

1 目的

本実験では、汎用数値計算ソフトである Scilab を使用し、過渡反応についてのシュミレーションを行う。Scilab でのシュミレーションから、過渡反応による数値変化を確認することを目的とする。

2 実験概要

2.1 実験環境

実験環境を以下の表 1 に示す。

表 1: 実験環境

デバイス	OS	ソフト
MacBookAir2019 13inch	MacOS BigSur 11.3.1	Scilab 6.1.0

2.2 実験方法

本実験では、1 次系伝達関数と 2 次伝達関数に入力する値を適当に 4 通り変更させ、プロットされる 4 つのグラフの変化から数値変化を読み取るものとする。

3 理論

3.1 Scilab とは

Scilab について、以下に ESI-Grope のホームページより引用する。 [1]

Scilab/Xcos はフランスの INRIA(国立情報学自動制御研究所) 及び ENPC(国立土木学校) で開発されたオープンソースの数値計算システムです。一般に公開するにあたり設立された Scilab Enterprises 社が開発を担当していましたが、2017 年に ESI-Group の一員となりました。

ベースとなる行列や数値計算機能だけではなく、多項式の計算や信号処理或いはそれらを視覚的に表示するプロット機能等、様々な用途に用いられています。そして、Windows/Linux/Mac OS の各プラットフォームに対応しています。

3.2 Scilab の基本



図 1: Scilab のアイコン

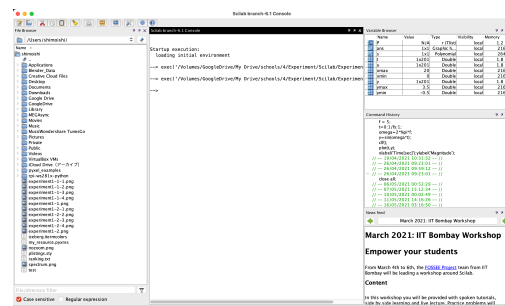


図 2: Scilab のデスクトップ

本実験では MacOS を使用したため、MacOS 上での操作手順を示す。

3.2.1 起動・終了方法

Scilab を起動する際は、Scilab を起動する際は、Launchpad 内の Scilab のアイコン (図??) をクリックする。または、Spotlight の機能を用いて起動することもできる。Scilab を終了する際は、左上の×ボタンをクリックする、もしくは、ショートカットキー (Command + Q) から終了することができる。

3.2.2 数値計算

数値計算を行う際は、コンソールと呼ばれる領域に数値、コマンドを入力する。コンソールを用いる以外にも、SciNotes というウィンドウを使い数値計算をすることもできる。Scilab のデスクトップ (図 2) の右上にあるアイコンから起動することができる。SciNotes はテキストエディタのように使う事ができ、まとまった形でプログラムを作成する事ができる。

3.2.3 主な関数・構文

Scilab を利用するにあたり、よく利用する関数や使用例を以下の表 2 に示す。

表 2: よく利用される関数

関数	意味	注意
%pi	π	%pi を付けなければならない
abs	絶対値	$abs(-1) = 1$
sqrt	平方根	$sqrt(2) = 1.4142136$
exp	指数関数	$exp(1) = 2.7182818$
round	四捨五入	$round(1.4) = 1$
clear	これまでの変数をクリア	clear; これまでのデータを削除

3.3 過渡反応

過渡反応 [2] とは、回路素子の値が変化や、スイッチの切り替わりにより、回路がある定常状態から別の定常状態に移るまでの電流・電圧の時間的变化のことである。

ここで、

$$G(s) = \frac{K}{Ts + 1}$$

(1)

で与えられたとき、このシステムを 1 次系、もしくは 1 次遅れ系という。この式のゲイン K は大きくなるほど信号が大きく増幅されることを意味し、時定数 T は応答の速さを示すものである。また、時刻 $t = T$ のとき、最終値の 63.2% に達する。つまり、式 1 にステップ入力を与えたときの出力の定常値とその 63.2% に達するまでの時間を調べることで、ゲイン K と時定数 T を求めることができる。

また、

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

(2)

で与えられたとき、このシステムを 2 次系、もしくは 2 次遅れ系という。この式の K はゲイン、 ζ は減衰係数、 ω_n は自然角周波数を表している。 $\zeta > 1$ は非振動的で、 $0 < \zeta < 1$ では振動的となる。

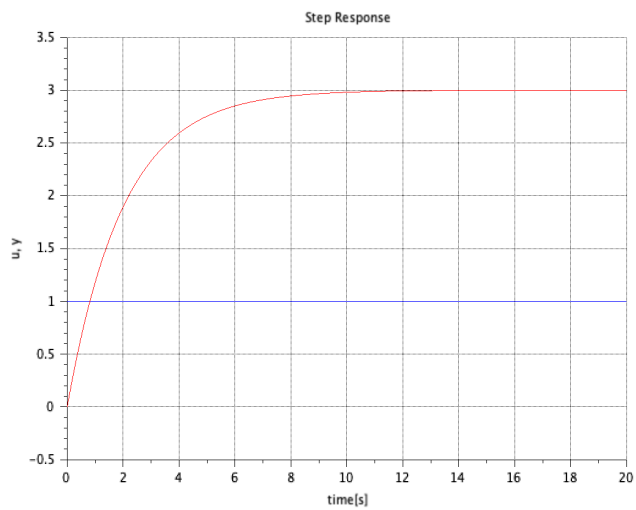
4 実験

4.1 1 次系伝達関数

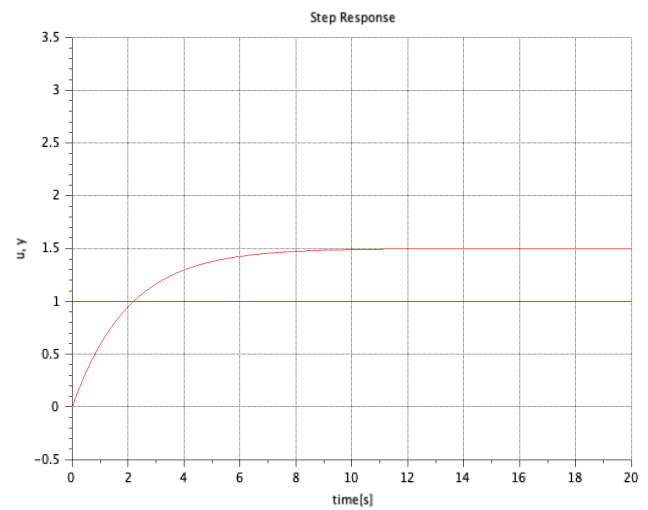
式 1 に対して適当なゲイン K , T を代入しグラフの変化を確認する。Scilab のソースコードを以下の Code1 に示す。 [3]

```
1 clear();
2 s = %s;
3 K = 0.5;
4 T = 3;
5 P = syslin('c', K/(T*s+1));
6 t = 0:0.1:20;
7 u = ones(t);
8 y = csim(u,t,P);
9 xmin = 0; xmax = 20; ymin = -0.5; ymax = 3.5;
10 scf(0);
11
12 plot2d(t,u,style=color(0,0,255),rect=[xmin ymin xmax ymax]);
13
14 plot2d(t,y,style=color(255,0,0),rect=[xmin ymin xmax ymax]);
15
16 xtitle('Step Response', 'time[s]', 'u, y')
17 xgrid();
18
19 //save file as PNG
20 xs2png(0, 'experiment1-1-4.png')
```

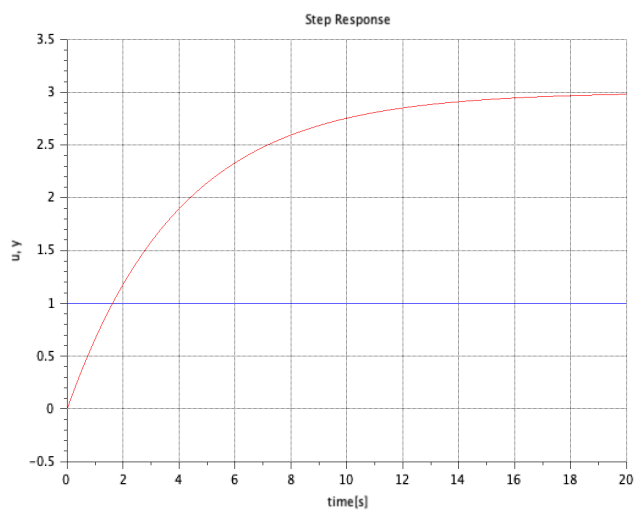
また、以下の図 3 にそれぞれ値を変化させたグラフを示す。



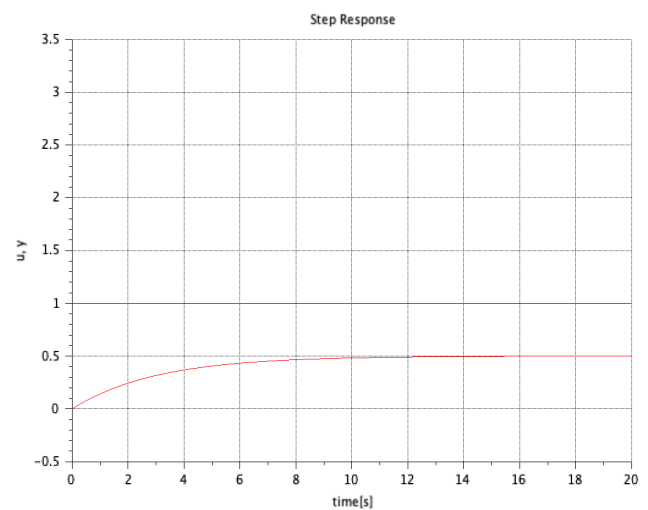
(a) $K = 3, T = 2$ のときのグラフ



(b) $K = 1.5, T = 2$ のときのグラフ



(c) $K = 3, T = 4$ のときのグラフ



(d) $K = 0.5, T = 3$ のときのグラフ

図 3: 1 次系伝達関数の値を変化させたグラフ

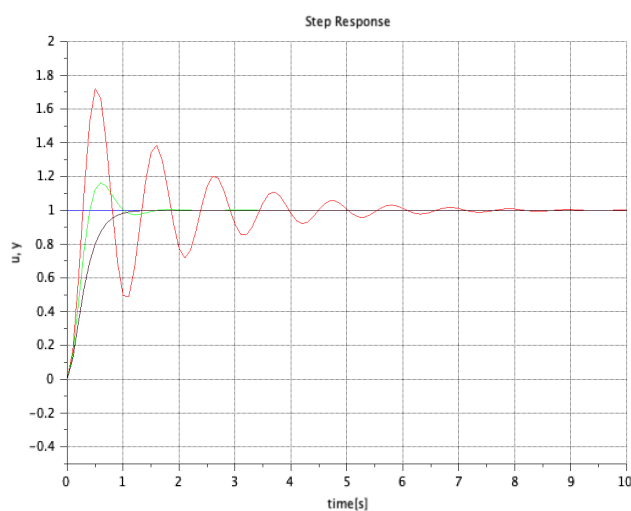
4.2 2次系伝達関数

式 2 に対して適当なゲイン K , ω_n , ζ を代入しグラフの変化を確認する. Scilab のソースコードを以下の Code2 に示す. [3]

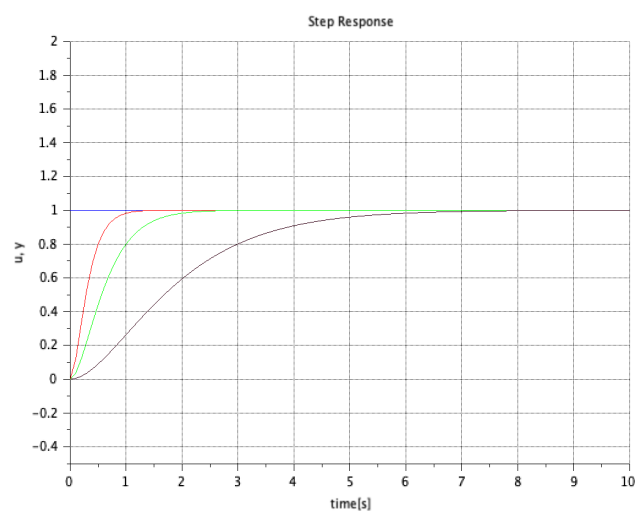
Code 2: 2次系伝達関数のソースコード

```
1 clear();
2 s = %s;
3 K = 1;
4 omega = 6;
5 zeta = 0.1;
6 P1 = syslin('c', K*omega^2/(s^2 + 2 * zeta * omega*s + omega^2));
7
8
9 zeta = 0.5;
10 P2 = syslin('c', K*omega^2/(s^2 + 2 * zeta * omega*s + omega^2));
11
12
13 zeta = 1.0;
14 P3 = syslin('c', K*omega^2/(s^2 + 2 * zeta * omega*s + omega^2));
15
16 t = 0:0.1:20;
17 u = ones(t);
18
19 y1 = csim(u,t,P1);
20 y2 = csim(u,t,P2);
21 y3 = csim(u,t,P3);
22 xmin = 0; xmax = 10; ymin = -0.5; ymax = 2;
23 scf(0);
24
25 plot2d(t,u,style=color(0,0,255),rect=[xmin ymin xmax ymax]);
26
27 plot2d(t,y1,style=color(255,0,0),rect=[xmin ymin xmax ymax]);
28 plot2d(t,y2,style=color(0,255,0),rect=[xmin ymin xmax ymax]);
29 plot2d(t,y3,style=color(50,0,20),rect=[xmin ymin xmax ymax]);
30
31 xtitle('Step Response', 'time[s]', 'u, y')
32 xgrid();
33 //save file as PNG
34 xs2png(0, 'experiment1-2.png')
```

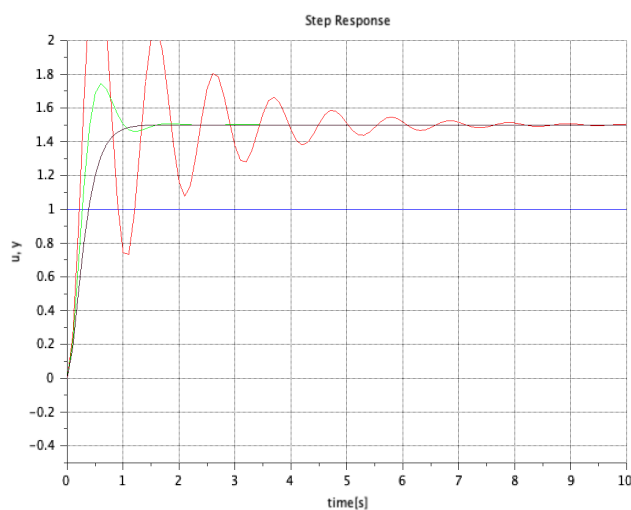
また, 以下の図 4 にそれぞれ値を変化させたグラフを示す.



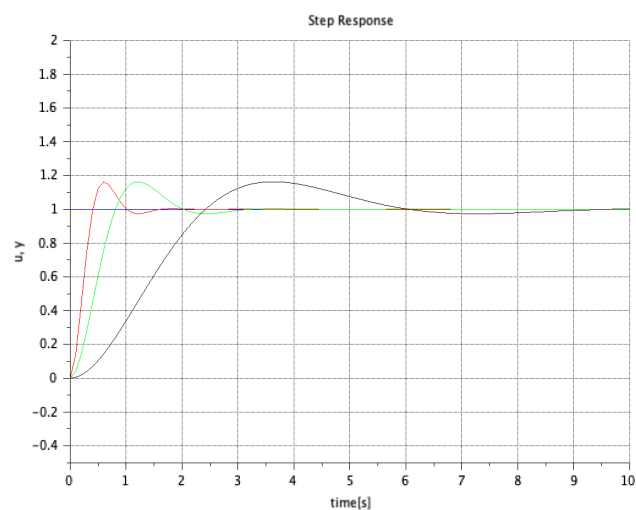
(a) $K = 1, \omega_n = 6, \zeta = 0.1, 0.5, 1.0$ のときのグラフ



(b) $K = 1, \omega_n = 6, 3, 1, \zeta = 1$ のときのグラフ



(c) $K = 1.5, \omega_n = 6, \zeta = 0.1, 0.5, 1.0$ のときのグラフ



(d) $K = 1, \omega_n = 6, 3, 1, \zeta = 0.5$ のときのグラフ

図 4: 2 次系伝達関数の値を変化させたグラフ

4.3 研究課題

実験 1 において、SIMULINK シミュレーションのサンプル時間を 0.001[sec] から 1[sec] まで変化させたときの計算結果を比較し、考察しなさい。

以下に研究課題の実行結果を示す。ここで、Scilab ではソースコード 1 を使用し、 $K = 3, T = 1$ を代入している。サンプル時間は $t = 0 : 1 : 20$ (図 5) と $t = 0 : 0.001 : 20$ (図 6) で比較しており、比較しやすいように、x 軸の表示範囲を 0 ~ 10 に変更している。

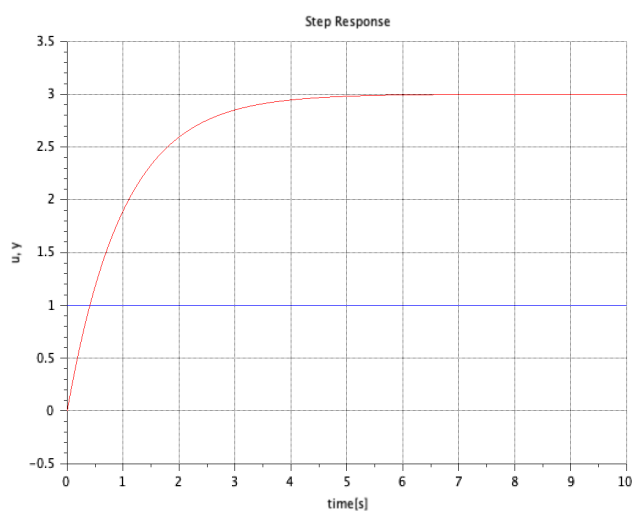


図 5: $t = 0 : 0.001 : 20$ のグラフ

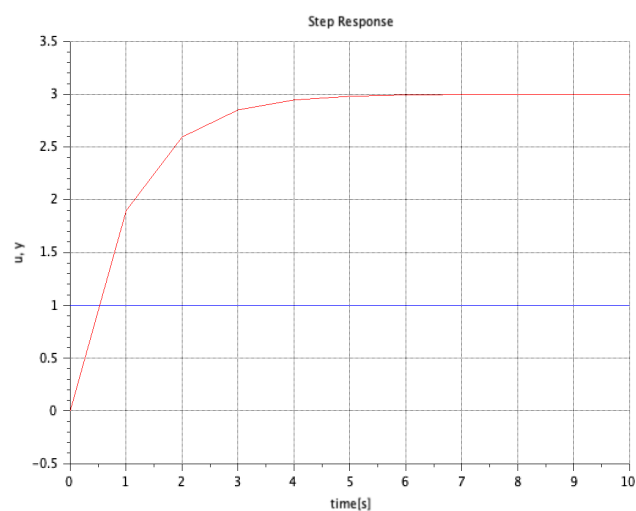


図 6: $t = 0 : 1 : 20$ のグラフ

5 考察

研究課題において、サンプル時間を変更したことでなめらかなグラフとカクカクしたグラフを出力することができた。これはサンプル時間が小さくなるほど図 5 のような精度の高いグラフを出力することができることを表しており、図 6 のようなグラフは精度が低いと言える。したがって、正しい数値をグラフから読み取るためには、十分なサンプルを取得しグラフを出力することが大切である。

参考文献

- [1] ESI-Grope "Scilab/Xcos とは (1)" (最終閲覧日 2021 年 5 月 16 日)
<https://solution.esi.co.jp/scilab/blog/introduction1>
- [2] Joseph Edminister, Mahmood Nahvi "Electric Circuits, 6th edition" (出版年 2013/11/8)
- [3] 永田 正伸 "制御系 CAD:Scilab による過渡応答シュミレーション" 制御情報システム工学科 制御工学実験 指導書 2020.6