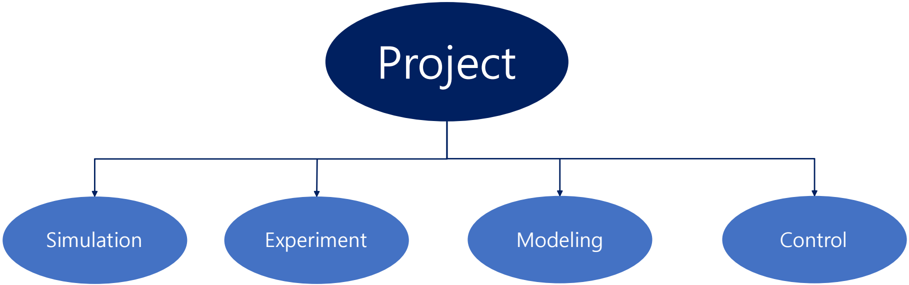
캡스톤 디자인 최종발표

강민구, 박주형, 허홍석

Louis Vuittol

목표



- Matlab
- Simulink

- Hardware
- Embedded
- Python

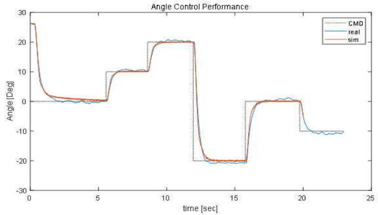
- Plant
- Sensor
- Actuator
- Disturbance
- Noise

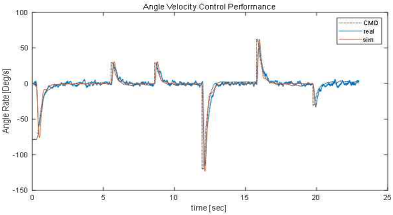
- Stability
- Performance

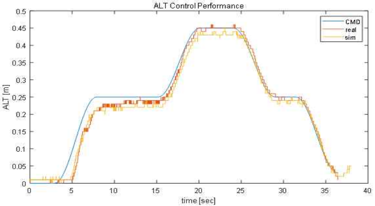
실험 영상

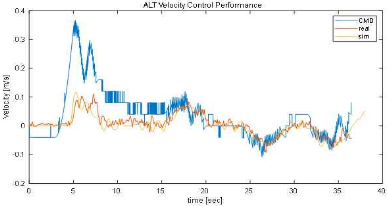
실험 결과

Angle, Height control : experiment & simulation



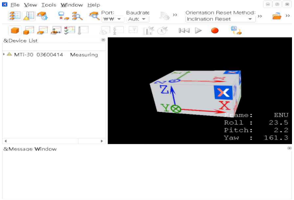


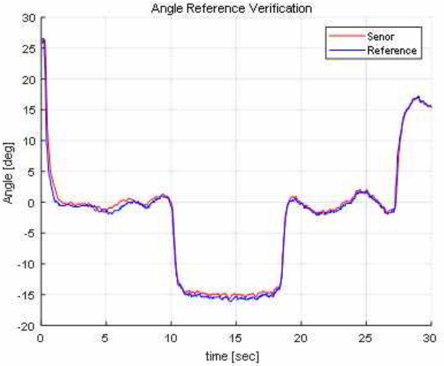




실험 결과

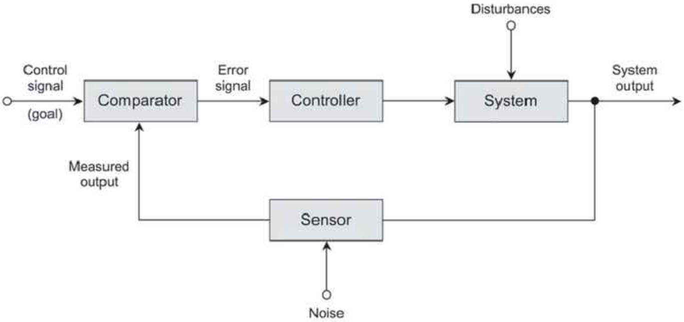
comparison with reference angle





System modeling

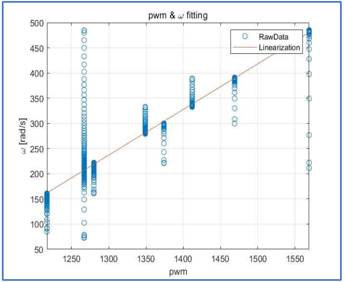
Modeling error

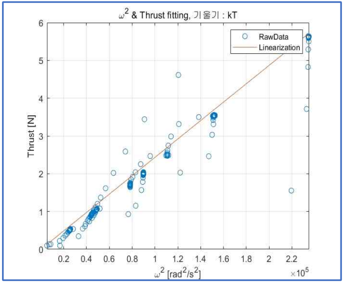


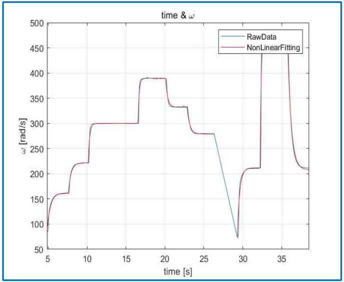
- Sensor modeling
- Plant modeling
- Disturbance
- Noise
- Discrete sampling time

Actuator modeling

Thrust & Propeller test







$$\omega = a \cdot pwm + b$$

$$F = k_T \cdot \omega^2 \longrightarrow Actuator(s) = \frac{K}{\tau s + 1}$$

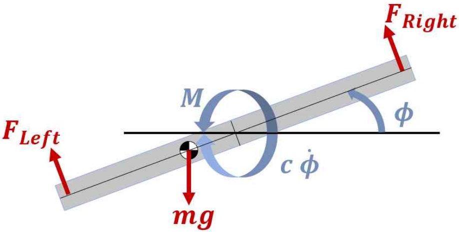
$$\tau = 0.2sec$$

Table 1 Actuator parameter

a	$0.90555s^{-1}$
b	$-940.17s^{-1}$
k_T	$2.4344 \times 10^{-5} kg \cdot m$
τ	$0.2s$

Plant modeling

Dynamic equations



$$m\ddot{z} = F\cos\phi - mg$$
$$I\ddot{\phi} = M - c\dot{\phi} - mgd\cos\phi$$

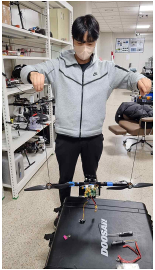
$$\omega = a \cdot pwm + b$$

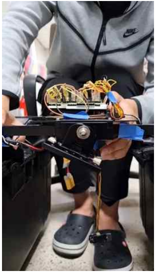
$$F = F_{Right} + F_{Left} = k_T(\omega_{right}^2 + \omega_{left}^2)$$

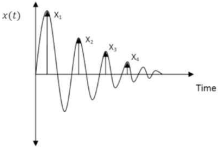
$$M = (F_{Right} - F_{Left})l = k_T l(\omega_{right}^2 + \omega_{left}^2)$$

Plant parameter

experimental measurement







Logarithmic decrement method

$$\delta = \frac{1}{n} ln \left| \frac{x_1}{x_{n+1}} \right|$$

$$\text{Damping} = \zeta = \frac{\delta}{\sqrt{4\pi^2+\delta^2}}$$

$$J = \frac{mgD^2}{16\pi^2 f_n^2 h} = \frac{0.895kg \times 9.81m/s^2 \times (0.56m)^2}{16\pi^2 \times (1.37s^{-1})^2 \times (0.62m)} = 0.015kg \cdot m^2$$

$$\delta = 0.289$$

$$\zeta = \frac{0.288}{\sqrt{4\pi^2 + 0.288^2}} = 0.0458$$

$$c = 2 \times J \times \zeta \times w_n = 2J\zeta \frac{w_d}{\sqrt{1-\zeta^2}}$$

$$= 2 \times 0.00014 \times 0.0458 \times 14.526 = 0.00019kg \cdot m^2/s$$

$$m\ddot{z} = F\cos\phi - mg$$

$$I\ddot{\phi} + c\dot{\phi} + mgd\cos\phi = M$$

$$\omega = a \cdot pwm + b$$

$$F = k_T\,(\omega_{right}^2 + \omega_{left}^2)$$

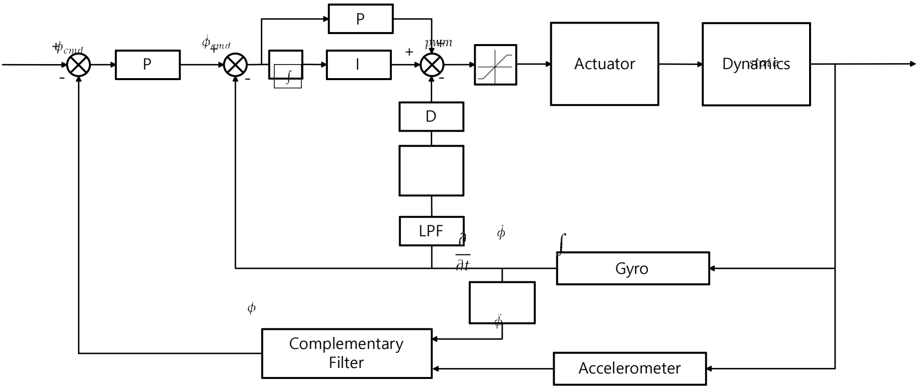
$$M = k_T\, l\,(\omega_{right}^2 - \omega_{left}^2)$$

Table 2 Plant parameter

m	$1.45kg$
d	$4mm$
l	$280mm$
c	$0.00019kg \cdot m/s$
I	$0.015kg \cdot m^2$

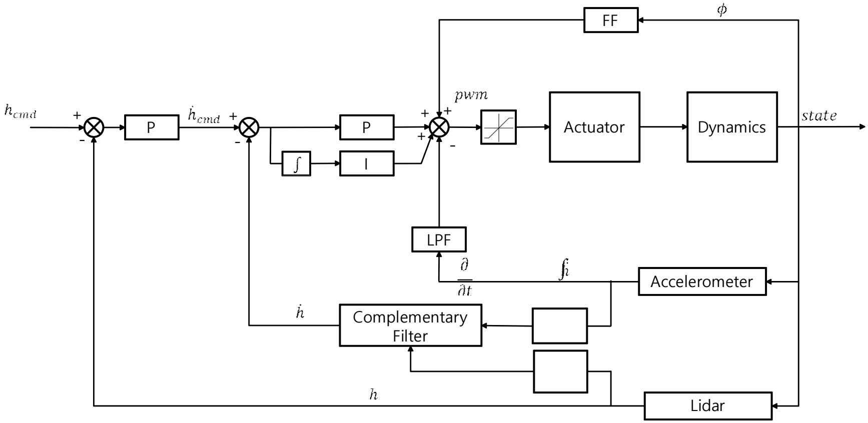
Controller diagram

Attitude controller



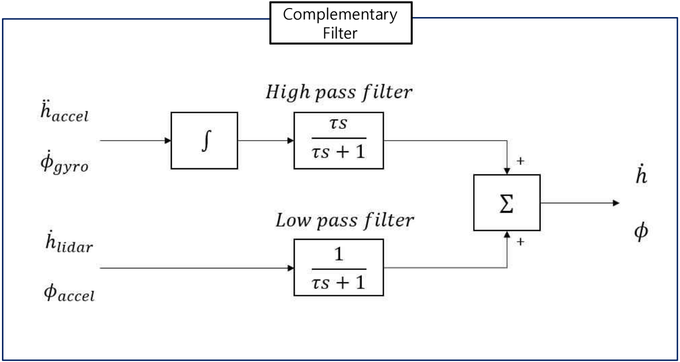
Controller diagram

Position controller



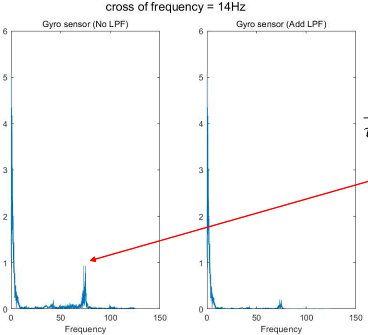
Complementary filter

height rate & angle



Low pass filter

Fourier transform



$$\frac{1}{\tau s + 1}$$

프로펠러 회전에 의한 노이즈 성분

propeller rotation speed $= 453rad/s$

$$f = \frac{\omega}{2\pi} = 72.3Hz$$

주파수의 1/5에 해당하는 지점에 교차주파수 설정

$$\omega_{cross} = 85.7rad/s$$

$Low\ pass\ filter$

Embedded code

backward difference method

Sampling time (dt) = $5ms$

Low pass filter

$$\frac{y(s)}{u(s)} = \frac{1}{\tau s + 1}$$

$$y(s)(\tau s + 1) = u(s)$$

$$\tau \frac{dy}{dt} + y = u$$

$$\tau \frac{y_i - y_{i-1}}{dt} + y_i = u_i$$

$$\tau y_i - \tau y_{i-1} + y_i dt = u_i dt$$

$$y_i(\tau + dt) = \tau y_{i-1} + u_i dt$$

$$y_i = \frac{\tau}{\tau + dt} y_{i-1} + \frac{dt}{\tau + dt} u_i$$

$$y_i = \alpha y_{i-1} + (1 - \alpha) u_i$$

Complementary filter

$$y(s) = \frac{\tau s}{\tau s + 1} \frac{u_1(s)}{s} + \frac{1}{\tau s + 1} u_2(s)$$

$$(\tau s + 1) y(s) = \tau u_1(s) + u_2(s)$$

$$\tau \frac{dy}{dt} + y = \tau u_1 + u_2$$

$$\tau \frac{y_i - y_{i-1}}{dt} + y_i = \tau u_1 + u_2$$

$$\tau(y_i - y_{i-1}) + y_i dt = \tau u_1 dt + u_2 dt$$

$$y_i(\tau + dt) = \tau y_{i-1} + \tau u_1 dt + u_2 dt$$

$$y_i = \frac{\tau}{\tau + dt}(y_{i-1} + u_1 dt) + \frac{dt}{\tau + dt} u_2$$

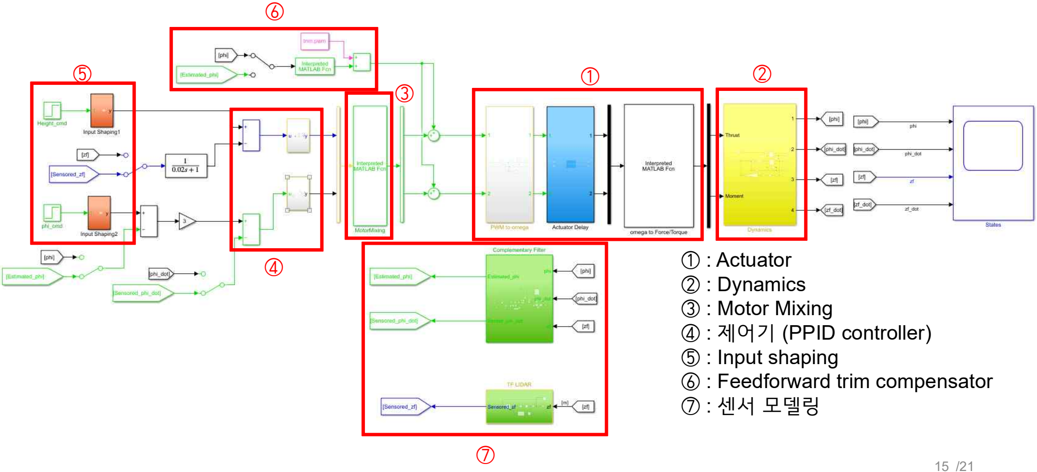
$$y_i = \alpha(y_{i-1} + u_1 dt) + (1 - \alpha) u_2$$

```python
def disPPID(CMD, distance,z2dot, dt, kp1, kp2, ki1, kd1, tau,d_tau):
    global comp_vel, predistance, H_iterm, H_dterm_prev, distance_vel_prev
    height_Error = CMD - distance
    VelCMD = height_Error * kp1
    Vel = (distance - predistance)/dt
    distance_vel = (distance-predistance)/dt
    distance_vel = tau*distance_vel_prev + (1-tau)*(distance_vel)
    distance_vel_prev = distance_vel
    predistance = distance
    comp_vel = tau * (comp_vel + z2dot*dt) + (1-tau)*(distance_vel)
    VelError = VelCMD - comp_vel
    pterm = VelError * kp2
    H_iterm += ki1 * VelError * dt
    dterm = -kd1 * z2dot
    H_dterm_LPF = d_tau*(H_dterm_prev) + (1-d_tau)*(dterm)
    H_dterm_prev = H_dterm_LPF
    output = pterm + H_iterm + H_dterm_LPF
    output = min(output,100)
    output = max(output,-100)
    return output,VelCMD, pterm, H_iterm, dterm,comp_vel, H_dterm_LPF
```

```python
def PPID(CMD, roll, gyro, dt, kp1, kp1_1, ki1, kd1, tau2):
    global iterm1, prevgyro, dterm_prev, RateCMD

    PPID_Error = CMD - roll
    RateCMD = kp1 * PPID_Error

    RateError = RateCMD - gyro
    RateDelta = gyro - prevgyro
    prevgyro = gyro

    pterm1 = kp1_1 * RateError
    iterm1 += ki1 * RateError * dt
    dterm_now = -kd1 * (RateDelta)/dt
    dtermLPF = tau2 * dterm_prev + (1-tau2) * dterm_now
    dterm_prev = dtermLPF

    iterm1 = max(iterm1, -30)
    iterm1 = min(iterm1, 30)
    dtermLPF = max(dtermLPF, -50)
    dtermLPF = min(dtermLPF, 50)
    output = pterm1 + iterm1 + dtermLPF
    output = min(output, 100)
    output = max(output, -100)
    return output, RateCMD, pterm1, iterm1, dterm_now, dtermLPF
```

Simulation structure

Simulink



① : Actuator
② : Dynamics
③ : Motor Mixing
④ : 제어기 (PPID controller)
⑤ : Input shaping
⑥ : Feedforward trim compensator
⑦ : 센서 모델링

제어기 설계

Transfer function

$$\begin{aligned}\dot{\mathrm{x}} &= f(\mathrm{x}, u) \\ y &= h(\mathrm{x}, u)\end{aligned} \longrightarrow \begin{aligned}\dot{\mathrm{x}} &= A\mathrm{x} + Bu \\ y &= C\mathrm{x}\end{aligned}$$

$$f(\mathrm{x}, u) = f(\mathrm{x}_{0}, u_0) + \frac{\partial f}{\partial \mathrm{x}}\mathrm{x} + \ldots$$

$$\mathrm{x}_0 = \begin{bmatrix}\phi_0 \\ \dot{\phi}_0 \\ z_0 \\ \dot{z}_0\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ 0 \\ 0\end{bmatrix}$$

시스템 선형화

$$G(s) = C(sI - A)^{-1}B$$

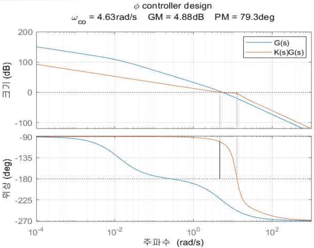
$$\Big\downarrow \; ss2tf$$

$$G_1(s) = \frac{\phi(s)}{\phi_{cmd}(s)} = \frac{203.6}{s^3 + 5.013s^2 + 0.0633s}$$

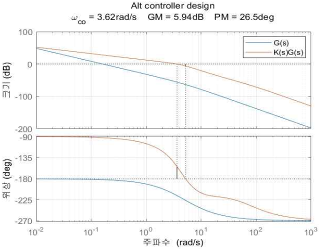
$$G_2(s) = \frac{h(s)}{h_{cmd}(s)} = \frac{0.1313}{s^3 + 5s^2}$$

제어기 설계

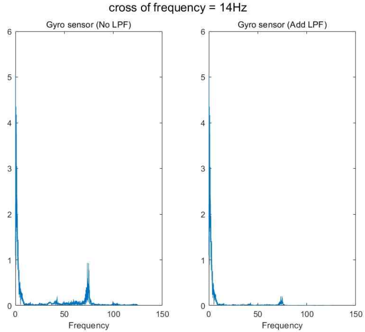
bode plot



K_{P1}	4
K_{P2}	0.7
K_D	0.01
K_I	0.25



K_{P1}	4
K_{P2}	150
K_D	7
K_I	7



$$S_G^C = \frac{G}{C}\frac{\partial C}{\partial G} = \frac{1}{1+GH}$$

$$S_H^C = \frac{H}{C}\frac{\partial C}{\partial H} = -\frac{GH}{1+GH}$$

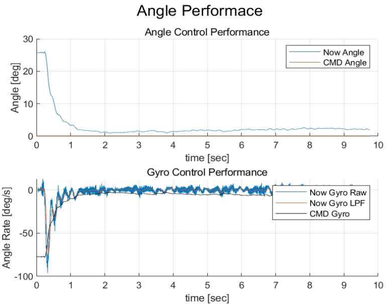
$$\lim_{GH\to\infty} S_G^C = 0$$

$$\lim_{GH\to\infty} S_H^C = -1$$

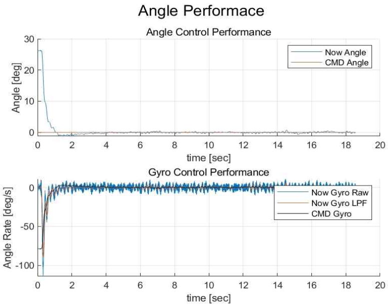
제어기 설계

편심 질량과 적분제어기 영향

K_{P1}	4
K_{P2}	0.7
K_D	0.01
K_I	0.25



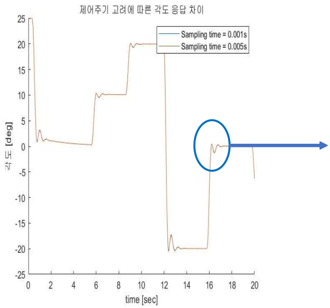
I Gain = 0.01
편심질량에 의한 정상상태 오차 발생

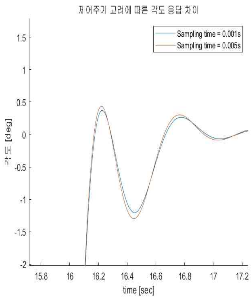


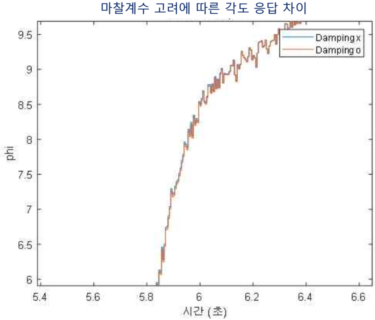
I Gain 0.25
편심질량에 의한 정상상태 오차 제거

모델링 정확성

Sampling time, Damping coefficient

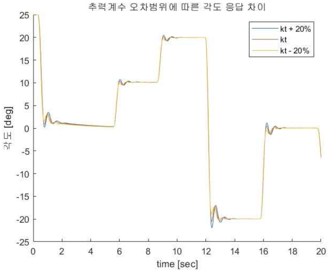


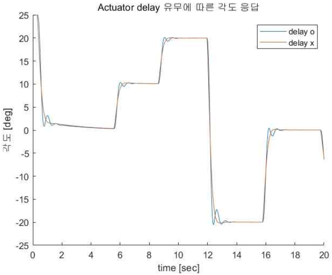


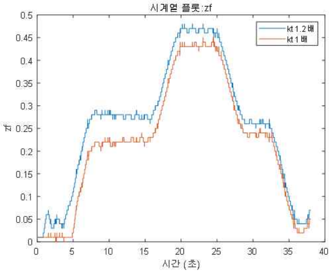


모델링 정확성

Actuator Parameter







감사합니다

강민구, 박주형, 허홍석

Louis Vuittol