

AN ELECTRONIC CONTROL UNIT DESIGN FOR A MINIATURE JET ENGINE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CUMA POLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2009

Approval of the thesis:

**AN ELECTRONIC CONTROL UNIT DESIGN FOR A
MINIATURE JET ENGINE**

submitted by **CUMA POLAT** in partial fulfillment of the requirements for the degree of
Master of Science Mechanical Engineering, Middle East Technical University by,

Prof Dr. Canan ÖZGEN
Dean, Graduate School of **Natural and Applied Sciences**

Prof Dr. Süha ORAL
Head of Department, **Mechanical Engineering**

Asst. Prof. Dr. Melik DÖLEN
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr.Kahraman ALBAYRAK
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Melik DÖLEN
Mechanical Engineering Dept., METU

Prof. Dr. Nafiz ALEMDAROĞLU
Aerospace Engineering Dept., METU

Asst. Prof. Dr. Buğra KOKU
Mechanical Engineering Dept., METU

Asst. Prof. Dr. A. İlhan KONUKSEVEN
Mechanical Engineering Dept., METU

Date:

15.12.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Cuma POLAT

Signature :

ABSTRACT

AN ELECTRONIC CONTROL UNIT DESIGN FOR A MINIATURE JET ENGINE

POLAT, Cuma

MS., Department of Mechanical Engineering

Supervisor: Asst. Prof. Dr. Melik DÖLEN

December 2009, 139 pages

Gas turbines are widely used as power sources in many industrial and transportation applications. This kind of engine is the most preferred prime movers in aircrafts, power plants and some marine vehicles. They have different configurations according to their mechanical constructions such as turbo-prop, turbo-shaft, turbojet, etc. These engines have different efficiencies and specifications and some advantages and disadvantages compared to Otto-Cycle engines. In this thesis, a small turbojet engine is investigated in order to find different control algorithms.

AMT Olympus HP small turbojet engine has been used to determine the mathematical model of a gas turbine engine. Some important experimental data were taken from AMT Olympus engine by making many experiments. All components of the engine have been modeled by

using laws of thermodynamics and some arithmetic calculations such as numerical solution of nonlinear differential equations, digitizing compressor and turbine map etc. This mathematical model is employed to create control algorithm of the engine. At first, standard control strategies had been considered such as P (proportional), PI (proportional integral), and PID (proportional-integral-differential) controllers. Because of the nonlinearities in gas turbines, standard control algorithms are not commonly used in literature. At the second stage fuzzy logic controllers have been designed to control the engine efficiently. This control algorithm was combined with mathematical of the engine in MATLAB environment and input-output relations were investigated. Finally, fuzzy logic control algorithm was embedded into an electronic controller.

Keywords: Turbojet, Modeling, Control, Fuzzy Logic

ÖZ

KÜÇÜK BİR JET MOTORU İÇİN ELEKTRONİK KONTROL ÜNİTESİ TASARIMI

POLAT, Cuma

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Melik DÖLEN

Aralık 2009, 139 sayfa

Günümüzde küçük gaz türbin motorları, model uçaklarda, elektrik üretim tesislerinde, insansız hava araçlarında, dronlarda ve benzeri hava platformlarında sıklıkla kullanılmaktadır. Bu motorlar turbo-prop, turbo-şaft, turbo jet gibi farklı kombinasyonlarda kullanılabilir. Gaz türbin motorları güç/boyut ilişkisi dikkate alındığında oldukça büyük avantajlara sahiptir. Yüksek yakıt tüketimi ise motorların dezavantajları arasında sayılabilir. Ülkemizde de son zamanlarda farklı alanlarda kullanılmaya başlanan bu motorların verimli ve güvenli bir şekilde kontrol edilebilmesi çalışmamızın ana konusunu oluşturmaktadır.

Bu çalışmamızda öncelikle AMT Olympus HP gaz türbin motoru için matematik model oluşturulmuştur. Kodun oluşturulması aşamasında gereken teknik veriler ve test değerleri yine benzer motor ile yapılan testlerden elde edilmiştir. Her bir parça modellenerek motorun MATLAB ortamında termodinamik yasalar kullanılarak tüm modeli oluşturulmuştur.

Oluřturulan matematik model kullanılarak motorun kontrol algoritması geliřtirilmiř. Bu ařamada oransal, oransal-tümlevsel ve oransal-tümlevsel-türevsel kontrolcüler tasarlanmaya çalıřılmıř fakat motorun dođrusal olmayan yapısı göz önünde bulundurularak bulanık mantık algoritması oluřturulmuřtur. Tasarlanan bulanık mantık algoritması motorun matematik modeli ile birleřtirilerek çeřitli girdilere karřılık motorun davranıřı incelenmiřtir. Son ařamada ise oluřturulan bulanık mantık algoritması dsPIC30F4013 mikroilemcisine gömölerek motorun kontrolü sađlanmıřtır.

Anahtar Kelimeler: Turbojet, Modelleme, Kontrol, Bulanık Mantık

ACKNOWLEDGMENTS

I am deeply grateful to my thesis Supervisor Asst. Prof. Dr. Melik Dölen his guidance, advice, criticism, encouragement and insight throughout the research. I would also like to thank to Prof. Dr. Nafiz Alemdarođlu for his suggestions, comments, helps. Special thanks to my friend Çađlar Atalayer and Murat Ceylan for their assistance on my experiments and study.

I am grateful to my family, İlyas Polat, Fikriye Polat and Dilek Polat and to my love Bahar Sarıkaya for their endless love, support, trust and encouragement. This work has been supported by METU Microjet Project directed by Prof. Dr. Nafiz Alemdarođlu in Aerospace Engineering Department.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ.....	vi
ACKNOWLEDGMENTS	viii
TABLE OF CONTENTS.....	ix
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF SYMBOLS	xvi
CHAPTERS	
1. INTRODUCTION	1
2. REVIEW OF THE STATE OF THE ART	4
2.1 Introduction.....	4
2.2 Gas Turbine Engine Control	4
2.3 Engine Control Programs.....	5
2.4 Mathematical Modeling	7
2.4.1 Mathematical Modeling of Gas Turbine Engines	7
2.4.1.1 Mathematical Model of the Compressor.....	9
2.4.1.2 Mathematical Model of the Combustion Chamber	11
2.4.1.3 Mathematical Modeling of the Turbine	12
2.4.1.4 Gas Turbine Computer Simulation	15
2.4.1.5 Simulation Results	17

2.4.2	Some Important Studies about Modeling, Simulation and Control of Gas Turbine Engines	19
2.4.3	Outline of the Current Work	21
3.	MATHEMATICAL MODELING OF A SMALL TURBOJET ENGINE.....	22
3.1	Introduction.....	22
3.2	Mathematical Modeling	23
3.2.1	Component Models	25
3.2.1.1	Assumptions.....	25
3.2.1.2	Compressor	26
3.2.1.3	Combustion Chamber	31
3.2.1.4	Turbine	34
3.2.1.5	Work Balance.....	36
3.2.1.6	Thrust	38
3.2.2	Simulation Code.....	38
3.3	Closure	45
4.	CONTROL OF A SMALL TURBOJET ENGINE	46
4.1	Introduction.....	46
4.2	PI Controller Design	46
4.3	Fuzzy Logic	49
4.3.1	Fuzzy Rules.....	51
4.3.2	Membership Functions.....	53
4.3.3	Defuzzification.....	55
4.3.4	Fuzzy Logic Simulation.....	56
4.3.5	Comparing PID and Fuzzy Logic Controller for a Second Order System. 58	
4.3.5.1	Overdamped System with $\zeta=1.5$, $\omega_n = 10$ [rad/s].....	59
4.3.5.2	Underdamped System with $\zeta = 0.1$, $\omega_n = 10$ [rad/s].....	60
4.3.5.3	Underdamped System with $\zeta=0.6$, $\omega_n = 1$ [rad/s].....	63

4.4	Closure	66
5.	DESIGN AND IMPLEMENTATION OF ELECTRONIC CONTROL UNIT	67
5.1	Introduction.....	67
5.2	Electronic Control Unit of AMT Olympus HP	67
5.3	Circuit Design	70
5.3.1	Main Circuit	71
5.3.2	Temperature Measurement	72
5.3.3	Kerosene Pump and Glow Plug Driver.....	73
5.3.4	Kerosene and Propane Valve Drivers	75
5.3.5	Other Standard Elements	75
5.4	Closure	80
6.	ECU TESTS AND RESULTS.....	81
6.1	Introduction.....	81
6.2	Test on ECU Sub-Components	81
6.3	Firmware Tests.....	83
6.4	ECU Test Results.....	85
6.5	Original ECU Test Results.....	89
6.6	Closure	98
7.	CONCLUSION AND FUTURE WORK	99
7.1	Introduction.....	99
7.2	Conclusion and Future Work	99
	REFERENCES	103
	APPENDICES	
	A. MATLAB CODES	106
	B. MICROCONTROLLER PROGRAMS	130

LIST OF TABLES

Table 2-1 Dimensionless Compressor Parameters [1]	11
Table 2-2 Dimensionless Turbine Parameters [1].....	12
Table 3-1 Compressor Pressure Ratio Values from Beta Lines.....	30
Table 3-2 Compressor Mass Flow Rate Values from Beta Lines	30
Table 3-3 Compressor Efficiency Values from Beta Lines	30
Table 3-4 Comparison between Code and Test Results.....	45
Table 4-1 Fuzzy Rule Table.....	53
Table 4-2 Comparison of PI and Fuzzy Logic for the Same Input at Steady State	58
Table 5-1 Inputs and Outputs of the Electronic Control Unit.....	68

LIST OF FIGURES

Figure 2-1 Acceleration and Deceleration Fuel Schedules [3]	7
Figure 2-2 Main Parts of the Gas Turbine [5].....	9
Figure 2-3 General Compressor Characteristic Map [2].....	9
Figure 2-4 Turbine Characteristic Map [1]	13
Figure 2-5 Computer Simulation Flowchart [1]	16
Figure 2-6 Variation of thermal efficiency and specific fuel consumption at	17
Figure 2-7 Dynamic Response of the Number of Revolution [5, 15].....	18
Figure 2-8 Dynamic Responses of the Total Temperatures [5, 15]	18
Figure 3-1 AMT Olympus HP Turbojet Engine	22
Figure 3-2 Gas Turbine Application Areas [24, 25]	23
Figure 3-3 AMT Olympus HP Components	24
Figure 3-4 Radial Compressor	27
Figure 3-5 Problems with Reading Compressor Maps [1].....	28
Figure 3-6 Compressor Map [32].....	29
Figure 3-7 Combustion Chamber of AMT Olympus HP.....	31
Figure 3-8 Combustion Chamber Volume.....	33
Figure 3-9 Turbine of AMT Olympus HP	35
Figure 3-10 AMT Olympus HP Turbine Map	36
Figure 3-11 AMT OLYMPUS HP Engine's Shaft	37
Figure 3-12 AMT Olympus HP Test Stand	39
Figure 3-13 Measurement Points	40
Figure 3-14 AMT Olympus HP Turbojet Simulation Inputs/Outputs	41
Figure 3-15 Flowchart of the AMT Olympus HP Turbojet Engine	42
Figure 3-16 Shaft Speed Value Versus Time.....	43
Figure 3-17 Total Temperature after Combustion Chamber.....	44
Figure 3-18 Thrust Output	44
Figure 4-1 Schematic Representation of PI Controller	47
Figure 4-2 Shaft Speed, Tt4 and Thrust Output Graphs	48

Figure 4-3 Shaft Speed, Tt4 and Thrust Output Graphs for Different Initial Conditions	49
Figure 4-4 Fuzzy Logic Controller [27].....	50
Figure 4-5 Definition of Middle Age in Conventional Sets and Fuzzy Sets.....	50
Figure 4-6 Relation between the Shaft Speed and Pump Voltage of the AMT Olympus HP Turbojet Engine	52
Figure 4-7 Triangular, Trapezoidal and Gaussian Membership Function	54
Figure 4-8 Membership Function of the Error of the Shaft Speed	54
Figure 4-9 Membership Function of the Change in Error of the Shaft Speed	54
Figure 4-10 Membership Function of the Output	55
Figure 4-11 Fuzzy Logic Controller Block Diagram.....	55
Figure 4-12 Shaft Speed Output.....	56
Figure 4-13 Tt4 Output	57
Figure 4-14 Thrust Output	57
Figure 4-15 Overdamped System Results for PID and FLC ($\zeta=1.5$, $\omega_n =10$).....	60
Figure 4-16 Underdamped System Results for PID and FLC ($\zeta=0.1$, $\omega_n =10$ [rad/s])	62
Figure 4-17 Underdamped System Results for PID and FLC ($\zeta=0.1$, $\omega_n =10$ [rad/s], $K_i=2$).	63
Figure 4-18 Underdamped System Results for PID and FLC ($\zeta=0.6$, $\omega_n =1$ [rad/s])	64
Figure 4-19 Underdamped System Results for PID and FLC ($\zeta=0.6$, $\omega_n =1$ [rad/s], $K_p=1$, $K_d=2$, $K_i=2$).....	65
Figure 5-1 AMT Olympus HP Engine ECU	68
Figure 5-2 Schematic Diagram of the AMT Olympus HP	70
Figure 5-3 Main Circuit	72
Figure 5-4 Temperature Circuit	73
Figure 5-5 L298 Circuit	74
Figure 5-6 Revised Glow Plug Circuit Design	74
Figure 5-7 Kerosene and Propane Valve Circuit	75
Figure 5-8 RS 232 Circuit.....	76
Figure 5-9 Shaft Speed Input Circuit	76
Figure 5-10 Analog Input Circuit.....	77
Figure 5-11 Starter Circuit	78
Figure 5-12 3D View of the Controller (Beta Version)	78
Figure 5-13 ECU Circuit.....	79
Figure 5-14 Top-, Bottom, and the Silk Layer of the Circuit	80

Figure 6-1 AMT Olympus HP Test Stand and Designed ECU	82
Figure 6-2 Automatic Engine Cooling and Propane Gas Ignition Algorithm.....	84
Figure 6-3 ECU Test with AMT Olympus Hp Turbojet Engine (Test 1).....	85
Figure 6-4 ECU Test with AMT Olympus Hp Turbojet Engine (Test 2).....	86
Figure 6-5 ECU Test with AMT Olympus Hp Turbojet Engine (Test 3).....	86
Figure 6-6 ECU Test with AMT Olympus Hp Turbojet Engine (Test 4).....	87
Figure 6-7 ECU Test with AMT Olympus Hp Turbojet Engine (Test 5).....	87
Figure 6-8 ECU Test with AMT Olympus Hp Turbojet Engine (Test 6).....	88
Figure 6-9 ECU Test with AMT Olympus Hp Turbojet Engine (Test 7).....	88
Figure 6-10 Original ECU Test Results (Test 1)	90
Figure 6-11 Original ECU Test Results (Test 2)	91
Figure 6-12 Original ECU Test Results (Test 3)	92
Figure 6-13 Original ECU Test Results (Test 4)	93
Figure 6-14 Original ECU Test Results (Test 5)	94
Figure 6-15 Pump Voltage and Shaft Speed Graphs for Automatic Start Operation (Test 2)	95
Figure 6-16 Shaft Speed vs. Pump Voltage for Test 1.....	96
Figure 6-17 Shaft Speed vs. Pump Voltage for Test 4.....	96
Figure 6-18 EGT vs. Pump Voltage for Test 1	97
Figure 6-19 EGT vs. Pump Voltage for Test 4.....	97

LIST OF SYMBOLS

F_{ef}	Effective Thrust
H	Altitude
M	Mach number
\dot{m}	Mass flow rate
C_n	Final air velocity of the air
U	Air speed
ΔE	Kinetic energy of the unit mass of the air
τ	Ratio of stagnation temperatures
d	Diameter
P	Pressure
η	Efficiency
N	Shaft Speed
T	Temperature
C_p	Specific heat at constant pressure
C_v	Specific heat at constant volume
γ	Ratio of specific heat
θ	Ratio between stagnation temperature and ambient static temperature
δ	Ratio between stagnation pressure and ambient static pressure

W	Power
m	mass
ξ_{cc}	Stagnation pressure drop
f	fuel / air ratio
LHV	Lower heating value
h	Specific enthalpy
u	Specific internal energy
Q_f	Lower thermal value of the fuel
σ	Constant pressure loss coefficient
R	Gas constant
U	Energy storage effect
π	Pressure ratio
β	Beta line
Q	Heat flux
ρ	Density
V	Volume
T	Torque
I	Inertia
ω	Angular velocity
α	Angular acceleration
t	Time
K_p	Proportional constant

K_I	Integral constant
K_d	Differential constant
ω_n	Natural frequency
ζ	Damping ratio

CHAPTER 1

INTRODUCTION

The aim of this thesis work is to design and build up an electronic control unit for a small turbojet engine. Gas turbines have various application areas as prime movers in planes, in power plants for electricity generation and in naval vessels for propulsion [3, 5, 15]. Gas turbine engine design and manufacturing process has its origins back to mid 1940's. Because of their broad application fields, engineers tried to design more efficient, powerful, economic and reliable gas turbines. Nowadays, almost all aircrafts are powered by gas turbines, which have different configurations such as turbo-fan, turbo-prop, turbo-shaft, and turbo-jet.

In the early years of 1930's, gas turbine improvement efforts were started simultaneously by a team directed by Frank Whittle in England and another group headed by Hans von Ohain and Max Hahn. Although engineers decided to utilize from gas turbines only for transportation systems at the beginning, some decades later, they started to design new configurations for oil or gas compression, power generation, stand-by power unit etc. Increasing demands for gas turbines in many fields have caused different customer needs. Especially more efficient and reliable engines are the most important desirable features in industry. Although gas turbines have some disadvantages such as high fuel consumption rate, high technology production techniques, high technology materials usage, engineers are still working to increase their efficiency. Gas turbines are very important power generator alternatives for today's airplanes. For example, jet fighters utilize small gas turbines as a main engine starter and auxiliary power sources. Small gas turbines are generally composed of a single stage radial compressor, a diffuser, a combustion chamber, a single stage axial turbine and a nozzle. These different structures are combined by complicated aerothermodynamic rules.

Another important application field for gas turbines is distributed power generation. Fully automatic startup, fast response capability, dimension / power relation and different fuel options make it more preferable than other power generators. Recently, increasing power demands in big cities has caused economic and physical difficulties for transportation of the electricity so designers investigate alternative energy sources. Consumers have started to prefer more economic and uninterruptable power supplies for their house or plants. Therefore, in the last decades, people have begun to utilize distributed generation method in many fields. Distributed generation means, consumers can provide their own energy from the same building or same area where they live or work. For this purpose, engineers designed micro-turbines that can produce 25-300 kW power while their shaft speeds are almost 50-120 [krpm]. Not only micro-turbines have some instability due to their small inertia but also they have some vital problems while the electricity cut off or instantaneous load reductions. High rotational speed of the shaft may be critical for rotating parts such as turbine, compressor, bearings, and shaft. Because of these reasons, mathematical modeling of these engines is very important to understand their behavior when they are subjected to big and instantaneous disturbance loads.

In this study, a MATLAB code for mathematical modeling of AMT Olympus HP small turbojet engine is presented. Code is composed of both aero thermal and nonlinear ordinary differential equations. It's intended to analyze transient response of small gas turbine engines at off-design points and to maintain a starting point for an appropriate engine control unit design. Code is written in MATLAB and results are verified with engine performance and transient test data. AMT Olympus HP is a 250 N of thrust level engine whose main application areas are UAV platforms and model aircrafts was utilized in those experiments for the engine transient performance analysis. AMT Olympus Engine configuration is composed of single stage radial compressor, annular type combustion chamber and a single stage axial turbine. An electric starter engine is used for the startup and kerosene is utilized as fuel which is vaporized by burning of propane gas. Engine maximum rotating speed is 108 [krpm].

Modeling code is utilized not only for one operating point - on-design point – on operating line but also for other operating points known as off-design points on the operating line. Because of this reason, code is suitable for investigating the behavior of engines at all points on their operating lines. In this study, operating line of the AMT Olympus HP is taken from experimental data.

Mathematical modeling, transient behavior analysis of a small gas turbine engine and electronic controller design are the focus of this study. Transient intervals are the operating regions where the most critical conditions arise in gas turbine operations [2, 6, 9]. Gas turbine components are exposed to extreme loading cases during these transient conditions. Exceeding critical shaft rotating speed and critical turbine inlet – outlet temperatures occur frequently in these operating regions. Because of these reasons, transient response analyses of gas turbine engines are crucial to improve their stability and lifetime. For this purpose, dynamic modeling of gas turbine engine should be constructed and its responses to different operating conditions with different fuel inputs should be analyzed. Various aero-thermal and differential equations are used for transient analyses [2, 5, 15, 4, 6]. The resulting differential equations are non-linear in nature and can be solved simultaneously with aero-thermal equations.

The organization of this thesis is as follows: Chapter 2 describes the state of the art in the corresponding fields. In Chapter 3, mathematical modeling of a small turbojet engine was investigated. Mechanical parts such as turbine, compressor, etc. were modeled by using laws of thermodynamics. Different techniques for non-linear equations are investigated such as linearization, piece-wise linear modeling, and nonlinear modeling. At each step, equations are solved algebraically and Runge-Kutta method is utilized to calculate the behavior at next data step. Results are compared and verified with experimental data. This mathematical modeling was used to design different control algorithm in MATLAB. In Chapter 4, control of a small turbojet engine was discussed. Controller design and electronic control unit were searched. In this chapter, designed control algorithm was embedded in to the digital controller by using C compiler. At last, a small gas turbine controller board was designed by using suitable programs. This circuit was employed in experiments for correcting our control algorithm.

CHAPTER 2

REVIEW OF THE STATE OF THE ART

2.1 Introduction

Gas turbines need sophisticated control systems because they commonly operate at temperature and shaft speed, which are very close to their thermal and mechanical limits. Furthermore, especially in aircraft gas turbines, operating conditions may vary in very large range. As a result, the number of the parameters, which the control system has to manipulate, increases [14].

Mathematic modeling of gas turbine engines is very important for designers. They may utilize mathematic models to find engine control algorithm or its critical behaviors. Mathematical modeling is more economic and faster technique rather than experimental methods to find engine transients [1, 3]. Because of these reasons gas turbine controller designers have to model and analyze their engine to build up an effective controller.

There are many different modeling techniques in literature such as linear, nonlinear, and piecewise linear, etc. In this section engine control parameters, main control aims and some advantages and disadvantages of different modeling techniques in literature are investigated.

2.2 Gas Turbine Engine Control

The first objective of a gas turbine used in aircraft is to obtain a reference value of effective thrust F_{ef} . Effective thrust depends on flight parameters altitude H , Mach Number M , operating conditions of the engine and nozzle [19].

Aero derivative power plants based on gas turbines, which produce effective thrust from gas flow. Effective thrust obtained from gas turbine may be represented by

$$F = \dot{m}_a \cdot (U_n - U_f) \quad (2.1)$$

where,

- \dot{m}_a : is the mass flow rate of the air
- U_n : is the final air velocity
- U_f : is the air speed with respect to the aircraft

Kinetic energy of unit mass of air is given by

$$\Delta E = \frac{U_n^2}{2} - \frac{U_f^2}{2} \quad (2.2)$$

The additional energy is obtained from conversion of the chemical energy of the fuel to thermal energy and into mechanical energy by using the thermodynamic cycle. All power plants have two main tasks;

- Converts the chemical energy of the fuel into mechanical energy
- Uses this mechanical energy to increase the acceleration of the air flows into the engine

Any gas turbines have two important parts given in the following:

- Compressor, compresses the air
- Turbine, drives the compressor shaft

The main tasks of the control system in aircrafts may change with the flight conditions. For example, engine needs maximum thrust for take-off, maximum efficiency for cruise flight or minimum specific fuel consumption. Design of aircraft gas turbine controller requires gas turbine theory, control theory and differential calculus theory [19].

2.3 Engine Control Programs

Engine control programs are divided to adjust control parameters for different engine operating conditions. For example, engine control program utilizes different algorithms for

start-up, transient operation, maximum thrust, steady state operation, minimum thrust and afterburner operation. On the other hand, maximum maneuverability, maximum efficiency, landing, or take of tasks are very important for the control programs. Each program is determined for optimization of some criterion or some group of criterion. Aircraft gas turbine engines' control systems have to ensure some characteristics such as;

- Optimal thrust and maximum efficiency at demanded conditions,
- Obtaining some requirements for transient operation and control accuracy,
- Prevent the system from the excessive thermal and mechanical loads.

Today's modern gas turbine engines have minimum fifteen or more control points. For example, variable stator vanes of fan, fuel flow into the combustion chamber, fuel flow into the afterburner are some control points [19]. To control these control variables, automatic control system has to measure some values such as inlet temperature, shaft speed, inlet pressure, etc. [18]. During design of the gas turbine automatic control system, control programs in the control device drives the engine for maximum efficiency and accuracy. Complexity of the gas turbine engine control systems and increase of aircraft requirements have resulted to use on-board computers in automatic control systems instead of hydraulic or pneumatic parts [18, 19]. Computational capabilities of on-board computers enable optimization of control programs accounting for flight conditions and characteristics of the engine itself [19].

Communication between the pilot and the engine is obtained by adjusting power lever, which gives fuel flow input to the engine. However, the pilot does not give the input directly from lever. First, power lever input reaches to control system and then control system chooses the best manipulation algorithm for that condition. At last, controller manipulates the fuel flow into the combustion chamber or into the afterburner [18]. An ideal control programs meet the acceleration and deceleration fuel schedules, which are given in Figure 2-1. These are basic control requirements for maintaining the gas turbine to operate within the safe margins during the steady state and transient operations. A standard acceleration control allows the engine to accelerate at the maximum acceleration without the engine being driven into the surge region or overheating the engine components. Similarly, the fuel supply can be reduced without causing flameout problem [3] to a minimum deceleration level.

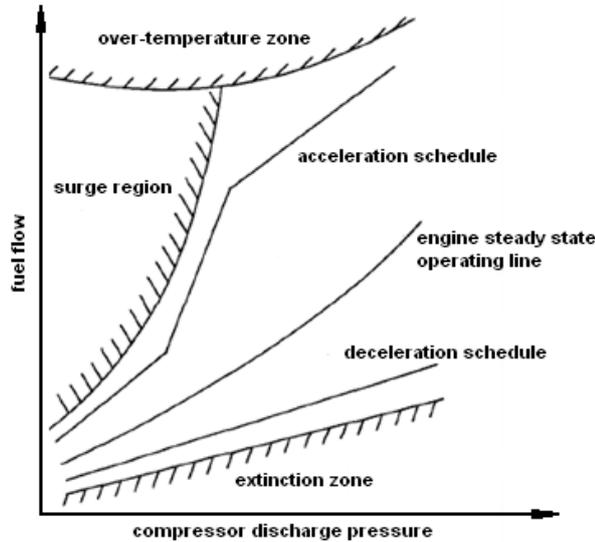


Figure 2-1 Acceleration and Deceleration Fuel Schedules [3]

2.4 Mathematical Modeling

There are various techniques available in literature for gas turbine modeling and simulation. Especially linear, nonlinear and piecewise linear modeling methods are discussed in several studies. Mathematical modeling of a gas turbine is significant for the controller plan. Designers need to have a mathematical model to investigate efficient control algorithm such as P, PD, PID or fuzzy logic controllers. In this section, some important studies will be investigated that are used in this thesis work. First of all the most important studies were discussed in detail and then other works were summarized.

2.4.1 Mathematical Modeling of Gas Turbine Engines

Gas turbine engines are widely used prime movers in transportation system such as aircrafts and cars. Besides of these areas, gas turbines can be utilized for power generation in power plants, gas or liquid compression, pump drivers, aircrafts and etc. Investigation of steady-state behavior of gas turbine in engineering is more preferred. This models based on components' static characteristics. Detailed steady-state discussion is crucial to develop

engine dynamic behavior [1, 5, 15]. Gas turbine engine is a complex combination of aerothermodynamics and mechanical rules. The most important case for gas turbine control occurs at off-design points of the engines. Off-design point means, the points, which are not on the operating line of the engine. At first stage of design processes, designers preferred experimental methods to found steady state or transient behavior of their gas turbines. Nevertheless, this method was time consuming and expensive. Therefore, they tried to model their engine by using mathematical and thermodynamic equations [1, 2, 6, 18]. By choosing engine configuration differently such as, single shaft, twin shaft, triple shaft, they may be used in different areas. They studied a gas turbine produced for power generation application in [1], which consists of the following components;

- Inlet,
- Compressor,
- Combustion chamber,
- Turbine,
- Nozzle,
- Auxiliary components like pumps, valves, etc.

Figure 2-2 shows main parts of a standard gas turbine engine. All components' mathematical models were developed by using physical rules and empirical data. Thermodynamic properties of combustion gases and air were calculated using variation of temperature [1, 2, 4]. The working procedure of gas turbine engines is the same and given in the following:

- Air is taken into the engine through the inlet duct by the compressor.
- Compressor compresses the air and delivers it to the combustion chamber.
- In combustion chamber, the fuel is added to the air and the mixture is ignited.
- After ignition, combustion processes occurs and the temperature of the mixture increases while pressure remains almost constant.
- High temperature combustion products go through the turbine and it turns the turbine blades. Produced torque is delivered to the compressor again.
- Combustion products go out form the nozzle.

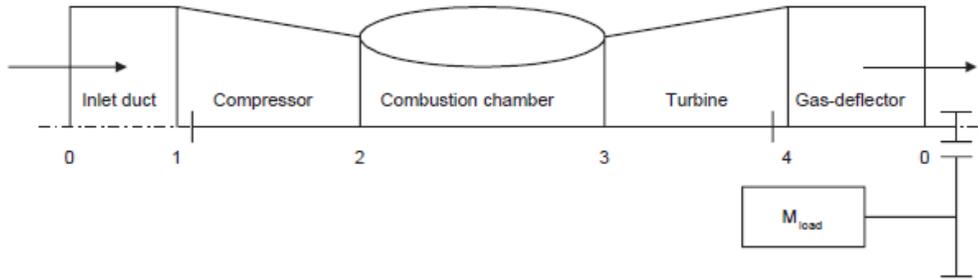


Figure 2-2 Main Parts of the Gas Turbine [5]

2.4.1.1 Mathematical Model of the Compressor

Mathematical modeling and analysis of the compressor of gas turbine used in [1] was determined by dimensionless parameters. The dimensionless parameters are the same for all system of units. Compressor performance, sometimes called compressor map shown in Figure 2-3. These maps in general, were drawn by using experimental data or geometric specifications of compressors.

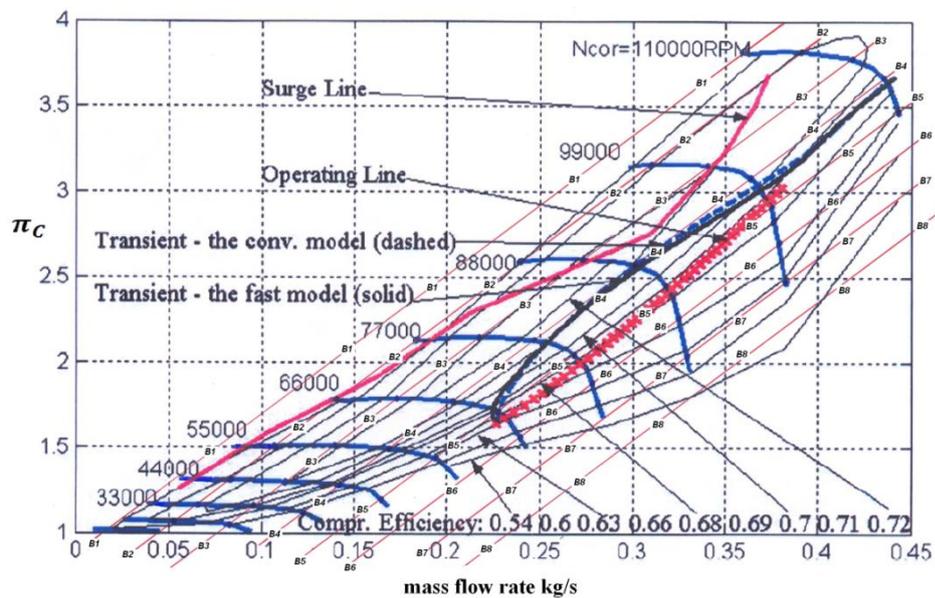


Figure 2-3 General Compressor Characteristic Map [2]

Al-Hamdan and Y. Ebaid presented compressor characteristic equations with dimensionless parameters. Eq. (2.3) is in complete dimensionless form, and Eq. (2.4) is in general form.

$$\frac{\tau_c}{d_2^2 \cdot P_{01}} = \frac{1}{2\pi} \cdot \frac{1}{\eta_c} \cdot \left(\frac{d_2 \cdot N}{\sqrt{C_{Pa} \cdot T_{01}}} \right)^{-1} \cdot \frac{\dot{m}_a \sqrt{C_{Pa} \cdot T_{01}}}{d_2^2 \cdot P_{01}} \cdot \left[\left(\frac{P_{02}}{P_{01}} \right)^{(\gamma_a - 1)/\gamma_a} - 1 \right] \quad (2.3)$$

$$\frac{\tau_c}{\delta} = f \left(\eta_c, \frac{N}{\sqrt{\theta}}, \frac{\dot{m}_a \sqrt{\theta}}{\delta}, \frac{P_{02}}{P_{01}} \right) \quad (2.4)$$

Compression power is given by

$$W_c = d_2^2 \cdot P_{01} \cdot \sqrt{C_{Pa} \cdot T_{01}} \cdot \frac{\dot{m}_a \sqrt{C_{Pa} \cdot T_{01}}}{d_2^2 \cdot P_{01}} \cdot \frac{1}{\eta_c} \cdot \left[\left(\frac{P_{02}}{P_{01}} \right)^{(\gamma_a - 1)/\gamma_a} - 1 \right] \quad (2.5)$$

Using the compressor map, if any two parameters are known then the rest of the parameters can be determined [1, 4]. The final stagnation temperature in the compressor can be calculated as follow:

$$T_{02} = T_{01} + \frac{T_{01}}{\eta_c} \cdot \left[\left(\frac{P_{02}}{P_{01}} \right)^{(\gamma_a - 1)/\gamma_a} - 1 \right] \quad (2.6)$$

Dimensionless compressor parameters are given in Table 2-1 utilized in [1]. To solve Eqs. (2.3) - (2.6) required input data can be taken from the compressor characteristic map. This data reading operation was made by special interpolation technique will be discussed in the following paragraphs. Eq. (2.7) shows compressor output conditions and Eq. (2.8) shows temperature in the middle of the compressor for steady state conditions [4, 5, 8, 15, 19].

$$\left(\frac{T_{T2}}{T_{T1}} \right) = \left(\frac{P_{T2}}{P_{T1}} \right)^{[R/(C_p \cdot \eta_c)]} \quad (2.7)$$

$$\frac{dT_m}{dt} = \frac{1}{m \cdot c_v} \cdot \left[(\dot{m} \cdot c_p \cdot T)_{in} - (\dot{m} \cdot c_p \cdot T)_{out} + \dot{W}_c - \dot{W}_t - c_v \cdot T_m \cdot \frac{dm}{dt} \right] \quad (2.8)$$

Table 2-1 Dimensionless Compressor Parameters [1]

PARAMETER	MEANINGS
$\left(\frac{\dot{m}_a \cdot \sqrt{C_{Pa} \cdot T_{01}}}{d_2^2 \cdot P_{01}}\right), \left(\frac{\dot{m}_a \cdot \sqrt{\theta}}{\delta}\right)$	Compressor dimensionless mass flow parameters
$\left(\frac{d_2 \cdot N}{\sqrt{C_{Pa} \cdot T_{01}}}\right), \left(\frac{N}{\sqrt{\theta}}\right)$	Compressor dimensionless speed parameters
$\left(\frac{\tau_c}{d_2^2 \cdot P_{01}}\right), \left(\frac{\tau_c}{\delta}\right)$	Compressor dimensionless torque parameters
$\left(\frac{P_{02}}{P_{01}}\right)$	Compressor dimensionless pressure ratio parameters
$\left[\frac{(P_{02}/P_{01})^{(\gamma_a-1)/\gamma_a}}{(T_{02}/T_{01}-1)}\right]$	Isentropic compressor efficiency
$\left(\theta = \frac{T_{01}}{T_{0,ref}}\right)$	Dimensionless temperature parameter
$\left(\delta = \frac{P_{01}}{P_{0,ref}}\right)$	Dimensionless pressure parameter

2.4.1.2 Mathematical Model of the Combustion Chamber

Al-Hamdan and Y. Ebaid discussed about the modeling and analysis of combustion chamber in their work. Combustion chamber performance was normally given by its efficiency η_{cc} and stagnation pressure drop ξ_{cc} during the combustion process. Using these parameters, fuel / air ratio f and stagnation pressure at the exit of the combustion chamber P_{03} were determined from Eq. 2.9 and 2.10 respectively [1].

$$f = \frac{1}{\frac{\eta_{cc} \cdot (LHV)}{C_{Pg} \cdot (T_{03} - T_{02})} - 1} \quad (2.9)$$

$$P_{03} = (1 - \xi_{cc}) \cdot P_{02} \quad (2.10)$$

In their work, S. M. Camporeale et al modeled the combustion chamber as a pure energy accumulator. They assumed that the inside temperature and pressure were equal to the combustor respective outlet values. They obtained the equation below from the unsteady energy conservation;

$$\frac{d(m_{cc} \cdot u_{cc})}{dt} = \dot{m}_{in} \cdot h_{in} + \dot{m}(h_b + \eta_b \cdot LHV) - \dot{m}_{out} \cdot h_{out} \quad (2.11)$$

where m_{cc} and u_{cc} are the mass inside the burner and the internal specific energy respectively. LHV is lower heating value of the fuel. P.Ailer et al derived Eqs. (2.12) and (2.13) form the conservation of the total mass and conservation of the internal energy [5, 15].

$$\frac{dm_{comb}}{dt} = \dot{m}_C + \dot{m}_{fuel} - \dot{m}_T \quad (2.12)$$

$$\frac{dP_3}{dt} = \frac{P_3}{m_{comb}} \cdot (\dot{m}_C + \dot{m}_{fuel} - \dot{m}_T) + \frac{P_3}{T_3 \cdot c_{vmed} \cdot m_{comb}} \cdot [\dot{m}_C \cdot c_{P_{air}} \cdot T_2 - \dot{m}_T \cdot c_{P_{gas}} \cdot T_3 + Q_f \cdot \eta_{comb} \cdot \dot{m}_{fuel} - c_{vmed} \cdot T_3 \cdot (\dot{m}_C + \dot{m}_{fuel} - \dot{m}_T)] \quad (2.13)$$

2.4.1.3 Mathematical Modeling of the Turbine

The performance characteristics of a turbine, such as compressor, were fully described by number of dimensionless parameters given in Table 2-2 [1]. Turbine performance is presented by overall performance characteristics, also known as turbine map is shown in Figure 2-4.

Table 2-2 Dimensionless Turbine Parameters [1]

PARAMETER	MEANINGS
$\left(\frac{\dot{m}_g \cdot \sqrt{C_{Pg} \cdot T_{03}}}{d_2^2 \cdot P_{03}} \right), \left(\frac{\dot{m}_g \cdot \sqrt{\theta}}{\delta} \right)$	Turbine dimensionless mass flow parameters
$\left(\frac{d_2 \cdot N}{\sqrt{C_{Pg} \cdot T_{01}}} \right), \left(\frac{N}{\sqrt{\theta}} \right)$	Turbine dimensionless speed parameters
$\left(\frac{\tau_t}{d_2^2 \cdot P_{03}} \right), \left(\frac{\tau_t}{\delta} \right)$	Turbine dimensionless torque parameters
$\left(\frac{P_{03}}{P_{04}} \right)$	Turbine dimensionless pressure ratio parameters
$\left[\frac{(P_{03}/P_{04})^{(\gamma_a-1)/\gamma_a}}{(T_{03}/T_{04}-1)} \right]$	Isentropic turbine efficiency
$\left(\theta = \frac{T_{03}}{T_{0,ref}} \right)$	Dimensionless temperature parameter
$\left(\delta = \frac{P_{03}}{P_{0,ref}} \right)$	Dimensionless pressure parameter

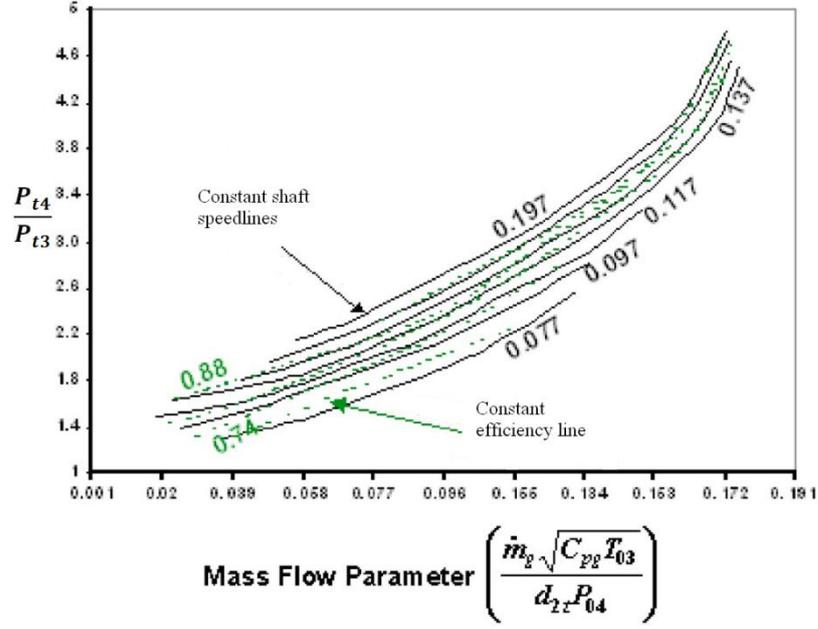


Figure 2-4 Turbine Characteristic Map [1]

In their work, Al-Hamdan and Y. Ebaid did not calculate turbine-cooling effects. Using turbine characteristics, Al-Hamdan and Y. Ebaid developed Eqs. (2.14) and (2.15). They also stated that if any two parameters are known, rest of the parameters can be calculated by these equations [1].

$$\frac{\tau_t}{d_2^2 \cdot P_{03}} = \frac{1}{2\pi} \cdot \eta_t \cdot \left(\frac{d_2 \cdot N}{\sqrt{C_{Pg} \cdot T_{03}}} \right)^{-1} \cdot \frac{\dot{m}_g \sqrt{C_{Pg} \cdot T_{03}}}{d_2^2 \cdot P_{03}} \cdot \left[1 - \left(\frac{P_{04}}{P_{03}} \right)^{(\gamma_g - 1)/\gamma_g} \right] \quad (2.14)$$

$$\frac{\tau_t}{\delta} = f \left(\eta_t, \frac{N}{\sqrt{\theta}}, \frac{\dot{m}_g \sqrt{\theta}}{\delta}, \frac{P_{03}}{P_{04}} \right) \quad (2.15)$$

Eq. (2.14) is completely dimensionless form and Eq. (2.15) in general form. The expansion power and turbine exit temperature calculated by using Eqs. (2.16) and (2.17), respectively [1].

$$W_t = d_2^2 \cdot P_{03} \cdot \sqrt{C_{Pg} \cdot T_{03}} \cdot \frac{\dot{m}_g \sqrt{C_{Pg} \cdot T_{03}}}{d_2^2 \cdot P_{03}} \cdot \eta_t \cdot \left[1 - \left(\frac{P_{04}}{P_{03}} \right)^{(\gamma_g - 1)/\gamma_g} \right] \quad (2.16)$$

$$T_{04} = T_{03} + T_{03} \cdot \eta_t \cdot \left[1 - \left(\frac{P_{04}}{P_{03}} \right)^{(\gamma_g - 1)/\gamma_g} \right] \quad (2.17)$$

Steady state turbine performance was modeled using the Eq. (2.18) which shows the output pressure of the turbine. To calculate pressure value from this equation designer has to read the turbine performance map given in Figure 2-4. In addition, turbine temperature value can be calculated from Eq. (2.8) [8].

$$\left(\frac{T_{T5}}{T_{T4}}\right) = \left(\frac{P_{T5}}{P_{T4}}\right)^{[R \cdot \eta_T / C_p]} \quad (2.18)$$

In their work P. Ailer et al worked on fourth type constitutive equations describes the mass flow rate and the isentropic efficiency of the turbine. They stated that these constitutive equations were needed to solve their nonlinear dynamic equations.

$$\dot{m}_T = \text{const}(2) \cdot q(\lambda_3) \cdot \frac{P_3}{\sqrt{T_3}} \quad (2.19)$$

where $q(\lambda_1)$ was calculated as follow:

$$q(\lambda_3) = f_2 \left(\text{const}(3) \cdot \frac{n}{\sqrt{T_3}}, \frac{P_3}{P_4} \right) \quad (2.20)$$

P. Ailer et al stated that $q(\lambda_3)$ is the function of the corrected revolutions and the turbine pressure ratio. The equations of the isentropic efficiencies of the turbine could be calculated as follow:

$$\eta_T = g_2 \left(\text{const}(3) \cdot \frac{n}{\sqrt{T_3}}, \frac{P_3}{P_4} \right) \quad (2.21)$$

P. Ailer et al stated that the parameters and the constants used in these functions could be determined with the help of the results of the experiments. To find $q(\lambda_3)$ there are two approximations. First approximation $q(\lambda_3)$ is the linear function of the corrected number of revolution and linear function of the dimensionless mass flow rate of the turbine, second approximation is the dimensionless mass flow rate of the turbine is the linear function of the corrected number of revolutions and parabolic function of the compressor pressure ratio. Detailed solutions of these functions are given in [5, 15].

2.4.1.4 Gas Turbine Computer Simulation

Experimental methods to understand the engine characteristic were expensive and time consuming. Therefore, modeling and simulation of the engines by using simulation programs was crucial for determining its performance. Matlab/Simulink has become the most important software for modeling and simulating the dynamic system. In their work, Y. Yu et al discussed digital control model adopted for the simulation of three-shaft gas turbine. Different types of simulation programs might be as follows:

1. Program might simulate the engine performance at design stage where the real engine cannot perform it.
2. Simulation at the application stage where engine is already constructed.
3. Simulation at the application stage with the auxiliary equipments such as oil pumps, valves, hydraulics, etc.
4. Simulation for the performance estimation of the gas turbine to meet higher output requirements.

According to [1], transient operation of the gas turbines is more complicated than the steady state operation. Transient operation of gas turbines was not investigated in their study. The flowchart for gas turbine engine control program logic is shown in Figure 2-5. This flowchart shows general procedure for the gas turbine engine control. When engine program starts, it takes ambient conditions and fuel properties from the sensors. It calculates compressor input and output conditions from the beta line method discussed in [1]. From the compressor output conditions, simulation code calculates the combustion chamber output conditions which are also input conditions of the turbine. At the end, code calculates the output conditions of the turbine and it stops the loop [1, 2, 4, 6].

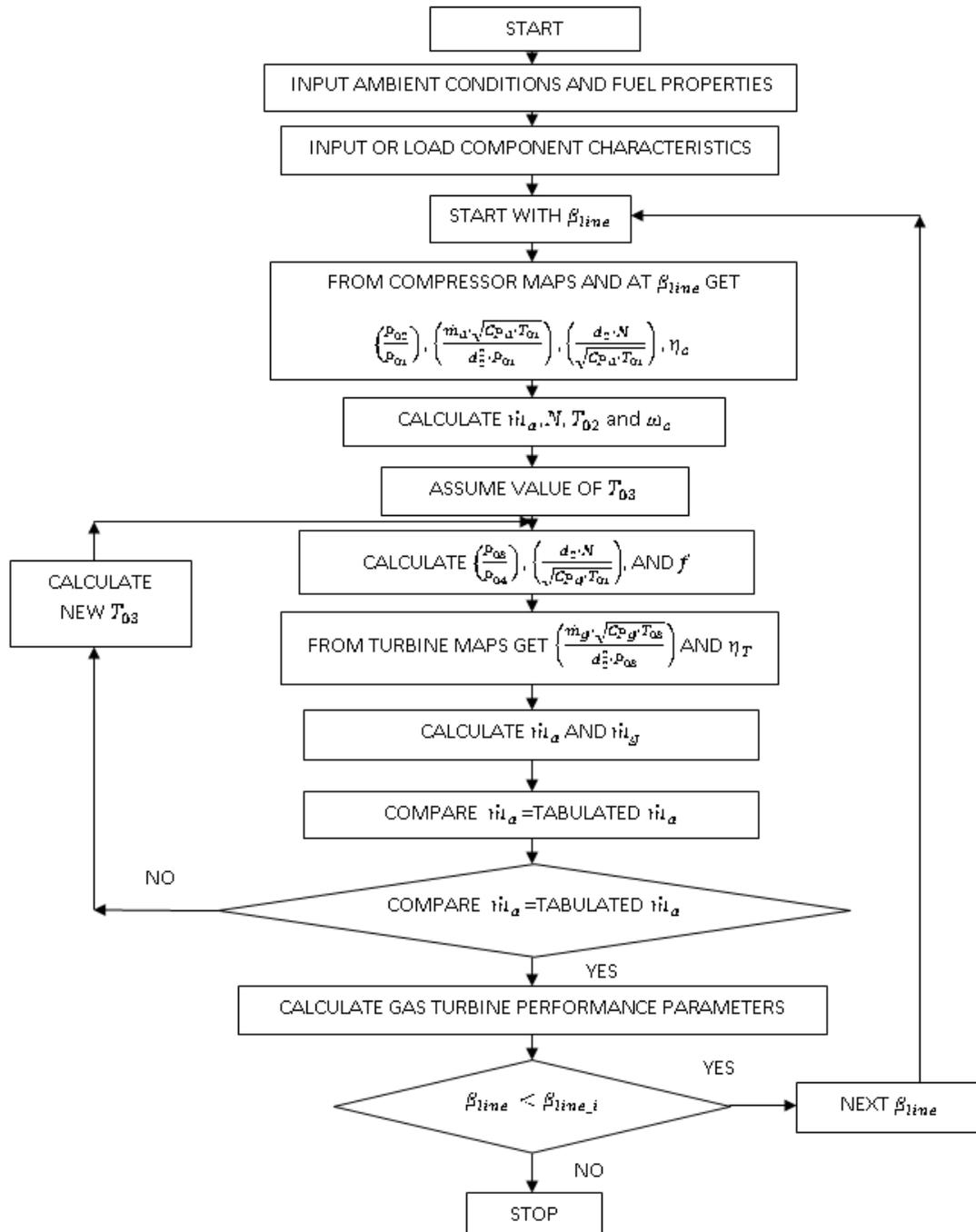


Figure 2-5 Computer Simulation Flowchart [1]

2.4.1.5 Simulation Results

Simulation program gave the following result, which was produced by using the dimensionless parameters [1]. As can be seen from Figure 2-6, the specific fuel consumption decreases while thermal efficiency increases. This means that after 140 kW power output, test engine becomes more efficient than lower power output. Therefore, designer could choose their operating conditions close to these points.

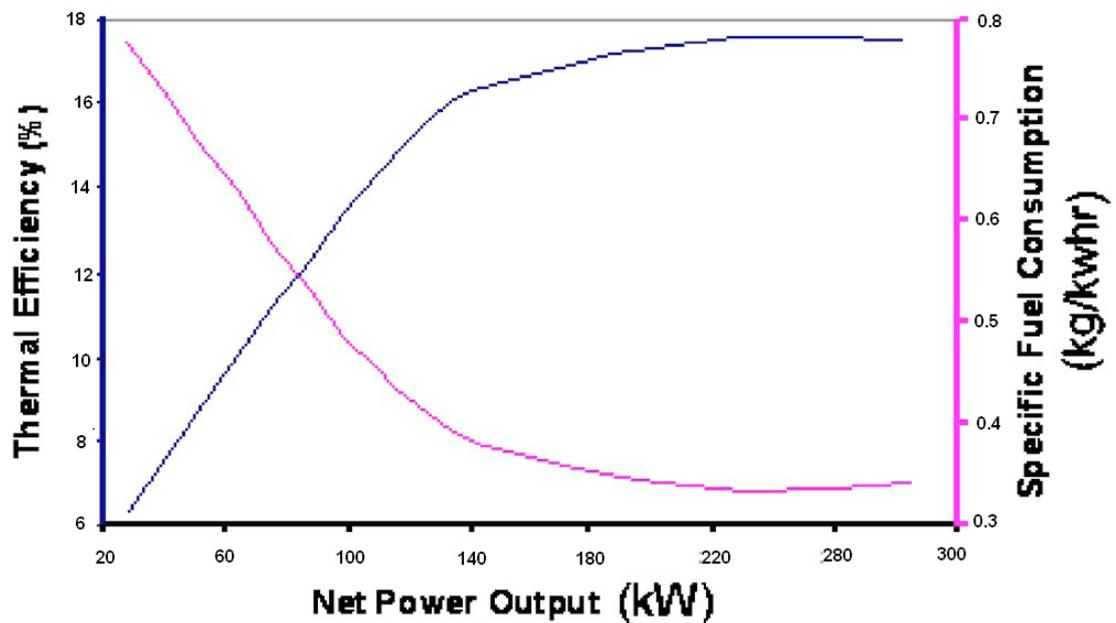


Figure 2-6 Variation of thermal efficiency and specific fuel consumption at 42 [krpm] [1]

Figure 2-7 and Figure 2-8 shows the dynamic response of the shaft speed and total temperatures. From the Figure 2-7 it is obvious that shaft speed is increasing with respect to the time. On the other hand, according to the results given in Figure 2-8, total temperatures before and after turbine is decreasing after a small peak while total temperature after compressor remains almost constant [5, 15].

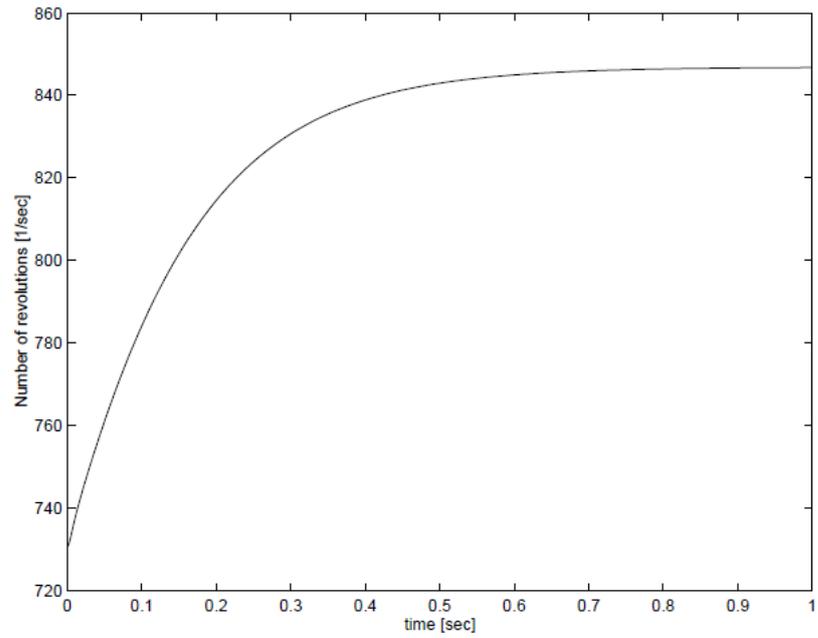


Figure 2-7 Dynamic Response of the Number of Revolution [5, 15]

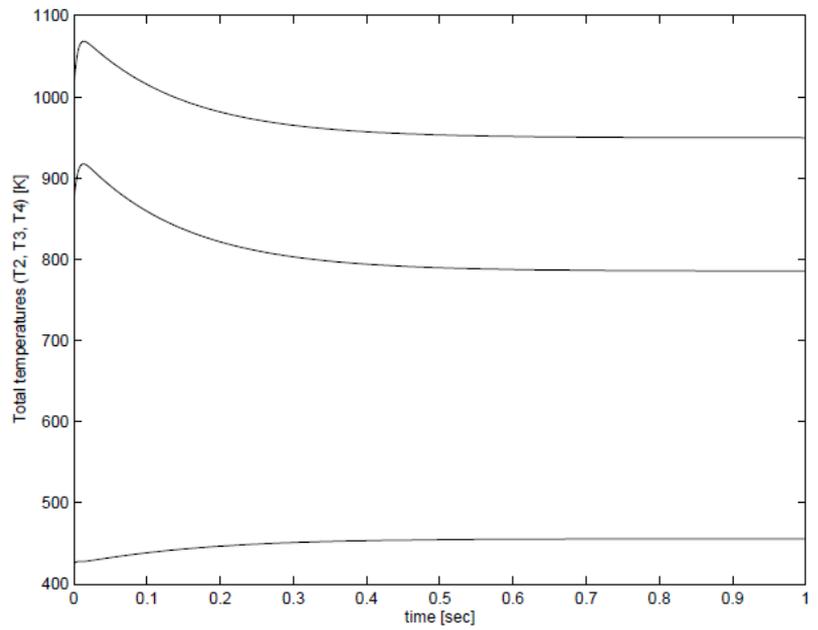


Figure 2-8 Dynamic Responses of the Total Temperatures [5, 15]

2.4.2 Some Important Studies about Modeling, Simulation and Control of Gas Turbine Engines

Lichtsinder et al [2] discussed control and real-time simulation model for their jet engine. Main objective of this paper is to investigate simple real time transient model for jet engines. They stated that linear and piecewise linear techniques were not always suitable for turbine engine controller design. According to them, linear engine model, describing dynamic processes around an operating point does not provide adequate engine behavior to design a controller for the course of entire flight. On the other hand piecewise linear models provide greater accuracy for fast simulations but they are not convenient for controller design also. Piecewise linear model coefficients depend on input/output variables, so it is nonlinear model because of the nonlinear input/output coefficients. Application of linear model theory may be unsuccessful and stability examination of a piecewise linear control system utilizing linear theory may give incorrect solutions. Therefore, Lichtsinder et al in [2] use fast engine model for nonlinear system solution. They employed AMT Olympus engine, which was used in that study.

M. Nagpal et al [7] worked on testing and modeling of gas turbines. In 1996, the North American western power grid experienced two important power disturbances. Western System Coordinating Council (WSCC) stated that all systems in western grid greater than 10 MVA were tested to verify the generator reactive power limits and the dynamic model data being utilized for system studies. This paper presents field experiences of the authors in testing and modeling of gas turbines of this power grid. The analysis report of two disturbances stated that some of the generators did not response dynamically or steady state as predicted by studies. M. Nagpal et al investigated these gas turbines' behavior in order to find the solution for transient response problem.

M. Camporeale et al [4] made a review for the real time dynamic simulation of gas turbines using SIMULINK in [4]. They developed a mathematical model and numerical solution technique to solve the set of algebraic and ordinary differential equations that describe the behavior of their gas turbine. To reduce the computational time, M. Camporeale et al [4] discussed linear models for real time simulation. However, the simulation programs based on linearized models should be used only neighborhood of the working point which the linearization fulfilled.

W.W. Hung discussed dynamic simulation of gas turbine generating unit in his study [3]. He stated that there is an increasing demand for accurate engine model in order to develop more efficient gas turbine controller design. In his work, W.W. Hung utilized linear modeling technique and he said that this approach had been extensively utilized in gas turbine control studies. Linearized equations giving dynamic behavior of a gas turbine was based on quasi-static transient thermodynamic and flow processes [3]. He stated that these processes continuously progress from one equilibrium state to another along an equilibrium curve. This assumption gave a functional relation between inputs and outputs. In his study, W.W. Hung gave an acceleration and deceleration fuel schedule given in Figure 2-1 for a gas turbine engine. According to W.W. Hung, an ideal acceleration form allows the engine to accelerate at a reasonably fast rate without the engine driven into the overheating its components. On the other hand, there was a minimum level to which the fuel flow can be reduced without causing flameout problem [3].

G. Kocer et al [6] investigated real time simulation code based on aero-thermal model of a turbojet engine. In their study, they used a small turbojet engine with no bleed and no turbine cooling. The algorithm they discussed was composed of a set of differential equations and a set of nonlinear algebraic equations. G. Kocer et al utilized Beta Line Method in order to digitize the compressor characteristics. They found different results for two different input scenarios.

T. Korakianitis et al [8] presented instantaneous response and transient-flow component models for the prediction of the transient response of gas turbines. They utilized the principles of conservation of mass, energy, and momentum.

J.H. Kim et al worked on dynamic simulation of a heavy-duty gas turbine in [9] and [11]. They developed a simulation program that was for transient analysis of startup process of a heavy duty gas turbine used for power generation [9]. J.H. Kim et al discussed unsteady one dimensional conservation equations and those equation sets are solved numerically. They stated that gas turbine might suffer transient operation because of startup, load change, shutdown and other environmental disturbances [9, 11].

In their study, S.M. Camporeale et al [10] investigated nonlinear mathematical model to simulate the dynamic behavior of a regenerative single shaft power plant. Their main purpose was to provide a fast and reliable engine model to understand the power plant and to develop an effective controller. Their model was also convenient for other gas turbine power

plants for instance a gas steam combined cycle power plant, steam injected gas turbines, etc [10].

A. Watanabe et al [12] worked on PID and fuzzy logic algorithm in order to control SR-30 turbojet engine. They obtained transfer function of the SR-30 by using frequency response method. They tested and simulated both closed loop controller PID and fuzzy logic controller. They developed their model with MATLAB environment and tested it by NI LabVIEW.

Y. Yu et al [13] performed MATLAB/SIMULINK based simulation for control system of marine three-shaft gas turbine. They stated that for the control studies a detailed dynamic model was necessary. In their study, nonlinear gas turbine model was adopted for calculations. They connected the gas turbine components appropriately by means of thermodynamic and mechanical links.

J. Mu et al [14] gave neural network representation suitable for modeling a gas turbine engine. This model then utilized to provide model based control strategies. They stated that due to the nonlinearities of the engine, PID controller could not cope with its whole operating range. Hence, a gain scheduling PID controller was required.

In their study, R. Andoga et al [16] discussed digital electronic control of a small turbojet engine. They stated that main purpose of control of gas turbine was increasing its safety and efficiency. Their engine was controlled by PIC 16F84A microcontroller, which was manipulating the fuel flow valve.

2.4.3 Outline of the Current Work

In this thesis work combination of different gas turbine modeling and control methods are investigated. Linear, nonlinear and piecewise linear techniques are discussed. After developing the mathematical model of the gas turbine engine, some control algorithms are studied such as P, PID or fuzzy logic controller.

It can be concluded that there are some advantages and disadvantages of linear, nonlinear and piecewise linear methods. Because of simulation restrictions of linear and piecewise linear methods, nonlinear modeling technique was preferred in the current study.

CHAPTER 3

MATHEMATICAL MODELING OF A SMALL TURBOJET ENGINE

3.1 Introduction

Contemporary gas turbine engines are produced in various configurations: single, twin and triple spool engines are examples of these. Apart from these configurations, engine dimensions vary depending on application areas. These are the factors for classification of gas turbines. In this study, AMT Olympus HP turbojet engine with 250 N class has been used for controller design as illustrated in Figure 3-1.

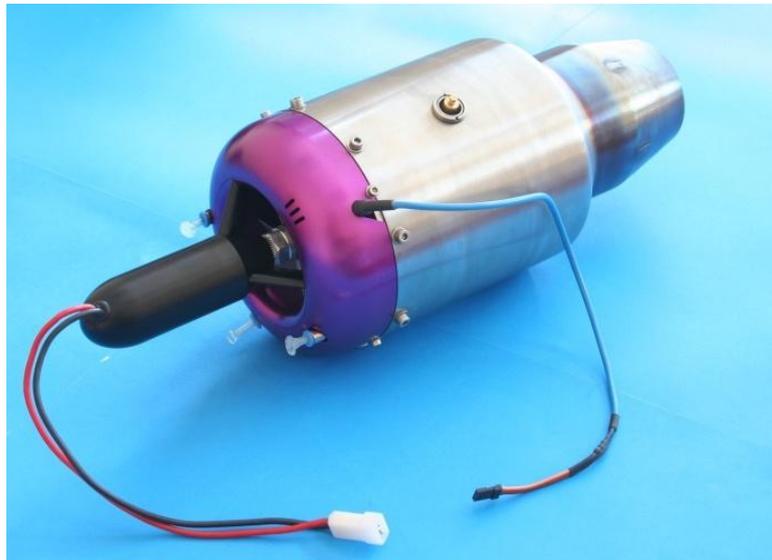


Figure 3-1 AMT Olympus HP Turbojet Engine

Small gas turbine engines are utilized in many fields such as unmanned air vehicles, cruise missiles, drones, tactical UAV's, marines and model aircrafts (See Figure 3-2).



Figure 3-2 Gas Turbine Application Areas [24, 25]

In this part of the thesis, nonlinear mathematical modeling of AMT Olympus turbojet engine is given. All components' mathematical representations are investigated. Compressor characteristics are digitized by using Beta line method. At last, modeling results are discussed and suitable controller is proposed.

3.2 Mathematical Modeling

A small gas turbine engine is composed of the parts given in Figure 3-3. In this figure, 1, 2, 3, 4, 5 and 9 refer to inlet, compressor input, diffuser output, combustion chamber output, turbine output and nozzle output points respectively.

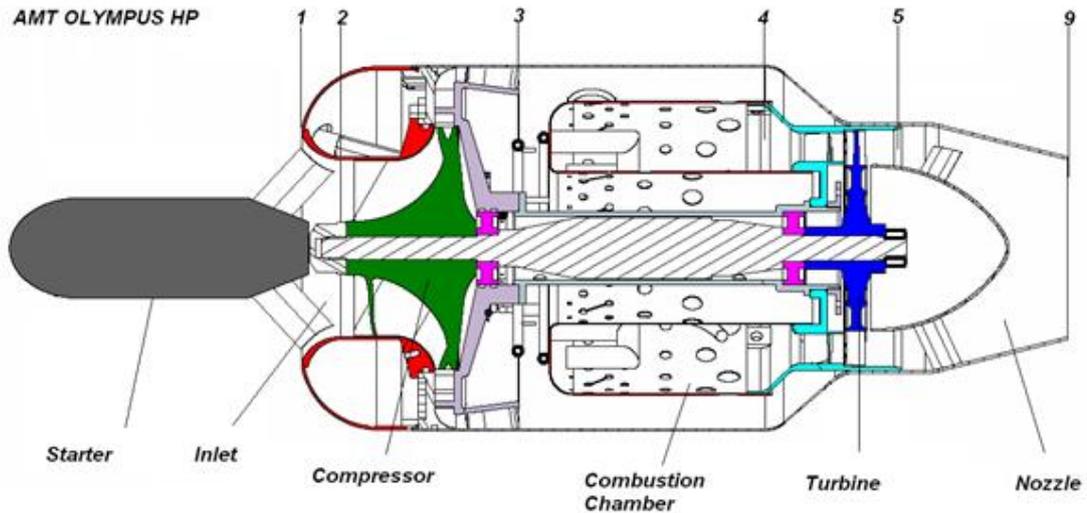


Figure 3-3 AMT Olympus HP Components

Gas turbine engines generally run in following sequence: First, the air is sucked and compressed by the compressor to increase its total pressure and temperature. High-pressure air is axially directed towards combustion chamber and mixed up with fuel and is ignited with a glow plug. The mixture is combusted at constant pressure (in non-ideal case, there is a small pressure loss). Energy and momentum of air and fuel mixture is increased. Thus, chemical energy is transformed into kinetic one. This high temperature and pressure mixture, produced by combustion, transfers its kinetic energy to turbine while colliding on to its blades. At this state, the kinetic energy, which is transferred to turbine, is transmitted to compressor by a shaft. The mixture passing from turbine losses its temperature and pressure then it exits from nozzle at high velocity. The momentum difference of the air entering and exiting the engine produces engine propulsion.

Specification of AMT Olympus HP turbojet engine is as follows:

- Maximum thrust: 250 N
- Mass: 2900 g.
- Diameter: 140 mm
- Length: 280 mm (excluding the electric motor starter)
- Fuel Consumption: 37.63 kg/h at max. thrust.
- Maximum Shaft Speed: 108 [krpm]

In this section, mathematical models of the individual components will be presented. After giving the mathematical model of components, simulation code will be discussed. At last, simulation results will be interpreted.

3.2.1 Component Models

As mentioned in the previous sections, it is crucial to find mathematical modeling of gas turbines in order to understand and investigate their transient and steady state behaviors under different working conditions. Designers have to find the mathematical model of all individual components to study entire engine. In this part of the thesis, first of all modeling assumptions will be given. Afterwards, compressor modeling, beta line method, combustion chamber modeling and turbine modeling will be discussed respectively. Finally, work balance, thrust, and simulation code results will be taken into consideration.

3.2.1.1 Assumptions

Conservation laws are used for obtaining dynamic equations of gas turbines [5]. Non-linear differential equations are obtained by using internal energy and conservation of mass principles. The assumptions made can be listed as follows:

- a) Heat loss is neglected.
- b) For the inlet; constant pressure loss is assumed (σ_1).
- c) For the compressor;
 - Constant mass flow rate; $\dot{m}_{Cin} = \dot{m}_{Cout} = \dot{m}_C$
 - No energy storage effect; $U_2 = \text{constant}$
- d) For the combustion chamber;
 - Constant pressure loss coefficient σ_{comb}
 - Constant efficiency η_{comb}
 - Enthalpy of fuel is neglected.
- e) For the turbine;
 - Constant mass flow rate ; $\dot{m}_{Tin} = \dot{m}_{Tout} = \dot{m}_T$
 - No energy storage effect; $U_4 = \text{constant}$
- f) For the nozzle; constant pressure loss coefficient is assumed σ_N

- g) Specific heat coefficients are not constant. There are some look-tables for these constants. In this study, polynomial forms of these tables are used for specific heat at constant pressure and specific heat at constant temperature [1]. These polynomials are given in below:

for air at the low temperature range of 200-800 K:

$$C_{Pa} = 1.0189 \cdot 10^3 - 0.13784 \cdot T_a + 1.9843 \cdot 10^{-4} \cdot T_a^2 + 4.2399 \cdot 10^{-7} \cdot T_a^3 - 3.7632 \cdot 10^{-10} \cdot T_a^4 \quad (3.1)$$

for the air at the high temperature range of 800-2200 K:

$$C_{Pa} = 7.9865 \cdot 10^2 + 0.5339 \cdot T_a - 2.2882 \cdot 10^{-4} \cdot T_a^2 + 3.7421 \cdot 10^{-8} \cdot T_a^3 \quad (3.2)$$

for specific heats of products of combustion:

$$C_{Pg} = C_{Pa} + [f/(1 + f)] \cdot B_T \quad (3.3)$$

where B_T at the low-temperature range of 200-800 K:

$$B_T = -3.59494 \cdot 10^2 + 4.5164 \cdot T_g + 2.8116 \cdot 10^{-3} \cdot T_g^2 - 2.1709 \cdot 10^{-5} \cdot T_g^3 + 2.8689 \cdot 10^{-8} \cdot T_g^4 - 1.2263 \cdot 10^{-11} \cdot T_g^5 \quad (3.4)$$

and B_T at high temperature range of 800-2200 K:

$$B_T = 1.0888 \cdot 10^3 - 0.1416 \cdot T_g + 1.916 \cdot 10^{-3} \cdot T_g^2 - 1.2401 \cdot 10^{-6} \cdot T_g^3 + 3.0669 \cdot 10^{-10} \cdot T_g^4 - 2.6117 \cdot 10^{-14} \cdot T_g^5 \quad (3.5)$$

where T_a and T_g are mean temperature in compressor and mean temperature in turbine respectively.

3.2.1.2 Compressor

AMT Olympus HP turbojet engine has a single-stage radial compressor as shown in Figure 3-4. Station numbers are given in Figure 3-3 for the compressor. Number “2” refers to the inlet of the compressor, number “3” refers to the output point of the compressor. Total

temperature, mass flow rate and efficiency are found by using Eqs. (3.6) - (3.8). Eqs. (3.7) and (3.8) show that the mass flow rate and efficiency of the compressor could be calculated via Beta Line Method which is a kind of digitizing method [1]. Different digitizing processes for map reading can be given as follows:

1. Deriving an equation to determine the performance map of the components.
2. Storing the components' performance characteristics in look-up tables then using an interpolation or extrapolation techniques to determine all points.
3. Using neural network technique to represent the characteristic of compressor or turbine.

$$T_{t3} = T_{t2} + \frac{T_{t2}}{\eta_c} \cdot \left[\left(\frac{P_{t3}}{P_{t2}} \right)^{(\gamma_a - 1)/\gamma_a} - 1 \right] \quad (3.6)$$

$$\dot{m}_c = f(N, \pi_c) \quad (3.7)$$

$$\eta_c = g(N, \pi_c) \quad (3.8)$$

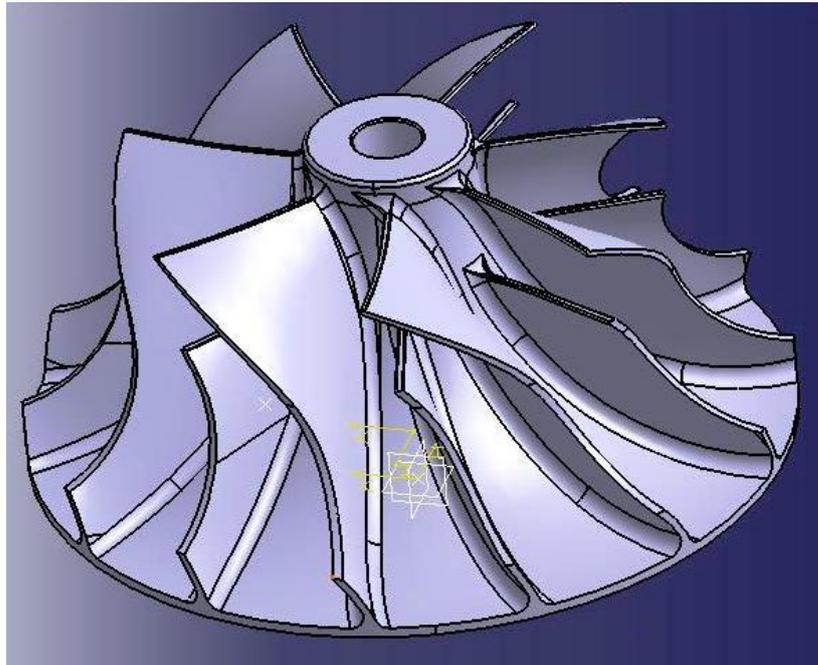


Figure 3-4 Radial Compressor

Al-Hamdan et al [1], S.M. Camporeale et al [4] and G.Koçer et al [6] used the second method to digitize their compressor maps. This method is called “Beta Line Method”. The compressor or turbine maps in the standard format given in Figure 2-3 and Figure 2-4 respectively could not be directly utilized in a computer program therefore they must be converted into a good numerical representation. There are some difficulties associated with the numerical read-outs. For instance, it is not possible to read compressor map parameters with given N_{Dim} and pressure ratio P_{02}/P_{01} . Because there are two values for the mass flow rate parameters \dot{m}_{Dim} at given pressure ratio (See point A in Figure 3-5). It is also not possible to read the efficiency η_c , from the compressor map with N_{Dim} and mass flow rate parameter \dot{m}_{Dim} , because speed lines can be vertical in some parts of the map, so there might be two values for the pressure ratio P_{02}/P_{01} at a single value of mass flow parameter \dot{m}_{Dim} (See point B in Figure 3-5). Beta Line Method allowed extensive map reading capabilities using the shape of the parameter lines with the β lines and speed parameter N_{Dim} [1]. Auxiliary coordinates called β lines could be selected arbitrarily with only to restrictions,

1. There is no intersection between the β lines,
2. β lines are equally spaced.

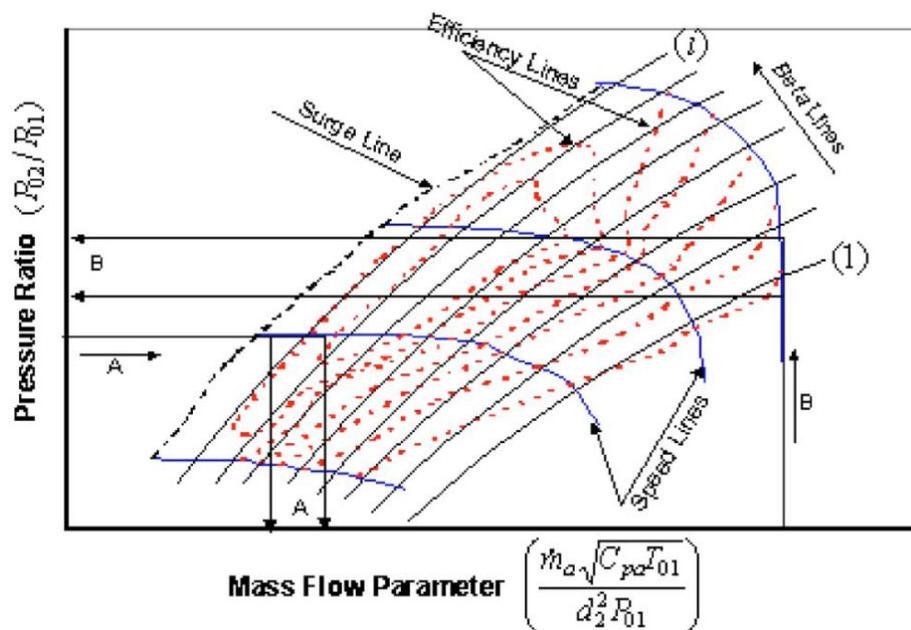


Figure 3-5 Problems with Reading Compressor Maps [1]

According to the beta line method, these lines have numbers starting from one. They could be parabolic or straight lines. If one of the parameters and rotational speed parameter N_{Dim}

are known, the other two parameters may be calculated from these look-up tables by using Beta line method. Compressor map used for the beta-line calculations is shown in the Figure 3-6. This compressor map was obtained from the Advanced Micro Turbines Netherlands, which is the producer of the engine utilized in this study. As can be seen from Figure 3-6, Beta Line points are determined from the intersections between β_1 - β_{12} lines and shaft speed N. Pressure ratio, mass flow rates and efficiency values picked from these intersections are given in Table 3-1, Table 3-2 and Table 3-3 respectively.

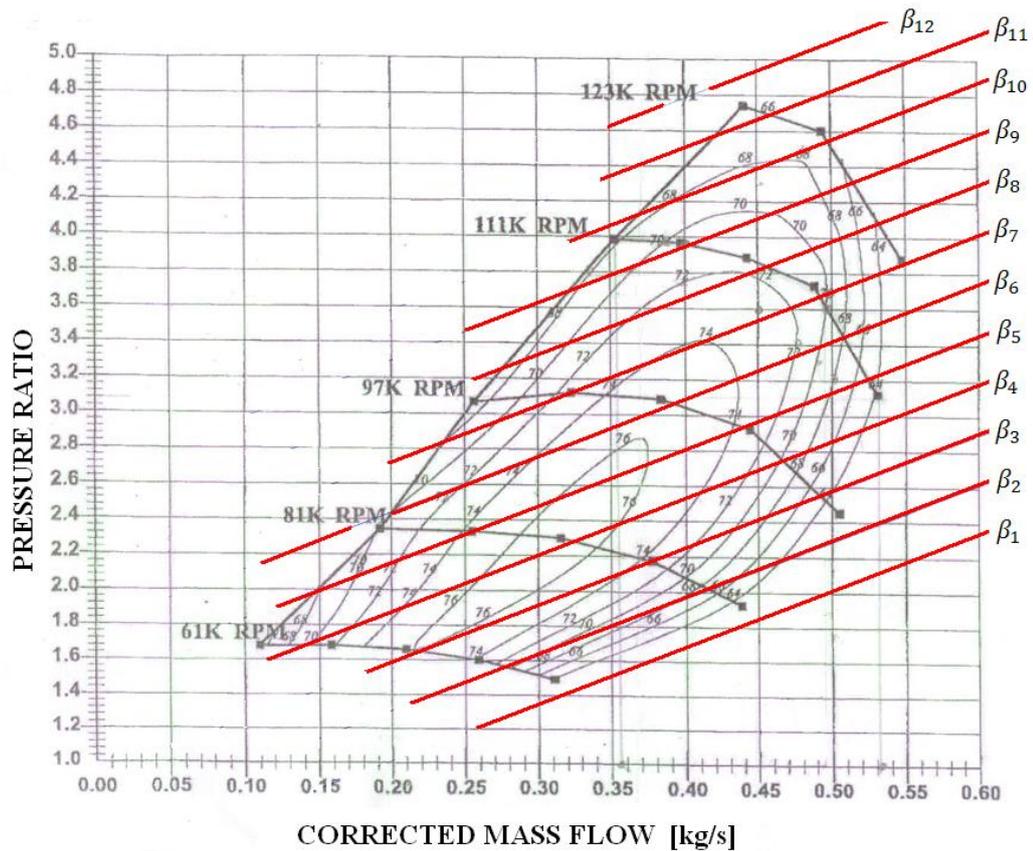


Figure 3-6 Compressor Map [32]

Table 3-1 Compressor Pressure Ratio Values from Beta Lines

N [krpm]	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}
61	0	1.56	1.64	1.67	0	0	0	0	0	0	0	0
81	0	2.01	2.17	2.28	2.33	0	0	0	0	0	0	0
97	0	0	2.54	2.75	2.93	3.06	3.22	0	0	0	0	0
111	0	0	0	0	3.21	3.43	3.67	3.84	3.95	0	0	0
123	0	0	0	0	0	0	3.86	4.09	4.31	4.54	4.69	0

Table 3-2 Compressor Mass Flow Rate Values from Beta Lines

N [krpm]	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}
61	0	0.27	0.21	0.13	0	0	0	0	0	0	0	0
81	0	0.41	0.37	0.32	0.25	0	0	0	0	0	0	0
97	0	0	0.49	0.46	0.43	0.38	0.31	0	0	0	0	0
111	0	0	0	0	0.52	0.50	0.48	0.45	0.39	0	0	0
123	0	0	0	0	0	0	0.54	0.52	0.50	0.49	0.45	0

Table 3-3 Compressor Efficiency Values from Beta Lines

N [krpm]	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}	β_{11}	β_{12}
61	0	0.71	0.75	0.69	0	0	0	0	0	0	0	0
81	0	0.66	0.73	0.76	0.74	0	0	0	0	0	0	0
97	0	0	0.64	0.69	0.73	0.75	0.72	0	0	0	0	0
111	0	0	0	0	0.65	0.68	0.70	0.71	0.70	0	0	0
123	0	0	0	0	0	0	0.62	0.64	0.65	0.66	0.66	0

For example, compressor efficiency and mass flow rate can be calculated as follows:

- A pressure ratio and shaft speed inputs can be taken from the experiments,
- Beta values for this pressure ratio and N are calculated by iteration,
- A Beta constant is calculated from the iteration,
- Using Beta constant and shaft speed N, compressor efficiency and mass flow rate may be calculated.

As a result, compressor exit conditions T_{i3} , η_C , \dot{m}_C and P_{i3}/P_{i2} may be calculated from the equations and tables given above. Compressor exit conditions are used in combustion

chamber performance equations. In other words, compressor exit conditions are the input conditions of combustion chamber.

3.2.1.3 Combustion Chamber

AMT Olympus HP turbojet engine has an annular type combustion chamber given in Figure 3-7.



Figure 3-7 Combustion Chamber of AMT Olympus HP

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 . Derivation of the model equations is given in the following steps;

1. Conservation balance of the total mass:

$$\frac{dm}{dt} = \dot{m}_{in} - \dot{m}_{out} \quad (3.9)$$

2. Conservation balance of the internal energy, where the heat energy flows and the work terms also taken into account:

$$\frac{dU}{dt} = \dot{m}_{in} \cdot h_{in} - \dot{m}_{out} \cdot h_{out} + Q + W \quad (3.10)$$

In addition, internal energy may be written as:

$$U = c_v \cdot m \cdot T \quad (3.11)$$

Derivation of Eq. (3.11) is,

$$\frac{dU}{dt} = c_v \cdot \frac{d}{dt} (m \cdot T) \quad (3.12)$$

$$\frac{dU}{dt} = c_v \cdot T \cdot \frac{dm}{dt} + c_v \cdot m \cdot \frac{dT}{dt} \quad (3.13)$$

Eq. (3.10) and (3.13) are equal thus,

$$\frac{dU}{dt} = c_v \cdot T \cdot \frac{dm}{dt} + c_v \cdot m \cdot \frac{dT}{dt} = \dot{m}_{in} \cdot h_{in} - \dot{m}_{out} \cdot h_{out} + Q + W \quad (3.14)$$

From the Eqs. (3.9) and (3.14) state equation for the temperature can be written as,

$$\frac{dT}{dt} = \frac{\dot{m}_{in} \cdot h_{in} - \dot{m}_{out} \cdot h_{out} + Q + W - c_v \cdot T \cdot (\dot{m}_{in} - \dot{m}_{out})}{c_v \cdot m} \quad (3.15)$$

The ideal gas equation ($P \cdot V = m \cdot R \cdot T$) is used together with Eq. (3.15) and the following state equation is derived:

$$\frac{dP}{dt} = \frac{P}{m} \cdot (\dot{m}_{in} - \dot{m}_{out}) + \frac{P}{T} \cdot \left(\frac{\dot{m}_{in} \cdot h_{in} - \dot{m}_{out} \cdot h_{out} + Q + W - c_v \cdot T \cdot (\dot{m}_{in} - \dot{m}_{out})}{c_v \cdot m} \right) \quad (3.16)$$

Using Eqs. (3.15) and (3.16) following dynamic equations were derived;

$$\frac{dP_{t4}}{dt} = \frac{P_{t4}}{m_{Comb}} \cdot (\dot{m}_C + \dot{m}_{fuel} - \dot{m}_T) + \frac{P_{t4}}{T_{t4} \cdot c_{vmed} \cdot m_{Comb}} \cdot \left[\dot{m}_C \cdot c_{P_{air}} \cdot T_{t3} - \dot{m}_T \cdot c_{P_{gas}} \cdot T_{t4} + Q_f \cdot \eta_{comb} \cdot \dot{m}_{fuel} - c_{vmed} \cdot T_{t4} \cdot (\dot{m}_C + \dot{m}_{fuel} - \dot{m}_T) \right] \quad (3.17)$$

$$\frac{dT_{t4}}{dt} = \frac{\dot{m}_C \cdot c_{P_{air}} \cdot T_{t3} - (\dot{m}_C + \dot{m}_{fuel}) \cdot c_{P_{gas}} \cdot T_{t4} + Q_f \cdot T_{t4} \cdot \dot{m}_{fuel}}{c_{vmed} \cdot m_{Comb}} \quad (3.18)$$

Pressure drop and efficiency in the combustion chamber are evaluated from the experiments and given as $\sigma_{\text{comb}} = 0.95$ and $\eta_{\text{comb}} = 0.65$.

ρ_{comb} can be evaluated from the ideal gas equation.

$$\rho_{\text{Comb}} = \frac{P_m}{R \cdot T_m} \quad (3.19)$$

where P_m and T_m are the mean pressure and mean temperature in the combustion chamber. To find mass of the gas in the combustion chamber, volume of the combustion chamber must be computed. Form of the combustion chamber used in AMT Olympus HP turbojet engine is given in Figure 3-8. It is assume that combustion chamber has a simple cylindrical form. Thus, angular parts of the combustion chamber were neglected.

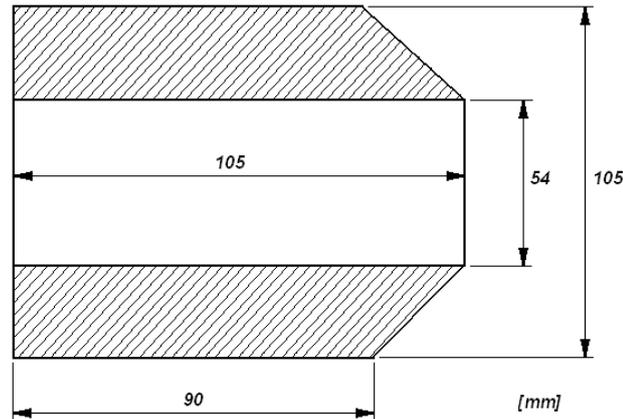


Figure 3-8 Combustion Chamber Volume

Volume of the combustion chamber is calculated as follow:

$$V_{\text{Comb}} = \pi \cdot \frac{(105 \cdot 10^{-3})^2}{4} \cdot (105 \cdot 10^{-3}) - \pi \cdot \frac{(54 \cdot 10^{-3})^2}{4} \cdot (54 \cdot 10^{-3}) = 6.6872 \cdot 10^{-4} \text{ m}^3 \quad (3.20)$$

therefore, mass of the gas in the combustion chamber is evaluated from the equation given below.

$$m_{Comb} = \rho_{Comb} \cdot V_{comb} \quad (3.21)$$

Q_f value given in Eq. (3.17) is known as lower calorific value of the fuel and sometimes it is called as LHV (lower heating value). This value which is for kerosene fuel is given below.

$$Q_f = 43250 \cdot 10^3 \text{ [Joule/kg]} \quad (3.22)$$

Specific heat constant at constant volume can be determined from a thermodynamic rule. It is known that enthalpy and internal energy may be written as follows:

$$dh = C_p \cdot dT \quad (3.23)$$

$$du = C_v \cdot dT \quad (3.24)$$

Relation between dh and du is given below.

$$dh = du + R \cdot dT \quad (3.25)$$

Substituting Eqs. (3.23) and (3.24) into (3.25) gives the following relation;

$$C_p \cdot dT = C_v \cdot dT + R \cdot dT \quad (3.26)$$

Therefore

$$C_{vgas} = C_{pgas} - R \quad (3.27)$$

where $R = 286.9$.

As a result, combustion chamber exit conditions can be calculated from the equations and constants given above. As it can be seen from Eqs. (3.17) and (3.18), nonlinear differential equations are determined in order to model combustion chamber. These nonlinear equations are solved numerically. Runge-Kutta fourth order rule is selected for numerical solution. Combustion chamber exit conditions were used as input conditions for the turbine in following part.

3.2.1.4 Turbine

AMT Olympus HP turbojet engine has a single stage axial turbine given in Figure 3-9. Station numbers are given in Figure 3-3 for the turbine. Number “4” refers to inlet of the

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

$$T_{t5} = T_{t4} + T_{t4} \cdot \eta_t \cdot \left[1 - \left(\frac{P_{t5}}{P_{t4}} \right)^{(\gamma_g - 1) / \gamma_g} \right] \quad (3.28)$$

$$\dot{m}_T = f(N, \pi_T) \quad (3.29)$$

$$\eta_T = g(N, \pi_T) \quad (3.30)$$



Figure 3-9 Turbine of AMT Olympus HP

Turbine map is demonstrated in Figure 3-10. 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.

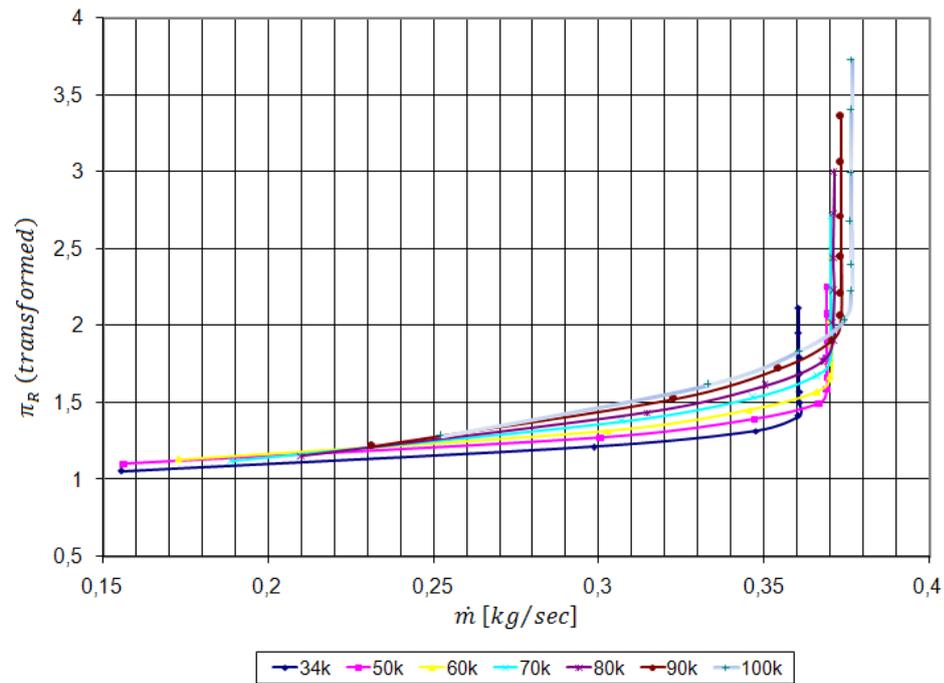


Figure 3-10 AMT Olympus HP Turbine Map

Turbine map was obtained by the theoretical calculations and experimental data. These investigations were made in Middle East Technical University Aerospace Engineering Department. These measurements were used in order to find turbine working conditions. On the other hand efficiency values for turbine were taken from these experiments and theoretical calculations. Because it was impossible to take efficiency directly from the experiments, theoretical calculations were utilized. Turbine exit conditions calculated above may be 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. Because of this reason, these kinds of engines are called as self-sustaining engines. Work balance of the small gas turbine engine is given in the following section.

3.2.1.5 Work Balance

Work balance of a gas turbine engine can be determined as equality between compressor and turbine work. In a small gas turbine engine with a single-spool, compressor is connected to the turbine with a shaft as shown Figure 3-11.



Figure 3-11 AMT OLYMPUS HP Engine's Shaft

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 is cut off. Shaft dynamic can be derived from the power, torque and inertia relations given in the following.

$$Power = Torque \cdot Angular Velocity = T \cdot \omega \quad (3.31)$$

Also it's known as,

$$T = I \cdot \alpha \quad (3.32)$$

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

$$W = I \cdot \alpha \cdot \omega = I \cdot \frac{d\omega}{dt} \cdot \omega \quad (3.33)$$

Angular velocity of the shaft may be calculated from Eq. 3.33.

$$\frac{d\omega}{dt} = \frac{1}{I \cdot \omega} \cdot (W_t - W_c - W_f) \quad (3.34)$$

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

$$W_c = \dot{m}_c \cdot (h_3 - h_2) = \dot{m}_c \cdot c_{pair} \cdot (T_{t3} - T_{t2}) \quad (3.35)$$

$$W_t = \dot{m}_t \cdot (h_4 - h_5) = \dot{m}_t \cdot c_{Pgas} \cdot (T_{t4} - T_{t5}) \quad (3.36)$$

W_f is taken from experiments.

Substituting Eqs. (3.36) and (3.35) into Eq. (3.34) gives the following equation;

$$\frac{d\omega}{dt} = \frac{1}{I \cdot \omega} \cdot \{ [\dot{m}_t \cdot c_{p_{gas}} \cdot (T_{t4} - T_{t5})] - [\dot{m}_c \cdot c_{p_{air}} \cdot (T_{t3} - T_{t2})] - W_f \} \quad (3.37)$$

3.2.1.6 Thrust

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.

$$F = \dot{m} \cdot (U_e - U_0) \quad (3.38)$$

(3.40) may be written as follows:

$$F = (\dot{m}_f + \dot{m}_c) \cdot U_e - \dot{m}_c \cdot U_0 \quad (3.39)$$

Thrust can be calculated from Eq. (3.39). 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.

3.2.2 Simulation Code

Simulation code is written in MATLAB environment and its results are verified by the test data. This code is valid for single-spool small turbojet engines which were tested on a test stand shown in Figure 3-12.

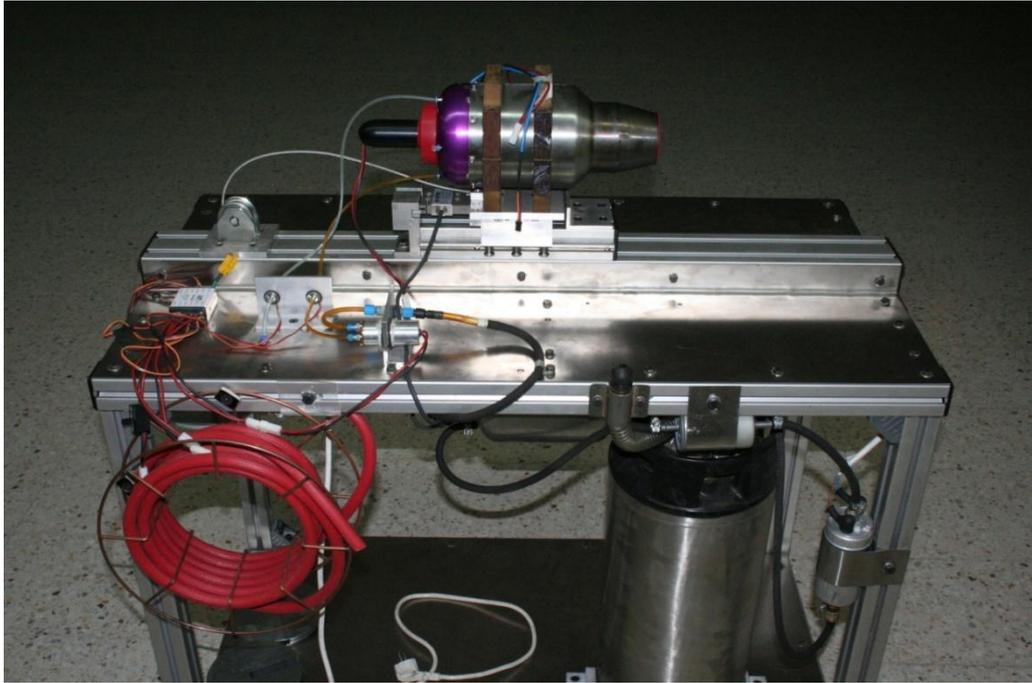


Figure 3-12 AMT Olympus HP Test Stand

This test equipment is used to determine the engine characteristics at steady-state conditions. On the other hand, it is possible to read thrust, nozzle temperature and shaft speed from PC or LCD display. There are three points for measurement equipments that are given in Figure 3-13. As can be seen from the Figure, the measurement points are selected after the compressor, combustion chamber and turbine respectively. Temperature data are taken from one side of the engine while the pressure data are collected from the opposite.

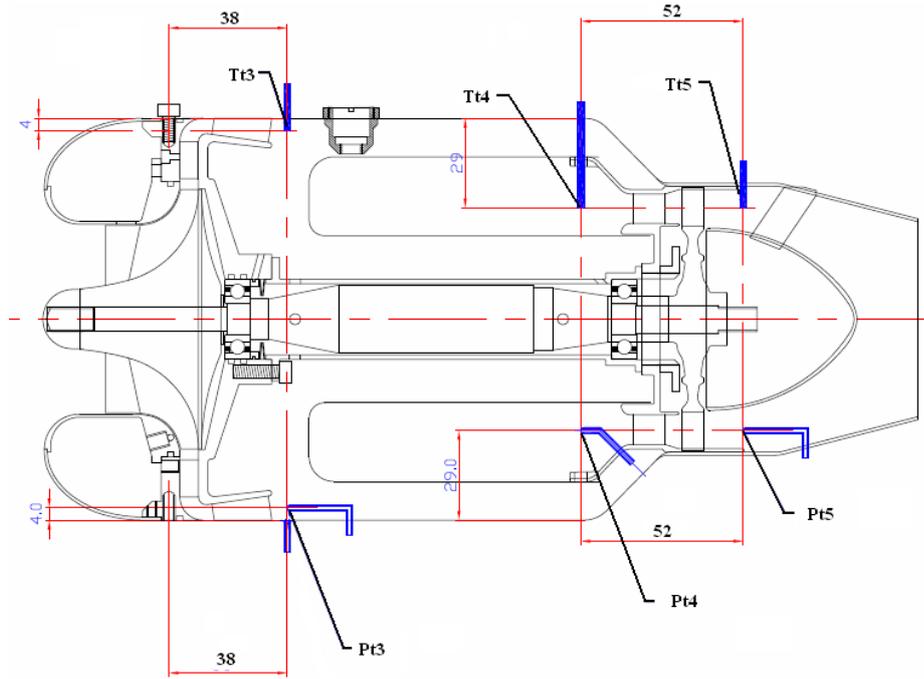


Figure 3-13 Measurement Points

Basic algorithm of the simulation code is demonstrated in Figure 3-15. When the code is started, it takes the initial conditions from experimental data. Initial values are used to calculate the first output data for the first step in numerical solution and in Beta Line Method. These initial conditions taken from experiments are given as follows:

Initial pressure ratio of the compressor:

$$\pi_C = 9.06 \cdot 10^{-21} \cdot N^4 - 1.365 \cdot 10^{-15} \cdot N^3 + 3.3806 \cdot 10^{-10} \cdot N^2 - 9.7556 \cdot 10^{-6} \cdot N + 1.1929 \quad (3.40)$$

Initial specific heat:

$$c_{Pair} = -1.9747 \cdot 10^{-28} \cdot N^6 + 8.386 \cdot 10^{-23} \cdot N^5 - 1.4447 \cdot 10^{-17} \cdot N^4 + 1.2943 \cdot 10^{-12} \cdot N^3 - 6.2674 \cdot 10^{-8} \cdot N^2 + 1.5743 \cdot 10^{-3} N + 9.8846 \cdot 10^2 \quad (3.41)$$

Initial turbine inlet temperature:

$$T_{t4} = 5.5099 \cdot 10^{-17} \cdot N^4 - 1.1969 \cdot 10^{-11} \cdot N^3 + 9.7144 \cdot 10^{-7} \cdot N^2 - 3.7423 \cdot 10^{-2}$$

$$\cdot N + 1.4273 \cdot 10^3 \quad (3.42)$$

Other initial condition values are given below:

- $P_{t2} = 93.38$ [kPa]
- $T_{t2} = 303$ [K]
- $P_{tref} = 100$ [kPa]
- $T_{tref} = 298$ [K]
- $M_9 = 1$ (Choke Condition)
- $M_0 = 0$ (Mach Number at Inlet)
- $H = 950$ [m] (for Ankara)
- $I = 1.787 \cdot 10^{-4}$ [kg*m²]
- $\gamma_c = 1.4$
- $\gamma_t = 1.35$
- $R = 286.9$ [J/(kg*K)]
- $\dot{m}_f = 0.00456$ [kg/s]

Using these initial data given above, simulation code calculates compressor input conditions with beta-line method. After that, compressor output temperature and pressure may be calculated. Combustion chamber output parameters are calculated by the aero thermal relations and differential equations which are solved by Runge-Kutte fourth order numerical solver. Turbine output conditions are calculated by using Figure 3-10. In the end, the shaft speed is computed and turbine input temperature, shaft angular velocity and thrust graphs are given by the code. Figure 3-14 and Figure 3-15 shows simulation input/outputs and flowchart of the simulation code respectively.

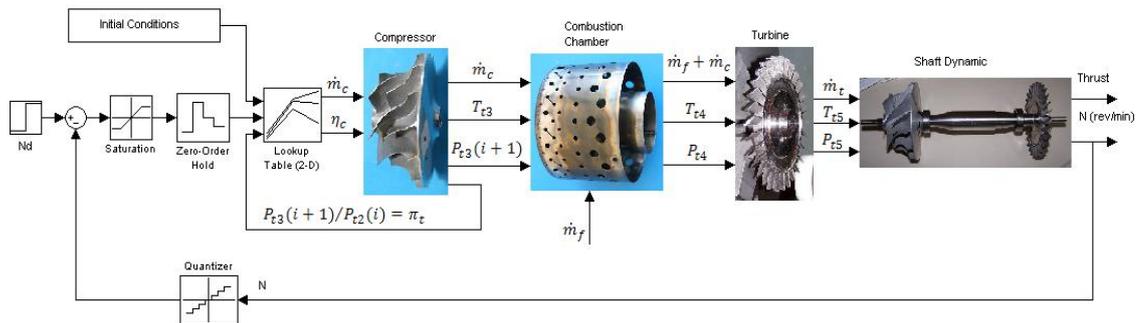


Figure 3-14 AMT Olympus HP Turbojet Simulation Inputs/Outputs

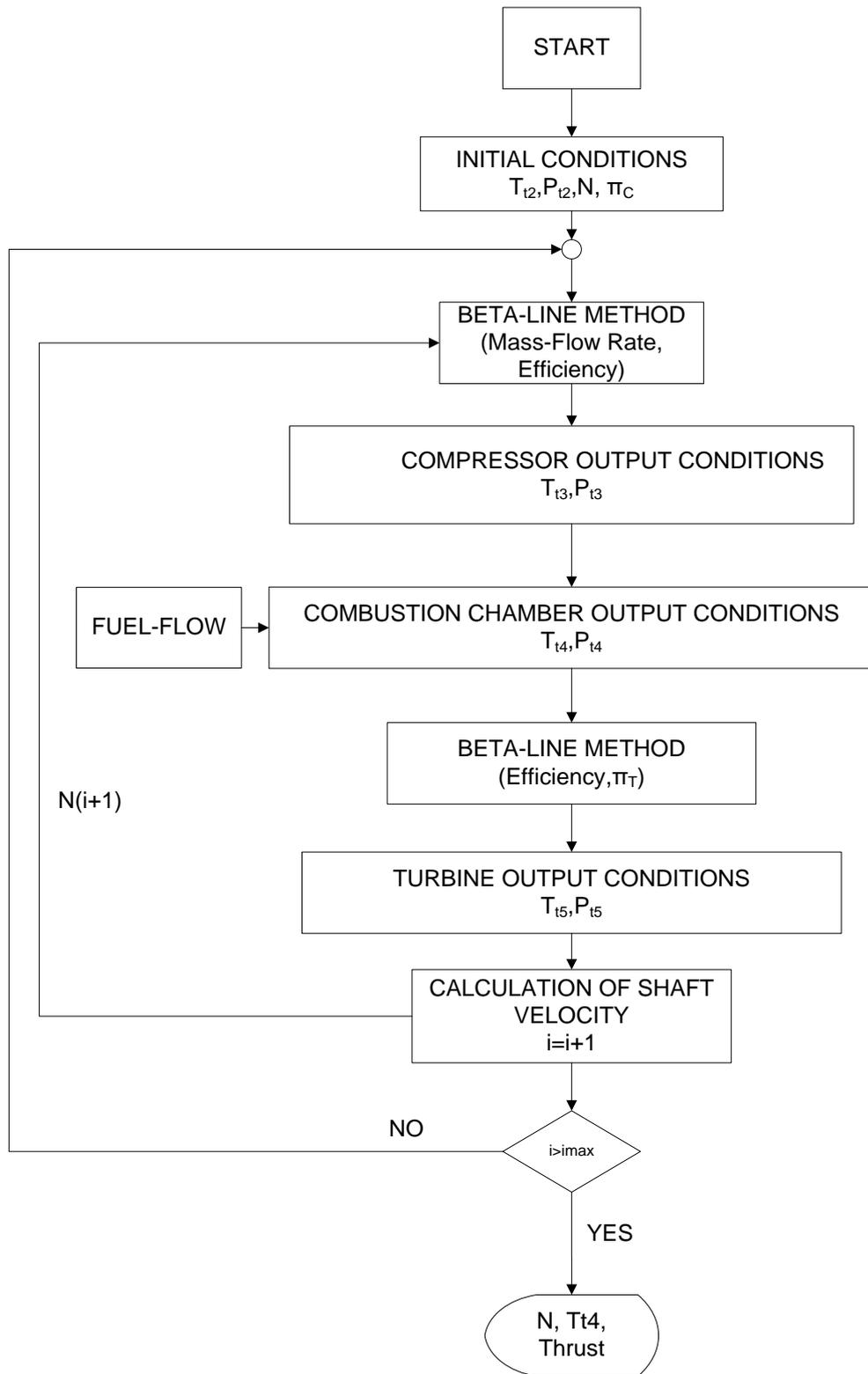


Figure 3-15 Flowchart of the AMT Olympus HP Turbojet Engine

When the simulation code is started, the following results which are given in Figure 3-16, Figure 3-17 and Figure 3-18 are established. The following scenario is selected in this simulation code. While the engine runs at 70 [krpm], 20% fuel flow increment is applied to the system and the results are compared with the reference data which are taken from experiments.

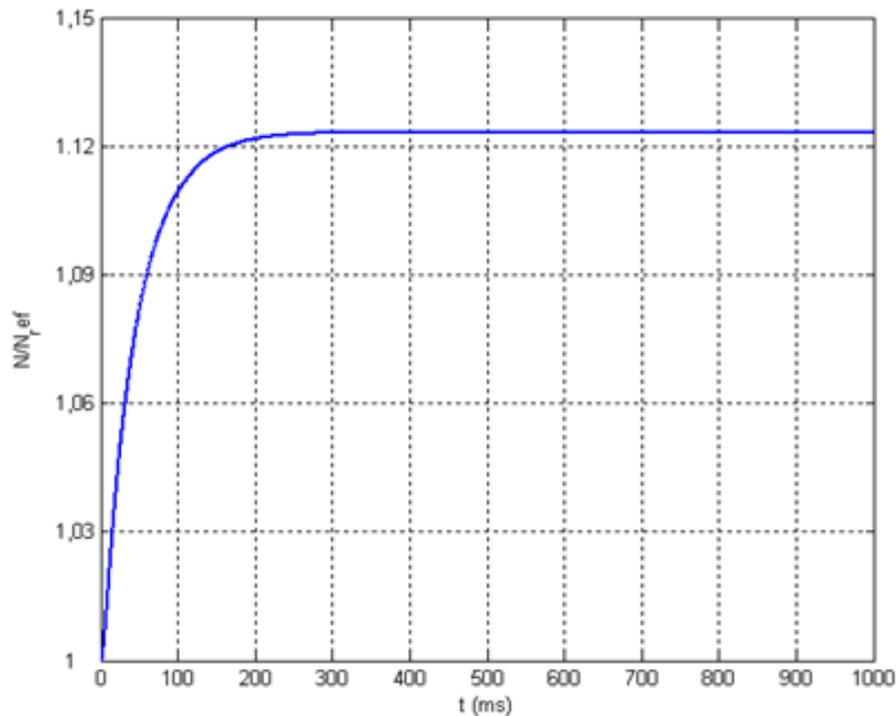


Figure 3-16 Shaft Speed Value Versus Time

As can be seen from the Figure 3-16, there is a 12% increment in shaft speed with respect to the reference shaft speed. On the other hand, it may be seen that small gas turbine engine behaves like a first order system. Total temperature after combustion chamber is given in Figure 3-17. It shows that temperature of the combustion chamber exit point is increasing 10 ms and then it decreases about 3% with respect to the reference temperature.

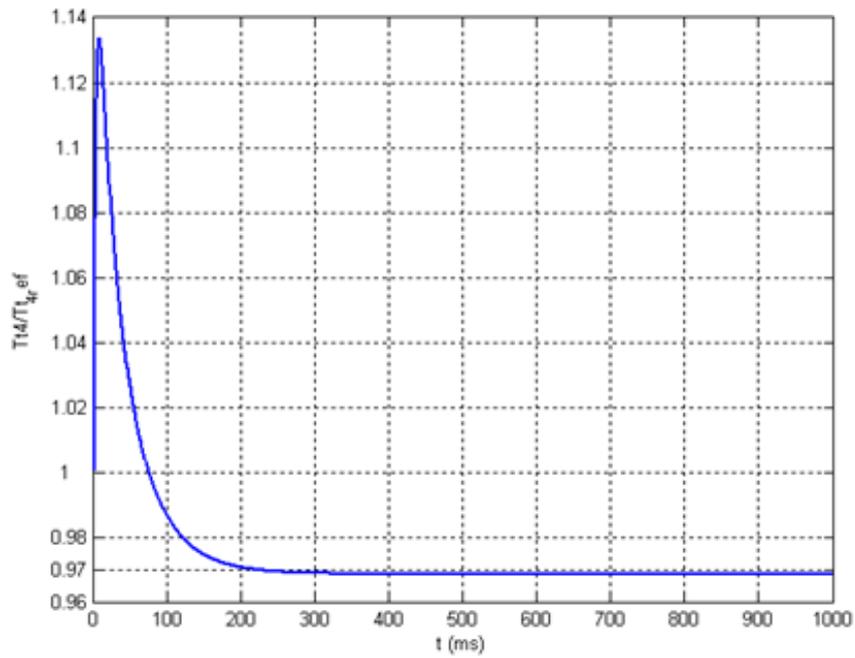


Figure 3-17 Total Temperature after Combustion Chamber

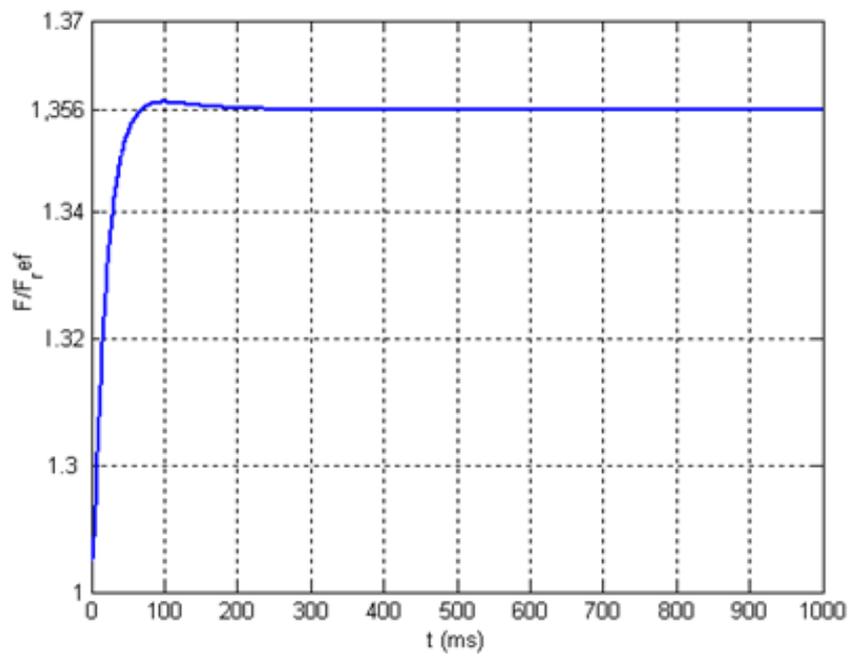


Figure 3-18 Thrust Output

Figure 3-18 gives the thrust increment with respect to the fuel flow increment. Simulation code results are compared with the experiments are given in Table 3-4.

Table 3-4 Comparison between Code and Test Results

	Experiment	Simulation	Reference Values
Thrust	% 40 increment	% 35 increment	60 [N]
T_{t4} (Turbine Inlet Temperature)	% 1.3 decrease	% 3 decrease	785 [K]
Shaft Speed	% 14 increment	% 12.5 increment	70 [krpm]

As can be seen from Table 3-4, simulation code works properly. It gives very close results to the test outputs. In order to design a controller, this simulation code may be used. In the following parts, PID and fuzzy logic controller were discussed and applied to the simulation code.

3.3 Closure

In this chapter, the mathematical modeling of the AMT Olympus HP turbojet engine was discussed. Individual components' dynamic equations and thermodynamic relations were developed by using conservation principals. Beta line map reading technique was discussed and applied to the compressor map. Nonlinear dynamic equations of the overall engine were developed and a simulation code is investigated. Flowchart of the simulation code was searched and given schematically. Finally, the simulation results were compared and verified with the test data. Results shows that the simulation code with reasonable assumptions gives outputs which are in good agreement with those of the experiments. In the following chapter, these simulation code and its results will be utilized to design a suitable controller for the miniature turbojet engine.

CHAPTER 4

CONTROL OF A SMALL TURBOJET ENGINE

4.1 Introduction

Gas turbine engine's mathematical model given in the preceding chapter is composed of aero-thermal and nonlinear differential equations. Gas turbines, which cannot model led by classical methods and have some parameters depend on look up tables, need effective controller for their all-operating points. There are many different methods in literature to control gas turbine engines. For example in [14], authors were deal with gain scheduling PID controllers in which they chose instantaneous linearization of their nonlinear equations. Y. Yu et al [13] used PI controller for the acceleration and temperature controls of their three-shaft gas turbine engine. On the other hand, in some studies, the researcher chose new control methods. For example, W. Airo et al [12] investigated fuzzy logic controller for a SR-30 turbojet engine.

In this chapter, PI and fuzzy logic controller will be discussed. Their advantages and disadvantages will be given. After suitable controller is designed, an appropriate digital controller will be selected and digital control circuitry along with a firmware will be implemented.

4.2 PI Controller Design

To understand a first- or a second order system's transient or steady state behavior, derivation of a mathematical model using standard modeling laws is necessary. Designers may employ these mathematical models to design a controller for their system. On the other

hand, for a gas turbine that has highly nonlinear in nature, it is difficult to find a readily available mathematical model of the system. In this thesis work, after establishing mathematical model of the AMT Olympus HP turbo jet engine, a PI controller is designed and simulated by using MATLAB environment. Because of nonlinearities and uncertainties (such as immeasurable time varying efficiencies in the differential equations), there are some restrictions about applying PI controller into the simulation in all operating region. Hence, PI parameters are found by trial-and-error for only one operating point and the results are compared with fuzzy logic controller outputs in the next section. MATLAB codes for PI controller are given in App. A-1. Schematic representation of PI controller is shown in Figure 4-1.

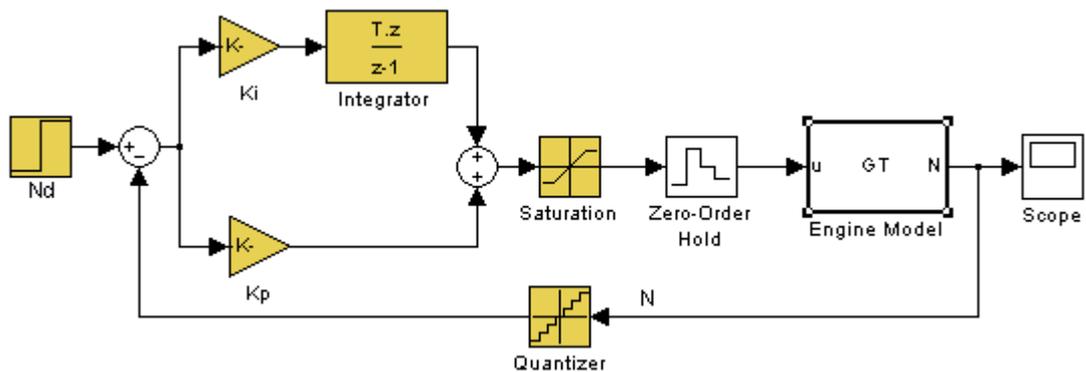


Figure 4-1 Schematic Representation of PI Controller

PI parameters and sampling time selected for simulation code is as follows:

- $K_p = 456 \cdot 10^{-12}$
- $K_I = 35 \cdot 10^{-11}$
- $t = 2$ [ms]

Simulation results for step input 85 [krpm] are given in Figure 4-2. Initial conditions for this step input are taken from the experiments. AMT Olympus HP turbojet engine reaches the neighborhood of 85 [krpm]. As can be seen from Figure 4-2, the shaft speed reaches 85 [krpm] at steady state. Thus, PI controller works properly for this condition. On the other hand, PI parameters have to adjust for all working conditions. For this purpose, J. Mu et al

[14] and Jie M.S. et al [23] were utilized gain scheduling method and fuzzy logic controller respectively for their nonlinear system. Temperature after combustion chamber increases about 9 ms and it decreases after this point. In steady state conditions, temperature after combustion chamber decreases to 750 K from 785 K which is about 4.45%. Experiment showed that temperature after combustion chamber decreases about 4.15% thus simulation gives proper temperature output for these conditions. Figure 4-2 shows that thrust reaches 76 N at steady-state. This value was found 69 N from the experiments. This difference between the model and experiment is occurred from the assumptions and measurement errors.

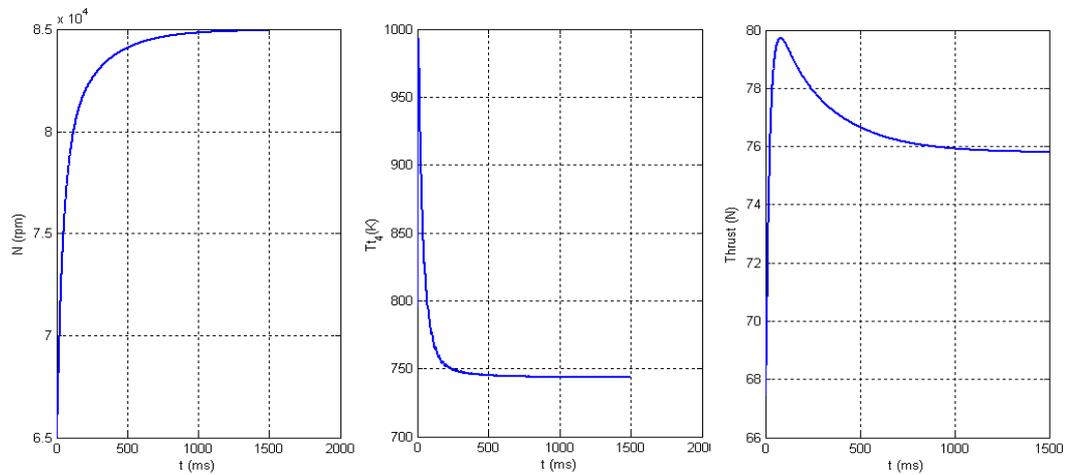


Figure 4-2 Shaft Speed, Tt4 and Thrust Output Graphs

In this study, fuzzy logic controller is utilized to manipulate the engine at all operating points. On the other hand, if initial values are replaced by $N_{\text{initial}} = 77$ [krpm], $m_f = 6.5$ [g/s] the simulation results changes such as the one shown in Figure 4-3.

Figure 4-3 illustrates that for initial conditions given above and step input 85 [krpm], the shaft speed makes an overshoot and it reaches to almost 93 [krpm]. Furthermore, thrust output decreases within the first 150 ms and reaches the steady state point of 76 [N]. To solve these overshoot and thrust decrement problem, the parameters of the PI controller and simulation must be changed. Hence, the PI controller without gain tuning or etc. is not effective controller for this kind of nonlinear gas turbine systems.

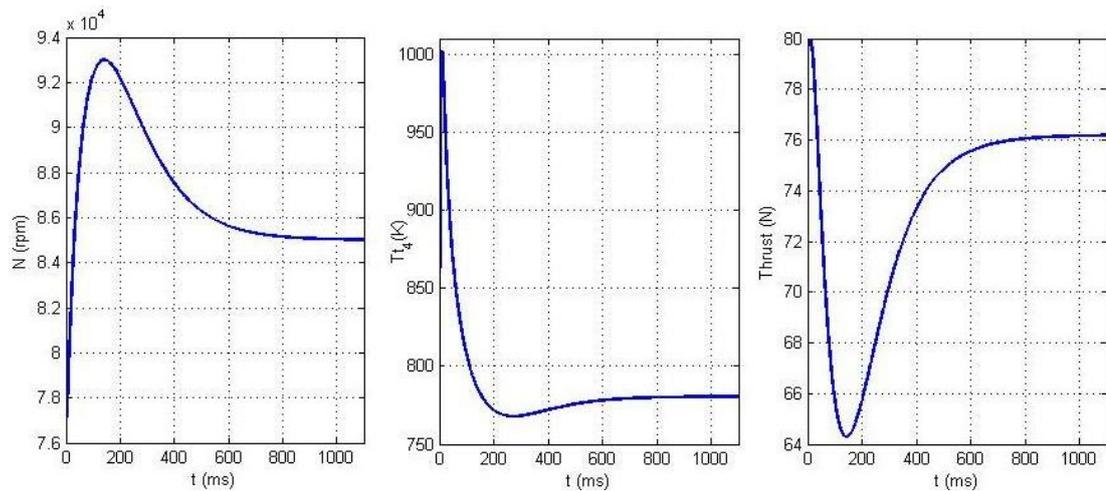


Figure 4-3 Shaft Speed, Tt4 and Thrust Output Graphs for Different Initial Conditions

4.3 Fuzzy Logic

Fuzzy logic is a practical method to control nonlinear system by using *a priori* knowledge. This heuristic information is taken from an operator who works in the process need to be controlled. In fuzzy logic control, there are two methods to develop control algorithms. First, the control engineer asks the operator how s/he controls the process and the answers are employed for the fuzzy rules. Second, the designer develops a mathematical model to understand the behavior of system. Sometimes, improving the mathematical model of the system is not enough for the conventional controller design. Because of the complexities and restrictions in the model, the designer needs to make some assumptions. Thus, traditional controllers have to utilize simple mathematical model of the main system. Prof. L.A. Zadeh introduced the fuzzy logic theory in 1965 [27, 29, 30, 31]. Fuzzy logic theory makes computer or microprocessor emulate the thought of a humanbeing. Classical computers employ Boolean Logic 1: true, 0: false. There is not any value for “almost true.” A simple fuzzy logic controller block diagram is shown in Figure 4-4. As can be seen from this figure, fuzzy logic controller is embedded in a closed-loop control system. Plant output is denoted by $y(t)$, its inputs are denoted by $u(t)$, and reference input to the fuzzy logic controller is denoted by $r(t)$. The fuzzy logic controller has four principle components: (1) the “rule-base” holds the knowledge, in the form of a set of rules. (2) The inference mechanism evaluates which control rules are relevant at the current time. (3) The fuzzification interface modifies

the input to use in rule-base. (4) The defuzzification interface converts the conclusion of fuzzy logic controller into the form of appropriate input for the plant [27].

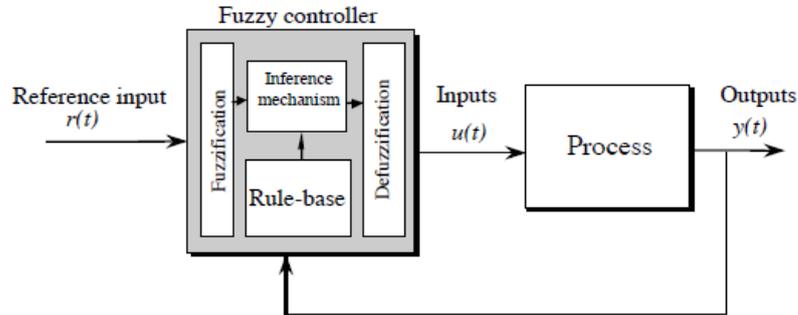


Figure 4-4 Fuzzy Logic Controller [27]

For example, human age might be classified as middle age between 40 and 60. If this difference was defined in a computer, corresponding figure is demonstrated in Figure 4-5. As can be seen from the Figure 4-5 (a) transition in the boundaries is too sharp. This means that if a person who is at the end of his age 39 will not be in middle age region. On the other hand, if this question was asked to a human, s/he would give the answer given in Figure 4-5 (b). This figure shows that human beings think that a person who is 25 years old is in middle age class but he or she has a degree of middle age between 0 and 1. This degree is called as membership value. For example, a 40 years old person is almost 50% (means 0.5) in middle age and he or she is still in young person region.

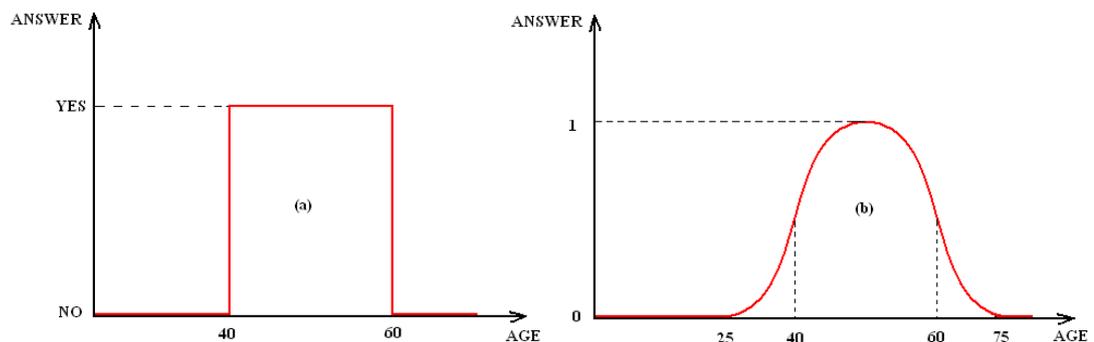


Figure 4-5 Definition of Middle Age in Conventional Sets and Fuzzy Sets

Because of the rule-based operation, several inputs can be processed and numerous outputs generated. Prof. Zadeh proposed the concept linguistic variables. He thought that variables of the systems could be evaluated as linguistic terms rather than numerical values. The sensor input of the system is a noun, for instance, “temperature”, “displacement”, “velocity”, “pressure” etc. Since the error is only the difference, it can be considered similarly. The fuzzy variables are adjectives (“large positive” error, “zero” error, “negative” error and etc.). At least, designer has to simply have “positive”, “zero” and “negative” variables for each of the parameters. Additional ranges such as “very large” and “very small” could also be calculated to enlarge the response capability of the controller for highly nonlinear systems.

4.3.1 Fuzzy Rules

Linguistic rules telling the control system consist of two parts: an antecedent block (between the “if” and “then”) and a consequent block (following “then”). Depending on the system, it may be necessary to calculate every possible input combination since some may rarely occur. By making, this type of evaluation fewer rules can be evaluated [28]. Fuzzy rules might be developed by using the experimental data or simulation of the system. For example, a fuzzy logic controller for an oven might have the following rules:

- If the temperature is HIGH, the heat is LOW.
- If the temperature is LOW, the heat is HIGH.

These rules may be increased. It depends on the number of experiments and the accuracy of the controller. Some experiments and their results for AMT Olympus HP turbojet engine are given in Chapter 6. These experiments are utilized to design suitable fuzzy rules for the controller. The relation between the shaft speed and pump voltage is shown in Figure 4-6. Using this figure, basic fuzzy rules can be developed. For groups of point A, B and C simple fuzzy rule can be designed as follows:

- Point A1 shows that the desired input increases in two seconds while shaft speed cannot increase as fast as the input. Thus, it starts to rise to follow the input. For example, for $t = 5$ [s] error between shaft speed and the input signal is positive while change in error is negative. This position shows that it does not necessary to apply a manipulation to the system due to the behavior of the shaft speed, which converges to the input. Thus first rule becomes; “If $en(i)$ is P and $cen(i)$ is N then Δu is

Z” where **en** is error, **cen** cen is change in error, P is positive, N is negative, Z is zero and Δu is manipulation value.

- Point B1 shows that pump voltages reduces to 1.1 V and shaft speed follows the input. For this point, error between shaft speed and input is negative although change in error is positive. For $t = 30$ [s] shaft speed reduces to 38 [krpm] and it follows the input at $t = 32$ [s]. For $t = 30$ [s], error descends to zero. Thus, it does not necessary to manipulate the system for this point. Second rule becomes; “If **en (i)** is N and **cen (i)** is P then Δu is Z”.

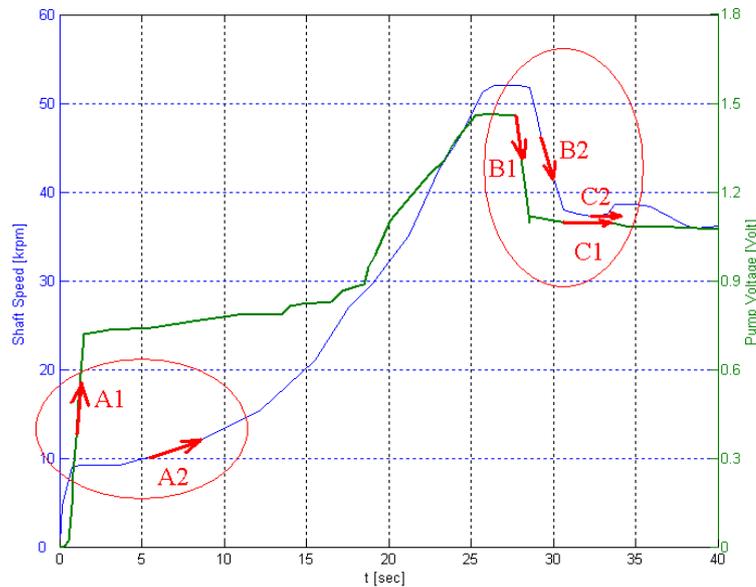


Figure 4-6 Relation between the Shaft Speed and Pump Voltage of the AMT Olympus HP Turbojet Engine

All points are evaluated such as given above and fuzzy rules and rule-table used in the AMT Olympus Hp turbojet engine is given below;

- If **en (i)** is P and **cen (i)** is P then Δu is P
- If **en (i)** is N and **cen (i)** is N then Δu is N
- If **en (i)** is P and **cen (i)** is Z then Δu is P
- If **en (i)** is N and **cen (i)** is Z then Δu is N
- If **en (i)** is P and **cen (i)** is N then Δu is Z
- If **en (i)** is N and **cen (i)** is P then Δu is Z

- If **en** (i) is Z and **cen** (i) is Z then Δu is Z
- If **en** (i) is Z and **cen** (i) is N then Δu is N
- If **en** (i) is Z and **cen** (i) is P then Δu is P

Table 4-1 summarizes fuzzy rules being developed. On this table, N, P, Z mean negative, positive and zero respectively. Similarly, **en** and **cen** represent error and change in error respectively. Δu is defuzzification result. MATLAB code for fuzzy logic controller is given in App. A.

Table 4-1 Fuzzy Rule Table

Input (Δu)		Change in Error (cen)		
		+	0	-
Error (en)	+	P	P	Z
	0	P	Z	N
	-	Z	N	N

4.3.2 Membership Functions

Membership function is a graphical representation of the magnitude of participation of each input. The rules utilize the input membership values as weighting factors to determine their influence on the fuzzy logic output sets of the final output conclusion [27, 28]. Different membership functions are used in literature. For example, most common membership functions are given in Figure 4-7. Shape of the membership function is not very important. Most important point is their number and their relative position. As can be seen from Figure 4-7 (a), (b), (c) triangular, trapezoidal and Gaussian membership functions may be utilized. Membership value $\mu(x)$ changes between 0 and 1. Zero means 0% and 1 means 100% for the control processes. Membership function of the AMT Olympus HP turbojet engine's shaft speed error, change in error and output are shown in Figure 4-8, Figure 4-9 and Figure 4-10 respectively where triangular and trapezoidal forms are utilized. Figure 4-8, Figure 4-9 and Figure 4-10 shows that **en**, **cen** and Δu changes between -1 and +1. This means that error, change in error values must be multiplied by some constants, which are selected as

maximum shaft speed of the engine taken from experiments. Output values are multiplied by the “**de**” constant, which is found from the trial and error method.

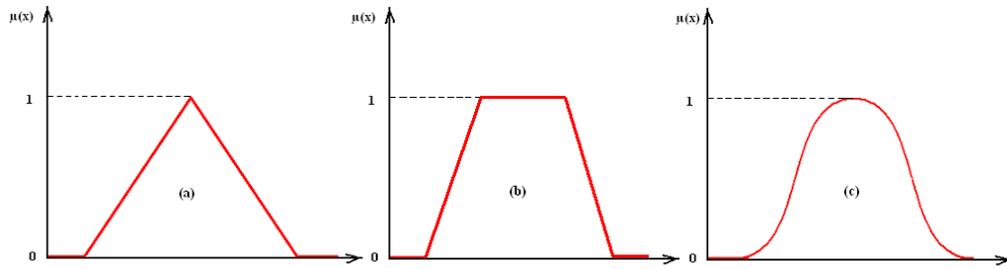


Figure 4-7 Triangular, Trapezoidal and Gaussian Membership Function

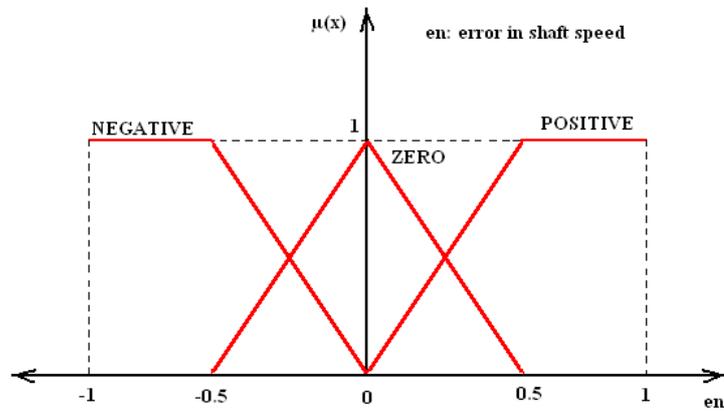


Figure 4-8 Membership Function of the Error of the Shaft Speed

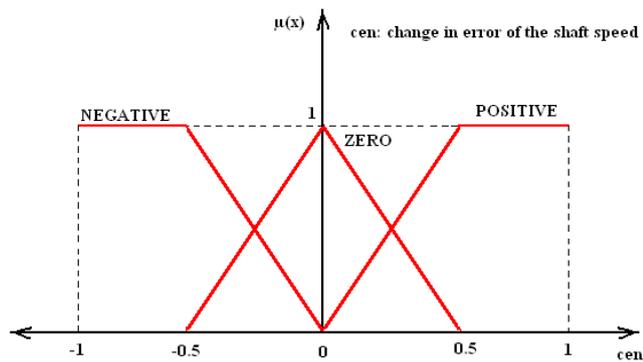


Figure 4-9 Membership Function of the Change in Error of the Shaft Speed

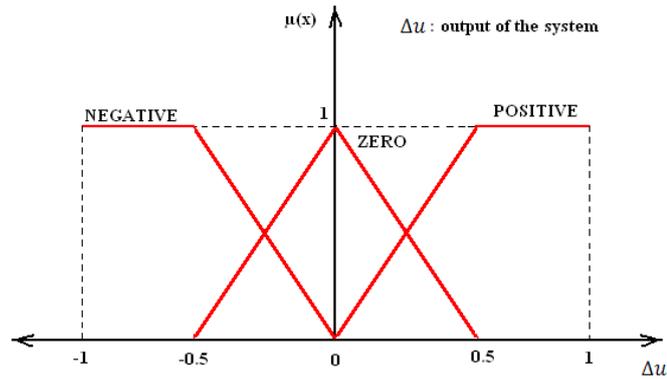


Figure 4-10 Membership Function of the Output

4.3.3 Defuzzification

Different defuzzification methods are employed in literature. In this study, center of area method was used for defuzzification process. In defuzzification process of the fuzzy logic control, controller calculates the manipulation value of the fuzzy logic controller. This manipulation value is found from the area of the membership function of the output. Block diagram of the fuzzy logic controller is given in Figure 4-11. As can be seen from Figure 4-11, the desired shaft speed values “ N_d ” goes into the system. After subtraction, error and the change in error enter to the fuzzy rule block diagram. At this point fuzzy logic makes decision how to manipulate to the system for this input data. Finally “ Δu ” multiples by “ de ” and “ u ” are calculated. This “ u ” value is utilized as a manipulated input to the plant.

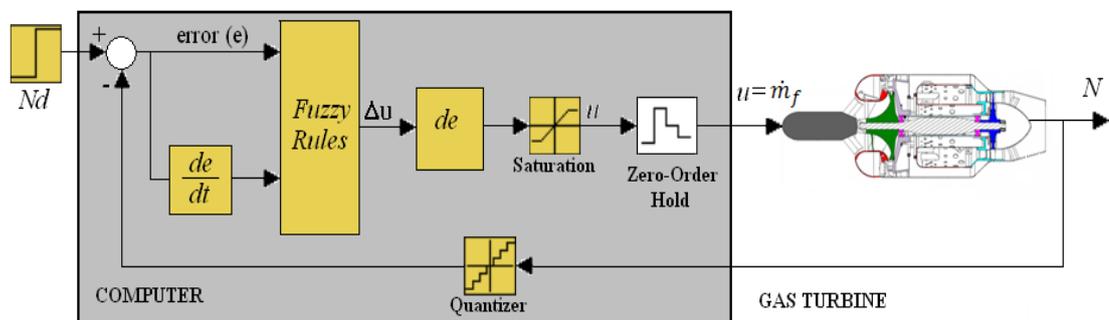


Figure 4-11 Fuzzy Logic Controller Block Diagram

4.3.4 Fuzzy Logic Simulation

Fuzzy logic controller simulation results are given in Figure 4-12, Figure 4-13 and Figure 4-14. This simulation code is given in App. A-2. In this code, input value is 85 [krpm] and, the results show that fuzzy logic controller manipulates the model correctly and speed value reaches to 85 [krpm] in 3 second from 70 [krpm].

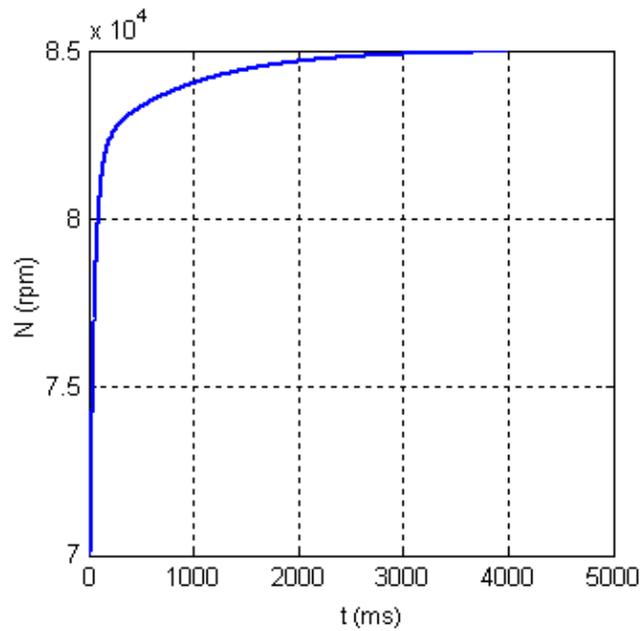


Figure 4-12 Shaft Speed Output

Temperature decreases to 745 K at steady state condition. On the other hand, the temperature value reaches almost 1000 K that is under the critical temperature 1100 K. This behavior has to be limit by the controller because most dangerous conditions may cause the engine to fail.

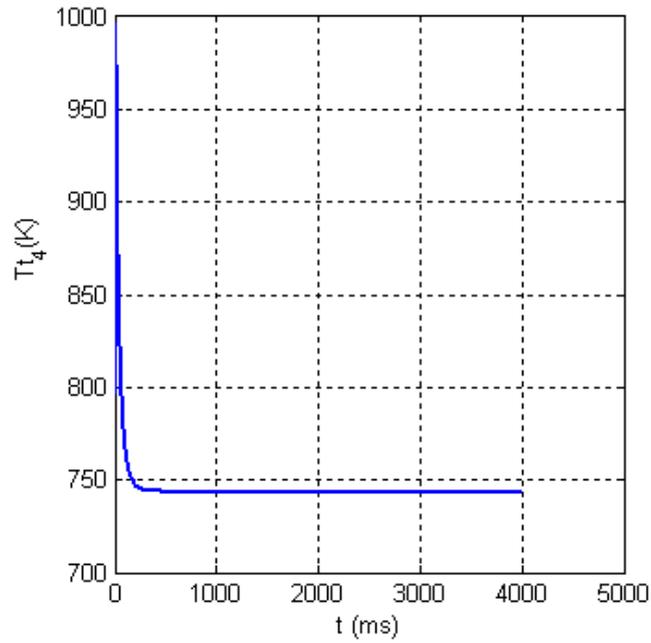


Figure 4-13 Tt4 Output

As can be seen from Figure 4-14, maximum thrust reaches 79 N in transient region. After system progresses towards the steady state conditions, thrust value decreases to almost 76 N.

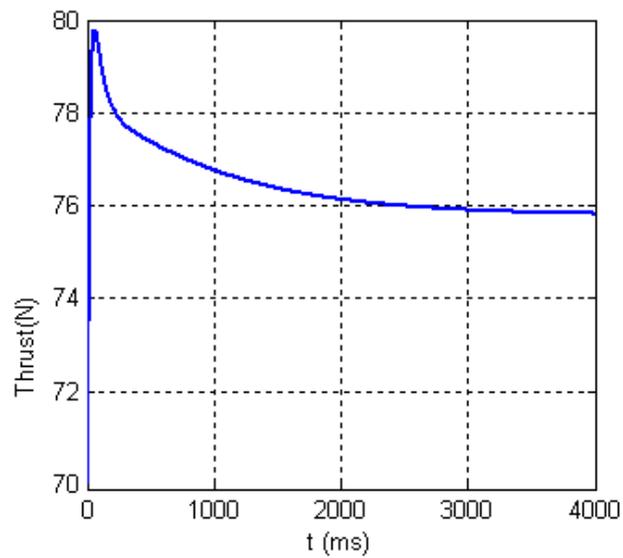


Figure 4-14 Thrust Output

Figure 4-12, Figure 4-13, and Figure 4-14 show that fuzzy logic controller regulates the gas turbine similar to the PI controller. As can be seen from Table 4-2, shaft speed, T_{t4} and thrust values are very close to the each other. There is only one difference between PI and fuzzy logic controller is that, maximum T_{t4} value reaches 1000 K with fuzzy logic controller while it is 990 K with PID controller. Difference between conventional controller and fuzzy logic controller for a second order system with different parameters was given in the following section.

Table 4-2 Comparison of PI and Fuzzy Logic for the Same Input at Steady State

	Shaft Speed [RPM]	T_{t4} [K]	Thrust [N]
PID	85000	750	75
Fuzzy Logic	85000	745	76

4.3.5 Comparing PID and Fuzzy Logic Controller for a Second Order System

To understand the similarities and differences between classical controller PID and fuzzy logic, a simple experiment is to be conducted. In this experiment, a trapezoidal input is applied to a second order system for different ζ and ω_n values. Response of the system and the error are compared for PID and fuzzy logic controller. In previous chapter, it is stated that gas turbine engines have nonlinear differential equations and parameters, which depend on lookup tables such as compressor efficiency, mass flow rate, compressor pressure ratio, turbine pressure ratio etc. Hence, the mathematical model of the gas turbine has time varying constants and they change the system behavior. For example, a second order system given in Eq.(4.1) has two important parameters ζ and ω_n , damping ratio and natural frequency respectively. In the following section, these two parameters will change while control system parameters remain constant. Thus, parameter variation will be simulated simply. Second order system equation for a car throttle angle vs. speed relation becomes;

$$G(s) = \frac{y(s)}{u(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (4.1)$$

where $y(s)$ denotes output of the system (for example speed of a car [m/s]), while $u(s)$ denotes input (for example throttle angle of the engine).

4.3.5.1 Overdamped System with $\zeta=1.5$, $\omega_n = 10$ [rad/s]

In this section, an overdamped control system is discussed. System parameters are given in the following:

- For overdamped case ($\zeta = 1.5$)
- PID Parameters: $\omega_n = 10$ [rad/s], $K_p = 0.01$, $K_d = 0.01$, $K_i = 5$
- Fuzzy Parameters: $\omega_n = 10$ [rad/s], $de = 25$

Transfer function of the system is;

$$G(s) = \frac{y(s)}{u(s)} = \frac{100}{s^2 + 30s + 100} \quad (4.2)$$

Using standard z-transform method with ZOH, discrete time model of the system can be derived as follows (sampling time: 0.1 second);

$$G(z) = \frac{y(z)}{u(z)} = \frac{0.2134z + 0.08097}{z^2 - 0.7555z + 0.04979} = \frac{0.2134z^{-1} + 0.08097z^{-2}}{1 - 0.7555z^{-1} + 0.04979z^{-2}} \quad (4.3)$$

Corresponding constant coefficient difference equation is;

$$y(k) = 0.7555 \cdot y(k-1) - 0.04979 \cdot y(k-2) + 0.2134 \cdot u(k-1) + 0.08097 \cdot u(k-2) \quad (4.4)$$

Using Eq. (4.4), the outputs presented in Figure 4-15 are obtained. As can be seen from the Figure 4-15, PID and fuzzy logic controller outputs are similar to those with the given conditions and parameter. As can be seen from the error graph, PID error is less than the fuzzy logic error. Fuzzy logic controller manipulates the second order system slower than PID controller does.

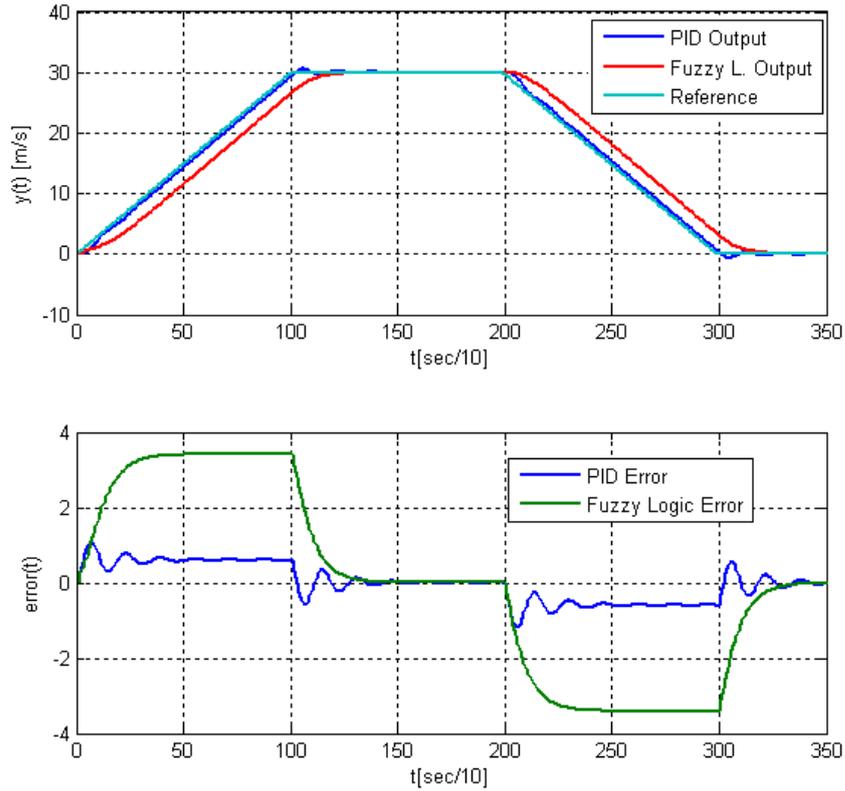


Figure 4-15 Overdamped System Results for PID and FLC ($\zeta=1.5$, $\omega_n=10$)

4.3.5.2 Underdamped System with $\zeta = 0.1$, $\omega_n = 10$ [rad/s]

In this section, an underdamped control system is discussed. System parameters are given in the following:

- For underdamped case ($\zeta=0.1$)
- PID Parameters: $\omega_n=10$ [rad/s], $K_p=0.01$, $K_d=0.01$, $K_i=5$
- Fuzzy Parameters: $\omega_n=10$ [rad/s], $de=25$

Transfer function of the system is

$$G(s) = \frac{y(s)}{u(s)} = \frac{100}{s^2+2s+100} \quad (4.5)$$

Using standard z-transform ($T=0.1$ second);

$$G(z) = \frac{y(z)}{u(z)} = \frac{0.431z+0.4023}{z^2-0.9854z+0.8187} = \frac{0.431z^{-1}+0.4023z^{-2}}{1-0.9854z^{-1}+0.8187z^{-2}} \quad (4.6)$$

Corresponding constant coefficient difference equation is;

$$y(k) = 0.9854 \cdot y(k - 1) - 0.8187 \cdot y(k - 2) + 0.431 \cdot u(k - 1) + 0.4023 \cdot u(k - 2) \quad (4.7)$$

Using Eq. (4.7), the outputs given in Figure 4-16 are determined. Figure 4-16 shows that the PID Controller is not following the trapezoidal input for these initial conditions and parameters. It oscillates and exceeds the reasonable limits. After choosing suitable integral constant ($K_i = 2$), it works properly. Results for $K_i = 2$ is shown in Figure 4-17. It is obvious that the PID controller can control the system if its parameters are replaced by proper values for varying situations. However, fuzzy logic controller's parameters are still remains constant for varying system parameters.

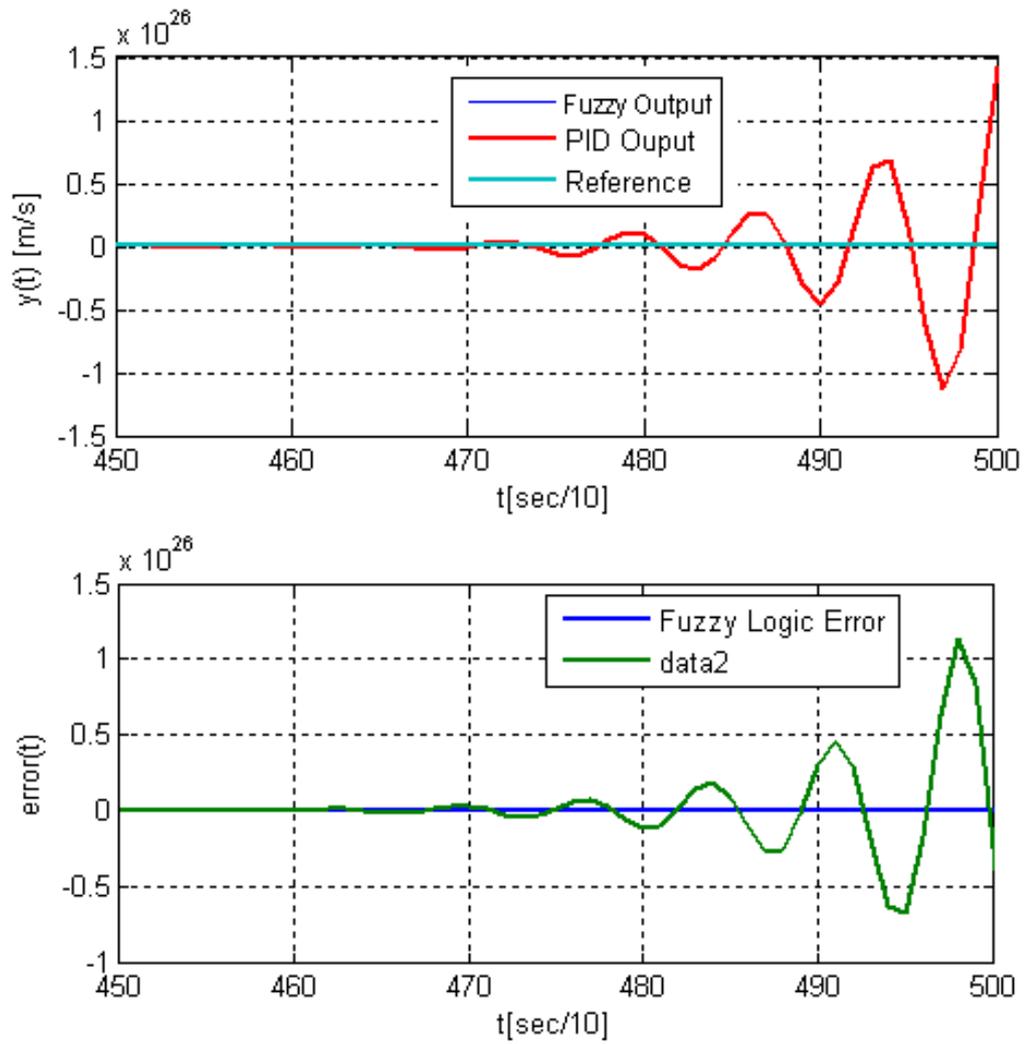


Figure 4-16 Underdamped System Results for PID and FLC ($\zeta=0.1$, $\omega_n =10$ [rad/s])

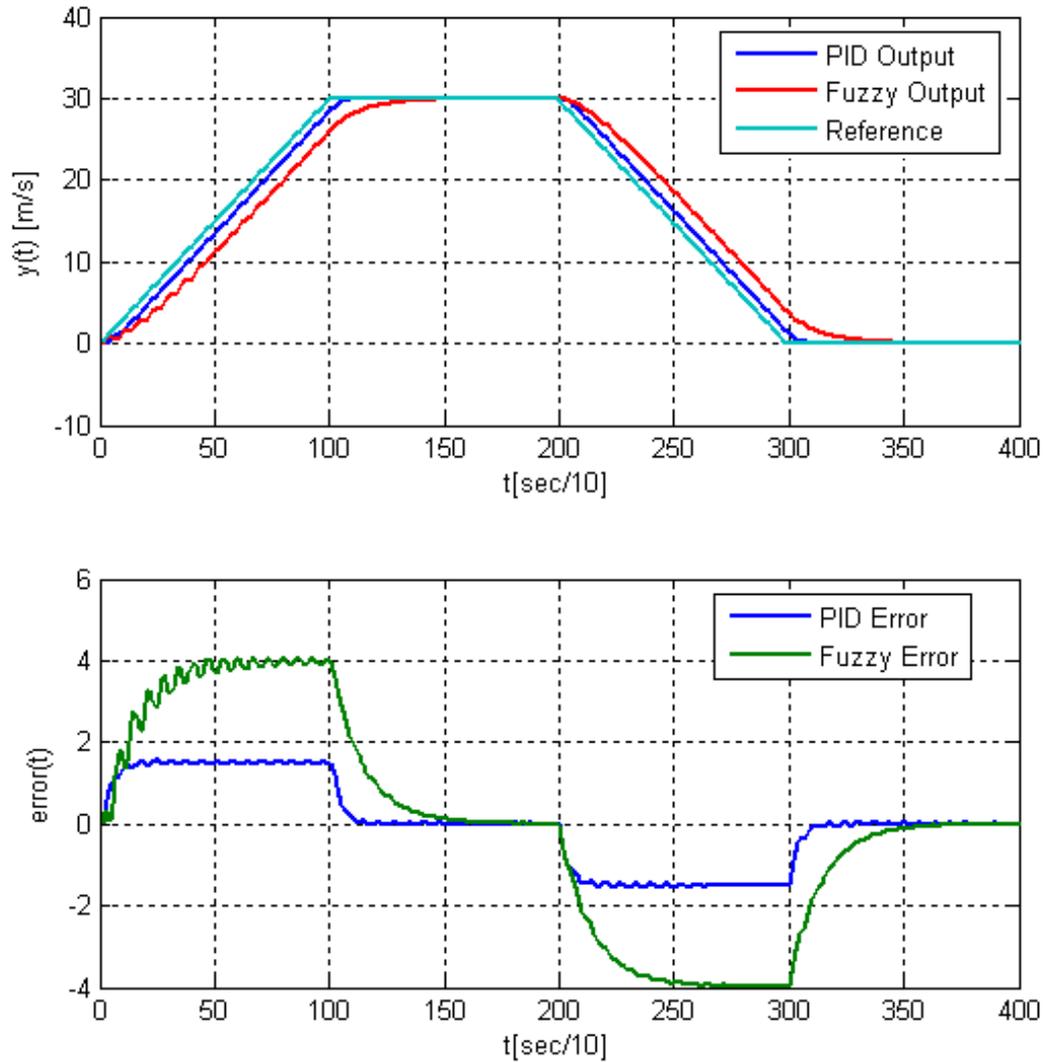


Figure 4-17 Underdamped System Results for PID and FLC ($\zeta=0.1$, $\omega_n=10$ [rad/s], $K_i=2$)

4.3.5.3 Underdamped System with $\zeta=0.6$, $\omega_n = 1$ [rad/s]

In this section, an underdamped control system is discussed. System parameters are given in the following:

- For underdamped case ($\zeta = 0.6$)
- PID Parameters: $\omega_n = 1$ [rad/s], $K_p = 0.01$, $K_d = 0.01$, $K_i = 5$
- Fuzzy Parameters: $\omega_n = 1$ [rad/s], $d_e = 25$

Transfer function of the system is

$$G(s) = \frac{y(s)}{u(s)} = \frac{1}{s^2+1.2s+1} \quad (4.8)$$

Using standard z-transform (T=0.1 second);

$$G(z) = \frac{y(z)}{u(z)} = \frac{0.004802z+0.004614}{z^2-1.878z+0.8869} = \frac{0.004802z^{-1}+0.004614z^{-2}}{1-1.878z^{-1}+0.8869z^{-2}} \quad (4.9)$$

Corresponding constant coefficient difference equation is;

$$y(k) = 1.878 \cdot y(k-1) - 0.8869 \cdot y(k-2) + 0.004802 \cdot u(k-1) + 0.004614 \cdot u(k-2) \quad (4.10)$$

Using Eq. (4.10), the outputs presented in Figure 4-18 are found. Figure 4-18 shows that the PID Controller is not following the trapezoidal input for these initial conditions and parameters such as Figure 4-16. It oscillates and exceeds the reasonable limits again. After choosing suitable constants ($K_p=1, K_d=2, K_i=2$), it works properly. Results for $K_p=1, K_d=2$ and $K_i=2$, is shown in Figure 4-19.

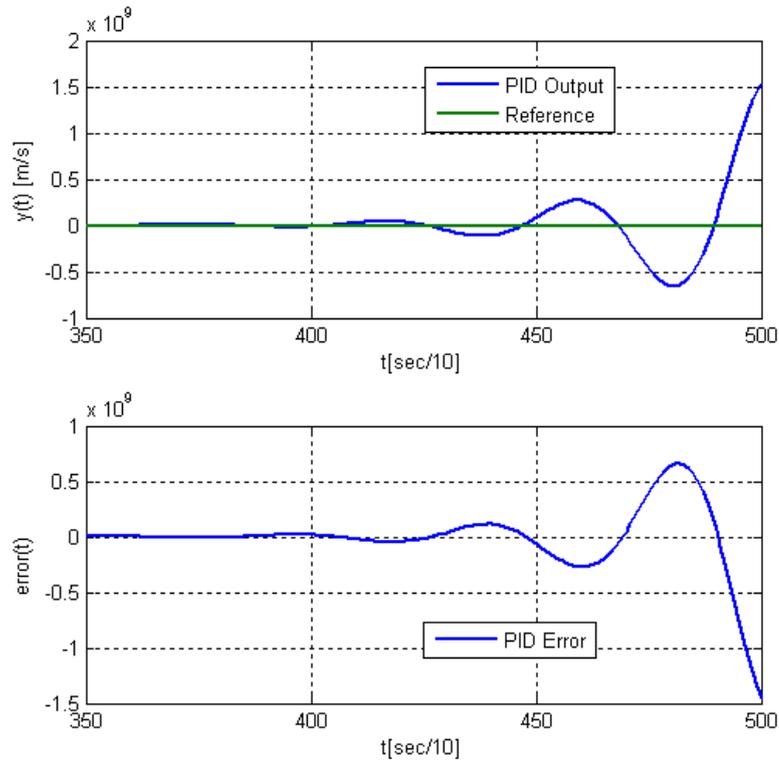


Figure 4-18 Underdamped System Results for PID and FLC ($\zeta=0.6, \omega_n=1$ [rad/s])

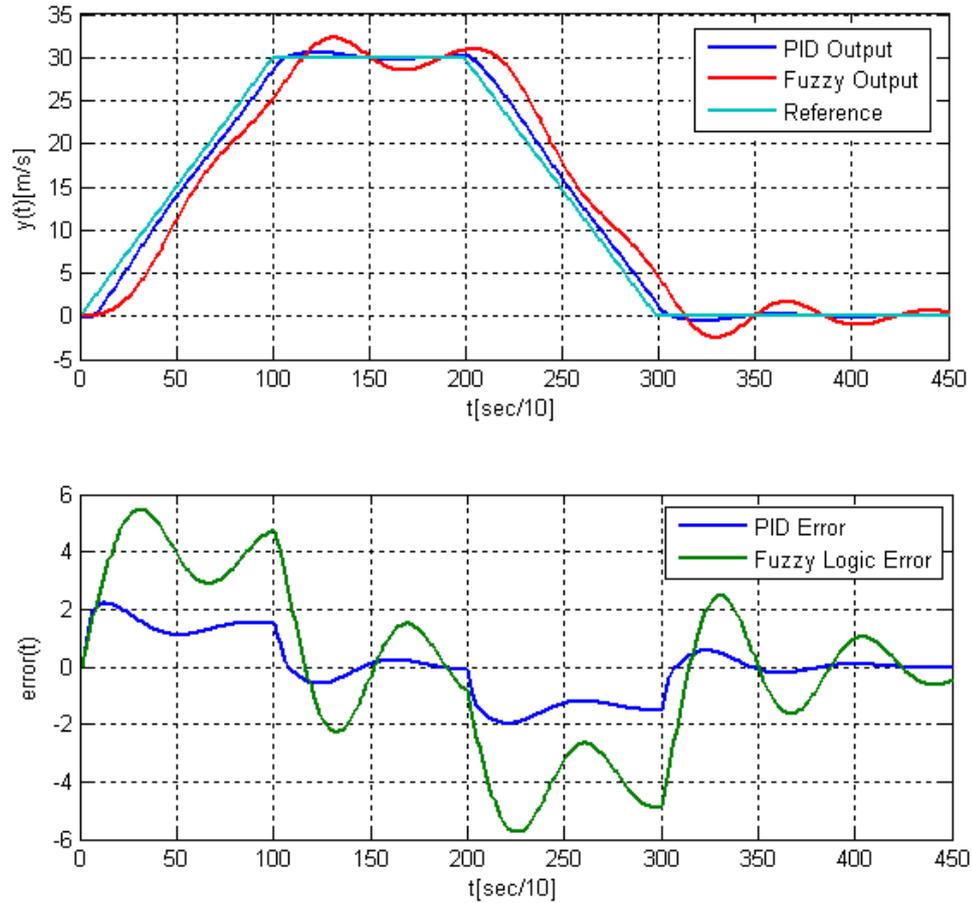


Figure 4-19 Underdamped System Results for PID and FLC ($\zeta=0.6$, $\omega_n=1$ [rad/s], $K_p=1$, $K_d=2$, $K_i=2$)

As a result, Figure 4-16 and Figure 4-18 show that with different ζ and ω_n , the PID parameters need to be replaced by the suitable values while fuzzy logic parameters remains constant. On the other hand, simulation results show that if PID parameters are set for varying situation, its steady state error is less than fuzzy logic controller error. This problem may be solved by trial-and-error method using different parameters or using more precise membership function that can manipulate system faster with precision. For gas turbine engines, compressor-working conditions always change with the characteristic map of the compressor. These different inputs for example, efficiency, pressure ratio, and mass flow rate may cause the system failure. Consequently, the fuzzy logic controller may adapt to the different conditions and it sets itself in order to give convenient manipulation.

4.4 Closure

In this chapter, PI, PID and fuzzy logic control algorithms are designed and compared. First of all PI controller is designed and embedded into the simulation. Secondly, fuzzy logic method is discussed and embedded into the simulation. Results of these two controllers are compared. It was observed that the fuzzy logic controller can be applied to the small gas turbine engine. Finally, a second-order system with varying natural frequency and damping ratio is investigated. Fuzzy logic and PID controllers are designed and compared for this second order system. Results show that if the system parameters change, the PID controller parameters need to be replaced by the suitable parameters while fuzzy logic controller parameters remain constant.

CHAPTER 5

DESIGN AND IMPLEMENTATION OF ELECTRONIC CONTROL UNIT

5.1 Introduction

In previous chapter, PI, PID and fuzzy logic controllers are discussed and their performances are compared under different conditions. Due to its apparent advantages, fuzzy logic controller is chosen for the engine control hardware. Amt Olympus HP turbojet engine has only one manipulated input, which is the mass flow rate of the kerosene flowing into the combustion chamber. Note that the fuzzy controller in simulation controls this flow rate. In this section, the controller hardware design for AMT Olympus HP turbojet engine is investigated. First, a circuit along with its PCB layout is designed. Then the corresponding firmware (control software) is developed utilizing C language for this embedded system. Details of the design follow.

5.2 Electronic Control Unit of AMT Olympus HP

Inputs and outputs for the original electronic control unit of AMT Olympus HP engine are given in Table 5-1. There are four inputs and five outputs of the AMT Olympus. Electronic Control Unit (ECU) of the gas turbine adjusts the shaft speed of the engine with respect to the throttle valve position. In fact, the ECU controls the max. temperature and max. shaft speed of the engine under all operating conditions. It takes temperature feedbacks from a K-type temperature sensor and shaft speed feedback from an inductive proximity sensor.

Table 5-1 Inputs and Outputs of the Electronic Control Unit

	INPUTS	OUTPUTS
KEROSENE PUMP		*
KEROSENE VALVE		*
PROPANE VALVE		*
THROTTLE	*	
TEMPERATURE	*	
GLOW PLUG		*
SHAFT SPEED	*	
ELECTRIC STARTER		*
ON/OFF SWITCH	*	

Original controller of the AMT Olympus HP turbojet engine is shown in Figure 5-1. It has one thermocouple input, one speed input and one throttle input. On the other hand, controller has one each fuel output, glow plug output, propane, kerosene and electric starter output.

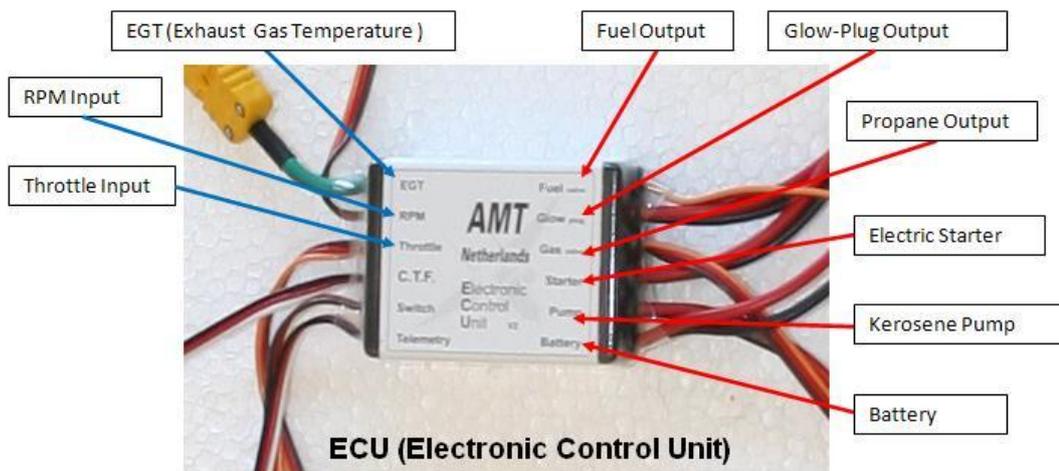


Figure 5-1 AMT Olympus HP Engine ECU

As can be seen from Figure 5-2, the designed ECU in this study incorporates the following components:

- Two (0-6V) solenoid valves
- Fuel pump
- LCD
- Speed sensor
- Thermocouple
- Glow plug
- 12V starter engine

ECU basically takes input signal from speed sensor and thermocouple. Note that the shaft speed, throttle state, and exhaust gas temperature readings are also displayed on the LCD. These signals, which are fed back to the controller, are basically used to prevent high temperature inside the engine as well as high speed of the turbine shaft during the transient and steady-state regimes. Manual controller given in Figure 5-2 is utilized in the initial phase of the research to select the parameters of the start, stop and auto-stop mode. Due to the harsh conditions encountered in the combustion chamber, nozzle, and turbine; the developed ECU has to be able to stop the engine safely. For this purpose, auto-stop mode (allowing the engine to cool itself during stop process) is incorporated to the design.

In the following sections, the circuit design as well as the fuzzy logic control algorithm developed for this specific engine will be elaborated.

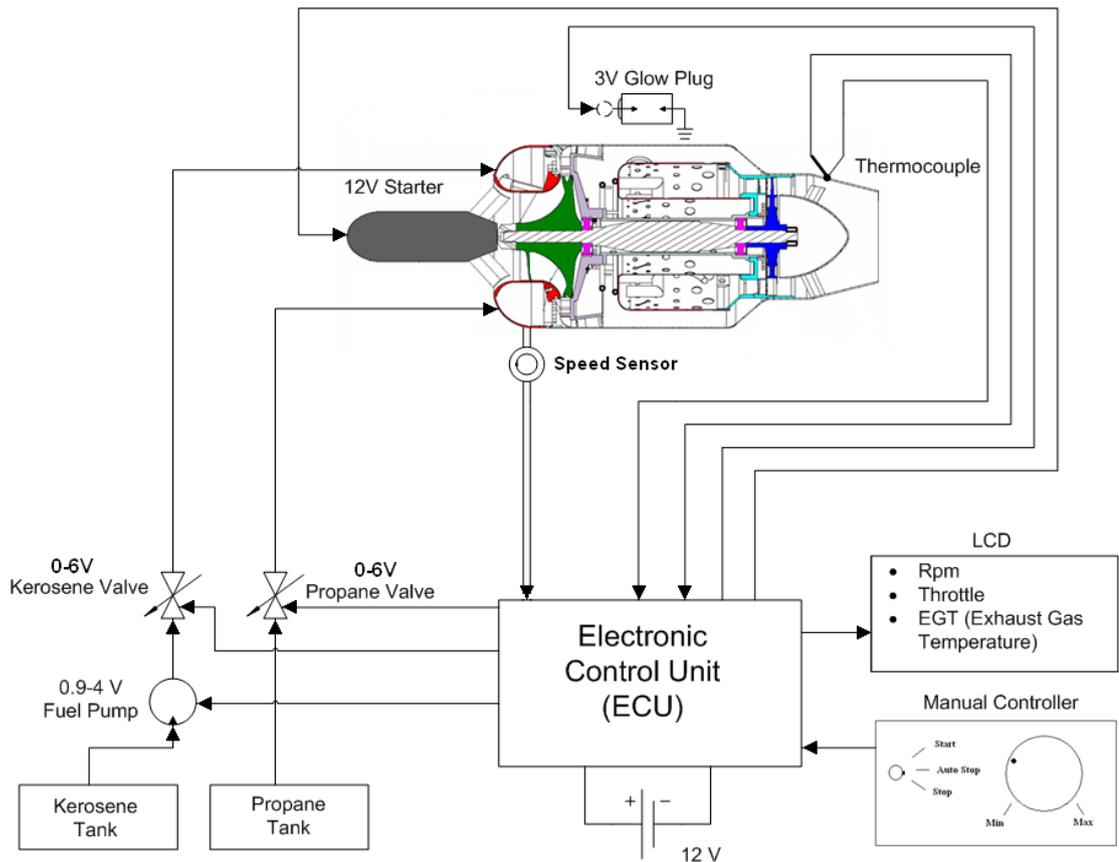


Figure 5-2 Schematic Diagram of the AMT Olympus HP

5.3 Circuit Design

AMT Olympus Hp turbojet engine controller was designed by Mikroelektronika EasydsPIC3 development tool, which includes a microcontroller, and a number of useful peripheral units like tactile buttons, LEDs, LCD, and RS232 level converter, etc. Hence once can implement and test a specific embedded control system directly using this tool. That is, microcontroller can be programmed with EasydsPIC3 using the integrated development environment. Key features of this board are as follows:

- External or USB power supply can be utilized,
- It has as USB programmer,
- RS232 serial communication is available,
- LCD and GLCD can be connected,

- 32 buttons allow the designer to control all inputs in the microcontroller,
- PC keyboard connector.

Using this development tool, an electronic control unit for AMT small gas turbine can be realized by dsPIC30f4013. This 16-bit digital signal controller (DCS) has the following specifications:

- Five 16-bit timers/counters,
- Four 16-bit capture inputs,
- 12-bit AD convertor with
- 48 Kbytes on-chip Flash program space,
- 2 Kbytes of on-chip data RAM,
- Programmable code protection,
- Selectable power management modes,
- Low power consumption.

5.3.1 Main Circuit

Main controller circuit can be seen in Figure 5-3. As can be seen, almost all inputs and outputs of the DSC running at 10 MHz are utilized in the controller design. Four analog inputs are dedicated to the thermocouple. On the other hand, four outputs are needed to drive L298. The LCD (16 chars by 2 lines) on the ECU displays shaft speed, temperature and thrust level for monitoring purposes.

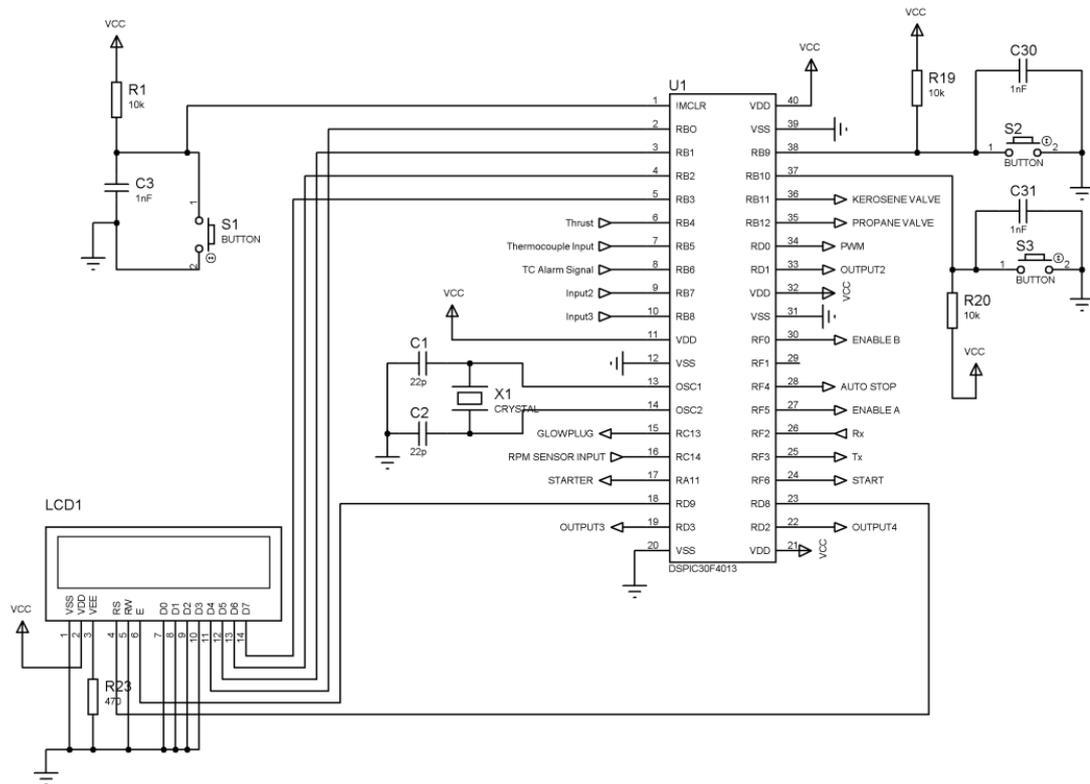


Figure 5-3 Main Circuit

5.3.2 Temperature Measurement

For temperature measurement, AD595 is employed to read temperature value from the temperature sensor as shown in Figure 5-4. Temperature signal voltage is too small for the microcontroller to convert efficiently. Hence, AD595 integral circuit amplifies this low voltage to a proper voltage level (0-5V). On the other hand, the AD595 produces an error signal if the temperature sensor is not connected to the system or it does not work properly [26].

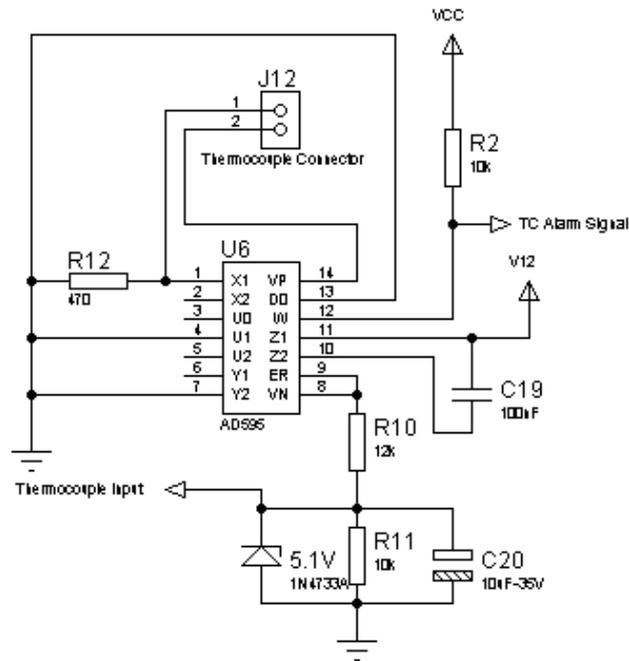


Figure 5-4 Temperature Circuit

Note that AD595 IC output changes in between -1454 and 12524 [mV] ranges when temperatures between -200° to 1250° C degrees are observed. Since the maximum input to the ADC of the DSC is $5V$, AD595 output voltage is attenuated by $5/11$ by using appropriate resistance combination [26].

5.3.3 Kerosene Pump and Glow Plug Driver

In this study, L298 IC (Dual H-bridge Driver) is used to drive the glow plug as well as the fuel pump as illustrated in Figure 5-5. Note that the glow plug needs $3.5 V$ to ignite the propane in the combustion chamber. Finally, the fuel pump requires $0.9 - 4 V$ of supply voltage to drive fuel to the chamber. Experiments show that fuel pump and glow plug need almost $4 A$ of current. Hence, BJTs are not suitable to drive these units. Consequently, L298, which can provide a maximum of $4 A$, is selected to accomplish these tasks in the preliminary design [27].

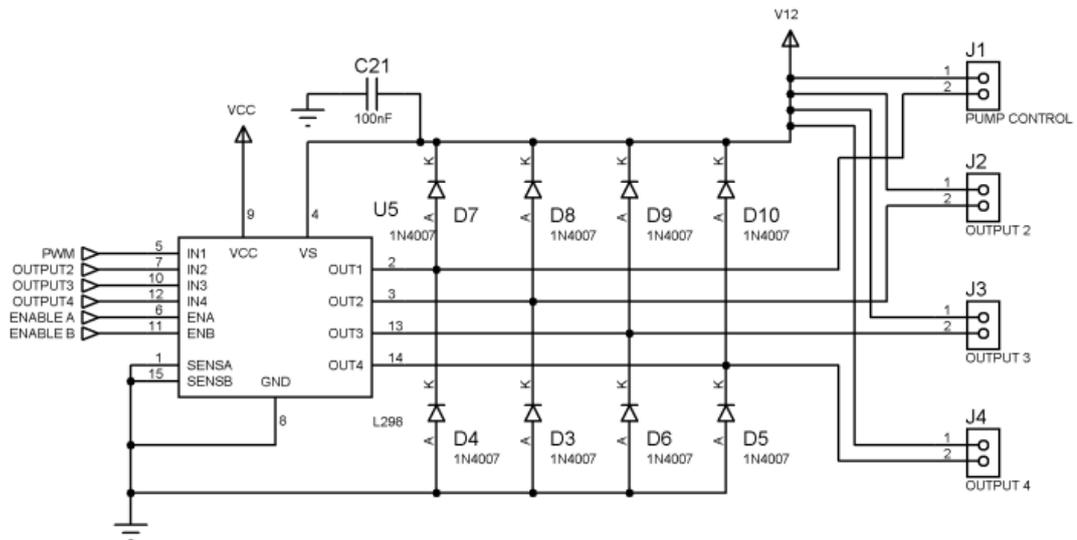


Figure 5-5 L298 Circuit

Note that it has been observed in the experiments that the glow plug actually draws approximately 6 A from the driver in the tests. Since the L298 can yield a maximum of 4 A, the resulting driver circuitry cannot heat the glow plug to burn the propane gas efficiently. To overcome this problem, a new design shown in Figure 5-6 is adopted. In this configuration, a battery pack (6V @ 3000 mAh) is connected to the system to energize the glow plug.

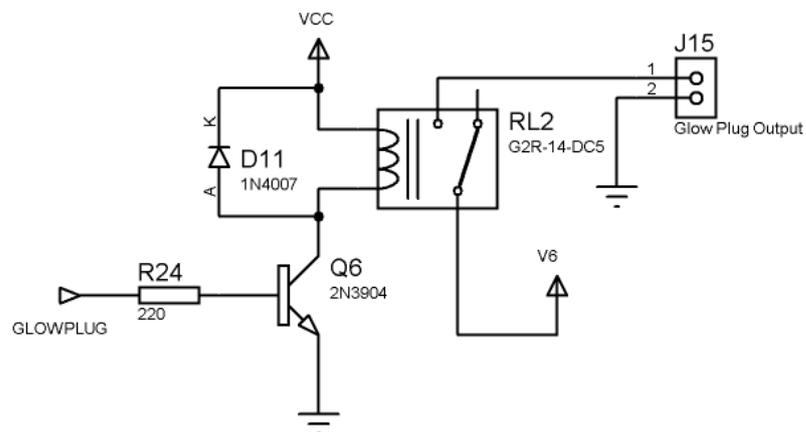


Figure 5-6 Revised Glow Plug Circuit Design

5.3.4 Kerosene and Propane Valve Drivers

Kerosene and propane valves are both normally closed valves. Hence, they need 6 V (at 0.5A) input to open. Fig. 5-7 shows the kerosene and propane valve (open collector) drivers. In this topology, 6-V linear voltage regulator (LM7806) is utilized to drop 12V voltage supply voltage to the desired level.

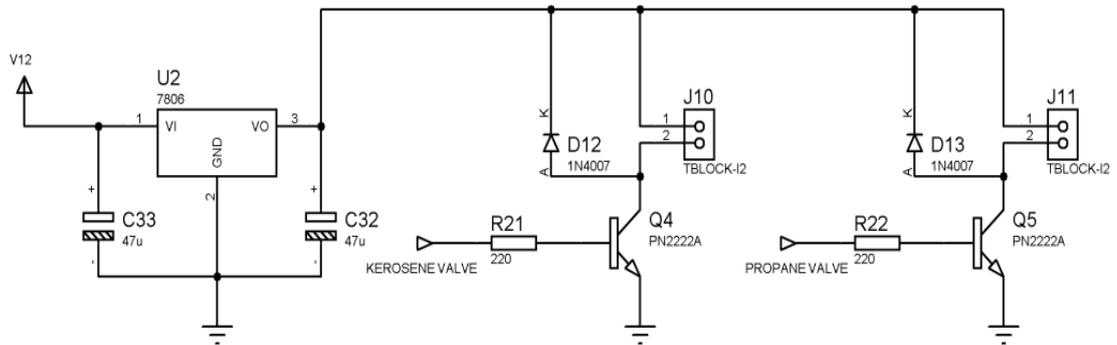


Figure 5-7 Kerosene and Propane Valve Circuit

5.3.5 Other Standard Elements

Figure 5-8 illustrates the standard RS232 communication circuit of the ECU controller. Despite the fact that in this study, the communication link is primarily utilized to debug the firmware, it may be used for more utilization purposes (i.e. data collection, human-machine-interfacing, machine state monitoring, real-time diagnostics, etc.) in the future work.

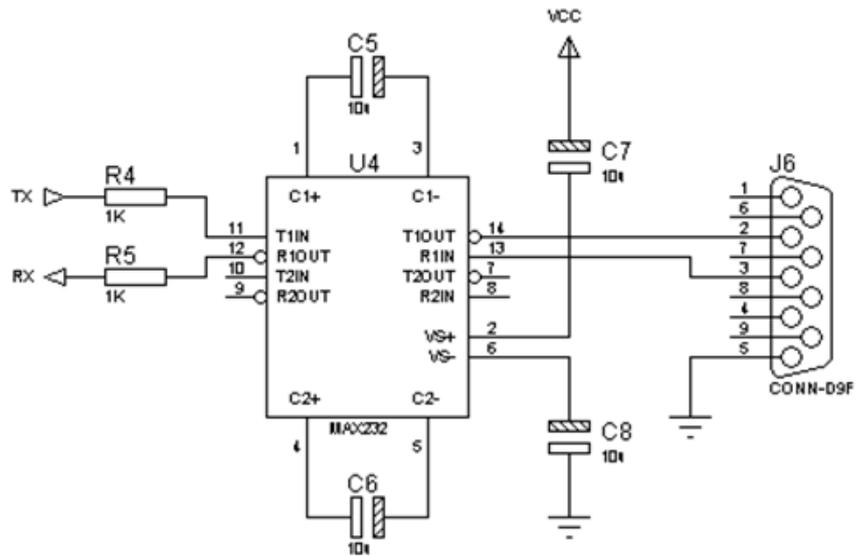


Figure 5-8 RS 232 Circuit

Consequently, Figure 5-9 shows shaft speed input circuit of the ECU. Speed sensor signal voltage may be 0-12 V. Hence, this high voltage level must be reduced to the 5 V.

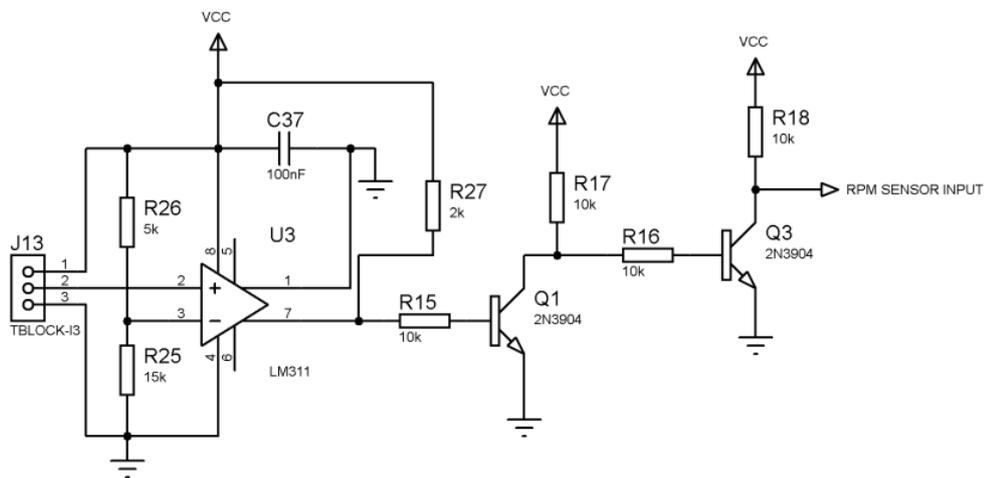


Figure 5-9 Shaft Speed Input Circuit

Shaft speed measuring circuit of the ECU is improved due to the behavior of inductive proximity sensor. First of all, an inductive sensor of the INFRA (Italy) was initially tested. It

was observed that the sensor emits a TTL pulse as the compressor blade passes in front of the sensor. Unfortunately, the sensor cannot cope with high frequency inputs and exhibits poor performance when shaft speed exceeds 9000 [rpm]. Hence, OMRON proximity sensor is utilized to measure shaft speed. This sensor outputs 4.6 V when it detects a compressor blade while yielding 2.9 V elsewhere. This voltage difference is used to shape square waveform for the microcontroller. For this purpose, a voltage comparator circuit is designed as illustrated in Figure 5-9.

Similarly, Figure 5-10 shows thrust input and other two inputs for ECU. In addition to the thrust input, two more input may connect to the ECU by using the following circuit.

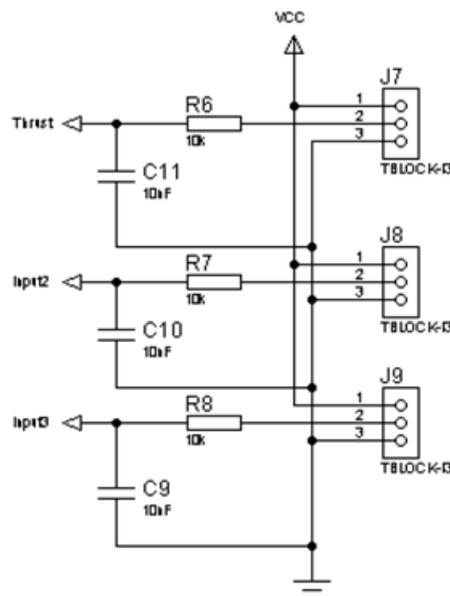


Figure 5-10 Analog Input Circuit

Figure 5-11 shows electric starter's driver circuit. Electric starter motor (ESM) is driven by a 12 V supply and could reach to 7500 [rpm]. Note that the ESM is employed as a part of a start procedure of the ECU control program. Furthermore, it cools down the engine nozzle and turbine in the auto stop procedure.

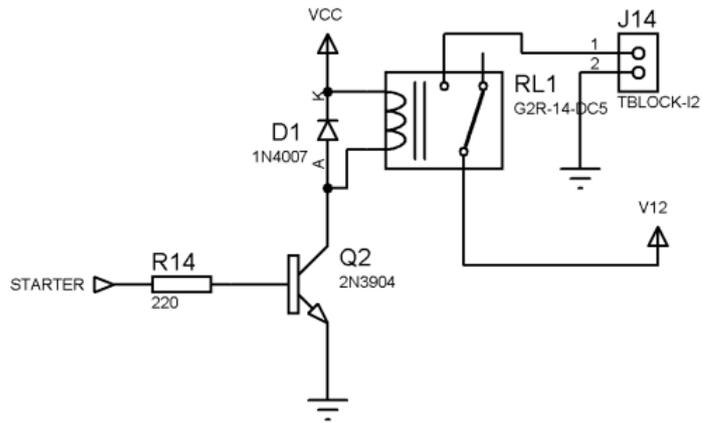


Figure 5-11 Starter Circuit

Using these sub circuits, main circuit board of AMT Olympus Hp turbojet engine is designed and the overall 3D view of the card is given in Figure 5-12.

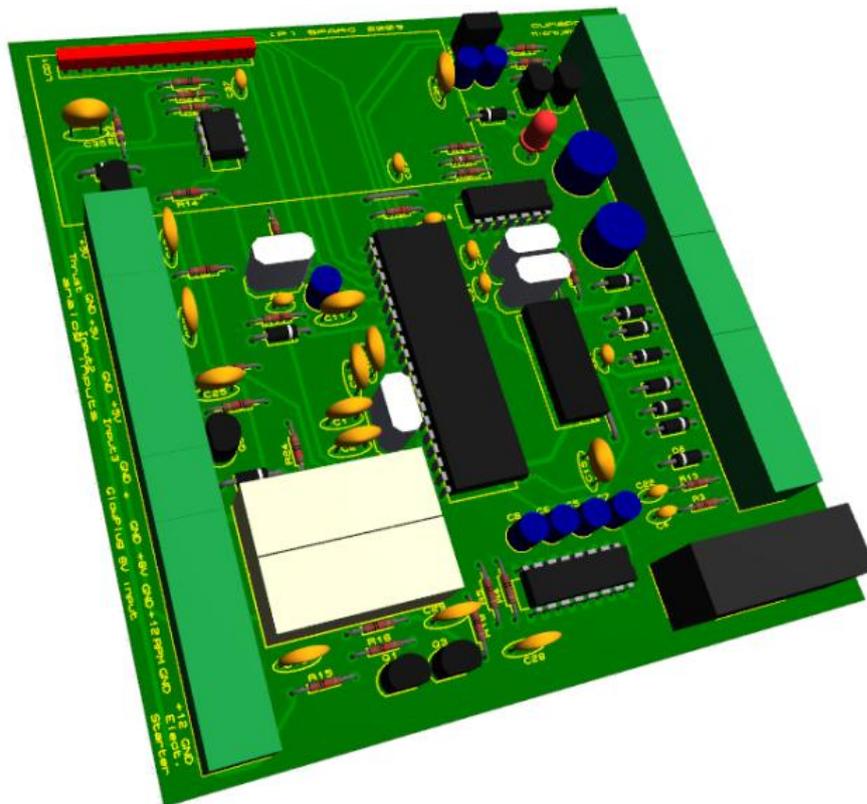


Figure 5-12 3D View of the Controller (Beta Version)

Assembled controller card (alpha version) is given in Figure 5-13. This circuit can control a small gas turbine with one temperature sensor, one speed sensor input. The controller can manipulate two ON/OFF fuel valves, one electric starter, one glow plug, and one fuel pump. Besides, it can send and receive data from RS232 ports. Furthermore, this circuit can show shaft speed, throttle and temperature values through its LCD. Fuzzy logic controller was embedded into the dsPIC30f4013 by using easydsPIC3 development board. Finally, Figure 5-14 shows artwork of the printed circuit board (top and bottom copper layer) of the ECU.



Figure 5-13 ECU Circuit

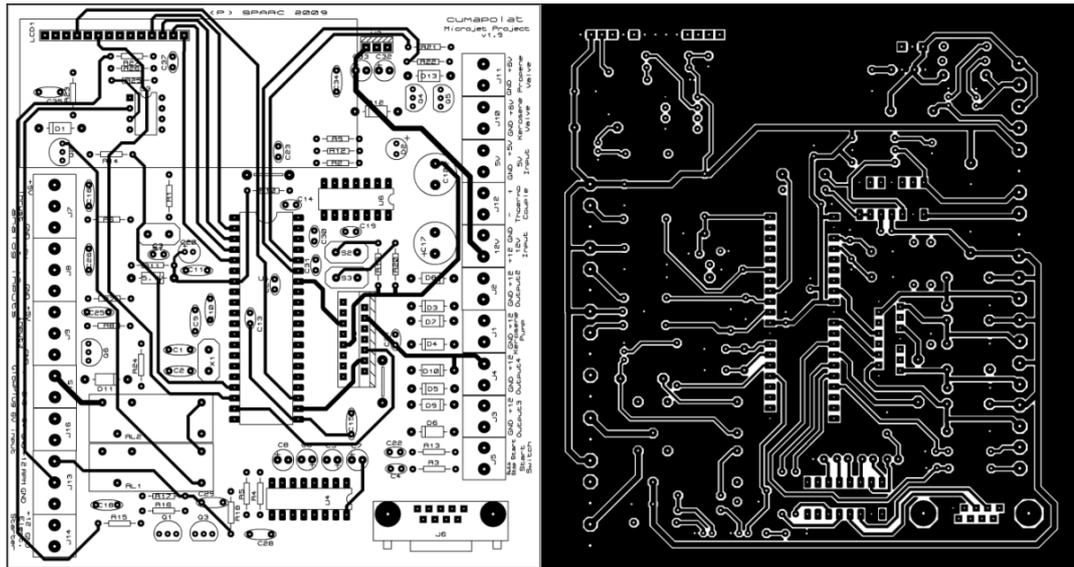


Figure 5-14 Top-, Bottom, and the Silk Layer of the Circuit

5.4 Closure

In this section of the thesis, the attributes of the ECU specifically designed for small gas turbine engine were elaborated. Inputs and outputs of this ECU were discussed. The functions of the main circuit and its sub circuits including the development tool used for programming were described. Finally, the revisions on the circuit were elaborated to aid the on-going (and future) studies. Next section will focus on the performance of the designed system.

CHAPTER 6

ECU TESTS AND RESULTS

6.1 Introduction

The design and implementation of the ECU for a small turbojet engine was discussed in the previous chapter. In this chapter, all functions of this ECU are to be tested using the engine test stand at hand and the results are to be documented.

6.2 Test on ECU Sub-Components

Figure 6-1 shows the engine test stand used in this study. As can be seen, the stand includes the original jet engine housed on a load-cell, the fuel (propane, kerosene) sources, peripheral units (pumps, solenoid valves), the ECU, and an adjustable DC power supply. First test is conducted on the various subcomponents of the ECU. Most important problem encountered during the tests is the high current demand of the glow plug. At the initial stage of the research, the glow plug was tested and observed that it can be heated via a DC power source supplying 3.5 V at 3-4 A current, which was located 30 cm away from the plug. On the other hand, the distance between ECU and gas turbine engine was taken 1.5 m in the actual tests due to safety reasons. First test shows that 3.5 V was not enough to heat glow plug at this relatively long distance. Hence, to heat the glow plug circuit 6 V power source supplying 6-7A was needed. Hence, a suitable battery pack was prepared to provide this power.

A second major drawback was associated with the shaft speed measurement circuit. The existent proximity sensors were investigated and their signal input and output were determined by using oscilloscope. Type of the proximity sensor, which was utilized in AMT

Olympus HP turbojet engine, asked to the manufacturer of the engine and it was noticed that they selected inductive proximity sensor of OMRON. On the other hand, similar inductive proximity sensor of INFRA Italy was tested and experimental results showed that this proximity sensor did not work after 9000 [rpm]. Output of the OMRON proximity sensor changes between 2.9 - 4.6 V. Hence, a comparator circuit was designed and embedded into the main ECU circuit. Another important problem during designing ECU circuit was voltage drop in the 12 V power supply. Due to the high current necessities of starter, glow plug and kerosene pump, voltage drop occurs on the circuit. Hence, kerosene pump voltage changed while other components were working and it caused undesired input to the system. To prevent this problem, glow plug power supply was separated from the main circuit.

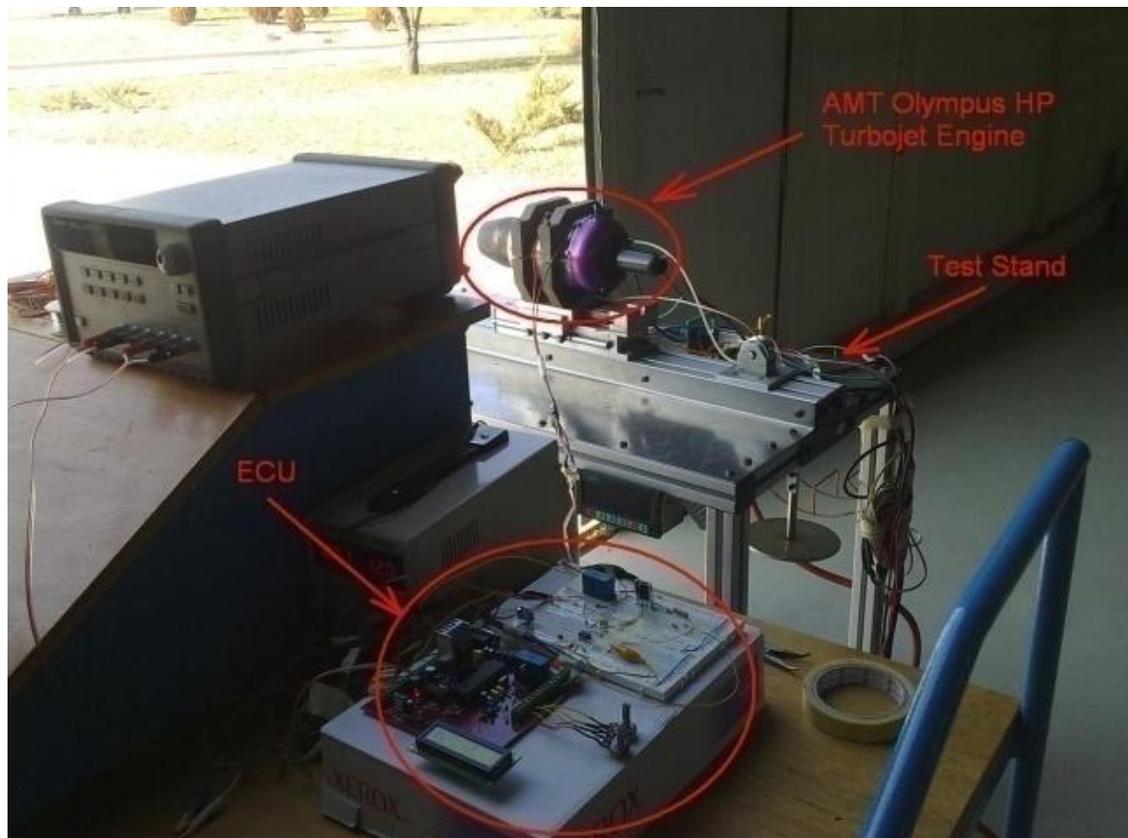


Figure 6-1 AMT Olympus HP Test Stand and Designed ECU

6.3 Firmware Tests

Required hardware revisions on the circuit were performed and discussed in the previous article. This section concentrates on the performance of the overall system with an emphasis on the firmware being developed.

First of all, an emergency stop program was designed and tested. According to this program, microcontroller checks the nozzle temperature to see whether it is greater than 70°C. If this condition is met, the microcontroller uses the electric starter to decrease the nozzle under 70°C degrees.

After automatic cooling, the ignition of propane gas proceeds. In the first design, when the shaft speed increases to 7000 [rpm], the propane valve opens and the glow plug is heated up respectively. Unfortunately, the developed program was not able to ignite the propane gas since the glow plug did not heat up as effective as expected. To overcome this difficulty, the supply voltage of the glow plug was increased. However, this did not solve the ignition problem entirely. During the tests, it was observed that the glow plug was rapidly cooling down due to the intake of the ice-cold air in the hangar. To solve this problem, an air conditioner was utilized to supply hot air into the gas turbine. Note that extremely hot air might create haphazard conditions for compressor (i.e. blade deformations). Thus, the heated air (to an allowable degree) did not help much to ignite propane gas. Finally, the ignition problem was solved by decreasing shaft speed from 7000 [rpm] to 3000 [rpm]. Nevertheless, providing propane gas into the combustion chamber before heating glow plug caused a small blow out problem from the nozzle. Hence, the glow plug was heated first and then propane gas was provided into the combustion chamber. After obtaining proper ignition, electric starter increased the shaft speed from 3000 [rpm] to 10 [krpm] and it awaits 15s before starting automatic cooling program. Automatic engine cooling and propane gas ignition algorithm is given in Figure 6-2. This control algorithm given in Figure 6-2 was tested several times and was observed that the code worked properly. After this control algorithm applied to the ECU and tested effectively; an open loop control strategies was investigated. It was noticed that the shaft speed of the gas turbine could exceed 15 [krpm] when a suitable voltage to the kerosene pump through the trust input pot is applied.

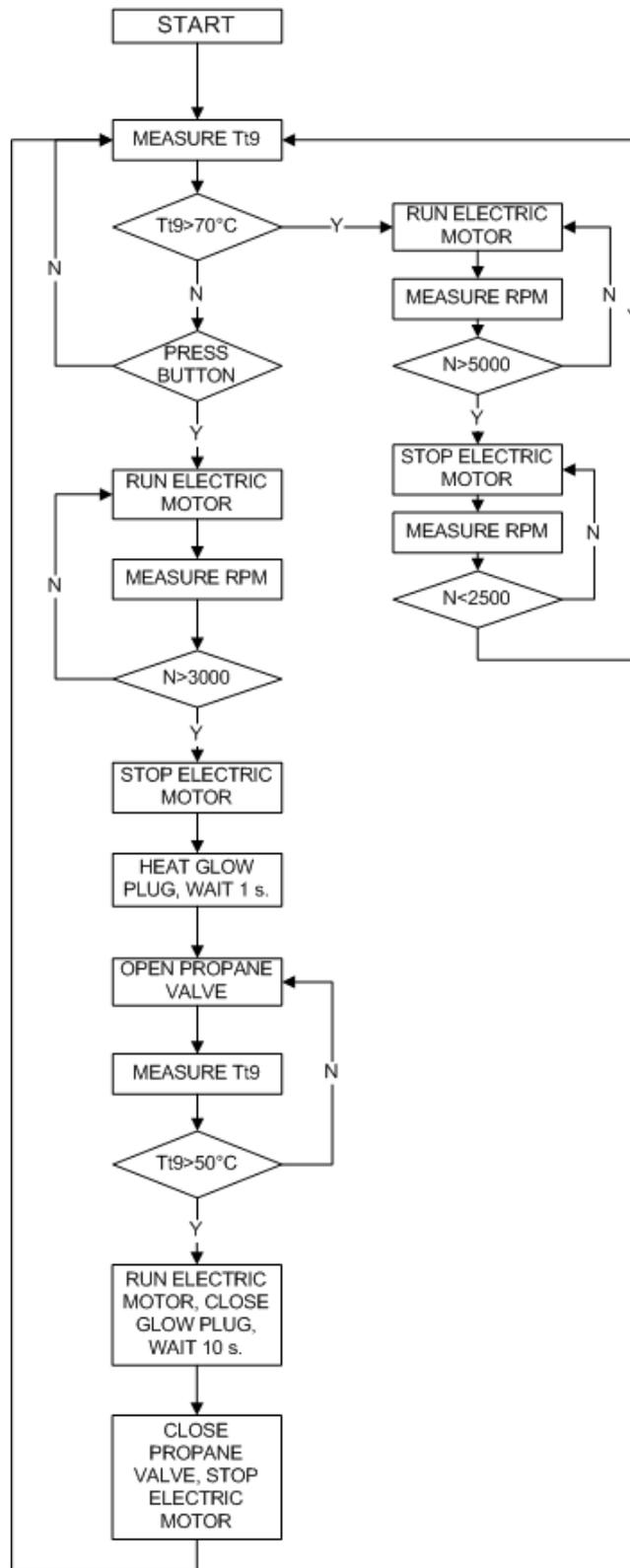


Figure 6-2 Automatic Engine Cooling and Propane Gas Ignition Algorithm

6.4 ECU Test Results

After designing proper engine starting algorithm, the ECU is connected to the AMT Olympus Hp turbojet engine and several tests are conducted to find suitable operating point of kerosene fuel. Seven tests are conducted for all intensive purposes.

In the first test given in Figure 6-3, there are three start-up attempts within the first 50 seconds. In this section of the test, the temperature of the nozzle does not increase due to an ignition problem. After this point, the glow plug ignites the propane and thus the temperature of the nozzle increases to 520 °C. The controller waits 10 s to initiate cooling process. As can be seen, the controller increases the shaft speed from 2500 to 5000 [rpm] (see zigzagging portion) and regulates the nozzle temperature. When the nozzle temperature goes down to 70 °C, the ECU terminates the starting cycle and awaits the next starting command.

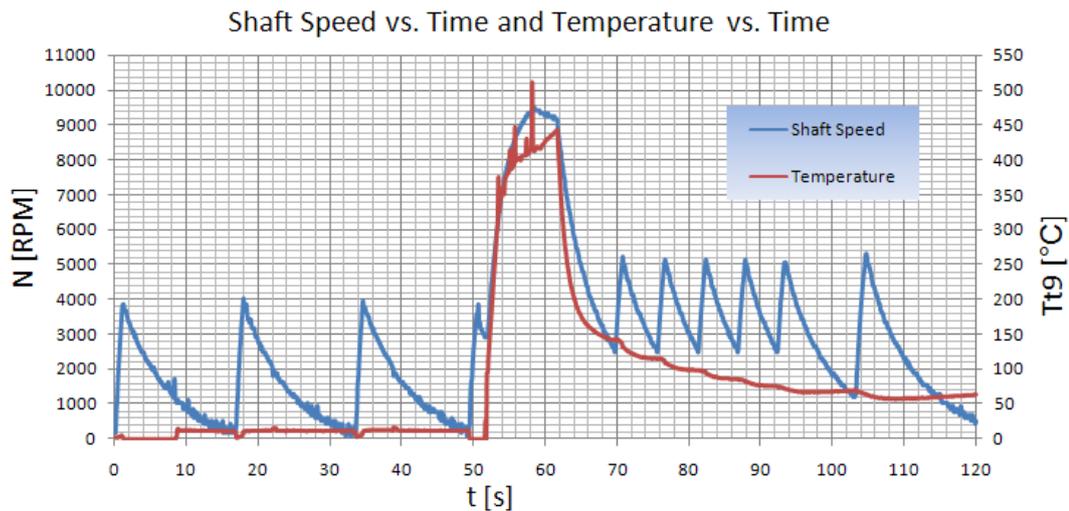


Figure 6-3 ECU Test with AMT Olympus Hp Turbojet Engine (Test 1)

In the second test shown in Figure 6-4, the ECU fails to start the engine in the first two trials. As can be seen, there are two peak points of the nozzle temperature that reaches almost 450°C. Since the ECU cannot start the engine, the shaft speed remains under 3000 [rpm]. Third portion of the result illustrates that the ECU runs the electric motor for 88 second and its speed reaches to 10 [krpm] within 6 seconds. Afterwards, the ECU waits for 15 seconds

(5 seconds longer than the first test). As can be seen from Figure 6-4, the temperature goes up to 550 °C at the end of this section due to low mass flow rate of the compressed air.

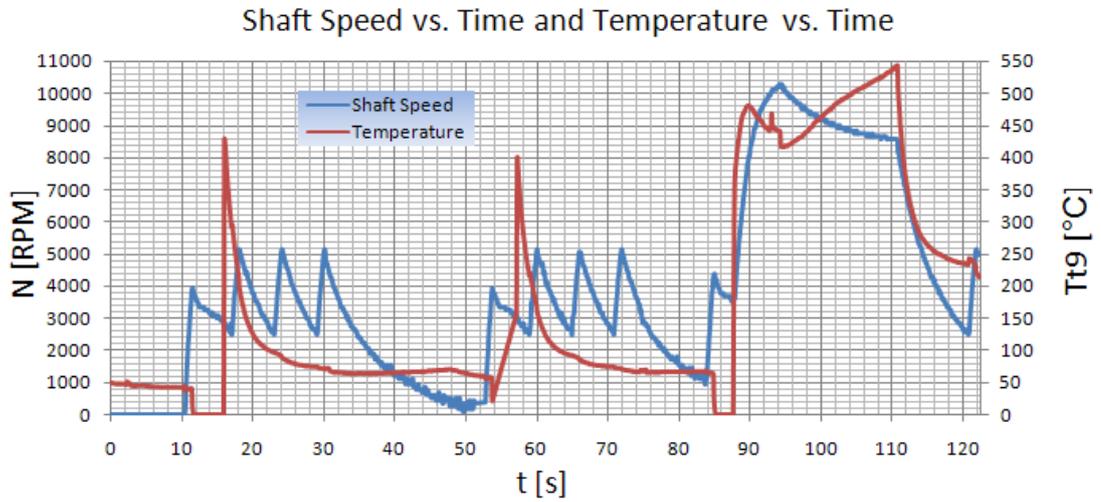


Figure 6-4 ECU Test with AMT Olympus Hp Turbojet Engine (Test 2)

Consequently, Figure 6-5 to Figure 6-8 show the other unsuccessful start-up attempts. In these tests, the ECU cannot start the engine because of the glow plug current values. New battery pack, which gives 7.2 V @ ~7 A current, is used to solve this problem but a maximum current of 7 A frequently damages the glow plug and cannot ignite the propane thus the temperature cannot increase. For the next and the last experiment, 6 V battery package and new glow plugs are utilized.

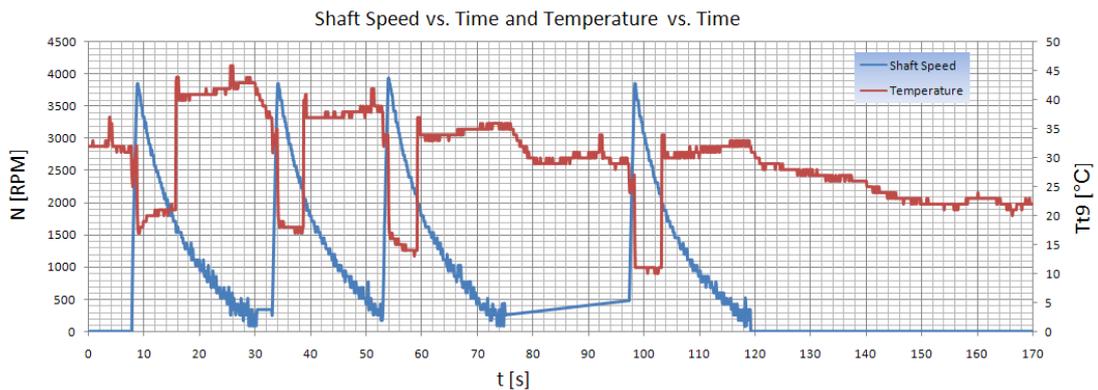


Figure 6-5 ECU Test with AMT Olympus Hp Turbojet Engine (Test 3)

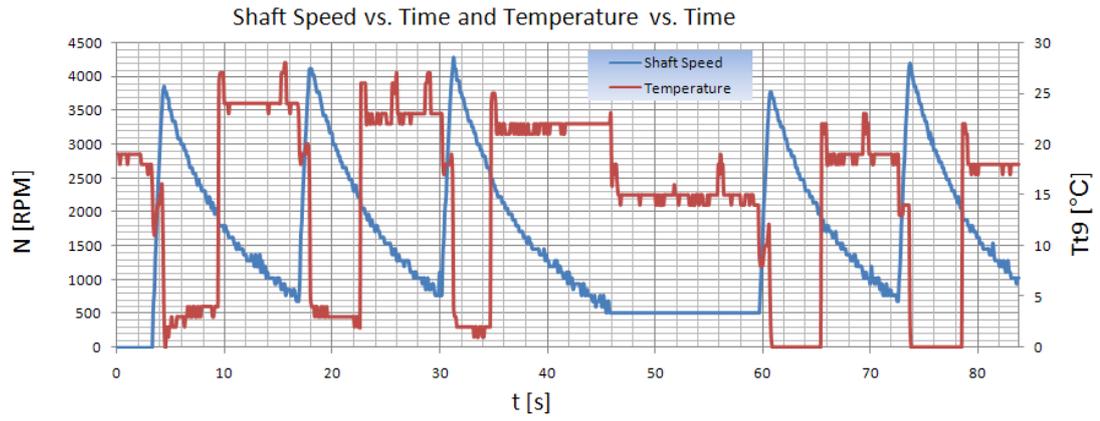


Figure 6-6 ECU Test with AMT Olympus Hp Turbojet Engine (Test 4)

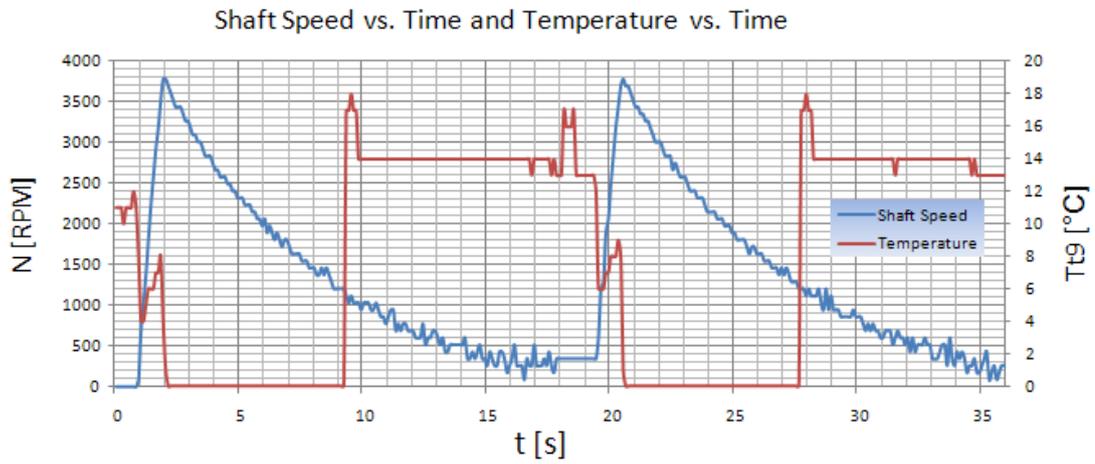


Figure 6-7 ECU Test with AMT Olympus Hp Turbojet Engine (Test 5)

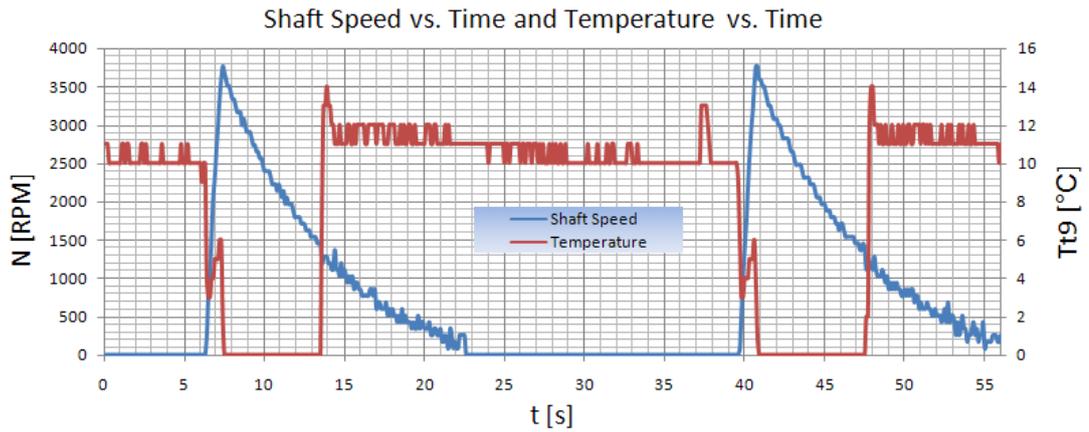


Figure 6-8 ECU Test with AMT Olympus Hp Turbojet Engine (Test 6)

Figure 6-9 summarizes the last attempt. As can be seen, the test starts at the 10th second but the ECU could not read the temperature value. In second trial, the ECU starts the engine up and the shaft speed rises to 10 [krpm] while temperature reaches to 600°C. At this point in the test, the kerosene is applied to the engine. Unfortunately, the temperature reaches to very high values (920°C) while the shaft speed may be still regarded as low (13 [krpm]). Note that a temperature of 920 °C is quite close to the meltdown temperature of the turbine material. Hence, the fuel flow is cut off and the ECU cools the engine down through the use of auxiliary engine startup motor to reduce the nozzle temperature to 70°C. The last experiment shows that applying the right amount of kerosene fuel is very crucial in controlling the engine.

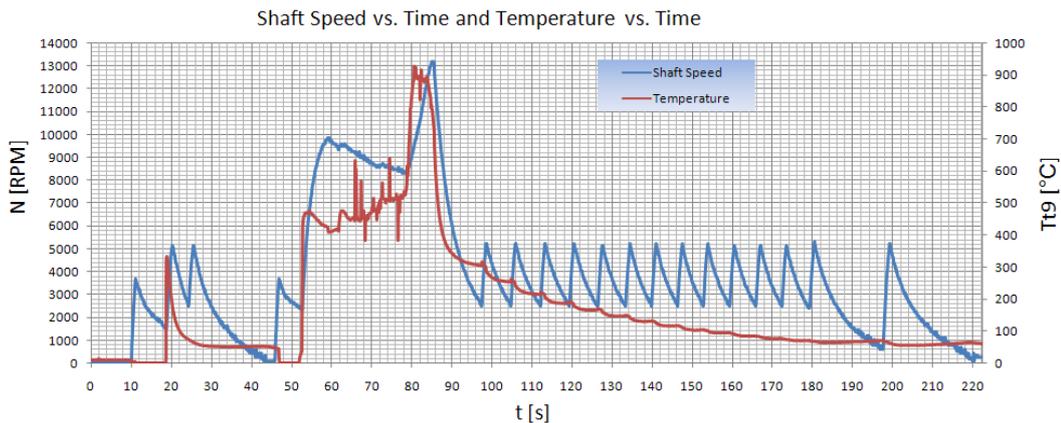


Figure 6-9 ECU Test with AMT Olympus Hp Turbojet Engine (Test 7)

The tests show that the developed ECU is capable of controlling the turbojet engine. However, a reliable map among important engine parameters (fuel rate, engine thrust, and shaft speed) of the engine is needed to devise an effective control algorithm. Hence, to gain a better understanding about the engine's behavior, a number of tests are conducted using the original ECU of the engine. The details of these tests follow.

6.5 Original ECU Test Results

In this part of the thesis, the experiment results of the original ECU of AMT Olympus HP turbojet engine are presented. Original controller of the AMT Olympus HP is given in Figure 5-1. This ECU not only can control the engine but also it can measure and store the kerosene pump voltage, throttle, EGT (T_{t9}), and shaft speed of the engine. These stored data can be collected by using V2 ECU Toolsuite V1.6.1 that is the original software of the AMT turbojet engine. Five different test results are obtained from ECU Toolsuite [32].

Figure 6-10 shows that throughout first 40 seconds, ECU controls the shaft speed by its own manipulation values. As can be seen from the throttle graph, throttle value is zero during the first 40 seconds while pump voltage, shaft speed and EGT change with respect to time. This behavior of the ECU shows that it starts the engine automatically in this period. On the other hand, the angular acceleration of the AMT Olympus HP turbojet engine can be found nearly from the shaft speed graph. Controller increases shaft speed from zero to 50 [krpm] in 25 second. Thus, acceleration can be calculated as

$$50 \text{ [krpm]} = 5235.987 \text{ [rad/sec]}$$

Average angular acceleration of the engine becomes

$$\alpha = 5235.987 / 25 \approx 209.439 \text{ [rad/sec}^2\text{]}$$

After waiting 4-5 seconds at 50 [krpm], the controller decreases the shaft speed from 50 [krpm] to 36 [krpm], which is the idle velocity of the engine. It waits almost 15 second at 36 [krpm] and then it gives the control to the operator. In this experiment, the shaft speed of the engine is increased to 100 [krpm], which is almost the highest shaft speed of the engine. EGT graph shows that the nozzle temperature almost remains constant (400°C) throughout the steady-state regime. Thus, it is clear that the compressor can cool the turbine and nozzle if the suitable acceleration is selected.

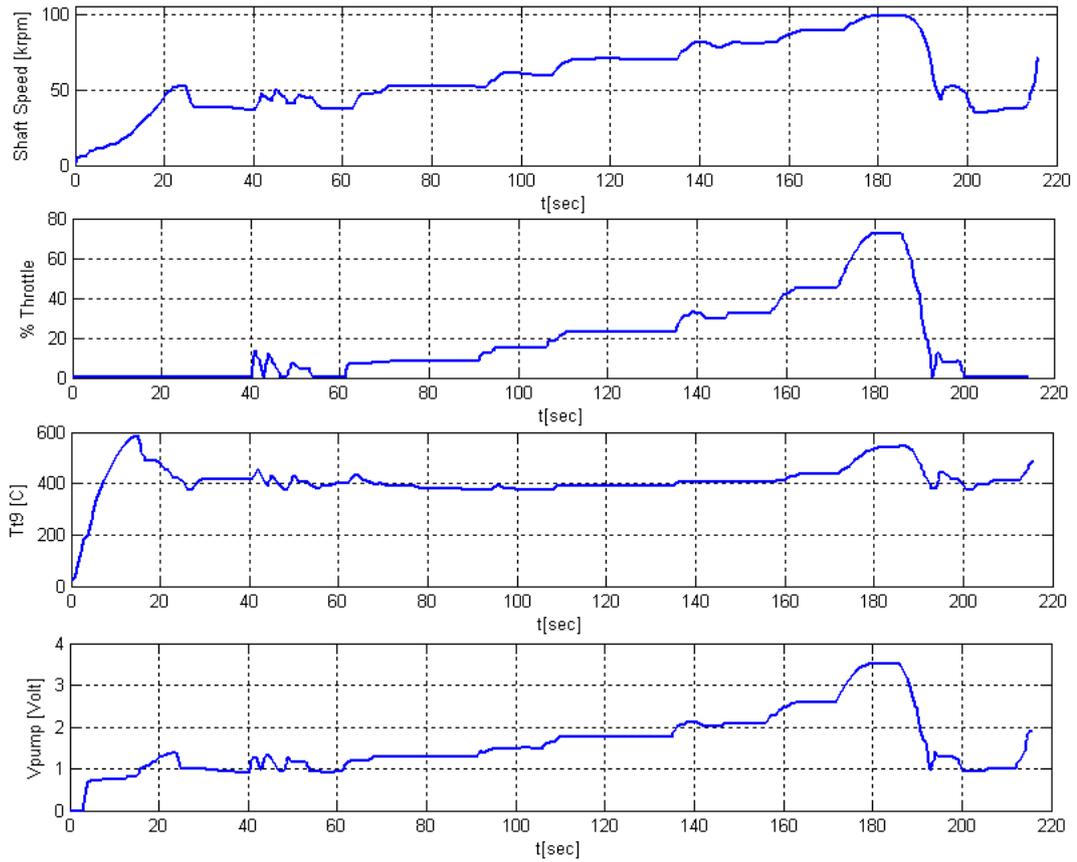


Figure 6-10 Original ECU Test Results (Test 1)

Likewise, Figure 6-11 shows that the ECU increases the shaft speed from zero to 50 [krpm] in 25 second such as given in Figure 6-10. This illustrates that the ECU applies the same angular acceleration to the system in the automatic start procedures. Maximum nozzle temperature reaches to 680°C in the automatic start interval. Furthermore, EGT almost remains constant at 420°C in the steady-state regime. Finally, the operator adjusts the switch to the automatic stop position. Then, the ECU increases shaft speed to 80 [krpm] and closes the kerosene pump.

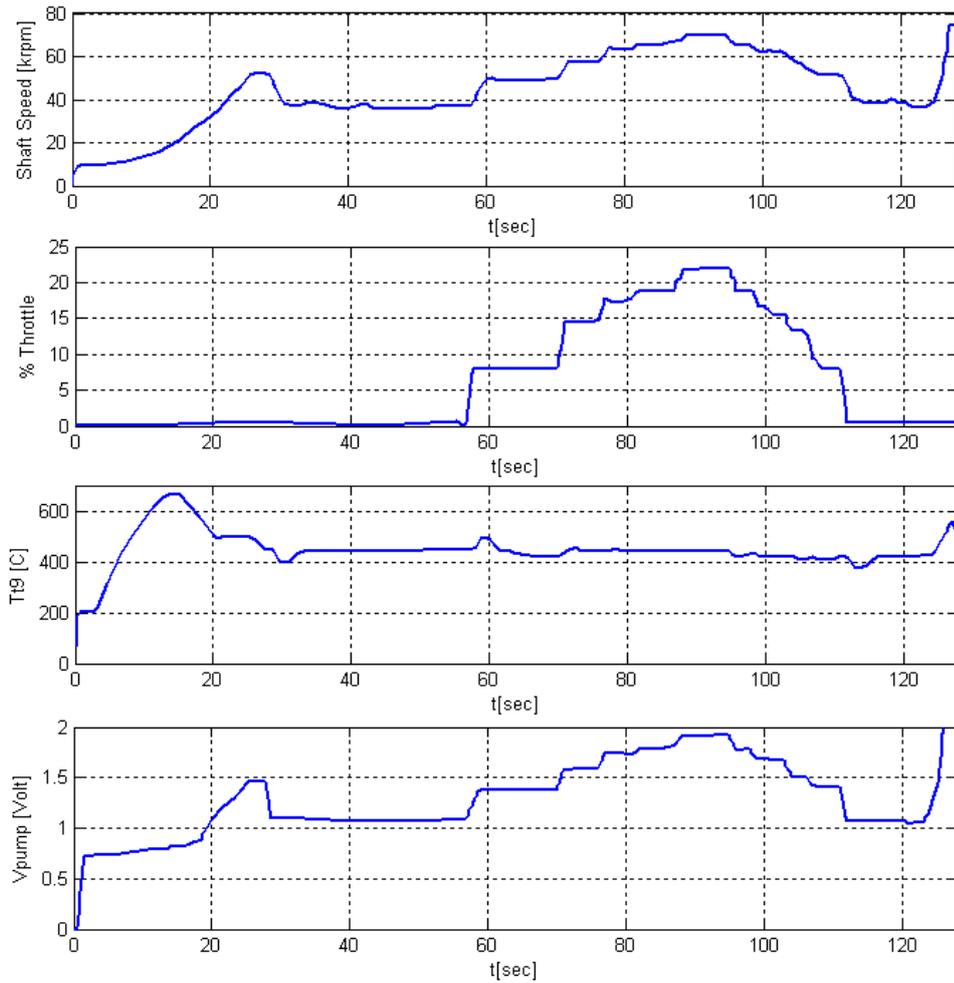


Figure 6-11 Original ECU Test Results (Test 2)

In the test shown in Figure 6-12, the ECU initiates the automatic start algorithm in the first 40 seconds just like the first two experiments. In this experiment, the shaft speed of the AMT Olympus reaches to 91 [krpm] and nozzle temperature arrives at 660°C. Note that the nozzle temperature remains constant at 420°C in the steady-state operating regime.

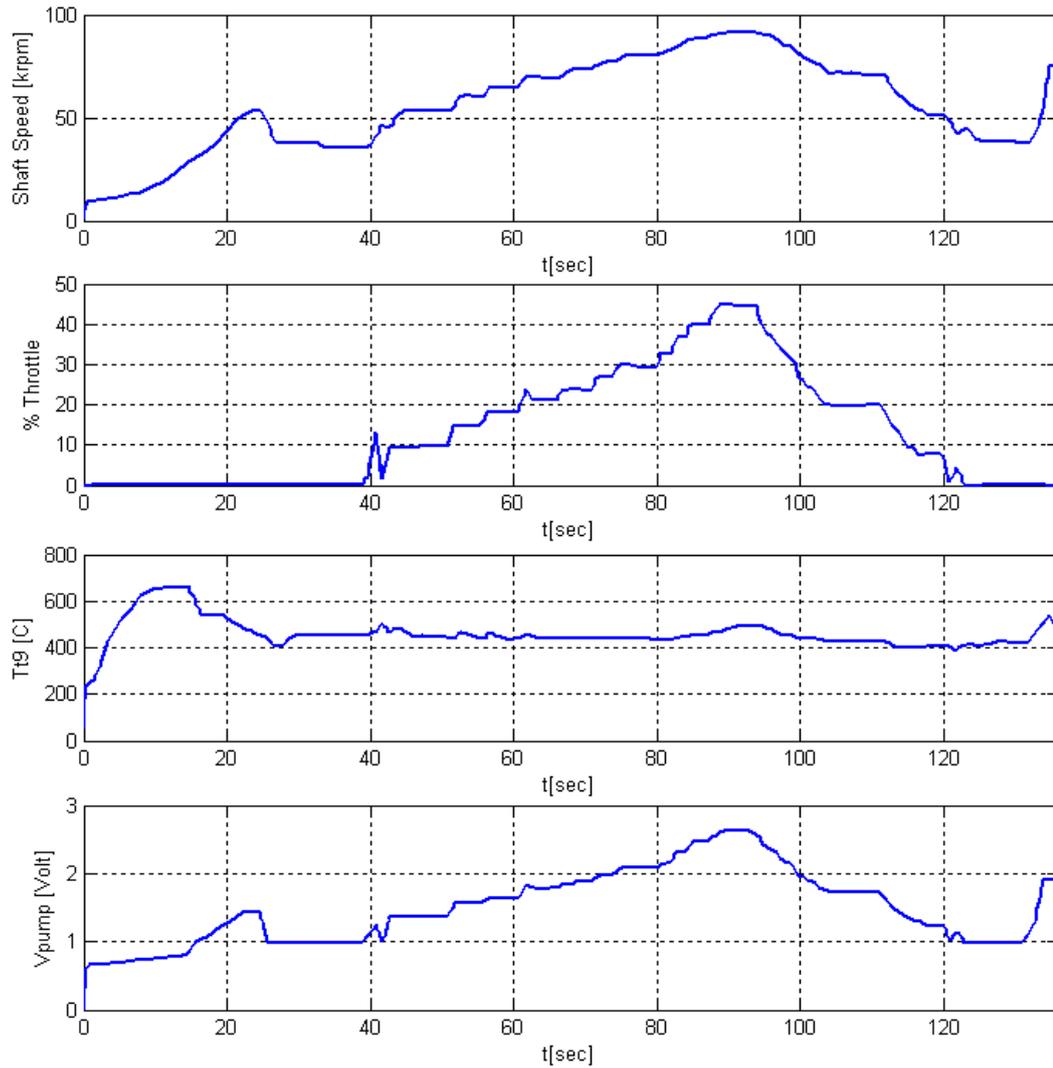


Figure 6-12 Original ECU Test Results (Test 3)

Figure 6-13 shows that the engine reaches 105 [krpm], which is almost the highest shaft speed of the engine. On the other hand, the nozzle temperature does not exceed 800°C for both steady state and transient operating conditions. It is obvious that the nozzle temperature does not remain constant after 90 [krpm]. Since the temperature values become critical after 90 [krpm], the controller has to manipulate the system to cool the nozzle and turbine after 90 [krpm].

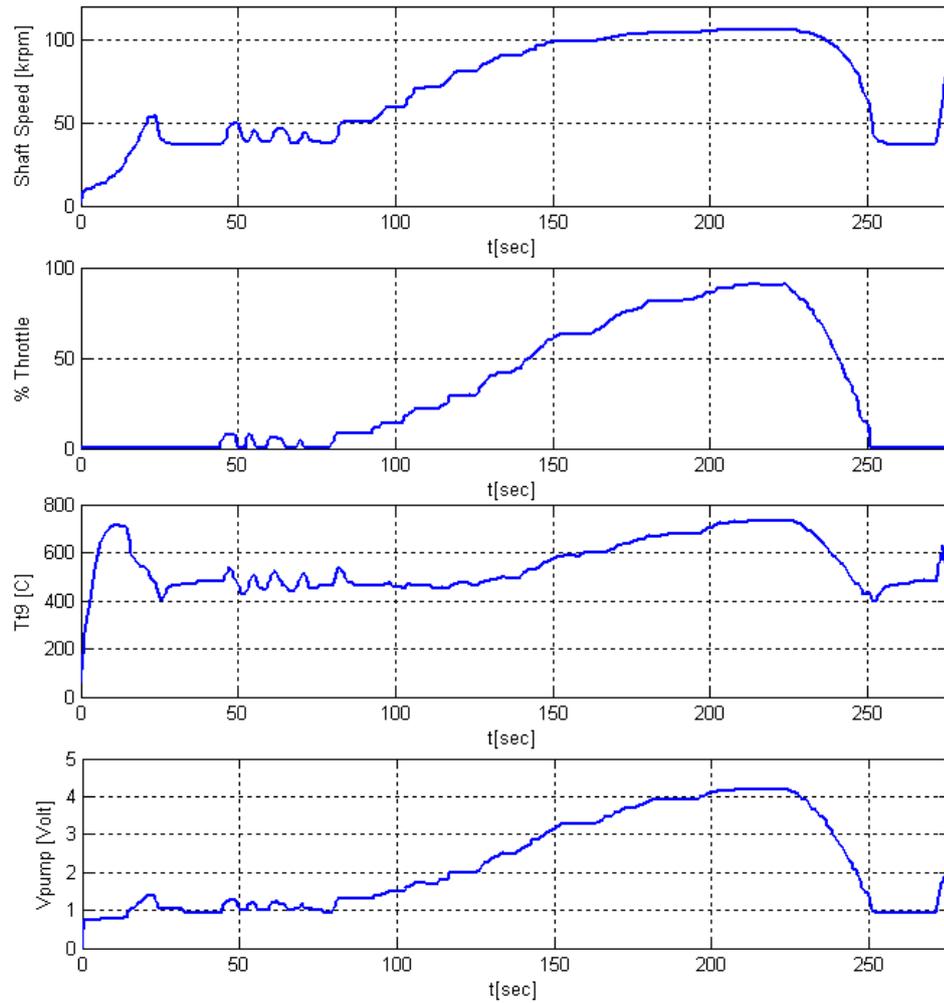


Figure 6-13 Original ECU Test Results (Test 4)

Last experiment presented in Fig. 6-14 shows that the shaft speed of the AMT Olympus HP reaches 60 [krpm] for 15% throttle input. Nozzle temperature of the engine remains constant after 25 second. Furthermore, ECU applies the same acceleration value given in Eq. (6.2).

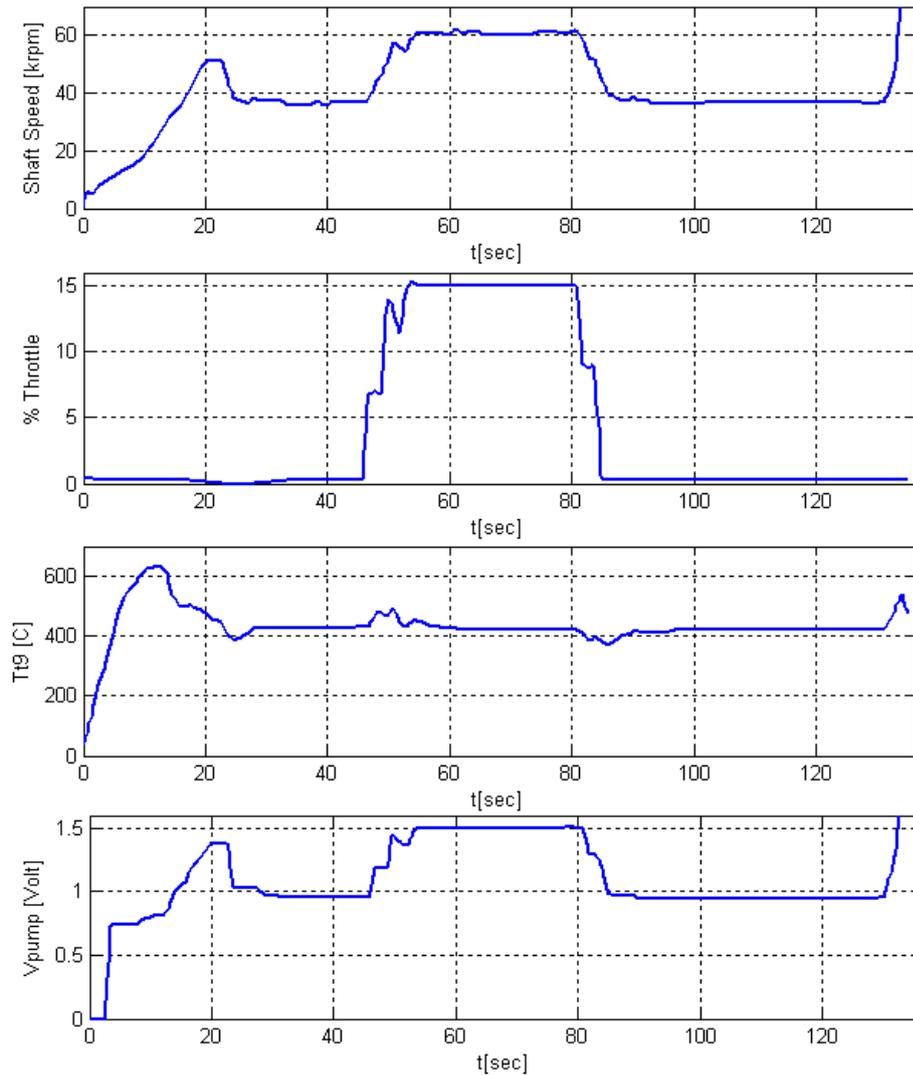


Figure 6-14 Original ECU Test Results (Test 5)

As can be seen from Figure 6-10, Figure 6-11, Figure 6-12, Figure 6-13 and Figure 6-14, for the first 40 seconds, the original ECU of AMT Olympus HP turbojet engine steers the system to an automatic start operation. Detailed pump voltage and shaft speed graph is given for the automatic start period of the engine in Figure 6-15. Figure 6-15 illustrates that the ECU uses two different accelerations for the pump voltage increment in the automatic start interval. Thus, the shaft speed increases more slowly between 0 and 18 seconds than between 18 and 25. It waits 5 second at 52 [krpm] and then it returns to the idle shaft speed of the engine. Finally, the ECU stops the automatic manipulation operation and transfers the control to the operator.

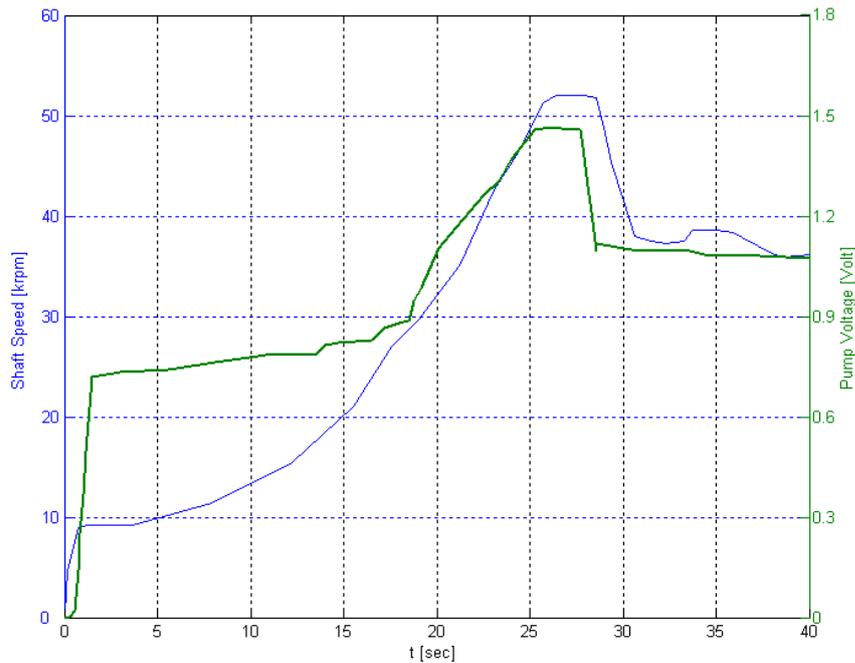


Figure 6-15 Pump Voltage and Shaft Speed Graphs for Automatic Start Operation (Test 2)

Figure 6-16 and Figure 6-17 demonstrate that there exist a quadratic relationship between the shaft speed and the pump voltage. It is clear that the shaft speed changes from 10 [krpm] to 80 [krpm] while pump voltage changes from 0.7 V to 2 V. It follows that the shaft speed increases 70 [krpm] for 1.3 V pump voltage increment. On the other hand, the shaft speed increases to 105 [krpm] from 80 [krpm] while pump voltage changes from 2 V to 4.2 V. Pump voltage increases 2.2 V for 25 [krpm] shaft speed increment.

These relations play a key role in understanding engine behavior for different pump voltage inputs. For instance, Figure 6-16 and Figure 6-17 shows that the pump voltage increment for different shaft speed are not similar. Thus, the ECU has to regulate the system in a different way for special conditions.

Pump voltage vs. EGT graphs are given in Figure 6-18 and Figure 6-19. It is obvious that the relations between the pump voltage and nozzle temperature are not parabolic such as shaft speed vs. pump voltage graphs. However, both graphs are similar to each other which show that the temperature of the nozzle fluctuates around the neighborhood of 400° – 500°C.

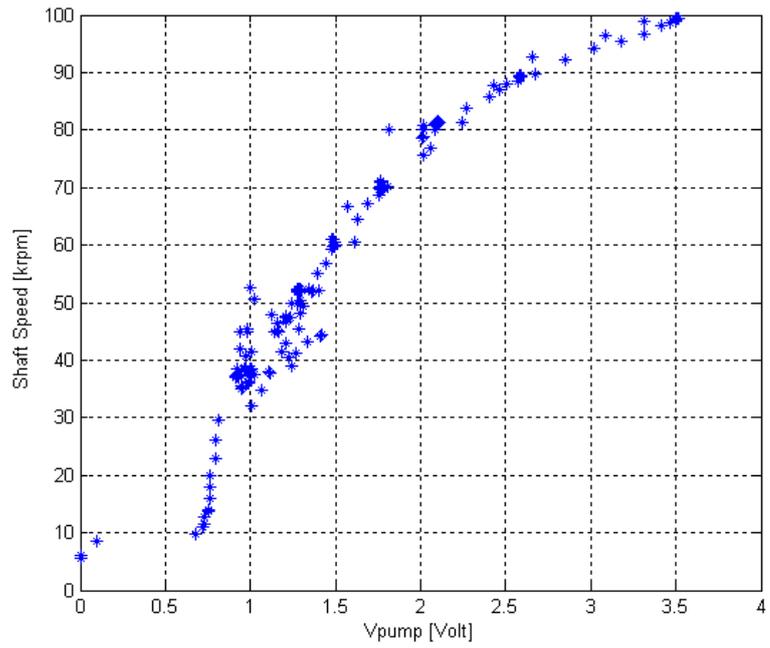


Figure 6-16 Shaft Speed vs. Pump Voltage for Test 1

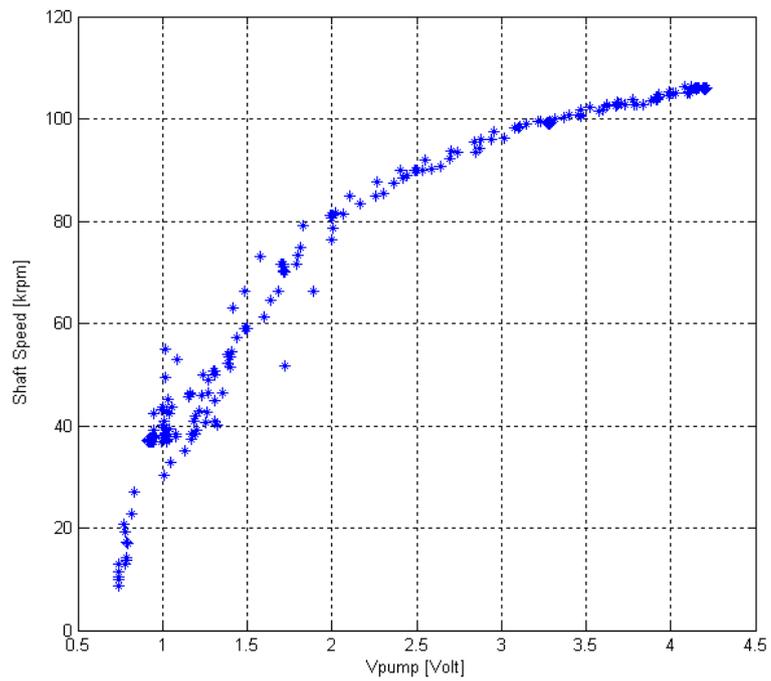


Figure 6-17 Shaft Speed vs. Pump Voltage for Test 4

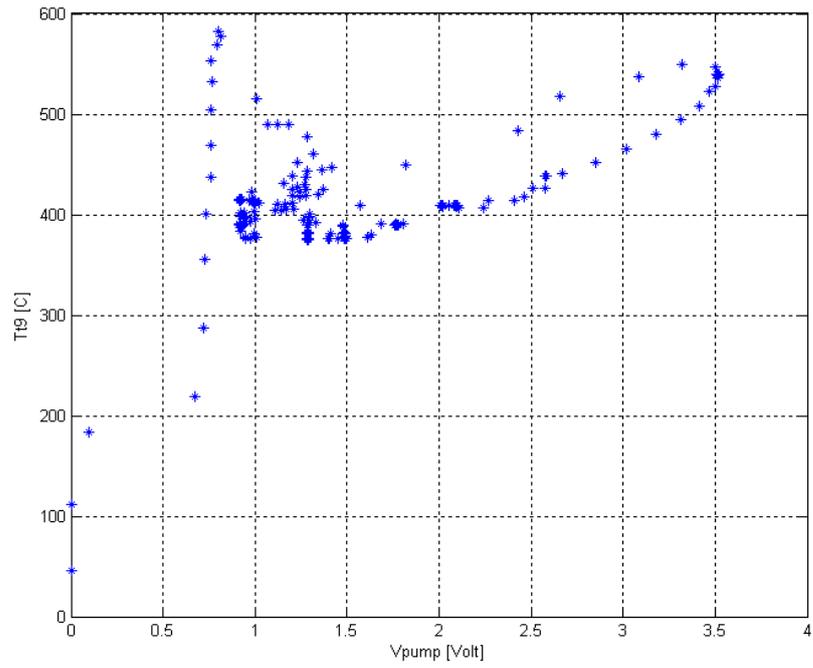


Figure 6-18 EGT vs. Pump Voltage for Test 1

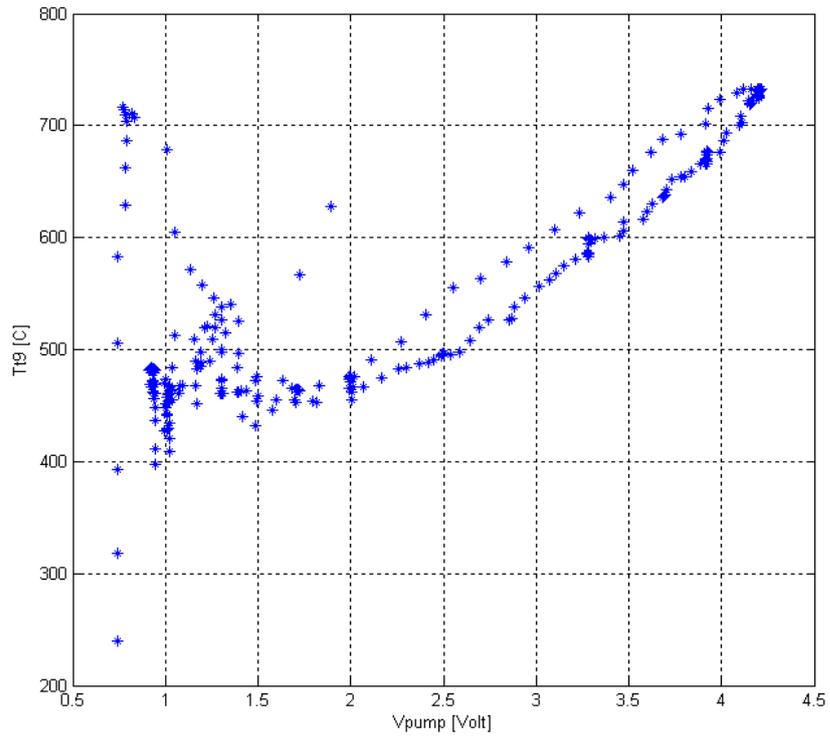


Figure 6-19 EGT vs. Pump Voltage for Test 4

This experimental data given in this part of the thesis can be utilized while designing the firmware of the AMT Olympus HP turbojet engine. Starting accelerations, temperature limits and pump voltages can be utilized to manipulate the system by the kerosene fuel. It is obvious that the controller design of the ECU works effectively for different input conditions. This data can be used to find suitable controller parameters for fuzzy logic or any other controller type.

6.6 Closure

In this chapter, the tests on sub-components of the ECU were discussed. The problems during these tests (and their corresponding) solutions were given. After finishing sub-components tests, ECU was connected to the gas turbine engine. Due to the use of extremely flammable mixtures, considerable attention must be exercised to avoid hazardous situations. Therefore, the experiments must be done in a safe region. Several starting algorithms were tested and the most suitable one was selected to start up the engine. The test results (temperature, shaft speed data, etc.) were collected over the RS232 serial communication port of the ECU. Finally, for comparison purposes, the original ECU of the AMT Olympus HP turbojet engine was tested and the accompanying results were presented.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Introduction

In consequence of the literature survey, it was noticed that gas turbine control researches are very limited in our country. Hence, literature survey was focused on the studies which were continuing in abroad. Most of them were about the commercial aircraft gas turbine engines or high power class power generators. These engines had a lot of control points and measurement devices. In this thesis work, a small gas turbine engine mathematical model was discussed. Afterwards, a control algorithm and an electronic control unit were designed and ECU was manufactured. Designed ECU is connected to the turbojet engine and some different auto start algorithms were developed. Also, kerosene fuel was applied to the system. Finally, original ECU was tested and result were examined.

7.2 Conclusion and Future Work

Developing a mathematical model of a small gas turbine engine was complicated due to the nonlinear differential equations as discussed in the previous sections. To design an effective mathematical model, it is crucial to determine the physical limitations of the system by consulting with the designer of the engine. It is known that gas turbine engine has four important sections that are compressor, combustion chamber, turbine and shaft. These components have some constants and assumptions to help the designer while developing a model. This theoretical knowledge has to be taken from the designers of the individual parts of the gas turbine engine. In this thesis work, these constants and assumptions were taken from the similar studies in literature and from “Small Gas Turbine Engine” project that was

continuing in Middle East Technical University Aerospace Engineering Department and was led by Prof. Dr. Nafiz Alemdaroğlu.

Literature survey shows that standard control methods such as PI, PID etc. have some restrictions that were given in Chapter 4. For instance, linearization of the model for a specific point is a method for this kind of control algorithms. Due to the problems taken into account, a fuzzy logic control algorithm, which can adjust itself to various situations, was selected. Fuzzy logic controller was compared to PID controller and it was shown that fuzzy logic controller could control the system whose parameters are varying under different circumstances. Finally, this fuzzy logic control algorithm was simulated through Matlab Simulink which includes a detailed mathematical model of the gas turbine engine under study. The simulation results demonstrated that fuzzy logic controller can control the small gas turbine under different circumstances.

In Chapter 5, an ECU was designed for a small gas turbine engine. While designing this ECU, commercial controllers were investigated to determine their inputs and outputs properties. As a result, a K-type thermocouple and shaft speed sensor measurements were used to provide the feedback signals to ECU. All sub components of the engine were tested, and their current-voltage limits were determined and convenient driver circuits were developed.

In this study, main objective is to design a controller for a small turbojet engine. For this purpose, a controller designed and tested with AMT Olympus HP turbojet engine and the result are discussed in the previous chapter. Most important problem about the experiment was the physical limitations for the engine components. For example, if the nozzle temperature exceeds 900°C , turbine can contact to the stator. Because of this reason, experiments were stopped manually if the temperature was exceeded to $800^{\circ} - 850^{\circ}\text{C}$. To specify the characteristics of AMT Olympus HP turbojet engine, original ECU was connected to the engine, several tests were done, and results are given in the previous chapter. These experiments show that;

- Shaft speed of the engine increases to 50 [krpm] in 25 seconds,
- Nozzle temperature reaches $600^{\circ} - 700^{\circ}\text{C}$ in automatic start intervals,
- Automatic start operation continues 40 second in all experiments,

- Nozzle temperature remains almost constant between 400° – 450° C at steady state conditions until shaft speed reaches 90 [krpm],
- Nozzle temperature reaches 750° – 800° C if the shaft speed is greater than 90 [krpm],
- Throttle increment between 30-80 [krpm] is lower than throttle increment between 80-105 [krpm]. This shows that the fuel consumption of the AMT small turbojet engine is increasing after 80 [krpm].
- Figure 6-16 and Figure 6-17 shows that the relations between shaft speed and pump voltage changes parabolic,
- Figure 6-18 and Figure 6-19 shows that the relations between temperature and pump voltage are both the same in different tests.

As a result, this work showed that designed electronic controller unit is capable of controlling a small gas turbine engine effectively. However, it has a great margin for improvement.

In the future work, this control unit can be improved by applying the data collected through a number of detailed investigations and experiments. New fuzzy logic parameters and membership functions can be developed to control the engine more effectively. To develop engine control algorithms on the devised ECU, a designer needs more detailed engine characteristics that can be acquired from these detailed experiments. Note that minor changes in the control algorithm (i.e. fuzzy logic controller parameters) may have a drastic effect on the controlled system dynamics and may cause the engine operate in a unsafe zone. For instance, the period, which the propane gas has to be opened or when the kerosene valve will open, must be determined with the designer of the gas turbine. Similarly, maximum shaft speed, maximum acceleration-deceleration values have to be determined beforehand. Thus, it is obvious that designing a gas turbine controller is a multidisciplinary research. After making several tests on the designed ECU, it was noticed that the hardware design of the ECU is capable of controlling a small gas turbine engine. However, firmware of the controller has to be built up to control the engine safely. When the kerosene is applied to the system, temperature of the nozzle increases drastically to 900° C. Due to this dangerous circumstances, original ECU was tested and results are given in Chapter 6. In the future works, these results can be utilized to obtain a map which is generated for steady state and transient conditions of AMT Olympus HP turbojet engine. The map can be used to

developed new fuzzy logic controller algorithms in the future. It is obvious that firmware investigations will be more important than the hardware developments. For this purpose, new control algorithms such as self-tuning, adaptive control, gain-scheduling etc. have to be studied in the future.

Consequently, with this study, what kinds of difficulties encountered by the control engineers are discussed and given some suggestions about gas turbine modeling, control and ECU design are given.

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] Lichtsinder, Michael, and Yeshayahou Levy. "Jet Engine Model for Control and Real-Time Simulations." *Journal of Engineering for Gas Turbines and Power* 128: 745-753, OCTOBER 2006.
- [3] Hung, W.W. "Dynamic Simulation of Gas-Turbine Generating Unit." *IEE PROCEEDINGS-C* 138: 342-350, JULY 1991.
- [4] 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.
- [5] Ailer, Piroska, Imre Santa, Gabor Szederkenyi, and M. Katalin Hangos. "Nonlinear Model-Building of a Low-Power Gas Turbine." *Periodica Polytechnica Ser. Transp. Eng.* 29: 117-135, OCTOBER 2001.
- [6] Koçer, Gülru, and Oğuz Uzol. "Real-Time Simulation of a Small Turbojet Engine." 4. *Ankara International Aerospace Conference*. Ankara, 2007.
- [7] Nagpal, M., A. Moshref, G.K. Morison, and P. Kundur. "Experience with Testing and Modeling of Gas Turbines." *Power Engineering Society Winter Meeting, IEEE.* 652-656, 2001.
- [8] 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.

- [9] Kim, J.H., T.W. Song, T.S. Kim, and S.T. Ro. "Dynamic Simulation of Full Startup Procedure of Heavy-Duty Gas Turbines." *Journal of Engineering for Gas Turbines and Power* 124: 510-516, JULY 2002.
- [10] 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.
- [11] Kim, J.H., T.W. Song, T.S. Kim, and S.T. Ro. "Model Development and Simulation of Transient Behavior of Heavy Duty Gas Turbines." *Journal of Engineering for Gas Turbines and Power* 123: 589-594, JULY 2001.
- [12] Watanabe, Airo, Semih M. Ölçmen, Robert Leland, and Kevin W. Whitaker. "Soft Computing Applications on SR-30 Turbojet Engine." *1. AIAA Intelligent Systems Technical Conference*. Chicago, IN, 2004.
- [13] Yu, Youhong, Lingen Chen, Fengrui Sun, and Chih Wu. "Matlab/Simulink-Based Simulation for Digital Control System of Marine Three-Shaft Gas Turbine." *Applied Energy* 80: 1-10, 2005.
- [14] Mu, Junxia, David Rees, and G.P. Liu. "Advanced Controller Design for Aircraft Gas Turbine Engines." *Control Engineering Practice* 13 : 1001-1015, 2005.
- [15] Ailer, P., G. Szederkenyi, and K.M. Hangos. "Model-BAsed Nonlinear Control of a Low-Power Gas Turbine." *15. IFAC Triennial World Congress*. Barcelona, 2002.
- [16] Andoga, Rudolf, Ladislav Madarasz, and Ladislav Fozo. "Digital Electronic Control of a Small Turbojet Engine - MPM 20." *12. International Conference on Intelligent Engineering Systems*. Miami, Florida. 37-40, 2008.
- [17] Kim, J.H., T.S. Kim, and S.T. Ro. "Analysis of the Dynamic Behaviour of Regenerative Gas Turbines." *Proceedings of the Institution of Mechanical Engineers* 215 : 339-346, 2001.
- [18] Kerrebrock, Jack L. *Aircraft Engines and Gas Turbines*. Cambridge, Massachusetts: The MIT Press, 1996.
- [19] Kulikov, Gennady G., and Haydn A. Thompson. *Dynamic Modelling of Gas Turbines*. London: Springer-Verlag, 2004.

- [20] Chipperfield, Andrew J., Beatrice Bica, and Peter J. Fleming. "[u] Fuzzy Scheduling Control of a Gas Turbine Aero-Engine: a Multiobjective Approach." *IEEE Transactions on Industrial Electronics*,; 536-548, JUNE 2002.
- [21] Sijak, Tomislav, Ognjen Kuljaca, Ljubomir Kuljaca, and Sejid Tesnjak. "Design of Fuzzy Regulator for Power System Secondary Load Frequency Control." *Proceedings of the 10th Mediterranean Conference on Control and Automation*. Lisbon, 2002.
- [22] Lin, Shih Tin, Jau Huai Lu, Ming Chong Hong, and Hsu Kwang Wu. "Real Time Dynamic Simulations of Single Spool Turbojet Engines." *Simulation*: 352-361, 1999.
- [23] Jie, Min Seok, Eun Jong Mo, Gyo Young Hong, and Kang Woong Lee. "Fuzzy Logic Controller for Turbojet Engine of Unmanned Aircraft." *Lecture Notes in Computer Science* 4251: 29-36, 2006.
- [24] India Defence. "<http://www.india-defence.com/>." *Last Checked: 21.11.2009*.
- [25] The Naval Database Project. "<http://www.naval-database.com/>." *Last Checked: 15.11.2009*.
- [26] Analog Devices, Datasheet of AD594/AD595.
- [27] ST Microelectronics, Datasheet of L298.
- [28] Seattle Robotics. "<http://www.seattlerobotics.org/encoder/Mar98/fuz/flindex.html>." *Last Checked: 15.12.2209*.
- [29] An Introduction To Fuzzy Control Systems. "<http://www.faqs.org/docs/fuzzy/>." *Last Checked:12.07.2009*.
- [30] Yen, Johnand, and Reza Langari. *Fuzzy logic : intelligence, control and information*. New Jersey: Prentice Hall, 1999.
- [31] Kovačić, Zdenko, and Stjepan Bogdan. *Fuzzy controller design : theory and applications*. Boca Raton, FL: CRC/Taylor & Francis, 2006.
- [32] AMT Netherlands. "<http://www.amtjets.com/download.php>." *Last Checked: 25.12.2009*.

APPENDIX A

MATLAB CODES

In this section of the thesis work, MATLAB code of PI and fuzzy logic controller are given. Sub programs, which were used in the main program, are also given in this section.

A-1 PI Controller

```
%-----initial conditions-----
clc;clear;
format short e
i=1;m=1;n=1;c=1;q=1;v=1;kl=0;
Pt_2=93380; %pascal
Tt_2=303; %Kelvin
Pt_ref=100000; %pascal (from compressor map)
Tt_ref=298; %Kelvin (from compressor map)
sigma_comb= 0.95; %Combustion chamber pressor ratio
nu_comb=0.6; %Combustion chamber efficiency
Pi_n=0.95; %Nozzle pressure ratio
M9=1; %Choke condition
M0=0; %Assumption
H=0; %Hight
teta=(Tt_2/Tt_ref); %Dimensionless temperature parameter
delta=(Pt_2/Pt_ref); %Dimensionless pressure parameter
N=65000 ; %RPM
I=1.787e-4; % Moment of Inertia _from CATIA [kg*m^2]
eff_t=0.85; %efficiency of the turbine (Can Dedekarginoğlu)
gama_c=1.4; %specific heat ratio of compressor
gama_t=1.35; %specific heat ratio of turbine
Qf=43250e3; % [J/kg] Lower calorific value of fuel
R=286.9; % [J/(kg*K)] Individual gas constant
V_comb=6.687233e-4; % Volume of the combustion chamber
PI_C=9.06e-21*N^4-1.365e-15*N^3+3.3806e-10*N^2-9.7556e-6*N+1.1929;
Cp_a=-1.9747e-028*N^6+ 8.3860e-023*N^5-1.4447e-017*N^4+1.2943e-
012*N^3-6.2674e-008*N^2+1.5743e-003*N+9.8846e+002;
Tt_4=5.5099e-017*N^4-1.1969e-011*N^3+9.7144e-007*N^2-3.7423e-
002*N+1.4273e+003;
mf_dot=0.00456;
Kp=0.00000456/10000;
Ki=0.00000000035;
```

```

MODD=2;
A_0=0.001378;
A_9=0.00302;
%-----BETA LINE METHOD-----
h=0.001;      %Delta "t"
E(1)=0;
for j=1:(1.5/h);
    N_Ref(j)=80000;
    N_R(j)=N(j)/(teta)^0.5;      % Corrected RPM
    Tt_2(j+1)=Tt_2(j);
    Pt_2(j+1)=Pt_2(j);

PI_C_m=[0 1.560 1.648 1.678 0 0 0 0 0 0 0 0
         0 2.019 2.172 2.282 2.337 0 0 0 0 0 0 0
         0 0 2.545 2.750 2.938 3.067 3.121 0 0 0 0 0
         0 0 0 0 3.219 3.438 3.676 3.843 3.955 0 0 0
         0 0 0 0 0 3.861 4.090 4.317 4.542 4.695 0];

N_cor=[61000 81000 97000 111000 123000 ];
while m==1;
    if N_cor(i) <= N_R(j)
        if N_R(j) <= N_cor(i+1)

            k=(N_cor(i+1)-N_R(j))./(N_cor(i+1)-N_cor(i));
            m=0;t=i;
        end
    end
    i=i+1;
end
i=1;
%-----**-----%
for q = 1 : 12;
    N_R_matrix(q)=PI_C_m(t+1,q)-(PI_C_m(t+1,q)-PI_C_m(t,q)).*k;
end
while n==1;
    if N_R_matrix(c) <= PI_C(j)
        if PI_C(j) <= N_R_matrix(c+1)

            Beta=c-(-1*(N_R_matrix(c)-PI_C(j)))./(N_R_matrix(c)-
N_R_matrix(c+1));
            n=0;
        end
    end
    c=c+1;
end
c=1;
%-----Mass Flow Rate Calculations-----
x=[1 2 3 4 5 6 7 8 9 10 11 12];
y=[61000 81000 97000 111000 123000 ];
z=[0 0.2777 0.2185 0.1387 0 0 0 0 0 0 0 0
   0 0.4155 0.3758 0.3226 0.2531 0 0 0 0 0 0 0
   0 0 0.4904 0.4638 0.4358 0.3894 0.318 0 0 0 0 0
   0 0 0 0 0.5206 0.5045 0.4881 0.4514 0.3988 0 0 0
   0 0 0 0 0 0.5435 0.5271 0.5094 0.4918 0.4517 0];
%-----Efficiency Calculations-----

```

```

x_1=[1 2 3 4 5 6 7 8 9 10 11 12];
y_1=[61000 81000 97000 111000 123000 ];
z_1=[0 0.711 0.759 0.69 0 0 0 0 0 0 0 0
      0 0.669 0.733 0.76 0.742 0 0 0 0 0 0 0
      0 0 0.646 0.698 0.735 0.752 0.726 0 0 0 0 0
      0 0 0 0 0.65 0.68 0.704 0.715 0.706 0 0 0
      0 0 0 0 0 0.62 0.64 0.655 0.664 0.66 0];
%-----Results-----
m_dot_MAP(j)=interp2(x,y,z,Beta,N_R(j));
eff_c(j)=interp2(x_1,y_1,z_1,Beta,N_R(j));
%-----Beta line end-----
%-----Compressor-----
mc_dot(j)=m_dot_MAP(j)*(delta)/((teta)^0.5);
Tt_3(j)=Tt_2(j)*(1+((1/eff_c(j))*((PI_C(j)^((gama_c-1)/gama_c))-
1)));
%Tt_3(j)=370;
Pt_3(j)=PI_C(j)*Pt_2(j);
Ta(j)=(Tt_2(j)+Tt_3(j))/2;
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;
%-----Combustion Chamber-----
mf_dot(j+1)=mf_dot(j);
kont=rem(j,MODD);
if kont==0 ;
E(j)=N_Ref(j)-N(j);
mf_dot(j)=mf_dot(j-1)+ (Kp+Ki*MODD). *E(j)+Kp*E(j-1);%-----
mf_dot(j+1)=mf_dot(j);
end
Tg(j)=(Tt_4(j)+Tt_3(j))/2;
Tm(j)=Tg(j);
f(j)=mf_dot(j)/mc_dot(j);
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;
Cp_g(j)=Cp_a(j)+(f(j)/(1+f(j)))*Bt(j);
Cv_comb(j)=Cp_g(j)-R;
Pt_4(j)=sigma_comb*Pt_3(j);
Pm(j)=(Pt_4(j)+Pt_3(j))/2;
Rho(j)=Pm(j)/(R*Tm(j));
M_comb(j)=Rho(j)*V_comb;
%-----Runge Kutta fourth order for Tt_4 -----
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));
K1(j)=F(j);
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
t(j))/(Cv_comb(j)*M_comb(j));
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
t(j))/(Cv_comb(j)*M_comb(j));
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));

```

```

Tt_4(j+1)=Tt_4(j)+(h/6)*(K1(j)+2*K2(j)+2*K3(j)+K4(j));
Pt_4_D(j)=Tt_4(j)*R*M_comb(j)/V_comb;
%-----Runge Kutta fourth order for Pt_4-----
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));
M1(j)=H(j);
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));
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));
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));
Pt_4(j+1)=Pt_4(j)+h/6*(M1(j)+2*M2(j)+2*M3(j)+M4(j));
Pt_3(j+1)=Pt_4(j+1)/sigma_comb;
PI_C(j+1)=Pt_3(j+1)/Pt_2(j);
%-----TURBINE-----
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)-
1.0868e+000;
Tao_c(j)=Tt_3(j)/Tt_2(j);
Teta_2(j)=1;
Teta_3(j)=Tt_4(j)/Tt_2(j);
Tao_t(j)=1-
((Cp_a(j)*Teta_2(j))/(Cp_g(j)*(1+f(j))*Teta_3(j)))*(Tao_c(j)-1);
Tt_5(j)=685;
%-----Work Balance-----
Pt(j)=((mc_dot(j)+mf_dot(j))*(Tt_4(j)-Tt_5(j))*Cp_g(j));
Pc(j)=mc_dot(j)*Cp_a(j)*(Tt_3(j)-Tt_2(j));
Pf(j)=500;
if Pf(j)<0
Pf(j)=0;
end
W(j)=N(j)*pi/30;
W(j+1)=sqrt((2*(Pt(j)-Pc(j)-Pf(j))/I)*h+W(j)^2);
N(j+1)=W(j+1)*30/pi;
%-----THRUST-----
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;
m9_dot(j)=mc_dot(j)+mf_dot(j);
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;
V9(j)=(gama_t*R*T_9(j))^0.5;
V0(j)=mc_dot(j)/(Rho_0(j)*A_0);
F(j)=m9_dot(j)*V9(j)-mc_dot(j)*V0(j);
end
subplot(1,2,1);plot(N);grid on;xlabel('t (ms)');ylabel('N
(rpm)')
subplot(1,2,2);plot(Tt_4);grid on;xlabel('t
(ms)');ylabel('Tt_4 (K)')

```

A-2 Fuzzy Logic Controller

```

%-----initial conditions-----17/11/2007-----
clc;clear;
format short e
i=1;m=1;nn=1;c=1;q=1;v=1;kl=0;
Pt_2=93380;           %pascal
Tt_2=303;            %Kelvin
Pt_ref=100000;       %pascal (from compressor map)
Tt_ref=298;          %Kelvin (from compressor map)
sigma_comb= 0.95;    %Combustion chamber pressor ratio
nu_comb=0.6;         %Combustion chamber efficiency
Pi_n=0.95;           %Nozzle pressure ratio
M9=1;                %Choke condition
M0=0;                %Assumption
H=0;                 %Hight
teta=(Tt_2/Tt_ref); %Dimensionless temperature parameter
delta=(Pt_2/Pt_ref); %Dimensionless pressure parameter
N=65000 ;            %RPM
I=1.787e-4;          % Moment of Inertia _from CATIA [kg*m^2]
eff_t=0.85;          %efficiency of the turbine (Can Dedekargınođlu)
gama_c=1.4;          %specific heat ratio of compressor
gama_t=1.35;         %specific heat ratio of turbine
Qf=43250e3;          % [J/kg] Lower calorific value of fuel
R=286.9;             % [J/(kg*K)] Individual gas constant
V_comb=6.687233e-4; % Volume of the combustion chamber
PI_C=9.06e-21*N^4-1.365e-15*N^3+3.3806e-10*N^2-9.7556e-6*N+1.1929;
Cp_a=-1.9747e-028*N^6+ 8.3860e-023*N^5-1.4447e-017*N^4+1.2943e-
012*N^3-6.2674e-008*N^2+1.5743e-003*N+9.8846e+002;
Tt_4=5.5099e-017*N^4-1.1969e-011*N^3+9.7144e-007*N^2-3.7423e-
002*N+1.4273e+003;
mf_dot=0.00456;
A_0=0.001378;
A_9=0.00302;
%-----BETA LINE METHOD-----
h=0.001;           %Delta "t"
Tau=0.15;T=0.001;n=1/120000;nce=1/120000;de=0.015;ty=0.5;s=0;
global nu_p uu
for j=1:(4/h);
    N_Ref(j)=85000;
    e(j)=N_Ref(j)-N(j);
    if j==1;
        ce(j)=e(j)-0;
    else
        ce(j)=e(j)-e(j-1);
    end
    en(j)=e(j)*n;
    cen(j)=ce(j)*nce;
%-----FUZZYYYY-----
%-----RULE 1 -----

```

```

%if en(i) is P and cen(i) is P then delU_n is P
    positive(en(j));
    nue1=ans;
    positive(cen(j));
    nuce1=ans;
        if nue1<=nuce1
            nu1=nue1;
        else
            nu1=nuce1;
        end
%-----RULE 2 -----
%if en(i) is N and cen(i) is N then delU_n is N
    negative(en(j));
    nue2=ans;
    negative(cen(j));
    nuce2=ans;
        if nue2<=nuce2
            nu2=nue2;
        else
            nu2=nuce2;
        end
%-----RULE 3 -----
%if en(i) is P and cen(i) is Z then delU_n is P
    positive(en(j));
    nue3=ans;
    zero(cen(j));
    nuce3=ans;
        if nue3<=nuce3
            nu3=nue3;
        else
            nu3=nuce3;
        end
%-----RULE 4 -----
%if en(i) is N and cen(i) is Z then delU_n is N
    negative(en(j));
    nue4=ans;
    zero(cen(j));
    nuce4=ans;
        if nue4<=nuce4
            nu4=nue4;
        else
            nu4=nuce4;
        end
%-----RULE 5 -----
%if en(i) is P and cen(i) is N then delU_n is Z
    positive(en(j));
    nue5=ans;
    negative(cen(j));
    nuce5=ans;
        if nue5<=nuce5
            nu5=nue5;
        else
            nu5=nuce5;
        end
%-----RULE 6 -----
%if en(i) is N and cen(i) is P then delU_n is Z

```

```

negative(en(j));
nue6=ans;
positive(cen(j));
nuce6=ans;
    if nue6<=nuce6
        nu6=nue6;
    else
        nu6=nuce6;
    end
%-----RULE 7 -----
%if en(i) is Z and cen(i) is Z then delU_n is Z

zero(en(j));
nue7=ans;
zero(cen(j));
nuce7=ans;
    if nue7<=nuce7
        nu7=nue7;
    else
        nu7=nuce7;
    end
%-----RULE 8 -----
%if en(i) is Z and cen(i) is N then delU_n is N

zero(en(j));
nue8=ans;
negative(cen(j));
nuce8=ans;
    if nue8<=nuce8
        nu8=nue8;
    else
        nu8=nuce8;
    end
%-----RULE 9 -----
%if en(i) is Z and cen(i) is P then delU_n is P
zero(en(j));
nue9=ans;
positive(cen(j));
nuce9=ans;
    if nue9<=nuce9
        nu9=nue9;
    else
        nu9=nuce9;
    end
%-----DEFUZZIFICATION-----
dR1=positive2(nu1);
dR2=negative2(nu2);
dR3=positive2(nu3);
dR4=negative2(nu4);
dR5=zero2(nu5);
dR6=zero2(nu6);
dR7=zero2(nu7);
dR8=negative2(nu8);
dR9=positive2(nu9);
d=(nu1*dR1+nu2*dR2+nu3*dR3+nu4*dR4+nu5*dR5+nu6*dR6+nu7*dR7+nu8*dR8+n
u9*dR9)/(nu1+nu2+nu3+nu4+nu5+nu6+nu7+nu8+nu9);

```

```

delta_U(j)=d*de;
if j==1 mf_dot(j)=d*de;
else
mf_dot(j)=mf_dot(j-1)+T*d*de;
end
%-----BETALINEE-----

N_R(j)=N(j)/(teta)^0.5; % Corrected RPM
Tt_2(j+1)=Tt_2(j);
Pt_2(j+1)=Pt_2(j);
PI_C_m=[0 1.560 1.648 1.678 0 0 0 0 0 0 0 0
0 2.019 2.172 2.282 2.337 0 0 0 0 0 0 0
0 0 2.545 2.750 2.938 3.067 3.121 0 0 0 0 0
0 0 0 0 3.219 3.438 3.676 3.843 3.955 0 0 0
0 0 0 0 0 3.861 4.090 4.317 4.542 4.695 0];

N_cor=[61000 81000 97000 111000 123000];
while m==1;
if N_cor(i) <= N_R(j)
if N_R(j) <= N_cor(i+1)

k=(N_cor(i+1)-N_R(j))./(N_cor(i+1)-N_cor(i));
m=0;t=i;
end
end
i=i+1;
end
i=1;
%-----**-----%
for q = 1 : 12;
N_R_matrix(q)=PI_C_m(t+1,q)-(PI_C_m(t+1,q)-PI_C_m(t,q)).*k;
end
while nn==1;
if N_R_matrix(c) <= PI_C(j)
if PI_C(j) <= N_R_matrix(c+1)

Beta=c-(-1*(N_R_matrix(c)-PI_C(j)))./(N_R_matrix(c)-
N_R_matrix(c+1));
nn=0;
end
end
c=c+1;
end
c=1;
%-----Mass Flow Rate Calculations-----
x=[1 2 3 4 5 6 7 8 9 10 11 12];
y=[61000 81000 97000 111000 123000 ];
z=[0 0.2777 0.2185 0.1387 0 0 0 0 0 0 0 0
0 0.4155 0.3758 0.3226 0.2531 0 0 0 0 0 0 0
0 0 0.4904 0.4638 0.4358 0.3894 0.318 0 0 0 0 0
0 0 0 0 0.5206 0.5045 0.4881 0.4514 0.3988 0 0 0
0 0 0 0 0 0.5435 0.5271 0.5094 0.4918 0.4517 0];
%-----Efficiency Calculations-----
x_1=[1 2 3 4 5 6 7 8 9 10 11 12];

```

```

y_1=[61000 81000 97000 111000 123000 ];
z_1=[0 0.711 0.759 0.69 0 0 0 0 0 0 0 0
      0 0.669 0.733 0.76 0.742 0 0 0 0 0 0 0
      0 0 0.646 0.698 0.735 0.752 0.726 0 0 0 0 0
      0 0 0 0 0.65 0.68 0.704 0.715 0.706 0 0 0
      0 0 0 0 0 0.62 0.64 0.655 0.664 0.66 0];

%-----Results-----
m_dot_MAP(j)=interp2(x,y,z,Beta,N_R(j));
eff_c(j)=interp2(x_1,y_1,z_1,Beta,N_R(j));
%-----Beta line end-----
%-----Compressor-----
mc_dot(j)=m_dot_MAP(j)*(delta)/((teta)^0.5);
Tt_3(j)=Tt_2(j)*(1+((1/eff_c(j))*((PI_C(j))^((gama_c-1)/gama_c))-
1)));
Pt_3(j)=PI_C(j)*Pt_2(j);
Ta(j)=(Tt_2(j)+Tt_3(j))/2;
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;
%-----Combustion Chamber-----
Tg(j)=(Tt_4(j)+Tt_3(j))/2;
Tm(j)=Tg(j);
f(j)=mf_dot(j)/mc_dot(j);
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;
Cp_g(j)=Cp_a(j)+(f(j)/(1+f(j)))*Bt(j);
%Cp_g(j)=1088.2;
Cv_comb(j)=Cp_g(j)-R;
Pt_4(j)=sigma_comb*Pt_3(j);
Pm(j)=(Pt_4(j)+Pt_3(j))/2;
Rho(j)=Pm(j)/(R*Tm(j));
M_comb(j)=Rho(j)*V_comb;
%-----Runge Kutta fourth order for Tt_4 -----
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));
K1(j)=F(j);
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
t(j))/(Cv_comb(j)*M_comb(j));
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
t(j))/(Cv_comb(j)*M_comb(j));
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));
Tt_4(j+1)=Tt_4(j)+(h/6)*(K1(j)+2*K2(j)+2*K3(j)+K4(j));
Pt_4_D(j)=Tt_4(j)*R*M_comb(j)/V_comb;
%-----Runge Kutta fourth order for Pt_4-----
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));
M1(j)=H(j);
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));

```

```

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));
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));
Pt_4(j+1)=Pt_4(j)+h/6*(M1(j)+2*M2(j)+2*M3(j)+M4(j));
Pt_3(j+1)=Pt_4(j+1)/sigma_comb;
PI_C(j+1)=Pt_3(j+1)/Pt_2(j);
%-----TURBINE-----
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)-
1.0868e+000;
Tao_c(j)=Tt_3(j)/Tt_2(j);
Teta_2(j) =1;
Teta_3(j) = Tt_4(j) / Tt_2(j);
Tao_t(j)=1-
((Cp_a(j)*Teta_2(j))/(Cp_g(j)*(1+f(j))*Teta_3(j)))*(Tao_c(j)-1);
Tt_5(j)=685;
%-----Work Balance-----
Pt(j)=(mc_dot(j)+mf_dot(j))*(Tt_4(j)-Tt_5(j))*Cp_g(j);
Pc(j)=mc_dot(j)*Cp_a(j)*(Tt_3(j)-Tt_2(j));
Pf(j)=500;
if Pf(j)<0
    Pf(j)=0;
end
W(j)=N(j)*pi/30;
W(j+1)=sqrt((2*(Pt(j)-Pc(j)-Pf(j))/I)*h+W(j)^2);
N(j+1)=W(j+1)*30/pi;
%-----THRUST-----
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;
m9_dot(j)=mc_dot(j)+mf_dot(j);
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;
V9(j)=(gama_t*R*T_9(j))^0.5;
V0(j)=mc_dot(j)/(Rho_0(j)*A_0);
F(j)=m9_dot(j)*V9(j)-mc_dot(j)*V0(j);
end
subplot(1,3,1);plot(N);grid on;xlabel('t (ms)');ylabel('N
(rpm)')
subplot(1,3,2);plot(Tt_4);grid on;xlabel('t
(ms)');ylabel('Tt_4(K)')
subplot(1,3,3);plot(F);grid on;xlabel('t
(ms)');ylabel('Thrust(N)')

```

A-2.1 Subprograms

positive.m

```

function [nu_p]=positive(ee)
    if ee>=0 && ee<=0.5
        nu_p=2*ee;
    elseif ee>=0.5 && ee<=1
        nu_p=1;
    end

```

```

else
    nu_p=0;
end

```

negative.m

```

function [nu_n]=negative(ee)
    if ee<=-0.5 && ee>=-1
        nu_n=1;
    elseif ee<=0 && ee>=-0.5
        nu_n=-2*ee;
    else
        nu_n=0;
    end

```

zero.m

```

function [nu_z]=zero(ee)
    if ee<=0 && ee>=-0.5
        nu_z=1+2*ee;
    elseif ee>=0 && ee<=0.5
        nu_z=1-2*ee;
    else
        nu_z=0;
    end

```

positive2.m

```

function [dd]=positive2(nu)
if nu==0
    dd=0;
else
A1=(1-nu/2)*nu;
A2=( (nu)^2)/4;
d1=nu/4+0.5;
d2=nu/3;
dd=(d1*A1+d2*A2)/(A1+A2);
end

```

negative.m

```

function [dd]=negative2(nu)
if nu==0
    dd=0;
else
A1=(1-nu/2)*nu;
A2=( (nu)^2)/4;
d1=- (nu/4+0.5);
d2=-nu/3;
dd=(d1*A1+d2*A2)/(A1+A2);
end

```

zero2.m

```

function [dd]=zero2(nu)
    dd=0;

```

A-3 PID and Fuzzy Logic Code for a Second Order System

A-3.1 PID Simulation Code for $\zeta=1.5$ and $\omega_n=10$ [rad/s]

```
clc;clear;
T      = 0.1;          %Sampling Time
initial = 0;
y(1)   = initial;
y0     = initial;
yd=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Kp      = 0.01;
Kd      = 0.01;
Ki      = 5;
b0      = Kp+Kd/T+Ki*T;
b1      = -Kp-(2*Kd)/T;
b2      = Kd/T;
%-----
for i=1:50/T;
    if i<100
        yd=yd+0.3;
    elseif i>=100 && i<200
        yd=30;
    elseif i>=200 && i<300
        yd=yd-0.3;
    else
        yd=0;
    end
    yd_graph(i)=yd;
    %y(1)=0;
    e(1)=yd-y(1);
%-----PID Controller-----
if i==1
    uu(i)=b0*e(i);
end
if i==2
    uu(i)=uu(i-1)+b0*e(i)+b1*e(i-1);
end
if i>2
    uu(i)=uu(i-1)+b0*e(i)+b1*e(i-1)+b2*e(i-2);
end
%-----Second Order System-----
%-----
if i==1
    y(i)=0;
end
if i==2
    y(i)=0.2134*uu(i)
end
if i>2
    y(i)=0.7555*y(i-1)-0.04979*y(i-2)+0.2134*uu(i-1)+0.08097*uu(i-2);
end
end
```

```

%-----
e(i+1)=yd-y(i);
end
subplot(2,1,2); plot(e);ylabel('error(t)');xlabel('t[sec/10]');grid
on;
hold all
subplot(2,1,1); plot(y);ylabel('y(t)');xlabel('t[sec/10]');grid on;
hold all
plot(yd_graph)

```

A-3.2 Fuzzy Logic Simulation Code for $\zeta=1.5$ and $\omega_n=10$ [rad/s]

```

clc;clear;
Tau=0.5;T=0.1;n=1/30;nce=1/30;de=25;ty=0.5;s=0;
global nu_p uu
initial =0;
y(1) =initial;
y0 =initial;
%y=initial;
Zeta = 0.1;
Wn = 10;
KK = 0; % initial condition (velocity)...
tt = 0;
yd=0;
for i=1:50/T;
    tt=tt+T;
    if i<100
        yd=yd+0.3;
    elseif i>=100 && i<200
        yd=30;
    elseif i>=200 && i<300
        yd=yd-0.3;
    else
        yd=0;
    end

    yd_graph(i)=yd;
    e(1)=yd-y(1);
    ce(1)=e(1)-0;
    en(i)=e(i)*n;
    cen(i)=ce(i)*nce;

%-----RULE 1 -----
%if en(i) is P and cen(i) is P then delU_n is P
    positive(en(i));
    nuel=ans;
    positive(cen(i));
    nucel=ans;
    if nuel<=nucel
        nul=nuel;
    else
        nul=nucel;
    end
%-----RULE 2 -----
%if en(i) is N and cen(i) is N then delU_n is N

```

```

negative(en(i));
nue2=ans;
negative(cen(i));
nuce2=ans;
    if nue2<=nuce2
        nu2=nue2;
    else
        nu2=nuce2;
    end
%-----RULE 3 -----
%if en(i) is P and cen(i) is Z then delU_n is P
positive(en(i));
nue3=ans;
zero(cen(i));
nuce3=ans;
    if nue3<=nuce3
        nu3=nue3;
    else
        nu3=nuce3;
    end
%-----RULE 4 -----
%if en(i) is N and cen(i) is Z then delU_n is N
negative(en(i));
nue4=ans;
zero(cen(i));
nuce4=ans;
    if nue4<=nuce4
        nu4=nue4;
    else
        nu4=nuce4;
    end

%-----RULE 5 -----
%if en(i) is P and cen(i) is N then delU_n is Z
positive(en(i));
nue5=ans;
negative(cen(i));
nuce5=ans;
    if nue5<=nuce5
        nu5=nue5;
    else
        nu5=nuce5;
    end
%-----RULE 6 -----
%if en(i) is N and cen(i) is P then delU_n is Z
negative(en(i));
nue6=ans;
positive(cen(i));
nuce6=ans;
    if nue6<=nuce6
        nu6=nue6;
    else
        nu6=nuce6;
    end
%-----RULE 7 -----

```

```

%if en(i) is Z and cen(i) is Z then delU_n is Z
    zero(en(i));
    nue7=ans;
    zero(cen(i));
    nuce7=ans;
    if nue7<=nuce7
        nu7=nue7;
    else
        nu7=nuce7;
    end
%-----RULE 8 -----
%if en(i) is Z and cen(i) is N then delU_n is N

    zero(en(i));
    nue8=ans;
    negative(cen(i));
    nuce8=ans;
    if nue8<=nuce8
        nu8=nue8;
    else
        nu8=nuce8;
    end
%-----RULE 9 -----
%if en(i) is Z and cen(i) is P then delU_n is P
    zero(en(i));
    nue9=ans;
    positive(cen(i));
    nuce9=ans;
    if nue9<=nuce9
        nu9=nue9;
    else
        nu9=nuce9;
    end
dR1=positive2(nu1);
dR2=negative2(nu2);
dR3=positive2(nu3);
dR4=negative2(nu4);
dR5=zero2(nu5);
dR6=zero2(nu6);
dR7=zero2(nu7);
dR8=negative2(nu8);
dR9=positive2(nu9);
d=(nu1*dR1+nu2*dR2+nu3*dR3+nu4*dR4+nu5*dR5+nu6*dR6+nu7*dR7+nu8*dR8+nu9*dR9)/(nu1+nu2+nu3+nu4+nu5+nu6+nu7+nu8+nu9);
delta_U(i)=d*de;
if i==1 U(i)=d*de;
else
U(i)=U(i-1)+T*d*de;
end
uu=U(i);
%-----Second Order System-----
%-----
if i==1
    y(i)=0;
end

```



```

end
if i>2
    uu(i)=uu(i-1)+b0*e(i)+b1*e(i-1)+b2*e(i-2);
end
%-----
%-----Second Order Sytem-----
%-----
if i==1
    y(i)=0;
end
if i==2
    y(i)=0.431*uu(i)
end
if i>2
    y(i)=0.9854*y(i-1)-0.8187*y(i-2)+0.431*uu(i-1)+0.4023*uu(i-2);
end
%-----
e(i+1)=yd-y(i);
end
subplot(2,1,2); plot(e);ylabel('error(t)');xlabel('t[sec/10]');grid
on;
hold all
subplot(2,1,1); plot(y);ylabel('y(t)');xlabel('t[sec/10]');grid on;
hold all
plot(yd_graph)

```

A-3.4 Fuzzy Logic Simulation Code for $\zeta=0.1$ and $\omega_n=10$ [rad/s]

```

clc;clear;
Tau=0.5;T=0.1;n=1/35;nce=1/35;de=25;ty=0.5;s=0;
global nu_p uu
initial =0;
y(1)    =initial;
y0      =initial;
Zeta    = 0.1;
Wn      = 10;
KK      = 0;  % initial condition (velocity)...
tt      = 0;
yd=0;
for i=1:50/T;
    tt=tt+T;
    if i<100
        yd=yd+0.3;
    elseif i>=100 && i<200
        yd=30;
    elseif i>=200 && i<300
        yd=yd-0.3;
    else
        yd=0;
    end
    yd_graph(i)=yd;

```

```

e(1)=yd-y(1);
ce(1)=e(1)-0;
en(i)=e(i)*n;
cen(i)=ce(i)*nce;
%-----RULE 1 -----
%if en(i) is P and cen(i) is P then delU_n is P

positive(en(i));
nue1=ans;
positive(cen(i));
nucel=ans;
    if nue1<=nucel
        nul=nue1;
    else
        nul=nucel;
    end
%-----RULE 2 -----
%if en(i) is N and cen(i) is N then delU_n is N

negative(en(i));
nue2=ans;
negative(cen(i));
nuce2=ans;
    if nue2<=nuce2
        nu2=nue2;
    else
        nu2=nuce2;
    end
%-----RULE 3 -----
%if en(i) is P and cen(i) is Z then delU_n is P

positive(en(i));
nue3=ans;
zero(cen(i));
nuce3=ans;
    if nue3<=nuce3
        nu3=nue3;
    else
        nu3=nuce3;
    end
%-----RULE 4 -----
%if en(i) is N and cen(i) is Z then delU_n is N

negative(en(i));
nue4=ans;
zero(cen(i));
nuce4=ans;
    if nue4<=nuce4
        nu4=nue4;
    else
        nu4=nuce4;
    end
%-----RULE 5 -----
%if en(i) is P and cen(i) is N then delU_n is Z

```

```

positive(en(i));
nue5=ans;
negative(cen(i));
nuce5=ans;
    if nue5<=nuce5
        nu5=nue5;
    else
        nu5=nuce5;
    end
%-----RULE 6 -----
%if en(i) is N and cen(i) is P then delU_n is Z
negative(en(i));
nue6=ans;
positive(cen(i));
nuce6=ans;
    if nue6<=nuce6
        nu6=nue6;
    else
        nu6=nuce6;
    end
%-----RULE 7 -----
%if en(i) is Z and cen(i) is Z then delU_n is Z
zero(en(i));
nue7=ans;
zero(cen(i));
nuce7=ans;
    if nue7<=nuce7
        nu7=nue7;
    else
        nu7=nuce7;
    end
%-----RULE 8 -----
%if en(i) is Z and cen(i) is N then delU_n is N
zero(en(i));
nue8=ans;
negative(cen(i));
nuce8=ans;
    if nue8<=nuce8
        nu8=nue8;
    else
        nu8=nuce8;
    end
%-----RULE 9 -----
%if en(i) is Z and cen(i) is P then delU_n is P
zero(en(i));
nue9=ans;
positive(cen(i));
nuce9=ans;
    if nue9<=nuce9
        nu9=nue9;
    else
        nu9=nuce9;
    end
dR1=positive2(nu1);
dR2=negative2(nu2);
dR3=positive2(nu3);

```



```

        yd=yd+0.3;
    elseif i>=100 && i<200
        yd=30;
    elseif i>=200 && i<300
        yd=yd-0.3;
    else
        yd=0;
    end
    yd_graph(i)=yd;
    e(1)=yd-y(1);
%-----PID Controller-----
if i==1
    uu(i)=b0*e(i);
end
if i==2
    uu(i)=uu(i-1)+b0*e(i)+b1*e(i-1);
end
if i>2
    uu(i)=uu(i-1)+b0*e(i)+b1*e(i-1)+b2*e(i-2);
end
%-----Second Order System-----
if i==1
    y(i)=0;
end
if i==2
    y(i)=0.004802*uu(i)
end
if i>2
    y(i)=1.878*y(i-1)-0.8869*y(i-2)+0.004802*uu(i-1)+0.004614*uu(i-
2);
end
%-----
e(i+1)=yd-y(i);
end
subplot(2,1,2); plot(e);ylabel('error(t)');xlabel('t[sec/10]');grid
on;
hold all
subplot(2,1,1); plot(y);ylabel('y(t)');xlabel('t[sec/10]');grid on;
hold all
plot(yd_graph)

```

A-3.4 Fuzzy Logic Simulation Code for $\zeta=0.6$ and $\omega_n=1$ [rad/s]

```

clc;clear;
Tau=0.5;T=0.1;n=1/30;nce=1/30;de=20;ty=0.5;s=0;
global nu_p uu
initial =0;
y(1)    =initial;
y0      =initial;
Zeta    = 0.1;
Wn      = 10;
KK      = 0; % initial condition (velocity)...

```

```

tt      = 0;
yd=0;
for i=1:50/T;
    tt=tt+T;
    if i<100
        yd=yd+0.3;
    elseif i>=100 && i<200
        yd=30;
    elseif i>=200 && i<300
        yd=yd-0.3;
    else
        yd=0;
    end
    yd_graph(i)=yd;
    e(1)=yd-y(1);
    ce(1)=e(1)-0;
    en(i)=e(i)*n;
    cen(i)=ce(i)*nce;
%-----RULE 1 -----
%if en(i) is P and cen(i) is P then delU_n is P
    positive(en(i));
    nue1=ans;
    positive(cen(i));
    nuce1=ans;
    if nue1<=nuce1
        nul=nue1;
    else
        nul=nuce1;
    end
%-----RULE 2 -----
%if en(i) is N and cen(i) is N then delU_n is N
    negative(en(i));
    nue2=ans;
    negative(cen(i));
    nuce2=ans;
    if nue2<=nuce2
        nu2=nue2;
    else
        nu2=nuce2;
    end
%-----RULE 3 -----
%if en(i) is P and cen(i) is Z then delU_n is P
    positive(en(i));
    nue3=ans;
    zero(cen(i));
    nuce3=ans;
    if nue3<=nuce3
        nu3=nue3;
    else
        nu3=nuce3;
    end
%-----RULE 4 -----
%if en(i) is N and cen(i) is Z then delU_n is N
    negative(en(i));
    nue4=ans;
    zero(cen(i));

```

```

nuce4=ans;
  if nue4<=nuce4
    nu4=nue4;
  else
    nu4=nuce4;
  end
%-----RULE 5 -----
%if en(i) is P and cen(i) is N then delU_n is Z
positive(en(i));
nue5=ans;
negative(cen(i));
nuce5=ans;
  if nue5<=nuce5
    nu5=nue5;
  else
    nu5=nuce5;
  end
%-----RULE 6 -----
%if en(i) is N and cen(i) is P then delU_n is Z
negative(en(i));
nue6=ans;
positive(cen(i));
nuce6=ans;
  if nue6<=nuce6
    nu6=nue6;
  else
    nu6=nuce6;
  end
%-----RULE 7 -----
%if en(i) is Z and cen(i) is Z then delU_n is Z
zero(en(i));
nue7=ans;
zero(cen(i));
nuce7=ans;
  if nue7<=nuce7
    nu7=nue7;
  else
    nu7=nuce7;
  end
%-----RULE 8 -----
%if en(i) is Z and cen(i) is N then delU_n is N
zero(en(i));
nue8=ans;
negative(cen(i));
nuce8=ans;
  if nue8<=nuce8
    nu8=nue8;
  else
    nu8=nuce8;
  end
%-----RULE 9 -----
%if en(i) is Z and cen(i) is P then delU_n is P
zero(en(i));
nue9=ans;
positive(cen(i));
nuce9=ans;

```

```

        if nue9<=nuce9
            nu9=nue9;
        else
            nu9=nuce9;
        end
dR1=positive2(nu1);
dR2=negative2(nu2);
dR3=positive2(nu3);
dR4=negative2(nu4);
dR5=zero2(nu5);
dR6=zero2(nu6);
dR7=zero2(nu7);
dR8=negative2(nu8);
dR9=positive2(nu9);
d=(nu1*dR1+nu2*dR2+nu3*dR3+nu4*dR4+nu5*dR5+nu6*dR6+nu7*dR7+nu8*dR8+nu9*dR9)/(nu1+nu2+nu3+nu4+nu5+nu6+nu7+nu8+nu9);
delta_U(i)=d*de;
if i==1 U(i)=d*de;
else
U(i)=U(i-1)+T*d*de;
end
uu=U(i);
%-----Second Order System-----
%-----
if i==1
    y(i)=0;
end
if i==2
    y(i)=0.004802*U(i)
end
if i>2
    y(i)=1.878*y(i-1)-0.8869*y(i-2)+0.004802*U(i-1)+0.004614*U(i-2);
end
%-----
e(i+1)=yd-y(i);
ce(i+1)=e(i+1)-e(i);
end
subplot(2,1,2); plot(e);ylabel('error(t)');xlabel('t[sec/10]');grid on;
hold all
subplot(2,1,1); plot(y);ylabel('y(t)');xlabel('t[sec/10]');grid on;
hold all
plot(yd_graph)

```

APPENDIX B

MICROCONTROLLER PROGRAMS

B-1 PIC-C Codes for the Automatic Ignition and Cooling Program

```
/*
***ADC***
    channel 4: Thermocouple
    channel 5: Thrust Input

***PIN F***
    F5: Enable A of the L298
    F0: Enable B of the L298

***PIN A***
    A_11: Electric starter

***PIN B***
    B_9: start Button
    B_10: auto stop button
    B_11: kerosene Valve
    B_12: Propane Valve

***PIN C***
    C_13: Spark Plug

*/
#include <30F4013.h>
#define device adc=10
#define use delay(clock=10000000)
#define FUSES HS
#define use
rs232(UART2,baud=9600,xmit=PIN_F3,rcv=PIN_F2,parity=N,bits=8)
//#use fast_io(B)
//#use fast_io(C)
//#use fast_io(D)
#define use fast_io(F)
```

```

#include <flex_lcd.c>

int advall=0;int x=1;int t1=0;int t2=1;int t3=1;int y=1;
int y1=1;int pB1;int pB2;int adval2=0;int Tt9=0;
unsigned int16 timer_PRx;
unsigned int16 duty1=0;
unsigned int32 external=0;
unsigned int32 devir=0;

#int_timer4
void kesme_isr()
{
    external=get_timer1();
    devir=external*600/7;
    delay_us(20);
    set_adc_channel(5);
    delay_us(20);
    adval2=read_adc();
    Tt9=adval2*(1080/1024);
    lcd_gotoxy(1,1);
    printf lcd_putc,"devir:%8u\nTt9:%3d
%4d",devir,Tt9,duty1);
    set_timer1(0);
    set_timer4(0);
}

void main(){
timer_PRx=249;
lcd_init(); // Always call this first.
setup_adc(ADC_CLOCK_DIV_2);
setup_timer2(TMR_INTERNAL|TMR_DIV_BY_1,timer_PRx);
setup_compare(2,COMPARE_PWM | COMPARE_TIMER2);
    setup_timer4(TMR_INTERNAL|TMR_DIV_BY_256,1952);
    setup_timer1(TMR_EXTERNAL|TMR_DIV_BY_1);
    set_timer1(0);
    set_timer4(0);
    clear_interrupt(INT_TIMER4);
    enable_interrupts(INT_TIMER4);
    enable_interrupts(INTR_GLOBAL);
setup_adc_ports(sAN4|sAN7|sAN5);
set_tris_f(0x00);
while (true){
if (Tt9>=70) t1=1;//-----Cooling Program-----
while (t1==1){

        if ((Tt9>70)&&(devir<2500)){
            OUTPUT_HIGH(PIN_A11);
            t3=1;
            while (t3==1){
                if (devir>5000){
                    OUTPUT_LOW(PIN_A11);

```



```
}  
}  
}
```

B-2 PIC-C Codes for the Fuzzy Logic Controller

This code was designed but did not test on the ECU and gas turbine. Its automatic start and automatic cooling program works theoretically but it has to test on the gas turbine and its parameters have to be adjusted for effective control.

```
/*  
***ADC***  
        channel 4: Thermocouple  
        channel 5: Thrust Input  
  
***PIN F***  
        F5: Enable A of the L298  
        F0: Enable B of the L298  
  
***PIN A***  
        A_11: Electric starter  
  
***PIN B***  
        B_9: start Button  
        B_10: auto stop button  
        B_11: kerosene Valve  
        B_12: Propane Valve  
  
***PIN C***  
        C_13: Spark Plug  
  
*/  
#include <30F4013.h>  
#device adc=10  
#use delay(clock=10000000)  
#FUSES HS  
#use  
rs232(UART2,baud=9600,xmit=PIN_F3,rcv=PIN_F2,parity=N,bits=8)  
#use fast_io(F)  
#include <flex_lcd.c>  
int adval1;int adval2;int adval3;  
int x=1;int y=1;int yy=1;int yyy=1;  
int t1=0;int t2=1;int t3=1;int t4=1;int t5=0;int t6=0;int  
t7=1;int t8=0;int t9=0;  
int pB1;int pB2;  
unsigned int16 timer_PRx;  
unsigned int16 duty1;unsigned int16 duty2;  
unsigned int32 external=0;
```

```

unsigned int32 devir=0;unsigned int32 devir_d=0;unsigned int32
devir_R=0;
int Tt9=0;
//FUZZY PARAMETERS*****
float T=0.01;float n=1/60000.0;float nce=1/60000.0;
float de=0.1;
int16 i=0;float ce=0.0; float en=0.0;float cen=0.0;float
e_new=0.0;float e_old=0.0;
float nue1=0.0;float nucel=0.0;float nu1=0.0;
float nue2=0.0;float nuce2=0.0;float nu2=0.0;
float nue3=0.0;float nuce3=0.0;float nu3=0.0;
float nue4=0.0;float nuce4=0.0;float nu4=0.0;
float nue5=0.0;float nuce5=0.0;float nu5=0.0;
float nue6=0.0;float nuce6=0.0;float nu6=0.0;
float nue7=0.0;float nuce7=0.0;float nu7=0.0;
float nue8=0.0;float nuce8=0.0;float nu8=0.0;
float nue9=0.0;float nuce9=0.0;float nu9=0.0;
float
dR1=0.0,dR2=0.0,dR3=0.0,dR4=0.0,dR5=0.0,dR6=0.0,dR7=0.0,dR8=0.
0,dR9=0.0,numer=0.0,denumer=0.0,d=0.0;
float U_new=0.0;
float U_old=0.0;
//*****FUZZY SUB PROGRAMS*****
float positive (float ee){
    float nu_p;
    if (ee>=0 && ee<=0.5) nu_p=2.0*(float)ee;
    else if (ee>=0.5 && ee<=1.0) nu_p=1.0;
    else nu_p=0;
    return ((float)nu_p);
}
float negative (float ee){
float nu_n;
    if (ee<=-0.5 && ee>=-1.0) nu_n=1.0;
    else if (ee<=0.0 && ee>=-0.5) nu_n=-2.0*(float)ee;
    else nu_n=0;
    return ((float)nu_n);
}
float zero (float ee){
float nu_z;
    if (ee<=0 && ee>=-0.5) nu_z=1.0+2.0*(float)ee;
    else if (ee>=0.0 && ee<=0.5) nu_z=1.0-2.0*(float)ee;
    else nu_z=0;
    return ((float)nu_z);
}
float positive2 (float nu){
float A1,A2,d1,d2,dd;
    if (nu==0)dd=0;
    else {
        A1=(1-(float)nu/2.0)*(float)nu;
        A2=((float)nu*(float)nu)/4.0;
        d1=(float)nu/4.0+0.5;

```

```

        d2=(float)nu/3.0;

dd=((float)d1*(float)A1+(float)d2*(float)A2)/((float)A1+(float)
)A2);
    }
    return ((float)dd);
}
float negative2 (float nu){
float A1,A2,d1,d2,dd;
if (nu==0) dd=0;
    else{
        A1=(1.0-(float)nu/2.0)*(float)nu;
        A2=((float)nu*(float)nu)/4.0);
        d1=-((float)nu/4.0+0.5);
        d2=-((float)nu/3.0;

dd=((float)d1*(float)A1+(float)d2*(float)A2)/((float)A1+(float)
)A2);
    }
    return ((float)dd);
}
float zero2 (float nu){
float dd;
    dd=0;
    return ((float)dd);
}

//-----
#int_timer4
void kesme_isr()
{
    external=get_timer1();
    devir=external*600/7;
    set_adc_channel(5);
    delay_us(30);
    adval2=read_adc();
    Tt9=adval2*(1080/1024);
    lcd_gotoxy(1,1);
    printf(lcd_putc, "devir:%8u\nTt9:%3d %cC",devir,Tt9,22);
    set_timer1(0);
    set_timer4(0);
}

void main(){
timer_PRx=249;
lcd_init(); // Always call this first.
setup_adc(ADC_CLOCK_DIV_2);
setup_timer2(TMR_INTERNAL|TMR_DIV_BY_1,timer_PRx);
setup_compare(2,COMPARE_PWM | COMPARE_TIMER2);
    setup_timer4(TMR_INTERNAL|TMR_DIV_BY_256,1952);
    setup_timer1(TMR_EXTERNAL|TMR_DIV_BY_1);

```



```

        OUTPUT_HIGH(PIN_A11);
        if (devir>=9000){
            OUTPUT_HIGH(PIN_B11);
            devir_R=devir;
            while(yyy==1){
                delay_ms(T);
                if (t5==1){
                    devir_d=devir_d+((20000-devir_R)/600);
                    if (devir>=20000){
                        t5=0;
                        t6=1;
                    }
                }
                if (t6==1){
                    t7=t7+1;
                    devir_d=20000;
                    if (t7>=500){
                        t6=0;
                        t8=1;
                    }
                }
                if (t8==1){
                    devir_d=devir_d-10;
                    if (devir<=15000){
                        devir_d=15000;
                        t8=0;
                        t9=1;
                    }
                }
                if (t9==1){
                    set_adc_channel(4);
                    delay_us(30);
                    advall=read_adc();

                    devir_d=15000+(advall*30000/1024);
                }
                //--error calculation-----
                e_new=devir_d-devir;
                ce=e_new-e_old;
                en=e_new*n;
                cen=ce*nce;
                e_old=e_new;
            }
            //-----RULE 1 -----
            //if en(i) is P and cen(i) is P then delU_n is P
            nuel=positive(en);
            nucel=positive(cen);
            if (nuel<=nucel)nul=nuel;
            else nul=nucel;
            //-----RULE 2 -----

```

```

//if en(i) is N and cen(i) is N then delU_n is N
nue2=negative(en);
nuce2=negative(cen);
if (nue2<=nuce2) nu2=nue2;
else nu2=nuce2;
//-----RULE 3 -----
//if en(i) is P and cen(i) is Z then delU_n is P
nue3=positive(en);
nuce3=zero(cen);
if (nue3<=nuce3) nu3=nue3;
else nu3=nuce3;
//-----RULE 4 -----
//if en(i) is N and cen(i) is Z then delU_n is N
nue4=negative(en);
nuce4=zero(cen);
if (nue4<=nuce4) nu4=nue4;
else nu4=nuce4;
//-----RULE 5 -----
//if en(i) is P and cen(i) is N then delU_n is Z
nue5=positive(en);
nuce5=negative(cen);
if (nue5<=nuce5) nu5=nue5;
else nu5=nuce5;
//-----RULE 6 -----
//if en(i) is N and cen(i) is P then delU_n is Z
nue6=negative(en);
nuce6=positive(cen);
if (nue6<=nuce6) nu6=nue6;
else nu6=nuce6;
//-----RULE 7 -----
//if en(i) is Z and cen(i) is Z then delU_n is Z
nue7=zero(en);
nuce7=zero(cen);
if (nue7<=nuce7) nu7=nue7;
else nu7=nuce7;
//-----RULE 8 -----
//if en(i) is Z and cen(i) is N then delU_n is N
nue8=zero(en);
nuce8=negative(cen);
if (nue8<=nuce8) nu8=nue8;
else nu8=nuce8;
//-----RULE 9 -----
//if en(i) is Z and cen(i) is P then delU_n is P
nue9=zero(en);
nuce9=positive(cen);
if (nue9<=nuce9) nu9=nue9;
else nu9=nuce9;
////////////////////////////////////
dR1=positive2(nu1);
dR2=negative2(nu2);
dR3=positive2(nu3);

```

