creative commons

COMMONS DEED

저작자표시-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

[UCI]I804:24011-200000241939

2011년 8월
석사학위논문

무인기용 터보프롭 엔진의 GUI기반 온라인 상태감시 및 진단 프로그램 개발에 관한 연구

조선대학교 대학원

항 공 우 주 공 학 과

김 건 우

무인용 터보프롭 엔진의 GUI기반 온라인 상태감시 및 진단 프로그램 개발에 관한 연구

- Development of GUI based On-line Condition Monitoring and Diagnostic Program for a UAV's Turboprop Engine Using -

2011년 8월 25일

조 선 대 학 교 대 학 원

항 공 우 주 공 학 과

김 건 우

무인기용 터보프롭 엔진의 GUI기반 온라인 상태감시 및 진단 프로그램 개발에 관한 연구

지도교수 공 창 덕

이 논문을 공학 석사학위신청 논문으로 제출함.

2011년 4월

조 선 대 학 교 대 학 원

항 공 우 주 공 학 과

김 건 우

김건우의 석사학위 논문을 인준함

위원장	국방과학연구소 책임연구원	오 성 환 (인)
위 원	(주)이지가스터빈 대표	기 자 영 (인)
위 원	조선대학교 교수	공 창 덕 (인)

2011년 5 월

조 선 대 학 교 대 학 원

목 차

ABSTRACT

제 1 장 서 론 ····· 1
제 1 절 개 요 ····· 1
제 2 절 연구배경 ····· 4
1. 성능모사 연구 동향 ····· 4
2. 상태진단 연구 동향 ····· 7
제 3 절 연구내용 및 범위 ····· 11

제 2 장 이론 및 수학적 배경 ····· 12
제 1 절 성능해석 및 알고리즘 ····· 12
제 2 절 상태감시 및 진단 기법 ····· 14
1. 퍼지로직 ····· 15
2. 신경회로망 ····· 16
제 3 절 GUI(Graphic User Interface) ····· 17

제 3 장 연구대상 엔진 ····· 18
제 1 절 개 요 ····· 18
제 2 절 운영영역 및 설계점 성능 ····· 19

제 4 장 GUI기반 온라인 상태감시 LabVIEW® 프로그램 ····· 21
제 1 절 개 요 ····· 21
1. 입·출력 인터페이스 ····· 21
가. 프로그램 입력 ····· 22
나. 프로그램 출력 ····· 22
제 2 절 온라인 상태감시 프로그램 ····· 23
1. 기준 엔진 성능모델 ····· 23
가. 주요 구성품 모델링 ····· 24

(1) 대기조건 및 흡입구(Ambient & Intake) ····· 24
(2) 압축기(Compressor) ····· 25
(3) 연소기(Combustor) ····· 26
(4) 압축기 터빈(Compressor Turbine) ····· 27
(5) 동력 터빈(Power Turbine) ····· 28
(6) Subsystem ····· 28
① Gas Property ····· 28
② Component Map Search ····· 30
2. 가상 엔진계측 신호생성 모듈 모델링 ····· 31
3. 엔진 계측데이터 통신 모듈 모델링 ····· 32
4. 온라인 상태감시 프로그램 검증 ····· 34
가. 기준엔진 성능해석 ····· 34
(1) 고도 변화에 따른 성능해석 ····· 34
(2) 비행 마하수에 따른 성능해석 ····· 38
나. 가상 엔진 계측데이터를 이용한 상태감시 ····· 41
다. 실제 엔진 계측데이터 통신을 이용한 상태감시 ····· 43

제 5 장 GUI기반의 상태진단 LabVIEW® 프로그램 ····· 47
제 1 절 개 요 ····· 47
1. 퍼지-로직을 이용한 손상 구성품 식별 기법 ····· 48
2. 구성품 손상의 정량적 진단을 위한 신경회로망 설계 및 훈련 ····· 50
제 2 절 상태진단 프로그램 구성 및 검증 ····· 52

제 6 장 결 론 ····· 59

참 고 문 헌 ····· 60

부 록 ····· 64

LIST OF FIGURES

Fig. 1 GASTURB, gas turbine performance diagnosis program ····· 4

Fig. 2 Using GSP for gasturbine engine performance analysis and diagnosis program example ·· 5

Fig. 3 Main/starting window, FRAME ····· 6

Fig. 4 Neuro-fuzzy based FDI scheme ····· 8

Fig. 5 Major elements of NPSS ····· 9

Fig. 6 GE90-94B:Coupling of 3-D full engine model with 0-D cycle model ····· 9

Fig. 7 Flowchart of base performance simulation program ····· 13

Fig. 8 Schematic diagnostic flow of the proposed GUI type on-line diagnostic program · 14

Fig. 9 Three stage of Fuzzy system ····· 15

Fig. 10 Neural network study method ····· 16

Fig. 11-1 Layout of the study engine ····· 18

Fig. 11-2 Station No. of the study engine ····· 19

Fig. 12 Condition monitoring program configuration ····· 22

Fig. 13-1 GUI based front panel of base performance simulation program ····· 23

Fig. 13-2 GUI based block diagrams of base performance simulation program ····· 24

Fig. 14-1(a) Ambient module ····· 24

Fig. 14-1(b) Intake module ····· 25

Fig. 14-2 Compressor module ····· 26

Fig. 14-3 Combustor module ····· 27

Fig. 14-4 Compressor turbine module ····· 27

Fig. 14-5 Power turbine module ····· 28

Fig. 14-6 Gas property subsystem module ····· 30

Fig. 14-7 2D Look-up table component map search subsystem module ····· 30

Fig. 15 Signal generation module for virtual measuring performance data ····· 31

Fig. 16-1 Engine data acquisition system(a) ····· 32

Fig. 16-2 Engine data acquisition system(b) using Ethernet communication ····· 32

Fig. 17-1 Ethernet communication for TCP/IP module ····· 33

Fig. 17-2 Call the data and made text file module ····· 33

Fig. 18-1 Performance with Altitude variation at Flight Mach No. = 0.0 ····· 34~35

Fig. 18-2 Performance with Flight Mach No. variation at Altitude = 0.0 ····· 38

Fig. 19-1 Monitoring results of parameter found at sea level static standard atmospheric condition ····· 41

Fig. 19-2 Monitoring results of parameter found at sea level static standard Mach No. 0.15 standard, atmospheric condition ····· 42

Fig. 19-3 Engine performance condition monitoring results displayed on front panel ····· 42

Fig. 20-1 Engine ground performance test data during about 33minutes ····· 43

Fig. 20-2 Test data sampling during about 3minutes ····· 44

Fig. 20-3 2minutes test data for validation of the proposed on-line performance monitoring system ····· 44

Fig. 20-4 Comparison results between measured engine performance data('Measured") and simulated engine performance data('Simulation') at various Rating Code ····· 45

Fig. 20-5 Comparison results between measured engine performance data('Measured") and simulated engine performance data('Simulation') at various %rpm ····· 45

Fig. 21 Model-based multi fault diagnosis algorithm ····· 47

Fig. 22 MAMDANI type Fuzzy Inference System for isolatin faulted components ····· 49

Fig. 23-1 Feed forward Neural Network program using MATLAB/SIMULINK ····· 50

Fig. 23-2 Back propagation algorithms for training Neural Network ····· 51

Fig. 23-3 Fuzzy-Neural Network fault diagnosis program ····· 51

Fig. 24-1 MATLAB GUIDE layout editor ····· 52

Fig. 24-2 MATLAB GUIDE layout editor by m-file ····· 52

Fig. 24-3 Lay out of diagnostics monitoring ····· 53

Fig. 24-4 Results of faulted components quantified by diagnostic program ····· 53

Fig. 25-1 Results of faulted components quantified by GUI based diagnostic program(CASE 1 : single fault) ····· 55

Fig. 25-2 Results of faulted components quantified by GUI based diagnostic program(CASE 2 : single fault) ····· 56

Fig. 25-3 Results of faulted components quantified by GUI based diagnostic program(CASE 3 : single fault) ····· 56

Fig. 25-4 Results of faulted components quantified by GUI based diagnostic program(CASE 4 : double fault) ····· 57

Fig. 25-5 Results of faulted components quantified by GUI based diagnostic program(CASE 5 : double fault) ····· 57

Fig. 25-6 Results of faulted components quantified by GUI based diagnostic program(CASE 6 : double fault) ····· 58

Fig. 25-7 Results of faulted components quantified by GUI based diagnostic program(CASE 7 : triple fault) ····· 58

LIST OF TABLES

Table 1 Propulsion system damage rate and Down-Time ····· 2

Table 2 Multimedia GUI components ····· 17

Table 3-1 Operating range of PT6A-67 ····· 19

Table 3-2 Design point performance of PT6A-67 ····· 19

Table 4-1(a) Steady-stage performance at Mach No. = 0.0, Alt = 5000ft(1524m) ····· 35

Table 4-1(b) Steady-stage performance at Mach No. = 0.0, Alt = 25000ft(7620m) ····· 36

Table 4-1(c) Steady-stage performance at Mach No. = 0.0, Alt = 35000ft(10668m) ····· 36

Table 4-1(d) Steady-stage performance at Mach No. = 0.0, Alt = 40000ft(12192m) ····· 37

Table 4-1(e) Steady-stage performance at Mach No. = 0.0, Alt = 45000ft(13716m) ····· 37

Table 4-2 Uninstallation performance Flight Mach No. ····· 39

Table 5-1 Considered component fault patterns ····· 48

Table 5-2 Measured parameter change (MPC) trends depending on component fault patterns ····· 49

Table 6 Implanted fault values (IFV) of engine major components ····· 54

Table 7 Measured parameter changes due to implanted faults (%) ····· 54

Table 8 Results of faulted components isolated by Fuzzy Inference System (IFC: Input fault cases, OFC: Output fault cases) ····· 55

NOMENCLATURE

ALT	Normal pressure
COMA	Compressure Mass Flow Rate [kg/s
COEF	Compressure Isentropic Efficiency
DEL_T	Delta Temperature [K]
EHM	Engine Health Monitoring
EGT	Exhaust Gas Temperature [K]
FP	Fault Pattern
GPA	Gas Path Analysis
GSP	Gas Turbine Simulation Program
HTMA	High Pressure Turbine Mass Flow Rate [kg/s]
HTEF	High Pressure Turbine Isentropic Efficiency
ISA	International Standard Atmosphere
Mach No.	Flight Mach Number
PLA	Power Lever Angle
PTMA	Power Turbine Mass Flow Rate [kg/s]
PTEF	Power Turbine Isentropic Efficiency
S.F.C	Specific Fuel Consumption [kg/kW · hr]
S/L	Sea Level
T	Total Temperature [K]
TRQ	Torque [N · m]
ITT	Inter Turbine Temperature [K]
WFT	Fuel Flow Rate [Kg/s]

Subscripts

1	Swirl frame Inlet
2	Compressor Inlet
3	Compressor Exit

4	High Pressure Turbine Inlet
5	Power Turbine Inlet
6	Power Turbine Exit
7	Exhaust

ABSTRACT

Development of GUI based On-line Condition and Health Monitoring Program for a Turboprop Engine Using LabVIEW

by Kim, Keon-woo

Advisor : Prof. Kong, Changduk, Ph. D.

Department of Aerospace Engineering,

Graduate School of Chosun University

Recently, an aero gas turbine health monitoring system has been developed for precaution and maintenance action against faults or performance degradations of the advanced propulsion system which occurs in severe environments such as high altitude, foreign object damage particles, hot and heavy rain and snowy atmospheric conditions. However to establish this health monitoring system, the on-line condition monitoring program is firstly required, and the program must monitor the engine performance trend through comparison between measured engine performance data and base performance results calculated by base engine performance model.

This work aims to develop a GUI type on-line condition monitoring program for the PT6A-67 turboprop engine of a high altitude and long endurance operation UAV using LabVIEW. The base engine performance of the on-line condition monitoring program is simulated using component maps inversely generated from the limited performance deck data provided by engine manufacturer. The base engine performance simulation program is evaluated because analysis results by this program agree well with the performance deck data. The proposed on-line condition program can monitor the real engine performance as well as the trend

through precise comparison between clean engine performance results calculated by the base performance simulation program and measured engine performance signals. In the development phase of this monitoring system, a signal generation module is proposed to evaluate the proposed on-line monitoring system. For user friendly purpose, all monitoring program are coded by LabVIEW, and monitoring examples are demonstrated using the proposed GUI type on-condition monitoring program.

제 1 장 서 론

제 1 절 개 요

가스터빈 엔진은 대기의 공기를 흡입, 압축한 후 연료를 주입하여 연소시켜서 고온고압의 가스를 생성하는 가스발생기를 이용하며 노즐을 통하여 분사하여 제트 추진력을 얻거나 터빈을 통해 기계식 동력을 얻는다.

가스터빈 엔진의 구성요소는 모두 회전운동을 하며 진동이 적고 고속운전이 가능하다. 또한 가속성이나 감속성이 우수하며 마찰부분이 적어 윤활유와 냉각수의 소비량이 극히 적다. 외형은 비행체에 알맞은 원통형으로 구성되어 있으며 대기 온도가 낮아짐에 따라 출력이 높아지는 장점을 가지고 있어 개발 초기부터 항공기의 주 추진기관으로 그 위치를 확고히 하였다. 최근에는 항공기 추진기관 뿐만 아니라 산업용 및 기타 수송 시스템의 동력원으로서 중추적인 역할을 담당하며 기계, 발전 분야까지 그 영역이 확대되고, 기계적 요소뿐만 아니라 전기, 전자 및 재료 등 다양한 분야에서의 기술이 집약된 시스템으로 발전되어 지고 있다.

따라서 가스터빈 엔진의 기술적 발전으로 항공용뿐만 아니라 산업용 및 기타 수송 시스템에 의한 엔진 수요가 점차 증가되는 추세에 따라 엔진의 성능유지 및 보수에 비용 절감과 효율적 운용을 위하여 엔진의 성능모사와 운용상태를 감시하고 진단하는 연구가 선진국을 중심으로 활발히 진행되고 있다.

터보프롭 엔진은 동력터빈으로 프로펠러를 회전하여 기계식 동력을 얻으며 배기노즐을 부착하여 동력터빈에서 나오는 가스를 분사시켜 제트 추진력을 얻는 가스터빈 엔진의 한 종류이다. 터보프롭 엔진은 저아음속에서 우수한 추진효율을 가지며 이륙과 상승시에도 유리하며 엔진 작동이 안정적이어 단·중거리 소형 항공기의 동력원으로 많이 이용되고 있다.

국내에서 자체 개발 중인 무인항공기용 추진기관으로 Pratt-Whitney사의 PT6A-67 터보프롭 엔진을 선정하여 유지 및 보수를 위한 성능의 상태감시와 진단에 관한 연구의 필요성이 매우 높아졌다.

무인항공기용 가스터빈 엔진의 성능 유지를 위해서는 정기적인 점검과 보수와 같은 예방 정비가 필수적이다. 예방 정비방식 중 하나인 오버홀(Overhaul)방식은 정기적으로 엔진을 탈착하여 점검과 정비를 수행하나 이 방법은 결함이 언제 발생할지 모르는 불확실성과 불필요한 부분까지 탈착하여 점검 및 정비를 해야 함으로 인적, 시간적, 금전적 손실에 따라 비경제적인 방식이다. 따라서 무인기용 가스터빈

엔진은 지상에서 뿐만 아니라 비행 중에도 엔진의 상태를 감시하고 진단하여 운용시 예기치 않은 결함에 대하여 비행 후 필요한 부분만 점검 및 정비를 수행하는 상태진단방식이 이용되며 이를 위해서는 보다 정확하게 엔진의 상태를 진단하고 원인을 분석해야한다. 따라서 엔진의 상태를 감시하고 진단할 수 있는 시스템 개발과 적용이 일반화되고 있다.

Table 1은 추진시스템의 고장 원인을 분석한 것으로 엔진 자체의 주요 구성품인 압축기, 터빈 등에 손상이 발생하는 경우와 연료 시스템, 윤활 시스템과 같은 보기류의 고장, 로터 축, 베어링과 같은 기계적 부품의 마모로 손상 패턴을 분류할 수 있다[1].

Table 1. Propulsion system damage rate and Down-Time

Outage Rate	%	Outage Total Down-Time	%
Control system	24.2	Generator	17.0
General gas turbine	19.2	Turbine	14.4
Fuel system	13.1	Compressor	12.0
Lubrication	11.1	General gas turbine	12.0
Combustion	8.3	Lubrication	11.5
Generator control	6.5	Power distribution	9.6
Power distribution	5.4	Combustion	7.9
Generator	4.8	Control system	7.5
Cooling system	3.8	Fuel system	4.7
Compressor	1.9	Generator control	2.2
Turbine	1.1	Cooling system	0.9
Exhaust system	0.6	Exhaust system	0.3

통계에서 알 수 있듯이 압축기, 터빈과 같은 추진시스템의 주요구성품의 경우 손상 발생 빈도는 낮으나 이상이 발생하였을 경우 엔진 전체 성능에 미치는 영향이 치명적이며 수리 및 교체를 위한 비용과 Down-time도 다른 부품에 비해 매우 크다. 이는 시스템의 유용성을 감소시키는 주요 원인이 된다.

주요구성품의 손상에 따라 내구성이 저하되고, 효율 감소 등의 성능저하로 인한 항공기 전체의 안전성과 신뢰성을 감소시키게 된다. 로터 축, 베어링과 같은 기계적 손상은 엔진 수명과 직결되는 것으로 상태진단을 통한 적절한 관리, 운용으로

엔진의 수명 연장과 함께 경제적 손실을 최소화 할 수 있다. 연료 시스템, 윤활 시스템과 같은 보기류의 이상은 연료소모율의 증가를 가져오게 되며, 이것이 지속되었을 경우 엔진의 성능에도 영향을 미치게 된다.

이러한 추진시스템의 고장을 최소화하기 위해서는 지상정비창과 연계된 온라인 상태감시 및 진단 시스템의 구축이 요구된다. 상태진단 시스템은 압축기 및 터빈과 같은 주요구성품의 성능진단, 보기류 및 부품의 상태 감시를 통해 시스템의 고장을 최소화하고 고장 후 정비에 소요되는 시간과 비용을 최소화 할 수 있도록 함으로써 최적의 상태에서 운용될 수 있도록 하는 것이다.

특히 본 연구 대상인 무인항공기용 터보프롭 엔진은 높은 고도에서 장시간 운용되고, 무인 제어됨에 따라 요구 출력 및 운용한계의 자동제어가 요구되므로 이를 위해 정확한 엔진 모델에 대한 연구가 필요하고, 열악한 환경에서 운용되는 엔진의 신뢰성과 내구성을 확보하고 운용효율성을 높이기 위해 모델기반 상태감시 및 진단 연구가 필요하다.

본 연구에서는 무인항공기가 실제 운용 중에 지상에서 엔진의 상태를 감시하고, 진단할 수 있으며 사용자가 이해하기 쉽고 조작이 용이한 그래픽 사용자 인터페이스(Graphic User Interface, 이하 GUI) 기반의 온라인 상태감시와 진단프로그램 제안하였다. 제안된 상태감시 프로그램은 기준엔진 성능모델과 실제 엔진계측데이터를 획득할 수 있는 통신모듈로 구성되어 있다. 기준엔진 성능모델은 제작사에서 일부 제공된 성능데이터를 비교하여 검증하였고, 상태감시 검증을 위해 실제 엔진 계측데이터 획들이 어려우므로 가상의 엔진계측데이터를 생성할 수 있는 모듈을 제안하여, 상태감시를 검증하였다. 온라인 상태감시 검증을 위해 실제 엔진의 계측데이터를 송·수신할 수 있도록 이더넷(Ethernet) 통신을 이용하여 데이터를 전송받아 기준엔진 성능모델과 비교하여 계측변수의 변화량 데이터를 획득하였다.

온라인 상태감시를 통하여 획득된 계측변수 변화량 데이터를 입력 값으로 다중 손상 진단프로그램을 제안하여 프로그램을 통한 손상 진단을 수행하였다. 진단 프로그램은 퍼지-신경회로망을 이용하여 진단 후 사용자가 쉽고 편하게 결과를 확인할 수 있도록 상태감시와 마찬가지로 GUI기반의 프로그램을 제안하였다.

제 2 절 연구배경

1. 성능모사 연구 동향

가스터빈 엔진의 성능해석에 관한 연구는 1970년 초부터 위험도가 높으며 고가의 비용이 드는 엔진 시험을 대체하기 위해 성능해석에 관한 연구가 활발하게 진행되었다. Seller and Daniele(1975)은 다양한 형태의 Turbojet와 Turbofan 엔진의 정상 및 천이상태 성능해석이 가능한 DYNGEN 코드를 개발하였으며 현재에도 참고자료로 활용되고 있다[2]. MacMillan(1974)은 각 구성품별 성능해석 코드를 블록화 하여 거의 모든 종류의 가스터빈에 대한 정상상태 성능해석이 가능한 TURBOMACH 프로그램을 개발하였는데, 현재까지도 Cranfield University(U.K)에서 수정 및 보완하여 이용하고 있다[3].

Palmer and Yan(1985)은 TUTBOMACH 프로그램을 보완하여 TURBOTRANS 프로그램을 개발하였으며 주 연소기 유량, 후기연소기 유량, 바이패스 및 가스발생기 노즐 면적 제어 알고리즘이 추가되었다[4]. EI-Masri(1988)가 개발한 GASCAN 프로그램은 열역학, 열전달, 유체역학을 공식화하여 보다 세밀한 열역학적 현상을 해석할 수 있도록 하였다[5]. Schoveiri et al (1994)은 SIMULINK를 이용하여 GUI 성능해석 프로그램을 개발하였다[6][7]. Figure 1과 같이 L. Kurzke(1998)는 임의의 가스터빈 엔진의 정상 및 천이상태 성능해석을 위한 GUI 프로그램인 GASTURB를 개발하여 현재 상용프로그램을 널리 이용되고 있다[8].

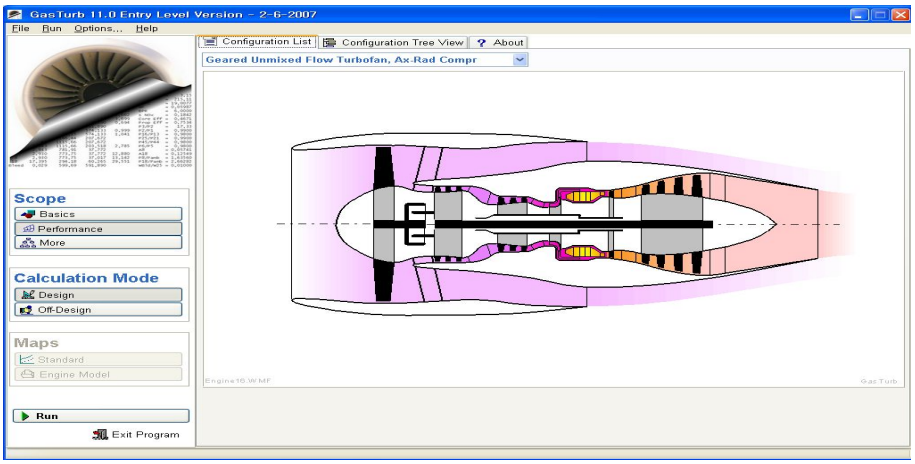


Figure 1. GASTURB, gas turbine performance diagnosis program

NLR(National Aerospace Laboratory, The Netherlands)에서는 군용 Turbojet 엔진과 Turbofan 엔진의 제어시스템 손상 효과를 해석할 수 있고, 성능데이터를 이용한 Health Monitoring이 가능한 GSP(Gas turbine Simulation Program)을 개발하여 상용화 하였으며, Read and Abdollah(1999)는 Java를 기반으로 한 ONYX system을 개발하였다. 이 프로그램은 Object-oriented 시스템으로 Web 상에서 구동이 가능하다.

AFFDL (Air Force Flight Dynamics Laboratory)의 기술보고서에서는 장착에 의한 흡입구와 배기노즐의 성능데이터를 포함한 성능해석 프로그램 개발에 관한 내용을 포함하고 있다. 또한 Cranfield의 "Propulsion System Performance and Integration" Short Course 자료에는 장착에 따른 흡입구과 배기노즐의 성능에 대해 제시되어 있으며, ESDU 기술보고서는 장착에 의한 손실, 비행속도 및 대기조건의 변화가 발생시키는 성능손실에 대해 분석하고 있다.

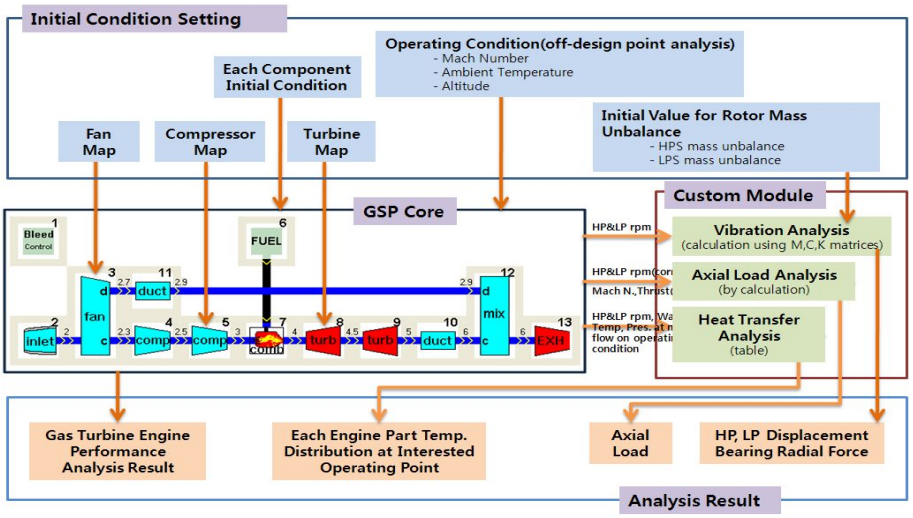


Figure 2. Using GSP for gasturbine engine performance analysis and diagnosis program example

Baxevanidis(2001)는 고고도에서 장시간 운용되는 무인항공기의 성능에 대한 연구를 수행하였으며, Maggio et al(2002)등은 추진시스템의 신뢰성 평가에 대한 연구 결과를 발표하였다. 또한, Milanes et al(2004) 등은 연소기의 내구성을 강화하기 위한 방안에 대해 연구를 수행하고, Veer et al(2004)등은 가스터빈 엔진의 신뢰

성 및 유용성 평가와 함께 상태를 예측할 수 있는 FRAME을 개발한 바 있다.

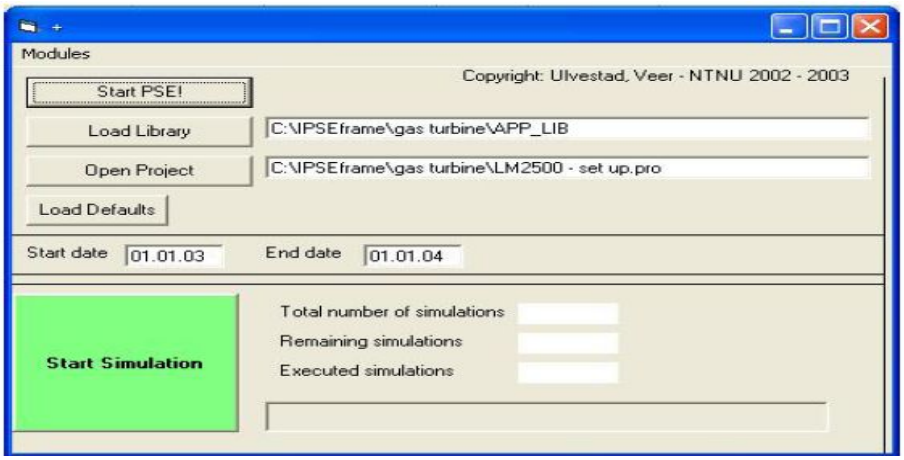


Figure 3. Main/starting window, FRAME

국내에서도 1990년대에 들어 성능모사에 대한 중요성을 인식하고 그에 대한 연구가 진행되고 있으나 선진국에 비해 아직 그 기술력이 미흡한 실정이다. 배학성(1995), 김석균(1997)[9], 공창덕(1999) 등은 BR715-56 터보팬 엔진의 성능해석을 역시 DYNGEN프로그램을 이용하여 수행한 바 있다[10]. 그러나 이러한 연구는 모두 상용 프로그램을 이용한 것으로 GUI형 성능해석 프로그램 개발에 관한 연구로는 미흡하였다. 2001년에는 보다 운용범위가 넓은 항공기용 가스터빈엔진의 성능해석 프로그램의 개발에 관한 연구를 통해 국내에서 개발한 초등훈련기 추진기관인 PT6A-62 터보프롭 엔진의 정상상태 및 천이상태 해석 프로그램이 개발되었다[11]. 동일 엔진을 대상으로 SIMULINK를 이용한 성능해석 프로그램이 개발되었으며[12], 최근 공창덕(2009) 등은 터보축 엔진의 흡입구 손실을 고려한 장착성능해석과 SIMULINK를 이용한 성능해석 연구가 진행된 바 있다[13][14].

또한 MATLAB과 미국 보잉사에서 개발한 범용 공학용 소프트웨어인 EASY5를 이용한 GUI방식의 프로그램에 관한 연구도 수행된 바 있다[15].

2. 상태진단 연구 동향

가스터빈엔진의 신뢰성을 향상시키기 위해 운용 상태를 실시간으로 감시하고 진단할 수 있는 시스템에 관한 연구와 개발이 연구기관 및 관련 산업체를 중심으로 이루어지고 있으며 최근 상용화된 시스템들이 운용 중에 있다.

모델 기반 온라인 상태감시에 대한 연구는 1974년 Urban이 공기와 연소가스가 통과하는 구성품의 효율 감소와 질량유량의 변화량으로 엔진성능을 예측하는 방법인 선형 GPA 이론을 제시하며 본격화 되었다[16]. 이 방법은 고장 시 엔진 전체의 성능과 운용에 치명적인 영향을 미치는 주요 구성품 (압축기, 연소기, 터빈 등)의 상태를 정량적으로 분석하여 주요구성품의 상태를 진단하고 고장의 원인을 예측한다. 그 후로 GPA 기법의 성능을 향상시키기 위한 연구들이 활발히 진행되어 Grewal(1988)은 선형 GPA 이론을 이용하여 다양한 성능저하의 원인을 분석하고 모델링 하였으며 Diakunchak(1992)은 산업용 가스터빈의 성능저하 실험을 통해 고장 원인과 엔진 상태의 관계를 연구하였다[17][18]. House(1992)는 터보축 엔진을 위한 선형 GPA 프로그램을 개발하였으며 Esher(1995)는 선형 GPA 기법의 한계를 극복하기 위한 비선형 GPA 기법을 제안하고 임의 엔진에 적용할 수 있는 비선형 GPA 기법을 이용한 객체지향 프로그램을 개발하였다[19][20][21]. MacDonald(1993)는 GE-F404 엔진의 EHM을 위한 동적거동 모사에 관한 연구를 수행하였으며 천이상태에서의 GPA(Gas Path Analysis)를 수행하고 그 결과를 제어기와 연계하는 알고리즘을 제시하였다. 성능진단에 인공지능 이론을 적용함으로써 진단 시스템의 성능을 향상시키기 위한 연구도 다수 진행되었는데 Vivian(1993)은 GPA 기법에 KBS (Knowlege Based System)를 적용한 가스터빈 엔진의 진단 전문가 시스템 개발에 관한 연구를 수행하였으며 Boucherma(1994)는 신경회로망 이론을 이용한 터보 제너레이터의 손상탐지와 진단에 관한 연구를 수행하였다. Zhou(1998)는 퍼지 논리를 이용하여 엔진 손상을 탐지하고 신경회로망 이론을 적용하여 진단과 손상 패턴을 분류하는 지능형 진단 시스템의 개발에 관한 연구 결과를 발표하였으며 Patel(1998)은 가스터빈 엔진의 모델링을 SIMULINK를 이용하여 수행하고 신경회로망을 통해 엔진 고장을 진단하는 연구를 수행하였다. Kim(1998)은 신경회로망 이론을 적용함으로써 GPA의 가스 경로 해석 방법을 이용할 때 발생하는 다수의 측정 변수 선정 문제를 해결하여 보다 적은 계측변수로 원하는 진단 결과를 얻고자 하였다[22]. DePold(1999)는 진단을 위해 신경회로망을

이용하고 엔진의 상태를 예측할 수 있는 전문가 시스템에 관한 연구를 수행하였다. Merrington(1994)은 Model-Based 기법을 이용하여 엔진 고장을 진단하였으며 Sampath(2002)는 유전자 알고리즘을 이용하여 엔진 손상을 진단하는 방법을 제안하였다[23]. Palade(2002)등은 산업용 가스터빈 엔진을 상용프로그램인 SIMULINK를 이용하여 모델링하고 뉴로-퍼지(Neuro-Fuzzy) 기법을 이용하여 손상을 탐지하고 분리하는 FDI (Fault Detection and Isolation) 시스템에 관한 연구를 수행하였다. 이 연구에서는 신경회로망의 학습 및 시스템 식별 능력과 Fuzzy 시스템의 다양한 환경에서 보다 인간의 지능에 가깝게 지식을 표현하는 기능을 접목하여 진단 시스템의 기능이 향상됨을 보여주었다.

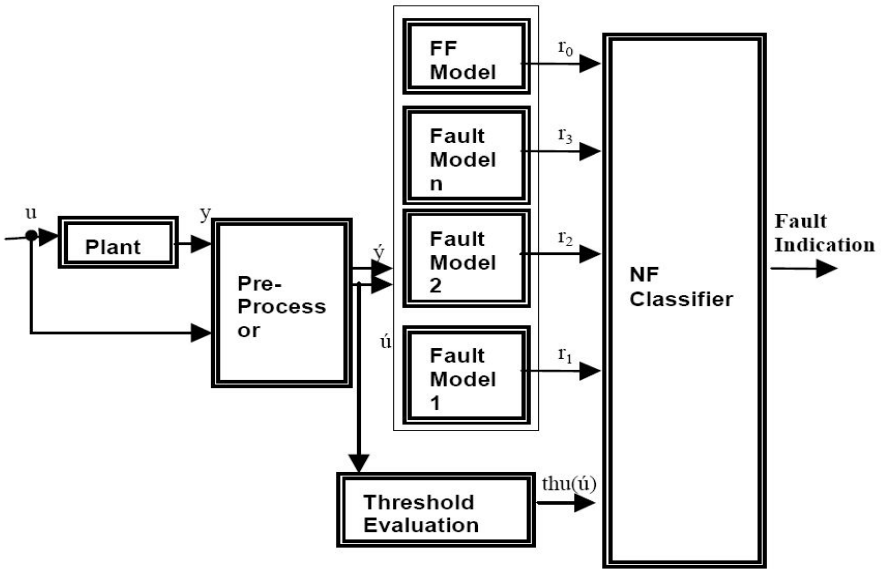


Figure 4. Neuro-fuzzy based FDI scheme

2000년 초부터 미국의 NASA Glenn Research Center를 중심으로 추진 시스템의 통합 3D (three-dimension) 해석이 가능한 NPSS (Numerical Propulsion System Simulation) 개발 프로젝트가 시작되었다. 그림 6과 같이 NPSS는 객체지향프로그래밍 (object oriented programming)을 사용하여 고 신뢰 해석 (high-fidelity analysis)과 수치 확대 기법 (numerical zooming technique)이 가능한 프로그램으로써 현재 미국 우주항공 표준 시뮬레이션으로 사용되고 있다. 이에 유럽에서도 추진

시스템의 개발 시간과 비용을 절감하기 위한 유럽 가스터빈 표준 시뮬레이션인 PROOSIS(PRopulsion Object Oriented Simulation Software) 개발 프로젝트를 시작하여 현재 수정 보완 중에 있다.

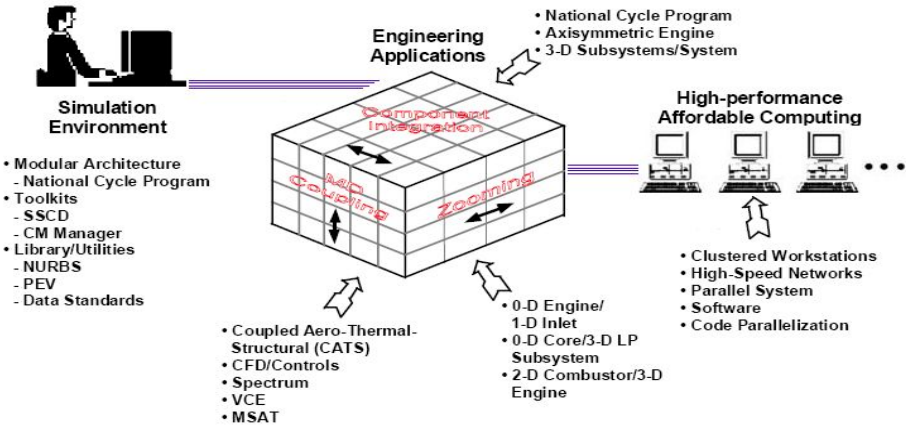


Figure 5. Major elements of NPSS

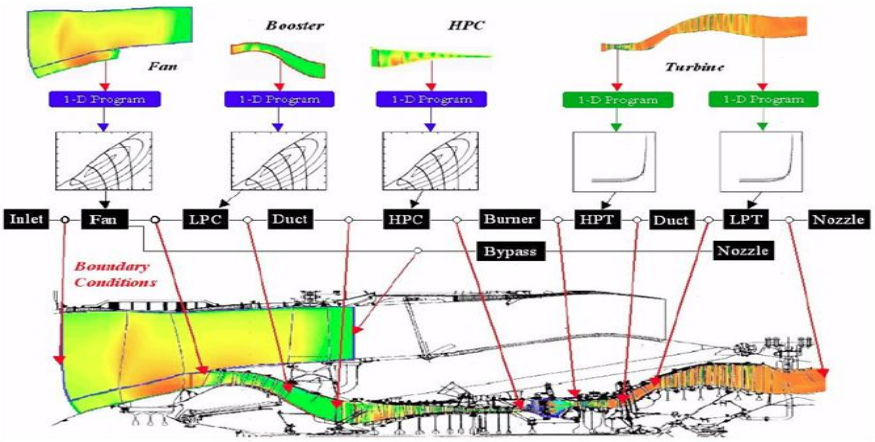


Figure 6. GE90-94B: Coupling of 3-D full engine model with 0-D cycle model

엔진의 상태를 실시간으로 감시하고 고장시 안전한 운용을 위한 제어 로직에 관한 연구도 진행되었는데, Cue(1991)등은 CF-18 항공기를 위한 엔진의 성능 진단과 고장 시 대체 로직의 가동을 비행 상태에서 실시간으로 수행하는 IECMS

(In-flight Engine Condition Monitoring System)의 개발에 관한 연구를 발표하였으며 Kerr(1992)등은 Kalman Filter 알고리즘을 이용하여 가스터빈 엔진의 손상 여부를 실시간으로 예측하는 방법을 제안하였다.

국내에서도 최근 가스터빈 엔진 상태진단 시스템에 관한 관심이 증가하며 항공기 및 가스터빈 엔진 개발 프로젝트와 함께 연구가 증가하고 있다. 1990년대에는 압축기와 터빈의 성능저하를 진단할 수 있는 GPA (Gas Path Analysis) 알고리즘에 관한 연구가 많이 진행되었다. 최근 스마트무인기, 한국형 헬리콥터개발과 관련하여 신경회로망, 퍼지 알고리즘 등 인공지능 이론을 적용한 고장진단 기술개발에 대한 연구가 이루어지고 있다. 공창덕(2004) 등은 신경회로망을 이용하여 가스터빈 엔진의 성능진단 연구를 수행하였으며[24], 2006년 공창덕 등에 의해 신경회로망을 이용한 스마트 무인기용 가스터빈엔진의 성능진단 연구가 수행된 바 있다[25].

제 3 절 연구내용 및 범위

본 논문에서는 무인항공기용 터보프롭 엔진을 위한 GUI기반 온라인 상태감시와 진단 프로그램에 관한 연구로 나누어진다.

1장에서는 연구 개요와 배경, 연구내용과 논문의 전체 구성에 대해 소개하였다. 2장에서는 본 연구주제에 사용된 알고리즘과 이론에 관한 수학적 배경과 프로그램에 관해 설명하였으며 3장에서는 연구대상 엔진에 대해 소개하였다.

4장에서는 GUI기반 온라인 상태감시를 위해 엔진성능해석을 수행하였다. 제안된 프로그램은 유량 조합의 시간을 단축시키기 위해 엔진 제작사에서 제공한 데이터를 토대로 역생성된 구성품 성능선도를 2D look-up 테이블 형으로 변환하여 기준 엔진 성능모사를 수행하였다. 실제 엔진 계측데이터를 모사히기 위해 기준 엔진 성능해석 프로그램 모듈로부터 계산된 정상상태 성능값에 가·감속 등의 천이상태 및 비행 중 외부교란에 의한 변화를 모의하기 위하여 최대 ±5%까지 무작위 신호를 생성할 수 있는 모듈을 제안하고 상태감시 여부를 확인하기 위해 다양한 조건에서의 계측데이터 변화를 살펴보았다. 또한, 온라인 상태감시를 위하여 실제 엔진의 계측데이터를 획득하기 위하여 엔진 계측데이터 통신 모듈을 구성하여 실제 엔진의 계측데이터를 획득한 후 상태감시를 수행하였다.

5장에서는 온라인 상태감시로부터 획득된 계측 데이터 변화량을 아용하여 개발된 다중손상 진단프로그램을 사용자 편의성을 제공하는 GUI기반으로 재구성하였다. GUI기반 상태진단 프로그램은 퍼지와 신경회로망 알고리즘을 사용하여 정성적 및 정량적으로 구성품의 손상을 진단하는 방법을 적용하였다.

제 2 장 이론 및 수학적 배경

제 1 절 성능해석 및 알고리즘

가스터빈 엔진은 일반적으로 극지방에서 -60℃, 열대지방에서는 50℃의 대기온도에서 작동할 수 있어야하며 지상 뿐 아니라 수천 m 고도의 높은 고도에서 운행되어 진다. 특히, 항공기용 엔진은 이보다 더 넓은 대기온도와 대기압 영역 그리고 저속에서 고속에 이르는 비행속도 영역이 작동 범위이다. 이러한 대기상태의 영향은 엔진의 설계가 확정되기 전에 예측되어야 하며 이를 위해 정상상태 성능모사 과정이 필요하다.

정상상태 성능해석은 엔진의 요구조건(추력, 비연료소모율)을 만족시키도록 사이클 해석을 통해 각 구성품의 성능을 예측하는 설계점 성능해석, 다양한 대기조건과 운용조건에 따른 탈 설계점 해석, 회전속도의 저하에 따른 부분부하에서의 성능해석으로 크게 나누어질 수 있다.

정상상태 성능해석은 흡입구에서부터 노즐까지의 가스흐름을 따라 각 구성품 들에 대한 열역학적 관계식을 이용하여 순차적으로 수행된다. 이 때 각 구성품에서의 유량은 일정해야하며 같은 로터 축으로 연결된 압축기와 터빈의 일은 같아야 한다는 적합 조건들을 만족시켜야 한다[26]. 이를 만족시키기 위해서는 다음과 같은 제반 조건을 만족하여야 한다[27].

1) 전 구성품의 유량은 일정해야 한다.
2) 압축기와 같은 로터 축으로 연결된 압축기 터빈의 일은 같아야 한다.
3) 압축기와 같은 로터 축으로 연결된 터빈의 회전수는 같아야 한다.
4) 전 구성품의 성능은 구성품 성능도를 따른다.

Figure 7은 본 연구에서 제안한 성능해석 알고리즘으로 먼저 비행조건으로부터 고도, 대기온도변화, 속도, Rating Code를 지정하여 흡입구 모델을 지나 구성품 들의 β선(작동점)을 가정한 후 압축기 성능도로부터 유량함수, 압력비. 효율을 읽는다. 열역학적 방정식을 이용하여 압축기 출구 온도와 압력, 출력이 계산한 후 연소기 모델을 지나 다시 가정된 β선의 압축기 터빈 성능도로부터 성능 값을 읽어 들여 압축기 터빈 입구온도, 압력, 연료유량, 출구 온도, 압축기 터빈 출력 등이 계산된다. 그 후 압축기 출력과 압축기 터빈 출력이 같아질 때까지 β선을 재가정하여

반복 계산한 후, 유사한 β선이 찾아지면 다시 동력터빈 성능도로부터 성능 값을 읽어 들여 동력터빈 출구 온도, 압력, 출력 등 엔진의 성능을 계산하고 마친다.

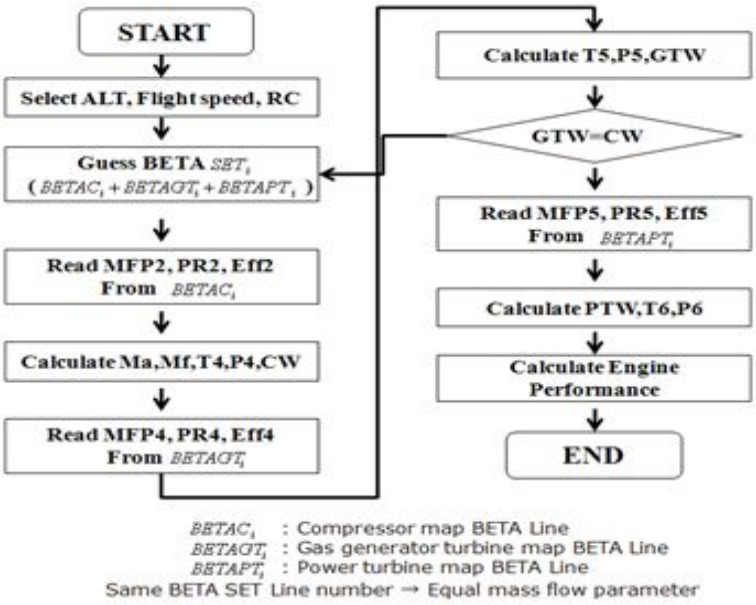


Figure 7. Flowchart of base performance simulation program

성능해석을 위해서는 엔진의 주요 구성품인 압축기, 연소기, 압축기 터빈, 동력터빈의 특성을 알아야 하며 이를 위해서는 각 구성품의 성능선도가 필요하다. 본 연구에서는 공창덕(2010) 등이 제안한 구성품 역 생성 기법[28]을 이용하고 이를 다시 2D Look-up 테이블형태로 변형하여 성능해석 프로그램에 적용하였다.

제 2 절 상태감시 및 진단 기법

엔진의 신뢰성을 향상시키기 위해서는 고장의 빈도를 낮추고 시간을 연장함으로써 비행시간 당 소요정비시간을 단축해야한다. 또한 확보해야할 부품량 감소, 고장 또는 마모로 인해 추가로 획득해야 하는 예비 추진시스템 수의 감소 등을 통한 비용 절감을 제공하고 고장 위험으로부터 비행안전을 확보할 수 있어야 한다. 즉 엔진의 신뢰도 및 내구성 향상을 위해서는 엔진 상태감시를 통한 보다 체계적인 정비 방식을 도입함으로써 엔진의 유용성과 구성품의 수명을 극대화시켜야 한다.

엔진의 상태 진단에는 주요 구성품과 보기류, 부분품의 기계적 손상을 정성적으로 진단하는 방법과 가스가 지나가는 주요 구성품의 손상을 정량적으로 진단하는 Model-based 성능진단 기법으로 나눌 수 있다. 이중 Model-Based 진단 기법은 추진 시스템 성능모델링을 사전 개발하고 엔진의 실제 계측 데이터와 시뮬레이션 결과를 비교 분석하여 이상 유무를 판단하는 방법이다.

Model-Based 상태진단 방법에는 대표적으로 일반적인 엔진 손상 분석은 엔진 구성품의 가스 경로를 실시간으로 모니터링 하는 가스경로해석 (GPA : Gas Path Analysis) 방법이 사용되고 있으나 모든 엔진손상상태를 정확히 진단하지는 못한다. 최근 들어 엔진의 헬스모니터링 시스템 성능향상을 위해 퍼지 로직, 신경망 이론, 유전알고리즘과 같은 인공지능 이론을 적용한 전문가 시스템이 이용되고 있다. 본 연구에서는 대상엔진의 기존 GPA 방식에 정성적, 정량적 구성품 손상 진단을 위해 퍼지-신경회로망을 적용한 프로그램과 실제 엔진계측데이터를 활용할 수 있는 온라인 상태감시 프로그램을 사용자 친화적인 GUI형태 제안하였다[29]. Figure 8은 제안된 온라인 상태감시 및 진단프로그램 개략도이다.

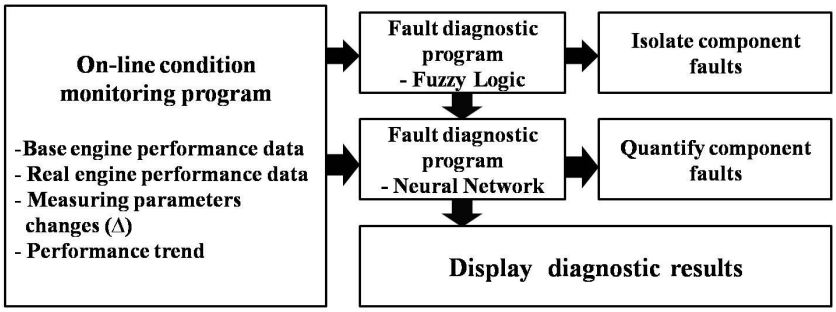


Figure 8. Schematic diagnostic flow of the proposed GUI-type on-line diagnostic program

1. 퍼지로직

L.A. Zadeh(1965)에 의해 퍼지 집합 이론이 소개된 이래로 많은 분야에서 퍼지이론이 응용되어져 오고 있다. 퍼지이론은 컴퓨터가 인공적인 지능을 가지고 인간의 의사대로 수행하기 위해 인간이 사용하는 수치와 언어적 애매한 표현들을 처리할 수 있도록 한다. 기존의 디지털(digital)논리 체계는 0과 1의 개념이 확실한 반면 퍼지 논리는 어떤 집합에 완전히 속하면 1, 완전히 속하지 않으면 0, 그 외에 0과 1 사이의 값을 가지게 되며, 인간의 애매모호한 상황도 표현할 수 있는 것이 퍼지 논리 이론이다. 퍼지이론은 퍼지는 인간의 언어적 개념을 수치화 한 이론으로 인공지능(Artificial Intelligent)라고 알려져 있다.

일반적으로 논리란 명제의 참·거짓 또는 복합 명제에서 추론되는 명제의 참·거짓을 각각의 명제에 대해서 참·거짓으로부터 명확하게 알아내는 것이다. 논리에서 다루어지는 명제는 참·거짓을 0과 1로 어느 한 값으로 나타내는 것이 많다. 이 이론을 2진 논리 또는 Boolean논리라고 한다.

인간의 사고 체계는 0과 1의 2진 논리로만으로는 인간의 애매한 사고를 표현 할 수 없다. 이러한 표현을 처리 할 수 있는 이론이 퍼지 집합(Fuzzy set Theory)이론으로 이 이론에 근거하여 참·거짓 정도를 0과 1사이의 적당한 실수 값으로 나타내며, 이 애매한 표현의 명제를 추론하여 참·거짓을 밝히는 것을 퍼지 논리(Fuzzy Logic)이라고 한다. Figure 9는 퍼지 시스템을 도식화 한 것이다[30].

본 연구에서는 계측데이터의 변화량에 따른 대상엔진 구성품의 다중손상의 정성적 분석을 위해 퍼지로직을 활용하였다.

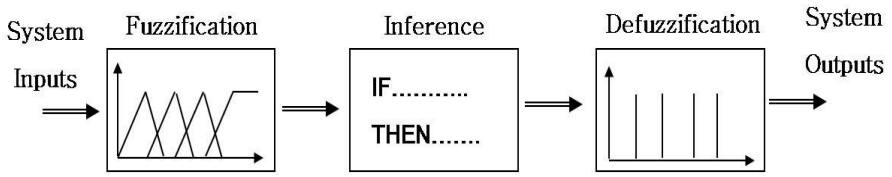


Figure 9. Three stage of Fuzzy system

2. 신경회로망

신경회로망(Neural Network)이란 단위 신경세포(neuron)이 병렬로 모인 집합체로서, 생물체의 신경계(nervous system)를 묘사한 것을 가리킨다. 신경망의 기능은 각 요소간의 연결방식으로 결정되며, 단위 신경세포간의 연결 가중치(weight)를 조절함으로서 신경회로망을 특별한 기능을 수행할 수 있도록 학습(learning)시킬 수 있다. Figure 10은 신경회로망의 학습방법을 도식화한 것이다.

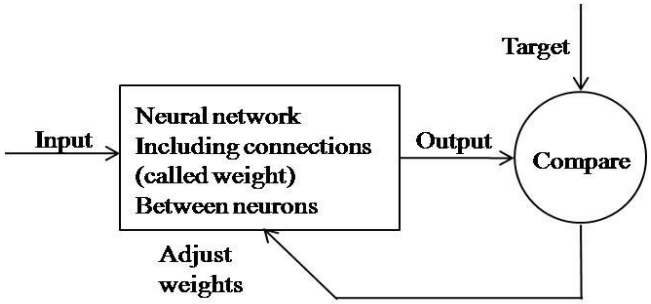


Figure 10. Neural network study method

일반적으로 신경회로망의 특징은 입력이 가해졌을 때 특정 출력이 나오도록, 신경회로망의 출력과 목표 출력을 비교하여, 위의 그림과 같이 조절 즉 학습한다. 신경회로망의 학습방법은 감독학습, 일괄학습, 순차학습이 있다. 감독학습(supervised learning) 방식은 입력과 목표 쌍의 데이터가 필요한 학습방법으로 신경회로망에 특정 입력이 가해졌을 때 특정 출력이 나오도록, 신경회로망의 출력과 목표 출력을 비교하여 학습하는 방법이다. 일괄 학습 (batch training) 방법은, 주어진 전체 입력 데이터에 대하여 가중치(weight)와 역치 (threshold, bias)를 변경하는 방식으로 진행된다. 순차학습 (incremental training)은 주어진 개별 입력 데이터 쌍을 사용하여 가중치와 역치를 차례로 학습하는 방식으로서, 온라인(on-line)또는 적응(adaptive) 학습 방식이라 부리기도 한다.

본 연구에서는 계측데이터의 변화량에 따른 대상엔진 구성품의 단일 및 다중손상의 정량적 분석을 위해 대상엔진의 구성품 손상 계측 데이터로 훈련된 신경회로망을 활용하였다[31].

제 3 절 GUI(Graphic User Interface)

원래 인터페이스(Interface)는 '이질적인 두 가지 물질이 접촉 한다'는 의미의 화학용어에서 나왔다. 서로 다른 두 시스템·장치·소프트웨어 등을 서로 이어주거나 접속 장치 등을 의미한다[32].

최근 컴퓨터와 관련된 복잡한 기능의 소프트웨어가 일반화되고 사용자들이 그 기능들을 이해하고 쉽게 사용하는 데 관점을 둔 커뮤니케이션이 이슈가 되면서, 사용자의 편리함과 정확한 의미 전달을 위해 작동하기 쉬운 순서와 표현방법을, 특히 사람과 기계간의 인지적 인터페이스 측면에서 찾아가고 있다. 예를 들면 판넬 버튼의 기능적인 정의, 개수, 배치, 형태 등을 들 수 있고, 스크린 디스플레이로 표현되는 소프트웨어의 사용 구조 및 시각적인 디자인을 총괄해서 말하기도 한다.

GUI(Graphic User Interface, 이하 GUI라 칭함)란 작업을 달성하는 데 필요한 시각정보를 그래픽적으로 사용자에게 제공하는 인터페이스를 말한다. 즉 디스플레이 그래픽을 이용한 컴퓨터 소프트웨어의 인터페이스 사용자 대화방식이다[33]. 1990년대 이후 대부분의 소프트웨어 인터페이스는 GUI의 범주에 해당할 만큼 그 중요성이 커지고 있다.

기존의 휴먼 컴퓨터 인터페이스(Human Computer Interface)는 대부분 문자로 명령어를 입력하여 컴퓨터에게 명령을 내리는 방법으로 진행되었다. 그러나 GUI는 화면에 미리 각 명령어들을 구현하거나, 명령을 내리고 작업을 수행할 수 있도록 하는 시각적인 요소를 미리 배치하여 사용자로 하여금 복잡한 절차 없이 직접 컴퓨터와 커뮤니케이션을 할 수 있도록 해준다. Table 3은 멀티미디어 GUI의 구성요소이다.

Table 2. Multimedia GUI components

윈도우	화면상에 정보가 표현되는 영역 ; 초기화면 메뉴화면, 수행 및 내용화면, 대화상자
아이콘	정보의 종류나 기증을 의미하는 상징적인 그림
메 뉴	일련의 명령어를 논리적으로 연관된 것끼리 묶어 사용자가 원하는 것을 선택할 수 있도록 보여주는 명령어 목록
포인터	작업 영역상에서 마우스나 다른 포인팅 장비의 위치를 나타내주는 그래픽 이미지
메타포	은유, 비유, 상징, 구체적인 것, 메뉴, 아이콘 등을 사용자가 보고 어떤 기능을 하는지 쉽게 알 수 있도록 암시

제 3 장 연구대상 엔진

제 1 절 개 요

Pratt-Whitney사의 PT6A-67 엔진은 분리 축 방식 터보프롭 엔진으로 감속기어를 통하여 프로펠러를 구동한다. 대상엔진의 최대출력이 1,200마력으로 엔진 감속기어부의 과응력을 방지하기 위하여 마력을 제한하였다. 이 엔진은 가스발생기부와 동력부로 구성되어 있고 가스발생기는 압축기, 연소기 및 압축기 터빈으로 구성되어 있다. 또한 가스발생기는 오일펌프, 시동 발전기 및 유압 펌프를 구동한다. Figure 11-1은 엔진의 외형 및 내구 구성도이다.

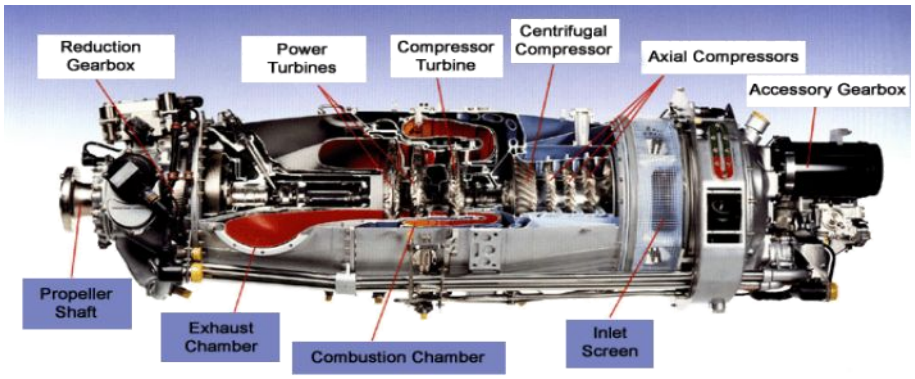


Figure 11-1. Layout of the study engine

동력터빈은 가스발생기부와 기계적으로 분리(Free Turbine Type)되어 있으며 연소기 및 압축기터빈을 거친 연소가스에 의해 구동된다. 동력터빈은 감속기어를 통하여 프로펠러를 구동시킨다. 압축기터빈과 동력터빈은 거의 엔진 중앙에 위치하며 서로 반대 방향으로 회전한다.

압축기는 4단의 축류 및 1단의 원심이 결합된 혼합형으로 구성되어 있으며. 연소기는 역환형 기화식 연소기 방식이다. 압축기 터빈은 1단이며 동력터빈은 2단 축류형으로 구성되어 있다. 대부분의 에너지는 동력터빈의 구동에 소요되며 잔여량은 엔진 양쪽의 배기구를 통하여 대기로 방출된다. Figure 11-2는 본 논문의 성능해석 및 진단에 사용된 엔진의 Station No. 이다.

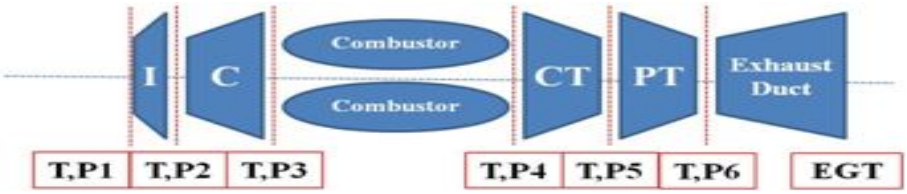


Figure 11-2. Station No. of the study engine

제 2 절 운영영역 및 설계점 성능

상태감시 및 진단에 앞서 정확한 성능해석이 요구되기 때문에 설계점 성능의 확정이 필요하다. 엔진의 설계점은 일반적으로 운용시간이 가장 긴 영역, 또는 최대 추력을 내는 영역으로 정의되고, 설계 단계나 성능해석 단계에 따라 조금씩 다르게 정의된다[34].

제작사로부터 제공된 운영영역은 Table 3-1과 같고, 설계점 성능은 다음 Table 3-2과 같다.

Table 3-1. Operating range of PT6A-67

Altitude	0 ~ 15 km
Mach No.	0 ~ 0.5
Ambient air Temperature	-30 ~ +30 ℃

Table 3-2. Design point performance of PT6A-67

	Thermodynamic Performance
Mass Flow Rate (kg/s)	4.613
Fuel Flow Rate (kg/s)	0.087643
Compressor Pressure Ratio	9.26
Compressor Exit Temp. (K)	615.33
Compressor Turbine Inlet Temp. (K)	1288.2
Shaft Horse Power (SHP)	1200
S.F.C(kg/kW·hr)	0.35259
Gas Generator Speed (RPM)	36308
Propeller Speed (RPM)	1700

설계점 성능 해석에서는 위에서 제시된 성능을 만족하도록 열역학적 관계식을 이용하여 각 구성품의 성능을 계산하였다. 하지만, 엔진 성능모사를 위해서는 성능 데이터가 부족하여 같은 엔진 계열인 PT6A-62 엔진의 지상 설계점 성능을 참고하여 시스템 식별기법을 이용하여 압축기 터빈 입구온도(T_4), 압축기 터빈 입구압력(P_4), 동력터빈 입구압력(P_4) 등을 획득하였다[35].

제 4 장 GUI기반 온라인 상태감시 LabVIEW® 프로그램

제 1 절 개 요

가스터빈 엔진의 운용에 있어서 신뢰성과 내구성, 운용비용 절감 등은 엔진 제작사와 구매자 모두에게 중요한 문제이다.

특히, 고고도의 열악한 환경에서 장시간 운용되는 무인항공기용 가스터빈 엔진의 경우 성능저하나 손상에 대한 신속한 대처를 위해 상태감시 시스템이 필수적으로 요구된다. 따라서 추진시스템의 신뢰성과 내구성 확보를 위한 모델기반의 상태감시 연구가 필요하다.

모델 기반 상태진단은 주요 구성품의 상태를 정량적으로 진단하는 기법으로 각 구성품의 유량함수와 효율의 변화량을 분석함으로써 이루어진다. 그러나 유량함수와 효율 등은 실제 운용 중에는 계측이 어렵기 때문에 온도, 압력, 회전수 등과 같은 계측 가능한 데이터 변화량을 이용하여 계산하게 된다. 따라서 추진시스템의 진단을 위해서는 성능저하가 없다고 가정된 기준엔진 성능과 계측센서로부터 얻어진 실제데이터와의 차를 분석하는 단계가 선행되어야 한다.

따라서 본 연구에서는 연구대상 엔진의 온라인 상태감시 프로그램 개발을 위하여 기준엔진 성능모델, 실제 엔진 계측데이터 획득을 위한 통신모델이 포함된 사용자 친화적인 GUI기반의 온라인 상태감시 프로그램을 제안하고 검증하였다.

본 연구에서 제안한 GUI기반의 상태감시 프로그램은 NI(National Instruments)사의 LabVIEW® 8.6버전을 이용하여 프로그램을 개발하였다. LabVIEW®의 장점은 사용자 중심의 인터페이스를 제공하며 제어 및 계측에 최적화 되고 하드웨어와의 호환성 등이 좋다는 장점을 가지고 있으며 명령어 방식이 아닌 아이콘을 사용한 코딩으로 처음 접하는 사람도 손쉽게 프로그램을 개발할 수 있다.

1. 입·출력 인터페이스

상태감시 프로그램은 제어와 계측에 최적화 되고 하드웨어와 호환성이 우수하며, GUI(Graphic User Interface)기반의 LabVIEW의 G-Programming을 이용하여 구성하였으며 프로그램의 주요 프로그램 구성항목은 다음과 같고, Figure 10은 대상엔진의 온라인 상태감시 프로그램 전체 구성도이다.

- 입·출력 인터페이스(Input & Output Interface; GUI)
- 엔진 성능모델(Base Engine Performance Model)

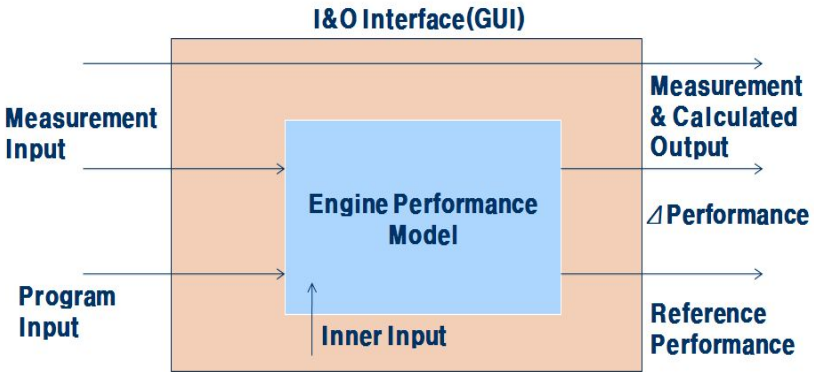


Figure 12. Condition monitoring program configuration

가. 프로그램 입력

- 성능측정변수(Measurement Parameters) : 실제 엔진에서 측정된 성능변수들로서 기준 엔진 성능계산의 입력 및 계산 값과의 차를 구하는데 사용되는 입력변수. 예) 엔진의 온도 및 압력, 회전수, 토크 또는 출력 등.
- 프로그램 입력(Program Input) : 프로그램 실행 및 엔진의 성능계산에 필요한 입력값. 예) 데이터의 입, 출력 방식, 데이터 저장 및 출력 방식 선택과 조정, 엔진 성능계산을 위한 입력, 설계점 성능 데이터 등
- 내부 입력 값(Inner Input) : 프로그램 내부 성능계산 모듈에 필요한 입력 값.

나. 프로그램 출력

- 성능측정변수(Measurement Parameters) : 엔진에서 측정된 성능변수들 중 주요 변수들을 차트나 테이블 등의 디스플레이로 출력하고 저장.
- 프로그램 출력(Program Output) : 엔진 모델의 성능계산으로부터 얻어진 출력값. ⊿Performance, Reference Performance 등
- 출력 값 저장 : 계산된 출력 값들을 외부 데이터베이스에 저장

제 2 절 온라인 상태감시 프로그램

1. 기준 엔진 성능모델

2장에서 제안된 정상상태 성능해석 알고리즘과 2D Look-up 테이블로 구성된 주요 구성품 성능도를 적용하여 Figure 13-1과 같은 기준 엔진 성능해석 프로그램을 제안하였다.

기준 엔진 성능해석 프로그램에는 비행 대기조건 계산과 흡입구 성능 계산을 위한 'AMBIENT' 모델 'INTAKE' 모델, 압축기 성능계산을 위한 'COMPRESSOR' 모델, 연소기 성능계산을 위한 'COMBUSTOR' 모델, 압축기 터빈 성능계산을 위한 'COMPRESSOR TURBINE' 모델, 동력 터빈 성능계산을 위한 'POWER TURBINE' 모델, 각 구성품의 성능선도를 찾아주는 'Map search'모델로 구성되어 있다. Figure 13-2는 제안된 기준 엔진 성능해석 프로그램의 내부 블록다이어그램이다[36][37][38][39].

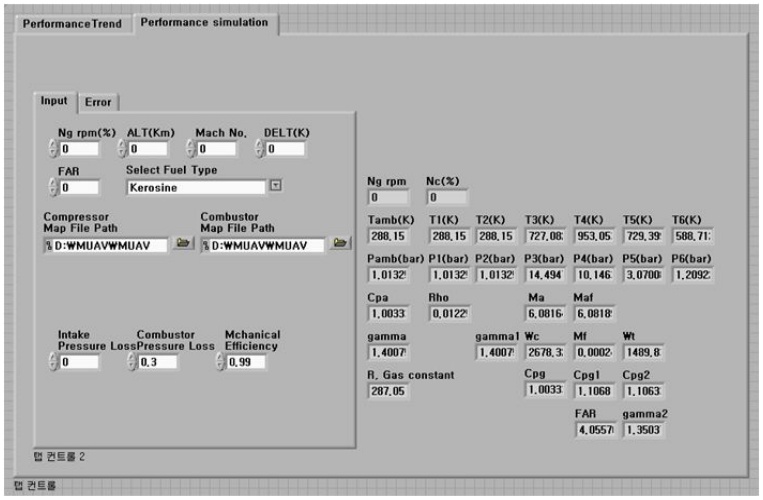


Figure 13-1. GUI based front panel of base performance simulation program

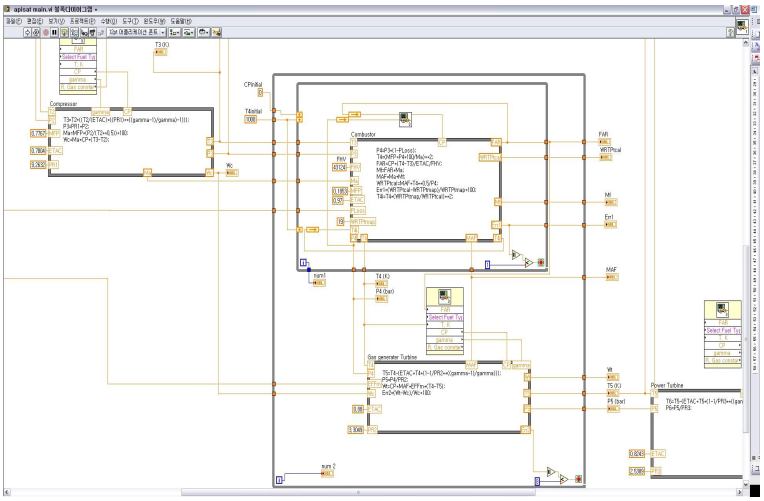


Figure 13-2. GUI based block diagrams of performance simulation programming

가. 주요 구성품 모델링

(1) Ambient & Intake

Figure 14-1(a)은 대기상태(Ambient) 모듈로 ALT와 DEL_T의 입력을 받아 수식노드를 통해 정온도, 정압력이 계산된다.

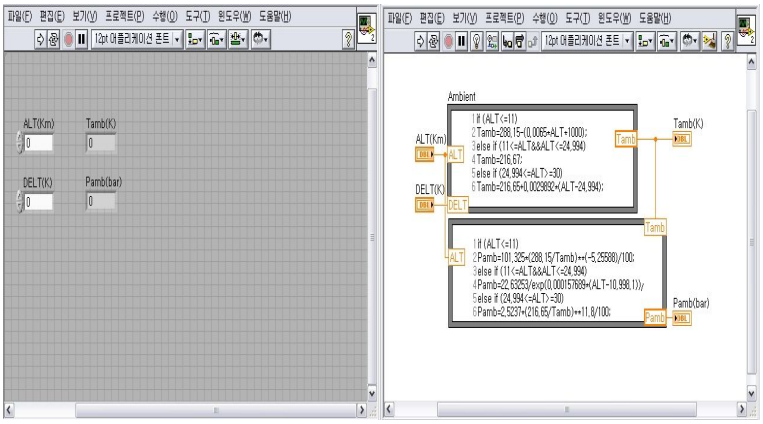


Figure 14-1(a). Ambient module

Figure 14-1(b)는 대기온도 및 대기 압력 계산 후 대상엔진의 흡입구(Intake)부분

의 성능을 계산하는 모델로 대기조건의 변화에 따라 Intake의 성능 값을 열역학적 관계식을 이용하여 계산할 수 있는 모델이다.

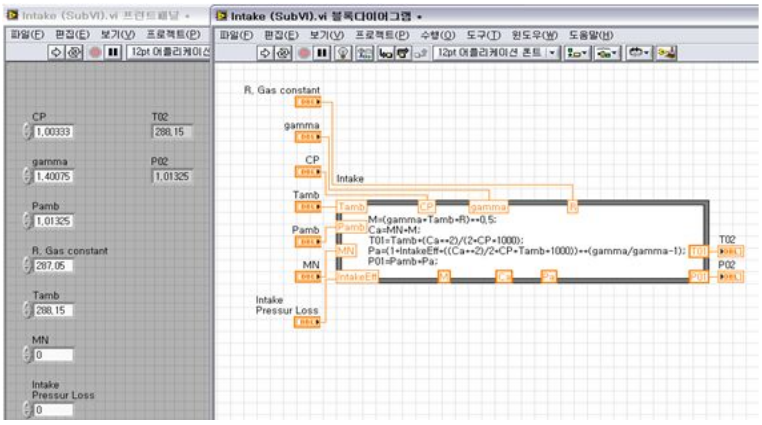


Figure 14-1(b). Intake module

(2) Compressor

'COMPRESSOR' 모델에서는 'INTAKE' 모델로부터, 압축기 입구 온도 및 압력 데이터와 압축기 성능선도를 이용하여 압축기 성능을 계산한다.

압축기 성능 계산에는 고고도 특성을 고려한 압축기 성능선도가 이용된다. 비행 상태에 따른 유량함수, 압력비, 효율을 각각 LOOK-UP 테이블화 하여 비행 상태에 따라 압축기 입구 온도(T_2)로부터 유량함수(MFP), 압력비(PR1), 효율(ETAC)을 찾는다.

성능선도 데이터로부터 열역학적 관계식을 통해 공기유량(Ma), 압축기 출구 온도(T_3) 및 압력(P_3)과 일(W_c)을 계산한다.

압축기 일(Wc)은 압축기와 압축기터빈 일 조합 여부를 판단하여 Matching시 파워터빈에 대한 계산이 가능하다.

압축기에 사용된 주요 열역학적 관계식은 다음과 같다.

$$T_3 = T_2 + (\frac{T2}{\eta_c}) \times [(\frac{P_1}{P_2})^{\frac{\gamma-1}{\gamma}} - 1] \quad (1)$$

$$W_c = Ma \times C_p \times (T_3 - T_2) \quad (2)$$

$$Ma = MFP_1 \times \frac{P2}{\sqrt{T_2}} \tag{3}$$

식 (1)을 이용하여 압축기 출구 온도와 식 (2)를 이용하여 압축기 일을 구한다. 압축기에 과한 세부모델은 Figure 12-2와 같다.

압축기 일을 구하는 순서는 우선 압축기의 유량함수 (MFP1)을 선정하고, 압축기 맵으로부터 압력비 (PR1 : $\frac{P2}{P1}$)와 효율(η_c)을 읽어 들여 각각 유량과 압축기입구 압력, 온도를 구하여 열역학적 관계식에 대입하여 구한다.

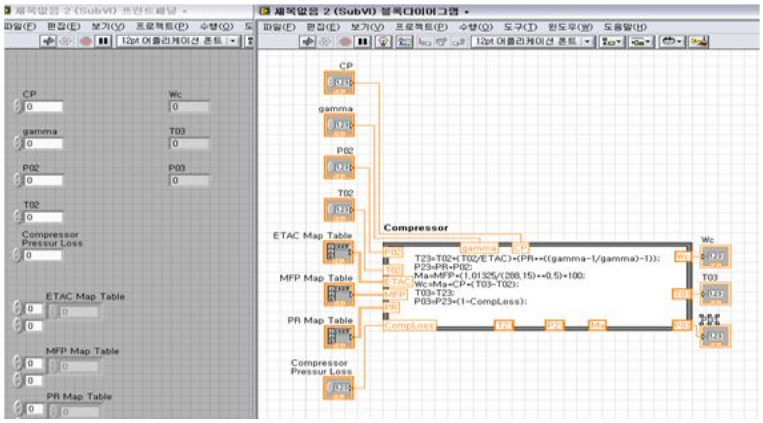


Figure 14-2. Compressor module

(3) Combustor

Figure 12-3의 'COMBUSTOR' 모델에서는 'COMPRESSOR' 모델로부터 입력된 공기유량, 압축기 출구 온도 및 압력 데이터와 압축기 터빈 성능선도 데이터로부터 연소기 성능을 계산한다.

압축기 출구 압력(P_3)과 연소기 압력손실(PLoss)값으로부터 연소기 출구 압력(P_4)를 계산한다. 압축기 터빈의 성능선도의 데이터 탐색 알고리즘은 압축기와 동일하게 프로그래밍 되었으며 압축기 출구 온도와 비행 상태로부터 압축기 터빈의 유량함수(MFP), 압력비(CTPR), 효율(ETA_CT)을 찾는다. 성능선도 데이터로부터 열역학적 관계식(T_4, FAR)을 통해 압축기터빈 입구 온도(T_4)와 연료유량(Mf)를 계산한다.

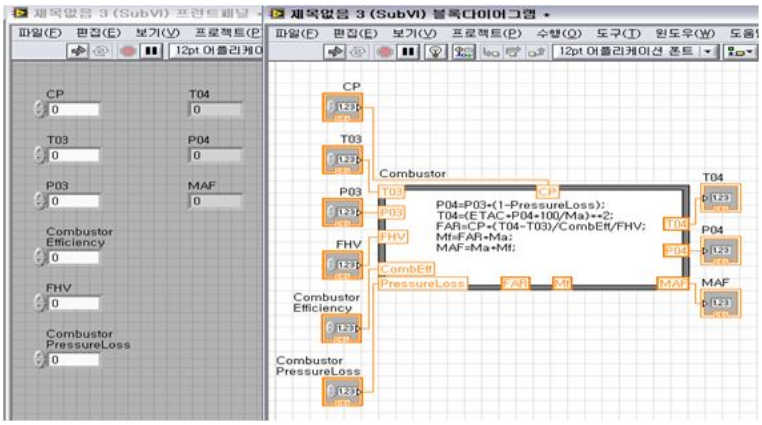


Figure 14-3. Combustor module

(4) Compressor Turbine

Figure 12-4의 'COMPRESSOR TURBINE' 모델에서는 'COMBUSTOR' 모델로부터 입력된 압축기 터빈 압력비 및 효율, 연소기 출구(압축기 터빈 입구) 온도 및 압력으로부터 압축기 터빈 성능을 계산한다.

열역학적 관계식(T_4, CTWf)으로부터 압축기 터빈 출구 온도(T_4)와 일(Wt)을 계산한다. 압축기 터빈 일(CWt)은 압축기와 압축기터빈 일 조합 여부를 판단하기 위해 Matching 될 때까지 계산된다.

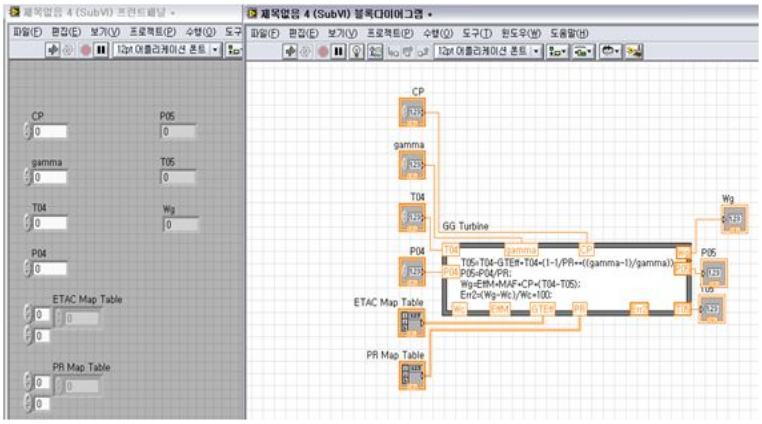


Figure 14-4. Compressor Turbine module

(5) Power Turbine

Figure 12-5의 'POWER TURBINE' 모델에서는 'COMPRESSOR TURBINE' 모델로부터 입력된 동력터빈 입구(압축기 터빈 출구) 온도 및 압력 데이터와 동력터빈 성능선도 데이터로부터 엔진 출력 및 연료소모율을 계산한다.

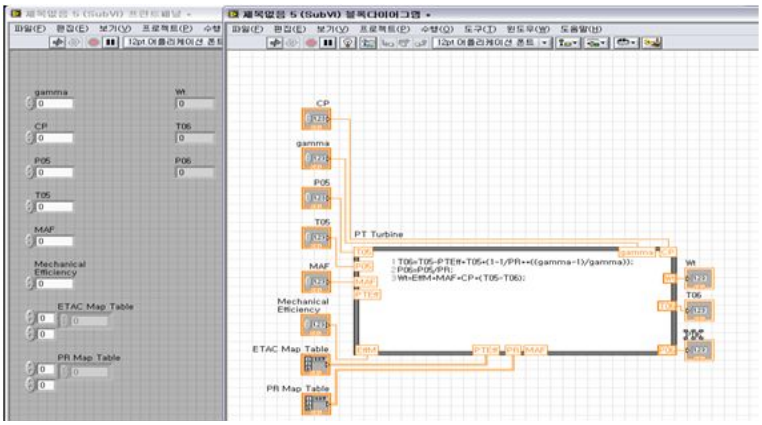


Figure 14-5. Power turbine module

동력 터빈의 성능선도의 데이터 탐색 알고리즘 또한 LOOKUP 테이블을 이용하였으며 동력터빈 비행 상태로부터 동력터빈의 유량함수(PTMA), 압력비(PTPR), 효율(PTEF)을 찾는다. 성능선도 데이터로부터 열역학적 관계식(Ti_5)을 통해 동력터빈 출구 온도(T_5)를 계산하고 동력터빈 유량함수로부터 공기유량(MA)을 계산한다.

또 다른 열역학적 관계식으로부터 동력터빈 출구 전압력(PT_5) 및 정압력(P_5)을 계산하고 정압력 값은 '엔진 출구 정압력은 대기 압력과 같아야 한다.'는 조건의 만족 여부를 판단한다.

(6) Subsystem

① Gas Property

Gas property는 비행고도에 따른 각 구성품의 온도변화에 따른 공기비열(CP)과 감마(γ)를 구하는 서브모델로 수학적 모델은 다음과 같다[40].

$$C_p = A0 + A1 \times TZ + A2 \times TZ^2 + A3 \times TZ^3 + A4 \times TZ^4 + A5 \times TZ^5 + A6 \times TZ^6 + A7 \times TZ^7 + A8 \times TZ^8 \quad (4)$$

$$TZ = T/1000 \tag{5}$$

여기서 dry air에 관한 상수 A0부터 A8은 다음과 같다.

A0=0.992313	A1=0.236688
A2=-1.852148	A3=6.083152
A4=-8.893933	A5=7.097112
A6=-3.234725	A7=0.794571
A8=-0.081873	

$$\begin{aligned} C_p = {} & A0 + A1 \times TZ + A2 \times TZ^2 + A3 \times TZ^3 + A4 \times TZ^4 \\ & + A5 \times TZ^5 + A6 \times TZ^6 + A7 \times TZ^7 + A8 \times TZ^8 \\ & + FAR/(1 + FAR) \times (B0 + B1 \times TZ + B2 \times TZ^2 \\ & + B3 \times TZ^3 + B4 \times TZ^4 + B5 \times TZ^5 + B6 \times TZ^6 \\ & + B7 \times TZ^7) \end{aligned} \tag{6}$$

여기서 상수 B0부터 B7은 다음과 같다.

B0=-0.718874	B1=8.747481
B2=-15.863157	B3=17.254096
B4=-10.233795	B5=3.081778
B6=-0.361112	B7=-0.003919

또한 FAR은 dry air의 Fuel Air Tatio로서

$$\text{FAR=DH/(LHV·ETAB) where) ETAB=0.97} \tag{7}$$

본 모델에서 쓰인 FAR식은 LHV가 43,124[kJ/kg]인 연료에 대한 식이므로

$$FAR1 = 0.10118 + 2.00376E-05 \times (700 - T_3) \tag{8}$$

$$FAR2 = 3.7078E\text{-}03 - 5.2368E\text{-}06 \times (700 - T_2) - 5.2632E\text{-}06 \times T_3 \tag{9}$$

$$FAR3 = 8.889E\text{-}09 \times |T_4 - 950| \tag{10}$$

$$FAR0 = (FAR1 - \sqrt{FAR1^2 + FAR2} - FAR^3)/\mathrm{ETAB} \qquad (11)$$

Gas property subsystem 패널과 블록다이어그램은 Figure 12-6과 같다.

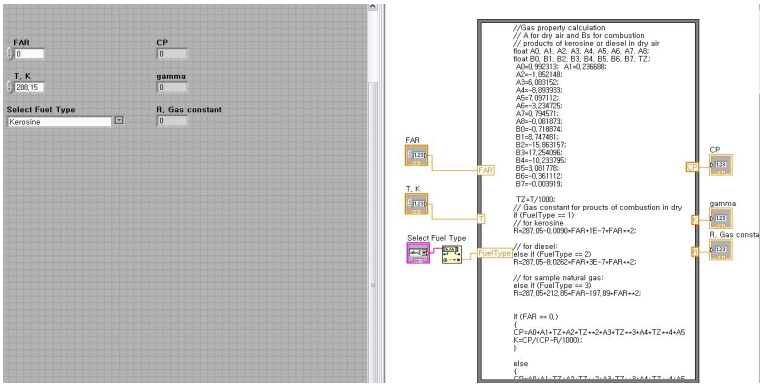


Figure 14-6. Gas property subsystem module

② Component Map Search

Figure 64는 구성품 맵 서치 서브 모델로써 기준 성능해석 프로그램의 경우 표준대기조건으로 수정된 성능선도 맵을 이용하여 계산하므로 유량 및 일 적합조건에 따른 구성품 특성 값을 반복 계산하여 구하므로 계산속도가 느리고 반복 계산 시 발생하는 오차를 줄이기 위하여 2D Look-up 테이블식으로 변경하여 전체적인 계산 속도를 향상시키고 보다 정확한 계산이 가능하도록 개선하였다.

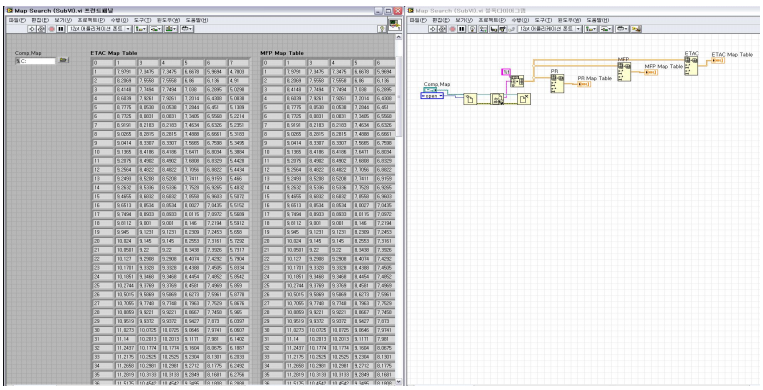


Figure 14-7. 2D Look-up table component map search subsystem module

2. 가상 엔진 계측 신호생성 모듈 모델링

본 연구에서는 실제 엔진의 데이터를 획득할 수 없을 경우를 가정하여 가상의 데이터를 만들어 실제 엔진의 상태를 감시할 수 있는 데이터를 모사하였다. 이때 사용되는 가상의 데이터는 일정 시간동안 임의의 손상을 가한(유량 감소나 효율 저하) 데이터를 사용하였다. 5가지 계측 변수는 압축기 출구 압력, 동력터빈 입구 온도 및 배기가스 온도, 연료 유량, 파워 등이다. Figure 15는 기준엔진 성능에 ±5%의 무작위 신호를 생성할 수 있는 신호발생 모듈이다.

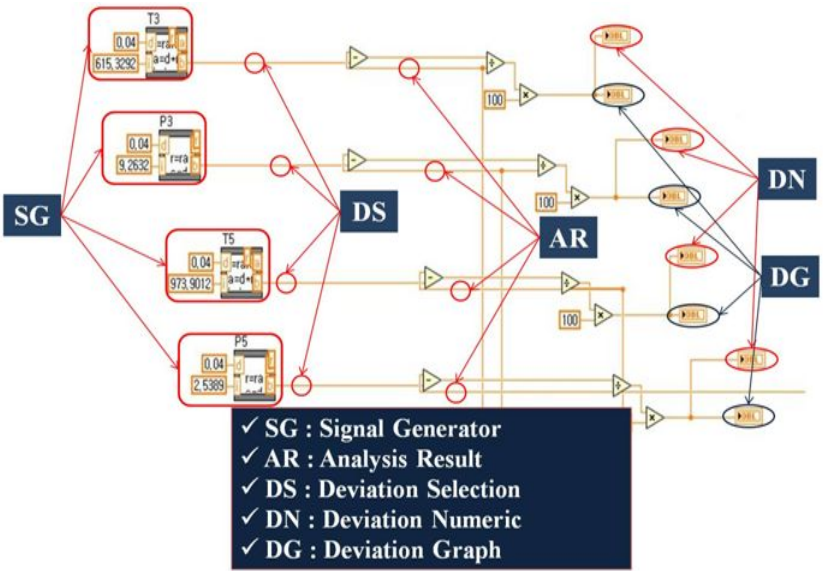


Figure 15. Signal generation module for virtual measuring performance data

3. 엔진 계측데이터 통신 모듈 모델링

본 연구에서는 실제 엔진이 가상시 측정되는 측정값을 데이터 통신을 통한 온라인 상태에서 엔진의 상태를 감시하기 위한 프로그램으로 개발되었다.

본 연구에서는 엔진 실험 데이터를 주요 데이터 저장 서버(Main Database Serve, MDS)에서 통신을 통하여 데이터를 실제 엔진 생성 모듈로 받아들인다.

다음 Figure 16-1과 16-2은 일반적인 데이터 통신을 통한 모니터링 방법으로 이 중 본 연구에서는 Figure 16-2의 개념을 이용하여 프로그램을 제안하였다[41][42].

Figure 16-1. Engine data acquisition system(a)

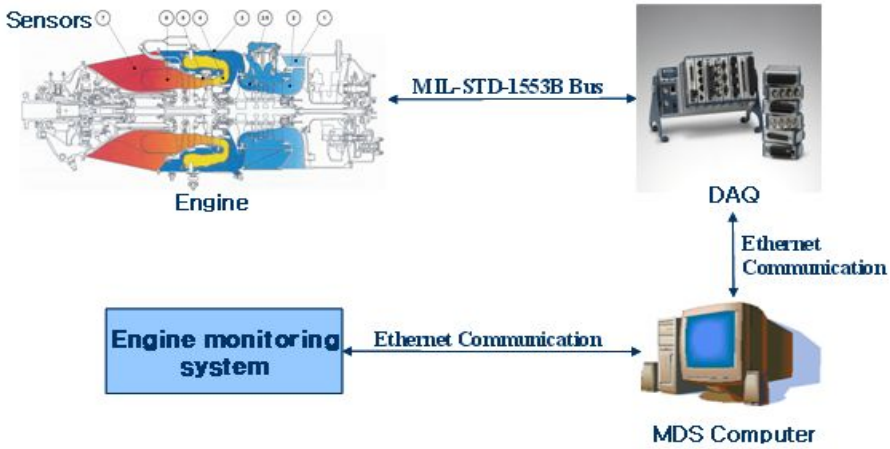


Figure 16-2 Engine data acquisition system(b) using Ethernet communication

데이터 통신 모듈은 이더넷(Ethernet)을 이용한 TCP/IP를 이용하여 프로그램을 개발하였다. MDS에 저장된 계측 데이터는 TCP/IP 통신을 통하여 저장된 데이터를 개발된 상태감시 프로그램으로 불러드린 후 계측된 데이터와 기준엔진 성능 프

로그램에서 계산된 값과 비교한다. Figure 17-1은 이더넷(Ethernet) 통신을 위한 TCP/IP 프로그램 이며, Figure 15-2은 Main data computer에서 통신을 통해 상태 감시 프로그램에서 실험 data를 불러들이는 프로그램이다[43].

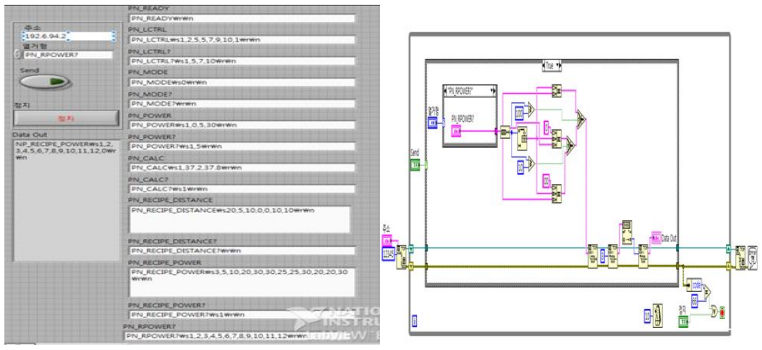


Figure 17-1. Ethernet communication for TCP/IP module

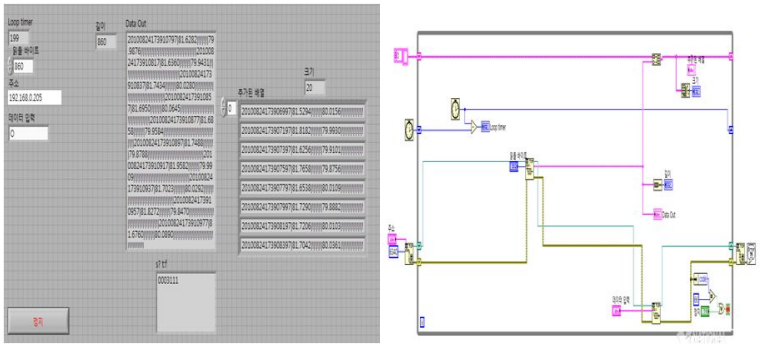


Figure 17-2. call the data and made text file module

또한 전체적인 센서 에러 등을 감안하여 비행 상태의 변화 시 추진시스템의 상태가 안정화 된 후의 정리된 데이터를 받아들인다. 기본적으로 앞서 설명하였듯이 센서에서 나오는 에러를 고려하여 MDS에서 센서 오류에 대한 보정을 통하여 데이터를 전송 받으나, 데이터 통신 시에 생기는 에러에 대한 보완은 차후 연구에서 진행할 예정이다.

4 온라인 상태감시 프로그램 검증

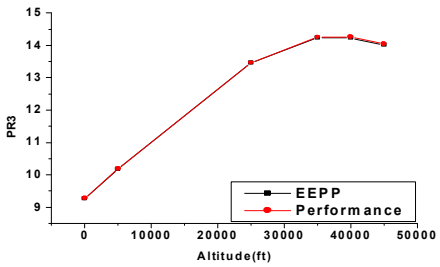
가. 기준 엔진 성능해석

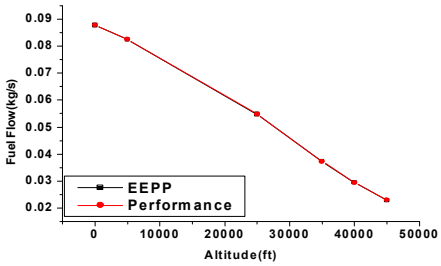
고도 및 비행마하수의 변화에 따른 성능해석을 수행하였다. 공기 흡입구 압력, 블리드 및 출력 손실이 없고, Charge Heating이 3℃일 때 가스발생기 및 동력 터빈 회전수를 100%가정하였다. 프로그램의 검증을 위해 해석 결과는 EEPP 데이터와 해석 결과를 비교하였다.

(1) 고도 변화에 따른 성능해석

가스발생기와 동력터빈의 회전수가 100%인 Rating code 1을 기준으로 비행마하수가 0인 조건에서 지상표준대기상태에서 45,000ft(13,716m)까지 변화시키며 해석을 수행하였으며 그 결과는 Figure 18-1 및 Table 4-1과 같다.

비교 결과 모든 운용영역에서 평균 1% 내외의 오차율을 보여 본 프로그램에 사용된 성능해석 기법과 알고리즘의 타당성을 검증할 수 있었다. 발생된 오차의 가장 큰 원인은 개발프로그램의 경우에는 압축기, 압축기터빈, 동력터빈의 성능도가 사용되는데 반해 EEPP 데이터의 경우에는 깨끗한 엔진성능을 설계점으로 고정하고 해석하기 때문으로 예측되며 사용된 열역학적 방정식이나 조합을 위한 알고리즘 차이도 원인일 것으로 판단된다.





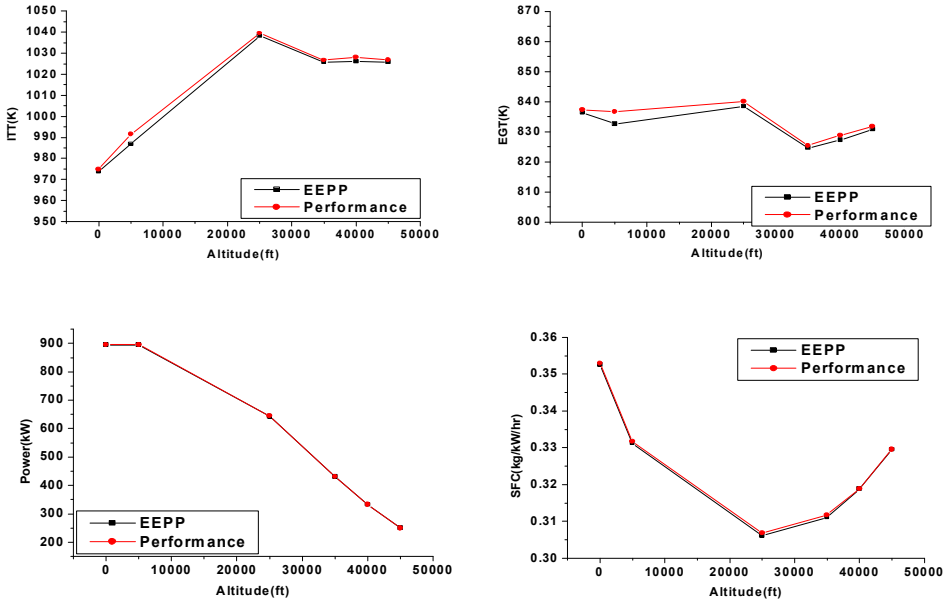


Figure 18-1. Performance with Altitude variation at Flight Mach No. = 0.0

Table 4-1(a). Steady-stage Performance at Mach = 0.0, Alt. = 5000ft (1524m)

Variable	Performance	Design Point	Error (%)
Compressor Output Pressure Ratio	10.1964	10.1851	1.0011
Inter Turbine Temperature (K°)	991.3936	986.8934	1.0046
Exhaust Gas Temperature (K°)	836.6700	832.6210	1.0049
Fuel Flow Rate (kg/sec)	0.0824	0.0823	1.0010
Shaft Horse Power (kW)	895.7214	894.8408	1.0010
S.F.C (kg/kW·hr)	0.3318	0.3312	1.0016

Table 3-1(b). Steady-stage Performance at Mach = 0.0, Alt. = 25000ft (7620m)

Variable	Performance	Design Point	Error (%)
Compressor Output Pressure Ratio	13.4662	13.4689	0.9998
Inter Turbine Temperature (K°)	1039.4577	1038.2004	1.0012
Exhaust Gas Temperature (K°)	840.0914	838.5309	1.0019
Fuel Flow Rate (kg/sec)	0.0548	0.0547	1.0017
Shaft Horse Power (kW)	644.9590	643.3759	1.0025
S.F.C (kg/kW·hr)	0.3069	0.3061	1.0025

Table 3-1(c). Steady-stage Performance at Mach = 0.0, Alt. = 35000ft (10668m)

Variable	Performance	Design Point	Error (%)
Compressor Output Pressure Ratio	14.2557	14.2397	1.0011
Inter Turbine Temperature (K°)	1026.6956	1025.6780	1.0010
Exhaust Gas Temperature (K°)	825.4604	824.5830	1.0011
Fuel Flow Rate (kg/sec)	0.0373	0.0372	1.0013
Shaft Horse Power (kW)	431.3762	430.9790	1.0009
S.F.C (kg/kW·hr)	0.3118	0.3111	1.0022

Table 3-1(d). Steady-stage Performance at Mach = 0.0, Alt. = 40000ft (12192m)

Variable	Performance	Design Point	Error (%)
Compressor Output Pressure Ratio	14.2644	14.2342	1.0021
Inter Turbine Temperature (K°)	1028.2003	1026.1834	1.0020
Exhaust Gas Temperature (K°)	828.8425	827.2990	1.0019
Fuel Flow Rate (kg/sec)	0.0295	0.0295	1.0000
Shaft Horse Power (kW)	332.7064	332.7076	1.0000
S.F.C (kg/kW·hr)	0.3189	0.3188	1.0002

Table 3-1(e). Steady-stage Performance at Mach = 0.0, Alt. = 45000ft (13716m)

Variable	Performance	Design Point	Error (%)
Compressor Output Pressure Ratio	14.0422	14.0138	1.0020
Inter Turbine Temperature (K°)	1026.8096	1025.6609	1.0011
Exhaust Gas Temperature (K°)	831.7024	830.7126	1.0012
Fuel Flow Rate (kg/sec)	0.0229	0.0229	1.0000
Shaft Horse Power (kW)	249.9885	249.9983	1.0000
S.F.C (kg/kW·hr)	0.3295	0.3295	1.0001

(2) 비행 마하수에 따른 성능해석

비행마하수가 0.0, 0.1, 0.15, 0.25, 0.35 일 때의 성능을 해석하였다. 이 때도 가스발생기와 동력터빈의 회전수는 100%로 고정된 Rating code 1을 이용하여, 지상표준대기 조건에서 해석을 수행하였다. 해석 결과는 EEPP 데이터와 해석 결과를 비교하였으며 Figure 18-2 와 Table 4-2와 같다. 비교결과는 모든 운용영역에서 1% 내의 오차율을 보임을 확인하였다.

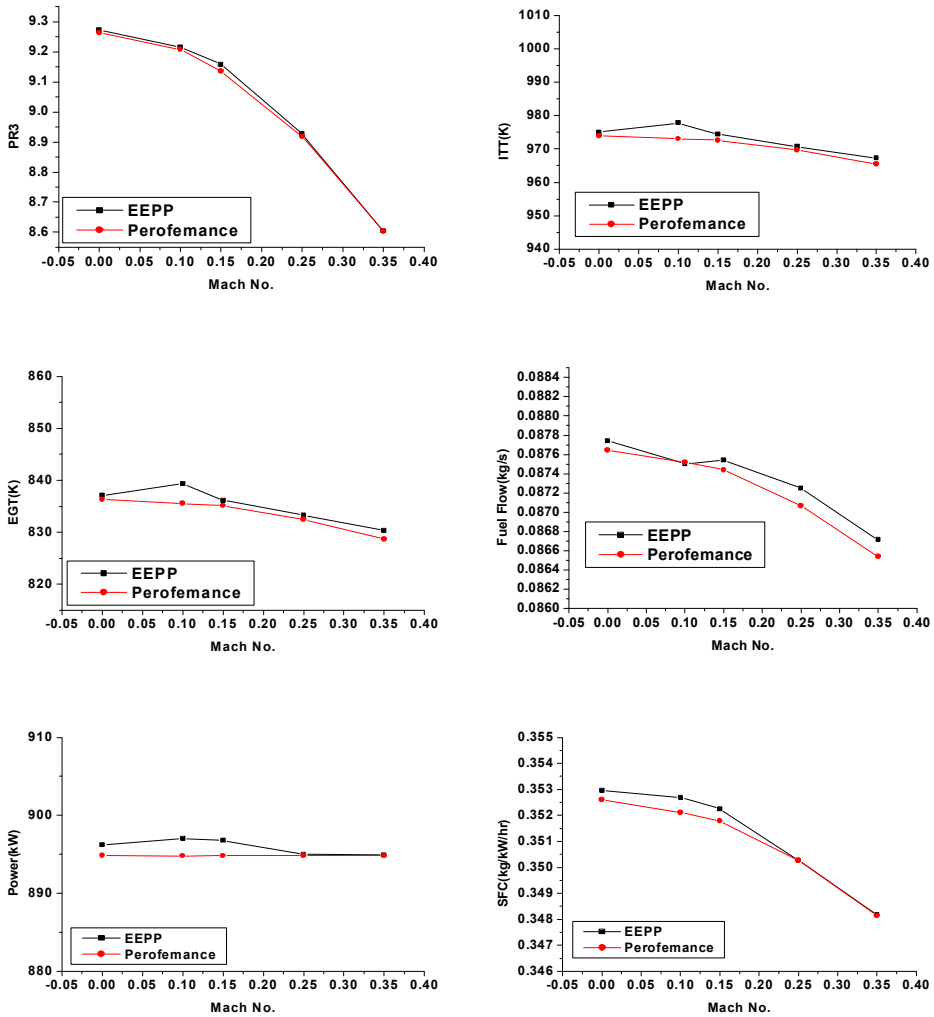


Figure 18-2. Performance with Flight Mach No. variation at Altitude = 0.0

Table 4-2. Uninstallation Performance with Flight Mach No.

Mach No.	Variable	Performance	Design Point	Error(%)
0	Compressor Output Pressure Ratio	9.272462	9.2632	1.0010
	Inter Turbine Temperature (K°)	975.0225	973.9012	1.0012
	Exhaust Gas Temperature (K°)	837.0955	836.296	1.0010
	Fuel Flow Rate (kg/sec)	0.08774	0.087643	1.0011
	Shaft Horse Power (kW)	896.2353	894.8351	1.0016
	S.F.C (kg/kW·hr)	0.352953	0.352595	1.0010
0.1	Compressor Output Pressure Ratio	9.216561	9.2075	1.0010
	Inter Turbine Temperature (K°)	977.7685	973.0366	1.0049
	Exhaust Gas Temperature (K°)	839.3286	835.5186	1.0046
	Fuel Flow Rate (kg/sec)	0.087503	0.08752	0.9998
	Shaft Horse Power (kW)	897.0227	894.8139	1.0025
	S.F.C (kg/kW·hr)	0.352694	U0.352111	1.0017

0.15	Compressor Output Pressure Ratio	9.15898	9.1365	1.0025
	Inter Turbine Temperature (K°)	974.4206	972.6106	1.0019
	Exhaust Gas Temperature (K°)	836.1466	835.1353	1.0012
	Fuel Flow Rate (kg/sec)	0.087538	0.087439	1.0011
	Shaft Horse Power (kW)	896.7598	894.8199	1.0022
	S.F.C (kg/kW·hr)	0.352248	0.351783	1.0013
0.25	Compressor Output Pressure Ratio	8.92732	8.9191	1.0009
	Inter Turbine Temperature (K°)	970.6767	969.645	1.0011
	Exhaust Gas Temperature (K°)	833.2913	832.4654	1.0010
	Fuel Flow Rate (kg/sec)	0.08725	0.087065	1.0021
	Shaft Horse Power (kW)	895.0148	894.846	1.0002
	S.F.C (kg/kW·hr)	0.350266	0.350268	1.0000
0.35	Compressor Output Pressure Ratio	8.60387	8.6039	1.0000
	Inter Turbine Temperature (K°)	967.247	965.4458	1.0019
	Exhaust Gas Temperature (K°)	830.3218	828.6931	1.0020
	Fuel Flow Rate (kg/sec)	0.086712	0.086537	1.0020
	Shaft Horse Power (kW)	894.9162	894.8269	1.0001
	S.F.C (kg/kW·hr)	0.348184	0.348149	1.0001

나. 가상 엔진 계측데이터를 이용한 상태감시

기준엔진 성능 성능에 ±4% 임의의 데이터를 생성하여 기준엔진 성능해석 결과의 비교를 통하여 상태감시를 시행하였다. 이때 가상 엔진 계측데이터에서 생성된 임의의 손상 데이터는 기준 엔진 성능해석결과와 비교하여 식(12)와 같이 계산되어 상태감시 된다.

$$\Delta Z = \frac{Z_b - Z}{Z_b} \times 100 \tag{12}$$

여기에서 Z_b는 기준 성능 값을 Z는 저하된 성능 값을 나타낸다. 계산된 ΔZ는 잔단 적용 단계에서 퍼지 및 학습된 신경회로망을 이용한 엔진 구성품의 손상을 식별하고 손상정도를 제시한다.

Figure 19-1과 19-2는 각 비행 상태에서 마하수 변화에 따른 상태감시를 모사한 것으로 오차율이 1%내외로 매우 정확하다고 판단하기 어려우나 상태감시 프로그램으로써 크게 문제가 없음을 확인할 수 있다.

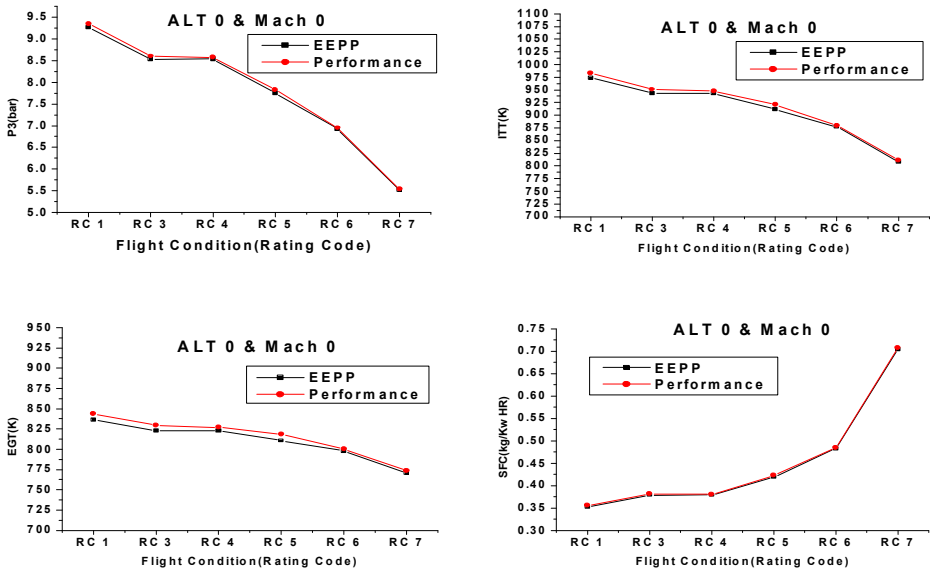


Figure 19-1. Monitoring results of parameter found at sea level static standard atmospheric condition

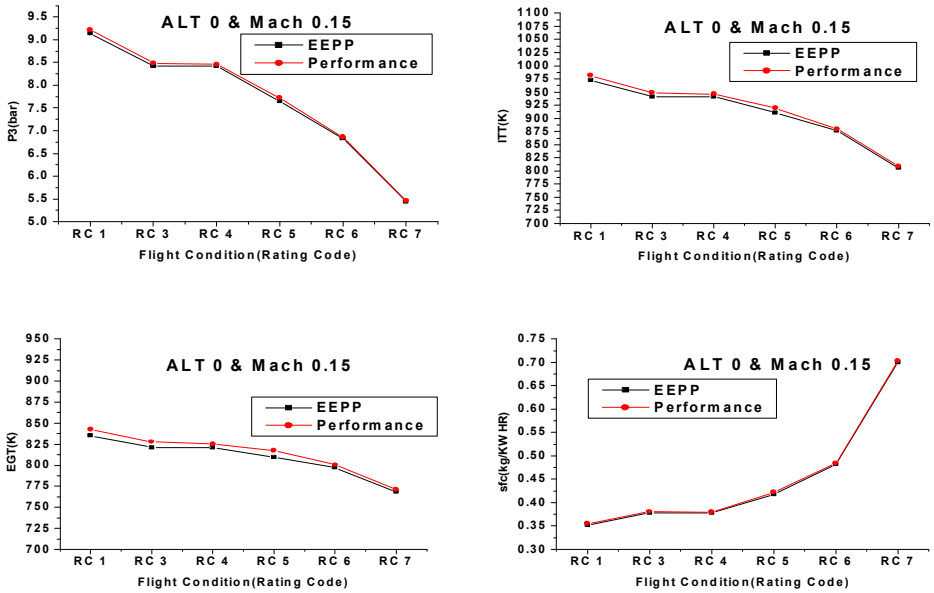


Figure 19-2. Monitoring results of parameter found at sea level static standard Mach No. 0.15, atmospheric condition

또한 제안된 입력모듈을 통한 신호를 감시하고 있는지를 확인하기 위해 Figure 19-3와 같이 엔진 계측 신호 발생기 프로그램에 임의의 신호 ±4%를 발생시켰을 때, 상태감시 결과를 비교하여 실제 엔진의 경향을 파악할 수 있을 것으로 판단된다.

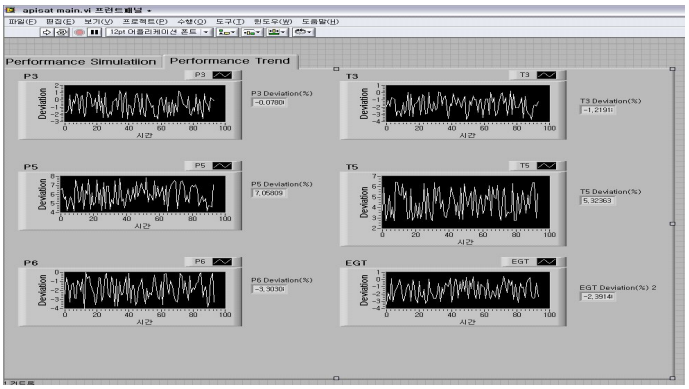


Figure 17-3. Engine Performance condition monitoring results displayed on front panel

다. 실제 엔진 계측데이터 통신을 이용한 상태감시

본 연구는 온라인 상태감시를 통하여 엔진의 상태에 대한 경향을 파악하고 이를 통하여 EMS의 기초 단계로써 실제 엔진에서의 데이터를 이용하여 엔진의 상태를 감시할 수 있는지 여부에 대하여 연구하였다. 실제로 앞서 연구 결과에서 볼 수 있듯이 on condition monitoring system으로 on-line 상태에서도 충분히 엔진의 상태에 대하여 감시할 수 있다 판단하여, 실제 엔진의 실험데이터를 데이터통신을 통하여 연구된 프로그램에 적용하였다.

다음은 실제 엔진의 데이터를 Ethernet 통신을 통하여 데이터를 전송 받아 엔진의 상태를 모니터링 한 결과로써 고도는 0ft로 제한하고 Mach No.를 수정하였을 때의 결과로 최대오차가 약 5%이다. 또한 그래프 상으로도 상태에 대한 변화가 일정함을 알 수 있다. 이를 통하여 데이터 통신을 통한 On-line condition monitoring system으로써 확인 할 수 있음을 검증하였으며, 추후 프로그램의 수정을 통하여 엔진 비교할 수 있는 엔진의 파라미터 수를 늘릴 예정이다.

다음 Figure 20-1은 실제 시험을 통해 획득된 엔진 계측데이터를 분석한 것이다.

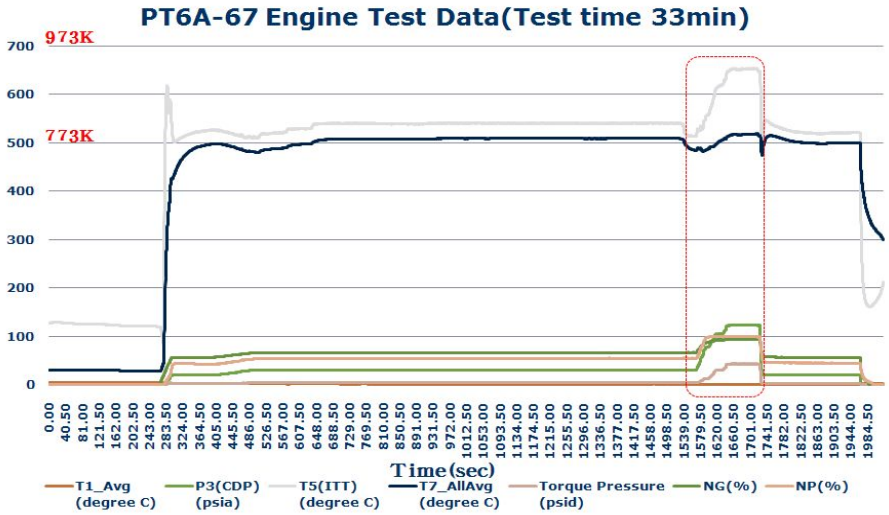


Figure 20-1 Engine ground performance test data during about 33minutes

제공된 실험의 실제 시험 시간이 33분이나 검증을 위하여 엔진 시동 및 안정화 시간을 제외하고 실제 엔진의 운용 영역에 대한 부분을 추출하여 상태감시 프로그

램의 검증을 위하여 사용하였다. Figure 20-2는 실제 상태감시 프로그램 검증을 위한 데이터 샘플이다. 이때 샘플 시간은 2분 56초 동안의 데이터를 사용하였다.

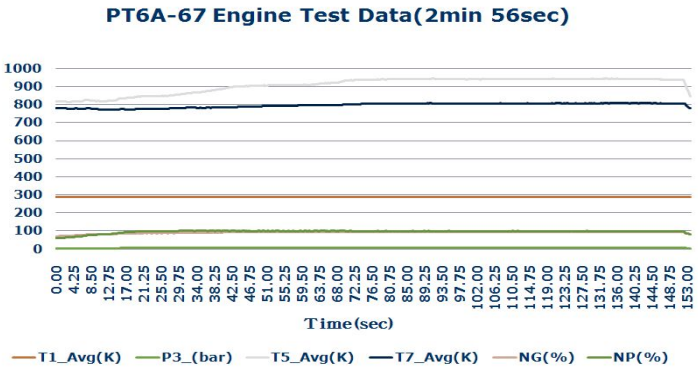


Figure 20-2 Test data sampling during about 3minutes

Figure 20-3은 약 140초 동안의 엔진 상태에 대한 모니터링 결과를 다시 1초당 4개의 샘플링 된 데이터의 평균값과 비교하여 경향을 확인하였다.

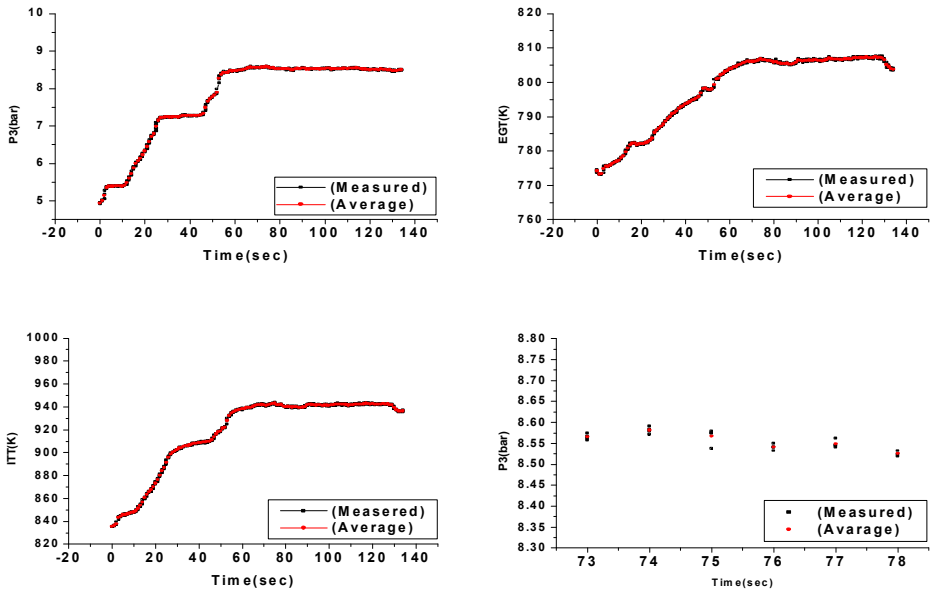


Figure 20-3 2 minutes test data for validation of the proposed on-line performance monitoring system

Figure 20-4은 Rating Code 변화에 따른 성능차를 해석한 것으로 최초 프로그램의 입력방식이 Rating Code방식이었으므로 비교해본 결과 rpm에 비해서는 차이가 나나 거의 일치함을 확인하였다.

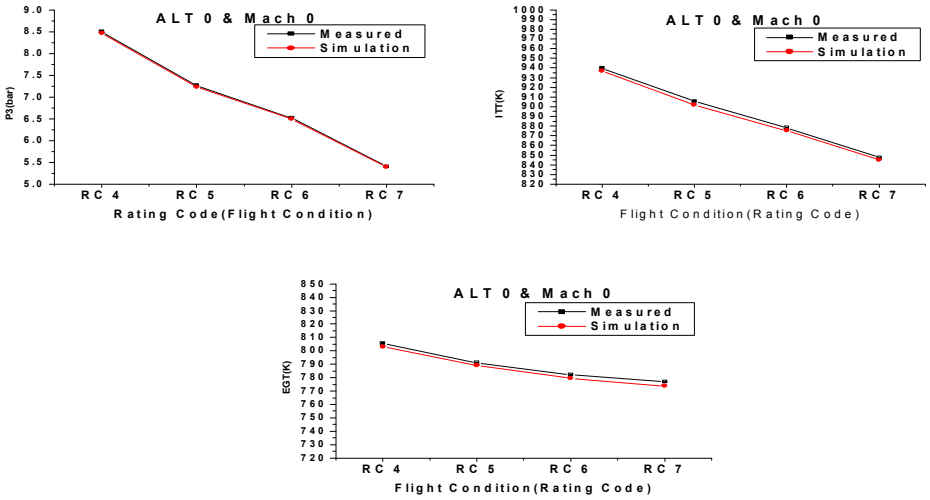


Figure 20-4. Comparison results between measured engine performance data ('Measured') and simulated engine performance data ('Simulation') at various Rating Codes

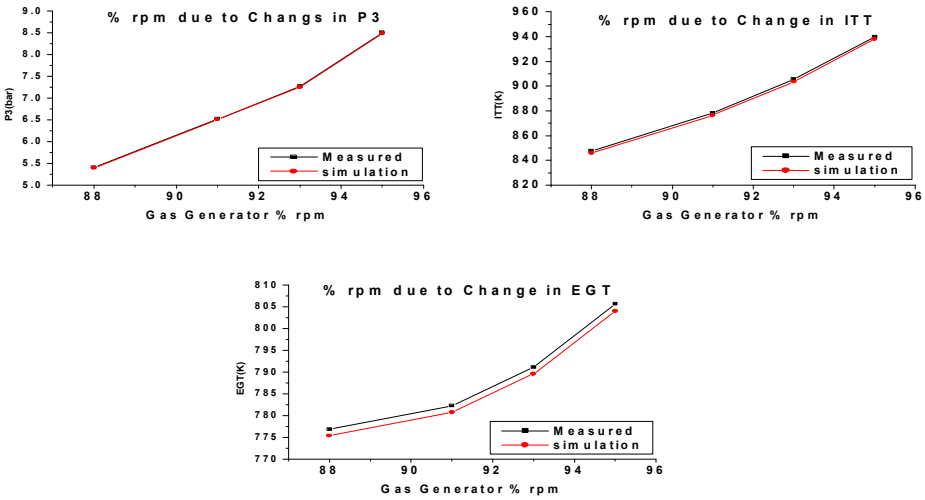


Figure 20-5. Comparison results between measured engine performance data ('Measured') and simulated engine performance data ('Simulation') at various %rpm

Figure 20-5는 가스발생기 rpm을 이용하여 특정 영역에 대한 비교를 통하여 전체적인 경향과 실제 특정 영역에 대한 부분을 확인함으로써 프로그램의 신뢰성을 검토하였다.

따라서 실제 엔진 계측 데이터는 Rating Code방식과 rpm입력 방식을 모두 사용할 수 있을 것으로 판단된다.

제 5 장 GUI기반 상태진단 MATLAB® 개발

제 1 절 개 요

본 연구의 최종 목표인 상태진단 프로그램은 앞서 제안된 LabVIEW를 이용한 상태감시 프로그램에서 획득된 계측데이터 변화량을 분석하여 퍼지-신경회로망으로 구성된 프로그램을 사용하여 엔진의 상태를 진단 할 수 있는 프로그램을 제안하였다.

제안된 프로그램은 퍼지-신경회로망을 이용한 다중 손상 진단 기법을 이용하였다. 제안된 기법은 Model-based 진단 기법의 성능향상을 위해 신경망 이론, 퍼지 논리, 유전알고리즘과 같은 인공지능 이론을 적용한 진단 전문가 시스템으로 그중 신경회로망은 우수한 학습 성능으로 인해 진단시스템에 가장 많이 이용되고 있으나 학습할 데이터가 많아지면 훈련하는데 시간이 오래 걸리고 정확성이 떨어질 수 있는 단점이 있다. 또한 각 구성품의 단일 손상에서 여러 구성품에서 발생하는 다중 손상 탐지를 효과적으로 하기 위해서는 그 구조가 매우 복잡해진다.

따라서 본 연구에서는 1차적으로 엔진의 손상된 구성품을 식별하고 그에 맞게 훈련된 신경회로망을 적용하여 엔진의 상태를 정량적으로 진단하는 알고리즘을 이용한 프로그램을 기본으로 다중손상 진단 프로그램을 제안하였다. 다중손상 진단을 위한 계산 순서는 Figure 21과 같다.

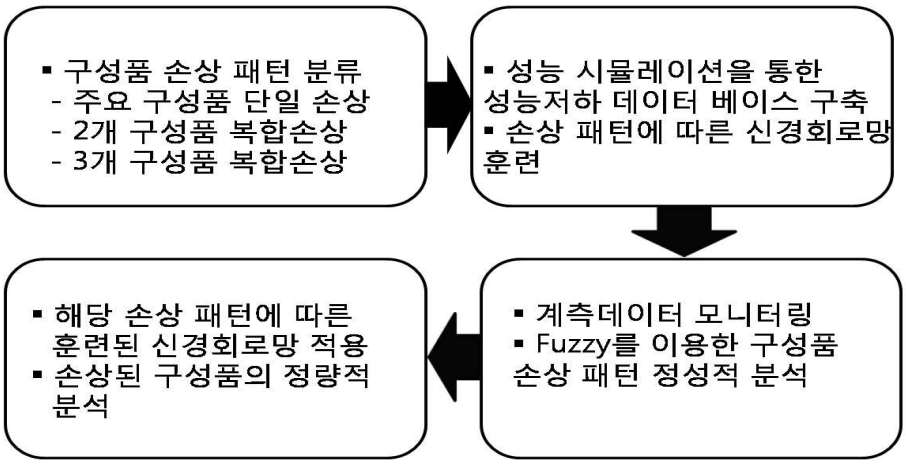


Figure 21. Model-based multi fault diagnosis algorithm

1. 퍼지 로직을 이용한 손상 성능 데이터 축적

온라인 상태감시 프로그램으로 획득된 계측데이터들의 변화량을 이용하여 엔진의 고장 여부를 판단하기 위해서는 먼저 다양한 엔진 상태에서 계측된 파라미터들의 데이터베이스를 구축하여야 한다. 대상엔진의 운영영역과 운용조건으로부터 모니터링 된 파라미터를 분석하여 분석된 변화의 추세로부터 퍼지 로직을 이용하여 진단을 수행할 수 있는 퍼지 규칙을 생성한다. 퍼지는 물리적인 변화의 정도를 나태내줄 수 있으며 엔진 상태에 대한 분석 결과를 0 ~ 1까지의 범위의 값으로 제시하여 가능성이 높은 손상 원인에게 1에 가까운 값을 부여한다[44][45].

추진시스템의 계측변수는 동력터빈 입구온도, 배기가스 온도, 연료유량, 토크이다. 주요 구성품의 손상 패턴은 압축기, 압축기터빈, 동력터빈에서 각각 손상이 발생하는 경우와 2개 구성품에서 동시에 손상이 발생하는 경우, 그리고 세 구성품에서 동시에 손상이 발생하는 경우 등을 고려해볼 수 있다.

본 연구에서는 주요 구성품에서 발생할 수 있는 손상패턴을 Table 5-1과 같이 7가지 경우로 분류하였고, 각각의 손상 패턴에 따라 계측파라미터의 변화를 살펴보면 Table 5-2와 같은 경향을 갖는다.

Table 5-1. Considered component fault patterns

FP 1	Compressor fouling
FP 2	Compressor turbine erosion
FP 3	Power Turbine Erosion
FP 4	Comp. Fouling & Comp. turbine erosion
FP 5	Comp. Fouling & & Power Turbine Erosion
FP 6	Comp. turbine erosion & Power Turbine Erosion
FP 7	Comp. Fouling & Comp. turbine erosion & power turbine erosion

Table 5-2. Measured parameter change (MPC) trends depending on component fault patterns

	ITT	EGT	WF	Torque
FP 1	+	+	++	+
FP 2	+	+	+++	+
FP 3	−	+	−	−−
FP 4	++	++	+++	++
FP 5	++	++	++	+
FP 6	++	++	+++	+
FP 7	+++	+++	+++	+

Figure 22는 진단 프로그램의 일부인 MATLAB의 FIS 에디터를 이용하여 [46][47] 손상된 구성품의 식별을 위한 mamdani-type 퍼지추론 시스템으로 분류된 각각의 손상 패턴에 따라 계측파라미터의 변화를 분석하여 퍼지추론 법칙을 만들어 사용된다. 이 때 사용된 이론은 'MAMDAI' 이론을 적용하였으며 비퍼지화에는 'Centroid'기법이 적용되었다.

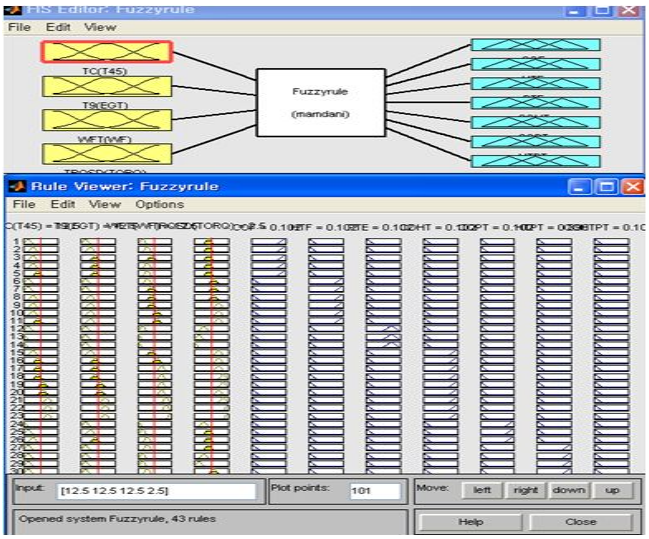


Figure 22. MAMDANI type Fuzzy Inference System for isolating faulted components

2. 구성품 손상의 정량적 진단 위한 신경회로망 설계 및 훈련

신경회로망의 훈련은 기존에 구축한 손상 조건에 따른 성능 저하율 데이터베이스를 사용하였다. 신경회로망의 훈련에는 FFBP(Feed Forward Back Propagation) 알고리즘을 적용하였으며 Figure 23-1과 같이 각각 1개의 입력층과 은닉층과 출력층을 갖는다. 은닉층의 전달함수에는 Tangent Sigmoid 함수, 식(13)을 적용하였으며 출력층의 전달함수에는 선형 함수는 식(14)을 적용하였다[48].

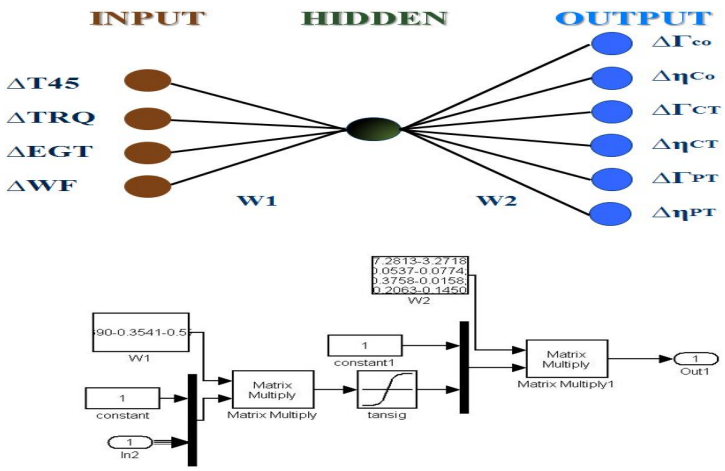


Figure 23-1. Feed forward Neural Network program using SIMULINK

$$\varphi = \frac{e^{ax} - e^{ax}}{e^{ax} - e^{-ax}} \tag{13}$$

$$\varphi = x \tag{14}$$

진단의 정확성을 계산하는 RMS 오차는 식(15)과 같이 정의하였으며 신경회로망은 RMS 오차가 1% 이내에 들도록 훈련된다.

$$RMS\,error = \sqrt{\frac{\sum_{i=1}^{n}(y_n - T_n)^2}{n}} \tag{15}$$

여기에서 T는 엔진에 임의로 이식한 성능저하율이며 y는 신경회로망의 진단 결과이다.

Figure 23-2는 역전파 신경회로망 훈련의 알고리즘이며 Figure 23-3은 MATLAB/SIMULINK를 이용하여 제작한 퍼지-신경망 인공지능 진단 프로그램이다.

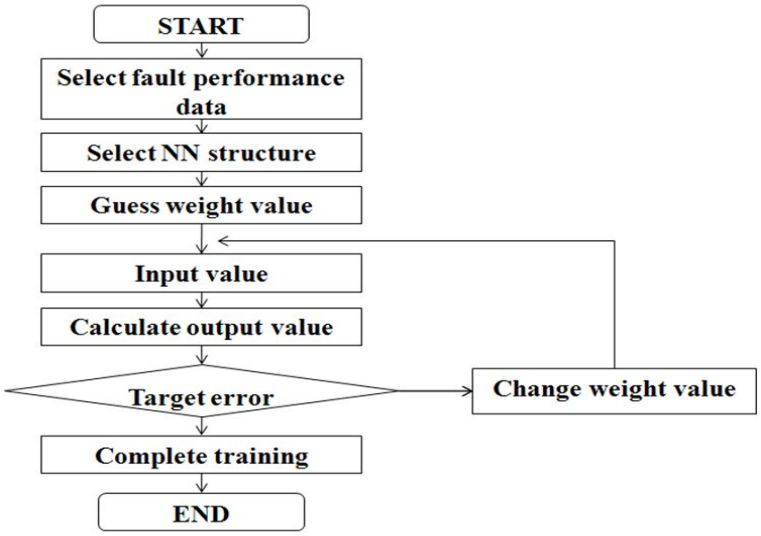


Figure 20-2. Back propagation algorithms for training Neural Network

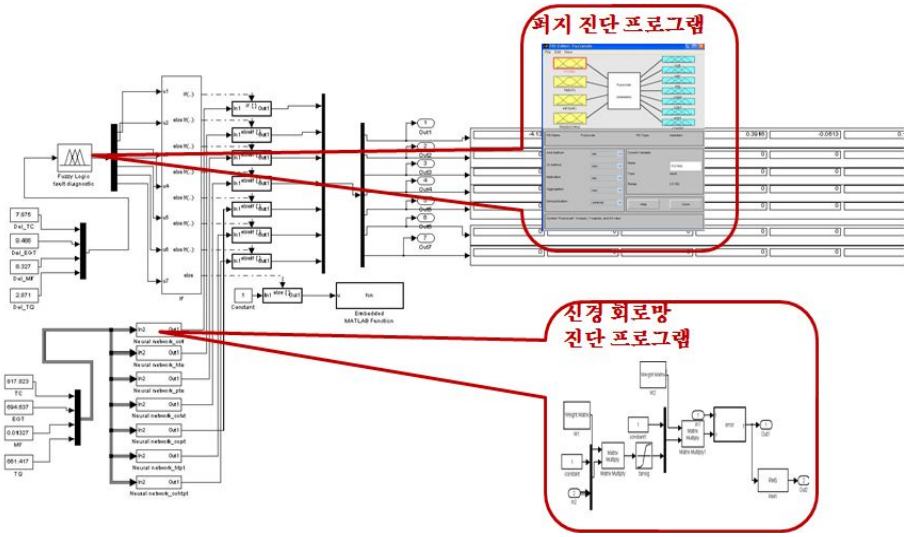


Figure 20-3. Fuzzy-Neural Network fault diagnosis program

제 2 절 상태진단 프로그램 개발 및 검증

고장진단 프로그램 환경 개발을 위해 MATLAB의 GUI design environment(GUIDE) tool을 이용하였다. Figure 24-1과 Figure 24-2와 같이 Layout Editor를 이용하여 창을 만들면 MATLAB의 m-file을 얻을 수 있으며, 이 Layout에서 얻은 m-file을 각 창의 기능에 맞도록 수정하였다.

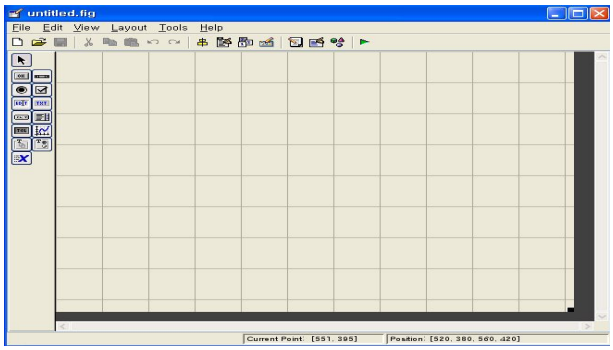


Figure 24-1. MATALB GUIDE layout editor

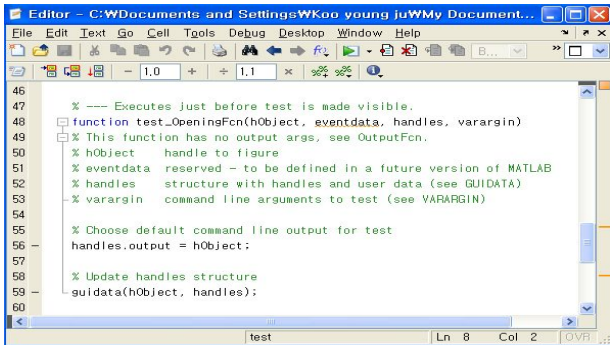


Figure 24-2. MATLAB Layout Editor by m-file

제안된 퍼지-신경회로망 기법을 적용한 다중손상 진단 프로그램은 통합형 진단 시스템 구축을 위한 기초연구 단계로 먼저 LabVIEW를 이용한 상태감시에서 획득된 계측데이터 변화량과 실제 엔진 계측데이터를 획득하여 입력하면 퍼지-신경회로망 프로그램에 의해서 정량적, 정성적 해석이 이루어지고 실제 엔진의 손상 정도를 진단하여 디스플레이 시키도록 제안하였다.

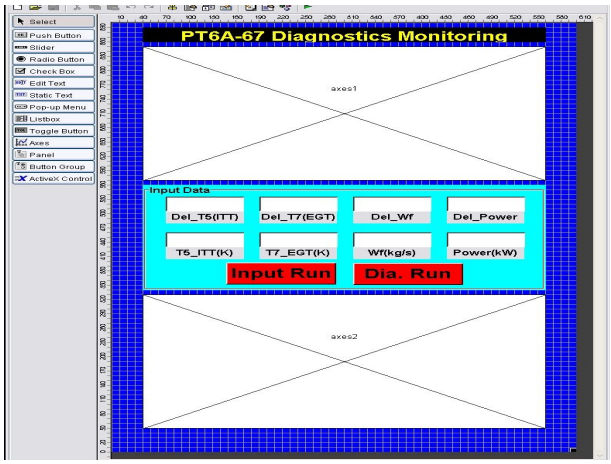


Figure 24-3. Lay out of diagnostics monitoring

프로그램 구성은 상태감시 결과인 계측데이터를 입력하고 실제 엔진에서 계측된 데이터를 입력하고 'Input run' 시키면, 퍼지-신경회로망 프로그램과 연동되어 퍼지룰과 앞서 학습되었던 신경회로망 프로그램이 작동된다. 그 후 'Dia.run'을 실행시키면 Figure 24-4와 같은 결과를 얻을 수 있다.

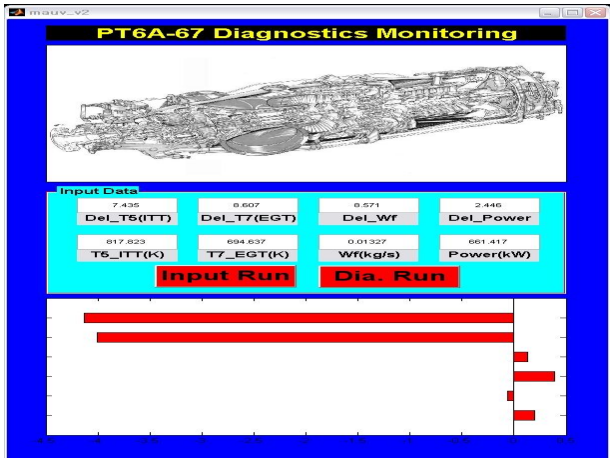


Figure 24-4 Results of faulted components quantified by diagnostic program

제안된 GUI기반 인공지능 진단 프로그램 검증을 위해 임의의 손상을 가정하였다. Table 6은 임의로 가정한 엔진 성능 저하율이다.

Table 6. Implanted fault values (IFV) of engine major components

	COMA	COEF	GTMA	GTEF	PTMA	PTEF
1	-5	-3	0	0	0	0
2	0	0	5	-3	0	0
3	0	0	0	0	5	-3
4	-4	-2	4	-2	0	0
5	-4	-2	0	0	4	-2
6	0	0	4	-2	4	-2
7	-5	-5	5	-5	4	-4

Table 7은 각 경우의 계측변수 변화량을 나타며 이를 이용하여 기 확정된 퍼지 규칙과 훈련된 신경망을 적용하여 가정된 손상패턴과 손상량을 가능한 정확히 예측할 수 있다면 제시된 방법의 검증이 이루어진다고 할 수 있을 것이다.

Table 7. Measured parameter changes due to implanted faults (%)

	ITT	EGT	MF	Torque
1	7.435	8.067	8.571	2.446
2	7.817	7.027	14.367	8.231
3	-3.051	-0.933	-4.408	-6.078
4	14.385	14.072	21.714	10.588
5	5.196	7.226	5.959	-0.762
6	5.463	6.372	10.531	3.643
7	19.986	21.518	27.755	10.456

먼저 정성적인 손상특성 구별을 위해 퍼지 로직 알고리즘을 이용하여 손상특성을 구별한 결과 각 손상 CASE별로 Table 8과 같은 결과를 얻었다.

Case1의 경우 fault pattern이 1인 경우가 0.51로 가장 1에 가까운 결과를 보여 압축기 오염의 가능성이 높음을 알 수 있다. case 7의 경우에는 fault pattern이 7인 경우가 0.566으로 세 구성품 모두 손상된 다중손상임을 확인할 수 있다.

각 CASE별로 퍼지를 이용해 손상특성을 파악 후 훈련된 신경회로망을 통해 정량적인 손상 정도를 진단하였다. Figure 25-1 ~ Figure 25-7은 가정된 손상

CASE에 대한 본 연구에서 제안한 GUI기반의 진단 프로그램으로 신경회로망 방법에 의해 식별된 손상과 비교를 나타낸 것으로 상당히 정확하게 손상을 예측할 수 있음을 보여준다.

Table 8. Results of faulted components isolated by Fuzzy Inference System (IFC: Input fault cases, OFC: Output fault cases)

FP / case	1	2	3	4	5	6	7
1	0.51	0.091	0.089	0.087	0.431	0.263	0.092
2	0.47	0.586	0.089	0.087	0.092	0.454	0.092
3	0.098	0.091	0.688	0.087	0.092	0.085	0.092
4	0.098	0.414	0.089	0.570	0.092	0.085	0.434
5	0.401	0.091	0.207	0.087	0.568	0.085	0.092
6	0.430	0.091	0.228	0.087	0.280	0.527	0.092
7	0.453	0.270	0.089	0.453	0.280	0.473	0.566

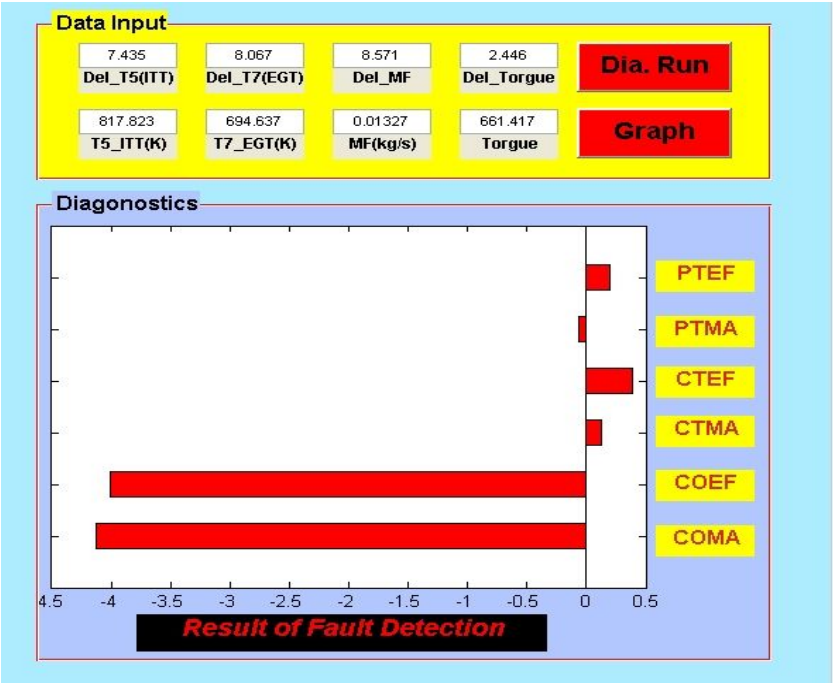


Figure 25-1. Results of faulted components quantified by GUI based diagnostic program(CASE 1 : single fault)

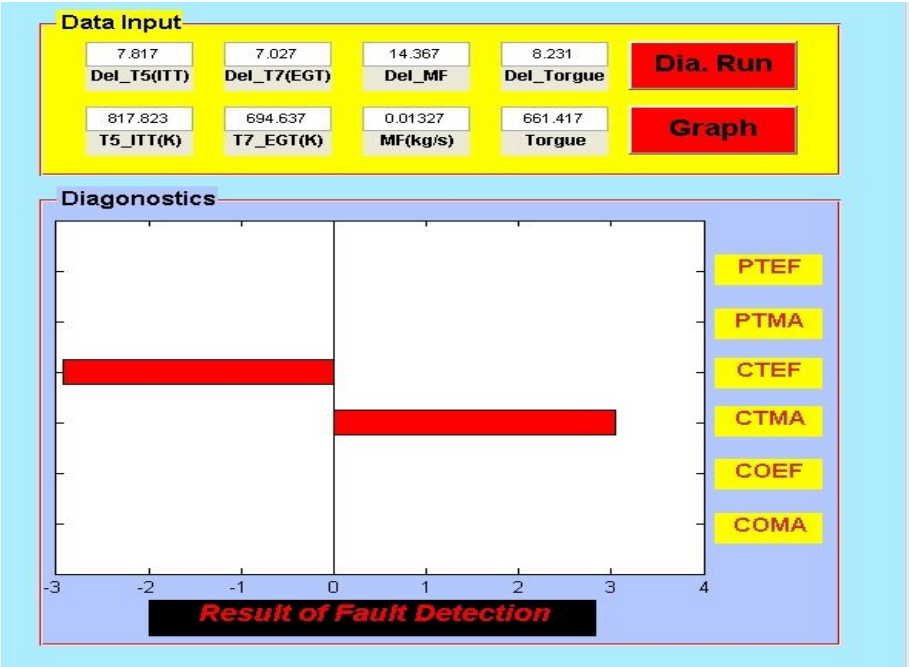


Figure 25-2. Results of faulted components quantified by GUI based diagnostic program(CASE 2 : single fault)

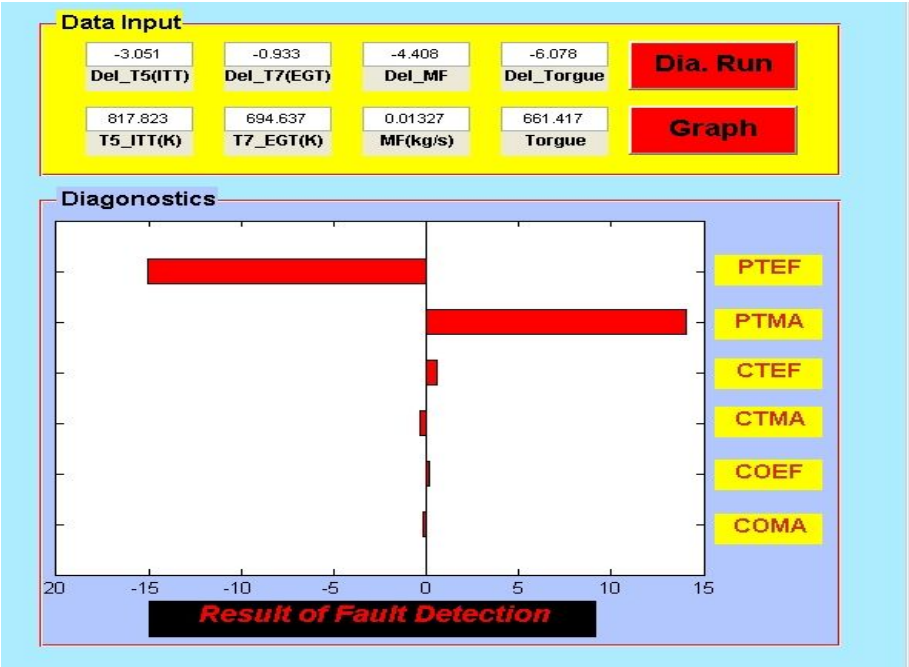
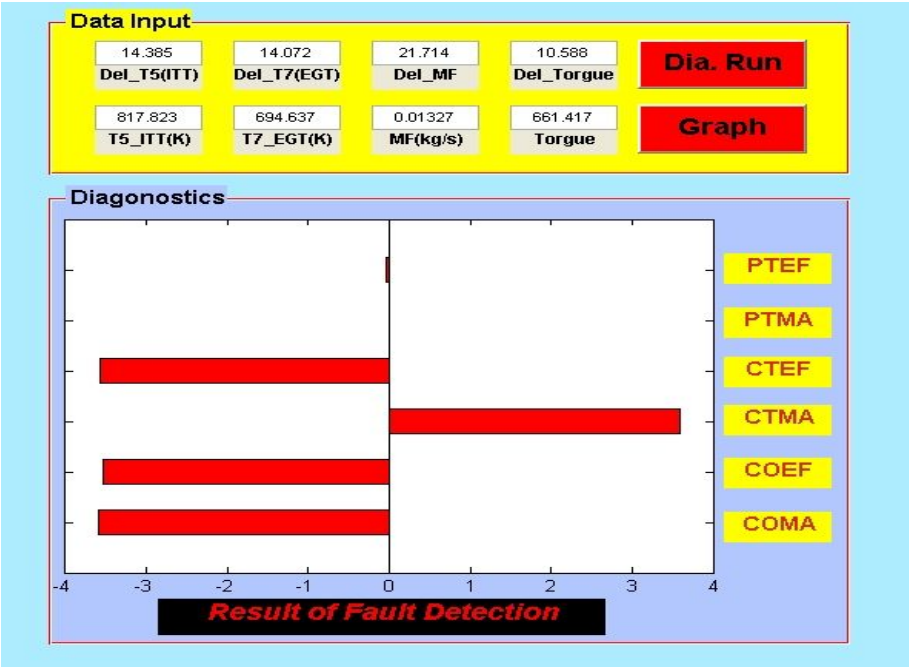


Figure 25-3 Results of faulted components quantified by GUI based diagnostic program(CASE 3 : single fault)



Figue 25-4 Results of faulted components quantified by GUI based diagnostic program(CASE 4 : double fault)

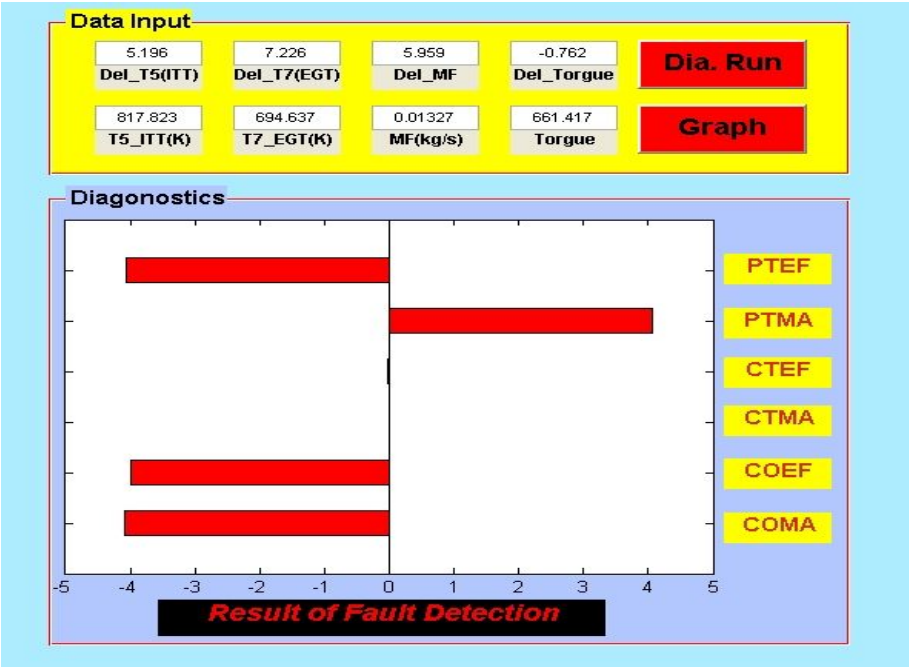


Figure 25-5 Results of faulted components quantified by GUI based diagnostic program(CASE 5 : double fault)

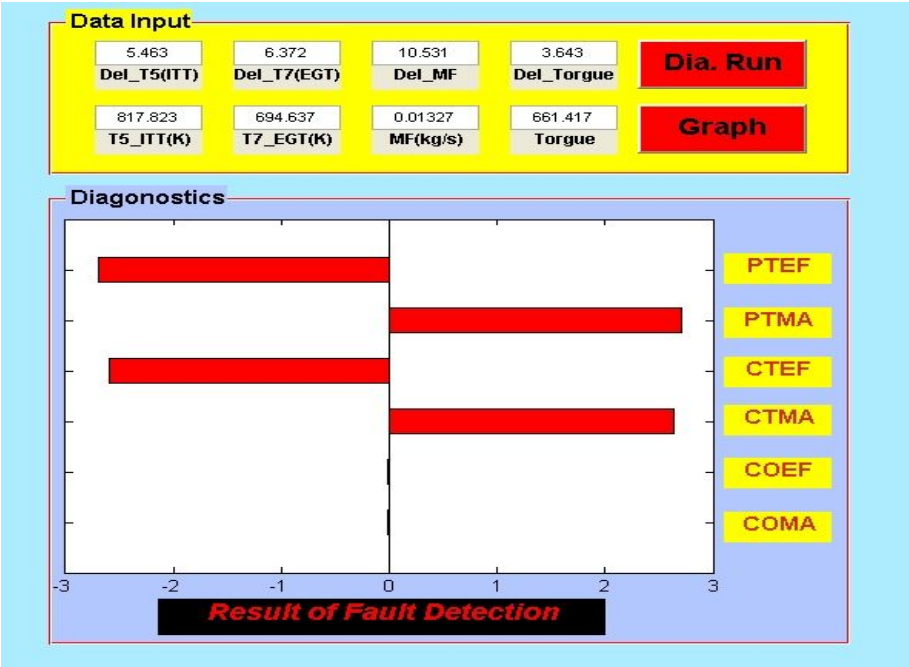


Figure 25-6 Results of faulted components quantified by GUI based diagnostic program(CASE 6 : double fault)

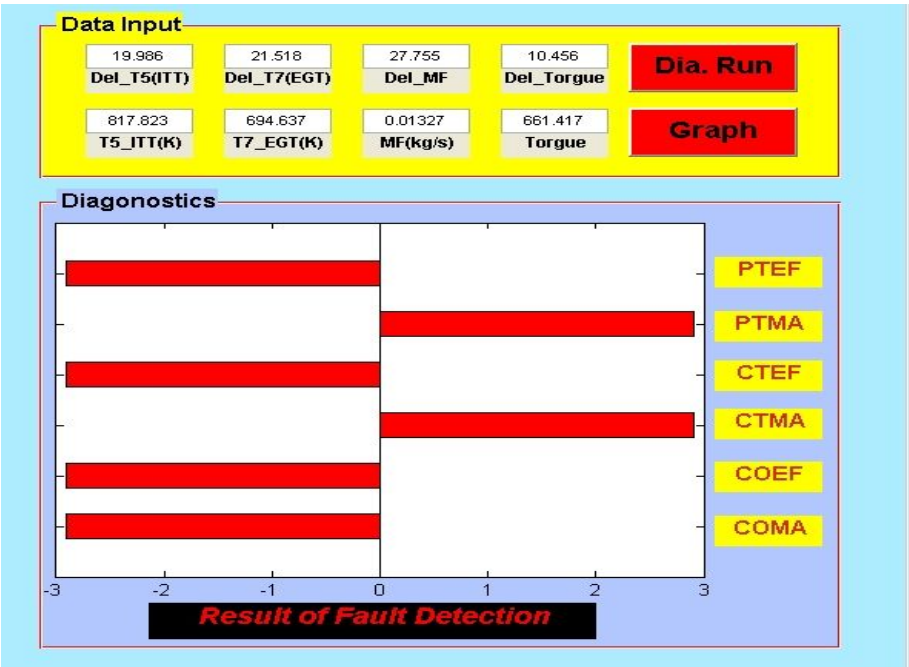


Figure 25-7 Results of faulted components quantified by GUI based diagnostic program(CASE 7 : triple fault)

제 6 장 결 론

본 논문에서는 사용자의 편리함과 정확한 의미 전달을 위해 작동하기 쉬운 순서와 표현방법을 제시하는 GUI(Graphic User Interface)기반 상태감시 및 진단 프로그램을 제안하고 개발된 프로그램을 이용하여 상태감시 및 진단을 수행하였다.

먼저, 엔진의 정확한 상태감시를 위하여 기준 엔진 성능모델을 제안하였고, 기준 엔진 성능모델을 엔진 제작사에서 일부 제공된 성능덱 데이터를 이용하여 프로그램의 효용성을 검증하였다. 상태감시를 위한 실제 엔진의 계측데이터 획득이 어려움에 따라 가상의 엔진신호 계측 프로그램을 제안하여 기준 엔진 성능모델과 임의의 데이터를 생성하는 가상의 엔진신호 계측 데이터를 이용한 상태감시 여부를 검증하였다. 또한, 실제 엔진계측데이터 획득을 위해 실시간에 거의 가까운 온라인 상태감시를 위하여 엔진계측데이터를 획득할 수 있은 통신 프로그램을 제안하여 실제 통신을 통한 데이터획득과 획득된 데이터와 기준 엔진 성능모델과의 상태감시를 통하여 실제 엔진계측데이터를 이용한 상태감시 또한 검증하였다.

이를 바탕으로 다양한 조건에서의 손상에 따른 성능저하 데이터베이스 구축을 위한 기초 연구로 활용되었으며, 이렇게 구축된 데이터베이스는 퍼지 법칙을 생성하고 신경회로망을 구축하여 인공지능 진단프로그램 연구에 기초 자료로 활용되었다.

또한 이렇게 구축된 진단프로그램을 다시 한 번 사용자 편의성을 제공하고 사용이 편리한 GUI형태로 변환하였다. 이때 퍼지-신경회로망 기법이 들어간 프로그램의 구성을 적절히 사용하여 다중손상 진단이 가능한 GUI형태의 진단 프로그램을 제안하고 검증하였다.

따라서 본 연구 결과가 추후 실제 항공기용 추진시스템의 온라인 상태진단 시스템에 적용이 가능할 것으로 판단된다.

참 고 문 헌

1. Yi Guang Li, "Gas Turbine Diagnostics", Cranfied University, 1997
2. Sellers, J. F. and Daniele, C. J.. "DYNGEN - A Program for Calculating Steady-state and Transient Performance of Turbojet and Turbofan Engines." Technical Report TN-D-7901, NASA Lewis Research Center, 1975
3. MacMillan, W. L.. "Development of a Modular Type Computer Program for the Calculation of Gas Turbine Off Design Performance." Ph.D. Thesis, Cranfield University, UK, 1974
4. Palmer, J. R. and Yan, C. Z.. "TURBOTRANS - A Programming Language for the Performance Simulation of Arbitrary Gas Turbine Engines with Arbitrary Control Systems." International Journal of Turbo and Jet Engines : 19-28, 1985
5. El-Masri, M. A.. "GASCAN - An Interactive Code for Thermal Analysis of Gas Turbine Systems." Journal of Engineering for Gas Turbines and Power, Vol. 110 : pp.201-209, 1988
6. Bettocchi, R., Spina, P. R. and Fabbri, F.. "Dynamic Modeling of Single-Shaft Industrial Gas Turbine." ASME 96-GT-332, 1996
7. Crosa, G., et al. "Heavy-Duty Gas Turbine Plant Aerothermodynamic Simulation using SIMULINK." Journal of Engineering for Gas Turbines and Power, Vol. 120 : 550-556, 1998
8. Kurzke, J., 2001, "A Program to calculate Design and Off-Design Performance of Gase Turbines", GasTurb 9 User's Manual
9. 김석균. "항공기용 터보제트 엔진의 동적모사와 제어에 관한 연구" 조선대학교 대학원 항공우주공학과 석사학위논문, 1997
10. Kong, C. D., Ki, J. Y. and Kho, K. W.. "Performance Analysis, Real Time Simulation and Optimal Control of Medium-Scale Commercial Aircraft Turbofan Engine" International Society for Air Breathing Engines, 1999
11. Kong, C. D. and Ki, J. Y.. "Performance Simulation of Turboprop Engine

for Basic Trainer" ASME Turbo Expo, 2001-GT-391, 2001
12. 노홍석, 기자영, 공창덕.. “성능해석을 위한 터보프롭 엔진의 SIMULINK 모델링” 한국추진공학회, 춘계학술발표회, pp.76-79, 2001
13. 공창덕, 고성희, 기자영, 전용민, 안이기.. “흡입구 손실을 고려한 헬리콥터 추진시스템의 장착성능해석 모델에 관한 연구”, 한국추진공학회지, Vol. 12, no.1, pp.51-59, 2008. 14. 공창덕, 고성희, 기자영, 유혁, 안이기.. “SIMULINK를 이용한 헬리콥터 추진시스템의 성능해석에 관한 연구”, 한국항공우주학회 학술대회 발표논문, pp943-947, 2007
15. 최인수., “EASY5를 이용한 터보프롭 엔진의 성능 해석”, 조선대학교 대학원 석사학위논문, 2002
16. Urban, L. A.. "Gas Turbine Engine Parameter Interrelationships." Hamilton Standard Division of United Aircraft Windsor Locks, 1969
17. rewal, M. S.. "Gas Turbine Engine Performance Deterioration Modelling and Analysis." Ph.D. Thesis, SME, Cranfield University, UK, 1988
18. Diakunchak, I. S.. “Performance Deterioration in Industrial Gas Turbines”, Journal of Engineering for Gas Turbines and Power, Vol.114 : pp.161-168, 1992
19. House, P.. "Gas Path Analysis Techniques Applied to a Turboshaft Engine." MSc Thesis, SME, Cranfield University, UK, 1992
20. Escher, P. C.. "PYTHIA : An Object-Oriented Gas Path Analysis Computer Program for General Applications." Ph.D. Thesis, SME, Cranfield University, UK, 1995
21. English, L.. "Application of Gas Path Analysis, Gas Path Debris Monitoring and Expert System Technology to the Allison T56 Turboprop Engine." MSc Thesis, SME, Cranfield University, UK, 1995
22. Kim, S. K.. "Application of Neural Networks to Gas Path Analysis for Health Monitoring of Gas Turbine Engine." MSc Thesis, SME, Cranfield University, UK, 1998
23. Mathioudakis, K.. "Compressor Fault Identification from Overall Performance Data Based on Adaptive Stage Stacking." Trans. ASME Journal of Engineering for Gas Turbines and Power 116 : 156-164, 1994

24. 공창덕, 고성희, 공창덕.. “신경회로망을 이용한 가스터빈 엔진의 지능형 성능 진단에 관한 연구”, 한국항공우주학회 Vol.32 No.3. pp.51-57, 2004
25. 공창덕. 기자영. 이창호.. “스마트무인기용 가스터빈 엔진(PW206C)의 성능모사 및 지능형 진단에 관한 연구”, 한국항공우주학회, 추계학술발표회, pp.180-184, 2006
26. Fishbach, L, H. and Caddy, M. J.. "NNEP - The Navy NASA Engine Program", Technical Report TM-X-71857, NASA Lewis Research Center, 1975
27. Cohen, H. A., Rogers, G. F. C. and Saravanamutto, H. I. H.. "Gas Turbine Theory", Longman Scientific and Technical, 1996
28. 공창덕, 임세명, 김지현.. “성능데이터를 이용한 고고도운용 터보프롭엔진 역모델링 연구”, 한국추진공학회지 제 14권 3호, pp. 16-22, 2010. 6
29. 공창덕, 구영주, 고성희, 유혁.. “ 헬리콥터 터보축 엔진의 온라인 상태진단 프로그램 연구”, 한국항공우주학회,Vol.37 No.12, pp.1238-1244, 2009
30. 임영도, 이상부, “퍼지·신경망·유전진화”, 인솔미디어, 2002
31. 구영주, “퍼지 및 신경망을 이용한 터보축 엔진의 다중손상 진단에 관한 연구”, 조선대학교 대학원 ,항공우주공학과 석사학위논문, 2010
32. Brenda Kaurel "The Art of Humman-Computer Interaction Design", New York, Addison Wesley, pp.12, 1990
33. 황지연.. “디자인사전”, 안그라픽스, pp.232, 2000
34. Philip. p. Walsh and Paul Fletcher, "Gas Turbine Performance", Blackwell Scinece and ASME, 1998.
35. 국방과학연구소 “ 터보프롭엔진 역모델링 및 상태진단 연구 1차년도 보고서”, 2009
36. 곽두영.. “컴퓨터 기반의 제어와 계측 LabVIEWTM", Ohm사, 2007
37. 성기안.. “공학도가 알아야 할 LabVIEW 입문”, 신화전산기획, 2010
38. 박홍복.. “ LabVIEW 8 그래픽컬 프로그래밍”, 정익사, 2007
39. 홍승범, 지만석.. “따라하면서 배우는 LabVIEW기초”, 과학기술, 2010
40. Philip. p. Walsh and Paul Fletcher, "Gas Turbine Performance", Blackwell Scinece and ASME, Chapter 3, pp.103-116, 1998.
41. G. Vachtsevanos, F. L. Lewis, M. Roemer, A.Hess, B. Wu.."Intelligent

Fault Diagnosis and Prognosis for Engineering System", WILEY, pp.374, 2007

42. Link. C. Jaw, Jack. D. Mattingly.. " Aircraft Engine Controls-Design, System Analysis, and Health Monitoring" AIAA EDUCATUIN SERIES, pp.246, 2009
43. 곽두영.. "LabVIEW 컴퓨터 기반의 제어와 계측 Solution", Ohm사, 2002
44. Tsoukalas, L.H. and Uhrig, R.E., 1997, "Fuzzy and Neural Approaches in Engineering", John Wiley & Sons, Inc.
45. Lotfi, A. Z., 1998, "Fuzzy Logic Toolbox", MATLAB User's Guide
46. Lee, H. Y., Mun, G. I., 1999, "Fuzzy-Neuro using MATLAB", A-Jin
47. Changduk Kong, Seonghee Kho, Jayoung Ki, Changho Lee, "A Studys Multi Fault Diagnostics of Smart Unmanned Aerial Vehicle Propulsion System Using Data Sorting and Neural Networks", ASME TURBO EXPO 2008, 2008-GT-50769.
48. Heykin, S., 1994, "Neural Networks - A Comprehensive Foundation", Macmilian

부 록

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Neural Network Design for PT6A-67 Turboprop Engine Fault Detection
%Programmed by K.K.W
%2011.06.02
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function varargout = main_1(varargin)
% MAIN_1 M-file for main_1.fig
%      MAIN_1, by itself, creates a new MAIN_1 or raises the existing
%      singleton*.
%
%      H = MAIN_1 returns the handle to a new MAIN_1 or the handle to
%      the existing singleton*.
%
%      MAIN_1('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MAIN_1.M with the given input
arguments.
%
%      MAIN_1('Property','Value',...) creates a new MAIN_1 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before main_1_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to main_1_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help main_1

% Last Modified by GUIDE v2.5 02-Jun-2011 05:59:15

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @main_1_OpeningFcn, ...
                   'gui_OutputFcn',  @main_1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before main_1 is made visible.
function main_1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to main_1 (see VARARGIN)

% Choose default command line output for main_1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes main_1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = main_1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
org_image=imread('PT6A-67.jpg');
image(org_image, 'parent' , handles.axes1);
set(handles.axes1, 'xticklabel',[],'yticklabel',[],'xtick',[],'ytick',[]);
% Get default command line output from handles structure
varargout{1} = handles.output;


function a_Callback(hObject, eventdata, handles)
% hObject    handle to a (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of a as text
%        str2double(get(hObject,'String')) returns contents of a as a double


% --- Executes during object creation, after setting all properties.
function a_CreateFcn(hObject, eventdata, handles)
% hObject    handle to a (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function b_Callback(hObject, eventdata, handles)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of b as text
%        str2double(get(hObject,'String')) returns contents of b as a double


% --- Executes during object creation, after setting all properties.
function b_CreateFcn(hObject, eventdata, handles)
% hObject    handle to b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function c_Callback(hObject, eventdata, handles)
% hObject    handle to c (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c as text
%        str2double(get(hObject,'String')) returns contents of c as a double


% --- Executes during object creation, after setting all properties.
function c_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function d_Callback(hObject, eventdata, handles)
% hObject    handle to d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of d as text
%        str2double(get(hObject,'String')) returns contents of d as a double


% --- Executes during object creation, after setting all properties.
function d_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to d (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function e_Callback(hObject, eventdata, handles)
% hObject    handle to e (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of e as text
%        str2double(get(hObject,'String')) returns contents of e as a double


% --- Executes during object creation, after setting all properties.
function e_CreateFcn(hObject, eventdata, handles)
% hObject    handle to e (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function f_Callback(hObject, eventdata, handles)
% hObject    handle to f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of f as text
%        str2double(get(hObject,'String')) returns contents of f as a double


% --- Executes during object creation, after setting all properties.
function f_CreateFcn(hObject, eventdata, handles)
% hObject    handle to f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function g_Callback(hObject, eventdata, handles)
% hObject    handle to g (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of g as text
%        str2double(get(hObject,'String')) returns contents of g as a double


% --- Executes during object creation, after setting all properties.
function g_CreateFcn(hObject, eventdata, handles)
% hObject    handle to g (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function h_Callback(hObject, eventdata, handles)
% hObject    handle to h (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of h as text
%        str2double(get(hObject,'String')) returns contents of h as a double


% --- Executes during object creation, after setting all properties.
function h_CreateFcn(hObject, eventdata, handles)
% hObject    handle to h (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in run1.
function run1_Callback(hObject, eventdata, handles)
% hObject    handle to run1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a = str2num(get(handles.a,'string'));
b = str2num(get(handles.b,'string'));
c = str2num(get(handles.c,'string'));
d = str2num(get(handles.d,'string'));
e = str2num(get(handles.e,'string'));
f = str2num(get(handles.f,'string'));
g = str2num(get(handles.g,'string'));
```

```
h = str2num(get(handles.h,'string'));
assignin('base','a',a);
assignin('base','b',b);
assignin('base','c',c);
assignin('base','d',d);
assignin('base','e',e);
assignin('base','f',f);
assignin('base','g',g);
assignin('base','h',h);
[tout,xout,yout] = sim('pt6a');

% --- Executes on button press in run2.
function run2_Callback(hObject, eventdata, handles)
% hObject    handle to run2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[tout,xout,yout] = sim('pt6a');

df1 = yout(1,1);
df2 = yout(2,2);
df3 = yout(3,3);
df4 = yout(4,4);
df5 = yout(5,5);
df6 = yout(6,6);

Y = [df1 df2 df3 df4 df5 df6];
barh(Y,0.5,'r','parent',handles.axes2)
set(handles.axes2,'yticklabel',[]);
```

감사의 글

우여곡절 끝에 진학하게 된 대학원 생활이 벌써 끝을 향하여 달리고 있습니다. 길게만 느껴졌던 대학원 생활이 벌써 지나갔다고 생각하니 후련함과 아쉬움, 섭섭함 등이 남습니다. 그 동안 나름대로 열심히 생활한 적도, 어영부영 한 적도 있어 제대로 마무리를 하지 못한 것 같은 아쉬움이 남지만 끝이 곧 새로운 시작을 의미한다는 말도 있듯이 다시 뛸 준비를 해야겠습니다.

부족한 논문이지만 이 논문을 완성하기까지 주변에서 많은 도움을 주신 분들이 없었더라면 지금 이렇게 감사의 글도 쓰기가 힘들었을 것 같습니다.

가스터빈 엔진에 대해 가르쳐 주시고 연구의 길을 열어주신 지도교수님인 공창덕 교수님께 많은 감사를 드립니다. 많이 부족하였고, 지금도 많이 부족하지만 교수님께서 항상 말씀하신 “이게 다 훈련이다”라는 말씀이 큰 자극이 되었고, 연구에 문제가 생길 때면 새로운 돌파구를 마련해주셨기에 오늘의 제가 있지 않나 생각됩니다. 또한, 학위 심사를 위해 먼 길을 마다하고 오셔서 꼼꼼히 논문 지도를 해주신 오성환 박사님, 기자영 선배님께 다시 한 번 감사드립니다.

실험실 생활을 하면서 많은 조언과 도움을 주었던 현범 형에게 먼저 고맙다는 말을 해야 할 것 같습니다. 매우 독특한 성격의 저를 받아주시고 이해해주시고 또한 박사과정이란 이런 것이구나 하는 생각을 갖게 해주셔서 감사하는 마음을 전합니다. 더불어 같이 생활한 실험실 식구에게도 감사하는 마음을 전합니다. 자신의 능력을 120% 발휘하는 세명 형 표현을 잘못해서 항상 고맙다는 표현을 못하였는데 학문적으로 그리고 기타 여러 가지 점에서 너무나도 많은 도움을 주셔서 너무나도 고맙고 감사합니다. 블레이드 연구로 무척 바빴던 동기생 하승이도 열심히 한 결과대로 논문이 잘 마무리되어 축하한다. 현범 형을 제외하고 가장 긴 실험실 생활을 한 민웅, 모든 운동과 어떠한 일에도 자신만만한 성진, 완벽주의자 경선 마지막으로 실험실 막내 역할을 하고 있는 길수, 동빈이도 옆에서 많은 부분을 도와주어서 고맙다는 말을 전한다.

마지막으로 지금의 나를 키워주시고 공부를 계속할 수 있도록 뒷바라지 해주신 부모님께 더 없는 감사를 드립니다. 부모님의 고생과 희생 그리고 저를 믿어주신 마음이 있어 제가 이렇게 편하게 공부를 할 수 있었던 원동력이 되었던 것 같습니다. 아울러 옆에서 힘이 되어준 친구들아 고맙다.