

情報処理 MATLAB シミュレーション

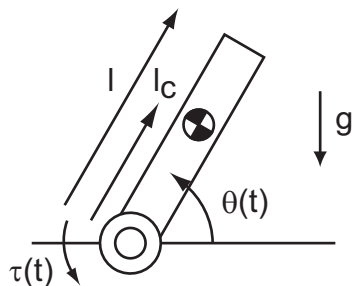
平井 慎一

立命館大学 ロボティクス学科

講義の流れ

- 1 自由度リンク機構
- 2 P 制御
- 3 PD 制御
- 4 PI 制御
- 5 まとめ

1 自由度リンク機構



運動方程式

$$(J_c + ml_c^2)\ddot{\theta} = -mgl_c \cos \theta + \tau$$

m : 質量, J_c : 重心周りの慣性モーメント, g : 重力加速度
 l_c : 回転中心と重心の距離, $\tau(t)$: 外部トルク

1 自由度リンク機構

定数: $J = J_c + ml_c^2$, $P = mgl_c$

運動方程式

$$J\ddot{\theta} = -P \cos \theta + \tau$$

⇓

$\dot{\theta} = \omega$ とおくと $\ddot{\theta} = \dot{\omega}$ なので

$$\begin{cases} \dot{\theta} = \omega \\ \dot{\omega} = \frac{1}{J}(-P \cos \theta + \tau) \end{cases}$$

⇓

標準形

$$\mathbf{q} = \begin{bmatrix} \theta \\ \omega \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \omega \\ \frac{1}{J}(-P \cos \theta + \tau) \end{bmatrix}$$

1 自由度リンク機構

`one_dim_link.m` 1 自由度リンク機構の運動方程式の標準形

`torque.m` 外部トルク

`one_dim_link_solve.m` 1 自由度リンク機構の運動方程式を解く

`parameters.m` パラメータの値を計算

`video_file.m` 動画ファイルを作成

1 自由度リンク機構

スクリプトファイル parameters.m

```
global J P
```

```
% 一様な円柱のパラメータ
```

```
l = 0.4; % リンク長
```

```
lc = 0.2; % 重心の位置
```

```
r = 0.02; % 円の半径
```

```
rho = 5e+3; % 密度 kg/m3
```

```
% 一様な円柱の質量と慣性モーメント kg, m, s
```

```
m = rho*(pi*r2)*l; % 質量
```

```
Jc = m * (r2/4 + l2/12); % 重心まわりの慣性モーメント
```

```
g = 9.8; % 重力加速度
```

```
J = Jc + m*lc2; % 関節周りの慣性モーメント
```

```
P = m*g*lc; % 定数 mg lc
```

1 自由度リンク機構

スクリプトファイル `parameters.m` を実行せよ.

変数 J , P の値を確認せよ.

1 自由度リンク機構

関数ファイル one_dim_link.m

```
function dotq = one_dim_link (t, q)
    global J P
    theta = q(1);
    omega = q(2);
    dottheta = omega;
    dotomega = (1/J)*(-P*cos(theta) + torque(t));
    dotq = [ dottheta; dotomega ];
end
```


1 自由度リンク機構

関数ファイル torque.m

```
% 関節トルク  
function tau = torque(t)  
    tau = 0;  
end
```

1 自由度リンク機構

関数 `torque` の値を確認せよ.

```
>> torque(0)
```

```
ans =  
      0
```

```
>>
```

関数 `one_dim_link` の値を確認せよ.

```
>> one_dim_link(0, [2;3])
```

```
ans =  
      3.0000  
     15.2648
```

```
>>
```

1 自由度リンク機構

スクリプトファイル `one_dim_link_solve.m`

```
parameters; % パラメータの設定
```

```
interval = 0.00:0.001:10.00;
```

```
qinit = [0.00; 0.00];
```

```
[time,q] = ode45(@one_dim_link,interval,qinit);
```

1 自由度リンク機構

スクリプトファイル `one_dim_link_solve.m` を実行せよ.

画像ファイル `one_dim_link_angle.png` と
動画ファイル `one_dim_link.mp4` が
作成されていることを確認せよ.

関数ファイル `torque.m` 内の `tau = 0;` を以下のように書き換えて、スクリプトファイル `one_dim_link_solve.m` を実行せよ.

- `tau = 2.00*(t<=3.0);`
- `tau = 1.50*sin(2*pi*t);`

P 制御

P 制御

$$\tau = -K_p(\theta - \theta^d)$$

運動方程式

$$J\ddot{\theta} = -P \cos \theta - K_p(\theta - \theta^d)$$

標準形

$$\mathbf{q} = \begin{bmatrix} \theta \\ \omega \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \omega \\ \frac{1}{J}\{-P \cos \theta - K_p(\theta - \theta^d)\} \end{bmatrix}$$

P 制御

`one_dim_link_P.m` 1 自由度リンク機構の P 制御の標準形

`one_dim_link_P_solve.m` P 制御の標準形を解く

P制御

スクリプトファイル `one_dim_link_P_solve.m`

```
parameters; % パラメータの設定
```

```
global Kp thetad
```

```
Kp = 10.00; % 比例ゲイン
```

```
thetad = pi/4; % 目標角度
```

```
interval = 0.00:0.001:10.00;
```

```
qinit = [0.00; 0.00];
```

```
[time,q] = ode45(@one_dim_link_P,interval,qinit);
```

P制御

関数ファイル one_dim_link_P.m

```
function dotq = one_dim_link_P (t, q)
    global J P Kp thetad
    theta = q(1);
    omega = q(2);
    dottheta = omega;
    dotomega = (1/J)*(-P*cos(theta) -Kp*(theta-thetad));
    dotq = [ dottheta; dotomega ];
end
```


P 制御

スクリプトファイル `one_dim_link_P_solve.m` を実行せよ.

画像ファイル `one_dim_link_P_angle.png` と
動画ファイル `one_dim_link_P_movie.mp4` が
作成されていることを確認せよ.

振動が減衰しているか. 目標角度に収束しているか.

PD 制御

PD 制御

$$\tau = -K_p(\theta - \theta^d) - K_d\dot{\theta}$$

運動方程式

$$J\ddot{\theta} = -P \cos \theta - K_p(\theta - \theta^d) - K_d\dot{\theta}$$

標準形

$$\mathbf{q} = \begin{bmatrix} \theta \\ \omega \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \omega \\ \frac{1}{J}\{-P \cos \theta - K_p(\theta - \theta^d) - K_d\omega\} \end{bmatrix}$$

PD 制御

`one_dim_link_PD.m` 1 自由度リンク機構の PD 制御の標準形
`one_dim_link_PD_solve.m` PD 制御の標準形を解く

PD 制御

スクリプトファイル `one_dim_link_PD_solve.m`

```
parameters; % パラメータの設定
```

```
global Kp Kd thetad
```

```
Kp = 10.00; % 比例ゲイン
```

```
Kd = 0.100; % 微分ゲイン
```

```
thetad = pi/4; % 目標角度
```

```
interval = 0.00:0.001:10.00;
```

```
qinit = [0.00; 0.00];
```

```
[time,q] = ode45(@one_dim_link_PD,interval,qinit);
```

PD 制御

関数ファイル one_dim_link_PD.m

```
function dotq = one_dim_link_PD (t, q)
    global J P Kp Kd thetad
    theta = q(1);
    omega = q(2);
    dottheta = omega;
    dotomega = (1/J)*(-P*cos(theta) -Kp*(theta-thetad) -Kd*omega);
    dotq = [ dottheta; dotomega ];
end
```

PD 制御

スクリプトファイル `one_dim_link_PD_solve.m` を実行せよ.

画像ファイル `one_dim_link_PD_angle.png` と
動画ファイル `one_dim_link_PD_movie.mp4` が
作成されていることを確認せよ.

振動が減衰しているか. 目標角度に収束しているか.

PI制御

PI制御

$$\tau = -K_p(\theta - \theta^d) - K_i \int_0^t \{\theta(\tau) - \theta^d\} d\tau$$

運動方程式

$$J\ddot{\theta} = -P \cos \theta - K_p(\theta - \theta^d) - K_i \int_0^t \{\theta(\tau) - \theta^d\} d\tau$$

時間積分を $\xi(t)$ とおく.

$$\xi(t) = \int_0^t \{\theta(\tau) - \theta^d\} d\tau$$

時間微分

$$\dot{\xi} = \theta - \theta^d$$

PI制御

運動方程式

$$J\ddot{\theta} = -P \cos \theta - K_p(\theta - \theta^d) - K_i \int_0^t \{\theta(\tau) - \theta^d\} d\tau$$

↓

$$J\dot{\omega} = -P \cos \theta - K_p(\theta - \theta^d) - K_i \xi$$

標準形

$$\mathbf{q} = \begin{bmatrix} \theta \\ \omega \\ \xi \end{bmatrix}, \quad \dot{\mathbf{q}} = \begin{bmatrix} \omega \\ \frac{1}{J} \{-P \cos \theta - K_p(\theta - \theta^d) - K_i \xi\} \\ \theta - \theta^d \end{bmatrix}$$

PI制御

`one_dim_link_PI.m` 1自由度リンク機構のPI制御の標準形

`one_dim_link_PI_solve.m` PI制御の標準形を解く

PI制御

スクリプトファイル `one_dim_link_PI_solve.m`

```
parameters; % パラメータの設定
```

```
global Kp Ki thetad
```

```
Kp = 10.00; % 比例ゲイン
```

```
Ki = 0.500; % 積分ゲイン
```

```
thetad = pi/4; % 目標角度
```

```
interval = 0.00:0.001:10.00;
```

```
qinit = [0.00; 0.00; 0.00];
```

```
[time,q] = ode45(@one_dim_link_PI,interval,qinit);
```

PI制御

関数ファイル one_dim_link_PI.m

```
function dotq = one_dim_link_PI (t, q)
    global J P Kp Ki thetad
    theta = q(1);
    omega = q(2);
    xi = q(3);
    dottheta = omega;
    dotomega = (1/J)*(-P*cos(theta) -Kp*(theta-thetad) -K
    dotxi = theta - thetad;
    dotq = [ dottheta; dotomega; dotxi ];
end
```

PI制御

スクリプトファイル `one_dim_link_PI_solve.m` を実行せよ.

画像ファイル `one_dim_link_PI_angle.png` と
動画ファイル `one_dim_link_PI_movie.mp4` が
作成されていることを確認せよ.

振動が減衰しているか. 目標角度に収束しているか.

まとめ

MATLABによるシミュレーション

- 運動方程式を導く
- 常微分方程式を標準形に変換する
- 常微分方程式を数值的に解く
- シミュレーション結果を可視化する

レポート

1自由度リンク機構をPID制御する。ゲイン K_p , K_i , K_d を適切に選び、そのときの1自由度リンク機構の運動をグラフで表せ。Simulink等を使わず、プログラムを書いてシミュレーションを実行する。ゲインの値を変えてシミュレーションを実行し、ゲインの値が結果にどのように影響するかを考察せよ。レポートには、プログラムと実行結果、考察を載せる。

レポートは一個のpdfファイル(pdfファイル以外は採点対象外)(+動画ファイル)でmanaba+Rに提出。
ワードや写真のファイルはpdfに変換し、アップすること

ファイル名：学籍番号(11桁半角数字) 名前(空白なし).pdf
例えば 12345678901 平井慎一.pdf
12345678901HiraiShinichi.pdf

期限 2024年5月20日(月曜日) 1:00 AM

レポート

- サンプルプログラムを参考に
- 違いに注意

- , コンマ 要素の区切り
- ; セミコロン 文の最後, 列ベクトル (dotq, qinit)
- : コロン 等間隔の要素列 (interval)
- . ピリオド