

修士論文

摩擦を含む変化する接触状態と
多関節構造に関する
リアルタイムインタラクティブ
シミュレーション

00N8100010D 黒田 大介

中央大学理工学研究科情報工学専攻 鈴木研究室

2002年3月

あらまし

ロボットハンドなどの摩擦を含む接触状態と多関節構造のリアルタイムインタラクティブシミュレーションは仮想空間での物理運動表現やロボットハンドなどの機械系の設計や制御の評価などにおいて有用である。そうしたシミュレーションでは速度や精度を目的に合わせて幅広く調整できるようにロバスト性が必要とされる。本研究では多面体で表現された多関節物体同士の摩擦を含む変化する接触状態のシミュレータを、リアルタイムかつインタラクティブにおこなうのに適していると思われる手法を組み合わせ実現した。このとき、関節構造の制約力や衝突時の撃力や摩擦を計算する場合にはペナルティ法、制約ベース法を組み合わせ手法を、衝突の検出をおこなう場合は連続体による判定を元にした手法を使用した。本シミュレータを使用してロボットハンドが物をつかみあげるバーチャルクレーンゲームのプロトタイプなどを作成し、シミュレータの検証をおこなった。

キーワード: リアルタイムシミュレーション, 剛体, 多関節, Coulomb 摩擦.

目次

第1 序論	1
第2 物体の運動モデル	4
2.1 剛体 (rigid body)	4
2.1.1 位置と方向	4
2.1.2 線形速度	5
2.1.3 角速度	5
2.1.4 剛体の質量	7
2.1.5 力とトルク	7
2.1.6 線形運動量	8
2.1.7 角運動量	8
2.1.8 慣性テンソル	9
2.1.9 剛体の運動方程式	9
2.2 制約つきバネ	9
2.3 粘性摩擦	10
2.4 Coulomb 摩擦	11
2.4.1 動摩擦	11
2.4.2 静止摩擦	11
第3 微分方程式の初期値問題の解法	13
3.1 オイラー法	13
3.2 1次シンプレクティック法	14
3.3 制約ベース法	15
第4 制約つき最適化問題の解法	16
4.1 最適化問題	16
4.2 ペナルティー関数法	16

4.3	最急降下法	17
4.4	直線探索によるステップ幅の決定	17
第5	多面体の衝突	19
5.1	形状の表現	19
5.1.1	Boundary Representation(B-reps)	19
5.2	接触の判定	19
5.2.1	頂点と面の場合	20
5.2.2	辺同士の場合	20
5.3	衝突の取り扱い	21
5.3.1	ペナルティ法と制約ベース法	21
5.3.2	収縮を利用した衝突判定	21
5.4	制約つきバネの設定	23
5.4.1	頂点と面が接触する場合	24
5.4.2	辺と辺が接触する場合	24
5.5	摩擦	28
5.5.1	動摩擦	28
5.6	動摩擦から静止摩擦への移行条件	28
5.6.1	静止摩擦	29
5.6.2	制約つきバネの設定と解除	30
第6	1軸回転関節の取り扱い	31
6.1	1軸回転関節の表現	31
6.2	回転角が変化しない場合	32
6.3	回転角が変化する場合	33
第7	効率的な衝突判定と設計	34
7.1	Bounding Volume	34
7.2	Axis Aligned Bounding Box(AABB) Trees	34
7.3	凸多面体	35
第8	検証	37
8.1	摩擦	37

8.1.1	動摩擦	37
8.1.2	静止摩擦	41
8.2	関節	43
8.3	バーチャルクレーンゲームへの適用例	47
第9	結論	52
	謝辞	54
	参考文献	55

第 1 章

序論

ロボットハンドなどの摩擦を含む接触状態と多関節構造のリアルタイムインタラクティブシミュレーションは

1. 仮想空間での物理運動表現
2. 教師付き学習制御の教師パラメータの入力と評価のためのシミュレーション
3. ロボットハンドなどの機械系の初期の設計における挙動確認

などで有用である。

3Dリアルタイムコンピュータグラフィックスを応用したビデオゲームなどの仮想空間では、説得力のある表現が必要とされることが多い。例えば、ビリヤードなどでの球の転がりや衝突、積み上げられている箱が崩れる様子などである。多関節構造の摩擦を含む接触状態のシミュレーション、例えば物をつかむような場合においてもロバスタかつ高速におこなうことができれば、表現できる事象を広げられる。例えばバーチャルクレーンゲームのような場合では、形状に合わせて物のつかみ方が影響を受けるというような様子が表現されている。その他の仮想空間のプロトタイピングをおこなう場合は汎用的な取り扱いが必要になるため、このような状態も扱えることは意義があるものとする。

実際の機械が存在しない場合、たとえばコスト面から実機を用意できない場合や設計段階で制御までの評価をおこないたい場合などでは、シミュレーションによって制御の評価をおこなうことが有用である。その一方で、制御パラメータなどを学習によってもとめる方法の一つに教師付き学習がある。これは問題とそれに対して与えられた解から背景にある法則を推定するというものである。インタラクティブにシミュレーションがおこなえれ

ば、ユーザがスムーズに行動の結果を確認しつつ教師パラメータを入力することができる。これは、機械の振る舞いなどを直感的に把握することができるため機械の設計の初期段階においても有用と考えられる。

ユーザの入力に対してインタラクティブにシミュレーション結果を出力する上では、計算速度が必要になる。計算速度には

1. ステップ幅
2. 衝突判定の計算量, ロバスト性

などがかわってくる。ステップ幅はシミュレーションをどのくらい細かくおこなうかを示す値で、小さければ小さいほど精度がよいが、小さくするほど計算量も増加するという関係がある。ステップ幅を大きく取った場合は単に精度が悪くなるだけではなく、場合によってはシミュレーション自体の破綻を招く。目的とするシミュレーションする内容、つまり解の精度や物体の形状や量や運動のパラメータによってユーザがステップ幅などをできる限り自由に決められるようにするためには、たとえ大きいステップ幅などシミュレーションをおこないにくい条件が与えられたとしても破綻は避けられるようにすることが望ましい。

また、物をつかむ際は摩擦力の表現が重要である。たとえば四角い箱を両側から挟み込むように持った場合などでは、摩擦が無ければ滑ってしまう持ち方をする。ロボットハンドにおいてもそのようなつかみ方ができれば限られたロボットハンドでより多くの状態の物体の操作をおこなうことができるものと考えられる。

本研究では、

1. 多面体による衝突判定
2. 剛体を1軸回転関節で連結した多関節物体
3. Coulomb 摩擦を含む変化する接触状態

を含むシミュレーションを、リアルタイムかつインタラクティブに解くのに適した手法として、

1. ペナルティ法
2. 制約ベース法

3. 連続体による判定

4. AABB Treeと凸多面体による衝突判定の効率化手法

を組み合わせてソフトウェアを実現した。

ペナルティ法と制約ベース法を組みあらせることで多少大きめのステップ幅でも振動や発散が生じにくいシミュレーションをおこなうことができるが、離散的な衝突判定方法を使用している場合は大きめのステップ幅で物体がその大きさに比べて高速移動すると衝突判定がおこなわれずに、すり抜けたりめり込んだりする場合がある。連続体による判定を組み合わせることでそのような場合でもロバストな衝突の判定を期待できる。

さらに実現したソフトウェアに対して

1. 物理モデルにしたがうかの検証
2. 基本的な事例(バーチャルクレーンゲームなど)への適用

などをおこなった。

第 2 章

物体の運動モデル

2.1 剛体 (rigid body)

本シミュレータでは物体や多関節構造のリンクなどを剛体で表している。剛体は体積を持った変形しない物体である。ここでは剛体について解説をおこなう。

2.1.1 位置と方向

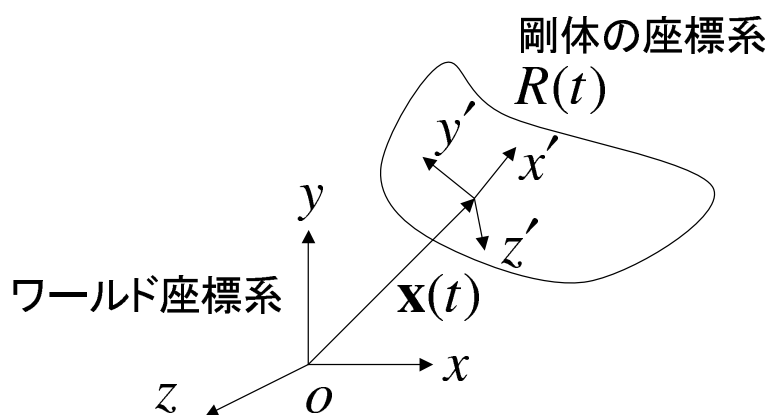


図 2.1. 位置と方向

時刻 t における剛体の位置を平行移動ベクトル $\mathbf{x}(t)$ で表現する。剛体は回転も表現する必要があるため、 3×3 の回転行列 $R(t)$ で表現する。剛体は質点と違って体積を持ち、特有

の形を持つ。剛体は変形と平行移動のみを受けるため、剛体の座標系で剛体の形状は固定である。剛体座標系での幾何学的な記述が与えられた場合、 $\mathbf{x}(t)$ と $R(t)$ は剛体の座標系からワールド座標系への変換に使用する。いくつかの問題を単純化するために、剛体の重心が原点であるように剛体の座標系での剛体の表現をおこなう。 $R(t)$ が重心を中心とした回転を表しているとするならば、剛体座標系上の任意の点 \mathbf{r}_0 のワールド座標系上の位置 $\mathbf{r}(t)$ は次のように表される。

$$\mathbf{r}(t) = R(t)\mathbf{r}_0 + \mathbf{x}(t). \quad (2.1)$$

このとき $\mathbf{x}(t)$ は剛体の位置、 $R(t)$ は剛体の向きととらえることができる。

2.1.2 線形速度

剛体の位置と向きが時間とともにどのように変わるかを、どのように定義するかというのが次の課題である。つまり $\dot{\mathbf{x}}(t)$, $\dot{R}(t)$ をどう与えるべきか定義する必要がある。 $\mathbf{x}(t)$ がワールド座標系での剛体の重心を表すのならば、線形速度 $\mathbf{v}(t)$ を次のように定義する。

$$\mathbf{v}(t) = \dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t). \quad (2.2)$$

2.1.3 角速度

剛体は平行移動に加えて回転もおこなうことができる。この回転はベクトル $\boldsymbol{\omega}(t)$ で表すことができる。 $\boldsymbol{\omega}(t)$ の向きは剛体の回転の軸を表し、 $\boldsymbol{\omega}(t)$ の大きさ $|\boldsymbol{\omega}(t)|$ はどのくらい速く剛体が回転しているかを表す。 $R(t)$ と $\boldsymbol{\omega}(t)$ の関係はどのように表されるのであろうか。線形速度の場合とは違って $\dot{R}(t)$ はそのまま $\boldsymbol{\omega}(t)$ で表すことはできない。ここで、剛体上の点 $\mathbf{r}(t)$ を考え、それが $\boldsymbol{\omega}(t)$ によってどのように動くかを考える。

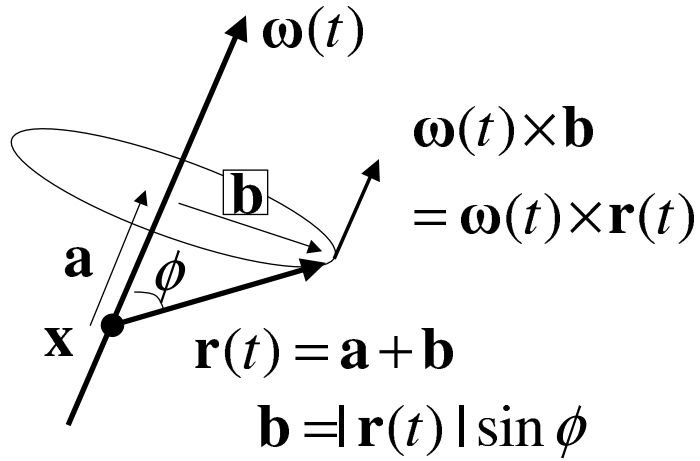


図 2.2. 角速度

図 2.2 から見て取れるように $\mathbf{r}(t)$ は半径 $b = |\mathbf{r}(t)| \sin \phi$ の円の上を動く．速度 $\dot{\mathbf{r}}(t)$ は次のように与えられる．

$$\dot{\mathbf{r}}(t) = \boldsymbol{\omega}(t) \times \mathbf{r}(t). \quad (2.3)$$

ここで $R(t)$ に立ち戻って考えてみる． $R(t)$ の 1 列目は剛体の座標系の x 軸成分である．

$$\begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}. \quad (2.4)$$

この $R(t)$ の 1 列目の導関数は次のように表される．

$$\boldsymbol{\omega}(t) \times \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix}. \quad (2.5)$$

他の列についても同様におこなうと，次のように表すことができる．

$$\left[\boldsymbol{\omega}(t) \times \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} \quad \boldsymbol{\omega}(t) \times \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \quad \boldsymbol{\omega}(t) \times \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} \right]. \quad (2.6)$$

ここで \mathbf{a} というベクトルが与えられたとき、 \mathbf{a}^* は次のような行列と定義する。

$$\begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}. \quad (2.7)$$

このとき、 $\dot{R}(t)$ は次のように表される

$$\dot{R}(t) = \omega(t)^* R(t). \quad (2.8)$$

2.1.4 剛体の質量

剛体は離散的な質点の集まりで表される。これらの質点は $0 \sim N-1$ で番号をつける。 i 番目の質点の質量を m_i として、それぞれの質点は剛体の座標系で一定の位置 \mathbf{r}_{0i} をもつ。時刻 t における i 番目の質点の世界座標系での位置 $\mathbf{r}_i(t)$ は次のように表される。

$$\mathbf{r}_i(t) = R(t)\mathbf{r}_{0i} + \mathbf{x}(t). \quad (2.9)$$

剛体の総質量 M は、次のように与えられる。

$$M = \sum_{i=0}^{N-1} m_i. \quad (2.10)$$

また、このときの剛体の重心は次のように表される。

$$\mathbf{x}(t) = \frac{\sum m_i \mathbf{r}_i(t)}{M}. \quad (2.11)$$

2.1.5 力とトルク

剛体にかかる力を考えるとき、剛体内部の質点にかかる力を考える。 $\mathbf{F}_i(t)$ を i 番目の質点にかかる外力の総和とする。また i 番目の質点にかかる外力によるトルクを次のように定義する。

$$\tau_i(t) = (\mathbf{r}_i(t) - \mathbf{x}(t)) \times \mathbf{F}_i(t). \quad (2.12)$$

トルクは力と違って重心の位置 $\mathbf{x}(t)$ と質点の位置 $\mathbf{r}_i(t)$ の影響を受ける。剛体にかかる外力の総和は、

$$\mathbf{F}(t) = \sum \mathbf{F}_i(t). \quad (2.13)$$

と、外力によるトルクの総和は、

$$\mathbf{F}(t) = \sum \tau_i(t) = \sum (\mathbf{r}_i(t) - \mathbf{x}(t)) \times \mathbf{F}_i(t). \quad (2.14)$$

と表される。

2.1.6 線形運動量

質量 m と速度 v をもつ質点の線形運動量 \mathbf{p} を次のように定める。

$$\mathbf{p} = m\mathbf{v}. \quad (2.15)$$

そのとき剛体の総線形運動量 $\mathbf{P}(t)$ は

$$\mathbf{P}(t) = \sum m_i \mathbf{v}(t) = M\mathbf{v}(t). \quad (2.16)$$

と表される。このことから剛体の加速度は

$$\dot{\mathbf{v}}(t) = \frac{\dot{\mathbf{P}}(t)}{M}. \quad (2.17)$$

と表される。 $\dot{\mathbf{P}}(t) = \mathbf{F}(t)$ であるため、式2.17は次のようになる。

$$\dot{\mathbf{v}}(t) = \frac{\mathbf{F}(t)}{M}. \quad (2.18)$$

2.1.7 角運動量

剛体の重心が原点に固定されている場合を考える。このとき角速度 ω から剛体上の質点の速度 \mathbf{v}_i は

$$\mathbf{v}_i(t) = \omega \times \mathbf{r}_i. \quad (2.19)$$

と表される。このとき、この質点の角運動量は

$$\mathbf{r}_i \times m_i \mathbf{v}_i = \mathbf{r}_i \times m_i (\omega \times \mathbf{r}_i). \quad (2.20)$$

と表される。剛体の全角運動量は次のように表される。

$$L = \sum_i \mathbf{r}_i \times m_i (\omega \times \mathbf{r}_i). \quad (2.21)$$

質量が連続的に分布しているものと考えると、

$$L = \int_{\Omega} \mathbf{r}_P \times (\omega \times \mathbf{r}_P) dm_P, \quad (2.22)$$

である。ここで P は無限小質量 dm_P がある点を表し、 Ω は剛体の占有する体積空間を表す。

2.1.8 慣性テンソル

角運動量 L を次の形で表す.

$$L(t) = I(t)\omega(t). \quad (2.23)$$

このとき $I(t)$ を慣性テンソルと呼ぶ. そうすると式2.22より,

$$I = \begin{bmatrix} \int (y_P^2 + z_P^2) dm_P & -\int (x_P y_P) dm_P & -\int (x_P z_P) dm_P \\ -\int (y_P x_P) dm_P & \int (z_P^2 + x_P^2) dm_P & -\int (y_P z_P) dm_P \\ -\int (z_P x_P) dm_P & -\int (z_P y_P) dm_P & \int (x_P^2 + y_P^2) dm_P \end{bmatrix}, \quad (2.24)$$

と求まる.

2.1.9 剛体の運動方程式

剛体の質量を m と, 重心にかかる力を \mathbf{f} と, 剛体の慣性テンソルを I と, 剛体にかかるトルクを τ と, 剛体の重心の位置を \mathbf{x} と, 剛体の重心の速度を \mathbf{v} とした場合, オイラーの運動方程式から, 剛体の運動は次のように表される.

$$\mathbf{f} = m \frac{d\mathbf{v}}{dt}, \quad (2.25)$$

$$\tau = I \frac{d\omega}{dt} + \omega \times I\omega, \quad (2.26)$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}, \quad (2.27)$$

$$\frac{dR}{dt} = \omega^* R. \quad (2.28)$$

2.2 制約つきバネ

制約つきバネとは伸縮の範囲が定められていて, それ以上伸びも縮みもしないバネである.

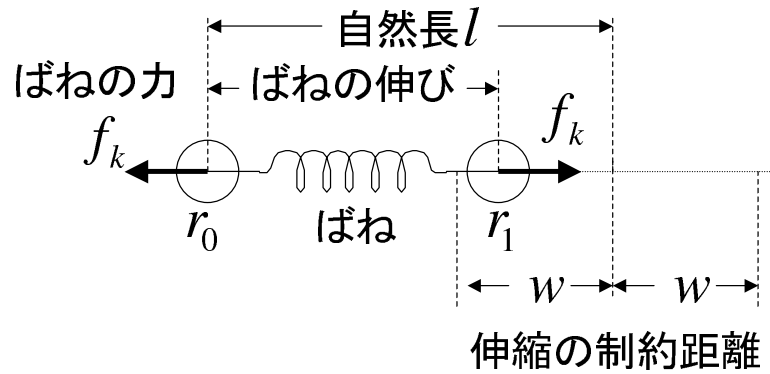


図 2.3. 制約つきバネ

関節や衝突などもすべて制約つきのバネで表すことでさまざまな状況を一貫したシミュレーションでおこなうことができる。

$$l_r = (|\mathbf{r}_1 - \mathbf{r}_0| - l), \quad (2.29)$$

$$\mathbf{f}_k = kr \frac{\mathbf{r}_1 - \mathbf{r}_0}{|\mathbf{r}_1 - \mathbf{r}_0|}, \quad (2.30)$$

$$|l_r| \leq w. \quad (2.31)$$

ここで l はバネの自然長, $\mathbf{r}_0, \mathbf{r}_1$ はバネの端点, w は伸縮の制約距離, k はバネ係数, \mathbf{f}_k はバネの力である。

2.3 粘性摩擦

バネ制約には振動を減衰させる粘性摩擦の項も与えている。

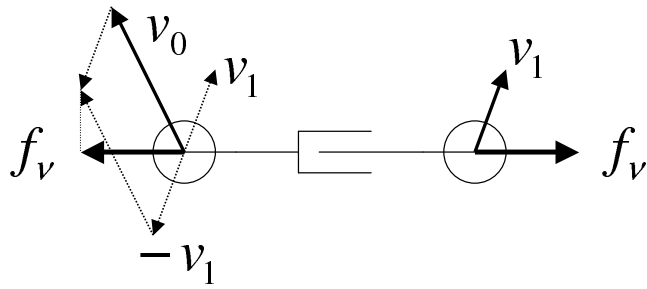


図 2.4. 粘性摩擦

粘性摩擦力 f_v は次のように与えられる.

$$\mathbf{n}_l = \frac{\mathbf{r}_1 - \mathbf{r}_0}{|\mathbf{r}_1 - \mathbf{r}_0|}, \quad (2.32)$$

$$\mathbf{f}_v = \{\nu(\mathbf{v}_1 - \mathbf{v}_0) \cdot \mathbf{n}_l\} \mathbf{n}_l. \quad (2.33)$$

ここで ν は粘性摩擦係数である.

2.4 Coulomb 摩擦

ロボットハンドなどで物を持ち上げる場合などでは、接触状態に応じた摩擦の働きが重要である. Coulomb 摩擦のモデルでは、物体が滑っているときに働く動摩擦と物体が静止しているときに働く静摩擦の二つの状態をとる.

2.4.1 動摩擦

動摩擦は二つの物体が接触面で滑っているときにすべるのを妨げる方向に力がかかる状態である. このときにかかる力を動摩擦力と呼ぶ. 接触面の相対速度を \mathbf{v} , 動摩擦係数を μ , 垂直抗力を f_N とおくと, 動摩擦力 \mathbf{f}_d は

$$\mathbf{f}_d = -\mu f_N \frac{\mathbf{v}}{|\mathbf{v}|}. \quad (2.34)$$

である.

2.4.2 静摩擦

静摩擦は接触面に対して水平方向に力を与えても, その力を打ち消す力が働いて滑らないという状態である. 静摩擦はある程度大きな力がかかると滑り出し, それ以下の力

の場合はかかった力と同じ大きさの力がかかった力の逆方向にかかる．水平方向の力を f_h ，静止摩擦係数を μ' とおくと滑り出す条件は

$$f_h \leq \mu' f_N. \quad (2.35)$$

である．

第 3 章

微分方程式の初期値問題の解法

微分方程式の初期値問題とは，独立変数を t ， t の未知関数を $x(t)$ として $x(t)$ の満たすべき条件

1. 微分方程式: $\frac{d}{dt}x = f(x)$

2. 初期条件: $x(a) = x_0$

を与えて関数 $x(t)$ ($a \leq t \leq b$) を求める問題である．物理運動シミュレーションでは t は時刻に，物体の運動は $x(t)$ にあたる．

3.1 オイラー法

オイラー法とは，ステップ幅 δt を定めて離散的な時刻

$$t' = a + n\Delta t. \tag{3.1}$$

における未知関数 $x(t')$ の近似値 x_n を

$$x_{n+1} = x_n + f(x_n, t)\Delta t. \tag{3.2}$$

によって次々と定めていく方法である．2階の微分方程式の場合は

$$v_{n+1} = v_n + f(x_n, t)\Delta t, \tag{3.3}$$

$$x_{n+1} = x_n + v_n\Delta t. \tag{3.4}$$

で定める． v_n は x_n を微分した値とする．しかし，オイラー法でバネのシミュレーションをおこなうと振動が時間とともに拡大されるというような結果になる．

3.2 1次シンプレクティック法

1次シンプレクティック法ではにおける未知関数 $x(t')$ の近似値 x_n を

$$v_{n+1} = v_n + f(x_n, t)\Delta t, \quad (3.5)$$

$$x_{n+1} = x_n + v_{n+1}\Delta t. \quad (3.6)$$

によって定める。このように処理を変えると計算の量はオイラー法と同程度にもかかわらず、バネなどのハミルトン系ではエネルギーが高精度に保証されるためバネを含むシミュレーションをおこなう場合は、オイラー法よりも好ましい。そのため本研究ではこの手法を採用した。

ここで、バネを例にとってシンプレクティック法の振る舞いを見ていく。バネを次の微分方程式で表した場合、

$$\frac{d^2 x}{dt^2} = -\frac{k}{m}x, \quad (3.7)$$

シンプレクティック法では次のようになる。

$$v_{n+1} = v_n - \frac{k}{m}x_n\Delta t, \quad (3.8)$$

$$x_{n+1} = x_n + v_{n+1}\Delta t. \quad (3.9)$$

ここで $v_n = \frac{x_n - x_{n-1}}{\Delta t}$ のため、この差分方程式は次の形に変形できる。

$$x_{n+2} + \frac{k}{m}(\Delta t^2 - 2)x_{n+1} + x_n = 0. \quad (3.10)$$

この差分方程式の補助方程式を次のように与え、

$$s^2 + \frac{k}{m}(\Delta t^2 - 2)s + 1 = 0 \quad (3.11)$$

s_1, s_2 はこの方程式の根とする。 s の根は次のとおりである。

$$s = \left(1 - \frac{k}{2m}\Delta t^2\right) \pm \sqrt{\left(1 - \frac{k}{2m}\Delta t^2\right)^2 - 1} \quad (3.12)$$

$$= 1 - \frac{k}{2m}\Delta t^2 \pm \sqrt{\frac{k^2}{4m^2}\Delta t^4 - \frac{k}{m}\Delta t^2} \quad (3.13)$$

$\frac{k}{2m}\Delta t^2 \leq 2$ の場合、 $\cos(\theta) = 1 - \frac{k}{2m}\Delta t^2$ とおくと

$$s = \cos \theta \pm i \sin \theta \quad (3.14)$$

と表せるためこの差分方程式の一般解は

$$x_n = A \cos(k\theta + B), \quad (3.15)$$

である。ここでA, Bは初期値によって決まる係数である。 $\frac{k}{2m} \Delta t^2$ が小さい場合は解の数列の振幅は1であるが、 $\frac{k}{2m} \Delta t^2$ が2に近づくと振動が大きくなりシミュレーションとしては好ましくない結果が得られる。 $\frac{k}{2m} \Delta t^2 > 2$ の場合、一般解は

$$x_n = C_1 s_1 + C_2 s_2, \quad (3.16)$$

で与えられる。ここでC, Cは初期値によって決まる係数である。 s_1, s_2 のどちらかは絶対値が1より大きいいため、解の数列は発散する。

つまり、質量が小さく、バネ係数が大きい場合のバネのシミュレーションなどは過度な振動や発散が発生しやすくなりステップ幅を細かくする必要がある。

3.3 制約ベース法

単純にバネのみで衝突や関節のシミュレーションをおこなうと衝突時に過度のめりこみが発生したり、関節が離れるなど不自然なシミュレーション結果となる可能性がある。そこで、制約バネなどをつかい、その制約を満たすような力を発生させてシミュレーションをおこなう制約ベース法を採用する。

ここで s を物体の位置や姿勢などをあらわすベクトルと、 u はその導関数、 f_c は制約力、 f_{ext} は制約力以外の力とする。運動方程式は

$$\frac{ds}{dt} = u, \quad (3.17)$$

$$\frac{du}{dt} = f_c(t) + f_{ext}(t). \quad (3.18)$$

と与えられるため、これを次のように離散化する。

$$s^{t+\Delta t} = s^t + u^{r+\Delta t} \Delta t, \quad (3.19)$$

$$u^{t+\Delta t} = u^t + (f_c(t + \Delta t) + f_{ext}(t)) \Delta t. \quad (3.20)$$

制約を満たす制約力 f_c は最急降下法などで求める。

第 4 章

制約つき最適化問題の解法

制約ベース法では制約力 f_c を求める際に最適化問題の解法の一つである最急降下法を解いている。本章では制約つき最適化問題の解法について解説する。

4.1 最適化問題

制約つき最適化問題を次のように定義する。 $f: R^n \rightarrow R, g_i: R^n \rightarrow R (i = 1, \dots, m), h_j: R^n \rightarrow R (j = 1, \dots, l)$ としたとき

$$g_i \leq 0 \quad (i = 1, \dots, m) \quad (4.1)$$

$$h_j = 0 \quad (j = 1, \dots, l) \quad (4.2)$$

の、条件で $f(x)$ を最小にする \hat{x} を求める問題である。式 4.1 を不等式制約条件、式 4.2 を等式制約条件と呼ぶ。また $f(x)$ を目的関数、 $g_i(x)$ や $h_j(x)$ を制約関数と呼ぶ。

制約条件のついていない最適化問題は無制約最適化問題とよばれる。これは、 $f: R^n \rightarrow R$ のとき $f(x)$ を最小にする \hat{x} を求める問題である。

4.2 ペナルティ関数法

不等式条件付の最適化問題を解く方法の一つにペナルティ関数法がある。この手法では x が許容領域 $X = \{x | g_i \leq 0 (i = 1, \dots, m)\}$ から外れれば外れるほど大きな値 (ペナルティ) をとる外点ペナルティ関数 $P(x, \rho)$ を設定する。例えば、

$$P(x, \rho) = \rho \sum_{i=1}^m (\max\{0, g_i(x)\}), \quad (4.3)$$

などである。ペナルティー関数 $P(x, \rho)$ を導入して、 $f(x) + P(x, \rho)$ を最小にする無条件最適化問題に変換して、不等式条件つき最適化問題を解く。本研究ではペナルティー関数法で制約力を求めている。

4.3 最急降下法

無条件最適化問題の解法の一つである、最急降下法について解説する。

最小化する関数 $f(x)$ が与えられ、関数 $f(x)$ が微分可能とする。 $x = \bar{x}$ が与えられたとき、すべての $\alpha \in (0, \delta)$ において $f(\bar{x} + \alpha d) < f(\bar{x})$ を満たす $\delta > 0$ が存在するとき、 d を f の \bar{x} における降下方向と呼ぶ。特に、

$$\nabla f(\bar{x})^T d = \lim_{\alpha \rightarrow 0^+} \frac{f(\bar{x} + \alpha d) - f(\bar{x})}{\alpha} < 0, \quad (4.4)$$

のとき d は降下方向である。

最急降下法では $|d| = 1$ として局所的に最小になる降下方向 d の方向に \bar{x} を修正し最終的に解に収束させようとする。 $\nabla f(\bar{x}) \neq 0$ ならばその方向は、

$$\bar{d} = \frac{-\nabla f(\bar{x})}{|\nabla f(\bar{x})|} = \frac{-\nabla f(\bar{x})}{\sqrt{f(\bar{x})^T f(\bar{x})}}, \quad (4.5)$$

である。もし $\nabla f(\bar{x}) = 0$ のときは \bar{x} 極小値である。 $\bar{d} = -\nabla f(\bar{x})$ は \bar{x} における最急降下方向と呼ばれる。これは $\bar{d} \nabla f(\bar{x}) = -\nabla f(\bar{x})^T \nabla f(\bar{x}) < 0$ であることから確かめられる。最急降下法のアルゴリズムは次のように記述することができる。

1. x^0 を与え、 $k \leftarrow 0$ とする。
2. $d^k = -\nabla f(x^k)$ とする。もし $d^k = 0$ ならば終了する。
3. 直線探索によってステップ幅 α^k を選ぶ。すなわち、 $f(\bar{x} + \alpha d^k)$ が最小となるような α を求め、 $\alpha^k \leftarrow \alpha$ とする。
4. $x^{k+1} \leftarrow x^k + \alpha^k d^k, k \leftarrow k+1$ としたのち、2にもどる。

4.4 直線探索によるステップ幅の決定

最急降下法のアルゴリズムでは、それぞれの反復においてステップ幅 α^k を直線探索によって決定している。すなわち一方向の最適化問題、 $f(\bar{x} + \alpha d^k)$ が最小となるような α を求める問題を解いている。

本研究では Armijo の条件によってステップ幅を決定している。これは、 $0 < \xi < \frac{1}{2}$ であるような定数 ξ に対して、

$$f(x^k + \alpha d^k) \leq f(x^k) + \xi \alpha \nabla f(x^k)^T d^k, \quad (4.6)$$

を満たす $\alpha > 0$ を選ぶというものである。Armijo の条件は確実に収束することが保証されている ([11])。

第 5 章

多面体の衝突

5.1 形状の表現

物体の衝突を取り扱うには物体の内側と外側の区別をつける必要がある。内外が定義されている物体をソリッドオブジェクトと呼ぶ。

5.1.1 Boundary Representation(B-reps)

Boundary Representation(B-reps)とはソリッドオブジェクトを面境界, つまり頂点(vertex), 辺(edge), 面(face)で表現する方法である。面は頂点のリストを持っており, 面の表方向から見たときに反時計回りの順番で並ぶものとする。処理を単純化するために本研究では三角形から構成される多面体のみを取り扱うものとする。

ある辺が与えられてその辺に隣接する面を参照する場合などでは, 単純なデータ構造を使用していると計算量が増えるため, 次のようなデータ構造を使用している。B-repを表現するデータ構造は有効辺 e を中心としたデータ構造で辺 e は辺の頂点 v_{e0}, v_{e1} , 隣接する面 f_{e0}, f_{e1} , の参照をもつ。面は隣接する3つの辺 e_{f0}, e_{f1}, e_{f2} , 頂点 v_{f0}, v_{f1}, v_{f2} の参照を持つ。頂点は隣接する n 個の辺の参照 $e_{v0}, e_{v1}, \dots, e_{vn}$ を持つ。

5.2 接触の判定

多面体同士の衝突判定は, 面と頂点, 辺同士の軌跡の交差判定によっておこなう。

5.2.1 頂点と面の場合

頂点 v と有向面 F が与えられたとき、時刻 t における頂点の座標を $\mathbf{p}(t)$ 、面の方程式を $\mathbf{n}(t)\mathbf{x} + \mathbf{n}_0(t) = 0$ とする。面と頂点の符号付距離を $d = \mathbf{n}^T \mathbf{p} + \mathbf{n}_0$ と定義して時刻 $t, t + \Delta t$ における距離 $d^t, d^{t+\Delta t}$ を求める。面と頂点の距離を線形補間すると $d(t) = d^t + t/\Delta t(d^{t+\Delta t} - d^t)$ と表すことができる。交差する時刻を $t_c = \Delta t d^t / (d^t - d^{t+\Delta t})$ とする。このとき $t_c < 0$ あるいは $t_c > \Delta t$ ならば、時刻 t から $t + \Delta t$ までの間に頂点と面が交差することはない。さらに、時刻 t_c において面の中に頂点の有無で交差を判定するが計算の安定性から正確に面の上に頂点があるかではなく、面と法線 n からなるプリズムで判定をおこなう。

5.2.2 辺同士の場合

2つの辺 e_0 と e_1 が与えられたとき、時刻 t における両端点の座標を $\mathbf{a}_0(t), \mathbf{b}_0(t)$ と $\mathbf{a}_1(t), \mathbf{b}_1(t)$ とする。 g_0, g_1 を e_0, e_1 を通る直線とする。

頂点と面の場合のときのように、辺の間に有向な距離を定義することはできない。しかし、辺が衝突しない限りは e_0 の端点は面 F の同じ側にある。そこで、それぞれ時刻 t_0, t_1 に対応する直線 g_0, g_1 間の距離 d_0, d_1 を次のように定義する。

$$d_i = \frac{\det(\mathbf{b}_0 - \mathbf{a}_0, \mathbf{b}_1 - \mathbf{a}_1, \mathbf{a}_0 - \mathbf{a}_1)}{|(\mathbf{b}_0 - \mathbf{a}_0) \times (\mathbf{b}_1 - \mathbf{a}_1)|} \quad (5.1)$$

ここで $\mathbf{a}_0, \mathbf{b}_0, \mathbf{a}_1, \mathbf{b}_1$ の評価は $i = 0, 1$ に対する時刻 t_i でおこなう。つぎに、交差判定のためにこの距離を線形補間する。

$$t_c = \frac{d_0}{d_0 - d_1} \quad (5.2)$$

t_c は直線 g_0, g_1 が交差する時刻である。もし、 $t_c < t_0$ あるいは $t_c > t_1$ のときは直線 g_0, g_1 は交差しないため、 e_0, e_1 も交差しない。それ以外の場合では交差点が辺の中に存在するか判定する必要がある。以前のように t_c における辺の位置を線形補間する。安定性のために交差点を計算するのではなく（線形補間では線が必ずしも交差しないため）直線上の一番近い点の組をもとめる。

5.3 衝突の取り扱い

5.3.1 ペナルティ法と制約ベース法

本研究では衝突時の撃力，接触力のシミュレーションをおこなう際，ペナルティ法¹と制約ベース法を組み合わせた手法を使用している．ペナルティ法では物体の微小なめり込みを許し，めり込んでいる点に反発するようなバネとダンパを設定することで，接触力を与える．このときのバネ係数は一つの値を選ばなければならないが適切な値を選ぶ必要がある．例えば，バネ係数が小さすぎると物体は過度にめり込み不自然なシミュレーションとなってしまう．一方でバネ係数が大きすぎるとシミュレーションが発散してしまうためステップ幅を細かくしなければいけない．しかしステップ幅を細かくすると計算コストが高くなってしまう．

制約ベース法のみで衝突の計算をおこなう場合はバネ係数のようなパラメータの設定の難しさはないが，ペナルティ法に比べると計算コストが高いが，めり込みの発生を防止することができる．

ここではある一定量以上のめり込みがあった場合は接触力を制約ベース法で求め，めり込みがある一定量以下になったらペナルティ法を適用するという方法をとる．主にめり込みが深くなるのは物体が衝突したときで，物体が接触しているときのめり込み量はそれと比べると浅い．そのため，衝突時は多少計算コストが高くなるがその後接触状態に落ち着くと計算コストはペナルティ法の分のみになる．

5.3.2 収縮を利用した衝突判定

ペナルティ法を使用している関係でめり込みがあっても衝突判定がおこなえるような手法が必要である．ただ，そのめり込みの量は制約ベース法を併用しているためめり込み量には限りがある．

そこで，めり込み量の分だけ収縮させた多面体を考える．

¹こちらのペナルティ法も制約問題を無制約問題に変えて解いているため，ペナルティ関数法と同じ意味と思われる．本稿では衝突などでバネをシミュレーションすることで制約を表現する方法をペナルティ法，制約力を求めるために1ステップの間に無制約問題化した制約問題を解く方法をペナルティ関数法と呼ぶことにする．

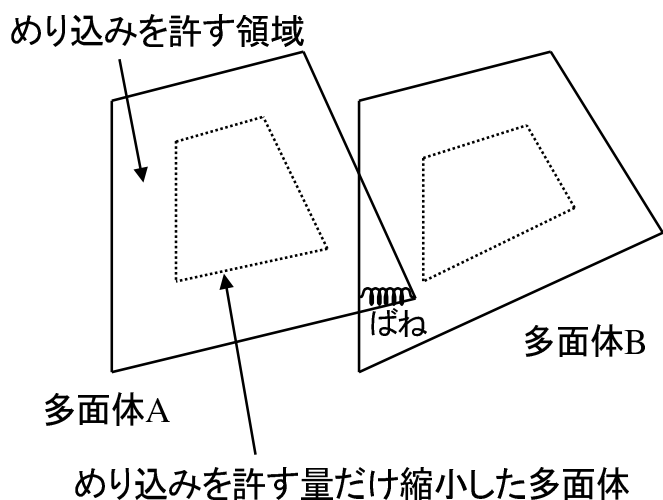


図 5.1. 多面体の衝突の取り扱い

図のように時刻 t_0 から t_1 にかけての収縮した図形の軌跡 L_0 と，時刻 t_1 での収縮した図形から元の図形への軌跡 L_1 ，これら 2 つの軌跡を利用して判定をおこなう。

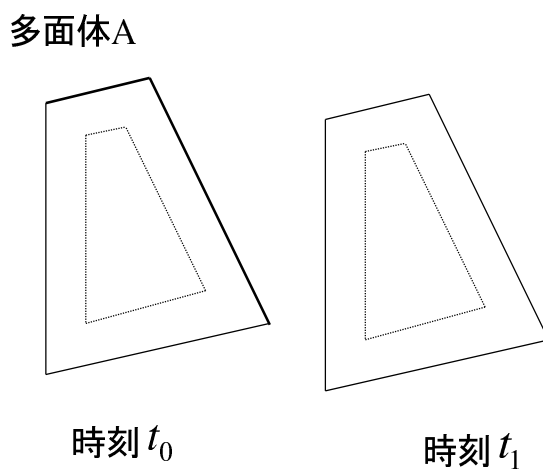
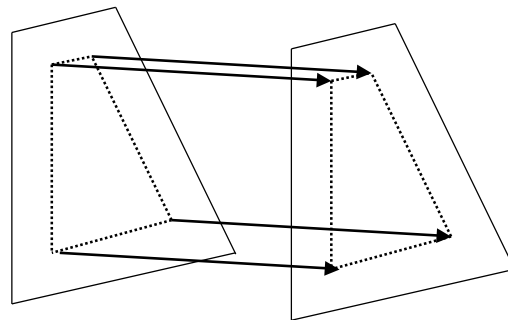


図 5.2. 多面体の衝突の取り扱い

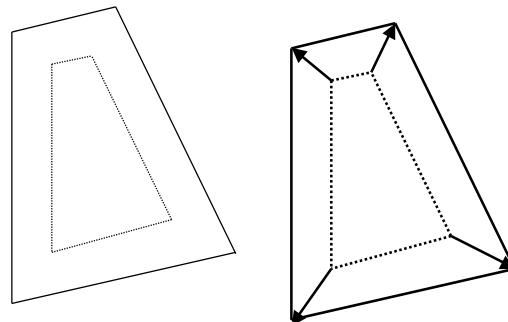
まず，収縮した図形の軌跡 L_0 は物体の過度なめり込みの検出をおこなう．連続体による判定のため，ステップごとの移動量が大きくなっても過度なめり込みのロバストな検出をおこなうことができる．



めり込み防止用軌跡

図 5.3. 多面体の衝突の取り扱い

収縮した図形から元の図形への軌跡 L_1 は物体の接触判定をおこなう。これにより、現在の面が接触しているかを判定することができる。



接触判定用軌跡

図 5.4. 多面体の衝突の取り扱い

5.4 制約つきバネの設定

本研究では多面体の頂点と面、あるいは辺と辺が接触しているときに制約つきバネを設定するが、その設定の仕方によっては物体が他の物体に沈み込んでしまったり、辺に引っ

かかってしまうなどの不自然なシミュレーション結果が生成されてしまう。そのためどのような位置に制約つきバネを設定するかが問題になる。物体が滑るときの制約つきバネの設定の仕方を頂点と面が接触する場合と辺と辺が接触する場合で変えている。

5.4.1 頂点と面が接触する場合

頂点と面の場合は面の法線方向に頂点を押し戻すような力をかける。制約つきバネを S としたとき、制約つきバネ S と頂点 v の接点は \mathbf{p}_v 、制約つきバネ S と面 f との接点は \mathbf{p}_f とする。 \mathbf{p}_v は頂点の座標に等しい。面 f をとおる平面の方程式を $\mathbf{n}\mathbf{x} + \mathbf{n}_0 = 0$ とおくと \mathbf{p}_f は

$$\mathbf{p}_f = -(\mathbf{n} \cdot \mathbf{p}_v + \mathbf{n}_0)\mathbf{n} + \mathbf{p}_v \quad (5.3)$$

である。

5.4.2 辺と辺が接触する場合

辺と辺の場合は辺に隣接する面との関係からさらに細かく場合わけをおこなう。まず、辺 e_0 に着目し、仮の押し戻し方向のベクトル n_{e_0} をもとめる。このとき辺 e_0 と e_1 上で相手の直線までの距離がもっとも短くなる点の座標を p'_{e_0}, p'_{e_1} とおく。

$$n'_{e_0} = e_0 \times e_1, \quad (5.4)$$

$$l = p'_{e_1} - p'_{e_0}, \quad (5.5)$$

$$n_{e_0} = \begin{cases} -n'_{e_0} & (l \cdot n'_{e_0} \geq 0) \\ n'_{e_0} & (l \cdot n'_{e_0} < 0) \end{cases} \quad (5.6)$$

辺 e_0 に隣接する面の法線を $n_{e_{00}}, n_{e_{01}}$ とおき、この方線の中に n_{e_0} があるかの判定をおこないその結果を q_0 に保存する。このとき辺 e_0 の両端点の座標を a_0, b_0 とする。

$$q_0 = \begin{cases} \text{if } (n_{e_0} \times n_{e_{00}}) \cdot (b_0 - a_0) \geq 0 \text{ and } (n_{e_0} \times n_{e_{01}}) \cdot (b_0 - a_0) \leq 0 \text{ then } & \text{True} \\ \text{else} & \text{False} \end{cases} \quad (5.7)$$

同様にして e_1 に対応する q_1 も求める。

q_0, q_1 から

1. q_0, q_1 がともに *True* の場合
2. q_0, q_1 のうち片方が *True* の場合

3. q_0, q_1 がともに *False* の場合

これらの場合に分けて処理をおこなう。

q_0, q_1 がともに *True* の場合

辺 e_0 と e_1 上で相手の直線までの距離がもっとも短くなる点の座標 p'_{e_0}, p'_{e_1} をそのままバネの接続点 p_{e_0}, p_{e_1} とする。

q_0, q_1 のうち片方が *True* の場合

本来物体同士にめり込みがない場合は q_0, q_1 が *False* になるような形では接触はおこらない。たとえば q_0 が *False* の場合は、辺 e_0 が交差点から辺 e_1 に隣接する面の内側に入り込んでしまうためである。本来ならば辺 e_0 の別の部分が他の辺と交差しているか辺 e_0 の頂点がいずれかの面と交差しているはずである。この場合に“ q_0, q_1 がともに *True* の場合”の処理を適用するとどうなるであろうか。辺 e_1 があってそれに隣接する面が平行か山形に折れているものとする。

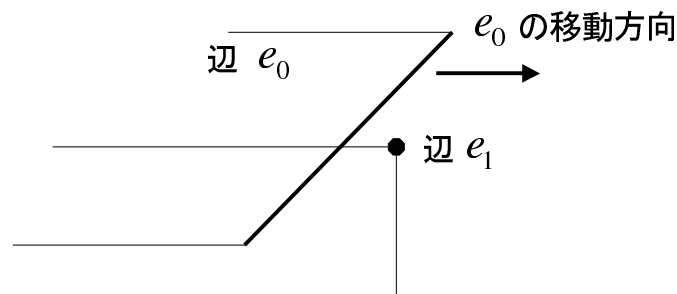


図 5.5. 本来交差しないはずの辺での衝突

このとき立方体が辺 e_1 に隣接する面の上を滑った後、立方体の隣接面に対して垂直な辺 e_0 が e_1 に交差した場合、本来はそのまま滑っていくはずが立方体の進行方向を妨げるような方向に力がかかり、つまづいたような結果が出力されてしまう。

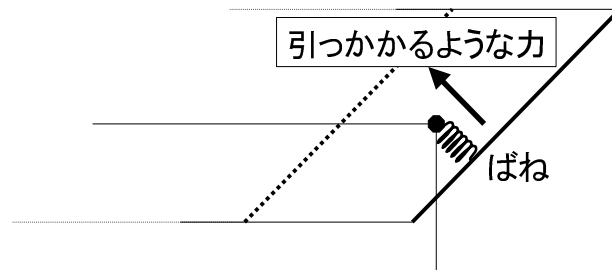


図 5.6. 引っかかり

しかし、だからといってこのような場合ただ単純に制約つきバネを無効にするだけだと、2つの立方体が辺のところのみで積み重なっているような場合では振動して沈み込むなどの不安定な出力となってしまう。

そこで q_0 が *False* の場合は辺 e_0 の隣接面 f_{e00}, f_{e01} の法線 n_{e00}, n_{e01} から押し戻す方向を決定する。この二つの法線と n_{e_0} との内積をもとめ小さいほうの隣接面を選択する。

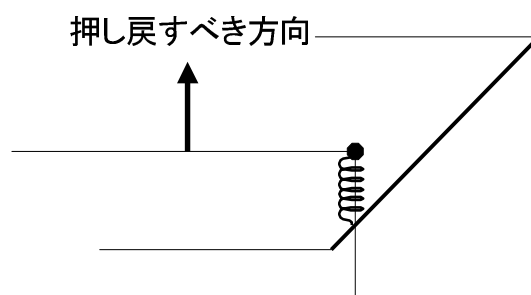


図 5.7. 適切な押し戻し方向

p'_{e_0} を e_0 の接続点 p_{e_0} として、選択された隣接面と p_{e_0} から選択された隣接面の法線方向に伸びる直線の交点を p_{e_1} とする。

q_1 が *False* の場合も同様である。

q_0, q_1 の両方が *False* の場合

q_0, q_1 の判定は e_0, e_1 それぞれに着目しておこなうため両方が *True* になる場合もありえる。この場合は p'_{e_0}, p'_{e_1} と辺 e_0, e_1 の頂点 a_0, b_0, a_1, b_1 との距離から取り扱いを変える。 p'_{e_0} が a_1, b_1 に近いかの判定をおこないその結果を q'_0 に保存する

$$q'_0 = \begin{cases} \text{if } |p'_{e_0} - a_1| \leq \sigma \text{ or } |p'_{e_0} - b_1| \leq \sigma \text{ then } & \text{True} \\ \text{else} & \text{False} \end{cases} \quad (5.8)$$

q'_0, q'_1 から

1. q'_0, q'_1 がともに *True* の場合
2. q'_0, q'_1 のうち片方が *True* の場合
3. q'_0, q'_1 がともに *False* の場合

これらの場合に分けて処理をおこなう。

p'_{e_0}, q'_1 がともに *True* の場合

q'_0 に近い頂点の法線を n_{v1} , q'_1 に近い頂点の法線を n_{v0} とする。 $n'_{e_0} \cdot n_{v1} < n'_{e_1} \cdot n_{v0}$ のときは“ q_0, q_1 のうち片方が *True* の場合”で e_0 が *False* のとき設定されるバネのみを使用する。そうでない場合は、 e_1 が *False* のとき設定されるバネのみを使用する。

p'_{e_0}, q'_1 のうち片方が *True* の場合

q'_0 が *True* の場合は“ q_0, q_1 のうち片方が *True* の場合”で e_0 から設定されるバネのみを使用する。

q'_1 が *True* の場合も同様。

p'_{e_0}, q'_1 がともに *False* の場合

“ q_0, q_1 のうち片方が *True* の場合”にしたがって e_0 が *False* の場合と e_1 が *False* の場合で作られる2つバネを設定する。

5.5 摩擦

本手法で摩擦力を求めるときは動摩擦時と静止摩擦時の場合で制約つきバネなどの取り扱いを変えている。

5.5.1 動摩擦

動摩擦時は節5.4の規則にしたがって制約つきバネを更新するとともに、動摩擦力をかける。

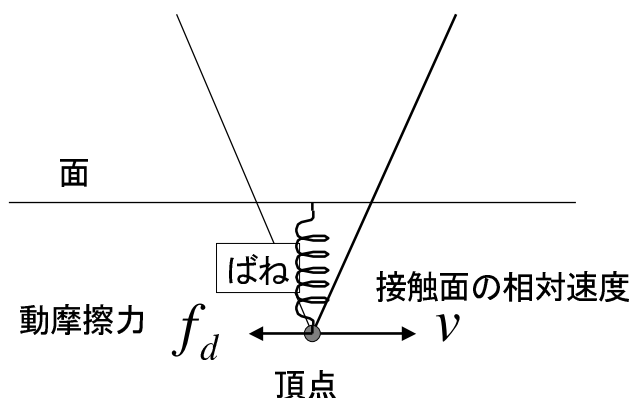


図 5.8. 動摩擦力

ペナルティ法で計算する領域の場合ではバネの力を垂直抗力とみなして取り扱う。接触面の水平方向の相対速度を v ，動摩擦係数を μ ，バネ係数を k ，めり込み量を l_v とすると、動摩擦力 f_d は

$$f_d = -\mu k l_v \frac{v}{|v|}. \quad (5.9)$$

とあらわされる。

5.6 動摩擦から静止摩擦への移行条件

動摩擦から静止摩擦に移る条件は Coulomb 摩擦のモデルでは静止したときであるが、安定した静止の判定のために、相対速度が 0 かどうかの判定するのではなく次のような条件

で判定をおこなう。時刻 t における接触面の相対速度 $\nu(t)$ と、1ステップ前の時刻 $t - \Delta t$ における接点の1ステップ当たりの変位の量を $\nu(t - \Delta t)$ とすると

1. $\nu(t) \cdot \nu(t + \Delta t) < 0$

2. $|\nu(t)| < \epsilon$

である。 ϵ は微小な値である。

$\nu(t)$ はバネの接点のパラメータから求める。頂点と面の接触の場合は面の座標を $f_0(t), f_1(t), f_2(t)$ として面上の接点 p_f を次のようにあらわす。

$$p_f = \alpha(t)f_1(t) + \beta(t)f_2(t) + (1 - \alpha(t) - \beta(t))f_0(t). \quad (5.10)$$

面の接点をパラメータのベクトル $(\alpha(t), \beta(t))$ であらわし、 $(\alpha(t + \Delta t) - \alpha(t), \beta(t + \Delta t) - \beta(t))$ を相対速度 $\nu(t)$ とする。辺同士の接触の場合は辺 e_0 と e_1 の時刻 t における両端点の座標を $a_0(t), b_0(t)$ と $a_1(t), b_1(t)$ として e_0, e_1 の接点の p_0, p_1 座標を次のようにあらわす。

$$p_0 = \alpha(t)b_0(t) + (1 - \alpha(t))a_0(t), \quad (5.11)$$

$$p_1 = \beta(t)b_1(t) + (1 - \beta(t))a_1(t). \quad (5.12)$$

この場合も、同様に取り扱う。

つぎに1と2の条件を2つ併用しておこなう理由について解説する。これは、1のみの判定ではステップ幅が細かくなつたときに接点のパラメータのベクトルが回転するようなことが起こりやすくなり、静止と判定されないことがあるためである。2のみの判定ではステップ幅が大きいときに動摩擦で接点が振動し、 ϵ が小さい場合は静止の判定がおこなわれない場合がある。かといって ϵ を大きくすると大きな速度で静止してしまう。2つの条件を設けることで広いステップ幅で安定した判定を期待することができる。

5.6.1 静止摩擦

静止摩擦時は制約つきバネの接点を保持し元の位置に戻そうとする力をかける。静止摩擦から動摩擦に移る判定はバネの伸びを垂直方向 l_v と水平方向 l_h に分解し、そこから垂直抗力と水平方向にかかる力を求めて判定をおこなう。

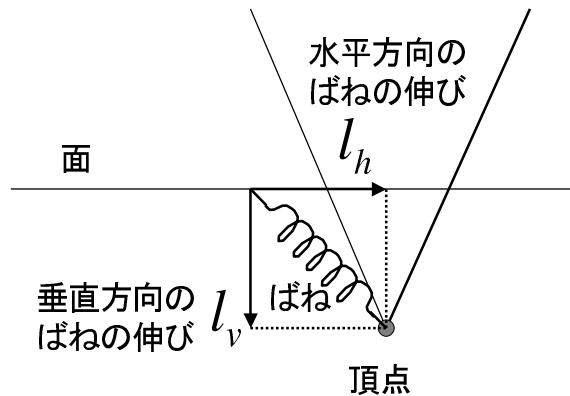


図 5.9. 静止摩擦の取り扱い

バネの接点を p_0, p_1 . 接触面の法線を N とした場合次のように求められる.

$$d = p_1 - p_0, \quad (5.13)$$

$$l_v = |d - (N \cdot d)N|, \quad (5.14)$$

$$l_h = k(N \cdot d). \quad (5.15)$$

静止摩擦から動摩擦に移る条件はバネ係数を k , 静止摩擦係数を μ' , 接触面の法線を N とおいた場合, 次の判定式が真のときである.

$$kl_h + |f_c - (f_c \cdot N)| > \mu' \{kl_v + (f_c \cdot N)\}. \quad (5.16)$$

5.6.2 制約つきバネの設定と解除

静止摩擦の取り扱いを見てわかるように制約つきバネの接続位置の情報は複数ステップに渡って保持しなければならない. これは次のように取り扱っている

1. 頂点と面(あるいは辺と辺)が交差していて制約つきバネがない場合は制約つきバネを設定
2. 頂点と面が交差していて制約つきバネがある場合は制約つきバネを保持
3. 頂点と面が交差していなくて制約つきバネがある場合は制約つきバネを解除

第 6 章

1 軸回転関節の取り扱い

ロボットハンドやロボットアームはいくつかの剛体リンクを回転することのできる関節によって結合されたリンク構造からなる。本シミュレータでおこなっている1軸回転関節の取り扱い方について述べる。

6.1 1 軸回転関節の表現

関節は2つの剛体を結ぶ4つの制約つきバネであらわす。

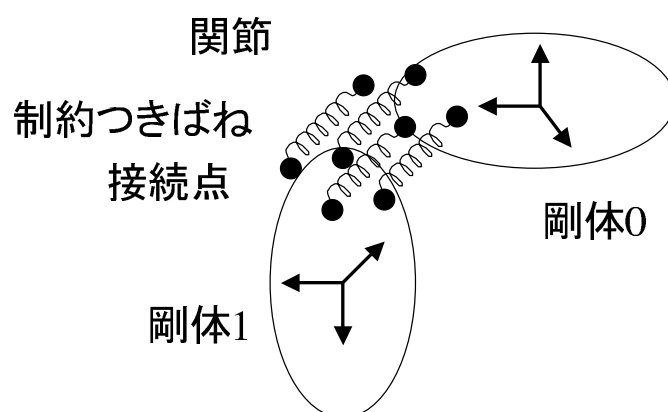


図 6.1. 制約つきバネによる関節の表現

ここでバネが剛体に接続している点を接続点と呼ぶ。4つのバネの自然長は0でバネの接続点は一致しようとする。4つのバネのうち回転軸上にある2つのバネを軸バネと呼ぶ。その接続点を軸接続点と呼ぶ。それ以外の2つのバネはトルクバネと呼ぶものとする。その接続点をトルク接続点と呼ぶ。

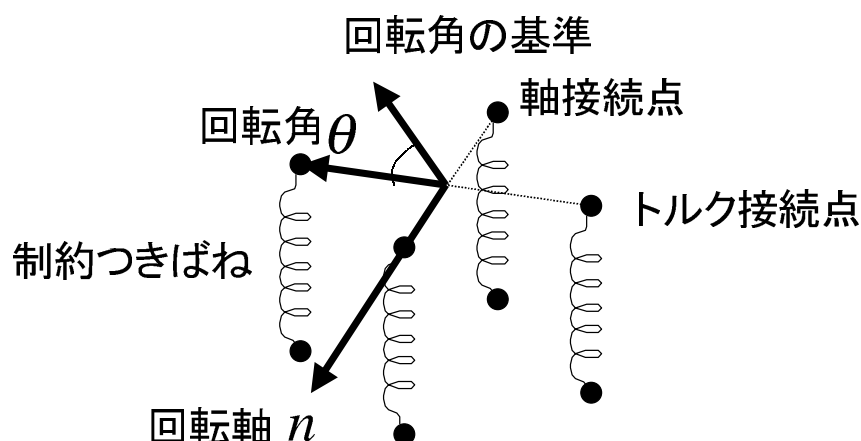


図 6.2. 関節のパラメータ

1軸回転関節は回転角 θ をもっており回転の基準とトルク接続点を結んだ線との角度は回転角に相当する。また、回転角の限界をあらわす最大回転角 θ_{\max} 、最小回転角 θ_{\min} をもつ。この接続点は関節の角度によって位置を変え、この点にはトルクに相当する力がかかる。回転角が変化しない場合と変化する場合でトルクの取り扱いをかえている。

6.2 回転角が変化しない場合

回転の限界や静止摩擦状態の場合は回転角が変化しない。この場合は元のトルク接続点の位置を保持しつづける。それにより接続している剛体には回転角を保持するような力がかかる。

6.3 回転角が変化する場合

2つの剛体を剛体 B_0, B_1 とする. 関節にかかるトルクを τ , 回転軸を n , 回転軸から接続点の距離を r , そのときのトルクバネを s_{t0}, s_{t1} , s_{t0} の接続点のうち剛体0に接続するものを p_{t00} , 剛体1に接続するものを p_{t01} とする. s_{t1} の接続点も同様にする. このときの処理の手順は次のとおりである.

1. 回転の基準と剛体1の接続点を結んだ線との角度をもとめそれを回転角 θ とする.
2. $\theta > \theta_{\max}$ ならば θ を θ_{\max} と, $\theta < \theta_{\min}$ ならば θ を θ_{\min} として回転角が変化しない場合の処理に移る.
3. トルクに相当する力 $f_t = \frac{\tau(n \times r)}{2r^2}$ をトルク接続点に加える.

第 7 章

効率的な衝突判定と設計

7.1 Bounding Volume

複雑な物体の正確な衝突判定をおこなう場合、面、辺、頂点の交差判定の計算コストが問題になる。そこで、Bounding Volume と呼ばれる衝突判定の対象の物体を内包する単純な形状でおおまかな判定をおこない、この形状が交差しているときのみより詳細な判定をおこなうことで計算コストの削減を図る。

このときに使用する形状は球、直方体などさまざまなものがあるが本研究では

1. Axis Aligned Bounding Box(AABB) Trees
2. 凸多面体

を使用している。

7.2 Axis Aligned Bounding Box(AABB) Trees

Axis Aligned Bounding Box(AABB) とは辺が軸に対して整えられている直方体である。ここではすべての AABB の辺はワールド座標の軸にそろえられているものとする。

AABB は頂点の x, y, z 軸の最小値 $P = (a, b, c)$ と x, y, z 軸の最大値 $Q = (d, e, f)$ で表現する。

2つの AABB を B_0, B_1 とおき、それらを $P_0 = (a_0, b_0, c_0)$, $Q_0 = (d_0, e_0, f_0)$, $P_1 = (a_1, b_1, c_1)$, $Q_1 = (d_1, e_1, f_1)$ と表す。AABB の交差の判定は、次の条件式が真ならば交差と判定する。

$$a_0 \leq d_1 \text{ and } a_1 \leq d_0 \text{ and}$$

$$\begin{aligned}
b_0 \leq e_1 \quad \text{and} \quad b_1 \leq e_0 \quad \text{and} \\
c_0 \leq f_1 \quad \text{and} \quad c_1 \leq f_0.
\end{aligned}
\tag{7.1}$$

バウンディングボリュームを階層化させると交差の可能性がない物体同士の判定の拒否を非常に効率的におこなうことができる。AABBを階層化したものを Axis Aligned Bounding Box Trees(AABB Trees) と呼ぶ。

AABB Treesの構築は再帰的な細分化をトップダウンにおこなう。ここで図形を構成する基本単位をプリミティブとよぶ。本研究では頂点、辺、面である。再帰のステップではプリミティブを内包するもっとも小さいAABBを作成し、よくプリミティブをよく分ける面で集合を分割する。本研究では分割する面の方向は再帰のステップにおいて x, y, z 軸を順番に切り替えており、分割する面の位置はプリミティブを囲むAABBの重心の平均値を通る位置である¹。

2つの物体の交差判定は再帰的に根元からおこなう。2つのノードの組を訪れる際、AABBが重なっているものに対してのみさらに探索をおこなう。もし2つのノードがプリミティブだった場合は交差の判定をおこない結果を親に返す。もし片方のノードが葉でもう片方が子を持つノードならば葉と子を持つノードのそれぞれの子と交差判定をおこなう。もし、両方とも子を持つノードならば小さいAABBを持つノードと大きいAABBをもつノードの子との交差判定をおこなう。本研究ではAABBの大きさは x, y, z 軸の幅の総和としている。

階層化した Bounding Volume の計算コストは次のように表される

$$T = N_v C_v + N_p C_p + N_u C_u \tag{7.2}$$

T は衝突判定にかかる総コスト、 N_v は重なっている Bounding Volume の組の交差数、 N_p は衝突しているプリミティブの組の数、 N_u は更新する必要があるノードの数、 C_v は Bounding Volume の組の交差判定コスト、 N_p はプリミティブの組の交差判定コスト、 N_u はノードの更新にかかるコストである。

7.3 凸多面体

もし片方のノードが葉でこれのAABBが極端に大きく、もう片方が子を持つノードでAABBの大きさが非常に小さいうえに多くの子を持っている場合を考える。大きい葉をもつノードのAABBの中にもう片方のノードが内包されてしまう場合、すべての葉でプリミ

¹文献 [10] には、一般に中央値で分割をおこなうとよいと記されている。

タイプ同士の判定がおこなわれてしまう。これは、接触の状態によって極端に計算コストが変わることになり、リアルタイムシミュレーションの上では必ずしも好ましくない。そこで、もし片方のノードが葉でもう片方が子を持つノードの場合は、葉のほうをAABBより元の形状に近い形の凸多面体で近似して判定をおこない、交差している場合においてのみ子のノードとの交差判定をおこなうことで、判定にかかる計算コストの平均化を図る。

多面体は α 個の面の法線ベクトル n_i につき2つの面をつくる。これらの面は β 個のプリミティブの頂点の座標 v_j から次のように作る。

$$n_{\min} = -\min(-n_i \cdot v_j) \quad (7.3)$$

$$n_{\max} = -\max(n_i \cdot v_j) \quad (7.4)$$

$$-n_i \mathbf{x} + n_{\min} = 0 \quad (7.5)$$

$$n_i \mathbf{x} + n_{\max} = 0 \quad (7.6)$$

多面体とAABBの判定は次のような手順でおこなう。 n_i の x, y, z 軸の成分を n_{ix}, n_{iy}, n_{iz} として、面から見たときのAABBの最も遠い頂点と最も近い頂点を法線の x, y, z 軸の成分から選択する。

$$\begin{cases} \text{if } n_{ix} > 0 \text{ then } a' = a, d' = d \\ \text{else } a' = d, d' = a \end{cases}, \quad (7.7)$$

$$\begin{cases} \text{if } n_{iy} > 0 \text{ then } b' = b, e' = e \\ \text{else } b' = e, e' = b \end{cases}, \quad (7.8)$$

$$\begin{cases} \text{if } n_{iz} > 0 \text{ then } c' = c, f' = f \\ \text{else } c' = f, f' = c \end{cases}, \quad (7.9)$$

$$P' = (a', b', c'), \quad (7.10)$$

$$Q' = (d', e', f'). \quad (7.11)$$

このとき次の判定式がすべての面の法線 n_i において真ならば、さらに子のノードと詳細な判定をおこなう。

$$(n_i \cdot P' + n_{\max} \leq 0) \text{ and } (n_i \cdot Q' + n_{\min} \geq 0) \quad (7.12)$$

第 8 章

検証

本シミュレータが実際にモデルに従ったシミュレーションをおこなうか検証をおこなった。

8.1 摩擦

8.1.1 動摩擦

物体が床の上を滑るときの軌跡が理想の曲線に接近するか，停止位置から静止摩擦へ移行する判定条件が妥当かどうかの検証をおこなった。

初速 v_0 ，動摩擦係数 μ ，重力係数 g ，としたときの物体の重心の軌跡 x は，初期 ($t = 0$) の重心の位置を x_0 ，重心の初速度を v_0 とすると

$$x = -\frac{1}{2}\mu g t^2 + v_0 t + x_0, \quad \left(0 < t \leq \frac{v_0}{\mu g}\right) \quad (8.1)$$

$$x = \frac{v_0^2}{2\mu g} + x_0. \quad \left(t > \frac{v_0}{\mu g}\right) \quad (8.2)$$

である。

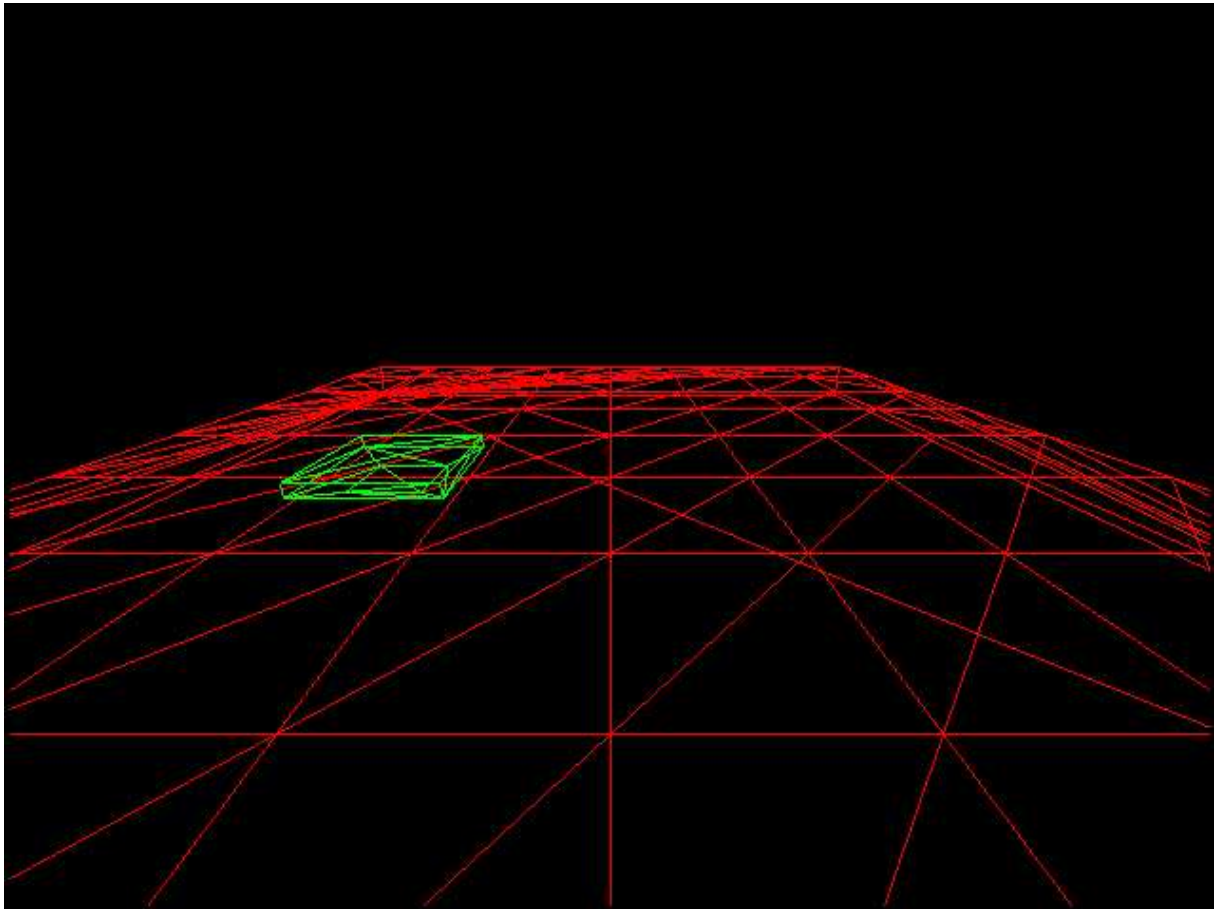


図 8.1. 摩擦のシミュレーションのワイヤーフレーム表示

図8.1にあるように、緑色の線で表されている物体を赤い線で表されている床の上で滑らせた場合の摩擦の挙動に関して検証をおこなう。

物体の質量を $m = 1.0(\text{kg})$ 、重力加速度を $g = 9.8(\text{m/s}^2)$ 、動摩擦係数を $\mu = 0.5$ 、静止摩擦係数を $\mu' = 0.5$ として、初速 v_0 を x 軸方向に $9.8, 19.6, 29.4(\text{m/s})$ を与えたときの軌跡の x 座標の値をステップ幅 Δt が $1/50(\text{s})$ のときについて求めた。

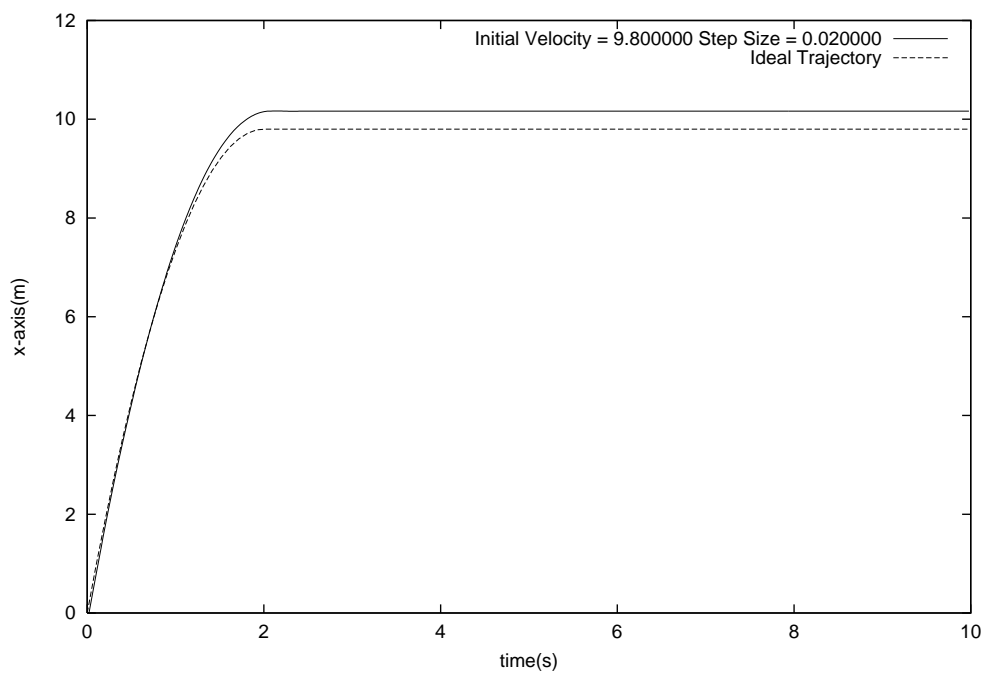


図 8.2. 初速と動摩擦の関係 $v_0 = 9.8(m/s)$

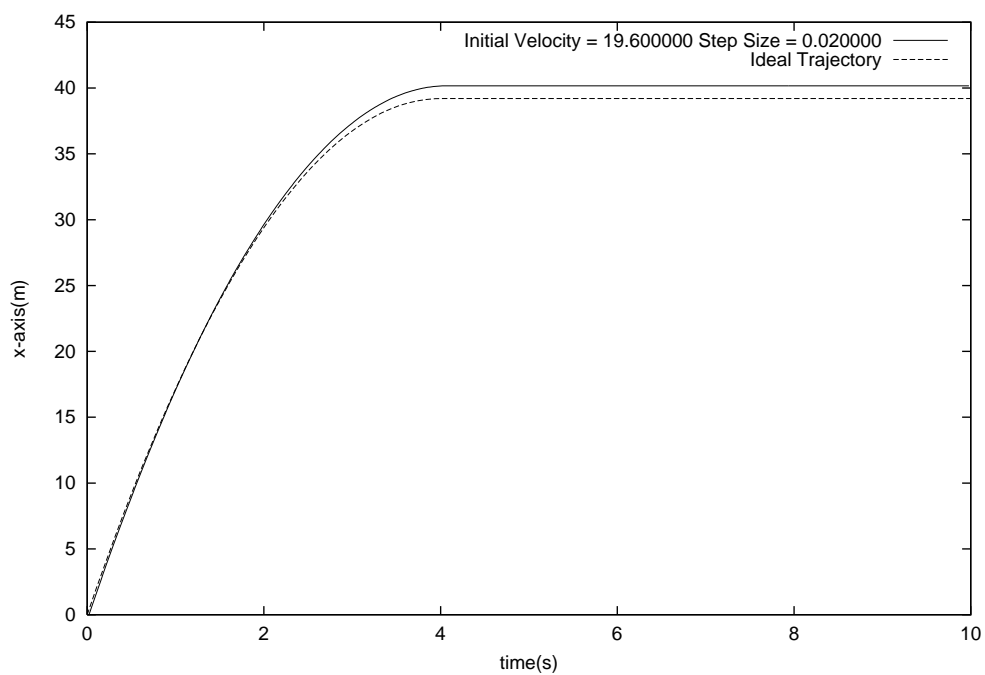


図 8.3. 初速と動摩擦の関係 $v_0 = 19.6(m/s)$

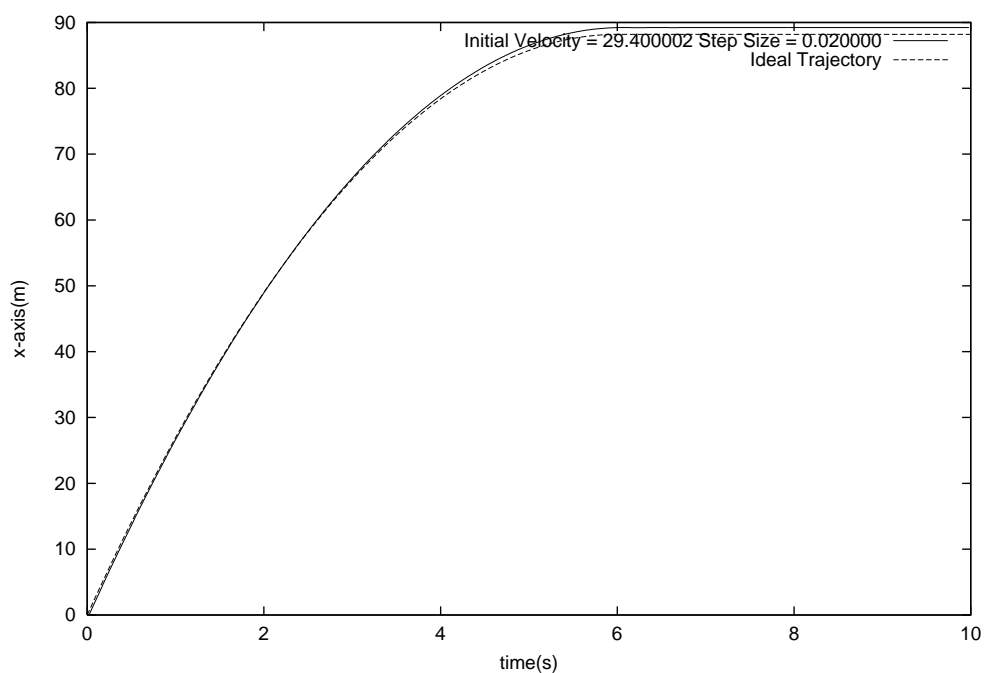


図 8.4. 初速と動摩擦の関係 $v_0 = 29.4(m/s)$

シミュレーションから求められた軌跡は理想のものとほぼ近いが多少の誤差が見られる。
次に初速 v_0 を $19.6(m/s)$ としてステップ幅 Δt を $1/50, 1/100, 1/200(s)$ と変化させた場合のグラフを図 8.5 に示す。

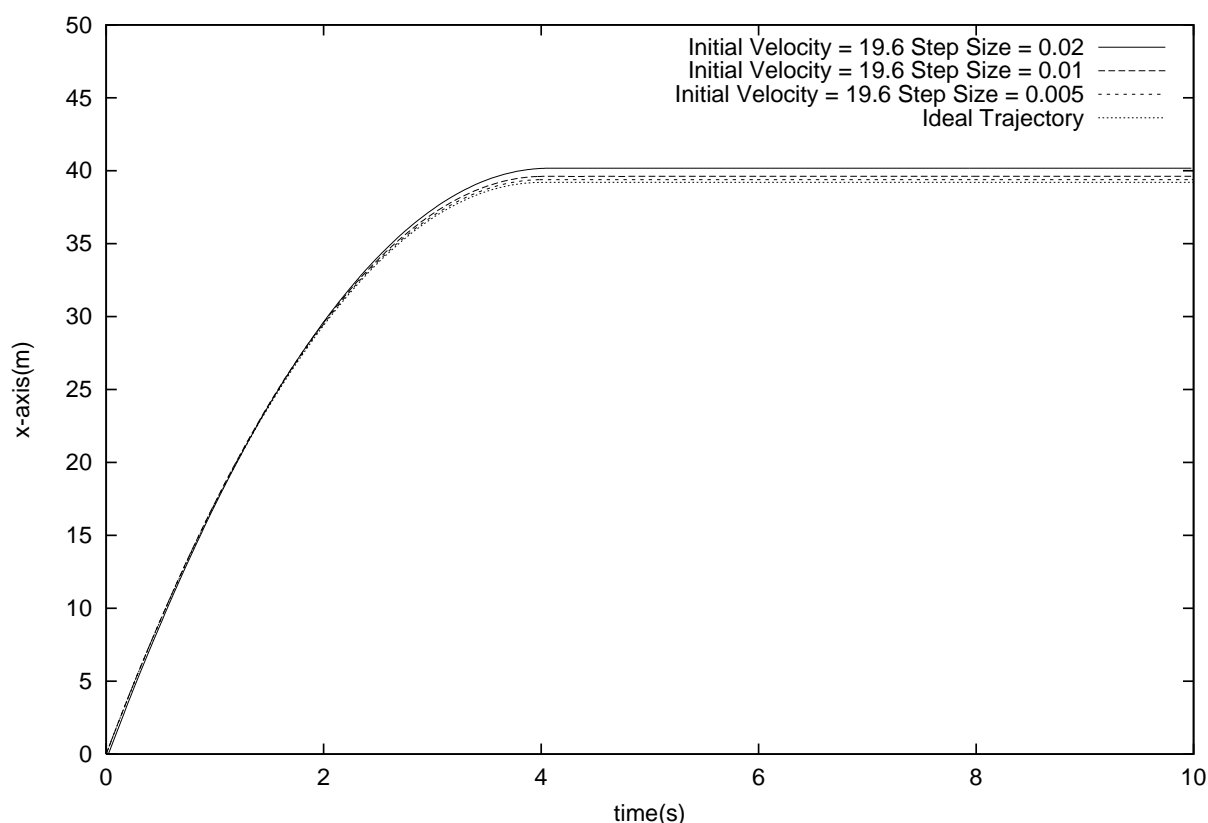


図 8.5. ステップ幅と動摩擦の関係

ステップ幅を細かくすることで軌跡が理想のものに近づく傾向が見られる。

8.1.2 静止摩擦

静止している物体に対して横にすべる方向に力を与えたときの滑り出す判定についての検証をおこなった。物体の質量を $m = 1.0(kg)$ ，重力加速度を $g = 9.8(m/s^2)$ ，動摩擦係数を $\mu = 0.5$ ，静止摩擦係数を $\mu' = 0.5$ として，ステップ幅 Δt が $1/50, 1/100, 1/200, 1/400(s)$ のときの水平方向に与える力 f を $1.2250, 2.4500, 2.9400, 3.4300, 3.9200, 4.9000, 9.8000$ と変化させた。力は $0 \sim 2(s)$ までの間を線形に f まで増加させ，時刻 $0s$ と $15s$ での位置が $0.1m$ 以上変化した場合は動状態 (“dynamic”)，それより小さい場合は静止状態 (“static”) と判定した。この場合滑り出す力の大きさ f_h の理論値は次のように与えられる。

$$f_h = \mu' mg = 0.5 \times 1.0 \times 9.8 = 4.9(kgm/s^2). \quad (8.3)$$

物体が滑るかどうかの判定結果を表 8.1 に示す。

表 8.1. 静止摩擦の力とステップ幅との関係

力	2.4500	3.4300	3.6750	3.9200	4.4100	4.9000	9.8000
$\Delta t = 1/50$	static	static	static	static	dynamic	dynamic	dynamic
$\Delta t = 1/100$	static	static	static	static	dynamic	dynamic	dynamic
$\Delta t = 1/200$	static	static	static	static	dynamic	dynamic	dynamic
$\Delta t = 1/400$	static	static	static	static	dynamic	dynamic	dynamic

滑り出す力は、ステップ幅が変わってもほぼ一定であるが、理論値より小さい値で滑り出している。ここで $f = 3.92$ の場合のステップ幅ごとの x 軸の軌跡を図 8.6 に示す。

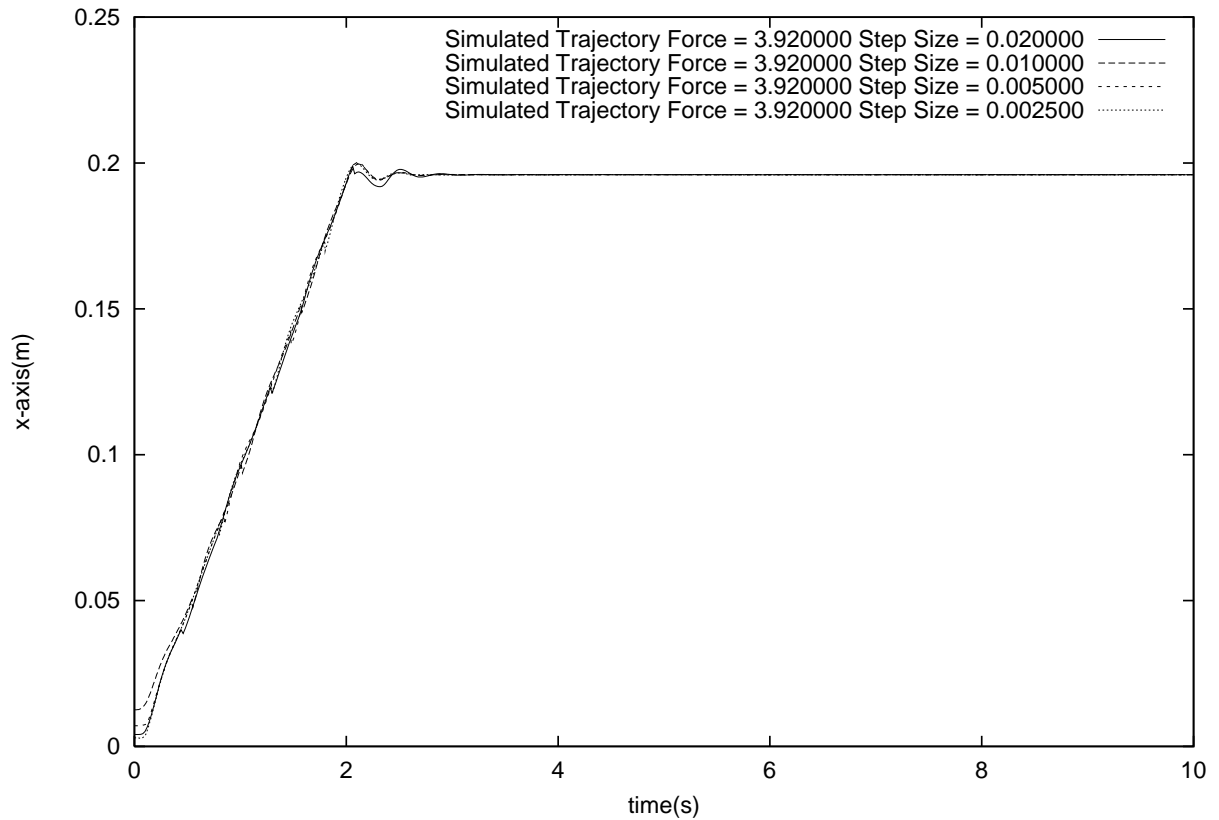


図 8.6. $f = 3.92$ の場合の x 軸の軌跡

緩やかに位置がずれた後、振動してからつりあう位置に落ち着いている。バネで接触をあらわしているため、物体が振動し、垂直抗力も振動する結果理論値より多少小さい値で

滑り出すものと考えられる.

8.2 関節

図8.7のような天井から吊り下げられた2重振り子型のロボットを考える.

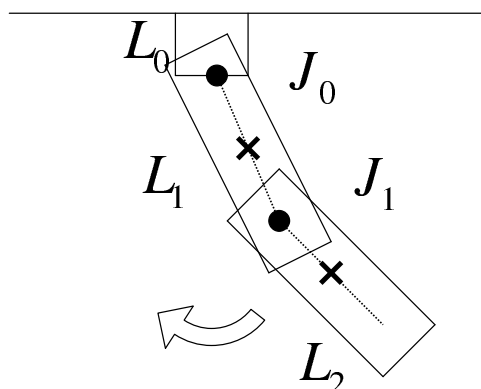


図 8.7. 2重振り子型のロボット

リンク L_0 , L_1 , L_2 からなり, 関節 J_0 は L_0 と L_1 を結び, 関節 J_1 は L_1 と L_2 を結んでいる. リンク L_0 は天井に固定されており, リンク L_1 , L_2 はそれぞれの物体の座標系から見たときに x, y, z 軸の幅が $2, 2, 6(m)$ の軸にそろえられた直方体で, 軸の位置を $(0, 0, 0)$ としたときの直方体の頂点座標の最小値を $(-1, -1, -1)$, 最大値を $(1, 1, 6)$ で与える. リンク L_2 の軸は L_1 から見て $(0, 0, 5)$ の位置に設定する. このとき回転軸の方向は x 軸とする. リンク L_1, L_2 の質量 m は $5(kg)$ で, これらは一様な密度を持つものとする. 質量 m の直方体が軸 x で回転するときの慣性モーメントは次のとおりであるから,

$$\int_{z_{\min}}^{z_{\max}} \int_{y_{\min}}^{y_{\max}} \int_{x_{\min}}^{x_{\max}} \frac{m(y^2 + z^2)}{(z_{\max} - z_{\min})(y_{\max} - y_{\min})(x_{\max} - x_{\min})} dx dy dz$$

$$= \frac{m(y_{\max}^2 + y_{\max}y_{\min} + y_{\min}^2 + z_{\max}^2 + z_{\max}z_{\min} + z_{\min}^2)}{3}. \quad (8.4)$$

リンク L_1, L_2 の慣性モーメント I は $\frac{100}{3}$ である. 動摩擦トルク f_{mu} は回転の逆方向に $0.3(Nm)$ の大きさでかかるものとする.

J_0 から L_1 の重心までの距離は $s = 2(m)$ で J_1 から L_2 の重心までの距離も同じ値である。
 J_0 から J_1 までの距離を $l = 4(m)$ と, J_0, J_1 にかかるトルクを τ_0, τ_1 と, J_0, J_1 の回転角を θ_0, θ_1 と表すと, ラグランジュの運動方程式は次のように表される.

$$\begin{cases} \frac{d}{dt} \left\{ (I + ml^2)\dot{\theta}_0 + I(\dot{\theta}_0 + \dot{\theta}_1) + mls(2\dot{\theta}_0 + \dot{\theta}_1) \right\} \cos \theta_1 \\ + (ms + ml)g \sin \theta_0 + msg \sin(\theta_0 + \theta_1) = \tau_0 \\ \frac{d}{dt} \left\{ I(\dot{\theta}_0 + \dot{\theta}_1) + mls\dot{\theta}_0 \cos \theta_1 \right\} \\ + mls\dot{\theta}_0(\dot{\theta}_0 + \dot{\theta}_1) \sin \theta_1 + msg \sin(\theta_0 + \theta_1) = \tau_1 \end{cases} \quad (8.5)$$

初期の J_0, J_1 の回転角を $\pi/4, 0$ としてラグランジュの運動方程式を1次のシンプレクティック法から求めた軌跡と, 本シミュレータを使用して求めた軌跡を比較する. 本シミュレータではステップ幅 Δt を $0.02(s)$ として軸バネの伸縮の制約距離 w を $0.4, 0.2, 0.1, 0.05(m)$ と動かした場合の関節の角度と, 軸バネの伸縮の制約距離 w を $0.05(m)$ としてステップ幅を $0.02, 0.01, 0.005(s)$ と変化させた場合の関節角の軌跡を求めた. ラグランジュの運動方程式を解く場合はステップ幅 Δt を $0.005(s)$ として求めた.

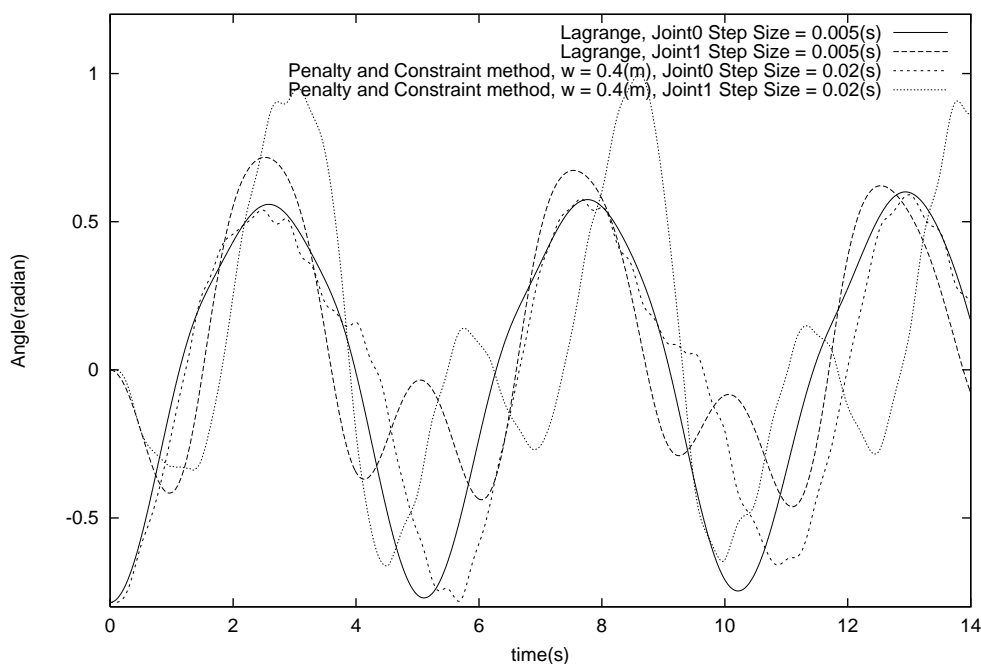


図 8.8. 関節の角度の比較 $w = 0.4, \Delta t = 0.02$

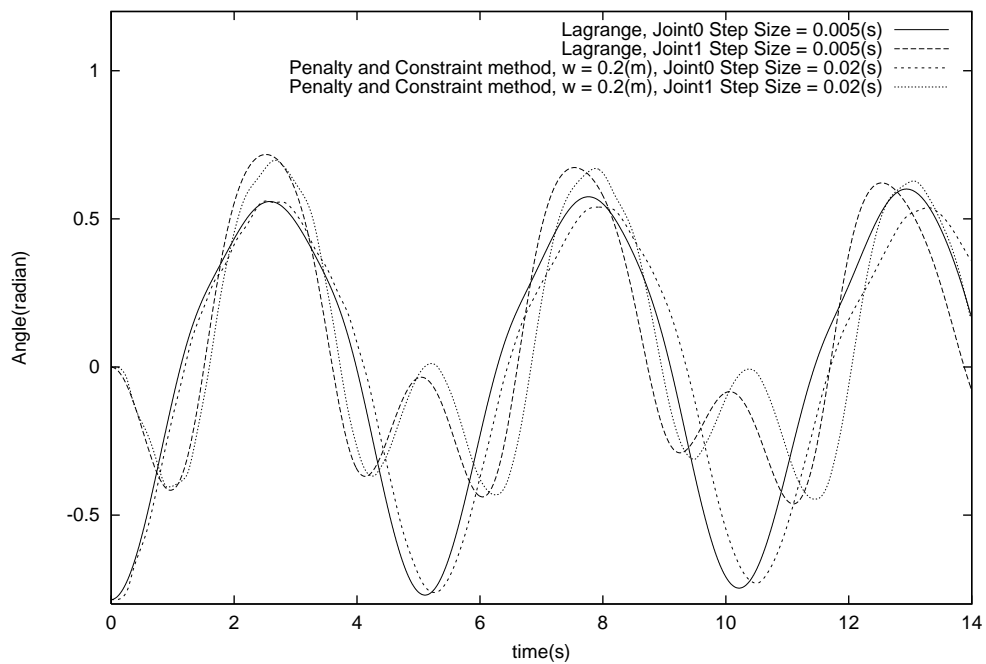


図 8.9. 関節の角度の比較 $w = 0.2, \Delta t = 0.02$

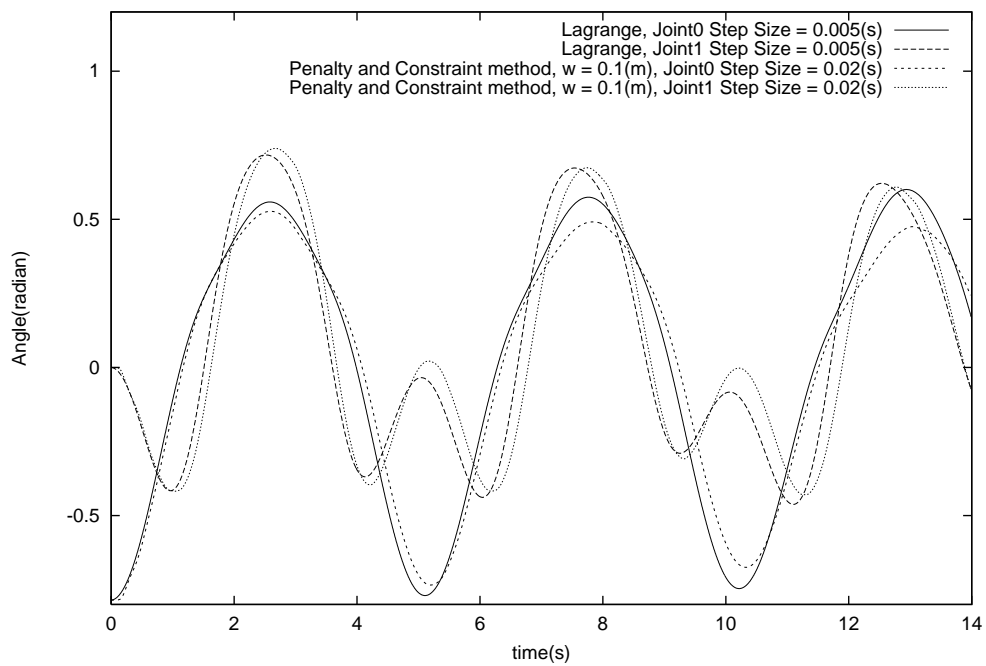


図 8.10. 関節の角度の比較 $w = 0.1, \Delta t = 0.02$

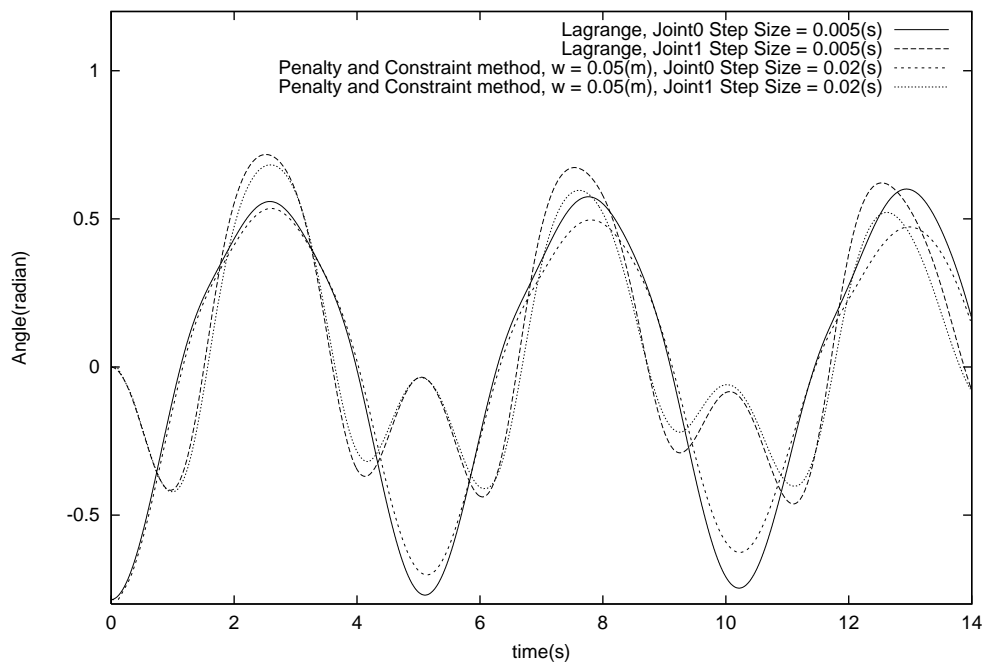


図 8.11. 関節の角度の比較 $w = 0.05, \Delta t = 0.02$

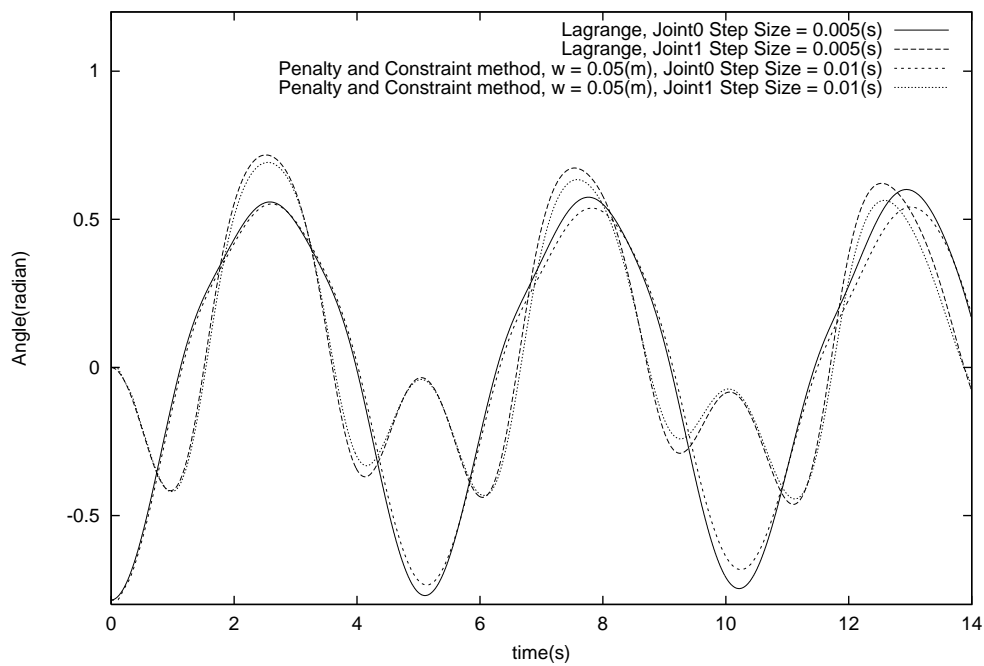


図 8.12. 関節の角度の比較 $w = 0.05, \Delta t = 0.01$

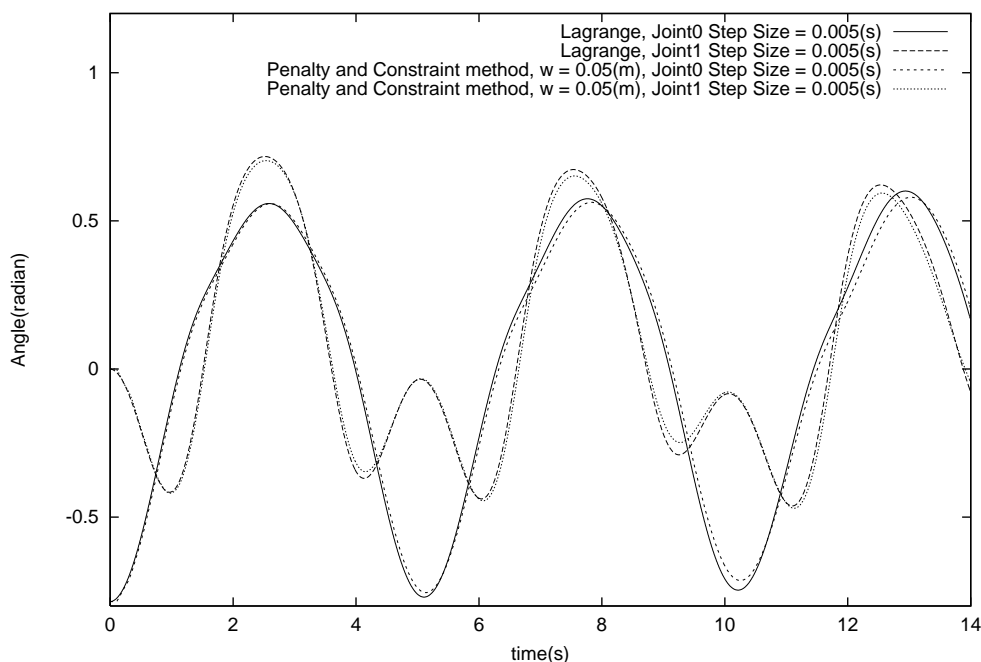


図 8.13. 関節の角度の比較 $w = 0.05, \Delta t = 0.005$

$\Delta t = 0.02(s), w = 0.4(m)$ の場合はバネの振動のために関節がゆれ、その結果角度にも細かい振動が現れている。これは軸バネの伸縮の制約距離 w を小さくするとともにこの振動はなくなっていくが、まだ軌跡の誤差は大きい。さらにステップ幅 Δt を細かくすると、本シミュレータで求めた軌跡はラグランジュの運動方程式から求めたものに近づく様子が見られる。

8.3 バーチャルクレーンゲームへの適用例

摩擦を含む接触状態と多関節構造のシミュレーションの応用例としてバーチャルクレーンゲームのプロトタイプを作成した。浮遊する赤い蟹型のクレーンで箱やハート型の物体などを持ち上げることができる(図8.14, 8.15)。形状によってつかみ方などが変わる様子も見て取れる。図8.16ではクレーンを急激に動かしたために箱がすべって落下する様子が見て取れる。

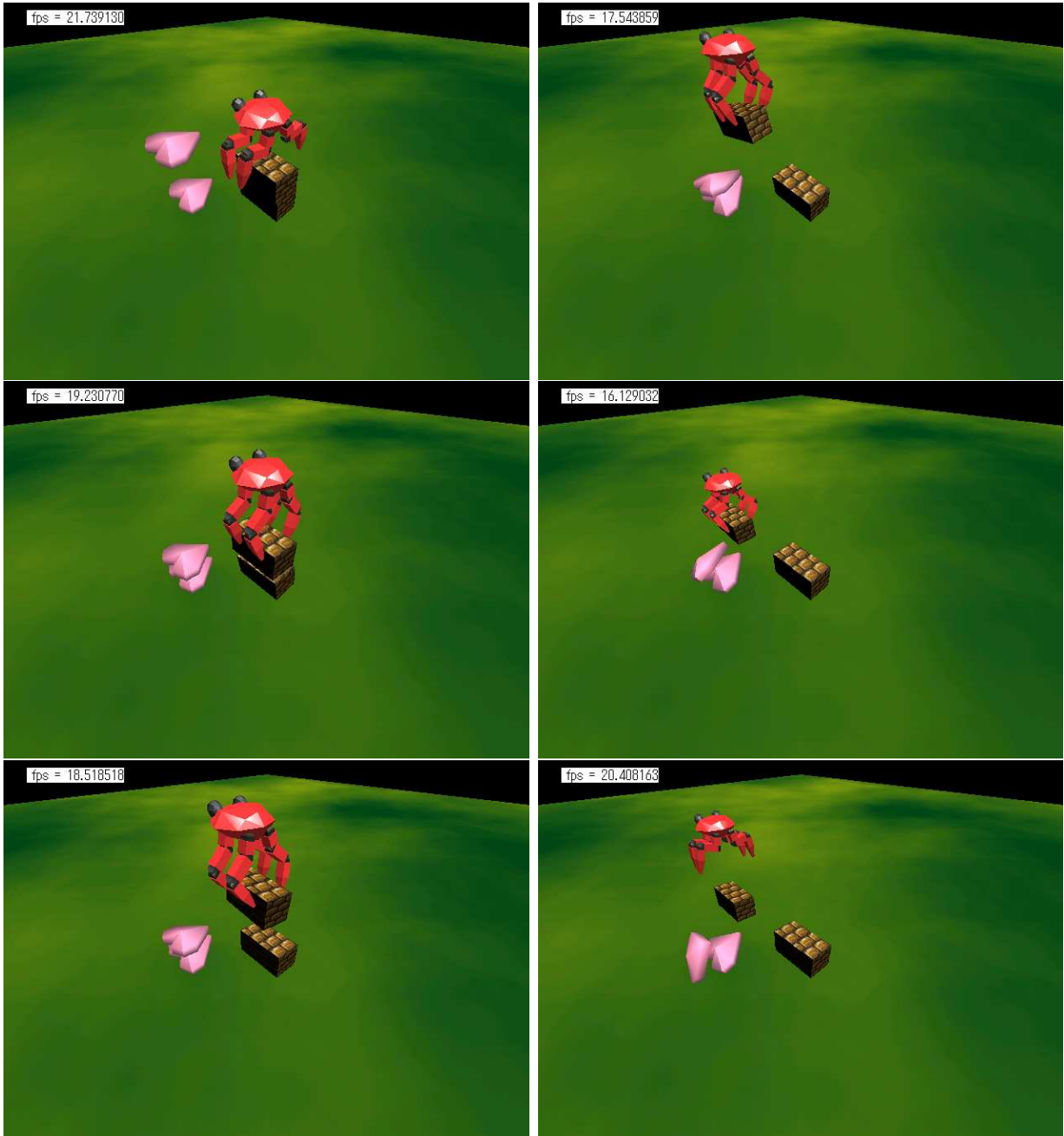


図 8.14. 箱を持ち上げた後離す様子

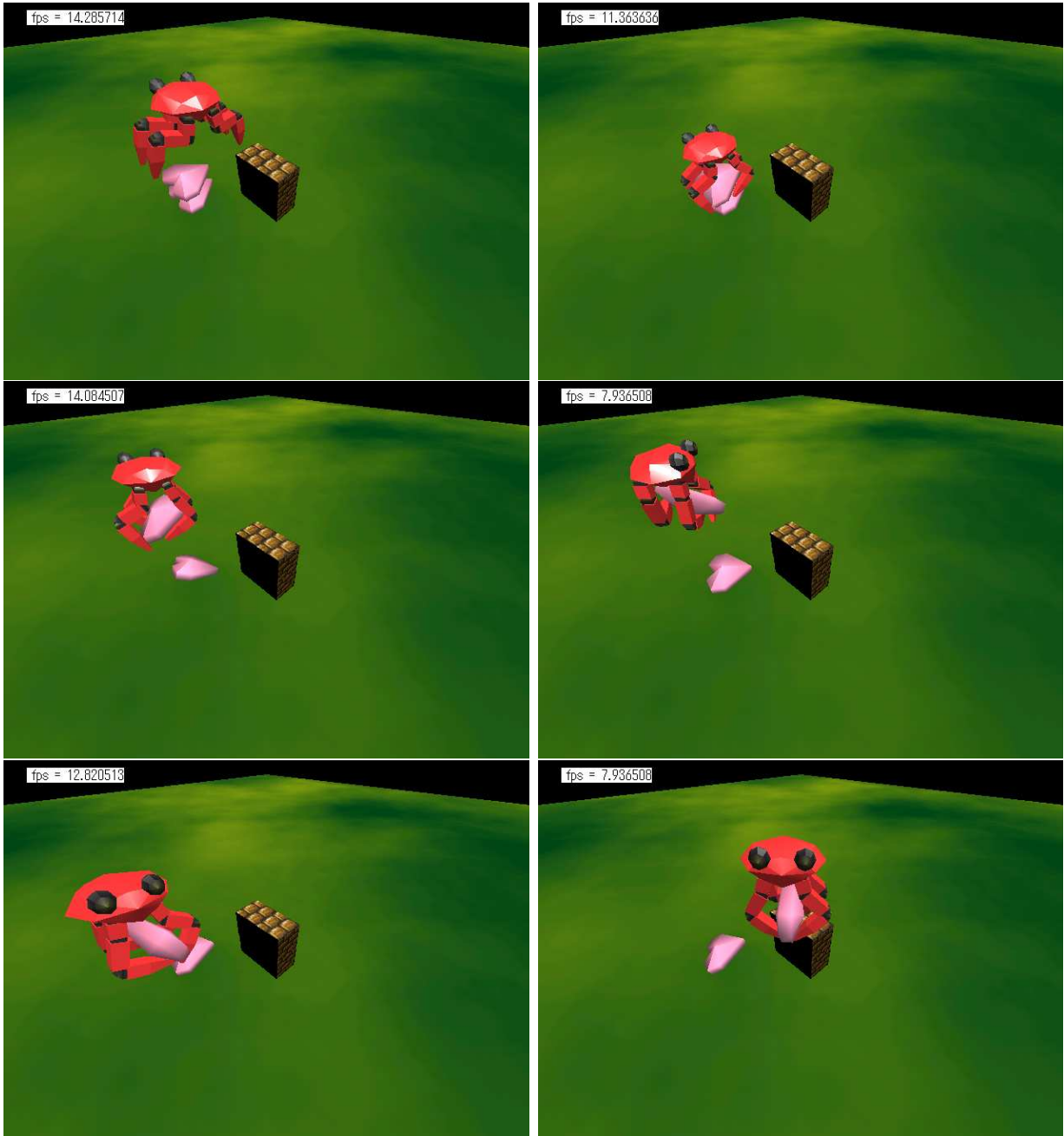


図 8.15. ハート型の物体を持ち上げる様子

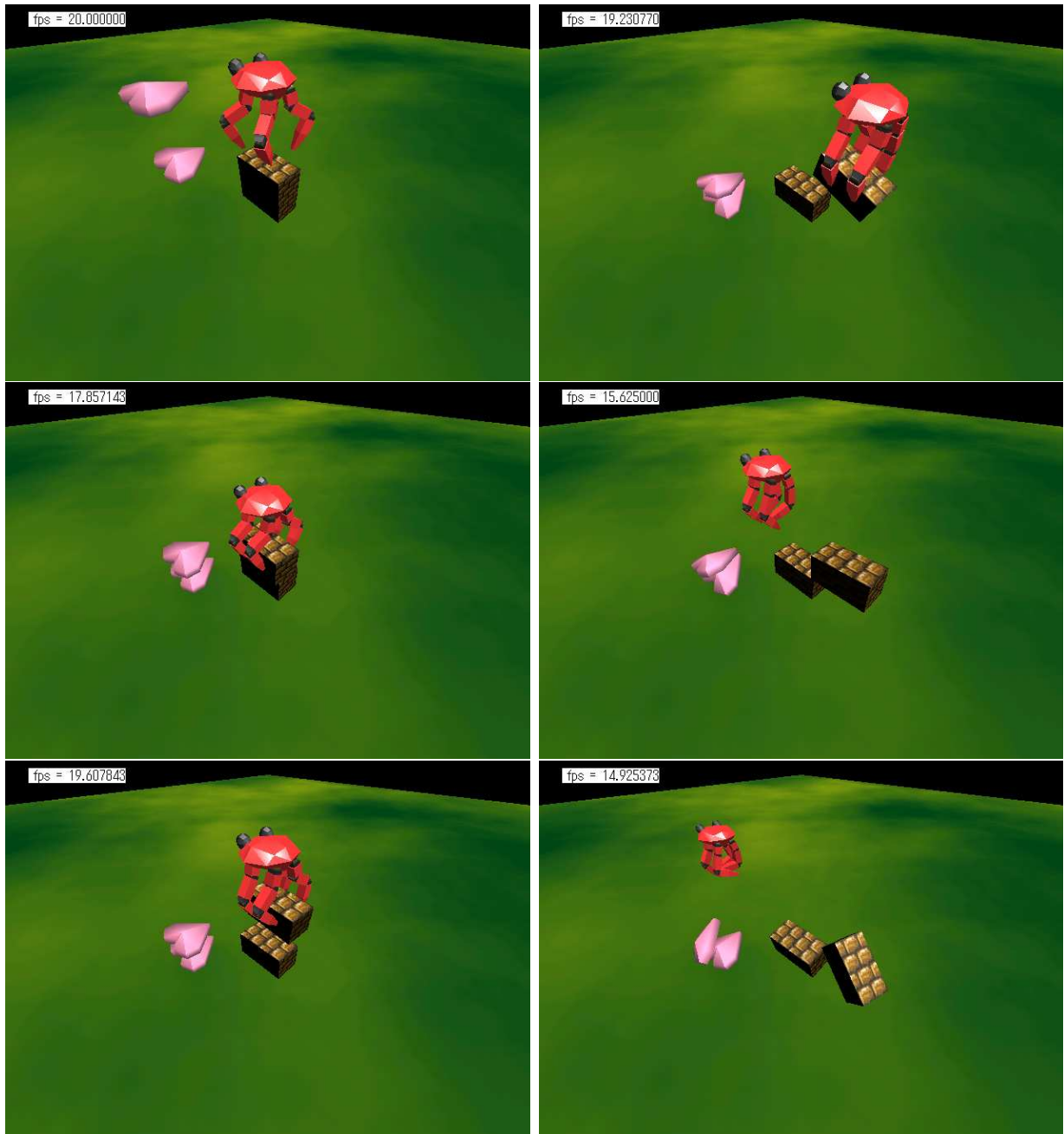


図 8.16. 無理な動きをしたために箱が滑って落ちる様子

図 8.14では箱の両側から挟み込むようなつかみ方をしているため、静止摩擦がないと滑ってしまうはずである。そのような場合においてもしっかりと物体を持ち上げることができる。図 8.5ではハート型の物体はちょうど凹面となっているところに指を引っ掛けて持ち上げられており、形状に忠実に衝突判定がおこなわれていることを確認できる。本シミュレータはPentium3 500MHzのPCでアプリケーションを実行した場合15FPS程度で描画がおこ

なわれ，ユーザの入力に対してインタラクティブに反応する．また，制約ベース法と組み合わせている関係で $1/50s$ の粗いステップ幅でもめり込みを抑えられる．

第 9 章

結論

バーチャルクレーンゲームのプロトタイプを作成し、物体をつかむ動作などをシミュレーションすることで感覚的な範囲で摩擦を含む接触状態と多関節構造のシミュレーションが実現できていること、多面体の衝突判定が機能していることなどを確認した。摩擦が発生している様子など定性的な範囲で表現はできているため、アミューズメント目的などでは本シミュレータの応用は可能と考えられる。

さらにペナルティ法の範囲における摩擦に関して精度などに関して検証をおこなった。動摩擦に関してはステップ幅を細かくすることで理想的な解に近づくことを確認した。静止摩擦に関しては理論値より小さい値で滑り出すのはバネによる表現のためと考えられる。また、2重振り子型のロボットを本シミュレータを使用して解いた場合とラグランジュの方程式から解いた場合では、本シミュレータを使用した場合の解がステップ幅と軸バネの伸縮の制約距離を細かくすることでラグランジュの方程式の解に近づくことを確認した。

今後の課題としては

- 制約ベース法の範囲での精度の検証
- 変形する物体に使用した場合の評価
- 計算コストの評価
- 機械制御や設計に応用した場合の評価

などがあるものと考えている。

現在摩擦についてはペナルティ法の範囲でしか検証をおこなっていない。バネの力が弱い場合や、制約つきバネの伸縮の制約距離が小さい場合など制約ベース法によって摩擦が

発生する場合の計算精度の検証を今後おこなう必要がある。

本研究ではリアルタイムインタラクティブシミュレーションに向くと思われる手法を集めてシミュレータを開発したが、計算コストの評価は十分なものをおこなえていない。しかし、単に剛体を取り扱う場合は多面体を前処理で凸多面体分割をおこない、シミュレーション時は凸多面体で判定をおこなう手法が高速に働くものと考えられる。しかし、変形する物体などに対しては凸多角形分割は前処理化できないため、計算コストが本手法での取り扱いのほうが高速に働く可能性があるものと考えている。今後この点について確認することで新たな知見が得られるのではないかと考えている。

謝辞

この卒業論文の作成にあたり，中央大学大学院 理工学研究科 鈴木 寿教授から研究の方向性などでさまざまな指導をいただき，ここに厚く御礼を申し上げます．本研究を進める上ではX Fellows Conference¹，IRCチャンネル#lab_dodgson²などで情報交換をしたことが非常に重要だったと思います．特に，Kano³氏とは摩擦のことに関して，東京大学大学院工学系研究科精密機械工学専攻博士課程木村・毛利・鈴木研究室の川地克明⁴氏とは階層化バウンディングボリュームなどで，Kaneko⁵氏とは効率のよい衝突判定手法などで，有意義なやり取りをおこなうことができました．X Fellows Conferenceの運営に協力してくださった方々，参加してくださった方々，#lab_dodgsonの皆様には感謝したいと思います．最後に，有意義な研究の環境を提供していただいた鈴木研究室一同，とくにいろいろ協力をいただいた宇津木修一氏，大條 雅彦氏に感謝します．

¹<http://www13.big.or.jp/ akasata/dycoon/xfc/>

²<http://www.dodgson.org/>

³<http://cgi3.tky.3web.ne.jp/ tkano/>

⁴<http://www.cim.pe.u-tokyo.ac.jp/ kawachi/>

⁵<http://homepage1.nifty.com/kaneko/>

参考文献

- [1] C. Lennerz, E. Schomer, T. Warken, “A Framework For Collision Detection And Response,” in 11th European Simulation Symposium, ESS’99, S. 309–314, 1999.
- [2] D. Baraff, “Non-penetrating Rigid Body Simulation,” *State of the Art Reports of EUROGRAPHICS’93, Eurographics Technical Report Series*, 1993.
- [3] D. Baraff, “Rigid Body Simulation,” In SIGGRAPH Course Notes , ACM SIGGRAPH, 2001.
- [4] 川地克明, “機械機構の挙動の直観的な把握を目的とした剛体運動シミュレーション手法,” 東京大学大学院工学系研究科精密機械工学専攻修士論文, 1998.
- [5] 有本 卓, **ロボットの力学と制御** , システム制御情報ライブラリー1 , 朝倉書店, 東京, 1990.
- [6] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes , *Computer Graphics PRINCIPLES AND PRACTICE* , ADDISON-WESLEY PUBLISHING COMPANY, 1990.
- [7] 鈴木 誠道, 矢部 博, 飯田 義久, 中山 隆, 田中 政次, **数値計算法**, オーム社, 東京, 1994.
- [8] 伊理正夫, 藤野 和建, **数値計算の常識**, 共立出版, 東京, 1985.
- [9] Leimkuhler, B. and Skeel, R.D., “Symplectic Numerical Integrators in Constrained Hamiltonian Systems,” *J. Comp. Phys.*, vol. 112, 117–125. S. Reich, Higher-order PRK methods for constrained systems 16, 1994.
- [10] Gino van den Bergen, “Efficient Collision Detection of Complex Deformable Models using AABB Tree,” appeared in *Journal of Graphics Tools*, vol. 112, number 4, pp 1–14, 1999.
- [11] 並川 宏彦, **差分方程式入門**, 現代数学社, 京都, 1978.

- [12] Tan Tiow Seng, Huang Zhiyong, “Collision Detection lecture notes, ”
http://www.comp.nus.edu.sg/~cs3246/CS3246_01/collision.pdf,
<http://www.comp.nus.edu.sg/~cs3246/>.
- [13] Marina A. Epelman, “The steepest descent algorithm for unconstrained optimization,
” http://www-personal.engin.umich.edu/~mepelman/IOE511/class_notes.html, Continuous Optimization Methods, Fall 2001 Lecture notes, section 3.
- [14] 小島政和, “非線形計画法, ” <http://www.is.titech.ac.jp/~kojima/articles/nlp.pdf>,
<http://www.is.titech.ac.jp/~kojima/>.