임베디드 시스템 Final Presentation

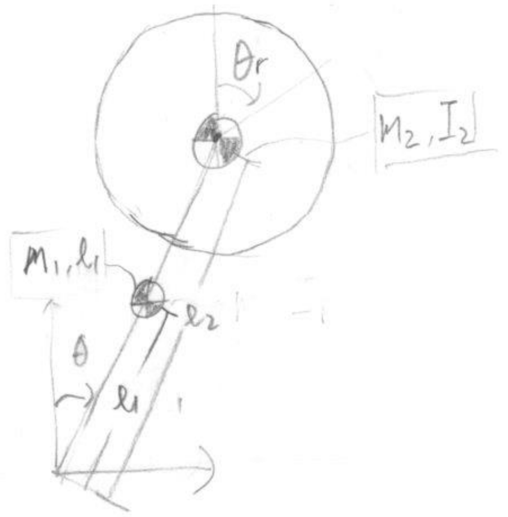
Inverted Pendulum

20510046 박동규

INDEX

I. System Modeling & Linearization

- **System Modeling & Linearization**



$\theta = \varphi_1$, $\theta_r = \varphi_1 + \varphi_2$

φ : Encoder Angle

$$J = m_1 l_1^2 + m_2 l_2^2 + I_1$$

$$J_r = I_2$$

$$T = \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} J_r \dot{\theta}_r^2$$

$$V = -(m_1 l_1 + m_2 l_2) g (1 - \cos\theta)$$

$$L = T - V = \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} J_r \dot{\theta}_r^2 + (m_1 l_1 + m_2 l_2) g (1 - \cos\theta)$$

I. System Modeling & Linearization

- **System Modeling & Linearization**

$$L = T - V = \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} J_r \dot{\theta}_r^2 + (m_1 l_1 + m_2 l_2) g (1 - \cos\theta)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = -\tau \quad \Rightarrow \quad J\ddot{\theta} - (m_1 l_1 + m_2 l_2) g \sin\theta = -\tau$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_r}\right) - \frac{\partial L}{\partial \theta_r} = \tau \qquad J_r \ddot{\theta}_r = \tau$$

$$\boxed{\begin{aligned} \ddot{\theta} &= \frac{(m_1 l_1 + m_2 l_2) g \sin\theta}{J} - \frac{\tau}{J} \\ \ddot{\theta}_r &= \frac{\tau}{J_r} \end{aligned}}$$

non-linear

I. System Modeling & Linearization

- **System Modeling & Linearization**

$$x_1 = \dot{\theta} \quad x_2 = \dot{\theta}_r, \quad x_3 = \theta \quad (\because \sin\theta \approx \theta)$$

$$\tau = k_u u, \quad k_u = \frac{K I_{max}}{10} = 0.005 \quad (|u| \le 10), \; F \approx 0$$

(Friction)

$$(m_1 l_1 + m_2 l_2) g = A$$

$$\dot{x}_1 = \frac{A}{J} x_3 - \frac{k_u}{J} u$$

$$\dot{x}_2 = \frac{k_u}{J_r} u$$

$$\boxed{\dot{x} = \begin{bmatrix} 0 & 0 & \frac{A}{J} \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} -\frac{k_u}{J} \\ \frac{k_u}{J_r} \\ 0 \end{bmatrix} u} \quad \text{linear}$$

I. System Modeling & Linearization

- **System Modeling**

$$m_{rotor} = 8.31 * 10^{-5}\ kg, \qquad m_{motor} = 265 * 10^{-3}\ kg$$

$$m_{pendulum} = 6.23 * 10^{-5}\ kg$$

$$J_{motor} = 4.5 * 10^{-6}\ kg \cdot m^2,$$

$$m_1 = m_{rotor} + m_{motor}, \qquad m_2 = m_{pendulum}$$

$$l_1 = 40 * 10^{-3}\ m, \qquad l_2 = 75 * 10^{-3}\ m$$

$$r = 35 * 10^{-3}\ m$$

$$\boldsymbol{J_r = J_{motor} + m_{rotor} * r^2 = 4.60 * 10^{-6}\ kg \cdot m^2}$$

$$J_{pendulum} = 7.20 * 10^{-8}\ kg \cdot m^2$$

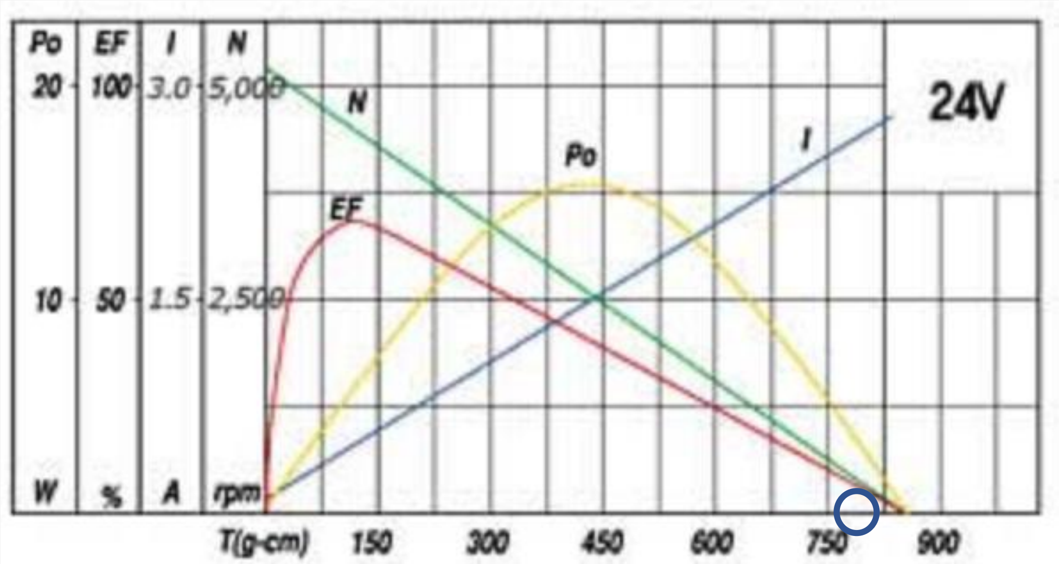
$$\boldsymbol{J = m_1 l_1^2 + m_2 l_2^2 + J_{pendulum} = 4.24 * 10^{-4} kg \cdot m^2}$$

모터 특징 Characteristics

Static friction	F_m	0.0028 Nm
Armature resistance	R_m	9.52 Ω
Motor torque constant	K_t	0.039 Nm / ampere
Back_emf	K_b	0.055 V/rad/sec
Motor inertia	J_m	4.5 * 10^{-6} Kg·m^2

I. System Modeling & Linearization

- **System Modeling**



$$\dot{x} = \begin{bmatrix} 0 & 0 & \frac{A}{J} \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} -\frac{k_u}{J} \\ \frac{k_u}{J_r} \\ 0 \end{bmatrix} u$$

$$T_{max} \approx 800\ g \cdot cm = 0.0784\ N \cdot m$$

$$u_{max} = 255$$

$$\boldsymbol{k_u = \frac{T_{max}}{255} = 2.941 * 10^{-4}}$$

$$\boldsymbol{A = (m_1 l_1 + m_2 l_2) g = 0.1041}$$

$$\boldsymbol{J = 4.24 * 10^{-4} kg \cdot m^2}$$

$$\boldsymbol{J_r = 4.60 * 10^{-6}\ kg \cdot m^2}$$

I. System Modeling & Linearization

- **System Modeling**

```
g = 9.81;
Mp = 6.2370e-05;
Mr = 8.3127e-05;
Mm = 265e-3;
Jm = 4.5e-6;

m_1 = Mr + Mm;
m_2 = Mp;

L_1 = 40e-3;
L_2 = 75e-3;
r = 35e-3;

Jp = 7.2058e-08;
Jr = Jm + Mr*r*r;
```

```
ku = 3.0745e-04;

A_eq = (m_1*L_1 + m_2*L_2)*g

J = m_1*L_1*L_1 + m_2*L_2*L_2 + Jp;

A = [0 0 A_eq/J; 0 0 0; 1 0 0]
B = [-ku/J;ku/Jr;0]
C = eye(3)
D_eq = [0 0 0]'

pole_K = [-3 -4 -5]*2;
K = place(A,B,pole_K)
```

```
A_eq = 0.1041
A = 3x3
         0         0  245.1138
         0         0         0
    1.0000         0         0

B = 3x1
   -0.7242
   66.8104
         0

C = 3x3
     1     0     0
     0     1     0
     0     0     1

D_eq = 3x1
     0
     0
     0
```

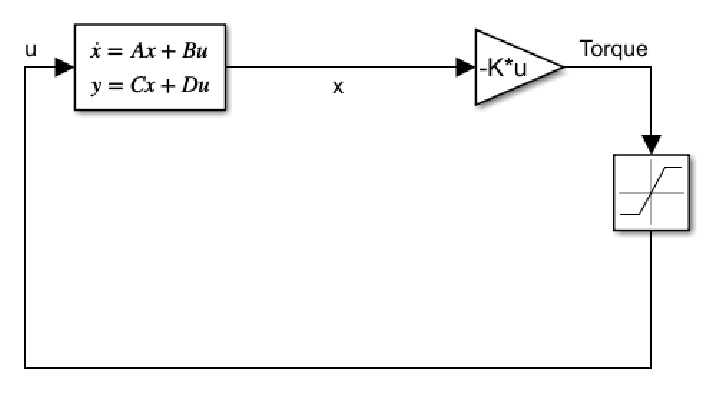
```
K = 1x3
   -35.8456   -0.0293 -598.0843
```

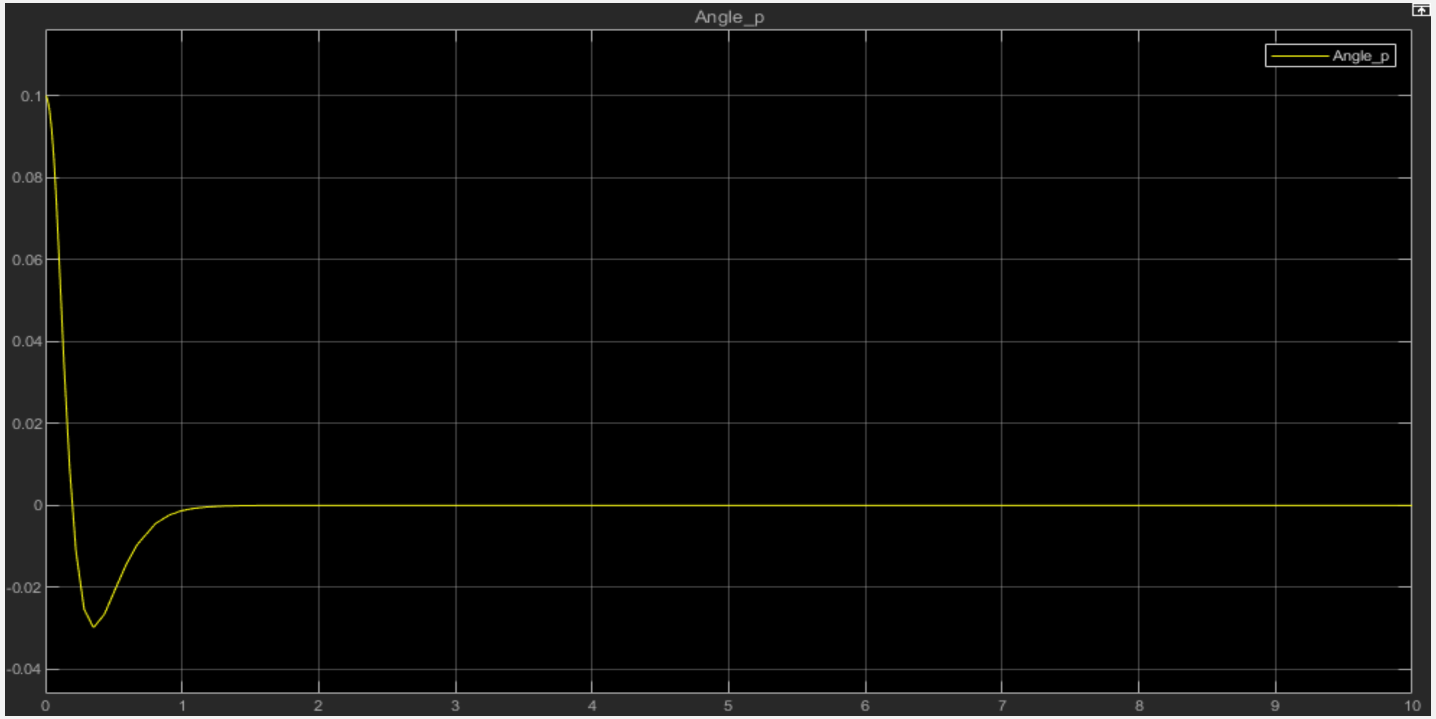
I. System Modeling & Linearization

- **Result – Full State Feedback**

```
pole_K = [-6 -8 -10];
K = place(A,B,pole_K)
```

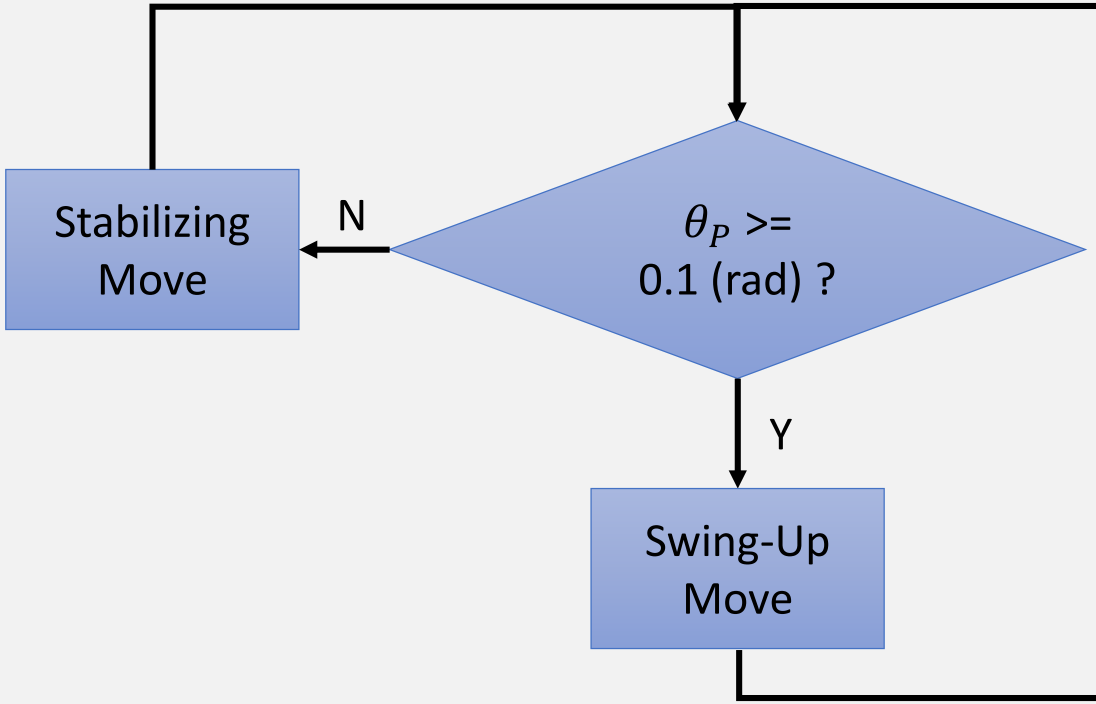
```
K = 1x3
10^3 x
     -0.5315    -0.0001    -5.0537
```





II. Porting State Feedback into Controller

- **Main Control Algorithm**



II. Porting State Feedback into Controller

- **Main Control Algorithm**

X1 : Pendulum velocity X2 : Motor velocity X3 : Pendulum Angle

```
K = 1x3
    -35.8456   -0.0293 -598.0843
```

```
double k1 = -35.8456276569173;
double k2 = -0.0293109303833444;
double k3 = -598.084291979568;
```

```
double StateSpace::get_u(double dt)
```

```
u1 = k1 * x1 * k_all / 255.0;
u2 = k2 * x2 * k_all / 255.0;
u3 = k3 * x3 * k_all / 255.0;
double u_ = u1 + u2 + u3;
```

```
if (u_ >= 1) u_ = 1;
else if (u_ <= -1) u_ = -1;

return u_;
```

```
if(fabs(cur_angle_P) >= 0.1)
{
    if(vel_P>0) u = 50;
    else u = -50;
}
else
{
    u = ss->get_u(dt_sec);
    target->u = u;
    u_float = (float)u;
}
```

Swing-Up Move

Stabilizing Move

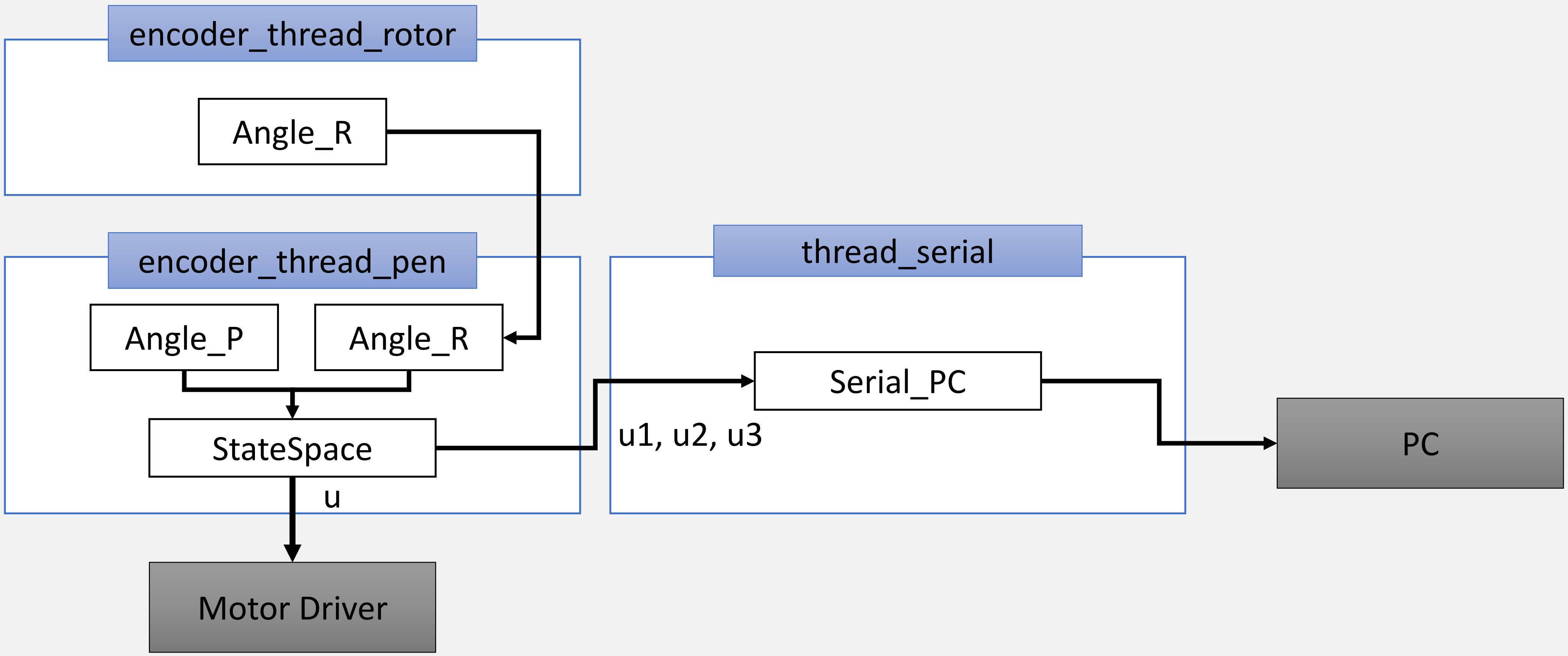
III. Code Explanation

- **Class**

  - thread_encoder : **엔코더 카운트 및 각도 계산, 모터 구동을 담당하는 스레드 클래스.**
  - thread_serial : **시리얼 플로팅을 담당하는 스레드 클래스.**
  - StateSpace : State Space **모델링 및 모터 명령 값 계산 클래스.** thread_encoder **내부에 내장됨.**

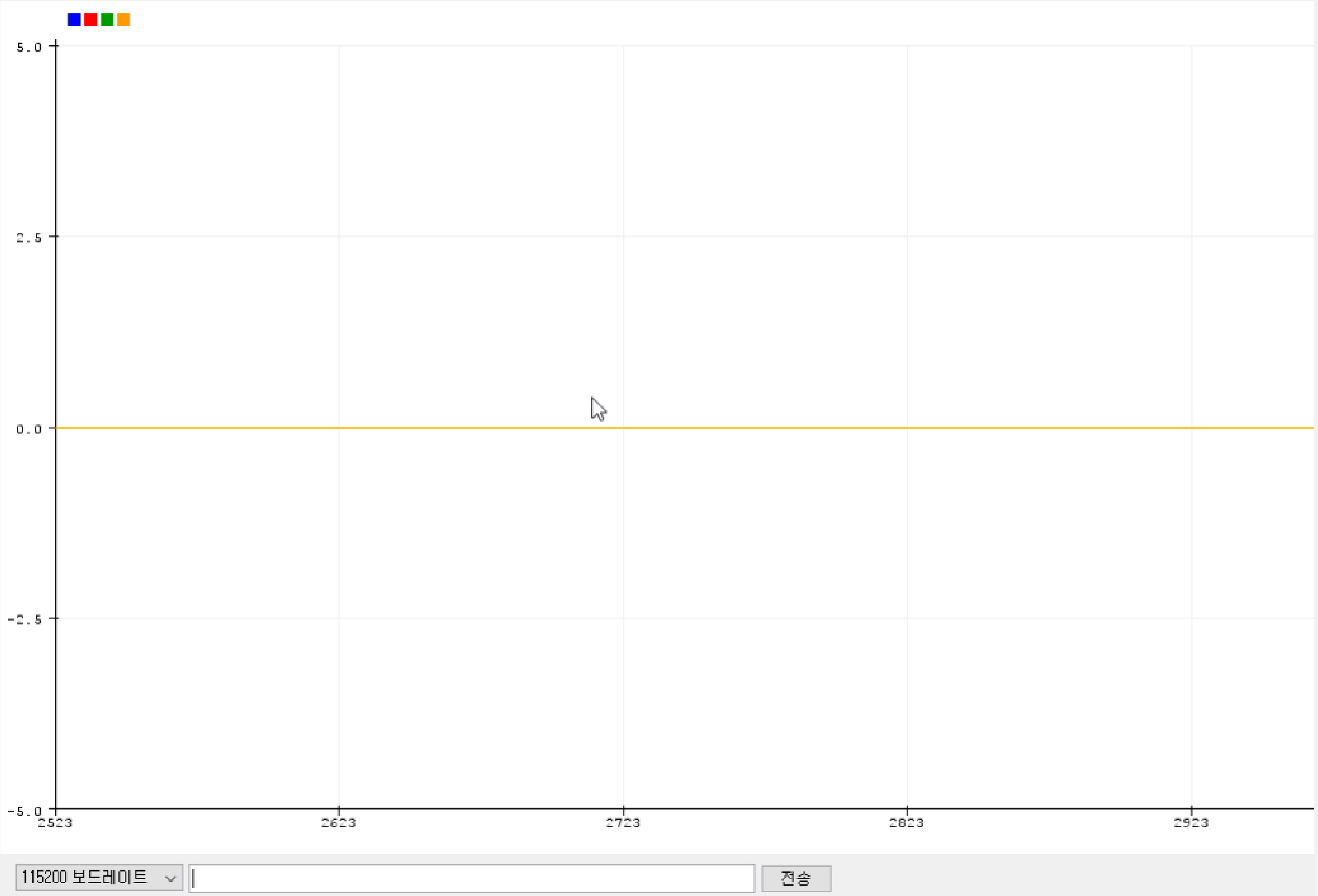
III. Code Explanation

- **Class**



IV. Result

- **Result**



IV. Result

- **Conclusion**

  - **시간이 부족해 세부적인 튜닝 및** Swing-Up **동작까지는 테스트를 못함**.
  - **정상상태 에러를 보정할 만한 추가적인 수단이 필요해보임**.