

## 論文

# SIMULINK를 이용한 소형 항공기용 자동비행시스템 시뮬레이터 구현

이동규\*, 채동한\*\*, 이상철\*\*\*, 오화석\*\*\*\*

## Implementation of the small aircraft simulator with autopilot system using SIMULINK

Dongkyu Lee\*, Dong-han Chae\*\*, Sangchul Lee\*\*\*, Hwasuk Oh\*\*\*\*

### ABSTRACT

In modern aircraft, an autopilot system is getting more important. There are not many autopilot systems applied to small aircraft. Also the autopilot system in large or medium aircraft is difficult to apply to small aircraft directly. It is necessary to make a new autopilot system for small aircraft. In this paper, we implement the small aircraft simulator with autopilot system using SIMULINK. The various modes of autopilot - such as altitude select/hold, attitude hold, heading hold, etc. - are implemented to the flight simulator and tested. We also implement the VOR mode for aircraft guidance.

Key Words : Autopilot(자동비행조종장치), DHC-2 'Beaver', DME(거리 측정 장치), Flight Simulator(비행 시뮬레이터), SIMULINK(시뮬링크), Simulator(시뮬레이터), VOR(극초단파 전방향 무선 표지)

### I. 서 론

현대 항공기 운항에 있어서 자동비행시스템의 중요도와 비중은 계속 증가되고 있다[1]. 또한 중, 대형기 이상에서 주로 사용되던 자동비행시스템이 점차 소형기에까지 그 적용이 확대되고 있는 상황이다. 그러나 중·대형기의 자동비행시스템은 그 구성과 성능을 소형 항공기에 바로 적용하기에는 무리가 있으며, 이러한 이유로 소형 항공기를 위한 자동비행시스템 개발이 필요하다.

이러한 시스템 구성 과정에서 시뮬레이터의 구현은 중요한 역할을 한다. 실제 시스템을 물리적으로 구성하기 전에 다양한 상황에 대한 시스템의 반응을 미리 확인 또는 검증할 수 있고, 결

과적으로 새로운 시스템에 대한 신뢰도를 높일 수 있기 때문이다.

본 논문에서는 SIMULINK[2]를 활용하여 소형 항공기에 적합한 자동비행시스템을 구현하였다. 또한 항공기가 다수의 경로점을 경유하여 목표 위치에 도달할 수 있는 시스템을 구현하고 이에 대하여 시뮬레이션을 수행하였다. 이 시스템은 VOR 시스템을 기반으로 하여 구현하였으며, 이미 광범위하게 사용되고 있기 때문에 활용 가능성이 높다.

### II. 항공기 모델

본 논문의 시뮬레이터 구성을 위하여 The De Havilland사(社)에서 제작된 DHC-2 'Beaver'(이하 Beaver) 항공기 동특성을 SIMULINK로 모델링한 FDC 1.4를 활용하였다[3, 4]. FDC 1.4는 실시간 시뮬레이션이 불가능하다. 실시간 시뮬레이션이 가능하도록 이를 재구성하고 추가 Coding을 수행하였다.

<Fig 1>은 Beaver 항공기를 모델링한 Blockset

2008년 9월 24일 접수 ~ 2008년 9월 29일 심사완료

\* 한국항공대학교 대학원 항공우주 및 기계공학과

\*\* 한국항공대학교 항공우주 및 기계공학부 학부생

\*\*\* 한국항공대학교 항공우주 및 기계공학부

연락처자, E-mail : slee@kau.ac.kr

경기도 고양시 덕양구 화전동 항공대길 100

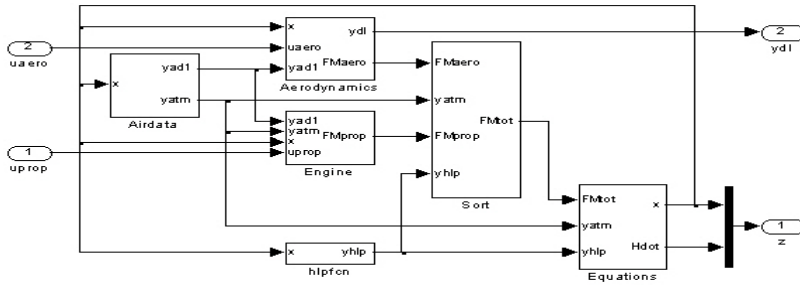


Fig. 1 Aircraft Model

이다. 대기를 모델링한 Airdata Blockset, 엔진을 모델링한 Engine Blockset과 더불어 항공기에 작용하는 공기역학적 힘 F와 모멘트 M을 계산하는 Aerodynamics Blockset이 있다. 이렇게 계산된 값들은 Sort Blockset을 통하여 항공기에 작용하는 모든 힘과 모멘트로 통합되고, Equation Blockset을 통하여  $y_{dl}$ (롤, 피치, 요각의 변화율을 무차원화한 변수)과  $z$ (항공기 상태를 나타내는 13개의 항공기 상태 변수)를 산출한다.

### III. 자동비행시스템 구현

본 논문의 시뮬레이터에는 총 6가지의 자동비행 모드가 구현되었다. 항공기의 피치각 변화와 연관되는 Pitch Attitude Hold(PAH), Altitude Hold(ALH), Altitude Select(ALS) 모드와 롤각과 요각을 제어하는 Roll Attitude Hold(RAH), Heading Hold(HH) 모드, 마지막으로 VOR 모드이다.

#### 3.1 Autopilot Blockset

자동비행을 구현하기 위한 Blockset이며, <Fig 2>에

도시되어 있다.  $y_{dl}$ ,  $z$ ,  $AP_{modref}$ (수행하고자 하는 자동비행 모드에 대한 10개의 on/off 데이터와 5개의 Reference 데이터),  $VOR_{data}$ 를 입력변수로 사용한다.

또한 Autopilot Blockset은 자동비행 모드가 시작되는 순간의 항공기 상태에 대한  $z$ 데이터를 필요로 한다. 본 논문에서는 시뮬레이터가 가동되는 것과 동시에 자동비행 모드에 들어가게 되므로, 시뮬레이터가 시작되는 순간의 항공기 초기 상태값  $x_{inco}$ 와  $\dot{H}_0$ 를 그대로 사용하였다. 그러므로 시뮬레이터를 가동하고 있는 중간에 자동비행 모드를 재설정하게 되면 정확한 비행이 이루어지지 않을 수도 있다.

이 Blockset에는 종방향, 횡방향 운동에 대한 제어를 수행하는 Symmetric, Asymmetric Blockset을 포함하고 있다. 여기서 계산된 데이터는 Act & Cable Dynamics를 통하여 비로소 롤, 피치, 요 값으로 계산되어 출력된다. 마지막으로 VOR Blockset은 VOR 기지국에 대한 모델링과 VOR 주변에서의 항공기의 비행 상태를 지시해주는 역할을 한다.

#### 3.2 모드 선택 패널

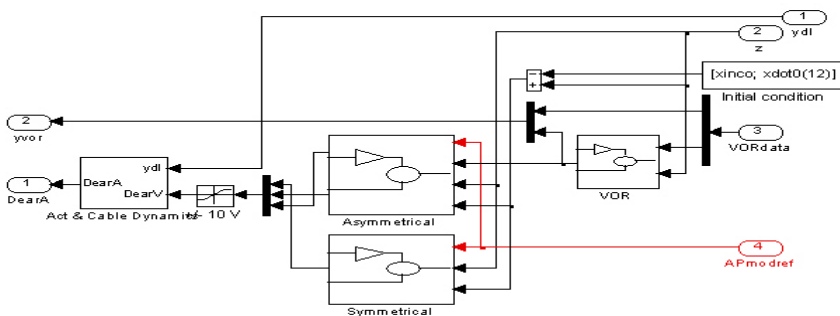


Fig. 2 Autopilot Blockset

Table 1. 각 요소의 정의 및 Armed/Coupled 상황에서의 On(= 1)/Off(= 0)

Symmetrical	Selected Mode			Asymmetrical	Selected Mode		
Switch	PAH	ALH	ALS	Switch	RAH	HH	NAV
Mode	1/0	1/0	1/0	Mode	1/0	1/0	1/1
Outer-Loop	0/0	1/0	1/0	Outer-Loop	0/0	1/0	1/1
ALH	0/0	1/0	0/0	HH	0/0	1/0	1/0
ALS	0/0	0/0	1/0	NAV	0/0	0/0	0/1

<Fig 3>은 본 프로그램에서 가능한 6가지의 모드(그림의 좌측)와 각 모드에서 지정할 수 있는 5가지 값(그림의 우측)을 선택할 수 있는 제어 패널이다. 2개 이하의 모드를 지정해주고 일정 시간동안 시뮬레이션 하는 종래의 방식을 개선하여, 프로그램이 구동되는 동안에도 원하는 자동비행 모드를 선택할 수 있도록 스위치를 제작하였다.

각각의 모드는 두 가지의 상황으로 구별된다. <Table 1>을 보면 항공기가 순항하면서 특정 모드를 작동시켰을 때의 Armed 상태와 VOR의 전파 수신 범위 안에 들어왔을 경우의 Coupled 상태이다.

따라서 항공기가 현재 어떠한 상태로 비행을 하고 있는냐에 따라서 패널에서 선택한 각 모드가 서로 다른 역할을 하게 된다. 예를 들어 NAV 모드를 선택하였을 경우 항공기가 VOR의 전파 범위 내에 없을 때에는 HH 모드로 자동비행을 하게 되고, 전파 범위 내에 들어오면 NAV 모드가 작동한다.

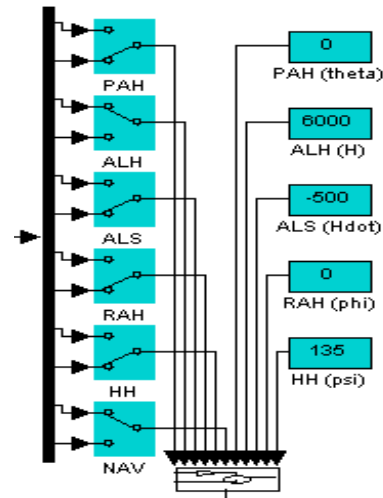


Fig. 3 모드 선택 패널

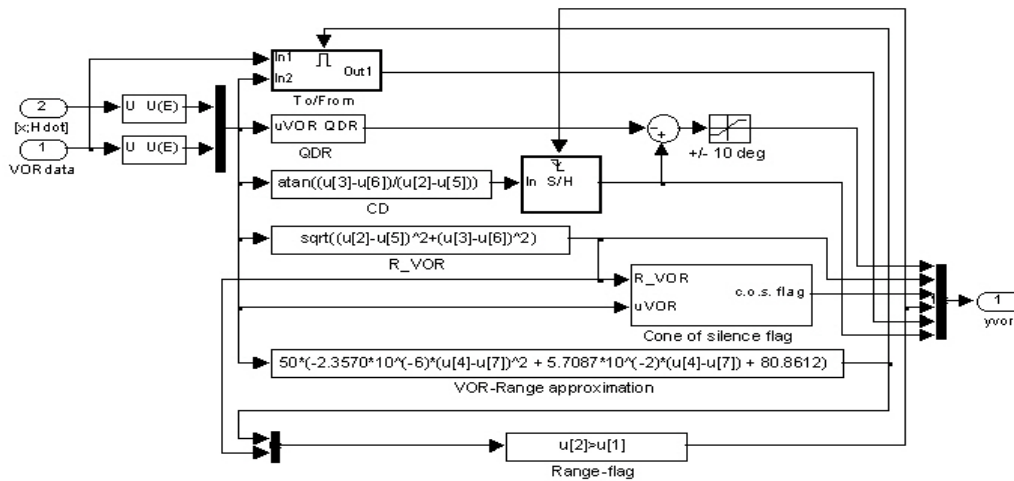


Fig. 4 VOR Blockset

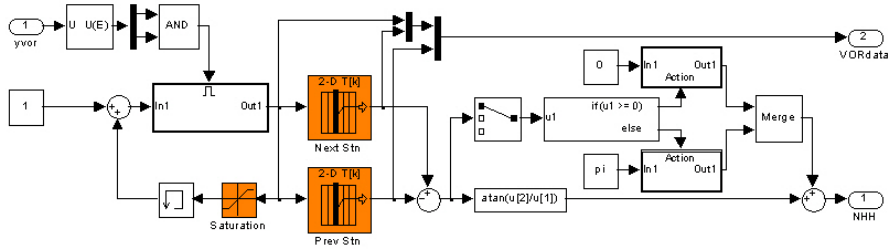


Fig. 5 VOR data Blockset

#### IV. VOR 모델링

본 논문에서는 VOR 기능을 활용하여 경로비행이 가능하도록 하였다. 다수의 VOR 기지국을 지정하였을 경우 최초의 기지국은 사용자가 기지국의 개략적 위치를 생각하여 입력한 Heading 데이터 값을 따라 이동하게 하였고, 입력된 좌표 데이터를 순차적으로 따라간 후, 최종 기지국을 통과하게 되면 초기에 지정된 Heading 방향으로 이동하도록 하였다.

VOR은 전파를 송출하는 장치이고, 항공기가 이 전파 범위 안에 들어 왔을 때 자신의 방위 정보를 알 수 있다. 이러한 기능이 가능하도록 하기 위해서는 전파 송·수신에 대한 모델링이 필요하다. 모델링을 단순화하기 위하여 항공기와 VOR 기지국 사이의 거리를 이용하였다. 또한 시뮬레이션을 수행함에 있어 실제 VOR의 범위를 그대로 적용할 경우 시뮬레이션 시간이 매우 길어지게 된다. 그러므로 VOR의 전파 범위를 축소(8870m)시켜서 순항 비행시간을 줄였다.

##### 4.1 VOR Blockset

<Fig. 4>는 항공기의 요각과 3개의 좌표점, 현재 VOR 기지국과 앞으로 찾아갈 VOR 기지국의 좌표값을 이용하여 항공기가 나아갈 방향을 찾아주는 역할을 한다. 또한 항공기가 VOR 범위에 들어왔는지의 여부와 들어왔을 경우의 진행 방향, Cone of Silence에서 항공기가 VOR 기지국을 통과했는지의 여부를 판단한다. Course Datum(CD)값은 VOR 범위 내에서 좌표계의 정북 방향( $X_E$ 축)을 기준으로 VOR을 향해 항공기가 진행해야 할 값이다. VOR의 접근 방향은

To/From Blockset을 통하여 찾는다.

$R_{VOR}$ 은 항공기와 VOR 기지국 사이의 거리를 나타내며, 일반적으로 Distance Measurement Equipment(DME)를 통하여 구한다.

##### 4.2 VOR data Blockset

<Fig. 5>의 VOR data Blockset에는 항공기의 항로점인 VOR 기지국의 3축 좌표가 저장되어 있다. 좌표는 표의 형태로 저장되어 있으며, 각각 지나온 기지국과 찾아가려는 기지국으로 나뉘어져 있다. 항공기가 VOR 전파 범위 내의 Cone of Silence 범위 내에 있고, VOR 기지국을 통과했다는 의미인 From 상태가 모두 만족되는 경우 출력 좌표가 다음 기지국의 좌표로 갱신된다. 또한 기하학적인 방법을 통하여 항공기와 VOR 기지국 사이의 각을 계산하여, 항공기가 다음 기지국으로 진행하기 위한 기수의 방향이 된다.

##### 4.3 VOR Reference Correct Blockset

항공기가 VOR 모드 상태에서 이동하는 동안 다음과 같은 순서로 진로 데이터가 변하게 된다.

- ① HH 모드로 진행하던 항공기가 VOR 기지국의 레이다 범위 내로 진입하는 순간의 CD값을 사용한다.
- ② 항공기가 Cone of Silence 내에서 진행할 때는 ①에서 얻은 CD값을 그대로 사용한다.
- ③ 항공기가 기존의 VOR 기지국을 통과하여 다음 VOR 기지국으로 진행하게 되면 두 VOR 기지국 사이의 방위각을 나타내는 NHH 값을 사용한다.
- ④ 마지막 VOR 기지국을 통과한 항공기는 초기

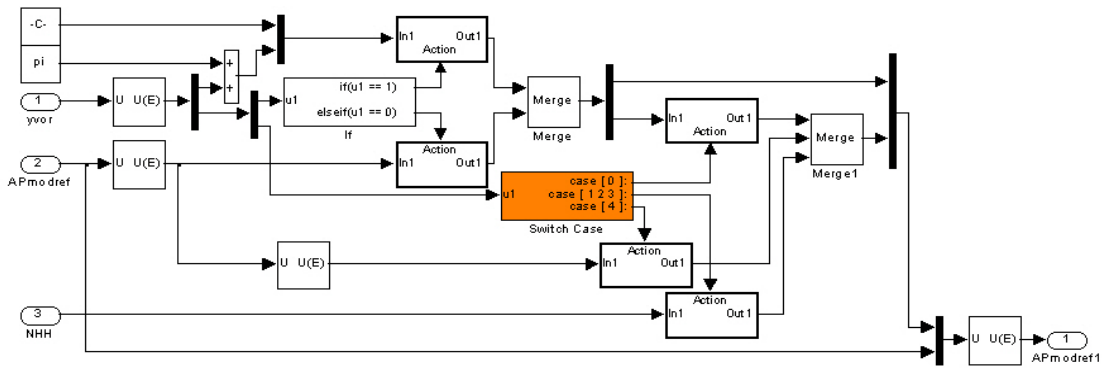


Fig. 6 출력 화면 구성

에 지정된 HH값으로 진행한다.

위 조건들을 구현하기 위하여 <Fig. 6>에서 각 조건에 대하여 Autopilot Mode Reference값을 재설정 하도록 하였다.

### V. 시뮬레이터 시험

#### 5.1 출력 화면 구성

<Fig. 7>에서 볼 수 있듯이 출력화면은 크게 입력부와 출력부로 나누어져 있다. 입력부는 앞에서 언급한 스위치 형식과 직접 입력 방식으로 되어 있는 모드 선택 패널로 구성하였다. 또한 출력부는 실제 Beaver와 동일한 계기 모습으로

나타내었고, VOR 모드에서 기지국 순번, 범위 진입 여부, Cone of Silence 진입 여부, To/From 상태를 확인할 수 있도록 구성하였다.

출력부 계기들은 Global Majic사에서 제작한 각종 항공 계기와 게이지를 활용하였다[6]. 이는 SIMULINK 상에서 Blockset 형식으로 제공이 되며 Active X를 기반으로 제작되어 있다.

항공기의 위치는 2가지의 Graph로 표현되며 지구고정 좌표계를 기준으로 한 x-y graph와, 시간에 따른 고도의 변화를 나타내는 t-H graph로 구성하였다.

#### 5.2 각 모드별 Autopilot 시험

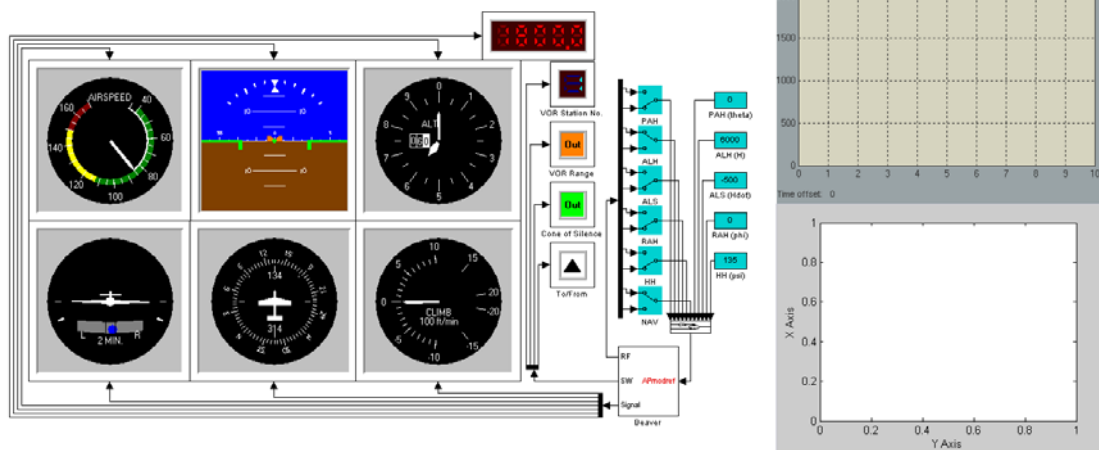


Fig. 7 VOR Reference Correct Blockset

Beaver의 Trim값은 6000ft에서 45m/s의 속도로 순항할 때를 기준으로 한 값을 사용하였다. 그러므로 이러한 조건이 만족되지 못할 경우에는 항공기가 편향되거나 불안정한 모습을 보이게 된다. 이를 보상하기 위하여 4000ft, 45m/s에서의 Trim값을 활용하여 다른 고도에서의 Trim값을 선형보간 하여 사용하였다.

5.2.1 ALH 모드 시험

<Fig. 8>은 아래 세 가지에 대한 결과이다.

- ① ALH 5500ft 인 경우
- ② ALH 5500ft & RAH 0° 인 경우
- ③ ALH 5500ft & HH 000인 경우

모드의 조합에 따라 약간씩의 차이를 보이지만, 약 60초 후에는 목표했던 5500ft에 도달했다.

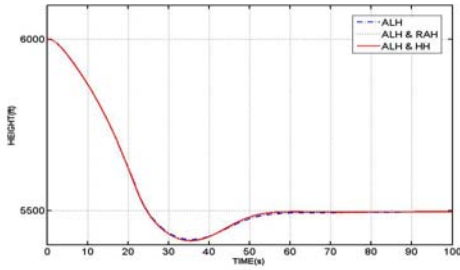


Fig. 8 ALH 모드 시험

5.2.2 ALS 모드 시험

<Fig. 9>는 아래 세 가지에 대한 결과이다.

- ① ALS -500ft/min
- ② ALS -500ft/min & RAH 0°
- ③ ALS -500ft/min & HH 000

<Fig 9>에서 200초를 전후하여 -500ft/min의 속도로 하강률이 일정해 지는 것을 확인하였다. 초기 항공기의 하강률이 0ft/min이 아닌 87.83ft/min인 것을 확인할 수 있다. 이는 항공기 하강률에 대한 초기 설정값  $H_0$ 에 의한 영향이다.

5.2.3 RAH 모드 시험

<Fig. 10, 11>은 아래 세 가지에 대한 결과이다.

- ① RAH 15°
- ② RAH 15° & PAH 0°
- ③ RAH 15° & ALH 6000ft

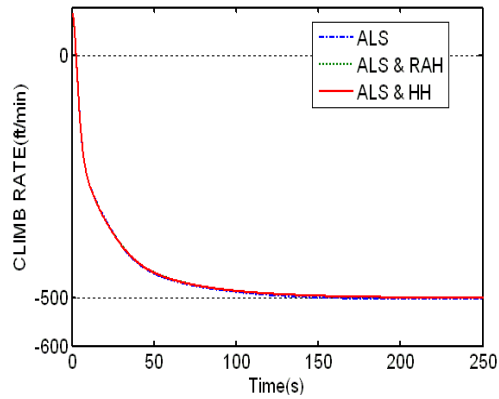


Fig. 9 ALS 모드 시험

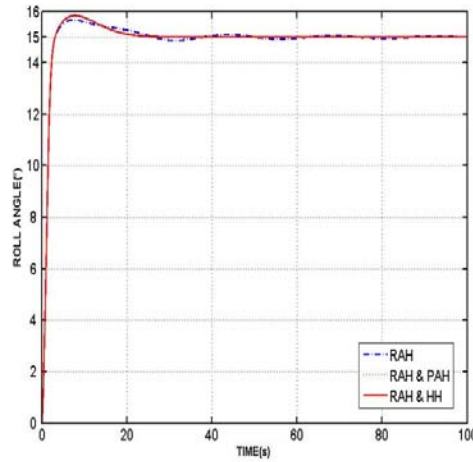


Fig. 10 RAH 모드 시험 - Roll Angle

Autopilot 모드에서의 Roll 각은 20°로 제한되어있으므로 RAH 시험에서는 15°로 입력하였다. 모든 모드에 대하여 Roll값이 동일한 것은 <Fig. 10>을 통하여 확인할 수 있다. 그러나 선형보간으로 얻은 Trim값은 정확한 값이 아니므로 동일 Roll값이지만 항공기의 고도가 일정하지 못하여 선회 반경이 모두 다르게 나타났다. 특히 RAH 모드만 실행했을 경우 다른 2가지 시험과 비교해서 시간에 따른 고도의 하강과 항공기의 진동이 심하다.

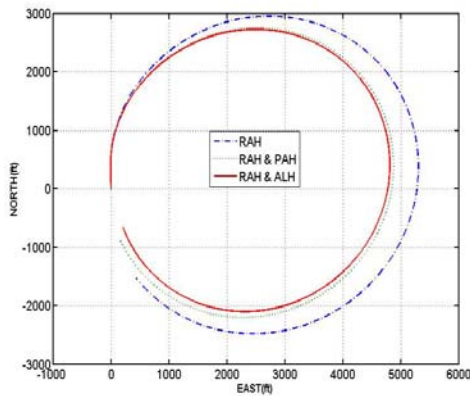


Fig. 11 RAH 모드 시험

### 5.2.4 HH 모드 시험

<Fig. 12>는 아래 세 가지에 대한 결과이다.

- ① HH 045
- ② HH 045 & PAH 0°
- ③ HH 045 & ALH 6000ft

세 경우 모두 045로 진행한다. 그러나 HH만 작동했을 경우 다른 모드에 비해 불안정하다.

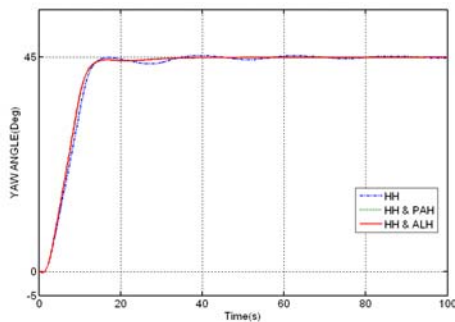


Fig. 12 HH 모드 시험 - Yaw Angle

### 5.2.5 NAV(VOR) 모드 시험

항공기의 초기 조건은 고도  $H=6000\text{ft}$ 에서 엔진의 분당 회전수는  $1800\text{rpm}$ , Manifold 압력은  $21.072\text{ inHg}$ , 초기 속도는  $45\text{m/s}$ ,  $\alpha = 0.144$ ,  $\beta = -0.014$ ,  $p, q, r = 0$ ,  $\psi, \phi = 0$ ,  $\theta = 0.144$ ,  $x_e, y_e = 0$ 으로 하였다. 또한 Simulation이 진행하

는 동안 고도를 초기  $6000\text{ft}$ 로 고정하였고, HH를 135로 지정하였다.

ALH, NAV 모드를 작동시킨 후 Autopilot을 작동시키면 초기 HH값을 따라 135 방향으로 진행을 한다. 첫 번째 VOR station의 전과 범위 내에 들어오게 되면 네 번째 station까지 스스로 찾아가게 되고, 마지막 Station을 통과하면 다시 135 방향으로 비행한다. <Fig. 13>은 Navigation 모드를 시뮬레이션 하였을 때 항공기의 이동경로를 나타낸 것이고, <Table 2>는 VOR Station 좌표를 나타낸 것이다.

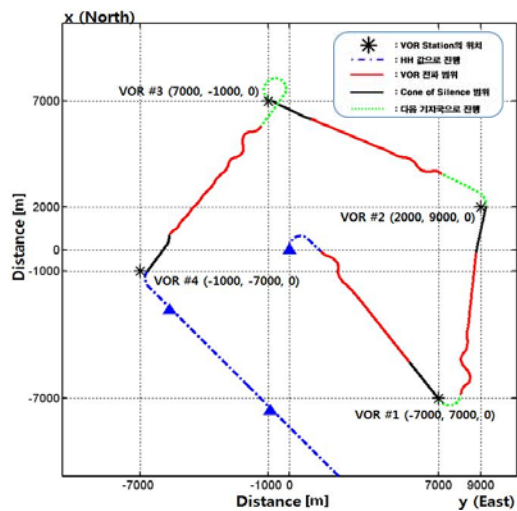


Fig. 13 NAV(VOR) 모드 시험

Table 2. VOR Station의 좌표

VOR station No.	좌표 [m]		
	$x_e$	$y_e$	$H$
1	-7000	7000	0
2	2000	9000	0
3	7000	-1000	0
4	-1000	-7000	0

## VI. 결 론

본 논문에서는 SIMULINK를 이용하여 소형 항공기용 자동 비행 시스템을 구현하고 시뮬레이션을 수행하였다. 구현된 출력 화면은 실제 항공기와 동일한 형태로 항공기의 비행 상황을 확인할

수 있다. 또한 각각의 모드에서 요구하는 제어 기능을 수행하는 것을 확인할 수 있었다.

그러나 단일 모드로 사용하였을 경우 부정확한 Trim값 등에 의해 정확한 제어를 수행하지 못하는 경우가 있었고, 항공기의 불안정이나 원거리 선회 등 개선해야할 문제점을 확인할 수 있었다. 차후 이러한 문제점을 보완하고 이·착륙에 대한 제어 알고리즘을 추가하는 연구가 필요하다.

### 참고문헌

- [1] Yoon, S. J., Simulation and Simulator, 선학사, Mar. 2003, pp.27.
- [2] Mathworks, SIMULINK 6 Using SIMULINK, Online version, 2007
- [3] Marc Rauw, 2005, FDC 1.4 - A SIMULINK Toolbox for Flight Dynamics and Control Analysis Draft Version7, May 25 2005 pp.302
- [4] Marc Rauw, 1993, FDC 1.2 - A SIMULINK environment for Flight Dynamics and Control analysis-application to the DHC-2 'Beaver' Part I & II, Sep. 1993
- [5] Robert C. Nelson, 1998, Flight Stability and Automatic Control, McGraw-Hill, Second edition, 1998, pp.97-105
- [6] <http://www.globalmajic.com/>
- [7] Bernard Etkin, Lloyd D. Reid, Dynamics of Flight Stability and Control, 3rd edition, 1996
- [8] Sang-won Chae, 3D Flight Simulator for Education of Flying Tactics, 2003