認識行動システム論

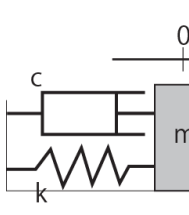
第10回
梶本裕之

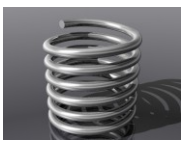
日程

01/14 ロボティクスの基礎の基礎
01/21 画像処理
01/28 期末テスト(授業時間中)

復習：PD制御

(復習) 制御をしたい

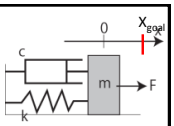




時刻0に，x=0の位置にあったおもりmを，
$x=X_{goal}$に移動したい．

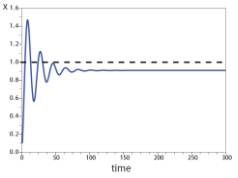
どのような力F(t)を加えたらよいだろうか？

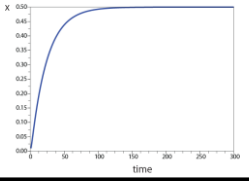
(復習) PD制御



•P(比例)制御では，
振動を抑えるために大きなダンパを用意する必要．
•言いかえれば，システムを物理的に変える必要．

•これを改善するため，バーチャルなダンパ(ブレーキ)を用意する．





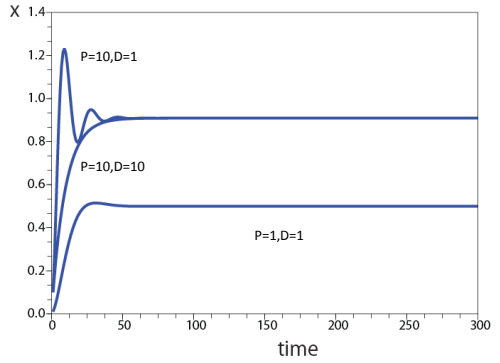
(復習) PD制御のシミュレーション

•P成分：目標値と現在地の差に比例
•D成分：速度に比例したブレーキ．つまりダンパ

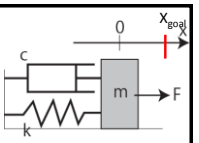
Scilabコード

```
m=1.0;          //質量
c=1.0;          //ダンパ
k=1.0;          //バネ
xgoal=1.0;      //目的地
x=0;            //現在地
v=0;            //現在速度
dt = 0.1;       //時間間隔
xrecord =[];    //データ記録用
P=1.0;          //P成分
D=1.0;          //I成分

for t=1:300,
 F=P * (xgoal-x) - D * v;
 a = F - k*x - c*v;
 v = v+a*dt;
 x = x+v*dt;
 xrecord = [xrecord,x];
end

plot(xrecord);
```

(復習) PD制御のシミュレーション(結果)



(復習) PD制御の数学



•システム

$$m\ddot{x}+c\dot{x}+kx=f$$

•力の制御の仕方

$$f=a(x_{goal}-x)-b\dot{x}$$

•つまり

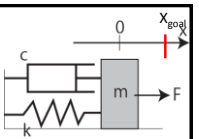
$$m\ddot{x}+c\dot{x}+kx=a(x_{goal}-x)-b\dot{x}$$

•これは，P制御において，ダンパ成分がcからb+cに増えたことを意味する.

$$m\ddot{x}+(b+c)\dot{x}+kx=a(x_{goal}-x)$$

インピーダンス制御

一般化：インピーダンス制御



•システム（バネ成分は無いことも多い）

$$m\ddot{x}+c\dot{x}+kx=f$$

•力の制御の仕方

•つまり

•これは，元のインピーダンスm,c,kを，m+m', c+c', k+k'に変化させたことを意味する.

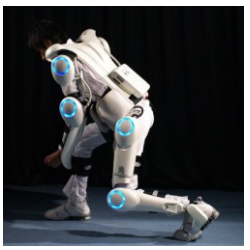
インピーダンス制御≒ロボットを「軽く」する.

$$(m+m')\ddot{x}+(c+c')\dot{x}+(k+k')x=ax_{goal}$$

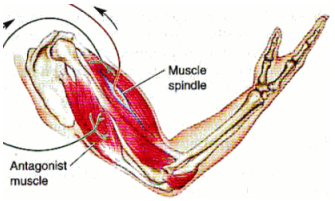
•m'=-m，c'=-c，k'=-kとしたら理想的には「無くなる」
•ロボットの場合，xの代わりにθ(関節角)，mの代わりにI(慣性トルク)等々.

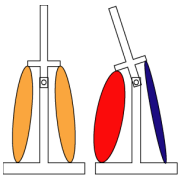
⇒ロボットのインピーダンスは0になる.

•このように制御されたロボットは，見かけ上「軽く」なる.
⇒対人安全性の向上.
⇒ 操作性の向上.



人間＝インピーダンス可変ロボット





•一つの自由度(関節)に対して，二つの筋肉で駆動(拮抗筋)

•力とインピーダンス(やわらかさ)の二つの情報を出力している.
•筋Aと筋Bの差＝外力
•筋Aと筋Bの和＝柔らかさ

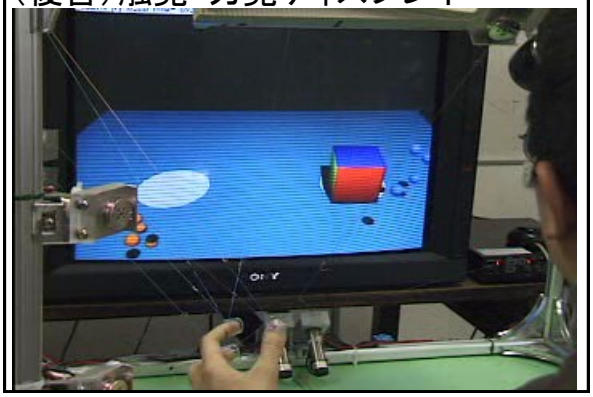
インピーダンス提示装置としての力覚ディスプレイ

（復習）視覚ディスプレイ

- Sutherland ”The Ultimate Display” (1965)



（復習）触覚・力覚ディスプレイ



触覚・力覚ディスプレイに関して

世界の何を再生すればよい？

答（1） 世界の「形」

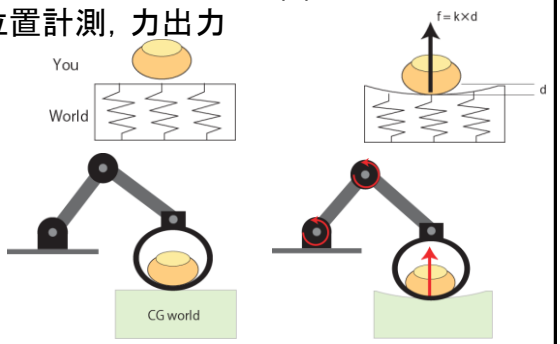
答（2） 世界の「柔らかさ」

•柔らかさ＝インピーダンス

•空中＝インピーダンス0

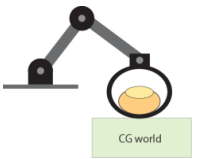
•物体と接触＝インピーダンス発生
•物体と押し合う＝インピーダンスを感じている

インピーダンスの提示(1):位置計測，力出力



ロボットアームは，「位置」を計測し，必要な「力」を出すことで世界のインピーダンスを表現する．

位置計測，力出力

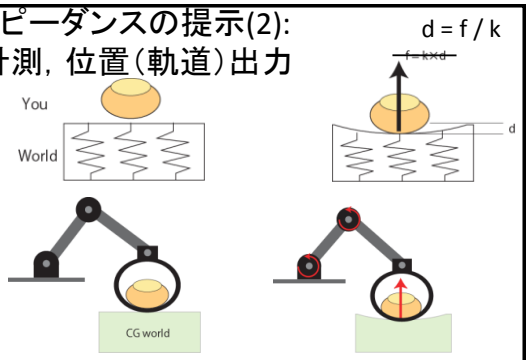


ロボットアームは，「位置」を計測し，必要な「力」を出すことで世界のインピーダンスを表現する．

•ロボットアームはものすごく軽い必要
（軽く作る＋インピーダンス制御で軽くする）

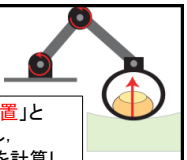
•位置センサ（モータのエンコーダ）があればよい．
（ユーザの力を計測する必要はない）

インピーダンスの提示(2): 力計測，位置（軌道）出力



ロボットアームは，「位置」とユーザの「力」を計測し，
必要な「変位（軌道）」を計算し，実現することで
世界のインピーダンスを表現する．

力計測，軌道出力



ロボットアームは，「位置」と
ユーザの「力」を計測し，
必要な「変位（軌道）」を計算し，
実現することで世界のインピーダンスを表現する．

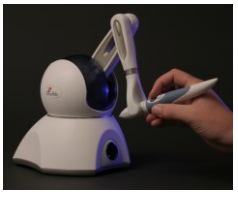
- ロボットアームは固くて重い産業用ロボットでOK
- 位置センサ（モータのエンコーダ）と
  ユーザの力を計測するカセンサが必要．

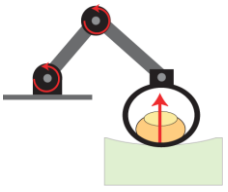
「振る舞い（軌道）」によってインピーダンスを表現する：
位置制御ベースインピーダンス制御と全く同じ

ロボティクスの基礎の基礎：

ロボットの姿勢・力・速度

座標変換の必要性

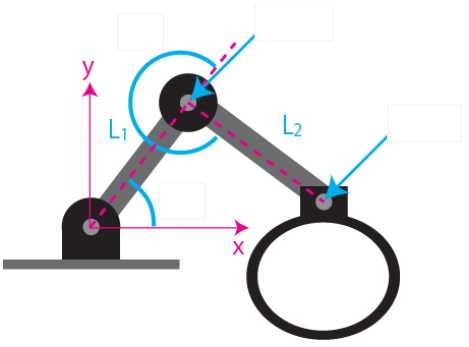




- 関節の角度から，ロボット末端の位置を知りたい．
- ロボット末端の位置から，関節の角度を知りたい．

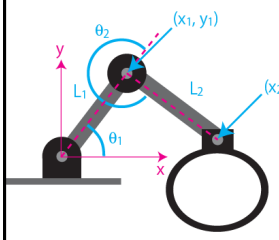
- ロボット末端をある速度で動かすための関節の速度は？
- ロボット末端にある力を出すための，関節のトルクは？

座標の定義



順キネマティクス

関節の角度(θ1,θ2)から，
ロボット末端の位置(x2,y2)を知りたい．



$x_1 =$

$y_1 =$

$x_2 =$

$=$

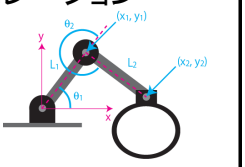
$y_2 =$

$=$

順キネマティクスのシミュレーション

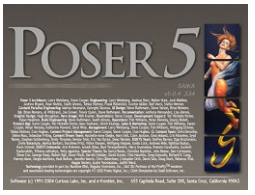
Scilabコード

```
L1 = 1.0;
L2 = 1.0;

for t=0:0.1:%pi,
  theta1 = t; //関節1の角度
  theta2 = t*2;//関節2の角度
  //関節座標
  x1 = L1 * cos(theta1);
  y1 = L1 * sin(theta1);
  x2 = x1 + L2 * cos(theta1+theta2);
  y2 = y1 + L2 * sin(theta1+theta2);

  armX = [0,x1,x2]; //関節のx座標
  armY = [0,y1,y2]; //関節のy座標

  plot(armX,armY,'O-'); //描画
  sleep(100); //100ms休む
end
```



POSER中の順キネマティクス

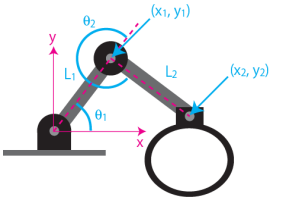
デモ



ロボットの知識はCGアニメーションに必須．
（基礎的な知識はほぼ共通）

逆キネマティクス

ロボット末端の位置を(x2,y2) に移動したい．
関節の角度(θ1,θ2) は何度回せば良いか？

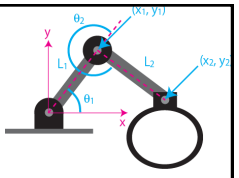


$$x_2 = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

$$y_2 = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

逆キネマティクス

θ1+θ2をθ12と書いて



$$x_2 = L_1 \cos\theta_1 + L_2 \cos\theta_{12}$$

$$y_2 = L_1 \sin\theta_1 + L_2 \sin\theta_{12}$$

$x_2^2 + y_2^2 =$

$=$

$=$ 単なる余弦定理

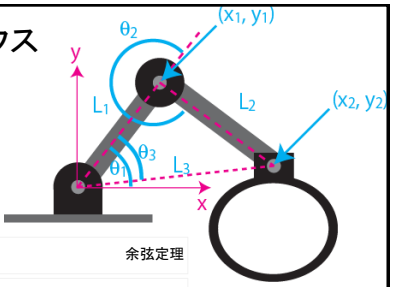
$\cos(\theta_2) =$

$\theta_2 =$ 任意性あり

逆キネマティクス

L3，θ3を定義

$L_3 = \sqrt{x_2^2 + y_2^2}$

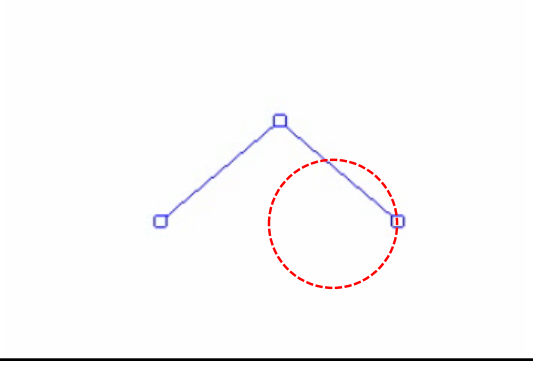


$L_2^2 =$ 余弦定理

$\theta_3 =$

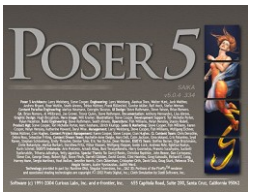
$\theta_1 =$

逆キネマティクス:
先端に円を描かせるシミュレーション結果



POSER中の逆キネマティクス

デモ



ロボットの知識はCGアニメーションに必須.
（基礎的な知識はほぼ共通）

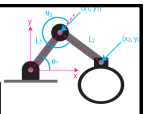
レポート課題

以下は逆キネマティクスの式を用いてロボット先端に円を描かせたプログラムである. 完成させよ.
※acosには正負の任意性がある事に注意. 両方試すしかない.

```
L1 = 1.0;
L2 = 1.0;

for t=0:0.1:2*%pi,
 //目標の先端位置. 円を描かせる
 x2 = 1+0.5*cos(t);
 y2 = 0.5*sin(t);

 L3 =
 theta2 =                    ;
 theta3 =                    ;

 theta1 = t
```

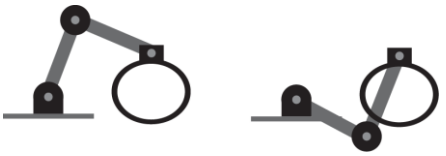
```
//以下は順キネティクス
x1 = L1 * cos(theta1);
y1 = L1 * sin(theta1);
x2 = x1 + L2 * cos(theta1+theta2);
y2 = y1 + L2 * sin(theta1+theta2);

//以下は描画用
armX = [0,x1,x2]; //関節のx座標
armY = [0,y1,y2]; //関節のy座標
square(-2.5,-2.5,2.5,2.5);
plot(armX,armY,'O-'); //描画
sleep(100); //100ms休む
end
```

逆キネマティクスまとめ



ロボット末端の位置を(x2,y2) に移動したい.
関節の角度(θ1,θ2) は何度回せば良いか？

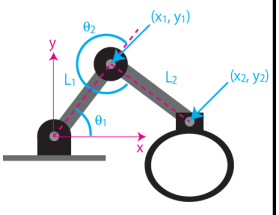
頑張って式変形し，θ1，θ2をx2，y2で表す.
一般的な解法は無い. とても大変.
解が複数個あることも.



先端速度の計算

関節の速度$\dot{\theta}_1, \dot{\theta}_2$ からロボット末端の速度を計算.

$$x_2 = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2)$$
$$y_2 = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2)$$

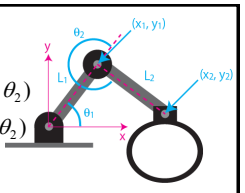


例えば $\frac{d\cos\theta_1}{dt} = -\dot{\theta}_1 \sin\theta_1$ から

$$\dot{x}_2 =$$
$$\dot{y}_2 =$$

先端速度の計算

$$\dot{x}_2 = -L_1\dot{\theta}_1 \sin\theta_1 - L_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2)$$
$$\dot{y}_2 = L_1\dot{\theta}_1 \cos\theta_1 + L_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2)$$



$$\begin{bmatrix}\dot{x}_2 \\ \dot{y}_2\end{bmatrix} = \begin{bmatrix} & \\ & \end{bmatrix}\begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2\end{bmatrix}$$

これを $\begin{bmatrix}\dot{x}_2 \\ \dot{y}_2\end{bmatrix} = \mathbf{J}\begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2\end{bmatrix}$ と書き，Jをヤコビアンと呼ぶ.

ヤコビアンは時々刻々と変化する. 毎サイクル計算

ヤコビアン

•元の式 $x_2 = L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2)$
$y_2 = L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2)$

•一般的に $x_2 = f(\theta_1,\theta_2)$
$y_2 = g(\theta_1,\theta_2)$

•偏微分で $\dot{x}_2 = \frac{\partial f}{\partial\theta_1}\dot{\theta}_1 + \frac{\partial f}{\partial\theta_2}\dot{\theta}_2$

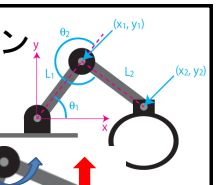
$\dot{y}_2 = \frac{\partial g}{\partial\theta_1}\dot{\theta}_1 + \frac{\partial g}{\partial\theta_2}\dot{\theta}_2$

•まとめると

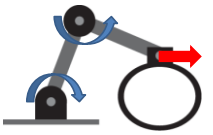
$$\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix} = \begin{bmatrix}\frac{\partial f}{\partial\theta_1} & \frac{\partial f}{\partial\theta_2}\\ \frac{\partial g}{\partial\theta_1} & \frac{\partial g}{\partial\theta_2}\end{bmatrix}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$$

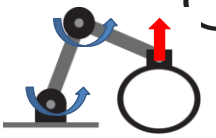
ヤコビアン

先端速度の実現とヤコビアン



ロボット末端をある速度で動かすための関節の角速度は？

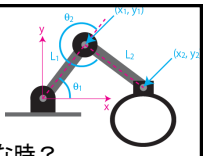




$$\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix} = \mathbf{J}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} \Rightarrow \begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} = \mathbf{J}^{-1}\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix}$$

•ヤコビアンの逆行列で求めることができる！
•逆行列がない場合は？

特異点



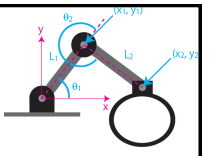
$$\mathbf{J} = \begin{bmatrix}-L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) & -L_2\sin(\theta_1+\theta_2)\\ L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) & L_2\cos(\theta_1+\theta_2)\end{bmatrix}$$

ヤコビアンの逆行列がないのはどんな時？
ad – bc = 0より

$$(-L_1\sin\theta_1 - L_2\sin\theta_{12})L_2\cos\theta_{12} + (L_1\cos\theta_1 + L_2\cos\theta_{12})L_2\sin\theta_{12} = 0$$

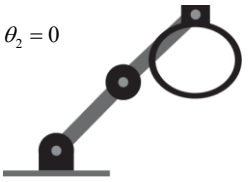
ただし $\theta_{12} = \theta_1 + \theta_2$

特異点

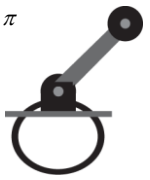


ヤコビアンの逆行列がないのは
$\theta_2 = 0, \pi$ のとき.

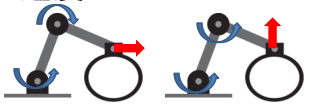
$\theta_2 = 0$



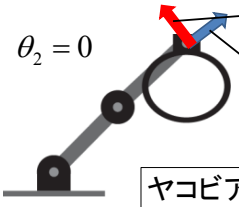
$\theta_2 = \pi$



特異点と速度



$\theta_2 = 0$

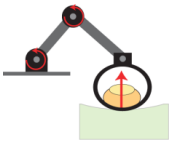


この方向には動かせる

関節をどう動かしても
この方向に動かせない

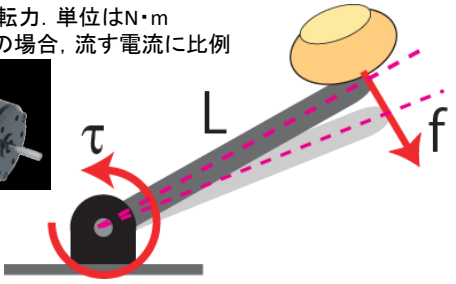
ヤコビアンの逆行列がない
＝動かせない方向あり（特異点）

先端力の実現とヤコビアン

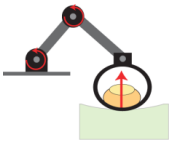


ロボット末端にある力を出すための，関節のトルクは？

トルク：回転力．単位はN・m
DCモータの場合，流す電流に比例



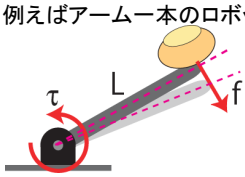
先端力の実現とヤコビアン



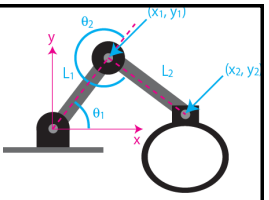
ロボット末端にある力を出すための，
関節のトルクは？

＜仮想仕事の原理＞を用いる
力 f で dx だけ微小変位したとき　仕事 = f･dx
トルク τ で dθ だけ微小回転したときの仕事 = τ･dθ

例えばアーム一本のロボットでは，このこつが釣り合うから，



2軸の場合



＜仮想仕事の原理＞を用いる
モータの出すトルクによる仕事：

$$W_{motor} =$$

先端の力による仕事：

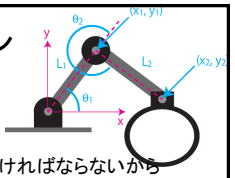
$$W_{hand} =$$

これが釣り合うから，$W_{motor}=W_{hand}$となり，

$$\left(\therefore \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right)$$

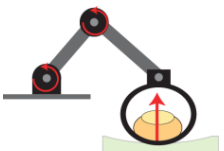
先端力の実現とヤコビアン

$$\begin{bmatrix} \tau_1 & \tau_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} f_x & f_y \end{bmatrix} \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$



これが任意の角速度 $\dot{\theta}_1, \dot{\theta}_2$ で成立しなければならないから

$\therefore$



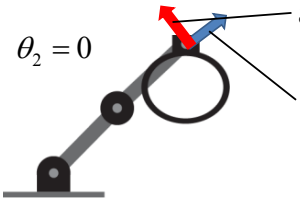
つまりロボット先端にある力を出したいときは，
ヤコビアンの転置をかけることにより，
関節に必要なトルクに変換できる．

特異点と先端力

モータにあるトルクを加えた時，先端にかかる力を求める．

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} f_x \\ f_y \end{bmatrix} \Leftrightarrow$$

ヤコビアンの逆行列がない＝力の出ない方向あり（特異点）

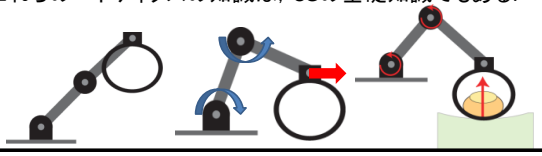


ロボティクスの基礎の基礎：まとめ

●ロボット各関節の角度，角速度，トルク

●ロボット先端の位置，速度，力

この二つは相互に変換可能である．
変換には単純な幾何学の知識と，ヤコビアンの知識が必要
これらのロボティクスの知識は，CGの基礎知識でもある．



以下参考資料

インピーダンス制御の問題点

$$(m+m')\ddot{x}+(c+c')\dot{x}+(k+k')x=ax_{goal}$$

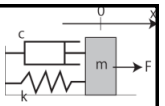
•m'=-m，c'=-c，k'=-kとしたら，
理想的には「無くなる」

⇒バネ，マス，ダンパの正確な値が必要（システム同定問題）

⇒パラメータは動的に変化することがある（摩擦の変化等）

⇒安定な制御は難しい

システム同定（周波数ベース）

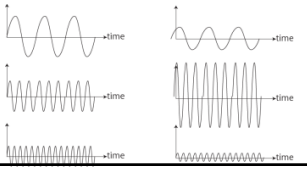


$$m\ddot{x}+c\dot{x}+kx=f$$

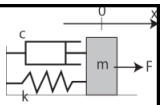
•既知の力 f を加えることによってm，c，kを求めたい．
•振幅1，角周波数ωの入力を加えた場合，

$$f=\exp(j\omega t)$$
$$x=X\exp(j\omega t)$$

•ただしXは複素数（位相遅れを含む）．

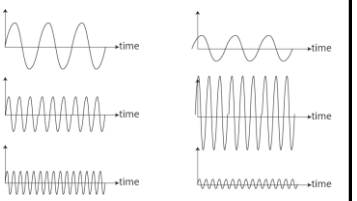


システム同定（周波数ベース）



$$m\ddot{x}+c\dot{x}+kx=f$$

$$\begin{cases} f=\exp(j\omega t) \\ x=X\exp(j\omega t) \end{cases}$$

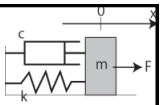


$$\dot{x}=j\omega X\exp(j\omega t)$$
$$\ddot{x}=(j\omega)^2X\exp(j\omega t)$$

$$(m(j\omega)^2+cj\omega+k)X\exp(j\omega t)=\exp(j\omega t)$$

$$X=\frac{1}{m(j\omega)^2+cj\omega+k}=\frac{1}{(k-m\omega^2)+j\omega c}$$

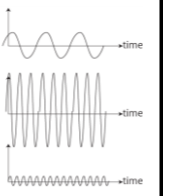
システム同定（周波数ベース）



$$X=\frac{1}{m(j\omega)^2+cj\omega+k}=\frac{1}{(k-m\omega^2)+j\omega c}$$

$$\left|\frac{1}{X}\right|=\left|(k-m\omega^2)+j\omega c\right|$$

$$\left|\frac{1}{X}\right|^2=(k-m\omega^2)^2+(\omega c)^2$$
$$=m^2\omega^4+(c^2-2km)\omega^2+k^2$$



•左辺：計測結果Xの絶対値（振幅）の逆数の二乗
•右辺：未知係数m²，$(c^2-2km)^2$，k^2を係数とし，
ω^2を変数とした2次の多項式．

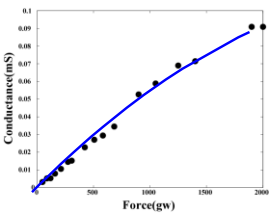
（復習）多項式近似

$$y=a_1x^2+a_2x+a_3$$

- x：力：既知入力
- y：測定出力
- $a1, a2, a3$：未知パラメータ

何度も測定する

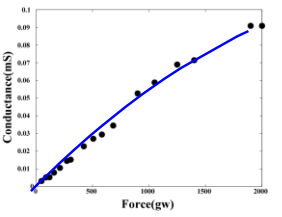
$$y_1=a_1x_1^2+a_2x_1+a_3$$
$$y_2=a_1x_2^2+a_2x_2+a_3$$
$$\vdots$$
$$y_M=a_1x_M^2+a_2x_M+a_3$$



（復習）多項式近似

$$y_1=a_1x_1^2+a_2x_1+a_3$$
$$y_2=a_1x_2^2+a_2x_2+a_3$$
$$\vdots$$
$$y_M=a_1x_M^2+a_2x_M+a_3$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}=\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_M^2 & x_M & 1 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

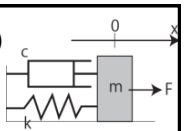


$\mathbf{y}=\mathbf{Xa}$ の形に出来たので，

$\mathbf{a}=\mathbf{X}^{\#}\mathbf{y}$ where $\mathbf{X}^{\#}=(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}$

により3つの未知パラメータを求めることが出来る．

システム同定（周波数ベース）



$$\left|\frac{1}{X}\right|^2 = m^2(\omega^2)^2 + (c^2 - 2km)\omega^2 + k^2$$

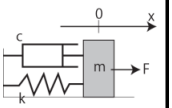
- 左辺： 計測結果Xの絶対値の逆数の二乗
- 右辺： 未知係数：m^2，$(c^2\text{-}2km)^2$，k^2
  変数：ω^2
  の2次の多項式．

- 角周波数ωを変化させて出力振幅を計測することで，多項式近似により未知係数m^2，$(c^2\text{-}2km)^2$，k^2を求めることができる．

- 最終的にm,c,kを計算する

システム同定（時間ベース，オンライン推定）

$$m\ddot{x} + c\dot{x} + kx = f$$



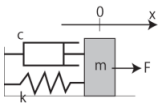
- 実際の稼働中にパラメータを推定する．（オンライン推定）

- 結局時間はデジタル．刻々と変化する力，位置，速度，加速度を記録する．

$$\mathbf{f} = \begin{bmatrix} f_1 & f_2 & \cdots & f_N \end{bmatrix}$$
$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}$$
$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 & \dot{x}_2 & \cdots & \dot{x}_N \end{bmatrix}$$
$$\ddot{\mathbf{x}} = \begin{bmatrix} \ddot{x}_1 & \ddot{x}_2 & \cdots & \ddot{x}_N \end{bmatrix}$$

システム同定（時間ベース，オンライン推定）

$$\begin{aligned} m\ddot{x}_1 + c\dot{x}_1 + kx_1 &= f_1 \\ m\ddot{x}_2 + c\dot{x}_2 + kx_2 &= f_2 \\ &\vdots \\ m\ddot{x}_N + c\dot{x}_N + kx_N &= f_N \end{aligned}$$



$$\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} = \begin{bmatrix} \ddot{x}_1 & \dot{x}_1 & x_1 \\ \ddot{x}_2 & \dot{x}_2 & x_2 \\ \vdots & \vdots & \vdots \\ \ddot{x}_N & \dot{x}_N & x_N \end{bmatrix} \begin{bmatrix} m \\ c \\ k \end{bmatrix}$$

- 擬似逆行列を用いることによりパラメータm，c，kを求めることができる．

インピーダンス制御の問題点（再掲）

$$(m+m')\ddot{x} + (c+c')\dot{x} + (k+k')x = ax_{goal}$$

- m′=-m，c′=-c，k′=-kとしたら理想的には「無くなる」

⇒バネ，マス，ダンパの正確な値が必要（システム同定問題）

⇒パラメータは動的に変化することがある（摩擦の変化等）

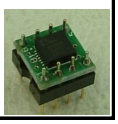
⇒安定な制御は難しい

安定な制御の難しさ

- システム $m\ddot{x} + c\dot{x} + kx = f$
- 制御則 $f = k'(x_{goal} - x) - c'\dot{x} - m'\ddot{x}$
- 結果 $(m+m')\ddot{x} + (c+c')\dot{x} + (k+k')x = ax_{goal}$

制御のためには，位置，速度，加速度の計測が必要．
- ✓位置：エンコーダ（回転角センサ）
- ✓速度：大抵は位置を**微分**して求める．場合によりタコジェネレータ
- ✓加速度：大抵は速度を**微分**して求める．場合により加速度センサ

速度，加速度は位置ほど正確に得られず，
ノイズを多く含む．



安定な制御の難しさ

- システム $m\ddot{x} + c\dot{x} + kx = f$
- 制御則 $f = k'(x_{goal} - x) - c'\dot{x} - m'\ddot{x}$
- 結果 $(m+m')\ddot{x} + (c+c')\dot{x} + (k+k')x = ax_{goal}$

速度，加速度は位置ほど正確に得られず，
ノイズを多く含む．

⇒不安定な振動
⇒重さ（加速度項）の完全な補償(m′=-m)は無理．
経験的には半分程度「軽く」するのがせいぜい．

⇒システムは設計段階から軽く作る必要．

位置制御ベースインピーダンス制御へ

- 従来の手法
  - ●システム　$m\ddot{x}+c\dot{x}+kx=f$
  - ●制御則　$f=k'(x_{goal}-x)-c'\dot{x}-m'\ddot{x}$
  - ●結果　$(m+m')\ddot{x}+(c+c')\dot{x}+(k+k')x=ax_{goal}$
- システム同定が必要（m, c, k)
- 位置だけでなく，速度，加速度を求める必要
- 発振等の不都合を生じやすい

そもそも何をしたかったのか？

位置制御ベースインピーダンス制御へ

- そもそもやりたかったこと:

「固い」ロボットに，
まるで人のような「柔らかい」振りをさせたい．

- つまり，　$m_o\ddot{x}+c_o\dot{x}+k_ox=f$

という元々の固さ（動特性）は無視して，あたかも

$$m_s\ddot{x}+c_s\dot{x}+k_sx=f$$

という動特性をもったロボットであるかのように「振舞わせれば」よい．

（o: original, s: simulated)

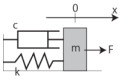
位置制御ベースインピーダンス制御

（1）外力 f を計測する．(力センサ）

（2）もしロボットが，

$$m_s\ddot{x}+c_s\dot{x}+k_sx=f$$

というインピーダンスを持っていたらこう動く**はず**，
という**未来の軌道**をシミュレーション．



（3）軌道に沿った運動をPD(PID)制御で実現．
P, Dゲインは非常に高く設定し，軌道を忠実に再現

⇒まるで柔らかいようにふるまう

位置制御ベースインピーダンス制御

（短所）
- カセンサが必要．
- カセンサはロボットアームの先端に付けることが多い
  ⇒先端以外を押しても反応しない．
- カセンサもノイズが大きい．
- 早い制御ループが必要．

（長所）
- システム同定が不要．
- ものすごく「固い」産業用ロボットも柔らかく出来る．
- コンピュータの発達した今向きと言える．