

- 卒業論文 -

力学シミュレータを用いた歩行アルゴリズム  
の安定性に関する研究

平成18年2月13日

指導教官: 堀 洋一 教授

東京大学工学部電子工学科

30418 金間 泰樹

## 内容梗概

現在、ロボットの歩行アルゴリズムの主流は、足首トルク主体の歩行安定化手法であり、ZMP(Zero Moment Point) 制御法となっている。しかし、人の歩行を補助する器具を考えた場合、ZMP 制御法では足首に重いアクチュエータを装備する設計になるため、歩行自体が不安定になりやすくなってしまふ。そこで本論文では、歩行支援装具に用いることを前提として、歩行モデルに倒立振子近似モデルを用い、その歩行アルゴリズムの安定性を検討することを目的とした。

手法としては、矢状面（進行方向を含む面）での倒立振子近似モデルを前額面（進行方向に垂直な面）にも適用し、逐次的に歩幅を推定する制御を用いる。シミュレータとして高度な力学ロボットシミュレータ ROCOS を用い、歩行アルゴリズムを適用することで、その歩行アルゴリズムの有用性を示した。

# 目次

<b>第1章</b>	<b>序論</b>	<b>3</b>
1.1	研究の背景	3
1.2	研究の目的	3
<b>第2章</b>	<b>歩行原理についての考察</b>	<b>4</b>
2.1	ZMP法による歩行アルゴリズム	4
2.2	倒立振子近似モデルによる歩幅推定法をもちいた歩行アルゴリズム	5
2.2.1	歩行モデルの数式化	5
2.2.2	歩行予測システムの導出	7
<b>第3章</b>	<b>ROCOS(Robot Control Simulater)の紹介</b>	<b>10</b>
<b>第4章</b>	<b>歩行アルゴリズムの適用</b>	<b>13</b>
4.1	ロボットのモデルから倒立振子近似モデルへの変換	13
4.2	倒立振子近似モデルからロボットモデルへの変換	14
<b>第5章</b>	<b>歩行アルゴリズムの安定性の検証</b>	<b>17</b>
<b>第6章</b>	<b>結論</b>	<b>21</b>
6.1	本論文のまとめ	21
6.2	今後の展望	21

# 第1章 序論

## 1.1 研究の背景

現在、日本における肢体不自由者の数は、およそ250万人にまで及んでいる。程度は様々であるが、その原因としては疾病が6割を占めている。事故よりも2倍程度多く、中高年にそのウェイトが集中している。このような状況において、義肢の厚生省による交付は、毎年およそ1万件である。同様に下肢装具では、年間2万件の交付が報告されている。将来、高齢者の増加により肢体不自由者はさらに加速して増えていくことが予想される。

今後、超高齢者社会への移行に伴っておきる様々な問題を解決するために、医療福祉への工学的技術の導入が期待されている。近年、歩行に関する技術は、発展が目覚しく、ロボットの斬新な機構、歩行理論の研究が多く報告されている。そして、医療の分野でも歩行技術が必要とされている。それは義足を例に挙げられる。70年代ごろから、メカトロ化が多く、義足使用者やリハビリテーションスタッフから切望されるようになった。このようなことから、医療分野に工学的支援が必要であるという意識が確立し始めたのである。

わが国における社会がはらむ問題には、人口減少・少子高齢化・歩行困難者増加が顕在化しており、行政・各団体の働きにより福祉制度や活動の強化が図られている。これらの問題から社会全体の活力減退が起こることは必死である。制度や福祉の充実を図ることも急務だが、我々工学者にはさらに工学的支援という責務を果たし、各個人の活力を奮起することで、社会・経済のボトルアップを実現する時代となっている。

## 1.2 研究の目的

本研究においては、歩行補助装具に応用するための歩行アルゴリズムとして、倒立振り子近似モデルによる逐次歩幅推定法に注目し、3次元でのモデル化が複雑で困難なため、既存のロボットシミュレータ ROCOS を用いて、歩行アルゴリズムの安定性について検証する。

## 第2章 歩行原理についての考察

歩行補助装具を研究するに当たり、人の歩行とはいかなるものであるのかを知る必要がある。一言で歩行といっても、それが表す意味は非常に大きく、細かく挙げればきりが無いほど多くの歩き方が存在している。しかし、全ての歩行に共通して言えることは、着床地点がいったん決まってしまうたら後はその場で、次の歩みだしまで着床した足先はじっと留まっているということである。歩行の違いは、強いて言えば着地までの経過の違いといえる。早く移動するための足運びや何時でもとまれるための足運びなどは遊脚中の軌道の差であると思われる。歩行の根本的な原理は、「倒れずに移動を続けること」であり、非常に単純な命題であるにもかかわらず、さまざまな軌道により実現される歩容というオブラートで包み込まれてしまっているため、通常の歩行を検証するとしても人の歩行の根本原理を見つけることは難しいのである。

### 2.1 ZMP 法による歩行アルゴリズム

ZMP とは、足裏全体に分布してかかっている床反力の法線成分を、ある一点にかかっているとして置き換えたときの作用点のことである。つまり、力が分布していると扱いが面倒なので、1つの力に置き換えて考えるのである。多関節系のヒューマノイドにおいて使用されるケースが多く、各質点（ $n$  個）それぞれの状態の総和による導出が一般的である。その表現式は次式のとおりである。

$$X_{ZMP} = \frac{\sum_{i=1}^n m_i [x_i (\ddot{y} + g) - y_i \ddot{x}_i]}{\sum_{i=1}^n m_i (\ddot{y} + g)} \quad (2.1)$$

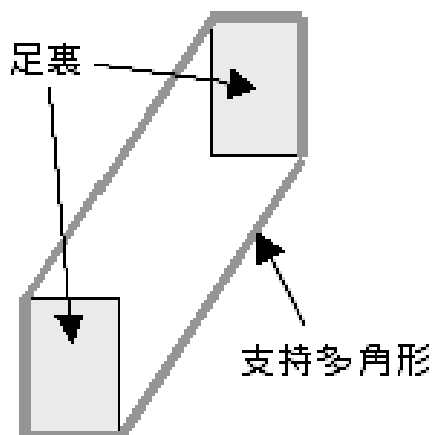


図 2.1: ZMP

床の上に立っているロボットは常に支持多角形内のどこかにある ZMP で床反力を受けている。片足で立っている二足歩行ロボットの ZMP は軸足裏のどこかにあり、足首の角

度を微妙に変えることによって、ZMP の位置を足裏の範囲内で自由に変えることができる。両足で立っているときの ZMP は、どちらかの足の裏か両足裏の間にあり、胴体の位置や膝の曲げ具合を微妙に変えることで両足裏間の範囲内で自由に変えることができる。

片足で立っているロボットは足首の角度を微妙に変えることによって、ZMP の位置を足裏の範囲内で自由に変えることができるため、目標 ZMP が足裏の範囲内にあれば実際の ZMP を目標 ZMP に一致させられる。しかし、目標 ZMP が足裏の外にあると、そんな ZMP は実現不可能なので、目標加速度を変えて摩擦力  $F_x$  を調整しなければならない。目標 ZMP が支持多角形の中にある動作は実現不可能である。

ロボットを「こう動かしたい」という軌道を決めたら、その動作の目標 ZMP を計算してみて、もし支持多角形の範囲をでていた瞬間があったら、そんな動作は実現不可能ということになる。そのまま実行させると足裏が床から離れてしまい、たいていの場合には転倒することになる。

## 2.2 倒立振り子近似モデルによる歩幅推定法をもちいた歩行アルゴリズム

本研究では歩行安定化システムとして、歩幅を状況にあわせて変化させるテクニックを用いる。歩幅を可変とした歩行アルゴリズムは 1974 年にオハイオ州立大学教授であった Robert B. McGhee 氏により発表された。しかしながら、現在の歩行アルゴリズムの主流派、足首トルク主体の歩行安定化手法であり、ZMP (Zero Moment Point) 制御法となっている。歩幅は基本的には固定されたパラメータとして扱われている。本研究では歩行補助装具についての応用を考えているため、ZMP 制御法などのように、各関節を高精度に制御するような手法では、装具と人間との間の剛性が弱いことなどから効果的な歩容補正が行えないので都合が悪い。また、ZMP 制御法で足首トルクを用いるためには、足首に重いアクチュエータを装備する設計になるため装具の重量バランスを大きく崩すこととなり、歩行事態が不安定になりやすくなってしまふ。そのため、足首アクチュエータは軽量でなければならず、あくまでも歩行中の補助的な役割程度しか望めない。一方、可変歩幅による歩行安定アルゴリズムであれば、最終的な着床までの姿勢が整っていれば歩行が安定化可能なため、装具と人間とのコンタクトに神経質になる必要が少ない。

### 2.2.1 歩行モデルの数式化

本研究においては歩行モデルとして倒立振り子近似モデルを採用する。簡単のため、モデルは、矢状面 (進行方向を含む面) に限定する。また、仮定として歩行モデルにおいて足の持つ質量は、無視する。これは人間の脚質量が全体重の約 10% であるということから、歩行に際して脚質量が歩行運動に影響を与えにくいと考えるためである。したがって、歩行中は上半身の運動に比べ、脚部の運動は十分小さいと仮定する。歩行モデルを図 2.2 に示す。

図 2.2 のモデルの運動方程式を求めると、ラグランジ法を用いることにより、式 (2.2)-(2.6) の操作で式 (2.7)(2.8)(2.9) が導かれる。

$$P_s(t) = (P_{sX}, P_{sY}) = (-r(t) \sin \phi_1(t) - l \sin \phi_2(t), l \cos \phi_2(t) + r(t) \cos \phi_1(t)) \quad (2.2)$$

$$K(t) = \frac{1}{2} J m \left( \frac{\partial \phi_2(t)}{\partial t} \right)^2 + \frac{1}{2} m \left( \frac{\partial P_s(t)}{\partial t} \right)^2 \quad (2.3)$$

$$P(t) = m g P_{sY}(t) \quad (2.4)$$

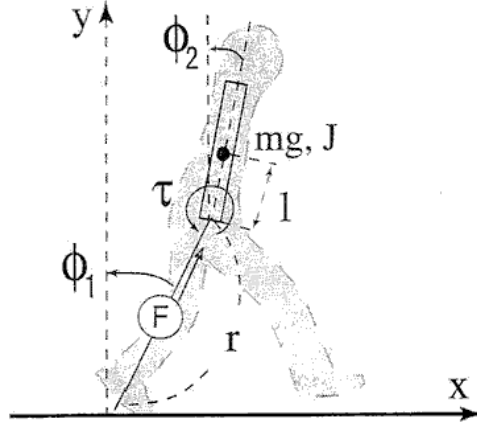


図 2.2: 歩行モデルの概要

$$L = K(t) - P(t) \quad (2.5)$$

$$\begin{bmatrix} \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{r}(t)} \right) - \frac{\partial L}{\partial r(t)} \\ \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\phi}_1(t)} \right) - \frac{\partial L}{\partial \phi_1(t)} \\ \frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\phi}_2(t)} \right) - \frac{\partial L}{\partial \phi_2(t)} \end{bmatrix} = \begin{bmatrix} F \\ -\tau \\ \tau \end{bmatrix} \quad (2.6)$$

式 (2.2) で重心の座標を求め、式 (2.3) で重心の運動量を、式 (2.4) で位置エネルギーを求める。式 (2.5) でラグランジアン  $L$  を定義し、式 (2.6) でラグランジ運動方程式を導いている。

$$\ddot{r} + l \sin(\phi_1 - \phi_2) \ddot{\phi}_2 + g \cos \phi_1 - r \dot{\phi}_1^2 - l \cos(\phi_1 - \phi_2) \dot{\phi}_2^2 = \frac{F}{m} \quad (2.7)$$

$$r^2 \ddot{\phi}_1 + l \cos(\phi_1 - \phi_2) r \ddot{\phi}_2 - g r \sin \phi_1 + r l \sin(\phi_1 - \phi_2) \dot{\phi}_2^2 + 2 r \dot{r} \dot{\phi}_1 = -\frac{\tau}{m} \quad (2.8)$$

$$l \sin(\phi_1 - \phi_2) \ddot{r} + r l \cos(\phi_1 - \phi_2) \ddot{\phi}_1 + (l^2 + J) \ddot{\phi}_2 + g l \sin \phi_2 = \frac{\tau}{m} \quad (2.9)$$

式 (2.7)(2.8)(2.9) をまとめて、より簡単に表し  $\mathbf{M}(\phi) \ddot{\phi} + \mathbf{G}(\phi, \dot{\phi}) = \mathbf{B}u$  とする。 $\mathbf{M}, \mathbf{G}, \mathbf{B}$  は、以下のように表される。

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & l \sin(\phi_1 - \phi_2) \\ 0 & r^2 & l r \cos(\phi_1 - \phi_2) \\ l \sin(\phi_1 - \phi_2) & l r \cos(\phi_1 - \phi_2) & l^2 + J \end{bmatrix} \quad (2.10)$$

$$\mathbf{G} = \begin{bmatrix} -r\dot{\phi}_1^2 - l \cos(\phi_1 - \phi_2)\dot{\phi}_2^2 + g \cos \phi_1 \\ lr \sin(\phi_1 - \phi_2)\dot{\phi}_2^2 - gr \sin \phi_1 + 2r\dot{r}\dot{\phi}_1 \\ -lr \sin(\phi_1 - \phi_2)\dot{\phi}_1^2 + 2l \cos(\phi_1 - \phi_2)\dot{r}\dot{\phi}_1 - gl \sin \phi_2 \end{bmatrix} \quad (2.11)$$

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \\ 0 & 1 \end{bmatrix} \quad (2.12)$$

ここで、 $\phi = [r, \phi_1, \phi_2]$ ,  $\mathbf{u} = [F/m, \tau/m]$  である。

## 2.2.2 歩行予測システムの導出

状態方程式  $\dot{\phi} = \mathbf{M}^{-1}(-\mathbf{G} + \mathbf{B}\mathbf{u})$  を非干渉化、線形化し、漸近安定となる極配置法を適用する。このシステムにおいて各状態変数は、 $(r - r_{ref}, \dot{r})$ ,  $(\phi_2, \dot{\phi}_2)$  のようにサブシステムとして扱うことが、非干渉化により可能となっている。 $(r - r_{ref}, \dot{r})$  においては、任意の極  $\lambda_1, \lambda_2$  を用いて、極配置法を適用する。 $u_1$  を式 (2.13) に示す。

$$u_1 = -\lambda_1 \lambda_2 (r_{ref} - r) + (\lambda_1 + \lambda_2) \dot{r} \quad (2.13)$$

また、 $(\phi_2, \dot{\phi}_2)$  においても同様で、任意の極  $\lambda_5, \lambda_6$  により、 $u_2$  は式 (2.14) で表せる。

$$u_2 = -\frac{Jr}{r+l}(\lambda_5 \lambda_6 + \frac{gl}{J})\phi_2 + \frac{Jr}{r+l}(\lambda_5 + \lambda_6)\dot{\phi}_2 \quad (2.14)$$

この状態フィードバックで上半身は安定化可能である。 $\phi_2 \uparrow 0$  であることから、 $\ddot{\phi}_1 - \frac{g}{r+l}\phi_1 = 0$  となり、 $\phi_1$  は不安定極を持つ。この倒れこみ角  $\phi_1$  の不安定極が人の行う動歩行を表す。

ヒトの歩行は簡略化に倒立振り子モデルと同等となることがわかっている。脚の倒れこみ角度の状態変数  $(\phi_1, \dot{\phi}_1)$  を以下のように表す。

$$\begin{bmatrix} \dot{\phi}_1 \\ \ddot{\phi}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{r+l} & 0 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \dot{\phi}_1 \end{bmatrix} \quad (2.15)$$

歩行を離散時間計の運動であると捉え、一步進むのに必要な時間  $T$  をサンプリング時間として、式 (2.15) を離散化すると、次式となり次歩の歩幅が推定される。

$$\phi_D[k+1] = \begin{bmatrix} \phi_{end}[k+1] \\ \dot{\phi}_{end}[k+1] \end{bmatrix} = \begin{bmatrix} \cosh bT & \frac{1}{b} \sinh bT \\ b \sinh bT & \cosh bT \end{bmatrix} \phi_D[k] - \begin{bmatrix} \cosh bT \\ b \sinh bT \end{bmatrix} u_D[k] \quad (2.16)$$

ここで、 $b^2 = \frac{g}{r+l}$ ,  $T$  はサンプル時間である。離散系も同様に、極配置法により安定化を行う。以下に極配置までの流れを示す。

$$\begin{bmatrix} \phi_1(T) \\ \dot{\phi}_1(T) \end{bmatrix} = \begin{bmatrix} \cosh bT & \frac{1}{b} \sinh bT \\ b \sinh bT & \cosh bT \end{bmatrix} \begin{bmatrix} \phi_1(0) \\ \dot{\phi}_1(0) \end{bmatrix} + D - \begin{bmatrix} \cosh bT \\ b \sinh bT \end{bmatrix} \begin{bmatrix} h_3 & h_4 \end{bmatrix} \begin{bmatrix} \phi_1(0) \\ \dot{\phi}_1(0) \end{bmatrix} \quad (2.17)$$



ここで、一步の開始時間 ( $t = 0$ ) から、次の一步までの時刻 ( $t = T$ ) に対し極配置を行う。そのため、目標値は  $(\phi_r, v_0) = (0, 0)$  とする。D は外乱要素であり、入力  $u_D$  には状態フィードバックのため、状態変数  $(\phi_1(0), \dot{\phi}_1(0))$  を用いている。上式をまとめ、

$$\begin{bmatrix} \phi_1(T) \\ \dot{\phi}_1(T) \end{bmatrix} \begin{bmatrix} (1 - h_3) \cosh bT & -h_4 \sinh bT + \frac{1}{b} \sinh bT \\ (1 - h_3)b \sinh bT & -h_4 \sinh bT + \cosh bT \end{bmatrix} \begin{bmatrix} \phi_1(0) \\ \dot{\phi}_1(0) \end{bmatrix} + D \quad (2.18)$$

を得る。これより、所望の極  $(\lambda_3, \lambda_4)$  を実現するために、

$$\begin{aligned} (z - \lambda_3)(z - \lambda_4) = & (z - (1 - h_3) \cosh bT)(z - \cosh bT + h_4 b \sinh bT) \\ & - (1 - h_3)b \sinh bT \left( \frac{1}{b} \sinh bT - h_4 \cosh bT \right) \end{aligned} \quad (2.19)$$

からフィードバック係数  $(h_3, h_4)$  が決定される。実際に  $u_D$  は、切り替える直前の  $\phi_1$  を  $\phi_{end}$  として、式 (2.20)(2.21) となる。

$$u_D = h_3 \left( \phi_{end}[k] - \frac{\phi_r}{2} \right) + h_4 \left( \dot{\phi}_{end}[k] - \frac{v_0}{r} \right) + \phi_r \quad (2.20)$$

$$\begin{cases} h_3 = 1 - \lambda_3 \lambda_4 \\ h_4 = [(1 + \lambda_3 \lambda_4) \cosh bT - \lambda_3 - \lambda_4] / (b \sinh bT) \end{cases} \quad (2.21)$$

ここで、 $\lambda_3, \lambda_4$  は離散系における任意の極である。 $\phi_r, v_0$  はそれぞれ、目標歩幅、目標前進速度である。図 2.3 を参考に、切り替え直後の各状態を推定すると、式 (2.22)(2.23)(2.24) が導かれる。

$$\phi_{st} = \phi_{end} - u_D \quad (2.22)$$

$$r_{st} = r_{end} \cos \phi_{end} / \cos \phi_{st} \quad (2.23)$$

$$\dot{\phi}_{st} = \frac{\dot{r}_{end}}{r_{st}} \sin u_D + \frac{r_{end}}{r_{st}} \dot{\phi}_{end} \cos u_D \quad (2.24)$$

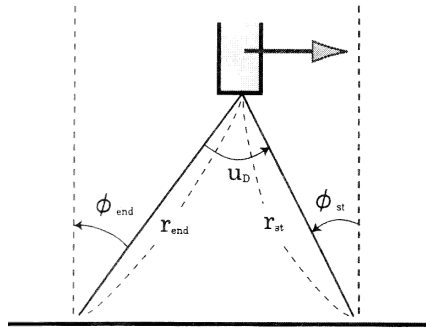


図 2.3: 支持脚切り替え時の歩行モデル

さらに逐次歩幅推定を行うことによって、遊脚期中に最終的な着床歩幅に向けての軌道生成が可能となる。次の脚切り替えの直前の立脚姿勢を推定し、その推定値 ( $\hat{\phi}_1[k+1], \dot{\hat{\phi}}_1[k+1]$ ) を参考に歩行安定化のための着床歩幅を決定していく。まず、立脚中の姿勢が次の着床までにどこまで倒れこむかを予測しなければならない。そのため、式 (2.16) によるシステムを用いて時間  $T_{rest}$  秒後の姿勢を算出する。

歩行中のある時刻  $t$  における一步の終了時刻  $T_{end}$  までの残り時間  $T_{rest}(= T_{end} - t)$  と立脚角度である  $\phi_1$  とその速度  $\dot{\phi}_1$  を推定の初期値として、次式に示す推定により一步終了時刻の立脚姿勢が得られる。

$$\hat{\phi}_D[k+1] = \begin{bmatrix} \hat{\phi}_1(T_{end}) \\ \dot{\hat{\phi}}_1(T_{end}) \end{bmatrix} = \begin{bmatrix} \cosh bT_{rest} & \frac{1}{b} \sinh bT_{rest} \\ b \sinh bT_{rest} & \cosh bT_{rest} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \dot{\phi}_1 \end{bmatrix} \quad (2.25)$$

これにより切り替え歩幅  $u_D$  は次式に書き換えられ、たえず歩行補助装具の遊脚中に繰り返し計算される。

$$u_D = h_3\left(\hat{\phi}_1(T_{end}) - \frac{\phi_r}{2}\right) + h_4\left(\dot{\hat{\phi}}_1(T_{end}) - \frac{v_0}{r}\right) + \phi_r \quad (2.26)$$

# 第3章 ROCOS(Robot Control Simulater)の紹介

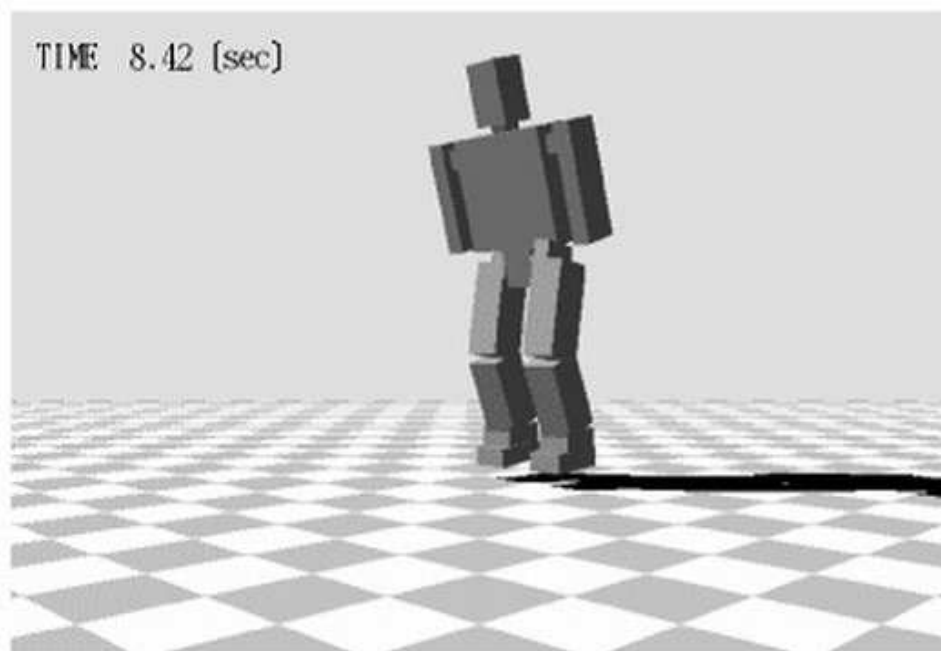


図 3.1: ROCOS

歩行ロボットの研究において、その制御アルゴリズムの検証は、まず簡略化されたシミュレーションによって行われ、その後、実機による実験によって行われることが多い。ROCOSは路面との衝突、摩擦を考慮した3次元シミュレータで横浜国立大学の藤本康孝助教授によって開発されてきた。ROCOSにおいては床との衝突・摩擦が物理的に厳密に計算されているため、床との相互作用に関してより現実に近いシミュレーションが可能となっている。3次元空間上を動作するロボットではロボットの足先などにかかる力・トルクなどが厳密に計算される。

また、質量・形状などのロボットのパラメータを実機と同様に設定できるので、現実に近いシミュレーションが可能であり、歩行アルゴリズムの検証に用いるのに適していると考えられる。

ROCOSでは図3.2のように15の部品から成り立ち、14の関節によってつながっている。また脚の関節における回転方向を図3.3に示す。

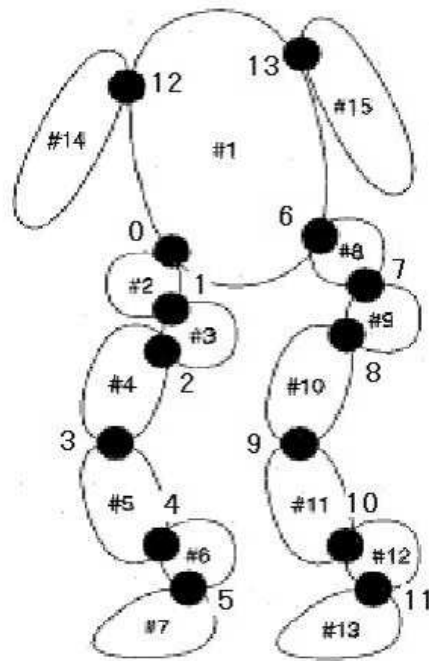


図 3.2: ROCOS の構成

ロボットの大きさは図 3.4 のようになっている。各部位の重さは設定できるようになっており、また、各関節の角度などの初期状態も自由に設定することができる。ロボットの姿勢制御は 14 の関節のトルクを入力として行われる。その際に使うことができるパラメータは、各関節の角度、各関節の角度の微分値、上体の重心の座標、上体の傾き、上体の速度、上体の回転速度である。

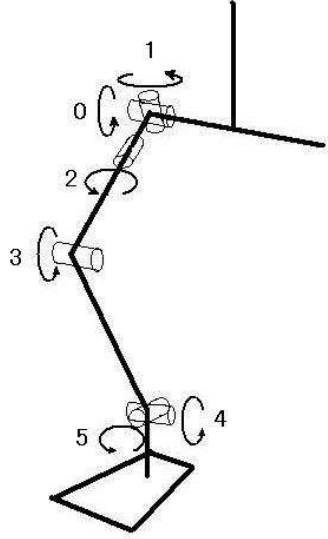


図 3.3: 脚関節の回転方向

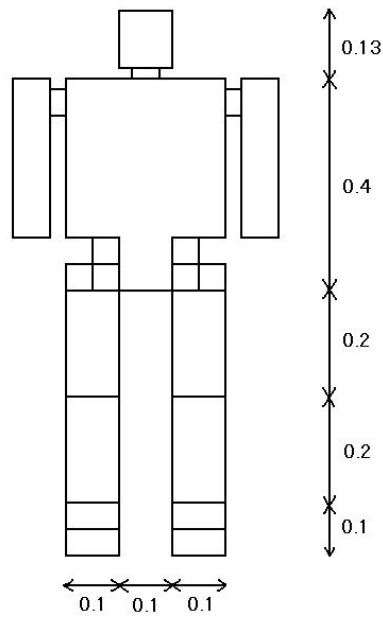


図 3.4: ロボットのサイズ

# 第4章 歩行アルゴリズムの適用

## 4.1 ロボットのモデルから倒立振り子近似モデルへの変換

倒立振り子近似モデルによる歩幅推定法をもちいた歩行アルゴリズムを実現するために、まずロボットを倒立振り子モデルに変換する。

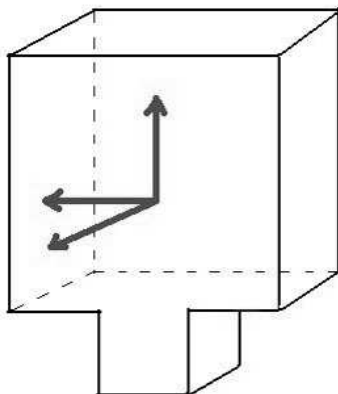


図 4.1:  $\mathbf{Q1}$

パラメータ $\mathbf{Q1}$ は図4.1に示すように上体# 1に固定した三つの単位ベクトルの行列でこれにより、上体の傾きを知ることができる。また $\mathbf{Q1}$ を、関節0と関節1の角度 ( $q[0], q[1]$ )を用いて回転させることで、# 4の部分の傾き $\mathbf{Q4}$ を導くことができる。

$$\mathbf{Q4} = \begin{bmatrix} \cos q[0] & 0 & -\sin q[0] \\ \sin q[0] \sin q[1] & \cos q[1] & \cos q[0] \sin q[1] \\ \sin q[0] \cos q[1] & -\sin q[1] & \cos q[0] \cos q[1] \end{bmatrix} \mathbf{Q1} \quad (4.1)$$

同様にして $\mathbf{Q4}$ と $q[3]$ から# 5の傾き $\mathbf{Q5}$ を導き出すことができ、またその他の全ての部分に対して同様に傾きを導くことができる。

倒立振り子近似モデルを採用した場合、立脚は上体を支えながらも重力の影響で滑らかに倒れていくことが望ましい。そのため立脚の脚の裏面は地面に接地し、もし右足が立脚であるならば、関節4、5のトルクをゼロにし、膝から上だけで倒立振り子を実現するのがよいと考えられる。また、ロボットの重心が丁度股の間にあると近似して、膝までの部分で振り子の足を計算すると、重心から見て、

$$\mathbf{R} = \begin{bmatrix} -Q1[1][0] * 0.1 + Q4[2][0] * 0.25 + Q5[2][0] * 0.2 \\ -Q1[1][1] * 0.1 + Q4[2][1] * 0.25 + Q5[2][1] * 0.2 \\ -Q1[1][2] * 0.1 + Q4[2][2] * 0.25 + Q5[2][2] * 0.2 \end{bmatrix} \quad (4.2)$$

となる。これにより矢状面、前額面から見た足の長さも求めることができる。

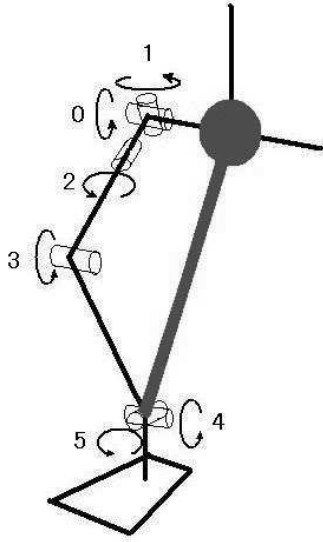


図 4.2: 倒立振子近似モデルの適用

左足についても同様の議論により左足 $L$ も計算でき、矢状面、前額面から見た脚の長さを求めることができる。次に $R, L$ から歩幅推定に必要な $\phi$ を求める。仮に立脚右足だとすると、矢状面では

$$\phi_y = \text{atan} \frac{R[1]}{R[2]} \quad (4.3)$$

前額面では

$$\phi_z = \text{atan} \frac{R[0]}{R[2]} \quad (4.4)$$

となる。

これから微分値を求め、式(2.25)により、脚切り替えの直前の立脚姿勢を推定し、さらに式(2.26)から、矢状面、前額面の切り替え歩幅を推定する。

## 4.2 倒立振子近似モデルからロボットモデルへの変換

倒立振子モデルにより、推定された歩幅は、倒立振子の脚の歩幅であるため、そのままではロボットの制御には使えない。ここでは推定値から、ロボットの各関節の目標値の導出を示す。

ここで右足が遊脚であると仮定する。まず、矢状面、前額面の歩幅より、式(2.22),(2.23),(2.24)から、切り替え時の倒立振子の脚の長さ $r_{sty}, r_{stz}$ が導かれる。 $Q1$ が与えられているので、図4.3の腿の付け根から膝の先までの $S$ が求まる。

$$S = \begin{bmatrix} r_{sty} \sin(u_{dy} - \hat{\phi}_y) + Q1[1][0] * 0.1 \\ -r_{stz} \sin(u_{dz} - \hat{\phi}_z) + Q1[1][1] * 0.1 \\ p1[2] + Q1[1][2] * 0.1 \end{bmatrix} \quad (4.5)$$

$S$  の長さが分かるので余弦定理によってまず、 $q[3]$  を求めることができる。 $q[3]$  が分かると、 $Q4$  を求めることができ、 $Q1$  と  $Q4$  との関係式

$$Q4 = \begin{bmatrix} \cos q[0] & 0 & -\sin q[0] \\ \sin q[0] \sin q[1] & \cos q[1] & \cos q[0] \sin q[1] \\ \sin q[0] \cos q[1] & -\sin q[1] & \cos q[0] \cos q[1] \end{bmatrix} Q1 \quad (4.6)$$

から  $q[0], q[1]$  が導かれる。

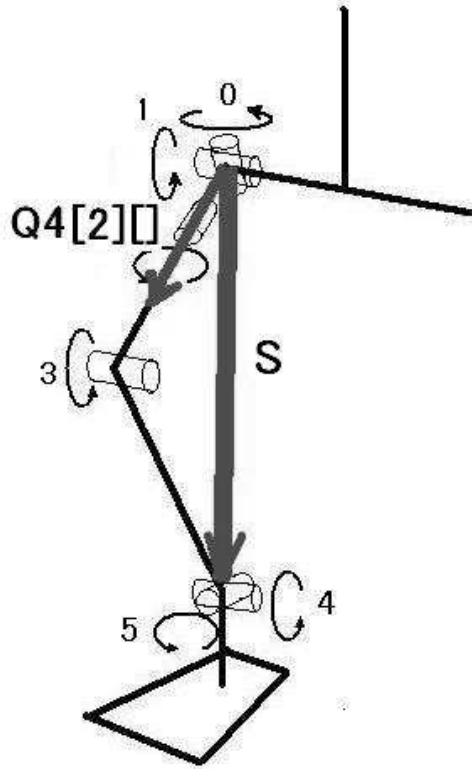


図 4.3: 関節角度の計算

求まった目標値に  $T_{rest} = 0$  で達するように軌道を作る。遊脚が予想周期よりも早く着地したり、着地しないことがないように気をつける。遊脚が正確に地面に着地するために遊脚の足の裏面が地面と平行になるように制御する。すなわち  $Q5$  に対して  $q[4], q[5]$  の角度で回転させた  $Q5$  が地面と平行になるような  $q[4], q[5]$  を求める。 $Q5, Q7$  の関係式より

$$\begin{bmatrix} \sin q[4] \cos q[5] & \sin q[5] & \cos q[4] \cos q[5] \end{bmatrix} Q5 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.7)$$

を解くことにより  $q[4], q[5]$  を得る。

より安定した推定値を得るために、また歩行が不安定にならないために上体を垂直に保つことは重要である。上体の制御は立脚によって行う。右足が立脚の場合、関節  $0, 1$  によって制御する。上体の回転速度を表す  $w1$  を用い、上体前後を直立に保つために  $y$  軸周



りの回転を制御するように

$$u[0] = -50 * w1[1] \quad (4.8)$$

とトルクを与え、上体左右を直立に保つために x 軸周りの回転を制御するように

$$u[1] = -50 * w1[0] \quad (4.9)$$

としている。

歩行は周期で右左の役割を交代させている。万が一周期の途中で遊脚がついたとすると、一定のトルクが発生した時点で脚を切り替える制御を行っている。歩行を制御するパラメータは、左右、前後の歩幅の目標値と極の値である。

## 第5章 歩行アルゴリズムの安定性の検証

シミュレーションを以下の図において示す。図 5.1 に歩幅（前後方向）の目標値を示す。歩幅の目標値は時間とともに増加させ、3 秒後に 0.5[rad] になるようにし、その後は一定である。シミュレーションは推定に用いる極を、0.01, 0.02, 0.1, 0.5 と変化させて行った。歩行の周期は一步当たり 0.4[sec]、横方向の歩幅の目標値は 0.1[rad] として、速度の目標値も 0 としている。

図 5.2, 5.3, 5.4, 5.5 において y 軸は右足と左足の関節角度を示しており、極が原点に近くなるにつれて、目標値までの収束が早くなっていく様子が見られる。極が 0.5 の時には、歩行は安定せず、4.5 秒付近で転んでしまった。極の指定には少なくとも安定条件から 1 より小さくしなければならないが、実用を考慮すると、極の配置は 0.02~0.01 に設定することで適度な収束特性と、外乱抑圧特性を有するということが分かった。

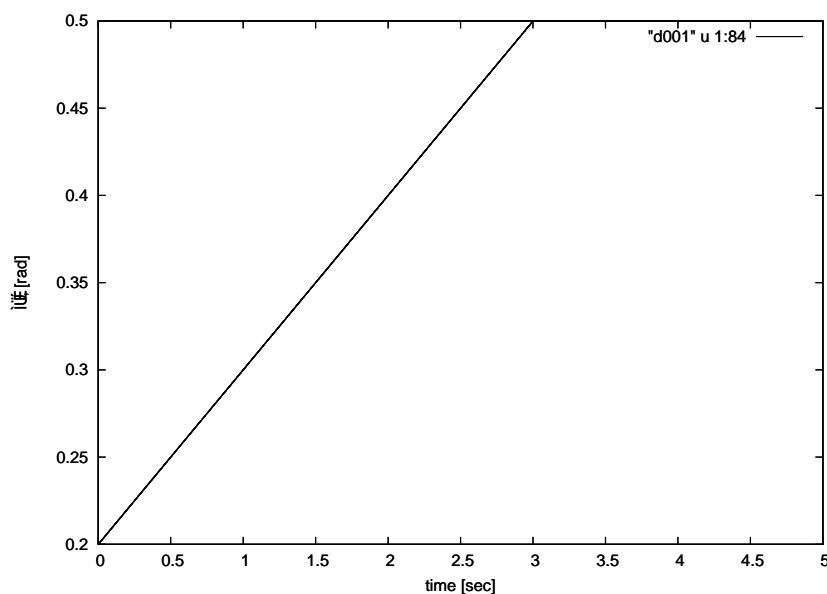


図 5.1: 歩幅の目標値

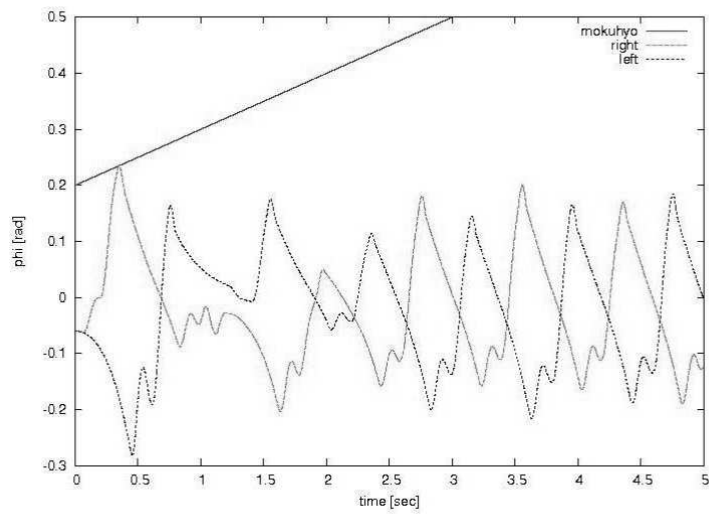


図 5.2: 極 0.01

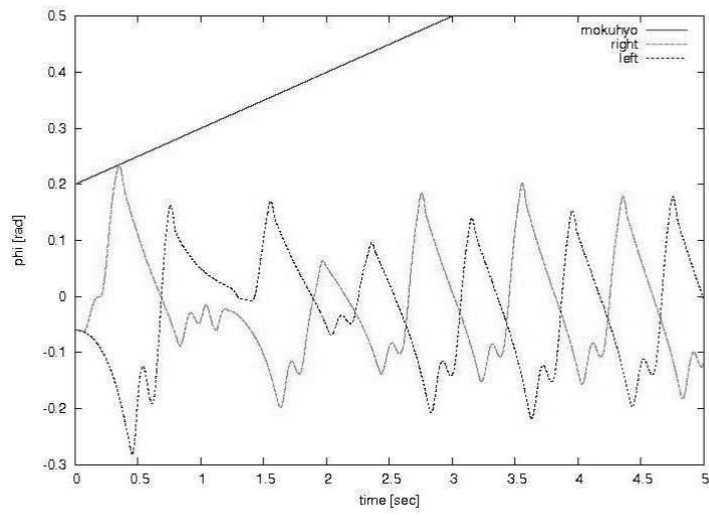


図 5.3: 極 0.02

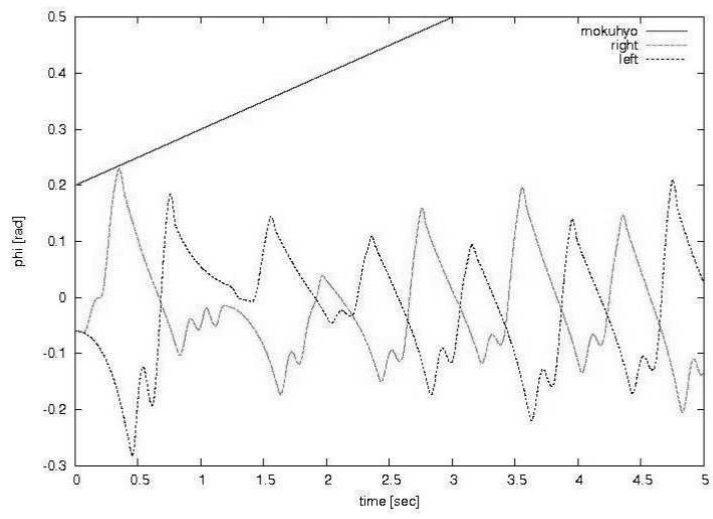


図 5.4: 極 0.10

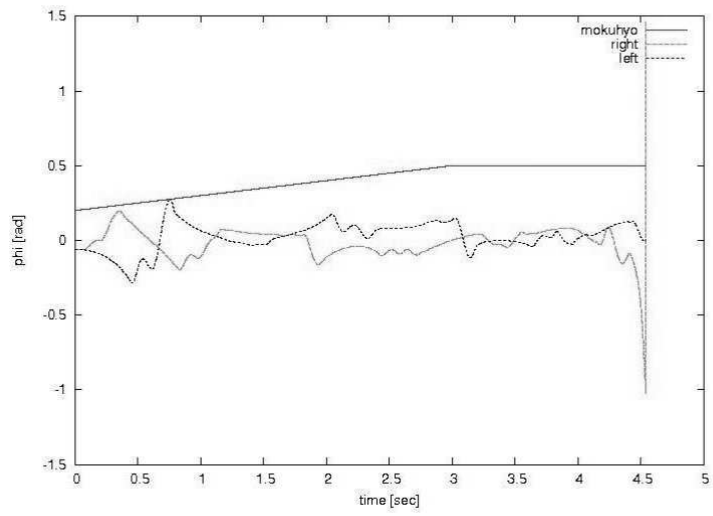
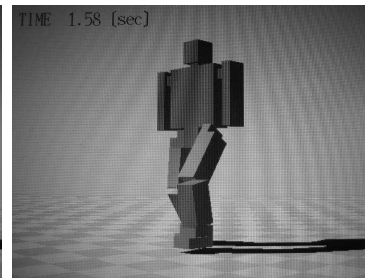
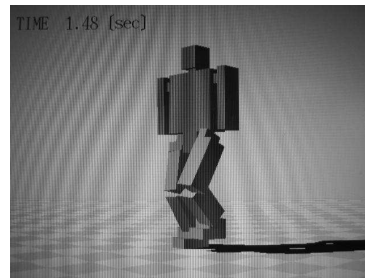
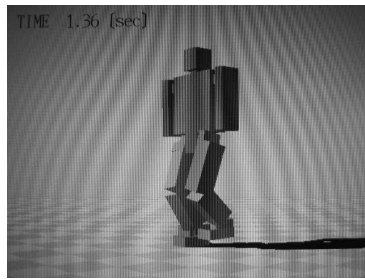
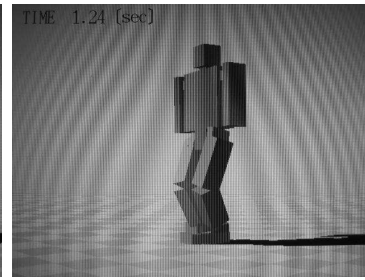
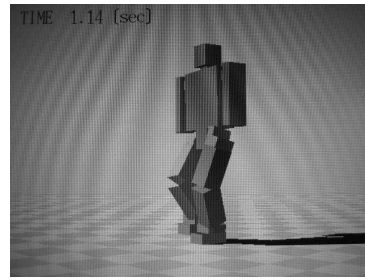
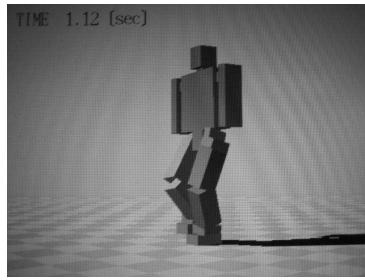
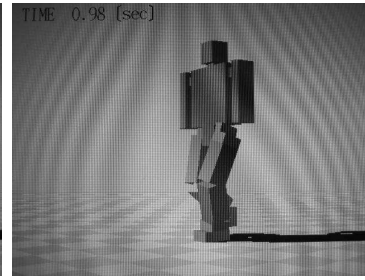
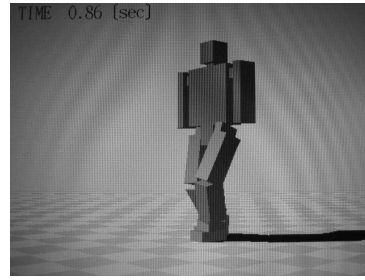
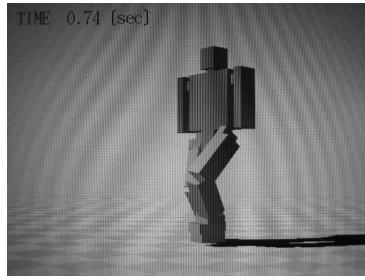


図 5.5: 極 0.50

以下にシミュレーション中のロボットの歩行を示す。  
歩行の周期は一歩あたり 0.4(sec) である。



# 第6章 結論

## 6.1 本論文のまとめ

本研究では歩行補助祉機器に用いられることを考えた、歩行安定化のアルゴリズムの有効性を ROCOS により検証した。倒立振子近似モデルを用いた逐次歩幅推定によるアルゴリズムのロボットへの適用について提案し、床反力を考慮した場合においても、正常に歩行を行えることが明らかにした。

## 6.2 今後の展望

今後は提案した歩行安定化アルゴリズムの適用範囲を理解し、より多彩な環境での安全な歩行を実現するための実機による検証実験を行うことが必要である。高精度で正確な歩幅推定を行うためには、着床検出や、重心の位置などが正確に求められなければならない。それらについての問題は今後の課題としたい。

# 関連図書

- [1] 畠 直輝,” 制御工学応用による移動用福祉機器の高機能化に関する研究”,2004
- [2] Yasutaka Fujimoto,Atsuo Kawamura,”Simulation of an Autonomous Biped Walking Robot Including Environmental Force Interaction”,IEEE Robotics & Automation Magazine,pp33-42,1998
- [3] 川崎 晴久,” ロボット工学の基礎”, 森北出版株式会社,1991
- [4] 伊藤 浩司, 藤井 久一, 伊藤 正美,”2 足歩行系のサーボモデルによる解析”, バイオメカニズム 4, pp. 263-270, 1978, 東京出版会
- [5] 伊藤 聡, 西垣 智啓, 川崎 晴久,” 床反力に基づいた一定外力場での規律姿勢に対する制御法”, 計測自動制御学会論文集, Vol.38, No.1, pp.79-86, January 2002
- [6] 加藤 久, 平田 崇,” 歩行補助装置の研究”, 日本機械学会, 福祉工学シンポジウム CD-ROM 論文集, W-418, 2001