

力覚インタラクションのためのロボットシミュレーション*

長谷川晶一^{*1}, 田崎勇一^{*2}, 石川俊明^{*1}, 長田陽介^{*1}, 佐藤誠^{*1}

Robot simulation for haptic interaction

Shoichi Hasegawa, Yuuichi Tazaki, Ishikawa Toshiaki, Yosuke Osada and Makoto Sato

^{*1} P & I Laboratory, Tokyo Institute of Technology

^{*2} Graduate School of Information Science and Engineering, Tokyo Institute of Technology

We create an environment for haptic interaction with controlled articulated bodies like robots. We develop a novel simulator which can simulate articulated bodies and multi bodies in haptic rates. We combine contact force solver based on penetration volume and Featherstone's articulated body solver. The simulator realizes both haptic rate updates and accurate contact force simulation.

Key Words : haptic, simulation, contact force

1. はじめに

機構やロボットの挙動を設計図や数式などから推測することは難しい。最近のCADシステムは、機構の形状や動作、干渉チェックなどをシミュレーションによって行うことができるが、実際に動いているロボットに直接触れて関節の硬さを感じたり、機構に力を加えてみて動作を確かめる場合に比べると、得られる情報は非常に限定されたものになってしまう。

力覚インタフェースとリアルタイムシミュレーションを組み合わせることで、シミュレーションされた物体モデルにユーザが力覚インタフェースを介して触れることができる。このようなインタラクションは現実の機構やロボットに直接触れる場合に近い情報をユーザに与えることができると考えられる。

我々は、剛体の接触体積に基づくペナルティ法を用いた剛体運動のシミュレーション手法⁽¹⁾を提案し、蝶番や関節を持つ物体の運動のシミュレーションを統合した⁽²⁾。これらのシミュレーションによって、計算機内にロボットの動力学モデルを構築することができる。この動力学モデルを実際のロボットと同様に制御すると、動力学モデルは、実際のロボットと同様に動作する。本稿では、シミュレータと力覚インタフェース SPIDAR⁽³⁾を用いてロボットモデルとの力覚インタラクション環境を構築し、6足歩行機構のシミュレーションとインタラクションを実現する。

2. シミュレーションへの要求

2-1 シミュレーションの精度 リアルタイム物理シミュレーションはゲームなどに応用されているが、見た目と計算量が第一となるため、正確性はあまり考慮されていない。ロボットシミュレーションでは、実際のロボットの挙動を知ることが目的となるので、現実のロボットの動作にできるだけ近い動作を作り出すことが目標となる。

物体の運動は、物体に働く力が求まれば、運動方程式の数値解として求まる。重力やバネ・ダンパによる力は、モデル化が簡単な上、簡単な計算で求まる。また、関節に働く力も拘束条件として精度良くモデル化することができる。

しかし、物体同士が接触したときに働く接触力のモデル化は簡単ではない。物体が十分に硬ければ、接触力は撃力となり、物体は瞬時に速度を変える。現実には、物体は変形し、弾性・粘性に応じた力を発生すると考えられる。

ゲームなどには、接触力を拘束力としてモデル化する方法⁽⁴⁾⁽⁵⁾がよく用いられるが、この方法は物体を十分に硬いものとしてモデル化してしまう。ペナルティ法⁽⁶⁾⁽⁷⁾は接触をバネダンパでモデル化するため、衝突の際の変形をある程度模擬することができる。

また、接触力の向きには摩擦力の制約が働く。クーロンの摩擦モデルは、日常環境での摩擦力を良く近似する。このモデルは1点で接触した場合のモデルなので、領域での接触に適用するためには接触力の分布を考慮したモデル化が必要になる。

^{*1} 東京工業大学 精密工学研究所

^{*2} 東京工業大学 情報理工学研究所

Email: {hase|ishikawa|yosa|msato}
@hi.pi.titech.ac.jp,
tazaki@cyb.mei.titech.ac.jp

2.2 シミュレーションの速度 安定な力覚インタラクションを行うためには力覚インタフェースを300Hz から 1kHz 以上で制御しなければならないことが知られている．このため、グラフィックスに必要な60Hz に比べて5倍以上の速度で1ステップのシミュレーションを計算する必要がある．

ゲームなどで用いられる拘束力を解く方法⁽⁴⁾ ⁽⁵⁾では、接触の数 n に対して $O(n^3)$ の計算時間を要する．ペナルティ法⁽⁶⁾ ⁽⁷⁾は、接触の数 n に対して計算時間が $O(n)$ と高速で、リアルタイムシミュレーションに向いている．

2.3 力覚レンダリング 力覚インタラクションを行うためには、力覚ポインタ (=シミュレータ内のユーザの手のモデル) に働く力を計算し、力覚インタフェースを介してユーザに提示する必要がある．この計算は、力覚レンダリングと言われる．力覚ポインタは、シミュレータ内のほかの物体と異なり、動力学モデルがシミュレータの外部に存在する．ペナルティ法による接触力計算は、拘束力を侵入量だけから計算するため、力覚ポインタを直接扱うことができる．

3. シミュレータの構築

前節で述べた要求を満たすため、接触力の計算には、接触体積に基づくペナルティ法⁽¹⁾を用い、関節の拘束力の計算には計算が高速な Featherstone の方法⁽⁸⁾を用いてシステムを構築した．

3.1 シミュレーションの流れ 構築したシミュレータは、次の計算を行う．

1. 重力など場の力の計算
2. 接触判定
3. 接触体積と抗力、摩擦力の計算
4. 関節に働く拘束力の計算
5. 加速度、速度を積分し、速度、位置を求める．

以下で各処理を説明する．

3.2 接触力の計算 物体に働く接触力は、接触の法線に平行な抗力と、垂直な摩擦力に分けて考える．

抗力 接触面には、接触面上の任意の点が互いに侵入しないという拘束を満たすような力、すなわち抗力が働く．そこで、接触領域全体にバネダンパモデルが分布すると考え、バネダンパモデルが発生する力の総和が物体に働いたと考える．これにより接触面に働く力とトルクを計算する．

摩擦力 接触面上の微小面積に働く動摩擦力や最大静止摩擦力は、その微小面積に働く抗力に比例する．接触面全体に働く摩擦力はこれの総和となる．

そこで接触領域全体にクーロンの摩擦モデルが分布

し、摩擦力を発生すると考える (Fig.1)．これにより、摩擦力だけでなく、摩擦トルクも計算できる．

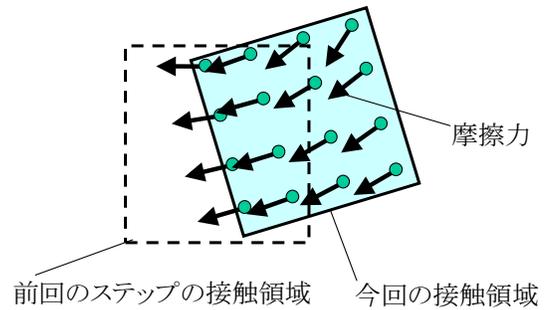


Fig. 1 摩擦力のモデル

接触体積の計算 構築したシミュレータでは、物体形状を凸多面体の集合で表し、凸多面体単位で接触判定を行い、接触体積を求める．

接触判定には、データ構造が単純で計算が高速な GJK アルゴリズム⁽⁹⁾を用いた GJK アルゴリズム⁽⁹⁾は、凸形状対が与えられたときに、最近傍点对もしくは接触体積上の1点を求める．

Muller と Preparata⁽¹⁰⁾は、凸多面体の共通部分の内部の点が1点与えられたときに、共通部分全体(を構成する3角形)を求めるアルゴリズムを提案している．構築したシミュレータでは Muller のアルゴリズムを用いて共通部分を構成する3角形を求め、3角形ごとに抗力や摩擦力を計算する．

3.3 関節の拘束力の計算 関節に働く拘束力の計算には Featherstone の方法⁽⁸⁾を用いた．ここでは簡単にこの方法を説明する．

木構造のリンクでは、あるリンクは1つのより根に近い側のリンクといくつかのより葉に近い側のリンクと関節で接続している．

あるリンク i が根側のリンクから受ける力とトルク (=拘束力) を \hat{f}_i^l 、リンクの加速度と角加速度を \hat{a}_i とすると、

$$\hat{f}_i^l = \hat{f}_i^A \hat{a}_i + \hat{Z}_i^A \quad (1)$$

のように、リンク i とより葉側のリンク全体が一つの剛体であるかのような式が書け、 \hat{f}_i^A, \hat{Z}_i^A は関節加速度によらない (Fig.2 参照)．そこで、次のように、拘束力を求める．

1. 木構造の根から葉に向かって、各リンクの速度と角速度を求める．
2. 木構造の葉から根に向かって、式 (1) の \hat{f}_i^A, \hat{Z}_i^A を求める．また、このとき拘束力 \hat{f}_i^l も求める．
3. 木構造の根から葉に向かって、各関節の加速度と、リンクの加速度・角加速度を求める．

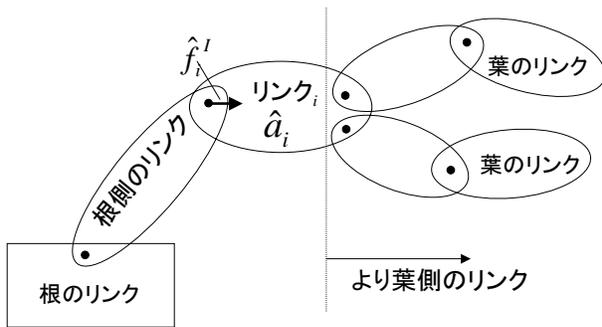


Fig. 2 木構造のリンク

このように Featherstone の方法では、拘束力と同時に各リンクの加速度・角加速度を求めることができる。また、2. のときに、各リンクにペナルティ法で求めた接触力を加えることで、接触力を考慮に入れた各リンクの加速度を求めることができる。

4. 評価

構築したシミュレータと力覚インタラクションのための環境を評価するため、いくつかの実験を行った。

4.1 抗力の計算時間の評価 構築したシミュレータが、拘束を解く方式に比べて、高速で力覚提示に向いていることを確認するために、計算時間を比較した。拘束を解く方式のシミュレータとしては、Open Dynamics Engine⁽¹¹⁾を用いた。

両方のシミュレータで Fig.3 のように積み上げられたブロックをシミュレートし、ブロックの数と1ステップの計算時間の平均の関係を調べた。1ステップのシミュレータ内の時間は5ms、重力加速度を $9.8m/s^2$ 、ブロックの大きさを $1m \times 1m \times 2m$ 、質量を1kgとした。シミュレーションは、Intel Pentium 4 1.6GHzCPU を1個搭載したPCで行った。

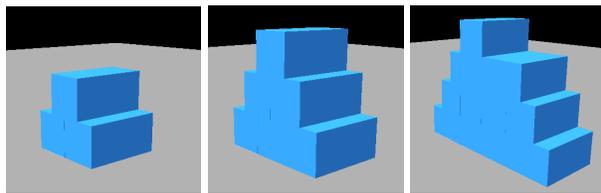


Fig. 3 評価のために用いたシーン：3個、6個、13個のブロック

Fig.4 に、ブロックの数と1ステップの計算時間の関係を示す。拘束を解析する Open Dynamics Engine では、ブロックの数が増えるにつれて、急激に計算時間が増加するが、構築したシミュレータでは、ほぼ線形に計算時間が増加する。

以上から、特にブロック数がある程度多くなると

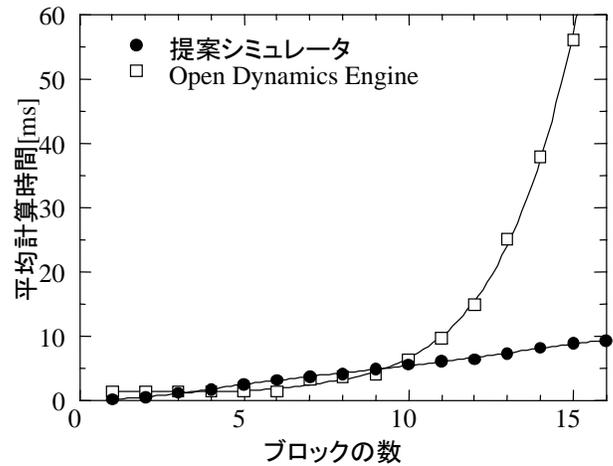


Fig. 4 計算時間の比較

ときには、ペナルティ法の方が1ステップのシミュレーションに必要な計算時間が短く、接触点数の増加に対する計算時間の増加が緩やかなので、力覚提示に向いていることが分かる。

4.2 摩擦力計算の評価 摩擦力のシミュレーションの正確さを調べるため、実世界、構築したシミュレータ、Open Dynamics Engine で同じ状況を用意し、摩擦力を計測した。摩擦力の計測は、重りをバネを介して一定の速度で引いた際に発生する力を計測することで行った。Fig.5 に実世界での実験の様子を示す。このよ

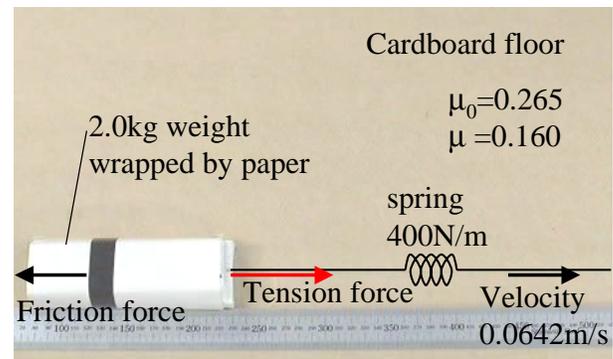


Fig. 5 摩擦力評価のための実験の様子

うな条件では、重りは静止摩擦状態と動摩擦状態を行き来するスティックスリップ現象を起こすことが知られている Fig.6 に実験とシミュレーションの結果を示す。摩擦力を見ると、実世界と Springhead ではスティックスリップ現象が発生しているが、Open Dynamics Engine ではシミュレーションできないことがわかる。

4.3 6足歩行機構のシミュレーション 構築したシステムによって、制御された歩行機構との力覚インタラクションを行った。Fig.7 に6足歩行機構(カブトムシ)とインタラクションの様子を示す。

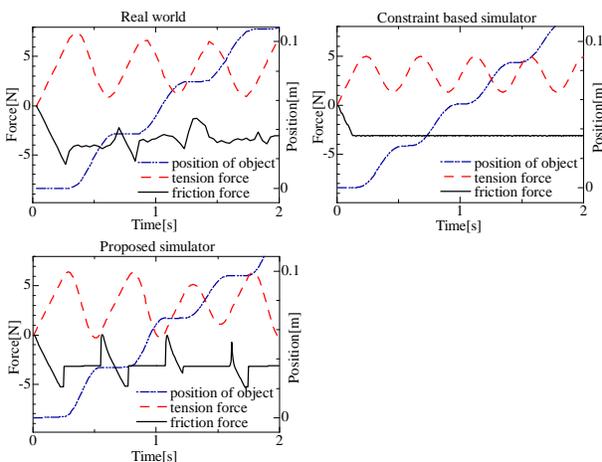


Fig. 6 摩擦評価実験とシミュレーションの結果



Fig. 7 歩行機構とのインタラクション

このモデルは、胴体に3自由度、各足の付け根に2自由度、各足の間と足先に1自由度、計27自由度の関節を持つ。

モデルを歩行させるために、足の先端が楕円を描くように各関節をPD制御して足を動かした。昆虫の歩容をまねて3本の足を組みにし、各組の位相を180度ずらして動かした。水平方向の移動の向きを変えることで前後移動と左右回転を実現した。

力覚インタフェースを介して歩行機構に触れると、ユーザは足の動きやそれによって生じる振動を指先に感じる事ができた。さらに機構全体を持ち上げたり、足を押さえた際の動作の様子や足に加わる力を観察する事ができた。

このシミュレーションには、Pentium 3 1GHzのCPUを持つPCを用いたが、500Hzの更新周期を維持する事ができた。

5. おわりに

我々は、剛体運動シミュレーションと力覚インタフェースを組み合わせ、ロボットのような制御された物体と力覚インタフェースを介してインタラクションできる環境を構築した。

今後は、このような環境をロボットの機構や制御の開発に役立てたいと考えている。

- (1) 藤井伸旭, 長谷川晶一, 橋本直己, 小池康晴, 佐藤誠: “ペナルティ法を用いた剛体運動シミュレータの開発”, 日本バーチャルリアリティ学会第7回大会論文集
- (2) 長谷川晶一, 田崎勇一, 佐藤誠: “ハプティックインタラクションのための物理ベースモデリング”, 日本バーチャルリアリティ学会第8回大会論文集
- (3) Masahiro Ishii and Makoto Sato: A 3D Spatial Interface Device Using Tensed Strings, Presence Vol. 3, No. 1, Winter 1994, pp.81-86
- (4) D. Baraff: “Analytical methods for dynamic simulation of non-penetrating rigid bodies”, Computer Graphics Proceedings(SIGGRAPH 89) Vol. 23, pp. 223-232, 1989.
- (5) D. Baraff: “Fast contact force computation for nonpenetrating rigid bodies”, Computer Graphics Proceedings(SIGGRAPH94), pp. 23-34, 1994.
- (6) M. Moore, J. Wilhelms: “Collision Detection and Response for Computer Animation”, Computer Graphics (SIGGRAPH 88), Vol. 22, pp. 289-298, August 1988.
- (7) H. Keller, H. Stolz, A. Ziegler: “Virtual Mechanics: Simulation and Animation of Rigid Body Systems”, <http://citeseer.nj.nec.com/keller94virtual.html>, 38 pages, 1993.
- (8) R. Featherstone: The Calculation of Robot Dynamics using Articulated-Body Inertias, Int. J. Robotics Research, vol. 2, no. 1, pp. 13-30, 1983.
- (9) E. G. Gilbert, D. W. Johnson and S. S. Keerthi, “A fast procedure for computing the distance between complex objects in three-dimensional space,” IEEE Journal of Robotics and Automation 4(2), 1988, pp. 193-203.
- (10) D. E. Muller, F.P.Preparata, “Finding the intersection of two convex polyhedra,” Theoretical Computer Science, 7(2), 1978, pp. 217-236.
- (11) Russell Smith: Open Dynamics Engine, <http://opende.sourceforge.net/>