MATLAB Practice Problem 5

Komsun Tamana

1TE18968Y

Guidance and Control Laboratory

2021-05-07

図のような倒立台車の運動方程式を平衡点周りで線形化し安定化するような入力を加えた自律系で平衡状態でない任意の初期状態からの θ, x の時間変化を求め，その結果を用いて簡易的なアニメーションを作成せよ．また，線形化して求めた安定化する入力を線形化まえの非線形の運動方程式に入力し，平衡状態でない任意の初期状態からの θ, x の時間変化を求めよ．ただし，必要な条件は自分で設定し，ode を用いること．

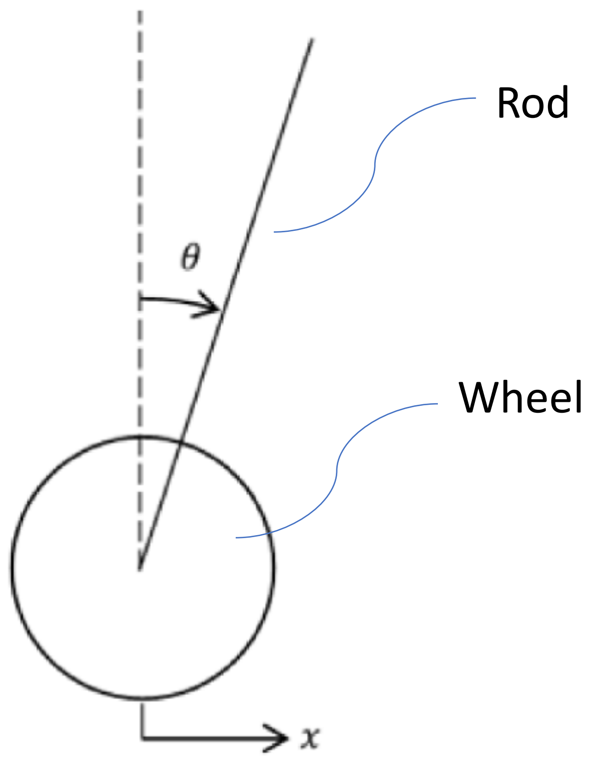


図 3: 問５の状況図

Find the Equation of Motion

↓

State variables form

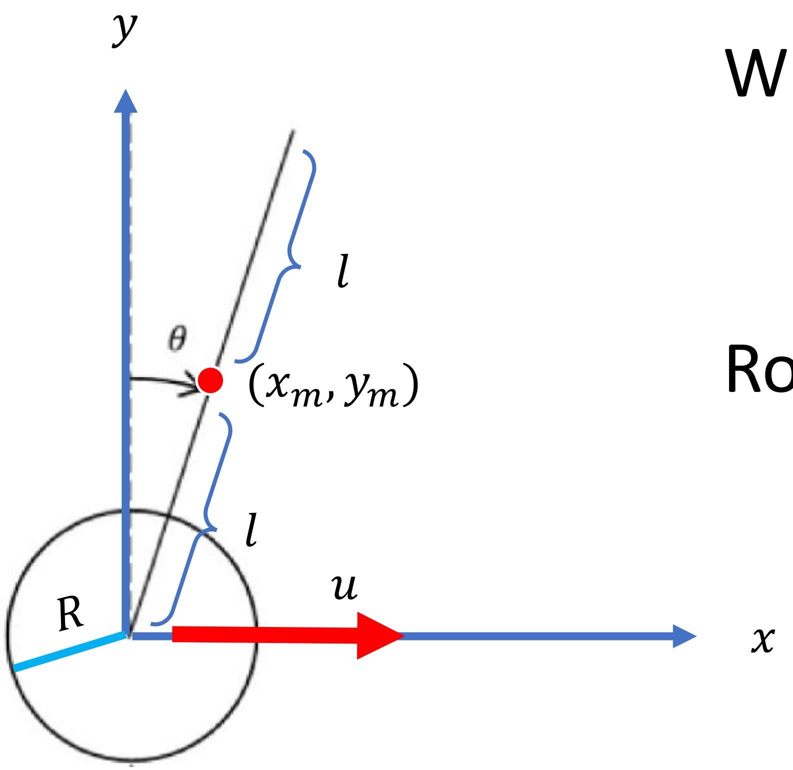
↓

ODE45

↓

Animation

Find the Equation of motion



Wheel

Horizontal displacement : x
Horizontal velocity : $\dot{x}$

Rod

$$x_m = x + l sin\theta$$
$$\dot{x_m} = \dot{x} + l\dot{\theta} cos\theta$$

$$y_m = l cos\theta$$
$$\dot{y_m} = -l\dot{\theta} sin\theta$$

Find the Equation of motion

Total Kinetic Energy

$$T = T_{wheel} + T_{rod}$$

$$= \left\{\frac{1}{2}M\dot{x}^2\right\} + \left\{\frac{1}{2}m\left(\dot{x_m}^2 + \dot{y_m}^2\right)\right\}$$

$$= \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 + 2\dot{x}\dot{\theta}lcos\theta + \dot{\theta}^2l^2\cos^2\theta + \dot{\theta}^2l^2\sin^2\theta)$$

$$= \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}cos\theta$$

Find the Equation of motion

Total Potential Energy

$$U = U_{wheel} + U_{mass}$$
$$= 0 + mgy_m$$
$$= mglcos\theta$$

Virtual work

$$\delta W^{nc} = (-d\dot{x} + u)\delta x$$

Nonconservative force

$$Q_x^{nc} = -d\dot{x} + u$$

Find the Equation of motion

Lagrangean

$$L = T - U$$
$$= \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}ml^2\dot{\theta}^2 + ml\dot{\theta}\dot{x}cos\theta - mglcos\theta$$

$$\frac{\partial L}{\partial \dot{x}} = (M+m)\dot{x} + ml\dot{\theta}cos\theta$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2\dot{\theta} + ml\dot{x}cos\theta$$

$$\frac{\partial L}{\partial x} = 0$$

$$\frac{\partial L}{\partial \theta} = -ml\dot{\theta}\dot{x}sin\theta + mglsin\theta$$

Find the Equation of motion

Langrangean's Equation of Motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = Q_x^{nc}$$

$$(M + m)\ddot{x} + ml\ddot{\theta}cos\theta - ml\dot{\theta}^2 sin\theta = -c\dot{x} + u \qquad (1)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = Q_\theta^{nc}$$

$$l\ddot{\theta} + \ddot{x}cos\theta - gsin\theta = 0$$

$$\ddot{\theta} = \frac{gsin\theta - \ddot{x}cos\theta}{l} \qquad (2)$$

From (1), (2)

$$\ddot{x} = \frac{-m^2 l^2 g cos\theta sin\theta + ml^2\left(ml\dot{\theta}^2 sin\theta - d\dot{x}\right) + ml^2 u}{ml^2(M + m(1 - \cos^2\theta))}$$

$$\ddot{\theta} = \frac{(m+M)mglsin(\theta) - mlcos\theta\left(ml\dot{\theta}^2\sin(\theta) - d\dot{x}\right) + mlcos(\theta)u}{ml^2(M+m(1-\cos^2\theta))}$$

State variables form

State Vector

$$[s] = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$[\dot{s}] = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} s_2 \\ \dfrac{-m^2 l^2 g\cos(s_3)\sin(s_3) + ml^2(mls_4^2\sin(s_3) - ds_2) + ml^2 u}{ml^2(M + m(1 - \cos^2(s_3))} \\ s_4 \\ \dfrac{(m + M)mglsin(s_3) - mlcos(s_3)\left(ml\dot{\theta}^2 \sin(s_3) - ds_2\right) + mlcos(s_3)u}{ml^2(M + m(1 - \cos^2(s_3)))} \end{bmatrix}$$

$$
[\dot{s}] = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} s_2 \\ \dfrac{-m^2 l^2 g\cos(s_3)\sin(s_3) + ml^2(mls_4^2\sin(s_3) - ds_2) + ml^2 u}{ml^2(M + m(1-\cos^2(s_3))} \\ s_4 \\ \dfrac{(m+M)mglsin(s_3) - mlcos(s_3)\left(ml\dot{\theta}^2\sin(s_3) - ds_2\right) + mlcos(s_3)u}{ml^2(M + m(1-\cos^2(s_3)))} \end{bmatrix}
$$

```matlab
function ds = wheelpend(s,m,M,L,g,d,u)

Sx = sin(s(3));                    % sin(theta)
Cx = cos(s(3));                    % cos(theta)
D = m*L*L*(M+m*(1-Cx^2));         % Denominator

ds(1,1) = s(2);
ds(2,1) = (1/D)*(-m^2*L^2*g*Cx*Sx + m*L^2*(m*L*s(4)^2*Sx - d*s(2))) + m*L*L*(1/D)*u;
ds(3,1) = s(4);
ds(4,1) = (1/D)*((m+M)*m*g*L*Sx - m*L*Cx*(m*L*s(4)^2*Sx - d*s(2))) - m*L*Cx*(1/D)*u;
end
```

Main script – Free fall

```
clc, clear, close all

% Set-up
m = 1;          % rod's mass (kg)
M = 5;          % wheel's mass (kg)
L = 2;          % rod's half length (m)
g = -9.81;       % Earth's gravity (m/s^2)
d = 5;          % drag coefficient (kg/s)

tspan = 0:0.1:20;               % Time span for the simulation
y0 = [0; 0; pi-pi/4; 0];        % Initial conditions

[t,state_values] = ode45(@(t,y)wheelpend(y,m,M,L,g,d,0),tspan,y0);

x = state_values(:,1);
xdot = state_values(:,2);
theta = state_values(:,3);
thetadot = state_values(:,4);

for k = 1:length(t)
    drawwheel(state_values(k,:),m,M,L);
    title(num2str(t(k),'time = %4.3f s'));  % timer display
end
```

Initial position: 0 m

Initial angle: +45 (from +y axis)

Initial velocity: 0 m/s

Initial angular velocity: 0 rad/s

Use ODE45 referring to *wheelpend* function

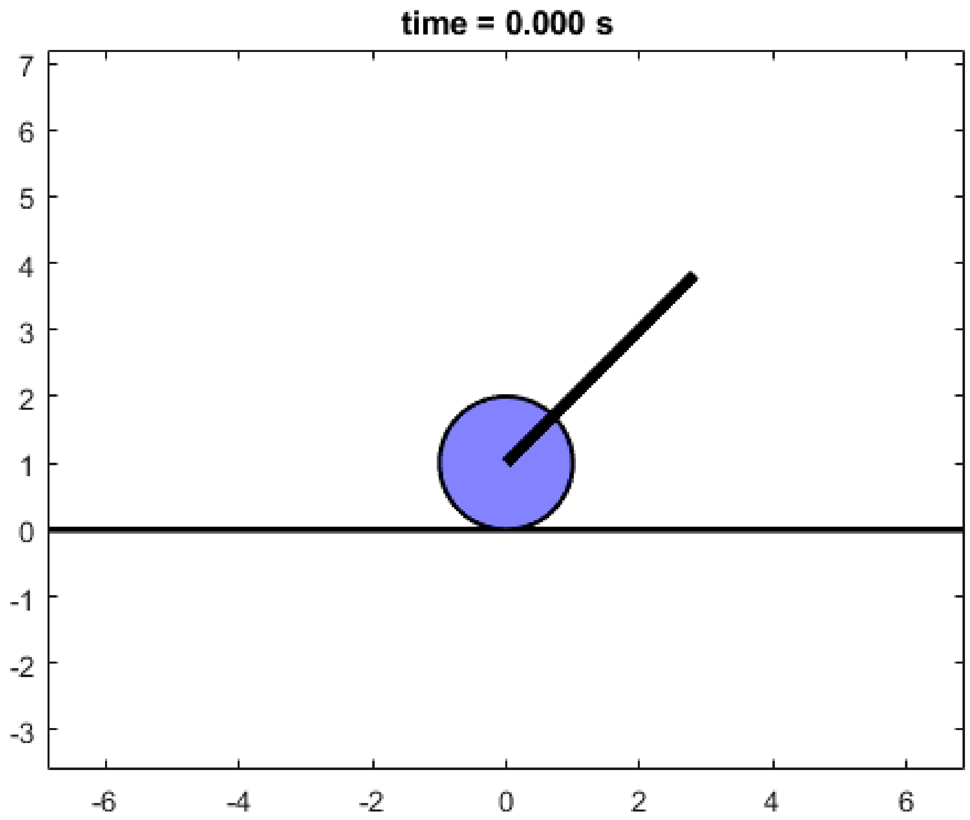
Making an animation

Animation creation

```matlab
function drawwheel(state,m,M,L)
x = state(1);
th = state(3);

% dimensions
R = sqrt(M/5);                  % Wheel's radius

% positions
y = R;                          % Wheel's vertical position
x_rod = x + 2*L*sin(th);        % Rod's horizontal position
y_rod = y - 2*L*cos(th);        % Rod's vertical position

% Drawing
plot([-10 10],[0 0],'k','LineWidth',2), hold on % Draw a ground level
rectangle('Position',[x-R,y-R,2*R,2*R],'Curvature',[1 1]...
    ,'FaceColor',[.5 0.5 1],'LineWidth',1.5);   % Draw a Wheel
plot([x x_rod],[y y_rod],'k','LineWidth',4);    % Draw a Rod

axis([-1.5*L 1.5*L -L 1.8*2*L]);
axis equal
set(gcf,'Position',[100 100 1000 400])
drawnow, hold off
```

My own setup (arrow pointing to `R = sqrt(M/5);`)

Free fall result

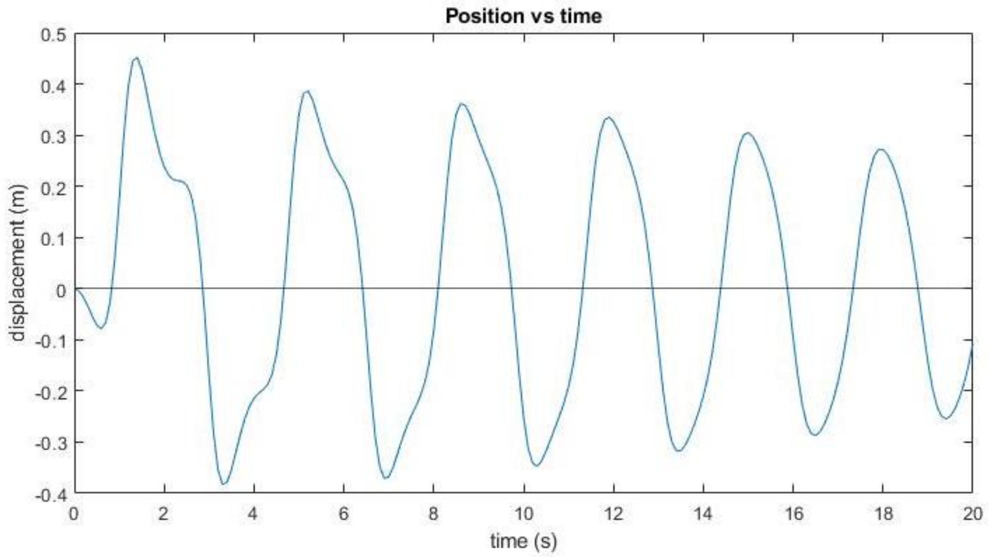


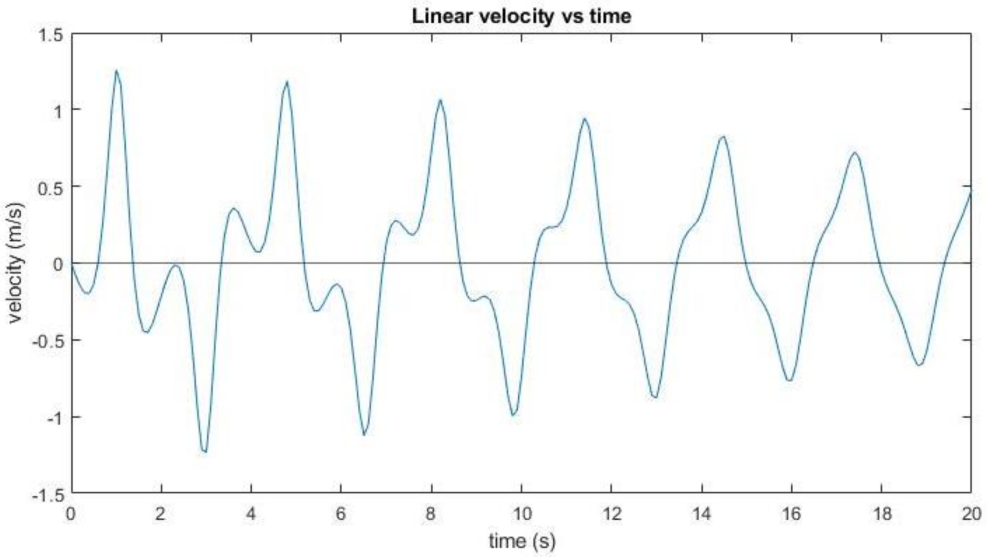
Initial position: 0 m

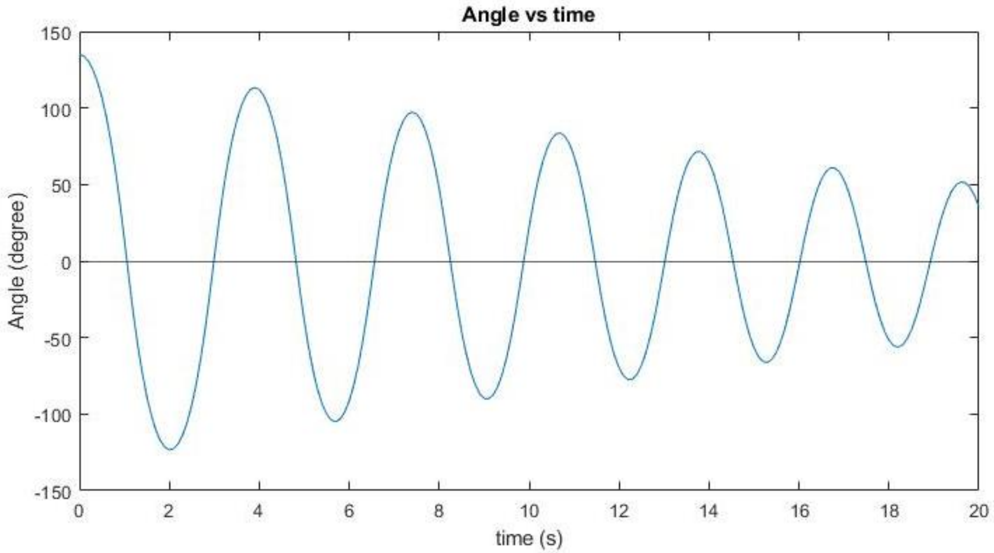
Initial angle: +45 (from +y axis)

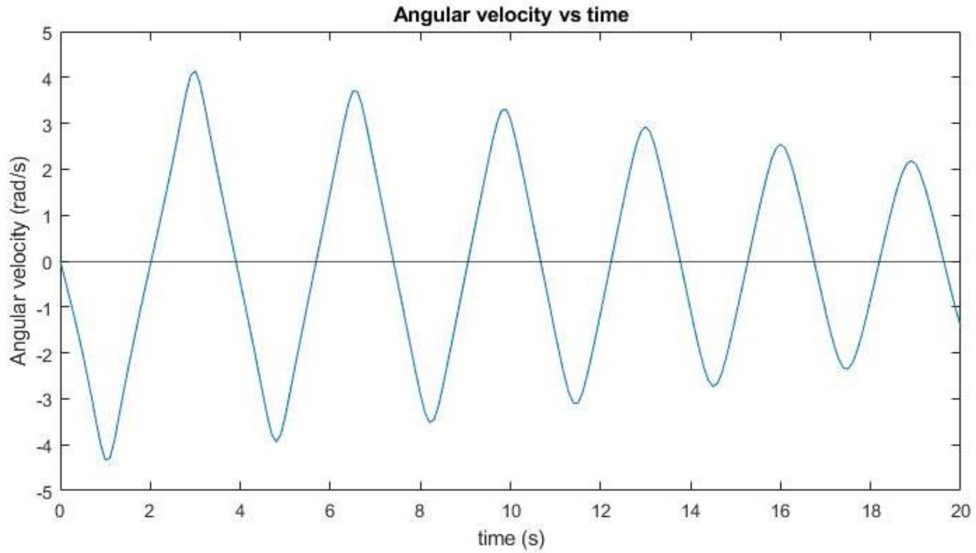
Initial velocity: 0 m/s

Initial angular velocity: 0 rad/s


Position vs time
displacement (m)
time (s)
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
0 2 4 6 8 10 12 14 16 18 20


Linear velocity vs time
velocity (m/s)
time (s)
1.5
1
0.5
0
-0.5
-1
-1.5
0 2 4 6 8 10 12 14 16 18 20

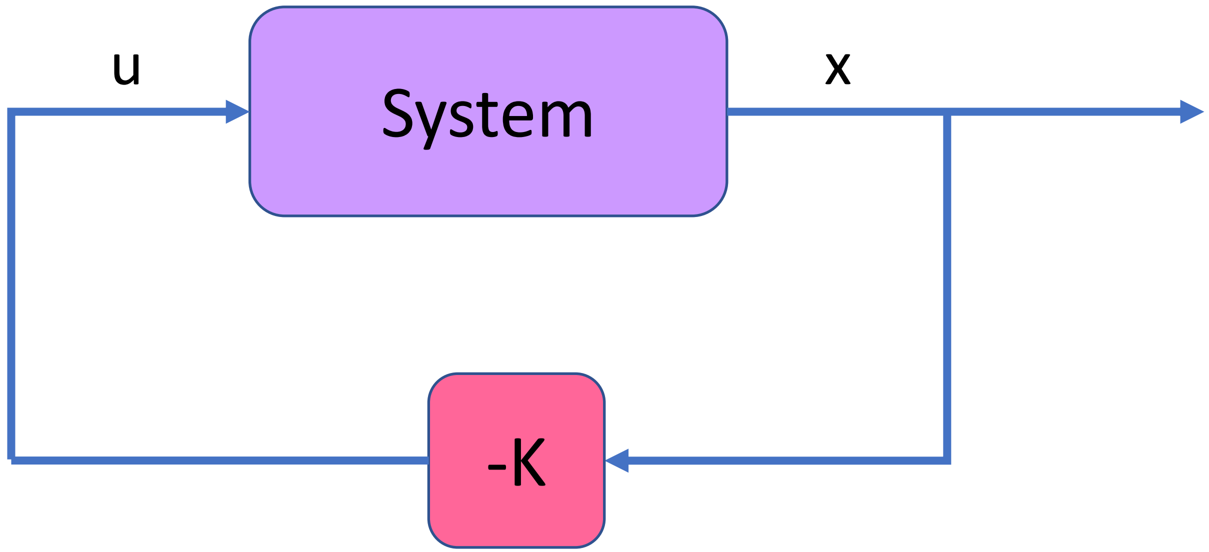

Angle vs time
Angle (degree)
time (s)
150
100
50
0
-50
-100
-150
0 2 4 6 8 10 12 14 16 18 20


Angular velocity vs time
Angular velocity (rad/s)
time (s)
5
4
3
2
1
0
-1
-2
-3
-4
-5
0 2 4 6 8 10 12 14 16 18 20

How to control?

- Full-state feedback control

Full-state feedback control



$$u = -Kx$$

$$\dot{x} = Ax + Bu = (A - BK)x$$

Full-state feedback control

$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$\dot{\boldsymbol{s}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} s_2 \\ \dfrac{-m^2 l^2 g\cos(s_3)\sin(s_3) + ml^2\left(mls_4^2\sin(s_3) - ds_2\right) + ml^2 u}{ml^2(M+m(1-\cos^2(s_3))} \\ s_4 \\ \dfrac{(m+M)mglsin(s_3) - mlcos(s_3)\left(ml\dot{\theta}^2\sin(s_3) - ds_2\right) + mlcos(s_3)u}{ml^2(M+m(1-\cos^2(s_3)))} \end{bmatrix}$$

Linearize

$$\dot{\boldsymbol{s}} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{B}\boldsymbol{u}$$

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{d}{M} & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{d}{ML} & -\frac{(m+M)g}{ML} & 0 \end{bmatrix}$$

$$\boldsymbol{B} = \begin{bmatrix} 0 & \frac{1}{M} & 0 & \frac{1}{ML} \end{bmatrix}^T$$

Main script – Controlled

```
clc, clear all, close all

% Set-up
m = 1;       % rod's mass (kg)
M = 5;       % wheel's mass (kg)
L = 2;       % rod's half length (m)
g = -9.81;   % Earth's gravity (m/s^2)
d = 1;       % drag coefficient (kg/s)

A = [0    1     0     0;
     0  -d/M -m*g/M 0;
     0    0     0     1;
     0 d/(M*L) -(m+M)*g/(M*L) 0];

B = [0; 1/M; 0; 1/(M*L)];

eig(A)  % This shows that the system is currently unstable

new_eig = [-1.3, -1.4, -1.5, -1.6];   % slow
% new_eig = [-2, -2.1, -2.2, -2.3];      % medium
% new_eig = [-3, -3.1, -3.2, -3.3];     % fast


K = place(A, B, new_eig);

tspan = 0:0.1:20;
y0 = [-3; 0; pi-pi/6; 0];
targ_pos = 2; % Targetted position (m)
[t,state_values] = ode45(@(t,y)wheelpend(y,m,M,L,g,d,-K*(y -[targ_pos; 0; pi; 0])),tspan,y0)
```

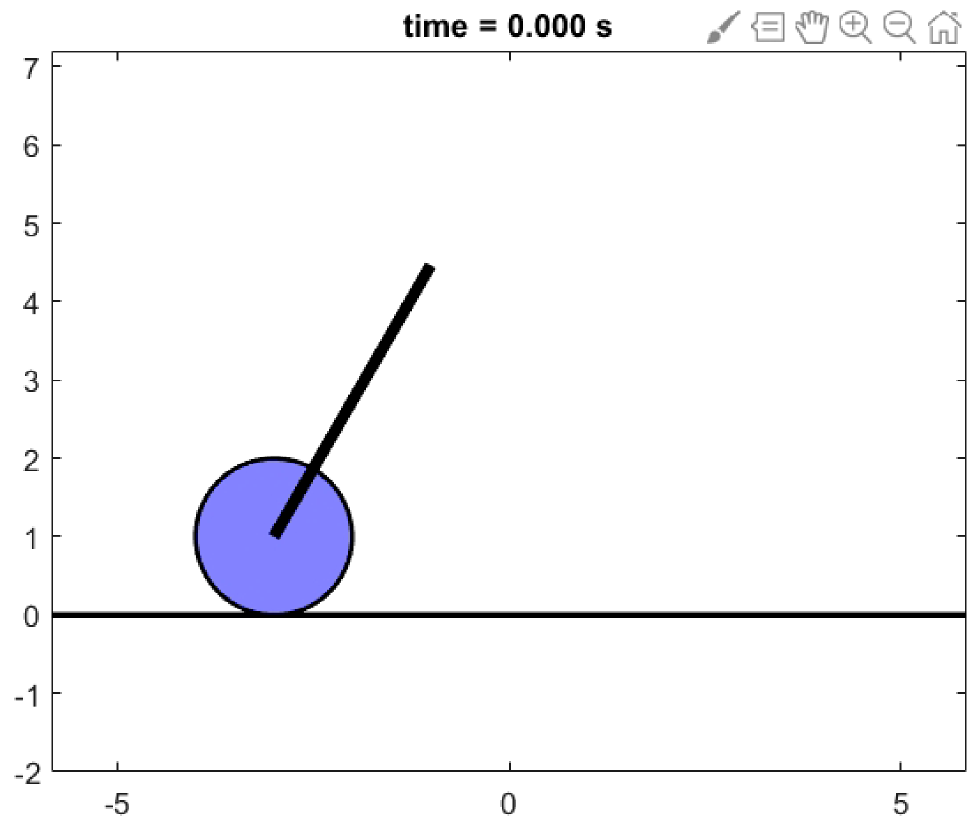
```
ans =
                  0
  -2.40772302316959
 -0.233645380020364
   2.44136840318995
```

Control input

Assigning a new set of eigen values

Find a controller K that make the closed-loop system's eigen values as specified

Assigned eigen values = [-1.3 -1.4 -1.5 -1.6]



Initial position: -3 m

Targeted position: +2 m

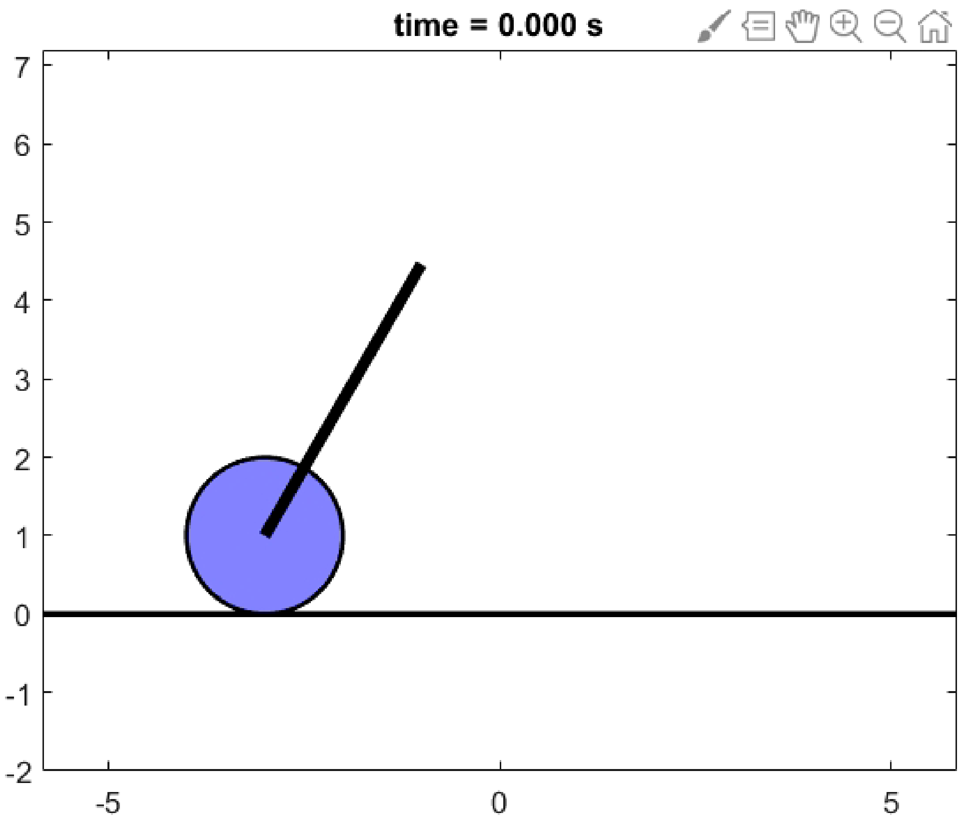
Initial angle: +30° (from +y axis)

Targeted angle: 0° (from +y axis)

Initial velocity: 0 m/s

Initial angular velocity: 0 rad/s

Assigned eigen values = [-2 -2.1 -2.2 -2.3]



Initial position: -3 m

Targeted position: +2 m

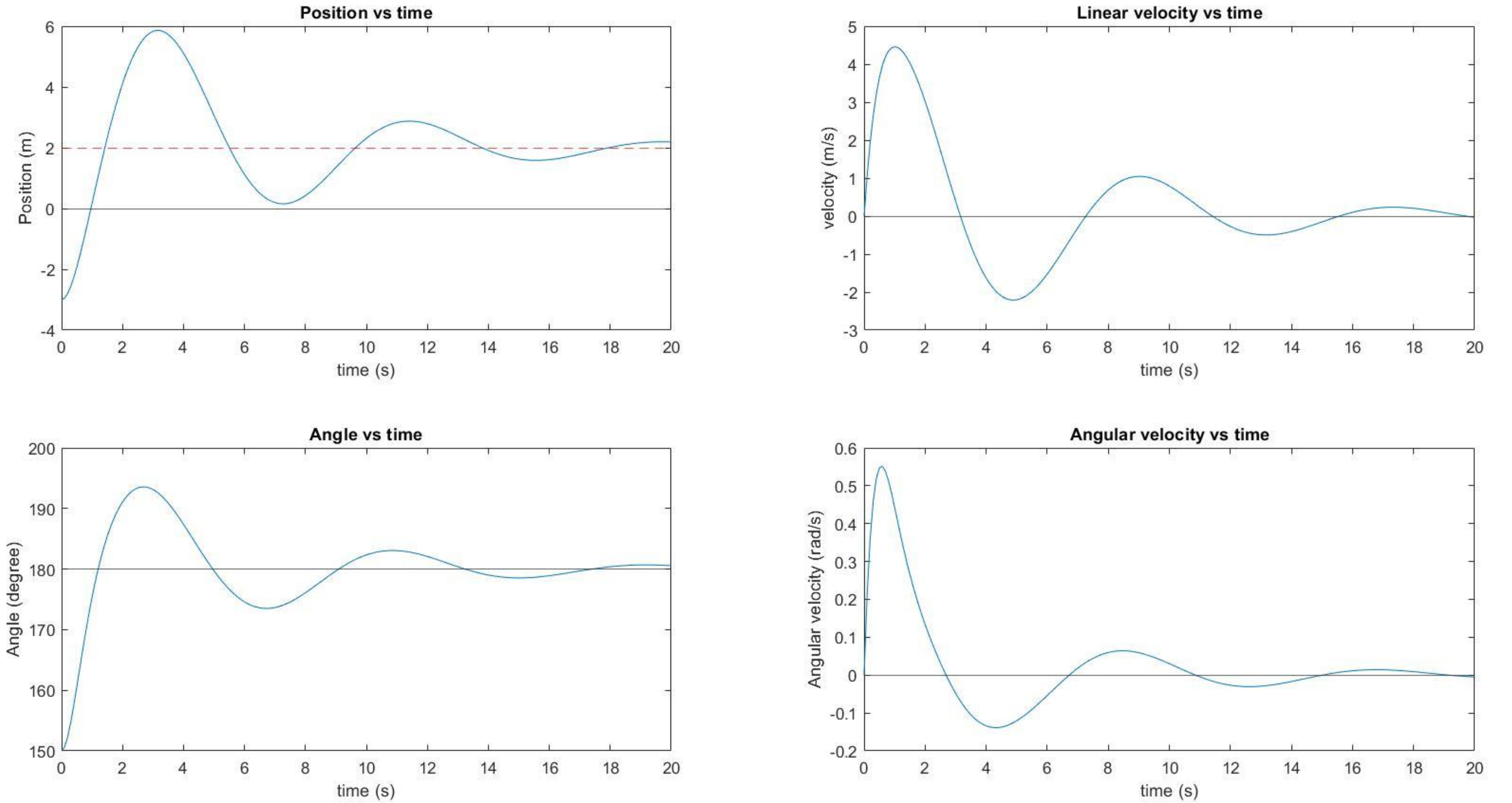
Initial angle: +30° (from +y axis)

Targeted angle: 0° (from +y axis)

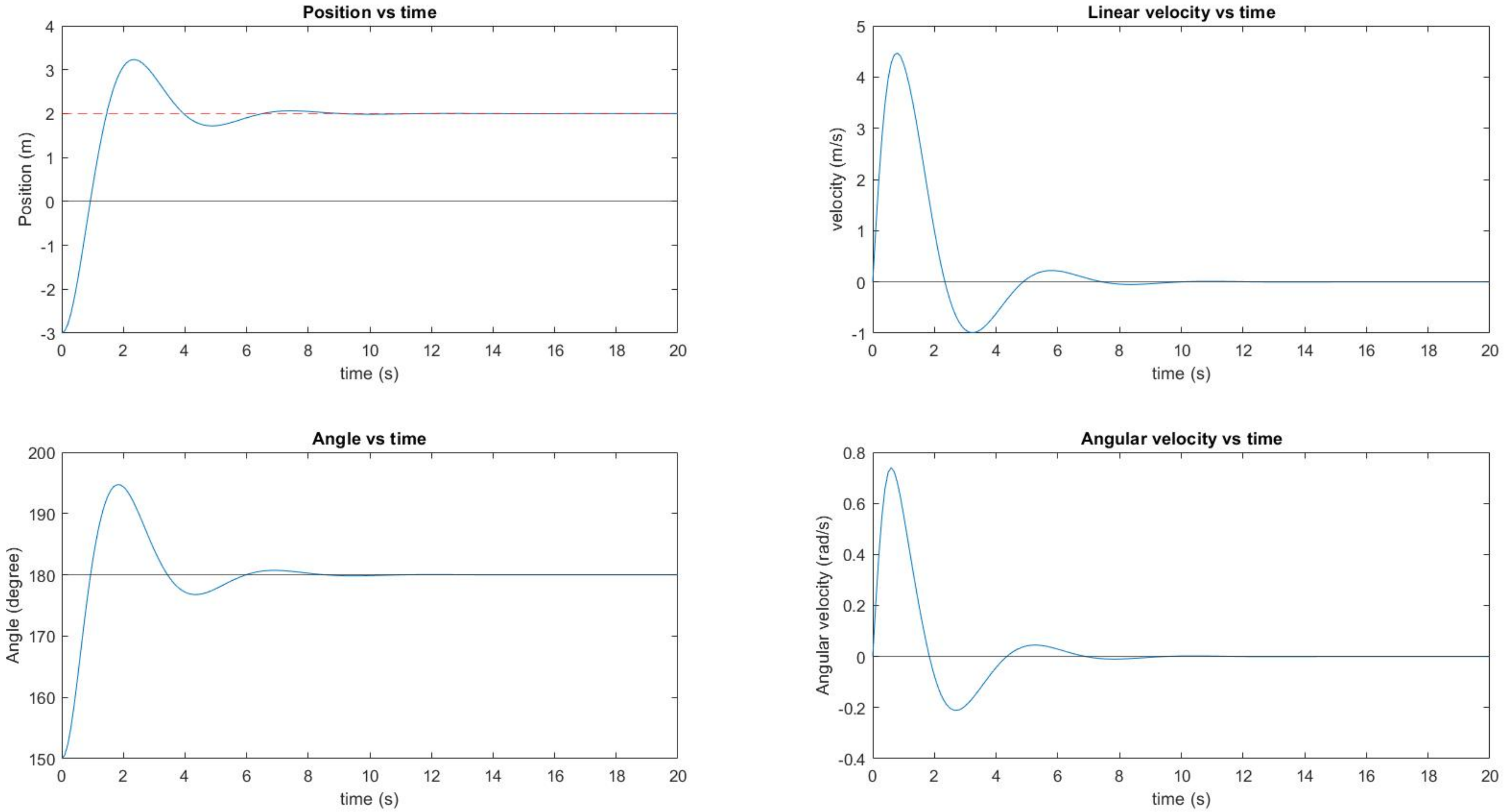
Initial velocity: 0 m/s

Initial angular velocity: 0 rad/s

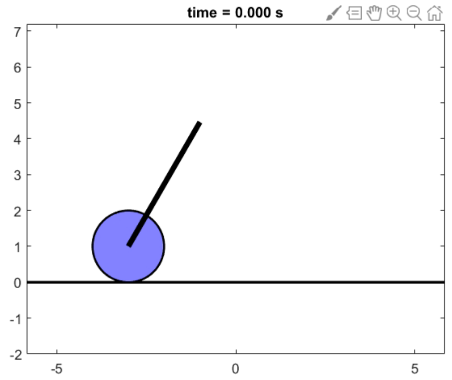
Assigned eigen values = [-1.3 -1.4 -1.5 -1.6]



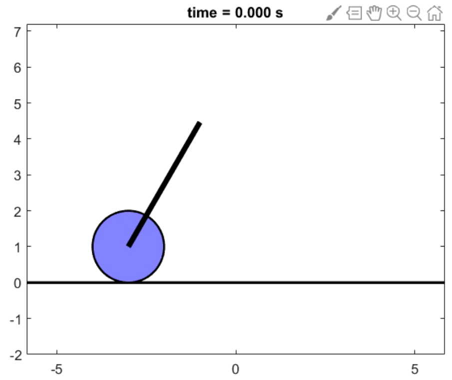
Assigned eigen values = [-2 -2.1 -2.2 -2.3]



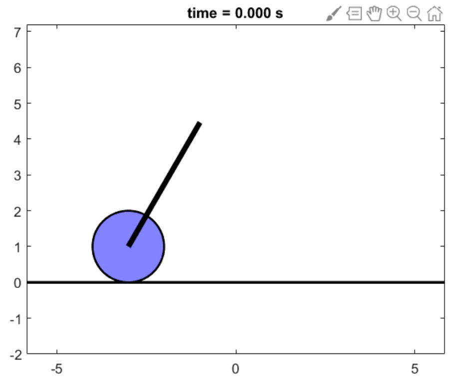
Assigned eigen values



[-1.3 -1.4 -1.5 -1.6]



[-2 -2.1 -2.2 -2.3]



[-3 -3.1 -3.2 -3.3]

References

- [(8) Inverted Pendulum on a Cart [Control Bootcamp] – YouTube]
- [(8) Pole Placement for the Inverted Pendulum on a Cart [Control Bootcamp] - YouTube]