

Design of a Turbojet for UAV Applications

A B.Tech Project Report

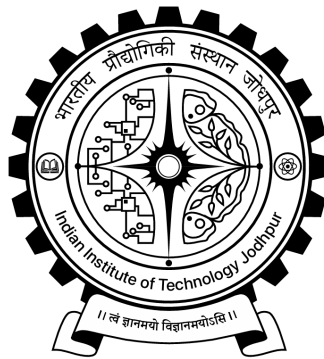
Submitted by

Aniksha Mahala (B21ME008)

Khushi Katara (B21ME032)

Under the Supervision of

Dr. Sudipto Mukhopadhyay



Department of Mechanical Engineering

Indian Institute of Technology Jodhpur

November, 2024

Contents

List of Figures	2
Acknowledgements	3
Abstract	4
1 Introduction	5
1.1 Motivation	5
1.2 Problem Statement	5
1.3 Objective	5
1.4 Working Of turbojet engine	6
1.4.1 Compressor	6
1.4.2 Combustor	7
1.4.3 Turbine	7
2 Work Done	9
2.1 Mathematical Modeling of the Turbojet Engine	9
2.1.1 Compressor Modeling	9
2.1.2 Combustion Chamber Modeling	10
2.1.3 Turbine Modeling	11
2.2 Work Balance and Thrust Calculation	12
2.3 PI Controller Design	13
2.4 Implementation of Fuzzy Logic Control	15
2.5 Design of Simscape Model of Engine	16
3 Results and Discussion	20
3.1 MATLAB Code Results	21
3.2 Simscape Model Results	22
4 Conclusion	25
A Appendix	26
A.1 MATLAB Codes	26
References	31

List of Figures

Figure 2.1: Compressor Map for Beta Line Method

Figure 2.2: Turbine Map

Figure 2.3: PI Controller

Figure 2.4: Fuzzy Logic Controller

Figure 2.5: Fuzzy Logic Control System

Figure 2.6: Turbojet Simscape Model

Figure 2.7: Dual Controller Block

Figure 2.8: Compressor Block Parameters

Figure 2.9: Turbine Block Parameters

Figure 3.1: PI controller Results

Figure 3.2: Fuzzy Logic controller Results

Figure 3.3: Compressor Map

Figure 3.4: Properties of air at compressor outlet

Acknowledgements

We would like to express our heartfelt gratitude to our supervisor, Dr. Sudipto Mukhopadhyay, for his invaluable guidance, unwavering support, and dedication throughout the semester. His insights and advice played an essential role in the successful completion of our project, and we feel fortunate to have benefited from his expertise.

Dr. Mukhopadhyay's constructive feedback and thoughtful suggestions consistently encouraged us to refine our ideas and overcome challenges, enabling us to achieve our project goals with confidence and clarity. We are deeply appreciative of the time and effort he invested in supporting us at every stage of this journey.

We also affirm that this project has been completed solely by us, Aniksha and Khushi, and represents our own work, reflecting the knowledge and skills we have gained under Dr. Mukhopadhyay's mentorship.

With warm regards and sincere thanks,
Aniksha & Khushi

Abstract

This project focuses on the design and evaluation of a virtual turbojet engine tailored for UAV applications, implemented in MATLAB. Aimed at optimizing UAV propulsion systems, the study explores two control strategies: a conventional PI controller and a Fuzzy Logic controller. The virtual model allows for simulating key engine parameters and control responses without the need for costly physical prototypes. Results indicate that the Fuzzy Logic controller outperforms the PI controller, providing smoother thrust control and enhanced stability under non-linear conditions. The mathematical models and simulations lay a foundation for improved UAV propulsion, emphasizing adaptable control strategies for efficiency and performance.

Introduction

1.1 Motivation

The motivation behind developing a virtual turbojet engine model for UAV applications is rooted in the rapid expansion of unmanned aerial vehicles (UAV), that are being used across multiple sectors. UAVs are gaining popularity in many fields for example: in the defense sector, disaster relief, environmental monitoring, and logistics. These operations need UAVs with high flight performance, exact control, and the capability to adjust to different situations, all depending on the propulsion system that is good in performance and reliability.

This project seeks to connect the gap by utilizing a virtual model of a turbojet engine in MATLAB, which entails the simulation of engine parameters and control strategies over the physical prototypes, which are very expensive. Key to this effort is the design and evaluation of control systems, specifically comparing PID and Fuzzy Logic Control. Each approach has unique strengths: PID control is generally preferred for its simplicity and flexibility, while Fuzzy Logic Control can deal with uncertainties and offer smoother changes, necessary in complex environments where UAVs fly in quickly changing conditions.

1.2 Problem Statement

This project addresses the challenge of designing a control mechanism that is helpful to generate the desired thrust by controlling the fuel mass flow rate, at different flight levels. As we all know external parameters are not always the same in the case of aviation. pressure and temperature changes at different flight levels. So considering this, there should be a control mechanism that can control the thrust produced by the engine according to the need. In this project, we had to design a control mechanism for the same and also design the virtual turbojet model to check the pressure and temperature at the inlet and outlet of each part of the component (compressor, combustor, and turbine) of the engine. With these parameters, we can calculate the thrust produced.

1.3 Objective

We have selected a small turbojet model for our project named AMT Olympus HP which has specifications as follows: Centrifugal Compressor, Annular Type Combustion Chamber, Single Spool Turbine.

The study seeks to assess and contrast PID and Fuzzy Logic Control systems to identify the most efficacious method for attaining stability and responsiveness in engine control. A MATLAB-based virtual turbojet engine has been designed specifically for UAV applications, enabling the simulation of essential performance characteristics. The project aims to optimize thrust by employing a control mechanism to adjust the fuel mass flow rate, with the objective of maximizing thrust while improving fuel efficiency and overall engine performance. These aims will guarantee enhanced engine performance in UAV systems.

1.4 Working Of turbojet engine

Large amounts of surrounding air are continuously brought into the engine inlet. We have shown here a tube-shaped inlet, like one you would see on an airliner. But inlets come in many shapes and sizes depending on the aircraft's mission. At the rear of the inlet, the air enters the compressor. The compressor acts like many rows of airfoils, with each row producing a small jump in pressure. A compressor is like an electric fan. We have to supply energy to turn the compressor. At the exit of the compressor, the air is at a much higher pressure than a free stream. In the burner, a small amount of fuel is combined with the air and ignited. (In a typical jet engine, 100 pounds of air/sec is combined with only 2 pounds of fuel/sec. Most of the hot exhaust has come from the surrounding air).

Leaving the burner, the hot exhaust is passed through the turbine. The turbine works like a windmill. Instead of needing the energy to turn the blades to make the airflow, the turbine extracts energy from a flow of gas by making the blades spin in the flow. In a jet engine, we use the energy extracted by the turbine to turn the compressor by linking the compressor and the turbine by the central shaft. The turbine takes some energy out of the hot exhaust, but there is enough energy left over to provide thrust to the jet engine by increasing the velocity through the nozzle. Because the exit velocity is greater than the free stream velocity, thrust is created as described by the thrust equation. For a jet engine, the exit mass flow is nearly equal to the free stream mass flow, since very little fuel is added to the stream.

The thrust equation for a turbojet is given by:

$$F = \dot{m} \cdot V_e - \dot{m} \cdot V_0$$

1.4.1 Compressor

Jet engines take in a great amount of surrounding air through their inlet. Behind the inlet is the compressor section. The compressor section, as you may have already guessed, compresses this incoming air. The compressor's job is to increase the pressure of the incoming air before it reaches the combustion section.

There are two main types of compressors: axial and centrifugal. Many modern turbojet and turbofan engines have axial compressors because of their performance efficiency. Centrifugal compressors can increase the total pressure of the air by a factor of 4, and axial-flow compressors can only increase the pressure by a factor of 1.2. So, how do axial compressors become more efficient? They're made to be multi-staged, therefore, the pressure increase is multiplied row by row. Axial compressors have an advantage over

centrifugal compressors because of their ability to have multiple stages. These advantages are ideal for an application where the thrust of the engine itself is the driving force of the aircraft or in our case the driving force of the vehicle.

In an axial-flow compressor, the flow enters in an axial direction, which means it is parallel to the axis of rotation. This compressor first compresses the incoming fluid by accelerating it and then diffusing it, which creates an increase in pressure.

A fluid is anything that flows, so do not be confused with the term “fluid” in reference to air. Although liquids are most commonly known as fluids, anything that is loosely held together by gas particles is in fact a fluid. Since air is a gas, it does flow and it will take the shape of its container

1.4.2 Combustor

The combustion chamber of a jet engine is where the well-known, powerful “bang!” is produced. Fuel mixes with the air that comes from the compressor and ignites, which is what produces this loud sound. Large quantities of fuel are burned within the combustion chamber, which is supplied by fuel spray nozzles. Heat is released to cause an expansion of the air within, as well as accelerating it for a stream of heated gas. Maximum heat release and minimum pressure loss are required.

The combustion chamber experiences a very high rise in temperature; just how much of an increase in temperature is determined by the amount of fuel within the chamber. A maximum temperature allowance is determined by the material used for the chamber, turbine blades, and nozzles. The combustion chamber must be able to maintain stable and efficient combustion over a varying range of engine operating conditions since the temperature of the mixed gas determines the engine’s thrust. Our jet racing applications make combustion temperatures even hotter. To manage that, we go to Florida Tech’s Center for Advanced Coatings where we can take advantage of ceramic coatings that provide resistance to extreme heat.

Annular combustion chambers typically have a more consistent exit temperature as well. As mentioned before, a minimum pressure drop is desired within the combustion chamber, and annular combustors have the lowest pressure drop of all the types of combustion chambers, making them very efficient. These aspects make it one of the most commonly used types of combustion chamber.

The exit of the combustion chamber is the hottest region of the engine, therefore a material that is able to withstand these high temperatures needs to be used. Inconel is typically the material of choice because of its thermal properties. It is high strength and corrosion resistant as well. Also, it is quite easily fabricated, making it convenient for manipulation into complex shapes.

1.4.3 Turbine

The turbine, like the compressor, is composed of several rows of airfoil cascades. Some of the rows, called rotors, are connected to the central shaft and rotate at high speed. Other rows, called stators, are fixed and do not rotate. The job of the stators is to keep the flow from spiraling around the axis by bringing the flow back parallel to the axis.

Depending on the engine type, there may be multiple turbine stages present in the engine. Turbofan and Turboprop engines usually employ a separate turbine and shaft to power the fan and gearbox respectively. Such an arrangement is termed a two-spool engine. In

turbojets, most of the hot exhaust comes from the surrounding air.) Leaving the burner, the hot exhaust is passed through the turbine. The turbine works like a windmill. Instead of needing energy to turn the blades to make the airflow, the turbine extracts energy from a flow of gas by making the blades spin in the flow. In a jet engine, we use the energy extracted by the turbine to turn the compressor by linking the compressor and the turbine by the central shaft. The turbine takes some energy out of the hot exhaust, but there is enough energy left over to provide thrust to the jet engine by increasing the velocity through the nozzle.

Since the turbine extracts energy from the flow, the pressure decreases across the turbine. The pressure gradient helps keep the boundary layer flow attached to the surface of the turbine blades. Since the boundary layer is less likely to separate on a turbine blade than on a compressor blade, the pressure drop across a single turbine stage can be much greater than the pressure increase across a corresponding compressor stage. A single turbine stage can be used to drive multiple compressor stages. Because of the high-pressure change across the turbine, the flow tends to leak around the tips of the blades. Turbine blades exist in a much more hostile environment than compressor blades. Sitting just downstream of the burner, the blades experience flow temperatures of more than a thousand degrees Fahrenheit. Turbine blades must be made of special materials that can withstand the heat, or they must be actively cooled.

Work Done

2.1 Mathematical Modeling of the Turbojet Engine

By creating mathematical models, we can predict how the engine will respond to different inputs and conditions, such as changes in fuel flow, airspeed, or environmental conditions. This insight is crucial, where the interaction between components (compressor, turbine, combustion chamber) significantly affects performance. Mathematical model provides a basis for analyzing how to maximize thrust, fuel efficiency, and operational stability. By simulating various operating conditions, we can identify optimal configurations and operating points, reducing fuel consumption and improving thrust without the need for extensive physical testing. Experimentally testing every possible condition is impractical due to high costs and time constraints. Modeling enables us to simulate different scenarios digitally, allowing for rapid iteration and analysis. This approach is especially important in aerospace applications where real-world testing can be very costly. Mathematical models help establish a foundation for control system design. By understanding how the engine responds dynamically (especially during transient events like startup or shutdown), so that we can develop control algorithms to maintain stability, protect components from extreme conditions, and ensure consistent performance.

2.1.1 Compressor Modeling

A compressor in a turbojet engine increases the pressure and temperature of incoming air before it enters the combustion chamber. The compressor in AMT Olympus HP engine is a single-stage radial compressor.

Assumptions:

- Steady-state mass flow through the compressor.
- No significant heat loss from the compressor to the surroundings.
- Air behaves as an ideal gas, allowing simplification of thermodynamic properties.

The performance of the compressor can be represented by the following equations, which involve total (stagnation) pressure, temperature, efficiency, and specific heat. The stagnation temperature at the compressor outlet T_{t3} can be calculated using the inlet stagnation temperature T_{t2} , isentropic efficiency η_c , and the pressure ratio π_c across the compressor.

$$T_{t3} = T_{t2} + \frac{T_{t2}}{\eta_c} \cdot \left[\left(\frac{P_{t3}}{P_{t2}} \right)^{\frac{\gamma_a - 1}{\gamma_a}} - 1 \right]$$

$$\pi_c = \frac{P_{03}}{P_{02}} : \text{Compressor pressure ratio}$$

$$\gamma_a : \text{Ratio of specific heats}$$

The compressor's performance can also be characterized by dimensionless maps or "compressor maps" that relate various operating parameters such as corrected mass flow, pressure ratio, and efficiency. The Beta Line Method is commonly used to digitize and interpolate these maps for computational modeling. From Compressor Map Interpo-

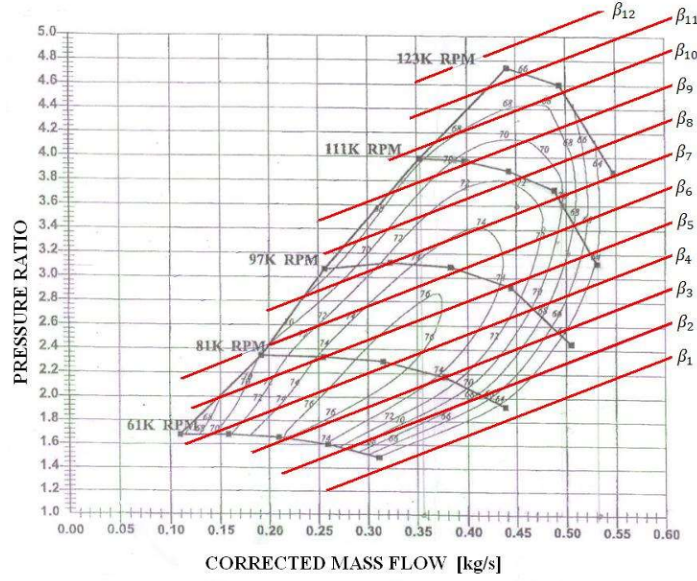


Figure 2.1: Compressor Map for Beta Line Method

lation, values of efficiency η_c and mass flow rate \dot{m}_C can be determined based on the pressure ratio and shaft speed (N).

Mass Flow Rate:

$$\dot{m}_C = f(N, \pi_c)$$

Efficiency:

$$\eta_c = g(N, \pi_c)$$

2.1.2 Combustion Chamber Modeling

The combustion chamber mixes compressed air with fuel and ignites this mixture to increase the air's temperature and energy before it enters the turbine. This rise in temperature and energy is crucial for driving the turbine, which, in turn, powers the compressor. Nonlinear combustion chamber equations were calculated from the laws of conservation principles. Dynamic equations were established from the conservation of mass m and internal energy U . Nonlinear equations for combustion chamber derived using conservation of mass and internal energy. Uses Runge-Kutta numerical method to solve these equations which is basically a Numerical Integration. Update fuel mass flow rate (\dot{m}_f) using a Proportional-Integral (PI) controller. Combustion chamber exit conditions are being calculated from the equations. (Eq. 2.1 and 2.2)

Assumptions:

- Constant pressure loss coefficient σ_{comb}
- Constant efficiency η_{comb}
- Enthalpy of fuel is neglected.

$$\frac{dP_{t4}}{dt} = \frac{P_{t4}}{m_{comb}} \cdot (\dot{m}_c + \dot{m}_{fuel} - \dot{m}_T) + \frac{P_{t4}}{T_{t4} \cdot c_{vmed} \cdot m_{comb}} \cdot \left[\dot{m}_c \cdot c_{pair} \cdot T_3 - \dot{m}_T \cdot c_{pgas} \cdot T_{t4} + Q_f \cdot \eta_{comb} \cdot \dot{m}_{fuel} - c_{vmed} \cdot T_{t4} \cdot (\dot{m}_c + \dot{m}_{fuel} - \dot{m}_T) \right] \quad (2.1)$$

$$\frac{dT_{t4}}{dt} = \frac{\dot{m}_c \cdot c_{pair} \cdot T_3 - (\dot{m}_c + \dot{m}_{fuel}) \cdot c_{pgas} \cdot T_{t4} + Q_f \cdot T_{t4} \cdot \dot{m}_{fuel}}{c_{vmed} \cdot m_{comb}} \quad (2.2)$$

2.1.3 Turbine Modeling

The turbine receives high-energy, high-temperature gas from the combustion chamber, which it expands to produce mechanical work that powers the compressor. In our turbojet engine, the turbine is a single-stage axial turbine, which can be modeled using performance maps, efficiency, and conservation equations.

Assumptions:

- Heat loss is neglected.
- Constant mass flow rate.
- No energy storage effect; $U_4 = \text{constant}$.

Total temperature, mass flow rate and efficiency of the turbine are found by using the subsequent equations. Number “4” refers to inlet of the turbine, number “5” refers to output point for the turbine.

$$T_{t5} = T_{t4} + T_{t4} \cdot \eta_t \cdot \left[1 - \left(\frac{P_{t5}}{P_{t4}} \right)^{(\gamma_g - 1)/\gamma_g} \right] \quad (2.3)$$

$$\dot{m}_T = f(N, \pi_T) \quad (2.4)$$

$$\eta_T = g(N, \pi_T) \quad (2.5)$$

Turbine map is demonstrated in Figure 2.2. There are shaft speed lines on this map and pressure ratio versus mass flow rate for turbine. Since efficiency data are taken from tables, there are no efficiency lines on the map. Turbine map was obtained by the theoretical calculations and experimental data. These investigations were made in Middle East Technical University Aerospace Engineering Department. Turbine exit conditions calculated above are employed for the work balance of the engine. At this point turbine produces energy to rotate to the compressor shaft. This energy is facilitated for the compression of the air again.

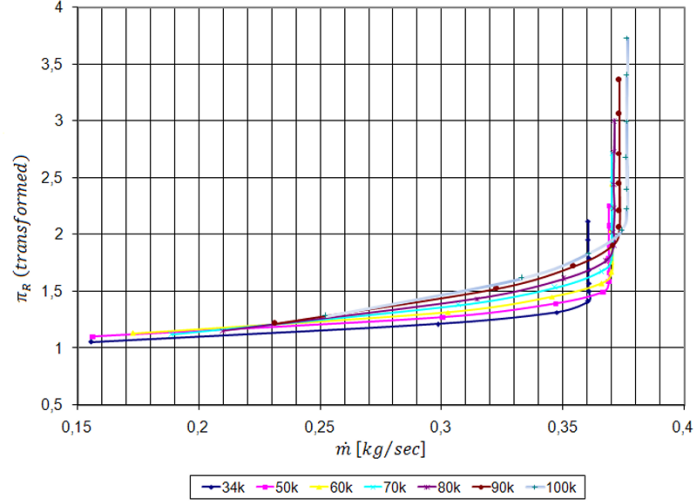


Figure 2.2: Turbine Map

2.2 Work Balance and Thrust Calculation

Work balance of a gas turbine engine can be determined as equality between compressor and turbine work. All gas turbine engine work with the following sequence, first of all, compressor compresses the air with the energy which it takes from the starter engine. When the engine reaches its idle speed, starter stops and compressor takes the energy from the turbine which is connected with a single shaft. This relation continues until fuel flow is cut off.

$$\text{Power} = \text{Torque} \cdot \text{Angular Velocity} = T \cdot \omega \quad (2.6)$$

$$T = I \cdot \alpha \quad (2.7)$$

where I refers to moment of inertia and α refers to angular acceleration. Therefore,

$$W = I \cdot \alpha \cdot \omega = I \cdot \frac{d\omega}{dt} \cdot \omega \quad (2.8)$$

$$\frac{d\omega}{dt} = \frac{1}{I \cdot \omega} \cdot (W_t - W_c - W_f) \quad (2.9)$$

where W_t , W_c , W_f are refer to turbine, compressor and friction power respectively.

$$W_c = \dot{m}_c \cdot (h_3 - h_2) = \dot{m}_c \cdot c_{p,air} \cdot (T_3 - T_{t2}) \quad (2.10)$$

$$W_t = \dot{m}_t \cdot (h_4 - h_5) = \dot{m}_t \cdot c_{p,gas} \cdot (T_4 - T_{t5}) \quad (2.11)$$

$$\frac{d\omega}{dt} = \frac{1}{I \cdot \omega} \{ [\dot{m}_t \cdot c_{p,gas} \cdot (T_4 - T_{t5})] - [\dot{m}_c \cdot c_{p,air} \cdot (T_3 - T_{t2})] - W_f \} \quad (2.12)$$

Gas turbine engine converts chemical energy into kinetic energy by changing U_0 to U_e , which are inlet and exit velocities of the gases, respectively. This process creates the thrust.

$$F = \dot{m} \cdot (U_e - U_0) \quad (2.13)$$

$$F = (\dot{m}_f + \dot{m}_c) \cdot U_e - \dot{m}_c \cdot U_0 \quad (2.14)$$

Thrust can be calculated from Eq. (2.14). Nevertheless, some algebraic equations of main mechanical parts of the engine are needed to model the overall engine. These equations are found by using some assumptions given in previous and certain significant thermodynamic laws such as isentropic relations.

2.3 PI Controller Design

The proportional controller commonly known as PI controller is an essential part of the Industrial Automation and Control system. It is a closed-loop feedback control mechanism that aims to adjust the process variable by manipulating the variable based on the error between the setpoint and the process variable. It strikes a balance between quick response to deviations and long-term error elimination. Tuning the controller allows adjustment to meet the desired value.

What is Proportional Integral Controller?

PI controller or Proportional controller is a combination of Proportional controller action and Integral controller action which is designed to regulate a process variable based on its setpoint and manipulated variable. Also, it can be identified as a combination of proportional and integral controllers.

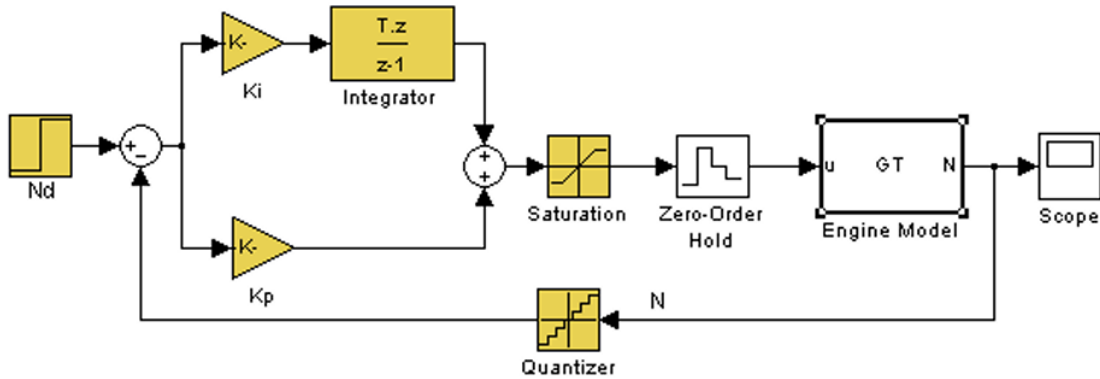


Figure 2.3: PI Controller

The mathematical representation of the proportional plus integral controller is given as:

$$m(t) = K_p \cdot e(t) + K_i \cdot \int e(t) dt$$

As shown in figure 2.3, a proportional controller operates in a step-by-step manner to regulate a control system. It starts by calculating the error, which is the difference between the desired setpoint and the current process variable. The proportional(P) part of the controller multiplies this error with proportional gain(K_p) and generates an immediate action. which is directly proportional to the error. The Integral(I) part computes the cumulative sum of past errors by integrating gain(K_i). The two components P and I are then added together to determine the control output, which is applied to the system. This output adjusts the ultimate output of the system by minimizing the error over time

and maintaining the setpoint.

Proportional Controller: Proportional Controller in electronics engineering continuously adjusts the output based on the current error signal. The p term or proportional term calculates the current error from the setpoint and the current manipulated variable output. It generates a control signal to regulate the error.

$$m(t) \propto e(t)$$
$$m(t) = K_p \cdot e(t)$$

Where:

- $m(t)$ = Output of P-controller at time t
- $e(t)$ = Setpoint - Output at time t
- K_p = tuning constraints for proportional action

Integral Controller: In the Integral Controller, the I term or Integral term actuates the past error over time and generates a control action to eliminate the accumulated steady-state error. It ensures that even small errors are eventually corrected. It eliminates offset but can lead to sluggish responses and overshooting if too aggressive.

$$m(t) \propto \int e(t) dt$$
$$m(t) = K_i \cdot \int e(t) dt$$

Where:

- K_i = tuning constraints for integral action

The Characteristics of the P and I Terms:

Increasing the proportional gain (K_p) has the effect of proportionally increasing the control signal for the same level of error. The fact that the controller will "push" harder for a given level of error tends to cause the closed-loop system to react more quickly, but also to overshoot more. Another effect of increasing K_p is that it tends to reduce, but not eliminate, the steady-state error. Adding an integral term to the controller (K_i) tends to help reduce steady-state error. If there is a persistent, steady error, the integrator builds and builds, thereby increasing the control signal and driving the error down. A drawback of the integral term, however, is that it can make the system more sluggish (and oscillatory) since when the error signal changes sign, it may take a while for the integrator to "unwind."

The general effects of each controller parameter (K_p , K_i) on a closed-loop system are summarized in the table below. Note, that these guidelines hold in many cases, but not all. If you truly want to know the effect of tuning the individual gains, you will have to do more analysis or perform testing on the actual system.

2.4 Implementation of Fuzzy Logic Control

Fuzzy logic control is a technique used to handle the complexity and non-linearity of systems where a precise mathematical model may be hard to obtain or apply effectively. It's based on fuzzy logic, a form of logic that can deal with degrees of truth rather than binary true/false values. In control applications, fuzzy logic uses linguistic rules (e.g., "If-Then" rules) to govern system behavior, allowing it to mimic human decision-making.

Basic Components of a Fuzzy Logic Controller A fuzzy logic control system which is shown in Figure 2.4, generally consists of the following four components:

- **Fuzzification:** Converts crisp input values into fuzzy values using membership functions.
- **Rule Base:** Contains a set of "If-Then" rules that describe the desired control behavior in terms of fuzzy logic.
- **Inference Mechanism:** Evaluates which rules are relevant based on the current inputs and determines the controller's output using the rule base.
- **Defuzzification:** Converts fuzzy output values back into a crisp value that can be used by the control system.

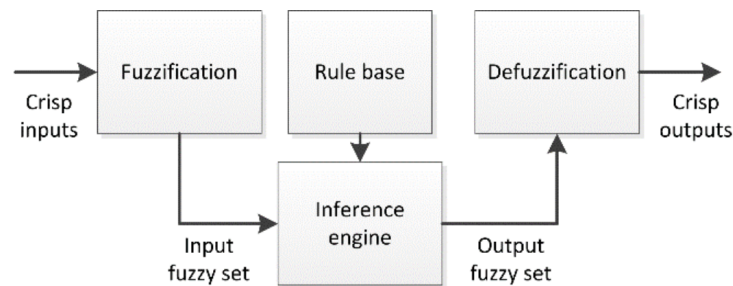


Figure 2.4: Fuzzy Logic Controller

For a turbojet engine, typical inputs include Error (difference between desired thrust or speed and current thrust or speed) and Tate of Change of Error (how fast the error is changing). And the output should be Fuel Flow Rate. Adjusting fuel flow rate, controls the energy input to the combustion process, which directly affects thrust and engine speed. Fuzzification involves mapping input values to a set of membership functions, each representing a fuzzy set. For example, the error could be described with linguistic terms such as: Negative Large (NL), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Large (PL). These terms are assigned fuzzy membership functions, such as triangular or trapezoidal shapes, that define their degree of truth for a given input value. The rule base consists of a collection of "If-Then" statements that describe the control action under various input conditions. Here's an example of some rules you might have for controlling fuel flow rate based on error (E) and change in error (dE):

- If (E is PL) and (dE is PS), Then (Fuel Flow is Increase Small)
- If (E is NL) and (dE is NS), Then (Fuel Flow is Decrease Large)
- If (E is Z), Then (Fuel Flow is No Change)

These rules allow the controller to adjust fuel flow based on how far off the current engine speed or thrust is from the target and how rapidly that error is changing.

The inference mechanism evaluates which rules are triggered based on the current values of error and change in error. It combines the results of the relevant rules using fuzzy logic operations to calculate a fuzzy output. Defuzzification translates the fuzzy output from the inference mechanism into a crisp output value (e.g., an exact fuel flow rate adjustment). Common defuzzification methods include:

Center of Gravity (COG): Takes the weighted average of all possible outputs.

Max Membership: Chooses the output with the highest degree of membership.

For instance, if the fuzzy output suggests a fuel flow rate between "Increase Small" and "Increase Medium," defuzzification would calculate a precise fuel flow rate adjustment.

Example of Fuzzy Logic Control in Operation.

Initial Condition: Engine speed is below target, creating a positive error.

Rule Activation: The controller evaluates rules like "If Error is Positive Large and dE is Positive Small, Then Increase Fuel Flow by a Large amount."

Output Adjustment: Based on the rules, the controller increases fuel flow.

System Response: As fuel flow increases, engine speed rises, reducing the error.

Controller Adjustment: The controller continues to adjust fuel flow dynamically as the error approaches zero.

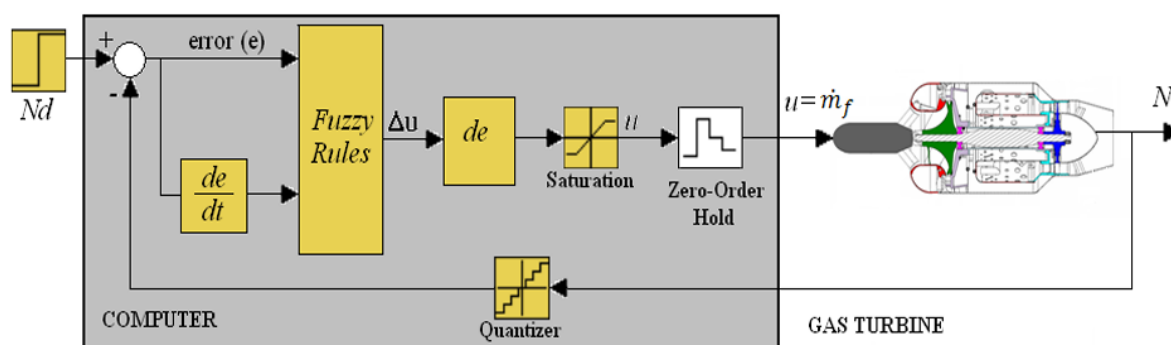


Figure 2.5: Fuzzy Logic Control System

2.5 Design of Simscape Model of Engine

This model represents a simplified turbojet engine based on the Brayton cycle, excluding the Auxiliary Power Unit (APU) typically found in auxiliary applications. The system comprises three primary components: a compressor, a combustion chamber, and a turbine. In this setup, power input is managed by controlling the shaft speed, which has a direct effect on the temperature within the combustion chamber.

The model incorporates a dual-PI controller system to regulate both shaft speed and combustion chamber temperature. The "Speed Regulator" PI adjusts the shaft speed (N) to match a desired setpoint N_{sp} . This regulated speed then feeds into the "Temperature Regulator" PI, which fine-tunes the combustion chamber temperature (T_3) to achieve a specified setpoint T_{3sp} . By coordinating these two PI controllers, the model achieves precise thermal control, with the speed control feeding into temperature control for enhanced stability. Temperatures are measured at the inlet and outlet of each component, providing real-time insights into the thermal profile across the compressor,

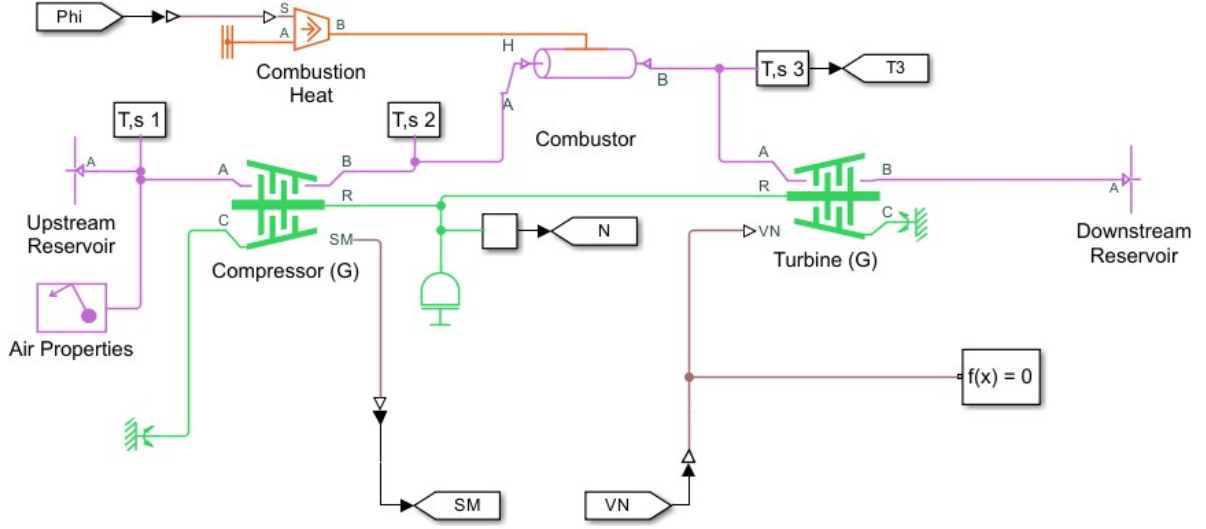


Figure 2.6: Turbojet Simscape Model

combustion chamber, and turbine. This streamlined representation focuses on temperature control through shaft speed adjustments, avoiding the complexity of additional operating scenarios.

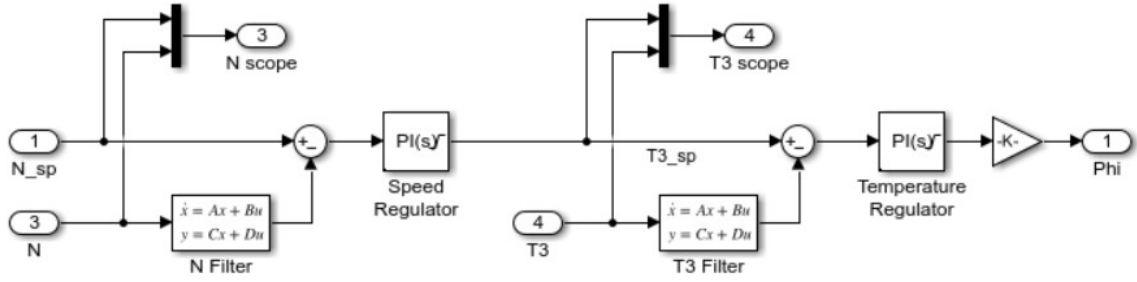



Figure 2.7: Dual Controller Block

The provided block diagram illustrates the dual-PID control system for the turbojet engine model. Here, the "Speed Regulator" PI controls the shaft speed (N) to meet the setpoint N_{sp} , while the "Temperature Regulator" PID controls the temperature inside the combustion chamber (T3) according to the setpoint $T3_{sp}$. The flow works as follows:
Speed Regulation: The Speed Regulator PI takes the difference between the actual speed (N) and the desired speed N_{sp} , adjusting the output to maintain the target speed.
Temperature Regulation: The Temperature Regulator PI receives input from the difference between the actual temperature (T3) and the target temperature $T3_{sp}$, allowing it to control combustion chamber temperature indirectly by adjusting parameters like fuel flow or air mixture. Both controllers work together, with the speed control feeding into the temperature control, ensuring stable combustion and effective control of thermal conditions within the combustion chamber.

Description of the parameters that we've to set for compressor and Turbine blocks are given below in the fig respectively


Block Parameters: Compressor (G)
✕

Compressor (G)
☒ Auto Apply
?

Settings
Description

NAME	VALUE
<div> Compressor Map </div>	
Parameterization	Tabulated <div> </div>
<div> Corrected speed index vector, N </div>	<div> compressor.corrected_speed_ve... rpm <div> </div> </div>
<div> Beta index vector, beta </div>	<div> compressor.beta_vector < 1x11 double > </div>
<div> Pressure ratio table, pr(N,beta) </div>	<div> compressor.map.pressure_ratio < 10x11 double > </div>
<div> Corrected mass flow rate table, mdot(N,b... </div>	<div> compressor.map.corrected_mas... kg/s <div> </div> </div>
<div> Isentropic efficiency table, eta(N,beta) </div>	<div> compressor.map.eta < 10x11 double > </div>
Report when surge margin is negative	None <div> </div>
<div> Reference Data </div>	

Figure 2.8: Compressor Block Parameters

Block Parameters: Turbine (G)
✕

Turbine (G)

☒ Auto Apply
 ?

Settings

Description

NAME	VALUE
<div> <div>▼</div> Parameterization </div>	
Turbine map parameterization	Tabulated data - flow rate and efficiency vs. pressure ratio <div>▼</div>
<input checked="" type="checkbox"/> Enable variable nozzle input port	
<div> <div>➤</div> Minimum nozzle opening fraction </div>	0.15
<div> <div>➤</div> Maximum nozzle opening fraction </div>	1
<div> <div>▼</div> Flow Rate Data </div>	
<div> <div>➤</div> Pressure ratio vector, pr </div>	turbine.flow_map.pressure_ratio <div>< 1x16 double ></div>
<div> <div>➤</div> Corrected mass flow rate vector, mdot(pr) </div>	turbine.flow_map.corrected_mas... <div>kg/s</div> <div>▼</div>
<div> <div>▼</div> Efficiency Data </div>	
<div> <div>➤</div> Pressure ratio vector, pr </div>	turbine.efficiency_map.pressure_ratio <div>< 1x8 double ></div>
<div> <div>➤</div> Isentropic efficiency vector, eta(pr) </div>	turbine.efficiency_map.isentropic_efficiency <div>< 1x8 double ></div>
<div> <div>▼</div> Reference Data </div>	
<div> <div>➤</div> Reference pressure for corrected flow </div>	turbine.reference_pressure <div>1.5</div> <div>MPa</div> <div>▼</div>
<div> <div>➤</div> Reference temperature for corrected flow </div>	turbine.reference_temperature <div>...</div> <div>K</div> <div>▼</div>
<div> <div>➤</div> Mechanical efficiency </div>	mechanical_efficiency <div>0.98</div>
<div> <div>➤</div> Inlet area at port A </div>	flow_area * scale_factor <div>0.002</div> <div>m^2</div> <div>▼</div>
<div> <div>➤</div> Outlet area at port B </div>	flow_area * scale_factor <div>0.002</div> <div>m^2</div> <div>▼</div>

Figure 2.9: Turbine Block Parameters

Results and Discussion

Feature	PI Controller	Fuzzy Logic Controller
Response Speed	Moderate: Generally effective but may be slower to settle, especially in nonlinear regions of operation.	Faster: Adaptive rule-based control allows for quicker adjustments, typically achieving the target faster.
Overshoot	Minimal overshoot, though it may vary based on gain settings and engine conditions.	Very low or minimal overshoot due to the adaptive nature of the rule set, which helps control changes more softly.
Settling Time	Approx. 0.8 to 1 second (for speed change from 65,000 to 85,000 rpm).	Faster settling, typically around 0.7 seconds, as FLC can adapt to changing error rates better than PI.
Steady-State Error	Near-zero error in steady state, though minor adjustments may still occur if system load changes significantly.	Very low steady-state error due to adaptive rule-based control, which is generally effective even with disturbances.
Control of Non-linearities	Limited: The PI controller requires different gain settings for different operating points, as it's less effective at handling the nonlinear behavior of the engine, especially during transient operations.	Strong: FLC is naturally suited for nonlinear systems, with its rule-based approach adapting well to rapid changes and variations in system behavior.
Combustion Chamber Temperature	Moderate: Possible higher peaks in temperature due to step-change responses inherent in PI control.	Better control over temperature, as FLC reduces abrupt fuel adjustments, leading to smoother changes in chamber temperature.
Thrust Response	Thrust rises as expected but may see oscillations or require longer to stabilize due to fixed gain settings.	Thrust stabilizes smoothly with fewer fluctuations, benefiting from the fuzzy controller's adaptive adjustments.

Table 3.1: Comparison of PI and Fuzzy Logic Controllers for Turbojet Engine Control

3.1 MATLAB Code Results

We are using initial condition as follows:

Initial Engine Speed: 65,000 rpm

Target Engine Speed: 85,000 rpm (a step input)

Sampling Time: 2 ms

Controller Parameters: After tuning for the turbojet, we assume:

$K_p = 4.56 \times 10^{-4}$

$K_i = 3.5 \times 10^{-5}$

Based on this setup, here's the simulation results: The speed should gradually increasing from 65,000 rpm to 85,000 rpm following the step input. Overshoot is minimal, ideally under 5%. Settling time is round 0.8 to 1 second for the engine speed to stabilize near 85,000 rpm. Combustion Chamber Temperature (T_4) will initially rise due to the increased fuel flow. After reaching peak, temperature should stabilize just below a maximum operational limit. The thrust output should increase in response to the rise in fuel flow, showing an initial peak. Steady-State Thrust is settling at the desired level to maintain 85,000 rpm, typically around 76 N. Thus this is ensuring a smooth ramp-up to target speed without abrupt jumps or oscillations.

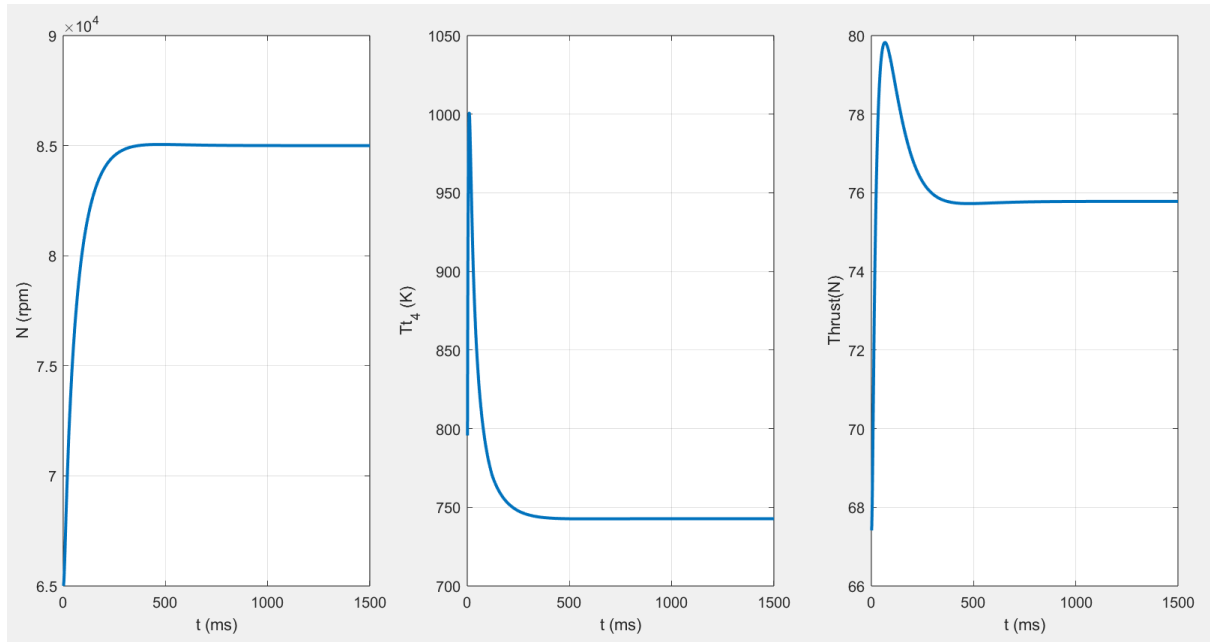


Figure 3.1: PI controller Results

By applying Fuzzy Logic Control, engine speed increasing from 65,000 rpm to 85,000 rpm with minimal overshoot and oscillation. The adaptive nature of FLC allows for a smoother response, especially in nonlinear regions, compared to a fixed-gain PI controller. Combustion Chamber Temperature vs. Time Graph shows a modest peak in turbine inlet temperature as the speed ramps up, followed by stabilization within the target range. FLC avoids excessive heating by gradually adjusting fuel flow based on error and rate of change, leading to safer temperature control. Thrust output rises smoothly and stabilizes at a value consistent with the target speed (85,000 rpm). The FLC's rule-based adjustments help keep thrust stable, providing responsive yet controlled changes without sharp spikes.

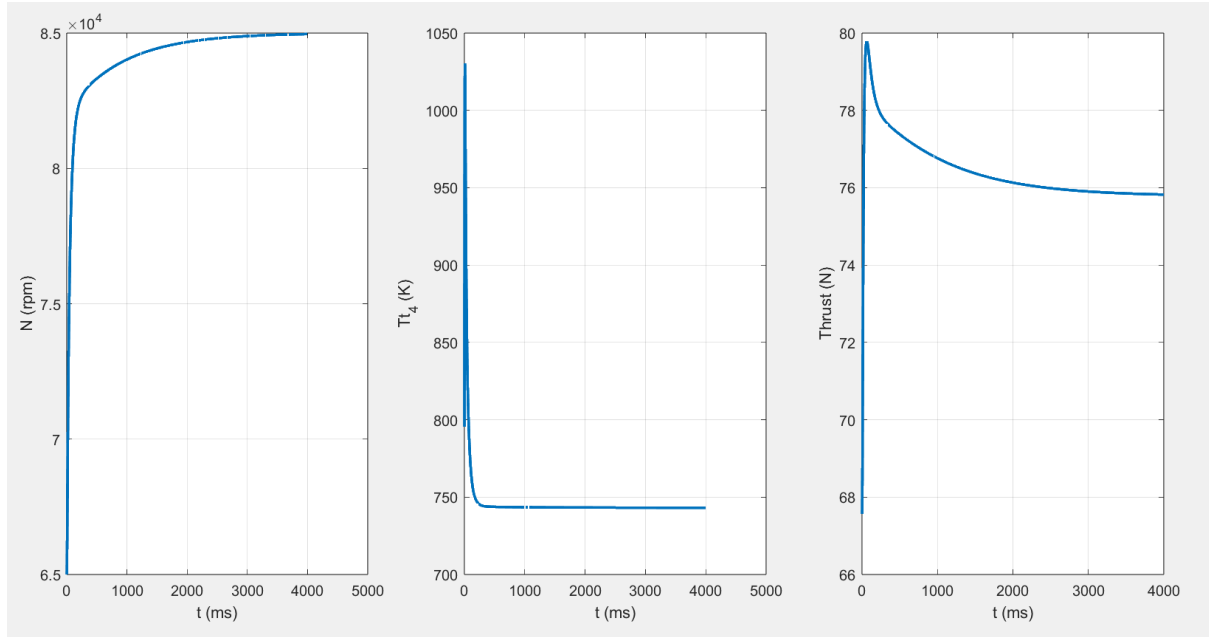


Figure 3.2: Fuzzy Logic controller Results

3.2 Simscape Model Results

A compressor map like the one we've generated is a graphical representation of the performance characteristics of a compressor, typically in terms of pressure ratio, corrected mass flow rate, and efficiency at various operating conditions. Let's break down the key elements of your compressor map:

Axes:

Corrected Mass Flow Rate (kg/s): Shown on the x-axis, this represents the mass flow rate through the compressor, adjusted for inlet conditions. It helps to standardize performance across different operating environments.

Pressure Ratio: Shown on the y-axis, this is the ratio of the outlet pressure to the inlet pressure. A higher pressure ratio indicates greater compression capability.

Lines and Regions:

Isentropic Efficiency Contours (Yellow Lines): These lines indicate constant efficiency levels across different points on the map. Efficiency is a measure of how effectively the compressor is compressing the air relative to ideal (isentropic) compression. Higher efficiency values are typically desired.

Constant Speed Lines (Blue Lines): These lines show the performance of the compressor at different rotational speeds, measured in revolutions per minute (RPM). Each line represents a specific speed, labeled with its corresponding RPM (e.g., 6000, 8000, 10000 RPM). The flow and pressure ratio vary along these lines as the compressor operates at constant speed.

Choke Line (Orange Line): The choke line indicates the maximum flow rate at each speed where the compressor can operate before reaching flow instability. Operating beyond this line leads to choking, where further increases in mass flow do not significantly increase the pressure ratio.

Surge Line (Red Line): The surge line represents the minimum stable flow rate for each speed. Operating below this line can cause the compressor to "surge," a condition where

airflow reverses or becomes unsteady, leading to potential damage. It's crucial to avoid operating close to or below this line.

Interpreting the Map:

At lower RPMs, the map shows a limited range of mass flow and pressure ratio, indicating that the compressor cannot achieve high pressure ratios or flow rates at lower speeds.

As the RPM increases (moving up the blue speed lines), the compressor achieves higher pressure ratios and larger corrected mass flows.

Efficiency tends to peak in a certain region of the map (often around mid-to-high RPMs and a balanced flow rate) and falls off as you move closer to the choke or surge lines.

This region represents the most efficient operating conditions for the compressor.

Optimal Operating Range:

The area between the surge and choke lines, particularly around the higher efficiency contours, represents the optimal operating range. Running the compressor in this region ensures stable and efficient performance without risking surge or choke conditions.

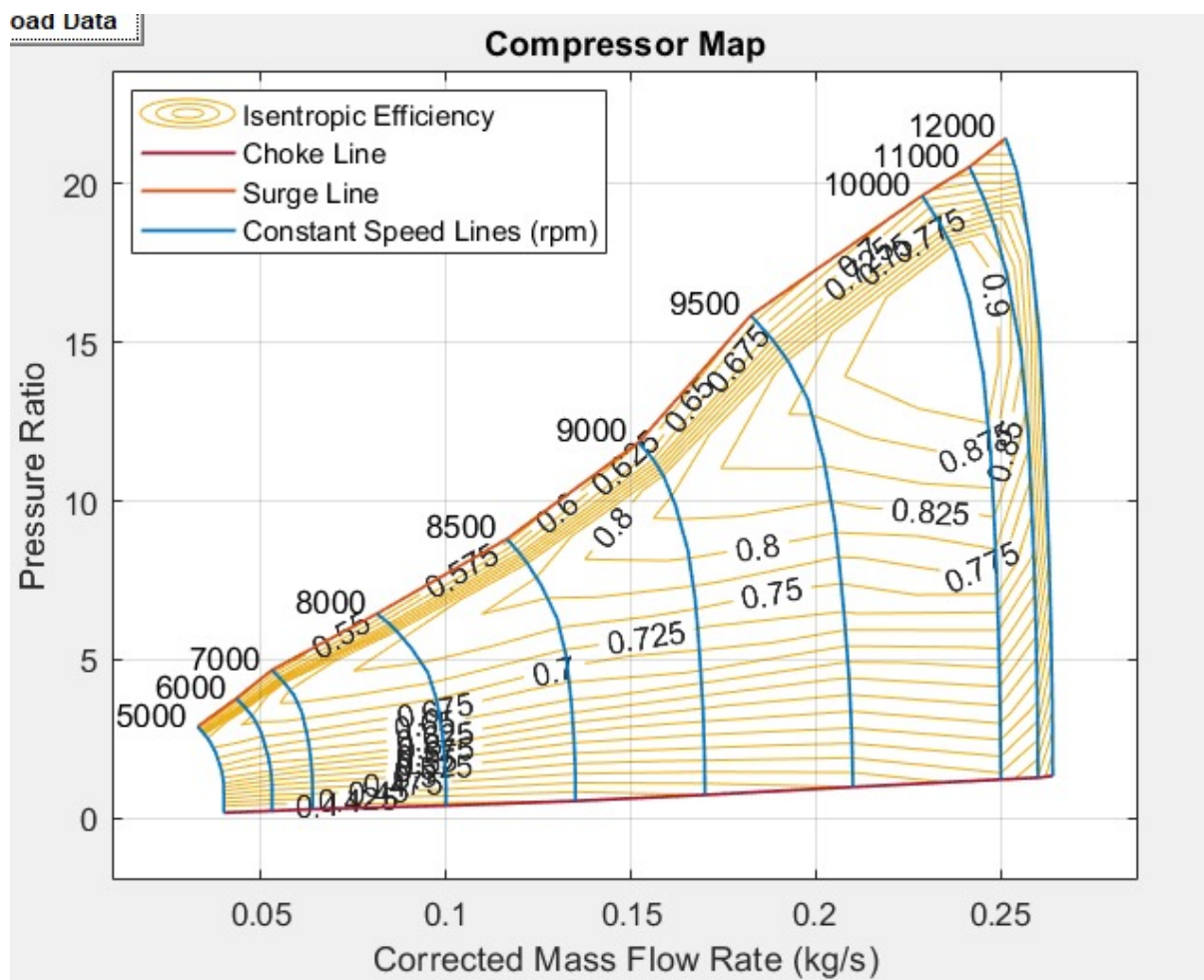


Figure 3.3: Compressor Map

The figure below is a subsystem that consists of the properties of Air

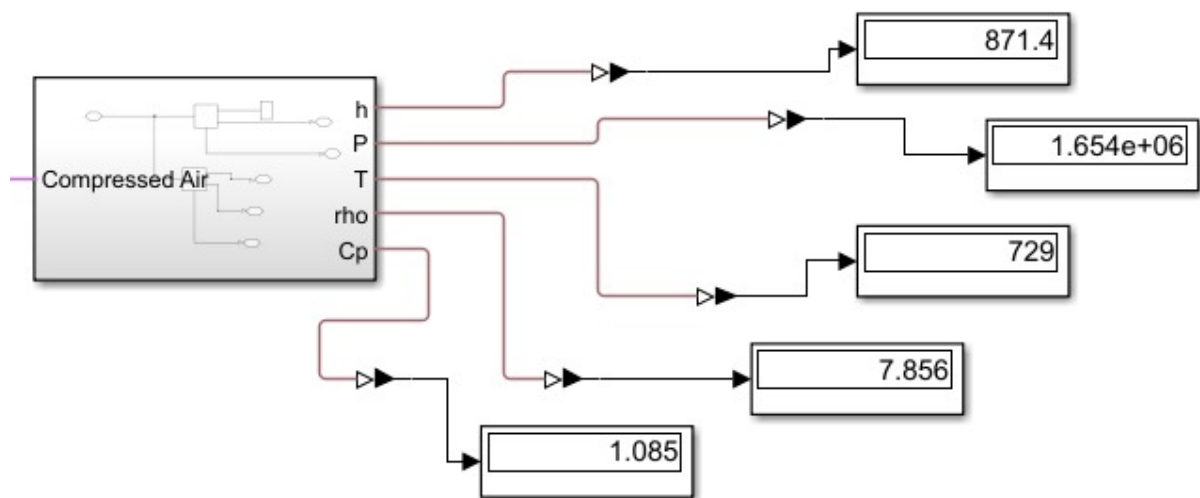


Figure 3.4: Properties of air at compressor outlet

Conclusion

We successfully developed and evaluated a virtual model of a turbojet engine tailored for UAV applications, emphasizing the design and implementation of control strategies using both PI and Fuzzy Logic controllers. Through MATLAB and Simulink simulations, it was observed that the Fuzzy Logic controller outperformed the PI controller, providing smoother thrust control, reduced overshoot, and more precise management of combustion chamber temperature, particularly in non-linear operating conditions. The mathematical modeling and control system analysis provided key insights into optimizing fuel efficiency and thrust response. These results lay a solid foundation for future enhancements in UAV propulsion systems, focusing on adaptable and efficient control mechanisms.

Appendix

A.1 MATLAB Codes

MATLAB Code for PI Control

```
1  clc; clear;
2  format short e
3
4  % Parameters and initialization
5  i = 1; m = 1; n = 1; c = 1; q = 1; kl = 0;
6  Pt_2 = 93380; % Pascal
7  Tt_2(1) = 303; % Kelvin
8  Pt_ref = 100000; % Pascal (reference)
9  Tt_ref = 298; % Kelvin (reference)
10 sigma_comb = 0.95; % Combustion chamber pressure ratio
11 nu_comb = 0.6; % Combustion chamber efficiency
12 Pi_n = 0.95; % Nozzle pressure ratio
13 M9 = 1; % Choke condition
14 M0 = 0; % Assumption
15 H = 0; % Height
16
17 % Additional initialization
18 teta = (Tt_2(1) / Tt_ref); % Dimensionless temperature parameter
19 delta = (Pt_2 / Pt_ref); % Dimensionless pressure parameter
20 N(1) = 65000; % RPM
21 I = 1.787e-4; % Moment of Inertia from CATIA [kg*m^2]
22 eff_t = 0.85; % Turbine efficiency
23 gama_c = 1.4; % Compressor specific heat ratio
24 gama_t = 1.35; % Turbine specific heat ratio
25 Qf = 43250e3; % [J/kg] Lower calorific value of fuel
26 R = 286.9; % [J/(kg*K)] Individual gas constant
27 V_comb = 6.687233e-4; % Volume of the combustion chamber
28
29 % Compressor-related parameters
30 PI_C = 9.06e-21 * N(1)^4 - 1.365e-15 * N(1)^3 + 3.3806e-10 * N(1)^2 -
    9.7556e-6 * N(1) + 1.1929;
31 Cp_a(1) = -1.9747e-028 * N(1)^6 + 8.3860e-023 * N(1)^5 - 1.4447e-017 *
    N(1)^4 + 1.2943e-012 * N(1)^3 - 6.2674e-008 * N(1)^2 + 1.5743e-003 *
    N(1) + 9.8846e+002;
32 Tt_4(1) = 5.5099e-017 * N(1)^4 - 1.1969e-011 * N(1)^3 + 9.7144e-007 * N
    (1)^2 - 3.7423e-002 * N(1) + 1.4273e+003;
33 % Initialize mass flow rates
34 mf_dot(1) = 0.00456;
35 Kp = 0.00000456 / 10000;
```

```

36 Ki = 0.000000000035;
37 MODD = 2; % MODD value based on your code
38 A_0 = 0.001378;
39 A_9 = 0.00302;
40 % Delta "t"
41 h = 0.001;
42 E(1) = 0;
43
44 %-----BETA LINE METHOD-----
45 for j=1:(1.5/h)
46     N_Ref(j)=85000;
47     N_R(j)=N(j)/(teta)^0.5; % Corrected RPM
48     % Ensure correct initialization of Tt_2 and Pt_2
49     Tt_2(j+1)=Tt_2(j);
50     Pt_2(j+1)=Pt_2(j);
51
52     % Map data initialization (simplified for readability)
53     PI_C_m=[0 1.560 1.648 1.678 0 0 0 0 0 0 0 0
54             0 2.019 2.172 2.282 2.337 0 0 0 0 0 0 0
55             0 0 2.545 2.750 2.938 3.067 3.121 0 0 0 0 0
56             0 0 0 0 3.219 3.438 3.676 3.843 3.955 0 0 0
57             0 0 0 0 0 0 3.861 4.090 4.317 4.542 4.695 0];
58     N_cor=[61000 81000 97000 111000 123000 ];
59
60     % Compressor interpolation using Beta Line Method
61     while m==1
62         if N_cor(i) <= N_R(j) && N_R(j) <= N_cor(i+1)
63             k=(N_cor(i+1)-N_R(j))/(N_cor(i+1)-N_cor(i));
64             m=0;
65             t=i;
66         end
67         i=i+1;
68     end
69     i=1;
70
71     % Interpolation of compressor map values
72     for q = 1 : 12
73         N_R_matrix(q) = PI_C_m(t+1,q) - (PI_C_m(t+1,q) - PI_C_m(t,q)) *
74             k;
75     end
76
77     % Calculation of Beta for mass flow rate calculations
78     while n==1
79         if N_R_matrix(c) <= PI_C(j) && PI_C(j) <= N_R_matrix(c+1)
80             Beta = c - (-1 * (N_R_matrix(c) - PI_C(j))) / (N_R_matrix(c
81                 ) - N_R_matrix(c+1));
82             n=0;
83         end
84         c=c+1;
85     end
86     c=1;
87
88     % Mass Flow Rate Calculations
89     x = [1 2 3 4 5 6 7 8 9 10 11 12];
90     y = [61000 81000 97000 111000 123000 ];
91     z = [0 0.2777 0.2185 0.1387 0 0 0 0 0 0 0 0
92         0 0.4155 0.3758 0.3226 0.2531 0 0 0 0 0 0 0
93         0 0 0.4904 0.4638 0.4358 0.3894 0.318 0 0 0 0 0

```

```

92         0 0 0 0 0.5206 0.5045 0.4881 0.4514 0.3988 0 0 0
93         0 0 0 0 0 0 0.5435 0.5271 0.5094 0.4918 0.4517 0];
94
95 % Efficiency Calculations
96 x_1 = [1 2 3 4 5 6 7 8 9 10 11 12];
97 y_1 = [61000 81000 97000 111000 123000 ];
98 z_1 = [0 0.711 0.759 0.69 0 0 0 0 0 0 0 0
99         0 0.669 0.733 0.76 0.742 0 0 0 0 0 0 0
100        0 0 0.646 0.698 0.735 0.752 0.726 0 0 0 0 0
101        0 0 0 0 0.65 0.68 0.704 0.715 0.706 0 0 0
102        0 0 0 0 0 0 0.62 0.64 0.655 0.664 0.66 0];
103
104 % Calculate mass flow rate and efficiency
105 m_dot_MAP(j)=interp2(x,y,z,Beta,N_R(j));
106 eff_c(j)=interp2(x_1,y_1,z_1,Beta,N_R(j));
107
108 %-----Compressor-----
109 mc_dot(j)=m_dot_MAP(j)*(delta)/((teta)^0.5);
110 Tt_3(j)=Tt_2(j)*(1+((1/eff_c(j))*((PI_C(j)^((gama_c-1)/gama_c))-1))
    );
111 Pt_3(j)=PI_C(j)*Pt_2(j);
112 Ta(j)=(Tt_2(j)+Tt_3(j))/2;
113 Cp_a(j+1)=1.0189e3 - 0.13784*Ta(j) + 1.9843e-4*Ta(j)^2 + 4.2399e-7*
    Ta(j)^3 - 3.7632e-10*Ta(j)^4;
114
115 %-----Combustion Chamber-----
116 mf_dot(j+1)=mf_dot(j);
117 kont=rem(j,MODD);
118 if kont==0
119     E(j)=N_Ref(j)-N(j);
120     mf_dot(j)=mf_dot(j-1)+ (Kp+Ki*MODD).*E(j)+Kp*E(j-1);
121     mf_dot(j+1)=mf_dot(j);
122 end
123
124 Tg(j)=(Tt_4(j)+Tt_3(j))/2;
125 Tm(j)=Tg(j);
126 f(j)=mf_dot(j)/mc_dot(j);
127
128 % Beta coefficients for gas
129 Bt(j)=-3.59494e2 + 4.5164*Tg(j) + 2.8116e-3*Tg(j)^2 - 2.1709e-5*Tg(
    j)^3 + 2.8689e-8*Tg(j)^4 - 1.2263e-11*Tg(j)^5;
130 Cp_g(j)=Cp_a(j)+(f(j)/(1+f(j)))*Bt(j);
131 Cv_comb(j)=Cp_g(j)-R;
132 Pt_4(j)=sigma_comb*Pt_3(j);
133 Pm(j)=(Pt_4(j)+Pt_3(j))/2;
134 Rho(j)=Pm(j)/(R*Tm(j));
135 M_comb(j)=Rho(j)*V_comb;
136
137 %-----Runge Kutta fourth order for Tt_4 -----
138 F(j)=(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+mf_dot(j))*Cp_g(j)*Tt_4(
    j)+Qf*nu_comb*mf_dot(j))/(Cv_comb(j)*M_comb(j));
139 K1(j)=F(j);
140 K2(j)=(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+mf_dot(j))*Cp_g(j)*(
    Tt_4(j)+(h/2)*K1(j))+Qf*nu_comb*mf_dot(j))/(Cv_comb(j)*M_comb(j)
    );
141 K3(j)=(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+mf_dot(j))*Cp_g(j)*(
    Tt_4(j)+(h/2)*K2(j))+Qf*nu_comb*mf_dot(j))/(Cv_comb(j)*M_comb(j)
    );

```

```

142 K4(j)=(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+mf_dot(j))*Cp_g(j)*(
      Tt_4(j)+h*K3(j))+Qf*nu_comb*mf_dot(j))/(Cv_comb(j)*M_comb(j));
143 Tt_4(j+1)=Tt_4(j)+(h/6)*(K1(j)+2*K2(j)+2*K3(j)+K4(j));
144
145 %-----Runge Kutta fourth order for Pt_4-----
146 H(j)=Pt_4(j)*(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+mf_dot(j))*Cp_g(
      j)*Tt_4(j)+Qf*nu_comb*mf_dot(j))/(Tt_4(j)*Cv_comb(j)*M_comb(j));
147 M1(j)=H(j);
148 M2(j)=(Pt_4(j)+h/2*M1(j))*(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+
      mf_dot(j))*Cp_g(j)*Tt_4(j)+Qf*nu_comb*mf_dot(j))/(Tt_4(j)*
      Cv_comb(j)*M_comb(j));
149 M3(j)=(Pt_4(j)+h/2*M2(j))*(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+
      mf_dot(j))*Cp_g(j)*Tt_4(j)+Qf*nu_comb*mf_dot(j))/(Tt_4(j)*
      Cv_comb(j)*M_comb(j));
150 M4(j)=(Pt_4(j)+h*M3(j))*(mc_dot(j)*Cp_a(j)*Tt_3(j)-(mc_dot(j)+
      mf_dot(j))*Cp_g(j)*Tt_4(j)+Qf*nu_comb*mf_dot(j))/(Tt_4(j)*
      Cv_comb(j)*M_comb(j));
151
152 Pt_4(j+1)=Pt_4(j)+h/6*(M1(j)+2*M2(j)+2*M3(j)+M4(j));
153 Pt_3(j+1)=Pt_4(j+1)/sigma_comb;
154 PI_C(j+1)=Pt_3(j+1)/Pt_2(j);
155 %-----TURBINE-----
156 PI_t(j)=-4.5415e-029*N(j)^6+1.7103e-023*N(j)^5-2.6255e-018*N(j)
      ^4+2.1127e-013*N(j)^3-9.4373e-009*N(j)^2+2.1571e-004*N(j)
      -1.0868;
157 Tao_c(j)=Tt_3(j)/Tt_2(j);
158 Teta_2(j)=1;
159 Teta_3(j)=Tt_4(j)/Tt_2(j);
160 Tao_t(j)=1-((Cp_a(j)*Teta_2(j))/(Cp_g(j)*(1+f(j))*Teta_3(j)))*(
      Tao_c(j)-1);
161 Tt_5(j)=685;
162 %-----Work Balance-----
163 Pt(j)=((mc_dot(j)+ mf_dot(j))*(Tt_4(j)-Tt_5(j))*Cp_g(j));
164 Pc(j)=mc_dot(j)*Cp_a(j)*(Tt_3(j)-Tt_2(j));
165 Pf(j)=500;
166 if Pf(j)<0
167     Pf(j)=0;
168 end
169 W(j)=N(j)*pi/30;
170 W(j+1)=sqrt((2*(Pt(j)-Pc(j)-Pf(j))/I)*h+W(j)^2);
171 N(j+1)=W(j+1)*30/pi;
172 %-----THRUST-----
173 Rho_0(j)=-1.4267e-029*N(j)^6 + 5.8500e-024*N(j)^5 - 9.7835e-019*N(j)
      ^4 + 8.5570e-014*N(j)^3 - 4.1601e-009*N(j)^2 + 1.0194e-004*N(j)
      + 1.9642e-001;
174 m9_dot(j)=mc_dot(j)+ mf_dot(j);
175 T_9(j)=8.6554e-026*N(j)^6 - 3.4961e-020*N(j)^5 + 5.8197e-015*N(j)^4
      - 5.0939e-010*N(j)^3 + 2.4663e-005*N(j)^2 - 6.2630e-001*N(j) +
      7.2095e+003;
176 V9(j)=(gama_t*R*T_9(j))^0.5;
177 V0(j)=mc_dot(j)/(Rho_0(j)*A_0);
178 F(j)=m9_dot(j)*V9(j)-mc_dot(j)*V0(j);
179 end
180 % Plot results
181 subplot(1,3,1); plot(N); grid on; xlabel('t (ms)'); ylabel('N (rpm)')
182 subplot(1,3,2); plot(Tt_4); grid on; xlabel('t(ms)'); ylabel('Tt_4(K)')
183 subplot(1,3,3); plot(F); grid on; xlabel('t (ms)'); ylabel('Thrust(N)')

```

MATLAB Code for Fuzzy Logic Control

Click here to open the MATLAB code: [fuzzy_logic.m](#)

References

1. Al-Hamdan, Qusai Z., and Munzer S.Y. Ebaid. "Modeling and Simulation of a Gas Turbine Engine for Power Generation." *Journal of Engineering for Gas Turbine and Power* 128: 302-311, APRIL 2006
2. Camporeale, S.M., B. Fortunato, and M. Mastrovito. "A Modular Code for Real Time Dynamic Simulation of Gas Turbines in Simulink." *Journal of Engineering for Gas Turbines and Power* 128: 506-517, JULY 2006
3. Korakianitis, T., J.I. Hochstein, and D. Zou. "Prediction of the Transient Thermodynamic Response of a Closed-Cycle Regenerative Gas Turbine." *Journal of Engineering for Gas Turbines and Power* 127: 57-64, JANUARY 2005.
4. Camporeale, S.M., B. Fortunato, and A. Dumas. "Non-Linear Simulation Model and Multivariable Control of a Regenerative Single Shaft Gas Turbine." *IEEE International Conference on Control Applications*. Hartford, CT,. 721-723, 1997.
5. Tang, W., Wang, L., Gu, J., & Gu, Y. (2020). Single neural adaptive PID control for small UAV micro-turbojet engine. *Sensors*, 20(2), 345.
6. Official Matlab Website