

M1-0104-00-0028

국가지정연구실사업

National Research Laboratory Program

지능형 통합 비행제어시스템 설계기법 연구

A Study on the Integrated and Intelligent Flight Control System Design

서울대학교 비행역학 및 제어 연구실

과학기술부

제 출 문

과학기술부 장관 귀하

본 보고서를 “지능형 통합비행제어시스템 설계기법 에 관한 연구” 의 보고서로 제출합니다.

2003 . 8 .

주관연구기관명 : 서울대학교
비행역학 및 제어 연구실

주관연구책임자 : 김 유 단

연 구 원 : 신 동 호 김 기 석
문 관 영 형 승 용
김 형 석 정 보 경
임 은 성 김 승 균
김 영 진 안 채 익
이 호 진 정 인 성
박 상 균 이 호 진

여 백

보고서 초록

과제관리번호	M1-0104-00-0028	해당단계 연구기간	2001. 7. 11 ~ 2003. 7. 10 (2 년)	단계 구분	(1단계) / (총 2단계)
연구사업명	중 사업명	특정연구개발사업			
	세부사업명	국가지정연구실사업			
연구과제명	중 과제명				
	세부(단위)과제명	지능형 통합 비행제어시스템 설계기법 연구			
연구책임자	김 유 단	해당단계 참여연구원수	총원: (14)명 박사(1), 석사(6), 학사(7) 총 : 명 내부: 명 외부: 명	해당단계 연구비	정부: 379,618천원 기업: 0 천원 계: 379,618천원
연구기관명 및 소속부서명	서울대학교 비행역학 및 제어 연구실		참여기업명		
국제공동연구	상대국명 :		상대국연구기관명 :		
위탁연구	연구기관명 :		연구책임자 :		
요약(연구결과를 중심으로 개조식 500자 이내)				보고서 면수	
<ul style="list-style-type: none"> ○ 하드웨어 다중화 관리기법에 관한 이론 연구 및 적용 가능성에 대한 검증 수행 ○ 3중 배치 센서에 대해 voting 알고리즘 구성 및 다중화 시스템 하드웨어 해법 연구 ○ 해석적 알고리즘을 구현하기 위한 필터 설계 기법 연구 ○ Mid-level 비행제어 알고리즘의 개발 ○ 신경회로망, 퍼지로지, 유전자 알고리즘을 이용한 지능형 제어기 설계 방법 연구 ○ 항공기의 생존성을 높이기 위한 고장허용 제어시스템 설계기법 연구 ○ 고장인식 과정이 필요 없는 직접 적응제어기법, 고장인식 과정을 포함한 슬라이딩 모드제어 기법, 입출력 정보를 이용한 시스템 인식기법을 사용한 기법 개발 ○ 제안된 알고리즘을 하드웨어적으로 검증하기 위한 MB(Moving Base)-HILS 시스템 구현 					
색인어 (각 5개 이상)	한 글	비행제어, 자동조종장치, 고장진단 및 제어, 재형상 제어, 디지털제어기술			
	영 어	Flight Control, Fault Diagnosis and Control, Reconfigurable Control, Digital Control Technique			

여 백

요 약 문

I. 제 목

지능형 통합 비행제어시스템 설계기법 연구

II. 연구개발의 목적 및 필요성

항공기가 보다 넓은 비행영역에서 개선된 비행성(Flying quality)를 갖도록 지능형 비행제어시스템을 구축하고, 더 나아가 이상현상(Malfunction)이 발생하였을 경우에 이를 검출 및 분리하고 적절히 대처할 수 있도록 제어법칙을 재설계(Reconfiguration)할 수 있는 통합 비행제어시스템을 구현한다.

III. 연구개발의 내용 및 범위

본 연구실에서는 지능형 통합 비행제어 시스템에 대한 연구를 효과적으로 수행하기 위해, 다음과 같은 세분화된 해당 연구분야를 연구 주제별로 통합하여 연구를 수행하였다. 본 연구의 1단계에서는 지능형 통합비행제어 시스템 구축에 주안점을 두고 연구를 추진하였으며, 추진된 연구 세부항목은 다음과 같다.

- Mid-Level 비행제어시스템 연구

지능형 통합 비행제어시스템을 위한 mid-level 비행제어 알고리즘 개발에 대한 연구를 수행하였다. 항공기의 비선형성, 공력계수의 불확실성, 외란 등 다양한 비행상태에 대한 비행체의 안정성 및 강건성을 보장하고, 성능을 향상시킬 수 있는 제어시스템을 설계하고, 고장이나 조종면 파손과 같은 상황에서도 비행임무를 수행하기 위하여 모드전환 비행제어기법 연구, 신경회로망/퍼지로지 지능형 비행제어 알고리즘 연구, 비선형 제어 이론 연구 등을 수행하였다.

- RM 시스템 하드웨어 해법 구현

시스템의 신뢰도를 향상시키기 위하여 다중화 관리기법(Redundancy Management, RM)을 도입하여 비행제어시스템을 개발하였다. 모듈별 다중화를 통하여 고장 발생에도 시스템이 영향을 받지 않도록 하였으며, 이를 관리하기 위한 알고리즘을 개발하였다. 이를 위하여 Sensor Suit 3중 RM 기법 개발, RM 시스템의 하드웨어 최적 알고리즘 개발, RM 시스템의 하드웨어 해법 구현 등의 연구를 수행하였다.

- RM 시스템의 해석적 알고리즘 개발

시스템의 신뢰성을 향상시키기 위하여 항공기에 사용되고 있는 하드웨어 여분 방식은 알고리즘이 간단하고 성능이 우수하지만, 비용과 중량이 많이 소요된다. 따라서 수학적 동특성과 확률특성에 의한 정보를 활용한 해석적 여분방식을 통하여 이를 보완할 필요가 있다. 본 연구에서는 해석적 알고리즘 구현을 위한 필터 설계, 알고리즘의 실시간 계산을 위한 수치기법 개발 등과 같은 해석적 여분방식에 대한 연구를 수행하였다.

- 6-DOF NASP를 이용한 제어 알고리즘 성능검증

본 연구에서 개발된 지능형 비행제어시스템의 성능과 타당성을 검증하기 위해 6자유도 비선형 시뮬레이션 프로그램을 개발하였다. 각기 다른 방식으로 설계된 비행제어 알고리즘을 동일한 시스템에 적용하여 성능평가를 수행함으로써, 개발된 비행제어 알고리즘의 타당성과 효율성을 파악할 수 있다. 이를 위해 제어 알고리즘 검증을 위한 NASP 인터페이스 구축, Mid-level 비행제어법칙 모듈별 검증, Mid-level 비행제어 알고리즘 개선 등을 수행하였다.

- RM 시스템 해석적 알고리즘과 하드웨어 해법과의 통합설계

다중화관리를 위한 하드웨어 기법과 해석적 기법을 통합하기 위한 연구를 수행하였다. 이는 하드웨어 여유 방식인 교차채널 방법의 한계를 극복하기 위한 알고리즘 연구이며, RM 시스템 하드웨어 해법 보완모델 설계, RM 시스템 보완모델의 해석적 알고리즘 개발, RM 시스템 통합설계 등을 수행하였다.

- FB(Fixed-Base) HILS를 통한 RM 시스템 성능검증

제어 알고리즘의 실시간 하드웨어 적용가능성과 효율성 분석, 다중화관리 시스템에 고장과 같은 이상현상이 발생하였을 경우 이를 하드웨어 해법과 해석적 알고리즘을 통해 지능적으로 시스템의 고장검출 및 분리수행이 정상적으로 수행되는가를 평가하기 위해서 6-DOF 비선형 시뮬레이션과 Fixed-Base(FB) HILS 시스템을 이용해야 한다. 본 과제에서 구축한 FB-HILS 시스템을 이용하여 다중화 비행제어시스템의 성능을 검증하였다. 하드웨어의 구조적 문제와 구동 소프트웨어, 제어 알고리즘, 고장진단 알고리즘 등을 단계적으로 검증하였으며 실시간 RM 시스템 성능, RM 시스템 하드웨어 해법, 해석적 알고리즘 필터, RM 시스템 통합기능 등을 FB-HILS 시스템을 통하여 검증하였다.

IV. 연구개발결과

- 고장대처 알고리즘 개발

비행체에 발생가능한 고장에 대한 사례연구를 통하여 고장에 취약한 비행체에 발생한 고장을 검출 분리하고, 탑승객의 안전을 보장할 수 있는 제어 알고리즘에 대한 연구를 수행하였다. 이러한 고장대처 알고리즘은 현재 우리나라 항공분야의 기술수준이 미흡한

첨단 핵심연구 분야로, 본 연구실의 연구성과는 이러한 연구의 선행연구로 매우 의미가 크다고 판단된다. 고장에 대처하는 지능형 통합 비행제어시스템은 고장을 인식함으로써 고장에 대처하는 기법과 발생한 고장의 종류와 관계없이 지능적인 구조로 제어기를 구축하는 기법으로 구분된다. 본 연구의 1단계에서는 비행체에 발생한 고장을 인식하여 고장 모드를 인식하고 그에 대처하는 지능형 제어 알고리즘에 대한 연구를 주제로 하였으며, 연구 수행 중 제어기의 재형상을 통해 고장에 강건한 지능형 비행제어 시스템에 대한 연구를 병행하였다. 재형상 제어에 의한 고장 대처기법은 2단계 연구내용에 포함되어 있는 것으로 사전에 고려되지 않은 고장에 대해서도 일정 범위 내에서 시스템의 안정성을 보장할 수 있다는 장점이 있다.

- 비행제어알고리즘 검증 시스템 구축

비행제어시스템 알고리즘의 타당성을 검증하기 위해 본 연구실에서 구축한 FB-HILS 시스템은 제어 알고리즘과 같은 소프트웨어 시스템뿐만 아니라, 하드웨어 모듈의 성능을 평가할 수 있는 제어시스템 검증 도구이다. 비교적 저가의 실험장비로 시스템을 구축하여 국내 대학의 관련 연구를 수행하는 연구실에서 쉽게 시스템을 구축할 수 있는 방안을 제시하여, 비행제어기법의 기술향상에 큰 파급효과를 줄 것으로 예상된다. 또한 FB-HILS 시스템의 구축을 통해 축적된 기술은 다음 단계 연구에서 수행될 MB-HILS 구현의 선행 연구자료로 사용될 예정이며, 그 활용가치가 매우 높을 것으로 판단된다.

- 다중화 시스템 구현

기존의 MIL-STD-1553B나 ARINC에 의하여 독점적으로 운영되던 항공기의 버스시스템은 높은 가격과 기술 의존도로 인하여 개발에 많은 제약이 따르고 있다. 특히, MIL-STD-1553 버스 등은 신뢰성 때문에 항공기에서 많이 사용하고 있지만, 과거의 기술을 사용하고 있으므로 본 연구에서 수행하고 있는 지능형 비행제어시스템과 같이 많은 전송량과 다수의 모듈을 필요로 하는 시스템에서는 속도 제약 고가의 시스템이라는 단점이 있다. 이에 본 연구에서는 CAN 버스를 이용하여 독자적인 비행제어시스템 버스를 제안하였고, 이를 이용하여 다중화 시스템을 개발하였다. 앞으로 버스 다중화 및 환경조건에 대한 강건성을 통하여 신뢰도 향상을 이루면 차세대 버스로서 다양한 수요가 창출될 수 있을 것으로 기대된다. 본 연구실에서 국내에는 아직 기초 수준에 머물러 있는 비행제어기술인 시스템 다중화를 고려한 비행제어시스템 구현을 위한 연구를 수행함으로써 해당 분야의 선도적 역할을 수행함은 물론, 선진국 대비 비교적 낙후되어 있는 비행제어시스템 설계 분야의 발전에 기여한 바가 크다고 판단된다. 특히, 센서를 비롯한 비행제어 컴퓨터 및 그에 수반되는 다양한 신호들의 다중화에 대한 해석적 연구와 실험 연구는 높은 신뢰성을 요구하는 타산업 종목의 기술발전에 파급효과도 예상된다.

V. 연구개발결과의 활용계획

항공우주선진국에서는 개발된 비행제어시스템 설계기술을 지능형 자동차의 항법 및 제어시스템 개발에 응용하고 있으며, 차세대 쾌속선박의 항법 및 운항제어시스템 개발에도 사용하고 있다. 본 연구의 수행결과인 통합 비행제어시스템 설계기술은 다양한 환경에 대해 지능적으로 대처하는 전체 시스템의 해결능력 배양이라는 측면에서 전 산업분야에 영향을 미칠 수 있다. 항공우주분야에서는 새로운 개념의 대형여객기, 고성능 전투기 등에 대한 설계가 매우 빠른 속도로 진행되어 개념설계 단계를 넘어서고 있다. 특히, 미국 NASA와 보잉에서는 날개를 회전하여 수직으로 이륙한 후에 회전하는 날개를 고정시켜 일반 항공기와 같이 비행하는 무인 시험비행기 X-50을 2003년 중반에 시험비행할 것을 계획하고 있다. 이 항공기에 탑재될 비행제어시스템은 이제까지의 어느 항공기보다 그 중요성이 강조되고 있으며, 고장이 발생하거나 이상상태 발생시에 대한 대비를 갖춘 제어시스템을 탑재해야 한다. 우리나라에서는 틸트로터 방식의 무인항공기를 개발하는 “스마트 무인기 기술개발” 사업이 과기부 프론티어 사업으로 선정되어, 앞으로 10년동안 이러한 개념의 항공기를 개발하기 위해 연구개발에 많은 노력을 기울이고 있다. 또한, 앞으로 20년 이내에 우리나라도 독자적인 기술로 F-16급 항공기 및 X-45급 자율무인항공기를 개발하고자 하고 있다. 따라서 현재 본 국가지정연구실에서 수행하고 있는 연구의 중요성이 더욱 부각되고 있는 실정이며, 본 과제를 수행함으로써 얻어지는 연구결과가 스마트 무인기 및 고성능항공기 비행제어시스템의 개발자료로 적극 활용될 수 있을 것으로 사료된다.

S U M M A R Y

I. TITLE

A Study on the Integrated and Intelligent Flight Control System

II. Objective and Requirement of the Research

This research is performed for the construction of the Integrated Flight Control System which can control the aircraft system intelligently over the full flight envelope. Fault detection, isolation and reconfiguration control system for various malfunctions are studied.

III. The Contents and Categories of the Research

This research is divided into several categories to improve the efficiency of the each subject. The first step of this research is concentrated on the construction of the structure of intelligent flight control system as follows.

– Mid–Level Flight Control System

Algorithm for designing the intelligent flight control system was developed. Developed algorithm should guarantee the stability and robustness of the system for the system nonlinearity, uncertainty of aerodynamic coefficient, and disturbance etc. The proposed algorithm can deal with the damaged situation such as the fault on the sensors and/or actuators. Mode switching control, intelligent neural network, fuzzy logic control algorithm and nonlinear control laws are studied for the purpose.

– Development of Redundancy Management System in Hardware Method

To improve the system reliability the redundant management system is introduced. In this study, the developed system is modularized in order not to be effected by the fault of other module and managed by developed program. Sensor suit triple RM technique and optimal RM algorithm were developed.

– Development of Redundancy Management System in analytic Method

Hardware redundancy method takes much cost and weight. To overcome these problems the analytic redundancy method utilizing the information of system dynamics and stochastic characteristics were proposed. In this research, real-time algorithms for designing filter for analytic method and numerical method were studied.

- Verification of the performance of the control system with 6DOF Nonlinear Aircraft Simulation Program

To verify the adequacy and performance of the developed control system NASP(Nonlinear Aircraft Simulation Program) was developed. Using NASP the performance of each module and Mid-level flight control algorithms were verified.

- Verification of the performance of the control system using Fixed Base(FB) Hardware_in_the_Loop Simulation(HILS) system.

HILS system is required for verifying the performance of the control logic in the real system including interfaces, and noisy environment. In this research the control algorithm, fault diagnosis algorithm, and RM system integration performance were tested step by step.

IV Result

- Development of Fault Diagnosis Algorithm

The research on reliable flight control system which can detect the fault and isolate the faulty module was performed. Developed redundancy management algorithm can be used for developing high performance reliable aircraft system.

- Construction of The Verification System of Flight Control System

FB-HILS system not only verifies the integrated control system but also tests the performance of each module. It is considered that the developed test system can be easily constructed to verify the flight control system at low cost.

- Construction of Multiple Redundancy Management System with CAN

BUS system for RM uses STD-MIL_1553 or ARINC. However this system takes high cost. In this study, triple CAN bus system is proposed with low cost and high reliability. The research on the redundancy management system can be used for developing advanced flight control system.

V. Application Plan of The Research Result

The design technology of the flight control system can be applied for intelligent automobile navigation and mobile robot system. The research on the fault tolerant control system can be used to design the system on the whole industrial area in that intelligent

system have the ability to determine the behavior for the various environment including malfunction and faults.

Especially, NASA and Boeing plan perform flight test of X-50 that takes off vertically with rotating wings and fly like the conventional aircraft. Intelligent flight control system which will be used for X-50 will play a dominant role than other airplane. In Korea, the 'Smart Unmanned Aerial Vehicle development project has been launched. Our researches on this sphere are expected to be embossed and may give the extraordinary progress in that area.

CONTENTS

CHAPTER 1. OUTLINE OF THE RESEARCH AND THE DEVELOPMENT	25
§ 1. Outline of the Research	25
§ 2. Objective of the Research	26
§ 3. Annual Plan for Research	26
CHAPTER 2. CURRENT SITUATION OF THE CONTROL SYSTEM TECHNOLOGY	27
§ 1. The Current Situation of the Foreign Technology	27
§ 2. The Current Situation of the Domestic Technology	28
§ 3. The Weakness of the Domestic Technology	28
CHAPTER 3. THE CONTENT OF THE RESEARCH	30
§ 1. Mid-level Flight Control Law	30
1. The Research of the Intelligent Flight Control System	30
A. Neural Network Control Law	30
B. Fuzzy Logic Control Law	34
C. Genetic Algorithm	38
2. 6DOF Nonlinear Simulation Program	44
A. Outline	44
B. Aircraft Model	45
C. Dynamics Modeling	46
D. External Force Modeling	50
E. Environment Modeling	56
F. NASP Construction Using Matlab Simulink	57
3. Mid-level Flight Control Algorithm	58
A. Mid-level Flight Control using NASP	58
B. System Linearization	59
4. Improvement of the Mid-level Flight Control Algorithm	66
A. Design of Aim Angle Guidance Law Using Lyapunov Theory	66
B. Evasive Guidance Law Using Artificial Intelligence	74
C. Optimum Flight Path Design Passing Through Waypoints	81
5. Study on the Fault Detection and Isolation Method	86
A. Software Approach(Model Based Method)	87
B. Hardware Approach(Signal Based Method)	114
6. Filter Design	115
A. Introduction of Kalman Filter	115
B. Simulation and Real Application	117
7. Development of Numerical Method for Real-time Computation	121
A. System ID Method Using Input-Output History	121

B. Reconfigurable Flight Control Using Model Following Approach	125
C. Model Following Reconfiguration with Estimator	127
D. Numerical Simulation	129
E. Summary	133
§ 2. Study on Fault Tolerance Flight Control Design	134
1. Concept of Fault Tolerance Flight Control System	134
A. Objectives and Classification	134
B. Fault Detection and Isolation	135
C. Hardware Based Redundancy Management	135
D. Analytical Method	136
2. Aircraft System Fault	138
A. Definition of Fault	138
B. Classification and Cause of Fault	138
C. Fault Detection and ID Methods	139
3. Nonlinear Control System Approach Using Neural Network	140
A. Introduction	140
B. Basic Concept of Neural Networks	141
C. System Model and Control Law Design	143
D. Stability Analysis	145
E. Numerical Example	147
F. Conclusion	150
4. Reconfigurable Control System Approach Using Adaptive Controller	151
A. Comparison of Reconfigurable Flight Control System Design Methods	151
B. System Dynamics	151
C. Control System Structure and Fault Modeling	152
D. Reconfigurable Control System Using Direct Method and Model Following Method	153
E. Simulation Result	157
5. Sliding Mode Based Reconfigurable Flight Control System	162
A. System Dynamics	162
B. Model Following Approach	162
C. Sliding Mode Approach	163
D. Actuator Fault Modeling and Reconfiguration	165
E. Simulation Result	166
6. Mode Change Flight Controller	168
A. Outline	168
B. Approaches	169
§ 3. Hardware Redundancy Management System	171
1. Outline of Flight Control System	171
A. Basic Concept	171
B. Function of Control System	172
C. Hardware Redundancy Management Method	173

2. Redundancy Management System	178
A. Contents of Redundancy Management System	178
3. Introduction of Flight Control System Module Processor, DSP(Digital Signal Processor)	180
4. Categorization of Functional Module	188
A. Sensor Input Part	189
B. Flight Control Part	190
C. Control Signal Generation Part	190
D. Bus Management Part	191
5. Data Bus System	191
A. Outline of CAN Bus System	191
B. Structure and Physical Protocol of CAN	192
C. Software Protocol	194
D. Bus Operation Method	195
E. Bus Fault	195
6. Redundancy Flight Control System Programming	196
A. Sensor Input Part Program	197
B. Flight Control Part Program	198
C. Sensor Signal Generation Part Program	200
7. Analytical Algorithm of Redundancy Compensation Model	201
A. Hardware Compensation Method Using Kalman Filter	202
B. Analytical Fault Detection Method Using Observer	203
C. Bank Fault Filter System	205
§ 4. Research on Integrated Intelligent Flight Control System	207
1. Outline of HILS System	207
A. Definition and Classification of Simulation	207
B. Necessity of HILS System	208
C. Contents	209
2. Test Based on HILS	215
A. Basic Mission Test	215
B. RM Verification with Hardware Approach	215
C. Analytical Redundancy Approach Using Kalman Filter	226
CHAPTER 4. PLAN ACHIEVEMENT AND CONTRIBUTION	231
§ 1. Achievement Comparing with the Plan	231
§ 2. Research Result of the 1st Phase	232
CHAPTER 5. PRACTICAL USE OF THE PROPOSED RESEARCH	234
CHAPTER 6. REFERENCE	236

목 차

초 록	3
요 약 문	5
SUMMARY	9
CONTENTS	12
목 차	15
제 1 장. 연구개발과제의 개요	25
제 1 절 연구개요	25
제 2 절 연구 목표	26
제 3 절 연차별 연구목표 및 내용	26
제 2 장 국내외 기술개발 현황	27
제 1 절 국외의 연구 개발 내용 및 결과	27
제 2 절 국내의 연구개발 내용 및 결과	28
제 3 절 현기술상태의 취약성	28
제 3 장 연구개발수행 내용 및 결과	30
제 1 절 Mid-level 비행제어 법칙 연구	30
1. 지능형 비행제어 시스템 연구 개요	30
가. 신경회로망 유도법칙	30
나. 퍼지 논리 유도법칙	34
다. 유전자 알고리즘	38
2. 6자유도 비선형 시뮬레이션 프로그램	44
가. 개요	44
나. 항공기 모델	45
다. 운동학 모델링 및 구현	46
라. 외력 모델링 및 구현	50
마. 외부환경 모델	56
바. Matlab Simulink를 이용한 NASP 구현	57

3. Mid-level 비행제어 알고리즘 연구	58
가. NASP를 이용한 Mid-level 비행제어연구	58
나. 시스템 선형화	59
4 Mid-level 비행제어법칙 알고리즘 개선	66
가. 르야프노프 이론을 이용한 목표각 추종 유도법칙 설계	66
나. 지능형 제어기를 이용한 항공기의 회피 유도법칙	74
다. 경로점을 통과하는 최적 비행궤적 설계	81
5. 고장 검출 및 인식 (FDI) 기법 연구	86
가. 소프트웨어 기법(모델기반 방식)	87
나. 하드웨어 기법(신호기반 방식)	104
6. 필터 설계 기법 연구	115
가. Kalman Filter의 유도과정	115
나. 시뮬레이션과 실제 적용	117
7. 알고리즘의 실시간 계산을 위한 수치기법 개발	121
가. 입출력 궤적정보를 이용한 시스템 식별기법	121
나. 모델 추종제어기법을 이용한 재형상 비행제어	125
다. 추정기를 가진 모델추종 재형상 제어시스템의 설계	127
라. 수치 시뮬레이션	129
마. 연구요약	133
제 2 절 고장허용 비행제어 설계기법 연구	134
1. 고장허용 비행제어시스템 연구개요	134
가. 연구 범위 및 분류	134
나. 고장검출 및 분리	135
다. 다중화관리 하드웨어 해법	135
라. 해석적 방법	136
2. 비행체 고장의 개요 및 특성	138
가. 고장의 정의	138
나. 고장의 분류 및 고장 요인	138
다. 고장 검출 및 인식 기법의 분류	139
3. 신경회로망을 이용한 비선형 제어시스템 설계기법 연구	140
가. 서론	140
나. 신경회로망의 구조와 기본적인 성질	141
다. 항공기 시스템 모델 및 제어기 설계	143
라. 안정성 해석	145
마. 수치예제	147
바. 결론	150
4. 적응제어기법을 이용한 재형상 제어시스템 설계기법 연구	151
가. 적응기법을 이용한 재형상 비행제어 시스템 설계기법 비교	151
나. 운동방정식	151

다. 제어시스템 구조 및 고장 모델링	152
라. 직접법 및 모델추종기법을 이용한 재형상 비행제어시스템 설계	153
마. 수치 시뮬레이션 및 결과	157
5. 슬라이딩 모드기법 기반 재형상 제어시스템 설계기법 연구	162
가. 시스템 운동방정식	162
나. 모델 추종제어 기법의 적용	162
다. Sliding Mode기법 적용	163
라. 조종면 파손 모델링 및 재형상 제어	165
마. 시뮬레이션	166
6. 모드전환 비행제어 기법 연구	168
가. 개요	168
나. 연구방법	169
제 3 절. 하드웨어 다중화 관리 시스템 연구	171
1. 비행제어 시스템 개요	171
가. 개요	171
나. 제어 시스템의 기능	172
다. 하드웨어 다중화 관리 기법	173
2. 다중화 관리 시스템 구성	178
가. RM 시스템 구성	178
3. 비행제어시스템 모듈 프로세서 DSP(Digital Signal Processor)의 소개	180
4. 기능별 모듈의 종류	188
가. 센서입력부	189
나. 비행제어부	190
다. 제어신호생성부	190
라. 버스 관리 부	191
5. 데이터 버스 시스템	191
가. CAN의 개요	191
나. CAN의 구조 및 물리적 프로토콜	192
다. 소프트웨어 프로토콜 (요소기술 및 구현기술)	194
라. 버스 운영 방법	195
마. BUS FAULT	195
6. 다중 비행제어시스템 프로그래밍	196
가. 센서입력부 프로그램	197
나. 비행제어부 프로그램	198
다. 센서신호발생부	200
7. 다중화 시스템 보완모델의 해석적 알고리즘 개발	201
가. 칼만 필터를 이용한 하드웨어 보완 방식	202
나. 관측기를 이용한 해석적 고장 진단 방법	203
다. 뱅크 고장 필터 시스템	205

제 4 절. 지능형 통합 비행제어 시스템 설계 기법 연구	207
1. HILS 시스템 개요	207
가. 시뮬레이션의 정의 및 종류	207
나. HILS의 필요성	208
다. 구성	209
2. HILS 시스템을 이용한 실험 연구	215
가. 기본 임무수행 테스트	215
나. HARDWARE 해법 RM 검증	215
다. 칼만필터를 사용한 해석적 여분 방식	226
제 4 장 목표달성도 및 관련분야에의 기여도	231
제 1 절 계획대비 달성도	231
제 2 절 1단계 연구 결과	232
제 5 장 연구개발결과의 활용계획	234
제 6 장 참 고 문 헌	236

<표 차례>

표 2.3.1 우리나라의 항공우주 기술수준	28
표 3.1.1 유전자 알고리즘의 주요 용어 설명	41
표 3.1.2 시뮬레이션 시나리오	71
표 3.1.3 평균 잡음 크기	72
표 3.1.4 시뮬레이션 결과	73
표 3.1.5 퍼지 규칙 베이스	78
표 3.1.6 회피자의 초기값	79
표 3.1.7 빗맞힘 거리와 최종 시간	81
표 3.1.8 경로점 및 최적 결과 (단위:km)	85
표 3.1.9 퍼지규칙	100
표 3.1.10 하드웨어 기법의 분류	104
표 3.1.11 고장 상황의 예	107
표 3.1.12 패리티 벡터의 표현	109
표 3.1.13 패리티 식	113
표 3.1.14 더치롤 기동의 고유치	129
표 3.2.1 제어면 운용범위	148

표 3.3.1 비행제어시스템 기본기능	172
표 3.3.2 센서사양	198
표 3.3.3 센서 문턱값 및 한계시간	199
표 3.4.1 레이트 테이블 사양	213
표 3.4.2 고장 발생 시나리오	216
표 3.4.3 자이로에 대한 고장 검출 결과	225
표 3.4.4 고장 선언 결과	226

<그림 차례>

그림 3.1.1 신경회로망 구조	31
그림 3.1.2 퍼지추론 과정	38
그림 3.1.3 최적화 알고리즘의 계보	39
그림 3.1.4 유전자 알고리즘의 흐름도	43
그림 3.1.5 비선형 시뮬레이터 구조	44
그림 3.1.6 비선형 6자유도 시뮬레이터	45
그림 3.1.7 F-16 모델 물리적 변수	46
그림 3.1.8 받음각과 옆미끄럼각	46
그림 3.1.9 병진방향 동역학 시물링크 모델	47
그림 3.1.10 회전방향 동역학 시물링크 모델	48
그림 3.1.11 병진방향 기구학 시물링크 모델	48
그림 3.1.12 회전방향 기구학 시물링크 모델	49
그림 3.1.13 병진방향 시물링크 모델	49
그림 3.1.14 회전진방향 시물링크 모델	49
그림 3.1.15 NASP 동역학 시물링크 모델	50
그림 3.1.16 비행체에 작용하는 외력	50
그림 3.1.17 공력계수 구성(C_x)	51
그림 3.1.18 공력계수 구성(C_y)	51
그림 3.1.19 공력계수 구성(C_z)	51
그림 3.1.20 공력계수 구성(C_l)	51
그림 3.1.21 공력계수 구성(C_m)	52
그림 3.1.22 공력계수 구성(C_n)	52
그림 3.1.23 공력계수 구성(병진방향)	52
그림 3.1.24 공력계수 구성(회전방향)	52
그림 3.1.25 공력계수 유차원화(병진방향)	53
그림 3.1.26 공력계수 유차원화(회전방향)	53
그림 3.1.27 공력 모델링	53
그림 3.1.28 추력계수 모델	53
그림 3.1.29 엔진에 의한 추력 모델	54
그림 3.1.30 엔진에 의한 모멘트 모델	54
그림 3.1.31 엔진에 의한 외력 모델링	55

그림 3.1.32	중력에 의한 외력 모델링	55
그림 3.1.33	외력 모델링	56
그림 3.1.34	대기온도 모델	57
그림 3.1.35	외부환경 모델링	57
그림 3.1.36	입출력 관계	58
그림 3.1.37	Simulink를 이용한 NASP	58
그림 3.1.38	선정된 정상수평비행 트림 점	59
그림 3.1.39	수치구배기법의 개념	61
그림 3.1.40	상태변수 피드백을 걸지 않았을 경우의 시스템 궤적	65
그림 3.1.41	상태변수 피드백을 걸었을 경우의 시스템 궤적	65
그림 3.1.42	2차원 추적상황	66
그림 3.1.43	충돌삼각형과 목표각	67
그림 3.1.44	미사일과 표적의 궤적 (시나리오 1, 2, 3, 11)	74
그림 3.1.45	가속도 명령	74
그림 3.1.46	가속도 명령 (시나리오 7, 8, 12)	74
그림 3.1.47	추적자와 회피자의 형상	75
그림 3.1.48	퍼지 소속함수	78
그림 3.1.49	퍼지 곡면	79
그림 3.1.50	교전 1의 비행 궤적	80
그림 3.1.51	교전 7의 비행 궤적	80
그림 3.1.52	3차원 좌표계	82
그림 3.1.53	로절 영역 부등식 구속조건	83
그림 3.1.54	종말시간 정규화 개념도	84
그림 3.1.55	시간 종속 매개변수 선정	84
그림 3.1.56	입력 매개변수 초기화	85
그림 3.1.57	초기값에 의한 비행궤적	85
그림 3.1.58	최적 입력 매개변수	86
그림 3.1.59	생성된 최적 비행궤적	86
그림 3.1.60	FDI 기법의 분류	86
그림 3.1.61	소프트웨어 FDI 기법	87
그림 3.1.62	소프트웨어 기법 시스템 블록선도	87
그림 3.1.63	잔차 생성 기법의 블록선도	88
그림 3.1.64	관측기 기반 기법의 블록선도	88
그림 3.1.65	고장 신호	90
그림 3.1.66	센서 Fault Cases	90
그림 3.1.67	고장의 형상	91
그림 3.1.68	외란시 고장에 따른 출력	91
그림 3.1.69	고장유무에 따른 추정오차	91
그림 3.1.70	고장이 있을 경우의 잔차	91
그림 3.1.71	구조화된 잔차 생성 기법	92

그림 3.1.72	노이즈의 영향과 plant fault	94
그림 3.1.73	노이즈를 상쇄한 잔차	95
그림 3.1.74	plant fault 가 존재하지 않는 경우	95
그림 3.1.75	MA parity equation(with fault and noise)	95
그림 3.1.76	MA patity equation(with fault and noise)	96
그림 3.1.77	MA parity equation(with fault)	96
그림 3.1.78	MA parity equation(No fault)	96
그림 3.1.79	고장 출력 공간	97
그림 3.1.80	매개변수 인식 방법	98
그림 3.1.81	고장 검출 및 진단 알고리즘	99
그림 3.1.82	퍼지 시스템	99
그림 3.1.83	퍼지 시스템의 기본 구조	100
그림 3.1.84	퍼지출력	101
그림 3.1.85	고장에 대한 고장검출 신호	101
그림 3.1.86	검출된 고장 모드	101
그림 3.1.87	고장 발생 시 각도 상태변수 시간반응	102
그림 3.1.88	고장 발생 시 각속도 상태변수 시간반응	103
그림 3.1.89	Quadraplex System Architecture	105
그림 3.1.90	4중 센서의 고장 검출 알고리즘 - 첫 번째 고장 발생	105
그림 3.1.91	3중 센서의 고장 검출 알고리즘 - 두 번째 고장 발생	106
그림 3.1.92	신호의 패턴을 이용한 고장의 분류	106
그림 3.1.93	고장 신호의 예	108
그림 3.1.94	Block Diagram Depicting System, Measurement and Estimator	118
그림 3.1.95	Simulink 상에서의 시스템과 추정기의 관계	118
그림 3.1.96	System 내부구성	119
그림 3.1.97	Estimator의 구성	119
그림 3.1.98	Gain Matrix의 내부구성	119
그림 3.1.99	각도 측정치	120
그림 3.1.100	잔차	120
그림 3.1.101	필터 시스템의 구성과 역할	121
그림 3.1.102	모델추종기법을 이용한 재형상 제어기	126
그림 3.1.103	추정기를 가진 모델추종 재형상 제어기	129
그림 3.1.104	Responses of dutch roll maneuver	130
그림 3.1.105	Responses of roll maneuver	130
그림 3.1.106	Aileron Responses of roll maneuver(estimator included)	130
그림 3.2.1	고장허용제어의 영역	134
그림 3.2.2	고장허용 제어기법	135
그림 3.2.3	해석적 여분을 이용한 고장검출	136
그림 3.2.4	고장허용제어 분류 및 개괄도	137
그림 3.2.5	항공기 사고 발생 요인별 분석	138

그림 3.2.6 항공기 임무(mission) 수행 중 고장 요인	139
그림 3.2.7 FDI 기법의 분류	139
그림 3.2.8 신경회로망의 구조	142
그림 3.2.9 제안된 제어기의 구조	145
그림 3.2.10 상태변수 시간 반응	149
그림 3.2.11 제어변수 시간반응	149
그림 3.2.12 신경회로망 가중치 Frobenius Norm의 시간반응	150
그림 3.2.13 연속적 구조의 항공기 제어 시스템	152
그림 3.2.14 고장모델링	153
그림 3.2.15 시뮬레이션 시나리오	158
그림 3.2.16 Roll rate response	159
그림 3.2.17 Pitch rate response	159
그림 3.2.18 Yaw rate response	159
그림 3.2.19 Roll angle response	160
그림 3.2.20 Angle of attack response	160
그림 3.2.21 Sideslip angle response	160
그림 3.2.22 Aileron command history	161
그림 3.2.23 Elevator command history	161
그림 3.2.24 Rudder command history	161
그림 3.2.25 상수입력 형태의 고장	167
그림 3.2.26 검출된 고장의 형상	167
그림 3.2.27 수행한 ϕ	168
그림 3.2.28 RCS 수행한 β	168
그림 3.2.29 sin입력 형태의 고장	168
그림 3.2.30 검출된 고장의 형상	168
그림 3.2.31 RCS 수행한 ϕ	168
그림 3.2.32 RCS 수행한 β	168
그림 3.2.33 개인 스케줄링 개략도	169
그림 3.2.34 예측제어의 블록선도	169
그림 3.2.35 단일 모델과 다중모델의 적응방식 비교	170
그림 3.2.36 IDCL/MPC 통합제어기	170
그림 3.3.1 다중화 관리시스템의 데이터 흐름도	174
그림 3.3.2 4중 다중화 관리 시스템에서의 투표 과정	176
그림 3.3.3 센서신호와 기준신호 사이의 차이	177
그림 3.3.4 정규분포 곡선과 막대분포도	177
그림 3.3.5 정상신호 유형과 유효시간 결정	178
그림 3.3.6 비행제어시스템의 다중 구조	179
그림 3.3.7 DSP 플랫폼의 특징	181
그림 3.3.8 DSP 내부 구조 개념도	182
그림 3.3.9 DSP의 Pin 배열	183

그림 3.3.10 DSP 내부 기능 블록 다이어그램	183
그림 3.3.11 TMS320LF2407A 내부 메모리 맵	184
그림 3.3.12 비행제어컴퓨터 설계 개념도	185
그림 3.3.13 SCSR1 레지스터	186
그림 3.3.14 PLL 클럭 선택 방법	186
그림 3.3.15 채널 A-D 선택 방법	186
그림 3.3.16 XR16C854 블록 다이어그램	187
그림 3.3.17 실제 구현된 비행제어컴퓨터	188
그림 3.3.18 센서입력부 인터페이스	190
그림 3.3.19 서보모터의 PWM 방식의 신호처리	191
그림 3.3.20 OSI 계층도	193
그림 3.3.21 CAN의 데이터 프레임	193
그림 3.3.22 3중화 버스	196
그림 3.3.23 다중 비행제어시스템 알고리즘	197
그림 3.3.24 센서입력부 프로그램	197
그림 3.3.25 비행제어부 프로그래밍	200
그림 3.3.26 제어신호생성부 프로그래밍	201
그림 3.3.27 2개의 센서만을 사용하였을 때 고장발생 상황	202
그림 3.3.28 관측기를 이용한 고장진단기	204
그림 3.3.29 고장 센서 신호	205
그림 3.3.30 잔차 생성 고장 감지	205
그림 3.3.31 하드웨어 여분과 해석적 여분방식의 통합	206
그림 3.4.1 FB-HILS 구성도	209
그림 3.4.2 센서 인터페이스 모듈	212
그림 3.4.3 Open GL Display 모듈	213
그림 3.4.4 라이트 테이블	214
그림 3.4.5 기본 임무 수행 테스트	215
그림 3.4.6 자이로 센서 신호분포도	216
그림 3.4.7 자이로 센서 확률분포도	216
그림 3.4.8 가속도계 센서의 신호분포도	217
그림 3.4.9 가속도계 센서의 확률분포도	217
그림 3.4.10 자이로 센서의 유효시간	218
그림 3.4.11 가속도계 센서의 유효시간	218
그림 3.4.12 1회 온도 : 24.8	223
그림 3.4.13 2회 온도 : 25.1	223
그림 3.4.14 3회 온도 : 25.1	223
그림 3.4.15 4회 온도 : 26.8	223
그림 3.4.16 5회 온도 : 26.6	223
그림 3.4.17 실험 배선도	224
그림 3.4.18 실험 장치	224

그림 3.4.19 운용 개념도	226
그림 3.4.20 고장진단 실험장치	226
그림 3.4.21 비행선 진행 궤적	228
그림 3.4.22 상태변수	228
그림 3.4.23 자이로 센서의 신호와 기준값	229
그림 3.4.24 고장센서 채널의 상태변수 추정치	229
그림 3.4.25 정상센서 채널의 상태변수 추정치	230
그림 3.4.26 각속도 센서1과 각속도 센서2 의 잔차 생성치	230

제 1 장. 연구개발과제의 개요

제 1 절 연구개요

지능형 통합 비행제어시스템에 관한 연구는 비교적 최근에 와서 연구가 활발히 이루어지고 있는 분야로 현재 그 개념이 정립되고 있는 단계라 할 수 있다. 통합 비행제어시스템의 출현은 최근 복잡하고 다양하게 바뀌고 있는 항공기의 임무범위 확대와 밀접한 관계가 있다. 특히, 현대의 고성능 항공기의 임무는 매우 구체적으로 제시되며, 신뢰성 있는 임무수행을 보장할 수 있는 비행제어시스템을 요구하게 되었다. 기존의 고전적 의미의 비행제어시스템으로서 이러한 요구조건을 만족하기 어렵게 되었고, 세밀한 임무의 적용을 위해서는 통합 비행제어시스템 설계기술이 필요하게 되었다.

통합 비행제어시스템에서의 주요 연구주제는 다양한 비행 상황변화에 대해 적극적으로 대처하여 임무수행 능력을 높이고, 임무수행이 불가능할 시에도 차선의 선택을 할 수 있는 비행제어시스템의 설계라고 할 수 있다. 하드웨어적인 면에서 보면 고신뢰성을 위해 충분한 여유분(redundancy)을 가져야 하고, 이에 따라 복잡해진 하드웨어를 관리하고 판단할 수 있는 소프트웨어의 개발이 요구되었다. 또한, 다양한 상황변화에서도 임무수행을 하기 위해서는 현재의 비행 상황을 인식하고 비상사태에 대비할 수 있는 적절한 알고리즘의 개발도 필요하게 되었다. 이러한 요구조건으로부터 전통적인 비행제어시스템은 지능적이며 통합적인 현대의 설계기법으로 진화하게 되었다.

지난 20여 년간 우리나라는 여러 가지 형식으로 항공우주 선진국들과의 교류를 통해 항공기 생산 및 설계기술을 습득하였고, 일부 분야에서는 괄목할만한 성장을 이루었다. 그러나 비행제어시스템 분야에 있어서는 항공우주 선진국들의 기술독점과 기술이전 회피로 인하여 아직 국내의 비행제어 시스템 설계기술은 타 분야에 비해 상당히 낙후되어 있는 실정이다. 앞에서 설명하였듯이 항공우주 선진국의 비행제어시스템 설계기술은 전통적 영역을 넘어서 통합 설계의 개념으로 진보하고 있는 상태에 있으며, 비행제어시스템의 기술 발전은 항공운항에 있어서 필수적인 항공기의 안전성을 증대를 가져온다. 통합 비행제어시스템 설계기술은 고성능 항공기 개발과 Fly-By-Wire 또는 Fly-By-Light 같은 개념의 항공기 시스템에 있어서 핵심기술로 자리매김하고 있고, 통합 비행제어시스템 설계 기술의 파급효과는 매우 커서 로켓 시스템, 미사일 발사 시스템, 항법 및 교통관리 시스템, 자율이동로봇 제어시스템, 쾌속 선박 운항 시스템에 널리 응용될 수 있다. 이러한 핵심기술에 대한 미흡한 국내 실정을 하루 빨리 보완하지 않으면 계속적으로 선진국에 종속적인 관계를 유지하게 될 수 밖에 없을 것이다.

통합 비행제어시스템에서는 비행제어시스템을 계층적으로 파악한다. 즉, 특정한 설계 점에서의 제어기 설계, 안정성 해석, 센서 및 구동기의 하드웨어 신호처리 등 고전적 의미의 제어시스템 설계에 해당하는 하위계층 제어시스템(low-level control system), 제어시스템을 통합하고 관리하며 필요에 따라 비행모드의 전환하는 중간계층 제어시스템(mid-level control system), 그리고 비행상황 및 임무에 따라 적절한 판단을 하고 모든 제어시스템을 통제하는 상위계층 제어시스템(high-level control system)으로 통합 비행제어 시스템은 구성된다. 서울대학교 비행역학 및 제어 연구실은 지난 2001년 7월 국가지정연구실로 선정되어 통합 비행제어시스템의 설계 기술에 관한 연구를 수행하고 있다. 연구의 1단계에서

는 하위계층 제어시스템에 대한 이해를 바탕으로 효율적인 중간계층 제어시스템 설계기술을 확보하는데 연구의 중점을 두고 있다. 그리고 2단계에서는 상위계층 제어시스템의 설계기술을 확보하여, 독자적으로 설계 개발한 HILS 시스템에서 통합 비행제어시스템을 구현하는 것을 연구목표로 하고 있다.

제 2 절. 연구 목표

항공기가 보다 넓은 비행영역에서 개선된 비행성(Flying quality)를 갖도록 지능형 비행 제어시스템을 구축하고, 더 나아가 이상현상(Malfunction)이 발생하였을 경우에 이를 검출 및 분리하고 적절히 대처할 수 있도록 제어법칙을 재설계(Reconfiguration)할 수 있는 통합 비행제어시스템을 구현한다.

제 3 절 연차별 연구목표 및 내용

구분	연구목표	연구내용
1차년도 (’01)	Mid-level 비행제어 설계기법 연구	모드전환(Mode Switching) 비행제어기법 연구 비행제어시스템을 위한 신경회로망/퍼지이론의 연구 비선형 비행제어이론 연구
	Fixed-base HILS 시스템 구축	RM 비행제어시스템을 위한 제어신호 프로토콜 구현 DSP Board를 이용한 시스템 구축
	RM 시스템의 하드웨어 해법 구현	Sensor Suite 3중/4중 RM 기법 개발 RM 시스템의 하드웨어 최적 알고리즘 개발 RM 시스템의 하드웨어 해법 구현
2차년도 (’02)	RM 시스템의 해석적 알고리즘 개발	모드별 해석적 알고리즘 개발 해석적 알고리즘 구현을 위한 필터 설계 알고리즘의 실시간 계산을 위한 수치기법 개발
	6-DOF NASP를 이용한 제어 알고리즘 성능검증	제어 알고리즘 검증을 위한 NASP 인터페이스 구축 Mid-level 비행제어법칙 모듈별 검증 Mid-level 비행제어법칙 알고리즘 개선
	RM 시스템 해석적 알고리즘과 하드웨어 해법과의 통합설계	RM 시스템 하드웨어 해법 보안모델 설계 RM 시스템 보안모델의 해석적 알고리즘 개발 RM 시스템 통합설계
	FB(Fixed-base) HILS를 통한 RM시스템 성능검증	RM 시스템 하드웨어 해법 검증 해석적 알고리즘 필터의 FB-HILS를 통한 검증 실시간 RM 시스템 성능검증 FB-HILS 시스템을 통한 RM 시스템 통합기능 검증

제 2 장 국내외 기술개발 현황

제 1 절 국외의 연구 개발 내용 및 결과

최근에 항공우주선진국에서 개발하고 있는 고성능 항공기는 대부분 비행조종 컴퓨터를 탑재한 전자식 비행관리시스템을 채택하고 있다.[5] 이러한 비행제어시스템은 미국의 NASA를 중심으로 X-29, X-31과 같은 시험기에 탑재하여 기술적 타당성을 검증하였고, 기존의 항공기를 개조한 YF-16, AFTI/F-16, F/A-18E/F Super Hornet, F/A HARV 등을 이용한 다양한 시험비행을 통하여 실제 시스템에 적용 가능성을 검증하였다. 이렇게 검증된 비행제어시스템 기술은 F-22, Eurofighter 2000, 러시아의 Su-37 등으로 실용화되고 있다. 선진국에서 개발되고 있는 고성능 전투기, 중대형 여객기, 소형 Business Jet 등과 같은 고성능 항공기에는 디지털 FBW(Fly-By-Wire) 또는 광신호에 의해 작동되는 FBL(Fly-By-Light)를 채용하여, 원하는 비행성능을 발휘하기 위해서 비행조종 컴퓨터가 감지기로부터 얻은 신호를 이용하여 직접 각종 작동기에 명령을 내릴 수 있게 되었으므로 항공기의 조종성능이 획기적으로 개선되었다.[6-10] 특히, 조종사의 수동조작에 의해서는 근본적으로 불가능한 복잡한 조종문제를 디지털 컴퓨터를 통해 해결하여, 항공기 이착륙 비행을 용이하게 할 뿐만 아니라, 안전성 향상, 탑승감 개선 등 기능면에서도 비약적인 발전을 가능하게 하고 있다. 전투기의 경우 디지털 컴퓨터의 도움을 받아 임무를 수행하는 통합 비행제어시스템 기술이 응용되고 있는데, 이 시스템은 조종시스템과 화기관제시스템, 추진시스템 등 모든 시스템을 통합제어하여 성능을 최대로 향상시키고 있다. 맥도널더글러스사와 NASA가 공동으로 설계하여 F-15에 장착한 시험용 자기진단 비행제어시스템은 항공기 조종면에 부분적인 고장이 일어났을 때에도 소프트웨어에 의해 고장발생 여부를 판단하여, 항공기가 정상적으로 비행할 수 있도록 비행제어시스템을 자동적으로 재편성하는 기능을 가지고 있다. 이 시스템 개발을 위해서 신경회로망 이론을 적용하여 파손된 항공기를 실시간으로 진단하며 제어시스템을 학습시키는 지능형 통합 비행제어시스템을 개발하였으며, 1999년에 보잉사에서 개조한 F-15를 이용하여 고장이 발생하였을 경우에도 안전하게 비행하여 착륙할 수 있도록 하는 지능형 비행제어시스템을 탑재한 첫 비행시험에 성공하였다. 한편, 유럽에서 개발하고 있는 Eurofighter 2000의 조종계통은 4중 FBW로 되어 있으며, 조종시스템이 고도로 자동화되어 있고 조종계통의 자기진단이 가능하다. 최근 개발되고 있는 여객기들도 기본적 기능의 다중화 된 비행관리시스템을 장착하고 있다.

최근에는 NASA, 보잉 등과 같은 산업체 및 연구소에서 MIT, 조지아 공과대학, 스탠포드 대학 등과 밀접한 관계를 가지며, 변화하는 비행환경이나 고장이 발생할 때에도 항공기가 최적의 조건으로 정상운항이 가능하도록 하기 위한 공동연구를 수행하고 있다. 이를 위해 비행제어시스템에 시스템 식별기법을 통합 적용하거나, 강건제어이론, 신경회로망 이론, 적응제어이론 등 최신 제어이론을 이용하여 비행제어시스템을 개선하고자 하는 연구를 활발히 수행하고 있다. 특히, 조지아 공과대학에서는 파손과 같은 고장에 적응할 수 있도록 신경회로망에 기반을 둔 재형상(reconfigurable) 통합 비행제어시스템을 연구하고 있는데, 2000년에는 이러한 연구결과를 응용하여 NASA Dryden 연구소, NASA Ames 연구소, 보잉사, 미공군(USAF)과 공동으로 수직꼬리날개가 없는(tailless) 형상의 무인시험기 X-36에 적용한 비행시험에 성공하였다.[22-26]

제 2 절 국내의 연구개발 내용 및 결과

국내의 연구소에서는 국방과학연구소와 한국항공우주연구원에서 복합재 항공기, 초등훈련기, 그리고 과학로켓 개발을 통해서 비행제어시스템에 대한 연구를 수행한 바 있다. 이들 비행체는 비행에 가장 필수적인 최소한의 제어시스템을 탑재한 비행체로, 이들을 개발하는 과정에서 비행제어시스템 설계기술에 관련한 잠재력을 어느 정도 확보하였다고 할 수 있다. 그러나 고기동 특성을 가지며 정밀한 임무를 수행할 수 있는 비행체에 요구되는 고성능 비행제어시스템 설계기술의 축적은 아직 미약한 수준이며, 이러한 기술의 개발이 필수적이라는 것은 절실히 느끼고 있는 실정이다. 국내 항공우주산업체의 경우에는 소형항공기, 훈련기, 전투기, 헬기를 면허생산, 공동생산 또는 자체 개발한 경험이 있으나, 역시 기본형 비행제어시스템을 장착한 항공기를 면허생산 하였거나, 해외협력기관에서 블랙박스 형태로 제공된 비행제어시스템을 장착하였으므로 관련된 기술축적이 미약하다고 할 수 있다. 대학의 경우 서울대학교, 인하대학교, 한국과학기술원 등에서 비행제어시스템 설계에 대한 기초 연구를 활발히 수행하고 있으며, 국제 전문학술지에 관련 연구결과를 발표하는 등 기초연구 수준은 상당부분 선진국 수준에 이르렀다고 판단된다. 그러나 실제 비행체를 대상으로 비행제어시스템을 설계하고 하드웨어 시스템이나 비행시험을 통해 설계한 시스템의 성능을 검증한 경험이 적다. 따라서 학계가 보유하고 있는 기초기술 능력을 국내 연구소, 산업체와 함께 실제 항공기나 로켓의 비행제어시스템 설계에 응용하기 위해서는 보다 많은 노력이 필요하다고 사료된다.

제 3 절 현기술상태의 취약성

표 2.3.1 우리나라의 항공우주 기술수준

분 류		선 진 국				한 국	분 류	선 진 국			한 국
대 분 류	중 분 류	미국	유럽	일본	미국			유럽	일본		
완 제 기	고정익기	100	90	70	40	로켓설계 핵심기반 기술	100	80	70	30	
	회전익기	100	80	70	30		우주조립시험 기술	100	70	60	30
	무인항공기	100	70	60	30			로켓 완제시스템 기술	100	95	80
추 진 기 관	엔진시스템	100	95	80	30	로켓 구조체 기술	100		95	90	40
	압축기/터빈	100	95	90	40		로켓 추진기관 기술		100	95	80
	연소기	100	95	80	30			로켓 자세제어 기술	100	90	70
기 계 보 기	주변기기	100	90	70	30	로켓 발사판제장치 기술	100		95	90	40
	착륙장치	100	100	80	50		[자료] 2010년을 향한 과학기술 발전 장기계획, 대형복합기술부문, 항공기술분야 (요약), 항공우주, 통권 제17호, 1994년 6월호, 한국항공우주산업진흥협회. 우주기술분야 (요약), 항공우주, 통권 제18호, 1994년 7월호, 한국항공우주산업진흥협회.				
	유압시스템	100	95	90	40						
	여압시스템	100	95	80	30						
항 공 전 자	FBL	100	90	70	30						
	컴퓨터	100	90	80	30						
	전자식 지시계기	100	90	90	20						
핵심요소기술	자동제어계	100	95	95	30						
	주변기기	100	90	80	75						
	공기역학	100	90	80	50						
	구조역학	100	95	90	50						
	추진기관	100	90	80	50						
	비행제어	100	95	90	40						
시험검사	100	95	80	40							

항공우주선진국의 경우 비행제어시스템 기술은 상당한 수준에 이르러 있다. 그러나 항공기 핵심부품에 이상현상이 발생했을 때 신속하고 효율적으로 대처하기 위해서는 실시간으로 시스템을 식별하고, 또한 실시간으로 최적의 비행제어시스템을 설계할 수 있는 기술수준이 요구된다. 이러한 실시간 기능을 구현하기 위해서는 빠른 속도로 안정되게 계산할 수 있는 알고리즘의 개발이 요구되고 있으며, 이를 위해서 선진국에서도 효율적이고 체계적으로 관리할 수 있는 통합 비행제어시스템을 개발하고자 하는 노력을 기울이고 있다. 한편, 우리나라의 항공우주분야 세부기술별 선진국 대비 기술수준은 표 2.3.1과 같다. 표 2.3.1에서 보듯이 핵심기술 중에서도 비행제어 및 시험비행에 관한 기술이 선진국 대비 30~40% 정도로 매우 낙후되어 있다.

제 3 장 연구개발수행 내용 및 결과

제 1 절. Mid-level 비행제어 법칙 연구

1. 지능형 비행제어 시스템 연구 개요

초창기의 비행기는 안전하게 비행을 하는 것이 가장 큰 목표였다. 그러나 두 번의 세계대전과 지속적인 여타 산업의 개발은 이러한 비행기의 고전적인 임무 이외에도 보다 복잡하고 정밀한 임무 수행을 요구하기에 이르러, 최근에는 자율적으로 장애물을 회피하고, 목표를 인식하여 타격하며, 고장이 발생한 경우에도 안전하게 기지에 귀환하는 고성능 무인기의 개발에 박차를 가하고 있다. 넓은 비행영역에서 비행하는 고성능 항공기의 경우, 비선형 특성, 공력계수의 불확실성, 외란 등이 존재하며, 이에 대해서 항공기 시스템의 안정성 및 강건성을 보장하고 원하는 성능을 보장하는 제어시스템을 설계하고, 구동기 고장이나 조종면 파손과 같은 상황에서도 비행임무를 원활히 수행할 수 있는 비행제어시스템을 설계해야 한다. 이러한 문제점을 보완하기 위해 본 연구에서는 지능형 제어기법을 이용한 비행제어 시스템에 대한 연구를 수행하였다. 신경회로망 지능형 제어기법은 시스템의 수학적 모델링이 어렵거나 많은 가정이 필요한 다변수 함수의 근사화 문제를 해결하는 데에도 설계자가 함수의 구조를 수학적으로 정확히 표현하지 않고 그 입력과 출력의 관계만을 명확히 설정하여 함수 근사가 가능하므로, 수학적 모델링이 복잡한 비행제어 시스템에 활용될 수 있다. 퍼지제어기는 입출력의 관계가 간단하고, 전문가의 의견이 제어법칙에 직접 들어가며 외란이나 불확실성에 민감하지 않은 제어기이기에 고성능 항공기에 적절한 제어시스템으로 사용될 수 있다. 본 장에서는 다양한 지능형 제어기법에 대한 연구 개요를 살펴보고자 한다.

가. 신경회로망 유도법칙

신경회로망이란 인간의 뇌의 작동을 흉내 낸 것으로서 1960년에 처음으로 제안 되었다. 인간의 뇌에 존재하는 뉴런들은 복잡한 네트워크 구조로 연결되어 있으며 각각의 처리 속도는 매우 느리나 전체적으로는 병렬 처리의 구조를 가지기 때문에 현재의 컴퓨터 기술로 처리하기 힘든 문제를 빠른 시간에 처리한다. 뉴런 한 개가 하는 일은 매우 단순하다. 하나의 뉴런은 8만개 이상의 다른 뉴런으로부터 신호를 입력 받아 한 개의 신호를 출력한다. 이 신호는 뉴런 간의 네트워크를 통하여 다른 뉴런으로 전달된다.

인간의 두뇌는 기존의 컴퓨터가 가지지 못하는 여러 가지 특성을 가지고 있다. 첫째, 오차에 대한 강건성(robustness) 및 내고장성(fault-tolerance)을 가지고 있다. 둘째, 학습을 통한 새로운 환경에 대한 적응성(adaptiveness), 유연성(flexibility)이 있다. 셋째, 명확하지 않고(fuzzy) 확률적(probabilistic)이고 잡음(noise)이 있으며 일관되지 않은(inconsistent) 정보를 다룰 수 있다. 넷째, 고차원적인 병렬 분산처리(parallel distributed processing)체계이다. 이러한 인간의 두뇌가 가지는 여러 가지 특성을 시스템 공학적으로 접목시켜 탄생한 이론이 신경회로망 이론이다. 인공적인 신경회로망 회로도 이와 같은 구조를 가진다. 여러 개의 스칼라(scalar)값을 입력 받은 뉴런은 각각의 값에 가중치를 곱한 값을 주어진 전달함수에 입력한다. 전달함수의 출력 값은 다시 다른 뉴런의 입력 값이 된

다. 이처럼 인간의 두뇌와 유사한 병렬처리 능력이 있는 신경회로망은 입출력의 관계가 비선형적이고 복잡한 경우에도 학습 과정을 통하여 그 관계를 사상하는 함수 근사화 문제를 해결할 수 있음이 증명되었다.

신경회로망을 이용하는 경우 시스템의 수학적 모델링이 어렵거나 많은 가정이 필요한 다변수 함수의 근사화 문제를 해결하는 데에도 설계자가 함수의 구조를 수학적으로 정확히 표현하지 않고 그 입력과 출력의 관계만을 명확히 설정하여 함수 근사가 가능하므로, 기존의 함수 근사화 방법에 비하여 복잡한 수학적 모델링이 요구되지 않는 장점이 있다. 또한 신경회로망은 입력과 출력의 관계를 사상함에 있어 각 입력소자(input node)와 출력소자(output node)간의 연결이 집단적인 병렬 관계에 있으므로 입력에 오차가 있거나 각 연결 구조의 일부가 파손되어도 사상이 가능한 장점도 가지고 있다. 특히 입력과 출력의 함수 관계가 명확하지 않을 때에도 신경회로망은 단지 학습패턴에 의해서 학습이 가능하며 학습된 사상 능력을 통하여 실제 해석함수를 구하지 않고도 입력에 따른 함수 값을 적절히 근사하는 능력을 가지고 있다.

(1) 신경회로망의 구조

신경회로망의 구조는 그림 3.1.1과 같이 뉴런 입력(input), 시냅스 가중치(synaptic weight), 뉴런 출력(output), 임계치(bias), 활성화 함수(activation function)로 이루어져 있다.

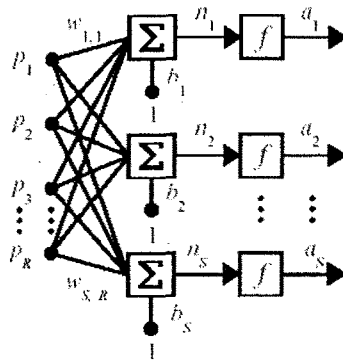


그림 3.1.1 신경회로망 구조

R개의 입력에 대해 S개의 뉴런이 존재한다면 위와 같은 구조가 된다. 여기서 활성화함수는 선형함수(Linear), 임계논리 함수(Saturating Linear), 계단함수(Hard Limit), 단극 시그모이드(Log-Sigmoid), 양극 시그모이드(Hyperbolic Tangent Sigmoid), 단극 다중 시그모이드, 방사 기저 함수(RBF), 최대값 함수(Maximum) 등이 쓰일 수 있다.

입출력 관계를 수식으로 나타내면 다음과 같다.

$$a_j = f\left(\sum_i^R p_i \cdot w_i + b_j\right) \quad (3.1.1)$$

여기서 a_j, p_i, w_i, b_j 는 각각 j 번째 뉴런 출력, i 번째 뉴런 입력, i 번째 시냅스 가중치, j 번째 임계치를 나타낸다. 또, j 는 1부터 S 까지의 값을 갖는다.

(2) 역전파(Back-Propagation) 알고리즘

역전파 알고리즘에서 중요한 요소는 가중치에 대한 적절한 평가에 있다. 이 방법의 문제점 중의 하나는 오차 함수의 지역 최소점에 빠질 수 있다는 점이다. 그러나 원하는 값을 갖게 학습 시키는 가장 일반적인 방법이다.

Step 1. 초기 가중치를 임의의 작은 값으로 설정한다.

이런 초기화는 최종 출력에 영향을 미친다. 초기 가중치가 지역 최소점에 빠졌다면 가중치가 고정된 값에서 머물게 되므로 다른 지점에서 학습을 다시 시작하면 원하는 값을 얻을 수 있다.

Step 2. 신경회로망의 성능지수를 계산한다.

성능지수는 다음과 같은 방법으로 정한다.

$$\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\} \tag{3.1.2}$$

여기서 p_q 와 t_q 는 각각 뉴런 입력과 목표 출력이다.

이 알고리즘은 MSE(mean square error)를 최소화하는 방향으로 가중치를 줄여야 하므로 다음 식을 만족해야 한다.

$$F(x) = E[e^T e] = E[(t-a)^T (t-a)] \tag{3.1.3}$$

$$\hat{F}(x) = e^T(k)e(k) = \{t(k)-a(k)\}^T \{t(k)-a(k)\} \tag{3.1.4}$$

Step 3. 민감도(sensitivity)를 구한다.

$$s^m \equiv \frac{\partial \hat{F}}{\partial n^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix} \tag{3.1.5}$$

Step 4. 정방향 전파(forward propagation) 되는 값을 계산한다.

$$\begin{aligned}
\mathbf{a}^0 &= \mathbf{p} \\
\mathbf{a}^{m+1} &= \mathbf{f}^{m+1}(\mathbf{W}^{m+1}\mathbf{a}^m + \mathbf{b}^{m+1}) \\
\mathbf{a} &= \mathbf{a}^M
\end{aligned} \tag{3.1.6}$$

여기서 m 은 $-1, 0, 1, 2, \dots, M-1$ 이다.

Step 5. 역방향 전파(backward propagation) 되는 값을 계산한다.

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(t - \mathbf{a}) \tag{3.1.7}$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1} \tag{3.1.8}$$

여기서 $\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_m^m) \end{bmatrix}$, $\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m}$ 이다.

Step 6. 가중치를 업데이트 시킨다.

이는 Steepest Descent 방법과 유사하게 알고리즘을 수행하면 된다.

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T \tag{3.1.9}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \tag{3.1.10}$$

Step 7. 공차를 만족시킬 때까지 Step 2부터 알고리즘을 계속 반복한다.

Levenberg-Marquardt Back-Propagation(LMBP)

Levenberg-Marquardt Back-Propagation은 가장 성능이 우수한 역전파 알고리즘으로 학습 수렴 속도가 굉장히 빠르다고 알려져 왔다.

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) \tag{3.1.11}$$

$$\mathbf{v}^j = [v_1 \ v_2 \ \dots \ v_N] = [e_{1,1} \ e_{1,2} \ \dots \ e_{S^M,1} \ e_{1,2} \ \dots \ e_{S^M,Q}] \tag{3.1.12}$$

$$\mathbf{x}^j = [x_1 \ x_2 \ \dots \ x_n] = [w_{1,1}^1 \ w_{1,2}^1 \ \dots \ w_{S^1,R}^1 \ b_1^1 \ \dots \ b_{S^1}^1 \ w_{1,1}^2 \ \dots \ b_{S^2}^M]$$

여기서 $N = Q \times S^M$, $n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1)$ 이다.

$$J(x) = \begin{bmatrix} \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \dots \\ \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & \dots & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \dots \\ \vdots & \vdots & & \vdots & \vdots & \ddots \end{bmatrix} \quad (3.1.13)$$

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \tilde{s}_{l,h}^m \times a_{j,q}^{m-1} \quad \text{단, 가중치에 대해}$$

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \tilde{s}_{l,h}^m \quad \text{단, 임계치에 대해}$$

Marquardt의 민감도는 다음과 같다.

$$\tilde{s}_{i,h}^m \equiv \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \quad (3.1.14)$$

$$\tilde{S}_q^M = -\dot{F}^M(n_q^M) \quad (3.1.15)$$

$$\tilde{S}_q^m = \dot{F}^m(n_q^m)(W^{m+1})^T \tilde{S}_q^{m+1} \quad (3.1.16)$$

$$\tilde{S}^m = [\tilde{S}_1^m \mid \tilde{S}_2^m \mid \dots \mid \tilde{S}_Q^m] \quad (3.1.17)$$

나. 퍼지 논리 유도법칙

전문가 시스템인 퍼지 논리를 이용하여 제어기를 설계할 수 있다. 특히 퍼지 논리 제어기는 최적화 결과로 나온 입출력을 기반으로 적절한 퍼지 규칙 베이스와 룰을 만들면 되므로 간단하게 제어기를 설계할 수 있다는 장점과 외란과 비선형성이 강한 시스템에 유리하다.

(1) 퍼지 논리 제어기의 특징

퍼지 논리 제어기는 입력과 출력의 관계를 쉽게 얻을 수 있는 일종의 방법으로써 다음과 같은 이유에서 널리 사용된다.

- 가) 퍼지 논리는 정확하기 않은 자료에 대해서 민감하지 않다.
- 나) 임의의 복잡성을 가지고 있는 어떤 비선형 함수도 모델링할 수 있다.
- 다) 숙련자의 경험에 바탕을 두어 설계할 수 있는 특성을 가지고 있다.
- 라) 전통적인 제어, 유도 방법과 쉽게 같이 사용될 수 있다.

마) 퍼지 논리는 인간의 의사소통에 기초를 두고 있어 비선형성과 불확실성을 다루는데 매우 효율적이다.

(2) 퍼지 논리의 개요

먼저 퍼지 논리 제어기의 추론과정을 설명하는데 필요한 용어를 정의하면 다음과 같다.

1) 소속함수(membership function)란 입력공간의 각 점들을 0과 1사이의 소속정도로 나타낸 곡선이다. 소속함수로는 그 형태에 따라 선형함수, 가우시안(Gaussian) 함수, 시그모이드(sigmoid)함수, 2차 3차 다항함수 등이 있다. 그리고 가장 간단한 소속함수는 직선들로 구성된 함수로부터 얻어질 수 있는데 삼각형 소속함수(triangular membership function)와 사다리꼴 소속함수(trapezoidal membership function)가 있다. 이런 직선 소속함수들은 계산이 단순하다는 장점이 있다. 비록 가우시안 소속함수와 종형의 소속함수는 매끈하다는 장점이 있으나 어떤 응용분야에서는 중요할 수도 있는 비대칭적인 소속함수를 나타낼 수 없다는 단점이 있다. 소속함수의 형태는 주관적으로 결정되는데, 여기에는 제어 대상 시스템의 특성과 원하는 제어기의 특성이 고려되어야 한다. 그러나 색다른 형태의 소속함수가 완벽하게 좋은 퍼지 추론 시스템을 만들기 위해서 꼭 필요한 건 아니다. 외란의 영향이 심한 경우에는 소속함수의 형태를 충분히 넓게 잡아서 외란에 대한 민감도(sensitivity)를 낮추는 게 좋다.

2) 퍼지집합(fuzzy set)은 고전적인 집합(crisp set)의 확장으로써 다음과 같이 나타낼 수 있다.

$$A = \{x, \mu_A(x) | x \in X\}$$

여기서 $\mu_A(x)$ 는 x 의 소속함수를 나타낸다.

3) If-Then Rules : If x is A, then y is 'B'.

여기서 A와 B는 퍼지 집합으로 정의된 언어변수이다. 'x is A'는 조건부라고 불리며 'y is B'는 결론부'로 불린다. 일반적으로 한 개의 규칙만으로는 좋은 결과를 얻기가 어렵다. 요구되는 것은 서로 경쟁할 수 있는 여러 개의 규칙이며, 각 규칙의 출력은 퍼지집합이다. 각 규칙에 대한 출력 퍼지 집합들은 하나의 출력퍼지집합으로 합쳐지며, 결과적인 집합은 비퍼지화(defuzzified)되어 한 개의 값이 출력되도록 한다. 결론부가 어떻게 조건부로부터 영향을 받는가는 퍼지추론에 의해서 정해진다.

4) 퍼지추론(fuzzy inference)은 여러 규칙을 바탕으로 주어진 입력벡터의 값들을 해석하여 출력벡터를 결정하는 과정이다. Mamdani 형태와 Sugeno 형태, 이 두 종류의 퍼지추론시스템이 많이 쓰이는데, 특히 Mamdani에 의해서 제안된 추론 방법이 가장 널리 쓰인다. 1985년에 처음 도입된 Sugeno, 즉 Takagi-Sugeno-Kang 방법의 퍼지추론은 많은 점에서 Mamdani 방법과 비슷하다. 처음 퍼지 추론의 입력을 퍼지화하고 퍼지연산자를 적용하는 과정은 똑같으며 Mamdani 형태의 추론방법과 가장 큰 차이는 출력 소속함수들이 단지 선형이나 상수 값으로 주어진다는데 차이가 있다. 0 차의 Sugeno 퍼지모델은 다음과 같은 형

태를 갖는다.

if x is A and y is B, then z=k.

여기서 A와 B는 조건부의 퍼지집합이고 반면에 k는 결론부에서 정의된 상수이다. 결론부가 단순하므로 계산면에서 효율적이고 출력면의 연속성이 보장된다는 장점이 있을 뿐만 아니라 수학적으로도 분석하기 적합하다. 반면에 Mamdani 방법의 장점은 직관적이고 인간의 언어적인 정보를 함유하는데 효과적이다.

퍼지 논리 제어기의 추론 과정

구체적인 Mamdani 형태의 퍼지추론과정은 다음과 같이 5개의 단계로 구성되어 있다.

Step 1. 입력값의 퍼지화 (fuzzification)

첫 번째 단계는 명확한 값(crisp value)으로 주어지는 입력을 받아들여 이 입력이 소속함수들을 통해 주어진 퍼지 집합에 소속된 정도를 결정하는 단계이다. 즉 입력 값을 적절한 언어적인 값으로 변환시킨 후, 이 퍼지 언어집합에 어떻게 소속되는가를 결정한다. 여기서 언어적인 값은 일반적으로 입력부 전체집합 내에서 정의된 퍼지집합들의 이름이 사용된다(퍼지규칙에서 조건부와 결론부의 언어적 변수는 퍼지집합에 대응한다). 대체로 퍼지변수는

LN : Large Negative
MN : Medium Negative
SN : Small Negative
ZE : Zero
SP : Small Positive
MP : Medium Positive
LP : Large Positive

등의 퍼지값을 가지며, 각각 고유한 소속함수를 가진다.

Step 2. 조건부에서 퍼지연산자(fuzzy operations) 적용 (AND 또는 OR)

일단 입력이 퍼지화되면, 조건부의 각 부분이 규칙을 어느 정도 만족시키는가를 알 수 있다. 퍼지연산자의 입력은 2개 이상의 퍼지화된 입력변수의 소속함수로서 조건부의 규칙은 그림 8과 같이 적용될 수 있다. 퍼지 AND 연산자는 단순히 두 값중 최소값을 선택하게 만든다(OR 연산자의 경우에는 두 값 중 최대값을 선택하게 만든다). 이로서 그림 2에 기술된 두 개의 퍼지연산자 규칙은 실행된 것이다.

Step 3. 조건부와 결론부의 관계 결정

결론부도 소속함수로 나타내진 퍼지집합으로, 이 단계에서의 입력은 한 숫자이고 출력은 퍼지집합으로서 모든규칙에 대해서 계산이 행해진다. AND 방법(Min)은 그림 3.1.2와 같이

출력 퍼지집합을 절단함으로써 얻어진다.

Step 4. 규칙베이스를 통해서 결론들의 집합 결정

결론은 모든 규칙을 다 시도한 후 이를 기초로 얻어진다. 결론을 얻기 위해서는 규칙들은 어떤 방식으로든 결합해야 한다. 이 단계에서의 통합(aggregation)은 각 규칙의 출력들을 나타내는 퍼지집합들을 결합해서 하나의 퍼지집합을 얻는 과정으로서 일반적으로 이 과정은 교환가능한 속성을 가지고 있으므로 규칙이 적용되는 순서는 중요하지 않다. 그림 2는 모든규칙들이 다 시도되어진 후 어떻게 각 규칙의 출력들이 결합되는지를 보여준다.

Step 5. 비퍼지화를 통한 출력값 생성

위에서 계산된 퍼지값은 실제제어입력으로 사용할 수 없으므로, 이를 명확한 비퍼지값으로 변환시키는 과정이 필요하다. 이 과정을 비퍼지화 또는 일점화라고 하고 이 과정의 입력은 퍼지집합이고 출력은 명확한 수가 된다. 즉 비퍼지화 과정은 이러한 하나의 수를 결정하는 과정이며 가장 널리 쓰이는 방법이 무게중심법(centroid defuzzification method)으로서 곡선아래의 영역에서 중심을 계산하는 방법이다. 일반적으로 무게중심법은 다른 방법들에 비해서 우월한 성능을 보이는 경향이 있다. 이 방법의 단점으로는 퍼지소속함수의 결과가 오목한 경우 오목한 부분이 출력으로 선택될 수도 있다는 점이다.

$$w^0 = \frac{\int \mu(w)w dw}{\int \mu(w) dw} \quad (3.1.18)$$

무게 중심법외에도 합성된 출력부 퍼지집합에서 소속함수 값이 가장 큰 부분에 해당되는 명확한 값을 출력하는 최대값 방법(max criterion method), 출력부 퍼지집합에서 소속함수가 최대값을 가지는 곳의 값들의 평균을 사용하는 최대평균법(mean of maximum method) 등이 있다.

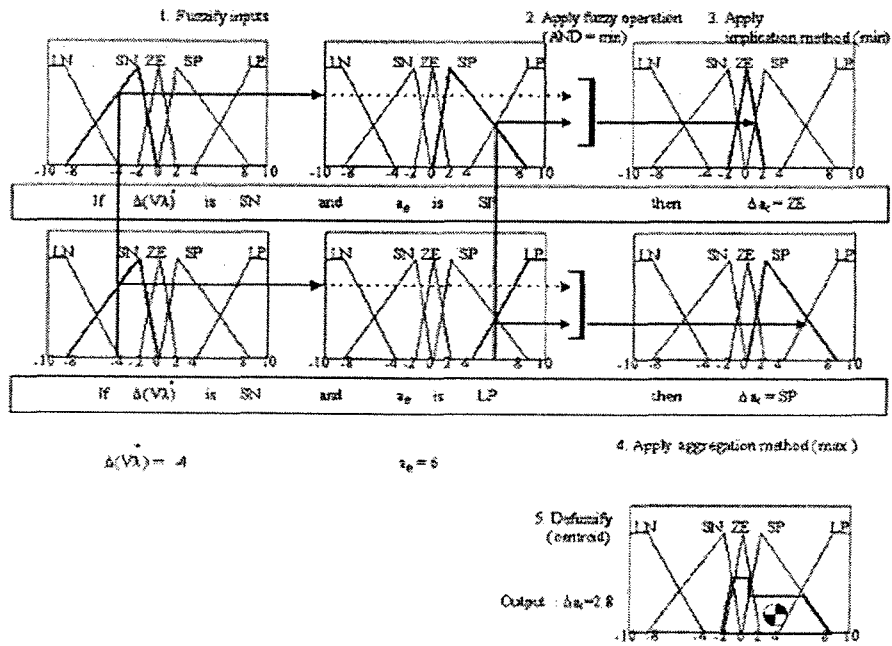


그림 3.1.2 퍼지추론 과정

다. 유전자 알고리즘

유전자 알고리즘은 다아윈이 주장한 자연 진화의 특징인 적자생존(survival of fittest)과 자연 도태(natural selection)의 원리를 바탕으로 하여 이론이 세워진 최적화 알고리즘의 일종이다. 자연 진화의 법칙에 의하여, 일정한 환경에 무리를 지어 살고 있는 생명체들은 그 환경에 적합한 형질을 가진 개체가 세대가 이어지면서 생존할 확률을 크게 가지며 교배와 돌연변이의 과정을 통하여 자식 세대에서 부모 t대보다 더 좋은 방향으로 진화해 나가고, 부적합한 형질의 개체들은 진화의 과정에서 점차 도태되어 간다. 이와 같은 진화의 과정을 되풀이 하면 결국 주어진 환경에 가장 적합한 형질의 개체들만 생존할 것이다. 유전자 알고리즘은 이러한 생명체의 자연 진화 법칙을 일반 최적화 문제에 적용한 방법으로써, 설계 영역에 다수의 설계점을 분포 시켜 목적함수 값과 제한 조건의 위반 정도에 따라 각 설계점에 적합성(fitness)을 부여한다. 설계점의 적합성이 클수록 다음 단계인 번식(reproduction)에서 교배와 돌연변이의 과정에 참여할 확률을 크게 하여, 적합성이 좋은 설계점에 비슷한 형질을 가진 설계점들이 다음 단계에 좀 더 많이 형성되고, 세대가 진행될수록 전체 설계점들은 좋은 방향으로, 즉 최적의 값으로 탐색이 진행된다.

엔지니어링이나 경영과학의 많은 문제가 최적화 문제로 귀결된다. 최적화 문제에 대해 국소적 방법을 적용 시, 얻어진 국소 해가 전역해라고 판별하기는 용이하지 않다. 즉, 도함수나 기울기 등을 사용하는 기존의 최적화 방법은 일반적으로 전역해를 찾거나 식별하기 어렵다고 할 수 있다. 비록 많은 노력을 들여 특정 유형의 최적화 문제를 푸는 정확한 방법을 개발할 지라도, 다른 유형의 문제에 적용하기 곤란하여 단지 일정한 부류의 문제에 대해서만 유용한 방법이 되기 쉽다. 지금까지 최적화 문제를 풀기 위한 수많은 기법들이 제시되었다. 그중의 일부는 제한된 영역에서만 사용되기도 한다. 지금까지 개발된 최적화 기법을 분류하여 보면, 미분(Gradient)에 의한 방법, 임의 탐색(Random Search)에 의한 방법, 반복

(enumerative)탐색에 의한 방법 등 세 가지로 나눌 수 있다.

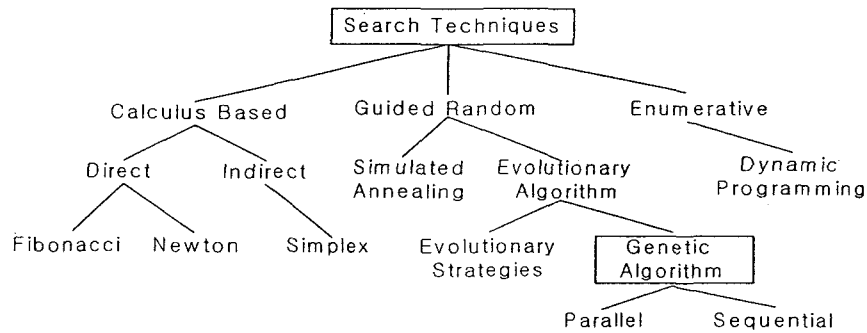


그림 3.1.3 최적화 알고리즘의 계보

최근 진화 알고리즘이 최적화 문제의 전역해나 근사해를 구하는 방법으로 각광을 받고 있다. 진화 알고리즘은 자연세계의 생물체가 자연 선택과 유전 법칙을 통하여 진화하는 과정에 기반한 탐색 및 최적화 알고리즘이다. 즉, 실세계의 문제를 풀기 위해 잠재적인 해들을 컴퓨터상에서 코딩된 개체로 나타내고, 여러 개의 개체를 모아 개체집단을 형성한 뒤 세대를 거듭하면서 이들의 유전정보를 서로 교환하거나 새로운 유전 정보를 부여하면서 적자 생존의 법칙에 따라 모의 진화를 시킴으로써 주어진 문제에 대한 최적해를 찾는 모델이다.

기존의 최적화 알고리즘들은 함수의 연속성, 미분 가능성, 선형성 등이 보장되지 않을 경우 함수를 최적화하는데 많은 어려움을 겪었으나, 진화 알고리즘은 점 탐색 알고리즘이어서 함수의 연속성, 미분 가능성, 선형성 등이 보장되지 않아도 함수를 최적화하는 것이 가능하다. 또한, 다중 목적 함수의 최적화는 수학적으로 복잡하고 어려운 문제이나 진화 알고리즘에서는 변수 집단을 동시에 부호화하여 유전 진행을 행하므로 수월하게 다중 목적 함수의 최적화를 처리할 수 있다. 진화 알고리즘과 기존 최적화 알고리즘과의 차이점은 진화 알고리즘은 확률과 통계 법칙을 사용하나, 기존 최적화 알고리즘은 결정적인 (deterministic) 방법을 이용하여 최적화를 행한다. 진화 알고리즘의 특징을 요약하면 다음과 같다.

확률적 탐색기법

진화 알고리즘은 적합도에 따른 선택에 의해 탐색이 유도되는 확률적인 탐색 기법의 일종이다.

함수의 보조 정보 미사용

진화 알고리즘은 대상 함수의 연속성이나 도함수를 필요치 않으며 대상 함수를 블랙박스 처럼 취급한다.

개체집단 기반 탐색

한 점이 아닌 개체집단을 대상으로 병렬적인 탐색을 한다. 탐색공간의 다수의 점들에서 샘플링 한 대상 함수의 정보를 진화 연산자들에 의해 재결합하고 평가하며 점차적으로 해

를 개선해 간다. 다수의 잠재적인 해 집단이 얻어질 수 있으므로 다목적 함수의 최적화 등에 유용하다.

강인성

진화 알고리즘은 대상 함수에 국소 점이 많거나 작은 노이즈가 가미되더라도 쉽게 국소 해에 빠지지 않는 강인한 특성을 가지고 있다. 현재 진화 알고리즘은 복잡한 문제를 푸는데 있어서 그 가능성을 주목받고 있으며 다양한 분야에서 적용되고 있다. 진화 알고리즘에서는 해의 정밀도, 전역적 신뢰성과 국소적 개선과의 조화, 매개 변수의 최적화 그리고 문제에 대한 사전 지식의 활용 여부 등이 중요한 문제이다.

해의 정밀도

알려진 최적해나 기존의 정확한 방법으로 구한 해와 진화 알고리즘으로 얻어진 해의 차이를 비교하면 해의 정밀도가 평가될 수 있다. 진화 알고리즘은 휴리스틱한 방법이므로, 일반적으로 준 최적해를 구하는데 있어 효율적이나 해의 정밀도는 예상하기 어렵다.

전역적 신뢰성과 지역적 세밀화

전역해는 탐색 영역의 어느 위치에서라도 존재할 수 있으므로 탐색 기법의 전역적 신뢰성(global reliability)은 중요하다. 지역적 세밀화(local refinement)는 특정한 영역에 초점을 두어 유망한 점들을 대상으로 세밀한 탐색을 수행하기 위해 필요하다. 따라서 전역 최적화를 위해서는 전역적 신뢰성을 얻기 위한 노력과 국소적 세밀화를 위한 노력을 적절히 조화시키는 것이 중요하다.

매개 변수의 최적화

진화 알고리즘의 적용에 있어서 알고리즘의 성능에 영향을 미치는 변수는 다양하다. 적합도 함수를 정의하는 방법에 따라서 성능이 변할 수 있다. 또한 개체집단의 크기, 세대수, 교배 확률, 돌연변이 확률 등의 매개 변수 값에 따라서도 알고리즘의 성능은 크게 변한다. 개체집단의 크기가 작으면 적합도 계산에 필요한 시간을 절약할 수 있으나 개체간의 다양성의 빠른 손실로 인해 최적의 해를 구하기 전에 수렴할 위험성을 내포한다. 반면에 개체 집단의 크기가 크면 최적해에 도달할 확률은 높으나 많은 기억용량과 계산시간을 필요로 한다. 이들의 성능 평가 항목을 모두 만족시켜 주는 최저그이 개체집단의 크기를 정하는 방법은 문제의 성격과 다른 제어 파라미터들의 값에 따라 다르다.

문제에 대한 사전 지식

일반적으로 휴리스틱한 알고리즘의 경우 문제에 대한 사전 지식이 있는 경우에 탐색의 성능을 크게 증진시킬 수 있는데, 이는 진화 알고리즘에도 해당한다.

유전자 알고리즘의 기본 원리

유전자 알고리즘은 하나의 해 대신에 가능한 해의 집단(population)을 병렬로 탐색한다. 유전자 알고리즘은 유전자학과 유전공학 및 컴퓨터 사이언스 관련 용어에 기반하고 있으며

로, 생물학적 전문용어와 인위적인 용어가 혼재되어 있다. 생물학에서 유전 물질은 DNA, 즉 염색체(chromosome)이다. 유전자 알고리즘의 기본 구조는 이러한 생물의 진화 과정을 문제 해결 과정으로 옮겨 놓은 것이다. 문제 해결상의 임의의 해는 유전자 알고리즘이 이해하는 형태로 표현되어 염색체라 부른다. 유전자 알고리즘에서는 대부분의 경우 정해진 수의 염색체 집단을 해집단(population)이라 한다. 염색체상의 각 인자를 유전자(gene)라 한다. 생물학에서 유전자형(genotype)은 유전자의 조합이고, 표현형(phenotype)은 관찰되는 형질을 말한다. 유전자 알고리즘에서도 비슷하게 염색체 그 자체를 유전자형이라 하고, 이와 대응되는 해의 성격이나 품질 등을 표현형이라 한다. 유전인자의 값들을 alleles라 하고 유전자의 위치를 locus라 한다. 유전자 알고리즘에서 많이 사용되는 주요 용어는 표 3.1.1과 같다.

표 3.1.1 유전자 알고리즘의 주요 용어 설명

Genetic Algorithms	Explanation
Chromosome(string, individual)	Solution
Genes(bits)	Parts of solution
Locus	Position of gene
Alleles	Values of gene
Phenotype	Decoded solution
Genotype	Encoded solution

Holland의 유전자 알고리즘은 보통 SGA(Simple Genetic Algorithms)라고 불리는 것으로서, 2진수의 문자열을 여러 개 형성하여 최적화를 수행하게 된다. 유전자 알고리즘은 유전 연산자를 사용하여, 현재 세대의 개체군에서 다음 세대의 개체군을 만들어 내게 된다. 이러한 세대교체는 원하는 조건을 만족시킬 때까지 계속해서 이루어지게 된다. 최초로 설계변수의 초기 값에 해당하는 문자열을 사용자가 정한 개수만큼 무작위로 2진수 문자열을 생성한다.

유전자 알고리즘에서 각각의 해-개체(individual)는 목적함수의 적합도(fitness)에 따라 평가된다. 개체 집단에 적용되는 선택(selection), 교배 및 돌연변이와 같은 유전자 연산자는 기본적으로 다음과 같은 역할을 수행한다. 선택은 다음 세대(generation)에 각 개체가 더 좋은 적합도를 가지게 함으로써 탐색 점들을 더 좋은 탐색 영역으로 유도한다. 교배 및 돌연변이는 탐색공간에 새로운 점들을 생성시킨다. 교배 연산은 임의의 두 개체의 정보를 혼합하는 방식으로 각 세대에 걸쳐 부모개체 집단의 정보를 자손개체 집단에 전달한다. 돌연변이 연산은 일종의 배경 노이즈(background noise)를 탐색 점들에 가해 이들을 새로운 위치로 이동시키는 역할을 한다. 유전자 알고리즘의 초기화 과정에서 개체집단은 임의로 초기화되고, 반복적인 세대교체 과정이 도입된다. 먼저 각 개체들의 적합도가 목적함수를 이용하여 계산된 후, 선택연산이 개체집단에 대해 행해진다. 다음에 교배 및 돌연변이 연산이 현재의 개체집단에 대해 적용되면, 새로운 점들이 생성되어 다음 세대의 개체집단을 형성하게 된다. 결과적으로 충분한 세대교체 후에 개체집단의 최우수 개체는 문제의 최적 해나 준 최적 해를 표현하게 된다. 단일 개체집단(single population)에 기반한 단순한 유전자 알고리즘은 그림 3.1과 같다. 이와 같이, 유전자 알고리즘은 기존의 탐색 기법이나 최적화 알고리즘과는 여러 면에서 다르다고 할 수 있다.

유전알고리즘의 전형적인 구조

유전자 알고리즘은 진화 알고리즘의 일종으로 또 그 변형으로서 여러 형태가 있으나 표준적인 것은 다음과 같다. 유전자 알고리즘은 개체군이라 불리는 해들의 집합을 진화시킨다. 개체군 X 의 각 개체 $x_i (i=1, \dots, n)$ 는 대상 문제의 해의 한 후보가 된다. 개체는 보통 변수의 열로써 표현되며 그 열의 각 원소는 유전인자로 불린다. 그 유전인자의 값은 유전자값이라 불리는데 보통 $[0,1]$ 의 값을 가진다.

유전자 알고리즘은 각 개체마다 계산된 적합도 함수를 최대화할 수 있다. 만약 목적함수를 최소화하는 문제라면 목적함수의 감소가 적합도 함수의 증가가 되도록 조절하여야 한다. 그 순서는 다음과 같다.

- 1) 문제를 정의하고 모든 잠재적 해의 적합도를 표현할 수 있는 목적함수를 정의한다.
- 2) 후보 해들로 구성되는 하나의 그룹을 구속조건에 맞게 형성한다. 각 후보 해 들은 염색체(chromosome)이라 불리는 벡터 x 로 코딩된다. 홀랜드는 모든 해들이 이진스트링으로 표현되어야 한다고 제안했다. 예를 들어 만일 다음과 같은 함수를 최대화 하는 스칼라 x 값을 구하고자 한다고 하자.

$$F(x) = -x^2$$

그러면 x 의 어느 유한한 범위가 선택되고, 그중 가장 작은 값은 $\{0 \dots 0\}$ 으로, 가장 큰 값은 $\{1 \dots 1\}$ 로 표현된다. 원하는 정확도에 따라 이진 스트링의 적합한 길이가 달라지게 된다.

- 3) 그룹안에 있는 각 염색체 $x_i, (i=1, \dots, P)$ 는 적합도 평가를 위해 적당한 형태로 디코딩 되며, 목적함수에 따라 적합도값 $\mu(x_i)$ 가 정해진다.

- 4) 각 염색체는 재생산 확률, p_i 이 부여된다. 각 개체의 선택될 확률은 그 적합도에 비례하여 결정된다.

- 5) 현재의 그룹에서 선택확률에 따른 선택을 거쳐, 새로운 개체들의 그룹이 생성된다. 선택된 염색체들은 교배와 돌연변이 같은 유전 연산자를 통하여 자손을 생성한다. 교배는 두 개의 염색체(부모)에 작용하여 일차원 문자열로 된 염색체 상에서 교차로 가로지르는 방법을 통해 두개의 새로운 염색체(자손)을 생성한다. 비트 돌연변이는 코딩에서 각 비트를 바꾸는 것이다.

- 6) 만일 적합한 해가 발견되거나 진화 시간이 초과된다면 그 진화는 정지한다. 그렇지 않은 경우 진화는 (3)단계로 되돌아가 계속된다.

앞의 설명을 간단히 하면 다음과 같다.

step 1. 초기 개체군을 임의로 생성한다 $X(0) = (x_1, \dots, x_n)$.

step 2. 현재 개체군 $X(t)$ 의 모든 개체(x_i)의 적합도 $F(x_i)$ 를 계산한다.

step 3. 재생산 연산자를 사용하여 과도기 개체군 $X_r(t)$ 을 형성한다.

step 4. 과도기 개체군에 다른 연산자를 적용하여 다음세대 개체군 $X(t+1)$ 을 생성한다.

step 5. $t = t + 1$; 종료 조건이 만족되지 않으면 step 2로 이동한다

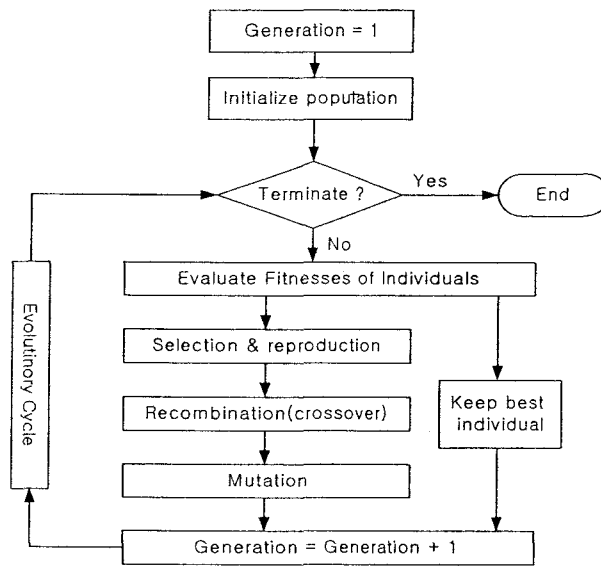


그림 3.1.4 유전자 알고리즘의 흐름도

2. 6자유도 비선형 시뮬레이션 프로그램

가. 개요

NASP(Nonlinear Aircraft Simulation Program)는 비행체의 운동을 묘사하는 프로그램이다. NASP를 구성하기 위해서는 해당 비행체의 질량, 스핀, 코드 등의 물리적 변수는 물론 이력니와, 공력계수, 엔진모델 등 여러 비행자료 및 데이터를 요구한다. NASP 구성에 가장 큰 걸림돌은 이같이 비행체 고유의 기동 특성을 결정짓는 데이터를 획득하기 어렵다는 점이다. 그러나 수많은 실험과 시험비행을 통해 비행 특성 데이터를 획득하는 시스템 인식을 성공적으로 수행하게 되면, 컴퓨터 속에 비행체를 보유하고 있는 것과 같은 결과를 갖는다. 한편 현대에 개발되어 운영되는 대부분의 고성능 항공기에는 FBW기법을 이용한 제어시스템이 장착되어 있다. FBW에 의한 제어시스템을 장착한 경우, 비행시스템에 대하여 시스템 인식을 통하여 비행 실험 이전에 비행시스템 모델을 구축하여야 하며, 이와 같은 역할을 수행하는 것이 NASP이다. 따라서 비행시스템 특성을 보다 정확히 묘사하는 NASP를 구성할 수록, 비행시스템에 대해 보다 효과적이고 지능적인 제어기 구축이 가능하고, 이는 연쇄적으로 비행체의 비행성능을 향상시키는 결과를 야기한다.

그림 3.1.5은 본 연구에서 개발하고 있는 비선형 시뮬레이터의 구조를 보이고 있다. 본 프로그램이 완성되면 제어기 시험(Controller Design Contest)을 수행해 연구원들에게 새로운 지능형 통합비행제어기의 설계 동기를 부여하는 한편, 설계된 제어기의 타당성을 검증하는 척도를 제공할 것이다. 디자인될 시스템의 성능을 제시하고, 평가하기 위해서는 실험 및 수치계산을 통해 기준값을 작성해야 한다.

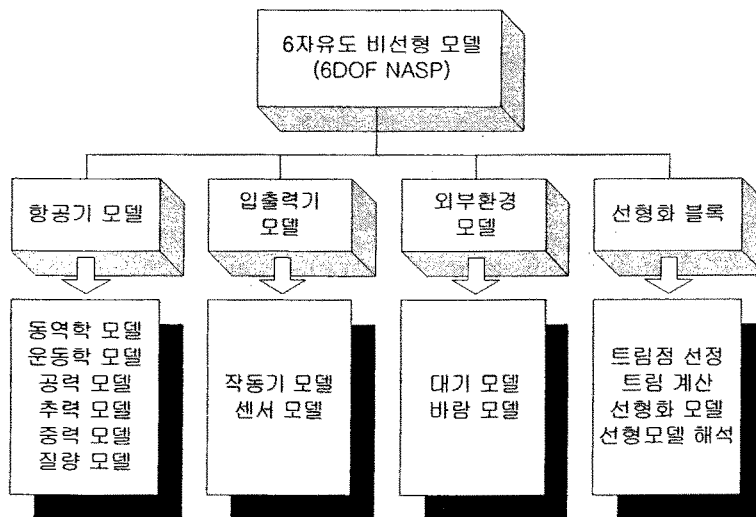


그림 3.1.5 비선형 시뮬레이터 구조

앞서 살펴본 것과 같은 특성을 갖는 NASP는 비행제어시스템 개발에 있어 필수불가결한 요소이며, 지능형 통합비행제어시스템 설계 기술 확보를 위한 연구를 수행하는 본 연구에 있어 NASP는 그 중요성이 매우 크다 할 수 있다. 지능형 통합비행 제어시스템 연구에서 제안된 다양한 제어기에 대해 성능검증 및 해석은 NASP를 이용해서 수행하게 된다. 이와

같은 분석은 특정 고장이나 비행 상황에 대해 보다 안정적이며 성능이 우수한 제어를 설계할 수 있게 하며, 이를 통해 고장에 보다 능동적으로 대처하는 알고리즘을 연구할 수 있게 된다. 본 연구에서 고려하는 6자유도 비선형 시뮬레이션 프로그램의 블록선도는 그림 3.1.5과 같으며, NASP가 본 연구에서 차지하는 역할을 나타내고 있다.

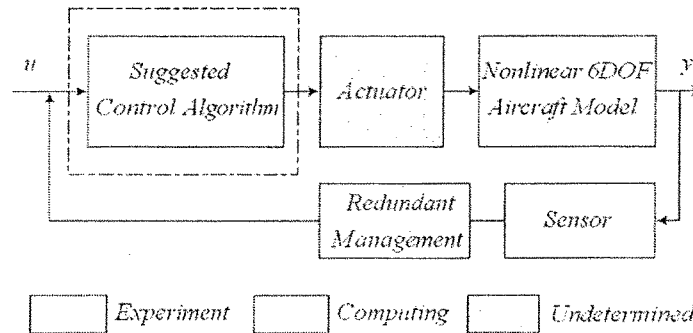


그림 3.1.6 비선형 6자유도 시뮬레이터

NASP는 다음의 과정을 통해 본 연구에 활용될 예정이다. 일단, 해석적 알고리즘 개발팀에서 지능형 통합비행제어시스템에 대한 연구를 통해 새로운 기법의 제어알고리즘 개발을 수행한다. 이때 개발된 제어기의 성능은 실제 비행기에 적용하여 비행시스템의 반응을 살펴봄으로써 해석할 수도 있으나 고가에 비행시스템을 이용하여 이러한 실험을 수행하는 것은 타당치 못하다. 따라서 비행시스템을 모사하는 NASP를 비행시스템으로 이용하게 된다. NASP를 통한 제어기의 성능 검증 방법은 크게 두 가지로 나뉜다. 첫 번째 방법은 고전적인 방법으로 제안된 제어기를 이용하여 NASP, 제어기 폐회로를 구성하고 입력에 대한 시스템의 반응을 살펴보는 것이다. 이는 일반적인 비선형 수치 시뮬레이션 기법이다. 비행제어시스템 성능검증의 두 번째 방법은 HILS(Hardware In the Loop Simulation)를 이용한 제어시스템의 성능평가이다. 이는 다음과 같은 절차를 거친다. 우선, 제안된 제어기에서 산출되는 신호를 NASP에 입력 신호로 부가한다. 입력신호에 의해 NASP에서는 비행상태 신호를 외부에 송신한다. 이 신호를 비행상태의 물리적 모사가 가능토록 구현된 모션 테이블에 연결하여 비행체의 움직임 신호와 물리적 움직임을 동기화 시킨다. 이 모션 테이블 위에 비행체에 장착된 센서와 동일한 센서를 설치한 후 센서의 출력신호를 판독하여 비행시스템의 비행상태변수의 정보를 획득한다. 획득된 정보는 다시 제어기의 비행상태를 묘사하는 입력변수로 사용되며, 제어 알고리즘은 이를 이용하여 출력 제어 신호를 생성한다. 이와 같은 일련의 작업을 설정된 비행시간 동안 반복 수행하고, 그 결과를 분석하여 제안된 비행제어 시스템의 성능을 평가하는 기법을 작업을 HILS를 통한 시스템 분석이라 한다.

본 연구에서 해석적 방법과 HILS에 의한 방법을 이용하여 제안된 비행제어시스템의 성능을 파악할 예정이며, 이를 위해서는 정확한 NASP 모델을 구축해야만 한다.

나. 항공기 모델

고려한 비행체는 F-16 모델로서 다음과 같은 물리적 특성을 갖는다.

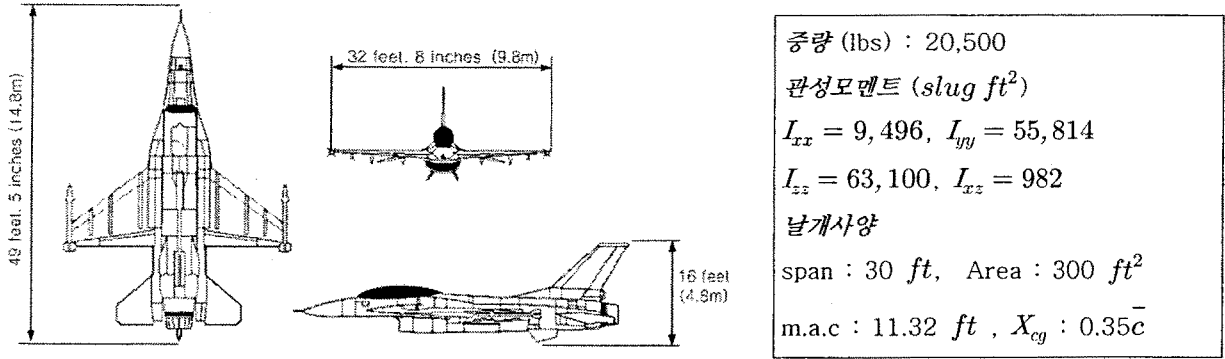


그림 3.1.7 F-16 모델 물리적 변수

한편, 비행체에 작용하는 공력을 계산하기 위해서는 비행체에 작용하는 받음각과 옆미끄럼 각에 대한 정보를 알아야 한다. 그림3.1.8와 같이 안정축 방향으로 받음각과 옆미끄럼 각을 정의하면, 다음 식에 의해 받음각과 옆미끄럼각을 얻을 수 있다.

$$\alpha = \frac{u\bar{w} - w\bar{u}}{u^2 + w^2}, \quad \beta = \frac{vV_T - v\bar{V}_T}{V_T^2 \cos\beta} \quad (3.1.19)$$

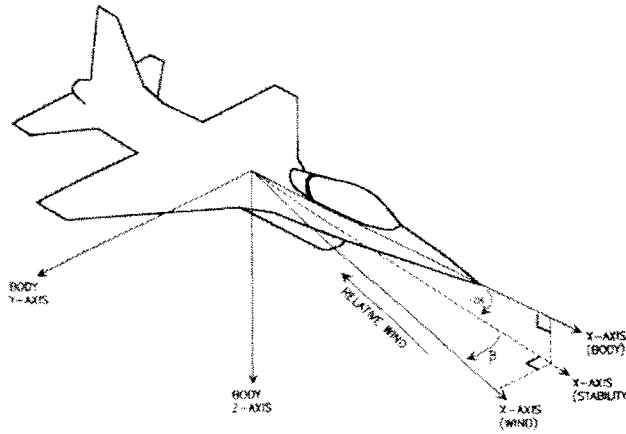


그림 3.1.8 받음각과 옆미끄럼각

본 연구에서 개발한 NASP모델은 공력계수로 속도 0에서부터 마하 1, 고도 해수면에서부터 50000ft의 범위의 데이터를 갖고 있으며, 자체 엔진모듈 모델을 포함하고 있다.

다. 운동학 모델링 및 구현

일반적으로 비행체는 공간상에서 자유로이 움직이는 강체로 가정한다. 이와 같이 비행체를 공간상에서 자유로이 움직이는 강체로 가정하는 경우 비행체는 x, y, z축의 병진운동(Translational motion)과 각 축 방향으로의 회전운동(Rotational motion)에 따라 위치와 자세가 결정된다. 이는 6개의 매개변수로 비행체를 묘사할 수 있음을 의미하며, 이는 공간

상에서 기동하는 비행체는 6자유도를 갖는 시스템임을 의미한다. 이때 비행체의 기동을 묘사하기 위해 사용되는 변수는 비행체의 병진방향 위치(x, y, z) 와 비행체의 관성 좌표계에 대한 움직 좌표계의 상대 자세 변위량인 (ϕ, θ, ψ)이다.

한편 물체의 운동방정식은 외력이 작용되어 상태변수에 영향을 미치는 동역학 모델과 상태변수간의 상관관계에 의해 정식화 되는 기구학 모델로 나누어 고려해 볼 수 있다. 시스템을 이처럼 분석하면 다소 복잡해지는 문제가 발생하나 단위 실험을 수행하기 편리하며, 시스템을 모듈화 하였기에 모델이 바뀌는 경우 해당 블록만 변경하면 되어 향후 유지보수가 편리하다는 장점이 있다.

(1) 동역학 모델

3차원 공간상에서 움직이는 6자유도 비행체의 병진운동을 나타내는 동역학 모델은 다음과 같다.

$$\begin{aligned} \dot{u} &= vr - wq - g \sin \theta + (X + T)/m \\ \dot{v} &= wp - ur + g \sin \phi \cos \theta + Y/m \\ \dot{w} &= uq - vp + g \cos \phi \cos \theta + Z/m \end{aligned} \tag{3.1.20}$$

이때 작용 외력은 X, Y, Z, T, g로 각각 x축 방향 공력, y축 방향 공력, z축 방향 공력, 엔진 추력, 중력을 의미한다. 식 3.1.20을 안정축 방향으로 좌표변환 후 이를 Simulink를 구성하였다. 아래의 그림은 구현된 병진방향 동역학 모델이다.

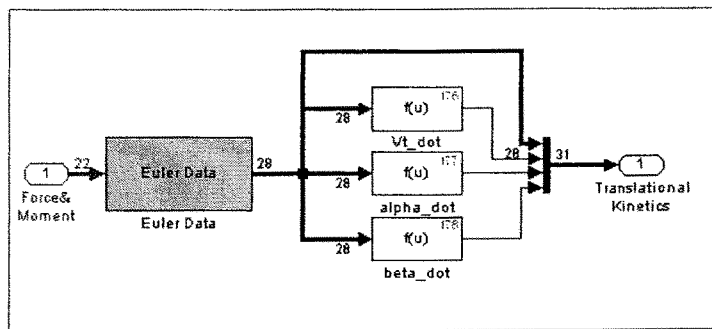


그림 3.1.9 병진방향 동역학 시뮬링크 모델

회전운동을 나타내는 동역학 모델은 다음과 같다.

$$\begin{aligned} \dot{p} &= [\{ (I_{xx} - I_{yy} + I_{zz}) I_{xz} p + (I_{yy} I_{zz} - I_{zz}^2 - I_{xz}^2) r + I_{xz} HE \} q + I_{zz} L_A + I_{xz} N_A] / (I_{xx} I_{zz} - I_{xz}^2) \\ \dot{q} &= [\{ (I_{zz} - I_{xx}) p - HE \} r - I_{xz} (p^2 - r^2) + M_A] / I_{yy} \\ \dot{r} &= [\{ (I_{xx}^2 + I_{zz}^2 - I_{yy} I_{xx}) p - (I_{xx} - I_{yy} + I_{zz}) I_{xz} r + I_{xz} HE \} q + L_A I_{xz} + N_A I_{xx}] / (I_{xx} I_{zz} - I_{xz}^2) \end{aligned} \tag{3.1.21}$$

이 때 시스템에 작용하는 외력은 L_A, M_A, N_A, HE 로 각각 롤, 피치, 요우 방향 모멘트와 엔진 모멘트를 나타낸다. 이를 시뮬링크를 이용하여 구성하면 아래의 그림과 같다.

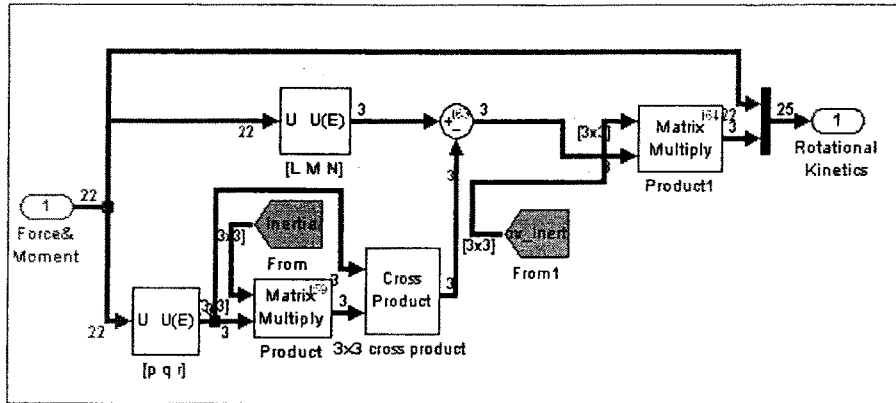


그림 3.1.10 회전방향 동역학 시뮬링크 모델

(2) 기구학 모델

절대좌표에 대한 비행체 기체좌표의 상대적 움직임을 나타내는 모델이 기구학 모델이다. 병진방향의 기구학 모델은 흔히 항법식으로도 불리우며, 비행체의 위치를 나타낸다. 비행체의 절대좌표에 대한 위치는 다음 식으로 계산된다.

$$\begin{aligned} \dot{x}_E &= u \cos \psi \cos \theta + v (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) + w (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) \\ \dot{y}_E &= u \sin \psi \cos \theta + v (\sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi) + w (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) \\ \dot{z}_E &= u \sin \theta - v \cos \theta \sin \phi - w \cos \theta \cos \phi \end{aligned} \quad (3.1.22)$$

이를 시뮬링크로 구현하면 다음 그림과 같다.

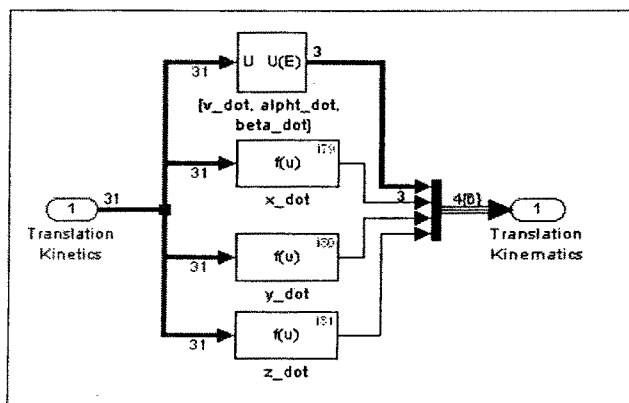


그림 3.1.11 병진방향 기구학 시뮬링크 모델

비행체의 자세정보를 얻기 위해서는 지구 관성좌표계와 비행체 고정좌표계와의 사이의 각인 오일러 각을 계산하여야 한다. 항공기에서 주로 사용하는 3-2-1 좌표변환을 이용하여 오일러 각을 계산하면 다음과 같다.

$$\begin{aligned}\dot{\phi} &= p + (q \sin \phi + r \cos \phi) \tan \Theta \\ \dot{\Theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= (q \sin \phi + r \cos \phi) \sec \Theta\end{aligned}\tag{3.1.23}$$

위와 같이 오일러 각을 사용할 경우 피치각이 90°에서 singular하게 되며, 이를 해결하기 위해 쿼터니안(Quaternion)을 이용하여 계산하기도 한다. 그러나, 본 연구에서 다루는 시스템은 항공기이며, 예각의 범위에서 공력계수들은 의미를 갖게 된다. 따라서 위의 오일러 관계식을 이용하여 자세각을 계산하여도 의미가 있다. 위의 식을 시뮬링크를 이용하여 나타내면 다음과 같다.

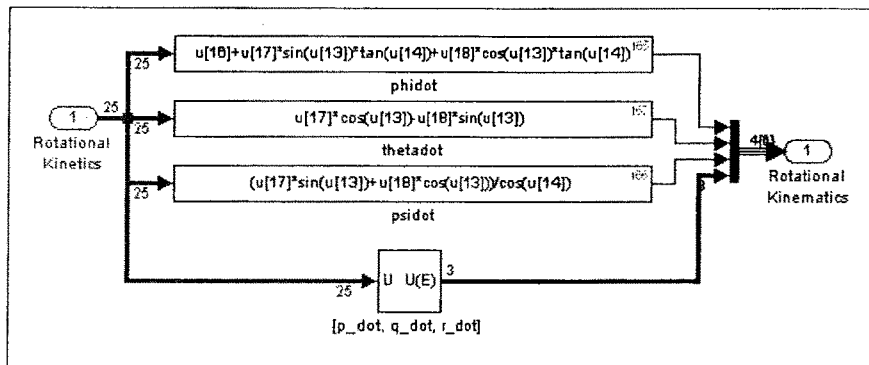


그림 3.1.12 회전방향 기구학 시뮬링크 모델

병진방향 모델과 회전방향의 모델의 운동학 모델과 기구학 모델을 통합하면 다음의 시뮬링크 모델을 얻는다.

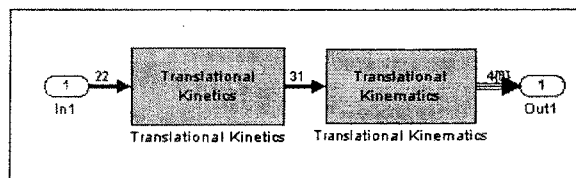


그림 3.1.13 병진방향 시뮬링크 모델

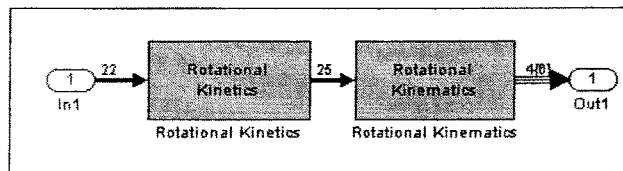


그림 3.1.14 회전진방향 시뮬링크 모델

위의 블록을 모두 통합하여 NASP의 동역학 모델을 다음과 같이 구성하였다.

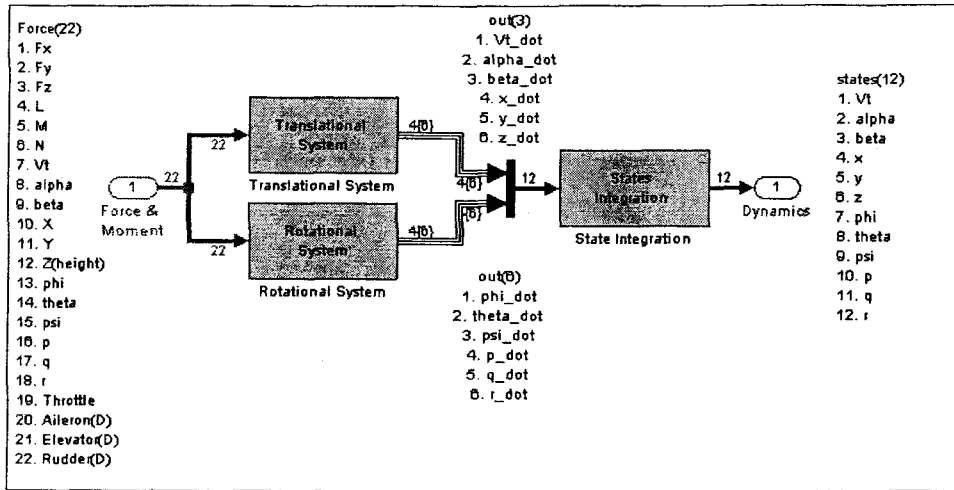


그림 3.1.15 NASP 동역학 시뮬링크 모델

라. 외력 모델링 및 구현

3차원 공간에서 움직이는 비행체는 강체로 가정하였을 경우, 병진운동과 회전운동 각각의 3자유도를 갖게 된다. 이때, 비행체에 작용하는 외력으로는 공력, 추력 및 중력을 고려할 수 있다. 또한, 비행체는 연료를 소모하면서 비행을 하게 되므로 연료 소모에 의한 질량 감소 효과도 고려할 수 있다. 이와 같은 사항을 정리하면 다음과 같다.

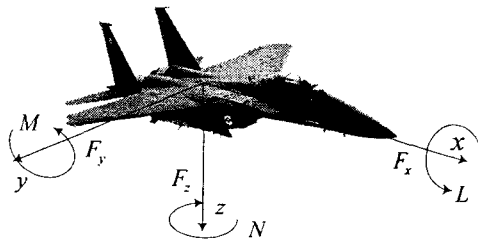


그림 3.1.16 비행체에 작용하는 외력

(1) 공력 모델

현재 연구실에 보유되어 있는 공력자료는 C 언어 형태로 구조화 되어 있다. 속도는 0 부터 최대 마하 1.6까지, 그리고 고도는 100ft부터 최대 5000ft 범위를 포함하는 비행 포화곡선을 구성할 수 있는 공력자료를 갖추고 있다. 공력데이터는 $C_x, C_y, C_z, C_p, C_m, C_n$ 의 출력자료 구조로 되어 있다. 무차원변수인 공력계수를 이용하여 다음의 공력계수 블록을 생성하였다.

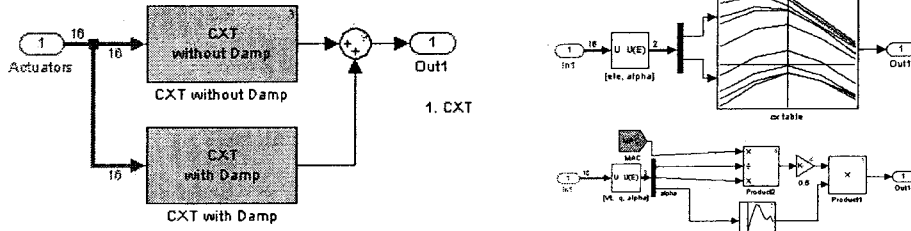


그림 3.1.17 공력계수 구성(Cx)

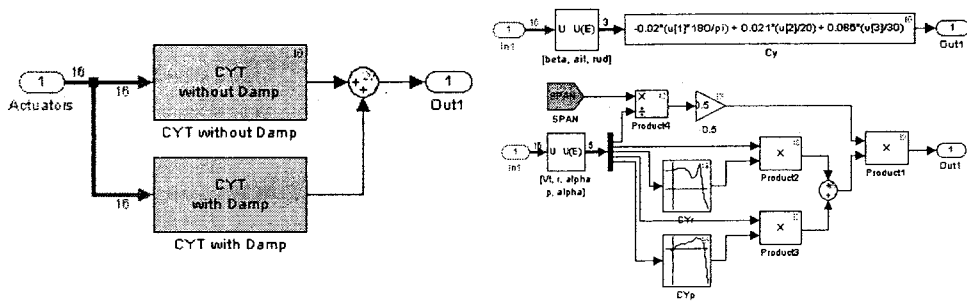


그림 3.1.18 공력계수 구성(Cy)

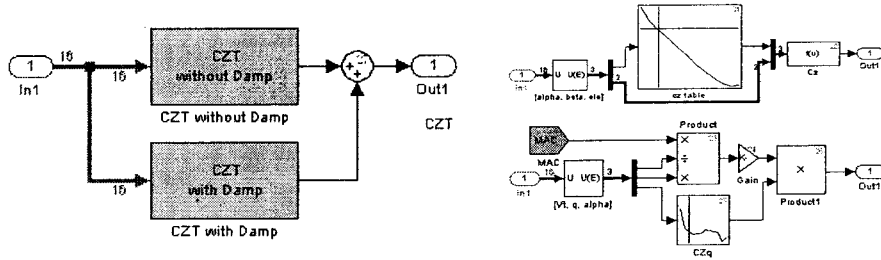


그림 3.1.19 공력계수 구성(Cz)

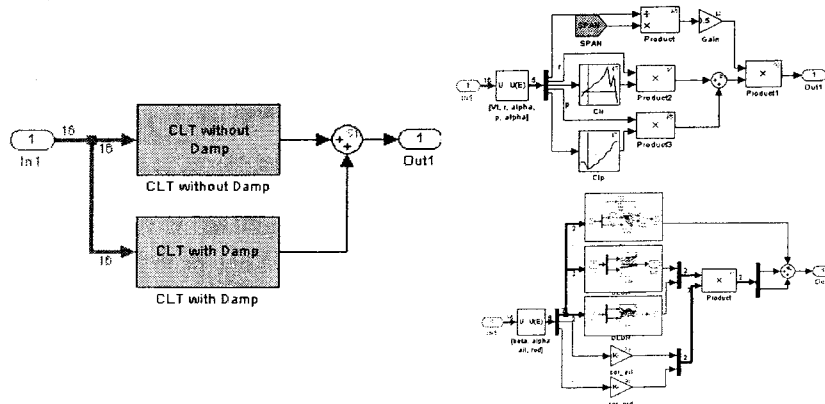


그림 3.1.20 공력계수 구성(Ci)

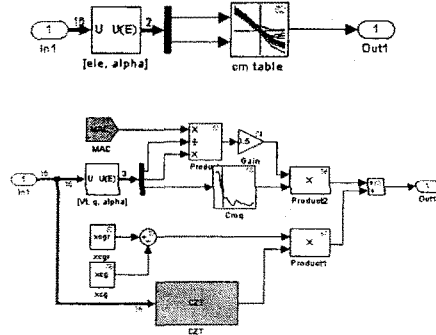
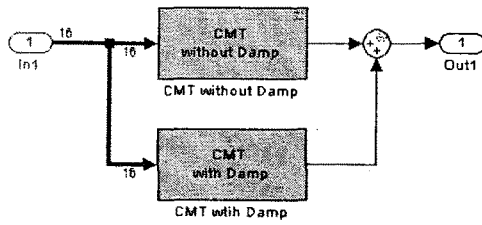


그림 3.1.21 공력계수 구성(Cm)

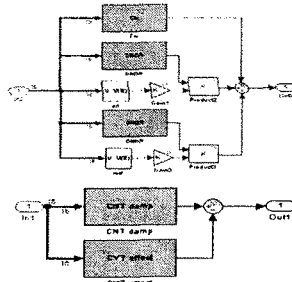
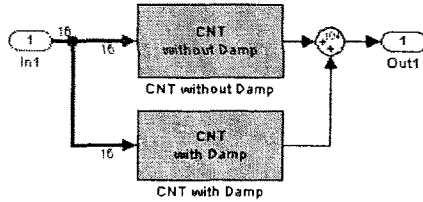


그림 3.1.22 공력계수 구성(Cn)

위의 공력계수를 병진방향과 회전방향 향으로 나누어 정리하여 다음 블록과 같이 구성하였다.

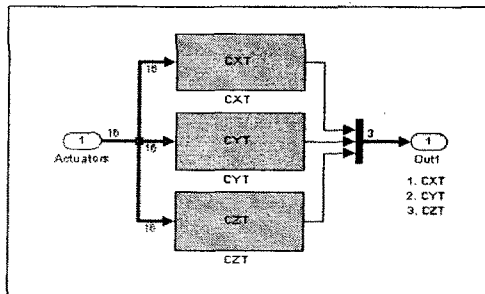


그림 3.1.23 공력계수 구성(병진방향)

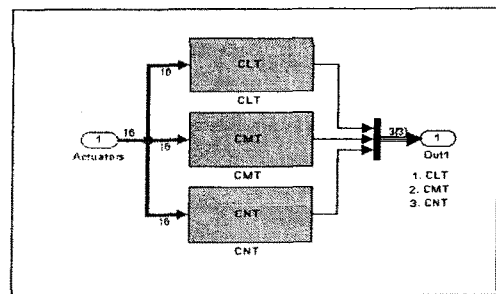


그림 3.1.24 공력계수 구성(회전방향)

공력계수들은 무차원 계수이므로 이를 이용하여 운동방정식에 외력으로 부가하기 위해서는 유차원화가 필요하고 이를 시플링크로 구성하면 다음과 같다.

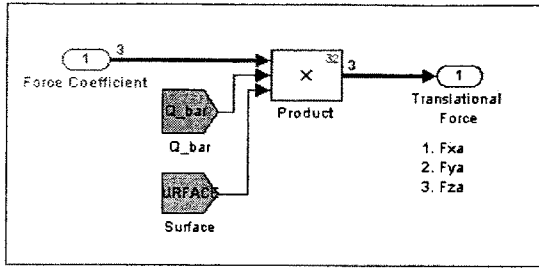


그림 3.1.25 공력계수 유차원화(병진방향)

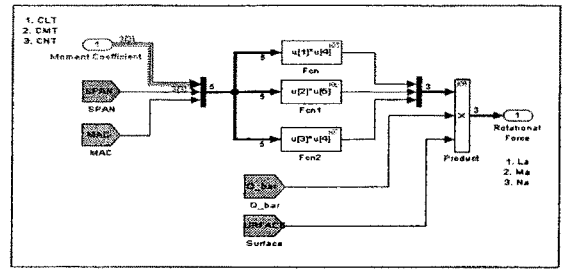


그림 3.1.26 공력계수 유차원화(회전방향)

위의 시뮬링크 블록들을 모두 정리하여 공력블록을 구성한 결과는 다음과 같다.

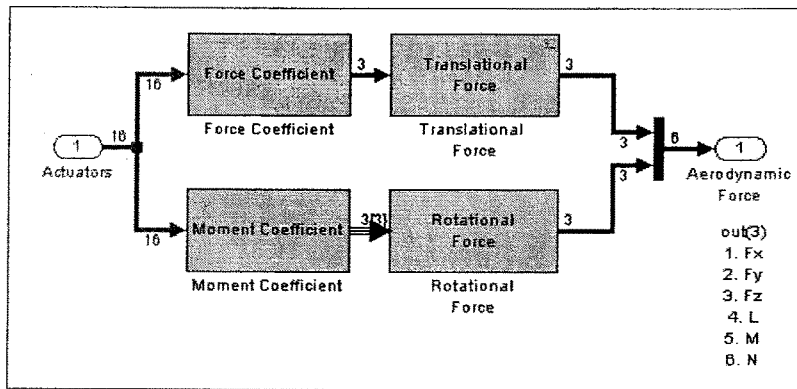


그림 3.1.27 공력 모델링

(2) 추력 모델

엔진에 의해 발생한 외력을 모델링하였다. 엔진은 기계적 시스템으로 정확한 모델링이 매우 어려운 것으로 알려져 있다. 본 연구에서는 참고문헌의 자료를 이용하여 1차 지연 엔지 모델링을 수행하였다. F-16의 엔진은 터보엔진이므로 엔진자체에 회전모멘트 성분이 포함되어 있다.

엔진의 추력 모델은 비행체의 속도와 고도를 입력으로 하며, ideal, military, maximum 모델에 의한 추력값을 조종사의 명령과 적절히 결합하여 최종 비행 추력을 산출해 낸다. 이를 시뮬링크로 구성한 결과는 다음과 같다.

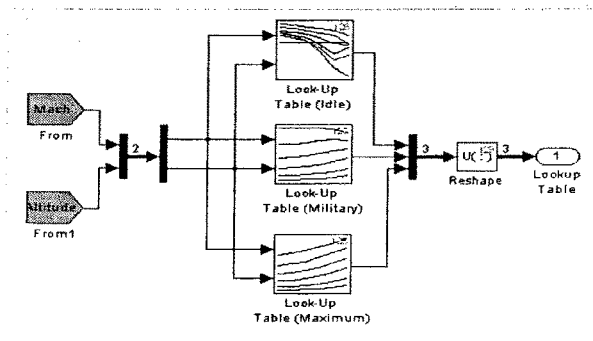


그림 3.1.28 추력계수 모델

조종사의 입력은 1차 시간지연을 갖으며 엔진에 명령으로 부가되며, 이 명령은 기어구조를 통해 비선형적으로 엔진시스템에 부가되도록 구성되어 있다. 이를 시뮬링크 블록으로 구성하면 다음과 같은 엔진 모델을 만들 수 있다.

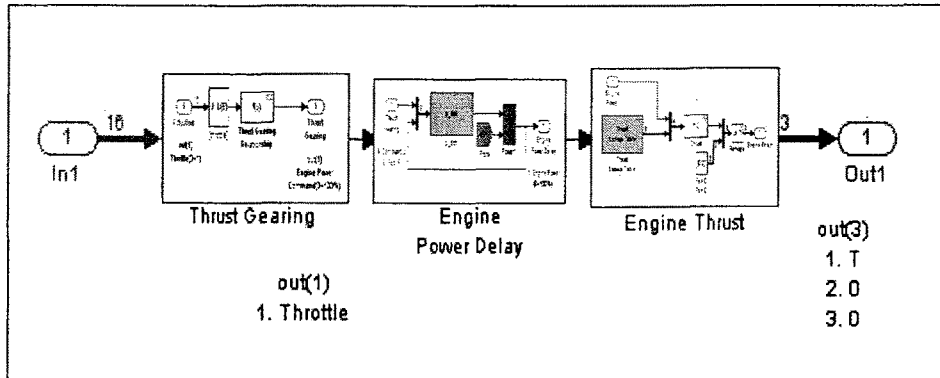


그림 3.1.29 엔진에 의한 추력 모델

한편 엔진에 의해 시스템에 부가되는 회전방향 모멘트는 다음과 같이 정식화 된다.

$$\begin{aligned}
 L_e &= qh_z - rh_y \\
 M_e &= rh_x - ph_z \\
 N_e &= ph_y - qh_x
 \end{aligned}
 \tag{3.1.24}$$

이 모델을 이용하여 시뮬링크를 구성하면 다음과 같다.

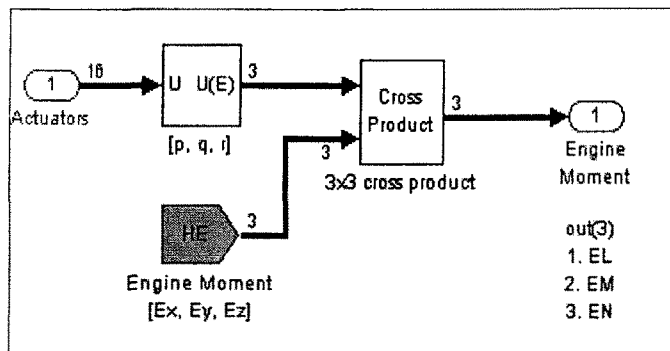


그림 3.1.30 엔진에 의한 모멘트 모델

엔진에 의한 추력과 모멘트를 엔진 모델로 구성하면 엔진에 의한 외력을 계산할 수 있으며, 이를 시뮬링크로 구성하면 다음과 같다.

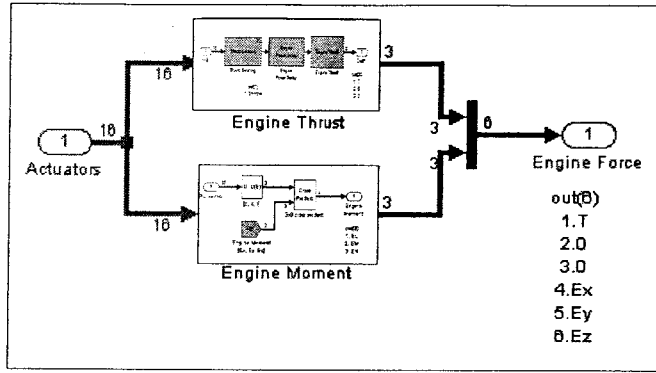


그림 3.1.31 엔진에 의한 외력 모델링

(3) 중력 모델

중력가속도는 고도가 상승할수록 작아지며, 아래의 식과 같이 모델링 된다.

$$g = g_0 \frac{R_E^2}{(R_E + h)^2} \quad (3.1.25)$$

이때 R_E 는 지구 반경이며, h 는 지표면부터 항공기까지의 고도이다. 계산된 중력가속도는 다음과 같이 시스템 운동방정식에 추가된다.

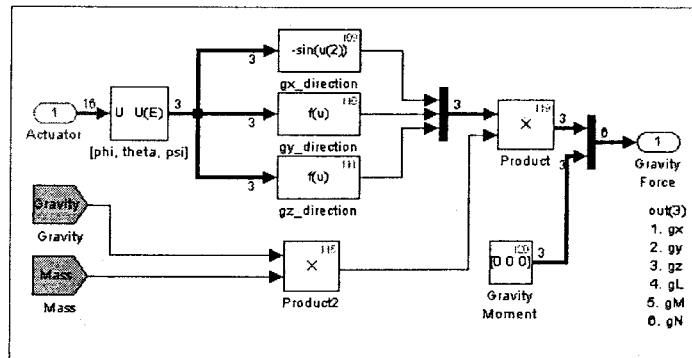


그림 3.1.32 중력에 의한 외력 모델링

비행체에 고려하는 외력은 공력, 추력 및 중력으로 구성하였으며, 이 전체 구조는 다음과 같이 구성되어 있다.

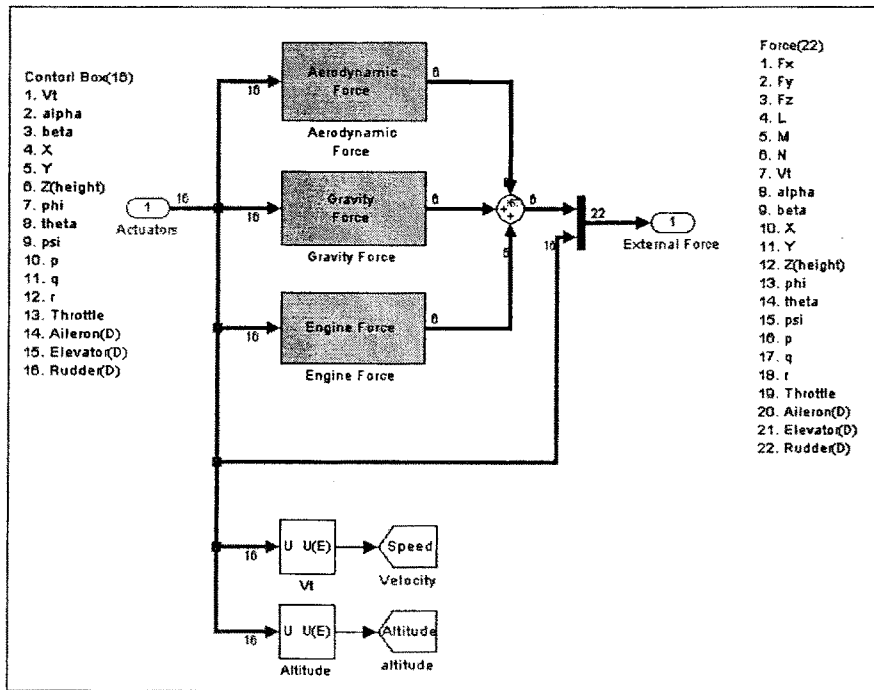


그림 3.1.33 외력 모델링

마. 외부환경 모델

비행체는 대기에서 기동을 수행한다. 공력의 경우 대기의 상태에 많은 영향을 받으며, 대부분의 비행데이터 역시 대기상태에 민감한 경향을 띠고 있다. 따라서 보다 사실적인 시뮬레이션을 수행하기 위해서는 외부 대기 모델에 대한 모델링을 수행해야 한다.

(1) 질량 및 관성 모멘트 모델

비행체의 무게는 약 60%의 연료와 탑재체를 포함하고 있다. 비행시간이 길어지면 추력 양에 따라 질량이 변화하므로 추력에 따른 적절한 질량 모델을 설정해야만 한다. 그러나 제어기가 이루어지는 시점이 1분 이내 이므로 추력에 따른 질량 및 관성모멘트 변화를 무시하여 모델링 할 수도 있다.

(2) 대기 모델

일반적으로 대기는 고도가 상승할수록 밀도가 떨어지는 함수형태로 정식화된다. 한편, 공기의 온도는 대기층에 따라 다른 것으로 알려져 있다. 대기온도 모델은 그림 3.1.34와 같다.

$$\rho(h) = \rho_0 e^{-h/H} \quad (3.1.26)$$

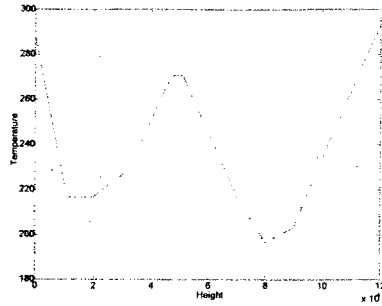


그림 3.1.34 대기온도 모델

구성된 외부 환경 모델 블록은 다음과 같다.

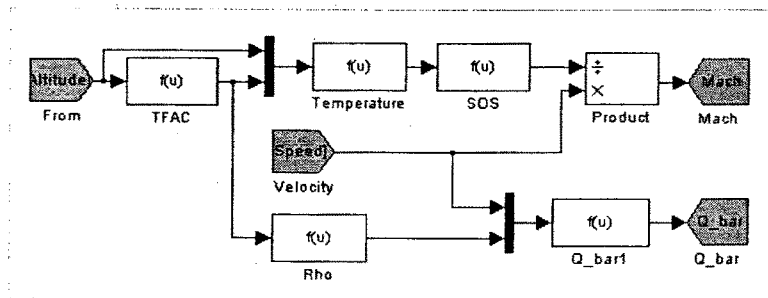


그림 3.1.35 외부환경 모델링

(3) 바람 모델

대기 상에 불규칙적으로 작용하는 바람은 항공기 시스템에 외란으로 작용하게 된다. 이와 같이 불규칙적인 바람은 통계적으로 분석되어 고도에 따른 평균 풍속 모델이 작성되어 있으며, 본 연구에서는 이 결과를 사용한다. 바람은 전단 방향으로의 바람모델, 규칙이 없는 모델, 거스트(Gust) 모델 등에 대한 조사를 통하여 모델을 구성하고 있다.

바. Matlab Simulink를 이용한 NASP 구현

설계된 비행제어시스템의 타당성을 검증하기 위해 HILS를 사용하는 경우 시스템의 실시간 성능 향상을 위해 C code를 이용하여 NASP를 구성하는 것이 바람직하다 판단된다. 한편 Matlab은 매우 편리한 제어시스템 개발 툴로써 제어 알고리즘 개발 시 매우 효과적이다. 따라서 보다 효과적이고 능률적인 시스템 개발 환경을 구축하기 위해 Matlab Simulink를 이용하여 NASP개발을 수행하였다. Simulink는 시스템 블록을 이용하여 시스템 모델을 구현하는 시각적인 제어시스템 개발도구로 널리 사용되는 상용프로그램이다. 특히 xPC 컴파일러를 이용하여 시스템을 컴파일 하는 경우 의사 실시간 코드를 자체적으로 생성하여 실행 화일을 작성하므로, 매우 손쉽게 개발된 시스템을 하드웨어에 적용할 수 있다.

본 연구에서 구성될 6자유도 비선형 시뮬레이터는 고성능 항공기모델을 기반으로 하고 있으며, 그림 3.1.36는 본 연구에서 고려하고 있는 NASP의 입출력 자료들을 나타내고 있으며, 그림 3.1.37은 1단계 연구수행을 통해 개발된 Simulink를 이용하여 구현된 NASP의 전체 구조도이다.

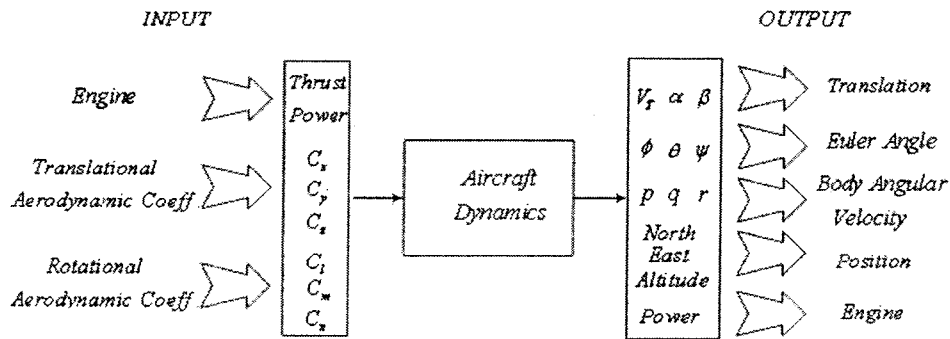


그림 3.1.36 입출력 관계

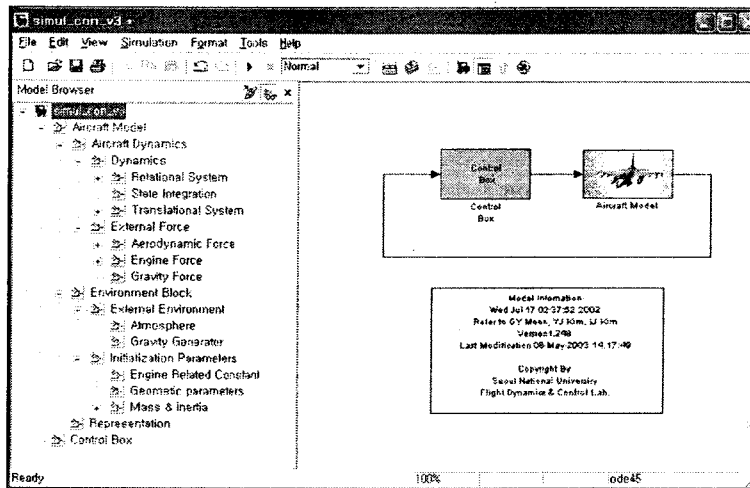


그림 3.1.37 Simulink를 이용한 NASP

3. Mid-level 비행제어 알고리즘 연구

가. NASP를 이용한 Mid-level 비행제어연구

Mid-level 비행제어 시스템은 High-level 비행제어시스템으로 가기위한 전 단계로서 비행시스템 안정성을 보장하는 제어를 구축하고 이러한 제어기의 성능을 향상시키는 기법에 대한 연구를 수행한다. 기존의 비행제어 설계 기법은 비선형 시스템인 항공시스템을 작동점 부근에서 선형화 시켜 선형 모델을 산출한 후 이 선형 모델을 이용하여 고전제어기법이나 선형제어 이론을 적용하여 시스템 안정화를 수행하였다. 이와 같은 제어시스템 개발은 그동안 많은 연구가 수행되었고 기술이 정착된 분야이다. 그러나 이같은 제어기는 시스템에 고장이 발생하거나 이상현상이 발생하여 가정하였던 동작점 혹은 선형 시스템이 변하는 경우 설계된 제어기에 의해 오히려 안정성이 악화되는 경우도 있다.

본 연구에서 연구 개발하고자 하는 비행제어 시스템은 이러한 Mid-level의 제어기 보다 수준이 높은 High-level 제어이며, 시스템에 고장이 난 경우에도 비행체의 귀환이 가능하며, 고장의 인식 및 분리를 통해 시스템 제어기를 스스로 재형상하는 등의 지능적이고 통합적인 제어기 설계를 목표로 한다. 따라서 설계된 제어시스템은 mid-level 제어시스템보다

우수한 성능을 갖게 될 것이다. 그러나 제안된 제어기가 기존의 제어시스템 보다 성능이 우수한 것을 확인하기 위해서는 기존 제어시스템과의 비교, 평가를 수행해야 한다. 따라서 본 절에서는 앞으로 연구 개발된 high-level 제어시스템의 비교군이 될 제어시스템을 설계하는 것을 목적으로 하였다. Mid-level 비행제어 시스템은 앞서 언급한 바처럼 일반적으로 선형시스템을 가정하고 제어기 설계를 수행한다. 따라서 본 연구에서 다루고 있는 비선형 모델인 NASP을 이용하여 trim점을 산출한 후 선출된 트림점을 이용하여 선형모델을 계산하였다. 계산된 선형 모델을 이용하여 기존의 선형 제어이론을 이용한 안정성 증대장치를 구성한다. 구성된 안정성 증대장치를 NASP에 적용하여 시스템의 안정성이 향상됨을 확인하고 High-level 제어 시스템의 비교 대조군으로 사용될 수 있도록 여러 다양한 기동 특성 및 동특성을 분석하여 자료를 구조화 한다.

나. 시스템 선형화

본 시뮬레이션 프로그램은 비행체의 선형 모델을 제공하여, 선형이론을 바탕으로 하는 제어기 설계에 사용할 수 있도록 하였다. 이를 위해 30개의 트림점을 선정하여 선정된 지점에 대해 선형모델을 제공하며, 각 지점에 대한 안정성 분석을 수행할 수 있다. 이를 바탕으로 고전 제어기를 설계하여 제안된 제어이론에 의한 결과와 비교 평가를 수행할 수 있다. 선정된 30개의 트림점은 다음과 같다.

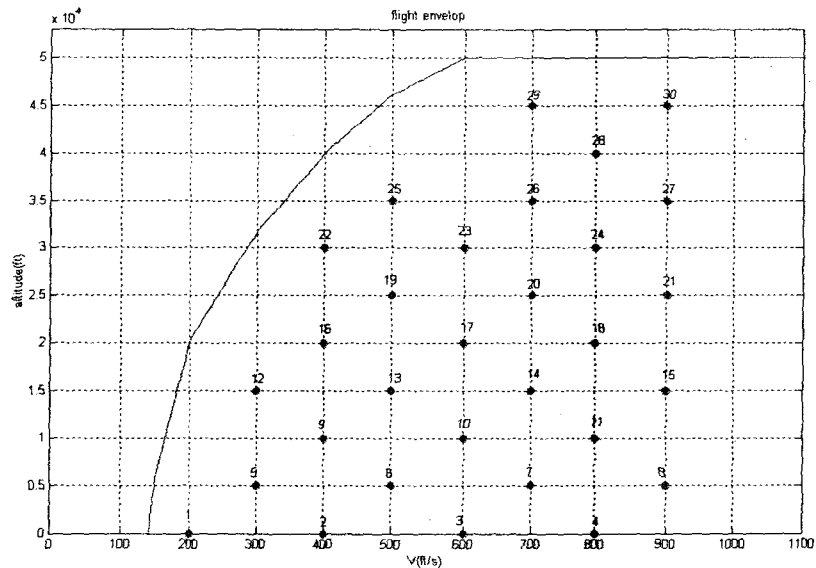


그림 3.1.38 선정된 정상수평비행 트림 점

본 연구에서 구성한 NASP는 마하 1, 고도 50000피트의 한계를 갖고 있는 공력모델이다. 이 영역 안에서 비행포화곡선을 산출하기 위해서는 트림점 계산을 수행하여 트림조건을 찾을 수 없는 영역을 찾아야 하며, 이와 같은 방식으로 비행포화곡선의 경계 영역을 찾아내었다. 이에 대한 자세한 사항은 아래에서 설명하기로 한다.

(1) 트림(trim)상태

평형상태란(equilibrium state) 시간이 지나도 그 평형상태를 유지하는 상태를 의미한다. 평형상태는 불안정한 평형상태와 안정한 평형상태로 나누어 생각해 볼 수 있는데, 시스템이 불안정한 경우에 찾게 되는 평형점이 불안정한 평형상태이며, 시스템이 안정화한 영역에서 찾게 되는 평형점이 안정화 평형상태이다. 시스템의 안정성 여부를 떠나서 평형상태를 찾을 수 있으며, 이는 수학적으로 시간에 대한 변화량이 0인 상태점을 의미한다. 항공기에서 이러한 평형 상태를 트림점이라 칭하며 운동의 기동에 따라 각각 다른 트림점을 갖는다. 본 연구에서 고려하는 기동은 정상 수평비행 기동이다.

정상수평비행의 트림상태는 물리적으로 비행체가 직선 기동을 하는 것을 의미한다. 비행체가 일정속도로 직선 기동을 수행하기 위해서는 식(3.2.2) - 식(3.2.5)의 미분항이 모두 0이 되어야 하며 $\beta=0, p=0, q=0, r=0, \phi=0, \theta=0, \psi=0$ 의 조건을 만족해야 한다.

(2) 트림상태 계산

비행시스템은 공간상에서 기동하게 되므로 비선형 6자유도 미분방정식의 형태의 운동방정식을 지배방정식으로 갖게 된다. 운동방정식은 운동학 식 6개, 기구학 식 6개로 총 12개의 식이며, 구해야할 변수는 외력에 영향을 주는 제어면의 편향각 및 엔진 추력이다. 이 같은 비선형 연립방정식 문제는 미지수의 개수와 연립식의 개수에 따라 최소제곱방법이나 최적해를 이용한 문제 풀이등의 해법을 갖는다. 최소제곱방법은 결정해야 할 미지수의 개수가 연립식의 개수보다 많아, 연립식과 최소한으로 벗어나는 해를 찾아가는 기법이다. 반면 최적해를 이용한 문제 풀이법은 결정할 미지수의 개수가 연립식의 개수보다 많아 만족해야 할 연립식은 모두 만족하게 설정된다. 이때 성능지수 최소화 문제인 경우 문제의 특성상 주어진 성능지수를 최소화 하는 방향으로 주어진 구속조건을 만족시키며, 문제의 해를 찾게 된다. NASP에서 트림을 계산하는 경우는 후자에 해당하는 것으로, 12개의 운동방정식 중 6개는 종속적인 관계로 식을 풀 필요가 없으며, 남은 관계식들 중 지정된 기동을 묘사하기 위해 구속조건을 부가하는 상태 변수들의 개수 및 각종 변수의 개수는 비행 역학식에 의한 6개 이상의 변수를 요구한다. 따라서 트림계산을 위해서는 최적 이론에 의해 문제 풀이에 접근해야 하며, 본 연구에서는 수치 구배 기법(gradient method)과 SQP(Sequential Quadratic Programming)을 각각 이용하여 트림점을 계산한 다음, 각각의 결과를 서로 비교하여 오류를 최소화할 수 있는 방법을 선택하였다.

수치 구배 기법은 운동방정식의 구속조건을 만족시키면서 식(3.1.27)과 같은 성능지수를 최소화시키는 제어입력을 찾는 수치기법이다.

$$J = L(x, u) \tag{3.1.27}$$

$$\text{subject to } f(x, u) = 0$$

위 성능지수와 구속 방정식의 미분변화량은 다음과 같다.

$$dL = \frac{\partial L}{\partial x} dx + \frac{\partial L}{\partial u} du \tag{3.1.28}$$

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial u} du \tag{3.1.29}$$

최대값을 가지는 점(Stationary Point)에서 필요조건은 $dL=0$ 과 $df=0$ 이므로, $f=0$ 을 만족하면서 u 의 변화량에 대해서 성능지수와 구속조건의 변화량이 0이 되는 제어 입력을 찾아야 하므로, 다음과 같은 식을 만족하게 된다.

$$d\underline{x} = -\left[\frac{\partial f}{\partial \underline{x}}\right]^{-1}\left[\frac{\partial f}{\partial \underline{u}}\right]d\underline{u} \quad (3.1.30)$$

$$\frac{\partial L}{\partial \underline{u}} - \frac{\partial L}{\partial \underline{x}}\left[\frac{\partial f}{\partial \underline{x}}\right]^{-1}\left[\frac{\partial f}{\partial \underline{u}}\right] = 0 \quad (3.1.31)$$

u 의 초기값을 추정한 다음 $f=0$ 을 만족시키면서 식(3.1.31)에 의하여 \underline{x} 를 갱신하고, 식(3.1.31)을 만족하는 방향으로 \underline{u} 를 갱신하는 알고리즘을 사용하면 된다. 그림 3.1.39에서 볼 수 있듯이 구배 벡터방향으로 계속 이동하면 최대값을 가지는 지점에 이르게 되며, 이 알고리즘을 사용하기 위해서는 성능지수 함수가 연속적이고 부분적 미분가능하여야 한다. 이 알고리즘의 단점은 성능지수의 골곡이 많아 극대값이 많이 발생할 경우, 최대값이 아닌 극대값에 수렴할 수 있다는 단점을 가지고 있다.

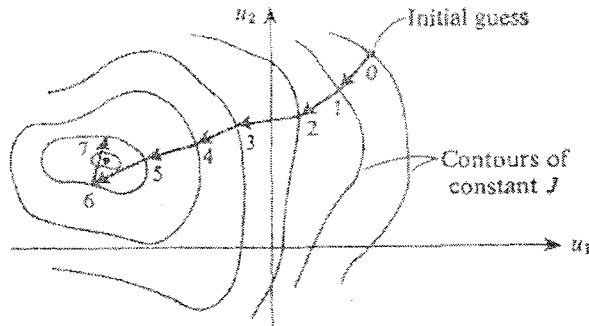


그림 3.1.39 수치구배기법이 개념

SQP도 식(3.1.27)과 같은 구속조건과 성능지수를 고려하지만, 비선형 성분을 선형화하여 평형점 근처에서의 2차 성능지수와 선형 구속조건을 만족시키는 방향으로 상태변수와 제어입력을 갱신하는 방법이다.

$$\min L(\underline{x}) = L(\underline{x}_0) + \nabla L(\underline{x}_0)\delta\underline{x} + \frac{1}{2}\delta\underline{x}^T \nabla^2 L(\underline{x}_0)\delta\underline{x} \quad (3.1.32)$$

$$\text{subject to } \underline{f}_i(\underline{x}) = \underline{f}_i(\underline{x}_0) + \nabla \underline{f}_i(\underline{x}_0)\delta\underline{x} = 0 \quad (3.1.33)$$

(3) 시스템 선형화

비선형 항공기 시스템은 다음과 같은 식으로 표현할 수 있다.

$$\dot{\underline{x}} = \underline{h}(\underline{x}, \underline{u}) \quad (3.1.34)$$

$$\underline{y} = \underline{f}(\underline{x}, \underline{u}) \quad (3.1.35)$$

여기서 h 와 f 는 비선형 함수이며, 이러한 함수 형태를 가지는 시스템의 기동은 매우 복잡하며 해석하기 어렵다. 그러나 어떤 상태 하에서는 비선형 함수를 선형으로 근사화하여 표현할 수 있다. 식(3.1.34)을 만족하는 어떠한 제어입력 \underline{u}_0 와 초기상태변수 \underline{x}_0 를 가정해보자.

$$\dot{\underline{x}}_0 = h(\underline{x}_0, \underline{u}_0) \quad (3.1.36)$$

그 다음 제어입력과 초기상태변수가 미세한 섭동(perturbation)을 받아서 $\underline{u}_0 + \Delta u$ 와 $\underline{x}_0 + \Delta x$ 가 된다면, 식(3.1.37)을 다음과 같이 전개할 수 있다.

$$\begin{aligned} \dot{\underline{x}}_0 + \Delta \dot{\underline{x}} &= h(\underline{x}_0 + \Delta \underline{x}, \underline{u}_0 + \Delta \underline{u}) \\ &= h(\underline{x}_0, \underline{u}_0) + \frac{\partial h}{\partial \underline{x}} \Delta \underline{x} + \frac{\partial h}{\partial \underline{u}} \Delta \underline{u} + H.O.T \end{aligned} \quad (3.1.37)$$

섭동량은 매우 작으므로 고차항을 무시하고 식(3.1.36)을 이용하면, 양변은 식(3.1.38)과 같이 정리되고 상태변수와 제어입력의 섭동량으로 선형화될 수 있다.

$$\Delta \dot{\underline{x}} = \frac{\partial h}{\partial \underline{x}} \Delta \underline{x} + \frac{\partial h}{\partial \underline{u}} \Delta \underline{u} \quad (3.1.38)$$

$$A = \frac{\partial h}{\partial \underline{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \frac{\partial h_1}{\partial x_3} & \dots \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \frac{\partial h_2}{\partial x_3} & \dots \\ \frac{\partial h_3}{\partial x_1} & \frac{\partial h_3}{\partial x_2} & \frac{\partial h_3}{\partial x_3} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \quad B = \frac{\partial h}{\partial \underline{u}} = \begin{bmatrix} \frac{\partial h_1}{\partial u_1} & \frac{\partial h_1}{\partial u_2} & \frac{\partial h_1}{\partial u_3} & \dots \\ \frac{\partial h_2}{\partial u_1} & \frac{\partial h_2}{\partial u_2} & \frac{\partial h_2}{\partial u_3} & \dots \\ \frac{\partial h_3}{\partial u_1} & \frac{\partial h_3}{\partial u_2} & \frac{\partial h_3}{\partial u_3} & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix} \quad (3.1.39)$$

(4) 선형 시스템 결과

위에서 언급한 과정들을 통하여 선정된 30개의 트림점에 대하여 시스템 행렬을 구할 수 있다. 여기서는 속도 800 ft/s, 고도 0에서 정상수평비행을 하는 실제 선형 트림점과 시스템 행렬을 살펴보자. 선형 트림점은 다음과 같다.

$$\underline{x}_{trim} = [800 \quad -0.0008 \quad 0 \quad 0 \quad -0.0008 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \quad (3.1.40)$$

$$\underline{u}_{trim} = [0.3779 \quad -0.9424 \quad 0 \quad 0]^T \quad (3.1.41)$$

트림점에서의 종방향 시스템 행렬은 다음과 같다.

$$A_{lon} = \begin{bmatrix} -0.0270 & -3.3682 & -32.2 & 0.8280 \\ -0.0001 & -1.7507 & 0 & 0.9085 \\ 0 & 0 & 0 & 1 \\ 0 & 6.6251 & 0 & -1.6990 \end{bmatrix} \quad (3.1.42)$$

$$B_{lon} = \begin{bmatrix} 0 & 0.5693 \\ 0 & -0.0034 \\ 0 & 0 \\ 0 & -0.4467 \end{bmatrix} \quad (3.1.43)$$

중방향 고유치는 다음과 같다.

$$A_{lon} = \begin{bmatrix} -4.1786 & 0 & 0 & 0 \\ 0 & 0.7364 & 0 & 0 \\ 0 & 0 & -0.0172 + 0.0817i & 0 \\ 0 & 0 & 0 & -0.0172 - 0.0817i \end{bmatrix} \quad (3.1.44)$$

트림점에서의 횡방향 시스템 행렬은 다음과 같다.

$$A_{lat} = \begin{bmatrix} -0.5134 & -0.0402 & -0.0023 & -0.9926 \\ 0 & 0 & 1 & -0.0008 \\ -63.5844 & 0 & -5.9767 & 0.7627 \\ 21.3992 & 0 & 0.0128 & -0.7568 \end{bmatrix} \quad (3.1.45)$$

$$B_{lat} = \begin{bmatrix} 0.0005 & 0.0013 \\ 0 & 0 \\ -1.8472 & 0.3442 \\ -0.0830 & -0.1574 \end{bmatrix} \quad (3.1.46)$$

횡방향 고유치는 다음과 같다.

$$A_{lat} = \begin{bmatrix} -0.6061 + 4.6175i & 0 & 0 & 0 \\ 0 & -0.6061 - 4.6175i & 0 & 0 \\ 0 & 0 & -6.0248 & 0 \\ 0 & 0 & 0 & -0.0098 \end{bmatrix} \quad (3.1.47)$$

다. 자동 조종 장치

자동 조종 장치의 제어이득은 시스템의 트림점에 대한 시스템 행렬을 구한 후 전체 상태 변수 피드백을 이용한 LQR(Linear Quadratic Regulator)을 이용하여 결정하였다. 다음의 선형화된 시스템에 대하여 식(3.1.50)과 같은 목적함수를 최소화하는 제어이득을 구하게 된다.

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \quad (3.1.48)$$

$$u = -kx \quad (3.1.49)$$

$$J = \frac{1}{2} \underline{x}^T(t_f) S(t_f) \underline{x}(t_f) + \frac{1}{2} \int_{t_0}^{t_f} (\underline{x}^T(t) Q \underline{x}(t) + \underline{u}^T(t) R \underline{u}(t)) dt \quad (3.1.50)$$

LQR기법에서는 Q와 R 행렬에 대한 제한조건, 즉 Q는 음이 아닌 대칭행렬, R은 양의 대칭행렬이어야 한다는 조건을 만족시키면 안정된 제어이득을 구할 수 있다. 하지만, Q와 R은 설계자가 조정할 수 있는 설계변수이므로, 그 때의 제어이득은 Q와 R에 국한되어 최적화된 결과일 수 밖에 없다. 이러한 LQR의 특징을 고려하여 본 연구에서는 Q와 R의 대각성분 값을 변화시키면서 반복계산(iteration)을 수행하여 댐핑(damping) 계수가 0.6이상인 제어이득을 선정하였다.

시스템은 3.2절에서 구한 속력 800 ft/s, 고도 0에서 정상수평비행을 하는 실제 선형 트림점의 데이터를 사용하여 LQR을 통하여 제어이득을 구하였다. 이 시스템은 다음의 고유치에서 볼 수 있듯이 제어이득이 없을 경우 종방향으로 불안정한 요소를 가지고 있다.

$$A_{lon} = \begin{bmatrix} -4.1786 & 0 & 0 & 0 \\ 0 & 0.7364 & 0 & 0 \\ 0 & 0 & -0.0172 + 0.0817i & 0 \\ 0 & 0 & 0 & -0.0172 - 0.0817i \end{bmatrix} \quad (3.1.51)$$

LQR을 통하여 구한 제어이득은 다음과 같다.

$$k_{lon} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.9742 & -11.9624 & -30.6460 & -11.3353 \end{bmatrix} \quad (3.1.52)$$

$$k_{lon} = \begin{bmatrix} 2.433 & -2.1420 & -0.5889 & -1.7195 \\ 0.0080 & -0.0816 & -0.0094 & -0.6658 \end{bmatrix} \quad (3.1.53)$$

위에서 구한 제어이득을 가지고 피드백을 걸지 않을 경우 시스템은 불안정한 트림점이기 때문에 그림 3.1.40과 같이 시간이 지나면 정상수평비행을 하지 못하고 발산하는 경향을 보이게 된다.

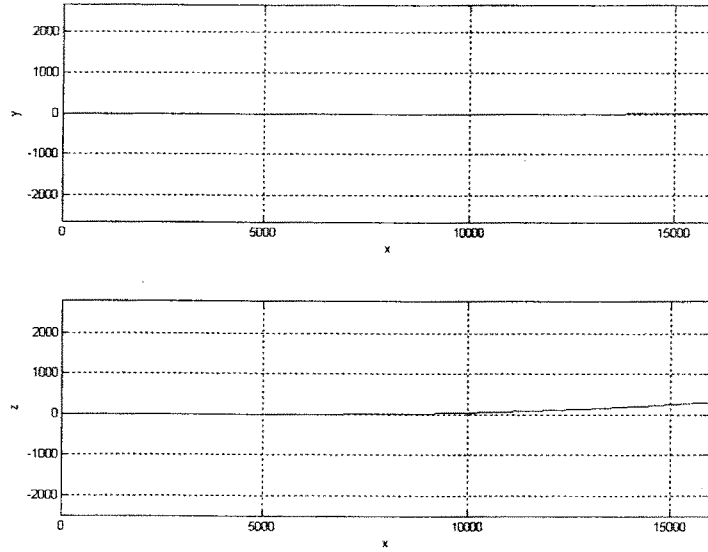


그림 3.1.40 상태변수 피드백을 걸지 않았을 경우의 시스템 궤적

그러나 LQR로 구한 제어이득을 가지고 전체 상태변수 피드백을 걸어주면 그림 3.1.41와 같이 정상수평비행을 하는 것을 살펴볼 수가 있다.

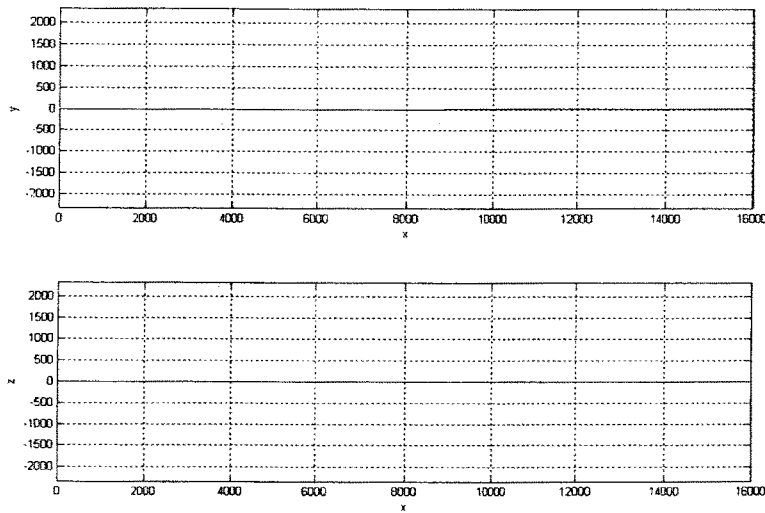


그림 3.1.41 상태변수 피드백을 걸었을 경우의 시스템 궤적

모든 트림점에 대해서 본 연구를 수행하여 제어이득을 얻었고 그 결과를 선형 보간하여 제어이득 스케줄링(scheduling)을 수행하였다.

4 Mid-level 비행제어법칙 알고리즘 개선

가. 르야프노프 이론을 이용한 목표각 추종 유도법칙 설계

현재까지 다양한 형태의 유도법칙이 개발되어 왔다. 유도법칙은 크게 3가지로 분류할 수 있다. 첫 번째는 일반적으로 잘 알려진 시선벡터(Line of sight vector)에 기반을 둔 유도법칙이고, 두 번째는 일정 방위경로(constant bearing course) 유도법칙이다. 일정 방위경로 유도법칙은 미사일과 표적의 충돌경로를 고려하여 미사일을 항상 충돌삼각형 상에 위치시키려는 유도법칙으로, 미사일의 비행방향각(heading angle) 오차를 줄임으로써 표적을 요격하고자 하는 방식이다. 세 번째 유도법칙은 표적의 기동을 예측하여 요격점을 계산한 뒤, 미사일이 종말시간에 예상 요격점에 도달하도록 하는 예측 유도법칙에 기반한 접근방식이다.

(1) 운동방정식

미사일과 표적의 운동방정식은 다음과 같고, 기하학적 배치는 그림 3.1.42에 도시하였다.

$$\dot{R} = V_T \cos(\beta - \theta_T) - V_M \cos(\beta - \theta_M) \quad (3.1.54)$$

$$\dot{\beta} = \frac{1}{R} [-V_T \sin(\beta - \theta_T) + V_M \sin(\beta - \theta_M)] \quad (3.1.55)$$

$$\dot{\theta}_M = \frac{a_M}{V_M} = a_C \quad (3.1.56)$$

$$\dot{\theta}_T = \frac{a_T}{V_T} = a_D \quad (3.1.57)$$

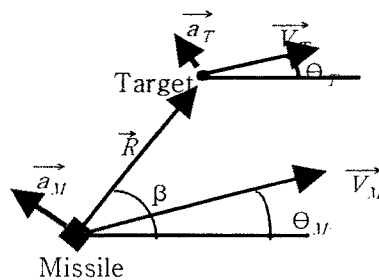


그림 3.1.42 . 2차원 추적상황

(2) 목표각을 이용한 유도법칙의 설계

본 절에서는 미사일과 표적의 충돌코스에서 정의되는 목표각의 개념을 도입하고 목표각을 추종하는 유도법칙을 제안한다. 그림 3.1.43는 미사일과 표적의 충돌삼각형을 나타내고 있다.

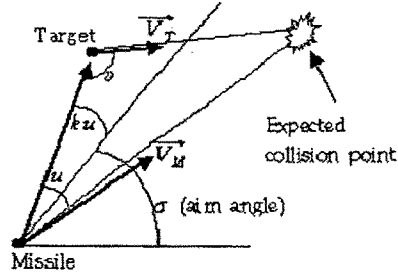


그림 3.1.43 충돌삼각형과 목표각

이 때, 미사일의 속도벡터가 충돌코스에 존재한다면 다음과 같은 관계식이 성립한다.

$$v = \pi - \text{sgn}[\beta - \theta_T] (\beta - \theta_T) \quad (3.1.58)$$

$$V_M \sin(u) = V_T \sin(v) \quad (3.1.59)$$

위 식을 시간에 대해 미분하여 정리하면 다음과 같다.

$$\dot{v} = -\text{sgn}[\beta - \theta_T] (\dot{\beta} - \dot{\theta}_T) \quad (3.1.60)$$

$$\dot{u} = -\text{sgn}[\beta - \theta_T] \frac{V_T \cos(v)}{V_M \cos(u)} (\dot{\beta} - \dot{\theta}_T) \quad (3.1.61)$$

그림 2의 충돌삼각형상에서 정의된 각 σ 를 목표각이라 하고, 다음과 같이 정의한다.

$$\sigma = \beta - k u \quad (0 \leq k \leq 1) \quad (3.1.62)$$

여기서 k 는 설계자에 의해 주어지는 값으로, 0에서 1사이의 값을 가질 수 있다. 목표각 σ 와 미사일 속도벡터 사이의 각을 목표각 오차 ΔS 라 하면, ΔS 는 다음과 같이 표현된다.

$$\Delta S = \sigma - \theta_M = \beta - k u - \theta_M \quad (3.1.63)$$

목표각을 추종하는 유도법칙을 설계하기 위해 르야프노프 후보 함수를 다음과 같이 정의하자.

$$V = \frac{1}{2} \Delta S^2 = \frac{1}{2} (\beta - k u - \theta_M)^2 \quad (3.1.64)$$

위 식을 미분한 뒤, 식 (3.1.61)을 대입하면 다음과 같다.

$$\dot{V} = (\beta - k u - \theta_M) \{ (1 + k d_1) \dot{\beta} - k d_1 \dot{\theta}_T - \dot{\theta}_M \} \quad (3.1.65)$$

여기서 $d_1 = \text{sgn}[\beta - \theta_T] \frac{V_T \cos(v)}{V_M \cos(u)}$ 이다. 미사일 자동조종장치의 동역학을 고려하지 않았으므로

$\ddot{\theta}_M = \ddot{e}$ 가 성립하고, 이를 식 (3.1.65)에 대입하면 다음과 같다.

$$\dot{V} = (\beta - ku - \theta_M) \{ (1 + kd_1) \dot{\beta} - kd_1 \ddot{\theta}_T - a_c \} \quad (3.1.66)$$

위 식에서 $\dot{V} \leq 0$ 이 성립하도록 유도 명령 a_c [유도법칙 I]을 다음과 같이 결정할 수 있다.

$$a_c = (1 + kd_1) \dot{\beta} - kd_1 \ddot{\theta}_T + N(\beta - ku - \theta_M) \quad (3.1.67)$$

여기서 N 은 항법상수이다. 위 유도법칙 I은 다음 정리 3.4.1로 요약할 수 있다.

정리 3.4.1. 식 (3.1.67)로 표시된 유도법칙 I에 대하여, 목표각은 지속적으로 감소하며, 그 반응은 가진력이 없는 1차 시스템과 같다.

(증명) 식 (3.1.67)을 식 (3.1.66)에 대입하여 정리하면 다음과 같다.

$$\dot{V} = -N(\beta - ku - \theta_M)^2 \quad (3.1.68)$$

식 (3.1.64)을 시간에 대해 미분하면 다음과 같다.

$$\dot{V} = (\beta - ku - \theta_M) \frac{d}{dt} (\beta - ku - \theta_M) \quad (3.1.69)$$

식 (3.1.68)와 식 (3.1.69)으로부터 다음 관계식을 얻을 수 있다.

$$\frac{d}{dt} (\beta - ku - \theta_M) + N(\beta - ku - \theta_M) = 0 \quad (3.1.70)$$

위 식은 목표각 오차가 가진 없는 1차 시스템의 반응을 보이면서 지속적으로 0으로 수렴함을 의미한다. (증명끝)

(3) 자동조종장치 동역학을 고려한 유도법칙의 설계

자동조종장치가 빠른 동역학을 가지고 있다면 1차 시스템 반응과 유사하지만, 자동조종장치가 느릴 경우에는 유도법칙의 성능저하가 발생할 수 있다. 이 절에서는 자동조종장치와 구동기의 전체 동역학이 다음의 1차필터 형태를 따른다고 가정하였다.

$$\ddot{\theta}_M = -\tau \dot{\theta}_M + \tau a_c \quad (3.1.71)$$

시선각 변화율을 나타내는 식 (3.1.55)를 시간에 대해 미분한 뒤, 식 (3.1.54)을 사용하여 정리하면 다음 식을 얻을 수 있다.

$$\ddot{\beta} = -2\frac{R}{R}\ddot{\beta} - c_1\ddot{\theta}_M + c_2\ddot{\theta}_T \quad (3.1.72)$$

여기서 $c_1 = \frac{V_M \cos(\beta - \theta_M)}{R}$, $c_2 = \frac{V_T \cos(\beta - \theta_T)}{R}$ 이다. 또한, 식 (3.1.64)을 미분한 뒤 식 (3.1.62)에 대입하여 정리하면 다음과 같다.

$$\Delta \dot{S} = (1 + kd_1)\dot{\beta} - kd_1\ddot{\theta}_T - \ddot{\theta}_M \quad (3.1.73)$$

위 식을 한 번 더 미분하면 다음과 같다.

$$\Delta \ddot{S} = (1 + kd_1)\ddot{\beta} + k\dot{d}_1(\dot{\beta} - \ddot{\theta}_T) - kd_1\ddot{\theta}_T - \ddot{\theta}_M \quad (3.1.74)$$

식 (3.1.65)에서 정의된 d_1 을 고려하면, 식 (3.1.74)의 우변 두 번째 항은 다음과 같이 쓸 수 있다.

$$\begin{aligned} zk\dot{d}_1(\dot{\beta} - \ddot{\theta}_T) &= k\frac{d}{dt} \left\{ \text{sgn}(\beta - \theta_T) \frac{V_T \cos(\nu)}{V_M \cos(u)} \right\} (\dot{\beta} - \ddot{\theta}_T) \\ &= k \text{sgn}(\beta - \theta_T) \frac{V_T}{V_M} \left\{ -\frac{\sin(\nu)}{\cos(u)} \dot{\nu} + \frac{\sin(u) \cos(\nu)}{\cos^2(u)} \dot{u} \right\} (\dot{\beta} - \ddot{\theta}_T) \end{aligned} \quad (3.1.75)$$

식 (3.1.60)과 식 (3.1.61)을 식 (3.1.75)에 대입하면 다음 식을 얻을 수 있다.

$$\begin{aligned} k\dot{d}_1(\dot{\beta} - \ddot{\theta}_T) &= k \text{sgn}(\beta - \theta_T) \frac{V_T}{V_M} \left\{ -\frac{\sin(\nu)}{\cos(u)} [-\text{sgn}(\beta - \theta_T)] (\dot{\beta} - \ddot{\theta}_T) \right. \\ &\quad \left. + \frac{\sin(u) \cos(\nu)}{\cos^2(u)} \left[-\text{sgn}(\beta - \theta_T) \frac{V_T \cos(\nu)}{V_M \cos(u)} \right] (\dot{\beta} - \ddot{\theta}_T) \right\} (\dot{\beta} - \ddot{\theta}_T) \\ &= kc_3(\dot{\beta} - \ddot{\theta}_T)^2 \end{aligned} \quad (3.1.76)$$

여기서 $c_3 = \left(\frac{V_T}{V_M}\right) \left[\frac{\sin(\nu)}{\cos(u)}\right] - \left(\frac{V_T}{V_M}\right)^2 \left[\frac{\sin(u) \cos^2(\nu)}{\cos^3(u)}\right]$ 이다. 식 (3.1.76)을 식 (3.1.74)에 대입하면 다음 식을 얻을 수 있다.

$$\Delta \ddot{S} = (1 + kd_1)\ddot{\beta} - kd_1\ddot{\theta}_T + kc_3(\dot{\beta}^2 - 2\dot{\beta}\ddot{\theta}_T + \ddot{\theta}_T^2) - \ddot{\theta}_M \quad (3.1.77)$$

또한, 식 (3.1.73)을 식 (3.1.77)에 대입하면 다음과 같다.

$$\begin{aligned} \Delta \ddot{S} &= \left[-2(1 + kd_1)\frac{R}{R} \right] \ddot{\beta} - c_1(1 + kd_1)\ddot{\theta}_M + c_2(1 + kd_1)\ddot{\theta}_T \\ &\quad - kd_1\ddot{\theta}_T + kc_3\dot{\beta}^2 - 2kc_3\dot{\beta}\ddot{\theta}_T + kc_3\ddot{\theta}_T^2 - \ddot{\theta}_M \end{aligned} \quad (3.1.78)$$

식 (3.1.74)과 식 (3.1.78)를 이용하면 다음 관계식을 얻을 수 있다.

$$\begin{aligned} \Delta \dot{S} + 2\zeta\omega_n \Delta \dot{S} + \omega_n^2 \Delta S = & -\bar{\Theta}_{11} - kd_1 \bar{\Theta}_T + kc_3 \bar{\beta}^2 - 2kc_3 \bar{\beta} \bar{\Theta}_T + kc_3 \bar{\Theta}_T^2 \\ & + (1+kd_1) \left[-2\frac{\bar{R}}{R} + 2\zeta\omega_n \right] \bar{\beta} - [c_1(1+kd_1) + 2\zeta\omega_n] \bar{\Theta}_{11} + [c_2(1+kd_1) \\ & - 2\zeta\omega_n kd_1] \bar{\Theta}_T + \omega_n^2 (\beta - ku - \Theta_{11}) \end{aligned} \quad (3.1.79)$$

이제 다음과 같은 유도법칙 II를 제안한다.

$$a_c = N_1(\beta - ku + \Theta_{11}) + N_2 \bar{\beta} + N_3 \bar{\Theta}_T + N_4 \bar{\Theta}_{11} + N_5 \bar{\beta}^2 + N_6 \bar{\beta} \bar{\Theta}_T + N_7 \bar{\Theta}_T^2 \quad (3.1.80)$$

유도법칙에서의 시변항법계수 $N_1 \sim N_7$ 는 정리 3.4.2를 이용하여 결정한다.

정리 3.4.2. 유도법칙을 나타내는 식 (3.1.80)에서의 계수를 다음과 같이 설정하자.

$$N_1 = \frac{1}{\tau} \omega_n^2 \quad (3.1.81)$$

$$N_2 = \frac{1}{\tau} (1+kd_1) \left[-2\frac{\bar{R}}{R} + 2\zeta\omega_n \right] \quad (0 \leq \zeta \leq 1) \quad (3.1.82)$$

$$N_3 = \frac{1}{\tau} [(1+kd_1)c_2 - 2\zeta\omega_n kd_1] \quad (3.1.83)$$

$$N_4 = \frac{1}{\tau} [-(1+kd_1)c_1 - 2\zeta\omega_n] + 1 \quad (3.1.84)$$

$$N_5 = \frac{1}{\tau} kc_3 \quad (3.1.85)$$

$$N_6 = -\frac{1}{\tau} 2kc_3 \quad (3.1.86)$$

$$N_7 = \frac{1}{\tau} kc_3 \quad (3.1.87)$$

식 (3.1.64)에서 정의된 목표각 오차 ΔS 는 댐핑이 ζ 이고, 고유진동수가 ω_n 인 이차시스템의 반응을 따른다. 이 이차시스템에 대하여 $kd_1 \bar{\Theta}_T$ 항만이 가진력으로 작용한다.

(증명) 식 (3.1.80)을 식 (3.1.71)에 대입하여 정리하면 다음과 같다.

$$-\bar{\Theta}_{11} + \tau(N_4 - 1)\bar{\Theta}_{11} + \tau N_1(\beta - ku - \Theta_{11}) + \tau N_2 \bar{\beta} + \tau N_3 \bar{\Theta}_T + \tau N_5 \bar{\beta}^2 + \tau N_6 \bar{\beta} \bar{\Theta}_T + \tau N_7 \bar{\Theta}_T^2 = 0 \quad (3.1.88)$$

식 (3.1.81)-(3.1.87)을 식 (3.1.88)에 대입하여 식 (3.1.79)과 비교하면 다음을 얻을 수 있다.

$$\Delta \dot{S} + 2\zeta\omega_n \Delta \dot{S} + \omega_n^2 \Delta S = -kd_1 \bar{\Theta}_T \quad (3.1.89)$$

위 식은 이차시스템을 나타내며, $kd_1 \bar{\Theta}_T$ 항만이 가진력으로 작용한다. (증명끝)

수치 시뮬레이션

제안한 유도법칙의 성능을 평가하기 위해서 수치 시뮬레이션을 수행하였으며, 성능비교를 위해 다음과 같은 증강비례항법을 이용한 시뮬레이션도 수행하였다.

$$a_c = \frac{3}{\cos(\Theta_M - \beta)} \dot{\beta} + \ddot{\Theta}_T \cos(\Theta_T - \beta) \quad (3.1.90)$$

시뮬레이션 시나리오는 표 3.1.2에 정리하였으며 미사일과 표적의 초기위치는 다음과 같다.

표 3.1.2 시뮬레이션 시나리오

시나리오	유도법칙	k	시정수	잡음레벨
1	유도법칙 I	0	5	mid-level
2		0.5	5	mid-level
3		1	3	low-level
4		1	5	mid-level
5		1	5	high-level
6	유도법칙 II	0	5	mid-level
7		0.5	5	mid-level
8		1	3	low-level
9		1	5	mid-level
10		1	5	high-level
11	APN	×	3	low-level
12		×	5	mid-level
13		×	5	high-level

$$X_M(0) = 0(m), Y_M(0) = 0(m)$$

$$X_T(0) = 8660.3(m), Y_T(0) = 5000.0(m)$$

미사일과 표적의 속도는 다음과 같이 가정하였다.

$$V_M = 500(m/sec) \quad (3.1.91)$$

$$V_T = \begin{cases} 200(m/sec), & (t < 2) \\ 200 + 50(t-2)(m/sec), & (2 \leq t \leq 6) \\ 400(m/sec), & (t \geq 6) \end{cases} \quad (3.1.92)$$

표적은 다음과 같은 기동을 하는 것으로 설정하였다.

$$a_p = -3 + 7 \sin(t) \quad (3.1.93)$$

미사일의 속도로 정규화된 가속도에 대해서 최대 가속도 크기를 다음과 같이 설정하였다.

$$a_c = \begin{cases} -5, & a_c < -5 \\ a_c, & -5 \leq a_c \leq 5 \\ 5, & a_c > 5 \end{cases} \quad (3.1.94)$$

미사일의 입장에서는, 변수 $V_M, V_T, R, \dot{R}, \beta, \dot{\beta}, \theta_T, \dot{\theta}_T, \ddot{\theta}_T$ 에 대한 정보를 얻기 위해서 다음과 같은 추정동역학이 필요하다고 가정하였다.

$$H(s) = \frac{\omega_{e,n}^2}{s^2 + 2\zeta_e \omega_{e,n} s + \omega_n^2} \quad (3.1.95)$$

식 (3.1.95)에서 $\omega_{e,n}=7, \zeta_e=0.7$ 로 설정하였다. 변수 $\beta, \dot{\beta}, \theta_T, \ddot{\theta}_T$ 의 추정값에는 대역이 제한된 가우시안 잡음을 추가하였다. 잡음의 평균 절대값 크기는 표 3.4.2와 같이 세 가지 경우를 고려하였다. 또한 연료 소비량 J_F 와 에너지 소비량 J_E 도 계산하였다.

$$J_F = \int_0^T |a_M| dt \quad (3.1.96)$$

$$J_E = \int_0^T a_M^2 dt \quad (3.1.97)$$

표 3.1.4에 시뮬레이션 결과를 요약하여 정리하였다. 시나리오 1-5는 제안한 유도법칙 I, 시나리오 6-10은 제안한 유도법칙 II, 시나리오 11-13은 증강비례항법을 이용한 시뮬레이션 결과이다. 제안한 유도법칙에서 $k=0$ 에 해당하는 시나리오 1, 6과 $k=0.5$ 에 해당하는 시나리오 2, 7에서는 요격이 실패하였고, $k=1$ 에 해당하는 시나리오 3-5, 8-10에서는 요격이 이루어졌다. $k < 1$ 인 경우, 미사일과 표적의 거리가 가까워질수록 미사일이 정확히 충돌 코스에 있지 않으므로, 요구되는 가속도의 크기가 커지는 경향이 있다. 그러나, 미사일에서 성취될 수 있는 가속도의 크기는 제한되므로 빗맞힘거리가 발생한다.

표 3.1.3 평균 잡음 크기

	low-level	mid-level	high-level
β	0.0906	1.9289	2.4745
$\dot{\beta}$	0.2593	2.1813	2.5928
θ_T	0.0864	1.9241	2.3673
$\ddot{\theta}_T$	0.2706	2.6440	3.3017

표 3.1.4 시뮬레이션 결과

시나리오	요격시간(sec)	빗맞힘거리(m)	연료소비량	에너지소비량
1	×	781.9	×	×
2	×	590.4	×	×
3	33.0	2.1	2.414	0.201
4	32.5	5.9	2.591	0.219
5	32.5	7.1	2.622	0.222
6	×	1038.3	×	×
7	×	626.2	×	×
8	33.0	0.9	2.540	0.212
9	32.5	6.9	2.701	0.231
10	32.5	8.4	2.715	0.233
11	33.2	0.8	2.433	0.202
12	35.0	62.6	2.945	0.253
13	35.3	374.2	2.952	0.253

시나리오 3-5, 8-10, 11-13에서는 각 유도법칙의 잡음의 크기 및 자동조종장치에 의한 영향을 살펴 볼 수 있다. 유도법칙 I과 유도법칙 II의 경우, 잡음 크기가 커질수록 빗맞힘거리(miss distance) 및 연료소비량이 증가하지만 그 증가 정도는 크지 않다. 이에 비해서 증강비례항법의 경우, 잡음크기가 작을 때는 느린 자동조종장치인 경우에도 뛰어난 성능을 발휘하였지만, 잡음이 증가할수록 급격히 성능이 저하되고 있다. 잡음 크기가 어느 정도를 넘으면 빗맞힘거리가 급격히 증가하며, 연료 및 에너지도 제안한 유도법칙에 비해 증가속도가 큼을 알 수 있다.

제안한 유도법칙들을 비교해 보면, 자동조종장치의 동역학이 느린 경우인 시나리오 3, 8에서는 자동조종장치의 동역학을 고려한 유도법칙 II의 빗맞힘거리가 유도법칙 I보다 더 작음을 알 수 있다. 따라서 자동조종장치의 동역학이 느릴 경우, 이를 보상해주는 유도법칙 II의 성능이 자동조종장치의 동역학을 고려하지 않는 유도법칙 I보다 더 나을 것임을 추론할 수 있는 사실이다. 그림 3.1.44에는 값에 따른 유도명령 I 및 증강비례항법에 의한 미사일과 표적의 궤적을 제시하였다. k 가 0에서 1로 커짐에 따라 종말유도에서 기동이 작아짐을 알 수 있다. 증강비례항법의 경우, 종말유도가 가까워질수록 k 값이 큰 유도법칙의 궤적 방향으로 급격한 기동을 하고 있다.

그림 3.1.45는 유도법칙 I을 사용하였을 경우 정규화 된 가속도 명령의 변화를 보여주고 있다. $k=0$ 일 때는 기동 초기에서 중기 정도까지 가속도 명령 방향이 바뀔 수 있으며, $k=0.5$ 일 때는 기동 중기에 이러한 경향이 나타난다. 두 가지 경우 모두 기동말기에서는 최대가속도로 회전하게 되나 빗맞힘거리가 발생한다. 그러나 $k=1$ 일 때는 기동 중기까지 최대 가속도로 회전하다가 기동 말기에는 진동함을 알 수 있다.

그림 3.1.46에는 유도법칙 II를 사용하였을 경우와 증강비례항법의 경우를 같이 도시하였다. 잡음의 크기가 작을 때 유도법칙 II는 기동 말기에서 0 근처의 값을 가짐을 알 수 있다. 증강비례항법의 경우, 기동 초기에서 중기에 걸쳐 간헐적으로 계속 유도명령의 방향이 바뀌어 이로 인해 기동말기에 최대 가속도 명령이 주어지지만 약간의 빗맞힘거리가 발생하게 된다.

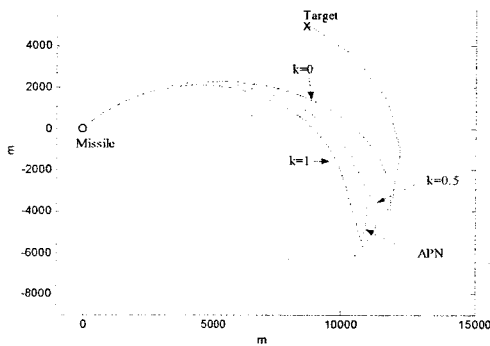


그림 3.1.44 미사일과 표적의 궤적
(시나리오 1, 2, 3, 11)

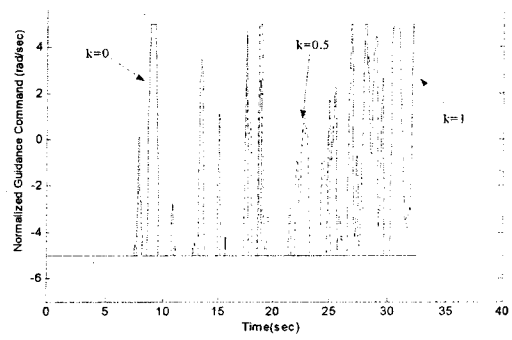


그림 3.1.45 가속도 명령
(시나리오 1, 2, 4)

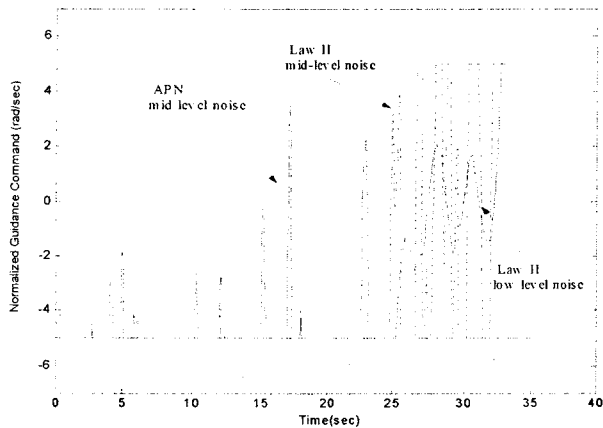


그림 3.1.46 가속도 명령 (시나리오 7, 8, 12)

결론

본 연구에서는 매개변수의 값에 따라 목표각을 정의하고, 미사일의 속도벡터 방향이 목표각과 일치하도록 하는 유도법칙을 제안하였다. 르야프노프 이론을 적용하여 제안한 유도법칙이 항상 목표각 오차를 점근적으로 0으로 만들 수 있음을 보였으며, 자동조종장치의 동역학을 무시할 경우에 오차의 반응이 1차 시스템의 반응과 같음을 증명하였다. 자동조종장치에 대한 1차 동역학을 고려하였을 때는 목표각 오차의 반응을 2차 시스템으로 모사할 수 있는 유도법칙을 제안하였다. 제안된 유도법칙은 매개변수의 설정에 따라 다양한 유도 목적을 구현할 수 있다. 제안한 유도법칙에 대하여 다양한 잡음 환경하에 수치 시뮬레이션을 수행하였으며, 시뮬레이션 결과 제안한 유도법칙의 효용성을 확인할 수 있었다. 특히, 기존의 증강비례항법보다 강건성이 뛰어나음을 알 수 있다.

나. 지능형 제어기를 이용한 항공기의 회피 유도법칙

항공기는 미사일의 공격에 대해 적절한 회피를 해야 하므로, 비례항법유도(PNG) 미사일에 대한 항공기의 최적 회피기동과 반비례항법 유도에 대한 연구가 활발히 수행되어 왔다.

최적화 방법은 계산시간이 많이 걸리기 때문에 실시간 유도법칙으로 사용할 수 없다는 단점이 있고, 반비례항법 유도법칙은 최적궤적과는 다른 경향을 갖는다는 문제점이 있다. 특히, 항공기는 비선형 특성과 불확실성에 민감하기 때문에, 기동이 심한 비행에는 선형 제어기가 적합하지 않다. 이러한 문제점을 보완하기 위해 본 연구에서는 지능형 제어기를 이용한 근최적 유도법칙을 제안하였다.

본 장에서는 교전 순간 미사일과 항공기의 상태변수와 제어입력 사이의 관계를 구하기 위해 다양한 시나리오를 고려하였다. 또한, 실시간 응용이 가능하고, 외란이나 불확실성에 민감하지 않은 제어기를 신경회로망과 퍼지논리를 이용하여 설계하였다. 2차원 수직 평면에 대한 수치 시뮬레이션을 통하여 지능형 제어기의 성능을 검증하였다.

(1) 운동방정식

2차원 공간에서의 추적자와 회피자의 기하학적 형상은 그림 3.1.47과 같다. x축은 거리를 z축은 고도를 나타내고, 하첨자 e는 회피자인 항공기를, p는 추적자인 미사일을 나타낸다.

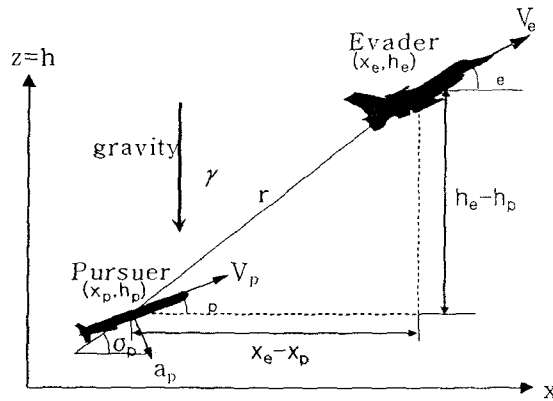


그림 3.1.47 추적자와 회피자의 형상

추적자의 운동방정식은 다음과 같이 표현할 수 있다.

$$\dot{V}_p = \frac{T_p - D_p}{m_p} - g \sin \gamma_p \quad (3.1.98)$$

$$\dot{x}_p = V_p \cos \gamma_p \quad (3.1.99)$$

$$\dot{h}_p = V_p \sin \gamma_p \quad (3.1.100)$$

$$\dot{\gamma}_p = \begin{cases} 0 & : 0 \leq t \leq t_1 \\ -\frac{a_p + g \cos \gamma_p}{V_p} & : t > t_1 \end{cases} \quad (3.1.101)$$

$$\dot{a}_p = \begin{cases} 0 & : 0 \leq t \leq t_s \\ \frac{a_{pc} - a_p}{\tau_p} & : t > t_s \end{cases} \quad (3.1.102)$$

$$D_p = 0.0001 V_p^2 + \frac{\dot{a}_p^2}{V_p^2} \quad (3.1.103)$$

$$\dot{m}_p = \begin{cases} 165 \text{ kg} & : t=0 \\ 150 \text{ kg} & : t \geq t_b \end{cases} \quad (3.1.104)$$

여기서 a 는 피치 가속도 명령을 나타내고, 유도명령에는 1차의 시간지연 $\tau_p=0.1\text{sec}$ 이 있다고 가정하였다. 일반적으로 지대공 미사일의 비행은 4단계로 나눌 수 있다. 본 연구에서는 미사일이 발사대를 떠나는 시간 t_b , 발사대를 떠나서 유도가 될 때까지의 시간 t_s , 유도를 시작하면서 연료를 다 소모할 때까지의 시간 t_b , 연료를 다 소모한 이후부터 끝까지 비행하는 시간을 고려하였다.

회피자의 운동방정식은 다음과 같이 표현된다.

$$\dot{V}_e = \frac{T_e \cos \alpha_c - D_e}{m_e} - g \sin \gamma_e \quad (3.1.105)$$

$$\dot{x}_e = V_e \cos \gamma_e \quad (3.1.106)$$

$$\dot{h}_e = V_e \sin \gamma_e \quad (3.1.107)$$

$$\dot{\gamma}_e = \frac{(L_e + T_e \sin \alpha_c)}{m_e V_e} - \frac{g \cos \gamma_e}{V_e} \quad (3.1.108)$$

양력과 항력에 관계된 식과 계수들을 정리하면 다음과 같다.

$$L_e = \frac{1}{2} C_L \rho v_e^2 S_e, \quad C_L = C_{L_0} \alpha_e, \quad C_{L_0} = 4.01$$

$$S_e = 26 \text{ m}^2, \quad D_e = \frac{1}{2} C_D \rho v_e^2 S_e, \quad C_D = C_{D_0} + k C_L^2$$

$$C_{D_0} = 0.0169, \quad k = 0.179, \quad T_e = 65000 \text{ N}, \quad m_e = 7500 \text{ kg}, \quad |\alpha_e| \leq \alpha_{\max} = \alpha_{\text{stall}}$$

여기서 항공기는 최대추력으로 회피한다고 가정하였고, 최대 하중계수는 다음과 같다.

$$\left| \frac{L_e}{m_e g} \right| \leq n_{\max} \quad (3.1.109)$$

(2) 비례항법 추적 유도법칙

비례항법 유도법칙은 진비례항법 방식과 순비례항법 방식으로 구분할 수 있다. 본 연구에서는 미사일의 속도 방향에 수직으로 가속도 명령을 주는 순비례항법을 사용했고, 미사일에 수동형 탐색기가 장착되어 있다고 가정하였다. 유도명령은 다음과 같다.

$$a_c^{\text{PN}} = \begin{cases} N\sigma V + g \cos \gamma & : |a_c| \leq a_{c \max} \\ a_{c \max} \text{ sign}(a_c) & : |a_c| > a_{c \max} \end{cases} \quad (3.1.110)$$

여기서 N 은 항법상수를, σ 는 $90^\circ - \sigma_p$ 로 계산된 시선각을 나타낸다. 식 (3.1.110)에서 시선각 변화율은 다음 식으로 표현된다.

$$\ddot{\sigma} = \frac{\Delta h \Delta \dot{x} - \Delta \dot{h} \Delta x}{r^2} \quad (3.1.111)$$

(3) 반비례항법 회피 유도법칙

반비례항법 유도법칙은 PNG 미사일에 대한 적절한 회피방법으로 알려져 있다. 유도법칙은 다음과 같이 표현된다.

$$a_e^{-PN} = -N_e \ddot{\sigma} V_e \quad (3.1.112)$$

$$a_e = \begin{cases} \frac{m_e(a_e^{-PN} + g \cos \gamma_e)}{0.5\rho V_e^2 S_e C_{L_e} + T_e} & : |a_e| \leq a_{e_{\max}} \\ a_{e_{\max}} \text{sign}(a_e^{-PN}) & : |a_e| > a_{e_{\max}} \end{cases} \quad (3.1.113)$$

(4) 최적회피 회피 유도법칙

최적제어를 수행하기 위해서는 적절한 성능지수를 선정해야 한다. 미사일의 공격을 피하기 위해서 항공기는 빗맞힘 거리를 최대화하는 기동을 해야 하므로 성능지수는 식 (3.1.114)와 같이 정할 수 있고, 빗맞힘 거리의 개념에 의해서 식 (3.1.115)이 유도된다

$$\max J = \max r^2(t_f) \quad (3.1.114)$$

$$\frac{1}{2} \frac{d}{dt} r^2(t_f) = (\Delta x \Delta \dot{x} - \Delta \dot{h} \Delta h) \Big|_{t=t_f} = 0 \quad (3.1.115)$$

(5) 신경회로망 회피 유도법칙

신경회로망은 입력, 시냅스 가중치, 출력, 임계치, 활성화 함수로 구성된다. 입출력 관계를 수식으로 나타내면 다음과 같다.

$$a_j = f\left(\sum_{i=1}^R p_i w_{ij} + b_j\right) \quad (3.1.116)$$

여기서 a_j , p_i , w_{ij} , b_j 는 각각 j 번째 출력, i 번째 입력, i 번째 가중치, j 번째 임계치를 나타낸다.

뉴런들을 학습시키기 위해 최적회피 유도법칙에서 계산된 최적궤적을 사용하였다. 항공기의 속도와 시선각 변화율에 따른 받음각을 역전과 방법을 사용하여 훈련시켰다. 이때, 7개의 뉴런으로 구성된 1개의 은닉층과 1개의 뉴런으로 구성된 1개의 출력층을 구성하였고, 모든 활성화 함수는 탄젠트 시그모이드 함수를 이용하였다. 가장 성능이 우수하고 학습 속도가 빠르다고 알려진 Levenberg-Marquardt 역전과 알고리즘을 이용하였다. 시뮬레이션에서는 목표와의 오차가 0.000001 이하, 최대 훈련 epoch는 200으로 설정하였다.

(6) 퍼지논리 회피 유도법칙

전문가 시스템으로 잘 알려진 퍼지논리는 복잡한 비선형 함수도 모델링할 수 있다. 또한, 전통적인 제어 및 유도방법에 쉽게 응용될 수 있고, 인간의 의사소통에 기반을 두고 있어서 비선형성과 불확실성을 다루는 데 효과적이다.

본 연구에서는 퍼지논리에 3개의 언어변수 NL, ZE, PL을 사용하였고, 추론방법으로는 Mamdani 방법을, 비퍼지화 방법은 무게중심법을 사용하였다. 제어입력인 받음각을 결정하는 가장 중요한 상태변수인 항공기의 속도와 시선각 변화율을 퍼지 입력변수로 선정하였다. 따라서 퍼지 출력변수는 받음각이 된다. 그림 3.1.48에 퍼지 소속함수를 나타내었다. 모든 소속함수는 최적화 결과를 이용하여 구성하였다.

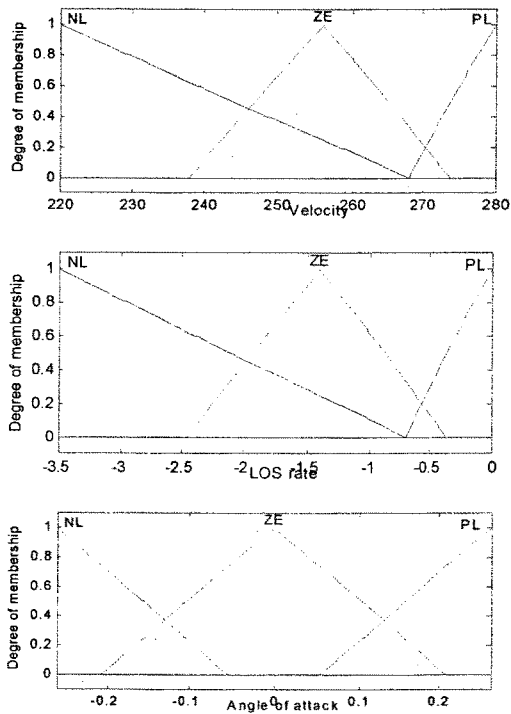


그림 3.1.48 퍼지 소속함수

최적회피 유도법칙의 결과를 바탕으로 퍼지 규칙 베이스를 만들면 표 3.4.4과 같다.

표 3.1.5 퍼지 규칙 베이스

받음각		시선각 변화율		
		NL	ZE	PL
속도	NL	NL	NL	ZE
	ZE	NL	NL	ZE
	PL	NL	NL	PL

이러한 퍼지 소속함수와 규칙 베이스를 이용하여 퍼지 곡면을 나타내면 그림 3.1.49과 같다.

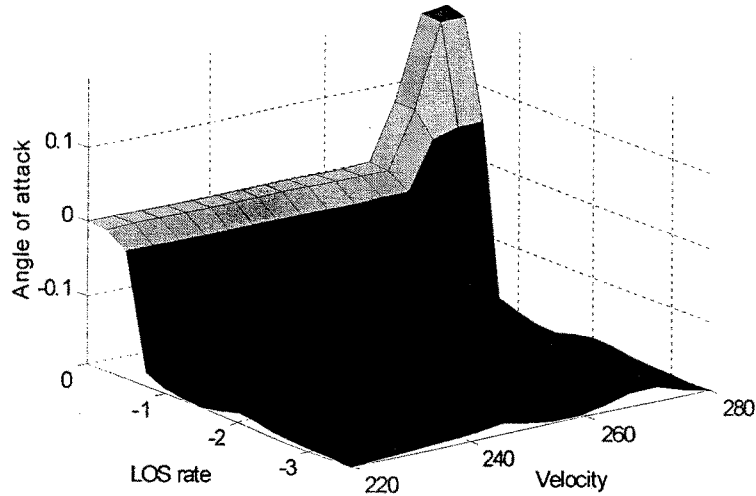


그림 3.1.49 퍼지 곡면

(7) 수치 시뮬레이션

지능형 제어기의 성능을 검증하기 위하여 수치 시뮬레이션을 수행하였고, 항공기 모델은 F-16 데이터를 사용하였다. 미사일의 기지는 원점에 위치해 있다고 가정하였다. 항공기의 초기 위치, 속도, 비행경로각은 표 3.1.6와 같다.

표 3.1.6 회피자의 초기값

	지면거리	고도	속도	비행경로각
교전1	1500m	1000m	280m/s	0°(접근)
교전2	1500m	1000m	350m/s	0°(접근)
교전3	2500m	2000m	280m/s	0°(접근)
교전4	0m	2000m	280m/s	0°(접근)
교전5	1500m	1000m	280m/s	15°(접근)
교전6	1500m	1000m	280m/s	0°(회피)
교전7	1500m	1000m	280m/s	15°(회피)

그림 3.1.50-51에 교전 1과 교전 7에 대한 결과를 나타냈다. 전체적으로 신경회로망과 퍼지논리를 사용해 구성한 유도법칙이 최적제어 유도법칙과 유사한 궤적을 나타냄을 알 수 있다.

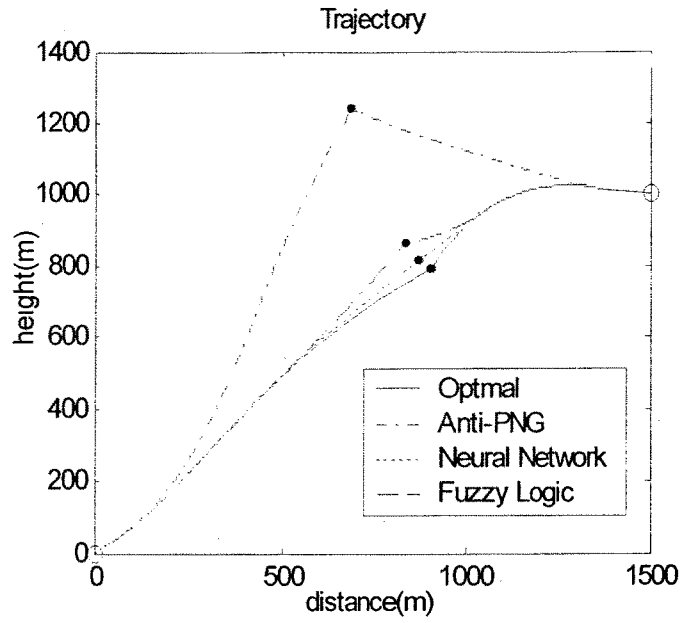


그림 3.1.50 교전 1의 비행 궤적

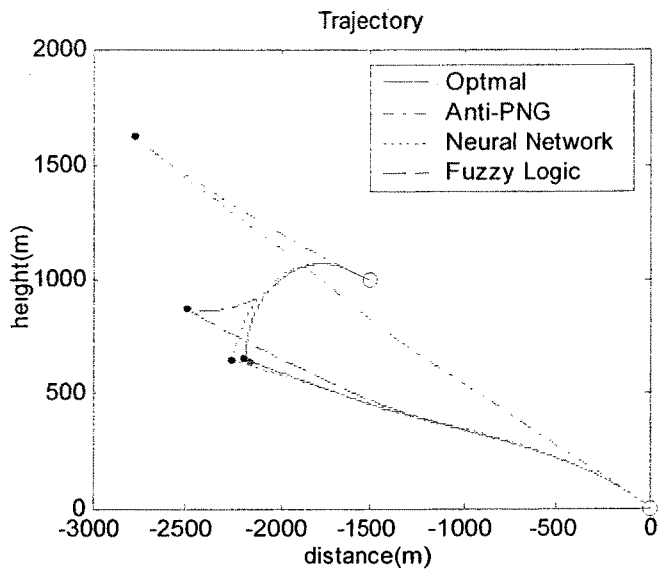


그림 3.1.51 교전 7의 비행 궤적

표 3.1.7에 전체 교전상황에 대한 시뮬레이션 결과를 나타내었다.

표 3.1.7 빗맞힘 거리와 최종 시간

	반비례항법	최적제어	신경회로망	퍼지논리
교전1	3.1810m	7.6209m	2.4978m	2.6876m
	3.0006s	2.7585s	2.7321s	2.7434s
교전2	1.5786m	8.0837m	6.2725m	2.7417m
	3.0831s	2.6551s	2.7167s	2.6152s
교전3	0.5737m	7.5321m	3.7283m	4.1333m
	4.3303s	3.9035s	3.9285s	3.8353s
교전4	0.0866m	0.9051m	6.1217m	2.1163m
	4.2646s	4.5296s	3.3967s	4.3257s
교전5	1.8854m	18.9182m	6.5487m	1.9710m
	3.2464s	2.9197s	2.9108s	2.9186s
교전6	0.0434m	3.8277m	2.2937m	1.2133m
	4.5869s	3.5398s	3.6006s	3.9613s
교전7	0.0635m	27.8200m	2.7435m	0.6270m
	4.8344s	3.8398s	3.9458s	4.2313s

위의 결과를 보면 전체적으로 신경회로망이나 퍼지논리를 이용하여 유도법칙을 설계한 것이 반비례항법 유도법칙보다 우수한 성능을 보임을 알 수 있다. 특히, 교전 4의 경우 지능형 제어기를 사용한 것이 최적회피 유도법칙보다 더 큰 빗맞힘 거리를 갖는다. 이 경우 최적해가 국소점으로 수렴하여 위와 같은 결과를 초래한 것이라 해석되어진다.

(7) 결론

본 연구에서는 비례항법을 이용하는 지대공 미사일의 공격에 대한 항공기의 다양한 회피 유도법칙들을 살펴보았다. 지능형 제어기를 사용해 실시간 근최적 유도법칙을 설계하였다. 신경회로망의 LMBP 알고리즘을 이용하여 최적회피 기동의 결과를 신경회로망 유도법칙에 학습시켰다. 그리고 최적해를 이용하여 퍼지 규칙과 소속함수를 구성하였다. 여러 가지 교전 상황에 따른 시뮬레이션의 결과를 통하여 제안한 지능형 제어기의 성능이 기존의 제어기보다 성능이 우수함을 검증할 수 있었다.

다. 경로점을 통과하는 최적 비행궤적 설계

비행경로 설계문제는 비행체 유도항법제어의 중요한 문제 중 하나로, 무인항공기의 기준 궤적 설정문제나 미사일 경로 설계문제 등 적용분야가 다양하다. 비행경로 생성은 일반적으로 비행체가 통과해야 할 주요 지점을 비행경로점(waypoint)으로 선정함으로써 해결한다. 그러나 비행경로를 생성할 때 비행체의 기동성, 순항거리, 최대 회전반경 등 비행체의 동특성과 지정된 경로점과의 관계를 고려하여 비행궤적을 생성하여야 한다. 주어진 비행성능을 최대한 이용하여 생성된 최적 비행경로는 비행체에 주어진 임무를 원활하게 달성하게 할 뿐만 아니라, 임무반경을 증대시키는 효과가 있다. 최근에는 미사일이나 항공기가 넓은 비행영역에서 고기동성을 가지고 비행할 수 있도록 개발되고 있다. 근접한 경로점들을 통과하여 최소시간에 목표점에 도달하도록 하는 문제도 이러한 경우에 해당하는데, 이때 구

동기 포화문제가 발생할 수 있다. 이러한 문제들은 근접한 경로점을 지나면서 소요시간이나 연료소모 등의 성능지수에 대해 최적경로를 찾는 문제의 해석을 복잡하게 한다. 본 논문에서는 비행체의 동역학을 고려하며, 주어진 경로점을 통과하도록 비행경로를 생성하는 최적 알고리즘을 제안하였다. 비행경로점을 통과하기 위해 제어변수와 경로점 통과시간을 매개변수로 설정하였으며 이에 대한 적절한 매개화 과정을 제안하였다. 수치시뮬레이션을 통해 그 타당성을 검증하였다.

(1) 경로점을 통과하는 최적 비행경로

본 논문에서 고려하는 비행체를 질점계로 가정하여 문제를 정식화하기 위해 그림 3.1.52 같은 지구중심 좌표계를 선정하였다.

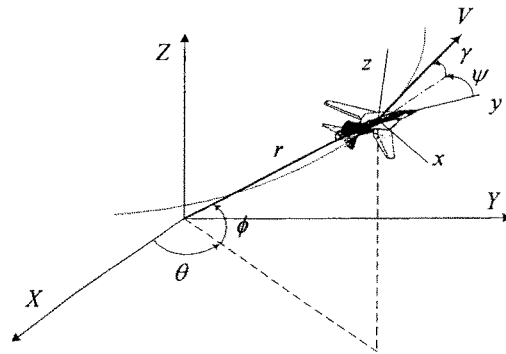


그림 3.1.52 3차원 좌표계

여기서 $r, \theta, \phi, V, \gamma, \psi$ 는 각각 비행체의 거리, 경도, 위도, 비행체의 속도, 종방향 비행경로각, 횡방향 비행경로각을 나타낸다. 이때, 운동방정식은 다음과 같다.

$$\begin{aligned} \dot{r} &= V \sin \gamma \\ \dot{\theta} &= \frac{V \cos \gamma \cos \psi}{r \cos \phi} \\ \dot{\phi} &= \frac{V \cos \gamma \sin \psi}{r} \end{aligned} \quad (3.1.117)$$

$$\ddot{V} = \frac{1}{m} F_T - g \sin \gamma + \omega^2 r \cos \phi (\sin \gamma \cos \phi - \cos \gamma \sin \psi \sin \phi)$$

$$\ddot{\gamma} = \frac{1}{m} F_N \cos \sigma - g \cos \gamma + \frac{V^2}{r} \cos \gamma + 2\omega V \cos \psi \cos \phi + \omega^2 r \cos \phi (\cos \gamma \cos \phi + \sin \gamma \sin \psi \sin \phi) \quad (3.1.118)$$

$$\ddot{\psi} = \frac{1}{m} \frac{F_N \sin \sigma}{\cos \gamma} - \frac{V^2}{r} \cos \gamma \cos \psi \tan \phi + 2\omega V (\tan \gamma \sin \psi \cos \phi - \sin \phi) - \frac{\omega^2 r}{\cos \gamma} \cos \psi \sin \phi \cos \phi$$

$$\begin{aligned} \dot{x} &= V \cos \gamma \cos \psi \\ \dot{y} &= V \cos \gamma \sin \psi \\ \dot{z} &= V \sin \gamma \end{aligned} \quad (3.1.119)$$

여기서 접선, 법선 방향의 힘과 이를 구성하는 각종 공력계수는 다음과 같다.

$$F_T = T \cos \alpha - D, \quad F_N = T \sin \alpha + L$$

$$D = \frac{1}{2} \rho V^2 C_D S_{ref}, \quad L = \frac{1}{2} \rho V^2 C_L S_{ref}, \quad C_L = C_{L_0} \alpha, \quad C_D = C_{D_0} + k C_L^2$$

에너지를 최소화하는 성능지수를 고려하면 다음과 같다.

$$\text{minimize} : J = \int_{t_0}^{t_f} u^2 dt \quad (3.1.120)$$

비선형 운동방정식은 비행체의 운동특성을 표현하는 미분형 등식 구속조건으로 고려된다. 운동방정식에 의한 비선형 미분방정식 형태의 등식 구속조건은 매개변수를 입력으로 하여 적분을 수행함으로써 구현할 수 있다.

비행경로는 지정된 경로점을 통과해야 하지만, 사용자가 정한 경로점을 정확히 통과하는 비행경로 생성은 매우 어렵다. 생성된 비행궤적이 지정된 경로점에서 일정 영역의 안쪽을 통과하는 경우, 비행체가 경로점을 통과한 것으로 고려하는 기법을 사용할 수 있으며, 이는 부등식 구속조건으로 정식화 할 수 있다. 그림 3.1.53 경로점을 통과하는 비행경로의 경로점 영역 부등식 조건을 도시한 것이다. 경로점 영역의 반경을 사용자가 임의로 지정함으로써, 비행체의 기동력을 고려하며 지정된 경로점을 통과하는 비행경로를 생성할 수 있다.

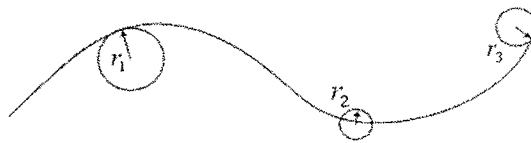


그림 3.1.53 로점 영역 부등식 구속조건

(2) 미지통과 시간을 고려한 최적화 알고리즘

본 연구에서 다루고 있는 문제는 종말점의 시간에 대한 구속조건이 없는 문제이므로, 각 경로 종말점의 시간 역시 매개변수로 선정하여야 한다. 시스템은 시간에 대한 미분방정식의 형태로 구성되어 있다. 시간을 매개변수로 설정하였으므로 시간에 영향을 받는 운동방정식을 수정하여야 한다. 경로점에 할당되어 있는 입력 매개변수를 종말시간 매개변수에 대한 종속관계로 만들어 주기 위해 다음과 같이 시간 정규화를 수행한다.

$$k \text{가 홀수인 경우} : \tau = \frac{t - \sum_{i=1}^{k-1} T_i}{T_{k-1}}, \quad (\sum_{i=1}^{k-1} T_i, \sum_{i=1}^k T_i) \rightarrow (0, 1) \quad (3.1.121)$$

$$k \text{가 짝수인 경우} : \tau = \frac{\sum_{i=1}^k T_i - t}{T_k}, \quad (\sum_{i=1}^{k-1} T_i, \sum_{i=1}^k T_i) \rightarrow (1, 0) \quad (3.1.122)$$

이를 도시하면 그림 3.1.54과 같다.

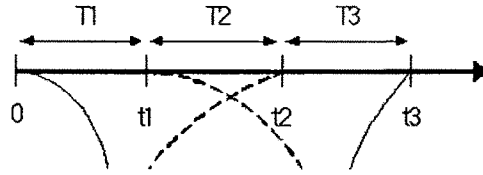


그림 3.1.54 종말시간 정규화 개념도

위의 식과 그림 3.1.54에서 보듯이 시간 매개변수를 정규화 시켜주는 경우, 각 구간의 종말점 상태변수 조건이 다음 구간의 초기조건으로 자동으로 설정되는 이점이 있다.

$$X_1(1) = X_2(1) \quad (3.1.123)$$

이와 같이 시간 매개변수를 정규화하게 되면 입력 매개변수는 시간 매개변수에 종속적 관계를 가져야 하는데, 이는 운동방정식을 이용하여 구현된다.

$$\frac{d\tau}{dt} = (-1)^{i+1} \frac{1}{T_i} \quad (i = 1, 2, \dots, n) \quad (3.1.124)$$

$$\frac{dx}{dt} = f(x, u) \Rightarrow \frac{dx}{d\tau} \frac{d\tau}{dt} = f(x, u, t) \quad (3.1.125)$$

위의 식을 정리하면 다음과 같이 (i)번째 구간의 비행체 운동방정식을 얻는다.

$$x_i = (-1)^i T_i f(x_i, u_i) \quad (3.1.126)$$

$$\dot{T}_i = 0$$

입력변수는 각 경로점의 종말 시간에 종속되어 설정되어야 한다. 이러한 개념을 도시하면 그림 3.1.55와 같다. 그림 3.1.55와 같이 각 경로점 말기시간에 대한 종속변수로 입력변수 매개화를 수행하면, 각 경로점 사이에는 항상 일정한 개수의 제어변수가 매개변수로 선정되게 된다. 결국, 최적화를 수행하기 위해 입력변수가 변할 때에도 물리적으로 연속적인 구간의 값으로 변하게 되어 수렴성이 증가할 것이다.

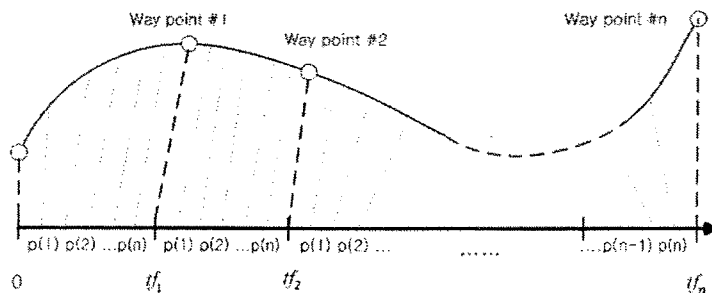


그림 3.1.55 시간 종속 매개변수 선정

(3) 수치 시뮬레이션

수치 시뮬레이션에 고려된 경로점은 표 1에 정리하였다. 비행경로가 경로점의 10m 반경에 접근하면 주어진 경로점을 통과한 것으로 설정하였다. 최적화를 위한 받음각과 롤각의 초기값은 그림 3.1.56와 같으며 이에 의해 생성된 초기 비행경로는 그림 3.1.57과 같다. 그림 3.1.58에는 최적입력을, 그림 3.1.59은 생성된 최적경로를 나타내고 있다. 피치면상에 주어진 경로점을 모두 잘 추종하고 있음을 살펴볼 수 있으며, 4, 5번 경로점 구간에서는 같은 고도임에도 최적경로를 생성하기 위해 고도 변화가 있음을 살펴볼 수 있다.

표 3.1.8 경로점 및 최적 결과 (단위:km)

경로점	x	y	z	초기오차	최적결과
1	1.5	0	0.12	83m	10m
2	2.5	0	0.25	663m	10m
3	5	0	0.34	779m	10m
4	8.5	0	0.35	360m	9.9m
5	12	0	0.35	347m	10m
6	14	0	0.6	180m	10m
7	15.5	0	0.36	349m	10m
8	16	0	0	874m	10m

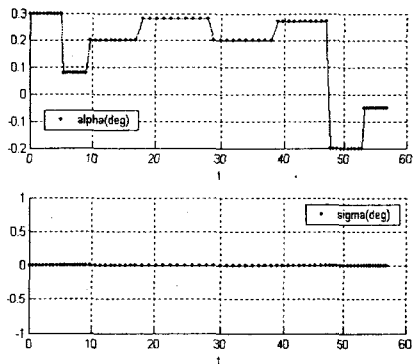


그림 3.1.56 입력 매개변수 초기화

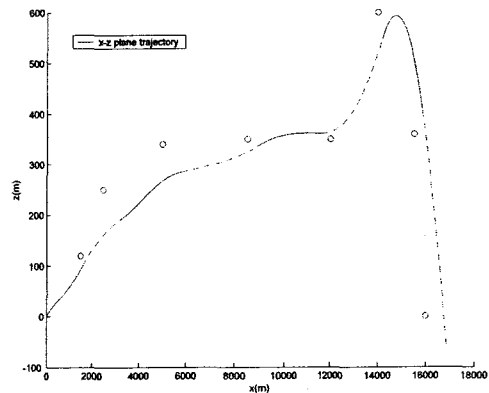


그림 3.1.57 초기값에 의한 비행궤적

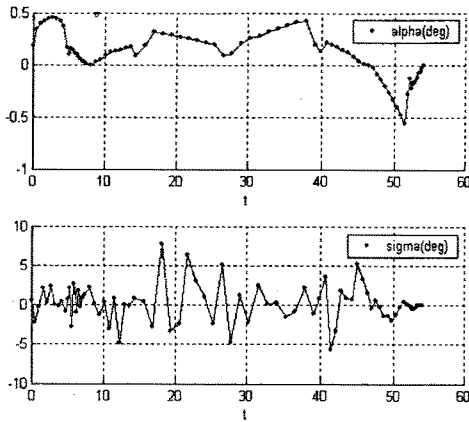


그림 3.1.58 최적 입력 매개변수

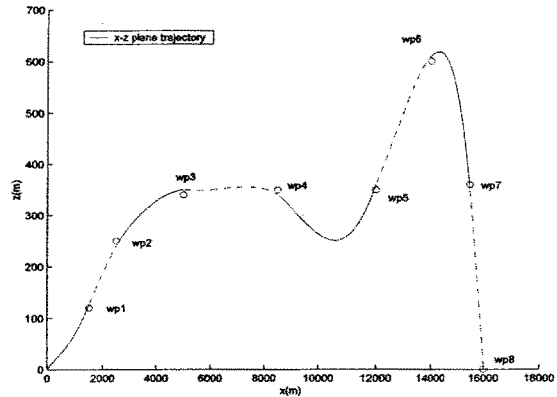


그림 3.1.59 생성된 최적 비행궤적

(4) 결론

주어진 경로점을 통과하는 최적 비행경로를 도출하기 위한 수치기법을 제안하였다. 경로점이 있는 비행경로 최적화의 경우 경로점 통과 시간이 미지수이며, 그 시점에서 경로점을 통과해야 하는 구속조건을 만족하도록 매개변수를 선정하였다. 입력 매개변수의 초기치가 수렴 결과에 많은 영향을 주게 되므로, 적절한 초기치 선정이 요구된다.

5. 고장 검출 및 인식 (FDI) 기법 연구

실제 항공기에서는 미션을 수행하는 과정에서 많은 고장 요인들이 존재하며 심각한 사고로 이어지기도 한다. 항공기의 신뢰성을 높이고, 부가가치를 높이기 위하여 근래 많은 FDI 기법들이 연구되고 있으며, 실제 항공기에서 적용되는 사례들도 많이 발견할 수 있다. 본 연구에서는 이러한 고장 검출 및 인식(Fault Detection and Isolation : FDI)에 관한 여러 기법들을 연구하고 시뮬레이션을 수행하였다.

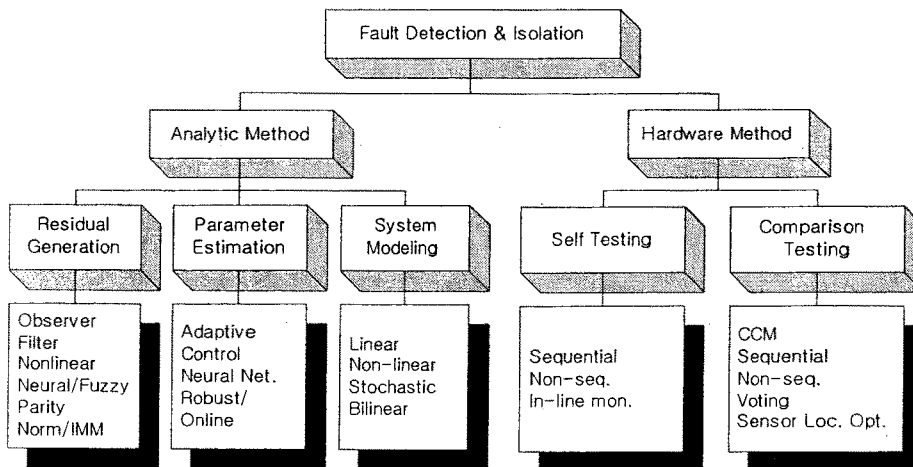


그림 3.1.60 FDI 기법의 분류

가. 소프트웨어 기법(모델기반 방식)

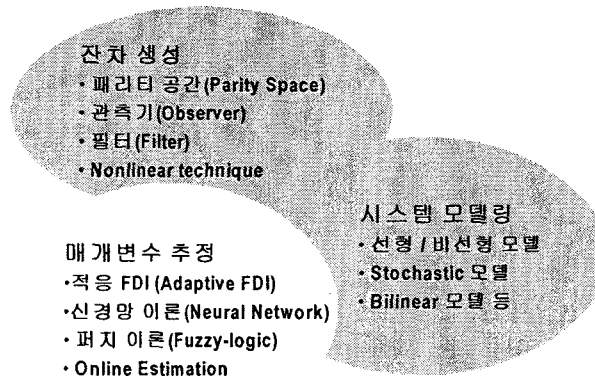


그림 3.1.61 소프트웨어 FDI 기법

소프트웨어 기법을 이용한 FDI 기법은 하드웨어 중복을 사용할 경우에 발생했던 비용/공간상의 문제점을 발생시키지 않으며, 서로 다른 역할을 맡고 있는 기기라 하더라도 시스템의 내부 동특성에 대한 정보를 이용하면 하드웨어 중복을 이용한 것과 같은 효과를 얻을 수 있다는 사실을 이용한다. 하드웨어를 이용한 기법들이 고장의 유무만을 판단할 수 있는 정성적(qualitative)인 방법인 것에 비하여, 소프트웨어 기법은 정량적(quantitative)으로 고장의 검출 및 분리를 수행한다는 장점을 가지고 있다.

소프트웨어 FDI 기법은 다음과 같은 구성 요소를 가지고 있다.

- actuator, sensor, plant model 의 fault
- 실제 시스템과 수학적 모델사이의 Modelling error
- system noise와 measurement noise

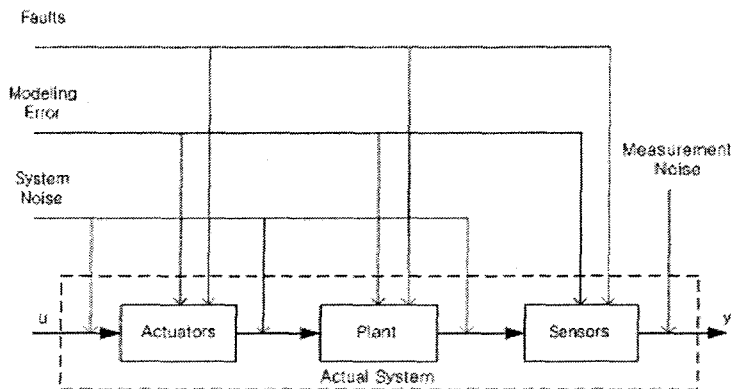


그림 3.1.62 소프트웨어 기법 시스템 블록선도

선형 시스템이라 가정할 경우

$$\dot{x}(t) = Ax(t) + Bu(t) + Ed(t) + Kf(t) \tag{3.1.127}$$

$$y(t) = Cx(t) + Fd(t) + Gf(t) \tag{3.1.128}$$

여기서 f 는 fault, d 는 알려지지 않은 입력(unknown input)을 의미하고, A, B, C 는 고장이 발생하기 전의 시스템 행렬들이다. E, F, K, G 는 시스템의 특성과 f, d 의 영향에 따라 적절히 선택되어질 수 있다.

(1) 잔차 생성 기법

잔차(residual)를 생성하여 이를 통해 고장 발생 여부를 판단하고 고장에 대한 정보, 예를 들면 고장 발생 시각과 위치, 크기 등의 정보를 추출한다. 잔차를 이용한 고장 검출 기법의 기본적인 개념은 다음의 그림에서 볼 수 있다.

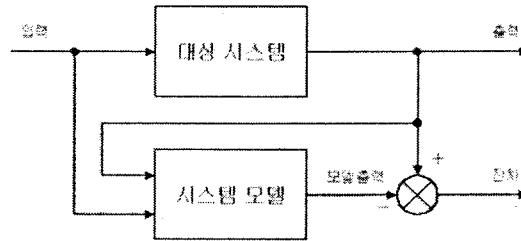


그림 3.1.63 잔차 생성 기법의 블록선도

(가) 관측기(Observer)기반 기법

관측기를 설치하여 이를 통해 나온 값과 원래 실제 시스템의 출력을 비교하여 잔차를 생성한다. 선형 시스템의 예를 보면 다음과 같다.

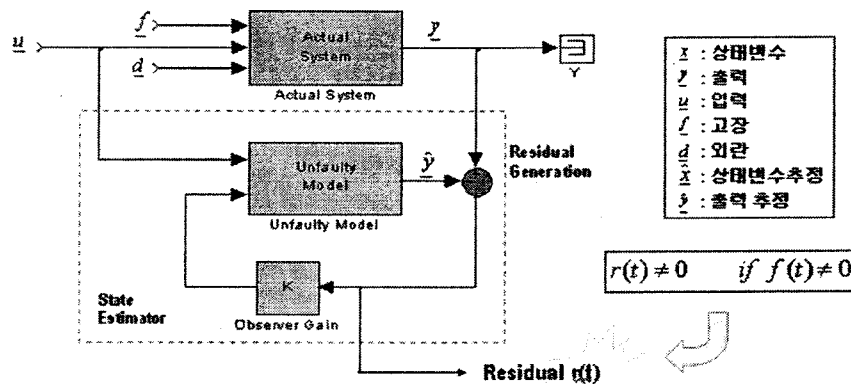


그림 3.1.64 관측기기반 기법의 블록선도

시스템 / 관측기 동역학은 다음과 같다.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Ed(t) \\ y(t) &= Cx(t) + Du(t) + f(t) \end{aligned} \tag{3.1.129}$$

$$\begin{aligned} \dot{\hat{x}} &= (A - KC)\hat{x}(t) + (B - KD)u(t) + Ky(t) \\ \hat{y}(t) &= C\hat{x}(t) + Du(t) \end{aligned} \tag{3.1.130}$$

시스템 에러는 다음과 같다.

$$\begin{aligned}
 e(t) &= x(t) - \hat{x}(t) \\
 \dot{e}(t) &= A_c e(t) + Ed(t) - Kf(t) \\
 (A_c &= A - KC \quad H = WC)
 \end{aligned} \tag{3.1.131}$$

잔차(Residual)은 다음과 같이 표현된다.

$$\begin{aligned}
 r(t) &= WCe(t) + Wf(t) = He(t) + Wf(t) \\
 r(s) &= [W - WC(SI - A_c)^{-1}]f(s) + WC(SI - A_c)^{-1}Ed(s)
 \end{aligned} \tag{3.1.132}$$

고장이 발생하지 않았을 경우 r 은 unknown input d 에만 영향을 받는다. 하지만 고장이 발생할 경우 r 은 고장 f 의 영향을 받아 값이 커지게 된다. Inverted Pendulum에 대하여 관측기 기반 기법을 간단하게 적용해 볼 수 있다.

Linearized system variables은 $[x \ \phi \ \dot{x} \ \dot{\phi}]$ 이며, Linearized system matrix는 다음과 같다.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1.93 & -1.99 & 0.009 \\ 0 & 36.9 & 6.26 & -0.174 \end{bmatrix} \tag{3.1.133}$$

$$B = [0 \ 0 \ -0.3205 \ -1.009]^T$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D = 0_{3 \times 1}, \text{ state feedback}$$

$$\text{Observer Poles} : \{-14, -20, -8 \pm 8i\}$$

Sensor 1, 2, 3 (x, ϕ, u)에 대하여 고장 유형을 설정하였으며, 시스템과 관측기 상태 방정식은 다음과 같다.

$$\begin{aligned}
 \dot{x}(t) &= Ax(t) + Bu(t) \\
 y(t) &= Cx(t) + Du(t) + f(t)
 \end{aligned} \tag{3.1.134}$$

$$\begin{aligned}
 \dot{\hat{x}} &= (A - LC)\hat{x} + Bu + Ly \\
 \hat{y} &= C\hat{x} + Du(t)
 \end{aligned} \tag{3.1.135}$$

output estimation error (residual)는 다음과 같이 표현될 수 있다.

$$e = y - \hat{y} = C\underline{\xi} + f \tag{3.1.136}$$

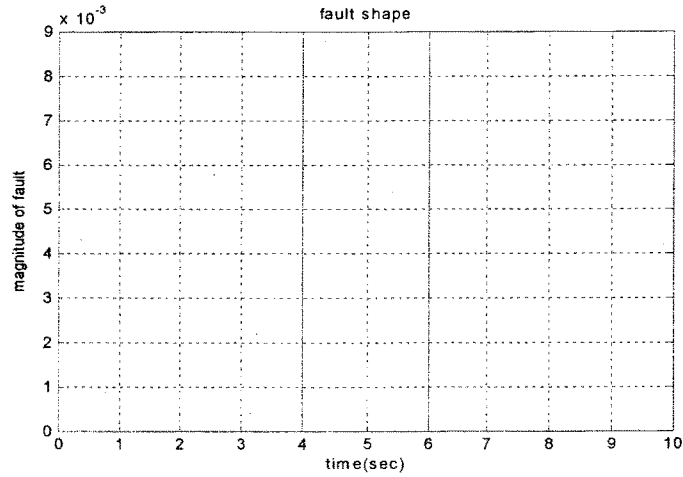


그림 3.1.65 고장 신호

위의 그림과 같은 고장 신호를 발생시킬 경우 다음과 같은 잔차 신호(Norm of Residual)를 통하여 고장을 진단할 수 있다.

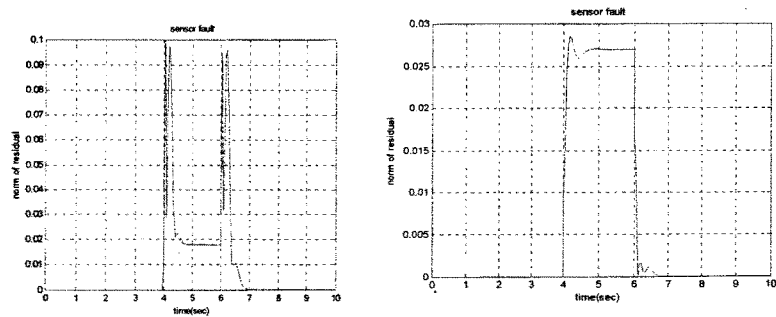
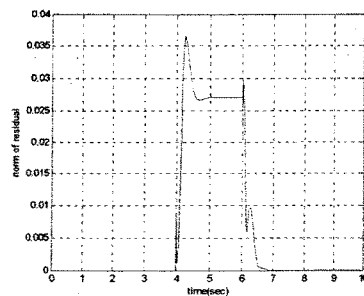


그림 3.1.66 센서 Fault Cases



제트 엔진 시스템에 대해서도 관측기 기법을 이용한 FDI 수치예제를 수행하였다.

$$A = \begin{bmatrix} -0.9813 & 7.5320 & -0.5983 & 0.4857 & -0.6979 \\ 0.2838 & -0.0826 & 0.0779 & -0.0617 & 0.0928 \\ -6.8588 & 28.9161 & -2.0561 & 1.6083 & -2.2612 \\ 1.2235 & -5.6607 & 0.4020 & -0.3192 & 0.4141 \\ 13.2662 & -53.4047 & 4.7390 & -3.7710 & 5.3669 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.00139 & 0.000195 \\ 0.000067 & -0.000005 \\ 0.003188 & 0.000601 \\ 0.007840 & -0.000273 \\ 0.003123 & -0.001516 \end{bmatrix}$$

$$C = I_{5 \times 5}$$

$$D = O_{5 \times 2}$$

(3.1.137)

다음과 같은 고장의 형상에 대하여 고장 검출할 수 있을 뿐만 아니라 외란과 고장을 분리하는데 매우 효과적이라는 것을 살펴볼 수 있다.

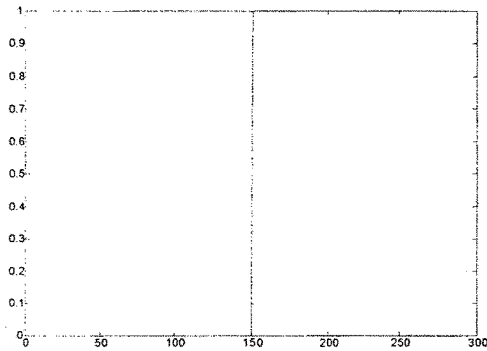


그림 3.1.67 고장의 형상

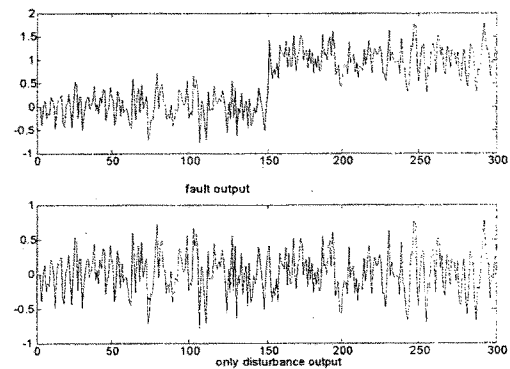


그림 3.1.68 외란시 고장에 따른 출력

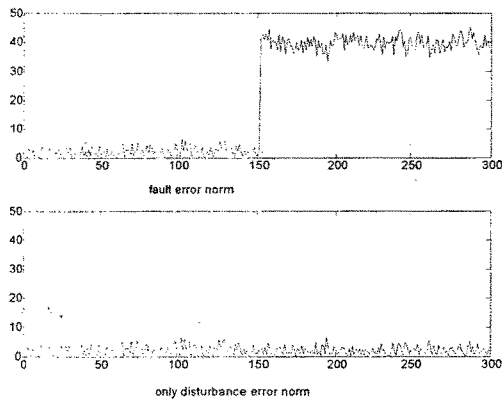


그림 3.1.69 고장유무에 따른 추정오차

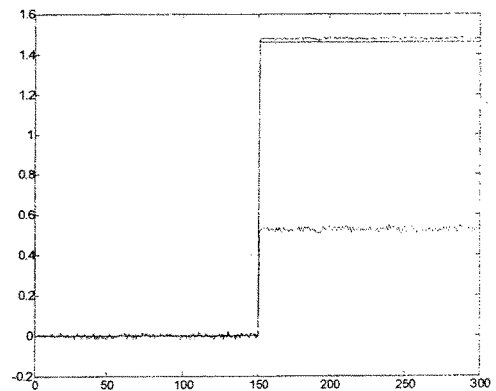
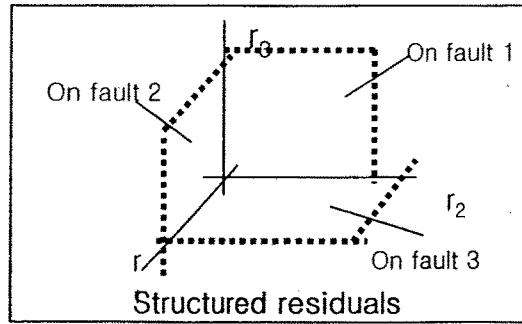


그림 3.1.70 고장이 있을 경우의 잔차

(나) 구조화된 잔차 생성 기법



	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4
r_1	1	1	0	0	1	1	0	0	1	1	0	0
r_2	1	1	1	0	1	0	1	0	1	0	1	0
r_3	1	1	0	1	1	1	0	0	0	1	0	1
r_4	0	0	1	1	0	0	1	1	0	0	1	1

고장의 미분리

고장의 약한 분리

고장의 강한 분리

$p_1, p_2, p_3, p_4 = \text{fault}, \quad r_1, r_2, r_3 = \text{residual}$

그림 3.1.71 구조화된 잔차 생성 기법

ARMA와 MA 형식의 구조화된 parity equation을 이용하여 구조화된 Residual 생성하는 기법을 연구하였다. 구조화된 유수를 생성하는 기법은 고장시 고장에 의해 span된 subset만 non-zero값을 가지는 원리를 이용하였으며, 고장 분석이 매우 용이하고 외란을 분리시키는 장점을 가지고 있다. 이 기법은 고장 code로 이루어지는 coding set을 생성하기 위한 한계값(Threshold) 실험을 필요로 한다.

- ARMA parity equations을 이용한 잔차 생성 기법

일반적인 discrete system의 State-space form의 기본 시스템 모델은 다음과 같다.

$$x(t+1) = Ax(t) + Bu(t) \quad (3.1.138)$$

$$y(t) = Cx(t) + Du(t)$$

이 모델을 input-output form의 형태로 바꾸면 다음과 같은 식이 된다.

$$y_m(t) = S(z)u(t) \quad (3.1.139)$$

기본 모델이 설정 되면, 원 모델에 외란이나 기계적인 고장 요소를 고려한 고장 vector $p(t)$ 를 포함시킨 고장 모델을 정의한다.

$$y(t) = S(z)u(t) + R(z)p(t) \quad (3.1.140)$$

여기서 고장이 포함된 모델과 원 모델의 차를 ARMA parity equation으로 정의를 하면 다음과 같이 고장에 대한 요소만이 남게 된다.

$$\alpha(t) = y - y_m = R(z)p(t) \quad (3.1.141)$$

이 식은 관찰 가능한 것(observables)과 유리함수 형태의 고장(faults) 등으로 표현된다. 이 식은 외란과 고장을 동시에 표현을 하고 있기 때문에 고장을 검출해내기 위한 잔차를 생성하기 위해서는 가중치를 적용하여 외란의 영향을 없애고 고장에 의한 영향만을 고려하도록 해야 한다. 따라서 다음과 같이 parity equation에 transforming vector $w(z)$ 를 넣어서 고장에 의한 잔차를 생성한다. $w(z)$ 는 외란의 영향을 없애줌으로써 구할 수 있다.

$$r(t) = w^T(z)\alpha(t) \quad (3.1.142)$$

- MA parity equations을 이용한 잔차 생성 기법

ARMA parity equation을 이용한 잔차 생성 기법에서 설정된 고장 모델을 다음과 같이 다시 정의한다.

$$y(t) = S(z)u(t) + R(z)p(t) \quad (3.1.143)$$

기본 모델의 state-space form에서 input-output form으로 전환을 할 때, 전환된 모델에서 $S(z)$ matrix의 각 요소는 전달 함수의 형태가 되며 다음과 같이 표현할 수 있다.

$$S(z) = \frac{G(z)}{H(z)} \quad (3.1.144)$$

이를 위의 고장 모델에 적용 시키면 다음과 같은 식이 된다.

$$\begin{aligned} H(z)y(t) &= G(z)u(t) + F(z)p(t) \\ F(z) &= H(z)R(z) \end{aligned} \quad (3.1.145)$$

이 식에서 입력(input)을 포함한 우변의 첫 번째 항을 좌변으로 보내면, 좌변에는 입출력 항을 포함한 식의 형태가 되고, 우변에는 외란과 고장을 포함한 요소를 고려한 fault vector $p(t)$ 에 의한 항만 남게 된다.

이러한 형태의 식을 MA parity equation으로 정의한다.

$$o^*(t) = H(z)y(t) - G(z)u(t) = F(z)p(t) \quad (3.1.146)$$

ARMA 기법과 마찬가지로 외란의 영향을 없애고 고장만을 검출해내기 위해 가중치를 적용하기 위해서 transforming vector를 고려해준다. 위와 같은 방법으로 이를 구할 수 있다. 따라서 잔차의 표현은 다음과 같은 형태가 된다.

$$r^*(t) = w^T(z)o^*(t) \quad (3.1.147)$$

Fault를 포함한 discrete model에 대한 수치예제를 수행하였다.

$$x(t+1) = \begin{bmatrix} 0.4 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.9 \end{bmatrix} x(t) + \begin{bmatrix} 0.5 \\ 1.0 \\ 0.1 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} p(t) \quad (3.1.148)$$

$$y(t) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} x(t)$$

여기서 fault vector는 다음과 같이 임의의 값을 주었다.

$$p(t) = \begin{bmatrix} \text{white noise}(0.001) \\ 3 \end{bmatrix} \quad (3.1.149)$$

$p_1(t)$ 는 외란으로 white noise를 주었고, $p_2(t)$ 는 plant fault의 값을 임의로 설정 하여 시뮬레이션을 수행하였다.

- ARMA parity equation을 이용한 Residual 생성 기법 시뮬레이션

① ARMA parity equation ($\alpha(t) = y(t) - y_m(t)$)

다음은 ARMA parity equation을 시뮬레이션한 결과이다.

이 결과에는 노이즈의 영향과 plant fault의 모습을 동시에 나타내고 있다.

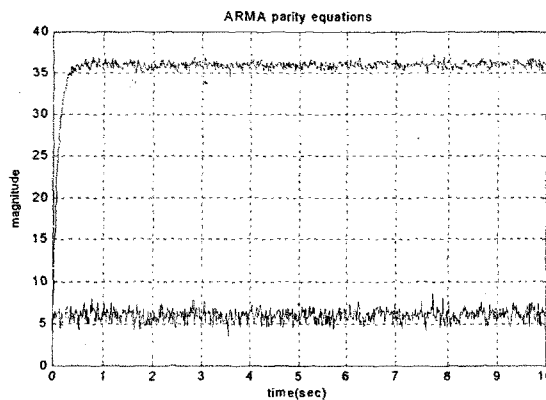


그림 3.1.72 노이즈의 영향과 plant fault

② Plant fault가 존재하는 경우의 Residual 결과 ($r(t) = w^T(z)\alpha(t)$)

Parity equation에서 나온 결과를 transforming vector $w(z)$ 에 의한 가중치를 줘서 noise의 영향을 상쇄시키면, 다음과 같이 plant fault에 영향만을 고려한 residual을 검출할 수 있다.

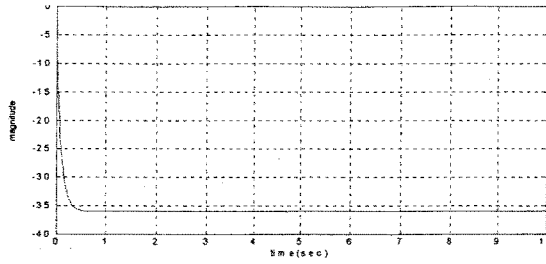


그림 3.1.73 노이즈를 상쇄한 잔차

③ Plant fault가 존재하지 않는 경우의 Residual 결과

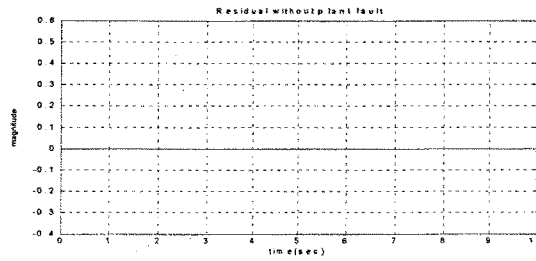


그림 3.1.74 plant fault 가 존재하지 않는 경우

위의 시뮬레이션 결과는 plant fault가 없는 경우에 대해서 시뮬레이션을 한 것이다. 즉, noise의 영향만을 고려했는데, noise의 영향은 Residual 생성할 때, 상쇄가 되어 residual에는 영향을 주지 않는다. 따라서 plant fault가 없는 residual이 0인 결과가 나오게 되어 고장이 발생하지 않았음을 확인할 수 있다.

- MA parity equation을 이용한 Residual

① MA Parity equation ($o_1^*(t)$ & $o_2^*(t)$)

다음 시뮬레이션 결과는 MA parity equation 결과이다. 각 그림은 noise의 영향과 plant fault의 영향을 동시에 나타내고 있다.

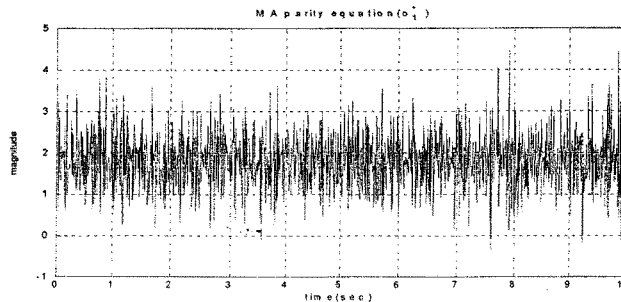


그림 3.1.75 MA parity equation(with fault and noise)

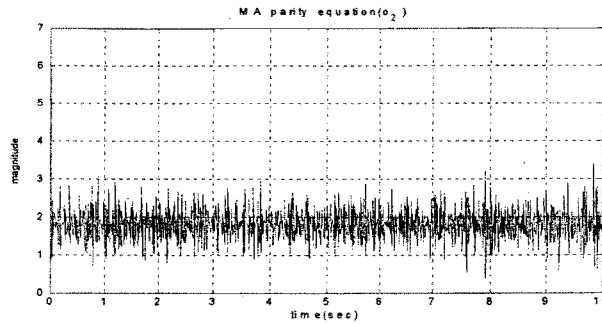


그림 3.1.76 MA parity equation(with fault and noise)

② Plant fault가 존재하는 경우의 Residual 결과 ($r^*(t) = w^T(z)o^*(t)$)

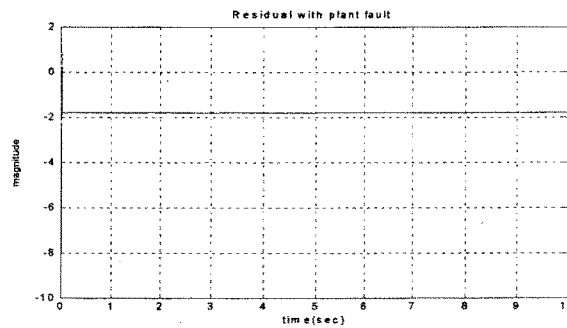


그림 3.1.77 MA parity equation(with fault)

이 결과는 ARMA와 마찬가지로 noise의 영향을 상쇄 시킨 plant fault에 의한 residual만을 검출한 결과이다.

③ Plant fault가 존재하지 않는 경우의 Residual 결과

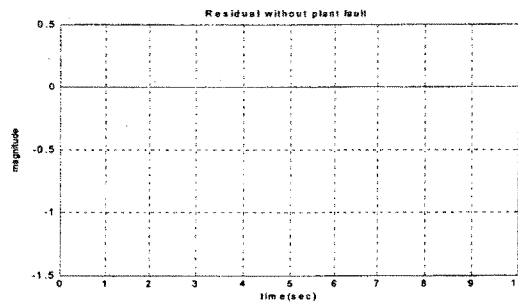


그림 3.1.78 MA parity equation(No fault)

위의 시뮬레이션 결과는 plant fault가 없고, noise만이 존재할 경우에 대한 residual 결과이다. 마찬가지로 residual을 생성할 경우에 noise의 영향은 고려하지 않고, plant fault의 영향만을 고려하므로 plant fault가 없을 경우, noise는 상쇄되어 residual 값이 0으로 plant fault가 없음을 확인할 수 있다.

(다) 고장 검출 필터 (Fault Detection Filter)

fault detection filter는 H행렬을 특별히 선택하여 full-order 상태 추정기를 관측기로 선택한 경우이다. 관측기의 추정 오차에 해당하는 동특성 구조를 고장현상에 대한 단일방

항성을 가지면서 변형될 수 있도록 이득 H를 선정한다.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + k_i f_i(t) \\ y(t) &= Cx(t) + \bar{k}_j \bar{f}_j(t) \end{aligned} \quad (3.1.150)$$

k_i 는 $n \times 1$ 의 fault 방향을 나타내고 총 fault 방향의 개수가 r 개라면 $i = 1, \dots, r$ 이 된다. 예를 들어 i 번째 구동기의 고장은 $k_i = b_i$ 로 나타낼 수 있다. 이 때 b_i 는 B 행렬의 i 번째 열이 된다. $f_i(t)$ 는 임의의 시간에 관한 함수이다. 마찬가지로 \bar{k}_j 와 $\bar{f}_j(t)$ 도 정의될 수 있다.

$$\begin{aligned} \dot{\hat{x}} &= (A - HC)\hat{x} + Bu + Hy \\ \hat{y} &= C\hat{x} \end{aligned} \quad (3.1.151)$$

H는 feedback gain 행렬이고 여기서는 잔차 벡터 $r = y - \hat{y}$ 가 어떠한 fault의 발생에 대해서 어떠한 방향성을 가지도록 선정된다.

구동기나 부품에 고장이 발생한다면 다음과 같은 잔차가 생성되고,

$$\dot{\underline{\epsilon}} = (A - HC)\underline{\epsilon} + k_i f_i, \quad r = C\underline{\epsilon} \quad (3.1.152)$$

센서에 고장이 발생한다면 다음과 같은 잔차가 생성된다.

$$\dot{\underline{\epsilon}}_j = (A - HC)\underline{\epsilon}_j + h_j \bar{f}_j, \quad r_j = C\underline{\epsilon}_j + \bar{k}_j \bar{f}_j \quad (3.1.153)$$

여기서 h_j 는 gain matrix의 j 번째 열이다.

($k^*_j = Af^*_j - \alpha f^*_j$)로 하면

$$\dot{\underline{\epsilon}}_j = (A - HC)\underline{\epsilon}_j - k^*_j \bar{f}_j + f^*_j, \quad r_j = C\underline{\epsilon}_j \quad (3.1.154)$$

식을 사용할 수도 있다.

아래의 그림은 고장이 출력 공간에서 어느 특정한 방향으로만 나타나는 것을 보여준다.

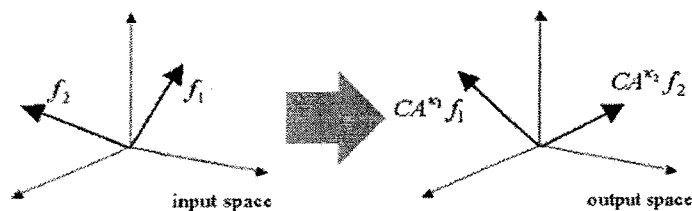


그림 3.1.79 고장 출력 공간

(라) 매개변수 인식(identification) 방법

시스템의 변수 모델을 보통 다음과 같은 형태로 선택한다.

$$a_n y^{(n)}(t) + \dots + a_1 y(t) + y = b_0 u(t) + \dots + b_m u^{(m)}(t) \quad (3.1.155)$$

그리고 새로운 모델 변수와 물리적 변수 사이의 관계식을 설정한다.

$$\underline{\theta} = f(p) \quad (3.1.156)$$

실제 시스템의 입력과 출력을 통하여 모델 변수 벡터($\underline{\theta}$)를 찾은 후, 얻어진 모델 변수 벡터를 이용하여 물리적 변수의 값을 알아낸다.

$$p = f^{-1}(\underline{\theta}) \quad (3.1.157)$$

원래의 값과 오차인 Δp 를 계산해낸 다음, 물리적 변수의 변화량인 Δp 과 고장 사이의 관계를 통하여 고장여부를 판단한다. 이 과정을 그림으로 살펴보면 다음과 같이 나타난다.

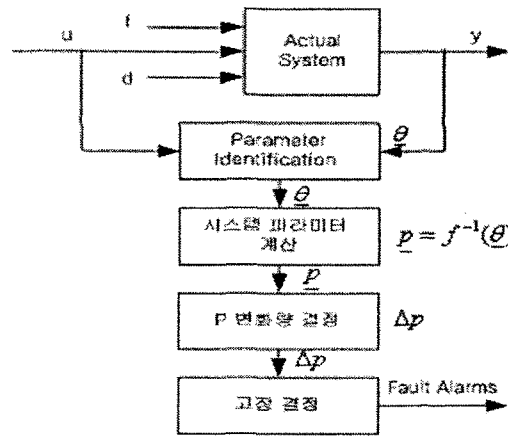


그림 3.1.80 매개변수 인식 방법

(마)매개변수 추정(Estimation) 기법 - 퍼지(Fuzzy) 이론

퍼지로직은 고장이 발생하였을 경우, 상태변수의 동특성을 이용하여 단일 고장일 경우와 복수의 고장이 발생하였을 경우에 대하여 적용이 가능하도록 구성하였다. 특히, 퍼지 시스템에서 각 작동기 고장의 특성을 독립적으로 나타낼 수 있는 시스템 출력신호를 사용하여 출력신호의 연성효과를 최소화하였다. 고장이 발생하였다고 판단하면 재형상 제어기가 작동하여 원하는 임무를 수행하도록 하는 시스템을 설계하였다. 제안한 고장검출 및 분리 알고리즘의 성능을 검증하기 위해서 비선형 항공기 모델에 대한 수치 시뮬레이션을 수행하였다.

고장검출

항공기에 고장이 발생하면 고장 여부를 판별하고, 항공기 상태변수의 동적 특성을 이용하여 고장을 검출해 내어야 한다. 항공기의 수학적 모델과 실제 출력을 비교함으로써 고장검출 신호를 정의하면, 정상상태일 때에는 고장검출 신호의 값이 작을 것이고, 고장이 발생하면 출력 신호가 일정한 문턱값을 넘어서게 되어 고장발생 여부를 판별할 수 있다.

고장위치 검출

고장이 발생했다는 것을 감지하면 어느 작동기에서 어느 정도의 고장이 발생하였는가를 검출하여야 하는데, 이 과정이 고장진단 과정이다. 일단 고장진단이 이루어지면, 그 정

보를 이용하여 제어기를 적절히 재설계하여 주어진 임무를 수행하게 하는 고장허용제어를 가능하게 한다. 따라서 정확한 고장진단 과정은 매우 중요한 과정이므로, 본 연구에서는 지능형 시스템인 퍼지로지 기반의 고장검출 기법에 대해 연구하였다.

퍼지로지 소개

퍼지 시스템은 지식기반 혹은 법칙기반 시스템이다. 따라서 항공기에 고장이 발생하였을 경우, 상태변수 특성을 퍼지 규칙으로 설정하여 고장 위치를 검출하는 검출기를 만들 수 있다. 상태변수의 동특성을 연속적인 멤버십 함수로 퍼지화 하여 퍼지입력을 구성하였다. 퍼지 입력은 퍼지 추론 엔진의 규칙에 의해 퍼지출력값을 내보내며, 최종적으로 출력 멤버십 함수에 의해 고장 위치를 검출하게 된다. 이러한 일련의 과정을 그림 3.1.81에 도시화 하였다. 그림 3.1.82에는 퍼지로직을 이용하여 고장을 검출하고 진단하는 과정을 나타내었다.

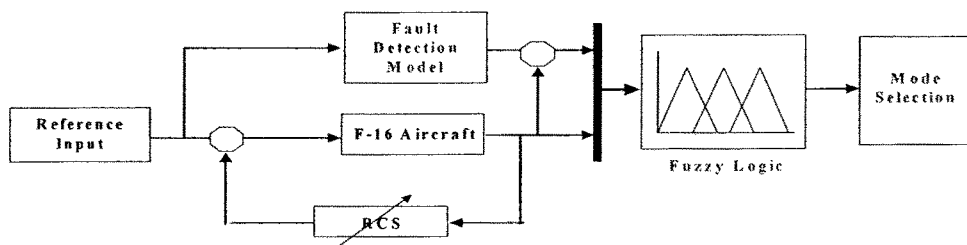


그림 3.1.81 고장 검출 및 진단 알고리즘

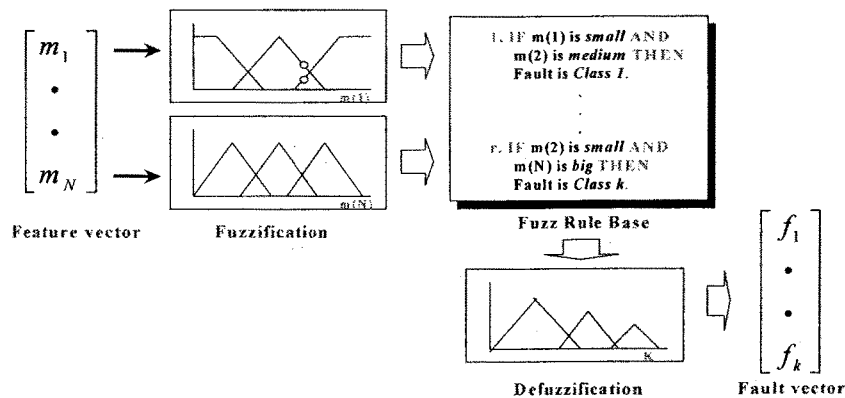


그림 3.1.82 퍼지 시스템

- 퍼지입력

퍼지 시스템으로 들어가는 입력은 항공기에서 출력되는 상태변수로 구성한다. 고장의 발생유무를 판별한 후에 일정한 샘플 시간동안 상태변수의 평균값을 얻어, 이 값들을 이용하여 퍼지 시스템에서 고장발생 위치를 검출하게 된다. 퍼지로직에 들어가는 입력변수는 에일러론, 엘리베이터, 그리고 러더 각각의 구동기에 고장 발생 시 특성을 독립적으로 잘 나타낼 수 있는 상태값을 사용하였다. 본 연구에서는 퍼지 시스템으로 들어가는 3개의 입력을 $[signal_A, signal_E, signal_R]$ 로 표현하였다. 각 멤버십 함수는 Trap 함수로 구성하였다. 퍼지입력으로 들어가기 위한 퍼지화 및 퍼지 규칙에 의해 계산된 퍼지 출력이 시스템으로부터 출력되는 과정을 그림 3.1.83에 나타내었다.

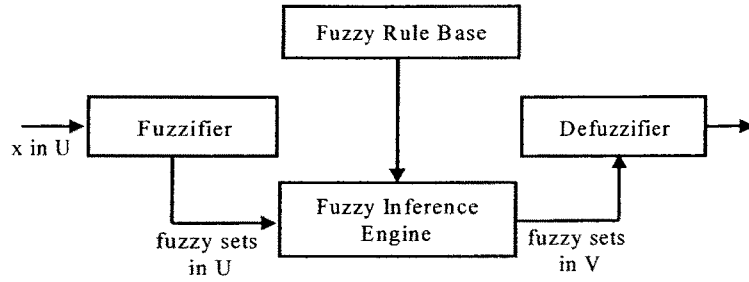


그림 3.1.83 퍼지 시스템의 기본구조

- 퍼지규칙

고장 위치의 검출은 퍼지 규칙에 의해 계산된다. 본 연구에서 사용한 각각의 퍼지 출력에 대한 퍼지 규칙을 표 3.5.1에 정리하였다. 퍼지 입력의 모든 신호들의 값이 모두 *small*을 나타내면, 퍼지 출력은 정상상태인 *normal*을 나타내도록 퍼지 규칙을 설정하였다.

고장검출 신호로써 고장발생 여부를 검출하는 경우에 외란 등에 의해 오경보를 발생할 수가 있다. 따라서 고장이 발생하지 않았는데 고장검출 신호가 문턱값을 넘어 고장이 발생하였다고 잘못된 정보를 발령하는 것을 방지하기 위해서, 퍼지 시스템으로 들어가는 신호들을 조사하여 오경보 발생 요소를 제거하였다. 일반적으로 각각의 입력신호들을 살펴보면, 각 고장에 대해서 고유의 크기를 가지고 있으므로 이러한 정보를 활용하였다. 한편, 다중 고장이 발생할 경우에는 일반적으로 신호들의 연성으로 인하여 고장검출 및 분리의 성능이 떨어진다. 따라서 각 작동기 고장의 특성을 독립적으로 나타낼 수 있는 시스템 출력신호를 사용하여 신호들의 연성 효과를 최소화하였고, 연성에 의한 고장검출 성능 저하를 방지하도록 하였다.

표 3.1.9 퍼지규칙

Class \ Features		Signal_A	Signal_E	Signal_R
Normal		<i>small</i>	<i>small</i>	<i>small</i>
S I N G L E	Aileron	<i>big</i>	<i>small</i>	<i>medium</i>
	Elevator	<i>small</i>	<i>big</i>	<i>small</i>
	Rudder	<i>medium</i>	<i>small</i>	<i>big</i>
D O U B L E	Aileron+Elevator	<i>big</i>	<i>big</i>	<i>medium</i>
	Aileron+Rudder	<i>big</i>	<i>small</i>	<i>big</i>
	Elevator+Rudder	<i>medium</i>	<i>big</i>	<i>big</i>

- 퍼지출력

퍼지출력은 그림 3.1.84과 같이 구성하였다. 각 출력값에 대하여 단일고장은 0.5-5.5의 모드값이 출력되도록 설정하였고, 동시에 발생하는 고장에 대해서는 7.5-12.5의 모드값이 출력되도록 하였다. 이때, OK, A, E, R, AE, AR, ER은 각각 Normal, Aileron 고장, Elevator 고장, Rudder 고장, Aileron & Elevator 고장, Aileron & Rudder 고장,

Elevator & Rudder 고장을 나타낸다. 퍼지출력 멤버십 함수는 퍼지입력 멤버십 함수와 마찬가지로 Trap 함수로 구성하였다.

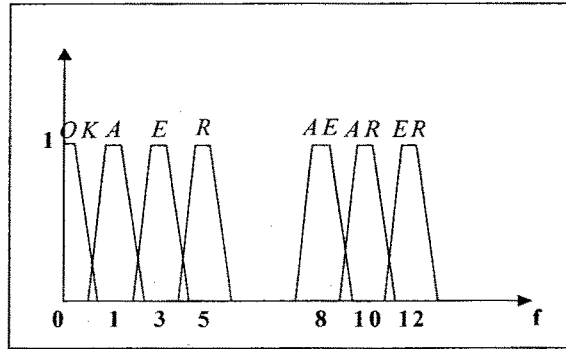


그림 3.1.84 퍼지출력

- 고장검출 및 재형상 제어 시뮬레이션

제안한 기법의 성능을 검증하기 위한 고장상황으로, 일정한 받음각을 유지하면서 20도의 롤기동을 수행한 뒤에 원래의 자세로 돌아오는 항공기의 기동 중, 에일러론 작동기에 고장이 발생한다고 가정하고, 14초 후에 에일러론, 엘리베이터에 각각 8도, 5도의 bias가 발생한다고 놓고 시뮬레이션을 수행하였다. 이때 모델은 F-16 항공기의 비선형 6자유도 운동방정식을 사용하였다.

그림 3.1.85은 시스템과 정상모델 간의 신호를 비교함으로써 고장발생을 적절히 검출하는가를 평가하였다. 문턱값은 0.7로 설정하였으며, 14초에서 고장이 발생하여 문턱값을 넘어서게 되어 고장이 발생하였다는 것을 알려주고 있다. 그림 3.1.86는 고장발생 경보가 울린 후에 약 1.2초간 퍼지 시스템에서 입력신호를 검사하여 고장을 발생여부를 한번 더 확인하고, 어느 작동기에서 고장이 발생하였는지를 판단하기 위한 고장모드를 나타내고 있다. 앞의 퍼지출력에 의해 에일러론과 엘리베이터에서 동시에 고장이 발생하는 경우는 8번 모드로 설정되어 있고, 시뮬레이션 수행 결과, 약 15초 후에 정상모드에서 8번 모드를 가리키고 있는 것을 확인할 수 있다. 따라서 고장의 위치를 정확히 검출하고 있음을 알 수 있다.

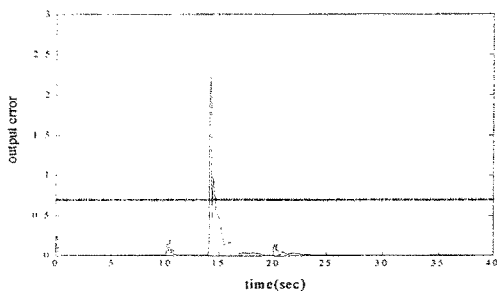


그림 3.1.85 고장에 대한 고장검출 신호

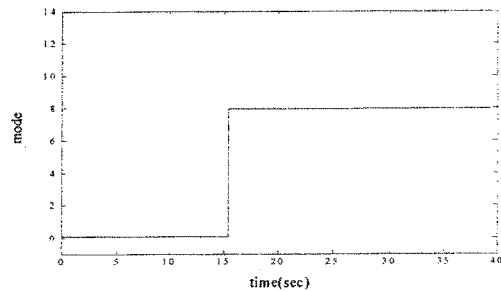


그림 3.1.86 검출된 고장 모드

고장이 발생하기 전에는 고정 제어이득을 사용하다가 고장 진단한 후, 재형상 제어를 작동시켰을 때의 상태변수 시간반응은 그림 3.1.87과 그림 3.1.88과 같다. 고장 판단 후,

고장 위치를 검출한 후 재형상 제어기가 작동된다. 이때, 실선은 시스템 출력값을 나타내고, 점선은 모델 출력값을 나타낸다. 재형상 제어기는 적응제어 이득을 사용하여 고장 발생에 대해 잘 대처하고 있음을 알 수 있다. 물각을 보면 고장이 발생한 14초에 고장발생에 의한 영향으로 롤 기동명령을 추종하지 못하다가 재형상 제어기의 작동으로 기동명령을 잘 추종하게 됨을 볼 수 있다. 따라서 본 연구에서 제안한 퍼지로그직을 이용한 고장진단 시스템은 복수의 고장을 효과적으로 검출할 수 있음을 보여주고 있다.

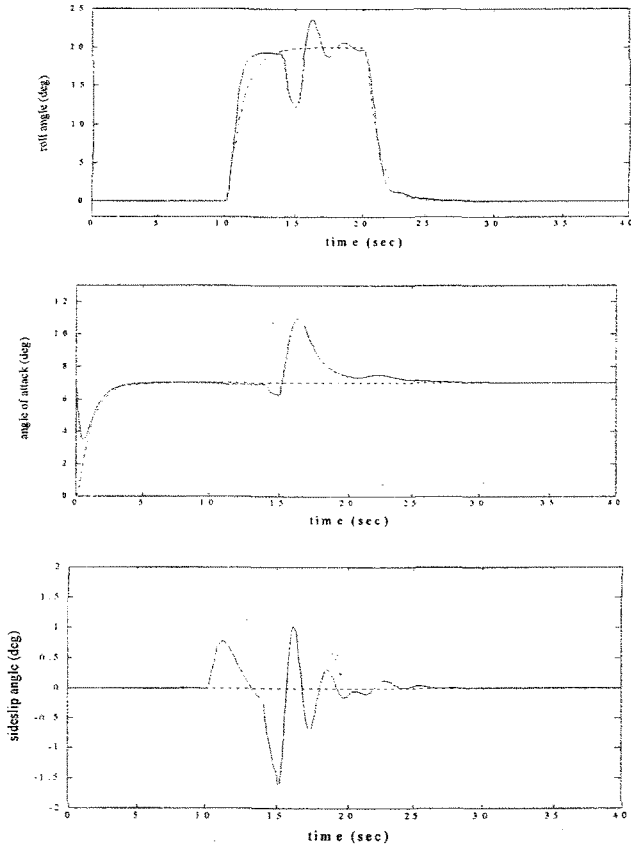
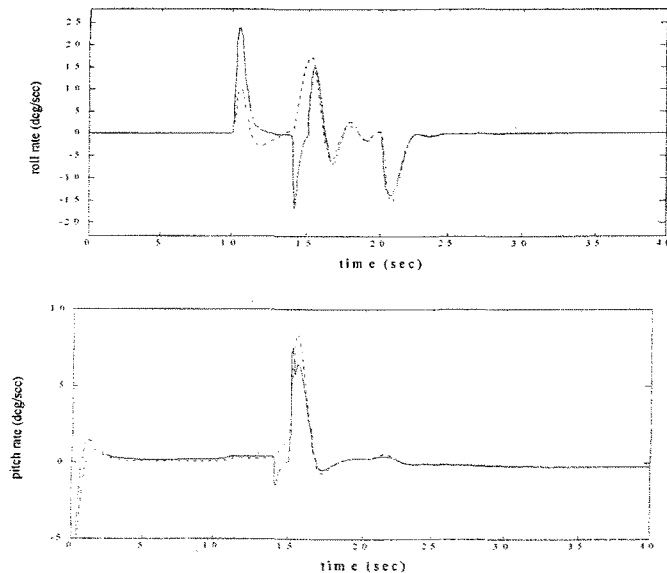


그림 3.1.87 고장 발생 시 각도 상태변수 시간반응



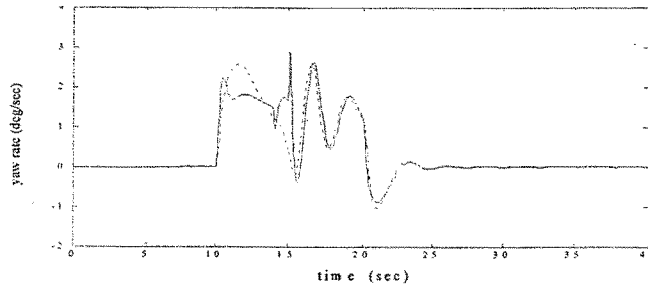


그림 3.1.88 고장 발생 시 각속도 상태변수 시간반응

지금까지 퍼지로그직을 이용한 항공기 작동기의 고장 및 고장 위치를 검출에 대해 살펴보았다. 제안한 기법에 의해 고장이 검출되면 모델을 추종하는 재형상 제어기가 작동하도록 하였다. 기존의 연구들이 단일 고장에 대해서는 효과적으로 검증하고 있으나, 동시에 발생하는 복수의 고장에 대해서는 신호들의 연성으로 인하여 고장검출 및 분리의 성능이 떨어진다. 한편, 검출필터를 이용하거나 관측기를 기반으로 한 고장검출 기법은 측정잡음이나 외란에 대해 민감한 반응을 나타낸다. 본 연구에서는 강건한 특성을 가지는 지능형 제어기인 퍼지로그직을 이용하여 고장을 검출하는 기법을 제안하였다. 동시에 발생하는 두 개의 고장에 대해서 각 작동기 고장 특성을 독립적으로 나타낼 수 있는 시스템 출력신호를 사용하였다. 퍼지로그직에서는 출력신호의 연성의 효과를 최소화하여 비교적 정확하게 고장을 검출하고 고장 모드를 선언하도록 하였다. 제안한 알고리즘의 성능을 검증하기 위해서 6자유도 F-16 비선형 항공기 모델을 이용한 수치 시뮬레이션을 수행하였다.

퍼지로그직의 향후 연구 방향은 첫째, 감지기의 동역학적 특성에 의해서 강건성이 약화될 수 있으므로 이에 대한 영향을 고려한 연구가 필요하다. 둘째, 고장 모델에 대한 연구도 병행되어야 할 것이다. 본 연구에서 고려한 고장은 작동기에 발생하는 bias를 고장으로 설정하였지만, 조종면의 고정(stuck), 조종면 파손과 같은 고장이 발생할 경우에는 재형상 제어가 어렵게 되므로 이러한 경우에 대해 재형상 제어를 수행할 수 있는 연구가 필요하다고 사료된다.

내용누락

여분을 가지고 있는 경우에만 적용할 수 있다.

(라) 교차 채널 감시 기법(CCM: Cross Channel Monitoring)

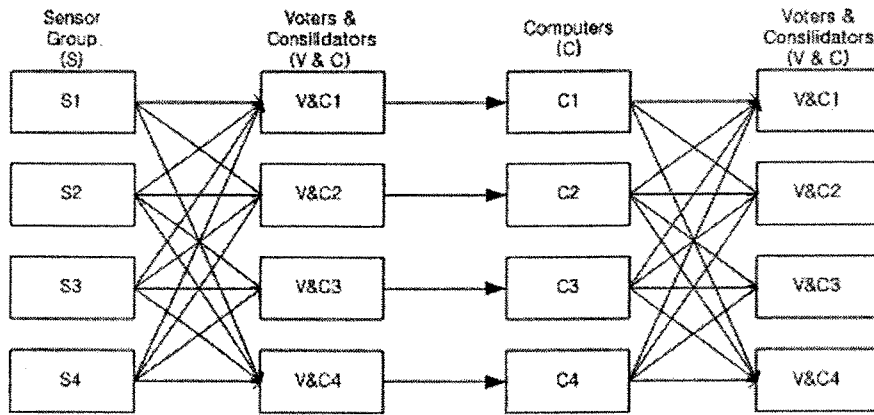


그림 3.1.89 Qudraplex System Architecture

다중화 되어 있는 시스템의 출력을 서로 비교하여 고장을 진단한다. 출력의 기준값을 정하고, 그 기준값으로부터 각 채널이 얼마나 벗어나 있는지, 그리고 어느 정도 시간동안 벗어나 있는지를 기준으로 그 채널의 고장을 판단한다. 위에서 언급한 두 값을 각각 한계치 (Threshold)와 유효 시간(Confirmation Time)이라고 한다.

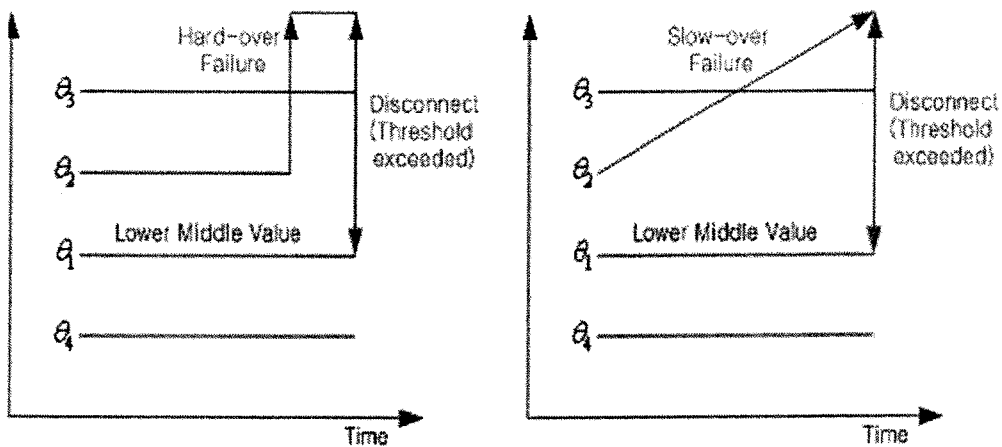


그림 3.1.90 4중 센서의 고장 검출 알고리즘 - 첫 번째 고장 발생

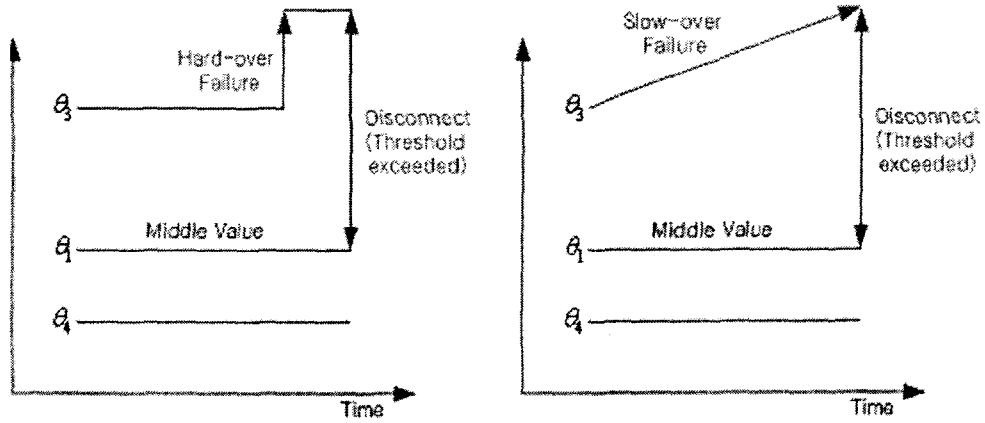


그림 3.1.91 3중 센서의 고장 검출 알고리즘 - 두 번째 고장 발생

(2) Non-Sequential Method

(가) Pattern Recognition Methods

고장 상황에 대한 시스템의 반응을 여러 카테고리로 저장하여, 실제 테스트에서 시스템이 나타내는 반응을 저장된 반응과 비교하여 고장을 검출한다. 패턴 인식 방법의 대부분이 고장 사전(Fault Dictionary)이나 확인 행렬(Recognition Matrix)을 필요로 한다. 확인 행렬은 다음과 같이 고장 상황에서의 시스템 반응의 변화량(deviation)들로 표현된다.

$$x = [x_1 \ x_2 \ \dots \ x_m]^T$$

$$R = \{x_{ij}\} \quad i=1,2,\dots,M \text{ measurements}$$

$$j=1,2,\dots,N \text{ fault cases}$$

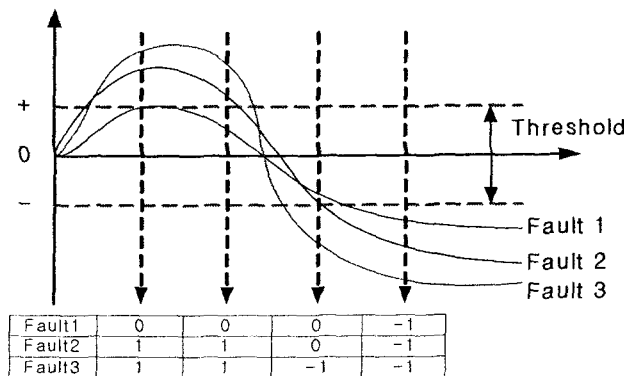


그림 3.1.92 신호의 패턴을 이용한 고장의 분류

Matrix Recognition

정상시와 고장이 일어났을 경우의 영역(region)을 정의하여 그에 알맞도록 테스트 점들을 정한다. 행렬은 입력 벡터와 그 자신의 곱이 출력 벡터를 생성하는 방식으로 결정되며, 절대값이 가장 큰 출력 벡터가 고장 영역을 알려 준다. 이 영역에 해당하는 행렬 성분이 고장 성분이 된다.

Sequential Recognition Technique

행렬 확인(Matrix Recognition)방식과 유사하지만, 각 영역의 경계치가 측정치와 부합하는지의 여부를 확인하기만 하면 된다.

Bode Diagram Technique

이 기법은 지정된 주파수에서 모든 구성 요소들의 측정치의 변화량과 그와 연관된 이득값의 변화를 망라한 고장 사건을 만드는 작업이 필요하다. 이득값의 변화는 기준값보다 위/아래인지, 아니면 기준값 이내인지에 따라 "+", "-", "0"으로 정해진다. 시스템의 고장은 패턴 벡터(Pattern Vector)를 이미 저장된 패턴과 비교하여 진단할 수 있다. 테스트 주파수는 가장 낮은 브레이크 점(break point)아래, 가장 높은 브레이크 점 위 그리고 인접한 두 브레이크 점 사이 로그 스케일의 중간점을 선택한다. 그리고 각각의 복소 브레이크 점에 하나의 테스트 주파수가 추가된다.

Voting Technique

보팅(Voting) 기법은 보드 다이어그램 기법과 같이 정상 상태에서 +1, -1 혹은 0으로의 응답 변화를 측정한다. 그러므로 확인 행렬은 +1, -1 혹은 0 으로만 구성된다. 시스템의 응답 변화 벡터도 같은 방법으로 측정한다. 시스템의 응답 변화 벡터는 이미 저장된 패턴들과 비교하여 보팅이 이루어진다.

- ① 고장인 경우
- ② 고장이 아닌 경우
- ③ 보팅이 이루어지지 않는 경우

보팅을 하기 위한 논리 기법은 여러 형태가 있다.

Template Matching

시스템의 반응 데이터를 정상 상태에서의 변화량의 크기에 따라 코드화 한다. 예를 들어 각의 변화량이 4°, -8°, -35° 그리고 -12° 인 경우 범위에 따라서 1, 1, 4, 2 로 코드화 된다. 코드화된 측정치는 이미 알려진 고장들의 유사한 코드 패턴과 비교하여 고장을 진단할 수 있다. 그러나 일반적인 변화량을 가지는 아날로그 시스템에서 정확한 패턴 매칭이 이루어지기는 어려울 것으로 보인다.

Cross-Product Method

저장되어 있는 응답 변화 벡터와 측정된 벡터의 외적 값이 가장 클 때 두 벡터의 연관성이 가장 크다는 성질을 이용한다.

표 3.1.11 고장 상황의 예

시간	0	5	10	15	20	25	30	35	40
fault 1	0	-1.5	0	0.5	0.5	0.4	0.3	0.2	0.1
fault 2	0	-3.0	-3.0	-1.5	0.5	1.5	1.6	1.4	1.0
fault 3	0	0.6	1.5	2.4	3.0	2.5	2.0	1.5	1.0
y	0	-1.0	-0.5	1.0	0	0.9	-0.2	0.7	-0.4

계산된 inner product는 다음과 같다.

$$F1 = 2.40, F2 = 4.61, F3 = 3.55$$

결과적으로 2번 고장 상황일 확률이 크다. 그러나 다음과 같은 경우를 생각해보자.

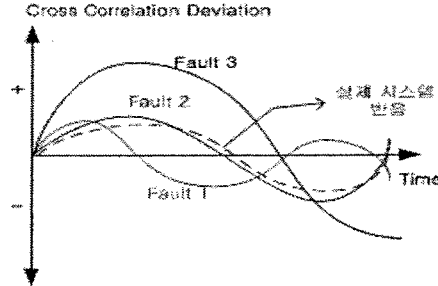


그림 3.1.93 고장 신호의 예

분명히 Fault 2 이라고 판정되어야 한다는 것을 알 수 있다. 그러나 위와 같은 방법으로 계산을 해보면 Fault 3의 편차의 절대값이 크기 때문에 고장이 Fault 3 라고 판정될 것이다. 이러한 현상을 방지하기 위하여 정규화(normalization)를 수행한다.

$$F_j^* = \frac{\sum_{i=1}^M x_{ij}y_i}{\|x_{ij}\|_E} \quad (3.1.158)$$

위의 예제를 정규화하면,

$$F1 = 1.37, F2 = 0.87, F3 = 0.64$$

실제로는 Fault 1 에 더 가까운 고장 상황인 것을 알 수 있다.

Bi-Linear Transformation

파라미터 p를 다음과 같은 방식으로 좌표 변환을 한다.

$$T(s, p) = \frac{A(s)p + B(s)}{C(s)p + D(s)} \quad (3.1.159)$$

여기서 A, B, C 그리고 D는 쌍일차(bi-linear) 변환이라고 불리는 복소 계수이며, p 평면의 직선을 T 평면의 원으로 변환시킨다. 또한 s는 라플라스(Laplace) 연산자이며 jw로 바꾸면 시스템의 주파수 응답을 살펴볼 수 있다. 주어진 주파수에서 각각의 p에 대하여 T 평면에서 원형 궤적(locus)들이 그려질 수 있다. 원형 궤적은 시스템이 정상 상태인 지점에서 교차할 것이며 하나의 고장이 발생한 경우에는 원형 궤적들 중의 하나 위에 시스템이 존재할 것이다. 그러나 고장이 아닌 자연적인 변화량으로도 고장이라고 진단할 수 있는 가능성이 많으며, 파라미터 수가 많을수록 매우 힘든 방법이다.

Nearest Neighbor Rule

시스템의 테스트 벡터가 어떠한 고장 벡터와 가장 가까운 거리를 가지는지를 계산하여 고장을 인식하는 방법이다. 이 방법은 매우 간단하지만, 많은 파라미터의 변화를 다루고,

최소의 차원을 가진 확인 행렬로 효율적인 데이터 처리를 하기 위해서는 데이터를 정규화(Normalizing)해야 한다.

Fuzzy Sets

어떠한 실제 시스템이든 측정 잡음과 고장에 의하지 않은 측정값 변화가 고장 계층의 경계를 모호하게 한다(fuzziness). 0과 1사이의 값들을 가지는 가중 함수(Weighting function)가 고장 상황의 소속성(membership)을 평가하는데 사용된다. 1에 가까운 값을 가질수록 그 고장 상황에 해당될 확률이 커진다.

(나) Probability Methods

Key Element Search Method

민감성 함수(Sensitivity Function)의 정보로 예측한 편차와 관찰된 정상 상태와의 편차와의 차를 최소화하는 고장 지표(index)를 계산한다. 민감성 함수는 다음과 같다.

$$\frac{\partial f_i}{\partial x_i} : \text{시스템 반응에 대한 고장의 변화율}$$

다음과 같은 성능 지수(performance index)를 최소화하는 구성 원소의 값은 고장일 확률이 크다고 여겨진다.

$$S_J = \sum_{i=1}^M (y_i - \frac{\partial f_i}{\partial x_J} \Delta x_J)^2 \tag{3.1.160}$$

Inverse Probability

시스템의 정상 상태 값과 실제 응답 값과의 차이는 고장 확률을 추정하는데 사용된다. 가장 큰 확률을 가지는 구성 원소가 고장일 확률이 크다.

Entropy Method

엔트로피는 불확실성의 통계학적 척도이며, 다음과 같이 정의된다.

$$H = -p_i \log(p_i) \tag{3.1.161}$$

여기서 p_i 는 i 번째 구성 원소의 고장 확률이다. H 는 알려진 고장 확률을 가지고 있는 구성 원소간의 산포도를 측정하며 확률적으로 고장 진단 결정을 하기 위하여 사용된다.

(다) 유사 확률 기법(GLT: Generalized Likelihood Ratio Test)

센서 시스템에 고장이 없는 경우를 H_0 , 고장이 있는 경우를 H_1 이라 하고 두 경우를 비교해 보자.

표 3.1.12 패러티 벡터의 표현

	H_0	H_1
패러티 벡터	$p = VHx + V\epsilon$	$p = VHx + V\epsilon + Vb$
패러티 벡터의 평균	$E[p] = 0$	$E[p] = Vb = \mu$
패러티 벡터의 분산	$E[pp^T] = VV^T$	$E[(p - \mu)(p - \mu)^T] = VV^T$

고장이 발생하면 패러티 벡터의 분산 값은 변화가 없지만, 패러티 벡터의 평균값은 0이 아닌 일정값을 가지게 된다. 그러므로 패러티 벡터의 평균값을 이용하면 고장에 대한 정보를 얻을 수 있다는 것이 이 기법의 기본 개념이다.

측정 잡음 ϵ 이 가우시안(Gaussian)이므로, p 도 가우시안 랜덤 벡터이다. 그러므로 위의 두 가지 상황에 대하여 다음과 같은 조건 확률 밀도 함수를 생각할 수 있다.

$$P(p | H_0) = \frac{1}{(2\pi)^{(l-n)/2} |VV^T|^{1/2}} \exp\left(-\frac{1}{2} p^T (VV^T)^{-1} p\right) \quad (3.1.162)$$

$$P(p | H_1) = \frac{1}{(2\pi)^{(l-n)/2} |VV^T|^{1/2}} \exp\left(-\frac{1}{2} (p - \mu)^T (VV^T)^{-1} (p - \mu)\right) \quad (3.1.163)$$

위의 두 식을 이용하여 Log likelihood ratio $\Lambda(p)$ 를 정의한다.

$$\Lambda(p) = \max_p \ln \frac{P(p | H_1)}{P(p | H_0)} = \max_p \frac{1}{2} [p^T (VV^T)^{-1} p - (p - \mu)^T (VV^T)^{-1} (p - \mu)] \quad (3.1.164)$$

$\Lambda(p)$ 값을 극대화하기 위해서는 위 식의 첫 항은 고정값이므로, 뒷항의 값을 크게 해야 한다. 그러므로 최대 유사 추정치(Maximum likelihood estimate) $\hat{\mu} = p$ 가 된다.

이 경우 $\Lambda(p)$ 값이 고장의 영향을 가장 잘 반영하는 함수이므로 고장 검출 함수(FD) 로 정한다.

$$FD = p^T (VV^T)^{-1} p \quad (3.1.165)$$

H_k 를 k 번째 센서에 고장이 발생한 상황이라고 한다면, H_k 상황에서의 고장 분리 함수(FI)는 다음과 같이 구할 수 있다.

$$FI_k = \frac{(p^T (VV^T)^{-1} v_{ck})^2}{v_{ck}^T (VV^T)^{-1} v_{ck}} \quad (3.1.166)$$

만약 FD 함수를 통하여 고장이 확인되면, 위 식을 이용하여 주어진 p에 대하여 센서 수 만큼의 FI 함수 값을 계산해서 가장 큰 값을 보여주는 k 번째 센서가 고장이 났다고 판단할 수 있다.

순차 확률 기법(SPRT: Sequential Probability Ratio Test)

현재의 측정값만을 이용하는 것이 아니라 현재의 유수 값과 과거의 유수 값들을 누적하여 고장을 검출하는 방법이다. 이 기법의 특징은 작은 고장 신호도 검출할 수 있으며 Faule alarm 확률을 줄일 수 있다는 것이다. 그러나 고장 검출 시에 시간 지연이 크게 발생하며, 시간에 따른 유수 값들이 서로 독립성이 보장되어야만 한다. 실제 관성 센서 배치에서는 장착 비정렬 각과 변환 계수 오차(scale factor error) 등의 오차가 존재하기 때문에 독립성

을 보장 받기가 쉽지 않다. 그러므로 실제 적용을 위해서는 이러한 문제들을 개선해야 한다.

(라) Parameter Estimation Methods

Interactive Method

이 기법은 시스템의 위상(topology)과 구성 원소의 정상 상태 값이 필요하다. 파라미터 값들의 변화 양상도 미리 알고 있어야 한다. 테스트 데이터는 구성 원소의 값들을 추정하기 위해 사용된다. 이 추정 값을 실제 정상 상태 값과 비교하여 고장을 판단한다. 이러한 확률적 과정으로 안정적인 추정을 하기 위해서는 많은 양의 테스트 데이터가 필요하다.

Model Adjusting Technique

이 기법은 시스템 설계(design)기법을 확장했다고 할 수 있다. 먼저 시스템의 수학적 모델인 전달 함수(transfer function)를 결정한다. 전달 함수의 계수들을 추정하기 위하여 테스트 데이터를 이용한 최소 자승법을 시행한다. 여기서 결정된 계수들과 실제 구성 원소 값들 사이의 관계를 이용하여 고장을 진단한다.

Parameter Updating

이 기법은 파라미터 값들을 온라인(on-line)으로 업데이트(update)해야 하는 시스템에 적합하다. 칼만 필터(Kalman filter)와 잠재 함수(potential function)를 사용하는 방법이 많이 쓰인다.

(마) Residual Method

1)추정 오차 제곱근 기법(SE: Squared Error Method)

이 기법은 1자유도 센서 6개를 대칭적으로 배치할 경우에 대해서 유도된 방법으로서, 다른 형태의 배치에는 사용할 수 없다. 먼저 유도 과정을 살펴보면 다음과 같다.

$$\hat{m} = H\hat{x} \tag{3.1.167}$$

$$r = m - \hat{m} = m - H\hat{x} \tag{3.1.168}$$

여기서

$$\hat{x} = \overline{(H^T H)^{-1} H^T m} \tag{3.1.169}$$

$m = [m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6]^T$: 6개의 센서 측정치(각속도 혹은 가속도)

H : 6개의 센서들의 동체 좌표축에 대한 방향 벡터로 구성된 측정 행렬($H \in R^{6 \times 3}$)

x : 동체 좌표축을 기준으로 한 입력 각속도 혹은 가속도($x \in R^3$)

$r = [r_1 \ r_2 \ r_3 \ r_4 \ r_5 \ r_6]^T$: 유수(residual)

대칭적 배치구조의 측정행렬 H 를 직접 대입하면, 1번 센서의 유수 값은 다음과 같다.

$$r_1 = \frac{1}{2} [m_1 - (0.2)^{1/2}(m_2 + m_3 + m_4 + m_5 - m_6)] \tag{3.1.170}$$

그리고 고장이 발생하지 않았을 때의 추정오차의 제곱의 합(TSE_0 : total squared error)을 다음과 같이 정의한다.

$$TSE_0 = r^T r = r_1^2 + r_2^2 + r_3^2 + r_4^2 + r_5^2 + r_6^2 \quad (3.1.171)$$

고장이 발생하지 않는다면, 식(3.1.170)과 같이 6개의 다른 센서의 유수 값을 계산하여 식(3.1.171)에 대입해보면 $TSE_0=0$ 을 유지하게 되는 것을 알 수 있다. 하지만 임의의 센서에 고장이 발생하면, 그 센서의 추정 오차 값이 커지면서 다른 센서의 추정 오차 값에도 영향을 주게 되어 TSE_0 는 0을 유지하지 못한다. 실제로 i 번째 센서가 고장 났다고 가정하면, TSE_0 는 다음과 같다.

$$TSE_0 - 2r_i^2 = 0 \quad (3.1.172)$$

이것은, 고장이 발생한 i 번째 센서의 추정오차 값의 제곱이 TSE_0 의 50%에 해당한다는 의미를 가진다. 그러므로 각 센서의 추정오차값의 제곱을 TSE_0 의 값과 비교한다면 고장 센서를 분리할 수 있다.

2) 최소 자승 유수 기법(LSRA: Least Square Residuals Approach)

최소 자승 추정치(Least Square Estimate)와 유수 벡터는 다음과 같다.

$$\begin{aligned} \hat{x} &= \operatorname{argmin} \|m - Hx\|^2 = (H^T H)^{-1} H^T m \\ \eta &= m - \hat{m} = m - H\hat{x} = m - H[(H^T H)^{-1} H^T] m \\ &= (I - H(H^T H)^{-1} H^T) m = V^T V m \end{aligned} \quad (3.1.173)$$

패러티 벡터의 정의에 의하여 유수 벡터는 다음과 같은 식이 된다.

$$\eta = V^T V m = V^T p \quad (3.1.174)$$

그리고 유수 벡터는 $\eta = m - \hat{m} = (Hx + \epsilon) - H\hat{x} = H(x - \hat{x}) + \epsilon$ 이므로 측정 잡음의 추정치이며 $H(x - \hat{x})$ 는 추정 오차가 되는 것을 알 수 있다.

유수 벡터의 평균과 분산은 다음과 같다.

$$E[\eta] = E[V^T p] = V^T E[p] = 0 \quad (3.1.175)$$

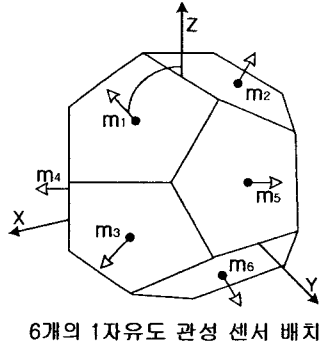
$$E[\eta\eta^T] = E[(V^T p)(p^T V)] = V^T E[pp^T] V = I - H(H^T H)^{-1} H^T \quad (3.1.176)$$

이를 바탕으로 χ^2 분포에 기초한 FDI를 수행할 수 있다.

(바) 패러티 공간을 이용하는 방법

센서의 기하학적 배치 구조를 이용한 기법

센서를 동일 평면상에 3개 이상 놓이지 않도록 배치하면 3개의 센서쌍은 서로 독립이므로 4개의 센서를 선형 결합하여 입력 값에 무관한 패러티 식을 구성할 수 있다. 만약 센서가 n 개라면 패러티 식의 개수는 ${}_n C_4$ 이다. 그러나 센서의 개수가 증가할수록 패러티 식의 개수는 급격히 증가하여 복잡해지는 단점이 있다.



$$H^T = \begin{bmatrix} s & -s & c & c & 0 & 0 \\ 0 & 0 & s & -s & c & c \\ c & c & 0 & 0 & s & -s \end{bmatrix}$$

where $c = \cos(64.435^\circ)$, $s = \sin(64.435^\circ)$

표 3.1.13 패리티 식

패리티 식 번호	사용된 센서	Parity Equations	SCODE
1	1 2 3 4	$p_1 = (m_1 - m_2)c - (m_3 + m_4)s \approx 0$	1 1 1 1 0 0
2	1 2 3 5	$p_2 = (m_2 + m_3)c - (m_1 + m_5)s \approx 0$	1 1 1 0 1 0
3	1 2 3 6	$p_3 = (m_3 - m_1)c - (m_2 - m_6)s \approx 0$	1 1 1 0 0 1
4	1 2 4 5	$p_4 = (m_4 - m_1)c - (m_2 + m_5)s \approx 0$	1 1 0 1 1 0
5	1 2 4 6	$p_5 = (m_2 + m_4)c - (m_1 - m_6)s \approx 0$	1 1 0 1 0 1
6	1 2 5 6	$p_6 = (m_5 - m_6)c - (m_1 + m_2)s \approx 0$	1 1 0 0 1 1
7	1 3 4 5	$p_7 = (m_4 + m_5)c - (m_1 + m_3)s \approx 0$	1 0 1 1 1 0
8	1 3 4 6	$p_8 = (m_6 - m_3)c + (m_1 + m_4)s \approx 0$	1 0 1 1 0 1
9	1 3 5 6	$p_9 = (m_1 + m_6)c - (m_3 + m_5)s \approx 0$	1 0 1 0 1 1
10	1 4 5 6	$p_{10} = (m_5 - m_1)c + (m_4 - m_6)s \approx 0$	1 0 0 1 1 1
11	2 3 4 5	$p_{11} = (m_5 - m_3)c + (m_4 - m_2)s \approx 0$	0 1 1 1 1 0
12	2 3 4 6	$p_{12} = (m_6 + m_4)c + (m_2 - m_3)s \approx 0$	0 1 1 1 0 1
13	2 3 5 6	$p_{13} = (m_2 - m_5)c + (m_3 + m_6)s \approx 0$	0 1 1 0 1 1
14	2 4 5 6	$p_{14} = (m_2 + m_6)c - (m_4 - m_5)s \approx 0$	0 1 0 1 1 1
15	3 4 5 6	$p_{15} = (m_3 - m_4)c + (m_5 + m_6)s \approx 0$	0 0 1 1 1 1

패리티 공간 기법(PSA: Parity Space Approach)

PSA는 고장검출 및 분리를 위하여 가장 많이 사용되는 방법 중의 하나이며, 여분의 센서를 사용하는 시스템에서는 대부분의 문헌에서 사용되고 있다. 고장 감지를 위해서는 시스템의 상태변수에는 독립적이면서 고장요인에만 영향을 받는 변수가 필요하다. PSA는 이러한 변수로써 패리티 벡터(Parity Vector)를 사용한다. 패리티 벡터를 구하기 위하여 행렬 V 를 계산하는 작업이 필요하다. 행렬 V 는 다음과 같은 세 가지 가정을 만족해야 한다.

- ① $VH = 0$: (H 의 null space이다.)
- ② $VV^T = I_{l-n}$: (V 의 각 $l-n$ 개의 row v_i 가 orthonormal 하다.)
- ③ 행렬 V 는 Upper(Downer) Trapezoidal 이면서 Positive Definite이다.

여기서 l 은 센서의 개수이며, n 은 추정하고자 하는 변수의 차원이다.

$\text{rank}(H)=n$ 이므로, $v_i H = 0$ 을 만족하는 $1 \times l$ 의 v_i 가 $l-n$ 개 존재한다. 이 n 개의 v_i 를 이

용하여 V 를 나타내면 다음과 같다.

$$V = [v_1^T \ v_2^T \ \dots \ v_{l-n}^T]^T = [v_{c1} \ v_{c2} \ \dots \ v_l] \quad (V \in R^{(l-n) \times l}) \quad (3.1.177)$$

이러한 행렬 V 의 열공간(column space)을 행렬 H 의 패리티 공간(Parity Space)라고 정의 하며, 다음과 같은 벡터 p 를 패리티 벡터라고 정의한다.

$$p = Vm = VHx + V\epsilon + Vf = V\epsilon + Vf \quad (3.1.178)$$

여기서,

$$m(t) = Hx(t) + \epsilon + f \quad (3.1.179)$$

$\epsilon = [\epsilon_1 \ \epsilon_2 \ \dots \ \epsilon_l]^T$: 센서의 측정 잡음 ($\epsilon_i \sim (0, 1)$), $E[\epsilon\epsilon^T] = I_{l \times l}$)

$f = [f_1 \ f_2 \ \dots \ f_l]^T$: 고장 신호 벡터 ($f \in R^l$)

결국, 패리티 벡터는 V 를 통한 패리티 공간 위로의 '센서 측정치 m 의 투영(Projection)'을 뜻한다. 위 식에서 볼 수 있듯이 $VH=0$ 이라는 V 의 성질 때문에 패리티 벡터는 상태 변수 x 에 독립적이면서 측정 잡음과 고장 신호 성분에 의해서만 영향을 받게 되었다. 이제 패리티 벡터는 고장을 감지하는 변수로 사용할 수 있다.

고장이 발생하지 않는다면, $f=0$ 이고, $E[p]=0$ 이다. 그러나 만약 i 번째 센서에서 고장이 발생하면, $f=f_i$ 가 된다.

또한, 각각의 측정치 m_i 에 대응되는 v_{ci} 가 지시하는 방향을 고장 방향(Fault Direction)이라고 한다. 이것은 i 번째 센서에서 고장이 발생하면 패리티 공간에서 v_{ci} 방향으로 패리티 벡터 p 의 크기가 크게 증가함을 의미한다. 고장 나지 않은 경우는 $\|p\|$ 의 크기가 매우 작으나, 고장이 발생할 경우 고장 방향을 따라 p 의 크기가 늘어난다. 이러한 고장 방향을 통하여 센서의 고장 유무를 판단할 수 있다.

최적 패리티 기법(OPT: Optimal Parity Test)

기존의 FDI 기법들이 측정 행렬의 영공간 벡터들 중에서 서로 직교하는 벡터들을 패리티 벡터로 이용한 반면에 최적 패리티 기법은 n 개의 측정 센서들에 대해서 n 개의 패리티 식을 구하는 과정에서 각각의 패리티 벡터들을 구해야 한다. 이러한 기법은 영공간 벡터들 중에서 I 번째 센서 측정 신호는 I 번째 패리티 식에 최대도로 반영되도록 하고, 다른 패리티 식에는 최소한으로 반영되도록 하는 패리티 식을 구하는 것이다. 이렇게 구한 패리티 식 중에서 가장 큰 패리티 식을 문턱 값과 비교하여 고장 검출과 분리를 수행한다. 이 기법은 고장 검출과 분리를 동시에 수행한다는 특징이 있다.

직렬 감시 기법(ILS: In-Line Monitoring)

타 채널에 대한 정보가 없이 자신의 채널에서 나온 신호를 이용해서 미리 설정된 기준값을 가지고 고장을 감지하는 방법이다. 이 기법은 각 구성요소별 자기진단(BIT, Built-In-Test)과 교차채널감시기법에서 두 채널이 상이한 출력을 내어 비교가 어려울 때 (Tie-Break) 기준신호로 사용한다.

6. 필터 설계 기법 연구

자이로, 가속도계를 포함해서 항공기에는 많은 센서가 쓰인다. 이 센서들은 항공기를 제어하는데 있어 중요한 역할을 하고 있으며 센서들의 고장은 항공기의 안전에 큰 영향을 끼친다. 그런 이유로 고장에 대처하기 위해 모든 센서들은 다중화 되어 있으며 고장 시에는 고장 검출과 분리과정을 거치게 된다.

고장의 검출과 고장센서의 분리는 센서 자체의 잡음 특성들 때문에 상당한 주의를 요한다. 고장으로 인한 신호와 센서 잡음은 사실상 같은 신호적 특성을 띠기 때문에 잡음의 크기가 무시할 수 있을 정도로 작지 않을 경우 고장과 잡음은 구분하기가 힘들다. 따라서 고장을 판별하기 이전에 센서 자체의 잡음을 걸러서 영향을 줄여 주는 보정 과정이 선행되어야 한다. 센서 잡음의 보정을 위해 필터 시스템이 사용된다. 항공기의 제어 시스템에 사용되는 센서들의 노이즈 특성들을 연구하고 그에 맞는 필터 시스템을 구성하는 일은 항공기의 안전을 위해서 필수적이다. 본 연구실의 국가지정 연구인 통합비행제어 시스템 설계에서 항공기의 안전을 위한 센서 고장 검출 및 분리에는, 센서들의 노이즈들을 보정해주는 필터를 연구하는 일이 한 부분을 이루게 된다. 본 연구에서는 이미 연구된 전기 신호에 대한 필터들 중 칼만 필터를 고장 검출 분리를 위해 사용한다.

칼만 필터는 1960년대에 R. E. Kalman에 의해서 발표된 필터이론으로 백색잡음을 가지는 측정 노이즈에 대해 특히 뛰어난 성능을 보인다. 자이로 등의 센서가 가지는 오차원인은 백색잡음, 바이어스, 드리프트의 세 가지로 나눌 수 있다. 백색잡음은 평균값이 0인 잡음으로 일정한 크기의 편차를 가지는 난수이다. 백색잡음의 영향으로 센서에 대한 입력이 고정되어 있을 때 출력 값은 일정한 값에 고정된 채로 특정 크기의 표준편차를 가지는 잡음이 실제 값에 중첩되어 나타난다. 한편, 바이어스는 센서에 대한 입력이 없을 때에도 출력 값이 일정하게 편향되어 나타나는 특성이다. 바이어스 오차는 온도에 따라 변하게 되는데 그에 상응하는 전압을 가해줌으로써 보정할 수 있다. 드리프트는 센서를 장기간 사용할 경우 나타나는 오차로 시간이 지날수록 점진적으로 커지는 특성을 가진다. 드리프트는 실험을 통해 특성을 파악한 후 바이어스와 마찬가지로 보정을 한다. 여기서는 바이어스 오차와 드리프트 오차를 보정했다고 가정하고 백색잡음을 보정하는 방법을 설명한다.

가. Kalman Filter의 유도과정

다음과 같이 표현되는 이산 시스템에 대해서 칼만 필터의 수학적식을 유도한다.

$$\underline{x}_k = \Phi_{k-1} \underline{x}_{k-1} + \underline{w}_{k-1} \quad (3.1.180)$$

여기서 x_k 는 스텝 k에서의 상태변수를 Φ_{k-1} 은 k-1 번째 스텝에서의 상태변환 행렬이며 w_{k-1} 은 공분산 행렬 Q_k 를 갖는 영평균 백색잡음이다. 측정은 상태변수의 선형조합과 백색잡음의 합으로 이루어지므로 그 관계식은 아래와 같다.

$$z_k = H_k \underline{x}_k + \underline{v}_k \quad (3.1.181)$$

여기서 z_k 는 측정된 값들의 벡터이고 H_k 는 관측행렬, v_k 는 공분산 행렬 R_k 를 갖는 백색잡음이다. 이 두식에 의해서 센서로 측정하는 대상시스템의 동역학과 관측변수가 정해진다. 측정된 z_k 를 기반으로 최소자승 방법을 통해 상태변수 \hat{x}_k^- 를 계산할 수 있고 필터를 통해서 이 상태변수를 \hat{x}_k^+ 로 갱신해준다. 기억장소의 거대화를 막기 위해서 회귀적 방법을 사용하는데 시변 가중 행렬 K'_k, K_k 를 이용해서 다음과 같은 선형식을 구성할 수 있다.

$$\hat{x}_k^+ = K'_k \hat{x}_k^- + K_k z_k \quad (3.1.182)$$

이 선형식에서 실제 상태변수 x_k 와 갱신된 추정치 \hat{x}_k^+ 와의 오차가 최소가 되게 하는 행렬 K'_k, K_k 를 구함으로써 칼만 필터를 설계하게 된다.

추정치 \hat{x}_k^- 와 최종 추정치 \hat{x}_k^+ 의 오차를 각각 $\tilde{x}_k^-, \tilde{x}_k^+$ 로 설정하면 다음과 같은 식이 유도된다.

$$\begin{aligned} \hat{x}_k^+ &= x_k + \tilde{x}_k^+ \\ \hat{x}_k^- &= x_k + \tilde{x}_k^- \end{aligned} \quad (3.1.183)$$

수식 4를 수식 3에 대입해서 정리하면 다음과 같다.

$$\tilde{x}_k^+ = (K'_k + K_k H_k - I)x_k + K_k \tilde{x}_k^- + K_k v_k \quad (3.1.184)$$

기대치를 나타내는 $E[v_k] = 0, E[\tilde{x}_k^-] = 0$ 의 두 조건을 만족한다는 가정 하에 식 (3.1.184)의 괄호 안 행렬이 0이면 $E[\tilde{x}_k^+] = 0$ 을 만족해서 최종 추정치의 오차는 바이어스가 없게 된다. 따라서 K'_k 은 다음 식을 만족한다.

$$K'_k = I - K_k H_k \quad (3.1.185)$$

그리고 추정기는 다음 식의 형태를 띈다.

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (z_k - H_k \hat{x}_k^-) \quad (3.1.186)$$

식 (3.1.185)을 적용했을 때의 추정오차의 식은 다음과 같다.

$$\tilde{x}_k^+ = (I - K_k H_k) \tilde{x}_k^- + K_k v_k \quad (3.1.187)$$

식 (3.1.186)의 오차 크기를 최소화 하는 행렬 K_k 를 구하기 위해 오차 공분산 행렬을 정의하고 갱신하는 수식을 세운다. 오차 공분산 행렬은 다음과 같이 정의된다.

$$P_k^+ = E(\tilde{x}_k^+ \tilde{x}_k^{+T}) \quad (3.1.188)$$

식 (3.1.187)을 식 (3.1.188)에 대입하면

$$P_k^+ = E\{(I - K_k H_k) \tilde{x}_k^- [\tilde{x}_k^-^T (I - K_k H_k)^T + \underline{v}_k^T K_k^T] + K_k \underline{v}_k [\tilde{x}_k^-^T (I - K_k H_k)^T + \underline{v}_k^T K_k^T]\} \quad (3.1.189)$$

정의에 의해서,

$$E(\tilde{x}_k^- \tilde{x}_k^{-T}) = P_k^- \quad (3.1.190)$$

$$E(\underline{v}_k \underline{v}_k^T) = R_k \quad (3.1.191)$$

또, 측정 잡음이 백색이므로,

$$E(\tilde{x}_k^- \underline{v}_k^T) = E(\underline{v}_k \tilde{x}_k^{-T}) = 0 \quad (3.1.192)$$

식 (3.1.190), (3.1.191), (3.1.192)을 이용해서 식 (3.1.189)을 정리하면 다음 식과 같다.

$$P_k^+ = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \quad (3.1.193)$$

식 (3.1.193)는 공분산의 값을 갱신하는 역할을 한다. 구해진 공분산 행렬은 K_{k+1} 을 구하는데 사용된다. 오차 크기를 최소화하는 최적 K_k 를 구하기 위해서 다음과 같은 성능함수를 설정한다.

$$J_k = E(\tilde{x}_k^+{}^T S \tilde{x}_k^+) \quad (3.1.194)$$

여기서 S는 가중 행렬로서 실제 최적 해를 구하는데 있어서는 영향을 미치지 않으므로 I로 한다. 결국 J_k 는 다음과 같다.

$$J_k = \text{trace}(P_k^+) \quad (3.1.195)$$

이 성능함수를 최소로 하는 최적의 K_k 행렬을 구하기 위해 J_k 를 K_k 로 미분하여 K_k 에 관해서 정리하면 다음과 같다.

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (3.1.196)$$

식 (3.1.196)이 칼만 필터 이득 행렬이다. 이 이득행렬을 식 (14)에 대입하면 갱신된 추정 오차 공분산 행렬을 구할 수 있다.

$$P_k^+ = (I - K_k H_k) P_k^- \quad (3.1.197)$$

마지막으로, 현 스텝, k에서 다음 스텝, k+1로 넘어가기 위해서 다음 식들을 이용한다.

$$\hat{x}_k = \Phi_{k-1} \hat{x}_{k-1} \quad (3.1.198)$$

$$P_k^- = \Phi_{k-1} P_{k-1} \Phi_{k-1}^T + Q_{k-1} \quad (3.1.199)$$

나. 시뮬레이션과 실제 적용

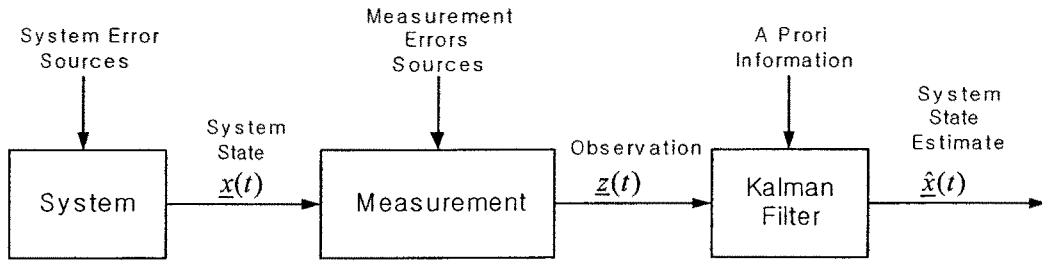


그림 3.1.94 Block Diagram Depicting System, Measurement and Estimator

그림 3.1.94은 시스템의 상태변수가 어떤 식으로 측정과 추정 과정을 거치는지를 보여 준다. 시스템은 자체의 동역학적 특성에 의해서 상태변수 $x(t)$ 를 출력하고 측정 장치는 상태변수들의 선형조합에 측정오차가 중첩된 값을 측정하게 된다. 측정된 값 $z(t)$ 는 필터에 의해서 오차 원인의 영향을 최소화한 상태변수 추정값을 출력한다. 이때 필요한 'a priori' 정보는 센서나 시스템의 오차 원인들에 대한 정보를 포함한다.

앞서 소개한 필터 이론의 하드웨어에의 실제적인 적용에 앞서서 시뮬레이션 과정을 거쳤다. 시뮬레이션 프로그램으로 Matlab의 Simulink를 사용했다. 대상 시스템은 실제 적용 시스템인 1축 모션 테이블을 고려하기 위해서 다음과 같이 설정하였다.

$$\underline{x}_k = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix} \underline{x}_{k-1} + \underline{w}_{k-1} \quad (3.1.200)$$

여기서 x_k 는 시스템의 상태변수로서 각도와 각속도 두 가지를 원소로 가진다. w_{k-1} 은 시스템 오차이다. 관측 방정식은 다음과 같다.

$$z_k = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} x_k + v_k \quad (3.1.201)$$

식에서 알 수 있듯이 두개의 상태변수 중 각도만을 측정하는 관측을 설정한다. 이것은 센서의 다중화를 염두에 둔 것으로 실제 시스템에서는 한 개의 자이로와 한 개의 시뮬레이션 컴퓨터가 내는 출력을 의미한다. 시뮬레이션의 시스템 구성은 다음 그림과 같다.

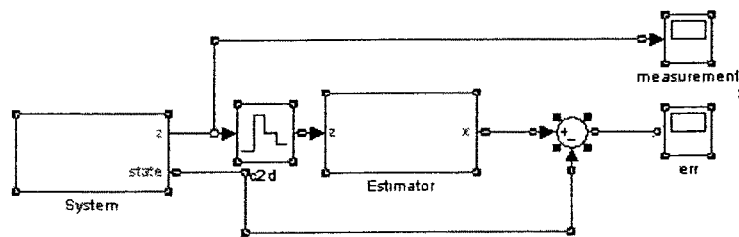
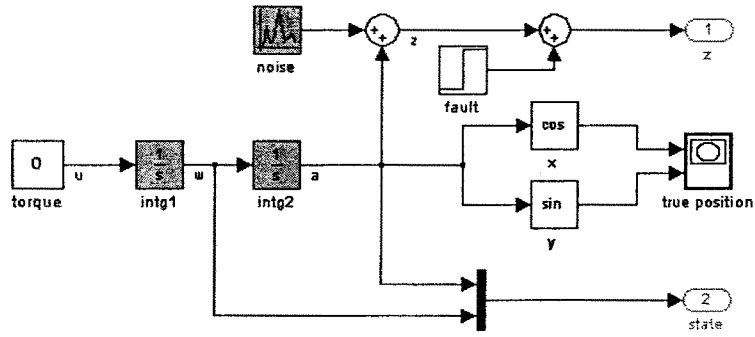


그림 3.1.95 Simulink 상에서의 시스템과 추정기의 관계

System의 내부는 다음 그림과 같이 구성된다. System은 두 개의 출력을 내는데 하나는 잡음이 중첩된 상태변수 값이고 다른 하나는 상태변수만의 값이다. 전자는 Estimator의 입력으로 제공되고 후자는 추정기의 출력과 비교해서 추정기의 성능을 비교하는데 사용된다.



C

그림 3.1.96 System 내부구성

Estimator의 구조는 그림 3.1.97와 같다.

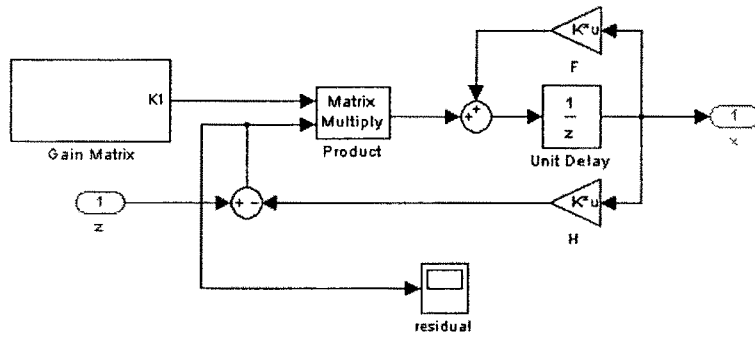


그림 3.1.97 Estimator의 구성

마지막으로 Gain Matrix 내부의 구조는 다음 그림과 같다. 이 Gain matrix는 센서의 잡음 특성을 고려하여 칼만 필터의 이득 행렬을 출력한다.

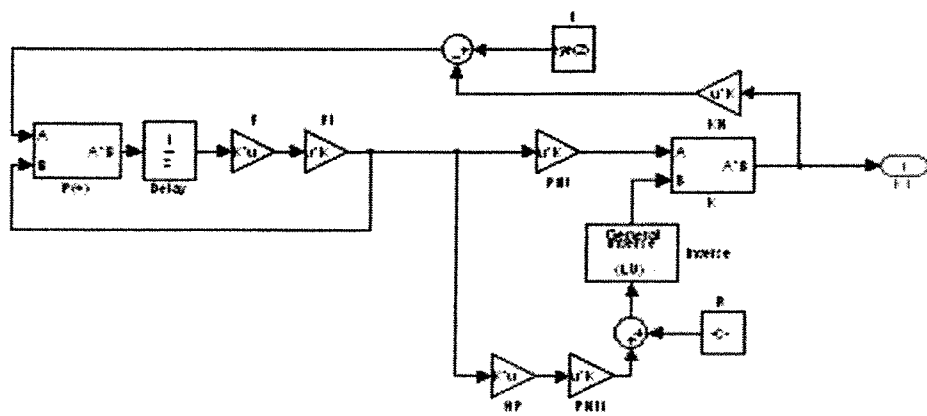


그림 3.1.98 Gain Matrix의 내부구성

이런 식으로 구성된 Simulink의 시뮬레이션 결과는 다음 그림들과 같다. 일정한 각속도로 회전하는 동체의 각도 변화를 측정한다. 두개의 센서를 사용하고 있으며 두 센서 모두

각도를 측정하고 있다. 360초간의 측정값은 그림 6과 같다. 시뮬레이션 상황으로 백색잡음을 섞어 줬기 때문에 그 결과값이 그림처럼 잡음 특성을 나타낸다. 또, 250초가 경과한 후 첫 번째 센서에 고장을 일으켰는데 적색선에서 고장으로 인한 변화가 나타난다. 잡음이 고장에 비해서 무시할 수 있을 정도로 작지 않기 때문에 잡음으로 인한 측정값의 폭이 고장으로 인한 변화폭과 거의 같음을 알 수 있다. 일정한 문턱값(Threshold Value)을 정하고 그 값 일정시간(Confirmation time) 이상 지속될 경우 고장이 선언되게 되는데, 본 시뮬레이션과 같은 상황에서는 문턱값을 너무 작게 잡게 되면 오경보(Fault Alarm)가 자주 발생하게 되므로 문턱값을 잡음의 편차에 비해서 크게 잡을 수밖에 없다. 하지만 문턱값이 상대적으로 커지게 되면 고장이 작을 경우 고장선언이 늦어지게 되는 단점이 있다. 정리하면, 측정 잡음이 고장의 크기에 비해서 크게 되면 고장 진단이 늦어지게 된다는 것이다. 이 측정 잡음의 영향을 줄이기 위해서 필터를 사용하게 되는데 그 결과는 그림 3.1.99에서 보면 알 수 있다.

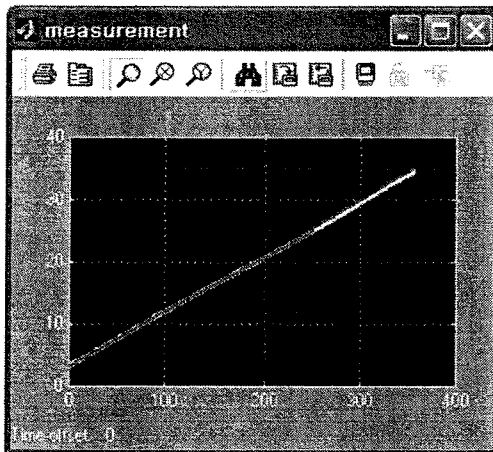


그림 3.1.99 각도 측정치

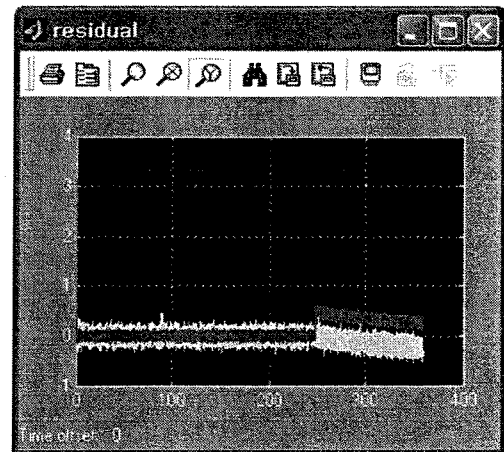


그림 3.1.100 잔차

잔차(Residual)란 측정치와 추정치의 차이이다. 그림 3.6.7은 전 스텝에서 필터를 거친 추정치와 측정치의 차이를 보여준다. 250초 부근에서 고장 난 센서의 반응이 뚜렷하게 나타나는 것을 확인 할 수 있다.

필터 시스템의 활용은 고장 진단의 과정에 필수적으로 선행되어야 한다. 필터의 구성과 역할은 그림과 같다. 관측하고자 하는 시스템에 대해서 센서 세 개를 설치할 경우 각각에 필터를 설치하고 최종적으로 마스터 필터를 통해서 시스템의 상태변수를 추정한다. 그 결과는 고장진단을 하는 동시에 비행제어 컴퓨터에 전송된다. 고장이 발생하면 고장이 비행제어 컴퓨터에 알려지게 되고 비행제어 컴퓨터는 적절한 대응을 하게 된다.

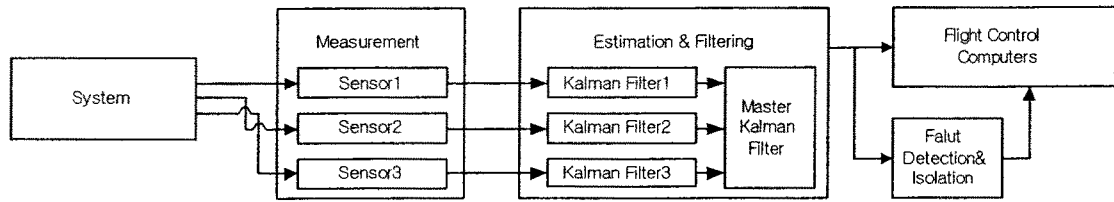


그림 3.1.101 필터 시스템의 구성과 역할

7. 알고리즘의 실시간 계산을 위한 수치기법 개발

이 절에서는 가변 기준모델 추종제어기법을 이용하여 항공기의 고장에 대처할 수 있도록 새로 연구한 재형상 제어시스템의 설계기법을 정리하였다. 항공기의 수학적 모델을 기술하기 위하여 입출력 궤적정보를 이용한 ARX(Auto Regressive eXternal) 모델을 사용하였다. 제안된 ARX 모델은 상태변수로 기술된 모델과 달리 일반적인 최소자승 시스템 식별 알고리즘을 통해 시스템의 매개변수를 직접 식별할 수 있는 장점이 있다. 시스템 식별을 통해 얻어진 입출력 모델을 대상으로 선형 최적제어기법을 이용하여 안정한 제어기를 설계하고, 이로부터 실제 시스템이 추종할 안정한 기준궤적을 생성한다. 본 연구에서 제안된 모델 추종제어기법은 기준모델이 고정되어 변화하지 않는 기존의 모델 추종제어기법과 달리, 기준모델이 실제 시스템의 변화에 따라 수정되는 특징을 갖는다. 이렇게 생성된 기준모델의 궤적을 고장이 발생한 시스템이 추종하도록 하는 추종제어기를 최적제어기법을 이용하여 설계하였다. 또한, 측정잡음을 제거하여 시스템의 추종성능을 향상시킬 수 있도록 제안한 재형상제어기에 추정기를 통합하였다. 제안된 재형상 제어시스템의 성능을 검증하기 위하여 고성능 항공기인 F-16 모델을 대상으로 조종면이 파손된 경우에 대한 수치 시뮬레이션을 수행하였다.

가. 입출력 궤적정보를 이용한 시스템 식별기법

시스템의 수학적 모델을 기술하는 방법으로는 크게 상태공간 모델을 이용한 방법과 입출력 모델을 이용한 방법이 있다. 일반적으로 시스템은 상태공간 모델로 기술한다. 그러나 시스템의 입출력 궤적정보를 이용하여 상태공간 모델로 시스템을 식별하기 위해서는 칼만 필터와 같은 추정기(estimator)를 이용한 추정과정이 필요하다. 반면에, 입출력 모델로 기술된 시스템의 경우에는 실험정보로부터 직접 시스템 매개변수를 식별해 낼 수 있는 장점이 있다. 특히, 실시간으로 고장이 발생한 시스템을 식별하고 제어기를 재설계해야 하는 재형상 제어시스템의 경우에는 빠르고 정확한 시스템 식별과정이 필수적이므로, 입출력 모델을 이용한 시스템 모델이 더 적절하다고 할 수 있다. 이 절에서는 재형상 제어시스템의 설계를 위해 사용되는 입출력 궤적정보로 기술되는 시스템 모델과 시스템 식별과정을 살펴본다.

1. 입출력 궤적정보를 이용한 시스템 모델

상태공간 모델로 기술된 다음과 같은 이산화 시스템을 고려하자.

$$x_{k+1} = Ax_k + Bu_k \quad (3.1.202)$$

$$y_k = Cx_k \quad (3.1.203)$$

여기서 상태변수 $x_k \in R^n$, 제어변수 $u_k \in R^m$, 출력변수 $y_k \in R^p$ 이며, A, B, C 는 이산화된 시스템 행렬이다. 식 (3.1.202)에서 A 행렬은 상태천이행렬이므로 항상 역행렬이 존재한다. 식 (3.1.202)과 식 (3.1.203)에 주어진 상태공간 모델로부터 다음과 같은 관계식이 성립한다.

$$y_{k-j} = CA^{-j}x_k - \sum_{i=0}^{j-1} CA^{-j+i}Bu_{k-i-j}, \quad (j=0, \dots, q-1) \quad (3.1.204)$$

위 식을 행렬식으로 표현하면 다음과 같이 쓸 수 있다.

$$Y_k = Hx_k + PW_k \quad (3.1.205)$$

여기서

$$Y_k = \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-q+1} \end{bmatrix}, \quad H = \begin{bmatrix} C \\ CA^{-1} \\ \vdots \\ CA^{-q+1} \end{bmatrix}, \quad W_k = \begin{bmatrix} u_{k-1} \\ u_{k-2} \\ \vdots \\ u_{k-q+1} \end{bmatrix}$$

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -CA^{-1}B & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ -CA^{-q+1}B - CA^{-q+2}B & & & -CA^{-1}B \end{bmatrix}$$

식 (3.1.205)로부터 다음과 같은 관계식이 성립한다.

$$Y_{k+1} = Hx_{k+1} + PW_{k+1} \quad (3.1.206)$$

식 (3.1.202)을 식 (3.1.206)에 대입하여 정리하면 다음과 같다.

$$Y_{k+1} = Hx(Ax_k + Bu_k) + PW_{k+1} \quad (3.1.207)$$

한편, 식 (3.1.208)에서 상태변수 x_k 는 다음과 같이 쓸 수 있다.

$$x_k = H^+(Y_k - PW_k) \quad (3.1.208)$$

여기서 H^+ 는 행렬 H 의 의사역행렬(pseudo inverse)이다. 행렬 H 의 랭크(rank)가 시스템 차수 n 보다 작은 경우에는 주어진 입출력 신호 Y_k 와 W_k 에 대하여 위 식을 만족하는

해가 무수히 존재하므로, 식 (3.7.7)에 의해 얻어진 상태변수는 현재 상태의 상태변수와 동일하지 않을 수 있다. 즉, 식 (3.7.7)에 의해 표현되는 시스템의 방정식은 실제 시스템의 동특성을 제대로 표현하지 못 할 수도 있다. 만약, 행렬 H 의 랭크가 n 보다 크면 임의의 Y_k 나 W_k 에 의해 만족하는 x_k 는 없으나, 동일한 시스템에 대해 Y_k 와 W_k 가 얻어지는 경우에는 유일해 x_k 가 결정된다.

행렬 W_k 의 정의로부터 W_{k+1} 는 다음과 같이 쓸 수 있다.

$$W_{k+1} = \begin{bmatrix} u_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ I_{m \times m} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & I_{m \times m} & 0 \end{bmatrix} = I_m u_k + J_m W_k \quad (3.1.209)$$

여기서 $I_m = [I_{m \times m} \ 0 \ \cdots \ 0]^T$, $I_{m \times m}$ 은 $m \times m$ 단위행렬(Identity Matrix)을 의미하며, J_m 은 다음과 같이 정의된다.

$$J_m = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I_{m \times m} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & I_{m \times m} & 0 \end{bmatrix}$$

식 (3.1.209)과 행렬 H 와 P 의 정의로부터 다음과 같은 관계식을 유도할 수 있다.

$$\begin{aligned} HBu_k + PW_{k+1} &= HBu_k + P(I_m u_k + J_m W_k) \\ &= (HB + \Pi_m)u_k + PJ_m W_k \end{aligned} \quad (3.1.210)$$

$$= \begin{bmatrix} CBu_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} + PJ_m W_k = I_p CBu_k + PJ_m W_k$$

$$Y_{k+1} = HAH^+(Y_k - PW_k) + I_p CBu_k + PJ_m W_k \quad (3.1.211)$$

여기서 $I_p = [I_{p \times p} \ 0 \ \cdots \ 0]^T$ 이고, $I_{p \times p}$ 은 $p \times p$ 단위행렬이다. 식 (3.1.208)과 식 (3.1.210)를 식 (3.1.207)에 대입하여 정리하면 다음 식을 얻을 수 있다.

한편, 행렬 P 는 정의로부터 다음과 같은 관계식을 만족한다.

$$J_p P = PJ_m \quad (3.1.212)$$

여기서

$$J_p = \begin{bmatrix} 0 & 0 & 0 & 0 \\ I_{p \times p} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & I_{p \times p} & 0 \end{bmatrix}$$

식 (3.1.212)을 식 (3.1.211)에 대입하여 정리하면 다음과 같이 쓸 수 있다.

$$Y_{k+1} = HAH^+ Y_k + [I_p CBJ_p P - HAH^+ P] \begin{bmatrix} u_k \\ W_k \end{bmatrix} \quad (3.1.213)$$

$$HAH^{-1} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n \\ I_{p \times p} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & I_{p \times p} & 0 \end{bmatrix} 0 = J_p + I_p \alpha \quad (3.1.214)$$

위 식을 살펴보면 $(k+1)$ 에서의 출력은 그 이전의 입력과 출력으로부터 결정되는 형태임을 알 수 있다. 만약, 행렬 H 의 크기가 $n \times n$ 이고, 랭크가 n 이면 $H^+ = H^{-1}$ 이 성립하고, HAH^+ 는 다음과 같은 구조를 가진다. 여기서 $\alpha = [\alpha_1 \cdots \alpha_n]$ 이고, $\alpha_i \in R^{p \times p}$ 이다. 식 (3.1.214)을 식 (3.1.213)에 대입하여 정리하면 다음 식을 얻을 수 있다.

$$Y_{k+1} = (J_p + I_p \alpha) Y_k + I_p [CB - \alpha P] \begin{bmatrix} u_k \\ W_k \end{bmatrix} \quad (3.1.215)$$

위 식을 행렬식으로 다시 쓰면 다음과 같은 ARX 형태의 모델을 얻을 수 있다.

$$Y_{k+1} = I_p [\alpha CB - \alpha P] \begin{bmatrix} Y_k \\ u_k \\ W_k \end{bmatrix} = \theta \phi(k) \quad (3.1.216)$$

위 식에서 입출력 레적정보를 이용해서 식별된 시스템 매개변수는 α , CB , αP 값이 된다.

2. 회기적 시스템 식별 알고리즘

이 절에서는 식 (3.1.216)에 주어진 ARX 모델의 매개변수를 식별하는 알고리즘을 살펴본다. 식 (3.1.216)에서 시스템 매개변수 α , CB , αP 는 각각 $p \times pq$, $p \times m$, $p \times (q-1)m$ 의 차수를 가지고 있으므로, 식별할 전체 매개변수의 개수는 $p \times (pq + qm)$ 개가 된다.

매개변수들을 식별하기 위하여 s 개의 입출력 레적정보를 고려하자. 감지기를 통해 얻은

입출력 제적정보를 이용하여 다음과 같은 형태의 행렬식을 구성한다.

$$\begin{aligned}
 Y &= [y_{k+1} \ y_{k+2} \ y_{k+3} \ \dots \ y_{k+s}] \\
 &= \theta [\phi_k \ \phi_{k+1} \ \dots \ \phi_{k+s-1}] \\
 &= \theta \Phi
 \end{aligned}
 \tag{3.1.217}$$

위 식은 $p \times s$ 개의 구속 방정식과 $p \times (p+m)q$ 개의 매개변수로 구성된 방정식으로, 신빙성 있는 시스템 식별 결과를 얻기 위해서는 s 가 $(p+m)q$ 보다 큰 값을 가져야 한다. 한편, 시스템 매개변수가 시간에 따라 변화하면, 현재의 추정 매개변수와 실제매개 변수값 사이에 오차가 생기게 된다. 따라서 $(k-1)$ 에서의 매개변수 값을 $\theta(k-1)$ 이라 하고, k 에서의 매개변수 값을 $\theta(k)$ 라고 하면 다음과 같은 관계식이 성립한다.

$$Y - \theta(k-1)\Phi = \theta(k)\Phi - \theta(k-1)\Phi = \delta\theta\Phi \tag{3.1.218}$$

여기서 $\delta\theta = \theta(k) - \theta(k-1)$ 이다. $\delta\theta$ 를 최소자승법을 이용하여 구하면 다음과 같이 주어진다.

$$\delta\theta = [Y - \theta(k-1)\Phi]\Phi^T[\Phi\Phi^T]^{-1} \tag{3.1.219}$$

이상에서 식별할 매개변수는 다음과 같은 회귀 방정식(recursive equation)을 통하여 얻을 수 있다.

$$\begin{aligned}
 \theta(k) &= \theta(k-1) + k\delta\theta \\
 &= \theta(k-1) + k[Y - \theta(k-1)\Phi]\Phi^T[\Phi\Phi^T]^{-1}
 \end{aligned}
 \tag{3.1.220}$$

나. 모델 추종제어기법을 이용한 재형상 비행제어

이 장에서는 앞 절에서 유도한 시스템 인식기법을 기반으로 새로이 연구한 고장이 발생한 항공기의 재형상 제어시스템 설계기법을 정리한다. 조종면의 일부가 파손된 항공기를 대상으로 고장 검출기를 설계하여 고장을 검출하고, 가변 기준모델 추종제어기법을 이용하여 제어기를 재설계한다. 재형상 제어시스템은 입출력 모델로 기술된 기준 항공기 모델과 이를 안정하게 하는 되먹임 제어기로 구성된 기준모델, 그리고 기준모델의 기준궤적을 추종하기 위한 추종제어기로 구성된다. 본 연구에서 제안한 재형상 제어시스템은 가변 기준모델 추종제어기법을 이용하여 항공기의 고장에 대처할 수 있는 재형상 제어시스템이다. 제안된 재형상 제어기의 구성도를 Fig. 3.1.102에 보였다. 새로이 제안한 재형상 제어기의 특징은 시스템의 상태를 기준궤적을 생성하는 기준모델을 구성할 때 시스템 인식기법에 의해 지속적으로 시스템의 변화를 반영한다는 데 있다. 이러한 특성으로 인해 실제 고장난 시스템이 추종하기 매우 어려운 기준궤적을 생성할 우려가 있는 기존의 모델추종제어기법과 달리, 본 연구에서 제안된 가변 기준모델 추종제어기법은 시스템의 고장 등과 같은 상황을

고려할 수 있다는 장점을 갖는다.

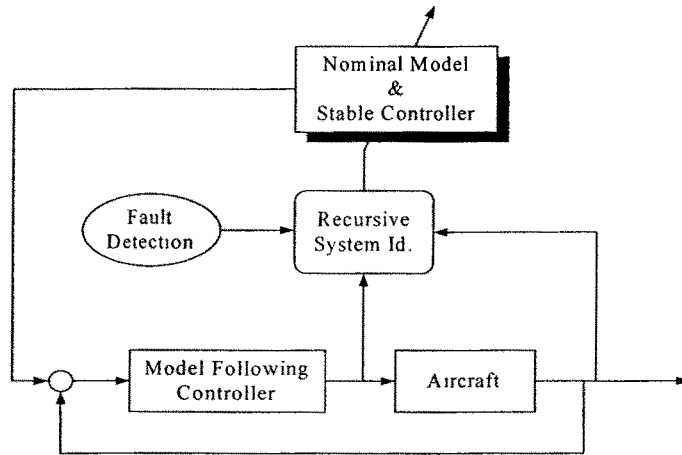


그림 3.1.102 모델추종기법을 이용한 재형상 제어기

고장이 발생한 항공기가 추종할 안정한 기준궤적을 우선적으로 생성해야 한다. 식 (3.1.217)과 식 (3.1.215)에 주어진 시스템을 행렬식으로 재구성하면 다음과 같이 쓸 수 있다.

$$\begin{bmatrix} Y_{k+1} \\ W_{k+1} \end{bmatrix} = \begin{bmatrix} J_p + I_p \alpha & -I_p \alpha P \\ 0 & J_m \end{bmatrix} \begin{bmatrix} Y_k \\ W_k \end{bmatrix} + \begin{bmatrix} I_p CB \\ I_m \end{bmatrix} u_k \quad (3.1.221)$$

식 (3.1.221)를 살펴보면 상태공간 모델로 기술된 이산 선형시스템과 동일한 형태임을 알 수 있다. 따라서 식 (3.1.221)를 안정하게 하는 되먹임 제어입력 u_k 는 선형 최적제어기법으로부터 다음과 같이 구할 수 있다.

$$u_k^* = -K_1 Y_k - K_2 W_k = -K \begin{bmatrix} Y_k \\ W_k \end{bmatrix} \quad (3.1.222)$$

여기서 제어이득 K 는 선형 최적제어기법으로 설계한 제어이다. 식 (3.1.222)를 식 (3.1.214)에 대입하여 정리하면, 항공기가 추종할 기준궤적 y_k^* 는 다음과 같이 주어진다.

$$y_{k+1}^* = \alpha Y_k - \alpha P W_k + C B u_k^* \quad (3.1.223)$$

회귀적 방법을 이용하여 실시간으로 시스템 식별을 하는 방식은 제한된 가진으로 인하여 오프라인에서 시스템 식별을 수행하는 것보다 정확하지 못하다. 따라서 본 연구에서는 시스템의 모델링 오차에 강인한 재형상 제어를 설계하기 위하여 모델추종 제어기법을 도입한다.

$$\begin{aligned} y_{k+1}^* &= y_{k+1} \\ &= \alpha Y_k - \alpha P U_k + C B u_k \end{aligned} \quad (3.1.224)$$

시스템이 추종할 기준궤적을 y_{k+1}^* 이라고 하면, 완벽한 모델추종 조건은 다음과 같이 쓸 수 있다.

식 (3.1.224)에서 주어진 기준궤적을 추종하는 제어입력은 다음과 같이 주어진다.

$$u_k = (CB)^+(y_{k+1}^* - \alpha Y_k + \alpha P U_k) \quad (3.1.225)$$

위 식에서 추종제어기의 존재 여부는 CB 행렬에 의해 결정됨을 알 수 있다. 만약, 출력의 개수가 입력의 개수보다 크면 최소자승 개념의 근사해 형태로 추종제어 입력을 설계할 수 있다. 반면에 출력의 수가 입력의 수보다 클 경우에는 시스템이 원하는 모델의 출력을 따라가는 다양한 입력 해를 구할 수 있게 된다. 이러한 경우에는 다음과 같은 성능지수를 최소로 하는 제어입력을 설계할 수 있다.

$$J = [y_{k+1}^* - y_{k+1}]^T Q [y_{k+1}^* - y_{k+1}] + u_k^T R u_k \quad (3.1.226)$$

여기서 Q 와 R 은 각각 상태와 입력에 가중치를 부여하는 양(positive definite)의 행렬이다. 식 (3.1.226)에서 첫 번째 항은 모델추종 성능과 관련된 항이고, 두 번째 항은 입력에너지 최소화 관련된 항이다. 식 (3.1.226)에 주어진 성능지수를 최소로 하는 제어입력은 최적제어이론으로부터 다음과 같이 주어진다.

$$u_k = G[y_{k+1}^* - \alpha Y_k + \alpha P U_k] \quad (3.1.227)$$

여기서 $G = [R + B^T C^T Q C B]^{-1} B^T C^T Q$ 이다. 위 식에서 α , αP , BC 값은 시스템 식별을 통해 변화하게 되므로, 제어기의 제어이득 또한 시스템의 변화된 상태에 따라 바뀌게 된다.

다. 추정기를 가진 모델추종 재형상 제어시스템의 설계

일반적으로 출력신호는 측정잡음을 포함하고 있다. 이 절에서는 감지기의 측정잡음을 최소화하기 위하여 추정기(estimator)를 사용해 출력신호를 획득하는 과정을 설명한다. 제어시스템의 구성은 그림 3.1.103와 같다. 식 (3.1.215)로 주어진 시스템 방정식의 추정기는 다음과 같이 주어진다.

$$\hat{Y}_{k+1} = (J_p + I_p \alpha) \hat{Y}_k - I_p \alpha P U_k + I_p C B u_k + L(Y_k - \hat{Y}_k) \quad (3.1.228)$$

추정오차 $E_k = Y_k - \hat{Y}_k$ 에 대한 추정오차 방정식은 다음과 같이 쓸 수 있다.

$$E_{k+1} = (J_p + I_p \alpha - L) E_k \quad (3.1.229)$$

시스템 매개변수 α 의 초기값을 α_0 라고 하자. 시스템 행렬 $(J_p + I_p \alpha_0)$ 를 안정화시킬 수 있는 추정기 이득을 L_0 라고 하면, 시스템 식별과정을 통하여 새로 획득한 매개변수 α 에 대해 $(J_p + I_p \alpha)$ 를 안정화시킬 수 있는 추정기 이득은 다음과 같이 설계된다.

$$L = I_p \alpha - I_p \alpha_0 + L_0 \quad (3.1.230)$$

위의 과정은 추정기 이득을 구하는 매우 효율적인 방법이다. 추정기 이득을 식(3.1.228)에 대입하여 얻은 \hat{Y}_k 를 이용하여 구하면 출력 추정치는 다음과 같이 구할 수 있다.

$$\bar{y}_{k+1} = \alpha \hat{Y}_k - \alpha P U_k + C B u_k \quad (3.1.231)$$

이제 기준출력 Y_{k+1}^* 을 추종하는 입력을 만들기 위하여 다음과 같은 성능지수를 고려한다.

$$J = [y_{k+1}^* - \bar{y}_{k+1}]^T Q [y_{k+1}^* - \bar{y}_{k+1}] + u_k^T R u_k \quad (3.1.232)$$

위 식과 같이 주어진 목적함수를 최소화하는 제어입력은 다음과 같이 구할 수 있다.

$$u_k = G [y_{k+1}^* - \alpha \hat{Y}_k + \alpha P U_k] \quad (3.1.233)$$

여기서 $G = [R + B^T C^T Q C B]^{-1} B^T C^T Q$ 이다.

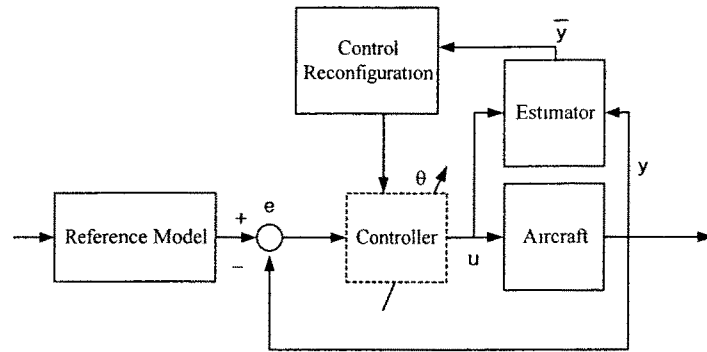


그림 3.1.103 추정기를 가진 모델추종 재형상 제어기

라. 수치 시뮬레이션

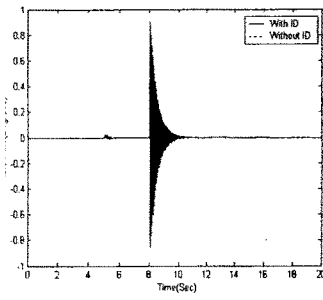
F-16 항공기의 더치롤(dutch roll) 기동

이 절에서는 조종면이 파손된 F-16 항공기의 더치롤 기동을 대상으로 이 절장에서 제안된 재형상 제어기법의 성능을 평가한다. 수치 시뮬레이션을 위해 500 [ft/s]의 속도로 1.43°의 받음각으로 순항하는 비행상태를 고려한다. 수치 시뮬레이션 시작 후 5초 후에 항공기의 러더와 에일러론에 각각 30%와 50%의 면적손실이 생기고, 항공기는 8초 후에 러더를 이용해 더치롤 기동을 시작한다고 가정한다. 조종면이 파손된 경우의 시스템 행렬을 A_f, B_f, C_f 라 하고, 정상적인 항공기의 시스템 행렬을 A_n, B_n, C_n 이라 하자. 정상적인 시스템에 대한 제어기의 제어이득을 G_n 이라고 하고 재형상 과정을 통해 새롭게 설계된 제어기의 제어이득을 G_f 라 하자. 조종면이 파손된 항공기에 각 제어기를 적용하였을 때의 폐회로 시스템 고유치를 표 3.7.1에 보였다. 폐회로 시스템의 고유치로부터 알 수 있듯이 재형상 과정을 거친 제어기를 사용할 경우에 항공기의 안정성이 높아지게 된다.

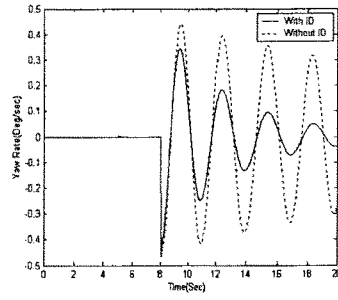
제안한 재형상 제어시스템을 고장난 항공기에 적용한 수치 시뮬레이션 결과를 그림 3.1.104에 보였다. 그림에서 실선은 재형상 제어시스템을 적용한 시스템의 반응을 나타내고, 점선은 초기의 기본 제어시스템을 적용한 반응을 의미한다. 그림에서 볼 수 있듯이 F-16 항공기의 러더와 에일러론이 파손되었을 경우에 재형상 제어시스템이 초기의 기본 제어시스템보다 더치롤 운동을 상당히 빠르게 감소시킴을 알 수 있다. 즉, 시스템 식별과정을 수행하여 파손된 항공기 모델을 파악하고 제어시스템을 재설계하면 항공기의 전반적인 성능이 향상됨을 확인할 수 있다.

표 3.1.14 더치롤 기동의 고유치

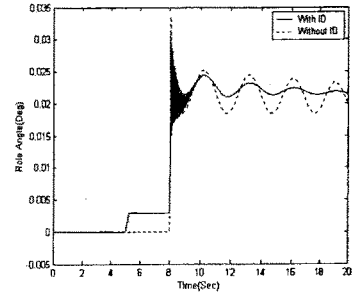
폐회로 시스템 행렬	고유치
$A_f - B_f G_f C_f$	$-0.0428 \pm 2.1191i$
$A_f - B_f G_n C_f$	$-0.0344 \pm 2.1257i$



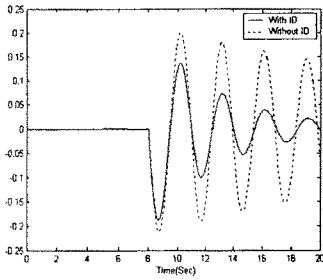
(a) Roll rate



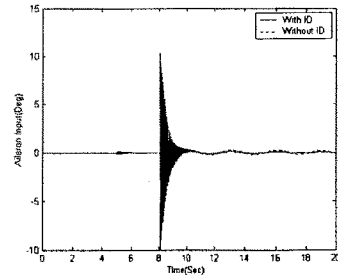
(b) Yaw rate



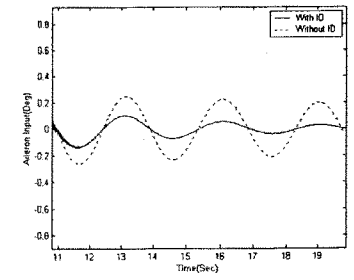
(c) Roll angle



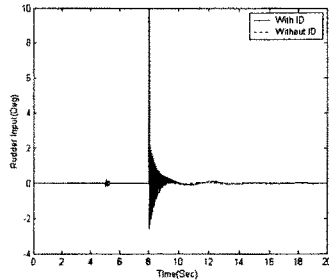
(d) Yaw angle



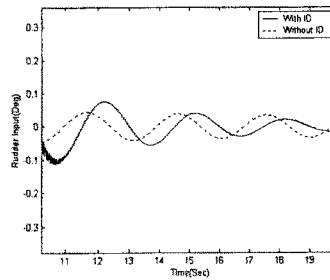
(e) Aileron input



(f) Aileron(close)



(g) Rudder input



(h) rudder(close)

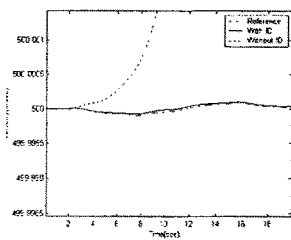
그림 3.1.104 Responses of dutch roll maneuver

F-16 항공기의 롤 기동

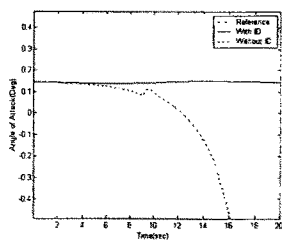
이 절에서는 앞 수치예제와 동일한 비행조건에서 F-16 항공기가 롤 기동을 수행하는 경우를 고려한다. 수치 시뮬레이션 시작 후 2초 후에 항공기의 러더와 에일러론이 파손되어 각각 20%와 10%의 면적이 손실되었으며, 다시 9초 후에 면적손실이 각각 30%과 50%로 확대되었다고 설정한다. 추종해야 할 롤 기동은 다음과 같다.

$$\phi = \begin{pmatrix} 0^\circ & 0 \leq t < 2 \\ 10^\circ & 2 \leq t < 7 \\ 0^\circ & 7 \leq t < 10 \\ -10^\circ & 10 \leq t < 15 \\ 0^\circ & 15 \leq t < 20 \end{pmatrix}$$

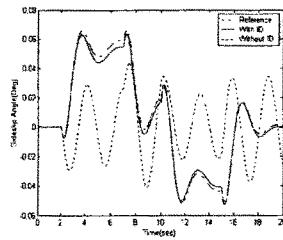
추종해야 할 롤 궤적은 불연속이지만 최적 궤적생성기를 사용하므로 완만한 곡선의 궤적이 생성된다. 수치 시뮬레이션 결과를 그림 3.1.105에 보였다. 그림에서 일점쇄선은 고장이 발생하지 않았을 때의 궤적이고, 점선은 고장이 발생하였으나 초기의 기본 제어시스템을 계속 적용한 결과를 나타내며, 실선은 재형상 제어시스템을 적용한 결과를 의미한다. 그림에서 볼 수 있듯이 재형상 제어시스템을 적용한 경우에는 조종면의 파손에도 불구하고 조종면이 파손되지 않은 항공기의 궤적을 잘 추종하고 있음을 알 수 있다. 반면에 재형상 과정을 거치지 않은 기본 제어시스템의 경우에는 피치축 방향의 기동이 발산하게 됨을 볼 수 있다. 그러나 롤축 방향의 기동은 조종면의 파손에도 불구하고 큰 영향을 받지 않는 것을 알 수 있다.



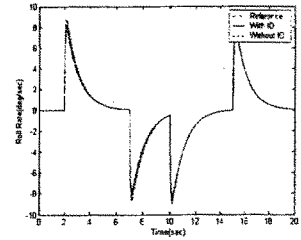
(a) Velocity



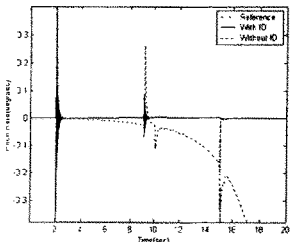
(b) Angle of attack



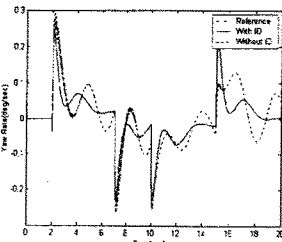
(c) Sideslip angle



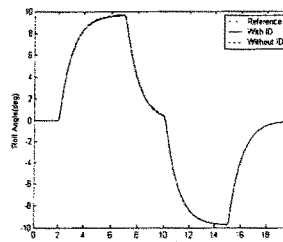
(d) Roll rate



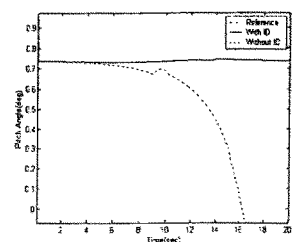
(e) Pitch rate



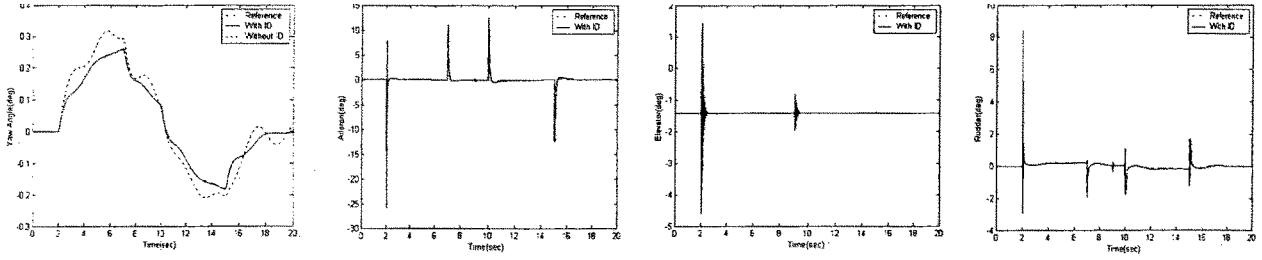
(f) Yaw rate



(g) Roll angle



(h) Pitch angle



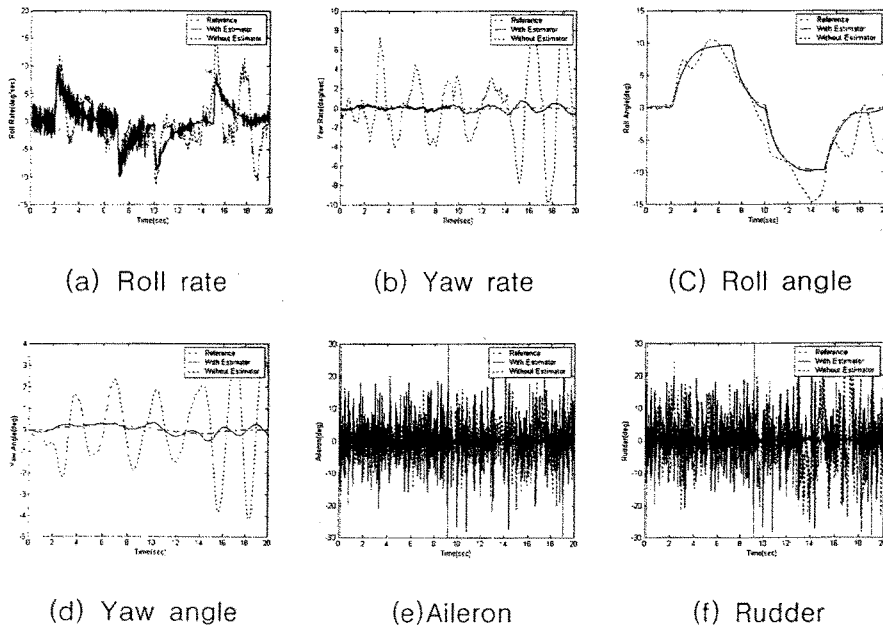
(i) Yaw angle (j) Aileron input (k) Elevator input (l) Rudder input

그림 3.1.105 Responses of roll maneuver

입력잡음을 고려한 재형상 제어시스템

이 절에서는 앞에서 제안한 추정기를 가진 재형상 제어시스템의 성능을 평가한다. 앞의 수치예제와 동일하게 F-16 항공기가 롤 기동을 수행하며, 수치 시뮬레이션 시작 후 2초 후에 항공기의 러더와 에일러론이 파손되어 각각 20%와 10%의 면적이 손실되었으며, 다시 9초 후에 면적손실이 각각 30%과 50%로 확대되었다고 가정한다. 센서 측정값에 평균 0, 최대 오차 0.01의 백색 잡음을 고려하였다.

수치 시뮬레이션 결과를 Fig. 3.1.106에 보였다. 그림에서 점선은 추정기를 사용하지 않은 재형상 제어시스템을 적용한 결과이고, 실선은 추정기를 추가한 재형상 제어시스템의 궤적을 나타낸다. 그림에서 볼 수 있듯이 입력잡음이 있는 경우에는 시스템 식별 오차로 인해 추정기를 적용하지 않으면 항공기의 안정성이 크게 떨어지는 것을 알 수 있다. 특히, 요축 방향의 안정성이 크게 영향을 받는 것을 볼 수 있다.



(a) Roll rate (b) Yaw rate (c) Roll angle
(d) Yaw angle (e) Aileron (f) Rudder

그림 3.1.106 Responses of roll maneuver(estimator included)

마. 연구요약

재형상 비행제어시스템을 적용하면 항공기의 감지거나 구동기, 그리고 동체의 파손이나 고장에 대처할 수 있어 항공기의 생존성을 향상시킬 수 있다. 본 연구에서는 실시간 시스템 식별이 용이한 입출력 궤적정보를 이용하여 항공기의 수학적 모델을 유도하고, 기존의 시스템 식별 알고리즘을 통해 시스템의 매개변수를 식별할 수 있음을 보였다. 항공기의 고장 상태를 고려하여 기준궤적을 생성하는 기법을 제안하였으며, 재형상 제어시스템 설계를 위해 기준궤적을 추종하는 모델 추종제어기를 설계하였다. 또한, 감지기의 측정잡음을 제거하기 위한 추정기와 이를 이용한 재형상 제어기법을 제안하였다. 본 연구에서 제안된 재형상 제어시스템의 타당성을 검증하기 위해 F-16 항공기를 대상으로 조종면의 파손으로 제어면이 손실된 경우에 대한 수치 시뮬레이션을 수행하였다. 그 결과 F-16 항공기의 다양한 기동에 대해 재형상 제어시스템의 제어성능이 만족스러움을 확인할 수 있었다. 본 연구에서 수행된 연구결과를 요약하면 다음과 같다.

첫째, 궤적정보로 기술되는 시스템 모델을 구성하였으며, 선형 최적제어기법을 이용하여 식별된 시스템을 안정화시키는 제어입력을 설계하였다. 둘째, 항공기의 고장 여부를 파악할 수 있는 고장 검출기를 설계하였으며, 고장이 발생하였을 때 고장을 고려한 기준궤적을 생성하고, 이를 추종하는 새로운 모델 추종 재형상 제어시스템을 설계하였다. 셋째, 제안된 재형상 제어기법을 조종면이 파손된 F-16 항공기에 적용하여 다양한 궤적 및 측정 잡음, 그리고 고장에 대한 제어기의 성능을 검증하였다.

제 2 절 고장허용 비행제어 설계기법 연구

1. 고장허용 비행제어시스템 연구개요

고성능 항공기는 요구되는 복잡하고 다양한 임무수행을 만족스럽게 수행하며, 광범위한 비행환경에서도 견실한 비행성능을 가져야 된다. 이를 위해 최근에는 고성능 항공기에 대해서 다양한 기법을 이용한 비행제어시스템이 설계되고 있다. 한편, 항공기에 발생하는 고장(fault)은 경제적 손실을 초래할 뿐만 아니라, 귀중한 인명의 손상을 가져올 수 있다. 따라서 항공기의 설계단계에서부터 성능 못지않게 안전성을 고려하는 것이 일반적인 현상이 되었다. 특히, 최근에 개발되고 있는 항공기는 성능이 고도화되고 복잡한 구조를 갖게 되어, 시스템이 고장을 일으킬 가능성은 더욱 커졌다. 이에 따라 항공기 안정성에 치명적인 고장이 발생하는 경우에 대해 효과적으로 대처하기 위한 연구의 중요성은 계속 증대되고 있다.

본 연구에서는 항공기의 정상상태뿐만 아니라, 항공기의 이상상태에 대해서도 적절히 대처할 수 있도록 RM(Redundancy Management)시스템과 RCS(Reconfigurable Control System)에 대한 통합 설계기술을 연구하여, 궁극적으로 지능형 비행제어 알고리즘과 결합한 통합 비행제어시스템을 구축하는 것을 연구 목적으로 하고 있다.

가. 연구 범위 및 분류

시스템의 고장이 항공기의 안정성에 영향을 줄 수 있는 주요 시스템은 구동기와 감지기, 그리고 항공기 구조시스템으로 구분할 수 있다. 고장허용제어(Fault Tolerant Control)란 비행체가 정상적으로 작동할 때는 물론, 항공기 시스템에 문제가 발생한 경우에도 비행체의 안정성을 보장하고 만족할 만한 성능을 발휘할 수 있도록 고장에 대처하는 제어기를 구축하는 개념을 의미한다. 이와 같은 고장허용제어 기법은 그림 3.2.1과 같이 구분할 수 있다. 그림 3.2.1에서 보듯이 시스템의 내외부 변화에 대해 성능저하가 일어나지 않는 제어시스템인 강건제어기(Robust Control)는 소극적 의미의 고장허용제어로 분류할 수 있다. 반면, 고장이 발생하였을 때 고장의 특성을 고려하여 제어시스템을 설계하는 재형상제어기(Reconfigurable Control)는 적극적으로 고장에 대처한다고 할 수 있다. 고장에 효율적으로 대응하기 위해서는 고장진단 및 분리(FDI, Fault Detection & Isolation)과정이 요구되는 경우가 많으며, 이러한 전체 프로세스를 관장하는 모듈(Supervision)이 함께 사용되기도 한다.

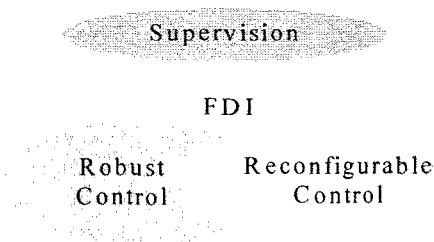


그림 3.2.1 고장허용제어의 영역

고장허용 제어시스템을 구축하기 위해, 대상이 되는 비행체 시스템은 각 핵심부품의 고장

에 대한 분석이 가능해야 하며, 고장모드 및 그 파급효과에 대한 진단을 수행할 수 있어야 한다. 또한, 이러한 분석이 신뢰할 수 있는 통계적 분포를 이루는 선행 조건을 만족해야 한다.

시스템의 고장에 대처할 수 있는 비행제어시스템을 설계하기 위한 방법에는 크게 하드웨어 여분방법과 해석적 여분방법이 있다. 최근 개발되는 항공기의 경우, 대부분 3중 또는 4중의 구동기와 감지기를 장착하여 고장에 대비하고 있다. 그러나 이러한 하드웨어 여분방식은 동일한 부품을 복수로 장착하기 때문에 비용이 많이 들고, 비행제어 알고리즘이 복잡해지는 등의 문제점을 내포하고 있다. 이에 따라 최근에는 항공기의 수학적 모델을 이용하여 고장에 대처하는 해석적 여분에 대한 연구가 활발히 수행되고 있다.

앞에서 설명한 것과 같이 강건제어기처럼 외란에 강건한 제어기를 구축하여 시스템의 고장에 대처하는 방식을 수동적 고장허용 제어 방식이라고 하며, 고장을 검출하고 시스템을 재인식한 후 다양한 제어기법을 적용하여 재형상제어를 거치는 제어기법을 능동적 고장허용 제어기법이라 한다. 재형상(reconfigurable) 비행제어시스템에 대한 연구는 고장검출 및 분리에 기반을 둔 분야와 실시간으로 시스템을 식별하여(system identification) 제어기를 자동으로 재설계하는 분야로 구성되며, 이를 도표화 하여 정리하여 보면 그림 3.2.2와 같다.

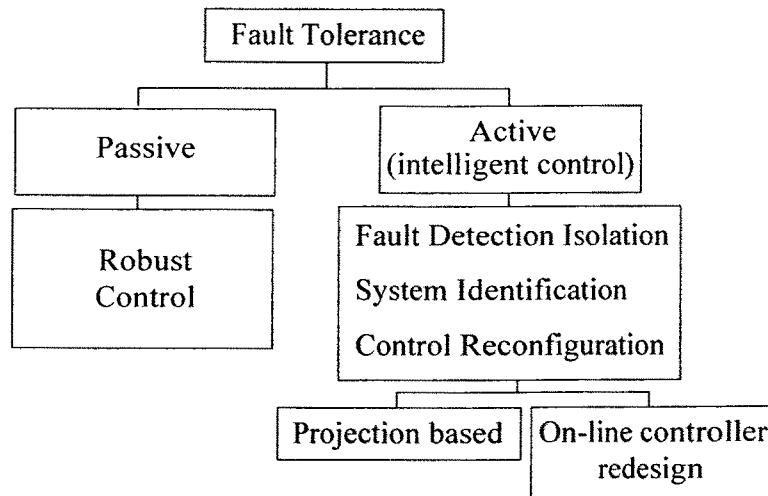


그림 3.2.2 고장허용 제어기법

나. 고장검출 및 분리

고장검출 및 분리(Fault Detection and Isolation)에 기초한 연구는 사전에 발생할 가능성이 있는 다양한 고장 상황에 대해 대처할 수 있는 다양한 제어기를 탑재하고 있다가, 실제 항공기에 고장이 발생하면 고장의 종류를 파악하여 적절한 고장 대처 제어기를 적용한다는 개념에 기초하고 있다. 이 때, 고장의 검출 및 분리는 실제 시스템에 인가된 제어입력과 동일한 제어입력을 명목상의(nominal) 모델에 인가했을 때 발생하는 출력과 실제 시스템의 출력을 비교하여 가장 유사한 반응을 보이는 고장 모델을 선택하는 방식으로 이루어진다.

다. 다중화관리 하드웨어 해법

다중화관리의 하드웨어 해법은 항공기 시스템의 하드웨어에 대한 전반적인 이해를 바탕으로 구축된다. 센서와 구동기, 비행조종 컴퓨터, 데이터 통신버스 등이 다중화 되어 있으며, 다중화관리 시스템은 각종 하드웨어의 고장진단, 분리, 그리고 관리기능을 수행한다. 하드웨어 여분 방법은 단순히 고장의 유무를 판단하는 알고리즘만을 적용하는 것에 비해, 여러 종류의 고장모델에 대해 고장의 종류와 크기도 검출해 낸다는 점에서 고장의 정량적 분석이 가능한 특징을 가지고 있다. 그러나 이러한 접근방법은 고려하고자 하는 고장의 종류가 많아지면 발생한 고장을 검출하고 파악하는데 많은 시간이 요구되며, 사전에 예측한 고장이 아닌 다른 고장이 발생할 경우에는 대처하기가 힘들다는 단점을 가지고 있다. 또한, 외란이나 부정확한 시스템 모델로 인한 오차신호를 고장으로 오인하여 잘못된 제어를 적용할 가능성도 있다.

라. 해석적 방법

시스템의 수학적 모델에 신호를 입력시켜 발생하는 출력과 실제로 측정된 시스템의 출력을 비교하여 고장을 찾아내는 방법을 해석적 여분(analytical redundancy)을 이용한 고장진단기법이라고 한다. 하드웨어 여분을 이용하는 기법이 단지 고장의 유무를 선언하는 정성적(qualitative) 기법인데 비해, 수학적 모델을 기준으로 하여 고장의 검출 및 분리를 수행한다는 점에서 정량적(quantitative)인 기법이라고 할 수 있다. 해석적 여분을 이용한 고장검출기법의 구성도는 그림 3.2.3에 정리되어 있다.

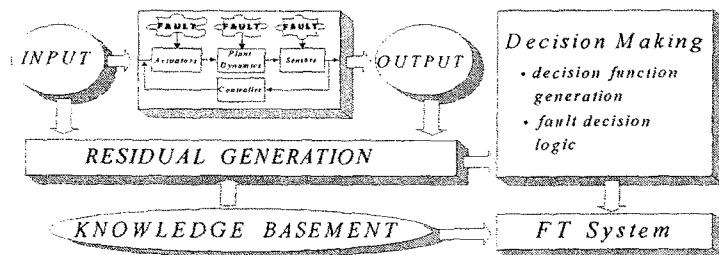


그림 3.2.3 해석적 여분을 이용한 고장검출

그림 3.2.3에 보이고 있는 수학적 모델과 실제 시스템의 출력으로부터 계산되는 잔차(residual)는 해석적 여분을 이용한 고장진단기법의 공통적인 특성이라고 할 수 있으며, 잔차를 어떻게 만들어내느냐에 따라 고장검출 및 분리기법들은 다양하게 분류된다. 한편 시스템을 식별하여 제어를 재설계하는 방법은 고장이 감지되면 시스템 식별 알고리즘을 통해 실시간으로 파손된 항공기의 수학적 모델을 구하고 발생한 고장의 종류를 파악한다. 이렇게 얻어진 시스템 모델과 고장 정보를 토대로 항공기의 비행제어시스템을 재설계하여 고장에 대처한다. 그림 3.2.4는 본 연구에서 수행하고 있는 고장허용 제어알고리즘에 대한 흐름도를 보이고 있다.

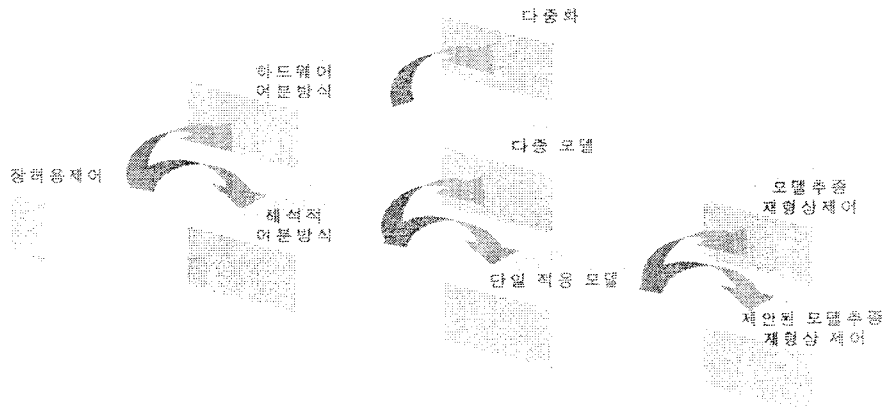


그림 3.2.4 고장허용제어 분류 및 개괄도

2. 비행체 고장의 개요 및 특성

항공기는 주어진 임무를 수행하는 과정에서 많은 고장 요인들을 지니고 있으며, 고장은 심각한 사고로 이어지기도 한다. 따라서 항공기의 신뢰성을 높이고, 부가가치를 높이기 위하여 근래에 고장 검출 및 분리에 대한 여러가지 기법들이 연구되고 있으며, 실제 항공기에서 적용되는 사례들도 많이 발견할 수 있다.

가. 고장의 정의

고장(Fault)은 내, 외적인 요인에 의한 시스템의 비정상적인 반응이라고 정의할 수 있으며, 대처 불가능한 상태를 지칭하는 파손(Breakdown)과는 다른 의미를 가지고 있다. 이러한 고장에 대해 대처하는 여러 과정을 일컫는 말이 있는데, 고장 검출(Fault Detection)은 시스템에 발생한 고장을 찾아내는 기법을 말하며 고장 경고 등에 의해 구현된다. 고장 분리(Fault Isolation)는 발생한 고장신호를 다른 신호들과 분리하여 고장의 위치를 파악하는 것을 말하고, 고장 인식 (Fault Identification)은 고장의 정도를 인식하여 고장의 종류를 판별하는 것을 일컫는다.

나. 고장의 분류 및 고장 요인

항공기 사고 발생 요인을 분석한 표를 그림 3.2.5에 도시하였다. 그림에서 알 수 있듯이 승무원의 실수나 불찰에 의한 사고가 가장 큰 요인을 차지하고 있고, 비행체 자체의 고장이 그 다음으로 사고에 영향을 주고 있음을 알 수 있다.

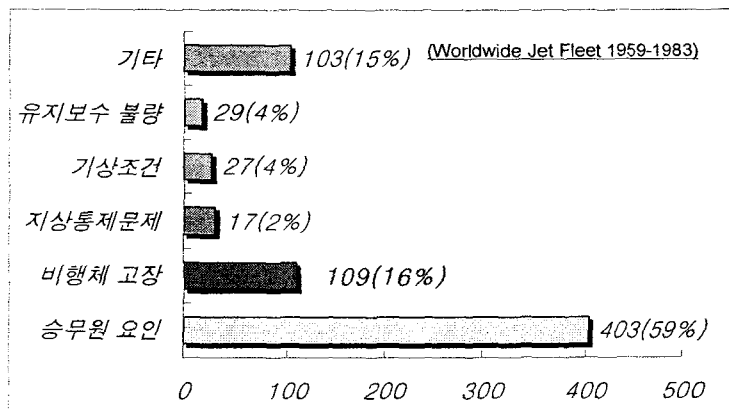


그림 3.2.5 항공기 사고 발생 요인별 분석

실제 항공기는 임무를 수행하는 과정에서 많은 고장의 요인들이 존재하며, 이륙하는 순간부터 착륙할 때까지 다양한 원인에 의해서 고장이 발생할 수 있다. 항공기 임무 수행 중 발생할 수 있는 고장의 요인들을 그림 3.2.6 에 나타내었다.

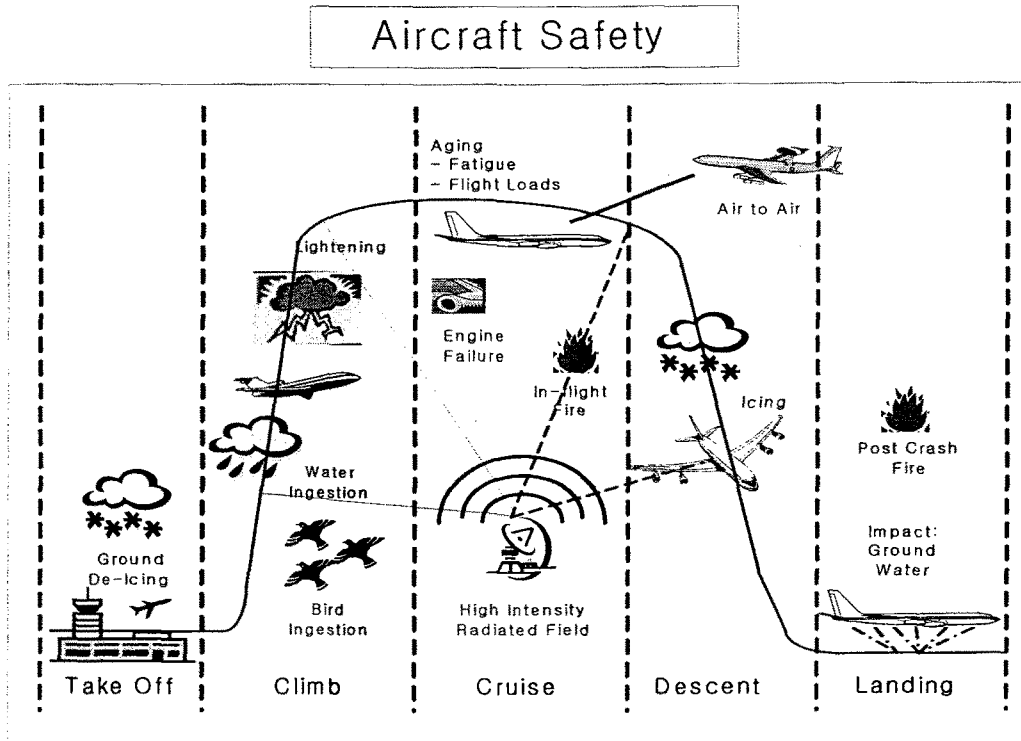


그림 3.2.6 항공기 임무(mission) 수행 중 고장 요인

다. 고장 검출 및 인식 기법의 분류

비행기는 작은 고장도 큰 사고의 원인이 되므로 모든 고장의 발생에 대한 적절한 대처를 하여 항공기의 신뢰성을 높여야 한다. 본 연구에서는 이러한 고장 검출 및 인식(Fault Detection and Isolation : FDI)에 관한 여러 기법들을 연구하고 시뮬레이션을 수행하였다. 고장 검출 및 인식 기법 분류를 그림 3.2.7에 도시하였다.

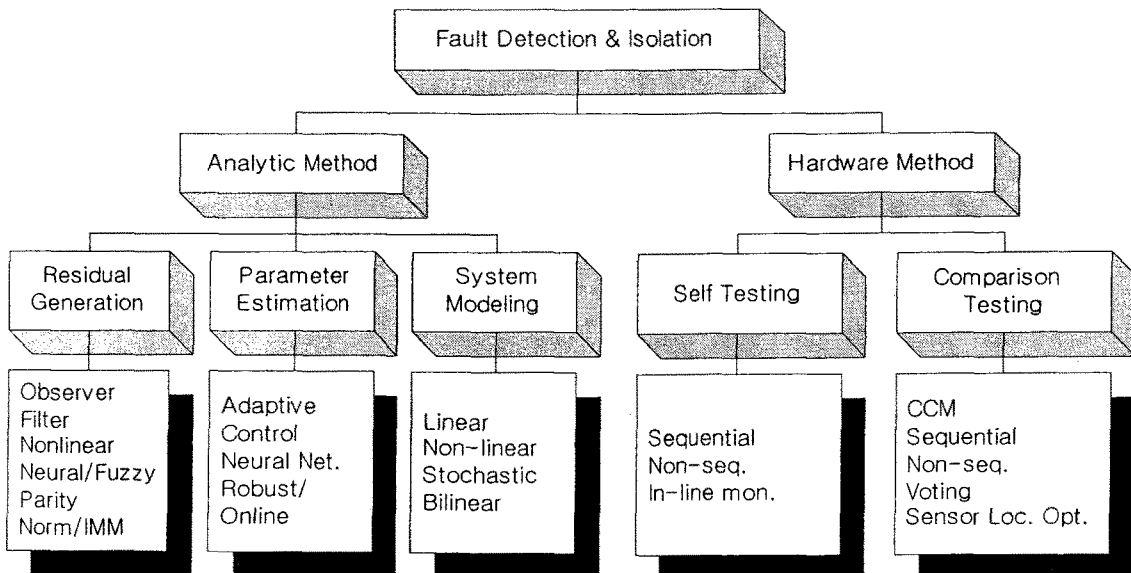


그림 3.2.7 FDI 기법의 분류

3. 신경회로망을 이용한 비선형 제어시스템 설계기법 연구

가. 서론

항공기 제어시스템은 일반적으로 특정한 평형점에서 선형화된 모델을 이용해서 설계한다. 이렇게 설계된 제어시스템은 특정한 동작점 근처에서는 훌륭한 성능을 보장하지만, 보다 넓은 비행영역에서는 동일한 성능을 보장 받을 수 없다. 따라서 현재는 넓은 비행영역에서 운용할 수 있는 비행제어시스템을 설계하기 위해서 여러 동작점에서 선형화된 모델을 이용해 제어기를 설계한 후, 설계된 제어기들을 보간 하여 사용하는 게인 스케줄링 방법이 사용되고 있다. 게인 스케줄링 방법은 실제로 운용 탑재하기 위해서는 많은 동작점에서 반복적인 설계가 수행되어야 하고, 동작점과 동작점 사이에서는 제어기 성능이 저하되는 단점이 있다. 이러한 문제를 해결하기 위해서 최근에는 비선형 모델을 이용한 비선형 제어기법이 많이 연구되고 있다. 특히, 항공기 동특성을 고려하여 two-time scale separation 가정을 이용한 제어기법이 제시되었다. 그러나 이 방법은 불확실성에 대해 강건성을 유지할 수 없다는 단점이 있다.

한편, 최근의 항공기는 높은 받음각으로 빠른 기동을 할 수 있는 성능이 요구되는데, 이러한 경우 정확한 수학적 모델을 얻기는 매우 어려우며 그로 인한 시스템 불확실성의 영향이 더욱 커지고 있다. 조종면 파손, 날개 파손, 구동기 고장, 센서고장 등이 발생하는 경우에는 모델링이 되지 않은 비선형 동특성이 영향을 줄 뿐 아니라, 공력계수 등의 불확실성 영향이 심해진다. 따라서 이러한 경우에도 자동적으로 시스템의 손상을 보상하여 원하는 성능을 얻을 수 있는 재형상 제어기(Reconfigurable Controller)에 관한 연구들이 최근에 활발히 진행 중이다. Morse와 Ossman은 AFTI/F-16 비행기의 선형시스템에 대해 모델추종기법을 이용하여 재형상 비행제어기를 설계하였고, Ochi 와 Kanai는 되먹임 선형화 기법과 온라인 파라미터 식별기법을 이용하여 비선형시스템에 대한 재형상 비행제어기를 제안하였다. 또한, Shtessel은 슬라이딩 모드기법을 이용한 제어기를 설계하여, 항공기 시스템을 빠른 동특성과 느린 동특성을 갖는 운동방정식을 서로 분리해서 생각하는 two-time separation 가정을 통해 제안한 제어기를 적용하였다.

불확실성을 포함한 시스템에 대해 최근에 활발히 연구가 되고 있는 제어기법은 신경회로망을 이용한 제어기법이다. 신경회로망은 임의의 복잡한 비선형 함수를 표현할 수 있는 성질로 미지의 시스템의 제어와 식별을 위한 도구로 많이 사용되어지고 있다. 항공기 시스템에 신경회로망을 적용한 많은 연구들이 수행되어져 왔는데 그 중 중요한 연구를 살펴보면 다음과 같다.

Kim과 Calise는 신경회로망을 이용한 직접적인 적응 추종제어기를 개발하여, 이를 F-18 전투기의 CAS(Control Augmentation System)적용하였다. 그들은 2개의 신경회로망을 사용하였는데 하나는 되먹임선형화를 위해 지상에서 off-line으로 운동방정식의 역변환을 학습시키는데 사용하고, 또 다른 하나는 역치환으로 인한 오차항을 on-line으로 학습하여 보상하는데 사용하였다. 이들 연구의 제어목적은 각속도를 제어하기 위해 운동학 방정식을 통해 생성된 명령 뱅크각, 피치각, 요각을 추종하도록 하는 것이다. 그들의 연구를 바탕으로 tilt-rotor 항공기와 헬리콥터에 적용된 연구가 발표되었고, McFarland와 Calise는 이전의 연구를 확장하여 미사일의 받음각, 뱅크각, 옆미끄럼각을 제어하기 위해 two-time separation 가정을 사용하지 않고 백스테핑(Backstepping) 기법을 사용한 제어기를 설계

하였다. 그러나 그들은 빠른 동특성을 갖는 각속도 운동방정식 성분만을 잘 알지 못한다는 가정 하에 유도하였다. 즉, 제어하고자 하는 상태변수 관련 운동방정식은 잘 안다는 가정 하에 유도를 하였다. 최근에는 tail-less 항공기에 대해 직접적 신경회로망을 기반으로 한 적응제어기에 대한 연구가 수행되어 졌는데, 이전의 연구와 마찬가지로 2차 시스템을 기반으로 하였고, off-line 신경회로망 학습이 요구된다. Lee와 Kim은 공력계수 변화에 의한 시스템 오차를 신경회로망을 통해 보상할 수 있는 제어기를 제안하였는데, MacFarland와 Calise가 연구한 방법과 마찬가지로 제어하고자 하는 상태변수 관련 운동방정식은 정확히 알고 있다는 가정 하에 제어기를 유도하였다. 이러한 가정 하에 유도한 이유는 다른 연구와 마찬가지로 일반적으로 오차항을 보상하기 위해 제어기에 생기는 강건성을 보장하기 위한 항이 생기는데, 이들은 일반적으로 절대값 형태를 취하고 있기 때문에 특정한 값에 대해서는 미분 불가능하게 된다. 따라서 직접적으로 백스테핑 기법을 사용하는 것이 불가능해진다. 이러한 문제를 해결하기 위해서는 미분가능한 강건한 항을 찾거나 다른 방법을 강구하여야 한다.

이와 같이 기존의 신경회로망을 이용한 비행제어시스템 연구들은 시스템의 일부 동특성만 불확실하다는 가정을 기반으로 하여 제어기를 설계하였다. 본 연구에서는 기존의 연구와는 달리 시스템의 모든 동특성에 불확실성이 존재한다는 가정 하에 백스테핑 기법과 공력계수의 불확실성에 의해 나타날 수 있는 오차, 그리고 조종면 파손으로 인한 제어효과 감소 하에서도 원하는 성능을 보장할 수 있는 신경회로망을 이용한 적응제어기를 제안하였다. 제안한 비선형 제어기에 사용된 신경회로망의 가중치 값은 르야프노프 안정성 이론을 바탕으로 안정한 on-line 학습 알고리즘을 유도하였고, 백스테핑 기법을 적용하기 위해 모든 점에서 미분가능한 형태의 비레이득값을 유도하였다. 제안한 제어기의 효용성을 보이기 위해 6자유도 F-16 비선형 항공기 시스템에 적용하였다.

나. 신경회로망의 구조와 기본적인 성질

본 연구에서 사용하고자 하는 신경회로망의 구조는 3개의 층을 가지는 다층 신경회로망이다. 일반적으로 한 개의 은닉층을 가진 신경회로망은 임의의 연속적인 함수를 근사화 할 수 있다. $\bar{x} \in R^{N+1}$ 에 대한 신경회로망의 출력을 행렬 형태로 표현하면 다음과 같다.

$$y = W^T \sigma(V^T \bar{x}) \quad (3.2.1)$$

여기서

$$W = [w_1 \ w_2 \ \dots \ w_{N_2}] \in R^{N_2 \times N_2}$$

$$V = [v_1 \ v_2 \ \dots \ v_{N_1}] \in R^{(N_1+1) \times N_2}$$

$$\bar{x} = [1 \ x_1 \ \dots \ x_{N_1}]^T \in R^{N_1+1}$$

$$\sigma(z) = [\sigma(z_1) \ \sigma(z_2) \ \dots \ \sigma(z_{N_1})]^T$$

또한, w 는 은닉층과 출력층을 연결하는 가중치 행렬, v 는 입력층과 은닉층을 연결하는 가중치 행렬, 그리고 $\sigma(\cdot)$ 는 은닉층 활성화 함수이다. 그림 3.2.8은 본 연구에서 사용된 신경회로망 구조이다. 신경회로망의 활성화 함수는 다음과 같은 Sigmoidal 함수를 사용하였다.

$$\sigma(z) = 1/(1 + \exp^{-z}) \quad (3.2.2)$$

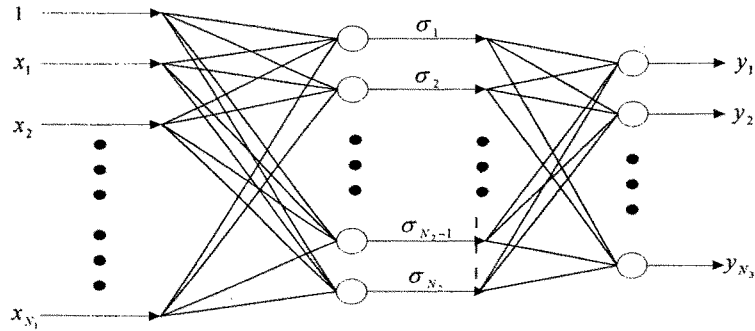


그림 3.2.8 신경회로망의 구조

우선, 신경회로망의 이상적인 가중치 행렬 w 와 v 의 추정치 행렬을 각각 \hat{w} 와 \hat{v} , 그리고 가중치 추정오차를 각각 $\tilde{w} = w - \hat{w}$, $\tilde{v} = v - \hat{v}$ 로 정의한다. 테일러급수 정리를 이용하면 다음의 정리 3.2.1을 유도할 수 있다.

정리 3.2.1 신경회로망 식(3.2.1)에 대해서 신경회로망 추정오차는 다음과 같이 표현될 수 있다.

$$W^T \sigma(V^T \bar{x}) - \hat{W}^T \sigma(\hat{V}^T \bar{x}) = \tilde{W}^T (\hat{\sigma} - \hat{\sigma}') \hat{V}^T \bar{x} + \tilde{W}^T \hat{\sigma}' \hat{V}^T \bar{x} + \theta \quad (3.2.3)$$

$$\theta = W^T \sigma(V^T \bar{x}) - W^T (\hat{\sigma} - \hat{\sigma}') \hat{V}^T \bar{x} - \tilde{W}^T \hat{\sigma}' \hat{V}^T \bar{x} \quad (3.2.4)$$

여기서 $\hat{\sigma}' = \frac{d\sigma(z)}{dz} \Big|_{z=\hat{V}^T \bar{x}}$, $\hat{\sigma} = \sigma(\hat{V}^T \bar{x})$ 이다.

이상적인 가중치 행렬들의 Frobenius 노름이 한정적이라고 가정하고, 노름의 일반적인 성질을 이용하면 다음 정리를 얻을 수 있다.

정리 3.2.2 정리 3.2.1에서 여분의 항인 θ 는 다음 부등식을 만족한다.

$$\|\theta\| \leq c_1 + c_2 \|\bar{x}\| \|\tilde{w}\|_F + c_3 \|\bar{x}\| \|\tilde{v}\|_F \quad (3.2.5)$$

증명: 식 (3.2.4)의 양쪽에 노름(norm)을 취하고, 식 (3.2.2)에서 정의된 활성화 함수와 그 미분값은 유한하다는 점을 고려하면, 다음 관계식을 유도할 수 있다.

$$\begin{aligned} \|\theta\| &\leq \|\tilde{W}^T \hat{\sigma}'\|_F \|\hat{V}^T \bar{x}\| + \|\tilde{W}\|_F \|\hat{\sigma}\| + \|\tilde{W}^T \hat{\sigma}'\|_F \|\hat{V}^T \bar{x}\| + \|\tilde{W}\|_F \|\sigma\| \\ &\leq W_M a_1 \|\tilde{w}\|_F \|\bar{x}\| + W_M a_2 + W_M a_1 \|\tilde{v}\|_F \|\bar{x}\| + W_M a_2 \\ &= c_1 + c_2 \|\bar{x}\| \|\tilde{w}\|_F + c_3 \|\bar{x}\| \|\tilde{v}\|_F \end{aligned} \quad (3.2.6)$$

여기서 c_1, c_2 그리고 c_3 는 계산이 가능한 계수이다. 그러나 실제 제어시스템을 구현할 때

는 사용되지 않는다.

다. 항공기 시스템 모델 및 제어기 설계

본 연구에서 고려하고 있는 6자유도 항공기 모델은 다음과 같다. 항공기는 강체이고 중력, 추력은 일정하다고 가정하였다.

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\phi} \end{bmatrix} = \frac{1}{mV} \begin{bmatrix} -\frac{S_0}{C_\beta} [T+c_x \bar{q} S] + \frac{C_x}{C_\beta} c_x \bar{q} S \\ -C_n S_\beta [T+c_x \bar{q} S] + C_\beta c_x \bar{q} S - S_n S_\beta c_x \bar{q} S \\ -C_n S_\beta [T+c_x \bar{q} S] + C_\beta c_x \bar{q} S - S_n S_\beta c_x \bar{q} S \end{bmatrix} + \begin{bmatrix} -C_\beta T_\beta & 1 & -S_\beta T_\beta \\ S_\beta & 0 & -C_\beta \\ 1 & S_\beta T_\beta & C_\beta T_\beta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ + \frac{\rho S}{4m} \begin{bmatrix} 0 & -\frac{S_0}{C_\beta} c_{xq} \bar{c} + \frac{C_x}{C_\beta} c_{xq} \bar{c} & 0 \\ C_{\beta c_{yp}} b - C_n S_\beta c_{xq} \bar{c} - S_n S_\beta c_{xq} \bar{c} & C_{\beta c_{yr}} b & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{\rho VS}{2m} \begin{bmatrix} -\frac{S_n}{C_\beta} c_{x\alpha} + \frac{C_x}{C_\beta} c_{x\alpha} & 0 & 0 \\ -C_n S_\beta c_{x\alpha} - S_n S_\beta c_{x\alpha} & C_\beta c_{y\alpha} & C_\beta c_{y\alpha} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \end{bmatrix} \quad (3.2.7)$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & \frac{C_\phi}{C_\theta} & -\frac{S_\phi}{C_\theta} \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (3.2.8)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_2 p q + I_1 q r \\ I_3 p r - I_6 (p^2 - r^2) \\ -I_2 q r + I_8 p q \end{bmatrix} + \begin{bmatrix} I_3 c_{\bar{q}} S b + I_1 c_n \bar{q} S b \\ I_7 c_m \bar{q} S \bar{c} \\ I_1 c_{\bar{q}} S b + I_9 c_n \bar{q} S b \end{bmatrix} + \frac{\rho VS}{4} \begin{bmatrix} I_3 c_{\beta p} b + I_1 c_{n p} b & 0 & I_3 c_{\beta} b + I_4 c_m b \\ 0 & I_7 c_{m q} \bar{c} & 0 \\ I_1 c_{\beta p} b + I_9 c_{n p} b & 0 & I_1 c_{\beta} b + I_9 c_m b \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \\ + \bar{q} S b \begin{bmatrix} 0 & I_3 c_{\beta} + I_1 c_{n \beta} & I_3 c_{\beta} + I_1 c_{n \beta} \\ I_7 c_{m \beta} \frac{\bar{c}}{b} & 0 & 0 \\ 0 & I_4 c_{\beta} + I_9 c_{n \beta} & I_4 c_{\beta} + I_9 c_{n \beta} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \end{bmatrix} \quad (3.2.9)$$

여기서 S_n, C_n, T_n 는 각각 $\sin A, \cos A, \tan A$ 를 나타내며, I_i 는 관성모멘트로 구성된 항이다. 공기역학적인 힘과 모멘트 계수는 받음각과 옆미끄럼각의 함수로서 표현되며, 본 연구에서는 F-16 항공기 모델을 이용하였다. 상태변수와 제어입력을 다음과 같이 정의하자.

$$\begin{aligned} x_1 &= [\alpha \ \beta \ \phi]^T \\ x_2 &= [p \ q \ r]^T \\ x_3 &= [\theta \ \psi]^T \\ u &= [\delta_e \ \delta_a \ \delta_r]^T \end{aligned}$$

정의된 상태변수와 제어입력에 따라 운동방정식을 다음과 같이 표현할 수 있다.

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_3) + g_1(x_1, x_3)x_2 + h_1(x_1)u \\ \dot{x}_2 &= f_2(x_1, x_2) + g_2(x_1)u \\ \dot{x}_3 &= h_3(\phi, \theta)x_2 \end{aligned} \quad (3.2.10)$$

제어기 설계 단계를 설명하기 위해 다음과 같은 가정을 필요로 한다.

[가정 3.2.1] 시스템을 구성하는 비선형 함수는 정확하게 알지는 못한다. 즉, 공력계수에

의한 불확실성이 존재한다.

[가정 3.2.2] $\|h_1(x_1)u\| \leq c_h$ 인 작은 값의 c_h 가 존재한다. 원하는 받음각의 궤적을 x_{1d} 이라고 하고, x_1 상태변수의 추종오차 및 Σ 를 다음과 같이 정의하자.

$$z_1 = x_1 - x_{1d} \quad (3.2.11)$$

$$\Sigma = \{f_1(x_1, x_3) + g_1(x_1, x_3)x_2\} - \{\hat{f}_1(x_1, x_3) + \hat{g}_1(x_1, x_3)x_2\} \quad (3.2.12)$$

이제, x_1 운동방정식의 오차방정식은 다음과 같다.

$$\dot{z}_1 = \Sigma + \hat{f}_1(x_1, x_3) + \hat{g}_1(x_1, x_3)x_2 + h_1(x_1)u - \dot{x}_{1d} \quad (3.2.13)$$

여기서 $\hat{f}_1(x_1, x_3), \hat{g}_1(x_1, x_3)$ 는 각각 실제 모델의 비선형 함수 $f_1(x_1, x_3)$ 과 $g_1(x_1, x_3)$ 값에 대해 설계 전에 알고 있는 추정치이다. 식 (3.2.13)에서 오차항인 Σ 를 신경회로망을 이용하여 표현하면 다음과 같다.

$$\Sigma = \bar{W}_1^T \sigma(V_1^T \bar{x}_1) + \epsilon(\bar{x}_1), \|\epsilon(\bar{x}_1)\| \leq \epsilon_1 \quad (3.2.14)$$

여기서 $\bar{x}_1 = [1 \ x_1^T \ x_2^T \ x_3^T]^T$ 이다.

백스테핑 방법을 적용하기 위해 x_2 를 가상의 제어변수로 설정한 후, 오차방정식을 안정화 할 수 있는 안정화 제어기 v 를 다음과 같이 정하자.

$$v = \hat{g}_1(x_1, x_3)^{-1} [-\hat{f}_1(x_1, x_3) - K_1 z_1 + \dot{x}_{1d} - \widehat{W}_1^T \sigma(\widehat{V}_1^T \bar{x}_1)] \quad (3.2.15)$$

식 (3.2.11)과 (3.2.15)를 식 (3.2.13)에 대입하고, 새로운 오차변수 $z_2 = x_2 - v$ 로 정의하여 정리 3.2.1을 이용하면, z_1 에 대한 폐회로 오차방정식을 다음과 같이 얻을 수 있다.

$$\dot{z}_1 = -K_1 z_1 + \hat{g}_1(x_1, x_3)z_2 + \bar{W}_1^T (\hat{\sigma}_1 - \hat{\sigma}_1^*) V_1^T \bar{x}_1 + \widehat{W}_1^T \hat{\sigma}_1^* V_1^T \bar{x}_1 + \theta_1 + \epsilon(\bar{x}_1) \quad (3.2.16)$$

한편, z_2 에 대한 오차방정식을 얻기 위해, 안정화 제어기 v 의 시간에 대한 미분값을 변수에 대한 편미분값에 대해 표현하면 다음과 같이 표현할 수 있다.

$$\dot{v} = \frac{\partial v}{\partial x_1} \dot{x}_1 + \frac{\partial v}{\partial x_2} \dot{x}_2 + \frac{\partial v}{\partial x_3} \dot{x}_3 + \Omega_1 \quad (3.2.17)$$

여기서

$$\Omega_1 = \frac{\partial v}{\partial x_{1d}} \dot{x}_{1d} + \frac{\partial v}{\partial x_{1d}} \ddot{x}_{1d} + \sum_{i=1}^{N_1} \frac{\partial v}{\partial w_1} \dot{w}_1^i + \sum_{i=1}^{N_1} \frac{\partial v}{\partial v_{1,i}} \dot{v}_{1,i} \quad (3.2.18)$$

식 (3.2.17)을 이용하여 z_2 에 대한 오차방정식을 정리하면 다음과 같다.

$$\begin{aligned} \dot{z}_2 &= f_2(x_1, x_2) - \frac{\partial v}{\partial x_2} f_2(x_1, x_2) - \frac{\partial v}{\partial x_1} f_1(x_1, x_3) - \frac{\partial v}{\partial x_3} g_1(x_1, x_3)x_2 - \frac{\partial v}{\partial x_3} f_3(\theta, \psi)x_2 - \Omega_1 \\ &\quad + \left\{ g_2(x_1) - \frac{\partial v}{\partial x_2} g_2(x_1) - \frac{\partial v}{\partial x_1} h_1(x_1) \right\} u \\ &\triangleq f_3(x_1, x_2, x_3, \Omega_1, \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2}) + g_3(x_1, \frac{\partial v}{\partial x_1}, \frac{\partial v}{\partial x_2})u + g_2(x_1)u \end{aligned} \quad (3.2.19)$$

식 (3.2.13)를 식 (3.2.16)까지의 과정과 같은 방법으로 z_2 에 관한 폐회로 운동방정식과 제어기에 관한 식을 유도하면 다음과 같다.

$$\dot{z}_2 = -K_2 z_2 - \hat{g}_1(x_1, x_3)^T z_1 + \bar{W}_2^T (\hat{\sigma}_2 - \hat{\sigma}_2^* \bar{V}_2^T \bar{x}_2) + \bar{W}_2^T \hat{\sigma}_2^* \bar{V}_2^T \bar{x}_2 + \theta_2 + \epsilon(\bar{x}_2) \quad (3.2.20)$$

$$u = \hat{g}_2^{-1} (-\hat{f}_3 - K_2 z_2 - \hat{g}_1^T z_1 - \bar{W}_2^T \sigma (\bar{V}_2^T \bar{x}_2)) \quad (3.2.21)$$

여기서 $\hat{f}_3 = \hat{f}_2 - \frac{\partial v_1}{\partial x_2} \hat{f}_2 - \frac{\partial v_1}{\partial x_1} \hat{f}_1 - \frac{\partial v_1}{\partial x_3} \hat{g}_1 x_2 - \frac{\partial v_1}{\partial x_3} f_3 x_2 - \Omega_1$, $\hat{g}_3 = -\frac{\partial v}{\partial x_2} g_2 - \frac{\partial v}{\partial x_1} h_1$ 이다.

또한, $\bar{x}_2 = [1 \ x_1^T \ x_2^T \ x_3^T \ \Omega_1 \ (\frac{\partial v}{\partial x_1}) \ (\frac{\partial v}{\partial x_2}) \ u^T]^T$ 이며, (\bar{A}) 는 행렬 A 의 원소들을 1열로 나열하여 열 벡터로 만드는 operator이다. 본 연구에서 제안된 전체 제어기의 구조는 그림 3.2.9와 같다.

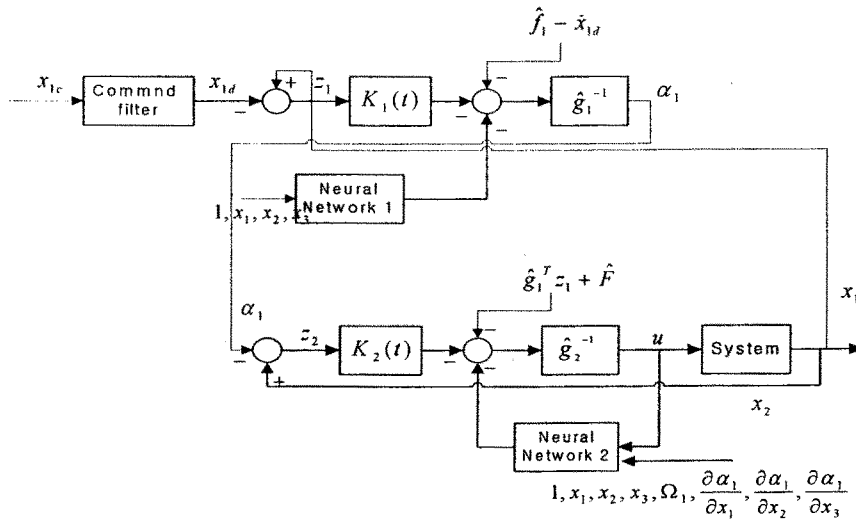


그림 3.2.9 제안된 제어기의 구조

라. 안정성 해석

앞 절에서 유도한 폐회로 오차방정식과 안정화 제어기, 시스템 제어기에 관한 식을 바탕으로 전체 시스템의 안정성을 분석하자. 먼저, 다음과 같은 르야프노프 함수를 고려하자.

$$V_1 = \frac{1}{2} z_1^T z_1 + \frac{1}{2\gamma_{w1}} \text{tr}[\bar{W}_1^T \bar{W}_1] + \frac{1}{2\gamma_{v1}} \text{tr}[\bar{V}_1^T \bar{V}_1] \quad (3.2.22)$$

위 식을 시간에 관해 미분하고, 식 (3.2.16)를 대입한 후에 다음과 같은 신경회로망 가중치 행렬의 적응법칙을 고려하자.

$$\dot{\widehat{W}}_1 = \gamma_{w1}(\widehat{\sigma}_1 - \widehat{\sigma}_1' \widehat{V}_1^T \overline{x}_1) z_1^T - \eta_{w1} \gamma_{w1} \widehat{W}_1 \quad (3.2.23)$$

$$\dot{\widehat{V}}_1 = \gamma_{v1} \overline{x}_1 (\widehat{\sigma}_1'^T \widehat{W}_1 z_1)^T - \eta_{v1} \gamma_{v1} \widehat{V}_1$$

식(3.2.6)과 식(3.2.23)를 이용하면, 결국 \dot{V}_1 은 다음과 같이 쓸 수 있다.

$$\begin{aligned} \dot{V}_1 \leq & -K_1 |z_1|^2 + (\epsilon_1 + c_1 + c_h) |z_1| + (c_2 \|\overline{x}_1\| \|\widehat{W}_1\|_F + c_3 \|\overline{x}_1\| \|\widehat{V}_1\|_F) |z_1| \\ & + z_1^T \widehat{g}_1 z_2 + \eta_{w1} \text{tr}[\widehat{W}_1^T \widehat{W}_1] + \eta_{v1} \text{tr}[\widehat{V}_1^T \widehat{V}_1] \end{aligned} \quad (3.2.24)$$

위 식에 $a|z_1| \leq \frac{\mu}{4} a^2 + |z_1|^2$ 의 부등식 성질과 $\text{tr}[\overline{A}^T \overline{A}] \leq \|\overline{A}\|_F A_M - \|\overline{A}\|_F^2$ (여기서 $\|\overline{A}\|_F \leq A_M$)을 이용하여 정리하면, 다음과 같은 부등식을 얻을 수 있다

$$\dot{V}_1 \leq -\left(K_1 - \left\{3 + \|\overline{x}_1\|^2 \|\widehat{W}_1\|_F^2 + \|\overline{x}_1\|^2 \|\widehat{V}_1\|_F^2\right\} |z_1|^2 - \frac{\eta_{w1}}{2} \|\widehat{W}_1\|_F^2 - \frac{\eta_{v1}}{2} \|\widehat{V}_1\|_F^2 + G \right) \quad (3.2.25)$$

여기서 $G = \frac{\mu}{4} (c_1^2 + c_h^2 + c_1^2 + c_2^2 + c_3^2) + \frac{\eta_{w1}}{2} W_M^2 + \frac{\eta_{v1}}{2} V_M^2$

최종적으로 다음과 같은 르야프노프 함수를 설정하자.

$$V_2 = V_1 + \frac{1}{2} z_2^T z_2 + \frac{1}{2\gamma_{w2}} \text{tr}[\widehat{W}_2^T \widehat{W}_2] + \frac{1}{2\gamma_{v2}} \text{tr}[\widehat{V}_2^T \widehat{V}_2] \quad (3.2.26)$$

두 번째 신경회로망의 가중치 행렬 적응법칙과 제어이득 K_1, K_2 를 다음과 같이 선택하자.

$$\dot{\widehat{W}}_2 = \gamma_{w2}(\widehat{\sigma}_2 - \widehat{\sigma}_2' \widehat{V}_2^T \overline{x}_2) z_2^T - \eta_{w2} \gamma_{w2} \widehat{W}_2 \quad (3.2.27)$$

$$\dot{\widehat{V}}_2 = \gamma_{v2} \overline{x}_2 (\widehat{\sigma}_2'^T \widehat{W}_2 z_2)^T - \eta_{v2} \gamma_{v2} \widehat{V}_2$$

$$K_1 = 3 + \|\overline{x}_1\|^2 \|\widehat{W}_1\|_F^2 + \|\overline{x}_1\|^2 \|\widehat{V}_1\|_F^2 + K_{11} \quad (3.2.28)$$

$$K_2 = 2 + \|\overline{x}_2\|^2 \|\widehat{W}_2\|_F^2 + \|\overline{x}_2\|^2 \|\widehat{V}_2\|_F^2 + K_{21} \quad (3.2.29)$$

여기서 $K_{11}, K_{21} > 0$ 이다. 식 (3.2.26)를 미분하고 식 (3.2.20), (3.2.25), (3.2.27)-(3.2.29)을 이용하여 정리하면, 최종적으로 다음 식을 얻을 수 있다.

$$\begin{aligned} \dot{V}_2 \leq & -K_{11} |z_1|^2 - K_{21} |z_2|^2 - \frac{\eta_{w1}}{2} \|\widehat{W}_1\|_F^2 \\ & - \frac{\eta_{v1}}{2} \|\widehat{V}_1\|_F^2 - \frac{\eta_{w2}}{2} \|\widehat{W}_2\|_F^2 - \frac{\eta_{v2}}{2} \|\widehat{V}_2\|_F^2 + C \end{aligned} \quad (3.2.30)$$

여기서 $C = G_1 + G_2$, G_2 는 두 번째 신경회로망에 의해 나타나는 계수들로 구성된 식으로서, G_1 과 같은 구조를 갖는다. 이제 다음 식을 만족하는 τ 를 설정하자.

$$0 < \tau < \min \left[K_{11}, K_{21}, \frac{\eta_w \gamma w_i}{2}, \frac{\eta_w \gamma v_i}{2} \right], (j = 1, 2) \quad (3.2.31)$$

식 (3.2.30)에 식 (3.2.22)과 (3.2.26), (3.2.31)을 고려하면, 다음 관계식을 얻을 수 있다.

$$\dot{V}_2 \leq -2\tau V_2 + C \quad (3.2.32)$$

그러므로 v_2 가 $C/2\tau$ 보다 크면, v_2 는 음수가 된다. 따라서 상태변수들의 오차와 가중치 파라미터의 오차들은 결국 ultimately bounded 된다. 그리고 τ 가 크게 되면 보다 빠른 수렴성을 보일 수 있는데, 이를 위해서는 이득값과 르야프노프 가중치를 적절히 조절하여야 한다.

마. 수치예제

설계된 제어기의 성능을 입증하기 위해서 $V=500$ ft/s, $h=10000$ ft의 정상수평 비행상태의 F-16 항공기에 다음과 같은 α, β, ϕ 명령을 주어 수치 시뮬레이션을 수행하였다.

$$\alpha^d = 0^\circ, \beta^d = 0^\circ, \phi^d = 0^\circ \quad 0 \leq t \leq 1(\text{sec})$$

$$\alpha^d = 10^\circ, \beta^d = 0^\circ, \phi^d = 50^\circ \quad 1 \leq t \leq 10(\text{sec})$$

$$\alpha^d = -2^\circ, \beta^d = 0^\circ, \phi^d = 0^\circ \quad 10 \leq t \leq 15(\text{sec})$$

미분가능한 명령을 생성하기 위하여 다음과 같은 3차 선형시스템 형태의 명령생성 필터가 사용되었다.

$$\frac{\alpha^d(s)}{\alpha^c(s)} = \frac{w_{n1}w_{n2}^2}{(s + w_{n1})(s^2 + 2\zeta_1 w_{n2}s + w_{n2}^2)}$$

$$\frac{\beta^d(s)}{\beta^c(s)} = \frac{w_{n1}w_{n2}^2}{(s + w_{n1})(s^2 + 2\zeta_1 w_{n2}s + w_{n2}^2)}$$

$$\frac{\phi^d(s)}{\phi^c(s)} = \frac{w_{n1}w_{n2}^2}{(s + w_{n1})(s^2 + 2\zeta_1 w_{n2}s + w_{n2}^2)}$$

여기서 $w_{n1} = 12\text{rad/sec}$, $w_{n2} = 3\text{rad/sec}$, 그리고 $\zeta_1 = 0.8$ 이다.

일반적인 제어면 고장으로는 조종면 파손을 생각할 수 있다. 조종면 파손은 전체 조종면의 일부 손실로 표현되고, 조종면 손실은 손상된 제어면에 해당하는 입력 비선형 행렬의 열 벡터에 파손되지 않은 비율만큼의 조종면 효과계수를 곱함으로써 모델링 될 수 있다.

시뮬레이션 상에 가정한 제어면 손상은 각각 엘리베이터는 50%, 에일러론은 50%, 그리고 러더 30%로 설정하였다. 고장이 발생한 시간은 모두 0초에 일어났다고 가정하였다. 안정화 함수에서 공력계수변화에 의한 시스템 모델링 오차를 보상하기 위해 추가된 첫 번째

신경회로망에 사용된 은닉층 뉴런의 개수는 50개이다. 제어기에 사용된 두 번째 신경회로망은 800개의 은닉층 뉴런이 사용되었다. 두 번째 신경회로망에 많은 수의 은닉층 뉴런이 사용된 이유는 다음과 같다. 빠른 동특성을 갖는 각속도 운동방정식은 제어면과 크게 연성되어 있으며, 조종면 효과 감소로 인한 시스템 모델 오차는 크게 발생하게 된다. 따라서 이와 같이 큰 오차를 보상하기 위해 두 번째 신경회로망의 은닉층의 뉴런의 수가 크게 설정하였다. 시뮬레이션에 사용된 파라미터들은 다음과 같이 정하였다.

$$\gamma_{w1} = \gamma_{w2} = \gamma_{v1} = \gamma_{v2} = 30$$

$$\eta_{w1} = \eta_{w2} = \eta_{v1} = \eta_{v2} = 0.01$$

$$K_{11} = 3, K_{21} = 8$$

신경회로망의 초기 가중치 행렬값은 -0.001 과 0.001 사이에서 무작위로 설정하였다. 제안된 제어기의 효용성을 보이기 위해 공력계수의 불확실성과 제어면 고장을 고려하지 않은 백스테핑 기법만으로 설계된 제어기와 비교하였다. 그림 3.2.10은 제어하고자 하는 변수들의 시간반응을 보여주고 있다. 백스테핑 기법만을 사용한 경우에는 제어면 손상에 의해 원하는 궤적을 잘 추종하지 않는다. 특히, 옆미끄럼각의 경우 -0.7 도 정도까지 성능이 떨어짐을 알 수 있는데, 옆미끄럼각의 경우 ± 0.15 도를 벗어나면 좋지 않은 영향이 발생하는 것으로 알려져 있다. 이에 반해 본 연구에서 제안한 제어기의 경우, 오차를 보상하기 위해 신경회로망이 추가되어 초기의 학습단계를 지나면 원하는 궤적을 잘 추종함을 알 수 있다. 또한, 옆미끄럼각도 ± 0.15 도 범위 내에 있음을 알 수 있다. 그림 3.2.11는 제어면 시간반응을 보여 주고 있다. 러더각의 경우 다소 높은 각을 보여주고 있는데, 이는 본 시뮬레이션에서 가정한 제어면 손상과 명령궤적의 시나리오가 상당히 어려운 경우이기 때문에 나타나는 현상으로 보여 진다. 본 연구에서 사용된 F-16 항공기의 제어면이 운용되는 범위는 표 3.2.1에 정리되어 있다.

표 3.2.1 제어면 운용범위

Lower Bound	Control Variables	Upper bound
-25°	δ_e	25°
-21.5°	δ_a	21.5°
-30°	δ_r	30°

표 3.2.1과 제어면 시간궤적을 보여주는 그림 3.2.11를 비교해 보면, 시뮬레이션 상에 가정된 어려운 시나리오 경우임에도 항공기 제어면이 운용 가능한 영역 내에 존재함을 알 수 있다. 그림 3.2.12는 사용된 신경회로망의 가중치의 Frobenius Norm의 시간반응을 보이고 있다. 앞에서 언급한 것처럼 두 번째 신경회로망의 가중치 Frobenius Norm의 값이 첫 번째 신경회로망보다 큼을 알 수 있다.

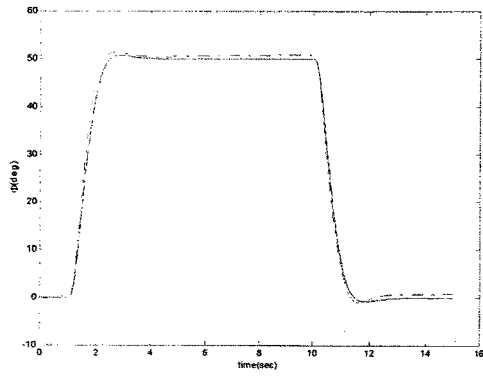
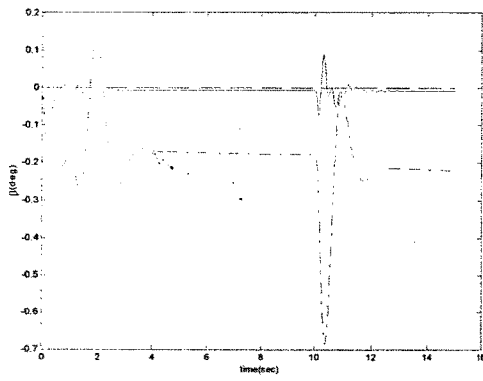
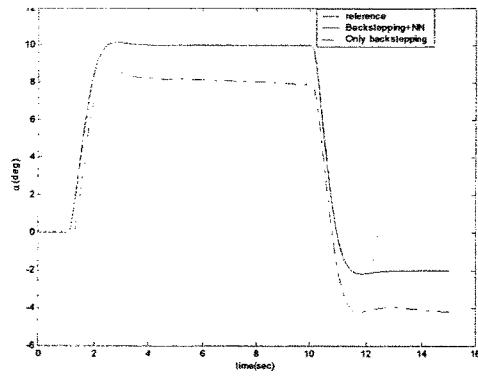


그림 3.2.10 상태변수 시간 반응

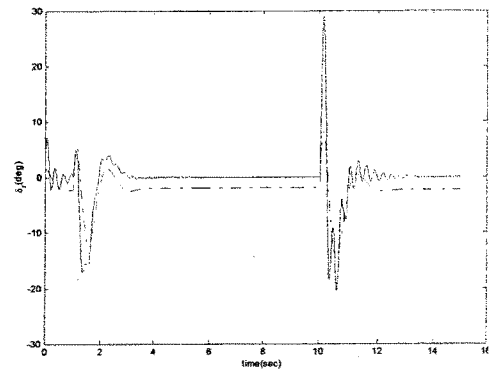
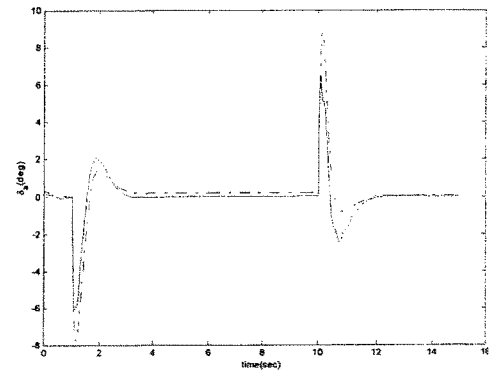
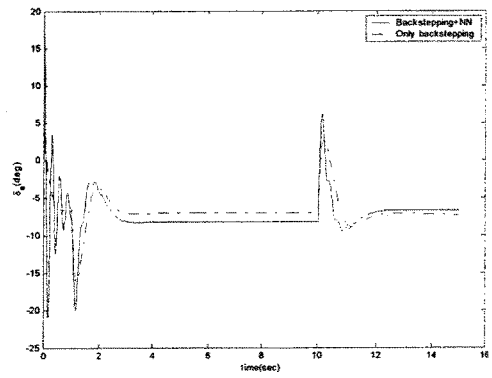


그림 3.2.11 제어변수 시간반응

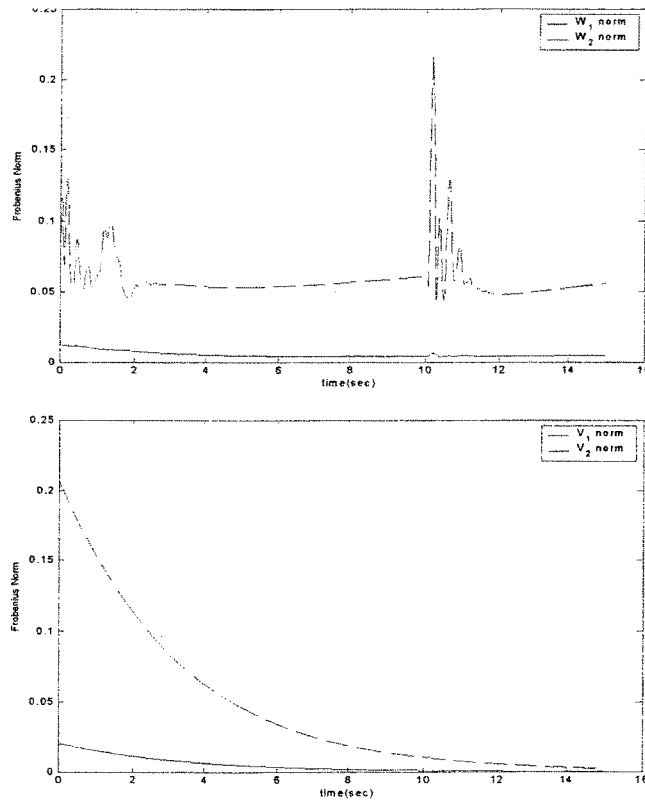


그림 3.2.12 신경회로망 가중치 Frobenius Norm의 시간반응

바. 결론

본 연구에서는 시스템의 모든 운동방정식을 정확하게 알지 못한다는 가정 하에 신경회로망을 기반으로 한 비선형 적응제어기를 제안하였다. 제안한 제어기법은 백스테핑 기법과 궤환선형화기법을 이용하였으며 공력계수의 불확실성과 조종면 손상으로 인한 시스템 모델 오차를 보상하기 위해 신경회로망을 비선형 제어기에 추가하였다. 또한 백스테핑 기법을 적용하기 위해 미분가능한 추종오차 비례 이득값을 도입하여, 안정화 함수가 모든 점에서 미분가능하도록 하였다. 전체시스템에 대한 안정성 해석과 신경회로망 가중치 갱신법칙을 르야프노프 안정화 이론을 통해 유도하였으며, 제안된 제어기의 성능을 보이기 위해 비선형 F-16 항공기 시스템에 적용하여 보통의 백스테핑 기법을 통한 제어기와 비교하였다. 시뮬레이션 결과 제안한 제어기를 통해 고장 감지 및 분리과정 없이, 조종면 파손에 따른 고장에 잘 대처할 수 있음을 알 수 있었다.

4. 적응제어기법을 이용한 재형상 제어시스템 설계기법 연구

본 연구에서는 모델추종 제어방법을 기반으로 하여 직접 적응기법을 사용한 새로운 재형상 비행제어 설계기법에 대한 연구를 수행하였다.

가. 적응기법을 이용한 재형상 비행제어 시스템 설계기법 비교

적응기법을 이용한 재형상 비행제어 시스템 설계기법에는 간접적 방법과 직접적 방법이 있다. 간접적 방법은 비행체의 동특성을 실시간으로 시스템 인식하여(system identification), 제어를 재설계하는 방법이다. 이 방법은 고장의 검출 및 분리과정이 필요 없기 때문에 예상하지 못한 고장에도 잘 대처할 수 있다는 장점이 있다. 그러나 간접적 방법은 시스템이 복잡해질수록 추정해야 할 매개변수가 많아지고, 적절한 비선형 모델구조를 찾아내야 하는 단점이 있다. 이에 비해서, 직접적 방법은 시스템 인식과정 없이 직접적으로 제어이득을 변화시키는 방법으로서, 간접적 방법에 비해 간단한 방법으로 재형상 비행제어를 수행할 수 있다. 그러나 현재까지 수행된 직접적 방법을 사용한 재형상 비행제어 시스템 설계 기법은 불충분한 가진 문제를 해결하기 위해서 구속항을 추가한 형태로 연구되어 왔다. 본 연구에서는 이러한 문제점을 개선하기 위한 연구를 수행하였다.

나. 운동방정식

본 연구에서 고려한 항공기의 비선형 6자유도 운동방정식은 다음과 같다. 항공기는 강체이며, 일정한 크기의 중력과 추력이 작용한다고 가정하였으며, 지구자전에 의한 영향은 고려하지 않았다.

$$\begin{aligned} \dot{V} = & \frac{\cos \alpha \cos \beta}{m} (T + F_x) + \frac{\sin \beta}{m} F_y + \frac{\sin \alpha \cos \beta}{m} F_z \\ & + g [-\cos \alpha \cos \beta \sin \theta + \sin \beta \sin \phi \cos \theta + \sin \alpha \cos \beta \cos \phi \cos \theta] \end{aligned} \quad (3.2.33)$$

$$\begin{aligned} \dot{\alpha} = & \cos \alpha \tan \beta p + \sin \alpha \tan \beta r - \frac{\sin \alpha}{mV \cos \beta} (T + F_x) + \frac{\cos \alpha}{mV \cos \beta} F_z \\ & + \frac{g}{V \cos \beta} (\sin \alpha \sin \theta + \cos \alpha \cos \phi \cos \theta) \end{aligned} \quad (3.2.34)$$

$$\begin{aligned} \dot{\beta} = & \sin \alpha p - \cos \alpha r - \frac{\cos \alpha \sin \beta}{mV} (T + F_x) + \frac{\cos \beta}{mV \cos \beta} F_y + \frac{\sin \alpha \sin \beta}{mV} F_z \\ & + \frac{g}{V} (\cos \alpha \sin \beta \sin \theta + \cos \beta \cos \theta \sin \phi - \sin \alpha \sin \beta \cos \phi \cos \theta) \end{aligned} \quad (3.2.35)$$

$$\Gamma \dot{p} = I_{zz} L + I_{xz} N + I_{xz} (I_{xx} - I_{yy} + I_{zz}) pq + [I_{zz} (I_{yy} - I_{zz}) - I_{xz}^2] qr \quad (3.2.36)$$

$$I_{yy} \dot{q} = M + (I_{zz} - I_{xx}) pr + I_{xz} (r^2 - p^2) \quad (3.2.37)$$

$$\Gamma \dot{r} = I_{xz} L + I_{xx} N + [I_{xx} (I_{xx} - I_{yy}) + I_{xz}^2] pq - I_{xz} (I_{xx} - I_{yy} + I_{zz}) qr \quad (3.2.38)$$

$$\dot{\phi} = p + \tan \theta (q \sin \phi + r \cos \phi) \quad (3.2.39)$$

$$\dot{\theta} = \cos \phi q - \sin \phi r \quad (3.2.40)$$

$$\dot{\varphi} = \frac{\sin \phi q + \cos \phi r}{\cos \theta} \quad (3.2.41)$$

여기서 $\Gamma = I_{xx}I_{zz} - I_{xz}^2$ 이다. 항공기의 공력계수들은 다음과 같이 표현된다.

$$F_x = [C_x(\alpha) + C_{x\delta_e} + \frac{cq}{2V} C_{x_q}(\alpha)]qS \quad (3.2.42)$$

$$F_y = [C_y + C_{y\delta_a} \delta_a + C_{y\delta_r} \delta_r + \frac{bq}{2V} C_{y_p}(\alpha) + \frac{br}{2V} C_{y_r}(\alpha)]qS \quad (3.2.43)$$

$$F_z = [C_z(\alpha, \beta) + C_{z\delta_e} \delta_e + \frac{cq}{2V} C_{z_q}(\alpha)]qS \quad (3.2.44)$$

$$L = [C_l(\alpha, \beta) + C_{l\delta_a}(\alpha, \beta) \delta_a + C_{l\delta_r}(\alpha, \beta) \delta_r + \frac{bp}{2V} C_{l_p}(\alpha) + \frac{br}{2V} C_{l_r}(\alpha)]qSb \quad (3.2.45)$$

$$M = [C_m(\alpha) + C_{m\delta_e}(\alpha) \delta_e + \frac{cq}{2V} C_{m_q}(\alpha)]qSc \quad (3.2.46)$$

$$N = [C_n(\alpha, \beta) + C_{n\delta_a}(\alpha, \beta) \delta_a + C_{n\delta_r}(\alpha, \beta) \delta_r + \frac{bp}{2V} C_{n_p}(\alpha) + \frac{br}{2V} C_{n_r}(\alpha)]qSb \quad (3.2.47)$$

다. 제어시스템 구조 및 고장 모델링

항공기 제어시스템에서의 변수는 동역학의 빠른 정동에 따라 내부루프 및 외부루프로 나누어 제어를 설계할 수 있다. 이러한 연속적 제어기의 형상을 그림 3.2.13에 제시하였다. 각속도 변수(p,q,r)는 내부루프를 구성하며, 롤각(ϕ), 받음각(α), 옆미끄럼짐각(β)은 외부루프를 구성한다.

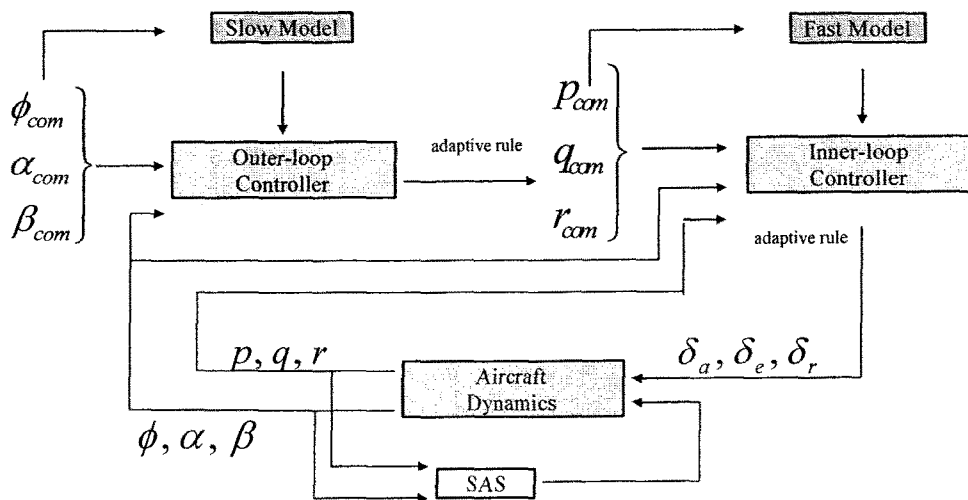


그림 3.2.13 연속적 구조의 항공기 제어 시스템

본 연구에서는 다항식과 랜덤 프로세스의 합으로 표현할 수 있는 구동기의 고장을 고려

하였다. 이러한 고장의 모델은 다음과 같이 표현할 수 있다.

$$\delta_{input} = F(\delta_{com}) = b_0 + b_1\delta_{com} + b_2\delta_{com}^2 + \dots + n(t) \quad (3.2.48)$$

여기서 $b=[b_0 \ b_1 \ b_2 \ \dots]$ 는 다항식 계수로 이루어진 벡터이며, $n(t)$ 는 랜덤 프로세스이다. 위의 고장 모델링을 그림 3.2.14에 도시하였다.

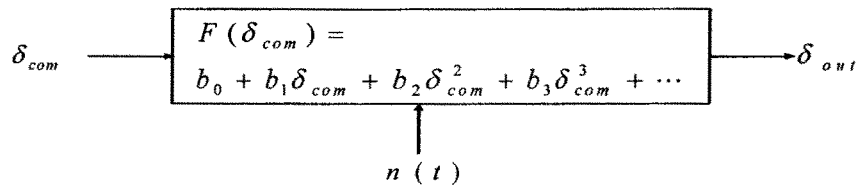


그림 3.2.14 고장모델링

라. 직접법 및 모델추종기법을 이용한 재형상 비행제어시스템 설계

본 절에서는 직접적 방법 및 모델추종 제어기법을 이용하여 본 연구에서 제안한 재형상 제어시스템을 설명하도록 한다.

모델추종 제어기

비선형 항공기의 운동방정식은 비행조건에 따라 다음과 같이 선형화할 수 있다.

$$\dot{x} = Ax + Bu + d \quad (3.2.49)$$

$$y = C\dot{x} \quad (3.2.50)$$

식(4.4.17)에서 A, B, d 는 비행영역에 따라 바뀌는 값으로 정확하게 알 수는 없다고 가정하였다. 본 연구에서는 직접 적응기법을 이용하여 변화하는 비행환경에 적응하도록 제어이득을 갱신시킨다. 그리고 입력행렬 B 및 출력행렬 C 에 대하여 다음 식이 성립한다고 가정한다.

$$\det(CB) \neq 0 \quad (3.2.51)$$

모델추종제어를 위한 모델 동역학은 다음과 같이 설정하였다.

$$\dot{y}_m = A_m y_m + B_m r \quad (3.2.52)$$

여기서 B_m 은 역행렬이 존재하도록 설정한다. 이제 모델 추종을 위한 다음과 같은 형태의

제어기를 제안한다.

$$u = C_0 r + C_0 G_0 x + C_0 v + C_0 K_0 y_m \quad (3.2.53)$$

모델추종 오차를 $e = y - y_m$ 으로 정의하자. 식 (3.2.53)을 시스템 방정식인 식 (3.2.49)에 대입한 후 식 (3.2.52)을 사용하여 모델추종 오차방정식을 구해보면 다음과 같다.

$$\begin{aligned} \dot{e} &= \dot{y} - \dot{y}_m \\ &= CAx + CBu + Cd - A_m y_m - B_m r \\ &= (CA + CBC_0 G_0)x + CBC_0 r + CBC_0 v + CBC_0 K_0 y_m + Cd - A_m y_m - B_m r \end{aligned} \quad (3.2.54)$$

식 (3.2.54)로부터 다음과 같은 완벽한 모델추종을 위한 조건을 얻을 수 있다.

$$CA + CBC_0^* G_0^* = A_e C \quad (3.2.55)$$

$$CBC_0^* = B_m \quad (3.2.56)$$

$$CBC_0^* v^* = -Cd \quad (3.2.57)$$

$$CBC_0^* K_0^* = A_m - A_e \quad (3.2.58)$$

여기서 상첨자 *는 정확한 매개변수 값을 의미하며, 실제 제어기에서는 추정해야 하는 값이다. 식 (3.2.56)와 식 (3.2.58)을 이용하면 다음 식을 얻을 수 있다.

$$K_0 = K_0^* = B_m^{-1}(A_m - A_e) \quad (3.2.59)$$

여기서 A_e 는 임의의 안정한 행렬이다. 위 제어기를 이용하면 추종오차 방정식은 $\dot{e} = A_e e$ 의 형태로 된다. 식 (3.2.55)-(3.2.59)을 이용하여 모델추종 오차방정식을 정리하면 다음과 같다.

$$\begin{aligned} \dot{e} &= A_e e + [CA + CB(C_0^* + C_0 - C_0^*)G_0 - CA - CBC_0^* G_0^*]x \\ &\quad + CB(C_0 - C_0^*)r + CBC_0^* v + CB(C_0 - C_0^*)v \\ &\quad + [CB(C_0^* + C_0 - C_0^*)K_0 + A_e - A_m]y_m - CBC_0^* v^* \\ &= A_e e + CBC_0^*(G_0 - G_0^*)x + CB(C_0 - C_0^*)(G_0 x + r + v + K_0 y_m) \\ &\quad + (CBC_0^* K_0 + A_e - A_m)y_m + CBC_0^*(v - v^*) \end{aligned} \quad (3.2.60)$$

위 식을 좀 더 정리하면 다음과 같다.

$$\dot{e} = A_e e + B_m(G_0 - G_0^*)x + CBC_0^*(C_0 r + C_0 G_0 x + C_0 v + C_0 K_0 y_m) + B_m(v - v^*) \quad (3.2.61)$$

오차행렬 및 오차벡터는 다음과 같이 정의하자.

$$\Delta G = G_0 - G_0^* \quad (3.2.62)$$

$$\Delta \psi = C_0^{*-1} - C_0^{-1} \quad (3.2.63)$$

$$\Delta v = v_0 - v_0^* \quad (3.2.64)$$

식 (3.2.62)-(3.2.64)를 식 (3.2.61)에 대입하여 정리하면 다음과 같다.

$$\dot{e} = A_e e + B_m \Delta G x + B_m \Delta \psi u + B_m \Delta v \quad (3.2.65)$$

적응규칙 유도

모델추종 오차방정식을 이용하여 제어이득에 대한 적응법칙을 구하기 위해서 다음과 같은 르야프노프 후보함수를 선택하였다.

$$V = e^T P e + tr \left[\frac{\Delta G^T \Delta G}{\gamma_1} \right] + tr \left[\frac{\Delta \psi^T \Delta \psi}{\gamma_2} \right] + \frac{\Delta v^T \Delta v}{\gamma_3} \quad (3.2.66)$$

여기서 행렬 P 는 다음의 방정식을 만족한다.

$$A_e^T P + P A_e = -Q, \quad Q > 0 \quad (3.2.67)$$

식 (3.2.66)를 시간에 대해 미분하면 다음과 같다.

$$\dot{V} = 2e^T P \dot{e} + 2tr \left[\frac{\Delta G^T \dot{\Delta G}}{\gamma_1} \right] + 2tr \left[\frac{\Delta \psi^T \dot{\Delta \psi}}{\gamma_2} \right] + 2 \frac{\Delta v^T \dot{\Delta v}}{\gamma_3} \quad (3.2.68)$$

식(4.4.33)을 식 (4.4.36)에 대입한 뒤 정리하면 다음 식을 얻을 수 있다.

$$\begin{aligned} \dot{V} = & e^T (P A_e + A_e^T P) e + 2e^T P (B_m \Delta G x + B_m \Delta \psi u + B_m \Delta v) \\ & + 2tr \left[\frac{\Delta G^T \dot{\Delta G}}{\gamma_1} \right] + 2tr \left[\frac{\Delta \psi^T \dot{\Delta \psi}}{\gamma_2} \right] + 2 \frac{\Delta v^T \dot{\Delta v}}{\gamma_3} \end{aligned} \quad (3.2.69)$$

식 (3.2.67)를 이용하면 다음 식을 얻을 수 있다.

$$\begin{aligned} \dot{V} = & -e^T Qe + 2tr \left[\frac{1}{\gamma_1} \Delta G^T (\Delta \dot{G} + \gamma_1 B_m^T Pex^T) + \frac{1}{\gamma_2} \Delta \psi^T (\Delta \dot{\psi} + \gamma_2 B_m^T Peu^T) \right] \\ & + \frac{1}{\gamma_3} \Delta v^T (\Delta \dot{v} + \gamma_3 B_m^T Pe) \end{aligned} \quad (3.2.70)$$

위 식 (3.2.70)을 유도하는 과정에는 다음의 행렬에 관한 방정식이 사용되었다.

$$e^T P B_m \Delta G x = tr \left[\Delta G^T B_m^T P e x^T \right] \quad (3.2.71)$$

$$e^T P B_m \Delta \psi u = tr \left[\Delta \psi^T B_m^T P e u^T \right] \quad (3.2.72)$$

르야프노프 후보함수의 미분값인 \dot{V} 가 음 또는 0이 되기 위해서 시불변 시스템의 가정하에 다음과 같은 적응규칙을 정의할 수 있다.

$$\dot{G}_0 = -\gamma_1 B_m^T P e x^T \quad (3.2.73)$$

$$\dot{C}_0 = -\gamma_2 C_0 B_m^T P e u^T C_0 \quad (3.2.74)$$

$$\dot{v} = -\gamma_3 B_m^T P e \quad (3.2.75)$$

식 (3.2.74)를 유도하는 과정에는 다음의 행렬 관계식이 사용되었다.

$$\dot{C}_0^{-1} = -C_0^{-1} \dot{C}_0 C_0^{-1} \quad (3.2.76)$$

식 (3.2.73)-(3.2.75)을 식 (3.2.70)에 대입하면 다음을 확인할 수 있다.

$$\dot{V} = -e^T Qe < 0 \quad (3.2.77)$$

제안한 적응규칙의 성질을 다음의 정리에서 요약하였다.

정리 3.2.1: 시불변 시스템에서 입력이 제한될 때, 식 (3.2.53)의 제어법칙과 식 (3.2.73)-(3.2.75)의 적응법칙은 출력신호의 크기가 제한됨과 $\lim_{t \rightarrow \infty} e(t) = 0$ 을 보장한다.

증명: 식(3.2.77)로부터 V 는 증가하지 않으므로 다음 식을 만족한다.

$$0 \leq V(t) \leq V(0) \quad (3.2.78)$$

따라서 $V(t) \in L_\infty$ 가 성립하고, $V(t)$ 는 radially unbounded 되었기 때문에 다음식이 성립한다.

$$\Delta G, \Delta \psi, \Delta v \in L_\infty \quad (3.2.79)$$

식 (3.2.78)에서 출력신호 크기의 제한이 보장된다. 즉 $y \in L_\infty$ 가 성립한다. 또한, 식 (3.2.78)으로부터 다음이 성립한다.

$$\int_0^\infty e^T Q e dt = - \int_0^\infty \dot{V} dt = V(0) - V(\infty) < \infty \quad (3.2.80)$$

행렬 Q 가 양의 한정행렬이므로 식(3.2.77)와 식(3.2.80)로부터 $e \in L_2$ 가 성립한다. 또한, 입력이 제한되면 식(3.2.65)으로부터 $\dot{e} \in L_\infty$ 도 성립한다. 최종적으로 $e, \dot{e} \in L_\infty, e \in L_2$ 로부터 $\lim_{t \rightarrow \infty} e = 0$ 이 성립한다.

마. 수치 시뮬레이션 및 결과

제안한 재형상 비행제어 법칙의 유용성을 확인하기 위하여 수치 시뮬레이션을 수행하였다. 수치 시뮬레이션에서 사용한 모델은 비선형 6 자유도 F-16 운동방정식이다. 비행환경은 고도 2,000ft에서의 속도 523 ft/sec 상태로 선정하였다. 내부루프를 위한 모델 방정식은 다음과 같이 설정하였다.

$$\dot{p} = -3p + 3p_c \quad (3.2.81)$$

$$\dot{q} = -3q + 3q_c \quad (3.2.82)$$

$$\dot{r} = -3r + 3r_c \quad (3.2.83)$$

외부루프를 위한 모델 방정식은 내부루프의 동특성보다 느리게 다음과 같이 설정하였다.

$$\dot{\phi} = -\phi + \phi_c \quad (3.2.84)$$

$$\dot{\alpha} = -\alpha + \alpha_c \quad (3.2.85)$$

$$\dot{\beta} = -\beta + \beta_c \quad (3.2.86)$$

내부루프와 외부루프를 위한 적응규칙에서의 제어이득은 다음과 같이 선정하였다.

outer-loop adaptive gains:

$$\gamma_1 = \gamma_2 = \gamma_3 = 2$$

inner-loop adaptive gains:

$$\gamma_1 = \gamma_2 = \gamma_3 = 5$$

수치 시뮬레이션에서 고려한 기동은 일정한 받음각을 유지하면서 30도의 롤기동을 수행하는 것으로 설정하였으며 이를 그림 3.2.15에 제시하였다. 고장은 롤기동 중에 에일러론에 50%의 효율감소와 3도의 바이어스 입력이 주어지는 것으로 가정하였다.

$$\phi_c = \begin{cases} 0, & 0 \leq t < 5 \text{ sec} \\ 30 \text{ deg}, & 5 \leq t < 10 \text{ sec} \\ 0, & 20 \leq t \end{cases}$$

$$\alpha_c = 4 \text{ deg}, \text{ for all } t$$

$$\beta_c = 0, \text{ for all } t$$

$$F(\delta_a) = 3 \text{ deg} + 0.5 \delta_{a,com} \quad (t \geq 27 \text{ sec})$$

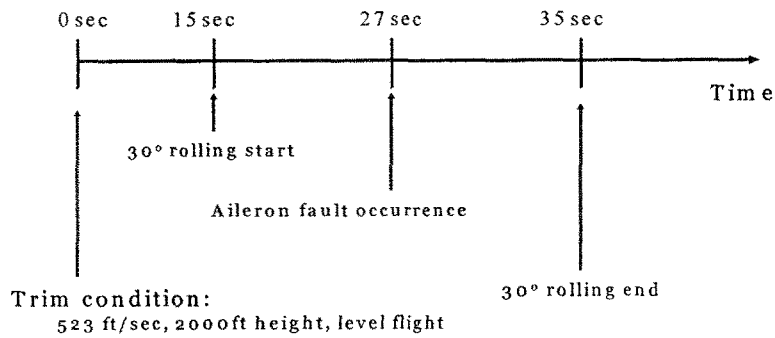


그림 3.2.15 시뮬레이션 시나리오

그림 3.2.16에서 3.2.18까지는 외부루프의 상태변수의 반응을 그림 3.2.19에서 3.2.21까지는 외부루프의 상태변수 반응을 나타내었다. 그림에서 실선은 실제 시스템의 출력을 점선은 모델의 출력을 나타낸다. 그림 3.2.22에서 3.2.24까지는 설계된 제어기로부터 생성되는 제어입력 명령값을 도시하였다.

내용누락

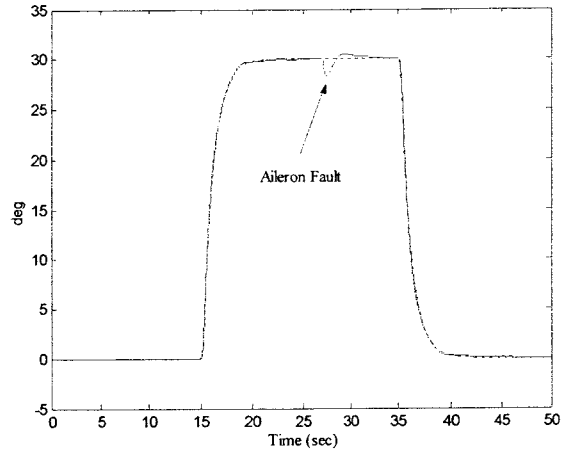


그림 3.2.19 Roll angle response

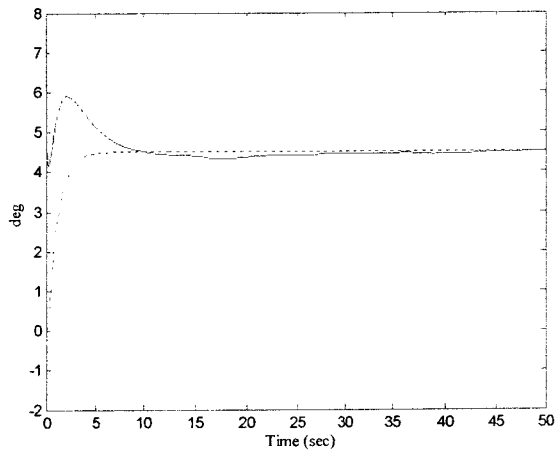


그림 3.2.20 Angle of attack response

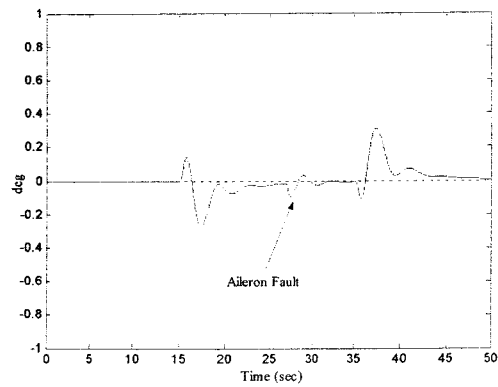


그림 3.2.21 Sideslip angle response

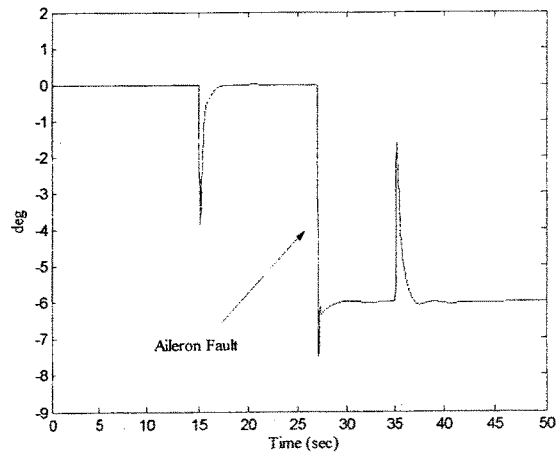


그림 3.2.22 Aileron command history

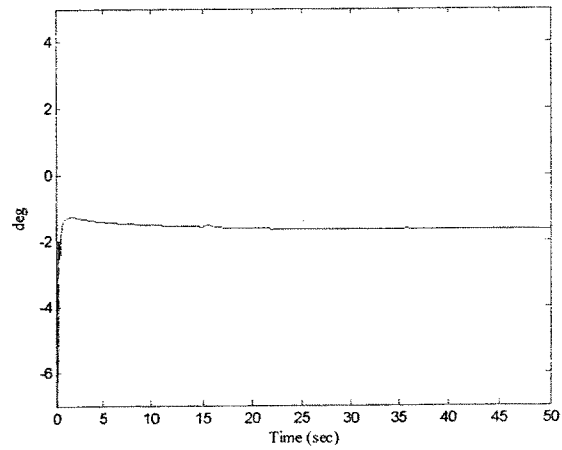


그림 3.2.23 Elevator command history

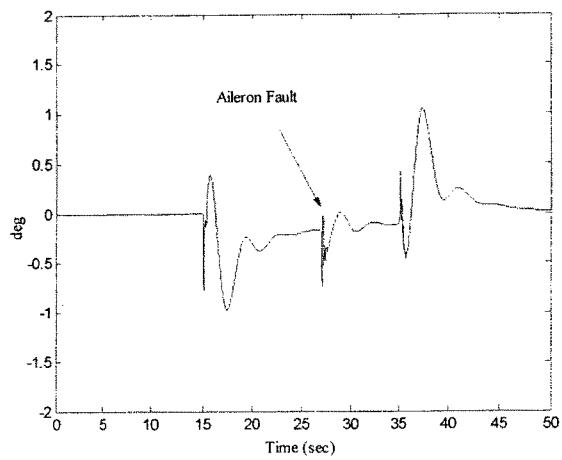


그림 3.2.24 Rudder command history

5. 슬라이딩 모드기법 기반 재형상 제어시스템 설계기법 연구

재형상 비행제어시스템은 비행기의 작동기에 고장이 발생하거나 조종면에 파손이 발생하였을 경우에 자동으로 시스템을 재설계하여 이를 보상함으로써 비행기가 보다 나은 생존성을 가지도록 한다. 본 연구에서는 항공기 운동특성을 빠른 동특성을 가지는 변수와 느린 동특성을 갖는 변수로 분리하고, 이 시스템에 대해 모델추종 제어기법과 슬라이딩 모드 제어기법을 이용한 재형상 제어시스템을 제안하였다. 시스템에 고장이나 외란이 발생하였을 경우에 슬라이딩 추정기를 이용하여 고장을 검출하는 강건한 고장탐지 기법을 제안하고, 고장 검출방식을 통해 고장을 보상할 수 있는 재형상 시스템을 설계하였다.

가. 시스템 운동방정식

항공기의 횡운동 운동방정식을 평형상태에 대해 선형화하여 얻은 선형시스템 운동방정식은 다음과 같다.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.2.87}$$

$$x^T = [x_1^T \ x_2^T], \quad u = [\delta_a, \delta_r]^T$$

여기서 x_1 은 느린 동특성을 갖는 변수이고 x_2 는 빠른 동특성을 갖는 변수이다. 그리고 시스템행렬은 동특성에 따라 다음과 같이 세분하여 표현할 수 있다.

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad C = I$$

나. 모델 추종제어 기법의 적용

앞에서 설정한 모델에 대해서 상태변수의 동특성에 따라 느린 동특성과 빠른 동특성을 갖는 변수로 분리하였다. 느린 동특성을 갖는 변수에 대해서는 모델추종 제어기법을 적용시켜 빠른 동특성을 갖는 변수에 적용하는 기준값을 설정하였다. 느린 동특성을 갖는 부분에 대해 모델추종 제어기법을 적용하기 위한 시스템 방정식을 구성하면 다음과 같다.

$$\begin{aligned} \dot{x}_s &= A_1 x_s + A_{12} u_s + d \\ y &= C x_s \end{aligned} \tag{3.2.88}$$

느린 동특성을 가지는 부분의 변수는 항공기의 횡운동에 관한 변수를 고려하므로 다음과 같이 설정될 수 있다.

$$x_k = \begin{bmatrix} \beta \\ \phi \end{bmatrix}, \quad u_k = \begin{bmatrix} p \\ r \end{bmatrix}, \quad C = I, \quad d = B_1$$

모델추종제어기법을 위한 모델은 다음과 같이 설정한다.

$$\dot{x}_m = A_m x_m + B_m r \tag{3.2.89}$$

여기서 상태변수 x_m 은 $[B_m, \phi_m]$ 이고 행렬 A_m 을 안정하도록 설정하여야 한다. 모델추종 오차방정식은 다음과 같다.

$$\dot{e} = \bar{x} - \bar{x}_m \quad (3.2.90)$$

강건한 특성을 보장하기 위해 모델의 출력을 외연적으로 추가한 형태의 제어기를 제안하였고 제어기의 형태는 다음과 같다.

$$u = C_0 r + C_0 G_0 x_k + C_0 v + C_0 K_0 x_m \quad (3.2.91)$$

위 식을 시스템 방정식에 대입한 후 모델추종 오차방정식을 구해보면 다음과 같은 제어매개변수 식을 유도할 수 있다.

$$C_0 = A_{12}^{-1} C^{-1} B_m \quad (3.2.92a)$$

$$G_0 = (C_0^{-1} A_{12}^{-1} C^{-1})(A_e C - CA) \quad (3.2.92b)$$

$$K_0 = B_m^{-1}(A_m - A_e) \quad (3.2.92c)$$

$$v = -(C_0^{-1} A_{12}^{-1} C^{-1}) C d \quad (3.2.92d)$$

위 식을 식(3.2.90)과 식(3.2.91)에 대입하여 오차방정식을 구해보면 다음과 같다.

$$\dot{e} = A_e e + B_m \Delta G_0 x + B_m \psi u + B_m \Delta v \quad (3.2.93)$$

$$(\Delta G_0 = G_0 - G_0^*, \psi = C_0^{*-1} - C_0^{-1}, \Delta v = v - v^*)$$

식(3.2.90)의 오차방정식을 이용하여 제어이득에 대한 회귀적 적응 법칙을 구하기 위해서 다음과 같이 르야프노프 함수를 선택한다.

$$V = e^T P e + tr \left(\frac{\Delta G_0^T}{\gamma_1} + \frac{\psi^T \psi}{\gamma_2} \right) + \frac{\Delta v^T \Delta v}{\gamma_3} \quad (3.2.94)$$

$$A_e^T P + P A_e = -Q$$

르야프노프 함수를 미분하여 미분방정식 조건을 만족한다고 가정하면 다음과 같은 적응 법칙을 유도할 수 있다.

$$\dot{G}_0 = -\gamma_1 B_m^T P e x^T$$

$$\dot{C}_0 = -\gamma_2 C_0 B_m^T P e u^T C_0$$

$$\dot{v} = -\gamma_3 B_m^T P e$$

다. Sliding Mode 기법 적용

느린 동특성 부분에서 모델추종제어 기법을 통해 얻은 기준 입력으로 빠른 동특성 부

분에 슬라이딩 모드 기법을 적용하여 시스템을 구성하여 고장이나 외란에 대해 슬라이딩 평면 안에서 슬라이딩 모드 추정기를 통해 고장을 검출하고 재형상 시스템을 설계한다. 빠른 동특성부분에 슬라이딩모드를 적용하기 위해서 시스템 방정식을 구성하면 다음과 같다.

$$\begin{bmatrix} \dot{p} \\ \dot{y} \end{bmatrix} = [A_{21}] \begin{bmatrix} \phi \\ \beta \end{bmatrix} + [A_{22}] \begin{bmatrix} p_{ref} \\ y_{ref} \end{bmatrix} + [B_2] \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} + \hat{\gamma} \quad (3.2.95)$$

시스템의 느린 동특성 부분에서 모델추종 제어기법을 통해서 얻어진 기준 입력값을 $\begin{bmatrix} p_{ref} \\ y_{ref} \end{bmatrix}$ 로 설정하고 위의 빠른 동특성부분의 시스템방정식과 연관하여 슬라이딩 모드를 구성하면 다음과 같다.

$$S = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = \begin{bmatrix} p - p_{ref} \\ y - y_{ref} \end{bmatrix} = 0 \quad (3.2.96)$$

시스템이 슬라이딩 영역 안으로 들어가도록 르야프노프 함수를 다음과 같이 설정한다.

$$v = -\frac{1}{2} s^T s \quad (3.2.97)$$

식(3.2.96)의 슬라이딩 모드 정의를 통해서 실제 입력을 다음과 같이 설정하였다.

$$\begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} = [B_2]^{-1} \left[-[A_{21}] \begin{bmatrix} \phi \\ \beta \end{bmatrix} - [A_{22}] \begin{bmatrix} p \\ y \end{bmatrix} + \begin{bmatrix} \dot{p}_{ref} \\ \dot{y}_{ref} \end{bmatrix} - K \text{sgn} \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \right] \quad (3.2.98)$$

식(3.2.95)에 위 식을 대입하면 다음과 같은 식이 얻어진다.

$$\dot{s} = -K \begin{bmatrix} \text{sgn}[S_1] \\ \text{sgn}[S_2] \end{bmatrix} + \hat{\gamma} \quad (3.2.99)$$

식(3.2.97)에서 설정한 르야프노프 함수의 미분치인 $\dot{v} = s^T \dot{s} \leq 0$ 가 슬라이딩 평면 도달조건을 만족하도록 설정하면 다음과 같은 슬라이딩평면 도달조건이 구해진다.

$$K \begin{bmatrix} \text{sgn}[S_1] \\ \text{sgn}[S_2] \end{bmatrix} \geq \hat{\gamma} \quad (3.2.100)$$

위 식에서 슬라이딩 모드의 제어이득을 고장보다 크게 설정한다면 항상 슬라이딩평면 안에 있도록 할 수 있다. 그러므로 고장보다 큰 제어이득 값을 설정해서 고장이 발생해도 항상 슬라이딩평면 안으로 들어오도록 설정한 후 슬라이딩모드 안에서 고장을 검출하게 된다.

슬라이딩 관측기 설계

식(3.2.99)에서 S값은 식(3.2.96)으로부터 계산이 가능하지만 \dot{s} 을 계산할 수 없으므로 추정해야 하는데 입력항이 없기 때문에 일반적인 관측기를 구성할 수 없으므로 시스템이 슬라이딩모드 안에 있다는 가정 하에 슬라이딩 모드 관측기를 사용하여 \dot{s} 을 추정하여 고장을 검출하게 된다. 일반적인 슬라이딩 모드 관측기는 다음과 같이 구성할 수 있다.

$$\widehat{s}_1 = s, \quad \widehat{s}_2 = \dot{s} \quad (3.2.101a)$$

$$\dot{\widehat{s}}_1 = -L_1 \text{sign}[\widehat{s}_1 - s] \quad (3.2.101b)$$

$$\tau \dot{w}_1 = -w_1 - L_1 \text{sign}[\widehat{s}_1 - s] \quad (3.2.101c)$$

$$\dot{\widehat{s}}_2 = -L_2 \text{sign}[\widehat{s}_2 - w_1] \quad (3.2.101d)$$

여기서 L_1, L_2 는 제어이득 값이고 τ 는 저주파수의 영역을 조절하는 시정수로 사용되었으며 w_1 은 보다 부드럽게 추정하기 위한 매개변수로 사용되었다. 식(3.2.99)에서 위식의 방식으로 \dot{s} 을 추정할 수 있기 때문에 고장을 검출할 수 있게 된다.

라. 조종면 파손 모델링 및 재형상 제어

본 연구의 기법의 타당성을 검증하기 위한 고장을 적용하기 위해서 항공기의 조종면이 파손되는 심각한 고장이 발생한 경우 중에서 엘리베이터의 파손을 가정하여 본 연구가 제시한 기법으로 고장을 검출하고 재형상 시스템을 구현하였다.

수평미익에 장착된 엘리베이터에 물리적인 파손이 생겨서 고장이 발생한 이후에는 조종면의 면적이 변할 뿐만 아니라 받음각에 대한 양력계수의 변화율과 같은 공력특성들 역시 변하게 되는데 본 연구에서는 부분적인 파손을 고려하여 물리적인 파손에 대해 면적의 변화에 의한 영향이 가장 지배적이라고 가정한 후 고장의 형상을 표현하였다. 고장이 발생한 경우 항공기의 횡운동 방정식은 다음과 같이 표현된다.

$$\dot{x} = (A + \Delta A)x + (B + \Delta B)u + Ef \quad (3.2.102)$$

상태변수와 입력은 다음과 같다.

$$x = [\beta, \phi, p, r]^T, \quad u = [\delta_a, \delta_r]^T$$

본 연구에서는 엘리베이터 고장을 적용하므로 $\Delta A = 0, \Delta B = 0$ 의 조건이 만족되므로 엘리베이터 고장의 경우 다음과 같이 표현할 수 있다.

$$Ef = d_a(1 - \zeta_e) \begin{bmatrix} 0 \\ 0 \\ \mu L_{\delta_a} + \sigma_1 N_{\delta_a} \\ \mu_{\delta_a} + \sigma_2 L_{\delta_a} \end{bmatrix} \delta_e \quad (3.2.103)$$

여기서 d_s 는 x축에 대해 에일러론의 공력중심에 대한 엘리베이터의 공력중심의 비를 표현하고 ζ_s 과손된 후의 전체면적에 대한 엘리베이터 유효한 면적의 비를 나타낸다.

고장이 위와 같이 모델링 된 것을 시스템에 적용하여 본 연구에서 제안한 방법으로 고장을 검출하고 그 검출한 크기만큼 실시간으로 보상을 해주면 실시간으로 재형상 시스템이 설계되어진다. 재형상시스템 방정식은 다음과 같다.

$$\dot{x} = Ax + Bu + f - \hat{f} \quad (3.2.104)$$

마. 시뮬레이션

본 연구에서 고려하는 시스템은 F-16항공기의 운동 방정식이다. 모든 변수의 단위는 도(degree)로 사용하였다. 고려한 평형비행 상태에서 선형화된 시스템 행렬은 다음과 같다.

$$A = \begin{bmatrix} -0.1295 & 0.0698 & 0.1308 & -0.9882 \\ -34.6234 & 0 & -2.7237 & 0.8917 \\ 7.8114 & 0 & -0.0478 & -0.3850 \\ 0 & 0 & 1.0000 & 0.1317 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0068 & 0.0186 \\ 0 & 0 \\ -2.93421 & 4.5153 \\ -1.2046 & -2.4006 \end{bmatrix}$$

그리고 느린 동특성 부분에 모델추종 제어 기법을 적용하는데 사용되는 모델은 다음과 같이 설정하였다.

$$A_m = \begin{bmatrix} -1.5 & 0 \\ 0 & -1.5 \end{bmatrix}, \quad B_m = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}$$

그리고 기준입력으로는 롤 운동을 가정하였으며 롤 운동을 위한 기준값은 다음과 같다.

$$\beta_{com} = 0^\circ$$

$$\phi_{com} = \begin{cases} 0^\circ, & 0 \leq t \leq 5 \\ 10^\circ, & 5 \leq t \leq 10 \\ 0^\circ, & 10 \leq t \leq 15 \end{cases}$$

기준입력으로 5-10초 사이에 롤각이 10° 를 적용하였다. 그리고 기준입력 모델값인 r_k 와 실제 제어 입력인 $u(\delta_s, \delta_r)$ 는 1차 시간 지연 필터가 있다고 가정하였다.

$$\dot{r}_{k_filter} = -20r_{k_filter} + 20r_k$$

$$\dot{u}_{filter} = -20u_{filter} + 20u$$

그리고 슬라이딩 모드에 사용되는 sgn 함수는 다음과 같이 정의되는 3차함수로 대체하여 사용하였다.

$$sat_2(s/\delta) = \begin{cases} -1 & \text{when } s < -\delta \\ \frac{1}{-2\delta^3} (s^3 - 3\delta^2 s) & \text{when } -\delta < s < \delta \\ 1 & \text{when } s > \delta \end{cases}$$

$\delta=0.1$ 로 크게 설정해서 대부분의 입력이 boundary 안에 포함될 수 있도록 설정하였고 마지막으로 엘리베이터 고장의 형상을 모델링에 대한 변수는 다음과 같이 설정하였다.

$$d_a = \frac{\text{distance of elevator c.g from x axis}}{\text{distance of aileron c.g from x axis}} = 2.5$$

ζ_e = 파손된 후의 전체면적에 대한 엘리베이터 유효한 면적의 비=0.3

$$\mu = \frac{J_z J_x}{\Gamma}, \quad \sigma_1 = \frac{J_z J_{xz}}{\Gamma}, \quad \sigma_2 = \frac{J_x J_{xz}}{\Gamma} \quad (\Gamma = J_x J_z - J_{xz}^2)$$

위의 수치예제를 통해 시뮬레이션을 수행하였다. 시뮬레이션에서 고려한 엘리베이터 고장은 한쪽 엘리베이터 면이 파손됐을 경우에 대해 2가지로 시뮬레이션을 수행하였다. 그림 3.2.25에서는 고장에 대한 입력이 상수값으로 들어가는 경우에 대해 고장의 입력의 형상을 보여주고 있다. 이것은 물리적으로 한쪽 엘리베이터 면이 파손된 형태를 그대로 유지하고 있는 경우라고 볼 수 있다. 그림2에서는 그림 3.2.25의 고장을 본 연구에서 제안한 기법으로 고장을 검출한 것이다. 그림 3.2.25과 그림 3.2.26를 비교해 보면 거의 고장을 잘 검출하고 있다는 것을 확인 할 수 있다. 5초와 10초에서 튀는 것은 기준입력 값이 영향을 미치기 때문이다. 그림 3.2.27과 그림 3.2.28는 고장이 그림 3.2.25과 같이 들어간 경우에 대해서 본 연구에서 제안한 기법으로 재형상 제어를 수행한 결과를 나타낸 것으로, 재형상제어가 잘 수행되고 있음을 확인 할 수 있다. 그림 3.2.29는 고장의 입력이 sin 값으로 들어가는 경우에 대해서 고장의 형상을 묘사한 것이고, 그림 3.2.31은 그림 3.2.30의 고장의 형상에 대해서 본 연구에서 제안한 기법으로 고장을 검출한 그래프로 고장이 상수입력이 들어갔을 때와 마찬가지로 고장이 잘 검출되고 있음을 확인 할 수 있다. 그림 3.2.31과 그림 3.2.32은 고장이 그림 3.2.29와 같이 들어간 경우 본 연구가 제안한 기법으로 재형상 제어를 수행한 결과를 나타낸 것으로 재형상 제어가 잘 수행되고 있음을 확인 할 수 있다.

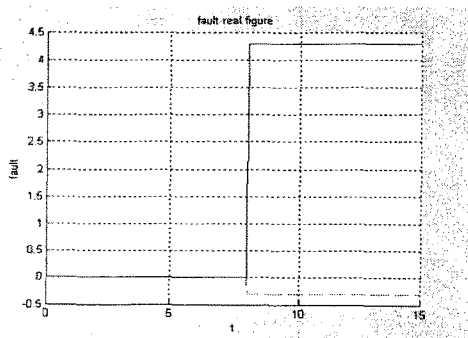


그림 3.2.25 상수입력 형태의 고장

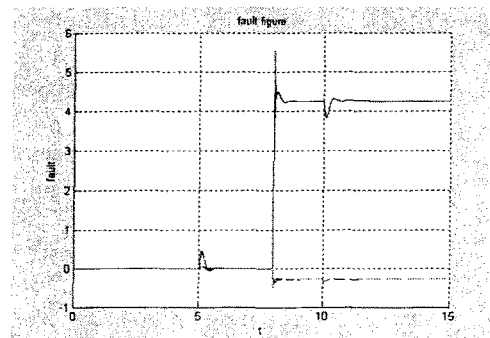


그림 3.2.26 검출된 고장의 형상

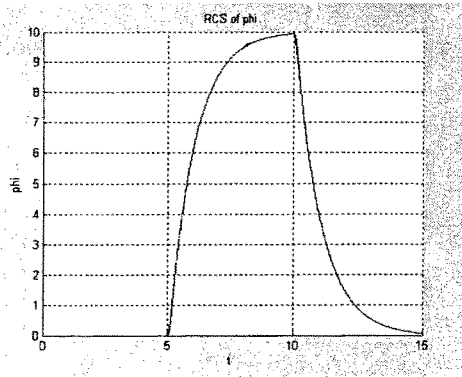


그림 3.2.27 수행한 phi

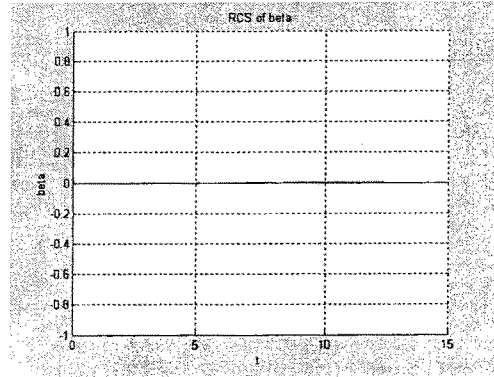


그림 3.2.28 RCS 수행한 beta

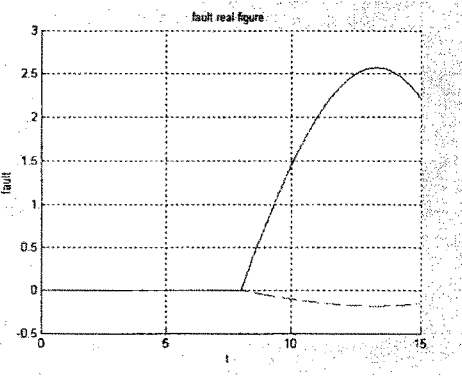


그림 3.2.29 sin입력 형태의 고장

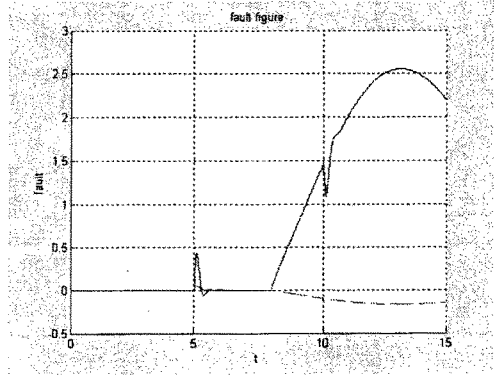


그림 3.2.30 검출된 고장의 형상

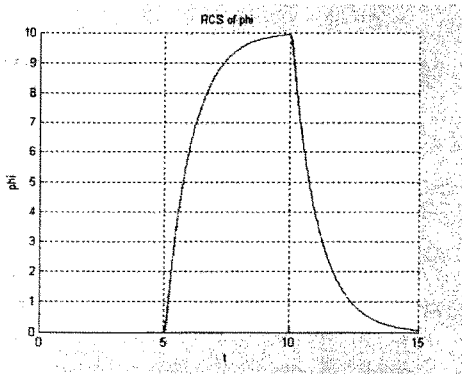


그림 3.2.31 RCS 수행한 phi

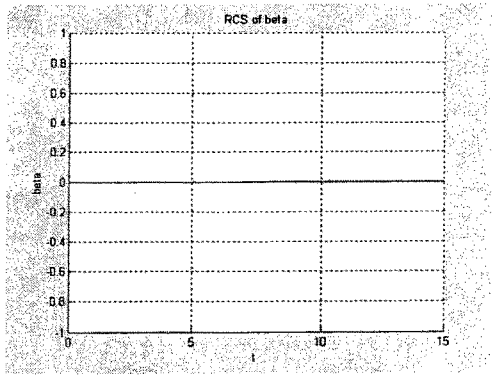


그림 3.2.32 RCS 수행한 beta

6. 모드전환 비행제어 기법 연구

가. 개요

다중 모드 제어는 대략적으로 서로 다른 임무를 수행하기 위해 미리 설계된 제어기의 집합이라고 정의할 수 있다. 이때 각각의 작동 모드는 비행체의 운동방정식과 임무환경에 의해 결정되며, 설정된 임무사이에서 전환 시 적절한 제어기법이 적용되어야 한다. 일반적으로 흔히 사용되는 모드 전환 기법 가운데 하나가 게인스케줄링 기법이다. 그림 3.2.33은 게인 스케줄링의 대략적인 블록선도를 나타내고 있다.

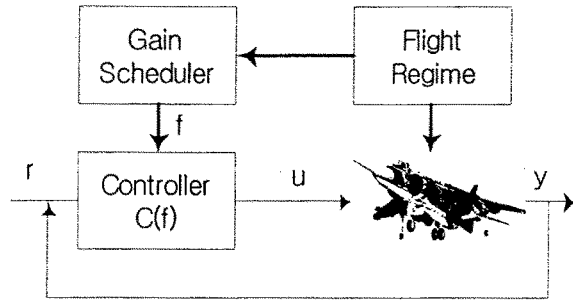


그림 3.2.33 게인 스케줄링 개략도

나. 연구방법

비행체에 적용되는 모드전환 기법은 시스템의 전체적 안정성을 보장할 수 없고, 특히 예측되지 않은 외란이나 고장상황에 대해 대처 능력이 떨어지는 것으로 알려져 있다. 따라서 이와 같은 점을 극복하기 위해 고장검출 및 진단에 대한 선행 연구가 필수적으로 요구된다. 불가피한 상황이 발생하여, 예측된 고장 이외의 사고가 발생하였을 경우라도 안정적인 시스템 운영을 위해 시스템 재형상화가 가능한 모드로의 전환도 연구 대상이 된다. 이러한 목적을 위해 지능 비행제어시스템을 설계한다. 지능 비행제어시스템은 다중화 관리, 모드 예측, 외란 감지 및 제거, 고장검출 및 진단, 시스템 변수 관측, 의사결정 알고리즘 등의 하부 구조를 갖게 된다.

모델예측제어 (Model Predictive Control)

예측모델이란 시스템의 미래 상태변수를 추정하기 위한 수학적 모델을 의미하며, 이러한 예측모델은 주어진 정상 시스템을 고려한 모델이거나 또는 다른 수학적 모델로 설정할 수 있다. 외란이나 불확실성의 크기가 작은 경우에는 주어진 시스템을 예측모델로 사용하는 것으로 충분하며, 그림 3.2.34에 일반적인 예측제어 기법이 도시되어 있다.

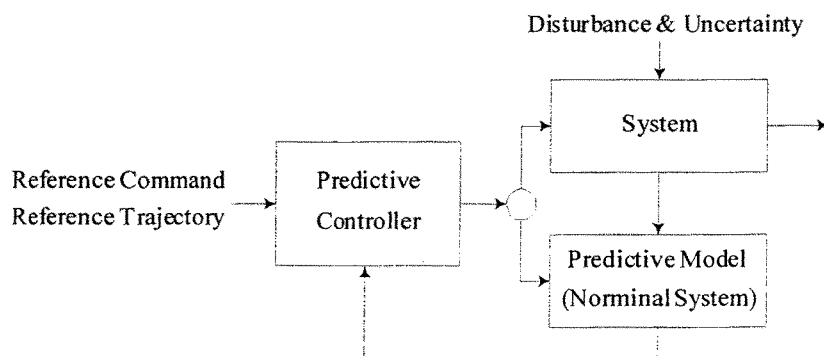


그림 3.2.34 예측제어의 블록선도

다중모델, 전환 및 조정 (Multiple Models, Switching, Tuning)

MMST기법을 기반으로 하는 알고리즘은 시스템에 고장이 발생한 경우, 시스템의 재형상 및 시스템 성능에서 단일 모델에 의한 적응제어기보다 안정적이다.

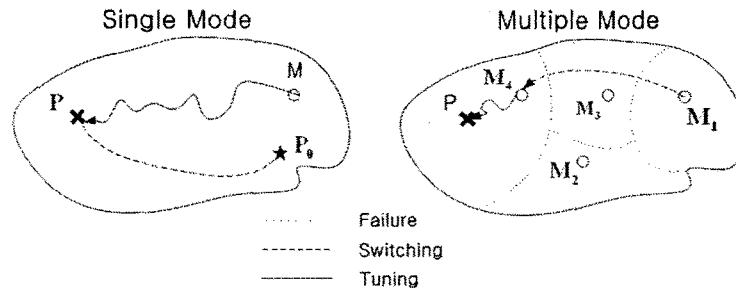


그림 3.2.35 단일 모델과 다중모델의 적응방식 비교

그림 3.2.35은 단일모델과 다중모델의 적응방식의 차이점을 보이고 있다. 위의 그림에서 볼 수 있는 것처럼, 고장이 발생하면 모델이 급변하여 정상사태에 있는 P_0 점이 고장이 발생한 점인 P 점으로 전환된다. 이때 단일 모델을 사용하게 되면 M 점의 모델이 P 점으로 적응하는데 오랜 시간과 시행착오를 거치게 된다. 반면 다중화 모델을 사용하게 되면, 고장이 발생한 지점의 모델로 이동한 후, 고장점으로 적응하게 되므로 단일 모드에 비해 안정적이며, 적응시간도 빠르게 된다.

통합 전환 제어기를 이용한 다중 모드 알고리즘 설계

항공기는 그 시스템의 복잡성과 운영 환경의 변화에 따라 운동특성이 변하므로, 고장 발생시 상이한 운동방정식을 갖는 여러 다른 모드를 갖는다. 이와 같이 다양한 모드를 가지므로, 모드에서 모드로 전환하는 천이 상태를 처리하는 적절한 알고리즘을 구축하는 것이 필요하다.

이를 위해 역운동역학제어(Inverse Dynamics Control law) 알고리즘과 모델예측제어 기법이 사용된다. IDCL 알고리즘은 적용하기에 간단하고 계산도 단순하지만, 물리적으로 의미 없는 수치결과를 도출하여 시스템이 불안정해 질 수 있는 단점이 있다. 반면 모델예측 제어기는 이러한 물리적 제한 조건을 고려하지만, 계산이 복잡하다. 따라서 이 두 제어기법을 통합하여 물리적 제한 조건이 문제가 되지 않는 경우에는 IDCL 기법을 적용하고, 제어 입력이 포화되는 등의 문제가 발생하는 경우 모델예측제어 기법을 사용하는 제어기법을 적용한다. 제안된 기법을 도시하면 다음과 같다.

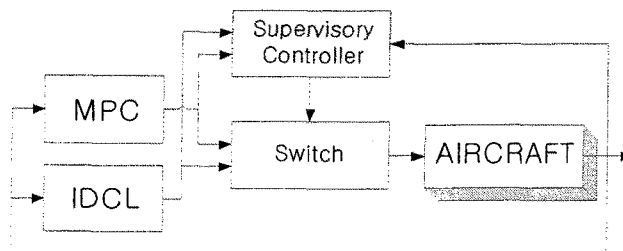


그림 3.2.36 IDCL/MPC 통합제어기

제 3 절. 하드웨어 다중화 관리 시스템 연구

1. 비행제어 시스템 개요

가. 개요

고도-마하수 영역의 경계는 여러 가지 다양한 요소들로 결정되는데 저속한계는 생성해낼 수 있는 최대 양력에 의해 결정되고 고속한계는 구조적 한계, 최고 고도는 최대 엔진 추력 등으로 결정된다. 최신 항공기의 경우 엔진성능의 향상과 구조 재질의 강화로 인해 이 비행 영역이 크게 확대 되었고 확대된 범위에 의해 항공기가 직면하게 되는 동압 범위도 매우 크게 확장되었다. 예를 들면 착륙 시 50psf에 불과한 동압이 마하 1.2로 지표면 순항시에는 2150psf까지 올라가게 된다. 이렇게 크게 확대된 동압 특성은 항공기 운동방정식의 공력계수에 큰 변화를 가져온다.

비행제어시스템의 기본 기능은 항공기의 안정성을 확보해주는 것이다. 항공기의 동특성 변화로 인하여 어느 비행 모드에서 안정성이 확보되는 비행체라고 할지라도 다른 비행모드에서는 불안정하거나 부적절한 댐핑 특성을 가지는 경우가 생기게 된다. 항공기의 운항을 위해서는 비행포화곡선의 확장으로 인하여 비행 구간 전반에 걸친 항공기 안정성의 확장을 필요로 하게 된 것이다. 현대 항공기에서는 조종계통에서는 보조동력으로 구동되는 조종면 시스템과 자동조종 시스템을 이러한 안정성확보에 대한 이유 때문에 반드시 필요로 하게 되었다. 비행영역 전체에 걸쳐 안정성을 확보하기 위해서는 센서를 사용한 되먹임 제어로 항공기 동특성을 변형시켜야 한다. 이러한 되먹임 제어이득은 비행 조건에 따라 많이 변화되는데 이러한 제어이득 선택 과정을 게인스케줄링이라 칭하며 이는 미리 정해진 계획에 따라 되먹임 이득값을 조절해주는 과정을 의미한다.

비행제어 시스템은 항공기의 안정성 확보뿐만아니라 조종사의 비행환경을 최적으로 구성해주는 역할도 수행한다. 과거의 저성능 항공기의 경우 비행범위가 한정되어 있어 기계식 링크로 연결된 조종면을 사람의 힘으로도 구동할 수 있었다. 따라서 항공기의 조종성능을 맞추기 위해 비행체의 형상단계에서부터 조종성을 고려해야 했으며 이로 인하여 항공기의 부적절한 동특성은 기본 디자인을 변경시켜야만 개선해 나갈 수 있었다. 항공기의 제어에 있어 사람이 손발로 조종하는 수준에는 한계가 있기 때문에 급격한 기동을 원하는 전투비행기 조차도 사람이 컨트롤 할 수 없는 1Hz미만의 동적 안전성과 정적 안정성은 기체 형상설계에서 확보해 주어야만 했다.

그러나 항공기 성능의 향상으로 인해 단순히 사람의 힘으로 조종면을 구동하기가 매우힘들어 지게 되었다. 이로 인해 별도의 조종면 동력을 공급받아 유압으로 구동되는 조종계통이 개발되었으며 이를 구동하기 위한 전기적 신호 시스템도 도입되었다. 이들 조종계통의 발달은 사람과 항공기 사이에 일종의 프로세서가 끼어들어 조종신호를 변조할 수 있는 기회를 가져다 주었다.

조종보조수단으로 아날로그 및 디지털 프로세서를 사용하게 되자 비행체 기체를 불안정하게 설계하여 기동성을 증가시키더라도 프로세서를 이용한 되먹임 제어를 수행하여 안정성을 강화시키며 비행할 수 있게 되었다. 통상 비행에서는 조종사가 조종시에는 과거의 안정한 비행체를 조종하듯이 비행할 수 있고 과거의 비행체에서는 구현할 수 없던 기동도가 가능하게 된 것이다.

나. 제어 시스템의 기능

(1) 항공기 동특성 제어

항공기는 기체의 형상에 의해 독특한 동특성을 가지게 되는데 이는 여러 가지 다른 범위로 분류 될 수 있다. 그 중 단주기 운동, 더치를 운동은 회전 자유도에 의해 결정되는 운동들로서 항공기의 관성 모멘트와 조종면에 의해 발생하는 모멘트에 의해 공진 주파수가 결정된다. 항공기의 반응은 회전 모드의 반응속도로 결정되는데 이들 모드들에 대한 주파수는 매우 높아서 항공기 자체가 이 모드들에 대하여 불안정하거나 댐핑성능이 나쁘면 조종사가 이를 제어하는 것은 거의 불가능하다. 따라서 자동 제어장치에서 이들 모드에 대한 적절한 댐핑특성이나 안정성을 부여해 주어야 한다. 이러한 제어 시스템을 안정성 강화 시스템(SAS)라고 한다. 만일 이러한 안정성 강화시스템이 조종사에 대한 입력에도 적용되어 조종사원하는 입력대로 항공기의 반응을 유도하는 제어시스템을 조종성 강화시스템(CAS)라고 한다.

휴고이드 모드나 나선 모드처럼 느린 동특성을 나타내는 모드는 조종사에 의해 제어가 가능하지만 조종사가 계속적인 제어입력을 주어야 하므로 조종에 편하지 않다. 자동 제어 시스템은 이러한 조종사에게 조종의 편의를 제공해야만한다. autopilot은 이러한 조종편의 기능이나 자동착륙과 같은 특별한 기능을 수행한다.

표 3.3.1 비행제어시스템 기본기능

SAS (Stability Augmentation System)	Roll Damper Pitch Damper Yaw Damper
CAS (Control Augmentation System)	Roll Rate Pitch Rate Normal Acceleration Lateral/Directional
Autopilot	Pitch Attitude Hold Altitude Hold Mach Hold Automatic Landing Bank-Angle Hold Turn Coordination Heading hold/VOR Hold

(2) 고성능기동 수행

최신 전투기의 경우 벡터 트러스트링이나 CCV방식의 비행 또는 VTOL 기능등의 고성능기동을 수행하게되는데 이러한 기동의 경우 기존의 단순한 조종간에 비례해서 조종면을 움직이는 단순한 제어로는 구현이 불가능하다. 제어시스템과 세밀한 시스템식별과정을 통하여 이러한 기동을 수행할 수 있도록 컴퓨터를 이용하여 제어해야 한다.

(3) 재형상기능 수행

시스템에 고장이 발생할 경우 공력의 변화로 인하여 정상상태에서와 같은 구동면 제어

를 하더라도 다른 동적 반응을 나타내게 된다. 따라서 고장 발생시에는 조종사가 원하는 조종이 구현하기가 쉽지 않게 된다. 비행제어시스템은 시스템의 변화로 인한 입출력 반응의 변화를 빠른시간내에 인식하여 원하는 출력을 낼 수 있도록 제어기를 재형상하는 과정을 수행한다.

다. 하드웨어 다중화 관리 기법

앞서 언급한 것처럼 비행제어컴퓨터로 인하여 조종사에게 편리한 조종 시스템이 구성되어가고 있다. 그러나 전자장비가 조종사와 기체가운데서 통제권을 독점하고 있기 때문에 비행제어시스템에 발생하는 고장은 곧 시스템 전체를 불능상태로 확산된다. 조종계통이 모두 기계계통으로 구성되어 있는 것에 비해 복잡한 전자장비로 조종계통을 구성하였을 때는 시스템자체의 신뢰성이 많이 떨어지게 된다. 특히 전자장비의 경우 벼락과 같은 전기충격과 자기충격에 매우 약하다. 이러한 점을 보완하기 위하여 FBW 방식의 비행제어 시스템은 모듈이 완전히 파괴되는 고장에서도 시스템이 정상적으로 동작하게 하기 위하여 다중화 개념을 도입할 필요가 있다.

(1) 안전성과 무결성

전자식 비행조종시스템은 기계식 비행조종시스템 못지 않게 안전해야 한다. 요구되는 안정성 수준은 항공기 조종을 어렵게하는 이상 현상으로부터 발생 가능한 고장의 확률로 규정된다. 일반적으로 비행조종시스템에서 고장 발생 확률은 군용기의 경우 비행시간당 1×10^{-7} 을, 민간항공기의 경우 1×10^{-9} 을 초과하지 않아야 한다. 이러한 낮은 확률 값은 충분히 인식하기 어렵고, 통계적으로 검증하기도 불가능하다. 이해를 쉽게 하기 위해서 설명하자면, 1×10^{-9} 이라는 고장 발생 확률 값은 연평균 3,000 시간의 비행을 하는 3,000대의 항공기가 100년 동안 한 번의 고장을 경험하는 것을 의미한다. 한편, 군용기의 연간 비행횟수는 민간항공기 보다 훨씬 적어서 1×10^{-7} 의 값을 사용할 수 있다. 현재 민간항공기에서 통용되고 있는 통계적인 안전성 수준은 비행시간당 1×10^{-6} 정도의 값을 갖는다. 이러한 수치는 연간 발생하는 민간항공기 사고의 총 횟수를 연간 총 비행시간으로 나눈 것로부터 얻어진다.

전자식 비행조종시스템 신호 채널의 고장 발생 사이의 평균 시간은 3,000 시간 정도이다. 따라서 전자식 비행조종시스템은 여러 개의 동일한 신호 채널을 갖는 여분을 갖추어서 적어도 두 번의 고장에 대해서도 정상 작동할 수 있어야 한다.

(2) 4중 다중화 관리시스템과 3중 감시 다중화 관리시스템

다중화 관리시스템은 요구되는 고장 복구 가능성을 맞추기 위해서 네 개의 완전히 독립적인 센서 채널과 컴퓨터 채널을 평행하게 구성한다. 이러한 구성을 4중 다중화 관리시스템이라고 한다. 네 개의 독립적인 채널은 구동기 시스템에도 사용한다. 따라서 센서, 컴퓨터, 구동기가 서로 연결된 전자식 비행조종시스템은 어떠한 형태로 발생하는 두 번의 고장에 대해서도 제대로 작동할 수 있다. 적절하게 구성된 다중화 시스템은 전기식 조종장치와 유압식 조종장치가 동시에 고장나더라도 시스템이 작동할 수 있도록 해준다.

4중 다중화 관리시스템에서 세 개 또는 네 개의 채널이 동시에 고장나는 확률은 무시할 수 있다고 가정한다. 따라서 조종면을 구동하는 한 채널에 고장(hardover)이 발생할 경우,

다른 세 개의 양호한 채널이 고장난 채널을 우선할 수 있다. 그러나 두 번째 고장이 발생할 경우를 대비해서 고장이 발생한 첫 번째 채널과 단절할 필요가 있다. 그렇지 않으면 양호한 두 채널과 고장난 두 채널이 상반된 신호를 제공해서 조종면이 움직이지 않게 되는 상황이 발생할 수 있다. 이러한 경우를 “수동적 고장”(fail passive)이라고 한다.

고장은 여러 채널 사이의 교차비교(cross comparison)와 여러 채널에서 하나를 선택하는 원리(odd man out principle)를 적용한 다수 투표(majority voting)에 의해서 검출된다. 따라서 4중 다중화 관리시스템은 첫 번째 고장 이후 고장난 채널을 차단하여 3중으로 낮아진 다중화 관리시스템을 운영함으로써 두 번째 고장 이후에도 다수 투표에 의해 제대로 작동할 수 있다. 세 번째 고장이 발생하면 양호한 하나의 채널과 고장난 하나의 채널이 상쇄하는 “수동적 고장” 상황을 초래한다.

한편, 매우 높은 신뢰도를 갖고 채널의 작동을 점검하는 감시 시스템도 고장난 채널을 구분하고 차단시켜 줄 수 있다. 이것은 3중 감시 다중화 관리시스템(monitored triplex)으로 불리는 고장 대처 방안이다. 3중 감시 다중화 시스템은 세 개의 완전히 독립된 평행한 채널로 구성되며, 각 채널은 고장을 검출하는 상이 시스템(dissimilar system)에 의해서 감시된다. 이러한 감시가 높은 무결성 정도와 신뢰도 수준을 제공한다면, 시스템에 두 번의 고장이 발생하여도 제대로 작동할 수 있다.

위에 설명한 두 시스템은 각각 장단점을 갖고 있다. 3중 감시 다중화 시스템은 보다 적은 하드웨어를 가지므로 보다 저렴하게 구성할 수 있다. 반면에 4중 다중화 시스템은 높은 신뢰도 수준의 자기 감시 시스템을 포함할 때, 고장 복구 신뢰도 측면에서 보다 좋은 성능을 갖는다.

(3) 하드웨어 다중화 관리시스템의 기능

하드웨어 다중화 관리시스템은 논리적인 소프트웨어와 물리적인 하드웨어로 구성된다. 소프트웨어 부분은 고장허용실행(Fault Tolerant Executive, FTE) 기능을 담당하고, 하드웨어 부분은 교차채널 데이터 링크(Cross Channel Data Link, CCDL) 기능을 담당한다. 각각의 기능은 서로 다른 보드에서 담당하도록 설계되고, 적절한 인터페이스를 통하여 데이터를 교환하는 이중 구조를 갖는다. 그림 5.2.1은 하드웨어 다중화 관리시스템의 데이터 흐름도(Data Flow Diagram, DFD)를 나타낸 것이다.

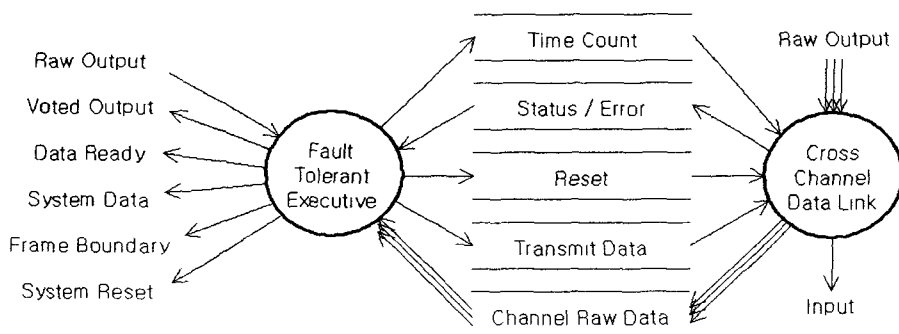


그림 3.3.1 다중화 관리시스템의 데이터 흐름도

고장허용실행 기능은 다음의 작업을 수행한다.

- 시스템 동기화 유지
- 데이터와 시스템 내부 변수에 대한 투표
- 고장 검출, 고장 분리 및 고장 복구
- 교차채널 데이터 링크 관리
- 초기화 단계에서 자기진단 수행
- 인터페이스 관리
- 이상의 작업을 정확히 수행하도록 하는 커널

교차채널 데이터 링크 기능은 다음의 작업을 수행한다.

- 다중화 채널 사이의 물리적인 인터페이스
- 데이터 메시지 변환 시 오류 점검

(가) 교차채널 감시기법

교차채널 감시(Cross Channel Monitoring, CCM)기법은 2개 이상의 채널에 대해서 투표를 하여 고장을 감시하는 기법이다. 투표기법은 1960년대에 처음 소개되었고, 고장허용 보장에 있어서 중요한 역할을 담당해왔다. 1970년대에는 아날로그 시스템에 적용되었지만, 시스템에 연결 가능한 편 수가 그리 많지 않았기 때문에 여러 개의 센서를 이용하기는 어려웠다. 그러나 디지털 방식의 전자식 비행제어시스템이 개발됨에 따라 다양한 방법의 해결책이 제시되었다. 예를 들면, 센서의 모든 신호를 비행제어 컴퓨터의 직렬 디지털 데이터버스에 연결할 수 있다.

투표기법은 엄밀한 의미에서의 다수 투표(majority voting)를 말하며 영과 일(one-or-zero)의 이중 구조를 갖는다. 전자식 비행제어시스템에서 투표는 연속된 신호에 대해서 신호 선택과 고장 검출로 구성된다. 신호 선택이란 유사한 여러 신호 가운데에서 하나의 기준신호를 선택하는 과정을 말한다. 이는 평균값, 중간값, 또는 다른 어떤 신호의 형태를 취할 수 있다. 이와 같이 선택된 신호를 기준값이라고 한다. 기준값 설정에 대해서는 1)에서 다루기로 한다.

고장 검출은 여러 개의 채널을 서로 비교하여 고장이라고 판단되는 채널을 검출하는 과정이다. 4중 다중화 관리 시스템은 2개의 채널에 고장이 발생한 경우에도 투표 기법을 사용하여 고장 채널을 검출하고 분리함으로써 보다 안전한 비행제어시스템의 구성이 가능하게 한다.

하드웨어로 투표 과정을 구성하는 것은 다소 복잡하다. 3중 다중화 관리 시스템에서 투표는 세 개의 신호 중에서 중간값을 출력값으로 결정한다. 4중 다중화 관리 시스템에서 투표는 두 개의 중간값 중에서 작은 값을 출력값으로 결정한다. 그러나 두 중간값도 계속 감시되어야 하며, 미리 설정된 기준값보다 작을 경우에만 네 개의 신호가 모두 유효한 것으로 본다.

전자식 비행제어시스템에서는 투표 과정을 소프트웨어 알고리즘으로 구성함으로써 복잡성을 감소시킬 수 있다. 그림 5.2.2는 4중으로 구성된 다중화 관리 시스템에서의 투표 과정의 한 예를 보여주고 있다. 4중으로 구성된 센서 신호가 비행제어 컴퓨터로 입력되면 신호는 신호 선택과 고장 검출의 투표 과정을 거쳐서 비행제어 알고리즘으로 입력된다. 마찬

가지로 4중으로 구성된 비행제어 알고리즘의 제어입력은 신호 선택과 고장 검출의 투표 과정을 거쳐서 작동기로 전달된다. 이와 같이 소프트웨어 투표 과정은 시스템의 복잡성을 감소시킨다.

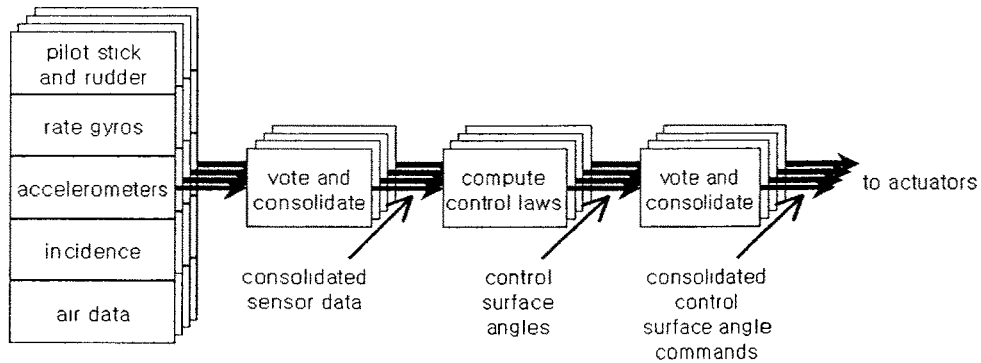


그림 3.3.2 4중 다중화 관리 시스템에서의 투표 과정

1) 기준값 설정 알고리즘

한계치(threshold value)는 교차채널 감시기법에서 투표 과정을 통해 결정된 기준신호의 허용범위를 의미한다. 여기서 기준신호는 여러 신호에 대한 중간값 또는 평균값으로 설정한다. 센서에서 출력되는 신호는 참값과 오차를 포함하고 있기 때문에 임의의 채널 신호와 기준신호 사이의 차이는 거의 오차의 차이로 구성된다. 따라서 이러한 차이는 정규분포 특성을 갖는다고 가정할 수 있다. 평균이 0이고 분산이 1인 센서신호 x_1 과 기준신호 x_2 가 그림 5.2.2와 같이 1Hz 측정되었다고 하면, 두 신호 사이의 차이 $x = x_1 - x_2$ 는 그림 5.2.3과 같이 된다. 두 신호 사이의 차이가 정규분포 특성을 갖는다는 가설에 대해서 95% 신뢰도로 χ^2 -검증을 수행해보면 가설이 기각된다. 그러나 그림 5.2.3과 같이 분포도를 그려보면 ± 1 범위 내에서는 정규분포 특성을 갖는 것을 확인할 수 있다.

한계치는 통계적인 방법을 통해서 설정할 수 있다. 임의의 신호와 기준신호 사이의 차이가 다음과 같은 정규분포함수를 갖는다고 가정하면 이를 이용하여 한계치를 설정할 수 있다.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

여기서 x 는 샘플 데이터, μ 는 x 의 평균, σ 는 x 의 분산이다.

일반적으로 고장허용시스템에서는 99%의 신뢰도를 사용하는데 이에 상응하는 신뢰구간은 다음과 같다.

$$|x| \leq 2.575\sigma$$

따라서 한계치는 $C_v = 2.575\sigma$ 로 설정할 수 있다.

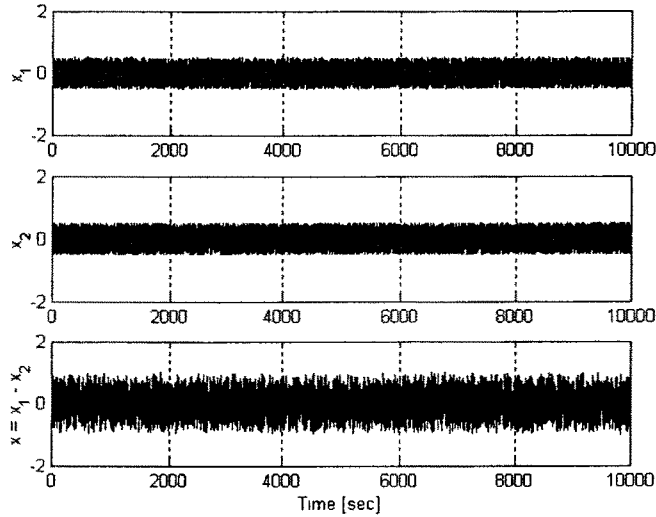


그림 3.3.3 센서신호와 기준신호 사이의 차이

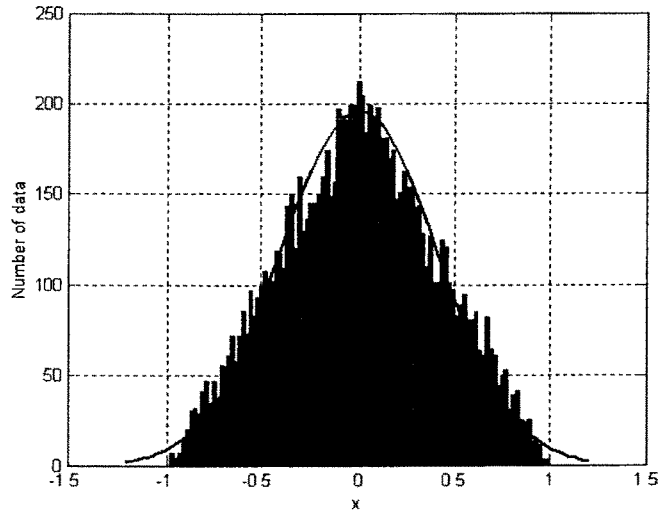


그림 3.3.4 정규분포 곡선과 막대분포도

2) 유효시간

고장이 발생하면 센서에서 출력되는 샘플 데이터는 연속해서 한계치를 넘어가게 된다. 유효시간(confirmation time)은 샘플 데이터가 한계치를 넘었을 때 고장으로 인정할 수 있는 초과횟수를 말한다. 따라서 분명하게 고장을 검출해 낼 수 있도록 정상신호의 경향을 실험적으로 분석할 필요가 있다.

그림 5.2.5는 T 시간 동안 측정된 정상신호의 유형을 나타낸 것이다. 여기서 μ 는 정상신호의 평균, σ_1 , σ_2 , σ_3 는 정규화 된 표준편차를 나타낸다. 앞서 설명한 바와 같이, 한계치는 샘플 데이터의 99%를 포함하는 값이다. 따라서 샘플 데이터의 1%는 한계치를 넘어서게 되고, 이것은 고장이 아니므로 유효시간을 결정하기에 적합하지 않다. 이러한 모호성을 해결하기 위해서 그림 5.2.5와 같이 몇 개의 정규화 된 표준편차를 사용한다. 각각의 표준편차에 대해서 연속적으로 넘는 샘플 개수를 기록하고, 신호의 특성을 고려하여 적절한 한계치

범위를 넘는 샘플 개수를 유효시간으로 결정한다. 일반적으로 정상신호에서 나타나는 특성과 고장 발생 시 나타나는 신호 특성이 유사하다고 가정함으로써 정상신호에 대해 설정한 유효시간을 고장 발생 시에도 그대로 사용한다.

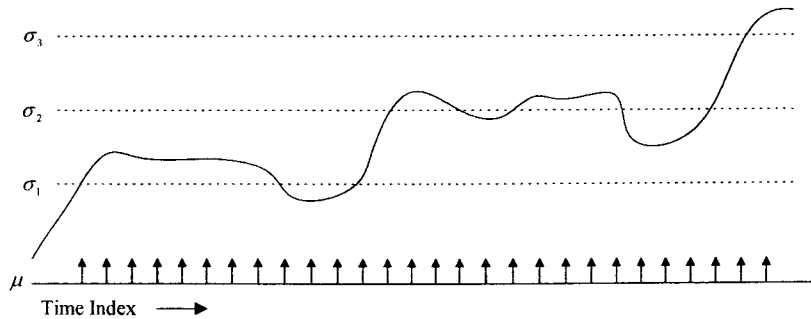


그림 3.3.5 정상신호 유형과 유효시간 결정

2. 다중화 관리 시스템 구성

가. RM 시스템 구성

하드웨어 여분 개념을 도입하고 있는 시스템은 다중화 된 신호처리를 위하여 기능별 모듈화를 시도하고 있다. 특히 비행제어시스템의 경우 센서입력부, 비행제어부, 제어입력생성부, 버스관리부등의 기능별로 모듈을 분류해 놓고 각각의 모듈에 프로세서를 탑재하여 독립적으로 구동한다. 이러한 병렬 처리방식을 도입함으로써 시스템은 네트워크 관리의 개념으로 운영이 되게 된다. 비행제어 시스템을 네트워크 방식으로 전환하여 운영할 경우 다음과 같은 장점을 가지게 된다.

고장의 분리

다중화 시스템의 가장 중요한 목적은 고장의 분리이다. 하나의 모듈에서 고장이 발생한다고 하더라도 고장의 영향이 다른 모듈로 파급되지 못하도록 시스템을 설계해야 할 것이다. 각각의 모듈에 별도로 프로세서를 부착하고 기능별 모듈을 중복하여 설치하게 되면 하나의 모듈의 고장에 대해 같은 기능을 하는 다른 모듈이 대체하여 기능을 수행하고 고장난 모듈에서의 신호전달은 분리시킴으로서 고장 영향을 최소화할 수 있다.

기능별로 세세하게 다분화한 시스템의 경우 하나의 고장에 대해 시스템 전체에 미치는 영향을 최소화 할 수 있다. 단 다중화 시스템의 경우 모듈간 통신량이 많아져서 버스에 병목현상이 발생할 문제가 있고 모듈이 많아지므로 중량과 가격이 커진다는 문제가 발생할 수 있다.

시스템 처리 부하 분담

병렬처리 방식으로 수행하면 이전에 하나의 제어컴퓨터에 모든 기능이 집중화 하여 시스템 부하가 커지게 되는 현상을 방지할 수 있으며 각각의 모듈에서 중앙 비행제어관련 작업

을 분배하게 되므로 처리속도도 빨라지게 된다. 실제로 본 연구에서 사용하는 센서입력부 모듈은 센서에서부터 데이터를 받는 작업을 전담함으로써 비행제어부에서 센서데이터를 받기위해 사용하는 시간을 다른 작업에 할애할 수 있게 되었다.

개발의 편의 및 객체 지향 가능

기능별 분화를 시킬 경우 각각의 모듈에서 필요한 기능만을 개발하면 되므로 시스템의 통합 및 관리를 객체지향적으로 할 수 있다는 장점을 가지게 된다. 이는 하드웨어적인 측면에서 객체지향적 개발을 할 수 있다는 점에서 개발의 편의를 가져다 준다.

예를 들면 센서 입력부에서는 센서 측정치만을 획득하여 전송하는 역할을 수행하고 비행제어부에서는 제어신호만을 계산하고, 제어신호생성부에서는 조종면 제어신호만을 생성해주면 되므로 개발시에도 간단하게 모듈 검증을 할 수 있으며 모듈교체가 용이하기 때문에 비교적 저렴하게 운영과 보수 유지를 할 수 있다는 장점을 가지고 있다.

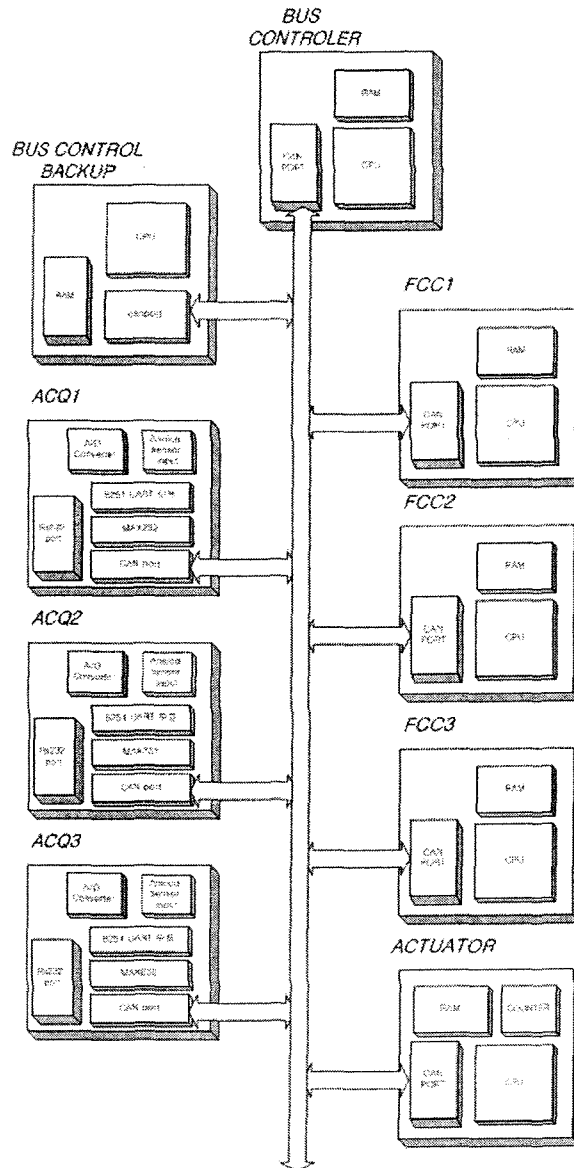


그림 3.3.6 비행제어시스템의 다중 구조

3. 비행제어시스템 모듈 프로세서 DSP(Digital Signal Processor)의 소개

CPU 기술이 매년 현저하게 발전되어 감에 따라 항공기의 비행제어 컴퓨터로 사용될 수 있는 마이크로컴퓨터의 선택 폭은 매우 확대되었다. 마이크로프로세서는 전문가들 사이에서 널리 받아들여지고 있는 분류 기준에 따라 다음과 같이 나눌 수 있다.

범용마이크로프로세서

PC에서 널리 사용되고 있으며, 마이크로프로세서 단독으로 동작하지는 않으며, 주변에 롬(ROM), 램(RAM), 주변장치를 위한 부가회로 등이 필요하다. 소프트웨어면에서는 주변장치와의 호환을 위하여 OS(Operating System ; 운영체제)가 필요하다. 하지만, 항공기의 실시간 제어용 컴퓨터로 사용하기에는 인터럽트의 지연 시간, 하드웨어 단계에서의 개발자의 직접적인 접근이 쉽지 않다는 단점이 있다.

마이크로컨트롤러

소형 기기의 제어용으로 사용되는 프로세서이며, 주변에 부가장치를 부착하거나 내부에 이를 내장할 수 있다. A/D(Analog/Digital) 변환기, PWM(Pulse and Width Modulation) 발생기, 카운터, 타이머 등을 내장하고 있는 경우가 많다. 일반적으로 개발자의 접근이 쉽고, 사용이 간편하여 널리 사용되고 있지만, 항공기의 제어용 컴퓨터로 사용하기에는 연산 속도가 너무 느리다.

DSP

다양한 신호처리 및 고속연산에 유리하도록 특별히 제작한 프로세서로 다음과 같은 특징을 가지고 있다.

- ① 하드웨어적인 부동 소수점(Floating Point) 연산
- ② MAC(Multiplication and Accumulation), 즉 덧셈과 곱셈을 한 사이클에 수행하는 연산
- ③ 빠른 외부 메모리 인터페이스
- ④ 하바드 구조(Harvard Architecture), 프로그램 메모리와 데이터 메모리가 구분되어 있는 구조

항공기의 제어용 컴퓨터에서 행렬 연산을 요구하는 경우가 많기 때문에, 각 행렬의 요소를 곱한 후 이를 더하는 과정을 동시에 진행할 수 있다는 것은 DSP가 가지고 있는 중요한 특징 중의 하나이다. 그 외에도 파이프 라인(Pipe Line)구조를 가지고 있어 동시에 2개 이상의 명령어를 수행할 수 있으며, 반복 연산을 빠르게 수행할 수도 있다. 그리고 기존에 널리 사용되던 CISC(Complex Instruction Set Computer)방식과는 달리, DSP는 RISC(Reduced Instruction Set Computer)방식을 채택하고 있으며 사용되지 않는 다양한 명령어를 제거하고 자주 쓰이는 간단한 명령어들이 보다 빨리 수행될 수 있도록 설계한 CPU이다.

최근에는 마이크로프로세서의 사용에 있어, 복잡한 실시간 처리를 요구하는 경향이 많기 때문에 마이크로컨트롤러와 DSP의 경계선이 많이 허물어지고 있다. 마이크로컨트롤러는 DSP의 장점인 부동 소수점 연산 기능이나 MAC의 기능을 추가하는 경우가 있고, DSP는 마이크로컨트롤러의 특징이 다양한 부가회로를 DSP 내부에 내장시킴으로써 상대방의 장점

을 취합하고 있다. 본 연구에서 사용하는 TMS320LF2407A도 이 같은 경향을 반영하여 다양한 CPU 내부에 10개의 AD채널과 16개의 PWM 신호생성채널, UART 기능, 및 Canbus 등 컨트롤에 필요한 많은 입출력 단자를 가지고 있기 때문에 입출력 관련 인터페이스를 자체적으로 소화해 낼 수 있다. 이 같은 내장 기능은 비행제어컴퓨터의 전력량을 획기적으로 줄이게 되고, 항공기 탑재중량에서 가장 문제가 되던 배터리의 중량도 감소시킬 수 있다. 실제로 미사일의 자세 제어와 군수용 레이더에 이르기까지 많은 제어시스템에 실제로 DSP가 운용되고 있다. 이러한 여러 장점을 가지고 있는 DSP를 본 연구의 비행제어컴퓨터로 선정하였고 센서데이터 획득, 비행제어 관리, 조종면 제어신호 생성, 비행데이터 저장, 통신 프로그램 관리 등의 역할을 수행할 것이다.

TMS320LF2407A의 특징

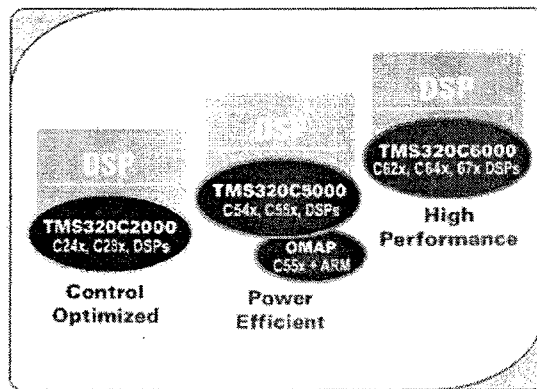


그림 3.3.7 DSP 플랫폼의 특징

본 연구에서 사용하는 DSP는 Texas Instrument 사의 C2000 제품군에 해당하는데, 디지털 제어용으로 최적화된 구조로 원칩 형태로 다양한 모터 제어에 용이하도록 제작되고 있다. TMS320LF2407A의 경우 고정소수점 방식이고, 16bit 레지스터(Register)를 가지고 있으며 내부에 많은 부가회로를 가지고 있다. 가격이 또한 저렴하여 마이크로컨트롤러와 비슷한 가격으로 시스템을 구성할 수 있다. TMS320LF2407A는 다음과 같은 특징을 가지고 있다.

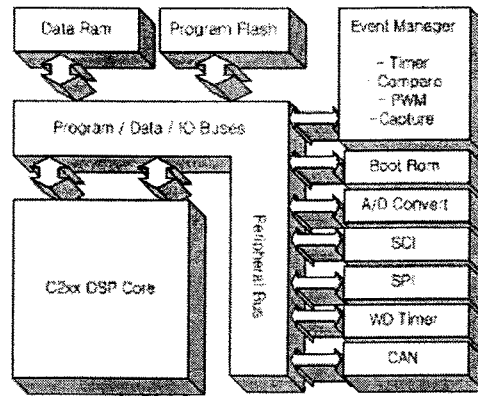


그림 3.3.8 DSP 내부 구조 개념도

- 1) 고성능 Static CMOS 기술 적용
 - 한 명령어당 최대 25ns의 빠르기
 - 연산속도는 최대 40MIPS까지 가능
 - 저전력 3.3V 코어
- 2) On-Chip 메모리
 - 최대 32K 워드 × 16Bits의 플래쉬 메모리
 - 최대 2.5K 워드 × 16Bits의 데이터/프로그램 메모리
- 3) Boot Rom
 - SCI/SPI 부트 로더(boot loader) 지원
- 4) 이벤트-매니저 모듈(Event Manager Module)
 - 16bit 범용 목적의 타이머
 - 8개의 16bit PWM 발생기
 - Center or Edge Alignment of PWM Channels
 - 외부 PDPINTx Pin을 이용한 Emergency PWM Channel Shutdown
 - 프로그래밍이 가능한 데드밴드(Deadband, Deadtime) 제공
 - 3개의 캡처(Capture) 유닛
 - 엔코더 인터페이스 회로 내장
 - 타이머 인터럽트에 동기된 A/D 변환 기능
- 5) 외부 메모리 인터페이스 기능 포함
 - 총 192K 워드 메모리 영역: 64K 프로그램 영역, 64K 데이터 영역, 64K I/O 영역
- 6) 와치독(Watch dog) 타이머 모듈
- 7) 10bit A/D 변환기
 - 500ns 최소변환시간
- 8) CAN(Controller Area Network) 2.0B 모듈 내장
- 9) UART 기능의 SCI(Serial Communication Interface)
- 10) 동기형 16bit SPI(Serial Peripheral Interface)
- 11) PLL(Phase-Locked-Loop)에 기반한 클럭 생성
- 12) 최대 40개까지 활용 가능한 GPIO(범용 I/O핀)
- 13) 5개의 외부 인터럽트

- 14) 3단계 전원모드 / 기능별 전원관리 가능
- 15) 동작온도범위 -40 ~ 85 °C

TMS320LF2407A의 기능

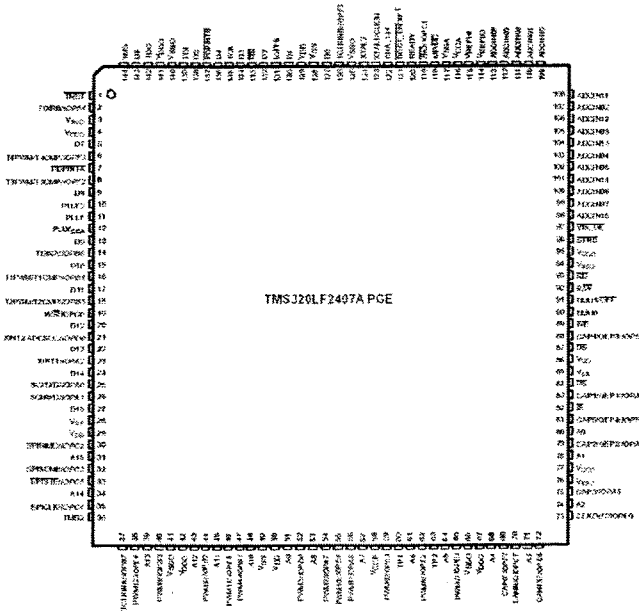


그림 3.3.9 DSP의 Pin 배열

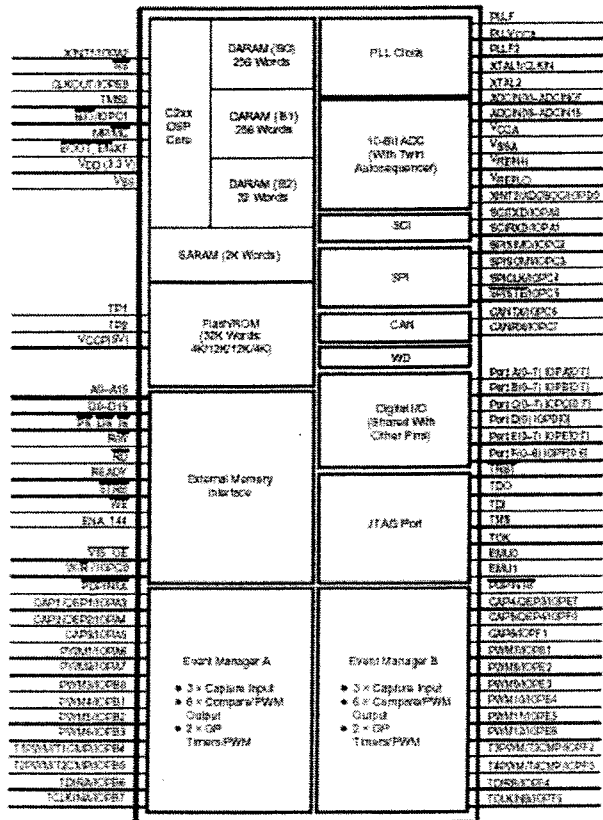


그림 3.3.10 DSP 내부 기능 블록 다이어그램

1) 내부 메모리

데이터 메모리용 활용공간이 최대 2.5K 워드가 있으며, 이 중 544 워드는 듀얼 액세스 메모리로서 필요에 따라 활용할 수 있다. 플래시 메모리는 32K 워드가 내장되어 있다.

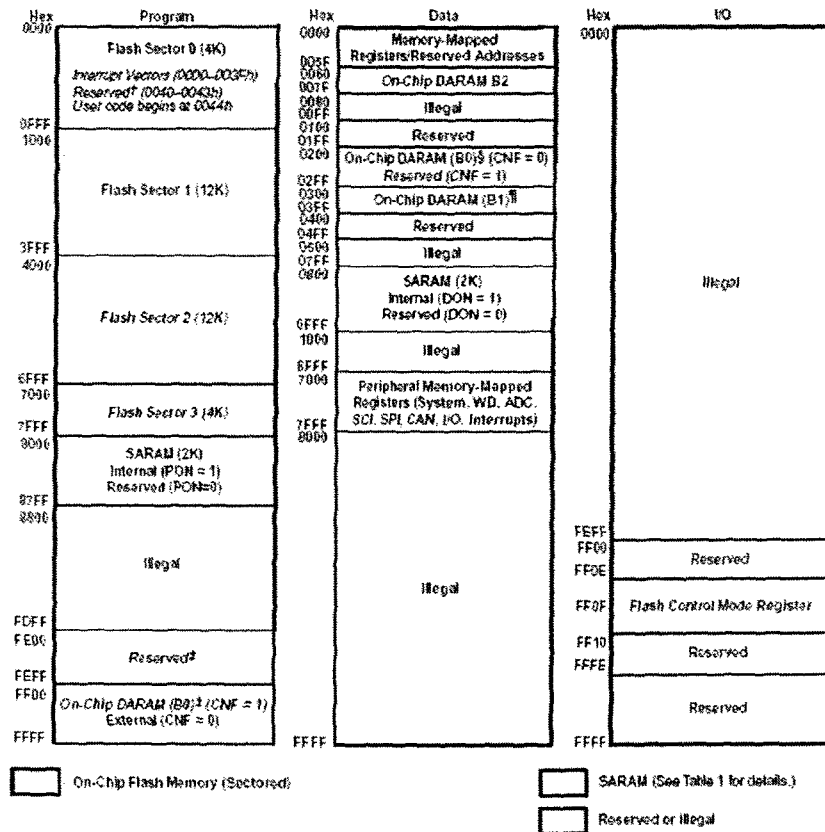


그림 3.3.11 TMS320LF2407A 내부 메모리 맵

2) 외부 메모리 인터페이스

외부 메모리를 부착하여 프로그램/데이터 영역을 할당할 수 있다. 별도의 어드레스 디코딩 회로가 필요하지 않으며, 빠른 속도를 보장하기 위하여 10ns 이하의 SRAM을 사용하는 것이 좋다.

3) EVA/EVB(Event manager module A, B)

PWM, Capture, Compare, Timer 등을 하나로 묶은 모듈을 이벤트 매니저 모듈이라고 부른다. 일부 기능이 서로 중첩된 핀에 할당되어 있기 때문에, 모든 기능을 동시에 사용할 수 있는 없다. 다른 DSP에서는 찾아보기 힘든 기능으로, 제어용 컴퓨터로서 2407A의 효용을 크게 증대시키는 역할을 한다.

4) 통신용 포트

SCI 시리얼 통신, 동기식 통신인 SPI 통신, 산업체에서 많이 사용되는 CAN 버스 통신 제어 기능이 내장되어 있다. SCI 시리얼 포트를 이용해 부트 로더 기능을 사용하면 저렴한 개발환경을 운용할 수 있다.

5) A/D 변환기

아날로그 신호를 10bit 디지털 신호로 바꾸어 주는 A/D변환기가 16채널이 있다. 변환시간이 375ns에 불과하여 고급 A/D변환기 수준의 빠른 변환 속도를 가지고 있다.

TMS320LF2407A를 이용한 비행제어컴퓨터 구현

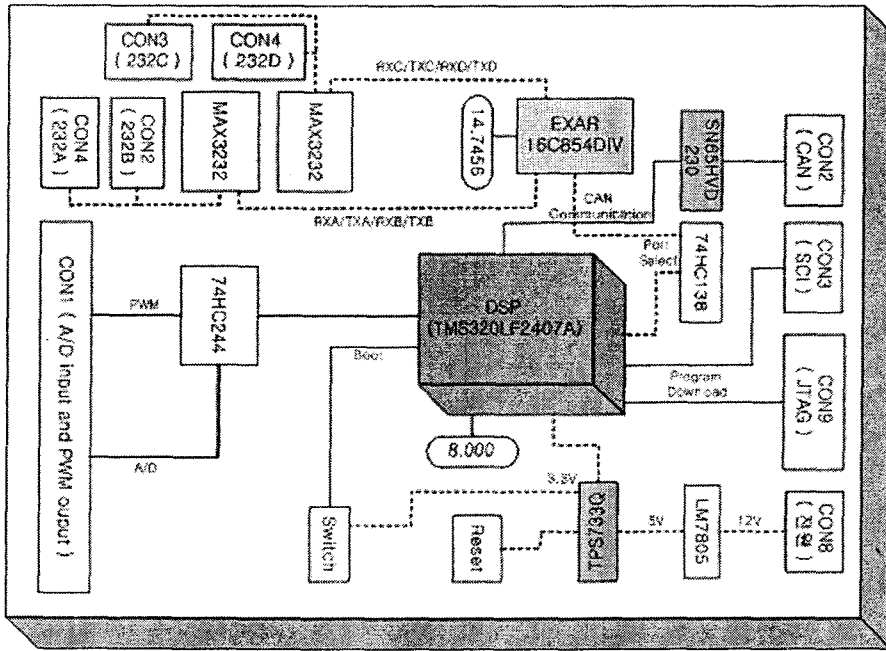


그림 3.3.12 비행제어컴퓨터 설계 개념도

가로 8.3cm, 세로 6.5cm의 4층 PCB로 구성된 비행제어컴퓨터(Flight Control Computer : FLCC) 보드는 크게 다섯 부분으로 구성되어 있다.

1) 마이크로 프로세서

마이크로프로세서로 사용되는 DSP320LF2407A는 PCB의 다른 모든 부분과 긴밀하게 연결되어 있다. 어드레스(Address) / 데이터(Data) 라인, ADC 신호 입력 라인, PWM 출력 라인, 캡처(Capture), SCI / SPI / CAN 통신 라인, 부트 로더(Boot Loader) 연결 라인, 클럭 입력 라인 등이 연결되어 있다. 2407A에는 8 MHz 클럭 소스가 입력되는데 PLL(Phase Locked Loop)을 소프트웨어적으로 조작하여 최대 40MIPS의 속도로 동작하는 것이 가능하다. PLL 기능은 SCSR1의 레지스터의 비트 11- 비트 9의 3 비트값을 조정하면 쉽게 이용할 수 있다. 프로세서의 성능이 날로 향상되면서 프로세서에 인가하는 클럭 속도가 점점 높아지고 있지만, 높은 주파수를 발생시키는 오실레이터(Oscillator)나 크리스탈(Crystal)을 사용할 경우 EMI(Electro Magnetic Interference)문제를 발생시키고, 부품의 가격도 상승시킬 수 있다. 그러므로, PLL을 사용하여 소프트웨어적으로 분주를 시키는 마이크로프로세서의 기능은 위에서 언급한 문제점을 해결해 줄 수 있다는 장점이 있다. 하지만 외부에서 들어오는 노이즈가 그래도 증폭이 되는 경향이 있기 때문에, 오실레이터와

DSP사이는 가능한 한 짧게 배선하였다.

07018h	—	CLKSRC	LPM1	LPM0	CLK PS2	CLK PS1	CLK PS0	—	SCSR1
	ADC CLKEN	SCI CLKEN	SPI CLKEN	CAN CLKEN	EVB CLKEN	EVA CLKEN	—	ILLADR	

그림 3.3.13 SCSR1 레지스터

CLK PS2	CLK PS1	CLK PS0	CLKOUT
0	0	0	$4 \times F_{in}$
0	0	1	$2 \times F_{in}$
0	1	0	$1.33 \times F_{in}$
0	1	1	$1 \times F_{in}$
1	0	0	$0.8 \times F_{in}$
1	0	1	$0.66 \times F_{in}$
1	1	0	$0.57 \times F_{in}$
1	1	1	$0.5 \times F_{in}$

Default multiplication factor after reset is (1,1,1), i.e., $0.5 \times F_{in}$.

그림 3.3.14 PLL 클럭 선택 방법

2) 통신부

FLCC 보드의 통신부는 네 개의 RS232용 커넥터와 두 개의 MAX3232, 74138, XR16C854DIV, SN65HVD230와 CAN 통신용 커넥터 등으로 이루어져 있다. DSP에서 74138에 A-D의 RS232 포트 선택 신호를 추가하면, XR16C854에서는 네 개의 UART(Universal Asynchronous Receiver and Transmitter)채널 중 하나를 선택하여 데이터를 받아들인다. MAX3232는 주변장치의 전압 신호와 보드 상의 전압 3.3V의 균형을 맞춰주고, SN65HVD230 역시 TTL 레벨의 보드 신호를 CAN 통신을 위한 전압으로 변환해 주는 소자이다.

CSA#	CSB#	CSC#	CSD#	FUNCTION
1	1	1	1	UART de-selected
0	1	1	1	Channel A selected
1	0	1	1	Channel B selected
1	1	0	1	Channel C selected
1	1	1	0	Channel D selected
0	0	0	0	Channels A-D selected

그림 3.3.15 채널 A-D 선택 방법

XR16C854는 향상된 4채널의 UART용 칩으로 각 채널마다 128bytes의 송/수신 FIFO(First-In, First-Out) 메모리를 가지고 있다. FIFO는 기억 공간에서 자료를 저장하고 회수하는 방법의 일종으로 먼저 저장된 자료가 먼저 회수되는 것을 말한다. 각 페이지가 기억 장치로 들어올 때마다 타임-스탬프를 찍어 그 시간을 기억하고 있다가 페이지가 교체될 필요가 있을 때 가장 먼저 기억 장소에 들어와 있는 페이지를 교체시키는 방법이며, 대기열(queue)은 이 방법을 사용하는 대표적인 예이다. 16C654가 64bytes, 16C554가 16bytes, 16C454가 1bytes의 FIFO 메모리를 제공했던 것에 비하면 매우 높은 FIFO 메모리를 내장하고 있고, 이러한 대용량의 FIFO 메모리 덕분에 매우 빠른 데이터 처리 속도가 가능하다. 이 보드에서는 비행제어컴퓨터가 주어진 시간에 더 많은 네트워크 작업을 수행할 수 있다.

예를 들어, FIFO 메모리를 64bytes를 가지고 있는 16C554는 1.53ms의 시간마다 16bytes를 송신 처리해야 한다(시작/끝 비트를 포함한 캐릭터 길이가 11bits이고 115.2Kbps의 데이터를 받을 경우). 이것은 비행제어컴퓨터가 1.53ms마다 한번씩 외부 통신을 수행해야 한다는 것과 같다. 반면에 128bytes의 FIFO 메모리를 가지고 있는 16C854는 12.2ms마다 한번씩만 송신 처리를 하면 된다. 이것은 비행제어컴퓨터가 다른 응용 작업에 더 많은 시간을 투자할 수 있도록 해주며, 전체 UART 통신 인터럽트를 처리하는 데 소비되는 시간을 많이 줄여 줄 수 있다.

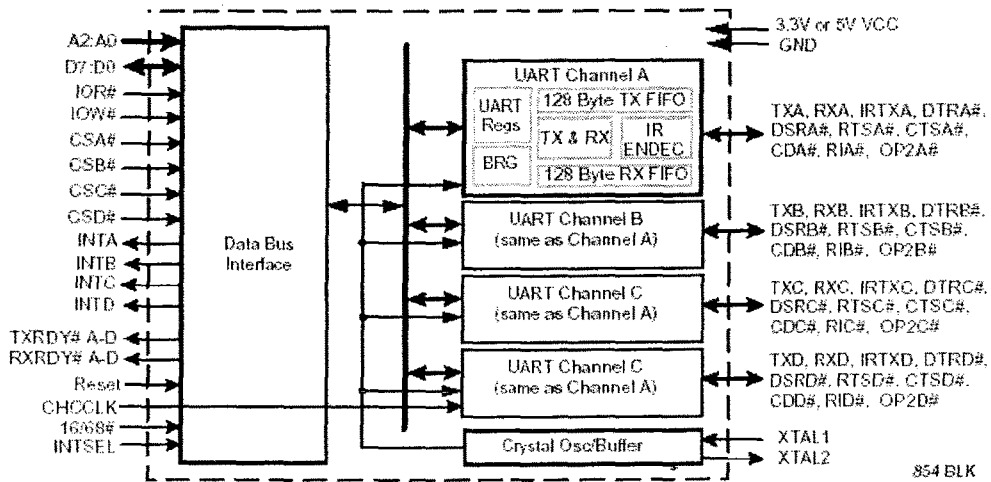


그림 3.3.16 XR16C854 블록 다이어그램

XR16C854는 하드웨어(RTS/CTS)/소프트웨어(Xon/Xoff)적인 자동(Automatic) 플로우(flow)컨트롤도 가능하다. 이것은 다중채널시스템에서 버퍼에 최대량의 데이터가 모였을 때 송신 처리를 하도록 구현될 수 있다. 이러한 여러 기능을 조합하면, 데이터 처리 속도가 무려 2Mbps에 이를 수 있고 물론 전력도 절약할 수 있다. 각각의 UART 채널은 사용자에게 운용 상태와 컨트롤, 수신기의 에러 알람 신호, 모뎀의 직렬 인터페이스 컨트롤을 제공하는 레지스터(Register)들을 자체적으로 가지고 있다. 가변적인 인터럽트 양극신호는 사용자의 설계 요구조건을 만족시킬 수 있으며, 내부 루프백(loopback : 신호 또는 신호의 지나온 경로가 되돌려지는 것)능력은 온보드(onboard)상에서의 자가 진단을 가능하게 해준다. 이러한 XR16C854는 항공기에서 비행제어컴퓨터와 주변장치 사이에 보다 빠른 데이터 통신을 가능하게 해주는 것은 물론이고, 자가 진단 능력을 잘 활용한다면 다중화 시스템의 고장 진단에 응용할 수 있다는 큰 장점을 가지고 있다.

3) 입/출력부

CON1로 표시된 15핀 2열 헤더 핀(Header Pin)에는 12개의 외부 센서신호가 입력되어 간단한 저주파 필터(LPF : Low Pass Filter)를 거쳐서 DSP의 A/D 12채널에 입력된다. 그리고 외부에서 4개의 QEP(Quadrature Encoder Pulse) 신호가 CON1으로 입력되어 74HC244를 거친 후 DSP의 QEP1-4 핀에 입력된다. QEP는 항공기의 서보 신호를 피드백 받는 데 이용할 수 있도록 배선하였다. DSP의 PWM1-4 핀에서는 4개의 PWM 신호가 출

력되어 74HC244를 거친 후 CON1을 통하여 서보 모터에 연결될 수 있도록 배선하였다.

4) 프로그램 입력부

DSP의 플래시 메모리에 프로그램을 기록하기 위해서는 다음의 두 가지 방법이 있다.

- ① JTAG 에뮬레이터 사용
- ② SCI 시리얼 포트를 사용하여 부트 로더 이용

JTAG 에뮬레이터를 사용하여 프로그래밍 하는 편이 빠르고 간편하지만, JTAG 에뮬레이터가 상당히 고가이고, 개인이 구입하기에는 많은 무리가 따른다는 단점이 있기도 하다. 그러므로 초기 개발 단계에서는 부트 로더를 이용하여 프로그래밍을 하는 방법을 택하였다. 본 보드의 프로그램 입력부는 CON3(SCI), CON9(JTAG), 스위치 등으로 이루어진다. CON3은 PC의 RS232C 직렬 포트와 연결되어 있고, 스위치는 DSP의 121번 핀(Boot EN/XF)에 연결되어 있다. 스위치는 시리얼 포트로 플래시 메모리에 프로그램을 기록하기 위해 부트 로더를 동작시키는 역할을 하는데 이 스위치를 Off 시킨 후 리셋 스위치를 누르면 부트 로더가 동작하게 된다. 플래시 메모리를 기록한 다음에 프로그램을 실행시키기 위해서는 다시 이 스위치를 On한 후 리셋을 걸면 플래시 메모리의 프로그램이 동작한다.

5) 전원부

CON8을 통하여 12V의 전원이 인가되면, LM7805에 의하여 5V 신호로 레귤레이팅(Regulating)된다. 5V 신호는 다시 TPS733Q 소자에 의하여 3.3V 신호로 변환되어 DSP에 입력된다. DSP의 A/D 15번 채널에는 12V 신호가 직접 입력되어 전원의 안정성을 체크하며, DSP의 VCCP 핀에는 플래시 메모리의 쓰기 동작을 위하여 특별히 5V가 입력되어야 한다.

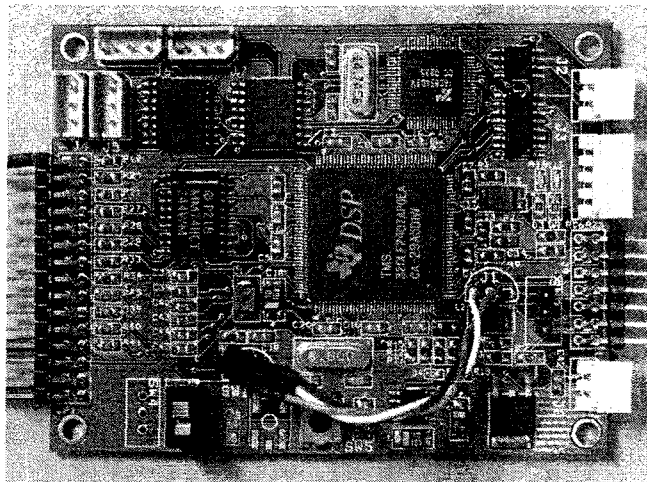


그림 3.3.17 실제 구현된 비행제어컴퓨터

4. 기능별 모듈의 종류

각각의 모듈에는 위에서 언급한 프로세서를 기본적으로 탑재하고 있다. 각 모듈은 CPU와 CAN PORT를 공통적으로 사용하게 되며 자체 상태진단용 센서 설치를 위한 A/D컨버터기능을 사용하고 있다. 비행제어부에서는 연산기능만을 수행하므로 외부 I/O를 사용하

지 않지만 센서 획득부나 조종면 신호 생성부의 경우 A/D 컨버팅기능, RS232수신기능, 카운터 기능을 사용하게 된다.

가. 센서입력부

센서입력부의 기본 기능은 센서신호를 획득하는 것이다. 센서신호는 전압이나 전류의 아날로그 형태로 제공되지만 센서데이터를 이미 A/D컨버전을 수행한 후 신호 변환을 통해 일정한 필터까지 설치되어 나오는 IMU와 같은 모듈 타입의 센서는 디지털 패킷형태의 신호가 제공된다. 아날로그 신호는 센서입력부에서 아날로그 디지털 변환기(A/D 변환기)를 사용하여 획득한다. A/D 변환기는 일정범위의 전압을 수천개의 기준전압으로 나누고 입력으로부터 들어오는 전압값과 가장 비슷한 기준 전압값을 선택하여 이를 디지털 값으로 출력해주는 장치이다. A/D 변환기는 일정한 전압 범위를 일정한 해상도를 가진 디지털 값으로 변환하여 주므로 센서의 정밀도와 측정범위를 고려하여 A/D변환기에 연결하여야 한다. 예를 들면 0~5V의 범위를 10비트의 해상도로 측정하는 A/D 변환기에 대해 측정가능한 최소범위는 $5V/2^{10}$, 즉 0.00122mV가 된다. 즉 이와 유사한 최소측정 단위를 가지는 센서를 부착해야 최적의 성능을 발휘할 수 가 있다. 이러한 센서를 구하기 어려운 경우 A/D 변환기와 센서사이의 해상도와 측정범위를 일치시키기 위하여 센서신호를 증폭하는 과정을 사용하는데 이를 시그널 컨디션과정이라 한다. $\pm 360\text{deg/sec}$ 을 0~5V에서 제공하는 자이로의 경우 본시스템이 측정하고자하는 측정범위가 $\pm 30\text{deg/sec}$ 으로 한정되어 있다고 하면 A/D 변환기의 1/12정도의 측정범위 밖에 사용하지 못하고 해상도 역시 현저하게 낮아지게 되므로 센서에서 제공되는 신호를 12배정도 증폭하여 넣어주게 된다. 이 경우 센서 노이즈도 증폭되므로 저주파 통과 필터를 설치하여 노이즈를 억제하는 과정도 진행해 주어야 한다.

디지털 패킷 형태의 신호를 제공하는 경우 대부분 RS232 시리얼 포트로 전송을 하게 된다. 디지털 패킷은 8bit 단위의 0과 1을 신호를 조합하는 방식으로 원하는 데이터를 전송한다. 이때 같은 신호가 연속적으로 전달될 경우 각각의 데이터를 분리하기가 어려우므로 0에서 1로 변화는 것을 1로 1에서 0으로 변하는 것을 0으로 변환하여 전송하는 맨체스터방식의 통신을 사용한다. RS232통신을 위해서 센서입력부에서는 RS232신호의 전압 레벨을 보드상의 전압레벨로 변환해주는 장치와 신호를 해석하고 8bit 디지털 신호로 변환해 줄 UART라는 장치를 설치해야 한다. UART에서는 직렬포트를 통하여 전송된 데이터를 병렬포트로 전달해 줄 레지스터를 가지고 있는데 CPU에서 UART에 접근하면 병렬버스를 통하여 획득한 데이터를 넘겨주게 된다. 일반적으로 데이터가 도착하면 UART에서 인터럽트를 통하여 CPU에게 알리고 CPU는 인터럽트 루틴을 수행하여 데이터를 전달받는 과정을 수행한다.

센서입력부의 또다른 기능은 센서값을 필요로 하는 모듈에 획득한 센서값을 전송하는 것이다. 시스템이 다중화 되어 있기 때문에 센서입력부는 센서값을 3개의 비행제어부에게 각각 전달하여야 한다. 센서를 직접 비행제어부에 연결하지 못하는 이유는 센서의 전력제한이나 전압불균형으로 인한 전류생성으로 인해 하나의 센서를 여러 모듈에 분기해서 동시에 연결할 수가 없다. 따라서 여러 개의 모듈에서 센서값을 획득하기 위해서는 하나의 모듈에서 센서를 측정하고 이 값을 다른 모듈로 직렬버스나 병렬버스를 이용하여 전송하는 과정을 거쳐야 한다. 또한 여러 모듈에서 동시에 센서를 분기해서 측정할 수 있다고 하더라도 고장 발생시 이 센서부분을 분기 하려면 각 모듈이 각각 분리작업을 수행하여야 하므로 고

장을 분리하기가 쉽지 않다. 즉 하나의 모듈에 프로세서를 탑재하여 센서 획득을 전담하는 기능을 할당하게 되면 고장발생시 이 모듈에서만 신호를 분리하면 되므로 여러가지 일련된 과정이 간단하게 처리된다.

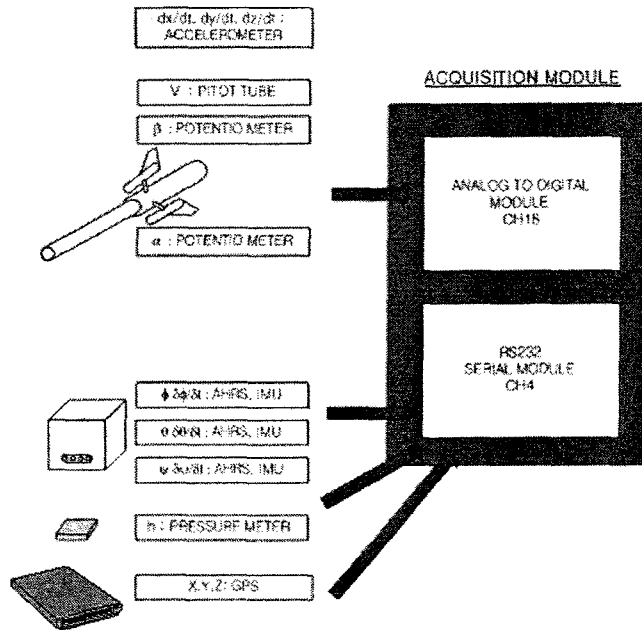


그림 3.3.18 센서입력부 인터페이스

나. 비행제어부

비행제어부에서는 획득한 센서값과 항법 정보 및 고장관리 정보를 통해서 원하는 제어신호를 계산해 내는 기능을 수행한다. 비행제어부에서는 비행상태에 따른 계인 스케줄 설정, 경로점을 순서대로 진행해나가는 항법 관리, 고장 진단 및 재형상 제어 등의 비행에 필요한 일련의 제어루틴을 수행한다. 이 모듈에서 수행하는 지능형 비행제어에 대한 알고리즘 설명은 3장에서 자세히 설명하고 있으므로 이장에서는 설명을 생략하기로 한다.

다. 제어신호생성부

이 모듈은 계산된 제어신호에 따라 제어면을 구동시켜 주는 역할을 수행한다. 비행제어부에서는 각종센서를 이용하여 계산된 제어각도를 전달해 주면 제어신호생성부는 구동면이 제어각을 따라갈 수 있도록 저레벨 제어를 수행한다. 제어신호생성부에서 관리하는 제어면은 기본적으로 추력밸브와 에일러론, 러더, 엘리베이터이다. 그러나 항공기의 RM개념을 도입하여 하나의 조종면이 파손되더라도 피해를 최소화 할 수 있도록 제어가능성을 확보해야 하는데 이를 위하여 카나드나 플랩, 엘리베이터의 좌우 별도 구동, 에일러론의 좌우 별도 구동 등의 조종면 여분을 두어 추가로 제어하게 된다.

본 연구에서 구동면을 제어하기 위해 사용하는 신호는 펄스 폭 변조(Pulse Width Modulation)신호이다. 한주기에 하나의 사각파를 생성하고 이 사각파의 듀티비율을 변경시켜서 서보모터를 제어한다. 예를 들면 duty 비가 5%일 때 서보모터가 좌측으로 최대로 향하게 되고 15%에서는 우측으로 향하게 된다. 이 신호를 생성하기 위하여 제어신호생성부에서는 카운터를 설치하게 된다. 카운터는 내부 발진 펄스를 세는 장치인데 주기적인 신호

를 발생시키는 오실레이터를 설치하고 이 신호를 세계 되면 타이머의 역할도 수행해 낼 수 있다. 카운터에는 여러 가지 기능이 있는데 본 연구에서는 ON 상태에 있던 신호가 카운터에 입력한 만큼 펄스를 세계되면 OFF 상태로 떨어지는 기능을 사용한다. 원하는 제어면 각에 해당하는 듀티비에 맞도록 카운터 레지스터에 카운팅값을 전달하면 카운터는 ON-OFF 시그널을 원하는 듀티비로 생성한다. 예를 들면 엘리베이터 조종면을 아래로 15°로 인가하는 조종을 원한다면 50Hz의 듀티비가 5%인 사각파가 생성되어야 한다고 하자. 4Mhz의 펄스를 오실레이터에서 카운터로 인가하고 80000회의 펄스를 셀 때 마다 카운터를 클리어하도록 설정하면 카운터 자체는 50Hz의 주기를 가지게 된다. 카운터 모드를 클리어할 때 ON 신호를 발생하도록 하고 CPU에서 전달해주는 카운트 수를 카운터 레지스터에서 입력받아 현재 카운터 값과 레지스터에 있는 값을 비교하여 카운터 값이 더 크게 되면 OFF신호를 발생하도록 한다. 5%에서 15°의 인가각이 나오므로 CPU에서는 이에 해당하는 카운팅 횟수로 4000(80000×0.05)회로 전달하면 원하는 사각파를 생성할 수 있다.

제어면에는 각도센서가 부착되어 제어신호생성부에서 제어면에 관한 low level 피드백을 통하여 제어면 제어를 수행한다.

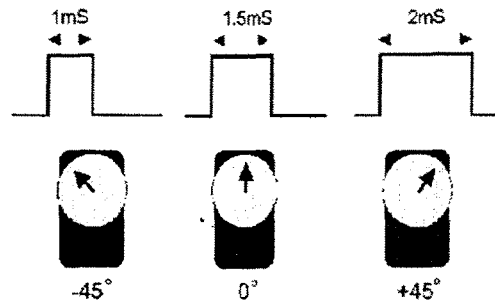


그림 3.3.19 서보모터의 PWM방식의 신호처리

라. 버스 관리 부

각 모듈간의 정보전달이 이루어지도록 버스를 관리해주는 역할을 수행한다.

다중화 개념을 도입하게 됨으로서 기존의 다일 프로세서 제어기와는 달리 새롭게 구성되는 기능이 된다. 모듈이 하나의 보드에 탑재되지 않고 외부 버스를 사용하게 됨으로서 모듈간의 통신에 대한 교통정리가 불가피해진 것이다. 시스템의 경우 하나의 버스라인을 여러 모듈이 공유해야 하므로 일정시간을 할당하여 그 시간안에 한 모듈에서 다른 모듈로 정보를 전달해야 한다. 모듈마다 통신할 수 있는 시간을 할당하는 것이 버스 관리부의 중요한 임무가 된다.

5. 데이터 버스 시스템

가. CAN의 개요

CAN버스는 마이크로 컨트롤러들 간의 통신을 위하여 설계되었다. 자동차 분야에서 많이 사용되는데 엔진 관리 시스템, 변속장치 제어, 계기 팩, 차체 전자기술등의 온보드 전자 어

장치 간의 정보 교환에 사용된다. 이론적으로 하나의 단일 네트워크는 최대 2032 디바이스들이 한 개의 CAN 버스(한 개의 ID 번호를 가진 한 개의 노드를 가정)에 연결될 수 있으나 하드웨어 (즉, 송수신기)의 현실적인 제한으로 인해, 실제적으로는 한 개의 단일 네트워크에 최대 110개 노드들을 (필립스 82C250 CAN 컨트롤러를 사용) 허용하고 있다. CAN은 최대 1 Mbit/sec의 데이터 통신을 제공하여, 실시간 제어를 수행할 수 있다. 오류 제한(Error Confinement)과 오류 탐색(Error Detection) 기능들은 noise-critical 환경들에서의 신뢰성을 더욱 높여주고 있다.

Controller Area Network는 원래 1980년대 후반에 자동차 산업을 위해 독일 회사 Robert Bosch GmbH에 의해 개발되었다. 그들이 CAN을 개발하게 된 동기는 - 고객들의 그들 차량에 대한 더욱더 많은 기능상의 요구들이, 대부분은 전자식으로 작동되며 더욱 많은 배선을 의미하는, 또 다른 온-보드 시스템과의 어떤 통신 형태를 필요로 하는 것이기 때문에, 현대 자동차에서의 내부-ECU 통신에 필요한 꾸준히 증가하고 있는 거대한 배선 작업의 문제에 대한 해결책을 제공하기 위해서였다. 그들의 결론은 모든 온-보드 주변장치들이 부착될 수 있는 하나의 단일 네트워크 버스를 설계하는 것이었다. 1993년 CAN은 표준 ISO 11898(고속 애플리케이션용)과 ISO 11519(저속 애플리케이션용)가 되었고 이것은 다중-마스터 직렬 통신 버스로 이것의 기본 설계 사양은 고속, 높은 잡음 면역성, 그리고 오류-검출 기능들에 적합하게 되어있다. 이러한 기능들의 결과로, CAN 버스는 또한 제조 산업과 항공 우주 산업들에서도 폭넓게 사용되게 되었다.

나. CAN의 구조 및 물리적 프로토콜

개방시스템 상호연결 모델(Open System Interconnecton Model)은 모든 종류의 컴퓨터 시스템 간의 통신을 허용하는 네트워크 시스템의 설계를 위한 계층 구조이다. 이 모델은 서로 연관된 7개의 계층으로 구성되어 있고 각 계층에는 네트워크를 통해 정보를 전송하는 일련의 과정이 규정되어 있다. OSI는 물리층(제1층), 데이터 링크층(제2층), 네트워크층(제3층), 전송층(제4층), 세션층(제5층), 표현층(제6층), 응용층(제7층)으로 구성되어 있다. 이 중 1,2,3층인 물리층, 데이터 링크층, 네트워크층은 한 장치에서 다른 장치로 데이터를 이동할 때 필요한 물리적인 면(전기적 규격, 물리적 연결, 물리주소, 전송시간과 신뢰도 등)을 처리한다. 제 5,6,7계층인 세션층, 표현층, 응용층은 사용자 지원계층으로 생각될 수 있다. 이들은 서로 상관없는 소프트웨어 시스템간의 상호 연동이 가능하게 한다. 전송층인 제4층은 지금 말한 두 그룹을 연결하고 하위 계층에서 전송한 것이 상위계층이 사용할 수 있는 형태가 되도록 보장한다.

CAN 통신 프로토콜은 CAN 버스에서 디바이스들 통신 사이로 데이터가 전달되는 방법을 명시한다. 이것은 ISO의 OSI를 따르며, 통신 네트워크 표준인 일곱 계층으로 되어 있다. 이 OSI 모델은 두 개 네트워크 노드들 간의 총화된 통신 시스템을 기술하며, 이론상 각 계층은 로컬 모드에서는 오직 자신의 직접적인 위, 아래의 계층들과 통신할 수 있으며, 원격 모드에서는 동등한 계층과 통신할 수 있다. OSI 모델의 계층들은 아래의 표에 나와 있다. 사실 CAN 프로토콜은 OSI 모델의 가장 낮은 두 개 층들로 설명될 수 있다. 데이터 링크 계층과 물리적 계층. 애플리케이션 계층 프로토콜들은 개별적인 CAN 사용자들에 의해 개발된 독점 구조, 또는 특정 산업 내에서 사용되는 신생 표준들 중의 하나가 될 수 있다. 자동차 산업에서 많은 제조업체들은 그들 자신의 독점적인 표준을 사용하고 있다.

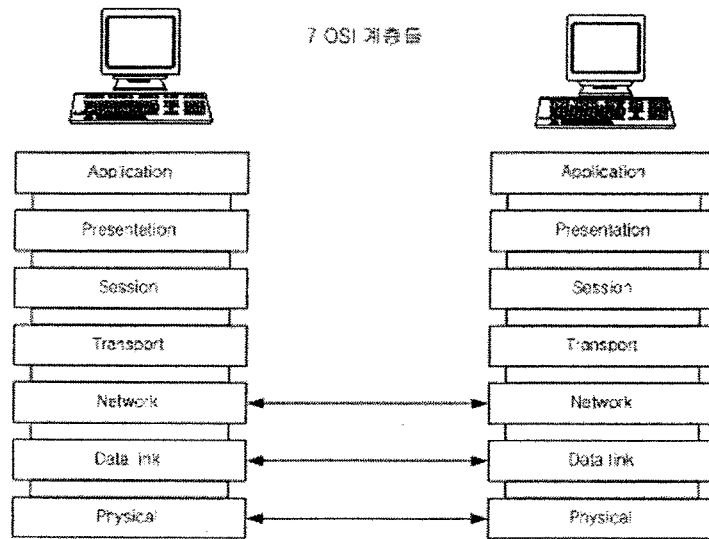


그림 3.3.20 OSI 계층도

CAN에서 사용하는 프레임에는 다음과 같이 4개의 타입의 프레임이 있다.

Data Frame : 데이터를 전송하는 프레임

Remote Frame : 하나의 노드에서 다른 노드로 데이터 송신을 요구하는 프레임(RTR로 구분)

Error Frame : 전송중에 에러가 발생하게 되면 즉시 에러프레임을 발생시켜 노드들에 알림

Overload Frame: CAN 회선이 전송중이어서 다른 프레임의 진입을 미루기위하여 전송하는 프레임

CAN은 노드간 통신을 위하여 데이터 프레임을 구성한다. 데이터 프레임의 구조는 표준 ID만을 사용하느냐 확장 ID를 사용하느냐에 따라 달라진다. 다음 그림은 확장 ID까지 하였을 경우에 데이터 프레임을 보여주고 있다.

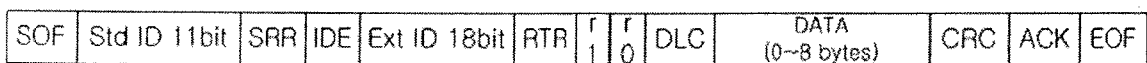


그림 3.3.21 CAN의 데이터 프레임

- SOF : 프레임의 시작을 알리는 비트
- STD ID : 프레임이 전송될 주소를 가지는 비트
- SRR : 다른노드에 전송을 요구하는 비트
- IDE : 확장 주소의 사용

RTR	: 데이터 프레임과 리모트 프레임을 구분하는 비트
r1,r0	: 예약 비트
DLC	: 데이터의 길이를 나타내는 코드
DATA	: 전송하고자 하는 데이터
CRC	: Cyclic Redundancy Check
ACK	: 2비트로 구성되어 전송을 확인하는 비트
EOF	: 프레임의 끝을 알리는 비트

다. 소프트웨어 프로토콜 (요소기술 및 구현기술)

시스템간 통신을 하기 위해서 양 모듈에서 신호를 분석할 수 있는 규약을 결정해야 한다. 이는 상호간 전압레벨과 한 바이트를 보낼 때 데이터 비트수 와 패리티 비트, 정지비트 및 속도와 같은 물리적인 전달 규약에서부터 바이트를 받았을 때 이를 문자로 해석할 것인지 이를 해독하여 그림의 형태로 해석할 것인지에 대한 소프트웨어적인 규약도 필요하다. 물리적인 프로토콜의 경우 국제적인 표준이 결정되어 있기 때문에 사용자가 임의로 변경할 수 없지만 소프트웨어적인 규약은 전달 정보의 내용이나 특성에 따라 시스템에 최적화 시켜야 하므로 사용자의 의도에 맞게 규정되어야만 한다. 본 연구에서 고려할 사항은 50Hz의 샘플링 시간동안 한주기의 시스템통신을 모두 끝낼 수 있도록 시스템간의 통신 데이터 패킷을 최소화 하면서도 센서 데이터 정보나 고장정보, 조종면 신호등의 최소해상도를 보장해 주도록 해야 한다는 것이다. 대부분의 센서들은 해상도를 일정하게 유지하고 있기 때문에 해상도에 맞추어 전송시에 필요한 데이터 비트수를 결정한다. 16비트의 해상도를 전달하기 위해서는 2byte신호를 구성하여 16진수로 바꾸어 전달하는 방식을 사용한다. 2byte는 0x00에서 0xFF까지 총 65536가지의 조합을 만들어 낼 수 있으므로 16비트의 센서 신호를 그대로 전달 할 수 있는 것이다.

데이터 프레임은 물리층에서 데이터 전송을 하기 위해서도 필요하지만 응용층에서도 전달된 데이터가 어떤 정보를 의미하고 있는지를 구분하기 위해서 필요하게 된다.

본 연구에서는 센서입력부에서 비행제어부로 전달하는 데이터, 비행제어부에서 제어신호 생성부로 전달하는 데이터의 기본 2종류의 프레임이 상시 전달되고, 다시 제어신호 생성부에서 비행제어부로 전달하는 데이터, 비행제어부에서 비행제어부로 전달하는 데이터, 버스 컨트롤러에서 백업 버스 컨트롤러로 전달하는 3가지 비상 데이터가 모듈의 자체진단 시 이상이 발생하였을 때 비상호출 방식으로 전달된다.

센서입력부에서 비행제어부로 전달되는 패킷은 센서입력부에서 설정한 센서 신호에 따라 롤각속도, 피치각속도, 요각속도, 툴각, 피치각, heading각, x가속도, y가속도, z가속도, x좌표, y좌표, 고도, 공기와의 상대속도, 받음각, 옆미끄럼각으로 구성된다. 각각의 데이터는 x,y좌표를 제외하고 2바이트로 구성되어 있다. x, y좌표는 범위가 매우 넓기 때문에 4바이트를 사용한다. 센서입력부에서 비행제어부로 전달되는 신호는 헤더(1바이트), 각센서 신호(13개의 2바이트 신호와 2개의 4바이트 신호), 체크섬(1바이트) 패킷은 36바이트로 구성된다.

비행제어부에서 제어신호 생성부로 전달되는 패킷의 내용은 추력, 에일러론각, 러더각, 엘리베이터 각으로서 각각 2바이트를 차지하게 된다. 차후에 카나드와 플랩과 같은 여분의 조종면을 제어할 필요가 있게되면 패킷에 추가로 첨가될 것이다. 역시 마찬가지로 헤더와,

조종면각, 체크섬으로 구성되어 10바이트로 구성된다.

라. 버스 운영 방법

기본적으로 버스의 구동방식은 Round Robin 방식과 Time Based 방식 두 가지 방식이 있다. Round robin 방법은 주소를 하나씩 올라가거나 내려가는 방식으로 순서대로 돌아가면서 송신권을 가지는 구동방식이다. 토큰링이라고도 하는 데 토큰이 바로 송신권으로서 토큰을 바로 옆의 모듈로 넘겨주면 토큰을 받자마자 원하는 데이터를 넘기게 된다. 만일 넘겨줄 데이터가 없다면 자동적으로 다음 모듈로 토큰을 넘기게 된다.

다른 하나는 시간 분할 방식의 메카니즘으로 각각의 모듈에 전송할 수 있는 시간을 주기적으로 할당해주는 방식이다. 시간분할 방식은 전송하려는 데이터 양에 따라 시간을 가변적으로 모듈에 할당해 줄 수 있지만 서로 다른 시간을 할당받게 되면 관리하기가 복잡해진다. 시간을 할당하는 방법은 타이밍을 관리해주는 메인 버스 컨트롤러가 타이밍 신호를 각 모듈에 전달해주는 방식과 처음 초기화 시점에서 전 모듈에 데이터를 전달하는 브로드 캐스트 방식의 명령으로 동기화 시킨 후 일정시간마다 스스로 데이터를 전달하는 방식이 있다. 첫 번째 방식은 운영이 중앙집중형이 되기 때문에 관리하기가 쉬운 장점이 있으나 이 모듈의 고장시 시스템의 전체적인 장애가 발생한다는 점에서 약점이 발생한다. 이러한 약점을 보완하기 위하여 메인 버스 컨트롤러를 감시하는 백업 컨트롤러를 두어 메인 컨트롤러에서 동기화 신호를 보내주지 않을 경우 대신해서 동기화 신호를 생성하는 방법을 사용한다. 메인 버스 컨트롤러는 기존 제어시스템과는 별도로 탑재 하여도 되나 기존 비행제어 모듈중 하나가 역할을 수행하여도 된다. 두 번째 방법은 일정한 메인 모듈이 존재하지 않고 오직 각자의 모듈에서 생성하는 동기신호를 사용하여 할당된 시간에 메시지를 보내주므로 어느 모듈의 파괴가 시스템 마비로 연결되지 않는다. 그러나 시스템의 모듈추가나 다른 기능을 첨가할 때 타임 테이블을 각각의 모듈마다 다 다시 프로그래밍 해주어야 하는 단점이 있으므로 시스템 변경이 매우 까다롭게 된다.

정보전달 방향	정보 전달 내용
센서입력부 1 → 비행제어부 1, 2, 3 센서입력부 2 → 비행제어부 1, 2, 3 센서입력부 3 → 비행제어부 1, 2, 3	센서입력내용 : 롤각속도, 피치각속도, 요각속도, 롤각, 피치각, 헤딩각, x가속도, y가속도, z가속도, x좌표, y좌표, 고도, 공기와의 상대속도, 받음각, 옆미끄럼각, 센서 Built in test 정보
비행제어부 1 → 제어신호생성부 1~N 비행제어부 2 → 제어신호생성부 1~N 비행제어부 3 → 제어신호생성부 1~N	조종면 각 정보 : 추력, 엘리베이터, 에일러론, 러더

마. BUS FAULT

버스 라인상에 발생하는 고장에 관해서는 버스를 회복하려는 시도는 고려하지 않는다. 따라서 버스 자체의 물리적인 구조는 하나의 라인에서 발생하는 고장에서도 시스템 전체가 영향을 받지 않도록 중복된 라인의 형태로 구성되어야 한다. 고장이나 오류 검증은 메시지 동기화와 체크섬 메카니즘을 이용하여 얻어낸다. 기본적으로 순간적인 고장은 CRC 전송된 체크섬과 자체적으로 연산한 체크섬을 비교함으로써 감지하고 영구적인고장은 동기화 장치를 통하여 신호가 지연될 경우를 감지하고 CRC체크를 통하여 잘못된 신호만을 한계시

간 동안 주고있다고 판단함으로써 감지할 수 있다.

버스는 3중의 물리적 라인을 가지게 된다. 이는 물리적으로 전송라인이 단절되면 시스템 전체가 마비 되는 것을 방지하고자 하는 것이다. 버스는 각각 3개의 송신 버퍼로 같은 내용의 데이터를 전달하고 물리적으로 다른 라인을 통하여 다른 모듈의 3개의 수신 버퍼로 각각 전달한다. 양 버퍼에 도달한 데이터를 상호 비교하여 다른 값이 도착하였을 때는 재전송 신호를 보내고 할당된 시간이 남지 않았을 경우 각각의 패리티 비트를 확인하여 원하는 패리티 비트를 가지는 신호를 선택한다. 버스관리에 있어서 실시간 데이터 전송이 데이터의 고장 유무 확인을 우선하므로 순간적인 고장으로 인해 데이터를 재전송 받을 필요가 있다고 하더라도 할당된 시간을 초과하면 다음 모듈로 전송권을 넘기게 된다. 버스관리에 있어 정보전달의 시간을 엄수하는 것이 정보를 전달하는 것보다 우선한다.

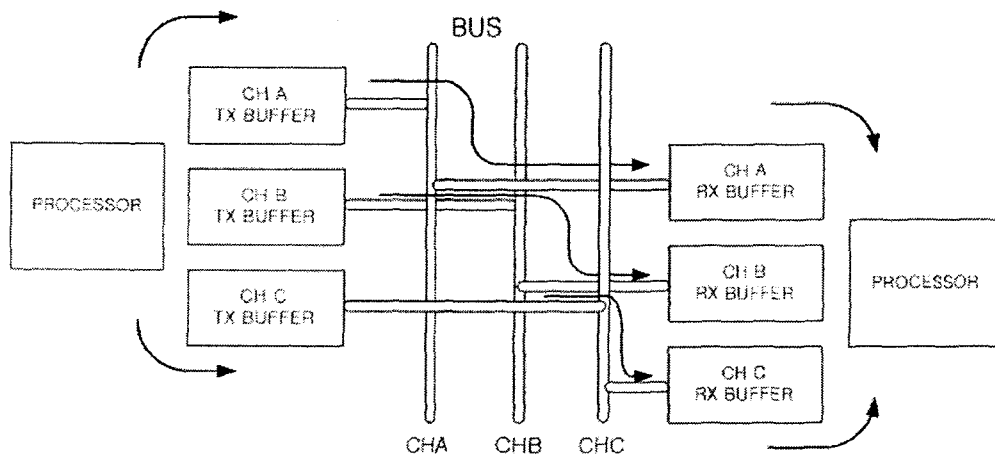


그림 3.3.22 3중화 버스

6. 다중 비행제어시스템 프로그래밍

4 절에서 언급한 비행제어시스템을 운영하기 위해 프로그램을 개발하였다. 아래 그림은 시스템 정보흐름에 대한 개념도이다. 각각의 센서입력부에서 3중화된 비행제어부로 센서정보를 넘겨주면 비행제어부에서는 센서값을 교차채널 방식을 통하여 검사하고 제어신호를 만들어 낸다. 센서모니터와 비행제어부 모니터, 구동기 모니터 정보를 고장 관리부에 전달 되면 고장관리부는 정상,고장 여부를 검증하여 모드스위칭을 수행한다. 각각의 비행제어부에서 만들어진 제어신호는 각각의 구동기에 설치된 제어신호생성부에 전달되는데 여기에서 세 개의 비행제어부의 신호를 다시 비교하여 비행제어부의 정상여부를 검증한다. 제어신호생성부에서는 제어신호와 PWM신호를 이용하여 구동기를 제어하는데 각각의 구동기에는 각센서가 설치되어 있어 인가 신호와 각센서의 신호를 비교함으로써 구동기 정상여부를 상시 검사하고 있다. 프로그램은 C와 C++로 작성되었으며 string(문자열관련), math(수학관련), pcl4c(RS232 통신), graphic(디스플레이 관련), 240x(DSP 레지스터 관련) 등의 공개 라이브러리를 사용하였다.

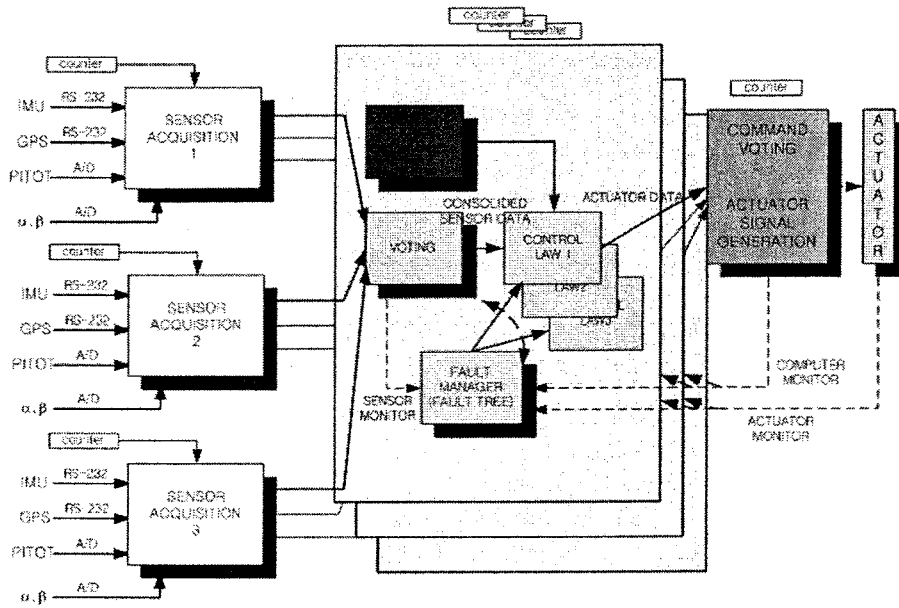


그림 3.3.23 다중 비행제어시스템 알고리즘

가. 센서입력부 프로그램

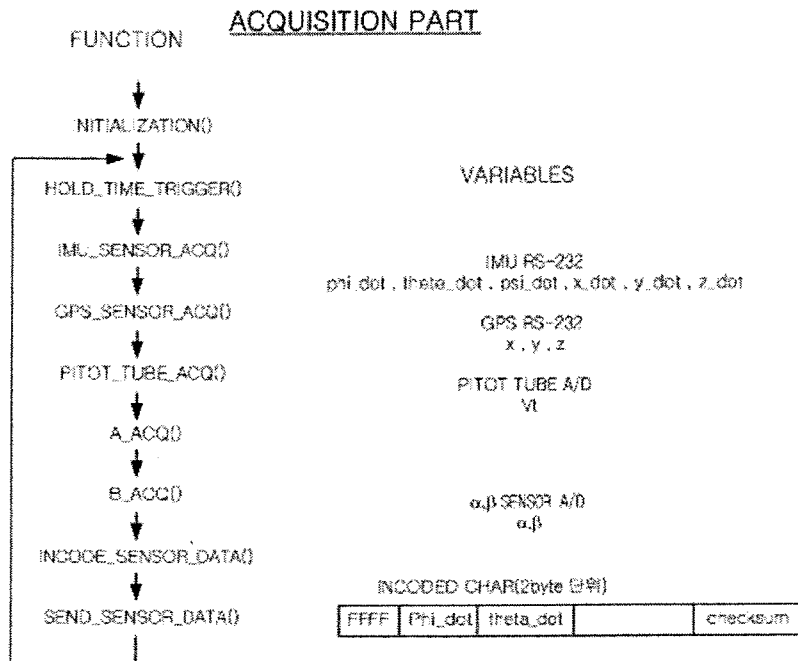


그림 3.3.24 센서입력부 프로그램

센서입력부의 프로그래밍은 다음과 같은 순서로 이루어진다.

초기화 : 센서입력부의 전원을 가동하면 센서 입력부에서 사용하는 여러 디바이스들에 대한 초기화를 시작한다. 센서입력부에는 A/D converter와 RS232 포트, 동기화를 위한 카운터 등의 디바이스가 탑재되어 있으므로 각각의 디바이스에 해당하는 주소를 할당하고 각각

의 디바이스에서 사용하고자 하는 기능을 선택하는 과정을 수행한다.
 각 센서의 해상도 및 측정범위는 다음과 같다.

표 3.3.2 센서사양

신호	측정범위	측정방식	해상도
속도	0 ~ 500m/s	A/D Conv	10 bit
받음각	-20~50deg	A/D Conv	10 bit
미끄럼각	-40~40deg	A/D Conv	10 bit
롤 각속도	±300 deg/s	RS-232	14 bit
피치 각속도	±300 deg/s	RS-232	14 bit
요 각속도	±100 deg/s	RS-232	14 bit
x축 가속도	10g	A/D Conv	14 bit
y축 가속도	10g	A/D Conv	14 bit
z축 가속도	10g	A/D Conv	14 bit
x 위치	∞	RS-232	32 bit
y 위치	∞	RS-232	32 bit
고도	0~30000ft	RS-232	32 bit

동기화 : 센서 입력부에서는 50Hz의 동기화를 수행하기 위하여 트리거 신호가 들어올 때 까지 루프를 중단하고 기다리는 부분이 있다. 시스템 동기화는 시스템 전체 초기화 수행시 타이머를 일제히 0으로 세팅함으로써 수행한다.

센서 신호 획득 : 50Hz 마다 들어오는 동기시그널을 받으면 차례로 IMU, GPS, 피토투브, α, β 센서 등에서 신호를 받아 메모리에 보관한다. 센서신호는 신호가 들어오기를 계속 기다리는 것이 아니라 FIFO 버퍼에 센서신호가 들어오는 대로 계속 신호를 받아들이고 CPU에서 불러들일 때만 버퍼를 호출하여 한번에 불러들여 센서신호를 받는 작업에 시간을 최소로 할애한다.

인코딩 : 보관된 센서 내용은 비행제어부로 전달하기 위하여 디코딩을 수행한다. 센서에서 얻은 데이터를 한 바이트씩 나누어 문자열을 만들고 이를 CAN의 전송층 라이브러리를 사용하여 전달함으로써 센서입력부의 인코딩 과정을 수행한다.

데이터 전달 : 디코딩된 문자열을 CAN의 송신버퍼로 보내 데이터를 전달한다.

그림 x.x는 소프트웨어의 흐름도를 보여주는 그림이며 사용된 함수명을 나열하였다.

나. 비행제어부 프로그램

초기화 : 비행제어부도 전원을 인가하면 각각의 기능에 해당하는 디바이스를 초기화한다. 비행제어부에서는 주로 연산만을 수행하므로 비행제어부에서 사용하는 디바이스는 외부와 연결하는 CAN 포트밖에 없다. 따라서 비행제어부의 초기화는 CAN Port의 설정과 타이머 설정, 각종 연산수행을 위한 변수 초기화 만을 수행한다.

동기화 : 초기화가 끝나면 트리거 신호를 기다리게 된다. 트리거 신호는 50Hz마다 생겨

나며 한 싸이클의 작업이 50Hz 이전에 모두 완료되면 다음 트리거 신호를 받을 때 까지 대기한다.

센서 신호 수신 : 신호를 받게 되면 센서입력부1,2,3을 통하여 차례로 센서신호를 받게 된다.

디코딩 : 센서입력부에서 얻은 센서신호들은 문자열의 형태로 받게 되므로 이를 이용하여 연산을 수행하기 위하여 각각의 신호를 디코딩하여 float 형 실수로 변환한다. 디코딩은 문자열의 헤더와 마지막 체크섬 바이트를 제외한 모든 바이트를 더하여 256의 나머지와 체크섬 바이트를 비교하여 동일하면 시작한다. 헤더의 다음 바이트부터 2바이트씩 나누어 센서값을 분리한다(단 x,y,좌표는 4바이트로 나눈다). 이를 공용체를 통하여 2바이트 캐릭터에서 short데이터로 바꾼 후 스케일 인자를 곱하여 원하는 물리량 값으로 바꾼다.

교차채널 감시 : 3개의 센서신호부에서 수거한 센서신호들을 교차채널 방식을 사용하여 센서를 검사하고 필요한 센서값을 골라낸다. 교차채널 방식은 세 개의 센서의 중간값을 기준값으로 잡고 기준값이 아닌 다른 센서의 값들이 문턱값을 한계시간 이상 벗어나게 되면 고장으로 인식하는 알고리즘이다. 고장정보를 다루기 위하여 문턱값을 넘어갔음을 검사하는 변수와 연속적으로 문턱값을 넘어간 시간을 검사하는 한계시간 변수를 설정하고 함수가 호출되더라도 초기화 되지 않게 하기 위하여 외부변수로 두어 관리한다. 각 센서에 대한 문턱값과 한계시간은 다음과 같다.

표 3.3.3 센서 문턱값 및 한계시간

신호	문턱값	한계시간
속도	10m/s	109.2 ms
받음각	-2.5~9°	109.2 ms
미끄럼각	-9~9°	109.2 ms
롤 각속도	±40deg/sec	100 ms
피치 각속도	±9deg/sec	100 ms
요 각속도	±9deg/sec	100 ms
x 가속도	0.25G	100 ms
y 가속도	10G	100 ms
z 가속도	0.75G	100 ms
x 위치	50m	100 ms
y 위치	50m	100 ms
고도	10m	100 ms

제어 : 위의 교차채널 알고리즘을 수행한 후에 결정된 센서 값을 이용하여 현재 비행상태를 파악한 후 계인 스케들을 수행한다. 이 과정에 앞서 기존의 입력과 출력의 데이터나, 하드웨어로부터 직접들어오는 신호등을 바탕으로 고장진단을 수행하며 정상상태에서는 정상모드로서 임무를 수행하고 고장으로 판별되었을 경우에는 고장분리 및 재형상제어를 수행한다. 정상상태일 경우 비행영역을 속도, 고도변수로 분할하여 비행모드를 결정하고 이에 해당하는 제어이득을 선택한다. 이 때 자동 비행일 경우 항법관리 시스템으로부터 유지해야할 기준 비행자세에 대한 정보를 얻어서 제어를 수행하며 방향 및 임무 등에 맞는 유도법칙을 적용하여 임무를 완수할 수 있도록 하는 다양한 알고리즘을 적용한다. 고장 상황으로 간주되었을 경우에는 고장의 각각의 상황을 CASE BY CASE로 나누어 이에 해당하는 재

형상제어 로직을 선정한다. 이때 고장상황을 결정하기 위하여 신경회로망이나 퍼지로직을 사용할 수 있고 현재 상황을 파악한 후에 다양하게 연구된 재형상 제어 법칙을 적용한다. 제어 항목의 프로그래밍은 연구가 진행되어 감에 따라 계속 프로그램이 업데이트를 수행할 것이다. 제어 로직을 수행하면 결과로서 조종면 구동각을 계산하게 된다.

인코딩 : 구동각을 제어신호생성부로 전달하기 위하여 인코딩을 수행한다. 각각의 제어면 정보는 2byte로 float를 2byte short형 변수로 변환하여 전달한다.

데이터 전달 : 디코딩된 문자열을 CAN의 송신버퍼로 보내 데이터를 전달한다.

그림 x,x는 비행제어부의 프로그램 흐름을 개념적으로 보여주고 있다.

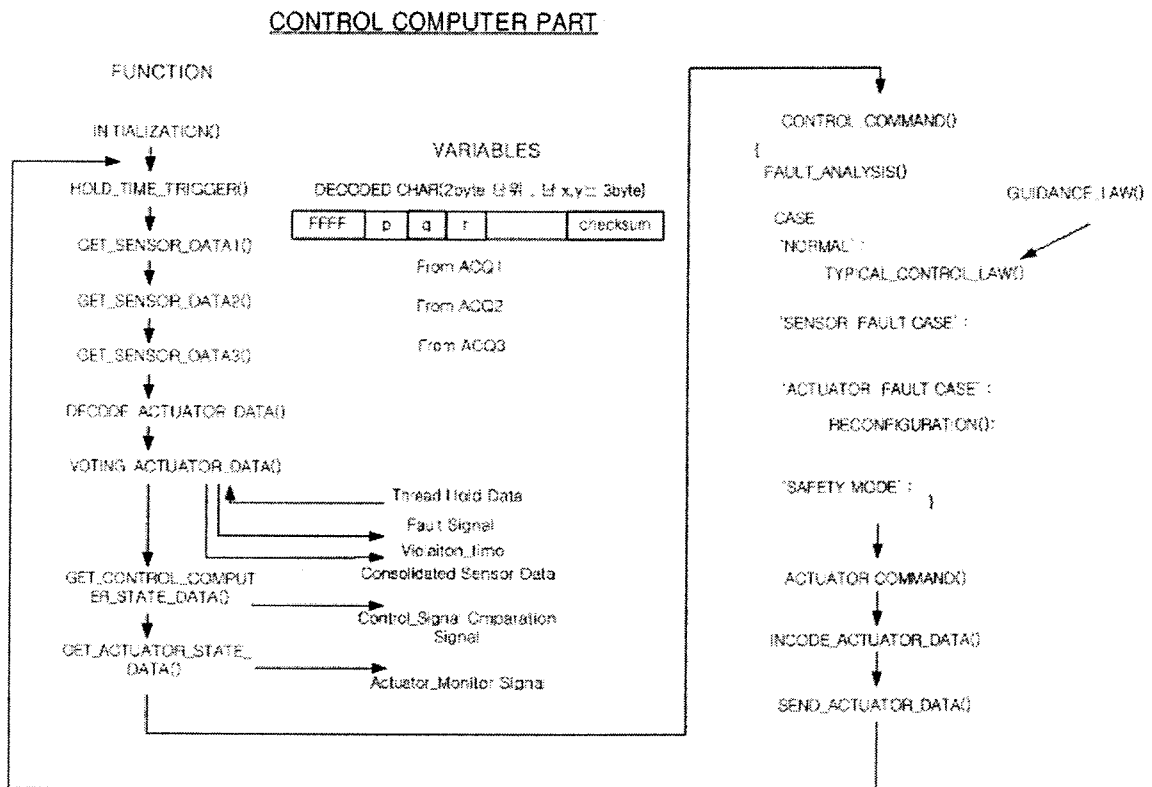


그림 3.3.25 비행제어부 프로그래밍

다. 센서신호발생부

초기화 : 센서신호발생부도 전원을 인가하면 각각의 기능에 해당하는 디바이스를 초기화한다. 센서신호발생부에서는 구동면에 부착된 구동면 각센서의 입력을 받을 A/D 컨버터와 구동면 신호를 생성해 내기 위한 카운터, 상태를 비행제어부로 보내기 위한 CAN포트의 초기화를 수행해 주어야 한다.

동기화 : 초기화가 끝나면 트리거 신호를 기다리게 된다. 트리거 신호는 50Hz마다 생겨나며 한 싸이클의 작업이 50Hz 이전에 모두 완료되면 다음 트리거 신호를 받을 때 까지 대기한다.

센서 신호 수신 : 동기 신호를 받게 되면 비행제어부 1,2,3을 통하여 차례로 센서신호를

받게 된다.

디코딩 : 비행제어부에서 얻은 센서신호들은 문자열의 형태로 받게 되므로 이를 이용하여 연산을 수행하기 위하여 각각의 신호를 디코딩하여 float 형 실수로 변환한다.

교차채널 감시 : 3개의 비행제어부에서 수거한 센서신호들을 교차채널 방식을 사용하여 제어입력값을 검사하고 필요한 제어입력을 골라낸다. 교차채널 방식은 비행제어부에서 센서입력을 선택하는 방법과 동일하다.

구동면 제어신호 생성 : 카운터에 원하는 각을 만들기 위한 카운팅값을 입력함으로서 원하는 duty비를 가지는 PWM신호를 생성하고 이를 서보모터에 전달한다.

구동각 감시 : A/D converter를 통하여 구동각의 정보를 얻고 인가한 구동면 신호와 차이가 많이 벌어지면 고장으로 인식한다.

고장신호 전달 : 구동면 고장발생 여부를 비행제어부로 전달한다. 고장이 발생하지 않으면 아무정보도 전달되지 않으나 고장이 발생하면 신호를 비행제어부로 전달하여 재형상 제어를 수행하도록 요청한다.

ACTUATOR COMPUTER PART

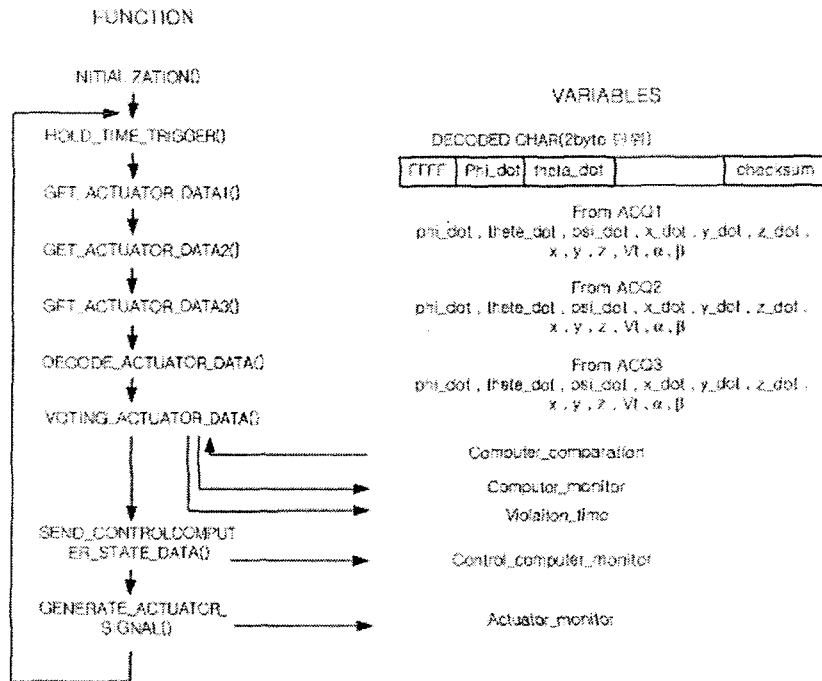


그림 3.3.26 제어신호생성부 프로그래밍

7. 다중화 시스템 보완모델의 해석적 알고리즘 개발

하드웨어적 여분방식을 사용하여 서로간의 센서신호를 감시하는 방법은 여러 센서 신호가 같은 시각에 같은 형태의 고장이 발생하기 어렵다는 점에서 가장 확실하게 신뢰성을 향상시킬 수 있는 방법이지만 중량 증가 및 가격 상승의 요인이 되므로 최적의 운영 방법이라고는 할 수 없다. 또한 본 연구에서 채택한 3중의 하드웨어 여분방식은 앞 x.x에서 실험에서와 같이 1개의 센서 고장에서는 고장 검출 및 분리에서 뛰어난 성능을 보여주고 있으나 1개의 센서의 고장 분리 이후부터는 고장 분리 기능이 작동하지 않는다. 1개의 센서가 고장 난 이후에는 나머지 두 개의 센서에서 획득한 신호의 평균값을 시스템 상태측정치로 선

정하게 된다. 이 값을 기준으로 두 개의 센서가 벗어난 정도를 측정하여 미리 정해놓은 한계값을 벗어나게 될 경우에는 두 개의 센서 신호도 믿을 수 없다는 결정을 하게 되는데 평균값을 기준값으로 취하기 때문에 고장이 검출하는 순간 센서 신호 두 개가 동시에 한계값을 벗어나게 된다. 즉 두 개의 센서만 남게 될 경우에는 고장 검출을 할 수는 있으나 어느 센서가 고장났는지를 알 수 없기 때문에 고장 분리를 할 수 없게 된다.

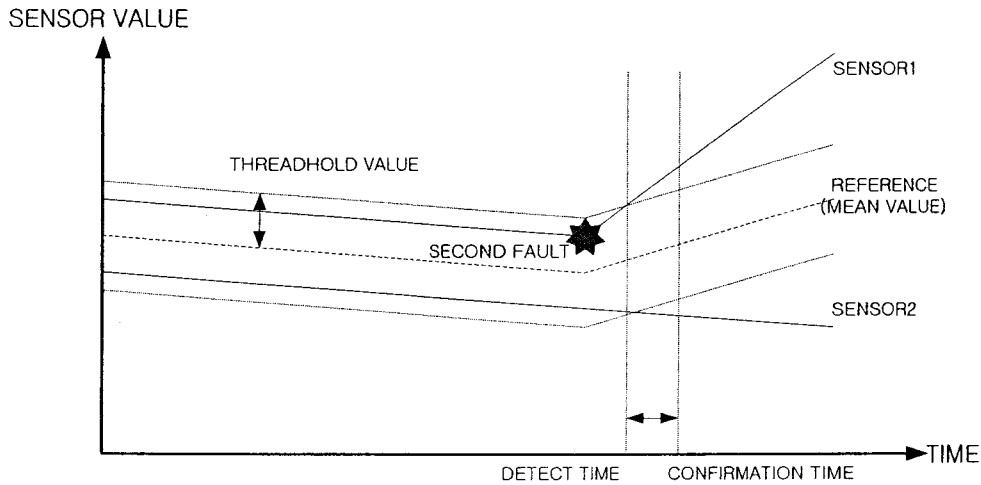


그림 3.3.27 2개의 센서만을 사용하였을 때 고장발생 상황

이런 상황에 직면하게 되었을 때 해결하는 방법으로 다음과 같은 방법이 있다

1) BUILT IN TEST : 고장이 검출되면 해당 센서에 대한 자체 진단을 한다. 고장 검출시 각각의 센서에서 제공하는 기준전원을 점검하거나 각각의 센서의 checksum등을 점검하여 어느 센서에서 고장이 발생했는지를 알아낼 수 있다. 그러나 센서 관리 시스템은 정상적인 데 자체 센서만 고장났을 경우 고장을 인지 할 수 없게 되므로 자체 진단 방식으로는 고장을 검출할 수가 없다 .

2) 다른 여분의 센서를 사용하여 비교: 예를 들면 고도계 고장시 GPS의 고도 신호를 사용하여 고장센서를 분리한다던지 IMU 요방향의 센서 고장시 자장계를 사용한다던지 하는 방식으로 고장센서를 분리할 수 있다. 하드웨어적인 다른 장치가 제공되어야 하기 때문에 제공된 장치역시 정상적으로 작동하고 오차정도가 고장 검출 대상 센서와 비슷해야 한다는 조건을 가지고 있다. 오차가 크다면 실제로 고장난 센서가 비교 센서와 더 가까운 값을 가지게 되어 정상적인 센서를 고장센서로 오인할 가능성이 다분하다.

3) 해석적 방법을 사용한 고장 검출: 센서의 확률적인 경향을 사용한 칼만 필터나 항공기 동역학을 고려한 관측기를 사용하여 제 3의 기준값을 제공할 수 있다. 고장 순간까지의 정보와 시스템 성향을 고려하여 센서의 값이 유효한 값인지 아닌지를 판단해 주는 방법으로 두 센서의 고장난 센서가 어느 것인지를 결정할 때 결정 인자로 작용한다. 1) 2) 방식과는 다르게 시스템 자체의 정보를 지능형 고장 진단 제어 방식이다.

가. 칼만 필터를 이용한 하드웨어 보완 방식

자이로, 가속도계를 포함해서 항공기에는 많은 센서가 쓰인다. 이 센서들은 항공기를 제

어하는데 있어 중요한 역할을 하고 있으며 센서들의 고장은 항공기의 안전에 큰 영향을 끼친다. 그런 이유로 고장에 대처하기 위해 모든 센서들은 다중화 되어 있으며 고장 시에는 고장 검출과 분리과정을 거치게 된다.

고장의 검출과 고장센서의 분리는 센서 자체의 잡음 특성들 때문에 상당한 주의를 요한다. 고장으로 인한 신호와 센서 잡음은 사실상 같은 신호적 특성을 띠기 때문에 잡음의 크기가 무시할 수 있을 정도로 작지 않을 경우 고장과 잡음은 구분하기가 힘들다. 따라서 고장을 판별하기 이전에 센서 자체의 잡음을 걸러서 영향을 줄여 주는 보정 과정이 선행되어야 한다. 센서 잡음의 보정을 위해 필터 시스템이 사용된다. 항공기의 제어 시스템에 사용되는 센서들의 노이즈 특성들을 연구하고 그에 맞는 필터 시스템을 구성하는 일은 항공기의 안전을 위해서 필수적이다. 본 연구실의 국가지정 연구인 통합비행제어 시스템 설계에서 항공기의 안전을 위한 센서 고장 검출 및 분리에는, 센서들의 노이즈들을 보정해주는 필터를 연구하는 일이 한 부분을 이루게 된다. 본 연구에서는 이미 연구된 전기 신호에 대한 필터들 중 칼만 필터를 고장 검출 분리를 위해 사용한다.

칼만 필터는 1960년대에 R. E. Kalman에 의해서 발표된 필터이론으로 백색잡음을 가지는 측정 노이즈에 대해 특히 뛰어난 성능을 보인다. 자이로 등의 센서가 가지는 오차원인은 백색잡음, 바이어스, 드리프트의 세 가지로 나눌 수 있다. 백색잡음은 평균값이 0인 잡음으로 일정한 크기의 편차를 가지는 난수이다. 백색잡음의 영향으로 센서에 대한 입력이 고정되어 있을 때 출력 값은 일정한 값에 고정된 채로 특정 크기의 표준편차를 가지는 잡음이 실제 값에 중첩되어 나타난다. 한편, 바이어스는 센서에 대한 입력이 없을 때에도 출력 값이 일정하게 편향되어 나타나는 특성이다. 바이어스 오차는 온도에 따라 변하게 되는데 그에 상응하는 전압을 가해줌으로써 보정할 수 있다. 드리프트는 센서를 장기간 사용할 경우 나타나는 오차로 시간이 지날수록 점진적으로 커지는 특성을 가진다. 드리프트는 실험을 통해 특성을 파악한 후 바이어스와 마찬가지로 보정을 한다. 여기서는 바이어스 오차와 드리프트 오차를 보정했다고 가정하고 백색잡음을 보정하는 방법을 설명한다.

나. 관측기를 이용한 해석적 고장 진단 방법

해석적 여분 방식으로 고장 진단을 수행하기 위해서는 시스템의 불확실성이나 외부 교란 등에 대하여 충분히 신뢰성있는 수학적 모델을 필요로 한다. 진단 관측기를 통한 고장 진단 방법이나 고유공간지정 법에 의한 고장 진단 법은 잔차 생성 방법에서 외부 교란 효과를 제외하기 위한 방법에서 접근을 하였지만 내부 파라미터에 대해 민감하다는 문제점을 안고 있다.

$$\dot{x}(t) = [A + \Delta A]x(t) + Bu(t) \quad (3.3.1)$$

$$y(t) = [C + C_f]x(t) + Du(t) \quad (3.3.2)$$

$x(t)$ 는 $n \times 1$ 상태변수를 의미하고, $u(t)$ 는 $m \times 1$ 의 입력 벡터, $y(t)$ 는 $p \times 1$ 의 측정벡터 ΔA 는 시스템 A시스템 파라미터 변화량, C_f 는 센서에서 발생한 고장이다.

고장진단기를 구성하기 위한 잔차생성 과정은 실제 측정벡터와 수학적 동역학 정보를 통하여 얻은 추정 출력 벡터 사이의 값을 비교하여 생성한다.

관측기 동역학은 다음과 같다.

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad (3.3.3)$$

$$y(t) = C\hat{x}(t) + Du(t) \quad (3.3.4)$$

$\hat{x}(t)$ 는 $n \times 1$ 상태 추정 벡터이고 $\hat{y}(t)$ 는 $p \times 1$ 출력 벡터이다. L 은 필터 게인을 의미하고 추정에러는 다음과 같다.

$$e(t) = x(t) - \hat{x}(t) \quad (3.3.5)$$

$$\begin{aligned} se(s) &= (A + \Delta A)x(s) - A\hat{x}(s) - L[y(s) - \hat{y}(s)] \\ &= A[x(s) - \hat{x}(s)] + \Delta Ax(s) - L[C_f x(s) + C_e(s)] \\ &= (A - LC)e(s) + \Delta Ax(s) - LC_f x(s) \end{aligned} \quad (3.3.6)$$

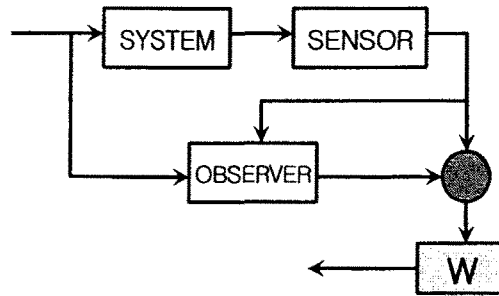


그림 3.3.28 관측기를 이용한
고장진단기

$$e(s) = [sI - (A - LC)]^{-1}[\Delta Ax(s) - LC_f x(s)] \quad (3.3.7)$$

출력 에러를 $e_y(t)$ 라 하고 이를 $y(t) - \hat{y}(t)$ 로 정의 하면

$$e_y(s) = C_f \hat{x}(s) + C[sI - (A - LC)]^{-1}[\Delta Ax(s) - LC_f x(s)] \quad (3.3.8)$$

이제 잔차를 다음과 같이 정의하자. W 는 잔차의 가중치 행렬이다. 확인하고자 하는 센서에 가중치를 두어 고장 진단을 수행할 수도 있고 시스템 파라미터의 변화량은 감지 하지 않도록 설정할 수도 있다.

$$r(t) = W e_y(s) = W[y(s) - \hat{y}(s)] \quad (3.3.9)$$

$$r(s) = W[I - C[sI - (A - LC)]^{-1}L]C_f x(s) + WC[sI - (A - LC)]^{-1}\Delta Ax(s) \quad (3.3.10)$$

시스템 파라미터에 대하여 잔차가 영향을 받지 않으려면 다음과 같이 ΔA 대한 식은 0이 되어야 한다.

$$WC[sI - (A - LC)]^{-1} \Delta A = 0 \quad (3.3.11)$$

위와 같은 조건을 만들어 가기 위하여 고유공간 지정법을 사용한다.

1) 다음 조건을 만족하는 잔차 가중 행렬 W 를 구한다.

$$WC\Delta A = 0 \quad (3.3.12)$$

2) 관측기의 고유공간을 결정한다. WC 의 행들은 관측기의 q left eigenvectors가 되어야만 한다.

3) 고유공간 조건에 만족하는 필터 게인 L 을 계산한다.

시뮬레이션

F-16항공기가 500ft/sec의 속도로 수평비행을 한다고 설정하였다. 이 때 observer에 사용된 시스템 매트릭스는 다음과 같다. 시스템의 중운동만을 감시 하였으며 고장 센서에 대한

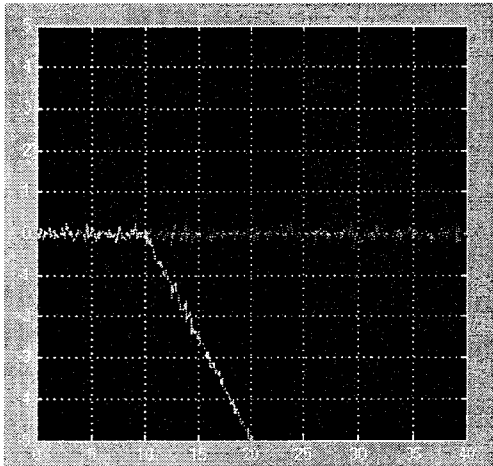


그림 3.3.29 고장 센서 신호

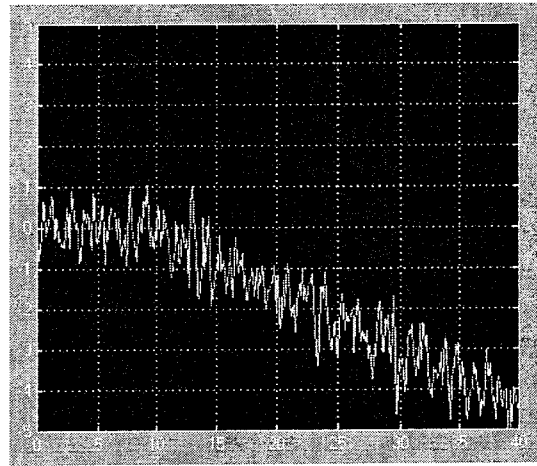


그림 3.3.30 잔차 생성 고장 감지

다. 뱅크 고장 필터 시스템

같은 상태값을 측정하는 센서가 3개 이상 될 때는 고장을 하드웨어 여분 방식으로만으로도 분리해낼 수 있지만 2개만 남게 되었을 경우 보조적인 선택 기준을 제공해주어야 한다. 따라서 위에서 언급한 해석적 고장진단 알고리즘을 선택 수단으로 사용하는 방법을 생각한다. 각각의 센서에 로컬필터로서 해석적 알고리즘을 통하여 실시간으로 잔차를 생성해 내게 된다. 하드웨어 여분방식으로 2개의 센서만 남았을 경우 센서값이 한계값을 넘었는지에 대한 감시를 마스터 필터에서 수행하고 고장이 감지되는 경우 로컬필터의 잔차가 한계값을 넘었는지 다시 확인함으로써 어느센서에 고장이 발생하였는지를 판단한다. 잔차가 크게 발생한 센서를 분리함으로써 정상적인 센서만을 사용하게 된다. 이러한 뱅크 고장 필터를 사용하여 2개의 센서만 남았을 경우에도 신뢰성을 보장할 수 있게 된다.

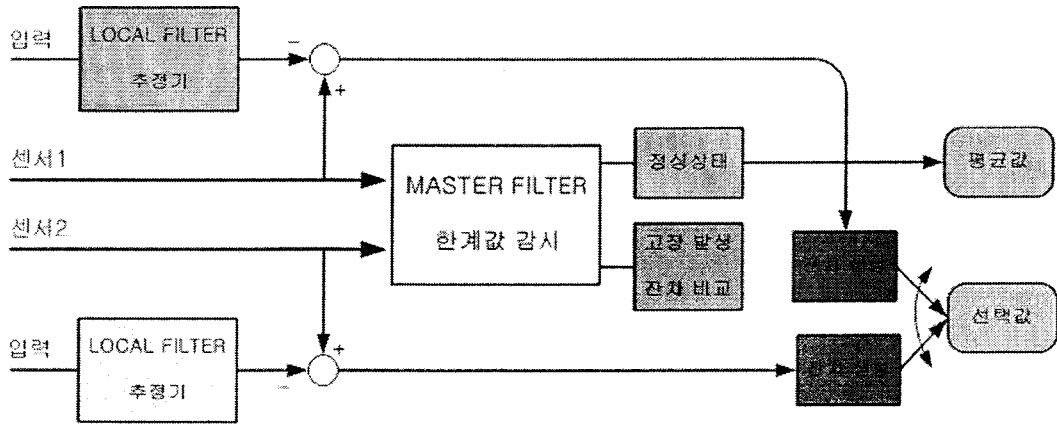


그림 3.3.31 하드웨어 여분과 해석적 여분방식의 통합

제 4 절. 지능형 통합 비행제어 시스템 설계 기법 연구

1. HILS 시스템 개요

가. 시뮬레이션의 정의 및 종류

시뮬레이션이란 실제 또는 가상의 동적 시스템 모형을 컴퓨터를 사용하여 연구하는 것을 말하며 모의실험 또는 모사라고도 한다. 물리적 모형을 써서 하는 실험으로서 예로부터 유명한 것은 풍동을 써서 항공기의 비행 중 상태를 조사하거나 물탱크 안에서 배의 항행 중 상태를 조사하는 등이 있다. 시뮬레이션은 제어기를 개발할 때에 센서 인터페이스나 구동 신호를 만들어 주지 않고 수식전개 만으로 빠르게 제어 이득을 획득할 수 있기 때문에 많이 이용되고 있다. 특히 항공기나 자동차와 같은 복잡한 시스템의 경우 시스템 운용자체가 많은 변수를 동시에 통제해야 하고 실험환경을 구성하는 것 자체가 어렵기 때문에 시뮬레이션을 통하여 제어기를 개발하는 것이 불가피하다.

시뮬레이션에서 사용하는 동적 모델은 주어진 시스템과 수학적 모델에 같은 입력을 주면 정확히 일치하는 출력을 내주어야 한다. 그러나 실제 모델에서는 시뮬레이션에서 미처 고려해주지 못해서 생기는 오차가 발생하기 마련이다. 항공기의 경우 바람의 영향이나 시스템 인터페이스에서 발생하는 잡음, 항공기 동적 모델의 분석오차, 센서의 입력 오차, 구동기의 구동시간 등의 각종 오차가 발생하기 때문에 시뮬레이션을 통해서 얻은 제어기를 그대로 실제 시스템에 탑재하여 사용하게 되면 미처 고려하지 못한 이러한 오차들로 인해 안정한 제어를 수행하지 못하게 된다.

항공기의 제어시스템 개발 단계에는 실제 시스템 환경에 점차 접근해 가는 방식으로 모사 환경을 구축하는데 실제 환경에 접근하는 정도에 따라 다음과 같이 분류된다.

SOFTWARE IN THE LOOP SIMULATION

일반적으로 수학적 가상 시스템을 구현하고 설계한 제어기를 수식으로 설치하여 시뮬레이션을 수행하는 수치적 시뮬레이션 시스템을 의미한다. 제어기의 기본방향설정과 제어기의 타당성검증을 수행하는 목적으로 사용된다. 수학적인 모델만을 사용하여 제어기를 구성하므로 단시간에 제어이득을 생성할 수 있으며 시스템의 주된 동특성을 가장 쉽게 구현할 수 있다. 환경에 대한 정보가 많이 축적되면 일반적인 환경의 잡음들도 수학적으로 모사할 수 있으며 Software in the loop 상에서 이러한 잡음들을 고려한 제어기를 구성할 수 있다.

FIXED BASED HARDWARE IN THE LOOP SIMULATION(PROCESSOR IN THE LOOP)

탑재형 제어시스템 일체를 검증하기 위한 시뮬레이션이다. 시스템과의 인터페이스 및 탑재형 제어시스템의 구성 오류 및 전력시스템의 기능, 버스시스템의 기능, 제어기의 실시간 제어 성능 등의 오류를 검증, 고장시의 대처사항을 검증할 수 있도록 구성한 모사 환경이다. 시스템의 운동을 구현하지 않지만 제어기를 실시간으로 구동 시킬 수 있도록 실시간 시뮬레이션을 구현하고 있으며 센서 및 구동기 등의 제반 인터페이스의 성능을 검증하기 위

하여 시스템 시뮬레이터와의 인터페이스는 실제 시스템에서와 같은 인터페이스를 통하여
를 통하여 신호를 주고 받는다.

MOVING BASED HARDWARE IN THE LOOP SIMULATION

지상에서 항공기의 모션까지 모사함으로써 센서 기능 검증과 이에 따른 제어기의 구동
능력을 검증하는 시스템을 말한다. 모션 테이블이라는 구동기를 사용하여 자세에 대한 모
션을 실시간으로 제공하고 탑재형 센서를 테이블 내에 설치하여 실시간 데이터 획득을 하
여 제어 성능을 검증할 수 있다.

나. HILS의 필요성

비행제어 시스템의 성능을 평가하기 위하여 단순히 수학적 모델을 대상으로 구현한 시뮬
레이터로는 성능검증에 한계가 있다. 하드웨어 인터페이스, 작동기 및 감지기 동특성, 데이
터 통신 버스의 시간지연 등 실제 시스템에서 발생하는 현실적인 문제들을 고려하기 때문
이다. 수학적 방정식만을 사용하여 재현된 모델상에서 제어시스템을 설계하였을 때는 실제
시스템에 존재하는 예측 불가능한 변수에 의하여 시스템이 불안정해 질 수 있기 때문이다.
100년에 달하는 항공기 개발의 역사가 이루어놓은 연구 덕분에 외부적인 불확정 변수(온도
변화, 기류변화)에 대한 경향은 예측할 수가 있으나 최근 프로세서를 이용한 항공기가 등장
함으로써 내적인 요인(고장이나 시스템의 복잡성으로 인한 버그, 전기적 충격에 의한 시스
템 작동정지)에 대하여 발생하는 시스템 불안정성은 제어기 개발에 있어서 새롭게 고려해
야 할 불확정 요인으로 등장하게 되었다. 이를 해결하기 위하여 직접 탑재시스템에 환경적
인 조건을 가하는 실험을 통하여 안정성 및 신뢰도, 실시간 제어성능 등을 검증해야한다.
항공기처럼 동적인 상태를 평형상태로 사용하는 불안정한 시스템의 경우 사고확률이 높고
시스템의 가격이 비싸므로 완벽하게 검증되지 않은 제어시스템을 탑재하여 공중에서 환경
테스트를 하는 것은 불가능하다. 따라서 지상에서 항공기 운동을 모사해주고 이 시스템의
신호를 받아 제어를 수행하는 가상 환경실험이 필요하게 된 것이다. HILS 시스템에서는 제
어대상인 비행체의 운동특성을 고성능을 컴퓨터에서 모사하여주고 실제 탑재 하드웨어를
극한 환경에 둬으로써 제어기 실시간 성능과 환경성능, 특수상황 설정 시뮬레이션 등을 수
행 할 수 있다. HILS 시스템을 이용함으로써 제어시스템이 실제 비행체에 탑재되기 전에
성능과 견고성을 실험실에서 충분히 검증할 수 있게 되고, 종래의 시행착오적인 제어시스
템 개발기술과 비교하여 시스템 개발비용과 기간, 위험성이 최소화된다.

국내에서 HILS 시스템의 도입은 비행제어시스템을 개발하는 작업과 궤를 같이한다. 초등
훈련기(KT-1)의 경우 Fly-By-Wire(FBW)시스템이 탑재되어 있지 않으므로, 고성능 비행
제어시스템과 관련되어 실제 축적된 연구개발 기술은 기초연구 수준의 미미한 정도이다.
그러나 이 사업을 수행하며 국방과학연구소에서는 비행제어 시스템의 성능을 지상에서 평
가할 수 있는 시뮬레이터 및 HILS장비를 구축 한바 있다. 중형항공기 사업과 관련해서 한
국항공우주연구원에서는 비행제어시스템 설계에 대한 기초연구를 수행한 바 있다. 그러나
아직까지는 국내에서 HILS를 활용하여 직접 비행제어 시스템을 구축하는 기술은 걸음마
단계에 머물러 있다고 볼 수 있다. 앞으로 다양한 항공기 시스템의 성능을 검증하기 위해서
연구용 시뮬레이터 및 HILS시스템을 활용할 수 있으리라 판단된다.

다. 구성

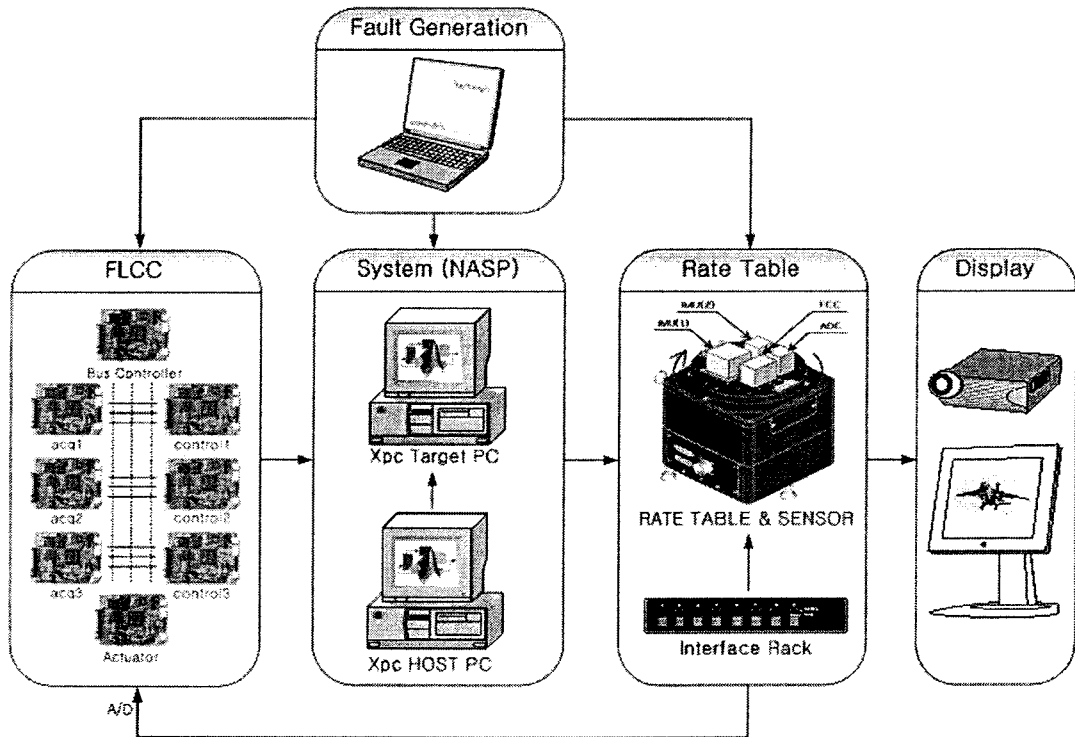


그림 3.4.1 FB-HILS 구성도

(1) NASP

앞서 Midlevel Control에서 설명한 비선형 항공기 시뮬레이션 시스템을 의미한다 HILS 시스템에서 NASP는 탑재 대상 시스템을 모사하여 줄 뿐만 아니라 시스템 환경의 고장도 모사해 줌으로서 실제로 발생시키기 어려운 고장에 대한 제어기 설계도 NASP를 통하여 가능하게 된다.

xPC Target 툴 박스는 Matlab 시뮬링크를 통하여 프로그램을 작성하고, 이를 C코드로 변환하여 다시 컴파일한 후, 시스템에 로딩하여 사용하는 일련의 임베디드 프로그래밍 과정을 PC를 대상으로 직접 수행하는 기능을 가지고 있다. 몇 년전만 하더라도 PC의 속도의 제약으로 몇 개의 CPU가 같이 병렬적으로 동작하는 워크스테이션을 통하여 시뮬레이터를 구동해야 했지만 현재 PC의 성능은 복잡한 공력데이터를 처리하더라도 빠른 샘플링타임으로 시뮬레이션을 수행할 수 있다.

일반적으로 하드웨어를 대상으로 하는 프로그램은 각각의 시스템에 따라 다르게 프로그래밍 해야 효율적이며, 최적화된 루틴을 수행한다. 하지만 PC에 대한 구조는 규격이 결정되어 있기 때문에, xPC target은 PC를 임베디드 타겟의 임무를 수행할 수 있도록 해주는 기본적인 프로시저와, 사용자가 개발한 제어로직을 변환하여 간단하게 임베디드 프로그래밍을 생성해준다. xPC target의 경우 보통 두 대의 컴퓨터를 사용한다. 하나는 프로그래밍과 컴파일, 모니터링을 전담하는 호스트 PC로 사용하며, 다른 하나는 PC를 하나의 임베디드 시스템처럼 실시간 제어기로 사용하게 된다. xPC target은 기본적으로 PC에서 제공하는 중앙처리장치 및 LAN, 병렬포트 등을 사용할 수 있다. 또한 아날로그/디지털 변환카드나 카운터, 모터 드라이버 등이 PCI슬롯을 통하여 PC에 연결되면, 그에 해당하는 드라이버

블록을 첨부하여, 어렵지 않게 I/O, 보조기억장치, 고속의 계산이 가능한 제어시스템 등을 완성할 수 있다. 기본적으로 호환하지 않는 특수한 하드웨어를 부가 시킬 때는 S-function 블록을 사용하여 드라이버를 만들어주거나, 직접 변환된 C코드에서 드라이버 코드를 첨가 시키면 응용된 제어시스템을 손쉽게 만들 수 있다.

본 연구에서는 Matlab에서 지원하는 xPC target을 사용하여 보급형 PC 상에서 실시간 NASP를 구축하였다. NASP는 MATLAB의 시뮬링크 블록으로 구성되고 기능별로 모듈화 되어 있기 때문에 비행체의 특성이 달라지면 해당블록을 교체하여 사용한다. 시뮬링크 블록 다이어그램을 매트랩 상에서 xPC target을 통하여 자동컴파일 하고 타겟 PC에 전송을 하면 외부 하드웨어가 인터페이스가 가능한 실시간 시뮬레이션을 구현할 수 있게된다. 시뮬레이션시 외부에서 받아야할 구동기 신호나 시뮬레이션을 통하여 얻은 시스템 운동 결과는 하드웨어의 RS232포트를 통하여 인터페이스 모듈로 송신 또는 수신된다.

본 연구에서 제안한 방법으로 시뮬레이터 구축할 경우 다른 방식으로 시뮬레이션 프로그램을 제작할 때 인터페이스 모듈을 만들며 변수설정 및 구축한 모델의 실시간 알고리즘 평가를 하는 일련의 복잡한 과정을 생략할 수 있으며 개발자에게 익숙한 매트랩환경에서 시뮬레이터를 구축할 수 있다는 점에서 많은 장점을 가지게 된다.

NASP의 구체적인 구동 방식과 모듈 설명, 본 연구에서 사용한 F-16 비행체의 특성 등은 3장에서 자세히 언급하고 있으므로 이 장에서는 설명을 생략하기로 한다.

(2) 고장 발생 구현

고장발생모듈은 별도의 시스템으로 구성되는 것이 아니라 NASP내부에 비행체 동특성이나 센서 전달 모듈을 변형시킴으로서 구현한다. 이는 여러 가지 실험과 문헌을 통하여 얻은 고장계통도를 통하여 각각의 모듈별로 고장이 발생시 발생하는 현상을 시뮬레이터에 첨가하는 방법으로 수행할 수 있다.

1) 센서고장모사

NASP를 구현한 시뮬링크 블록에는 시뮬레이션 결과를 외부로 전달하기 위하여 각 상태 변수를 수집하여 내보내는 REPRESENTATION 블록이 설치되어 있다. 센서의 고장 모사는 이 블록을 통하여 구현한다. 센서의 고장의 종류에는 앞서 고장계통도를 통하여 알 수 있듯이 여러 가지 양상으로 발현하는데 이런 양상을 NASP에서 출력되는 각 상태변수 15개에 각각 더하는 방식으로 수행한다.

2) 동체 고장 모사

동체에 고장이 발생한 경우에는 항공기의 공력특성이 변화하게 된다. 따라서 동체의 고장이 발생하게 될 경우를 모사하기 위해서는 공력성분 하나하나를 변형시킬 필요가 있는데 이렇게 될 경우 NASP상에서 여러 블록을 함께 변형시켜야 하는 단점이 있으므로 변형에 해당하는 항공기 동역학 박스 전체를 별도로 제작하여 고장 출현 시점에서 정상 동역학 박스에서 고장 동역학 박스로 전환하는 방식으로 구현한다.

3) 조동면 고장 모사

조동면 고장도 센서고장과 마찬가지로 제어입력을 받아들이는 포트에서 변환시킨다. 입력을 차단시켜 구동면을 고정시키거나 스케일 팩터를 곱하여 조종면 효율을 감소시키는 방법으로 조종면 고장을 모사할 수 있다. 항공기 조종면에 비대칭 고장이 발생하였을 경우에는 항공기 공력데이터를 바꿔야 할 필요가 있다. 이 경우에는 2)번과 마찬가지로 비대칭 고장 동역학 박스를 별도로 만들어 고장시 스위칭 시키는 방법을 사용한다.

4) 비행제어시스템 모듈의 고장 모사

비행제어시스템의 고장 발생상황은 직접 비행제어시스템에서 하나의 모듈을 정지시킴으로서 구현한다. 비행제어시스템 자체에서 발생한 고장은 시뮬레이터에서 모사할 수 없기 때문에 실제고장을 발생시켜 대처상황을 관찰할 수 밖에 없다. 각각의 센서신호 획득부, 비행제어부를 각각 하나씩 제거하거나 연속적으로 두 개를 제거하여 실험을 수행한다.

(3) 인터페이스 모듈

본 모듈은 실시간 NASP와 탑재형 비행제어시스템을 연결하기 위한 인터페이스 모듈이다. 탑재형 비행제어 시스템 자체를 그대로 검증해야 하므로 인터페이스 상자는 센서에서 생성하는 신호와 같은 형태로 입력신호를 제공하여야 하고 비행제어 시스템에서 시뮬레이터로 전달되는 제어신호도 구동서버 모터에 달려있는 각 센서로부터 아날로그 신호를 받는다.

1) 인터페이스 모듈의 구조

NASP는 RS232를 통하여 인터페이스 모듈로 센서신호를 전달하고 구동신호를 전달 받는다. 인터페이스 모듈은 4개의 채널로 구성되어 있는데 3개의 센서모사채널과 1개의 구동신호 입력채널이 있다. 3개의 센서모사채널은 3중화 된 센서 시스템을 모사하는데 정상 상태에서는 각각 노이즈만 다르게 설정하고 NASP에서 제공하는 같은 항공기운동결과를 D/A포트나 RS232를 통하여 출력하게 된다. 센서모사채널은 채널마다 각각 16개의 아날로그 포트와 4개의 디지털 포트가 있어서 탑재시스템에 설치된 센서 인터페이스의 양식대로 정보를 제공받을 수 있다. 구동신호 입력채널은 아날로그 타입으로 구동입력을 받아 NASP로 전달하는 역할을 수행한다.

인터페이스 모듈에는 DSP급 프로세서가 설치되어 있어 NASP 모듈으로부터 나오는 정보를 해석하게 되고 4채널 14비트 D/A 컨버터 12개와 RS232 12개, 14비트 A/D컨버터가 설치되어 있어 이 I/O 채널들을 통하여 비행제어 컴퓨터와 정보를 교환하게 된다. NASP와 인터페이스 모듈간에 주고받는 통신은 RS232 직렬 포트를 사용하며 'start,sensor#1,sensor#2,...,end' 형식의 캐릭터코드를 주고받는다. start는 헤더 역할을 하며 ',' 각 센서신호를 구분하는 역할을 하고 sensor#1, sensor#2는 각각의 센서값을 실수형(123.45 의 형태)캐릭터 형태로 표현한 것이다. 이 코드를 디코드하기 위해서는 ',' 별로 문자열을 구분하고 각각의 문자열 sensor#1, sensor#2를 문자열에서 실수형 변수로 바꾸는 함수를 통하여 수치의 형태로 변환시킨다. end는 한 사이클의 센서전달을 끝내겠다는 표시로 전달된다.

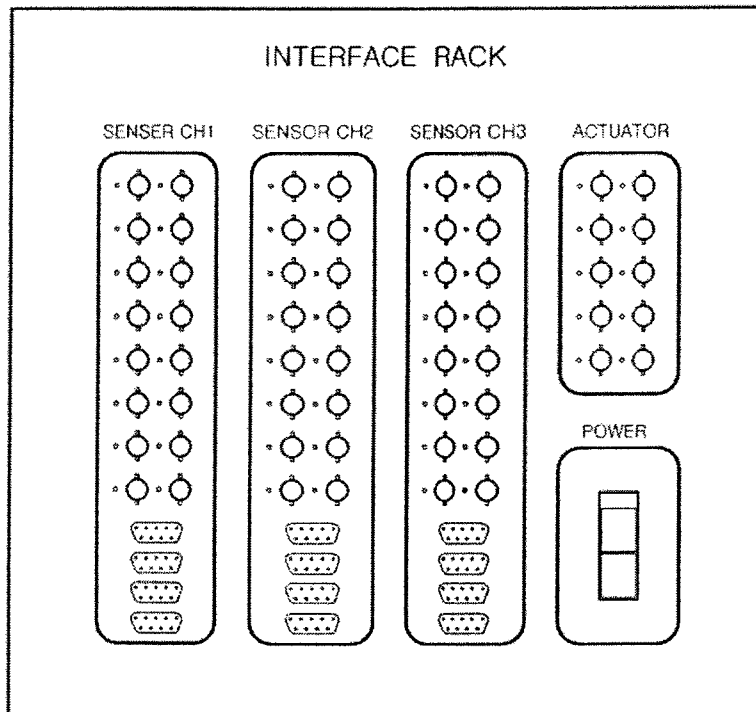


그림 3.4.2 센서 인터페이스 모듈

(4) 디스플레이 모듈

xPC target을 이용한 실시간 시뮬레이션을 수행할 경우 모든 상태변수가 그래프 형태로 도시된다. 그러나 그래프의 형태를 통해서만 항공기 상태 정보를 단번에 알 수가 없기 때문에 OPENGL을 통하여 항공기 모형을 구현하고 여기에 상태변수를 인가하여 실시간으로 비행체의 자세를 묘사하는 디스플레이 모듈을 추가하였다.

F-16의 비행체 형상 정보는 3차원의 점단위의 정보로 구성되어있다. 3개의 점이 폴리곤이라고 부르는 면을 만들고 면이 여러 개 모여 하나의 입방체를 만들게 되는 계층적 구조로 관리된다. 이러한 비행체의 모델링은 3D studio라는 모델링 프로그램을 통하여 구현되었고 이를 C++ 상에서 구현하기 위해 C type의 구조체로 변형하여 사용한다.

자세 표현은 비행체의 좌표계를 NASP에서 제공되는 ϕ, θ, ψ 각을 이용하여 회전변환 시켜 모델의 점의 좌표를 이동시켜서 표현한다. 각각의 점이 회전변환하게 되므로 비행체형태가 복잡하면 복잡할 수록 시간이 많이 소요된다. 위치의 표현은 지면을 의미하는 격자를 상대적으로 이동시킴으로서 표현할 수 있다. 역시 NASP를 통하여 얻은 북쪽 방향 좌표, 동쪽 방향 좌표, 고도 좌표를 사용하여 격자를 반대로 평행 이동시킨다.

위에서 묘사한 방법으로 얻은 좌표들을 모니터에 디스플레이하기 위하여 3차원 좌표를 화면 방향의 평면에 정사영시켜 2차원 좌표를 얻어낸다. 본 디스플레이 모듈은 원근 표현을 가미하여 정사영시 스크린에서 먼 화면은 좁게 가까운 점은 넓게 처리하여 정사영시켰다.

NASP에서 정보를 얻는 방법은 인터페이스 모듈이 NASP에서 정보를 얻는 방법과 같이 RS232시리얼 포트를 통하여 'start,sensor#1,sensor#2,...,end'형태의 아스키 패킷을 전달 받는 방식을 사용한다. 따라서 인터페이스 모듈에서 해석하는 방법과 동일하게 적용하여 위의 패킷을 해독할 수 있다. 디스플레이 모듈은 제어에 직접사용되지 않고 또한 사람의 눈

이 초당 24 프레임 이상을 인식하지 못하기 때문에 25Hz로 한정하여 화면에 비행체를 주사한다. 따라서 NASP로부터 받은 정보는 일일이 다 해석하지 않고 몇 번 씩 건너뛰어 가면서 해석하여 그래픽연산에 더 많은 시간을 할애하도록 한다.

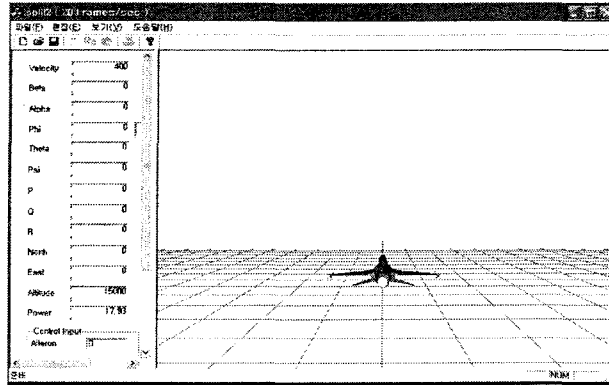


그림 3.4.3 Open GL Display 모듈

(5) MOTION SIMULATION

본 연구에서는 모션을 모사하여 센서에 입력을 주기위하여 RATE TABLE을 구매하여 HILS 시스템에 장착하였다. 모델은 IDEAL AEROSMITH사의 MODEL 1291 BR Single Axis Automatic Positioning And Rate Table이다. 본 테이블은 NASP와 연결되어 1축방향의 모션을 모사한다. 본 테이블의 성능은 다음과 같다.

표 3.4.1 레이트 테이블 사양

사양		성능
모터 토크(lbf-ft)		20
각한계		무제한
각속도	최대 각속도 (deg/sec)	±500
	최소 각속도 (deg/sec)	±0.000125
	해상도 (deg/sec)	0.000125
	정확도 (deg/sec)	0.01%±0.000125
가속도 (trapezoidal move)	최대 가속도(deg/sec ²)	4194
	최소 가속도(deg/sec ²)	0.064
	해상도(deg/sec ²)	0.064
가속도(sinusoidal)	최대가속도(deg/sec ²)	>8000
위치	정확도(deg)	±0.0056
	해상도(deg)	0.0000625

본 테이블을 구동하기 위하여 RS232형식의 통신을 이용하며 캐릭터 형태의 코드를 통하여 제어를 수행한다. 테이블 구동 명령은 다음과 같다.

- ACL 가속도 한계설정
- AMP SIN 명령의 진폭을 설정
- DIR 양의 회전방향을 설정

HOM 0도의 위치로 귀환
 JOG 일정한 각속도로 회전
 MOV 절대각으로 이동
 MXV 각속도 한계값 설정
 PER SIN 명령의 주기
 PPO 현재의 위치를 보고
 PVE 현재의 속도를 보고
 SAV 현재 세팅을 저장
 SIN 진동운동을 위한 파라미터 설정
 SGO 진동운동 시작
 STO 정지
 VEL 속도 설정
 ZER 0도 위치 설정

구동명령 바로 뒤에 숫자를 붙여 RS232를 통하여 테이블 컨트롤러로 전달하면 테이블은 명령한 대로 구동을 하게 된다. NASP에서 위의 명령을 사용하여 구성된 문자열(자세정보를 제공하여 MOVxx끝을 가짐)을 매 샘플링 타임마다 전송을 하면 한 축에 대한 시스템운동을 모사할 수 있다. 테이블에 질량이 존재하므로 위상지연이 발생하는데 지연시간을 측정하여 센서 반응시간에 추가하여 시스템을 제어하는 방식을 취한다.



그림 3.4.4 레이트 테이블

2. HILS 시스템을 이용한 실험 연구

가. 기본 임무수행 테스트

비행제어 시스템을 이용하여 기준 웨이 포인트를 찾아가는 임무를 수행한다. 본 통합실험에서는 8개의 8자형 웨이포인트를 거쳐서 원점으로 돌아오는 임무를 수행한다. 본 비행은 500ft/sec의 순항속도를 가지며 지표면 고도에서 비행한다. 그림 6.2.1은 다중화 제어기를 사용하여 8자의 경로점을 지나 원점으로 돌아오는 임무를 NASP의 비행체를 대상으로 수행하고 있는 모습을 보여주고 있다. 이 때 제어기는 50Hz로 제어를 수행하며 이착륙은 수행하지 않는다.

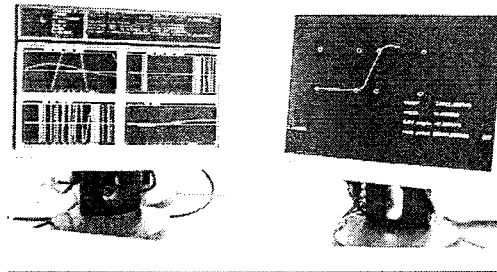


그림 3.4.5 기본 임무 수행 테스트

Way Point	x(ft)	y(ft)
1	0	0
2	8000	0
3	12000	8000
4	20000	8000
5	20000	0
6	12000	0
7	8000	8000
8	0	8000
9	0	0

본 연구정상적인 궤적 추정 비행 임무를 완수 하였다. 고장이 발생하지 않았으므로 고장 감지신호도 발생하지 않았으며 정상적인 자동비행을 수행하였음을 보여주고 있다.

나. HARDWARE 해법 RM 검증

Cross channel monitor method

가상 고장발생 모듈은 고장 검출 프로그램의 성능 검증을 위해 센서의 정상신호에 고장 신호를 추가하거나 정상신호의 파형을 변경하도록 구성되어 있다. 가상 고장발생 모듈은 표 6.2.1과 같이 설정된 고장 시나리오에 따라 고장을 일으킨다. 가상 고장발생 모듈은 고장지속시간, 고장발생시간, 고장의 크기, 고장발생채널의 매개변수를 갖고 임의의 고장을 발생시킨다.

표 3.4.2 고장 발생 시나리오

ID	Number of fault channels	Channel-1	Channel-2	Channel-3
1	1	1st	Normal	Normal
2		2	1st	Normal
3		3	Normal	1st
4	2	1st	2nd	Normal
5		2nd	1st	Normal
6		Normal	1st	2nd
7		Normal	2nd	1st
8		1st	Normal	2nd
9		2nd	Normal	1st
10	3	1st	2nd	3rd
11		1st	3rd	2nd
12		2nd	1st	3rd
13		2nd	3rd	1st
14		3rd	1st	2nd
15		3rd	2nd	1st

1. 한계치 설정 실험

자이로 센서의 한계치

그림 6.2.2에서 제시한 ENV-05F-03 자이로의 한계치를 설정하기 위해서 정지상태, 0.25 Hz 진동상태, 0.5Hz 진동상태의 세 가지 경우에 대해서 300Hz 샘플링 주파수로 측정하였다. 그림 5.4.2는 기준값 설정 프로그램의 결과로 출력된 자이로 신호분포도이다. 자이로의 경우, 10개의 구간으로 나누어 신호의 분포도가 그려지도록 프로그램 되어 있다. 내부함수는 신호분포도를 정규분포에 따라 근사적으로 계산하여 표현한 것으로, 데이터수가 많을수록 정규분포에 가까워진다.

그림 6.2.3은 자이로 센서의 확률 분포도를 나타낸 것이다. 그림에서 채워진 부분의 확률은 99.207%로 원하는 신뢰도 99%를 만족한다. 구해진 한계치값은 0.3V로, 이는 13.64 deg/sec에 해당한다. 보통 항공기용 자이로의 경우, 10deg/sec의 한계치를 갖고 있다. ENV-05F-03 자이로의 성능은 항공기용 자이로의 성능과 비교해 볼 때 약간 떨어진다.

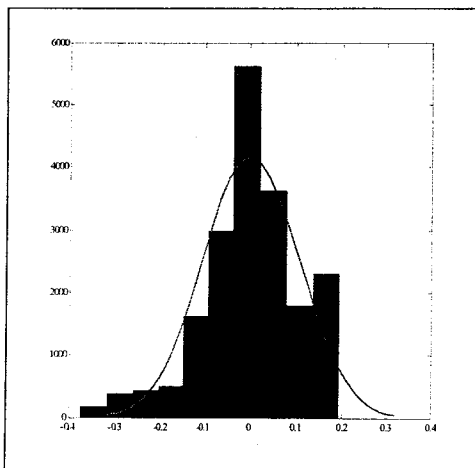


그림 3.4.6 자이로 센서 신호분포도

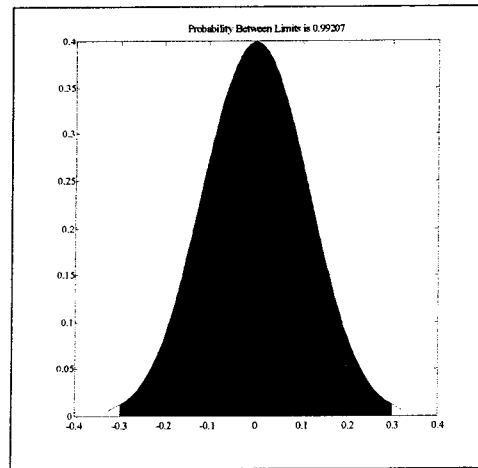


그림 3.4.7 자이로 센서 확률분포도

2. 가속도계 센서의 한계치

그림 6.2.5에서 제시한 ADXL105EM-1 가속도계의 한계치를 설정하기 위해서 정지상태, 0.3Hz 진동상태, 0.5Hz 진동상태, 1Hz 진동상태, 3Hz 진동상태의 다섯 가지 경우에 대해서 300Hz 샘플링 주파수로 측정하였다. 그림 6.2.5는 기준값 설정 프로그램의 출력으로 얻어진 가속도계의 신호분포도이다. 가속도계의 경우, 8개의 구간으로 나누어 신호의 분포도가 그려지도록 프로그램 되어 있다. 내부함수는 신호분포도를 정규분포에 따라 근사적으로 계산하여 표현한 것으로, 데이터수가 많을수록 정규분포에 가까워진다.

그림 6.2.5는 가속도계 센서의 확률분포도를 나타낸 것이다. 그림에서 채워진 부분의 확률은 99.32%로 원하는 신뢰도 99%를 만족한다. 이때 계산된 한계치는 0.196V로, 0.39g에 해당한다. 보통 항공기용 가속도계 센서의 경우, 0.3g의 한계치를 갖고 있다. 따라서 ADXL105EM-1가속도계 센서의 성능은 항공기용 가속도계 센서의 성능과 비교해 볼 때 약간의 차이를 나타낸다.

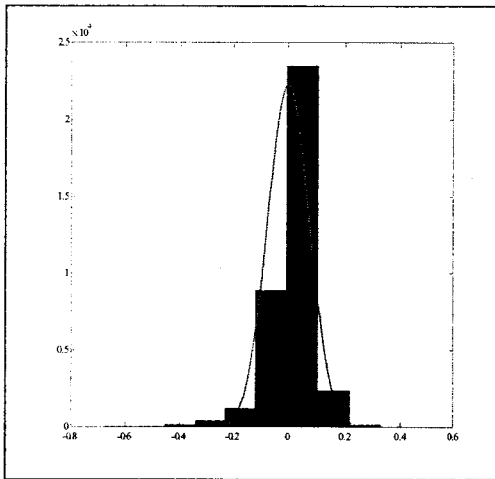


그림 3.4.8 가속도계 센서의 신호분포도

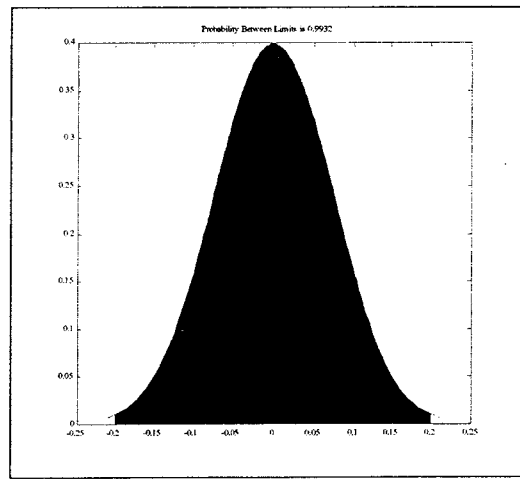


그림 3.4.9 가속도계 센서의 확률분포도

3. 하드웨어 해법의 RM 검증

자이로 센서의 유효시간 설정

자이로 센서의 표준편차로부터 얻어진 정규화된 표준편차를 이용하여 유효시간을 계산하게 된다. 그림 6.2.6은 자이로 센서에 대한 유효시간을 나타낸 것이다. 그림에서 보는 바와 같이 정규화된 표준편차가 작으면 유효시간은 짧아지고, 반대로 정규화된 표준편차가 크면 유효시간이 길어진다. 당해연도 연구에서는 정규화된 표준편차값 1.5σ에 해당하는 유효시간을 기준으로 고장진단을 수행하였다. 이때 자이로 센서의 유효시간은 0.35sec이다.

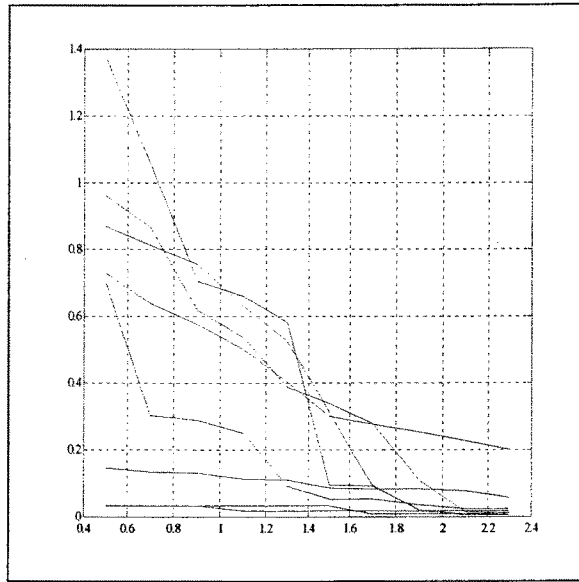


그림 3.4.10 자이로 센서의 유효시간

가속도계 센서의 유효시간 설정

가속도계 센서의 표준편차로부터 얻어진 정규화된 표준편차를 이용하여 유효시간을 계산하게 된다. 그림 6.2.7은 가속도계 센서에 대한 유효시간을 나타낸 것이다. 그림에서 보는 바와 같이 정규화된 표준편차가 작으면 유효시간은 짧아지고, 반대로 정규화된 표준편차가 크면 유효시간이 길어진다. 본 연구에서는 정규화된 표준편차값 1.5σ 에 해당하는 유효시간을 기준으로 고장진단을 수행하였다. 이때 가속도계 센서의 유효시간은 0.2sec이다.

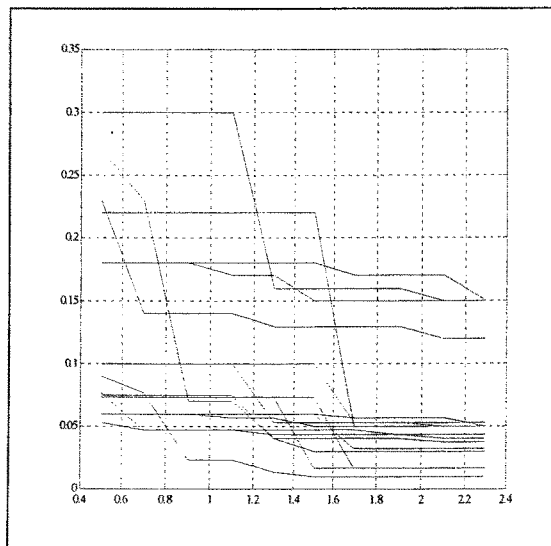


그림 3.4.11 가속도계 센서의 유효시간

4. 한계값 및 유효시간 설정 실험

가. 실험 목적

자이로 단위환산자(scale factor) 측정 및 선형성을 확인한다.

나. 실험 장비

자이로 : muRata (ENV 05A) No. 132562

Data Acquisition Card : PCI-6024

rate table

소프트웨어 : labview

파워서플라이

온도계

시계

37핀-5핀 케이블, 37핀 케이블

BNC 케이블

다. 실험 방법

- 1) 레이트 테이블에 자이로를 장착한다.
- 2) 컴퓨터의 전원의 켜고 labview를 실행한다.
- 3) 파워서플라이의 공급전원을 10볼트로 조정해 놓고 전원을 끈는다.
- 4) 실험온도를 측정한다.
- 5) 케이블을 연결하고 전원을 공급한다.
- 6) 1분 동안 자이로를 예열한다.
- 7) Labview를 실행하고 sampling rate 100Hz로 조정한다.
- 8) 1분 동안 자이로의 각속도를 바꾸어 가며 측정한다.
- 9) 측정하는 각속도는 $-100^\circ/\text{s} \sim 100^\circ/\text{s}$ 를 10° 도 간격으로 나누어서 측정한다
($-100^\circ/\text{s} \sim 100^\circ/\text{s}$ 는 전체 측정범위 $-90^\circ/\text{s} \sim 90^\circ/\text{s}$ 의 120%)
- 10) 실험은 1분 측정 후 전원을 차단하고 1분후에 다시 예열을 시작한다.
- 11) 이와 같은 실험을 5회 반복하여 실시한다.
- 12) 측정된 데이터로 그래프를 그려서 스케일 팩터를 계산한다.

$$\text{스케일 팩터} = (V_w - V_0)/W$$

V_0 : 0deg/s 일때 출력

w : 각속도

V_w : 각속도 w에서 출력

- 13) 선형성을 계산한다.

$$\text{선형성(\%)} = ((S_{V_w} - S_{V_{w_{\max}}}) \times w / w_{\max}) / S_{V_{w_{\max}}} \times 100$$

w : 각속도

w_{max} : 최대 각속도

S_{V_w} : w 각속도에서 스케일 팩터

S_{V_{w_{max}}} : 최대각속도에서 스케일 팩터

마. 결과

가) 측정데이터

1회

Input Rate (deg/s)	Output Voltage	
	mean (V)	Variance (×10 ⁻⁶ V ²)
-100	0.2669	7.9161
-90	0.4227	9.3479
-80	0.6482	8.7421
-70	0.8785	8.1738
-60	1.1083	8.9756
-50	1.3378	10.689
-40	1.5692	11.093
-30	1.7995	14.888
-20	2.0295	15.968
-10	2.2598	13.533
0	2.4920	7.8592
10	2.7224	13.964
20	2.9527	17.600
30	3.1826	15.806
40	3.4123	11.979
50	3.6424	10.169
60	3.8723	9.9989
70	4.1023	9.6822
80	4.3316	8.7872
90	4.5595	10.123
100	4.6883	7.5134

관계식 : $y = 0.0230x + 2.4907$

2회

Input Rate (deg/s)	Output Voltage	
	mean (V)	Variance (×10 ⁻⁶ V ²)
-100	0.2660	7.8660
-90	0.4202	9.6549
-80	0.6480	8.3213
-70	0.8771	9.4870
-60	1.1076	8.4712
-50	1.3378	10.119
-40	1.5681	11.104
-30	1.7985	14.396
-20	2.0286	15.904
-10	2.2592	12.984
0	2.4899	7.5646
10	2.7204	13.447
20	2.9515	17.688
30	3.1813	15.854
40	3.4120	11.902
50	3.6439	9.5400
60	3.8729	10.015
70	4.1032	8.9331
80	4.3313	8.6696
90	4.5588	9.9362
100	4.6885	7.6450

관계식 : $y = 0.0230x + 2.4900$

3회

Input Rate (deg/s)	Output Voltage	
	mean (V)	Variance ($\times 10^{-6} \text{ V}^2$)
-100	0.2667	8.0973
-90	0.4224	9.6208
-80	0.6496	9.1452
-70	0.8792	9.4808
-60	1.1089	9.0090
-50	1.3390	10.354
-40	1.5692	11.148
-30	1.7995	14.724
-20	2.0293	16.916
-10	2.2598	13.156
0	2.4868	7.7719
10	2.7205	14.145
20	2.9515	18.120
30	3.1810	15.277
40	3.4113	11.846
50	3.6415	9.8575
60	3.8716	9.9110
70	4.1005	9.9013
80	4.3293	9.0182
90	4.5567	9.6099
100	4.6888	7.6416

관계식 : $y = 0.0230x + 2.4897$

4회

Input Rate (deg/s)	Output Voltage	
	mean (V)	Variance ($\times 10^{-6} \text{ V}^2$)
-100	0.2657	7.4783
-90	0.4194	9.4829
-80	0.6474	8.1325
-70	0.8771	9.3945
-60	1.1071	7.9185
-50	1.3372	10.038
-40	1.5676	11.159
-30	1.7981	15.201
-20	2.0283	17.054
-10	2.2588	12.567
0	2.4893	7.1898
10	2.7203	14.707
20	2.9516	17.648
30	3.1812	15.646
40	3.4119	12.190
50	3.6420	10.320
60	3.8726	10.152
70	4.1024	9.1805
80	4.3317	8.7994
90	4.5596	9.9170
100	4.6884	7.7800

관계식 : $y = 0.0230x + 2.4899$

5회

Input Rate (deg/s)	Output Voltage	
	mean (V)	Variance ($\times 10^{-6}$ V ²)
-100	0.2658	7.7713
-90	0.4199	9.2084
-80	0.6476	8.6377
-70	0.8768	9.2860
-60	1.1070	7.8019
-50	1.3368	10.151
-40	1.5669	10.805
-30	1.7972	14.172
-20	2.0272	15.101
-10	2.2578	12.475
0	2.4883	7.2044
10	2.7187	13.521
20	2.9499	16.874
30	3.1793	16.693
40	3.4100	12.023
50	3.6397	10.604
60	3.8701	10.080
70	4.0995	9.6999
80	4.3301	8.8341
90	4.5566	9.5951
100	4.6885	7.7530

관계식 : $y = 0.0230x + 2.4884$

나) 그래프

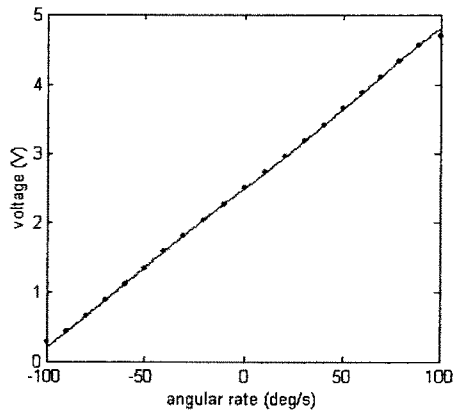


그림 3.4.12 1회 온도 : $24.8^{\circ}C$

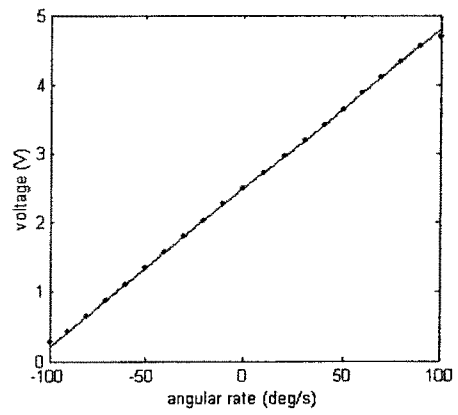


그림 3.4.13 2회 온도 : $25.1^{\circ}C$

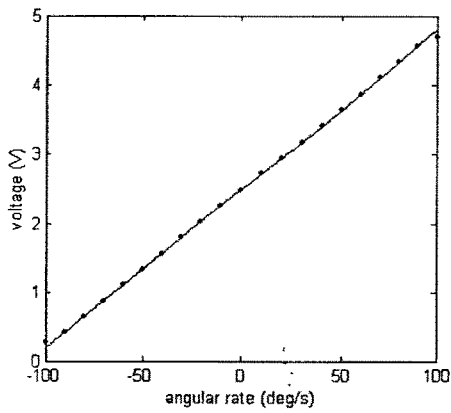


그림 3.4.14 3회 온도 : $25.1^{\circ}C$

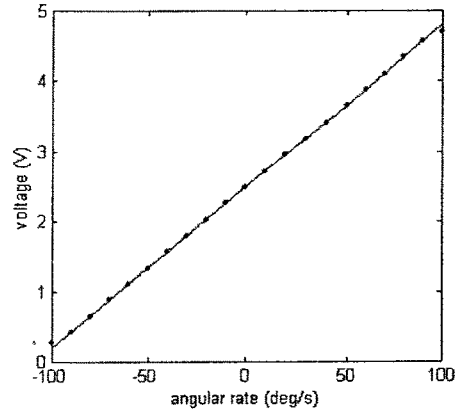


그림 3.4.15 4회 온도 : $26.8^{\circ}C$

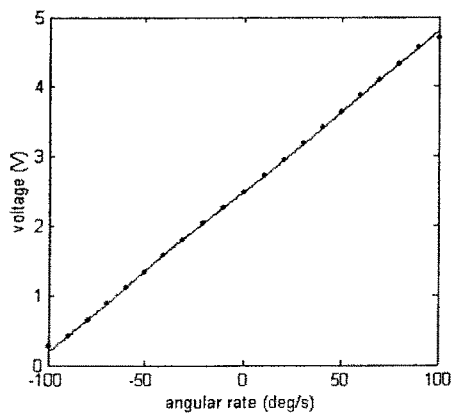


그림 3.4.16 5회 온도 : $26.6^{\circ}C$

다) 스케일 팩터

회수	환산 계수	오프셋 전압
1	0.0230	2.4907
2	0.0230	2.4900
3	0.0230	2.4897
4	0.0230	2.4899
5	0.0230	2.4884
평균	0.0230	2.4897

라) 선형성

회수	선형성 (%)	비 고
1	-0.06550	
2	0.00590	
3	0.01852	
4	0.15674	
5	0.00456	
평균	0.02400	

바. 결론

- 1) 스케일 팩터 : 0.0230 (V/deg/s)
- 2) 선형성 : 0.0240 %

5번의 실험과정에서 스케일 팩터는 모두 0.0230으로 측정되었다.
 정적 전압은 2.4894로 5회의 평균값으로 결정하였다.

$$y = 0.0230x + 2.4897$$

y : voltage (V)

x : angular rate (deg/s)

위의 관계식을 통하여 각속도와 자이로 출력의 관계를 정의할 수 있다.

사. 실험과정

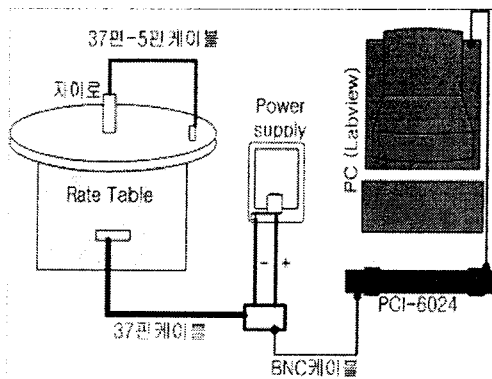


그림 3.4.17 실험 배선도

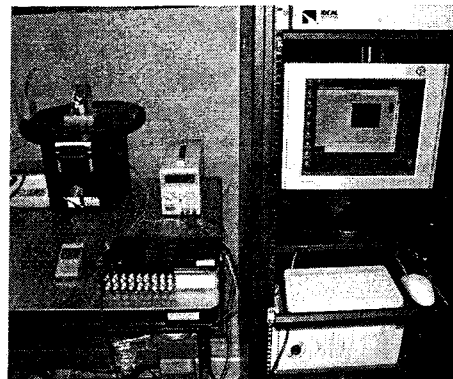


그림 3.4.18 실험 장치

5. 고장 검출 실험

3중으로 구성된 자이로 센서에 대해서 고장 모니터 프로그램과 자이로 센서의 기준값에 대한 신뢰도를 검증하였다. 고장진단 시뮬레이션의 매개변수로는 고장지속시간, 고장발생 시간과 고장의 크기 등이다. 기준값에 대한 신뢰도를 검증하기 위하여 한계치와 유효시간에 근접하는 가상 고장을 발생시켜 시뮬레이션함으로써 기준값에 대한 신뢰도를 검증할 수 있다. 또한 고장모니터 프로그램의 경우는 다양한 채널에 고장이 발생한 상황을 모사함으로써 고장 모니터 프로그램의 신뢰도를 검증하였다.

표 3.4.3는 고장 모니터 실험의 결과를 나타낸 것이다. 이로부터 각각의 고장이 발생한 경우에 대해서 고장을 효과적으로 분리해 내는 것을 확인하였다.

표 3.4.3 자이로에 대한 고장 검출 결과

ID	Number of fault channels	Channel-1	Channel-2	Channel-3	Alarm
1	1	1st	Normal	Normal	OK
2		2	1st	Normal	OK
3		3	Normal	1st	OK
4	2	1st	2nd	Normal	OK
5		2nd	1st	Normal	OK
6		Normal	1st	2nd	OK
7		Normal	2nd	1st	OK
8		1st	Normal	2nd	OK
9		2nd	Normal	1st	OK
10	3	1st	2nd	3rd	OK
11		1st	3rd	2nd	OK
12		2nd	1st	3rd	OK
13		2nd	3rd	1st	OK
14		3rd	1st	2nd	OK
15		3rd	2nd	1st	OK

표 3.4.4은 가상 고장 발생을 10,000회 수행하여 발생한 고장 발생 선언의 결과를 나타낸 것이다. 고장이 발생한 경우에 고장을 선언하지 못하는 “고장 선언 누락”(missed alarm)이 10,000회 중 49회 발생하였다. 고장 선언이 누락된 경우 중 고장지속시간이 유효시간과 동일한 경우는 고장으로 선언하도록 설정함으로써 고장 선언 누락 문제를 해결할 수 있다. 그러나 2개 센서에서 동시에 고장이 발생한 경우에는 직렬 감시기법을 이용하여 검증해야 한다. 고장이 아닌 경우 고장으로 선언하는 “고장 선언 오류”(false alarm)는 한 번도 발생하지 않았다.

표 3.4.4 고장 선언 결과

Count	Alarm	Missed Alarm	False Alarm
10,000	9,951	49	0

다. 칼만필터를 사용한 해석적 여분 방식

정상운용모드에서 비행선은 그림 4.10과 같이 임무영역을 중심으로 등속 선회비행을 하게 된다. 이와 같은 비행조건에서는 고도를 유지하며 일정한 요각운동을 수행하게 된다. 이러한 운동에 대한 모의실험을 위해서 그림 4.11과 같은 실험장치를 구성하였다. 1축 레이저 테이블에 관성항법장치와 비행제어컴퓨터를 2중으로 장착하였고, 다른 항법장치와 구동기의 출력신호는 DSP 컴퓨터 모듈을 이용하여 가상으로 출력하였다.

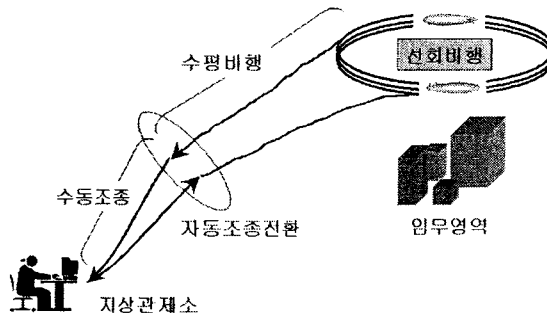


그림 3.4.19 운용 개념도

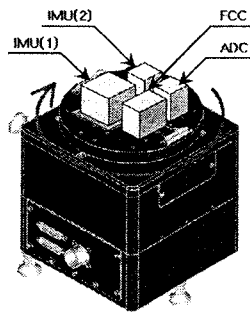


그림 3.4.20 고장진단 실험장치

센서로는 각속도 센서로 MURATA ENV05A를 2개 사용하였다.

각속도 센서는 센서 보정 테스트에 의해서 바이어스와 스케일 팩터를 보정하였으며 센서의 분산은 $1.1e-7$ 로 설정하였다. 따라서 한계값은 2.575σ 인 $8.5403e-4$ 로 정한다. 각각의 센서는 AD 컨버터를 통하여 FCC에 전송된다. AD 컨버터에서는 고장센서에 0.3V를 추가하는 장치를 설치하여 고장을 인위적으로 생성할 수 있게 한다.

비행선의 운행궤적은 임무 영공을 840m반경의 원궤도를 20m/s(5.556m/sec) 로 선회비

행한다고 가정한다. 위의 궤적을 따라 비행선이 선회하기 위한 요각속도는 0.05 rad/s이다. 따라서 실제 자이로에 0.05 rad/s값을 넣어주기 위하여 레이트 테이블을 다음과 같은 운동 방정식을 이용하여 회전시킨다. 비행선에서 리더와 비행선의 관계동역학을 다음과 같이 간단하게 가정한다.

$$A = \begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\dot{x} = Ax + Bu$$

이 때 $x = [x_1 \ x_2]$

x_1 : 각속도

x_2 : 방향각

고장 모듈을 사용하여 실험 시작 15초 후에 가상의 고장신호를 넣어준다. FCC는 2개의 센서정보로부터 각각 칼만 필터를 구동한다. 각각의 신호를 모니터링하여 2개의 신호가 평균값에서 벗어나 한계시간 이상 지속되었을 때 경보를 발생하고, 칼만 필터 잔차를 이용하여 센서신호 정상여부를 결정한다. 센서의 한계치는 2.575σ 인 $8.1429e-4$ 이 된다.

한편, 2개의 센서가 평균값에서 한계치 이상으로 벗어나게 되면 2개의 신호가 동시에 벗어난 것이 되므로 어느 센서에서 파손이 일어났는지 알 수 없게 된다. 이 때 칼만 필터를 이용한 잔차생성 방법을 통하여 어느 센서의 잔차가 0이 아닌지를 판별한다.

그림 3.4.21부터 그림 3.4.26까지는 시뮬레이션 결과를 보여주고 있다. 그림 3.4.23에서 보듯이 평균값이 15초후 크게 변화하고 있어 고장이 진단되고 있다. 그림 3.4.23에 각각의 센서의 잔차 비교결과를 나타내고 있다. 센서 1 채널의 잔차가 한계값 $8.1429e-4$ 를 벗어나 있으므로 센서 1에서 파손이 일어나고 있음을 인지 할 수 있다.

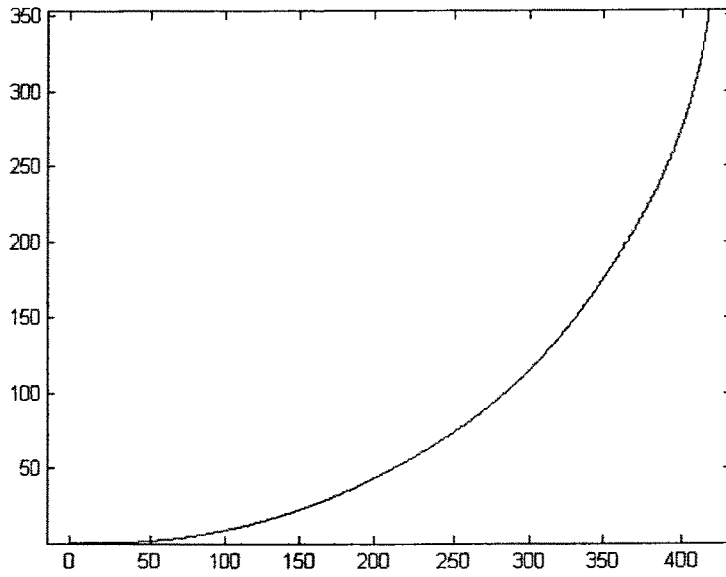


그림 3.4.21 비행선 진행 궤적

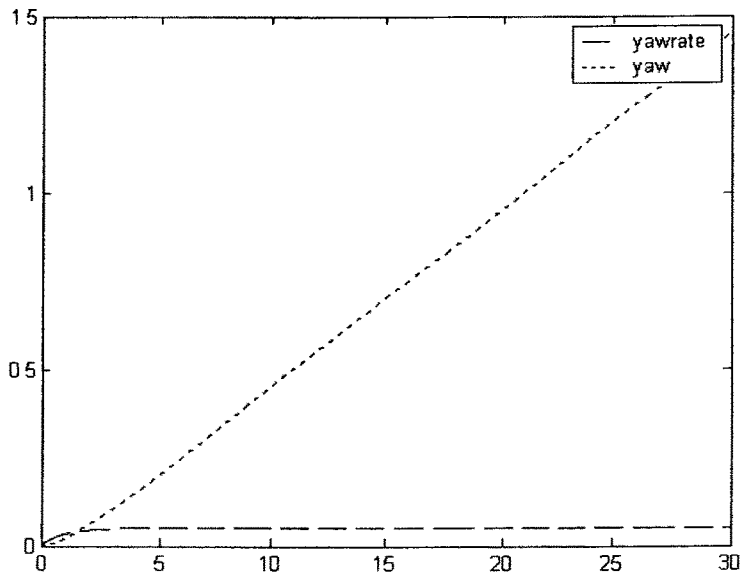


그림 3.4.22 상태변수

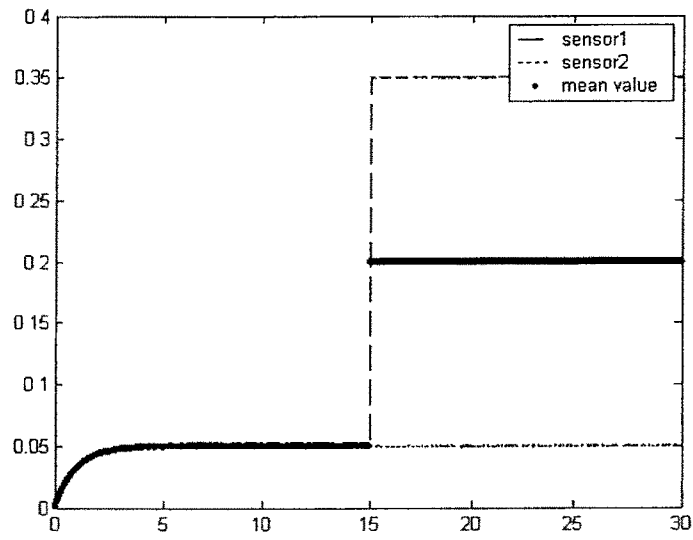


그림 3.4.23 자이로 센서의 신호와 기준값

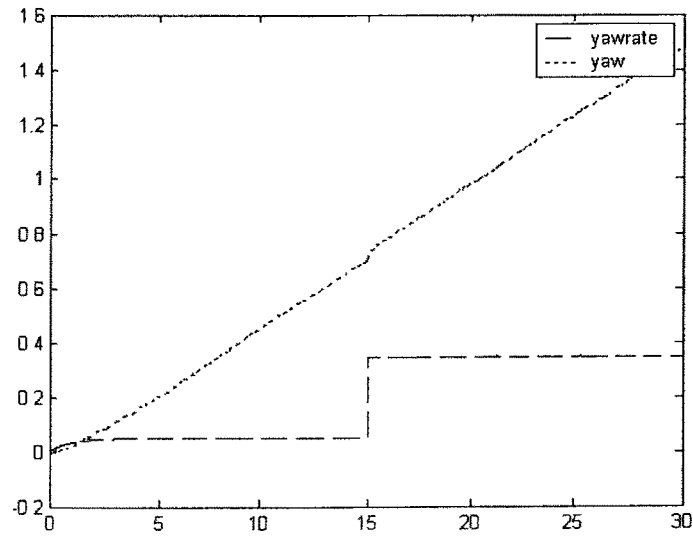


그림 3.4.24 고장센서 채널의 상태변수 추정치

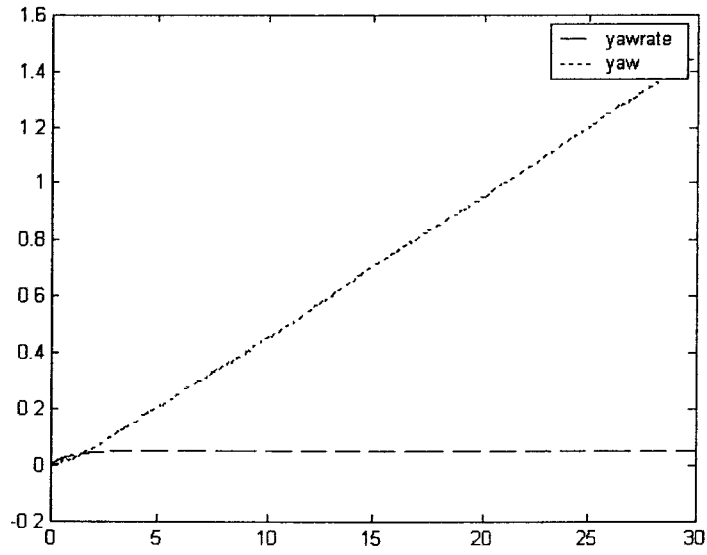


그림 3.4.25 정상센서 채널의 상태변수 추정치

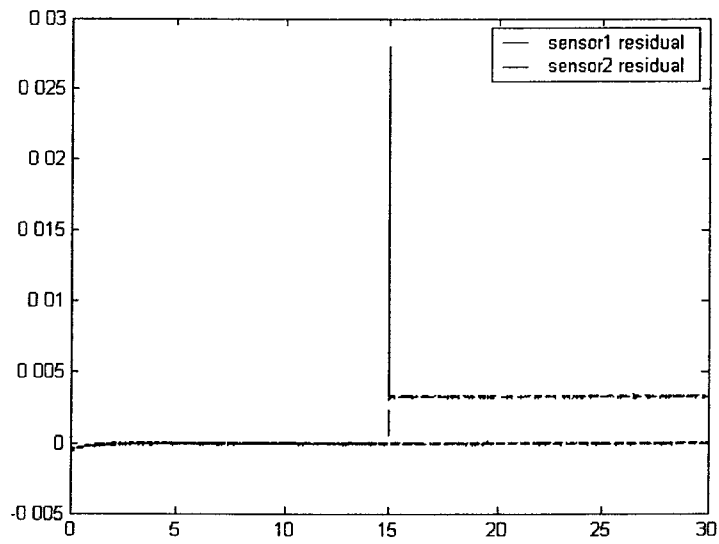


그림 3.4.26 각속도 센서1과 각속도 센서2 의 잔차 생성치

제 4 장 목표달성도 및 관련분야에의 기여도

제 1 절 계획대비 달성도

번호	세부연구개발목표	달성내용	달성도 (%)
1	Mid-level 비행제어 설계기법 연구	<ul style="list-style-type: none"> - 정상모드에서 고장모드로의 전환 중에 제안된 비행기법의 유용성을 판별하기 위한 검증 시스템 구축 및 검증 - 적응제어기법 및 신경회로망을 이용한 비선형 제어시스템 설계기법 연구 - 모드전환 및 실시간 계산 등 실시간 운용을 고려하여 시스템 인식 기반의 온라인 제어기법, 계산량이 적은 직접법 등의 연구수행 	100
2	Fixed Base HILS 시스템 구축	<ul style="list-style-type: none"> - 3중화 기법을 이용하여 비행제어 컴퓨터 및 데이터버스, 센서 suit 단위 구축 - PC기반의 비행제어 컴퓨터를 xPc target을 이용하여 보다 사용자에게 편리한 비행 제어시스템 모듈로 구현 - 구성된 FB HILS 시스템의 타당성을 검증하기 위한 단위 실험 수행 - 비행제어시스템 제어신호를 위한 RS232 통신기법의 프로토콜 규약 설정 및 FB HILS 시스템에의 적용 	100
3	RM 시스템의 하드웨어 해법 구현	<ul style="list-style-type: none"> - 교차채널과 직렬채널 기법을 이용한 3중 다중화관리 하드웨어 해법 구현 - 다중 센서에 대한 최대 고장검출 성능을 나타낼 수 있는 센서의 최적배치에 대한 연구 수행 - 제어 모듈에 독립적인 다중화관리 시스템 해법 성능 검증 	100
4	RM 시스템의 해석적 알고리즘 개발	<ul style="list-style-type: none"> - 기존의 다양한 해석적 고장검출기법에 대한 연구를 바탕으로 정상모드와 고장모드를 분리할 수 있는 고장검출 알고리즘 검토 - 효율적인 해석적 알고리즘 구현 위한 필터 설계 및 수치 시뮬레이션 수행 - 고장 발생시 시스템의 입출력 신호를 이용하여 고장에 대처하는 비행제어시스템에 대한 연구 개발 수행 	90
5	6-DOF NASP를 이용한 제어알고리즘의 성능검증	<ul style="list-style-type: none"> - 하드웨어와의 인터페이스를 고려한 6자유도 비선형 NASP 구축 - 단위 테스트를 통한 NASP 타당성 검토 및 각 모듈별 성능 검증작업 수행완료 - Mid-level 비행제어기법 성능향상을 위한 다양한 기법의 유도 알고리즘 제안 	100
6	RM 시스템 해석적 알고리즘과 하드웨어 해법의 통합설계	<ul style="list-style-type: none"> - BIT 기법을 이용한 하드웨어 보완모델 구현 - 하드웨어 보완을 위한 칼만필터를 이용한 해석적 알고리즘의 적용 - 신뢰도 확보를 위한 기능별 모듈화와 자체 테스트 수행 - CAN을 사용한 각 모듈간의 통신 및 건강성 확인 	90
7	Fixed-base HILS를 통한 RM 시스템 성능검증	<ul style="list-style-type: none"> - xPc target을 이용한 실시간 RM 시스템의 성능검증 수행 - FB-HILS를 이용한 RM 하드웨어 해법의 반복적인 성능검증 수행 - FB-HILS를 이용하여 RM 해석적 알고리즘의 성능 평가 - FB-HILS를 통한 탑재형 비행제어시스템의 전체 성능 평가 수행 	100

본 보고서에서는 비행역학 및 제어 국가지정연구실에서 1단계 연구기간 동안 수행한 통합 비행제어시스템 설계에 관한 연구내용을 정리하였다. 국가지정연구실 제안서에 제시한 연구 내용은 모두 계획대로 수행되었다. 1단계 연구기간의 연구 결과를 바탕으로 2단계 연구의 기반을 확보하였고, 통합 비행제어시스템의 설계 및 검증을 위한 기술과 이론을 축적하였다. 하드웨어 부문에서는 FB-HILS 시스템의 구축이 완료되어, 2단계 계획상의 MB-HILS 구축을 준비 중이다. 이론 부문에서는 신경회로망, 퍼지논리 등의 지능형 제어기법을 응용한 비행제어 알고리즘 및 고장발생 시 RCS에 관한 연구가 학술회의 및 저널에 게재되었다. 또한, 다중화 관리에 관한 소프트웨어가 개발되어 실제 실험을 수행하였고, NASP를 개발하여 설계된 비행제어 알고리즘들을 검증하였다. 2단계에서는 MB-HILS를

구축할 것이며, 1단계에서 개발한 비행제어 알고리즘을 개선하여 MB-HILS를 통해서 검증할 것이다. MB-HILS의 결과를 토대로 실제 항공기에 비행제어시스템을 적용하여 실제 시스템에서의 활용 가능성을 시험할 수 있을 것으로 사료된다. 1·2단계의 연구결과는 워크샵 및 세미나, 강연 등을 통해서 관련 산·학·연에 제공될 것이며, 홈페이지를 통해서 관련분야 정보를 얻고자하는 연구원들과 일반인에게 보급될 것이다.

제 2 절 1단계 연구 결과

1단계 연구에서는 mid-level 비행제어 설계기법, 고장허용 비행제어 설계기법, 하드웨어 다중화 관리 시스템, 지능형 통합 비행제어 시스템 설계기법에 관한 연구를 수행하였다. 계층적 비행제어 시스템 개념에서 살펴 볼 때, mid-level 비행제어 설계기법은 기존의 각 비행영역에 대해서 설계되던 제어 시스템을 통합한 개념으로, 최근에는 실제 항공기 설계에 응용되고 있다. 넓은 비행영역에 적용 가능한 비행제어 알고리즘의 개발을 위해서 본 연구에서는 지능제어 제어기법의 하나인 신경회로망, 퍼지로지, 유전자 알고리즘을 이용한 제어기 설계 방법을 연구하였으며, 모드 전환 비행제어 알고리즘에 대한 연구도 수행하였다. 항공기의 생존성을 높이기 위한 고장허용 제어시스템 설계기법 연구도 소프트웨어 기법과 하드웨어 기법으로 나누어 광범위하게 진행되었으며, 해석적 알고리즘을 구현하기 위한 필터 설계 기법도 연구되었다. 대형여객기나 고성능 군용 항공기에는 생존성 확보를 위한 필수적인 시스템으로 자리매김하고 있는 재형상 비행제어 시스템에 대한 연구도 수행되었다. 이러한 재형상 비행제어 알고리즘으로 고장인식 과정이 필요 없는 직접 적응제어기법, 고장인식 과정을 포함한 슬라이딩 모드제어기법, 입출력 정보를 이용한 시스템 인식기법을 사용한 기법 등을 개발하였다. 이러한 알고리즘을 검증하기 위하여 F-16 항공기의 운동 방정식에 적용된 6자유도 비선형 시뮬레이션 프로그램(NASP)을 개발을 완성하였다. NASP는 모듈화 개념을 적용하여 개발했으며 모듈간 인터페이스를 강화하고 안정성을 높였다. NASP의 성능검증 결과는 비행제어 알고리즘 개선 자료로 쓰일 수 있으며, HILS에서는 실제 항공기 시스템을 모사하는 중요한 역할을 수행하였다.

다중화 시스템은 다중화 된 하드웨어에 대하여 고장을 감시하고, 고장이 발생한 하드웨어를 시스템으로부터 분리시키는 역할을 수행한다. 그러므로 통합 비행제어시스템에서는 하드웨어 다중화가 필수적이라 할 수 있다. 하드웨어가 다중화가 되었을 경우, 이를 통합 관리하는 소프트웨어의 개발이 필요하며, 개발된 다중화관리 소프트웨어는 실제 하드웨어와 연동하여 적합성을 확인하여야 한다. 본 연구에서는 하드웨어 다중화 관리기법에 관한 이론 연구와 적용 가능성에 대한 검증을 수행하였으며, 3중으로 배치된 센서에 대해서 voting 알고리즘을 구성하여 다중화 시스템의 하드웨어 해법을 구현하는 연구를 수행하였다.

제안된 알고리즘을 하드웨어적으로 검증하기 위해서 MB(Moving Base)-HILS 시스템을 구현하기 위한 전단계로 FB(Fixed Based)-HILS 시스템을 개발하기 위한 연구를 수행하였다. DSP(Digital Signal Processor) 보드를 이용하여 1축 FB-HILS 시스템 실험 장치를 구성하기 위한 비행제어 컴퓨터를 설계, 제작하였다. 또한, 향후 개발하고자 하는 최종적인 HILS 시스템의 구조를 설계하였다. 모듈화 개념을 통해서 비행제어 알고리즘 및 다중화 시스템의 개념을 어느 정도 구현하였으며, FB-HILS 시스템의 각 부 시스템 또한 신뢰

성 높게 설계되었다. 부 시스템 간에는 효율적인 interface 모듈을 가지도록 하였으며, 통신을 위한 데이터 버스(Bus) 시스템에 대한 연구도 진행되었다. 실제 군용 항공기에서 사용되는 데이터버스에 대한 분석 연구를 수행하였으며, 데이터버스 역시 다중화하기 위한 연구가 진행되었다. HILS을 통한 실험결과는 분석과정을 통해 다중화 관리와 제어 알고리즘에 관한 개선점을 제시하게 된다.

이러한 연구를 바탕으로 다중화 시스템의 해석적 알고리즘과 하드웨어 해법을 통합 설계하였다. 다중화 시스템의 해석적 알고리즘은 수학적 모델을 이용해서 다중화 관리를 수행하는 시스템을 의미한다. 해석적 다중화 시스템은 적절한 시스템 모델이 필요하며, 하드웨어적 다중화 시스템과 연동하여 구현하면 더 뛰어난 성능을 얻을 수 있다. 따라서 다중화 시스템 연구는 효율적 해석적 다중화 알고리즘 개발 및 하드웨어적 다중화 해법과의 통합 설계에 중점을 두어 연구가 수행되었다. 이러한 하드웨어와 소프트웨어와의 통합 설계로 시스템의 고장 진단 및 고장대처 능력에 뛰어난 성능향상이 기대된다.

제 5 장 연구개발결과의 활용계획

항공우주선진국에서는 개발된 비행제어시스템 설계기술을 지능형 자동차의 항법 및 제어 시스템 개발에 응용하고 있으며, 차세대 쾌속선박의 항법 및 운항제어시스템 개발에도 사용하고 있다. 본 연구의 수행결과인 통합 비행제어시스템 설계기술은 다양한 환경에 대해 지능적으로 대처하는 전체 시스템의 해결능력 배양이라는 측면에서 전 산업분야에 영향을 미칠 수 있다. 항공우주분야에서는 새로운 개념의 대형여객기, 고성능 전투기 등에 대한 설계가 매우 빠른 속도로 진행되어 개념설계 단계를 넘어서고 있다. 특히, 미국 NASA와 보잉에서는 날개를 회전하여 수직으로 이륙한 후에 회전하는 날개를 고정시켜 일반 항공기와 같이 비행하는 무인 시험비행기 X-50을 2003년 중반에 시험비행할 것을 계획하고 있다. 이 항공기에 탑재될 비행제어시스템은 이제까지의 어느 항공기보다 그 중요성이 강조되고 있으며, 고장이 발생하거나 이상상태 발생시에 대한 대비를 갖춘 제어시스템을 탑재해야 한다. 우리나라에서도 유사한 개념의 무인항공기를 개발하는 “스마트 무인기 기술개발” 사업이 과기부 프론티어 사업으로 선정되어, 앞으로 10년동안 이러한 개념의 항공기를 개발하기 위해 연구개발에 많은 노력을 기울이고 있다. 또한, 앞으로 20년 이내에 우리나라도 독자적인 기술로 F-16급 항공기 및 X-45급 자율무인항공기를 개발하고자 하고 있다. 따라서 현재 본 국가지정연구실에서 수행하고 있는 연구의 중요성이 더욱 부각되고 있는 실정이며, 본 과제를 수행함으로써 얻어지는 연구결과가 스마트 무인기 및 고성능항공기 비행제어시스템의 개발자료로 적극 활용될 수 있을 것으로 사료된다.

통상적인 환경에서 작동하는 자동제어시스템 기술이나 시스템에 이상이 생겼을 경우에도 지능적으로 작동하는 통합 관리시스템 기술은 자동차 분야에서 항법정보와 교통정보를 통합한 자동운항시스템, 각종 센서와 자동차 운항정보를 이용한 충돌방지시스템, 고장진단시스템 등에 적용될 수 있으며, 자율이동로봇의 제어시스템 설계에 사용이 가능하다. 본 연구의 수행결과는 향후 국내의 항공우주 산업발전, 차세대 자동차 운항시스템 개발, 차세대 쾌속선 운항시스템 개발 등을 위해 항법 및 제어시스템 소프트웨어 자체개발 및 성능검증을 할 수 있는 기술을 확보할 수 있는 계기로 활용될 것으로 사료된다. 이렇듯 광범위하게 적용될 통합 비행제어시스템의 설계기술은 다음에 제시되는 방법을 통하여 활용, 전파될 수 있다.

▶ 기술 세미나 및 논문발표회 개최를 통한 통합 비행제어시스템 기술의 향상도모

정기적으로 국내외 대학 및 연구소의 전문가 모임을 개최하여 관련기술의 발전 동향이나 기술향상에 도움이 되는 현장기술, 이론 등을 발표 및 토론함으로써 관련분야의 기술에 대한 이해증진을 통한 기술향상을 도모한다. 또한, 관련 연구를 수행하는 국내대학 연구실에 기술지도 및 공동연구를 통해 분산된 연구능력을 통합하여 국내 기술발전 및 국제적 수준으로의 학문수준 발전을 도모한다.

▶ 국내 산학연 협력을 통한 기술지원 및 컨설팅 역할 수행

비행제어시스템 관련 산업체와 연구소에 산학연 협력의 일환으로 본 연구에서 구현된 통합 비행제어시스템 기술지원을 수행한다. 한 예로 현재 국내에서 자체개발하고 있는 무

인비행체, 성층권 비행선, 과학로켓 시스템 등의 시스템 설계에 본 연구기술을 활용함으로써, 자체 설계능력 확보를 위한 기술지원으로 쓰일 수 있을 것이다. 또한, 현재 논의하고 있는 국내 항공산업체와의 기술교류 및 공동연구를 활성화시켜 기술개발을 촉진시킨다. 그리고 축적된 기술 및 노하우를 활용하기 위한 방안으로 본 국가지정연구실 홈페이지를 통해 현장에서 발생하는 시스템 설계문제와 성능향상에 대한 자문 및 해결방안을 제시하여 주고 개발방향을 제시하여 준다.

▶ 개발된 통합 비행제어기술의 홍보

국내외 학술지 논문발표, 개설되어 운영중인 인터넷 홈페이지를 통하여 기술개발 과정에서 획득한 새로운 이론들과 해석, 설계과정 중에서 학술적 가치가 있는 결과들을 홍보한다. 예를 들면, 비행체의 다양한 수행업무 특성에 맞는 구체적인 통합 비행제어시스템 설계 결과를 인터넷 홈페이지를 통해 게시함으로써, 연구된 기술의 우수성과 활용가능성을 검색할 수 있도록 제시 및 홍보한다.

▶ 고장진단 및 대처 관련 기술 특허획득

제안된 연구기술 개발과정에서 축적된 고장진단 및 대처 기술 중 상품화 가치가 있는 결과에 대한 특허획득을 시도한다. 또한, 고장진단과 이상현상 발생시 대처하기 위한 시스템 설계가 필요한 산업부문에 공급함으로써, 독자적인 기술력으로 제품생산과 성능해석이 가능해질 수 있는 기반을 마련하도록 한다.

제 6 장 참고 문헌

- [1] Clark, R.N., Frank, P.M., and Patton, R.J., *Fault Diagnosis in Dynamic Systems: Theory and Applications*, Patton R.J. et al. (ed.), Prentice Hall, New York, 1989, Chap.1.
- [2] Federal Aviation Administration, *Federal Aviation Administration Plan for Research, Engineering & Development(R,E&D) Plan*, 1997.
- [3] Patton, R.J., "Robust Model Based Fault Diagnosis: The State of the Art," *Proceedings of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 94*, 1994, pp.1~24.
- [4] Beard, R., *Failure Accommodation in Linear Systems through Self-Reorganization*, Rep. MVT-71-1, Man-Vehicle Lab., MIT, Cambridge, MA, February 1971.
- [5] Jones, H.L., *Failure Detection in Linear Systems*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, MIT, 1973.
- [6] White, J.E., and Speyer, J.L., "Detection Filter Design: Spectral Theory and Algorithms," *IEEE Transactions on Automatic Control*, Vol.32, No.7, 1987, pp.593~603.
- [7] Massoumnia, M., "A Geometric Approach to the Synthesis of Failure Detection Filters," *IEEE Transactions on Automatic Control*, Vol.31, No.9, 1986, pp.839~846.
- [8] Park, J., and Rizzioni, G., "An Eigenstructure Assignment Algorithm for the Design of Fault Detection Filters," *IEEE Transactions on Automatic Control*, Vol.39, No.7, 1994, pp.1521~ 1524.
- [9] Douglas, R.K., and Speyer, J.L., "Robust Detection Filter Design," *Journal of Guidance, Control, and Dynamics*, Vol.19, No.1, 1996, pp.214~218.
- [10] Kim, D., and Kim, Y., "On the Optimal Location of Sensors for Robust Fault Detection Filter," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, August 1997, pp.1486~1493.
- [11] Shen, L.-C., Chang, S.-K., and Hsu, P.-L., "Robust Fault Detection and Isolation with Unstructured Uncertainty with Eigenstructure Assignment," *Journal of Guidance, Control, and Dynamics*, Vol.21, No.1, 1998, pp.50~57.
- [12] Frank, P.M, and Wünnenberg, J., *Robust Fault Diagnosis using Unknown Input Observer*, *Fault Diagnosis in Dynamic Systems: Theory and Applications*, Patton R.J. et al. (ed.), Prentice Hall, New York, 1989, Chap.3.
- [13] Frank, P.M., *Fault Diagnosis in Dynamic Systems via State Estimation-A Survey*, *System Fault Diagnostics, Reliability and Related Knowledge Based Approaches*, Tzafetas et al, (ed), D. Reidel Press, 1987.
- [14] Saif, M., and Guan, Y., "A New Approach to Robust Fault Detection and Identification," *IEEE Transactions on Aerospace and Electronic Systems*, Vol.29, No.3, 1993, pp.685~695.
- [15] Chen, J., and Patton, R.J., "Optimal Filtering and Robust Fault Diagnosis of Stochastic Systems with Unknown Disturbances," *IEE Proceedings-D: Control Theory and Applications*, Vol.143, No.1, 1996, pp.31~36.
- [16] Chen, J., Patton, R.J., and Zhang, H.Y., "Design of Unknown Input Observers and Robust Fault Detection Filters," *International Journal of Control*, Vol.63, No.1, 1996, pp.85~105.
- [17] Kim, D., and Kim, Y., "Comments on Design of Unknown Input Observers and Robust Fault Detection Filter," *International Journal of Control*, Vol.67, No.2, 1997, pp.305~306.
- [18] Jin, J.H., Tahk, M.J., and Park, C.S., "Time-Delayed State and Unknown Input Observation," *International Journal of Control*, Vol.66, No.5, 1997, pp.733~745.
- [19] Patton, R.J., and Chen, J., "Robust Fault Detection of Jet Engine Sensor Systems Using Eigenstructure Assignment," *Journal of Guidance, Control and Dynamics*, Vol.15, No.6, 1992, pp.1491~1497.

- [20] Patton, R.J., and Chen, J., "Optimal Unknown Input Distribution Matrix Selection in Robust Fault Diagnosis," *Automatica*, Vol.29, No.4, 1993, pp.837~841.
- [21] Patel, Y., Patton, R.J., and Burrows, S.P., "Design of Insensitive Multirate Aircraft Control using Optimized Eigenstructure Assignment," *Journal of Guidance, Control and Dynamics*, Vol.16, No.1, 1993, pp.118~123.
- [22] 남윤수, 고준수, "연구개발용 항공기 시뮬레이터 응용기술," 제어자동화시스템공학회지, 제3권, 제2호, 1997, pp.64~77.
- [23] Kim, B.S., and Calise, A.J. "Nonlinear Adaptive Flight Control Using Neural Networks," *Journal of Guidance, Control and Dynamics*, Vol.20, No.1, 1997, pp.26~33.
- [24] Lee, T., and Kim, Y., "Nonlinear Adaptive Flight Control Using Backstepping and Neural Networks Controller," accepted for publication in *Journal of Guidance, Control, and Dynamics*, 2001.
- [25] Ward, D.J., Monaco, J.F., and Bodson, M., "Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control," *Journal of Guidance, Control and Dynamics*, Vol.21, No.6, 1998, pp.948~956.
- [26] Bodson, M., and Groszkiewicz, J.E., "Multivariable Adaptive Algorithms for Reconfigurable Flight Control," *IEEE Transactions on Control Systems Technology*, Vol.5, No.2, 1997, pp.217~229.
- [27] Kim, D., and Kim, Y., "Robust Variable Structure Controller Design for Fault Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol.23, No.3, 2000, pp.430~437.
- [28] Ochi, Y., and Kanai, K., "Application of Restructurable Flight Control System to Large Transport Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol.18, No.2, 1995, pp.365~370.
- [29] Kim, D., and Kim, Y., "Robust Variable Structure Controller Design for Fault Tolerant Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol.23, No.3, 2000, pp.430~437.
- [30] Schrage, D.P., and Vachtsevanos, G., "Software Enabled Control for Intelligent UAVs," 1999 IEEE International Conference on Control Applications, 1999.
- [31] O.Sokolsky, "Verification of the Redundancy Management System for Space Launch Vehicle: A Case Study", Fourth IEEE Proceedings of Real-Time Technology and Applications Symposium, 1998, pp.220~229.
- [32] S. Osder, "Practical View of Redundancy Management Application and Theory," *Journal of Guidance, Control and Dynamics*, Vol. 22, No. 1, 1999.
- [33] J. K. Ramage, "Development of the AFTI/F-16 Triplex Digital Flight Control System", AIAA Paper 83-2278-CP, 1983.
- [34] Tom Sadeghi and Farzin Motamed, "Evaluation and Comparison of Triple and Quadruple Flight Control Architectures", *IEEE AES Systems Magazine*, march, 1992
- [35] V. A. Megna, "Multi-Flight Computer Redundancy Management for Digital Fly-By-Wire Aircraft Control," *COMPCON Fall '77*, 1977, pp.378-384.
- [36] M. K. Jeerage, "Reliability Analysis of Fault Tolerant IMU Architecture with Redundant Inertial Sensors," *IEEE AES Magazine*, Vol. 5, No. 7, 1990.
- [37] J. H. Lala and R. E. Harper, "Architectural Principles for Safety-Critical Real-Time Application," *Proceeding of IEEE*, Vol. 82, No 1, 1994. pp. 25-40.
- [38] T. F. Westermeier, "Triplex Digital Fly-by-Wire Redundancy Mangement Technique", AIAA, 1978.
- [39] Y. C. Yeh, "Design Considerations in Boeing 777 Fly-By-Wire Computers," *Proceedings of the High-Assurance Systems Engineering Symposium*, IEEE International, 1988, pp. 64-72
- [40] B. W. Kowal, . J. Scherz and R. Quinlivan, "C-17 Flight Control System", *IEEE/AIAA 11th Digital Avionics Systems Conference*, 1992, pp. 580-585

특정연구개발사업 연구결과 활용계획서

사업명	중사업명	특정연구개발사업		
	세부사업명	국가지정연구실사업		
과제명	지능형 통합비행제어시스템 설계기법 에 관한 연구			
연구기관	서울대학교 비행역학 및 제어 연구실	연구책임자	김 유 단	
총연구기간	2001년. 7월. 11일. ~ 2003년. 7월. 10일. (24개월)			
총 연구비 (단위 : 천원)	정부출연금	민간부담금	합계	
	379,618	0	379,618	
기술분야				
참여기업				
공동연구기관				
위탁연구기관				
연구결과활용 (해당항목에(√) 표시)	1. 기업화 ()	2. 기술이전()	3. 후속연구추진()	4. 타사업에 활용()
	5. 선행 및 기 초연구(√)	6. 기타목적활용 (교육,연구)(√)	7. 활용중단(미활용)()	8. 기타()

특정연구개발사업 처리규정 제 31조(연구개발결과의 보고) 제 2항에 의
거 연구결과 활용계획서를 제출합니다.

첨부 : 1. 연구결과 활용계획서 1부.
2. 기술요약서 1부

2003 년 9 월 1 일

연구책임자 : 김 유 단 (인)

연구기관장 : 서울대학교 총장 (직인)

과학기술부장관 귀하

[첨부1]

연구결과 활용계획서

1. 연구목표 및 내용

항공기가 보다 넓은 비행영역에서 개선된 비행성(Flying Quality)을 갖도록 지능형 비행제어시스템을 구현하고, 더 나아가 이상현상(Malfunction)이 발생하였을 경우에 이를 검출 및 분리하고 적절히 대처할 수 있도록 제어법칙을 재설계(Reconfiguration)할 수 있는 통합 비행제어시스템을 구현한다. 통합 비행제어시스템의 구현을 위하여 1단계에서는 신경회로망/퍼지로지 제어 등을 이용한 지능형 Mid-level 비행제어 알고리즘에 대한 연구를 수행하고, 비행제어시스템의 고장진단에 대한 하드웨어 해법(Hardware Solution)에 해당하는 알고리즘인 다중화 관리기법(RM: Redundancy Management) 시스템의 설계와 이를 검증하기 위한 HILS(Hardware In the Loop Simulation) 시스템을 구축하는 것을 목표로 한다. 2단계에서는 1단계에서 완성된 다중화 관리시스템을 바탕으로 제어시스템을 재설계하여 이상현상에 대처할 수 있는 통합 비행제어시스템을 구현한다. 이는 고성능 전투기나 여객기의 지능형(Intelligent) 비행제어시스템을 구축하기 위한 핵심적인 기반기술로 사용할 수 있을 것이다.

2. 연구수행결과 현황(연구종료시점까지)

가. 특허(실용신안) 등 자료목록

발명명칭	특허공고번호 출원(등록)번호	공고일자 출원(등록)일자	발명자 (출원인)	출원국	비고

나. 프로그램 등록목록

프로그램 명칭	등록번호	등록일자	개발자	비고

다. 노하우 내역

라. 발생품 및 시작품 내역

마. 논문게재 및 발표 실적

【총괄표】

구분	기술이전		기술료 수입			특허출원			특허등록			논문게재			학술발표		
	건수	건수	금액 (억원)	국내	국외	계	국내	국외	계	국내	국외	계	국내	국외	계		
1차년도 ('01)	0	0		0	0	0	0	0	0	0	0	0	1	4	5		
2차년도 ('02)	0	0			0		0	0	0	4*	5* (3**)	9* (3**)	7	8	15		
총계	0	0			0		0	0	0	4*	5* (3**)	9* (3**)	8	12	20		

* : 게재/게재 승인된 논문 편수, **: 심사 중인 논문 편수

논문제목	저자	학술지명	제출일	발행년도 권호	발행기관 (국명)	SCI여부 (O,X)
Reconfigurable Flight Control System Using Direct Adaptive Method	김기석 이금진 김유단	Journal of Guidance, Control, and Dynamics	4/26/ 2002	2003 Vol.26 No.4	AIAA (USA)	○
Design of Generalized Conceptual Guidance Law Using Aim Angle	김기석 김유단	Control Engineering Practice	7/2/ 2002	2003 게재승인	Elsevier (UK)	○
Robust Backstepping Control for Slew Maneuver Using Nonlinear Tracking Function	김기석 김유단	IEEE Transactions on Control Systems Technology	5/17/ 2002	2003 게재승인	IEEE (USA)	○
Reconfigurable Flight Control System Design Using Adaptive Neural Networks	신동호 김유단	IEEE Transactions on Control Systems Technology	12/10/ 2002	2003 게재승인	IEEE (USA)	○
Optimum Design of Neural Networks for Nonlinear Flight Control System	최규호 최동욱 김유단	Engineering Optimization	2/18/ 2003	2003 게재승인	Taylor & Francis (UK)	○
Adaptive Structural Control Experiment Using Recursive System Identification	형승용 김유단 석진영	Journal of Intelligent Material Systems and Structures	10/8/ 2002	심사중	SAGE (UK)	○
Design of Impact Angle Controlled Guidance Law Considering Practical Implementation	김기석 김유단	Control Engineering Practice	3/8/ 2003	심사중	AIAA (USA)	○
Design of Reconfigurable Flight Control System Using Adaptive Sliding Mode Control - Actuator Fault	신동호 김유단	IEEE Transactions on Aerospace and Electronic Systems	11/27/ 2002	심사중	IEEE (USA)	○

논문제목	저자	학술지명	제출일	발행년도 권호	발행기관 (국명)	SCI여부 (O,X)
백스테핑기법과 신경회로망을 이용한 적응 재형상 비행제어법칙	신동호 김유단	Journal of Control, Automation, and Systems Engineering	10/16/ 2002	2003 Vol.9 No.4	ICASE (KOR)	X
직접 적응기법을 이용한 모델추종 재형상 비행제어시스템 설계	김기석 이금진 김유단	Journal of Control, Automation, and Systems Engineering	9/13/ 2002	2003 Vol.9 No.2	ICASE (KOR)	X
Reconfigurable Flight Control System Design Using Sliding Mode Based Model Following Control Scheme	조동현 김기석 김유단	항공우주학회지	11/15/ 2002	2003 Vol.4 No.1	KSAS (KOR)	X
신경회로망 구조 최적화를 통한 비행제어시스템 설계	최규호 최동욱 김유단	항공우주학회지	1/20/ 2003	2003 Vol.31 No.7	KSAS (KOR)	X

*주저자가 본 연구실 소속인 경우는 주저자명을 볼드체로 표시

⑦ 학술발표

학술발표제목	저자(발표자*)	학술회의명 (국명)	일 시
모델추종제어기법을 이용한 슬라이딩 모드 재형상 비행제어	조동현*, 김기석, 김유단	항공우주학회 (한국)	2002. 11
Design of Reconfigurable Flight Controller using Sliding Mode Control-Actuator Fault	신동호*, 김유단	International Conference on Control, Automation and Systems (한국)	2002. 10
Design of Generalized Conceptual Guidance Law Using Aim Angle	김기석*, 김유단	일본항공우주학회 JSASS 항공기심포지엄 (국제세션) (일본)	2002. 10
Reconfigurable Flight Control System Design Using Sliding Mode Based Model Following Control Scheme	조동현*, 김유단	일본항공우주학회 JSASS 항공기심포지엄 (국제세션) (일본)	2002. 10
Adaptive Structural Control Experiment Using Recursive System Identification	김유단*, 형승용, 석진영	International Conference on Adaptive Structures and Technologies (독일)	2002. 10
Model Following Reconfigurable Flight Control System Design Using Direct Adaptive Scheme	김기석*, 이금진, 김유단	AIAA, Guidance, Navigation, and Control Conference (미국)	2002. 8
A Study on the Design of Reconfigurable Flight Control System Using Input-Output Information	형승용*, 김유단, 김성필, 김희섭	AIAA, Guidance, Navigation, and Control Conference (미국)	2002. 8
Implemental Impact Angle Controlled Guidance Law Design	김기석*, 김유단	Korea/Japan Joint Workshop on Aeronautics and Astronautics, Seoul National University (한국)	2002. 7
Flight Control System Design Based on System Identification	형승용*, 김성필, 김희섭, 김유단	Korea/Japan Joint Workshop on Aeronautics and Astronautics, Seoul National University (한국)	2002. 7

학술발표제목	저자(발표자*)	학술회의명 (국명)	일 시
직접 적응기법을 이용한 재형상 비행제어 시스템 설계	김기석*, 이금진, 김유단	항공기 개발기술 심포지엄 (한국)	2002. 5
입출력 제적정보를 이용한 재형상 비행제어시스템 설계를 위한 연구	형승용*, 김성필, 김희섭, 김유단	항공우주학회 (한국)	2002. 4
편의 유연모드를 고려한 발사체의 롤축 제어기 설계	김기석*, 김유단	우주발사체 기술 심포지움 (한국)	2002. 3
Design Of Neural Network Adaptive Control Law for Aircraft System Including Uncertainty	신동호*, 김유단	International Conference on Nonlinear Problems in Aviation and Aerospace (미국)	2002. 5
Adaptive Reconfigurable flight Control System Based on Recursive System Identification	김성필*, 김유단, 김희섭, 남창호	일본항공우주학회 JSASS 항공기심포지엄 (국제세션) (일본)	2001. 10
Robust Fault Tolerant Flight Control Systems Using Adaptive Schemes	안충일*, 김유단	일본항공우주학회 JSASS 항공기심포지엄 (국제세션) (일본)	2001. 10
기동력 제한을 고려한 미사일의 비행궤적경로 생성 알고리즘 연구	문관영*, 김기석, 김유단	유도무기 학술대회(한국)	2001. 10
Design of Aircraft Trajectory Considering Constrained Waypoints	문관영*, 김기석, 김유단	Korea/Japan Joint Workshop on Aeronautics and Astronautics, Seoul National University (일본)	2001. 10
Autonomous Mobile Robot Trajectory Following Based on Image Recognition	형승용*, 김성필, 정동원, 문관영, 김유단	Korea/Japan Joint Workshop on Aeronautics and Astronautics, Seoul National University (일본)	2001. 7
Optimum Flight Path Design Passing Through Waypoints for Autonomous Flight Control System	문관영*, 김유단	AIAA, Guidance, Navigation, and Control Conference (미국)	2003. 8
Guidance Law for Evasive Aircraft Maneuvers Using Artificial Intelligence	정보경*, 김기석, 김유단	AIAA, Guidance, Navigation, and Control Conference (미국)	2003. 8

3. 연구성과

※ 기술이전이나 기업화 완료(추진중 포함) 실적

4. 기술이전 및 연구결과 활용계획

가. 당해연도 활용계획(6하원칙에 따라 구체적으로 작성)

항공우주연구원, 국방과학연구소, 한국항공우주산업과 공동연구 등을 통해서 실제 활용 가능한 기술로 발전시키기 위한 공동노력을 수행할 것이다.

나. 활용방법

스마트 무인기, 기술검증기 사업의 비행제어시스템 설계 기초자료로 활용
다. 차년도이후 활용계획(6하원칙에 따라 구체적으로 작성)

항공우주연구원, 국방과학연구소, 한국항공우주산업과 공동연구 등을 통해서 실제 활용 가능한 기술로 발전시키기 위한 공동노력을 수행할 것이다.

5. 기대효과

▶ 고성능 항공기 비행제어시스템 수입대체 효과

우리나라가 독자적으로 연구, 개발하여 통합 비행제어시스템 설계기술을 확보할 경우, 국내 산업체 및 연구소에서 제작하는 항공기에 자체적으로 개발한 통합 비행제어시스템을 장착할 수 있기 때문에 직접적인 수입대체 효과를 얻을 수 있다. 우리나라의 항공분야 해외 기술도입 추진실정을 [표 1]에 정리하였다. (추진중에 중단된 사업도 있으나 기술도입료 수준을 가늠하기 위해서 원래의 자료를 그대로 정리하였다)

표 1. 해외 기술도입 실적

(단위: 천달러)

구분	도입기술명	기간	국가	기술도입료	
완제기	고정익	F16 C/D전투기의 기체조립 및 부품 제작기술	91-06	미국	189,261
		F5 전투기 수리개조기술	94-05	미국	3,000
		KTX-2 항공기 상세설계, 제작, 시험평가 관련기술	97-07	미국	237,000
	회전익	UH60 블랙호크 헬기 및 동부분품	90-05	미국	28,000
		W-3A SOKOL 헬기 제조기술	96-05	폴란드	5,500
		소형 5인승 정찰헬기의 국내조립 생산기술	98-07	독일	2,370
동체 및 날개	MD95 날개제조기술	95-10	미국	20,000	

출처 : 한국항공우주산업 진흥협회 조사, '국내 항공우주산업 통계'

차세대 항공기를 개발할 경우에 국내에서 개발한 비행제어시스템을 장착하지 않더라도, 동급의 기술을 우리나라가 보유하고 있다고 항공산업 선진국들이 판단할 경우에는 선진국에서 개발한 우수한 비행제어시스템을 저가로 공급하거나 한 수준 높은 비행제어시스템을 공동개발 형식으로 제안할 가능성도 있으므로, 이에 따른 경제적 효과는 상당히 크다고 사료된다. 통합 비행제어시스템 설계기술은 항공기 설계기술에서 핵심적 위치에 있기 때문에 비행제어시스템 설계기술이 우리나라의 항공기술을 평가하는 지표로 사용되어, 비행제어시스템 설계기술분야 이외의 항공기 설계기술분야에 그대로 적용될 수 있을 것이다. 장기적인 관점에서 바라볼 때 비행제어시스템 설계기술은 급격하게 변화하기 때문에, 기술개발 투자시기가 늦어지면 늦어질수록 미래에 국내에서 추진할 항공기 개발사업에서 더 높은 가격으로 통합 비행제어시스템을 수입해야 할 것이다.

▶ 경제적 기대효과의 정량적 예측

본 연구과제를 통해 얻어지는 경제적 효과를 보다 구체적으로 이해하기 위해 정량적인 예측을 수행해 보기로 한다. [표 2]은 1985년에서 1995년까지 11년간 항공우주산업이 제조업 전체의 총산출액 평균성장률인 12.8%를 훨씬 상회하는 21.8%의 성장률을 보이고 있다는 것을 보여주고 있다.

표 2. 항공우주산업 총산출액

(단위 10억원)

연 도	1985	1990	1995	2000	CAGR(85-95)	CAGR(85-200)
제조업 총산출액	97,007	176,440	364,822		12.8%	
항공우주산업 총산출액	77	156	675	1,309	21.8%	22.6%

출처 : 한국항공우주산업 진흥협회 조사, '한국 항공우주산업통계', 1997-1998년

[표 2]에서 볼 수 있듯이 항공산업의 발전추세는 전체적으로 경제상황이 좋지 않은 1995-2000년에도 지속되어, 1985년에서 2000년까지의 연평균 성장액은 22.6%에 이르고 있다. 그러나 이와 같은 급격한 발전에도 불구하고 항공우주분야는 대규모의 무역적자가 발생하고 있는 상황이다. 1999년 말 현재 항공우주분야의 수입초과액은 무려 515억불에 달해, 항공우주산업 분야의 기술발전이 시급히 요구됨을 알 수 있다.

본 연구과제는 이처럼 시급한 항공우주분야의 기술개발 과정에서 여타 기술들에 비해 기술도입이 극히 어려운 항공기 비행제어분야의 기초기술을 확보하고자 하는 것이다. 해외 사례를 보면 일반적으로 항공우주산업에서 항공전자장비를 포함한 비행제어기술의 비중은 30% 내외인 것으로 파악되며, 그 비중이 확대되는 추세에 있다. 따라서 본 연구가 성공적으로 완료된다면, [표 3]에서 정리되어 있는 것과 같은 경제적 효과를 창출할 것이라고 예측할 수 있다.

표 3. 본 연구의 경제적 효과예측 (단위 10억원)

연도	2005	2010	2015	2020
항공우주산업 예상매출액	4,553	12,557	34,780	96,334
항공우주 제어기술 관련 매출액	1,360	3,767	10,434	28,900
연구과제를 통한 가치창출 효과 ⁽¹⁾	68	111	182	298

(주1) 본 연구과제의 효과가 2005년에는 전체 항공제어기술의 5%가 될 것으로 전망하였으며, 여타 기술의 발전에 따라 그 효과가 매년 10%씩 감소할 것으로 가정하였는데, (예: 2020에는 전체 항공제어기술에서 1%만 차지함) 사실상 이는 매우 보수적인 가정임.

이와 같은 예측은 매출액을 매년의 기준으로 계산하여 산출한 것이므로, 본 연구과제에 투자되는 예산 및 이익을 2000년 불변가격을 기준으로 계산하면, 결론적으로 [표 4]와 같은 예측을 할 수 있다. 즉, 본 연구과제를 통해 창출될 수 있는 효과는 보수적으로 산출하더라도 1000배가 넘는 것으로 판단된다.

표 4. 본 연구의 투자대비 수익예측 (단위: 10억원)

총투자액	1.39	2000년 불변가액 기준, 매년 5%의 물가상승 고려
총 가치창출액 (2005~2020년)	1,620	
연구개발 투자 대비 수익 (배)	1,165	

6. 문제점 및 건의사항(연구성과의 제고를 위한 제도·규정 및 연구관리 등의 개선점을 기재)

- 1) 기술의 발전이 매우 빠르게 이루어짐에 따라 실험용기자재의 신제품 시판도 빈번하게 이루어진다. 기자재 및 재료비의 사용용도 변경을 신속하고 간편하게 할 수 있는 제도적인 장치가 마련되었으면 한다.
- 2) 결제 수단 경직화(신용카드만을 사용)로 인하여 연구재료 구입 시 애로사항이 많다.

[첨부2]

기술 요약서

■ 기술의 명칭

지능형 통합 비행제어시스템 설계기법

■ 기술을 도출한 과제현황

과제관리번호	M1-0104-00-0028		
과제명	지능형 통합 비행제어시스템 설계기법 연구		
사업명	특정연구개발사업		
세부사업명	국가지정연구실사업		
연구기관	서울대학교 비행역학 및 제어 연구실	기관유형	산(),학(O),연()
참여기관(기업)			
총연구기간	2001. 7. 11 ~ 2003. 7. 10 (2년)		
총연구비	정부(379,618)천원 민간(0)천원 합계(379,618)천원		
연구책임자 1	성명	김 유 단	주민번호
	근무기관 부서	서울대학교 기계항공공학부	E-mail ydkim@snu.ac.kr
	직위/직급	교 수	전화번호 02-880-7398
연구책임자 2	성명		주민번호
	근무기관 부서		E-mail
	직위/직급		전화번호
실무연락책임자	성명		소속/부서
	직위/직급		E-mail
	전화번호		FAX
	주소	(-)	

■ 기술의 주요내용

[기술의 개요]

- 넓은 비행영역에 적용 가능한 비행제어 알고리즘의 개발
- 신경회로망, 퍼지로지, 유전자 알고리즘을 이용한 지능형 제어기 설계 방법 연구
- 항공기의 생존성을 높이기 위한 고장허용 제어시스템 설계기법 연구
- 하드웨어 다중화 관리기법에 관한 이론 연구와 적용 가능성에 대한 검증을 수행
- 다중 센서에 대해 voting 알고리즘을 구성하여 다중화 시스템의 하드웨어 해법 구현
- 해석적 알고리즘을 구현하기 위한 필터 설계기법 연구
- 고장인식 과정이 필요 없는 직접 적응제어기법, 고장인식 과정을 포함한 슬라이딩 모드제어 기법, 입출력 정보를 이용한 시스템 인식기법을 사용한 기법 등을 개발
- 제안된 알고리즘을 하드웨어적으로 검증하기 위한 MB(Moving Base)-HILS 시스템 구현

<기술적 특징>

- (1) 지능적 비행제어 시스템을 개발함으로써 항공기 신뢰성 및 성능 향상
- (2) 저가형 비행제어시스템 개발환경 구축을 통한 비행제어시스템 개발의 국산화 가능

[용도·이용분야]

- (1) 항공기 비행제어시스템 설계
- (2) 차세대 자동차 자동운항시스템 등 여타 분야로의 기술적 파급효과
- (3) 각종 군장비의 안정성 보장 및 신뢰성 확보

■ 본 기술과 관련하여 추가로 확보되었거나 개발중인 기술

[기술개요]

기술명	고장진단 및 다중화 기술
개발단계	<input type="checkbox"/> 연구개발 계획 <input checked="" type="checkbox"/> 연구개발 중 <input type="checkbox"/> 연구개발 완료
기술개요	무인항공기에 고장이 발생하였을 경우에 이를 검출 및 분리하고 적절히 대처할 수 있도록 하는 고장진단 및 다중화 관리 기술을 연구하여 다중화 시스템 구현을 위한 설계 기술을 확보하는 것을 최종 목표로 한다.

[기술을 도출한 과제현황]

과제관리번호	M1-02-KQ-01-0001-02-K17-01-011-0-0			
과제명	고장진단 및 다중화 기술 개발			
사업명	연구개발사업			
세부사업명	21세기 프론티어 연구개발사업(스마트 무인기 기술개발사업)			
연구기관	서울대학교 비행역학 및 제어 연구실	기관유형	산(), 학(O), 연()	
참여기관(기업)				
총연구기간	2002. 11. - 2005. 3 (28.5개월)			
총연구비	합계 : (124)백만원 - 정부 : (124)백만원 민간 : (0)백만원			
연구책임자	소속	서울대학교 기계항공공학부	성명	김 유 단
	전화번호	02-880-7398	E-mail	ydkim@snu.ac.kr
연구개발 주요내용				
<ul style="list-style-type: none"> ▶ 일반화 된 고장검출 및 다중화 기법 연구 ▶ 시스템 신뢰성 분석 기초 연구 ▶ 무인기 고장검출 및 다중화 관리 기법 연구 ▶ 고장원인 및 영향분석을 위한 고장진단 체계 정립 ▶ 스마트 무인기 신뢰성 분석 연구 				