ROCO218: Control Engineering

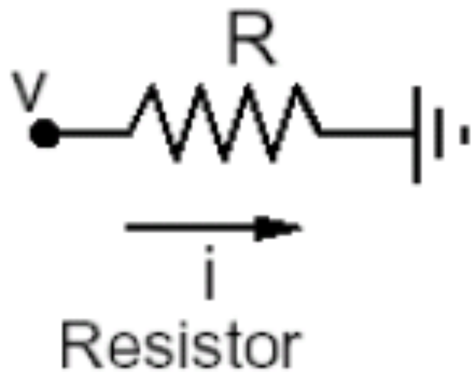
Dr Ian Howard

Lecture 2

Modelling electrical circuits with differential equations

Electrical resistance

Resists the flow of current



$$v = iR$$

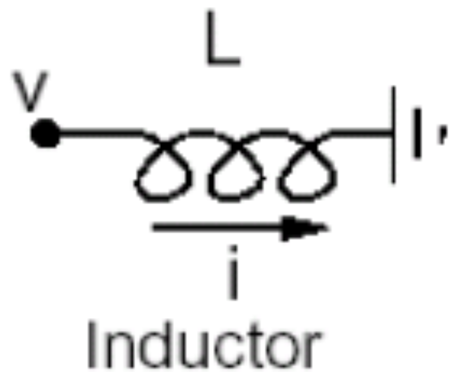
Where

i is the current in A

v is voltage in V

R is the resistance in Ω

Electrical inductance



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

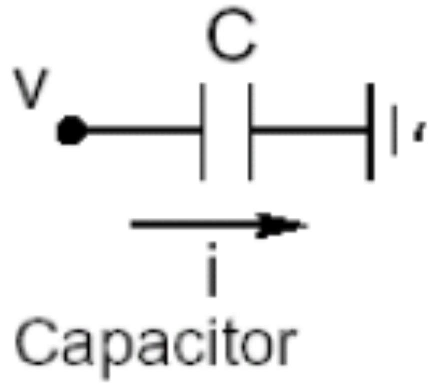
Where

i is the current in A

v is voltage in V

L is the inductance in H

Electrical capacitance



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

Therefore

$$v = \frac{1}{C}\int i(t).dt$$

Where

i is the current in A

v is voltage in V

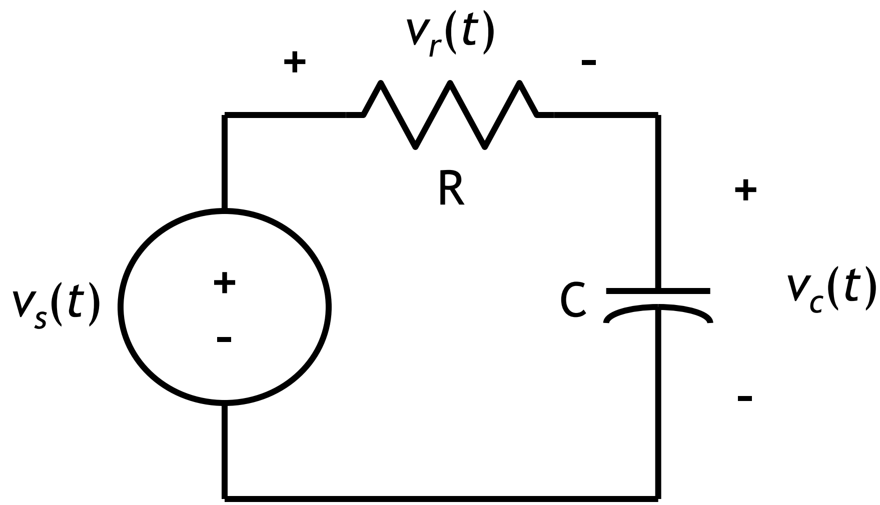
C is the capacitance in F

Also

$$Q = Cv$$

Where Q is charge

A First-order RC Circuit



- One capacitor and one resistor in series
- The source and resistor may be equivalent to a circuit with many resistors and sources

Voltages in an RC circuit

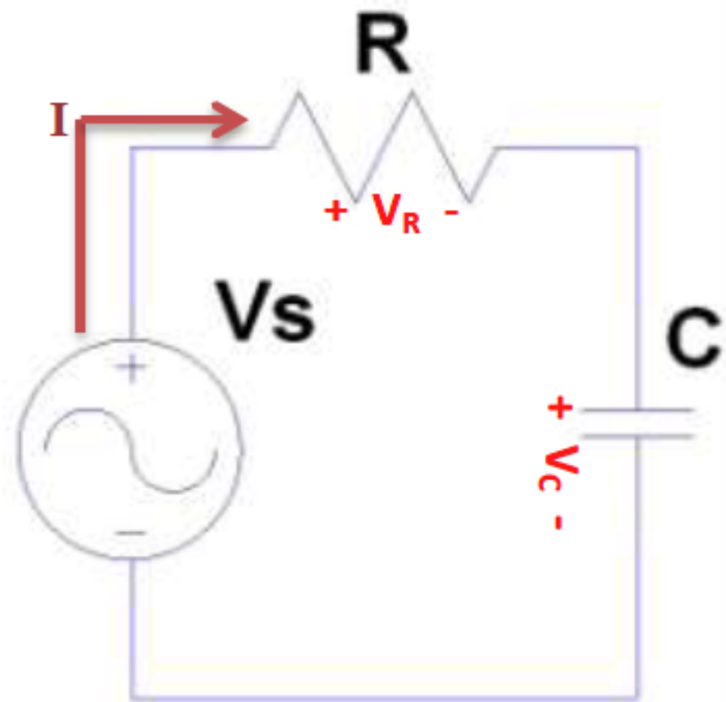
$$V_S = V_R + V_C$$ Adding up voltages

$$V_R = IR$$ Ohms law

$$I = \frac{dQ}{dt} = C\frac{dV_C}{dt}$$ Current proportional to rate of change of charge

$$V_S = RC\frac{dV_C}{dt} + V_C$$ Substitute I from capacitor expression

$$\frac{dV_C}{dt} = \frac{1}{RC}\left(V_S - V_C\right)$$ Rearranged



This is a first order linear differential equation

Current in an LR circuit

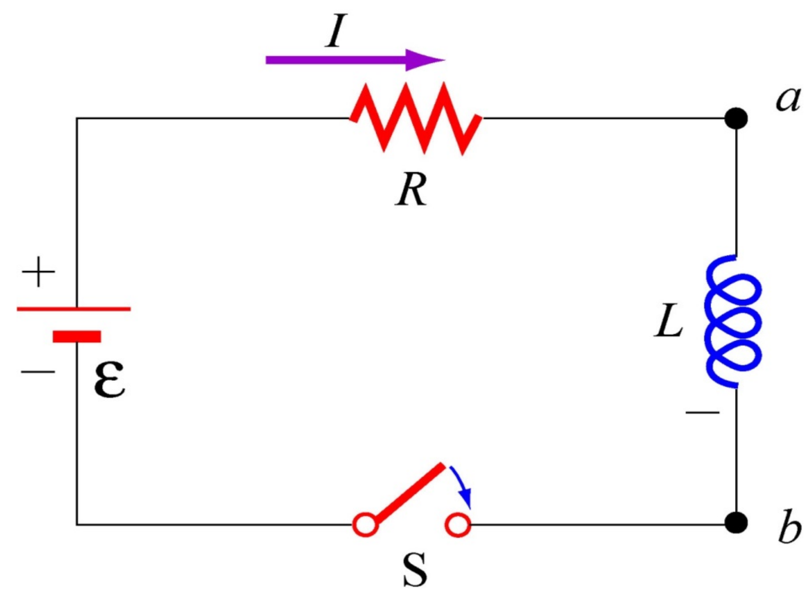
$$V_S = V_R + V_L$$ Adding up voltages

$$V_R = IR$$ Ohms law

$$V_L = L\frac{dI}{dt}$$ Voltage proportional to rate of change of current

$$V_S = IR + L\frac{dI}{dt}$$ Substituted values

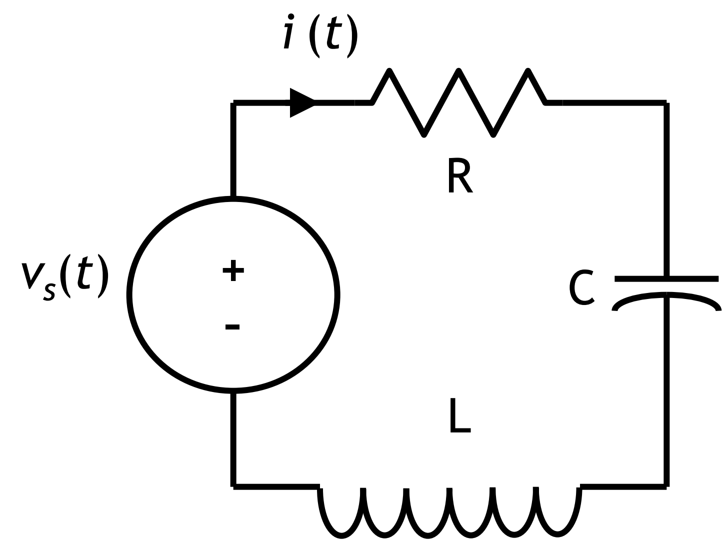
$$\frac{dI}{dt} = \frac{R}{L}\left(\frac{V_S}{R} - I\right)$$ Rearranged



This is also a first order linear differential equation

2nd order LCR circuits

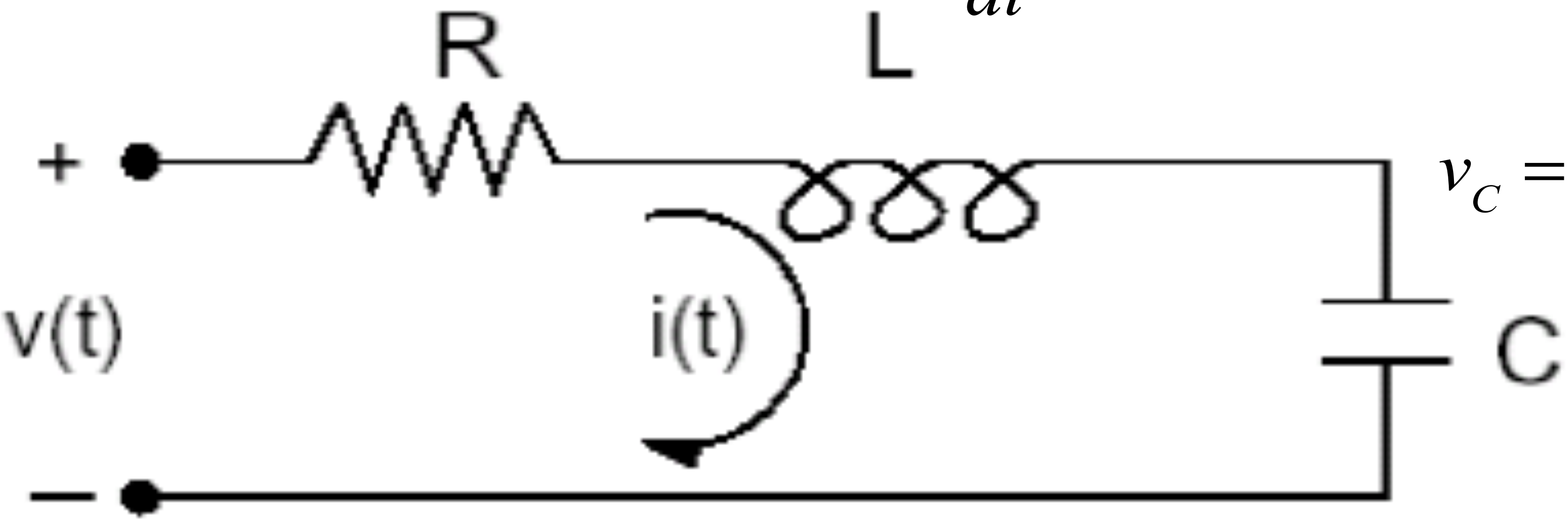
- Any circuit with a single capacitor, a single inductor, an arbitrary number of sources, and an arbitrary number of resistors is a circuit of order 2
- The source and resistor may be equivalent to a circuit with many resistors and sources
- Any voltage or current in such a circuit is the solution to a 2nd order differential equation



Series RLC circuit

$$v_R = iR$$

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



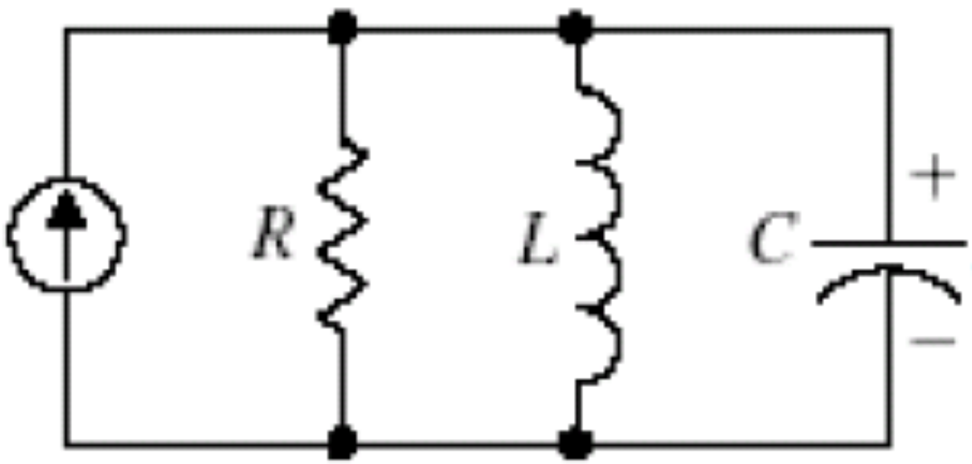
$$v_C = \frac{1}{C}\int i(t).dt$$

Adding up the contributions to overall applied voltage

$$v(t) = iR + L\frac{di}{dt} + \frac{1}{C}\int i(t).dt$$

Parallel RLC circuit

$$i_R = \frac{v}{R} \qquad i_L = \frac{1}{L}\int v\,dt$$



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

Adding up the contributions to overall applied current

$$i(t) = \frac{v}{R} + \frac{1}{L}\int v(t)\,dt + C\frac{dv}{dt}$$

Summary: RLC Characteristics

Element	V/I Relation	Comments
Resistor	$v_R(t) = R\, i_R(t)$	V = I R R has SI unit Ohms
Capacitor	$i_C(t) = C \dfrac{d\, v_C(t)}{dt}$	Q = CV C has SI unit is Farads
Inductor	$v_L(t) = L \dfrac{d\, i_L(t)}{dt}$	L has SI unit Henrys

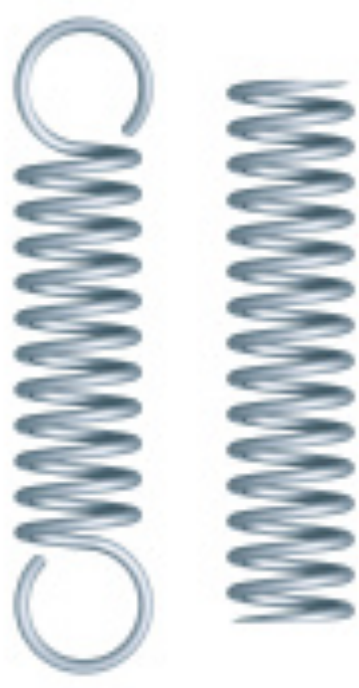
ROCO218: Control Engineering
Dr Ian Howard

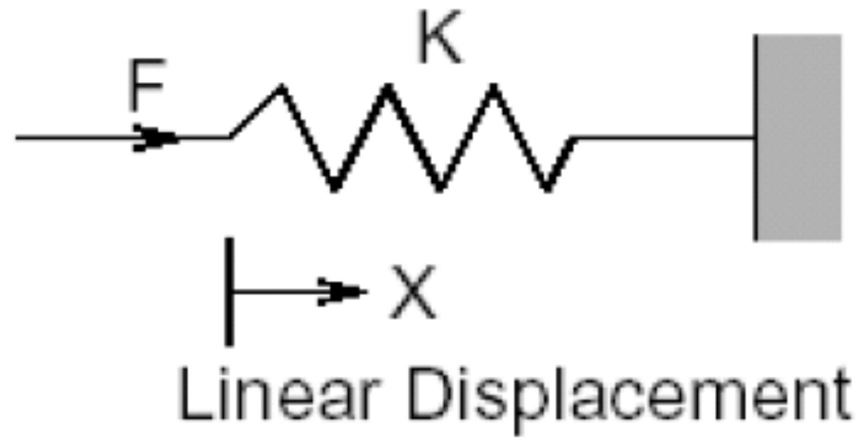
Lecture 2

Modelling mechanical systems
using differential equations

Compression & extension springs

Resists with opposing force proportional to extension of movement





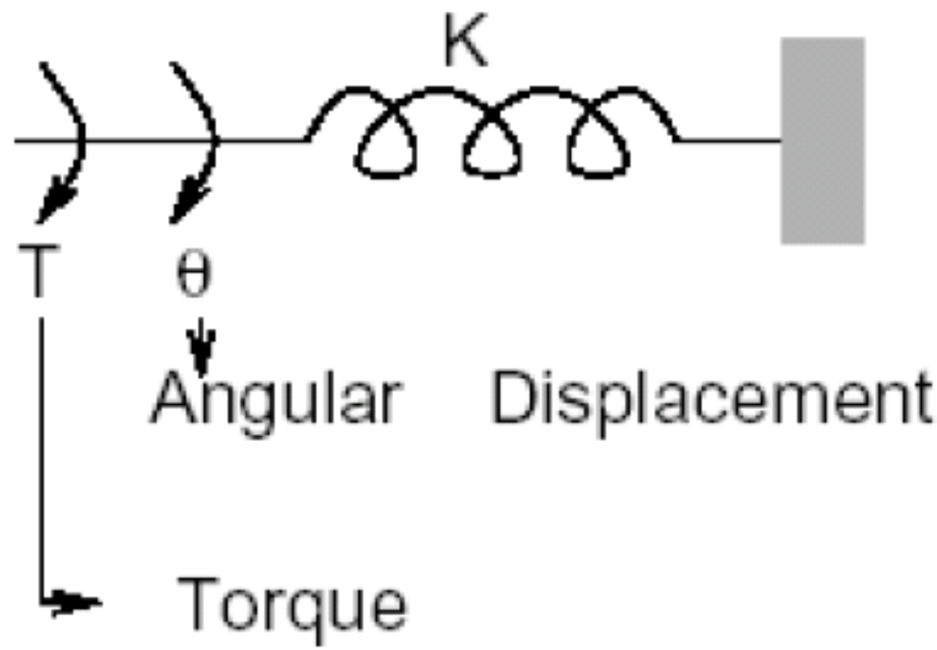
$$f = kx$$

where

f is force in N

k is the spring constant in N/m

Torsional springs



$$T = k\theta$$

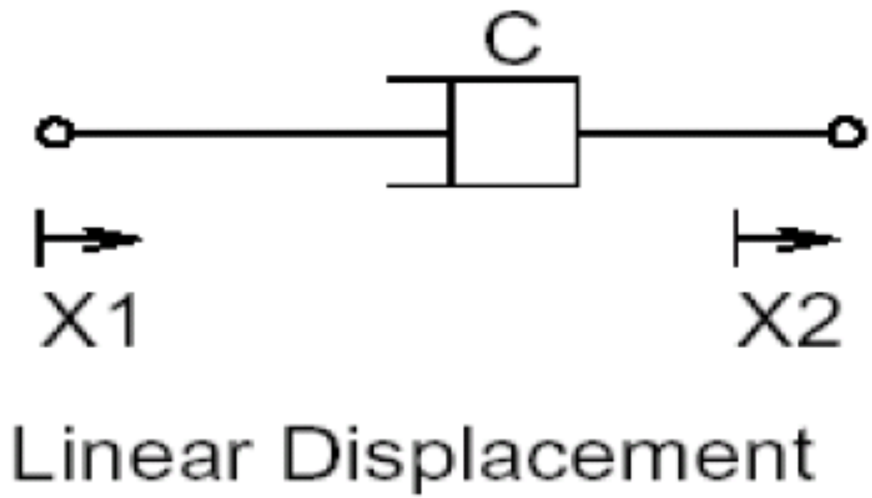
where

T is torque in Nm

k is the spring constant in Nm/rad

Dampers and dashpots

Resists with opposing force proportional to velocity of movement
Like moving your hand in water – resistance proportional to how fast you move



$$f = C\frac{dx}{dt}$$

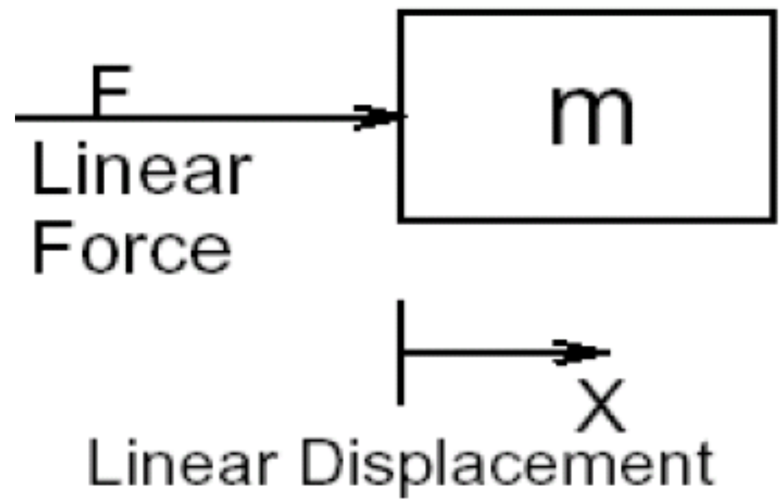
Where

f is force in N

C is viscosity in Ns/m

Inertial Mass

Resists with opposing force proportional to acceleration of movement



Where

f is force in N

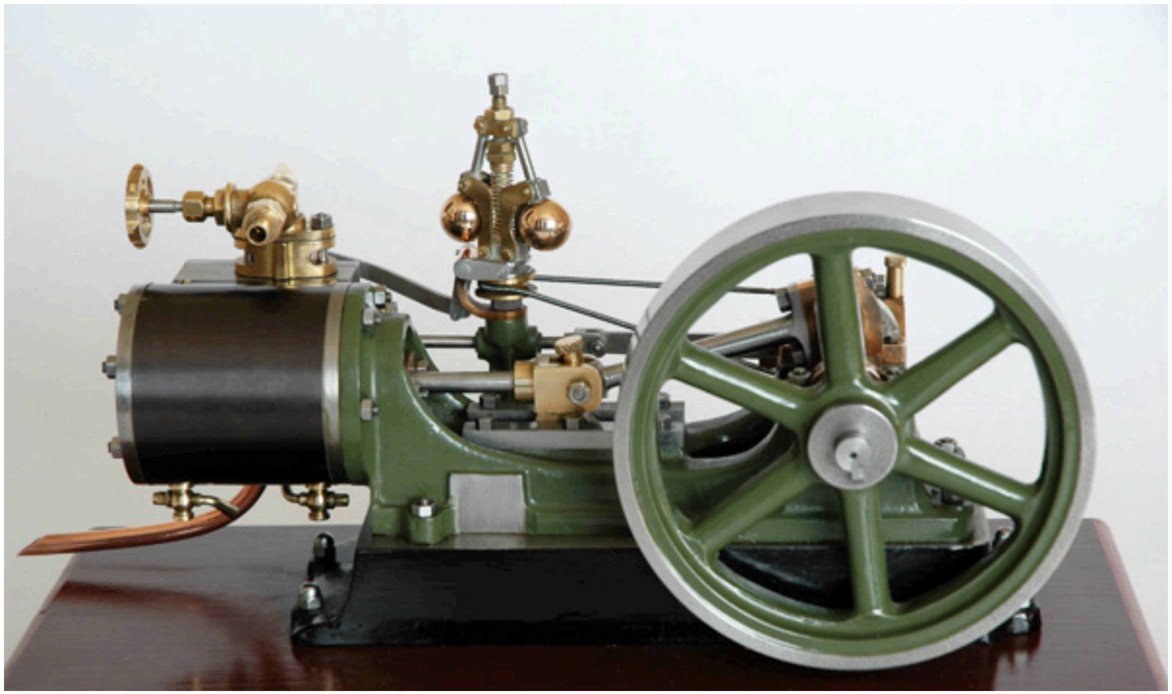
m is mass in Kg

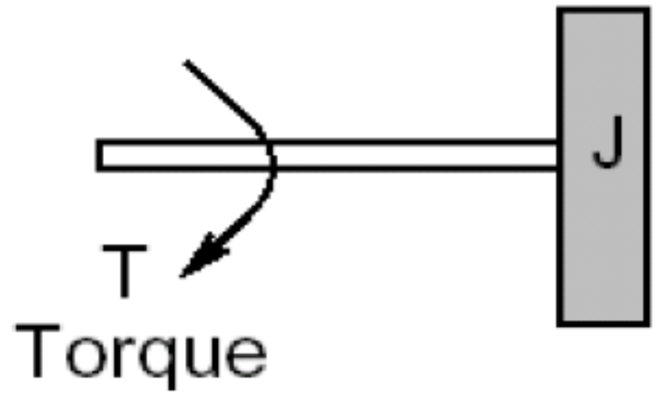
a is linear acceleration in m/s^2

$$f = m\frac{d^2x}{dt^2} = ma$$

Moment of Inertia

Resists with opposing torque proportional to angular acceleration





Where

T is torque in Nm

J is moment of inertial in Kgm^2

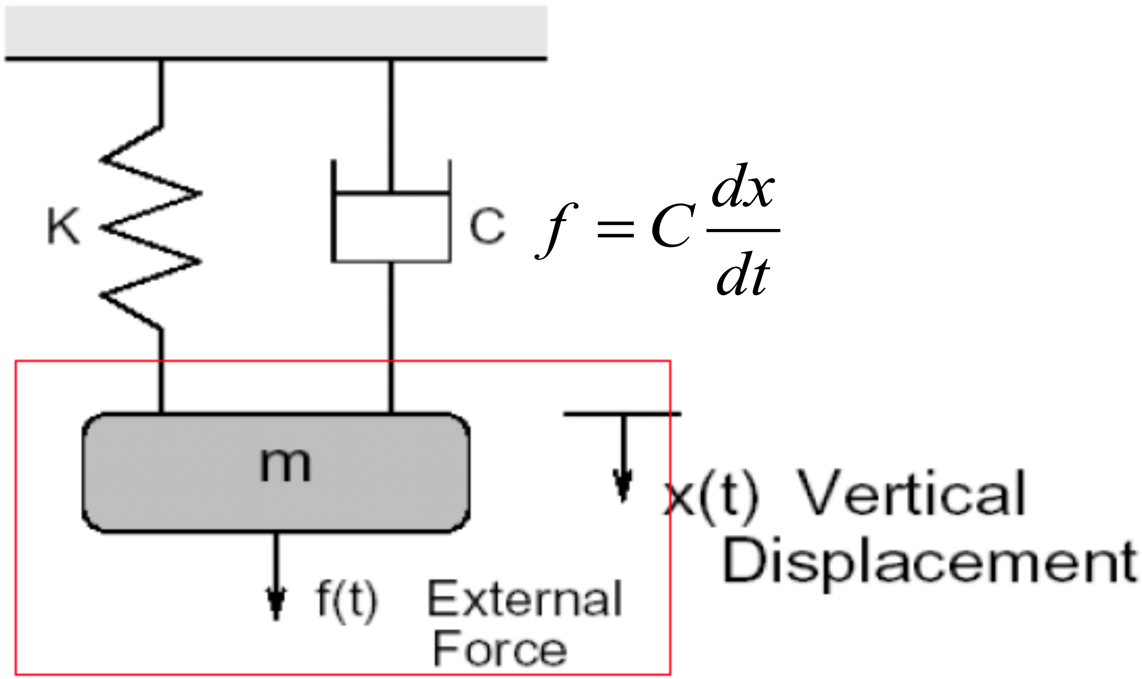
$$T = J\frac{d^2\theta}{dt^2}$$

Spring mass damper system

$$f = kx$$

$$f = C\frac{dx}{dt}$$

$$f = m\frac{d^2x}{dt^2}$$



Adding up the contributions to overall force

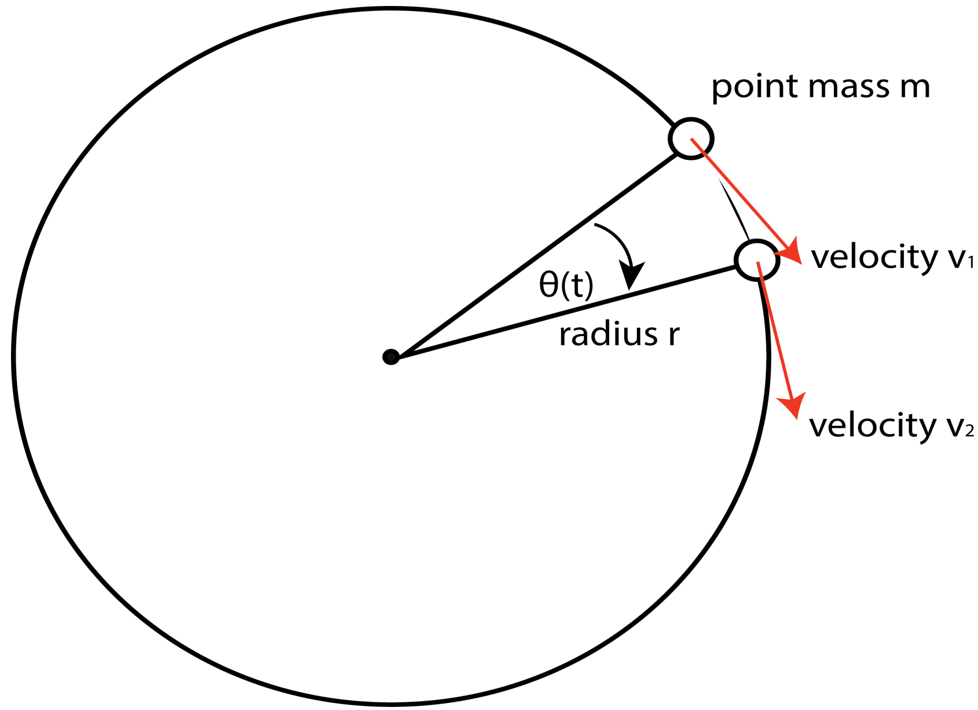
$$f(t) = m\frac{d^2x}{dt^2} + C\frac{dx}{dt} + kx$$

ROCO218: Control Engineering
Dr Ian Howard

Lecture 2
Circular movement

Constant speed circular movement

Consider a point mass travelling around central point in a circle at a constant speed with rotational period
Therefore in T seconds is rotates though an angle of 2π radians
Thus angular velocity ω is given by



$$\omega = \frac{2\pi}{T} \qquad \Rightarrow T = \frac{2\pi}{\omega}$$

The distance travelled by the mass in time T seconds is given by

$$d_{circ} = 2\pi r$$

The average speed at which is does so it therefore

$$v = \frac{d_{circ}}{T} = \frac{2\pi r}{T}$$

Substituting in $T = \frac{2\pi}{\omega}$

$$\Rightarrow v = \frac{2\pi r}{\left(\frac{2\pi}{\omega}\right)}$$

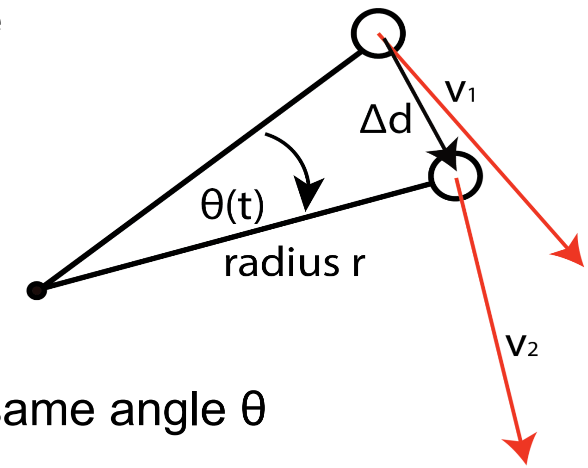
$$\boxed{\Rightarrow v = \omega r}$$

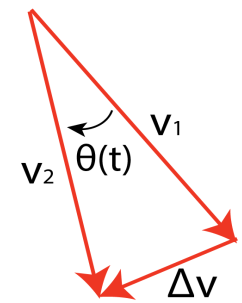
Note also
The rotational frequency can be related to rotational period T

$$f = \frac{1}{T} \qquad \Rightarrow \omega = 2\pi f$$

Simple derivation of centrifugal acceleration

Consider the moving mass
As the mass rotates through an angle of θ the tangential velocity of the mass changes from v_1 to v_2
Mass moves a distance Δd
Velocity magnitude stays constant but its direction changes so velocity changes by Δv





By consideration of similar triangles with the same angle θ

$$\frac{\Delta d}{r} = \frac{\Delta v}{v}$$

if Δd is very very small is can be approximated by $\Delta d \approx v \Delta t$

Substituting in this value

$$\Rightarrow \frac{v\Delta t}{r} = \frac{\Delta v}{v} \qquad \Rightarrow = \frac{\Delta v}{\Delta t} = \frac{v^2}{r}$$

As Δt tends to 0, term Δv/Δt represents acceleration

$$\Rightarrow a = \frac{v^2}{r}$$

Since from before $v = \omega r$ $\quad \Rightarrow a = \omega^2 r$

Chain rule for single variable differentiation

- Consider the one variable equation corresponding to a function of a function

$$z = f(g(x))$$

- Writing

$$y = g(x)$$

- Chain rule gives

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}$$

- For example if

$$z = e^{x^2}$$

- In this case let

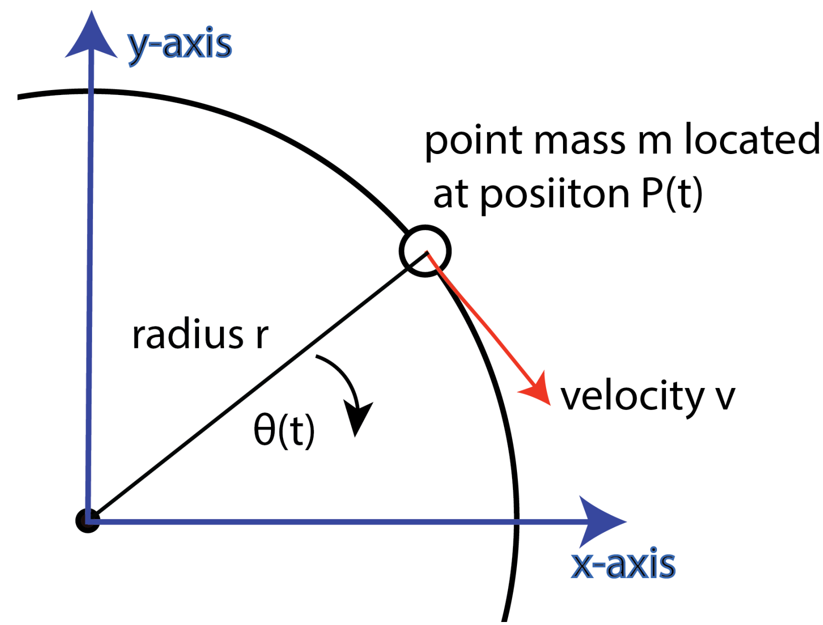
$$y = x^2 \qquad \Rightarrow \frac{dy}{dx} = 2x$$

$$z = e^y \qquad \Rightarrow \frac{dz}{dy} = e^y$$

$$\Rightarrow \frac{dz}{dx} = e^y .2x \qquad \boxed{= 2xe^{x^2}}$$

Calculus derivation of centrifugal acceleration

We can write down the projections of the mass position along the x and y axes



$$x(t) = r\cos(\theta(t))$$

$$y(t) = r\sin(\theta(t))$$

We can write mass position vector P(t) as sum of its base vector components

$$P(t) = r\cos(\theta(t))\overline{x} + r\sin(\theta(t))\overline{y}$$

$$P(t) = r\left[\cos(\theta(t))\overline{x} + \sin(\theta(t))\overline{y}\right]$$

Differentiating both sides wrt time t gives

$$\frac{dP(t)}{dt} = V(t) = r\frac{d}{dt}\left[\cos(\theta(t))\overline{x} + \sin(\theta(t))\overline{y}\right]$$

$$\Rightarrow V(t) = r\left[-\sin(\theta(t))\frac{d\theta}{dt}\overline{x} + \cos(\theta(t))\frac{d\theta}{dt}\overline{y}\right]$$

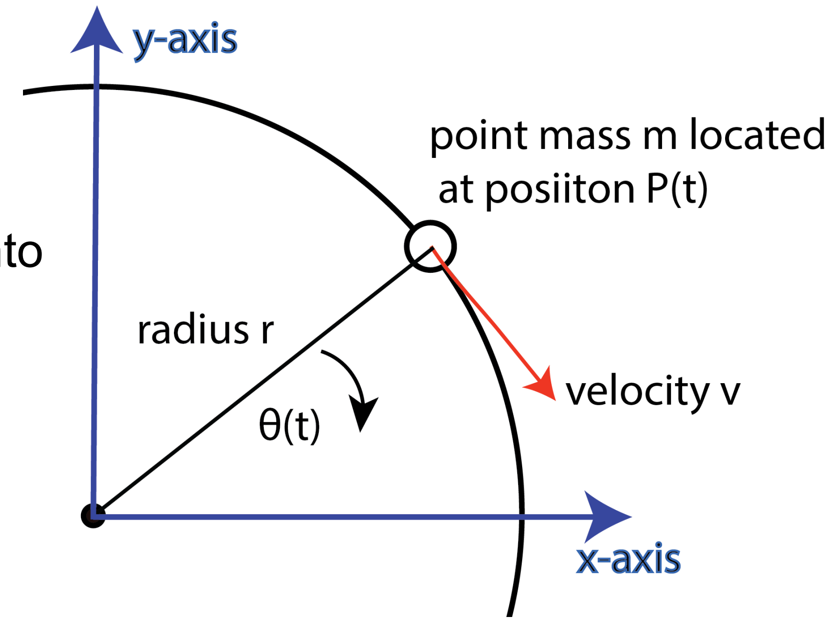
$$\Rightarrow V(t) = -r\frac{d\theta}{dt}\left[\sin(\theta(t))\overline{x} - \cos(\theta(t))\overline{y}\right]$$

Calculus derivation of centrifugal acceleration

Now angular velocity is given by

$$\frac{d\theta}{dt} = \omega$$

Which is a constant if speed is constant. Substituting into



$$V(t) = -r\frac{d\theta}{dt}\left[\sin(\theta(t))\overline{x} - \cos(\theta(t))\overline{y}\right]$$

$$\Rightarrow V(t) = -r\omega\left[\sin(\theta(t))\overline{x} - \cos(\theta(t))\overline{y}\right]$$

Differentiating the expression again

$$\Rightarrow \frac{dV(t)}{dt} = -r\omega\frac{d}{dt}\left[\sin(\theta(t))\overline{x} - \cos(\theta(t))\overline{y}\right]$$

$$\Rightarrow \frac{dV(t)}{dt} = -r\omega\left[\cos(\theta(t))\frac{d\theta}{dt}\overline{x} + \sin(\theta(t))\frac{d\theta}{dt}\overline{y}\right] = -r\omega^2\left[\cos(\theta(t))\overline{x} + \sin(\theta(t))\overline{y}\right]$$

$$\Rightarrow \frac{dV(t)}{dt} = -w^2 P(t)$$

Calculus derivation of centrifugal acceleration

Now acceleration A(t) is given by

$$A(t) = \frac{dV(t)}{dt}$$

$$\Rightarrow A(t) = -\omega^2 P(t)$$

Taking the modulus of both sides

$$\Rightarrow |A(t)| = |-w^2 P(t)|$$

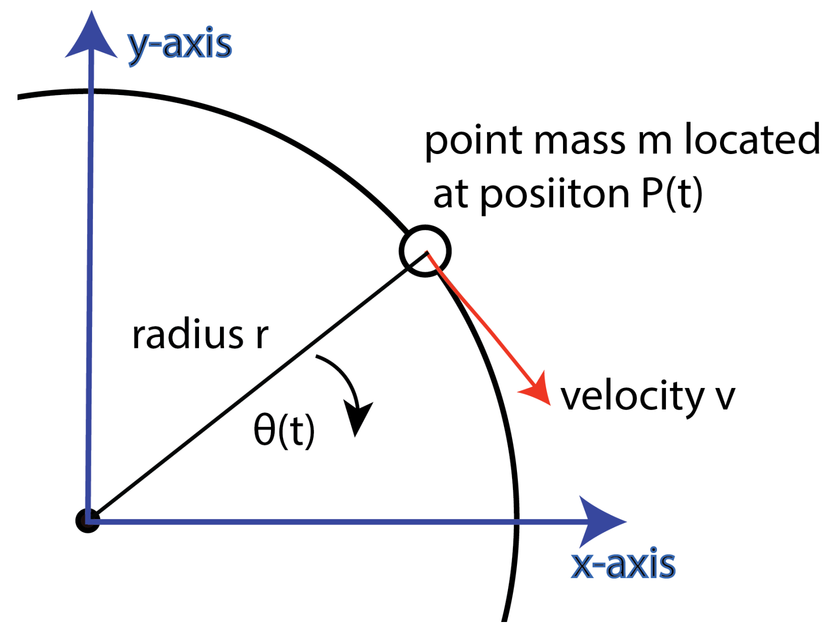
Note that $|P(t)|$ Is just the radius of the circle

$$\Rightarrow a = \omega^2 r$$

Or as from before

$$v = \omega r$$

$$\Rightarrow a = \frac{v^2}{r}$$



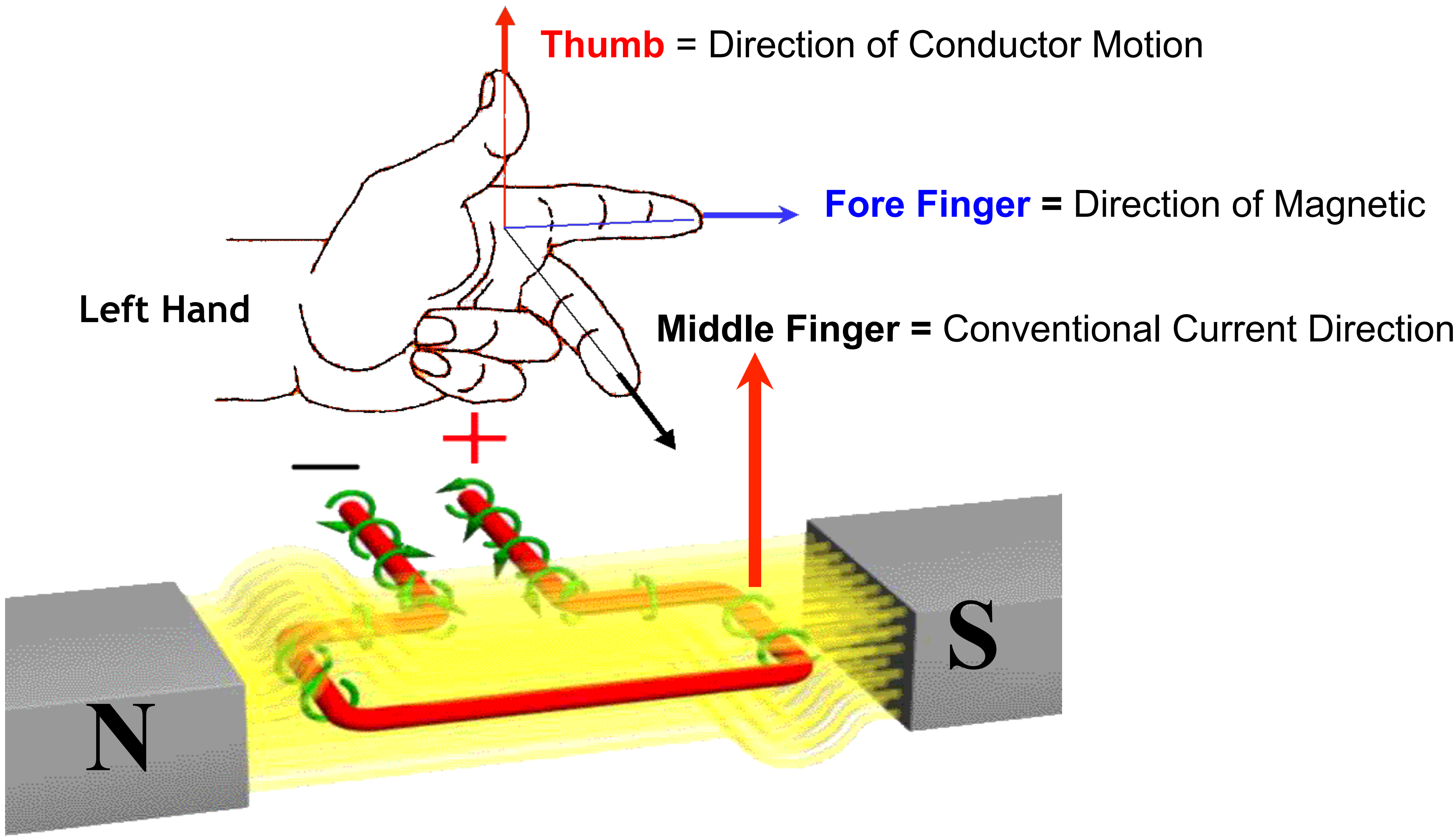
ROCO218: Control Engineering
Dr Ian Howard

Lecture 2

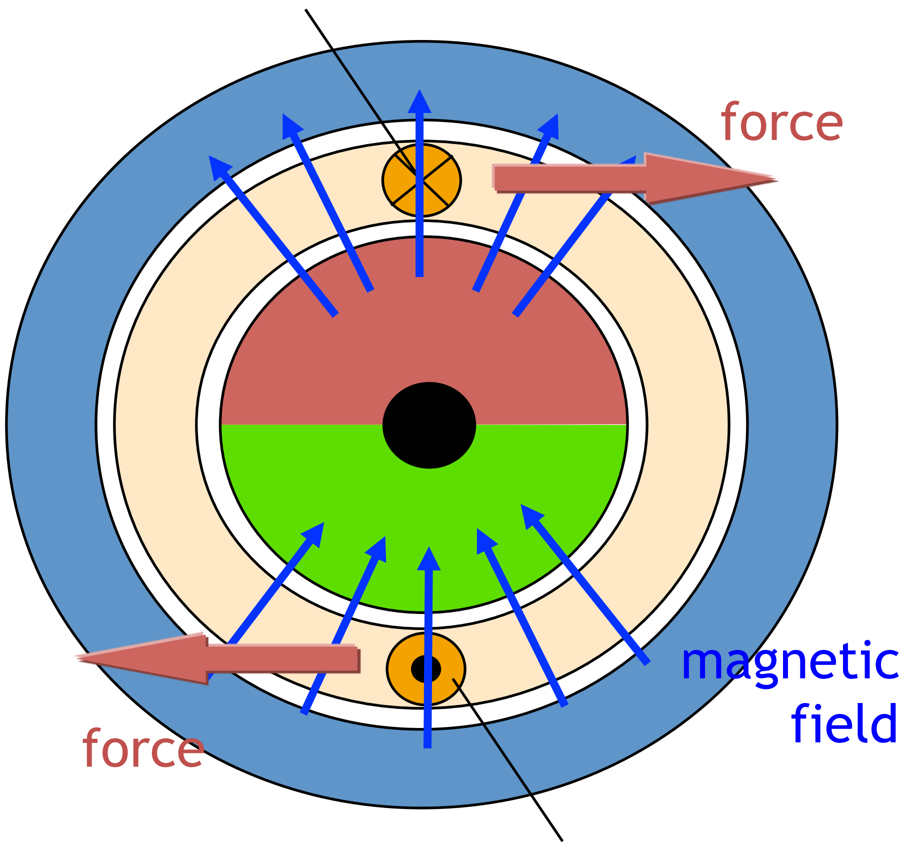
Modelling a DC motor

Fleming's Left Hand (Motor) Rule

The Left Hand Rule to determines the movement direction of conductor carrying current



Motor Torque constant K_t



forces:
force on current leading conductor in a magnetic field

torque:
sum of all forces at the distance to the rotating axis

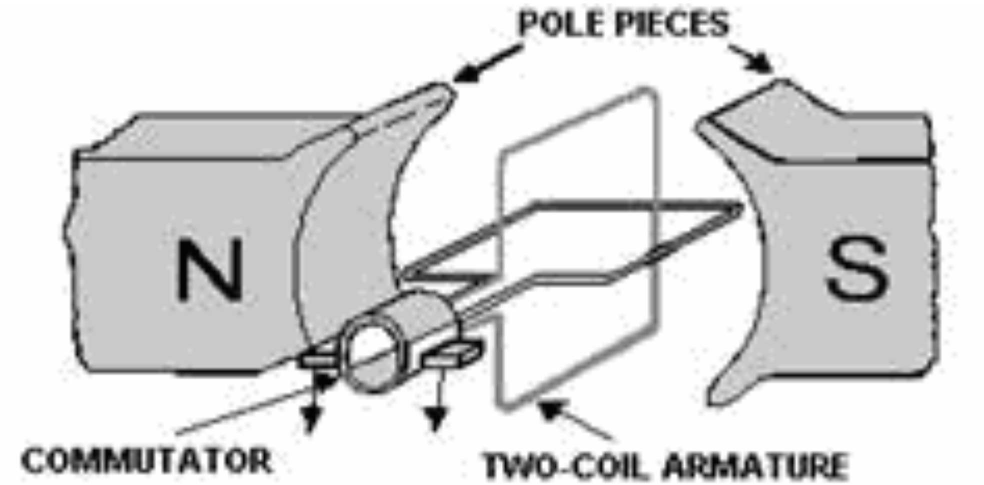
influencing parameters:

$$\left.\begin{array}{l}\text{geometry}\\\text{field density}\\\text{winding number}\end{array}\right\}\ \textbf{Depend on the design}$$

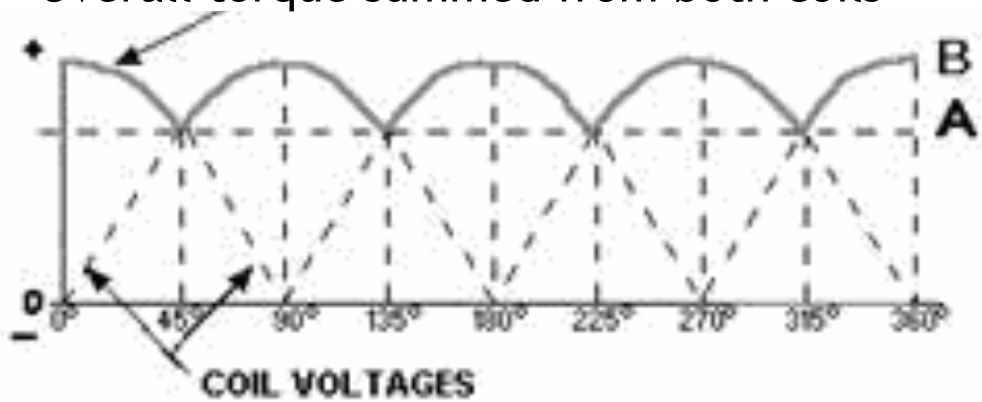
Motor torque T_m is given by

$$T_m = K_t i(t)$$

Multiple coils reduces torque ripple



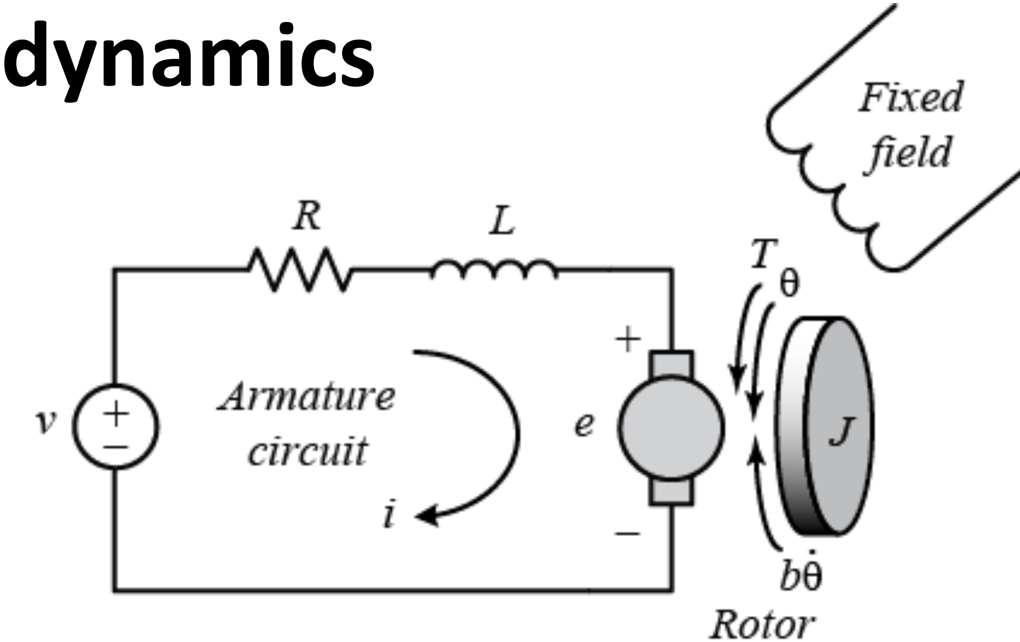
Overall torque summed from both coils



Typical small armature
with multiple multi-turn coils

DC motor dynamics

- V applied voltage, I is the current drawn
- R resistance of coil and brush contact
- L is the coil inductance
- EMF is back emf of motor



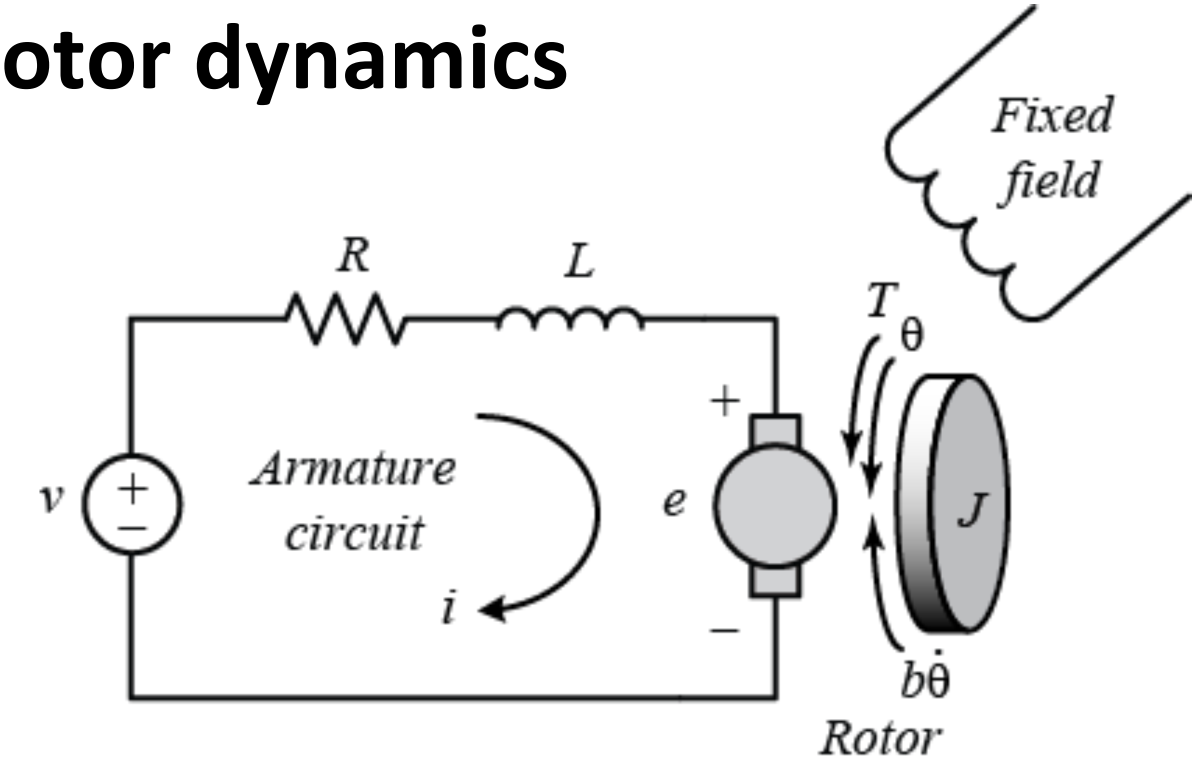
- Lets relate output speed of motor to the applied voltage!

- R = Armature resistance (in ohms)
- L = Armature inductance (in Henrys)
- J = Moment of inertia for the motor rotor (kg.m2)
- b = Motor viscous friction constant (in N.m.s)
- K_t = Motor torque constant (in N.m/A)
- K_e = Electromotive force constant (in V/rad/sec)

DC motor dynamics

Motor torque T_m is given by the current multiplied by the torque constant of the motor

$$T_m = K_t i(t)$$



Mechanical resisting torque T_r is given by inertia and viscous friction

$$T_r = b\frac{d\theta}{dt} + J\frac{d^2\theta}{dt^2}$$

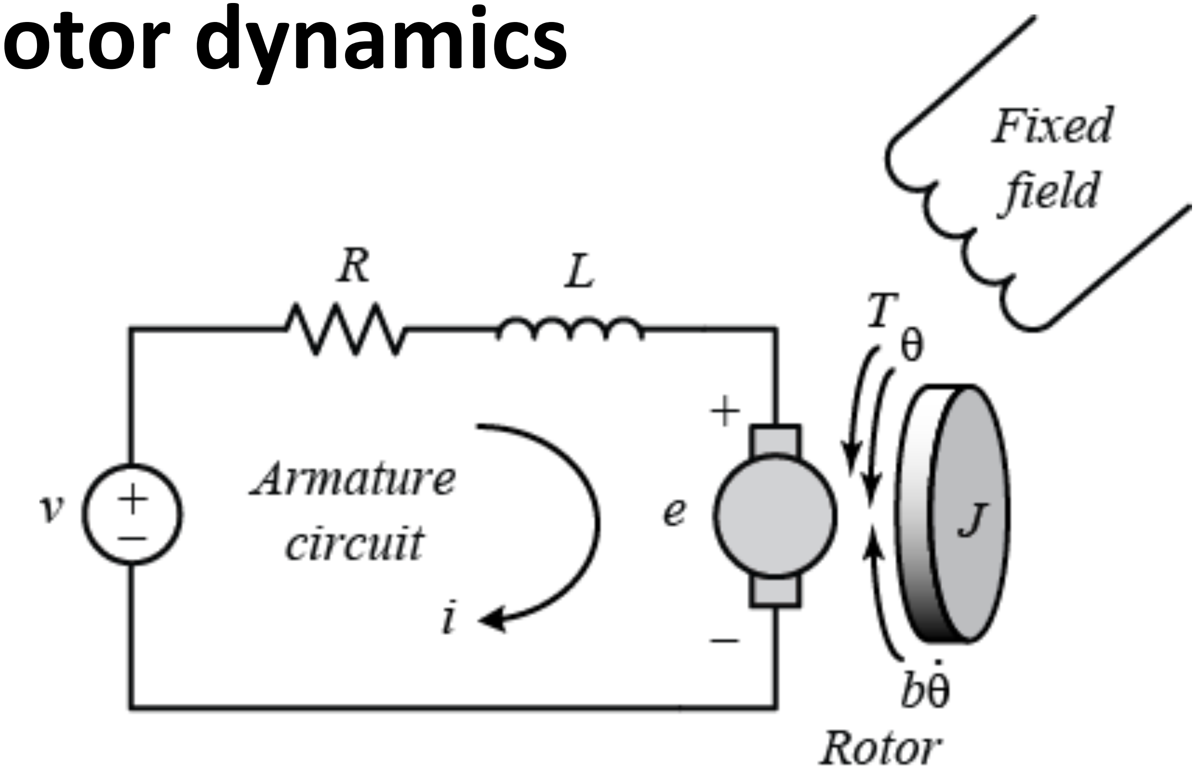
Therefore equating the two terms gives torque equation

$$K_t i(t) = b\frac{d\theta}{dt} + J\frac{d^2\theta}{dt^2}$$

Rearranging torque equation

$$\Rightarrow J\frac{d^2\theta}{dt^2} = -b\frac{d\theta}{dt} + K_t i(t) \quad \Rightarrow \frac{d^2\theta}{dt^2} = -\frac{b}{J}\frac{d\theta}{dt} + \frac{K_t}{J} i(t)$$

DC motor dynamics



Summing voltages around the circuit gives voltage equation

$$v(t) = i(t)R + L\frac{di}{dt} + K_e\frac{d\theta}{dt}$$

$$\Rightarrow L\frac{di}{dt} = -K_e\frac{d\theta}{dt} - i(t)R + v(t)$$

$$\Rightarrow \frac{di}{dt} = -\frac{K_e}{L}\frac{d\theta}{dt} - \frac{R}{L}i(t) + \frac{1}{L}v(t)$$

State space DC motor dynamics

Now writing

$$\frac{d^2\theta}{dt^2} = \frac{d}{dt}\left(\frac{d\theta}{dt}\right) \qquad \text{from} \qquad \frac{d^2\theta}{dt^2} = -\frac{b}{J}\frac{d\theta}{dt} + \frac{K_t}{J}i(t)$$

$$\Rightarrow \frac{d}{dt}\left(\frac{d\theta}{dt}\right) = -\frac{b}{J}\frac{d\theta}{dt} + \frac{K_t}{J}i(t)$$

Now writing

$$\frac{di}{dt} = \frac{d}{dt}\left(i(t)\right) \qquad \text{from} \qquad \frac{di}{dt} = -\frac{K_e}{L}\frac{d\theta}{dt} - \frac{R}{L}i(t) + \frac{1}{L}v(t)$$

$$\Rightarrow \frac{d}{dt}\left(i(t)\right) = -\frac{K_e}{L}\frac{d\theta}{dt} - \frac{R}{L}i(t) + \frac{1}{L}v(t)$$

State space DC motor dynamics

So we have the two equations

$$\frac{d}{dt}\left(\frac{d\theta}{dt}\right)=-\frac{b}{J}\frac{d\theta}{dt}+\frac{K_t}{J}i(t)$$

$$\frac{d}{dt}(i(t))=-\frac{K_e}{L}\frac{d\theta}{dt}-\frac{R}{L}i(t)+\frac{1}{L}v(t)$$

We can write these two equations in matrix form

$$\frac{d}{dt}\begin{bmatrix}\dfrac{d\theta}{dt}\\ i(t)\end{bmatrix}=\begin{bmatrix}-\dfrac{b}{J} & \dfrac{K_t}{J}\\ -\dfrac{K_e}{L} & -\dfrac{R}{L}\end{bmatrix}\begin{bmatrix}\dfrac{d\theta}{dt}\\ i(t)\end{bmatrix}+\begin{bmatrix}0\\ \dfrac{1}{L}\end{bmatrix}v(t)$$

If we want to observe the speed of the motor as the output y

$$y=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\dfrac{d\theta}{dt}\\ t(t)\end{bmatrix}$$

State space DC motor dynamics

As we will later see the two equations describe the motor in state space form

$$\frac{d}{dt}\begin{bmatrix} \dfrac{d\theta}{dt} \\ i(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{b}{J} & \dfrac{K_t}{J} \\ -\dfrac{K_e}{L} & -\dfrac{R}{L} \end{bmatrix}\begin{bmatrix} \dfrac{d\theta}{dt} \\ i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{1}{L} \end{bmatrix} v(t) \qquad y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \dfrac{d\theta}{dt} \\ t(t) \end{bmatrix}$$

By observation of standard state space form

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

We see that

$$A = \begin{bmatrix} -\dfrac{b}{J} & \dfrac{K_t}{J} \\ -\dfrac{K_e}{L} & -\dfrac{R}{L} \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ \dfrac{1}{L} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad D = 0$$

ROCO218: Control Engineering
Dr Ian Howard

Lecture 2

Numerical solutions of 1st order differential equation

Differential equations

Ordinary differential equation (ODE)

- contains one or more functions of one independent variable and its derivatives
- For example

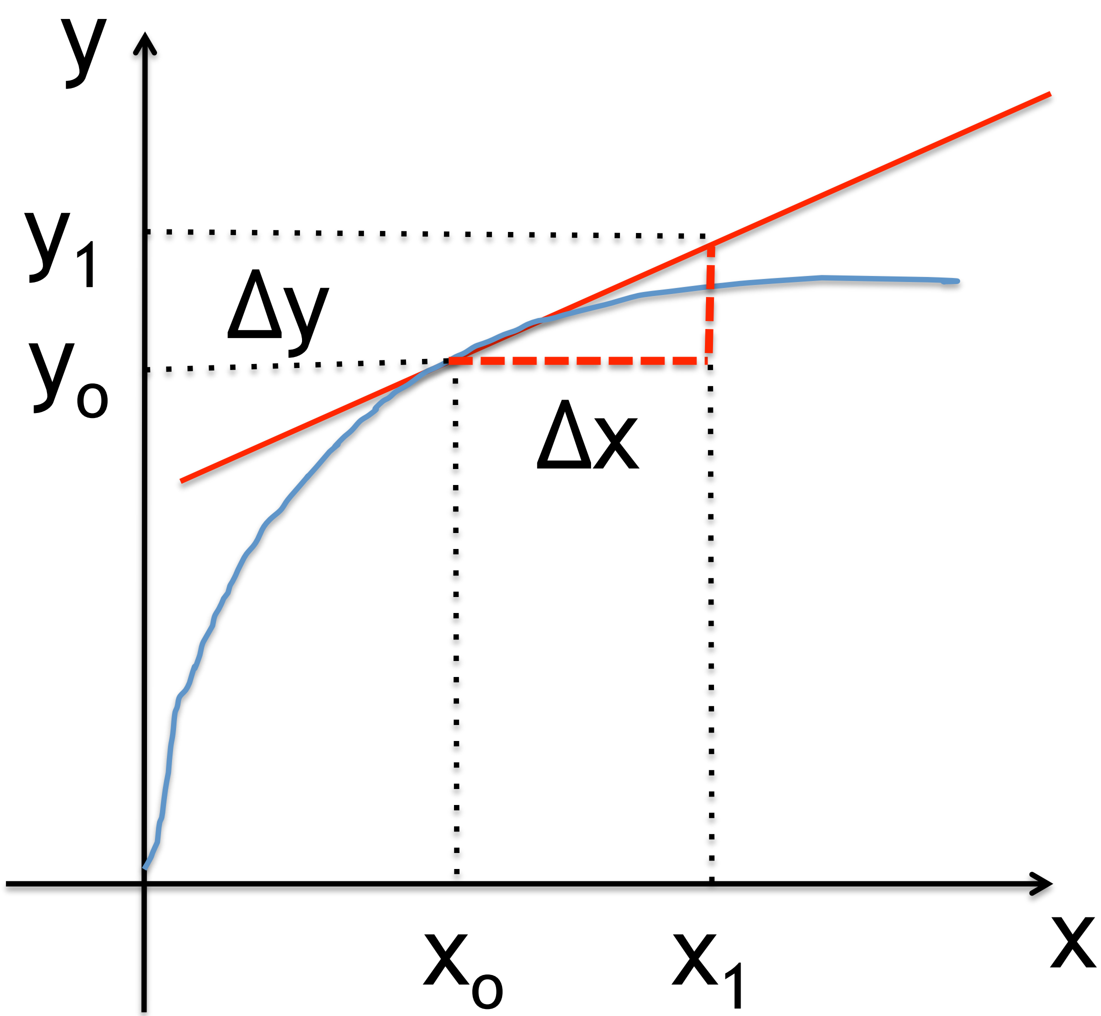
$$\frac{\partial u}{\partial x}(x, y) = 0.$$

Partial differential equation

- may compare more than one independent variable
- contains unknown multivariable functions and their partial derivatives.
- For example

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Local linear approximation of gradient



Consider a function $y=f(x)$

When $x=x_0$ then $y=y_0$

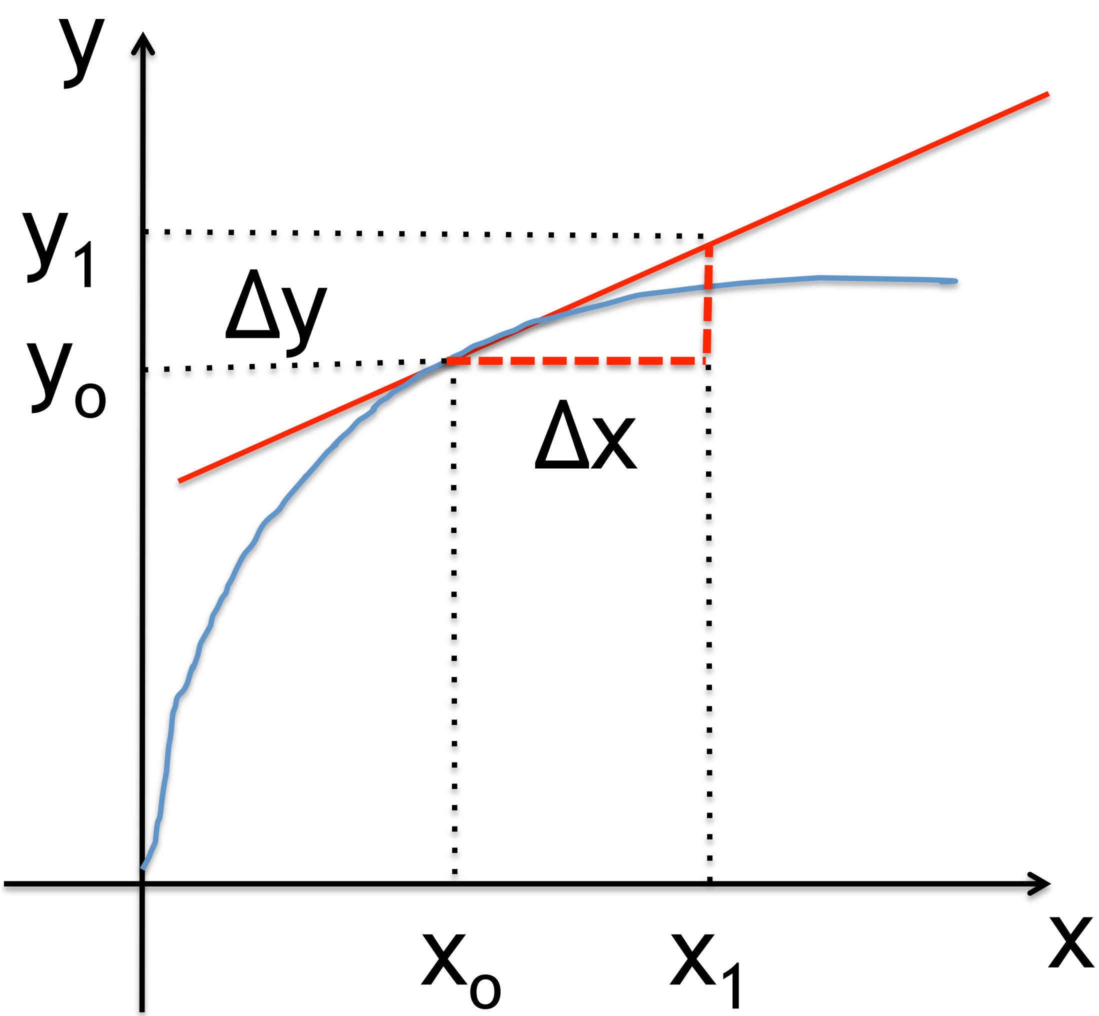
As we increase the value of x by Δx we reach a point where $x_1=x_o+\Delta x$
Similarly this increases y by Δy reaching the value $y_1=y_o+\Delta y$

Thus

$$\left(x_1, y_1\right) = \left(x_0 + \Delta x, y_0 + \Delta y\right)$$

The gradient of the curve at (x_o, y_o) is the tangent to the curve at this point

Local linear approximation of gradient



In general the differential of a function is given by

$$f'(x) = \lim_{dx \to 0} \frac{f(x+dx) - f(x)}{dx}$$

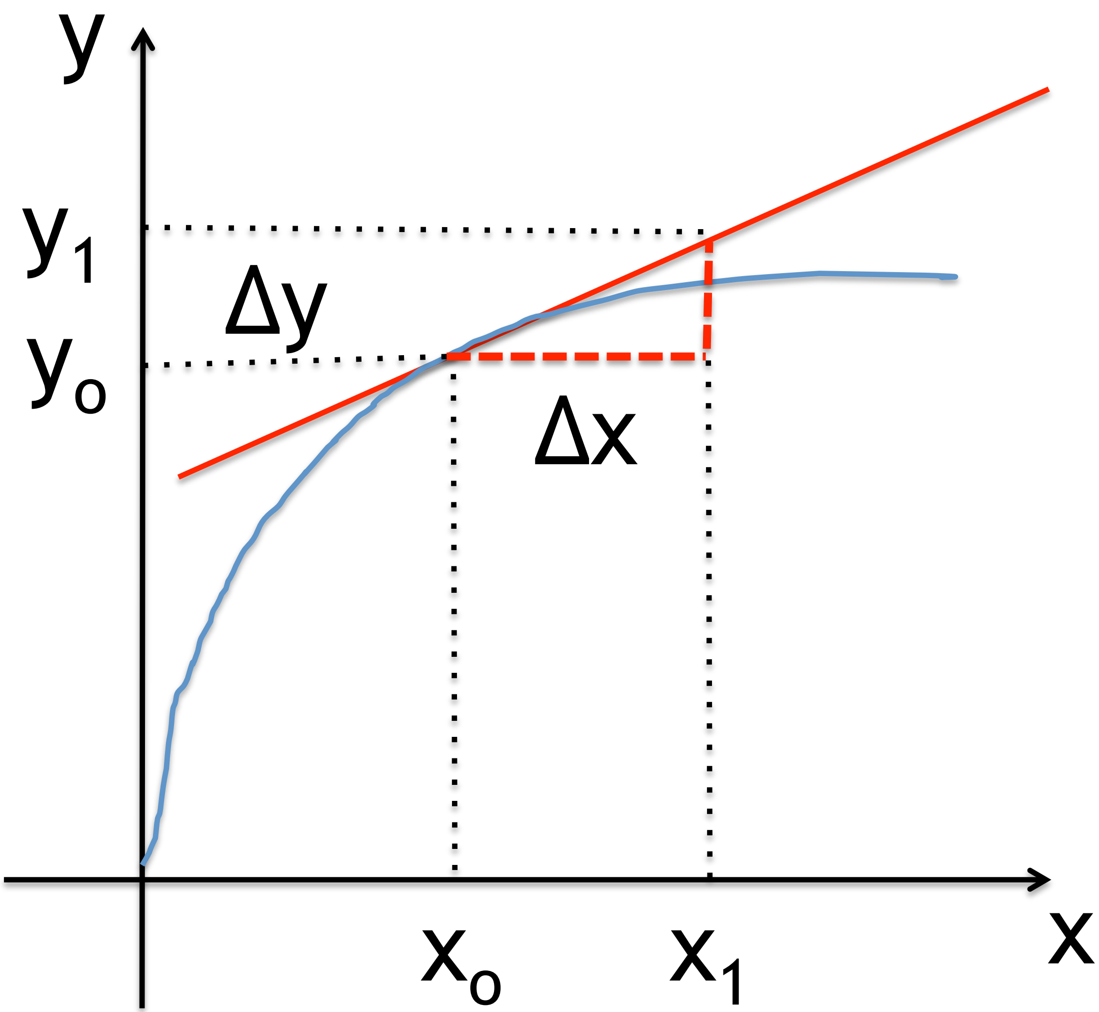
Which is only strictly true in the limit where dx tends to zero
In our case

$$\frac{dy}{dx} \approx \frac{\Delta y}{\Delta x}$$

So the gradient at (x_o, y_o) can be approximated by

$$\left.\frac{dy}{dx}\right|\left(x_0, y_0\right) \approx \frac{\Delta y}{\Delta x}$$

Local linear approximation of gradient



We can use the relationship to iteratively estimate y_1 from y_o since

$$\left.\frac{dy}{dx}\right|(x_0,y_0) \approx \frac{\Delta y}{\Delta x} \approx \frac{(y_1 - y_0)}{\Delta x}$$

Writing Δx as step size h

$$\Rightarrow y_1 - y_0 = h.\left.\frac{dy}{dx}\right|(x_0,y_0)$$

$$\Rightarrow y_1 = y_0 + h.\left.\frac{dy}{dx}\right|(x_0,y_0)$$

We can use this to give next estimate of y_1 if we know x_o and the gradient at $(x_o\, y_o)$

Euler's method of integration

- This is Euler's method to solve a 1st order differential equation!
- It uses a local linear approximation to the gradient
- We can iteratively estimate the value of the function at the next step
- Given the relationship

$$y' = f(t, y), \;\; y(t_0) = y_0$$

$$y_{n+1} = y_n + h.f\left(t_n, y_n\right)$$

Where
t is time
y is the output value of the function
At initial time t_o the output is y_o
At future time $t_{(n+1)}$ the output is $y_{(n+1)}$
h is the step size

- Notice the simulation error will rise as the step size increases!

Remember: Voltages in an RC circuit

$$V_S = V_R + V_C$$

Adding up voltages

$$V_R = IR$$

Ohms law

$$I = \frac{dQ}{dt} = C\frac{dV_C}{dt}$$

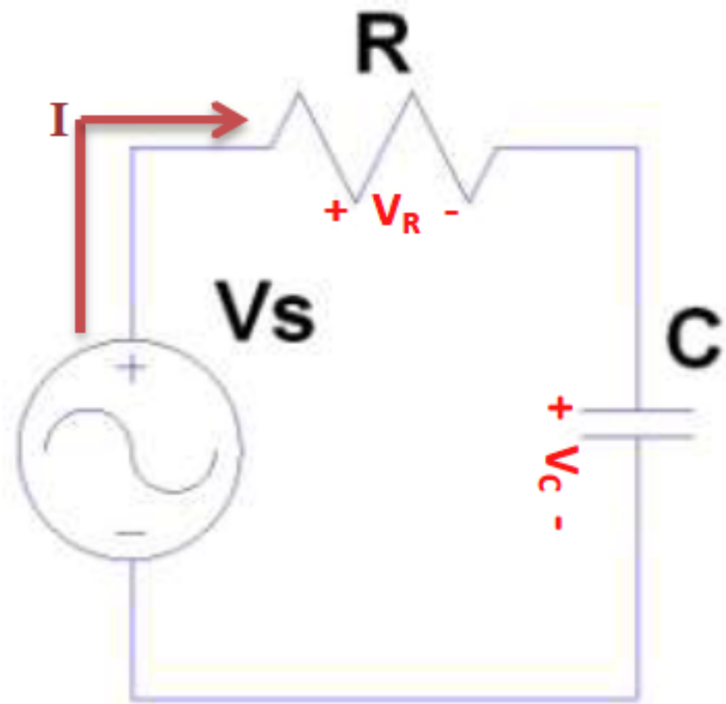
Current proportional to rate of change of charge

$$V_S = RC\frac{dV_C}{dt} + V_C$$

Substitute I from capacitor expression

$$\frac{dV_C}{dt} = \frac{1}{RC}\left(V_S - V_C\right)$$

Rearranged



This is a first order linear differential equation

Simulation of voltages in an RC circuit

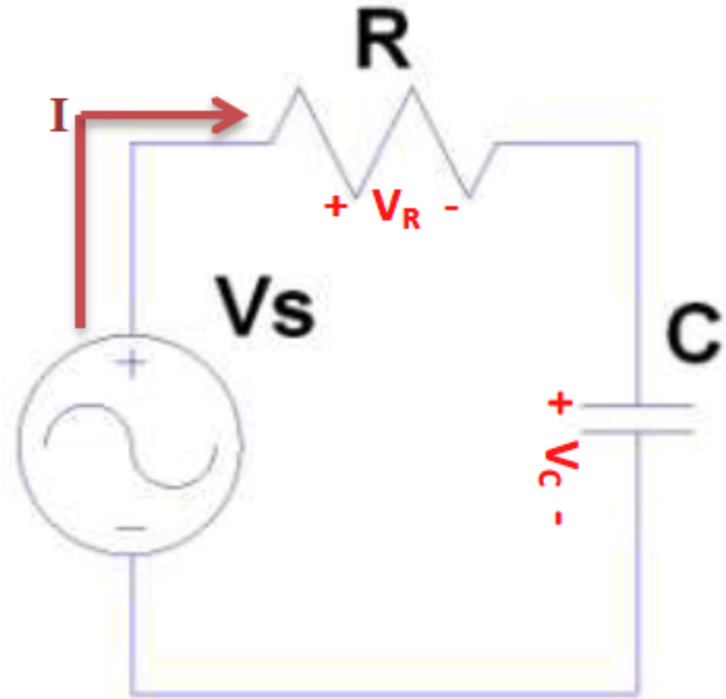
Here we wish to compute the evolution of V_c with time. We know that

$$\frac{dV_C}{dt} = \frac{1}{RC}\left(V_S - V_C\right)$$

From the iterative relationship we derived previously

$$y_1 = y_0 + h.\left.\frac{dy}{dx}\right|_{(x_0,y_0)}$$

$$\Rightarrow V_C(t+1) = V_C(t) + h\frac{1}{RC}\left(V_S - V_C(t)\right)$$



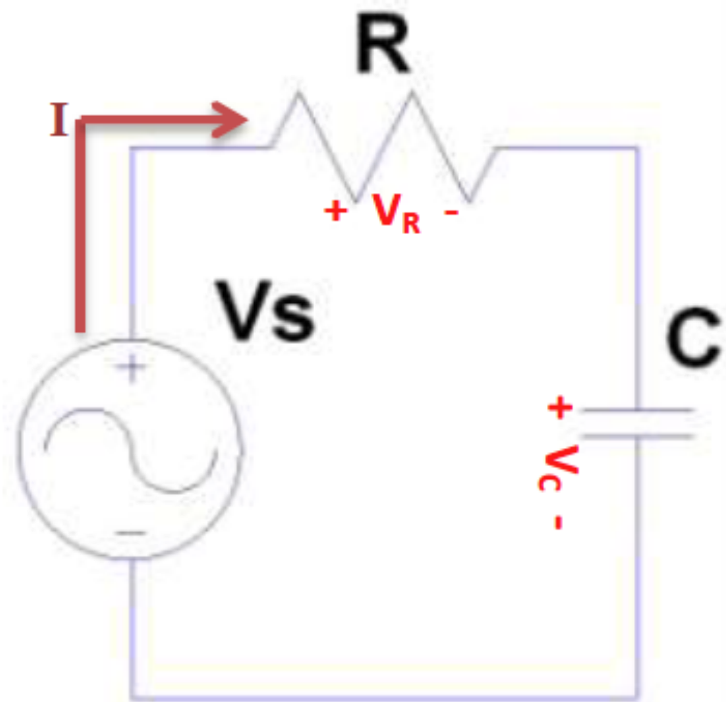
- One of your lab assignments is to implement this simulation in Matlab!
- Matlab also has more sophisticated numerical methods to solve differential equations- e.g. ode45

Implementing Euler integration in Matlab

We can write simple script using the update equation to estimate capacitor voltage is given by the equation

$$\Rightarrow V_C(t+1) = V_C(t) + h\frac{1}{RC}\left(V_S - V_C(t)\right)$$

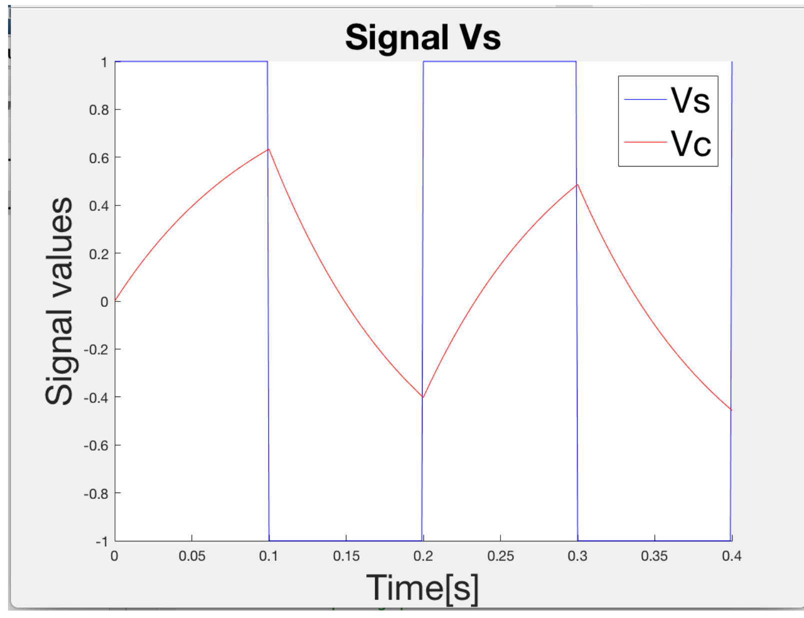
```
function Vc = GetCapVoltage(R, C, Vs, h)
% calling function to get Vc
% given resistance R,capacitance C, driving voltage Vs
% and time step h

% got input sample count
len = length(Vs)-1;

% set initial votage to zero
Vc(1) = 0;

% loop over all samples -1
for n = 1:len

    % Euler update equation
    Vc(n+1) = Vc(n) + h * (1/(R*C)) * (Vs(n) - Vc(n));
end

end
```



Implementing Euler integration in Matlab

```
% reset the workspace and display
close all
clear all
clc

% parameters
timeLen = 0.4; % time duration of seconds
samplingRate = 1000; %sampling frequency in Hz
freq = 5; % signal frequency in  Hz

% resistance 1000 ohms
R = 1000;
% capacitor value 100 microFarads
C = 100e-6;
% step in secoinds
h = 1/samplingRate;
% time sampling points
T = 0:h:timeLen;

% number of dataseamples
sampleCnt = length(T);

% build square wave driving voltage
Vs = square(2 * pi * freq * T);

% calling function to get Vc
Vc = GetCapVoltage(R, C, Vs, h);
```

```
% plot the figure
figure
% want all plots to remain on figure
hold on
% write a title
h = title('Signal Vs');
set(h, 'FontSize', 25)
% plot driving voltage Vs
plot(T, Vs, 'b');
% plot capacitor voltage Vx
plot(T, Vc, 'r');
% label plots
h=legend('Vs','Vc');
set(h, 'FontSize', 25)
% label the axes
h= xlabel('Time[s]')
set(h, 'FontSize', 25)
h = ylabel('Signal values');
set(h, 'FontSize', 25)
```



Using ode45 to perform integration

help ode45

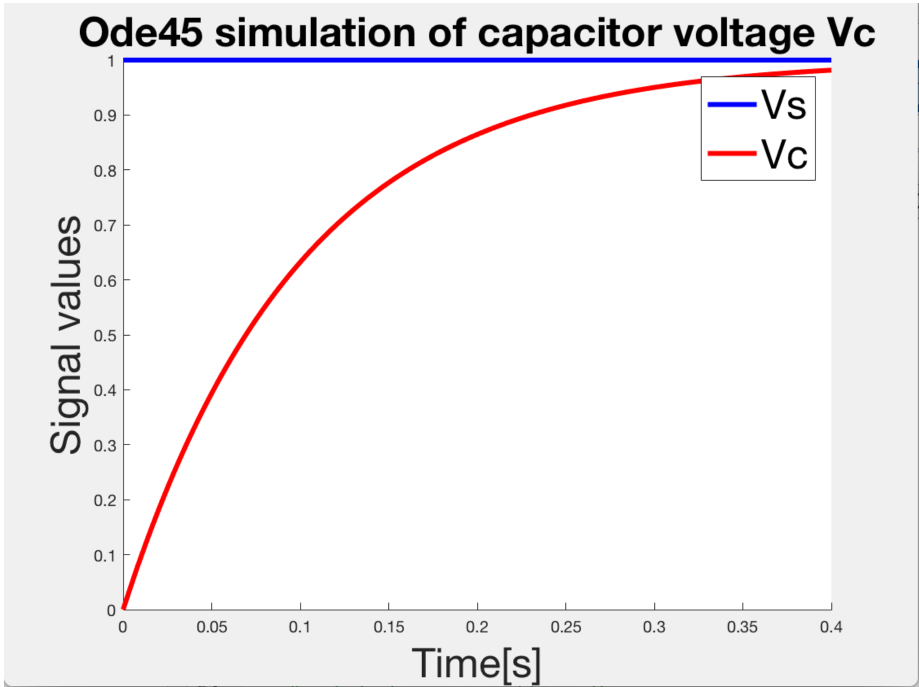
- Solve non-stiff differential equations, medium order method.
- [TOUT,YOUT] = ode45(ODEFUN,TSPAN,Y0)
- with TSPAN = [T0 TFINAL] integrates the system of differential equations y' = f(t,y) from time T0 to TFINAL with initial conditions Y0.
- ODEFUN is a function handle.
- For a scalar T and a vector Y, ODEFUN(T,Y) must return a column vector corresponding to f(t,y).
- Each row in the solution array YOUT corresponds to a time returned in the column vector TOUT.
- To obtain solutions at specific times T0,T1,...,TFINAL (all increasing or all decreasing), use TSPAN = [T0 T1 ... TFINAL].

Using ode45 to perform integration

```
function VcDot = RCDE(Vc, R, C, Vs)
% Vc is capatotor voltage
% R resistance
% C capacitance
% Vs input control applied voltage
% xdot is the returned 1st time derivative of capacitor voltage

% differential equation for capacitor voltage
VcDot = (Vs - Vc)/(R*C);
```

```
% parameters
timeLen = 0.4; % time duration of seconds
samplingRate = 1000; %sampling frequency in Hz
freq = 5; % signal frequency in  Hz

% resistance 1000 ohms
R = 1000;
% capacitor value 100 microFarads
C = 100e-6;
% step in secoinds
h = 1/samplingRate;
% time sampling points
T = 0:h:timeLen;

% number of data samples
sampleCnt = length(T);

% build target unity driving voltage
Vs = ones(size(T));

% initial conditions - no capacotr votlage
V0 = 0;

% run simulation
[tSS, Vc] = ode45(@(t,x)RCDE(x, R, C,  Vs(1)), T, V0);
```

```
% plot the figure
figure
% want all plots to remain on figure
hold on
% write a title
h = title('Ode45 simulation of capacitor voltage Vc');
set(h, 'FontSize', 25)
% plot driving voltage Vs
h = plot(T, Vs, 'b');
set(h,'LineWidth', 3);
% plot capacitor voltage Vx
h = plot(T, Vc, 'r');
set(h,'LineWidth', 3);
% label plots
h=legend('Vs','Vc');
set(h, 'FontSize', 25)
% label the axes
h= xlabel('Time[s]')
set(h, 'FontSize', 25)
h = ylabel('Signal values');
set(h, 'FontSize', 25)
```



Interlude

10 minute break

ROCO218: Control Engineering

Lecture 2

Introduction to Simulink

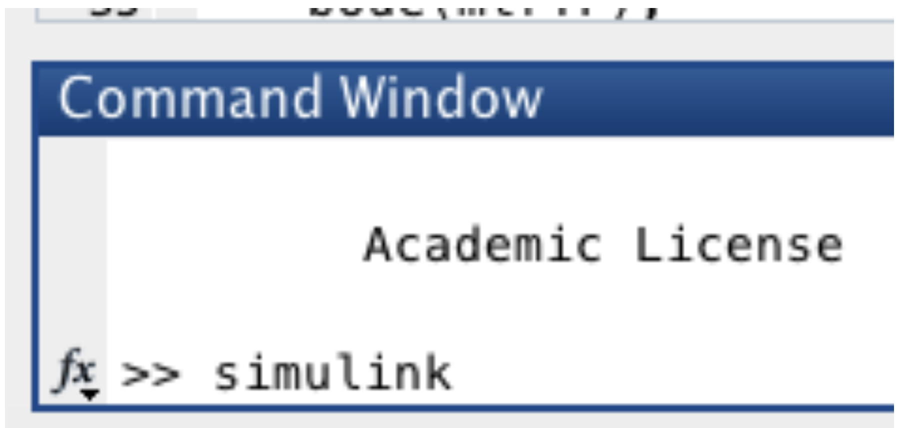
Simulink

- Used to model, analyze and simulate dynamic systems using block diagrams.
- Provides a graphical user interface for constructing block diagram of a system – therefore is easy to use.

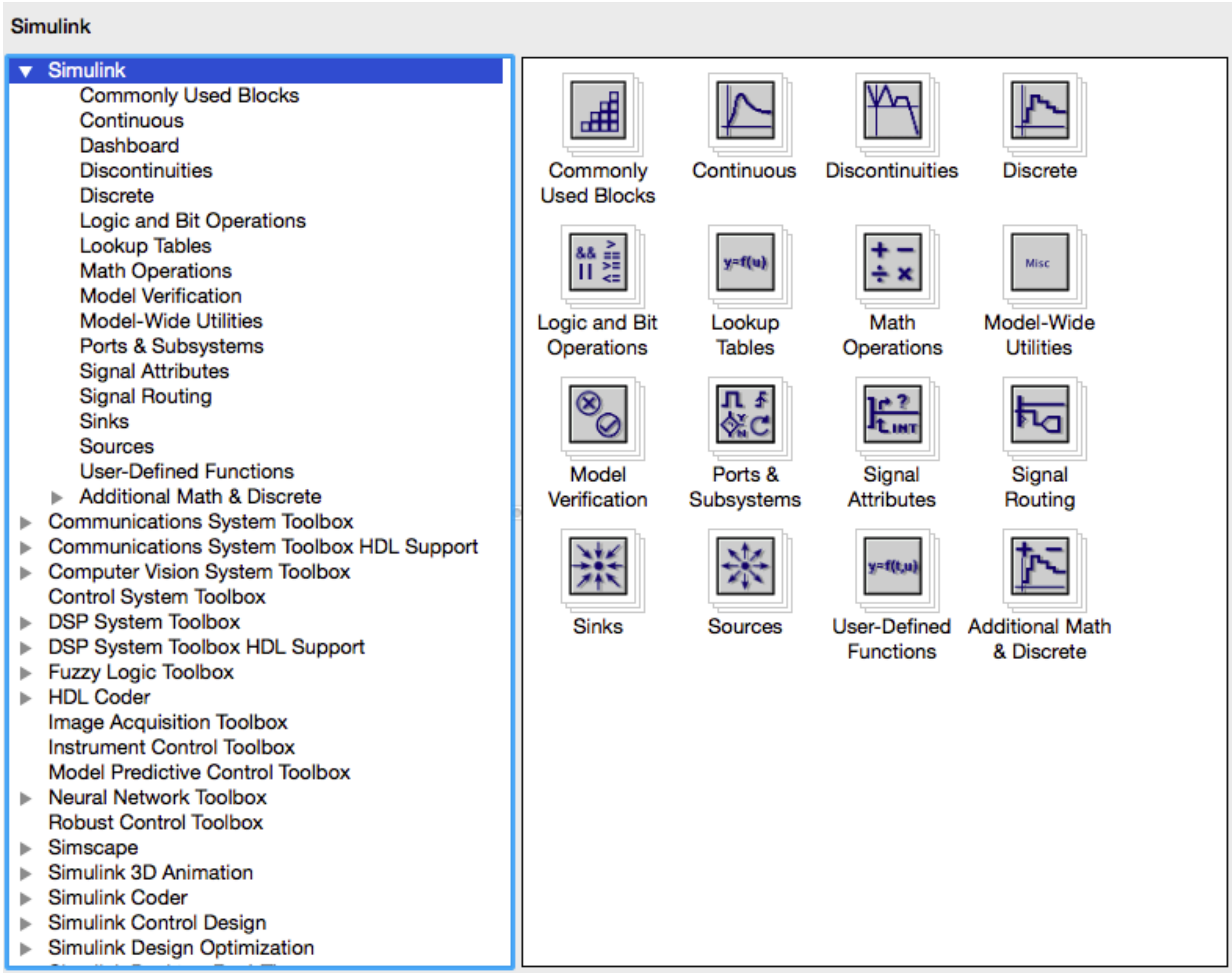
- Can be used for simulation of various systems:
  - Linear, nonlinear
- Input signals can be arbitrarily generated:
  - Standard: sinusoidal, polynomial, square, impulse
  - Customized: from a function, look-up table
- Output signals can be stored or demonstrated in different ways.
  - However modeling a system is not necessarily easy !

Launching Simulink

- First launch Matlab
- Then in the Matlab command window launch slimulink
- Do this at the >> prompt by typing **simulink**
- and press ↵Enter

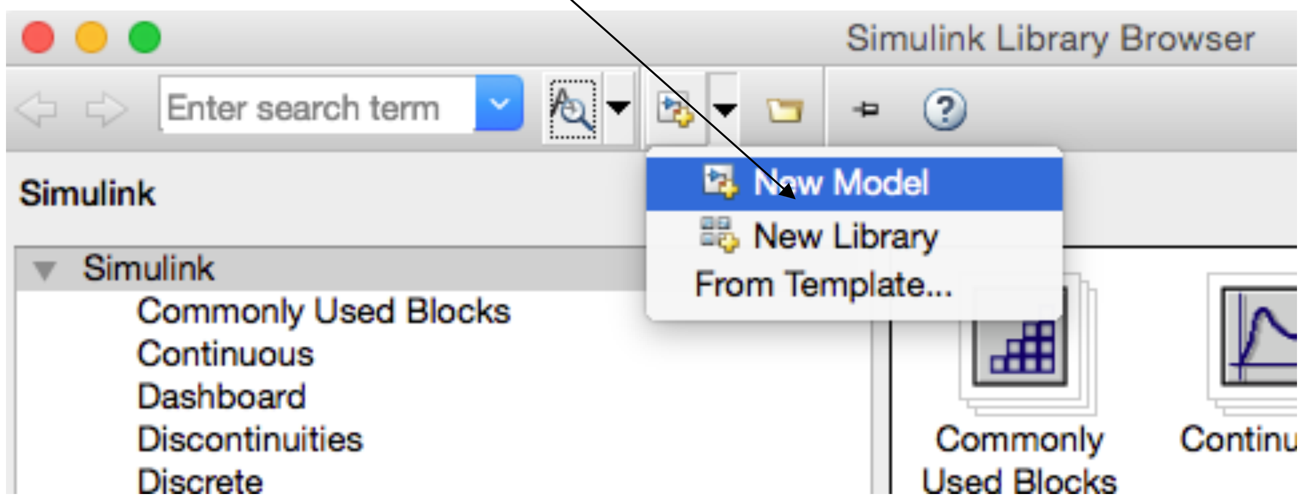


Simulink window



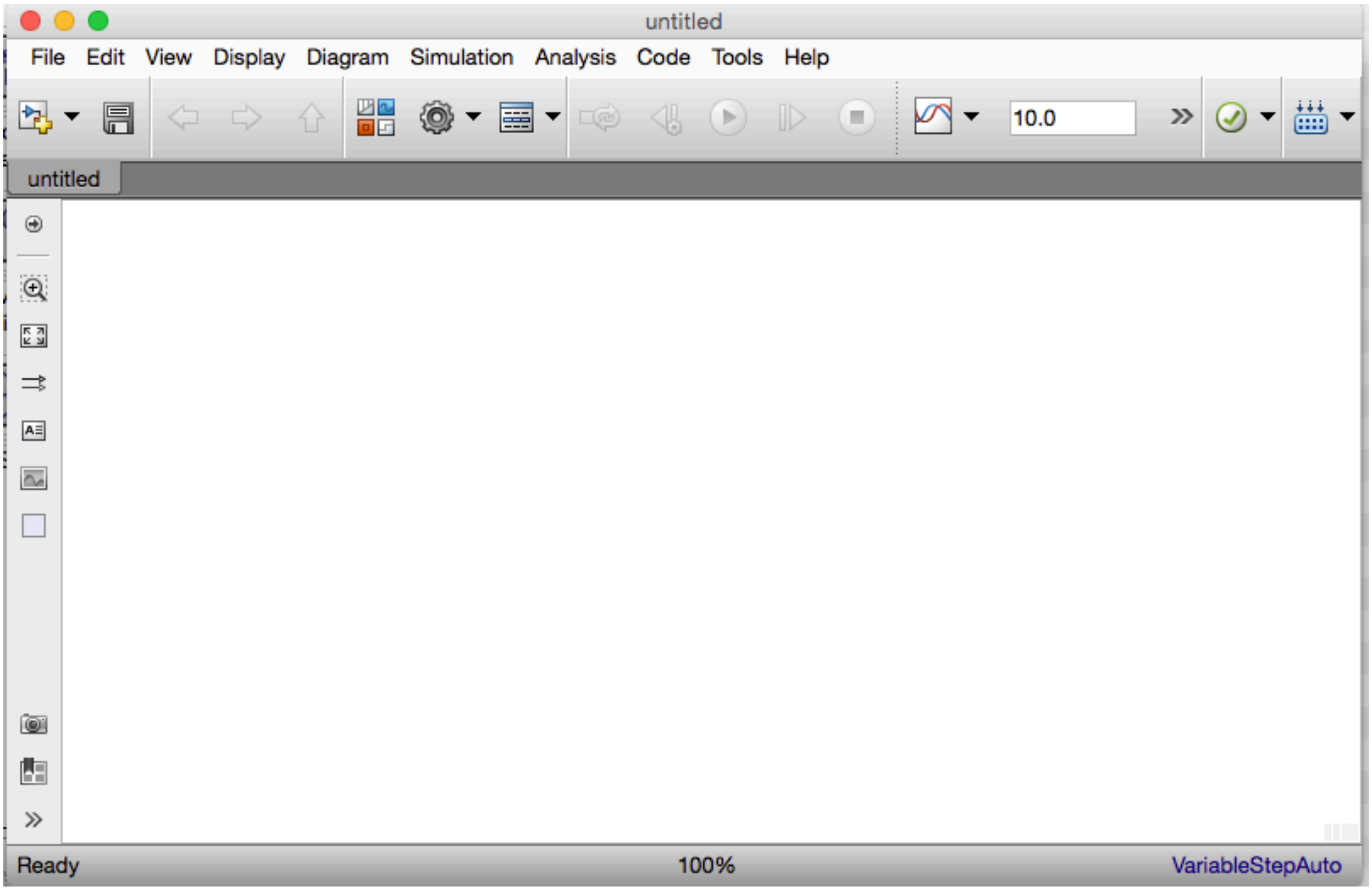
Create a new model

- Next click the new-model icon in the upper left corner to start a new Simulink file



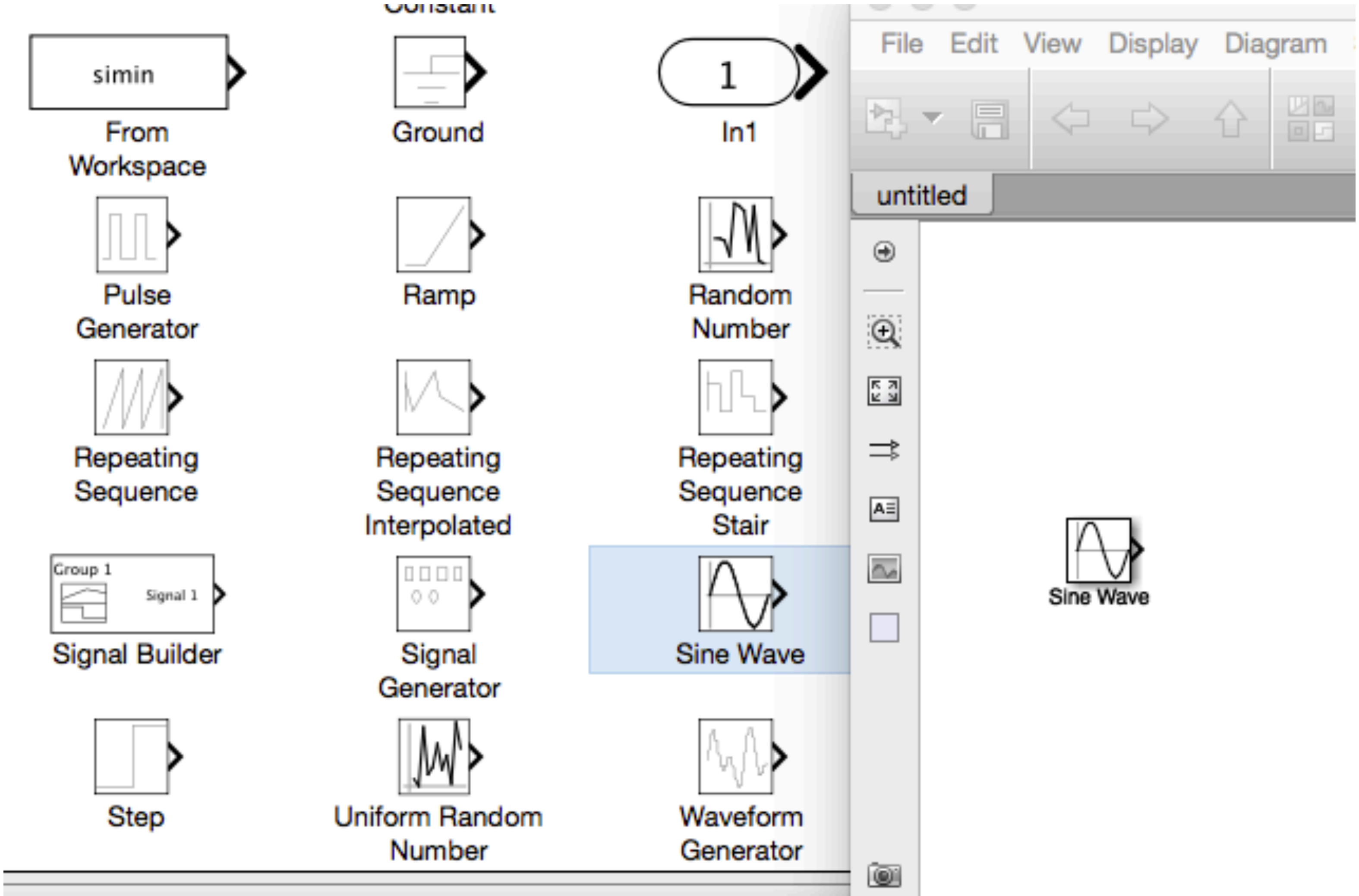
Your workspace

- This opens a window



Add components

- Drag selected components into workspace

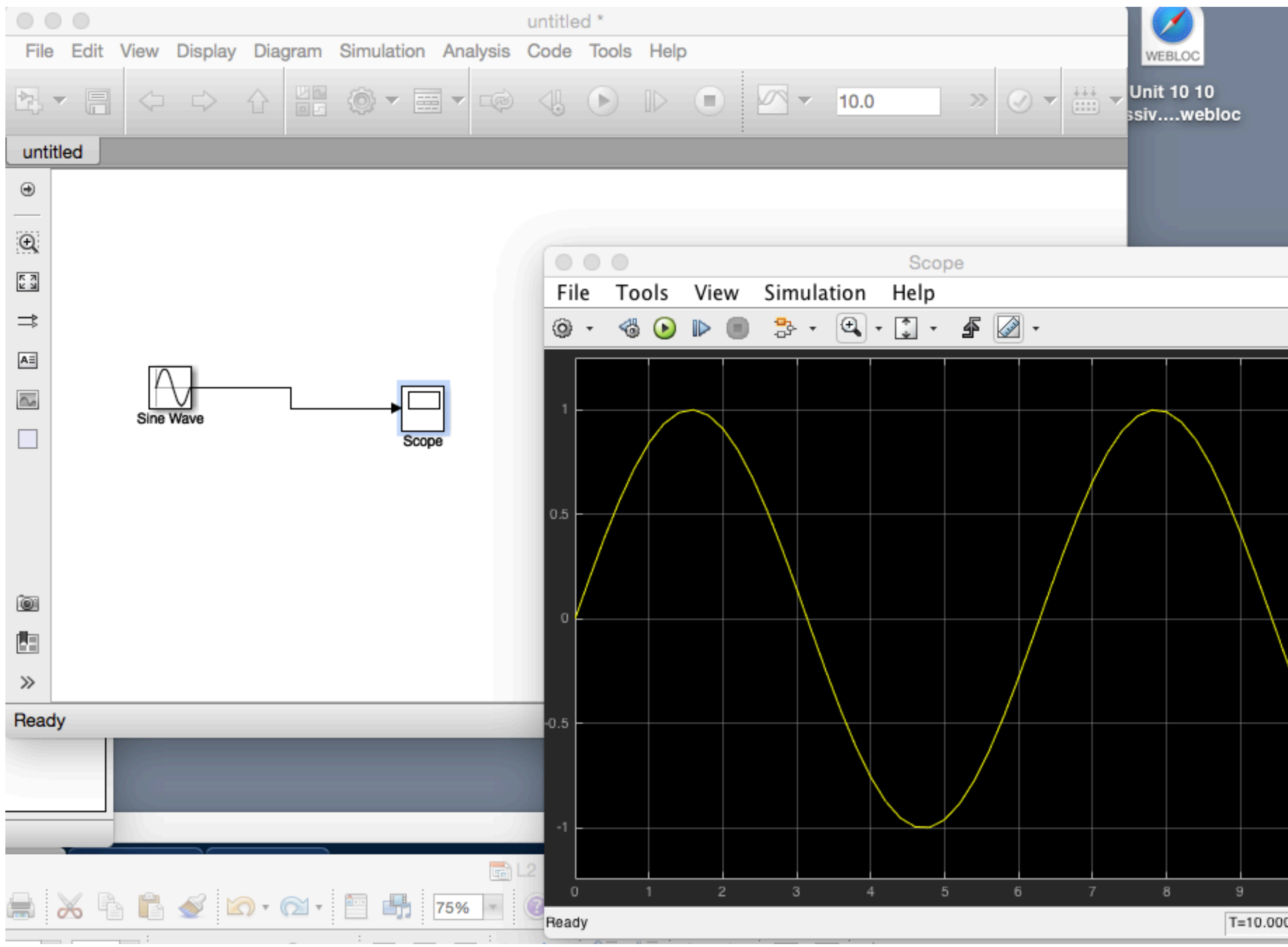


Connect components and run model

- Connect components

- Click run button

- Click on scope to open display

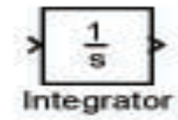


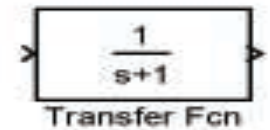
Available components blocks

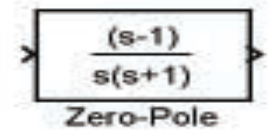
Main components exhibit dynamics:

- integrators,
- transfer function
- zero-pole description

Parameters can be set for various components





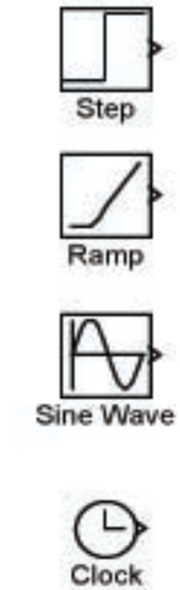


Also math components:

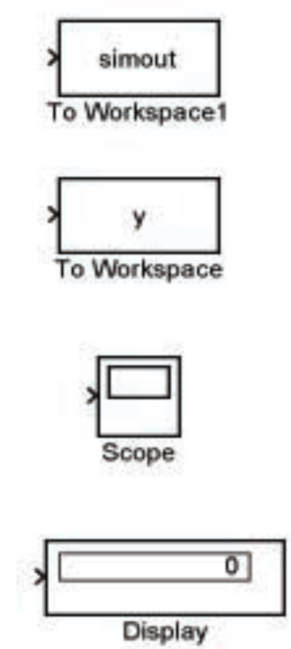
- gain (amplifier) kx : x a scalar
- addition (a+b+c)
- product (abc)
- number of i/o points and the sign of each are adjustable

Many components blocks available

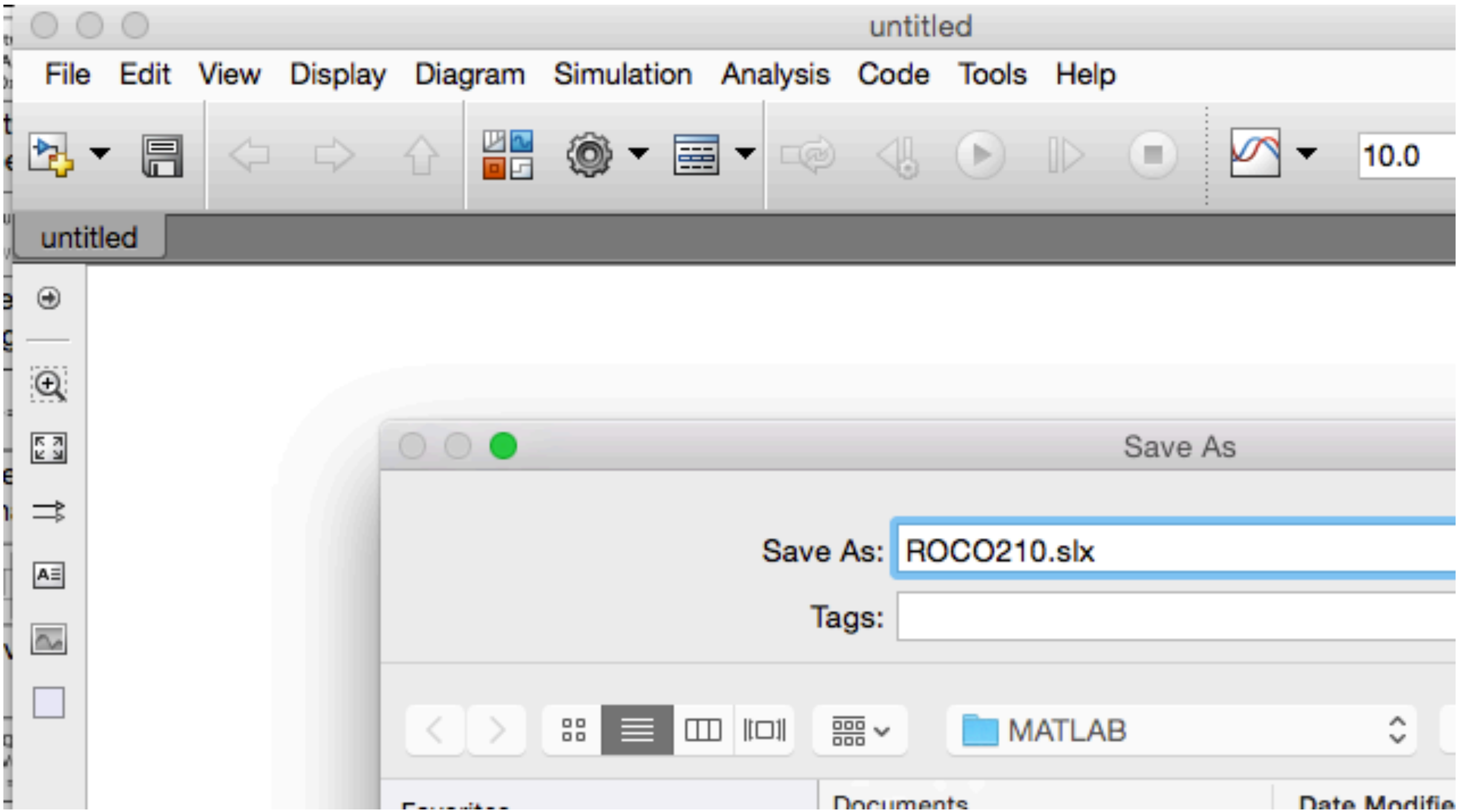
Sources: input signals
- constant, step, ramp
- pulse, sine wave, square wave
- from data file
- signal generator
- a clock to record time



Sinks: for output display or storage
- export to workspace; you can give a name to the variable, such as u, y, x, etc.
- scope
- digital display



Save your model



ROCO218: Control Engineering
Dr Ian Howard

Lecture 1

Simulink 2^{nd} order differential equation

Mass-spring-damper system

- Build a Simulink model that solves the following differential equation
  - 2nd-order mass-spring-damper system
  - zero initial conditions
  - input $f(t)$ is a step with magnitude 3
  - parameters: $m = 0.25$, $c = 0.5$, $k = 1$

$$m\ddot{x} + c\dot{x} + kx = f(t)$$

Create the simulation diagram

The simulation diagram for solving the ODE is created step by step.

After each step, elements are added to the Simulink model.

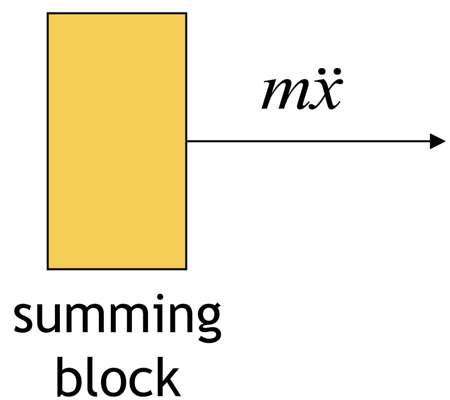
$$m\ddot{x} + c\dot{x} + kx = f(t)$$

Create the simulation diagram

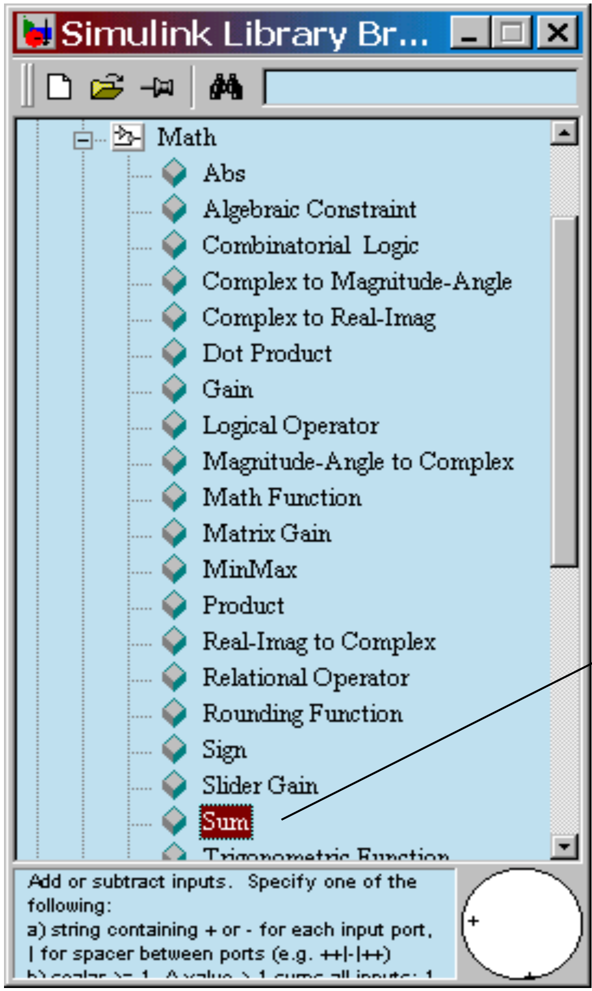
- First, solve for the term with highest-order derivative

$$m\ddot{x} = f(t) - c\dot{x} - kx$$

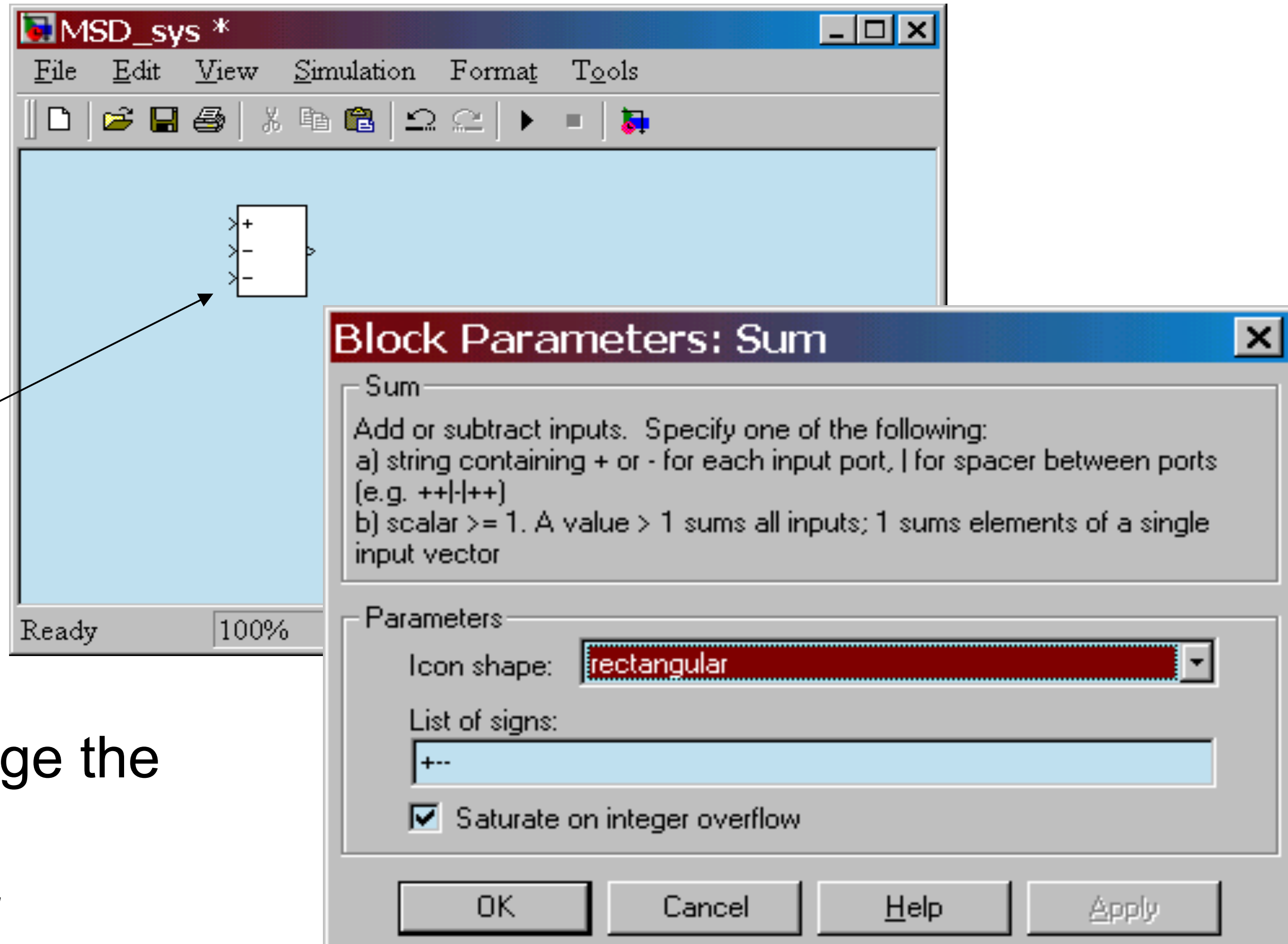
- Make the left-hand side of this equation the output of a summing block



Create the simulation diagram



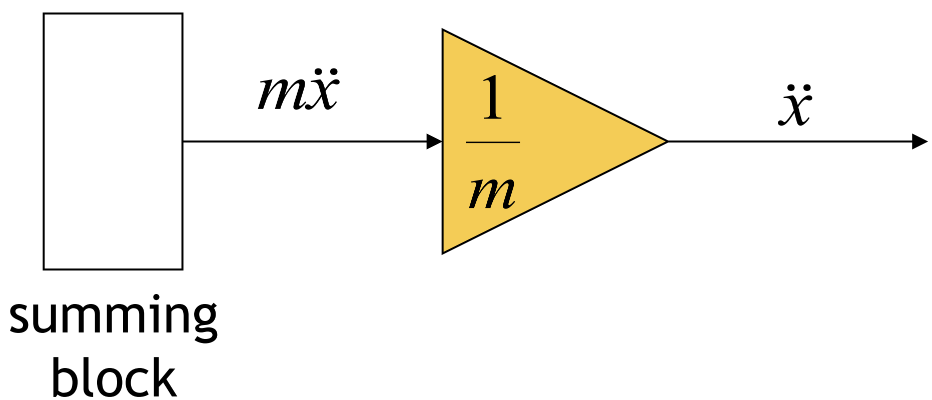
Drag a *Sum* block from the *Math* library



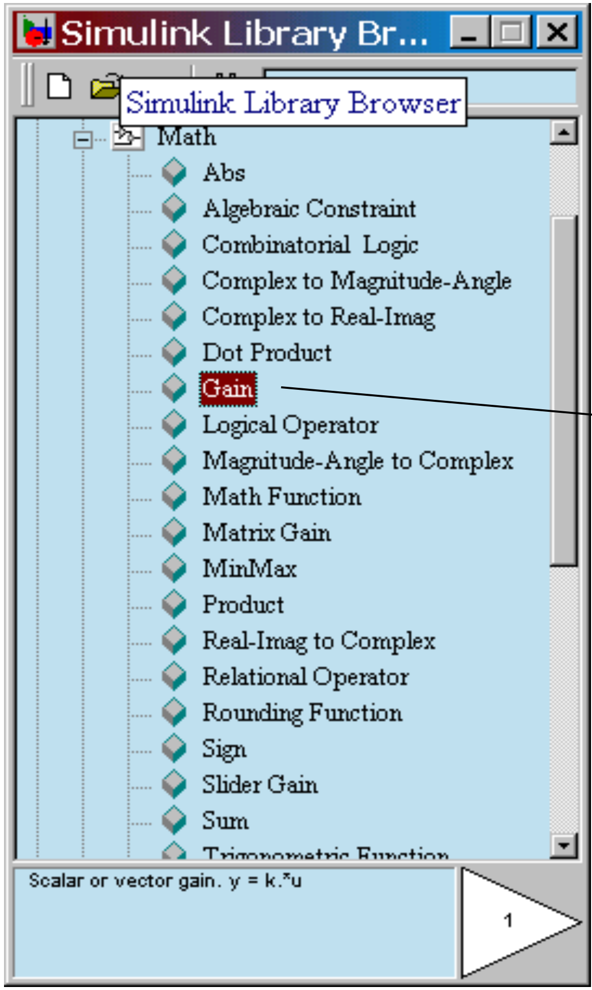
Double-click to change the block parameters to *rectangular* and + - -

Create the simulation diagram

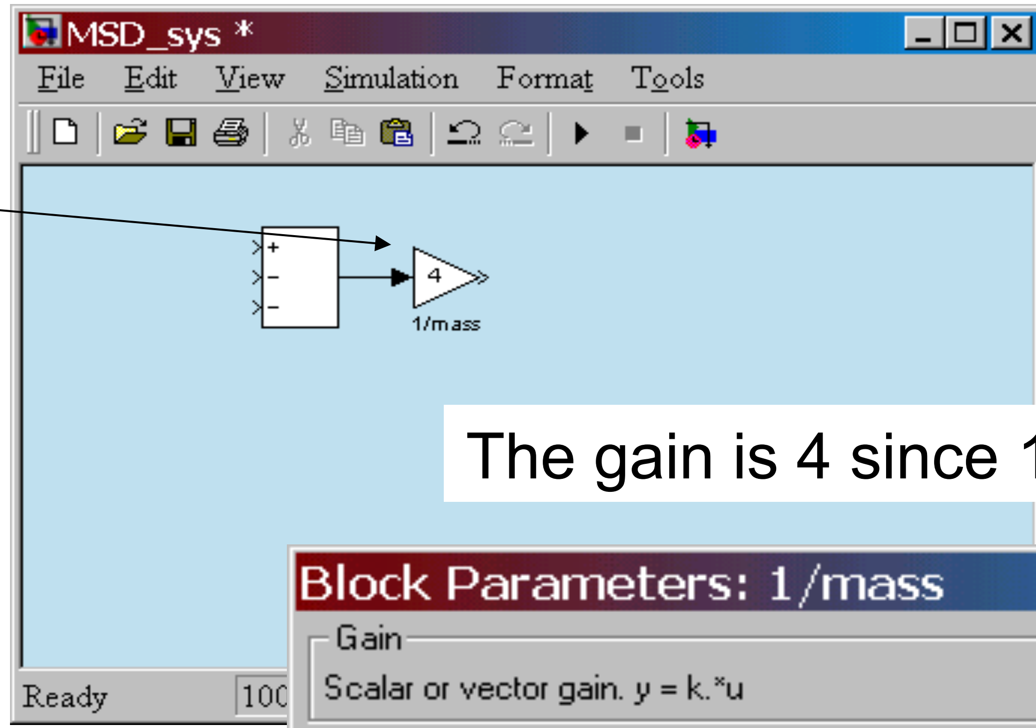
- Add a gain (multiplier) block to eliminate the coefficient and produce the highest-derivative alone



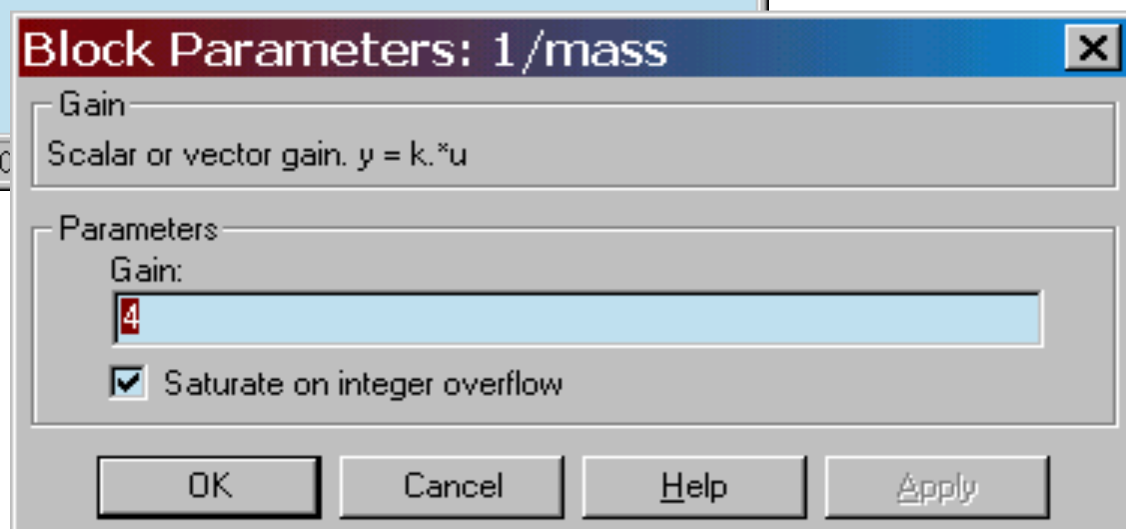
Create the simulation diagram



Drag a *Gain* block from the *Math* library



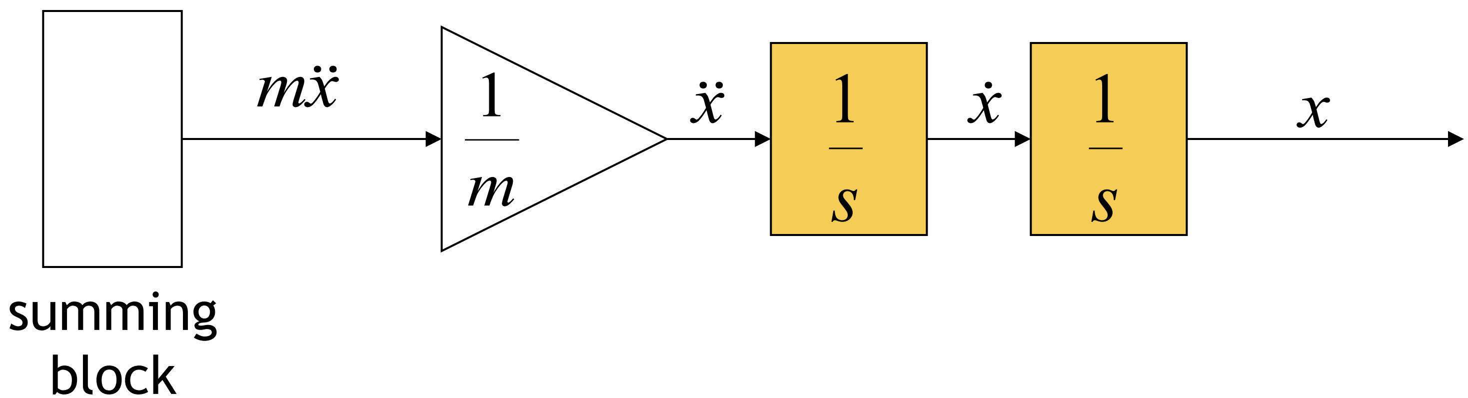
The gain is 4 since 1/m=4.



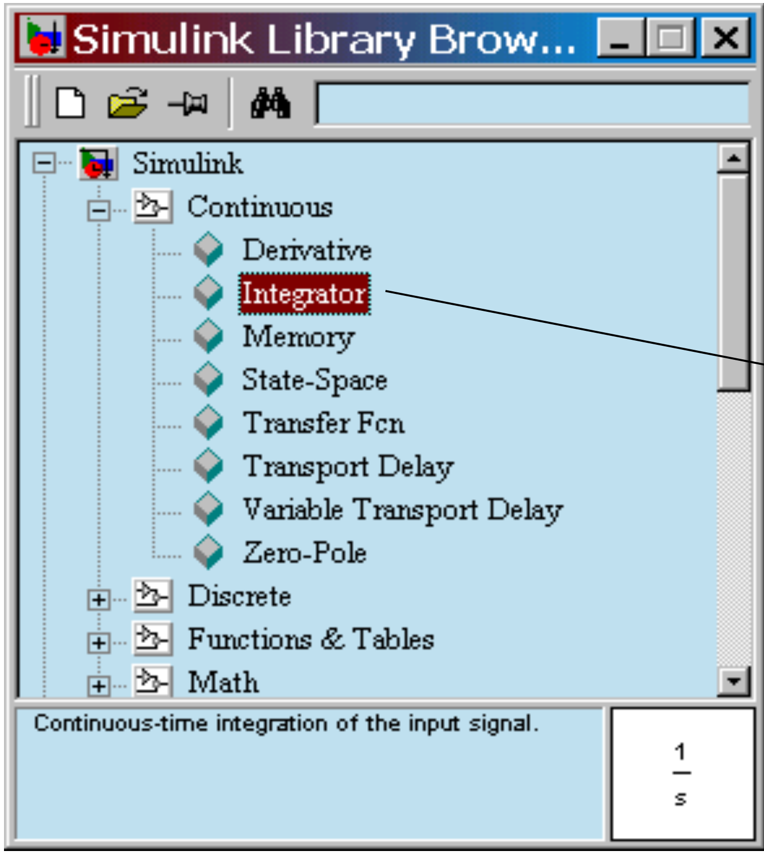
Double-click to change the block parameters.
Add a title.

Create the simulation diagram

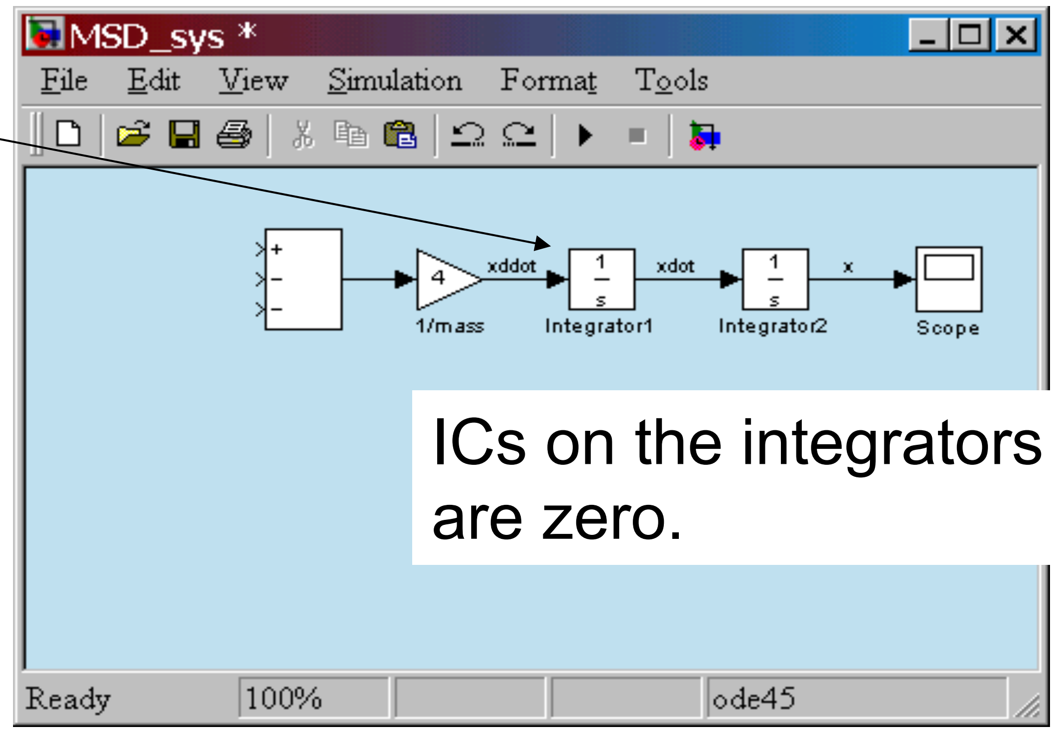
- Add integrators to obtain the desired output variable



Create the simulation diagram



Drag *Integrator* blocks from the *Continuous* library



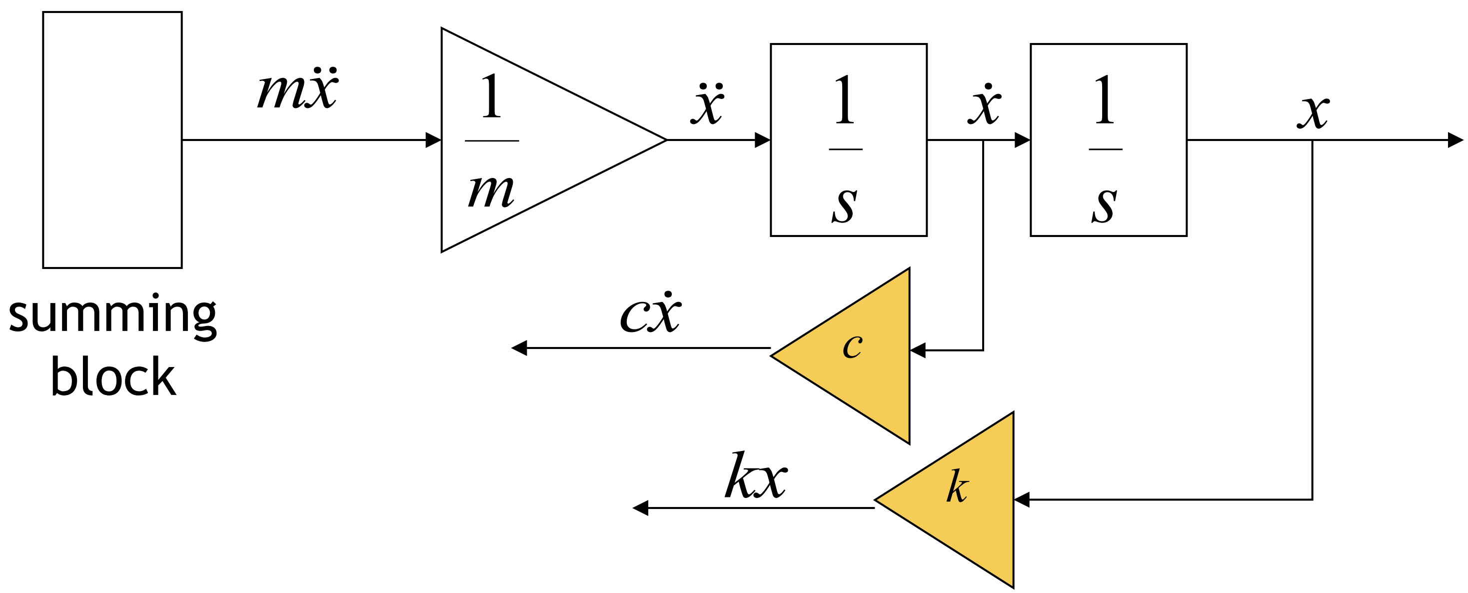
ICs on the integrators are zero.

Add a scope from the *Sinks* library.
Connect output ports to input ports
Label the signals by double-clicking on the leader line

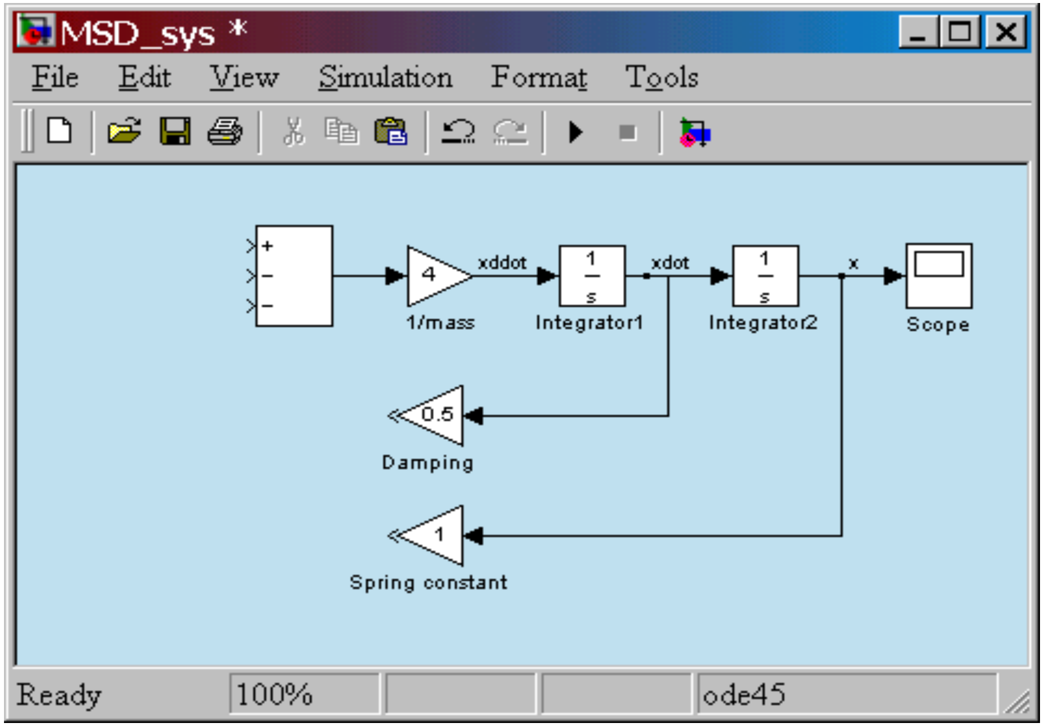
Create the simulation diagram

$$m\ddot{x} = f(t) - c\dot{x} - kx$$

- Connect to the integrated signals with gain blocks to create the terms on the right-hand side of the equation of motion

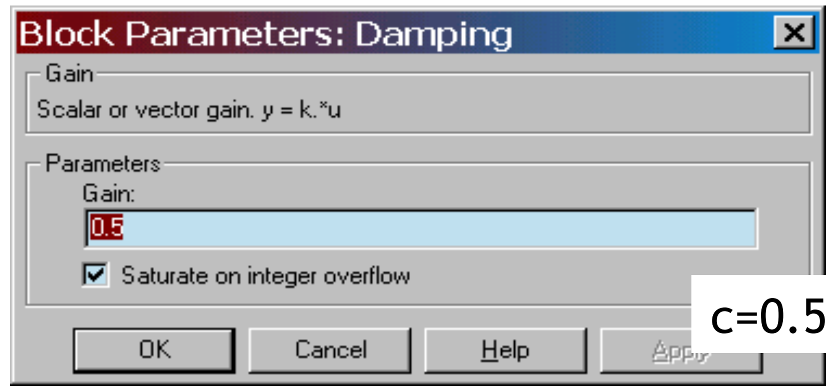


Create the simulation diagram



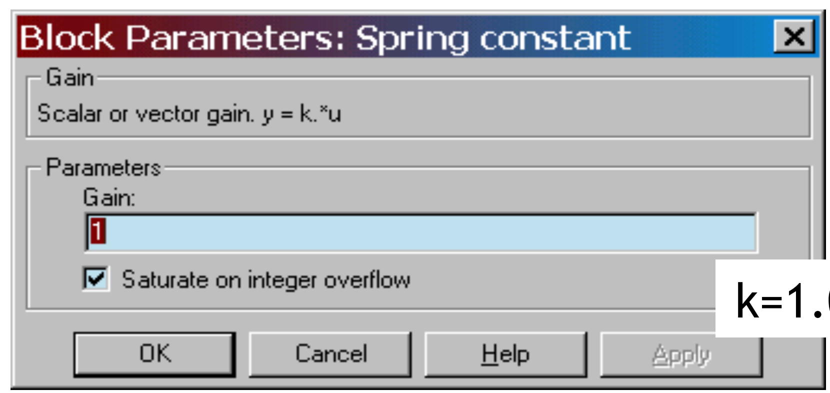
Drag new *Gain* blocks from the *Math* library

To flip the gain block, select it and choose *Flip Block* in the *Format* pull-down menu.



c=0.5

- Double-click on gain blocks to set parameters
- Connect from the gain block input backwards up to the branch point.
- Re-title the gain blocks.

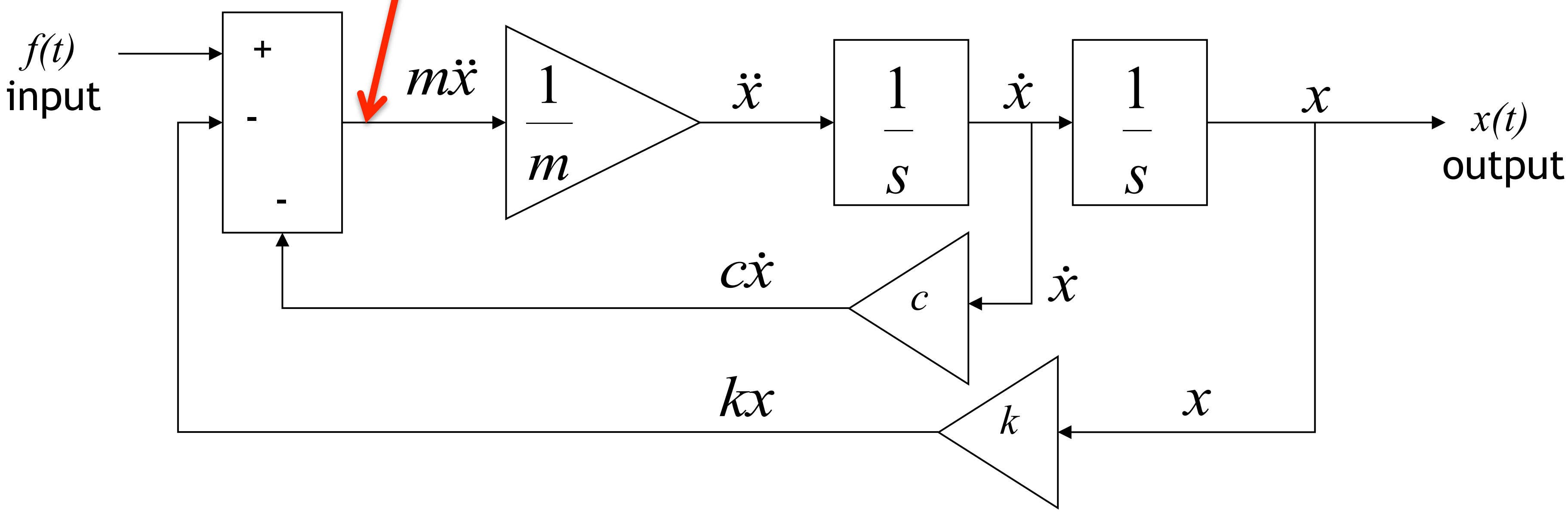


k=1.0

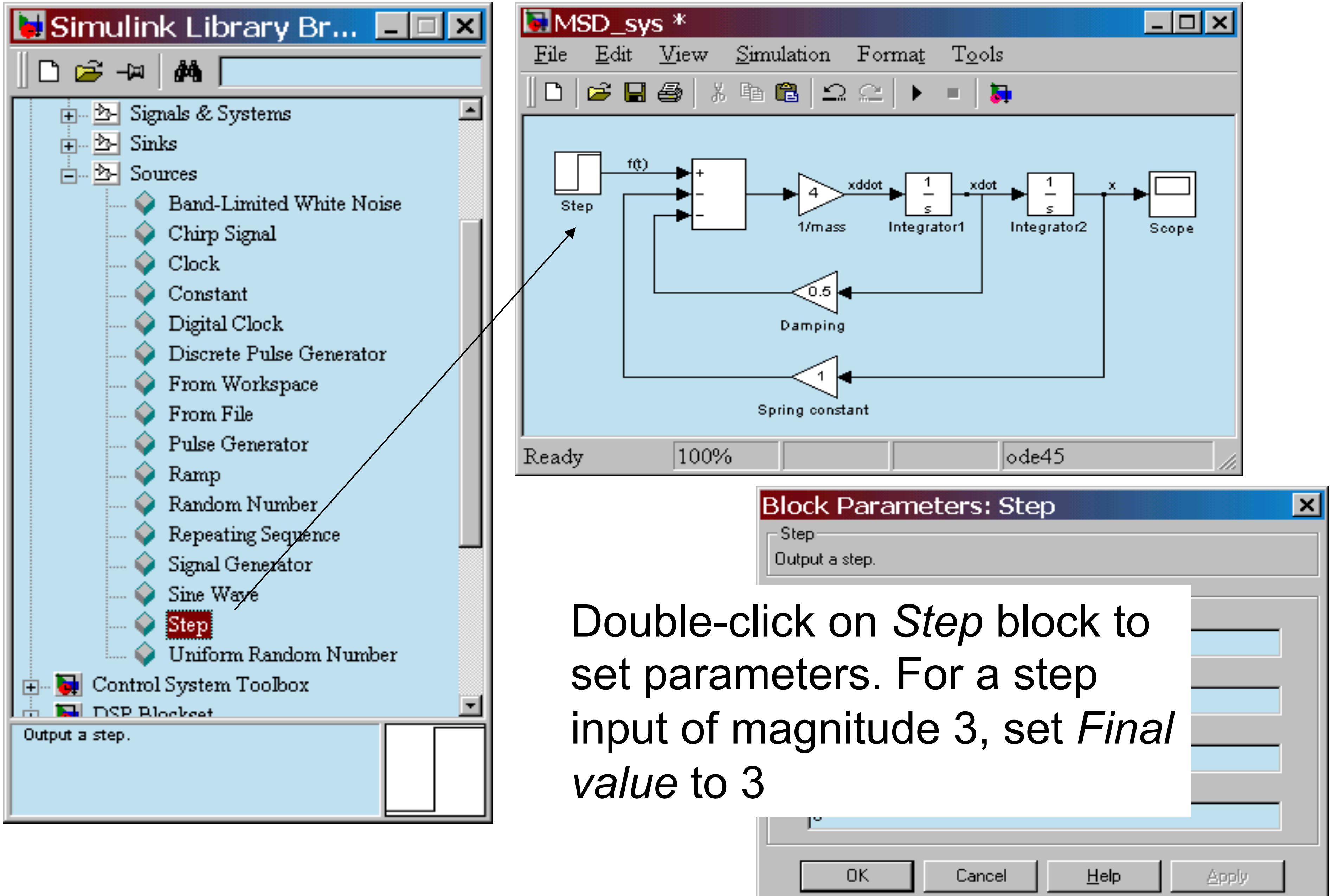
Complete the model

- Bring all the signals and inputs to the summing block
- Check signs on the summer

$$m\ddot{x} = f(t) - c\dot{x} - kx$$

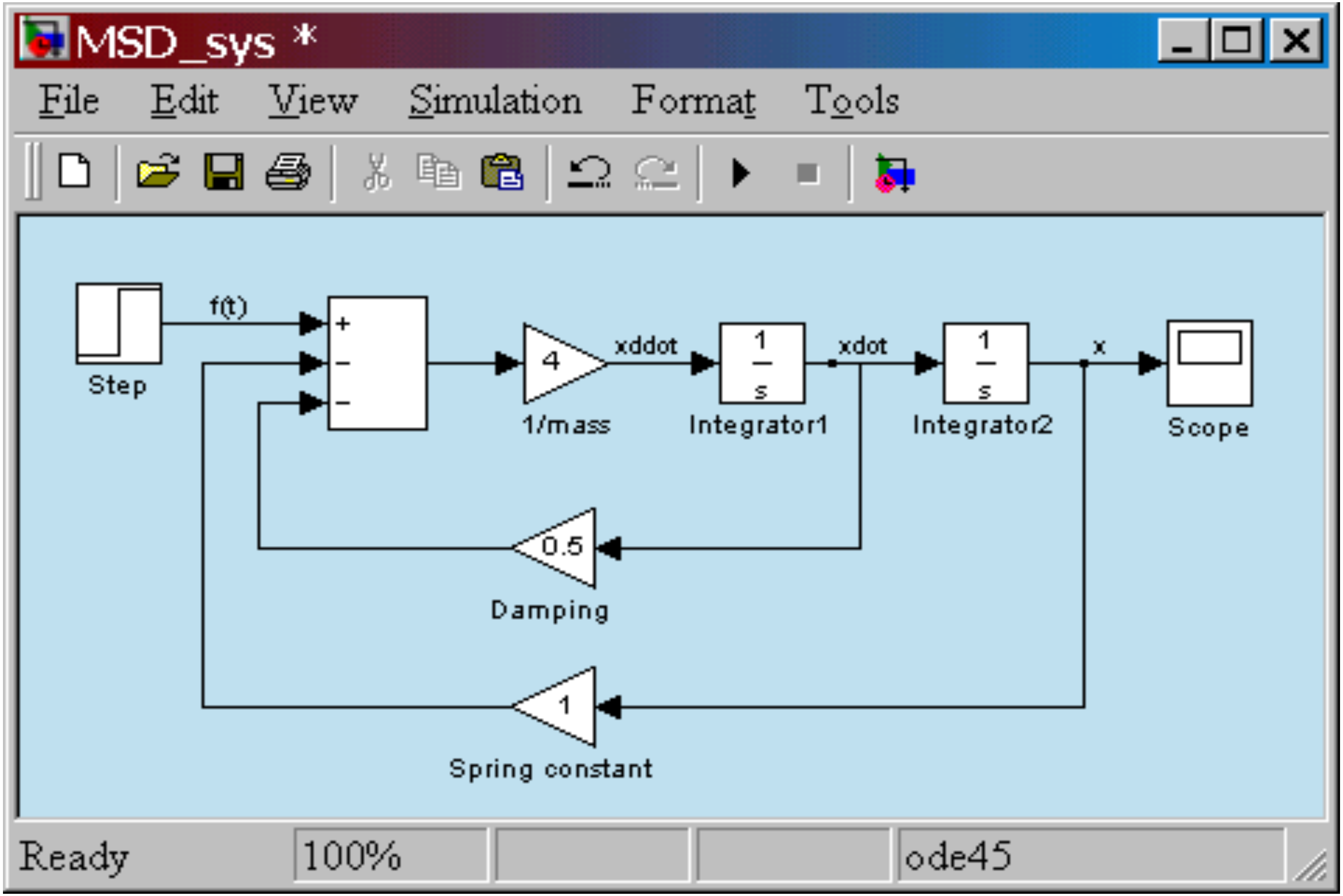


Drive with a step function



Double-click on *Step* block to set parameters. For a step input of magnitude 3, set *Final value* to 3

Final Simulink model



Run the simulation