

修士論文

動力学演算パッケージを用いた
歩行ロボットシミュレーションとその応用

豊田 篤史

2002年 3月 15日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

豊田 篤史

審査委員： 小笠原 司 教授
杉本 謙二 教授
松本 吉央 助教授

動力学演算パッケージを用いた 歩行ロボットシミュレーションとその応用*

豊田 篤史

内容梗概

歩行ロボットの製作にあたって必要とされるのはハードウェアと制御ソフトウェアの開発である．多くの場合は実機動作を行う前に誤動作による危険を防ぐため，シミュレーションによる安全かつ効率的なソフトウェアの検証を行う．従来のシミュレーションにおいては歩行を二次元空間上で簡略化を行うことが多い．しかし三次元の現実世界での歩行ロボットとの誤差が大きくなってしまう．また，精密な三次元シミュレーションを行った場合では，その計算に時間がかかりすぎてしまって開発効率に悪影響を与えてしまう．そこで三次元空間において数値的に安定し，高速性を備えた動力学計算環境が歩行シミュレーションで必要となる．

本研究ではゲーム用高速動力学演算パッケージである MathEngine を歩行シミュレーションに用いることを提案する．構築したシミュレーション環境を歩行ロボットに適用し，多くのロボットにおいて歩行の基本となるアルゴリズムの検証や転倒安定性の確認を行う．そしてシミュレーション上で用いた歩行コントローラを四脚歩行ロボット TITAN-VIII に実装し，歩行実験を行うことで実機のソフトウェアへの適用の可能性を示した．

キーワード

歩行ロボット，三次元シミュレーション，動力学演算，動歩行，静歩行

* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT0051069, 2002 年 3 月 15 日.

Simulation of Walking Robot using Dynamical Calculation Package and Application*

Atsushi Toyoda

Abstract

Most important task to develop a walking robot is the design of hardware and control software. The software is usually verified safely and effectively using simulation system to prevent fatal malfunction. The previous simulation system often simplifies walking motion by restricting two-dimensional motions. However, there is a big difference between simulated motions and three dimensional motions in real world. If the system precisely simulate three dimensional motion, it is time consuming and inefficient to develop the control software. Therefore, dynamical computational environment is necessary for simulation of walking machines. The simulation must be numerically stable in the three-dimensional motions and the system have to be equipped with high speed computational power.

In this research, we propose a walking simulation environment using dynamical calculation package MathEngine. It enables high speed simulation of walking robot. The simulation environment is applied to a quadruped robot to verify the control algorithm and stability of walking motions. The walking controller generated by the simulator is equipped into a quadruped robot TITAN-VIII. During the walking environment, the applicability of the simulation system is demonstrated.

* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0051069, March 15, 2002.

Keywords:

walking robot, 3D simulation, , dynamical caluculation, statical walk, dynamicalwalk

目次

| | |
|--|-----------|
| 1. 序論 | 1 |
| 1.1 研究背景 | 1 |
| 1.2 従来の研究 | 2 |
| 1.3 本研究の意義と目的 | 2 |
| 1.4 本論文の構成 | 3 |
| 2. シミュレータの構成 | 4 |
| 2.1 MathEngine について | 4 |
| 2.1.1 MathEngine Dynamics Toolkit | 6 |
| 2.1.2 MathEngine Collision Toolkit | 8 |
| 2.2 歩行ロボットシミュレータの構成 | 9 |
| 2.3 歩行ロボットシミュレータの応用 | 14 |
| 3. 歩行シミュレーション | 15 |
| 3.1 静歩行シミュレーション | 16 |
| 3.1.1 静的安定性評価基準 | 17 |
| 3.1.2 間歇クロール歩容 | 18 |
| 3.1.3 パラメータ設定 | 19 |
| 3.1.4 静歩行シミュレーション結果 | 21 |
| 3.2 動歩行シミュレーション | 24 |
| 3.2.1 トロット歩容 | 25 |
| 3.2.2 胴体重心の左右揺動補償軌道 | 27 |
| 3.2.3 動歩行シミュレーションの結果 | 34 |
| 3.3 歩行シミュレーションの考察 | 38 |
| 4. 実験 | 39 |
| 4.1 実験システム | 39 |
| 4.2 静歩行実験 | 41 |
| 4.3 実験の考察 | 43 |

| | |
|---------------------|----|
| 5. 結論 | 45 |
| 5.1 結論 | 45 |
| 5.2 今後の課題 | 45 |
| 謝辞 | 46 |
| 参考文献 | 47 |

図 目 次

| | | |
|----|--------------------------------------|----|
| 1 | MathEngine の構成 | 5 |
| 2 | Collision Toolkit のアーキテクチャ | 8 |
| 3 | シミュレーションの構成 | 10 |
| 4 | Model data part の構成 | 11 |
| 5 | Robot simulator part の構成 | 12 |
| 6 | シミュレーションの流れ | 13 |
| 7 | 安定余裕 | 17 |
| 8 | 間歇クロール歩容の動き | 18 |
| 9 | パラメータ変化による比較 | 20 |
| 10 | 静歩行シミュレーション | 22 |
| 11 | 縦安定余裕値 | 23 |
| 12 | 重心の軌道 | 23 |
| 13 | 様々な動歩行 | 25 |
| 14 | トロット歩容 | 26 |
| 15 | 支持脚対角線パターン図 | 27 |
| 16 | 左右方向 (y 軸) の加速度の推移 | 28 |
| 17 | 本体の重心軌道計画 | 30 |
| 18 | 動歩行シミュレーション | 35 |
| 19 | トロット歩容における胴体の Pitch 角 | 36 |
| 20 | 重心の左右揺動補償軌道 | 36 |
| 21 | 右前脚と左後脚の第一関節の角度推移 | 37 |
| 22 | TITAN-VIII | 39 |
| 23 | 各脚の接地状況 | 41 |
| 24 | 静歩行実験 | 42 |
| 25 | 重心の軌道 | 44 |
| 26 | 縦安定余裕 | 44 |

表 目 次

| | | |
|---|----------------------------|----|
| 1 | TITAN-VIII の基本仕様 | 40 |
|---|----------------------------|----|

1. 序論

1.1 研究背景

近年、ロボットの歩行をテーマにした研究は盛んに行われている．ロボットの制御ソフトウェアの開発過程において、多くの場合は実機によるソフトウェアの検証を行う前に、誤動作などによる危険を防ぐためにシミュレーションによる検証が行われる．三次元運動を行うロボット動作のシミュレーションを開発し、実用に耐えられるようなレベルにまで熟成させられた研究はいくつか試みられているものの非常に困難であり DADS や ADAMS といった商用の機構解析ソフトウェアが用いられることが多い [1]．これらのソフトウェアにおいては動力学演算結果の正確さを非常に重要視している．しかしながら複雑な環境との接触を含むロボットシミュレーションを行う場合、計算量が膨大となり、結果として実時間に対して数倍から数百倍といった計算時間がかかってしまう．

一方で SONY の PlayStation2 を始めとするゲーム機の性能向上にともなって、その提供されるゲームコンテンツに対する要求も厳しくなっており、三次元世界におけるキャラクタのリアルな振る舞いを含んだソフトウェア開発は年々向上している．このような背景を元にこれらのゲーム機上でリアルタイムの動力学演算を行うための開発環境が MathEngine[2] や Havok[3] などからリリースされている．これらのパッケージでは計算の正確性よりは実時間性を重視しているために、物理的な精密性には欠ける点がある．しかし制御アルゴリズムの開発の初期段階や行動制御ソフトウェアの開発など、正確性を多少落としても高速性が重視される局面では有効と考えられる．高速なシミュレータを用いて解の範囲を絞った後に低速ながらも精密なシミュレータに切り替えるといったような利用法も可能である．これらの観点からゲーム用動力学演算パッケージを用いたシミュレーション環境は非常に有用であると考えられる．

1.2 従来の研究

従来の多くの歩行ロボットのシミュレーションは複雑な動力学や運動方程式を解くことでモデルを作り、動作を計算するように設計されている。様々な状況においても数値的に安定した動作を行うレベルにまで作り上げるには、それ自体が多くの研究課題 [4][5] を含んだものであり、シミュレーションの開発に時間がかかりすぎてしまう。また、数値計算的にも三次元空間におけるシミュレーションは精度が高ければ高いほど計算時間自体も長くなり、結果として開発効率に大きな悪影響を及ぼしてしまう。床との衝突を考慮に入れた2足歩行シミュレーションの研究 [6][7] ではロボットのリンク数が増えるほど、計算量は増大してしまい、10秒の2足歩行シミュレーションにも10分以上かかってしまう。これらは衝突計算に要する時間がかかりかかってしまうためである。また、近年では「人間協調・共存型ロボットシステム」というテーマの研究開発プロジェクトにおいて、ヒューマノイドロボットのシミュレーションに関する研究が多く発表されている [8][9]。これらの研究では人間型ロボットのシミュレーションを基準としているため、複雑な構造になってしまっている。

1.3 本研究の意義と目的

前述のような現状から、本研究では MathEngine 社の動力学演算パッケージである MathEngine Toolkits を歩行ロボットシミュレーションの構築に用いた。このパッケージは動力学演算や衝突計算についての高速化が図られており、様々な状況に対するロボットの反応を動力学に基づいて計算する負担を軽減できると思われる。多くのロボットの歩行の基礎である歩行アルゴリズムの検証や転倒安定性の評価などを構築したシミュレータ上で試した後、その設計したコントローラを歩行ロボットに実装することでこのシミュレータの有効性を考察する。またシミュレータの構造的にも、多くの歩行ロボットにも応用できるような自由度を持たせるのが本研究の目的である。

1.4 本論文の構成

本論文では5章から構成される．以下，各章の概要について述べる．
第1章では，研究の背景と目的，および本論文の構成について述べる．第2章では歩行ロボットシミュレーションの構築に用いた MathEngine Toolkits について説明し，シミュレーション本体の構造について述べる．第3章ではロボットにおいて用いられる歩行アルゴリズムについて説明し，それを本シミュレーション上で適用した結果を示す．第4章では前章で検証した歩行アルゴリズムを実機に搭載し，歩行実験を行った結果を報告する．第5章では，結論および今後の課題について述べる．

2. シミュレータの構成

本章ではまずシミュレーション制作においてコアとなった MathEngine Toolkits について 2.1 節で説明する．この演算パッケージは動力学演算や衝突計算だけでなくシミュレーション世界の可視化を行うのに必要な機能も含まれており，高速シミュレーション環境を構築するのに適している．そこで本研究では歩行ロボットのシミュレータにこの MathEngine Toolkit を用いて構築を行った．その概要について 2.2 節で説明した後，2.3 節でその応用について述べる．

2.1 MathEngine について

このツールキットはアメリカの MathEngine 社によって開発されており，三次元空間における様々なシミュレーションの開発をサポートできるように設計されている．この MathEngine のシステムは 2 つのツールキットがお互いに協力して働くように意図されており，一つは Dynamics Toolkit と呼ばれ，主に剛体間の動力学シミュレーションを行う．多数の拘束や接触摩擦のタイプも含まれており，様々なレベルの力学シミュレーションも可能となっている．

また，もう一つの Collision Toolkit は，物体間の干渉検査や衝突計算を行うツールキットとして提供されている．

この 2 つのツールキット同士を仲介し，シミュレーション環境を構築をサポートするのが McdDtBridge と呼ばれる部分である．シミュレーション環境の可視化やパラメータの設定など開発のサポートに必要な機能も持ち合わせている．

全体的な構成は図 1 で示す．

本節ではツールキットの主要部分である Dynamics Toolkit を 2.1.1 節で説明する．またもう一つの主要部分である Collision Toolkit は 2.1.2 節で説明する．

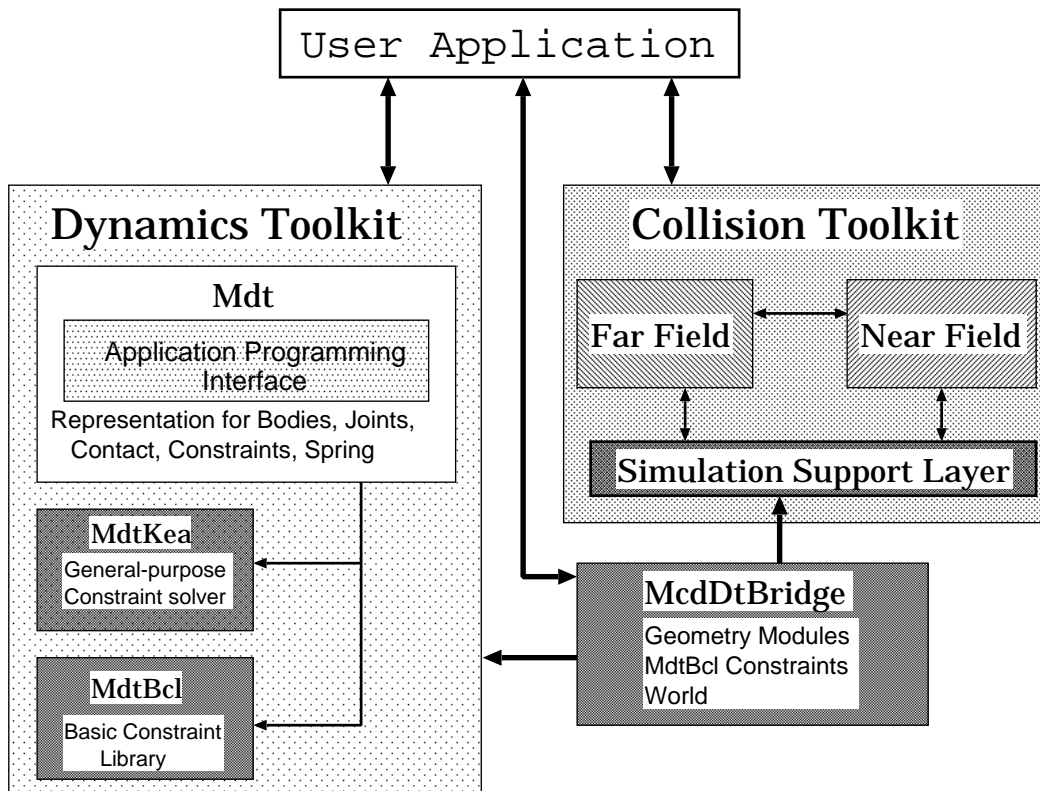


図 1 MathEngine の構成

2.1.1 MathEngine Dynamics Toolkit

MathEngine Dynamics Toolkit(Mdt) は主に剛体の動力学シミュレートを行うツールキットであり，剛体の運動，拘束，接触などを受け持っている．他の物体との拘束がある場合，連結された剛体のツリーとして積分を行っており，それらの構造は必要なときに追加・削除することができる．また連結の種類はボール・ソケット・ヒンジ等多種の関節を定義することができる．関節は発生できる力に制限のあるバネとしてシミュレートされており，これによってバネのシミュレーション時に発生しやすい不安定性を解消している．以下では Dynamics Toolkit に含まれているコンポーネントについて述べる．

- Mdt Library

Mdt Library は様々な動力的なシミュレーションを行うため，多様な動きに対するシミュレートを計算する関数を提供している．主なライブラリとしては次のようなものが用意されている．

- MdtBody シミュレーション上における剛体の運動を主に計算する．質量や体積，位置や速度，加速度といったものを設定でき，そのシミュレーション結果の位置，速度，加速度などの計算結果を出力できる．また外力による反応や慣性などの働きも動力学に基づいてシミュレートしている．
- MdtHinge シミュレーション上の拘束に関するライブラリの一つ．剛体同士のヒンジ連結を設定する．これを用いることでロボットのような構造物の作成も可能としている．
- MdtContact シミュレーション上の接触に関するライブラリ．物体同士の接触判定，接触位置などを検出する．
- MdtWorld シミュレーション環境構築に関するライブラリ．剛体の数や剛体間の拘束条件数，メモリーの割り当てなどを設定できる．

他にも多くのライブラリが用意されており，シミュレーション作成をサポートしている．

- MdtKea Library

MdtKea Library は剛体の計算をサポートする関数を主に用意している . Mdt Library のような複数の剛体を考慮に入れた計算ではなく , 単体の剛体の動力学計算を行う場合に用いられ , 高速な演算を可能としている .

- MdtBcl Library

MdtBcl Library は拘束条件が 1 つのみの場合 , 高速な演算を必要とする場合に用いられる .

2.1.2 MathEngine Collision Toolkit

MathEngine Collision Toolkit(Mcd) は三次元環境内における物体間の衝突や干渉検査を行うツールキットである．形状データとしては球，直方体，平面，円柱といったプリミティブ形状のものが用意されている．距離の離れたオブジェクト間の検査を省略する干渉カリングやメモリ上のレイアウト等の工夫によってかなりの高速化が計られているため，衝突計算の短縮を可能としている．以下において MathEngine Collision Toolkits に含まれるコンポーネントについて述べる．

Near Field と Far Field

Collision Toolkit においては3次元上での距離から Near Field と Far Field との二つのモジュールに分けられている．Far Field Module は個々のモデルの詳細な形を無視し，広域でのオブジェクト判別を行っている．Near Field Module は個々のモデルの詳細なジオメトリックな形から衝突の有無を判別している．この2つのモジュールを統合し振り分けを行っている部分が Simulation Support Layer と呼ばれる階層であり，このシステムを利用することで距離の離れている物体の衝突計算を省略できる．Collision Toolkits の構成図は以下に示す．

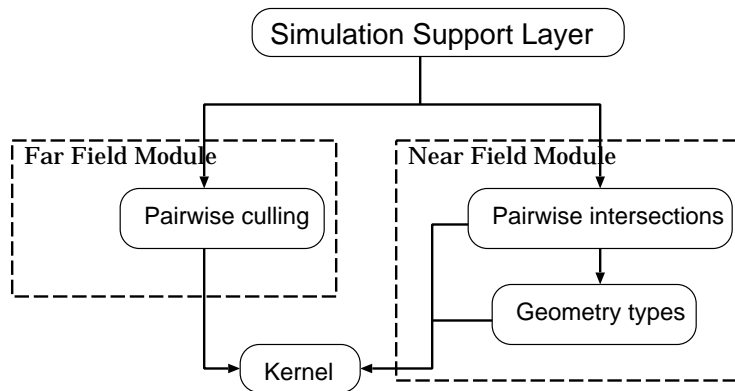


図 2 Collision Toolkit のアーキテクチャ

2.2 歩行ロボットシミュレータの構成

本研究ではこの動力学演算パッケージ MathEngine Toolkit を歩行ロボットのシミュレータに組み込んだ．これは MathEngine が三次元空間上でのシミュレーションにおいて動力的な演算部分を自動的に計算を行う機能を持つためである．従って，従来のように手動で運動方程式を解かなければならない負担をかなり軽減することが可能である．シミュレーション自体の精密性は高速性を優先させるため，多少落ちてしまう．しかし初期の歩行アルゴリズムの確認やパラメータの絞り込みなどには効果的であり，歩行ロボットにも適用可能と思われる．

シミュレータの構成は図3のように構築を行った．大きく分けると Model data , Robot simulator , Walking program と3つに区分される．シミュレーションの出力はディスプレイに写され，視覚的な検証を可能としている．以下にこのシステムの手順について説明する

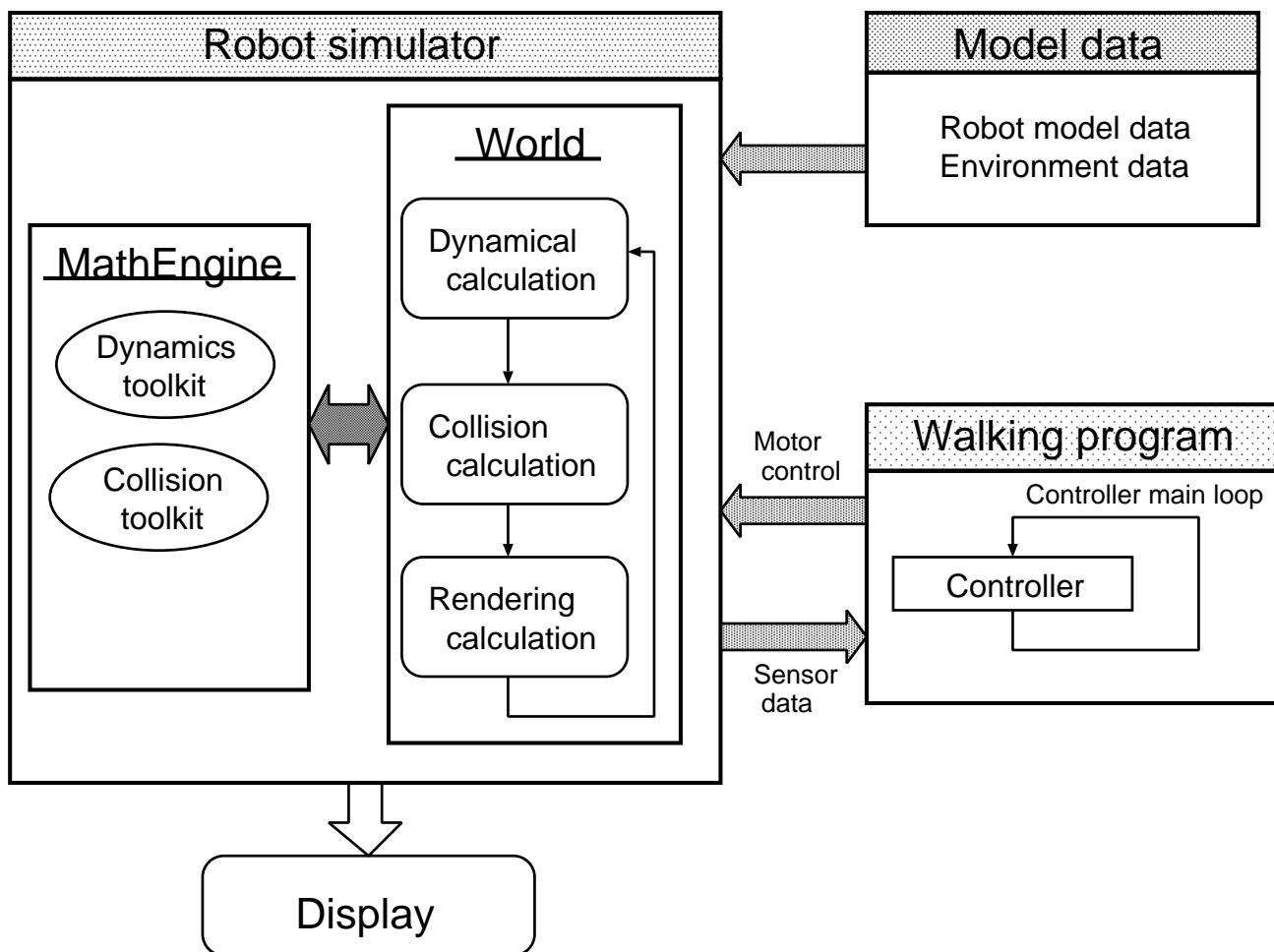


図 3 シミュレーションの構成

- Model data

まず Model data においてはシミュレーションにおけるロボットのデータとして各関節機構の構成やそれぞれのリンクの質量，大きさ，形状といったパラメータについての情報などを配列データで記述している．またシミュレーションにおける外環境についてのデータも含まれており，障害物や傾斜などの路面状況を作り出すことも可能としている．これらの情報を元としてロボットや周りの環境の構築に反映されるように設計を行った．図4はロボットモデルとして四脚歩行ロボット TITAN-VIII[10] のデータを入力した場合とヒューマノイドロボット PINO[11] のデータを入力した場合とを比較表示している．ロボットのモデルデータは Robot simulator の World 内で Robot frame として形成する．周囲環境は Environment data から World 内に構築する．ここで World はシミュレータ内に形成される三次元空間であり，Robot frame は1つ以上の関節を持つオブジェクトである．

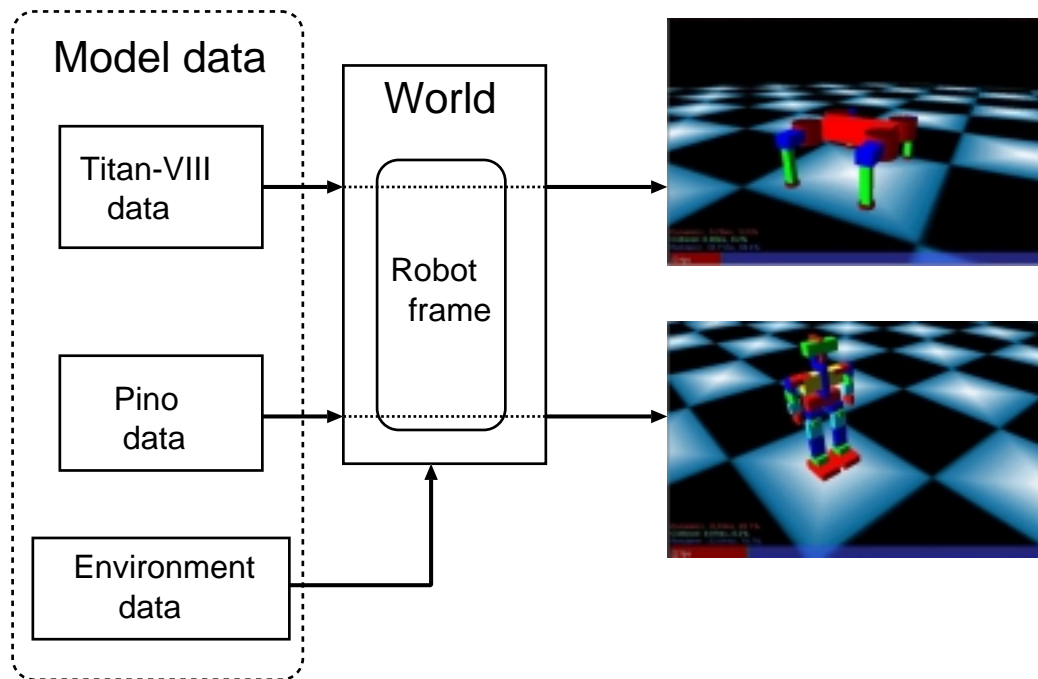


図 4 Model data part の構成

- Robot simulator

Robot simulator の主な役割はシミュレータ内に形成される三次元空間 World を構築することである．World 内ではロボットのような一つ以上の関節を持った Robot frame が定義され，周囲環境が形成される．シミュレーションにおける計算時間は動力学計算を行っている Dynamics time，衝突判定を行っている Collision time，モニタに表示するシーン作成を行う Rendering time とに区切っており，これらの合計の 1 ステップあたり 20ms でシミュレーションを行っている．

Robot simulator においては他の機能も含んでいる．その一つはロボットの各関節に用いられるモータの制御シミュレーションを行っている Motor Simulation である．目標値を Walking program 内の Controller から指令されるとモータの駆動制御則に従って計算結果を出力し，シミュレータ上で関節のトルク駆動を行う．もう一つは内部や外部からの情報をデータとして検出する Sensor simulation である．これは重心位置や関節の現在角度などの検出を行っており，そのデータを Controller へ送っている．

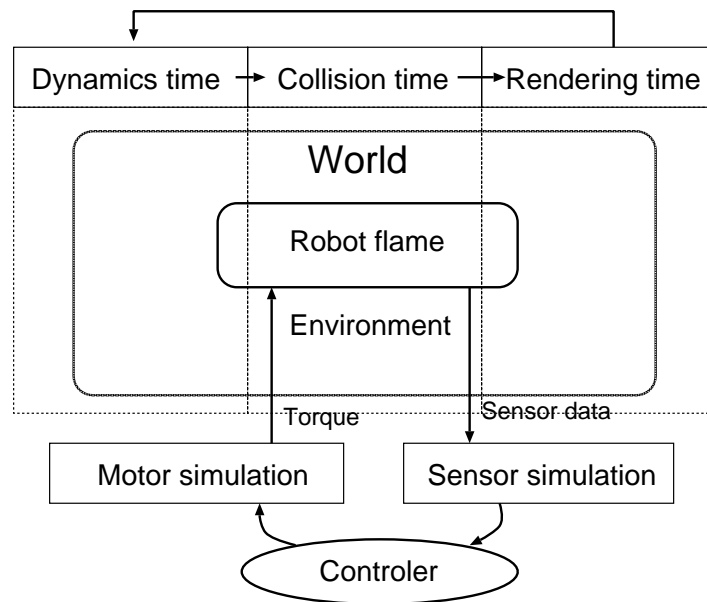


図 5 Robot simulator part の構成

ロボットがシミュレータ上で動くのは Motor simulation によって出力されたトルクが各関節に加わり，それぞれのリンクに作用するためである．また，その結果を Sensor simulation でコントローラにフィードバックすることで歩行などの制御計算を行う．このようなパターンを繰り返すことでロボットの歩行シミュレーションを可能としている．

- Walking program

Walking program はその歩行シミュレーションにおける歩行アルゴリズムを決定し，それを元に Controller 部の構築をおこなう．シミュレーションのスタート指令が入力されると Walking program 内の Controller は歩行アルゴリズムに従った運動指令を Robot simulator に送る．この2つのパートは図6のようにそれぞれ独立したループを回っている．時間管理は積分計算を行う Robot simulator 側で行っており，Controller 側の時間刻みは Robot Simulator の時間に同調する必要がある．

本シミュレータの全体的な流れは，Model data からの情報を元にシミュレーション環境を構築した後，図6のように Robot simulator と Walking program とでそれぞれ，動力学演算と制御計算を並列に実行させる．

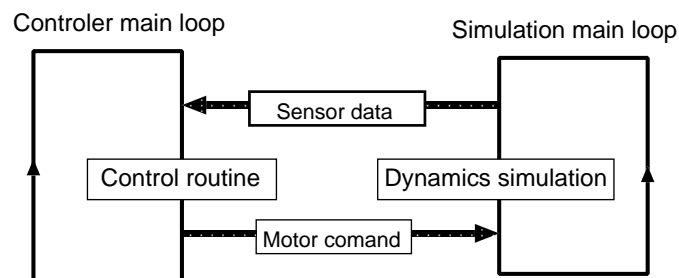


図 6 シミュレーションの流れ

2.3 歩行ロボットシミュレータの応用

以上のような並列計算を行う構成により，実機における動作実験とシミュレーション世界での動作確認をシームレスに切り替えることが可能となっている．本研究では完全な切り替えまでには至らなかったが，実機コントローラ的设计のベースに持ってくることは実現した．

3. 歩行シミュレーション

四脚歩行ロボット [12] は以下のような 2 つの歩行方法がある .

- 静歩行

不整地であっても静止した安定な脚場を持ち , 常に静的安定性を保持しつつ移動する歩行 .

- 動歩行

動的安定性を保持する期間を持ちつつ , 振動的であるがより高速な移動をおこなう歩行 . [13]

これらの歩行機能を自在に駆使することで四脚歩行ロボットは今後多くの分野で利用できると思われる .

そこで本研究では 2 章で述べた歩行ロボットシミュレーション環境内で静歩行と動歩行の二つの歩行方法を四脚歩行ロボットに組み入れ , 歩行コントローラのシミュレーションをおこなう . そしてシミュレーション内で歩行アルゴリズムやパラメータの調整をおこなった後に実機への組み込みを行う .

3.1 静歩行シミュレーション

本研究ではまず，四脚歩行ロボットの静歩行シミュレーションを行った．2章で述べたシミュレーション環境上で四脚歩行ロボットのモデルと，その歩行アルゴリズムをコントローラにのせた制御プログラムを組み込んでいる．このシミュレーションでは，歩行中の重心位置や脚の接地状況，本体の姿勢などが数値だけではなくモニター出力されることにより直感的な歩行アルゴリズムの新しい構築に用いることが可能と考えられる．まず 3.1.1 節では静歩行時の転倒安定性を評価する基準について述べ [14]，3.1.2 節では静歩行シミュレーションに用いた歩行アルゴリズムについて説明する．3.1.3 節ではシミュレーションによるパラメータの設定方法について説明する．そして 3.1.4 節ではこのアルゴリズムによる静歩行シミュレーションの結果をのせる．

3.1.1 静的安定性評価基準

静的安定性を最大化するような間歇クロール歩容の生成法を検討するにあたって、まず始めに静的安定性についての評価基準を明確化することが必要である。本研究ではまず、静的安定性を保つことを第一の条件としてシミュレーション上で評価することから、四脚静歩行において測度として多く用いられる安定余裕を評価基準とする。この値は重心と支持脚多角形の関係で与えられ、重心の高さや脚のコンプライアンスなどによっても支配される。しかし、ここでは解析を容易にするため縦安定余裕を測定する。すなわち、重心推進軌道上で重心から支持脚多角形の辺までの最短距離をその測度の第一近似として用いる。

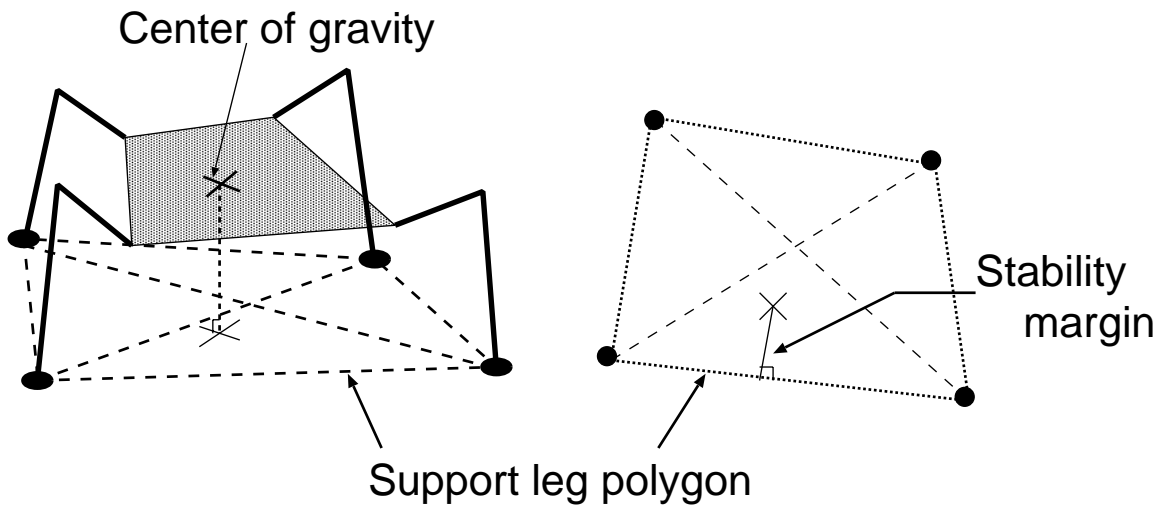


図 7 安定余裕

3.1.2 間歇クローल歩容

安定性を保持しながら最大限滑らかな移動を可能とする四脚歩行ロボットのための歩容として本研究では塚越ら [15] の間歇クローल歩容を用いた．この歩容は四脚歩行運動時の静的安定性をできる限り最大化するような歩容である．具体的には図 8 に示すように，まず (1)，(2) のように胴体重心を支持脚三角形の中のできるだけ安定性の高い位置に静止させる．その状態で残りの一脚を復帰運動させて，目標位置に接地する．そして四脚支持状態となった (3) の状態で重心移動をおこない，次の新しい支持脚三角形の内部に重心を移動させる．その後，後脚の復帰，前脚の復帰をおこなうことで (6) のようにジグザクな重心移動軌道をたどりながら推進をおこなうものである．

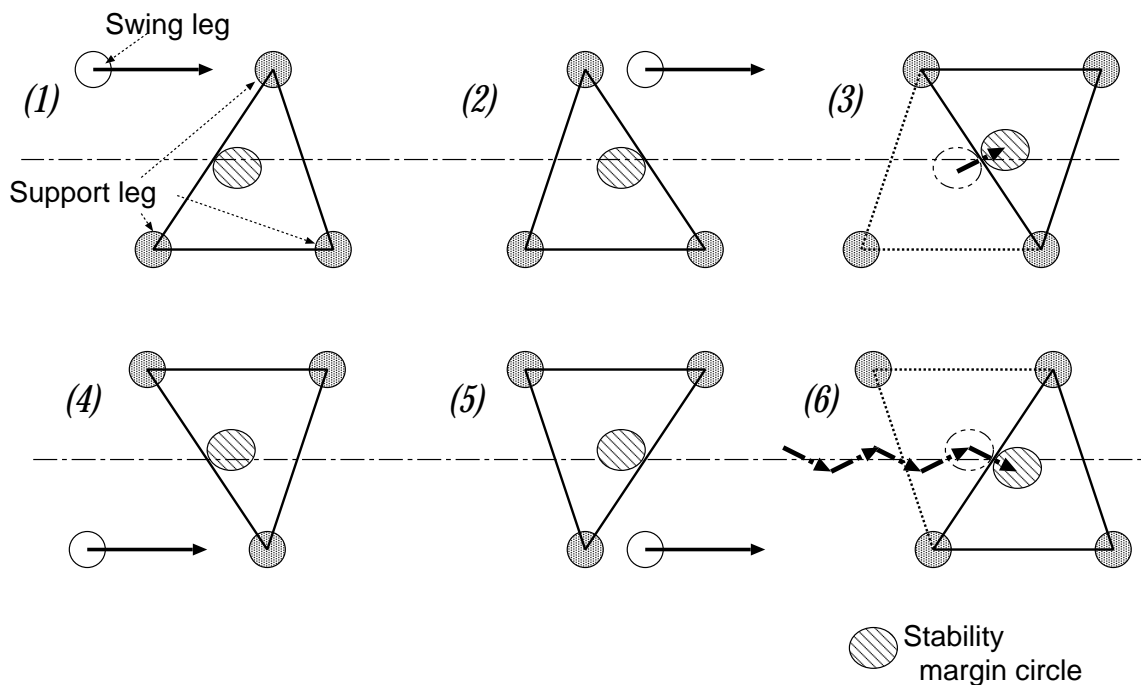


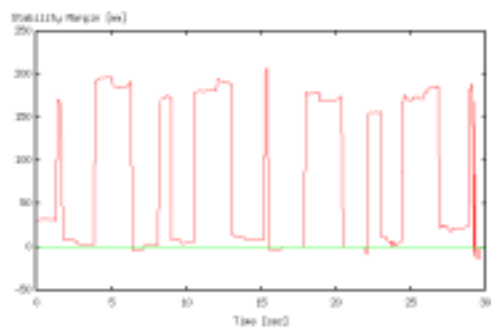
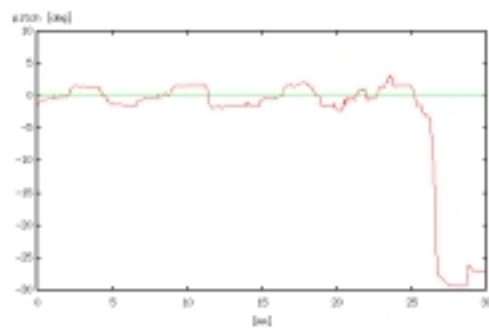
図 8 間歇クロール歩容の動き

ここで重心移動を行う際，3.1.1 節で述べた安定余裕を持てるように予想安定余裕値をあらかじめ設定し，重心を中心にして予想安定余裕値を半径とする円を描く．この円が接地脚多角形内に存在する限り，ロボットの本体の転倒する危険性

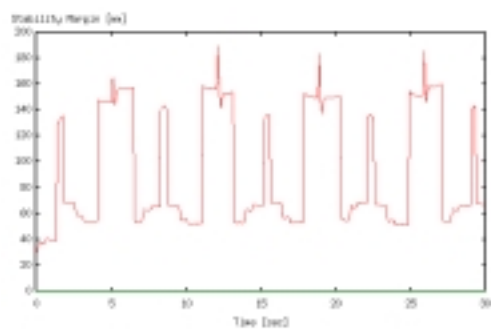
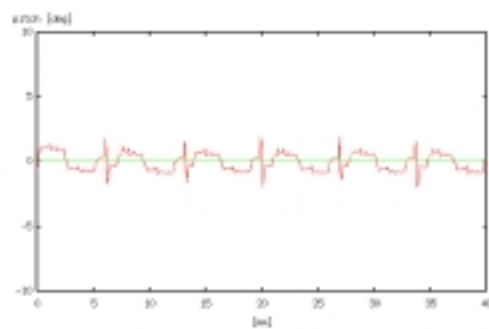
は低減する．すなわち接地脚対角線を越えようとする(3)，(6)の重心が移動する距離は予想安定余裕値によって変動する．

3.1.3 パラメータ設定

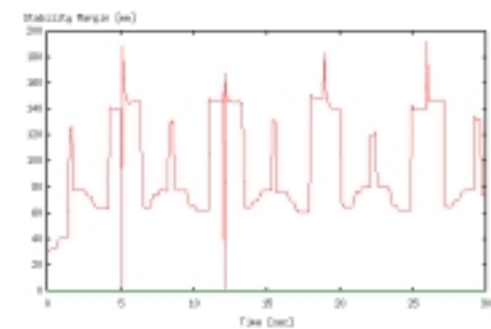
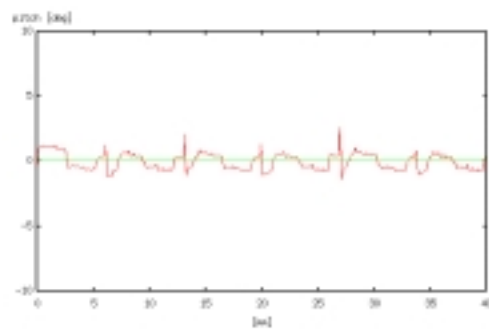
3.1.2節で述べたように，間歇クロール歩容では重心を中心にとった予想安定余裕値による円の半径を変化させることでその移動距離や安定性が変動する．このパラメータ設定を実際のロボットで行うのは転倒する危険性が大きく効率も悪い．そこでこのパラメータ設定を変化させるのに，シミュレータにおける視認した結果や縦安定余裕値などの数値的結果を参考にし，より安定した歩行ができるように調整を行った．その結果を図9に示す．それぞれの半径の場合の胴体のPitch角と縦安定余裕値をグラフで表している．図9-(1)のように半径が小さい場合，その重心の安定余裕値も小さくなり転倒してしまう．図9-(3)のように半径が大きすぎると数値的には安定しているが，脚が大きく動くことで前後の脚が接触してしまい不安定な部分が発生してしまう．そこでシミュレータ上で調整した結果として，図9-(2)の半径が70mmを実際のロボットにおける予想安定余裕の設定値とした．



(1) 半径 10mm



(2) 半径 70mm



Pitch角 (3) 半径 80mm 縦安定余裕値

図 9 パラメータ変化による比較

3.1.4 静歩行シミュレーション結果

3.1.2 節で述べたアルゴリズムを用い、パラメータ値を設定して静歩行シミュレーションを行った。歩を行空間は障害物のない平地と仮定し、初期姿勢をとった後に間歇クロール歩容をとりつつ歩行する様子を図 10 に示す。また、その結果として安定余裕の数値変動図 11 に示す。予想安定値による円の半径を 70mm で設定しており、全体的には縦安定余裕は 50mm 以上を保ったまま歩行することができた。初期に縦安定余裕値は低いのはパラメータ比較時の設定によるもので初期姿勢によるものである。また実際に移動した本体の重心位置シミュレーション上での軌道を図 12 に示す。このグラフにおいて、軌道の折り返し点は脚が浮いている状態のため、重心が多少ぶれているのがわかる。しかし本体を動かさず、安定した状態で静止しているので転倒する危険は少ない。

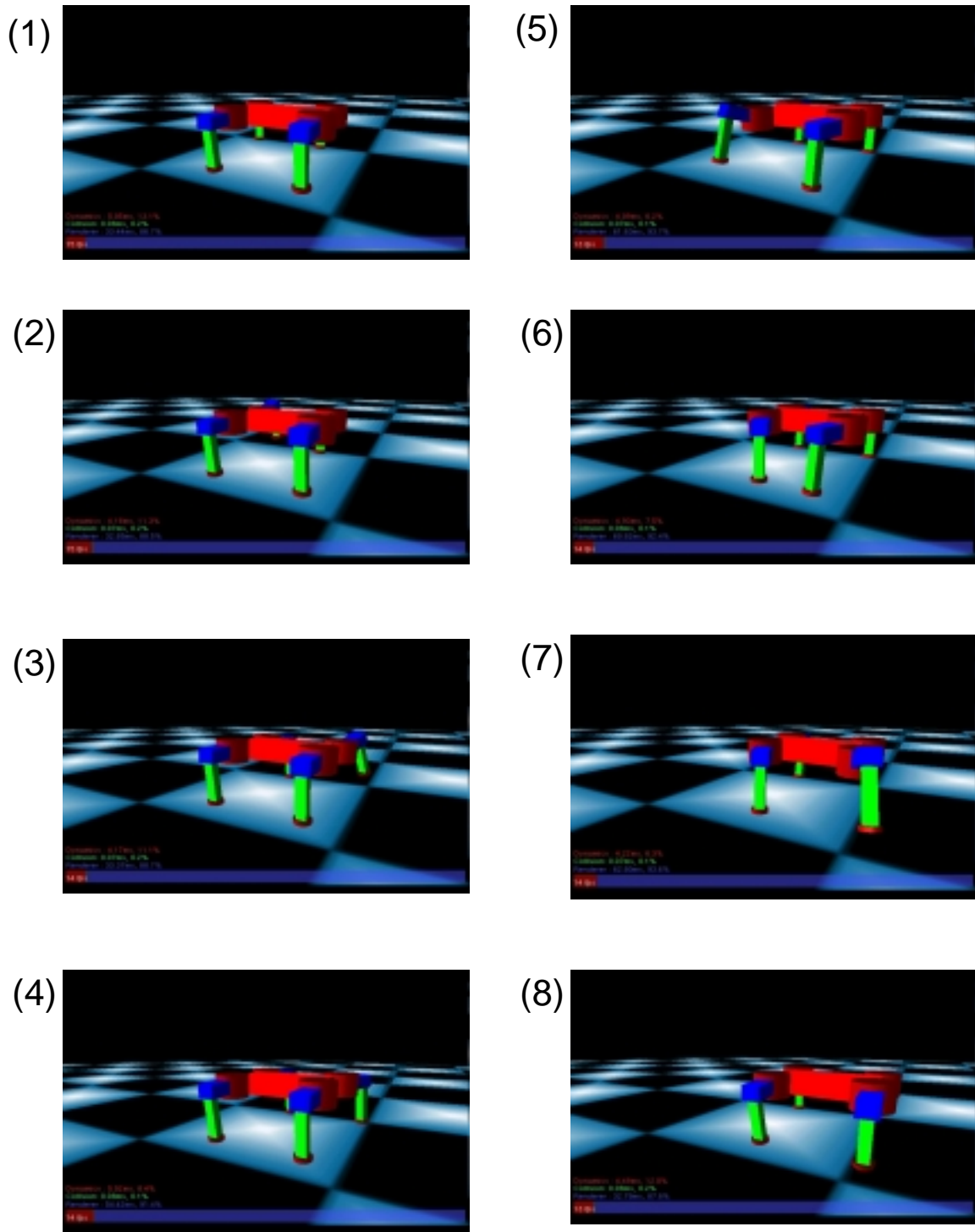


図 10 静歩行シミュレーション

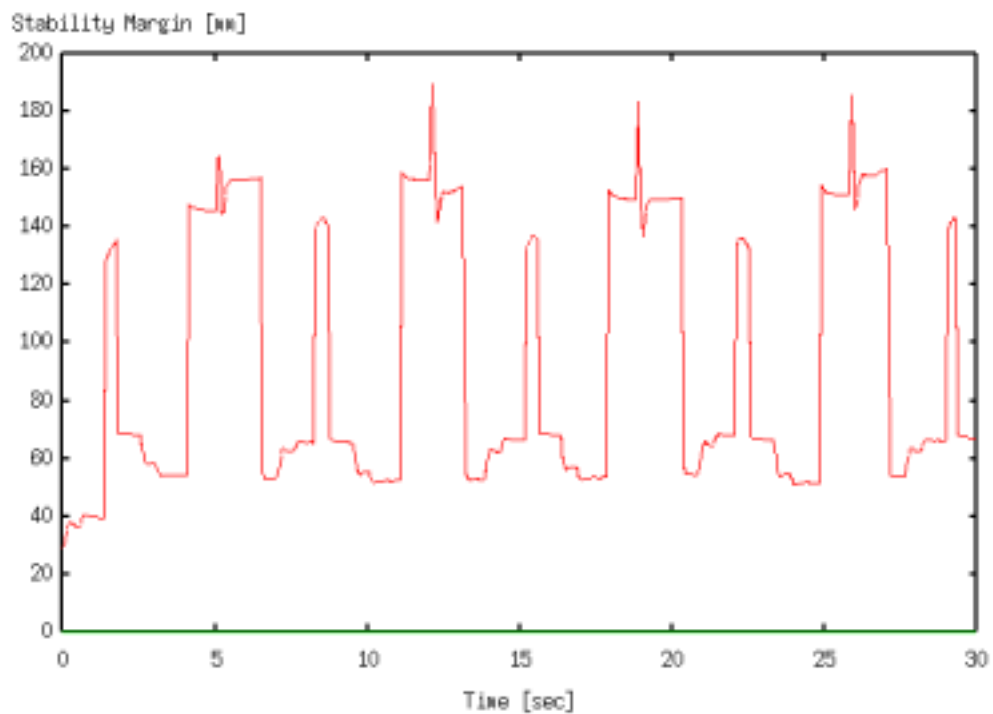


図 11 縦安定余裕値

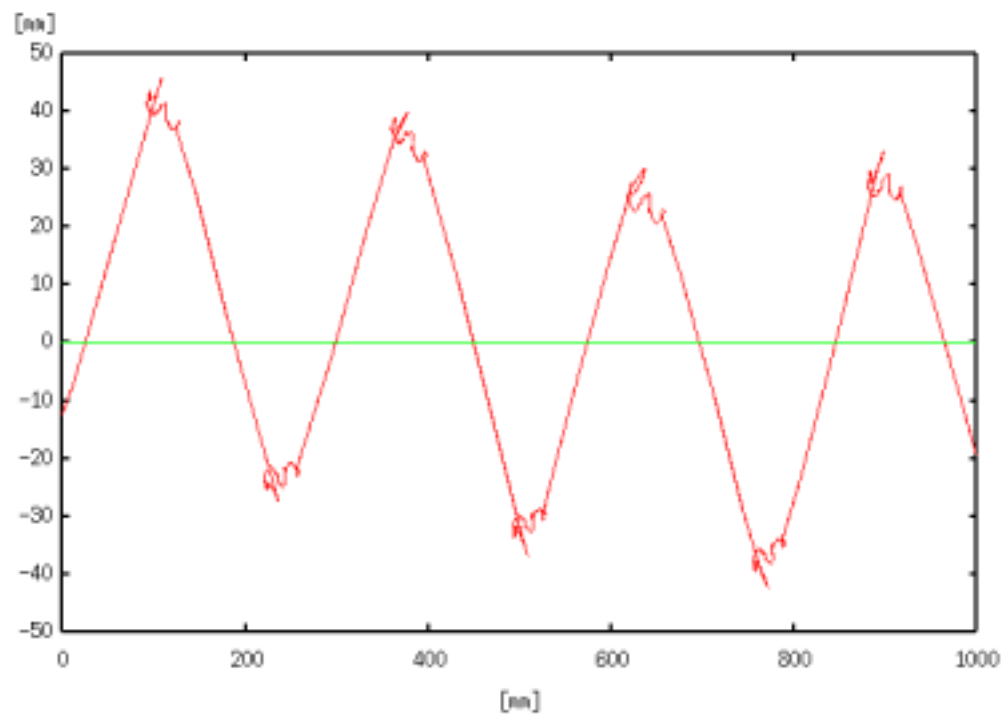


図 12 重心の軌道

3.2 動歩行シミュレーション

3.1 節では静歩行について述べたが，静的安定性を保ったままの歩行のみではロボットの歩行の検証には不十分である．

そこで本節では動力学的安定性を保ちながら歩行する動歩行シミュレーションを行う．動歩行の一つであるトロット歩容に重心軌道のフィードフォワード的な制御を行う左右揺動補償軌道を組み込み動歩行シミュレーションを行った．まず，3.2.1 節ではトロット歩容 [16] [17] の説明を行い，3.2.2 節では胴体重心の左右揺動補償について述べる．また 3.2.3 節ではそのシミュレーション結果をのせる．

3.2.1 トロット歩容

一般に四脚歩行ロボットの移動速度は，脚の最大振り速度を V ，デューティ比を β とするとき，以下のように表せる．

$$V_G = \frac{1-\beta}{\beta} V \quad (1)$$

ただし，ここでは脚の加減速時間や上下動に必要とする時間は無視している．一方，歩行時の安定余裕を高めるためにはデューティ比 β は大きくしておき脚をできる限り支持脚状態に保つべきである．そのため，ある移動速度を得るためには1式から脚のサイクル運動の遊脚相を最大振り速度に設定し，デューティ比をできるだけ大きな値に保つようにして歩行すればよい．

β が1であると静止状態を示す． β が1から減少していくと， $\beta = 0.5$ までは移動速度が増大してゆき $\beta = 0.5$ で最大速度 $V_G = V$ が得られる． β を0.5より小さくしても移動速度 V_G の最大値は V より大きくできない．そのため以降の歩容決定は $1 \geq \beta \geq 0.5$ の範囲でデューティ比を調整するものとする．

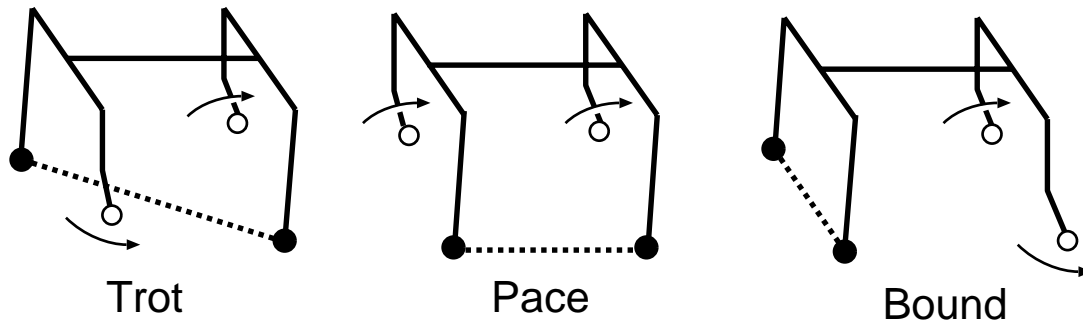


図 13 様々な動歩行

$0.75 > \beta \geq 0.5$ の範囲では二脚支持期が必ず生じ動的歩行が必要となる．二脚支持状態での動的歩容は6組，左右対称性を考えると図13のような3組が存在する．これらはトロット歩容，ペース歩容，バウンド歩容と呼ばれている．これらのうちトロット歩容は以下の示す特徴により，四脚歩行機械の動的歩行の実現に特に重要であると考えられる．

トロット歩容の第一の特徴は動的歩容を行うときの安定性の高さである．図14のようなトロット歩容においては歩行機械全体は支持脚 r, \bar{f} を結ぶ直線（以降支

持脚対角線と呼ぶ)を回転軸とする倒立振子になる．このため転倒のモードは，支持脚対角線まわりの回転モードのみに束縛される．一方，遊脚 \bar{f} あるいは r は地表面に沿って復帰動作を行っている．そのためたとえ転倒が生じたとしても，どちらかの遊脚の接地するタイミングがずれるだけで完全に転倒してしまうことを防止できる．またトロット歩容は動的歩行であるが，二脚支持相を切り替える一瞬四脚支持相が形成される．そのため発生した姿勢の乱れはこのとき修正できる．

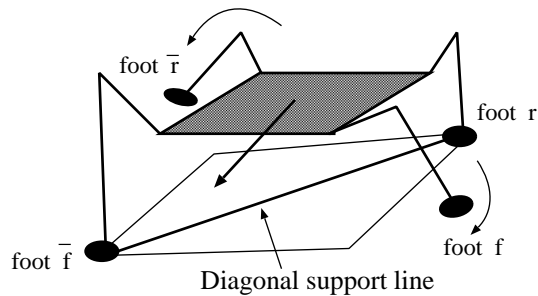


図 14 トロット歩容

この特性はトロット歩容による動的歩行の実現を予想外に単純化する．なぜなら，ある軌道運動を実現するとき，まずフィードフォワード的に胴体の軌道運動を補償する．これだけでも歩行は十分持続できると考えられるが，もし地表の凹凸などの影響で歩行パターンが大きく乱れた場合には，それを姿勢センサや足先の力センサで検出しフィードバック的に修正する．このように二段階の制御構成も可能とする．

ただし本論文ではすべての足裏の面積は考えず点接地と仮定している．また歩行する環境も平面としている．

3.2.2 胴体重心の左右揺動補償軌道

本研究では広瀬らの研究 [18] における左右揺動補償を用いた．これはフィードフォワード的に胴体重心軌道の補償を行うものである．

x 軸を指令軌道とするような $\beta = 0.5$ のトロット歩容を考えると，このとき支持脚対角線が地表に残すパターンは図 15 のように斜めに交差を繰り返すものとなる．このようなトロット歩容の場合，転倒することなく単に歩行を続けることが目的であれば，胴体重心の動的制御は必ずしも必要としない．

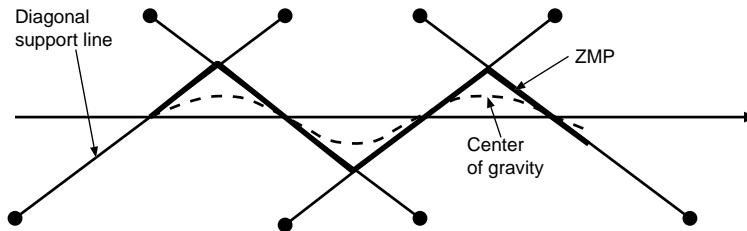


図 15 支持脚対角線パターン図

ただし，指令軌道をそのまま追従しようとする，歩行機械の胴体は支持脚対角線回りの倒立振子的な計画外の運動を繰り返すことになってしまう．そのため二脚を切り替えるごとに，遊脚が計画したタイミングと異なる時点で接地し，胴体が不必要に振動してしまうことになる．この不具合をなくし，動的歩行中においてもなめらかな運動を生成できるようにするには，指令された胴体重心の軌道をあらかじめ補償しておくフィードフォワード制御が有効である．この制御法を検討するため，ここで ZMP(zero moment point) の概念を導入することにする． [19]

ZMP とは以下のように定義されている．歩行体が水平面上を移動しており，その脚には床から力 F とモーメント M の組み合わせからなる床反力が作用しているとする．この時 ZMP とは床面上にある作用点であり，その床反作用力のうちのモーメント M の水平成分がゼロとなる点である．

ZMP を考慮して指令された胴体重心の軌道を補償するということは，ZMP が支持脚に囲まれる領域に常時入ることを補償し，胴体運動をできる限りなめらか

に保つような胴体軌道の生成を意味している．これらの補償軌道は 3 方向の加速運動と 3 軸まわりの角速度運動のいずれか，もしくは組み合わせて調整するものである．

そこで本研究では胴体重心の軌道制御法を図 16 のように左右方向の加速度のみを調整し，前後には等速で進むことで ZMP を支持脚対角線上に乗せようとする左右揺動補償軌道を生成する手法を用いる．

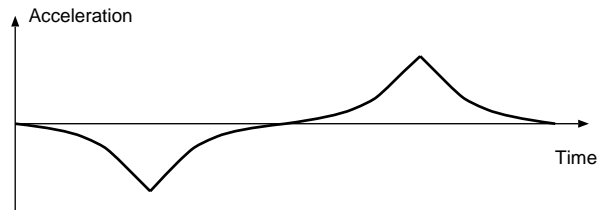


図 16 左右方向 (y 軸) の加速度の推移

左右揺動補償軌道の生成法

ここで導入する歩容と重心軌道生成に使われる「ウェーブ切替点」と「ウェーブ」について説明する．ウェーブ切替点とはトロット歩容などにおいて後脚が復帰運動を完了した時点である．またウェーブとはウェーブ切替点の間である．一般にトロット歩容ではウェーブ切替点は1サイクル中2回形成される．現在のウェーブを W_k , 次のウェーブを W_{k+1} で表される．またその終了時のウェーブ切替点は WE_k , WE_{k+1} というように表される．この概念を用いて左右揺動補償軌道の生成について以下で説明する．

まず立案する胴体重心の左右揺動補償を詳細を論ずるため、いくつかの前提条件を示す．

1. 既知パラメータはウェーブ W_{k+1} での胴体重心の目標軌道、この区間でのデューティ比 β , 二脚支持と四脚支持の実施期間、そしてウェーブ切替点 WE_k と WE_{k+1} での胴体重心の位置、速度、姿勢、そして脚先位置である．
2. 胴体重心の目標軌道は、地表面から高さ H の水平面上をたどる直線軌道であり、胴体の方向は変動しないものとする．
3. 胴体重心の目標軌道に沿った方向の加速度は、ウェーブ W_{k+1} の期間では一定であるとする．
4. 胴体重心が y 軸方向に生ずる補償軌道の速度は、 W_{k+1} の期間で連続的に変化するものとする．
5. 脚の振り運動が胴体に及ぼす動的効果は第一近似として無視する．これは胴体と脚の運動計画を独立して行うことで補償軌道計算の演算アルゴリズムを著しく単純化できる．また、この近似による ZMP の変動誤差も比較的小さいため、最終的には軌道のフィードフォワード補償だけで調整できる．
6. 左右揺動補償軌道を論ずるための基準座標系であるウェーブ座標系 $\Sigma\omega$ は、原点をウェーブ切替点 WE_k における目標軌道上の点とし、目標軌道を x 軸とするように設定する．また、この座標形状には図のようにウェーブ W_{k+1} 初期の四脚支持期間、中期の動歩行期間、そして終了期の四脚支持期間を

それぞれ第 1 , 2 , 3 相と名付けて設定しておく . また支持脚対角線もその座標系上に設定しておく .

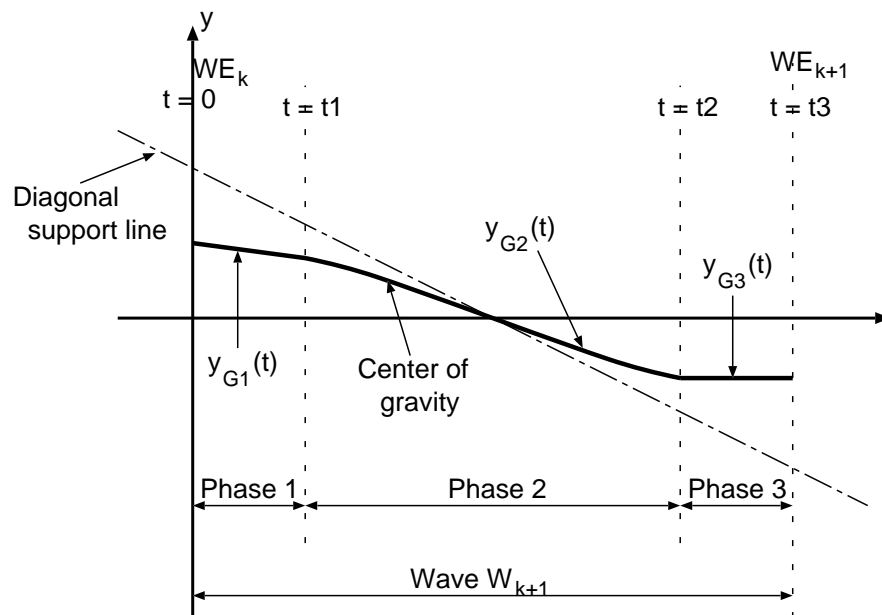


図 17 本体の重心軌道計画

具体的な生成手順

左右揺動軌道の具体的な生成手順は以下の通りである．

まず，胴体重心の進行方向（ x 軸方向）の座標は前提条件 (3) より第 1，2，3 相すべてにおいて以下の式から求まる．

$$x_G = V_{x0}t + \frac{1}{2}\alpha t^2 \quad (2)$$

ここで，初速度 V_{x0} はウェーブ切替点 WE_k での胴体重心速度である．加速度 α は与えられたウェーブ切替点 WE_k ， WE_{k+1} の速度とその間の経過時間から計算される定数である．

各相での胴体重心の y 方向座標は以下のように求める．第 1 相の胴体重心の y 方向座標 $y_{G1}(t)(t = 0 \sim t_1)$ は y 軸方向の速度は等速とすることから

$$y_{G1}(t) = y_{G0} + V_{y0}t \quad (3)$$

と与える．ただし y_{G0} はウェーブ切替点 WE_k での重心の y 座標である．また， V_{y0} は設定する初速度であり，ここでは未知数である．

第 2 相の胴体重心の y 座標 $y_{G2}(t)(t = t_1 \sim t_2)$ は ZMP を支持脚対角線の上に乗せる条件で求める．一般に前提条件 (2) から胴体重心が $P_G(x_G, y_G)^T$ ， \dot{P}_G の運動を行うとすると，地表面上の ZMP の座標 $P_Z(x_z, y_z)^T$ は

$$P_Z = P_G - A \cdot \ddot{P}_G \quad (4)$$

$$A = (H/g) \quad (5)$$

となる．ただし， H ：重心の高さ， g ：重力加速度 とする．

よって第 2 相で胴体重心の y 方向加速度が $\ddot{y}_G(t)$ であるとき，ZMP の軌道は次式となる．

$$P_Z(t) = \begin{bmatrix} x_z(t) \\ y_z(t) \end{bmatrix} = \begin{bmatrix} V_{x0}t - A\alpha + \frac{1}{2}\alpha t^2 \\ y_{G2}(t) - A\ddot{y}_{G2}(t) \end{bmatrix} \quad (6)$$

一方，支持脚の座標 $A(x_a, y_a)$, $B(x_b, y_b)$ を結ぶ支持脚対角線は

$$y_z(t) = y_a + \frac{y_a - y_b}{x_a - x_b}(x_z(t) - x_a) \quad (7)$$

である．

以上のことから，ZMP を支持脚対角線上に保つような胴体重心の y 座標 $y_{G2}(t)$ は，(6) 式を (7) 式に代入して誘導される微分方程式の解として次式のように求まる．

$$y_{G2}(t) = C_1 \exp\left(\frac{t}{\sqrt{A}}\right) + C_2 \exp\left(\frac{-t}{\sqrt{A}}\right) + B_2 t^2 + B_1 t + 2AB_2 + y_{z0} \quad (8)$$

ただし，

$$B_1 = \frac{y_a - y_b}{x_a - x_b} V_{x0}, \quad B_2 = \frac{y_a - y_b}{x_a - x_b} \cdot \frac{\alpha}{2}$$

$$y_{z0} = y_a - \frac{y_a - y_b}{x_a - x_b}(A\alpha + x_a)$$

となる．ここで C_1 , C_2 は未知の定数である．

さらに第 3 相では，左右方向速度 $\dot{y}_{G3}(t = t_2 \sim t_3)$ をゼロに設定するので

$$y_{G3} = Y_{G3} \quad (9)$$

となる．ここで Y_{G3} は未知の定数である．

以上，(3) , (8) , (9) 式に含まれる未知の定数は左右方向の運動の連続条件を満たすため，時刻 t_1 , t_2 について以下の 4 つの関係から誘導できる．

$$y_{G1}(t_1) = y_{G2}(t_1) \quad (10)$$

$$\dot{y}_{G1}(t_1) = \dot{y}_{G2}(t_1) \quad (11)$$

$$y_{G2}(t_2) = y_{G3}(t_2) \quad (12)$$

$$\dot{y}_{G2}(t_2) = 0 \quad (13)$$

誘導した結果を以下に示す .

$$C_1 = \frac{\sqrt{A}(2B_2t_2) \left(1 + \frac{t_1}{\sqrt{A}}\right) \exp\left(\frac{t_2 - t_1}{\sqrt{A}}\right) - B_2t_1 + 2AB_2 + y_{z0} - y_{G0}}{\left(1 + \frac{t_1}{\sqrt{A}}\right) \exp\left(\frac{2t_2 - t_1}{\sqrt{A}}\right) + \left(1 - \frac{t_1}{\sqrt{A}}\right) \exp\left(\frac{t_1}{\sqrt{A}}\right)}$$

$$C_2 = C_1 \exp\left(\frac{2t_2}{\sqrt{A}}\right) + \sqrt{A}(2B_2t_2 + B_1) \exp\left(\frac{t_2}{\sqrt{A}}\right)$$

$$V_{y0} = \frac{C_1}{\sqrt{A}} \exp\left(\frac{t_1}{\sqrt{A}}\right) - \frac{C_2}{\sqrt{A}} \exp\left(\frac{-t_1}{\sqrt{A}}\right) + 2B_2t_1 + B_1$$

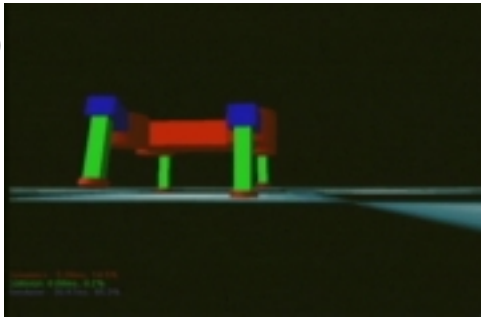
$$Y_{G3} = y_{G2}(t_2)$$

3.2.3 動歩行シミュレーションの結果

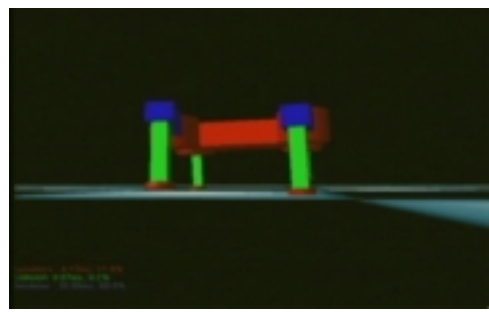
以上のアルゴリズムを用い動歩行シミュレーションを行った．歩行空間は障害物のない平地と仮定し，速度ゼロの状態から徐々に進行方向への速度を上げていき，速度 80mm/sec まで一定加速を行うようコントローラの設計を行った．重心軌道は左右揺動補償を用いて動歩行における安定性を向上させた．

トロット歩容をとりつつ歩行する様子を図 18 に示す．その時の胴体の傾き角である Pitch 角は図 19 のようになった．胴体のピッチ角が ± 4 度の振動があるが，規則的な安定した姿勢に収まっている．また左右揺動補償による重心軌道の推移は図 20 のような結果が得られた．床との接触で滑りがあり，左方へのずれがみられるが，規則的な重心軌道をとっているのがわかる．図 21 はトロット歩容において対角上にある脚の角度推移を出力した．この角度は胴体から 1 番目の関節で進行方向への脚をせり出す動きを行う．二脚ともトロット歩容では同じ角度で動く様子がこのように示すことができた．

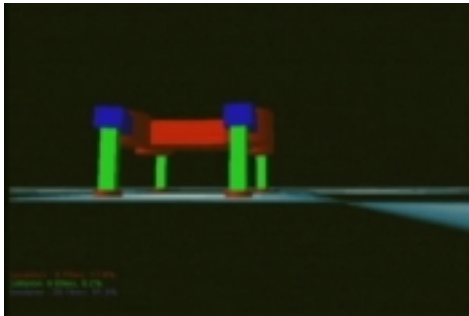
(1)



(5)



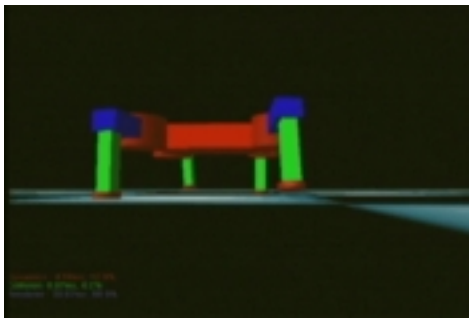
(2)



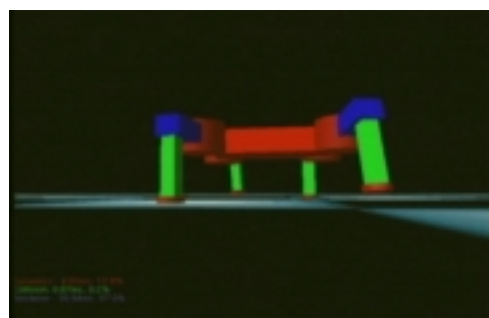
(6)



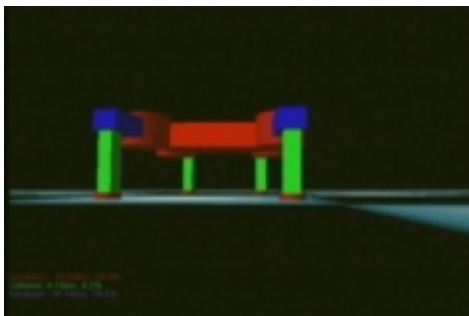
(3)



(7)



(4)



(8)

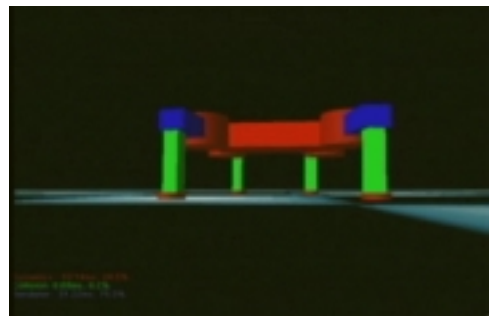


図 18 動歩行シミュレーション

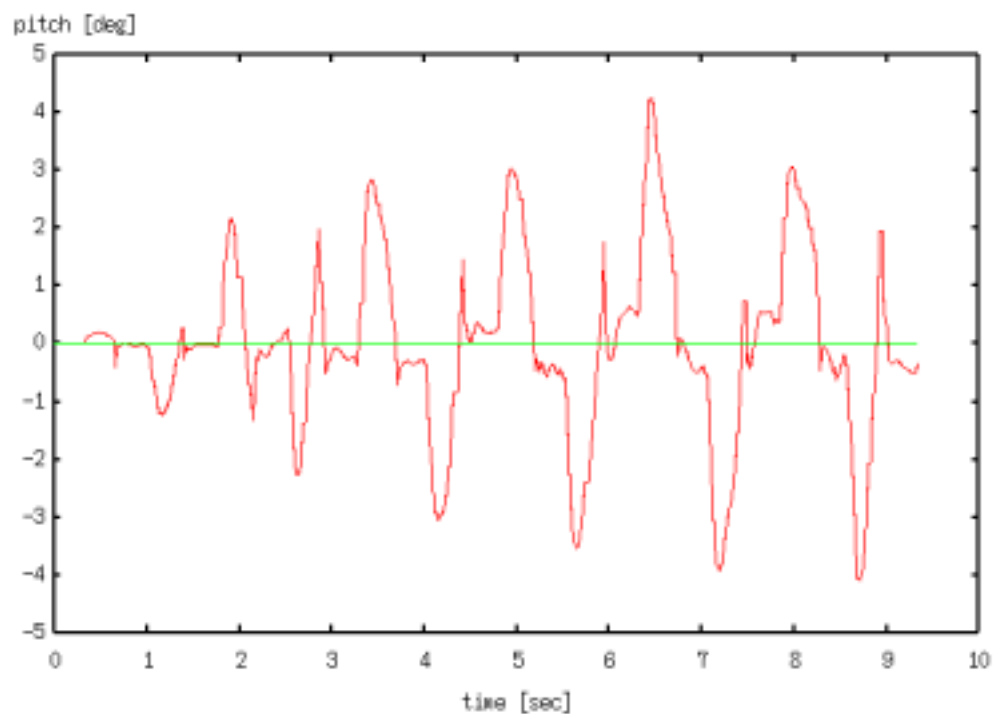


図 19 トロット歩容における胴体の Pitch 角

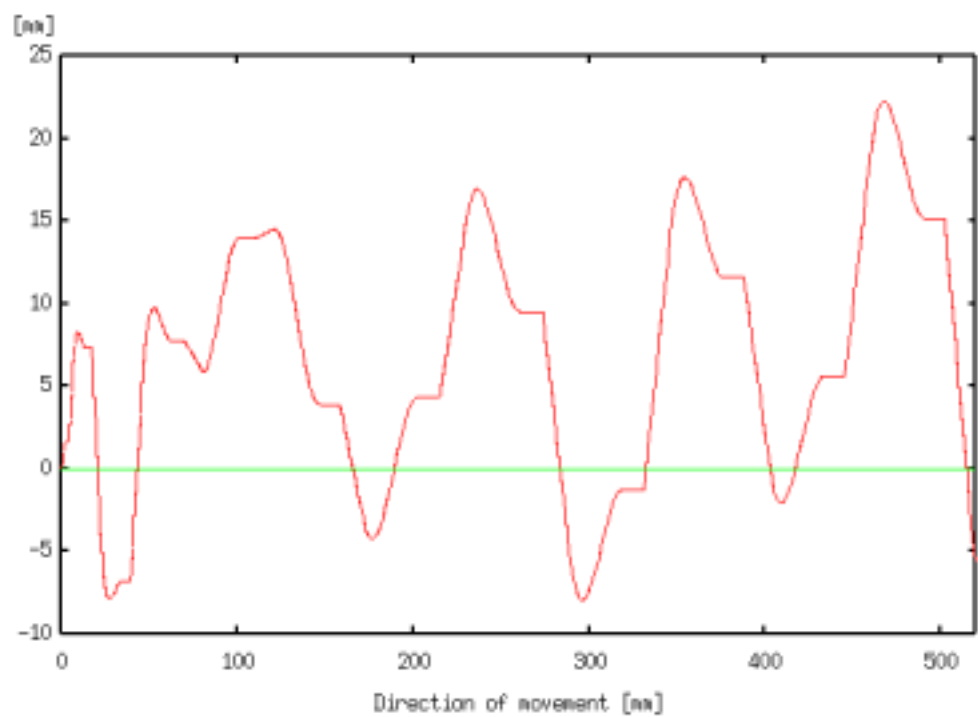


図 20 重心の左右揺動補償軌道

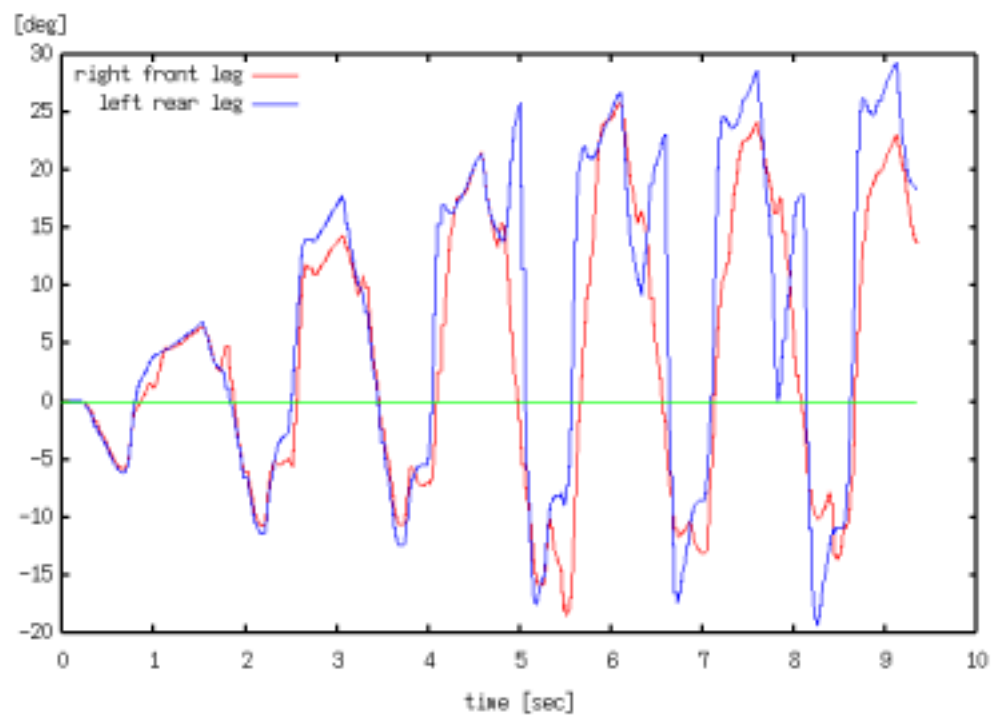


図 21 右前脚と左後脚の第一関節の角度推移

3.3 歩行シミュレーションの考察

ここでは各シミュレーションの結果について考察を行う．

まず静歩行においては図 10 のように安定した歩行を行うことができた．パラメータの設定においてもシミュレーション上において最適な値を絞り込むことができた．数値的にも静的安定評価基準である安定余裕を一定値以上に保ったままであるのを図 11 のように確認できた．

動歩行においてはフィードフォワード的な安定補償である左右揺動重心軌道を歩行に適用させるて歩行させることを実現できた．しかし歩行速度を大きくすると床との衝突やすべりによる影響が大きくなりすぎてしまい，不安定になることが多かった．これはコントローラ部分において床からの外力を考慮した別の補償や床衝突とのシミュレーション部分の調整，関節モータ制御の改良が必要だと思われる．

このようにシミュレーションにおいては一定の歩行が可能であることは示されたが，より安定した歩行のシミュレーションを行うには以下のような改良の必要性がある．

- 力センサシミュレーションの制作
- 関節モータの速度制御
- 動歩行コントローラの改良
- 床との接触シミュレーションの調整

より正確な歩行シミュレーションが行えるにはこれらの部分について，さらなる検討が必要と考察する．

4. 実験

3章で説明した静歩行のシミュレーションで設計した歩行コントローラの実機への適用を試みるため、実際に4脚歩行ロボットで実験を行った。

4.1 実験システム

本研究では以前の研究 [22] で制作された四脚歩行ロボット TITAN-VIII を用いて実験を行った。このロボットは3自由度の脚を四本持ち、合計12自由度の動きが可能である。脚と胴体部分の開発は東京工業大学の広瀬らに [20] [21] によって行われており、それをベースにバッテリー、モータドライバ、コンピュータ、DC/DCコンバータ、ネットワークインターフェイス、各種センサなどを搭載し、自立歩行を可能にしたものである。全体の構成は図 22 のようになっている。

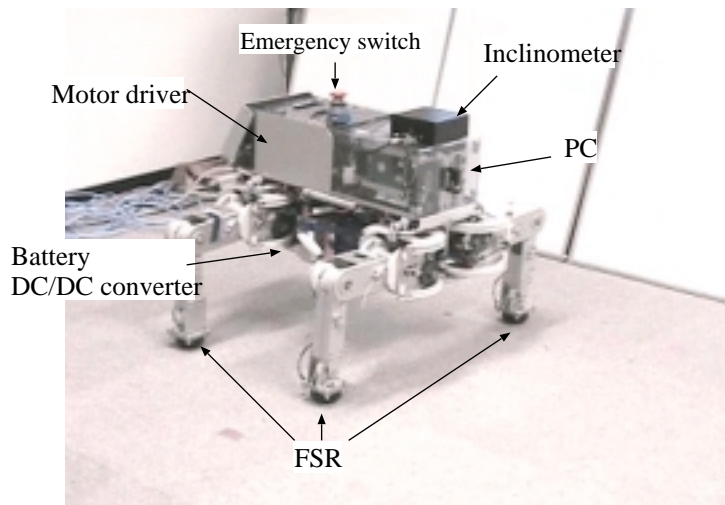


図 22 TITAN-VIII

ハードウェアシステムは大きく分けると制御用コンピュータ部、TITAN-VIII による機構部から構成されている。コンピュータ部は CPU:Pentium III-450[MHz] , Main Memory:128MBytes DRAM , PCMCIA Card Slot , Radio Ethernet Card , I/O Board:RIF-01 , OS:Linux 2.2.14 , および RT-Linux 2.2 を使用している。

また各種センサとして、姿勢センサにはデータテック社の GU-3013 を胴体中央に設置した。これは振動ジャイロおよび加速度を各々 3 軸検出する事ができ、姿勢角、角速度、加速度を出力する。また接触センサとして FSR (Force Sensing Resistors) を各足裏に 1 機ずつ合計 4 機搭載しており、脚の接地状況を確認することができる。ロボットの関節にはそれぞれポテンショメータを装備しており、各関節角を出力している。

また本体部分の仕様は以下のようにになっている。

表 1 TITAN-VIII の基本仕様

| | |
|---------|--------------------------|
| 寸法 | 400 × 600 × 250 |
| 脚数 | 4 |
| 自由度 | 12 (3 自由度 × 4 脚) |
| 本体重量 | 約 20kg |
| 可搬重量 | 約 7kg |
| アクチュエータ | DC サーボモータ (50[W] × 12) |
| 駆動回路 | TITECH ROBOT DRIVER × 12 |
| 位置センサ | ポテンショメータ |
| 電源 | DC6 ~ 48[V] (モータ駆動用) |
| 各脚の最大速度 | 0.9m/sec (遊脚時) |

4.2 静歩行実験

4.1 節で述べた歩行アルゴリズムを用いてシミュレーション上で転倒安定を確認した後，実機に制御コントローラを搭載して静歩行実験を行った．図 23 は各脚の接地状況を示している．シミュレーション時に設計したとおり脚の接地を左後，左前，右後，右前の順に行っているのがわかる．その様子を図 24 にのせる．

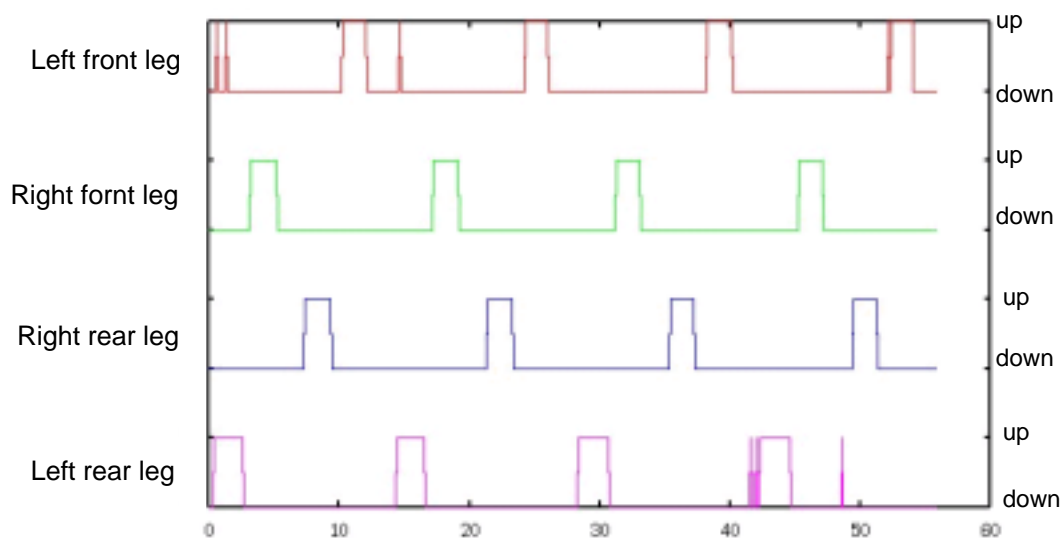


図 23 各脚の接地状況

(1)



(5)



(2)



(6)



(3)



(7)



(4)



(8)



図 24 静歩行実験

4.3 実験の考察

静歩行においては図 24 のように安定した歩行が実現できた．重心軌道は図 25 に表示したようにシミュレーションにおける軌道とほぼ変わらない結果を得ることができた．この図において実機の軌道が安定しているのは，シミュレーションのように毎時軌道位置を出力できないためである．また図 26 に表示した縦安定余裕値の結果においてもほぼ安定した結果が得られたが，シミュレーション上ではそれぞれの正確なリンク位置から重心位置を求めたため，脚の昇降によって安定余裕値が若干低くなってしまっている．実機においては胴体部分の中心を擬似的に重心とおかれるため，その誤差が現れている．

シミュレーションから反映させたのは次の点が揚げられる．間歇クロール歩容においては 3.1.2 節で述べたように重心を中心とした安定余裕値円が接地脚支持多角形内に存在するような重心軌道を接地時に決定する．従って安定余裕の設定した値によって重心軌道は変動するため，シミュレーション時にその値を最適な数値にシミュレーション上で設定し，より安定した歩行を実現できた．

また実機とシミュレーションとのコントローラの切り替えは完全にそのまま移植できるまでには至らなかったもの，若干の修正によって実機に実装できるまでに設計した．

これらから本シミュレーションにおける静歩行実験での有効性を示すことができた．

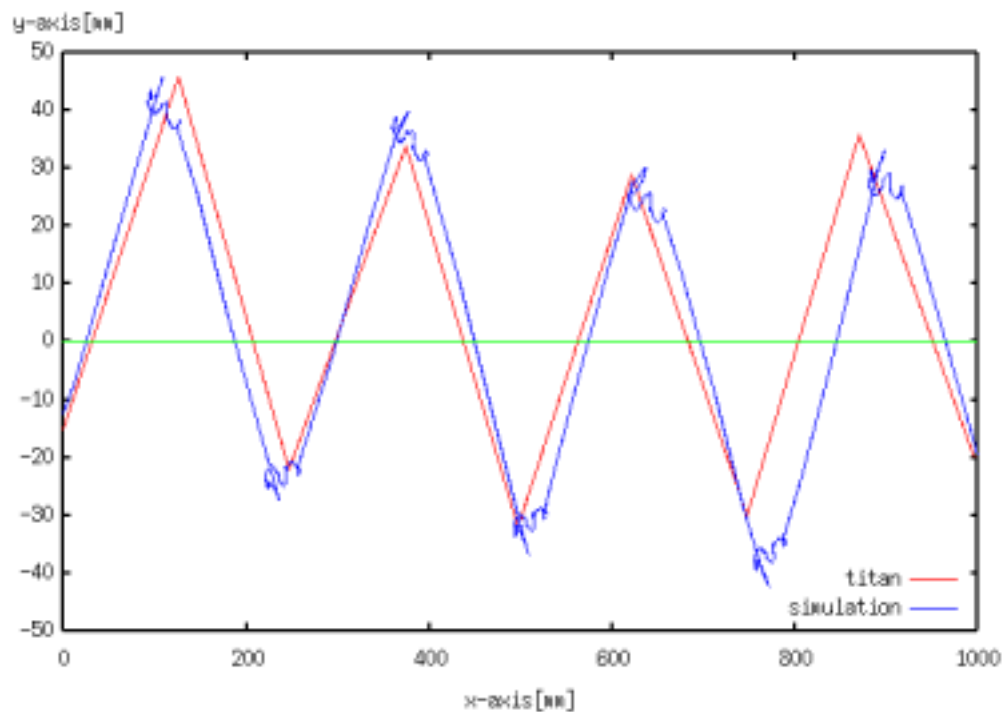


図 25 重心の軌道

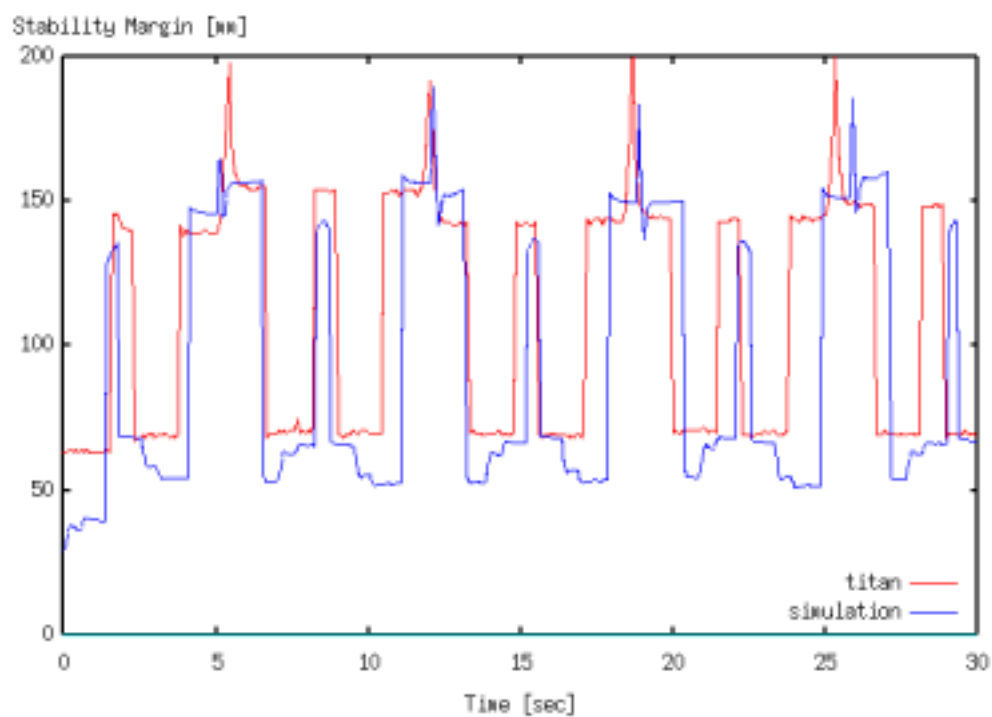


図 26 縦安定余裕

5. 結論

5.1 結論

本論文では、動力学演算パッケージである MathEngine Toolkits を歩行ロボットシミュレーションに適用する手法を提案した。また、四脚歩行ロボット TITAN-VIII における静歩行、動歩行アルゴリズムの検証に用いることにより、本手法の有効性を確認した。実機への適用においては静的安定性を考慮した間歇クロール歩容における実装は実現できたが、動歩行であるトロット歩容の実装までには至らなかった。これは動歩行においては床との衝突や、脚関節の移動速度などに実機との誤差があるため、搭載できるレベルに達しなかったためである。この誤差を低減するように改良することで四脚歩行ロボットの動歩行にも実装が可能と思われる。

これらのように本手法で示した動力学演算の負担を軽減することや、MathEngine の高速性を利用することにより、多く回数のシミュレーションを必要とするロボットの行動学習や高速演算を生かした制御設計などに対して有効であると思われる。

5.2 今後の課題

今後の課題としては、第一に歩行シミュレータの改良が挙げられる。具体的には以下の部分の開発が必要である。

- シミュレーションからそのまま実機へ移行が可能なコントローラ設計
- カセンサシミュレーション
- 関節モータの速度制御
- 動歩行コントローラの改良設計
- 床との衝突シミュレーションの調整

これらを考慮した歩行シミュレータの改良が今後の研究課題である。

謝辞

本研究を進めるにあたって，終始一貫して御指導，御助言を賜りました奈良先端科学技術大学院大学 情報科学研究科，小笠原 司 教授に心から深く感謝の意を表します．

修士論文執筆にあたり，適切な御助言を賜りました奈良先端科学技術大学院大学 情報科学研究科，杉本 謙二 教授に心から深く感謝の意を表します．

本研究遂行にあたり適切な御検討と御助言を賜りました奈良先端科学技術大学院大学 情報科学研究科，松本 吉央 助教授に心から深く感謝の意を表します．

本研究の方針の決定，研究の遂行など，御検討と御助言を賜りました奈良先端科学技術大学院大学 情報科学研究科，中村 恭之 助手に心から深く感謝の意を表します．

研究の遂行や日常生活等，様々な面で御助言を頂きました本研究科 多田 充徳氏，上田 悦子氏，竹村 裕氏，足立 佳久氏に深く感謝致します．

研究や日常生活で相談にのって頂きました本研究科 怡土 順一氏，栗田 雄一氏，小枝 正直氏，後藤 健志氏，佐々木 晋介氏，谷 裕章氏，出口 将人氏，乗松 泰明氏に深く感謝致します．

日常生活において，色々ご支援頂きました本研究科 金岡 恵 秘書に深く感謝致します．

最後に，本研究をまとめるにあたり，様々な形で御協力，御支援を頂きました研究室内外のすべての皆様に深く心より感謝致します．

参考文献

- [1] 西脇光一，長坂憲一郎，稲葉雅幸，井上博允．ヒューマノイド H5 におけるダイナミック動作構築のための動力学シミュレーション環境．第 16 回日本ロボット学会学術講演会予稿集，pp.839–840, 1998.
- [2] mathengine. MathEngine Toolkits.

<http://www.mathengine.com/>
- [3] havok.

<http://www.havok.com/>
- [4] 木村浩, 下山勲, 三浦宏文. 四足歩行ロボットの力学的解析. 日本ロボット学会誌, Vol. 6, No. 5, pp.367–378, 1988.
- [5] 中村仁彦，山根克．拘束条件が不連続に変化するリンク系の動力学 日本ロボット学会誌，Vol. 18, No. 3, pp.435–443, 2000
- [6] 藤本康孝，川村篤男 2 足歩行ロボットの精密なシミュレーションモデルの確率 日本ロボット学会学術講演会，pp.1183–1184, 1994.
- [7] 藤本康孝，川村篤男．床との衝突および摩擦を考慮した 2 足歩行ロボットの三次元運動シミュレーション 日本ロボット学会，Vol. 15, No. 6, pp.857–863, 1997.
- [8] 金広文男，宮田なつき，梶田秀司，藤原清司，比留川博久．ヒューマノイドの実時間制御ソフトウェア開発環境：OpenHRP. 第 19 回日本ロボット学会学術講演会予稿集 pp.167–168, 2001.
- [9] 金広文男．高速シミュレーション環境 FAST を用いたヒューマノイドシミュレーション 第 19 回日本ロボット学会学術講演会予稿集 pp.931–932, 2001.

- [10] 広瀬茂男, 有川敬輪. 研究用プラットフォームとしての普及型歩行ロボット TITAN-VIII の開発. 日本ロボット学会誌, Vol. 17, No. 8, pp.1191–1197, 1999.
- [11] 山崎文敬, 松井龍哉, 宮下宏敬, 北野宏明 ヒューマノイド PINO -ヒューマノイドプラットフォームとしての外装と構造- 第 18 回日本ロボット学会学術講演会予稿集 pp.921–922, 2000.
- [12] 米田完. 脚移動. 日本ロボット学会誌, Vol. 16, No. 7, pp.897–901, 1998.
- [13] 木村浩. 4 足歩行ロボットの動歩行について. 日本ロボット学会誌, Vol. 11, No. 3, pp.372–378, 1993.
- [14] 広瀬茂男, 塚越秀行, 米田完. 不整地における歩行機械の静的安定性評価基準. 日本ロボット学会誌, Vol. 16, No. 8, pp.1076–1082, 1998.
- [15] 塚越秀行, 広瀬茂男. 間歇クロール歩容の提案とその生成原理. 日本ロボット学会誌, Vol. 17, No. 2, pp.301–309, 1999.
- [16] 木村浩, 下山勲, 三浦宏文. 四足歩行ロボットの力学と制御 (複数脚接地によって生じる制御入力冗長性を利用した動歩行). 日本機械学会論文集 (C 編), 55 巻, 517 号, 1989-9.
- [17] 米田完, 飯山浩幸, 広瀬茂男. 4 足歩行機械の間欠トロット歩容. 日本ロボット学会誌, Vol. 14, No. 6, pp.881–886, 1996.
- [18] 広瀬茂男, 米田完. 4 足歩行機械の静動融合歩容とその連続軌道生成. 日本ロボット学会誌, Vol. 9, No. 3, pp.267–275, 1991.
- [19] 米田完, 広瀬茂男. 歩行機械の転倒安定性. 日本ロボット学会誌, Vol. 14, No. 4, pp.517–522, 1996.
- [20] 広瀬茂男, 米田完. 実用的 4 足歩行機械の開発に向けて. 日本ロボット学会誌, Vol. 11, No. 3, pp.360–365, 1993.

- [21] 広瀬茂男，福田靖，菊池秀和. 4 足歩行機械の制御システム. 日本ロボット学会誌，Vol. 3, No. 4, pp.304–323, 1985.
- [22] 竹村裕．リズム生成に基づく四足歩行ロボットの歩容生成に関する研究．奈良先端科学技術大学院大学 2001 年度修士論文．