Robotics II
Day 10: Control Systems II

Chua Tzong Lin
B7TB1703
chuatl@mems.mech.tohoku.ac.jp

The equation of motion of the given system is given as,

$$\begin{pmatrix} 0 \\ \mathrm{f} \end{pmatrix} = \begin{pmatrix} \mathrm{I} + \mathrm{m}_\theta l_g{}^2 & \mathrm{m}_\theta l_g cos(\theta) \\ \mathrm{m}_\theta l_g cos(\theta) & \mathrm{m}_\mathrm{x} + \mathrm{m}_\theta \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{x} \end{pmatrix} - \begin{pmatrix} 0 \\ \mathrm{m}_\theta l_g sin(\theta)\,\dot{\theta}^2 \end{pmatrix} - \begin{pmatrix} \mathrm{m}_\theta g l_g sin(\theta) \\ 0 \end{pmatrix}$$

After Linearization and Laplace Transforming,

$$\begin{cases} \mathrm{F(s)} = (\mathrm{m_x} + \mathrm{m_\theta}) s^2 \mathrm{X(s)} + \mathrm{m_\theta} l_g s^2 \theta(\mathrm{s}) \\ 0 = \mathrm{m_\theta} l_g s^2 \mathrm{X(s)} + (\mathrm{I} + \mathrm{m_\theta} l_g{}^2) s^2 \theta(\mathrm{s}) - \mathrm{m_\theta} g l_g \theta(\mathrm{s}) \end{cases}$$

The Transfer function is given by the following equation,

$$\mathrm{G(s)} = \frac{\theta(\mathrm{s})}{\mathrm{F(s)}}$$

$$\mathrm{G(s)} = \frac{1}{-\left[\frac{\mathrm{I(m_x+m_\theta)}}{\mathrm{m_\theta} l_g} + \mathrm{m_x} l_g\right] s^2 + (\mathrm{m_x} + \mathrm{m_\theta})\mathrm{g}}$$

By utilizing the control theory for PD control,

$$\mathrm{F(s)} = -\mathrm{K_p} E(\mathrm{s}) - \mathrm{K_v} \mathrm{s} E(\mathrm{s})$$

$$\mathrm{F(s)} = -\mathrm{K_p}[R(s) - \theta(\mathrm{s})] + \mathrm{K_v} \mathrm{s}\dot{\theta}(\mathrm{s})$$

Taking R(s) as zero for out assumption,

$$\mathrm{F(s)} = \mathrm{K_p}\theta(\mathrm{s}) + \mathrm{K_v}\mathrm{s}\theta(\mathrm{s})$$

The combing with the transfer function resulting system is shown as the following equation,

$$\left\{\left[\frac{\mathrm{I(m_x + m_\theta)}}{\mathrm{m_\theta} l_g} + \mathrm{m_x} l_g\right] s^2 + \mathrm{K_v}\,\mathrm{s} + \left[\mathrm{K_p} - (\mathrm{m_x} + \mathrm{m_\theta})\mathrm{g}\right]\right\} \theta(\mathrm{s}) = 0$$

The poles of the system is given by the following equation,

$$s = \frac{-\mathrm{K_v} \pm \mathrm{j}\sqrt{4\left[\frac{\mathrm{I(m_x+m_\theta)}}{\mathrm{m_\theta} l_g} + \mathrm{m_x} l_g\right]\left[\mathrm{K_p} - (\mathrm{m_x} + \mathrm{m_\theta})\mathrm{g}\right] - \mathrm{K_v}{}^2}}{2\left[\frac{\mathrm{I(m_x+m_\theta)}}{\mathrm{m_\theta} l_g} + \mathrm{m_x} l_g\right]}$$

1. “Stability Limit” behavior using PD control.

   For a system operating at the stability limit, the K_v gain is zero, while the K_p gain is that satisfies the condition

$$K_p - (m_x + m_\theta)g > 0$$

   Under this condition, the system behaves similarly to an un-damped mass spring system, which will oscillate about a given point.

   The selected K_p gain values are:

   - K_p = 15
     The pole values are given as:
     - s = 0.0 + 4.963j
     - s = 0.0 - 4.963j

   - K_p = 30
     The pole values are given as:
     - s = 0.0 + 8.073j
     - s = 0.0 - 8.073j

   The implemented python script is included under the filename [InvertedPendulum_odeint_Stability_Limit_1.py]() and [InvertedPendulum_odeint_Stability_Limit_2.py]().
   The results are shown in figures 1.1 and 1.2.

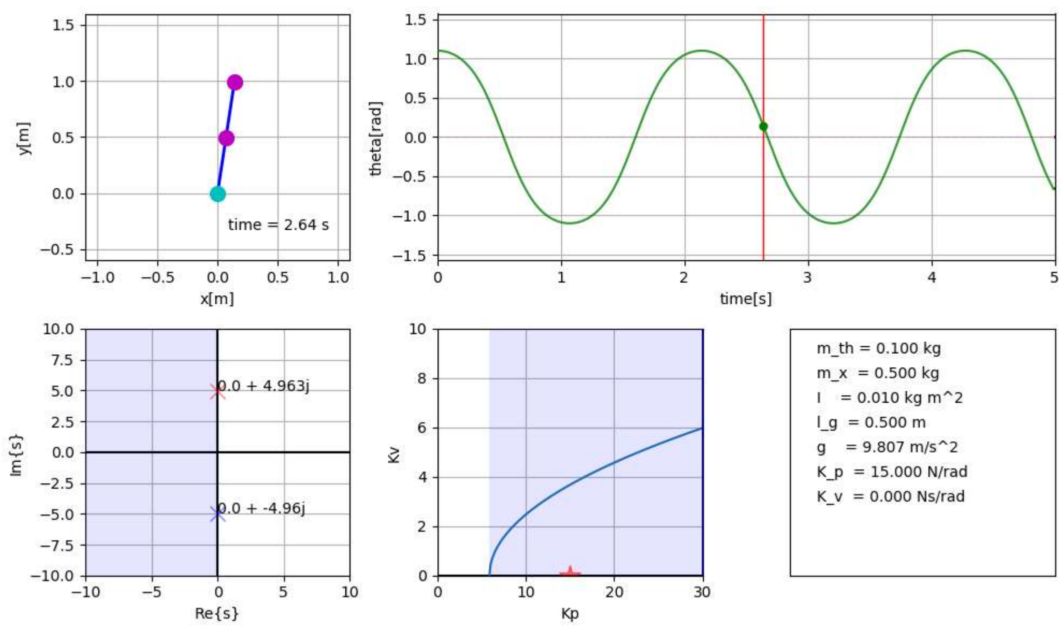

Figure 1.1 “Stability Limit” behavior with K_p gain of 15

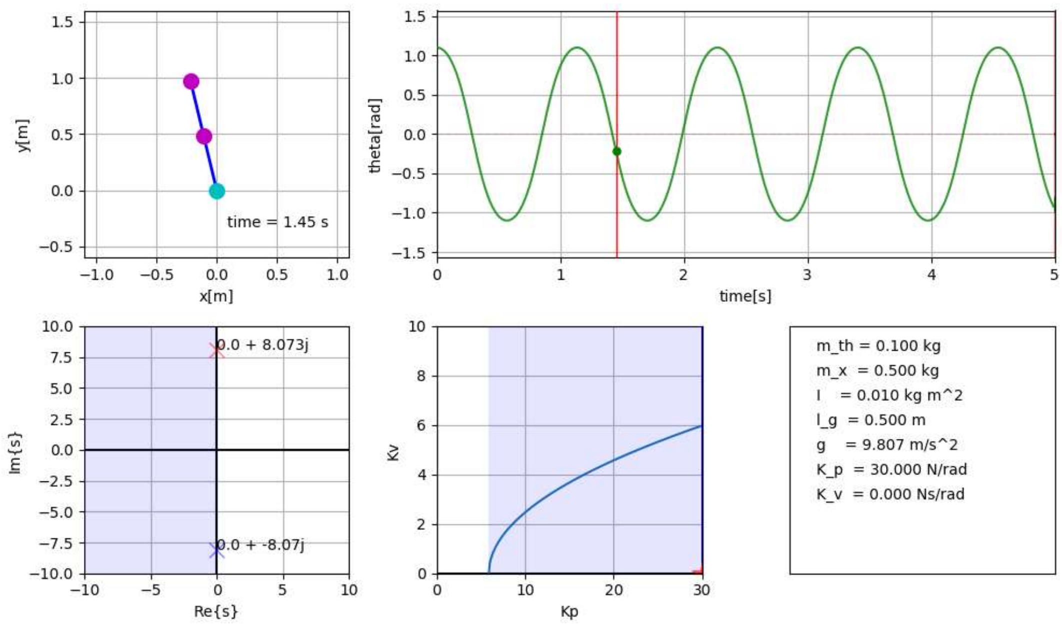

Figure 1.2 “Stability Limit” behavior with K_p gain of 30

2. “Critically Damped” behavior using PD control.
   For a critically damped system, the K_v and K_p gain is selected to satisfy the following condition.
   $$4\left[\frac{I(m_x + m_\theta)}{m_\theta l_g} + m_x l_g\right]\left[K_p - (m_x + m_\theta)g\right] - {K_v}^2 = 0$$
   The selected K_p and K_v gain values are:
   - K_p = 15, K_v = 3.673
     The pole values are given as:
     - s = -4.96 + 0.0j

   The implemented python script is included under the filename InvertedPendulum_odeint_Critically_Damped.py.
   The results are shown in figure 2.1.

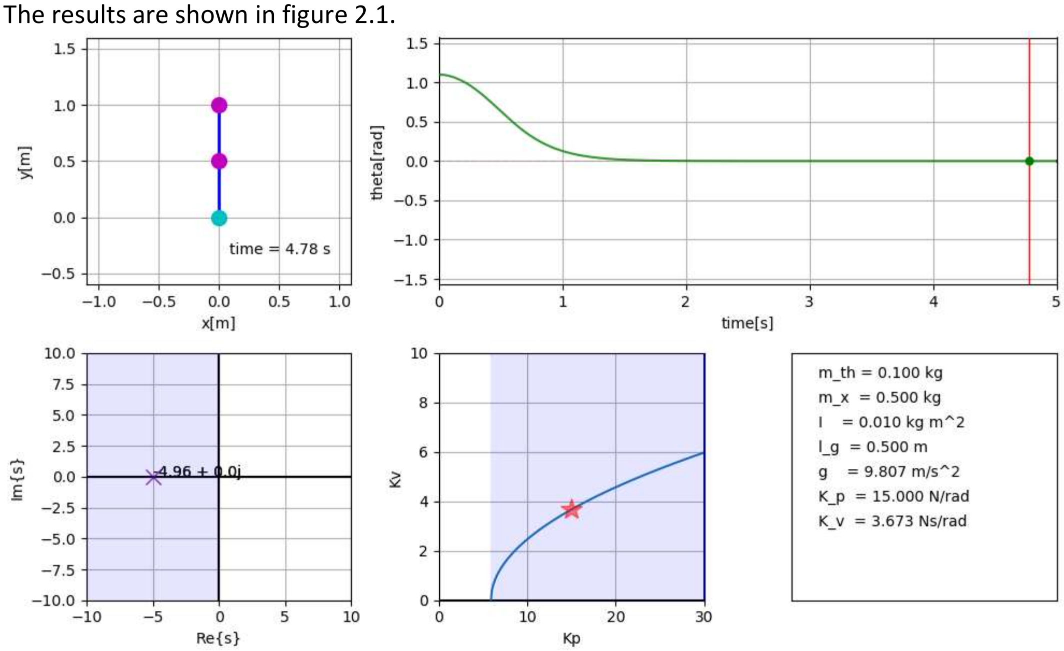

Figure 2.1 “Critically Damped” behavior with K_p gain of 15 and K_v gain of 3.673

3. “Over-damped” behavior using PD control.

   An over-damped system realized when the K_p and K_v gains selected satisfies the following condition.

$$4\left[\frac{\mathrm{I}(\mathrm{m_x}+\mathrm{m_\theta})}{\mathrm{m_\theta} l_g}+\mathrm{m_x} l_g\right]\left[\mathrm{K_p}-(\mathrm{m_x}+\mathrm{m_\theta})\mathrm{g}\right]-\mathrm{K_v}^2<0$$

   The selected K_p and K_v gain values are:

   - K_p = 15, K_v = 9
     The pole values are given as:
     - s = -23.2 + 0.0j
     - s = -1.05 + 0.0j

   The implemented python script is included under the filename InvertedPendulum_odeint_Overdamped.py.
   The results are shown in figure 3.1.

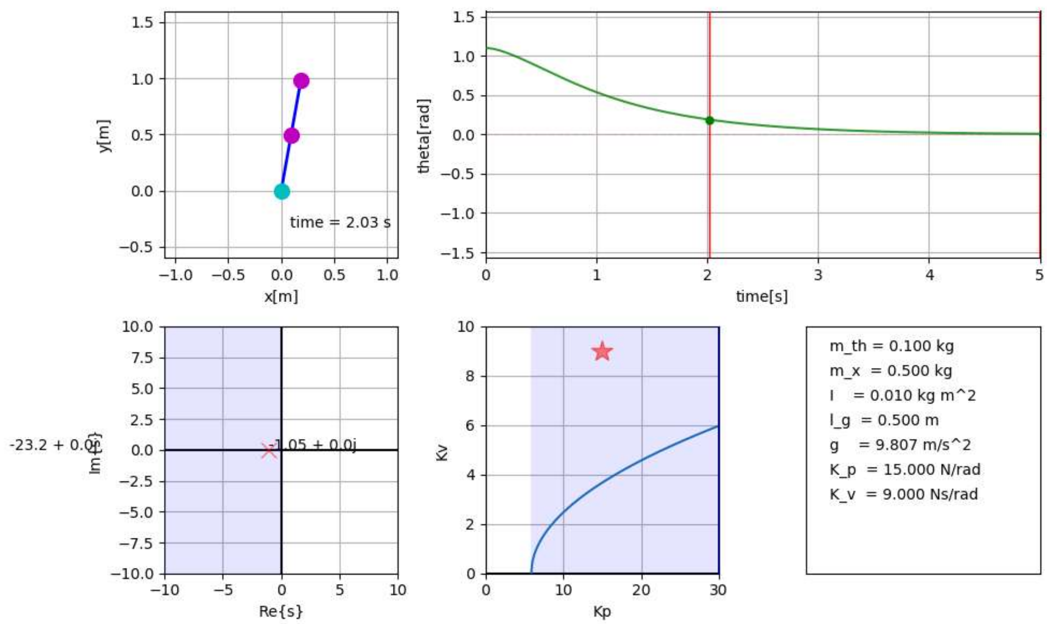

Figure 3.1 “Over-damped” behavior with K_p gain of 15 and K_v gain of 9

4. “Under-damped” behavior using PD control.

   An over-damped system realized when the K_p and K_v gains selected satisfies the following condition.

$$4\left[\frac{\mathrm{I}(\mathrm{m_x}+\mathrm{m_\theta})}{\mathrm{m_\theta} l_g}+\mathrm{m_x} l_g\right]\left[\mathrm{K_p}-(\mathrm{m_x}+\mathrm{m_\theta})\mathrm{g}\right]-\mathrm{K_v}^2>0$$

   The selected K_p and K_v gain values are:

   - K_p = 15, K_v = 1
     The pole values are given as:
     - s = 1.35 + 4.776j
     - s = 1.35 - 4.776j

The implemented python script is included under the filename [InvertedPendulum_odeint_Underdamped.py]().

The results are shown in figure 4.1.

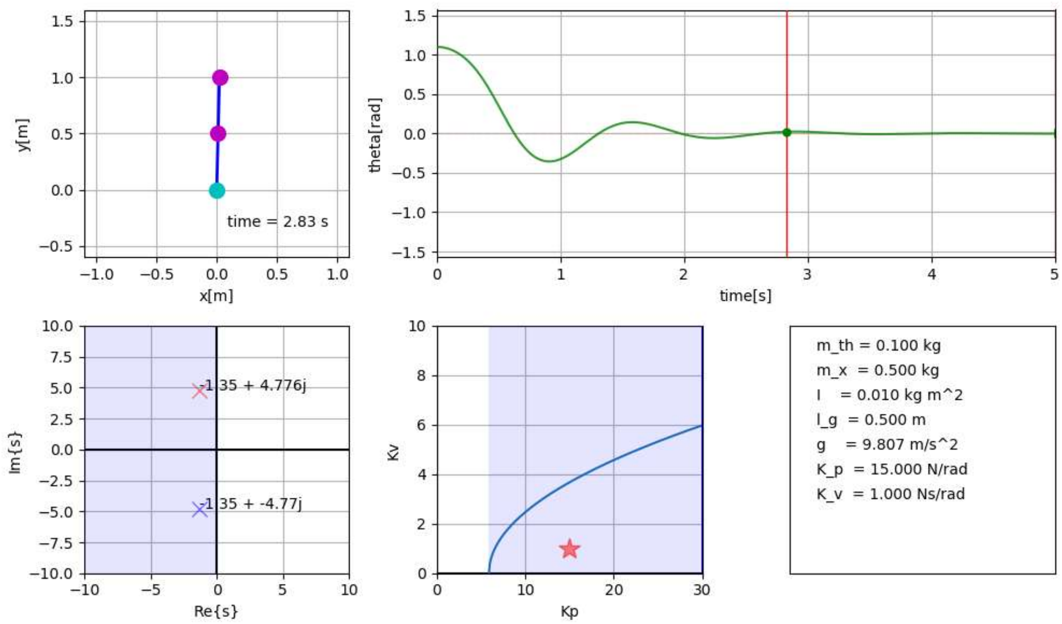

Figure 4.1 “Over-damped” behavior with K_p gain of 15 and K_v gain of 1