Quiz #6: Two Link Robot Arm Dynamics

- Run the attached code with different control of fd0 to fd11
- Show the robot pose of each of the control of fd0 to fd11

Matlab Sample Code for Robot Arm : fd0

```
function [dxdt] = fd0(t, x)
%fd forward dynamics of robot arm
%   called from ode45, input should by t and x
% Robot arm paraeters
dxdt = [x(3); x(4); -x(1); -x(2)];
end
```

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}$$

$$\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = \begin{pmatrix} -\theta_1 \\ -\theta_2 \end{pmatrix}$$

Matlab Sample Code for Robot Arm: fd1

```
function [dxdt] = fd1(t, x)
%fd forward dynamics of 2-link robot arm
%   small dumper coefficient
%   x(1) = th1;  x(2) = th2
%   x(3) = omg1; x(4) = omg2
%   tau = [0; 0]

% Gravity parameter
g = 9.8;

% Robot arm parameters
m1 = 1.0; m2 = 1.0;
l1 = 1.0; l2 = 1.0; lg1 = 0.5; lg2 = 0.5;
d1 = 0.01; d2 = 0.01;
I1 = 1/12 * m1 * l1.^2;
I2 = 1/12 * m2 * l2.^2;
```

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}$$

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_1 \\ \tau_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$I_1 = \frac{1}{12} m_1 l_1^2, I_2 = \frac{1}{12} m_2 l_2^2$$

Matlab Sample Code for Robot Arm : fd1

```
% Joint torque
Tau = [0; 0];

% Differential set equation
omg_d = inv(M)*(Tau - H - G);
dxdt = [x(3); x(4); omg_d(1); omg_d(2)];
end
```

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}$$

$$\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = M(\boldsymbol{\theta})^{-1} \left(\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} - \begin{pmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ h_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} - \begin{pmatrix} g_1(\boldsymbol{\theta}) \\ g_2(\boldsymbol{\theta}) \end{pmatrix} \right)$$

Matlab Sample Code for Robot Arm : fd1

```
% Joint torque
Tau = [0; 0];

% Differential set equation
omg_d = inv(M)*(Tau - H - G);
dxdt = [x(3); x(4); omg_d(1); omg_d(2)];
end
```

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}$$

$$\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = M(\boldsymbol{\theta})^{-1} \left(\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} - \begin{pmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ h_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} - \begin{pmatrix} g_1(\boldsymbol{\theta}) \\ g_2(\boldsymbol{\theta}) \end{pmatrix} \right)$$

Matlab Sample Code for Robot Arm : dynamics (main: ode45 side)

```
%   Dynamics 2link arm
%       Author: Keitaro Naruse
%       Date: 2019-06-4

% Solve differential equation of equations of robot motion
% fd1: small dumping coefficent d1, d2 = 0.01
[t,x] = ode45(@fd1, [0, 10],[0.1; 0.1; 0; 0]);

% Plot th1 = x(1) and th(2) = x(2)
figure(1);
plot(t, x(:,1), 'r-', t, x(:,2), 'b-');
title('th1(red) and th2(blue)');

% Plot omg1 = x(4) and omg2 = x(2)
figure(2);
plot(t, x(:,3), 'r-', t, x(:,4), 'b-');
title('omg1(red) and omg2(blue)');
```

P-control for Joint Angle
（関節角度に対するP制御）

Suppose control a robot arm to a given pose, how do we apply a torque for it?
（ロボットアームの姿勢が与えられたときに，どのようにトルクを決定するか）

$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix}$$

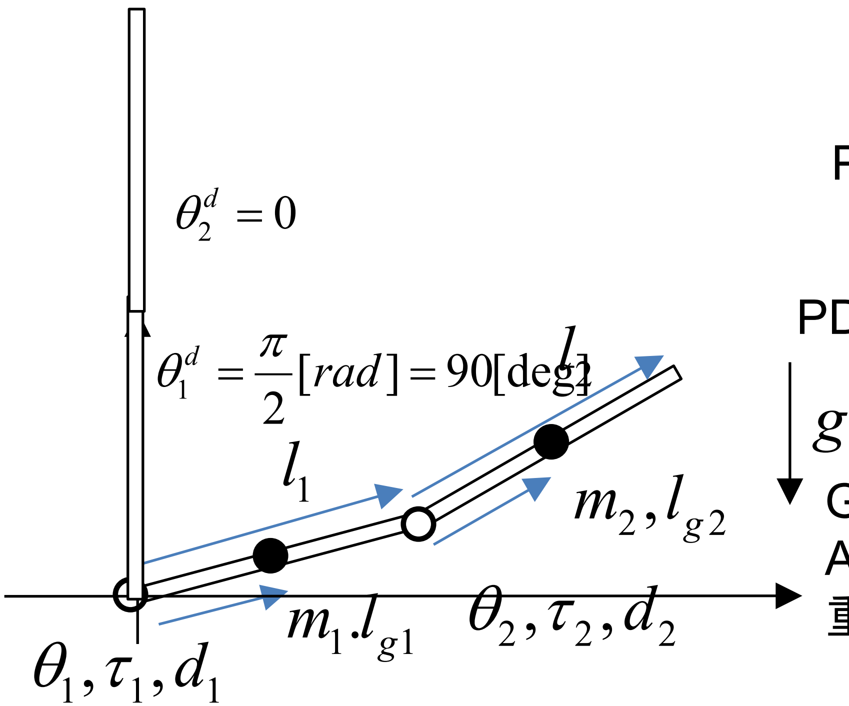
$$\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = M(\boldsymbol{\theta})^{-1} \left(\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} - \begin{pmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ h_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} - \begin{pmatrix} g_1(\boldsymbol{\theta}) \\ g_2(\boldsymbol{\theta}) \end{pmatrix} \right)$$

One of the solutions is feedback control（方法の一つはフィードバック制御）

The simplest method is P-control: Apply a torque proportional to the difference between a target and current angle
（もっとも簡単な手法はP制御: 目標角度と現在角度の差に比例的にトルクをかける）

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} k_p(\theta_1^d - \theta_{1,t}) \\ k_p(\theta_2^d - \theta_{2,t}) \end{pmatrix}$$

Control of arm and gravity compensation
アームの制御と重力補償



Position control: 位置制御
Apply a force proportional to error to target 目標までの誤差に比例的に力をかける

P-control

$$\boldsymbol{\tau}_t = \begin{pmatrix} \boldsymbol{\tau}_{1,t} \\ \boldsymbol{\tau}_{2,t} \end{pmatrix} = \begin{pmatrix} k_{p1}\left(\theta_{1d} - \theta_{1,t}\right) \\ k_{p2}\left(\theta_{2d} - \theta_{2,t}\right) \end{pmatrix}$$

PD-control

$$\boldsymbol{\tau}_t = \begin{pmatrix} \boldsymbol{\tau}_{1,t} \\ \boldsymbol{\tau}_{2,t} \end{pmatrix} = \begin{pmatrix} k_{p1}\left(\theta_{1d} - \theta_{1,t}\right) - k_{d1}\dot{\theta}_{1,t} \\ k_{p2}\left(\theta_{2d} - \theta_{2,t}\right) - k_{d2}\dot{\theta}_{2,t} \end{pmatrix}$$

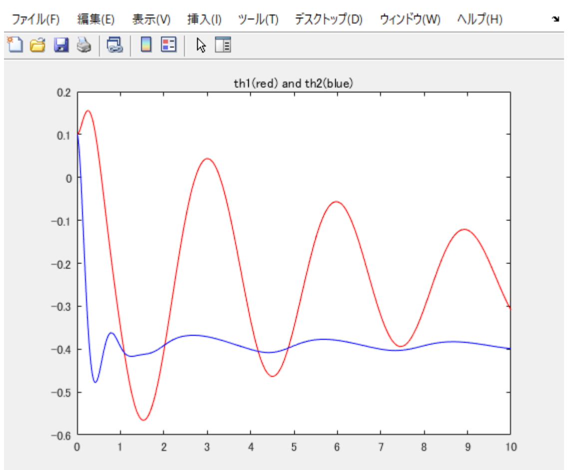
Gravity compensation: 重力補償
Add an extra force equivalent to gravity term
重力に起因する力の分だけ余分に力をかける

$$\boldsymbol{\tau}_t = \begin{pmatrix} \boldsymbol{\tau}_{1,t} \\ \boldsymbol{\tau}_{2,t} \end{pmatrix} = \begin{pmatrix} k_{p1}\left(\theta_{1d} - \theta_{1t}\right) + g_1(\theta) \\ k_{p2}\left(\theta_{1d} - \theta_{12}\right) + g_2(\theta) \end{pmatrix}$$

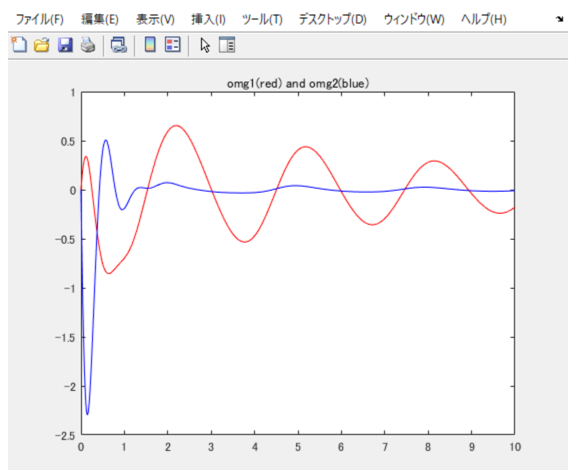
P-control for Joint Angle
（関節角度に対するP制御）

```
% Joint torque
%   Target angle x(1) = pi/2; x(2) = 0;
kp1 = 10; kp2 = 10;
xd1 = pi/2; xd2 = 0;
Tau = [kp1 * (xd1 - x(1));...
    kp2 * (xd2 - x(2))];
```

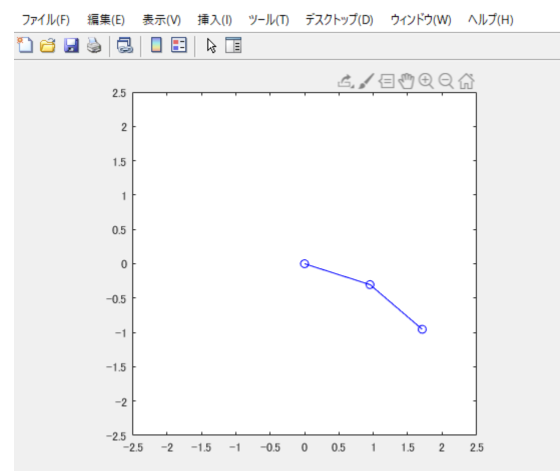
Not enough torque to stand up
-> Gravity compensation is needed
（トルク不足，重力の分を補う必要がある）



Joint angles
（関節角度）



Joint angular velocities
（関節角速度）



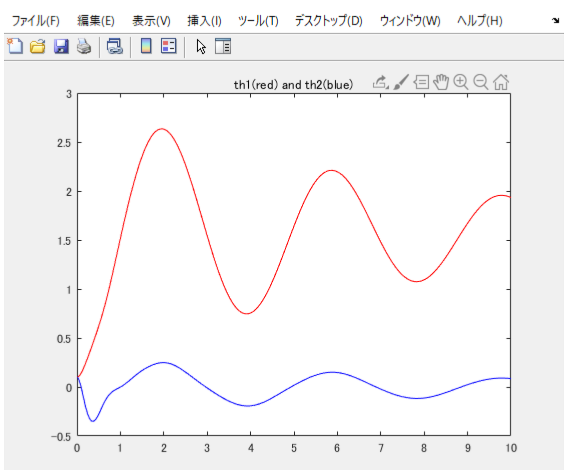
Arm pose
（アームの姿勢）

P-control with Gravity Compensation（重力補償ありのP制御）

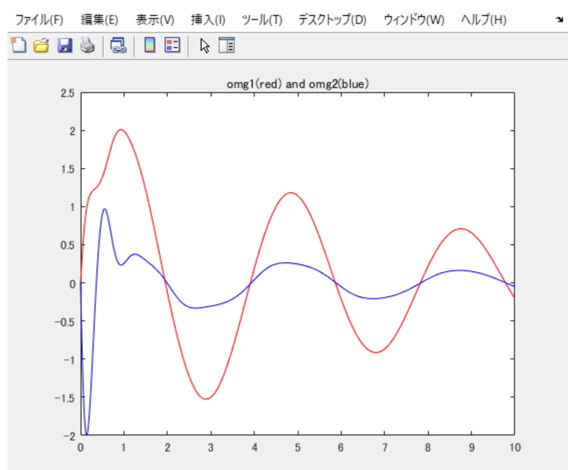
```
% Joint torque
%   Target angle x(1) = pi/2; x(2) = 0;
kp1 = 10; kp2 = 10;
xd1 = pi/2; xd2 = 0;
Tau = [kp1 * (xd1 - x(1)) + G(1);...
    kp2 * (xd2 - x(2)) + G(2)];
```

Torque is enough, but oscillation（トルクは十分だが振動）

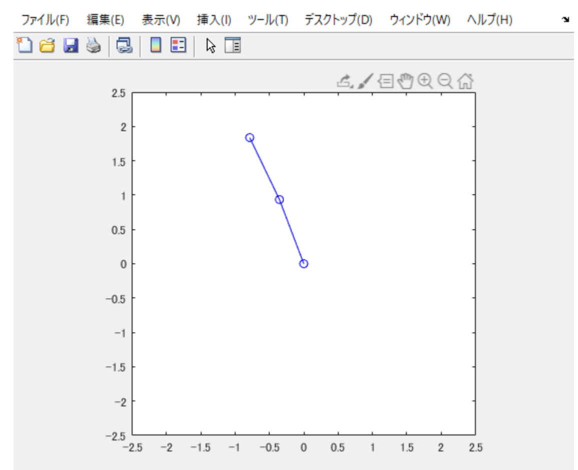
We introduce zero-velocity norm to control（制御に速度0の基準を導入する）



Joint angles（関節角度）



Joint angular velocities（関節角速度）

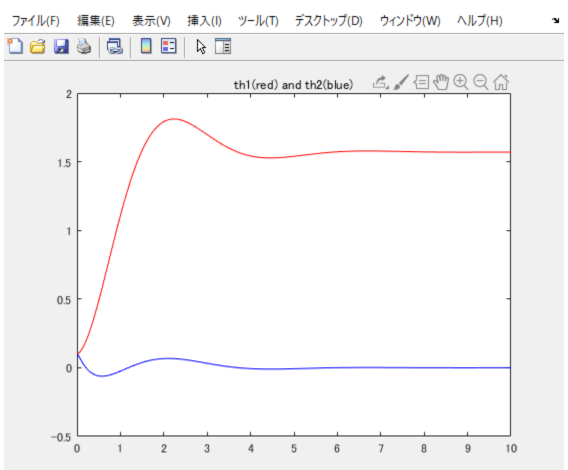


Arm pose（アームの姿勢）

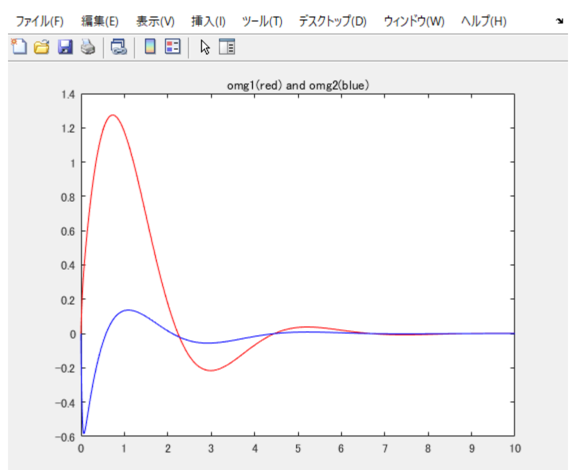
PD-control with Gravity Compensation（重力補償ありのPD制御）

```
% Joint torque
%   Target angle x(1) = pi/2; x(2) = 0;
kp1 = 10; kp2 = 10;
kd1 = 5; kd2 = 5;
xd1 = pi/2; xd2 = 0;
Tau = [kp1*(xd1 - x(1)) + kd1*(-x(3)) + G(1);...
    kp2*(xd2 - x(2)) + kd2*(-x(4)) + G(2)];
```

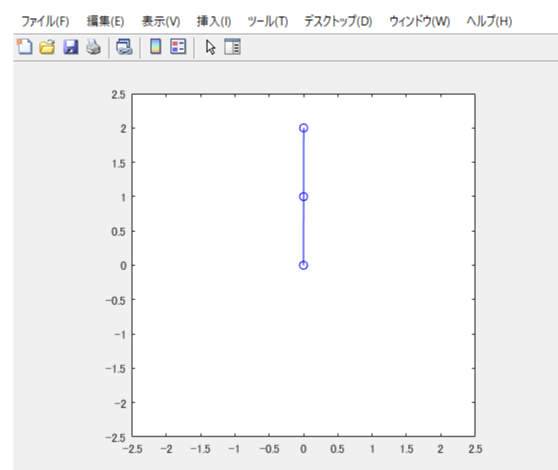
Fine, stayed at a target position
（目標位置に静止，問題なし）
We often introduce PD-control with gravity compensation for robot dynamics control
(重力補償ありのPD制御はよく使われる)



Joint angles
（関節角度）



Joint angular velocities
（関節角速度）



Arm pose
（アームの姿勢）