Genetic algorithm based active fault-tolerant control system for air fuel ratio control of internal combustion engines

Muhammad Sajid Iqbal and Arslan Ahmed Amin

Abstract
Internal Combustion (IC) engines are commonly used in the process industry, and proper Air-Fuel Ratio (AFR) regulation in their fuel system is essential for improved engine performance, fuel efficiency, and environmental protection. Since faults in the sensors of the AFR system cause the engine to shut down, fault tolerance is essentially required for them. In this paper, an advanced Active Fault-Tolerant Control System (AFTCS) for AFR control of an IC engine is proposed to prevent the engine shutdown. For analytical redundancy, the proposed AFTCS uses the Genetic Algorithm (GA) based observer model in the Fault Detection and Isolation (FDI) unit to generate estimated value for the faulty sensors using the values of other healthy sensors. The proposed framework was simulated in the MATLAB/Simulink environment to verify its effectiveness. In comparison to previous literature works, the results show that the AFR control system has superior fault tolerance output for sensor faults, especially for the MAP sensor, in terms of less oscillatory response. Finally, a comparison of the proposed GA-based AFTCS was carried out with the existing literature works and its superior performance was elaborated.

Keywords
Active fault-tolerant control, AFR control, fault detection and isolation unit, genetic algorithm, IC engine

Date received: 6 October 2021; accepted: 13 June 2022

Introduction
A system fault is characterized as a deviation of the parameter from its actual value. Fault tolerance refers to a system’s ability to continue operating in defective conditions. Any real-world machine can develop faults, which reduce the system’s stability and efficiency.1–3 Fault-tolerant control (FTC) is a technique that can be used to improve the stability of sensitive systems like nuclear power plants and airplanes. The two most common forms of FTC are active and passive.4–6 The Active Fault-Tolerant Control System (AFTCS) comprises a Fault Detection and Isolation (FDI) unit that is responsible for detecting and isolating defective system components. The FDI employs the observer principle, in which a plant parameter is compared to a predefined normal value to provide a residual.7 If the residual is below reasonable limits, the system is said to be free of errors. If the residual exceeds the specified threshold, the FDI unit declares faulty conditions.5 The controller is then reprogrammed to meet the new operational requirements. Output may be harmed by defective parts, but overall system reliability is assured.9,10

An FDI unit can be built using several approaches. For example in Amin and Mahmood-ul-Hasan,2 Kalman Filter (KF) is used and in general, the KF is not an ideal estimator. The measurement and the state transition model should both be linear for a KF to generate accurate estimations but the model used in Amin and Mahmood-ul-Hasan2 is nonlinear hence KF is generating less accurate values. A linear regression-based model is used in Amin and Mahmood-ul-Hasan,11 Linear Regression Algorithm is too conventional, hence when used for the estimation of a nonlinear sensor it is showing poor performance. An Artificial Neural Network (ANN)-based approach is reported in Shahbaz and Amin12 which is showing highly accurate
results but the critical data points on which the error percentage is high are ignored in that study. The model reported in Berredjem and Benidir\textsuperscript{13} is using fuzzy logic to diagnose the bearing faults. The study reports a high degree of accuracy but there is a systematic approach to applying Fuzzy Logic to solve a problem. As a result, several solutions to a single problem emerge, confusing. Furthermore, the control system built in Berredjem and Benidir\textsuperscript{13} must be updated on a regular basis.

In an aero-engine, KFs are used in Yuan et al.\textsuperscript{14} for fault detection and position in the case of sensor and actuator faults coexisting. In Berredjem and Benidir,\textsuperscript{13} FDI employs fuzzy logic to forecast nonlinear processes, while adaptive control is used to correct bias and locate actuator faults. Using simulated sensor values in FDI to estimate flow speeds for actuator faults, a nonlinear adaptive observer for a double pipe heat exchanger is built.\textsuperscript{15} FTC was important in designing a control strategy in quasi and procedure environments in flawed scenarios. FTC is a key component in maintaining system reliability and stability when it comes to actuator and sensor faults.\textsuperscript{16,17}\textsuperscript{18,19} configures multilevel reconfiguration for actuators and sensors using the NASA aircraft LPV system. In a recent configuration of the drone control system, NASA has used FTC.\textsuperscript{17}

To achieve good orbit-attitude tracking performance an improved finite-time sliding mode control approach is proposed in Shi et al.\textsuperscript{18} in which the soft docking procedure is separated into three sequential movements with set duration to reduce the complexity, and the high-precision performance of the generated simplified electromagnetic force/torque models is shown by numerical analysis. Another study related to spacecraft docking can be found in Zhang et al.\textsuperscript{19} in which the essential problems in spacecraft electromagnetic docking control, such as model uncertainties, unknown external disturbances, and inherent strong nonlinearity and coupling, are solved using a nonlinear sliding mode control that combines online model modification, a nonlinear sliding mode controller, and a disturbance observer. In Liu et al.,\textsuperscript{20} Delayed Electromagnetic Docking of Spacecraft in Elliptical Orbits with Active Disturbance Rejection Control is proposed and validated with a high degree of accuracy however the non-fragile control dealing with controller perturbations is not considered in this study. The authors Maki et al.\textsuperscript{21} have suggested a multi-controller solution without using an FDI decision-making unit. In Yang and Yin,\textsuperscript{22} a hybrid solution was proposed for restoring sensor values using simulated sensor values to ensure reliability in the event of sensor failure. To define the affected variables, the proposed FDI scheme in Li et al.\textsuperscript{23} uses the canonical variate analysis (CVA) technique and the exponentially weighted moving average (EWMA) paradigm based on Pearson correlation analysis. Some more nonlinear fault-tolerant control systems are reported in the literature, a proportional-integral observer is used for Switched Fuzzy Stochastic Systems in Han et al.,\textsuperscript{24} and a directed-graph based approach is reported in Han et al.\textsuperscript{25} in which a unique intermediate observer design approach is provided to estimate system states, actuator failures, and sensor faults. Another nonlinear fault estimation approach can be found in Han et al.\textsuperscript{26} in which a dynamic proportional-integral observer is used for nonlinear system. In Han et al.,\textsuperscript{24,26} a P-I controller is used and as stated in Sreekumar and Jiji\textsuperscript{27} the P-I controller has some drawbacks, including a large initial overshoot, sensitivity to controller gains, and a slow reaction to unexpected disturbances, hence an improved control strategy is necessary. In Han et al.\textsuperscript{25} directed graphs are used and for directed graphs, the memory complexity is quadratic due to which this approach will be less effective for devices with memory constraints. If complete data measurements are inaccessible, Ding and Fang\textsuperscript{28} proposed an FDI approach that deals with intermittent observations. In a case study, Yu et al.\textsuperscript{29} looked at performance degradations induced by incipient faults and estimated the residual existence of a mechatronic system. To increase the monitored system’s separation capability when multiple faults occur, two analytical redundancy relations were addressed.

It is quite evident from the literature that the existing models for the AFR control of an IC engine are either using the lookup tables, linear regression, KF, or ANN. Linear regression is a very basic technique and is less accurate. Lookup tables and ANN techniques are computationally less efficient and KF is limited to the linear region of the sensors. Some other techniques are also using some advanced algorithms like SVM, deep learning, CVA, or EWMA. But in these advanced techniques, we need a large dataset. Our contribution to this research is the implementation of a GA-based observer model in the FDI unit. The proposed method is producing excellent results (up to 77% improvement in the standard error of estimation) despite having a limited dataset. The proposed method uses very basic GA which tends to find the local optimum solution and can be used on the hardware with limited computational power or in time-critical applications. In comparison to previous literature works, the simulation results show superior fault tolerance efficiency, particularly for the MAP sensor, in terms of less oscillatory response.

The rest of the paper is organized in the following order. Problem formulation and computation of fitness functions are presented in section 2, section 3 is explaining results and finally, the paper is concluded in section 4.

**Research methodology**

Using the available AFR model\textsuperscript{30} of the IC gasoline engine, the proposed AFTCS was implemented in MATLAB/Simulink. For analytical redundancy, the proposed AFTCS uses the Genetic Algorithm (GA) based observer model in the Fault Detection and
Isolation (FDI) unit to generate estimated value for the faulty sensors using the values of other healthy sensors. The Fault Injection Unit (FIU) sends a fault to each sensor one at a time, leaving the others unaffected. The engine speed is set to 300 rpm for this analysis based on the MATLAB model’s design speed. We have used constant speed in this research work as the engines mostly run at constant speed in the process plant, and the 300 rpm of speed is provided to a controller by the design FDI when the speed sensor stops working. Load variations and their impact on speed are not considered as the paper is more focused on developing GA-based AFTCS. The modeling and development of the system are further explained in the next subsections:

**Air-fuel ratio control**

An Internal Combustion (IC) engine is a heat engine that uses air to help burn fuel in a combustion chamber. In production processes, IC engines are commonly used as prime movers. Chemical energy is translated into mechanical rotation, which is then used to power the engines’ compressors and alternators. Compression-ignition (CI) and spark-ignition (SI) engines are the two types of IC engines. The compression causes combustion in CI engines, while spark plugs cause combustion in SI engines. The Air Fuel Ratio (AFR) is a calculation of how much air and fuel are combined in a certain ratio during the combustion process. The general architecture of the air-fuel ratio (AFR) of an SI IC engine is shown in the Figure 1.

The mathematical equation of AFR is:

$$AFR = \frac{m_{air}}{m_{fuel}}$$ (1)

The chemical equation is given as:

$$25O_2 + 2C_8H_{18} \rightarrow 16CO_2 + 18H_2O + \text{Energy}$$ (2)

In this equation, the AFR is referred to as the stoichiometric ratio, and its value for gasoline is 14.6:1. The AFR can range from 6:1 to 20:1 depending upon the type of fuel. A rich mixture has a lower AFR value than the stoichiometric ratio and a lean mixture has a higher AFR value than the stoichiometric ratio. For example, the 16.5:1 AFR is lean, while the 13.7:1 AFR is rich in gasoline. Rich and lean mixtures are also hazardous to the engine because they weaken the catalyst and reduce engine performance and efficiency.2,3

Four sensors play an important role in maintaining AFR control of the SI IC engines:

**Manifold Absolute Pressure (MAP) Sensor:** It gives the controller an accurate suction air pressure rating.

**Throttle Sensor:** It provides the controller with an air throttle position.

**Exhaust Gas Oxygen (EGO) Sensor:** It measures the oxygen content in the exhaust gas and manages the fuel supply to ensure efficient combustion.

**Speed Sensor:** To measure the engine crankshaft’s rotational speed, the speed sensor is used. It also provides it to the controller.

An engine shut down can be caused due to faults in the above-mentioned sensors, hence fault tolerance is required. It is desirable to generate virtual redundant sensors in the FDI unit that have a nonlinear response like actual sensors. That is why we have used the GA technique in AFTCS architecture to approximate the system. For the speed sensor fault, we supplied the operating speed of the engine to the FDI unit. For relay type EGO sensor, we supplied 12 V to the FDI unit as zero value comes in the operating range which is from 0 to 1 V. The GA-based observers were built for the MAP and Throttle sensors. MATLAB model lookup tables31 were used to extract the MAP and throttle sensor data for 300 rpm.

**Genetic algorithm**

The genetic algorithm (GA) is a search heuristic algorithm inspired by evolutionary theory. We often use GA to look at alternatives to optimization issues. Genetic algorithms are appropriate for massive search spaces and NP-complete problems. We create several generations to achieve the final solution by GA, and the fittest generation has a better chance of surviving in the next generation.32 The generations of the current population go through crossover and mutation cycles. Each generation has a solution, and the most appropriate or applicable generation is chosen as the final solution. The GA operations are shown in Figure 2.
The steps performed while applying the genetic algorithm to an optimization problem are given below.

The evolution process normally begins with a population of randomly generated individuals and proceeds in an iterative manner, with each iteration’s population being referred to as a generation. Every set of values in the population is assessed in each generation; fitness is normally the value of the objective function in the optimization problem being solved. Once the first generation is evaluated, the next step is to generate the second-generation population. Crossover (also known as recombination) and mutation are two genetic operators that are used to pick a population of solutions. This process continues until an optimal solution is achieved or the termination condition is met. The common termination conditions are:

- An optimum solution is achieved.
- A fixed number of generations are reached.
- Computation time has reached to maximum allocated limit.
- The fitness of the highest-ranking solution is approaching or has reached a point where subsequent iterations no longer yield improved outcomes.
- Combination of above-listed conditions.

The overall working of the genetic algorithm can be seen in Figure 3.

Due to its impressive performance, GA is widely used in optimization and classification problems ranging from fault diagnosis in PV cells, OFDM and OQAM synchronization estimation, OFDM modulation, classification problem solving, hyperparameter optimization of power amplifiers, optimal resource allocation in cloud computing, linear array antenna designing, robotic locomotion optimization, compressor-based system’s slow speed characteristics estimation, and many more similar problems. In this study, we are using the genetic algorithm to predict the estimated values in case of a faulty MAP or throttle sensor in an AFTCS.

We can see from the literature that most of the work on fault detection uses Kalman filters, lookup tables, linear regression, ANN, machine learning, or fuzzy logic. For the nonlinear sensors, linear regression yields less reliable sensor values, while the Kalman filters and lookup table approaches are time-consuming. Computational capacity is needed for methods such as ANN, machine learning, and fuzzy logic. As a result, we have suggested a genetic algorithm-based AFTCS for extremely nonlinear sensors in an IC engine’s AFR system. The suggested methodology is both computationally and time-efficient without sacrificing precision.

Moreover, the traditional calculus-based methods work in the problems in which we start at a random point and go in the direction of the gradient until we reach the top of the hill. This method is fast and effective when dealing with single-peaked objective functions, such as the cost function in linear regression. However, in most real-world scenarios, like finding the optimal estimated values of nonlinear sensors in case of a failure, we face a dynamic problem known as landscapes, which consists of several peaks and valleys, causing certain approaches to fail because they have an intrinsic propensity to get trapped at the local optima. Hence we need a more versatile technique like GA to find the optimal solution for these sorts of problems.

**System modeling**

There are many types of air-fuel ratio control dynamics, including air dynamics, fuel dynamics, sensor model, and controller configuration. Each dynamics’ formulation is given here.

**Air dynamics.** Using the theory of mass conservation and the ideal air gas hypothesis, the air intake dynamics are described as follows:

\[
P_{in} = \frac{RT_{in}}{V_{in}} (m_{th} - m_{Cyr}) + P_{in} \frac{T_{in}}{T_{in}}
\]
\[ P_m' = \Psi(\phi_{ih}, P_m, T_m, N_e) \]  

In the above equations, \( T_m \) is the temperature at the input, \( P_m' \) is pressure manifold, and \( v_m \) is input volume; the mass flow through the valve and into the cylinders are denoted by \( m_{ih} \) and \( m_{Cyl} \), respectively; \( R \) is the gas constant; \( N_e \) is the engine speed and \( \phi_{ih} \) is the throttle opening position. Here the derivative of the intake temperature with respect to time is supposed to be zero. Now, the equation (3) becomes:

\[ P_m = \dot{\theta}_m (m_{ih} - m_{Cyl}) \]  

with

\[ \dot{\theta}_m = \frac{RT_m'}{v_m} \]

The air mass flow through the valve is:

\[ m_{ih} = C_d \frac{P_{id}}{\sqrt{RT_{id}}} S_{so}(\phi_{ih}) g(P_r) \]

\( C_d \) the coefficient of discharge. The variable \( P_{id} \) are the overhead loading pressure and the load ratio \( P_r \) is defined as \( P_r = \frac{P_{id}}{P_{id}} \). \( S_{so}(\phi_{ih}) \) is the throttle opening area. The product \( C_d S_{so}(\phi_{ih}) \) is specified as an effective opening throttle area. This variable is represented as:

\[ S_{so}(\phi_{ih}) = \alpha_1(1 - \cos(\alpha_2 \phi_{ih} + \alpha_3)) + \alpha_4 \]

Where \( g(P_r) \) is considered to be a non-linear term as:

\[ g(P_r) = \begin{cases} \sqrt{\frac{2}{\gamma - 1}(P_r - 1)\frac{\gamma + 1}{\gamma - 1}} & \text{if} \ P_r > \left( \frac{\gamma + 1}{\gamma - 1} \right) \frac{\gamma - 1}{2} \\ \sqrt{\gamma} \left( \frac{\gamma + 1}{2}\frac{\gamma - 1}{\gamma - 1} \right) & \text{if} \ P_r \leq \left( \frac{\gamma + 1}{\gamma - 1} \right) \frac{\gamma - 1}{2} \end{cases} \]

Here \( \gamma \) is the air heat ratio and its value is considered to be 1.4.

**Fuel Dynamics.** The fuel dynamics are well demonstrated in Sui and Hall\(^{44}\) as:

\[
\begin{align*}
\dot{\bar{m}}_f(t) &= -\frac{1}{\tau_f}(\bar{m}_f(t) + \alpha \bar{m}_f(t)) \\
\bar{m}_f(t) &= (1 - \alpha)\bar{m}_f(t) \\
\bar{m}_f(t) &= \bar{m}_f(t) + \bar{m}_f(t)
\end{align*}
\]

Where \( \tau_f \) is the fuel vapor process at constant time \([s]\), \( \bar{m}_f(t) \) the fuel flow injection \([kg/s]\), \( \bar{m}_f(t) \) the fuel in the cylinders \([kg/s]\), \( \bar{m}_f(t) \) the vapor flow \([kg/s]\), and \( \bar{m}_f(t) \) the liquid mass fuel flow \([kg/s]\), it is possible to include \( \alpha \) as a vector depending on the throttle opening or engine rpm \( N_e \), to achieve a more detailed model.\(^{45}\) In our case the second solution is more feasible:

\[
\tau_f(N_e) = \sigma_5 N_e^{-\sigma_6}
\]

\[ \kappa(N_e) = \sigma_7 + \sigma_8 N_e \]

The AFR will now become:

\[ \lambda_{cyl} = \frac{\bar{m}_{cyl}(t)}{\bar{m}_f(t)} \]

**Sensor model.** The expression of lambda (\( \lambda \)) sensor model is given by:

\[
\dot{\lambda}(t) = -\frac{1}{\tau_\lambda}\lambda(t) + \frac{1}{\tau_\lambda}\lambda_{cyl}(t - \tau(N_e(t)))
\]

Here \( \tau_\lambda \) is the fixed time delay and its value is 0.1 s.

An engine speed \( N_e(t) \) with time delay \( \tau \) is shown in the following equation:

\[ \tau(N_e(t)) = \frac{60}{N_e(t)} \left( 1 + \frac{1}{n_{cyl}} \right) \]

**State space representation.** We can represent the state-space. We are using the state-space representation.

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2
\end{bmatrix} = A \begin{bmatrix} x_1 \\
x_2
\end{bmatrix} + B \begin{bmatrix} u_1 \\
u_2
\end{bmatrix}
\]

\[
y = C \begin{bmatrix} x_1 \\
x_2
\end{bmatrix} + D \begin{bmatrix} u_1 \\
u_2
\end{bmatrix}
\]

\[
\begin{cases}
\dot{x}_1 = f_1(\cdot)x_1(t) - f_2(\cdot)u(t) \\
\dot{x}_2 = -\frac{1}{\tau_\lambda}\lambda(t) + \frac{1}{\tau_\lambda}\lambda_{cyl}(t - \tau(N_e(t)))
\end{cases}
\]

With \( x_1(t) = \lambda_{cyl}, x_2(t) = \lambda(t), \) and \( u(t) = \bar{m}_f(t) \):

\[
f_1(\cdot) = -\frac{1}{\tau_\lambda}(N_e) - \frac{\bar{m}_{cyl}(N_{e}, P_{in})}{\tau(N_e)}
\]

\[
f_2(\cdot) = \lambda \frac{\chi(N_e)}{\tau(N_e)} \bar{m}_{cyl}(N_e, P_{in})
\]

Bounded as follows: \( f \leq f_i(\cdot) \leq f_i, \) for \( i \in \{1, 2\} \).

**Controller design.** In Wang et al\(^{46}\) have demonstrated the working of AFTCS in state-space. We are using the same to explain the architecture of observer.

\[
\dot{\hat{x}} = Ax + Bu
\]

\[
y = Cx + Du
\]

\[
\dot{\hat{x}} = Ax + Bu
\]
where \( x \) is the state vector, \( u \) is the input vector and \( y \) is the output vector. \( A, B, C, \) and \( D \) are system matrices.

\[
\ddot{\dot{x}} - \dot{x} = A(\ddot{x} - x) 
\]

(25)

\[
(y - y) = C(\ddot{x} - x) 
\]

(26)

\[
\ddot{x} = Ax + Bu + L(y - y) 
\]

(27)

In (24), \( L \) is the feedback gain.

FDI will not declare any fault when the residual “\( e_x \)” goes to zero. If the residual is out of bound, an error will be declared and the faulty value will be replaced. The complex structure of AFTCS, as well as its slow response time due to excessive computations, are two of its main drawbacks.47

The non-linear observer design equation we get from (27) is.

\[
\dot{x}(t) = Ax(t) + Bu + g(\ddot{x}, u, t) + L(C\ddot{x} - y) 
\]

(33)

“\( g(x, u, t) \)” is a nonlinear function and is assumed to be globally Lipschitz. \( L \) is the gain for the nonlinear observer.

Let \( e_x(t) \) be the error vector.

\[
e_x(t) = \ddot{x}(t) - x(t) 
\]

(34)

From equations (33) and (34), we get.

\[
e_x = (A + LC)e_x + (g(\ddot{x}, u, t) - g(x, u, t)) 
\]

(35)

**Lemma 1:** The error \( e_x(t) \) asymptotically goes to zero if there exists a matrix \( R, X \), and scalar \( \mu \) such that \( R = R^T > 0 \) and \( \mu > 0 \) to satisfy the following linear matrix inequality (LMI):

\[
\begin{bmatrix}
RA + A^TR + XC + C^TX^T + \mu \lambda^2 I & R \\
R & -\mu I
\end{bmatrix} < 0
\]

(36)

The reliability of each sensor is denoted by \( R \).

The observer gain matrix can be selected as follows:

\[
L = R^{-1}X
\]

(37)

By considering the following Lyapunov function to prove its derivative to be zero we can justify the choice of observer gain matrix:

\[
V(t) = e_x^T(t)Re_x(t)
\]

(38)

Substituting (38) in (39) we get:

\[
\dot{V}(t) \leq e_x^T((RA + XC + A^TR + C^TX^T) + \mu \lambda^2 I + 1/\mu R^2)e_x
\]

(39)

If the following inequality holds, \( e_x \) asymptotically converges to zero.

\[
(RA + XC + A^TR + C^TX^T + \mu \lambda^2 I + 1/\mu R^2) < 0
\]

(40)

This inequality becomes equal to (36) by applying Schur complement, which completes the proof.

**Theorem 1:** The error \( e_x(t) \) approaches to zero exponentially with rate \( \kappa/2 \) if there exist matrices \( R, X \) and scalars \( \mu, \kappa \) such that \( R = R^T > 0 \) and \( \mu, \kappa > 0 \) to satisfy the following:

\[
\begin{bmatrix}
RA + A^TR + XC + C^TX^T + \mu \lambda^2 I + kR & R \\
R & -\mu I
\end{bmatrix} < 0
\]

(41)

(41)

To prove this, consider (40) and (42) to get

\[
\dot{V}(t) \leq -ke_x^TRe_x = -kV(t)
\]

(42)
Hence, we can write
\[ V(t) \leq e^{\lambda_1} V(0) \]  
(44)

From (38) we get
\[ \lambda_{\text{min}}(R) \| e_x(t) \|^2 \leq e^{-k_2} \lambda_{\text{max}}(R) \| e_x(0) \|^2 \]  
(45)

Where \( \lambda_{\text{min}} \) and \( \lambda_{\text{max}} \) are the minimum and maximum Eigenvalues of \( R \), respectively. Hence, we obtain the following norm:
\[ e_x(t) \leq \sqrt{\frac{\lambda_{\text{max}}(R)}{\lambda_{\text{min}}(R)}} e_x(0) e^{-k_2/2} \]  
(46)

Coming back to the residual equation:
\[ r(t) = \| C \dot{x}(t) - y(t) \| \]  
(47)
\[ r(t) \leq \sqrt{\frac{\lambda_{\text{max}}(R)}{\lambda_{\text{min}}(R)}} \| C e_x(0) \| e^{-k_2/2} \]  
(48)
\[ \| C \| \| e_x(0) \| = \| r(0) \| \]  
(49)

Finally, we can come up with the following criteria for fault detection of a sensor fault.
\[ r(t) \left\{ \begin{array}{ll}
\leq \sqrt{\frac{\lambda_{\text{max}}(R)}{\lambda_{\text{min}}(R)}} \| r(0) \| e^{-k_2/2}, & \text{there is no fault} \\
> \sqrt{\frac{\lambda_{\text{max}}(R)}{\lambda_{\text{min}}(R)}} \| r(0) \| e^{-k_2/2}, & \text{there is a fault}
\end{array} \right. \]  
(50)

Figure 4 depicts the working of the proposed AFTCS.

The sensor values are tested and the sensor-to-observer value threshold (\( \xi \)) is calculated when the device first starts up. The fault is detected by measuring the residual \( e_x \) and comparing it to the threshold (\( \xi \)) as follows:
- If \( e_x < \xi \), the sensor has no fault.
- If \( e_x \geq \xi \), a fault is diagnosed in the sensor and will be replaced by the observer output.

The engine works as expected if there is no fault. When a single sensor fails, the error signal, on the other hand, exceeds the threshold. To replace the faulty sensor value, the FDI unit feeds the Engine Control Unit (ECU) an estimated value derived from the observer model based on GA. The contribution of the estimated simulated value of the faulty sensor provides analytical redundancy to the model.

**Fitness function formulation**

In order to utilize the GA for the estimation of faulty sensor values, we need the fitness function. In this section, the mathematical formulation of fitness functions for the MAP and throttle sensor estimations is given.

---

**Fitness function for MAP estimation.** To formulate the fitness function for the MAP estimation the sum of the least-squares method is used. An exponential equation given by (51) is used to predict the values of MAP sensors in case of a fault.

\[ y_e = 1 - e^{ax} \]  
(51)

Here \( y_e \) is the estimated MAP value, \( x \) is throttle angle, \( a \) and \( b \) are unknown constants. The difference between estimated MAP value (\( y_e \)) and actual MAP value (\( y \)) is calculated and the sum of difference over the entire range of \( x \) is computed to find the fitness function \( f_{M} \).

The expression for the fitness function is given as.
\[ f_{M} = \frac{1}{2 \pi} \sum_{i=1}^{n} (y_i - y_{ei})^2 \]  
(52)

Using (51) in (52), we get
\[ f_{M} = \frac{1}{2 \pi} \sum_{i=1}^{n} \left[ y_i - \left(1 - e^{ax_i}\right)\right]^2 \]  
(53)

Here \( n = 17 \), and the values of \( x \) and \( y \) are given in Table 1 below.

The fitness function given by (53) is then minimized by using GA to find the best possible values of the unknown coefficients “a” and “b.”
Fitness function for throttle estimation. To formulate the fitness function for the throttle estimation the sum of the least-squares method is used. A polynomial of fourth order given by (54) is used to predict the values of throttle angle in case of a fault.

\[ y_e = ax^4 - bx^3 - cx^2 + dx - e \]  

(54)

Here \( y_e \) is the estimated throttle value, \( x \) is MAP value, \( a, b, c, d, \) and \( e \) are unknown constants. The difference between estimated throttle angle (\( y_e \)) and actual throttle angle (\( y \)) is calculated and the sum of difference over the entire range of \( x \) is computed to find the fitness function \( f_T \). The expression for the fitness function is given as.

\[ f_T = \frac{1}{2n} \sum_{i=1}^{n} (y_i - y_e)^2 \]  

(55)

Using (54) in (55), we get

\[ f_T = \frac{1}{2n} \sum_{i=1}^{n} \left[ y_i - (ax^4 - bx^3 - cx^2 + dx - e) \right]^2 \]  

(56)

Here \( n = 19 \), and the values of \( x \) and \( y \) are given in Table 2 below.

The fitness function given by (56) is then minimized by using GA to find the best possible values of the unknown coefficients \( a, b, c, d, \) and \( e \).

Despite having good performance, the GA has some limitations which are listed below.

- The GA used in this study is only suitable for the fitness functions of moderate complexity. If the fitness function is complex or too simple, then GA efficiency will be very low in terms of computation time and resources usage.
- The solution generated by GA is relatively better than the other solutions computed in GA, so we need to repeatedly plot the actual and estimated results.
- For multivariable fitness functions the GA sometimes generates the local minima, so we need to adjust the range of solution values accurately.
- Despite these limitations, we have used GA over the other techniques due to following reasons.

- Unlike other optimization algorithms like Particle Swarm Optimization (PSO), GA is discrete and is used for the problems in which we have a discrete set of data. In our case, we have used the finite set of values for the MAP and throttle sensors so GA is more suitable here.
- GA finds the local maxima/minima instead of finding the global optimum solution hence it tends to provide the solutions more quickly.

We have considered the full type sensor faults in this study and assumed that the sensor becomes a short circuit in a faulty condition. The partial type faults in which the sensor does not become fully faulty rather gives partial output are not considered as a limitation of the study. Moreover, zero switching time is assumed in the sensor switches process. Practically, a certain time delay occurs in the switching process.

Results and discussion

Fitness function optimization with GA

MAP estimation. The fitness function given by (53) is minimized to find the best possible values of unknown coefficients “a” and “b” by using the GA function in MATLAB. With the default settings of GA, the results were not satisfactory, so the GA parameters were changed to get suitable results. As the fitness function was

| Table 1. | Throttle angle (x) and MAP values (y) at 300 RPM. |
|----------|-----------------------------------------------|
| x (Throttle angle) | 0° | 3° | 6° | 9° | 12° | 15° | 18° | 21° | 24° |
| y (MAP value) | 0.091 | 0.114 | 0.191 | 0.329 | 0.545 | 0.745 | 0.857 | 0.915 | 0.946 |

| Table 2. | MAP value (x) and Throttle angles (y) at 300 RPM. |
|----------|-----------------------------------------------|
| x (MAP value) | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 |
| y (Throttle angle) | 0° | 1.979° | 4.687° | 6.259° | 7.471° | 8.482° | 9.357° |
| x (MAP value) | 0.40 | 0.45 | 0.50 | 0.55 | 0.60 | 0.65 |
| y (Throttle angle) | 10.131° | 10.824° | 11.452° | 12.061° | 12.704° | 13.402° |
| x (MAP value) | 0.70 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
| y (Throttle angle) | 14.188° | 15.107° | 16.243° | 17.754° | 20.033° | 24.497° |
having several minimums, so to find the optimum minimum, the number of generations was kept to 50, and the range of values was set to [0.001, 2]. With these settings the values of “a” and “b” are found to be:

\[ a = 0.0251 \quad b = 1.4392 \]

The fitness function value for each generation and the mean fitness values along with the best individual values of “a” and “b” are shown in Figure 5.

Once the fitness function is minimized and the optimum values of unknown coefficients are found, the estimated values are found by using MATLAB. The relation between original and estimated values is elaborated by the line fit plot as given below in Figure 6.

In Figure 6, the \( R^2 \) is equal to 0.987 and we can notice that the estimated values are very close to the original values which proves the efficiency of GA for the estimation of MAP values. We can also notice from Figure 5 the GA reached its optimum solution very quickly and it took only 42 generations to minimize the fitness function of the MAP sensor.

**Throttle estimation.** The fitness function given by (56) is minimized to find the best possible values of unknown coefficients “a,” “b,” “c,” “d,” and “e” by using the GA function in MATLAB. With the default settings of GA, the results were not satisfactory, so the GA parameters were changed to get suitable results. As the fitness function was having several minimums, so to find the optimum solution, the number of generations was kept to 250, and the range of parameter values was set to [40, 60]. With these setting the values of “a,” “b,” “c,” “d,” and “e” is found to be:

\[ a = 60.2781 \quad b = 40.6871 \quad c = 39.0451 \quad d = 49.5202 \quad e = 2.1916 \]

The fitness function value for each generation and the mean fitness values along with the best individual values of unknown coefficients are shown in Figure 7 below.

Once the fitness function is minimized and the optimum values of unknown coefficients are found, the estimated values are found by using MATLAB. The relation between original and estimated values is elaborated by the line fit plot as given below in Figure 8.

In Figure 8 the \( R^2 \) is equal to 0.997 and we can notice that the estimated values are very close to the original values which proves the efficiency of GA for the estimation of Throttle values. It can also be noticed from Figure 7 that the number of unknowns was 5 for throttle estimation so it took GA a little bit longer to reach the optimum solution. GA reached its optimum solution in 192 generations as compared to 42 for MAP estimation.

**MATLAB simulation results**

The proposed AFTCS for the AFR controller was implemented in MATLAB/Simulink in which GA estimation results obtained in section 3.1 were used to
construct the observer model in the FDI unit. The FIU injects the fault into each sensor one at a time, leaving the others in a healthy state. In standard conditions, the mixture AFR ratio is held at 14.6, but in defective conditions, it drops to 11.7 (rich mixture). The system, however, retains reliability despite decay, which is consistent with AFTCS’s architecture philosophy.

The AFTCS part is simulated with faults in the sensors taking one sensor at a time and the effects on the AFR are observed at $t = 5$ s due to internal warm-up delay incorporated in the model. Figure 9 depicts the performance of the proposed AFTCS under both normal and faulty situations.

Results of Figure 9 show that the AFR gets degraded to 11.7 in the steady-state with the fault in any one sensor with the AFTCS part alone. The slightly different transient behavior with the MAP sensor fault is caused by the MAP value at a throttle angle of $0^\circ$. The approximate output value obtained from the GA estimation block is slightly different from the actual value, hence it is causing a negligible oscillatory response.

Comparison with existing techniques

In this section, a comparison of the proposed GA-based AFTCS has been carried out with other existing techniques found in the literature. First, we have presented a detailed comparison for the MAP and throttle sensor estimation results with paper.11 Afterward, we have also given a comparison with some other similar papers to show the superior performance of the proposed control system.

MAP sensor. To back up the point of improved accuracy, the same data is subjected to linear regression and MAP values are calculated. Line fit plots, as shown in Figure 10, show a comparison of predicted MAP values for GA-based estimation and linear regression.

We can notice from Figure 10 that linear regression is poorly estimating the sensor values. On the other hand, the GA estimated values are very close to the original values of the sensor in the nonlinear range. Table 3 shows the estimation statistics for both GA and linear regressions to demonstrate it mathematically.

Table 3 demonstrates GA estimation’s superior efficiency, as the adjusted $R^2$ and $R^2$ values for GA estimation are much higher than their linear regression counterparts. In linear regression, the adjusted $R^2$ is 0.4722 which is an indicator of poor performance of linear regression. Only $R^2$ or adjusted $R^2$ values are insufficient for comparison in estimation problems, so we computed the standard error of estimation, which is the best measure of estimation accuracy. Table 3 shows that the SEE in GA estimation is 0.0406 while it is 0.2507 in linear regression. An 83.5053% increase in SEE is indicating that GA estimation produces more reliable results.

Throttle sensor. To back up the point of improved precision, the same data is subjected to linear regression, and throttle angles are calculated. The line fit plots for both GA and linear regressions, as shown in Figure 11, provide a comparison of approximate throttle angles for GA and linear regression.

The line fit plots show that GA estimation is more reliable than linear regression in generating accurate results. Table 4 shows the estimation statistics for both GA and linear regressions to demonstrate it mathematically.

The standard error of the estimate is calculated along with other estimation statistics, just as in MAP estimation. The SEE in GA estimation is 0.3302, while the
SEE in linear regression is 1.4361, demonstrating the superior output of GA estimation and a 77.0071% improvement in SEE. Similarly, for GA estimation, the adjusted $R^2$ and $R^2$ values are superior to their linear regression counterparts.

Apart from linear regression-based observer design, the existing models used lookup tables, KF, and ANN for the fault-tolerant AFR system design. Lookup tables and ANN techniques are computationally less efficient and KF is limited to the linear region of the sensors. Some other techniques are also using some advanced algorithms like SVM, deep learning, CVA, or EWMA. But in these advanced techniques, we need a large dataset. Our contribution to this research is the implementation of a GA-based observer model in the FDI unit. The proposed method is producing excellent results (up to 77% improvement in the standard error of estimation) despite having a limited dataset. The proposed method uses GA which tends to find the local optimum solution due to its discrete nature hence the GA will be computationally effective in this case. The tendency of premature convergence in PSO is also high as compared to GA. Forward propagation and backpropagation are the two main processes in ANN training and the complexities of these operations are of fifth-degree polynomial. Moreover, the backward propagation has the property of low convergence rate and weak generalization capacity which makes it computationally inefficient. Therefore, for an optimization problem, the computational complexity depends upon the problem itself instead of the algorithm. And the estimation problem considered in this study is having a limited dataset and is of moderate complexity, hence GA would be a suitable option in this case.

**Conclusion**

In this paper, an improved AFTCS for the AFR control system of an IC engine was proposed to improve reliability and avoid expensive shutdowns due to sensor failures. The FDI unit in the proposed AFTCS was developed using a genetic algorithm-based observer unit. The model was checked for different sensor faults in the MATLAB/Simulink environment. The proposed model produced excellent results (up to 77% improvement in the standard error of estimation) despite having a limited dataset.
The proposed method used GA which tends to find the local optimum solution and can be used on the hardware with limited computational power or in time-critical applications. In comparison to previous literature works, the simulation results showed superior fault tolerance efficiency, particularly for the MAP sensor, in terms of less oscillatory response.

Since we used full-type sensor faults as a limiting case of testing, future studies could use the AFTCS architecture for partial-type sensor faults with the hardware-in-loop-verification of the proposed algorithm. Another direction could be considering non-fragile control to deal with the various disturbances in a practical scenario.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD
Arslan Ahmed Amin https://orcid.org/0000-0001-8035-595X

References
1. Amin AA and Hasan KM. A review of fault tolerant control systems: advancements and applications. Measurement 2019; 143: 58–68.
2. Amin AA and 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.
3. Wang X. Active fault tolerant control for unmanned underwater vehicle with sensor faults. IEEE Trans Instrum Meas 2020; 69: 9485–9495.
4. Amin AA and 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.
5. Amin AA and 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.
6. Liu C, Yue X and Yang Z. Are nonfragile controllers always better than fragile controllers in attitude control performance of post-capture flexible spacecraft? Aerosp Sci Technol 2021; 118: 107053.
7. Ahmed S, Amin AA, Wajid Z, et al. Reliable speed control of a permanent magnet DC motor using fault-tolerant H-bridge. Adv Mech Eng 2020; 12(10): 1–14.
8. Amin AA and 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. Epub ahead of print 11 April 2021. DOI: 10.1080/03772063.2021.1906767
9. Han S-Y, Zhou J, Chen Y-H, et al. Active fault-tolerant control for discrete vehicle active suspension via reduced-order observer. IEEE Trans Syst Man Cybern Syst 2021; 51: 6701.
10. Escobar-Jimenez RF, Carbot-Rojas DA, Gómez-Aguilar JF, et al. Actuator fault tolerant control based on a MIMO-MPC: application in a double-pipe heat exchanger. Chem Eng Commun 2017; 204: 86–96.
11. Amin AA and 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.
12. Shahbaz MH and Amin AA. Design of active fault tolerant control system for air fuel ratio control of internal combustion engine.
14. Yuan Y, Liu X, Ding S, et al. Fault detection and location system for diagnosis of multiple faults in aeroengines. *IEEE Access* 2017; 5: 17671–17677.

15. Carbet-Rojas DA, Escobar RF, Gómez-Aguilar JF, et al. Experimental validation of an actuator fault tolerant control system using virtual sensor: Application in a double pipe heat exchanger. *Chem Eng Res Des* 2015; 104: 400–408.

16. Peni T, Vanek B, Szabo Z, et al. Supervisory fault tolerant control of the NASA AirStar aircraft. *Proc Am Control Conf* 2014; 666–671.

17. Stepanyan V, Krishnakumar K and Bencomo A. Identification and reconﬁgurable control of impaired multi-rotor drones. In: 2016 AIAA Guid Navig Control Conf, 2016, pp.1–20.

18. Shi K, Liu C, Sun Z, et al. Coupled orbit-attitude dynamics and trajectory tracking control for spacecraft electromagnetic docking. *Appl Math Model* 2022; 101: 553–572.

19. Zhang J, Zeng G and Zhang S. Disturbance observer-based nonsingular terminal sliding mode control for spacecraft electromagnetic docking. *Int J Aerosp Eng* 2020; 2020: 1–20.

20. Liu C, Yue X, Zhang J, et al. Active disturbance rejection control for delayed electromagnetic docking of spacecraft in elliptical orbits. *IEEE Trans Aerosp Electron Syst* 2022; 58: 2257–2268.

21. Maki M, Jiang J and Hagino K. A stability guaranteed active fault-tolerant control system against actuator failures. *Int J Robust Nonlinear Control* 2004; 14: 1061–1077.

22. Yang H and Yin S. Descriptor observers design for Markov jump systems with simultaneous sensor and actuator faults. *IEEE Trans Automat Contr* 2019; 64: 3370–3377.

23. Li X, Yang X, Yang Y, et al. A novel diagnostic and prognostic framework for incipient fault detection and remaining service life prediction with application to industrial rotating machines. *Appl Soft Comput* 2019; 72: 105564.

24. Han J, Zhang H, Wang Y, et al. Fault estimation and fault-tolerant control for switched fuzzy stochastic systems. *IEEE Trans Fuzzy Syst* 2018; 26: 2993–3003.

25. Han J, Liu X, Gao X, et al. Intermediate Observer-based robust distributed fault estimation for nonlinear multiagent systems with directed graphs. *IEEE Trans Ind Inform* 2020; 16: 7426–7436.

26. Han J, Liu X, Wei X, et al. A dynamic proportional-integral observer-based nonlinear fault-tolerant controller design for nonlinear system with partially unknown dynamic. *IEEE Trans Syst Man Cybern Syst* 2021; 52: 1–13.

27. Sreekumar T and Jiji KS. Comparison of Proportional-Integral (P-I) and Integral-Proportional (I-P) controllers for speed control in vector controlled induction motor drive. In: 2012 2nd International conference on power, control and embedded systems, Allahabad, Uttar Pradesh, India, 2012, pp.1–6. New York, NY: IEEE.

28. Ding B and Fang H. Fault estimation and prediction for nonlinear stochastic system with intermittent observations: Fault estimation and prediction for nonlinear system. *Int J Robust Nonlinear Control* 2018; 28: 1165–1181.

29. Yu M, Li H, Jiang W, et al. Fault diagnosis and RUL prediction of nonlinear mechatronic system via adaptive genetic algorithm-particle filter. *IEEE Access* 2019; 7: 11140–11151.

30. MathWorks. Modeling a fault-tolerant fuel control system - MATLAB & Simulink - MathWorks Benelux, https://nl.mathworks.com/help/simulink/examples/modeling-a-fault-tolerant-fuel-control-system.html (1997, accessed April 2021).

31. MathWorks. Modeling engine timing using triggered subsystems - MATLAB & Simulink - MathWorks France, https://fr.mathworks.com/help/simulink/sref/modeling-engine-timing-using-triggered-subsystems.html (1997, accessed April 2021).

32. Mitchell M. *An introduction to genetic algorithms*. Cambridge, MA: MIT Press, 1996.

33. Tao C, Wang X, Gao F, et al. Fault diagnosis of photovoltaic array based on deep belief network optimized by genetic algorithm. *Chin J Electr Eng* 2020; 6: 106–114.

34. Yongjin L, Xihong C and Yu Z. Joint synchronization estimation based on genetic algorithm for OFDM/OQAM systems. *J Syst Eng Electonics* 2020; 31: 657–665.

35. Shi Y, Lu X, Gao K, et al. Genetic algorithm aided OFDM with all index modulation. *IEEE Commun Lett* 2019; 23: 2192–2195.

36. Naveed S and Iqbal MS. Genetic approach to solve tower of Hanoi problem. Karachi, Pakistan: PAF-KIET, 2016.

37. Wang S, Roger M, Sarrazin J, et al. Hyperparameter optimization of Two-Hidden-Layer neural networks for power amplifiers behavioral modeling using genetic algorithms. *IEEE Microw Wirel Compon Lett* 2019; 29: 802–805.

38. Reis T, Teixeira M, Almeida J, et al. A recommender for Resource Allocation in compute clouds using genetic algorithms and SVR. *IEEE Lat Am Trans* 2020; 18: 1049–1056.

39. Hamza A, Attia H, Iqbal MS, et al. Optimization and simulation-time reduction of adaptive antenna array employing Schelkunoff nulling theory. *IEEE Trans Antennas Propag* 2020; 68: 3295–3300.

40. Chavez Estrada FA, Herrera Lozada JC, Sandoval Gutierrez J, et al. Performance between algorithm and micro genetic algorithm to solve the robot locomotion. *IEEE Lat Am Trans* 2019; 17: 1244–1251.

41. Wang Z, Li J, Fan K, et al. Prediction method for low speed characteristics of compressor based on modified similarity theory with genetic algorithm. *IEEE Access* 2018; 6: 36834–36839.

42. Lauber J, Khiar D and Guerra TM. Air-fuel ratio control for compression ignition engines with high dilution and boost levels. *Proc IMechE, Part D: J Automob Eng* 2019; 233: 1834–1850.
45. Hendricks E and Sorenson SC. Mean value modelling of spark ignition engines. SAE technical paper 900616, 1990.
46. Wang Y, Zhou D, Qin SJ, et al. Active fault-tolerant control for a class of nonlinear systems with sensor faults. *Int J Control Autom Syst* 2008; 6: 339–350.
47. Yang H, Yin S and Kaynak O. Neural network-based adaptive fault-tolerant control for Markovian jump systems with nonlinearity and actuator faults. *IEEE Trans Syst Man Cybern Syst* 2021; 51: 3687–3698.
48. Shahid S and Singla R. A comparative study of genetic algorithm and the particle swarm optimization. *Int J Electr Eng* 2016; 9: 215–223.
49. Sohail MS, Bin Saeed MO, Rizvi SZ, et al. Low-complexity particle swarm optimization for time-critical applications. *arXiv:14010546* 2014; 1–24.
50. Kachitvichyanukul V. Comparison of three evolutionary algorithms: GA, PSO, and DE. *Ind Eng Manag Syst* 2012; 11: 215–223.
51. Wang S, Zheng F and Xu L. Comparison between particle swarm optimization and genetic algorithm in artificial neural network for life prediction of NC tools. *J Adv Manuf Syst* 2008; 7: 1–7.