UNIVERSITY OF NOTRE DAME
Aerospace and Mechanical Engineering

AME 437: Control Systems Engineering
Homework 3 Solutions

B. Goodwine
February 17, 2003

1. (3.8.1:1) Given
$$y'''(t) + 4y''(t) + 3y'(t) = u(t),$$
and assuming all zero initial conditions, taking the Laplace transform gives
$$s^3Y(s) + 4s^2Y(s) + 3sY(s) = U(s).$$
Rearranging gives the answer
$$\frac{Y(s)}{U(s)} = \frac{1}{s^3 + 4s^2 + 3s}.$$

2. (3.8.1:3) Skipping the details,
$$\frac{Y(s)}{U(s)} = \frac{1}{s^3 + 5s^2 + 12s + 8}.$$

3. (3.8.1:5) Skipping the details,
$$\frac{Y(s)}{U(s)} = \frac{1}{s^3 + 5s^2 + 8s + 6}.$$

4. (3.8.1:7) Skipping the details,
$$\frac{Y(s)}{U(s)} = \frac{1}{s^2 + 11s + 10}.$$

5. Finding transfer functions experimentally.

   (a) For file 'data1.d'. First, load the data into Matlab and plot it.

```
>> load data1.d
>> x = data1;
>> length(x)

ans =

   101

>> t = linspace(0,10,101);
>> t = t';
>> plot(t,x);
```

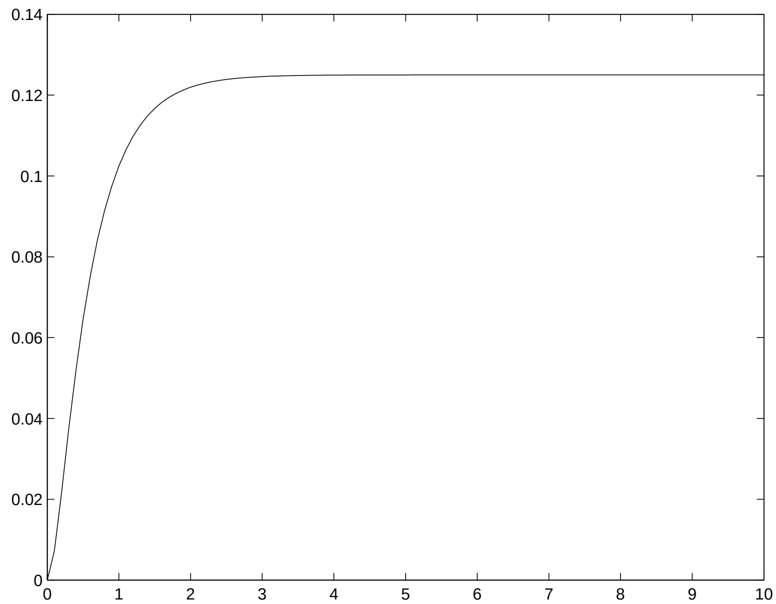

Figure 1. Plot of the data from file 'data1.d'.

The plot is illustrated in Figure 1.
Knowing that the step response is of the form

$$Y(s) = \frac{A}{s} + \frac{B}{s+p_1} + \frac{C}{s+p_2},$$

we can directly read from the plot that $A \approx 0.125$.
Now,

```
>> A = 0.125;
>> plot(t,log(A-x));
Warning: Log of zero.
>> grid on
```

produces Figure 2. From the figure, we can read that the slope of the log plot is -2. Therefore, $p_1 = 2$.
Using the formula,

$$B = \frac{y(t) - A}{e^{-p_1 t}}$$

```
>> p1 = 2;
>> plot(t,(x-A)./exp(-p1*t));
>> grid on
```

we can read from Figure 3 that $B \approx -0.166$.
The rest follow from the formulas:

$$C = -(A+B) = 0.0417,$$

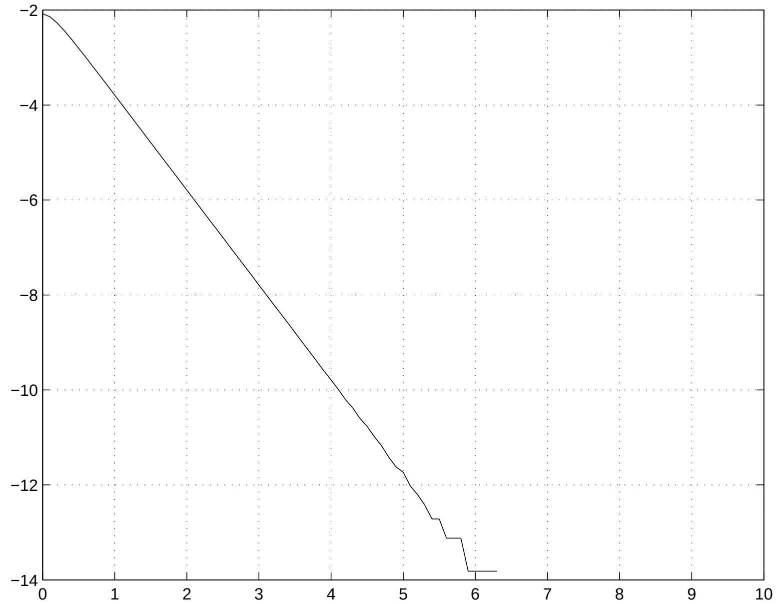

Figure 2. Plot of $\log(A - y(t))$.

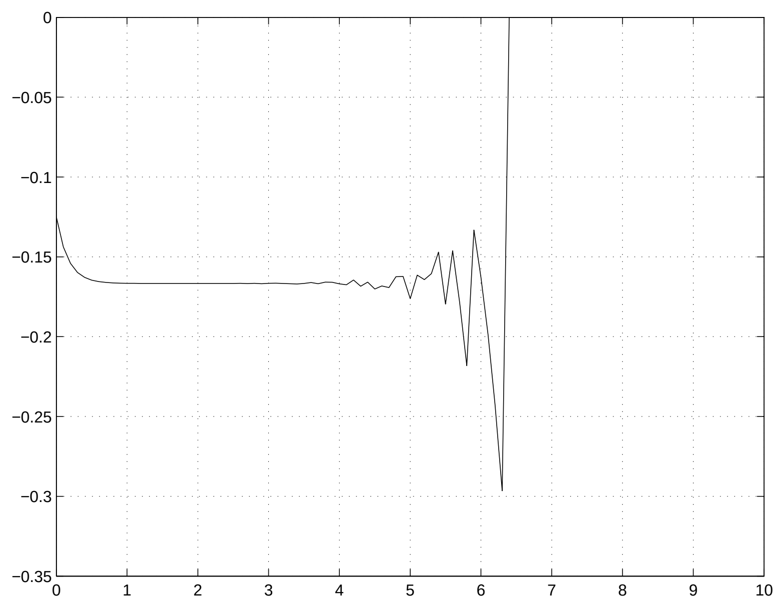

Figure 3. Plot of $(y(t) - A)/\exp(-p_1 t)$.

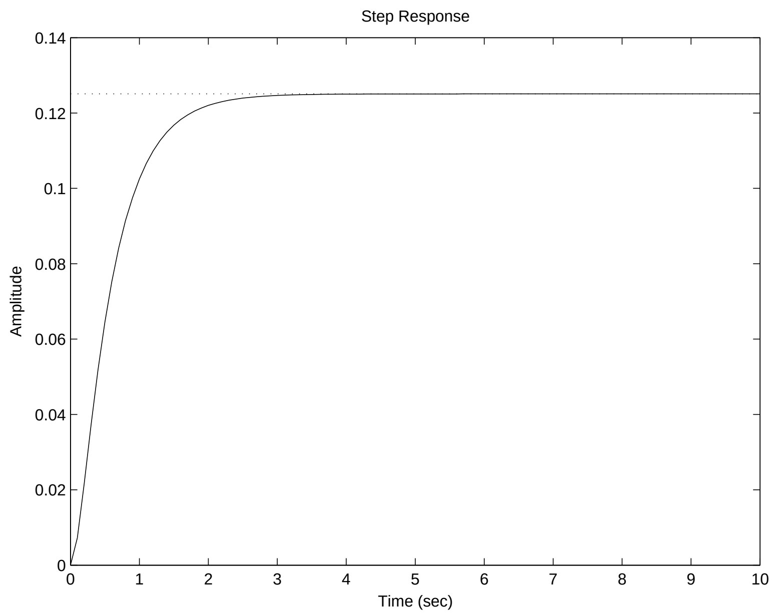

Figure 4. Plot of the step response after determining K, p_1 and p_2.

$$p_2 = -\frac{B}{C}p_1 = 8,$$
$$K = Ap_1p_2 = 2.$$

Figure 4 shows a plot that results from

```
>> step(K,conv([1 p1],[1 p2],10))
```

(b) With less detail this time,

```
>> load data2.d
>> x = data2;
>> length(x)

ans =

   101

>> t = linspace(0,10,101);
>> t = t';
>> plot(t,x)
```

produces Figure 5. Thus, $A = 1$.

```
>> A = 1;
>> plot(t,log(A-x))
```

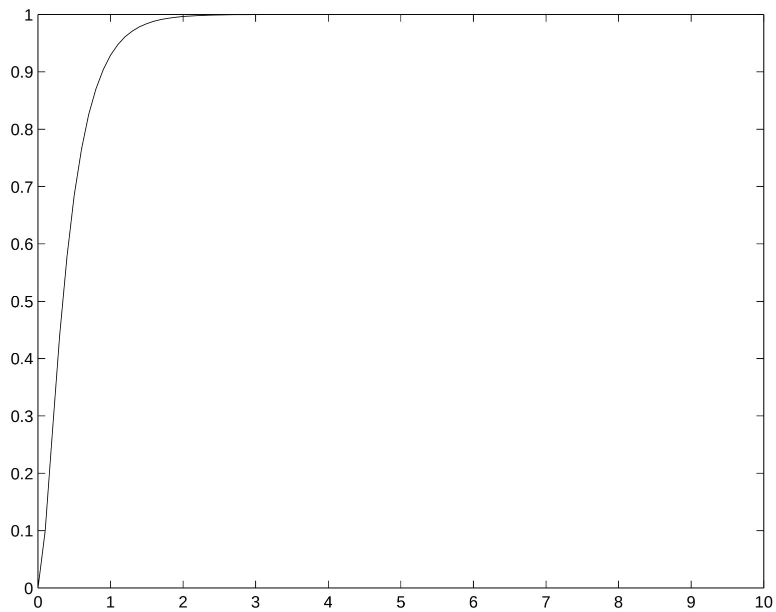

Figure 5. Plot of the data from file 'data1.d'.

```
Warning: Log of zero.
>> grid on
```

produces Figure 6. Thus, $p_1 = 3$.

```
>> p1 = 3;
>> plot(t,(x-A)./exp(-p1*t))
>> grid on
```

produces Figure 7. Thus, $B = -1.44$.

```
>> B = -1.44;
>> C = -(A+B)

C =

    0.4400

>> p2 = -B/C*p1

p2 =

    9.8182

>> K = A*p1*p2
```

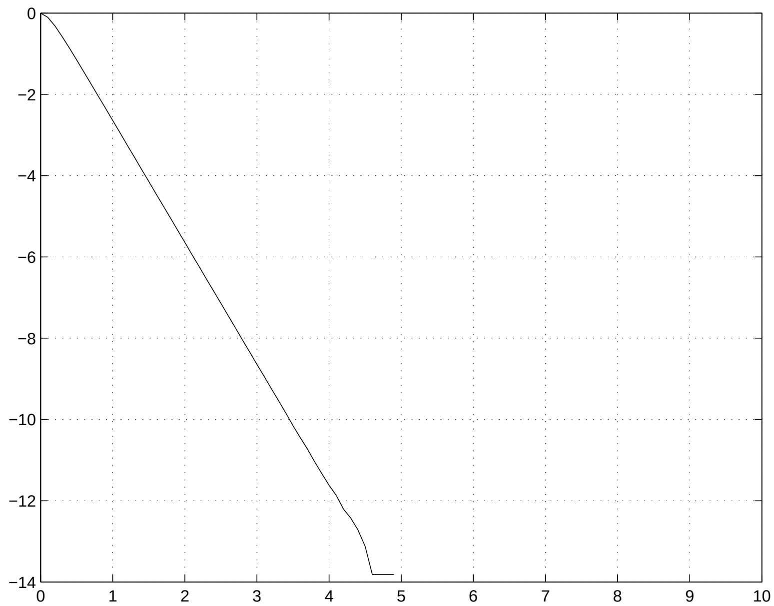

Figure 6. Plot of $\log(A - y(t))$.

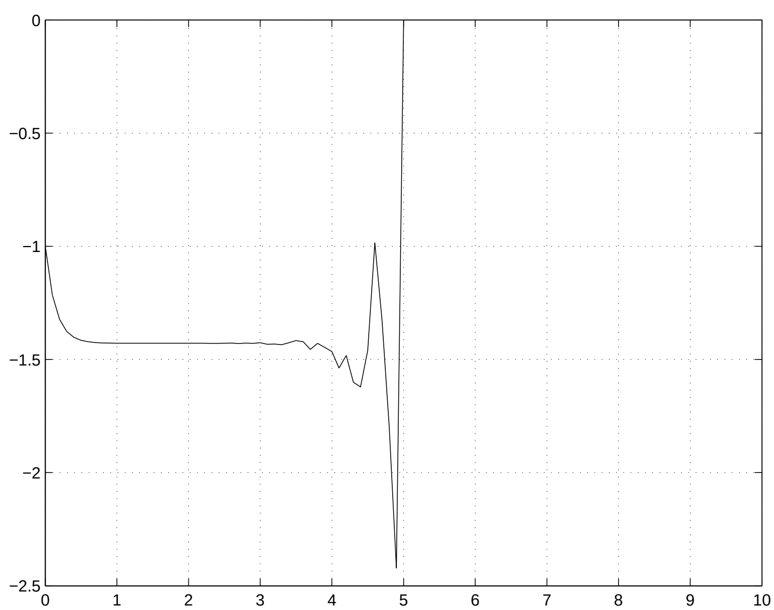

Figure 7. Plot of $(y(t) - A)/\exp(-p_1 t)$.

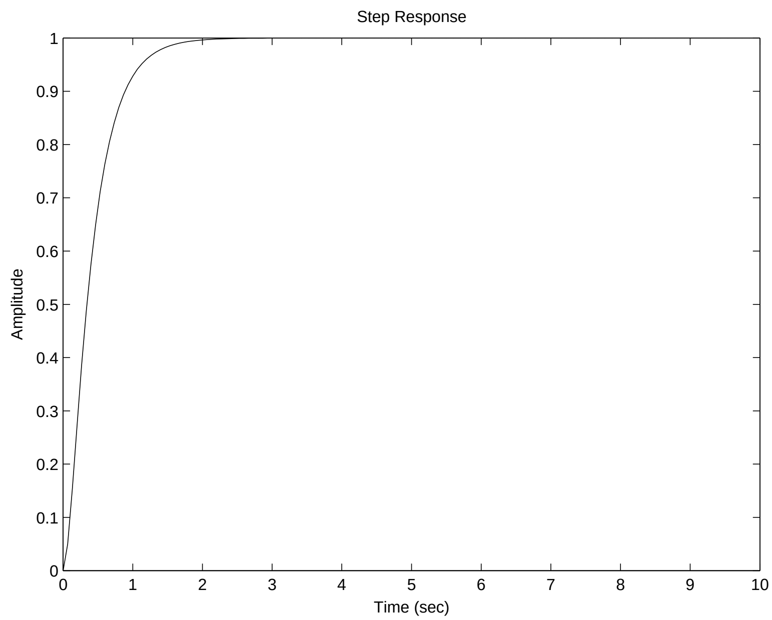

Figure 8. Plot of the step response after determining K, p_1 and p_2.

```
K =

   29.4545

>> step(K,conv([1 p1],[1 p2]),10)
```

produces produces Figure 8, which appears to be close to the data plot in Figure 5.

6. (3.8.8:5)

   (a) For

$$G(s) = \frac{16}{(s+2)(s+4)}$$

```
>> K = 16;
>> p1 = 2;
>> p2 = 4;
>> A = K/(p1*p2);
>> B = K/(p1*(p1-p2));
>> C = K/(p2*(p2 - p1));
>> step(k,conv([1 p1],[1 p2]))
>> t = linspace(0,3,100);
>> plot(t,A+B*exp(-p1*t))
>> hold on
>> plot(t,exp(-p2*t))
```

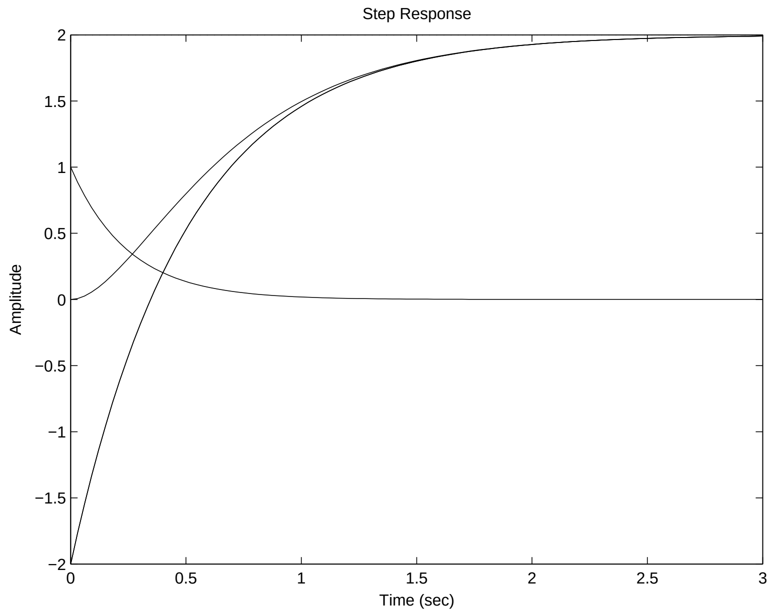

Figure 9. Decomposing the solution for 3.8.8:5(a).

```
>> step(K,conv([1 p1],[1 p2]),3)
```

produces Figure 9.

(b) For

$$G(s) = \frac{24}{(s+2)(s+6)}$$

```
>> K = 24;
>> p1 = 2;
>> p2 = 6;
>> A = K/(p1*p2);
>> B = K/(p1*(p1-p2));
>> C = K/(p2*(p2 - p1));
>> t = linspace(0,3,100);
>> plot(t,A+B*exp(-p1*t))
>> hold on
>> plot(t,C*exp(-p2*t))
>> step(K,conv([1 p1],[1 p2]))
```

produces Figure 10.

(c) For

$$G(s) = \frac{32}{(s+2)(s+8)}$$

```
>> K = 32;
>> p1 = 2;
```

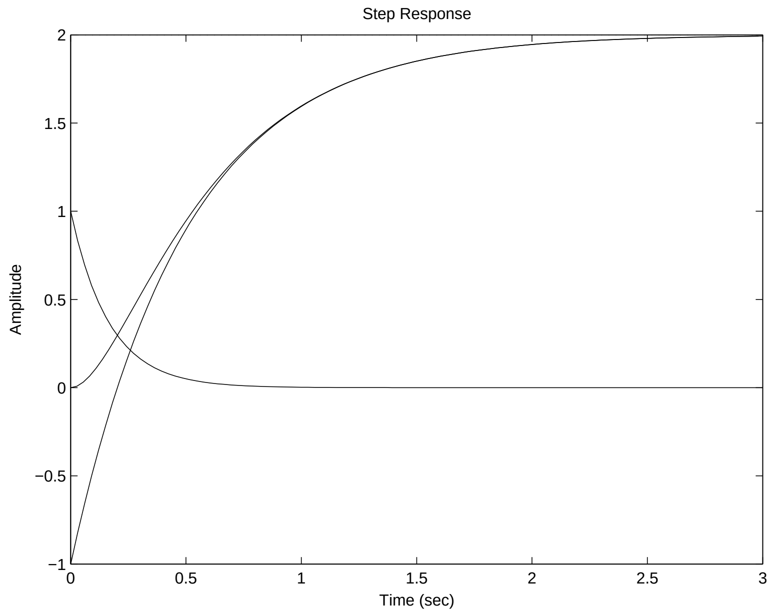

Figure 10. Decomposing the solution for 3.8.8:5(b).

```
>> p2 = 8;
>> A = K/(p1*p2);
>> B = K/(p1*(p1-p2));
>> C = K/(p2*(p2 - p1));
>> t = linspace(0,3,100);
>> plot(t,A+B*exp(-p1*t))
>> hold on
>> plot(t,C*exp(-p2*t))
>> step(K,conv([1 p1],[1 p2]))
```

produces Figure 11.

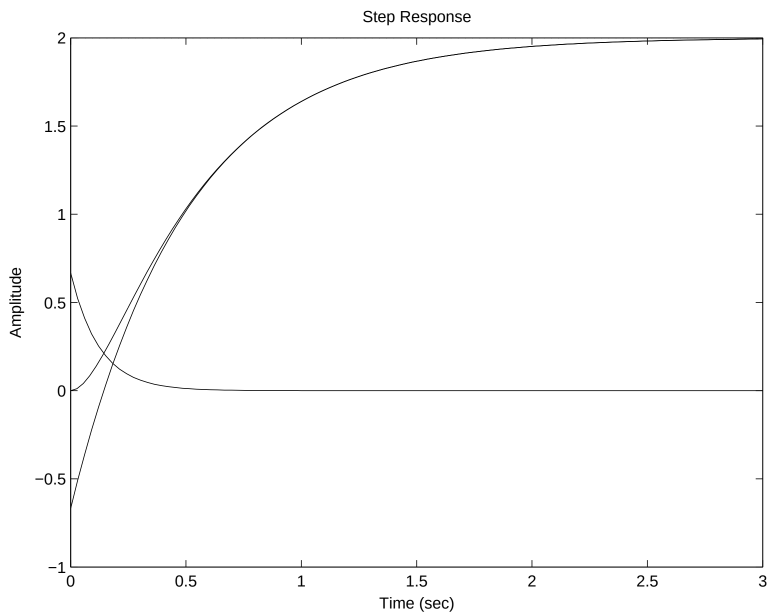

Figure 11. Decomposing the solution for 3.8.8:5(c).

7. The following C code implements the fourth order Runge-Kutta routine for the equation

$$\ddot{x} + 0.25\dot{x} - x + x^3 = 0.3\cos t.$$

For the problems assigned, you would have to have change the l1 through l4 lines as well as the initial conditions.

```
#include<stdio.h>
#include<math.h>

main() {

  float x[2],dt,tfinal,t;
  float k1,k2,k3,k4,l1,l2,l3,l4;
  FILE *fp;

  fp = fopen("rk.d","w");
  tfinal = 50.0;
  dt = 0.001;

  x[0] = 0.0;
  x[1] = 0.1;

  for(t=0;t<=tfinal;t+=dt) {
```

```
    k1 = dt*x[1];
    l1 = dt*(-.25*x[1] + x[0] - pow(x[0],3) + 0.3*cos(t));

    k2 = dt*(x[1] + k1/2);
    l2 = dt*(-.25*(x[1]+l1/2)+x[0]+k1/2-pow(x[0]+k1/2,3)+0.3*cos(t+dt/2));

    k3 = dt*(x[1]+k2/2);
    l3 = dt*(-.25*(x[1]+l2/2)+(x[0]+k2/2)-pow(x[0]+k2/2,3)+0.3*cos(t+dt/2));

    k4 = dt*(x[1]*k3);
    l4 = dt*(-.25*(x[1]+l3)+(x[0]+k3)-pow(x[0]+k3,3)+0.3*cos(t+dt));

    x[0] += (k1+k2*2+k3*2+k4)/6.0;
    x[1] += (l1+l2*2+l3*2+l4)/6.0;

    fprintf(fp,"%f\t%f\t%f\n",t,x[0],x[1]);
  }

  fclose(fp);
}
```