

Flight Simulator에서의 조종간 조작에 따른 비행자세 구현 방법

한용식, 김희대, 이철기

성균관대학교 전기전자 및 컴퓨터공학과 시뮬레이션 연구실

An Implementation Control Stick and Flying Attitude in Flight Simulator

Yong-sik Han, Hee-dae Kim, Chilgee Lee

Dept. of electrical and computer engineering, Simulation Laboratory
Sungkyunkwan University

요 약

비행 Simulator는 조종사가 실제 비행할 때와 같은 느낌을 받도록 구성되어야 하는데 조종사가 비행을 느끼게 되는 것은 여러 가지 감각기관을 통한 종합적인 느낌이다. 시각을 위하여 계기가 현재 비행 상태를 표시해야되고 외부 경치에서 고도, 속도, 자세 및 상승률을 감지하여야 한다. 이런 외부 상태를 표시하기 위해서는 조종사의 Stick조작에 의한 비행기의 위치 변화와 속도 등을 실시간으로 계산하여 Graphic Engine과 Actuator로 데이터를 전송하여야 한다.

이러한 Parameter를 결정하기 위해서는 조종사의 조작에 의한stick의 위치와 그로 인한 항공기의 자세 그리고 상태와 비행 속도 등에 따라 결정되며, 대상 항공기에 대한 계원으로 부터 조종사가 조종간의 이동크기와 조종면의 각 변위와의 관계를 계산하여 위치, 방향과 높이에 관한 Parameter를 계산해 줄 수 있는 방법 구현하였다.

1. 서론

시뮬레이터에는 많은 종류의 형태가 있다. 그중 비행 시뮬레이터는 컴퓨터가 장족의 발전하기 전에도 부분적인 시뮬레이터가 사용되었지만 지금은 시뮬레이터 탑승시간의 일부를 항공법규상 비행시간으로 인정하는 수준까지 시뮬레이터가 발전하였다. 미 공군의 한 보고서에서 보면 시뮬레

이터를 이용하여 훈련시 실제 항공기만 가지고 훈련할 때 보다 상당한 예산 절감이 된다는 것을 볼 수 있다. 이는 비행의 무 경험자를 실제 비행과 비슷한 상황에서 비행을 경험하고 비행훈련을 하면 그만큼 훈련 시간이 줄어든다는 것을 나타낸다. 아울러 (악천후, 계기고장 등) 비정상 상태에 대한 경험은 그보다도 더욱 더 효과적 가치를

갖고 있다. 이런 수준의 시뮬레이터를 구현하는데 있어서 비행 시뮬레이션은 과거 10년 동안 복잡한 계산의 증가에 따라 시뮬레이션 호스트 컴퓨터의 계산량도 점점 증가하게 됐다. 이로 인해 컴퓨터의 실시간 처리를 위해 여러 개의 프로세서를 사용하여 각기 다른 일을 처리하게 하는 방법을 사용하였다. 이 때문에 시스템의 가격상승을 불러왔고 재정적으로 어려운 곳에서는 프로세서 하나 또는 두개 정도를 사용하여야 하기 때문에 실시간 처리가 불가능하였다. 한두 개의 적은 프로세서나 컴퓨터에서 실시간 처리를 위해서는 최대한 연산처리부분을 줄이는 방법을 사용하여 하는데 모델을 최소화 시켜도 프로세서 자체 큰 부담을 가지게 된다. 이를 위해서 모델의 최소화와 비선형 부분은 항공기 제원으로 부터 계산된 Look-Up table 의 Parameter를 사용하여 처리 함으로 계산량을 줄이는 방법을 사용하였으며 실시간 처리를 가능하도록 하여 조종간의 조작으로부터의 신호를 테이블에서 검색하여 그에 맞는 데이터를 모션방정식에 의해 계산하여 비행 자세, 위치 등을 계산해주게 된다. 이를 위해 실시간 시뮬레이션에 대한 기본적인 개념과 그를 만족시킬 수 있는 비행 시뮬레이터의 기본적인 수학적 모델에 비행 역학적인 data와 조종간의 탄성 방정식을 이용하여 최종적인 비행 자세와 위치에 대한 데이터를 얻게된다.

2. 비행 시뮬레이션의 기본구조

비행시뮬레이션의 기본구조로는 시뮬레이션을 위한 항공기 시스템 모델링과 모델을 구현하기 위한 하드웨어적인 장비가 필요하며 앞의 두 요소에서 특정 개체를 만족하는 응용 기술의 적용이 필요하다.

비행 시뮬레이션의 기본모델은 비행환경모델, 항공기 모델 과 지형 환경 모델로 구성되어진다.

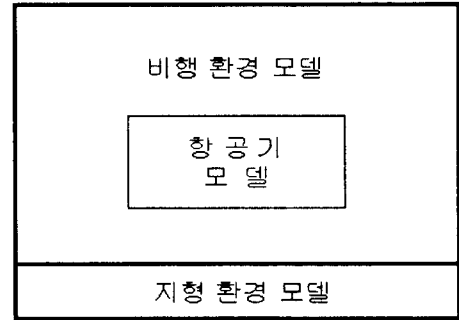


그림 1. 비행 시뮬레이션의 3가지 이론적 모델

위의 세 가지 요소 모델들은 새롭게 표현되는 시뮬레이션을 만들어 내기 위해서 분리 인식되어 작업을 해야한다는 것은 명백한 이치이다. 예를 들어 비행기 모델은 자세, Angle of attack과 높이 등이 계산되어진다. 이 정보는 지상을 볼 수 있는 각도를 결정하기 위해 날씨 조건을 따지는 모델링과 함께 비행환경 모델링에 사용되어질 것이다. 이 데이터는 비행기로부터 보여지는 지형의 모습을 보기 위한 지상 모델링과 결합되어진다.

3. 비행시뮬레이터에서의 수학적 모델

- 기초 개념

비행 중인 항공기에 작용하는 각종 힘들을 정확하게 계산하고, 이를 바탕으로 항공기의 복잡한 운동 방정식을 해결함으로써 항공기 위치 및 운동상태를 신속하게 결정 할 수 있는 능력은 물론 시뮬레이션 결과의 정도에 중요한 영향을 미치게 된다.

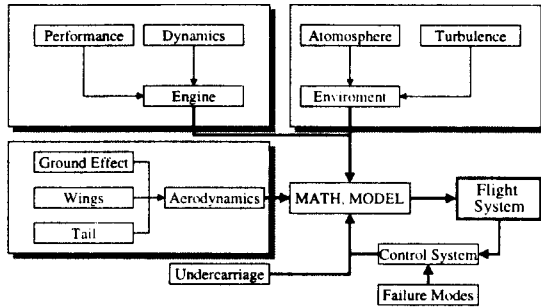


그림 1. Flight Simulation의 수학적 모델 구성도

시뮬레이션 결과를 얻기 위해서는 항공기의 운동상태를 수학적으로 표현한 모델과 대기 모델, 항공기 설계 제원, 성능 실험 자료, 항공기계의 동적 성능 자료, 지형 지물 자료가 절대적으로 필요하다. 항공기 운동에 관한 수학적 모형 이외의 자료들은 대부분 Simulation 대상 항공기에 대한 실험으로써 값을 구하거나 비행 특성Data Base 또는 경험에 의존하게 되므로 여기서는 Simulation에 필요한 수학적 모형에 대하여 설명하였다.

항공기 운동을 결정하는 요소들은 다섯 가지로 요약 할 수 있다. 1차 적으로 항공기의 공기 역학적 특성과 엔진특성, Tire와 Brake, Anti-Skid System등을 포함한 하부구조특성을 고려하여야 하며, 2차 적으로 대기 조건과 조종 장치로부터 항공기 조종면에 이르는 Control System이 항공기 운동을 결정하는 중요한 요인이 된다.

- 모션의 일반적인 방정식

항공기가 이동시 시뮬레이터에서는 프레임을 가지게 된다. 이때 각 프레임마다 항공기는 한 좌표계에서 다른 좌표계로 이동하게 되며 이를 Euler Angle로 나타내면 다음과 같은 Matrix로 나타낸다..

$$T = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \quad (1)$$

$$l_1 = \cos \theta \cos \phi$$

$$l_2 = \cos \theta \sin \phi$$

$$l_3 = -\sin \theta$$

$$m_1 = \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi$$

$$m_2 = \cos \phi \sin \theta \sin \psi + \cos \phi \cos \psi$$

$$m_3 = \sin \phi \cos \theta$$

$$n_1 = \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi$$

$$n_2 = \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi$$

$$n_3 = \cos \phi \cos \theta$$

ϕ : x축을 중심으로 회전한 각

ψ : z축을 중심으로 회전한 각

θ : y축을 중심으로 회전한 각

각각의 위상의 한계 범위는 다음과 같이 가정하였다.

$$-90^\circ \leq \phi \leq +90^\circ$$

$$-80^\circ \leq \theta \leq +80^\circ$$

$$-90^\circ \leq \psi \leq +90^\circ$$

Body Axes 상의 X, Y, Z방향의 속도 u, v, w는 다음과 같다.

$$u_1 = V_T \cos \alpha \cos \beta \quad (2)$$

$$v_1 = V_T \sin \beta \quad (3)$$

$$w_1 = V_T \sin \alpha \cos \beta \quad (4)$$

$\alpha = \phi$: 주변에서 일어나는 외력을 무시할 경우

$\beta = \psi$: 외부에서 작용하는 힘을 고려치 않을 경우 $\beta=0$ 이 된다. (Rudder를 비행기를 좌우로 이동할 경우는 0이 아님)

Stick의 상하 조작에 의해 α 각이 변함에 따라 항공기의 방향이 아래위로 운동을 하게되며, 좌우운동으로 ϕ 각의 변함으로 Roll운동을 하게되며, 수평을 기준으로 좌로 회전할 경우 $+\phi$ 값으로 증가하고 우로 회전할 경우 $-\phi$ 값으로 증가하게 된다. 동체가 수평 비행 시는 α 각의 변화는

일정하지만 Roll운동에 의해 동체가 좌나 우로 기울어졌을 때는 정상상태의 α 값에 의한 w_1 과 v_1 값이 변하게된다. α 값의 변화는 θ 에 의해 변하게되며, Roll운동은 ϕ 각에 의해 변하게 된다. 이를 식으로 나타내면 다음과 같다.

$$u_2 = V_T \cos \alpha \cos \beta \quad (5)$$

$$v_2 = w_1 \sin \phi \quad (6)$$

$$w_2 = w_1 \cos \phi \quad (7)$$

이 Matrix를 Body Axes상의 각 가속도인 식 (5), (6), (7)과 곱하여 항공기의 위치를 계산할 수 있다.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = T \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9)$$

식 (9)에 의해서 항공기의 매 시간 이동 변위에 대한 위치의 기본적인 값을 얻을 수 있다.

이러한 식의 운동방정식을 적분하여 항공기의 운동을 얻는 과정을 흐름도로 블록다이어그램으로 작성하면 다음과 같이 나타낼 수 있다.

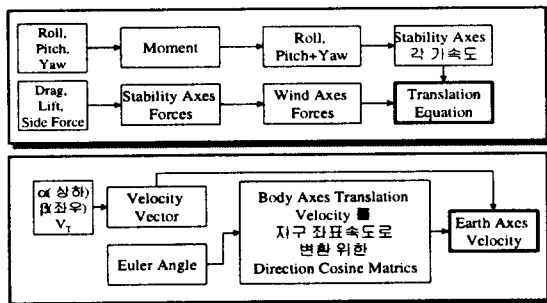


그림 2. Simulation Motion방정식의 Flow Diagram

- 조종간과 항공기간의 수학적 모델

조종에 필요한 Stick 이나 Throttle은 조작시 탄성력을 가지고 있는데, 탄성식은 정밀한 항공기로 고려하였을 때 분석모델에서는 생략될 수 있다. 이 부분을 이것은 간단하게 수학적인 공식화할 수 있으면 이로 인해 모델의 복잡성을 감소시킬 수 있다.

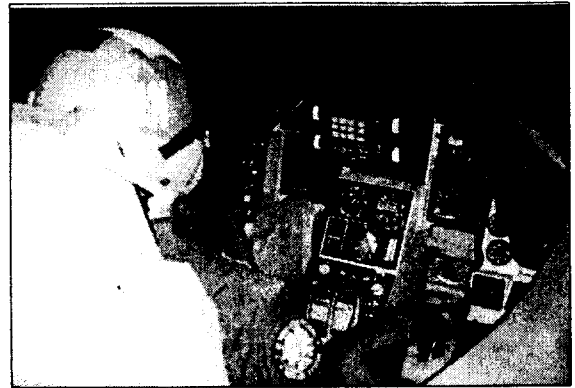


그림 4. Flight Simulator의 내부 구조

여기서는 항공기의 비행시 4가지 주요 요소들의 제어가 사용되어 진다.

Ailerons: ailerons는 날개의 끝 쪽에 위치했으며 항공기를 좌우로 회전시키는 역할을 한다.

Elevator : Elevator는 날개의 끝 쪽에 위치하였으며 항공기의 앞부분의 아래위로 움직이게 한다.

Rudder : 항공기의 수평꼬리날개에 있으며 항공기각을 좌우로 이동시키는 역할을 한다.

각 Control의 각도의 조작 범위는 다음 그림 3. 과 같이 나타난다.

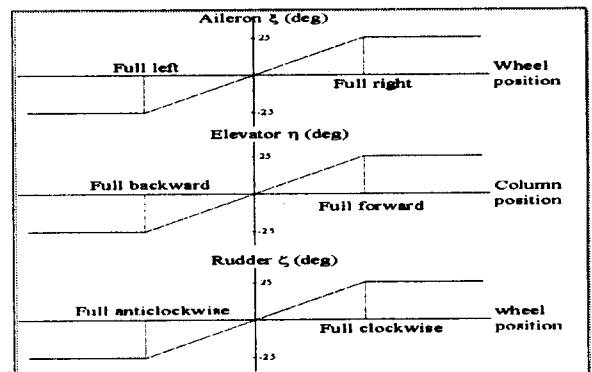


그림 5 Control 각의 조작 범위

위의 3가지 는 각각 조작 범위를 가지고 있으며 그 값은 다음과 같다.

$$-25^\circ \leq \xi, \eta, \zeta \leq +25^\circ$$

ξ : aileron control angle

η : elevator control angle

ζ : rudder control angle

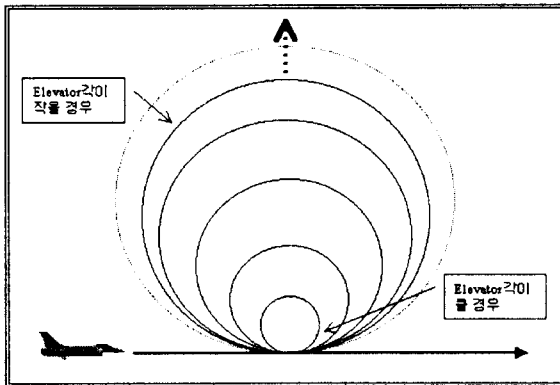


그림 6. Rudder각의 크기에 따른 Angle of attack의 변화

항공기는 비행 중에 Elevator의 각에 따라 항공기 원 운동을 하기 위한 충분한 속도를 가지고 있다고 가정하면 원운동을 하며 이로 인해 Angle of attack에 영향을 주게 되며, Elevator의 각의 크기에 따라 회전반경이 커지게 되고 Angle of attack의 변화가 작아지게 된다. 이를 이용하여 항공기 상승에 관련된 값을 얻을 수 있게 된다. 항공기의 특성에 따라 같은 각의 elevator라도 항공기의 특성에 따라 angle of attack이 달라진다.

- Representation of aerodynamic data

비행에 대한 어떤 영향이 전체적인 힘의 표시인지 미세한 작용에 의한 힘인지는 모델링의 형태에 따라 달라진다. 그래서 항공기의 공기역학적인 데이터는 여러 가지방법으로 표현되어진다. 미세하고 많은 변화를 가지고 있는 모델을 위해서는 motion식의 계수는 일반적인 안정된 비행을 위한 유도체일 것이고, 숫자들의 간단한 table은 비행상태의 항공기 비행 형태를 충분히 표시해줄 수 있다. 여기에는 6개의 유도힘과 15개의 유도모멘트가 있고 전체적으로 21개의 데이터의 아이템을 가지고 상당히 정밀한 비행 시뮬레이션 모델을 구현해 낼 수 있다.

힘과 모멘트는 다음에서 나타내는 하나 또는 그 이상의 작용으로 표현되어 진다.

- Angle of attack,
- Control deflection,
- Speed/Mach number,
- Rotation rates,
- Hight,
- Center of gravity position,
- Ground proximity, geometry 등 이 있

다.

Stick의 상하 조작에 의해 항공기는 Angle of attack의 변화를 가져오게 된다. 이는 뉴튼의 작용 반작용 법칙에 의해서 날개를 들어올리는 함으로 작용한다. 이와 기체와 날개의 형태에 따라 같은 각의 Elevator라도 상승 속도가 다르게 된다. 이를 수식적으로 계산할 경우 매우 많은 계산시간이 필요하므로 적은 시스템에서는 실시간으로 비행 시뮬레이션을 구현하는데 어려움이 있다. 이를 보완하기 위해 실험이나 항공역학 계산에 의한 Aerodynamic data를 사용하여 그 값을 table로 만들어 전진속도에 따른 상승 속도를 구할 수 있다.

이때 식 (4)에서의 $\sin \alpha$ 는 비행방향 속도에 대한 상승 속도이므로 실제 상승 비행 속도와는 차이를 가지게 되므로 실험 데이터를 사용하게 된다. 이 상승률 C를 사용하면 다음과 같이 된다.

$$w_1 = V_T C \cos \alpha \cos \beta \quad (10)$$

상승률

$$C = \tan \alpha \quad (11)$$

그리고 Flap의 각에 따른 상승 속도가 다르게 나타난다.

표 1. Flap 각과 Angle of attack각에 의한 상승률 Table의 예

Sectional lift coefficient						
Flap position (deg)	Angle of attack(deg)					
	0	5	10	15	20	25
0	-0.042	0.417	0.834	1.084	1.209	1.334
1	0.042	0.5	0.959	1.293	1.418	1.501
10	0.125	0.625	1.168	1.626	1.793	1.835
20	0.334	0.917	1.478	1.835	2.007	1.918
25	0.584	1.084	1.626	2.043	2.168	2.001
30	0.834	1.376	1.876	2.252	2.335	2.127

표 3 Transfer matrix에 의한 상승률 값

Sectional lift coefficient						
Flap position (deg)	Angle of attack(deg)					
	0	5	10	15	20	25
0	0	0.087	0.176	0.267	0.364	0.466

4. 결론 및 향후 계획

본 논문에서는 조종간 조작에 따른 엘리베이터의 움직임으로 변화해 가는 Flap각과 Angle of attack에 대해 항공기의 상승계수를 사용하였으며, 각각의 Angle of attack을 실험치로 구한 비행 상승 속도를 비행시뮬레이터에 적용함으로써 연산에 필요한 계산량이 줄어들 수 있고, 정밀한 시스템의 비행시뮬레이터를 2차원의 간단한 Table검색으로 항공기의 정밀한 모션을 실시간으로 표현해줄 수 있었다. 추후에는 비행상태의 시뮬레이션에서 기상 조건이 항공기에 미치는 비행 자세에 대한 연구와 함께 착륙과 이륙시의 발생할 수 있는 상황 즉 Undercarriage와 Ground effect를 등을 고려하여 완벽한 비행시뮬레이터를 구현해줄 계획이다.

참고문헌

[1] J. M. Rolfe, K. J. Staples, Flight Simulation, Cambridge aerospace series, 1997

[2] Hanke, C. R. & Nordwall, D. R. The Simulation of a large Jet Transport Aircraft Vol.1, Mathematical Model, Vol. 2, Modelling Data NASA CR 1756

[3] Hassan B. Diab, Design and Implementation of a Flight Simulation System, American University of Beirut, 1992

[4] 한국항공우주연구소, Development of a R&D Flight Simulator(4), 과학기술처, 1989

[5] Sinnett, Michael K., Alternate table look-up routine for real-time digital flight simulation, Journal of Aircraft, 1990, pp471-473

[6] Rolfe, J. M., and Staples, K. J., Representation of Aerodynamic Data, Flight simulation, Cambridge Univ. Press, Cambridge, England, 1986

[7] Barnes, A. G. , Modelling requirements in flight simulation, Aeronautical Journal, 1994, Vol. 98, pp.395-404