Robot Arm Dynamics

ICT03: Advanced Robotics

Keitaro Naruse

Email: naruse@u-aizu.ac.jp

Robot Arm Control

Generalized equation of motion for robots
ロボットの運動方程式の一般形

$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta})$$

$\boldsymbol{\tau}$: Generalized force(torque) vector 一般化力ベクトル

$\boldsymbol{\theta}$: Generalized position(angule) vector 一般化位置ベクトル

$\dot{\boldsymbol{\theta}}$: Generalized velocity vector 一般化速度ベクトル

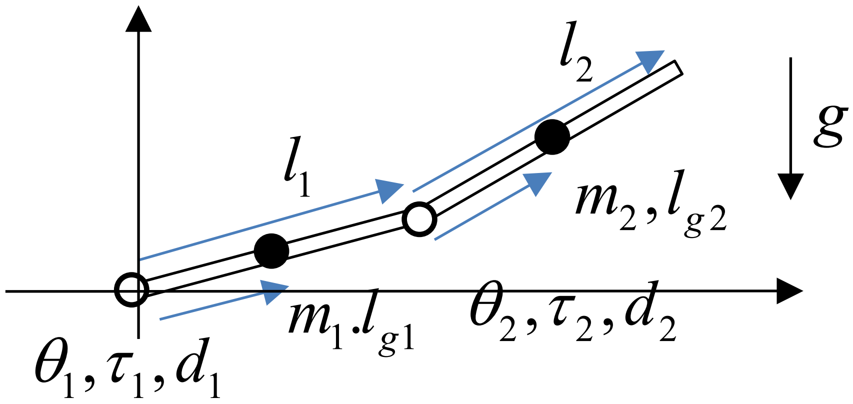
$\ddot{\boldsymbol{\theta}}$: Generalized acceleration vector 一般化加速度ベクトル

$M(\boldsymbol{\theta})$: Inertial matrix 慣性行列

$\boldsymbol{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$: Non linear term vector 非線形項ベクトル
such as Coriolis, centrifugal, damper force

$\boldsymbol{g}(\boldsymbol{\theta})$: Gravity force vector 重力項ベクトル

Equation of motion for 2-link arm
2リンクアームの運動方程式の例



$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta})$$

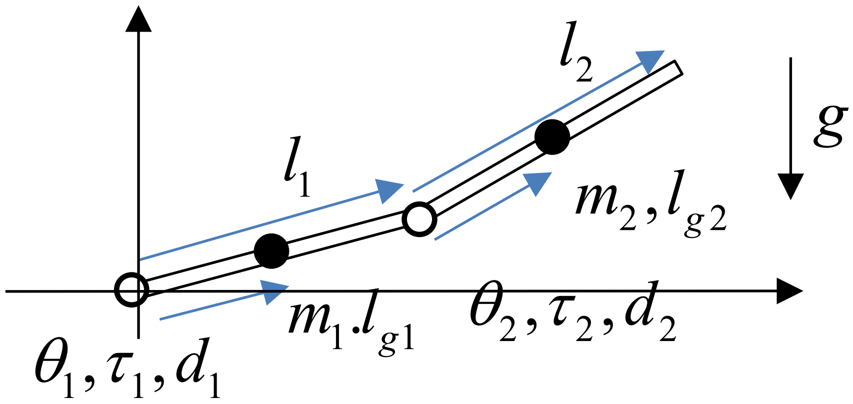
$$M(\boldsymbol{\theta}) = \begin{pmatrix} I_1 + I_2 + m_1 l_{g1}^2 + m_2 l_1^2 + m_2 l_{g2}^2 + 2 m_2 l_1 l_{g2} \cos\theta_2 & I_2 + m_2 l_{g2}^2 + m_2 l_1 l_{g2} \cos\theta_2 \\ I_2 + m_2 l_{g2}^2 + m_2 l_1 l_{g2} \cos\theta_2 & I_2 + m_2 l_{g2}^2 \end{pmatrix}$$

$$\boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \begin{pmatrix} -m_2 l_1 l_{g2} (2\dot{\theta}_1 \dot{\theta}_2 + \dot{\theta}_2^2) \sin\theta_2 + d_1 \dot{\theta}_1 \\ m_2 l_1 l_{g2} \dot{\theta}_1^2 \sin\theta_2 + d_2 \dot{\theta}_2 \end{pmatrix}$$

$$\boldsymbol{g}(\boldsymbol{\theta}) = \begin{pmatrix} m_1 g l_{g1} \cos\theta_1 + m_2 g l_1 \cos\theta_1 + m_2 g l_{g2} \cos(\theta_1 + \theta_2) \\ m_2 g l_{g2} \cos(\theta_1 + \theta_2) \end{pmatrix}$$

$$\boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{pmatrix}, \boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, I_1, I_2 : \text{Intertia of link}$$

Simulation as forward dynamics
順動力学としてのシミュレーション



Forward dynamics 順動力学

- Given: Generalized force
- Find: Generalized position, velocity
- as simulation

力を与えて位置と速度を求める
→シミュレーション

Given: $\boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{pmatrix},$

Find: $\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \dot{\boldsymbol{\theta}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}$

Subject to:

$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta})$$

$$\ddot{\boldsymbol{\theta}} = M(\boldsymbol{\theta})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \boldsymbol{g}(\boldsymbol{\theta})\right)$$

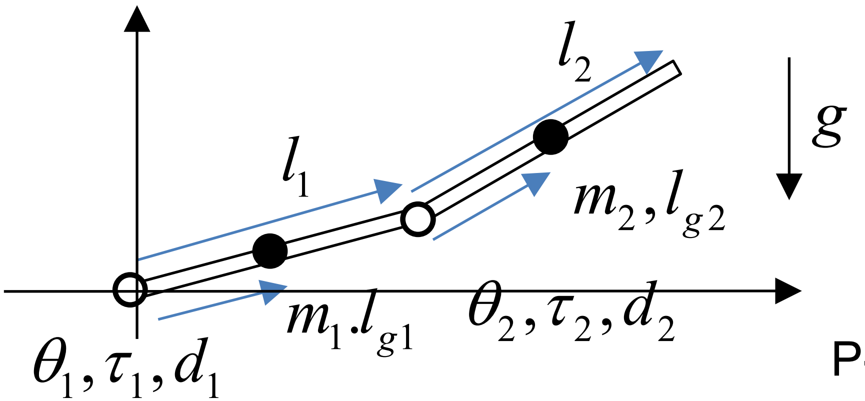
Solve: second order differential equation
as first order one

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

$$\frac{d}{dt}\begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\omega} \end{pmatrix} = \begin{pmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{\omega}} \end{pmatrix} = \underline{\begin{pmatrix} \boldsymbol{\omega} \\ M(\boldsymbol{\theta})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \boldsymbol{g}(\boldsymbol{\theta})\right) \end{pmatrix}}$$

Find $\boldsymbol{\theta}$ and $\boldsymbol{\omega}$ by solving with ode45 in matlab
Matlabで数値計算

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

Given: $\boldsymbol{\tau} = \begin{pmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{pmatrix}$,

Find: $\boldsymbol{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$,

Subject to:
$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta})$$
$$\ddot{\boldsymbol{\theta}} = M(\boldsymbol{\theta})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) - \boldsymbol{g}(\boldsymbol{\theta})\right)$$

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

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

How to Simulate Dynamics in matlab
(Matlabでの動力学のシミュレーションの方法)

Solve a Differential Equation Numerically in Matlab（Matlabで数値的に微分方程式を解く）

```
[t,y] = ode45(odefun,tspan,y0)
```

ODE: Ordinary Differential Equation（常微分方程式）

odefun: function hudnle of solved differential equations
（微分方程式を表した関数）

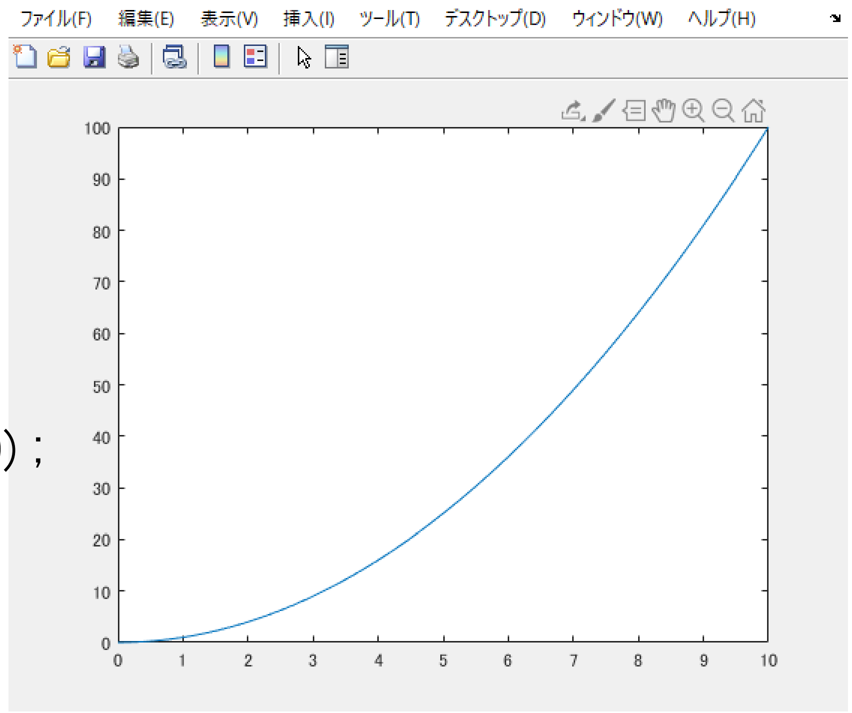
tspan: time span [t0, tf]
（時間区間）

y0: initial value vector
（初期状態ベクトル）

Simple Example of ODE45
（ODE45の使い方の簡単な例）

$$\dot{y} = \frac{dy}{dt} = 2t$$

```
tspan = [0 10];
y0 = 0;
[t,y] = ode45(@simple_ode, tspan, y0);
plot(t, y);

function dydt = simple_ode(t,y)
dydt = 2 * t;
end
```



Transfer from 2nd Order ODE to 1st Order ODE（二階常微分方程式から一階常微分方程式へ変形）

A single second-order ordinary differential equation
（一つの二階常微分方程式）

Multiple first-order ordinary differential equation
（多変数の一階常微分方程式）

$$\boldsymbol{\tau} = M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) + \boldsymbol{g}(\boldsymbol{\theta})$$

$$\ddot{\boldsymbol{\theta}} = M(\boldsymbol{\theta})^{-1}\left(\boldsymbol{\tau} - \boldsymbol{h}(\boldsymbol{\theta},\dot{\boldsymbol{\theta}}) - \boldsymbol{g}(\boldsymbol{\theta})\right)$$

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

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

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

Matlab Sample Code for Robot Arm: fd1

```
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;
```

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

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

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

Matlab Sample Code for Robot Arm : 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

```
% 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
```

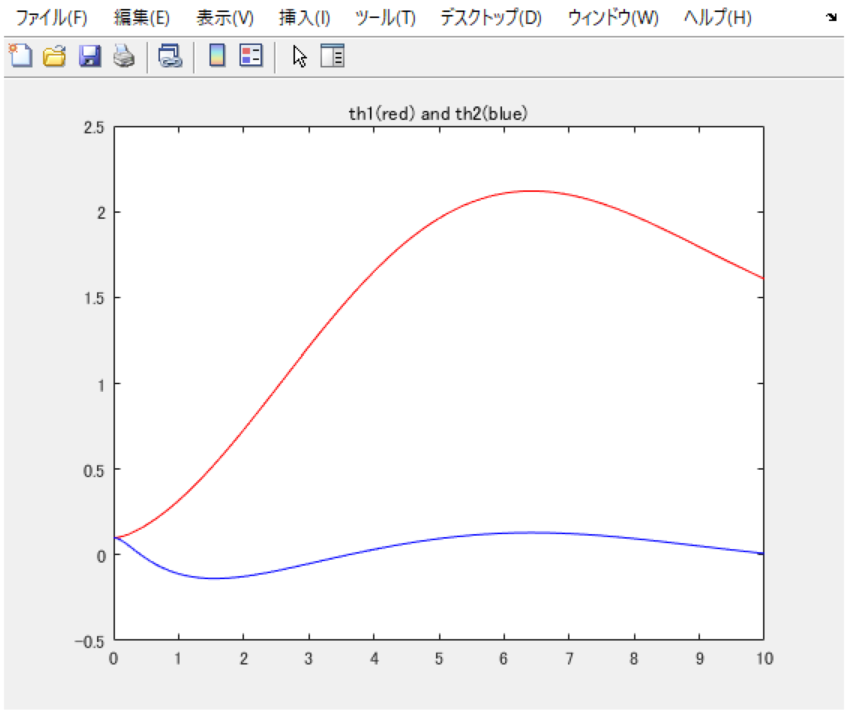
$$\boldsymbol{x} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{pmatrix}, \dot{\boldsymbol{x}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\omega}_1 \\ \dot{\omega}_2 \end{pmatrix}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

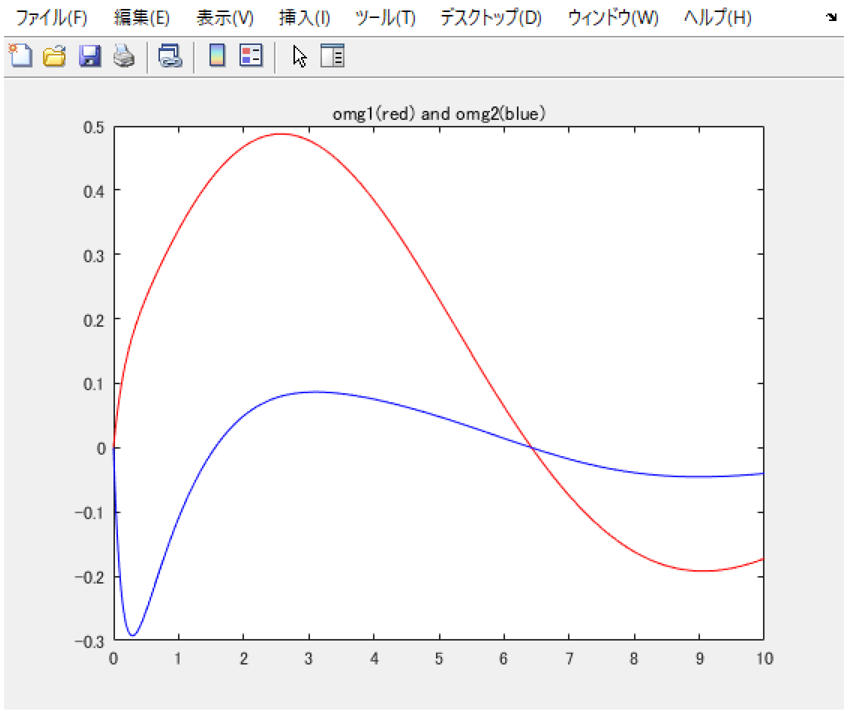
$$\begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} = M(\boldsymbol{\theta})^{-1} \left(\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} - \begin{pmatrix} h_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \\ h_2(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \end{pmatrix} - \begin{pmatrix}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

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)');
```

Results





Joint angles
（関節角度）

Joint angular velocities
（関節角速度）

Matlab Code for Pose Animation

```
l1 = 1.0; l2 = 1.0;
p0 = [0; 0];
p1 = zeros(2, length(t));
p2 = zeros(2, length(t));
figure(3);
for k=1:length(t)
    p1(:,k) = [l1*cos(x(k,1)); l1*sin(x(k,1))];
    p2(:,k) = p1(:,k)+[l2*cos(x(k,1)+x(k,2)); l2*sin(x(k,1)+x(k,2))];
    px = [p0(1), p1(1,k), p2(1,k)];
    py = [p0(2), p1(2,k), p2(2,k)];
    plot(px, py, 'b-o');
    axis equal;
    xlim([-2.5 2.5]);
    ylim([-2.5 2.5]);
    pause(1/100);
end
```

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）

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

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

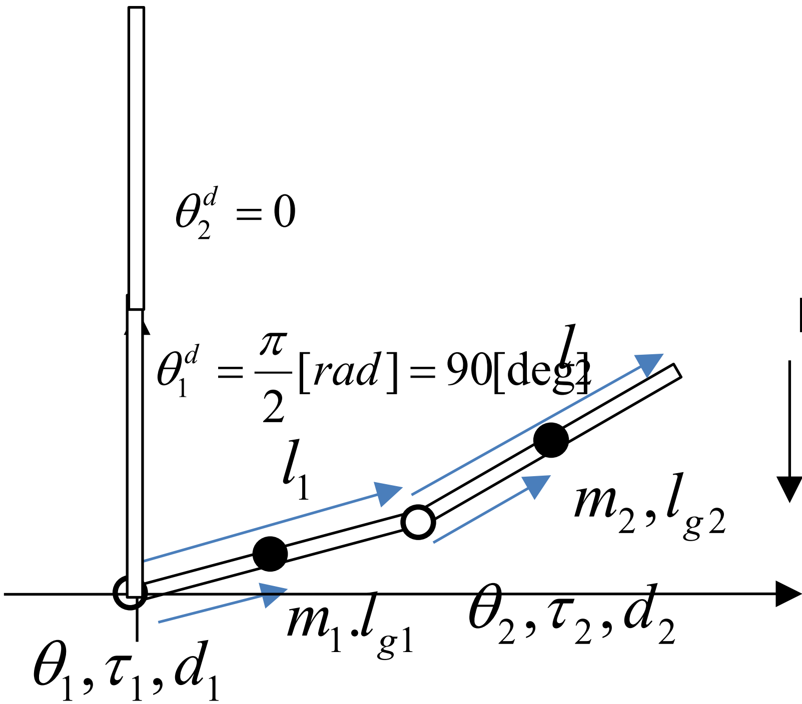
One of the solutions is feedback control（方法の一つはフィードバック制御）

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

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

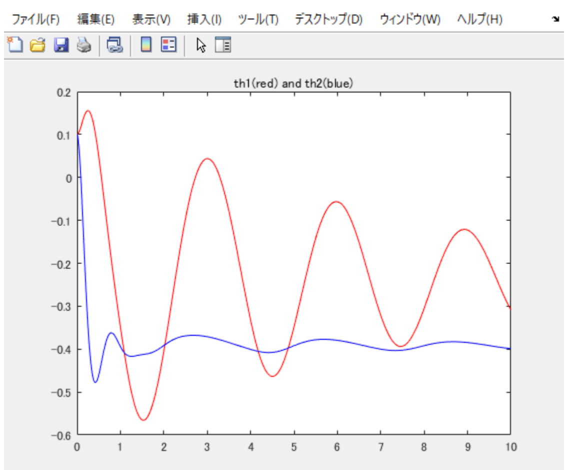
Gravity compensation: 重力補償
Add an extra force equivalent to gravity term
重力に起因する力の分だけ余分に力をかける

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

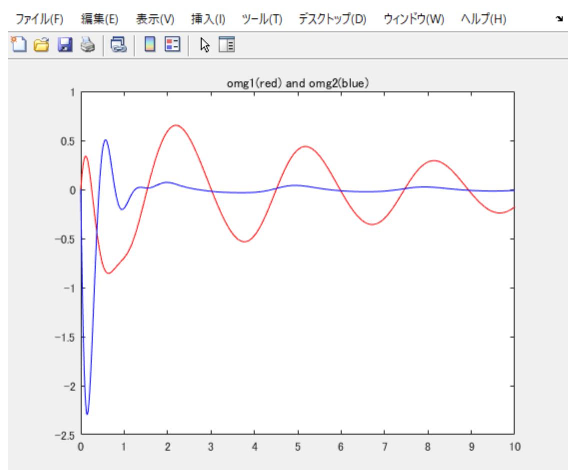
P-control for Joint Angle
（関節角度に対するP制御）

```
% 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))];
```

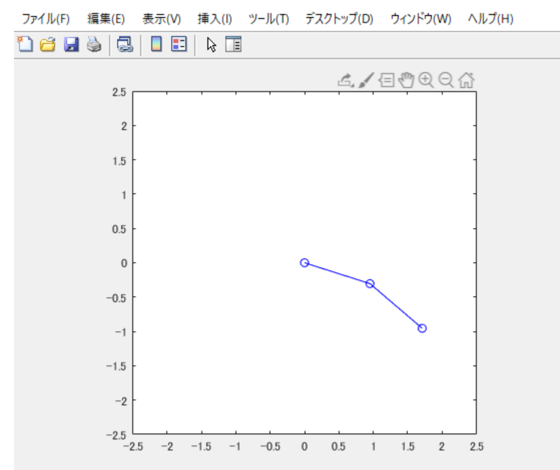
Not enough torque to stand up
-> Gravity compensation is needed
（トルク不足，重力の分を補う必要がある）



Joint angles
（関節角度）



Joint angular velocities
（関節角速度）



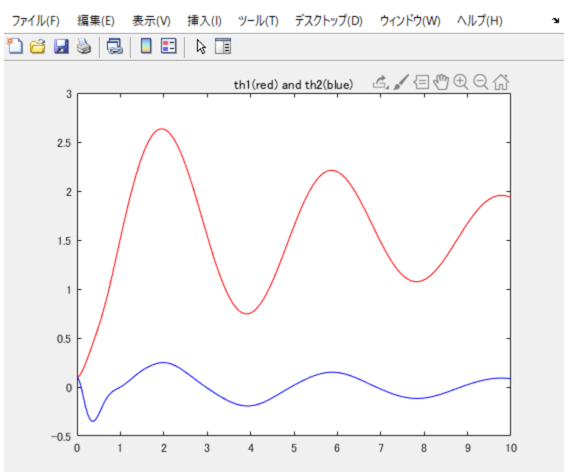
Arm pose
（アームの姿勢）

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

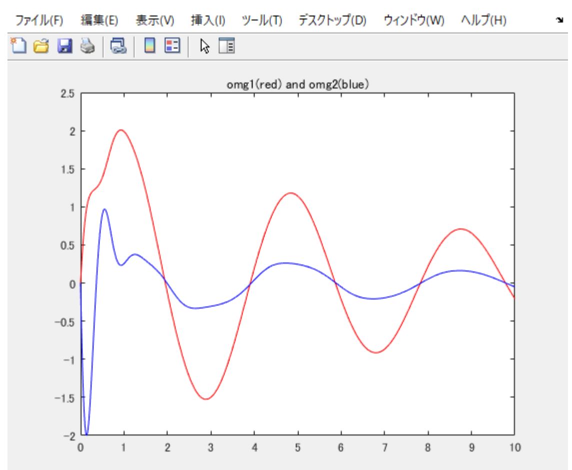
```
% Joint torque
%   Target angle x(1) = pi/2; x(2) = 0;
kp1 = 10; kp2 = 10;
xd1 = pi/2; xd2 = 0;
Tau = [kp1 * (xd1 - x(1)) + G(1);...
    kp2 * (xd2 - x(2)) + G(2)];
```

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

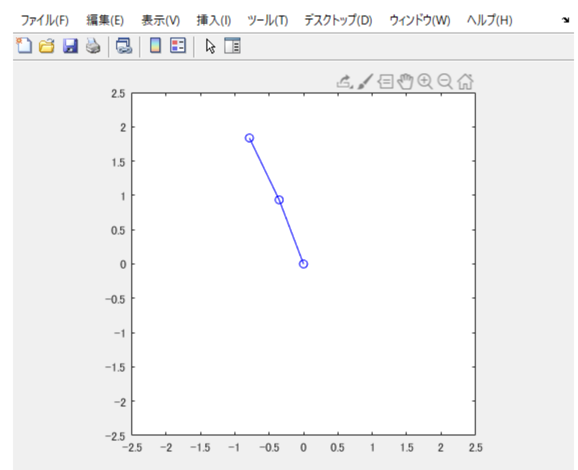
We introduce zero-velocity norm to control（制御に速度0の基準を導入する）



Joint angles
（関節角度）



Joint angular velocities
（関節角速度）

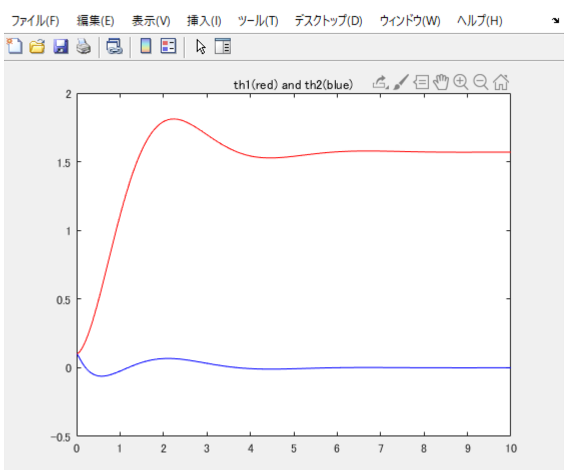


Arm pose
（アームの姿勢）

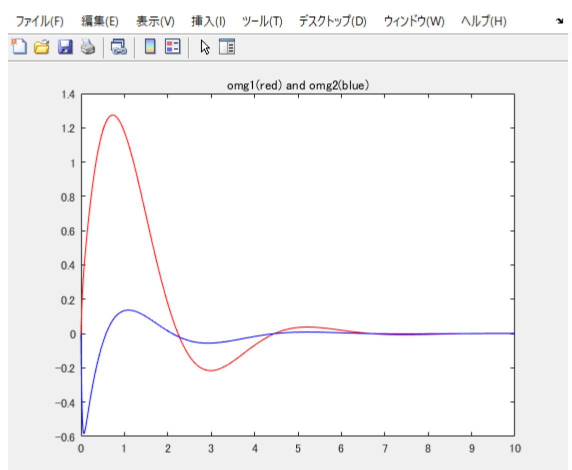
PD-control with Gravity Compensation（重力補償ありのPD制御）

```
% 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)];
```

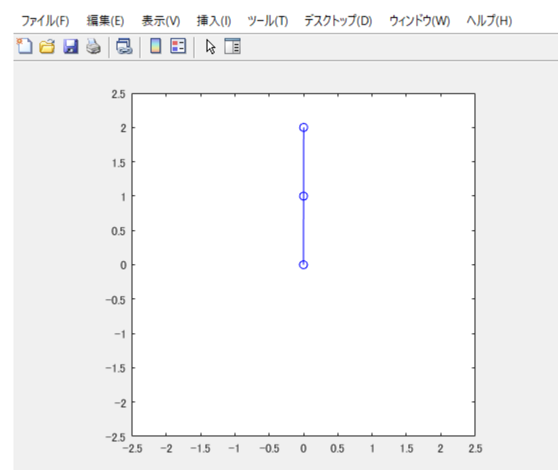
Fine, stayed at a target position
（目標位置に静止，問題なし）
We often introduce PD-control with gravity compensation for robot dynamics control
(重力補償ありのPD制御はよく使われる)



Joint angles
（関節角度）



Joint angular velocities
（関節角速度）



Arm pose
（アームの姿勢）

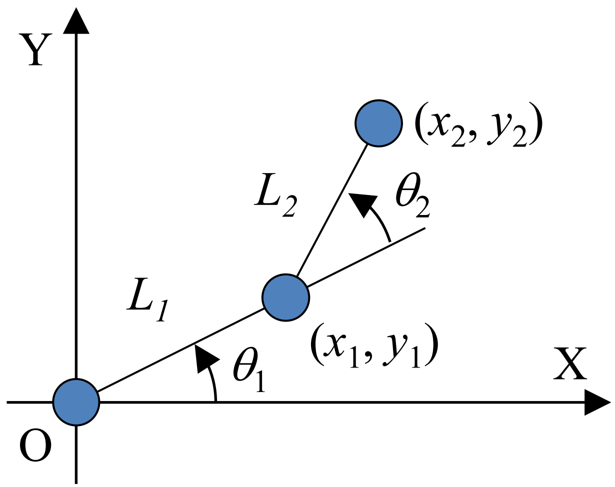
Comments on Dynamics Control
（動力学制御に関するコメント）

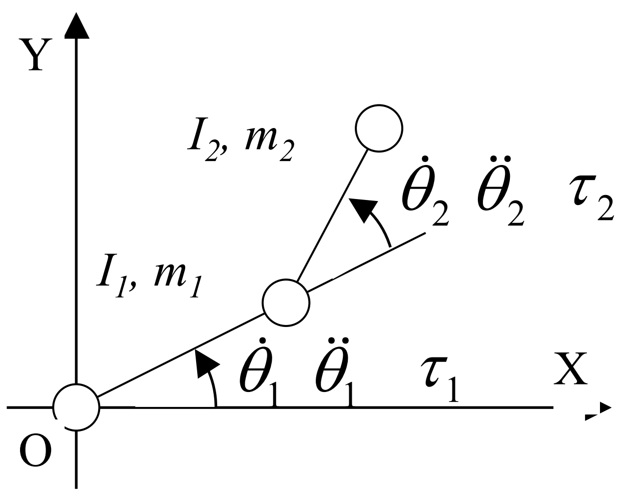
In most of practical industrial applications, kinematics (position control) is enough, because robot is geared down and moved slowly. We do not need to consider gravity
（産業界では動力学制御はほとんど使われず運動学で十分，なぜなら大きな力で低速で動くので，重力の影響は小さい）

On the other hand, we need fast and dexterous motion, we do need dynamics control, but it is still difficult because we should model every details of robot parameters.
（一方，高速で器用な動作が必要な際は動力学制御が必要，しかし，正確な制御にはすべてのパラメータが必要で実現は困難）

Derivation of Robot Dynamics (Equations of Motion)
ロボットの動力学（運動方程式）の導出

What is Dynamics?
動力学とは?





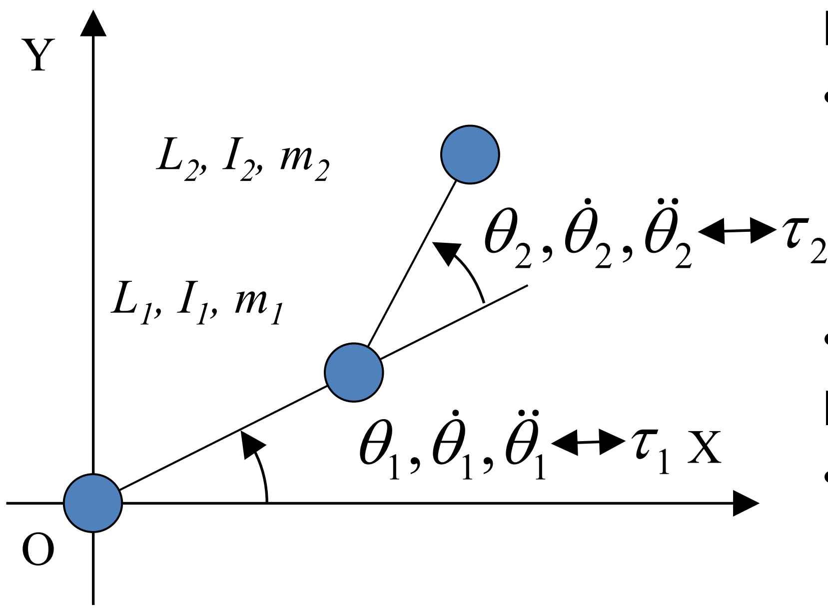
Kinematics / 運動学

- Relation between joint angles and joint positions
  関節角度と関節位置の関係
- Relation between angular velocity and translation one
  角速度と並進速度の関係
- Link length, etc.
  リンクの長さなど

Dynamics / 動力学

- Relation between joint torque and joint position and velocity
  関節トルクと関節位置と角速度の関係
- Link mass, inertia, etc.
  リンクの質量，慣性モーメントなど

Two Dynamics Problems
二つの動力学問題



Forward dynamics / 順動力学

- Initial conditions, joint torque -> Joint position, velocity
  初期状態と関節トルクを与えて，関節位置と角速度を求める
- Used for simulation / シミュレーション

Inverse Dynamics / 逆動力学

- Desired joint position, velocity -> Joint torque
  所望の関節位置と角速度を与えて，それを実現する関節トルクを求める
- Used in planning / プランニング
- Very difficult / 非常に難しい

General Form of Dynamics
動力学の一般形

$$\boldsymbol{\tau} = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})$$

$\boldsymbol{\tau}$: Joint torque vect. 関節トルクベクトル
$M(\boldsymbol{q})$:Inertia mat. 慣性行列
$\boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}})$:Centrifugral force , Coriolis force, Friction force
向心力，コリオリ力，摩擦力など
$\boldsymbol{g}(\boldsymbol{q})$:gravity vect. 重力項ベクトル
$\boldsymbol{q}$: Joint angle vect. 関節角度ベクトル
$\dot{\boldsymbol{q}}$: Joint speed vect. 関節角速度ベクトル
$\ddot{\boldsymbol{q}}$: Joint acceleration vect. 関節角加速度ベクトル

Two Derivation Methods
二つの導出法

Lagrange method
ラグランジュ法

- Lagrange function
  ラグランジュ関数
  - Kinetic energy
    運動エネルギ
  - Potential energy
    ポテンシャルエネルギ
- No need to consider internal force
  内力を考慮する必要がない→簡単
- Many symbolic differential calculation
  記号的な微分計算が必要
- We study it in this course
  この科目で学ぶ

Newton-Euler method
ニュートン･オイラー法

- Force interaction between links
  リンク間の力の相互作用を考える
- Need to consider 3D force and motion always
  常に三次元の力と位置を考慮する→複雑
- Less redundant calculation
  計算量は少ない
- We do not study it in this course
  この科目で学ばない

Principle of Lagrange Method
ラグランジュ法の原理

Suppose that a system is conservative (maintains a mechanical energy)
保存系（力学的エネルギーが保存される）を想定する

- K: Kinetic energy 運動エネルギ
- P: Potential energy ポテンシャルエネルギ

Define a Lagrangian, L as follows
ラグランジアンLを以下の様に定義する

$$\boxed{L = K - P}$$

We can derive an equation of motion as the following equations
次の式で運動方程式を導出することができる

$$\boxed{\frac{d}{dt}\frac{\partial L}{\partial \dot{x}_i} - \frac{\partial L}{\partial x_i} = 0}$$

It means that a mechanical energy is conserved if no external force is applied
上の式は，外力がかからないときには形の力学的エネルギーが保存されることを意味している

Idea of Lagrange Method
ラグランジュ法のアイディア

For example,

$$K = \frac{1}{2} m\dot{x}^2, P = mgx$$

$$L = K - P = \frac{1}{2} m\dot{x}^2 - mgx$$

$$\boxed{\frac{d}{dt}\frac{\partial L}{\partial \dot{x}_i} - \frac{\partial L}{\partial x_i} = 0}$$

The above equation represents a Newton's equation for a translational system. Similarly, it does an Euler's equation in a rotational system.
上の式は並進系ではニュートンの運動方程式を表している．同様に回転系ではオイラーの運動方程式を意味する．

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x} = p$$

Momentum / 運動量

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{x}} = m\ddot{x}$$

Force equals to time change of momentum
運動量の時間変化は力に等しい

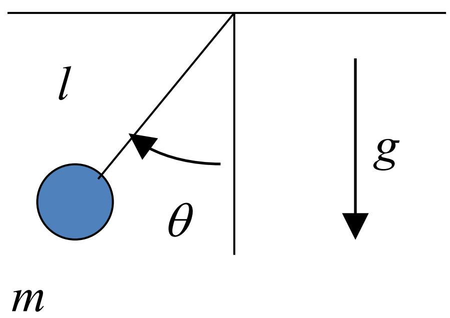
$$\frac{\partial L}{\partial x} = -mg = f$$

Potential force by position change
位置の変化によるポテンシャル力

$$m\ddot{x} - f = 0$$
$$f = m\ddot{x} = ma$$

Newton equation

Ex. Lagrange Method
ラグランジュ法の例



$$L = \frac{1}{2}m\left(l\dot{\theta}\right)^2 - mgl\left(1-\cos\theta\right)$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = 0$$

$$ml^2\ddot{\theta} + mgl\sin\theta = 0$$

$$K = \frac{1}{2}mv^2 = \frac{1}{2}m\left(l\dot{\theta}\right)^2$$

$$\ddot{\theta} = -\frac{g}{l}\sin\theta$$

$$P = mgl\left(1-\cos\theta\right)$$

Equations of Motion by Lagrange Method
ラグランジュ法による運動方程式の導出

$$L = K - P$$

If external force τ is applied to a robot
ロボットに外力τが加わる場合

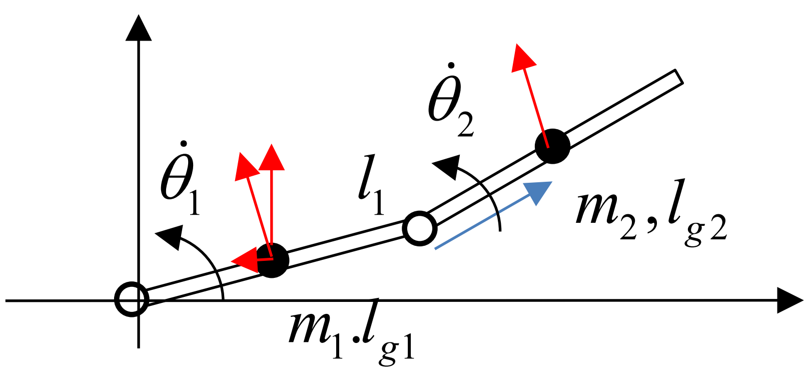
$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} \quad (i = 1, 2, \cdots, n)$$

If a robot includes dissipative term D (e.g., friction)
ロボットが散逸項（摩擦など）を含む場合

$$\tau_i = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \frac{\partial D}{\partial q_i} \quad (i = 1, 2, \cdots, n) \qquad D = \frac{1}{2} c \dot{\boldsymbol{q}}_i^T \dot{\boldsymbol{q}}_i$$

First, let us find K, P, and D

Ex: Lagrange Method: Finding Kinetic Energy 1
ラグランジュの方法: 運動エネルギ 1



Assume that each of the links is represented as a single point of CoG.
各リンクは重心の１点で表現されるものとする

Total kinetic energy: A sum of kinetic energy of the first link and the second one
全体の運動エネルギ: 第1リンクと第2リンクの運動エネルギの和

$$K = K_1 + K_2$$

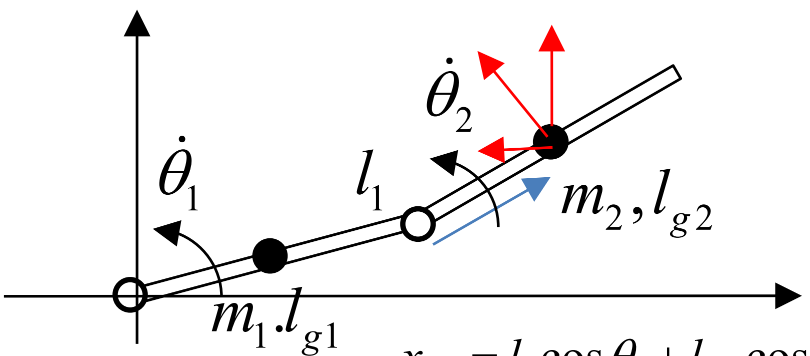
Kinetic energy of the first link
第1リンクの運動エネルギ

$$x_{g1} = l_{g1}\cos\theta_1, y_{g1} = l_{g1}\sin\theta_1$$

$$\dot{x}_{g1} = \frac{dx_{g1}}{dt} = -l_{g1}\dot{\theta}_1\sin\theta_1, \dot{y}_{g1} = \frac{dy_{g1}}{dt} = l_{g1}\dot{\theta}_1\cos\theta_1,$$

$$\begin{aligned} K_1 &= \frac{1}{2}m_1\left(\dot{x}_{gq}^2 + \dot{y}_{gq}^2\right) \\ &= \frac{1}{2}m_1\left(\left(-l_{g1}\dot{\theta}_1\sin\theta_1\right)^2 + \left(l_{g1}\dot{\theta}_1\cos\theta_1\right)^2\right) \\ &= \frac{1}{2}m_1\left(l_{g1}\dot{\theta}_1\right)^2 \end{aligned}$$

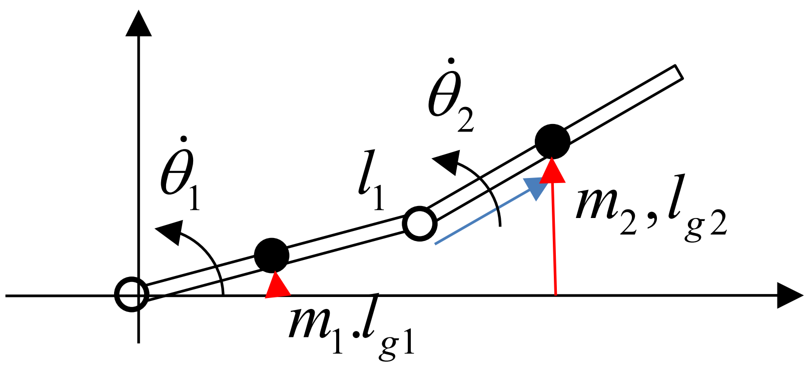
Ex: Lagrange Method: Finding Kinetic Energy 2
ラグランジュの方法: 運動エネルギ 2



Kinetic energy of the second link
第2リンクの運動エネルギ

$$x_{g2} = l_1 \cos\theta_1 + l_{g2} \cos(\theta_1 + \theta_2),$$
$$y_{g2} = l_1 \sin\theta_1 + l_{g2} \sin(\theta_1 + \theta_2),$$
$$\dot{x}_{g2} = \frac{dx_{g2}}{dt} = -l_1\dot{\theta}_1 \sin\theta_1 - l_{g2}\left(\dot{\theta}_1 + \dot{\theta}_2\right)\sin\left(\dot{\theta}_1 + \dot{\theta}_2\right),$$
$$\dot{y}_{g2} = \frac{dy_{g2}}{dt} = l_1\dot{\theta}_1 \cos\theta_1 + l_{g2}\left(\dot{\theta}_1 + \dot{\theta}_2\right)\cos\left(\dot{\theta}_1 + \dot{\theta}_2\right),$$
$$K_2 = \frac{1}{2} m_2 \left(\dot{x}_{gq}^2 + \dot{y}_{gq}^2\right)$$
$$= \frac{1}{2} m_2 \left(\left(l_1^2 + 2 l_1 l_{g2} \cos\theta_2 + l_{g2}^2\right)\dot{\theta}_1^2 + 2 l_{g2}\left(l_1 \cos\theta_2 + l_{g2}\right)\dot{\theta}_1\dot{\theta}_2 + l_{g2}^2\dot{\theta}_2^2\right)$$

Ex: Lagrange Method: Finding Potential Energy
ラグランジュの方法: ポテンシャルエネルギ



Assume that each of the links is represented as a single point of CoG.
各リンクは重心の１点で表現されるものとする

Total potential energy: A sum of potential energy of the first link and the second one
全体のポテンシャルエネルギ: 第1リンクと第2リンクのポテンシャルエネルギの和

$$P = P_1 + P_2$$

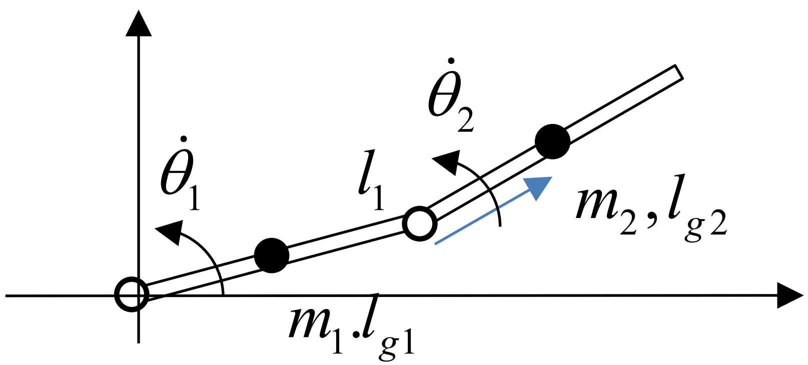
Potential energy of the first link
第１リンクのポテンシャルエネルギ

$$y_{g1} = l_{g1} \sin\theta_1$$
$$P_1 = m_1 g y_{g1} = m_1 g l_{g1} \sin\theta_1$$

Potential energy of the second link
第2リンクのポテンシャルエネルギ

$$y_{g2} = l_1 \sin\theta_1 + l_{g2} \sin(\theta_1 + \theta_2),$$
$$P_2 = m_2 g y_{g2} = m_2 g \left(l_1 \sin\theta_1 + l_{g2} \sin(\theta_1 + \theta_2) \right)$$

Ex: Lagrange Method: Finding Dissipative Energy
ラグランジュの方法: 散逸エネルギ



Assume that each of the links is represented as a single point of CoG.
各リンクは重心の１点で表現されるものとする

Total dissipative energy: A sum of dissipative energy of the first link and the second one
全体の散逸エネルギ: 第1リンクと第2リンクの散逸エネルギの和

$$D = D_1 + D_2$$

Dissipative energy of the first link
第１リンクの散逸エネルギ

$$D_1 = \frac{1}{2} c_1 \dot{\theta}_1^2$$

Potential energy of the second link
第2リンクのポテンシャルエネルギ

$$D_2 = \frac{1}{2} c_2 \dot{\theta}_2^2$$

Ex: Lagrange Method: Derivation
ラグランジュの方法: 運動方程式の導出

$$
\begin{aligned}
L &= K - P \\
&= K_1 + K_2 - P_1 - P_2 \\
&= \frac{1}{2} m_1 \left(l_{g1} \dot{\theta}_1 \right)^2 + \frac{1}{2} m_2 \left(\left(l_1^2 + 2 l_1 l_{g2} \cos\theta_2 + l_{g2}^2 \right) \dot{\theta}_1^2 + 2 l_{g2} \left(l_1 \cos\theta_2 + l_{g2} \right) \dot{\theta}_1 \dot{\theta}_2 + l_{g2}^2 \dot{\theta}_2^2 \right) \\
&\quad - m_1 g l_{g1} \sin\theta_1 - m_2 g \left(l_1 \sin\theta_1 + l_{g2} \sin\left(\theta_1 + \theta_2 \right) \right)
\end{aligned}
$$

$$
D = \frac{1}{2} c_1 \dot{\theta}_1^2 + \frac{1}{2} c_2 \dot{\theta}_2^2
$$

$$
\tau_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} + \frac{\partial D}{\partial \dot{\theta}_1}
$$

$$
\frac{\partial L}{\partial \dot{\theta}_1} = \left(m_1 l_{g1}^2 + m_2 \left(l_1^2 + 2 l_1 l_{g2} \cos\theta_2 + l_{g2}^2 \right) \right) \dot{\theta}_1 + m_2 l_{g2} \left(l_1 \cos\theta_2 + l_{g2} \right) \dot{\theta}_2
$$

$$
\begin{aligned}
\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} &= \left(m_1 l_{g1}^2 + m_2 \left(l_1^2 + 2 l_1 l_{g2} \cos\theta_2 + l_{g2}^2 \right) \right) \ddot{\theta}_1 + m_2 l_{g2} \left(l_1 \cos\theta_2 + l_{g2} \right) \ddot{\theta}_2 \\
&\quad - 2 m_2 l_1 l_{g2} \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{g2} \sin\theta_2 \dot{\theta}_2^2
\end{aligned}
$$

Ex: Lagrange Method: Derivation

ラグランジュの方法: 運動方程式の導出

$$\frac{\partial L}{\partial \theta_1} = -m_1 g l_{g1} \cos\theta_1 - m_2 g\left(l_1 \cos\theta_1 + l_{g2}\cos\left(\theta_1 + \theta_2\right)\right)$$

$$\frac{\partial D}{\partial \dot{\theta}_1} = c_1 \dot{\theta}_1$$

$$\begin{aligned}
\tau_1 &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} + \frac{\partial D}{\partial \dot{\theta}_1} \\
&= \left(m_1 l_{g1}^2 + m_2\left(l_1^2 + 2 l_1 l_{g2}\cos\theta_2 + l_{g2}^2\right)\right)\ddot{\theta}_1 + m_2 l_{g2}\left(l_1\cos\theta_2 + l_{g2}\right)\ddot{\theta}_2 \\
&\quad -2 m_2 l_1 l_{g2}\sin\theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{g2}\sin\theta_2 \dot{\theta}_2^2 \\
&\quad + m_1 g l_{g1}\cos\theta_1 + m_2 g\left(l_1\cos\theta_1 + l_{g2}\cos\left(\theta_1 + \theta_2\right)\right) + c_1\dot{\theta}_1
\end{aligned}$$

Ex: Lagrange Method: Derivation

ラグランジュの方法: 運動方程式の導出

$$\tau_2 = \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} + \frac{\partial D}{\partial \dot{\theta}_2}$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_{g2}\left(l_1 \cos\theta_2 + l_{g2}\right)\dot{\theta}_1 + m_2 l_{g2}^2 \dot{\theta}_2$$

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} = m_2 l_{g2}\left(l_1 \cos\theta_2 + l_{g2}\right)\ddot{\theta}_1 + m_2 l_{g2}^2 \ddot{\theta}_2 - m_2 l_1 l_{g2} \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 l_1 l_{g2} \sin\theta_2 \dot{\theta}_1 \left(\dot{\theta}_1 + \dot{\theta}_2\right) - m_2 g l_{g2} \cos\left(\theta_1 + \theta_2\right)$$

$$\frac{\partial D}{\partial \dot{\theta}_2} = c_2 \dot{\theta}_2$$

$$\begin{aligned}\tau_2 &= \frac{d}{dt}\frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} + \frac{\partial D}{\partial \dot{\theta}_2} \\ &= m_2 l_{g2}\left(l_1 \cos\theta_2 + l_{g2}\right)\ddot{\theta}_1 + m_2 l_{g2}^2 \ddot{\theta}_2 + m_2 l_1 l_{g2} \sin\theta_2 \dot{\theta}_1^2 + m_2 g l_{g2} \cos\left(\theta_1 + \theta_2\right) + c_2 \dot{\theta}_2\end{aligned}$$

Ex: Lagrange Method: Derivation
ラグランジュの方法: 運動方程式の導出

Summarizing the above two equations in a vector-matrix format, we have got the following generalized form of robot motion equation.
上の2つの運動方程式をベクトルと行列の形でまとめて，以下のロボットの運動方程式の一般形が得られる.

$$
\begin{aligned}
\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} &= \begin{pmatrix} m_1 l_{g1}^2 + m_2\left(l_1^2 + 2 l_1 l_{g2}\cos\theta_2 + l_{g2}^2\right) & m_2 l_{g2}\left(l_1\cos\theta_2 + l_{g2}\right) \\ m_2 l_{g2}\left(l_1\cos\theta_2 + l_{g2}\right) & m_2 l_{g2}^2 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{pmatrix} \\
&+ \begin{pmatrix} -m_2 l_1 l_{g2}\sin\theta_2\left(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2\right) + c_1\dot{\theta}_1 \\ m_2 l_1 l_{g2}\sin\theta_2\dot{\theta}_1^2 + c_2\dot{\theta}_2 \end{pmatrix} \\
&+ g\begin{pmatrix} m_1 l_{g1}\cos\theta_1 + m_2\left(l_1\cos\theta_1 + l_{g2}\cos\left(\theta_1+\theta_2\right)\right) \\ m_2 l_{g2}\cos\left(\theta_1+\theta_2\right) \end{pmatrix} \\
&= M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q})
\end{aligned}
$$

Lagrange Method by Matlab
Matlabによるラグランジュ法

```
% Robot parameters
syms L1 L2 L1g L2g m1 m2 c1 c2
% Gravity
syms g
% Joint angle: main variable
syms t q1(t) q2(t)
% Robot position: sub variable
syms x1g y1g x2g y2g
% Robot control
syms tau1 tau2


% Position of CoG of the first link
x1g = L1g*cos(q1(t));
y1g = L1g*sin(q1(t));
% Position of CoG of the second link
x2g = L1*cos(q1) + L2g*cos(q1+q2);
y2g = L1*sin(q1) + L2g*sin(q1+q2);

% Velocity of CoG of the first link
vx1g = diff(x1g, t);
vy1g = diff(y1g, t);
% Velocity of CoG of the second link
vx2g = diff(x2g, t);
vy2g = diff(y2g, t);
```

Lagrange Method by Matlab
Matlabによるラグランジュ法

```
% Kinetic energy of the first link
K1 = m1/2*(vx1g^2+vy1g^2);
% Kinetic energy of the second link
K2 = m2/2*(vx2g^2+vy2g^2);
% Total kinetic energy
K = K1 + K2;

% Potential energy of the first link
P1 = m1*g*y1g;
% Potential energy of the second link
P2 = m2*g*y2g;
% Total potential energy
P = P1 + P2;

% Lagurangian
L = K - P;

% Dissipative energy
D = 1/2*c1*diff(q1(t),t)^2 + 1/2*c2*diff(q2(t),t)^2;
```

Lagrange Method by Matlab

Matlabによるラグランジュ法

```
% Equation of Motion: i=1
% d/dt * dL/dq1'
eqn11 = simplify(diff(diff(L, diff(q1(t),t)),t));
% dL/dq1
eqn12 = simplify(diff(L, q1(t)));
% dD/dq1'
eqn13 = simplify(diff(D, diff(q1(t),t)));
% Equation for q1
tau1 == simplify(eqn11-eqn12+eqn13)

% Equation of Motion: i=2
% d/dt * dL/dq2'
eqn21 = simplify(diff(diff(L, diff(q2(t),t)),t));
% dL/dq2
eqn22 = simplify(diff(L, q2(t)));
% dD/dq2'
eqn23 = simplify(diff(D, diff(q2(t),t)));
% Equation for q1
tau2 == simplify(eqn21-eqn22+eqn23)
```

Derived Equation of Motion
導出された運動方程式

tau1 == c1*diff(q1(t), t) + L1^2*m2*diff(q1(t), t, t) + L1g^2*m1*diff(q1(t), t, t) + L2g^2*m2*diff(q1(t), t, t) + L2g^2*m2*diff(q2(t), t, t) + L2g*g*m2*cos(q1(t) + q2(t)) + L1*g*m2*cos(q1(t)) + L1g*g*m1*cos(q1(t)) - L1*L2g*m2*sin(q2(t))*diff(q2(t), t)^2 + 2*L1*L2g*m2*cos(q2(t))*diff(q1(t), t, t) + L1*L2g*m2*cos(q2(t))*diff(q2(t), t, t) - 2*L1*L2g*m2*sin(q2(t))*diff(q1(t), t)*diff(q2(t), t)

$$
\begin{aligned}
\tau_1 = {} & c_1\dot{\theta}_1 + m_2 l_1^2 \ddot{\theta}_1 + m_1 l_{1g}^2 \ddot{\theta}_1 + m_2 l_{2g}^2 \ddot{\theta}_1 + m_2 l_{2g}^2 \ddot{\theta}_2 \\
& + m_2 g l_{2g} \cos\left(\theta_1 + \theta_2\right) + m_2 g l_1 \cos\theta_1 + m_{1g} g l_{1g} \cos\theta_1 \\
& - m_2 l_1 l_{2g} \sin\theta_2 \dot{\theta}_2^2 + 2 m_2 l_1 l_{2g} \cos\theta_2 \ddot{\theta}_1 + m_2 l_1 l_{2g} \cos\theta_2 \ddot{\theta}_2 \\
& - 2 m_2 l_1 l_{2g} \sin\theta_2 \dot{\theta}_1 \dot{\theta}_2
\end{aligned}
$$

Derived Equation of Motion
導出された運動方程式

tau2 == c2*diff(q2(t), t) + L2g^2*m2*diff(q1(t), t, t) + L2g^2*m2*diff(q2(t), t, t) + L2g*g*m2*cos(q1(t) + q2(t)) + L1*L2g*m2*sin(q2(t))*diff(q1(t), t)^2 + L1*L2g*m2*cos(q2(t))*diff(q1(t), t, t)

$$\begin{aligned}\tau_2 &= c_2\dot{\theta}_1 + m_2 l_{2g}^2 \ddot{\theta}_1 + m_2 l_{2g}^2 \ddot{\theta}_2 \\ &\quad + m_2 g l_{2g} \cos\left(\theta_1 + \theta_2\right) + m_2 l_1 l_{2g} \sin\theta_2 \dot{\theta}_1^2 + m_2 l_1 l_{g2} \cos\theta_2 \ddot{\theta}_1\end{aligned}$$