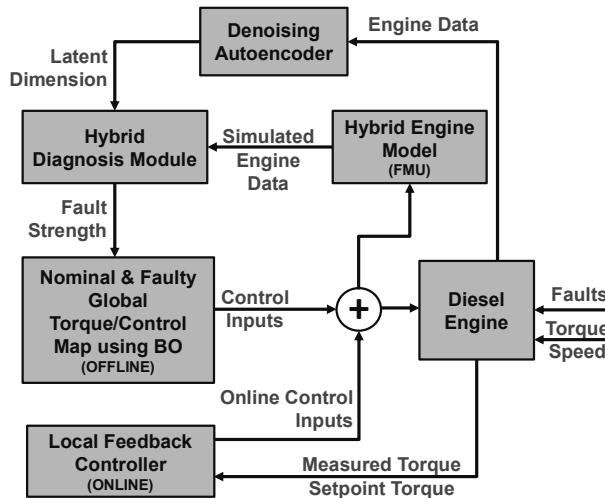


Fig. 1. The complete framework for fault detection and mitigation.

Main Contribution: The paper provides a comprehensive framework for autonomous fault detection and mitigation in a diesel engine system as shown in Fig. 1. The paper first develops a hybrid (analytical and ML-based) engine model tuned using experimental data. Then, a Denoising Autoencoder (DAE) is developed to remove the noise and to represent the system in a fixed-dimension latent representation. A machine learning based hybrid Twin-Deep Neural Network (DNN) framework is developed to diagnose faults and predict fault severity.

The paper further develops a hierarchical control module that provides setpoint control parameters and an online feedback controller to mitigate faults. The setpoint control parameters are obtained offline using Bayesian Optimization (BO) to minimize fuel consumption and the online feedback control is based on proportional-integral (PI) and Extremum seeking (ES) controllers. The paper finally shows the full integration results of fault mitigation for the fuel injector and intake manifold leak faults.

2 Hybrid Engine and Fault Model

The paper develops a high-fidelity hybrid engine model for a 7.6-liter 6-cylinder Navistar diesel engine installed at Clemson University. There exist multiple mean value models with various complexities and states in the literature [4,13]. The paper uses Wahlström and Eriksson [16] as the baseline model and improves the fidelity of the model by replacing and adding various components. We started with a detailed thermodynamic cylinder model that simulates the in-cylinder processes and their thermodynamic states but the detailed model was computationally expensive. Thus, to improve the computational speed of the overall physics-based model, the detailed thermodynamic cylinder model is replaced with a NN reducing the computational simulation time by an order of magnitude. This network is trained over the dataset generated from the thermodynamic cylinder model. The dataset was generated over an entire feasible range of input cylinder parameters. This surrogate cylinder model showed the Mean Squared Error (MSE) of 3.42×10^{-4} over the test dataset. Moreover, the outputs of the cylinder model give us a difference of less than 5% with experimental data. The intake and exhaust manifold systems are modeled using the principle of mass conservation and the ideal-gas law [16]. The turbocharger model is composed of a turbine model, a compressor model, and a turbo inertia model. Finally, the entire model is depicted in Fig. 2 (refer to [16] for more details) and is calibrated using experimental data for engine speed and torque ranges between

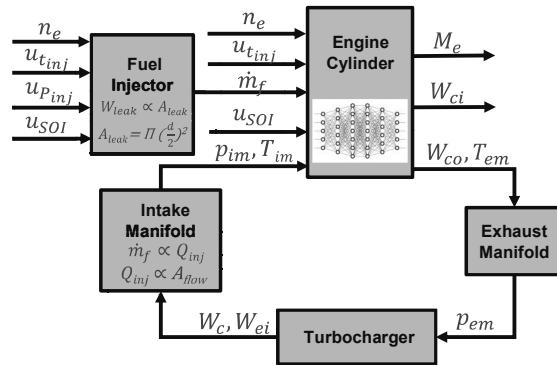


Fig. 2. Hybrid engine model structural diagram

800–2000 RPM and 100–600 Nm, respectively. The control input parameters are : engine speed (n_e), injection pressure (u_{Pinj}), injection duration ($u_{t_{inj}}$), and start of injection (u_{SOI}).

2.1 Different Fault Models

This subsection provides a brief description of intake manifold faults modeled using a pinhole leak and fuel injection system faults like fuel injector nozzle erosion due to cavitation and fuel injector nozzle clogging, which can be simulated by increasing and decreasing the area of the injector nozzle, respectively.

Intake Manifold Leak Model: The intake manifold is connected to the cylinder head using a seal to keep it airtight. However, over time these seals might wear off which introduces a leak in the intake manifold and as a result decreases the engine efficiency. The leak is modeled as a flow through a restriction and has been validated in [9] with good accuracy. The leakage mass flow rate from the intake manifold is:

$$W_{\text{leak}} = \frac{A_{\text{leak}} p_{im}}{\sqrt{R_{im} T_{im}}} \psi_{\kappa} \left(\frac{p_{\text{atm}}}{p_{im}} \right) \quad (1)$$

where R_{im} is the gas constant of air in the intake manifold, T_{im} is the temperature in the intake manifold, p_{im} is pressure in the intake manifold, A_{leak} is the area of the hole given by $\pi(d/2)^2$ with d as leak diameter, and the function ψ_{κ} is a complex function given in [9].

Fuel Injector Area Erosion/Clogging Model: The Navistar engine has a high-pressure common-rail fuel injection system and the corresponding injection flow rate is modeled as follows:

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

where α is a binary value used to incorporate injection duration, P_{cyl} is the cylinder pressure, C_d is the discharge coefficient for injector, A_{fl} is the injector nozzle area, and ρ is the fuel density. 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.

3 Online Hybrid Diagnostics Module for Fault Prediction

3.1 Denoising Autoencoder

The sensor readings collected from an engine can be a challenging dataset to directly ingest into a machine learning pipeline. The engine data needs to be checked for data consistency, in terms of missing or corrupted values and different noise levels. We address such issues through our data assimilation step which preprocesses the data and makes it ready for downstream hybrid machine learning models. A key component of the data assimilation stage is a denoising Autoencoder (DAE) model. Autoencoders (AE) are neural network models commonly

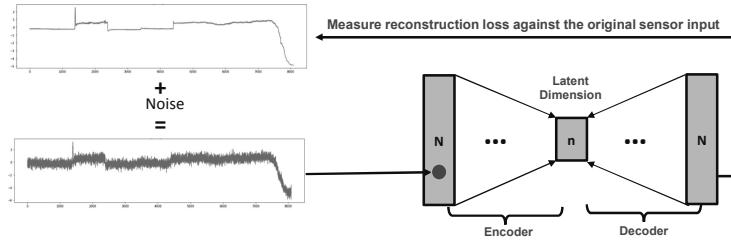


Fig. 3. Overview of the Denoising Autoencoder model.

used to model high-dimensional multivariate data signals. They learn the inter-variable relationships and project them to representative low-dimensional latent spaces. Compared to popular dimensionality reduction methods like principal component analysis (PCA), which are linear models, AE's can capture non-linear multivariate relationships. AE models comprise of an *encoder* and a *decoder* network. They are trained in such a way that the encoder takes in an input high-dimensional vector and produces a low-dimensional latent representation, whereas, the decoder takes in the latent vector and tries to reconstruct the original high-dimensional vector back. DAE's are a popular variant of AE, which take in high-dimensional noisy signals and reconstruct the original signal back as illustrated in Fig. 3. DAE learns to map the high-dimensional data space to latent space in the presence of noise. We utilize this property of DAE to learn the simplified latent representation of the multivariate engine data in the presence of noise. The mean squared error of reconstruction was used as the standard loss function to train our model. We trained the DAE to a maximum noise level of 30%. The generated latent representation capturing the important multivariate relationships is then used in the hybrid model development phase. The DAE training plot is given in Fig. 4(a) and the reconstruction of the torque variable is shown in Fig. 4(b).

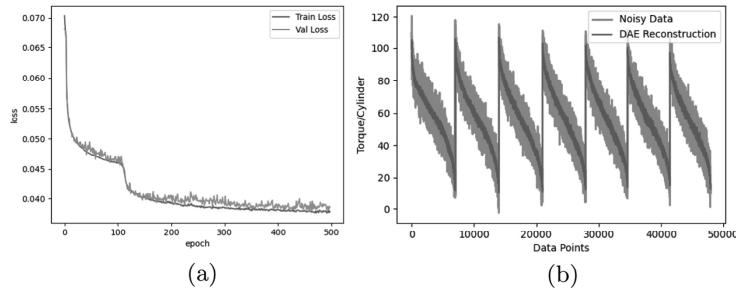


Fig. 4. (a) Training and validation loss for the training of the DAE. (b) The plot shows the noisy data and the reconstructed data for the cylinder torque for all the training data. The plot repeats itself for different values of RPMs.

3.2 Twin-Neural Network Framework for Fault Diagnosis

The proposed hybrid diagnosis framework, as shown in Fig. 5, is based on a twin neural network structure, where one deep neural network (DNN) based sub-model classifies fault while the other DNN-based regression model was used to estimate the severity of the fault. The input to both the DNN-based sub-models is a 42-dimensional feature vector obtained by concatenating a 4-dimensional latent variables vector from the denoising autoencoder and 38-dimensional data from the multi-physics simulation model of the diesel engine. The influx of physics into the neural network-based sub-models ensures that their solutions are physically meaningful and consistent with the underlying physical phenomena. Additionally, the physics of the system helps to regularize the DNN models, thus reducing overfitting, improving generalization, and ultimately enhancing the overall accuracy of the fault diagnosis framework. And the data-driven DNN models help to extract patterns and relationships from the dataset that may not be immediately apparent from the underlying physics. Besides, the DNN models help to identify outliers in the data that may indicate a fault or anomaly in the system. Combining this information with the physical constraints of physics-based models enables the hybrid model to diagnose faults accurately.

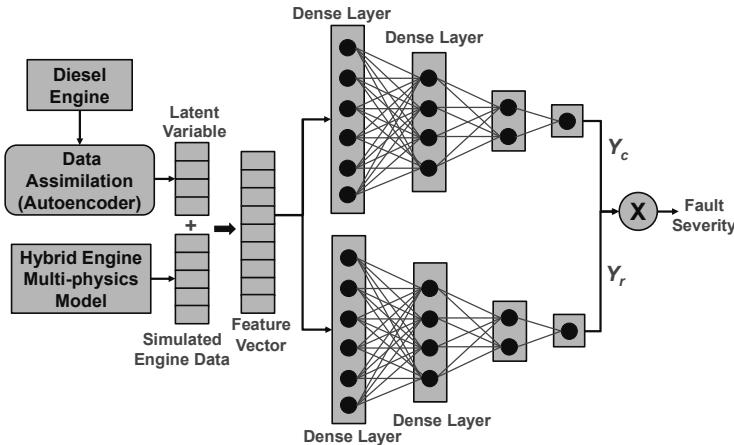


Fig. 5. Architecture for the hybrid diagnosis module.

As evident from Fig. 5, the top DNN model is the classification model whose output is Y_c , and the bottom model is the regression model with Y_r as its output. The two DNN models are trained separately with the output of the classification model as $Y_c = \{0, 1\}$, where $Y_c = 0$ represents the nominal condition and $Y_c = 1$ represents the fault condition. Fig. 6 shows the confusion matrix for the classification of intake manifold leaks and fuel injector faults. Notice that the intake leak fault is harder to detect and has higher false positives compared to

the fuel injector faults, which have a direct effect on the amount of fuel injected and thus generated torque.

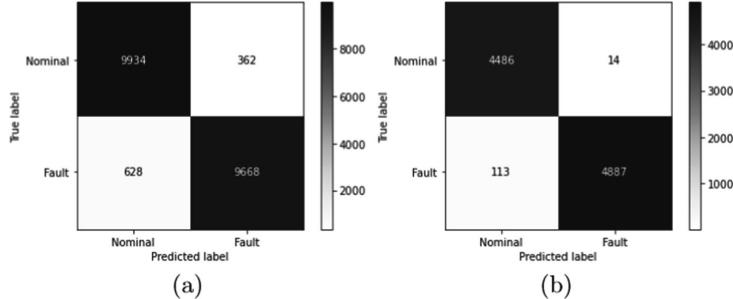


Fig. 6. (a) The confusion matrix for classification of intake manifold leaks faults. (b) The confusion matrix for classification of fuel injector faults.

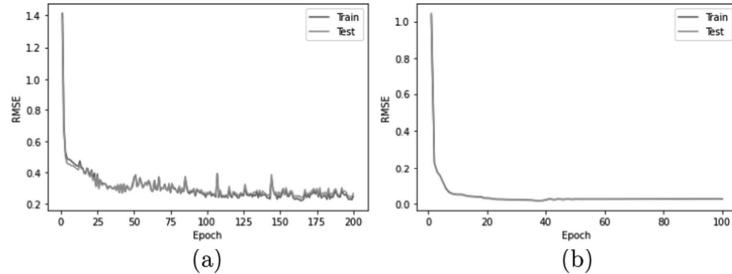


Fig. 7. The regression model training plots for (a) intake manifold leaks faults and (b) fuel injector faults.

The regression model output takes on continuous values from $Y_r = [0.5 - 1.5]$ for the case of fuel injector area clogging and $Y_r = [0 - 0.012]$ for the case of intake manifold leak. Figure 7 shows the training plots for the regression model of intake manifold leaks and fuel injector faults. The final output of the hybrid diagnosis module is the fault severity $Y_c \times Y_r$.

The reason for training the two models separately and using the multiplication of the two outputs is to reduce the chances of incorrect fault prediction for the nominal operating condition, as the regression model can have an incorrect small fault severity prediction due to overfitting. The twin neural network structure of the diagnostics framework helps in achieving higher accuracy and faster convergence compared to a single large neural network as each individual network in the framework can specialize in learning different features or patterns, allowing for more efficient and accurate learning.

4 Hierarchical Control Module for Fault Mitigation

The paper uses two model-free controllers as part of the hierarchical control module for fault mitigation. The first controller uses the Bayesian optimization (BO) algorithm to generate offline control calibration maps that minimize fuel consumption and obtain the desired engine torque. The second is the online extremum seeking (ES) feedback controller that is used on top of the nominal optimal engine control parameters found using BO to compensate for modeling errors and engine faults. This section provides a description of these two algorithms.

4.1 Control Calibration Maps for Different Fault Strength Using Bayesian Optimization

We used the BO algorithm to generate the control calibration maps for both nominal and faulty operating conditions. The BO is generally used to find the global optimum of a computationally expensive objective function $f(\mathbf{x})$ without a closed form expression [12]. The problem can be defined as: $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$, where \mathcal{X} is the design space of interest. The BO develops a Gaussian Process based surrogate model which is initialized with a prior belief about the behavior of the unknown objective function. The model is then sequentially refined with new data when the function is evaluated at new query points using a Bayesian posterior update. Acquisitions functions are used in the BO framework to provide a trade-off between exploration and exploitation and thus provide the next query point for function evaluation $f(\mathbf{x})$.

We used BO to generate the control calibration maps by optimizing the fuel efficiency \dot{m}_f for a combination of different engine operating points with engine speeds ranging from 1000 to 1600 RPM with a step of 1000 RPM and torque values ranging from 100 to 500 Nm with a step of 50 Nm. The maps are also generated for the fuel injector area faults ranging from $A_{ff} = 0.5$ to $A_{ff} = 1.5$ with a step of 0.1 and for the intake manifold leak with the leak diameter varying from $d = 2\text{mm}$ to $d = 12\text{mm}$ with a step of 2mm. The optimization problem for engine calibration can be described as:

$$\min_{u_1, u_2, u_3} \dot{m}_f(u_1, u_2, u_3, n_e, M_e) \quad \text{s.t. } M_e = M_e^s, n_e = n_e^s, \underline{u}_i \leq u_i \leq \bar{u}_i, \quad (3)$$

where $i = 1, 2, 3$ and $u_1 = u_{Pinj}$, $u_2 = u_{tinj}$, $u_3 = u_{SOI}$, are controllable engine parameters. The variables n_e^s , and M_e^s are the setpoint speed and torque. In the actual implementation, the torque constraint is relaxed to inequality constraints with a small number $\delta = 0.3$ Nm. The ranges for control calibration variables used in the simulation setup are as follows: $u_{Pinj} \in [4 - 25]\text{MPa}$, $u_{tinj} \in [0.3 - 3]\text{ms}$, and $u_{SOI} \in [345 - 380]\text{CAD}$. Figure 8(a) illustrates the estimated objective function (\dot{m}_f) obtained by running the BO algorithm for 150 iterations by varying the injection pressure and SOI.

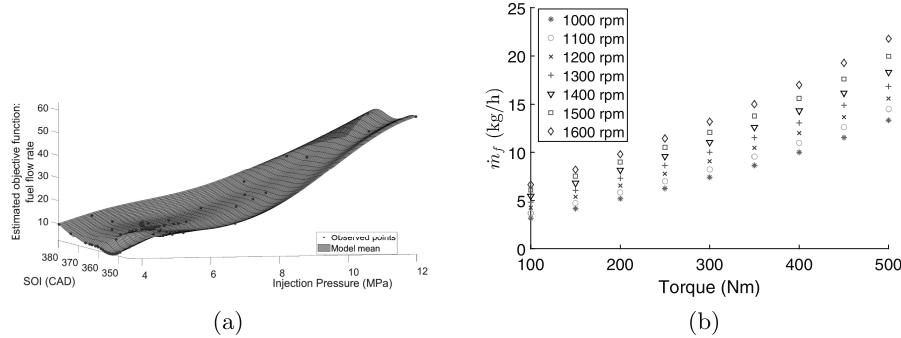


Fig. 8. (a) Objective function mean estimated by Bayesian optimization. (b) Optimized fuel vs torque map generated using BO for a range of torque and engine RPM.

Figure 8(b) illustrates the optimized fuel consumption rate for all the operating points. Notice that the fuel consumption rate increases with the increase in torque and engine speed. Figure 9(a) compares the obtained injection pressures for the nominal case with the area clogging in the fuel injector. As seen from the figure, with an increase in area clogging the optimized injection pressure increases as higher injection pressures are required to generate the same fuel flow rate as in the nominal case. Figure 9(b) compares the optimized fuel flow rate for the nominal case with the presence of a leak in the intake manifold. Notice that the fuel flow rate increases with an increase in speed and with an increase in leak diameter. This is because larger leaks result in a higher drop in torque value and as a result, a higher fuel flow rate is required to achieve the same torque as in the nominal case.

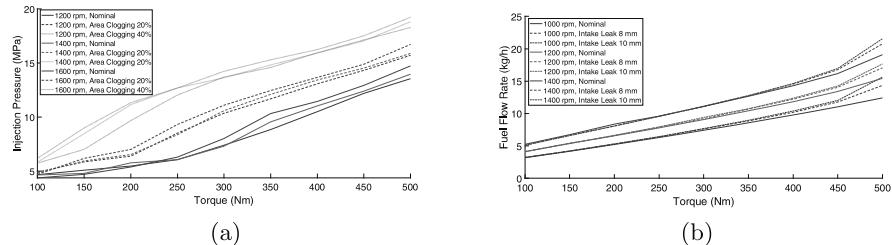


Fig. 9. (a) Comparison of optimized injection pressure vs torque map in the presence/absence of area clogging. (b) Comparison of optimized fuel vs torque map in the presence/absence of intake manifold leaks.

4.2 Online Feedback Controller for Fault Mitigation

This subsection describes a PI+ES online feedback controller for optimization of the engine control parameters in cases of modeling errors and engine faults.

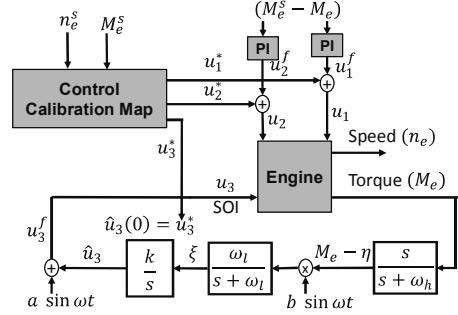


Fig. 10. Control architecture for online engine calibration

The ES controller [7] is a model-free approach for adaptive control, which is used for systems where the input-to-output map $f(\mathbf{x})$ is unknown but is known to have an extremum. This online feedback controller is used in conjunction with the calibration maps as the optimized control inputs might not achieve the desired torque setpoint M_e^s due to modeling errors and uncalibrated engine faults. The overall control architecture is shown in Fig. 10. The PI controller uses the tracking error $M_e^s - M_e$ to change the injection duration and pressure to maintain the desired setpoint torque and the ES controller to change the SOI to minimize the fuel consumption rate by maximizing the torque M_e . The PI controllers for injection pressure and duration are described as u_1^f and u_2^f and the ES controller can be described as: $u_3^f = \hat{u}_3 + a \sin \omega t$, $\dot{\eta} = -\omega_h \eta + \omega_h M_e$, $\dot{\xi} = -\omega_l \xi + \omega_l (M_e - \eta) b \sin \omega t$, $\dot{\hat{u}}_3 = k \xi$, where ω_h , ω_l , and ω are the frequencies of the high-pass filter, low-pass filter, and perturbation signal. The modulation amplitude is given by a , and the demodulation amplitude is given by b . The adaptation gain is given by k , and the low-pass and high-pass filter states are given by ξ and η . Finally, \hat{u}_3 represents the mean perturbation. The initial condition of the integrator is chosen as $\hat{u}_3(0) = u_3^*$. The final engine control parameters then fed to the engine are given as: $u_1 = u_1^f + u_1^*$, $u_2 = u_2^f + u_2^*$, $u_3 = u_3^f$, where u_1^* , u_2^* , and u_3^* represent the engine control parameters generated from the engine control map.

5 Integrated Simulation Results

This section shows the results of the complete framework of fault detection and mitigation for the case of two kinds of faults considered in the paper.

5.1 Fuel Injector Clogging

The simulation results were generated for the setpoint torque of 300 Nm and speed of 1400 RPM with injection area fault of value $A_{ff} = [1, 1, 0.6, 1, 1, 0.8]$ introduced at time $t = 8$ second. The diagnosis module is run every 5 secs.

Figure 11 shows the torque, fuel flow rate, and control injection parameters plot where the engine takes around 2 secs to reach the steady state value and then a sudden decrease in torque value is observed due to injector clogging in cylinders 3 and 6 at time $t = 8$ sec. When the torque value deviates from the setpoint torque value of 300 Nm, the online feedback control adjusts the control parameters to obtain the desired torque. Notice that at $t = 5$ secs, there is no fault and hence no change in injection parameters or torque, but at time $t = 10$ secs, the diagnosis module detects the fault and updates the control parameters based on the offline maps corresponding to the estimated fault strength. At $t = 15$ sec, no further degradation in engine health (constant clogging) does not alarm the diagnosis module further and the system steadily maintains the same control configuration and system state. Notice that there is no considerable change in fuel efficiency from the locally optimal feedback controller to the globally optimal fuel map generated using BO. This is due to the nature of the fault as erosion or clogging in the fuel injector has a minimal effect on the amount of fuel required to obtain the same setpoint torque. Also, notice that the local feedback controller increases the value of the injection pressure and injection duration for both cylinders 3 and 6 (higher for cylinder 3 due to 40% clogging compared to cylinder 6 with 20% clogging), but the global control maps shift the injection pressure even higher while bringing back the duration to the original value of 1.2 ms. This is due to the control input maps being generated for the fixed value of injection duration as explained in the previous section. The small perturbations in cylinders 3 and 6 are visible due to the excitation present as part of the ESC. The area fault factor parameters predicted at each diagnosis time point of 5, 10, and 15 secs are $A_{ff} = [1.02, 1.02, 1.02, 1.02, 1.02, 1.02]$, $A_{ff} = [1.02, 1.02, 0.61, 1.02, 1.02, 0.82]$, and $A_{ff} = [1.02, 1.02, 0.62, 1.02, 1.02, 0.81]$, respectively.

5.2 Intake Manifold Leak

This subsection shows the results for the case of intake manifold leak with the leak diameter of $d = 12\text{mm}$ introduced at time $t = 8$ second. The simulation results were generated for the setpoint torque of 300 Nm with a value of 1200 RPM and the diagnosis module is run every 5 secs. Figure 12(a) shows the torque and fuel flow rate and Fig. 12(b) shows the control injection parameters required to maintain the desired setpoint torque in the presence of a leak. The engine reaches the steady state in 2 secs, and then the torque drops due to the leak at time $t = 8$ second. When the torque value deviates from the setpoint torque, the online feedback control adjusts the control parameters to obtain the desired torque. Notice that at time $t = 10$ secs, the diagnosis module detects the leak in the intake manifold and updates the control parameters based on the offline maps corresponding to the estimated fault strength of diameter $d = 12\text{mm}$. Notice that there is small chattering observed due to added perturbations as a part of a local PI + ESC controller, but at $t = 10$ second, the control values are replaced with the globally optimal control injection parameters generated for a particular fault strength. This results in a slightly lower fuel consumption to obtain the same desired torque. Also, notice that the local feedback controller increases the value

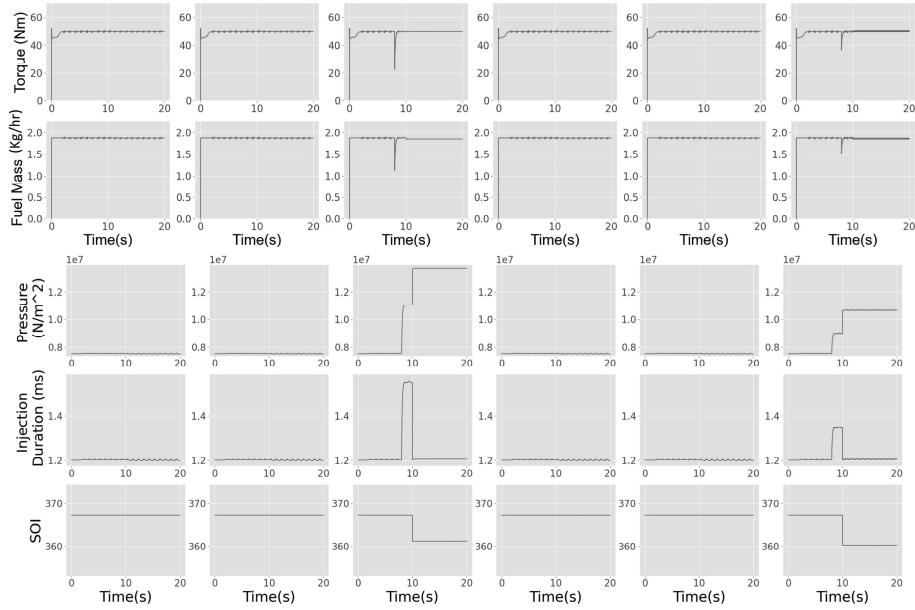


Fig. 11. The torque, fuel mass flow rate, and control input parameters (pressure, duration, and SOI) obtained for the six cylinders in the case of fuel injector clogging.

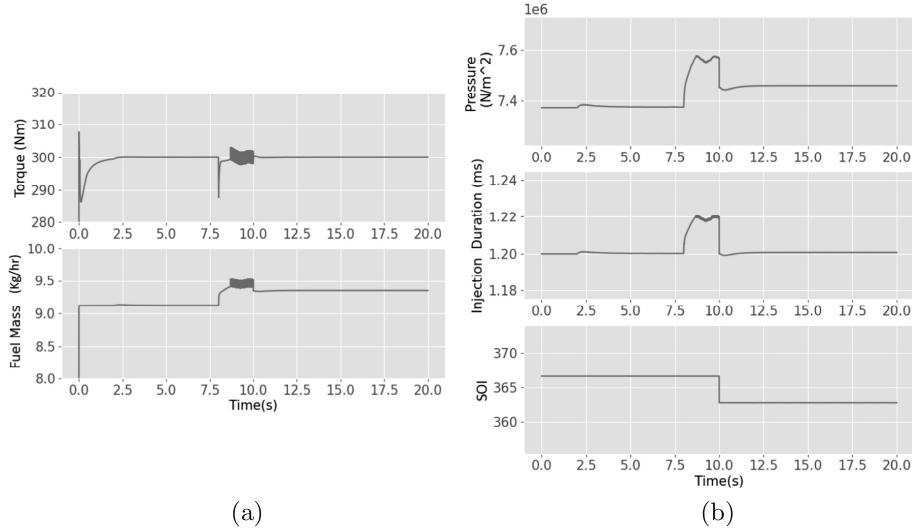


Fig. 12. (a) The total torque and fuel mass flow rate. (b) The control input parameters (pressure, duration, and SOI) for the case of intake manifold leak.

of the injection pressure and injection duration with no change in the value of SOI, but the global control map shifts the injection pressure and injection duration to a lower value and thus resulting in a more fuel-efficient result. The

fault magnitude detected at each diagnosis time point of 5, 10, and 15 secs are $d = 0\text{mm}$, $d = 11.8\text{mm}$, and $d = 11.6\text{mm}$, respectively.

6 Conclusion

The paper proposed a complete framework for fault-resilient autonomous diesel engine systems. We developed a hybrid machine learning architecture for fault classification, fault severity prediction, and a hierarchical control architecture for optimal fault mitigation. The hybrid diagnosis architecture consisted of a denoising autoencoder and twin neural network model for fault classification and prediction of fault severity for fuel injector area clogging/erosion, and intake manifold leaks. The denoising autoencoder helped improve the prediction accuracy of the hybrid diagnosis module by removing the noise from the engine data and allowed for a more general and computationally efficient diagnosis module by fixing the input latent dimension to the hybrid diagnosis module. The hybrid diagnosis module showed overall good prediction accuracy for both kinds of faults modeled in the system. The control calibration maps were generated offline using BO and provided the optimal control inputs to maintain the desired torque for the nominal and predicted value of fault severity. The control module further used an online PI and ES controller to compensate for modeling errors and inaccurate engine fault predictions. The PI controller controls the injection pressure and duration and maintains the desired torque, while the ES controller finds the optimal SOI to maximize the torque. We considered several scenarios for testing the complete framework with fuel injector clogging and erosion for different cylinders and intake manifold leaks of different sizes. The framework was able to detect and mitigate the faults to obtain the desired torque with the minimum possible fuel consumption.

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

References

1. Eck, C., Sidorow, A., Konigorski, U., Isermann, R.: Fault detection for common rail diesel engines with low and high pressure exhaust gas recirculation. Technical Report, SAE Technical Paper (2011)
2. Guoqiang, Z., Baozhu, J., Feng, X., Huaiyu, W.: Intelligent fault diagnosis of marine diesel engine based on deep belief network. Chin. J. Ship Res. **15**(3), 136–142 (2020)
3. Hellstrom, E., Lee, D., Jiang, L., Stefanopoulou, A.G., Yilmaz, H.: On-board calibration of spark timing by extremum seeking for flex-fuel engines. IEEE Trans. control Syst. Technol. **6**(21), 2273–2279 (2013)
4. Heywood, J.B.: Internal Combustion Engine Fundamentals. McGraw-Hill Education, New York (2018)
5. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L.: Physics-informed machine learning. Nat. Rev. Phys. **3**(6), 422–440 (2021)

6. Kitazono, S., Sugihira, S., Ohmori, H.: Starting speed control of SI engine based on extremum seeking control. IFAC Proc. Volumes **41**(2), 1036–1041 (2008)
7. Krstić, M., Wang, H.H.: Stability of extremum seeking feedback for general non-linear dynamic systems. *Automatica* **36**(4), 595–601 (2000)
8. Nohra, C., Noura, H., Younes, R.: A linear approach with μ -analysis control adaptation for a complete-model diesel-engine diagnosis. In: 2009 Chinese Control and Decision Conference, pp. 5415–5420. IEEE (2009)
9. Nyberg, M., Perkovic, A., Nielsen, L.: Model based diagnosis of leaks in the air-intake system of an SI-engine. SAE SPEC PUBL, SAE, WARRENDALE, PA, (USA), Feb 1998, vol. 1357, pp. 25–31 (1998)
10. Popovic, D., Jankovic, M., Magner, S., Teel, A.R.: Extremum seeking methods for optimization of variable cam timing engine operation. *IEEE Trans. Control Syst. Technol.* **14**(3), 398–407 (2006)
11. Rahimi, M., Pourramezan, M.R., Rohani, A.: Modeling and classifying the in-operando effects of wear and metal contaminations of lubricating oil on diesel engine: a machine learning approach. *Expert Syst. Appl.* **203**, 117494 (2022)
12. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: a review of bayesian optimization. *Proc. IEEE* **104**(1), 148–175 (2015)
13. Stefanopoulou, A.G., Kolmanovsky, I., Freudenberg, J.S.: Control of variable geometry turbocharged diesel engines for reduced emissions. *IEEE Trans. Control Syst. Technol.* **8**(4), 733–745 (2000)
14. Tan, Q., Divekar, P.S., Tan, Y., Chen, X., Zheng, M.: Pressure sensor data-driven optimization of combustion phase in a diesel engine. *IEEE/ASME Trans. Mech.* **25**(2), 694–704 (2020)
15. Vachtsevanos, G., Lee, B., Oh, S., Balchanos, M.: Resilient design and operation of cyber physical systems with emphasis on unmanned autonomous systems. *J. Intell. Robot. Syst.* **91**(1), 59–83 (2018)
16. Wahlström, J., Eriksson, L.: Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics. *Proc. Inst. Mech. Eng. Part D: J. Autom. Eng.* **225**(7), 960–986 (2011)
17. Yu, X., Zhu, L., Wang, Y., Filev, D., Yao, X.: Internal combustion engine calibration using optimization algorithms. *Appl. Energy* **305**, 117894 (2022)