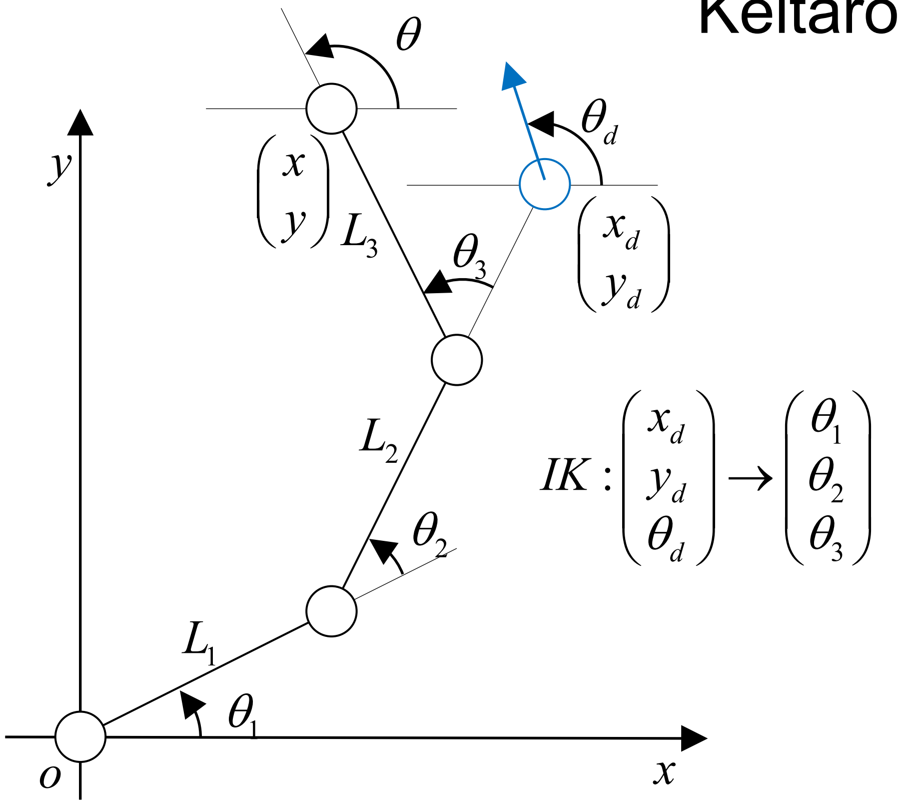
ロボットアームの逆運動学: 原理
Robot Arm Inverse Kinematics: Principle

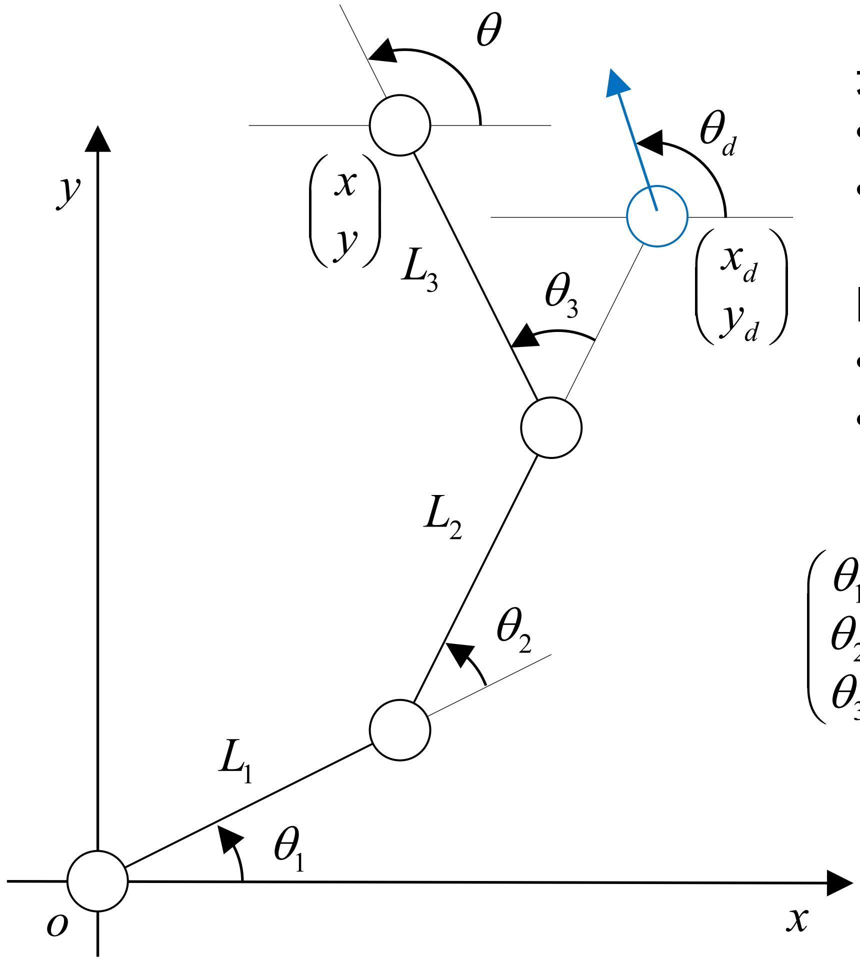
成瀬継太郎（会津大）
Keitaro Naruse (Univ. of Aizu)



まとめ
- 逆運動学：ロボットアームの目標の手先姿勢（位置と向き）からそれを実現する関節角度を求めること

Summary
- Inverse kinematics: Find joint angles which satisfy a target hand pose of a root arm

ロボットアームの逆運動学
Inverse Kinematics of Robot Arm



逆運動学
- 入力：手先の姿勢（位置と向き）
- 出力：関節角度

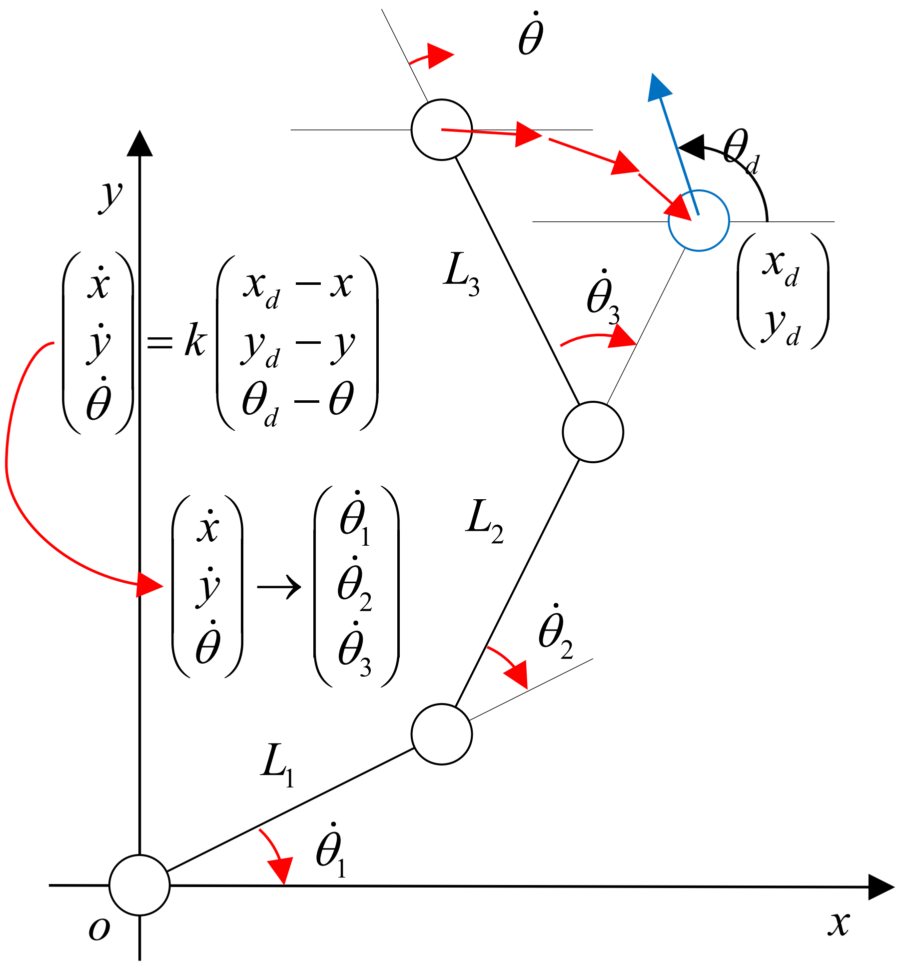
Inverse kinematics
- Given: A hand pose (position and orientation)
- Find: Joint angles

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = \begin{pmatrix} f_1(x_d, y_d, \theta_d) \\ f_2(x_d, y_d, \theta_d) \\ f_3(x_d, y_d, \theta_d) \end{pmatrix}$$

閉じた形式での逆運動学の解は
特殊な場合しか存在しない
We can find a closed form inverse kinematics solution only in a special case
一般的に繰り返し計算で求める
Generally, we solve it by iterative computation

繰り返し計算による逆運動学解法

Inverse Kinematics Solution by Iterative Computation

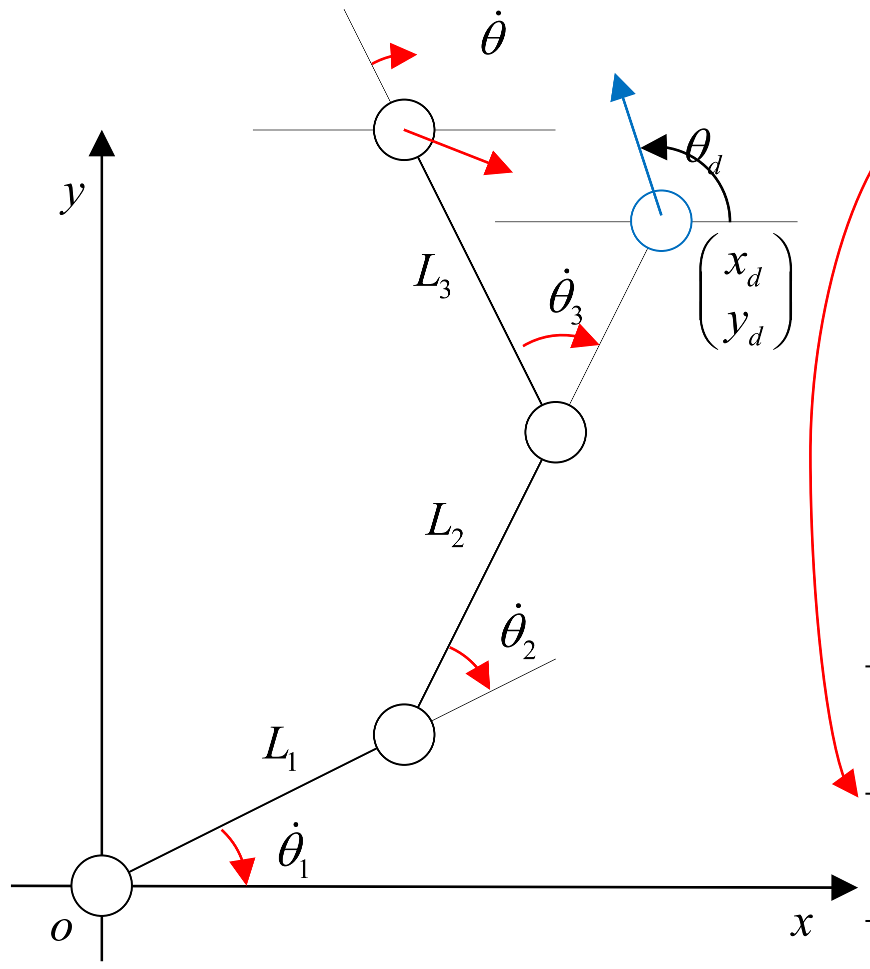


アイディア
- 現在から目標への手先姿勢の差を，手先の姿勢速度と解釈する
- 手先姿勢速度から関節角速度に変換する
- この関節角速度から関節角度の変位を求め，姿勢を変化させる
- これを繰り返すことにより，手先を目標位置に到達させる

Idea
- Interpret difference from current one to target pose as hand pose velocity
- Convert the hand pose velocity to joint angular one
- Find joint angle displacement from the joint angular velocity, and update a robot arm pose
- Iterating it, reach a hand at a target pose eventually

$$\begin{pmatrix}\theta_1(t+\Delta t)\\ \theta_2(t+\Delta t)\\ \theta_3(t+\Delta t)\end{pmatrix} = \begin{pmatrix}\theta_1(t)\\ \theta_2(t)\\ \theta_3(t)\end{pmatrix} + \Delta t\begin{pmatrix}\dot{\theta}_1(t)\\ \dot{\theta}_2(t)\\ \dot{\theta}_3(t)\end{pmatrix}$$

手先姿勢速度と関節角速度とヤコビアン
Hand Pose Velocity, Joint Angular Velocity, and Jacobian



$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$
$$\theta = \theta_1 + \theta_2 + \theta_3$$

順運動学関数の両辺を時間で微分することで，手先姿勢速度と関節角速度を，ヤコビアンと呼ばれる行列によって関連付けることができる

Taking time derivative of both sides of a forward kinematics function, we can associate hand velocity and joint angle one, and it can be represented in a matrix called Jacobian

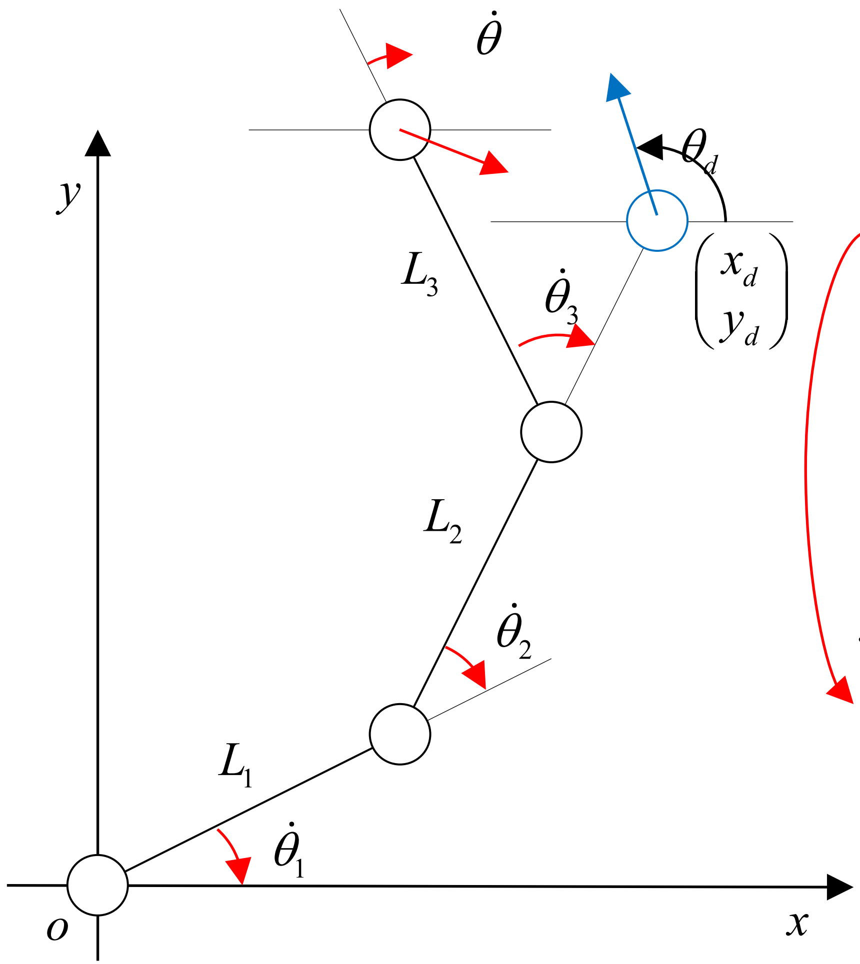
$$\frac{dx}{dt} = \frac{\partial x}{\partial \theta_1}\frac{d\theta_1}{dt} + \frac{\partial x}{\partial \theta_2}\frac{d\theta_2}{dt} + \frac{\partial x}{\partial \theta_3}\frac{d\theta_3}{dt}$$
$$\frac{dy}{dt} = \frac{\partial y}{\partial \theta_1}\frac{d\theta_1}{dt} + \frac{\partial y}{\partial \theta_2}\frac{d\theta_2}{dt} + \frac{\partial y}{\partial \theta_3}\frac{d\theta_3}{dt}$$
$$\frac{d\theta}{dt} = \frac{\partial \theta}{\partial \theta_1}\frac{d\theta_1}{dt} + \frac{\partial \theta}{\partial \theta_2}\frac{d\theta_2}{dt} + \frac{\partial \theta}{\partial \theta_3}\frac{d\theta_3}{dt}$$

手先姿勢速度
Hand pose velocity

関節角速度
Joint angular velocity

$$\longrightarrow \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial \theta}{\partial \theta_1} & \frac{\partial \theta}{\partial \theta_2} & \frac{\partial \theta}{\partial \theta_3} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = J \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix}$$

ヤコビアンの例
Instance of Jacobian



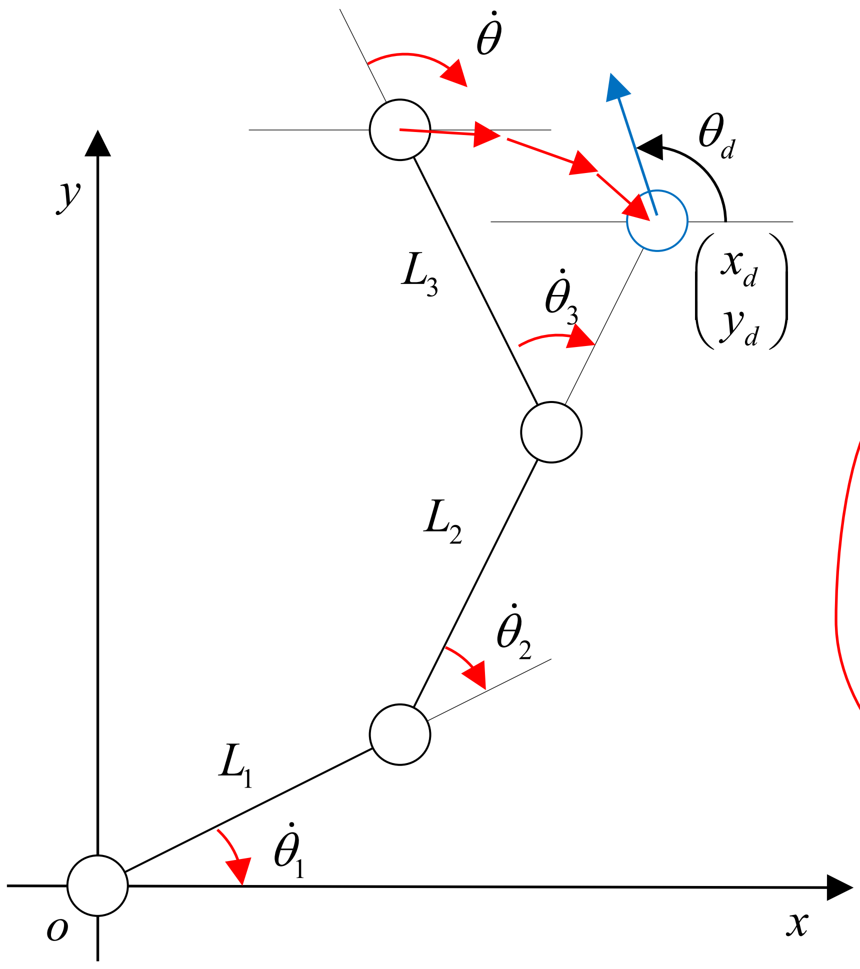
順運動学関数 Forward kinematics function

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1+\theta_2) + L_3 \cos(\theta_1+\theta_2+\theta_3)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1+\theta_2) + L_3 \sin(\theta_1+\theta_2+\theta_3)$$
$$\theta = \theta_1+\theta_2+\theta_3$$

対応するヤコビアン Corresponding Jacobian

$$J = \begin{pmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \\ \frac{\partial \theta}{\partial \theta_1} & \frac{\partial \theta}{\partial \theta_2} & \frac{\partial \theta}{\partial \theta_3} \end{pmatrix}$$
$$= \begin{pmatrix} -L_1\sin\theta_1 - L_2\sin(\theta_1+\theta_2) - L_3\sin(\theta_1+\theta_2+\theta_3) & -L_2\sin(\theta_1+\theta_2) - L_3\sin(\theta_1+\theta_2+\theta_3) & -L_3\sin(\theta_1+\theta_2+\theta_3) \\ L_1\cos\theta_1 + L_2\cos(\theta_1+\theta_2) + L_3\cos(\theta_1+\theta_2+\theta_3) & L_2\cos(\theta_1+\theta_2) + L_3\cos(\theta_1+\theta_2+\theta_3) & L_3\cos(\theta_1+\theta_2+\theta_3) \\ 1 & 1 & 1 \end{pmatrix}$$

ヤコビアンを用いた逆運動学解法
Inverse Kinematics Solution with Jacobian



関節角度の更新:手先姿勢の変位→手先姿勢速度
→関節角速度→関節角度の変位
Joint angle update: Hand pose displacement > Hand pose velocity
> Joint angular velocity > Joint angle displacement

$$\begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} \leftarrow \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} + \Delta t \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} \leftarrow \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{pmatrix} = J^{-1} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} \leftarrow \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = k \begin{pmatrix} x_d - x \\ y_d - y \\ \theta_d - \theta \end{pmatrix}$$

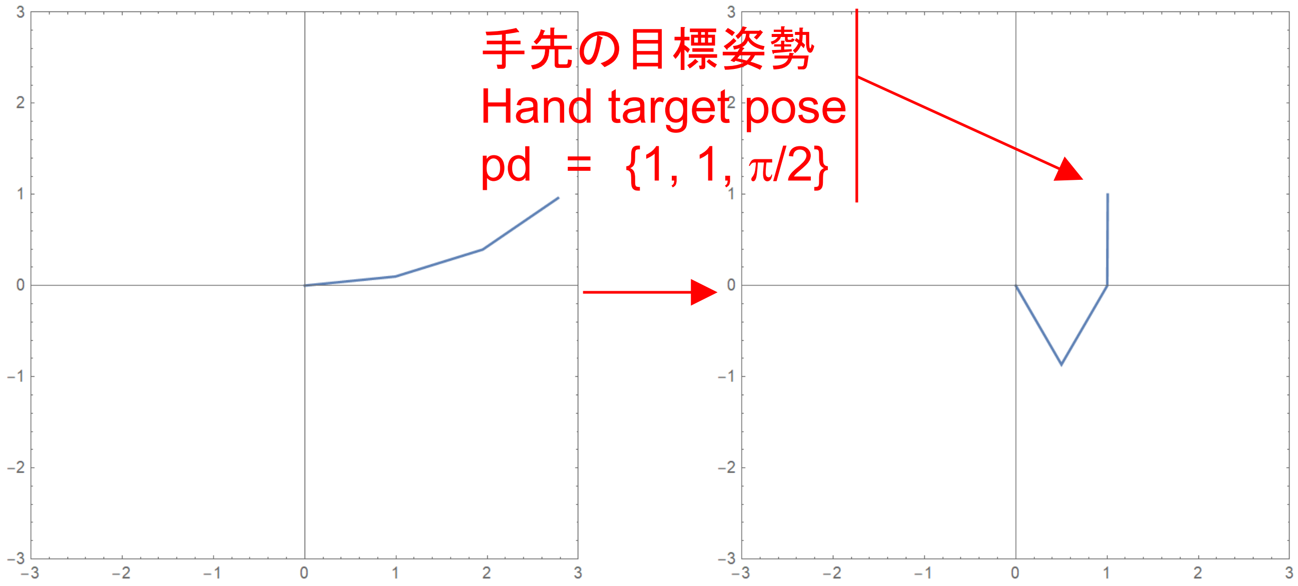
順運動学関数:関節角度→手先姿勢
Forward kinematics: Joint angles > hand pose

$$x = L_1 \cos\theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3)$$
$$y = L_1 \sin\theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3)$$
$$\theta = \theta_1 + \theta_2 + \theta_3$$

平面3リンクロボットアームの逆運動学のサンプルコード
Mathematica Sample Code of Inverse Kinematics of Planar 3-Link Robot Arm

初期姿勢
Initial pose
q0 = {0.1, 0.2, 0,3}

逆運動学解
Inverse kinematics solution
q = {-2.09196, 4.18453, -2.08948}



マセマティカによるサンプルコード(一部のみ)
Sample code in Mathematica (Only a part)

```
IK-3LinkPlanarArm.nb
(*Inverse kinematics solution: initialize*)
k = 0.1;
dt = 1;
q0 = {0.1, 0.2, 0.3};
pd = {1, 1, pi/2};

(*Inverse kinematics solution by function*)
q = IK[pd, q0, k, dt]

(*Inverse kinematics function*)
IK[pd_, q0_, k_, dt_] :=
 Module[{p, q = q0},
  For[i = 1, i <= 50, i++,
   p = FK[q];
   q += k dt Inverse[J[q]].(pd - p[[4]]);
   ];
  q]
```

Code is available at https://github.com/keitaronaruse/Naruse-robotics-tutorial/blob/main/src/mathematica/IK-3LinkPlanarArm.nb

マセマティカによる逆運動学のデモンストレーション

Demonstration of Inverse Kinematics by Mathematica