Solution for Exercise 1: Nonlinear simulation and the linear-quadratic regulator

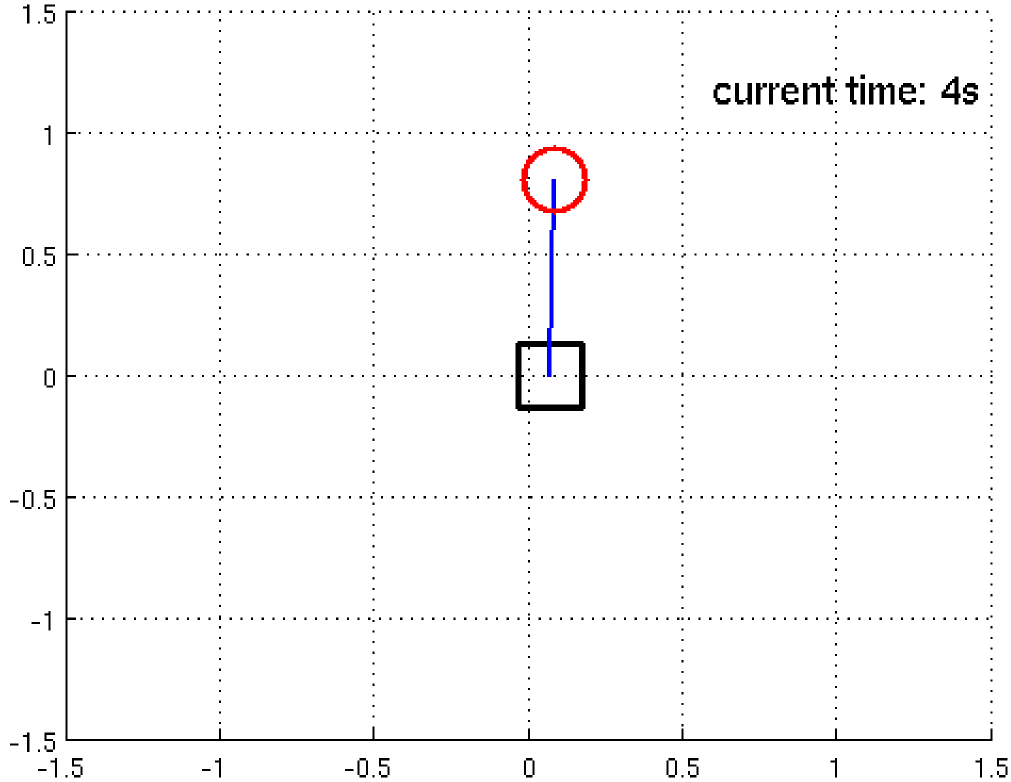
```
TEMPO Summer School on Numerical Optimal Control and Embedded Optimization
University of Freiburg, July 27 - August 7, 2015
Rien Quirynen, Dimitris Kouzoupis and Moritz Diehl
```

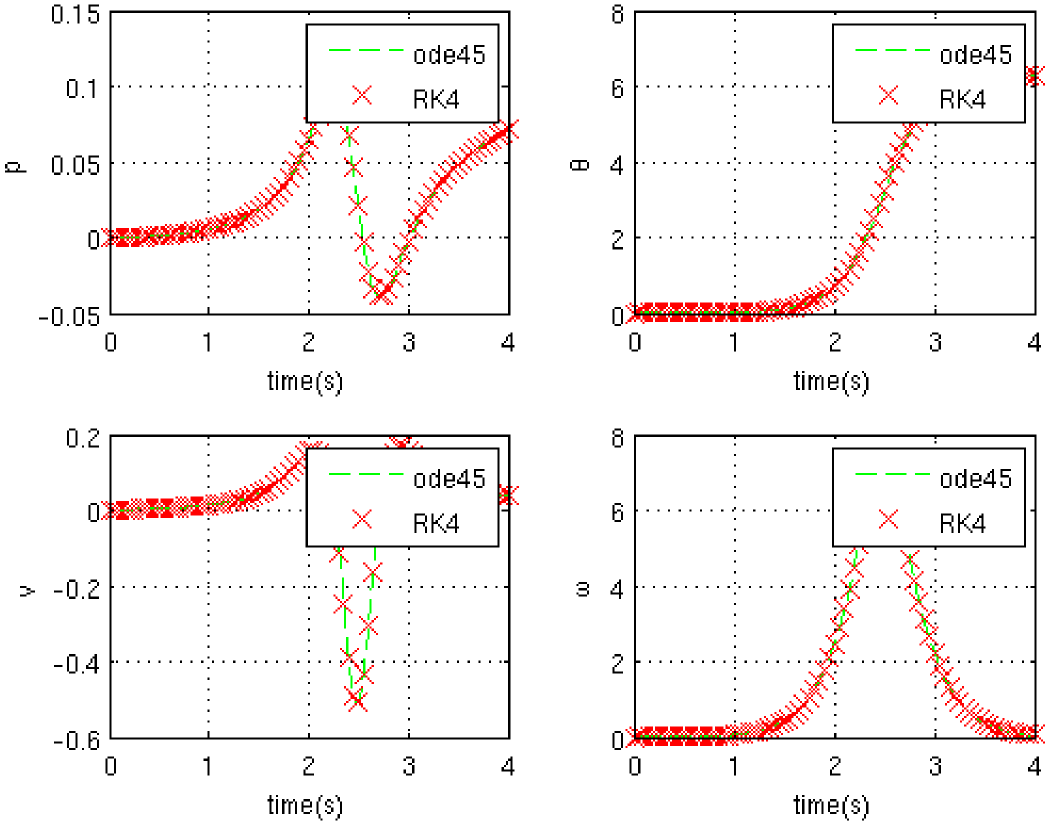
Contents

Part 1: Nonlinear simulation

```
clear all;
close all;
clc;

x0 = [0;0;0;0];
u = 0.01;
T = 4;
% Nonlinear simulation using the RK4 integrator
N = 80;
Ts = T/N;
x_rk4 = x0;
input.Ts = Ts;
input.nSteps = 2;
input.u = u;
t_rk4 = [0:N].*Ts;
for i = 1:N
    input.x = x_rk4(:,end);
    output = RK4_integrator( @ode, input );
    x_rk4(:,end+1) = output.value;
    visualize(t_rk4, x_rk4, 0.8);

%     pause(Ts/2)
end

% Comparison with ode45
[t_ode45,x_ode45] = ode45(@(t,x) ode(t,x,u), [0 T], x0);

% Plot results:
figure;
set(gcf,'Color',[1 1 1])
subplot(221);
plot(t_ode45,x_ode45(:,1),'--g'); hold on
plot(t_rk4,x_rk4(1,:),'rx','MarkerSize',10);
xlabel('time(s)')
ylabel('p')
legend('ode45', 'RK4');
grid on;

subplot(222);
plot(t_ode45,x_ode45(:,2),'--g'); hold on
plot(t_rk4,x_rk4(2,:),'rx','MarkerSize',10);
xlabel('time(s)')
ylabel('\theta')
```

```
legend('ode45', 'RK4');
grid on;

subplot(223);
plot(t_ode45,x_ode45(:,3),'--g'); hold on
plot(t_rk4,x_rk4(3,:),'rx','MarkerSize',10);
xlabel('time(s)')
ylabel('v')
legend('ode45', 'RK4');
grid on;

subplot(224);
plot(t_ode45,x_ode45(:,4),'--g'); hold on
plot(t_rk4,x_rk4(4,:),'rx','MarkerSize',10);
xlabel('time(s)')
ylabel('\omega')
legend('ode45', 'RK4');
grid on;
```





Part 2: Design of the LQR controller

Discretize and linearize the dynamic system:

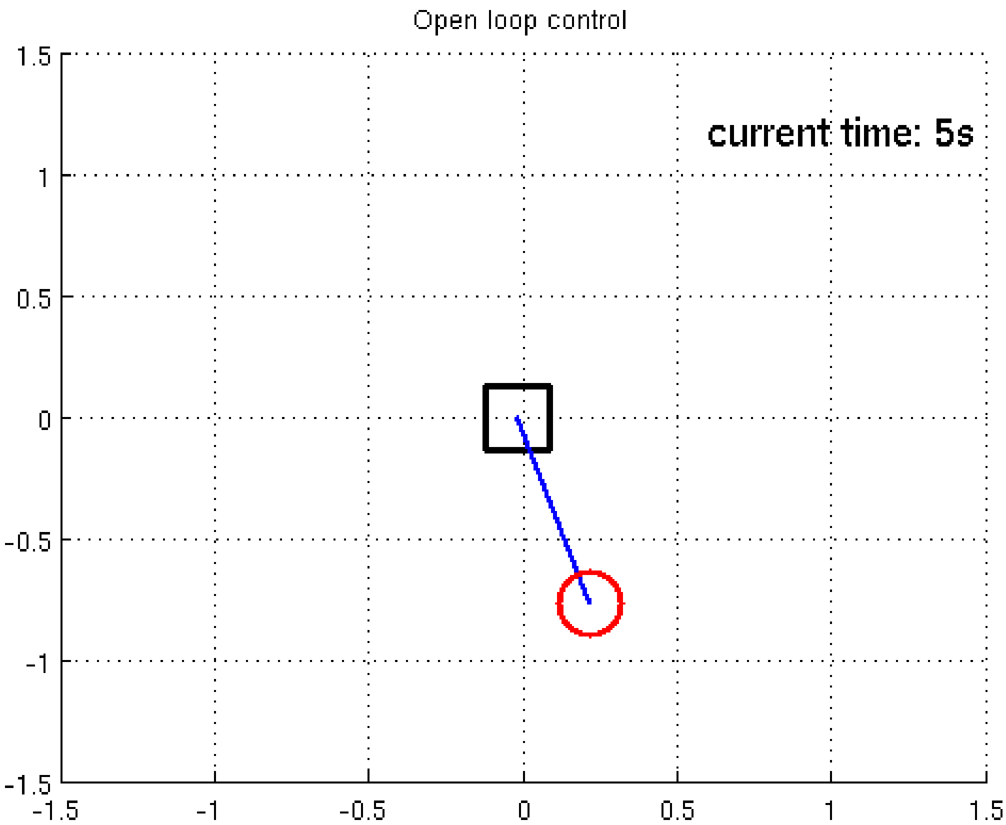
```
x_lin = zeros(4,1); u_lin = 0;
input.x = x_lin; input.u = u_lin;
output = RK4_integrator( @ode, input );
A = output.sensX;
B = output.sensU;

% Design of the LQR controller using dlqr:
Q = diag([1 1 1e-1 1e-1]); R = 1e-1;
[K,P] = dlqr(A,B,Q,R);
```

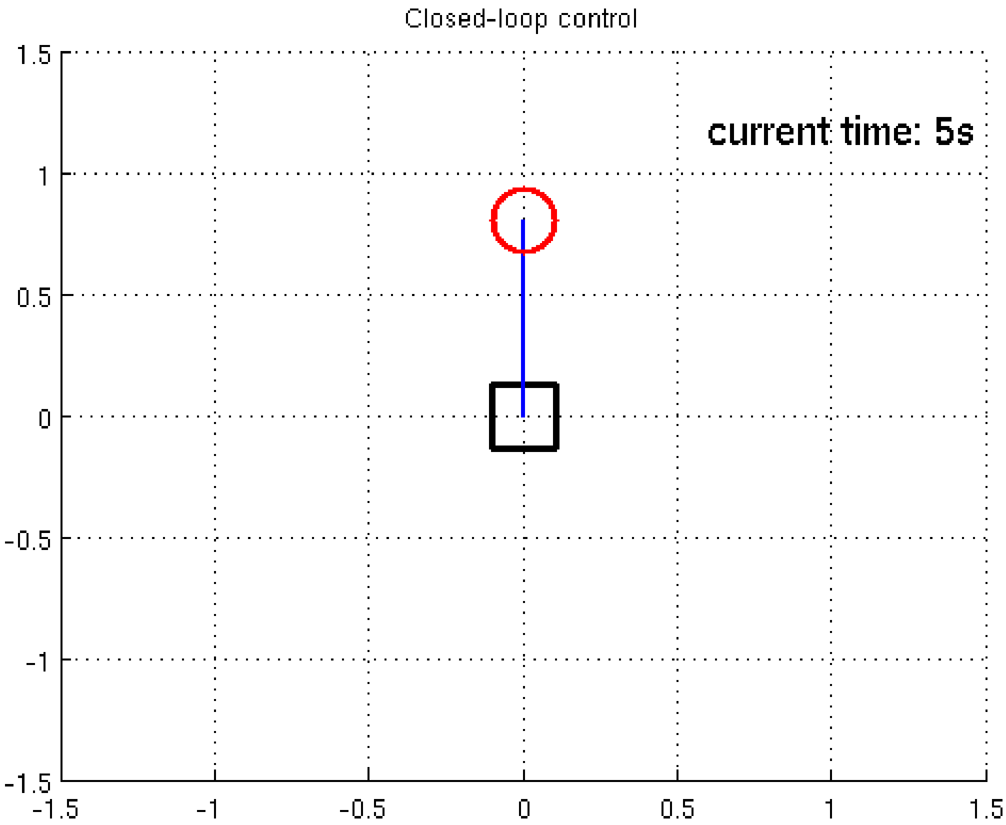
Part 3: Closed-loop simulation

Now let us perform a closed-loop simulation using the LQR controller by first defining the necessary parameters

```
l = 0.8;
x0 = [0 0.1 0 0].';
Ts = 0.05;
input.Ts = Ts;
input.nSteps = 3;

% We will perform two simulations:
% The first, applying the open loop control inputs to the nonlinear
% simulation. Afterwards, we close the loop by choosing the control input
% to be applied based on the full state feedback.

for open = [1 0]
    figure;
```

```
    time = 0;
    Tf = 5;
    state_sim = x0;
    state_lin = x0;
    iter = 0;
    while time(end) < Tf
        if open == 1
            u_LQR = -K*state_lin(:,end);
        else % closed
            u_LQR = -K*state_sim(:,end);
        end

        % apply control to nonlinear system
        input.x = state_sim(:,end);
        input.u = u_LQR;
        output = RK4_integrator( @ode, input );
        state_sim(:,end+1) = output.value;

        % linear system
        state_lin(:,end+1) = A*state_lin(:,end) + B*u_LQR;

        % next time step and visualize result
        iter = iter+1;
        time(end+1) = iter*Ts;
        visualize(time, state_sim, l);

        if open == 1
            title('Open loop control')
        else
            title('Closed-loop control')
        end

%          pause(Ts/2);
    end
end
```


Open loop control
current time: 5s
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5


Closed-loop control
current time: 5s
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5