Contents

List of Scilab Code

Chapter 1

Identification of transfer function of a Single Board Heater System through step response experiments

The Aim of this experiment is to perform step test on a single board heater system and to identify system transfer function using step response data. The target group is anyone who has basic knowledge of Control Engineering.

1.1 About this Experiment

- We have used Scilab and Scicos as an interface for sending and receiving data. Scicos diagram is shown in Figure 1.1.
- Heater current and fan speed are the two inputs for this system. They are given in PWM units.
- These inputs can be varied by the setting the properties of inpput block's properties in scicos.For details of setting of properties please refer "Instruction mannual for temperture controller"
- The plots of their amplitude versus no. of collected samples are also available on the scope windows.
- The output temperature profile, as read by the sensor, is also plotted.

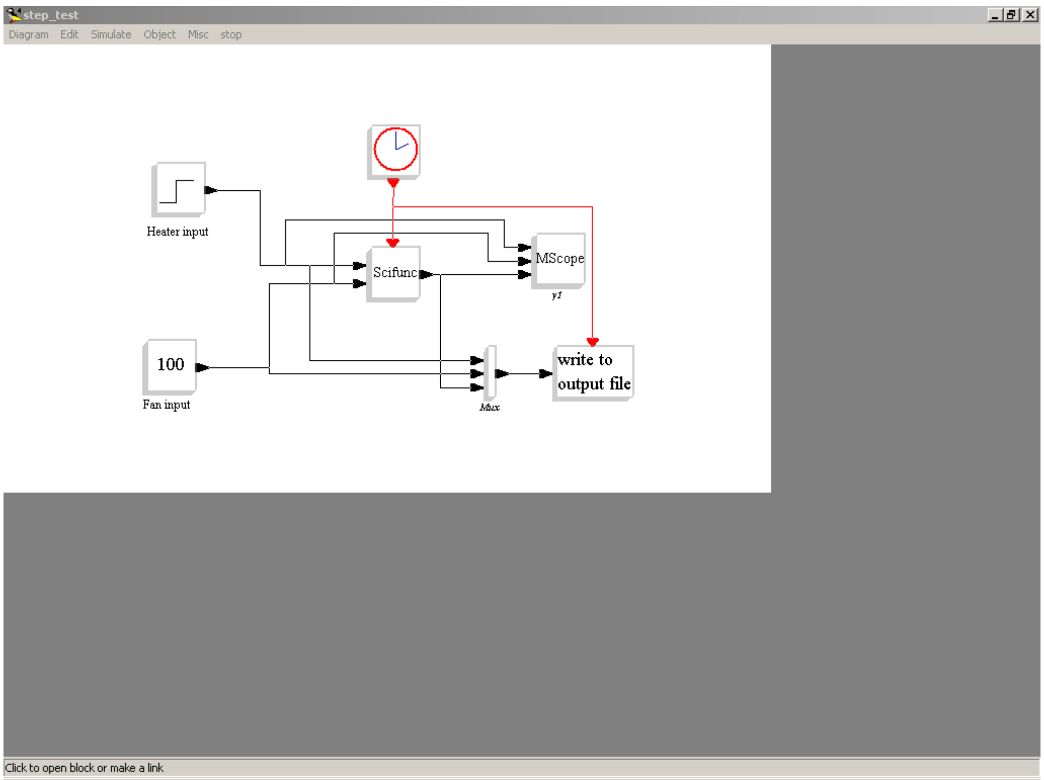

Figure 1.1: Scicos for this experiment

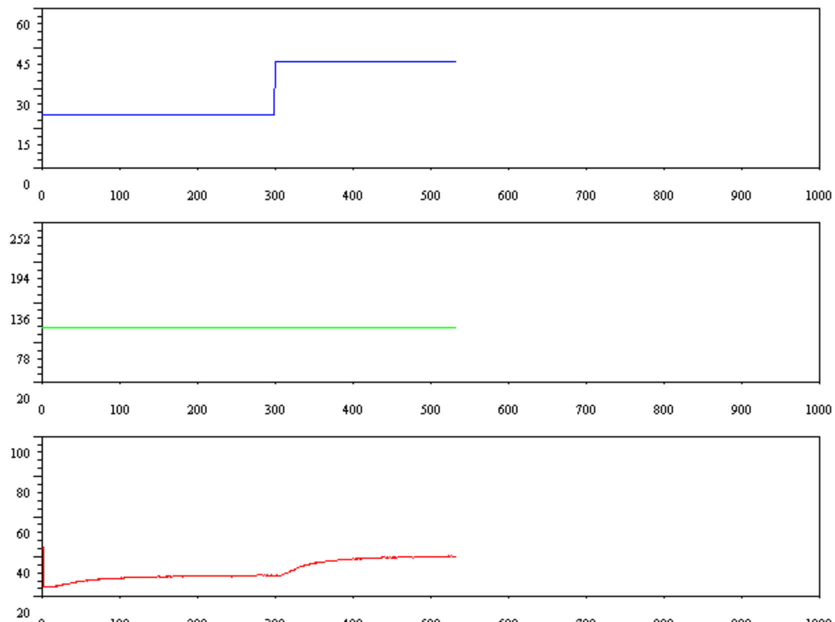

Figure 1.2: Graph shows Heater current , Fan speed and Output Temperature

- The data acquired in the process is stored on the local drive and is available to the user for further calculations.

1.2 Theory

Identification of the transfer function of a system is quite important since it helps us to represent the physical system, mathematically. Once the transfer function is obtained one can acquire the response of the system for various inputs without actually applying them to the system.

Consider the standard first order transfer function given below

$$G(s) = \frac{C(s)}{R(s)} \tag{1.1}$$

$$G(s) = \frac{1}{\tau s + 1} \tag{1.2}$$

Rewriting the Equation, we get

$$C(s) = \frac{R(s)}{\tau s + 1} \tag{1.3}$$

A step is given as an input to the first order system. The Laplace Transform of a step function is $\frac{1}{s}$. Hence, substituting $R(s) = \frac{1}{s}$ in equation 1.3, we obtain

$$C(s) = \frac{1}{\tau s + 1}\frac{1}{s} \tag{1.4}$$

Solving $C(s)$ using partial fraction expansion, we get

$$C(s) = \frac{1}{s} - \frac{1}{s + \frac{1}{\tau}} \tag{1.5}$$

Taking the Inverse Laplace transform of equation 1.5, we get

$$c(t) = 1 - e^{\frac{-t}{\tau}} \tag{1.6}$$

from the above equation it is clear that for t=0 the value of c(t) is zero. For t= ∞, c(t) approaches unity. Also as the value of 't'becomes equal to τ, the value of c(t) becomes 0.632. The τ is called as the time constant and represents the speed of response of the system. But it should be noted that, the smaller the time constant, the faster the system response.
By getting the value of τ, one can identify the transfer function of the system.

1.3 Step by step procedure to perform Step Test

Apply two step changes of say 20 PWM units as an input keeping fan speed constant. The data obtained is divided in to two parts, step1.dat and step2.dat therefore considering it as two different steps. The first column refers to Time, second-Heater, third-Fan and fourth-Temperature. The first column in this table denotes time. The second column in this table denotes heater current. It is zero to start with and increases with a step size of 20 units. The third column denotes the fan speed. It has been held constant at 100 units. Only one point is retained for zero current value and all other points are deleted. The last column denotes the plate temperature. Plot the graph of the data thus acquired using Scilab code

Table 1.1: Data obtained for the first step

```
data7 = [
 0.110E+01  0.000E+02  0.100E+03  0.247E+02
 0.210E+01  0.200E+02  0.100E+03  0.247E+02
 0.310E+01  0.200E+02  0.100E+03  0.247E+02
.
.
 0.296E+03  0.200E+02  0.100E+03  0.307E+02
 0.297E+03  0.200E+02  0.100E+03  0.307E+02
 0.298E+03  0.200E+02  0.100E+03  0.307E+02
 0.299E+03  0.200E+02  0.100E+03  0.307E+02
];
```

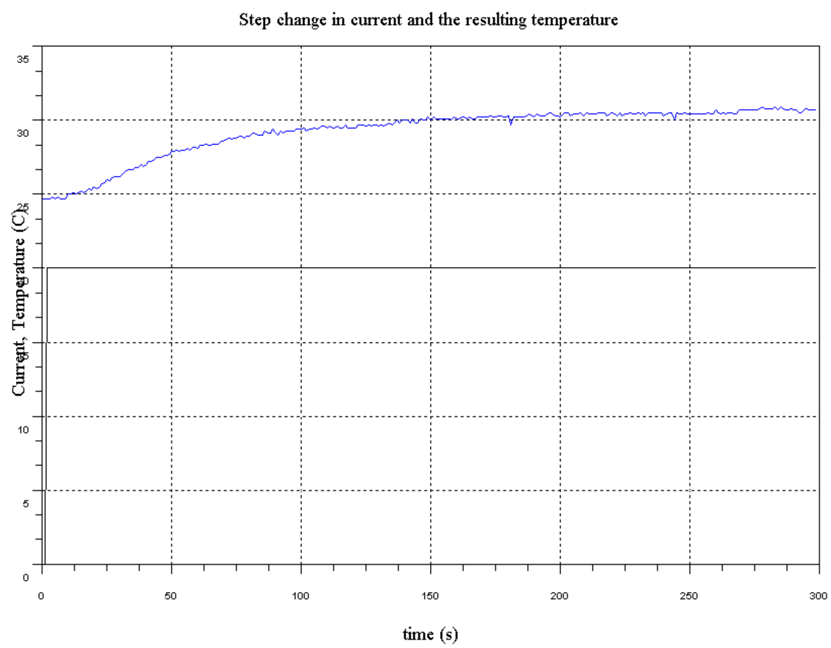

Figure 1.3: Output of the scilab code `first.sce`

`first.sce.` [1] Consider the system to be first order. We try to fit a first order transfer function of the form

$$G(s) = \frac{K}{\tau s + 1} \tag{1.7}$$

to the single board heater system. Because the transfer function approach uses deviational variables,$G(s)$ denotes the Laplace transform, of the gain of the system between the change in heater current and the change in the system temperature. Let the change in the heater current be denoted by Δu. We denote both the time domain and the Laplace transform variable by the same lower case variable. Let the change in temperature be denoted by y.Suppose that the current changes by a ramp of size u.Then, we obtain the following relation between the current and the temperature.

$$y(s) = G(s)u(s) \tag{1.8}$$

$$y(s) = \frac{K}{\tau s + 1}\frac{\Delta u}{s} \tag{1.9}$$

Note that Δ u is the height of the step and hence is a constant. On inversion, we obtain

$$y(s) = K[1 - e^{\frac{-t}{\tau}}]\Delta u \tag{1.10}$$

Run the scilab code `approx_step.sce` and obtain the following plot. This code uses the routines `label.sci` and `step.sci` Calculate τ and final value from the plotted graph and run the same scilab code with these values. τ is the time taken by the system to reach 63.2% of final value.

We obtain final value = 5.8, therefore τ = 62. The least square error turns out to be 4.438. As the current changes by 20 units, the steady state gain K turns out to be 5.8/20. Let the transfer function we obtain in this experiment be denoted by G_{11}. We obtain

$$G_{11}(s) = \frac{5.8}{20}\frac{1}{62s + 1}; error = 4.438 \tag{1.11}$$

The gain is $5.8/20 = 0.29$ and the time constant τ is 62s. Repeat the same steps for the second step. The plot thus obtained is

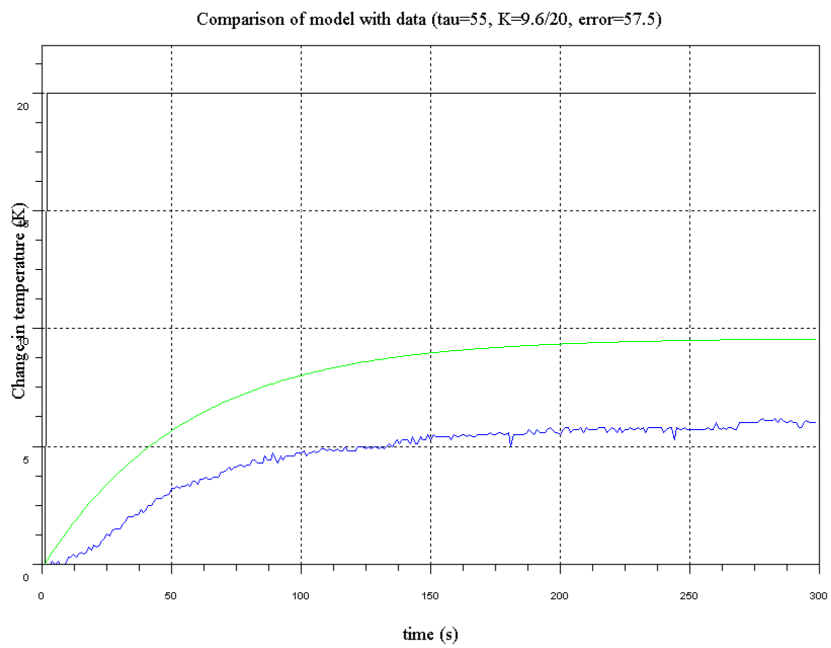

Figure 1.4: Output of the scilab code `approx_step.sce`

Table 1.2: Data obtained for the second step

```
data7 = [
 0.300E+03  0.400E+02  0.100E+03  0.307E+02
 0.301E+03  0.400E+02  0.100E+03  0.308E+02
 0.302E+03  0.400E+02  0.100E+03  0.307E+02
 0.303E+03  0.400E+02  0.100E+03  0.305E+02
.
.
.
.
 0.530E+03  0.400E+02  0.100E+03  0.402E+02
 0.531E+03  0.400E+02  0.100E+03  0.401E+02
 0.532E+03  0.400E+02  0.100E+03  0.402E+02
 0.533E+03  0.400E+02  0.100E+03  0.403E+02
];
```

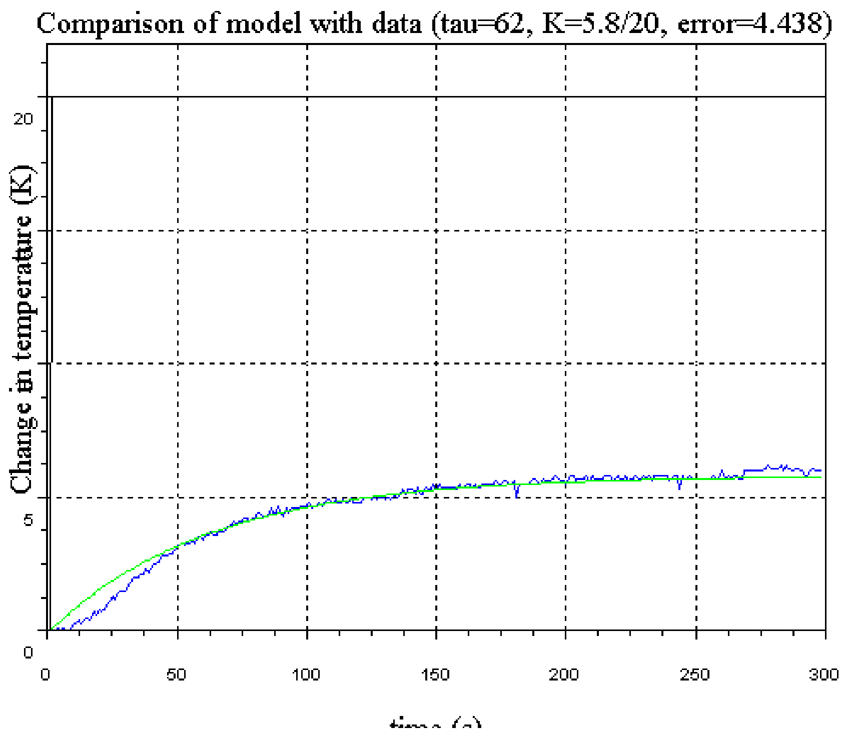

Figure 1.5: Plot obtained for first step with appropriate value of τ and final value

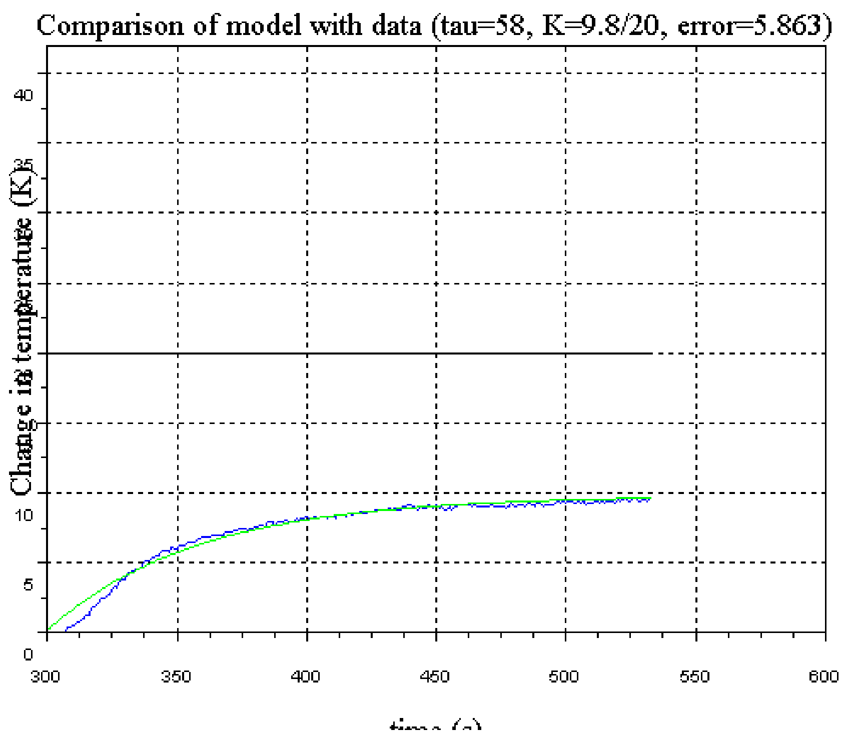

Figure 1.6: Plot obtained for second step

The plot is once again reasonably good. We obtain τ = 58, K = 9.8/20 and the least square error to be 5.862 . The time constant of 58 is less than the value of 62, obtained in the first step. This says that the plant is faster at higher temperature. The steady state gain K now comes out to be 5.8/20, which is larger than the earlier value of 5.8/20. Thus the transfer function of the plant varies with the operating point. Let the transfer function we obtain in this experiment be denoted as G_{12}.We obtain

$$G_{12}(s) = \frac{9.8}{20}\frac{1}{58s+1}; error = 5.862 \tag{1.12}$$

The gain is $9.8/20 = 0.49$ and the time constant τ is 58s. At the time of second step, the system is faster, with a larger gain, as compared to the first step.

1.4 Variation

- The gain is 9.8/20 = 0.49 and the time constant τ is 58s. As mentioned earlier, at the time of second step, the system is faster, with a larger gain, as compared to the first step.
- Negative steps can also be introduced to make the experiment more informative.
- One can even need not keep a particular input constant. By varying both the inputs, one can imagine it to be like a step varying disturbance signal.
- The system can also be treated as a second order system. This consideration is however necessary since it increases the accuracy of the acquired transfer function.[**?**]

1.5 Scilab Codes

Scilab Code 1.1 `label.sci`

[1]Note:- The associated Scilab codes are given at the end of this Document.

```
// Updated (9-12-06), written by Inderpreet Arora
// Input arguments: title, xlabel, ylabel and their
    font sizes

function label(tname,tfont,labelx,labely,xyfont)
a = get("current_axes")
xtitle(tname,labelx,labely)
xgrid
t = a.title;
t.font_size = tfont; //Title font size
t.font_style = 2; //Title font style
t.text = tname;

u = a.x_label;
u.font_size = xyfont; //Label font size
u.font_style = 2; //Label font style

v = a.y_label;
v.font_size = xyfont; //Label font size
v.font_style = 2; //Label font style

// a.label_font_size = 3;

endfunction;
```

Scilab Code 1.2 `approx_step.sci`

```
function lsterr = approx_step(T,u,y,final,tau,limits,
    no)
t0 = T(1); delta_u = u(2) - u(1);
u = u - u(1); y = y - y(1);
y_prediction = final*(1-exp(-(T-t0)/tau));
format('v',6);
lsterr = norm(y-y_prediction,2);
ord = [u y y_prediction]; x = [T T T];
xbasc(); plot2d(x,ord,rect=limits), xgrid();
title = 'Comparison of model with data (tau='
title = title+string(tau)+', K='+string(final)
```

```
title = title+'/'+string(delta_u)+', error='+string(
    lsterr)+')'
label(title,4,'time (s)','Change in temperature (K)'
    ,4);
```

Scilab Code 1.3 `first.sce`

```
clear data7; exec('data0.2.dat'); getf('label.sci');
T = data7(:,1); fan = data7(:,3) //T is time, fan is
    fan speed
u = data7(:,2); y = data7(:,4); // u is current, y is
    temperature
ord = [u y]; x = [T T]; // u and y are plotted vs.
    time and time
xbasc(); plot2d(x,ord); xgrid();
title = 'Ramp change in current and the resulting
    temperature'
label(title,4,'time (s)','Current, Temperature (C)',4)
    ;
```

Scilab Code 1.4 `approx_step.sce`

```
clear data7; exec('winter40.txt'); getf('label.sci');
getf('approx_step.sci');
T = data7(:,1); //T is time
u = data7(:,2); y = data7(:,4); // u is current, y is
    temperature
tau = 44; final = 9; limits = [300,0,600,42]; no=4000;
     // first step
//tau = 40; final = .3; limits = [400,0,900,26]; no
    =5000;// second step
lsterr = approx_step(T,u,y,final,tau,limits,no)
```