

Electric Circuits & The Heat Equation

Objective:

- Be able to write the voltage node equations for an electrical circuit
- Be able to place these dynamics in state-space form
- Be able to find the transfer function for an electrical circuit

Electronic Circuits:

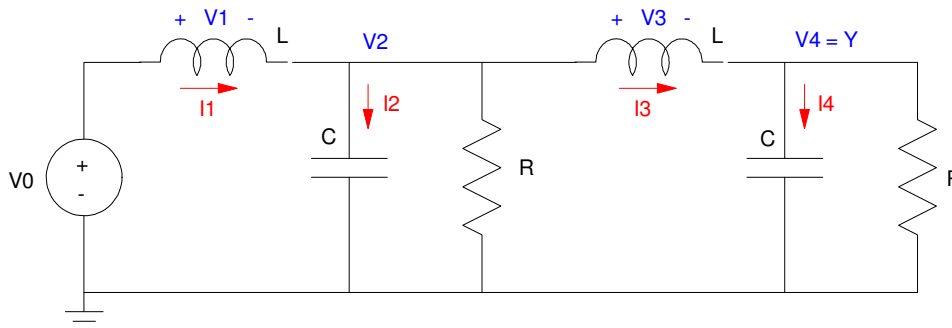
Circuits with elements which store energy (inductors and capacitors) are dynamic systems (as opposed to static systems which have only resistors). The transfer function for a circuit can be obtained by substituting the LaPlace value of the element's impedance:

	V/I Relationship	LaPlace value of Resistance
Resistor	$V = I R$	R
Capacitor	$V = \frac{1}{C} \int I \cdot dt$	$1 / sC$
Inductor	$V = L \frac{dI}{dt}$	$L s$

Current loops, voltage nodes, or other circuit analysis techniques can then be used to find the transfer function. It's a *lot* easier to use state-space techniques, however.

Example:

Find the transfer function from V_i to V_o . Assume $L = 100\text{mH}$, $C = 10\text{mF}$, $R = 100$



To use state-space methods, write this as

$$sX = AX + BU$$

$$Y = CX + DU$$

X defines the energy in the system. This is

- Current for inductors $E = \frac{1}{2}LI^2$
- Voltage for capacitors $E = \frac{1}{2}CV^2$

Since there are four energy storing devices (inductors and capacitors), you need to write four equations. Using the equations for inductors and capacitors

$$V = L \frac{dI}{dt} \quad (\text{inductors})$$

$$I = C \frac{dV}{dt} \quad (\text{capacitors})$$

you can write these as follows:

$$V_1 = LsI_1 = X - V_2$$

$$I_2 = CsV_2 = I_1 - I_3 - \frac{V_2}{R}$$

$$V_3 = LsI_3 = V_2 - V_4$$

$$I_4 = CsV_4 = I_3 - \frac{V_4}{R}$$

Place in matrix form

$$s \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & \left(\frac{-1}{L}\right) & 0 & 0 \\ \left(\frac{1}{C}\right) & \left(\frac{-1}{RC}\right) & \left(\frac{-1}{C}\right) & 0 \\ 0 & \left(\frac{1}{L}\right) & 0 & \left(\frac{-1}{L}\right) \\ 0 & 0 & \left(\frac{1}{C}\right) & \left(\frac{-1}{RC}\right) \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{L}\right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Since $y = V_4$

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

Plugging in numbers

$$s \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & -10 & 0 & 0 \\ 100 & -1 & -100 & 0 \\ 0 & 10 & 0 & -10 \\ 0 & 0 & 100 & -1 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ I_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} X$$

In MATLAB:

A =

$$\begin{bmatrix} 0 & -10 & 0 & 0 \\ 100 & -1 & -100 & 0 \\ 0 & 10 & 0 & -10 \\ 0 & 0 & 100 & -1 \end{bmatrix}$$

```
>> B = [10;0;0;0];  
>> C = [0,0,0,1];  
>> D = 0;  
>> G = ss(A,B,C,D);
```

The transfer function is from

```
>> tf(G)
```

$$\frac{1e006}{s^4 + 2 s^3 + 3001 s^2 + 3000 s + 1e006}$$

or if you prefer factored form

```
zpk(G)
```

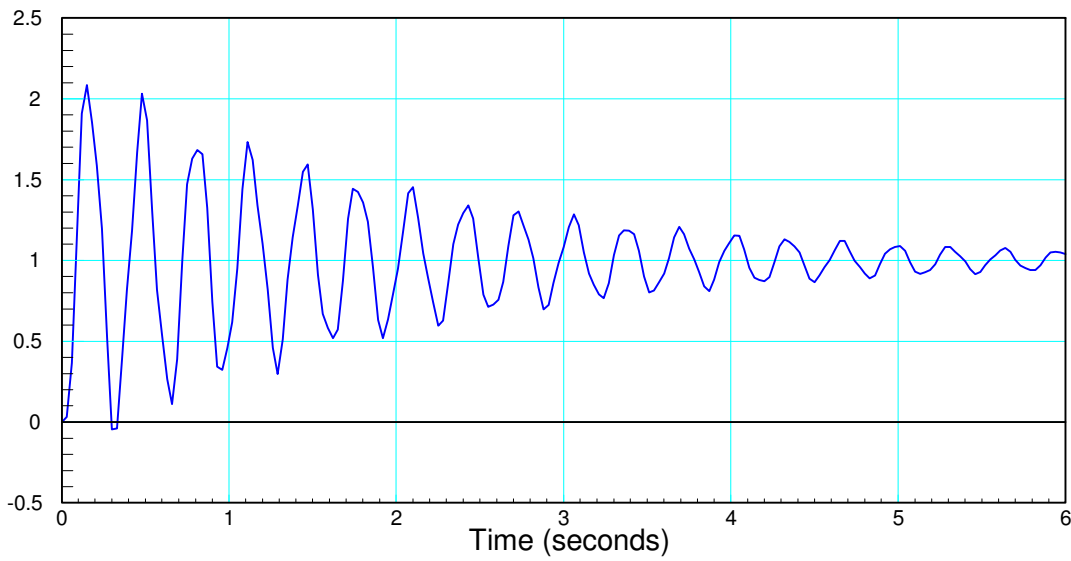
$$\frac{1000000}{(s^2 + s + 382) (s^2 + s + 2618)}$$

The poles can be found as

```
>> eig(G)  
  
-0.5000 +51.1643i  
-0.5000 -51.1643i  
-0.5000 +19.5376i  
-0.5000 -19.5376i
```

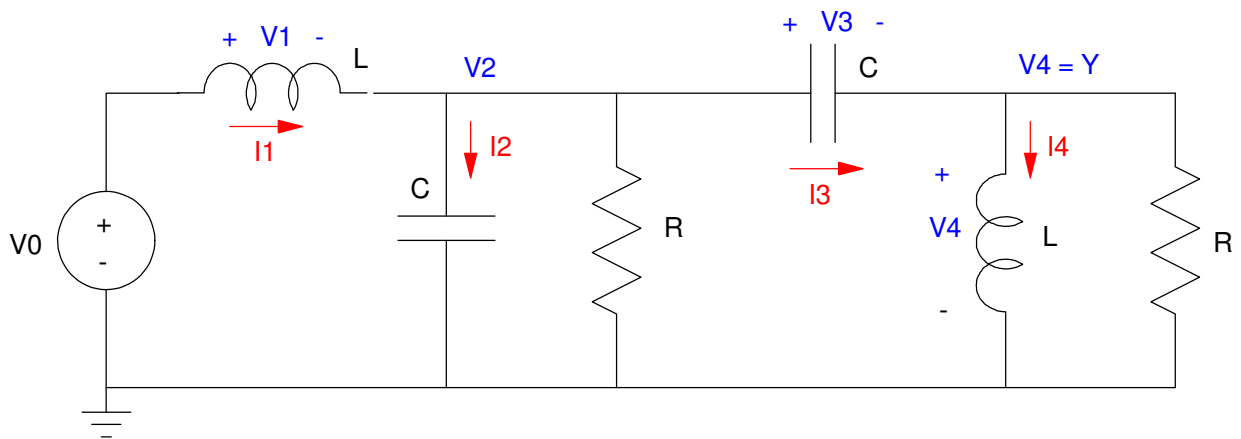
The step response can be found from:

```
-->t = [0:0.01:10]';  
-->y = step(G,t);  
-->plot(t,y);  
-->xlabel('Time (seconds)');  
-->ylabel('y (Volts)');
```



Step response for the RLC circuit

Example 2:



$$L = 100\text{mH}, C = 10\text{mF}, R = 50$$

Again, let the states define the energy in the system (I_1, V_2, V_3, I_4)

Write four coupled differential equations:

$$V_1 = LsI_1 = V_0 - V_2$$

$$I_2 = CsV_2 = I_1 - \frac{V_2}{R} - I_4 - \frac{V_2 - V_3}{R}$$

$$I_3 = CsV_3 = I_4 + \frac{V_2 - V_3}{R}$$

$$V_4 = LsI_4 = V_2 - V_3$$

Solve for the highest derivative

$$sI_1 = \frac{V_{in} - V_2}{L_1}$$

$$sV_2 = \frac{1}{C_2} \left(I_1 - \frac{V_2}{R} - I_4 - \frac{V_2 - V_3}{R} \right)$$

$$sV_3 = \frac{1}{C_3} \left(I_4 + \frac{V_2 - V_3}{R} \right)$$

$$sI_4 = \frac{V_2 - V_3}{L_4}$$

and place in matrix form

$$s \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} \left(\frac{-1}{L}\right) & 0 & 0 & 0 \\ \left(\frac{1}{C}\right) & \left(\frac{-2}{RC}\right) & \left(\frac{1}{RC}\right) & \left(\frac{-1}{C}\right) \\ 0 & \left(\frac{1}{RC}\right) & \left(\frac{-1}{RC}\right) & \left(\frac{1}{C}\right) \\ 0 & \left(\frac{1}{L}\right) & \left(\frac{-1}{L}\right) & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{L}\right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

$$y = V_2 - V_3$$

$$y = \begin{bmatrix} 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} + [0]V_{in}$$

Substituting $L = 100\text{mH}$, $C = 10\text{mF}$, $R = 50$

$$\begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} = \begin{bmatrix} -10 & 0 & 0 & 0 \\ 100 & -4 & 2 & -100 \\ 0 & 2 & -2 & 100 \\ 0 & 10 & -10 & 0 \end{bmatrix} \begin{bmatrix} I_1 \\ V_2 \\ V_3 \\ I_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_{in}$$

Throwing it in MATLAB:

```
-->A = [-10,0,0,0;100,-4,2,-100;0,2,-2,100;0,10,-10,0];  
-->B = [10;0;0;0];  
-->C = [0,1,-1,0];  
-->D = 0;  
  
-->G = ss(A,B,C,D);  
  
>> tf(G)
```

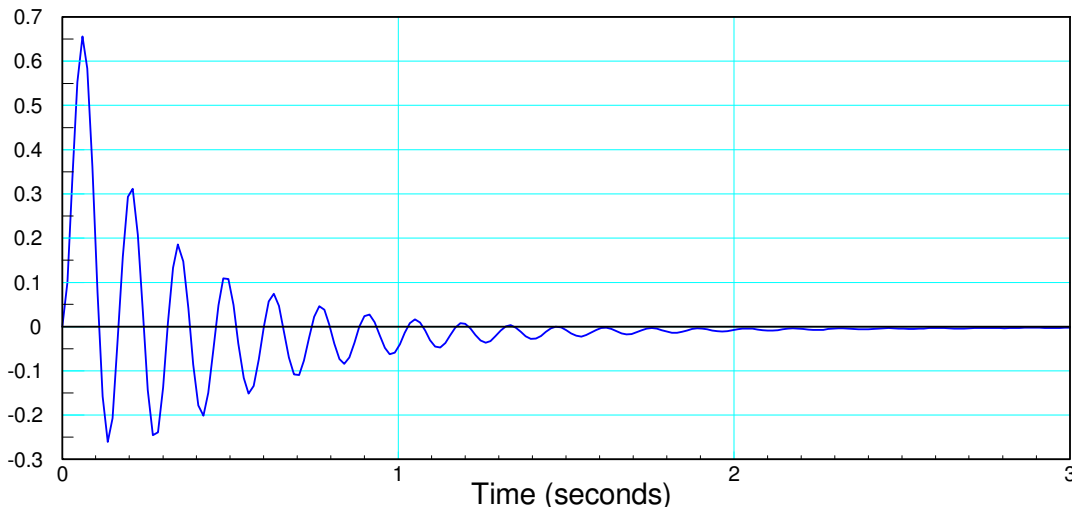
```
      1000 s^2 - 2.887e-012 s + 1.972e-027  
-----  
s^4 + 16 s^3 + 2064 s^2 + 2.204e004 s + 2e004
```

or if you prefer factored form:

```
>> zpk(G)  
  
      1000 s^2  
-----  
(s+10) (s+1.001) (s^2 + 4.999s + 1999)  
  
>> eig(G)  
  
ans =  
  
-2.4997 +44.6402i  
-2.4997 -44.6402i  
-1.0005  
-10.0000
```

The step response is from:

```
-->t = [0:0.001:3]';  
-->y = step(G,t);  
-->plot(t,y);  
-->xlabel('Time (seconds)');  
-->ylabel('y (Volts)');
```



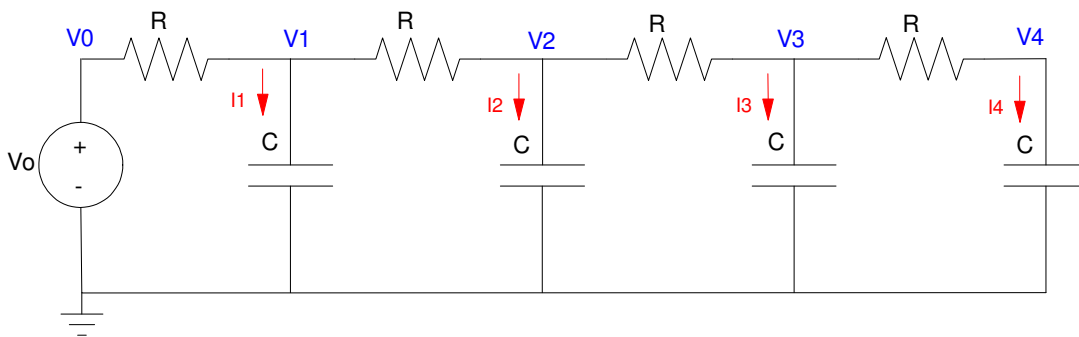
Step response for Example #2

RC Circuits: Heat Equation

In one dimension, the heat equation is defined as coupled first-order differential equations where

$$\frac{dx_i}{dt} = f(x_{i-1}, x_i, x_{i+1})$$

This also describes RC circuits where the capacitors are grounded such as



4-Stage RC Filter (heat equation)

The analogy with heat flow is

- Voltage Temperature
- Current Heat Flow
- Resistance Thermal Resistance
- Electrical Energy: Thermal Energy

Heat	Thermal Inertia (J / degree)	Temperature (degrees C)	Heat Flow (Watts)	Thermal Resistance (degree C / Watt)
Electrical	Capacitance (J / volt)	Voltage (Volts)	Current (Amps)	Resistance (R) (Ohm = Volts / Amp)

The coupled differential equations for such a circuit are

$$I_1 = CsV_1 = \left(\frac{V_0 - V_1}{R} \right) + \left(\frac{V_2 - V_1}{R} \right)$$

$$I_2 = CsV_2 = \left(\frac{V_1 - V_2}{R} \right) + \left(\frac{V_3 - V_2}{R} \right)$$

$$I_3 = CsV_3 = \left(\frac{V_2 - V_3}{R} \right) + \left(\frac{V_4 - V_3}{R} \right)$$

$$I_4 = CsV_4 = \left(\frac{V_3 - V_4}{R} \right)$$

Grouping terms and placing in matrix form

$$\begin{bmatrix} sV_1 \\ sV_2 \\ sV_3 \\ sV_4 \end{bmatrix} = \begin{bmatrix} \left(\frac{-2}{RC} \right) & \left(\frac{1}{RC} \right) & 0 & 0 \\ \left(\frac{1}{RC} \right) & \left(\frac{-2}{RC} \right) & \left(\frac{1}{RC} \right) & 0 \\ 0 & \left(\frac{1}{RC} \right) & \left(\frac{-2}{RC} \right) & \left(\frac{1}{RC} \right) \\ 0 & 0 & \left(\frac{1}{RC} \right) & \left(\frac{-1}{RC} \right) \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} \left(\frac{1}{RC} \right) \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

If $C = 0.01\text{F}$ and $R = 10\text{ Ohms}$,

$$\begin{bmatrix} sV_1 \\ sV_2 \\ sV_3 \\ sV_4 \end{bmatrix} = \begin{bmatrix} -20 & 10 & 0 & 0 \\ 10 & -20 & 10 & 0 \\ 0 & 10 & -20 & 10 \\ 0 & 0 & 10 & -10 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \\ 0 \end{bmatrix} V_0$$

Note that this is a lossless system: there is no thermal conductivity between the rod and the environment. This shows up in the above matrix with each row adding to zero. If there were losses, the diagonal elements would be more negative and the rows would sum to a negative number (lossy).

Problem: Estimate how this system will behave:

Solution: You need to know the systems dominant pole(s) and DC gain. In MATLAB:

```
A = [-20,10,0,0 ; 10,-20,10,0 ; 0,10,-20,10 ; 0,0,10,-10];
B = [10;0;0;0];
C = [0,0,0,1];
D = 0;
G = ss(A,B,C,D);
```

```
DC = evalfr(G,0)
DC = 1
```

```
zpk(G)
```

```
-----
10000
(s+35.32) (s+23.47) (s+10) (s+1.206)
```

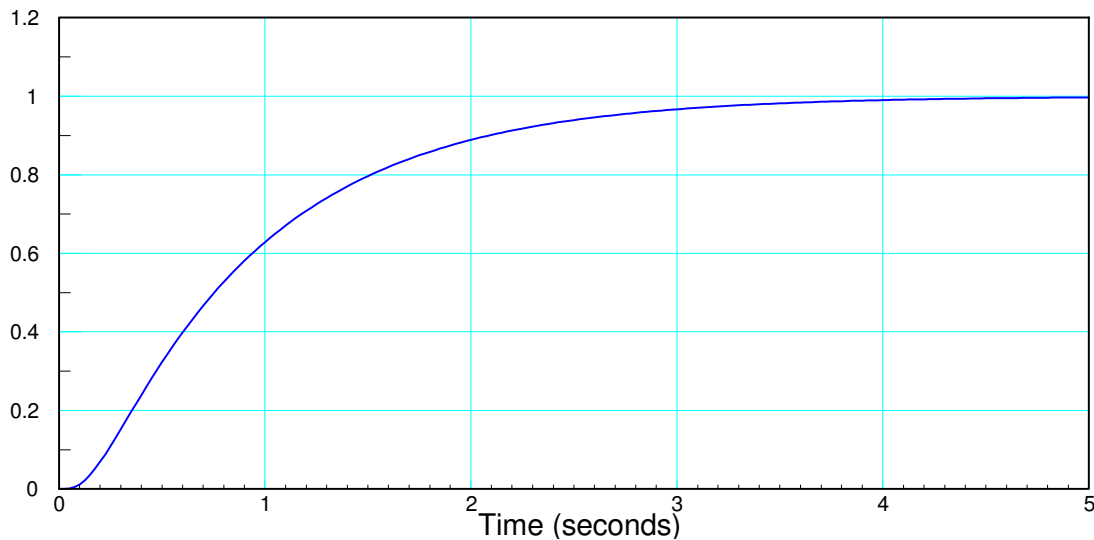
- The system has a DC gain of one. If the input is increased to 100C, the output will go to 100C as well.
- The system has a dominant pole at -1.206. It will take approximately 3.13 seconds ($4/1.206$) to reach steady state.
- The dominant pole is real. There should be no overshoot or oscillations in the step response.

Problem: Determine the actual step response:

Solution: Using MATLAB:

```
t = [0:0.025:5]';
y = step(G,t);
plot(t,y)

xlabel('Time (seconds)');
ylabel('Y (Celsius)');
```



Animation and Heat Equations (fun stuff):

With Matlab, you can show the metal bar heating up and cooling down. For example, simulate a 10-node RC filter with $1/RC = 10$ and the base temperature of 100C ($V_0 = 100$)

```
% 10-stage RC Filter

V = zeros(10,1);
dV = zeros(10,1);
V0 = 1;
dt = 0.01;
t = 0;

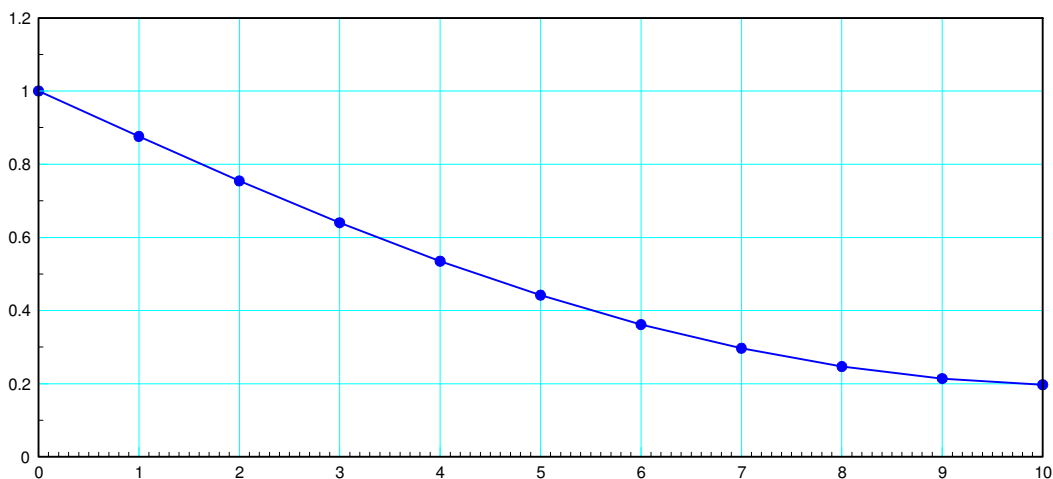
while(t < 100)

    dV(1) = 10*V0 - 20*V(1) + 10*V(2);
    dV(2) = 10*V(1) - 20*V(2) + 10*V(3);
    dV(3) = 10*V(2) - 20*V(3) + 10*V(4);
    dV(4) = 10*V(3) - 20*V(4) + 10*V(5);
    dV(5) = 10*V(4) - 20*V(5) + 10*V(6);
    dV(6) = 10*V(5) - 20*V(6) + 10*V(7);
    dV(7) = 10*V(6) - 20*V(7) + 10*V(8);
    dV(8) = 10*V(7) - 20*V(8) + 10*V(9);
    dV(9) = 10*V(8) - 20*V(9) + 10*V(10);
    dV(10) = 10*V(9) - 10*V(10);

    V = V + dV*dt;
    t = t + dt;

    plot([0:10], [V0;V], '.-');
    ylim([0,1.2]);
    pause(0.01);
end
```

This results in the following plot at $t = 2.0$ seconds



Step response at $t = 2.00$ seconds (voltage at each node)

Eigenvectors and Eigenvalues:

In state-space form, the dynamics are:

$$sX = AX + BU$$

$$Y = CX + DU$$

where A is

```
A = zeros(10,10);
for i=1:9
A(i,i) = -20;
A(i+1,i) = 10;
A(i,i+1) = 10;
end
A(10,10) = -10
```

```
-20    10    0    0    0    0    0    0    0    0
  10   -20   10    0    0    0    0    0    0    0
  0    10  -20   10    0    0    0    0    0    0
  0    0    10  -20   10    0    0    0    0    0
  0    0    0    10  -20   10    0    0    0    0
  0    0    0    0    10  -20   10    0    0    0
  0    0    0    0    0    10  -20   10    0    0
  0    0    0    0    0    0    10  -20   10    0
  0    0    0    0    0    0    0    10  -20   10
  0    0    0    0    0    0    0    0    10  -10
```

```
B = zeros(10,1);
B(1) = 10
```

```
10
 0
 0
 0
 0
 0
 0
 0
 0
 0
 0
```

```
C = zeros(1,10);
C(10) = 1;
D = 0;
G = ss(A,B,C,D);
```

```
zpk(G)
```

```
10000000000
```

```
-----
(s+39.11) (s+36.52) (s+32.47) (s+27.31) (s+21.49) (s+15.55) (s+10) (s+5.339) (s+1.981) (s+0.2234)
```

This system has

- A total of 10 poles, with
- A dominant pole at $s = -0.2234$ and
- A fast pole at $s = -39.11$

The eigenvectors and eigenvalues of A are:

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

m = Eigenvectors

-0.1286	-0.2459	0.3412	0.4063	0.4352	0.4255	0.3780	0.2969	-0.1894	0.0650
0.2459	0.4063	-0.4255	-0.2969	-0.0650	0.1894	0.3780	0.4352	-0.3412	0.1286
-0.3412	-0.4255	0.1894	-0.1894	-0.4255	-0.3412	-0.0000	0.3412	-0.4255	0.1894
0.4063	0.2969	0.1894	0.4352	0.1286	-0.3412	-0.3780	0.0650	-0.4255	0.2459
-0.4352	-0.0650	-0.4255	-0.1286	0.4063	0.1894	-0.3780	-0.2459	-0.3412	0.2969
0.4255	-0.1894	0.3412	-0.3412	-0.1894	0.4255	0.0000	-0.4255	-0.1894	0.3412
-0.3780	0.3780	0.0000	0.3780	-0.3780	-0.0000	0.3780	-0.3780	-0.0000	0.3780
0.2969	-0.4352	-0.3412	0.0650	0.2459	-0.4255	0.3780	-0.1286	0.1894	0.4063
-0.1894	0.3412	0.4255	-0.4255	0.3412	-0.1894	0.0000	0.1894	0.3412	0.4255
0.0650	-0.1286	-0.1894	0.2459	-0.2969	0.3412	-0.3780	0.4063	0.4255	0.4352

v = Eigenvalues

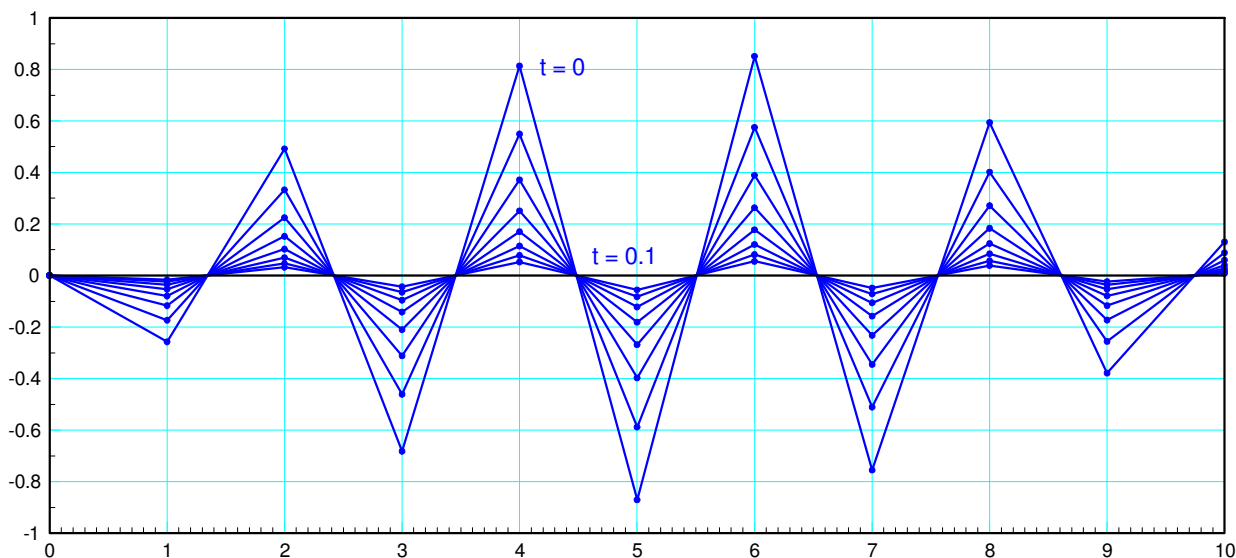
-39.1115	-36.5248	-32.4698	-27.3068	-21.4946	-15.5496	-10.0000	-5.3390	-1.9806	-0.2234
----------	----------	----------	----------	----------	----------	----------	---------	---------	---------

Note that the eigenvalues are the same as the poles. Eigenvalues are important: they tell you *how* the system behaves.

The eigenvectors tell you *what* behaves that way. For example, if you make the initial conditions equal to the fast eigenvector (show in red)

```
V = M(:,1) * 2;
V0 = 0;
dV = zeros(10,1);
```

the voltages look like this



Natural response with initial conidions equal to the fast eigenvector

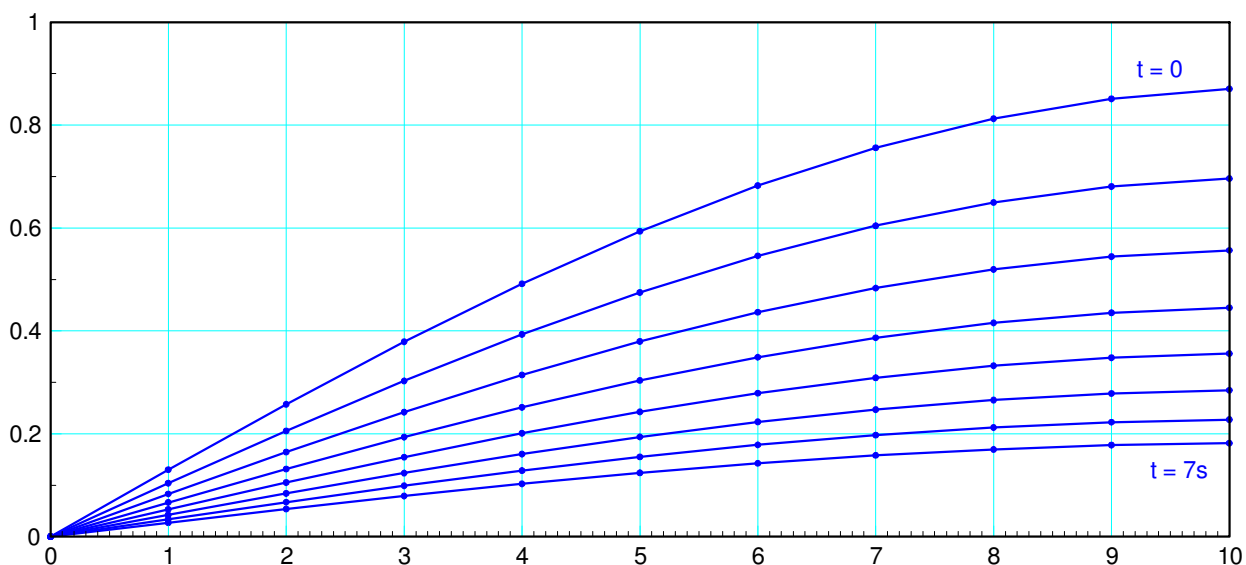
Note that

- The shape stays the same: this is the eigenvector
- The amplitude drops off quickly as $\exp(-39.11t)$: this is the eigenvalue.

If you make the initial condition the slow eigenvector:

```
V = M(:,10) * 2;
V0 = 0;
dV = zeros(10,1);
etc.
```

the response is



Natural Response with the initial condition being the slow eigenvector

Note that

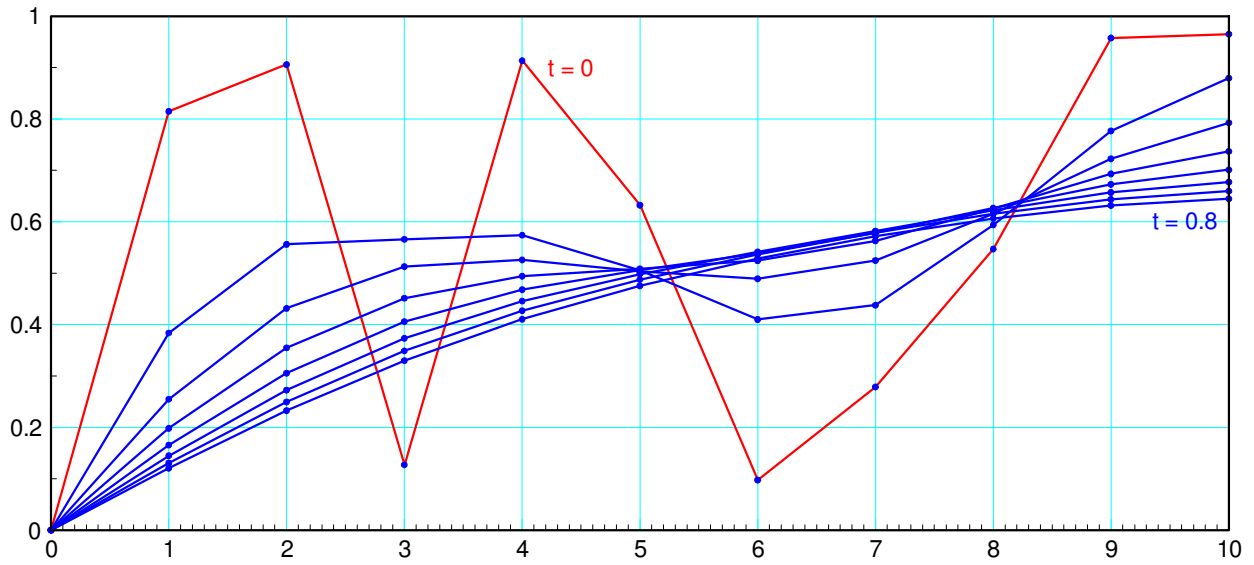
- The shape stays the same: this is the eigenvector
- The amplitude drops off slowly as $\exp(-0.2234t)$: this is the eigenvalue.

Finally, if you make the initial condition something random:

```
V = rand(10,1);
V0 = 0;
```

then all 10 eigenvectors are excited. This results in

- The fast modes decaying quickly,
- Leaving the slow (dominant) mode



Natural Response with Random Initial Conditions