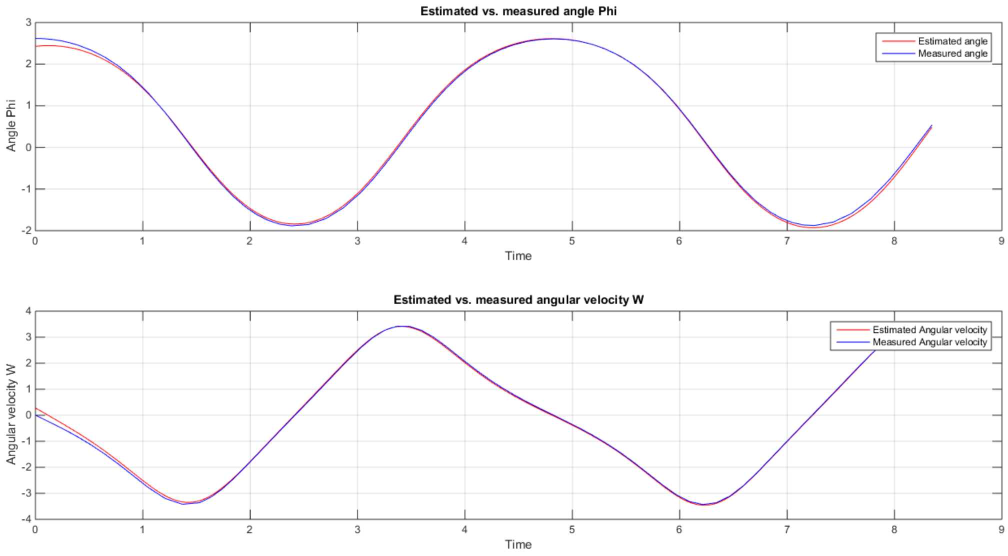
Exercises for Lecture Course on Modelling and System Identification (MSI) 1
euler_step function... 2
euler_loop function 2

Exercises for Lecture Course on Modelling and System Identification (MSI)

Exercise 7 - Task 2 Modelling and System Identification

```
clc;clear;close;
data = importdata('data7_new2.txt');
t = data(:,1);   % time [s]
phi = data(:,2); % angle [rad]
w = data(:,3);   % angular velocity [rad/s]

% Defining data
m = 1;              % mass [Kg]
L = 3;              % Rod length [m]
g = 9.81;           % Gravity [m/s^2]
Psi = pi/2;         % Handler (actuation) [rad]
deltaT = t(2)-t(1); % sample time
N = length(t);      % number of steps

theta0 = [pi/3,0,3]; % Initial guess theta = [phi,w,K]
% Estimation using lsqnonlin
[theta,resnorm,residual] = lsqnonlin(@(theta)...
    residuals_pendulum(N,m,L,g,Psi,deltaT,phi,w,theta),theta0);
phi0 = theta(1);w0 = theta(2);K = theta(3);

% Print values
formatSpec = '     %s* for non linear list square is %f \n';
C = {'Phi0','W0','K';phi0,w0,K};
disp(sprintf(formatSpec,C{:}))

% Get estimated data with optimum estimator
[phi_est, w_est] = euler_loop(phi0,w0,K,m,L,g,Psi,deltaT,N);

% Plot estimated data with optimum estimator vs measured data
figure(1);subplot(2,1,1);
plot(t,phi_est,'-r');grid on;hold on;plot(t,phi,'-b');hold off;
title('Estimated vs. measured angle Phi');xlabel('Time');
ylabel('Angle Phi');legend ('Estimated angle','Measured angle');

subplot(2,1,2);plot(t,w_est,'-r');grid on;
hold on;plot(t,w,'-b');hold off;
title('Estimated vs. measured angular velocity W')
xlabel('Time');ylabel('Angular velocity W');
legend ('Estimated Angular velocity','Measured Angular velocity');
```



euler_step function

```
function [phi_new,w_new] = euler_step(K,m,L,g,Psi,phi_prev,w_prev,deltaT)
phi_new = phi_prev + w_prev*deltaT;
w_new = w_prev + (K/(m*L^2)*(Psi-phi_prev) - g/L * sin(phi_prev))*deltaT;
```

euler_loop function

```
function [phi_k, w_k] = euler_loop(phi0,w0,K,m,L,g,Psi,deltaT,N)
phi_k = zeros(N,1);w_k = zeros(N,1);
phi_k(1) = phi0;w_k(1) = w0; % initial conditions
   for i=1+1:N
       phi_prev = phi_k(i-1);
       w_prev = w_k(i-1);
       [phi_k(i),w_k(i)] = euler_step(K,m,L,g,Psi,phi_prev,w_prev,deltaT);
   end
```

Published with MATLAB® R2014b