

単脚ロボットの跳躍動作と姿勢制御

Posture control of One-leg Jumping Robot

○小野田憲司*, 渡部慶二**, 村松鋭一**, 有我祐一**, 遠藤茂**

○ONODA Kenji* WATANABE Keiji** MURAMATSU Eiichi** ARIGA Yuichi** ENDO Shigeru**

*山形大学大学院理工学研究科,**山形大学工学部

**Yamagata University

キーワード：跳躍ロボット(Jumping robot), 終端状態制御(Final-State Control),
フィードフォワード制御(Feed-forward Control),

連絡先：〒992-0037 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科 渡部研究室
小野田憲司 E-mail: onodatoar13@hotmail.com

1. はじめに

近年、「人と共存するロボット」を目指した非産業用ロボットに注目が集まっている。中でも、二足歩行ロボットの研究開発は特に盛んである。現在では転倒などにより物を壊したり、人に危害を加えないようにするため、安定な動作に重点がおかれている。しかし、今後ロボットの活動範囲を広げるためには障害物を飛び越えたり、段差を飛び降りるといった動作が必要になってくる。

そのため、本研究では二足歩行ロボットの活動範囲を広げるために跳躍する脚型のロボットを実現することを目的とする。問題の簡略化のために単脚型ロボットとし、終端状態制御を用いて跳躍軌道の生成を行い、着地時の姿勢制御をする。

2. 単脚型ロボット

2.1 単脚ロボットの概要

試作したロボットは関節に(計4自由度)ラジコン用サーボモータを使用している。このサーボモータはマイコンにより角度制御をおこなっている。モータを繋ぐフレームは、足裏は3mm、それ以外は1.5mmのA5052系アルミニウム板を使用した。

2.2 脚部のモデルと全体図

図2-1, 2-2, 2-3に脚部のモデル、概観、全体図を示す。

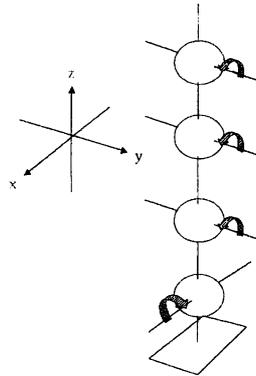


図 2-1 モデル

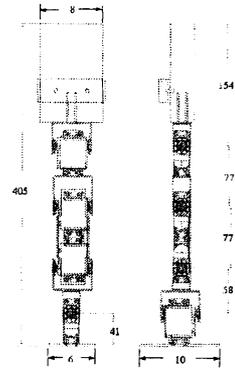


図 2-2 全体図

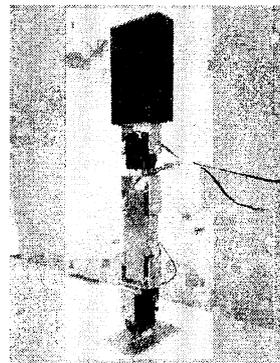


図 2-3 ロボットの概観

ロボットのモデルの状態方程式は式 (2.1) より

$$\dot{x} = A(\dot{\theta}, \theta)x + B(\theta)u \quad (4.2)$$

ただし、

$$G(\dot{\theta}, \theta) = N(\dot{\theta}, \theta) + g(\theta)$$

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

$$B(\theta) = \begin{bmatrix} 0_{1 \times 2} \\ M(\theta)^{-1}C \end{bmatrix}, \quad x = \begin{bmatrix} \dot{\theta} \\ \theta \end{bmatrix}$$

である。また、線形化した線形化モデルの状態方程式は以下のように表される。

$$\dot{x} = A_L x + B_L u$$

(4.3)

ただし、

$$A_L = \begin{bmatrix} -M^{-1}D & -M^{-1}G + \\ I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix},$$

$$B_L = \begin{bmatrix} 0_{1 \times 2} \\ M^{-1}C \end{bmatrix},$$

A_L, B_L は可制御行列である。

これを用いて終端状態制御を行った。

4.2 シミュレーション

まず、重力項のない状態で、線形化の値を変えたシミュレーションを2パターン行った。

次に重力項がある状態で線形化の値を変えたシミュレーションを2パターン行った。

シミュレーション1, 3は線形化値を

$$(\theta_1, \theta_2, \theta_3) = (0, 0, 0) \text{ とし、シミュレーション2, 4}$$

は $(\theta_1, \theta_2, \theta_3) = (60, -120, 60)$ とした。その他の条件はシミュレーション1, 2, 3, 4ともに、

$$\text{初期状態 } (\dot{\theta}_{1,2,3}, \theta_1, \theta_2, \theta_3) = (0, 60, -120, 60)$$

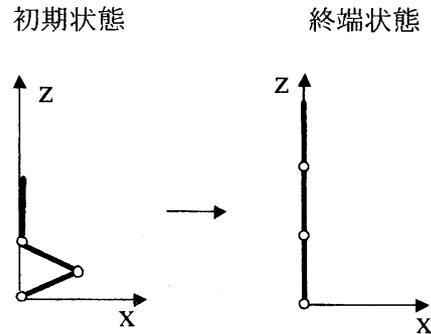
$$\text{終端状態 } (\dot{\theta}_{1,2,3}, \theta_1, \theta_2, \theta_3) = (0, 0, 0, 0)$$

$$\text{目標時間 } N\Delta t = 1 [\text{sec}]$$

$$\text{サンプリングタイム } \Delta t = 0.001 [\text{sec}]$$

とした。初期状態と、終端状態の各リンクの状態

が以下のような形になるように決めた。



シミュレーション1の結果は図4-1が制御入力の結果で図4-2が角度変化の結果である。目標時間の1秒後に目標値に収束している結果となった。

シミュレーション2の結果は図4-3が制御入力結果で図4-4が角度変化の結果である。制御入力の立ち上がり付近でシミュレーション1とは違いが出たが目標時間で目標値に収束していて、ほぼ同様の結果になった。

シミュレーション3の結果は図4-5が制御入力の結果で図4-6が角度変化の結果である。重力に対する適切な制御入力が必要で、目標の終端状態に到達しなかった。リンクの動きとしては重力方向に倒れていくような形となった。

シミュレーション4の結果は図4-7が制御入力結果で、図4-8が角度変化の結果である。目標状態に収束せず、シミュレーション3と同様な結果となった。シミュレーション1, 2とは違い、線形化の値によって制御入力の違いが大きく表れていて、その結果が角度変化にも出ている結果となった。

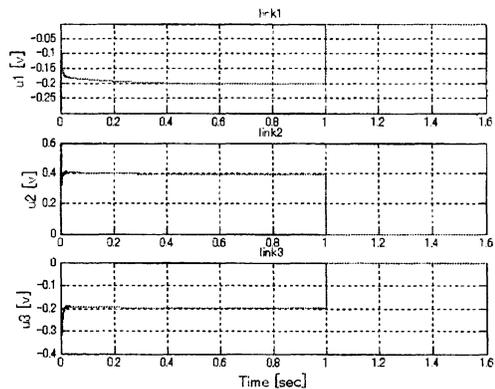


図 4-1 シミュレーション 1 結果

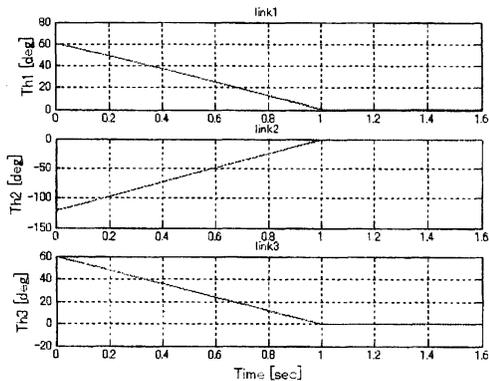


図 4-2 シミュレーション 1 結果

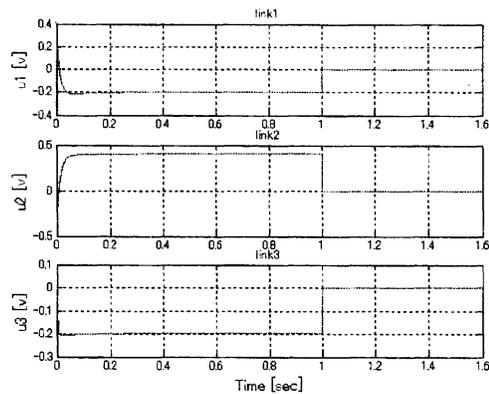


図 4-3 シミュレーション 2 結果

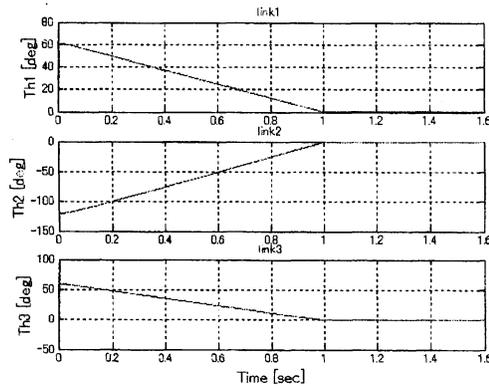


図 4-4 シミュレーション 2 結果

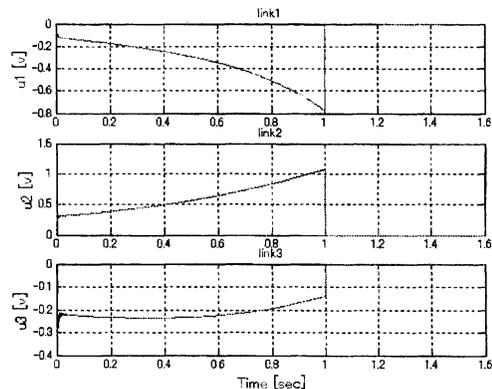


図 4-5 シミュレーション 3 結果

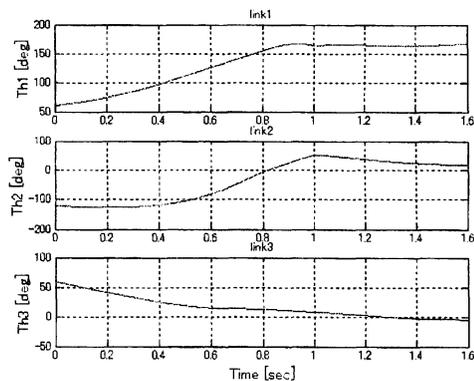


図 4-6 シミュレーション 3 結果

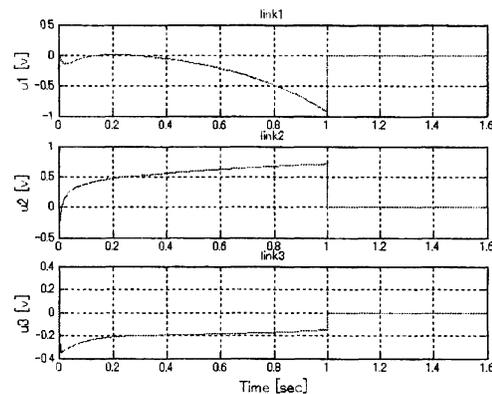


図 4-7 シミュレーション 4 結果

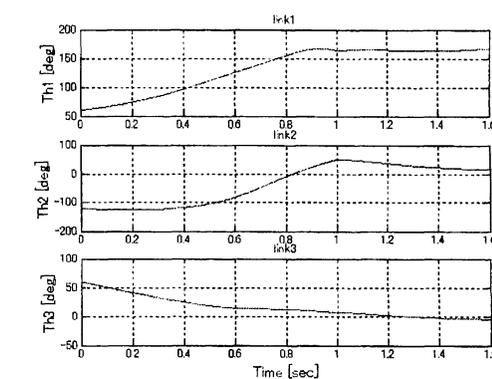


図 4-8 シミュレーション 4 結果

5. おわりに

5.1 まとめ

重力の影響を受けない状態でシミュレーションを行った場合では設定した終端状態に目標時間で収束し、この制御法にて任意の終端状態の制御入力を得られることが確認できた。

重力項を入れてシミュレーションを行った場合では、重力を考慮した適切な制御入力が求まらず、終端状態に収束しなかった。これは制御法が線形時不変系であるため適切な入力が求まらないものと考えられる。

5.2 今後の課題

重力項に対する制御入力が適切に考慮されるように時変系にする必要があるので、終端状態制御の方法を線形時変系に変えてシミュレーションを行っていく。また、今回のシミュレーションでは終端状態での速度を 0 としていたので速度の値を変化させた状態でのシミュレーションも行っていきたい。

終端状態制御法での最終的な目標は終端状態に重心位置の上向きの加速度を条件として持たせた状態での終端状態制御を行うことなので、その目標に持っていくための段階的な検証を今後行っていく。

参考文献

- 1) 西村 秀和, 高崎 賢治, 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
- 2) 計測自動制御学会(編) ロボット制御の実際, コロナ社 (1997)