J. Korean Soc. Aeronaut. Space Sci. 50(12), 857-866(2022)
DOI:https://doi.org/10.5139/JKSAS.2022.50.12.857
ISSN 1225-1348(print), 2287-6871(online)

고정익 UAV 모델링 및 비행조종컴퓨터 기반 오토파일럿 통합 시뮬레이션 HILS 환경 구축

김남수[1], 이동우[2], 이호형[3], 홍수운[4], 방효충[5]

Modelling of Fixed Wing UAV and Flight Control Computer Based Autopilot System Development for Integrated Simulation HILS Environment

Lamsu Kim[1], Dongwoo Lee[2], Hohyeong Lee[3], Suwoon Hong[4] and Hyochoong Bang[5]
Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea[1,2,3,5]
Hanwha System, Seongnam-si, Republic of Korea[4]

ABSTRACT

Fixed-wing UAVs have long endurance and range capabilities compared to other aerial platforms. These advantages led fixed-wing UAVs to become a popular platform for reconnaissance missions in the military. In this research, we modeled fixed-wing UAVs, including the landing gear model and developed a guidance and control system for flight control computers to construct a HILS environment. We also developed an autopilot system that includes automated take-off, cruise, and landing control for UAVs. We also retrived the Aerodynamic coefficients an UAV using Datcom and AVL software and used them for 6 degrees of freedom modeling. The Flight control computer calculates guidance commands using the Carrot chasing guidance law after distinguishing the condition of the UAV based on 16 pre-defined flight modes and calculates control inputs using Nonlinear Dynamic Inversion (NDI) control scheme. We used RTNngine to integrate the Simulink model and flight control computer for HILS environment formulation.

초 록

고정익 UAV는 다른 항공기 플랫폼보다 항속거리와 항속시간에서 큰 이점을 가진다. 이러한 이유로 군에서 정찰용으로 많이 사용된다. 본 연구에서는 랜딩기어를 포함한 고정익 UAV의 모델링을 실시하고, 비행조종컴퓨터에 사용될 유도 및 제어기 설계 및 HILS 환경 구축을 실시하였다. 또한 이륙, 순항, 착륙의 모든 과정을 자동으로 수행하는 오토파일럿 시스템을 제작하였다. 연구에 사용한 고정익 UAV를 Datcom 및 AVL 공력해석 소프트웨어를 사용하여 공력계수를 추출하고 6자유도 모델링을 실시하였다. 비행조종컴퓨터는 항공기의 16개의 비행모드를 분별하여 Carrot Chasing 기반 유도 명령을 생성하는 유도기와 Nonlinear Dynamic Inversion 기법을 사용한 제어기로 구성되어있다. SIMULINK를 사용하여 구현된 모델링과 비행조종컴퓨터는 RTNgine을 사용하여 HILS 환경을 제작하여 고정익 UAV의 통합 시뮬레이션 환경을 제작하였다.

Key Words : Carrot Chasing(캐롯체이싱), Nonlinear Dynamic Inversion(비선형동역학역변환), Flight Control Computer(비행조종컴퓨터), Hardware In the Loop Simulation(모의시험환경)

† Received : April 26, 2022 Revised : September 13, 2022 Accepted : September 21, 2022
[1] Graduate Student, [2] Graduate Student, [3] Graduate Student, [4] Senior Researcher, [5] Professor
[5] Corresponding author, E-mail : hcbang@kaist.ac.kr, ORCID 0000-00001-6016-8102
© 2022 The Korean Society for Aeronautical and Space Sciences

Ⅰ. 서 론

고정익 무인 항공기는 멀티로터와 헬리콥터와 같은 회전익 비행체가 등장하였음에도 불구하고 긴 항속거리와 항속시간, 비행속도, 높은 운용고도의 이점으로 인해 꾸준히 민간 및 군용분야에서 사용되어왔다. 이착륙을 위한 활주로가 필요하다는 단점이 존재하지만 넓은 영역에서 임무를 할 수 있다는 장점으로 현재에도 많은 분야에서 사용되고 있다.

고정익 항공기를 실제 제작하고 안전하게 운용하기 위해서는 시뮬레이션 환경에서 항공기의 모델과 FLCC에 사용될 유도 및 제어기를 제작하여 Hardware In the Loop Simulation(HILS)을 실시하는 과정이 필요하다. HILS는 실제 항공기에 사용되는 센서, 구동기, 비행조종컴퓨터등을 6자유도 시뮬레이터와 연동하여, 가상환경에서 계산되는 항공기의 상태변수로 각 구성품의 측정, 구동, 반응 및 계산이 정상적으로 이루어지는지 확인하는 실험이다. 이 HILS는 기체를 실제 비행하지 않고 모든 기체의 구성품들이 잘 작동하는지 확인하여 기체의 안정성을 실기체를 비행하기 전에 확인하는 방법이다.

[7]에서는 이러한 목적을 위해 HILS 환경을 구성하였다. [8]에서는 X-Plane 환경에서 HILS를 구성하고 모델기반 제어기를 적용하였다. [7]에서는 기계획된 항로점을 비행하는 HILS 시스템을 구성하고 실험하였다. 그러나 기체에 대한 모델링 내용은 포함하고 있지 않다. [8]의 참고문헌도 실기체에 대한 모델링을 포함하고 있지 않으며, 이륙이나 착륙에 대한 모델링을 고려하지 않았다. 또한 [9]에서도 소형제트기의 제어기에 대한 HILS 환경을 제작하였으나, 기체에 대한 모델링을 포함하지 않았으며, 제어기는 방위각 유지 제어기까지만 고려하였다.

본 연구에서는 한화시스템(주)에서 요청한 고정익 UAV의 공력계수를 추출하고, 6자유도 모델링을 실시하였다. 또한 실제 비행조종컴퓨터에 장입할 유도 및 제어기 API를 제작하고, REALTIMEWAVE社의 RTNgine을 사용하여 기타 센서, 지상국, 시뮬레이션 환경과의 통합 HILS 환경을 구성하여 가상 시뮬레이션 환경을 구축하였다. 제어기는 이륙, 순항, 로이터, 착륙 모든 과정에 대한 경로점 비행을 적용하였다. 6자유도 모델 및 항법센서의 구현은 시뮬링크를 사용하였고, 비행조종컴퓨터(FLCC)에 장입할 유도 및 제어기는 C 코드로 작성하였다. FLCC 하드웨어 장비는 한화시스템에서 제작한 장비를 사용하였다.

본 연구의 기여도는 실기체 공력계수 추출, 랜딩기어를 포함한 6자유도 시뮬레이션 환경을 제작, 비행조종시스템, 시뮬링크, 임무장입기를 사용한 HILS 구성, 경로점 정보를 주면 기체의 상태정보를 사용하여 비행모드를 설정하고 이륙, 순항, 착륙의 모든 과정을 자동으로 수행하는 비행조종시스템 기반 오토파일럿 시스템을 제작, 다양한 통신규격의 장비를 사용가능한 통합 HILS 환경을 구성한 것이다.

모델링에 사용한 기체는 MQ1 Predator와 비슷한 급의 기체로 Inverted-Tail 형상을 가진다. 본 기체의 공력 계수는 [1]을 참고하여 Datcom과 AVL를 사용하여 추출하였다. Inverted V-Tail를 가지는 고정익 기체는 Datcom에서 모델링이 불가하여 AVL 프로그램을 병행 사용하였다. 제어에는 Nonlinear Dynamic Inversion 기법을 사용하였다. 유도기법은 단계에 따라 16개로 나누어진 비행모드(flight mode)를 항공기의 속도, 고도, 랜딩기어 지면 접지 상태, 임무장입기에서 전달하는 경로점 행렬을 통해 결정하고, 알맞은 속도, 고도, 헤딩, 지면에서의 헤딩명령을 계산한다. 기본 유도기법은 Carrot Chasing 기법[2]을 사용하였다.

Ⅱ. 문제 정의

2.1 항공기 모델링

고정익 항공기는 일반적으로 6자유도 움직임으로 모사한다. 6자유도 움직임을 모사하기 위한 상태미분방정식은 수식 (1)에서 (4)까지 나타내었다.

$$
\begin{aligned}
\dot{V}_T &= \frac{1}{m}(-D + F_T c_\alpha c_\beta + m g_1) \\
\dot{\alpha} &= q - (p c_\alpha + r s_\alpha) t_\beta + \frac{1}{m V_T c_\beta}(-L - F_T s_\alpha + m g_2) \\
\dot{\beta} &= p s_\alpha - r c_\alpha + \frac{1}{m V_T}(Y - F_T c_\alpha s_\beta + m g_3) \\
g_1 &= g(-c_\alpha c_\beta s_\theta + s_\beta c_\theta s_\phi + s_\alpha c_\beta c_\theta c_\phi) \\
g_2 &= g(c_\alpha c_\theta c_\phi + s_\alpha s_\theta) \\
g_3 &= g(c_\beta c_\theta s_\phi + s_\beta c_\alpha s_\theta - s_\alpha s_\beta c_\theta c_\phi)
\end{aligned}
\tag{1}
$$

식 (1)은 기체의 속도 V_T, 받음각 α, 옆미끄럼각 β에 관한 상태미분방정식으로, m은 기체 중량, D는 항력, L은 양력, Y는 횡력, F_T는 추력을 나타낸다. 수식에서 $c_\bullet$, $s_\bullet$, $t_\bullet$는 각각 $\cos(\bullet)$, $\sin(\bullet)$, $\tan(\bullet)$을 의미한다. g는 중력 가속도 상수이다.

$$
\begin{aligned}
\dot{\boldsymbol{\omega}} &= -\boldsymbol{I}^{-1}(\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega}) + \boldsymbol{I}^{-1}(\boldsymbol{M}_{a_{trim}} + \boldsymbol{M}_{a_{act}}) \\
\dot{\boldsymbol{v}}_b &= -\boldsymbol{\omega} \times \boldsymbol{v}_b + (\boldsymbol{F}_a + \boldsymbol{F}_g + \boldsymbol{F}_t + \boldsymbol{F}_s)/m \\
\boldsymbol{I} &= \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix}
\end{aligned}
\tag{2}
$$

수식 (2)의 $\boldsymbol{w} = [p, q, r]^T$ 와 $\boldsymbol{v}_b = [u, v, w]^T$는 각각 관성좌표계기준 동체좌표계의 롤, 피치, 요축 속도 및 회전율을 동체기준좌표계에서 나타낸 상태변수이다. I는 관성모멘트행렬을 의미한다. $\boldsymbol{M}_{ext} = \boldsymbol{M}_{a_{trim}} + \boldsymbol{M}_{a_{act}}$는 각각 기체에 작용하는 총 모멘트, 트림 공력 모멘트, 조종면 공력 모멘트 항을 의미한다. $\boldsymbol{F}_{ext} = \boldsymbol{F}_a + \boldsymbol{F}_g + \boldsymbol{F}_t + \boldsymbol{F}_s$는 각각 총 외력, 공력, 중력, 추력, 지면에서 작용하는 외력을 의미한다.

$$\boldsymbol{M}_{a_{trim}} = \begin{bmatrix} M^l_{a_{trim}} \\ M^m_{a_{trim}} \\ M^n_{a_{trim}} \end{bmatrix} = \begin{bmatrix} QSb(c_{l_\beta}\beta + \frac{b}{2V_T}(c_{l_p}p + c_{l_r}r)) \\ QSc(c_{m_\alpha}\alpha + \frac{c}{2V_T}c_{m_q}q) \\ QSb(c_{n_\beta}\beta + \frac{b}{2V_T}(c_{n_p}p + c_{n_r}r)) \end{bmatrix} \quad (3)$$

$$\boldsymbol{M}_{a_{act}} = \begin{bmatrix} M^l_{a_{act}} \\ M^m_{a_{act}} \\ M^n_{a_{act}} \end{bmatrix} = \begin{bmatrix} QSb(c_{l\delta_a}\delta_a + c_{l\delta_r}\delta_r) \\ QSc(c_{m\delta_e}\delta_e) \\ QSb(c_{n\delta_a}\delta_a + c_{n\delta_r}\delta_r) \end{bmatrix}$$

수식 (3)에서 Q는 동압, S는 날개 면적, b는 날개 길이, c는 주익의 코드 길이를 나타내고, $c_{l_\cdot}$, $c_{m_\cdot}$, $c_{n_\cdot}$는 $\cdot$의 변수에 선형인 롤축 공력 모멘트, 공력 피치축 모멘트, 공력 요축 모멘트 유차원 미계수를 의미한다. 수식 (4)의 $\dot{\boldsymbol{x}} = [v_N, v_E, v_D]^T$는 NED 좌표계에서의 속도, $\dot{\boldsymbol{\zeta}} = [\dot{\phi}, \dot{\theta}, \dot{\psi}]^T$은 관성좌표계에서의 오일러각 회전율로 각각 롤각, 피치각, 헤딩각의 회전율을 의미한다. R_B^I은 동체좌표계에서 관성좌표계로의 변환행렬, C_B^I는 회전율의 동체좌표계에서 관성좌표계로의 변환행렬을 의미한다.

$$\dot{\boldsymbol{x}} = R_B^I \boldsymbol{v_b}$$

$$\dot{\boldsymbol{\zeta}} = C_B^I \boldsymbol{\omega}$$

$$R_B^I = \begin{bmatrix} c_\psi c_\theta & -c_\psi s_\psi - c_\psi s_\phi s_\theta & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\phi s_\theta & -c_\psi s_\phi - c_\phi s_\psi s_\theta \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \quad (4)$$

$$C_B^I = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}$$

$$\dot{\lambda} = \dot{x}_n / (R_n + h)$$

$$\dot{\Omega} = \dot{x}_e / ((R_e + h) * c_\lambda$$

$$R_n = \frac{a(1-e^2)}{(1-(e\,s_\lambda)^2)^{1.5}} \quad (5)$$

$$R_e = \frac{a}{(1-(e\,s_\lambda)^2)^{0.5}}$$

수식 (5)는 위도와 경도의 상태미분방정식이다. 본식의 λ와 Ω은 위도 경도를 의미한다. a는 지구의 장반경 상수이고, e는 지구의 이심율 상수이다.

2.1.1 공력계수 추출

6자유도 움직임을 모사하기 위해 필요한 항공기의 공력계수는 Datcom과 AVL 프로그램을 이용하였다. Datcom은 미공군에서 경험식을 기준으로 항공기의 공력계수를 계산하는 프로그램이다[3]. Datcom의 장점은 간단히 공력계산이 가능하고, 실속 속도 및 받음각에 대한 계산을 한다는 것이다. 또한 참고문헌을 통해 B747-200 기체의 실제 공력계수와 Datcom에서의 공력계수를 비교했을 때 큰 차이가 없음을 보여 일반적인 고정익 항공기에서는 계산값이 실제로도 적절한 해석값을 보여준다[10]. 단점으로는 주익과 수평, 수직 미익을 가지는 재래식항공기가 아닌 형태의 항공기는 계산이 부정확하거나 해석이 안되는 경우가 존재하고 또한 횡축 공력계수가 부정확하게 계산된다. 미공군에서 제공한 Digital Datcom 메뉴얼[3]을 보면 해석 가능한 형상에 대해 명시하고 있으며, 본 과제에서 사용한 Inverted V-tail 형상은 포함하지 않는다.

반대로 AVL은 다양한 기체 모델링이 가능하고, 다양한 모델에서의 횡축 공력계수 또한 계산 가능하다[4]. [11]에서는 AVL이 작은 받음각에서는 실제 공력계수와 큰차이를 보이지 않음을 보여주었다. 그러나 반대로 AVL에서는 실속 받음각에 대한 계산이 불가하고 작은 범위의 받음각 및 옆미끄럼각의 공력계수만 정확하게 계산된다는 단점을 가진다.

Ansys Fluent와 같은 CFD를 사용하여 본 형상과 같은 기체를 해석할 수 있다[12]. 그러나 CFD는 Mesh를 많이 구성할수록 해석 정확도가 커지는데, 그러면 컴퓨팅 파워가 매우 많이 소모되고 계산시간이 매우 길다는 단점을 가진다. 이러한 이유들로 본 연구에는 [1]에서와 같이 본과제의 UAV의 공력계수 추출에 Datcom과 AVL을 같이 사용하였다. Fig. 1은 Datcom과 AVL에 사용한 기체의 3차원 모델이다.

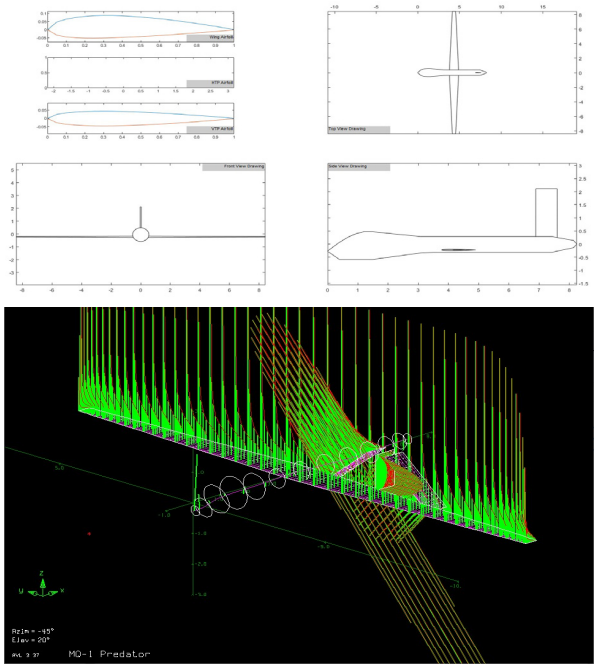

Fig. 1. Datcom (above) and AVL Modeling (bottom) of UAV with inverted V-tail

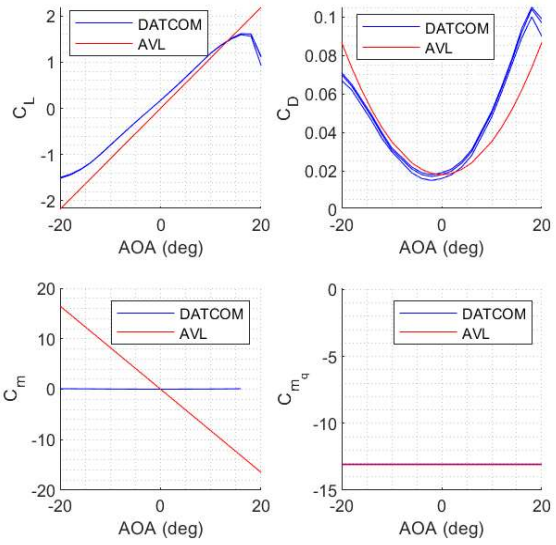

Fig. 2. Comparison of C_L, C_D, C_m and C_{m_q} aerodynamic coefficients versus angle of attack (AOA) of Datcom and AVL

Figure 2는 Datcom과 AVL에서 추출한 C_L, C_D, C_m, C_{m_q} 공력계수를 비교한 것이다. C_L과 C_D 계수는 두 프로그램이 비슷한 공력계수를 추정한 것을 볼 수 있으며, C_{m_q}는 매우 비슷한 값으로 계산됨을 확인할 수 있었다. 그러나 Datcom의 경우 실속 받음각에 대한 특성을 계산해주고, AVL은 계산하지 못하는 것을 볼 수 있다. 반대로 C_m 계수는 Datcom의 경우 전혀 계산하지 못한 것을 확인할 수 있으며, Inverted V-tail 형상에 대한 해석이 되지 않았음을 알 수 있다. 이렇게 Datcom으로 계산되지 못한 계수는 AVL이 추출한 공력계수를 사용하여 기체에 대한 모델링을 실시하였다.

Figure 2에서 Datcom과 AVL의 Inverted V-tail에 대한 양력 및 항력 값이 큰 차이를 보이지 않았기에 C_L와 C_D은 Inverted V-tail을 적용하지 않은 기체의 공력계수를 추출한 Datcom의 값을 사용하였고, 나머지 계수는 AVL에서 추출한 값을 사용하였다. 6자유도 항공기 모델링은 Simulink 프로그램을 사용하였다.

2.1.2 랜딩기어 모델링

랜딩기어모델링은 [6]을 참고하여 제작하였다. 지상에서 UAV의 운동을 모사하기 위해 랜딩기어 모델링을 실시하였다. 기체는 후미에 메인 랜딩기어 2개, 노즈에 1개의 랜딩기어가 장착되는 Tricycle 형태로 구성하였다. 각각의 랜딩기어의 서스펜션과 타이어는 감쇠계수와 스프링상수와 마찰계수를 설정하여 서스펜션과 타이어에 작용하는 힘을 모사하였다.

Figure 4는 랜딩기어에 작용하는 서스펜션의 힘 F_{susp}과 타이어에 작용하는 힘 F_{tire}을 보여준다. m_1은 기체의 중량, m_2는 타이어의 무게, k_1과 c_1은 서스펜션의 스프링 계수 및 감쇠 계수, z_g는 지면에서의 타이어 압축거리, z_w는 타이어와 타이어 중심휠과의 거리, z_a는 랜딩기어와 기체가 만나는 부분의 위치를 나타낸다.

2.2 비행조종컴퓨터 기반 유도 및 제어기

실 기체를 모사한 FLCC 기반 C 코드 유도 및 제어기 설계를 실시하였다. RTNgine, 임무장입기, 비행조종컴퓨터, 시뮬링크 모델에 대한 블록선도는 Fig. 3과 같이 구성하였다. 비행조종컴퓨터와 시뮬레이터는 1553B 인터페이스를 사용하여 데이터를 송수신한다. FLCC는 유도기와 제어기로 구성된다. 유도기는 경로점 행렬을 임무장입기를 통해 입력받아 헤딩명령 ψ_{cmd}, 속도 명령 V_{cmd}, 고도 명령 h_{cmd}를 계산하는 Guidance Command 블록과 16개의 비행모드(Flight mode)를 선택하는 Flight mode selector 블록으로 이루어져 있다. 제어기는 h_{cmd}을 기수각 명령 θ_{cmd} 으로, ψ_{cmd}을 롤각 명령 ϕ_{cmd}로 옆미끄럼각 β를 0으로 만드는 요축 회전율 명령 r_c를 계산하는 Heading & Height Controller, ϕ_{cmd}와 θ_{cmd}를 동체좌표계의 롤축과 기수축 회전율 명령 p_{cmd}, q_{cmd}로 계산하는

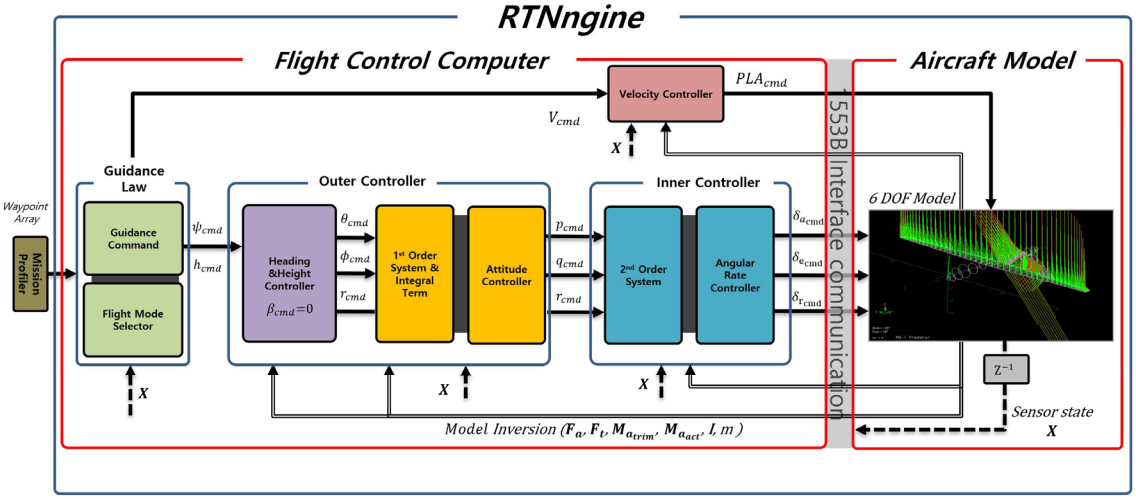

Fig. 3. Guidance and Control Block Diagram

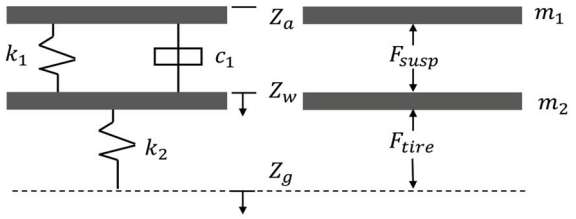

Fig. 4 . Suspension and Tire Newtonian dynamics

Outer Controller, p_{cmd}, q_{cmd}, r_{cmd}를 에일러론, 엘리베이터, 러더 변위 명령 δ_a, δ_e, δ_r를 계산하는 Inner Controller, 속도 명령 V_{cmd}를 엔진의 파워 레버 각도(Power lever angle) 명령인 PLA_{cmd}로 계산하는 Velocity Controller로 이루어져 있다.

2.2.1 유도기법

유도 명령생성은 Fig. 3에서 Guidance Command와 Flight Mode Selector 블록으로 이루어진 Guidance Law 블록이 수행한다. Guidance Law 블록은 경로점 행렬와 항공기 상태변수를 입력으로 받는다. 한 개의 경로점은 8개의 정보를 가진 구조체로 다음과 Table 1과 같이 이루어져 있다.

Guidance Law 블록은 16개의 비행모드를 기준으로 유도명령을 생성하여 제어기에 전달한다. 비행모드는 Flight Mode Selector에서 결정한다. 16개의 비행모드는 Table 2에 나타나있다. Table 2에서 TO로 시작하는 비행모드는 이륙을, CZ는 순항을, LD는 착륙을, TX는 택싱에 관한 과정이다. TO와 TX 비행모드는 경로점의 flag 정보가 TOflag일 때 CZ 비행모드는 CZflag, LD 비행모드는 LDflag일 때 선택된다. 이륙은 활주로에서 이륙속도 획득하는 TO_GAINSPEED 과정, 기수를 천천히 드는 TO_ROTATION 과정, 적정 고도 및 속도를 획득하면 이륙을 종료하는 TO_TERMINATE 구간으로 나뉘어져 있다. 순항은 이륙을 종료한 지점에서 다음 경로점으로 이동하는 CZ_TOINITPOINT 모드와 직선 및 로이터 기동을 하는 CZ_NORMAL 모드가 있다. 착륙 시에는 LD_CRUISE부터 LD_TERMINATE까지 7개의 단계로 각각 UAV 착륙을 위해 활주로로부터 6,000m 떨어진 지점으로 이동하는 과정, 활주로 접근 과정, 고도 하강 과정, 플래어 기동 과정, 활주로 접지 과정, 제동 과정, 랜딩 종료 과정으로 이루어져 있다.

각 비행모드는 Guidance Command에 입력되어 모드에 알맞은 ψ_{cmd}, h_{cmd}, V_{cmd} 명령을 생성하고, 제어기에 전달한다. 경로점 flag 정보가 TOflag일 때는 TO_GAINSPEED부터 TO_TERMINATE까지 비행모드를 거치기 위해 내부적으로 이륙 활주로와 이륙 방향 헤딩을 기준으로 경로점을 생성한다. 착륙도 7개의 비행모드를 거치기 위해 활주로와 진입 헤딩각을 기준으로 6,000m 후방 지점, 활주로, 활주로 끝점의 경로점을 내부적으로 생성한다. 이륙 및 착륙 활주로와 경로점은 측지좌표계로 주어지며 유도 명령을 계산하기 위해서는 NED 좌표계에서의 좌표점이 필요하기 때문에 초기 경로점과 3개의 경로점마다 항공기 위치와 경로점을 NED 좌표계로 초기화한다. Guidance Command에서는 Carrot Chasing 유도기법을 기준으로 ψ_{cmd}를 생성하고, h_{cmd}와 V_{cmd}는 경로점 행렬에 입력된 값을 사용한다. Carrot Chasing은 Fig. 5와 같이 2개의 경로점이 이루는 경로와 한 경로점을 기준으로 선회하는 로이터 기동에 대한 ψ_{cmd}을 계산한다. 유도명령은 참고문헌 [3]의 Carrot Chasing 기법을 사용하였고, 본 유도기법의 직선경로와 로이터 기동에 대한 ψ_{cmd} 계산식은 수식 (6)과 같다.

Table 1. Waypoint Array

Element	Description
lat	latitude command
lon	longitude command
alt	altitude command
Vt	Velocity command
Radius	Radius of trajectory for loiter mode
cw	counter/clock wise command for loiter mode
loiterTime	time command to loiter for current waypoint
flag	waypoint mode: TOflag (take off), LDflag (landing) and CZflag(cruise)

Table 2. Flight mode for guidance command

Flight Mode	Description	No.
TO_GAINSPEED	Take Off : gain speed	1
TO_ROTATION	Take Off : nose up	2
TO_TERMINATE	Take Off : end	3
TO_ERROR	Take Off : error	7
CZ_TOINITPOINT	Cruise : From Take off end point to next waypoint	8
CZ_NORMAL	Cruise : normal cruise	9
LD_CRUISE	Landing : current UAV location to landing approach point	10
LD_APPROACH	Landing : Landing approach point to runway	11
LD_DESCEND	Landing : descend	12
LD_FLARE	Landing : flare	13
LD_TOUCHDOWN	Landing : landing gear touch down	14
LD_BRAKE	Landing : brake	15
LD_TERMINATE	Landing : end	16
TX_BRAKE	Taxiing : brake	19
TX_TAXIING	Taxxing : normal taxiing	20

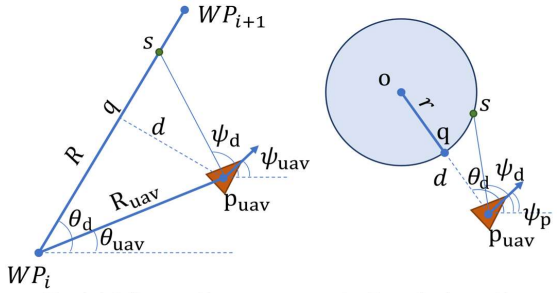

Fig. 5. Carrot Chasing Guidance Geometry

$$\beta = \theta_d - \theta_{uav}$$
$$R = \sqrt{R_{uav}^2 - (R_{uav} s_\beta)^2}$$
$$y_{target} = (R+\delta) s_{\theta_d}, x_{target} = (R+\delta) c_{\theta_d}$$
$$\psi_d = \text{atan2}(y_{target} - y_{uav}, x_{target} - x_{uav}) \quad (6)$$
$$y_{target} = r s_{\theta_d + \lambda}$$
$$x_{target} = r c_{\theta_d + \lambda}$$
$$\psi_d = \text{atan2}(y_{target} - y_{uav}, x_{target} - x_{uav})$$

유도명령 수식 (6)에서 δ와 λ는 각 기동에 대한 조절 변수이다.

2.2.2 제어기법

Guidance Law 블록에서 전달되는 V_{cmd}, ψ_{cmd}, h_{cmd}은 Fig. 3에 속도제어기(Velocity Controller), 외부제어기(Outer Controller), 내부제어기(Inner Controller)에서는 유도기에서 생성한 고도, 헤딩, 속도 명령을 엔진의 파워 레버 각도(PLA)와 조종면의 변위 명령으로 계산하는 역할을 한다. 각 제어기는 Slotine[5]의 Nonlinear Dynamic Inversion(NDI)기법을 사용하였고, 계단식(Cascade)으로 연결하였다. 아래와 같은 시스템이 있다고 하자. 여기서 $x \in R^n$, $b \in R^{n \times n_u}$, $u \in n_u$이다.

$$\dot{x} = f(x) + b(x)u \quad (7)$$

Slotine이 [5]에서 설명한 NDI 제어기법은 수식 (7)과 같은 시스템에 아래와 같은 제어입력을 사용하였다.

$$u = b^{-1}(\nu - f(x)) \quad (8)$$

위 제어입력을 수식 (7)에 적용하면,

$$\dot{x} = \nu \quad (9)$$

와 같은 선형 시스템을 얻을 수 있다. 위 수식을 사용하여 원하는 시스템으로 반응하도록 ν를 설계하여 제어기를 구성할 수 있다. NDI기법을 사용하여 속도와 외부 및 내부제어기를 설계하였다.

속도제어기는 수식 (7)과 같이 구성하였다. 수식 (7)은 수식 (1)에의 $\dot{V}_T$ 방정식을 $1/\tau_{V_T}(V_{cmd} - V_T)$로 변경하고 $F_{T_{cmd}}$의 식을 계산한 후 필요 일률(power)을 계산한 후 P_{eng}^{MAX} 엔진의 최대 일률 출력으로 나누어서 PLA_{cmd}를 계산하였다.

$$F_{T_{cmd}} = \frac{1}{\tau_{V_T}}(V_{cmd} - V_T) + k_{I_{V_T}} \int (V_{cmd} - V_T) dt + \frac{1}{c_\alpha}(D - mg s_{\theta-\alpha}) \quad (10)$$
$$PLA_{cmd} = \frac{F_{T_{cmd}} V_T}{P_{eng}^{MAX}}$$

$$\dot{\psi}_{cmd} = \frac{1}{\tau_{hdg}}(\psi_{cmd} - \psi)$$
$$\phi_{cmd} = \frac{V_T}{g}\dot{\psi}_{cmd}$$
$$r_{cmd} = \frac{1}{\tau_\beta}(\beta - \beta_{cmd}) + p s_\alpha + \frac{1}{m V_T}(Y - F_t c_\alpha s_\beta + mg_3) \quad (11)$$
$$\dot{h}_{cmd} = \frac{1}{\tau_h}(h_{cmd} - h)$$
$$\gamma_{cmd} = \sin\left(\frac{\dot{h}_{cmd}}{V_T}\right) \approx \frac{\dot{h}_{cmd}}{V_T}$$
$$\theta_{cmd} = \alpha + \gamma_{cmd}$$

외부제어기의 고도 및 헤딩제어기에서는 수식 (8)과 같이 ϕ_{cmd}, θ_{cmd}, r_{cmd}를 먼저 생성한다. ϕ_{cmd}와 θ_{cmd}은 Fig. 6의 UAV 종축 및 후면 기하학을 참고하여 계산하였다. 후면 기하에서 ψ_{cmd}를 1차 식 $\dot{\psi}_{cmd} = 1/\tau_\psi(\psi_{cmd} - \psi)$로 변경하고, $m V_T \dot{\psi} c_\phi = mg s_\phi$임을 이용하여 ϕ_{cmd}를 생성하였다. h_{cmd}는 Fig. 6의 종축 기하학을 참고하여 생성하였다. 1차 식을 사용하여 고도상승율 $\dot{h}_{cmd} = 1/\tau_h(h_{cmd} - h)$를 생성한 후 수식 (11)과 같이 비행경로각 γ_{cmd}로 변환, θ_{cmd}를 생성하였다. r_{cmd}는 수식 (1)의 $\dot{\beta}$ 방정식을 $1/\tau_\beta(\beta_{cmd} - \beta)$로 치환하여 생성하였다. β_{cmd}는 0으로 고정하였다.

외부제어기에서 생성한 ϕ_{cmd}와 θ_{cmd}는 수식 (3)의 $\dot{\phi}$와 $\dot{\theta}$의 상태미분식에서 $\dot{\phi} = 1/\tau_\phi(\phi_{cmd} - \phi)$로, $\dot{\theta} = 1/\tau_\theta(\theta_{cmd} - \theta)$로 변경하여 p_{cmd}와 q_{cmd}를 생성하였고, 그 수식은 (12)와 같다.

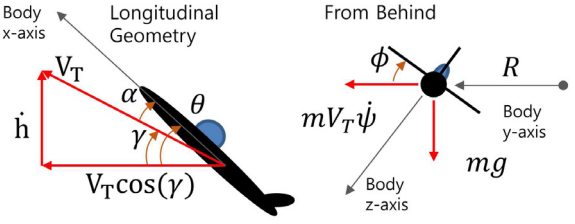

Fig. 6. Longitudinal(left) and rear(right) geometry

$$p_{cmd} = \frac{1}{\tau_\phi}(\phi_{cmd} - \phi) - t_\theta(q s_\phi - r c_\phi)$$
$$q_{cmd} = \frac{1}{\tau_\theta}(\theta_{cmd} - \theta) + r\, t_\phi \quad (12)$$

$$\dot{p}_{cmd} = 2\zeta_p \omega_{n_p} \int \frac{\omega_{n_p}}{2\zeta_p}(p_{cmd} - p)dt$$
$$\dot{q}_{cmd} = 2\zeta_q \omega_{n_q} \int \frac{\omega_{n_q}}{2\zeta_q}(q_{cmd} - q)dt \quad (13)$$
$$\dot{r}_{cmd} = 2\zeta_r \omega_{n_r} \int \frac{\omega_{n_p}}{2\zeta_p}(r_{cmd} - r)dt$$

외부제어기에서 계산한 p_{cmd}, q_{cmd}, r_{cmd}를 제어를 위한 항공기의 에일러론 $\delta_{a_{cmd}}$, 엘리베이터 $\delta_{e_{cmd}}$, 러더 $\delta_{r_{cmd}}$ 조종면 위치 명령을 계산한다. 내부제어기는 수식 (2)의 $\dot{\boldsymbol{w}} = [\dot{p}, \dot{q}, \dot{r}]^T$를 수식 (13)과 같은 2차 시스템으로 가정하고 제어 입력 δ_a, δ_e, δ_r의 식으로 정리하면 수식 (14)와 같이 계산된다.

$$\delta_a = \frac{1}{QSb\left(c_3 c_{l\delta_a} + c_4 c_{n\delta_a}\right)}\dot{p}_{cmd} + \delta_{a_{trim}}$$

$$\delta_e = \frac{2I_{yy}}{QSc\, c_{m\delta_e}}\dot{q}_{cmd} + \delta_{e_{trim}}$$

$$\delta_r = \frac{1}{QSb\left(c_4 c_{l\delta_r} + c_9 c_{n\delta_r}\right)}\dot{r}_{cmd} + \delta_{r_{trim}}$$

$$\delta_{a_{trim}} = -\left\{\frac{\begin{array}{l} c_3\left(c_{l_\beta}\beta + \frac{b}{2V_T}(c_{l_p}p + c_{l_r}r) + c_{l\delta_r}\delta_r\right) \\ + c_4\left(c_{n_\beta}\beta + \frac{b}{2V_T}(c_{n_p}p + c_{n_r}r) + c_{n\delta_r}\delta_r\right)\end{array}}{\left(c_3 c_{l\delta_a} + c_4 c_{n\delta_a}\right)}\right\} \quad (14)$$

$$\delta_{e_{trim}} = -\left\{\frac{1}{c_{m\delta_e}}\left(c_{m_\alpha}\alpha + \frac{c}{2V_T}c_{m_q}q\right)\right\}$$

$$\delta_{r_{trim}} = -\left\{\frac{\begin{array}{l} c_4\left(c_{l_\beta}\beta + \frac{b}{2V_T}(c_{l_p}p + c_{l_r}r) + c_{l\delta_r}\delta_a\right) \\ + c_9\left(c_{n_\beta}\beta + \frac{b}{2V_T}(c_{n_p}p + c_{n_r}r) + c_{n\delta_r}\delta_a\right)\end{array}}{\left(c_4 c_{l\delta_a} + c_9 c_{n\delta_a}\right)}\right\}$$

$$c_0 = I_{xx}I_{zz} - I_{xz}^2,\ c_3 = I_{zz}/c_0,\ c_4 = I_{xz}/c_0,\ c_9 = I_{xx}/c_0$$

유도 및 제어기에서 필요한 항공기의 상태변수, 공력 및 물성치는 MQ1 Predator 항공기의 6자유도 움직임을 모사한 시뮬링크 시뮬레이터와 1553B 인터페이스를 통해 전달되고, 유도 및 제어기 생성한 $\delta_{a_{cmd}}$, $\delta_{e_{cmd}}$, $\delta_{r_{cmd}}$ 또한 동일한 통신 규격을 사용하여 다시 시뮬레이터로 전달된다.

Ⅲ. 시뮬레이션

3.1 시뮬레이션

Figure 3과 같이 구성한 HILS 환경에서 Table 3과 같은 경로점 행렬을 임무장입기를 통해 입력하고 시뮬레이션을 진행하였다. Table 3에서 cw는 반·정시계방향을 정하는 값으로, 0은 직선 유도, 1은 정시계, -1은 반시계방향을 의미한다. UAV는 1번 경로점인 청주공항에서 이륙하고 6개의 경로점을 지나 최종적으로 다시 8번 경로점에 있는 청주공항에 착륙하는 시나리오로 구성되어 있다. 2, 3, 4, 5, 7번 경로점에서는 로이터 기동을, 6번 경로점은 직선 경로점으로 이루어져 있다. Fig. 7은 HILS 환경에서 경로점 비행을 수행한 UAV의 궤적이다. Fig. 7에서 빨간색 선은 경로점, 파란색 선은 실제 UAV의 궤적이다. 사용한 유도기법이 NED 좌표계에서의 위치를 기준으로 유도명령이 계산되기 때문에 위도 경도 고도(LLH) 좌표점을 NED 좌표계로 전환해야 한다. NED 좌표로 변환 후 좌표와 LLH 좌표점과의 오차는 NED 초기화 지점의 근접거리에서는 크지 않지만, 거리가 길어질수록 커지기 때문에 3개의 좌표점을 지날 때마다 NED 좌표점을 초기화하는 과정을 수행한다. UAV가 TOflag 및 LDflag의 경로점 명령을 수행할 때는 활주로와 활주로의 이착륙 헤딩을 기준으로 각각의 이착륙을 위한 경로점을 생성한다. TO terminate는 UAV가 이륙이 끝난 지점이고 비행모드는 3인 구간이다. LD approach 지점은 UAV가 착륙을 위해 활주로의 6,000m 후방 이동을 위한 좌표점이다.

Table 3. Test waypoint array

No.	lat. (deg.)	lon. (deg.)	alt (m)	Vt (m/s)
1	36.72091	127.49608	0	0
2	36.72632	127.50279	200	36
3	36.72362	127.50951	250	36
4	36.71551	127.50951	300	36
5	36.70289	127.51175	250	36
6	36.69568	127.49832	220	36
7	36.69568	127.48041	200	36
8	36.72091	127.49608	241	30
No.	Turn Radius (m)	cw (-1,0,1)	loiterTime (s)	flag
1	0	0	0	TOflag
2	100	1	100	CZflag
3	100	-1	100	CZflag
4	150	1	100	CZflag
5	100	1	100	CZflag
6	0	0	0	CZflag
7	150	-1	200	CZflag
8	0	0	0	LDflag

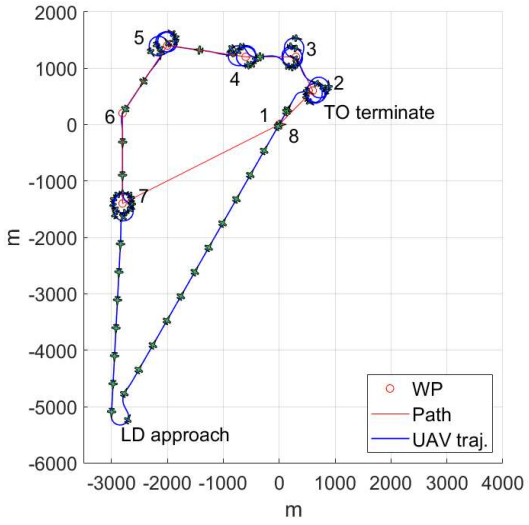

Fig. 7. Trajectory of UAV with given waypoint array

Figure 8은 Table 3의 경로점 행렬 명령에 따른 비행 모드와 경로점 번호의 변화를 보여준다. 기체는 최초 TX_BRAKE에서 시작한다. 이후 TO_GAINSPEED를 지나 TO_TERMINATE 지점까지 내부적으로 생성된 경로점을 기준으로 비행하고, 이후 경로점 2번부터 7번까지 이동하고, 8번 경로점에 근접하면, 착륙 과정을 수행한다. UAV 착륙과정은 내부적으로 좌표점을 생성된 지점을 기준으로 수행된다.

Figure 9는 제작한 HILS 환경과 Simulink 환경에서 제작한 모델의 헤딩명령 ψ_{cmd}, 고도 명령 h_{cmd}, 속도 명령 V_{cmd}, 동축 롤회전율 명령 p_{cmd}, 피치축 회전율 명령 q_{cmd}, 요축 회전율 명령 r_{cmd}, 파워레버각도 명령 PLA_{cmd}을 순서대로 비교하는 그래프이다. 비교결과 동일한 값을 출력하는 것을 확인하였다.

HILS 환경을 사용하여 Table 3으로 주어진 경로점 행렬을 따라가는 경로점 비행을 위해 유도기의 ψ_{cmd}, h_{cmd}, V_{cmd} 명령, 외부제어기 명령 ϕ_{cmd}, θ_{cmd}, β_{cmd}, 내부제어기 명령 p_{cmd}, q_{cmd}, r_{cmd}에 대한 반응은 Fig. 10에 나타나있다. 본 경로점 비행을 위해 생성한 유도 및 제어 명령에 UAV가 추종하는 것을 확인하였다.

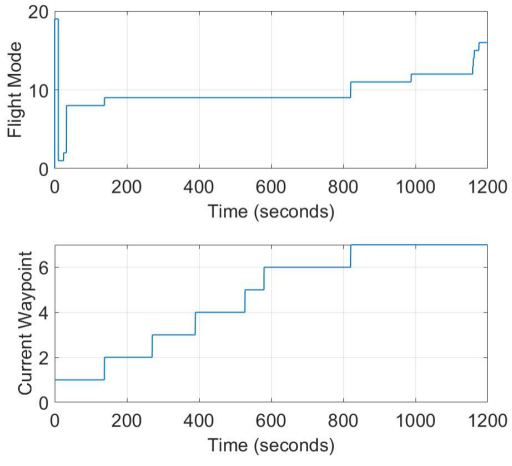

Fig. 8. Flight Mode (Up) and Current Waypoint No. (Down)

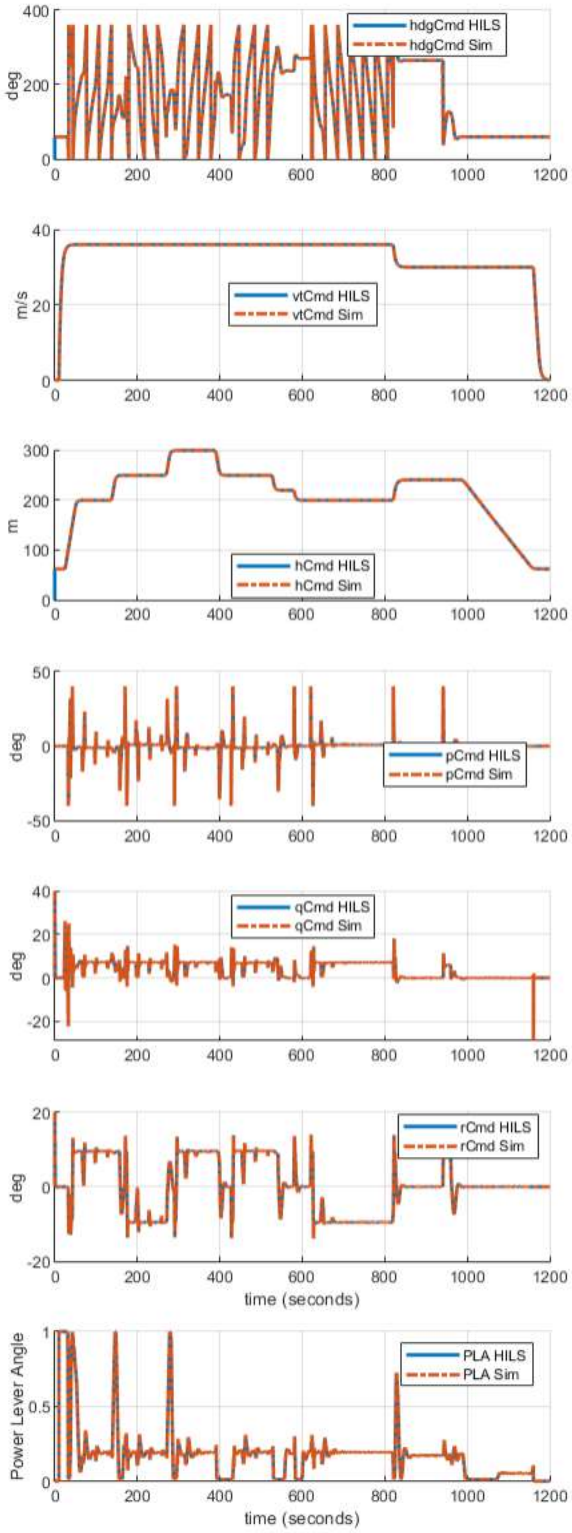

Fig. 9. Comparison between FLCC Command Output from HILS environment and Simulink

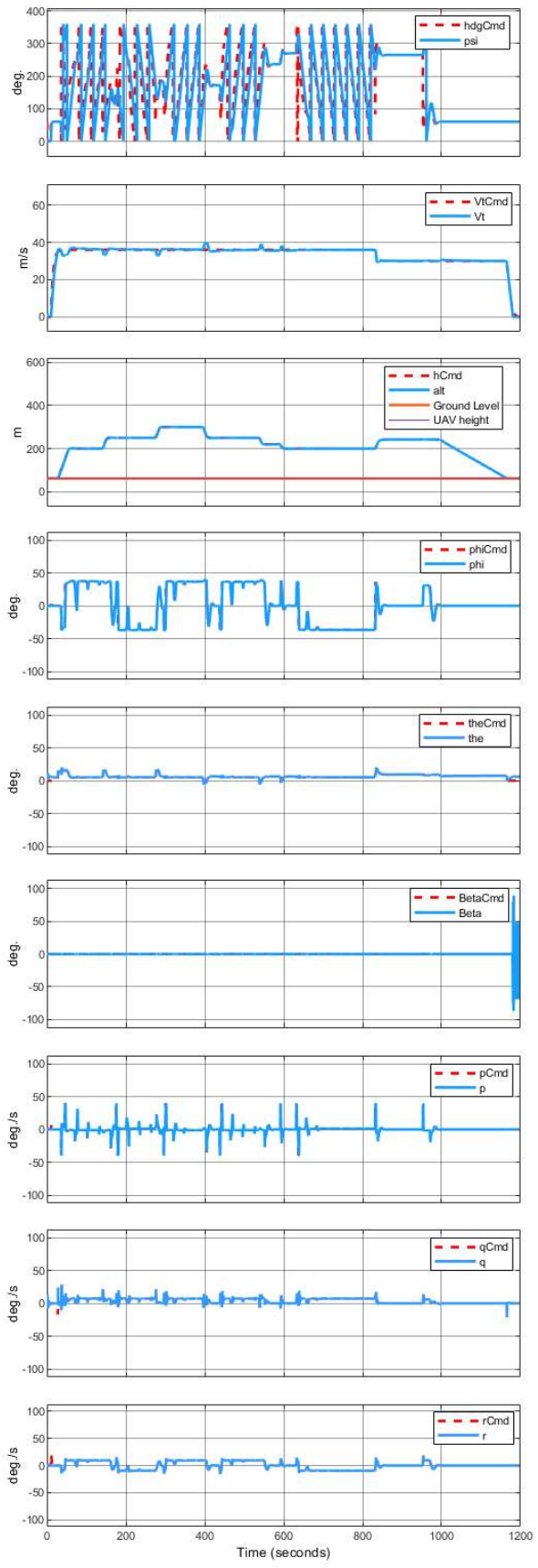

Fig. 10. Response to guidance and control input

Ⅳ. 결 론

고정익 UAV에 대한 HILS 환경을 제작하였다. FLCC에 장입될 유도기 및 제어기 API를 C 코드로 제작하였고, 6자유도 모델 및 항법센서 시뮬레이션 환경은 Simulink 소프트웨어를 사용하였다. 각 시스템을 통합하는 환경은 RTNgine을 사용하였다. 유도기 및 제어기법은 Carrot Chasing과 NDI 기법을 기준으로 제작하였고, 유도기는 16개의 비행모드를 기준으로 이·착륙시 내부 경로점 생성, 모드에 맞는 유도 명령을 생성하였다. 제작된 HILS 환경에서 시뮬레이션을 진행하였고, 결과를 확인하였다.

후 기

이 논문은 2020년도 한화시스템(주)의 재원을 지원받아 수행된 연구임

References

1) Frye, A. J. and Mehiel, E. M., “Modeling and simulation of vehicle performance in a UAV swarm using horizon simulation framework,” *AIAA SciTech 2019 Forum*, 2019, p. 1980.

2) Sujit, P. B., Saripalli, S. and Sousa, J. B., “Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial Vehicles,” *IEEE Control Systems Magazine*, Vol. 34, No. 1, 2014, pp. 42~59.

3) Williams, J. E. and Vukelich, S. R., “The USAF Stability And Control Digital Datcom, Volume I,” *USAF Technical Report AFFDL-TR-79-3032 (AD A086557) Users Manual*, April 1979.

4) Drela, M. and Youngren, H., “Athena Vortex Lattice (AVL) 3.35 User Guide,” *Massachusetts Institute of Technology Tech Report*, 2013.

5) Slotine, J. E. and Li, W., *Applied nonlinear control*, 1st Ed., Englewood Cliffs, Prentice hall, 1991 pp. 252 ~ 262.

6) Kim, L. S., Kim, S. J., Lee, D. W. and Bang, H. C., “Modeling of Landing Gears in 6 Degrees of Freedom Environment for Simulation of Fixed Wing Aircraft Landing Performance,” *Proceeding of The Korean Society for Aeronautical and Space Sciences Spring Conference*, July 2021, pp. 780~781.

7) Byun, J., Hur, G. B., Lee, K. H. and Suk, J., “A Study on UAV Flight Control System HILS Test Environment,” *Journal of the Korean Society for Aeronautical & Space Sciences*, Vol. 44, No. 4, 2016,

pp. 316~323.

8) Han, D. I., Kim, Y. S., Lee, C. Y., Lee, D. W. and Cho, K. R., "A Study on Verify of UAV Flight Control Software Simulated Flight using Model-Based Development and X-Plane simulator," *Journal of the Korean Society for Aeronautical & Space Sciences*, Vol. 43 No. 2, 2015, pp. 166~171.

9) Lee, J. H., Kim, E. T. and Seong, K. J., "HILS Test for the Small Aircraft Autopilot," *Aerospace Engineering and Technology*, Vol. 8 No. 1, 2009, pp. 172~178.

10) Ahmad, M., Hussain, Z. L., Shah, S. I. A. and Shams, T. A., "Estimation of Stability Parameters for Wide Body Aircraft Using Computational Techniques," *Applied Sciences*, Vol. 11 No. 5, 2021, p. 2087.

11) Ribeiro, F. C., de Paula, A., Scholz, D., Gil, R. and da Silva, A. "Wing geometric parameter studies of a box wing aircraft configuration for subsonic flight," *In 7th European Conference for Aeronautics and Space Sciences*, July 2017.

12) Nugroho, G., Zuliardiansyah, G. and Rasyiddin, A. A. "Performance Analysis of Empennage Configurations on a Surveillance and Monitoring Mission of a VTOL-Plane UAV Using a Computational Fluid Dynamics Simulation," *Aerospace*, Vol. 9, No. 4, 2022, p. 208.