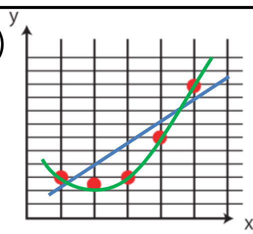
認識行動システム論
第12回

梶本裕之
Twitter ID kajimoto
ハッシュタグ #ninshiki

日程

- 10/07 イントロダクション
- 10/14 Scilabの紹介（3階PCルームにて）
- 10/21 フーリエ変換
- 10/28 フーリエ変換と線形システム
- 11/04 信号処理の基礎
- 11/11 信号処理応用1（相関）
- 11/18 調布祭準備のため休講
- 11/25 信号処理応用2（画像処理）
- 12/02 中間テスト(授業時間中)
- 12/09 ラプラス変換
- 12/16 古典制御の基礎
- 01/06 行列
- 01/13 行列と最小二乗法
- 01/20 ロボティクス
- 01/27 期末テスト(授業時間中)

前回のレポート課題(1)



次のデータ系列に対して，
Scilabを用いて，
(1) 直線による近似，
(2) 2次曲線による近似を適用，
パラメータを求め，
曲線とデータをグラフに描け

x	1.0	2.0	3.0	4.0	5.0
y	3.0	2.5	3.0	6.0	10.0

なおScilabでは行列Aの擬似逆行列はpinv(A)で直接求めることができる．
当然自分でinv(A'*A)*A'とやっても同じ．

(1-1)直線による近似

$$y = a_1 x + a_2$$

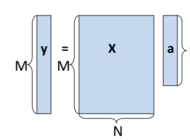
これは

$$y = a_1 x_1 + a_2 x_2 \quad \text{where} \quad x_2 = 1$$

とみなせる．

$$y_1 = a_1 x_{11} + a_2 1$$
$$y_2 = a_1 x_{21} + a_2 1$$
$$\vdots$$
$$y_M = a_1 x_{M1} + a_2 1$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} x_{11} & 1 \\ x_{21} & 1 \\ \vdots & \vdots \\ x_{M1} & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$



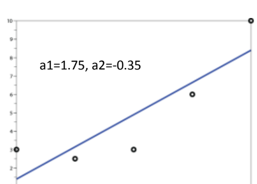
よって，

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

により二つの未知パラメータを求めることが出来る．

(1-1)直線による近似

$$\begin{bmatrix} 3.0 \\ 2.5 \\ 3.0 \\ 6.0 \\ 10.0 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ 4 & 1 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$



Scilabコード例

```
x=[1;2;3;4;5];
y=[3.0; 2.5; 3.0; 6.0; 10.0];
plot2d(x,y,style=[-9] ); //元のデータをプロット

X=[1,1;2,1;3,1;4,1;5,1];          //擬似逆行列のための行列
a=inv(X'*X)*X'*y                  //係数を最小二乗法で求める.
y_fitting = a(1) * x + a(2);      //フィッティング結果の直線
plot(x,y_fitting);                //表示
```

(1-2)２次曲線による近似

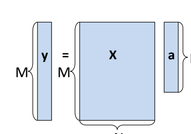
$$y = a_1 x^2 + a_2 x + a_3$$

これは

$$y = a_1 x_1 + a_2 x_2 + a_3 x_3$$

とみなせる．

$$y_1 = a_1 x_{11} + a_2 x_{12} + a_2 1$$
$$y_2 = a_1 x_{21} + a_2 x_{22} + a_2 1$$
$$\vdots$$
$$y_M = a_1 x_{M1} + a_2 x_{M2} + a_2 1$$



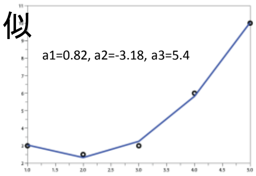
よって，

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

により二つの未知パラメータを求めることが出来る．

（1-2）2次曲線による近似

$$\begin{bmatrix}3.0\\2.5\\3.0\\6.0\\10.0\end{bmatrix}=\begin{bmatrix}1&1&1\\4&2&1\\9&3&1\\16&4&1\\25&5&1\end{bmatrix}\begin{bmatrix}a_1\\a_2\\a_3\end{bmatrix}$$

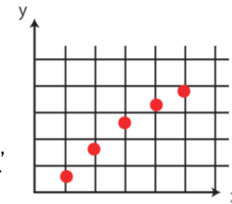


Scilabコード例

```
x=[1;2;3;4;5];
y=[3.0; 2.5; 3.0; 6.0; 10.0];
plot2d(x,y,style=[-9] );              //元のデータをプロット

X=[1,1,1;4,2,1;9,3,1;16,4,1;25,5,1];  //擬似逆行列のための行列
a=inv(X'*X)*X'*y                      //係数を最小二乗法で求める.
y_fitting = a(1).*x .*x + a(2)*x +a(3); //フィッティング結果を示す
plot(x,y_fitting);                    //表示
```

前回のレポート課題(2)

次のデータ系列に対して,
y=a1 * log(x) + a2
を仮定してパラメータを求め,
曲線とデータをグラフに描け
（やや難？）



x	1.0	2.0	3.0	4.0	5.0
y	0.5	1.9	2.7	3.3	3.7

8

（2）logによる近似

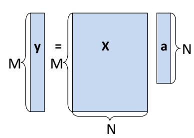
$$y=a_1\log(x)+a_2$$

これは

$$y=a_1x_1+a_2x_2 \quad \text{where} \quad x_2=1$$

とみなせる.

$$\begin{aligned}y_1&=a_1\log(x_{11})+a_2 1\\y_2&=a_1\log(x_{21})+a_2 1\\&\vdots\\y_M&=a_1\log(x_{M1})+a_2 1\end{aligned}\qquad\begin{bmatrix}y_1\\y_2\\\vdots\\y_M\end{bmatrix}=\begin{bmatrix}x_{11}&1\\x_{21}&1\\\vdots&\vdots\\x_{M1}&1\end{bmatrix}\begin{bmatrix}a_1\\a_2\end{bmatrix}$$



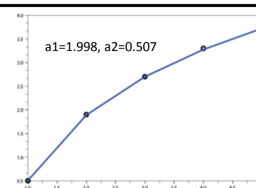
よって,

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

により二つの未知パラメータを求めることが出来る.

（2）logによる近似

$$\begin{bmatrix}0.5\\1.9\\2.7\\3.3\\3.7\end{bmatrix}=\begin{bmatrix}\log(1)&1\\\log(2)&1\\\log(3)&1\\\log(4)&1\\\log(5)&1\end{bmatrix}\begin{bmatrix}a_1\\a_2\end{bmatrix}$$



Scilabコード例

```
x=[1;2;3;4;5];
y=[0.5; 1.9; 2.7; 3.3; 3.7];
plot2d(x,y,style=[-9] );           //元のデータをプロット

//擬似逆行列のための行列
X=[log(1),1; log(2),1; log(3),1; log(4),1;log(5),1];
a=inv(X'*X)*X'*y                   //係数を最小二乗法で求める.
y_fitting = a(1)*log(x)+ a(2);     //フィッティング結果を示す
plot(x,y_fitting);                 //表示
```

10

レポートの成績評価とは関係なく,
これらの課題は，一度は打ち込んで
結果を見ることを強く勧めます.

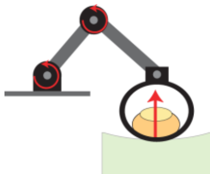
最小二乗法は信号処理，制御，統計
などあらゆる分野で使われています.

11

ロボティクスの基礎の基礎：

ロボットの姿勢・力・速度

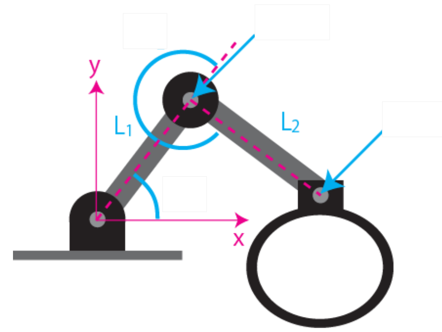
座標変換の必要性



- 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．

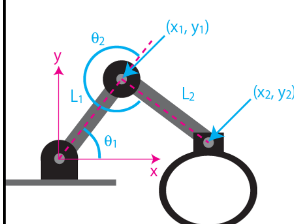
- ロボット末端をある速度で動かすための関節の速度は？
- ロボット末端にある力を出すための，関節のトルクは？

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．



$x_1 =$

$y_1 =$

$x_2 =$

$=$

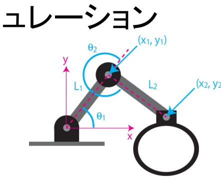
$y_2 =$

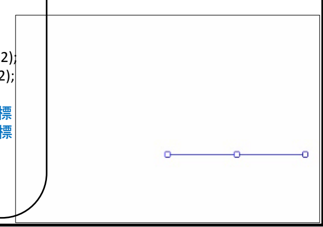
$=$

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
```





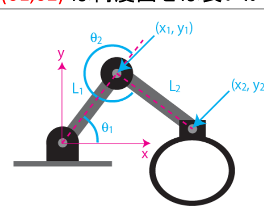
POSER中の順キネマティクス

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

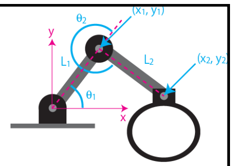
ロボット末端の位置を(x2,y2) に移動したい．
関節の角度(θ1,θ2) は何度回せば良いか？



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

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

逆キネマティクス



θ1+θ2をθ12と書いて

$$x_2 = L_1 \cos\theta_1 + L_2 \cos\theta_{12}$$
$$y_2 = L_1 \sin\theta_1 + L_2 \sin\theta_{12}$$

$x_2{}^2 + y_2{}^2 =$

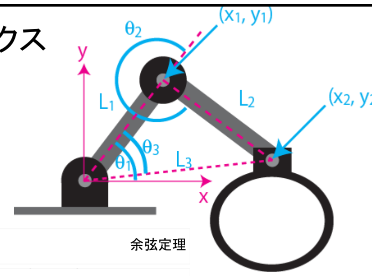
$=$

$=$ 　　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

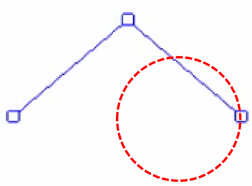
$$L_3 = \sqrt{x_2{}^2 + y_2{}^2}$$

$L_2{}^2 =$ 　　余弦定理

$\theta_3 =$

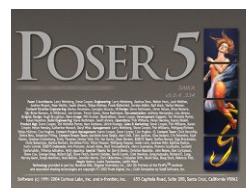
$\theta_1 =$

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



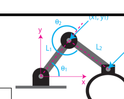
ロボットの知識はCGアニメーションに必須．
（基礎的な知識はほぼ共通）

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．

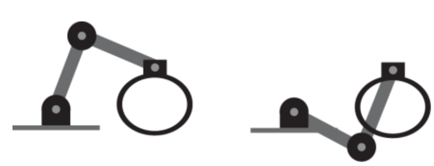
```
L1 = 1.0;
L2 = 1.0;

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 =

 //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



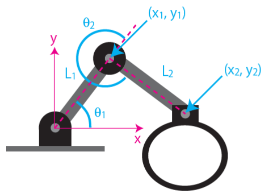
ロボット末端の位置を(x2,y2) に移動したい．
関節の角度(θ1,θ2) は何度回せば良いか？

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．<u>とても大変</u>．
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.



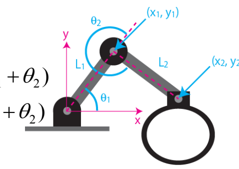
$$x_2 = L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2)$$
$$y_2 = L_1\sin\theta_1 + L_2\sin(\theta_1+\theta_2)$$

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

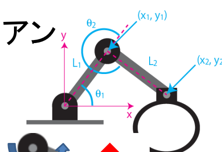
ヤコビアン

- 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

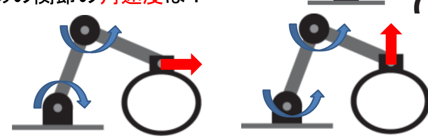
$$\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix} = \boxed{\begin{bmatrix}\frac{\partial f}{\partial\theta_1} & \frac{\partial f}{\partial\theta_2}\\ \frac{\partial g}{\partial\theta_1} & \frac{\partial g}{\partial\theta_2}\end{bmatrix}}\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}$$

ヤコビアン

先端速度の実現とヤコビアン



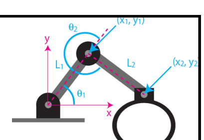
ロボット末端をある速度で動かす
ための関節の角速度は？



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

- 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



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

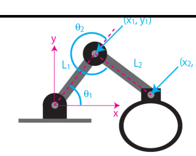
ヤコビアンの逆行列がないのはどんな時？

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

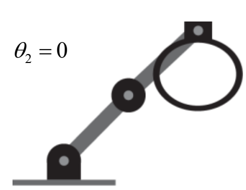
特異点



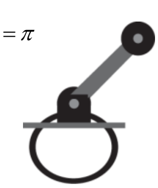
ヤコビアンの逆行列がないのは

$\theta_2 = 0, \pi$ のとき.

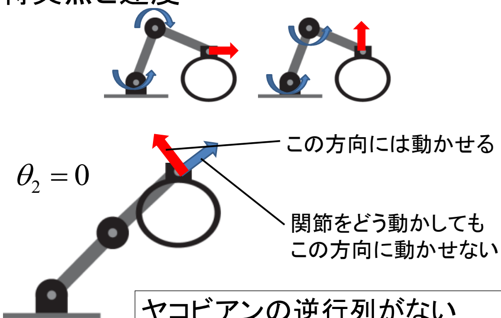
$\theta_2 = 0$



$\theta_2 = \pi$

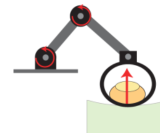


特異点と速度



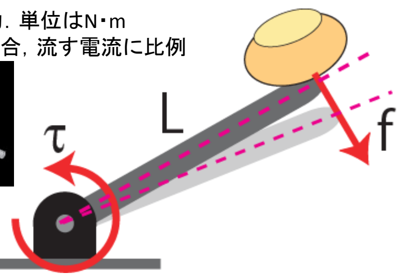
ヤコビアンの逆行列がない
＝動かせない方向あり（特異点）

先端力の実現とヤコビアン

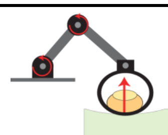


ロボット末端にある力を出すための，
関節のトルクは？

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



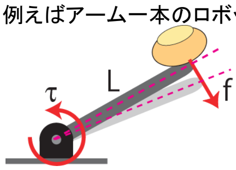
先端力の実現とヤコビアン



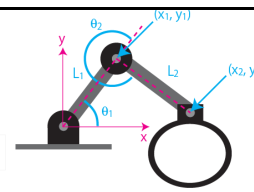
ロボット末端にある力を出すための，
関節のトルクは？

＜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，この二つが釣り合うから，



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：

$W_{hand} =$

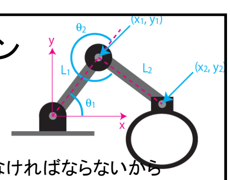
これが釣り合うから，Wmotor=Whandとなり，

$= \quad =$

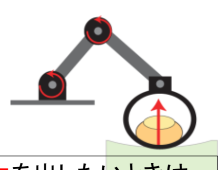
$\left(\therefore \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} \right)$

先端力の実現とヤコビアン

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



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



$\therefore$

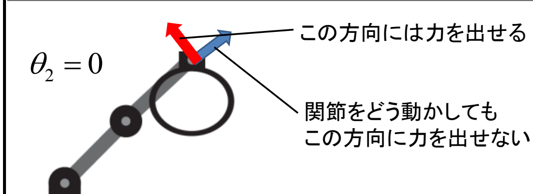
つまりロボット先端にある力を出したいときは，
ヤコビアンの転置をかけることにより，
関節に必要なトルクに変換できる．

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．

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

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．