

終端状態制御による 3 リンクロボットアームの制御

Control of Three Link Robot Arm by Final-State Control

○ 前田卓也*, 渡部慶二*, 村松鋭一*, 有我祐一*, 遠藤茂*

* (山形大学)

○ Takuya Maeda*, Keiji Watanabe*, Eiichi Muramatsu*, Yuichi Ariga*, Sigeru Endo*

* (Yamagata University)

キーワード: 終端状態制御(Final-state control), 誤差学習(Error Learning),

3 リンクロボットアーム(Three Link robot arm)

連絡先: 〒992-0061 米沢市城南 4-3-16 山形大学工学部応用生命システム工学 渡部・村松・有我研究室

前田卓也, tell : 080-1854-1591, E-mail : takujyun0722@co.jp

1 緒言

従来から、ロボットの運動制御に関する研究が多くなされている。それらの多くでは、フィードバック制御が用いられているため、逆運動学による軌道の生成と、それへの追従制御が一般的に行われている⁽¹⁾。しかしながら、人間や動物の運動制御では必ずしもフィードバック制御のみで行われているのではなく、反復学習動作や反射的動作はむしろフィードフォワード制御で行われているのではないかという指摘もある。

その反復学習動作の一つとして、川人らの研究⁽²⁾のフィードバック誤差学習が挙げられる。このフィードバック誤差学習は、学習初期はフィードバック制御で行われるが、反復学習するとフィードフォワード制御のみで動作できるようになる。しかし、このフィードバック誤差学習では、逆運動学で求める軌道が必要になってくる。

また、反射的動作の一つとして、翁らの研究⁽³⁾では猫ひねりの動作を逆運動学の軌道生成なしにフィードフォワード制御のみで運動制御が行えることを立証している。

そこで、本研究では、逆運動学による軌道生成なしにロボットアームをフィードフォワード制御するための手法として、西村らが提案している誤差学習を用いた終端状態制御⁽⁴⁾⁽⁵⁾を採用し、反復学習動作の一つである、投てき動作の制御を行う。また、この制御方法を用いることによってロボットアームが生物的動作になるかを検証する。

人が物を投げる動作を獲得する過程では、時々刻々の関節角度、ひいては手先の軌道を逐一求めているわけではない。物を投げる目標位置を決め、その飛距離に対して腕の初期状態とリリースポイント(終端状態)、またリリースポイントでの手先の速度を調整することによって動作を獲得している。このことから、終端状態制御を用いる

ことが適当であると考えられる。

本研究では、ロボットアームは3リンクとし、水平面内を動くものとする。このため、ロボットアームは横投げの動作をすることになる。また、投げる物は鉄

球とし、手先にある電磁石で保持されており、リリースポイントで電磁石 OFF となり瞬時に離れる。ロボットアームの上から見たx軸に平行に投げるモデルを Fig.1(a)に、横から見たモデルを Fig.1(b)に示す。このとき、Fig.1(b)の B 点を目標位置とし、AB 間を飛距離と考え、その飛距離より水平投射の法則を用いてリリースポイントでの手先の速度を求める。また、その速度より逆ヤコビ行列の法則を用いて各関節の角速度を算出する。算出された角速度とリリースポイント $[P_x, P_y]$ から得られる各関節の角度を終端の状態量とし、誤差学習を用いた終端状態制御により、投てきするためのフィードフォワード制御入力を求める。

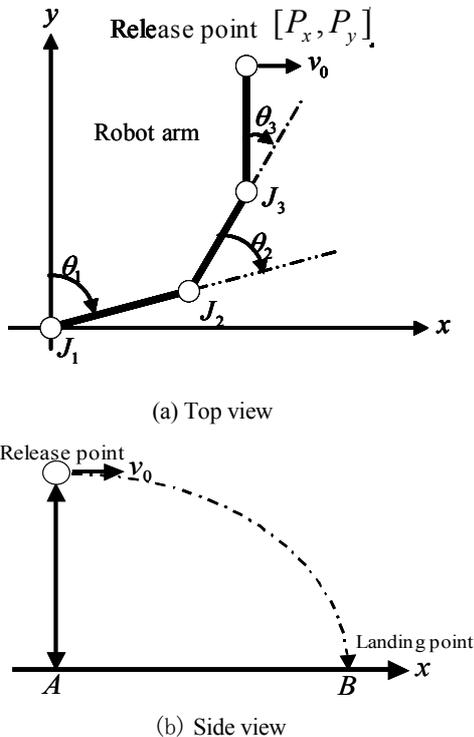


Fig.1 Sidethrow Model

本論文では、シミュレーションによりその有効性を検証する。

2 力学モデル

2.1 実システムと線形モデルの運動方程式

Fig.1 に 3 リンクロボットアームの力学モデルを示す。また、各パラメータは以下の通りである。

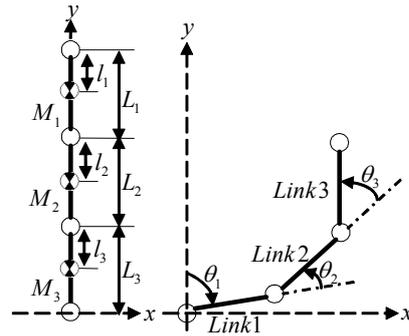


Fig.2 Dynamic model of three link robot arm

- $M_i[\text{kg}]$: 各リンクの質量 ($i = 1, 2, 3$)
- $L_i[\text{m}]$: 各リンクの長さ ($i = 1, 2, 3$)
- $l_i[\text{m}]$: 関節からリンクの重心距離 ($i = 1, 2, 3$)
- $u_i[\text{N} \cdot \text{m}]$: 各モータへの入力 ($i = 1, 2, 3$)
- $d_i[\text{kgm}^2/\text{s}]$: 各関節の減衰係数 ($i = 1, 2, 3$)

Fig. 1に示したモデルの運動方程式を、リンクの角変位 $\theta_1, \theta_2, \theta_3$ を一般化座標として、ラグランジュの運動方程式を解くことにより導出する。導出された運動方程式は次式で表される。

$$M(\theta)\ddot{\theta} + D\dot{\theta} + K(\theta, \dot{\theta}) = Fu \quad (1)$$

$M(\theta)$: 慣性行列、 D : 減衰係数

$K(\theta, \dot{\theta})$: 非線形項、 F : トルク係数

u : 制御入力、 $\theta = [\theta_1 \ \theta_2 \ \theta_3]^T$

この式(1)の非線形項 $K(\theta, \dot{\theta})$ はこのままでは状態方程式に組み込むことはできない。そこで、非線形項を三角関数ごとにまとめ、さらに三角関数

の加法定理を用いて $\sin \theta$ の項に整理して、以下の
ような式変形を施す。

$$p \times \sin \theta = p \times \frac{\sin \theta}{\theta} \times \theta \quad (2)$$

ただし、 p は多項式を表し、 $\theta = 0$ の時 $\sin \theta / \theta = 1$ と
する。これより以下の式を満たす行列 $K(\theta, \dot{\theta})$ が求
まる。

$$K(\theta, \dot{\theta}) = G(\theta, \dot{\theta}) \theta \quad (3)$$

これより、式(1)を以下とすることができる。

$$M(\theta) \ddot{\theta} + D \dot{\theta} + G(\theta, \dot{\theta}) \theta = Fu \quad (4)$$

また、式(4)を鉛直下向きの近傍で線形化した線
形モデルの運動方程式を以下のようにする。

$$M(\theta) \ddot{\theta} + D \dot{\theta} + E \theta = Fu \quad (5)$$

2.2 実システムと線形モデルの状態方程式

2.1 節の式(4)より実システムの状態方程式は

$$\dot{x} = A(\theta, \dot{\theta})x + B(\theta)u \quad (6)$$

ただし、

$$A(\theta, \dot{\theta}) = \begin{bmatrix} \mathbf{0}_{3 \times 3} & I_{3 \times 3} \\ -M(\theta)^{-1}G(\theta, \dot{\theta}) & -M(\theta)^{-1}D \end{bmatrix}$$

$$B(\theta) = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ M(\theta)^{-1}F \end{bmatrix} \quad x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$$

である。

また、2.1 節の式(5)より線形モデルの運動方
程式は以下のように表される。

$$\dot{x} = A_L x + B_L u \quad (7)$$

ただし、

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & I_{3 \times 3} \\ -M_L^{-1}E & -M_L^{-1}D \end{bmatrix} \quad B_L = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ M_L^{-1}F \end{bmatrix}$$

である。

3 終端状態制御

本研究では、参考文献(4)(5)の手法により、終
端状態制御を実現する。この概略を以下で述べる。

3.1 線形時変系に対する誤差学習による終端状 態制御

本研究では非線形である実システムを時変系
として扱い、誤差学習を取り入れた終端状態制御
によってフィードフォワード入力を求める。

制御対象の状態方程式(6)をオイラー法によ
って刻み時間 Δt で離散化した離散システムは以
下のように表される。

$$x_{k+1} = A_k x_k + B_k u_k \quad (8)$$

ただし、

$$A_k = A(\theta_k, \dot{\theta}_k) \Delta t + I \quad B_k = B(\theta_k) \Delta t$$

である。このシステムの拡大系は

$$\begin{aligned} \tilde{x}_{k+1} &= \tilde{A}_k \tilde{x}_k + \tilde{B}_k w_k \\ w_k &= -K \tilde{x}_k + \tilde{u}_k \end{aligned} \quad (9)$$

ただし、

$$\tilde{A}_k = \begin{bmatrix} A_k & B_k \\ \mathbf{0}_{3 \times 6} & I_{3 \times 3} \end{bmatrix}, \tilde{B}_k = \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ I_{3 \times 3} \end{bmatrix}, \tilde{x}_k = \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

と表され、さらに

$$F = \tilde{A}_k - \tilde{B}_k K \quad (10)$$

とすれば、式(8)は以下のように表される。

$$\tilde{x}_{k+1} = F \tilde{x}_k + \tilde{B} \tilde{u}_k \quad (11)$$

そして、鉛直上向きの近傍で線形化した線形時
不変の状態方程式を対象として求めた制御入力
 $\hat{v}_L = [\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{N-1}]^T$ を式(11)に順次与えた
ときの状態遷移行列 F_0, F_1, \dots, F_{N-1} は次式で求めら
れる。

$$\begin{aligned}\tilde{x}_1 &= F_0 \tilde{x}_0 + \tilde{B} \tilde{u}_0 \\ \tilde{x}_2 &= F_1 \tilde{x}_1 + \tilde{B} \tilde{u}_1 \\ &\vdots \\ \tilde{x}_N &= F_{N-1} \tilde{x}_{N-1} + \tilde{B} \tilde{u}_{N-1}\end{aligned}\quad (12)$$

\tilde{x}_N は以下のように表される。

$$\tilde{x}_N = F_{N-1} F_{N-2} \cdots F_0 \tilde{x}_0 + \tilde{U}_V \tilde{V} \quad (13)$$

$$\begin{aligned}\tilde{U}_V &= [F_{N-1} F_{N-2} \cdots F_1 \tilde{B}, F_{N-1} F_{N-2} \cdots F_2 \tilde{B}, \dots, F_{N-1} \tilde{B}, \tilde{B}] \\ \tilde{V} &= [\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{N-1}]^T\end{aligned}$$

目標状態を \tilde{x}^0 とすると式(13)の \tilde{x}_N が $\tilde{x}_N = \tilde{x}^0$ となる時終端状態制御が実現する。式(12)から得られる状態遷移行列 F_0, F_1, \dots, F_{N-1} と、式(13)によって実システムが初期状態 \tilde{x}_0 から、目標状態 \tilde{x}^0 に到達するための制御入力 \hat{v} は次式となる。

$$\hat{v} = \tilde{U}_V^{-T} (\tilde{U}_V \tilde{U}_V^{-T})^{-1} (\tilde{x}^0 - F_{N-1} F_{N-2} \cdots F_0 \tilde{x}_0) \quad (14)$$

となる。

以上で求めた制御入力 $\hat{v} = [\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{N-1}]^T$ を拡大システム(11)式に与えると拡大システムの状態ベクトル $\tilde{x}_k = [x_k \ u_k]^T$ が求められ、その要素から実システムへのフィードフォワード入力 u_k が得られる。しかし、式(14)で求めた \hat{v} を与えたシステム式(11)に与えた結果、終端状態は

$$\tilde{x}'_N = F'_{N-1} F'_{N-2} \cdots F'_0 \tilde{x}_0 + \tilde{U}'_V \hat{v} \quad (15)$$

となって目標状態 \tilde{x}^0 とは異なり、結局 \hat{v} は

$$\hat{v} = \tilde{U}'_V^{-T} (\tilde{U}'_V \tilde{U}'_V^{-T})^{-1} (\tilde{x}'_N - F'_{N-1} F'_{N-2} \cdots F'_0 \tilde{x}_0) \quad (16)$$

と表される。ここで、上付きの' は状態遷移の変化に伴うベクトルおよび行列の変化を表す。このときの終端誤差ベクトル e を

$$e = \tilde{x}^0 - \tilde{x}'_N \quad (17)$$

とし、入力 ΔV 以下のようにする。

$$\Delta V = \tilde{U}'_V^{-T} (\tilde{U}'_V \tilde{U}'_V^{-T})^{-1} e \quad (18)$$

そして、式(16)の \hat{v} に式(18)の ΔV を加えて

$$\begin{aligned}\hat{v} + \Delta V &= \\ \tilde{U}'_V^{-T} (\tilde{U}'_V \tilde{U}'_V^{-T})^{-1} (\tilde{x}^0 - F'_{N-1} F'_{N-2} \cdots F'_0 \tilde{x}_0)\end{aligned}\quad (19)$$

となる。式(19)はシステムが目標状態 \tilde{x}^0 に到達するための入力である式(16)と同じ形になる。式(19)を式(15)の \hat{v} に与えたとき、状態遷移が変わらなければ目標状態に到達する。ところが、入力の変化が大きいと状態遷移の変化が大きくなり ΔV による誤差補償は収束しない。そこで、学習係数 $\gamma (0 < \gamma \leq 1)$ を ΔV にかけて入力の変化を小さく抑える。そのため、以下のように入力の更新を行う。

$$\hat{v} \leftarrow \hat{v} + \gamma \Delta V \quad (20)$$

式(20)で求められた $\hat{v} = [\tilde{u}_0, \tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_{N-1}]^T$ を式(11)に与えると $\tilde{x}_k = [x_k \ u_k]^T$ が求められ、その要素から実システムへのフィードフォワード入力 u_k が得られる。この一連の流れをフローチャートとして、Fig. 3に示す。

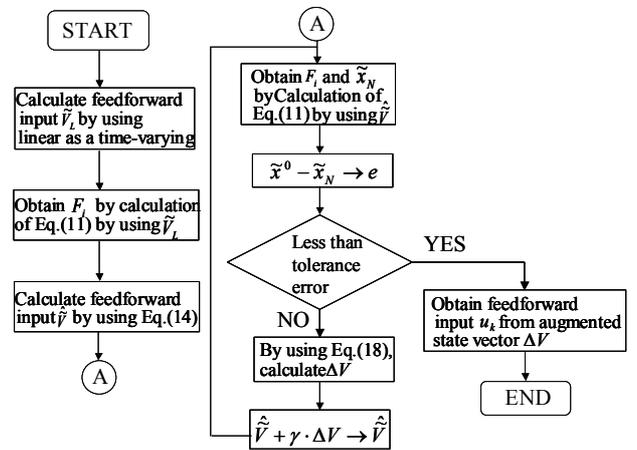


Fig.3 Flow chart of design of feedforward input

4 パラメータ導出方法

前章で述べた手法で制御入力を求めるためには、状態量 \tilde{x}_k の初期と終端での値が必要となる。初期状態は任意で構わないが、所望の投てきを実現するためには、終端状態をあらかじめ求めてお

かなければならない。ここでは、その手法について述べる。

4.1 終端状態での各関節角度の導出方法

Fig. 4 より逆運動学を用いて手先がリリースポイント達するための各関節角度を求める。

今回は、アームの先端にボールを取り付け投げられることを考え、リンク 3 を x 軸に平行になるようにした。この時の関節 J_3 の座標は次式となる。

$$\begin{aligned} P_{3x} &= P_x \\ P_{3y} &= P_y - l_3 \end{aligned} \quad (21)$$

よって、角度 d_3 は以下のように求まる。

$$d_4 = \tan^{-1} \left(\frac{P_{3x}}{P_{3y}} \right) \quad (22)$$

また、関節 J_1 と J_3 を結ぶ直線を l_{13} とすると、 d_1 と d_2 と d_4 は余弦定理を用いて以下のように求まる。

$$\begin{cases} d_1 = \cos^{-1} \left(\frac{l_1^2 + l_2^2 - l_{13}^2}{2l_1l_2} \right) \\ d_2 = \cos^{-1} \left(\frac{l_1^2 + l_{13}^2 - l_2^2}{2l_1l_{13}} \right) \\ d_3 = \cos^{-1} \left(\frac{l_2^2 + l_{13}^2 - l_1^2}{2l_2l_{13}} \right) \end{cases} \quad (23)$$

よって、各関節の角度は、以下のように求まる。

$$\begin{cases} \theta_1 = d_3 \pm d_1 \\ \pm(d_2 - \pi) \\ d_3 \pm d_4 \end{cases} \quad (24)$$

4.2 終端状態の各関節角速度の導出

ここでは、所望の投げきを実現するために必要な各関節角速度を求める。まず、Fig. 2 のように物理学の水平投射より速度 v_0 を (25) 式より求めることができる。

$$v_0 = AB / \sqrt{2H/g} \quad (25)$$

よって、手先の速度は $v = [v_0 \ 0]^T$ となる。

次に、手先の速度 v より各関節角速度を逆ヤコビ行列より求める。ロボットアームのベクトル方程式は以下のようなになる。

$$\begin{aligned} P_r &= \begin{bmatrix} \cos \theta_1 & \sin \theta_1 \\ -\sin \theta_1 & \cos \theta_1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} l_1 + \begin{bmatrix} \cos(\theta_1 + \theta_2) & \sin(\theta_1 + \theta_2) \\ -\sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} l_2 \\ &+ \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & \sin(\theta_1 + \theta_2 + \theta_3) \\ -\sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} l_3 \\ \therefore \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} &= \begin{bmatrix} \sin \theta_1 l_1 + \sin(\theta_1 + \theta_2) l_2 + \sin(\theta_1 + \theta_2 + \theta_3) l_3 \\ \cos \theta_1 l_1 + \cos(\theta_1 + \theta_2) l_2 + \cos(\theta_1 + \theta_2 + \theta_3) l_3 \end{bmatrix} \quad (26) \end{aligned}$$

となる。偏微分を行うことにより、微分関係は、

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos \theta_1 l_1 + \cos(\theta_1 + \theta_2) l_2 + \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin \theta_1 l_1 - \sin(\theta_1 + \theta_2) l_2 - \sin(\theta_1 + \theta_2 + \theta_3) l_3 \\ \cos(\theta_1 + \theta_2) l_2 + \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin(\theta_1 + \theta_2) l_2 - \sin(\theta_1 + \theta_2 + \theta_3) l_3 \\ \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin(\theta_1 + \theta_2 + \theta_3) l_3 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (27)$$

と求まる。よって、ヤコビ行列は以下のようなになる。

$$J = \begin{bmatrix} \cos \theta_1 l_1 + \cos(\theta_1 + \theta_2) l_2 + \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin \theta_1 l_1 - \sin(\theta_1 + \theta_2) l_2 - \sin(\theta_1 + \theta_2 + \theta_3) l_3 \\ \cos(\theta_1 + \theta_2) l_2 + \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin(\theta_1 + \theta_2) l_2 - \sin(\theta_1 + \theta_2 + \theta_3) l_3 \\ \cos(\theta_1 + \theta_2 + \theta_3) l_3 \\ -\sin(\theta_1 + \theta_2 + \theta_3) l_3 \end{bmatrix} \quad (28)$$

ここで、ヤコビ行列が横長行列のため逆ヤコビ行列を求めるために、擬似逆行列を用いて以下のようなになる。

$$J^+ = J^T (JJ^T)^{-1} \quad (29)$$

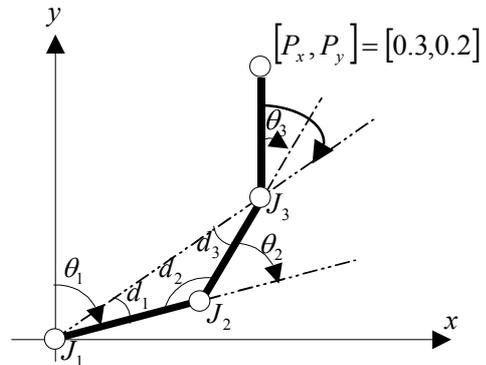


Fig.4 Final state

これより、各関節角速度は以下ようになる。

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J^+V \quad (30)$$

5 シミュレーション

シミュレーションに用いた力学モデルのパラメータを Table. 1 に示す。今回飛距離が 1.5m と 2.0m との二つのパターンのシミュレーションを行った。終端状態制御では初期状態と終端状態の状態量を任意に設定することができる。そこで、初期状態の状態量をすべて 0 とし、終端状態の状態量を 4 章の方法で求め Table. 2 の通りにした。

また、初期状態から終端状態までの時間を 1sec とし、3 章の方法で誤差学習をした。このときのオイラー法の刻み幅は 1.0ms とし、誤差の二乗和が 1×10^{-5} 以下になるまで学習を進めた。この目標の達成までには、飛距離 1.5m の時 36 回、飛距離 2.0m の時 46 回の学習を要した。

飛距離 1.5m のときのシミュレーション結果を Fig. 5 に、飛距離 2.0m のときのシミュレーション結果を Fig. 6 に示す。

Fig. 5 より、1sec で角度、角速度ともに目標状態にほぼ到達していることがわかる。また、Fig5(a)と(b)より、ロボットアームは初めはゆっくり動き、ある瞬間から急に目標位置に到達するまで速度をあげていることがわかる。そのため、Fig. 5(c)では後半で急激にフィードフォワード入力が大きくなっている。なぜこのような挙動になるかは、Fig. 5(d)を見るとよくわかる。ロボットアームは、初めゆっくりと振りかぶってタメを作り、そこから一気に目標位置まで腕を振っている。このように、人が振りかぶってから投げるよ

うな動作を得られた。

次に、より遠くへ投げようとしている Fig. 6 の結果を検証する。Fig. 5 のときよりも終端での角速度が大きく設定されているため、それを達成するためにフィードフォワード入力は先ほどより大きくなっていることがわかる。しかし、その変化量はさほど大きくならなかった。また、第二関節、第三関節が大きく変化し、先ほどよりも大きく振りかぶっていることが Fig. 5, Fig. 6 の (d) よりわかった。

このことから、ただ単にフィードフォワード制御入力だけを大きくして手先を加速するのではなく、大きく振りかぶることで手先が加速するための角度をかせぎ、その上で若干制御入力を増やしていることがわかった。

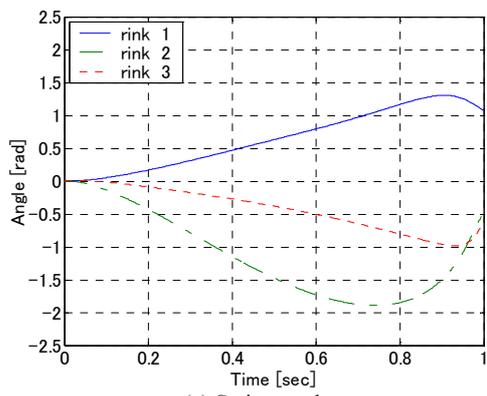
人も遠くに投げる場合に、投球動作自体も大きくしている。これにより過大な筋力を使って力ま

Table.1 Parameter of model

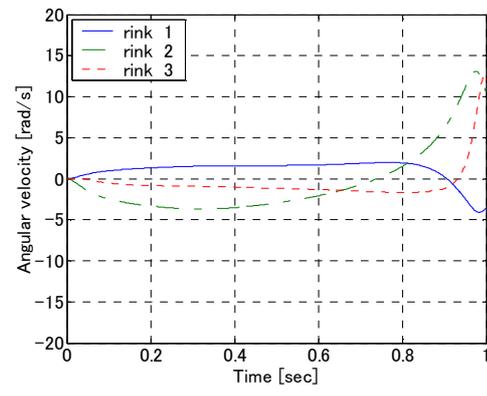
$M_1[kg]$	1.33	$l_1[m]$	0.114
$M_2[kg]$	0.872	$l_2[m]$	0.103
$M_3[kg]$	0.138	$l_3[m]$	0.079
$L_1[m]$	0.15	$d_1[kgm^2/s]$	0.247
$L_2[m]$	0.12	$d_2[kgm^2/s]$	0.075
$L_3[m]$	0.13	$d_3[kgm^2/s]$	0.046

Table.2 Values of final states

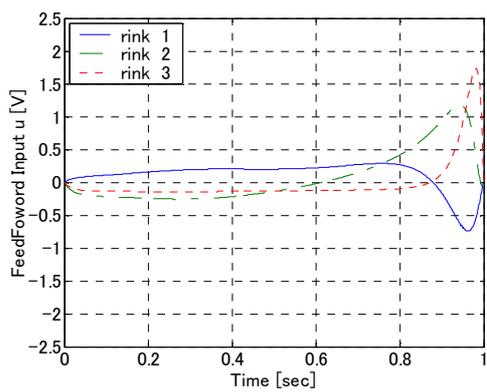
	Distance of a jump 1.5[m]	Distance of a jump 2.0[m]
θ_1	1.077	1.077
θ_2	-0.475	-0.475
θ_3	-0.601	-0.601
$\dot{\theta}_1$	-3.592	-4.789
$\dot{\theta}_2$	10.58	12.07
$\dot{\theta}_3$	14.11	16.09



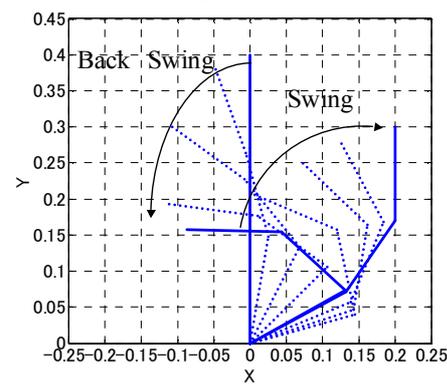
(a) Swing angles



(b) Angular velocities

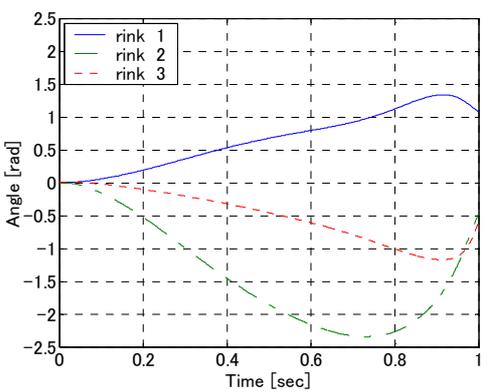


(c) Feedforward inputs

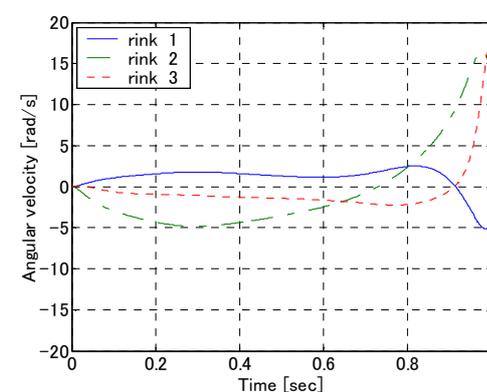


(d) Stick diagram

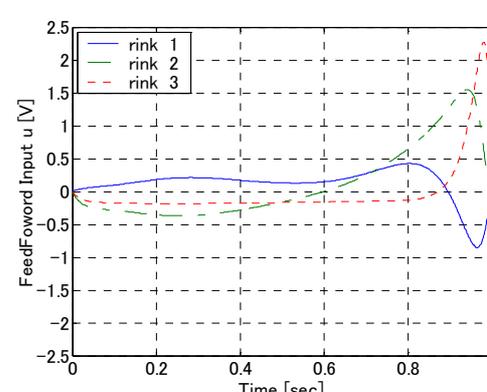
Fig.5 Simulation result of distance of a jump 1.5m



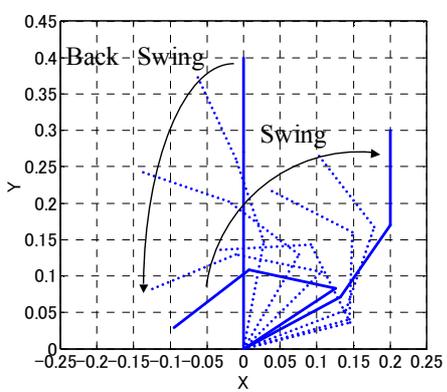
(a) Swing angles



(b) Angular velocities



(c) Feedforward inputs



(d) Stick diagram

Fig.6 Simulation result of distance of a jump 2.0m

かせに投げることを回避している。シミュレーション結果において、これと似た現象がみられたことから、終端状態制御を用いて獲得した動作は生物模倣的なものになっていると考えられる。

6 結言

本論文では、3リンクロボットアームによる投てき動作を誤差学習による終端状態制御を用いて導出し、シミュレーションによって検証した。その結果、以下の結論を得た。

誤差学習による終端状態制御によって導出したフィードフォワード制御入力を用いて、3リンクロボットアームの投てき動作の制御を行うことができた。

獲得した動作は、人が物を投げるときと同様に、一度振りかぶってタメを作ってから投げる動作に移行するものになった。

タメの大きさは飛距離により変化することが示された。

今後は、手先の終端速度を早くし過ぎるとアームがタメを作る際に可動範囲を越える場合がある

ので、角度に制限を与えることを検討する。さらに、実験による検証を行う予定である。

参考文献

- (1). 島田 明, 「モーションコントロール」, オーム社 p177-178
- (2). 川人 光男, 「脳の運動学習」, 日本ロボット学会誌, Vol.13, No.1, (1995), 11-19
- (3). 翁 志強, 西村 秀和, 「フィードフォワードのトルク入力による2リンク猫ひねりロボットの終端状態制御」 日本機械学会論文集(C編), Vol.66, No.643, C, (2007-3), 857-863
- (4). 西村 秀和, 高崎 堅治, 舟木 厚司, 戸谷 隆美, 「誤差学習による終端状態制御を用いたブラキエーションロボットの運動制御」, 日本機械学会論文集(C編), Vol.63, No.605, C, (1997-1), 182-189
- (5). 西村 秀和, 舟木 厚司, 戸田 隆美, 「フィードフォワード入力を用いた終端状態制御のパラメータ変動に対するロバスト性能」, (走行振り子の位置決め制御による検証), 日本機械学会論文集(C編) Vol.61, No.587, C, (1995-7), 2938-2944