

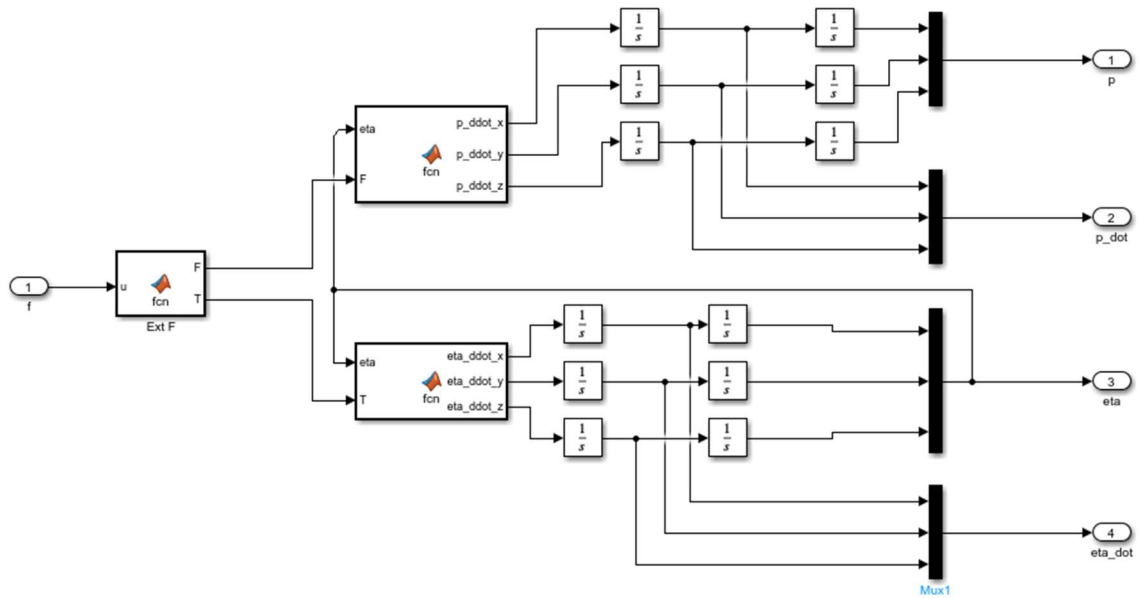
쿼드로터 위치 및 자세 제어 시뮬레이션

김명균 작성

제어기 구조 참조 - Lim, Jeongeun. "Autonomous target following and monitoring with collision avoidance based on an Lidar on a multi-copter"

쿼드로터의 동적 모델링을 기반으로 위치 및 자세를 제어하고자 한다.

앞서, 쿼드로터의 동역학 모델링을 MATLAB Simulink를 통해 표현하였고 쿼드로터의 무게를 이겨 내는 Throttle 입력을 통해 잘 작동함을 확인했다.



관성좌표계

-쿼드콥터의 위치: $p = [x \ y \ z]^T$

-쿼드콥터의 오일러 각: $\eta = [\phi \ \theta \ \varphi]^T$

바디좌표계

-쿼드콥터의 선속도: $v = [v_x \ v_y \ v_z]^T$

-쿼드콥터의 각속도: $\omega = [\omega_x \ \omega_y \ \omega_z]^T$

쿼드로터의 움직임은 관성 좌표계와 기체 좌표계에 대해서 다음과 같은 관계를 가진다.

$$\dot{p} = Rv, \quad \omega = C\dot{\eta}$$

$$\ddot{p} = \dot{R}v + R\dot{v}, \dot{\omega} = \dot{C}\dot{\eta} + C\ddot{\eta}$$

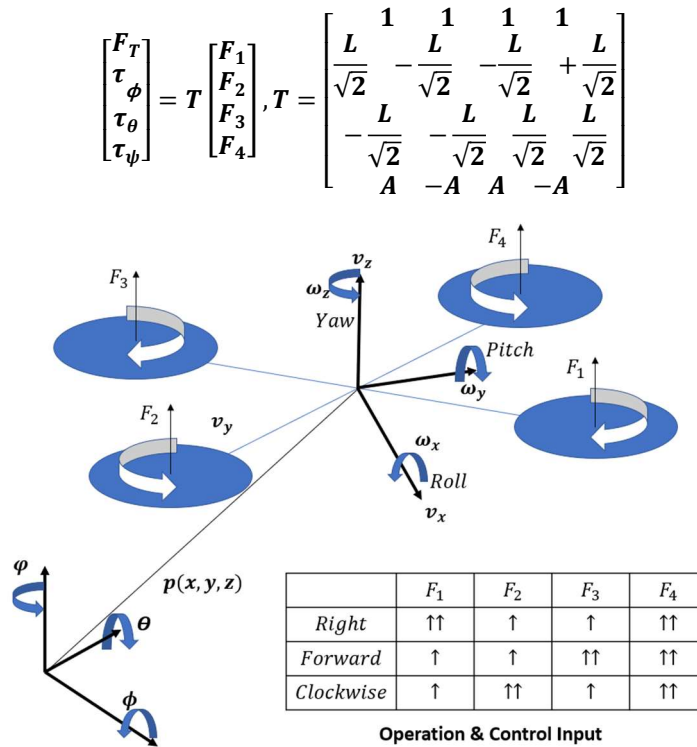
여기서는 식을 간단히 표현하기 위해 코리올리 효과는 무시한다.

쿼드로터의 움직임은 Force, Torque(roll, pitch, yaw) 명령에 의해 수행되고 기체 좌표계에서 입력된 힘을 모델링을 따라 관성 좌표계에서의 힘 및 토크로 변환하고 이를 적분하여 위치, 속도, 각도, 각속도로 반환했다.

$$p = [x \ y \ z], \quad \eta = [\phi \ \theta \ \varphi]$$

쿼드로터의 위치 및 자세를 제어하기 위해서 목적지의 위치, 목적 heading 값이 필요하여 이를 추종하기 위해 앞서 로봇으로부터 반환된 현재의 위치와 각도를 이용한다.

쿼드로터의 특성상 기체 좌표계를 기준으로 높이 방향(Z - axis)으로의 추력밖에 낼 수 없다는 것을 알 수 있고 X, Y - axis로의 움직임은 기체의 제어를 위한 힘과 모멘트와 4개 로터의 추력으로 표현될 수 있었다.



따라서 ϕ, θ 의 제어입력을 통해 X,Y축으로의 움직임을 가질 수 있으므로 둘 사이의 관계식은 다음과 같이 표현될 수 있다. (동적모델링에서 쿼드로터에 중력이 작용하고 있었다. 이를 보상해주기 위해서는 Z축방향으로 인가할 힘에 중력을 보상 해주어야 한다., $g=9.81$)

$$\ddot{x} = \dot{v}_z(\cos\phi\sin\theta\cos\varphi + \sin\phi\sin\varphi)$$

$$\ddot{y} = \dot{v}_z(\cos\phi\sin\theta\sin\varphi - \sin\phi\cos\varphi)$$

$$\ddot{z} = \dot{v}_z(\cos\phi\cos\theta) + g$$

$$\frac{\ddot{x}\sin\phi - \ddot{y}\cos\phi}{\dot{v}_z} = \sin\phi, \quad \frac{\ddot{x}\cos\phi + \ddot{y}\sin\phi}{\ddot{z}} = \tan\theta$$

Taylor expansion 활용해 0도 부근에서 다음과 같이 표현할 수 있다.

$$\sin\phi \approx \phi, \tan\theta \approx \theta$$

원하는 위치로 이동시키기 위해 목적 위치와 각도, 현재 위치와 각도의 Error가 제어입력이 될 것이다. 즉 위치 오차를 $(\ddot{x}, \ddot{y}, \ddot{z})$ 로 보겠다는 뜻이다. 그리고 유도된 수식을 통해 그때의 추종해야 할 각도를 구할 수 있고 현재의 각도와 Error를 통해 관성 좌표계에서의 롤 피치 요 입력 토크를 구할 수 있다.

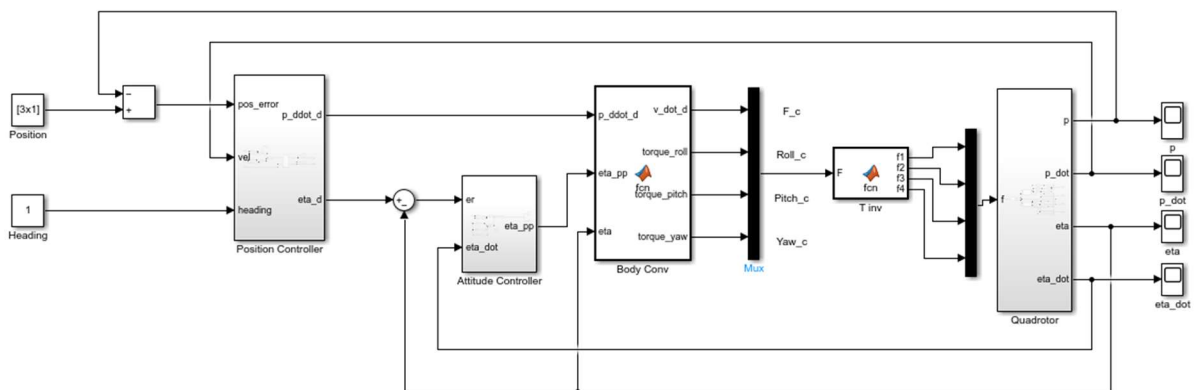
여기까지 관성 좌표계에서 쿼드로터의 힘(position Error), 토크(angle Error)를 구하였고 이는 기체 좌표계에서의 힘과 토크로 변환하여 각 로터의 제어입력을 구해야 한다.

$$\dot{v} = R^T \ddot{p}, \dot{\omega} = C \ddot{\eta}$$

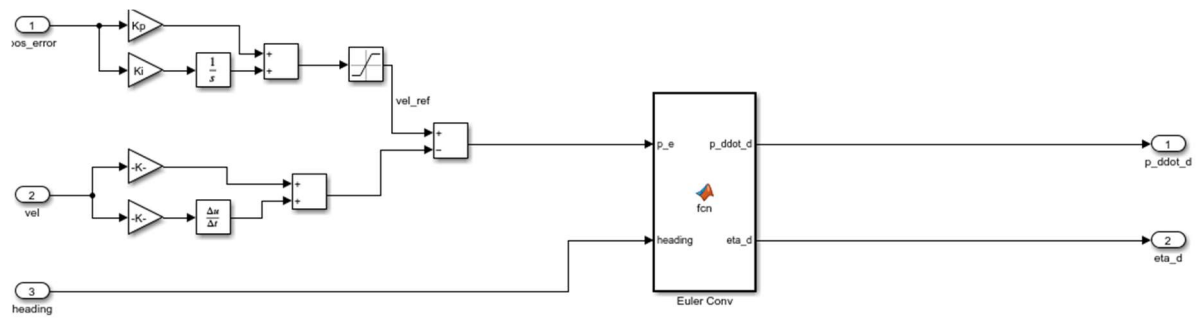
$$F = m\dot{v} = \begin{bmatrix} \ddot{z}\sin\theta \\ \ddot{z}\cos\theta\sin\phi \\ \ddot{z}\cos\theta\cos\phi \end{bmatrix}, \quad \tau = I\dot{\omega} = \begin{bmatrix} \ddot{\phi} - \ddot{\phi}\sin\theta \\ \ddot{\phi}\sin\phi\cos\theta + \ddot{\theta}\cos\phi \\ \ddot{\phi}\cos\phi\cos\theta - \ddot{\theta}\sin\phi \end{bmatrix}$$

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} = T^{-1} \begin{bmatrix} F_T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix}$$

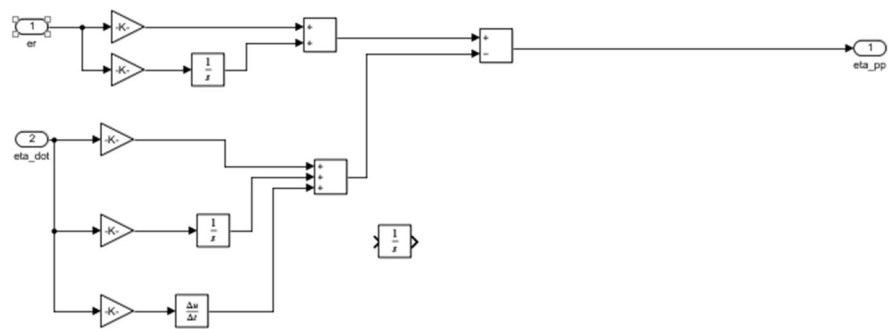
이를 MATLAB Simulink에서 구성하였다.



Position Controller



Attitude Controller



Gain

$$K_p=1, K_i=0$$

$$K_{rate_p} = 1, K_{rate_d} = 0.1$$

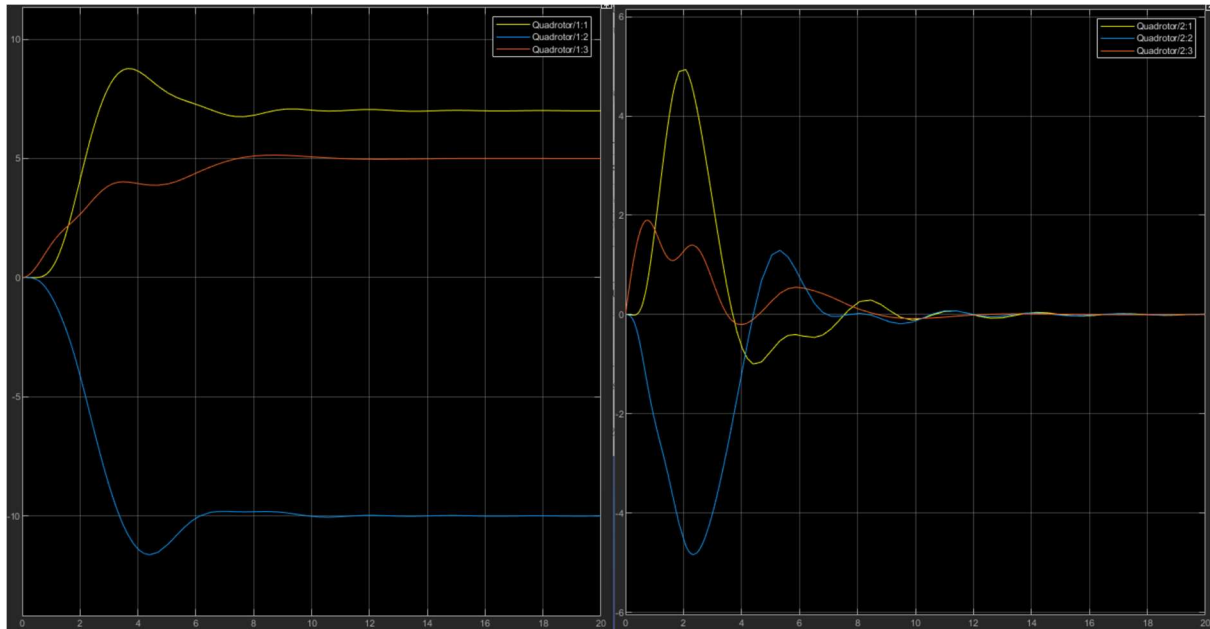
$$e_{Kp} = 10, e_{Ki}=3$$

$$ep_{Kp}=3, e_{Kd}=0.5$$

Simulation

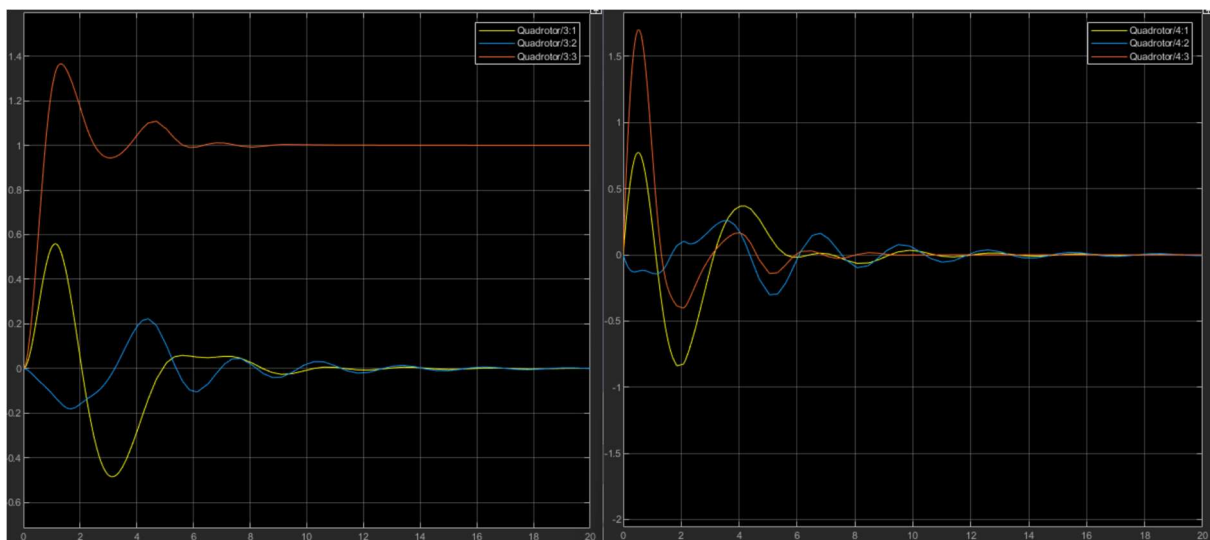
제어입력: Position(7, -10, 5), Heading(1 rad)

왼쪽 위치, 오른쪽 속도



노란선 X, 파란선 Y, 주황선 Z

왼쪽 각도, 오른쪽 각속도



노란선 Roll, 파란선 Pitch, 주황선 Yaw