

MATLAB Introductory courses

Lecture 3

Siamak Ghorbani Faal
sghorbanifaal@wpi.edu



WPI
IT - Academic and Research Computing Center

Course outlines

- Lecture 1: Introduction to MATLAB
- Lecture 2: Using MATLAB
- Lecture 3: Specialized tools

Current lecture outlines

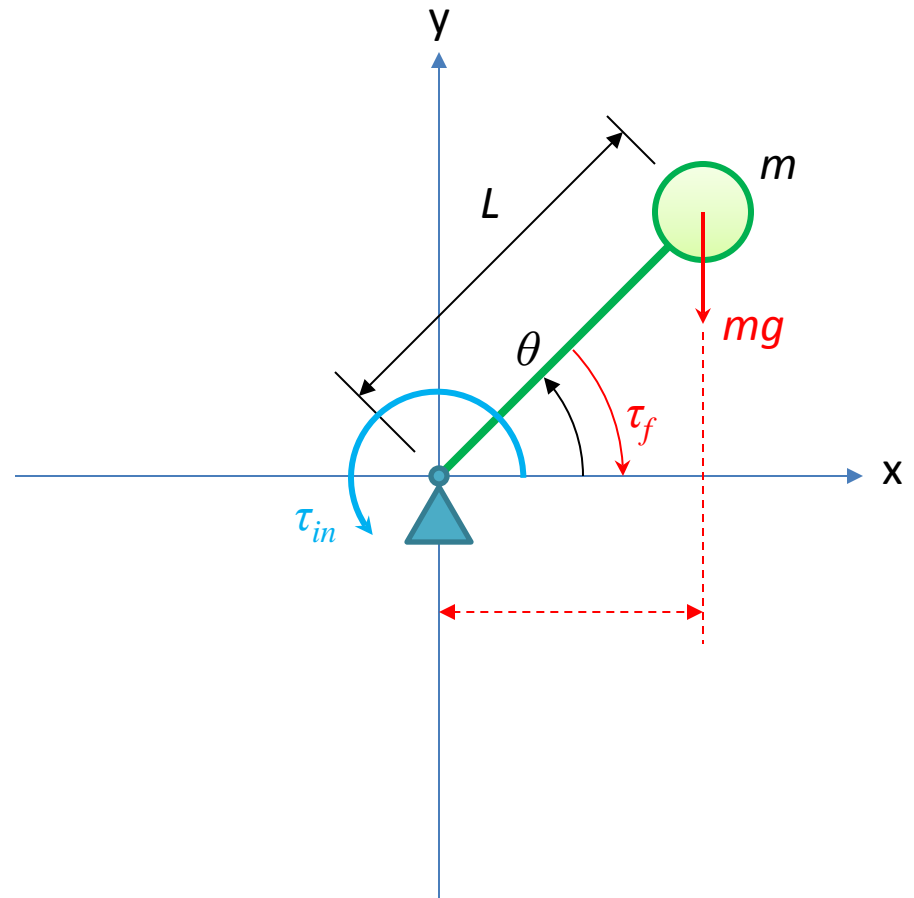
Modeling and simulation of a simple pendulum

Solving differential equations in MATLAB

Creating animations

Symbolic computations

Simple pendulum: Mathematical model



Simple pendulum: Mathematical model

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

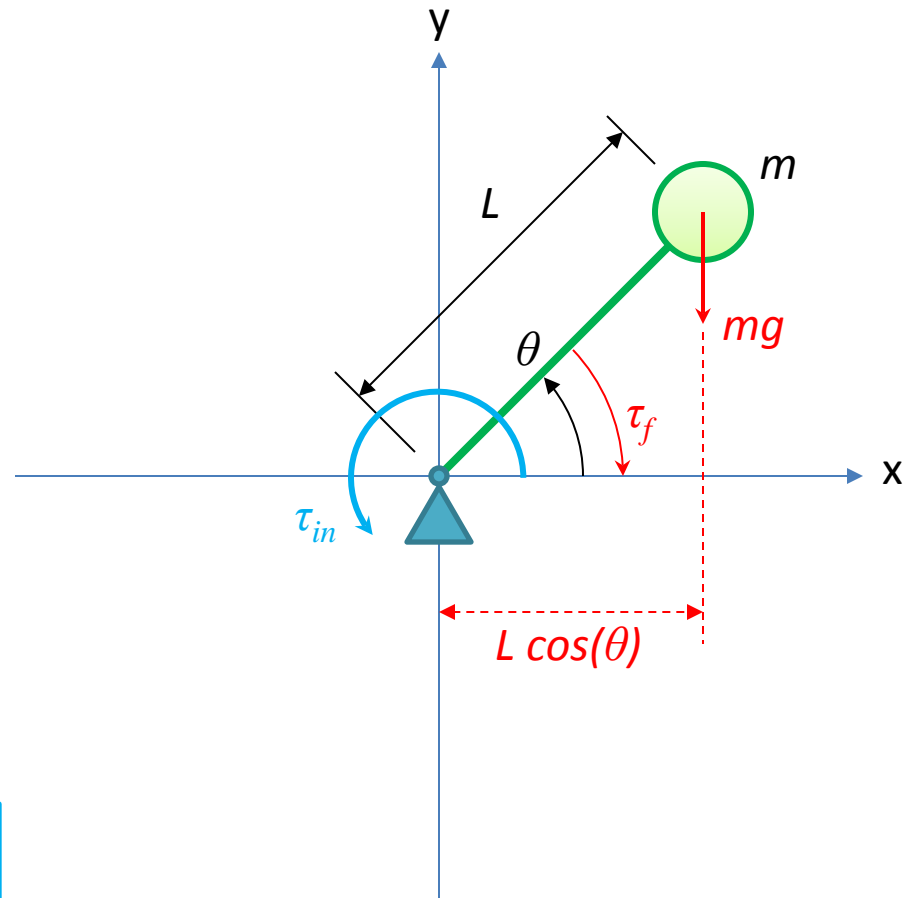
$$\frac{d^2\theta}{dt^2} = \frac{d\omega}{dt} = \ddot{\theta}$$

$$\tau_f = -\text{sign}(\dot{\theta})b\dot{\theta}$$

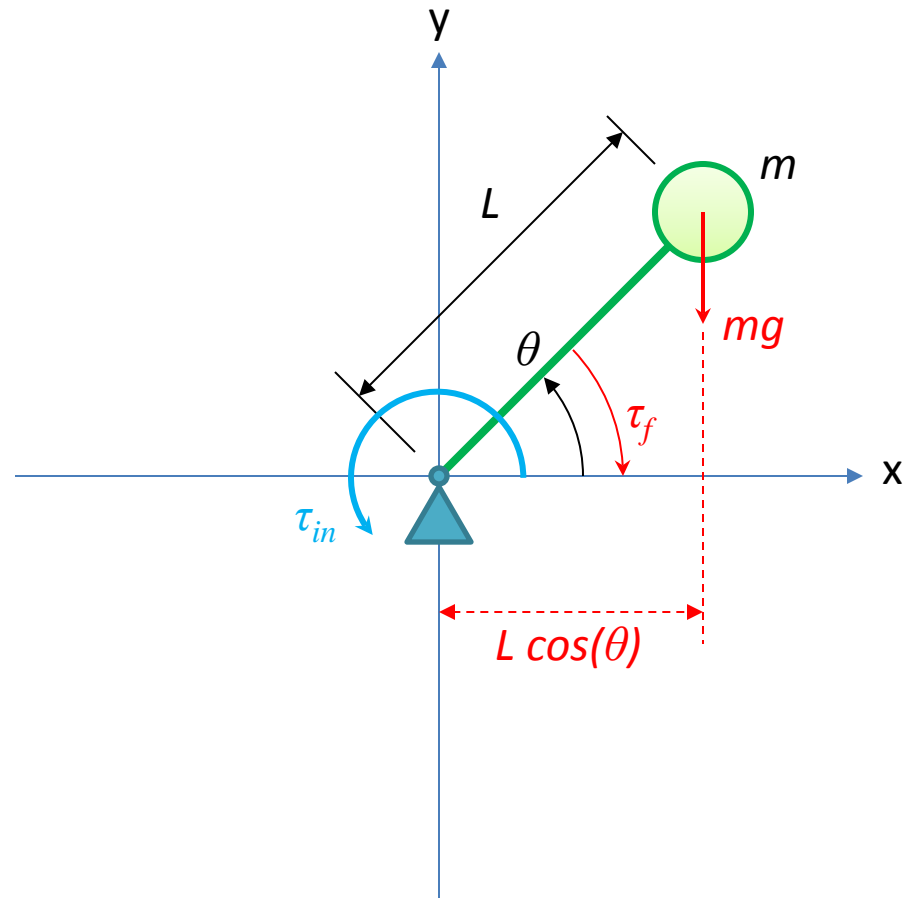
$$\sum \tau_o = I_o \alpha$$

$$-mgL \cos(\theta) - b\dot{\theta} + \tau_{in} = I\ddot{\theta}$$

$$-mgL \cos(\theta) - b\dot{\theta} + \tau_{in} = mL^2\ddot{\theta}$$



Simple pendulum: system of differential equations



Simple pendulum: system of differential equations

$$-mgL \cos(\theta) - b\dot{\theta} + \tau_{in} = mL^2\ddot{\theta}$$

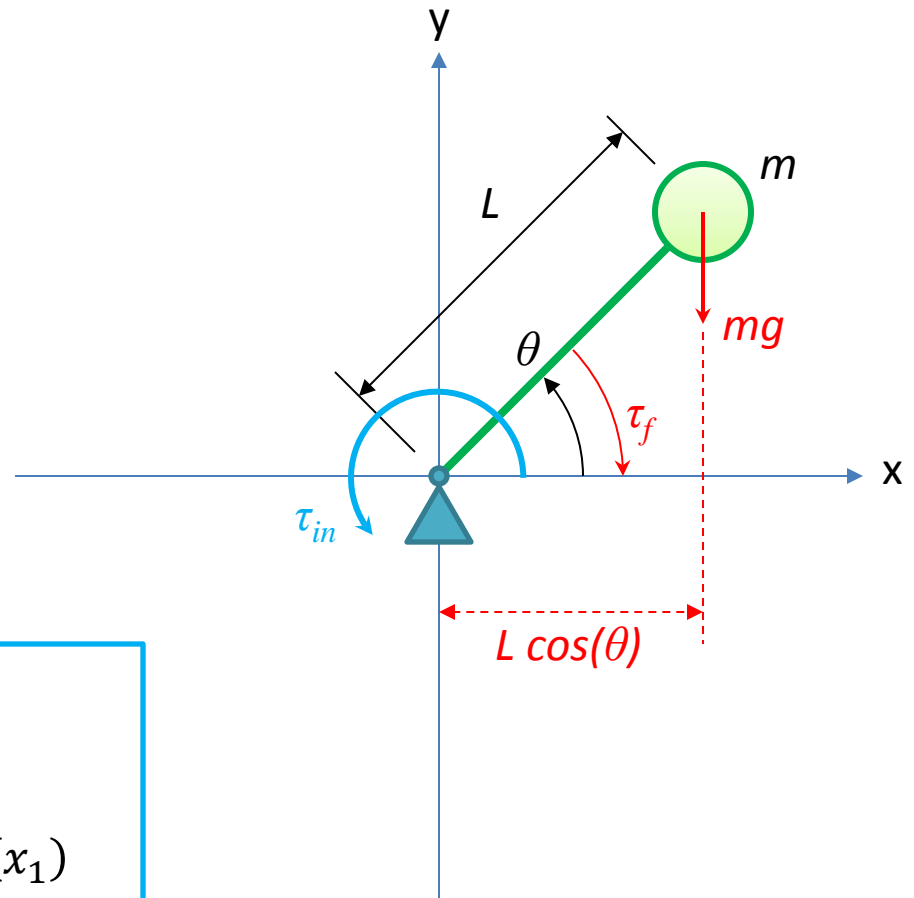
$$\ddot{\theta} = \frac{1}{mL^2}(\tau_{in} - b\dot{\theta}) - \frac{g}{L}\cos(\theta)$$

$$x_1 = \theta$$

$$x_2 = \dot{\theta}$$

$$\dot{x}_1 = x_2 = \dot{\theta}$$

$$\dot{x}_2 = \ddot{\theta} = \frac{1}{mL^2}(\tau_{in} - bx_2) - \frac{g}{L}\cos(x_1)$$



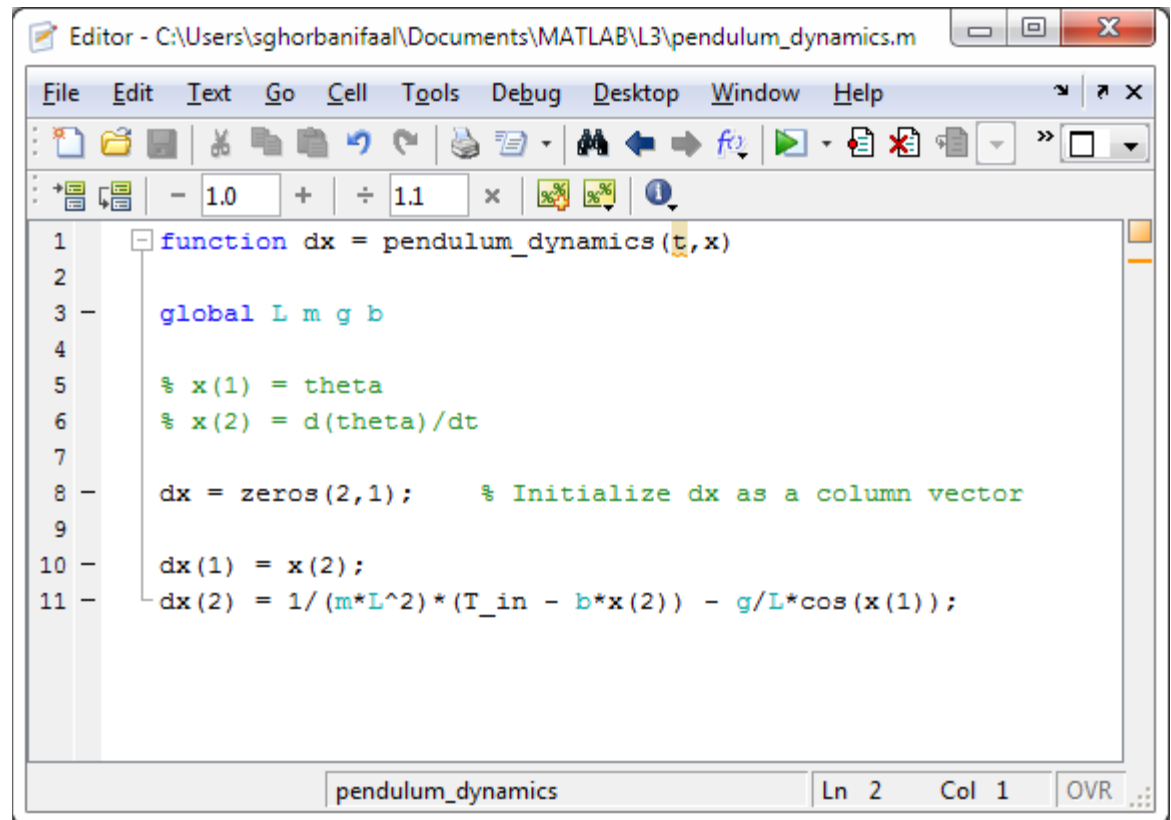
Solving ODE in MTALAB

- ODE solvers
 - ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb
- Fixed Step v.s. Variable Step
- Syntax

```
>> [t,x] = solver(odefun,tspan,x0)
>> [t,x] = solver(odefun,tspan,x0,options)
```
- **odefun**
 - a function handle that evaluates the right side of the differential equations.
- **tspan**
 - is a vector specifying the interval of integration
- **x0**
 - a vector of initial conditions.
- **options**
 - a structure of optional parameters that change the default integration properties.

ODE Function

- $\dot{x} = f(t, x)$



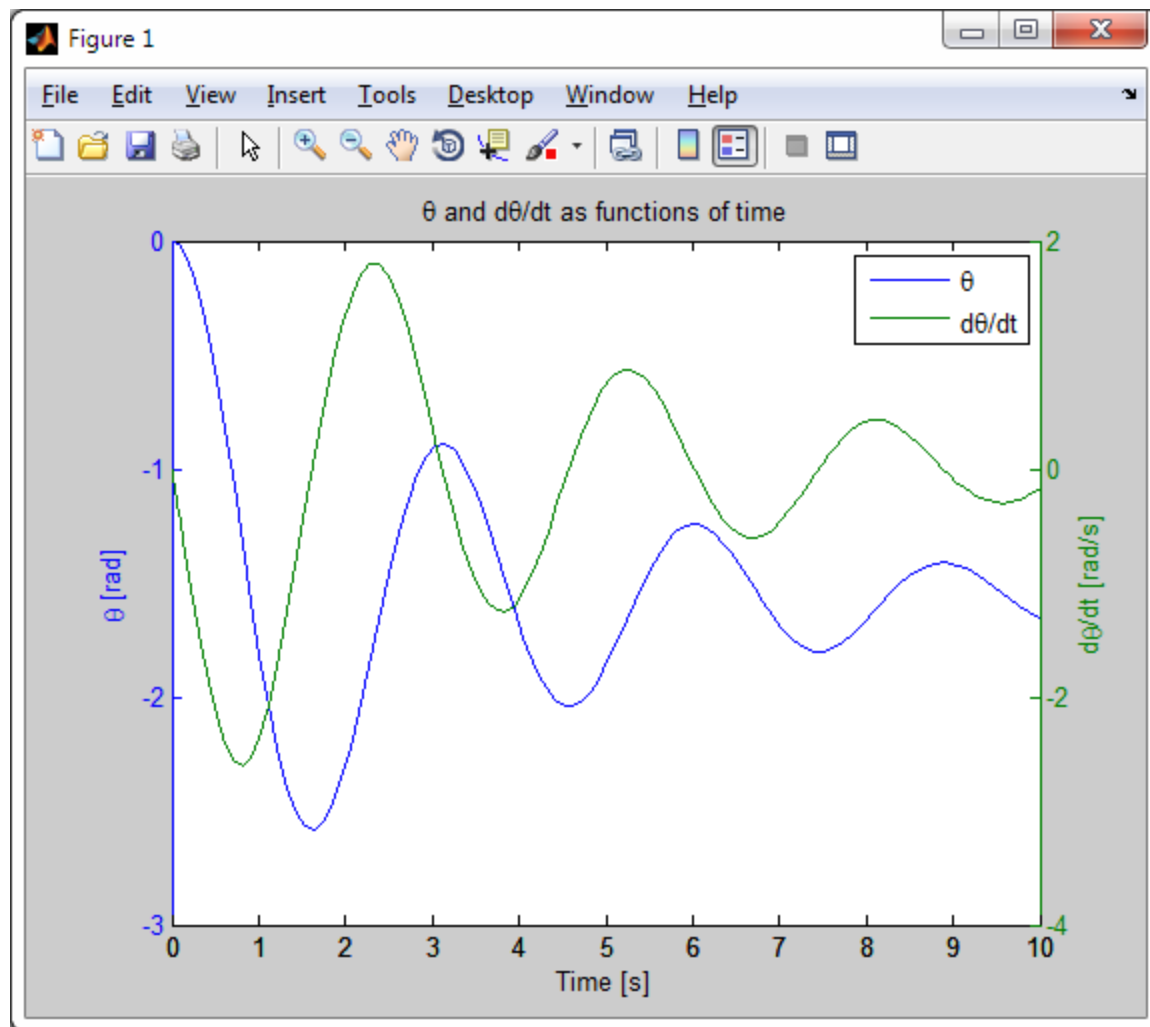
The image shows a MATLAB Editor window titled "Editor - C:\Users\sghorbanifaal\Documents\MATLAB\L3\pendulum_dynamics.m". The window contains the following MATLAB code:

```
1 function dx = pendulum_dynamics(t,x)
2
3     global L m g b
4
5     % x(1) = theta
6     % x(2) = d(theta)/dt
7
8     dx = zeros(2,1);    % Initialize dx as a column vector
9
10    dx(1) = x(2);
11    dx(2) = 1/(m*L^2)*(T_in - b*x(2)) - g/L*cos(x(1));
```

The status bar at the bottom indicates the file name "pendulum_dynamics", the current line "Ln 2", the current column "Col 1", and the view mode "OVR".

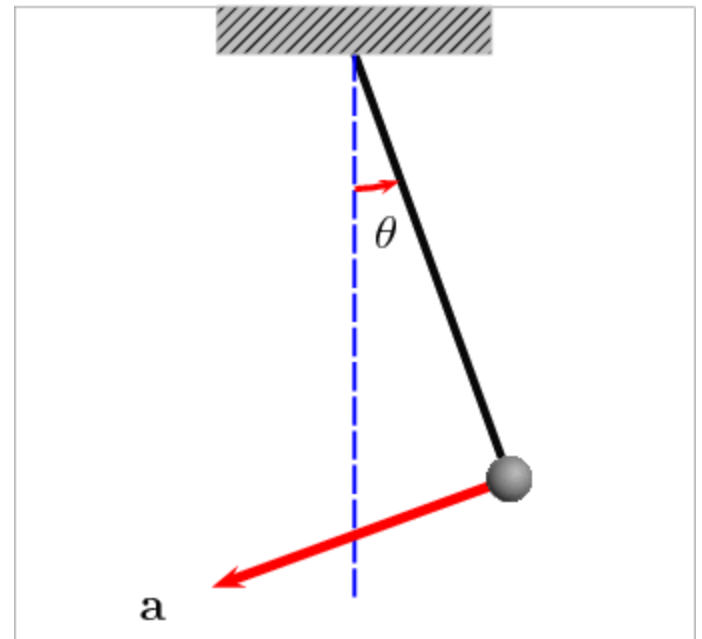
```
Editor - C:\Users\sghorbanifaal\Documents\MATLAB\L3\pendulum_main.m
File Edit Text Go Cell Tools Debug Desktop Window Help
+ 1.0 + 1.1 x % % %
1 %% Pendulum simulation
2
3 %% Initializations
4 clc; clear all; close all;
5
6 global L m g b
7
8 L = 2; % [m]
9 m = 1; % [Kg]
10 g = 9.81; % [m/s^2]
11 b = 2; % [Nms]
12
13 %% Simulation parameters and options
14
15 time = [0 10]; % [s]
16 x0 = [0;0];
17
18 [t, x] = ode45('pendulum_dynamics',time,x0);
19
20 %% Extracting the result
21
22 q = x(:,1);
23 dq = x(:,2);
24
25 [AX,H1,H2] = plotyy(t,q,t,dq,'plot');
26 set(get(AX(1),'Ylabel'),'String','\theta [rad]')
27 set(get(AX(2),'Ylabel'),'String','d\theta/dt [rad/s]')
28 xlabel('Time [s]');
29 title('\theta and d\theta/dt as functions of time');
30 legend('\theta','d\theta/dt');
```

```
Editor - C:\Users\sghorbanifaal\Documents\MATLAB\L3\pendulum_dyna...
File Edit Text Go Cell Tools Debug Desktop Window Help
+ 1.0 + 1.1 x % % %
1 function dx = pendulum_dynamics(t,x)
2
3 global L m g b
4
5 % x(1) = theta
6 % x(2) = d(theta)/dt
7
8 T_in = 0;
9
10 dx = zeros(2,1); % Initialize dx as a column vector
11
12 dx(1) = x(2);
13 dx(2) = 1/(m*L^2)*(T_in - b*x(2)) - g/L*cos(x(1));
```

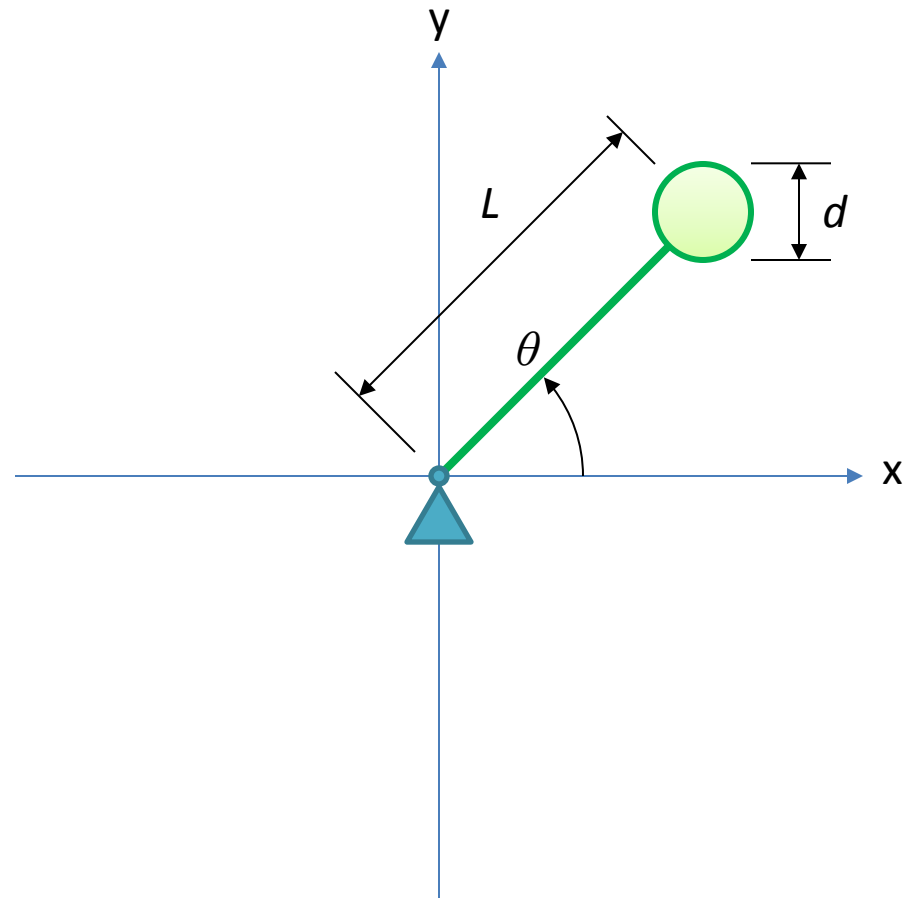


Animation

- Animation is the rapid display of a sequence of images to create an illusion of movement [1].
- To create simple animation in MATLAB you only need to rapidly display sequence of figures!



Plotting a pendulum



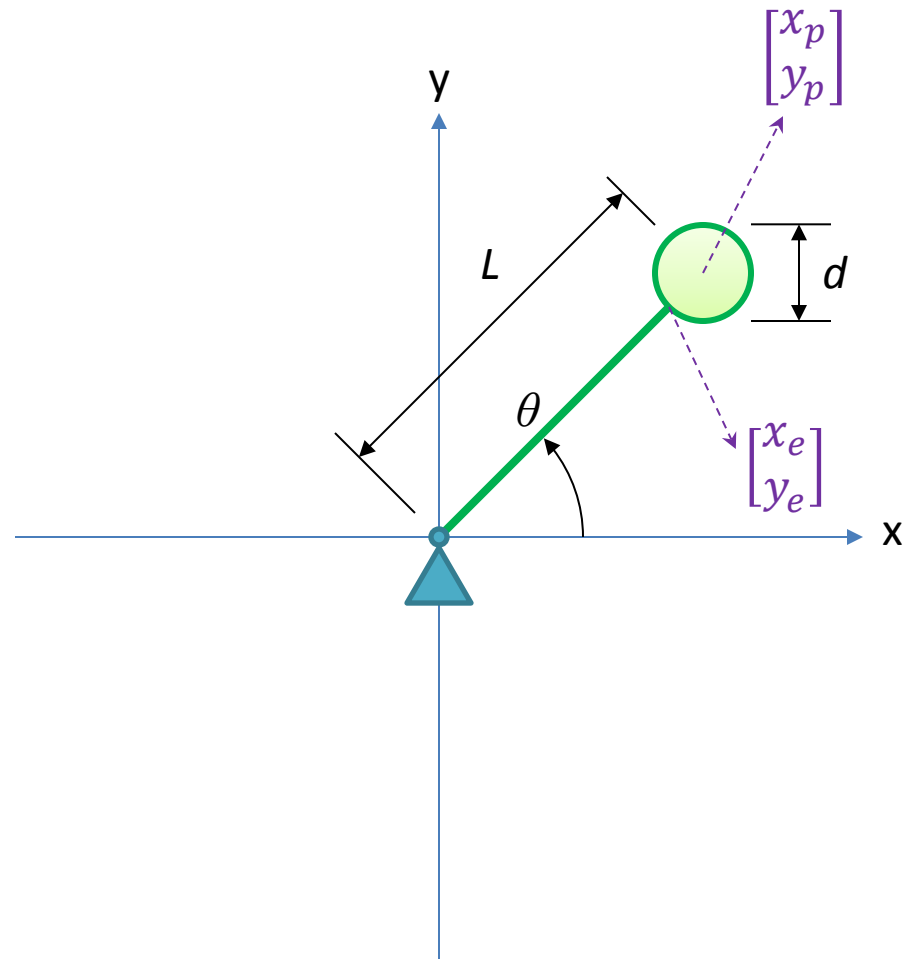
Plotting a pendulum

$$x_p = L \cos(\theta)$$

$$y_p = L \sin(\theta)$$

$$x_e = \left(L - \frac{d}{2}\right) \cos(\theta)$$

$$y_e = \left(L - \frac{d}{2}\right) \sin(\theta)$$



```
Editor - C:\Users\sghorbanifaal\Documents\MATLAB\L3\pendulum_main.m

File Edit Text Go Cell Tools Debug Desktop Window Help

+ [Icons] - 1.0 + ÷ 1.1 × [Icons] [Info]

9 - m = 1; % [Kg]
10 - g = 9.81; % [m/s^2]
11 - b = 2; % [Nms]
12
13 %% Simulation parameters and options
14
15 time = [0 10]; % [s]
16 x0 = [0;0];
17
18 [t, x] = ode45('pendulum_dynamics',time,x0);
19
20 %% Extracting the result
21
22 q = x(:,1);
23 dq = x(:,2);
24
25 %% Plotting theta and d(theta)/dt
26 F1 = figure();
27 [AX,H1,H2] = plotyy(t,q,t,dq,'plot');
28 set(get(AX(1),'Ylabel'),'String','\theta [rad]');
29 set(get(AX(2),'Ylabel'),'String','d\theta/dt [rad/s]');
30 xlabel('Time [s]');
31 title('\theta and d\theta/dt as functions of time');
32 legend('\theta','d\theta/dt');
33
34 %% Animation
35 F2 = figure();
36 animate_pendulum(t,q);
37
38 pendulum_dynamics.m × pendulum_main.m × animate_pendulum.m
script Ln 36
```

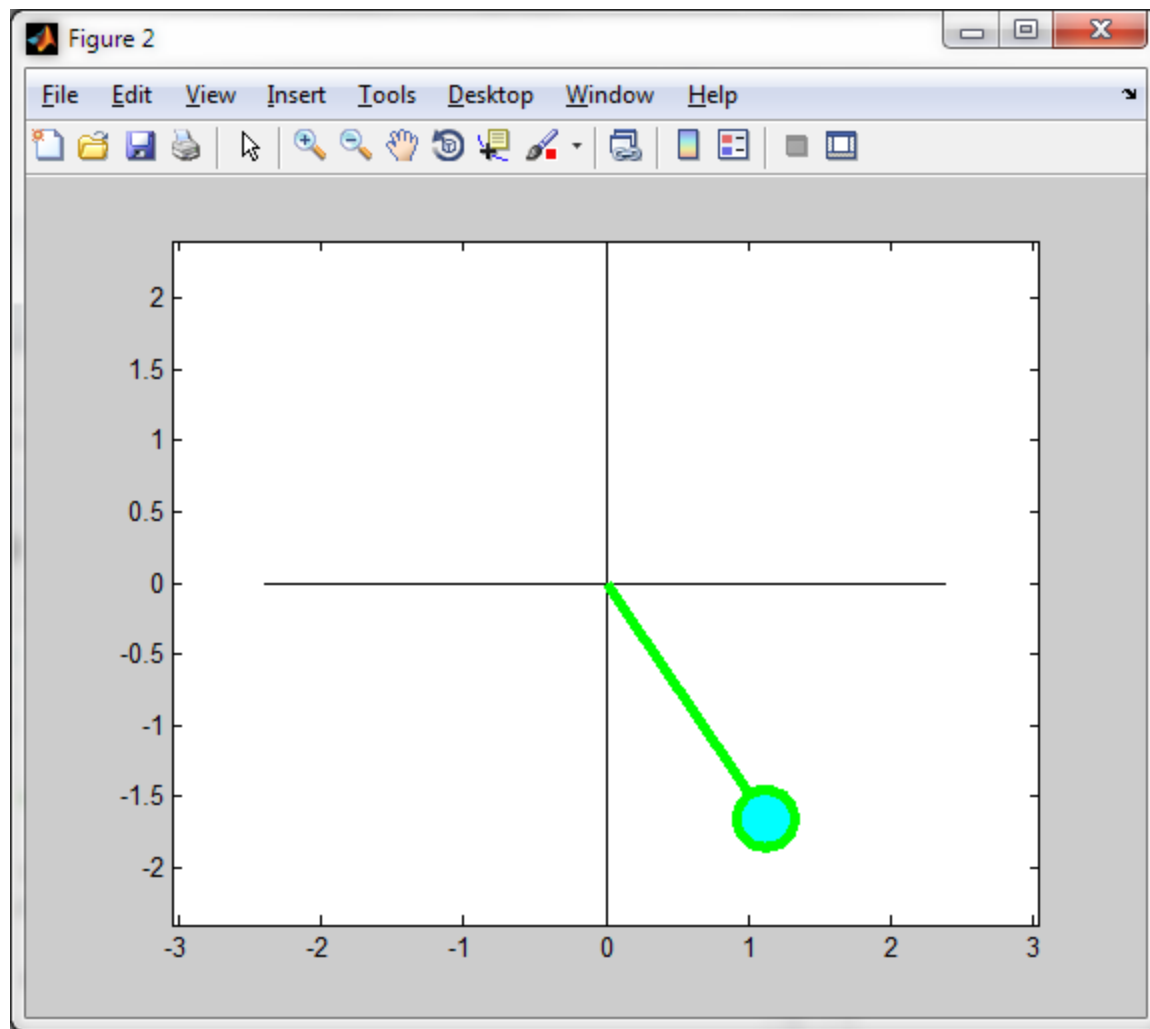
```
Editor - C:\Users\sghorbanifaal\Documents\MATLAB\L3\animate...

File Edit Text Go Cell Tools Debug Desktop Window » [Icons] [Info]

+ [Icons] - 1.0 + ÷ 1.1 × [Icons] [Info]

1 - function animation_frames = animate_pendulum(t,q)
2
3 - global L
4
5 - d = L/5; r = d/2; Limit = L + d;
6 - Q = 0:10:360;
7 - x_c = r*cosd(Q); y_c = r*sind(Q);
8
9 - for n=1:length(t)
10 - x_p = L*cos(q(n));
11 - y_p = L*sin(q(n));
12 - x_e = (L - r)*cos(q(n));
13 - y_e = (L - r)*sin(q(n));
14 - x_b = x_c + x_p;
15 - y_b = y_c + y_p;
16
17 - plot([-Limit Limit],[0 0],'k');
18 - hold on
19 - plot([0 0],[-Limit Limit],'k');
20 - fill(x_b,y_b,'c',...
21 - 'LineWidth',4,'EdgeColor','g');
22 - plot([0 x_e],[0 y_e],'g','LineWidth',4);
23 - axis(Limit*[-1 1 -1 1]);
24 - axis equal
25 - drawnow();
26 - hold off
27 - end
28 - animation_frames = 0;

pendulum_main.m × animate_pendulum.m* ×
animate_pendulum Ln 27 Col 4 OVR
```



getfarme and movie functions

```
Editor - C:\Users\sghorbanifaa\Documents\MATLAB\L3\pendulu...
File Edit Text Go Cell Tools Debug Desktop Window »
+ - 1.0 + ÷ 1.1 x % % %
10 - g = 9.81; % [m/s^2]
11 - b = 2; % [Nms]
12
13 %% Simulation parameters and options
14
15 - time = [0 10]; % [s]
16 - x0 = [0;0];
17
18 - [t, x] = ode45('pendulum_dynamics',time,x0);
19
20 %% Extracting the result
21
22 - q = x(:,1);
23 - dq = x(:,2);
24
25 %% Plotting theta and d(theta)/dt
26 - F1 = figure();
27 - [AX,H1,H2] = plotyy(t,q,t,dq,'plot');
28 - set(get(AX(1),'Ylabel'),'String','\theta [rad]')
29 - set(get(AX(2),'Ylabel'),'String','d\theta/dt [r]')
30 - xlabel('Time [s]');
31 - title('\theta and d\theta/dt as functions of ti');
32 - legend('\theta','d\theta/dt');
33
34 %% Animation
35 - F2 = figure();
36 - frames = animate_pendulum(t,q);
37 - movie(frames);
38
pendulum_main.m x animate_pendulum.m x
script Ln 36 Col 32
```

```
Editor - C:\Users\sghorbanifaa\Documents\MATLAB\L3\animate...
File Edit Text Go Cell Tools Debug Desktop Window »
+ - 1.0 + ÷ 1.1 x % % %
1 - function animation_frames = animate_pendulum(t,q)
2
3 - global L
4
5 - d = L/5; r = d/2; Limit = L + d;
6 - Q = 0:10:360;
7 - x_c = r*cosd(Q); y_c = r*sind(Q);
8
9 - for n=1:length(t)
10 - x_p = L*cos(q(n));
11 - y_p = L*sin(q(n));
12 - x_e = (L - r)*cos(q(n));
13 - y_e = (L - r)*sin(q(n));
14 - x_b = x_c + x_p;
15 - y_b = y_c + y_p;
16
17 - plot([-Limit Limit],[0 0],'k');
18 - hold on
19 - plot([0 0],[-Limit Limit],'k');
20 - fill(x_b,y_b,'c',...
21 - 'LineWidth',4,'EdgeColor','g');
22 - plot([0 x_e],[0 y_e],'g','LineWidth',4);
23 - axis(Limit*[-1 1 -1 1]);
24 - axis equal
25 - animation_frames(n) = getframe();
26 - hold off
27 - end
animate_pendulum Ln 27 Col 4 OVR
```