energies

Article

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

Turki Alsuwian [1], Muhammad Tayyeb [2], Arslan Ahmed Amin [2,*], Muhammad Bilal Qadir [3], Saleh Almasabi [1] and Mohammed Jalalah [1,4]

1 Department of Electrical Engineering, College of Engineering, Najran University, Najran 11001, Saudi Arabia
2 Department of Electrical Engineering, FAST National University of Computer and Emerging Sciences, Chiniot Faisalabad Campus, Chiniot 35400, Pakistan
3 School of Engineering & Technology, National Textile University, Faisalabad 37610, Pakistan
4 Promising Centre for Sensors and Electronic Devices (PCSED), Advanced Materials and Nano-Research Centre, Najran University, Najran 11001, Saudi Arabia
* Correspondence: arslan.amin@nu.edu.pk

Abstract: Fault-tolerant control systems (FTCS) are used in safety and critical applications to improve reliability and availability for sustained operation in fault situations. These systems may be used in process facilities to reduce significant production losses caused by irregular and unplanned equipment tripping. Internal combustion (IC) engines are widely used in the process sector, and efficient air–fuel ratio (AFR) regulation in the fuel system of these engines is critical for increasing engine efficiency, conserving fuel energy, and protecting the environment. In this paper, a hybrid fault-tolerant control system has been proposed, being a combination of two parts which are known as an active fault-tolerant control system and a passive fault-tolerant control system. The active part has been designed by using the genetic algorithm-based fault detection and isolation unit. This genetic algorithm provides estimated values to an engine control unit in case of a fault in any sensor. The passive system is designed by using the higher-order sliding mode control with an extra fuel actuator in the fuel supply line. The performance of the system was tested experimentally in MATLAB/Simulink environment. Based on the simulation results, the designed system can sustain the AFR despite sensor failures. A new method of managing the AFR of an IC engine has been demonstrated in this study, and it is highly capable, robust, reliable, and highly effective. A comparison with the existing works found in the literature also proves its superior performance. By inserting the fault in each sensor, it was clearly observed that proposed HFTCS was much better than the existing model as it was more fault-tolerant due to its ability to work in both online and offline modes. It also provided an exact value of 14.6 of AFR without any degradation.

Keywords: genetic algorithm; higher-order sliding mode control; fault detection and isolation unit; hybrid fault-tolerant control; robust control

check for updates

Citation: Alsuwian, T.; Tayyeb, M.; Amin, A.A.; Qadir, M.B.; Almasabi, S.; Jalalah, M. 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* **2022**, *15*, 5666. https://doi.org/10.3390/en15155666

Academic Editors: Jan Koscielny, Andrzej Ordys and Michal Syfert

Received: 30 June 2022
Accepted: 1 August 2022
Published: 4 August 2022

1. Introduction

1.1. Fault-Tolerant Control

Faults in an automatic system can cause the unexpected shutdown of the system. These faults can create huge damage to the system's technical components, employees, or the environment [1]. Reliability is a desired attribute in every control system, as well as a need in some critical systems that demand great precision. Although the failure-free operation is the objective, it is impossible to ensure that a system will be completely free of defects. Fault tolerance methods must be implemented to ensure dependability despite the existence of faults [2]. Fault-tolerant control (FTC) is a term that refers to a group of new approaches that have been created to improve plant availability while lowering the risk of safety issues. Its goal is to keep minor flaws from becoming major problems. To achieve this purpose, FTC combines numerous disciplines, including online fault detection, automated condition assessment, and the computation of corrective measures when a fault is discovered. A fault in any system is determined as a variation in the structure or the parameters of the system from its normal operational condition [3]. In a control system, there are three types of faults which are sensor faults, plant faults, and actuator faults, as shown in Figure 1. The list of abbreviations utilized in this paper is shown in Table 1. When there is no fault in the system, it is said to be in normal working condition [4]. The classification of these three types of faults is provided below.

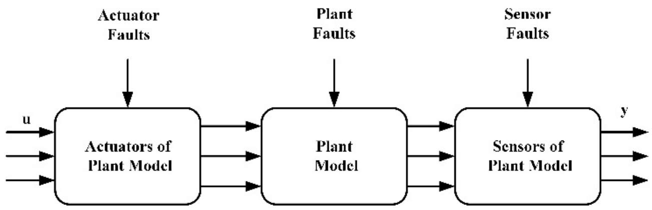

Figure 1. Classification of Faults [1].

Table 1. List of Abbreviations.

Abbreviation	Explanation
FTC	Fault-Tolerant Control
FTCS	Fault-Tolerant Control System
GA	Genetic Algorithm
PFTCS	Passive Fault-Tolerant Control System
HOSMC	Higher-Order Sliding Mode Control
IC	Internal Combustion
HFTCS	Hybrid Fault-Tolerant Control System
MAP	Manifold Absolute Pressure
PID	Proportional, Integral and Derivative
FDI	Fault Detection and Isolation
AFR	Air–Fuel Ratio
FIU	Fault Injection Unit
EGO	Exhaust Gas Oxygen
SI	Spark Ignition
ECU	Engine Control Unit
AFTCS	Active Fault-Tolerant Control System
MSE	Mean Square Error
LT	Lookup Table

Sensor Faults: These faults do not affect the properties of the plant, but there are errors in the readings of sensors when sensor faults occur, and these can negatively affect operation.

Actuator Faults: The properties of the plant are not disturbed in the case of these faults, but the effect of the controller on the system is disturbed.

Plant Faults: In the case of these faults, the input and output properties of the system change.

To fight against all these faults, fault-tolerant control is a well-known approach in control systems [5]. For ensuring fault tolerance, a large number of control techniques are proposed for different control systems. There are two major branches of fault tolerant control systems which are known as robust control and adaptive control. The robust control is also known as passive fault-tolerant control (PFTCS) and adaptive control is also known as active fault-tolerant control (AFTCS) [6–8]. The parameters of the robust control are fixed, but the parameters of adaptive control change according to condition. The response of robust control is fast as compared to the adaptive control, but it does not consider the faults during the running stage. On the other hand, the computational cost of adaptive control is very high as compared to robust control. A simple architecture of Fault-Tolerant Control

System (FTCS) is displayed in Figure 2. A huge quantity of research has been published on different kinds of FTCS approaches. A review of FTCS with detailed applications and advancements has been carried out by Arslan Amin in [9]. This paper provides a detailed review of FTCS with detailed advantages, applications and limitations. A comprehensive comparison between active and passive approaches of FTCS is provided by Jin Jiang in [10]. In this paper, the differences and similarities between active and passive approaches from a theoretical and practical point of view are discussed.

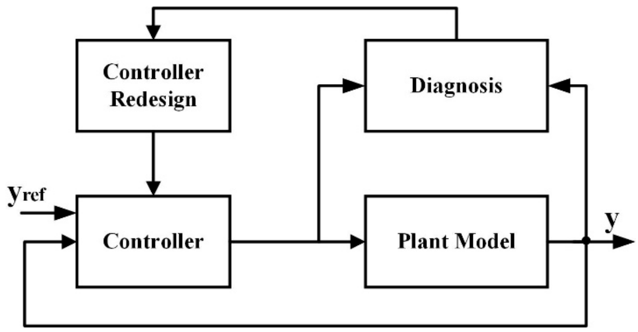

Figure 2. Architecture of FTCS [4].

FTCS for dynamic systems is proposed by Noura, as detailed in [11]. He applied the FTCS on a winding machine by introducing a novel fault estimation and compensation-based method. Here, in the first step, the fault was detected and isolated, then the effect of this fault on the system was reduced. In [12], Odgaard proposed an FTCS for detecting and resolving the faults of wind turbines. A neural network-based FTCS was used with a fault alarm by Shen, as detailed in [13]. In this research, an adaptive FTCS was developed, and the time delay between fault accommodation and fault occurrence was reduced. Niemann suggested an H-infinity-based FTCS for a double inverted pendulum in [14], and his work was validated and verified in a lab environment. M. Tayyeb created a highly redundant FTCS for an aircraft's elevator control system, as documented in [15]. The pitch of the plan was managed by using a mix of Proportional, Integral, and Derivative (PID) control in this study. In [16], Bateman developed fault diagnosis and FTCS for unmanned air vehicles. He used fault detection and diagnosis-based FTCS for a nonlinear model of aircraft. Many other papers on FTCS for different control systems can be seen in the literature [17–19]. Two major classes of FTCS are active FTCS and passive FTCS, which are described in the next subsections.

1.2. Active Fault-Tolerant Control

Adaptive control is known as AFTC, and in this control, the controller works in an online configuration. This controller uses a fault detection and isolation (FDI) unit for detecting the faults and then isolating them. According to the nature and magnitude of the fault, the controller is reconfigured to deal with the fault properly. A general diagram of active FTCS is illustrated in Figure 3. A representation of the FDI unit is provided in Figure 4. The FDI unit in the controller determines the residual which is the difference between the actual sensor measurement and the estimated value from the observer.

If the error lies within the defined limits, then there is no fault declared by the system, and the system works smoothly in normal conditions, but if the error goes out of bounds, then the FDI unit indicates a faulty condition and activates the reconfiguration mechanism to reconfigure the controller. A regression-based FDI unit for controlling the air–fuel ratio (AFR) of an IC engine was proposed in [20]. Similarly, a neural network-based AFTCS with the help of an FDI unit was proposed by M. H. Shahbaz for the AFR of the IC engine, as detailed in [21].

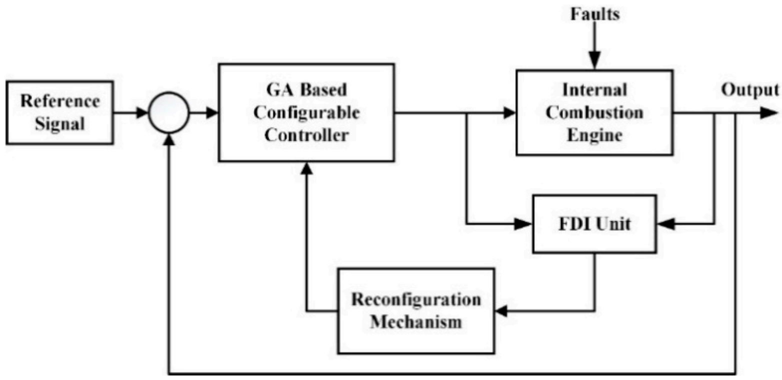

Figure 3. Block diagram of AFTCS [21].

In [22], Chen developed an active FTCS for the hydraulic pitch system of a wind turbine by using the linear parameter varying FTCS. In this research, a Genetic Algorithm (GA)-based AFTCS is utilized in the designing of the FDI unit.

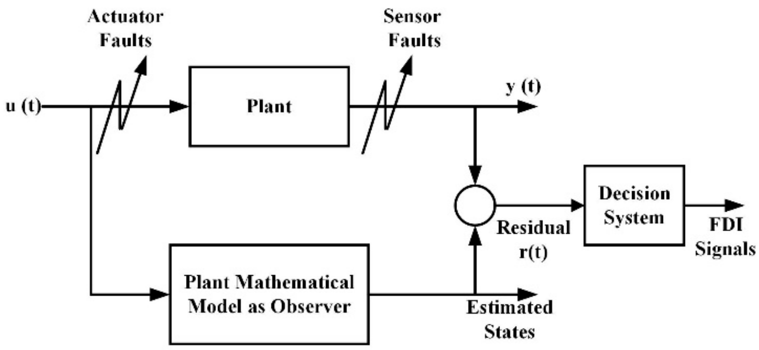

Figure 4. Block Diagram of FDI Unit [22].

The natural selection process, in which the fittest individuals are chosen for breeding to produce the next generation of offspring, is referred to as genetic algorithm. The concept of the genetic algorithm was provided by Charles Darwin. This algorithm works based on natural selection, and it works the same way as biological genes. Selection, crossover, mutation, evaluation, and replacement are the main parts of genetic algorithm. There are many research articles found in the literature for the genetic algorithm-based FTCS [23–25]. Omar proposed a way of utilizing the genetic algorithm to solve workshop scheduling in [26]. He used this approach to construct a random beginning population that included the results of various well-known priority principles, such as minimum and maximum processing time.

Snehal Kamalapur proposes a genetic algorithm-based efficient CPU scheduling strategy in [27], stating that CPU scheduling has a great impact on the operating system's speed and throughput. A genetic algorithm-based back propagation neural network was proposed by Ding, as detailed in [28]. This genetic algorithm was developed to improve the convergence rate and to enhance the training of back propagation. When both genetic algorithm and back propagation are merged, then the training of the neural network improves. A study on genetic algorithm and its applications by Halduri is represented in [29]. In his research, all the components of genetic algorithm were discussed in detail with all advantages and limitations. A simple block diagram of genetic algorithm with its parts is represented in Figure 5.

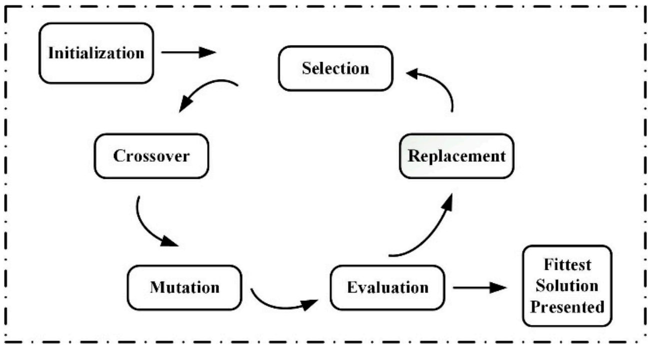

Figure 5. Block diagram of GA [27].

1.3. Passive Fault-Tolerant Control

The robust control is also named passive FTCS in the FTC literature, and in this control, the controller is designed the offline configuration. It handles only those faults which were considered during the design stage. A block diagram representation of passive FTCS is provided in Figure 6. Passive FTCS is a robust control as it is free from any kind of estimation, and that is why this control is mostly used for instant action in most critical applications, in order to avoid huge losses.

When there is a fault in the system, the passive FTCS masks the faulty value to run the operation of the system. This controller compares the setpoint with the feedback, and according to this feedback signal, it masks the error if the error goes out of bounds. A passive FTCS-based robust control for faults of the IC engine was developed in [30]. In this research, for the design of a robust control, a high feedback gain was utilized. In [31], Kordestani constructed an adaptive passive FTCS for a steam turbine, and in this passive FTCS principal component, the analysis-based neural network was utilized. He used a multi-layer neural network in the design of this passive FTCS.

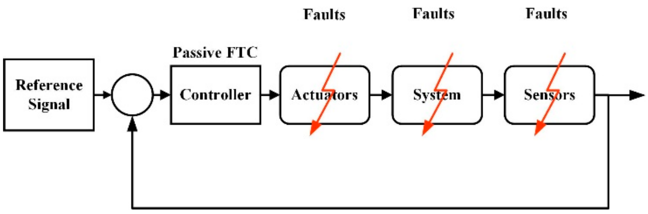

Figure 6. Block diagram of PFTCS [31].

For the energy management of an electric vehicle, Oubellil developed an H-infinity passive FTCS-based control, as detailed in [32]. In this electric drive system, two converters in parallel configuration were utilized, and fault tolerance was applied to these converters. Many control algorithms are known as robust controls, for example, proportional integral derivative (PID) control, ON-OFF control, high gain control, etc. but higher-order sliding mode control (HOSMC) is a very powerful robust passive FTCS, and due to this reason, this control is employed in this research. Sliding surface design and control law design are two main tasks in sliding mode control. A representation of the sliding phase and reaching phase is provided in Figure 7.

In the sliding mode control, the control law is established in such a way that the sliding variable must approach the origin in a finite time. When the sliding surface in SMC is achieved, the system will not be affected by any disturbance. A good amount of literature is found on higher-order sliding mode control-based FTCS, as can be seen in [33–35]. A detailed review of a sliding mode control with all of its types has been carried out by Umar Riaz, as detailed in [36]. In his review article, different types of sliding mode control along with their advantages, applications, and limitations in the area of FTCS were described. As described in [37], Asif created a sliding mode control for an electro-pneumatic-based actuator. He also studied the resilience of produced control laws under the influence of external disturbances in this study, and he discovered that control laws remain resilient in the face of any uncertainty. In [38], Boukadida developed an FTCS-based control for controlling a helicopter. In this research, he utilized a higher-order sliding mode control with a linear quadratic regulator for the pitch and yaw movement of the helicopter.

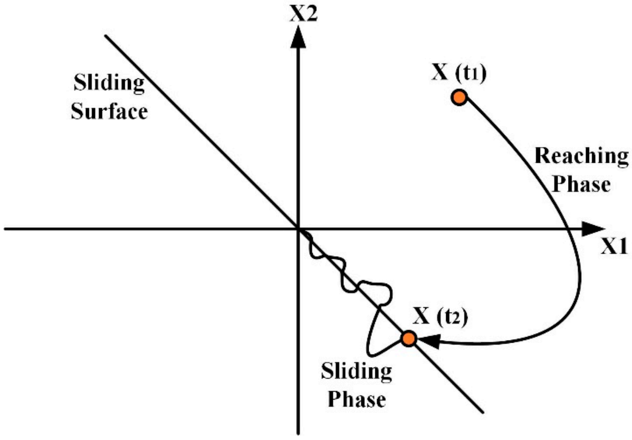

Figure 7. Concept of sliding phase and reaching phase [37].

1.4. Hybrid Fault-Tolerant Control

Although both AFTCS and PFTCS have many features, on the other hand, they also have some limitations, such as PFTCS works only for limited faults, and AFTCS has a slow response. Therefore, to overcome these deficiencies, a hybrid FTCS (HFTCS) is proposed, which is a hybrid of both AFTCS and PFTCS, as described in [39]. A block diagram of HFTCS is shown in Figure 8. This HFTCS works both in online and offline modes, and it considers all the faults in the running stage and designing stage. Due to this HFTCS, the system becomes more reliable, and its efficiency also increases, which is why it is highly recommended in critical applications.

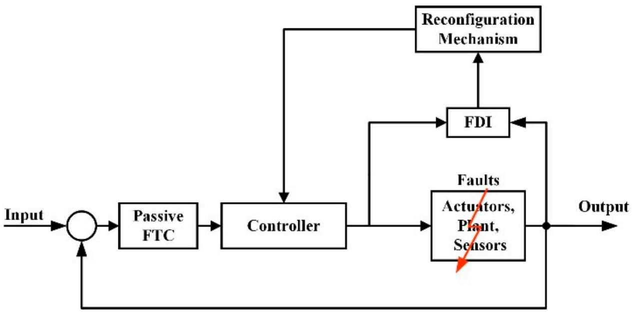

Figure 8. Block diagram of HFTCS [40].

Different research articles have been proposed on hybrid FTCS, as detailed in [40–42]. In his research [43], Jiang proposed a hybrid FTCS for partial failures of the actuator by utilizing the fault detection and diagnosis scheme. The whole hybrid FTCS was applied on an aircraft and simulated for the proper functioning of actuators. Xiang observed that the proposed hybrid FTCS lowers the magnitude of the fault and then sends it to the FDI unit for the reconfiguration of the controller. A hybrid FTCS for actuator and sensor faults of an active suspension system was described by Pang in [44]. In this paper, FTCS was applied for the proper working of the sensors and actuators in case of road disturbances.

1.5. Air–Fuel Ratio Control

IC engines are frequently employed as the primary mover in process applications. These engines turn the chemical energy of the fuel into mechanical rotational energy, which is then used to power the compressors and alternators. Spark ignition (SI) and compression ignition (CI) engines are the two types. In spark ignition engines, spark plugs are employed in the combustion process, but in compression ignition engines, no spark plugs are used, and combustion is exclusively accomplished by compression. In our research, we looked into SI IC engines. The proper mixing of air and fuel in the combustion process in a certain ratio is known as the air–fuel ratio (AFR), and it is critical for greater engine performance, reduced fuel consumption, and low harmful emissions for environmental protection. The proper AFR value is given by Equation (1).

$$AFR = \frac{m_{air}}{m_{fuel}} \tag{1}$$

where m_{air} represents the mass of the air, and m_{fuel} represents the mass of the fuel. The chemical equation of the combustion process for the gasoline fuel is given below:

$$25O_2 + 2C_8H_{18} \rightarrow 16CO_2 + 18H_2O + Energy \tag{2}$$

The AFR according to this equation is termed a stoichiometric ratio, and its value comes out to be 14.6:1 for gasoline fuel [45,46]. In an IC engine, the main purpose of the AFR controller is to maintain the value of AFR for efficient working of the engine [47–49].

The block diagram of the AFR system of the SI IC engine is shown in Figure 9. The air blows through the filter to remove dust particles and other undesired items, and the throttle valve controls how fast it moves. Similarly, the fuel is also filtered, and the pressure is changed to fit the engine's requirements. The air and fuel are then combined and sent to the engine cylinders. To maintain the AFR value at a set point of 14.6, the AFR controller is installed in the engine control unit (ECU). The combustion is disrupted and the engine stops working when a fault in the sensors or actuators arises. The FTCS is used to avoid the engine from shutting down as a result of this, allowing it to work reliably and constantly, even in the most difficult situations.

The AFR of the IC engines is controlled by four sensors [50,51] which are listed below.

Exhaust Gas Oxygen (EGO) Sensor: This sensor is used to monitor the fuel supply and checks the oxygen level in the exhaust stream to guarantee proper combustion.

Speed Sensor: This sensor measures the engine crankshaft speed.

The Manifold Absolute Pressure (MAP) Sensor: This sensor gives the controller an accurate reading of suction air pressure.

Throttle Sensor: The controller receives information from this sensor, indicating the position of the air throttle.

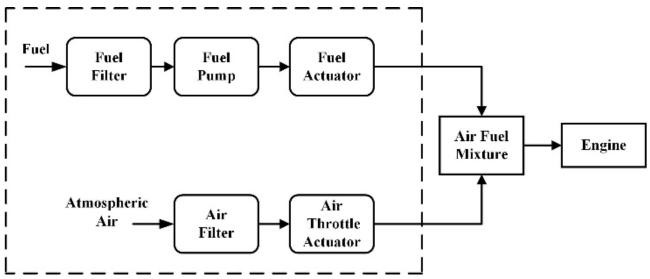

Figure 9. Block diagram of AFR system for SI IC Engine [52].

The assembly of these four sensors for the IC engine is given in Figure 10.

Our contribution is the development of a hybrid fault-tolerant control for an internal combustion gasoline engine's air–fuel ratio control that preserves engine stability in the case of single sensor faults, without any degradation in the value of the air–fuel ratio. In this paper, the design of a novel genetic algorithm and higher-order sliding mode control-based hybrid fault-tolerant control for non-linear sensors in an IC engine's air–fuel ratio control system has been implemented. Proposed system is developed by the combination of genetic algorithm-based active FTCS and higher-order sliding mode control-based passive FTCS.

This model for the IC engine was created using a MATLAB/Simulink environment, and the results demonstrate that the recommended model is stable in both normal and faulty conditions. As compared to the previous work, the proposed model for the IC engine is more fault tolerant, stable, and efficient, and it is more dependent on the MAP sensor's nonlinear behavior. When compared to the previous techniques, the suggested model provides better and more accurate results. Due to its fast response and the highest degree of robustness against faults, this method is very useful in the automobile industry and in many critical applications.

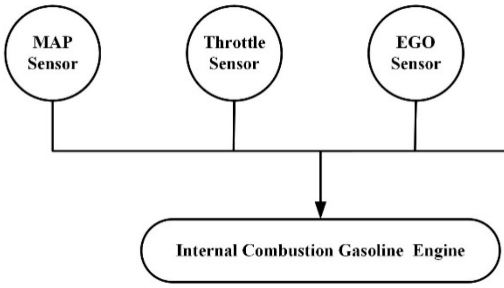

Figure 10. Main sensors for AFR control of IC Engine [53].

Further contents of the paper are structured as follows: The research methodology is proposed in Section 2. Section 3 contains the results and a discussion and a comparison with existing work are provided in Section 4. Finally, the conclusion is provided in the last Section 5.

2. Research Methodology

The suggested HFTCS for controlling the AFR of the IC engine is developed in the existing MATLAB/Simulink model of the IC engine. The complete internal working of this reference model with the help of all internal diagrams is provided in [52,53]. The AFR system of the gasoline engine in this model is based on Crossley and Cook's results [54] and was extensively confirmed using dynamometer test data. The Mean Value Engine Model [55] was utilized to create the mathematical equations for the model building. Furthermore, it provides realistic AFR as seen in real-world gasoline engines.

As detailed earlier, both AFTCS and PFTCS combine to make a unique HFTCS. Similarly, in this research, GA-based AFTCS and HOSMC-based PFTCS are combined to develop an HFTCS for controlling the AFR of the IC engine. Due to the design speed of the available MATLAB engine model, the engine speed for this investigation is set at 300 r/min. As a result, in the event of a speed sensor failure, the FDI unit sends a value of 300 to the controller. Because the engines in the process plant operate at a constant speed most of the time, and the intended FDI gives the controller 300 r/min of speed if the speed sensor fails, we used constant speed in this research. Load fluctuations and their influence on performance are not investigated, since the article is focused on creating an HFTCS system. Using the provided MATLAB model lookup tables (LTs), the data for the MAP sensor and the throttle sensor at 300 r/min is obtained. The GA method is used to establish nonlinear interactions between the MAP sensor and the throttle sensor. These nonlinear connections are used by the FDI unit to generate the estimated value of faulty sensors. The FDI unit develops an estimated value based on GA observations and feeds it to the ECU if the throttle and MAP sensors are malfunctioning. The design of AFTCS, PFTCS, and HFTCS is illustrated in the coming sections.

2.1. AFR System Modeling

The air intake dynamics are explained as follows using mass conservation theory and the ideal air gas principle [56].

$$\dot{P_{in}} = \frac{RT_{in}}{V_{in}}(\dot{m_{th}} - \dot{m_{Cyt}}) + P_{in}\frac{\dot{T_{in}}}{T_{in}} \tag{3}$$

$$\dot{P_{in}} = \psi(\varnothing_{th},\ P_{in},\ T_{in},\ N_e) \tag{4}$$

Here, "T_{in}" stands for the input temperature, and "P_{in}" stands for the manifold pressure. The mass flow into the valve is "m_{th}", mass flow into the cylinders is "m_{Cyt}", gas constant is represented by R, the temperature of the intake air is considered to be zero. The piston speed is N_e, and the opening position of throttle is $\varnothing_{th}$. The intake temperature's time derivative is assumed to be zero. As a result, differential Equation (3) is as follows.

$$\dot{P_{in}} = \dot{K_{in}}(\dot{m_{th}} - \dot{m_{Cyt}}) \tag{5}$$

$$\dot{K_{in}} = \frac{RT_{in}}{V_{in}} \tag{6}$$

Through the valve, air mass flow is given as

$$\dot{m_{th}} = C_d\frac{P_{id}}{\sqrt{RT_{id}}}S_{es}\ (\varnothing_{th})g(P_r) \tag{7}$$

Here, "P_{id}" is overhead loading pressure, "P_r" is throttle opening area and "C_d" is discharge coefficient. The product of "$C_dS_{es}\ (\varnothing_{th})$" is known as affective opening area of throttle and it is given as

$$S_{ett}(\varnothing_{th}) = C_dS_{es}(\varnothing_{th}) = \sigma_1\{1 - \cos(\sigma_2\varnothing_{th} + \sigma_3)\} + \sigma_4 \tag{8}$$

Here, $g\ (Pr)$ is known as non-linear concept in the sense that

$$g(P_r) = \begin{cases} \sqrt{\frac{2\gamma}{\gamma-1}(P_r)^{\frac{1}{\gamma}}}\sqrt{\left(1 - P_r^{\frac{\gamma-1}{\gamma}}\right)} & if\ P_r > \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \\ \sqrt{\gamma}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} & if\ P_r \le \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \end{cases} \tag{9}$$

Here, "γ" is specific heat ratio and its value is 1.4. The explanation of fuel dynamics is given below

$$\begin{cases} \ddot{m}_{ff}(t) = \frac{1}{\tau_f}\left(-\dot{m}_{ff}(t) + x\right) \\ \dot{m}_{fv} = (1-x)\dot{m}_{fi}(t) \\ \dot{m}_f(t) = \dot{m}_{fv}(t) + \dot{m}_{ff}(t) \end{cases} \tag{10}$$

Here, "τ_f" is fuel vapour process, "$\dot{m}_{fi}(t)$" is injection of fuel flow, "$\dot{m}_f(t)$" is flow of fuel into cylinders, "$\dot{m}_{fv}$" is vapour fuel flow and "$\ddot{m}_{ff}(t)$" is liquid mass fuel flow. The solution which is chosen for this research is provided below.

$$\tau_f(N_e) = \sigma_5 N_e^{-\sigma_6} \tag{11}$$

$$x(N_e) = \sigma_7 + \sigma_8 N_e \tag{12}$$

where σ_5, σ_6, σ_7, σ_8 are constant parameters. The injector model is given by a linear relationship between the mass fuel flows from the injectors.

Additionally, AFR can now be obtained as

$$\lambda_{cyl} = \frac{\dot{m}_{cyl\ (t)}}{\lambda_s \dot{m}_{f(t)}} \tag{13}$$

The sensor model is provided below:

$$\dot{\lambda}(t) = -\frac{1}{\tau_\lambda}\ \lambda\ (t) + \frac{1}{\tau_\lambda}\lambda_{cyl}(t - \tau\ (N_e(t))) \tag{14}$$

Here, "$\tau_\lambda = 0.1$ s" is constant time delay. Below, the given equation shows speed of engine "*Ne* (*t*)" with a time delay "τ"

$$\tau\ (N_e\ (t)) = \frac{60}{N_e(t)}\left(1 + \frac{1}{n_{cyl}}\right) \tag{15}$$

2.2. AFTCS Design

To design the proposed AFTCS, the GA-based FDI unit is utilized. This FDI unit contains an estimation block and a reconfiguration block. This FDI unit works as an observer to estimate the values of the MAP sensor and throttle sensor. The flowchart working of AFTCS is shown in Figure 11.

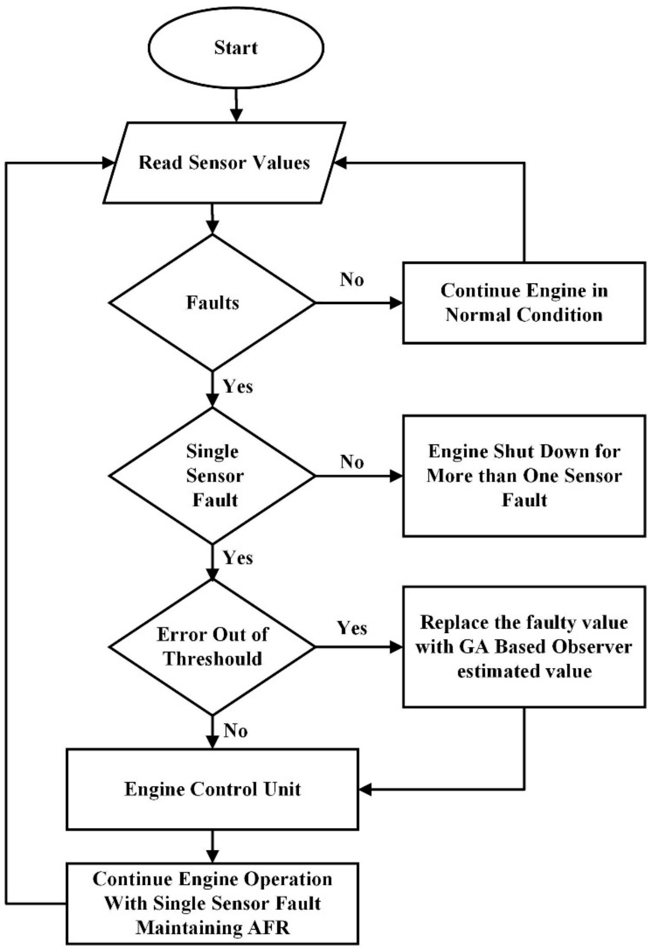

Figure 11. Flowchart of proposed AFTCS part.

Throttle and MAP sensor readings for 300 r/min were derived by using the available MATLAB model's lookup table (LT). The GA is used to derive nonlinear throttle and MAP relationships. The FDI unit utilizes these nonlinear correlations to estimate the faulty sensors' value. The estimated value of the faulty sensor is generated using the other healthy sensor value assuming only one sensor can go faulty at a time. The flowchart working of this GA-based FDI unit is provided in Figure 12. The state-space representation for the AFTCS observer is provided below:

$$\dot{x} = Ax + Bu \tag{16}$$

$$y = Cx + Du \tag{17}$$

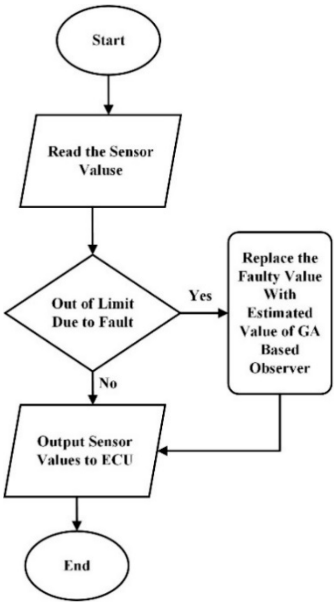

Figure 12. Flowchart of the GA-based FDI Unit.

Let $\overline{x}$ be the estimated state produced by the observer, then

$$\dot{\overline{x}} = A\overline{x} + Bu \tag{18}$$

$$\overline{y} = C\overline{x} + Du \tag{19}$$

where 'x', 'u', and 'y' symbolize the states, inputs, and outputs, respectively. The symbols A, B, C, and D represent the system matrices.

$$(\dot{\overline{x}} - \dot{x}) = A\,(\overline{x} - x) \tag{20}$$

$$(\overline{y} - y) = \mathrm{C}\,(\overline{x} - x) \tag{21}$$

$$\dot{\overline{x}} = A\overline{x} + Bu + L\,(\overline{y} - y) \tag{22}$$

Here, L is known as feedback gain

$$(\dot{\overline{x}} - \dot{x}) = A\,(\overline{x} - x) + L\,(\overline{y} - y) \tag{23}$$

$$(\overline{y} - y) \;=\; \mathrm{C}\,(\overline{x} - x) \tag{24}$$

$$(\dot{\overline{x}} - \dot{x}) = (A + LC)\,(\overline{x} - x) \tag{25}$$

$$\dot{e}_x = (A + LC)e_x \tag{26}$$

In the above equation "e_x" is termed as residual

$$(\overline{y} - y) = \;\mathrm{C}e_x \tag{27}$$

When there is no fault in the system, it works in the normal condition, but when a single sensor in the system fails, the GA-based observer provides the estimated value of this sensor to the ECU. When the MAP sensor fails, then its estimation is provided with the help of a throttle sensor, and when the throttle sensor fails, then its estimation is provided with the help of the MAP sensor. So, the system works smoothly in both faulty and normal conditions.

The fitness function is required in order to apply GA to the estimate of defective sensor readings. Because the results from GA's initial settings were unsatisfactory, the parameters of GA were modified to provide better ones. The number of generations was fixed at 50, and the range of values was set to [0.001, 2] in order to determine the best minimum. MATLAB is used to estimate the unknown coefficients' optimal values when the fitness function is reduced and the optimum values of the unknown coefficients are obtained. Despite the GA's excellent performance, the following drawbacks must be considered. The GA utilized in this work can only handle fitness functions of modest complexity. GA efficiency will be extremely poor in terms of computing time and resources if the fitness function is too complicated or too basic. We need to plot the actual and estimated results repeatedly, since the GA solution is superior to the other solutions. Our range of solution values must be precisely adjusted for multivariable fitness functions, since the GA occasionally yields local minima. Even though there are certain drawbacks to using GA, we chose it over other methods because of the following reasons: GA, in contrast to other optimization algorithms such as particle swarm optimization (PSO), is a discrete method that is used to issues with discrete data sets. GA is more suited to our situation, since we only have a limited number of MAP and throttle sensor readings. In contrast to the global optimal solution, GA focuses on the local maxima/minima and so provides the solutions more rapidly.

Steps for GA

Coding and Decoding: Many coding and decoding schemes are used in genetic algorithm. In this research, the binary coding scheme is selected for ease and simplicity. Each part of the parameter vector is encoded as a string of length, consisting of zeros and ones for the desired resolution, based on the binary coding process.

Fitness: In order to determine the feasibility of a chromosome, fitness is a criterion. According to the theory of survival of the fittest, a chromosome with a greater fitness score is more likely to lead to one or more siblings in the next generation. The success criteria are linked to the fitness function by the use of GA. For calculating the fitness of each chromosome, the roulette wheel selection is performed here. For calculating the fitness, the following relation was used.

$$P_i = F_i / \sum_{j=1}^{n} F_j \tag{28}$$

Here, "F_i" is the fitness for string,"F_j" is the fitness of each individual, "P_i" is probability of the string, and "n" is total number of individuals.

After this, the number of new evaluations is found as

$$\text{Number of new evaluations} = \text{E} = n \times P_i \tag{29}$$

Reproduction: Reproduction is the fundamental GA operator. It is based on the principle of the survival of the fittest individual. In each generation, depending on the possibility of replication, chromosome of existing population is replicated into next generation.

Crossover: Reproduction guides the GA's search for the right individuals. Via probabilistic decision in the mating pool, crossover procedure is carried out to swap the information between any two chromosomes. A single-point crossover scheme is utilized in this research. The expected number of generations after crossover are calculated as:

$$n_g = \frac{E}{n[1 - (1 - p_c)(1 - p_a)]} \tag{30}$$

Here, "p_c" is probability of crossover, "p_a" is probability of adjacent mutation, and "n_g" is expected number of generations.

Mutation: The gene pool starts to become more and more homogeneous in GA when one better gene starts to dominate after many generations and contributes to premature non-optimal solution convergence. The third genetic operator mutation is implemented with sufficient probability in GA to solve this undesirable convergence.

2.3. PFTCS Design

The estimation blocks of the original model have been eliminated for the design of PFTCs, and HOSMC is applied to this model. A fuel actuator is added to the fuel supply line. The HOSMC-based controller with a set point of 14.6 is used to control the AFR using this fuel actuator. The fuel actuator receives the command from this controller. The closed-loop system for robust HOSMC-based PFTCS is shown in Figure 13. The flowchart working of the proposed PFTCS is provided in Figure 14. From this figure, it can be seen that an extra fuel-adjusting actuator is introduced in the fuel supply line, and the output of this fuel-adjusting actuator is provided to the engine control unit.

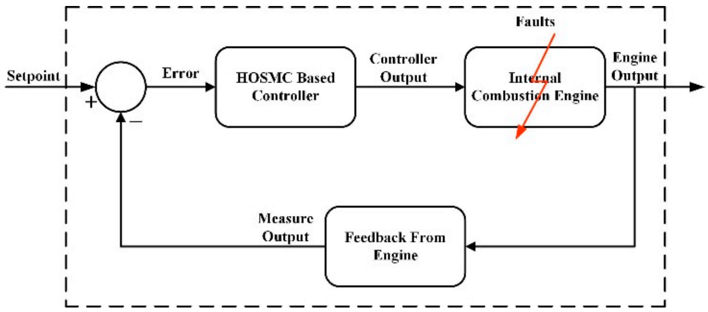

Figure 13. HOSMC-based closed-loop PFTCS.

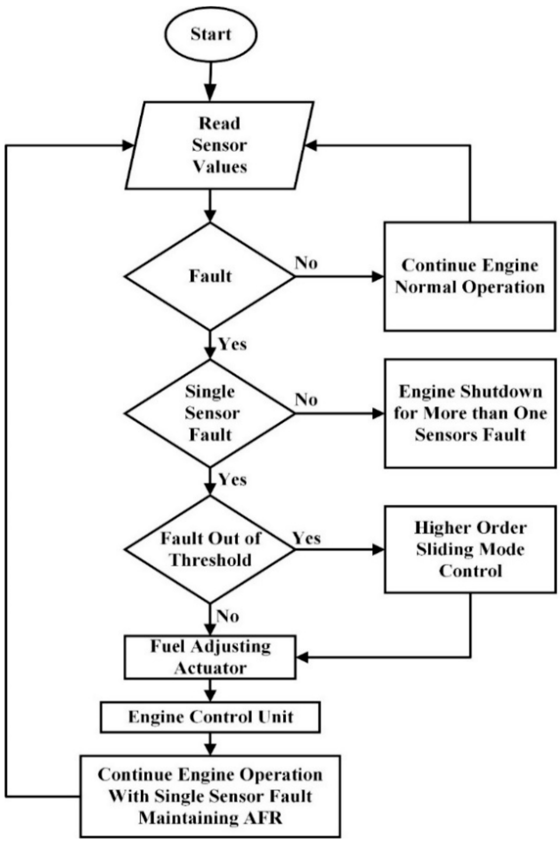

Figure 14. Flowchart of proposed PFTCS part.

In the design of the HOSMC controller, the nonlinear sliding variable is provided as

$$\sigma = \sigma\ (x_1,\ x_2) \tag{31}$$

Here, "σ" is a nonlinear sliding variable. The above equation can be written as

$$\sigma = x_2 + c\ |x_1|^{1/2}\ sign\ (x_1) \tag{32}$$

where $c > 0$.

In the above equation, "c" is a positive gain, and "*sign*" is called signum function. The next step is to design the sliding manifold which drives the controller output to zero in presence of a bounded disturbance. This sliding manifold is calculated with the help of the following equation.

$$x_2 + c\ |x_1|^{1/2}\ sign\ (x_1) = 0 \tag{33}$$

The control law to derive the sliding variable to zero is

$$u = -\rho\ sign\ (\sigma) \tag{34}$$

Here, $\rho =$ positive gain.

By putting the value of "σ" in the above equation, we get the following equation

$$u = -\rho\ sign\ (x_2 + c\ |x_1|^{1/2}\ sign\ (x_1)) \tag{35}$$

2.4. HFTCS Design

By combining both the architectures of AFTCS and PFTCS, the HFTCS is developed. In this proposed HFTCS, both the FDI unit and the robust control are used. The flowchart working of the proposed HFTCS is provided in Figure 15. This proposed HFTCS provides analytical redundancy with the help of GA-based FDI unit, and it also provides robustness with the help of robust HOSMC. When a sensor fails, the estimated value of this sensor is provided with the help of the FDI unit of this HFTCS. If there remains any kind of degradation in the amount of fuel, the HOSMC fulfills this degradation with the help of a fuel-adjusting actuator. With the help of both GA and HOSMC, the engine will work perfectly in both normal and faulty conditions. However, when the two sensors fail at the same time, then the engine will be shut down. As the AFR value degrades from 14.6 to 11.7 due to a fault in a sensor, so to avoid this degradation, HOSMC is implanted to maintain the value of AFR at a level of 14.6.

The assumptions of this study are as follows: (1) The engine operates at a constant speed without considering load variations as per a practical process plant scenario. (2) Within a faulty situation, the sensors yield zero value and the actuator fully closes. The study's limitations include that it only covers full-type failure of sensors and actuators and does not look at partially faulty components that produce partial outputs. Another limitation of this study is that it includes single sensor failures instead of multi-sensor failures.

The major challenge in this research work was the degradation in the value of AFR during AFTCS design. To address this challenge, the PFTCS was recommend because this robust controller eliminated the degradation in fuel with the help of a fuel actuator in fuel supply line. In this research, only single sensor faults were considered instead of two or more sensor faults.

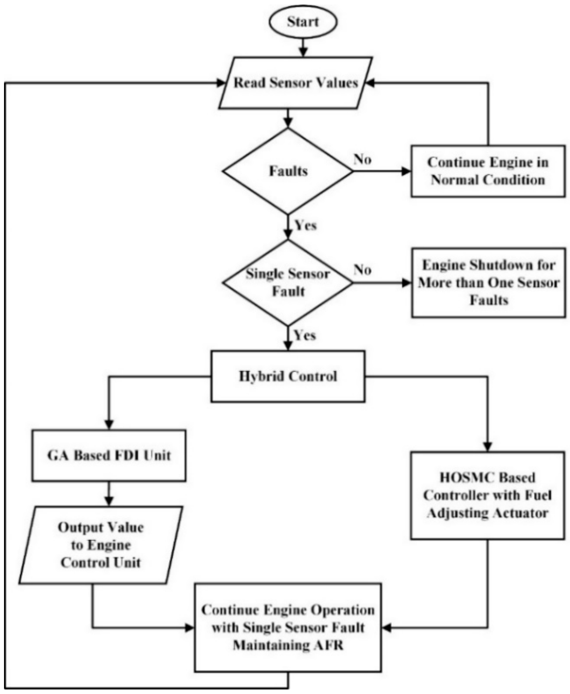

Figure 15. Flowchart for proposed HFTCS.

3. Results and Discussion

The simulation model for the AFTCS part of the proposed HFTCS is provided in Figure 16. GA is being used for the observer in the FDI unit to determine MAP and throttle estimated values. The MAP, EGO, throttle, and speed sensors are all included in this model. The fault switches of these sensors are also included in this model to introduce a fault in any sensor. The GA-based FDI unit is further subdivided into two units which are known as the estimation unit and reconfiguration unit. The block diagram of the estimation unit of FDI is given in Figure 17. The estimation unit of FDI estimates the values of throttle and MAP sensors in case of a fault in these sensors from other healthy sensor. The block diagram for the reconfiguration unit of the FDI unit is shown in Figure 18. The reconfiguration block determines the residual and supplies the output to the ECU for normal/faulty conditions.

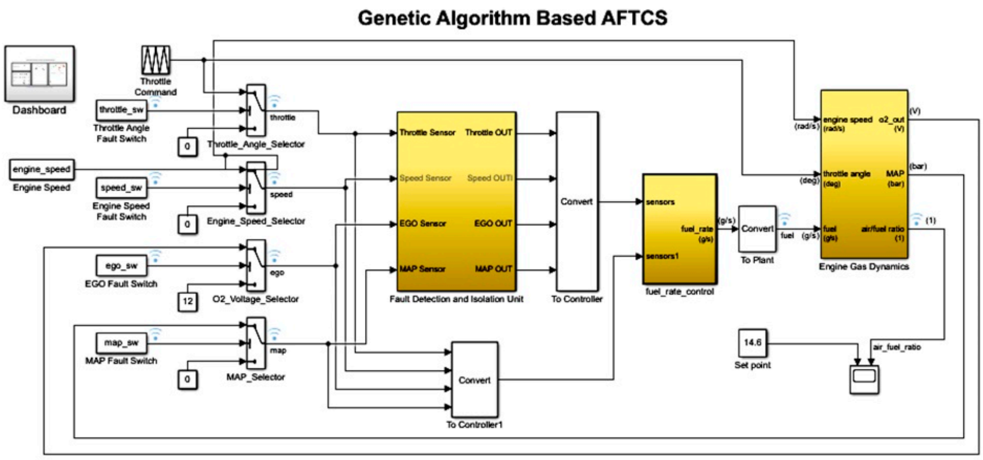

Figure 16. MATLAB Model for AFTCS Part.

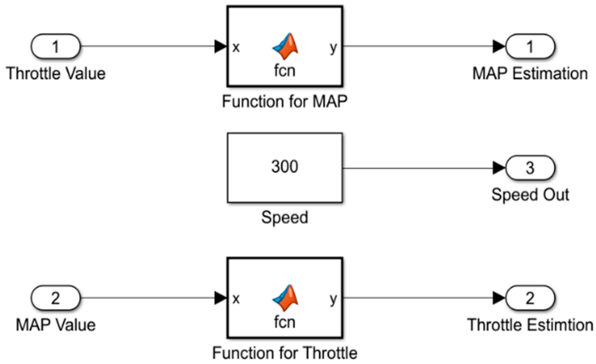

Figure 17. Block diagram of estimation unit.

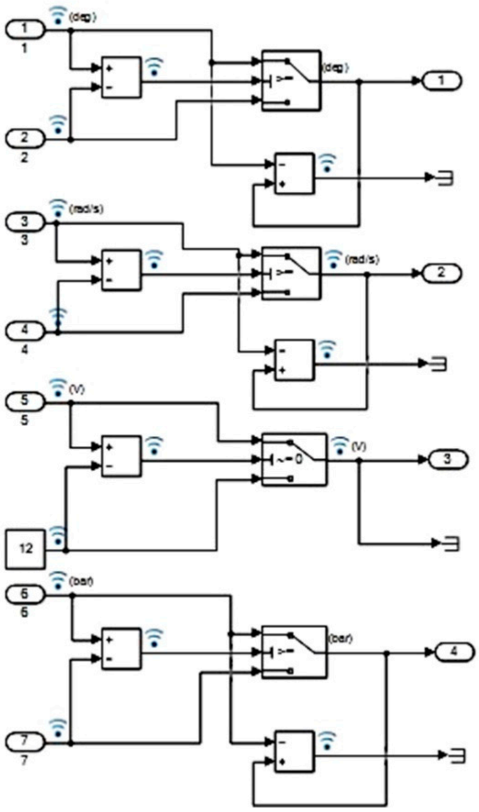

Figure 18. Block diagram of reconfiguration unit.

A fault is injected into the engine with the help of the fault injection unit (FIU). All the fault switches of all four sensors of the IC engine are linked with this FIU. The system detects a fault when its residual with the observer value crosses the threshold limit, and then the faulty value is updated with the GA-based observer's estimated value. By making one sensor faulty at a time, the AFTCS model is simulated and the effect of sensor fault is observed after t = 5 s due to the internal warm-up delay of the engine. The performance of this AFTCS for healthy conditions and for fault in any single sensor is given in Figure 19. The system ultimately attained stability, and this delay is tolerable given the steady output, which is the main purpose of AFTCS. This delay will be reduced by utilizing the PFTCS implementation or raising the gain of the passive controller, although this will increase chattering as well.

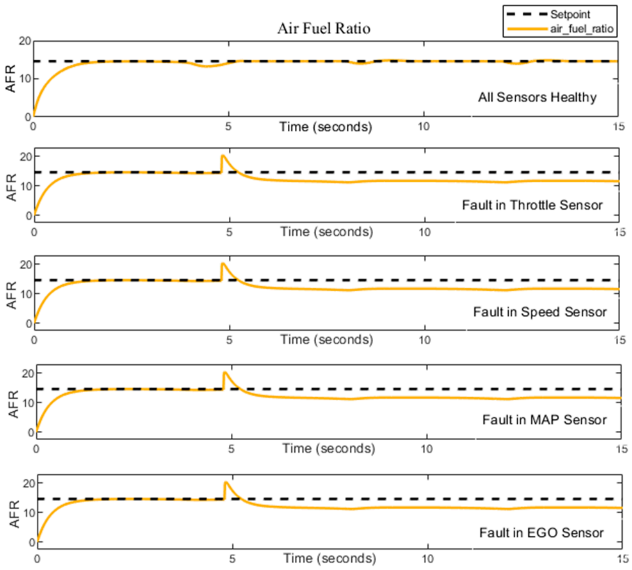

Figure 19. Performance of AFTCS.

The simulation model for the PFTCS part is represented in Figure 20. In this PFTCS model, an extra fuel actuator is added to the fuel supply line system for controlling purposes through HOSMC. The output value of the HOSMC controller and fuel output is provided to this fuel actuator and the output of this actuator controls the amount of fuel being delivered to the engine block.

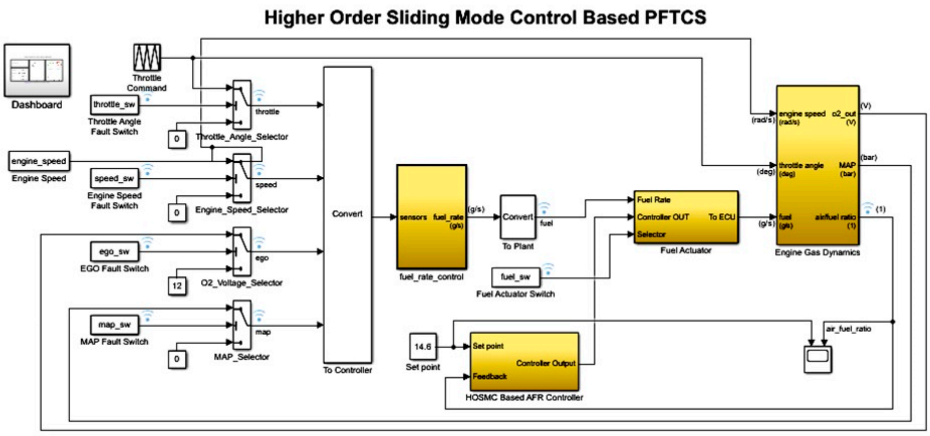

Figure 20. MATLAB Model for PFTCS Part.

The fuel actuator's architecture is given in Figure 21. The fault switch of this actuator is connected to the dashboard of the system. When a fault is added to this actuator through control port, it cuts off the flow of fuel for the engine by the selector switch operation.

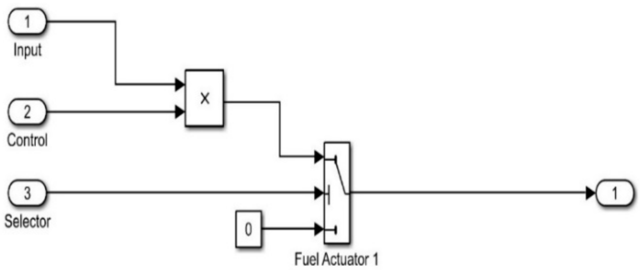

Figure 21. The architecture of the fuel actuator.

The performance of the PFTCS in the case of a single sensor fault is given in Figure 22. This figure shows that the response of the system remains stable and does not fluctuate from 14.6 in faulty conditions.

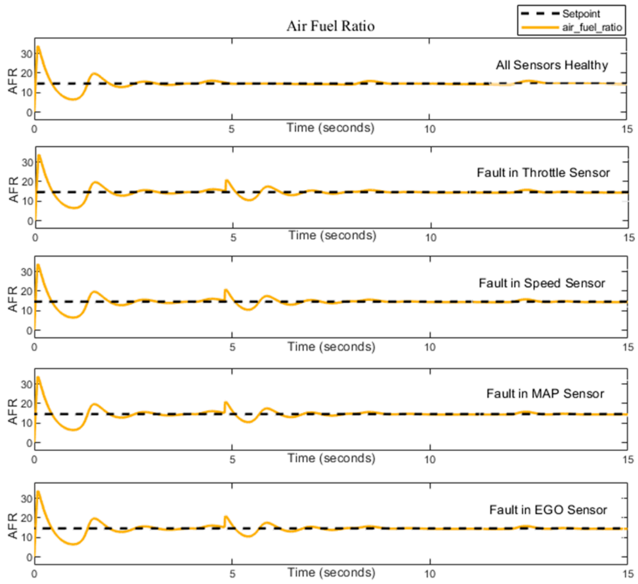

Figure 22. Performance of PFTCS.

By combining both controllers with the original model, the MATLAB diagram of the proposed HFTCS is shown in Figure 23. Both the FDI unit and AFR robust controller are present in this proposed HFTCS model.

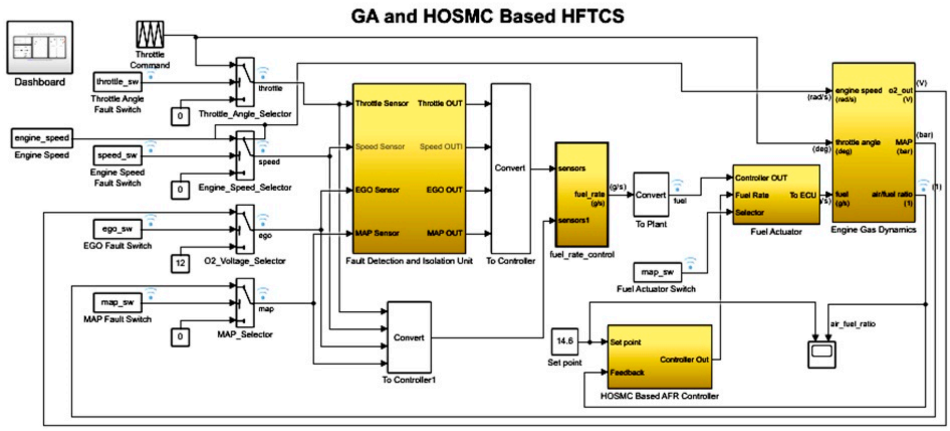

Figure 23. MATLAB model of proposed HFTCS.

Figure 24 displays the performance of the overall HFTCS when faults are added one by one in each of the four sensors. By keeping the AFR value at 14.6 in this figure, the suggested HFTCS is fault tolerant. The model's use of an estimated value instead of the EGO sensor value causes the spikes in AFR in case of an EGO sensor fault. The suggested HFTCS provides several benefits as compared to the current model. First, it incorporates both AFTCS and PFTCS features within their respective structures, while the previous model did not. Second, AFR drops from 14.6 to 11.7 when the fault occurs, but AFR does not degrade in the proposed HFTCS.

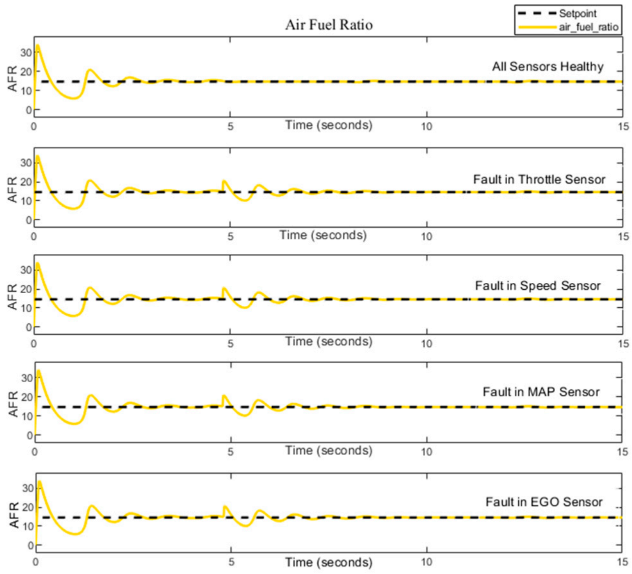

Figure 24. Overall performance of HFTCS.

In the output graphs in Figures 19, 22 and 24, the delay at the time of fault injection is tolerable. This is not a very large period delay, and it can also be minimized by adjusting the delay time in the simulation. Finally, the suggested model is found to be more robust and reliable. In SI IC gasoline engines, the suggested HFTCS provides an optimal and efficient solution for AFR control.

4. Comparison with Existing Works

This section compares the proposed design approach to prior work in the literature. Several studies have been undertaken for the AFTC design, such as the use of linear regression, lookup tables, Kalman filter, ANN, and Fuzzy Logic [7,21,22,40,48]. All of these methods have benefits, but they also have disadvantages, such as the high computational cost of lookup tables, Fuzzy Logic, and ANN. Linear regression and KF were found for the limited linear range of the MAP sensor. However, these had limitations in terms of degradation of AFR with the active part alone. The HFTCS was found with a PI controller, but it is not robust enough to handle highly nonlinear faulty behavior, thus instigating the need for advanced robust control. We created an HFTCS with specific non-linear controllers called GA and HOSMC. Previous work has not utilized this combination together for HFTCS design for the IC Engine's AFR system. HOSMC would comprise the passive half of the proposed system, reacting promptly to faults while minimizing chattering at the same time, and GA will maximize post-fault performance with active compensation.

The performance of GA estimates is evaluated in Tables 2 and 3 for some data points. The tables below show the predicted data for the MAP and throttle sensors, as well as the mean square error (MSE) error, which decreased significantly by using GA. Table 4 provides an overall comparison of the proposed HFTCS approach with the previous techniques.

Single controllers have fewer benefits than hybrid controllers, as these hybrid controllers have high processing speed, operational stability, and soft computing approaches. The suggested model for an IC engine was created by combining an FDI unit with a strong feedback controller, resulting in a structure that is simpler to use in both online and offline fault handling. The model's implementation is relatively simple in comparison to other methods. Due to all these reasons, the suggested HFTCS for the IC engine is created in this research.

Table 2. MAP estimation using GA-based observer.

LT Values for MAP	GA Values for MAP	Error	MSE
0.091	0.90	0.01	5.0×10^{-5}
0.113	0.11	0.003	4.5×10^{-6}
0.190	0.18	0.010	5.0×10^{-5}
0.329	0.32	0.009	8.0×10^{-5}
0.545	0.54	0.005	1.25×10^{-5}
0.745	0.74	0.005	1.25×10^{-5}
0.857	0.85	0.007	2.45×10^{-5}
0.915	0.90	0.005	1.25×10^{-5}
0.946	0.93	0.016	1.28×10^{-4}
0.964	0.95	0.014	9.8×10^{-5}
0.975	0.97	0.005	1.25×10^{-5}
0.985	0.98	0.005	1.25×10^{-5}
0.994	0.98	0.014	9.8×10^{-5}
0.997	0.98	0.017	1.45×10^{-4}
0.998	0.99	0.008	3.2×10^{-5}
0.999	0.99	0.009	4.05×10^{-5}
0.999	0.99	0.009	4.05×10^{-5}

Table 3. Throttle estimation using GA-based observer.

LT Values for Throttle	GA Values for Throttle	Error	MSE
0	0.5	−0.5	1.25×10^{-1}
1.979	1.95	0.029	4.21×10^{-4}
4.686	4.6	0.086	3.7×10^{-3}
6.258	6.25	0.008	3.2×10^{-5}
7.471	7.46	0.011	6.05×10^{-5}
8.482	8.45	0.032	1.02×10^{-3}
9.357	9.36	−0.003	4.5×10^{-6}
10.163	10.11	0.053	1.40×10^{-3}
10.824	10.78	0.044	9.6×10^{-4}
11.452	11.4	0.052	1.35×10^{-3}
12.061	12	0.061	1.86×10^{-3}
12.70	12.69	0.01	5.0×10^{-5}
13.402	13.40	0.002	2.0×10^{-6}
14.187	14.17	0.017	1.44×10^{-4}
15.107	15.10	0.007	2.45×10^{-5}
16.24	16.25	−0.01	5.0×10^{-5}
17.754	17.73	0.024	2.8×10^{-4}

Table 4. Comparison between proposed and previous techniques [7,21,22,40,48].

Name of Controller	Chattering Reduction	Degree of Robustness	Response against Noise
Proposed HFTCS	Eliminates chattering effect	Insensitive with the highest degree of robustness	Best for noisy systems
AFTCS based on ANN and Fuzzy Logic	Does not eliminate chattering	Unknown duration for handling faults	High Misfiring Observed
HFTCS based on Kalman Filter	Does not eliminate chattering	Does not provide robustness	High Misfiring Observed
AFTCS based on Linear Regression	Does not eliminate chattering	Not a robust technique	High Misfiring Observed

5. Conclusions

HFTCS was proposed in this study for AFR control of IC engines utilizing GA and HOSMC. This novel HFTCS was created in the MATLAB/Simulink platform to assure the reliability of an IC engine. The FDI unit was established with the help of GA to provide estimated values to the controller during the fault in the sensor. In faulty conditions, an HOSMC-based AFR controller was introduced to manage AFR by regulating the throttle actuator in the line of fuel supply. The main focus in this research was to achieve the desired value of air–fuel ratio, which was 14.6. The active controller provided this desired value with some degradation in it. Therefore, to fulfill this degradation, a robust passive controller was combined with this active controller, and this combination provided the exact desired value of air–fuel ratio with better results. To verify the robustness of the proposed HFTCS, faults were introduced one by one in the sensors. The simulation results demonstrated that the proposed HFTCS can tolerate faults for both normal and faulty sensor conditions while maintaining the AFR. Under faulty conditions, it effectively maintained AFR at 14.6 without any degradation. According to this research, the suggested HFTCS made the AFR control system extremely robust in terms of preventing output loss. Single controllers have fewer benefits than hybrid controllers. Moreover, hybrid controllers have high processing speed, operational stability, and soft computing approaches. The suggested model for an IC engine was created by combining an FDI unit with a strong feedback controller, resulting in a structure that is simpler to use in both online and offline fault handling. The model's implementation was relatively simple in comparison to other methods. Due to all these reasons, the suggested HFTCS for the IC engine was created in this research.

For future study, golden section optimization, hardware-in-the-loop testing, and fuzzy-based SMC can be utilized. A combination of generational or steady-state-based GA can be selected in the future due to their fast response. Other techniques such as terminal, conventional and advanced SMC can be considered to produce the best results in the future.

Author Contributions: Conceptualization, A.A.A.; Formal analysis, M.T. and M.B.Q.; Funding acquisition, T.A.; Investigation, M.J.; Methodology, M.T. and A.A.A.; Project administration, A.A.A. and M.J.; Resources, M.B.Q.; Software, M.B.Q.; Supervision, T.A.; Validation, S.A.; Visualization, S.A.; Writing—original draft, M.T.; Writing—review & editing, A.A.A., S.A. and M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program grant code (NU/RG/SERC/11/5).

Data Availability Statement: All necessary data used in the research is available in the open-access Mathworks model [52].

Acknowledgments: The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Groups Funding program grant code (NU/RG/SERC/11/5).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, H.; Yang, Y.; Li, L.; Ding, S.X. Performance-based fault detection and fault-tolerant control for nonlinear systems with T–S fuzzy implementation. *IEEE Trans. Cybern.* **2019**, *51*, 801–814. [CrossRef] [PubMed]
2. Amin, A.A.; Mahmood-Ul-Hasan, K. Advanced fault tolerant air-fuel ratio control of internal combustion gas engine for sensor and actuator faults. *IEEE Access* **2019**, *7*, 17634–17643. [CrossRef]
3. Sivagami, V.M.; Easwarakumar, K.S. An improved dynamic fault tolerant management algorithm during VM migration in cloud data center. *Future Gener. Comput. Syst.* **2019**, *98*, 35–43. [CrossRef]
4. Shen, H.; Dai, M.; Luo, Y.; Cao, J.; Chadli, M. Fault-Tolerant Fuzzy Control for Semi-Markov Jump Nonlinear Systems Subject to Incomplete SMK and Actuator Failures. *IEEE Trans. Fuzzy Syst.* **2021**, *29*, 3043–3053. [CrossRef]
5. Ahmed, S.; Amin, A.A.; Wajid, Z.; Ahmad, F. Reliable speed control of a permanent magnet DC motor using fault-tolerant H-bridge. *Adv. Mech. Eng.* **2020**, *12*, 1687814020970311. [CrossRef]
6. Saied, M.; Lussier, B.; Fantoni, I.; Shraim, H.; Francis, C. Active versus passive fault-tolerant control of a redundant multirotor UAV. *Aeronaut. J.* **2020**, *124*, 385–408. [CrossRef]

7. Riaz, U.; Amin, A.A.; Tayyeb, M. Design of active fault-tolerant control system for Air-fuel ratio control of internal combustion engines using fuzzy logic controller. *Sci. Prog.* **2022**, *105*, 00368504221094723. [CrossRef]
8. Guezmil, A.; Berriri, H.; Pusca, R.; Sakly, A.; Romary, R.; Mimouni, M.F. Experimental investigation of passive fault tolerant control for induction machine using sliding mode approach. *Asian J. Control* **2019**, *21*, 520–532. [CrossRef]
9. Amin, A.A.; Hasan, K.M. A review of fault tolerant control systems: Advancements and applications. *Measurement* **2019**, *143*, 58–68. [CrossRef]
10. Jiang, J.; Yu, X. Fault-tolerant control systems: A comparative study between active and passive approaches. *Annu. Rev. Control* **2012**, *36*, 60–72. [CrossRef]
11. Noura, H.; Sauter, D.; Hamelin, F.; Theilliol, D. Fault-tolerant control in dynamic systems: Application to a winding machine. *IEEE Control Syst. Mag.* **2000**, *20*, 33–49.
12. Odgaard, P.F.; Stoustrup, J.; Kinnaert, M. Fault-tolerant control of wind turbines: A benchmark model. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 1168–1182. [CrossRef]
13. Shen, Q.; Jiang, B.; Shi, P.; Lim, C.-C. Novel neural networks-based fault tolerant control scheme with fault alarm. *IEEE Trans. Cybern.* **2014**, *44*, 2190–2201. [CrossRef]
14. Niemann, H.; Stoustrup, J. Passive fault tolerant control of a double inverted pendulum—A case study. *Control Eng. Pract.* **2005**, *13*, 1047–1059. [CrossRef]
15. Tayyeb, M.; Riaz, U.; Amin, A.A.; Saleem, O.; Arslan, M.; Shahbaz, M.H. Design of highly redundant fault tolerant control for aircraft elevator system. *J. Appl. Eng. Sci.* **2021**, *19*, 37–47. [CrossRef]
16. Bateman, F.; Noura, H.; Ouladsine, M. Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV. *IEEE Trans. Aerosp. Electron. Syst.* **2011**, *47*, 2119–2137. [CrossRef]
17. Liu, Y.; Stettenbenz, M.; Bazzi, A.M. Smooth fault-tolerant control of induction motor drives with sensor failures. *IEEE Trans. Power Electron.* **2018**, *34*, 3544–3552. [CrossRef]
18. Li, X.; Karimi, H.R.; Wang, Y.; Lu, D.; Guo, S. Robust fault estimation and fault-tolerant control for Markovian jump systems with general uncertain transition rates. *J. Frankl. Inst.* **2018**, *355*, 3508–3540. [CrossRef]
19. Kavikumar, R.; Sakthivel, R.; Kwon, O.M.; Kaviarasan, B. Faulty actuator-based control synthesis for interval type-2 fuzzy systems via memory state feedback approach. *Int. J. Syst. Sci.* **2020**, *51*, 2958–2981. [CrossRef]
20. Amin, A.A.; Mahmood-ul-Hasan, K. Robust active fault-tolerant control for internal combustion gas engine for air–fuel ratio control with statistical regression-based observer model. *Meas. Control* **2019**, *52*, 1179–1194. [CrossRef]
21. Shahbaz, M.H.; Amin, A.A. Design of active fault tolerant control system for air fuel ratio control of internal combustion engines using artificial neural networks. *IEEE Access* **2021**, *9*, 46022–46032. [CrossRef]
22. Chen, L.; Shi, F.; Patton, R. Active FTC for hydraulic pitch system for an off-shore wind turbine. In Proceedings of the 2013 Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France, 9–11 October 2013; pp. 510–515.
23. Zhu, D.; Wang, L.; Hu, Z.; Yang, S.X. A Grasshopper Optimization-based fault-tolerant control algorithm for a human occupied submarine with the multi-thruster system. *Ocean. Eng.* **2021**, *242*, 110101. [CrossRef]
24. Reddy, G.T.; Reddy, M.; Lakshmanna, K.; Rajput, D.S.; Kaluri, R.; Srivastava, G. Hybrid genetic algorithm and a fuzzy logic classifier for heart disease diagnosis. *Evol. Intell.* **2020**, *13*, 185–196. [CrossRef]
25. Zhou, Q.; Liu, M.; Wang, W.; Xu, L.; Ao, T.; Zhang, H. Research on fault tolerant control strategy of multi-degree-of-freedom manipulator with single joint faults. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 5853–5859.
26. Omar, M.; Baharum, A.; Hasan, Y.A. A Job-shop Scheduling Problem (JSSP) Using Genetic Algorithm (GA). In Proceedings of the 2nd IM TG T Regional Conference on Mathematics, Statistics and Applications Universiti Sains Malaysia, Penang, Malaysia, 13–15 June 2006.
27. Kamalapur, S.; Deshpande, N. Efficient CPU scheduling: A genetic algorithm based approach. In Proceedings of the 2006 International Symposium on Ad Hoc and Ubiquitous Computing, Mangalore, India, 20–23 December 2006; pp. 206–207.
28. Ding, S.; Su, C.; Yu, J. An optimizing BP neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.* **2011**, *36*, 153–162. [CrossRef]
29. Haldurai, L.; Madhubala, T.; Rajalakshmi, R. A study on genetic algorithm and its applications. *Int. J. Comput. Sci. Eng.* **2016**, *4*, 139.
30. Amin, A.A.; Mahmood-ul-Hasan, K. Robust passive fault tolerant control for air fuel ratio control of internal combustion Gasoline engine for sensor and actuator faults. *IETE J. Res.* **2021**, 1–16. [CrossRef]
31. Kordestani, M.; Salahshoor, K.; Safavi, A.A.; Saif, M. An adaptive passive fault tolerant control system for a steam turbine using a PCA based inverse neural network control strategy. In Proceedings of the 2018 World Automation Congress (WAC), Stevenson, WA, USA, 3–6 June 2018; pp. 1–6.
32. Oubellil, R.; Boukhnifer, M. Passive fault tolerant control design of energy management system for electric vehicle. In Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 1–4 June 2014; pp. 1402–1408.
33. Zaihidee, F.M.; Mekhilef, S.; Mubin, M. Application of fractional order sliding mode control for speed control of permanent magnet synchronous motor. *IEEE Access* **2019**, *7*, 101765–101774. [CrossRef]
34. Djouima, M.; Azar, A.T.; Drid, S.; Mehdi, D. Higher order sliding mode control for blood glucose regulation of type 1 diabetic patients. *Int. J. Syst. Dyn. Appl. (IJSDA)* **2018**, *7*, 65–84. [CrossRef]

35. Basin, M.V.; Yu, P.; Shtessel, Y.B. Hypersonic missile adaptive sliding mode control using finite-and fixed-time observers. *IEEE Trans. Ind. Electron.* **2017**, *65*, 930–941. [CrossRef]
36. Riaz, U.; Tayyeb, M.; Amin, A.A. A review of sliding mode control with the perspective of utilization in fault tolerant control. *Recent Adv. Electr. Electron. Eng. (Former. Recent Pat. Electr. Eng.)* **2021**, *14*, 312–324. [CrossRef]
37. Chalanga, A.; Plestan, F. High-order sliding-mode control with predefined convergence time for electropneumatic actuator. *IEEE Trans. Control Syst. Technol.* **2020**, *29*, 910–917. [CrossRef]
38. Boukadida, W.; Benamor, A.; Messaoud, H.; Siarry, P. Multi-objective design of optimal higher order sliding mode control for robust tracking of 2-DoF helicopter system based on metaheuristics. *Aerosp. Sci. Technol.* **2019**, *91*, 442–455. [CrossRef]
39. Amin, A.A.; Mahmood-ul-Hasan, K. Hybrid fault tolerant control for air–fuel ratio control of internal combustion gasoline engine using Kalman filters with advanced redundancy. *Meas. Control* **2019**, *52*, 473–492. [CrossRef]
40. Chen, J.; Zhang, C.; Chen, A.; Xing, X. Fault-tolerant control strategies for T-type three-level inverters considering neutral-point voltage oscillations. *IEEE Trans. Ind. Electron.* **2018**, *66*, 2837–2846. [CrossRef]
41. Hagh, Y.S.; Asl, R.M.; Cocquempot, V. A hybrid robust fault tolerant control based on adaptive joint unscented Kalman filter. *ISA Trans.* **2017**, *66*, 262–274. [CrossRef]
42. Vargas-Martínez, A.; Minchala Avila, L.I.; Zhang, Y.; Garza-Castañón, L.E.; Badihi, H. Hybrid adaptive fault-tolerant control algorithms for voltage and frequency regulation of an islanded microgrid. *Int. Trans. Electr. Energy Syst.* **2015**, *25*, 827–844. [CrossRef]
43. Yu, X.; Jiang, J. Hybrid fault-tolerant flight control system design against partial actuator failures. *IEEE Trans. Control Syst. Technol.* **2011**, *20*, 871–886. [CrossRef]
44. Pang, H.; Liu, X.; Shang, Y.; Yao, R. A hybrid fault-tolerant control for nonlinear active suspension systems subjected to actuator faults and road disturbances. *Complexity* **2020**, *2020*, 1874212. [CrossRef]
45. Subramani, D.A.; Dhinagaran, R.; Prasanth, V.R. Introduction to turbocharging—A perspective on air management system. In *Design and Development of Heavy Duty Diesel Engines*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 85–193.
46. Nekvasil, H.; DiFrancesco, N.J.; Rogers, A.D.; Coraor, A.E.; King, P.L. Vapor-deposited minerals contributed to the Martian surface during magmatic degassing. *J. Geophys. Res. Planets* **2019**, *124*, 1592–1617. [CrossRef]
47. Amin, A.A.; Mahmood-ul-Hasan, K. Unified fault-tolerant control for air-fuel ratio control of internal combustion engines with advanced analytical and hardware redundancies. *J. Electr. Eng. Technol.* **2022**, *17*, 1947–1959. [CrossRef]
48. Na, J.; Chen, A.S.; Huang, Y.; Agarwal, A.; Lewis, A.; Herrmann, G.; Burke, R.; Brace, C. Air–Fuel Ratio Control of Spark Ignition Engines with Unknown System Dynamics Estimator: Theory and Experiments. *IEEE Trans. Control Syst. Technol.* **2019**, *29*, 786–793. [CrossRef]
49. Sui, W.; Hall, C.M. Combustion phasing modeling and control for compression ignition engines with high dilution and boost levels. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2019**, *233*, 1834–1850. [CrossRef]
50. Jafarian, K.; Mobin, M.; Jafari-Marandi, R.; Rabiei, E. Misfire and valve clearance faults detection in the combustion engines based on a multi-sensor vibration signal monitoring. *Measurement* **2018**, *128*, 527–536. [CrossRef]
51. Romani, L.; Bianchini, A.; Vichi, G.; Bellissima, A.; Ferrara, G. Experimental assessment of a methodology for the indirect in-cylinder pressure evaluation in four-stroke internal combustion engines. *Energies* **2018**, *11*, 1982. [CrossRef]
52. Modeling a Fault-Tolerant Fuel Control System—MATLAB & Simulink. Available online: https://www.mathworks.com/help/simulink/slref/modeling-a-fault-tolerant-fuel-control-system.html;jsessionid=6068d2d9cdc5c26e2c47bd6c6df1 (accessed on 29 July 2022).
53. Modeling Engine Timing Using Triggered Subsystems—MATLAB & Simulink. Available online: https://www.mathworks.com/help/simulink/slref/modeling-engine-timing-using-triggered-subsystems.html (accessed on 29 July 2022).
54. Crossley, P.R.; Cook, J.A. A nonlinear engine model for drivetrain system development. In Proceedings of the International Conference on Control 1991, Control'91, Edinburgh, UK, 25–28 March 1991; pp. 921–925.
55. Hendricks, E.; Sorenson, S.C. Mean value modelling of spark ignition engines. *SAE Trans.* **1990**, *99*, 1359–1373.
56. Lauber, J.; Guerra, T.-M.; Dambrine, M. Air-fuel ratio control in a gasoline engine. *Int. J. Syst. Sci.* **2011**, *42*, 277–286. [CrossRef]