Circuit Analysis with LaPlace Transforms

Background

Phasors allow you to analyze circuits with inductors and capacitors just like you analyze resistor circuits - the only difference is you need to use complex numbers. With phasor analysis, the basic assumption is that all functions are in the form of

$$x = a \cdot e^{j\omega t}$$

resulting in the phasor impedance for an inductor and capacitor being:

$$L \rightarrow j\omega L$$

$$C \rightarrow \frac{1}{j\omega C}$$

In contrast, LaPlace transforms assume all functions are in the form of

$$x(t) = a \cdot e^{st}$$

resulting in the LaPlace impedance being:

$$L \rightarrow Ls$$

$$C \rightarrow \frac{1}{Cs}$$

Component	Phasor Impedance	LaPlace Impedance
R	R	R
L	jwL	Ls
C	1 / jwC	1 / Cs

With LaPlace impedance's, everything that worked in Circuits I and II still apply:

- Impedance's in series add: A resistor, inductor, and capacitor in series have an impedance of:

$$Z = R + Ls + \frac{1}{Cs}$$

- Impedance's in parallel add as the sum of the inverses, inverted. A resistor, inductor, and capacitor in parallel have a total impedance of

$$Z = \left(\frac{1}{R} + \frac{1}{Ls} + \frac{1}{1/Cs}\right)^{-1}$$

- Current loops still work: The sum of the voltages around any closed path has to add to zero.
- Voltage nodes still work: The sum of the current from a node must add to zero.

There is a short-cut for analyzing electrical circuits using LaPlace transforms, however. This is to use a formulation called *state space*.

State Space

One way to describe a dynamic system is with a transfer function:

$$Y = G(s) \cdot U$$

Another way is to put the system in matrix form (called state space). If the energy in the system is defined by vector X (think of X as the voltages on capacitors and current in inductors: things that define the energy in the system), then the change in energy along with the output as function of the system state can be written as

$$X' = AX + BU$$

$$Y = CX + DU$$

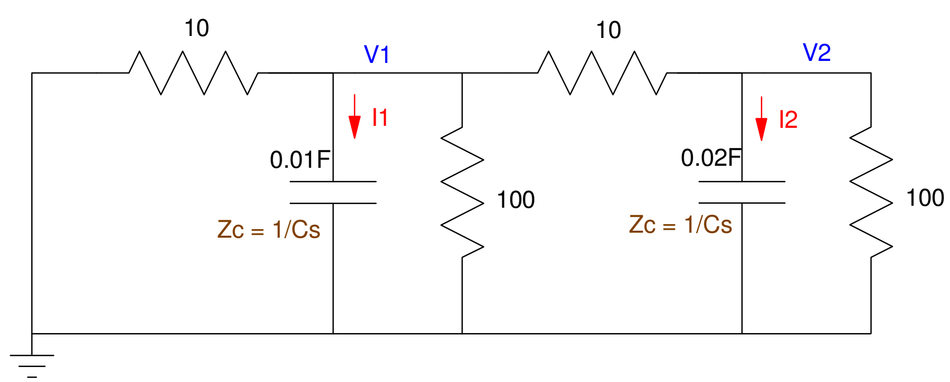
With state-space, there is a third way to input a dynamic system into Matlab:

```
G = ss(A, B, C, D);
```

It's probably easiest to explain this with examples.

State-Space and Natural Responses

Example 1: For the following circuit, find the voltage, y(t) = v4(t) assuming v1(0) = v2(0) = 10.



2nd-Order System: (there are two energy storage elements)

Step 1: Define the system states.

This is the voltage across the capacitors and the current through inductors. This defines the energy in the system.

$$X = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}$$

Step 2: Define the change in energy in terms of the input (none here) and the system states

From the equations for a capacitor

$$i_c = C\frac{dv}{dt}$$

In the LaPlace domain:

$$I_c = C(sV - v(0))$$

Defining the current to each capacitor in terms of the states

$$I_1 = 0.01(sV_1 - v_1(0)) = \left(\frac{0-V_1}{10}\right) + \left(\frac{V_2-V_1}{10}\right) + \left(\frac{0-V_1}{100}\right)$$

$$I_2 = 0.02(sV_2 - v_2(0)) = \left(\frac{0-V_2}{100}\right) + \left(\frac{V_1-V_2}{10}\right)$$

Group terms:

$$sV_1 = -21V_1 + 10V_2 + v_1(0) \qquad * 21$$

$$sV_2 = 5V_1 - 5.5V_2 + v_2(0) \qquad * 5$$

Step 3: Solve for y = V2. Place this in matrix (state-space form)

$$s\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} -21 & 10 \\ 5 & -5.5 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix}$$

$$Y = V_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + [0]$$

In Matlab:

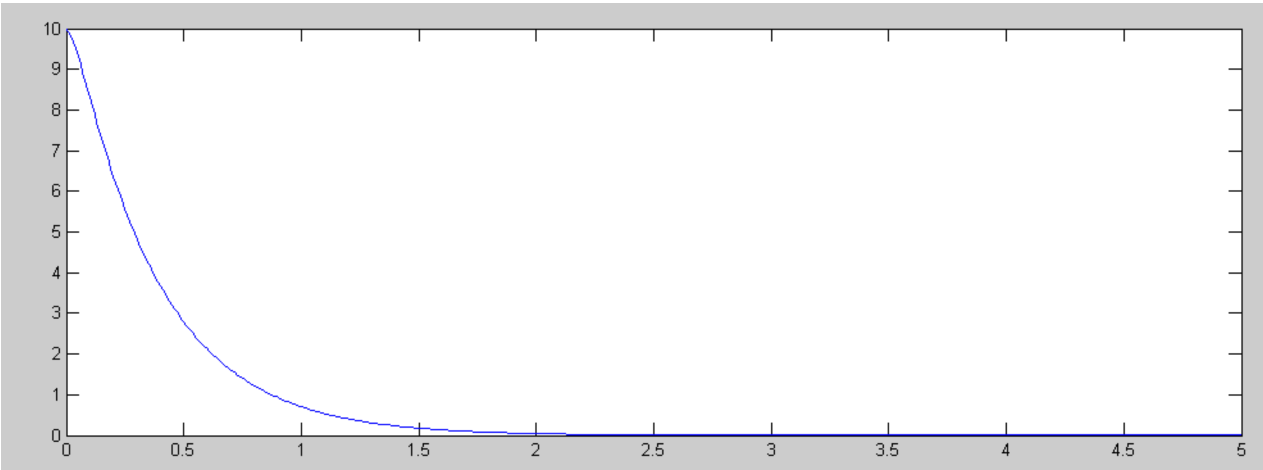
```
>> A = [-21,10 ; 5, -5.5]

  -21.0000   10.0000
    5.0000   -5.5000

>> B = [10 ; 10]

    10
    10

>> C = [0,1];
>> D = 0;
>> Y = ss(A,B,C,D);
```

At this point you can solve for Y(s):

```
>> zpk(Y)

              10 (s+26)
Y(s) = -------------------
        (s+23.74) (s+2.759)
```

y(t) is found using the impulse response function for Y(s)

```
t = [0:0.01:5]';
y = impulse(Y, t);
plot(t,y)
```



y(t) for initial conditions of v1(0) = v2(0) = 10

Eigenvalues and Eigenvectors: The eigenvalues of matrix X are the poles of the system. This tells you how the system behaves

```
>> eig(A)

  -23.7411
   -2.7589
```

The eigenvectors of A tell you what behaves each way:

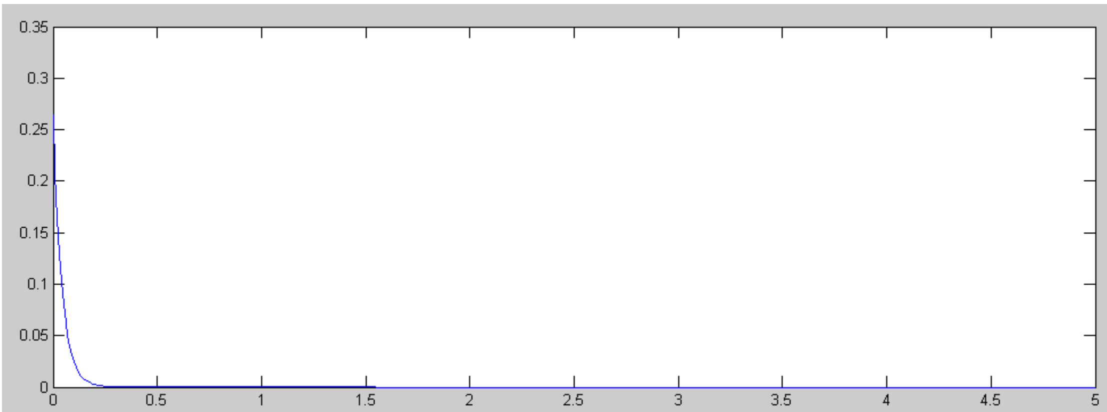
```
>> [a,b] = eig(A)

a =
   -0.9644   -0.4807
    0.2644   -0.8769

b =
  -23.7411         0
         0   -2.7589
```

If the initial condition was $\begin{bmatrix} -0.9644 \\ 0.2644 \end{bmatrix}$ (or a scalar multiple of this), then y(t) decays as $e^{-23.74t}$

If the initial condition was $\begin{bmatrix} -0.4807 \\ -0.8679 \end{bmatrix}$ (or a scalar multiple of this), then y(t) decays as $e^{-2.7589t}$

To illustrate this, using the fast (first) eigenvector:

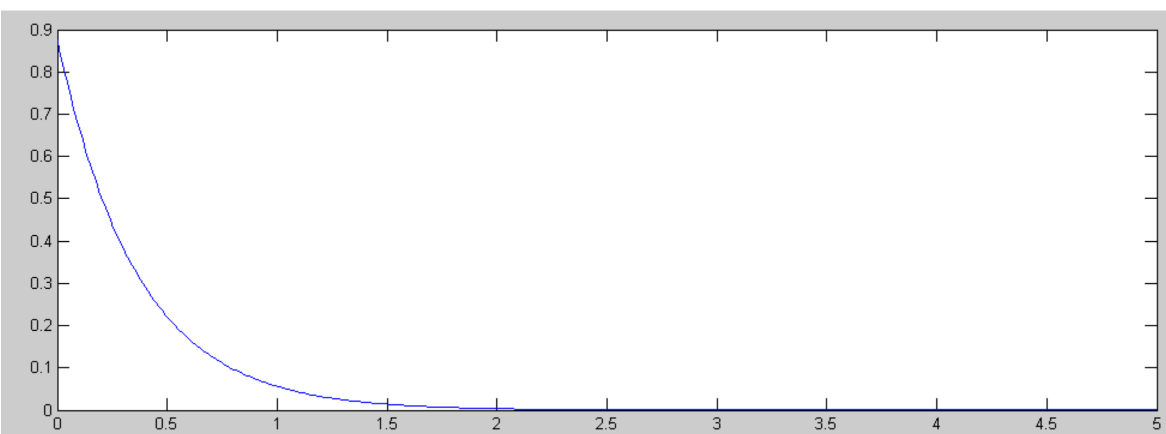
```
>> B = a(:,1)

   -0.9644
    0.2644

>> Y = ss(A,B,C,D);
>> y = impulse(Y, t);
>> plot(t,y)
```



y(t) when the initial conditions are equal to the fast eigenvector

Using the slow (second) eigenvector:

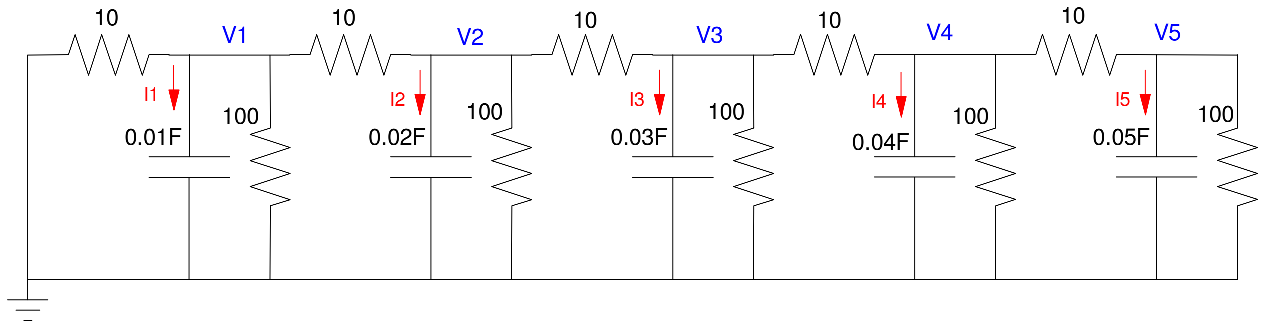
```
>> B = -a(:,2)

    0.4807
    0.8769

>> Y = ss(A,B,C,D);
>> y = impulse(Y, t);
>> plot(t,y)
```



y(t) when the initial conditions are equal to the slow eigenvector

Example 2: 5-stage RC filter.

Find V5(t) if the initial condition is v1(0) = v2(0) = v3(0) = v4(0) = v5(0) = 10V



This is where state-space really shines. You could use voltage nodes or current loops and solve for V5. That will take about two hours. It's much easier with state space.

Step 1: Define the state variables. The energy in the system is defined by

$$X = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix}$$

Step 2: Define the change in the state variables in terms of the other states

$$I_1 = 0.01\frac{dV_1}{dt} = 0.01(sV_1 - v_1(0)) = \left(\frac{0-V_1}{10}\right) + \left(\frac{0-V_1}{100}\right) + \left(\frac{V_2-V_1}{10}\right)$$

$$I_2 = 0.02\frac{dV_2}{dt} = 0.02(sV_2 - v_2(0)) = \left(\frac{V_1-V_2}{10}\right) + \left(\frac{0-V_2}{100}\right) + \left(\frac{V_3-V_2}{10}\right)$$

$$I_3 = 0.03\frac{dV_3}{dt} = 0.03(sV_3 - v_3(0)) = \left(\frac{V_2-V_3}{10}\right) + \left(\frac{0-V_3}{100}\right) + \left(\frac{V_4-V_3}{10}\right)$$

$$I_4 = 0.04\frac{dV_4}{dt} = 0.04(sV_2 - v_2(0)) = \left(\frac{V_3-V_4}{10}\right) + \left(\frac{0-V_4}{100}\right) + \left(\frac{V_5-V_4}{10}\right)$$

$$I_5 = 0.05\frac{dV_5}{dt} = 0.05(sV_5 - v_5(0)) = \left(\frac{V_4-V_5}{10}\right) + \left(\frac{0-V_5}{100}\right)$$

Solve for the derivative

$$sV_1 = -21V_1 + 10V_2 + v_1(0)$$
$$sV_2 = 5V_1 - 10.5V_2 + 5V_3 + v_2(0)$$
$$sV_3 = 3.33V_2 - 7V_3 + 3.33V_4 + v_3(0)$$
$$sV_4 = 2.5V_3 - 5.25V_4 + 2.5V_5 + v_4(0)$$
$$sV_5 = 2V_4 - 2.2V_5 + v_5(0)$$

Place in matrix form

$$s\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} = \begin{bmatrix} -21 & 10 & 0 & 0 & 0 \\ 5 & -10.5 & 5 & 0 & 0 \\ 0 & 3.33 & -7 & 3.33 & 0 \\ 0 & 0 & 2.5 & -5.25 & 2.5 \\ 0 & 0 & 0 & 2 & -2.2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \end{bmatrix} + \begin{bmatrix} v_1(0) \\ v_2(0) \\ v_3(0) \\ v_4(0) \\ v_5(0) \end{bmatrix}$$

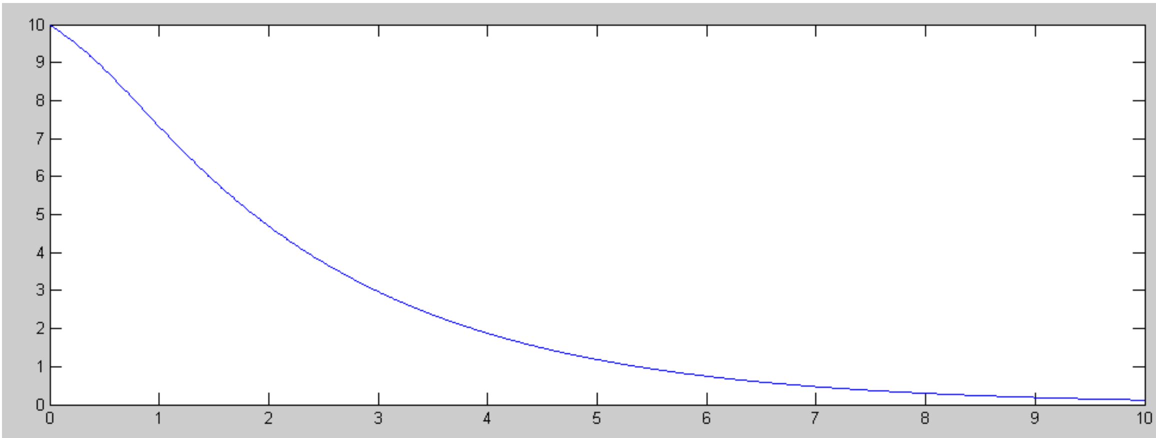
$$Y = V_5 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix} V_{1..5} + [0]$$

Step 3: Find Y(s)

```
A = [-21,10,0,0,0 ; 5,-10.5,5,0,0 ; 0,3.333,-7,3.333,0 ; 0,0,2.5,-5.25,2.5 ; 0,0,0,2,-2.2]

  -21.0000   10.0000         0         0         0
    5.0000  -10.5000    5.0000         0         0
         0    3.3330   -7.0000    3.3330         0
         0         0    2.5000   -5.2500    2.5000
         0         0         0    2.0000   -2.2000

B = [10;10;10;10;10]

    10
    10
    10
    10
    10

C = [0,0,0,0,1];
D = 0;
Y = ss(A,B,C,D);
zpk(Y)


              10 (s+24.75) (s+11.46) (s+6.061) (s+3.48)
Y(s = --------------------------------------------------
       (s+24.76) (s+11.31) (s+6.601) (s+2.822) (s+0.4601)
```

Step 4: Find y(t).

```
t = [0:0.01:10]';
y = impulse(G,t);
plot(t,y)
```



Natural Response v5(t) for initial condition of {10, 10, 10, 10, 10}

Sidelight:

- Find the initial condition which decays as slow as possible.
- Find the initial condition which decays as slow as possible.

Solution: This is asking for the fast and slow eigenvector.

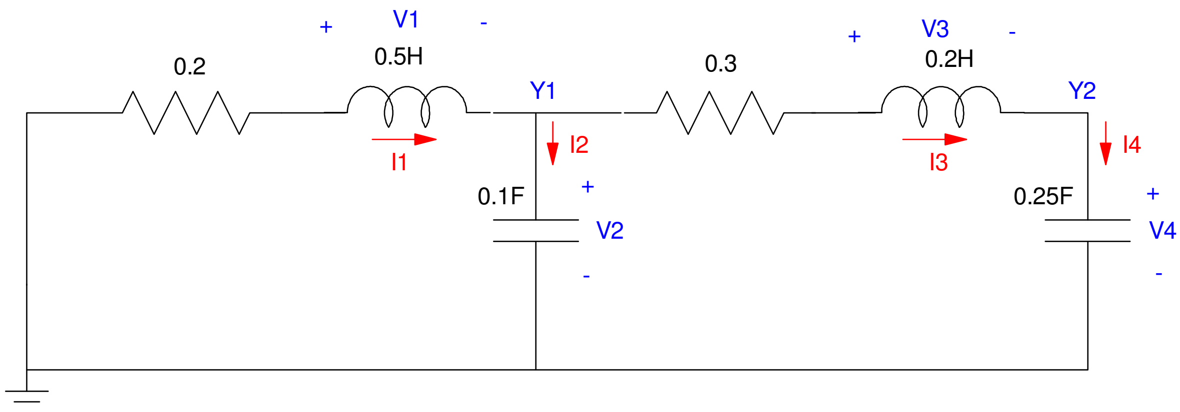
```
>> [a,b] = eig(A)

a =
   -0.9339   -0.5296   -0.4249   -0.3229    0.1366
    0.3511   -0.5133   -0.6118   -0.5870    0.2807
   -0.0675    0.6124   -0.0521   -0.5785    0.4269
    0.0088   -0.2780    0.6056   -0.1382    0.5570
   -0.0008    0.0610   -0.2752    0.4443    0.6403


b =

  -24.7599         0         0         0         0
         0  -11.3067         0         0         0
         0         0   -6.6013         0         0
         0         0         0   -2.8219         0
         0         0         0         0   -0.4601
```

- The slow eigenvector is in red. This mode decays as exp(-0.46t)
- The fast eigenvector is in blue. This mode decays as exp(-24.76t)

Example 3: RLC Circuit

Find V4(t) assuming

- v2(0) = v4(0) = 10V
- i1(0) = i3(0) = 2A



Solution: Use state-space (current loops or voltage nodes alwo work but state-space is easier if you have access to Matlab)

Step 1: Define the state variables. These define the energy in the system

$$X = \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix}$$

Step 2: Define the change in the states. This comes from

$$v = L\frac{di}{dt}$$

$$i = C\frac{dv}{dt}$$

which leads to

$$v_1 = 0.5\frac{di_i}{dt} = 0.5(sI_1 - i_1(0)) = (0 - 0.2I_1) - V_2$$

$$i_2 = 0.1\frac{dv_2}{dt} = 0.1(sV_2 - v_2(0)) = I_1 - I_3$$

$$v_3 = 0.2\frac{di_3}{dt} = 0.2(sI_3 - i_3(0)) = V_2 - 0.3I_3 - V_4$$

$$i_4 = 0.25\frac{dv_4}{dt} = 0.25(sV_4 - v_4(0)) = I_3$$

Step 3: Rewrite these equations as

$$sI_1 = -0.4I_1 - 2V_2 + i_1(0)$$

$$sV_2 = 10I_1 - 10I_3 + v_2(0)$$

$$sI_3 = 5V_2 - 1.5I_3 - 5V_4 + i_3(0)$$

$$sV_4 = 4I_3 + v_4(0)$$

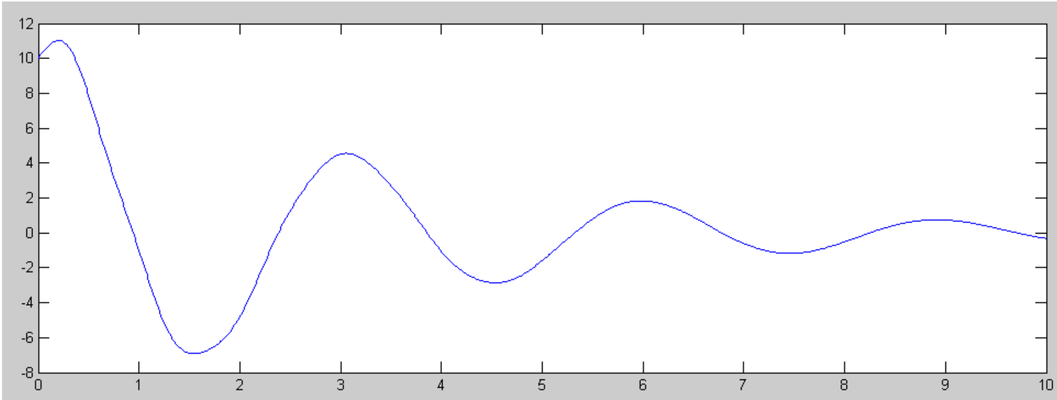
Place in matrix form

$$s\begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} -0.4 & -2 & 0 & 0 \\ 10 & 0 & -10 & 0 \\ 0 & 5 & -1.5 & -5 \\ 0 & 0 & 4 & 0 \end{bmatrix}\begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} i_1(0) \\ v_2(0) \\ i_3(0) \\ v_4(0) \end{bmatrix}$$

$$Y = V_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + [0]$$

Solve in Matlab

```
>> A = [-0.4,-2,0,0 ; 10,0,-10,0 ; 0,5,-1.5,-5 ; 0,0,4,0]

   -0.4000   -2.0000         0         0
   10.0000         0  -10.0000         0
         0    5.0000   -1.5000   -5.0000
         0         0    4.0000         0

>> B = [2 ; 10 ; 2 ; 10]

     2
    10
     2
    10

>> C = [0,0,0,1];
>> D = 0;
>> Y = ss(A,B,C,D);
>> zpk(Y)


              10 (s+1.279) (s^2 + 1.421s + 89.1)
Y(s) = -------------------------------------------
       (s^2 + 0.6102s + 4.7) (s^2 + 1.29s + 85.11)


>> t = [0:0.01:10]';
>> plot(t,y)
```



y(t) = v4(t)

Just for fun,

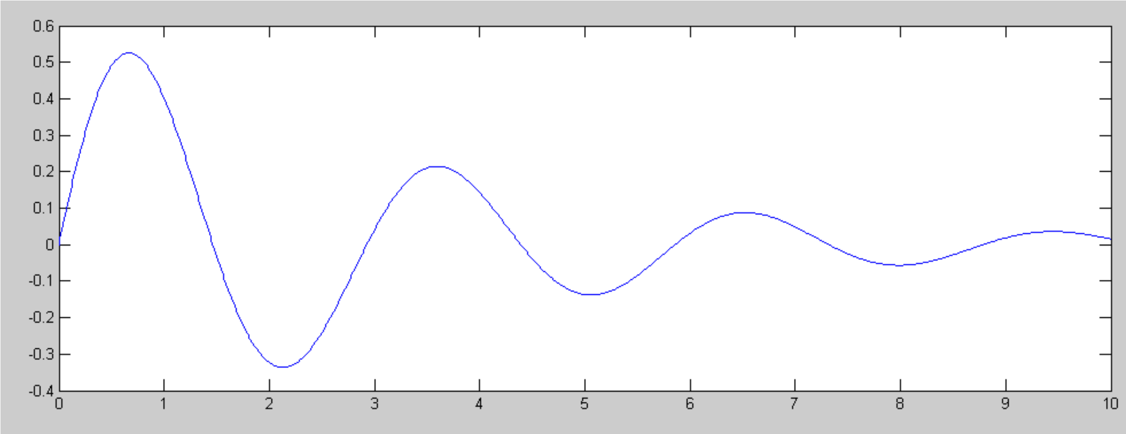
- Find the initial conditions which make this system decay as slow as possible.
- Find the initial conditions which make this system decay as slow as possible.

Solution: Find the eigenvalues and eigenvectors

```
>> [m,v] = eig(A)

m =

   0.0045 + 0.1696i   0.0045 - 0.1696i   0.0778 - 0.4519i   0.0778 + 0.4519i
   0.7808             0.7808            -0.4887 - 0.0621i  -0.4887 + 0.0621i
   0.0549 - 0.5490i   0.0549 + 0.5490i   0.0496 - 0.3489i   0.0496 + 0.3489i
  -0.2391 - 0.0071i  -0.2391 + 0.0071i  -0.6503            -0.6503


v =

  -0.6449 + 9.2031i        0                  0                  0
        0            -0.6449 - 9.2031i        0                  0
        0                  0            -0.3051 + 2.1463i        0
        0                  0                  0            -0.3051 - 2.1463i
```

Slow: Make the initial conditions equal to the slow eigenvector. This is complex, so you can use the real part(to get cosine) or imaginary part (to get sine).

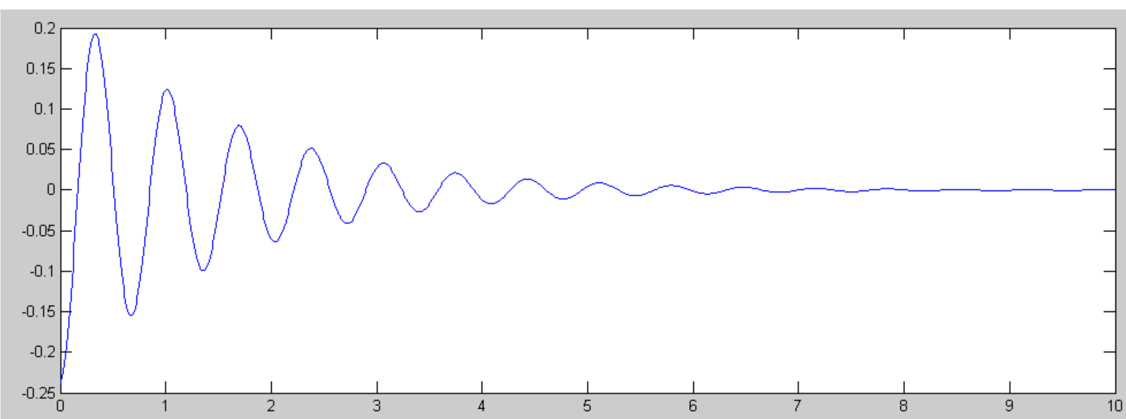
```
X0 = imag(m(:,4))

    0.4519
    0.0621
    0.3489
         0

Y = ss(A,X0,C,D);
y = impulse(Y, t);
plot(t,y)
```



V4(t) when the initial condition is equal to the slow eigenvector

Fast: Make the initial condition equal to the fast eigenvector. Again, use the real part for cosine, imaginary part for sine.

```
>> X0 = real( m(:,1) )

X0 =

    0.0045
    0.7808
    0.0549
   -0.2391

>> Y = ss(A,X0,C,D);
>> y = impulse(Y, t);
>> plot(t,y)
```



V4(t) when the initial condition is equal to the fast eigenvector