

# ヒューマノイドロボット勉強会

---

逆運動学



# 前回

---

- 順運動学を計算するプログラムを実装, 3リンクマニピュレータの手先の位置・姿勢を計算してみた
- エクセル上にてプログラムの計算結果があっているか検証した



# 今回

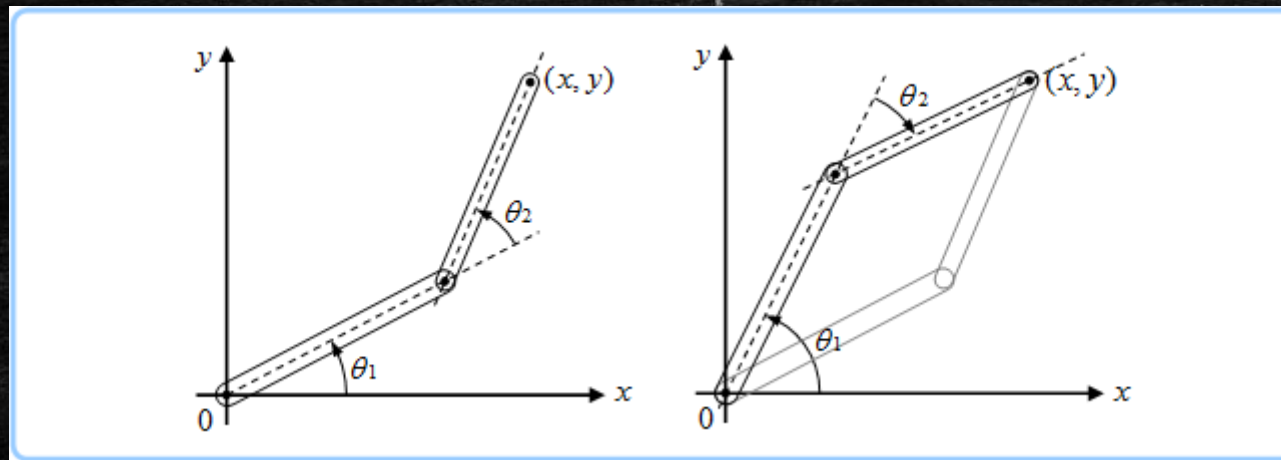
---

- 逆運動学について
- 今回6自由度の脚を持つヒューマノイドロボット対象に片足の関節角度を逆運動学を用いて計算してみる



# 逆運動学

- 手先位置の位置・姿勢から各関節角度の計算を行う
- 一般的には、解析的には解が一意に求まることは保証されない  
⇒自由度のせい
- 機構や自由度が増えると計算式も複雑化し、解析解法では汎用性 0





# 数値的解法

---

- 最適化手法を用いる

⇒収束計算により少しずつ目標値に近づけていく

- 最適化手法にも色々あるらしいです



# 最適化手法

---

- 最急降下法
- Newton-Raphson法
- Levenberg-Marquardt法
- Particle IK
- CCD IK

とか



# 最適化手法

---

- 最急降下法
- **Newton-Raphson法**
- Levenberg-Marquardt法
- Particle IK
- CCD IK

とか



# 一般的な数値解法の計算アルゴリズム

---

1. 目標リンクの位置・姿勢を与える
2. 胴体から目標リンクまでの関節角度を並べたベクトルを $q$ としておく
3. 順運動学によりリンクの位置・姿勢を更新
4. 位置・姿勢の誤差を計算
5. 誤差が十分小さければ計算終了⇒解の提示
6. 誤差が大きければこれを小さくするための関節角度修正量 $\Delta q$ を計算
7.  $q = q + \Delta q$ として、3に戻る



# 一般的な数値解法の計算アルゴリズム

---

1. 目標リンクの位置・姿勢を与える
2. 胴体から目標リンクまでの関節角度を並べたベクトルを $q$ としておく
3. 順運動学によりリンクの位置・姿勢を更新
4. 位置・姿勢の誤差を計算
5. 誤差が十分小さければ計算終了⇒解の提示
6. 誤差が大きければこれを小さくするための関節角度修正量 $\Delta q$ を計算
7.  $q = q + \Delta q$ として、3に戻る



# 誤差が十分小さいとは

---

- 最終的に誤差が0となるのが理想  
⇒0ぴったりになることはないと思う
- あらかじめ小さい値を定義( $0.0000000001$ とか?)しておき, 誤差がそれ以下になったら十分小さくなったとみなす

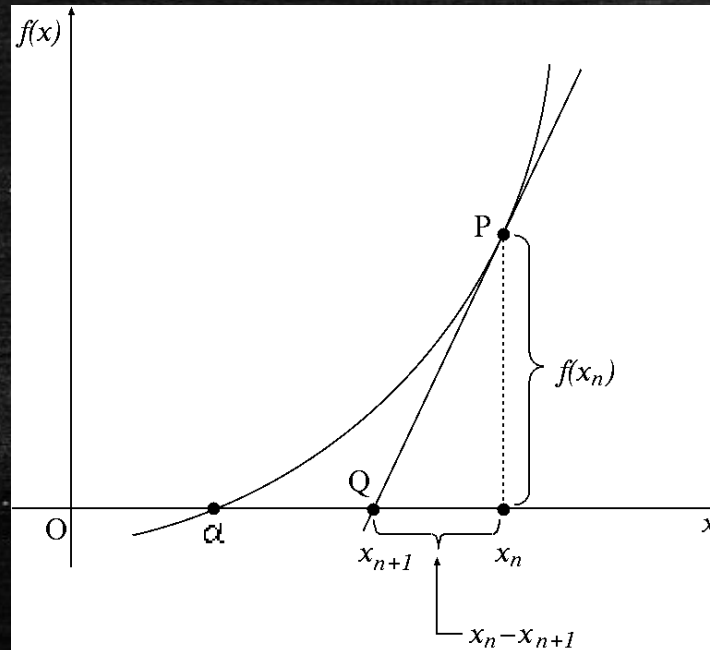


# 関節角度修正量はどのように計算するか

- ここでNewton-Raphson法

⇒関数 $f(x)$ が $x$ 軸と交わる点 $x_a$ の近似値を反復計算により得る

⇒グラフの接線を考え、その $x$ 切片を計算





# ヤコビアン

---

- ロボットの微小な関節角度と空間運動の関係を表す行列  
⇒ 関節トルクとEEFの外力との関係を表すことも可、様々な分野で出てきます(めっちゃ重要)
- 逆運動学では、EEFの現在の位置・姿勢と目標値の誤差から各関節の微小な角度変化量を計算するのに使用される



# ヤコビアン の 計算方法

---

- ある関節を微小に変化させた時のEEFの位置・姿勢の変化を計算する  
⇒図2.30のようにベクトルを計算してEEFの位置・姿勢の変化量を  
算出
- これを全ての関節において計算したものを足し合わせるにより, 実際の  
位置・姿勢の変化量を表すことが出来る(2.71)



# 関節角度の修正量

---

- 関節角度の修正量を求めたい場合はヤコビアンの逆行列を計算したものを  
使えば簡単に計算できます(Eigen使えば秒殺)

(2.70)

- これは自由度とEEFの自由度が一致してれば問題ないが冗長マニピュ  
レータ(7自由度とか)になるとヤコビアンが正方行列ではなくなる

⇒疑似逆行列を計算する必要がある



## 実際にコードを見してみる

---

- `open-rdc/HumanoidRobotLibrary/Kinematics`を更新しました
- cloneしてコンパイルしてみてください