Convolution

Dr. Mitch Pryor

Solving linear systems

Recall, the solution to $\dot{x} = ax$ is ce^{at}

A Taylor series of the exponential function of x is

$$e^{x} = 1 + x + \frac{x^{2}}{2!} + \frac{x^{3}}{3!} + \frac{x^{4}}{4!} + \dots$$

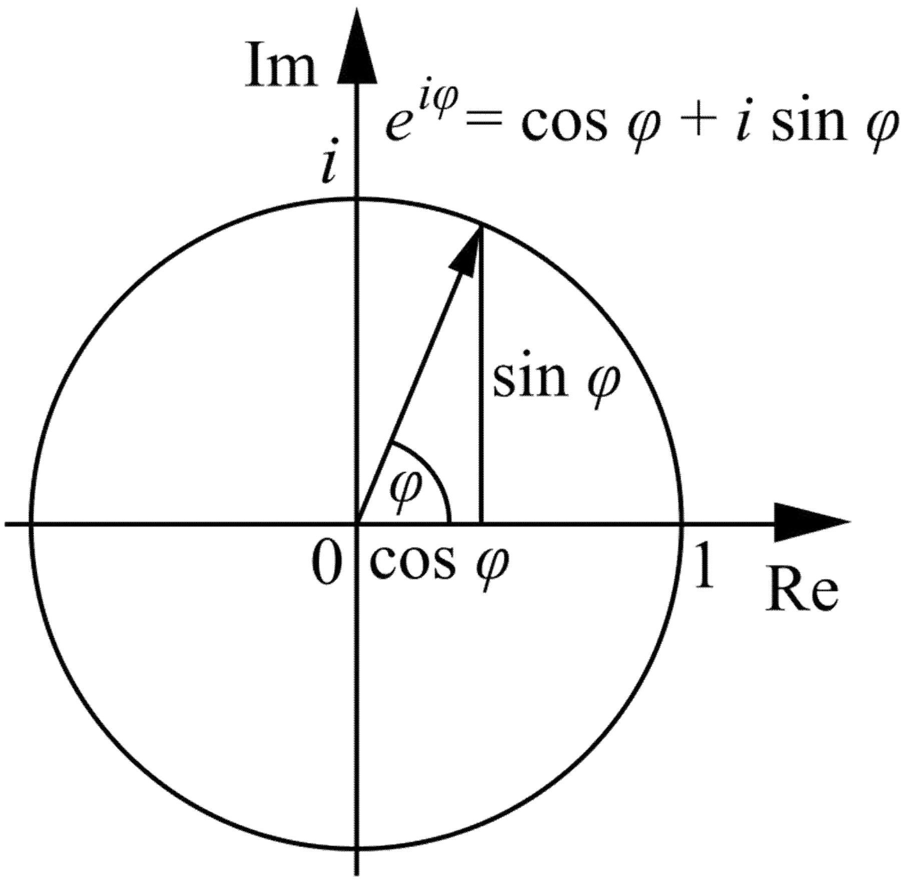
By analogy, we defined the Matrix Exponential

$$e^{\mathbf{A}} = 1 + \mathbf{A} + \frac{\mathbf{A}^{2}}{2!} + \frac{\mathbf{A}^{3}}{3!} + \frac{\mathbf{A}^{4}}{4!} + \dots$$

In MATLAB, the command expm(A) computes $e^{\mathbf{A}}$

So the solution to $$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u \Rightarrow \mathbf{z} = \underbrace{\mathbf{z}_{0}e^{\mathbf{A}t}}_{\textit{Homogeneous solution}} + \underbrace{???}_{\textit{Our goal today}}$$

e^(complex #) ➔ Oscillation



Euler's Formula

Recall for linear systems…

- Today (and for most of the class) we will focus on input / single output linear systems
  - Why can we do this?

- A **linear** system f(x) must obey these rules

$$f(x+y) = f(x) + f(y)$$
$$f(ax) = af(x)$$
$$f(ax+by) = af(x) + bf(y)$$

"Superposition"

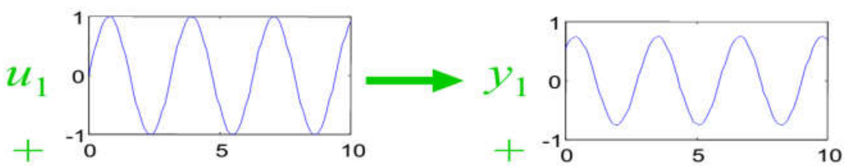
- So why is this important for controls?

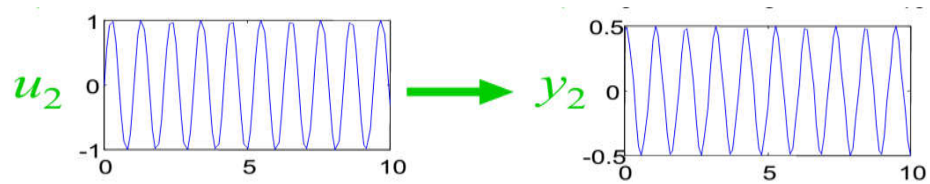
Linear control systems

If we have two known responses from two different inputs to our system…

$$\frac{d}{dt}\mathbf{z} = \mathbf{A}\mathbf{z} + \mathbf{B}u_1 \qquad\qquad \frac{d}{dt}\mathbf{z} = \mathbf{A}\mathbf{z} + \mathbf{B}u_2$$

$$y_1 = \mathbf{C}\mathbf{z} + \mathbf{D}u_1 \qquad\qquad y_2 = \mathbf{C}\mathbf{z} + \mathbf{D}u_2$$



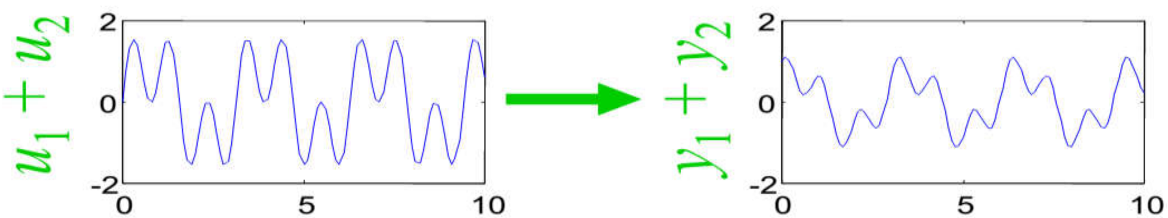


then we also can find the response to the sum of the input:

$$u_3 = \alpha u_1 + \beta u_2$$

$$\frac{d}{dt}\mathbf{z} = \mathbf{A}\mathbf{z} + \mathbf{B}u_3$$

$$y_3 = \mathbf{C}\mathbf{z} + \mathbf{D}u_3 = \alpha\, y_1 + \beta\, y_2$$



Common inputs: Impulse and Step

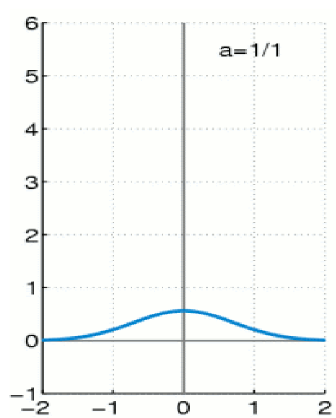
The Impulse Function

$$u(t) = u_0 \begin{Bmatrix} \infty, & t = t_0 \\ 0, & t \neq t_0 \end{Bmatrix}$$

Or more accurately…

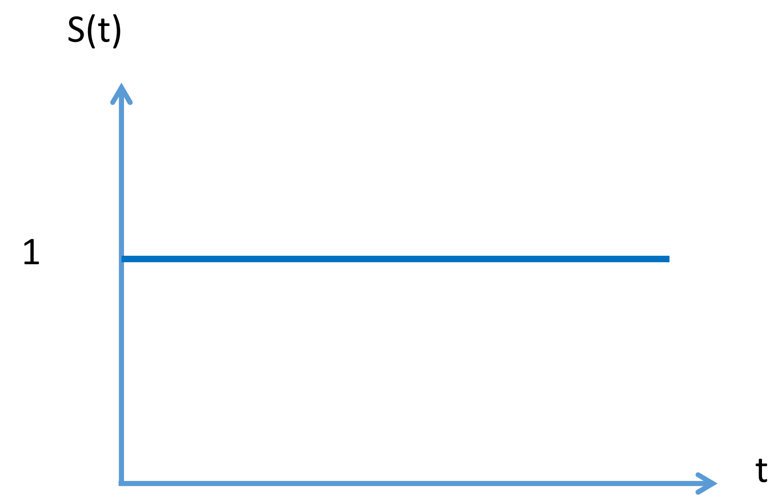
$$p_{\varepsilon}(t) = u_0 p_{\varepsilon}(t) = u_0 \begin{cases} 0 & t < 0 \\ \dfrac{1}{\varepsilon} & 0 \leq \mathrm{t} < \varepsilon \\ 0 & \mathrm{t} \geq \varepsilon \end{cases}$$

$$\delta(t) = \lim_{\varepsilon \to 0} p_{\varepsilon}(t)$$



The unit step function

$$s(t) = \begin{cases} 0 & t < t_0 \\ 1 & t \geq t_0 \end{cases}$$



Consider multiple step inputs

$$\underbrace{\underbrace{y(t, y_o, \delta u_1 + \cdots + \delta u_n)}_{\textit{Thus, this is the solution to}} = y(t, y_o, \delta u_1) + \cdots + y(t, y_o, \delta u_n)}_{\textit{Principle of superposition}}$$

Principle of superposition

Thus, this is the solution to

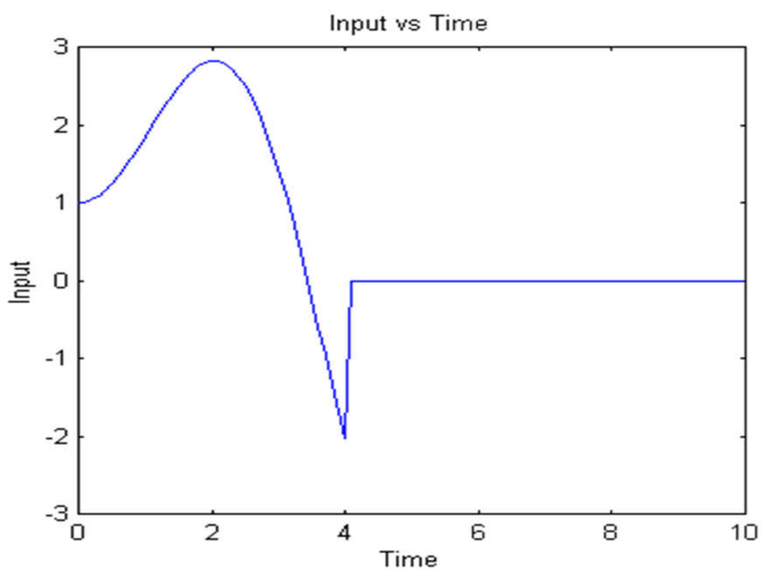
$$\frac{dx}{dt} = \mathbf{A}x + \mathbf{B}u$$
$$y = \mathbf{C}x + \mathbf{D}u$$

for any set of inputs

Let's assume that y_o=0 (i.e. we are only interested in the particular (not homogeneous) solution.

<u>Any input can be viewed as a combination of other inputs!</u>

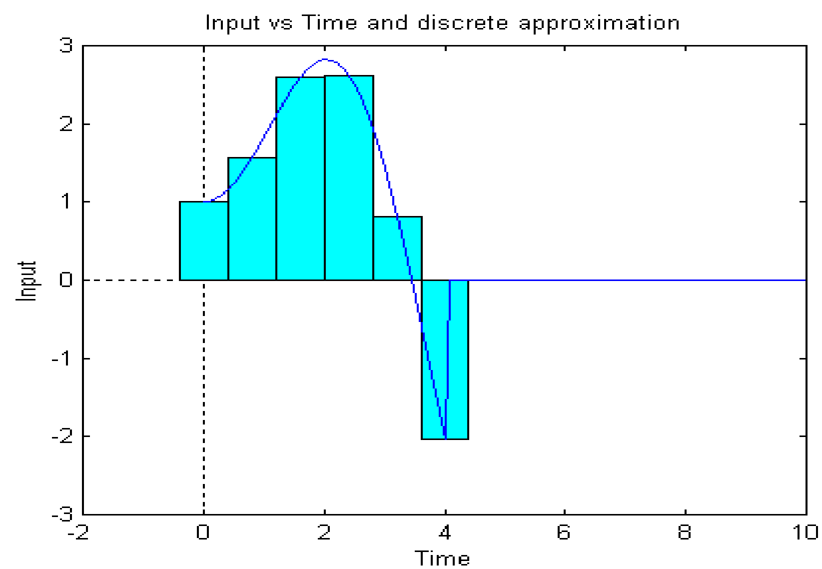
Example: “Complex” Input



Consider this rather complicated input u(t) to a linear system.

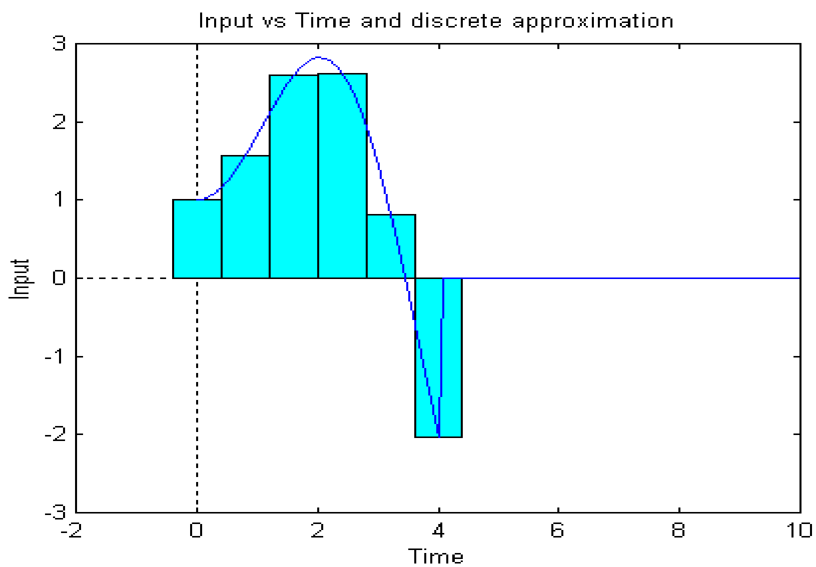
$$\ddot{z} + \dot{z} + z = u(t)$$

Objective: Find z(t) if the initial conditions are zero.

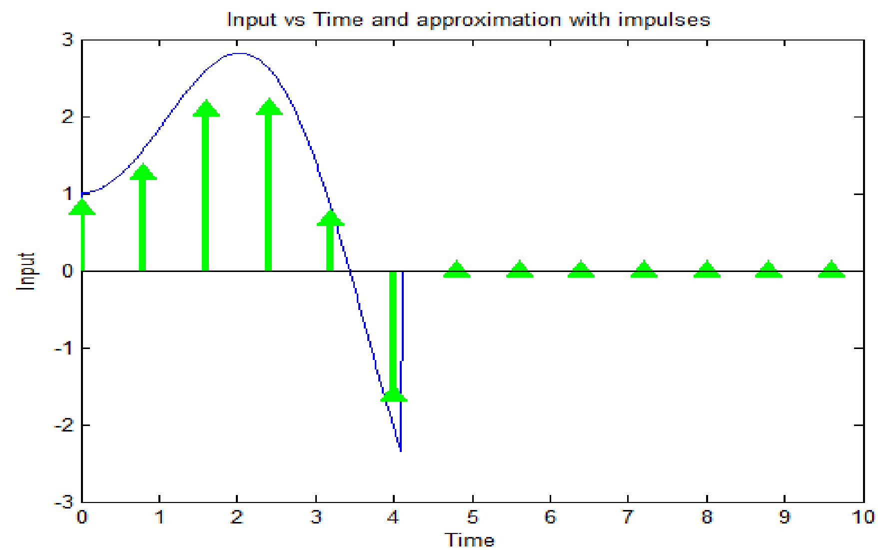


We can estimate this function as a set of discrete functions (here each is 0.8 seconds)

Example: “Complex” Input

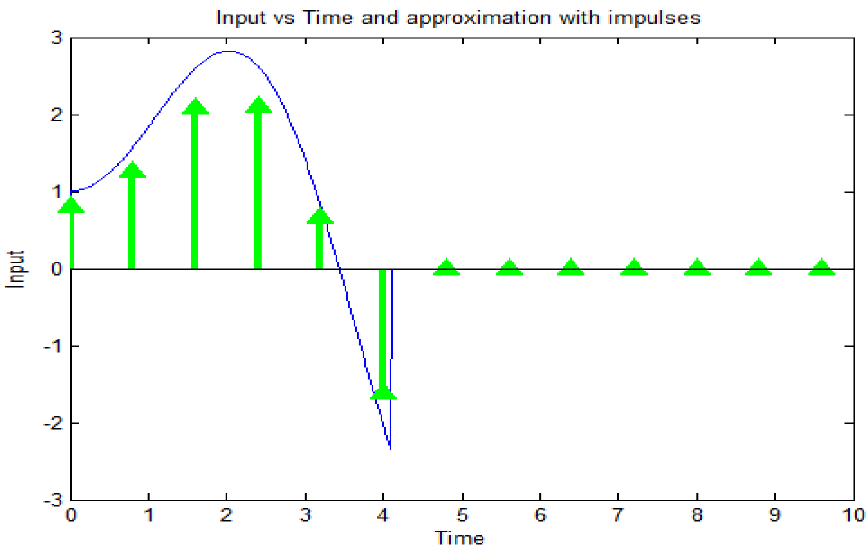


We can estimate this function as a set of discrete functions (here each is 0.8 seconds)

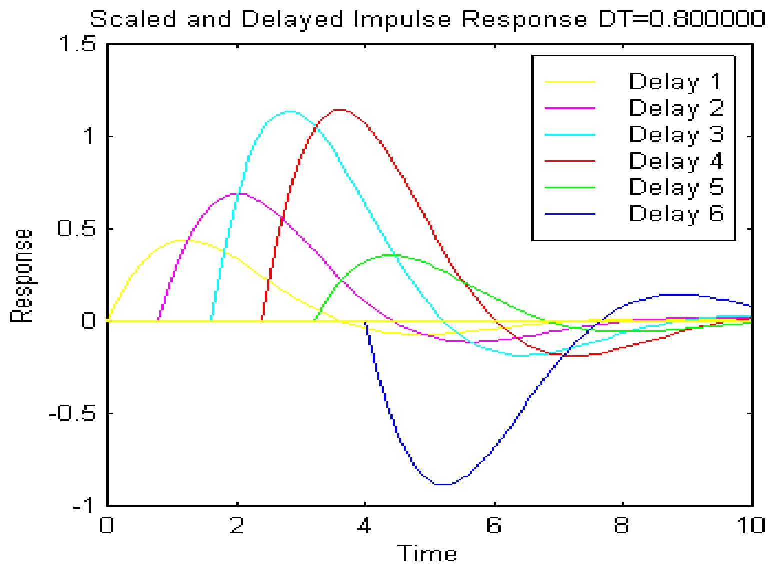


We take this one step further and replace each step with an equivalent impulse (they are shorter than the function since Δt=0.8s)

Example: “Complex” Input

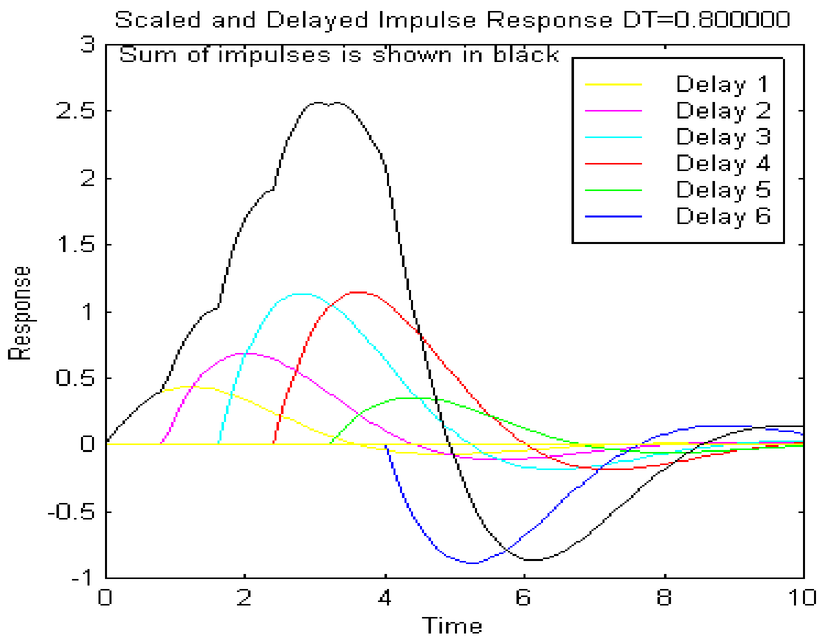


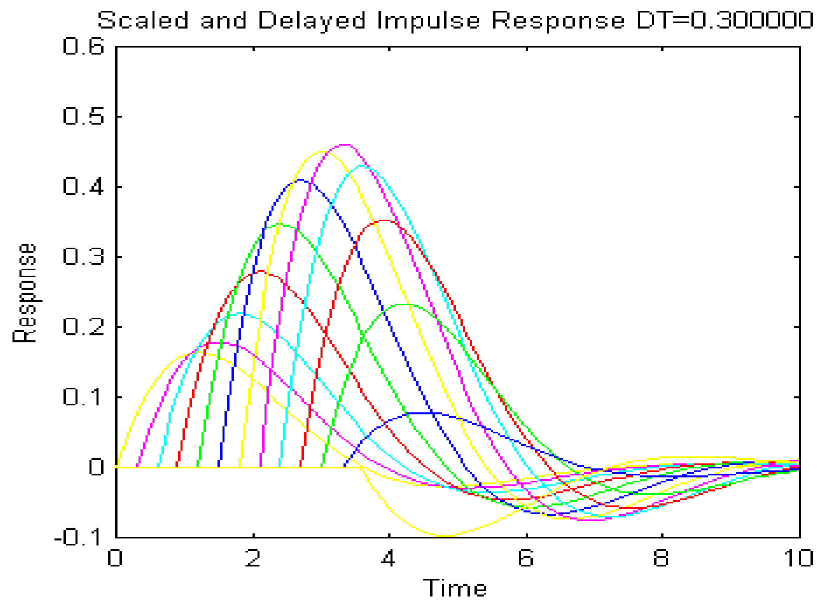
We take this one step further and replace each step with an equivalent impulse (they are shorter than the function since Δt=0.8s)



Here is the system response to each of the impulse. The final response will be sum of the impulse responses.

Example: “Complex” Input

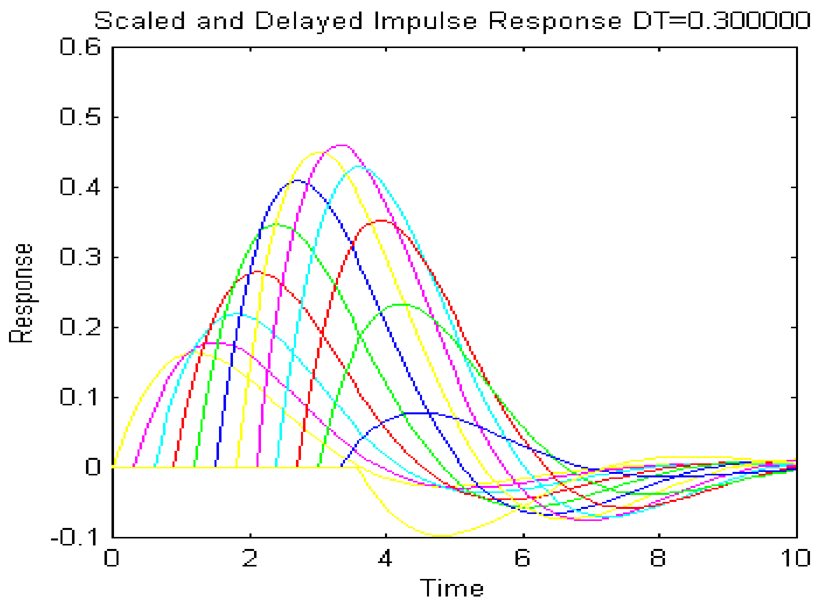




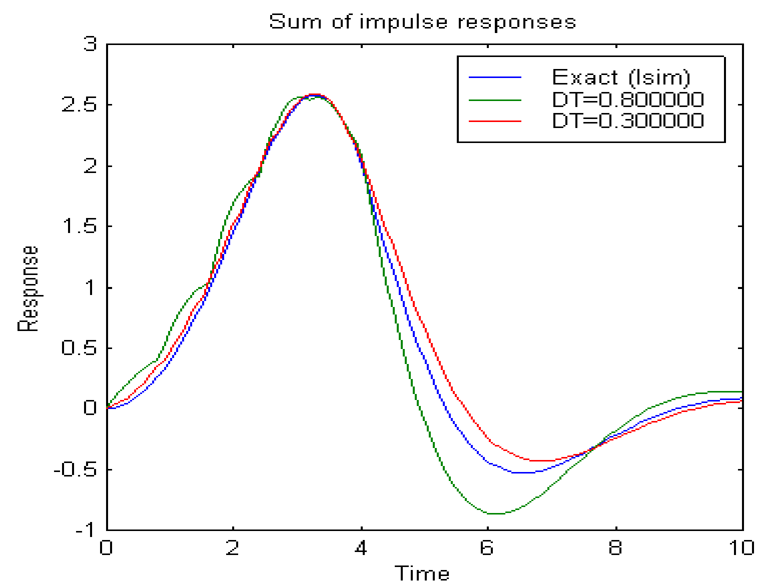
Here is the system response to each of the impulse. The final response will be sum of the impulse responses. The accuracy will improve by reducing Δt.

For example, here is the response if Δt=0.3.

Example: “Complex” Input



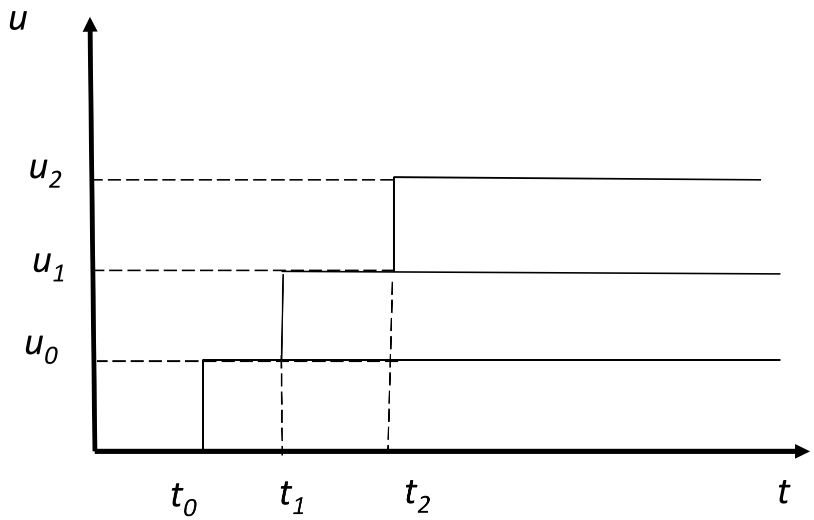
For example, here is the response if Δt=0.3.



If we sum up these impulse responses and compare them to the exact solution….

And what if we allow Δt to go to 0?

Output from a set of step inputs



Let H(t) be the response to a step input at t_j.

$$\Rightarrow \quad y(t) = H(t - t_j)u(t_j)$$

Can steps be negative? *Sure.*

Sum all of the inputs…

$$y(t) = H(t - t_0)u(t_0) + H(t - t_1)(u(t_1) - u(t_0)) + H(t - t_2)(u(t_2) - u(t_1)) + \cdots$$

$$= \left[H(t - t_0) - H(t - t_1)\right]u(t_0) + \left[H(t - t_1) - H(t - t_2)\right]u(t_1) + \cdots$$

Organize by common inputs…

$$= \sum_{i=1}^{t_n > t} \left[H(t - t_i) - H(t - t_{i+1})\right]u(t_i)$$

Rewrite as a summation of inputs…

$$= \sum_{i=1}^{t_n > t} \frac{\left[H(t - t_i) - H(t - t_{i+1})\right]}{t_{i+1} - t_i} u(t_i)\left(t_{i+1} - t_i\right)$$

Multiply numerator and denominator by delta T.

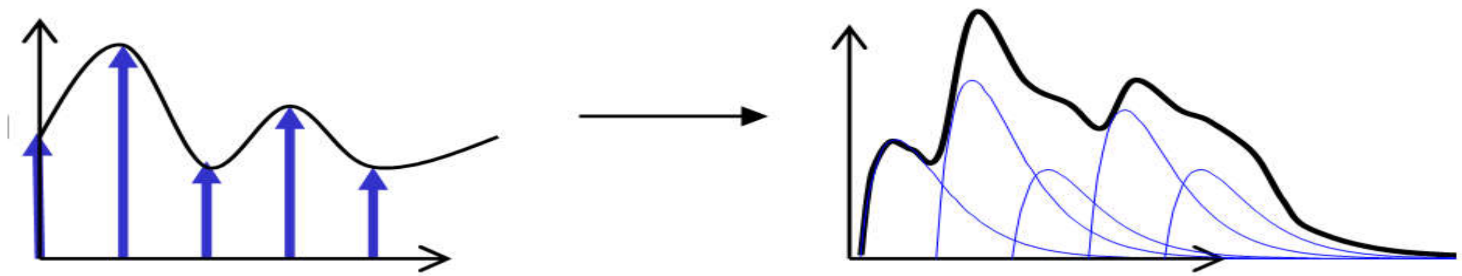
Output from a set of inputs

From previous page…

$$y(t) = \sum_{i=1}^{t_n > t} \frac{\left[H(t - t_i) - H(t - t_{i+1})\right]}{t_{i+1} - t_i} u(t_i)\left(t_{i+1} - t_i\right)$$

Let $\left(t_{i+1} - t_i\right) \to 0$

$$y(t) = \boxed{\int_0^t H'(t - \tau) u(\tau) d\tau}$$



We can use this to find the complete solution to:

$$\frac{dx}{dt} = \mathbf{A}x + \mathbf{B}u$$
$$y = \mathbf{C}x + \mathbf{D}u$$

Complete Solution

Given:

$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\,\mathbf{z} + \mathbf{B}\,u$$
$$\mathbf{y} = \mathbf{C}\,\mathbf{z} + \mathbf{D}\,u$$

We know:

$$\mathbf{z}(t) = \underbrace{e^{\mathbf{A}t}\mathbf{z}(0)}_{\text{Homogeneous sol}^{n}\text{ found with matrix exponential.}} + \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Response to system input}}$$

But what we are really interested in is the output

$$y = \mathbf{C}\,\mathbf{z} + \mathbf{D}\,u$$

Which must include….

$$\mathbf{y}(t) = \mathbf{C}\,e^{\mathbf{A}t}\mathbf{z}(0) + \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Response to system input}}$$

Complete Solution

$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$y = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

For a series of impulse inputs

$$y(t) = \int_0^t H'(t-\tau)\delta(\tau)\,d\tau$$
$$= \int_0^t C e^{A(t-\tau)} B u(\tau)\,d\tau$$

Put it all together, and we get…

$$\mathbf{y}(t) = \underbrace{\underbrace{\mathbf{C} e^{\mathbf{A}t}\mathbf{z}(0)}_{\textit{Homogeneous response}}}_{\textit{Init. condition response}} + \underbrace{\underbrace{\int_0^t \mathbf{C} e^{\mathbf{A}(t-\tau)}\mathbf{B} u(\tau)\,d\tau}_{\textit{System response to input(s)}} + \underbrace{\mathbf{D}u}_{\textit{Inclusion of input in measured signal}}}_{\underbrace{\textit{Transient response}}_{\textit{Mismatch between ICs and steady state}} \textit{ AND } \underbrace{\textit{steady state response}}_{\textit{Only periodic if input is periodic}}}$$

Does this look familiar?

It just might….

$$y(t) = c e^{at} z(0) + \int_0^t c e^{a(t-\tau)} u(\tau) d\tau + d u(t)$$

*is known as the **convolution equation**. We have just derived it using the Matrix Exponential for a system of 1st order differential equations.*

$$\mathbf{y}(t) = \mathbf{C} e^{\mathbf{A} t} \mathbf{z}(0) + \int_0^t \mathbf{C} e^{\mathbf{A}(t-\tau)} \mathbf{B} u(\tau) d\tau + \mathbf{D} u(t)$$

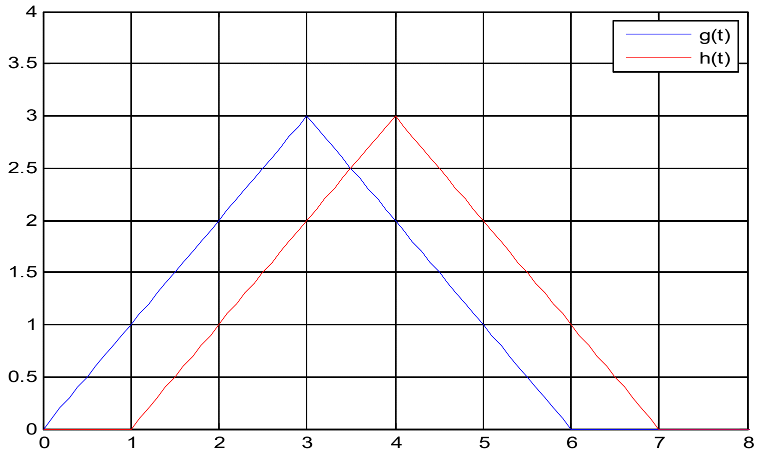
- What is convolution?
  - a twisting or folding together of two things
  - A summing (using integral instead of a addition)
- Many examples:
  - Digital signal processing
  - A sound that bounces off of a wall and interacts with the source sound is a convolution
  - A shadow is a convolution between the light source and the object producing the shadow
  - In statistics, a moving average is a convolution
  - Review convolution and its state-space examples in next lesson.

Convolution example

$$f(t) = g(t) \otimes h(t) = \int g(t) * h(t) dt$$

$$g(t) = \begin{cases} t & t < 3 \\ 6 - t & t \geq 3 \\ 0 & t > 6 \end{cases}$$

$$h(t) = \begin{cases} 0 & t < 1 \\ t - 1 & 1 \leq t \leq 4 \\ 7 - t & t > 4 \\ 0 & t > 7 \end{cases}$$




```
t = [0:.1:10];

for i=1:length(t)
if t(i)<3
    g(i)=t(i);
elseif t(i) <= 6
    g(i) = 6-t(i);
else
    g(i)=0;
end
end

for i=1:length(t)
if t(i)<1
    h(i)=0;
elseif t(i)<=4;
    h(i)=t(i)-1;
elseif t(i) <= 7
    h(i) = 7-t(i);
else
    h(i)=0;
end
end

plot(t,g, 'b')
hold on; grid on;
plot(t,h, 'r')
legend('g(t)','h(t)');
axis([0 8 0 4 ]);
```

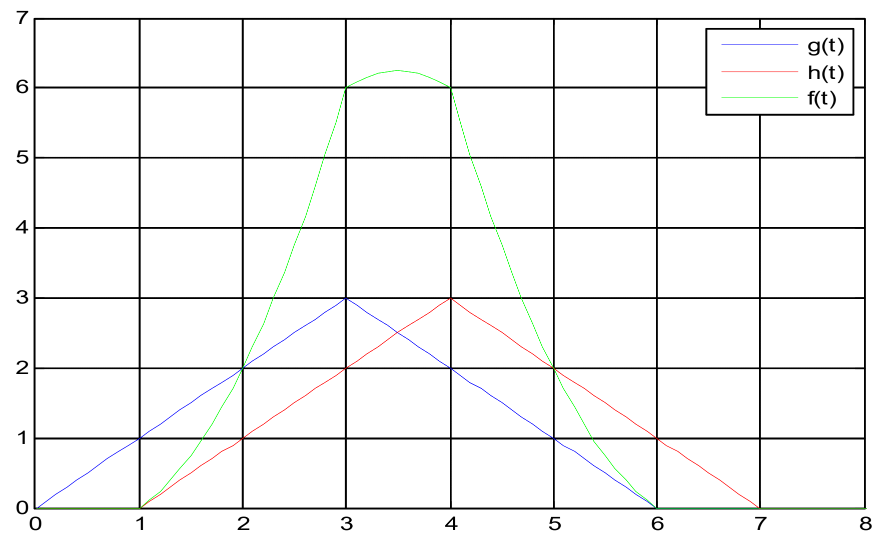
Example Result

$$f(t) = g(t) \otimes h(t) = \int_0^t g(t)h(t)\,dt$$

$$\text{Note: This is not } \int g(t)dt \times \int h(t)dt!$$

$$g(t) = \begin{cases} t & t < 3 \\ 6 - t & t \geq 3 \\ 0 & t > 6 \end{cases} \qquad h(t) = \begin{cases} 0 & t < 1 \\ t - 1 & 1 \leq t \leq 4 \\ 7 - t & t > 4 \\ 0 & t > 7 \end{cases}$$

$$g(t) * h(t) = \begin{cases} 0 & t < 1 \\ t(t-1) & 1 \leq t \leq 3 \\ (6-t)(t-1) & 3 < t \leq 4 \\ (6-t)(7-t) & 4 \leq t \leq 6 \\ 0 & t > 6 \end{cases}$$



$$\int_0^t g(t)h(t)dt = 0 + \left(\frac{t^3}{3} - \frac{t^2}{2}\right)\Bigg|_{t=1}^{t=3} + \left(3t^2 - \frac{t^3}{3}\right)\Bigg|_{t=3}^{t=4} + \left(42t - \frac{13t^2}{2} + \frac{t^3}{3}\right)\Bigg|_{t=4}^{t=6} + 0 = 15.5$$

Another derivation of convolution

Start with our model in state-space form

$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\,\mathbf{z}(t) + \mathbf{B}\,u(t)$$
$$\mathbf{y} = \mathbf{C}\,\mathbf{z}(t) + \mathbf{D}\,u(t)$$

The homogeneous solution is…

$$\mathbf{z}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{z}(t_0)$$

To find the particular solution, we let K(t) be an n x n matrix such that…

$$\frac{d\mathbf{K}(t)}{dt} = -\mathbf{K}(t)\mathbf{A}$$

So the matrix must be of the form….

$$\mathbf{K}(t) = e^{-\mathbf{A}(t-t_0)}$$
$$\mathbf{K}(t)\dot{\mathbf{z}} = \mathbf{K}(t)\mathbf{A}\,\mathbf{z}(t) + \mathbf{K}(t)\mathbf{B}\,u(t)$$
$$\mathbf{K}(t)\dot{\mathbf{z}}(t) - \mathbf{K}(t)\mathbf{A}\,\mathbf{z}(t) = \mathbf{K}(t)\mathbf{B}\,u(t)$$

$$\frac{d}{dt}\left[\mathbf{K}(t)\mathbf{z}(t)\right] = \dot{\mathbf{K}}\,\mathbf{z} + \mathbf{K}\,\dot{\mathbf{z}} = \mathbf{K}(t)\mathbf{B}\,u(t)$$
$$d\left[\mathbf{K}(t)\mathbf{z}(t)\right] = \mathbf{K}(t)\mathbf{B}\,u(t)\,dt$$

Integrating from t to t_0

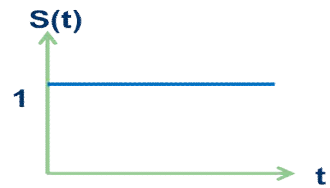
$$\mathbf{K}(t)\mathbf{z}(t) - \mathbf{K}(t_0)\mathbf{z}(t_0) = \int_{t_0}^{t}\mathbf{K}(\tau)\mathbf{B}\,u(\tau)\,d\tau$$
$$\mathbf{x}(t) = \mathbf{K}^{-1}(t)\mathbf{K}(t_0)\mathbf{z}(t_0) + \int_{t_0}^{t}\mathbf{K}^{-1}(t)\mathbf{K}(\tau)\mathbf{B}\,u(\tau)\,d\tau$$
$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{z}(t_0) + \int_{t_0}^{t}e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,d\tau$$
$$\mathbf{y}(t) = \mathbf{C}\,e^{\mathbf{A}(t-t_0)}\mathbf{z}(t_0) + \int_{t_0}^{t}\mathbf{C}\,e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,d\tau + \mathbf{D}\,u(t)$$

If t_0=0…..

$$\boxed{\mathbf{y}(t) = \mathbf{C}\,e^{\mathbf{A}(t)}\mathbf{z}(0) + \int_{0}^{t}\mathbf{C}\,e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)\,d\tau + \mathbf{D}\,u(t)}$$

Convolution w/ one step input

Recall our step input is….



$$S(t) = \begin{cases} 0 & t \leq 0 \\ 1 & t > 0 \end{cases}$$

Plug into our convolution equation….

$$\mathbf{y}(t) = \mathbf{C}\, e^{\mathbf{A}(t-t_0)}\mathbf{z}(t_0) + \int_{t_0}^{t} \mathbf{C}\, e^{\mathbf{A}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)$$

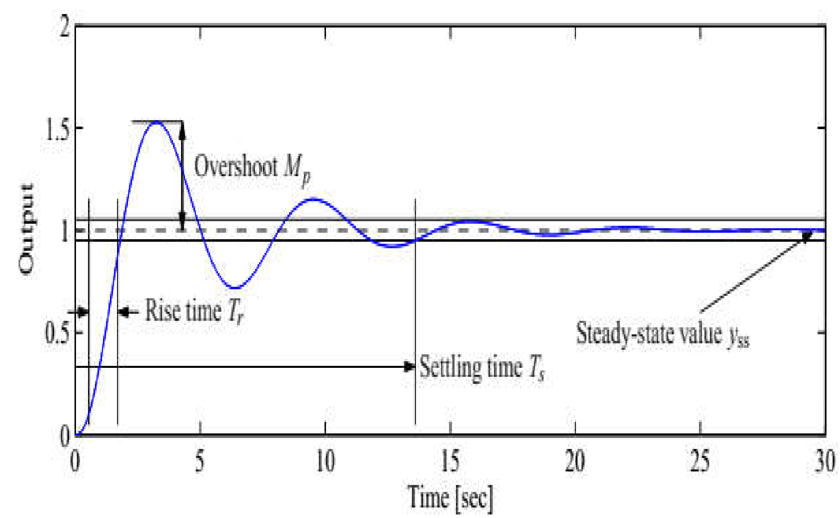
Note: order matters for matrix multiplication

Let the initial conditions be zero…

$$\begin{aligned} \mathbf{y}(t) &= \int_{0}^{t} \mathbf{C}\, e^{\mathbf{A}(t-\tau)}\mathbf{B}\, S(\tau)\, d\tau + \mathbf{D}\, S(t) \\ &= \mathbf{C} \int_{0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\, d\tau + \mathbf{D} \\ &= \mathbf{C} \int_{0}^{t} e^{\mathbf{A}\sigma}\mathbf{B}\, d\sigma + \mathbf{D} = \mathbf{C}\left(\mathbf{A}^{-1} e^{\mathbf{A}\sigma}\mathbf{B}\right)\Big|_{\sigma=0}^{\sigma=t} + \mathbf{D} \end{aligned}$$

$$A^{-1} \neq \frac{1}{A}$$

$$= \underbrace{-\mathbf{C}\mathbf{A}^{-1}\mathrm{e}^{\mathbf{A}t}\mathbf{B}}_{\text{transient response}} + \underbrace{\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{D}}_{\text{steady state terms}}$$



Convolution w/ sinusoidal input

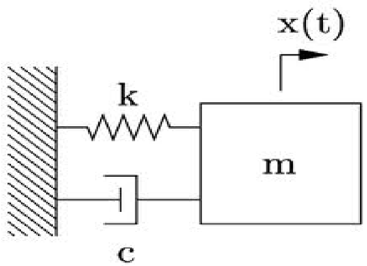
- Another common test function is a sinusoid for frequency response

$$u(t) = e^{st} \quad \text{where } s = \pm i\omega \qquad u(t) = \cos\omega t = \frac{e^{i\omega t} + e^{-i\omega t}}{2}$$

- Since we have a linear system, and assuming that the eigenvalues of A do not equal s

$$\begin{aligned}
y(t) &= \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}e^{s\tau}d\tau + \mathbf{D}e^{st} \\
&= \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \mathbf{C}e^{\mathbf{A}t}\int_0^t e^{(s\mathbf{I}-\mathbf{A})\tau}\mathbf{B}e^{s\tau}d\tau + \mathbf{D}e^{st} \\
&= \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \mathbf{C}e^{\mathbf{A}t}(s\mathbf{I}-\mathbf{A})^{-1}\left[e^{(s\mathbf{I}-\mathbf{A})t} - \mathbf{I}\right]\mathbf{B} + \mathbf{D}e^{st} \\
&= \underbrace{\mathbf{C}e^{\mathbf{A}t}\left(\mathbf{z}(0) - (s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}\right)}_{\text{Transient}} + \underbrace{\left(\mathbf{C}\left(s\mathbf{I}-\mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D}\right)e^{st}}_{\text{Steady State}}
\end{aligned}$$

Example: Mass – spring -damper



$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

$$\frac{d\mathbf{z}}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{k}{m} \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ \frac{F}{m} \end{bmatrix}u$$

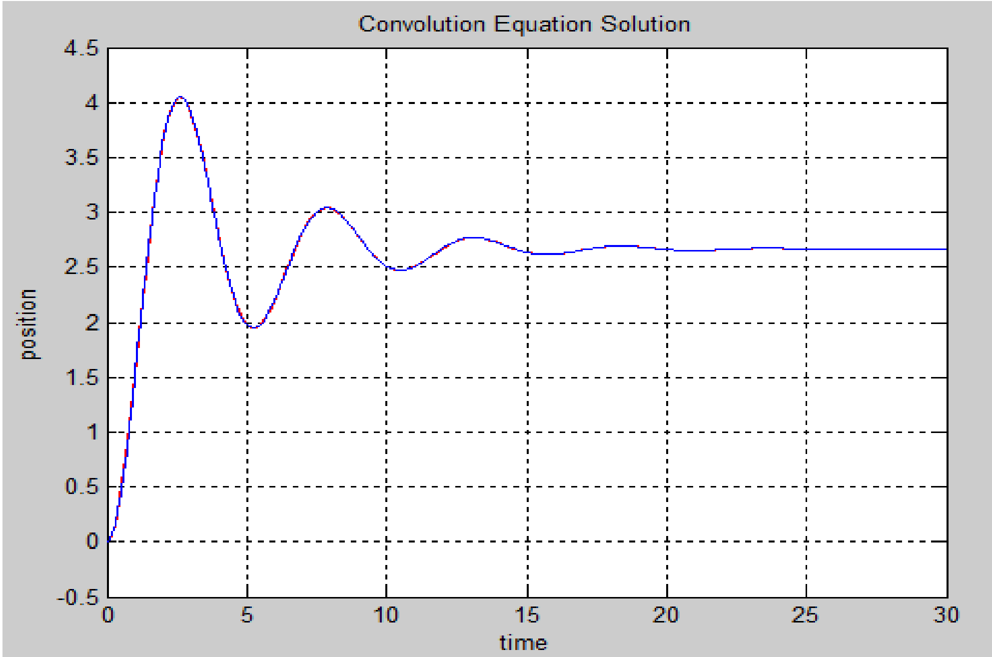
$$\mathbf{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}u$$

$$u(t) = \begin{cases} 0 & t < t_0 \\ 1 & t \geq t_0 \end{cases}$$

$$z_1(0) = 0$$
$$z_2(0) = 0$$

$$\begin{aligned}
y(t) &= \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \int_0^t \mathbf{C}e^{A(\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D}u \\
&= 0 + \mathbf{C}\int_0^t e^{\mathbf{A}(\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D} \\
&= \mathbf{C}\left[\mathbf{A}^{-1}e^{\mathbf{A}(\tau)}\mathbf{B}\right]_{\tau=o}^{\tau=t} + \mathbf{D} \\
&= \mathbf{C}\left[\mathbf{A}^{-1}e^{\mathbf{A}(t)}\mathbf{B} - \mathbf{A}^{-1}e^{\mathbf{A}(0)}\mathbf{B}\right] + \mathbf{D} \\
&= \mathbf{C}\mathbf{A}^{-1}e^{\mathbf{A}t}\mathbf{B} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{D}
\end{aligned}$$

Example: Mass – spring -damper

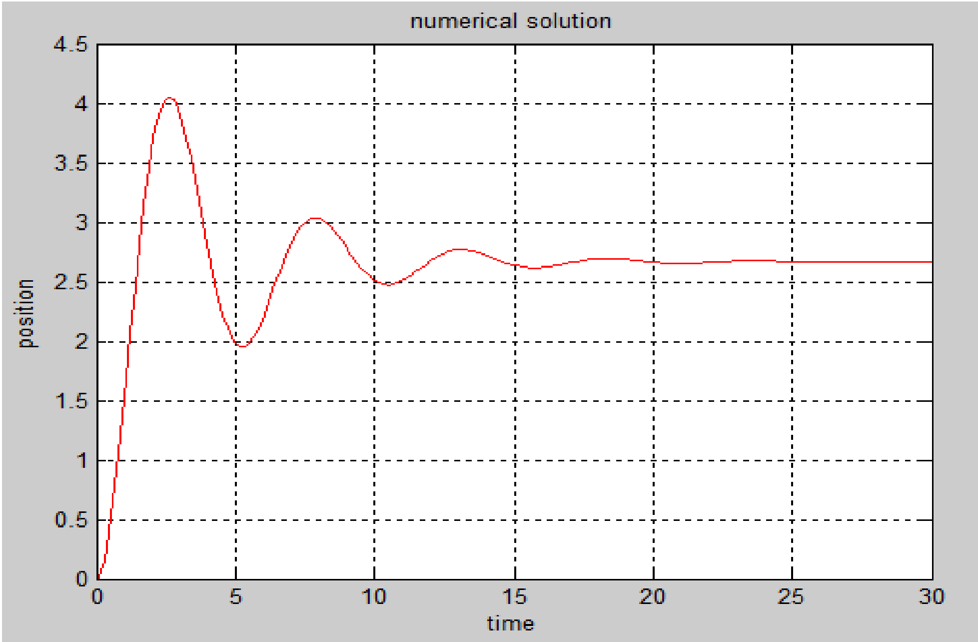



```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

[S, E] = eig(A);

tSpan = 0:.1:30;
for i = 1:length(tSpan)
    ME = [ exp(E(1,1)*tSpan(i)) 0; 0 exp(E(2,2)*tSpan(i)) ]; %could have used ME = expm(A)
    temp = C*inv(A)*S*ME*inv(S)*B - C*inv(A)*B + D;
    y(i,:) = real(temp'); %first check, then remove small numerical errors.

end
plot( tSpan, y(:,1), 'b' );
```

Example: Mass – spring -damper

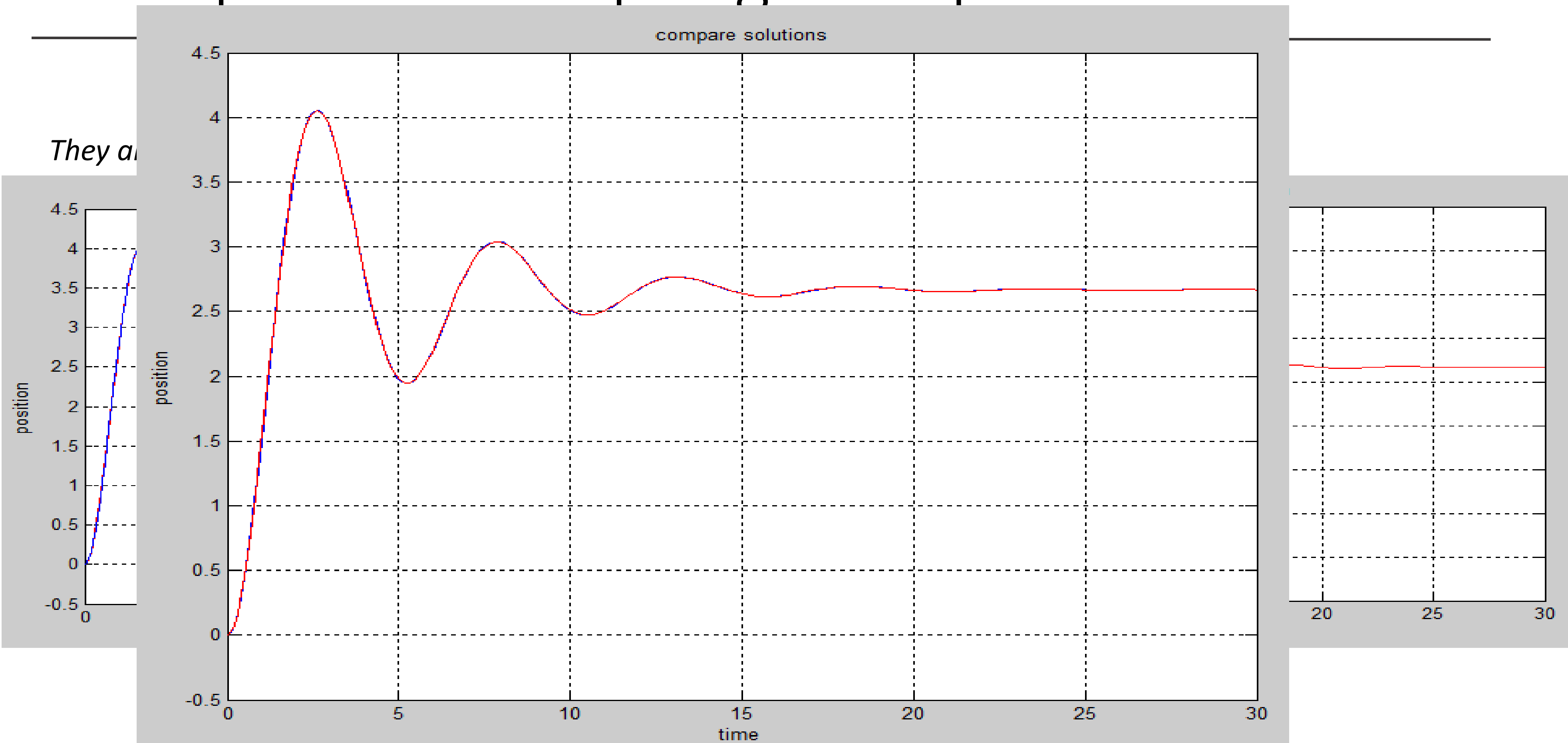
Double Check…

```
%numerical solution
[t, z] = ode45( @test, [0 30 ], [ 0 0 ]);

figure(2)
plot( t, z(:,1), 'r');

function zprime = test( t, z )

m = 2; c = 1; k = 3; F = 4;

zprime = [
    z(2);
    -(c/m)*z(2) - (k/m)*z(1) + 4;
];
```



Example: Mass – spring -damper

They a



Convolution and Transformations

- A property of matrix exponentials is that

$$e^{\mathbf{A}} = e^{\mathbf{T}\boldsymbol{\Lambda}\mathbf{T}^{-1}} = \mathbf{T}\, e^{\boldsymbol{\Lambda}}\, \mathbf{T}^{-1}$$

- Therefore

$$y(t) = \mathbf{C}\, e^{\mathbf{A}(t-t_0)}\mathbf{z}(t_0) + \int_{t_0}^{t} \mathbf{C}\, e^{\mathbf{A}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)$$

$$y(t) = \mathbf{C}\,\mathbf{T}\, e^{\boldsymbol{\Lambda}(t-t_0)}\mathbf{T}^{-1}\mathbf{T}\,\mathbf{z}(t_0) + \int_{t_0}^{t} \mathbf{C}\,\mathbf{T}\, e^{\boldsymbol{\Lambda}(t-\tau)}\mathbf{T}^{-1}\mathbf{T}\,\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)$$

$$y(t) = \mathbf{C}\,\mathbf{T}\, e^{\boldsymbol{\Lambda}(t-t_0)}\mathbf{z}(t_0) + \int_{t_0}^{t} \mathbf{C}\,\mathbf{T}\, e^{\boldsymbol{\Lambda}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)$$

$$y(t) = \mathbf{C}\,\mathbf{T}\, e^{\boldsymbol{\Lambda}(t-t_0)}\mathbf{z}(t_0) + \mathbf{C}\,\mathbf{T}\int_{t_0}^{t} e^{\boldsymbol{\Lambda}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)$$

We have a complete solution!

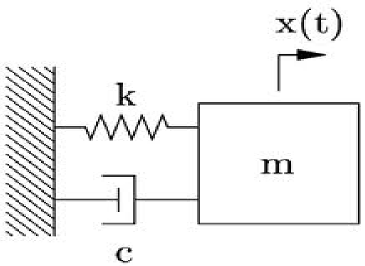
$$\mathrm{y}(t) = c e^{a t} x(0) + \int_0^t c e^{\mathrm{a}(t-\tau)} u(\tau) d\tau + d u(t)$$

is known as the convolution equation. We have just derived it using the Matrix Exponential for a system of 1st order differential equations.

$$\mathbf{y}(t) = \mathbf{C} e^{\mathbf{A} t} \mathbf{z}(0) + \int_0^t \mathbf{C} e^{\mathbf{A}(t-\tau)} \mathbf{B} u(\tau) d\tau + \mathbf{D} u(t)$$

- Now have complete analytical solution for sets of first order equations
  - Handle non-zero initial conditions
  - Handle multiple inputs
- MATLAB provides many tools to solve these equations for you.
- Next step: Let’s look at some of these tricks.

Example: Mass – spring -damper



$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} + \mathbf{B}u$$
$$\mathbf{y} = \mathbf{C}\mathbf{z} + \mathbf{D}u$$

$$\frac{d\mathbf{z}}{dt} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{k}{m} \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ \frac{F}{m} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\mathbf{z} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

$$u(t) = \begin{cases} 0 & t < t_0 \\ 1 & t \geq t_0 \end{cases}$$

$$z_1(0) = 0$$
$$z_2(0) = 0$$

$$\begin{aligned} y(t) &= \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \int_0^t \mathbf{C}e^{A(\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D}u \\ &= 0 + \mathbf{C}\int_0^t e^{\mathbf{A}(\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D} \\ &= \mathbf{C}\left[\mathbf{A}^{-1}e^{\mathbf{A}(\tau)}\mathbf{B}\right]_{\tau=o}^{\tau=t} + \mathbf{D} \\ &= \mathbf{C}\left[\mathbf{A}^{-1}e^{\mathbf{A}(t)}\mathbf{B} - \mathbf{A}^{-1}e^{\mathbf{A}(0)}\mathbf{B}\right] + \mathbf{D} \\ &= \mathbf{C}\mathbf{A}^{-1}e^{\mathbf{A}t}\mathbf{B} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{D} \end{aligned}$$

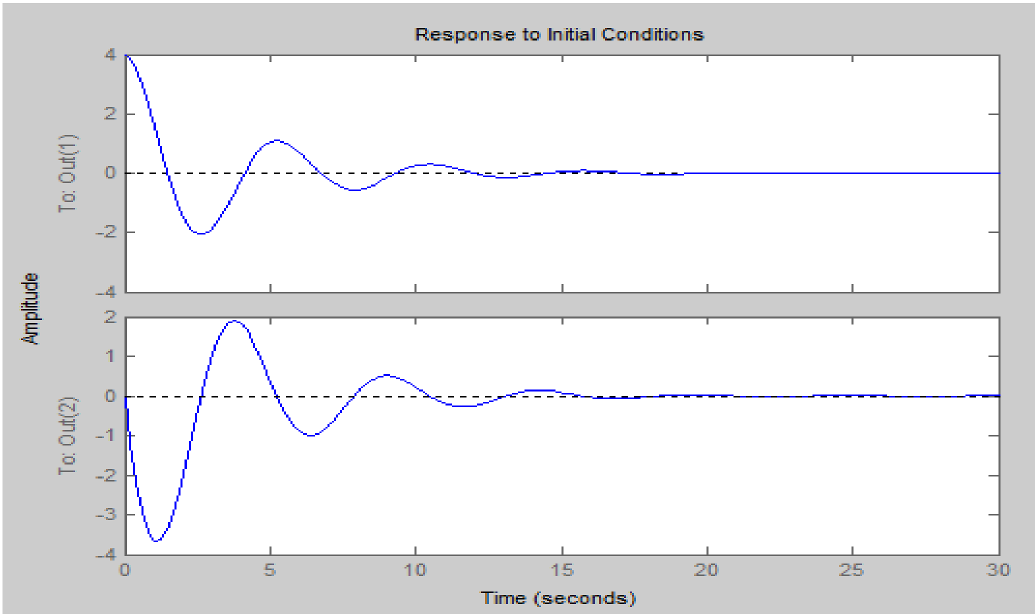
Other Cool MATLAB tricks

$$\frac{d\mathbf{z}}{dt} = \begin{bmatrix} 0 & 1 \\ -\dfrac{c}{m} & -\dfrac{k}{m} \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ \dfrac{F}{m} \end{bmatrix} u$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{z} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} u$$

$$y(t) = \underbrace{\mathbf{C} e^{\mathbf{A}t} \mathbf{z}(0)} + \int_o^t \mathbf{C} e^{\mathbf{A}(t-\tau)} \mathbf{B} u(\tau) d\tau + \mathbf{D} u(t)$$

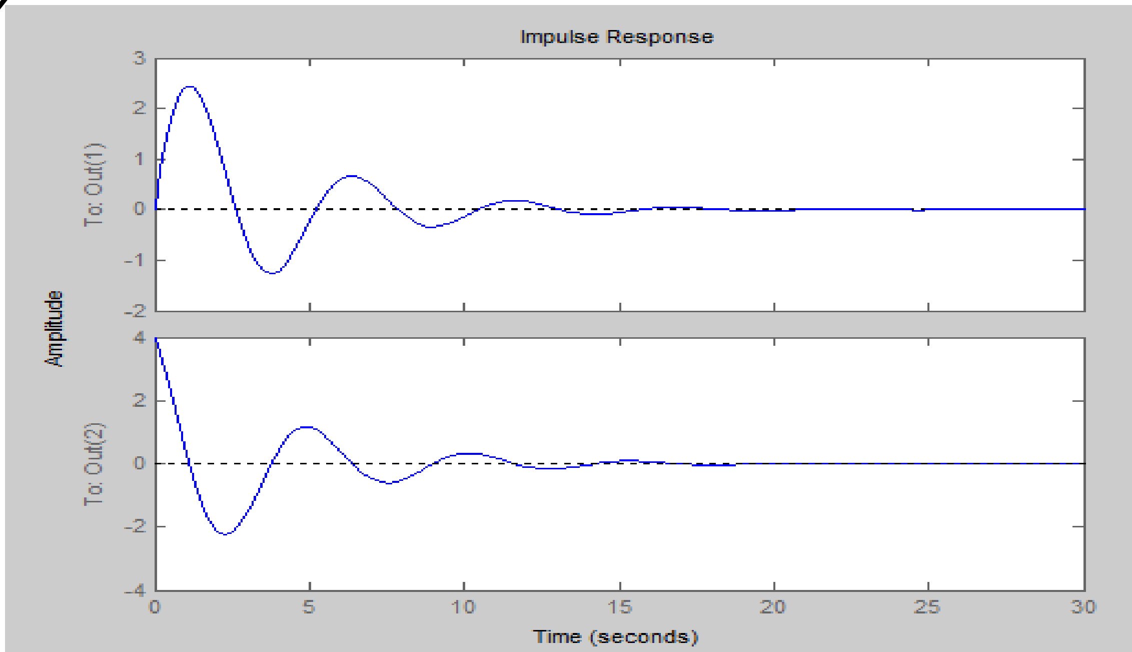
```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

sys = ss( A, B, C, D );

initial( sys, [ 4 0 ] );
```



Other Cool MATLAB tricks

$$y(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \underbrace{\int_o^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau + \mathbf{D}u(t)}$$

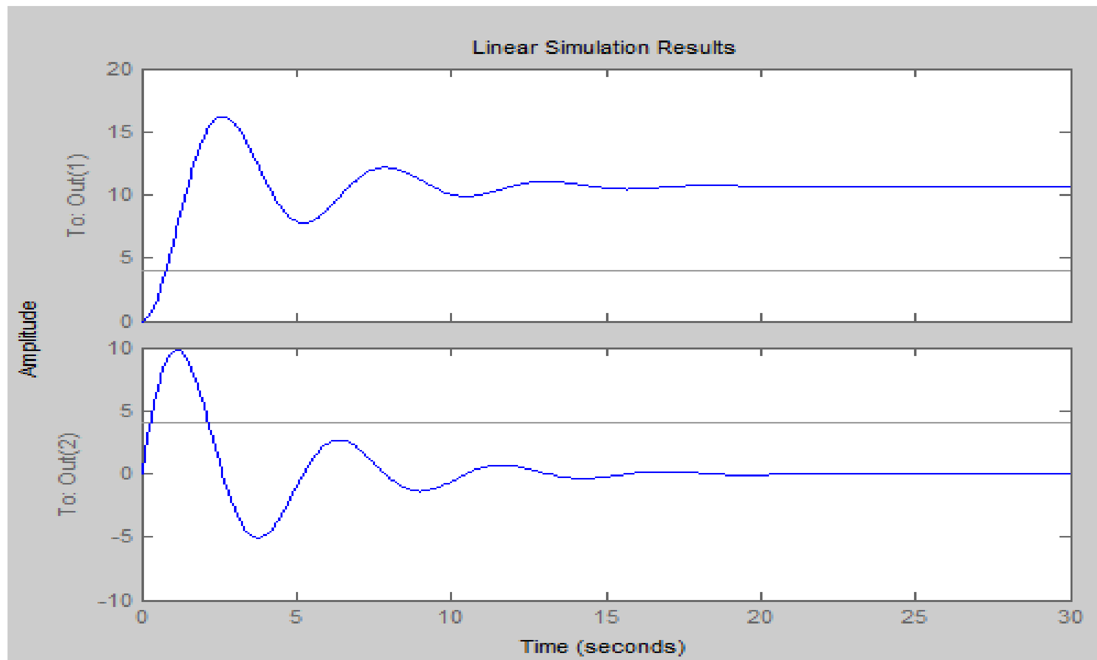
```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

sys = ss( A, B, C, D );
impulse( sys );
```



Other Cool MATLAB tricks (lsim)

$$\mathbf{y}(t) = \underbrace{\cancel{\mathbf{C}\, e^{\mathbf{A}t}\mathbf{z}(0)} + \int_{o}^{t} \mathbf{C}\, e^{\mathbf{A}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)}$$

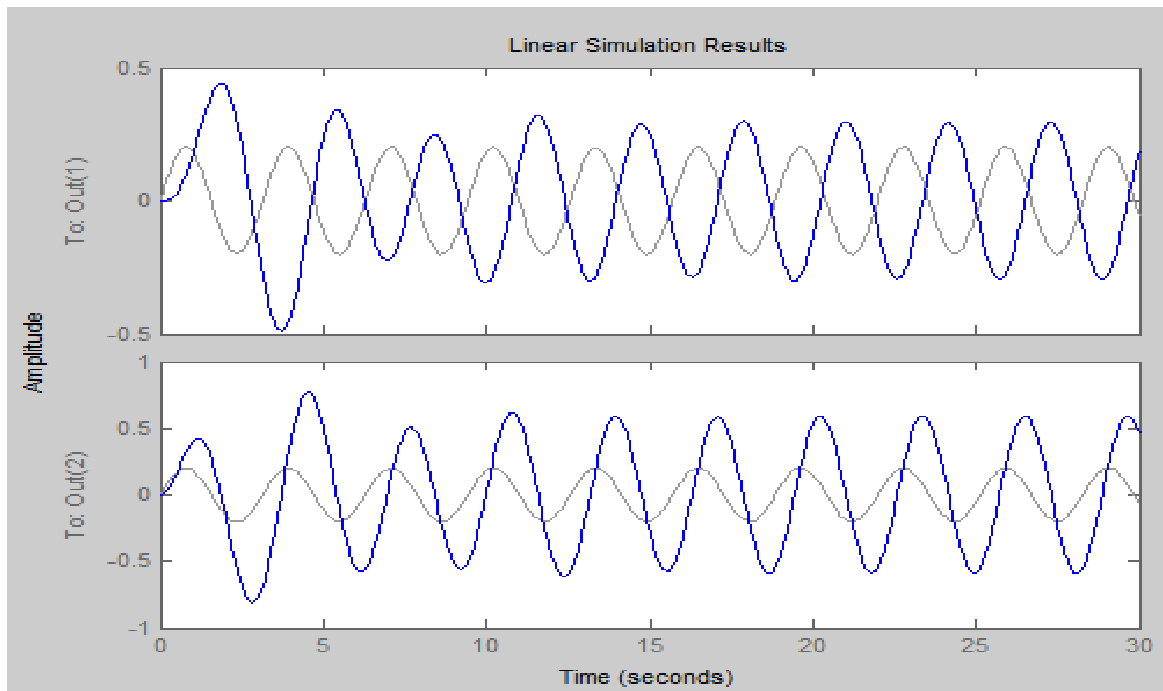
```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

sys = ss( A, B, C, D );
t = 0:0.1:30;
for i = 1:length(t)
    u(i) = 4;
end
lsim(sys, u, t, [0 0 ]);
```



Other Cool MATLAB tricks (lsim)

$$\mathbf{y}(t) = \underbrace{\cancel{\mathbf{C}\, e^{\mathbf{A}t}\mathbf{z}(0)} + \int_{o}^{t} \mathbf{C}\, e^{\mathbf{A}(t-\tau)}\mathbf{B}\, u(\tau)\, d\tau + \mathbf{D}\, u(t)}$$

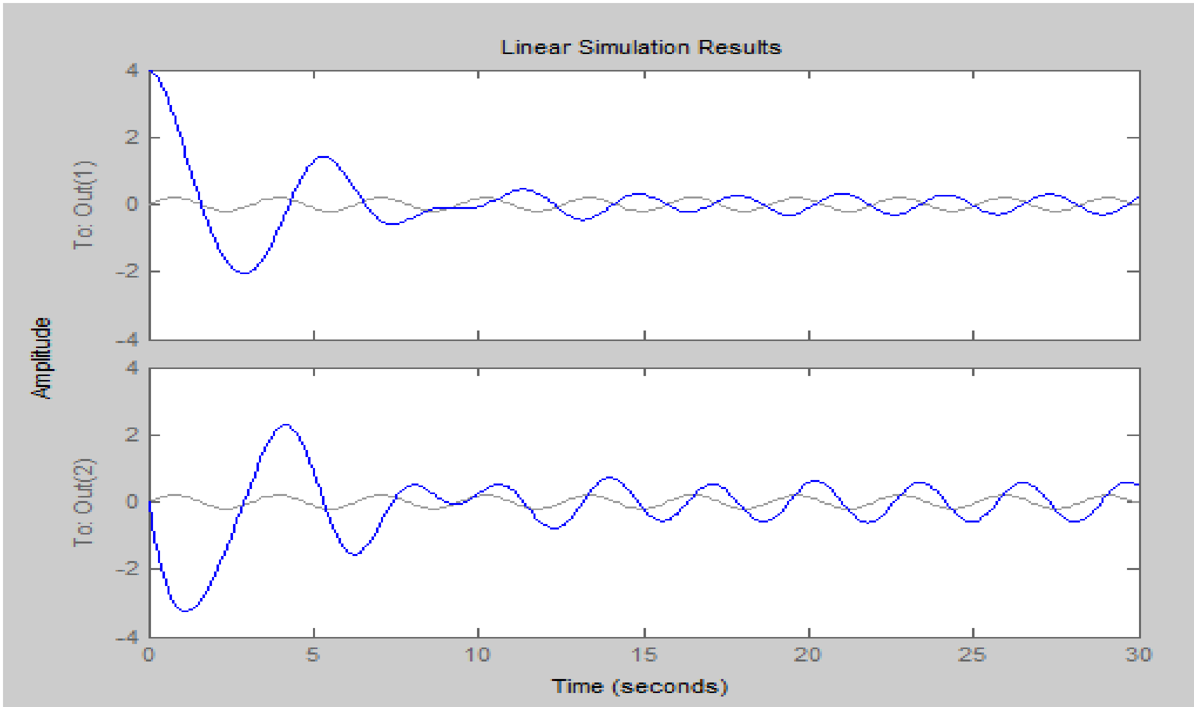
```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

sys = ss( A, B, C, D );
u = 0.2*sin(2*t);
lsim(sys, u, t, [0 0 ]);
```



Other Cool MATLAB tricks

$$\mathbf{y}(t) = \underbrace{\mathbf{C}\,e^{\mathbf{A}t}\mathbf{z}(0) + \int_o^t \mathbf{C}\,e^{\mathbf{A}(t-\tau)}\mathbf{B}\,u(\tau)d\tau + \mathbf{D}\,u(t)}$$

```
clear all;

m = 2; c = 1; k = 3; F = 4;

A = [ 0 1; -(k/m) -(c/m)];
B = [ 0; F ];
C = [ 1 0; 0 1];
D = [ 0; 0 ];

sys = ss( A, B, C, D );
u = 0.2*sin(2*t);
lsim(sys, u, t, [4 0 ]);
```



Summary

$$y(t) = ce^{at}x(0) + \int_0^t ce^{a(t-\tau)}u(\tau)\,d\tau + du(t)$$

is known as the convolution equation. We have just derived it using the Matrix Exponential for a system of 1st order differential equations.

$$\mathbf{y}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{z}(0) + \int_0^t \mathbf{C}e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)\,d\tau + \mathbf{D}u(t)$$

- Now have complete analytical solution for sets of first order equations
  - Handle non-zero initial conditions
  - Handle the summation of scalar inputs
- MATLAB provides many tools to solve these equations for you.