International Journal of
Turbomachinery
Propulsion and Power

MDPI

Article

Fuzzy Controller Structures Investigation for Future Gas Turbine Aero-Engines

Seyed Jalal Mohammadi Doulabi Fard [1] and Soheil Jafari [2,*]

[1] Department of Mechanical Engineering, Iranian University of Science and Technology (IUST), Tehran 13114-16846, Iran; jalalmohammady@gmail.com
[2] Centre for Propulsion Engineering, School of Aerospace Transport and Manufacturing (SATM), Cranfield University, Cranfield MK43 0AL, UK
* Correspondence: s.jafari@cranfield.ac.uk

Abstract: The Advisory Council for Aeronautics Research in Europe (ACARE) Flight Path 2050 focuses on ambitious and severe targets for the next generation of air travel systems (e.g., 75% reduction in CO_2 emissions per passenger kilometer, a 90% reduction in NOx emissions, and 65% reduction in noise emission of flying aircraft relative to the capabilities of typical new aircraft in 2000). In order to meet these requirements, aircraft engines should work very close to their operating limits. Therefore, the importance of advanced control strategies to satisfy all engine control modes simultaneously while protecting them from malfunctions and physical damages is being more crucial these days. In the last three decades, fuzzy controllers (FCs) have been proposed as a high potential solution for performance improvement of the next generation of aircraft engines. Based on an analytic review, this paper divides the trend of FCs design into two main lines including pure FCs (PFC) and min–max FCs (MMFC). These two main architectures are then designed, implemented on hardware, and applied in a case study to analyze the advantages and disadvantages of each structure. The analysis of hardware-in-the-loop (HIL) simulation results shows that the pure FC structure would be a high potential candidate for maneuverability and response time indices improvement (e.g., military applications); while min–max FC architecture has a great potential for future civil aero-engines where the fuel consumption and steady-state responses are more important. The simulation results are also compared with those of industrial min–max controllers to confirm the feasibility and reliability of the fuzzy controllers for real-world application. The results of this paper propose a general roadmap for fuzzy controllers' structure selection for new and next generation of aircraft engines.

Keywords: gas turbine engines; next generation of aircraft engines; fuzzy logic controller; min–max control strategy; hardware-in-the-loop (HIL) simulation

Citation: Mohammadi Doulabi Fard, S.J.; Jafari, S. Fuzzy Controller Structures Investigation for Future Gas Turbine Aero-Engines. *Int. J. Turbomach. Propuls. Power* **2021**, *6*, 2. https://doi.org/10.3390/ijtpp6010002

Received: 14 February 2020
Accepted: 17 February 2021
Published: 22 February 2021

1. Introduction

A gas turbine engine (GTE) is a type of continuous combustion engines. Main elements common in all GTEs are an upstream rotating gas compressor, a combustion chamber, and a downstream turbine on the same shaft as the compressor. This combination is usually called gas generator (GG). By adding other elements to the GG, the GTE can be used for different applications. Nowadays, GTEs have many applications like surface vehicles (race cars, tanks, locomotives, etc.), aircraft and rotorcraft engines, ships and marine applications, heavy-duty gas turbines for power plants, and integrated renewable and gas-fired energy generation systems. Concerning this variety of applications and different operating conditions, different control modes should be defined and satisfied in GTEs. Therefore, like any other mechanical system, a proper control strategy plays a vital role in GTEs safer operation. This control system should increase the engine performance efficiency to meet structural and aerodynamic limitations (control modes). The first aspect for design a proper control system is to know the dynamic behavior of the system and the limitations that should be satisfied during the engine operation. Among different types of GTEs, the issue of gas turbine aero-engine controller design should be taken into account seriously as it is related to the complexity of the engine. The recent designs of gas turbine aero engines are increasingly complex to meet the severe limitations and targets of the future flight paths set by governments and organizations (e.g., Advisory Council for Aeronautics Research in Europe (ACARE) Flight Path 2050). Historically the development of jet engine controllers can be divided into this classification:

1. Hydro-mechanical fuel control, which consists of a simple mechanical actuator controlled by the operator. In other words, in this embodiment, GTEs are manipulated by hydro-mechanical control systems.
2. Hydro-mechanical/electronic fuel control, which is the former fuel flow controller with added an electronic control unit. This electronic unit performed the function of thrust setting, speed governing, and acceleration and deceleration in response to power lever inputs.
3. Digital electronic engine control (DEEC), in this embodiment, functions carried out after input data from the airframe and engine were processed by the DEEC computer included setting the variable vanes, positioning compressor start bleeds, controlling gas-generator, adjusting the augmenter segment sequence valve, and controlling the exhaust nozzle position.
4. Full authority digital engine (or electronic) control (FADEC), works by receiving multiple input variables of the current flight condition including air density, throttle lever position, engine temperatures, engine pressures, and many other parameters. The inputs are received by the electronic engine controller (EEC) and analyzed up to 70 times per second. Engine operating parameters such as fuel flow, stator vane position, air bleed valve position, and others are computed from this data and applied as appropriate. FADEC also controls engine starting and restarting procedures. The FADEC's basic purpose is to provide optimum engine efficiency for a given flight condition [1].

However, principle of using only fuel flow for closed-loop speed control and limit its amount during transients, as in the first hydro-mechanical systems is still the main control strategy in many systems. Other control signals are often open-loop scheduled or used only for limiting engine parameters. As said before, there are several control strategies have been proposed to deal with the above-mentioned requirements dating back to 1952. Each of these algorithms has its advantages and disadvantages. Some of them are not capable of satisfying all engine control modes simultaneously and some of them are weak in some modes and strong in some other modes. A comprehensive review and analysis of the history of GTEs control strategies could be found in [2,3].

The more complicated the engine design, the more demands and limitations on control algorithms. So, this is the time to think differently about the control rules and strategies for gas turbine aero-engines to satisfy new advanced limitations and control requirements for GTEs. For this purpose, different control methods like model predictive control (MPC), linear–quadratic regulator (LQR), linear–quadratic–Gaussian (LQG), and fuzzy logic have been used in the literature recently. Among these algorithms, the fuzzy logic control method has many advantages like similarity to human reasoning, based on the linguistic model, using simple mathematics for nonlinear problems, ability to deal with integrated and complex systems, high precision, rapid operation, and also some disadvantages like needing more fuzzy grades for more accuracy, does not capability to receiving feedback for implementation of learning strategy, restricted number of usage of input variables [4]. Fuzzy logic control is also a heuristic approach that easily embeds the knowledge and key elements of human thinking in the design of nonlinear controllers. Qualitative and heuristic considerations, which cannot be handled by conventional control theory, can be used for control purposes in a systematic form, applying fuzzy control concepts. Fuzzy logic control does not need an accurate mathematical model, can work with imprecise inputs, can handle nonlinearity, and can present disturbance insensitivity greater than the most nonlinear controllers. Fuzzy logic controllers usually outperform other controllers in complex, nonlinear, or undefined systems for which a good practical knowledge exists [5]. Many pieces of evidence show the fuzzy logic theory applications in GTE's controller design [6–11]. Fuzzy and fuzzy-PI controllers for GTE rotor speed control during the startup phase as well as normal operating conditions were designed in [8,9] where the fuel flow is manipulated as a control variable for power plant gas turbine rotor speed control. Fuzzy-PID controller that introduced a better transient response than a PID controller for a power plant gas turbine was proposed and simulated with a linear model in [8]. A fuzzy modified model reference adaptive controller (FMRAC) for GTE rotor speed control was developed and showed its better time response than modified model reference adaptive controller (MRAC) in [9]. A master controller for micro gas turbine generator using the fuzzy control algorithm on fuzzy control processing for on line PID (proportional–integral–derivative) setting parameter was designed and indicated its advantages of fast response and small overshoot in [10]. A gas turbine aero-engine fuzzy controller that optimized by genetic algorithms was also declared in [11]. Moreover, during the last three decades, different investigations have been done on controller design for GTEs based on fuzzy logic methods. The main milestones in the design and simulations of FCs for GTEs are listed in Table 1.

Table 1. Milestones in fuzzy controller design for gas turbine engines (GTEs).

Paper Title	Main Achievement	Publication Year
Fuzzy Computing for Control of Aero Gas Turbine Engines	Certain stipulations, rules, and fuzzy logic are suggested for the control of a single spool aero gas turbine (pure fuzzy)	1994 [12]
Fuzzy Scheduling Control of a Gas Turbine Aero-Engine: A Multi-objective Approach	Combination of fuzzy logic and evolutionary algorithms (EA) to refine the control performance and to increase the flexibility of GTEs (pure fuzzy)	2002 [13]
Fuzzy Fuel Flow Selection Logic for a Real-Time Embedded Full Authority Digital Engine Control	In order to achieve proper performance, Typical control loops chosen by min–max theory are replaced by fuzzy logic loops (min–max fuzzy)	2003 [14]
Advanced Control of Turbofan Engines	Different control loops for turbofan engine control modes are designed and analyzed based on industrial min–max strategy and improved by fuzzy rules with respect to the implementation considerations (min–max fuzzy)	2012 [15]
Heavy-duty gas turbine monitoring based on adaptive neuro-fuzzy inference system: speed and exhaust temperature control	Using an adaptive neuro-fuzzy inference system (ANFIS) to maintain turbine operation at optimum performance. The results obtained, based on the use of the Rowen model, show the effectiveness of the proposed system (pure fuzzy)	2017 [16]
Design of an Interval Fuzzy Type-2 PID Controller for a Gas Turbine Power Plant	The selected model is Rowen's model to present the mechanical behavior of the gas turbine, the main goal is aimed to improve the system dynamic performance, all gains for conventional PID and interval fuzzy type-2 PID are tuned using social spider optimization(SSO) technique, and showed the performance improvement for interval fuzzy type-2 PID controller in comparison with conventional PID via simulation (min–max fuzzy)	2018 [17]
Turbojet engine industrial min–max controller performance improvement using fuzzy norms	The minimum and maximum functions in the industrial min–max strategy are replaced with the different fuzzy norms to improve the performance of the GTE FADEC (min–max fuzzy)	2018 [18]
Fuzzy modeling and fast model predictive control of gas turbine system	For achieving high tracking performance and disturbance rejection ability within less settling time under various operating conditions, an improved fuzzy modeling approach and corresponding fast model predictive control algorithm were introduced and applied to a gas turbine system (pure fuzzy)	2020 [19]

As it can be seen in the above-mentioned studies, there are two main lines for fuzzy controller structure design and two main architectures proposed for satisfying control requirements of GTEs:

- The first structure uses pure fuzzy control (PFC) strategy in which all control rules and loops are replaced by fuzzy rules.
- In the second structure, the controller keeps the industrial min–max structure in which the winner of different control loops will be selected by a pre-defined min–max strategy. However, control loops will be replaced by a fuzzy logic controller to result in a min–max fuzzy controller (MMFC).

The main contribution of this paper is to investigate the advantages and disadvantages of these two structures and to discuss the possibility of using them for the next generations of GTEs towards ACARE Flight Path 2050 requirements. Therefore, this paper will compare two different GTEs fuzzy controllers' structures with each other by developing a framework for real-time hardware implementation. For this purpose, a PFC and an MMFC for a single spool gas turbine aero-engine, as a case study, are firstly designed and described in detail. A hardware-in-the-loop (HIL) stand simulation will then be designed and developed by choosing proper hardware for implementation. Besides, an advanced communication method for software and hardware is designed and implemented to obtain reliable real-time results and control all criteria and requirements. Finally, the simulation results for both implemented controllers are presented and discussed to introduce an initial road-map for the application of fuzzy controllers in gas turbine aero engines.

2. PFC and MMFC Design

Without loss of generality, a single-spool turbojet engine (Figure 1) was selected as a case study in this paper. This engine has an axial three-stage compressor, an annular combustion chamber, and an axial one-stage turbine. The geometry structure of this engine is fixed and therefore applied fuel flow to the combustion chamber is the only parameter that can be used as the control variable. The GTE control system is required to meet the engine thrust regulation and safety constraints simultaneously. As mentioned earlier, the two different structures for the controller that have been selected for design, simulation, and implementation on the hardware in this paper are:

- Pure fuzzy controller (PFC)
- Min–max fuzzy controller (MMFC)

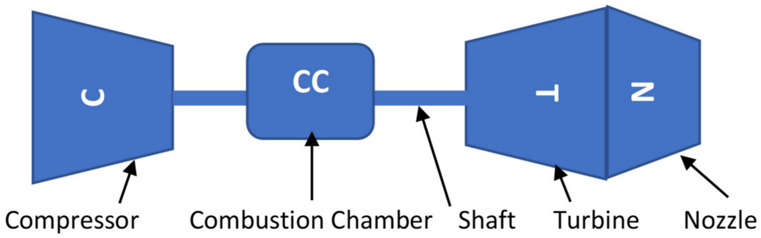


Figure 1. Schematic of single spool turbojet engine (image adopted from [20]).

The controllers will then be implemented on the hardware and the results of real-time simulation will be compared to introduce the suitable fuzzy control algorithm for different applications.

2.1. Pure Fuzzy Controller (PFC) Design

The main idea in this strategy is to use the fuzzy rules to satisfy all engine control modes. In other words, this structure will benefit from the speed, the constraints satisfaction ability, and the flexibility from fuzzy nature. Former GTEs have been manipulated particularly by rotor speed management. This method could deal with engine limitations and be implemented by a hydro-mechanical controller. By emerging digital controllers, the rotor speed derivative (acceleration/deceleration) has been added to controllers' structures as a control variable. Simultaneously speed and acceleration control has led to transient response improvement while engine functional characteristic reaching. As a sample, cold engine acceleration can lead to consuming more fuel for engine components warming and reducing the acceleration rate, while warm engine acceleration is smooth and quick. Rotor acceleration control provides stable acceleration that is self-governing from engine temperature situation. Besides that from the safety point of view, this method has avoided engines from aerodynamic instabilities like surge, rotor over speed, and turbine blades overheating. Respecting to this fact that the derivative part of a controller is more sensitive to rotor speed variation and has a proper time response in comparison with the integral controller part, and integrated signal hardware implementation was time-consuming for operation at HIL purpose, without loss of generality for our story approach this part of the controller was neglected and PD controller chosen for controller design. There is much evidence in scientific researches that illustrate the application of rotor speed and its derivative as control variables for GTEs controllers [21].

The schematic of this type of controller has been shown in Figure 2. As it can be seen in this figure, this controller has two input variables, route error (the difference between throttle command-power lever angle- or desired rotor speed and real rotor speed) and the derivative of this error. Both K1 and K2 coefficients are determined for normalizing membership function inputs, cause the acceptable determined range for these inputs is between −1 to 1, and they stand for mapping inputs to the range of acceptable membership function variables. Based on these inputs and the pre-defined fuzzy rules, the controller calculates transient fuel flow as the output. Moreover, as shown in Figure 2, injected fuel flow to the combustion chamber will be achieved by adding steady-state fuel flow and transient fuel flow. The steady-state part of the fuel flow is calculated by a gain scheduling controller as a function of the engine rotor speed. This function could be derived from the engine performance simulation or experimental results.

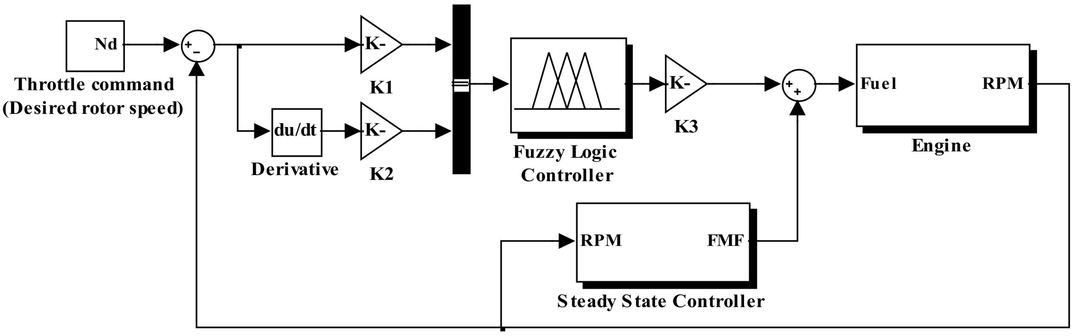


Figure 2. Schematic of pure fuzzy controller (PFC).

It should be mentioned that for controller design, the Mamdani fuzzy inference engine and center of area (COA) defuzzification method are used. In addition, minimum stands for "AND" and maximum stands for "OR". Moreover, for reducing the number of variables, calculation, and time expending for searching, the Gaussian fuzzy membership function is chosen. Each function has two tuning parameters and for PFC design seven linguistic variables were defined as Table 2.

Table 2. Linguistic variables for PFC.

Linguistic Variables	Symbol	Linguistic Variable	Symbol
Negative Big	NB	Positive Big	PB
Negative Medium	NM	Positive Medium	PM
Negative Small	NS	Positive Small	PS
Zero	Z		

The definition of fuzzy rules needs to know the interaction between engine components and awareness of physical damages and probable aerodynamic instabilities that can occur while increasing and decreasing engine rotational speed. As an example, at the primary sharp acceleration time (PB) the transient fuel flow must be Z or PS because the existing engine components' inertia can cause the turbine blade overheating. Also, at the primary sharp deceleration time (NB) the transient fuel flow must be Z or NS because the above-mentioned reason causes flame burnout. Table 3 shows the fuzzy rules for the designed PFC.

Table 3. Fuzzy rules to determine fuel flow changes (FMF) for the PFC.

ΔN \ **ΔNdot**	**Z**	**PS**	**PM**	**PB**	**NS**	**NM**	**NB**
Z	Z	PS	PS	PS	NS	NS	NS
PS	NS	Z	PS	PS	NM	NM	NM
PM	NM	NS	Z	PM	NM	NM	NB
PB	NB	NM	NS	Z	NB	NB	NB
NS	PS	PS	PM	PM	Z	NS	NS
NM	PM	PM	PM	PB	PS	Z	NM
NB	PB	PB	PB	PB	PM	PS	Z

In Table 3 DeltaN stands for PLA-RPM (pilot lever angle and shaft rotational speed those were normalized), DeltaNdot stands for error derivative, and finally FMF stands for transient fuel flow changes. All inputs after receiving by controller were normalized and getting to PFC for calculating proper transient fuel flow and because of this reason all membership function's inputs are between -1 and 1. The rules stated in Table 3 are coming from experts' knowledge and publicly available control laws for jet engine control systems [22].

2.2. *MIN–MAX Fuzzy Controller (MMFC) Design*

Each GTE has three different control modes.

1. Steady-state control mode to meet pilot thrust level requirement.
2. Transient control mode to reach the required thrust in a proper time.
3. Physical limitations control mode to prevent the engine from damages and malfunctions (e.g., over-speed, over-temperature, surge, stall, etc.).

The idea of min–max controller presented by Kreiner. A and Lietzau. K is to design different control loops for each control mode and select the best control loop at each time step based on a pre-defined strategy to satisfy all engine control modes simultaneously [23]. The schematic of a min–max control loop for a turbojet engine is shown in Figure 3. As can be seen in this figure, the controller consists of four control loops as follows:

1. PLA control loop: this loop has to supply the pilot desired thrust in each situation.
2. Maximum speed limitation loop (MSLL): this loop is to prevent the engine from exceeding the rotor speed from the permissible amount. This control loop takes this responsibility to guarantee the integrity of the GTE.
3. Maximum acceleration limitation loop (MALL): at the primary acceleration time abrupt fuel injection is the main cause of aerodynamic instability (surge and stall). MALL loop protects the engine against this fault.
4. Maximum deceleration limitation loop (MDLL): at the primary sharp deceleration time control system must prevent fuel flow from abruptly reducing because the rotor inertia could lead to flame burnout. Therefore, the fuel flow reducing rate must be limited.

After designing the above-mentioned control loops, a pre-defined "MIN–MAX selection strategy" will select transient state fuel flow by using a selection algorithm between these four control loops to satisfy all engine control loops simultaneously. A simple min–max selection strategy for a single spool turbojet engine is as follow:

$$\text{Wftr} = \text{Min (Min (Max (WfPLA, Wfdec), Wfacc), WfNmax)}$$

where WfPLA, Wfdec, Wfacc, WfNmax stand for fuel flow calculated by PLA, MDL, MAL, and MSL loops respectively. In addition, Wftr is the final transient fuel amount. In an acceleration operation, the min-select strategy will protect the engine from surge and over speed whereas, in a deceleration process, the max-select strategy will protect the engine from flameout. If the calculated pilot command transient (PLA) fuel does not exceed these limitations, it will be the winner of the min–max selection strategy. Otherwise, the fuel flow that imposes one of the limitations will be selected as the transient fuel flow in order to protect the engine against failure or malfunction.

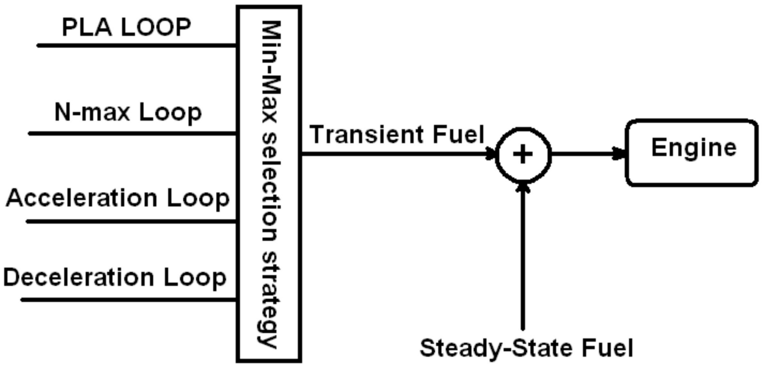


Figure 3. Schematic of a min–max control loop for a turbojet engine.

The min–max control strategy performs very well in satisfying all engine control modes simultaneously. However, regarding engine nonlinear nature applying a linear controller at PLA loop will result in reducing the capability of input order tracing as well as losing the proper flexibility. This reason led researchers to design another form of MIN–MAX controller that fitted by a fuzzy controller for determining transient fuel at PLA, MALL, MDLL loops. Fuzzy min–max control parameters Variety cause to meet multi-control purposes. Figure 4 shows the MMFC structure used in this study where all K1, K2, and K11 coefficients were determined for normalizing membership functions inputs, cause the acceptable determined range for these inputs are between −1 to 1, and they stood for mapping inputs to the range of acceptable membership functions variables. Again, for entering MIN–MAXs selection boxes outputs must be mapped so K22 and K3 were used for this purpose.

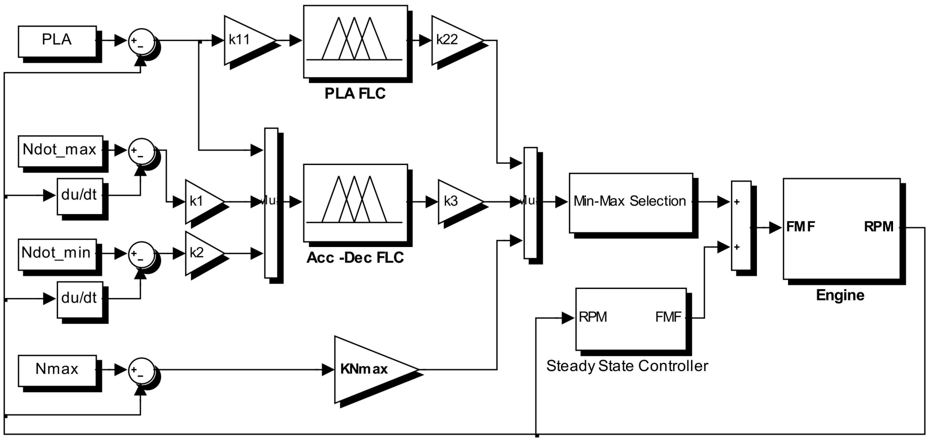


Figure 4. Min–max fuzzy controller (MMFC) Structure.

The MMFC design procedure is as follow:

PLA control loop design: as mentioned before, PLA control loop is to determine transient fuel for pilot command tracing in a proper response time. As shown at Figure 4, this fuzzy controller is a single input, single-output (SISO). The input is the error between the current engine rotational speed and the desired rotational speed (translated PLA to the engine rotational speed by using the thrust/rotational map of the engine) and the output is the transient fuel flow. For input and output, seven linguistic variables are determined with associated rules and membership functions as shown in Table 4 and Figures 5 and 6 (Delta N is PLA-RPM).

Table 4. Fuzzy rules for PLA control loop.

ΔN	NB	NM	NS	Z	PS	PM	PB
FMF (Transient Fuel Flow)	NB	NM	NS	Z	PS	PM	PB

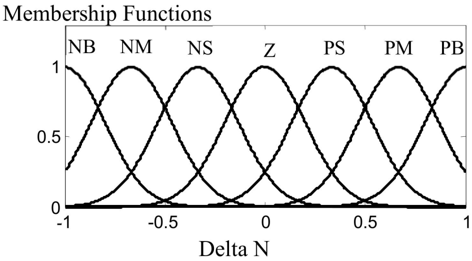


Figure 5. Input membership functions.

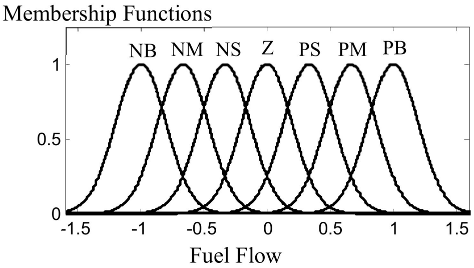


Figure 6. Output membership functions.

Fuzzy MALL and MDLL control loop design:

As instant fuel variation may cause aerodynamic instabilities like surge, stall, etc., MALL and MDLL control loops are to control acceleration/deceleration and to prevent the engine from exceeding its limits. For each loop, a SISO fuzzy controller is designed with three inputs and one output determined transient injected fuel. Rotor RPM error (Delta N), the difference between real acceleration with minimum (Ndot(min)-Ndot) and maximum (Ndot(max)-Ndot) allowed acceleration are three inputs as shown in Figures 7–9 and the transient fuel flow is the output.

The designed fuzzy rules are like the PLA loop as shown in Table 5. As PLA loop symmetry and uniformity between membership functions and rules have caused linear relation between inputs and output in the MADLL (Maximum acceleration and deceleration) loops. Again, FMF is transient fuel flow.

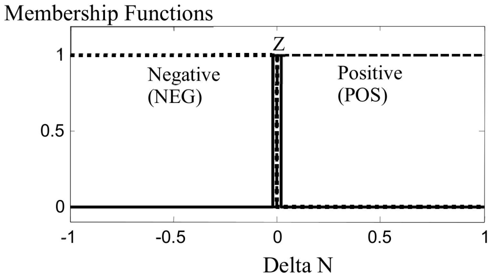


Figure 7. First input membership functions.

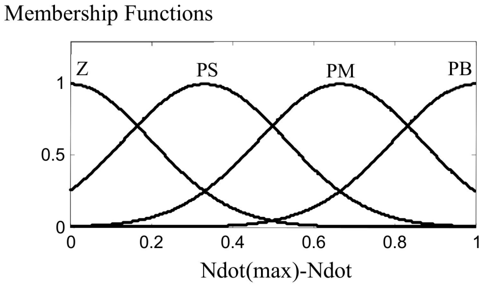


Figure 8. Second input membership functions.

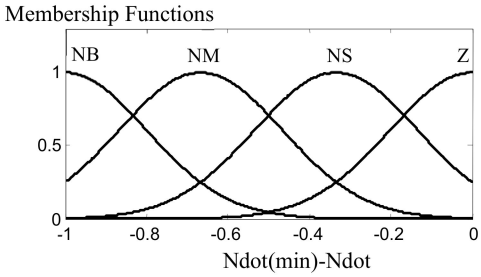


Figure 9. Third input membership functions.

Table 5. Fuzzy rules for MADLL loop.

ΔN = POS					**ΔN = NEG**				
$\dot{\mathbf{N}}_{\mathbf{max}} - \dot{\mathbf{N}}$	Z	PS	PM	PB	$\dot{\mathbf{N}}_{\mathbf{min}} - \dot{\mathbf{N}}$	Z	NS	NM	NB
FMF (Acc.)	Z	PS	PM	PB	**FMF (Dec.)**	Z	NS	NM	NB

It also should be mentioned that as MSLL loop is just to limit the maximum rotational speed of the GTE (and this value is constant for each engine), there is no need to manipulate it by fuzzy logic rules.

3. Hardware Implementation

After designing the PFC and MMFC, the implementation procedure will be described in this section in order to develop a HIL simulation platform to analyze the capabilities of the designed controllers in a real-time simulation feature. From implementation consideration point of view, selecting the proper hardware for real-world application is the main and the most important part of the controller manufacturing procedure. Chosen hardware must have an acceptable input reading speed, processing time, and resolution for outputs generation to be able to control the engine rapidly and correctly. For the GTE control problem, proper hardware must have minimum error and fault while reading engine rotational speed and PLA signals and producing the accurate fuel flow signal that should be injected into the engine to satisfy all engine control modes simultaneously.

Among different types of hardware, the AVR family has a lot of advantages including easy to program in C for most basic functions, adequate documentation, inexpensive, hobbyist friendly (parts in through-hole packages), nice peripherals (built-in oscillator, flash memory, onboard RAM, serial ports, ADC, EEPROM, etc.), low power consumption, and good cross-platform support. Therefore, the AVR Microcontroller (ATMEGA 32A) was chosen in this study. There is much evidence in the literature that confirms the use of ATMEGA family for controller hardware implementation because of the above-mentioned advantages [24–30]. The procedure of implementation is described in detail in this section:

3.1. Experimental Apparatus

In order to design proper hardware for GTE fuel controller with HIL test, well communicating between hardware and MATLAB Simulink is mandatory. For this purpose, the PC serial port was used, and thus, sending and receiving data between hardware and MATLAB Simulink environment have become achievable. The MAX232 is an integrated circuit that converts signals from a TIA-232 (RS-232) serial port to signals suitable for use in TTL-compatible digital logic circuits. The MAX232 is a dual transmitter/dual receiver that is typically used to convert the RX, TX, CTS, RTS signals. Any communication between PC and hardware needs this chip. However, some modern computers don't have a serial port. Therefore, a converter that can convert USB port to a serial port is also necessary. The used converter in this study is FT232 module. The FT232R is one of the latest devices to be added to FTDI's range of USB UART interface integrated circuit devices. The FT232R is a USB to serial UART interface with optional clock generator output.

The main processor unit is from AVR family (ATMEGA32A). Each data value will be received by the microcontroller and after calculating and output generating, the result will be returned to the software by this AVR. Figures 10 and 11 show the designed hardware schematic and its PCB (printed circuit board) separately. Components are generally soldered onto the PCB to both electrically connect and mechanically fasten them to it. This board is designed to simply achieve all requirements. It has one main processor (ATMEGA32), FT232 module to transferring data from PC to processor and vice versa, two capacitors, a resistor and regulator for power supply tuning, and a manual reset push button. After each process processor was automatically programmed for internal reset and it will be ready for another process.

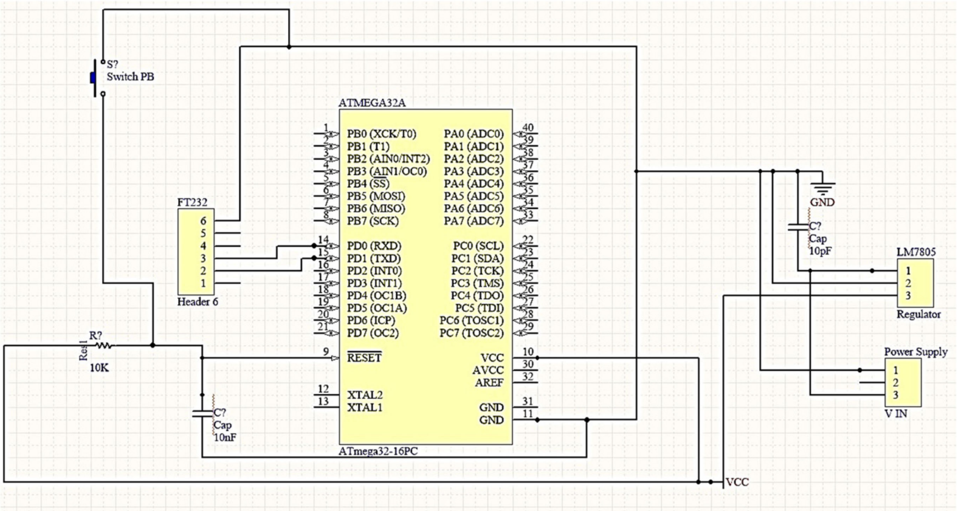


Figure 10. Designed board schematic.

For communicating between the abovementioned hardware and PC (Simulink environment), required codes have been written in CODEVISION environment. As mentioned earlier, after correct communicating between hard and software achievement, each controller was programmed at hardware and tested for proper programming and validating the hardware implementation by MATLAB Simulink. The ongoing picture shows the GTE controller hardware. Figure 12 shows the manufactured PCB in the real-world ready for control GTE. This PCB is the main GTE controller at our experiment and replaced with controller box in Simulink environment at hardware in the loop (HIL) test.

3.2. Initial Preparation

As shown in previous sections, fuzzy controllers take PLA and RPM as inputs and after calculation and execution of fuzzy rules, return the fuel flow signal as the output. This number is applied to the servo that controls engine fuel. The designed hardware can take PLA and RPM in the form of integer numbers (between 0–255) and output is a digital number between 0–255. An internal microcontroller clock was activated to counting transmission sensors pulses at a specified time for calculating the frequency and reading serial port for receiving PLA and engine rotational speed. The time-step is important because of calculating error derivative. The written codes for calculating time and receiving and transferring data are presented in the Appendix A in detail [Appendix A.1, Appendix A.2].

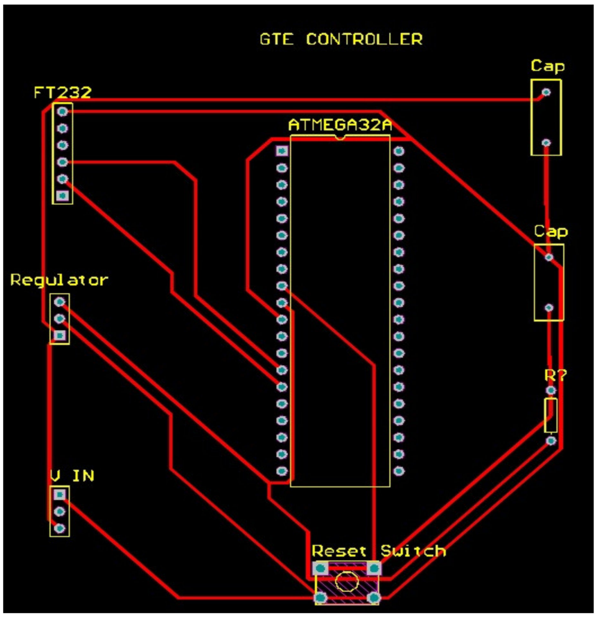


Figure 11. Designed board printed circuit board (PCB).

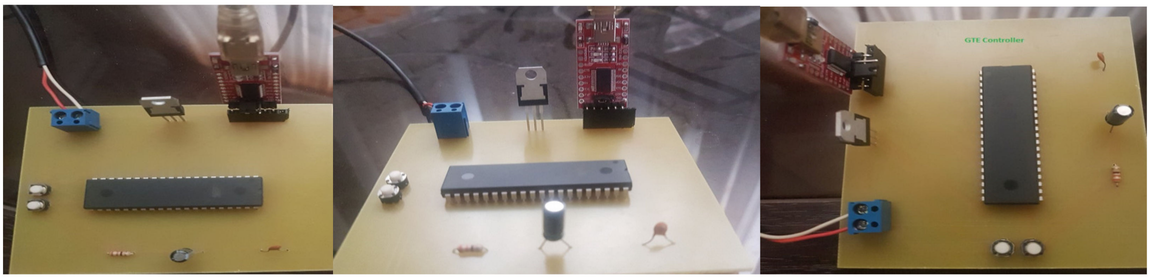


Figure 12. GTE controller.

3.2.1. PFC Controller Hardware Implementation

In order to implement the PFC, all seven Gaussian membership functions have been programmed in the Codevision environment. As an example, one of them is presented in the Appendix A [Appendix A.3]. It should be mentioned that the weighted average method is used for defuzzification. Computing output belonging value to membership functions was done with the center of the batch method as well [Appendix A.4]. After gaining transient output, this value must be added to the steady-state value read from table that is programmed at hardware and this summation must be changed to a digital number that will be sent by the microcontroller.

3.2.2. MMFC Hardware Implementation

Implementation of the fuzzy min–max controller has three separate steps. The first two steps are for a fuzzy controller and the final step is the MIN–MAX loops programming. The first fuzzy controller has one input and fifteen Gaussian membership functions (PLA control loop fuzzification). The second fuzzy controller has two inputs (error and its derivative) and four membership functions (acceleration/deceleration control loop fuzzification), [Appendix A.5, Appendix A.6].

3.3. Hardware in the Loop Simulation

Hardware in the loop simulation method has wide application in dynamic systems simulations. In this method, some parts of modeling will be done in the software environment and some others will be programmed and implemented physically on the hardware. The designed hardware must be acted synchronously with software to guarantee the consistency of the results. For a control system test, there are two different options:

- The controller could be modeled in the software to be running on target computer hardware while it is connected to your physical plant or system. (The target computer hardware acts as the controller.).
- The other option is to implement the controller on the hardware, which can include production or embedded controls implementation, using a simulation of your plant or system. (Here, the target computer acts as a physical plant or system.).

In recent years hardware in the loop method was used in many pieces of research for testing physical elements act accuracy. Some of these researches are performed based on controller implementation [31–34], and in some of the engine and sensors are real [35–37]. In this study, the controller is implemented as hardware in the real world and all other parts like engine and sensors were programmed at software.

Moreover, from a simulation speed point of view, the simulation could be run in the following features:

- Simulation without time limitations;
- Real-time simulation;
- Simulation faster than real-time.

Real-time simulation refers to a computer model of a physical system that can execute at the same rate as actual "wall clock" time. In other words, the computer model runs at the same rate as the actual physical system. Real-time simulation and testing extend beyond simulation by verifying algorithmic design behavior while running models at required speeds, respecting precise timing requirements. The executing model is connected to sensors, actuators, and other hardware.

4. Results Analysis

In order to validate the accuracy of hardware implementation, the result of HIL simulations was firstly compared with a model in the loop (MIL) run. A MIL simulation is a technique used to abstract the behavior of a system or sub-system in a way that this model can be used to test, simulate and verify that model. The implemented controllers were simulated in both the HIL platform and MIL with the engine model to investigate the effectiveness and performance of each controller. The engine model is a block-structured model created and validated against experimental results. All details about the engine modeling and validation procedure could be found in [18].

In order to simulate the dynamic behavior of the engine, the block-structured modeling approach has been used in this paper. These models consist of a linear dynamic part to simulate all engine lags and a nonlinear static part to simulate the relationship between the different engine parameters. The model parameters are usually tuned by the experimental results. There are three kinds of block-structure models, including Hammerstein, Wiener, and Wiener–Hammerstein models [38].

The novel generalized describing function (NGDF) is a recently proposed block-structured approach introduced by Lichtsinder et al. [38]. The NGDF is based on the error minimization concept and the difference between the NGDF model and the models proposed by other researchers is that in NGDF the transfer functions between different inputs and outputs have an incremental form to enhance the accuracy of the model. The schematic of the NGDF model is shown in Figure 13 to show that this method has the highest accuracy between the different block structures modelling approach, the jet engine is modelled using different modeling approaches, transfer function described in [39], Wiener block structure described in [40], and NGDF described in [38]. The engine specification is shown in Figure 14. The results are compared with the experimental results of the transient behavior of the engine in Figure 15. As shown in Figure 15, the NGDF is tracking the engine parameters with very high accuracy in both steady-state and transient operation. More details about the engine modelling, procedure, and used equations could be found in [41,42].

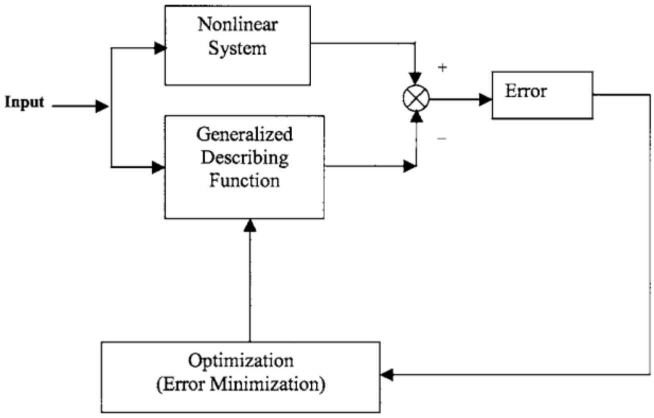
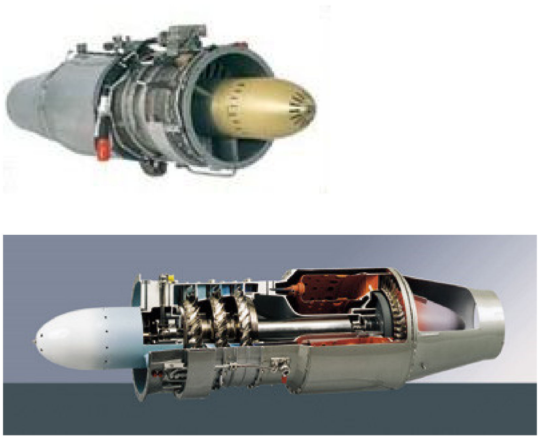

Input
Nonlinear System
Generalized Describing Function
+
−
Error
Optimization (Error Minimization)

Figure 13. The schematic of novel generalized describing function (NGDF) for modeling jet engines [38]



General characteristics

- Type: Single Spool Turbojet
- Length: 851 mm
- Diameter: 348 mm
- Dry weight: 61.2 kg

Components

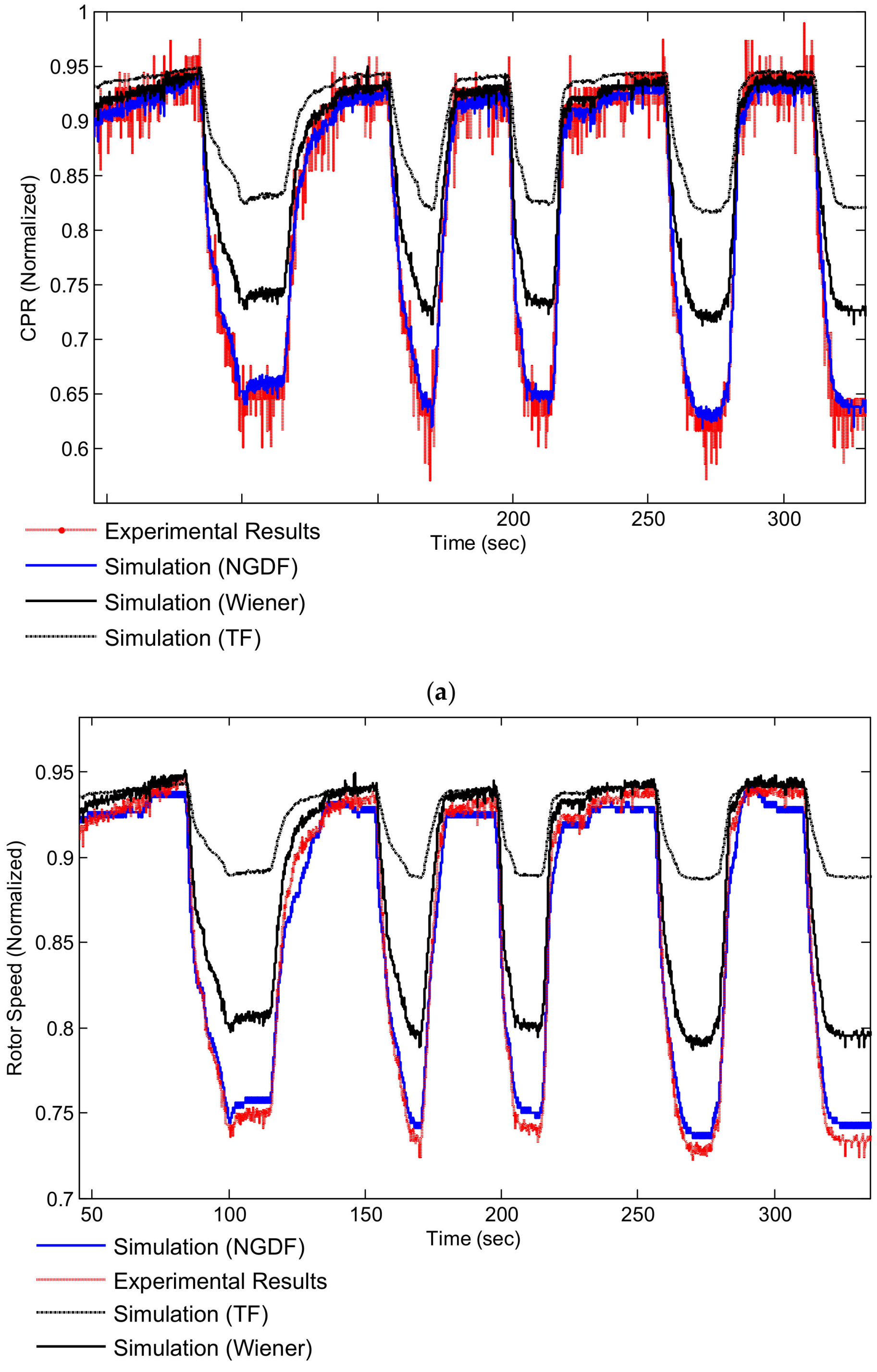
- Compressor: 4 stage axial
- Combustors: Annular
- Turbine: Single Stage

Performance

- Maximum thrust: 5.33 kN
- Overall pressure ratio: 6.3:1
- Air mass flow: 8.14 kg/s
- Specific fuel consumption: 1.1 kg/(daN h)
- Thrust-to-weight ratio: 8.9:1

Figure 14. The schematic and characteristics of the modelled turbojet engine (images from [43]).

In order to carry on simulations, the PLA command has been varied with step changes as a function of time in order to test the capability of controllers in dealing with sudden changes and difficult working conditions. Figures 16 and 17 compare the MIL and HIL results for the PFC and Figures 18 and 19 compare HIL and MIL results for MMFC. As can be seen in these figures, both controllers were implemented accurately and replicate the simulation situations without any steady-state or transient errors.



(**a**)

(**b**)

Figure 15. Modelling a turbojet engine with different reduced-order approaches: (**a**) normalized compressor pressure ratio (CPR) tracking; (**b**) normalized rotor speed tracking.

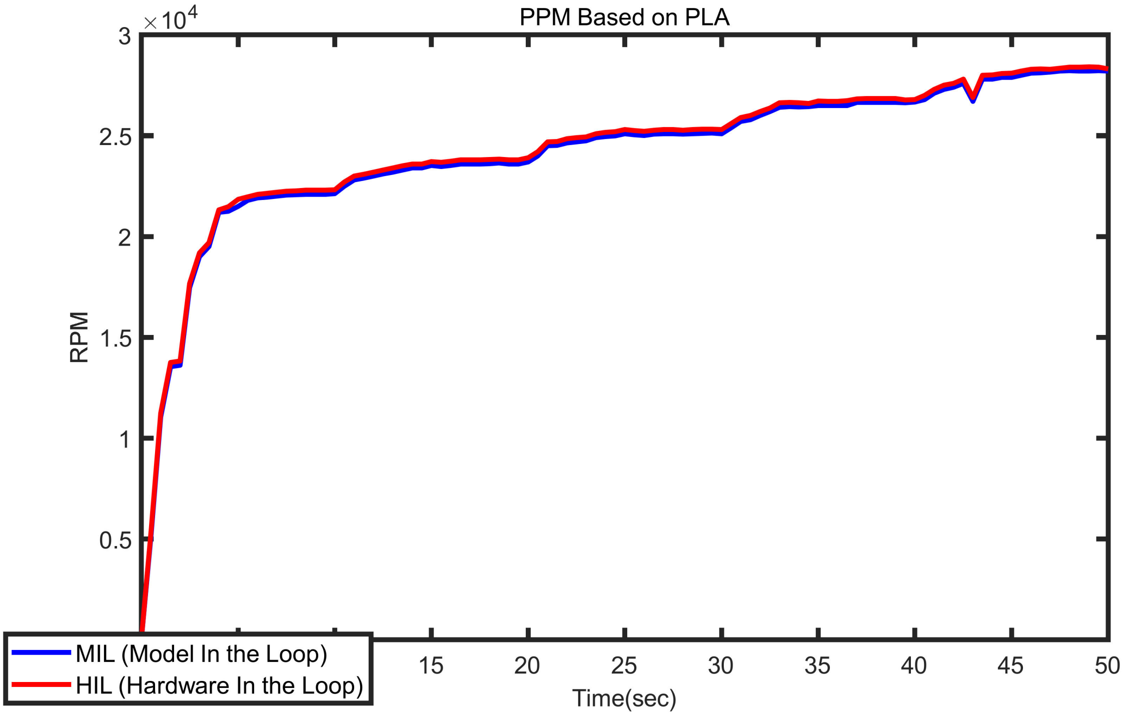


Figure 16. Model in the loop (MIL) and hardware in the loop (HIL) shaft rotational speed (RPM) signal comparison for PFC.

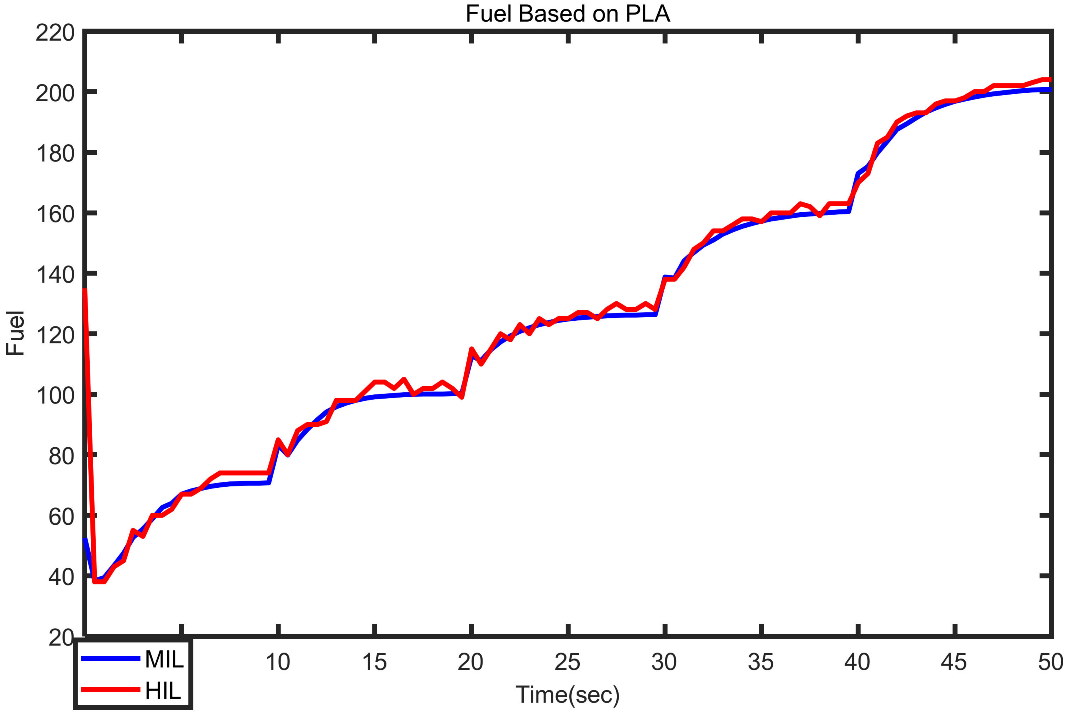


Figure 17. MIL and HIL fuel signal comparison for PFC.

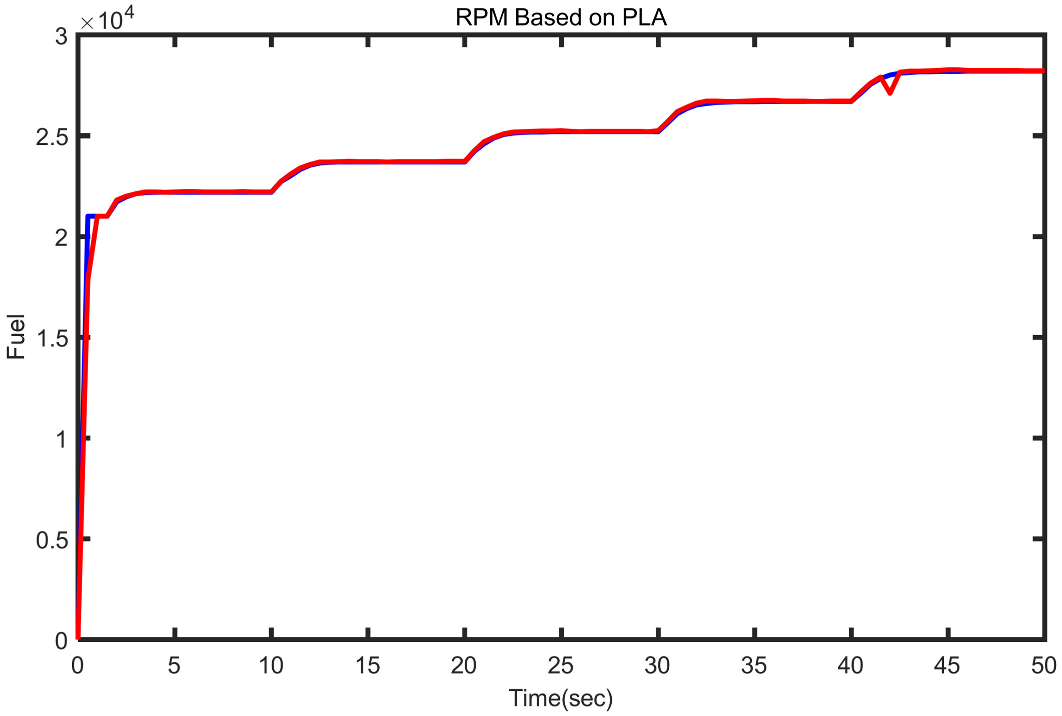


Figure 18. MIL and HIL RPM signal comparison for MMFC.

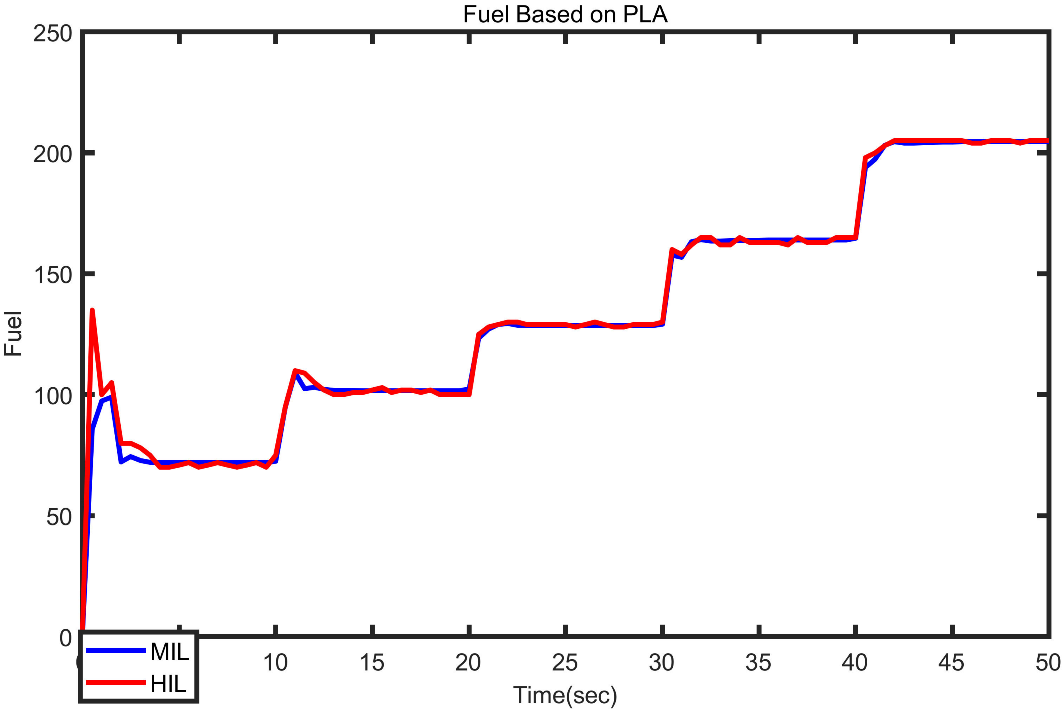


Figure 19. MIL and HIL Fuel signal comparison for MMFC.

Moreover, the industrial min–max controller has also been implemented on the hardware in order to explore the effectiveness of the designed fuzzy controllers in satisfying engine control modes. Figures 20 and 21 confirm the accuracy of the implemented min–max controller.

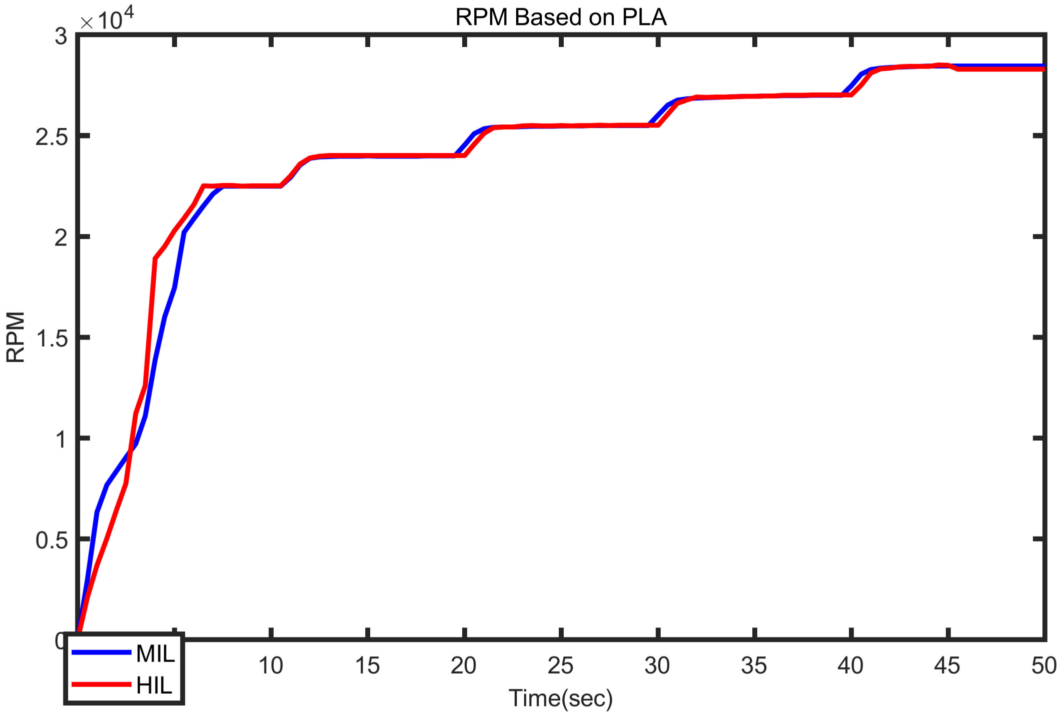


Figure 20. MIL and HIL RPM signal comparison for MIN–MAX controller.

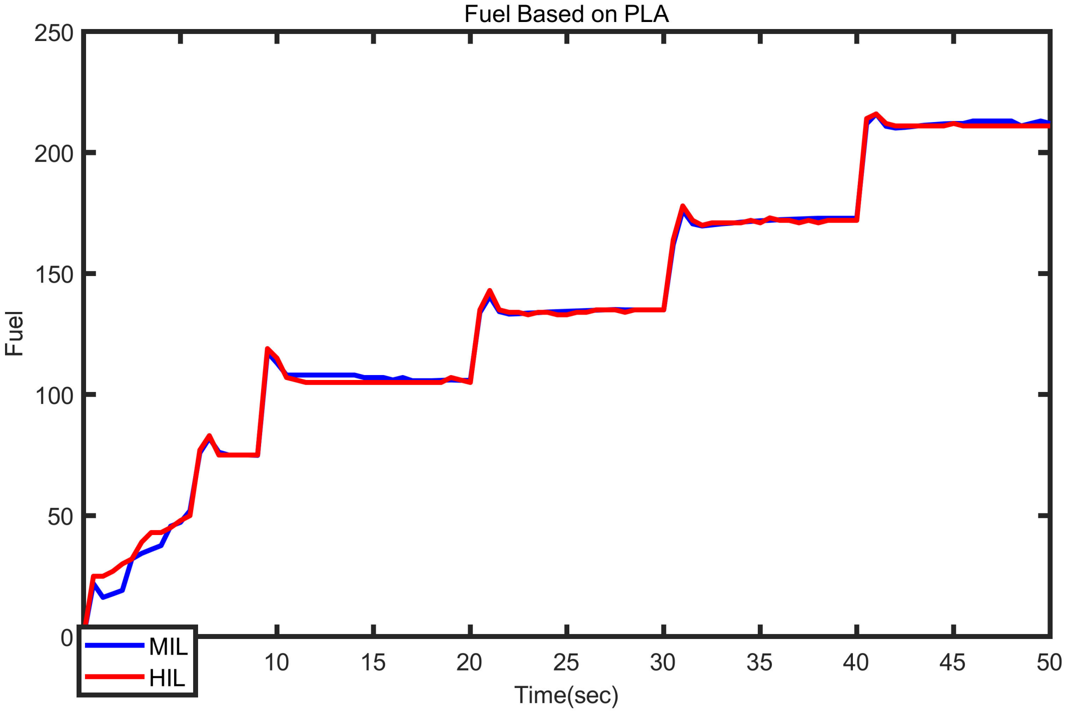


Figure 21. MIL and HIL Fuel signal comparison for MIN–MAX controller.

After confirming the validity of the implementation procedure, the dynamic behavior of the three controllers is compared. Figures 22 and 23 compare the results of HIL simulations for PFC and MMFC and also the min–max Controller. These figures confirm that both fuzzy controllers are able to satisfy all engine control modes simultaneously without exceeding engine physical limitations. Figure 22 shows that the PFC has a smaller response time than the MMFC. It means that this controller enables the engine for better maneuverability which is an important aspect for military and unmanned aerial vehicles (UAV) applications. Both controllers' behavior is in very good agreement with the industrial min–max controller and this confirms the feasibility of the designed controllers for real-world applications. On the other hand, Figure 23 shows that the MMFC has less fuel consumption in comparison with the PFC. It introduces this structure as a high potential candidate for applications where the fuel burn and specific fuel consumption (SFC) is more important (e.g., civil aircraft engines). Both fuzzy controllers perform better than the conventional min–max controller in terms of fuel economy. Table 6 compares the response time and fuel consumption of the controllers in a one-minute simulation shown in Figures 22 and 23.

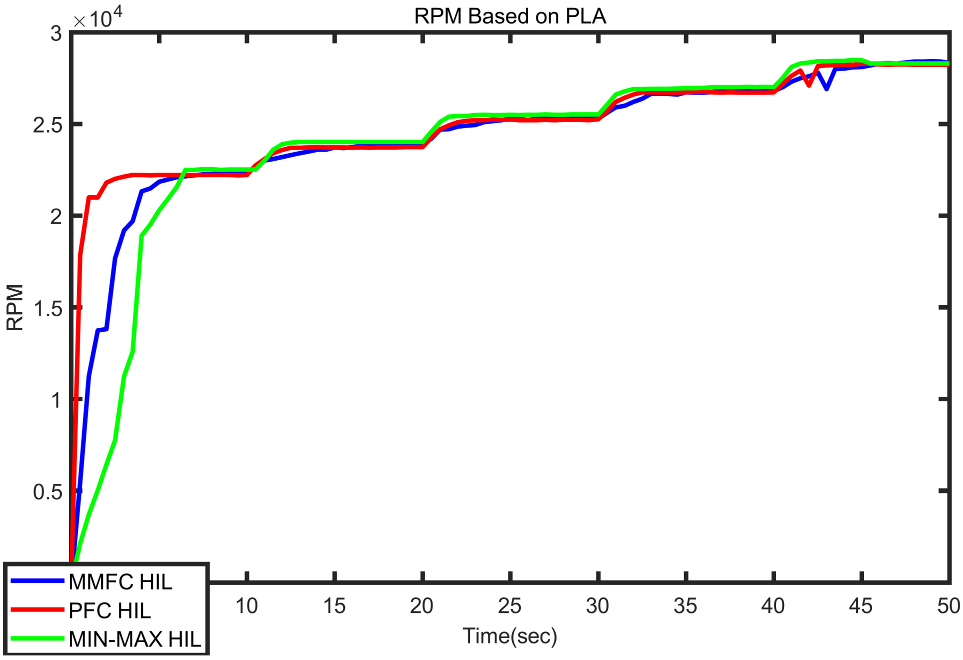


Figure 22. Command tracing comparison between PFC, MMFC and MIN–MAX controller in HIL test.

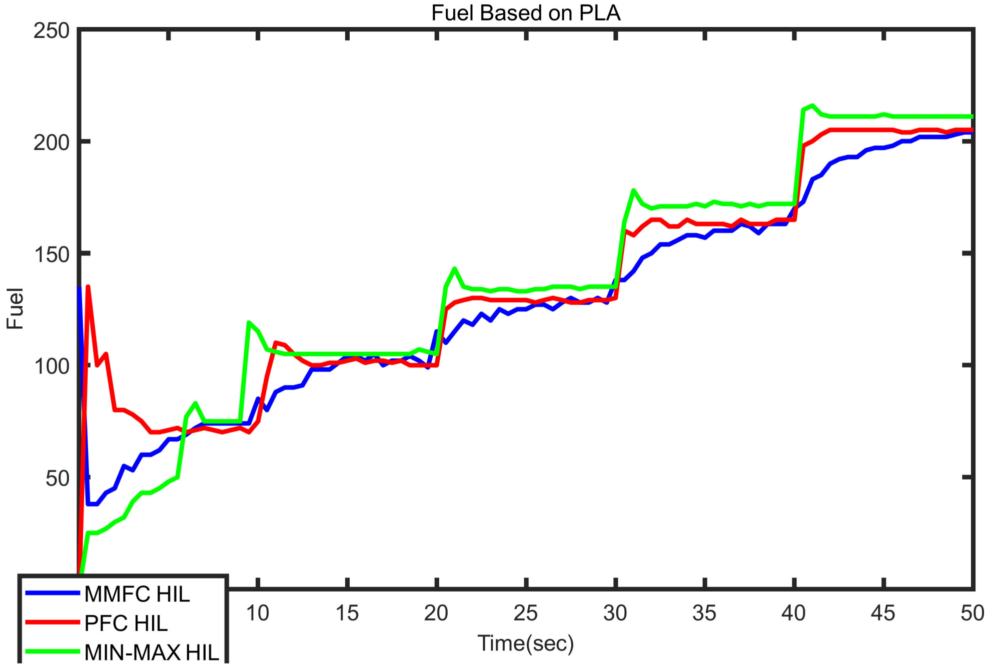


Figure 23. Fuel consumption comparison between PFC, MMFC and the MIN–MAX controller in the HIL test.

Table 6. Comparison between response time and fuel consumption for Pure Fuzzy Controller (PFC), Fuzzy Min-Max (FMM) controller, ans Min-Max controller.

	Mean Response Time (s)	**Fuel Consumption (kg/s)**
Pure Fuzzy Controller	4.01	7.05
Min–Max Fuzzy Controller	4.18	6.76
Min–Max Controller	3.95	7.21

5. Conclusions

Different fuzzy controller structures for gas turbine aero-engines are investigated in this paper. Based on an analytic review, it is shown that pure fuzzy controllers and min–max fuzzy controllers are the two main proposed architectures for the next generation of aero-engines control system design. Both architectures are designed described in detail with associated fuzzy rules and membership functions. The hardware in the loop platform is also developed and the validity of the implemented controllers is confirmed by a model in the loop approach. The simulation results for a step-change mission confirm that:

- The pure fuzzy controller structure performs better in terms of pilot command tracking and, therefore, it is an appropriate candidate for control of the next generation of military aero-engines.
- The min–max Fuzzy controller structure performs better from fuel consumption and economic points of view that makes it a strong candidate for the next generation of civil aero-engines.
- Both fuzzy controller structures are feasible for real-world application and perform better than the conventional min–max controller in terms of fuel economy.

The conclusion of this paper could be considered as the first step for defining the road map for the next generation of aero-engine advanced controller design. Next steps could focus on reliability and stability criteria of fuzzy controllers for real-world applications.

Author Contributions: Conceptualization, S.J.; Data curation, S.J.M.D.F.; Formal analysis, S.J.M.D.F.; Investigation, S.J. and S.J.M.D.F.; Methodology, S.J.M.D.F.; Project administration, S.J.; Resources, S.J.M.D.F.; Software, S.J. and S.J.M.D.F.; Supervision, S.J.; Writing—original draft, S.J.M.D.F.; Writing—review & editing, S.J.M.D.F. and S.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Appendix A

Appendix A.1. Activating Codes for Calculating Time (Each Interrupt Overflow Led to Read Serial Port)

```
interrupt [TIM0_COMP] void timer0_comp_isr(void)
{
if (cont <= 9) {
cont = cont + 1;
TCNT0 = 0;
}
else if(cont == 10) {
cont = 0;
TIMSK = 0x00;
ADMUX = 0x00;
ADCSRA = 0xCB;
TCNT0 = 0;
};
}
```

Appendix A.2. These Codes Activate Serial Port for Receive and Send Data to PC

```
interrupt [USART_RXC] void usart_rx_isr(void)
{
int data;
data = UDR;
rx_buffer[7] = rx_buffer[6];
rx_buffer[6] = rx_buffer[5];
rx_buffer[5] = rx_buffer[4];
rx_buffer[4] = rx_buffer[3];
rx_buffer[3] = rx_buffer[2];
rx_buffer[2] = rx_buffer[1];
rx_buffer[1] = rx_buffer[0];
rx_buffer[0] = data;
}
```

Appendix A.3. Gaussian Input Membership Function

```
sig = 0.2233;
c = −1;
ff = pow ((e-c),2);
fff = pow (sig,2);
n1 = pow (k, (−ff/(2*fff)));
```

Appendix A.4. Defuzzification—-Weighted Average Method

V = ((do1 + do2 + do3 + do4 + do5 + do6 + do7 + do8 + do9 + do10 + do11 + do12 + do13 + do14 + do15 + do16 + do17 + do18 + do19 + do20 + do21 + do22 + do23 + do24 + do25 + do26 + do27 + do28 + do29 + do30 + do31 + do32 + do33 + do34 + do35 + do36 + do37 + do38 + do39 + do40 + do41 + do42 + do43 + do44 + do45 + do46 + do47 + do48 + do49)/(o1 + o2 + o3 + o4 + o5 + o6 + o7 + o8 + o9 + o10 + o11 + o12 + o13 + o14 + o15 + o16 + o17 + o18 + o19 + o20 + o21 + o22 + o23 + o24 + o25 + o26 + o27 + o28 + o29 + o30 + o31 + o32 + o33 + o34 + o35 + o36 + o37 + o38 + o39 + o40 + o41 + o42 + o43 + o44 + o45 + o46 + o47 + o48 + o49));

A.5. Sample PLA Membership Function

```
sig = 0.0553;
c = −1;
ff = pow ((e-c),2);
fff = pow (sig,2);
a1 = pow (k, (−ff/(2*fff)));
```

A.6. Input Membership Functions for Accdcc

```
// FIrst input
if (e < 0){
c1 = 1;
c2 = 0;
c3 = 0;
}
if (e > 0){
c1 = 0;
c2 = 0;
c3 = 1;
}
if (e = 0){
c1 = 0;
c2 = 1;
c3 = 0;
}
```

```
//second Input
matfu = 0.049;
ndot1 = (matfu1-edot) * 18;
if (ndot1 < −1){ ndot1 = −1;
}
if (ndot1 > 1){ ndot1 = 1;
}
sig = 0.08625;
c = 0;
ff = pow ((ndot1-c),2);
fff = pow (sig,2);
d1 = pow (k, (−ff/(2*fff)));
sig = 0.1995;
c = 0.213;
ff = pow ((ndot1-c),2);
fff = pow (sig,2);
d2 = pow (k, (−ff/(2*fff)));
sig = 0.1511;
c = 0.4345;
ff = pow ((ndot1-c),2);
fff = pow (sig,2);
d3 = pow (k, (−ff/(2*fff)));
sig = 0.2808;
c = 1;
ff = pow ((ndot1-c),2);
fff = pow (sig,2);
d4 = pow (k, (−ff/(2*fff)));
//Third Input
matfu2 = (−0.05);
ndot2 = (matfu2-edot) * 19.7879;
if (ndot2 < −1){ ndot2 = −1;
}
if (ndot2 > 1){ ndot2 = 1;
}
sig = 0.2623;
c = −1;
ff = pow ((ndot2-c),2);
fff = pow (sig,2);
g1 = pow (k, (−ff/(2*fff)));
sig = 0.2285;
c = −0.5705;
ff = pow ((ndot2-c),2);
fff = pow (sig,2);
g2 = pow (k, (−ff/(2*fff)));
sig = 0.3;
c = -0.3065;
ff = pow ((ndot2-c),2);
fff = pow (sig,2);
g3 = pow (k, (−ff/(2*fff)));
sig = 0.1252;
c = 0;
ff = pow ((ndot2-c),2);
fff = pow (sig,2);
g4 = pow (k, (−ff/(2*fff)))
```

References

1. Lutambo, J.; Wang, J.; Yue, H.; Dimirovsky, G. Aircraft turbine engine control systems development: Historical Perspective. In Proceedings of the 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015.
2. Jaw, L.C.; Mattingly, J.D. *Aircraft Engine Controls*; American Institute of Aeronautics and Astronautics: New York, NY, USA, 2009; ISBN 978-1-60086-705-7.
3. Moir, I.; Seabridge, A. *Engine Control Systems*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2008.
4. Michels, K.; Klawonn, F.; Kruse, R.; Nürnberger, A. *Fuzzy Control*; Springer: Berlin/Heidelberg, Germany, 2006.
5. Silva, J.F.; Pinto, S.F. *Power Electronics Handbook*, 4th ed.; CRC Press: Boca Raton, FL, USA, 2018; pp. 1141–1220.
6. Ghodekar, M.; Jadhav, S.P.; Jadhav, S.R. Design and simulation of Fuzzy and Fuzzy-PI controller for speed of Gas Turbine. In Proceedings of the International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), Pune, India, 9–10 September 2016.
7. Mosaferin, R.; Fakharian, A. Fuzzy control for rotor speed of power plant gas turbine. In Proceedings of the 13th Iranian Conference on Fuzzy Systems (IFSC), Qazvin, Iran, 27–29 August 2013.
8. Karande, A.S.; Nigam, M.J.; Kadam, V. Design of a Fuzzy PID controller for a gas turbine power plant. In Proceedings of the Communication, Control and Intelligent Systems (CCIS), Mathura, India, 7–8 November 2015.
9. Shete, S.A.; Jape, V.S. Design of a fuzzy modified model reference adaptive controller for a gas turbine rotor speed control using T-S fuzzy mechanism. In Proceedings of the International Conference on Technological Advancements in Power and Energy (TAP Energy), Kollam, India, 21–23 December 2017.
10. Haijun, L.; Kai, Q. Design of master controller of 100kW micro gas turbine generator sets based on fuzzy-PID algorithm. In Proceedings of the 6th International Forum on Strategic Technology, Harbin, China, 22–24 August 2011.
11. Bica, B.; Akat, G.; Chipperfield, A.J.; Fleming, P.J. Multiobjective design of a fuzzy controller for a gas turbine aero-engine. In Proceedings of the UKACC International Conference on Control 98 (Conf. Publ. No. 455), Swansea, UK, 1–4 September 1998.
12. Balakrishnan, S.R.; Mishra, S.K.; Sundararajan, V.; Damodaran, K.A. Fuzzy Computing for Control of Aero Gas Turbine Engines. *Def. Sci. J.* **1994**, *44*, 295–304. [CrossRef]
13. Chipperfield, A.J.; Bica, B.; Fleming, P.J. Fuzzy scheduling control of a gas turbine aero-engine: A multiobjective approach. *IEEE Trans. Ind. Electron.* **2002**, *49*, 536–548. [CrossRef]
14. Martucci, A.; Volponi, A.J. Fuzzy Fuel Flow Selection Logic for a Real Time Embedded Full Authority Digital Engine Control. *ASME J. Eng. Gas Turbine Power* **2003**, *125*, 909–917. [CrossRef]
15. Richter, H. *Advanced Control of Turbofan Engines*; Springer: New York, NY, USA, 2012; ISBN 978-1-4614-1171-0. [CrossRef]
16. Hadroug, N.; Hafaifa, A.; Guemana, M.; Kouzou, A.; Salam, A.; Chaibet, A. Heavy duty gas turbine monitoring based on adaptive neuro-fuzzy inference system: Speed and exhaust temperature control. *Math. Ind. Case Stud.* **2017**, *8*, 8. [CrossRef]
17. Oglah, A.A.; Mohammed, A.J. Design of an Interval Fuzzy Type-2- PID Controller for a Gas Turbine Power Plant. *Am. Sci. Res. J. Eng. Technol. Sci. (ASRJETS)* **2018**, *44*, 155–169.
18. Jafari, S.; Nikolaidis, T. Turbojet engine industrial min-max controller performance improvement using fuzzy norms. *Electronics* **2018**, *7*, 314. [CrossRef]
19. Guolian, H.; Linjuan, G.; Congzhi, H.; Jianhua, Z. Fuzzy modeling and fast model predictive control of gas turbine system. *Energy* **2020**, *200*, 117465.
20. Available online: http://www.aviationchief.com/turbine-engines.html (accessed on 13 February 2021).
21. Bazazzadeh, M.; Badihi, H.; Shahriari, A. Gas Turbine Engine Control Design Using Fuzzy Logic and Neural Networks. *Int. J. Aerosp. Eng.* **2011**, *2011*, 156796. [CrossRef]
22. Spang, H.A., III; Brown, H. Control of jet engines. *Control Eng. Pract.* **1999**, *7*, 1043–1059. [CrossRef]
23. Kreiner, A.; Lietzau, K. *The Use of Onboard Real-Time Models for Jet Engine Control*; MTU Aero Engines: Munich, Germany, 2002.
24. Monika, S.; Anuradha, T. Design and Implementation of Traffic Density Controller using Wireless Communications. *Int. J. Innov. Technol. Explor. Eng. (IJITEE)* **2019**, *8*.
25. Altaf, K.; Akhtar, A.; Rehman, S.-U.; Iqbal, J. Design, implementation and real-time digital control of a cart-mounted inverted pendulum using Atmel AVR Microcontroller. In Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation, Corfu Island, Greece, 16–19 February 2007.
26. Wang, W.; Xia, L. Design and implementation of a control system using AVR microcontroller. In Proceedings of the 2009 International Conference on Image Analysis and Signal Processing, Linhai, China, 11–12 April 2009.
27. Nhivekar, G.S.; Nirmale, S.S.; Mudholker, R.R. Implementation of fuzzy logic control algorithm in embedded microcomputers for dedicated application. *Int. J. Eng. Sci. Technol.* **2011**, *3*, 276–283. [CrossRef]
28. Banerji, S. Design and Implementation of an Unmanned Vehicle using a GSM Network with Microcontrollers. *Int. J. Sci. Eng. Technol. Res.* **2013**, *2*.
29. Gupta, R. Body Control Unit Using the Implementation of AVR Microcontroller. *Int. J. Eng. Res. Technol. (IJERT)* **2013**, *2*.
30. Shaikh Yusuf, H.; Khan, A.R.; Behere, S.H. AVR Microcontroller Based Data Acquisition System for Laboratory Experiments. *Adv. Appl. Sci. Res.* **2012**, *3*, 208–215.
31. Duman, E.; Can, H.; Akin, E. Evaluating of a Fuzzy Chip by Hardware-in-the-Loop (HIL) Simulation. *Int. Rev. Comput. Softw.* **2008**, *3*.

32. Valencia, A.; Linares, N.; Amaya, D. Control system using HIL, PID and Fuzzy Logic with Rapid Prototyping. *Int. J. Appl. Eng. Res.* **2016**, *11*, 5480–5487.
33. Akbatı, O.; Üzgün, H.D.; Akkaya, S. Hardware-in-the-loop simulation and implementation of a fuzzy logic controller with FPGA: Case study of a magnetic levitation system. *Trans. Inst. Meas. Control.* **2019**, *41*, 2150–2159. [CrossRef]
34. Amal, S.; Vishnu, V.; Chacko, R.V.; Ghugal, S.; Mengaji, P.; Karle, U. HIL simulation and controller prototyping of EV/HEV systems using multi-core xEV Real time Simulator. In Proceedings of the 2017 IEEE Transportation Electrification Conference (ITEC-India), Pune, India, 13–15 December 2017.
35. Chu, L.; Chao, L. Hardware-in-the-loop Simulation of Traction Control Algorithm Based on Fuzzy PID. *Energy Procedia* **2012**, *16*, 1685–1692. [CrossRef]
36. Kumar, R.S.; Ganapathy, V. Hardware in the Loop Simulation and Implementation of Fuzzy Sliding Mode Control of Induction Motor Based on FPGA. *Int. J. Innov. Comput. Inf. Control ICIC* **2015**, *11*, 2215–2228.
37. Rasaienia, A. Hardware in the Loop (HIL) Analysis of Fuzzy Controller for Ball and Beam System. *Majlesi J. Mechatron. Syst.* **2016**, *5*, 29–34.
38. Lichtsinder, M.; Levy, Y. Jet engine model for control and real-time simulation. *J. Eng. Gas Turbine Power* **2006**, *128*, 745–753. [CrossRef]
39. Saravanamuttoo, H.I.H.; MacIsaac, M.S. An overview of engine dynamic response and mathematical modelling concepts. In *Engine Handling, Proceedings of the 60th Symposium of the AGARD Propulsion and Energetics Panel, Golden Coast Motel, Marathon, Attika, Greece, 11–14 October 1982*; AGARD: Neuilly-sur-Seine, France, 1983.
40. Kulikov, G.G.; Thompson, H.A. *Dynamic Modeling of Gas Turbines*; Industrial Control Center: Glasgow, UK, 2003.
41. Wills, A.; Ninness, B. Generalised Hammerstein-Wiener system estimation and a benchmark application. *Control Eng. Pract.* **2012**, *20*, 1097–1108. [CrossRef]
42. Paduart, J.; Lauwers, L.; Pintelon, R.; Schoukens, J. Identification of a Wiener-Hammerstein system using the polynomial nonlinear state space approach. *Control Eng. Pract.* **2012**, *20*, 1133–1139. [CrossRef]
43. Available online: http://www.leteckemotory.cz/motory/microturbo (accessed on 13 February 2021).