

JAERI-M
91-047

人間型ロボットの二足歩行シミュレーション

1991年3月

久米 悦雄

JAERI-Mレポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の間合わせは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）
あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department
of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun,
Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1991

編集兼発行	日本原子力研究所
印刷	日立高速印刷株式会社

人間型ロボットの二足歩行シミュレーション

日本原子力研究所東海研究所情報システムセンター

久米 悦雄

(1991年2月20日受理)

日本原子力研究所では、原子力知能化システム技術の研究の一環として、1987年からHASP (Human Acts Simulation Program) と題する研究を進めている。この研究では、人間型ロボットが日本語で書かれた作業命令書を読み、意味を理解し、自己の行動を計画し、動作列を生成し、装置や機器が保有する情報を採用し、その動作列を精密化してプラント保守作業を遂行する。これらの過程は、すべて論理計算と数値計算によってシミュレーションされる。HASPプロジェクトの主な目的は、知能ロボット設計に必要な基盤技術を開発することである。

本報告書は、人間型ロボット運動のシミュレーション法とその結果を述べる。ヴコブラトビッチ (M. Vukobratović) による腕固定定常歩行モデルを基に、腕及びロボット発進・停止運動のシミュレーション・モデルを新しく開発した。シミュレーション結果として、腕動作が歩行安定に及ぼす影響、発進から定常歩行への移行等について議論する。

Numerical Simulation of Biped Locomotion Robot

Etsuo KUME

Computing and Information Systems Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received February 20, 1991)

The Human Acts Simulation Program (HASP) has been performed since 1987 at JAERI. In this program, a human-shaped robot reads and understands instructions written in natural languages such as Japanese, planning and producing a required sequence of actions, approach to a device or an instrument recognizing its entity and does the ordered work for plant maintenance. All of these processes are simulated by logical and numerical computations. The major aim of the HASP is to develop fundamental technologies for design of intelligent robots.

This report will describe the simulation methods and results of a human-shaped robot motions. The simulation methods of robot's arms and robot's start and stop motions have been newly developed based on the steady walking model with fixed arms developed by M. Vukobratović. The influence of arm motions to the stable walking and the continuous connection from start to steady walking, etc. will be discussed.

Keywords: HASP (Human Acts Simulation Program), Robot, Robot Dynamics, Human-shaped Robot, Biped Locomotion, Zero Moment Point (ZMP), Rigid Body, Dynamic Walking, Stable Region, Compensating Motion

目 次

1. はじめに	1
2. 二足歩行ロボットモデル	5
2.1 ロボットモデルの概略	5
2.2 ロボット運動方程式	6
2.3 安定領域	9
2.4 腕可動（能動型）モデルの計算法	10
2.5 腕可動（受動型）モデルの計算法	21
2.6 発進・停止歩行動作の計算法	23
3. シミュレーション結果	32
3.1 シミュレーションモデルの分類	32
3.2 腕固定型	32
3.3 腕可動型（受動型）	33
3.4 腕可動型（能動型）	36
3.5 発進・停止歩行動作シミュレーション	37
3.6 シミュレーション結果の映像化による考察	41
4. おわりに	69
4.1 まとめ	69
4.2 ハードウェア化について	71
謝 辞	73
参考文献	74
付録A 定常歩行シミュレーションプログラムの概略（腕可動（能動型）モデル）	77
付録B 発進・停止歩行シミュレーションプログラムの概略	80

Contents

1. Introduction	1
2. The simulation model of biped locomotion robot	5
2.1 Outline of the robot model	5
2.2 Equation of motion	6
2.3 Stable region	9
2.4 Simulation method of the biped locomotion with passive swing arms	10
2.5 Simulation method of the biped locomotion with active swing arms	21
2.6 Simulation method of starts and stops walking	23
3. Simulated results	32
3.1 Classification of the simulation model	32
3.2 Simulated results of the biped locomotion with fixed arms	32
3.3 Simulated results of the biped locomotion with passive swing arms	33
3.4 Simulated results of the biped locomotion with active swing arms	36
3.5 Simulated results of starts and stops walking	37
3.6 Visualization of simulated results	41
4. Concluding remarks	69
4.1 Summary	69
4.2 Realization of robot hardware	71
Acknowledgements	73
References	74
Appendix A Construction of computing program for steady walking	77
Appendix B Construction of computing program for starts and stops walking	80

1. はじめに

日本原子力研究所では、原子力知能化システム技術の研究の一環として、1987年から HASP (Human Acts Simulation Program) と題する人工知能とロボティクスに関する研究を進めている^{1)~3)}。この研究では人間型ロボットが日本語で書かれた作業命令を読み、意味を理解し、自己の行動を計画し、動作列を生成し、装置や機器が保有する情報を採用し、その動作列を精密化してプラント保守作業を遂行する。これらの過程は全て論理計算と数値計算によってシミュレーションされる。HASPプロジェクトの目的は第一に知能ロボット設計の下敷きとなる技術の開発、第二にプラントの知能化・自動化技術の開発、第三には人工知能関連のシステム化された基盤技術を原子力分野の研究者、技術者に提供することである。HASPはいくつかの個別要素技術からなり、本報告はこの中の1つを構成するロボット運動に関する研究報告である。

HASPにおいて対象とするロボットは以下のような理由から人間型ロボットが望ましいとしている。

- ①現在のプラントが多くの場合人間の存在を（人間による保守・点検）前提に作られている。
- ②脚による移動は従来の車輪やクローラ型と比べて地形の変化や障害物回避に対する適応性が高い。
- ③HASPにおいてターゲットとするロボットを人間型とすることで、知能ロボットのための知識ベースの作成が容易になる。
- ④最終的には作業を限定しない汎用ロボットを目指している。

本報告は人間型知能ロボット設計に資する基盤技術開発を目的とした、ロボットの動作シミュレーションに関するものである。ロボットに必要とされる機能は移動機能とマニピュレーションの2つに大別され、ここでは始めに移動機能に着目して二足歩行動作を取り上げる。二足歩行には、定常歩行、発進・停止歩行、方向転換、傾斜地の歩行等いろいろな移動パターンが要求される。ロボット設計上重要となるこれらの安定歩行パターンを数値シミュレーションによって検討することがここでの目的である。

二足歩行に関する研究は現在国内の十数機関と国外の数機関で行われ（Table 1.1 参照）、早稲田大学のWL-12RⅢ⁴⁾や岐阜大学のBLR-G2⁵⁾等が実際にハードウェアを作成し歩行を成功させており、現在のところ安定な動歩行の実現に関してはこの2つ

が最先端を行っている。早稲田大学のWL-12RⅢではZMP (Zero Moment Point) 規範を用い、ZMPが常に安定領域に存在するような歩行パターンを作成している。このロボットは上体を持ち、任意の歩容に対しこの上体の運動によって安定歩行となるように補償させることができる。上体によって補償動作を行わせることで平坦地における完全歩行のみならず、非平坦路や未知の外力下での動歩行も成功させている。但し、非常に大きなトルクを必要とするため、アクチュエータに油圧システムを用いておりロボットの重量が大きくなっている。早稲田大学の研究は、将来的にホームロボットの実現を目指している。また、岐阜大学のBLR-G2ではロボットシステム全体が持つ角運動量に着目して、支持脚足首トルクを用いシステムの角運動量を制御することによって、安定で人間的な自然な歩行を目指している。このロボットは重力場に逆らわない(重力を利用した)自然な歩行であり、早稲田大学での研究に較べてあまり大きなトルクを必要としないため、アクチュエータにモータを用いており比較的軽量のロボットが実現されている。しかし、遊脚が接地する際の衝撃力の影響が大きく、平坦地での歩行に限定しており、発進+定常歩行までが実現されている。

ここでは、早稲田大学で用いているZMP規範を採用して、ユーゴスラビアのヴコブラトビッチ (M. Vukobratović) の人間型モデル⁶⁾を基本に、まず直進歩行に関する数値シミュレーションを行った⁷⁾。このオリジナルモデルは人間のボディに関するデータを元にしたもので、下肢と胴体及び両腕からなる腕固定の人間型二足歩行ロボットである。このモデルに対するヴコブラトビッチによる歩行シミュレーションプログラムでは、定常直進歩行についてのみシミュレーションが可能であり、発進・停止、方向転換、腕振り歩行等のシミュレーションは行えない。このため、今回このモデルを腕可動型に改良し、両腕の運動等の安定歩行に対する影響等についても考察が可能なように、この歩行シミュレーションプログラムを改良・整備した。また、発進や停止動作についてのシミュレーション手法を開発し、これらのシミュレーションプログラムを作成した。但し、方向転換の歩行シミュレーションについては未着手である。

今回の研究の目的はいろいろな移動パターンに対する安定な歩行パターンを得ることであり、これまでに平坦地形における直進完全歩行の安定な歩行パターンの作成ができてい
る。本報告では、

- ①平坦地形における直進完全歩行シミュレーションの方法(発進から定常歩行への移行、定常歩行から停止への移行)

②定常歩行における両腕運動等の歩行安定性に対する影響

③現状の問題点

について述べ、今後の方針について議論する。

Table 1.1 Recent studies for biped locomotion at related organizations

研究機関	特 徴	論文発表年度
早 大	上体補償型 (ZMP規範)、非平坦路歩行、未知の外力下の歩行 (足踏み)、3次元動歩行 (完全歩行)、WL-12RⅢ	H 2
岐 阜 大	角運動量制御型、トルク配分問題、3次元動歩行 (発進+定常歩行) BLR-G 2	H 2
静 岡 大	環境 (床面) の変化に対する適応歩行、歩行パターン生成法、2次元動歩行	H 1
熊 本 大	左右方向の運動に関し、歩行機械の持つ固有のリズムに着目 (足踏み) 3次元動歩行、SR-3	H 1
機 械 技 研	線形倒立振子モード、足踏み、動歩行	H 2
神 戸 大	ロボットの軌道生成シミュレーション (ニューロ・コンピューティングを応用)	S 6 3
芝 浦 工 大 *	ロボット機構の最適形状設計、静歩行シミュレーション	H 1
立 命 館 大	ロボットの重心及び遊脚の位置制御 (シミュレーション)	H 1
名 大	足首トルクによる踏ん張り、モデルの低次元化、2次元動歩行	S 6 3
東 工 大	高さの異なる水平面の歩行、3次元動歩行 (完全歩行)、左右旋回、非人間型構造の2足ロボット	S 6 2
東京電機大	コントローラ・バッテリー自己搭載、静歩行、足踏み	H 1
九工大 久留米工大 *	人間の歩行動作の解析、上体の質量と歩容との関係 (シミュレーション)	H 2
信 州 大	方向転換 ($\pm 5^\circ$) シミュレーション、(直進に関しては動歩行であるが、方向転換については動力学を考慮せず)	H 2
M. Pupin (Yugoslavia)	人間の歩行動作の解析、歩行制御のシンセシス (シミュレーション)	H 1
原 研 *	3次元動歩行 (定常歩行) シミュレーション、人間型 (下肢・上肢とも) モデル	H 1

* ソフトウェアのみでの研究例

その他の研究機関 (S 6 0 以前)

東 大: 竹馬型 (倒立振子)、足踏み B I P E R - 3 S 5 5

阪 大: 歩行パターンの学習制御 S 5 9

千葉大: けり動作 2次元動歩行 S 5 8

2. 二足歩行ロボットモデル

2. 1 ロボットモデルの概略

対象とするロボットモデルはヴコブラトビッチの人間型モデル⁶⁾を基本にしたもので、人間のボディに関する質量、寸法、慣性モーメントのデータを元にした剛体系の二足歩行ロボットモデルである。

この剛体系の二足歩行ロボットのモデル図を Fig. 2.1 に、各ボディの寸法、質量、慣性モーメント及び重心位置を Fig. 2.2 に示す。全質量は約69.2 [Kg] である。また、ボディの番号を括弧付数字 ((1)~(12)) で、関節の番号を数字で表し、座標系を Fig. 2.1 のように取り回転は右ねじの方向を正とした。オリジナルのモデルは腕を前に組んだ腕固定型のモデルで、12ボディ、7関節、8自由度であるが、ここでは肩関節をY軸まわりで可動にし、肘関節は時間的には変化しないがY軸まわりとZ軸まわりで可変として、片腕に3自由度を持たせ肩に1自由度、肘に2自由度とし、肘は一定の角度を保つようにした。これは、現段階ではまだマニピュレーション(両腕作業等)のシミュレーションについては着手していないため、複雑な腕の動きを考慮していないことと、安定歩行に簡単な腕の動作がどのように影響してくるかを考察するためである。すなわち、本二足歩行ロボットは12ボディ、11関節からなるモデルで、関節2、5、7、8、10はY軸まわりの回転の1自由度を持ち、関節1、3、4、6、9、11はX軸とY軸まわりの回転の2自由度を持っている。但し、直進歩行を前提とし方向転換のための自由度は与えていない。また、下肢のX軸まわりの回転は両脚ともに等しく(後に説明する補償動作の1つとして、両脚の左右方向の傾きを共に θ で常に等しいと仮定)骨盤は回転しないため全体の自由度は下肢7自由度、腰1自由度、片腕3自由度の合計14自由度である。しかし、実際に時間的に関節角度を変化させる部分は、 $\beta_{1L/R}$ 、 $\beta_{2L/R}$ 、 $\beta_{4L/R}$ 、 θ 、 ϕ の8自由度で、前述のように肘の関節角度 $A_{11L/R}$ 、 $A_{12L/R}$ と足(Foot:常に床面に水平)の角度は一定値とする。なお、足の動きを時系列的に変化させることはプログラム上は可能であるが、床面との衝突(干渉)や安定領域を考慮して現在は床面に水平と仮定している。また、今後本文中足はFootを脚はLegを意味するものとする。

2. 2 ロボット運動方程式

歩行に対する安定判別にはZMP (Zero Moment Point) 規範^{6), 8)}を用いている。剛体系のZMPとは床面上にある点でその点に関してのロボットの各ボディの重力によるモーメント、慣性力によるモーメント、回転モーメントによるトルクの総和が零になる点である。

一般に絶対座標系の原点における剛体系の運動方程式は式(2.1)で表される。Mは原点におけるモーメントの和を表している。

$$M = \sum_{i=1}^n r_i \times m_i G + \sum_{i=1}^n r_i \times m_i \frac{d^2 r_i}{dt^2} + \sum_{i=1}^n \frac{d(J_i \omega_i)}{dt} \quad (2.1)$$

ここで、

n : ボディの数

m_i : ボディ i の質量

$r_{ip} = (X_i, Y_i, Z_i)$: ボディ i の絶対座標系での重心位置

$G = (0, 0, g)$, g : 重力加速度

$J_i = \begin{pmatrix} J_{xi} & 0 & 0 \\ 0 & J_{yi} & 0 \\ 0 & 0 & J_{zi} \end{pmatrix}$: ボディ i の重心まわりの慣性モーメント

$\omega_i = (\omega_{xi}, \omega_{yi}, \omega_{zi})^T$: ボディ i の角速度

ただし、 \times は外積である。

式(2.1)に関し床面上の点において、左辺のX成分とY成分が共に0になる点を、Zero Moment Point (ZMP) と呼び歩行中の安定な状態の指標として用いている。ZMPも1つの関節と考え、ZMPにおける式(2.1)のX成分とY成分が零になるという力学的拘束条件の基に運動方程式を定式化する。また、床面に作用する力(重力、慣性力の合力)に関しては式(2.2)のように表され、歩行系が床面と接しているためには式(2.2)のZ成分

2. 二足歩行ロボットモデル

2. 1 ロボットモデルの概略

対象とするロボットモデルはヴコブラトビッチの人間型モデル⁶⁾を基本にしたもので、人間のボディに関する質量、寸法、慣性モーメントのデータを元にした剛体系の二足歩行ロボットモデルである。

この剛体系の二足歩行ロボットのモデル図を Fig. 2.1 に、各ボディの寸法、質量、慣性モーメント及び重心位置を Fig. 2.2 に示す。全質量は約69.2 [Kg] である。また、ボディの番号を括弧付数字 ((1)~(12)) で、関節の番号を数字で表し、座標系を Fig. 2.1 のように取り回転は右ねじの方向を正とした。オリジナルのモデルは腕を前に組んだ腕固定型のモデルで、12ボディ、7関節、8自由度であるが、ここでは肩関節をY軸まわりで可動にし、肘関節は時間的には変化しないがY軸まわりとZ軸まわりで可変として、片腕に3自由度を持たせ肩に1自由度、肘に2自由度とし、肘は一定の角度を保つようにした。これは、現段階ではまだマニピュレーション（両腕作業等）のシミュレーションについては着手していないため、複雑な腕の動きを考慮していないことと、安定歩行に簡単な腕の動作がどのように影響してくるかを考察するためである。すなわち、本二足歩行ロボットは12ボディ、11関節からなるモデルで、関節2、5、7、8、10はY軸まわりの回転の1自由度を持ち、関節1、3、4、6、9、11はX軸とY軸まわりの回転の2自由度を持っている。但し、直進歩行を前提とし方向転換のための自由度は与えていない。また、下肢のX軸まわりの回転は両脚ともに等しく（後に説明する補償動作の1つとして、両脚の左右方向の傾きを共に θ で常に等しいと仮定）骨盤は回転しないため全体の自由度は下肢7自由度、腰1自由度、片腕3自由度の合計14自由度である。しかし、実際に時間的に関節角度を変化させる部分は、 $\beta_{1L/R}$ 、 $\beta_{2L/R}$ 、 $\beta_{4L/R}$ 、 θ 、 ϕ の8自由度で、前述のように肘の関節角度 $\Lambda_{11L/R}$ 、 $\Lambda_{12L/R}$ と足 (Foot: 常に床面に水平) の角度は一定値とする。なお、足の動きを時系列的に変化させることはプログラム上は可能であるが、床面との衝突（干渉）や安定領域を考慮して現在は床面に水平と仮定している。また、今後本文中足はFootを脚はLegを意味するものとする。

2. 2 ロボット運動方程式

歩行に対する安定判別にはZMP (Zero Moment Point) 規範^{6), 8)}を用いている。剛体系のZMPとは床面上にある点でその点に関してのロボットの各ボディの重力によるモーメント、慣性力によるモーメント、回転モーメントによるトルクの総和が零になる点である。

一般に絶対座標系の原点における剛体系の運動方程式は式(2.1)で表される。Mは原点におけるモーメントの和を表している。

$$M = \sum_{i=1}^n r_i \times m_i G + \sum_{i=1}^n r_i \times m_i \frac{d^2 r_i}{dt^2} + \sum_{i=1}^n \frac{d(J_i \omega_i)}{dt} \quad (2.1)$$

ここで、

n : ボディの数

m_i : ボディ i の質量

$r_{ip} = (X_i, Y_i, Z_i)$: ボディ i の絶対座標系での重心位置

$G = (0, 0, g)$, g : 重力加速度

$J_i = \begin{pmatrix} J_{xi} & 0 & 0 \\ 0 & J_{yi} & 0 \\ 0 & 0 & J_{zi} \end{pmatrix}$: ボディ i の重心まわりの慣性モーメント

$\omega_i = (\omega_{xi}, \omega_{yi}, \omega_{zi})^T$: ボディ i の角速度

ただし、 \times は外積である。

式(2.1)に関し床面上の点において、左辺のX成分とY成分が共に0になる点を、Zero Moment Point (ZMP) と呼び歩行中の安定な状態の指標として用いている。ZMPも1つの関節と考え、ZMPにおける式(2.1)のX成分とY成分が零になるという力学的拘束条件の基に運動方程式を定式化する。また、床面に作用する力(重力、慣性力の合力)に関しては式(2.2)のように表され、歩行系が床面と接しているためには式(2.2)のZ成分

が下向き、すなわち式(2.3)が負でなければならない。

$$E = - \sum_{i=1}^n m_i \left(\frac{d^2 r_i}{d t^2} + G \right) \quad (2.2)$$

$$E_z = - \sum_{i=1}^n m_i (\ddot{Z}_i + g) < 0 \quad (2.3)$$

式(2.3)を満たし、ZMPがロボットの足底接地面の形成する多角形の安定領域内に存在していれば、ロボットはZMPで地面に固定され転倒することなく安定に歩行できる。

なお、ここでは以下のような仮定を設けている。

- ①ロボットのボディは剛体とする。
- ②床面は水平な平面で変形しない。
- ③床面での並進（X軸及びY軸方向）に対する摩擦力は十分に大きい。
- ④システムの床面との接地状況は点接地の集合とする。
- ⑤システムが1点接地の場合、床面での回転（X、Y及びZ軸まわり）に対する摩擦係数は無視できるほど小さい。
- ⑥システムには関節駆動トルク以外の外力は作用しない。
- ⑦直進歩行を対象とし、単脚支持期だけで両脚支持期を持たない。

また、式(2.1)で表される運動方程式は一般的に次の微分方程式で表現される⁹⁾。

$$[M] \ddot{[E]} + [B] \dot{[E]} + [C] [E] + [G] \quad (2.4)$$

ここで、一般化された駆動力（一般力）ベクトルを $[M]$ 、状態変数を $[E]$ で表し、係数行列 $[A]$ 、 $[B]$ 、 $[C]$ 及びベクトル $[G]$ は、定式化に使われた一般化座標の関数である。右辺第一項は慣性項、第二項は遠心力の項、第三項はコリオリ力の項、第四項は重力項である。ここでは $[E]$ は関節角変数である。 $[M]$ は1つの関節角に対しX、Y、Z方向の3つずつ存在するため、 $3n$ 次元（ n ：関節数）のベクトルとなる。 $[A]$ 、 $[B]$ は $3n \times m$ 行列（ m ：関節角変数の数）、 $[C]$ は $3n \times {}_m C_2$ 行列、 $[G]$ は $3n$ 次元のベクトルとなる。

この問題については以下の3つのタイプに大別できる。

- ①システムの運動が既知の場合

未知の一般化された駆動力は状態変数に既知の値を代入することによって、簡単に代数的に計算できる。

②駆動力が既知の場合

式(2.4)の一般化された加速度が式(2.5)で表現できる。

$$\{\ddot{\Xi}\} = [A]^{-1} \left(\{M\} - [B] \{\dot{\Xi}^2\} - [C] \{\dot{\Xi}\dot{\Xi}\} - \{G\} \right) \quad (2.5)$$

よって、 $\{\Xi\}$ 、 $\{\dot{\Xi}\}$ の初期条件を仮定し、式(2.5)を数値積分することで未知の運動を求めることができる。

③運動及び駆動力が部分的に既知の場合

運動及び駆動力は既知な部分 $\{\Xi_0\}$ 、 $\{M_0\}$ と、未知な部分 $\{\Xi_x\}$ 、 $\{M_x\}$ に分離できる。分離する変換行列を $[P]$ 、 $[R]$ とすると、

$$\begin{bmatrix} P_0 \\ \dots \\ P_x \end{bmatrix} \{M\} = \begin{Bmatrix} M_0 \\ \dots \\ M_x \end{Bmatrix} \quad (2.6)$$

$$[R_0 \ ; \ R_x] \begin{Bmatrix} \Xi_0 \\ \dots \\ \Xi_x \end{Bmatrix} = \{\Xi\}$$

行列 $[A]$ に対する演算において導出される次項を以下のように表す。

$$\begin{bmatrix} P_0 \\ \dots \\ P_x \end{bmatrix} [A] [R_0 \ ; \ R_x] = \begin{bmatrix} A_{00} & A_{0x} \\ \dots & \dots \\ A_{x0} & A_{xx} \end{bmatrix} \quad (2.7)$$

式(2.4)をまとめ直し、未知量 $\{\Xi_x\}$ 、 $\{M_x\}$ を求めると式(2.8)、式(2.9)のようになる。

$$\begin{aligned} \{\ddot{\Xi}_x\} &= [A_{xx}]^{-1} \left(\{M_0\} - [A_{00}] \{\ddot{\Xi}_0\} \right. \\ &\quad \left. - [P_0] \left([B] \{\dot{\Xi}^2\} + [C] \{\dot{\Xi}\dot{\Xi}\} + \{G\} \right) \right) \quad (2.8) \end{aligned}$$

$$\begin{aligned} \{M_x\} = & \{A_{xx}\} \{\ddot{x}\} + \{A_{x0}\} \{\ddot{0}\} \\ & + \{P_x\} (\{B\} \{\dot{x}^2\} + \{C\} \{\dot{x}\dot{x}\} + \{G\}) \end{aligned} \quad (2.9)$$

ZMP規範型の場合、前述のように運動方程式に対する拘束条件がZMPにおける式(2.1)のX成分とY成分が0になるという2つしかないため一部の運動を規定し、安定な歩行パターンとなるように補償動作を計算することになる。よって、ここでのモデルは上記の③のタイプに相当する。

2.3 安定領域

前節でも述べたように、歩行系が床面と接地しているためには式(2.3)を満たす必要がある。ここで安定領域についてのべる時にこの条件は常に満足してあるものとする。

この安定領域に関しては文献9)に詳しいのでここでは簡単に説明する。各接地点におけるモーメント(トルク)についてのつり合いを考えると、全ての接地点が常に接地している歩行をさせるためには、床面上でモーメントの和が0になる点、すなわち既に述べたZMPは接地点が形成する多角形の内側に存在しなければならない。ここではロボットの足は床面と常に水平であると仮定しているため足底面が床面と面接地していることになり、仮定④から多点接地と等価と見なして考える。本モデルでは足底の形状を長方形としているため単脚支持の場合4つの角による4点接地と見なし、これらの点が全て接地している歩行を行わせる。この足底面の4点で形成される領域が安定領域である。また、両脚支持の場合は各接地点を結ぶ線分によって作られる多角形のうち面積の最大となる多角形(凸多角形)が安定領域⁹⁾となる(Fig. 2.3)。

この安定領域は静歩行、動歩行ともに同じであるが、一般に静歩行の場合はロボットシステム全体の重心の床面への投影点が安定領域内に存在するように歩行させるもので、重心の静力学的なつり合いによりシステムを安定化する歩行である。この歩行は慣性力等の影響を無視できる程度の歩行速度となるため、かなりゆっくりとした歩行になる。それに対し、動歩行は前述のZMPが安定領域に存在するように歩行させるもので、重心の位置のみに着目した静歩行とは違い、重心及びリンク各部の位置、速度、加速度を考慮し動力学(dynamics)的に計算された歩行である。ここでのシミュレーションはこの動歩行を取り上げている。

2. 4 腕可動（能動型）モデルの計算法

2. 4. 1 運動方程式

2.2節で述べたように運動方程式に対する拘束条件がZMPにおける式(2.1)のX成分とY成分が0になるという2つしかないため、本来動作から決定される矢状面（X-Z平面）内の下肢の関節角（歩容：歩き方）、腕の振り角及びこの時の歩行が安定歩行となるように安定領域内で作成したZMP軌跡をデータで入力する。これらをまとめて基本歩行パターンと呼ぶ。この安定領域内に規定したZMP軌跡に追従させるための補償動作として、上体の矢状面内の動き ϕ （腰の曲がる角度）と両脚の前頭面（Y-Z面）内の左右の振れ θ を求める（Fig. 2.1 参照）。ここで両脚の左右の振れを同じにしているのは、片方の脚だけの動作では両脚の衝突の問題が生じてくるためである。

式(2.1)についてZMPを原点として方程式を定式化する。ZMPを原点にとることによって、直接式(2.1)のX成分とY成分を零と置くことができる。まず右辺の第一項及び第二項の導出について述べる。この項の計算には各ボディの重心位置座標が必要になる。この場合ZMPを原点とした座標系（ZMP座標系と呼ぶ、絶対座標系の原点をZMPまで平行移動したもの）で表示するため Fig. 2.1 に示したように、絶対座標系の原点からのZMP位置を ΔX 、 ΔY を用いて表す。なお、各ボディの寸法に関するパラメータについては以下のように定義する（Fig. 2.2 参照）。

- F : 下腿の1/2の長さ
- O : 大腿の1/2の長さ
- Q : 腰（関節No.7）から上体（胴）の重心までの高さ
- RR : 腰から肩関節までの高さ
- E : 上腕の1/2の長さ
- SS : 前腕の1/2の長さ
- H : 足の重心から足首関節までの高さ
- D : 骨盤の幅の1/2
- FF : 肩の関節から首の推軸までの長さ

PP : 股関節から骨盤の重心までの高さ

UH : 足裏から足の重心までの高さ

各ボディの重心のX座標

$$X_{12} = -\Delta X$$

$$X_1 = X_{12} + F \sin \beta_{2L}$$

$$X_2 = X_1 + F \sin \beta_{2L} + O \sin \beta_{1L}$$

$$X_3 = X_2 + O \sin \beta_{1L}$$

$$X_4 = X_3 + Q \sin \phi$$

$$X_5 = X_3 + R R \sin \phi - E \sin \beta_{4L}$$

$$X_6 = X_3 + R R \sin \phi - 2 E \sin \beta_{4L} - S S \sin(-A_{12L} + \beta_{4L}) \cos A_{11L}$$

$$X_7 = X_3 + R R \sin \phi - E \sin \beta_{4R}$$

$$X_8 = X_3 + R R \sin \phi - 2 E \sin \beta_{4R} - S S \sin(-A_{12R} + \beta_{4R}) \cos A_{11R}$$

$$X_9 = X_3 - O \sin \beta_{1R}$$

$$X_{10} = X_3 - 2 O \sin \beta_{1R} - F \sin \beta_{2R}$$

$$X_{11} = X_3 - 2 O \sin \beta_{1R} - 2 F \sin \beta_{2R}$$

各ボディの重心のY座標

$$Y_{12} = -\Delta Y$$

$$Y_1 = Y_{12} - F \cos \beta_{2L} \sin \theta$$

$$Y_2 = 2 Y_1 - O \cos \beta_{1L} \sin \theta - Y_{12}$$

$$Y_3 = Y_2 - O \cos \beta_{1L} \sin \theta - D$$

$$Y_4 = Y_3$$

$$Y_5 = Y_4 + F F$$

$$Y_6 = Y_5 - S S \sin A_{11L}$$

$$Y_7 = Y_4 - F F$$

$$Y_8 = Y_7 + S S \sin A_{11R}$$

$$Y_9 = Y_3 + O \cos \beta_{1R} \sin \theta - D$$

$$Y_{10} = Y_9 + (O \cos \beta_{1R} + F \cos \beta_{2R}) \sin \theta$$

$$Y_{11} = Y_{10} + F \cos \beta_{2R} \sin \theta$$

各ボデーの重心のZ座標

$$\begin{aligned}
 Z_{12} &= UH \\
 Z_1 &= Z_{12} + F \cos \beta_{2L} \cos \theta + H \\
 Z_2 &= Z_1 + (F \cos \beta_{2L} + O \cos \beta_{1L}) \cos \theta \\
 Z_3 &= Z_2 + O \sin \beta_{1L} \cos \theta + P P \\
 Z_4 &= Z_3 + Q \cos \phi + P P \\
 Z_5 &= Z_3 + R R \cos \phi - E \cos \beta_{4L} + P P \\
 Z_6 &= Z_3 + R R \cos \phi - 2 E \cos \beta_{4L} - S S \cos(-A_{12L} + \beta_{4L}) \cos A_{11L} \\
 &\quad + P P \\
 Z_7 &= Z_3 + R R \cos \phi - E \cos \beta_{4R} + P P \\
 Z_8 &= Z_3 + R R \cos \phi - 2 E \cos \beta_{4R} - S S \cos(-A_{12R} + \beta_{4R}) \cos A_{11R} \\
 &\quad + P P \\
 Z_9 &= Z_3 - O \cos \beta_{1R} \cos \theta - P P \\
 Z_{10} &= Z_9 - (O \cos \beta_{1R} + F \cos \beta_{2R}) \cos \theta \\
 Z_{11} &= Z_{10} - F \cos \beta_{2R} \cos \theta - H
 \end{aligned}
 \tag{2.10}$$

式(2.10)を用いて式(2.1)に従い第一項、第二項を導く。

次に式(2.1)の第三項の導出について述べる。回転モーメントに関する第三項の一般形は次のように導かれる(角運動量の微分)。

$$\begin{aligned}
 \frac{d(J_i \omega_i)}{dt} &= \frac{d}{dt} [J_i \omega_i] + \omega_i \times (J_i \omega_i) \\
 &= \left[\begin{array}{l} J_{x_i} \dot{\omega}_{x_i} + \omega_{y_i} \omega_{z_i} (J_{z_i} - J_{y_i}) \\ J_{y_i} \dot{\omega}_{y_i} + \omega_{z_i} \omega_{x_i} (J_{x_i} - J_{z_i}) \\ J_{z_i} \dot{\omega}_{z_i} + \omega_{x_i} \omega_{y_i} (J_{y_i} - J_{x_i}) \end{array} \right]
 \end{aligned}
 \tag{2.11}$$

ここで、 $\frac{d}{dt} [\]$ は運動座標系を固定座標系とみなした場合の微分を表す。

本モデルは1関節に最大2自由度を持っており、ボディの回転はY軸まわり→X軸まわりの順に行うように定義している。一般的にこの回転角をY軸まわり Ψ 、X軸まわり Θ と表すと、運動座標系における角速度とボディの回転角（オイラー角）との関係は式(2.12)で表される。

$$\omega_i = \begin{pmatrix} \omega_{ix} \\ \omega_{iy} \\ \omega_{iz} \end{pmatrix} = \begin{pmatrix} \dot{\Theta}_i \cos \Theta_i \\ \dot{\Psi}_i \\ \dot{\Theta}_i \sin \Theta_i \end{pmatrix} \quad (2.12)$$

式(2.12)を式(2.11)に代入し、式(2.11)を m_{Ri} で表し整理すると、

$$m_{Ri} = \begin{pmatrix} \ddot{\Theta}_i \cos \Psi_i J_{xi} - \dot{\Theta}_i \dot{\Psi}_i \sin \Psi_i (J_{zi} - J_{yi} - J_{xi}) \\ \ddot{\Psi}_i J_{yi} + \dot{\Theta}_i^2 \cos \Psi_i \sin \Psi_i (J_{xi} - J_{zi}) \\ \ddot{\Theta}_i \sin \Psi_i J_{zi} - \dot{\Theta}_i \dot{\Psi}_i \cos \Psi_i (J_{zi} + J_{yi} - J_{xi}) \end{pmatrix} \quad (2.13)$$

式(2.13)は運動座標系に対する表現であり、これを絶対座標系に変換すると以下のようになる。

$$\begin{aligned} M_{Ri} &= (M_{Rxi}, M_{Ryi}, M_{Rzi}) : \\ M_{Rxi} &= \ddot{\Theta}_i (J_{xi} \cos^2 \Psi_i + J_{zi} \sin^2 \Psi_i) \\ &\quad - 2 (J_{xi} - J_{zi}) \dot{\Theta}_i \dot{\Psi}_i \sin \Psi_i \cos \Psi_i \\ M_{Ryi} &= \ddot{\Theta}_i (J_{xi} - J_{zi}) \sin \Psi_i \cos \Psi_i \sin \Theta_i + \ddot{\Psi}_i J_{yi} \cos \Theta_i \\ &\quad + \dot{\Theta}_i \dot{\Psi}_i \sin \Theta_i \{ (J_{xi} - J_{zi}) (\cos^2 \Psi_i - \sin^2 \Psi_i) - J_{yi} \} \\ &\quad + \dot{\Theta}_i^2 (J_{xi} - J_{zi}) \sin \Psi_i \cos \Psi_i \cos \Theta_i \\ M_{Rzi} &= -\ddot{\Theta}_i (J_{xi} - J_{zi}) \sin \Psi_i \cos \Psi_i \cos \Theta_i + \ddot{\Psi}_i J_{yi} \sin \Theta_i \\ &\quad + \dot{\Theta}_i \dot{\Psi}_i \cos \Theta_i \{ (J_{xi} - J_{zi}) (\sin^2 \Psi_i - \cos^2 \Psi_i) + J_{yi} \} \\ &\quad + \dot{\Theta}_i^2 (J_{xi} - J_{zi}) \sin \Psi_i \cos \Psi_i \sin \Theta_i \end{aligned} \quad (2.14)$$

よって、式(2.1)の第三項、すなわち12ボディの各々における回転モーメントのX、Y成

分は式(2.14)から次のように導かれる。

$$\begin{aligned}
 M_{RX1} &= \ddot{\theta} (J_{x1} \cos^2 \beta_{2L} + J_{z1} \sin^2 \beta_{2L}) \\
 &\quad - 2 (J_{x1} - J_{z1}) \dot{\theta} \dot{\beta}_{2L} \sin \beta_{2L} \cos \beta_{2L} \\
 M_{RX2} &= \ddot{\theta} (J_{x2} \cos^2 \beta_{1L} + J_{z2} \sin^2 \beta_{1L}) \\
 &\quad - 2 (J_{x2} - J_{z2}) \dot{\theta} \dot{\beta}_{1L} \sin \beta_{1L} \cos \beta_{1L} \\
 M_{RX3}, M_{RX4}, M_{RX5}, M_{RX6}, M_{RX7}, M_{RX8} &= 0 \\
 M_{RX9} &= \ddot{\theta} (J_{x9} \cos^2 \beta_{1R} + J_{z9} \sin^2 \beta_{1R}) \\
 &\quad - 2 (J_{x9} - J_{z9}) \dot{\theta} \dot{\beta}_{1R} \sin \beta_{1R} \cos \beta_{1R} \\
 M_{RX10} &= \ddot{\theta} (J_{x10} \cos^2 \beta_{2R} + J_{z10} \sin^2 \beta_{2R}) \\
 &\quad - 2 (J_{x10} - J_{z10}) \dot{\theta} \dot{\beta}_{2R} \sin \beta_{2R} \cos \beta_{2R} \\
 M_{RX11}, M_{RX12} &= 0
 \end{aligned}$$

$$\begin{aligned}
 M_{RY1} &= \ddot{\theta} (J_{x1} - J_{z1}) \sin \beta_{2L} \cos \beta_{2L} \sin \theta + \ddot{\beta}_{2L} J_{y1} \cos \theta \\
 &\quad + \dot{\theta} \dot{\beta}_{2L} \sin \theta \{ (J_{x1} - J_{z1}) (\cos^2 \beta_{2L} - \sin^2 \beta_{2L}) - J_{y1} \} \\
 &\quad + \dot{\theta}^2 (J_{x1} - J_{z1}) \sin \beta_{2L} \cos \beta_{2L} \cos \theta \\
 M_{RY2} &= \ddot{\theta} (J_{x2} - J_{z2}) \sin \beta_{1L} \cos \beta_{1L} \sin \theta + \ddot{\beta}_{1L} J_{y2} \cos \theta \\
 &\quad + \dot{\theta} \dot{\beta}_{1L} \sin \theta \{ (J_{x2} - J_{z2}) (\cos^2 \beta_{1L} - \sin^2 \beta_{1L}) - J_{y2} \} \\
 &\quad + \dot{\theta}^2 (J_{x2} - J_{z2}) \sin \beta_{1L} \cos \beta_{1L} \cos \theta \\
 M_{RY3} &= 0 \\
 M_{RY4} &= J_{y4} \ddot{\psi} \\
 M_{RY5} &= J_{y5} \ddot{\beta}_{4L} \\
 M_{RY6} &= J_{y6} \ddot{\beta}_{4L} \\
 M_{RY7} &= J_{y7} \ddot{\beta}_{4R} \\
 M_{RY8} &= J_{y8} \ddot{\beta}_{4R} \\
 M_{RY9} &= \ddot{\theta} (J_{x9} - J_{z9}) \sin \beta_{1R} \cos \beta_{1R} \sin \theta + \ddot{\beta}_{1R} J_{y9} \cos \theta \\
 &\quad + \dot{\theta} \dot{\beta}_{1R} \sin \theta \{ (J_{x9} - J_{z9}) (\cos^2 \beta_{1R} - \sin^2 \beta_{1R}) - J_{y9} \} \\
 &\quad + \dot{\theta}^2 (J_{x9} - J_{z9}) \sin \beta_{1R} \cos \beta_{1R} \cos \theta \\
 M_{RY10} &= \ddot{\theta} (J_{x10} - J_{z10}) \sin \beta_{2R} \cos \beta_{2R} \sin \theta + \ddot{\beta}_{2R} J_{y10} \cos \theta \\
 &\quad + \dot{\theta} \dot{\beta}_{2R} \sin \theta \{ (J_{x10} - J_{z10}) (\cos^2 \beta_{2R} - \sin^2 \beta_{2R}) - J_{y10} \}
 \end{aligned}$$

$$\begin{aligned}
 & + \ddot{\theta}^2 (J_{x10} - J_{z10}) \sin \beta_{2R} \cos \beta_{2R} \cos \theta \\
 M_{RY11}, M_{RY12} & = 0
 \end{aligned}
 \tag{2.15}$$

式(2.1)の第三項の計算はこのように他の2項と比べて若干複雑であり、式全体の形を明確に把握しにくい。これは剛体系モデル特有のもので、ロボットの設計上剛体系でモデル化することは重要であるが、もし質量がボディの一部に集中しているような場合、モデルを質点系で記述¹⁰⁾することができ、この回転モーメントを示す第三項は省略できる。この場合、運動方程式全体がすっきりとした形になり多自由度モデルの記述においても式全体の見通しがよくなる、さらに計算時間も短縮されるというメリットもある。

式(2.8)の運動方程式は非線形の連立常微分方程式であり、基本歩行パターンの他に初期値として θ 、 ϕ の角度と角速度を与える必要がある。式(2.4)の解法③で示した未知量 $[M_x]$ を θ 、 ϕ とし、既知なモーメント $[M_0]$ はZMPにおけるX成分とY成分が0であるという拘束条件とすると、式(2.8)より以下の方程式を得る。

$$\begin{aligned}
 \begin{bmatrix} M_{XZMP} \\ M_{YZMP} \end{bmatrix} & = [E] \begin{bmatrix} \ddot{\theta} \\ \ddot{\phi} \end{bmatrix} + [F] \begin{bmatrix} \dot{\theta}^2 \\ \dot{\phi}^2 \end{bmatrix} + [H] [\dot{\theta} \dot{\phi}] + [K] \\
 & = \begin{bmatrix} 0 \\ 0 \end{bmatrix}
 \end{aligned}
 \tag{2.16}$$

E : 2 × 2 行列、F : 2 × 2 行列、H : 2 × 1 行列

K : 2 × 1 行列

式(2.16)で表される運動方程式を解き基本歩行パターンに対する補償動作の推移を計算するため、ここでは1秒を30分割して各タイムステップ(t1~t31)において運動方程式を解いて補償動作を決定する。ここで1秒の分割数を30としたのは、これらのシミュレーション結果を動画として表示する際に使用するVTRの1秒の駒割数が30であるためである。また、歩行時間は人間の歩行を前提に1歩1秒としている。

なお、腕固定型モデルの場合は上記の腕の角度($\beta_{L/R}$)を上体の動き(ϕ)と同じものとして解析できる。この場合、式(2.16)の係数行列が変わる。

2. 4. 2 境界条件

式(2.8)に基づき補償動作 (θ 、 ϕ) を決定するためには、前述のように基本歩行パターンの他に θ と ϕ の初期角度と初期角速度を入力する必要がある。この初期値の推定に関しては、定常歩行を取り上げ歩行動作の周期性と1歩ずれた左右対称性から、1歩の始め ($t = 0$ 秒) と終わり ($t = 1$ 秒) で θ 、 ϕ の角度と角速度が一致するという境界条件を設定している。正確には2歩で始めと全く同じ姿勢になるが、1歩について考えれば回転の方向を考慮して、周期解を得るための初期条件は次の式(2.17)で表される。

$$\begin{aligned}
 \theta |_{t=0} &= -\theta |_{t=1} \\
 \dot{\theta} |_{t=0} &= -\dot{\theta} |_{t=1} \\
 \phi |_{t=0} &= \phi |_{t=1} \\
 \dot{\phi} |_{t=0} &= \dot{\phi} |_{t=1}
 \end{aligned}
 \tag{2.17}$$

2. 4. 3 数値計算

入力された初期値 ($t = 0$ 秒での θ と ϕ の初期角度と初期角速度) から1歩後の最終値を計算するためには、式(2.16)より得られる θ と ϕ の角加速度と入力初期値を用いて、次のタイムステップにおける θ と ϕ の角度と角速度を数値計算し、再び式(2.16)より θ と ϕ の角加速度を計算するという形になる。これを1歩の間 (1秒間) 繰り返して計算をする。この数値計算にはルンゲ・クッタ、アダムス・モールトン (Runge Kutta, Adams Moulton) 法¹⁾を用いている。具体的には、はじめの3つのタイムステップ (t_2 から t_4 、 t_1 は入力値) は式(2.18)で示されるルンゲ・クッタ法を用いて数値計算を行い、これらの値を元に次のタイムステップ (t_5 から t_{31}) からはアダムス・モールトン予測子・修正子法を用いる。予測子としては式(2.19)で示されるアダムス・バッシュフォースの4次の陽解法を使用し初期の予測値を得る。そして、修正子として式(2.20)で示されるアダムス・モールトンの4次の陰解法を用いてより精度の高い修正値を得る。以下に数値計算の数式を示す。

$$\begin{aligned}
x^{(1)}_n &= x_n + \frac{h}{2} f^{(1)}_n, & f^{(1)}_n &= f(x_n, t_n), \\
x^{(2)}_n &= x_n + \frac{h}{2} f^{(2)}_n, & f^{(2)}_n &= f(x^{(1)}_n, t_n+h/2), \\
x^{(3)}_n &= x_n + \frac{h}{2} f^{(3)}_n, & f^{(3)}_n &= f(x^{(2)}_n, t_n+h/2), \\
x_{n+1} &= x_n + \frac{h}{6} (f^{(1)}_n + 2f^{(2)}_n + 2f^{(3)}_n + f^{(4)}_n), \\
&& f^{(4)}_n &= f(x^{(3)}_n, t_{n+1})
\end{aligned} \tag{2.18}$$

$$x_{n+1} = x_n + \frac{h}{24} (55\ddot{x}_n - 59\ddot{x}_{n-1} + 37\ddot{x}_{n-2} - 9\ddot{x}_{n-3}) \tag{2.19}$$

$$x_{n+1} = x_n + \frac{h}{720} (251\ddot{x}_{n+1} + 646\ddot{x}_n - 264\ddot{x}_{n-1} + 106\ddot{x}_{n-2} - 19\ddot{x}_{n-3}) \tag{2.20}$$

ここで、

t_n : 時間 ($t = 0 \sim 1$ sec.)

$h = 1/30$: タイムステップ

$x_n = (\theta_n, \phi_n, \dot{\theta}_n, \dot{\phi}_n)$

$f(x_n, t_n) = (\ddot{\theta}_n, \ddot{\phi}_n, \ddot{\theta}_n, \ddot{\phi}_n) |_{t=t_n}$

補償動作の決定に関する初期値探索のフローチャートを Fig. 2.4 に示す。図でcheckとあるのは推定初期値に対して補償動作の関節角が異常に増大した場合、収束しない(周期解にならない)ものと判断して処理を打ち切るためのものである。また、DERIVは運動方程式(式(2.16))を解くサブプログラム名である(付録A参照)。

このようにして1歩後の最終値(θ 、 ϕ の角度、角速度)を計算するが、式(2.17)で表される境界条件を満足する解を得るためにはこの入力初期値についての思考錯誤が必要である。

2. 4. 4 初期値修正法

運動方程式の境界条件を満足する周期解を得るためにはまず適当な推定初期値を入力する必要がある。しかし、この入力初期値がこのまま周期解になることはまずない。このため、1歩分の計算の後この境界条件のチェックを行い、境界条件が満足されなければ傾斜法または局所的方法を用いてこの入力初期値を修正し、境界条件を満足する初期値が見つかるまでこの処理を繰り返し行う。また、推定入力初期値によっては補償動作の関節角が増大し収束しない場合もあるため、いくつかの推定初期値に対して初期値修正が行えるようにプログラムした。この関節角の増大が起こるのは初期値、特に角速度が不適當であるためであるが、もう一つの要因として安定領域内に入るように作成したZMP軌跡が上げられる。本システムは歩行が安定となるように安定領域内に入るZMP軌跡をデータで入力し、これに追従するように上体の前後方向の動きや脚の左右の動きを決定している（補償動作）。このため、このZMP軌跡の取り方の違いで補償動作はいろいろに変化し、補償動作の入力初期値とZMP軌跡の組み合わせによっては収束しない場合も生じる。

前述の式(2.17)の境界条件に基づき解を求める。 $DEP = [\theta, \phi, \dot{\theta}, \dot{\phi}]^T$ と置き、初期値を DEP^0 、最終値を DEP^1 で表す。また、繰り返し条件（境界条件）を表すための初期値と最終値の関係を示す行列 η は以下のように表せ、初期値に η を乗じたものと最終値の差が評価値となる。

$$\eta = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.21)$$

評価値最小化に基づき以下の評価関数を用いる。なお、最終値は初期値の関数として表せる。

$$DE(DEP^0) = \|\eta DEP^0 - DEP^1\| \quad (2.22)$$

式(2.22)を最小にする DEP^0 を求めるために2つの修正法を用いる。

①傾斜法

推定初期値は次の反復法により修正される。これは、 i 番目での変化率から $i+1$ 番目の値を推定するものである。

$$DEP^0_{i+1} = DEP^0_i - \varepsilon \text{grad}(DE(DEP^0_i)) \quad (2.23)$$

具体的に $\text{grad}(DE(DEP^0_i)) = \nabla DE$ を計算する。ここで、 i は繰り返し回数、 ε は収束のための係数で単調に収束が見られない時にこのステップの半減を行うものである。

$$DE = \{(\theta^0 + \theta^1)^2 + (\phi^0 - \phi^1)^2 + (\tilde{\theta}^0 + \tilde{\theta}^1)^2 + (\tilde{\psi}^0 - \tilde{\psi}^1)^2\}^{1/2} \quad (2.24)$$

$$\nabla DE = [\nabla_1 DE, \nabla_2 DE, \nabla_3 DE, \nabla_4 DE]^T \quad (2.25)$$

$$\nabla_1 DE = \{(\theta^0 + \theta^1)(1 + \Delta\theta^1/\Delta\theta^0) - (\phi^0_i - \phi^1_i) \Delta\phi^1/\Delta\theta^0 + (\tilde{\theta}^0 + \tilde{\theta}^1) \Delta\tilde{\theta}^1/\Delta\theta^0 - (\tilde{\psi}^0_i - \tilde{\psi}^1_i) \Delta\tilde{\psi}^1/\Delta\theta^0\} / DE$$

$$\nabla_2 DE = \{(\theta^0 + \theta^1) \Delta\theta^1/\Delta\phi^0 + (\phi^0_i - \phi^1_i)(1 - \Delta\phi^1/\Delta\phi^0) + (\tilde{\theta}^0 + \tilde{\theta}^1) \Delta\tilde{\theta}^1/\Delta\phi^0 - (\tilde{\psi}^0_i - \tilde{\psi}^1_i) \Delta\tilde{\psi}^1/\Delta\phi^0\} / DE$$

$$\nabla_3 DE = \{(\theta^0 + \theta^1) \Delta\theta^1/\Delta\tilde{\theta}^0 - (\phi^0_i - \phi^1_i) \Delta\phi^1/\Delta\tilde{\theta}^0 + (\tilde{\theta}^0 + \tilde{\theta}^1)(1 + \Delta\tilde{\theta}^1/\Delta\tilde{\theta}^0) - (\tilde{\psi}^0_i - \tilde{\psi}^1_i) \Delta\tilde{\psi}^1/\Delta\tilde{\theta}^0\} / DE$$

$$\nabla_4 DE = \{(\theta^0 + \theta^1) \Delta\theta^1/\Delta\tilde{\psi}^0 - (\phi^0_i - \phi^1_i) \Delta\phi^1/\Delta\tilde{\psi}^0 + (\tilde{\theta}^0 + \tilde{\theta}^1) \Delta\tilde{\theta}^1/\Delta\tilde{\psi}^0 + (\tilde{\psi}^0_i - \tilde{\psi}^1_i)(1 - \Delta\tilde{\psi}^1/\Delta\tilde{\psi}^0)\} / DE$$

②局所的方法

この方法は推定初期値が周期解に十分近い場合に有効である。これは i 番目の周期解の近傍で線型関係を仮定し、部分行列がヤコビアン (Jacobian) の形からなる感度行列で $i+1$ 番目の値を修正するものである。もし、用いている初期値 DEP^0_i が最終的な周期解 DEP^{0*} の近傍にあるならば、最終値は次のように表される。

$$DEP^{0*} + \Delta DEP^{0*}_i = \eta (DEP^{0*} + \Delta DEP^0_i) \quad (2.26)$$

このとき、一步の始めと終わりでの微小なずれの間には以下の関係を仮定することができる。

$$\Delta \text{DEP}_{i'} = \frac{\partial (\eta \text{DEP}^0)}{\partial \text{DEP}^0} \Delta \text{DEP}^0_i \quad (2.27)$$

式(2.26)及び式(2.27)より、初期値の修正値を次のように得ることができる。

$$\Delta \text{DEP}^0_{i'} = \phi(\text{DEP}^0_i) \quad (2.28)$$

よって、反復法は以下のようになる。

$$\text{DEP}^0_{i+1} = \text{DEP}^0_i + \varepsilon \phi(\text{DEP}^0_i) \quad (2.29)$$

ここで、パラメータ i 、 ε は①の方法の場合と同じ意味を持つ。また、関数 ϕ は次のように表される。

$$\begin{aligned} \phi &= \mathbf{A}^{-1} \mathbf{q} \\ \mathbf{q} &= (-\theta^0 - \theta^1, \phi^0 - \phi^1, -\dot{\theta}^0 - \dot{\theta}^1, \dot{\psi}^0 - \dot{\psi}^1)^T \end{aligned} \quad (2.30)$$

$$\mathbf{A} = \begin{pmatrix} \Delta \theta^1 / \Delta \theta^0 + 1 & \Delta \theta^1 / \Delta \phi^0 & \Delta \theta^1 / \Delta \dot{\theta}^0 & \Delta \theta^1 / \Delta \dot{\psi}^0 \\ \Delta \phi^1 / \Delta \theta^0 & \Delta \phi^1 / \Delta \phi^0 - 1 & \Delta \phi^1 / \Delta \dot{\theta}^0 & \Delta \phi^1 / \Delta \dot{\psi}^0 \\ \Delta \dot{\theta}^1 / \Delta \theta^0 & \Delta \dot{\theta}^1 / \Delta \phi^0 & \Delta \dot{\theta}^1 / \Delta \dot{\theta}^0 + 1 & \Delta \dot{\theta}^1 / \Delta \dot{\psi}^0 \\ \Delta \dot{\psi}^1 / \Delta \theta^0 & \Delta \dot{\psi}^1 / \Delta \phi^0 & \Delta \dot{\psi}^1 / \Delta \dot{\theta}^0 & \Delta \dot{\psi}^1 / \Delta \dot{\psi}^0 - 1 \end{pmatrix} \quad (2.31)$$

ここでは、①、②の方法とも ΔDEP が必要になるが、 θ^0 、 ϕ^0 、 $\dot{\theta}^0$ 、 $\dot{\psi}^0$ のそれぞれに対して 0.005 変化させた場合を用いて計算を行う。すなわち、 $\Delta \theta^0$ 、 $\Delta \phi^0$ 、 $\Delta \dot{\theta}^0$ 、 $\Delta \dot{\psi}^0$ は 0.005 であり、 $\Delta \theta^1$ 、 $\Delta \phi^1$ 、 $\Delta \dot{\theta}^1$ 、 $\Delta \dot{\psi}^1$ は 0.005 変化させた時と元の場合との最終値の差となる。

ここでは上記の2つの修正法を併用して解を得ている。始めの推定初期値による評価値DEがある基準値より大きい場合には①の方法を、そうでない場合には②の方法を用いる。通常は①の方法で、ある基準値までDEを小さくし、その後②の方法に切り換えて行う。

2.5 腕可動（受動型）モデルの計算法

2.4節で述べた腕可動（能動型）モデルに対し、ここでのモデルは定常歩行時において受動的に動作させる腕の挙動を計算するものである。これは、両肩のジョイントをY軸まわりでそれぞれ自由回転とし、それぞれの肩においてY軸まわりのモーメントの和が0になる（ZMJ : Zero Moment Joint と呼ぶ）として運動方程式を定式化する。この条件のもとに運動方程式を解き両肩関節の動きを求める。基本になる運動方程式は式(2.1)と同じであり、両肩の関節において運動方程式を作成する。式(2.8)に関して言えば、これまで与えられていた $\beta_{4L/R}$ も未知量（ $\phi_{L/R}$ ）となり、補償動作に関するものと合わせて未知量 $\{E_x\}$ は θ 、 ψ 、 ϕ_L 、 ϕ_R 、の4つになり、既知のモーメント $\{M_0\}$ はこれまでの2つに加え、両肩でのY軸まわりのモーメントの和が0になるという2つが加わり4つとなる。よって、式(2.8)の運動方程式を整理すると、ZMP、ZMJにおけるモーメントの和は式(2.32)のように表され（腕可動（能動型）モデルの式(2.16)に相当する）、これより補償動作及び両肩の関節角を計算することができる。

$$\begin{bmatrix} M_{XZMP} \\ M_{YZMP} \\ M_{YZMJ-L} \\ M_{YZMJ-R} \end{bmatrix} = [E] \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \\ \ddot{\phi}_L \\ \ddot{\phi}_R \end{bmatrix} + [F] \begin{bmatrix} \dot{\theta}^2 \\ \dot{\psi}^2 \\ \dot{\phi}_L^2 \\ \dot{\phi}_R^2 \end{bmatrix}$$

$$+ [H] \begin{bmatrix} \dot{\theta} \dot{\psi} \\ \dot{\theta} \dot{\phi}_L \\ \dot{\theta} \dot{\phi}_R \\ \dot{\psi} \dot{\phi}_L \\ \dot{\psi} \dot{\phi}_R \\ \dot{\phi}_L \dot{\phi}_R \end{bmatrix} + [K]$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.32)$$

ϕ_L : 左肩関節角度 (Y軸まわり)、2.4節の β_{4L}

ϕ_R : 右肩関節角度 (Y軸まわり)、2.4節の β_{4R}

E : 4×4 行列、F : 4×4 行列

H : 4×6 行列、K : 4×1 行列

但し、定常歩行を対象としているため周期解になるものを求める。周期解を得るための初期値修正に関しては、2.4.4節と同様であるが、DEP及び η を以下のように変更しDE、 ∇ DE、 ϕ についてはDEPに合わせて拡張する。

$$\text{DEP} = [\theta, \phi, \dot{\theta}, \dot{\phi}, \phi_L, \phi_R, \dot{\phi}_L, \dot{\phi}_R]^T$$

$$\eta = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.33)$$

これ以外の数値計算の部分については腕可動 (能動型) モデルと全く同様にして解析できる。

2. 6 発進・停止歩行動作の計算法

2. 6. 1 シミュレーション手法

はじめに取り上げたシミュレーションはこれまでの定常歩行解析プログラムから、定常歩行になる補償動作（上体のY軸まわりの回転 ϕ と両脚のX軸まわりの回転 θ ）の周期解を得るための、補償動作の入力初期値修正部分を削除し、入力初期値に対する1歩後の値（以後、最終値と呼ぶ）を計算するものである。

この場合も腕可動（能動型）モデルと同じく、基本歩行パターンを入力する。定常歩行を始めに規定しこれに繋がる発進・停止動作（以後過渡歩行と呼ぶ）を生成する場合、この過渡歩行に対する端末値（発進動作の場合は最終値、停止動作の場合は初期値）は既に規定されたことになる。この周期解に接続する条件を満たす過渡歩行パターンを求める。例えば、発進動作の場合初期値とZMP軌跡を決定するが、現在のプログラムによるシミュレーションでは、数値計算法が不備なためこれらの決定は試行錯誤的に行っている。推定初期値が不相当である時、微調整にZMP軌跡を変えることもできるが周期解に接続するように調整することは難しく、歩容まで取り上げるとなお困難になる。このZMP軌跡を変化させた場合の補償動作に与える影響は、これまでのシミュレーション結果から定性的に分かっている。例えば「ZMPを前にシフトすればそれに伴い上体の補償動作が前に倒れる」。しかし、どの時点でZMPをどれだけ変化させれば上体の動きにどれだけ影響するのかを陽に示すことは難しい。これは、運動方程式が非線型であり数値的には解けるが、解析的には解けないためである。

この手法からも原理的には過渡歩行が決定されるはずであるが、前述のような思考錯誤に時間がかかり周期解に接続する安定な過渡歩行パターンは得られていない。このため、本来なら、初期値、ZMP軌跡、歩容の修正を統合した数値計算によるシミュレーション手法が望まれるが、今回は便宜的に以下に示すような手法を開発し、過渡歩行のシミュレーションを行っている。

ここでは解析された定常歩行を基に過渡歩行をシミュレーションする新たな方法を開発した。発進・停止動作をスムーズに行わせるため、完全歩行を全ての関節角について角度・角速度・角加速度を零（静止直立状態）からスタートし、零で終わるように仮定すれば（ここではこのように仮定したが一般性は失われない）、定常歩行の解析結果と合わせて

過渡歩行での初期値・最終値は決まる。そのため、この境界条件を満たすようなZMP軌跡を作成するよりも、与えられた過渡歩行での初期値・最終値を満足する補償動作を先に作成し、この時のZMP軌跡が安定領域内に入るように補償動作を修正する方が容易である（この場合、本来補償動作と呼ぶべきではないかもしれないが、ここではこれまでの呼び方にならない補償動作と呼ぶことにする）。

2.6.2 ZMPの算出

ZMPにおけるモーメントの総和は式(2.1)で示した通りであるが、絶対座標系においてZMPを算出する場合、原点を床面上に規定し（ここでは支持脚（左脚）の重心真下）ZMPをここから Δr ($X_{ZMP}, Y_{ZMP}, 0$) ずれた点とし、この点におけるモーメントの総和を零としてZMPを導く。すなわち、ZMPにおけるモーメントの総和は式(2.34)のように表し直せる。

$$\begin{aligned}
 M = & \sum_{i=1}^n (r_i - \Delta r) \times m_i G \\
 & + \sum_{i=1}^n (r_i - \Delta r) \times m_i \frac{d^2 r_i}{dt^2} \\
 & + \sum_{i=1}^n \frac{d(J_i \omega_i)}{dt}
 \end{aligned} \tag{2.34}$$

式(2.14)を用いて式(2.34)をスカラーで表し、MのX及びY成分を零として X_{ZMP} ・ Y_{ZMP} について解くと式(2.35)のようになる。

$$\begin{aligned}
 X_{ZMP} = & \frac{\sum m_i X_i (\ddot{Z}_i + g) - \sum m_i Z_i \ddot{X}_i - \sum M_{RYi}}{\sum m_i (\ddot{Z}_i + g)} \\
 Y_{ZMP} = & \frac{\sum m_i Y_i (\ddot{Z}_i + g) - \sum m_i Z_i \ddot{Y}_i + \sum M_{RXi}}{\sum m_i (\ddot{Z}_i + g)}
 \end{aligned} \tag{2.35}$$

但し、 $\sum_{i=1}^n$ を Σ と略記する。

ここで、各ボディの重心座標は式(2.15)の ΔX 及び ΔY を共に0としたものになる。また、他のパラメータについてはこれまでと同じである。

2. 6. 3 補償動作の規定

過渡歩行における補償動作を規定する条件は、与えられた両端での端末値である。これは定常歩行の時のそれに倣い境界での角度・角速度を一致をさせるものである。そのため初期値・最終値合わせて4つの条件を満足する動作を作成するが、スプライン関数での作成や三角関数の組み合わせによる作成にはサンプル点が少なく適当でないため、ここでは高次関数で動作を規定する。4つの条件からは3次関数を規定できるが、これでは動作が一意に決定されるためZMP軌跡が安定領域を外れた場合、補償動作の修正が行えない。そこで、安定領域に入るという条件を考慮すれば4次関数で作成するのが適当である。しかし、この場合発進での初期角加速度（停止での最終角加速度）が必ずしも零になるとは限らない。ここではスムーズな過渡歩行パターンを得るために初期角加速度（停止での最終角加速度）も零と仮定しているため、この条件も加味し6つの条件から5次関数で補償動作を作成する。また、これらの過渡歩行の歩行時間は任意に設定できるようにする。

補償動作の作成は以下に示す5次関数式から連立方程式を作成し、解を得ている。

$$\begin{aligned}\theta(t) &= a_i t^5 + b_i t^4 + c_i t^3 + d_i t^2 + e_i t + f_i \\ \frac{d\theta(t)}{dt} &= 5a_i t^4 + 4b_i t^3 + 3c_i t^2 + 2d_i t + e_i \\ \frac{d^2\theta(t)}{dt^2} &= 20a_i t^3 + 12b_i t^2 + 6c_i t + 2d_i\end{aligned}\quad (2.36)$$

θ に関して上記のように定義すると、完全歩行を全ての関節角について角度・角速度・角加速度を零からスタートし、零で終わるように仮定しているため、 d_i 、 e_i 、 f_i は一意に決定される。残りの3つの係数については、このうち2つは残りの定常歩行との接続のための境界条件より決定されるが、他の1つは安定領域内に入るように規定するためのものであり、一意には決まらない。そのため、 a_i 、 b_i 、 c_i のどれか1つの係数を入力

値として補償動作を作成し、このときのZMP軌跡を計算して、これが安定領域を外れていれば、この入力係数を修正することで安定領域に入る補償動作を作成する。この入力値の指定はシミュレーションプログラム上で任意に行えるが、基本的にはどの係数を入力値としても同じである。

この a_i 、 b_i 、 c_i の係数の決定については、本シミュレーションプログラムではまず大雑把な係数サーベイを行い、安定領域の近傍にある係数を特定し、次にその係数が含まれる範囲をステップを小さくしてサーベイをする、というプロセスを3～4回繰り返すことによって解を得た。実際この方法はやや原始的ではあるものの限られたサーベイを行えばよく、定常歩行のシミュレーションの場合のような思考錯誤は少なく、解の存在についても明確である。同様に、 ϕ の動作も規定できる。このシミュレーションプログラムの概略については付録Bに示す。

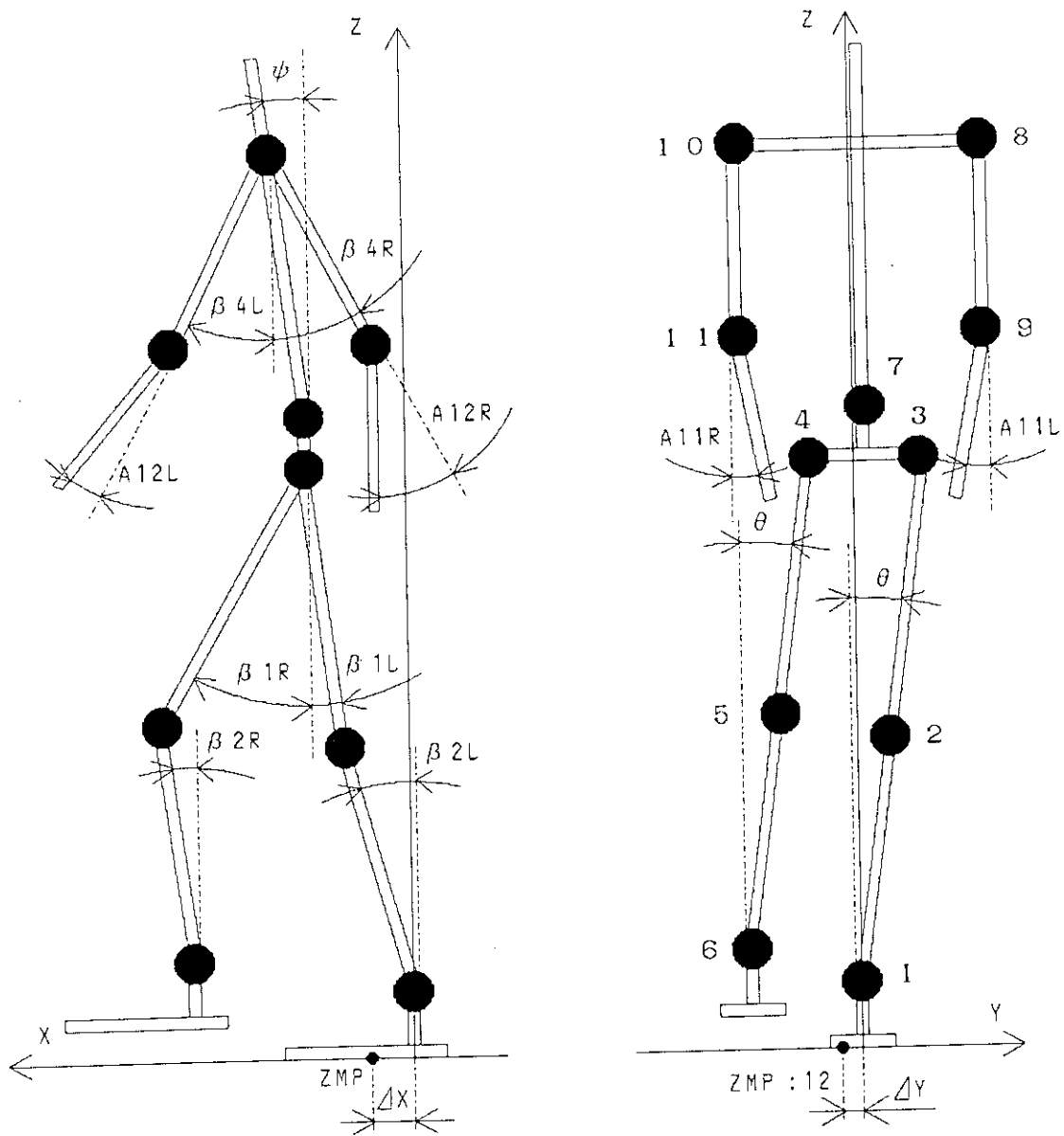
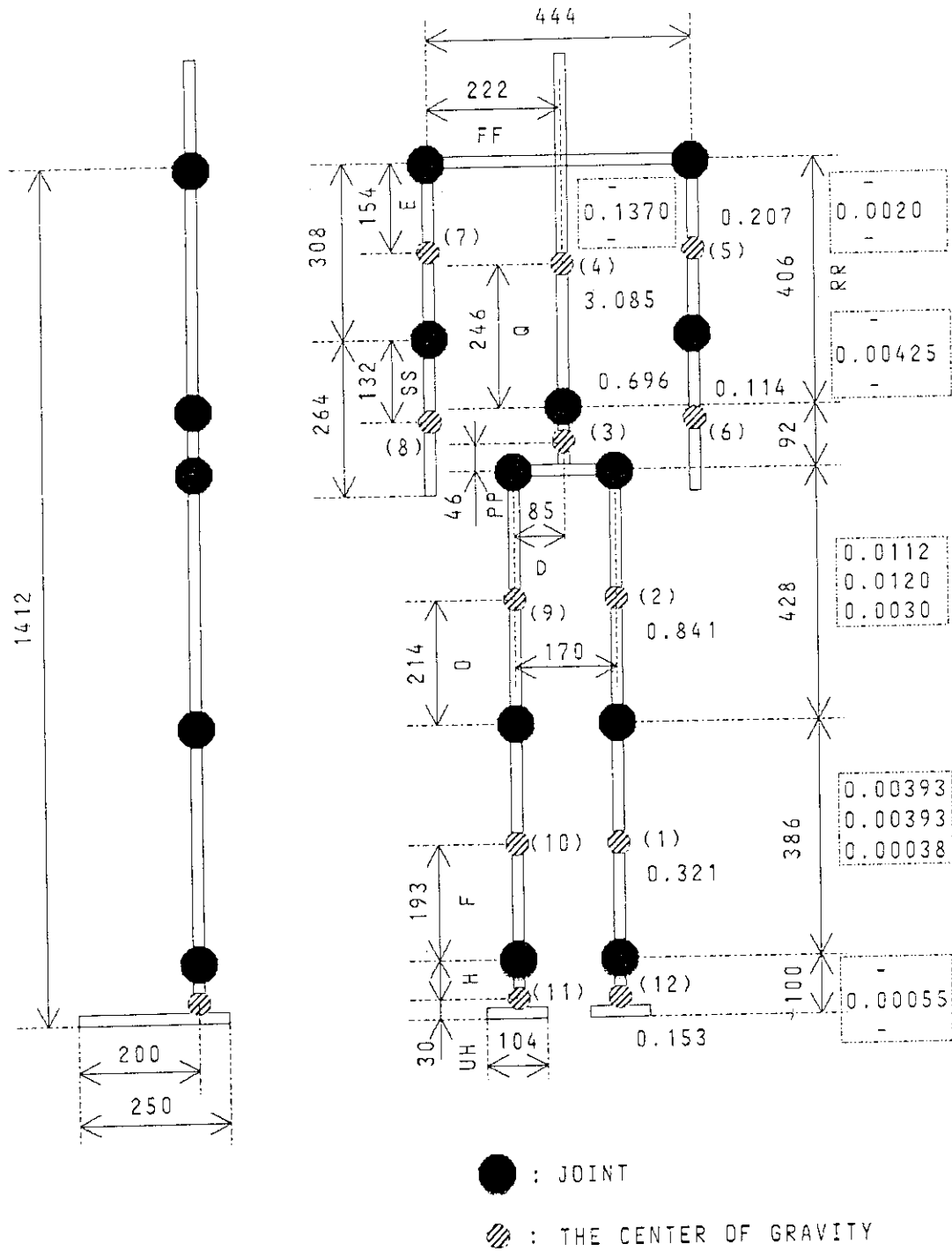


Fig. 2.1 Configuration of robot joints and bodies



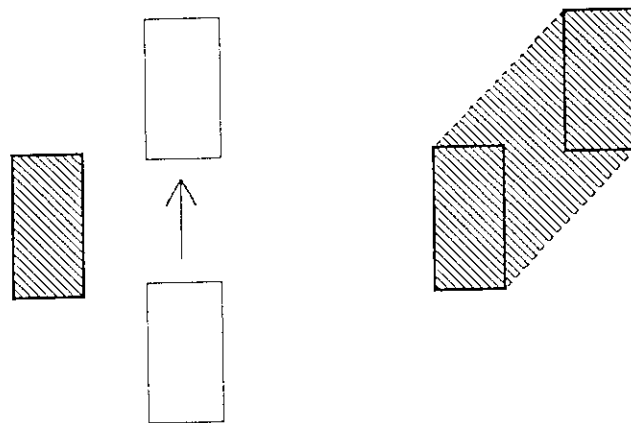
J_x
 J_y
 J_z

: MOMENT OF INERTIA
 [Kg·s²/m·m²]

● : JOINT
 ⊘ : THE CENTER OF GRAVITY
 (1) ~ (12) : BODIES

MASS [Kg·s²/m]

Fig. 2.2 Bodies data



a) 单脚支持期

b) 两脚支持期

□ : 支持脚 (接地) □ : 遊脚 ▨ : 安定領域

Fig. 2.3 Stable region

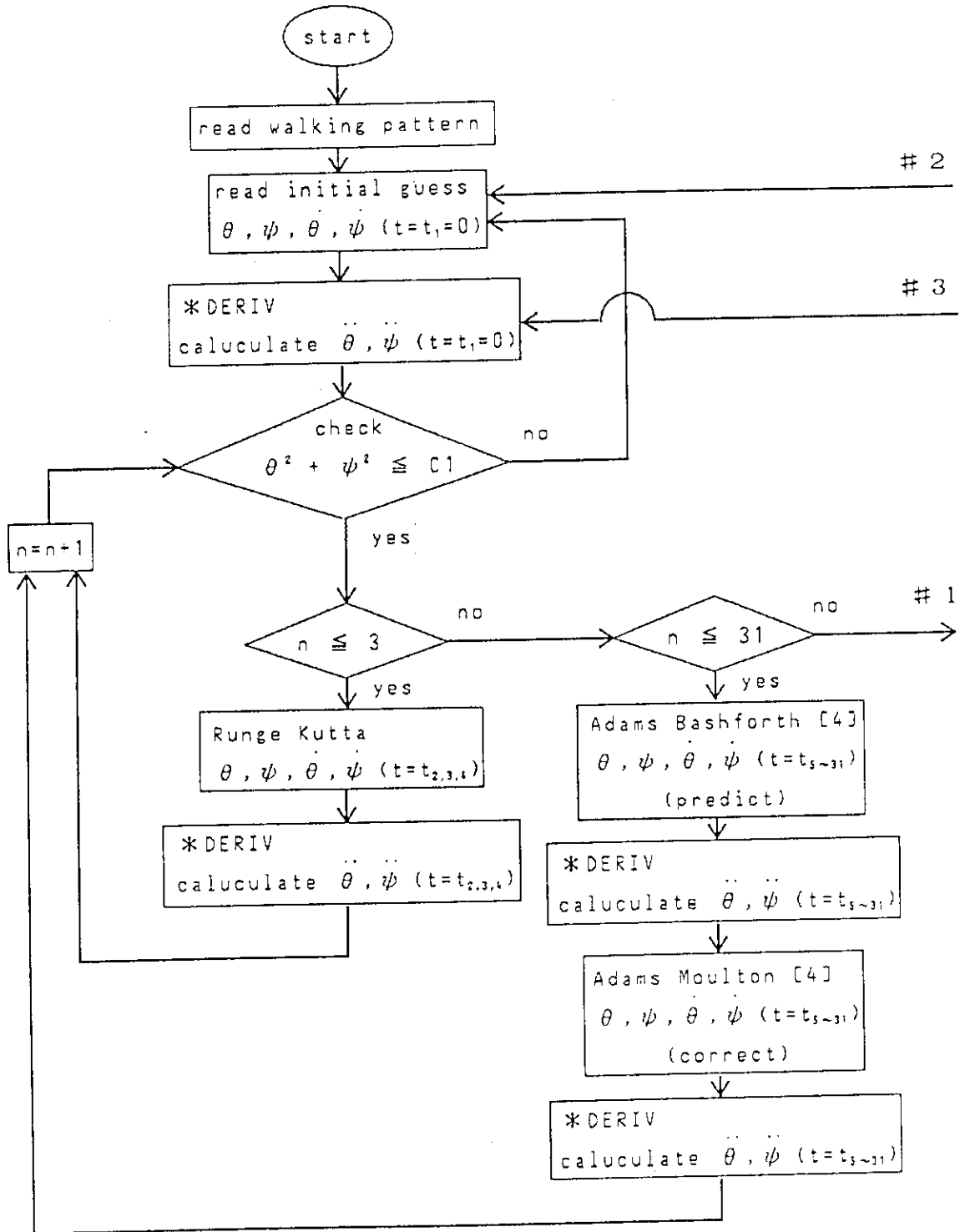


Fig. 2.4 Flow chart for iteration process

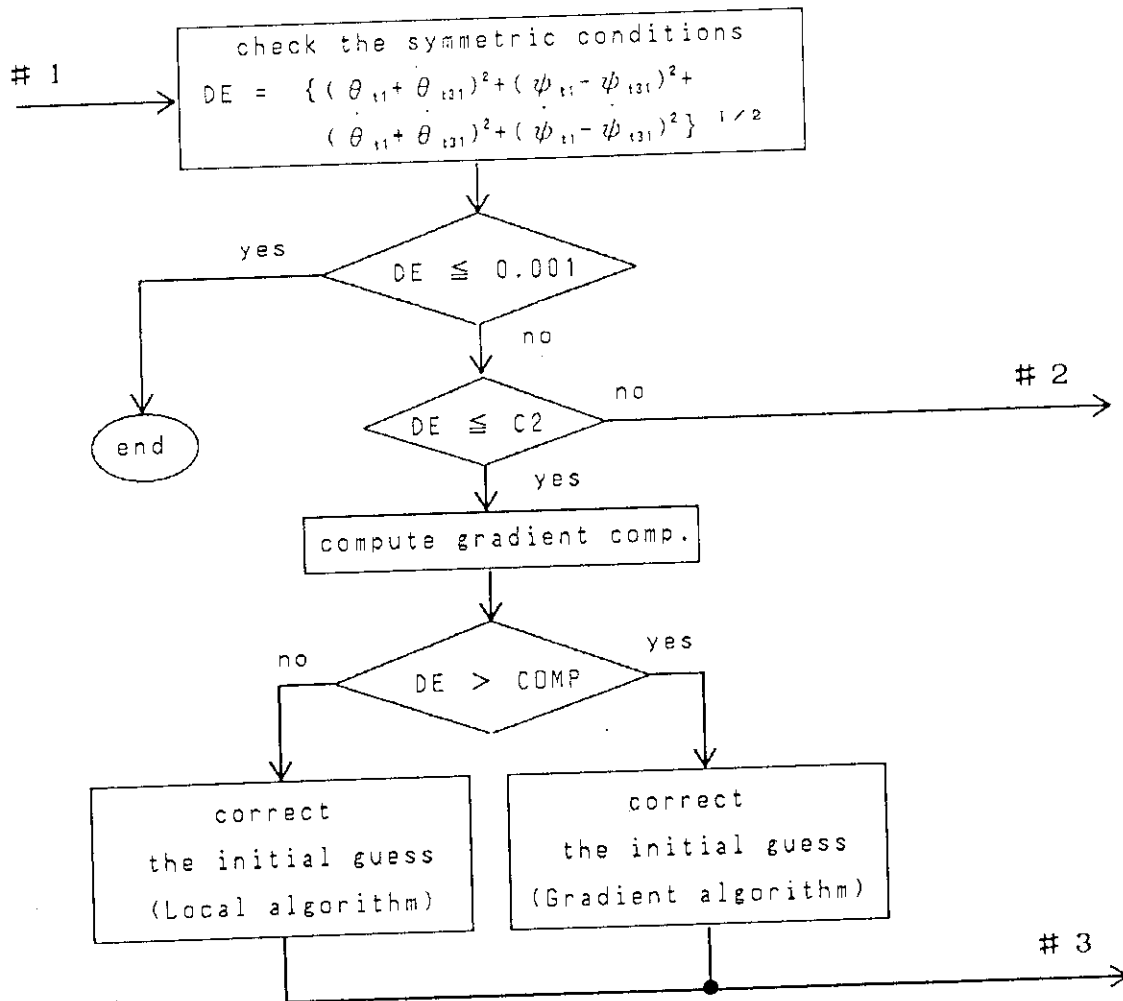


Fig. 2.4 (Continued)

3. シミュレーション結果

3.1 シミュレーションモデルの分類

これまでに開発してきた定常歩行シミュレーション用二足歩行ロボットモデルは Table 3.1 に示すように3つに分類される。ヴコプラトビッチのモデルは、腕固定型で下肢のY軸まわりの関節角及びZMP軌跡をデータとして入力し、そのZMP軌跡に追従するための補償動作として上体(胴)のY軸まわりの回転と両脚のX軸まわりの動きを計算するものであった。2章でも述べたようにこのオリジナルの腕固定型を改良し、腕可動型(肩関節Y軸回り)のモデルを2種類開発した。本章では、これらのモデルを用いて両腕の動き及び質量の違いによる補償動作への影響について考察する。また、腕可動(能動型)モデルを用いた発進・停止動作のシミュレーション結果についても述べる。

3.2 腕固定型

腕可動型との比較のために同一条件の下に腕固定型のシミュレーションを行う。基本歩行パターンは、歩容名H5、入力データ Fig. 3.1 であり、この時のシミュレーション結果を Fig. 3.2 に示す。この歩容を元に上体の質量を変化させ、解析された定常歩行のシミュレーション結果から補償動作($\theta \cdot \phi$)の初期値(角度、角速度)とその振幅及びピーク値を比較する。但し、腕可動型との比較を行うため腕(ボディNo.(5)~(8))の質量を変化させることでシミュレーションを行い、ボディの寸法は変化させない。質量の変更に伴い慣性モーメントの値の変更が必要であり、慣性モーメントは質量に比例し、長さの2乗に比例するため、質量及び慣性モーメントを定数倍してシミュレーションを行う。このため、付録Aに示される入力データの「(2)補償動作作成に関するデータ」に質量、慣性モーメントに関するデータを付け加え、これらのデータ入力型にプログラムを改良する。

Fig. 3.3 は腕の質量をロボットの全質量で正規化した場合(以後、腕の質量比と呼ぶ)で、Fig. 3.4 は上半身(両腕+胴)の質量をロボットの全質量で正規化した場合(以後、上半身の質量比と呼ぶ)である。当然のことながら、補償動作に必要なトルクは変わらないから、質量の増加に伴い振幅及び最大値は緩やかに減少している¹³⁾。線型に変化しないのはボディの慣性モーメントの影響である。また、上体の質量を小さくすればさらに回

転は大きくなり、最終的には物理的に不可能な回転になり周期解は得られないものと思われる。すなわち、補償動作としての機能を持つためにはある程度の質量を必要とするのである。例えば早稲田大学で開発された上体補償型の二足歩行ロボットWL-12R¹²⁾では、上体(腕は有していない)の質量が30 [Kg] 全質量は99 [Kg] である。この場合も上体の質量比からいえばさらに重い上体にしたほうがよいと考えられるが、ロボットのハードウェア的な要素(耐荷重)を考慮すればこのあたりが適当なのであろう。このロボットの全質量と上体の質量とのバランスの問題はロボットの設計の観点からも重要なファクターとなる。

3. 3 腕可動型(受動型)

Fig. 3.5 に腕可動(受動型)モデルのシミュレーション結果を示す。この結果については、基本となる歩容は3.2節と同じもので、質量等はオリジナルのデータを用いておりまだ腕の質量等の変更は行っていない。この図から分かるように腕の動きは上体の動きと同期して、左右同時に同方向に振る動作となっており、腕の動きはほぼ正弦波となっている。この場合、補償動作の1つである上体の動き ϕ は腕固定の場合と同様な動きを見せるが、腕固定の場合よりも動きは大きくなっている。例えば腕固定型の場合、上体が前に傾くのは動的な重心(ZMP)を前方に移動させるためであるが、これが腕可動型(肩自由関節)の場合、上体と同様な動きをする腕(上体が前方に傾く時、慣性力により腕は後方にさがる)により重心が後方に移動してしまう。そのためさらに上体は前方に傾くことになる。これらが互いに影響を及ぼし合って前述のような結果を得ていると考えられる。また、両脚のX軸まわり(左右方向:前頭面)の回転 θ は腕固定の場合と一致しており、腕の動作によるこの補償動作への影響は少ない。これは、腕の動きがY軸まわり(前後方向:矢状面)だけの回転であり、左右同じ動きであるため、前頭面内の動きにあまり影響してこないためである。一般に、矢状面と前頭面の両面間の運動の干渉は無視できるものとする場合が多い¹⁴⁾。

次に、腕固定の場合と同様に腕の質量を変化させてそのときのロボットの動きをシミュレーションする。式(2.32)を解くためには θ 、 ϕ 、 ϕ_L 、 ϕ_R に関する初期角度・角速度を決定しなければならない。これまでも述べてきたように、現時点では定常歩行を取り上げているため、これらに関する条件として周期解になるように境界条件を設けている。2

章でも述べたが、この境界条件を満たす初期値を見つけるために、まず適当な推定初期値を入力してこれを境界条件を満たすように修正を繰り返している。この修正を行うためには、境界となる末端での値（最終値）を知る必要があるが、入力初期値によっては、1歩の計算が終了する前に指数漸近していくものもあり、この推定初期値の選定にも思考錯誤が必要である。これは、入力ZMP軌跡に追従させるために補償動作が指数関数的に大きくなってしまう場合であり、入力された初期値がZMP軌跡が不適當であることを意味する。

ここでは、入力初期値やZMP軌跡に対する自由度が大きいため、シミュレーションを援用して動きの推定を行う。まず、周期解を見つけるための推定初期値を決定する。 θ 、 ϕ 、 $\dot{\theta}$ 、 $\dot{\phi}$ については腕固定の時のシミュレーション結果よりそれぞれ（0.0, 0.0, 0.3, 1.5）とするが、腕については未知である。これまでの腕固定のシミュレーション結果より入力初期値は角度よりも角速度にかなりセンシティブであることが分かっているため、腕の角度については固定し以下のように分類してサーベイを行う。

① $\phi_L, \phi_R = 0.0$ [rad.]

$\dot{\phi}_L, \dot{\phi}_R$ を等しい初期値で与える、

与える範囲は $-10 \sim 10$ [rad./sec.] step 1

② $\phi_L, \phi_R = 0.0$ [rad.]

$\dot{\phi}_L, \dot{\phi}_R$ を異なる初期値で与える、

与える範囲は①での結果を参考に $-6 \sim 6$ [rad./sec.] step 1

（この場合組み合わせは 13^2 通りになるため計算機のCPU時間を考慮して3つの部分に分けて行う）

①で初期値サーベイをした結果を Fig. 3.6 に示す。これで得られた解は1つのパターンに対しそれぞれ1つであり（質量比60%の場合は2つある；後述）、全て左右の腕の動きは一致している。腕の質量を大きくするにつれ、腕固定型とは逆に初期角速度は腕、上体の補償動作とも次第に大きくなる（タイプAと呼ぶ）。また、両脚の補償動作は腕固定の場合とほとんど変わりはなく、質量増加とともに僅かではあるが初期角速度及びピーク値は減少しており、質量増加の効果が現れている。すなわち、前述のように矢状面内の腕の動きによる前頭面内への影響は少ないと言うことがここでも確認できる。しかし、腕の質量比が60%を越えると上体と腕の動きが、今までの周期と反対（振幅の符号が逆）になる（タイプBと呼ぶ）。これは、補償動作としての役割を持つ上体（胴）より腕の質量が

大きくなり、腕の動きが支配的になるためと思われる。この場合も、両脚の補償動作は腕固定の場合とほとんど変わりはなく、腕の動きによる影響は少ない。なお、タイプAとタイプBの切り換え期と思われる60%の場合にのみタイプAとタイプBの両方の解が存在している。

タイプAでは腕・上体の初期角度は40%近傍でピークを見せその後減少傾向を見せるが、それらの初期角速度は増加傾向にあり、60%を越えてからは周期解は得られていない。この結果からは腕固定のものと比較して、腕可動（自由回転）にすることは、腕の質量を増加させても補償動作が逆に大きくなりシステムに対しネガティブに作用すると考えられる。しかし、タイプBでは60%近傍から解が得られて急激な増加を見せ、腕の質量比が70%（上半身の質量比約85%）を越えたあたりから一定値に漸近しつつあるが、ここまで上半身の質量を大きくすることは耐荷重の面から現実的は無理があると思われる。これら2つのタイプに共通していえることは、初期角速度の絶対値が大きくなり過ぎると安定領域内に規定したZMP軌跡に追従できなくなり、この範囲では周期解が得られていないということである。また、Fig. 3.7 はこれらの場合における振幅の比較である。ほぼ正弦波的な動きをしている ϕ 、 ϕ_L 、 ϕ_R であるが、タイプAの場合質量増加に伴い振幅は常に増加しており、タイプBの場合には質量減少に伴い振幅の増加が見られる。しかし、共に腕の振りの角度の最大値が $\pi/2$ rad. 程度までであり、これを越えた周期解は得られていない。これは、両腕が肩関節を支点に振子のように動くためこのような結果を得ているのである。

次に、②で初期値サーベイを行った結果を Fig. 3.8 に示す。②の範囲は①の部分を含んでいるため①で得られたもの以外に解の存在するものについて表示する。この場合は、①と異なり1つのパターンに対し1~4個の周期解が得られている。また、①で得られた腕の動きが左右同じであったタイプAとは異なり、腕の質量比が約50%を越えたあたりから左右の腕の動きが異なる解が得られている。これは、腕の動く方向は同じであるが左右の腕の振りの大きさが異なっており、右腕の振りが大きい場合と左腕の振りが大きい場合の2通り存在し、ほぼ左右を入れ換えたような値を示している（タイプC）。このため、図は右腕の振りが大きい時のみを示す。これは、質量増加に伴い腕の初期値（角度、角速度）は腕の振りが大きい方は増加し、振りの小さい方は減少しており、上体の初期値はほとんど変化していない。また、両脚の補償動作は腕の動きが左右異なるためにこれまでのような結果を示さず、質量増加に伴い初期角度についてはあまり変化はないが、初期角速

度については右腕の振りが大きい場合は①の場合よりも大きく減少し、左腕の振りが大きい場合は逆に増加している。このため、腕の左右の振り角を方向は同じで支持脚側だけ腕の角度を小さくすれば両脚の補償動作を小さくすることもできる。しかし、腕の質量比が50%以上の時であり、僅かな差でしかなく定量的な評価は行っていない。これとは別に、①でのタイプBと同様な腕・上体の動きをするが反対の傾向を持つ解も得られている（タイプD）。この場合、腕の質量増加に伴い角速度（マイナス側に）が腕・上体共に急激に増加し、60～67%の範囲でしか解は得られていない。このときの両脚の補償動作は①での結果とほぼ一致している。これらは全て50%近傍からの解であり、それ以下ではこのような解は得られていない。さらに、タイプCの場合は見方によってはタイプAとほぼ連続であるとも考えられる。

Table 3.2 にサーベイの結果得られた解の分類の一覧を示す。表でAはタイプAを、BはタイプBを、C_LはタイプCで左の振りが大きい場合を、C_RはタイプCで右の振りが大きい場合を、DはタイプDを表す。なお、実際には表に示した場合だけでなく質量比をもう少し細分化したサーベイも行っている。

上記のように腕可動で受動型の場合は質量増加に伴い一般的に上体の補償動作の振幅は大きくなる。これは、腕固定型に比較して全く逆の結果を示しており、システムに対してネガティブに作用している。この結果から、腕を受動的に振らせることのメリットは無く、腕受動型より固定型の方がこの両者の比較からは望ましいといえる。

3.4 腕可動型（能動型）

3.3節の場合とは別に、腕固定型モデルを改良し腕の動きをデータで与える腕可動型を開発した。この腕の動きをも含めて、ZMPにおける運動方程式から補償動作（ θ 、 ϕ ）を決定する。但し、ここでは腕の質量等の変化による補償動作の影響については考察しない。

腕の動きとして人間の振りを模倣した場合（脚の動きに同期して交互に動かす）は、補償動作である上体の動き（ ϕ ）に与える影響は非常に少ない（Table 3.3）。これは腕の質量及び慣性モーメントが小さいうえに左右対称な動きであるために、腕の振りによるモーメントがロボットシステム全体に与える影響が少ないためである。しかし、3.3節で述べたように腕可動の受動型の場合に腕と上体が同期した動きをしており、腕の動的な重

心が上体のそれと反対に動くために上体の補償動作はさらに大きくなっている。逆にいえば、本節の腕可動の能動型の場合において、予想される補償動作の動きに合わせて両腕を同時に同方向（上体の動きと同じ前後方向）へ振らせれば上体の補償動作を小さくすることができる。Fig. 3.9 はこれに基づき能動型の腕の動作を作成した場合のシミュレーションの結果を腕固定、腕可動（受動型）の2つの場合と上体の補償動作を比較した場合であり、明らかに能動型の場合が上体の動きが小さいことが分かる。このように上体と腕の協調による安定歩行の可能性もある。

3. 5 発進・停止歩行動作シミュレーション

3. 5. 1 過渡歩行シミュレーション第一段階

2.6.3節で補償動作を5次関数で規定すると述べたが、第一段階としては現在のシミュレーションが常に単脚支持状態であることを仮定しているため、静止状態については考えず4次関数で動作を規定した（規定方法は5次関数の場合と同じである）。これは、発進動作であれば $t = 0$ 秒ですぐに動き出すような動作のシミュレーションを行うため、 $t = 0$ 秒での補償動作の角加速度は必ずしも零である必要はなく、4次関数で動作を規定できる。

始めに、1秒間の発進の歩容を作成し、この動作についてシミュレーションを行った。ここでは、安定領域を若干修正し、オリジナルの足底の大きさをX方向 $-0.1 \sim 0.2$ 、Y方向 $-0.08 \sim 0.08$ [m] に変更した（但し、単脚支持期；両脚支持期はY方向が $-0.25 \sim 0.08$ である；以後この範囲を安定領域1と定義する）。この結果からは安定領域に入る動作は作成できなかった。これは、シミュレーション結果の一例（Fig. 3.10）を見ると分かるように発進直後と終了時に大きく安定領域を外れている。これは歩行動作の過渡的な動きによるロボットシステム全体に与えるモーメントが大きく、この補償動作では安定領域に入るように補償ができないのである。この改善には以下のような方法が考えられる。

- ① 静止状態を考慮する。
- ② 歩容（歩行動作）をもっとスムーズにする。
- ③ この結果を基に安定領域を外れた部分だけ局所的に補償動作を変化させる。

①については、単脚支持を考えた場合、発進動作に入る前に発進時に支持脚となる方の安定領域内にZMPが存在していなければならないが、このためには歩行を開始する前に動的な重心(ZMP)を移動させておく必要がある。これには、静止状態を考慮し(歩容及び腕の動作についてのみであり補償動作は別とする)この間に補償動作によって動的な重心を支持脚側に移動させればよい。静止状態を取り入れた発進動作を考えた場合、静止状態は当然のことながら両足が地面に拘束(接地)されているため、改めて両脚接地の拘束条件の検証を行う必要がなく、ZMPの解析についてはZMPの安定領域を単に両脚支持でのそれに拡大することで、両脚支持状態も単脚支持状態と同様にしてシミュレーションすることができる。この場合、発進動作での初期角加速度を零とし、補償動作を5次関数で規定する。また、歩行時間と静止時間との関係からも安定性を考慮できる。

②については、現在のプログラムが歩容の元データである各関節角のサンプル点での値を線型補間することによって歩容のデータを作成しているため、局部的最小二乗近似多項式によるスムージング(次数6次、観測点9個:1秒30分割)を行ってはいらぬものの、サンプル点近傍での動きがどうしても過渡的になりやすい、しかもこのスムージングによりサンプル点での値が若干変化してしまうという欠点もある。この改善には、サンプル点からの歩容の作成にスプライン関数を用いる方法が考えられる。これにより関節の動き全体をスムーズにすることができ、しかもサンプル点での値も保証される。

③については、補償動作を規定した時のZMP軌跡を算出しこれが安定領域を外れている場合、安定領域を外れた点の近傍の補償動作を変化させ安定領域に入るように再構成する。例えばZMPが後方に安定領域を外れていれば補償動作を前方に傾けるというような操作を行うことになるが、これは2.6.1節でも述べたように補償動作とZMPの関係であり、この関係を明らかにすることは難しく、端点での修正は境界条件を満足出来なくなる恐れがありあまり適当な方法ではない。よって、次節で①及び②について検討する。

3.5.2 過渡歩行シミュレーション第二段階

はじめに3.5.1節で述べた解決策①を施す。3.5.1節で用いた発進動作の前に静止状態を挿入した歩容を作成し、この静止時間を0, 0.5, 1.0, 1.5, 2.0秒と変化させた時のZMP軌跡との関わりを見る。Table 3.4は入力係数を b_T , b_P (ϕ の4次の係数)とし、係数サーベイの範囲をそれぞれ-100~100までステップ1で補償動作を作成した場合の、

安定領域内に入る歩行動作の個数を示したものである。Table 3.4 から分かるように、本来の安定領域（安定領域1）に入る歩行動作は作成出来ていない。しかし、静止状態を導入することによってY軸方向のZMP (Y_{ZMP}) は安定領域に入るようになり、静止状態を導入した効果が現れている。また、X軸方向のZMP (X_{ZMP}) については静止状態が無い場合よりは安定領域に近くなってきているが、まだ発進直後に安定領域を外れている（例えば Fig. 3.11 は静止時間 1.0 秒の場合）。

この歩容の場合ZMPの初期位置は (0.0006, -0.0850, 0.0) であり（原点は左足足首真下の足底、Fig. 2.1 参照）、この位置は当然両足の中央である。但し、この歩容は Y_{ZMP} については両足の中央であるが、ロボットの肘を曲げた (Y軸まわり -0.15 rad.) 歩容であるために X_{ZMP} は 0.0 ではない。このとき、静止状態が無い場合には、 Y_{ZMP} を支持脚側の安定領域内へ発進直後に移動させなければならず、かなり大きなトルクを要する補償動作が必要となり、安定領域内に移動させることは難しい。しかし、このことは静止状態を導入することで解決できる。特に前述のように、このY軸方向のZMPを安定領域に移動させるためには静止状態の導入が有効な手段であるため、静止状態を考慮した歩容に対してさらに次の解決策②についての検討を行う。その前に、静止時間を特定するため Table 3.4 の安定領域2について考える。これは安定領域1を拡大した場合であり、この場合は幾つかの補償動作が作成できている。但し、この大きさの足底をしたロボットについては考えない。この結果から静止状態の時間を1秒、発進の1歩1秒合計2秒の歩容として次の考察を行う。

解決策②については歩容の各サンプル点における値をスプライン関数 (B-Spline) によって補間することで全体の歩容を作成する。スプライン関数の次数は境界条件 (角速度、角加速度) を指定するため5次とした。これに伴い定常歩行のシミュレーションプログラムも歩容の補間をスプライン関数で行うように修正を加えた。定常歩行の場合は周期関数として歩容の補間を行い、スプライン関数の次数はサンプル点の個数を考慮して3次とした。また、元になる定常歩行の歩容に、肘を伸ばしたままの歩容にする等の若干の変更を加え、新たに定常歩行のシミュレーションを行って (歩容名H2、Fig. 3.12(a)~(d) は基本歩行パターン、(e) はシミュレーション結果)、そのシミュレーション結果に基づいた発進の歩容を作成し (Fig. 3.13)、発進動作の解析を行った (サーベイの係数は c_T , c_P とした)。このスプライン関数による補間の効果は良好で Table 3.5 に示すような結果を得、安定な発進歩行が作成できた。

Table 3.5 からも分かるように係数の決定には任意性があり、安定領域に入る補償動作がいくつか作成できる。これはサーベイのステップにも依存するが、ここでは有効数字を考慮して最小ステップを 0.001 とする。1、2 回目のサーベイは安定領域に入る補償動作の個数が少なくこれらの結果全てを比較することができるが、3 回目以降になるとそれは不可能である。そのため、この数多くの解の中から最適解を選択するための何らかの評価関数がどうしても必要になる。しかし、これまでも述べたように明確な評価関数が定義できないため、ここでは取り敢えず ZMP の安定領域について考察し、安定マージンが最大になる場合を最適解とした。言い換えれば、不慮の外乱に対応出来るように、ZMP が常に安定領域の中心に位置するような歩行パターンが適当であるとして、

$$| (\text{ZMP}) - (\text{Stable Region})_{\text{center}} | \quad (3.1)$$

を最小にする場合を最適解とした。但し、両脚支持期（静止状態）においてもこの評価を行うことは適当ではない。なぜなら、両脚支持期から単脚支持期に切り替わるまでに ZMP は単脚支持期の支持脚側の安定領域内に移動していなければならないため、このことは式(3.1)を最小にすることと相反するからである。このため、この評価を行うのは単脚支持期だけとした。なお、両脚支持期における Y_{ZMP} の範囲は常に $-0.25 \sim 0.08$ とはせず、下限については出来るだけ大きくとることとし、最終的には Y_{ZMP} の安定領域の範囲を $-0.10 \sim 0.08$ として解析を行った。また、足底で形成される安定領域は X 方向、Y 方向で長さが一致していないため、これらの長さで正規化した。すなわち、

$$SZ = \sum_{t=1}^{t_2} \frac{|X_{\text{ZMP}} - X_{\text{SRC}}|}{L_x} + \sum_{t=1}^{t_2} \frac{|Y_{\text{ZMP}} - Y_{\text{SRC}}|}{L_y} \quad (3.2)$$

$(X_{\text{SRC}}, Y_{\text{SRC}}, 0.0)$: 安定領域の中心点

L_x, L_y : 安定領域の X 方向・Y 方向の長さ

$t_1 \sim t_2$: 単脚支持期

を最小にする補償動作を作成した。Table 3.5 には示していないが、3 回目のサーベイからこの評価を取り入れ、この結果からこの近傍についてさらに最終ステップのサーベイとして、 c_T については $-0.155 \sim -0.135$ までステップ 0.001 で、 c_P については $1.53 \sim$

1.55 までステップ 0.001 で行った。Fig. 3.14 は式(3.2)に基づいて決定した発進動作のシミュレーション結果である。

同様な方法で停止動作のシミュレーションを行う。定常歩行(H2)に繋がる停止動作を1秒で作成し、この停止動作の後に静止状態を繋げる。静止時間を0, 0.2, 0.5, 1.0, 1.5, 2.0秒と変化させZMP軌跡との関わりを見る。Table 3.6 は入力係数を c_r , c_p とし、係数サーベいの範囲をそれぞれ-100~100までステップ1で補償動作を作成した場合の、安定領域内に入る歩行動作の個数を示したものである。なお、歩容の補間はスプライン関数で行っている。Table 3.6 から分かるように、発進動作の場合と違って安定な歩行が作成出来ていない。これは、定常歩行において動作中に直立状態に近くなる時があるが、この状態になるまでの時間は支持脚切り換え時から約0.3秒後であり(1歩1秒)、停止に至るまでの歩行速度がこれまでの歩容とかなり異なるためであると推測できる(但し、境界条件は満足している)。しかし、はじめに作成した停止の歩容はこれを1秒の動作として作成している。また、補償動作との調和がうまく取れていないためでもある。今回は、補償動作を5次関数で作成していることから、これは高次関数を用いているための欠点でもあるが、補償動作を柔軟に変化させることが出来ない。このため、ある程度は歩容に依存する部分が生じてくるのである。したがって、この停止状態になるまでの時間も変化させることで安定な歩行動作を作成した。Table 3.7 にこの結果を示す。係数サーベいはTable 3.6と同じである。Table 3.7からは停止までの時間を0.4秒、静止時間を0.5秒とするのが良いと思われるが、安定領域1について計算した場合、この組み合わせは安定領域1に入る補償動作は作成できなかった。これはサーベいのステップを小さくしても同じである。これに対して、第二候補の停止までの時間を0.5秒、静止時間を0.5秒とした場合(Fig. 3.15)は安定領域1に入る補償動作は60個作成できた。これより、停止動作の歩容を停止までの時間を0.5秒、静止時間を0.5秒として発進動作と同様に係数サーベいを行い(Table 3.8 参照、最終ステップは c_r : -1.42~-1.40、ステップ0.001、 c_p : 23.93~23.95、ステップ0.001)、式(3.2)に基づいて停止動作を決定した。この決定した停止動作のシミュレーション結果をFig. 3.16に示す。

3.6 シミュレーション結果の映像化による考察

シミュレーション結果の表示は前節までに示した各関節角の動きを示すグラフの他に、

これらの実際の動きを確認するため、専用の映像化システムを用いて動画化することができる。ここで映像化したものは腕固定モデル、腕可動（能動型）モデルのシミュレーション結果^{2, 3)}であり、腕可動（受動型）モデル及び発進・停止動作のシミュレーション結果に関しては、シミュレーション結果のデータを映像化ソフトウェア用の映像化データに変換するためのインタフェースプログラムが未整備であるため未だ行っていない。

シミュレーション結果の映像化は開発当初の SOLVER¹⁵⁾ (A Solid Modeler for Vision Engineering Research) と呼ばれる画像生成ソフトウェアを用いた駒撮り風の図形化から、CAP (Cellular Array Processor)¹⁶⁾を用いたこれらの動画の生成¹⁷⁾へと変更を行った。これは、駒撮り風の画像だけでは実際の動きがよくわからないためである。ここでは、Fig. 3.17 に示すようにロボットを2体使って正面からと側面からの両方の動作チェックができるようにしている。但し、階段等のシミュレーションも行うためロボット初期状態は足が地面に接地しないように設定している。この図はCAPで映像化した一画面のハードコピーである。このロボット躯体モデリングソフトウェア FUSION、この動画作成用プログラム及びシステムの構成については文献18) に詳しいのでこの文献を参照されたい。

始めに、腕固定型モデルを用いた基本歩行パターン（但し、 $\beta_{4L/R}$ は除く）の変更による補償動作への影響について述べる。ここでのシミュレーション結果は全てヴコプラトビッチの入力データに基づいたものであるが、動画化すると上体が非常にギクシャクとした動きをしているうえ歩容自体にも不自然なものが見られる。そこで、この動画を基に基本歩行パターンの修正を行い、上体の動きを滑らかにすることを試みた。この修正には歩容（下肢の関節角）の変更とZMPの軌道の変更が考えられるが、ZMPの軌道の変更に関してはこれまでのいくつかのシミュレーション結果よりZMPを前にシフトすればそれに伴って上体は前に傾く事が分かっている。また、下肢の関節角の変更では冗長自由度を利用して腰の位置を制御する¹⁹⁾ことにより上体の動きを修正することができる。すなわち、足の軌道は骨盤が回転しないためX-Z平面の二次元平面内での動きとして考えられる。一般に二次元平面内で任意の位置に移動させるためには2自由度必要であるが、下肢はX-Z平面で5自由度を持っている。そのため、この冗長自由度を利用して足の位置を変えずに腰の位置だけ変更することができるのである。この場合、腰を前に出せば上体は後ろに傾く。これは倒立振り子の制御と同じ方法である。これらを考慮して歩容の修正を行い比較的滑らかな平地歩行と階段歩行の歩容を作成した。Fig. 3.18 に歩容の修正前と修正

後のシミュレーション結果の比較を示す。図からも明らかに上体の動きが滑らかになっている様子が分かる。

次に腕可動（能動型）モデルを用いて、前述の基本歩行パターンに腕の動きを付加した場合のシミュレーション結果について述べる。このときの腕の振りのデータは左右対称になるように作成し、歩行動作とのタイミングについては人間の振りを模倣している。当初支持脚が切り替わる瞬間に腕の振りが最大になると考えていたが、作成した動画を見るとわずかながらタイミングがずれている様子が分かる。ここで、もう一度人間の歩行動作の観察を行った。通常の歩行では腕の振りは支持脚が切り替わる瞬間に最大になるのではなく、支持脚でない方（遊脚）の大腿の角度（ β_{1R} : Fig. 2.1参照）が最大となる時支持脚側の腕の振り角（ β_{4L} ）が最大となっているようである。このシミュレーション結果を Fig. 3.19 に示す。本来これらのシミュレーション結果は動画として表示しているが、本原稿のため動画のシーンの一部をハードコピーして作成したものである。

なお、上記以外にも骨盤のY軸方向の長さの調整で両脚の左右の振れ θ を、骨盤のZ軸方向の長さの調整で腰の曲がる角度 ϕ を変化させることができることもシミュレーションより分かった。このように、モデルの各部の寸法等の変更で補償動作を変えることもできる。

Table 3.1 Classification of simulation robot model

タイプ	両腕 (肩のY軸回りの回転)	下肢 (Y軸回り)	ZMP	上体・ 下肢(X軸回り)
腕固定	胴体に固定	データ 入力	データ 入力	補償動作としての 動きを決定する
腕可動 (能動型)	データ入力	データ 入力	データ 入力	補償動作としての 動きを決定する
腕可動 (受動型)	肩のジョイントを自由回 転として動きを決定する	データ 入力	データ 入力	補償動作としての 動きを決定する

Table 3.2 Influence of upper body mass to compensating motions

腕の質量 倍数 [倍]	腕の質量比 [%]	上半身の 質量比 [%]	周期解のタイプ	
			サーベイ①	サーベイ②
0.1	0.991	48.63	A	A
0.5	4.77	50.59	A	A
1.0	9.10	52.84	A	A
5.0	33.36	65.43	A	A
10.0	50.04	74.08	A	A
11.0	52.42	75.31	A	A
12.0	58.37	78.40	A	A, C _L , C _R
15.0	60.03	79.27	A, B	A, B, C _L , C _R , D
20.0	66.70	82.73	B	B, C _L , C _R , D
50.0	83.35	91.36	B	B, C _L , C _R

Table 3.3 Comparison of simulated results

時間	θ [rad.]		ϕ [rad.]	
	腕固定	腕可動	腕固定	腕可動
0.0000	-0.0022	-0.0021	0.1783	0.1721
0.0333	-0.0244	-0.0236	0.2138	0.2018
0.0667	-0.0441	-0.0427	0.2321	0.2181
0.1000	-0.0615	-0.0596	0.2440	0.2312
0.1333	-0.0770	-0.0746	0.2550	0.2414
0.1667	-0.0905	-0.0878	0.2631	0.2477
0.2000	-0.1023	-0.0993	0.2630	0.2486
0.2333	-0.1125	-0.1092	0.2562	0.2426
0.2667	-0.1212	-0.1177	0.2445	0.2316
0.3000	-0.1285	-0.1248	0.2306	0.2203
0.3333	-0.1345	-0.1307	0.2170	0.2090
0.3667	-0.1393	-0.1355	0.2040	0.1977
0.4000	-0.1430	-0.1391	0.1918	0.1874
0.4333	-0.1455	-0.1417	0.1816	0.1789
0.4667	-0.1470	-0.1432	0.1730	0.1716
0.5000	-0.1473	-0.1436	0.1605	0.1611
0.5333	-0.1466	-0.1430	0.1394	0.1435
0.5667	-0.1449	-0.1413	0.1156	0.1238
0.6000	-0.1420	-0.1385	0.0871	0.0996
0.6333	-0.1381	-0.1347	0.0576	0.0737
0.6667	-0.1333	-0.1299	0.0282	0.0471
0.7000	-0.1273	-0.1240	-0.0025	0.0195
0.7333	-0.1200	-0.1169	-0.0371	-0.0103
0.7667	-0.1113	-0.1083	-0.0675	-0.0400
0.8000	-0.1009	-0.0982	-0.0759	-0.0525
0.8333	-0.0885	-0.0861	-0.0548	-0.0316
0.8667	-0.0742	-0.0721	-0.0117	0.0099
0.9000	-0.0582	-0.0565	0.0385	0.0507
0.9333	-0.0402	-0.0391	0.0852	0.0897
0.9667	-0.0202	-0.0196	0.1308	0.1311
1.0000	0.0022	0.0021	0.1783	0.1721

Table 3.4 Influence of standstill time to stable walking
[starting motions]

	静 止 時 間 (sec.)				
	0.0	0.5	1.0	1.5	2.0
安定領域 1 に入る場合の個数 X: -0.1~0.2 Y: -0.08~0.08	0	0	0	0	0
安定領域 2 に入る場合の個数 X: -0.2~0.4 Y: -0.08~0.1	0	0	4	2	0

歩容は線型補間

Table 3.5 Number of calculated cases for stable walking
[starting motions]

係数のサーベイ 範囲/ステップ	C _T C _P	-100, 100, 1.0 "	-10, 10, 0.1 "	-0.2, 0.1, 0.005 0.0, 3.3, 0.005
安定領域 1 に入る場合の個数		3 6	6 4	2 2 1 9 0

安定領域 1 : X: -0.1~0.2、Y: -0.08~0.08

Table 3.6 Influence of standstill time to stable walking
[stopping motions]

	静 止 時 間 (sec.)					
	0.0	0.2	0.5	1.0	1.5	2.0
安定領域 1 に入る場合の個数 X: -0.1~0.2 Y: -0.08~0.08	0	0	0	0	0	0
安定領域 2 に入る場合の個数 X: -0.2~0.4 Y: -0.08~0.1	0	0	0	13	0	0

歩容はスプライン関数による補間

Table 3.7 Influence of standstill time and walking time to stable walking
[stopping motions]

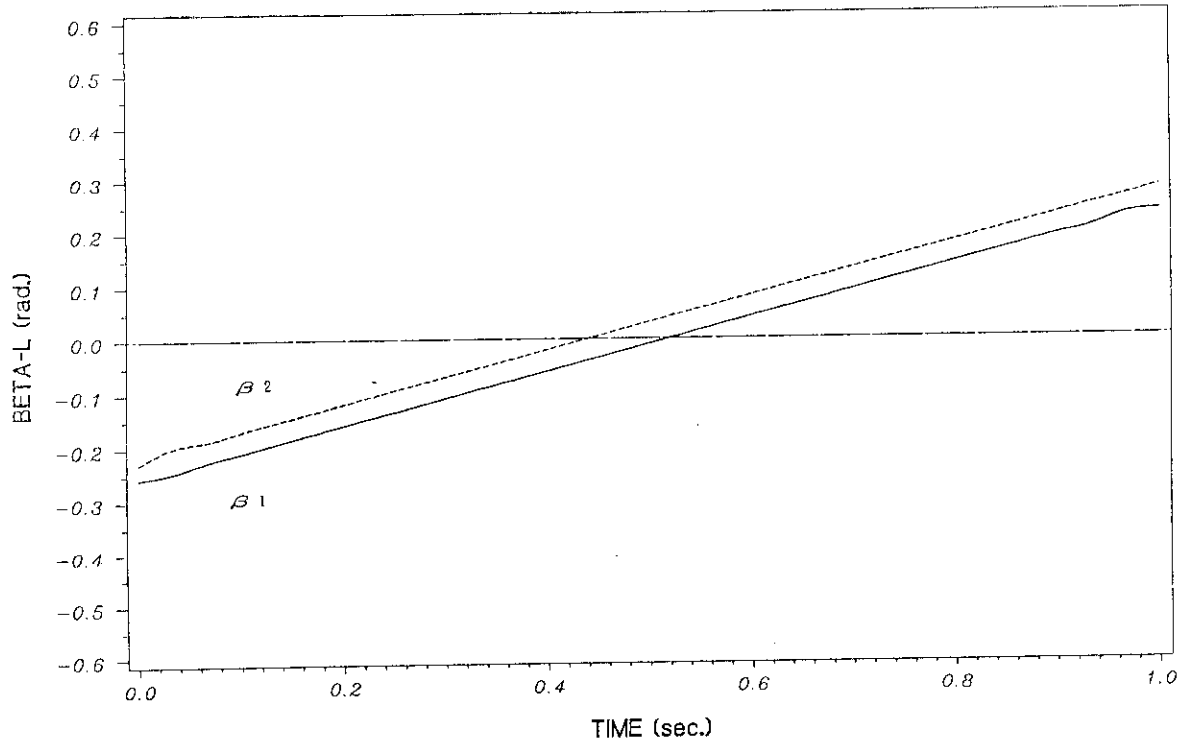
停止までの時間 (sec.)	安定領域 2 に入る場合の個数			
	静 止 時 間 (sec.)			
	0.5	1.0	1.5	*
0.8	66	17	0	0
0.5	141	34	18	—
0.4	164	20	10	81
0.3	0	0	0	0

* 停止動作の合計時間が1.0 秒
安定領域 2 : X:-0.2~0.4、Y:-0.08~0.1

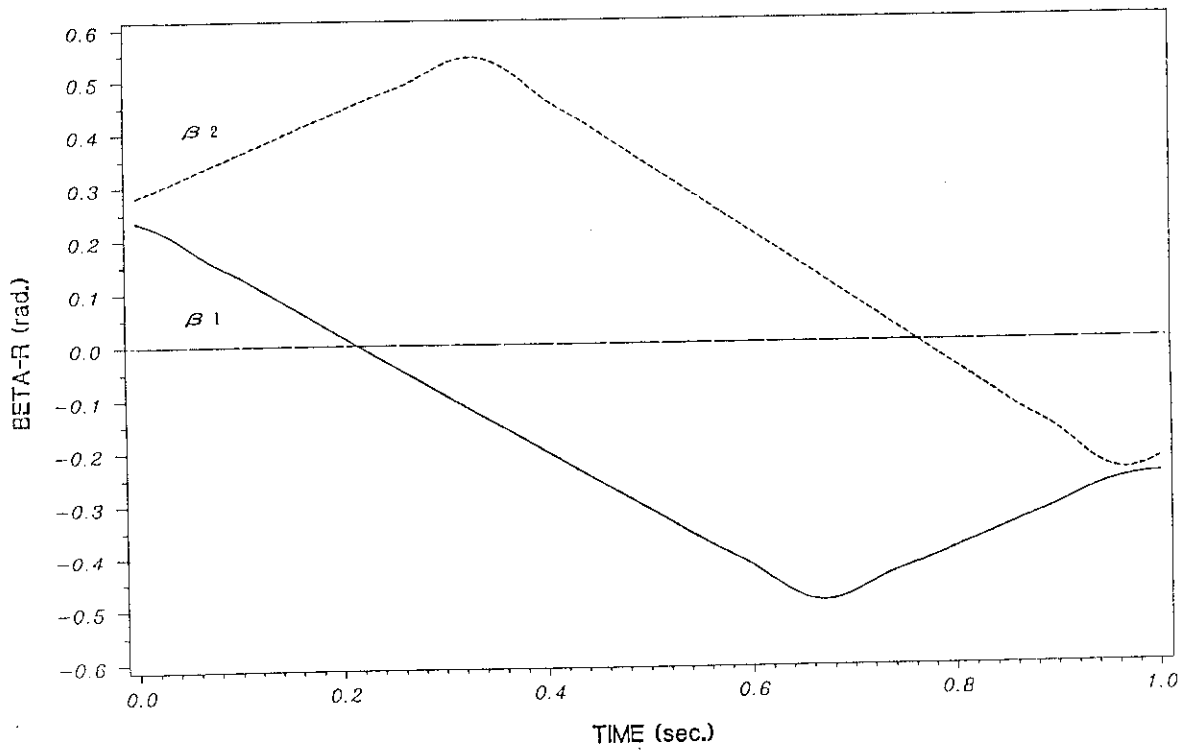
Table 3.8 Number of calculated cases for stable walking
[stopping motions]

係数のサーベイ 範囲/ステップ	C _T C _P	-100, 100, 1.0 "	-2, 1, 0.1 12, 33, 0.1	-1.6, -1.3, 0.005 12.7, 32.5, 0.005
安定領域 1 に入る場合の個数		6 0	1 3 3	7 0 8 1 6

安定領域 1 : X:-0.1~0.2、Y:-0.08~0.08

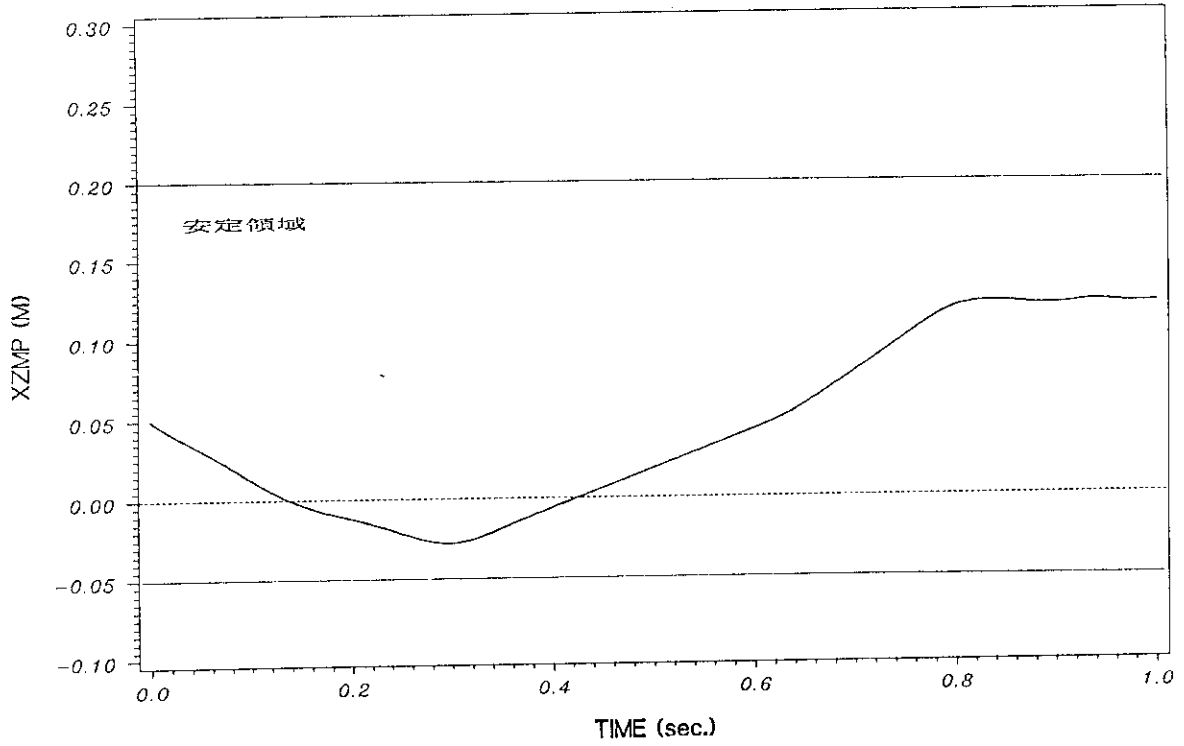


(a) Left leg

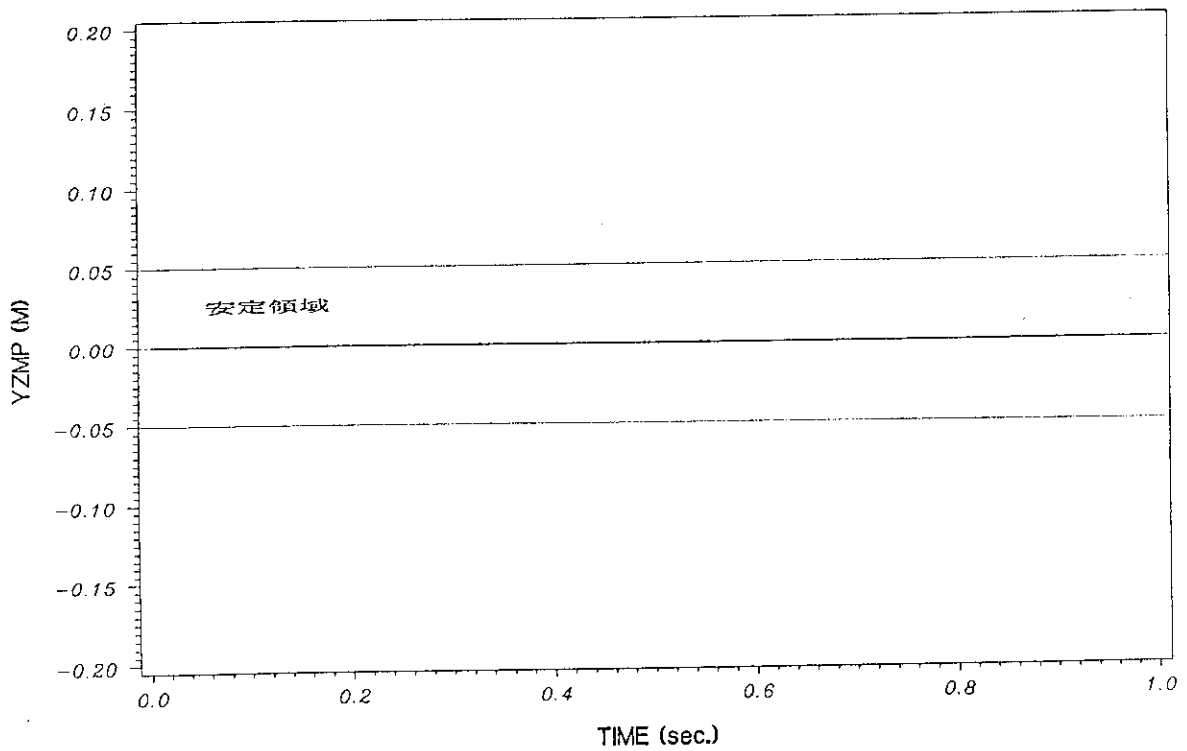


(b) Right leg

Fig. 3.1 Basic walking pattern of "H5"



(c) X_{ZMP}



(d) Y_{ZMP}

Fig. 3.1 (Continued)

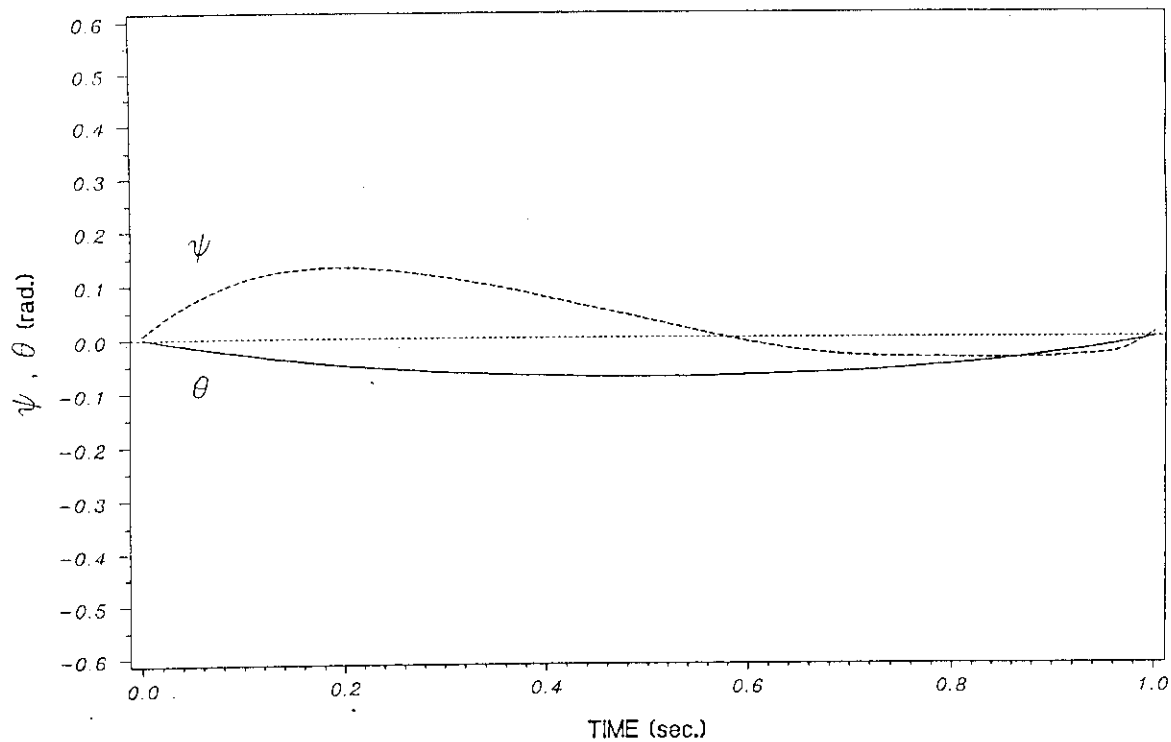


Fig. 3.2 Simulated results

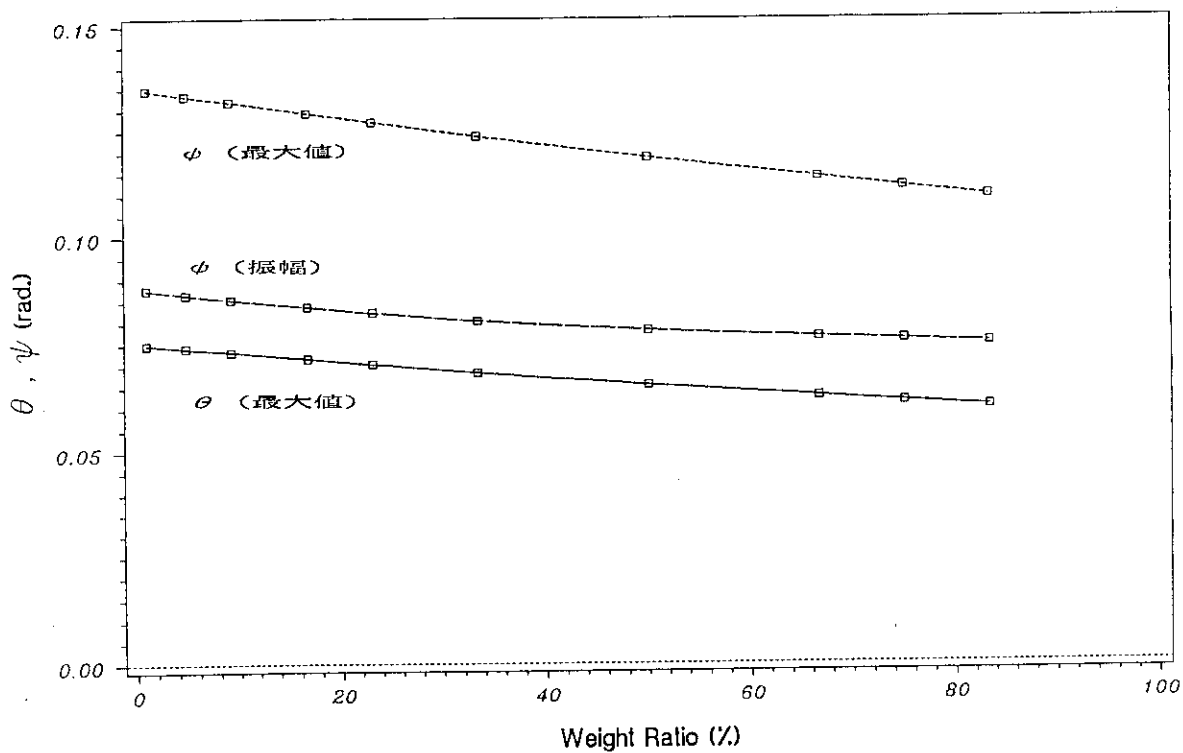


Fig. 3.3 Influence of upper body mass to compensating motions
[arms / bodies]

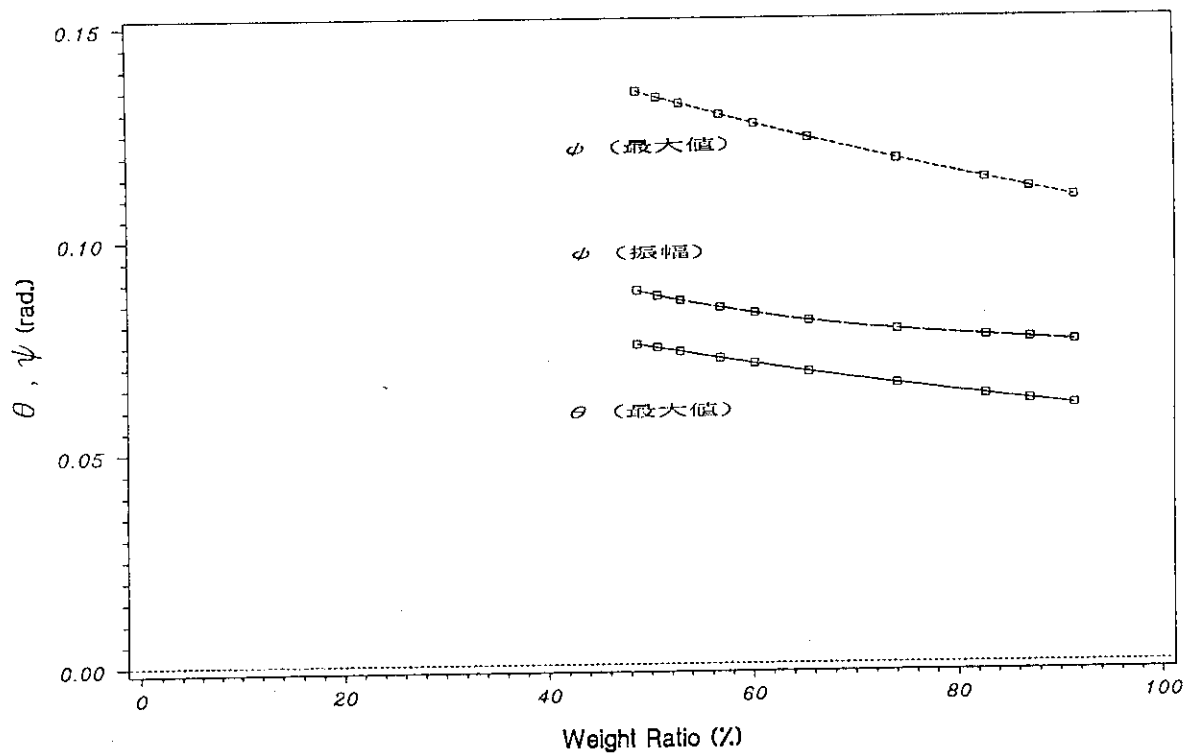


Fig. 3.4 Influence of upper body mass to compensating motions
 [(arms + trunk) / bodies]

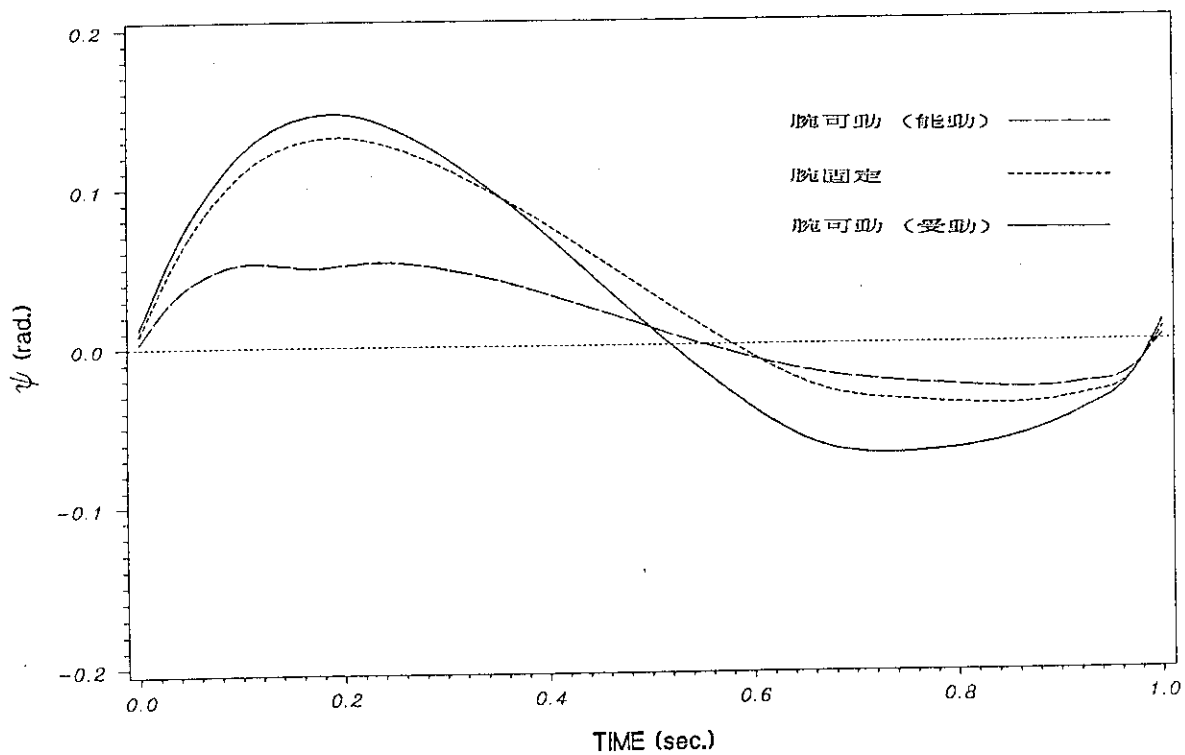
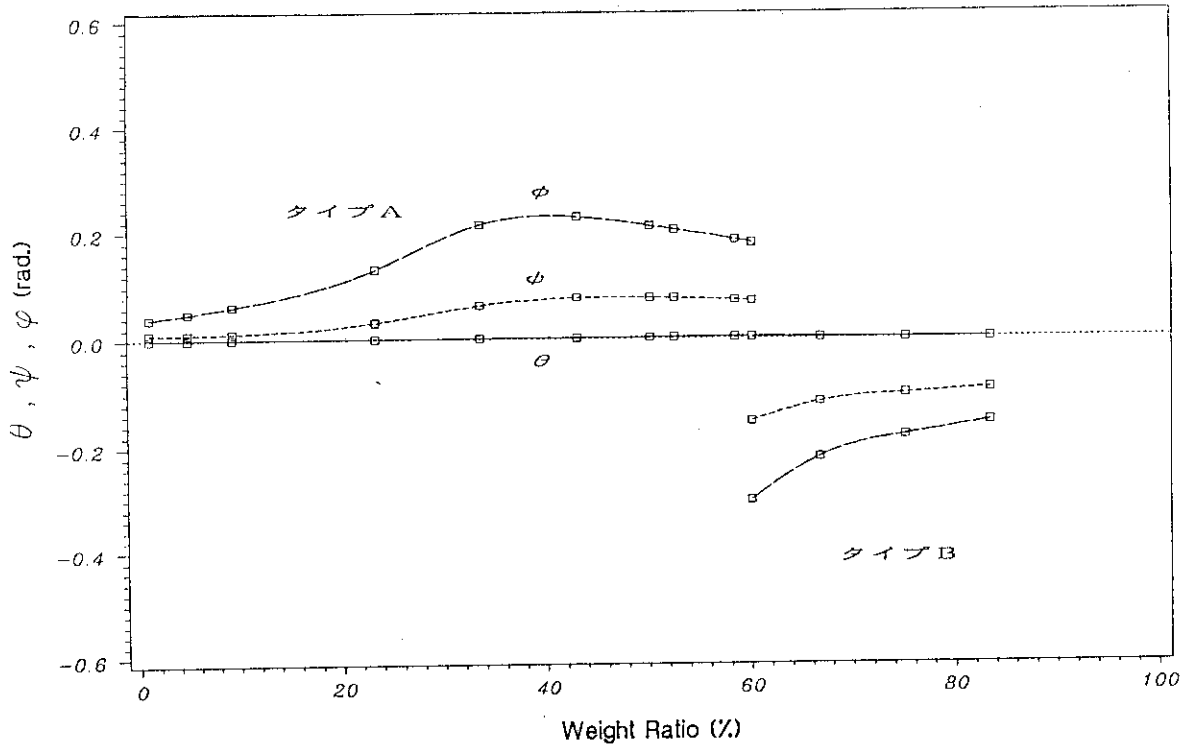
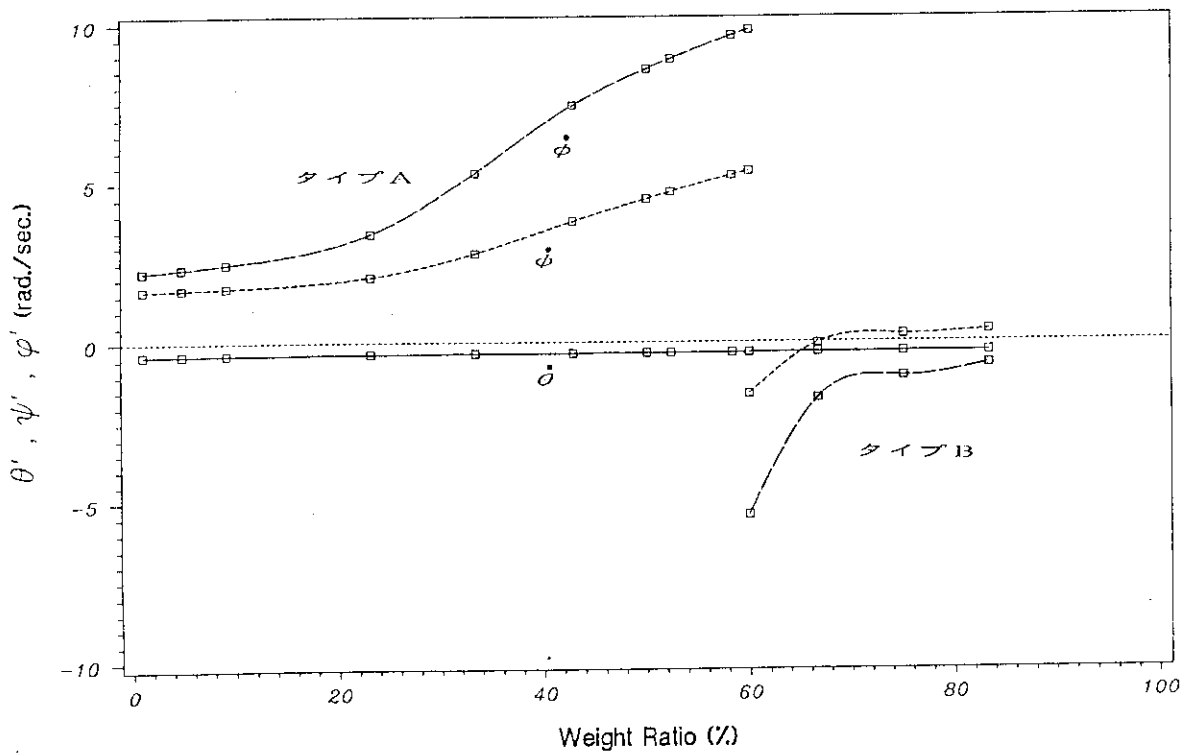


Fig. 3.5 Simulated results of the biped locomotion robot with passive swing arms



(a) angle



(b) angular velocity

Fig. 3.6 Influence of arm motion and arm mass to compensating motions ①

[initial value]

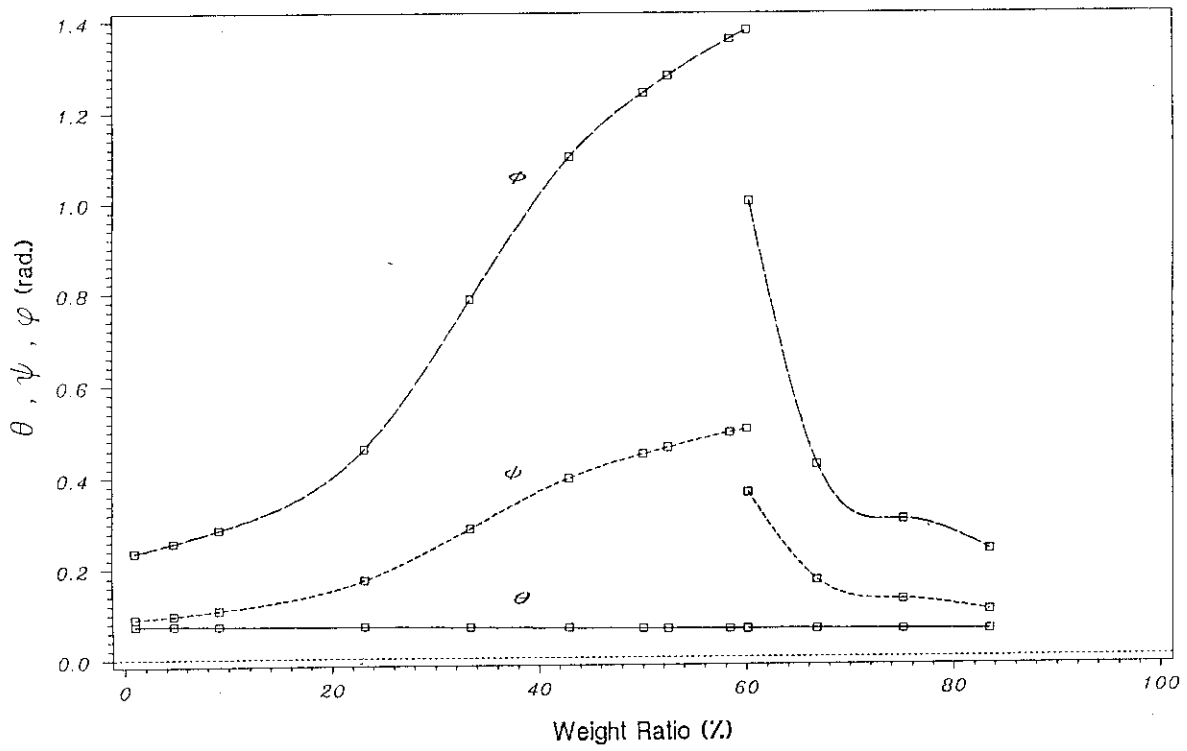
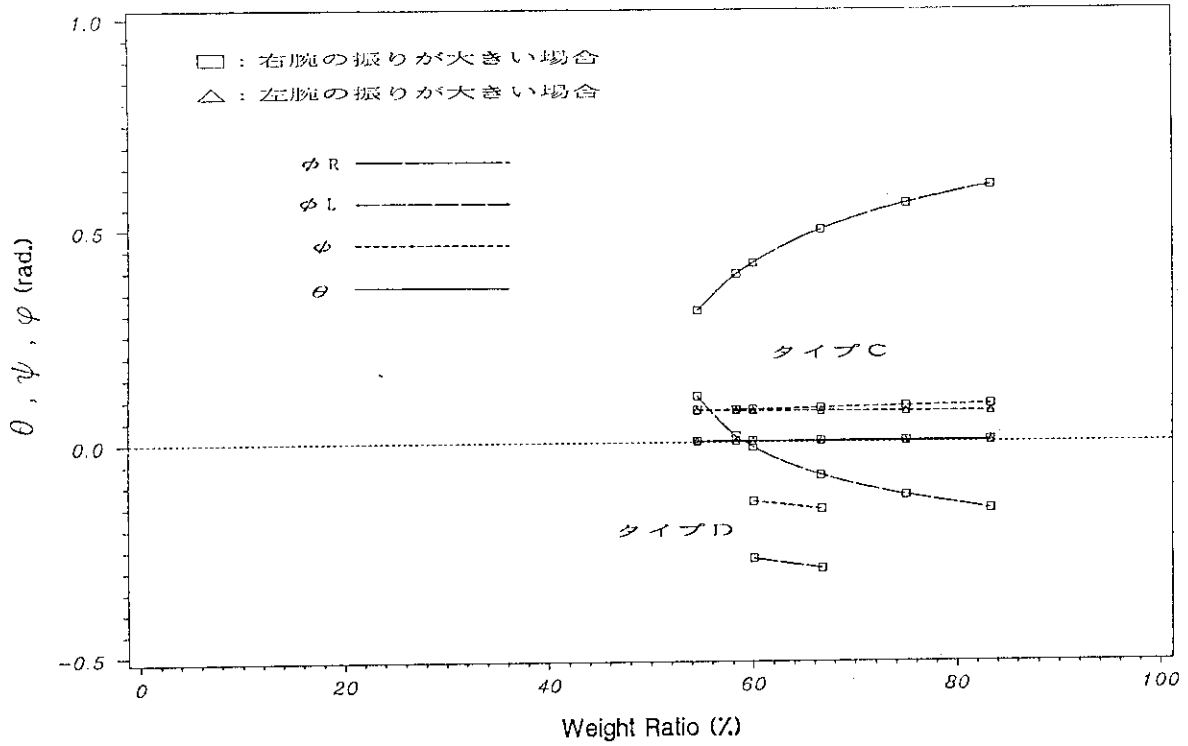
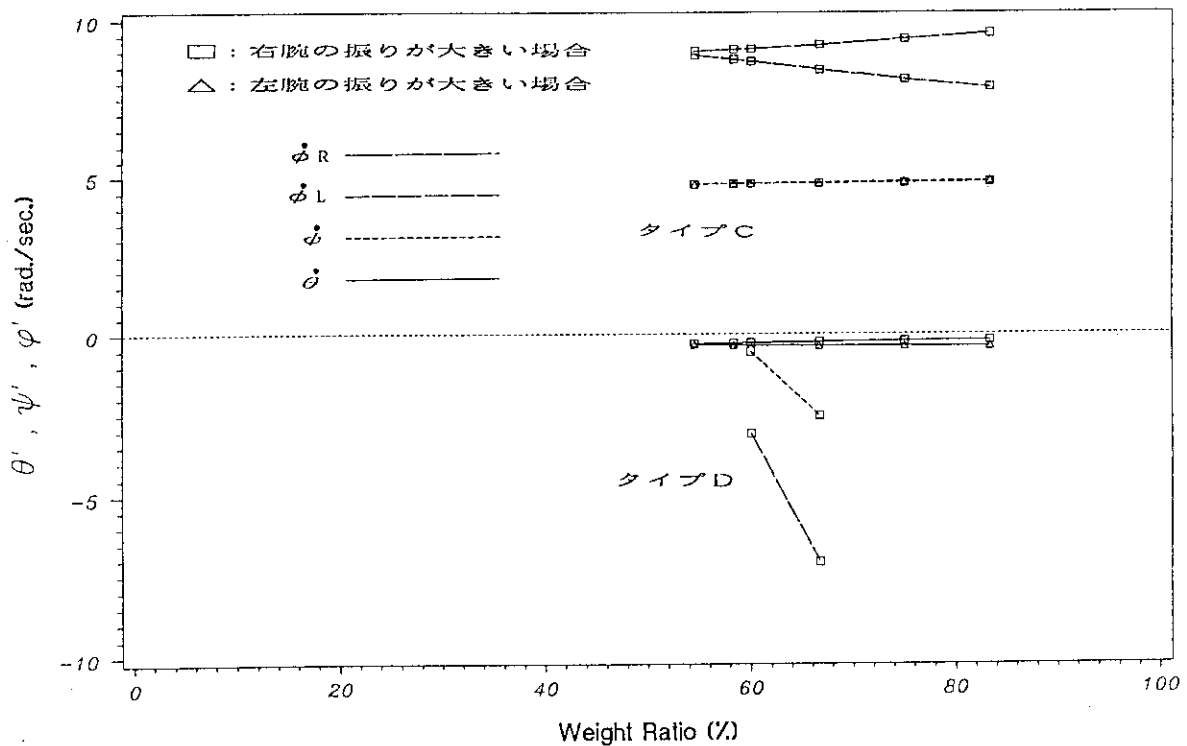


Fig. 3.7 Influence of arm motion and arm mass to compensating motions ①
[amplitude]



(a) angle



(b) angular velocity

Fig. 3.8 Influence of arm motion and arm mass to compensating motions ②
[initial value]

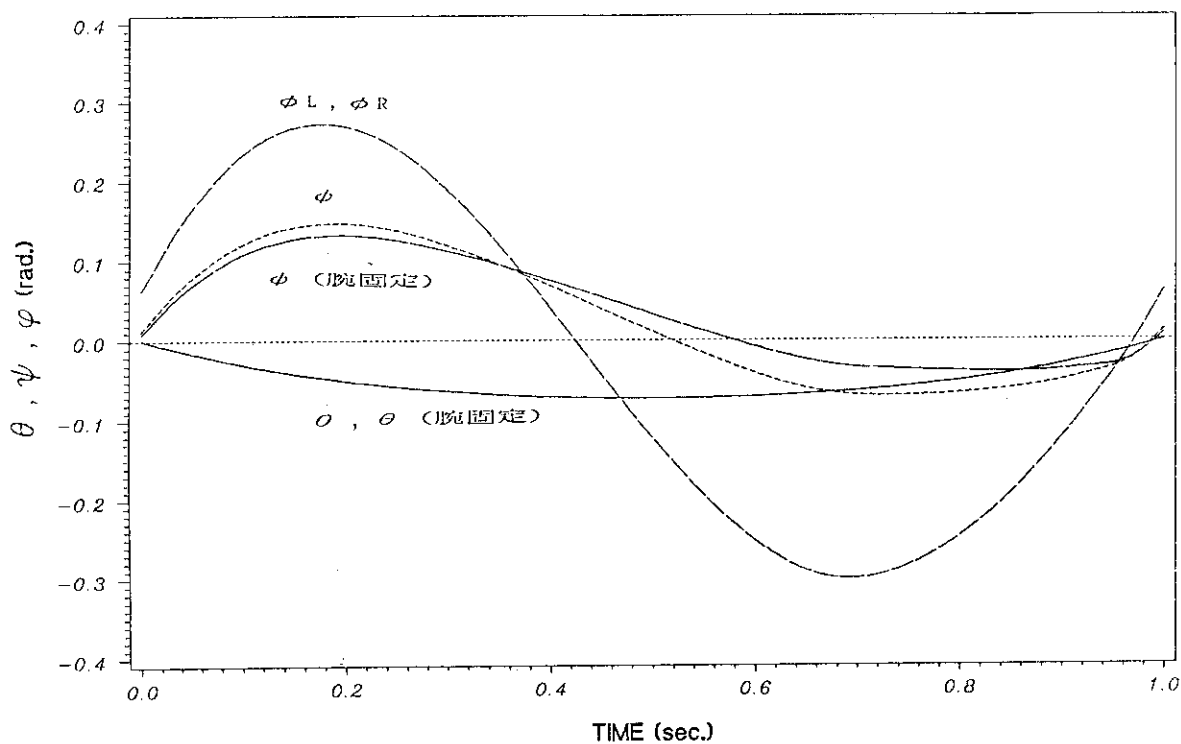
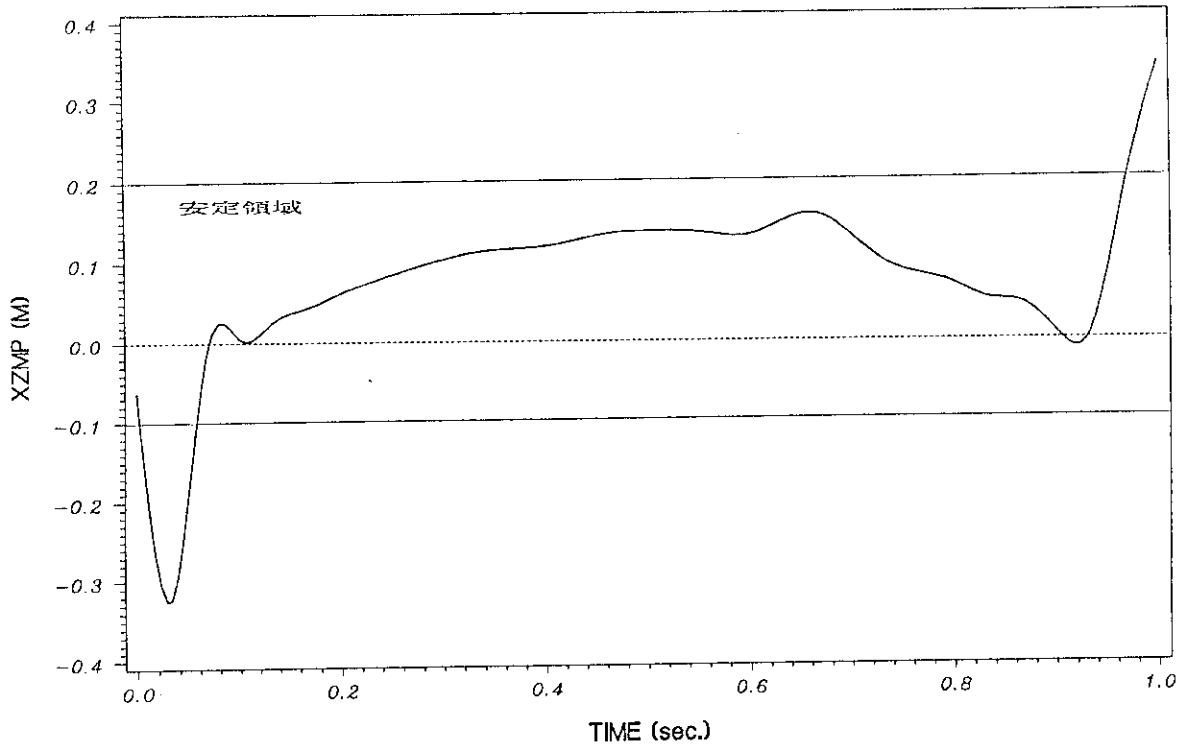
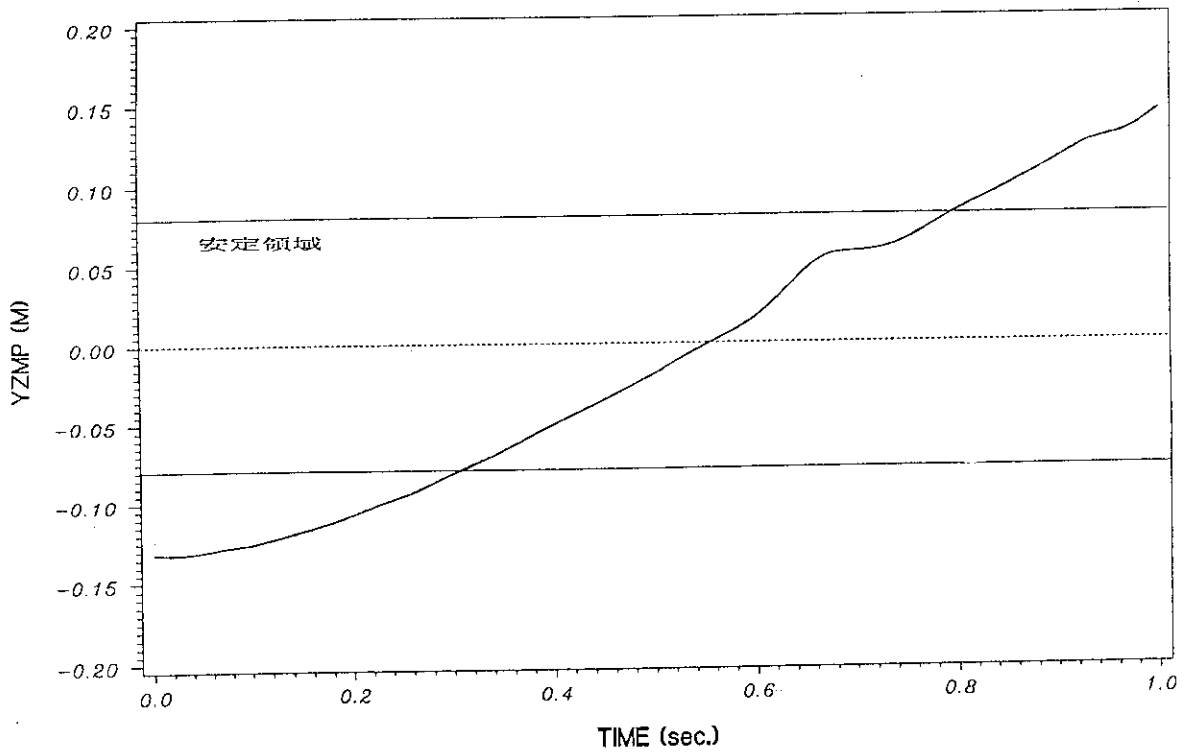


Fig. 3.9 Comparison of various simulated results

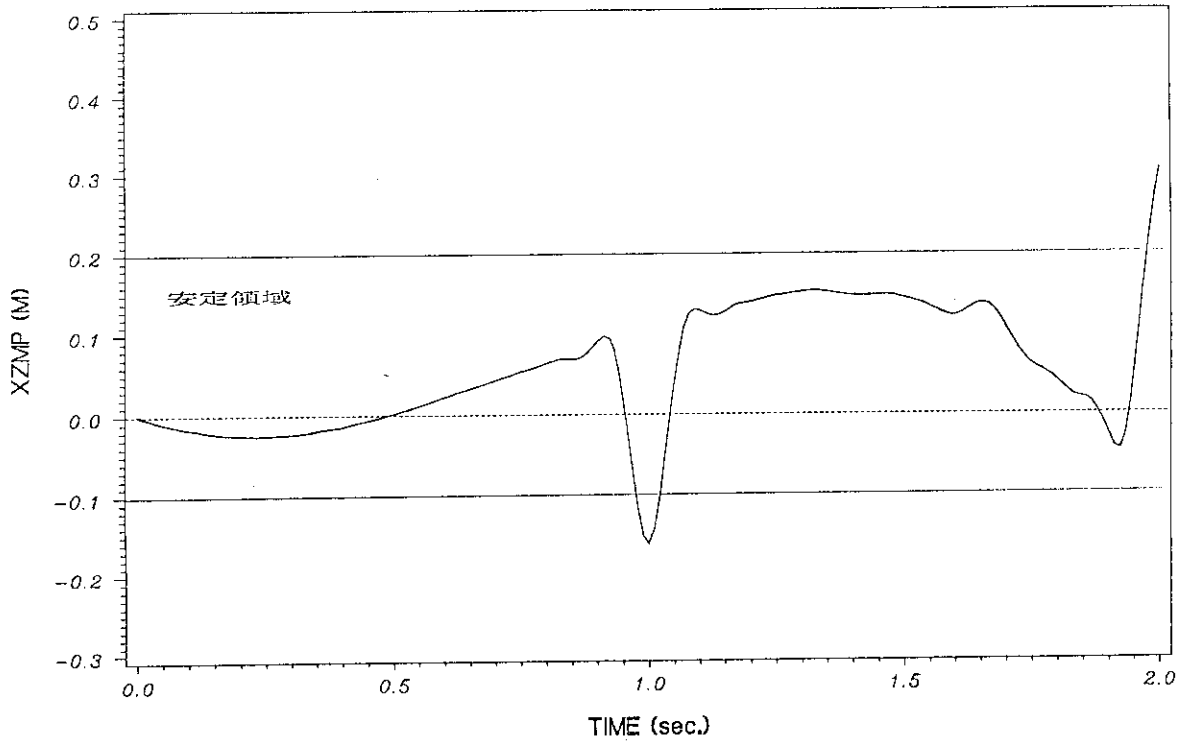


(a) X_{ZMP}

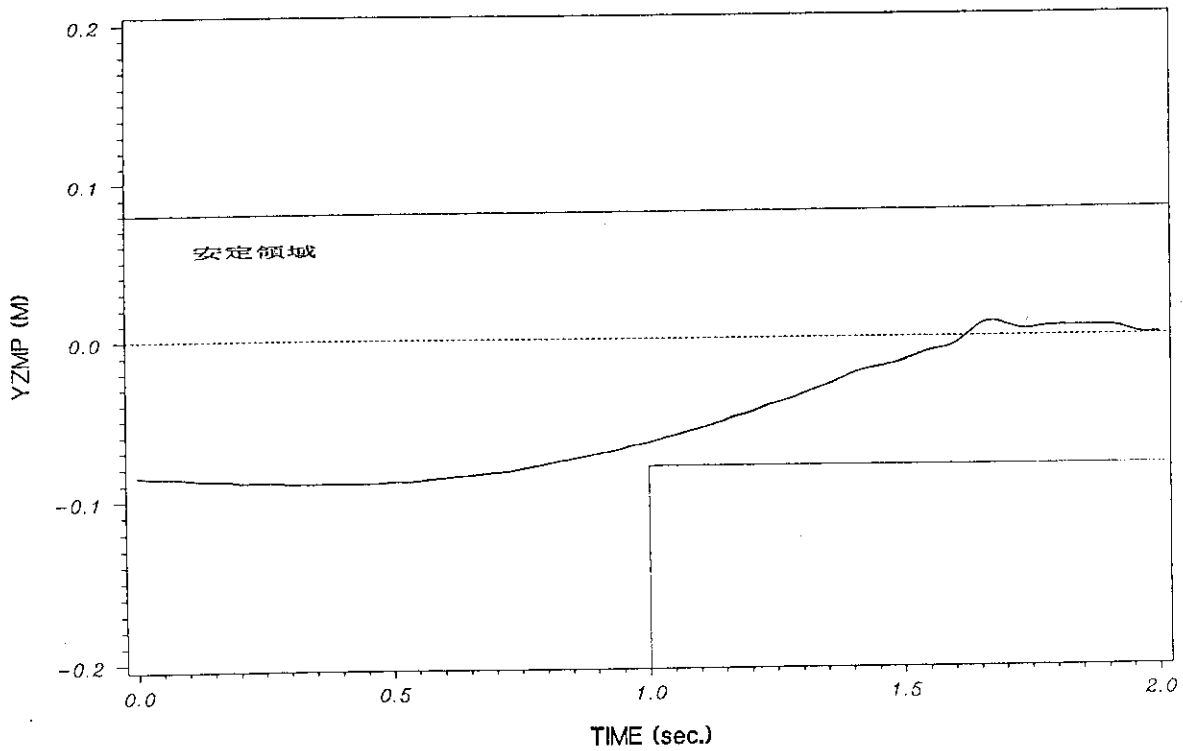


(b) Y_{ZMP}

Fig. 3.10 Example of simulated results for ZMP

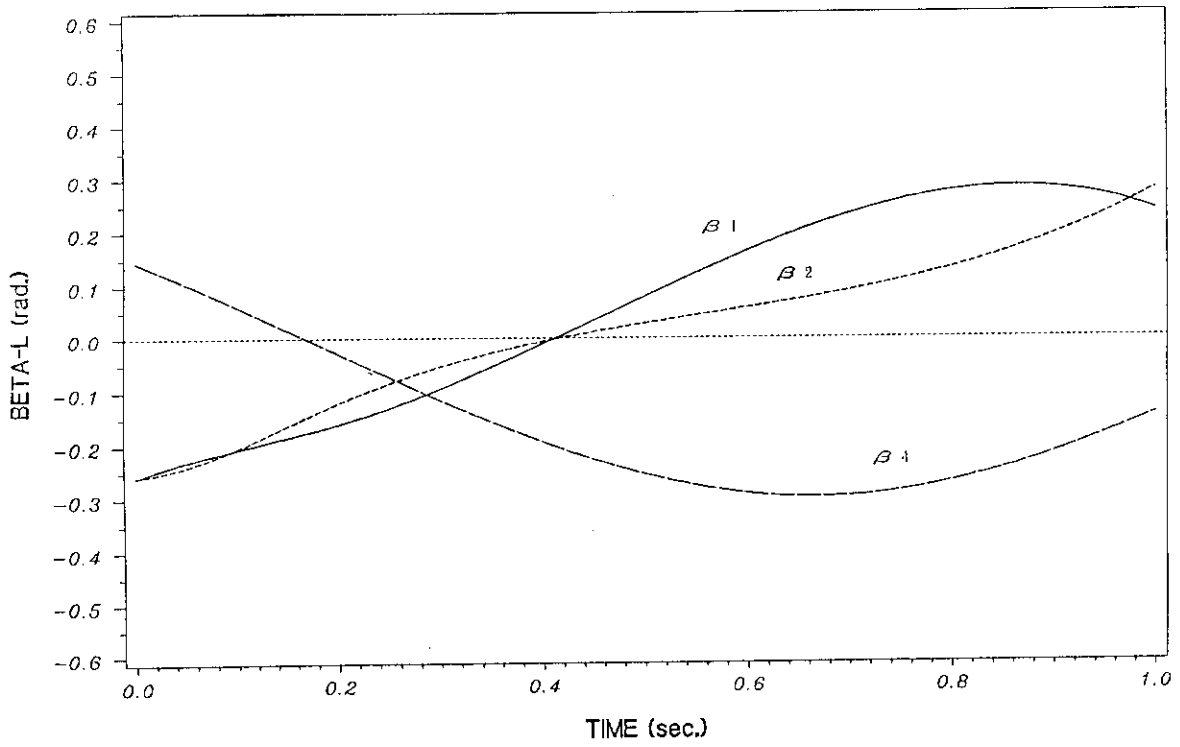


(a) X_{ZMP}

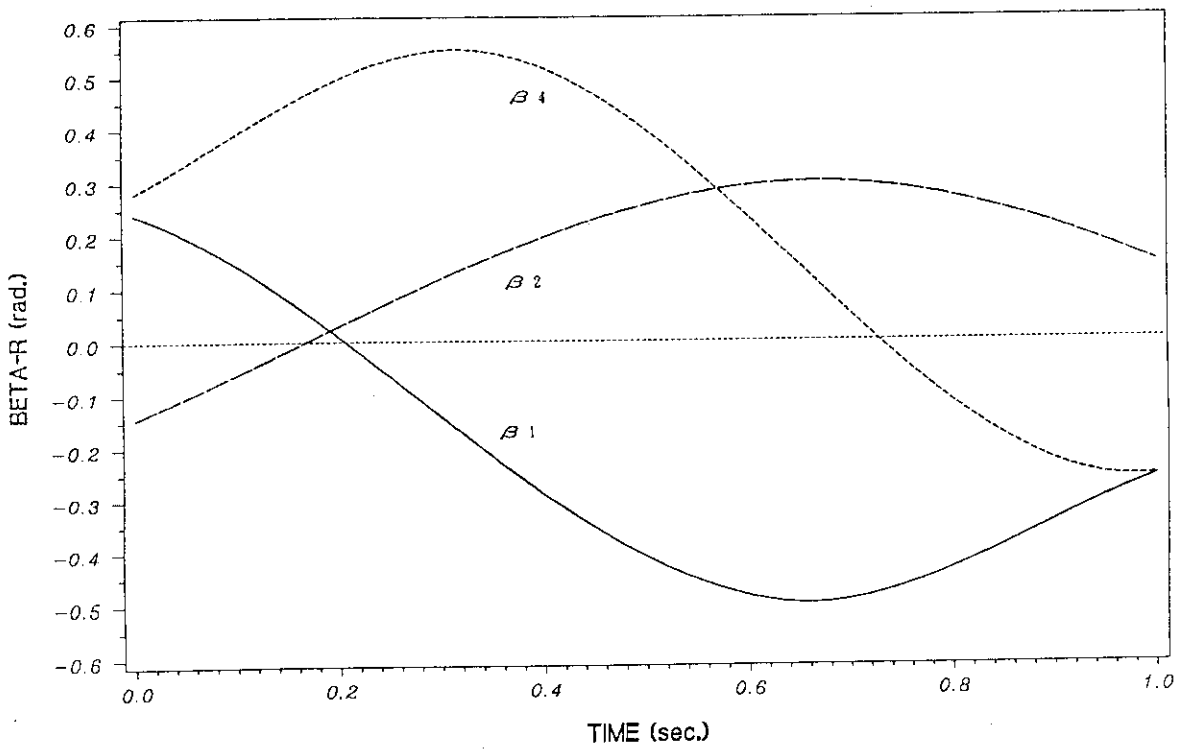


(b) Y_{ZMP}

Fig. 3.11 Example of simulated results for ZMP (standstill 1.0 sec.)

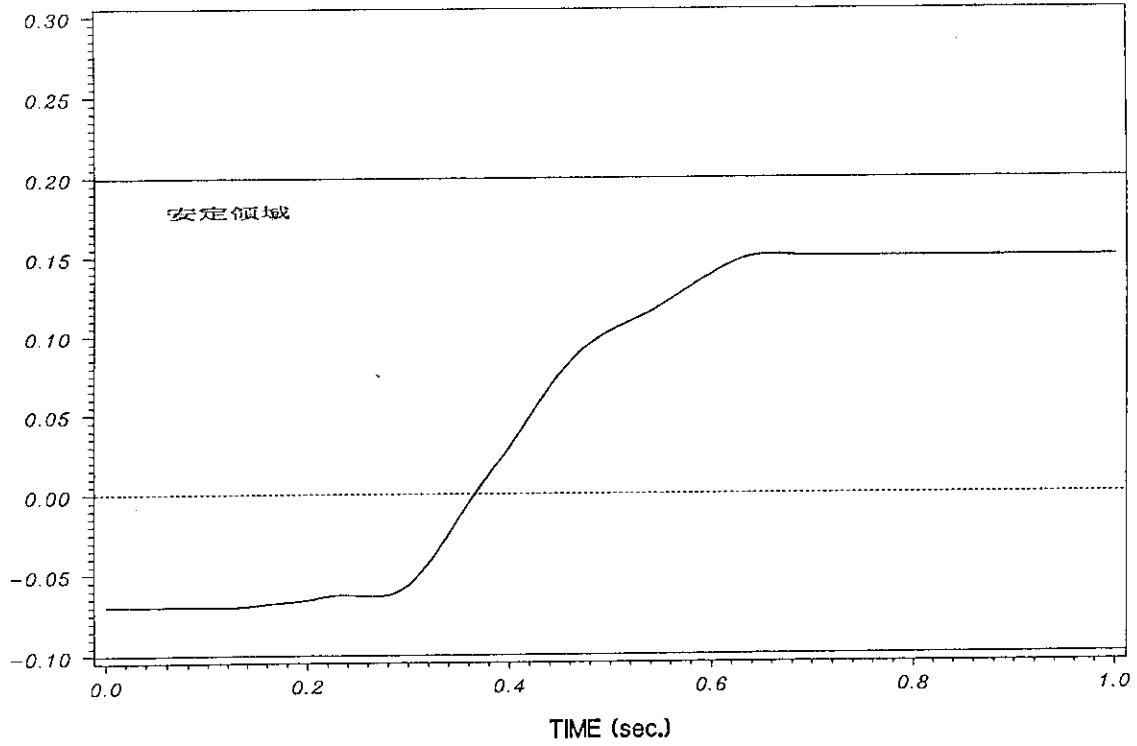


(a) Left leg and arm

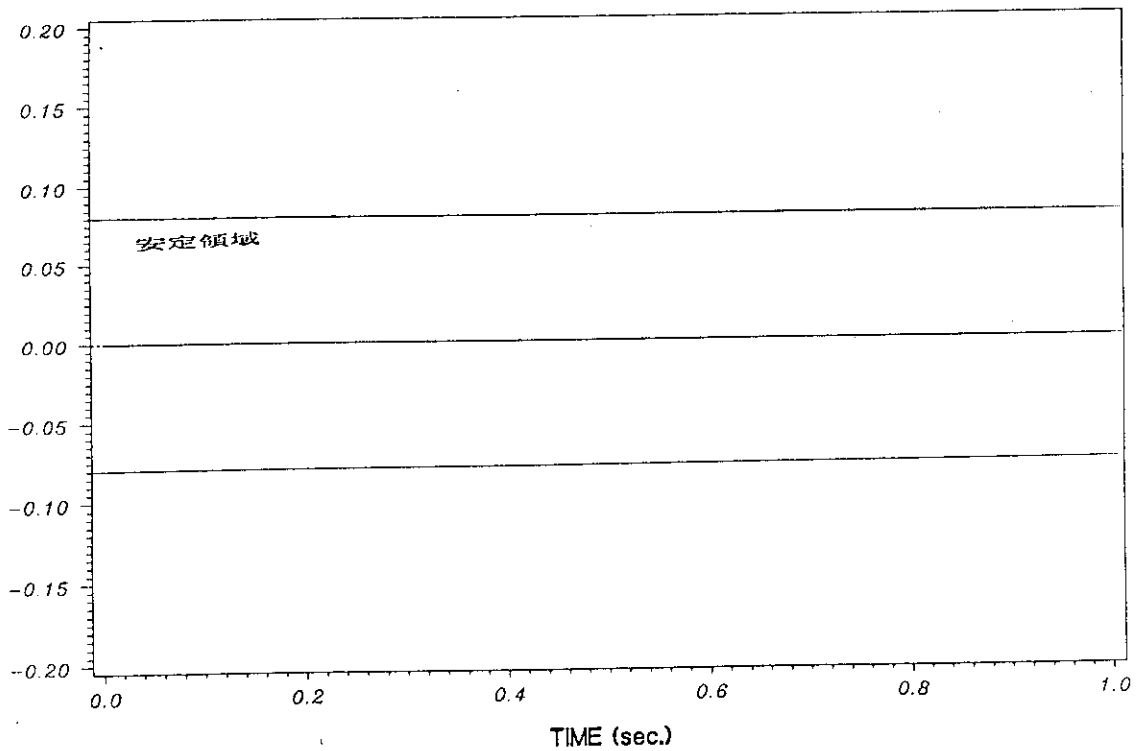


(b) Right leg and arm

Fig. 3.12 Basic walking pattern of "H2"

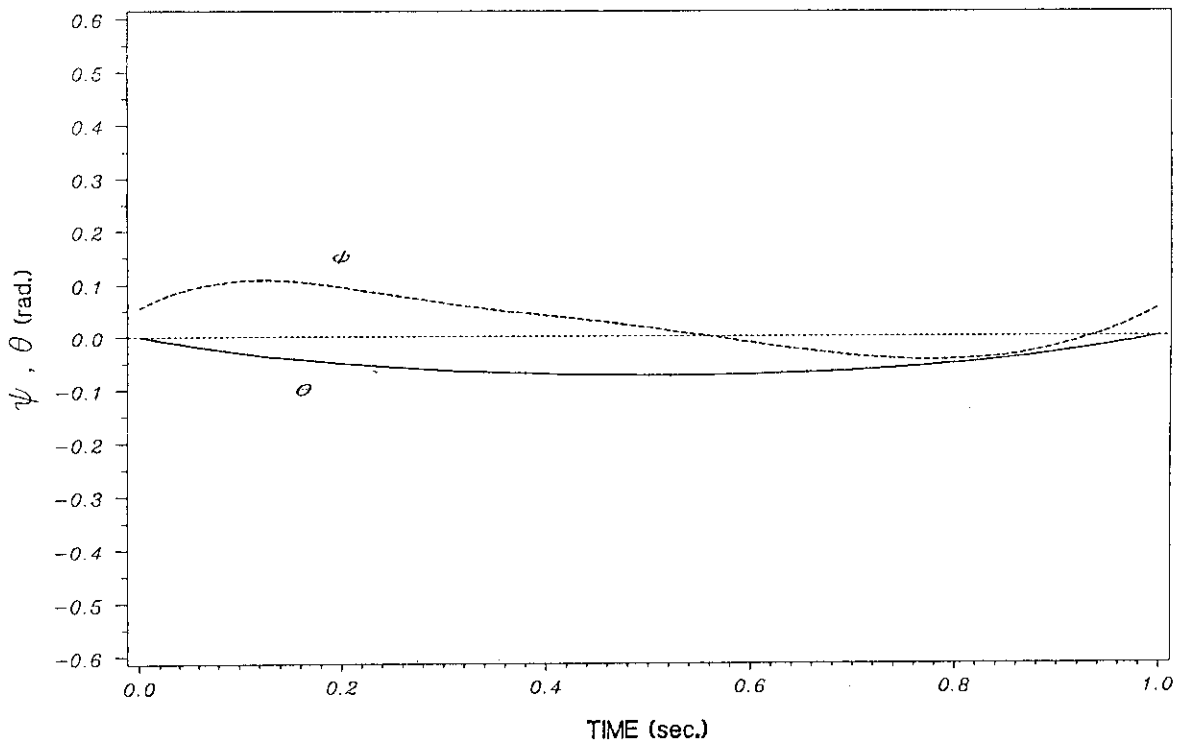


(c) X_{ZMP}



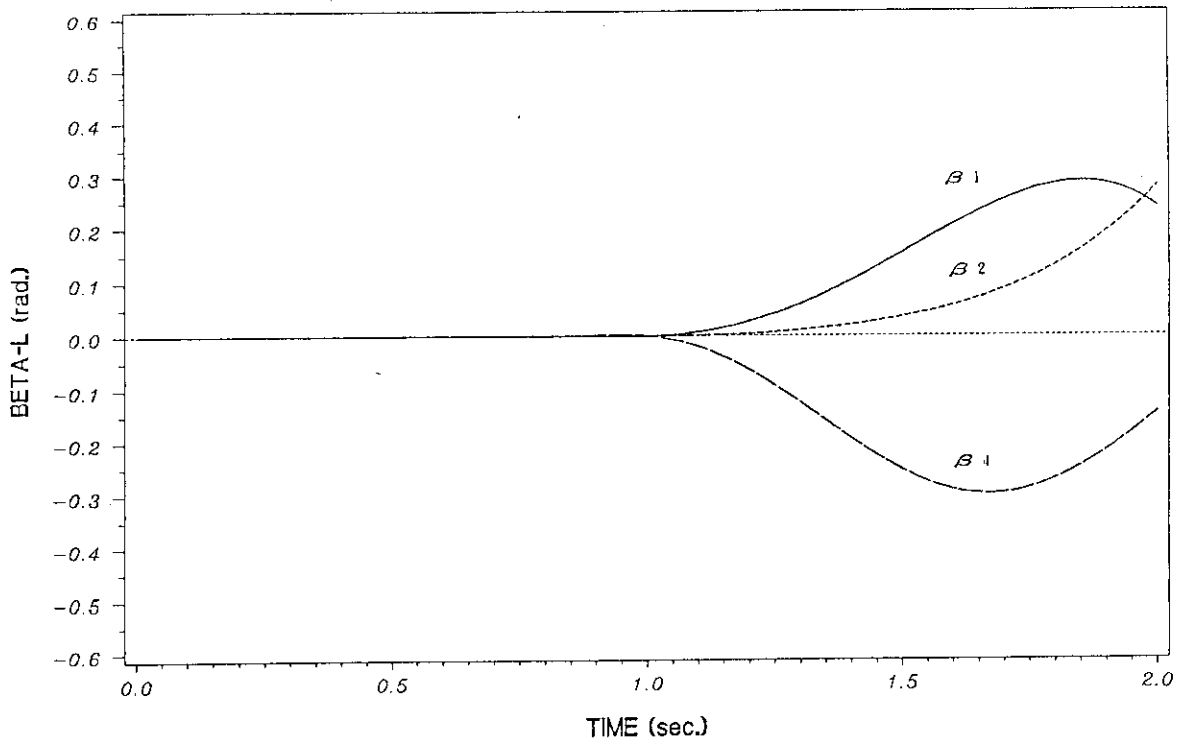
(d) Y_{ZMP}

Fig. 3.12 (Continued)

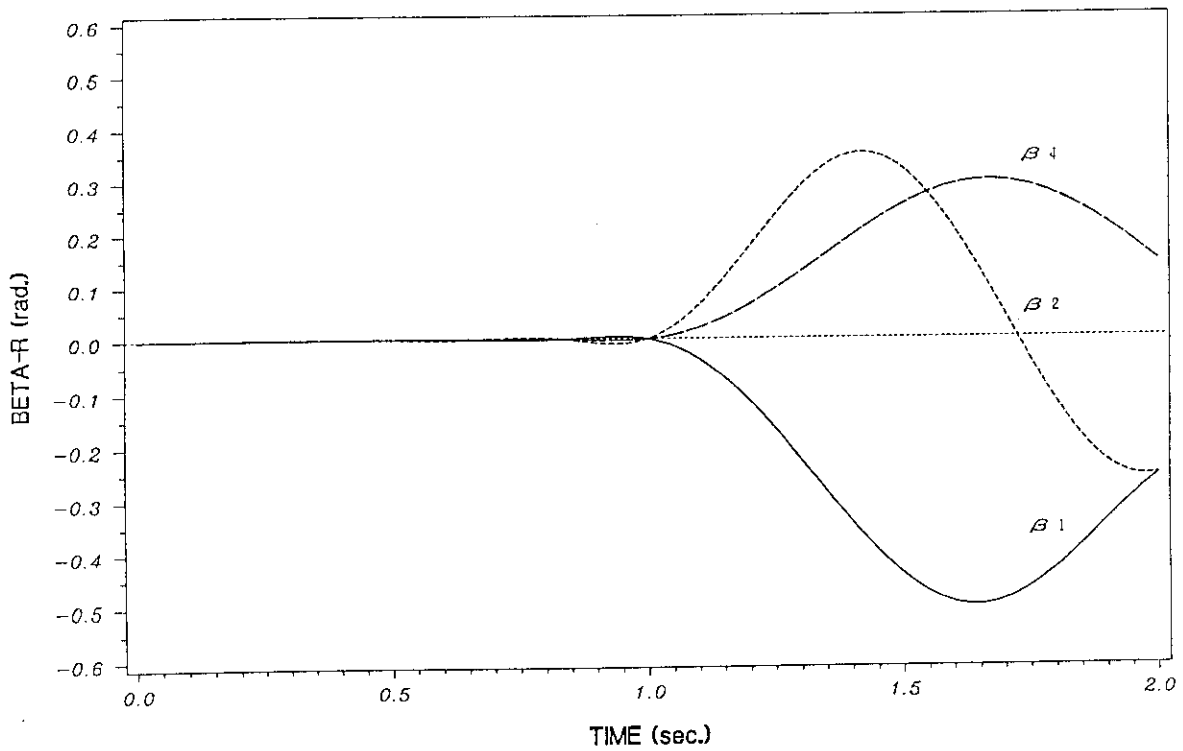


(e) Simulated results

Fig. 3.12 (Continued)

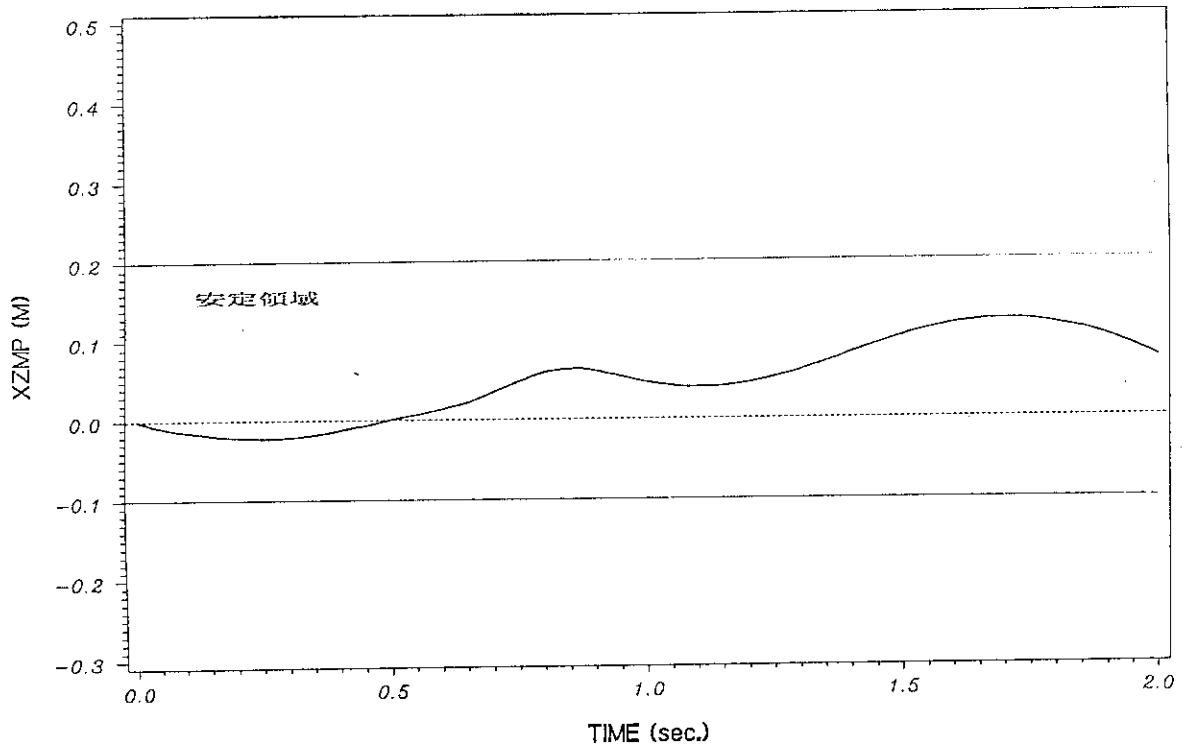


(a) Left leg and arm

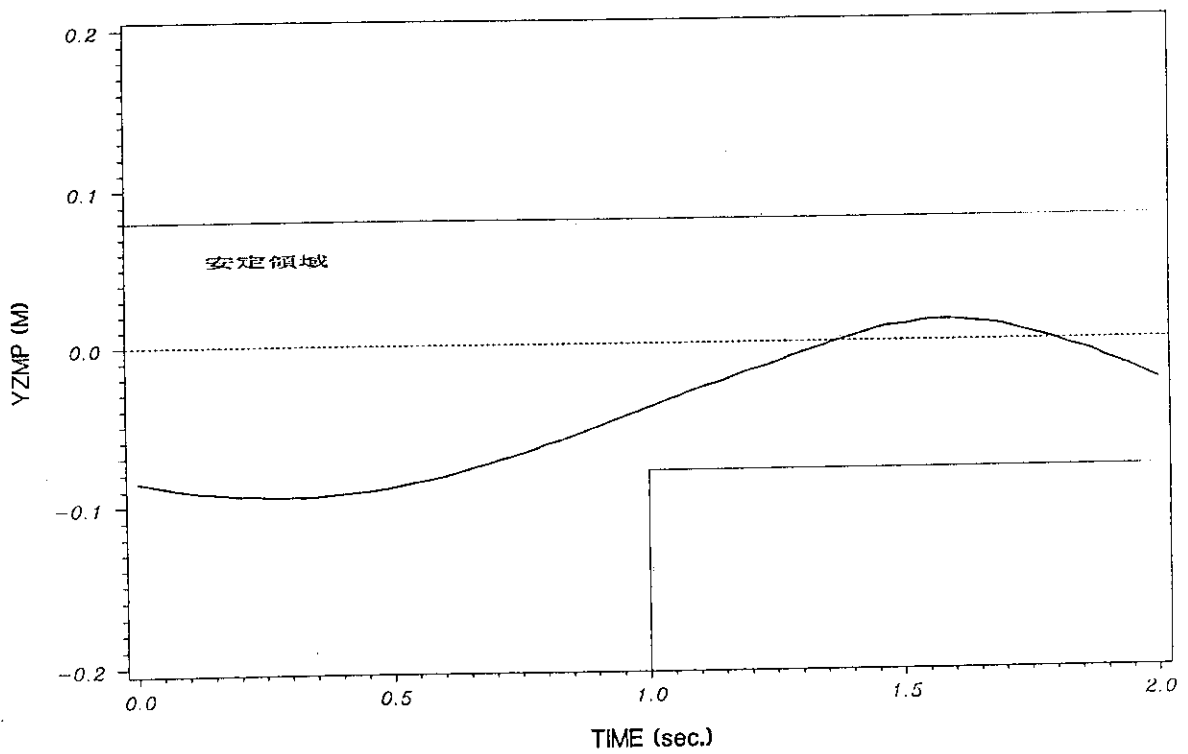


(b) Right leg and arm

Fig. 3.13 Walking pattern of starting motion

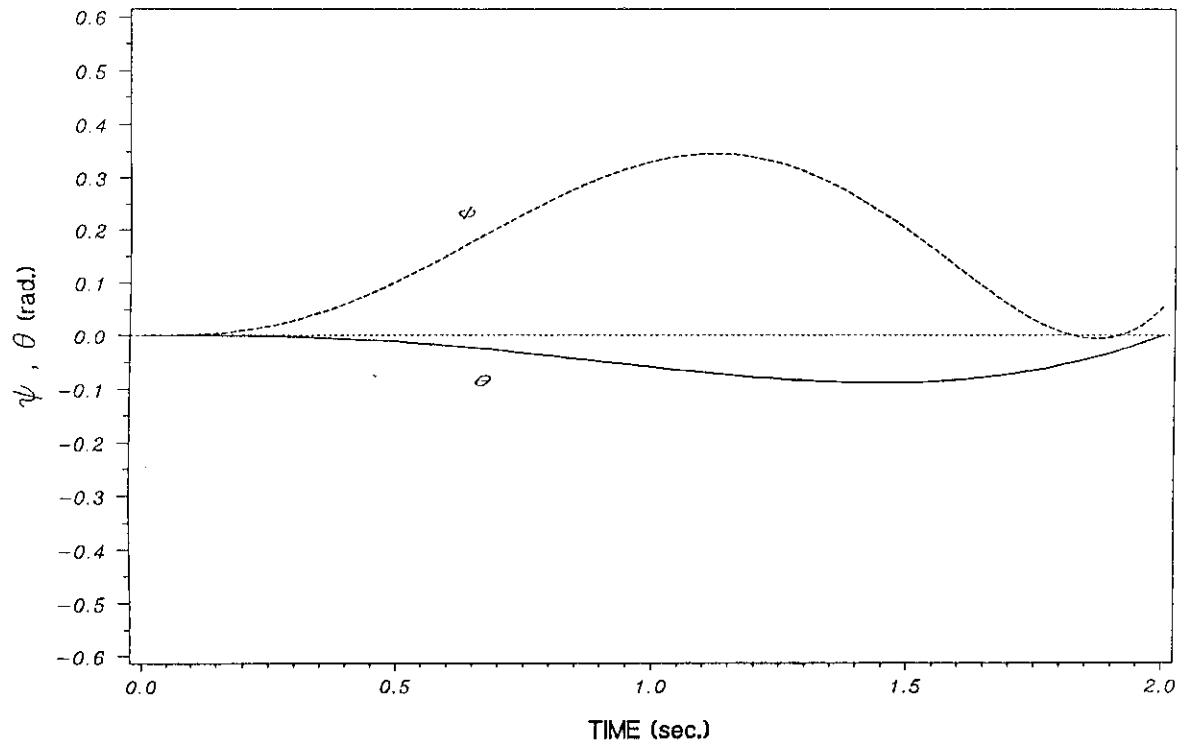


(a) X_{ZMP}



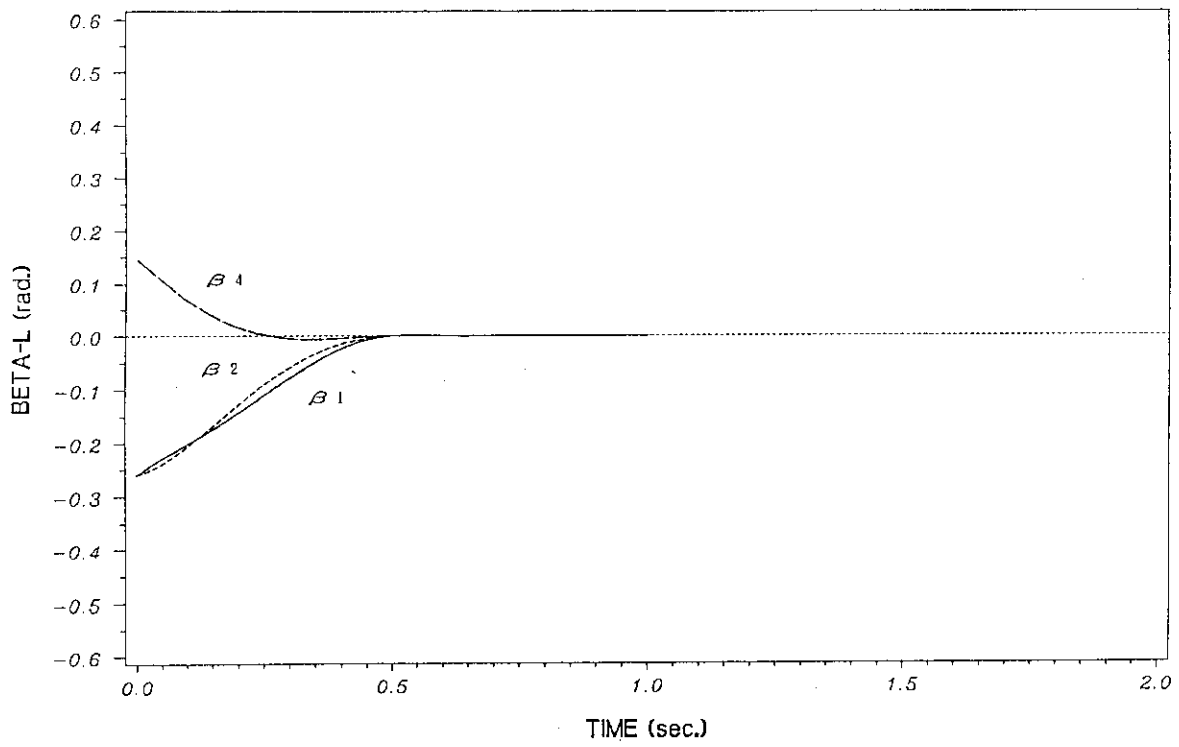
(b) Y_{ZMP}

Fig. 3.14 Simulated results of starting motion

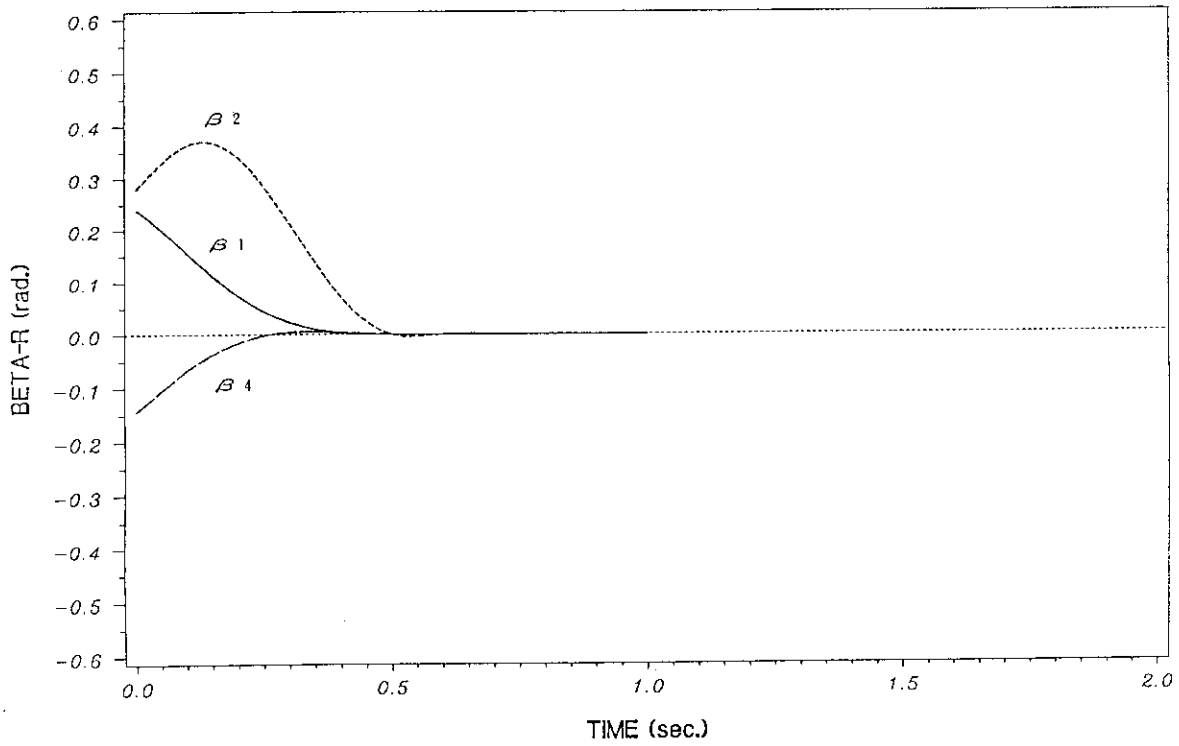


(c) Compensating motions

Fig. 3.14 (Continued)

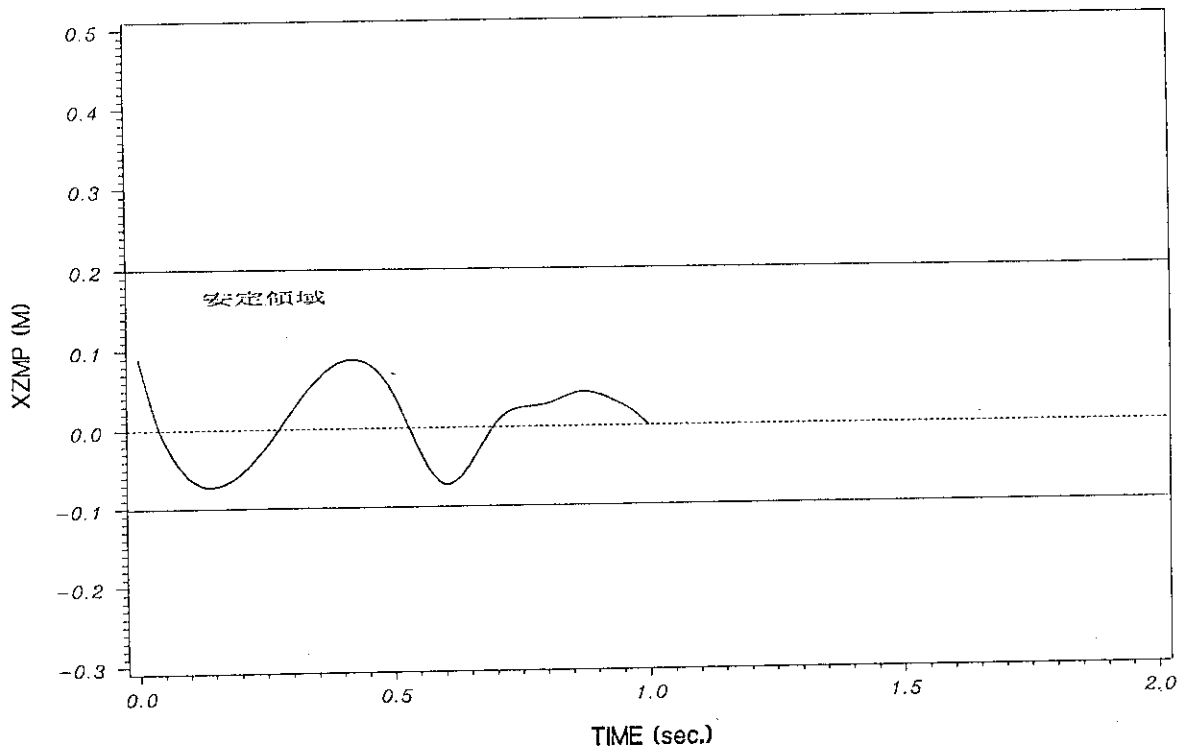


(a) Left leg and arm

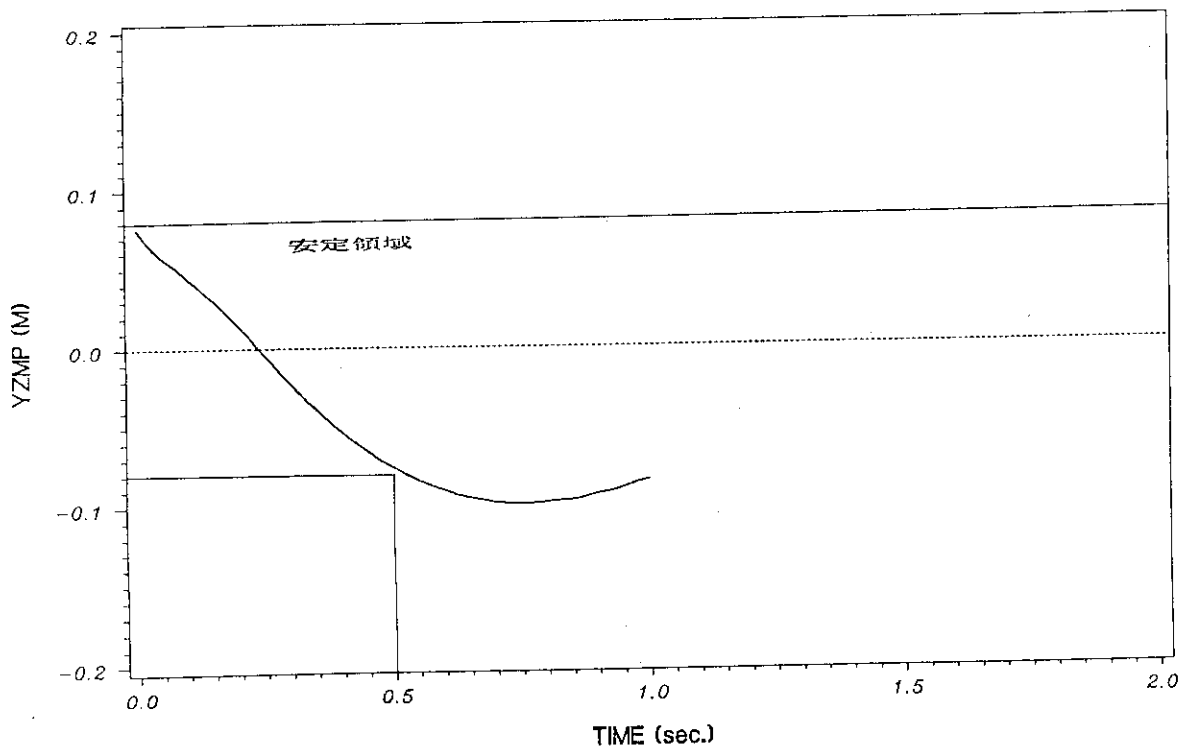


(b) Right leg and arm

Fig. 3.15 Walking pattern of stopping motion

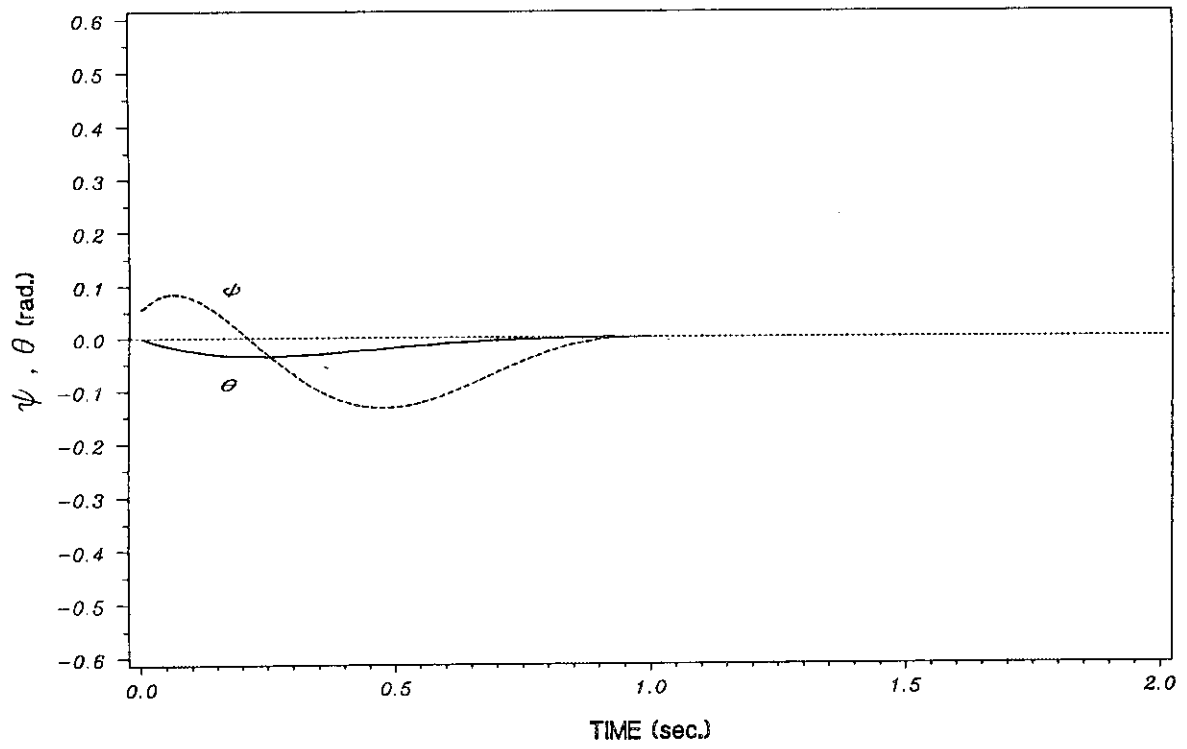


(a) X_{ZMP}



(b) Y_{ZMP}

Fig. 3.16 Simulated results of stopping motion



(c) Compensating motion

Fig. 3.16 (Continued)

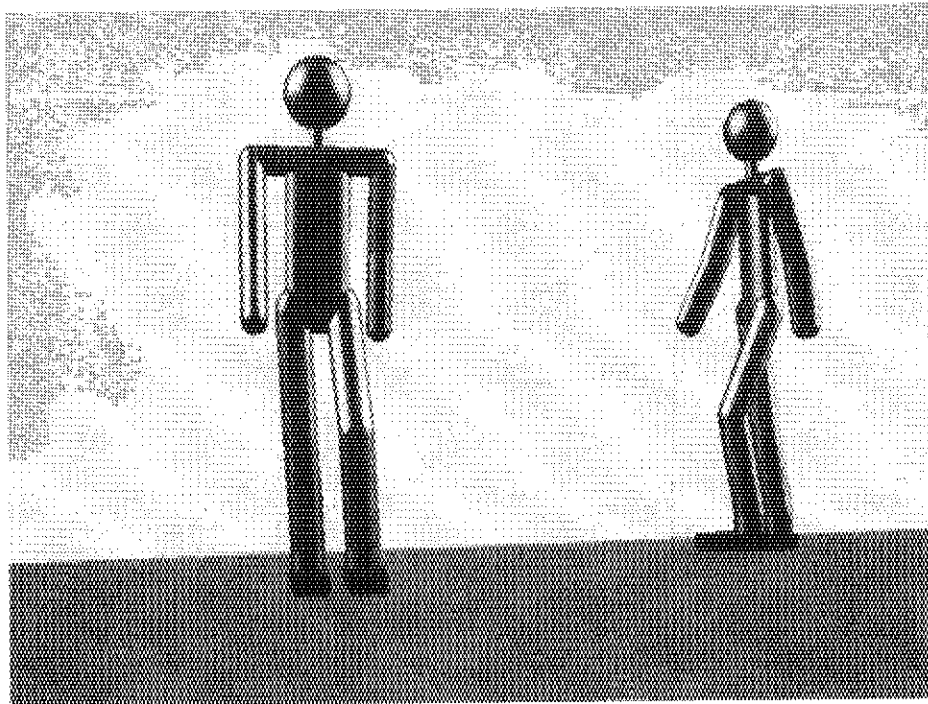


Fig. 3.17 Test scene of flat floor walking

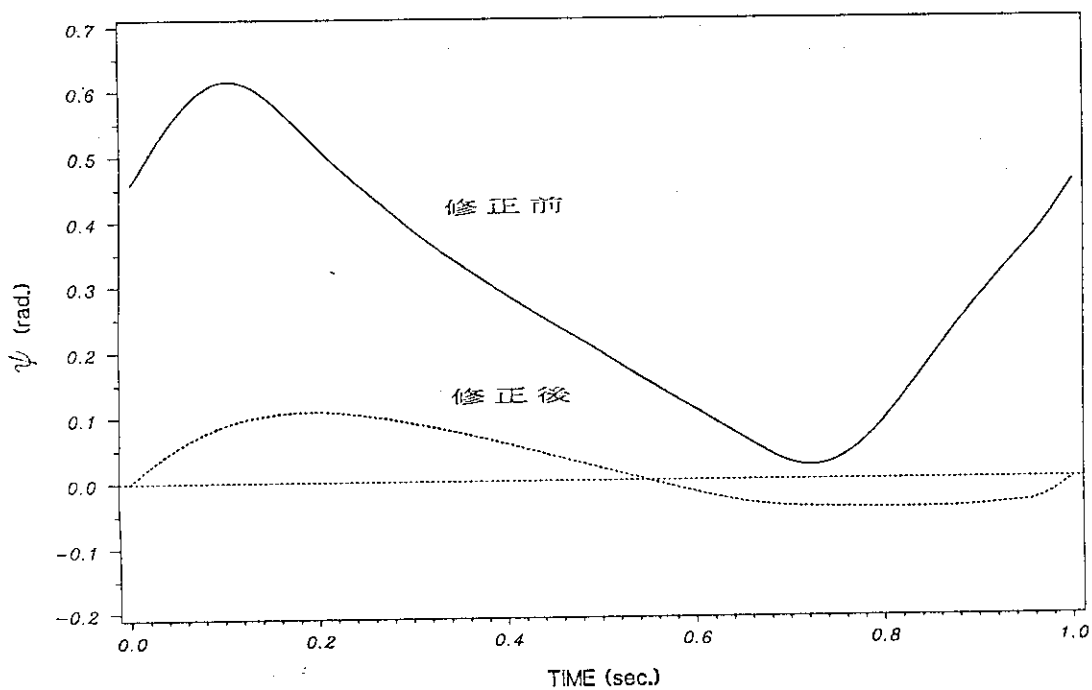
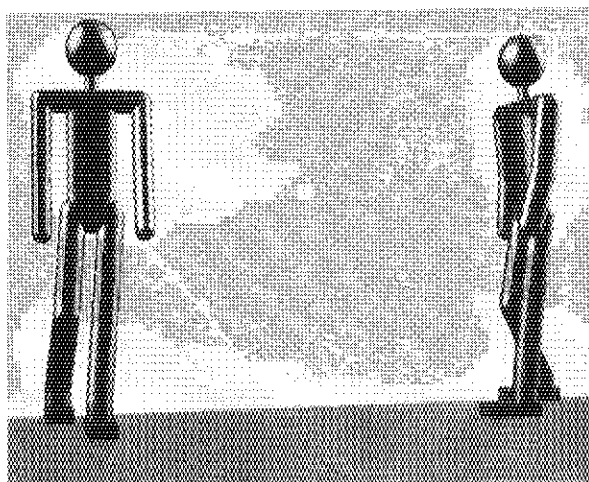
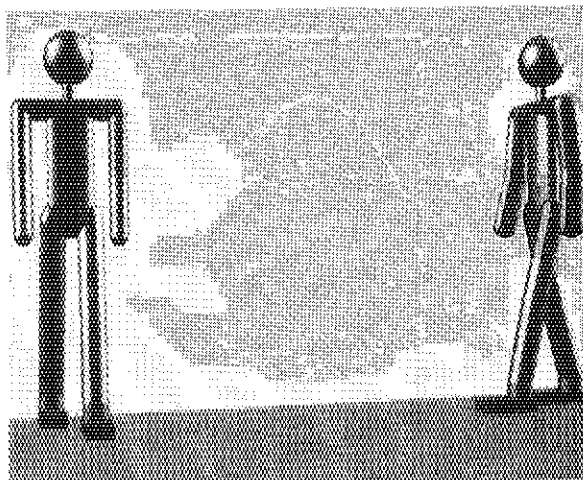


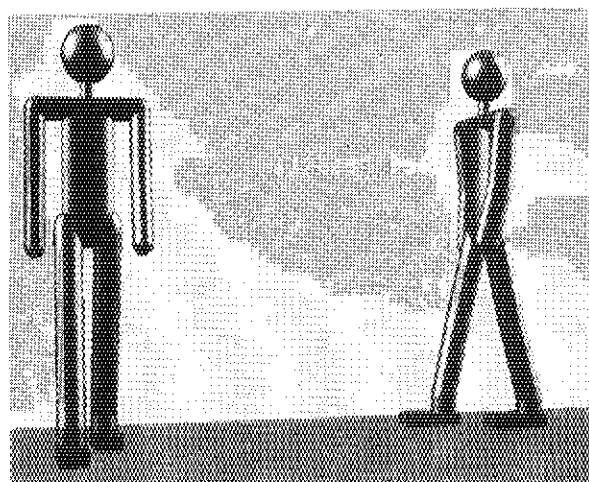
Fig. 3.18 Comparison of various simulated results



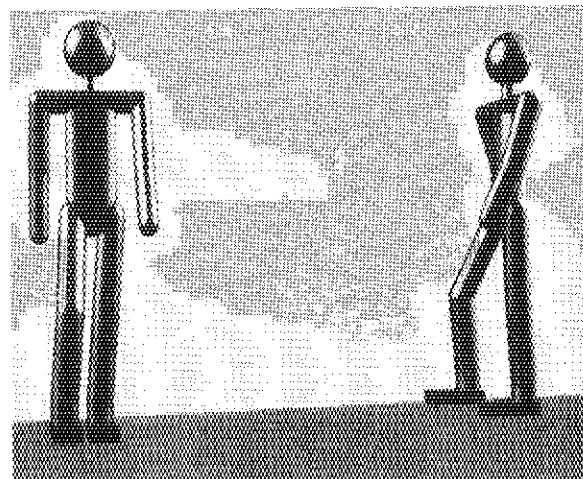
0.33 sec.



0.0 sec.



1.00 sec.



0.67 sec.

Fig. 3.19 Flat floor walking

4. おわりに

4. 1 まとめ

ここで得られたこれまでの結果を要約すると次のようになる。

- (1)腕を歩行動作に合わせ受動的に振らせた場合は、腕固定型と比較して補償動作は大きくなり、上肢（腕）の質量増加に伴う補償動作についても腕固定型と全く逆の結果を得ており、システムに対しネガティブに作用する。
- (2)腕を能動的に振らせる場合、上体との協調による補償動作を作成できる。但し、このためには何らかの評価関数が必要である。
- (3)定常歩行に繋がる過渡歩行をシミュレーションする補償動作規定型の手法を開発し、その有効性を確認した。
- (4)歩容のスプライン関数による補間の導入によってスムーズな歩容の作成ができ、安定歩行作成に非常に有効であった。
- (5)過渡歩行に対し静止状態の導入は有効である。

なお、以下に示すものについては定量的な評価は行っていないが、これまでのシミュレーションから得た結果である。

- (6)ZMPと補償動作との関係は、ZMPを変更すると補償動作はそれに伴い同方向に移動する。
- (7)腰の位置を制御することにより、上体の補償動作を小さくすることができる¹⁰⁾。
- (8)骨盤のY軸方向の長さを小さくすることにより両脚の補償動作を小さくできる。
- (9)骨盤のZ軸方向の長さの変更で上体の補償動作を変えることができる。但し、これについてはまだ正確なシミュレーションを行っていない。
- (10)前頭面と矢状面内の動きはお互いに与える影響は小さく、一般に無視できる。

二足歩行のシミュレーションは、これまで述べて来たようにプロトタイプではあるが平坦地形での直進完全歩行のシミュレーションが可能になった。しかしながら、現在のモデルをそのまま用いて今後の研究を進めて行くには以下のような問題点がある。

- ①現モデルは人間のデータ（質量、寸法等）を元にしたものでハードウェアを考慮したものではない。特にアクチュエータやジョイントについては、理想的なトルクを与えら

れた方向に発生することができ、無質量である。しかし、1関節に2自由度持たせているものが4つあり、さらに方向転換の自由度を与えると3自由度持つものも生じる。これはハードウェア的には機構面からもアクチュエータの大型化の面からも無視できない問題であり、モデルの再構成を要する。

②①と若干重複するが、ハードウェア的に考えると、現在指定している両脚の左右の動きによって補償動作を行わせたままで方向転換を考えた場合、股関節に3自由度必要となりアクチュエータの構成上困難になるため、この補償動作を行わせているボディが適当であるとは言い切れない。

③定常歩行における両脚支持期を持たないため、支持脚の切り換えが不明確である。現時点では単脚支持の繰り返しで瞬時に支持脚が切り替わるとしているが、接地時の衝撃力やZMPの繋がり等の考慮が不十分である。

④本モデルのような多自由度モデルでは、剛体系での取扱は慣性モーメントの計算式が複雑になりやすく、式全体を明確に把握するのが難しいうえに計算時間もかかる。一般にロボットをモデル化する場合、構造部材の考慮から剛体系での記述が普通であるが、質量がボディの一部に集中するような場合（例えば、各ボディの質量がアクチュエータの質量に対して無視できるほど小さい場合）は質点系で表すことができる。

このように現モデルはいくつかの問題点を抱えており、特にハードウェアのロボットを設計するために要求される、ロボットの移動機構に必要な基盤技術開発のためのモデルとしては不十分なものと言わざるをえない。このため、二足歩行ロボットの設計に関して調査を行い、この結果を元に今後次の問題について検討していく予定である。

①問題点の①～③を考慮し、ハードウェア化を前提としたモデルの開発を行う。これには、方向転換のための自由度も付加し、補償動作を行うボディについても再検討する。

②問題点⑤に対し、①でのアクチュエータの配置や構造部材を考慮して、質点系でのモデル化の可能性について検討する。

③①・②に基づき二足歩行ロボットの試作を行う。二足歩行ロボットの基本設計としては Fig. 4.1 に示すように、動作として平地や階段等の完全歩行を対象とし、その動作に対する判断や知識は外部のコンピュータに依存する。但し、関節駆動及び制御のためのマイクロプロセッサ等は搭載する。また、油圧アクチュエータを用いた場合、駆動源についてはロボットに搭載する場合と外部設置の場合の両方を考慮し、エネルギー（電力等）や動作命令及び情報は光ファイバーケーブル等で外部から伝達する。

4. 2 ハードウェア化について

4.1節で述べたように今後ハードウェア化を目指したモデルを開発し、実際にハードウェアを構築していく予定である。

ロボットのハードウェア化には大別して以下に示す3つの目的がある。

①シミュレーション結果の検証

実際の動作とシミュレーション結果の比較のみならず、ロボットを数学モデル化する際の前提条件、拘束条件、運動方程式の線型化・非干渉化に関する問題、動作生成アルゴリズムの有効性、機械モデルの基本性能等の妥当性の検証である。

②ハードウェアにのみ依存する部分の検討

第一に、ソフトウェアモデルのシミュレーションに必要な機構の特性や構造部材、アクチュエータ等に関するデータの収集。第二に、ハードウェア的な問題点（例えば駆動方法、構造部材にかかる応力、アクチュエータの耐荷重、床面の状況、ロボット製作上のコストや加工法、コンピュータ搭載の可否等）の明確化である。

③制御性の検討

制御方法の妥当性の検証である。人間型ロボットのように多自由度モデルでは必然的にアクチュエータの数が多くなり制御が難しくなる。また、制御に不可欠なセンサ情報の利用（センサの種類）、それらの情報のコンピュータによる演算処理能力に起因する実時間制御の問題等を明確にする。

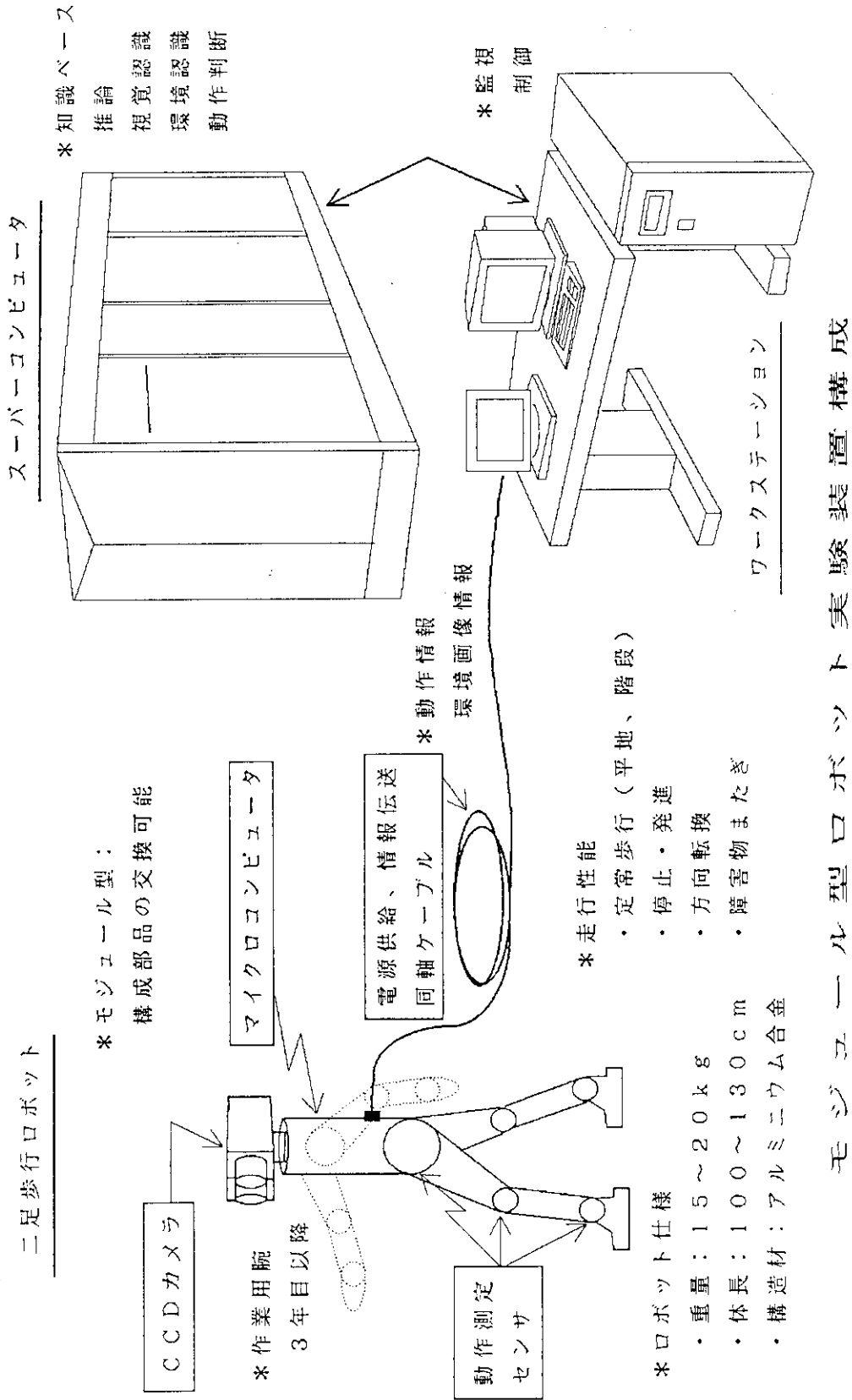


Fig. 4.1 Conception of a biped locomotion robot

謝 辞

本報告書をまとめるに当たり表現上の助言を頂きました情報システムセンター秋元正幸氏、藤井実氏に深く感謝します。また、研究を進めるに当り、適切な助言と指導を頂きました情報システムセンター室長浅井清氏に深く感謝します。

参考文献

- 1) 浅井清 他：原子力知能化システム技術の研究、私信、1988.
- 2) 浅井清・久米悦雄 他：原子力知能化システム技術の研究 ー昭和63年度作業報告ー、JAERI-M 89-023、1989.
- 3) 浅井清・久米悦雄 他：原子力知能化システム技術の研究 ー平成元年度作業報告ー、JAERI-M 90-060、1990.
- 4) Takanishi, A., et al. : A CONTROL METHOD FOR DYNAMIC BIPED WALKING UNDER UNKNOWN EXTERNAL FORCE、IROS '90、PP.795-801、1990.
- 5) 佐野明人・古莊純次：角運動量制御による2足歩行ロボットの3次元動歩行、計測自動制御学会論文集、Vol.26、No.4、pp.459-466、1990.
- 6) M. Vukobratović (加藤一郎・山下忠 訳)：歩行ロボットと人工の足、日刊工業新聞社、1975.
- 7) 石黒美佐子・藤崎正英：二足歩行ロボット運動学的シミュレーション、JAERI-M 88-080、1988.
- 8) 高西淳夫・加藤一郎・久米悦雄：二足歩行の運動学方程式化 ー直進における完全歩行の運動方程式ー、JAERI-M 90-093、1990.
- 9) 高西淳夫・加藤一郎 他：2足歩行WL-10RDによる動歩行の実現、日本ロボット学会誌3巻4号、PP.67-78、1985.
- 10) 高西淳夫・加藤一郎 他：上体補償機構を有する2足歩行ロボットWL-12の開発、第5回日本ロボット学会学術講演会、PP.579-582、1987.
- 11) 伊理正夫・藤野和建：数値計算の常識、共立出版株式会社、1985.
- 12) 高西淳夫・加藤一郎 他：既知の外力下で歩行可能な2足歩行ロボットWL-12Rの開発 ーシステムの開発と上体と腰の協調動作による動歩行の実現ー、第6回日本ロボット学会学術講演会、PP.111-114、1988.
- 13) 田川善彦・山下忠：上体の質量と二足歩行の歩容の関係、ロボティクス・メカトロニクス講演会'90、pp.195-196、1990.
- 14) 高西淳夫：2足歩行ロボットによる準動歩行、日本ロボット学会誌1巻3号、PP.36-43、1983.
- 15) 越川和忠：SOLVER、私信、1985.

- 16) 佐藤：高並列計算機C A Pとその応用、FUJITSU、Vol. 38、No. 3、1987.
- 17) 樋口健二・久米悦雄・藤崎正英：二足歩行シミュレーションとその動画化、第8回シミュレーション・テクノロジー・コンファレンス、pp. 261-264、1989.
- 18) 藤崎正英・久米悦雄：セルラアレイプロセッサC A Pによる動画作成、私信、1989.
- 19) 高西淳夫・加藤一郎 他：上体と腰の協調動作による2足歩行制御、第27回計測自動制御学会学術講演会、PP. 643-644、1988.

付録A

定常歩行シミュレーションプログラムの概略（腕可動（能動型）モデル）

シミュレーションプログラム（STEADY2C, FORT77）は以下の3つのメインプログラムとそれらを構成するサブプログラム、プログラムライブラリーからなる。

①GAIT

与えられた歩容及び腕の動きのサンプル点を補間し、1秒30分割した値をスムージングする。

ALINT：与えられたサンプル点を線型補間する。

SML E 2：局所的最小二乗近似多項式によってサンプル点での平滑化を行う。次数は6次、観測点の個数は9個。〔FACOM SSL II〕

②BIDYN

①での歩容のデータと初期値サーベイに関するデータを読み込み、2.4節で述べた手法により補償動作を決定する。（初期値修正ルーチン）

DERIV：運動方程式を解く（式(2.16)）。

NIODES：数値計算のコントロールを行う。

ASM：行列の和・差・積の計算を行う。

MINV2S：逆行列の計算を行う。〔FACOM SSL II〕

③MOMA

②の結果から、各ボディの重心位置・速度・加速度、関節でのモーメントの計算結果を表示し、映像化用データをファイルに書き出す。

次に入力データについて説明する。入力データは2つに分かれており、1つはプログラム①への入力である歩容に関するデータで、もう一方はプログラム②への入力である補償動作作成に関するデータである。ここでは、これらのデータの与え方とその事例を以下に示す。なお、歩容に関するデータは定常歩行を取り上げているため2歩分の左脚の関節角及び左腕の関節角のデータから、これとこれを1/2周期ずらしたパターンをもって1歩分の歩容とすることができる。これを支持期（支持脚：左脚）と揺動期（遊脚：右脚）に分けてプログラム②に与える。また、ZMP軌跡も歩容と同様な方法で補間をする。

(1)歩容に関するデータ (Fig. A-1)

Line 1 : 歩行動作時間

Line 2 : 揺動期の開始タイムステップ番号、
支持期の開始タイムステップ番号

Line 3 : コメント (関節角名)

Line 4 : サンプル点の個数

Line 5 : サンプル点のタイムステップ番号
(1秒30分割で1~31、但し1秒の場合)

Line 6 : サンプル点における角度 (rad.)

以下、Line 3 ~ Line 6 と同様にして他の関節角について記述する。記述の順番は β_1 、 β_2 、 β_4 である。

(2)補償動作作成に関するデータ (Fig. A-2)

Line 1 : コメント (データ名)

Line 2 : コメント

Line 3 : 両肘のZ軸回りの角度 (固定)、
左肘のY軸回りの角度 (固定)、
右肘のY軸回りの角度 (固定)

Line 4 : 時間ファクター (ここでは 1.0 一定)、
角度ファクター (歩容の入力データを定数倍する)、
腕の角度ファクター (腕の振り角の入力データを定数倍する)、

Line 5 : ZMP (X方向) のサンプル点の個数

Line 6 : ZMP (X方向) のサンプル点のタイムステップ番号

Line 7 : ZMP (X方向) のサンプル点における角度 (rad.)

Line 8 : ZMP (Y方向) のサンプル点の個数

Line 9 : ZMP (Y方向) のサンプル点のタイムステップ番号

Line 10 : ZMP (Y方向) のサンプル点における角度 (rad.)

Line 11 : 推定初期値 (θ^0 、 ϕ^0 、 $\dot{\theta}^0$ 、 $\dot{\phi}^0$)

以下、別の推定初期値を Line 11と同様に入力する。

2.0									
25	55								
BETA1									
6									
1	25	45	54	55	61				
-0.16		0.24		-0.50		-0.26	-0.26	-0.16	
BETA2									
6									
1	25	35	54	55	61				
-0.12		0.28		0.55		-0.26	-0.22	-0.12	
BETA4									
6									
1	15	30	45	60	61				
-0.02		-0.30		0.0		0.30	0.0	-0.02	

Fig. A-1 Input data of the gait

<<< H9B >>> FOR H9 STEADY

0.0	0.15	0.15							
1.0	0.9	1.0							
7									
1	5	10	15	20	25	31			
0.05		0.0		-0.03		0.01	0.05	0.12	0.12
2									
1	31								
0.0		0.0							
0.0			0.10				0.00		2.0
0.0			0.10				0.00		1.5
0.0			0.10				0.00		1.0
0.0			0.10				0.00		0.5
0.0			0.10				0.00		0.0
0.0			0.10				0.00		-0.5
0.0			0.10				0.00		-1.0
0.0			0.10				0.00		-1.5
0.0			0.10				0.00		-2.0

Fig. A-2 Input data for generating compensating motions

付録B

発進・停止歩行シミュレーションプログラムの概略

プログラム構成の大枠はこれまでのプログラムに倣い、3つのメインプログラムとそれらに引用されるサブプログラム、プログラムライブラリーからなる。プログラムの概略を以下に示す。

①G A I 3 A (ZMP, FORT77)

与えられた歩容及び腕の動きのサンプル点を補間し、1秒30分割した値をスムージングする。

AL I 3 : 与えられたサンプル点を線型補間する。

SML E 2 : 局所的最小二乗近似多項式によってサンプル点での平滑化を行う。次数は6次、観測点の個数は9個。〔FACOM SSL II〕

①' G A I 3 B (ZMPS, FORT77)

与えられた歩容及び腕の動きのサンプル点をスプライン関数によって補間する。

B I F 1 & B I C 1 : サンプル点をスプライン関数(m次)によって線型補間する。
両端における微係数入力型。

微係数の次数 $l = (m - 1) / 2$ 〔FACOM SSL II〕

②B I D 3 1 (共通)

①の結果と境界条件・安定領域等を入力として、その時のZMPを計算し、安定領域に入る補償動作を決定する。

DER 3 1 A : 与えられた基本歩行データからZMPを計算する。

L A X : 連立1次方程式を解く(クラウト法)。〔FACOM SSL II〕

R J E T R : 実係数高次代数方程式を解く(ジェンキンス・トラウプの方法)。

〔FACOM SSL II〕

③M O M 3 1 (共通)

②の結果から、各ボディの重心位置・速度・加速度、関節でのモーメントの計算結果を表示し、映像化用データをファイルに書き出す。

① "GA13C (定常歩行シミュレーションプログラム STEADY2C.FORT77)

与えられた歩容及び腕の動きのサンプル点をスプライン関数によって補間する。

BIF4 & BIC4 : サンプル点をスプライン関数 (m次) によって線型補間する。

周期関数の補間。

但し、サンプル点の個数 $n \geq m + 2$ [FACOM SSL II]

次に入力データについて説明する。入力データは2つに分かれており、1つはプログラム①への入力である歩容に関するデータで、もう一方はプログラム②への入力である補償動作作成に関するデータである。ここでは、これらのデータの与え方とその事例を以下に示す。

(1) 歩容に関するデータ (Fig. B-1)

Line 1 : コメント (データ名)

Line 2 : 歩行動作時間、
静止状態と動作との切り換えまでの時間

Line 3 : コメント (関節角名)

Line 4 : サンプル点の個数

Line 5 : サンプル点のタイムステップ番号
(1秒30分割で1~31、但し1秒の場合)

Line 6 : サンプル点における角度 (rad.)

Line 7 : 境界条件 :

初期角速度、

初期角加速度、

最終角速度、

最終角加速度

以下、Line 3 ~ Line 7 と同様にして他の関節角について記述する。記述の順番は β_{1L} 、 β_{1R} 、 β_{2L} 、 β_{2R} 、 β_{4L} 、 β_{4R} である。

(2) 補償動作作成に関するデータ (Fig. B-2)

Line 1 : コメント (データ名)

Line 2 : コメント

Line 3 : 発進か停止か (START or STOP)

- Line 4 : 両肘の Z 軸回りの角度 (固定)、
 左肘の Y 軸回りの角度 (固定)、
 右肘の Y 軸回りの角度 (固定)
- Line 5 : 時間ファクター (ここでは 1.0 一定)、
 角度ファクター (歩容の入力データを定数倍する)、
 腕の角度ファクター (腕の振り角の入力データを定数倍する)、
 係数サーベイの範囲に関するファクター
 (サーベイ範囲を定数倍する、ここでは 1.0 一定)、
 安定判別結果表示用ファクター
 (正 : X_{ZMP} , Y_{ZMP} の何方か一方が安定領域に在れば結果表示、
 負 : X_{ZMP} , Y_{ZMP} の両方とも安定領域に在れば結果表示)
- Line 6 : θ の入力係数に関するデータ [a_T , b_T , c_T , d_T , e_T , f_T]
 (Fig. B-2 の場合 c_T を入力係数としてサーベイすることを示す)
- Line 7 : ϕ の入力係数に関するデータ [a_P , b_P , c_P , d_P , e_P , f_P]
 (Fig. B-2 の場合 c_P を入力係数としてサーベイすることを示す)
- Line 8 : θ の係数サーベイの範囲 :
 初期値、
 最終値、
 ステップ、
 ϕ の係数サーベイの範囲 :
 初期値、
 最終値、
 ステップ
- Line 9 : X_{ZMP} の安定領域 (Min, Max)、
 Y_{ZMP} の安定領域 (Min, Max)
- Line 10 : 境界条件 : (括弧内は STOP の場合)
 θ の最終 (初期) 角度、
 θ の最終 (初期) 角速度、
 ϕ の最終 (初期) 角度、
 ϕ の最終 (初期) 角速度

(GAIT)		AH2							
2.0		1.0							
BETA1L									
8									
1	6	11	16	21	26	31	61		
0.0		0.0		0.0		0.0		0.0	0.0
0.0		0.0		-0.7045		-5.9158			
BETA1R									
9									
1	6	11	16	21	26	31	51	61	
0.0		0.0		0.0		0.0		0.0	0.0
-0.26		0.0		0.6947		-4.3190			
0.0		0.0							
BETA2L									
8									
1	6	11	16	21	26	31	61		
0.0		0.0		0.0		0.0		0.0	0.0
0.0		0.0		1.0238		3.3682			
BETA2R									
9									
1	6	11	16	21	26	31	43	61	
0.0		0.0		0.0		0.0		0.0	0.0
-0.2586		0.0		0.1991		8.8129			
0.0		0.0							
BETA4L									
9									
1	6	11	16	21	26	31	51	61	
0.0		0.0		0.0		0.0		0.0	0.0
-0.1444		0.0		0.7998		1.1992			
0.0		0.0							
BETA4R									
9									
1	6	11	16	21	26	31	51	61	
0.0		0.0		0.0		0.0		0.0	0.0
0.1446		0.0		-0.8001		-1.2052			
0.0		0.0							

Fig. B-1 Input data of the gait
[starting motion]

<<< HEICHI >>>	H2 LINK	FOR AH2	START	AH2B2
START				
0.0	0.0	0.0		
1.0	0.85	1.0	1.0	-5.0
0.0	0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0
-10.0	10.0	0.01	-10.0	10.0
-0.05	0.200	-0.090	0.080	
0.00000260		0.05467097		0.35687818
				1.01615985

Fig. B-2 Input data for generating compensating motions
[starting motion]