電子制御工学実験報告書

実験題目　：数値シミュレーション

報告者　　：4 年 32 番　平田 蓮

提出日　　：2020 年 11 月 6 日

実験日　　：2020 年 10 月 15 日, 10 月 22 日, 10 月 29 日, 11 月 5 日

実験班　　：

共同実験者：

※指導教員記入欄

評価項目	配点	一次チェック ・　・	二次チェック ・　・
記載量	20		
図・表・グラフ	20		
見出し，ページ番号，その他体裁	10		
その他の減点	–		
合計	50		

コメント：

1　目的

この実験では, 実際の系をモデルに置き換えてシミュレーションすることで挙動を調べる. しかし, シミュレーションがうまくいくとは限らないので, シミュレーションの結果を鵜呑みにせず, その結果の妥当性を検証して確かめることが大切である.

この実験を通して, 様々な方向からのシミュレーション結果の解析方法を身に付けることを目的とする.

2　バネ-マス-ダンパ系のシミュレーション

この節では, 実験テキスト [1] の Fig.25 に示されているバネ-マス-ダンパ系のシミュレーションを行う.

2.1　系の運動方程式

まずはシミュレーションモデルを構成するために系の運動方程式を導出する.

図より, 運動方程式は次のようになる.

$$m\ddot{x}(t) = f(t) - kx(t) - c\dot{x}(t)$$

次小節でモデルを構築するために左辺を $\ddot{x}(t)$ の形にすると,

$$\ddot{x}(t) = \frac{f_0}{m}u(t) - \frac{k}{m}x(t) - \frac{c}{m}\dot{x}(t) \tag{1}$$

となる.

2.2　シミュレーションモデルの構築

式 (1) に基づいて構築したモデルを図 1 に示す.

総和ブロックの出力を $\ddot{x}(t)$ と考え, $\frac{1}{s}$ のブロック二つを通して $x(t)$ まで積分処理を施している. よって, 一回だけ積分をした点は $\dot{x}(t)$ である.

運動方程式に基づき, $\dot{x}(t), x(t)$ それぞれに $\frac{c}{m}, \frac{k}{m}$ を掛けて負帰還をしている.

また, 物体に加わり続ける外力として, 大きさ f_0 のステップ入力を総和ブロックに加えている.

最後に, $x(t)$ と $\dot{x}(t)$ を出力として得たいので, それらを出力ブロックに入力している.

2.3　パラメータの計算

シミュレーションをする際の定数として各値を計算する. 各値は次のように与えられている.

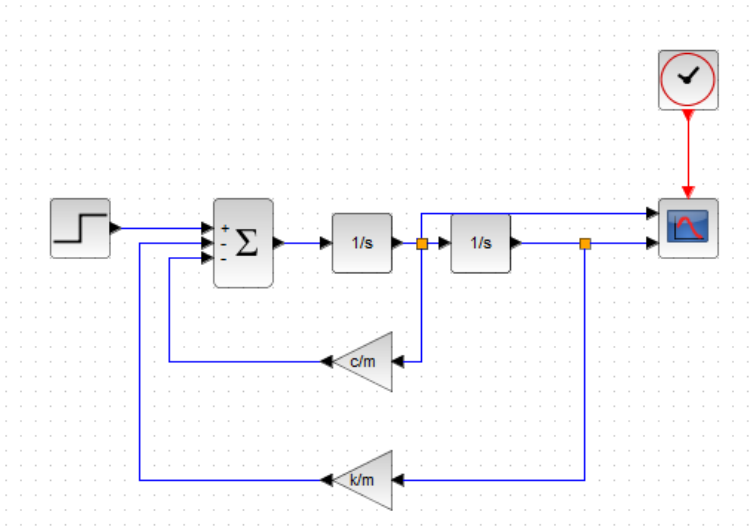

図 1　バネ-マス-ダンパ系のシミュレーションモデル

$$
\begin{aligned}
m &= \text{出席番号} \times 100 \text{ kg} \\
k &= \frac{|150 + (\text{出席番号} - 25)^2 \times 50|}{10} \text{ kg/s}^2 \\
c &= 650 - (\text{出席番号} - 25)^2 \text{ kg/s} \\
f_0 &= 300 \text{ kg} \cdot \text{cm/s}^2
\end{aligned}
$$

私の出席番号は 32 であるので, これを代入して各値を計算すると,

$$
\begin{aligned}
m &= 32 \times 100 = 3200 \text{ kg} \\
k &= \frac{|150 + (32 - 25)^2 \times 50|}{10} = 260 \text{ kg/s}^2 \\
c &= 650 - (32 - 25)^2 = 601 \text{ kg/s}
\end{aligned}
$$

を得られる.

これらを式 (1) に代入すると,

$$\ddot{x}(t) \approx 0.09375u(t) - 0.08125x(t) - 0.18781\dot{x}(t)$$

となる.

2.4　シミュレーション結果

構築したモデルと求めたパラメータを使いシミュレーションを行った結果を図 2 に示す.

上のグラフは $v(t)(= \dot{x}(t))$, 下のグラフは $x(t)$ を示している.

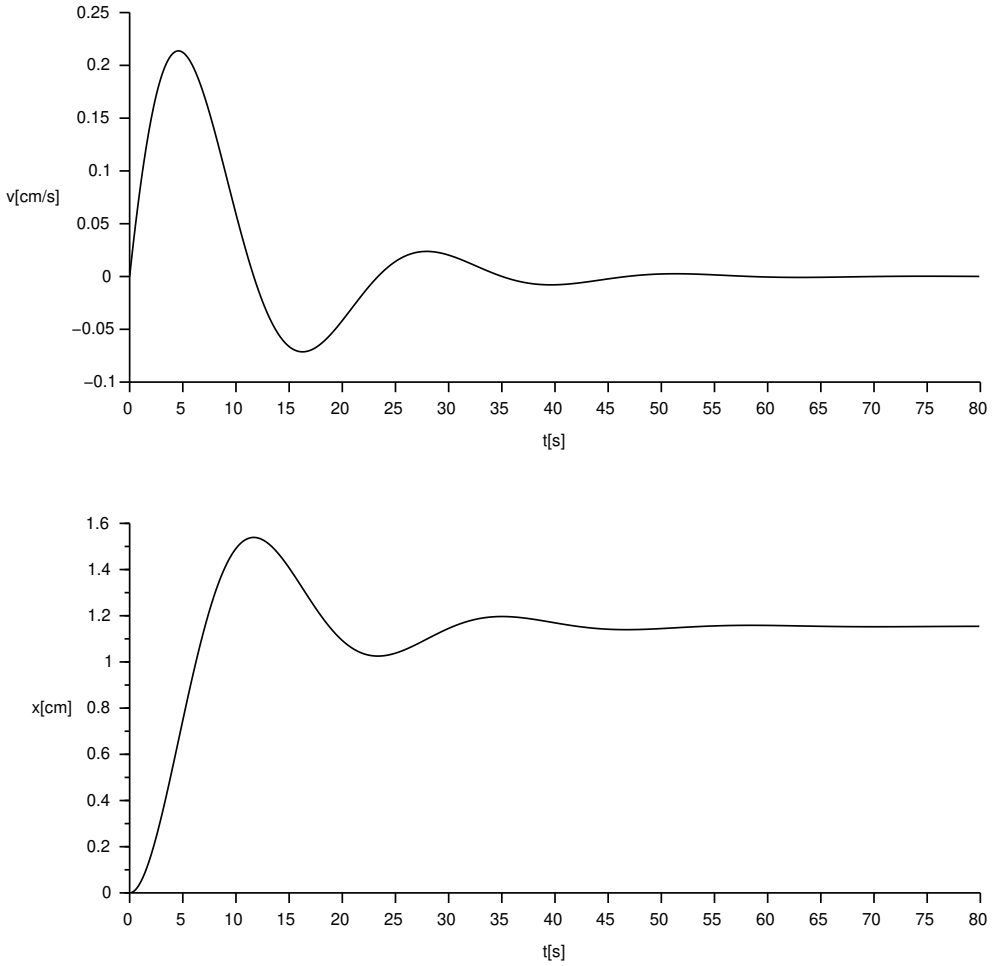

図 2　バネ-マス-ダンパ系のシミュレーション結果

2.5　結果の解析

前小節で行ったシミュレーションの結果の妥当性を調べるために解析を行う.

2.5.1　python を使ったシミュレーション

まずはじめに, 本質的には同じことであるが, Scilab ではない別のソフトウェアを使ってシミュレーションを行ってみる. 今回はプログラミング言語 Python を使ってシミュレーションする.

作成したコードをリスト 1 に示す.

リスト 1　simulate.py

```
1 import numpy as np
2 from scipy.integrate import odeint
3 import matplotlib.pyplot as plt
4
```

```
5 def f(x, t):
6     return (x[1], 0.09375 - 0.08125 * x[0] - 0.18781 * x[1])
7
8 x0 = [0, 0]
9 t = np.linspace(0, 80, 1000)
10 y = odeint(f, x0, t)
11
12 fig = plt.figure()
13 s1 = fig.add_subplot(2, 1, 1)
14 s2 = fig.add_subplot(2, 1, 2)
15
16 s2.plot(
17     t, y[:,[0]],
18     color='k'
19 )
20 s2.set_xlabel('t[s]')
21 s2.set_xlim([0, 80])
22 s2.set_ylabel('x[cm]', rotation=0)
23 s2.set_ylim([0, 1.6])
24
25 s1.plot(
26     t, y[:,[1]],
27     color='k'
28 )
29 s1.set_xlabel('t[s]')
30 s1.set_xlim([0, 80])
31 s1.set_ylabel('v[cm/s]', rotation=0)
32 s1.set_ylim([-0.1, 0.25])
33
34 plt.show()
```

scipy というライブラリの `integrate.odeint()` という関数を使用して値をシミュレーションをする.

シミュレーション結果を図 3 に示す.

図 2 と比べてみても, 同様の結果が出ていることがわかる.

2.5.2 初期値, 最終値の定理

$\mathcal{L}[f(t)] = F(s)$ として初期値の定理, 最終値の定理を式 (2), (3) に示す.

$$\lim_{s\to\infty} sF(s) = \lim_{t\to+0} f(t) \tag{2}$$

$$\lim_{s\to 0} sF(s) = \lim_{t\to\infty} f(t) \tag{3}$$

これらを使うために式 (1) の両辺をラプラス変換する. 初期条件を $x(0) = v(0) = 0$ とすると,

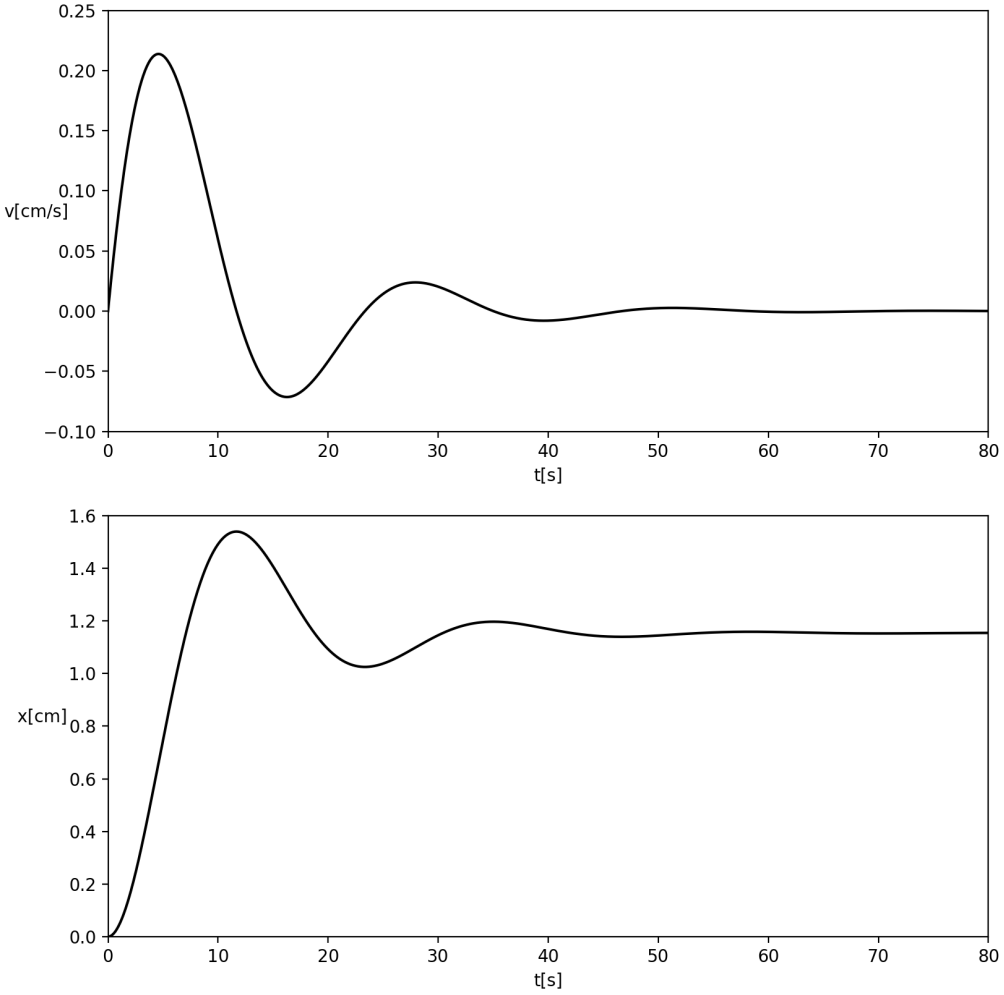

図 3　バネ-マス-ダンパ系の Python によるシミュレーション結果

$$s^2X(s) = \frac{f_0}{m}\frac{1}{s} - \frac{k}{m}X(s) - \frac{c}{m}sX(s)$$
$$\leftrightarrow X(s) = \frac{f_0}{s(ms^2 + cs + k)}$$

2.3 で計算したパラメータを代入して,

$$X(s) = \frac{300}{s(3200s^2 + 601s + 260)} \tag{4}$$

となる.

まず初期値の定理より,

$$\lim_{t\to+0} x(t) = \lim_{s\to\infty} sX(s) = \lim_{s\to\infty} \frac{300}{3200s^2+601s+260} = 0$$

次に最終値の定理より,

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s) = \lim_{s\to 0} \frac{300}{3200s^2+601s+260} \approx 1.154$$

また, $\mathcal{L}[v(t)] = sX(s)$ であるので,

$$\lim_{t\to+0} v(t) = \lim_{s\to\infty} s^2X(s) = \lim_{s\to\infty} \frac{300}{3200s+601+260s^{-1}} = 0$$
$$\lim_{t\to\infty} v(t) = \lim_{s\to 0} s^2X(s) = \lim_{s\to 0} \frac{300}{3200s+601+260s^{-1}} = 0$$

となる. 図 2, 3 をみると, これらの値がみて取れるため, 少なくとも初期値と最終値のシミュレーションは正しいことがわかる.

2.5.3 運動方程式の解

次に, 運動方程式の解を求めることで $x(t), v(t)$ を直接グラフ化してみる.

まず, 式 (4) の両辺を逆ラプラス変換して $x(t)$ を求める.

$$\begin{aligned} x(t) &= \mathcal{L}^{-1}\left[\frac{300}{s(3200s^2+601s+260)}\right] \\ &\approx 1.154\mathcal{L}^{-1}\left[\frac{1}{s} - \frac{s+0.0939}{(s+0.0939)^2+0.2691^2}\right] \\ &= 1.154(1-e^{-0.0939t}\cos 0.2691t) \end{aligned} \tag{5}$$

次に, $sX(s)$ をラプラス逆変換して $v(t)$ を得る.

$$\begin{aligned} v(t) &= \mathcal{L}^{-1}\left[\frac{300}{3200s^2+601s+260}\right] \\ &\approx 0.3483e^{-0.0939t}\sin 0.2691t \end{aligned} \tag{6}$$

式 (5), (6) をグラフ化したものを図 4 に示す.

グラフより, シミュレーションが正確であったと言える.

ここで, $v(t) = \dot{x}(t)$ であることに着目すると, $sX(s)$ をラプラス変換するのではなく, $x(t)$ を t で微分することで $v(t)$ を得られるのではないかと考えることができる. そのように求める $v(t)$ を $v_f(t)$ とすると, 式 (7) のようになる.

$$\begin{aligned} v_f(t) &\approx 1.154e^{-0.0939t}(0.2691\sin 0.2691t + 0.0939\cos 0.2691t) \\ &\approx e^{-0.0939t}(0.3105\sin 0.2691t + 0.1084\cos 0.2691t) \end{aligned} \tag{7}$$

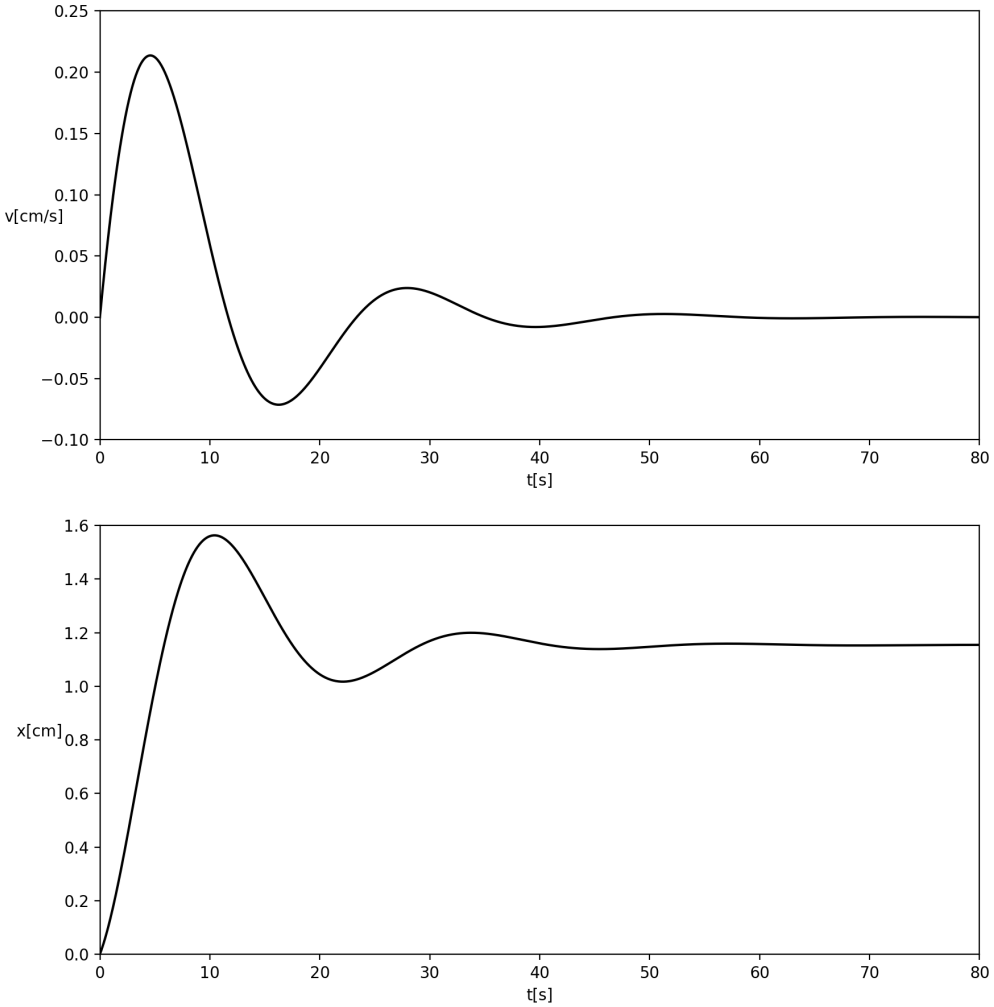

図 4　$x(t), v(t)$ のプロット

これを式 (6) とプロットして比較してみると, 図 5 のようになる.

グラフより, $v_f(t)$ は $v(t)$ の振幅を少しだけ大きくし, 位相を少しだけ早めたものであると考えられる.

この差は初期条件 $v(0) = 0$ を考慮しているか, また, $t < 0$ の領域を考慮しているかどうかによって現れていると考えられる. 実際に, $v_f(t)$ が 0 となる t では $x(t)$ が極値を迎えているが, $v(t) = 0$ となる点はずれている.

これは以下の計算によって確かめられる.

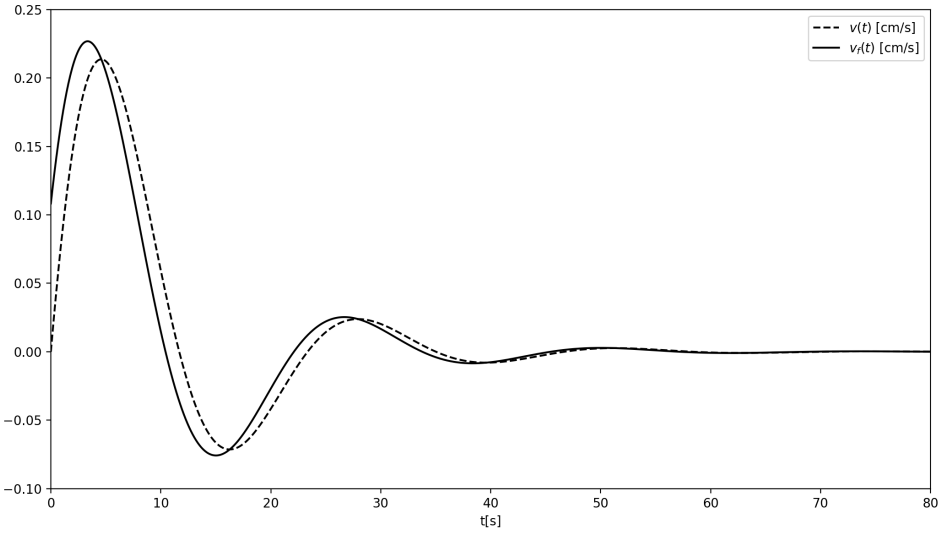

図 5　$v(t), v_f(t)$ のプロット

$$
\begin{aligned}
v(t) &\approx 0.3483e^{-0.0939t}\sin 0.2691t = 0 \\
\leftrightarrow \sin 0.2691t &= 0 \\
\leftrightarrow 0.2691t &= n\pi \ (n \in \mathbb{Z}) \\
\leftrightarrow t &\approx 11.674n
\end{aligned}
$$

$$
\begin{aligned}
v_f(t) &\approx e^{-0.0939t}(0.3105\sin 0.2691t + 0.1084\cos 0.2691t) = 0 \\
\leftrightarrow \tan 0.2691t &= -0.3491 \\
\leftrightarrow t &\approx n\pi - 1.2481 \ (n \in \mathbb{Z}) \\
\leftrightarrow t &\approx 11.674n - 1.2481
\end{aligned}
$$

$n = 1$ の点について, $v(t) \approx 0 \leftrightarrow t = 11.674$, $v_f(t) \approx 0 \leftrightarrow t = 10.426$ である.

この範囲に絞って $x(t)$ をプロットしてみると, 図 6 となる.

図では, $v_f(t) = 0$ となる点である $t = 10.426$ 付近で $x(t)$ が極値を迎えている. このこととシミュレーション結果より, 数学的には $v_f(t)$ が正しいが, 実際の系では $v(t)$ となることが予想される.

3　天井クレーンの制御シミュレーション

この節では, 実験テキスト [1] の Fig.28 に示されている天井クレーンのシミュレーションを行う.

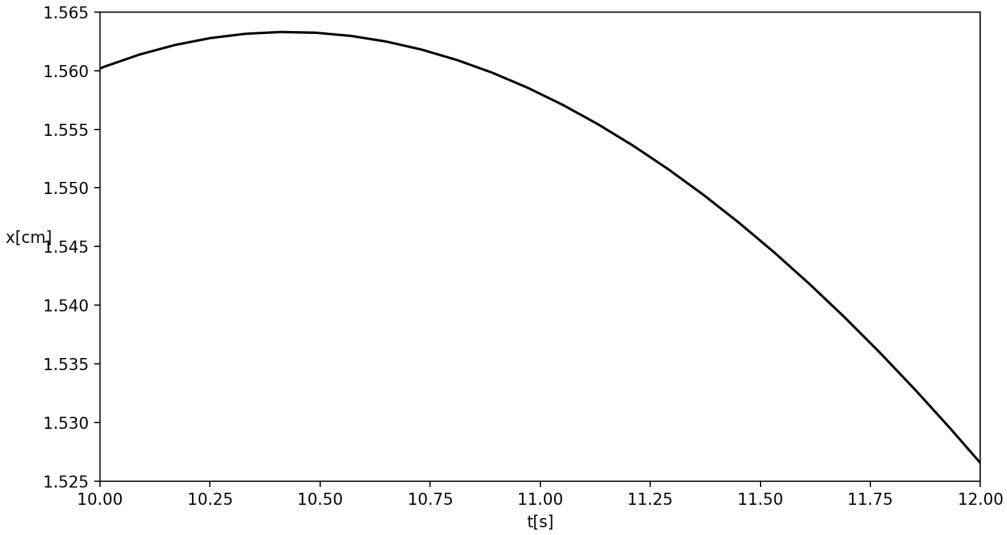

図 6　$x(t)$ のプロット $(10 \leq t \leq 12)$

3.1　吊り荷部の運動方程式

バネ-マス-ダンパ系のときと同様に, まずは吊り荷部の運動方程式をラグランジュ法を用いて導出する.

吊り荷の x 方向の変位を $x_l(t), y$ 方向の変位を $y_l(t)$ とすると,

$$
\begin{aligned}
x_l(t) &= x(t) + l\sin\theta(t) \\
y_l(t) &= l\cos\theta(t)
\end{aligned}
$$

となる. これらを t で微分することでそれぞれの方向の速度を出すことができる.

$$
\begin{aligned}
\dot{x_l}(t) &= \dot{x}(t) + l\dot{\theta}(t)\cos\theta(t) \\
\dot{y_l}(t) &= -l\dot{\theta}(t)\sin\theta(t)
\end{aligned}
$$

吊り荷の速度を $v_l(t)$ とすると, 以下のように表せる.

$$
\begin{aligned}
v_l(t)^2 &= \dot{x_l}(t)^2 + \dot{y_l}(t)^2 \\
&= \dot{x}(t)^2 + 2l\dot{x}(t)\dot{\theta}(t)\cos\theta(t) + l^2\dot{\theta}(t)^2
\end{aligned}
$$

これより, 吊り荷の運動エネルギーは,

$$
\frac{1}{2}mv_l(t)^2 = \frac{1}{2}m\{\dot{x}(t)^2 + 2l\dot{x}(t)\dot{\theta}(t)\cos\theta(t) + l^2\dot{\theta}(t)^2\}
$$

である．また, トロリーは x 方向にのみ運動をするので, その運動エネルギーは $\frac{1}{2}M\dot{x}(t)^2$ である．これらの和を取り, 系の運動エネルギー T は,

$$T = \frac{1}{2}m\{\dot{x}(t)^2 + 2l\dot{x}(t)\dot{\theta}(t)\cos\theta(t) + l^2\dot{\theta}(t)^2\} + \frac{1}{2}M\dot{x}(t)^2 \tag{8}$$

となる．

吊り荷の最下点を基準点とすると, 吊り荷の位置エネルギー U は,

$$U = mg\{l - y_l(t)\} = mgl\{1 - \cos\theta(t)\} \tag{9}$$

である．

また, 散逸エネルギー F は次のようにできる．

$$F = \frac{1}{2}\mu\dot{\theta}(t)^2 \tag{10}$$

ここで, ラグランジュの運動方程式方程式は次のように定義される．

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} + \frac{\partial F}{\partial \dot{q}_i} = \tau_i \tag{11}$$

τ_i は物体に加わる外力であるが, 今回は 0 である．また, q_i は一般化座標といわれ, 今回は $q_i = \theta(t)$ として計算をする．

式 (8) より,

$$\begin{aligned}
\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}} &= \frac{d}{dt}\frac{\partial}{\partial \dot{\theta}}\left(\frac{1}{2}m\{\dot{x}(t)^2 + 2l\dot{x}(t)\dot{\theta}(t)\cos\theta(t) + l^2\dot{\theta}(t)^2\} + \frac{1}{2}M\dot{x}(t)^2\right) \\
&= \frac{d}{dt}ml\{\dot{x}(t)\cos\theta(t) + l\dot{\theta}(t)\} \\
&= ml\{\ddot{x}(t)\cos\theta(t) - \dot{x}(t)\dot{\theta}(t)\sin\theta(t) + l\ddot{\theta}(t)\} \qquad (12) \\
\frac{\partial T}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(\frac{1}{2}m\{\dot{x}(t)^2 + 2l\dot{x}(t)\dot{\theta}(t)\cos\theta(t) + l^2\dot{\theta}(t)^2\} + \frac{1}{2}M\dot{x}(t)^2\right) \\
&= -ml\dot{x}(t)\dot{\theta}(t)\sin\theta(t) \qquad (13)
\end{aligned}$$

式 (9) より,

$$\begin{aligned}
\frac{\partial U}{\partial \theta} &= \frac{\partial}{\partial \theta}mgl\{1 - \cos\theta(t)\} \\
&= mgl\sin\theta(t) \qquad (14)
\end{aligned}$$

式 (10) より,

$$\begin{aligned}
\frac{\partial F}{\partial \dot{\theta}} &= \frac{\partial}{\partial \dot{\theta}}\frac{1}{2}\mu\dot{\theta}(t)^2 \\
&= \mu\dot{\theta}(t) \qquad (15)
\end{aligned}$$

となる.

式 (11) に式 (12), (13), (14), (15), $\tau_i = 0$ を代入して, 運動方程式は,

$$ml\ddot{x}(t)\cos\theta(t) + ml^2\ddot{\theta}(t) + \mu\dot{\theta(t)} + mgl\sin\theta(t) = 0 \tag{16}$$

となる.

3.2 パラメータの計算

シミュレーションをする際の定数として各値を計算する. 各値は次のように与えられている.

$$\begin{aligned} m &= 0.5 + \left(\frac{\text{出席番号} - 25}{10}\right)^2 \ \text{kg} \\ l &= 0.16 + \text{出席番号} \times 0.04 \ \text{m} \\ \mu &= 1.22 \times 10^{-3} \ \text{kg} \cdot \text{m}^2/\text{s} \end{aligned}$$

私の出席番号は 32 であるので, これを代入して各値を計算すると,

$$\begin{aligned} m &= 0.5 + \left(\frac{32 - 25}{10}\right)^2 = 0.99 \ \text{kg} \\ l &= 0.16 + 32 \times 0.04 = 1.44 \ \text{m} \end{aligned}$$

を得られる.

これらを式 (16) に代入すると,

$$1.426\ddot{x}(t)\cos\theta(t) + 2.053\ddot{\theta}(t) + 1.22 \times 10^{-3}\dot{\theta(t)} + 13.97\sin\theta(t) = 0$$

となる.

3.3 PID 制御器の設計とシミュレーション結果

PID 制御とは目標値と制御量から操作量を決める制御方法である. 今回は PID 制御器を使ってシミュレーションを行う.

PID 制御器が扱う制御則は三つあり, 比例 (Proportional) 動作, 積分 (Integral) 動作, 微分 (Derivative) 動作である. それぞれ英語の頭文字が P, I, D に対応している.

比例動作は, 観測値と目標値の偏差に比例する制御則である. 差が大きければ制御量も大きくし, そのまた逆も然りとする.

積分動作は偏差の積分に比例する制御則である. これにより, 比例動作だけでは残ってしまう一定の偏差を取り除くことができる.

微分動作は偏差の微分に比例する制御則である．これにより, 偏差の増減の動向を操作量に反映することができる．

今回はサポートページ [2] にて配布されているシミュレーションモデルを使用して実験を進める．

設計仕様として以下の二つが定められている．

- とろりは目標位置の印加後 5s 以内に整定し, オーバーシュートは 5% 以内とする．
- 吊り荷の揺れ角は $\pm 7^\circ$ 以内とする．

各パラメータでシミュレーションをした際のトロリの $x-t$ グラフと吊り荷の $\theta-t$ グラフをみながらこれらを満たすようにトロリ, 吊り荷それぞれの PID ゲインを定めていく．なお, モデル内で与えられている初期値はトロリの P ゲインが 3 で, 他は全て 0 である．以下, トロリの PID ゲインを P_t, I_t, D_t, 吊り荷の PID ゲインを P_l, I_l, D_l とする．

3.3.1　P ゲインの調整

まずはじめに, P ゲインの調整を行う．二つの P ゲインが動作にどのような影響を及ぼすのかをみるために, P ゲインのみを変化させながらシミュレーション結果をみてみる．

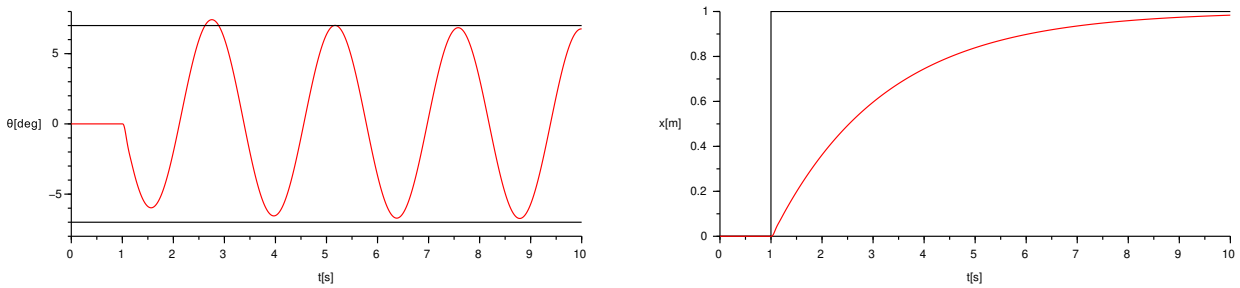

図 7　$P_t = 5, P_l = 0$ のシミュレーション結果

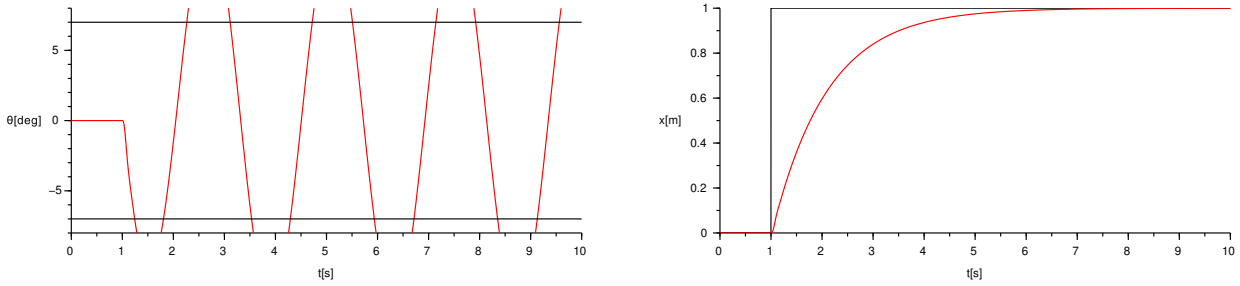

図 8　$P_t = 10, P_l = 0$ のシミュレーション結果

図 7, 8, 9 は, P_t 以外のパラメータを固定して, P_t を 5, 10, 15 と変化させたグラフである．グラフより, P_t が大きくなると, 吊り荷の揺れが大きくなり, トロリの整定が早まることがわかる．目標はトロリの整定がグラフ上で 6 秒以内であるので, ひとまず $P_t = 15$ として調整を進める．

次に, 図 10, 11, 12 は P_l のみを変化させたグラフである．P_l は大きくなると, 吊り荷の揺れが減り, トロリのオーバーシュートが大きくなることがわかる．トロリのオーバーシュートは 5% 以内に収めなくてはいけないので, ひとま

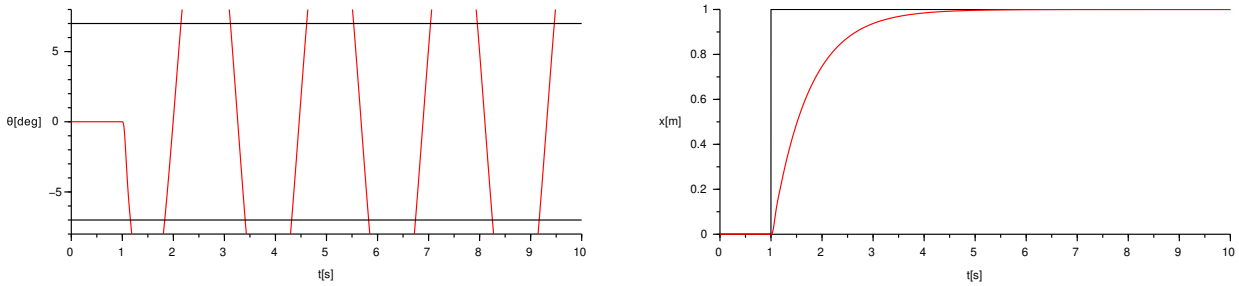

図 9　$P_t = 15, P_l = 0$ のシミュレーション結果

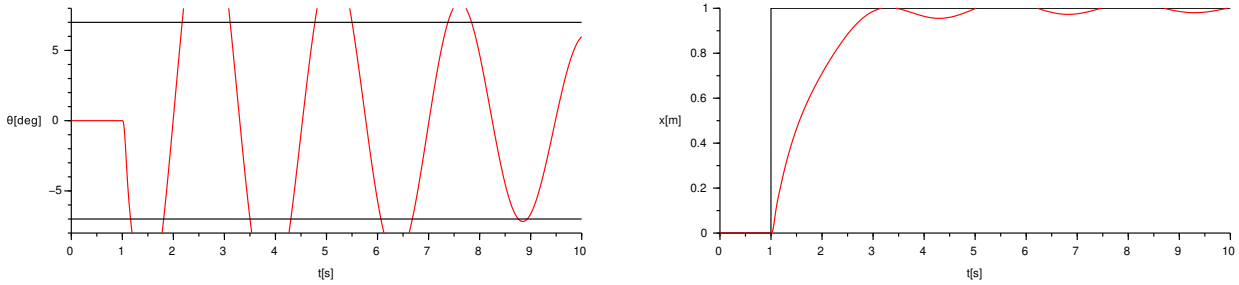

図 10　$P_t = 15, P_l = 5$ のシミュレーション結果

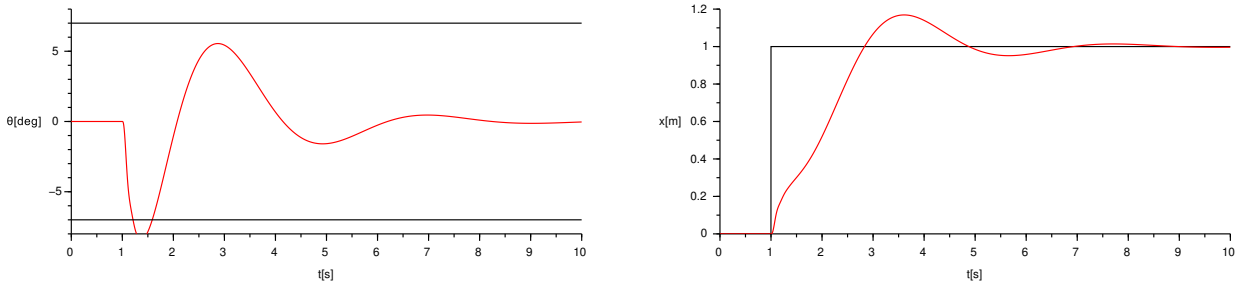

図 11　$P_t = 15, P_l = 50$ のシミュレーション結果

ずは $P_l = 50$ とする.

3.3.2　I ゲインの調整

次に I ゲインによる制御を検証する.

図 13, 14, 15 は I_t のみを変化させたグラフである. I_t が大きくなると, 吊り荷の挙動は変化せずに, トロリのオーバーシュートが大きくなっていることがわかる. 今回のシミュレーションではオーバーシュートは必要ないので, $I_t = 0$ として検証を進める.

次に図 16, 17, 18 は I_l のみを変化させたグラフである. I_l が増加すると吊り荷の振幅が抑えられ, トロリのオーバーシュートが抑えられるが, 大きくしすぎると図 18 のように整定が著しく遅くなってしまう. $I_l = 50$ として検証を進める.

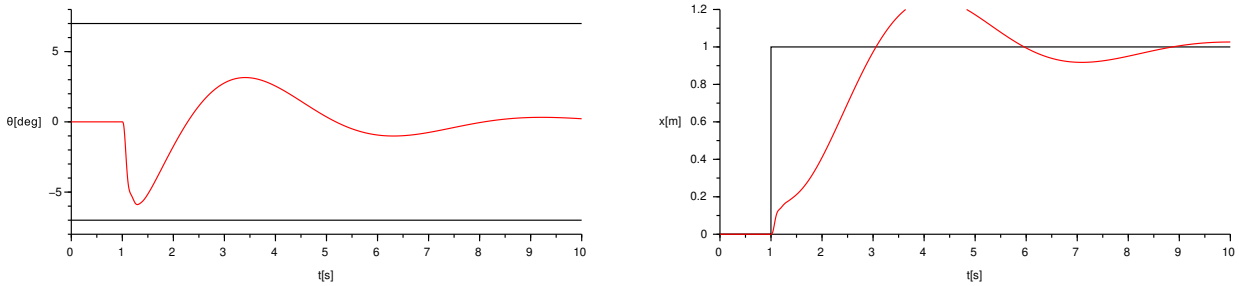

図 12　$P_t = 15, P_l = 100$ のシミュレーション結果

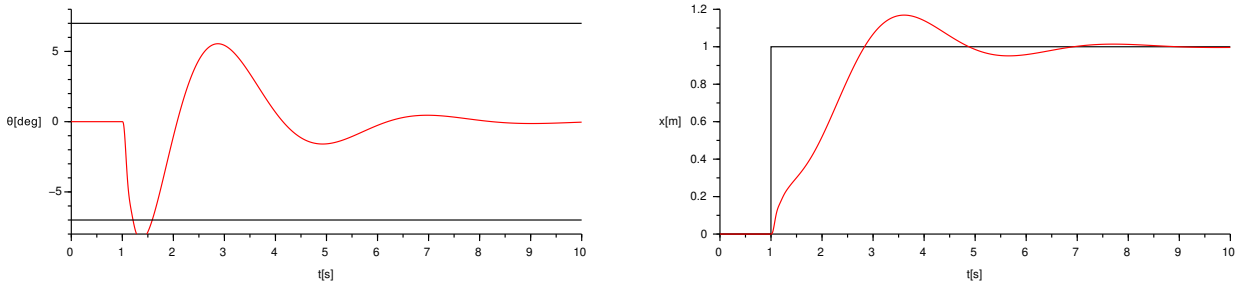

図 13　$I_t = 0, I_l = 0$ のシミュレーション結果

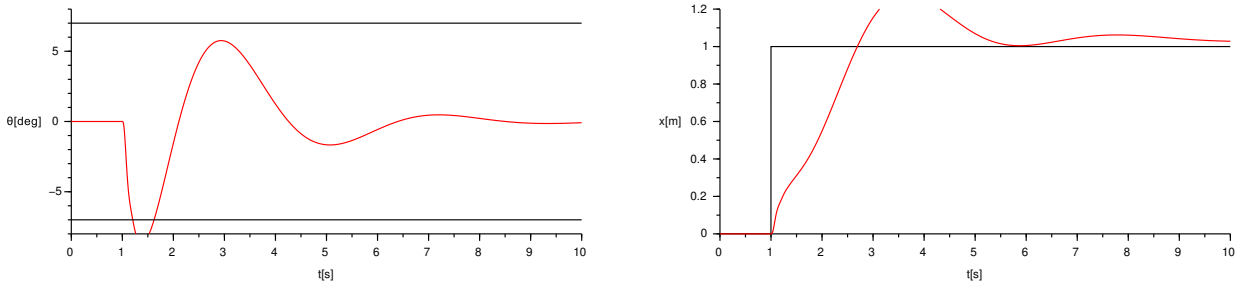

図 14　$I_t = 2, I_l = 0$ のシミュレーション結果

3.3.3　D ゲインの調整

最後に, D ゲインの働きをみる.

図 19, 20, 21 は D_t のみを変化させたグラフである. D_t が大きくなると吊り荷の振動, トロリの変位ともに小さくなることがわかる. 現状ではトロリの変位が小さくなってしまうと整定が間に合わないので, $D_t = 5$ とする.

最後に, 図 19, 図 22, 図 23 は D_l のみ変化させたグラフである. D_l が大きくなると, 吊り荷の振動, トロリの変位ともに大きくなり, 振動回数が増えてしまうことがわかる. ひとまず $D_l = 0$ とする.

3.3.4　ゲインの確定

ここまでで各ゲインがクレーンの応答に及ぼす影響がわかったので, これらを踏まえて値を確定する.

試行錯誤の結果, 表 1 に示すように値を定めた.

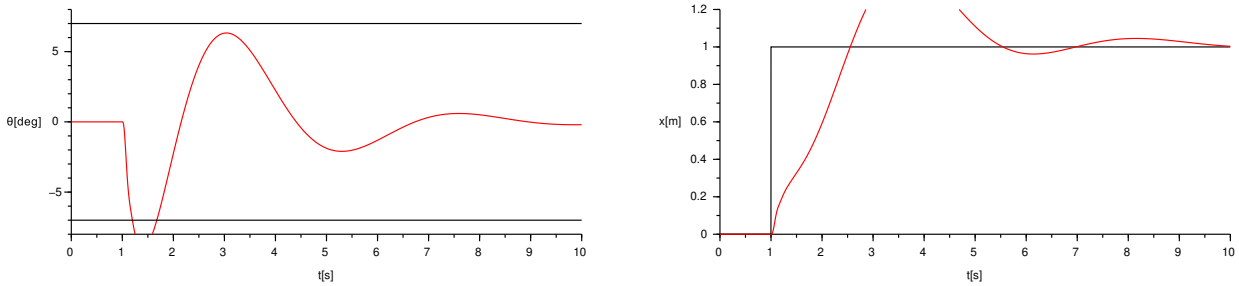

図 15　$I_t = 5, I_l = 0$ のシミュレーション結果

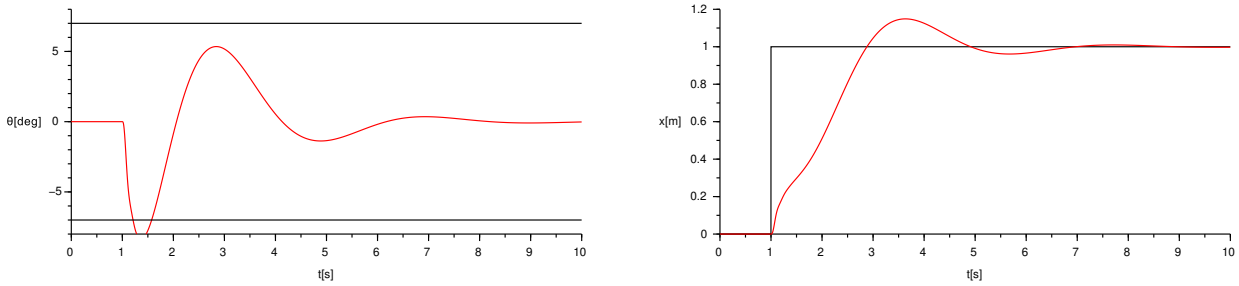

図 16　$I_t = 0, I_l = 5$ のシミュレーション結果

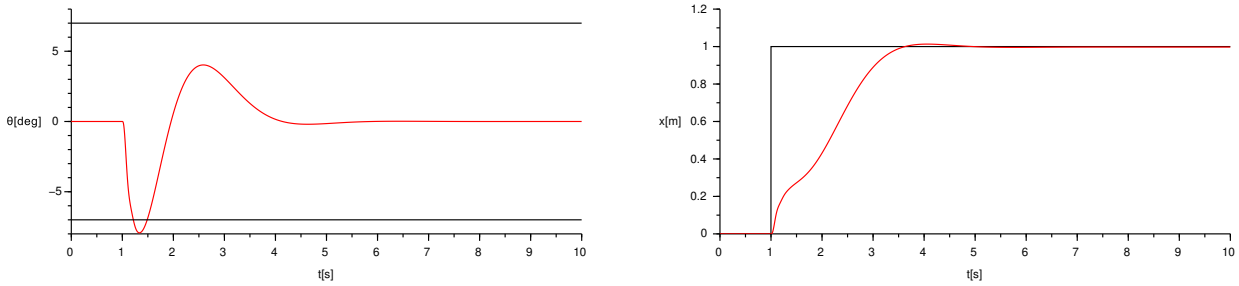

図 17　$I_t = 0, I_l = 50$ のシミュレーション結果

表 1　最終的なゲインの値

	P	I	D
トロリ	30	0	30
吊り荷	200	29	0

この値を用いてシミュレーションを行った結果を図 24 に示す.

図より, 二つの設計仕様を満たしていることがわかる.

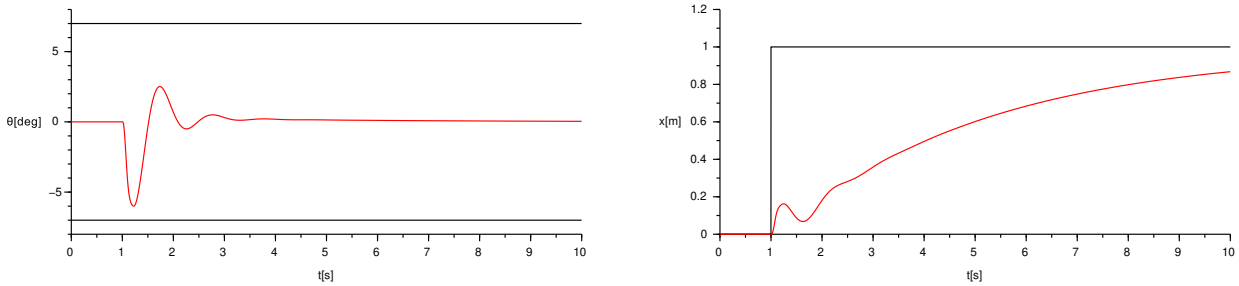

図 18　$I_t = 0, I_l = 500$ のシミュレーション結果

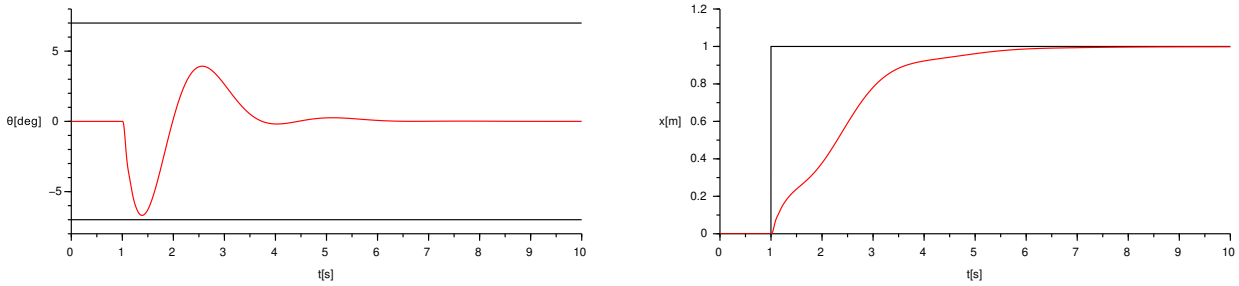

図 19　$D_t = 5, D_l = 0$ のシミュレーション結果

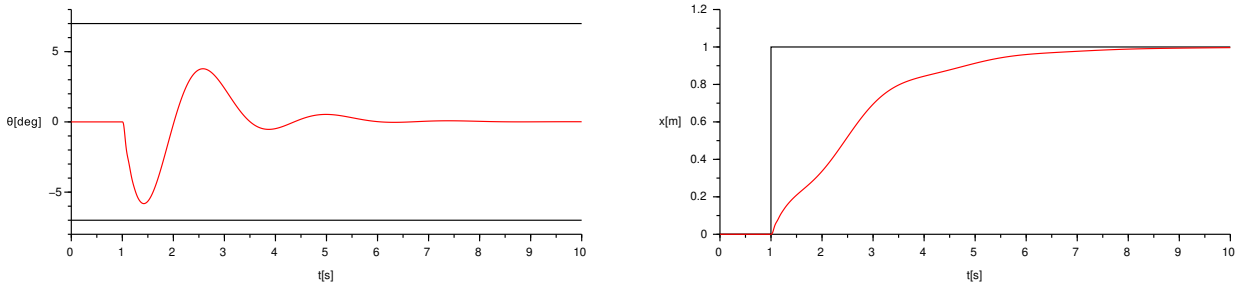

図 20　$D_t = 10, D_l = 0$ のシミュレーション結果

4　感想

案の定, 水曜日 (締め切り前々日) にレポートを書き始めることとなってしまった. 怠惰を招いた原因として, 普段使っている Mac では Scilab の一部機能がうまく動かなかったことがある. 私は Matlab のライセンスを所持しているのでそちらで同時にシミュレーションを確認しながら実験を進めた.

結果としてレポートを書き終えることができたが, 今後は今までのように時間に余裕を持って進めていきたい.

参考文献

[1] 数値シミュレーション, 電子制御工学実験 · 4 年後期テキスト, 2020

[2] 数値シミュレーションサポートページ, 佐藤 (拓)@電子制御,

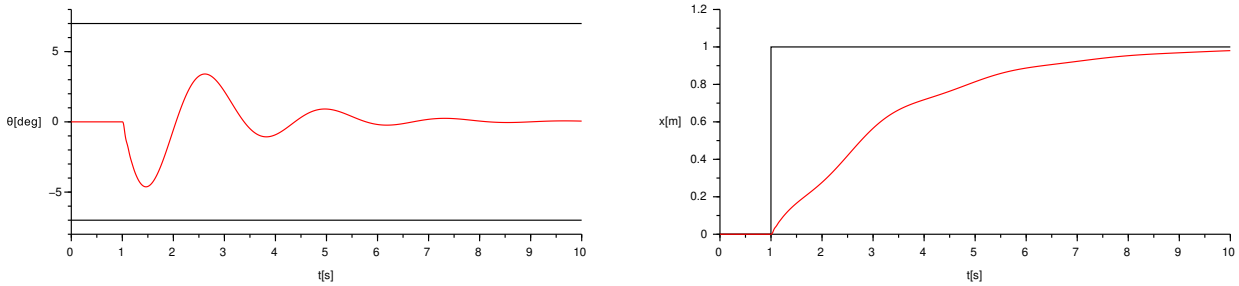

図 21　$D_t = 20, D_l = 0$ のシミュレーション結果

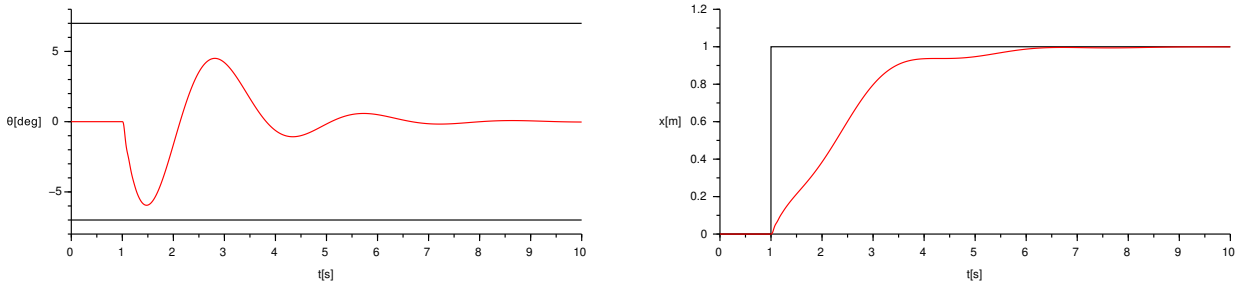

図 22　$D_t = 5, D_l = 10$ のシミュレーション結果

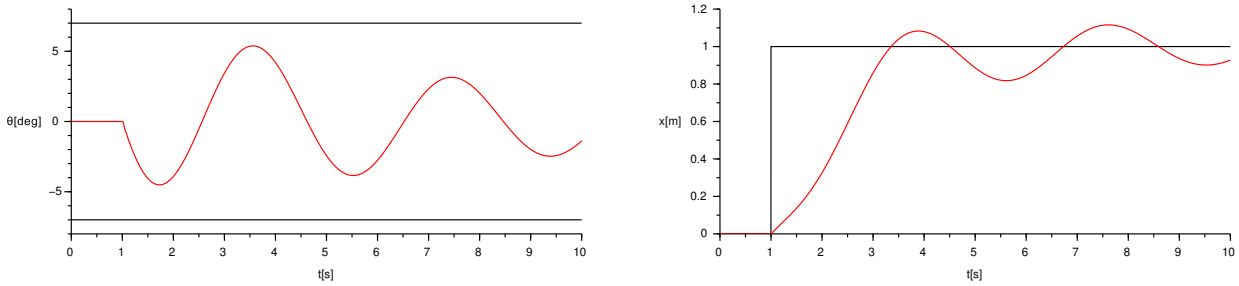

図 23　$D_t = 5, D_l = 50$ のシミュレーション結果

”https://www2.st.nagaoka-ct.ac.jp/ h-satoh/index.php?数値シミュレーション”, 2020

[3] 佐藤和也, 平元和彦, 平田研二: 初めての制御工学, 講談社, 2010

[4] 川谷亮治: 「Maxima」と「Scilab」で学ぶ古典制御, 工学社, 2014

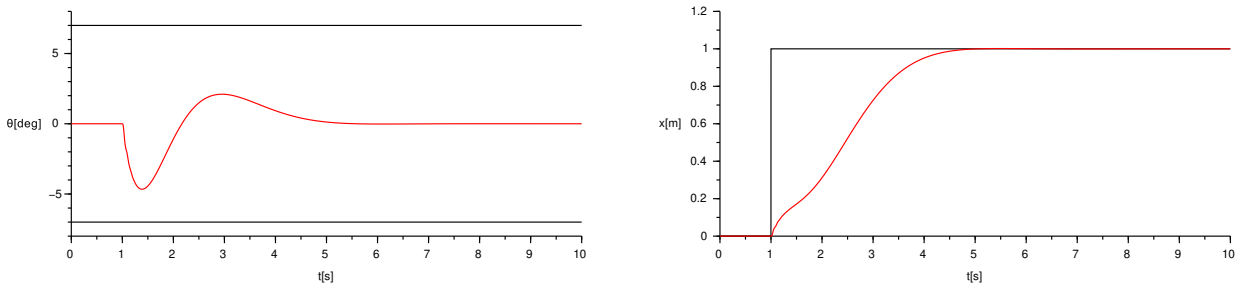

図 24　最終的なシミュレーション結果