

剛体シミュレーション世界との 力覚インタラクション

長谷川晶一

2006年9月

目次

概要	7
第1章 序論	9
1.1 バーチャルリアリティ概観	9
1.1.1 計算機とシミュレーション	9
1.1.2 バーチャルリアリティ世界	9
1.1.3 物理シミュレーション	10
1.1.4 力覚インタラクション	11
1.2 研究の目的	11
1.2.1 手による器用な物体操作	11
1.2.2 動作のリアリティ	12
1.2.3 システムと要素の提案	12
1.3 従来研究	12
1.3.1 力覚を伴わない3次元物体操作	12
1.3.2 力覚インタラクションの提案	13
1.3.3 物理シミュレーションによる動作生成	13
1.4 本論文の構成	14
第2章 力覚インタラクションシステムの基本構成	15
2.1 システムの概要	15
2.1.1 力覚インタフェース	15
2.1.2 力覚レンダリング	16
2.1.3 物理シミュレーション	16
2.1.4 映像提示	16
2.2 物理シミュレーション手法と更新周期	16
2.2.1 物理シミュレーションの概要	16
2.2.2 ペナルティ法による拘束力の計算	17
2.2.3 解析法による拘束力の計算	17
2.3 2つのシステム構成の提案	18
2.3.1 ペナルティ法によるシステム	19
2.3.2 解析法によるシステム	19
第3章 力覚インタフェース	21
3.1 力覚インタフェースの機構と制御の概要	21

3.1.1	力覚インタフェースの機構	21
3.1.2	力覚インタフェースの制御	21
3.2	高解像度力覚インタフェースの提案	23
3.2.1	解像度の高い力覚提示のための機構	24
3.2.2	応答の良い駆動方法	27
3.2.3	高速制御のためのプロセッサ	27
3.3	系駆動力覚インタフェースの姿勢・力覚計算の概要	28
3.3.1	位置計測	28
3.3.2	力覚提示	29
3.4	系駆動力覚インタフェースの姿勢計算	30
3.4.1	ヤコビ行列の利用	30
3.4.2	繰り返し計算による並列機構の姿勢計算	31
3.4.3	系駆動型力覚インタフェースの姿勢計算	31
3.4.4	SPIDAR の姿勢計算	33
3.4.5	提案手法の特徴	33
3.4.6	計算速度の評価	34
3.4.7	精度・繰り返し回数の評価	34
3.5	系駆動力覚インタフェースの力覚計算	35
3.5.1	系駆動型力覚インタフェースの定式化	35
3.5.2	完全提示域と部分提示域	36
3.5.3	領域境界での張力の不連続性	37
3.5.4	二次計画法による力覚計算	38
3.5.5	SPIDAR の力覚計算	39
3.5.6	精度と安定性	40
3.5.7	計算時間の評価	40
3.5.8	提示部の位置と張力の関係	40
3.5.9	正確性と安定性の評価	41
3.5.10	提示形状の評価	41
3.6	汎用制御ライブラリの構築	42
3.6.1	機構依存部分の分離	42
3.6.2	新規機構への対応	43
3.6.3	計算システムの位置付け	43
3.6.4	汎用性の評価	44
3.7	力覚提示性能の評価	44
3.7.1	剛性の高い面の提示	44
3.7.2	精細な形状の提示	45
3.8	この章のまとめ	45
第4章	力覚レンダリング	55
4.1	物体形状の提示	55
4.1.1	god object 法の詳細	55

4.1.2	バーチャルカップリング法	57
4.2	多指力覚インタフェースによる安定な把持の実現	57
4.3	実世界での把持	58
4.3.1	把持のモデル	59
4.3.2	把持力の計測	59
4.3.3	計測方法	60
4.3.4	計測結果	60
4.4	力覚インタフェースによる把持提示の問題点	61
4.5	合力分配方式	62
4.5.1	把持力の計算	62
4.5.2	合力の計算	63
4.5.3	合力の分配	63
4.6	多指 6 自由度物体の把持への拡張	64
4.6.1	把持力の計算	64
4.6.2	合力の計算	65
4.6.3	合力の分配	65
4.7	合力分配方式の評価実験	66
4.7.1	実験条件	66
4.7.2	実験結果	66
4.8	考察	68
4.9	この章の結論	68
4.10	今後の課題	68
4.11	この章のまとめ	68
第 5 章	ペナルティ法による力覚インタラクション	73
5.1	ペナルティ法による剛体シミュレーションの概要	73
5.2	提案	73
5.2.1	接触体積に基づく抗力計算	74
5.2.2	接触体積に基づく摩擦力計算	74
5.3	提案手法による剛体運動シミュレータ	75
5.3.1	接触判定	76
5.3.2	接触の法線	76
5.3.3	接触部分の形状の解析	77
5.3.4	接触力計算の準備	77
5.3.5	抗力の計算	80
5.3.6	摩擦力の計算	81
5.4	評価	85
5.4.1	実験に用いるシミュレータ	85
5.4.2	実験 1：計算時間の比較	85
5.4.3	実験 2：安定性の比較	86
5.4.4	実験 3：摩擦モデルの精度の比較	87

5.4.5	シミュレーション例 1:直接操作環境	88
5.4.6	シミュレーション例 2:分布摩擦力の効果	88
5.4.7	シミュレーション例 3:両手直接操作環境	89
5.5	この章のまとめ	90
第 6 章	解析法による力覚インタラクション	95
6.1	静的世界のための補間力覚レンダリング	95
6.1.1	動的世界での問題	95
6.1.2	通信情報の不足	96
6.2	動的世界への対応	97
6.2.1	問題の原因	97
6.2.2	物体の動きと力	97
6.2.3	力積を用いた通信	98
6.2.4	動的世界での効果	99
6.3	力覚インタラクションシステムの構築	100
6.3.1	形状特徴の求め方	100
6.3.2	形状特徴の補間	101
6.3.3	力覚レンダリング	102
6.3.4	力の積分	102
6.3.5	VR 世界の更新	103
6.4	評価	103
6.4.1	シミュレーション	103
6.4.2	実験	103
6.4.3	多物体とのインタラクション	104
6.5	考察	105
6.5.1	力覚提示と画像提示の違い	105
6.5.2	画像レンダリングプロセスによる時間補間	106
6.5.3	摩擦の表現	106
6.5.4	処理分割の階層性	106
6.5.5	物体の変形	107
6.6	この章のまとめ	107
6.6.1	リアルタイム性の確保	107
6.6.2	ペナルティ法と解析法の比較	107
第 7 章	結論	113
7.1	本論文の結論	113
7.2	今後の展望	114
	参考文献	119

概要

本論文では、バーチャルリアリティ(以下、VR)世界の物体モデルが実世界の物体と同様に動き、物体モデルを実世界と同様の感覚で操作できるようなVRシステム、力覚インタラクションシステムを提案する。

第1章「序論」では、VR研究を概観し、VR世界での物体モデルの動作生成手法と物体モデルを操作する手法について研究の流れを次のように概説する。近年ではヒューマンインタフェースの発展より、VR世界の物体モデルをより直感的・直接的に操作できるようになった。また、物理法則をシミュレーションして物体モデルの動作を生成することで、直接的な操作による多様な入力に応じた物体モデルの動作が生成できるようになった。

そして、次のように手による操作について考察し、具体的な研究目的を導く。人は手で物体を操作するとき、対象物体の運動を完全に拘束するわけではなく、対象物体が他の物体から受ける力と手で加える力をバランスさせて目的の運動を実現している。例えば、机の上の物体を移動させるときには、物体をつかみ持ち上げたあと目標の位置に移動して放すのではなく、物体を指で押したり、軽くつまんで引きずったりして移動させることが多い。このような操作をVR世界の物体モデルに対して行うためには、操作時に手に加わる反力の感覚(=力覚)が人に提示される必要がある。また、人が手で操作する物体は多くの場合変形を無視できるので、変形しない物体のモデルである剛体をシミュレーションすることで、実世界と同様の物体動作が生成できる。そこで、剛体シミュレーションによって物体動作を生成し、力覚インタフェースによって力覚提示と操作入力を行うことで、剛体シミュレーション世界との力覚インタラクションを実現し、実世界と同様の感覚で物体操作が可能な力覚インタラクションシステムを提案することを具体的な研究目的とする。

第2章「力覚インタラクションシステムの基本構成」では、提案する力覚インタラクションシステムが、ユーザに映像を提示するコンピュータグラフィックス、力を提示する力覚インタフェース、提示力を計算する力覚レンダリング、物体動作を生成する物理シミュレーションから構成されることを示す。そして、物理シミュレーションに必要な接触力の解法にペナルティ法と解析法があり、力覚インタラクションの精度を重視する場合は高速更新に向けたペナルティ法、計算効率を重視する場合は低速更新に向けた解析法が適することを明らかにし、2つのシステム構成を提案する。

第3章「力覚インタフェース」では、力覚インタフェースの機構と制御方法を検討し、糸を用いたパラレル機構によるインピーダンス提示型のインタフェースが高精細な力覚提示に適することを明らかにし、機構と制御方法を提案する。また、正確で安定な力覚提示には力覚インタフェースを高速に制御する必要があることを示し、専用のプロセッサを用いて高速制御することで、精細な形状や多様な材質感を提示することを提案する。

第4章「力覚レンダリング」では、まず、物体形状の提示手法について説明し、次に実

物体を手で操作する場合の基本操作である把持の安定な提示方法を提案する．具体的には，人の手による把持では多指が物体に触れるため，提示力の分配に冗長性があることを指摘し，人の把持特性を考慮して分配することで，安定な把持提示を実現する．

第5章「ペナルティ法による力覚インタラクション」では，力覚制御と物理シミュレーションの両方を高速に行う力覚インタラクションシステムの構成と，接触部分の形状を考慮した正確で安定な接触力計算を提案する．提案接触力計算を用いて力覚レンダリングを行い，第3章で提案した力覚インタフェースを用いた力覚インタラクションシステムを構築する．また，評価実験によって正確で安定な接触力計算と高速更新の維持ができることが確かめる．

第6章「解析法による力覚インタラクション」では，解析法による構成は物理シミュレーションの更新を低速にすることで計算負荷を抑えることができ，大規模なVR世界に適していることを示し，シミュレータの情報を力覚レンダリングで補間して自然な力覚を提示すること，ユーザが加えた力積に基づいて物体モデルの運動を正確に更新することを提案する．そして，第3章で提案した力覚インタフェースを用いて力覚インタラクションシステムを構築し，自然な力覚提示と物体モデルの運動生成ができることを実験により確かめる．

第7章「結論」では，本論文が，剛体シミュレーション世界との力覚インタラクションを実現する力覚インタラクションシステムの提案により，物体モデルが実世界の物体と同様に動き，物体モデルを実世界と同様の感覚で操作できるようなVR世界を実現するものであると述べる．具体的には，力覚インタラクションを実現するために，力覚インタラクションシステム全体を高速に更新して正確で安定な力覚インタラクションを実現する構成と，シミュレーションを低速に更新することで計算負荷を抑え大規模VR世界との力覚インタラクションを実現する構成の，2つの構成を提案・構築し，評価実験を行ってそれぞれを実現したことを確かめた．

以上のように本論文は，実世界と同様にVR世界の物体モデルを操作できる環境を提案，構築し，人とVR世界の自然なインタラクションを実現するものであり，VR研究の発展に貢献すると考えられる．

第1章 序論

1.1 バーチャルリアリティ概観

1.1.1 計算機とシミュレーション

今日では、コンピュータは、処理速度、ソフトウェア、インタフェースの発展とともに広がり、さまざまな用途に利用されている。その中でも、シミュレーションは、コンピュータの創成期から主要な用途だった。ニュートンの運動方程式以来、さまざまな現象が微分方程式によってモデル化され、その解を求めて現象をシミュレーションすることが行われてきた。コンピュータはその創成からこういったシミュレーションに用いられ、膨大な手計算を不要にしてきた。

シミュレーションをリアルタイムに行うことができるようになると、人がシミュレーションに組み込まれたシステムが構築できるようになった。1951年には、人の操作に反応するフライトシミュレータを構築するために、ブラウン管を持つ最初のコンピュータ Whirlwind が作られた [1]。このときのシミュレーション対象は1機の航空機の挙動だけであり、人への情報提示は計器の表示だけだった。しかし計算機モデルとのリアルタイムインタラクションを実現したという意味ではバーチャルリアリティ (VR) への第一歩だったといえる。

1.1.2 バーチャルリアリティ世界

3次元コンピュータグラフィックスがリアルタイムに提示されると、人は映像の先に世界が広がっているかのように感じる。そして、シミュレーション対象の計算機モデルを3次元コンピュータグラフィックスで提示すると、計算機モデルがその映像の世界の中にあたかも存在するかのような感覚を人に提示することができる。Ultimate Display [2] を開発した Sutherland は計算機が作り出す映像の世界のことを「It is a looking glass into a mathematical wonderland」と表現している。Sutherland が mathematical wonderland と呼んだ計算機が作り出す世界は、現在ではVR世界と呼ばれ、目的に応じてさまざまな物体のモデルをシミュレーションすることで、設計・評価・教育・訓練・エンタテインメントなどさまざまな目的に利用されている。

映像提示や視点移動入力に加えて、VR世界やその中の物体モデルとインタラクションできることは、ユーザがVR世界でとれる行動の種類を増やし、VR世界の可能性を広げる。物体モデルとのインタラクションを実現するためには、物体モデルの動きを作り出すシミュレーションと、ユーザの操作を入力する入力インタフェースが必要となる。

1.1.3 物理シミュレーション

1.1.1 節にも述べたように，計算機は開発されて以来，シミュレーションに使われ続けており，物理現象の予測や解明に貢献してきた．例えば，1946 年に開発された ENIAC は，大砲の弾道計算を目的として開発されており，2002 年に開発された地球シミュレータは，地球全体の気候・気象の予測などに役立っている．

物理法則を計算機でシミュレーションすることによって，日常の空間を高いリアリティで再現することは，コンピュータアニメーションの作成を目的に行われた．たとえば，Baraff [65] は，剛体の物理シミュレーションを用いて，高いリアリティで物体の動きをシミュレーションしている．計算機の処理速度が低かったこともあり，リアリティの高い物理シミュレーションは，映像制作に用いられた．映像制作では，事前に時間をかけて計算とレンダリングを行うことで動画像を作成するため，時間をかけて精度の高いシミュレーションを行うことができた．

一方 VR では，ユーザの入力がシミュレーションに影響しその結果が提示されなければならない．シミュレーションをその場でリアルタイムに行う必要がある．入力から提示までの反応時間が人間の感覚に対して長くなると，現実感が喪失したり操作が困難になるなどの問題が起こる．そのため汎用的な物理シミュレータを VR への応用するには時間がかかった．

初期の VR システム [3, 4, 5] では，システム製作者が一つ一つ物体モデルの動作をプログラムしていた．そのため，ユーザーのインタラクションはスイッチの ON・OFF やドアの開け閉めなどに限られており，物体モデルの動きも限られていた．そのため，ユーザは VR システム設計者があらかじめ用意したシナリオに沿った行動しか取れなかった．

その後，一部の物体モデルをなんらかの法則に基づいてシミュレーションして動かすことで，限られた状況ながらユーザの自由な行動に反応できる VR システムが構築されるようになった．例えば，Ruspini らのシステム [6] では，風車の羽根車やジェットコースターの車両だけが 1 自由度の運動をする．このような法則に基づくシミュレーションでは，物体モデルの動作を一つ一つ記述する場合に比べると少ない記述で多様な動きを表現できる．しかし，一部の物体モデルを限られた状況でのみ有効な法則に基づいてシミュレーションしているため，想定外の物体モデルを操作された場合や，法則が有効でない操作をされた場合は対応できない．例えば，Rasupini らのシステムでは，風車の羽根車をはずしたり，ジェットコースターから車両をはずして走らせたりすることはできない．

物理法則は，実物体の運動を支配する法則であり，広い範囲で有効な法則である．物理法則をシミュレーションして物体モデルの運動を生成すると，実物体に近い運動が生成できるほか，さまざまな操作や状況で不整合のない動作を生成できる．Virtual Basket [7] では，剛体の運動の法則にしたがってボールをシミュレーションしており，ユーザがボールに対してどのような操作を行っても実物体と同様に反応する．

VR 世界内の物体モデルすべてを剛体の運動の法則に基づいてシミュレーションすると，すべての物体モデルが実物体と同様に動作するため，実物体に対して可能な操作の多くが VR 世界でも実現できるようになり，より自由なインタラクションが実現できる．例えば，Fröhlich ら [8] は，物理シミュレーションされた複数の物体モデルを用いて，組み立て作業環境を構築している．

1.1.4 力覚インタラクション

Polhemus [9], Sonic Pen [10] のような 3 次元位置入力インタフェースは、物体モデルの位置・姿勢を直接入力することができるが、物体に触れたときに手にかかる力の感覚である力覚は再現できない。

人は物体を動かしたり操作したりするとき、手に働く力 (= 力覚) を感じ、手と物体の状態を把握しながら手を巧みに制御する。力覚が器用な操作に必要なだということは、テレオペレーションの研究 [11, 12] を通じて知られていた。物体モデルの操作のための 3 次元入力インタフェースが提案されると、やがてそれに力覚提示機能が付加された [13]。その後、さまざまな力覚インタフェースが提案され、卓上で使用するインタフェースが研究者の間に普及した [14]。

力覚インタフェースを用いると、ユーザは物体モデルに与えた力を感じながら物体モデルを操作できる。物理シミュレーションは、物体モデルに加わる力から運動を計算する。力覚インタフェースと物理シミュレーションを組み合わせると、物体モデルを実物体の場合と同様の力覚を感じながら、同様に操作できるようになる。物体モデルの直感的な操作を実現するため、物理シミュレータと力覚インタフェースを用いたインタラクションの研究が近年盛んに行われるようになってきた [17, 18]。

1.2 研究の目的

本研究では、物体モデルが実世界と同様の動作をし、実世界と同様に物体モデルを操作できるような力覚インタラクションの実現を目的とする。

1.2.1 手による器用な物体操作

人は手で物体を操作するとき、対象物体の運動を完全に拘束するわけではなく、対象物体が他の物体から受ける力と手で加える力をバランスさせて目的の運動を実現している。例えば、机の上の物体を移動させるときには、物体をつかみ持ち上げたあと目標の位置に移動して放すのではなく、物体を指で押したり、軽くつまんで引きずったりして移動させることが多い。このような操作は効率が良いためロボットの制御法としても研究されている [20]。

このような器用な操作を VR 世界の物体モデルに対して行うためには、物体モデルの動作が実世界と同様で人の推定に合致していることと、動作の推定に必要な情報が人に伝わる必要がある。1.2.2 節で述べたように、物体の動作については、物理シミュレーションによって実世界と同様の動作を生成できる。では、必要な情報を人に伝えるにはどうすればよいだろうか。

人は物体を視覚で捕らえて物体の運動を推定するとともに、手の感覚によって手に働く力 (= 力覚) を感じて手が物体に加える力を制御している。[11, 12]。そのため、力覚が提示されない場合、操作の精度が非常に悪くなることが知られている [19]。また、物体操作時に手に働く力を再現することで、実世界と同様の自然な操作感の提示が期待できる。そ

ここで、本研究では力覚インタフェースを用いて物体が手に加える力覚を再現することにより、器用で巧みな操作を自然な操作感で行えるような環境を構築する。

1.2.2 動作のリアリティ

これまで、VRの研究は、視野角、解像度、音像定位、提示物体の硬さなど、没入感や提示感覚のリアリティを追求し、訓練用シミュレーション、製品評価などへ応用されてきた。しかし、従来のVRシステムでは、ユーザが行える作業や行動の種類が限定されており、決まったシナリオや特定の作業の評価のためのシステムがほとんどだった[3]。これは、物体モデルの動作をひとつひとつキーフレームアニメーションなどで用意したり、物体モデルごとに限られた状況でのみ有効な法則を適用していたためである。

VR世界でユーザが自由に行動でき、任意の物体に対して自由に作業できるような環境を実現すれば、VRシステムの応用が広がると期待できる。そのような環境を構築するためには、多様な入力と状況に応じた物体の動作を生成する必要がある。実世界では、物体は運動の法則に従って動く。運動の法則に従って物体の動きをシミュレーションすることによって、VR世界の物体にも、現実と同様にさまざまな状況・入力に対応した動きをさせることができる。実世界にはさまざまな物体が存在し、その物理的特性もさまざまだが、手による操作の対象物体の運動は、物体の変形などを無視し、剛体の運動と考えてシミュレーションすることで再現できる物が多い。そこで、本研究ではVR世界内のすべての物体モデルの運動を剛体運動の物理シミュレーションによって作り出す。

1.2.3 システムと要素の提案

本研究では、剛体運動の物理シミュレーションと力覚インタフェースによって、さまざまな状況や入力に応じた自然な物体運動の生成と操作時の力覚提示が実現でき、現実に近い自由度と操作性を持つVRシステムを実現する。このようなVRシステムは、剛体運動の物理シミュレーションとの力覚インタラクションを実現するので、力覚インタラクションシステムと呼ぶ。

力覚インタラクションシステムは、1.2.2節で述べた動作のリアリティを実現する物理シミュレーション、1.2.1節で述べた手による器用な操作を実現する力覚インタフェース、力覚インタフェースが提示する力を計算する力覚レンダリングからなる。これらの要素技術と力覚インタラクションシステムを提案することが、本研究の具体的な目的である。

1.3 従来研究

1.3.1 力覚を伴わない3次元物体操作

VR世界の物体モデルの操作は、力覚インタフェースを用いない場合についても研究されている。Fröhlichら[8]は、物理シミュレーションされた物体モデルをポヒマスセンサ[9]を用いて直接操作するため、物体モデルとポヒマスセンサをバネダンパモデルによって

結合させ、力をバネの伸びとして映像提示することを提案している．Fröhlich らのシステムは力覚を提示することでより操作性が向上すると考えられる．

清川らは、VLEGO[59]において、物理法則以外の法則を適用し、特定の操作をしやすくすることを提案している．このシステムは、操作を助けるような法則を用いることで、力覚を用いずに直感的な操作を実現している．しかし、操作を助ける法則がユーザの行動に実世界とは異なる制約を加えるため、ユーザは、期待した操作ができないなどの不自由を感じることもある．

1.3.2 力覚インタラクションの提案

90年代前半には、さまざまなモデルとの力覚インタラクションが提案されている．平面[50, 51, 14, 52]や、物体の形状やテクスチャ[53]だけでなく、ボールのバウンド、倒立振り子[53]、ピアノの鍵盤[54]のように動力学モデルを持つ物体とのインタラクションも行われている．これらのシステムでは、モデルと力覚インタフェースの制御は分離されておらず、モデルを変えるたびにプログラミングが必要だった．

1.3.3 物理シミュレーションによる動作生成

90年代後半には、より柔軟で複雑なバーチャル世界とのインタラクションを実現する方法が提案された．Chang と Colgate [55] は、2次元平面内の物体運動シミュレータを構築し、バーチャルカップリングを用いて、2自由度力覚インタフェースによる直接操作を実現した．Mark ら [56] は、グラフィックシステムに用いる大規模なポリゴンモデルとの力覚インタラクションを実現した．

また、3次元物体モデルに対して、狭い隙間に部品が通せるか確認するアプリケーション[57]も提案されている．このような作業では、対象物体の自由な操作と、物体の回転を妨げる力を提示するためのトルク提示が必要となる．これを実現する力覚インタフェースは、6自由度力覚インタフェースと呼ばれ、Hollis らの Magnetic Levitation Haptic Device[17]、Sensable 社の PHANTOM Premium 1.5/6DOF、佐藤らの Spidar-G [21] などが提案されている．また、Hollerbach らは、4節リンクの1節をつかんで動かしたときに働く力を6自由度力覚インタフェースを用いて提示している [26]．

6自由度力覚インタフェースでは、ポインタも複雑な形状となるため、接触力の計算も複雑になる．また、近年では、より実用的なシステムを構築するために、多数の3次元物体モデルを扱うことが要求され、リアルタイム動力学シミュレータが注目されるようになった．これに伴い接触力の提示も動力学シミュレータの機能を利用した手法が用いられるようになった．Berkelman ら [17] は、動力学シミュレータと6自由度力覚インタフェースをバーチャルカップリング[50]と呼ばれるバネダンパモデルで結合し、接触力を提示する方法ことを提案している．Berkelman らのシステムは、複数の物体モデルを任意形状のポインタを用いて操作できる上、剛体の動力学シミュレーションを行うため、物体モデルが現実の物体と同じように運動する．Hollis らは、Magnetic Levitation Haptic Device と物理シミュレータ [60] をバーチャルカップリングや中間表現を用いて接続し、シャベル状

のポインタで複数の剛体を操作することを実現した。しかし、これらのシステムでは、シミュレータの更新頻度が低いいため、やわらかいバネダンパモデルを介した鈍い力覚になるなど、提示される力覚に不自然さが残る。このため、自然な操作感が再現できず、物体を器用に操作することは難しい。

本研究では、さまざまな状況や入力に応じた物体運動の生成と、品質の高い力覚提示を実現し、現実に近い自由度と操作性を持つVRシステムを提案する。

1.4 本論文の構成

本論文では、7章から構成され、第2章以降では以下のように議論を進める。

2章「力覚インタラクションシステムの基本構成」では、剛体シミュレーション世界との力覚インタラクションを実現するシステムの構成を示し、システム構成に、物理シミュレーション手法の特徴によって、自然な操作感の追及と、大規模なVR世界を取り扱うための効率性の追求の2つの方向性があることを明らかにする。

3章「力覚インタフェース」では、力覚インタフェースの機構と制御方法を検討し、系を用いたパラレル機構によるインピーダンス提示型のインタフェースが高精細な力覚提示に適していることを示し、機構と制御方法を提案する。また、精細な形状や多様な材質感を提示するため、専用のプロセッサを用いて力覚インタフェースの制御を非常に高速に更新することを提案する。

4章「力覚レンダリング」では、まず、物体の形状を提示する力覚レンダリングについて説明する。次に実物体を手で操作する場合の基本的な操作である把持を安定に提示する方法を提案する。把持では多指が物体に触れるため、提示力の分配に冗長性があることを指摘し、人の把持特性を考慮して分配することで、安定な把持提示を実現する。

5章「ペナルティ法による力覚インタラクション環境」では、ペナルティ法による力覚インタフェースの制御と剛体シミュレーションの両方を高速に行うシステムの構成法を示すと共に、接触部分の形状を考慮することで正確で安定な接触力計算を行うことを提案する。この接触力計算を用いて力覚レンダリングを行い、3章で提案した力覚インタフェースを用いた力覚インタラクション環境を構築する。また、評価実験によって、正確で安定な接触力計算ができること、高速な更新を維持できることが確かめる。

6章「解析法による力覚インタラクション」では、解析法が、更新を低速にすることで計算負荷を抑えることができ、大規模なVR世界に適していることを示し、更新が低速なシミュレータの情報を力覚レンダリングで補間することで自然な力覚を提示すること、ユーザの操作をシミュレーションの更新周期にあった形式に変換してからシミュレータに渡すことでユーザの操作を物体モデルの運動に正確に反映させることを提案する。そして、3章で提案した力覚インタフェースを用いて力覚インタラクション環境を構築し、自然な力覚提示と物体モデルの運動生成ができることを実験により確かめる。

7章「結論」では、これまでの議論をまとめ、2章で提案した二つ方向性を比較考察する。また、力覚インタラクションの今後の発展について考察する。

第2章 力覚インタラクションシステムの基本構成

本章では，力覚インタラクションシステムの構成要素と基本構成について述べる．

2.1 システムの概要

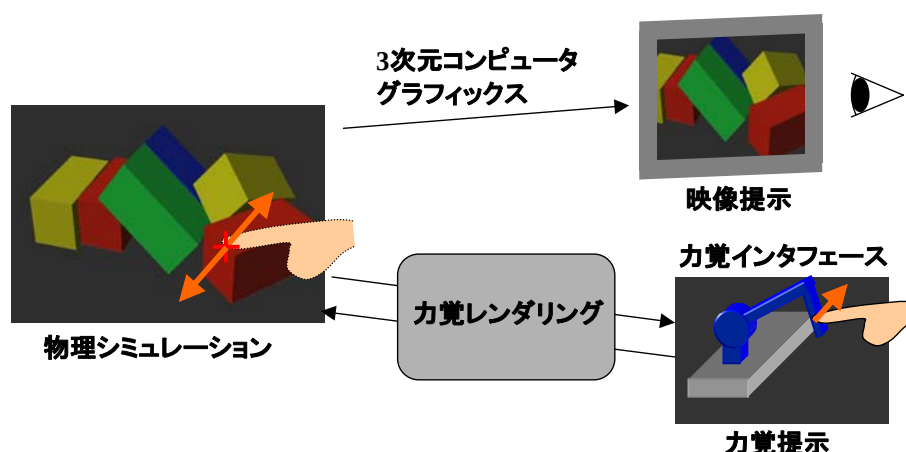


図 2.1: 力覚インタラクションシステムの構成要素

力覚インタラクションシステムは，ユーザに力を提示する力覚インタフェース，VR 世界の状態とユーザの手の位置・速度から手に提示する力を計算する力覚レンダリング，物体モデルの運動を生成する物理シミュレーションから構成される．また，ユーザが VR 世界を把握できるように 3 次元コンピュータグラフィックスによる映像も提示する．図 2.1 に，力覚インタラクションシステムの基本構成を示す．

2.1.1 力覚インタフェース

力覚インタフェースは，ユーザの手の位置・速度などを計測し，それに応じた力覚を提示するためにモータなどの出力を制御する．この処理を高速に繰り返すことで，力覚インタフェースによるリアルタイムに状況に応じた力覚の提示が実現する．制御と出力の更新が高速なほど，提示可能な物体の硬さや重さの範囲が広がることが知られている．

2.1.2 力覚レンダリング

力覚レンダリングは、物体モデルと力覚ポインタ (ユーザの手のバーチャル世界での表現) の間に働く力を求め、力覚インタフェースが提示すべき力を求める。物体モデルはシミュレーション手法に合った頻度で更新されているが、提示力は力覚インタフェース制御にあわせて出力する必要がある。

2.1.3 物理シミュレーション

物理シミュレーションは、実世界の物体の運動の法則である動力学を用いて、物体モデルの運動を生成する。バーチャル空間内の物体モデル全体をシミュレーションする必要があるため、多数の物体モデルを扱うためには、少ない計算量でシミュレーションを行う必要がある。

2.1.4 映像提示

力覚を用いることで、手と物体の位置関係や把持の状態などをユーザに提示することができるが、物体の位置姿勢など VR 世界全体の状態を伝えることはできない。実世界で人は、視覚を用いて物体の位置姿勢など空間の状態を把握している。そこで、ユーザに VR 世界の状態を提示するために、3次元コンピュータグラフィックスによる映像提示を行う。

2.2 物理シミュレーション手法と更新周期

本節では、物理シミュレータの概要を説明し、物理シミュレーション手法と更新周期の関係を明らかにする。剛体運動の物理シミュレーションは拘束力の計算方法で特徴付けられ、拘束力の計算法が解析法とペナルティ法に大別できる。そこで、解析法とペナルティ法を概説し、各手法に適した更新周期について述べる。

2.2.1 物理シミュレーションの概要

本研究は、手による自然な物体操作を目的とするので、物体モデルの運動は剛体の運動と考えてよい場合が多い。そこで、剛体運動の物理シミュレーションについて考える。

剛体の運動は、剛体の運動方程式

$$m\dot{\mathbf{v}} = \mathbf{f}, \mathbf{I}\dot{\mathbf{w}} = \mathbf{p} \times \mathbf{f}$$

であらわされるので、式 (2.2.1) を差分化した運動量・角運動量の更新式

$$\begin{aligned} m\mathbf{v}(t + \Delta t) &= m\mathbf{v}(t) + \mathbf{f}\Delta t \\ \mathbf{I}(t + \Delta t)\mathbf{w}(t + \Delta t) &= \mathbf{I}(t)\mathbf{w}(t) + \mathbf{p} \times \mathbf{f}\Delta t \end{aligned} \quad (2.1)$$

を繰り返し計算することで，シミュレーションが行われる．式 (2.1) の右辺なかで，物体に働く力 f だけは，自明ではなく，物体の接触状態などによって値が変化する．物体にはさまざまな力が加わるが，重力，バネダンパによる力のように物体の現在の状態から計算できる力と，抗力や摩擦力のようにある条件を満たすように働く拘束力に分けることができる．例えば，物体が接触したときに働く抗力と摩擦力は，「剛体は互いに侵入しない．」「静止摩擦力の最大値と動摩擦力は抗力に比例する．」といった条件を満たすように働く拘束力である．

シミュレータは物体モデルの現在の状態を保持しているので，前者はすぐに求まる．このため，物理シミュレータの主な仕事は後者の拘束力の計算になる．拘束力の計算法には大きく分けると，ペナルティ法と解析法の 2 つの方法が提案されている．

2.2.2 ペナルティ法による拘束力の計算

ペナルティ法は，拘束力を直接計算せずに，拘束条件を違反すると違反の量（ペナルティ）に応じた力を求め，これを拘束力の代わりに物体に加える方法である．例えば，物体間に働く抗力を求めるためには，2 物体の接触点の間にバネダンパモデルを考え，バネダンパモデルが発生する力を抗力として加える (図 2.2) ．

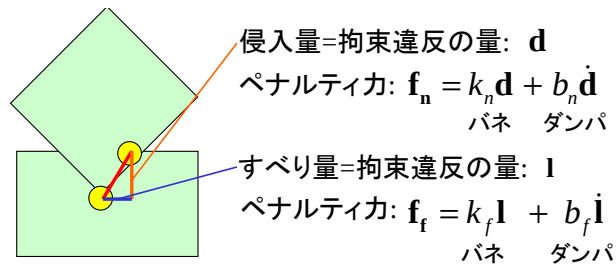


図 2.2: ペナルティ力の例

ペナルティ法では，拘束力はバネダンパモデルの状態からすぐに求まるため，計算量は $O(n)$ と少ない．また，運動方程式と無関係に拘束力が求まるため，力覚ポイントのような質量や慣性モーメントなどの動力学特性が不明なものと物体モデルの間に働く拘束力も計算できる．しかし，ペナルティ法で求めた拘束力は，拘束条件を満たす力ではなく，シミュレーションを繰り返すうちに条件違反が解消する（と思われる）力であり，ペナルティ量やペナルティ力が収束する保証はない．バネダンパモデルが収束する範囲は，シミュレーションの更新周期 Δt に反比例して狭くなることが知られている．硬いバネダンパモデルを用いて大きなペナルティ力を加えれば，拘束違反量を小さく抑えることができるが，収束させるために，更新周期 Δt を十分小さくしなければならなくなる．

2.2.3 解析法による拘束力の計算

解析法は，拘束力の条件を図 2.3 のように式で表し，運動方程式と連立させて，条件を満たすような拘束力を求める方法である．解析法では，条件式をすべて満たすような拘束

力を求める．

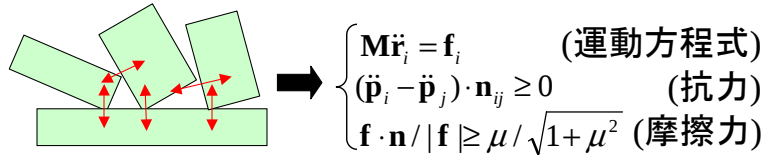


図 2.3: 拘束条件式の例

運動方程式と条件式を連立させるため，拘束力に関与している剛体の数が増えるとそれにつれて，連立させる式の数が増えていく．また，抗力のように条件式が不等式になるものや，摩擦力のように他の拘束力に条件式が依存するものもあるため，計算はあまり簡単ではない．

条件式を正確に解いて拘束力を解くためには，式の数 n として， $O(n^3)$ の計算量が必要になることが知られており，物体を積み上げた場合など多数の物体が接触する場合に多大な計算が必要になる．そのため，多くのシミュレータは繰り返し計算による近似解法を用いて拘束力を解いている．近似解法の計算量は，計算精度・安定性とのトレードオフとなるが， $O(n)$ の計算量で打ち切ることできる．

解析法の近似解法の変形として，2体単位の計算を繰り返し行う方法がある．これは，拘束条件と運動方程式をすべて連立させるのではなく，物体2つの運動方程式と2物体に関係する拘束条件だけを考慮して拘束力を計算するもので，1回の計算では条件を満たす拘束力を求めることはできないが，繰り返し計算することですべての条件を近似的に満たす拘束力を求めることができる [61]．また，衝突力については，衝突の発生する時刻を求め，発生順にこの計算を行う手法 [58] も提案されている．

解析法では，シミュレーションの更新周期 Δt によらず，拘束条件を満たす拘束力が計算できるので，更新周期 Δt を大きく取り，一定時間のシミュレーションに必要な計算回数を少なく抑えることができる．

2.3 2つのシステム構成の提案

計算機によるリアルタイムシステムでは，モデルや提示内容を定期的に更新することで，モデルや提示内容を変化させる．実世界では物事は連続に変化するが，計算機によるリアルタイムシステムでは，更新の周期ごとに離散的に変化することになる．更新速度を速くすると，変化が連続に近づき，シミュレーションやインタラクションの精度が高くなる．また，更新までの時間は遅れとなるため，更新が遅いと遅れも大きくなり，不安定なシステムとなる．このため，安定で精度の高い力覚インタラクションを実現するためには，システムを速く更新する必要がある．しかし，更新が速いと計算量が増えるため，扱えるモデルが小さくなってしまう．より大きなモデルを扱えるようにするためには，提示内容や提示装置の特性，人間の感覚を考慮して，必要最小限の更新速度に抑え，計算量を減らす必要がある．

本研究では，物理シミュレーションにペナルティ法を用いた，シミュレーションの更新が高速で，力覚インタラクションの精度の高いシステムと，解析法を用いた，シミュレーションの更新が低速で，大規模な VR 世界を扱うことができるシステムの2つシステム構成を考える．

2.3.1 ペナルティ法によるシステム

ペナルティ法は，1回の更新に必要な計算量が少ないが，高速更新が必要になる．力覚インタフェースも高速更新を必要とするので，両者を同じ速度で更新することができる．また，力覚ポインタ（ユーザの手のバーチャル世界での表現）をユーザの手の位置に設定し，シミュレーションを行うだけで，力覚ポインタに加わる接触力が計算できるため，力覚レンダリングに特別な処理を必要としない．

ペナルティ法によるシステム構成を図 2.4 に示す．更新は，物理シミュレーションと力覚提示の両方の処理が高速であり，映像提示処理だけが低速で行われる．力覚レンダリングは，物理シミュレータがペナルティ法による接触力計算を用いて行い，求めた接触力をそのまま力覚インタフェースで提示する．

5 章では，ペナルティ法による物理シミュレーションを提案し物理シミュレーション・力覚提示ともに高速に更新を行うシステムを構築する．

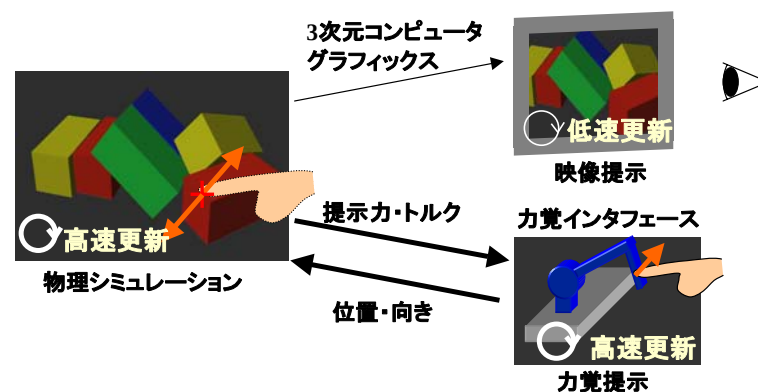


図 2.4: ペナルティ法の物理シミュレータによるシステム

2.3.2 解析法によるシステム

解析法は，1回の更新に必要な計算量が多いが，低速更新でも安定なシミュレーションを行うことができる．力覚インタフェースとは必要な更新速度が異なるため，両者を結ぶ力覚レンダリングにおいて，更新速度の差をうめるための補間が必要になる．また，物体モデル間の干渉の量が変化しても物体モデルに加わる力の大きさは変化しないため，力覚レンダリングにも特別な処理が必要となる．

解析法によるシステム構成を図 2.5 に示す．更新は，力覚提示処理だけが高速であり，物理シミュレーション処理と映像提示処理の両方が低速で行われる．この構成では，力覚提

表 2.1: シミュレーション手法とシステムの特徴

	解析法	ペナルティ法
1回の更新に必要な計算量	多い	少ない
シミュレーションの更新	低速	高速
時間あたりの計算量	少ない	多い
物体運動と力覚の正確性	低い	高い
力覚レンダリング・補間	必要	不要

示処理の更新に対して物理シミュレーション処理の更新が遅いので、力覚レンダリングが整合性を取ることになる。力覚レンダリングは、物理シミュレーションから低い更新周期で送られてくる中間表現（力覚ポイント周辺の物体形状の特徴）を受け取り、力覚インタフェースが高い更新周期で計測した位置・速度と補間処理を施した中間表現から提示力を求める。また、このとき物体に加わる力積を計算してシミュレータ内の物体モデルに反映させる。

このように、解析法によるシステムでは、力覚レンダリング処理が必要な上、更新周期の整合性を取るための処理を行う必要がある。また、整合性を取るための処理を行っても、全体を高速に更新した場合と異なる不自然な挙動を示す場合がある。

6章では、物理シミュレーションを低速に更新するために必要な力覚レンダリングと補間手法の提案し、低速な更新の影響と整合性を取るための処理の効果に評価と考察を行う。

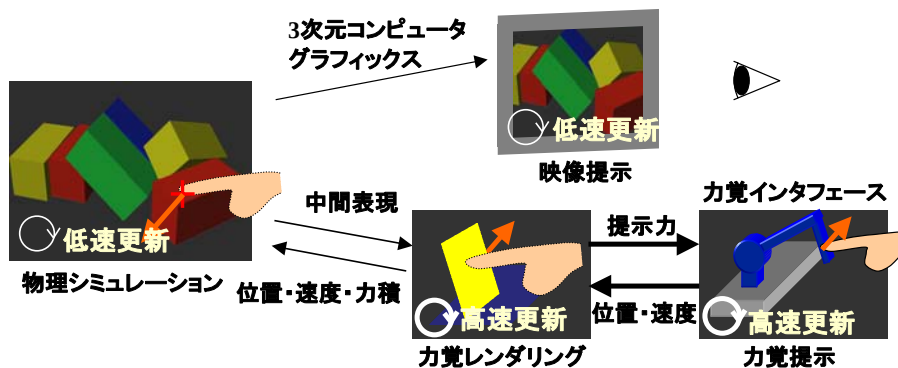


図 2.5: 解析法の物理シミュレータによるシステム

以上のように、解析法によるシステムでは、更新を低速で行うことで、総合的な計算負荷（ $\frac{1 \text{ 回の更新に必要な計算量}}{\text{更新周期}}$ ）を少なくすることができる。しかし、シミュレーションの精度や力覚インタフェースとの接続のしやすさではペナルティ法が勝っている。このため目的や用途に応じてシミュレーション手法を選択する必要がある。解析法によるシステムとペナルティ法によるシステムの特徴を表 2.1 にまとめる。

第3章 力覚インタフェース

力覚インタフェースは，ユーザの操作を入力し，ユーザに力覚を提示する装置である．本章では，力覚インタフェースの機構と機構の制御を概説し，剛体シミュレーション世界を高精度に提示するための機構と制御方法を提案する．

3.1 力覚インタフェースの機構と制御の概要

本節では力覚インタフェースの機構と制御方法の概要を説明する．

3.1.1 力覚インタフェースの機構

力覚インタフェースに限らず，機構には直列機構 (serial mechanism) と並列機構 (parallel mechanism) がある．

直列機構では，アクチュエータがリンクとリンクの間に入るため，根元のアクチュエータがより先端側のアクチュエータ全体を動かすことになり，機構の可動部分が重くなる．また，機構が直列につながるため，ある力を加えたときの先端部の変位は，各リンクと関節の変位の累積になり，剛性は低くなる．WAM Arm [36] のようにワイヤを用いてアクチュエータを地面に配置した機構もあるが，ワイヤの引き回しが難しいほか，ワイヤが伸びるため剛性が低くなってしまう [37] ．

並列機構では，アクチュエータを地面に置くことができるため，アクチュエータ自体が動く必要が無く，機構の可動部分を軽くできる．また，機構が並列につながるため，ある力を加えたときの剛性は各リンクの剛性の和になるため剛性が大きくなる．

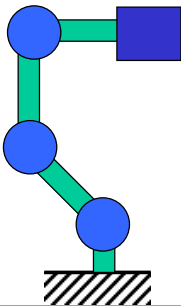
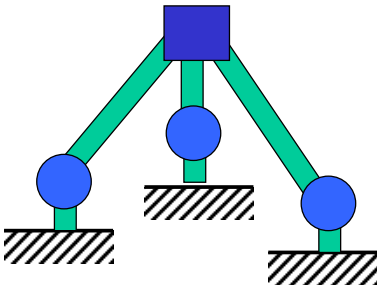
両者の特徴を 表 3.1 に示す．

3.1.2 力覚インタフェースの制御

力覚インタフェースは，

- 機構の状態を計測し，指先位置を求める．
- 指先位置と提示対象との関係から提示力を求める．
- 力を提示する．

表 3.1: 直列機構と並列機構の比較

	直列機構	並列機構
機構の例		
リンクの接続	直列	並列
アクチュエータの配置	地面とリンクの間 リンクとリンクの間	地面とリンクの間のみ
剛性と出力	低い	高い
可動域	広い	狭い
精度	計測誤差が累積する	計測誤差が平均化する

の処理の繰り返しによって制御され、物体にかかる力や物体の形状などの情報を提示する。この繰り返しを高速に行うことで、遅れが少なくなり、安定で高精度な力覚提示が実現する。これまで、力覚インタフェースは 100Hz から 1kHz 程度で制御されていたが、本節ではより高速に制御を行うことで力覚インタフェースの制御の応答を高速化し、高品位な力覚提示を行うことを考える。力覚インタフェースの制御方法には、

1. 軽い機構を力提示可能な摩擦の少ないアクチュエータで駆動するインピーダンス提示型
2. 力センサで提示力を計測しながら、比較的重い機構の位置・速度を制御するアドミタンス提示型

がある。

インピーダンス提示型 (1) では、指定の力を提示できるアクチュエータが必要となる。また、機構が十分軽く、アクチュエータが発生した力がそのまま指に伝わる必要がある。機構をモデル化することで、機構の慣性や粘性を補償し目的の力を提示することもできるが、計測とモデル化の精度と誤差、制御の遅れなどの影響で提示力に誤差が生じる。従って、機構が十分軽く補償の割合が小さい機構の方が安定で使いやすい。この方式では、外部から力が加わった際に機構が動かないようにする (= 硬い物体を提示する) ために高速な制御が必要となる [27]。PHANTom [14]、岩田の Haptic Master [28]、SPIDAR [29] などがこの方式をとっている。

アドミタンス提示型 (2) は、指にかかる力をセンサで計測し、指定の力が指に加わるように指先位置を動かす。アクチュエータが指定の力を提示する必要がないので、摩擦の大きな減速器などを用いた大きなアクチュエータを使用でき、重い機構を用いることもできる。この方式では、外部から力が加わった際にすばやく機構を動かす (= 軽い物体を提

表 3.2: 力覚インタフェースの制御方法

	インピーダンス提示型	アドミッタンス提示型
提示するもの	力	位置
計測するもの	位置	力・位置
高速制御が必要な状況	硬い物体の提示	物体を提示していないとき
機構の特性	軽量	重いものも可
アクチュエータの特性	力制御・低摩擦	位置制御
アクチュエータの例	DC コアレスモータ	ギヤードモータ・ボールねじ

表 3.3: 力覚インタフェースの分類

	インピーダンス提示型	アドミッタンス提示型
直列機構	PHANTOM[31]	Van der Linde らの Haptic Master [16] HBHD [15]
並列機構	Iwata の Haptic Master [28] SPIDAR[23] DELTA/OMEGA haptic device [30]	

示する、または物体を提示しない) ために高速な制御が必要となる [27]。Van der Linde らの HapticMaster[16] などがこの方式をとっている。

アドミッタンス提示型 (2) は、力を計測しなければならないが、力の計測を高速に行うことは簡単ではない。力センサの応答速度は 1kHz 程度のものがほとんどであり、(2) はあまり応答速度をあげられない。また、質量の大きな機構の位置・速度を制御するため、力計測されていない部分が人や物に触れた場合に大きな力が働く危険性がある。一方、(1) は、DC コアレスモータのような応答の速いアクチュエータを用いることができる。しかし、提示機構の慣性の完全な補償は難しいので、機構の慣性が小さくなければならない。また、機構が柔らかいと指先位置を正確に計測できないという問題が生じる。このため、インピーダンス提示型 (1) は、軽量で剛性の高い機構を必要とする。表 3.2 に 2 つの制御方法の特徴を示す。

また、表 3.3 に、力覚インタフェースの制御方法と機構による分類を示す。本研究では、高品位な力覚提示を実現するため、出来るだけ軽量で剛性の高い機構を (1) の方法で高速に制御することで、応答の速い力覚インタフェースを構築する。

3.2 高解像度力覚インタフェースの提案

この節では、動的な VR 世界を高精度に提示するための力覚インタフェースを提案する。物体モデルの形状を精細に提示し、動力学特性を正確に提示することは、高い時空間解像度で力覚を提示しているといえるので、本稿ではこのような力覚提示を高解像度力覚提示と呼び、これを実現する力覚インタフェースを高解像度力覚インタフェースと呼ぶ。高解像度力覚インタフェースは、高速に応答する必要があるため、軽量で高剛性な機構を持ち、

高速に制御される必要がある。

3.2.1 解像度の高い力覚提示のための機構

高解像度力覚インタフェースの機構は、軽量性と剛性を兼ね備えていなければならない。本節では、糸を用いた力覚提示機構 SPIDAR[23] を元に高解像度力覚インタフェースを実現する。糸やワイヤによるパラレル機構は、

- パラレル駆動なので、高剛性で軽量な力覚インタフェースを構築できる。
- 機構が非常に軽量なので、提示可能な機械的インピーダンスの範囲が広い。
- 装置の透明性が高く作業領域での映像提示を妨げない。
- 糸の本数や設置位置、グリップの形状を用途や必要な可動域にあわせて変更できる。

といった特徴を持ち、力覚インタフェースの機構の中でも性能がよく汎用性があり実用性の高い機構である。

糸駆動型力覚インタフェース機構の主な構成要素は、力をユーザの手に伝える糸、エンコーダ付モータ、糸を設置したグリップである。通常はフレームを用意し、フレーム上に設置されたモータとグリップとを糸で接続する。この機構を軽く高剛性にするためには、軽く剛性の高い糸を用い、モーターのローターの慣性を小さくする必要がある。

糸駆動型力覚インタフェースでは、使用する糸の数やその配置によって提示できる力や提示可能域を変更できる。そこで、高品位力覚提示のための糸駆動型力覚インタフェースとして SPIDAR(Fig.3.1) と SPIDAR-G(Fig.3.3) を構築した。SPIDAR は球状のグリップに 4 本の糸を設置した並進 3 自由度の力覚インタフェース [23]、SPIDAR-G は 8 本の糸を設置し並進力と回転力の合わせた 6 自由度の力覚インタフェースである [21]。また、SPIDAR では立方体の 8 つの頂点のうち 4 つを利用しているが、図 3.2 のように残りの 4 つを利用してもうひとつの SPIDAR を構成し、3 自由度 2 指力覚インタフェース SPIDAR-II[29] が構築できる。

使用した糸は、断面積が約 0.04mm^2 であり、材質は高強力ポリエチレンである。この材質は、弾性率約 100GPa、比重 0.97 であり、鋼鉄 (弾性率約 200GPa、比重 7.8) とくらべると、半分の弾性率を 8 分の 1 の質量で実現している。また、モータにはローターの慣性が小さな DC コアレスモータを使用した。これらにより、1 辺が 40cm の SPIDAR(Fig.3.1) の場合で、モータのローターの慣性などを含めた機構の慣性は、手先位置での質量に換算して約 50g となった。また、糸を張った状態でモータのロータを固定したときの手先位置での弾性は約 20kN/m となった。

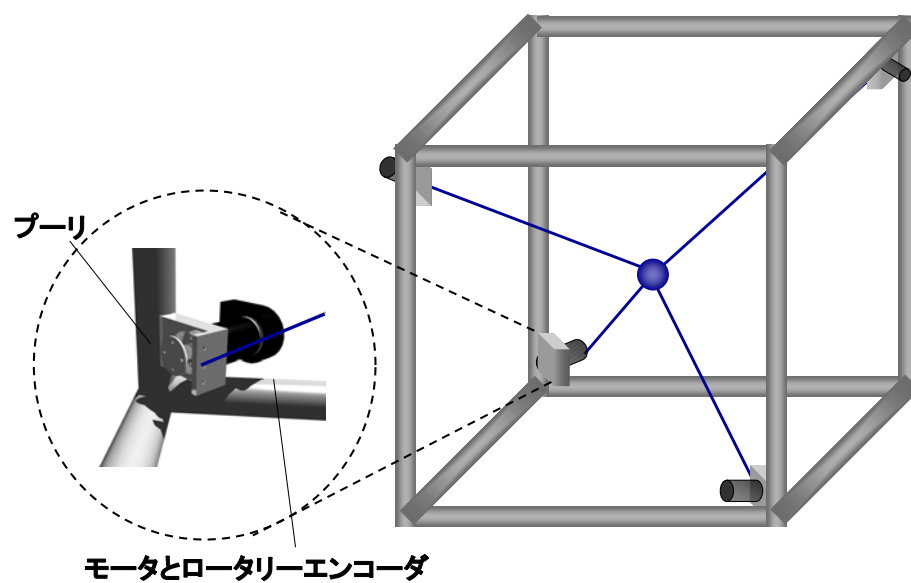


図 3.1: 3 自由度力覚インタフェース SPIDAR

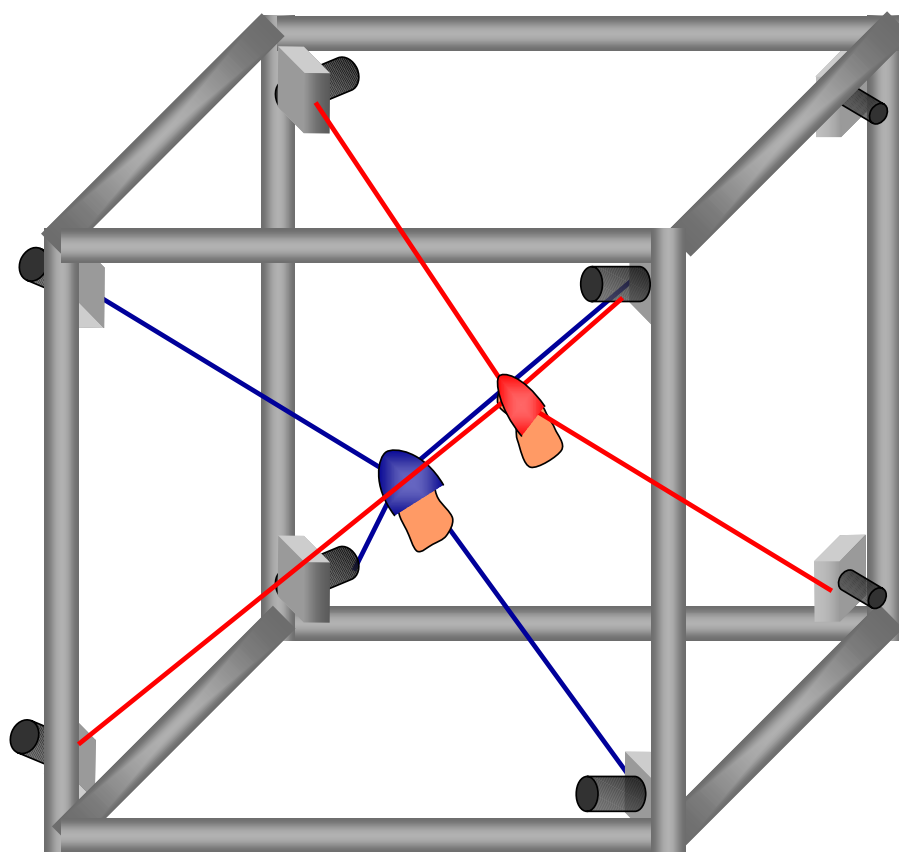


図 3.2: 3 自由度 2 指力覚インタフェース



図 3.3: 6 自由度力覚インタフェース SPIDAR-G

3.2.2 応答の良い駆動方法

軽量の機構を用意しても，アクチュエータやその駆動回路の応答が遅ければ，制御に遅れが出てしまい，高速な制御が実現できない．そこで応答の良いアクチュエータとその駆動回路が必要となる．

DC コアレスモータは，ロータが巻き線だけで構成されており鉄心を持たない．鉄心がないためロータが非常に軽量な上，インダクタンスが小さくヒステリシスによる遅れもないため高速に応答する．ここでは，アクチュエータに Maxon 社製のコアレスモータ RE25(118746) を用いた．

従来の力覚インタフェースでは，PWM 駆動によってモータを電流制御しているものもあるが，応答性を考えるとアナログ電流制御回路を用いる方が望ましい．そこで，高解像度力覚インタフェースでは，駆動回路に 図 3.4 のようなアナログ電流制御回路を用いた．この回路は，D/A コンバータで指令した電圧と，モータとグランドの間の電流検出用抵抗の電圧が等しくなるようにモータに電流を流す．図 3.5 にこの回路の周波数応答を示す．計測はモータをつないだ状態で行っているので，モータのコイルにこの電流が流れてトルクが発生したと考えてよい．

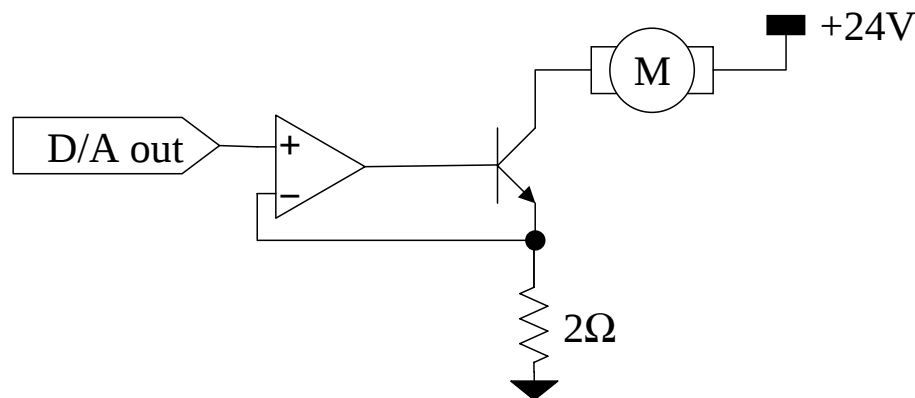


図 3.4: モータ駆動回路

3.2.3 高速制御のためのプロセッサ

高速制御を実現するためには，計測データからモータのトルクを求める計算までを高速に行う必要がある．最近の PC はこの計算を 10kHz の周期で行うことために十分な計算速度を持つ．しかし，Windows や Linux といった通常の PC 用の OS の上で正確に 10kHz 周期で処理を行うことは難しい．そこで，高解像度力覚インタフェースでは，PC とは別に制御専用のプロセッサを用意し，制御回路に組み込んだ．プロセッサには SH4 200MHz を使用した．

PC と制御用プロセッサは USB2.0 で結び，1kHz 程度の周期での更新と大容量の情報転送を可能にした．これにより，ユーザは PC と制御回路を USB でつなぐだけで，高解像度力覚インタフェースを使用できる．図 3.6 に製作した制御回路を示す．

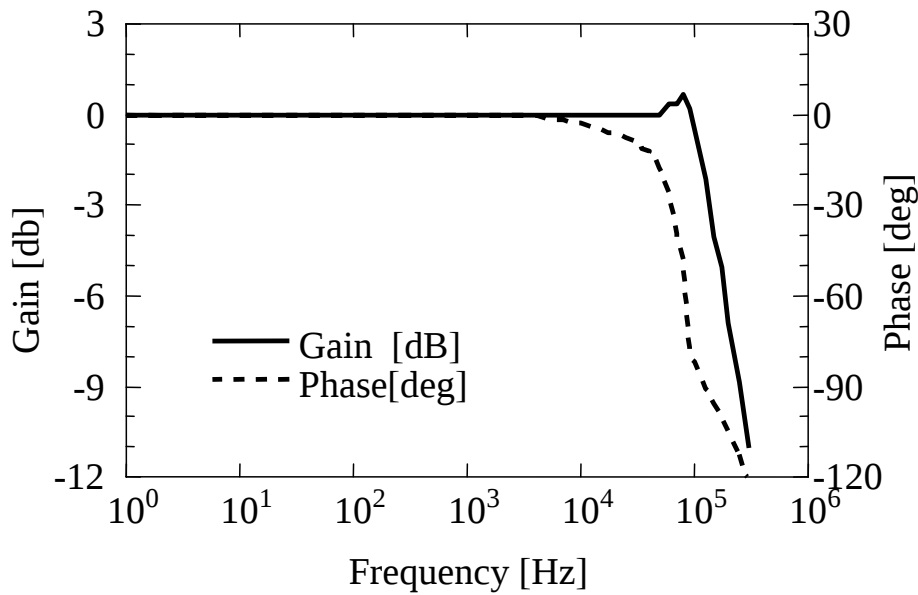


図 3.5: 周波数応答

3.3 系駆動力覚インタフェースの姿勢・力覚計算の概要

本節から，3.5 節では，系駆動力覚インタフェースの姿勢計算，力覚計算の手法を提案する．また，系駆動力覚インタフェースは，系の数や設置位置，グリップの形状を用途に応じて変更できるという汎用性を持つので，3.6 節では，これを生かせるような汎用制御ライブラリを構築する．

力覚インタフェースを用いて物体モデルの形状を提示する際には，まず，力覚インタフェースのグリップの位置・速度などが計測される．次に，力覚ポインタ (VR 世界でのユーザの指や操作に用いる道具の表象) の位置・速度などを計測値にあわせる．力覚ポインタが物体モデルに触れているときは，両者の位置・速度などに応じて提示力を計算する．最後にこの力を力覚インタフェースによってユーザに提示する．形状提示の手順を図 3.7 に示す．

3.3.1 位置計測

系駆動型力覚インタフェースでは，系を巻き取るプーリーにロータリーエンコーダなどの計測器が取り付けられており，系の長さを計測することができる．本節では計測された系長から，グリップの姿勢を求める方法を示す．

系を用いた力覚インタフェースでは，グリップの姿勢が分かれば系長がすぐに求まる．系はグリップとフレームを直接結んでいるので，グリップ上の端点とフレーム上の端点の距離が系長となる．逆に，系長からグリップの姿勢を計算することはそれほど容易ではない．グリップの姿勢から系長を求める式を整理することで，系長からグリップの姿勢を求める式が導出できる場合もあるが，一般の機構について導出する方法はない．

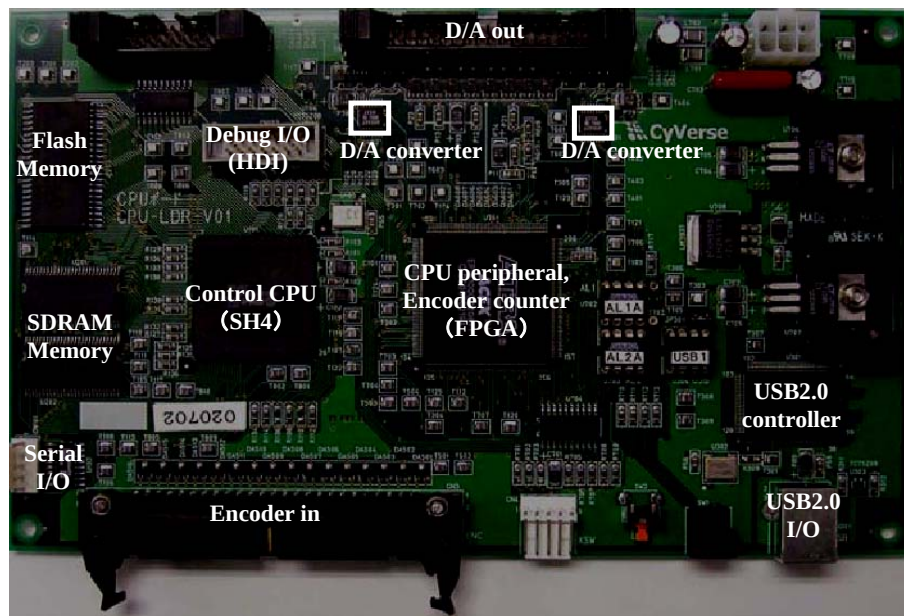


図 3.6: 制御回路

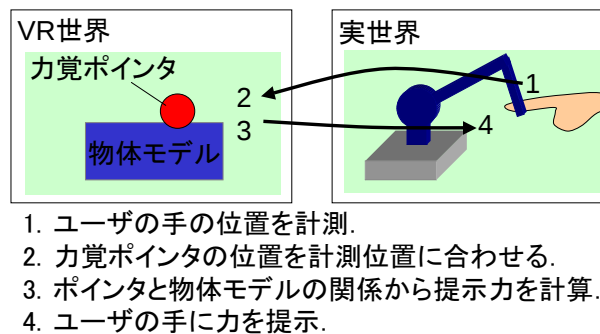


図 3.7: 形状提示の手順

糸駆動型力覚インタフェースの機構の柔軟性という特徴を有効に利用するためには、どのような機構に対しても、糸長からグリップの姿勢を計算できる汎用的な手法が望ましい。そこで、3.4節で姿勢と糸長の変化の関係であるヤコビアンを用いた繰り返し法による計算法を提案する。

3.3.2 力覚提示

糸駆動型力覚インタフェースでは、張力はワイヤや糸の向きにのみ働き、提示力は張力の合力となるため、提示可能な力が限られる。さらに、ワイヤや糸の向きは提示部(グリップ、指キャップなどのユーザが触れる部分)の位置によって大きく変化するため、提示部の位置によって提示可能な力も大きく変化する。このため、任意の向きに十分な力が提示できる領域は提示部の可動域の一部となる。

力覚インタフェースは、任意の向きに十分な力が提示できない場合でも、目標の力に近い力を提示することで物体形状の提示や物体操作の補助をすることができ、ユーザインタフェースとして使用することができる。このときに力覚の正確さのみを考慮して、目標の力にできるだけ近い力を提示するように張力を制御すると、張力が急激に変化する場所ができてしまう。張力が急激に変化すると位置計測の誤差などのため、提示力にも大きな誤差が含まれてしまう。また、位置に対して張力が急激に変化すると発振しやすくなり、安定した操作を行うことができなくなる。

そこで、3.5節で、提示力の正確さだけでなく張力変化の連続性を考慮に入れた新しい張力計算法を提案する。提案方法は張力計算を二次計画問題に帰着させている。二次計画問題は二次形式の最適化問題であり、効率の良い求解アルゴリズムが存在し実用性が高い。

3.4 系駆動力覚インタフェースの姿勢計算

系駆動型力覚インタフェースでは、糸を巻き取るプーリーにロータリーエンコーダなどの計測器が取り付けられており、糸の長さを計測することができる。本節では計測された糸長から、グリップの姿勢を求める方法を示す。

系駆動型力覚インタフェースでは、糸のグリップ上の端点の位置とフレーム上の端点の位置から糸長を求めることができる。このため、糸の数だけの条件式が得られる。

これらの式と機構によって定まる制約式(グリップ上の端点同志の距離の式など)を連立させて、グリップ上の端点の位置やグリップの姿勢が解ければよいのだが、この計算は、どのような機構でも解けるとは限らない。佐藤らのSPIDAR[23]や川村らのFalcon[24]では、糸の取り付け位置を工夫し、この式が解けるような機構を構築して利用している。

系駆動型力覚インタフェースの機構の柔軟性という特徴を有効に利用するためには、どのような機構に対しても、糸長からグリップの姿勢を計算できる汎用的な手法が望ましい。そこで、ここでは姿勢と糸長の変化の関係であるヤコビアンを用いた繰り返し法による計算法を提案する。

3.4.1 ヤコビ行列の利用

ロボットのマニピュレータとしては、直列機構が一般的であり、多くの教科書は直列機構を例に運動学を説明している[22]。直列機構では、系駆動型力覚インタフェースのような並列機構とは逆に、関節角(糸長に相当)から手先姿勢(グリップの姿勢に相当)を求めることは容易だが、手先姿勢から関節角を求める計算は煩雑になることが多い。一方、関節角の微小変化と手先姿勢の微小変化の関係は、変化が微小であるため線形と考えてよく、扱いやすい。この関係は、手先姿勢ベクトルを関節角で偏微分して求めるヤコビ行列(ヤコビアン)と呼ばれる行列によって表される。この関係は線形なため、手先姿勢の微小変化から関節角の微小変化を求めることは容易にできる。そこで、現在の関節角から求めた手先姿勢と目標の手先姿勢の差に基づいて関節角を変化させ、繰り返し計算によって手先姿勢を目標姿勢に近づけて行くことができる。

系駆動型力覚インタフェースは並列機構であり，直列機構の場合とは逆に手先姿勢を求めるためにこれと同様の繰り返し計算を用いることができる．木野らはパラレル系駆動システムにおいてヤコビ行列の疑似逆行列を導入した作業座標系制御法を提案している [35]．木野らは，目標姿勢のフィードバック制御について議論しているが，姿勢計測は外界センサを用いて行っている．

ここでは，機構の変更に柔軟に対応できるより汎用性の高い計算システムの構築を目的とし，ヤコビ行列を利用した繰り返し計算による姿勢計算法を示す．

系駆動型力覚インタフェースの機構は並列機構なので，まず一般の並列機構について議論し，次に系駆動型力覚インタフェースについて議論する．

3.4.2 繰り返し計算による並列機構の姿勢計算

センサからの n 個の入力値 (n 本の系の系長に相当する) のベクトル $l = (l_1, \dots, l_m)$ (以下入力ベクトルと呼ぶ) と，グリップの姿勢ベクトル $r = (r_1, \dots, r_n)$ の関係は，一般には非線形の関係となるので，次のように表す．

$$l = j(r) \quad (3.1)$$

式 (3.1) を r で偏微分して，

$$\Delta l = J \Delta r \quad (3.2)$$

を得る．式 (3.2) から算出される Δr によって r を更新することでグリップの姿勢情報を得ることができる．以下に手続きの概略を示す．

1. 1 ステップ前に得られた姿勢 r_0 から，入力ベクトルの計算値 $l_0 = j(r_0)$ とヤコビアン J を求める．
2. 計測によって得られた入力ベクトル l から，偏差 $\Delta l = l - l_0$ を求める．
3. 式 (3.2) より，姿勢ベクトルの変化量 Δr を求める．
4. Δr から，姿勢ベクトル r を更新する．

3.4.3 系駆動型力覚インタフェースの姿勢計算

次に，この計算法を系駆動型力覚インタフェースに適用する．この場合，入力ベクトルとして通常は各系の長さ (l) を使用する．図 3.8 にグリップ姿勢と系の関係を示す．系端点の位置を p_i とし，エンドエフェクタの姿勢ベクトル r との関係を表す関数を g_i とする (式 (3.3))．関数 g_i は姿勢ベクトル r の形式と端点の位置によって定まる．

m 本の系を使用する力覚インタフェースの場合，系の長さに関する条件式が m 個得られる (式 (3.4))． q_i は i 本目の系がフレーム上に設置された位置座標を表す．

$$p = g(r) \quad (3.3)$$

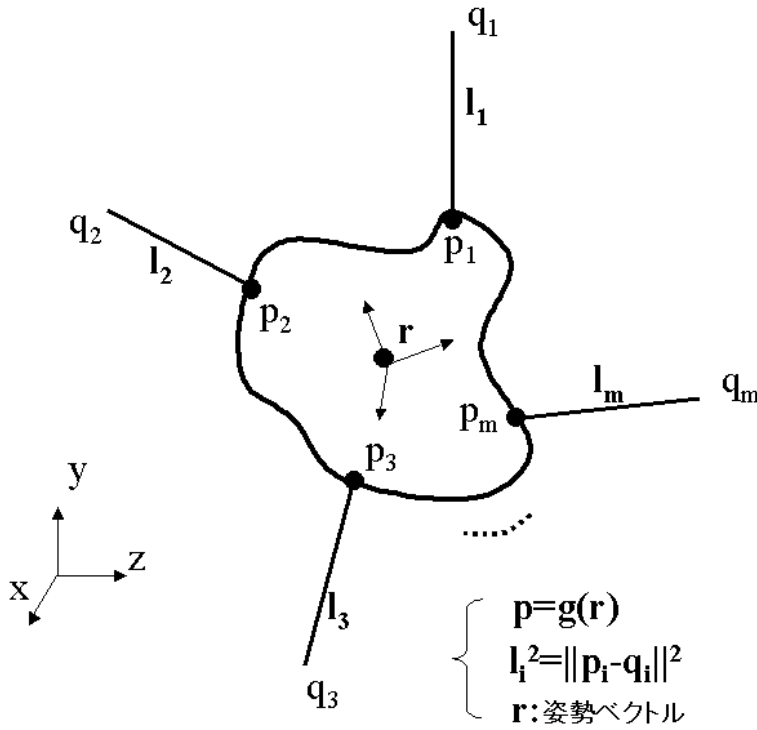


図 3.8: グリップ図：姿勢

$$l_i^2 = \|p_i - q_i\|^2 \quad (i = 1 \sim m) \quad (3.4)$$

これらの式について式 (3.2) と同様に偏微分を行い，差分の関係式

$$\Delta p_i = G_i \Delta r \quad (3.5)$$

$$\Delta l_i = -\phi_i^T \Delta p_i \quad (\phi = (q_i - p_i) / \|q_i - p_i\|) \quad (3.6)$$

を得る．ここで， ϕ_i は糸の方向ベクトルである．姿勢ベクトルの変化 Δr と糸の端点位置の変化 Δp_i の係数 G_i を並べた行列を G ，糸の方向ベクトル ϕ_i を並べた行列を Φ として， m 本の糸についての式 (3.5) 式 (3.6) を行列で表記すると，

$$\Delta p = G \Delta r \quad (3.7)$$

$$\Delta l = -\Phi \Delta p \quad (3.8)$$

$$\text{ここで } \Phi = (\phi_1, \dots, \phi_i, \dots, \phi_m)^T, \Delta p = (\Delta p_1, \dots, \Delta p_i, \dots, \Delta p_m)$$

となる．さらに，式 (3.5) を式 (3.6) に代入して，糸の長さの変化量とグリップの姿勢の変化量との関係式式 (3.9) を得る．

$$\begin{aligned} \Delta l &= -\Phi \Delta p \\ &= -\Phi G \Delta r \\ &= -M \Delta r \quad (M \equiv \Phi G) \end{aligned} \quad (3.9)$$

以上より、扱う系駆動型力覚インタフェースごとに行列 Φ や G 、またはその積で得られる行列 M を用意することで、任意の機構においてワイや長さの変化量 Δl からグリップの姿勢の変化量 Δr を得る式が得られる。よって先に記述した、一般の並列機構の手先姿勢を求める手続きによってグリップ姿勢を計算することができる。ここで、連立方程式 (3.9) は通常は冗長系であるので、 Δr を算出するために、そこで行列 M の疑似逆行列 M^+ を使用することになる。

$$\Delta r = -M^+ \Delta l \quad (3.10)$$

3.4.4 SPIDAR の姿勢計算

3.4.2 節では、系駆動型力覚インタフェースの一般的な姿勢計算の解法について記述した。本節では、この姿勢計算を、SPIDAR や SPIDAR-II のような並進運動のみの機構と SPIDAR-G のような並進・回転運動する機構 (3.2.1 節参照) について適用する。

並進運動のみの機構の場合、グリップ上の系の端点は1点であると考えられる。グリップの姿勢は端点の位置座標だけで表されるので、式 (3.3) の g は 3×3 の単位行列であればよく $p = r$ となる。またこのとき行列 M は、式 (3.6) で定義した系の方向ベクトル ϕ_i を用いて、

$$M = - \left(\phi_1, \quad \cdots \quad \phi_m \right)^T \quad (3.11)$$

と表すことができる。

並進・回転運動する機構では、グリップの姿勢を重心の位置座標と x, y, z 方向の回転角度で表すことにし、 $r = (x, y, z, \theta_x, \theta_y, \theta_z)$ とする。 x, y, z 軸方向に回転する回転行列をそれぞれ $R(\theta_x), R(\theta_y), R(\theta_z)$ として、式 (3.3) の g のうち、回転部分の変換を表す部分を g_r とすると、

$$g_r = R(\theta_z) \cdot R(\theta_y) \cdot R(\theta_x) \quad (3.12)$$

となる。これを $(\theta_x, \theta_y, \theta_z)$ で偏微分して差分の係数を求めると、

$$G_r = \begin{pmatrix} 1 & \theta_z & -\theta_y \\ -\theta_z & 1 & \theta_x \\ \theta_y & -\theta_x & 1 \end{pmatrix} \quad (3.13)$$

を得る。この行列と、式 (3.19) で定義した方向ベクトルを用いて行列 M を算出すると、式 (3.11) と同様の形式にまとめられ、3.4.3 節の方法で姿勢計算ができる。

3.4.5 提案手法の特徴

ここで、逐次計算による姿勢計算法の特徴を挙げておく。

- 機構ごとに使用する行列、ベクトルの形式やサイズを変更することで、同じアルゴリズムを用いながら機構の変化に対応できる。

- 計算式の生成，連立方程式や二次計画法などの各計算ライブラリの使用，プログラムのフレームワークなど，すべての装置で共通して使用するものを再利用できるので，新しい装置の開発者はこれらを新たに作り直す必要がない．
- 特定の未知数のみに誤差が乗るということではなく，未知数全体に誤差が分散されるので，その影響は少なくなる．また代数計算を行う方法よりも誤差の影響を評価しやすい．

3.4.6 計算速度の評価

姿勢計算の実行速度を測定した．シミュレーション上でグリップを連続的に運動させて，純粹に姿勢計算のみにかかる時間を計測した．なお実験環境は3.5.7節で行った実験と同様である．1ステップ内で計算を繰り返し行わなかったとした場合，装置(a)の平均実行速度は約357[kHz]（実行時間2.8[μ s]），装置(b)の場合，平均実行速度は約76[kHz]（実行時間13.2[μ s]）であった．また，1ステップ内での繰り返し回数と実行時間とがほぼ比例関係であることを確認した．今回の実験結果から，姿勢計算処理が装置の動作速度を遅くする原因となる可能性は低いといえる．なお今回の実行時間の測定はシミュレーション上で行ったため，エンコーダの出力値から糸の長さを得るまでの時間を考慮していない．そのため，実際の姿勢計算にかかる実行速度は今回の結果よりも遅くなる．

3.4.7 精度・繰り返し回数の評価

姿勢計算の精度と計算の繰り返し回数について検証した．今回提案した姿勢計算方法は1ステップ（装置の1更新周期）内で繰り返し計算を行うことで精度が上がる．本実験では，インタフェースとして使用するために十分な精度の解を得るには，何回の繰り返し計算が必要であるかを調べた．

並進移動の計測には装置(a)を，回転運動の計測には(b)を使用した．装置の中心付近でFig.3.9に示した方向に，グリップを瞬時に移動させる．このとき，何回の姿勢計算で精度の良い解が得られるかを調べた．各装置の更新時間は1[ms]であるとした．

装置(a)についてはグリップを瞬時に並進運動させ，グリップの重心位置について並進運動の移動速度と，重心位置座標の厳密解との絶対誤差との関係を調べた．絶対誤差として，厳密解と繰り返し計算による近似解との距離を算出した．その結果をFig.3.10に示す．今回の実験結果の限りでは，繰り返し計算を行わなかった場合は移動距離が大きくなるにつれて誤差が増大しているが，2回計算を行った場合は計測を行った範囲で，厳密解とほぼ等しい値が得られていることが分かる．

装置(b)についてはグリップを瞬時に回転運動させ，グリップ上の糸設置位置の一点について，移動速度（回転速度）と誤差の関係を調べた．先程と同様に，絶対誤差として厳密解と近似解との距離を算出した．その結果をFig.3.11に示す．並進運動の場合と同様に，繰り返し計算を行わなかった場合は移動距離が大きくなるにつれて誤差が増大しているが，2回計算を行った場合は比較的精度の良い解を得られたことが分かる．

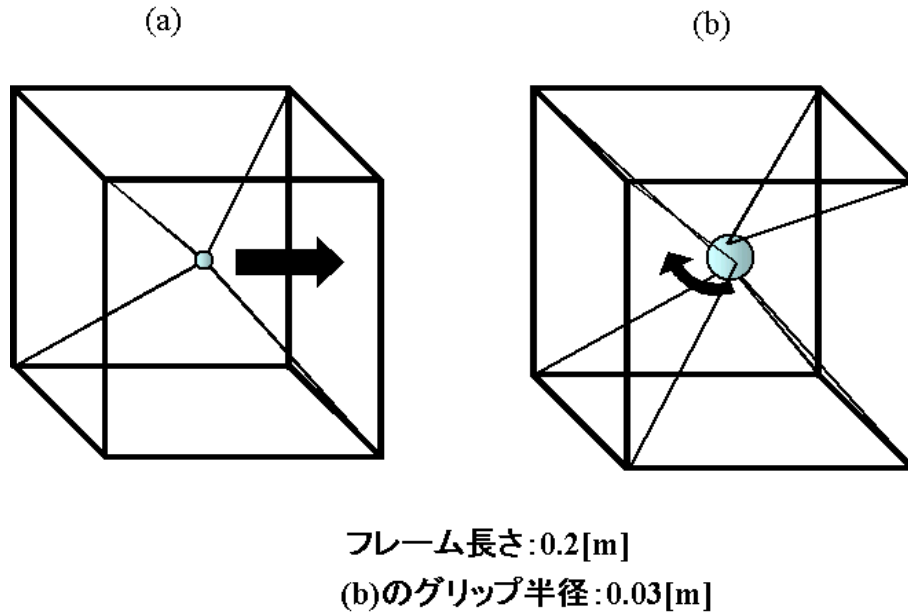


図 3.9: 実験図：グリップの操作方向

以上より，1 ステップ内に 2 回繰り返し姿勢の算出を行うことで，精度の良い解が得られることが分かった．但し，今回の実験はあくまで今想定した装置での結果であり，機構やグリップの計測位置，運動方向により結果は異なる可能性がある．しかし，人間が短い更新時間内にグリップを動かすことのできる距離は装置が異なってもそれほど大差ないと予想され，今回の結果とから大きく外れることは少ないと考えられる．

3.5 系駆動力覚インタフェースの力覚計算

3.5.1 系駆動型力覚インタフェースの定式化

本節では，力覚計算の一般形式を示す．提示したい力（例えば並進力，回転力）を成分としたベクトルを力覚ベクトル f ， m 本の系の各張力を成分としたベクトルを張力ベクトル τ とする．張力ベクトル τ により提示される力覚ベクトルは

$$f = W\tau$$

$$W = (w_1, w_2, \dots, w_m)$$
(3.14)

という形式で表される． $w_i (i = 1 \sim m)$ は各系の単位張力が発生する力覚ベクトルである． w_i の形式は機構と力覚ベクトル f の形式によって異なる．以降では， w_i を系ベクトル，行列 W を系行列と呼ぶことにする．一般的なグリップと系における力覚情報の関係図を図 3.12 に示す．各装置は m 本の系を使用しているとし， p_i は各系のグリップ上の設置位置を， q_i は各系のフレーム上の設置位置を表す．

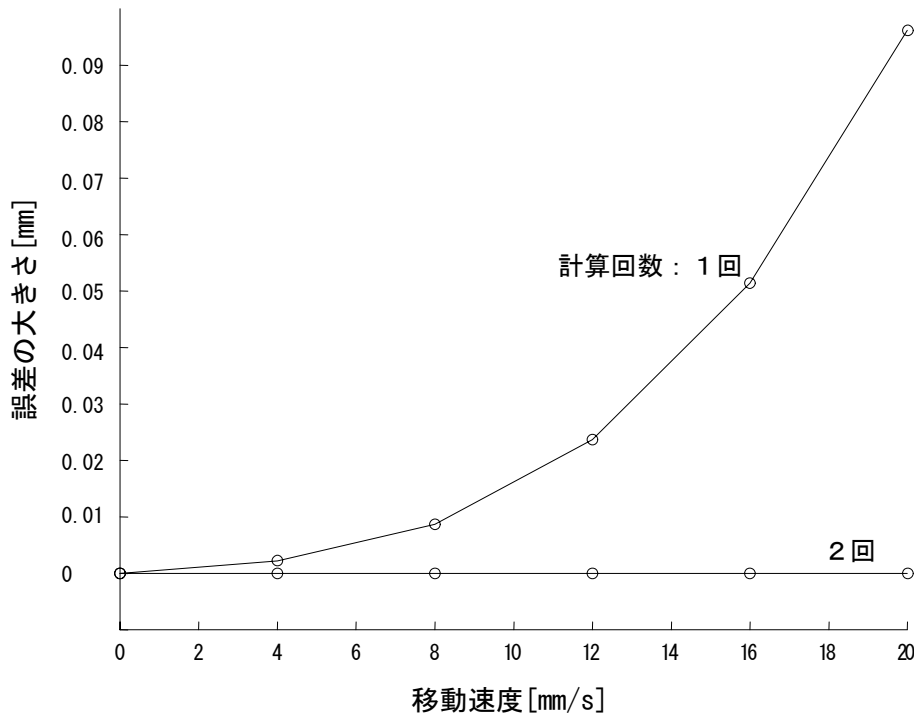


図 3.10: 並進運動 グリップ移動速度と姿勢誤差

3.5.2 完全提示域と部分提示域

張力型力覚提示装置では，張力が系の向き（提示部から系の繰り出し口に向かう向き）にしか働かない．このため，提示部の位置によっては一部の力覚ベクトルを生成することができない．

任意の力覚ベクトルを提示するためには，系の向きが Force-Closure [45] の条件を満たさなければならないことが知られている [46]．Force-Closure の条件とは，

1. m 本の系ベクトルのうち，どの n 本をとってもその系ベクトルが線形独立
2. 系ベクトルを w_i として，

$$\sum_{i=1}^m \alpha_i w_i = 0 \quad (\alpha_i > 0). \quad (3.15)$$

を満たす α_i が存在すること．

である．しかし，この条件を満たさない場合でも方向によっては力覚提示が可能なので，このような場合にも，この機構を力覚インタフェースとして使用することはできる [32]．本論文では，任意の向きの力が提示可能な領域を「完全提示域」と呼び，その外側の部分的な力提示しかできない領域を「部分提示域」と呼ぶ．つまり，提示部が完全提示域にあれば任意の向きに力が提示できる．図 3.13 に，2次元平面内の力を3本の糸で提示する力覚提示装置について，完全提示域と部分提示域を図示する．

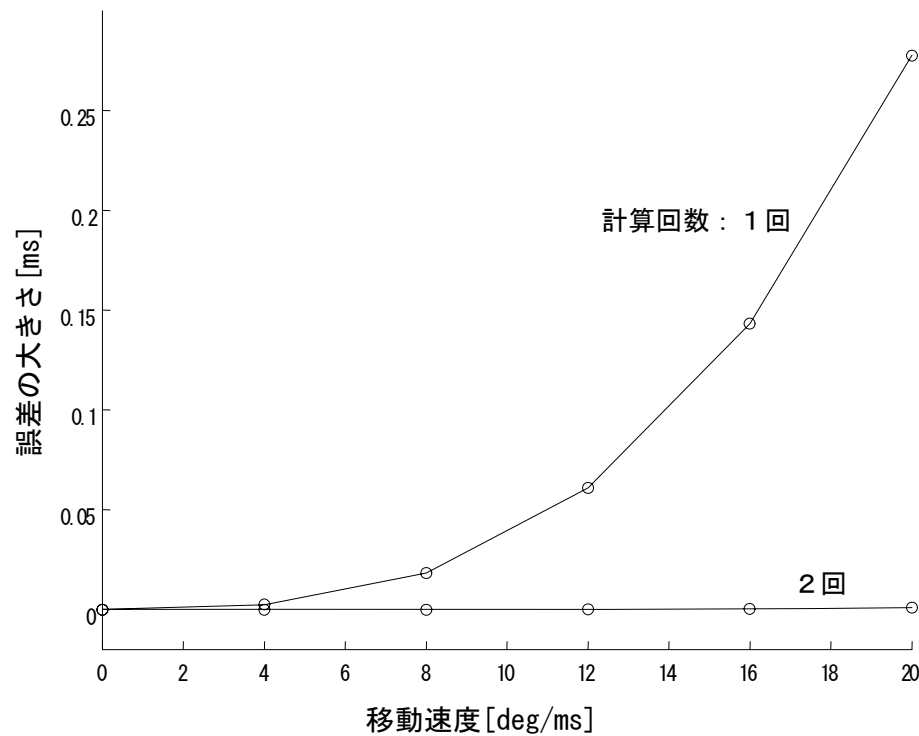


図 3.11: 回転運動 グリップ移動速度と姿勢誤差

3.5.3 領域境界での張力の不連続性

張力型力覚提示装置は完全提示域内では任意の向きの力を提示できるが、力提示に必要な張力は、提示部の位置によって大きく変化する。特に、系の向きと提示力の向きが大きく異なる場合、提示力に対して大きな張力が必要となる。

例として、2次元平面内の力を3本の系で提示する力覚提示装置を考える。図3.14は目標提示力 f にできるだけ近い力を提示したときの張力を示している。モータを各頂点とする三角形内が完全提示域となる。

提示部が完全提示域の中央に位置する場合(図3.14-A)、力提示に寄与する2本の系の方向ベクトルと提示力の向きが近いため、比較的小さな張力で目的の力を提示できる。提示部が完全提示域の周辺部に位置する場合(図3.14-B)、力提示に寄与する2本の系の方向ベクトルと提示力の向きが離れているため、目的の力を提示するために大きな張力が必要となる。提示部が部分提示域に位置する場合(図3.14-C)、提示力向き成分を含む方向ベクトルの系が存在しないため、力を提示することができない。このときは、系張力を0にして不要な力を提示しないようにする。

力覚提示装置が一定の力 f を提示し、ユーザーが提示部を中心から外側に向かって動いた場合、力覚提示装置は、図3.14-Aの状態から図3.14-Bの状態を経て図3.14-Cの状態に変化する。このとき張力は、図3.14-Bの状態ですごく大きくなり、その後、図3.14-Cの状態ですぐ0となる。

このように一定の力にできるだけ近い力を提示しつつ、提示部を境界面を跨いで動かすと、完全提示域の境界面で張力の大きさが急激に変化することがある。張力が急激に変化

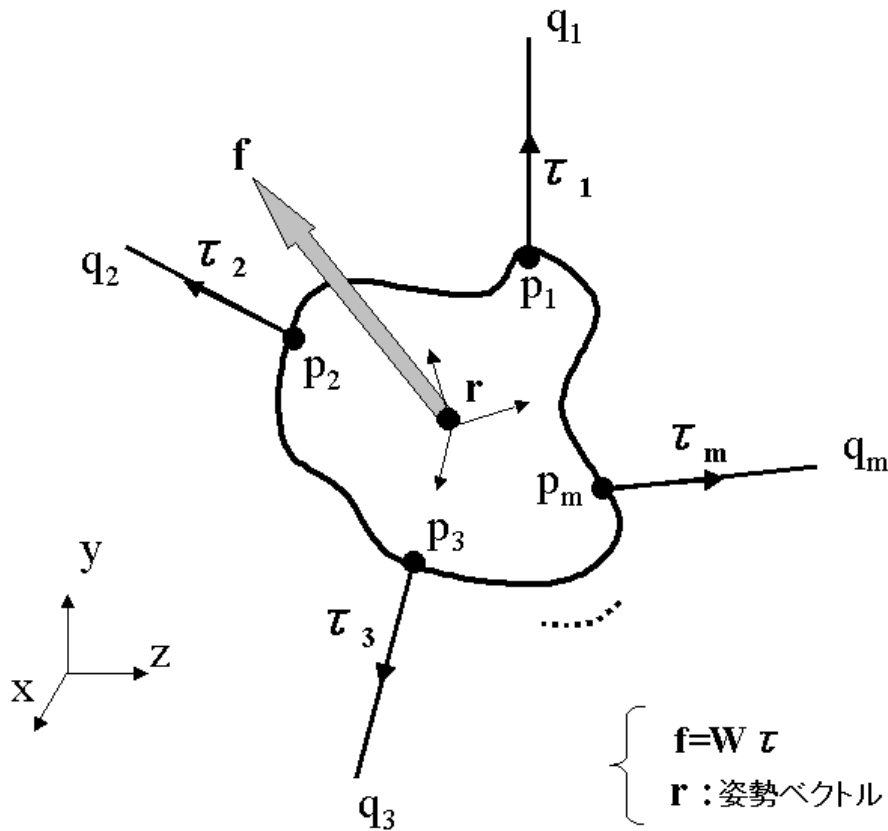


図 3.12: グリップ図：力提示

すると位置計測の誤差などのため、提示力にも大きな誤差が含まれてしまう。また、位置に対して張力が急激に変化すると発振しやすくなり、安定した操作を行うことができない。この現象は多自由度力覚インタフェースでも同様に生じる。

そこで、提示力の正確さだけでなく張力変化の連続性を考慮に入れた新しい張力計算法を提案する。提案方法は張力計算を二次計画問題に帰着させている。二次計画問題は二次形式の最適化問題であり、効率の良い求解アルゴリズムが存在し実用性が高い。

3.5.4 二次計画法による力覚計算

本節では、提案する計算方法を示す。力を安定に提示するために、張力ベクトル τ を求める問題を二次形式の最適化問題に置き換える。

$$Z = |\mathbf{W}\boldsymbol{\tau} - \mathbf{f}|^2 + \lambda|\boldsymbol{\tau}|^2 \longrightarrow \min \quad (3.16)$$

$$\tau_{\min_i} \leq \tau_i \leq \tau_{\max_i}$$

式 (3.16) 中の τ_i は張力ベクトルの各成分（各系の張力値）、また λ は正定数である。目的関数 Z の第1項目は式 (3.14) より、提示したい力と計算によって得られる力の大きさの

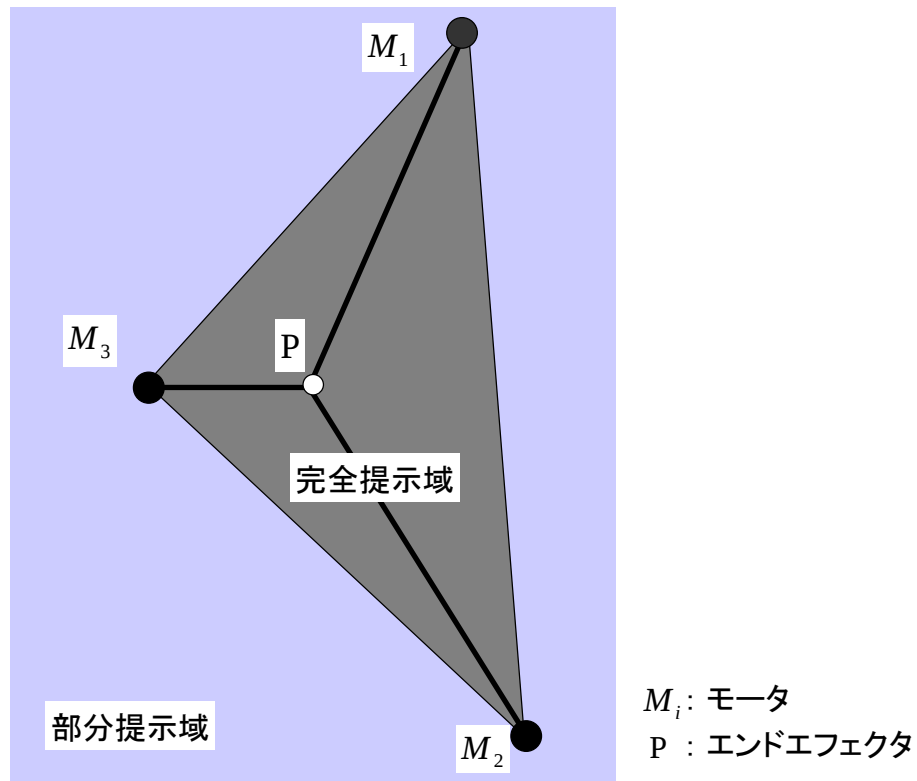


図 3.13: 3 本の系による力覚インタフェースの完全提示域と部分提示域

差をできるだけ小さくする目的で導入した項である．式 (3.16) の目的関数 Z はさらに式 (3.17) のような二次形式で表現できるが，式 (3.16) の Z の第 2 項目を追加したことで，二次係数行列 Q の対角成分に正値が加えられるので正定置対称行列となり，得られる各系の張力値の一意性・連続性が保証される．

$$Z = \frac{1}{2} \tau^T Q \tau + c^T \tau \quad (3.17)$$

このようにして張力を計算することで，振動の生じない安定した力の提示が可能となる．式 (3.16) のように，ある制限域内で二次形式の最適解を求める問題は二次計画法と呼ばれ，高速な解法が提案されている．問題の次元数は糸の本数で決定されるのでそれほど大きくならない．従って比較的小規模な二次計画法の解法を用いて高速に解を算出できることが期待できる．

3.5.5 SPIDAR の力覚計算

前節の計算方法を，姿勢計算の場合と同様に，SPIDAR のような並進力提示する機構と SPIDAR-G のような並進力・回転力を提示する機構 (3.2.1 節を参照) について適用する．

並進力 3 自由度を提示する装置の場合，グリップを点で表すことができる．力覚ベクトル f の成分は質点にかかる並進力の各軸方向成分となるので， $\phi_i = (q_i - p_i) / \|q_i - p_i\|$

を用いて，

$$w_i = \phi_i \quad (3.18)$$

と設定できる．

次に，並進・回転力 6 自由度を提示する装置の場合，グリップを剛体として表すことができ，力覚ベクトル f の成分は並進力，回転力の各軸方向成分となる．よって，グリップの重心から i 番目の系の設置位置までのベクトルを d_i とすると

$$w_i = \begin{pmatrix} \phi_i \\ \phi_i \times d_i \end{pmatrix} \quad (3.19)$$

と設定できる．

以上のように係行列を設定し，二次形式の目的関数 Z を作成して，二次計画法により Z を最小にするときの張力 τ を求めればよい．

3.5.6 精度と安定性

式 (3.16) の Z の第 2 項目の追加は，提示力の精度を落とすことに相当する．しかし，操作の安定性という要素はユーザインタフェースにとって必要不可欠なものであるので，無視することはできない． λ は各項の比重を決定する定数であるので，構築する仮想環境が精度と安定性のどちらを重視するかを考慮した上で， λ の値を適切に決定する必要がある．

3.5.7 計算時間の評価

力覚計算における目的関数 (3.17) の係数行列や係数ベクトルは，グリップの姿勢や提示力の指示値によって決定される．そこで，力を提示しながらグリップを連続的に動かす実験を複数回行うことで目的関数のパターンを変えながら，力覚計算の速度を求めた．使用した計算機環境は OS:Microsoft Windows2000，CPU:Pentium4 1.5GHz である．実験の結果，装置 (a) の平均実行速度は 159 ～ 250[kHz] (実行時間 4.0 ～ 6.3[μ s])，装置 (b) の平均実行速度は 35 ～ 45[kHz] (実行時間 22.2 ～ 28.6[μ s]) であった．グリップの状態や提示力の指示値によって解の探索時間が大きく異なる．ユーザに違和感を覚えさせない程度の自然な力を提示するには，システム全体でおよそ 1[kHz] 程度の実行速度が必要である．今回の実験結果は 1[kHz] を大きく上回っているため，力覚計算処理がシステム全体の動作速度を遅くする原因となる可能性は低いといえる．

3.5.8 提示部の位置と張力の関係

正確性に対する連続性，効率性の比重を決める定数 λ の効果を示すため，一定の力を目標提示力としたときの提示部の位置と張力の関係をさまざまな λ に対して計算した．実験には，図 3.23(a) の機構を一辺 0.5m の大きさに試作して用いた．

図 3.15 のグラフは，一定の力 $f = (5, 5, 5)(N)$ を目標の提示力とし，提示部をフレームの左下奥の点 $(-0.27, -0.27, -0.27)[m]$ と右上手前の点 $(0.27, 0.27, 0.27)[m]$ を結ぶ直線上に配

置した際 (図 3.16 を参照) に算出される系の張力を表している。グラフには対称な 3 本の系の張力を示した。グラフから、 $\lambda = 0$ のときには、完全提示域と部分提示域の境界面で、提示張力が最大張力から最低張力へ急激に変化することが分かる。 λ が大きくなるにつれて提示張力の最大値が小さくなり、境界での張力の変化も小さくなる。

3.5.9 正確性と安定性の評価

張力計算において、提示される力の「正確性」を重視した場合と「連続性」「効率性」を重視した場合を比較する実験を行なった。3.5.8 節と同様に、図 3.16 のように一方向に一定の力を発生させ、力と同じ向きに提示部を動かしたときのそれぞれの出力値と指先位置の変化を調べた。 $f = (5, 5, 5)[N]$ の力覚提示を行い、指先を頂点から向かいの頂点へ力と同方向へ移動させた。このとき、実際の使用状態に近い状態となるよう、指や腕にはガイドを付けず、目視で軌道を確認しながら手先を移動した。式 (3.16) で $\lambda = 0$ のときが提示力の「正確性」だけを条件とした場合に、 $\lambda \geq 0$ のときが「連続性」「効率性」を条件に加えた場合に相当する。 λ の値を大きくすると、より「連続性」「効率性」に比重を置くことになる。図 3.17 は λ の値を変化させたときの、提示部の x 座標と力の大きさの x 成分との関係を表すグラフである (対称性から y 方向、 z 方向についても同様の結果が得られる)。

$\lambda = 0$ のときは完全提示域の境界面を提示部が通過する際に張力の急激な変化により振動が生じた。目標提示力は一定値 ($(5, 5, 5)[N]$) だが、張力の急変が位置や提示力の急変を引き起こし、振動したと考えられる。

一方 $\lambda \geq 0$ のときは提示力は連続的に変化しており振動は生じなかった。 λ を大きくすると目的の力よりかなり小さい力しか提示することができなかったが、提示力の大きさはより緩やかに変化し、操作が容易になった。

正確性だけを考慮した力覚提示法では、可動域内に指先に振動が生じる場合があった。そのため振動の生じない、安定な操作が可能な領域は完全提示域内に限定されてしまう。またある場所を通過することで急に振動が生じるような状況下では、ユーザの心理的負担も大きくなる可能性がある。新しい提示法を導入したことによりこのような問題点が解消されて可動域全体で自由に指を動かせるようになり、使用感が向上した。

また、目的関数の係数を変化させると提示される力覚情報の正確性、安定性、出力範囲が変化することが分かった。正確性、安定性、操作域のすべての要素を完全に満足させることはできないが、使用目的に応じて目的関数の各項の比重を調整することで、用途に適した力覚提示が可能となると考えられる。

3.5.10 提示形状の評価

単純な形状を提示し形状をなぞった時の軌跡を比較することで、連続性を考慮したことの効果を検証する実験を行った。実験条件の概略を図 3.18 に示す。

提示する物体モデルは半径 0.20[m] の円筒体とし、円筒体の中心軸は、外枠の立方体の中心を空間座標の原点としたときの y 軸と一致するようにした。この円筒体を $y = 0$ 平面内で一周なぞったときの指先の軌跡を計測した。実験に用いた SPIDAR のフレームの一

辺は0.5[m]なので、円筒の断面は、 $y = 0$ 平面内で完全提示域と部分提示域に跨る。物体と指の間の摩擦はないものとし、ユーザの指先が円筒に侵入したときに、侵入距離に比例した反力を発生をさせるばねモデルを用いて、物体モデルを提示した。実験ではばね定数を2[kN/m]とし、指先があまり円筒に侵入しないようにした。

力覚提示方法は、目的関数を $\lambda = 0$ として「正確性」のみを条件とする場合 (A) と、 $\lambda = 0.2$ として「連続性」も考慮した場合 (B) を比較した。指先の軌跡を図 3.19 に示す。図内のひし形は完全提示域と部分提示域との境界面を表している。ひし形内が完全提示域である。

「正確性」のみを条件とした場合 (A) は、張力が不連続に変化するため振動が生じ、表面を滑らかになぞることができなかった。一方、「連続性」をも考慮した場合 (B) は、不要な力を感じることなく滑らかになぞることができた。「正確性」を多少犠牲にしても「連続性」を条件に加えたほうが、かえって精度の高い物体モデルを提示できるといえる。

3.6 汎用制御ライブラリの構築

姿勢計算ではヤコビ行列を用いた繰り返し計算法、力覚計算では二次計画法を適用することで、提示力、グリップやフレームの形状、系の設置位置や本数といった機構の特性の変更に対応可能となった。これらの利点を有効に活用できるようにするため、姿勢・力覚計算を包括する C++ によるクラスライブラリを作成する。また、数値計算ライブラリとして、連立方程式を解くためにガウスの消去法とコレスキー分解法、二次計画法を解くために有効制約法を用意した。

3.6.1 機構依存部分の分離

始めに計算処理部分を、機構の異なる様々な系駆動型力覚インタフェースで共通に使う部分（基幹部分）と、機構に特有な情報を含む部分（機構部分）に分離した。システムの汎用性を高めることと、装置の開発者が容易に理解できるシステムの製作を目的としているので、機構部分に記述すべき内容を可能な限り少なくすることを考慮した。その結果、機構部分は、

- 系の本数，算出点の数
- 系行列
- 初期状態におけるグリップ位置座標と系の長さ
- モータの位置，モータトルクの最大・最小値

の情報を持ち、基幹部分は、

- 姿勢，張力，提示力
- 連立方程式（姿勢計算）

- 二次形式関数（力覚計算）
- プログラムのフレームワーク

などの処理を行うことになった。

次に汎用化を達成するためのクラス階層設計を行った。図 3.20 にライブラリのクラスの階層図を示す。まず系駆動型力覚インタフェースを表す抽象的な基本クラスを作成した。基本クラスは連立方程式や二次計画法のライブラリを使用するためのフレームワークをメンバとして所有する。次にこの基本クラスを継承して、系の長さ情報を用いて並進力や回転力を提示する装置のためのサブクラスを作成した。この階層のクラスは、提示力の種類によって形式が決定される力覚ベクトル、姿勢ベクトルなどをメンバとして所有する。提示する力の自由度によって、プログラム中の処理過程やメモリ確保処理などが異なる。回転力を提示する場合の処理は、並進力のみを提示する装置の処理に加えて、回転モーメントを求める関数など、いくつかの処理関数をメンバに追加することで実現可能であるので、図 3.20 のように並進力提示装置の抽象クラスを継承して、並進・回転力提示装置用の抽象クラスを作成した。

3.6.2 新規機構への対応

新しい機構を持つ力覚インタフェースに対応するためには、機構に対応したクラスを用意する。多くの系駆動型力覚インタフェースは、系の長さ情報を使用し、並進力または並進・回転力を提示するためのものである。その種に属する新しい装置を開発したい場合には、Fig.3.20 のようにすでに定義された抽象クラスを継承して、新しいクラスを作成すればよい（Fig.3.21：装置 1）。そして、機構部分のみをメンバとして新たに定義すれば、姿勢・力覚計算を行うことができる。既存の装置に新しい機能や提示力の自由度を加えた装置を開発する場合は、既存のクラスを継承して新しい抽象クラスを作成する必要がある。そしてそのクラスを継承した新しいクラスを作成すればよい（図 3.21：装置 2）。また、系の長さ情報以外のものを使用する装置のように、既存のものとは異なる装置を開発する場合は、より上位の基本クラスから継承しなければならない（Fig. 図 3.21：装置 3）。計算処理システムをこのように基幹部分と機構部分に分離した仕様にするすることで、力覚インタフェースの開発者は処理システム全体を知ることなしに、必要となる部分のコーディングを行うだけで姿勢・力覚計算を行うことが可能となる。

3.6.3 計算システムの位置付け

系駆動型力覚インタフェースの処理の流れについては 3.3 節で説明したが、実際に系駆動型力覚インタフェースを使用するアプリケーションを作成した場合の、本システムの位置付けを図 3.22 に示す。制御システムには姿勢・力覚計算を行う部分の他に、ハードウェアとの通信処理や情報の変換処理を行う部分を含む。アプリケーション側では仮想環境内での運動・衝突などの物理演算を行い、グリップに返すべき力を毎更新周期ごとに計算する。アプリケーション側からは計算システムの中身を参照・変更することはできない。

行えることは、1. 各設定値の変更（姿勢計算の繰り返し回数，提示力の精度・安定性を調整するための値の変更など），2. 提示力の大きさの設定，3. 算出された姿勢情報の取得である．この制限は図 3.20 の各クラスの設定によって決定される．今後，この制限の設定方法の再検討など，装置の開発者やアプリケーションの開発者・ユーザの意見を取り入れることで両者にとってより使いやすいシステムの構築を行っていく必要がある．

3.6.4 汎用性の評価

作成したソフトウェアライブラリを用いて複数の機構のための制御ソフトウェアを作成し，動作確認を行うとともに，システムの汎用化が達成されたかどうか評価した．図 3.23 に今回想定した力覚インタフェースの機構を示す．(a) 4 本の糸を用いてグリップが並進運動を行う機構 (b) 8 本の糸を用いてグリップが並進・回転運動を行う機構 (c) 6 本の糸を用いてグリップが並進・回転運動を行う機構である (a) と (b) は継承する抽象クラスが異なる場合 (b) と (c) は同じ抽象クラスから継承して作成し，糸の本数，グリップ上の設置位置の数が異なる場合を確認するために想定した（図 3.20 参照）．

3 つの装置について，3.6 節で作成したクラスライブラリを用いて計算プログラムを作成し計算を行った結果，姿勢・力覚計算ソフトウェアがともに正しく動作することを確認した．装置 (a) の姿勢・力覚計算を行うために，新たに 60 行程度のコーディングが必要であった．また装置 (b) (c) の場合は 100 行程度のコーディングが必要であった．計算方法や製作者が違うことから開発量を単純に比較することはできないが，今回の場合は従来の 14 ～ 25% の開発量であった．以上のことから，システムの汎用化は達成され，開発者の負担が軽減されたといえる．

3.7 力覚提示性能の評価

本章で提案した高解像度力覚インタフェースが，精細な形状や硬い物体を安定に提示できることを確認するため 2 つの実験を行った．

3.7.1 剛性の高い面の提示

構築したシステムが，従来の力覚インタフェースに比べて硬い物体を提示できることを確かめるため，制御周期を 1kHz, 5kHz, 10kHz とした場合について次の実験を行った．

様々なダンパ定数について水平面を提示し，被験者に上から触れさせた．被験者は右手でマウスを操作し，提示面のバネ係数をスライダーで調整しながら，左手で力覚インタフェースを介して提示面に触れた．被験者には，平らな面に感じられる最大のバネ定数を探すよう指示した．

被験者が選んだバネ定数が最大となったときのバネ定数とダンパ定数を表 3.4 に示す．10kHz のときのバネ定数 28.5kN/m は，Sensible Technologies 社の製造する力覚インタフェース PHANTOM Premium 1.0 [31] が提示可能な硬さ 3.5kN/mm の 8 倍にあたる．

表 3.4: バネ定数の最大値

制御周期	バネ係数 K	ダンパ係数 B
1kHz	7.0kN/m	10N/ms ⁻¹
5kHz	25.5kN/m	75N/ms ⁻¹
10kHz	28.5kN/m	100N/ms ⁻¹

SPIDAR の機構のバネ定数が 20kN/m であり，指先に提示されるバネ定数は機構と制御によるバネを直列につないだものになると考えられるので，指先には約 12kN/m の硬さが提示されたと考えられる．

3.7.2 精細な形状の提示

提案システムがより詳細な形状を提示できることを確かめるため，物体形状をなぞった際の力覚ポイントの軌跡を計測する実験を行った．実験では，提案システムと制御周期の低速な従来システムで同じ形状（歯車）を映像と力覚で提示し，被験者に形状をなぞるように指示した．被験者には，抗力のほかに摩擦力（ $\mu = \mu_0 = 0.5$ ）を提示した．このときの，力覚ポイントの軌跡を図 3.24 に示す．

力覚インタフェースで摩擦力を提示すると面の向きの判別が難しくなるため，形状を把握することが難しくなる．このため従来システムでは，軌跡と物体形状が一致しておらず，被験者は詳細な形状を把握できていない．一方，提案システムでは形状と軌跡がほぼ一致しており，摩擦力の提示と詳細な形状の提示が両立していることが分かる．

3.8 この章のまとめ

本章では，高解像度力覚インタフェースの機構と制御方法を提案した．高解像度力覚インタフェースは応答の速い機構と制御を必要とするため，機構には糸駆動型力覚インタフェースを用い，電流制御回路で駆動された DC コアレスモータによって駆動した．

また，糸駆動型力覚インタフェースの姿勢・力覚計算を行う計算法を提案し汎用化した．力覚計算においては，二次計画法を用いて最適化問題を解く問題に置き換える方法を採用し，操作の安定性を考慮しつつ汎用性を高めた．姿勢計算においては，繰り返し計算を行い近似解を算出することで，機構の変更に対応できる上，比較的精度のよい姿勢情報が高速に得られるようになった．

さらに，提案した力覚インタフェースと姿勢と力覚の計算法を用いて，高解像度力覚提示が実現できていることを確認した．

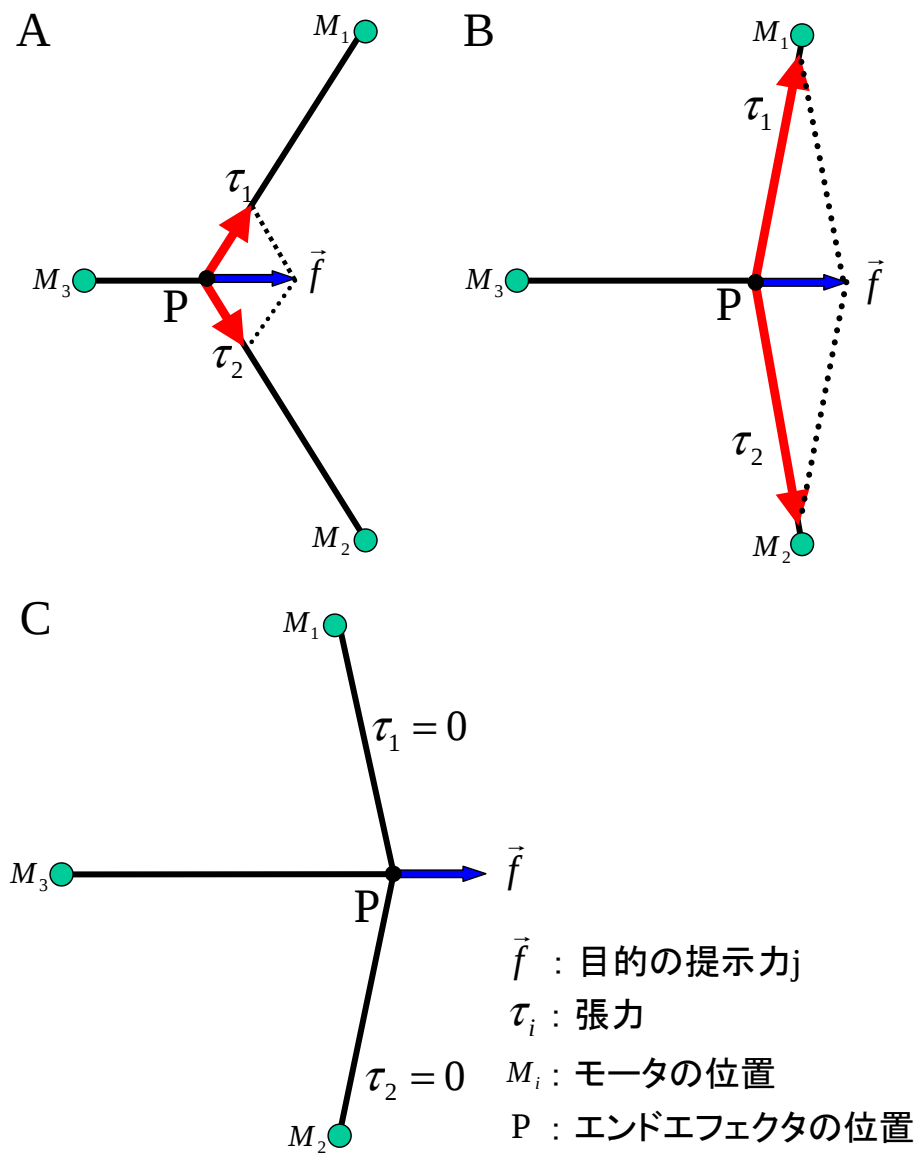


図 3.14: グリップ位置と張力

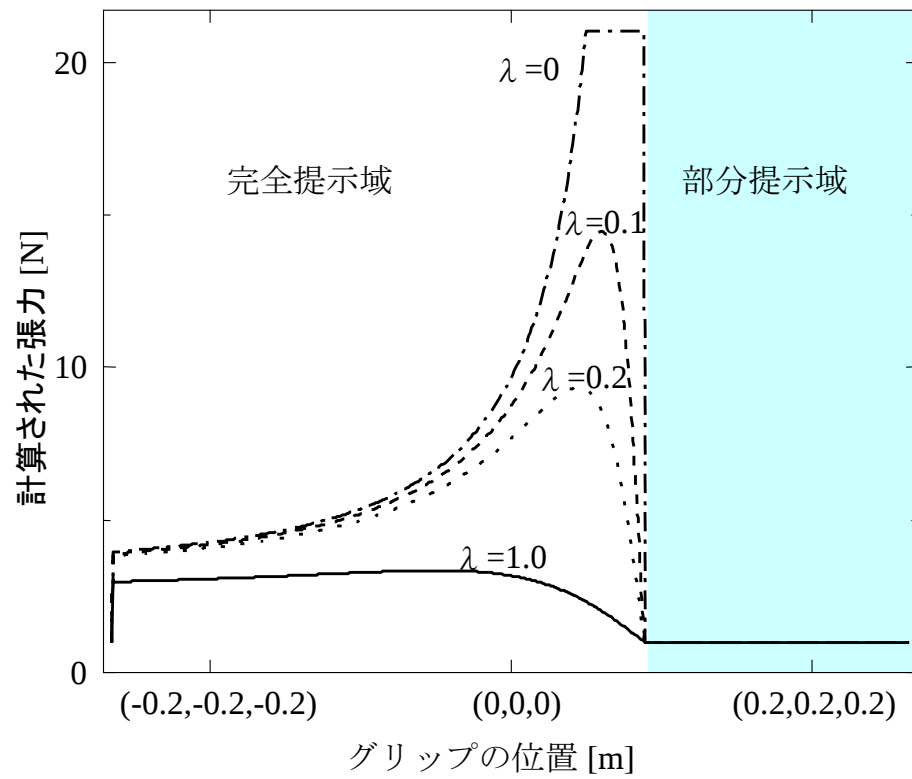
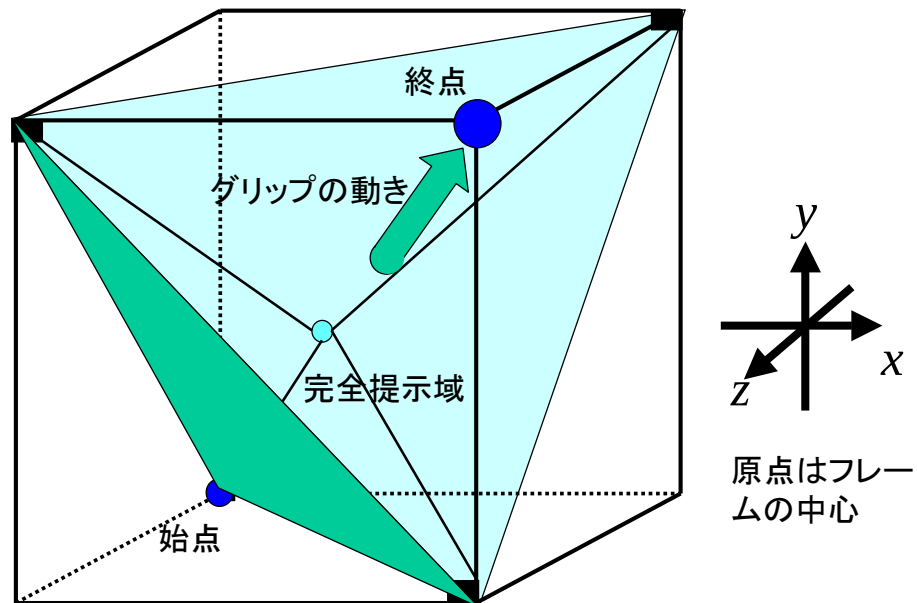
図 3.15: 連続性の係数 λ と計算された張力 τ の関係

図 3.16: 実験 1 の実験条件

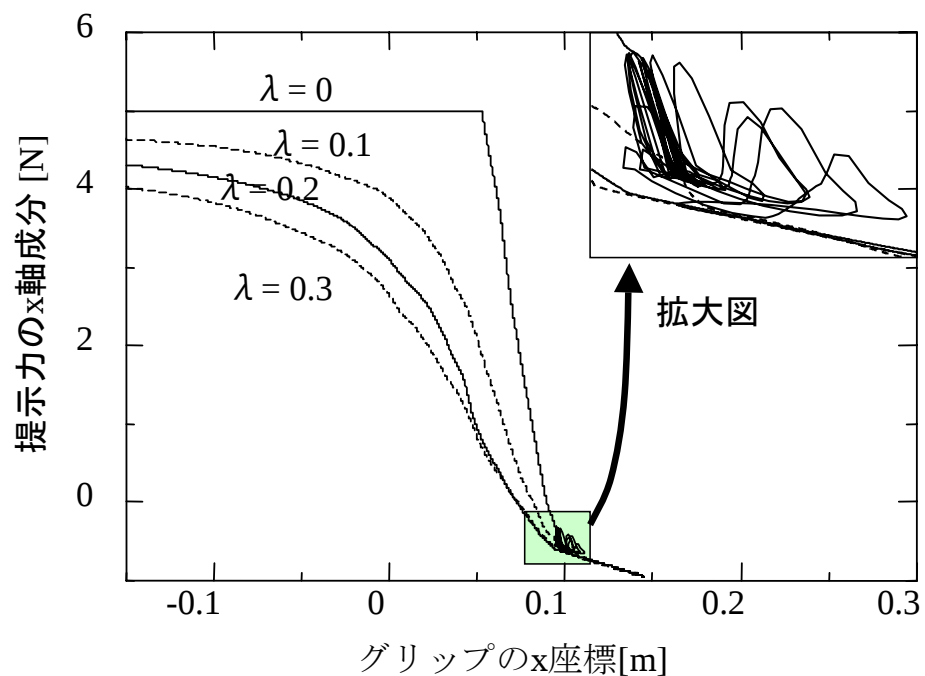


図 3.17: 実験 1 の結果

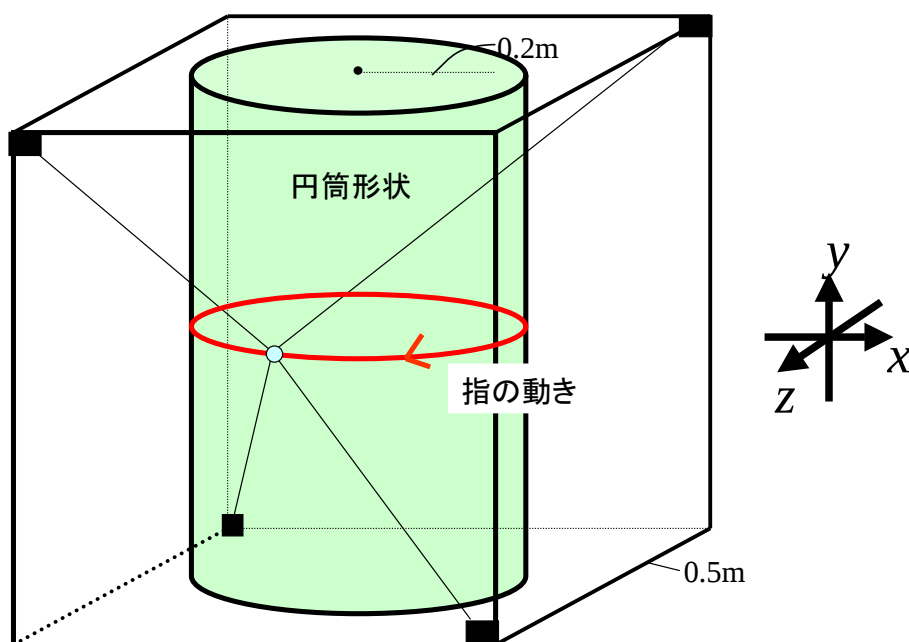


図 3.18: 実験 2 : 円筒形状の提示

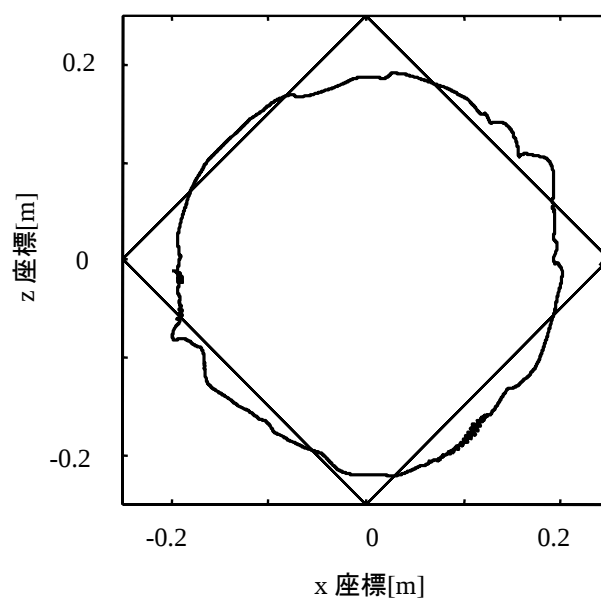
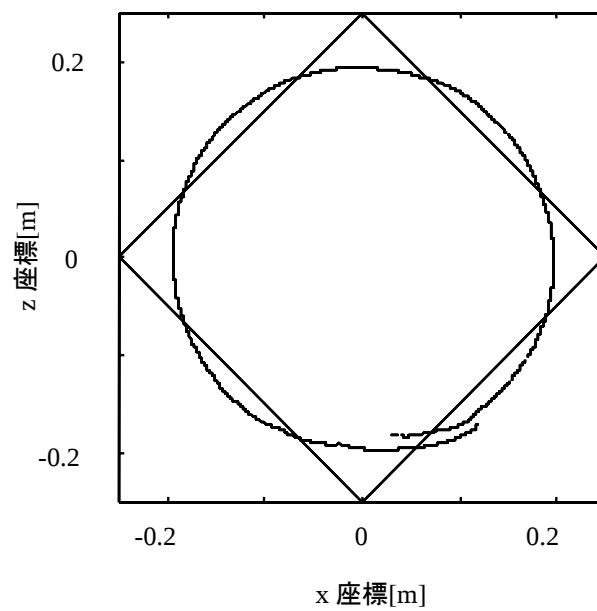
(A) 正確性のみを考慮した場合 ($\lambda=0$)(B) 連続性を考慮した場合 ($\lambda=0.2$)

図 3.19: 指先の軌跡

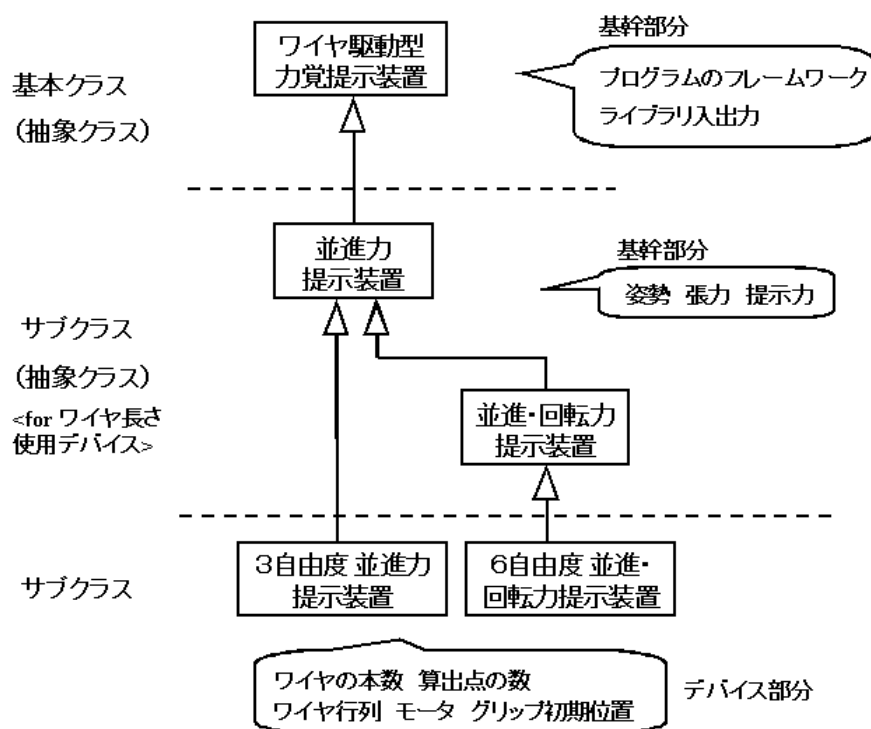


図 3.20: 定義したクラス階層図

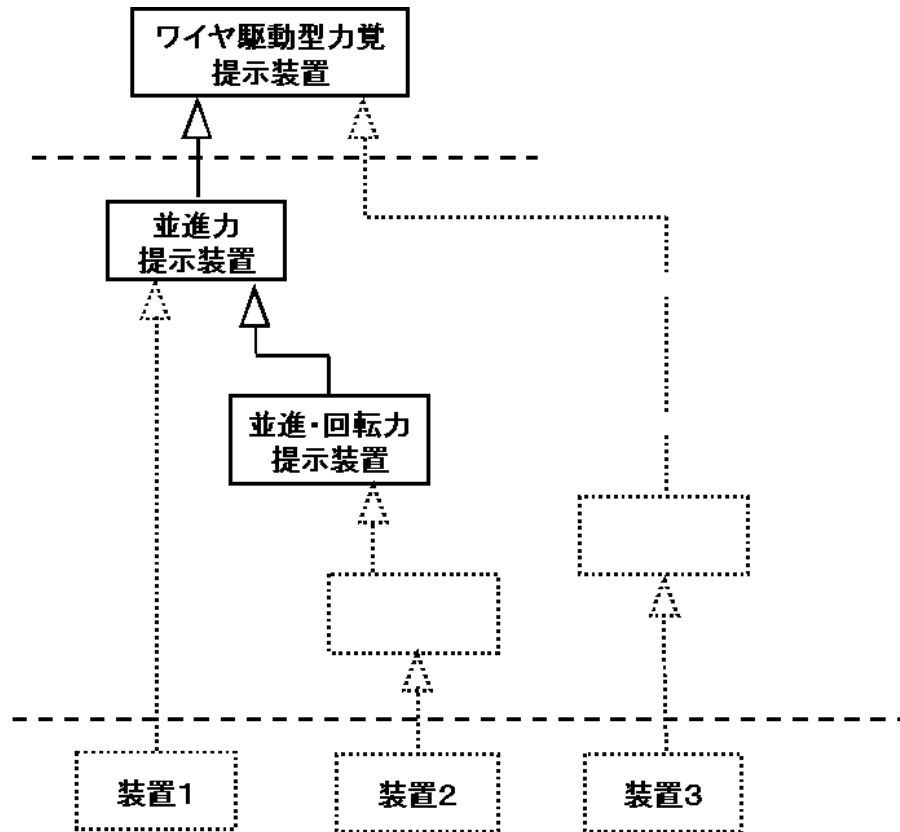


図 3.21: 新しい装置のためのクラス定義

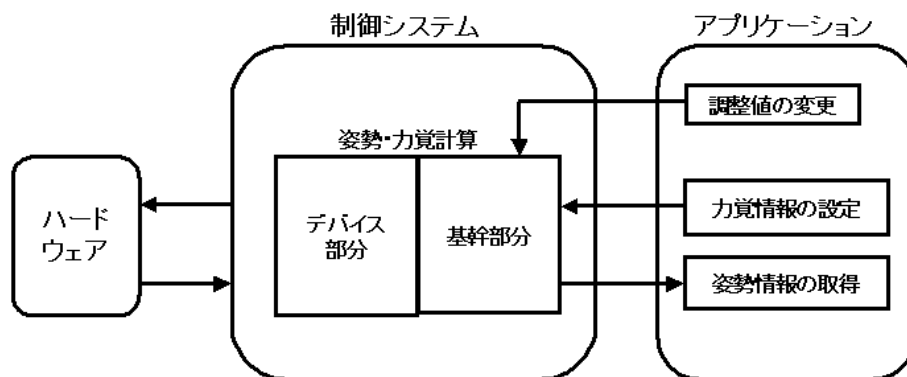


図 3.22: 機構，システム，アプリケーションの関係

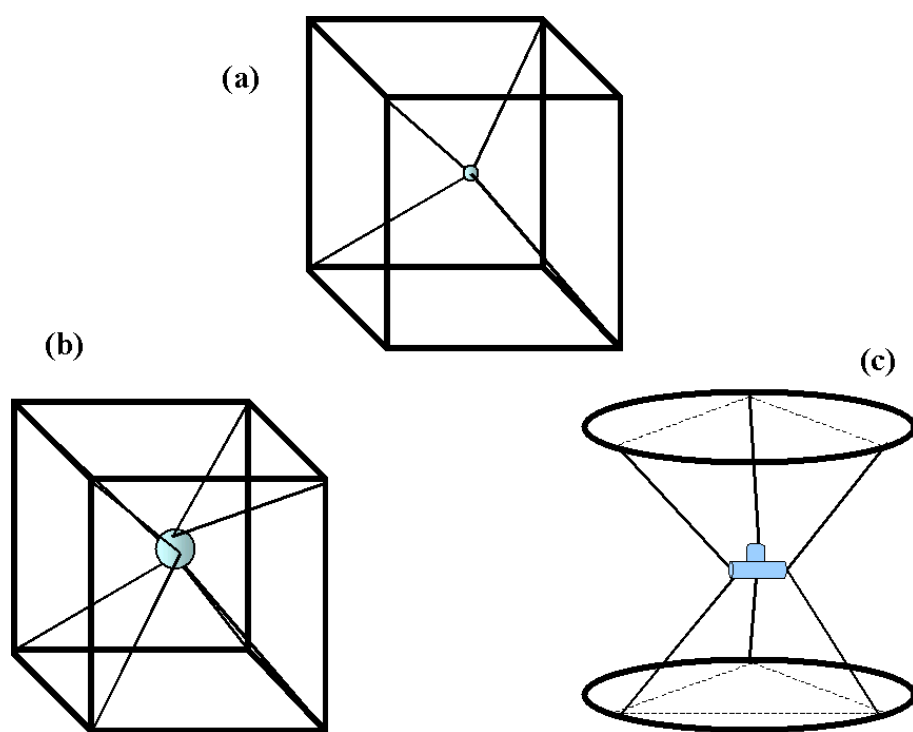


図 3.23: 想定した力覚インタフェースの機構

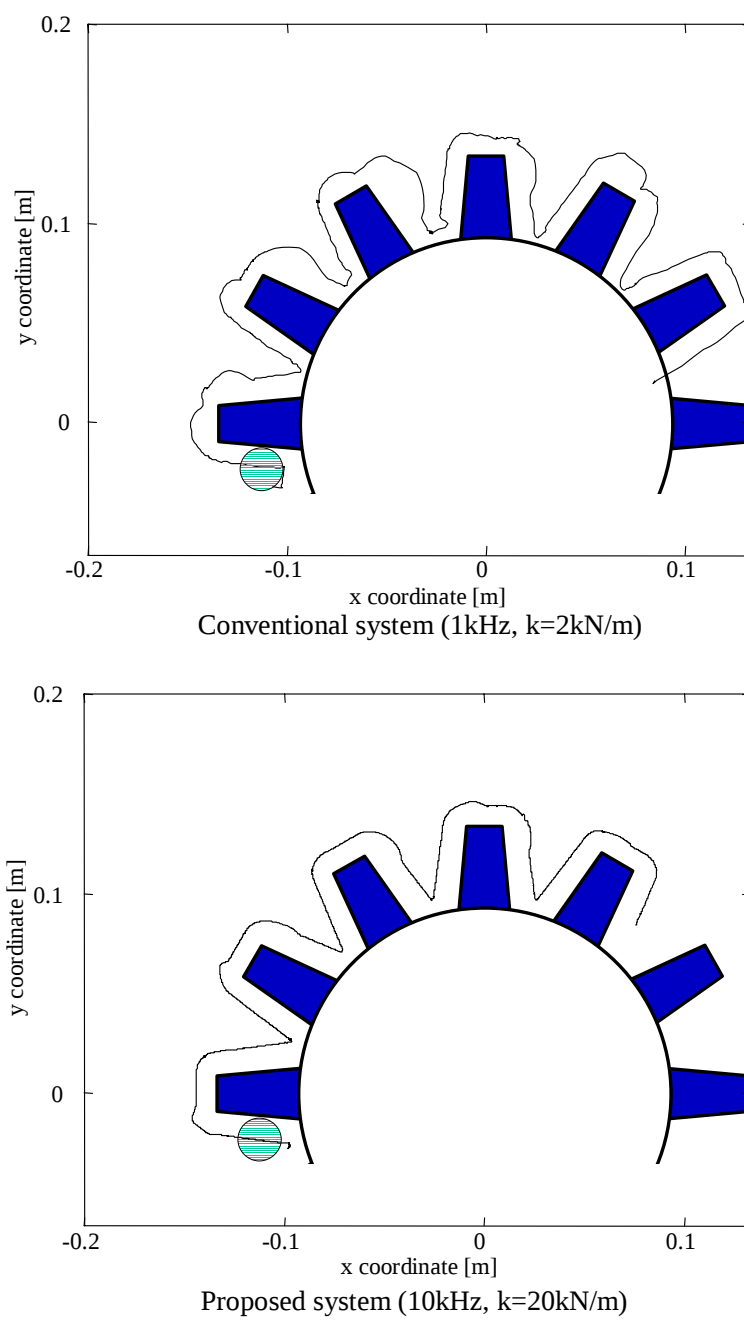


図 3.24: 提案システムと従来システムでの力覚ポイントの軌跡

第4章 力覚レンダリング

バーチャル世界の物体モデルに力覚ポインタが触れたときに、力覚インタフェースが提示すべき力を求め、物体モデルの存在をユーザに提示することを力覚レンダリングと呼ぶ。この章では、これまでに提案された物体形状を提示するための力覚レンダリングの手法を概説した後、実物体を手で操作する場合の基本的な操作である把持を安定に実現する方法を提案する。安定に把持を行うためには、指間の距離を一定に保つ必要があり、提案手法はこれを実現する。

4.1 物体形状の提示

実世界では、物体と手の間には抗力が働き、手が物体の中に侵入することはない。力覚インタフェースで抗力を提示できれば現実の物体に触れたのと同じ感覚が提示できる。手が力覚インタフェースに加えた力が分かれば、これと等しい力を提示すれば抗力の提示ができるが、インピーダンス提示型の力覚インタフェースは力を計測することができない。そこで、ユーザの手の位置に応じた力を提示することで抗力に近い力を提示することを考える。

抗力は、手が物体に侵入しないように保つ力なので、力覚ポインタが物体モデルに侵入した場合に、物体の外に手を追い出す向きに力を加えることで同じ働きをする力が提示できる。

物体内部に入ったとたんに強い力を加えてしまうと、物体表面で力覚ポインタが突然はじかれることになり、接触を維持することができなくなる(図 4.1-A)。そこで、物体への侵入距離に比例した力を提示すると、比例係数が小さければ安定な接触が提示できる(図 4.1-B)。

しかし、この方法にもまだ問題がある。図 4.1-B の軌跡のように、物体を力覚ポインタで強く抑えると途中で提示力の向きが不自然に変化する。そこで図 4.1-C のように、質量を持たず、物体に侵入せず、できるだけ力覚インタフェースのグリップ位置に近づこうとする力覚ポインタ (god object [48] または proxy [49] と呼ぶ) を考え、これとグリップの位置の間にバネモデルを考える god object 法 [48] が提案されている。また、このバネモデルをバネダンパモデルにすることにより、より硬い材質感と安定な提示ができる(図 4.1-D)。

4.1.1 god object 法の詳細

指が物体に触れたり、指で物体をなぞったりする際には、物体と指の間に抗力と摩擦力が生じる。God object 法は、力覚ポインタをクーロンの摩擦モデルに基づいて移動させる

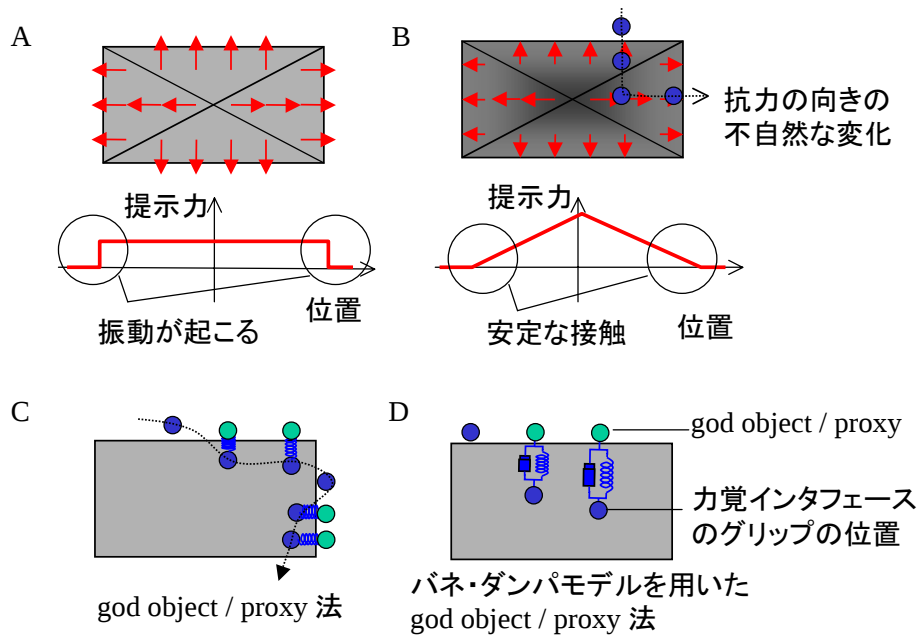


図 4.1: 物体形状の提示法

ことで、抗力と摩擦力の両方を提示することができる。

力覚インタフェースのグリップが物体モデルに侵入していないときは、力覚ポインタはグリップの位置に移動する。グリップが物体モデルに侵入したときは、力覚ポインタは、物体に侵入しない、静止摩擦の範囲では、滑らないといった拘束を満たしながら、グリップに近づく。以下に、力覚ポインタの移動の手順を示す。また、図 4.2 に力覚ポインタの移動の一例を示す。

1. グリップの位置を計測する。
2. グリップとポインタの間のバネモデルが発生する力を求める。
3. ポインタに働く抗力、摩擦力を求める。
4. ポインタ働く合力が 0 でない場合、合力の向きを求める。合力が 0 のときは手続きを終える。
5. ポインタを合力の向きに動かし、接触状態が変わる (接触が消える。新しい接触が生まれる) 位置を調べる。
6. ポインタを合力の向きに動かしたとき、合力が 0 になる位置を求める。
7. 5 6 で求めた位置のうち、現在の位置に近いほうにポインタを移動する。
8. 移動先が、5 の場合は、2 から繰り返す。
9. 移動先が、6 の場合は、手続きを終える。

なお，摩擦力は，手順 3 では，前回ポインタが移動した場合は動摩擦，移動しなかった場合は，静止摩擦として摩擦力を計算する．また，手順 6 は，ポインタが移動してからの停止条件なので，動摩擦として摩擦力を計算する．

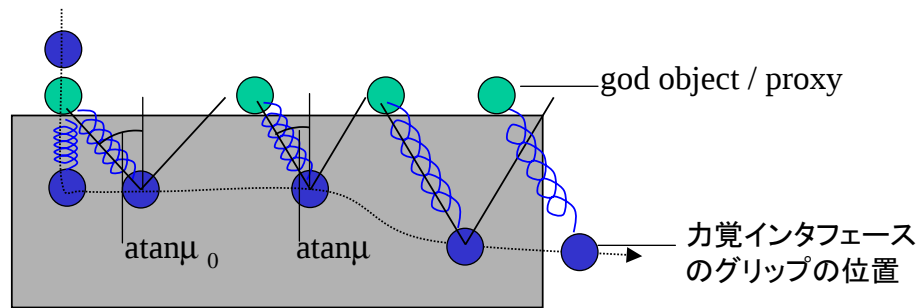


図 4.2: god object 法での力覚ポインタの動きの例

4.1.2 バーチャルカップリング法

これまでに説明した手法は，力覚ポインタが質量を持たないことを前提としている．この考え方では，力覚ポインタは実世界の手のバーチャル世界での表現であり，手の位置が力覚ポインタに反映され，力覚ポインタが受けた力が手に提示される．力覚ポインタには質量がないため，力覚ポインタ自体は重量や慣性力を発生せず，力覚ポインタの位置に手があった場合に手に加わる力が提示されることになる．

一方，バーチャルカップリング法では，力覚ポインタが質量や慣性を持ち，他の物体モデルと同様にシミュレーションの対象となる．力覚ポインタが質量を持つと，力覚ポインタ自体の重量や慣性力が提示力に含まれることになり，力覚ポインタという道具を手にとって，力覚ポインタを介してバーチャル世界の物体に触れていることになる．

図 4.3 にバーチャルカップリング法と god object 法の違いを示す．

バーチャルカップリング法では，物体モデルの運動シミュレーションの手法により，提示可能な力の種類が決まる．また，任意の物体モデルを力覚ポインタとして使用できるので，シミュレータが扱えれば，任意の形状の力覚ポインタが使用できる．さらに，バーチャルカップリングに用いるバネ・ダンパモデルに回転のバネダンパを追加することで，並進・回転 6 自由度の提示力を算出できる．

4.2 多指力覚インタフェースによる安定な把持の実現

3 次元物体を操作する場合，人は複数の指によって物体を把持する．力覚インタフェースを用いて物体モデルを操作する場合，ボタン操作などによって把持モードに切り替え，把持対象と物体モデルの間をバーチャルカップリングで接続する方法もあるが，自然な操作を実現するためには，複数の指を用いて直接物体モデルを把持操作できることが望ましい．把持操作を行うためには，物体の形状と物体が指に与える外力を提示する必要がある．

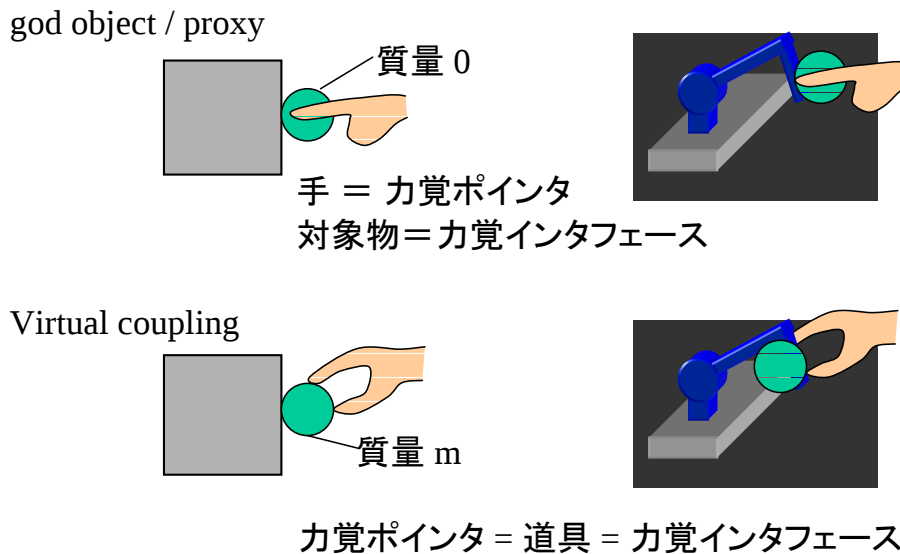


図 4.3: パーチャルカップリングと god object 法

多指による把持の提示については、Ishii と Sato [29] の報告がある。しかし、これは物体形状と物体が床面から受ける力の提示にとどまっており、慣性力など物理モデルを用いた場合に物体に作用する力を考慮していない。

Maekawa と Hollerbach [38] は、親指と人差し指による把持操作が可能な力覚インタフェースとその制御方法を提案しているが、接触モードと把持モードを分けており、拡張性に乏しい。Yoshikawa と Ueda [39] は、モード切り替えなしで外力と把持力を提示する方法を提案している。この手法は2本の指に対してそれぞれバネダンパモデルによる力を適用する方法であり、外力提示と把持提示が分離されていないため、外力提示が把持力や指間距離を変動させるといった問題がある。

川井と吉川 [40] は、内力(把持力)と外力それぞれに介在インピーダンスを設定することで、より硬い物体の把持を安定に提示する方法を提案している。しかしこの方法は、外力を2本の指に等分に分配しているため、外力提示が把持力に影響を与えるという問題は解決されない。

本章では、従来の提示方法の問題点を指摘し、外力提示が把持力提示に影響を与えないモデルを提案することで、物体を動かす際の操作感を向上させる。また、考案したモデルを拡張し、モード切り替えがなく、多指への適応が可能な、物体形状と外力の提示方法を示す。

4.3 実世界での把持

物をつかんで動かすという操作は、多指直接操作環境において多くのアプリケーションに必要な基本的な操作だと考えられる。そこで、物を水平に動かす際に生じる加減速が把持に及ぼす影響を端的に捕らえるため、親指と人差し指による水平1自由度の把持を考え、物体を水平に揺らすという作業を取り上げる。

4.3.1 把持のモデル

実世界で剛体を把持する場合，親指と人差し指の距離は把持対象の形状によって決まり，指が出す力によらず一定値となる．このときの剛体の運動方程式は，指が剛体に与える力を f_1, f_2 ，剛体の中心位置を x とすれば (図 4.4)，

$$m\ddot{x} = f_1 + f_2 \quad (4.1)$$

となる．指が物体に与える合力 (=物体が指に与える外力の反作用) は $f_1 + f_2$ だが，指が発生させる力の大きさの和 $|f_1| + |f_2|$ は，合力よりも大きくなる．この合力に影響を与えない力は，物体がすべり落ちることの無いように物体-指間の抗力を保ち，把持を持続させる．そこでこの力のことを把持力と呼ぶ．把持力は，

$$f_g = \begin{cases} |f_2| & (|f_1| > |f_2| \text{ のとき}) \\ |f_1| & (\text{それ以外のとき}) \end{cases} \quad (4.2)$$

合力は，

$$f_r = f_1 + f_2 \quad (4.3)$$

となる．以上から，ある合力 (f_r) を発生させる各指の力 (f_1, f_2) は，把持力 (f_g) の大きさ

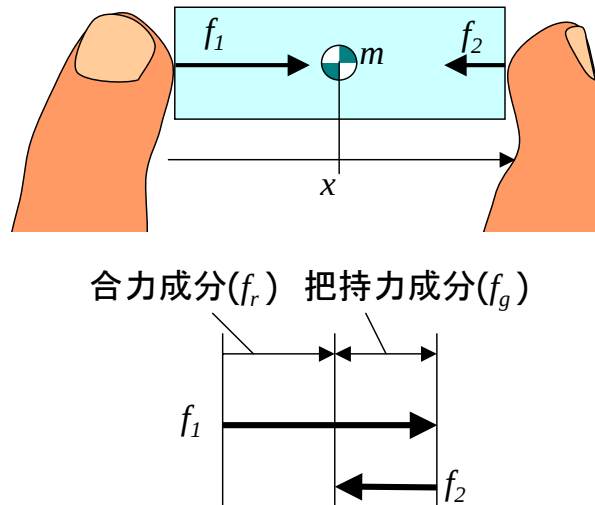


図 4.4: 1 自由度の剛体の把持と把持力

によっていろいろな値をとり得ることがわかる．

4.3.2 把持力の計測

合力と把持力の関係がわかれば，力覚インタフェースを用いて把持操作環境を構築する際に，合力に応じて把持力の大きさを制御することができるので，操作感の向上が期待できる．そこで物体を把持操作する際に人がどのように把持力を制御しているのかを計測した．

4.3.3 計測方法

質量 1kg・重量 0.2kg の把持対象を左右に振るときに各指が物体に与える力を計測した．把持対象の自由度のうち回転と奥行き方向を拘束し，上下左右方向だけに動くようにした．力センサの質量による慣性力が計測結果に影響を与えることを防ぐため，計測にはシート状の力センサ，FlexiForce¹を使用した．このセンサには出力特性にばらつきと非線形性があるので，校正済みの力センサを用いて特性を計測し，計測結果を用いて出力値を補正した．

物体の位置計測には光学式位置計測装置，OPTOTRAK²を用いた．

被験者には，物体を左右に 5cm 程度 8 回振ったあと，2 秒間静止させる作業を繰り返すように指示した．実験は 4 人の被験者に対して行った．図 4.5 に計測の様子を示す．

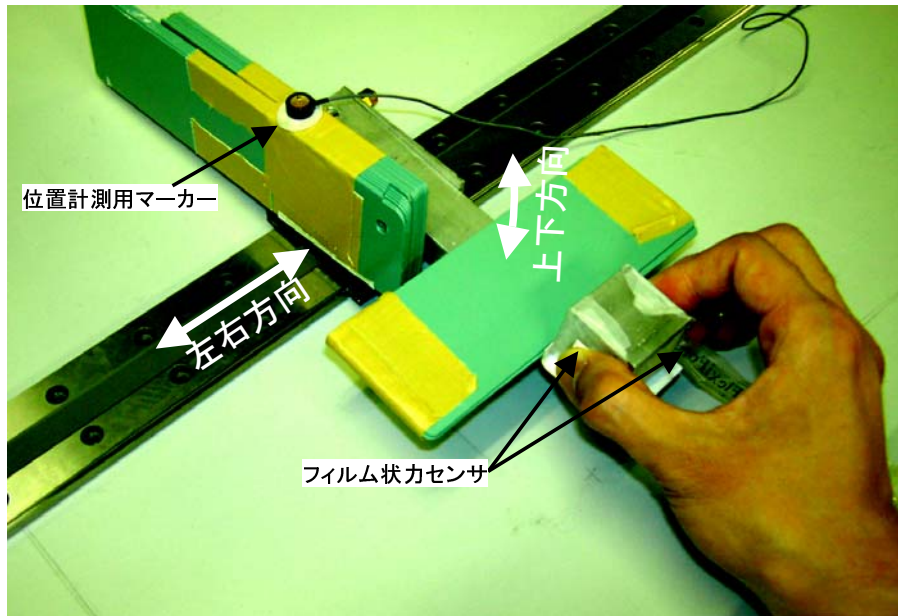


図 4.5: 実物体把持時に指が加える力の計測

4.3.4 計測結果

図 4.6 に計測結果の一例を示す．グラフから，物体を振っているときに指が物体に与える力は，静止時の力を下回らないことが分かる．これは，物体に加える力を大きくすることで，物体に合力を与えていることを示している．他の 3 人の被験者についても同様な結果を得た．

以上から，人は把持力が保たれるよう，指が物体に加える力を大きくすることで物体に与える合力を発生させていることがわかった．

¹<http://www.tekscan.com/flexiforce/flexiforce.html>

²<http://www.ndigital.com/optotrak.html>

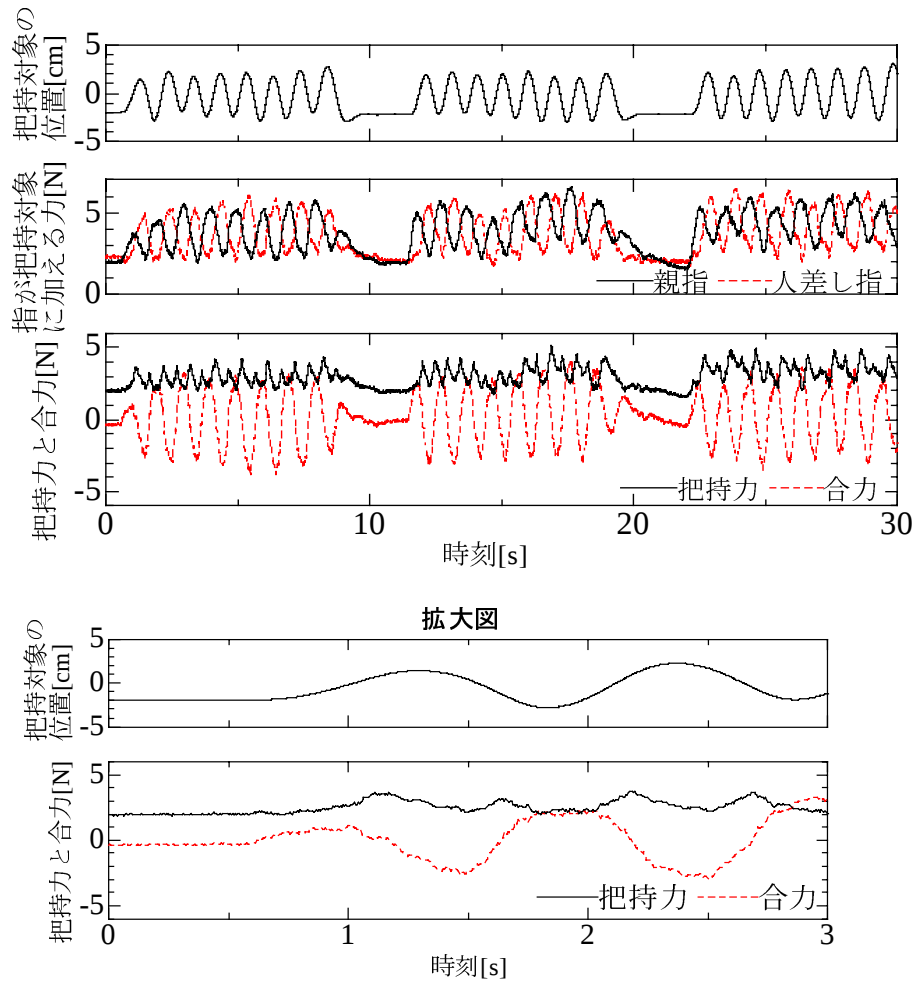


図 4.6: 実物体把持時に指が加える力

4.4 力覚インタフェースによる把持提示の問題点

位置を計測し力を出力するタイプの力覚インタフェースで物体の形状を提示する場合，物体と指の間にばねモデルを考えることで提示力を求める．従来は，把持対象の物体モデルに質量と自由度を設定し，物体モデルの運動をシミュレートすることで，把持操作可能な環境を構築していた [39]．図 4.7 に従来方式での把持提示方法を示す．

このときの物体モデルの運動方程式は，

$$m\ddot{x} = f_1 + f_2 \quad (4.4)$$

$$f_1 = k(x_1 - (x - l/2)) \quad (4.5)$$

$$f_2 = k(x_2 - (x + l/2)) \quad (4.6)$$

となる．

$|f_1| > |f_2|, f_1 > 0, f_2 < 0$ のとき，指が物体に与える力 f_1, f_2 を合力 f_r と把持力 f_g で書き直すと，

$$f_1 = f_g + f_r, f_2 = -f_g \quad (4.7)$$

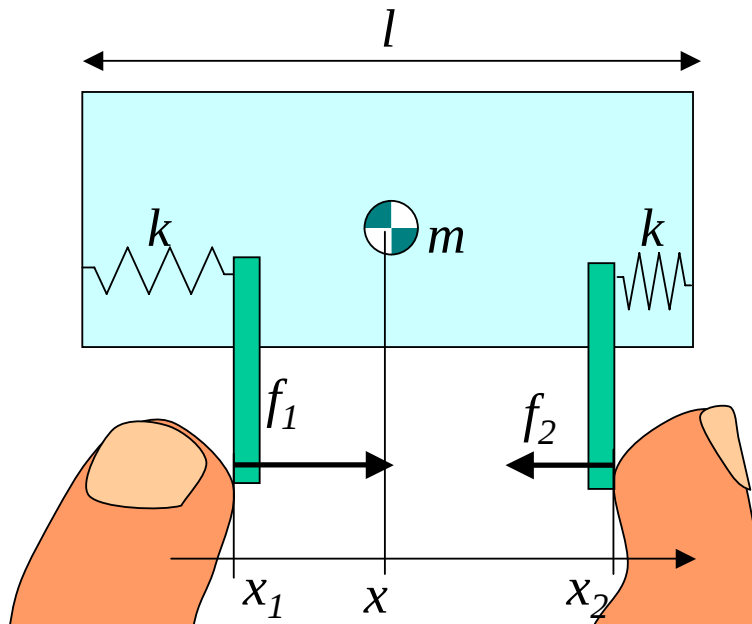


図 4.7: 従来の把持提示方法

なので，親指と人差し指の指間距離は，

$$x_2 - x_1 = l - \frac{2f_g + f_r}{k} \quad (4.8)$$

と表せる．式 (4.8) は，合力 (f_r) が増えると把持力 (f_g) が指間距離 ($x_2 - x_1$) のどちらかが減ることを示している．4.3.2 節の計測結果から，物体に合力が加わる際も把持力 (f_g) は保たれると言えるので，合力が加わると指間距離が減少すると考えられる．

このように，従来方式では物体を加速した場合など指が物体に合力を加えた場合に指間距離が減少する．このため，物体を加速する際に指間距離と把持力が減少し，把持の安定性が失われるという問題が生じる．

4.5 合力分配方式

指が物体に合力を加える際に指間距離が変化するのは，バネモデルに合力がかかりバネ長が変化するためである．合力を指から物体に直接加え，形状提示のためのバネモデルに把持力だけがかかるようにすれば，指間距離の変化を避けることができる．

4.5.1 把持力の計算

従来方式では，物体に加える合力と把持力を1つのバネモデルから求めていた．形状提示のためのバネモデルに把持力だけがかかるようにするためには，把持力と合力を独立に求める必要がある．

把持力と合力を独立に求めるために，把持対象物体を形状と動力的性質に分けて考える．把持力は，把持対象物体と同じ形状で質量が0の仮想の物体を考えることで計算する．

この仮想の物体は，把持対象の形状を表現しているが物体の中身の性質を持たないので，ここでは形状殻と呼ぶ．

指が形状殻をバネモデルを介して把持していると考えれば，指位置とバネモデルから形状殻のつりあいの位置を求めることができる．このときに各指にかかる力を把持力とする．

4.5.2 合力の計算

物体の運動方程式は位置を x ，指が物体に与える合力を f_r ，物体に働くその他の外力を f_e とすれば，

$$m\ddot{x} = f_r + f_e \quad (4.9)$$

なので，指が物体に与える合力は $m\ddot{x} - f_e$ である．

形状殻の位置は力覚インタフェースの位置から求めたつりあいの位置となるので，その加速度も，力覚インタフェースの加速度から求まるはずである．しかし力覚インタフェースの多くは加速度を計測できないので，形状殻の加速度を求めることはできない．

そこで物体と形状殻の間にバネモデルを考え，バネが発生する力を物体の慣性力として $m\ddot{x}$ の代わりに用いる．これにより，力覚インタフェースの加速度を計測せずに慣性力を求めることができる．

4.5.3 合力の分配

求めた合力は把持力計算のためのバネモデルを通さずに，直接各指に加えなければならない．

各指にかかる把持力を減らす向きに外力を分配すると把持力が減少してしまうので，把持力と外力が同じ向きとなる指だけに外力を分配する．

以上をまとめると図 4.8 になる．また，このときの親指と人差し指の指間距離が，

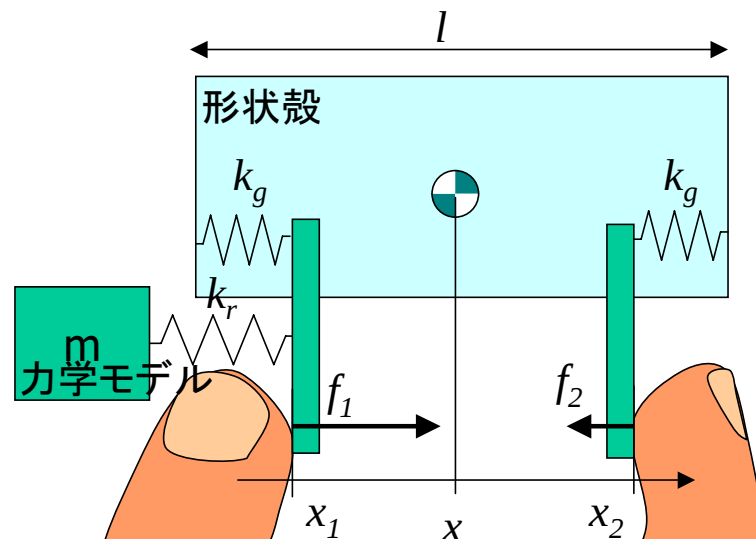


図 4.8: 合力分配方式による把持提示

$$x_2 - x_1 = l - \frac{2f_g}{k} \quad (4.10)$$

となることから，合力の形状提示への影響がなくなると考えられる．

4.6 多指6自由度物体の把持への拡張

4.5 節で提案した合力分配方式を，多指による6自由度物体の把持の提示に適用するために拡張する．

4.6.1 把持力の計算

バネモデルから形状殻のつりあいの位置を求める計算は， n 本の指による把持の場合には次のように表せる．

仮想世界での指位置 (指の god object の位置) を r_i ，指が形状殻に与える力を f_i ，バネモデルの係数を k_i とすると (図 4.9)，形状殻に指が与える外力の合力 F と合モーメント N

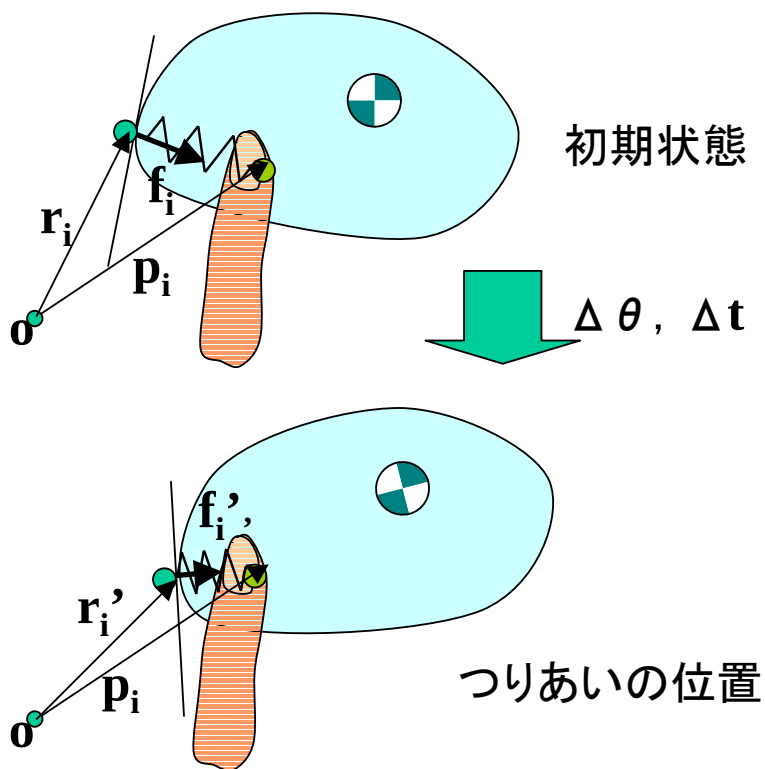


図 4.9: 形状殻のつりあいの位置

は，

$$F = \sum f_i \quad (4.11)$$

$$N = \sum r_i \times f_i \quad (4.12)$$

となる．つりあいの位置での指の god object の位置を r'_i , 指が形状殻に与える力を f'_i とし, 形状殻を Δt 移動し $\Delta\theta$ 回転させたときに, バネがつりあったとすれば (図 4.9), つりあいの位置では, 合力・合モーメントは 0 なので,

$$0 = \sum f'_i \quad (4.13)$$

$$0 = \sum r'_i \times f'_i \quad (4.14)$$

となる． $\Delta\theta$ が小さいので,

$$r'_i \approx r_i + \Delta\theta \times r_i + \Delta t \quad (4.15)$$

と近似でき,

$$f'_i = k(p_i - r'_i) \quad (4.16)$$

なので, 式 (4.13), 式 (4.14) を Δt , $\Delta\theta$ について解くことで, 形状殻のつりあいの位置を求めることができる．そこで, 形状殻をつりあいの位置に移動し, 各指の god object の位置も形状殻に合わせて移動し, このときの god object と各指を結ぶバネダンパモデルが各指に与える力を把持力とする．

4.6.2 合力の計算

6 自由度物体の場合も, 指が物体に加える合力は, 物体の慣性力と指以外から物体に加わる外力の和になる．6 自由度物体の慣性力を求めるために, 形状殻と物体の間に並進 3 自由度, 回転 3 自由度のバネダンパモデルを考え, バネダンパモデルが形状殻に与える力を慣性力とする．これと, 外力をあわせた力が形状殻に加わる合力となる．

4.6.3 合力の分配

形状殻に加わった合力をユーザの指に提示する際に, 各指に加える合力の配分を人の把持特性に合わせて調節することで安定な把持を提示することが 4.5.3 節の提案であった．そこで, 多指による 6 自由度把持についても, 同様に各指に加える合力の配分を調節する．合力の分配に求められる条件として次のことが挙げられる．

1. 分配された合力の和がもとの値に近くなる．
2. 把持力に応じた合力を分配する．
3. 指にかかる力が連続に変化する．
4. 指に把持力を減らす向きの力を加えない．

以上の条件は、各指が形状殻と接している点を r_i 、その点での接面の法線ベクトルを n_i 、各指が形状殻に加える把持力を f_{gi} 、各指に配分される合力を $k_i n_i$ 、指が物体に加える合力の力とモーメントを F, N とすれば、次のように定式化できる。

$$P = (F - \sum k_i n_i)^2 + \gamma_1 (N - r_i \times \sum k_i n_i)^2 + \gamma_2 (\sum (k_i - k |f_{gi}|)^2) + \gamma_3 (\sum k_i^2 + k^2)$$

として、 P を最小化する $k_i, k (k_i, k \geq 0)$ を求める。 (4.17)

この問題は2次計画問題なので解くことができる [41]。ただし、 $\gamma_1 \sim \gamma_3$ は各項の重みづけであり、上で挙げた4条件の重みを調整するためのパラメータである。ここで得られる分配された合力の和は、最適化問題の解なので、もとの合力と等しくならない。そこで残差を形状殻に加え、把持力を4.6.1節の方法で再計算することで、合力全体が指に伝わるようにする。

4.7 合力分配方式の評価実験

並進3自由度を持つ剛体の親指と人差し指による把持を力覚インタフェース (SPIDAR-II[29]) を用いて提示し、提案方法と従来方法を比較した。

4.7.1 実験条件

提示する物体は、幅6cm、高さ10cmの直方体とし、質量を1kgとした。重力加速度の大きさは、実世界より小さい $2.0m/s^2$ とした。これは、力覚インタフェースの提示力の制約により、実世界と同じ重力を提示することができなかったためである。

把持力計算のためのバネモデルと慣性力計算のためのバネモデルのバネ係数を共に800N/mとした。また、物体と指の間には摩擦力が働くものとし、摩擦係数を1.0とした。

被験者には、各指の仮想世界での位置、把持物体を示す映像を提示した。実験の様子を図4.10に示す。被験者には4.3.2節の計測と同様に、物体を左右に5cm程度8回振ったあと、2秒間静止させる作業を繰り返すように指示した。5人の被験者に対して実験を行った。

4.7.2 実験結果

実験結果の1つを図4.11と図4.12に示す。また、表4.1に把持力、表4.2に指間距離の時刻0秒から30秒までの平均と標準偏差を示す。表4.1、表4.2の標準偏差と図4.11から、提案方式では従来方式よりも把持力・指間距離とも変動が小さくなることが分かる。このことから、提案方式はより現実近く、安定した把持を提示したと言える。

拡大図 (図4.12) をみると、従来方式では合力の大きさが大きい時刻 (図の網掛け部分) に把持力・指間距離が減少するという問題が見られるが、提案方式では把持力の減少が小さくなり指間距離は増加している。これは、従来方式の把持力・指間距離と合力の関係式 (式(4.8)) が示す、合力の提示が把持力・指間距離の減少を引き起こすという仮説と合致す

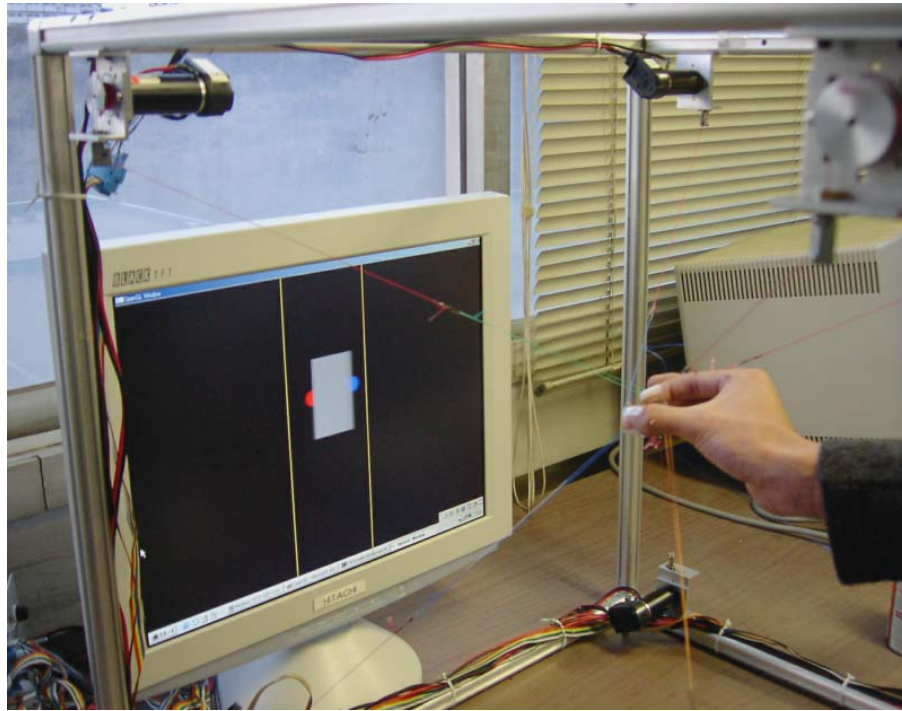


図 4.10: 比較実験の様子

被験者	A	B	C	D	E
平均 (従来)[N]	2.0	2.0	1.7	1.8	4.1
平均 (提案)[N]	2.2	2.4	1.6	1.6	4.7
偏差 (従来)[N]	0.39	0.35	0.28	0.35	0.77
偏差 (提案)[N]	0.34	0.29	0.21	0.22	0.35

表 4.1: 把持力の平均値と標準偏差

る．提案方式の指間距離の式 (式 (4.10)) は，指間距離と把持力は合力に拠らないことを意味するが，実際には，合力が加わると把持力がわずかに減少し，その分指間距離が増加するという結果になった．

次に図 4.11 の指間距離を見ると，従来方式では把持対象が運動する期間 (網掛け部分) の指間距離が静止期間の指間距離に比べて極端に小さくなっている．これは，運動期間に把持力が把持の保持に必要な値を下回らぬよう，被験者が指間距離を小さく保つためだと考えられる．一方，提案方式では従来方式のような運動期間の指間距離の減少は見られない．

実験終了後，被験者に内観を聞いたところ「提案方式の方が物体が硬く感じられる」「提案方式の方が物体が重く感じられる」「提案方式の方が実物体を揺らした感覚にちかい」といった回答のほか「違いが分からなかった」といった回答も得た！違いが分からなかった」と答えた被験者でも指の力や指間距離の変化などの客観的なデータには図 4.11 と同様の傾向が見られた．

被験者	A	B	C	D	E
平均 (従来)[cm]	5.0	5.4	5.5	5.4	4.8
平均 (提案)[cm]	5.4	5.4	5.6	5.6	4.8
偏差 (従来)[cm]	0.10	0.089	0.076	0.074	0.16
偏差 (提案)[cm]	0.085	0.073	0.054	0.056	0.087

表 4.2: 指間距離の平均値と標準偏差

4.8 考察

力覚インタフェースには表現できるバネモデルのバネ係数に上限がある．そのため，合力提示を行う際に，指間距離や把持力の変動といった問題が発生する．

実世界でもやわらかい物体を把持操作すれば同じ現象が生じると考えられるが，実世界の把持操作対象の多くは硬いため，実世界と同様の環境をフォースディスプレイを用いて再現しようとする場合に問題となる．

提案方式は，人が合力を与える方法に合わせて合力を分配することで，バネ係数を増すことなく指へ合力が伝わりやすくし，問題を解決したといえる．

4.9 この章の結論

本章では，まず実世界で物体を把持操作する際の把持力を計測した．そして，従来の提示方法には，合力が発生すると把持力が指間距離が減少するという問題があることを指摘した．

つぎに，合力の把持力への影響を軽減するために，把持力と合力を独立に算出し，合力を形状提示のためのばねモデルを介さずに，直接物体に加えることを提案した．さらに実験を行い提案方式の有効性を確認した．

4.10 今後の課題

本研究では把持方向と平行な合力を把持対象に加える場合について議論したが，把持方向と垂直に合力を加える際にも，人は把持力を変化させる [42]．把持方向と合力の向きに応じて提示力を制御することで，指間距離の変動を抑え，操作感を向上させることができると考えられる．

4.11 この章のまとめ

この章では，物体の形状を提示するための力覚レンダリング手法を概説した後，物体の把持を安定に提示する手法を提案した．物体の形状以外に，物体表面のテクスチャや材質を提示する試みもなされている [43, 44]．また，god object 法を発展させ，さまざまな形状のポインタを用いたり，並進 3 自由度に加えて回転 3 自由度を加えた 6 自由度の操作，は

さみやピンセットのような道具の動きを加えたさらに多自由度のポインタを利用できるような力覚レンダリング手法の開発も望まれる。

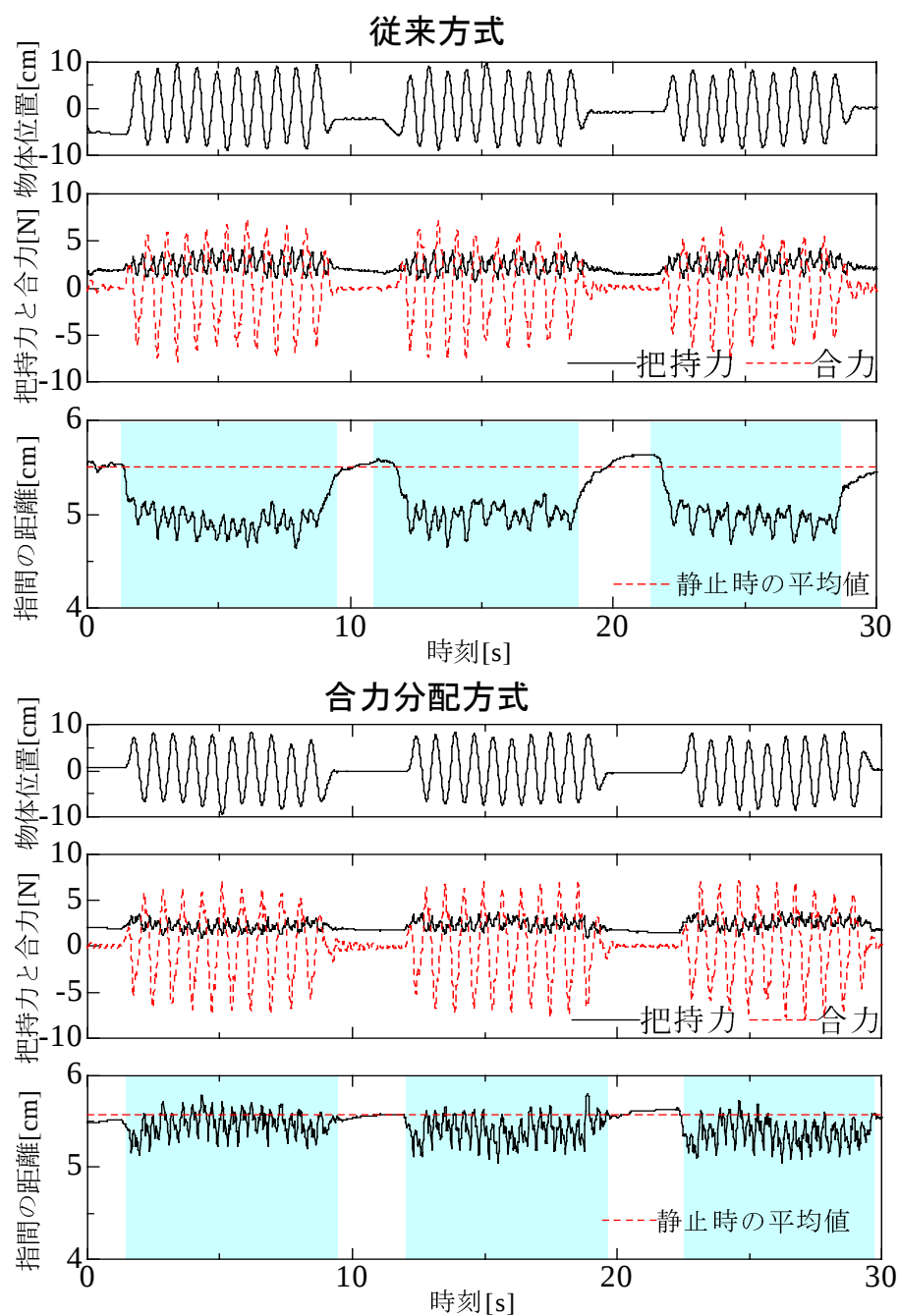


図 4.11: 従来方式と提案方式の比較

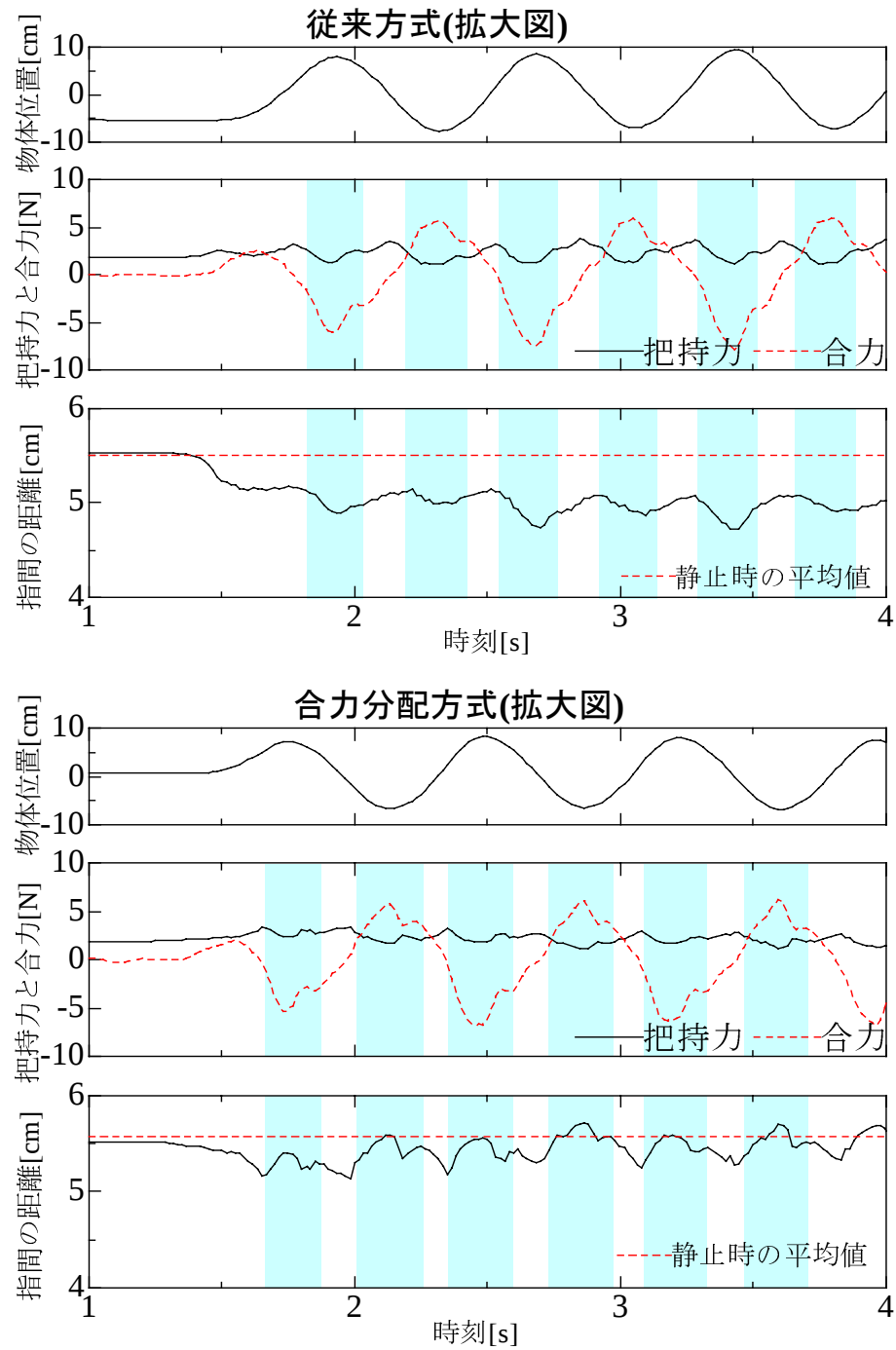


図 4.12: 従来方式と提案方式の比較 (拡大図)

第5章 ペナルティ法による力覚インタラクション

本章では，高速更新に適したペナルティ法による物理シミュレーションの方法を提案し，精度の高い力覚インタラクションを実現する．

5.1 ペナルティ法による剛体シミュレーションの概要

2.3.1 節で述べたように，ペナルティ法は1回の更新に必要な計算量が少なく，力覚インタフェースの制御に必要な高速更新に向いている．また，力覚ポイントに加わる接触力が計算できるため，力覚レンダリングに特別な処理を必要としない．このようにペナルティ法を用いると，単純な構成で更新の高速な物理シミュレーションと力覚インタラクションを実現できる．

しかし従来のペナルティ法を用いた物理シミュレータは，物体間の接触が点のみで起こると見なしており，床の上に立方体に乗った場合など，物体が広い領域で接触することを考慮していない．そのため，物体が面で接触する場合などに抗力や摩擦力を正しく計算できず，安定なシミュレーションと力覚提示が実現できなかった．

Terzopoulos ら [62] や Snyder ら [63] は，物体表面に多数のサンプル点を配置し各点の侵入量から反力を求めている．しかし，多数のサンプル点について計算が必要なため，計算量が非常に多く実時間で動作せず，力覚インタラクションには向かない．

5.2 提案

本章では，物体間の接触部分の形状を解析し，接触体積に基づいて接触力計算を行うことを提案する．これにより，従来のシミュレータでの問題を解決し安定なシミュレーションと力覚提示を実現する．提案手法では，物体を多面体で表現し，多面体同士の接触部分の3次元形状を求め，接触領域に分布バネダンパモデルを考え，発生する力とトルクから接触力を求める．提案手法はペナルティ法的一种だが，従来のペナルティ法のように接触が1点で起こると考えず，領域で起こると考え，接触部分の体積に基づいて接触力を計算する．以下に提案手法による接触力計算の特徴を説明する．提案手法は接触体積に基づいて接触力を求めるため，面同士が接触する場合でも接触力を正確に求めることができる．これは，従来の接触が点で起こると考えるペナルティ法では困難であった．

従来の点に基づく接触力計算の問題点と，提案する接触体積に基づく接触力計算の利点を抗力と摩擦力について示す．

5.2.1 接触体積に基づく抗力計算

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。

1点または頂点や稜線の交点だけを接触点と考え、接触点にペナルティ法のためのバネ・ダンパモデルを考えると、接触点が生成、消滅する際に抗力が不連続に変化するため運動が収束しない。

例えば、Kimらの手法[64]のように、最侵入点を接触点と考えて床の上の立方体の運動をシミュレートすることを考える。すると、立方体の回転によって最侵入点が切り替わるため、接触点が生成、消滅し、抗力とその作用点が不連続に変化する。この場合、特にトルクが急激に変化するため回転運動が収束しない(図5.1)。

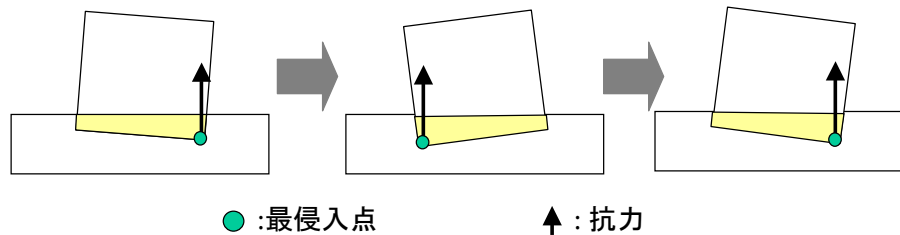


図 5.1: 抗力計算での問題点

解析的な手法[65][Bar94]では、拘束を満たす力を解くことで抗力を求めるため、接触点が生成、消滅した際に抗力や作用点が急激に変化することはない。この問題は、ペナルティ法に特有の問題だといえる。

本研究では、接触領域全体にバネダンパモデルが分布すると考える(図5.2)。これにより、抗力とその作用点が連続に変化するため、運動を収束させることができる。(図5.2)。

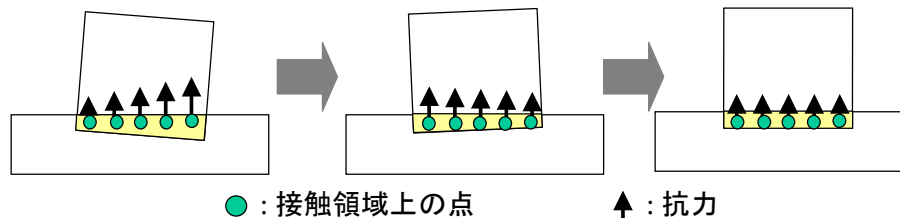


図 5.2: 抗力計算での問題点の解決法

5.2.2 接触体積に基づく摩擦力計算

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。

そのため、1点または頂点や稜線の交点だけで接触が起ると考えると、正確な摩擦力を計算できない。

例えば，摩擦力が1点に作用すると考えると，回転運動に対する摩擦トルクを計算できない．図 5.3 は，摩擦力が1点に作用するシミュレータ上の，床の上を滑る立方体を上から見た図である．

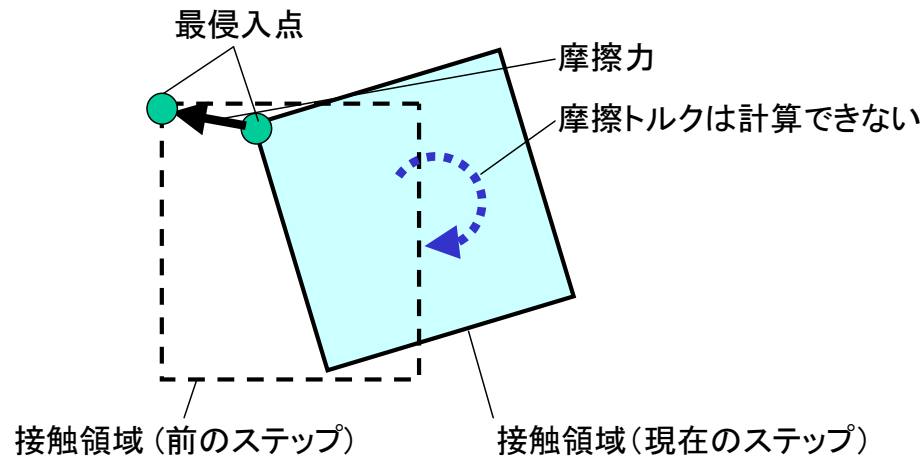


図 5.3: 摩擦力計算での問題点

本研究では，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 (図 5.4)．これにより，摩擦力だけでなく，摩擦トルクも計算できる．

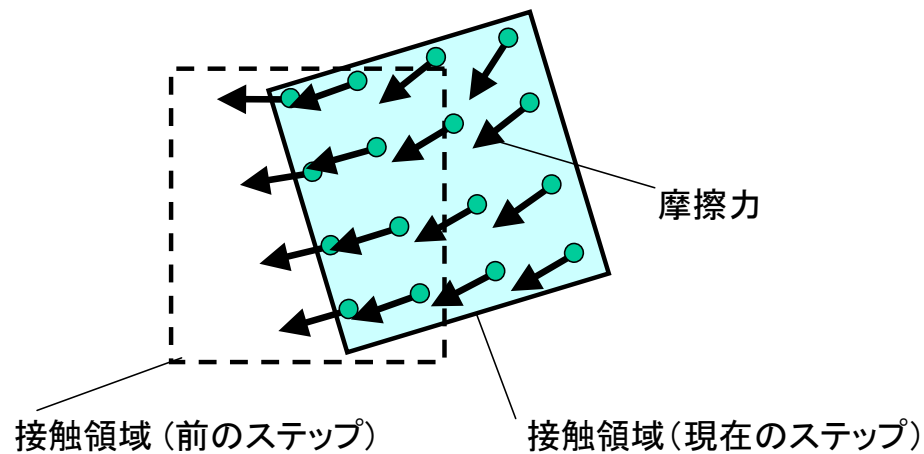


図 5.4: 摩擦力計算での問題点の解決法

5.3 提案手法による剛体運動シミュレータ

本研究では提案手法に基づいて剛体運動シミュレータを構築した．本節では構築したシミュレータの動作を説明する．

剛体の運動は，運動方程式 (5.1)，(5.2) によって表せる．

$$m\dot{\boldsymbol{v}} = \boldsymbol{F}(t) \quad (5.1)$$

$$I\dot{\boldsymbol{\omega}} = \boldsymbol{N}(t) \quad (5.2)$$

ここで， $\boldsymbol{v}$ は剛体の速度， $\boldsymbol{\omega}$ は剛体の角速度を表す．運動方程式を数値積分することで，剛体の位置と向きを求める．

剛体間に働く力を計算するためには，接触状態を調べ，接触力を求める必要がある．提案するシミュレータは形状を多面体で表現した剛体を扱う．多面体の接触状態の解析を効率よく行うため，多面体が非凸な場合，あらかじめ凸多面体に分割する．この凸多面体のすべての組み合わせについて対をつくり，各対について次の処理を行い，接触力 (= 抗力と摩擦力) を求める．

1. 接触判定，接触している場合のみ 2. 以降を行う．
2. 接触の法線と接触面を求める
3. 接触体積の形状を求める
4. 接触状態に応じた抗力と摩擦力を求める

以下，各ステップについて説明する．

5.3.1 接触判定

凸多面体対の接触判定についていくつかの手法が提案されている [66] [67] [68]．本シミュレータでは，データ構造が単純で計算が高速な GJK アルゴリズム [66] を用いた．GJK アルゴリズムは，凸形状対が与えられたときに，最近傍点対もしくは接触体積上の 1 点を求める．

5.3.2 接触の法線

本シミュレータでは，接触の法線 (抗力の向き) を決定するために，次の手続きを行う．

1. 接触している凸形状対を前のステップで求めた接触の法線の向きに離す．
2. GJK アルゴリズムを用いて最近傍点対を求め，対を結ぶ向きを接触の法線とする．
3. 凸形状対の位置を (1) の前の状態に戻し，最近傍点対の中点を含み接触の法線を法線とする面を求め，接触面とする．

凸多面体対には，ここで求めた接触面に垂直な抗力と平行な摩擦力が働くものとする．

5.3.3 接触部分の形状の解析

Muller と Preparata [70] 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、共通部分全体を求めるアルゴリズムを提案している。Muller らのアルゴリズムには、幾何学における双対変換 (図 5.5) が用いられている。この変換は、面 $ax + by + cz = 1$ を点 (a, b, c) に、点 (a, b, c) を面 $ax + by + cz = 1$ に写像する変換である。

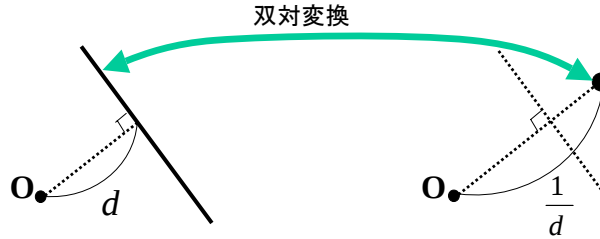


図 5.5: 双対変換

以下に Muller らのアルゴリズムを示す (図 5.6 参照)。

1. 共通部分の内部の点を原点と考え、各凸多面体を構成する平面を双対変換した頂点を求める。
2. この点集合を含む最小の凸多面体 (凸包) を求める。
3. 2. で求めた凸多面体を双対変換し、共通部分の形状を求める。

これにより、接触体積の形状 (頂点と面) が求められる。本シミュレータでは、QuickHull[71][72] を用いて、Muller らのアルゴリズムを実装し、共通部分の形状を求めた。なお、3 の手続きに必要な共通部分の内部の点には、GJK アルゴリズムによる接触判定で求まる共有点を用いる。

5.3.4 接触力計算の準備

剛体上の任意の作用点に働く力は、原点を作用点とする力とトルクで表現することができる。本シミュレータでは接触面上の各点に働く抗力と摩擦力を原点に作用点とする力とトルクに分けて積分し、剛体全体に加わる力とトルクを求める。

接触面上の任意の点 p を作用点とする力 f は、次のような原点に働く力 f_p とトルク m_p を発生させる。

$$f_p = f \quad (5.3)$$

$$m_p = p \times f \quad (5.4)$$

この f_p と m_p の総和が剛体全体に働く力とトルクとなる。

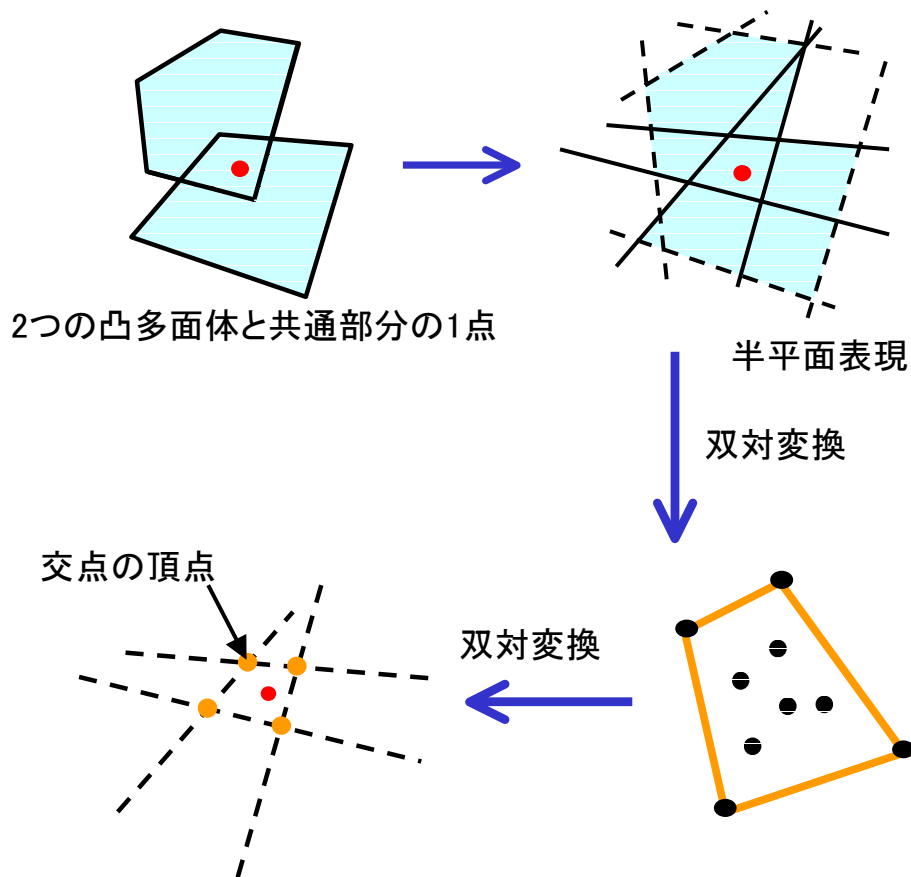


図 5.6: 2 つの凸多面体の共通部分を求める手順

2 つの凸多面体の共通部分は凸多面体になるので、接触部分は凸多面体になる。そこで、接触部分の凸多面体を構成する 3 角形について、各 3 角形ごとに力とトルクを積分し、全 3 角形について総和することで剛体全体に働く力とトルクを求める。

接触部分には、接触面に垂直な抗力と平行な摩擦力が働くと考える。本節では計算の準備として、接触面と 3 角形についての表記を示す。

接触体積の凸多面体を構成する 3 角形に対して、

i : 頂点の番号 (1,2,3) .

p : 接触面上の点 .

n : 接触面の法線ベクトル .

p_i : 接触面上に射影した 3 角形の頂点 .

$\bar{p}$: 3 頂点の重心 $((p_1 + p_2 + p_3)/3)$.

h_p : 接触面と凸多面体の面上の点との距離 .

$\bar{h}$: 3 頂点の接触面との距離の平均 $((h_{p_1} + h_{p_2} + h_{p_3})/3)$.

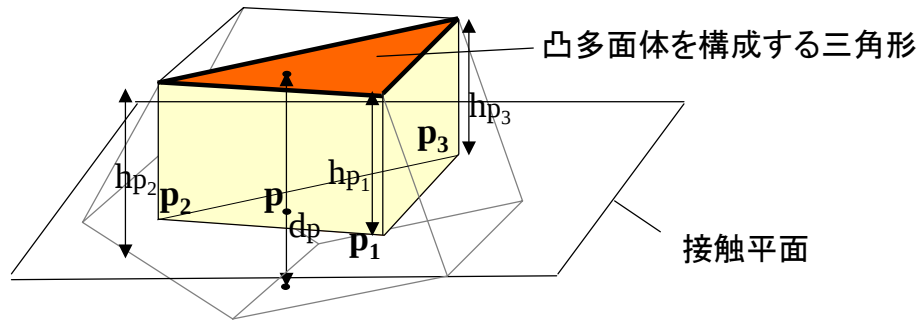


図 5.7: バネダンパモデルによる計算のための三角形分割

v_p : p での 2 つの剛体の相対速度 .

v_p^N : v_p の接触面に垂直な成分 $((v_p \cdot n)n)$.

v_p^T : v_p の接触面に平行な成分 $(v_p - (v_p \cdot n)n)$.

$\bar{v}$: 3 頂点での相対速度の平均 $((v_{p1} + v_{p2} + v_{p3})/3)$.

S : 3 角形の面積 $(\|(p_2 - p_1) \times (p_3 - p_1)\|/2)$.

f^N : 抗力 .

f^D : 動摩擦力 .

f^S : 静止摩擦力 .

f^M : 最大静止摩擦力 .

f^{*s} : バネモデルによる力 . (f^{Ns} : バネモデルが発生する抗力)

f^{*d} : ダンパモデルによる力 .

f_p^* : p に働く力 . バネモデルによる力とダンパモデルによる力の和 $(f_p^{*s} + f_p^{*d})$ になる .

m_p^* : 点 p に働く力 f_p^* によるトルク $(p \times f_p^*)$.

f_{tri}^* : 3 角形に働く力 .

と表記する (図 5.7) . 相対速度 v_i は , 接触している 2 剛体 a, b の速度 v_a, v_b 角速度 ω_a, ω_b , 重心位置 c_a, c_b に対して ,

$$\begin{aligned} v_i &= (v_b + \omega_b \times (p_i - c_b)) \\ &\quad - (v_a + \omega_a \times (p_i - c_a)) \end{aligned} \quad (5.5)$$

で与えられる .

5.3.5 抗力の計算

接触領域上にバネダンパモデルが分布している考え，モデルが発生する力を抗力とする．バネが発生する力は，各点での2物体の侵入量に比例し，ダンパが発生する力は各点での2物体の相対速度の法線成分 v_p^N に比例する．

バネモデルによる抗力 抗力のバネモデルは侵入量に比例した力を発生させる．点 p のバネモデルが発生する抗力 $f_p^{N_s}$ とそのトルク $m_p^{N_s}$ は，侵入量を d_p ，バネ定数を k^N とすると，

$$f_p^{N_s} = k^N d_p \mathbf{n} \quad (5.6)$$

$$m_p^{N_s} = k^N \mathbf{p} \times d_p \mathbf{n} \quad (5.7)$$

となる．

d_p を接触面の表側と裏側に分割すると，侵入量を3角形ごとに積分することができる．3角形上のバネモデルによる抗力 $f_{tri}^{N_s}$ とそのトルク $m_{tri}^{N_s}$ は，

$$\begin{aligned} f_{tri}^{N_s} &= k^N \int_{s=0}^1 \int_{t=0}^{1-s} ((1-s-t)h_{p_1} + sh_{p_2} + th_{p_3}) \\ &\quad (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1) dt ds \\ &= k^N S \bar{h} \mathbf{n} \end{aligned} \quad (5.8)$$

$$\begin{aligned} m_{tri}^{N_s} &= k^N \int_{s=0}^1 \int_{t=0}^{1-s} ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3) \\ &\quad \times ((1-s-t)h_{p_1} + sh_{p_2} + th_{p_3}) \\ &\quad (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1) dt ds \\ &= k^N S (\bar{h} \bar{\mathbf{p}} + \frac{1}{12} ((h_{p_1} - \bar{h})(\mathbf{p}_1 - \bar{\mathbf{p}}) \\ &\quad + (h_{p_2} - \bar{h})(\mathbf{p}_2 - \bar{\mathbf{p}}) \\ &\quad + (h_{p_3} - \bar{h})(\mathbf{p}_3 - \bar{\mathbf{p}}))) \times \mathbf{n} \end{aligned} \quad (5.9)$$

となる．

ダンパモデルによる抗力 ダンパは，接触面内の各点の相対速度の法線成分に比例した力を発生する．点 p のダンパモデルが発生する力 $f_p^{N_d}$ とトルク $m_p^{N_d}$ は，ダンパ定数を b^N とすると，

$$f_p^{N_d} = b^N v_p^N \quad (5.10)$$

$$m_p^{N_d} = b^N \mathbf{p} \times v_p^N \quad (5.11)$$

となる．バネモデルの時と同様に各3角形について積分したいが，物体が回転している場合，各点の相対速度 v_p^N と p の座標との関係が非線形になり，解析的な積分が難しい．そこでまず，3角形の頂点の相対速度 ($v_{p_1}^n, v_{p_2}^n, v_{p_3}^n$) を求め，これを線形補間したもので3角

形上の点の相対速度を近似する．これにより，3 角形上のダンパモデルが発生する力 $f_{tri}^{N_d}$ とトルク $m_{tri}^{N_d}$ は，

$$\begin{aligned} f_{tri}^{N_d} &= b^N \int_0^1 \int_0^{1-s} ((1-s-t)v_{p_1}^N + sv_{p_2}^N + tv_{p_3}^N) \\ &\quad \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| dt ds \\ &= b^N S \overline{\mathbf{v}^N} \end{aligned} \quad (5.12)$$

$$\begin{aligned} m_{tri}^{N_d} &= b^N \int_{s=0}^1 \int_{t=0}^{1-s} ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3) \\ &\quad \times (((1-s-t)v_{p_1}^N + sv_{p_2}^N + tv_{p_3}^N) \cdot \mathbf{n} \\ &\quad \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\|) dt ds \\ &= b^N S (\bar{\mathbf{p}} \times \overline{\mathbf{v}^N} + \frac{1}{12} ((\mathbf{p}_1 - \bar{\mathbf{p}}) \times (\mathbf{v}_{p_1}^N - \overline{\mathbf{v}^N}) \\ &\quad (\mathbf{p}_2 - \bar{\mathbf{p}}) \times (\mathbf{v}_{p_2}^N - \overline{\mathbf{v}^N}) \\ &\quad + (\mathbf{p}_3 - \bar{\mathbf{p}}) \times (\mathbf{v}_{p_3}^N - \overline{\mathbf{v}^N}))) \end{aligned} \quad (5.13)$$

と近似できる．

5.3.6 摩擦力の計算

提案シミュレータでは摩擦力のモデルとして，クーロンの摩擦モデルを用いる．クーロンの摩擦モデルは，物体の状態を運動状態と静止状態に分ける．静止状態の物体の接触点 p には，物体を静止させるために必要な拘束力（静止摩擦力 f_p^S ）が働く．静止摩擦力の大きさが最大静止摩擦力 f_p^M を超えると，物体は動き始め運動状態となる．運動状態の物体には，接触点の相対速度の接触面に平行な成分 v_p^T の向きに動摩擦力 f_p^D が働く．抗力を f_p^N とすると，最大静止摩擦力は， $f_p^M = \mu_0 \|f_p^N\|$ ，動摩擦力は $f_p^D = \mu \|f_p^N\| (v_p^T / \|v_p^T\|)$ となる．

提案シミュレータでは，接触面上の微小面積に発生する摩擦力がクーロンの摩擦モデルで表されるものとし，発生した摩擦力の総和が剛体に作用すると考える．

動摩擦力 接触面上の点 p での動摩擦力 f_p^D は，次の式で表せる．

$$f_p^D = \mu \|f_p^N\| (v_p^T / \|v_p^T\|) \quad (5.14)$$

この式を接触体積の凸多面体を構成する 3 角形毎に積分することで，剛体に働く動摩擦力を求めることができるが，この式の解析的な積分は難しい．そこで，ダンパによる抗力を求めたときと同様に，3 角形の頂点に働く動摩擦力 ($f_{p_1}^D, f_{p_2}^D, f_{p_3}^D$) の補間で近似すると，各

3 角形に働く動摩擦力 f_{tri}^D とそのトルク m_{tri}^D は ,

$$\begin{aligned} f_{tri}^D &= \int_0^1 \int_0^{1-s} ((1-s-t)f_{p_1}^D + sf_{p_2}^D + tf_{p_3}^D) \\ &\quad \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| dt ds \\ &= S\bar{\mathbf{f}} \end{aligned} \quad (5.15)$$

$$\begin{aligned} m_{tri}^D &= \int_0^1 \int_0^{1-s} ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3) \\ &\quad \times ((1-s-t)f_{p_1}^D + sf_{p_2}^D + tf_{p_3}^D) \\ &\quad \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| dt ds \\ &= S\bar{\mathbf{p}} \times \bar{\mathbf{f}} + \frac{S}{12} ((\mathbf{p}_1 - \bar{\mathbf{p}}) \times \mathbf{f}_{p_1}^D \\ &\quad + (\mathbf{p}_2 - \bar{\mathbf{p}}) \times \mathbf{f}_{p_2}^D + (\mathbf{p}_3 - \bar{\mathbf{p}}) \times \mathbf{f}_{p_3}^D) \end{aligned} \quad (5.16)$$

となる .

静止摩擦 静止摩擦力は抗力と同じく拘束力である . 提案シミュレータでは , 抗力の場合と同様に直接拘束を満たす力を解くことを避け , ペナルティ法を用いて求める . 提案シミュレータでは , 接触面上の各点にバネダンパモデルを考える . バネの両端は , 各々の剛体に固定する . これにより , 剛体の変位するとバネダンパが伸び復元力が働くようになる . 静止摩擦力は , この復元力で与えられるものとする .

点 p の静止摩擦力のバネモデルが発生する力 $f_p^{S_s}$ とそのトルク $m_p^{S_s}$ は , 点 p のバネモデルの伸びを l_p , バネ係数を k^S とすると ,

$$f_p^{S_s} = -k^S l_p \quad (5.17)$$

$$m_p^{S_s} = -\mathbf{p} \times k^S l_p \quad (5.18)$$

で与えられる .

剛体の変位は , 並進と回転で表すことができるので , 並進 $\mathbf{r}$ と原点周りの回転 $\boldsymbol{\theta}$ で表わす . 点 p のバネモデルの伸び l_p は ,

$$l_p = \mathbf{r} + \boldsymbol{\theta} \times \mathbf{p} \quad (5.19)$$

と表せる .

以上から，接触領域の各3角形に働く摩擦力 $f_{tri}^{S_s}$ とそのトルク $m_{tri}^{S_s}$ は，

$$\begin{aligned} f_{tri}^{S_s} &= \int_0^1 \int_0^{1-s} -k^S (\mathbf{r} + \boldsymbol{\theta} \times ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3)) \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| dt ds \\ &= -k^S (\mathbf{r} + \boldsymbol{\theta} \times \bar{\mathbf{p}}) S \end{aligned} \quad (5.20)$$

$$\begin{aligned} m_{tri}^{S_s} &= \int_0^1 \int_0^{1-s} -k^S ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3) \times (\mathbf{r} + \boldsymbol{\theta} \times ((1-s-t)\mathbf{p}_1 + s\mathbf{p}_2 + t\mathbf{p}_3)) \\ &\quad \|(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)\| dt ds \\ &= -k^S (\mathbf{r} \times \bar{\mathbf{p}} + \boldsymbol{\theta} \frac{1}{6} (\mathbf{p}_1^2 + \mathbf{p}_2^2 + \mathbf{p}_3^2 + \mathbf{p}_1\mathbf{p}_2 + \mathbf{p}_2\mathbf{p}_3 + \mathbf{p}_3\mathbf{p}_1)) S \end{aligned} \quad (5.21)$$

となり，接触領域 C に働くバネモデルによる摩擦力 f^{S_s} とそのトルク m^{S_s} は，

$$f^{S_s} = -k^S \left(\sum_{tri \in C} \mathbf{r} S + \boldsymbol{\theta} \times \sum_{tri \in C} \bar{\mathbf{p}} S \right) \quad (5.22)$$

$$\begin{aligned} m^{S_s} &= -k^S \left(\mathbf{r} \times \sum_{tri \in C} S \bar{\mathbf{p}} + \boldsymbol{\theta} \frac{1}{6} \sum_{tri \in C} (\mathbf{p}_1^2 + \mathbf{p}_2^2 + \mathbf{p}_3^2 + \mathbf{p}_1\mathbf{p}_2 + \mathbf{p}_2\mathbf{p}_3 + \mathbf{p}_3\mathbf{p}_1) S \right) \end{aligned} \quad (5.23)$$

となる．

接触面上の点 p のダンパモデルが発生する力 $f_p^{S_d}$ とトルク $m_p^{S_d}$ は， b^S をダンパ定数とすると，

$$f_p^{S_d} = -b^S \mathbf{v}_p^T \quad (5.24)$$

$$m_p^{S_d} = -\mathbf{p} \times b^S \mathbf{v}_p^T \quad (5.25)$$

となる． $\mathbf{v}_p^T$ は，

$$\mathbf{v}_p^T = \mathbf{v}^T + \boldsymbol{\omega}^T \times \mathbf{p} \quad (5.26)$$

($\mathbf{v}^T$: 相対速度の接触面に平行な成分)

$\boldsymbol{\omega}^T$: 相対角速度の接触面に平行な成分)

と表せるので，接触領域 C に働くダンパモデルによる摩擦力 f^{S_d} とそのトルク m^{S_d} は，

$$f^{S_d} = -b^S \left(\sum_{tri \in C} \mathbf{v}^T S + \boldsymbol{\omega}^T \times \sum_{tri \in C} \bar{\mathbf{p}} S \right) \quad (5.27)$$

$$\begin{aligned} m^{S_d} &= -b^S \left(\mathbf{v}^T \times \sum_{tri \in C} S \bar{\mathbf{p}} + \boldsymbol{\omega}^T \frac{1}{6} \sum_{tri \in C} (\mathbf{p}_1^2 + \mathbf{p}_2^2 + \mathbf{p}_3^2 + \mathbf{p}_1\mathbf{p}_2 + \mathbf{p}_2\mathbf{p}_3 + \mathbf{p}_3\mathbf{p}_1) S \right) \end{aligned} \quad (5.28)$$

となる．

最大静止摩擦力 接触領域に働く最大静止摩擦力は，接触領域上に分布させたクーロンの摩擦モデルの最大静止摩擦力の和だと考える．

接触領域上の点に働く摩擦力 $f_p^S = f_p^{S_s} + f_p^{S_d}$ は式 (5.17) 式 (5.24) から求まるので，各点での静止摩擦力の向き， $f_p^S / \|f_p^S\|$ は r, θ, v, ω から計算できる．

最大静止摩擦力は摩擦係数と摩擦力の向きが異なるほかは，動摩擦力と同じ式で与えられるので，点 p での最大静止摩擦力 f_p^M は，式 (5.14) の $v_p^T / \|v_p^T\|$ を $f_p^S / \|f_p^S\|$ ， μ を μ_0 に置き変えたものになる．3 角形に働く最大静止摩擦力 f_{tri}^M とそのトルク m_{tri}^M も，式 (5.15)，式 (5.16) の f_p^D を f_p^S で置き換えた式で近似できる．

状態の遷移と摩擦力の計算 各ステップに剛体に働く摩擦力を求めるためには，静止状態・運動状態の遷移を考慮しつつ，バネダンパモデルが発生する摩擦力を求める必要がある．図 5.8 に，状態と遷移条件，各状態でのバネモデルの伸び r, θ を示す．シミュレーション

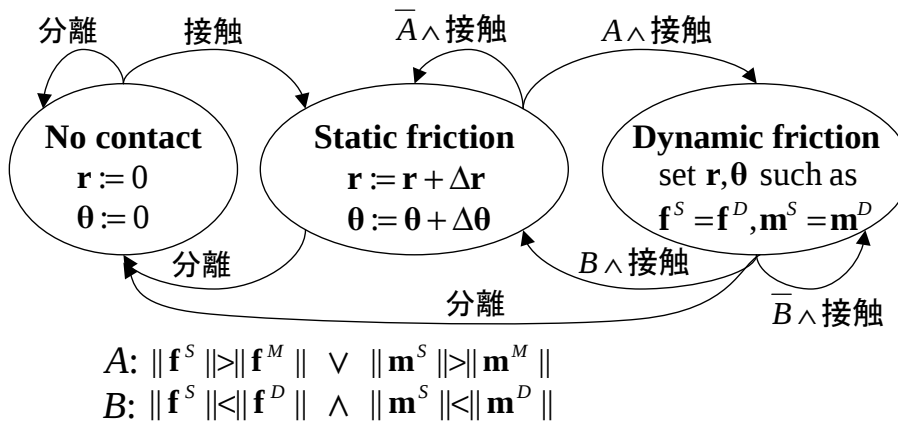


図 5.8: 摩擦力計算の状態遷移

1 ステップの更新手順は以下ようになる．

1. 2 剛体があはじめて接触したときに，剛体の変位 r, θ を 0 にする．また初期状態は静止状態とする．
 2. 静止状態では，2 剛体の相対変位 $\Delta r, \Delta \theta$ を r, θ に加算し，静止摩擦力 f^S, m^S を求める．
 3. 静止摩擦力の大きさが動摩擦力 f^D, m^D より小さければ静止状態に，最大静止摩擦力 f^M, m^M よりも大きければ運動状態に状態遷移する．
 4. 静止状態で，静止摩擦の大きさ $\|f^S\|, \|m^S\|$ が，最大静止摩擦の大きさ $\|f^M\|, \|m^M\|$ を超えていた場合， $f^S = f^D, m^S = m^D$ を満たすように r, θ を変更する． $f^S = f^{S_s} + f^{S_d}$ ， $m^S = m^{S_s} + m^{S_d}$ なので， r, θ は，式 (5.22) 式 (5.23) 式 (5.27) 式 (5.28) から求まる．
 5. 静止状態では静止摩擦力 f^S, m^S ，運動状態では動摩擦力 f^D, m^D を摩擦力とする．
- となる．

5.4 評価

提案シミュレータが従来手法の問題点を解決していることを確かめるために2つの実験を行った。また、提案手法の実用性を確かめるために、2つのシミュレーションを行った。

5.4.1 実験に用いるシミュレータ

評価実験では提案シミュレータと、Baraff[65, Bar94]などが提案する拘束を解く方式のシミュレータ、従来の点に基づくペナルティ法によるシミュレータを比較した。拘束を解く方式のシミュレータとしては、Open Dynamics Engine [73]を用いた。また、点に基づくシミュレータとしては、提案シミュレータから接触部分の形状の解析部分を削除し、最侵入点に侵入距離に比例した反力を加えるようにしたものを使用した。

5.4.2 実験1：計算時間の比較

提案手法が、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。

方法 3つのシミュレータで図5.9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 3 1.0GHzCPUを1個搭載したPCで行った。

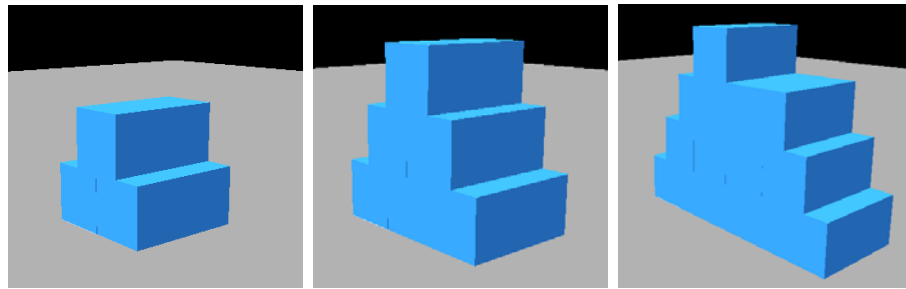


図 5.9: 評価のためにシミュレーションしたシーンの例：3,6,13 個のブロック。

結果 図 5.10 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、提案シミュレータと点に基づくシミュレータでは、ほぼ線形に計算時間が増加する。

点に基づくペナルティ法は、提案法の半分程度の計算時間になっている。しかし、点に基づくペナルティ法はシミュレーションが不安定になる。たとえば13個のブロックを積んだ場合、シミュレーション開始後1.8秒でブロックが崩れてしまう。

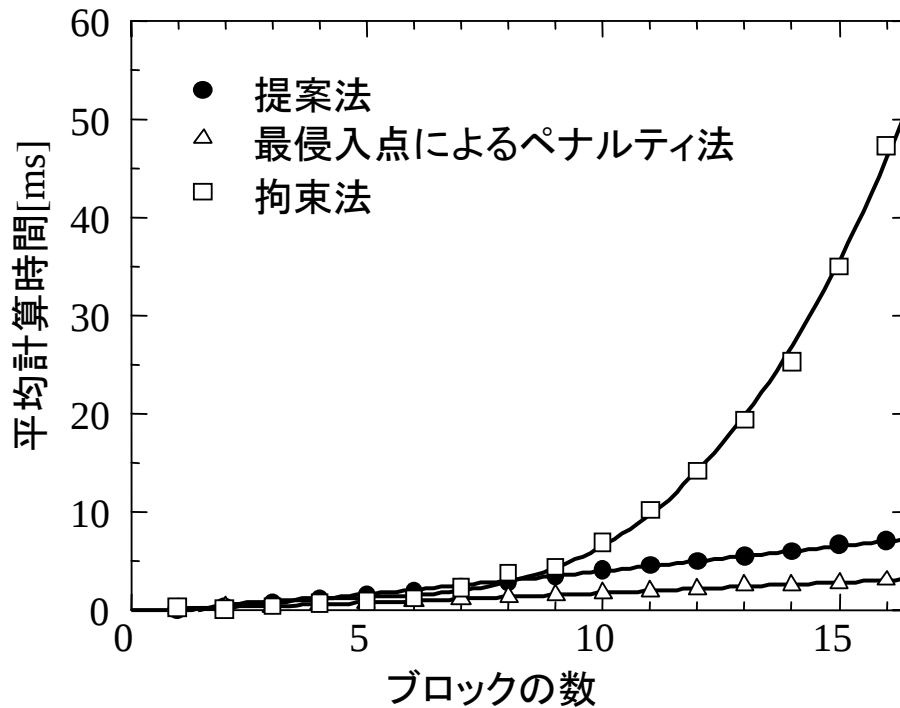


図 5.10: 計算時間の比較.

以上から、特にブロック数がある程度多くなったときには、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。また、提案法は従来の点に基づくペナルティ法よりも計算量が多いが、計算量が物体数に比例するという特長は失われていないことが分かる。

5.4.3 実験2：安定性の比較

シミュレーションの安定性を評価するため、3つのシミュレータで立方体を床に落としたときの運動のシミュレーションを以下の条件で行い、このときの立方体のZ軸まわりの角運動量を計測した(図5.11)。

- 立方体の1辺の長さは2m、質量は1kgとする。
- 初期位置は床から1m上方で、0.1rad傾いている。
- シミュレーションの時間刻みは10msとする。
- 提案手法と点に基づくペナルティ法では、バネ定数は1000N/m、ダンパ定数は50Ns/mとする。

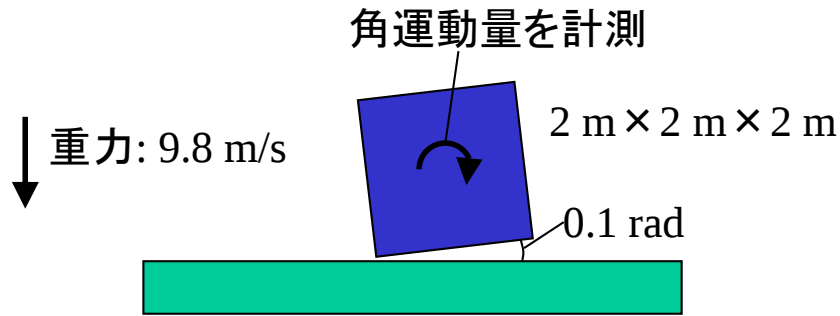


図 5.11: シミュレーションされたシーン .

結果 提案手法と拘束を解く方法では，立方体は床に着地したのち静止した．しかし，点に基づくペナルティ法では，立方体は床に着地したのち振動を続けた．図 5.12 に，立方体の角運動量をシミュレーション開始から 2 秒間について示す．提案手法と拘束を解く方法では 0.7 秒付近で収束するが，最侵入点を作用点とする手法では，さらに時刻が進んでも角運動量の振動は収まらなかった．以上から，提案シミュレータが第 5.2 節で述べた振動

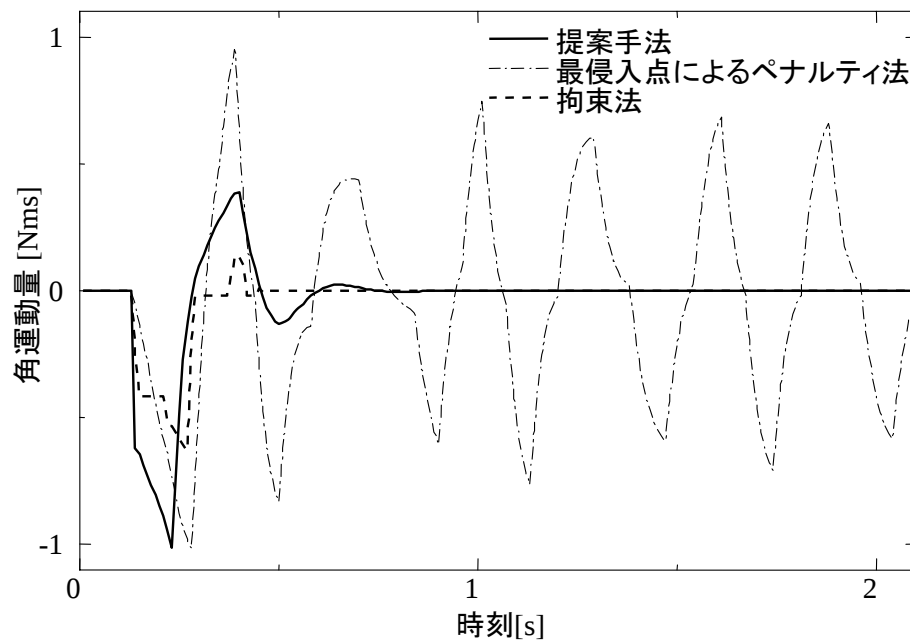


図 5.12: 立方体の角運動量の Z 軸周り成分

の問題を解決できることが確かめられた．

5.4.4 実験 3：摩擦モデルの精度の比較

シミュレータの摩擦力のシミュレーションの精度を調べるため，床の上のおもりをバネで引いた場合に起こるスティックスリップ現象を，実世界とシミュレータで比較した．図

5.13 に実世界での実験の様子を示す．実験では，おもりにバネをつなぎ，バネを一定の速度で引く．床の材質や他のパラメータはスティックスリップ現象が起こるように設定し，おもりの位置と張力を計測し，摩擦力を計算して求めた．また，3つのシミュレータで同じ状況をシミュレーションし，おもりの運動を比較した．

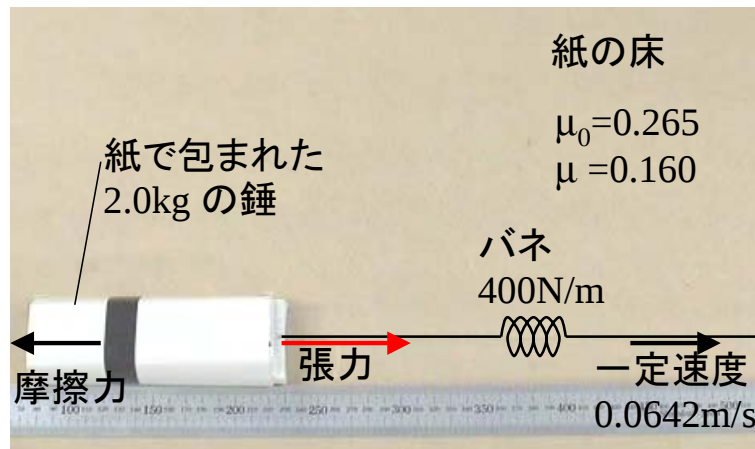


図 5.13: 実世界でのスティックスリップ運動の計測

図 5.14 に，おもりの運動と摩擦力，張力を示す．実世界と提案シミュレータではスティックスリップ運動が観測された．解析法のシミュレータでの結果は，解析法のシミュレータが動摩擦モデルだけしかシミュレーションできていないことを示している．一方，最侵入点を用いるペナルティ法のシミュレータは，接触点が接触領域内を動き回るため，静止摩擦の状態を保持できない．

実世界では，おもりのスティックスリップ運動は，少しずつ減少する．提案シミュレーションでもおもりを引くバネにダンパ成分を入れることで，この動きを再現することができた．

5.4.5 シミュレーション例 1: 直接操作環境

提案シミュレータと力覚インタフェースを用いて，直接操作環境を構築できることを確かめた．実験には力覚インタフェース SPIDAR [23] を用いて，6 個の積み木と立方体の力覚ポイントからなる直接操作環境を構築した．

その結果，積み木を動かすなどの直接操作を直感的に行えることが確認できた．図 5.15 に積み木を崩す様子を示す．

5.4.6 シミュレーション例 2: 分布摩擦力の効果

提案シミュレータは摩擦力の分布を考慮している．この効果を確認するため，摩擦力の分布を考慮せず，作用点だけに摩擦力が働くシミュレータと，摩擦力の分布を考慮したシミュレータでコマの運動のシミュレーションを行った．

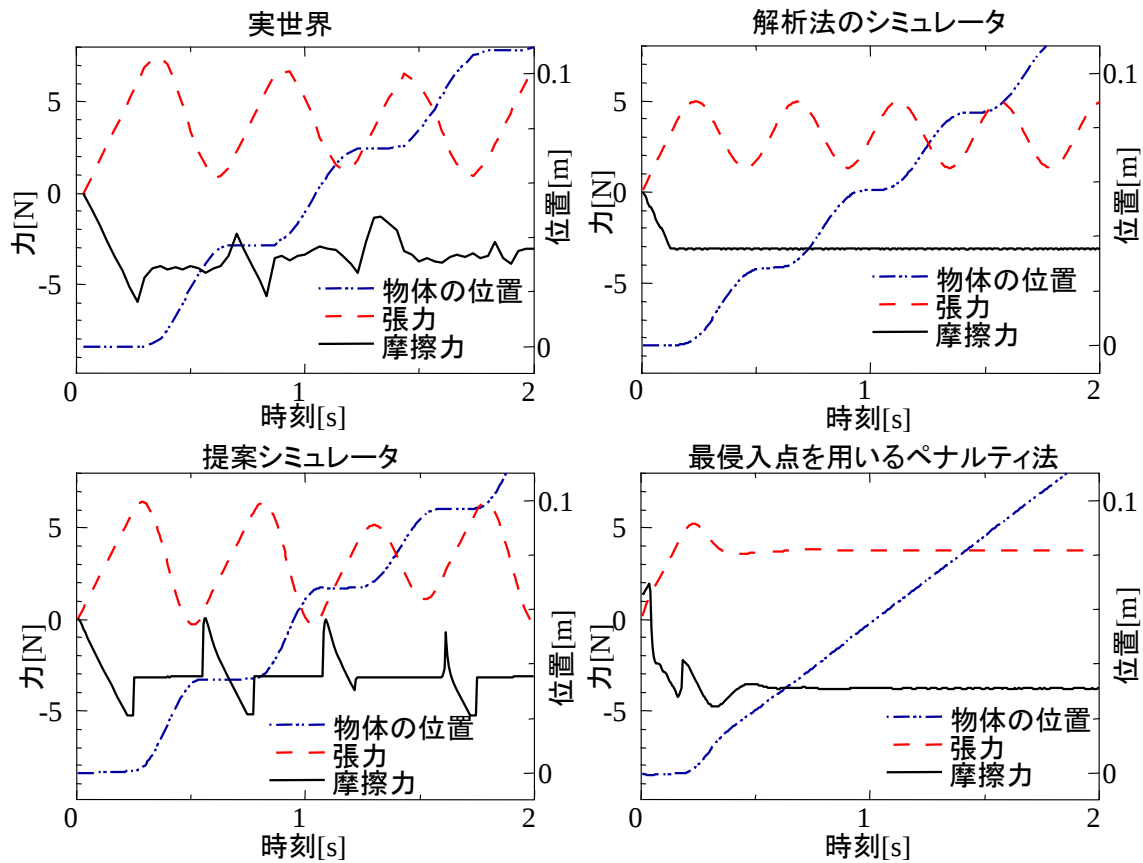


図 5.14: スティックスリップ運動の比較

提案シミュレータでは，コマが先端を支点に回転している状態から，床の上を転がり静止するまでを特別な処理を行うことなくシミュレートできたが，分布を考慮しないシミュレータでは，回転軸の変化が不自然な上，コマが倒れたあとも静止せずに床の上を回り続けた．図 5.16 にシミュレーションの様子を示す．

5.4.7 シミュレーション例 3：両手直接操作環境

提案シミュレータを用いて物体モデルを実世界と同様に直感的に操作できることを確認するため，両手操作環境を構築した．構築したシステムでは，2つの6自由度力覚インタフェース SPIDAR-G [21] を用いて直接操作を実現した．バーチャル世界には，床の上に6つのブロックが積まれており，力覚ポインタとしてシャベル状の物体を用意した．力覚レンダリングには5章で提案した物理シミュレーションで求めた接触力をそのまま用いる手法を用い，更新頻度は，物理シミュレーション，力覚提示とも 300Hz とした．

構築したシステムを用いることで，物体モデルを直感的に直接操作できることが確かめられた．図 5.17 に操作の様子と，各状況での計算時間を示す．ユーザは，抗力と摩擦力を感じながら，ブロックを拾い，保持し，受け渡し，積み上げることができた．操作中の計算時間は最大で 2.8ms で，ほとんどの状況で 2ms 以下であり，300Hz の更新頻度を保つ

ことができた．

5.5 この章のまとめ

本章では，力覚インタフェースの制御と剛体の物理シミュレーションの両方を高速更新する力覚インタラクションシステムを構築した．力覚提示のための応答性を得るために，ペナルティ法を用いた剛体運動シミュレータを用いた．この中で，従来のペナルティ法のシミュレータでは接触が点で起きると見なしているため，抗力と摩擦力が正しく計算できないという問題点があることを指摘し，接触体積の形状を求め，接触領域全体に働く抗力と摩擦力を解析的に積分する手法を提案した．評価実験を行い，面接触の安定なシミュレーションができること，正確な摩擦力がシミュレーションできることを確かめた．

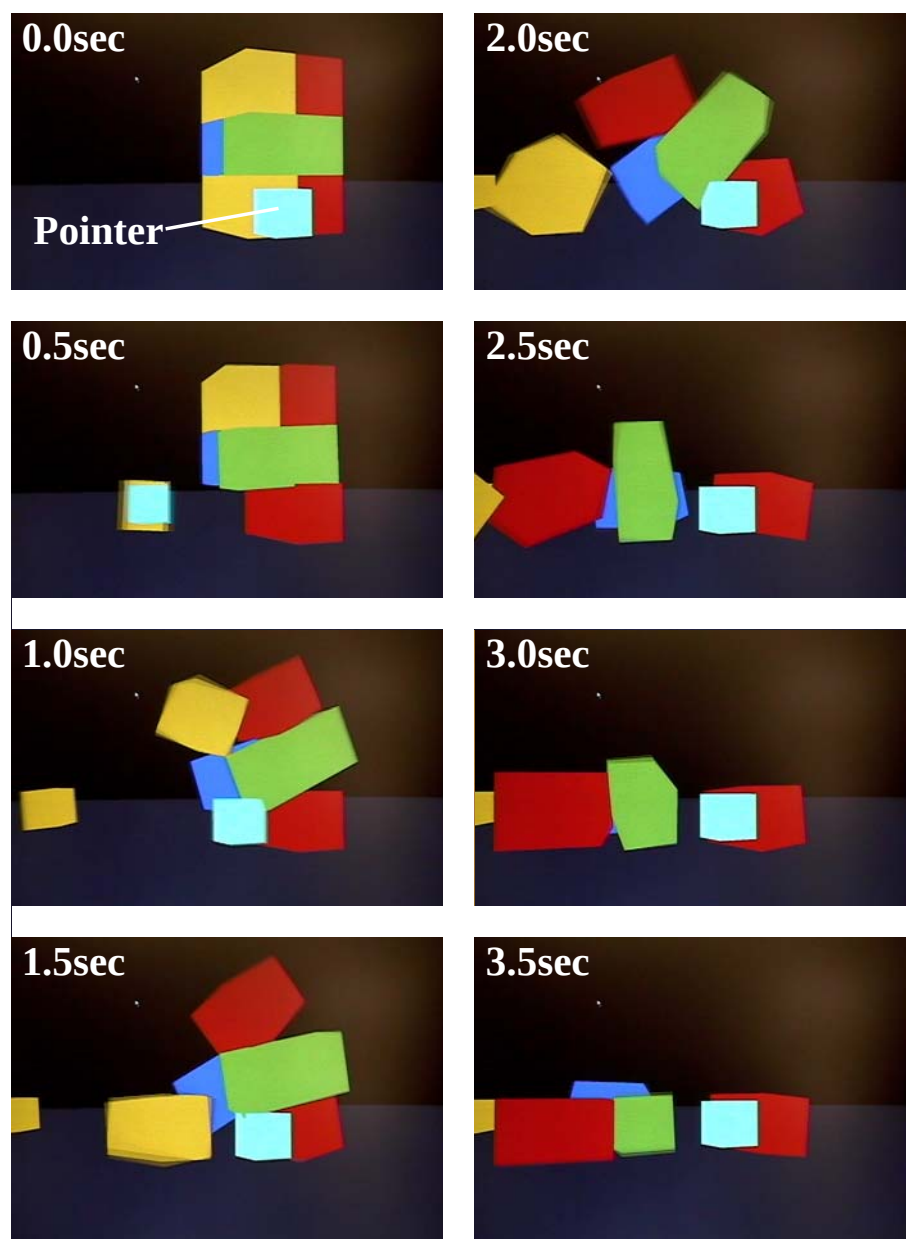


図 5.15: 力覚インタフェースを用いたインタラクション

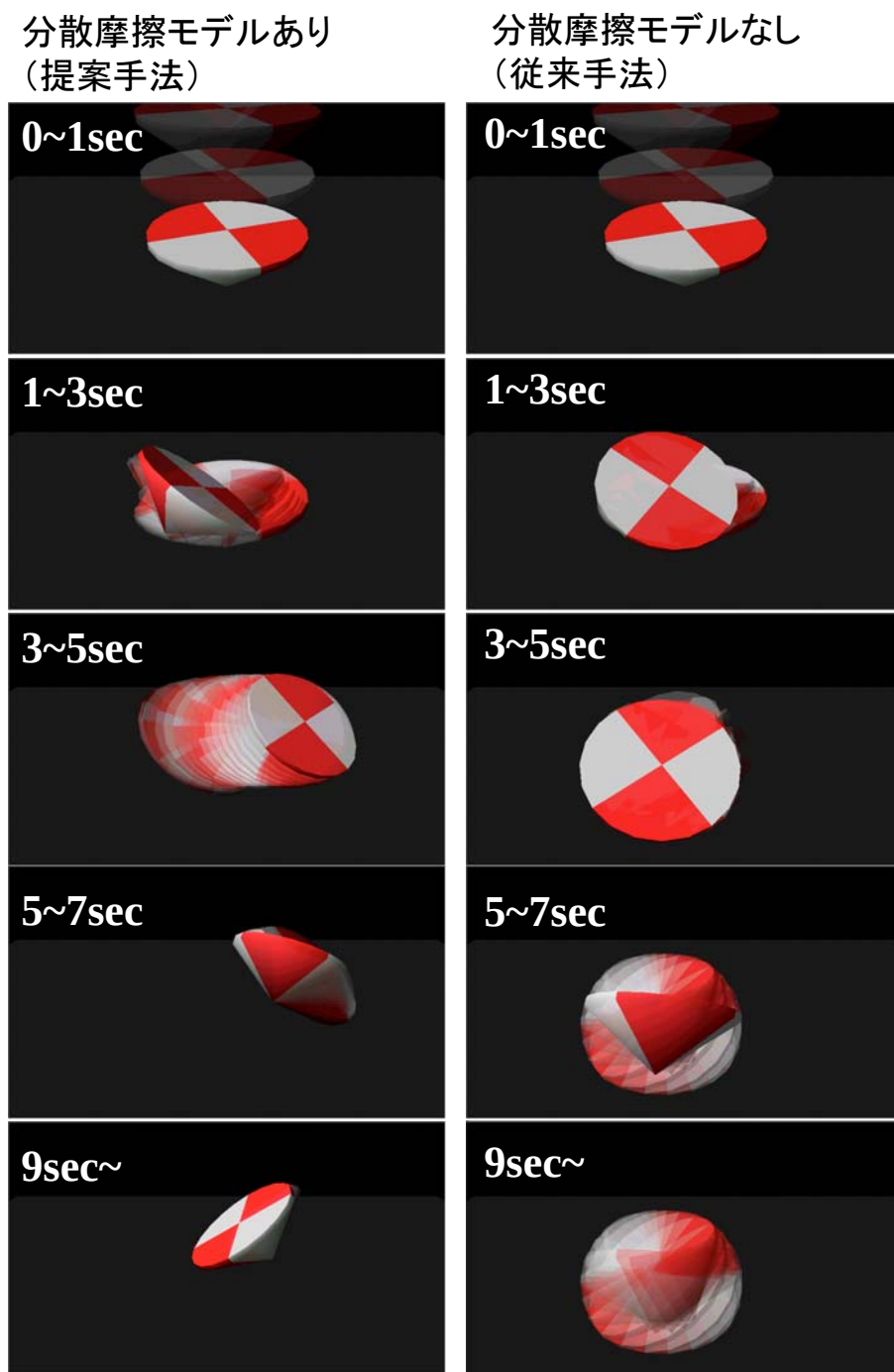


図 5.16: コマの回転運動の比較

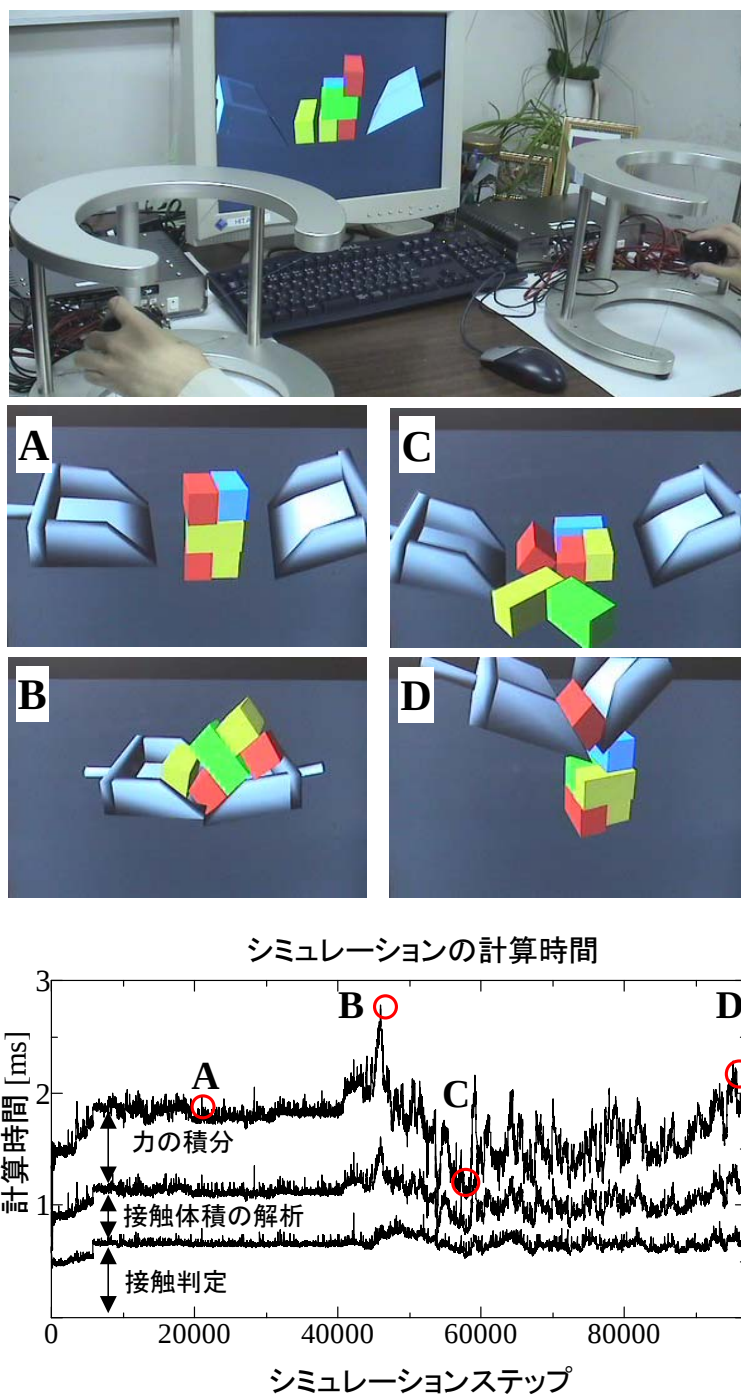


図 5.17: 両手直接操作環境の映像と計算時間．CPU が Pentium 4 2.8GHz の PC を用いてシミュレーションした．

第6章 解析法による力覚インタラクション

2章で述べたように，大規模なVR世界をシミュレーションするためには，物理シミュレーションの計算量を少なくするため，物理シミュレーションの更新を遅くしなければならない．安定な力覚提示のために高速に更新しなければならない力覚インタフェースの制御と，大規模なVR世界全体を管理しなければならない物理シミュレーションを同じ頻度で更新するのは効率が悪い．そこで，両者を別のプロセスに分割し，力覚インタフェースを制御するプロセスだけを高頻度で更新することが望ましい．

本章では，更新頻度の違いを補間する力覚レンダリング手法について概説したあと，動的な世界を扱うための補間力覚レンダリング手法を提案する．

6.1 静的世界のための補間力覚レンダリング

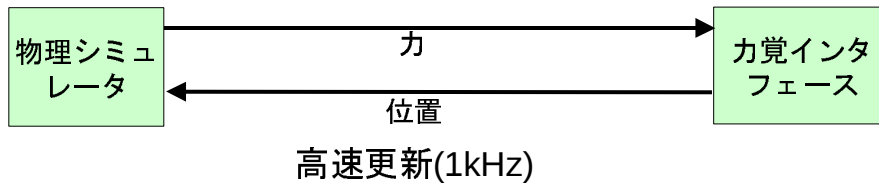
大規模なVR世界の力覚提示を行うため，VR世界の管理を低速で行い，力覚提示装置の制御を高速に行う方法とそのために力覚レンダリングプロセスで補間を行う手法が提案されている [74, 56, 75]．これらの研究では，VR世界の力覚ポインタ近傍の物体形状の特徴を中間表現（平面や球などの簡単な基本形状）に変換し，力覚レンダリングプロセスに送る．また力覚レンダリングプロセスは，中間表現に基づいてユーザーの指に働く力を高速（1kHz）に計算する（図 6.1）．この手法により複雑なバーチャル世界を表現可能な力覚提示システムが構築できるようになった．

6.1.1 動的世界での問題

6.1節の力覚レンダリング手法は静的なVR世界は上手く表現できるが，物理シミュレーションされた動的な世界は上手く表現できないという問題を持っている．

例としてVR世界に物体を置き，それを指で突付くことを考える．指で突付かれた後の物体の速度は指の位置計測の計測時刻によって変化する．図 6.2-A と 図 6.2-B の違いは物理シミュレータが指の位置を受け取る更新時刻だけである．A，B どちらの場合にも指は同じように動き，ユーザには同じだけの力が提示される．しかし，A の場合には指が物体に衝突したことが物理シミュレータに検出され物体は動くが，B の場合には検出されず物体は動かない．このように従来方式では物理シミュレータの更新時刻の影響で安定にVR世界を更新できない．

全体を高速に更新するシステム



力覚制御だけを高速に更新するシステム

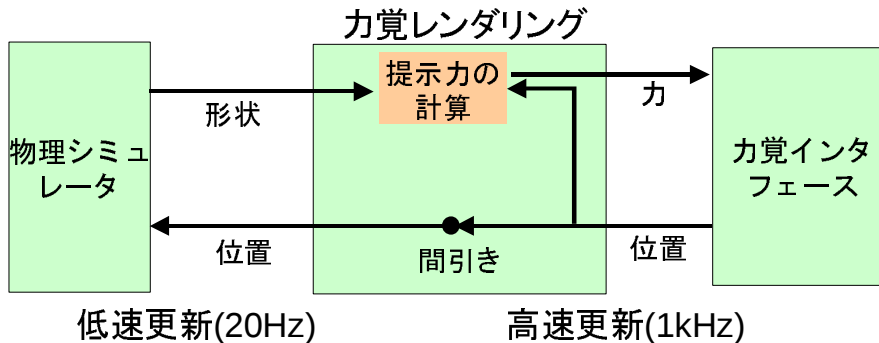


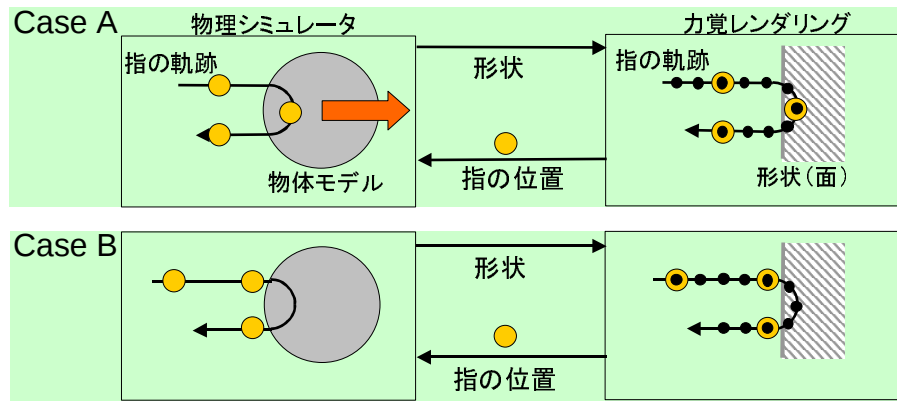
図 6.1: 処理によって更新周期が異なるシステム

6.1.2 通信情報の不足

6.1 節の力覚レンダリング手法 [74, 56, 75] では VR 世界は、形状特徴という高度な情報を力覚レンダリングプロセスに送っている。そして力覚レンダリングプロセスは形状特徴に基づき提示力の計算を行ない、形状特徴をユーザに提示する力の情報に変換して力覚インタフェースに送っている。しかし力覚レンダリングプロセスは計測した位置情報に処理を加えずに、位置情報を物理シミュレータに送り返している。

力覚レンダリングプロセスが送り返す情報をより高度な情報に変換することにより、ユーザの意図を物理シミュレータに正確に伝えることができると考えられる。本研究は動的な VR 世界を安定に更新するために必要な情報を明らかにし、高度な情報を送り返す力覚レンダリングプロセスを提案することで動的バーチャル世界を安定に更新可能な力覚インタラクションシステムを構築することを目的とする。

6.2 節では、問題が物理シミュレータが受け取る情報の不足によって発生することと、物体の運動が力積によって求まることを示し、力積を用いた通信方式を提案する。さらに例をあげて提案手法の効果を示す。6.3 節では、提案する力覚提示システム全体の動作を示す。6.4 節では、シミュレーションによって提案手法の効果を確認するとともに、実験を行ない提案方式が VR 世界の操作性を改善していることを示す。



- : 力覚レンダリングプロセスが計測した指の位置
- : 物理シミュレータが受け取った指の位置

図 6.2: 問題の例

6.2 動的世界への対応

6.2.1 問題の原因

更新時刻の影響は、物理シミュレータが受け取る位置情報の更新周期が粗いために起こる。力覚レンダリングプロセスが計測した指の軌跡をすべて物理シミュレータへ送ることによりこの問題は避けられるが、それでは通信量が大幅に増えてしまう。そこで、物体の運動の安定な更新のために必要な情報について考える。

6.2.2 物体の動きと力

物体の運動は運動方程式で表される。運動方程式を積分すると、物体の速度の変化は物体に加えられた力積によって決定されることがわかる。

ここで:

m : 物体の質量.

I : 慣性行列 (3×3 行列).

v : 重心速度.

r : 重心位置.

ω : 物体の角速度.

f : 物体に働く力.

p : f の作用点の位置.

とすると、剛体の運動は:

$$m \frac{dv}{dt} = f \quad (6.1)$$

$$\frac{dI\omega}{dt} = (p - r) \times f \quad (6.2)$$

と表される.

式 (6.1) を t から $t + \Delta t$ まで積分すると:

$$\int_t^{t+\Delta t} m \frac{dv}{dt} dt = \int_t^{t+\Delta t} f dt$$

更新後の物体重心の速度は:

$$v(t + \Delta t) = v(t) + \frac{1}{m} \int_t^{t+\Delta t} f dt$$

となる.

式 (6.2) を t から $t + \Delta t$ まで積分すると:

$$\int_t^{t+\Delta t} \frac{dI\omega}{dt} dt = \int_t^{t+\Delta t} (p - r) \times f dt$$

となる. ここで, Δt が十分小さく I, p, r が定数だとみなせるとすると:

$$\Delta\omega = I^{-1}((p - r) \times \int_t^{t+\Delta t} f dt)$$

となり, 更新後の物体の角速度は:

$$\omega(t + \Delta t) = \omega(t) + I^{-1}((p - r) \times \int_t^{t+\Delta t} f dt)$$

となる.

以上から, 物体の速度が力積によって決まることがわかった.

6.2.3 力積を用いた通信

前の章で述べたように, 物体にかかったある時刻での力は重要ではなく, ある期間にかかった力積が物体の運動を決める.

そこで, 力覚レンダリングプロセスから物理シミュレータへの通信に力積を用いることを考える. 図 6.3 は, 6.1 節の通信内容 [74, 56, 75] と 6.2.3 節の通信内容を対比している.

6.1 節の手法では, 力覚レンダリングプロセスは指の位置情報のみを物理シミュレータに送っていた. 本節の手法では, 力覚レンダリングプロセスは, 物理シミュレータの 1 周期の期間, ユーザが VR 世界に加えた力を積分し力積を求める. そして力積の情報を物理シミュレータに送る. 物理シミュレータは VR 世界を力積に基づいて更新する. 更新の手続きは以下の通りである.

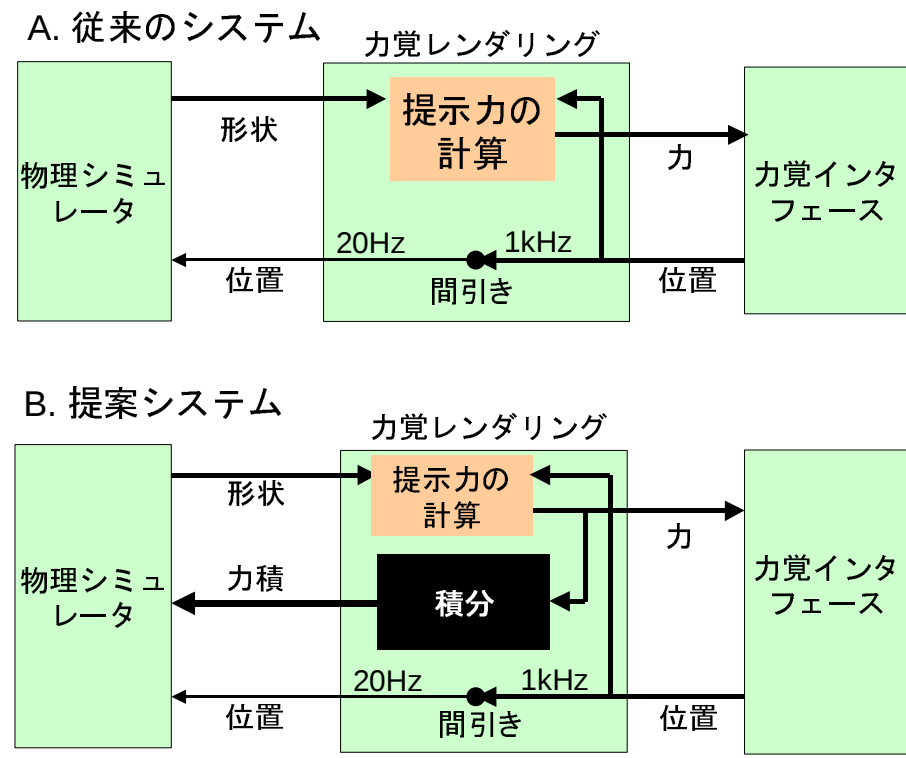


図 6.3: 動的世界のための補間力覚レンダリングと従来方式との比較

1. 物理シミュレータ: VR 世界の形状特徴を求める.
2. 物理シミュレータ: 形状特徴を力覚レンダリングプロセスに送る.
3. 力覚レンダリングプロセス高速更新部分:
 - (a) 指位置の計測する.
 - (b) 形状特徴を補間し, 提示する形状を求める.
 - (c) 力覚レンダリングを行ない, 形状を表すために提示する力を求める.
 - (d) 力を提示する.
 - (e) 提示した力の積分する.
4. 力覚レンダリングプロセス: 力積, 位置を物理シミュレータに送る.
5. 物理シミュレータ: 力積に基づいて VR 世界を更新.

6.2.4 動的世界での効果

動的世界での効果を示すため, 例をあげる. 図 6.4 は突付かれた後の物体の動きへの提案手法の効果を表している. 従来方式では, 物理シミュレータが受け取った指位置に基づいて, 指の物体モデルへの侵入量 (=図 6.4 上の長さ) に基づいて物体の速度を更新するため,

サンプリング時刻によって押された後の物体の速度が異なる。6.2.3節の方式では、力覚レンダリングプロセスが力積(=図6.4下の面積)を計算するため、サンプリング時刻の影響がほとんどない。

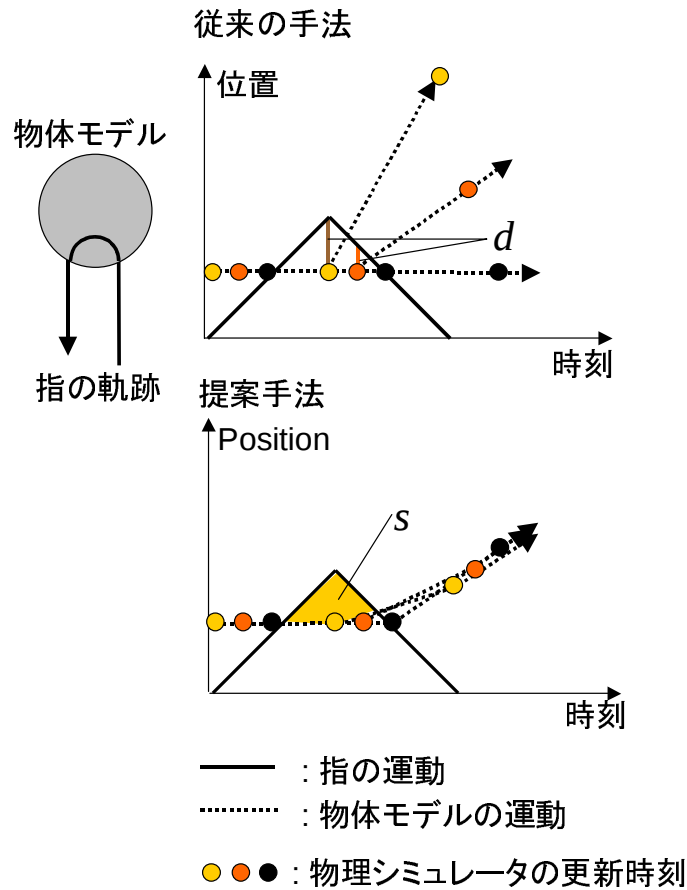


図 6.4: 提案手法の効果

6.3 力覚インタラクションシステムの構築

6.2節で示した力積を用いた通信方式に基づいて力覚提示システムを構築した。本節では、構築した力覚提示システムの詳細を述べる。

6.3.1 形状特徴の求め方

物理シミュレータはVR世界から形状特徴を求め、力覚レンダリングプロセスに送る。構築したシステムでは形状特徴として複数の平面を用いた。

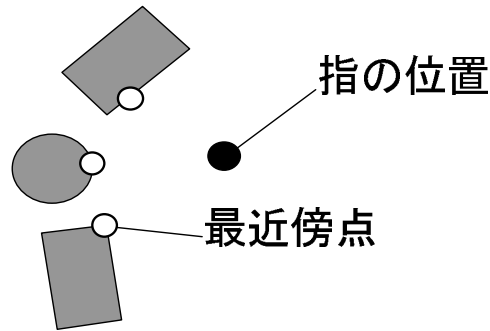
以下にVR世界から形状特徴である平面を求める手順を示す。

1. 指を中心としたある半径内の物体モデルを列挙する。

2. 1で列挙した物体モデル1つ1つについて, 指から一番近い点を求める.(図 6.5 を参照).
3. 1で列挙した物体モデル1つ1つについて, 2で見つかった点を通り, 法線が指位置に向きの平面を作る.(図 6.5 を参照).

ここで, 平面とは2で見つけた点と3で見つけた法線ベクトルの組のことを表す.

2. 最近傍点を見つける



3. 面を作る

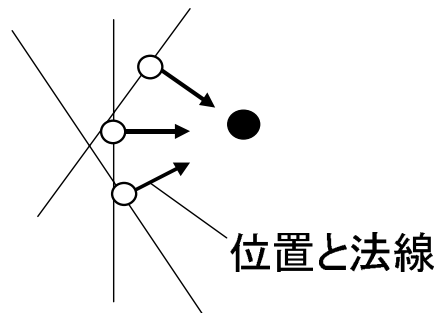


図 6.5: 平面の準備

6.3.2 形状特徴の補間

力覚レンダリングプロセスの更新は高速だが, 形状特徴は物理シミュレータの低速な更新にあわせて送られる. 形状に変化があった場合, 受け取った形状特徴をすぐにユーザに提示すると, 形状へのユーザの指の侵入量が突然変化することがあるため, 突然大きな力を提示してしまい, ユーザは不自然な力覚を感じることになる. そこで力覚レンダリングプロセスは受け取った形状特徴を時間軸上で補間し, 形状特徴が連続に変化するように形状提示する.

力覚レンダリングプロセスは最後に受け取った形状特徴 (B) とその1つ前に受け取った形状特徴 (A) を利用して補間を行なう. 2つの形状特徴の間を補間する方法として, A と B の間を補間する内挿と A と B から変化率を求め, 変化分を B に足す外挿が考えられる. 外

挿を行なうとプロセス間の通信遅れを除去できるが、形状特徴の変化を線形に予測することになるため、予測が外れた場合に誤った形状を提示することになる。そこで構築したシステムでは A と B を内挿した。

構築したシステムでは形状特徴として複数の平面を用いている。物理シミュレータが更新する間に力覚レンダリングプロセスが n 回更新するものとする。物理シミュレータプロセスの i 回目の更新から $i+1$ 回目の更新の間に、力覚レンダリングプロセスは ni 回目の更新から $ni + n - 1$ 回目の更新を行なう。

物理シミュレータプロセスが i 回目の更新時に送る平面の位置を P_{W_i} 法線ベクトルを N_{W_i} 、力覚レンダリングプロセスが $ni + j$ 回目の更新時に補間する平面の位置を $P_{H_{ni+j}}$ 法線ベクトルを $N_{H_{ni+j}}$ とすると補間した平面は

$$P_{H_{ni+j}} = \frac{j}{n}P_{W_i} + \frac{n-j}{n}P_{W_{i-1}}$$

$$N_{H_{ni+j}} = \frac{j}{n}N_{W_i} + \frac{n-j}{n}N_{W_{i-1}}$$

と表せる。実際には複数の平面が送られるので、各平面についてこの計算を行なう。

6.3.3 力覚レンダリング

力覚レンダリングプロセスは補間した形状特徴と計測したユーザの指位置からユーザに提示する力を決定する。構築したシステムでは、ばねダンパモデル [76] を用い、ユーザの指が平面に侵入した量とその微分に比例した力を提示している。

ユーザの指位置を C_j 、すべての平面の外側の点でユーザ指位置に最も近い点を D_j とすると提示する力を f_j は、

$$f_j = k_p(D_j - C_j) + k_d \frac{((D_j - C_j) - (D_{j-1} - C_{j-1}))}{\Delta t}$$

(k_p : ばね定数, k_d : ダンパ定数, Δt : 更新周期) と表される。

6.3.4 力の積分

物理シミュレータの 1 周期の間、力覚レンダリングプロセスはユーザが VR 世界に加えた力を積分する。ユーザが加えた力と、力覚インタフェースが発生した力は、大きさが等しく向きが反対である。物理シミュレータの i 回目の更新時に力覚レンダリングプロセスが送る力積を F_i 、力覚レンダリングプロセスの $ni + j$ 回目の更新時にユーザに提示する力を f_{ni+j} とすると、

$$F_i = \sum_{j=0}^{n-1} f_{n(i-1)+j}$$

となる。

6.3.5 VR世界の更新

物理シミュレータは受け取った力積に基づいて物体の運動を更新する．構築したシステムでは物体モデルとして剛体を扱った．

剛体の質量を m 慣性テンソルを I 速度を v_i 角速度を ω_i 重心位置を r 衝突点の位置を p とすれば, 更新後の物体の運動は

$$\begin{aligned} v_i &= v_{i-1} + \frac{1}{m} F_i \\ \omega_i &= \omega_{i-1} + I^{-1}((p - r) \times F_i) \end{aligned}$$

となる．

6.4 評価

6.4.1 シミュレーション

摩擦のある床の上に置かれた物体モデルを指で押した後の物体の動きを計算機上でシミュレートした．同じ状況を5つのサンプリング時刻についてシミュレートし, 形状特徴だけを通信する方式と力積を戻す手法を比較した．

図 6.6 は物体モデルの動きを比較したグラフである．物体モデルの位置更新はサンプリング時刻に行なわれるため, 動きは階段状になる．

グラフ上の5本の線は, 物体モデルの動きを表す．5本の違いは物理シミュレータの更新時刻だけである．シミュレーションの結果から, 提案手法を用いたBの場合には従来手法に比べて更新時刻の影響が小さくなっていることがわかる．

6.4.2 実験

実験のためのシステム

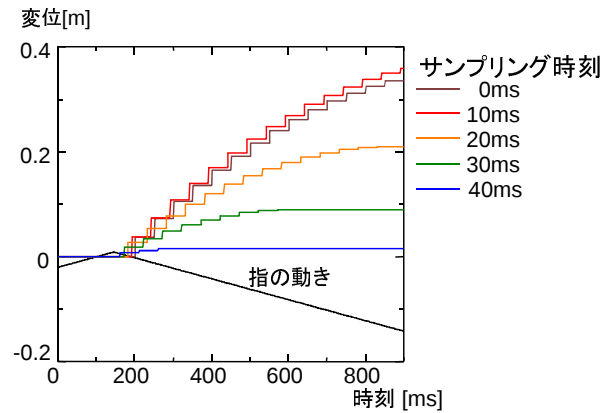
実験のためのVR世界の更新には3次元の物理シミュレータを用いた．物理シミュレータは重力, 空気抵抗, 物体の衝突, 物体同士の摩擦を考慮したものであった．実験のための

力覚提示装置	SPIDAR (図 3.1)
計算機	PC Pentium-II 300MHz
システムの詳細を以下に示す．	
世界管理プロセスの更新周期	20Hz
力覚提示プロセスの更新周期	1kHz
ばねモデルの係数	2kN/m

実験の作業

図 6.7 は実験時に被験者に提示する画面である．物体モデルは, 距離目盛として横縞に模様がつけられた床の上に置かれている．実験の作業は, 物体モデルを手前から奥に向かって一度だけ指で押し, 目標位置である 30cm の目盛まで物体モデルを動かすことである．被験者はこの作業を従来手法と提案手法について 30 回行った．

A: 押されたあとの物体の運動(従来手法)



B: 押された後の物体の動き(提案手法)

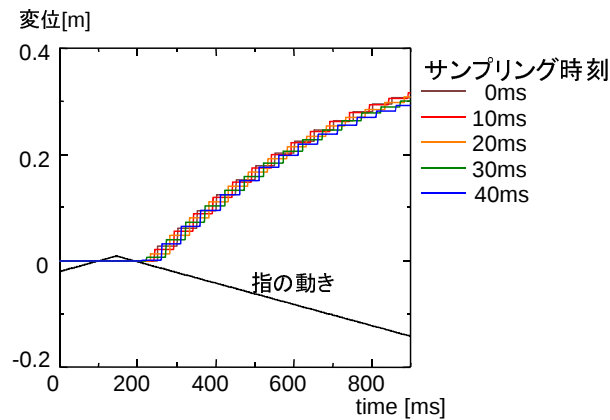


図 6.6: シミュレーション結果

実験結果

立方体の停止位置を被験者1人分について図6.8に示した．提案手法を用いることで従来手法に比べて停止位置のばらつきが小さくなっていることが分かる．図6.9は作業に十分慣れた後の20試行の誤差の偏差を表す．実験結果から提案手法が操作の正確性を改善することが分かる．

6.4.3 多物体とのインタラクション

解析法に基づく物理シミュレータと6章で提案した補間力覚レンダリングを用いて，多物体操作環境を構築し多数の物体を直接操作できるか確かめた．

システムは PentiumII 300MHz プロセッサを1つ持つPCを用いて構築した．バーチャル世界には，12個のボールと床と4つの壁を配置した．シミュレーションの更新周期は10Hz，力覚レンダリングと力覚インタフェースの制御は1kHzで行った．操作の様子を図6.10に示す．

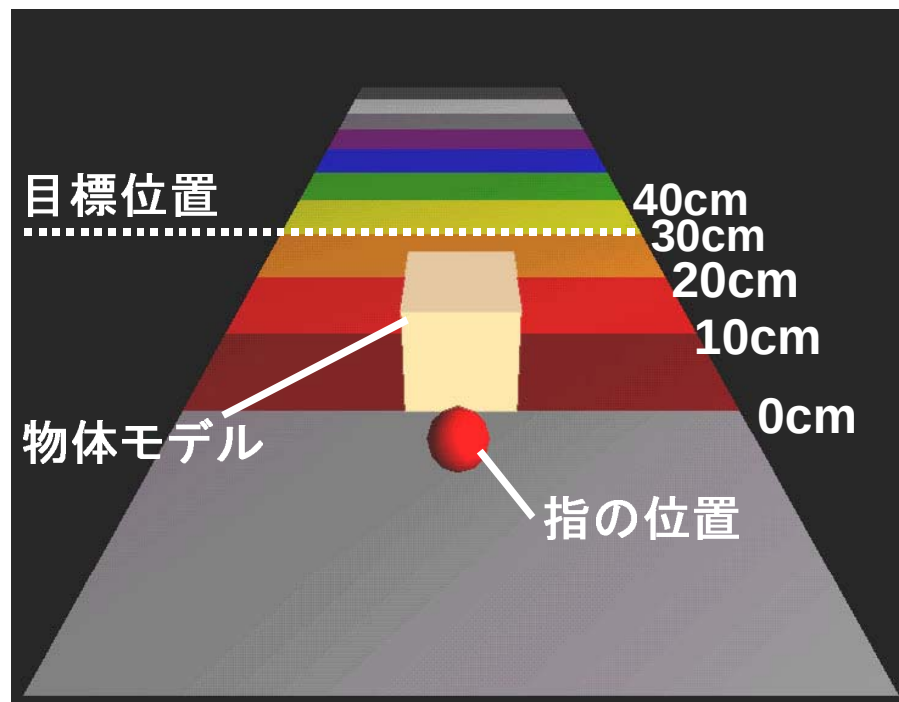


図 6.7: 実験の作業の画像

ユーザは、自然な力覚を感じながら、球状の力覚ポインタで球状の物体に触れたり押して動かしたりすることができた。また、実験中シミュレーション処理、力覚処理ともリアルタイムに行うことができた。

6.5 考察

6.5.1 力覚提示と画像提示の違い

VR 世界を画像提示するシステムでは、従来から VR 世界の更新処理と画像提示処理 (画像レンダリング) を分割し、画像レンダリングをハードウェアで行なうなどの分担を行なっている。

そこで、VR 世界の更新と力覚レンダリングの分割を、画像提示システムの分割と比較する。

画像提示システムでの処理の分割

画像提示装置はユーザに情報を提示する出力装置であるが、ユーザの意図をバーチャル世界に伝える入力装置ではない。そのため図 6.11 のように情報はバーチャル世界からユーザに向かって流れる。

ディスプレイと VR 世界の間グラフィックスエンジンが入り作業を分担すると、VR 世界が送り出す情報がより高度になる。

力覚インタラクションシステムでの分割

力覚インタフェースは形状と力の提示の他に、ユーザの指位置とユーザが加えた力をVR世界に入力する機能がある。そのため図6.12のように情報がVR世界とユーザの間で双方向に流れる。

形状特徴を通信するだけの静的世界しか扱えないシステムでは、力覚インタフェースとVR世界の間で力覚レンダリング処理が入り作業を分担する際に、VR世界が送り出す情報はより高度な情報へと変化する。しかしユーザからVR世界への情報は変化しないため、ユーザの意図をVR世界に正確に伝えることができなかった。

力積を通信に用いた動的世界を扱えるシステムでは、VR世界が受け取る情報が力積という高度な情報になるため、ユーザの意図を正確に伝えることができたと考えられる。

6.5.2 画像レンダリングプロセスによる時間補間

実験のためのシステムで、物理シミュレータの更新周期を10Hzにしたところ、画像中の物体モデルの更新が滑らかでなくなったが、物体モデルを触ったり押したりする感覚はそれほど不自然にはならなかった。この場合、画像レンダリングプロセスが、力覚レンダリングプロセスと同じように物体の位置を時間補間して表示することで、画像提示の滑らかさを改善できると考えられる。

6.5.3 摩擦の表現

力覚レンダリングプロセスは、面の位置を補間する。このときの面の位置は、物体と指の衝突点を表すため、面の位置の速度は面が表す物体の速度とは異なる。摩擦力を計算するためには、物体と指の相対速度が必要となる。摩擦を表現するために物理シミュレータは、面の位置とは別に物体の速度の情報を力覚レンダリングプロセスに伝達する必要がある。この考えを実装したところ、摩擦が表現できることが確かめられた。

6.5.4 処理分割の階層性

本研究は力覚レンダリングプロセスの機能を増やし、通信内容を高度にすることで物理シミュレーションの更新周期が低くても自然な力覚インタラクションが実現できることを示した。力覚レンダリングプロセスの機能を更に増やし、それにあわせてより高度な情報を通信すれば、より自然なバーチャル世界の更新ができると考えられる(図6.13)。

力積を用いる力覚レンダリングプロセスは、力積の計算を行ない、力積を通信することで操作の正確性を改善した。剛体の運動計算の一部を力覚提示プロセス行ない、剛体の情報の一部を通信に含めたところ、世界管理プロセスの更新を低く押さえた場合の力覚提示の感覚が改善された。

6.5.5 物体の変形

力覚提示装置で物体の変形を行う方法はいくつかあるが、それらは指と物体の重なりに基づいて物体を変形している [77]。提案手法を用いれば、力積に基づいて物体を変形することでより自然な変形を実現できる可能性がある。

6.6 この章のまとめ

VR 世界と力覚レンダリングの更新処理を分割しすることで、大規模な VR 世界との力覚インタラクションが実現する。本章では、このようなシステムで物理シミュレータによる動的な世界を取り扱うためには、力覚レンダリングプロセスから力積情報を VR 世界の更新を行う物理シミュレータに送ることが必要となることを示した。また、動的なバーチャル世界を安定に更新し、自然な力覚提示を行う力覚インタラクションを実現した。

力積を用いない力覚レンダリングプロセスは VR 世界の形状特徴情報を力情報に変換してユーザに提示するが、ユーザが VR 世界に加えた力を VR 世界に伝えておらず、単純な位置情報だけを VR 世界に送り返す。そのため、更新時刻の影響により物体モデルの運動にばらつきが発生する。提案した力積を用いた力覚レンダリングプロセスは、ユーザが力覚インタフェースに加えた力を積分し力積を求めて VR 世界に送り返すことで、ユーザの操作を VR 世界に正確に伝えることができ、更新時刻の影響を解消できた。

6.6.1 リアルタイム性の確保

これまで、映画やアニメーションなど映像の生成を目的に、精度の高い物理シミュレーションが行われてきたが、これらは時間をかけて事前に計算することを前提にしていた。物理シミュレーションによって VR 世界を実現するためには、ユーザの操作がシミュレーションに反映され、その結果がすぐにユーザに提示される必要がある。この操作から提示までのシステムの反応時間が十分に短いことが必要であることは、従来の VR 研究によって知られてきたが、5 章、6 章では、提示方法や提示する感覚 (視覚・力覚) によってシステムの反応時間への制約が異なることに着目し、特に反応時間の制約が厳しい力覚について、物理シミュレーションによって作られた VR 世界とのインタラクションを実現している。5 章では、シミュレーションの更新頻度が高く反応が早いペナルティ法を用いることで、また 6 章では、高速更新の対象を力覚ポインタ近傍に限ることでシステムの反応時間の制約を満たすことを提案しているといえる。

6.6.2 ペナルティ法と解析法の比較

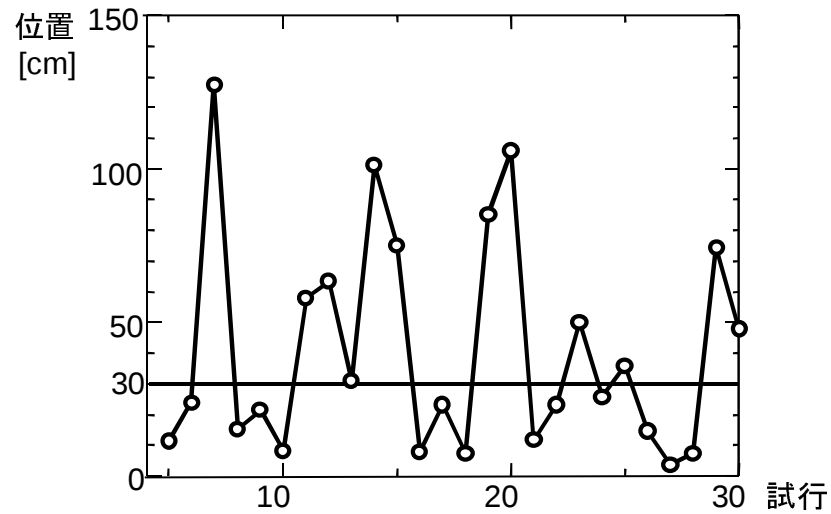
物理シミュレータを 300Hz で更新する 5.4.7 節のシステムでは、Pentium 4 2.8GHz のプロセッサを用いて、2 つの力覚ポインタと 6 つのブロックと床の計 9 個の物体モデルをシミュレーションしている。一方、物理シミュレータを 10Hz で更新する 6.4.3 節のシステムでは、Pentium II 300Hz のプロセッサを用いて、12 個のボールと床と 4 つの壁の計 16 個

の物体モデルをシミュレーションしている．力覚インタフェースの制御や映像提示も一定の計算量を必要とするので，1.8 倍の物体モデルを $1/10$ 以下の計算量でシミュレーションできたことになる．物理シミュレーションの更新周期が $1/30$ になっていることから，これは妥当な結果だといえる．

半導体の集積度が 2 年で 2 倍になり，計算機の処理速度が 2 年で 2 倍になるとしても，30 倍の速度になるまでには，6 年以上かかり，物理シミュレーションを 10kHz で更新するために必要な 1000 倍の処理速度を得るためには，20 年待たなければならない計算になる．6 年～20 年が長いかわかりは意見の分かれるところだと思うが，工学の分野では無視できない時間だと思う．

一方，力覚インタラクションの精度を考えると，物理シミュレーションの更新が遅いシステムでは，どうしても補間処理に破綻が起きる場合が考えられる．たとえば，6 章で提案した，力積を戻す力覚レンダリングの場合，6.5.4 節のような問題がある．これを解決するために，力覚レンダリングが局所的な物理シミュレーションを行うことも考えられる．しかし，剛体は他の剛体と接触していれば運動に影響を受けるので，接触している剛体を含めてシミュレーションしなければならなくなり，局所的な物理シミュレーションの規模が大きくなってしまう．また，これを補間する手法も提案されられるが，どこまで行っても全体を高速にシミュレーションする手法に漸近するだけである．このため，2 章で示した精度と規模の 2 つの方向性は，今後もどちらかに統合されることなく，それぞれに発展し続けると考えられる．

従来手法



提案手法

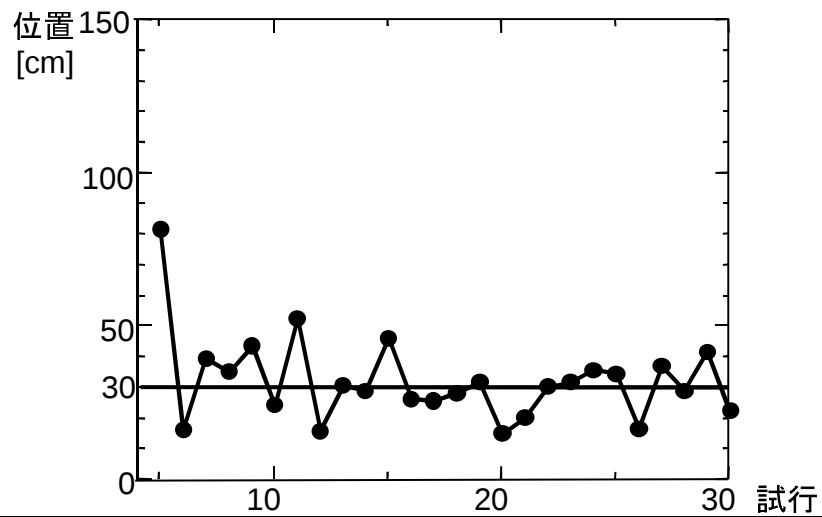


図 6.8: 物体モデルの停止位置

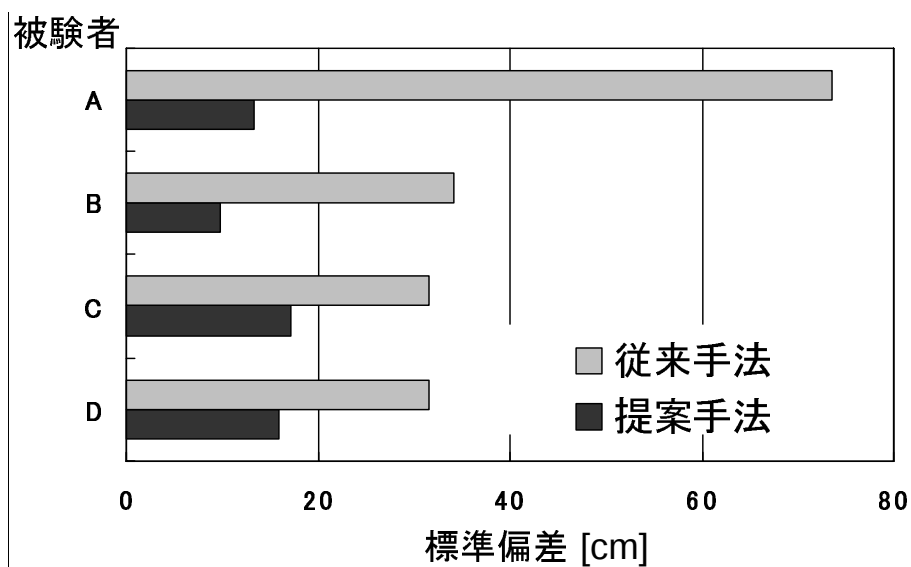


図 6.9: 作業精度の比較

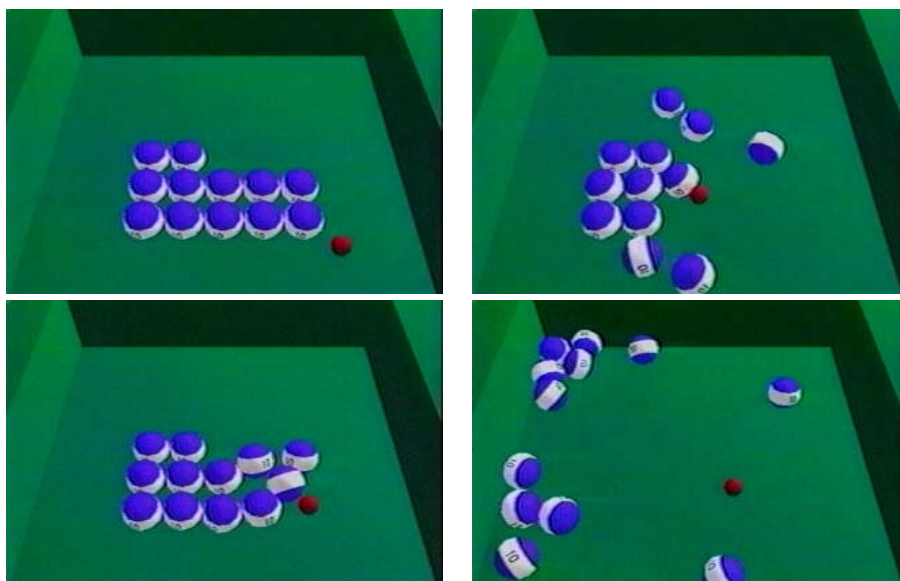


図 6.10: 低速更新シミュレーションによる多物体操作環境

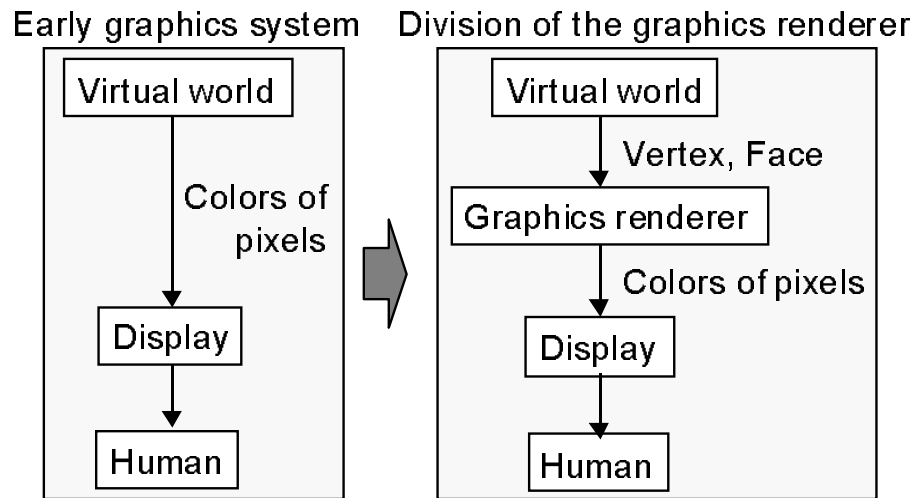


図 6.11: 画像提示処理の階層化

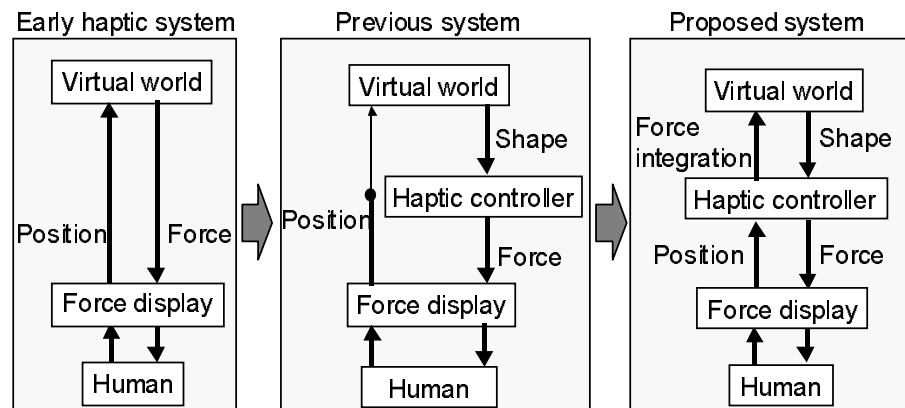


図 6.12: 力覚提示処理の階層化

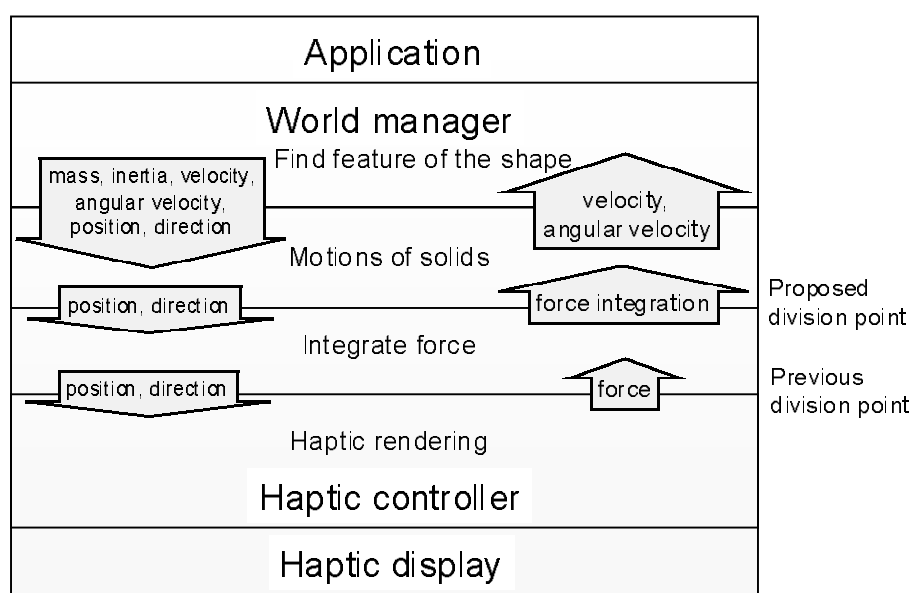


図 6.13: 力覚提示プロセスの役割と通信内容

第7章 結論

7.1 本論文の結論

本論文では、バーチャルリアリティ (VR) 世界の物体モデルが実世界の物体と同様に動き、物体モデルを実世界と同様の感覚で操作できるような VR システム、力覚インタラクションシステムを提案した。

第1章「序論」では、VR 研究を概観し、VR 世界での物体モデルの動作生成手法と物体モデルを操作する手法について研究の流れを次のように概説した。近年ではヒューマンインタフェースの発展より、VR 世界の物体モデルをより直感的・直接的に操作できるようになった。また、物理法則をシミュレーションして物体モデルの動作を生成することで、直接的な操作による多様な入力に応じた物体モデルの動作が生成できるようになった。

そして、次のように手による操作について考察し、具体的な研究目的を導いた。人は手で物体を操作するとき、対象物体の運動を完全に拘束するわけではなく、対象物体が他の物体から受ける力と手で加える力をバランスさせて目的の運動を実現している。例えば、机の上の物体を移動させるときには、物体をつかみ持ち上げたあと目標の位置に移動して放すのではなく、物体を指で押したり、軽くつまんで引きずったりして移動させることが多い。このような操作を VR 世界の物体モデルに対して行うためには、操作時に手に加わる反力の感覚 (= 力覚) が人に提示される必要がある。また、人が手で操作する物体は多くの場合変形を無視できるので、変形しない物体のモデルである剛体をシミュレーションすることで、実世界と同様の物体動作が生成できる。そこで、剛体シミュレーションによって物体動作を生成し、力覚インタフェースによって力覚提示と操作入力を行うことで、剛体シミュレーション世界との力覚インタラクションを実現し、実世界と同様の感覚で物体操作が可能な力覚インタラクションシステムを提案することを具体的な研究目的とした。

第2章「力覚インタラクションシステムの基本構成」では、提案する力覚インタラクションシステムが、ユーザに映像を提示するコンピュータグラフィックス、力を提示する力覚インタフェース、提示力を計算する力覚レンダリング、物体動作を生成する物理シミュレーションから構成されることを示した。そして、物理シミュレーションに必要な接触力の解法にペナルティ法と解析法があり、力覚インタラクションの精度を重視する場合は高速更新に向けたペナルティ法、計算効率を重視する場合は低速更新に向けた解析法が適することを明らかにし、2つのシステム構成を提案した。

第3章「力覚インタフェース」では、力覚インタフェースの機構と制御方法を検討し、糸を用いたパラレル機構によるインピーダンス提示型のインタフェースが高精細な力覚提示に適することを明らかにし、機構と制御方法を提案した。また、正確で安定な力覚提示には力覚インタフェースを高速に制御する必要があることを示し、専用のプロセッサを用

いて高速制御することで、精細な形状や多様な材質感を提示することを提案した。

第4章「力覚レンダリング」では、まず、物体形状の提示手法について説明し、次に実物体を手で操作する場合の基本操作である把持の安定な提示方法を提案した。具体的には、人の手による把持では多指が物体に触れるため、提示力の分配に冗長性があることを指摘し、人の把持特性を考慮して分配することで、安定な把持提示を実現する。

第5章「ペナルティ法による力覚インタラクション」では、力覚制御と物理シミュレーションの両方を高速に行う力覚インタラクションシステムの構成と、接触部分の形状を考慮した正確で安定な接触力計算を提案した。提案接触力計算を用いて力覚レンダリングを行い、第3章で提案した力覚インタフェースを用いた力覚インタラクションシステムを構築した。また、評価実験によって正確で安定な接触力計算と高速更新の維持ができることが確かめた。

第6章「解析法による力覚インタラクション」では、解析法による構成は物理シミュレーションの更新を低速にすることで計算負荷を抑えることができ、大規模なVR世界に適していることを示し、シミュレータの情報を力覚レンダリングで補間して自然な力覚を提示すること、ユーザが加えた力積に基づいて物体モデルの運動を正確に更新することを提案した。そして、第3章で提案した力覚インタフェースを用いて力覚インタラクションシステムを構築し、自然な力覚提示と物体モデルの運動生成ができることを実験により確かめた。

第7章「結論」では、本論文が、剛体シミュレーション世界との力覚インタラクションを実現する力覚インタラクションシステムの提案により、物体モデルが実世界の物体と同様に動き、物体モデルを実世界と同様の感覚で操作できるようなVR世界を実現するものであると述べた。具体的には、力覚インタラクションを実現するために、力覚インタラクションシステム全体を高速に更新して正確で安定な力覚インタラクションを実現する構成と、シミュレーションを低速に更新することで計算負荷を抑え大規模VR世界との力覚インタラクションを実現する構成の、2つの構成を提案・構築し、評価実験を行ってそれぞれを実現したことを確かめた。以上のように本論文は、実世界と同様にVR世界の物体モデルを操作できる環境を提案、構築し、人とVR世界の自然なインタラクションを実現するものであり、VR研究の発展に貢献すると考えられる。

7.2 今後の展望

本研究では、剛体シミュレーション世界との力覚インタラクションを実現した。提案した力覚インタラクションシステムでは、VR世界の物体が実世界と同様に動作するため、実世界での多くの作業を再現することができる。また、多自由度力覚インタフェースを用いた直接操作と力覚提示を実現しているため、手を用いた器用な操作を自然な感覚で行うことができる。これらにより、従来は難しかった、手を器用に用いて複数の物体を操作する作業が、VR世界で再現可能となる。このため、提案システムには、製品の設計支援・評価、組み立てや造形などの作業支援・作業評価・作業環境評価などをはじめ、高度なインタラクションを利用したエンタテインメントなどさまざまな応用が考えられる。

今後の研究の展望としては、まず力覚インタラクションの高精度化が挙げられるが、空

間と時間双方の精度をバランスよく追求していく必要があると考えられる．また，本研究では，VR 世界内の物体モデルとして剛体を扱ったが，剛体と関節を組み合わせたロボットや生き物の体，柔軟物や流体といった剛体以外の物など，単純な剛体以外の物体との力覚インタラクションを実現することも，VR の研究と応用に新しい展開をもたらすと考えられる．

謝辞

本研究の機会を与えてくださり，日ごろから熱心なご指導をいただいた佐藤誠教授に心より感謝致します．卒研配属以来，卒業研究，修士研究の指導教官として，また，助手着任後は上司として公私に渡り多大なご指導をいただきました．先生の研究とその進め方に対する考え方に多くのことを学びました．本研究を進め，論文をまとめるにあたり，討論を通じて数多くの有益なご助言とご意見をいただきました小池康晴助教授，高松亮助手（現埼玉大学助教授），石井雅博助手（現富山大学助教授）に深く感謝いたします．また，長橋宏教授，小野功助教授，長谷川修助教授には審査を通して多くの貴重なご意見を頂きました．深く感謝いたします．

橋本直己助手，丸田英徳技官（現長崎大学助手），赤羽歩研究支援推進員には，日常的な議論の相手をしていただきました．また，本論文で示した研究の一部は，研究室の学生の皆様と共同で行ったものです．3.3節～3.6節のアルゴリズムとソフトウェアの開発は井上雅晴氏（現日本ユニシス）との共同研究で，3.2.3節の高速制御プロセッサの制御ソフトウェア，3.7.1節の評価は，赤羽克人氏との共同研究で，3.7.2節のソフトウェアの開発は岩下克氏（現日立製作所）と共同で行いました．また，5章の物理シミュレータの構築は，田崎勇一氏，藤井伸晃氏との共同研究で行いました．また，3.2.3節の回路設計，ソフトウェアの開発では，(株)サイヴァース 川村英二社長，今橋正彦氏，長久寛氏に無理なお願いをいろいろと聞いていただきました．皆様に深く感謝いたします．

日ごろから議論の相手となっていた河原田・佐藤・小池・張研究室のOBの皆様，学生の皆様，東京工業大学ロボット技術研究会の皆様など，数え切れない多くの方に日頃から支えていただき，また苦楽を共にする喜びを頂きました．厚くお礼申し上げます．

参考文献

- [1] Photo Archives Project Whirlwind, http://www.mitre.org/about/photo_archives/whirlwind_photo
- [2] I. E. Sutherland: The Ultimate Display, Proceedings of IFIP Congress, pp. 506-508, 1965.
- [3] 野村淳二他, 人工現実感によるシステムキッチン体験システム, JSEP-57-08, pp.44-47, 1991
- [4] 平田幸広, 水口竹尚, 佐藤 誠, 河原田弘: 組立操作のための仮想作業空間, 電子情報通信学会論文誌, Vol.J76-D-11, No.8, 1993
- [5] 柳田康幸, 舘すすむ: クラスを用いた仮想環境の構成法, 電気学会論文誌 C 分冊, Vol. 115-C, No. 2, pp.236-244, 1995
- [6] Ruspini, Diego, Oussama Khatib, "Dynamic Models for Haptic Rendering Systems." Advances in Robot Kinematics: ARK'98, June 1998, Strobl/Salzburg, Austria, pp523-532.
- [7] Makoto Sato, Laroushi Bouguila, Tomohiro Yendo, Shoichi Hasegawa: Virtual Basketball, SIGGRAPH 97 Electric Garden Project.
- [8] B. Fröhlich, H. Tramberend, A. C. Beers, M. Agrawala, D. Baraff: "Physically-based Manipulation on the Responsive Workbench", Proc of IEEE VR 2000, pp.5-12, 2000.
- [9] F. H. Raab, E. B. Blood, T. O. Steiner, H. R. Jones, "Magnetic position and orientation tracking system", IEEE Trans. Aerosp. and Electronic Sys., vol. AES-15, no. 5, pp.709-718, 1979.
- [10] Brenner, A.E., de Bruyne, P. SONIC PEN: A digital stylus system. IEEE Transactions on Computers, V. C 19, No 6 (1970) pp. 546-548. —he device is made by Science Accessories Corp. of Southport, Conn.
- [11] K.B.Shimoga: "A Survey of Perceptual Feedback Issue in Dexterous Telemanipulation; Part I. Finger Force Feedback", Proc. IEEE VRAIS '93, pp. 263-270, 1993.
- [12] K.B.Shimoga, "A Survey of Perceptual Feedback Issue in Dexterous Telemanipulation; Part II., Finger Force Feedback", Proc. IEEE VRAIS '93, pp.271-279, 1993

- [13] F. P. Brooks, M. Ouh-Young, J. J. Batter, P. J. Kilpatrick: Project GROPE - Haptic Displays for Scientific Visualization, In Proc. ACM SIGGRAPH, pp. 177-185, Dallas, TX, 1990.
- [14] T. H. Massie and J. K. Salisbury: "The PHANTOM Haptic Interface: A Device for Probing Virtual Objects". Proc. of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Nov. 1994.
- [15] R. J. Adams, M. R. Moreyra, B. Hannaford "Stability and Performance of Haptic Displays: Theory and Experiments", Proc. ASME International Mechanical Engineering Congress and Exhibition, Anaheim, CA, pp. 227-34, 1998.
- [16] R.Q. Van der Linde, P. Lammertse, E. Frederiksen, B. Ruiter: "The HapticMaster, a new high-performance haptic interface", Proc. of Euro Haptics 2002
- [17] P.J. Berkelman, R.L. Hollis: Lorentz Magnetic Levitation for Haptic Interaction: Device Design, Performance, and Integration with Physical Simulations, The International Journal of Robotics Research, Vol. 19, Issue 07, 2000.
- [18] 白井暁彦, 長谷川晶一, 小池康晴, 佐藤誠: タンジブル・プレイルーム:『ペンギンホッケー』, 芸術科学会論文誌, Vol.1, No.3, pp.117-124, 2002
- [19] 竹村治雄, 北村喜文, シャネゾンアラン, 岸野文郎: " 仮想現実環境における物体配置タスクの一操作補助手法 ", 映像情報メディア学会誌, Vol.48, No.10, pp.1312-1317, 1994.
- [20] 相山康道, 稲葉雅幸, 井上博允: グラスプレス・マニピュレーションの研究: 操作形態の分類とピボット操作の実現, 日本ロボット学会誌, Vol.14 No.1 pp114-121, 1996
- [21] S. Kim, Y. Koike, M. Sato: "Tension Based 7 DOFs Force Feedback Device: SPIDAR-G", Transactions on Control, Automation, and Systems Engineering Vol. 4, No. 1, March, 2002.
- [22] 吉川恒夫, "ロボット制御基礎論", コロナ社, 1988.
- [23] 佐藤誠, 平田幸広, 河原田弘: 空間インタフェース装置 SPIDAR の提案, 電子情報通信学会論文誌 Vol.J74-D-2,o.7,pp.887-894 1991.
- [24] 川村貞夫, 崔源, 田中訓, 木野仁: パラレルワイヤ駆動方式を用いた超高速ロボット FALCON の開発, 日本ロボット学会誌, vol.15, no.1, pp.82-89, January, 1997.
- [25] 田所諭, 村田理絵, 松島俊之, 村尾良男, 甲川秀明, M.Hiller: パラレルケーブル駆動アーキテクチャによるモーションベースの提案,
日本バーチャルリアリティ学会論文集 Vol.6 No.2
<http://intron.kz.tsukuba.ac.jp/tvrsj/6.2/tr6.2.html>

- [26] A. Nahvi, D. D. Nelson, J. M. Hollerbach, and D. E. Johnson: "Haptic manipulation of virtual mechanisms from mechanical CAD designs", Proc. IEEE Intl. Conf. Robotics Automation, 1998, pp. 375-380.
- [27] R. J. Adams, M. R. Moreyra, B. Hannaford "Stability and Performance of Haptic Displays:Theory and Experiments"
- [28] T. Asano, H. Yano, H. Iwata: "Basic Technology of Simulation System for Laparoscopic Surgery in Virtual Environment with Force Display", Medicine Meets Virtual Reality, IOS Press, pp.207-215, 1997.
- [29] M. Ishii, M. Sato, "A 3D Spatial Interface Device Using Tensed Strings", Presence Vol. 3, No. 1, Winter 1994, pp.81-86
- [30] Force Dimension, <http://www.forcedimension.com>
- [31] PHANTOM Premium 1.0 Technical Specification, <http://www.sensable.com/>
- [32] M.Hirose, K.Hirota, T.Ogi, H.Yano, N.Takehi, M.Saito and M.Nakashige: HapticGEAR: The Development of a Wearable orce Display System for Immersive Projection Displays, IEEE VR2001,2001.03.
- [33] 井上雅晴, 長谷川晶一, 金時学, 佐藤誠: 2次計画法を用いたワイヤ駆動型力覚ディスプレイのための張力計算アルゴリズム, 日本バーチャルリアリティ学会第6回大会論文集, pp.91-94, 2001.
- [34] 井上雅晴, 長谷川晶一, 小池康晴, 佐藤誠: 6自由度ワイヤ駆動型力覚ディスプレイ - 提示力に対する評価 -, 日本バーチャルリアリティ学会第7回大会論文集, pp.63-66, 2002.
- [35] 木野仁, 矢部茂, Chien Chern Cheah, 川村貞夫, 有本卓: パラレルワイヤ駆動システムの作業座標系制御法とそのロバスト性, 日本ロボット学会学会誌, vol.18, no.3, pp.411-418, April, 2000.
- [36] J. P. Desai, R. D. Howe: "Towards the Development of a Humanoid Arm by Minimizing Interaction Forces Through Minimum Impedance Control", ICRA 2001,2001, pp.4214-4219
- [37] C. R. Wagner, O. Clatz, D. P. Perrin, R. L. Feller, H. Delingette, N. Ayache, R. D. Howe: Integrating Tactile and Force Feedback with Finite Element Models, 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.
- [38] Hitoshi Maekawa, JohnM. Hollerbach: Haptic Display for Object Grasping and Manipulating in Virtual Environment, Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pp. 2566-2573, 1998.

- [39] Tsuneo Yoshikawa, Hitoshi Ueda: Haptic Virtual Reality: Display of Operating Feel of Dynamic Virtual Objects, Robotics Research, The Seventh International Symposium, pp.214-221, 1996
- [40] 川井昌之, 吉川恒夫: 内力・外力分離型介在インピーダンスを用いた仮想物体把持の力覚提示, 日本バーチャルリアリティ学会論文集 Vol.6 No.1, 2001.
- [41] 古林 隆: 線形計画法入門, 産業図書, 1980.
- [42] Flanagan, J.R. and WING, A.M.: The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads. The Journal of Neuroscience, 17, 1519-1528, 1997.
- [43] M. A. Otaduy, N. Jain, A. Sud, M. C. Lin: "Haptic Display of Interaction between Textured Models", in Proc. of IEEE Visualization Conference, pp. 297-304, 2004.
- [44] A. M. Okamura, M. R. Cutkosky, J. T. Dennerlein: Reality-Based Models for Vibration Feedback in Virtual Environments ,IEEE/ASME Transactions on Mechatronics, Vol. 6, No. 3, 2001.9, pp-245-252, 2001.
- [45] V.D.Nguyen: "Constructing Force-Closure Grasps in 3D", Proc. 1987 IEEE Int. Conf. on Robotics and Automation, pp.240-245,(1987).
- [46] 川村貞夫, 木野仁, 崔源, 勝田兼: "パラレルワイヤ駆動システムにおけるワイヤ座標系制御法", 日本ロボット学会学会誌, vol.16 No.4, pp.546-552 (1998)
- [47] L. Love, W. Book: "Contact Stability Analysis of Virtual Walls", Proc. of Dynamic Systems and Control Division ASME, pp.689-694 (1995)
- [48] C. B. Zilles, J. K. Salisbury: "Constraint-based god-object method for haptic display", Proc. of International Conference on Intelligent Robots and Systems (IROS), vol. 3, pp.14-151, 1995.
- [49] D. C. Ruspini, K Kolarov, O. Khatib: "The haptic display of complex graphical environments", Proc. of SIGGRAPH 97, pp.345-352 (1997)
- [50] Colgate, J. E., Grafing, P. E., Stanley, M. C. and Schenkel, G., 1993, "Implementation of Stiff Virtual Walls in Force-Reflecting Interfaces," IEEE Virtual Reality Annual International Symposium, Seattle, WA, pp. 202-207.
- [51] R. B. Gillespie, M. Cutkosky: "Stable User-Specific Rendering of the Virtual Wall", Proc. of the ASME International Mechanical Engineering Conference and Exposition, DSC-Vol. 58, Atlanta, GA, Nov 17-22, 1996. pp. 397-406, 1996.
- [52] S. E. Salcudean, J. Yan: Towards a Force-Reflecting Motion-Scaling System for Microsurgery. ICRA 1994, pp.2296-2301, 1994.

- [53] Klatzky, R., Lederman, S. and Reed, C., 1989, "Haptic Integration of Object Properties: Texture, Hardness, and Planar Contour" J. of Exp. Psychology: Human Perception and Performance, Vol. 15, No. 1, pp. 45-57.
- [54] R. B. Gillespie: Haptic Display of Systems with Changing Kinematic Constraints: The Virtual Piano Action, Stanford University, January 1996.
- [55] B. Chang and J. E. Colgate: "Real-time impulse-based simulation for haptic display", Proc. of the 1997 ASME International Mechanical Engineering Congress and Exhibition.
- [56] W. R. Mark, S. C. Randolph, M. Finch, J. M. Van Verth, R. M. Taylor II: "Adding force feedback to graphics systems: issues and solutions", Proceedings of SIGGRAPH96, pp.447-452 (1996)
- [57] W. McNeely and K. Puterbaugh and J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling", Proc. ACM SIGGRAPH 1999.
- [58] B.Mirtich, J. Canny: "Impulse-based simulation of rigid bodies", Proceedings of 1995 Symposium on Interactive 3D Graphics, 1995
- [59] 清川 清, 竹村 治雄, 片山 喜章, 岩佐 英彦, 横矢 直和: "両手操作を用いた仮想物体モデル VLEGO" 電子情報通信学会論文誌 (A), Vol.J80-A, No.9, pp.1517-1526, September 1997.
- [60] D. Baraff: "Interactive simulation of solid rigid bodies", IEEE Computer Graphics and Applications, May 1995, pp. 63-74, 1995.
- [61] E. Guendelman, R. Bridson, R. Fedkiw, "Nonconvex Rigid Bodies with Stacking", SIGGRAPH 2003, ACM Transaction on Graphics, 22, pp871-878, 2003.
- [62] D. Terzopoulos, J. Platt, A. Barr, K. Fleischer: "Elastically deformable models", Computer Graphics 21, SIGGRAPH 87, 1987
- [63] J. Snyder, A. Woodbury, K. Fleischer, B. Currin, A. Barr: "Interval methods for multi-point collisions between time-dependent curved surfaces", Proc. ACM SIGGRAPH 1993, 1993.
- [64] Y. Kim, M. Otaduy, M. Lin, D. Manocha: "Six-degree-of-freedom haptic display using incremental and localized computations", , Presence 12, 3, June 2003.
- [65] D. Baraff: Analytical methods for dynamic simulation of non-penetrating rigid bodies. *Computer Graphics* 23 (1989). (SIGGRAPH 89).
- [Bar94] BARAFF D.: Fast contact force computation for nonpenetrating rigid bodies. *Computer Graphics* (1994). (SIGGRAPH94).

- [66] E. G. Gilbert, D. W. Johnson, 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), pp.193 - 203, 1988
- [67] J. Cohen, M. Lin, D. Manocha, and K. Ponamgi: “I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scaled Environments”, Proc. ACM Symposium on Interactive 3D Graphics, pp. 189-196, 1995
- [68] B. Mirtich: “V-Clip Collision Detection Library”, MERL
- [69] Brian Mirtich: “Rigid Body Contact: Collision Detection to Force Computation”, Workshop on Contact Analysis and Simulation, IEEE Intl. Conference on Robotics and Automation, May 1998
- [70] D. E. Muller, F.P.Preparata: “Finding the intersection of two convex polyhedra”, Theoretical Computer Science, 7(2), pp.217-236 1978
- [71] C. B. Barber, D. P. Dobkin, H. Huhdanpaa : The Quickhull algorithm for convex hulls , ACM Transactions on Mathematical Software , Vol. 22, pp. 469-483, 1996.
- [72] Franco P. Preparata, M. Shamos : 計算幾何学入門 , 総研出版 , pp. 341-356, 1992.
- [73] Russell Smith: Open Dynamics Engine, <http://opende.sourceforge.net/>
- [74] Y. Adachi, T. Kumano, K. Ogino, “Intermediate Representation for Stiff Virtual Objects”, Proc. IEEE VRIS 95, pp.203-210.
- [75] J. Yamashita, C. Yi, Y. Fukui, “Feature-Based Haptic Rendering and Protocol”, Proc. of the Virtual Reality Society of Japan Second Annual Conference, 1997, pp.195-197, 1997.
- [76] Hogan,N., “Stable execution of contact tasks using impedance control”, Proc. of IEEE Robotics and Automation Conference, pp.1047-1054, 1987.
- [77] K. Yamamoto, A. Ishiguro, Y. Uchikawa, “A Development of Dynamic Deforming Algorithm for 3D Shape Modeling with Generation of Interactive Force Sensation”, Proc. IEEE Virtual Reality Annual International Symposium, pp.505-511, 1993.

関連発表

論文

1. 長谷川晶一, 赤羽克仁, 岩下 克, 馬場次郎, 小池康晴, 佐藤 誠: ‘高解像度力覚インタフェースをもつ物理ベース VR システムの開発’, 電子情報通信学会論文誌, Vol.J88-D-I, No.2, 2005 2, 3 章
2. Shoichi Hasegawa, Makoto Sato: ‘Real-time Rigid Body Simulation for Haptic Interactions Based on Contact Volume of Polygonal Objects’, Computer Graphics Forum 23(3): 529-538 (2004), 5 章
3. 長谷川晶一, 藤井 伸旭, 赤羽 克仁, 小池 康晴, 佐藤 誠: ‘力覚インタラクションのための多面体の接触体積に基づく実時間剛体運動シミュレーション’, 計測自動制御学会論文集, Vol.40, No.2, 2004, 5 章
4. 長谷川晶一, 井上 雅晴, 金 時学, 佐藤 誠: ‘張力型力覚提示装置のための張力計算法’, 日本ロボット学会誌, Vol.22, No.5, 2004, 3 章 3.5 節
5. 長谷川晶一, 小池 康晴, 佐藤 誠: ‘フォースディスプレイを用いた剛物体操作のための物体形状と外力の提示方法’, 日本バーチャルリアリティ学会論文誌, Vol.7, No.3, 2002, 4 章
6. 長谷川晶一, 石井 雅博, 小池 康晴, 佐藤 誠: ‘動的な仮想世界の力覚提示のためのプロセス間通信’, 電子情報通信学会論文誌 D-II, Vol.J82-D-II, No.10, 1999, 6 章

国際会議

1. Shoichi Hasegawa, Naoki Okada, Juro, Yuuichiro Tazaki, Hiroshi Ichikawa, Akihiko Shirai, Yasuharu Koike, Makoto Sato: ‘Springhead: Open source haptic software for virtual worlds with dynamics simulations’, Proc. Euro Haptics, 2004, 5 章
2. Shoichi Hasegawa, , Yasuharu Koike, : ‘Real-time Rigid Body Simulation Based on Volumetric Penalty Method’, Proc. of Eleventh Symposium on Haptic Interface for Virtual Environment and Teleoperator Systems, 2003, 5 章
3. Shoichi Hasegawa, Masahiro Ishii, Makoto Sato, Yasuharu Koike: ‘Process division in haptic display system’, Proc. of Seventh International Conference on Parallel and Distributed Systems, 2000 7, 6 章

4. Shoichi Hasegawa, Masahiro Ishii, Yasuharu Koike, Makoto Sato: ‘Inter-Process Communicaton for Force Display of Dynamic Virtual World’, Proc. of the ASME-Dynamic Systems and Control Division-1999, 1999 10, 6 章
5. Shoichi Hasegawa, Yasuharu Koike, Makoto Sato: ‘Real-Time Rigid Body Simulation for Polyhedral Object Based on Accurate Penalty Method’, Proc. of 6th International Conference on Motion and Vibration Control, 2003, 5 章
6. Shoichi Hasegawa, Masahiro Ishii, Makoto Sato: ‘Haptic Interface Protocol for Complex Dynamic Virtual World’, Proc. of International Conference on Artificial Reality and Tele-Existence, 1998, 6 章

研究会

1. 長谷川晶一, 石井雅博, 佐藤 誠, 小池康晴: ‘力覚ディスプレイをもつ動的な仮想世界のための力積に基づくプロセス間通信’, 3D 映像, Vol.13, No.2, 1999 5, 6 章
2. 赤羽克仁, 長谷川晶一, 小池康晴, 佐藤 誠: ‘力覚クオリティ向上を目指した高解像度ハプティックコントローラの提案’, 信学技報 MVE2003-113 ~ 120, No.663, 2004 2, 3 章
3. 井上雅晴, 長谷川晶一, 金 時学, 佐藤 誠: ‘ワイヤ駆動型力覚ディスプレイのための提示張力の計算方法’, 信学技報 MVE2001-1 ~ 17, Vol.101, No.109, 2001 6, 3 章