Contents

List of Scilab Code

Chapter 1

Frequency Response Analysis of a Single Board Heater System by the application of Sine Wave

The aim of this experiment is to perform Frequency Response Analysis of a Single Board Heater System by the application of Sine Wave.The target group is anyone who has basic knowledge of Control Engineering.

1.1 About this Experiment

We have used Scilab with Scicos as an interface for sending and receiving data. This interface is shown in Fig.1.1. Heater current and Fan speed are the two inputs to the system. The Heater current is varied sinusoidally. A provision is made to set the parameters related to it, like Frequency, Amplitude and Offset. The temperature profile thus obtained is the output. In this experiment we are applying a sine change in the heater current by keeping the Fan speed constant. After application of sine change, wait for sufficient amount of time to allow the temperature to reach a steady-state.

1.2 Theory

Frequency Response of a system means its steady-state response to a sinusoidal input. For obtaining a Frequency Response of a system, we vary the frequency of

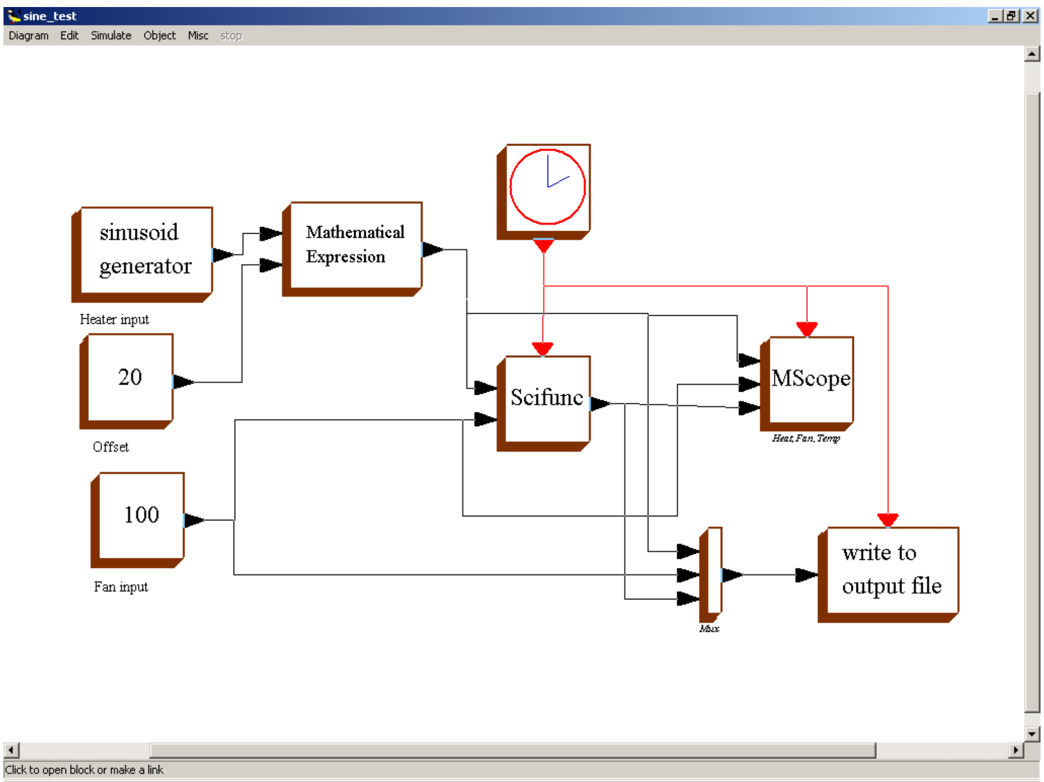

Figure 1.1: Scicos for this experiment

the input signal over a spectrum of interest. The analysis is actually quite useful and also simple because it can be carried out with the available signal generators and measuring devices. Consider a sinusoidal input

$$U(t) = A sin \omega t \tag{1.1}$$

The Laplace Transform of the above equation yields

$$U(s) = \frac{A\omega}{s^2 + \omega^2} \tag{1.2}$$

Consider the standard first order transfer function given below

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{s+1} \tag{1.3}$$

Putting the value of U(s) from equation, we get

$$Y(s) = \frac{KA\omega}{(\tau s + 1)(s^2 + \omega^2)} \tag{1.4}$$

$$= \frac{KA}{\omega^2\tau^2 + 1}\left[\frac{\omega\tau^2}{\tau s + 1} - \frac{\tau s \omega}{s^2 + \omega^2} + \frac{\omega}{s^2 + \omega^2}\right] \tag{1.5}$$

Taking Laplace Inverse, we get

$$y(t) = \left[\frac{KA}{\omega^2\tau^2 + 1}\right]\left[\omega\tau e^{\frac{-t}{\tau}} - \omega\tau cos(\omega t) + sin(\omega t)\right] \tag{1.6}$$

The above equation has an exponential term $e^{\frac{-t}{\tau}}$. Hence, for large value of time, its value will approach to zero and the equation will yield a pure sine wave. One can also use trigonometric identities to make the equation look more simple.

$$y(t) = \left[\frac{KA}{\sqrt{\omega^2\tau^2 + 1}}\right][sin(\omega t) + \phi] \tag{1.7}$$

where,

$$\phi = -tan^{-1}(\omega\tau) \tag{1.8}$$

By observing the above equation one can easily make out that for a sinusoidal input the output is also sinusoidal but has some phase difference. Also, the amplitude of the output signal, $\hat{A}$, has become a function of the input signal frequency, ω.

$$\hat{A} = \frac{KA}{\sqrt{\omega^2\tau^2 + 1}} \tag{1.9}$$

The amplitude ratio (AR) can be calculated by dividing both sides by the input signal amplitude A.

$$AR = \frac{\hat{A}}{A} = \frac{K}{\sqrt{\omega^2\tau^2 + 1}} \tag{1.10}$$

Dividing the above equation by the process gain K yields the normalized amplitude ratio (AR_n)

$$AR_n = \frac{AR}{K} = \frac{1}{\sqrt{\omega^2\tau^2 + 1}} \tag{1.11}$$

Because the process steady state gain is constant, the normalized amplitude ratio often is used for frequency response analysis.[2]

$$\tag{1.12}$$

1.3 Step by step procedure to perform Sine Test

Initiate a sine input to the system by setting Sinusoid generator block properties with some value of the frequency (here $0.007Hz$) and amplitude(hear 10). Note that at high frequencies the plant output is not sinusoidal,which is not of any use. Hence,avoid choosing frequencies above $0.04Hz$.

The first column represents time. The second column represents heater current. Here, it is sinusoidally varied. The third column represents fan speed. Note that its value is 100 throughout the experiment. The fourth column represents the output temperature. It should be taken in to consideration that all the values mentioned in the data file are in PWM (Pulse Width Modulation) units, except for the temperature which is in °C.

Now let us see the step for calculating Amplitude Ratio and Phase Difference. Input the arguments `f` and `data_width` on line 4 and 6 respectively for the

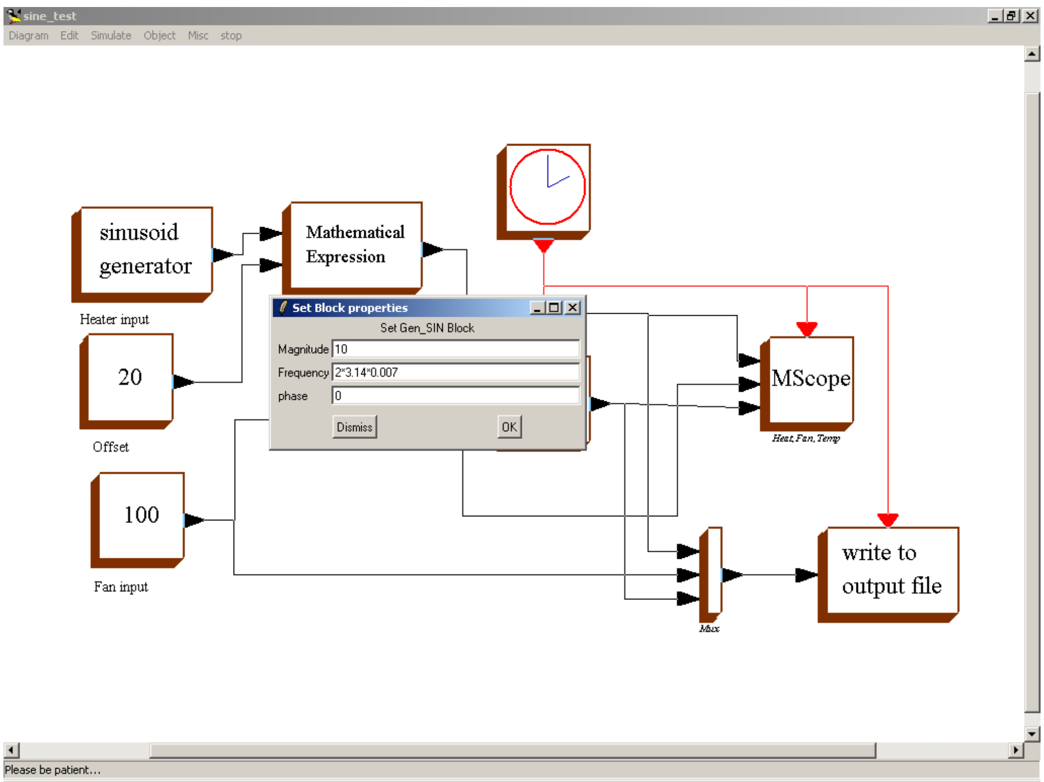

Figure 1.2: Setting Sine input parameters

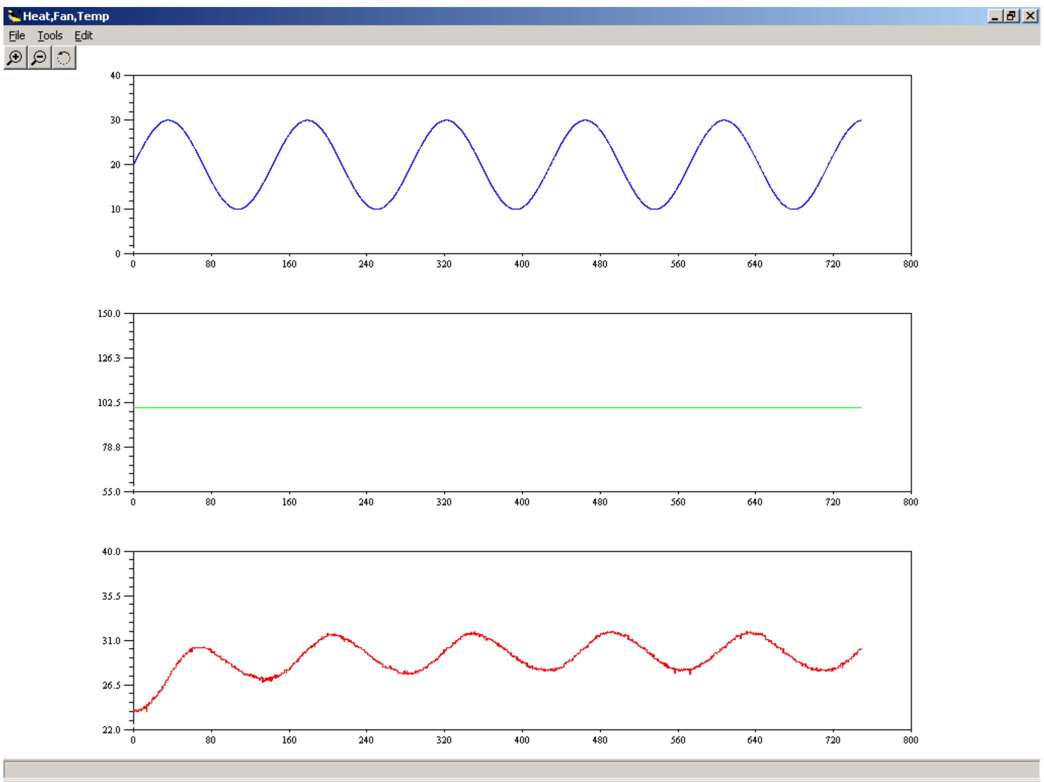

Figure 1.3: Plot for sine input 0.007Hz

Table 1.1: Data obtained after application of sine input of $0.04Hz$

```
data7=[
 0.100E+00  0.200E+02  0.100E+03  0.239E+02
 0.200E+00  0.201E+02  0.100E+03  0.238E+02
 0.300E+00  0.201E+02  0.100E+03  0.238E+02
.
.
.
 0.749E+03  0.300E+02  0.100E+03  0.301E+02
 0.749E+03  0.300E+02  0.100E+03  0.301E+02
 0.749E+03  0.300E+02  0.100E+03  0.302E+02
 0.749E+03  0.300E+02  0.100E+03  0.302E+02
]
```

scilab code `sine.sce` for the calculation of the above said parameters. Here `f` means input frequency and `data_width` means the range of data to be used for calculation. If data_width is 5, then last $\frac{1}{5}^{th}$ part of the whole data is used. It could be seen from figure 1.4 that the Amplitude Ratio turns out to be -8.6359dB and phase difference to be $-70.56°$. The plot thus obtained is shown in figure 1.5

Repeat this calculation over a range of frequencies and note down the values of Amplitude Ratio in dB and Phase Difference. Input these values for the appropriate frequencies in to the Scilab code `BodePlot` and execute it to get a Bode Plot of the plant which is illustrated in figure 1.6.

Bode Plot can be obtained directly from the plants Secon order Transfer function [1] with the help of scilab code `TFbode.sce`, as shown in figure 1.7. A visual comparison of the two bode plots can be done to validate the bode diagram obtained from the plant.

For comparing above two plots we are plotting it on same graph as shown in figure 1.8

1.4 Scilab Codes

Scilab Code 1.1 `sine.sce`

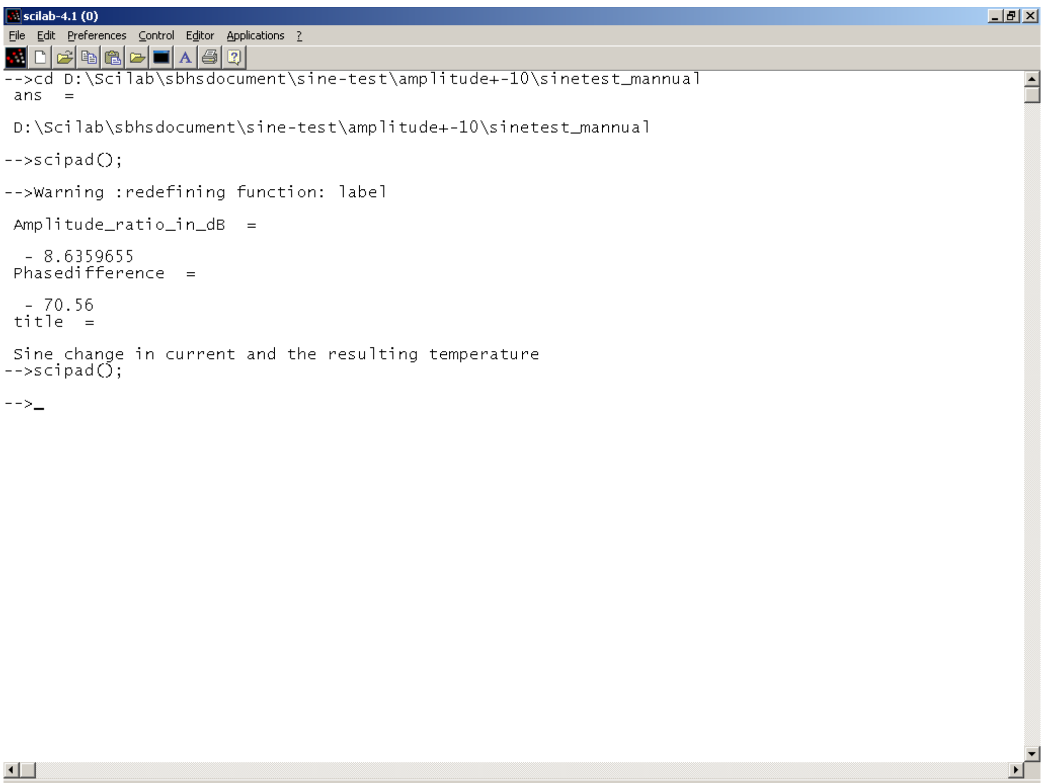

Figure 1.4: Scilab Output

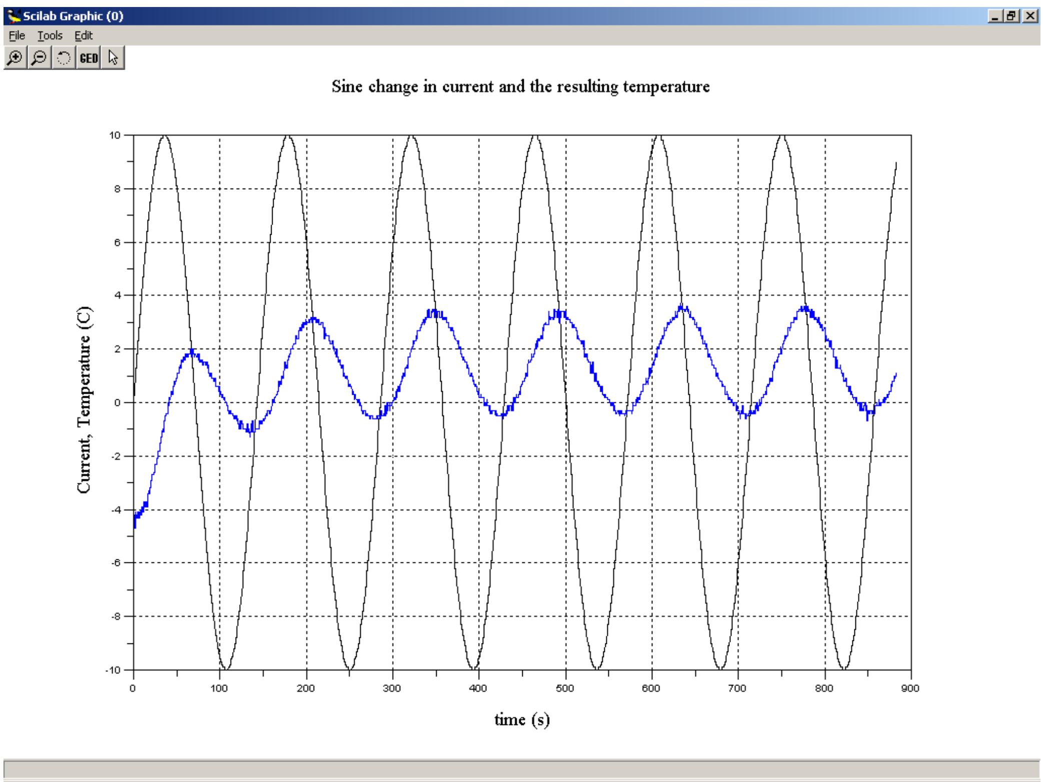

Figure 1.5: Plot of Input and Output vs time

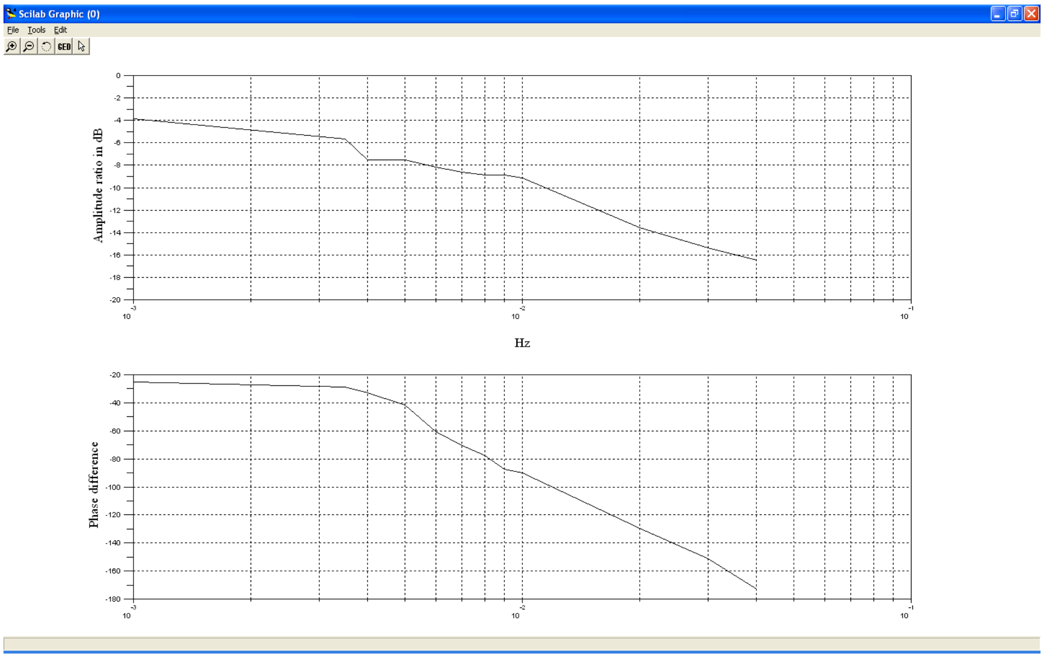

Figure 1.6: Bode Plot obtained from the plant

```
clear data7; exec('sec.007.txt'); getf('labelbode.sci'
   );
T = data7(:,1); fan = data7(:,3); //T is time, fan is
   fan speed
u = data7(:,2)-20; y = data7(:,4)-28; // u is current,
   y is temperature
f=0.007;
p=length(u);// u is current
data_width=3;
p2=(p/data_width);
p1=round(p2);
[q,k]=max(u((p-p1):p,1));//q is maximum input
   amplitude
x1=(p-p1)+k;//Input(Current)reference point(time)
ST=T(2,1)-T(1,1);// Sampling Time
x2=(x1+((1/ST)*(1/f)));
x2=round(x2);
```

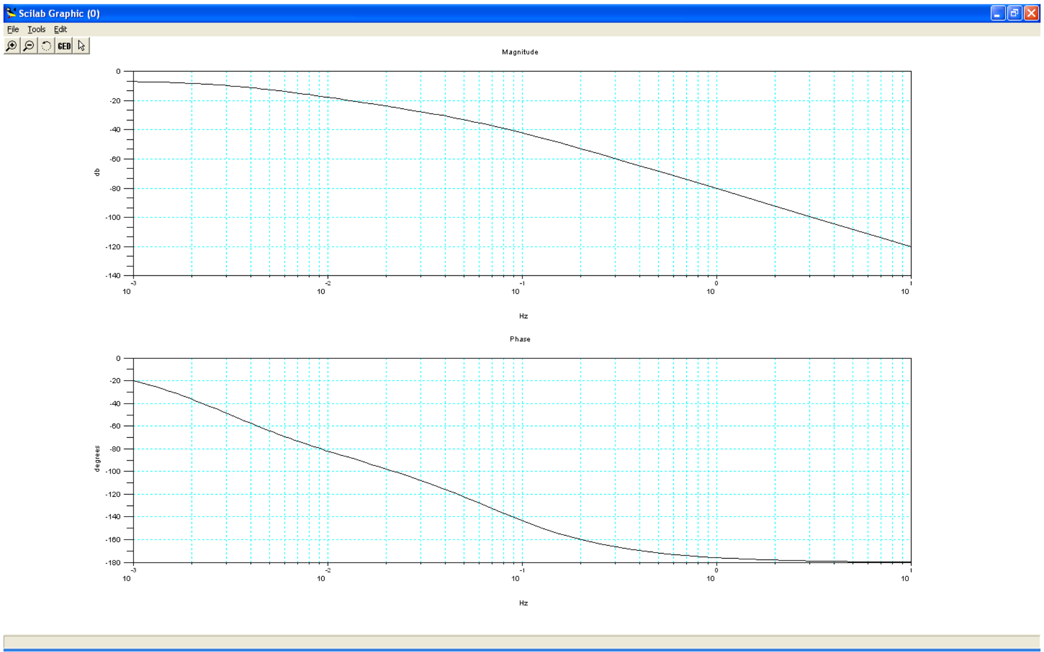

Figure 1.7: Bode Plot obtained through plants Transfer function

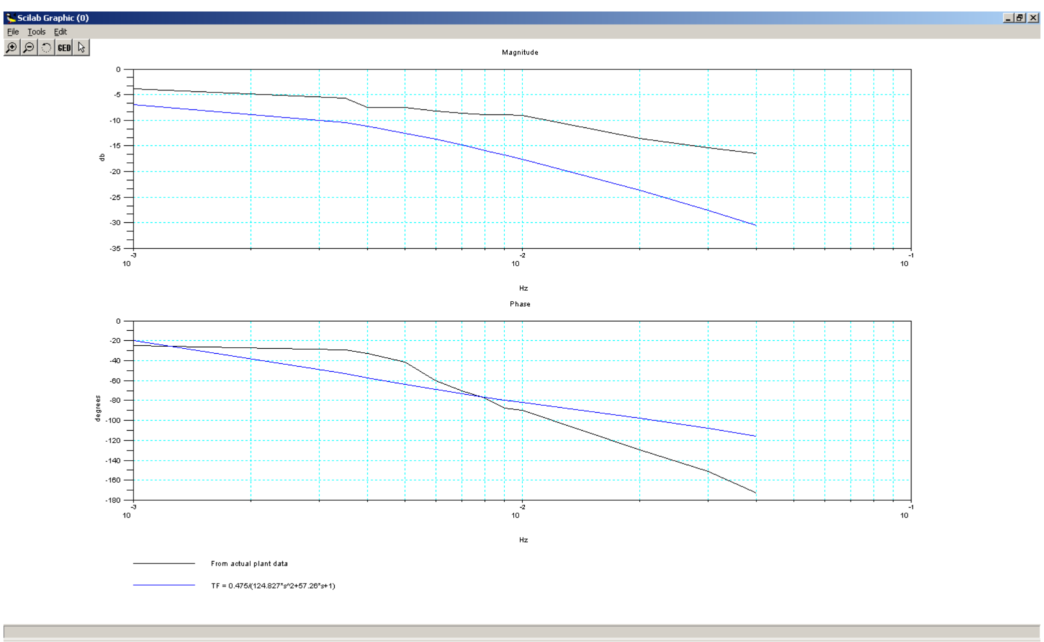

Figure 1.8: Comparison of Bode Plot's

```
[e,k1]=max(y(x1:x2,1)); //e is maximum output amplitude between x1 and x2
L=x1+k1;
t1=T(x1,1);
t2=T(L,1);
delt=(t2-t1);
Amplituderatio = (e/q);
Amplitude_ratio_in_dB=20*log10(e/q)
Phase_difference = -((delt)/(1/f))*360
ord = [u y]; x = [T T]; // u and y are plotted vs. time and time
xbasc(); plot2d(x,ord); xgrid();
title = 'Sine change in current and the resulting temperature'
label(title,4,'time (s)','Current, Temperature (C)',4);
```

Scilab Code 1.2 `label.sci`

```
// 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 1.3 `bodeplot.sce`

```
//bodeplot
getf('labelbode.sci');
x
  =[0.001,0.0035,0.004,0.005,0.006,0.007,0.008,0.009,0.01,0.02,0.03,0
  //Input frequency(Hz)
y
  =[-3.87,-5.67,-7.53,-7.53,-8.17,-8.64,-8.87,-8.90,-9.11,-13.55,-15.
  //Amplitude ratio(dB)
subplot(2,1,1);
plot2d(x,y,rect=[0.001,-20,0.04,0],logflag="ln");
xgrid();
y
  =[-25.2,-28.98,-33.11,-41.4,-60.48,-70.56,-77.76,-87.48,-90,-129.6,
  //Phase difference(degree)
title = ''
label(title,4,'Hz','Amplitude ratio in dB ',4);
subplot(2,1,2);
plot2d(x,y,rect=[0.001,-180,0.04,-20],logflag="ln");
label(title,4,'','Phase difference',4);
subplot(2,1,2);
xgrid();

s=poly(0,'s')
h=syslin('c',(0.475/(124.827*s^2+57.26*s+1)))
bode(h,0.001,0.04);
```

Scilab Code 1.4 `labelbode.sci`

```
// 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 1.5 `TFbode.sce`

```
s=poly(0,'s')
h=syslin('c',(0.475/(124.827*s^2+57.26*s+1)))
bode(h,0.001,10);
```