

手先軌道制御について

三武裕玄

2022 年 7 月 24 日

1 目的と概要

2 記号・用語の定義

Δt : シミュレーションの 1 ステップあたりの時間
刻み. したがってステップ n における時刻は
 $n\Delta t$ である.

$\text{traj}(t)$: 手先軌道. 時刻 t の手先通過位置を表す.

\mathbf{r}_n : ステップ n における目標手先位置.

θ_n : ステップ n における目標関節角度.

ω_n : ステップ n における目標関節角速度.

\mathbf{p}_n : ステップ n における手先位置.

\mathbf{q}_n : ステップ n における関節角度.

$J(\theta)$: 関節角度 θ における関節角微小変位 $\Delta\theta$ を
手先位置微小変位 $\Delta\mathbf{r}$ に変換するヤコビアン.
 $\Delta\theta = J(\theta)\Delta\mathbf{r}$.

3 逆運動学

逆運動学を用いて, 手先軌道から目標関節角を求
める. シミュレーション 1 ステップあたりの手先移
動量は微小であるため, ヤコビアンの擬似逆行列に
よる微分逆運動学を 1 ステップあたり 1 回行うこ
とで実現する.

3.1 目標関節角度の計算

ステップ n から $n+1$ にかけての目標手先位置の
変化は

$$\begin{aligned} \mathbf{r}_{n+1} - \mathbf{r}_n \\ (= \text{traj}(n\Delta t + \Delta t) - \text{traj}(n\Delta t)) \end{aligned}$$

である. ここで Δt が十分小さく, 比較的滑らかな
手先軌道ならば $\text{traj}(t + \Delta t) - \text{traj}(t)$ は微小である
と想定される. したがってステップ間の目標手先位

置変化 $\mathbf{r}_{n+1} - \mathbf{r}_n$ と目標関節角度変化 $\theta_{n+1} - \theta_n$ の関
係はヤコビアン $J(\theta_n)$ で近似できる (ヤコビアンの
具体的な内容は 8.1 節で述べる).

$$\mathbf{r}_{n+1} - \mathbf{r}_n = J(\theta_n)(\theta_{n+1} - \theta_n)$$

$J(\theta_n)$ は一般に正方行列ではないので, 擬似逆行列
 $J^\#$ を用いて $\theta_{n+1} - \theta_n$ について解くと

$$\theta_{n+1} - \theta_n = J^\#(\theta_n)(\mathbf{r}_{n+1} - \mathbf{r}_n)$$

したがって, ステップ n における目標手先位置
 \mathbf{r}_n , 目標関節角度 θ_n が求まっていれば, ステップ
 $n+1$ における目標関節角 θ_{n+1} は

$$\begin{aligned} \theta_{n+1} &= J^\#(\theta_n)(\mathbf{r}_{n+1} - \mathbf{r}_n) + \theta_n \\ (\text{ただし } \mathbf{r}_{n+1} &= \text{traj}((n+1)\Delta t)) \end{aligned}$$

で得られる.

3.2 目標関節角速度の計算

1 ステップで関節角度を $\theta_{n+1} - \theta_n$ だけ変化させる
ための目標関節角速度 ω_{n+1} は,

$$\omega_{n+1} = (\theta_{n+1} - \theta_n)/\Delta t$$

である*1.

4 解の安定化

擬似逆行列解は特異姿勢付近になると極端に大
きな値が出てしまい安定しない. そこで, Tikhonov
正則化を行い解を安定化する.

J の特異値分解を $U\Sigma V^T$ とする. 以下に示す $J^\#$
を $J^\#$ の代わりに用いて解を得る.

$$J^{\#'} = V\Sigma U^T$$

$$\text{ただし } \sigma_{ii} = \frac{\Sigma_{ii}}{\Sigma_{ii}^2 + \epsilon^2}$$

*1 メモ: 積分手法が関係すると思われるので検証のこと.

ϵ は正則化パラメータで、大きいほど解を安定させる（その一方で正しい解が出なくなる）。

5 基準姿勢復帰

$J^\#(\theta_n)$ による解は、ステップ間の目標関節角度変化の二乗和 $(\theta_{n+1} - \theta_n)^T(\theta_{n+1} - \theta_n)$ を最小化する。このことは軌道によっては不自然な姿勢や絡まりの原因になる。

不自然な姿勢を避ける方法として、自然な基準姿勢を設定し、リンクアームの冗長自由度を用いて基準姿勢に近い解を得る方法が考えられる（冗長自由度を制御して指定した関節速度に極力近い解を得る方法は例えば文献 [?] で紹介されている）。

基準姿勢を θ_{normal} とし、目標関節角 θ_n を θ_{normal} に近づける関節角速度を $\omega_{pullback}(\theta_n, \theta_{normal})$ とする（ $\omega_{pullback}$ の具体的な計算は 8.2 節で述べる）。

線形方程式

$$\Delta \mathbf{r} = J \Delta \theta$$

$$\text{ただし } \Delta \theta = \theta_{n+1} - \theta_n, \Delta \mathbf{r} = \mathbf{r}_{n+1} - \mathbf{r}_n, J = J(\theta_n)$$

の、冗長自由度を含む解は

$$\Delta \theta = J^\# \Delta \mathbf{r} + (I - J^\# J) \mathbf{k}$$

である（両辺に左から J を掛けると右辺第二項は擬似逆行列の性質 $J = JJ^\#J$ により消えるので $J \Delta \theta$ に影響しないことがわかる）。

ここで、基準姿勢に向かう関節角速度 $\omega_{pullback}(\theta_n, \theta_{normal})$ が可能な限り実現されるとすると、 $\Delta \mathbf{r} = \mathbf{0}$ のとき $\Delta \theta = \omega_{pullback}(\theta_n, \theta_{normal})$ である。したがって、

$$\omega_{pullback}(\theta_n, \theta_{normal}) = (I - J^\# J) \mathbf{k}$$

この線形方程式を \mathbf{k} について解く。右辺が冗長自由度を表す事を考えると $(I - J^\# J)$ はフルランクではないため、擬似逆行列を用いて

$$\mathbf{k} = (I - J^\# J)^\# \omega_{pullback}(\theta_n, \theta_{normal})$$

である。

ところで、実際には以下に示すように $(I - J^\# J)^\# = I$ であり、したがって $\mathbf{k} = \omega_{pullback}(\theta_n, \theta_{normal})$ で

ある。

$$\begin{aligned} & (I - J^\# J)^\# \\ &= (I - J^\# J)^T ((I - J^\# J)(I - J^\# J)^T)^{-1} \\ &< I - J^\# J \text{ は (たぶん) 対称行列なので} > \\ &= (I - J^\# J)((I - J^\# J)(I - J^\# J))^{-1} \\ &= (I - J^\# J)((I - 2J^\# J + J^\# J J^\# J))^{-1} \\ &< J J^\# J = J \text{ より} > \\ &= (I - J^\# J)((I - J^\# J))^{-1} \dots^{*2} \\ &= I \end{aligned}$$

これらより、 $\Delta \theta$ を求める式は以下の通りとなる。

$$\Delta \theta = J^\# \Delta \mathbf{r} + (I - J^\# J) \omega_{pullback}(\theta_n, \theta_{normal})$$

6 関節の重み付け

加重擬似逆行列を用いると、関節に使用優先度を与えることができる。大きな重みを与えられた関節を極力動かさないように動作する。

重みを表す対角行列を W_{ii} とすると、加重擬似逆行列解は

$$\Delta \theta = W^{-1} J^T (J W^{-1} J^T)^{-1} \Delta \mathbf{r}$$

となる^{*3}。

一方、正則化等の都合から、実際には J の特異値分解を用いて $J^\#$ を求める方法をとりたい。そこで、 $W^{-1} = D^2$ なる対角行列 D を定義し、以下のように式変形する。

$$\begin{aligned} \Delta \theta &= W^{-1} J^T (J W^{-1} J^T)^{-1} \Delta \mathbf{r} \\ \iff \Delta \theta &= D^2 J^T (J D^2 J^T)^{-1} \Delta \mathbf{r} \\ \iff \Delta \theta &= D D^T J^T (J D D^T J^T)^{-1} \Delta \mathbf{r} \\ \iff \Delta \theta &= D (J D)^T (J D (J D)^T)^{-1} \Delta \mathbf{r} \\ \iff \Delta \theta &= D (J D)^\# \Delta \mathbf{r} \end{aligned}$$

これより、 W を重み行列とする加重擬似逆行列解は、

$$\begin{aligned} \Delta \theta &= D (J D)^\# \Delta \mathbf{r} \\ (\text{ただし } D_{ii} &= \frac{1}{\sqrt{W_{ii}}}) \end{aligned}$$

^{*3} 何故こうなるかは左から J を掛けてみればわかる。

で得られることがわかる。

実装上は、ヤコビアン J の代わりに重み付きヤコビアン JD を用い、 JD による擬似逆行列解が得られたら D を掛けた上で目標関節角度に加える。これらの処理は関節ごとに行えるので実装しやすい。

なお、さらに以下のように変形すれば加重擬似逆行列解が $J\Delta\theta = \Delta r$ を満たす解であることも確かめられる。

$$\begin{aligned}\Delta\theta &= D(JD)^{\#}\Delta r \\ \iff D^{-1}\Delta\theta &= (JD)^{\#}\Delta r \\ \iff JDD^{-1}\Delta\theta &= \Delta r \\ \iff J\Delta\theta &= \Delta r\end{aligned}$$

7 軌道追従制御

7.1 柔らかい軌道追従制御

いずれ具体的に記述する予定。さしあたっては文献 [?] を参照されたい。

7.2 LCP 計算の反復回数と追従誤差

大きな関節トルクを要する運動では、正確なオフセットトルクを求めるのにより多くの LCP 反復計算を要する。例えば関節ダンパを最大限に設定した 2 リンクアーム（アームあたり質量は 1[kg]）を重力に逆らって静止させるのに反復回数 $niter = 50$ 程度以上を要した（Springhead のデフォルトは $niter = 15$ ）。多数回の反復を要するのはオフセットトルクを計算するテストシミュレーションの間に限られるので、テストシミュレーション時は反復回数を大きくとり、本番シミュレーションでは通常の反復回数に戻して計算すればよい。

関節のバネ・ダンパは手先位置を軌道上に留めるよう作用するので、バネ・ダンパ係数が比較的大きければオフセットトルク計算に誤差があっても吸収される。この場合、反復回数とアームの柔軟性がトレードオフとなる。

8 剛体関節モデルでの実装

8.1 ヤコビアン

8.2 姿勢復帰速度

8.3 関節運動の実現

8.4 順運動学