2023 Winter MESTER

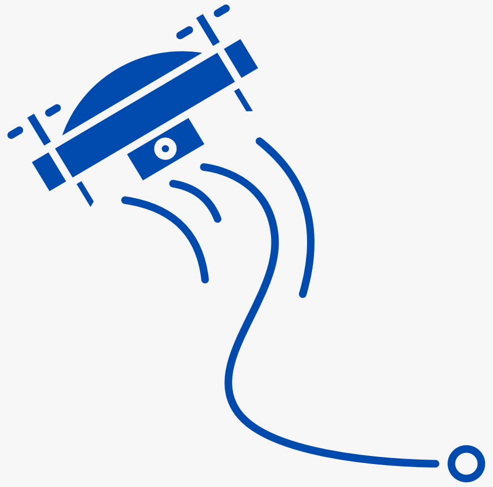
# 2024/02/29 Meeting

## Tilt-Rotor VTOL Modeling and Control

20210027 김지유
20193770 우영찬

# 01.
# 목차





# Task

# Task

### Task 1 (김지유)

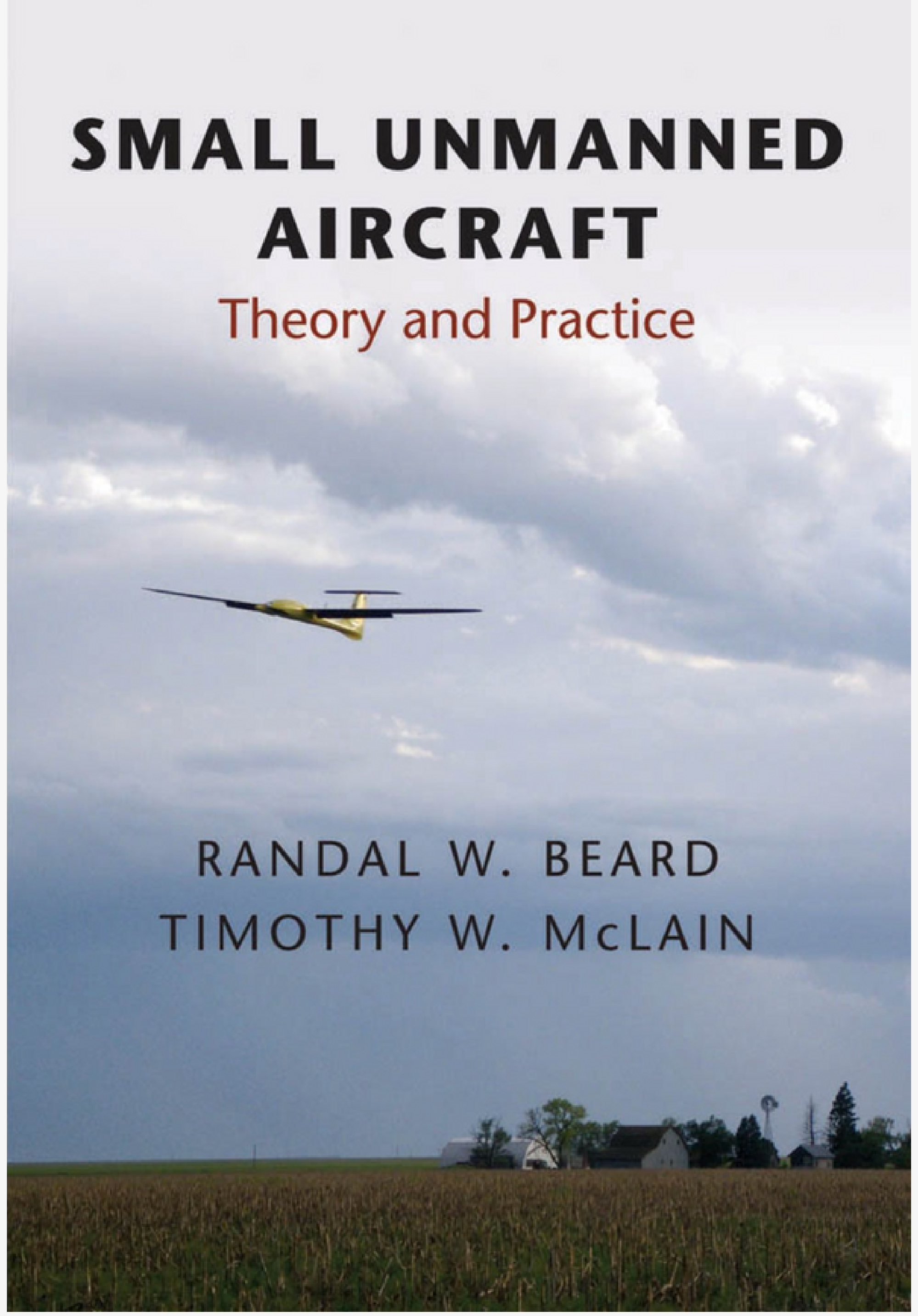
1. Trim >> 제어기 설계를 위한 방정식 / A,B Matrix 이해
2. 어떻게 제어기 설계?

### Task 2 (우영찬)

Straight, level flight에서 longitudinal linear state-space model의 A, B matrix를 얻기 위해
-
Matlab & Simulink를 통해
주어진 동역학 모델의 trim 상태 계산

# 수행내용

## Task 1




Contents

# 수행내용

## Task 1

### Chap3. Kinematics and Dynamics

$$\begin{pmatrix} \dot{p}_n \\ \dot{p}_e \\ \dot{p}_d \end{pmatrix} = \begin{pmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} rv - qw \\ pw - ru \\ qu - pv \end{pmatrix} + \frac{1}{m} \begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix}$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} \Gamma_1 pq - \Gamma_2 qr + \Gamma_3 l + \Gamma_4 n \\ \Gamma_5 pr - \Gamma_6 (p^2 - r^2) + \frac{1}{J_y} m \\ \Gamma_7 pq - \Gamma_1 qr + \Gamma_4 l + \Gamma_8 n \end{pmatrix}$$

# 수행내용

## Task 1

Chap4. Forces and Moments

Gravitational Forces

Aerodynamic Forces and Moments

Propulsion Forces and Moments

# 수행내용

## Task 1

## Chap4. Forces and Moments

**Gravitational Forces**

**Aerodynamic Forces and Moments**

**Propulsion Forces and Moments**

$$\mathbf{f}_g^b = \mathcal{R}_v^b \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \begin{pmatrix} -mg\sin\theta \\ mg\cos\theta\sin\phi \\ mg\cos\theta\cos\phi \end{pmatrix}$$

# 수행내용

## Task 1

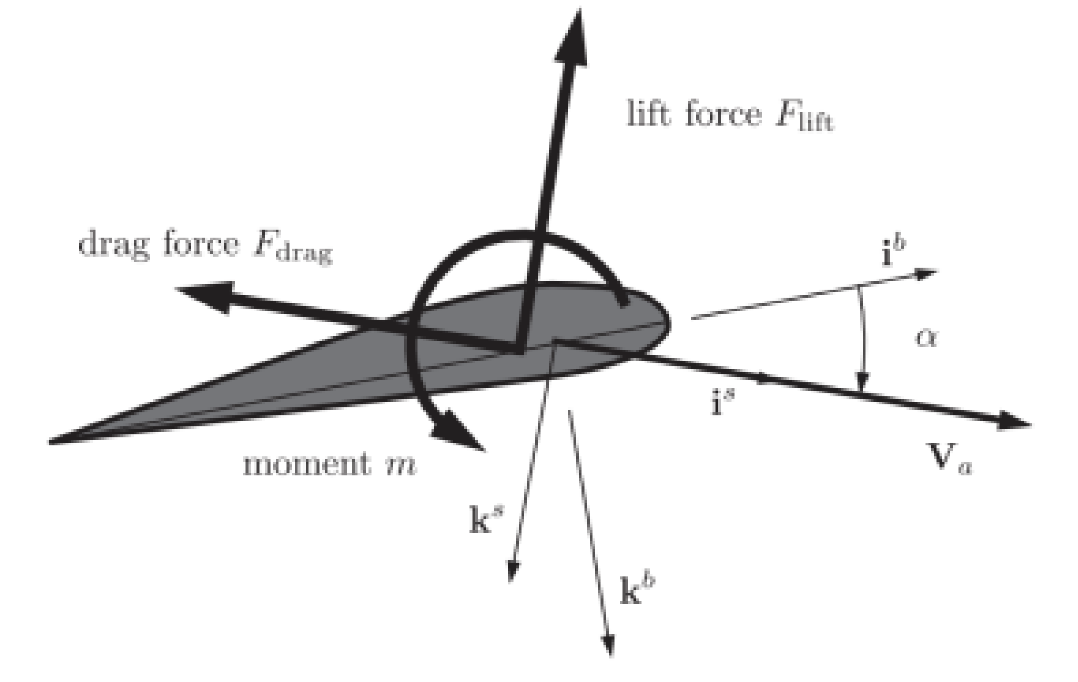
## Chap4. Forces and Moments

**Gravitational Forces**

**Aerodynamic Forces and Moments**

**Propulsion Forces and Moments**

<Longitudinal Aerodynamics>




$$
\begin{aligned}
F_{lift} &= \frac{1}{2}\rho V_a^2 S C_L(\alpha, q, \delta_e) \\
F_{drag} &= \frac{1}{2}\rho V_a^2 S C_D(\alpha, q, \delta_e) \\
m &= \frac{1}{2}\rho V_a^2 S c C_m(\alpha, q, \delta_e)
\end{aligned}
$$

<Lateral Aerodynamics>

$$
\begin{aligned}
f_y &= \frac{1}{2}\rho V_a^2 S C_Y(\beta, p, r, \delta_a, \delta_r) \\
l &= \frac{1}{2}\rho V_a^2 S b C_l(\beta, p, r, \delta_a, \delta_r) \\
n &= \frac{1}{2}\rho V_a^2 S b C_n(\beta, p, r, \delta_a, \delta_r)
\end{aligned}
$$

# 수행내용

## Task 1

### Chap4. Forces and Moments

<Linearization>

Gravitational Forces

Aerodynamic Forces and Moments

Propulsion Forces and Moments

$$F_{lift} = \frac{1}{2}\rho V_a^2 S \left[C_{L_0} + C_{L_\alpha}\alpha + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e\right]$$

$$F_{drag} = \frac{1}{2}\rho V_a^2 S \left[C_{D_0} + C_{D_\alpha}\alpha + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e\right]$$

$$m = \frac{1}{2}\rho V_a^2 S c \left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e\right]$$

$$f_y = \frac{1}{2}\rho V_a^2 S \left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{b}{2V_a}p + C_{Y_r}\frac{b}{2V_a}r + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right]$$

$$l = \frac{1}{2}\rho V_a^2 S b \left[C_{l_0} + C_{l_\beta}\beta + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a + C_{l_{\delta_r}}\delta_r\right]$$

$$n = \frac{1}{2}\rho V_a^2 S b \left[C_{n_0} + C_{n_\beta}\beta + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a + C_{n_{\delta_r}}\delta_r\right]$$

stability derivative

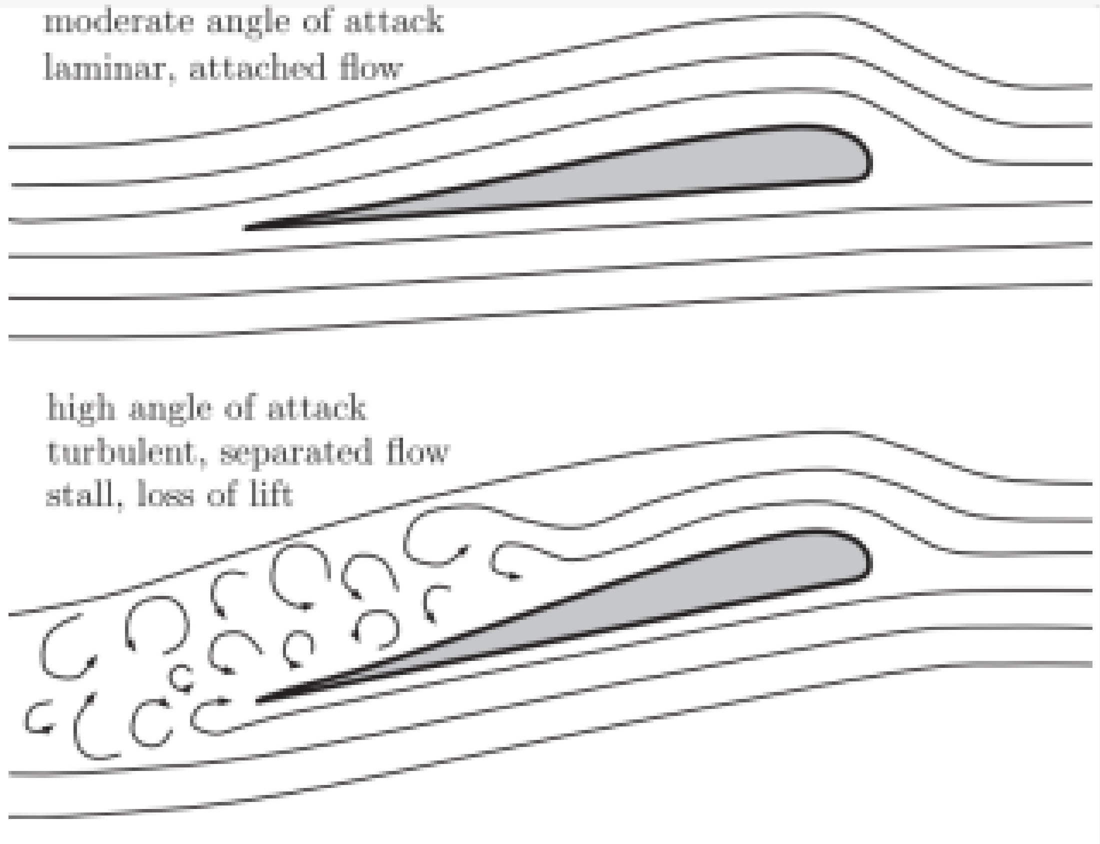
# 수행내용

## Task 1

## Chap4. Forces and Moments

<Stall>

Figure 4.6 The upper drawing depicts a wing under normal flow conditions. The flow is laminar and follows the surface of the wing. The lower drawing shows a wing under stall conditions due to a high angle of attack. In this case, the flow separates from the top surface of the wing, leading to turbulent flow and a significant drop in the lift force produced by the wing.




**Gravitational Forces**

**Aerodynamic Forces and Moments**

**Propulsion Forces and Moments**

$$F_{lift} = \frac{1}{2}\rho V_a^2 S \left[ C_L(\alpha) + C_{L_q}\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e \right]$$

$$C_L(\alpha) = (1-\sigma(\alpha))\left[C_{L_0} + C_{L_\alpha}\alpha\right] + \sigma(\alpha)\left[2\,\mathrm{sign}(\alpha)\sin^2\alpha\cos\alpha\right]$$

$$\sigma(\alpha) = \frac{1 + e^{-M(\alpha-\alpha_0)} + e^{M(\alpha+\alpha_0)}}{\left(1 + e^{-M(\alpha-\alpha_0)}\right)\left(1 + e^{M(\alpha+\alpha_0)}\right)}$$

# 수행내용

## Task 1

## Chap4. Forces and Moments

Gravitational Forces

Aerodynamic Forces and Moments

Propulsion Forces and Moments

$$
\begin{aligned}
T_p(\Omega_p, C_T) &= \frac{\rho D^4}{4\pi^2}\Omega_p^2(C_{T2}J^2 + C_{T1}J + C_{T0}) \\
&= \left(\frac{\rho D^4 C_{T0}}{4\pi^2}\right)\Omega_p^2 + \left(\frac{\rho D^3 C_{T1} V_a}{2\pi}\right)\Omega_p + (\rho D^2 C_{T2} V_a^2) \\
Q_p(\Omega_p, C_Q) &= \frac{\rho D^5}{4\pi^2}\Omega_p^2(C_{Q2}J^2 + C_{Q1}J + C_{Q0}) \\
&= \left(\frac{\rho D^5 C_{Q0}}{4\pi^2}\right)\Omega_p^2 + \left(\frac{\rho D^4 C_{Q1} V_a}{2\pi}\right)\Omega_p + (\rho D^3 C_{Q2} V_a^2)
\end{aligned}
$$

$$
\begin{aligned}
\rho &= \text{Air Density} \\
D &= \text{Propeller Diameter} \\
\Omega_p &= \text{Propeller Speed}(rad/s)
\end{aligned}
$$

$$
\begin{cases} C_T \\ C_Q \end{cases} = \text{Non-dimensional aerodynamic coefficient}
$$

# 수행내용

## Task 1

## Chap5. Linear Design Models

$$\dot{p}_n = (\cos\theta\cos\psi)u + (\sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi)v + (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)w$$

$$\dot{p}_e = (\cos\theta\sin\psi)u + (\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi)v + (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\phi)w$$

$$\dot{p}_d = -\sin\theta u + (\sin\phi\cos\theta)v + (\cos\phi\cos\theta)w$$

$$\dot{u} = rv - qw - g\sin\theta + \frac{\rho V_a^2 S}{2m}\left[C_X(\alpha) + C_{X_q}(\alpha)\frac{cq}{2V_a} + C_{X_{\delta_e}}(\alpha)\delta_e\right] + \frac{T_p}{m}$$

$$\dot{v} = pw - ru + g\cos\theta\sin\phi + \frac{\rho V_a^2 S}{2m}\left[C_{Y_0} + C_{Y_\beta}\beta + C_{Y_p}\frac{bp}{2V_a} + C_{Y_r}\frac{br}{2V_a} + C_{Y_{\delta_a}}\delta_a + C_{Y_{\delta_r}}\delta_r\right]$$

$$\dot{w} = qu - pv + g\cos\theta\cos\phi + \frac{\rho V_a^2 S}{2m}\left[C_Z(\alpha) + C_{Z_q}(\alpha)\frac{cq}{2V_a} + C_{Z_{\delta_e}}(\alpha)\delta_e\right]$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = q\sin\sec\theta + r\cos\phi\sec\theta$$

$$\dot{p} = \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho V_a^2 Sb\left[C_{p_0} + C_{p_\beta}\beta + C_{p_p}\frac{bp}{2V_a} + C_{p_r}\frac{br}{2V_a} + C_{p_{\delta_a}}\delta_a + C_{p_{\delta_r}}\delta_r\right] - \Gamma_3 Q_p$$

$$\dot{q} = \Gamma_5 pr - \Gamma_6(p^2 - r^2) + \frac{\rho V_a^2 Sc}{2J_y}\left[C_{m_0} + C_{m_\alpha}\alpha + C_{m_q}\frac{cq}{2V_a} + C_{m_{\delta_e}}\delta_e\right]$$

$$\dot{r} = \Gamma_7 pq - \Gamma_1 qr + \frac{1}{2}\rho V_a^2 Sb\left[C_{r_0} + C_{r_\beta}\beta + C_{r_p}\frac{bp}{2V_a} + C_{r_r}\frac{br}{2V_a} + C_{r_{\delta_a}}\delta_a + C_{r_{\delta_r}}\delta_r\right] - \Gamma_4 Q_p$$

$$\begin{aligned}
C_{p_0} &= \Gamma_3 C_{l_0} + \Gamma_4 C_{n_0} \\
C_{p_\beta} &= \Gamma_3 C_{l_\beta} + \Gamma_4 C_{n_\beta} \\
C_{p_p} &= \Gamma_3 C_{l_p} + \Gamma_4 C_{n_p} \\
C_{p_r} &= \Gamma_3 C_{l_r} + \Gamma_4 C_{n_r} \\
C_{p_{\delta_a}} &= \Gamma_3 C_{l_{\delta_a}} + \Gamma_4 C_{n_{\delta_a}} \\
C_{p_{\delta_r}} &= \Gamma_3 C_{l_{\delta_r}} + \Gamma_4 C_{n_{\delta_r}} \\
C_{r_0} &= \Gamma_4 C_{l_0} + \Gamma_8 C_{n_0} \\
C_{r_\beta} &= \Gamma_4 C_{l_\beta} + \Gamma_8 C_{n_\beta} \\
C_{r_p} &= \Gamma_4 C_{l_p} + \Gamma_8 C_{n_p} \\
C_{r_r} &= \Gamma_4 C_{l_r} + \Gamma_8 C_{n_r} \\
C_{r_{\delta_a}} &= \Gamma_4 C_{l_{\delta_a}} + \Gamma_8 C_{n_{\delta_a}} \\
C_{r_{\delta_r}} &= \Gamma_4 C_{l_{\delta_r}} + \Gamma_8 C_{n_{\delta_r}}
\end{aligned}$$

$$\begin{aligned}
C_X(\alpha) &\triangleq -C_D(\alpha)\cos\alpha + C_L(\alpha)\sin\alpha \\
C_{X_q}(\alpha) &\triangleq -C_{D_q}(\alpha)\cos\alpha + C_{L_q}(\alpha)\sin\alpha \\
C_{X_{\delta_e}}(\alpha) &\triangleq -C_{D_{\delta_e}}(\alpha)\cos\alpha + C_{L_{\delta_e}}(\alpha)\sin\alpha \\
C_Z(\alpha) &\triangleq -C_D(\alpha)\sin\alpha - C_L(\alpha)\cos\alpha \\
C_{Z_q}(\alpha) &\triangleq -C_{D_q}(\alpha)\sin\alpha - C_{L_q}(\alpha)\cos\alpha \\
C_{Z_{\delta_e}}(\alpha) &\triangleq -C_{D_{\delta_e}}(\alpha)\sin\alpha - C_{L_{\delta_e}}(\alpha)\cos\alpha
\end{aligned}$$

# 수행내용

## Task 1

### Chap5. Linear Design Models

~~5.1 Summary of Nonlinear Equations of Motion~~
5.2 Coordinated Turn
5.3 Trim Conditions
5.4 Transfer Function Models
5.5 Linear State-Space Models

# 수행내용

## Task 2

Longitudinal State-space Equations

$$x_{\mathrm{lon}} \triangleq (u,\ w,\ q,\ \theta,\ h)^\top$$

$$u_{\mathrm{lon}} \triangleq (\delta_e,\ \delta_t)^\top$$

$$\begin{pmatrix} \dot{\bar{u}} \\ \dot{\bar{w}} \\ \dot{\bar{q}} \\ \dot{\bar{\theta}} \\ \dot{\bar{h}} \end{pmatrix} = \begin{pmatrix} X_u & X_w & X_q & -g\cos\theta^* & 0 \\ Z_u & Z_w & Z_q & -g\sin\theta^* & 0 \\ M_u & M_w & M_q & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sin\theta^* & -\cos\theta^* & 0 & u^*\cos\theta^* + w^*\sin\theta^* & 0 \end{pmatrix} \begin{pmatrix} \bar{u} \\ \bar{w} \\ \bar{q} \\ \bar{\theta} \\ \bar{h} \end{pmatrix} + \begin{pmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & 0 \\ M_{\delta_e} & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{\delta}_e \\ \bar{\delta}_t \end{pmatrix}$$

Trim 상태의 u, w, alpha, q, airspeed, delta_e, delta_t, theta 를 알아야 함

Longitudinal State-space Model Coefficients

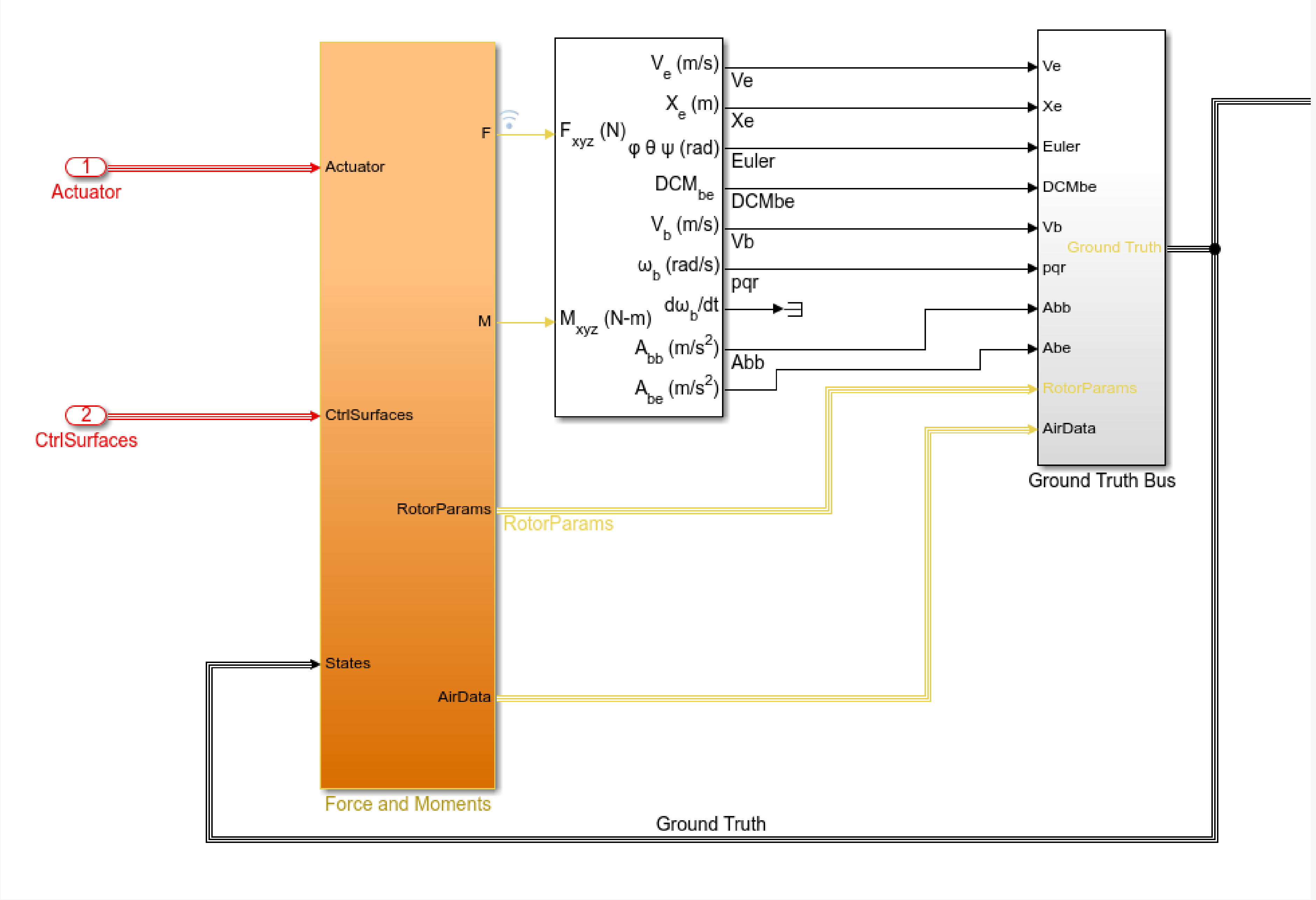
| *Longitudinal* | *Formula* |
|---|---|
| $X_u$ | $\frac{u^*\rho S}{m}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] - \frac{\rho S w^* C_{X_\alpha}}{2m} + \frac{\rho S c C_{X_q} u^* q^*}{4mV_a^*} - \frac{\rho S_{\mathrm{prop}} C_{\mathrm{prop}} u^*}{m}$ |
| $X_w$ | $-q^* + \frac{w^*\rho S}{m}\left[C_{X_0} + C_{X_\alpha}\alpha^* + C_{X_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{X_q} w^* q^*}{4mV_a^*} + \frac{\rho S C_{X_\alpha} u^*}{2m} - \frac{\rho S_{\mathrm{prop}} C_{\mathrm{prop}} w^*}{m}$ |
| $X_q$ | $-w^* + \frac{\rho V_a^* S C_{X_q} c}{4m}$ |
| $X_{\delta_e}$ | $\frac{\rho V_a^{*2} S C_{X_{\delta_e}}}{2m}$ |
| $X_{\delta_t}$ | $\frac{\rho S_{\mathrm{prop}} C_{\mathrm{prop}} k^2 \delta_t^*}{m}$ |
| $Z_u$ | $q^* + \frac{u^*\rho S}{m}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] - \frac{\rho S C_{Z_\alpha} w^*}{2m} + \frac{u^*\rho S C_{Z_q} c q^*}{4mV_a^*}$ |
| $Z_w$ | $\frac{w^*\rho S}{m}\left[C_{Z_0} + C_{Z_\alpha}\alpha^* + C_{Z_{\delta_e}}\delta_e^*\right] + \frac{\rho S C_{Z_\alpha} u^*}{2m} + \frac{\rho w^* S c C_{Z_q} q^*}{4mV_a^*}$ |
| $Z_q$ | $u^* + \frac{\rho V_a^* S C_{Z_q} c}{4m}$ |
| $Z_{\delta_e}$ | $\frac{\rho V_a^{*2} S C_{Z_{\delta_e}}}{2m}$ |
| $M_u$ | $\frac{u^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] - \frac{\rho S c C_{m_\alpha} w^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* u^*}{4J_y V_a^*}$ |
| $M_w$ | $\frac{w^*\rho S c}{J_y}\left[C_{m_0} + C_{m_\alpha}\alpha^* + C_{m_{\delta_e}}\delta_e^*\right] + \frac{\rho S c C_{m_\alpha} u^*}{2J_y} + \frac{\rho S c^2 C_{m_q} q^* w^*}{4J_y V_a^*}$ |
| $M_q$ | $\frac{\rho V_a^* S c^2 C_{m_q}}{4J_y}$ |
| $M_{\delta_e}$ | $\frac{\rho V_a^{*2} S c C_{m_{\delta_e}}}{2J_y}$ |

Beard and Mclain - Small Unmanned Aircraft Theory and Practice

# 수행내용

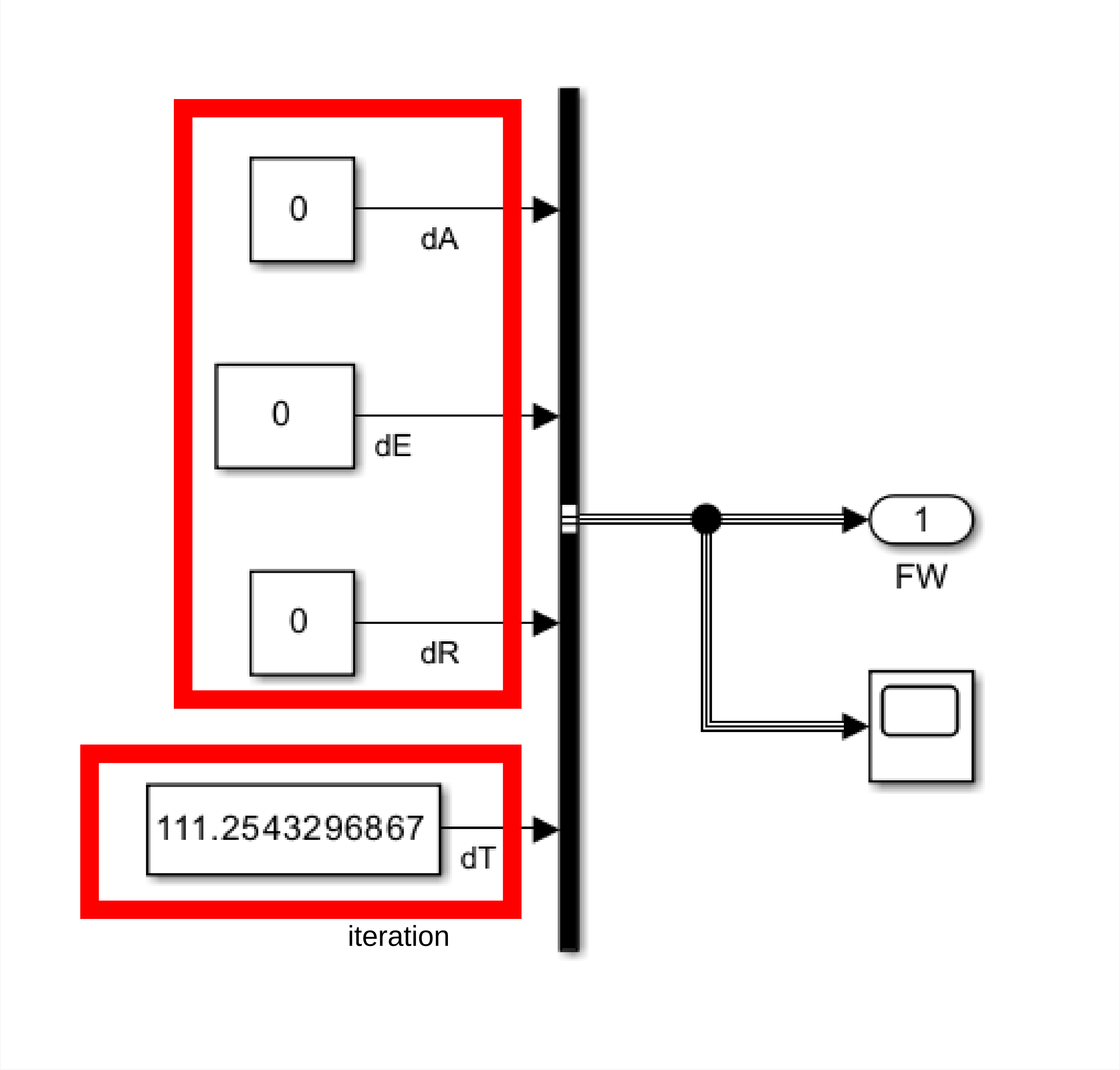
## Task 2

Longitudinal State-space Equations




Trim 상태 : 고도 유지(V_Z = 0)
pitch 변화율 = 0
airspeed = 일정

dA=dE=dR=0 일 때 dT의 값에 변화를 주며 Trim 상태를 만듦




(To Voltage) dT = 0.6675

# 수행내용

## Task 2 직접 계산 vs 모델선형기 앱

**동작점:** 선형 분석 작업 공간의 "op_dE0dT111"

**크기:** 2개 입력, 1개 출력, 4개 상태

**선형화 결과:**

A =

| | x1 | x2 | x3 | x4 |
|---|---|---|---|---|
| x1 | 0 | 1 | 0 | 0 |
| x2 | -5.684e-14 | -14.68 | 0.08472 | -9.824 |
| x3 | -9.793 | -0.08065 | -0.5585 | 0.4033 |
| x4 | -1.421e-14 | 15.1 | -1.463 | -3.796 |

B =

| | u1 | u2 |
|---|---|---|
| x1 | 0 | 0 |
| x2 | -163.2 | 3.415e-17 |
| x3 | -0.6519 | 0.06584 |
| x4 | -7.262 | -4.031e-18 |

C =

| | x1 | x2 | x3 | x4 |
|---|---|---|---|---|
| y1 | 1 | 0 | 0 | 0 |

D =

| | u1 | u2 |
|---|---|---|
| y1 | 0 | 0 |

**상태 이름:**
x1 - phi theta psi(2)
x2 - p,q,r (2)
x3 - ub,vb,wb(1)
x4 - ub,vb,wb(3)

**입력 채널 이름:**
u1 - dE
u2 - dT

**출력 채널 이름:**
y1 - pitchAngle

```
A =

   -0.1754   -0.4229   -0.1042   -9.8099         0
   -1.3546         0   13.6596   -0.0422         0
    0.0002   -8.8872  -13.2764         0         0
         0         0    1.0000         0         0
    0.0043   -1.0000         0   14.4738         0


B =

   -0.1624  133.9701
   -5.9516         0
 -133.5616         0
         0         0
         0         0
```

<직접 계산한 A, B 행렬>

```
A =

   -0.5585    0.4033   -0.0806   -9.7930
   -1.4630   -3.7960   15.1000   -0.0000
    0.0847   -9.8240  -14.6800   -0.0000
         0         0    1.0000         0


B =

   -0.6519    0.0658
   -7.2620   -0.0000
 -163.2000    0.0000
         0         0
```

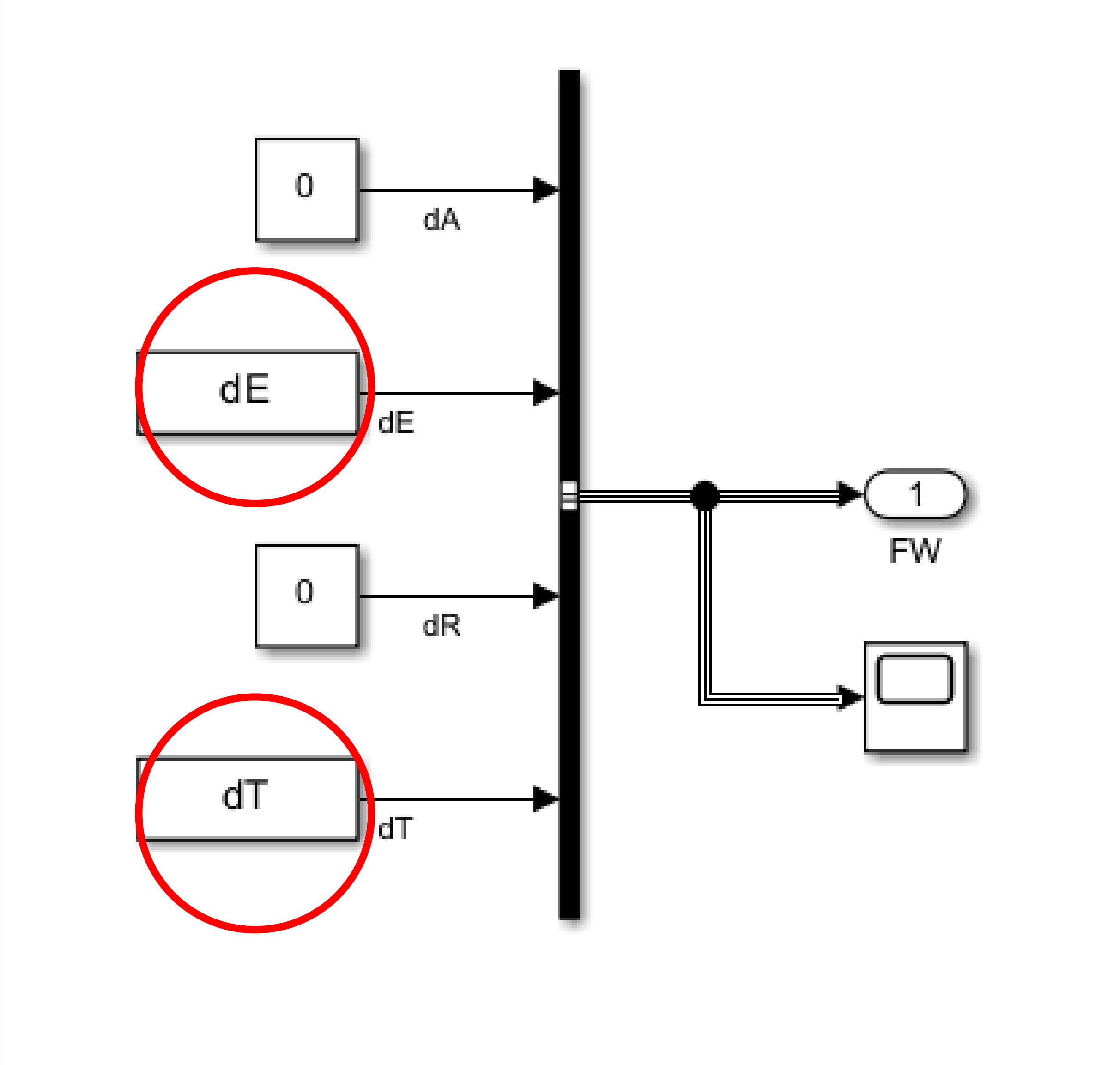
$\bar{u}$
$\bar{w}$
$\bar{q}$
$\bar{\theta}$

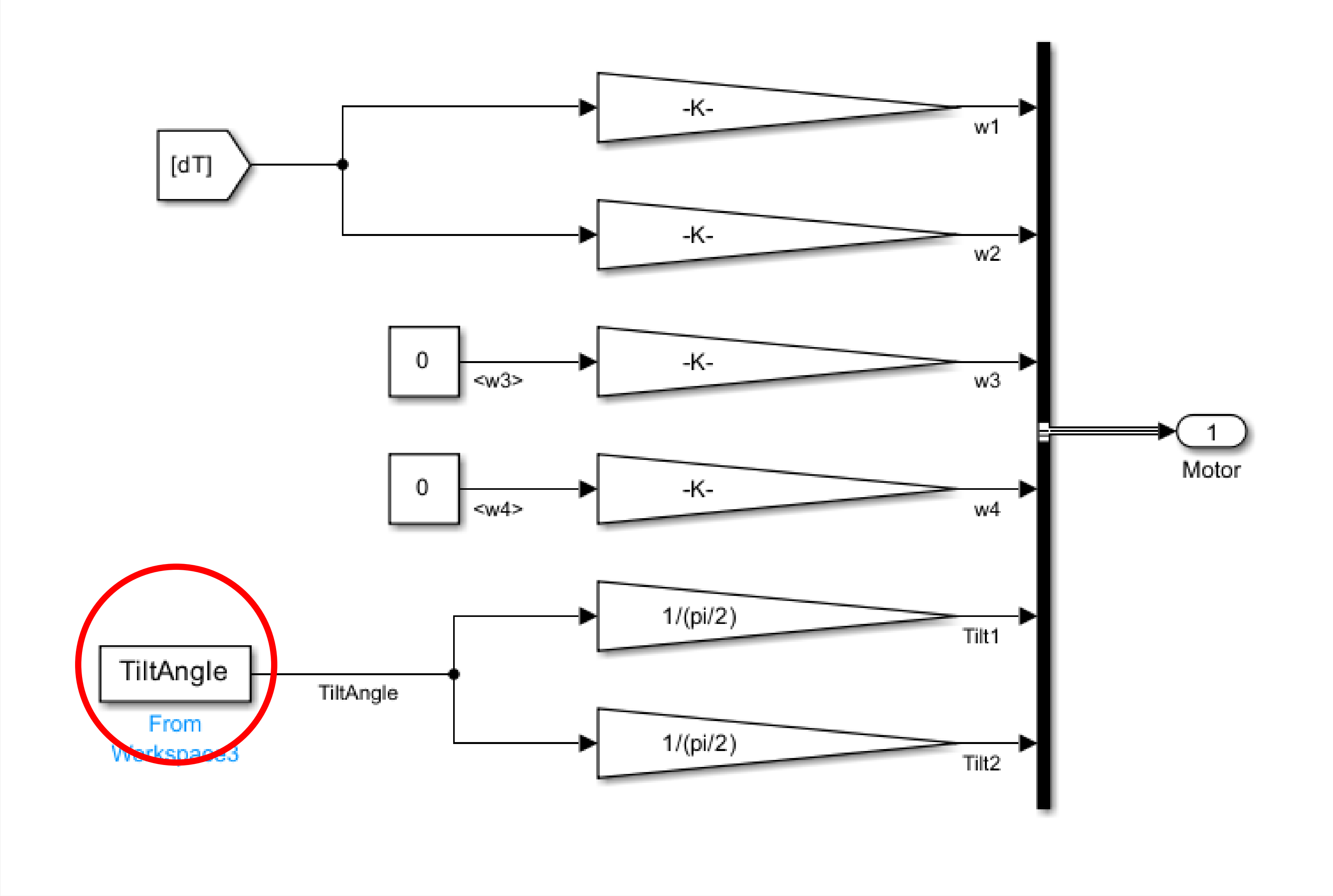
$\bar{\delta}_e$
$\bar{\delta}_t$

<시뮬링크 모델선형기 앱>

# 수행내용

## Task 2 매트랩에서 트림 상태 얻기



# 수행내용

## Task 2 매트랩에서 트림 상태 얻기(1)

```matlab
function optimalSolution()
    % TiltAngle 고정
    TiltAngle = [0 deg2rad(90)];

    % dE 고정
    dE = 0.0;

    % dT 범위 설정
    dT_range = [0, 160];

    % 목적 함수
    fun = @(dT) myfun(dT, dE, TiltAngle);

    % 그래프 설정
    figure;
    hold on;
    xlabel('Iteration');
    ylabel('dT');
    zlabel('f');

    % 그래프를 위한 데이터 저장소
    iter_data = [];
    dT_data = [];
    f_data = [];

    % dT에 대해서 최적화 수행
    iter = 0;
    while true
        % dT 범위를 10등분
        dT_values = linspace(dT_range(1), dT_range(2), 11);

        % 각 dT 값에 대한 목적 함수 값 계산
        f_values = arrayfun(fun, dT_values);

        % 목적 함수의 최솟값과 그 때의 dT 값 찾기
        [f_min, idx_min] = min(f_values);
        dT_min = dT_values(idx_min);

        % 최적해 출력
        fprintf('Iteration: %d, dT: %.10f, f: %.10f\n', iter, dT_min, f_min);
```


```matlab
    while true
        % dT 범위를 10등분
        dT_values = linspace(dT_range(1), dT_range(2), 11);

        % 각 dT 값에 대한 목적 함수 값 계산
        f_values = arrayfun(fun, dT_values);

        % 목적 함수의 최솟값과 그 때의 dT 값 찾기
        [f_min, idx_min] = min(f_values);
        dT_min = dT_values(idx_min);

        % 최적해 출력
        fprintf('Iteration: %d, dT: %.10f, f: %.10f\n', iter, dT_min, f_min);
```


```matlab

        % 그래프를 위한 데이터 저장
        iter_data = [iter_data; iter];
        dT_data = [dT_data; dT_min];
        f_data = [f_data; f_min];

        % 그래프 업데이트
        plot3(iter_data, dT_data, f_data, 'bo');
        drawnow;

        % 종료 조건 검사
        if f_min < 1e-6
            break;
        end

        % 다음 반복을 위해 dT 범위 조정
        if idx_min == 1
            dT_range = dT_values(1:2);
        elseif idx_min == 11
            dT_range = dT_values(10:11);
        else
            dT_range = dT_values(idx_min-1:idx_min+1);
        end

        iter = iter + 1;
    end
end
```

```matlab

function f = myfun(dT, dE, TiltAngle)
    % 변수 추출
    dE = [0 dE];
    dT = [0 dT];

    % 시뮬링크 모델 실행
    assignin('base', 'dE', dE);
    assignin('base', 'dT', dT);
    assignin('base', 'TiltAngle', TiltAngle);
    out1 = sim("Dynamics3");

    % 목적 함수 정의 (각 항을 시뮬레이션 마지막 값의 절대값으로 계산)
    Vb_dot_abs1 = abs(out1.Vb_dot(end,1));
    Vb_dot_abs3 = abs(out1.Vb_dot(end,3));
    Ve_abs3 = abs(out1.Ve(end,3));
    pitchRate_abs1 = abs(out1.pitchRate(end,1));
    pitchRate_dot_abs1 = abs(out1.pitchRate_dot(end,1));

    f = Vb_dot_abs1 + Vb_dot_abs3 + Ve_abs3 + pitchRate_abs1 + pitchRate_dot_abs1;
end

```

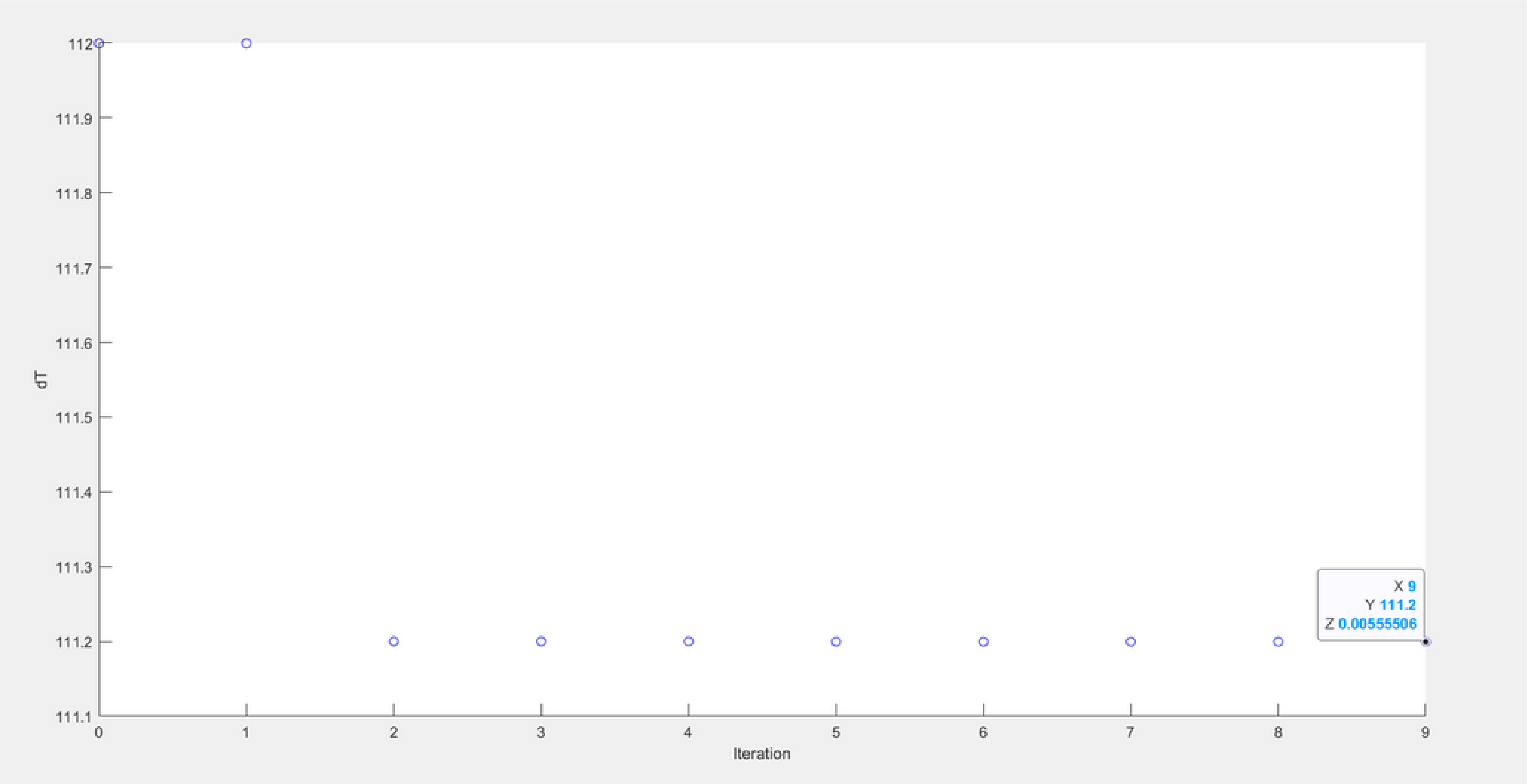
목적함수는
u_dot, w_dot, h_dot, theta_dot, q_dot의
절댓값의 합
-> 최소가 될 때, 각 항이 0에 가까워지도록

dE와 dT의 범위를 정하고
구간 안에서 목적함수가 최소가 되게 하는
dT를 포함한 양쪽 구간을 10등분 하는 것을
반복하여 목적 함수가 매우 작아질 때까지
반복

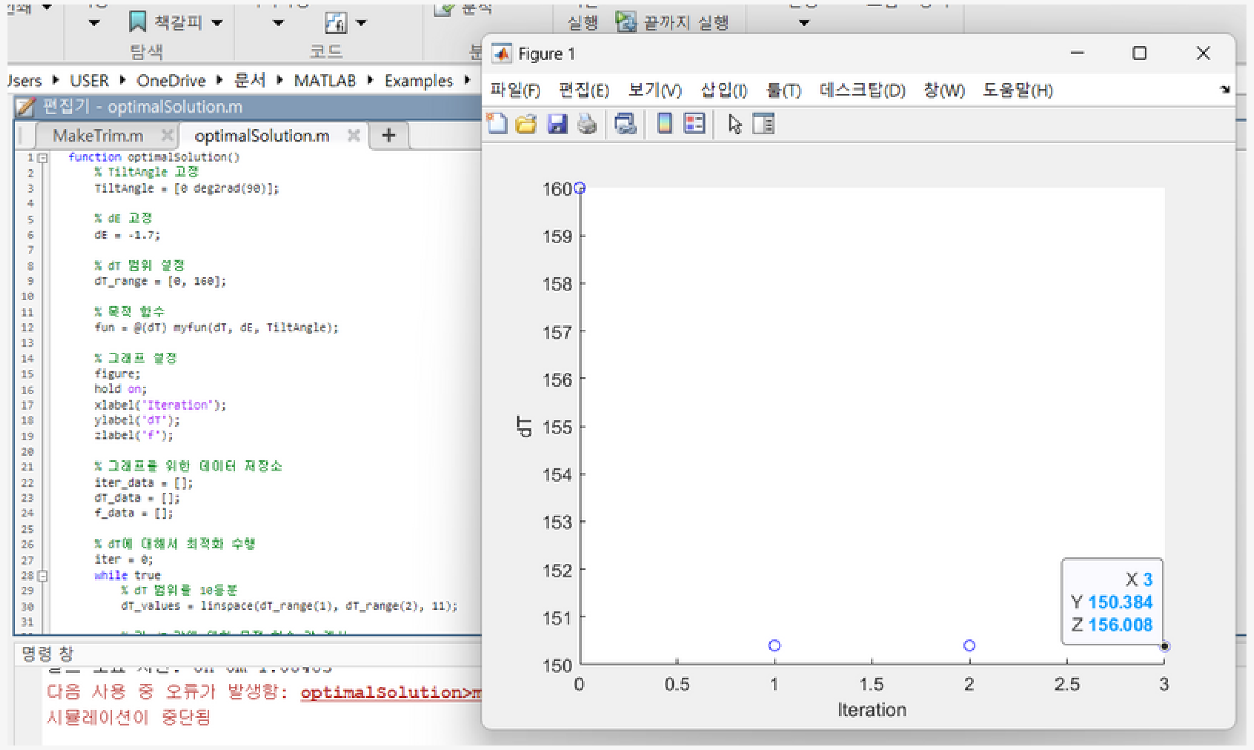
우선 dE=0 고정하여
그 때의 dT를 찾으려고 함

# 수행내용

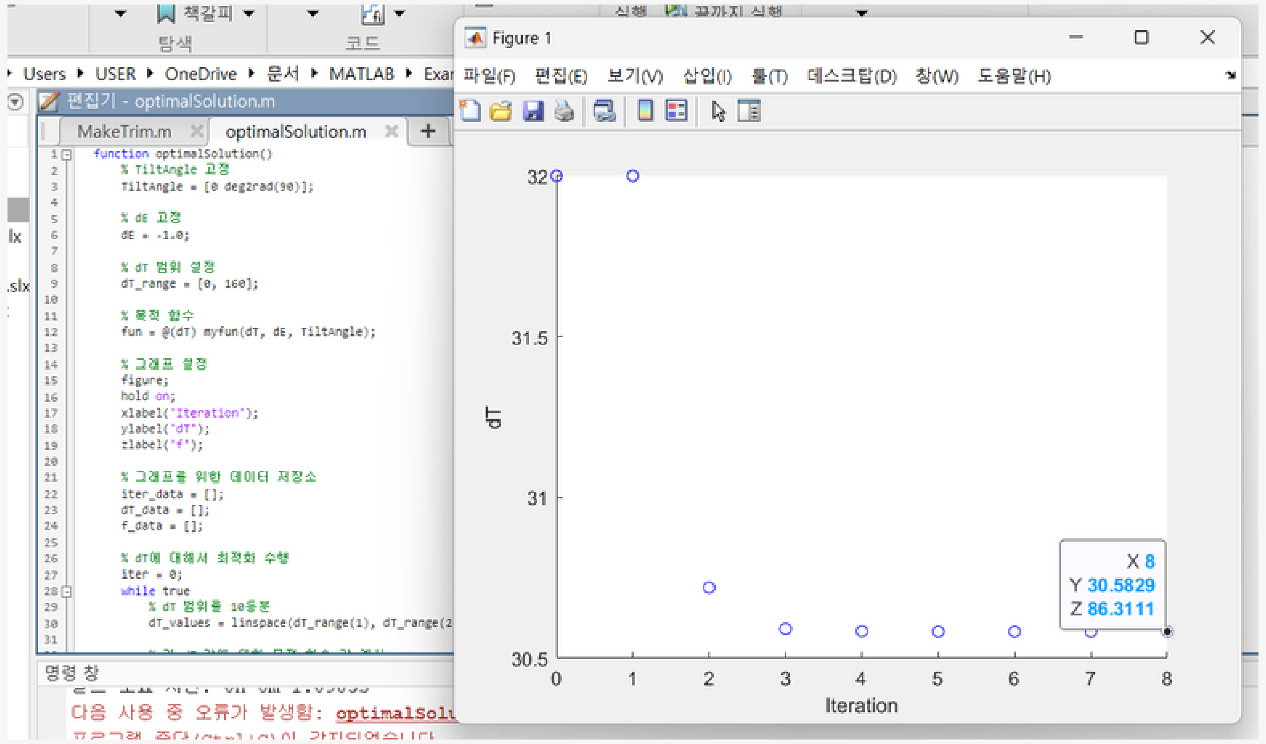
## Task 2 매트랩에서 트림 상태 얻기(1)



dE=0일 때, dT와 목적함수 f



dE=-1.7일 때, dT와 목적함수 f



dE=-1.0일 때, dT와 목적함수 f

# 수행내용

## Task 2 매트랩에서 트림 상태 얻기(2)

```
function optimalSolution()
    % 초기 추정값
    x0 = [0.0, 112.0];

    % 목적 함수
    fun = @myfun;

    % TiltAngle 고정
    TiltAngle = [0 deg2rad(90)];

    % 변수의 하한과 상한 설정
    lb = [-1.7, 0];
    ub = [1.7, 166.66666];

    % 검토한 값들을 저장할 배열 초기화
    global iter_values;
    iter_values = [];

    % 최적화 옵션 설정.
    options = optimoptions('fmincon', 'Algorithm', 'interior-point', 'OptimalityTolerance', 1e-12, 'StepTolerance', 1e-12, 'PlotFcn', @optimplotfval);

    % 최적화 수행
    x = fmincon(@(x) fun(x, TiltAngle), x0, [], [], [], [], lb, ub, [], options);

    % 최적해 출력
    fprintf('TiltAngle: [%.10f, %.10f], dE: %.10f, dT: %.10f\n', TiltAngle(1), TiltAngle(2), x(1), x(2));

    % 저장된 검토값들 출력
    for i=1:size(iter_values, 1)
        fprintf('Iteration: %d, dE: %.10f, dT: %.10f, f: %.10f\n', i, iter_values(i, 1), iter_values(i, 2), iter_values(i, 3));
    end
end

function f = myfun(x, TiltAngle)
    % 변수 추출
    dE = [0 x(1)];
    dT = [0 x(2)];

    % 시뮬링크 모델 실행
    assignin('base', 'dE', dE);
    assignin('base', 'dT', dT);
    assignin('base', 'TiltAngle', TiltAngle);
    out1 = sim("Dynamics3");

    % 목적 함수 정의 (각 항을 시뮬레이션 마지막 값의 절대값으로 계산)
    Vb_dot_abs1 = abs(out1.Vb_dot(end,1));
    Vb_dot_abs3 = abs(out1.Vb_dot(end,3));
    Ve_abs3 = abs(out1.Ve(end,3));
    pitchRate_abs1 = abs(out1.pitchRate(end,1));
    pitchRate_dot_abs1 = abs(out1.pitchRate_dot(end,1));

    f = Vb_dot_abs1 + Vb_dot_abs3 + Ve_abs3 + pitchRate_abs1 + pitchRate_dot_abs1;

    % 검토한 dE, dT 및 목적 함수 값 저장
    global iter_values;
    iter_values = [iter_values; x(1), x(2), f];
end
```

내부점법 : 볼록 최적화에서 최적해를 실현 가능영역 의 내부에서 찾아가는 방법

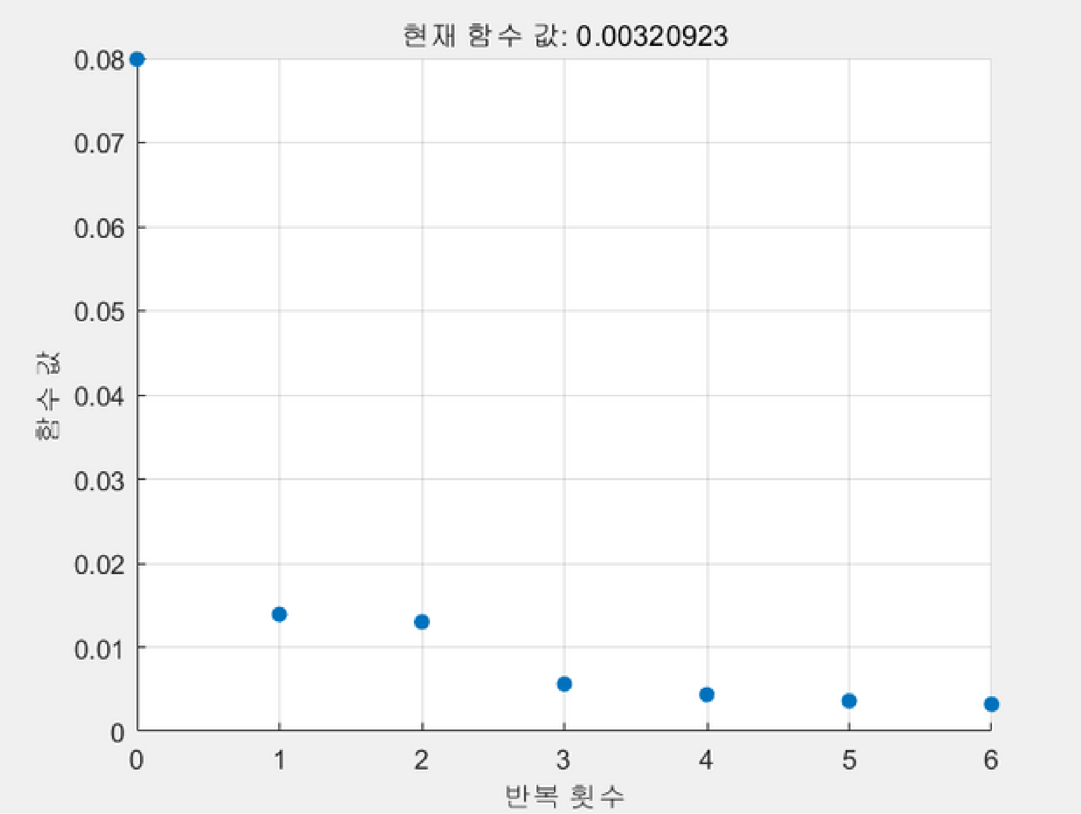
볼록하다면 비선형 계획법에서도 적용가능

```
Iteration: 1, dE: 0.0000000000, dT: 112.0000000000, f: 0.0798991293
Iteration: 2, dE: 0.0000000149, dT: 112.0000000000, f: 0.0768956004
Iteration: 3, dE: 0.0000000000, dT: 112.0000016689, f: 0.0780179210
Iteration: 4, dE: 0.4042683867, dT: 166.3933267000, f: 489.3581611340
Iteration: 5, dE: 0.2021341934, dT: 139.1966633500, f: 26.6847137586
Iteration: 6, dE: 0.1010670967, dT: 125.5983316750, f: 3.3632513480
Iteration: 7, dE: 0.0505335483, dT: 118.7991658375, f: 0.7159133912
Iteration: 8, dE: 1.0871713631, dT: 112.0102829043, f: 387.8193675074
Iteration: 9, dE: 0.5435856815, dT: 112.0051414521, f: 107.1064690874
Iteration: 10, dE: 0.2717928408, dT: 112.0025707261, f: 47.9982457897
Iteration: 11, dE: 0.1358964204, dT: 112.0012853630, f: 9.3471108498
Iteration: 12, dE: 0.0679482102, dT: 112.0006426815, f: 2.3188427292
Iteration: 13, dE: 0.0339741051, dT: 112.0003213408, f: 0.8261907938
Iteration: 14, dE: 0.0169870525, dT: 112.0001606704, f: 0.3172585432
```

.

.

```
Iteration: 78, dE: 0.0038131894, dT: 112.0110561670, f: 0.0060576610
Iteration: 79, dE: 0.0038131897, dT: 112.0110561670, f: 0.0055077938
Iteration: 80, dE: 0.0038131899, dT: 112.0110561670, f: 0.0037416338
Iteration: 81, dE: 0.0038131899, dT: 112.0110561670, f: 0.0034338270
Iteration: 82, dE: 0.0038131900, dT: 112.0110561670, f: 0.0039739442
Iteration: 83, dE: 0.0038131900, dT: 112.0110561670, f: 0.0092173641
Iteration: 84, dE: 0.0038131900, dT: 112.0110561670, f: 0.0068475442
Iteration: 85, dE: 0.0038131900, dT: 112.0110561670, f: 0.0062347389
Iteration: 86, dE: 0.0038131900, dT: 112.0110561670, f: 0.0038800391
Iteration: 87, dE: 0.0038131900, dT: 112.0110561670, f: 0.0043149727
Iteration: 88, dE: 0.0038131900, dT: 112.0110561670, f: 0.0059211798
```




국소 최솟값이 있을 수 있습니다. 제약 조건이 충족되었습니다.

현재 스텝의 크기가 스텝 크기 허용오차 값보다 작고
제약 조건이 제약 조건 허용오차의 값 이내에서
충족되기 때문에 fmincon이(가) 중지되었습니다.

<중지 기준 세부 정보>
TiltAngle: [0.0000000000, 1.5707963268], dE: 0.0038131900, dT: 112.0110561670

# 수행내용

**Task 2** 매트랩에서 트림 상태 얻기

TiltAngle : 90deg 고정
-> 목적함수 최소화하는 dE & dT 쌍 찾기

→

TiltAngle : 0~90deg 변화
-> 목적함수 최소화하는 dE & dT 쌍 찾기

# 질문

적절한 최적화 기법?

TiltAngle / dE / dT
-TiltAngle 90deg -> dE & dT
-TiltAngle 0~90deg -> dE & dT

트림 조건

목적함수는
u_dot, w_dot, h_dot, theta_dot, q_dot의
절댓값의 합
-> 최소가 될 때, 각 항이 0에 가까워지도록

dE & dT 의 범위

감사합니다

# 수행내용

## Task 2

### 질문

# 연구 목표

# 연구 목표

논문 공부
&기초 개념 공부

시뮬링크 예제
following

PID 튜닝
&
Tilt Corridor

제어기 수정&설계

MATLAB 그래프와
3D Animation을
통해 성능 검증

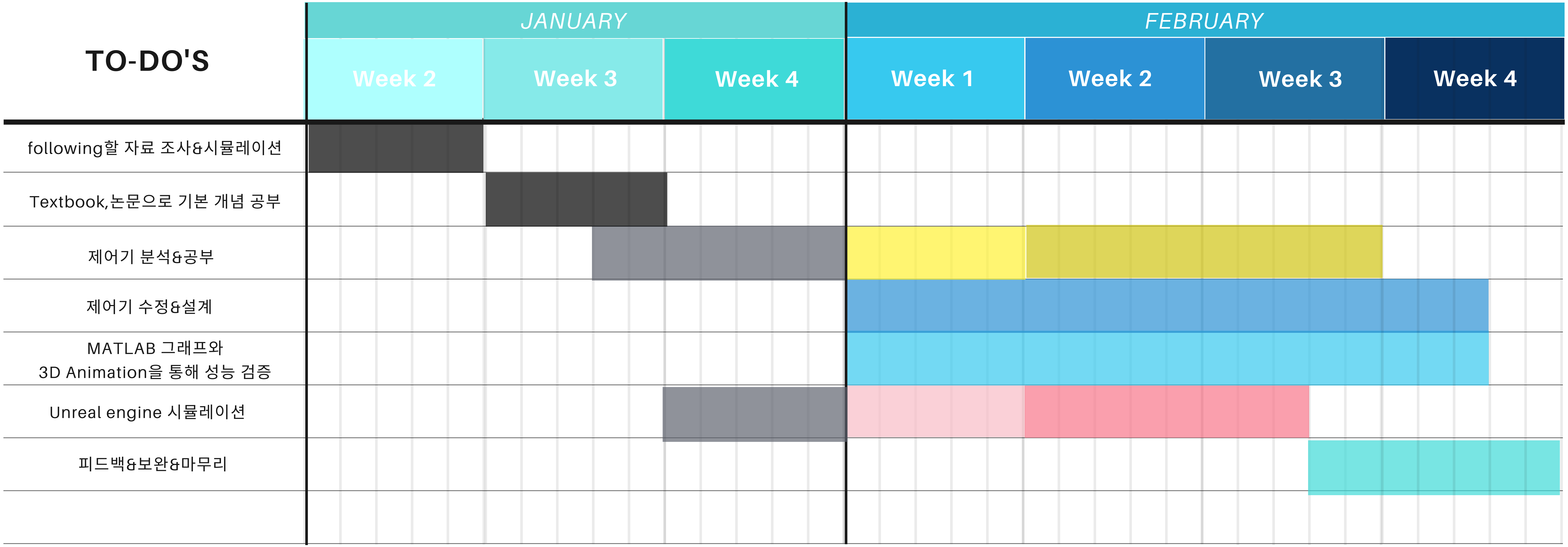
coordinate

Unreal Engine
시뮬레이션

# 스케줄

# gantt chart

Schedule



# 현재 연구 논의 내용

# 현재 연구 논의 내용

# 문의 사항

# 문의사항
## Q&A