

ハプティックデバイスによる操作感提示と
その軌道計画に関する研究

2010年

矢船 資洋

要旨

本論文は、ハプティックデバイスによる操作感提示とその軌道計画に関する研究について書かれたものである。ユーザインタフェースの把持部や接触部において、位置や力の情報を基に入出力を行う装置をハプティックインタフェースというが、ハプティックインタフェースの操作感は人間の感性に依存する部分も大きく、従来は官能評価に頼らざるを得ず、試作を繰り返すしかなかった。ハプティックインタフェースにおいて、操作感の定量的表現やそれに基づく系統的設計手法が求められている。また、VR(Virtual Reality) 技術の発展に伴い、試作の手間やコストを減らすためにバーチャル空間で試作評価を行うバーチャルプロトタイピング (virtual prototyping) が注目されている。ハプティックインタフェースは力やトルクを出力する機械であり、ある種のロボットであるとも言える。その装置は一般にハプティックデバイスと呼ばれるが、そのデバイスを適切に動作させるためには、その軌道を定める動作計画や軌道に沿って動作させるための制御などが必要となる。それらを行う方法として最適軌道計画や最適制御が挙げられる。

本論文は、ハプティックインタフェースの操作感の定量的表現や系統的設計手法を実践的に考えるために、第2章で実際的な対象として自動車のコラム軸アシスト型電動パワーステアリング装置 (C-EPS) を挙げて、C-EPS の準静的な操舵角-トルク特性に着目し、C-EPS において目標とする操舵角-トルク特性を実現するアシスト制御手法を提案する。提案手法により目標の特性を実現するアシスト制御マップを自動的に生成することが可能になる。

また、機器と人間のインタフェースにおいて、人との親和性や操作容易性などを評価する目的で近年ユーザビリティ評価などが注目を集め、操作性を考慮したインター

ス設計の重要性が増している。そこで、製品のプロトタイプ段階で実際に試作する手間やコストを軽減する為に、VR 技術を応用して、実際のモックアップの代わりにバーチャル空間内にバーチャルモックアップを構築して試用評価を行うことが考えられる。第 3 章では、様々なスイッチ類が配置された操作パネルを対象として、バーチャル空間内に構築された操作パネル上の複数のスイッチを操作する感覚を 1 つの遭遇型ハプティックデバイスで提示するシステムを構築する。最近接点情報に基づく動作計画手法を当該システムに適用した。その中で、スイッチボックスの設計及びデバイスの動作計画を見通しよく行えるようにスイッチの分類を行い、また、スイッチによってはその法線軸回りに配置の任意性を有する場合があります、その任意性を利用してデバイスの動作負担を軽減する提示姿勢の決定手法を提案した。さらに、デバイスの動作負担を軽減するために、動作計画で目標提示位置・姿勢の計算に用いる重み関数の改良を行い、また、バーチャルパネル面座標系を導入して、デバイスがバーチャルパネル面に沿って動くようにすることでユーザとの不用意な衝突が起こらないように動作計画の改善を行った。

最適制御問題に Pontryagin の最小原理を適用した場合、一般に常微分方程式の 2 点境界値問題に帰着されるが、第 4 章では問題に最小原理を適用して随伴変数を用いて媒介変数表示し、直接解法により厳密に解く手法を提案する。提案手法は従来手法に比べて解の精度や求解に掛かる計算時間の面で優れている。1 次元の最適制御問題と見なせる最小ドラッグノーズシェイプ問題を対象に実際に求解した例を挙げて、提案手法と従来手法について比較を行った。また、提案手法を実際のロボット・多リンク機構系などの機械システムに適用するための式展開を示し、1 自由度ロボット・リンク系において提案手法を適用した数値例を挙げてより実際的な問題における有効性を示す。

第 5 章では、機械システムにおいてリアルタイム準最適制御を行うために、1 次展開最適ゲインと名付けたゲインを導出し、それによりリアルタイム準最適制御を行えることを示す。第 4 章でも用いた 1 自由度ロボット・リンク系の例を対象として、1 次展開最適ゲインを求める数値例を示し、1 次展開最適ゲインを用いたリアルタイム準最適制御の数値シミュレーションを行って、提案手法の有効性を示す。

目次

要旨	i
第 1 章 序論	1
1.1 背景	1
1.2 電動パワーステアリング装置における操作感	4
1.3 遭遇型ハプティックデバイスとバーチャル操作パネル	5
1.4 機械システムにおける軌道計画問題	8
1.5 機械システムにおける最適軌道計画問題	9
1.6 最適制御問題の一般解法	10
1.7 本論文の構成	13
第 2 章 コラム軸アシスト型パワーステアリング装置における望ましい準静的 操舵角-トルク特性の実現	15
2.1 緒言	15
2.2 C-EPS のモデル化と解析	16
2.3 アシスト制御式の構成	19
2.4 目標の特性を実現するアシスト制御設定法	22
2.5 シミュレーションと実験による検証	27
2.6 結言	29
第 3 章 遭遇型ハプティックデバイスを用いたバーチャル操作パネル上の複数	

	スイッチ提示システム	33
3.1	緒言	33
3.2	バーチャル操作パネル提示システムとデバイスの動作計画	34
3.3	提案手法の検証	49
3.4	結言	58
第4章	最小原理を応用した機械システムの最適軌道計画	61
4.1	緒言	61
4.2	最適制御問題の随伴変数初期値最適化による解法	62
4.3	提案手法と従来手法の比較	69
4.4	機械システムへの随伴変数初期値最適化法の適用	73
4.5	結言	87
第5章	1次展開によるリアルタイム準最適制御の実現	89
5.1	緒言	89
5.2	1次展開によるリアルタイム準最適制御の実現	90
5.3	1次展開最適ゲインの適用と検証	94
5.4	結言	106
第6章	結論	107
	参考文献	109
	関連業績一覧	115

第 1 章

序論

1.1 背景

本論文ではハプティックデバイス (haptic device) による操作感提示について論じるが、まずはじめに研究背景として装置の操作感について述べる。自動車や航空機、建設機械といった旅客・運輸・作業などを行う機械類において、一般にレバーやジョイスティック、ステアリングなどの操縦装置を介して機械を操ることになる。それらの操縦装置は、機械類の運行・作業状況を視聴覚で認識・判断しながら、装置の把持部や接触部にユーザが力を加えて作動させ、その位置や反力を基に機械類の入力を制御するように設計されている。このように力や位置の情報を用いる操縦装置は、一般にハプティックインタフェース (haptic interface) とも呼ばれる。また、機械類や化学プラント、発電設備などでは、電子機器を用いて装置の安定化や高性能化が行われていることが多く、電子機器への指示は操作盤のスイッチを用いて行われる。スイッチも位置情報を基に入力を行うものであり、広義のハプティックインタフェースであると言える。機器類のユーザインタフェースは、機器類の操作容易性や信頼性、人との親和性において重要であり、機器類自体の設計と同様に、そのユーザインタフェースの設計も非常に重要である。ハプティックインタフェースの操作感是人間の感性に依存する部分も大きく、従来は、官能評価に頼らざるを得ず、試作を繰り返すしかなかった。ハプティックインタフェースにおいて、操作感の定量的表現や、それに基づく系統的設計手法が求められている。ま

た近年，VR(Virtual Reality) 技術の発展に伴い，試作の手間やコストを減らすためにバーチャル空間で試作評価を行うバーチャルプロトタイピング (virtual prototyping) が注目されている．

次に VR について概要を述べる．計算機を用いて構築され人間の感覚器官を対象としたバーチャル世界の始まりはコンピュータグラフィックス (Computer Graphics, CG) であったと言えるだろう．初期のコンピュータは数値やテキスト中心の入出力であり，専門技師のみが訓練を行って扱えるものであり，一般ユーザにとっては取っ付きにくいものであった．演算処理速度の向上に伴って CG 技術も発展し，データの視覚化が行われて出力を目で直感的に確認できるようになり，ユーザインタフェースが向上して，一般ユーザもコンピュータの恩恵を受けることができるようになった．

ハプティックインタフェースを用いた最初期の VR としては，1990 年に発表された Brooks ら [1] による Project GROPE が挙げられる．そのシステムでは，新薬開発や分子モデル研究のために計算機内で分子モデルを構築し，映像を視覚的に確認しながらハプティックデバイスを用いてモデルを操作することができ，計算された分子間力を基にハプティックデバイスを介してユーザにモデルを操作する反力が提示される．また，同じ 1990 年，Minsky ら [2] によって Virtual Sandpaper が発表された．その装置では，3 自由度ジョイスティックによってバーチャルなテクスチャの触感を提示する試みがなされた．岩田ら [4] も，同年，卓上で動作するハプティックデバイスを発表した．1994 年には，Massie ら [5] によって PHANToM が発表された．SensAble 社によって以後現在までに渡って様々な改良モデルが製造され，卓上ハプティックデバイスのベストセラーとなっている．装着型のハプティックデバイスとしては，Burdea ら [3] によって 1991 年に発表されたグローブ型のものが挙げられる．遭遇型のハプティックデバイスは，1993 年に McNeely[6] によって提案された．

黎明期の VR(Virtual Reality) システムは，主に HMD(ヘッドマウントディスプレイ) とデータグローブ，磁気センサによって構成されており，建築・インテリアデザインの検討などを用途として想定していた．それらのシステムでは，ユーザの頭部位置・姿勢を磁気センサにより計測し，その位置・姿勢に応じて HMD に両眼ステレオ映像が

映し出される．データグローブと磁気センサにより手の動きを検出することで操作を入力することができ，ジェスチャ入力を行うことで VR 空間内を移動したり，バーチャル物体を操作して VR 環境とインタラクションしたりすることができる．また，その当時からバーチャル空間内でユーザ同士がインタラクションを行うシステムが開発されており，VR システムについてコミュニケーション空間あるいはメディアとしての機能も指摘されている．今後，視聴覚のみでなく力覚インタラクションを伴う VR 技術がより現実的に求められることとなり，ハプティックインタフェースに関する研究の重要性は増すであろう．

また，ロボティクス分野において遠隔操作 (teleoperation) の概念が発展し，テレプレゼンス (telepresence) という概念が生み出された．単に遠隔環境でロボットを介して操作するのではなく，遠隔環境に居るかのように五官で感じながら実環境とインタラクションできる，いわば臨場感の高いシステムを実現しようという指向である．それにより，複数の感覚器官を用いて環境からより多くの情報が適切に得られるようになり，環境内での認識・判断やインタラクションの操作性が向上すると考えられている．遠隔操作やテレプレゼンスにおいても，ハプティックインタフェースの操作感は重要である．

ハプティックインタフェースは，力やトルクを出力する機械であり，ある種のロボットであるとも言える．その装置は一般にハプティックデバイスと呼ばれるが，そのデバイスを適切に動作させるためには，その軌道を定める動作計画や，軌道に沿って動作させるための制御などが必要となる．それらを行う方法として，最適軌道計画や最適制御が挙げられる．そこで次に，最適軌道計画や最適制御の研究背景について簡単に述べる．

最適軌道計画問題は，古くは変分問題として知られており，関数空間における非線形計画法を適用して関数漸化式の形で定式化する解法がある．関数空間における非線形計画法としては勾配法や逆掃法，共役勾配法などがあり，多くの亜種改良解法が提案されている．また，よく知られている Bellman が提案した動的計画法は，最適性の原理に基づき変分問題を偏微分方程式の境界値問題に帰着させるものである．一般には，その偏微分方程式の境界値問題を差分法により解く．もう一つのよく知られた解法として，Pontryagin の最小原理により，常微分方程式の 2 点境界値問題に帰着する方法がある．

その他に、関数近似を用いた直接解法として、Ritz 法や B スプラインを用いる方法が知られている。

1.2 電動パワーステアリング装置における操作感

自動車業界において車載装置の IT 化・知能化が進む中、より高性能な電動パワーステアリング装置 (Electric Power-assist Steering system, EPS) の開発が望まれている。しかし、近年では低コスト化や環境を考えた省エネルギー・高効率化など様々な開発上の制約を考慮した、トータルな視点での製品開発が求められている。本論文では、EPS の中でも低コスト化や省エネルギー・高効率化の点で今後より重要となるコラム軸アシスト型電動パワーステアリング装置 (Column-type Electric Power-assist Steering system, C-EPS) において、望ましい操舵感を実現する設計・制御手法を開発することを目的とする。従来のコラム軸アシスト型ではない EPS の制御に関する研究 [7] や、ステアリング装置の操舵感・操舵特性に関する研究 [8][9][10] はいくつかなされているが、C-EPS において機械的特性に踏み込んだ解析やアシスト制御に関する研究はほとんどなされていないのが現状である。

EPS は単に操舵を軽くするだけでなく、ドライバーに対して自然な操舵感を提供できるようにアシスト量を決める必要がある。また、近年研究が進められているバイワイヤ方式 [11][12][13] では、操舵量に対する実転舵量の対応付けや望ましい操舵感提供の点で大幅に設計自由度が増えるが、その際に適切な操舵特性や実転舵量を決定するための枠組みは、人間の感性に依存する部分も大きく、確立されていない。

ステアリング装置の操舵特性には、大きく分けて準静的特性と動的特性という二つの特性がある。準静的特性は主にゆっくり操舵したとき現れる大域的な特性であり、動的特性は主に素早く操舵したときに現れる周波数領域で解析されるような特性である。準静的特性は、操舵角に対して操舵トルクをプロットした図 1.1 のようなグラフで示される操舵角-トルク特性で表すことができる。これまで、ステアリング装置の設計・調整手法として望ましい操舵特性を実現する系統的手法はなく、経験的にアシスト制御マップや制御パラメータを与えて操舵特性を調整していたので、特性のある面を改善すると別

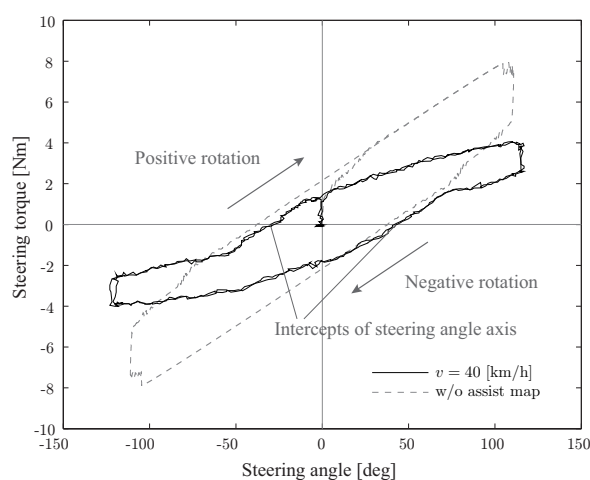


図 1.1 An example of steering angle-torque characteristics on an experimental bench

の面が悪化することもあった。また，制御や設計上のパラメータと操舵特性の因果関係が明らかにされていないため調整の効率が悪くなり，望ましい特性を実現できない場合もあった。

以上のような背景から，第 2 章では準静的な操舵角-トルク特性に着目し，C-EPS において目標とする操舵角-トルク特性を実現するアシスト制御手法を提案する。提案手法により目標の特性を実現するアシスト制御マップを自動的に生成することが可能になる。

1.3 遭遇型ハプティックデバイスとバーチャル操作パネル

近年，バーチャルな物体に触れる感覚を提示するデバイスの研究・開発が多く行われ，それに伴ってバーチャルリアリティ (Virtual Reality, VR) 技術のより広範な分野での応用が進むと期待されている。機器と人間のインタフェースにおいて，人との親和性や操作容易性などを評価する目的で近年ユーザビリティ評価などが注目を集め，操作性を考慮したインタフェース設計の重要性が増している。製品のプロトタイプ段階で VR 技術を応用し，実際のモックアップの代わりにバーチャル空間内にバーチャルモックアップを構築して試用評価を行うことが考えられる。

第3章では、様々なスイッチ類が配置された操作パネルを対象とする。操作パネル上のスイッチ類をVRとして視覚的のみならず力覚的に提示するシステムを構築することができれば、実際にモックアップを製作する手間やコストをかけることなく、様々な操作パネルを効率よく評価することが可能になる。また、用途によっては実際の操作パネルを作成することなく、バーチャル操作パネルを直接に対象機器の操作に用いることも可能であり、トレーニングへの応用も考えられる。大島ら [19] はデザイン評価への応用などを目的とした複合現実感 (Mixed Reality, MR) システムにおいて、部分的なモックアップによる触覚提示とCG・実写の合成による視覚提示を精度良く重畳することで、自動車のインテリアを視触覚で違和感なく体感できるシステムを開発した。そのシステムでは提示したい物体と同形状の樹脂製モックアップによって操作系の触覚提示を行っているため、触覚提示物の形状や場所を自由に変更することはできない。触覚提示物をハプティックデバイスで提示できるようになれば、バーチャルモックアップとしての応用性が高まると考えられる。

力覚を提示する装置は一般にハプティックデバイスと呼ばれるが、力覚の提示手法によって、ユーザの手に装着されたデバイスにより力覚提示を行う「装着型」、ユーザにデバイスの一部を把持させて力覚提示を行う「把持型」、ユーザが触れようとしているバーチャル物体表面の位置にあらかじめデバイスを待機させ、ユーザがデバイスに遭遇することで力覚提示を行う「遭遇型」に分類される [18]。把持型及び装着型ではユーザは常にデバイスに触れていなければならないが、遭遇型は「バーチャル物体の存在する場所では実際に物体に触れることができるが、バーチャル物体が存在しない場所では実際に何も触れることがない」という理想的な力覚を表現でき、またデバイスによるユーザへの装着負担も少ない。しかし、その一方で遭遇型のデバイスを理想的に動作させるためには、バーチャル環境内でユーザが物体に触れる前にデバイスを提示する目標位置・姿勢で待機させてユーザと遭遇できるようにしなければならない。またバーチャル物体のない空間ではデバイスとユーザの身体との不用意な接触を回避しなければならないので、適切に遭遇と回避を行えるようなデバイスの動作計画を考案する事が必要不可欠である。

遭遇型ハプティックデバイスの構想は Boeing 社の McNeely [6] によって提案され、

バーチャル操作パネルに配置されたスイッチやボタンを提示するシステムが開発された。舘ら [16] もバーチャル空間内の物体を提示する遭遇型の力覚提示システムを同時期に提案している。岩田ら [17] は足裏に対してバーチャルな接地面を力覚的に提示し、装置の上で平地歩行や階段昇降などの歩行シミュレーションが可能な遭遇型の力覚提示システムを開発した。横小路ら [18] は CG による視覚提示と遭遇型ハプティックデバイスによる触覚提示を組み合わせ、自動車のインパネ (instrument panel : 計器盤) 設計の操作性評価を想定して、スイッチの提示を行う遭遇型ハプティックデバイスとして用いたロボットアームの動作計画アルゴリズムの提案を行った。また、Provancher ら [20] は、卓上力覚提示装置の先端に取り付けられた円筒形の提示部を指先と遭遇・接触させることにより、任意の曲率部分を有する柱状のバーチャル物体との接触を指先に対して力覚的に提示可能なデバイスを開発し、知覚実験により物体の曲率や転がりなどの運動の提示に有効であることを検証した。

横小路ら [21] はこのような遭遇型の概念を発展させ、拇指・示指・中指の 3 本指による把持を模擬する事が可能な多指遭遇型ハプティックデバイスを提案し、重田ら [22] はこの多指遭遇型ハプティックデバイスに対して、指からバーチャル物体に対する最近接点の情報に基づく動作計画法を提案した。ただ、動作計画において先端のコンタクトモジュールの動きのみに着目していたため、それに結合されているベースモジュールに相当するロボットアームの動きに無理が出ることが、シミュレーションによる検証によって問題点として明らかになっていた。また、中河原ら [23] はトレイグジスタンス用途に、指先に対して力覚提示が可能な多指遭遇型のマスタハンドシステムを開発している。

横小路ら [18] が提案した手法は、インパネ上の各スイッチごとに 1 つの代表点とそこでの提示姿勢を定めておき、個々のスイッチの代表点を結んでできる多面体を基に周辺の空間を Voronoi 領域に分け、ユーザの指がその中のどの領域にあるかによって効率の良いデバイスの動作を決定するものであったが、構成した多面体の内部にもスイッチがあるときにデバイスの動作が不連続になるという問題があり、また提示できるスイッチは 1 種類のみであった。第 3 章で、1 種類のスイッチのみ提示可能であった横小路ら [18] のシステムを拡張し、バーチャル空間内に構築された操作パネル上の複数のスイッ

チを操作する感覚を 1 つの遭遇型ハプティックデバイスで提示するシステムを構築する．筆者らは既に，重田ら [22] が多指遭遇型ハプティックデバイスで用いるために提案した最近接点情報に基づく動作計画手法を基に，スイッチ提示システムで用いる遭遇型ハプティックデバイスの動作計画手法を提案してきた [24] ．

1.4 機械システムにおける軌道計画問題

第 4 章は，ロボットなどの機械システムにおける最適軌道計画問題を取り扱い，目的を達成する際のシステムの挙動を最適化することを主題としている．また，第 5 章は，実システムにおいてなるべく最適性を損なうことなくその軌道を実現するために，リアルタイム準最適制御の実現を目指したものである．

システムを現代制御論の枠組みで捉え，状態空間における方程式としてモデル化を行い，軌道計画問題を最適制御問題として定式化し，最適制御問題を解くことにより最適な軌道を得る．一般の機械システムを対象とするので，システムは非線形，あるいは時変系として扱う必要がある．それにより，最適制御理論の最もよく知られた結果である線形時不変系における定常最適レギュレータのように，解析的に単純な解が求まるということではなく，通常は計算機により最適化を行った結果，数値解として最適軌道が得られることとなる．

機械システムを対象としてこれまで行われてきた最適軌道計画問題の研究について 1.5 節で詳しく述べるが，これまでロボット分野において，殊にロボット・マニピュレータに関して最適軌道計画問題について多くの研究がなされてきた [25][26][27][28][29] ．また，高度な機械システムである航空機や，空気力を利用する宇宙船である AOTV (Aeroassisted Orbital Transfer Vehicle) について，姿勢を含む軌道を計画する問題は航空宇宙工学の分野における大きな課題であり，最適制御理論の重要な応用範囲となっている [35] ．1.5 節で挙げたような機械システムにおける最適軌道計画問題の研究に用いられている最適制御問題の解法は，関数空間における非線形計画法，B スプラインを用いた準最適化解法，Shooting Method の 3 つに概略される．

1.6 節で最適制御問題の一般解法について詳細に述べるが，最適軌道計画問題で対象

とする有限時間区間最適レギュレータ問題の、従来知られている一般解法は大きく4つに分けられ、第1に変分法の発展として動的計画法、第2に変分問題として扱い関数空間において非線形計画法を適用する方法、第3に直接解法、第4として最小原理により常微分方程式の2点境界値問題を解くことに帰着する方法がある。1.6節において述べられているように、計算機で解く段階を含めて考えて大きく分けると、従来の手法は結局、問題を離散化して解いているか、パラメータで表現して準最適解を得ているかの2種類に大別できる。

数値解法として問題になるのは計算に要する時間と得られる解の精度であるが、パラメータで表現して準最適解を得た場合、得られる解は準最適解であり解の最適性という点で精度はよくない。問題を離散化して解いた場合、離散化の時間ステップが粗いと解は不連続となり精度はよくない。また、解の精度をよくしようとすると、時間ステップを細かく取ることとなり、問題の次元数と相俟って爆発的に膨大な計算時間を要し、また解の保存に必要な記憶容量も膨大となり、問題のサイズが大きくなると求解は実際的でなくなる。

1.5 機械システムにおける最適軌道計画問題

近年、ロボット分野において、殊にロボット・マニピュレータに関して最適軌道計画問題について多くの研究がなされている。栗栖ら [25] は、マニピュレータによる押し作業における軌道計画において、最適制御問題として定式化し、その変分問題を関数空間における共役勾配法 (Fletcher-Reeves 法) により数値的に解いた。尾崎ら [26] は軌道を B スプラインを用いて表し、そのパラメータを最適化変数として評価関数を直接最適化することにより軌道のいわば準最適化を行い、最適化問題の解法として評価関数の勾配が不要なコンプレックス法を適用し、その結果を勾配を用いた解法による結果と比較検討した。泉ら [27] は、マニピュレータの重力下での PTP 制御において、消費エネルギーが最小となるように動作時間の最適化を行った。状態方程式を平衡点回りで線形化し、ハミルトン関数により正準方程式を導き、最適動作時間を解析的に検討した。また、厳密な数値解を、線形化したシステムにおける値を初期値とした Shooting Method に

より求めている．岩村らは非ホロノミック運動計画問題において，軌道を B スプラインにより表して準最適化 [28] を行った．また，軌道に複数の拘束条件がある場合の最適制御問題の解法として「階層勾配法」なる数値計算法を提案し，非ホロノミック系に適用 [29] している．非ホロノミック系に変分的解法を適用した場合，収束性が著しく悪化することが指摘されており，収束性を向上するために，制御時間区間を分割して初期状態と目標状態の双方向から軌道を生成し，ある中間点で接続を行って，問題の分割と連結を行うことにより，「双方向軌道計画法」としている．また，軌道を繰り返し最適化により求める際の収束性に関して，可操作性の考えに基づいた評価指標を導入し，数値計算の収束性を測る目安として用いている．竹内ら [30] により，リアルタイム向きな最適制御問題の解法，Receding Horizon Control と連続変形法の機械システム系への適用が行われた．また，竹内 [31] は等式拘束条件付きの場合の Receding Horizon Control の定式化を行い，非線形で複雑な機械システム系である脚式ロボットに適用し，目標 ZMP 軌道のリアルタイム生成を試みている．

1.6 最適制御問題の一般解法

最適制御問題は，システムの状態ベクトル，入力ベクトル，出力ベクトルから構成される評価関数を最小とする入力を求める最適化問題として一般に定式化されている．最適制御問題には大きく分けて 2 つの場合があり，有限時間区間制御問題と無限時間区間制御問題がある．最適レギュレータ問題は，状態ベクトルあるいは出力ベクトルをある目標値に漸近させることが目的であり，有限時間区間問題と無限時間区間問題の両方あり，最適制御入力あるいは最適制御則を得ることが目的である．最適トラッキング問題は有限時間区間制御問題であり，有限な時間区間で与えられたある軌道に追従するための最適制御入力を求めることが目的である．線形時不変システムにおいて，最適状態フィードバック則として得られる定常最適レギュレータが，最適制御理論の成果としてよく知られている．以下では特に断らない限り，有限時間区間最適レギュレータ問題について述べる．

最適制御問題は，入力を時間の関数とした関数空間における最適化問題であるので，

古くは、システムの状態、あるいは入力を時間関数とした関数空間における変分問題として扱われ、それに対して様々な解法が考案されてきた。最適制御問題一般の解法は、文献 [32][33][34] に詳しいが、大きく分けて 4 種類ある。1 つは変分法の発展として動的計画法。第 2 に変分問題として扱い関数空間において非線形計画法を適用する方法。第 3 に直接解法がある。第 4 として最小原理により常微分方程式の 2 点境界値問題を解くことに帰着する方法が挙げられる。

動的計画法は、最適性の原理と呼ばれる原理に基づき、変分問題を偏微分方程式の境界値問題に帰着させる。境界条件は制御時間区間終端において与えられ、偏微分方程式を逆時間方向に解く必要がある。システムの特徴が状態に依存する場合、逆時間方向に解いて初期条件を満たすことは解が既知ということであり、問題として矛盾を抱えている。偏微分方程式の境界値問題の数値解法は一般に差分法が知られている。

変分問題の解法としての関数空間における非線形計画法は、勾配法や逆掃法といった、変分を近似して得られる関数漸化式により漸化的に最適制御入力を得る方法である。変分を求めるために仮の制御入力より得られる軌道に沿って、幾つもの微分方程式を逆時間方向に解く。微分方程式の求解を行うために時間差分をとり、多くの変数値を記憶する必要がある。得られた変分を関数漸化式に代入して仮の制御入力を更新し、最適制御入力を漸化的に求める。勾配法は 1 次の変分しか考慮していないため、収束性に関して問題がある。また、変分解法の限界から、変分を小さい値に保つためと変分に対する拘束条件により、評価関数の変分を定義し直す必要がある。逆掃法は制御区間終端時刻より逆時間方向に積分することからついた名前であるが、変分係数の変分、つまり 2 次の変分を考慮するので最適解近傍で優れた収束性を示す。しかし、2 次の変分係数が正則となる必要があり、そのような条件を満たす仮の制御入力の初期値を見つけることが必要となる。

古典的変分法あるいは関数解析によって最適性の条件を導出し、それにより関数空間における勾配を導いて、最適制御問題を関数空間における非線形計画問題として詳しく扱った文献として [34] が挙げられる。関数空間における共役勾配法はある程度成果を挙げ、制約条件が与えられたときにもクリッピング法やペナルティ法を用いて数値的に解

くことが可能である [34][25]。しかし、拘束条件が多数加えられた場合にそれらが干渉して数値計算が不安定となり、また、変分的方法による解法を非ホロノミック系に適用した場合には収束性が著しく悪化することが指摘されており、そのような場合に適用できる一数值解法が、1.5 節でも触れた「階層勾配法」[29]として提案されている。

拘束条件のある最適制御問題を古典的変分法により付帯条件付き変分問題として扱って、詳しく述べた文献として [35] が挙げられる。文献の中でその優れた数値解法として、Sequential Conjugate-Gradient-Restoration Algorithm (SCGRA) [37][38] 及び Modified Quasilinearization Algorithm (MQA) [39][40] が紹介され用いられている。

変分問題の直接解法として Ritz 法が知られているが、それは状態あるいは入力の変関数を基底関数の線形和で表現し、その係数を最適化変数として評価関数を直接最適化する方法である。高精度の解を得るためには多くの基底関数が必要であり、最適化変数の数が増えて計算量も多くなる。

評価関数を直接最適化する方法として、軌道あるいは入力を区分多項式 (B スプライン) で表現し、そのパラメータを直接最適化する方法がある。ただし、区分多項式で表現された軌道は評価関数を厳密には最適化せず、準最適な軌道となる。軌道を区分多項式で表現した場合、制御入力逆システムを解くことにより得られる。

線形システムにおいては、動的計画法により得られる偏微分方程式の求解が、ある仮定により、常微分方程式の 2 点境界値問題を解くことに帰着される。あるいは最適性の必要条件である最小原理により同じ常微分方程式が得られる。常微分方程式の 2 点境界値問題の一般に知られている数値解法として緩和法 (Relaxation Method) 及び射撃法 (Shooting Method) [41] がある。また別な解法として有限要素法 (Finite Element Method, FEM) [42] が挙げられる。

変分に拠らない方法として、システムを離散時間系として表現し、動的計画法を適用する方法がある。離散時間システムとして表現した場合、多段決定過程となり、最適性の原理により逆時間方向に解くことができる。しかし、状態ベクトルの次元が大きくなるに従い、膨大な記憶容量と計算量が必要となる。また、軌道の精度を高めるために時間の分割数を多くした場合にも同様の問題が発生し、次元が高くなるにつれて爆発的に

計算量が増大する事実を、動的計画法を発見した Bellman は「次元性の呪い」と呼んだ。次元性の呪いを回避する方法として、状態増分動的計画法が知られている。状態増分動的計画法では、時間の増分を可変とすることで、時間変化による状態の遷移を常に近傍に限ることで計算範囲を限定し、それにより必要な記憶容量が軽減される。

離散時間系における動的計画法、あるいは微分方程式の差分法により、離散時間で表現された最適軌道、最適制御入力を得ることの問題点として、細かい時間分解能の軌道あるいは入力を得るためには、時間差分を多くとる、あるいは解の補間を行う必要があり、離散時間系や時間差分を用いて連続時間のシステム方程式を満たす高精度の最適解を簡単に得ることはできない。

これまで用いられてきた方法は計算機で解く段階では3種類に大別でき、第1に離散時間系の問題として表現する、あるいは差分法を用いて解く方法、第2に関数空間における非線形計画法を適用する方法、第3にパラメータで表現して直接最適化を行い準最適解を得る方法の3つに分けられる。第2の関数空間における非線形計画法についても実際は離散化されているので、第1の方法と同様であるということができ、従来の手法は結局、問題を離散化して解いているか、パラメータで表現して準最適解を得ているかの2種類に大別できる。

数値解法として問題になるのは計算に要する時間と得られる解の精度であるが、パラメータで表現して準最適解を得た場合、得られる解は準最適解であり解の最適性という点で精度はよくない。問題を離散化して解いた場合、離散化の時間ステップが粗いと解は不連続となり精度はよくない。また、解の精度をよくしようとすると、時間ステップを細かく取ることとなり、問題の次元数と相俟って爆発的に膨大な計算時間を要し、また解の保存に必要な記憶容量も膨大となり、問題のサイズが大きくなると求解は実際的でなくなる。

1.7 本論文の構成

本論文は、力覚的に操作感を提示するデバイスとその制御手法を扱うものである。第2章では、実環境中の操作対象機器として自動車を挙げ、力覚インタフェースであるス

テアリング装置において望ましい操作感を提示する手法を述べる．第 3 章では，バーチャル環境中の機器操作パネルを提示する装置とその制御について述べる．バーチャル操作パネルの提示には遭遇型ハプティックデバイスを用いるが，デバイスは複数の実スイッチを備えており，ユーザが指先でバーチャル操作パネルを操作しようとしたときに，バーチャルスイッチの位置において指先と実スイッチが適切に遭遇することによって，バーチャルスイッチを操作する感覚が提示される．デバイス上の実スイッチに実際に触れることによって当然スイッチに触れる感覚が提示されるが，バーチャルスイッチの位置でユーザとデバイスが適切に遭遇し，バーチャル物体のない空間でユーザとデバイスが衝突しないようにしなければならない．また，スイッチの種類によっては提示姿勢に法線回りの任意性があり，その任意性を利用することによってデバイスの動作負担を軽減できる．本章で，適切に遭遇と回避を行え，また，デバイスの動作負担が小さく効率的であるデバイスの動作計画を提案する．第 4 章では，デバイスの制御手法について考える．そこで，デバイスの軌道制御を最適軌道計画問題として捉え，その一般解法について考える．従来手法に比べて，解の精度と計算時間に優れている最適軌道計画問題の一般解法を提案する．第 5 章では，最適軌道計画問題を制御に用いる場合のことを考え，計算のリアルタイム性を実現するために，高速に計算が可能な時変ゲインを用いる準最適制御手法を提案する．

第 2 章

コラム軸アシスト型パワーステアリング装置における望ましい準静的操舵角-トルク特性の実現

2.1 緒言

1.2 節で述べたような背景から，本章では C-EPS の準静的な操舵角-トルク特性に着目し，C-EPS において目標とする操舵角-トルク特性を実現するアシスト制御手法を提案する．提案手法により目標の特性を実現するアシスト制御マップを自動的に生成することが可能になる．この問題に対する取り組みについては著者らによる既報 [14] でも述べているが，本章では提案手法についてより詳細で正確な導出と検証結果を示す．

以降では，2.2 節において C-EPS の準静的な操舵角-トルク特性をシミュレーションによって再現できる等価三慣性モデルを構築し，そのモデルについて準静的な解析を行い，その考察から 2.3 節でクーロン摩擦及び弾性変形の影響を補償して目標の特性を実現するアシスト制御手法を提案する．また，2.4 節において提案するアシスト制御で用いるアシスト量を決めるマップの生成手法を導出して示す．2.5 節において提案手法の数値例として実際にマップの生成を行い，シミュレーション及び実験ベンチによって，

目標の操舵角-トルク特性が実現できることを検証した。

2.2 C-EPS のモデル化と解析

2.2.1 モデルの定式化

一般的な C-EPS は図 2.1 に示すようなシステムになっている。ドライバが入力する操舵トルク T_w を検出するために、ハンドル部とコラム部の間にトルクセンサが設けられ、その検出トルク T_s と車速信号 v を用いて、ECU において制御演算を行い、コラム部に設けられたアシスト用電動モータに対して駆動信号の計算・出力を行う。電動モータにおいて発生したモータトルク T_m はウォーム減速器を介してアシストトルク T_a としてコラム軸に伝わる。電動モータにおいて指令したトルクを正確に発生させるために、電流センサによってモータ電流 I_m を検出して電流フィードバックを行っている。コラム部のトルクは、インターミディエイトシャフトを介してピニオン軸を回転させてラックを駆動し、タイロッドを左右に動揺させ、ナックルアームを介してタイヤの舵角を変更する。

本研究では、文献 [15] に見られるようなハンドル部、コラム軸等価部、ピニオン軸等価部の三つの代表等価慣性で構成される簡略化したモデル化を行う。等価変換した簡略

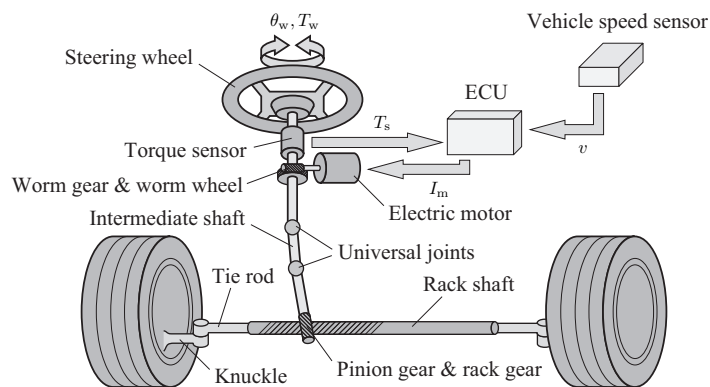


図 2.1 Column-type electric power-assist steering system

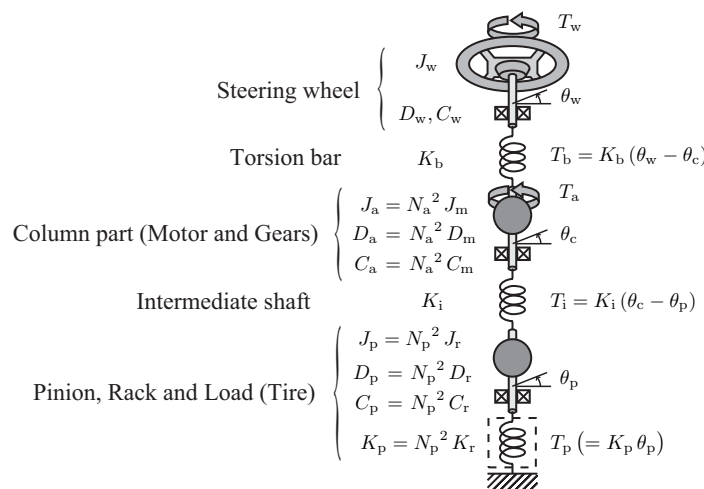


図 2.2 Equivalent model of C-EPS

化モデルを図 2.2 に示す．文献 [15] では摩擦としては粘性摩擦のみを考慮しているが，準静的な特性を詳細にシミュレーションし，操舵角-トルク特性を現実的に再現するために，本論文では更にクーロン摩擦の考慮を行う．トルクセンサのトーシヨンバー及びインターミディエイトシャフトは線形弾性変形をすると仮定し，その弾性係数をそれぞれ K_b ， K_i とする．アシスト用電動モータ及びウォーム減速器を含むコラム部分をコラム軸等価部としてまとめ，ピニオン軸より先の部分をピニオン軸等価部としてまとめる．ハンドル軸，コラム軸，ピニオン軸における代表等価定数としてそれぞれ，慣性モーメント J_w ， J_a ， J_p ，粘性摩擦係数 D_w ， D_a ， D_p ，クーロン摩擦トルクの大きさ C_w ， C_a ， C_p を導入する．ラック負荷として加わるタイヤのセルフアライニングトルクをピニオン軸に換算したトルクを T_p とする．それらの代表等価値を用いて表される等価三慣性モデルの運動方程式は，ハンドル軸，コラム軸，ピニオン軸における回転角度を θ_w ， θ_c ， θ_p として次のようになる．

$$J_w \ddot{\theta}_w + D_w \dot{\theta}_w + C_w \text{sign}(\dot{\theta}_w) + K_b (\theta_w - \theta_c) = T_w \quad (2.1)$$

$$J_a \ddot{\theta}_c + D_a \dot{\theta}_c + C_a \text{sign}(\dot{\theta}_c) - K_b (\theta_w - \theta_c) + K_i (\theta_c - \theta_p) = T_a \quad (2.2)$$

$$J_p \ddot{\theta}_p + D_p \dot{\theta}_p + C_p \text{sign}(\dot{\theta}_p) - K_i (\theta_c - \theta_p) + T_p = 0 \quad (2.3)$$

以降の節では，式 (2.1) ~ (2.3) の準静的部分に対して解析を行い，それに基づいて提案アシスト制御手法の導出を行う．そのために，次の 2.2.2 節で一定車速において準静的な操舵を行う状況を想定して関係式を導く．

2.2.2 準静的条件下での関係式の導出

準静的な解析を行うために，全ての軸がゼロに近い一定速度で同一方向に回転している状況を考える．すなわち，各軸における角加速度，及び，角速度が近似的にゼロであり，また，角速度の符号は全て同じ，

$$\ddot{\theta}_w \approx 0, \quad \ddot{\theta}_c \approx 0, \quad \ddot{\theta}_p \approx 0 \quad (2.4)$$

$$\dot{\theta}_w \approx 0, \quad \dot{\theta}_c \approx 0, \quad \dot{\theta}_p \approx 0 \quad (2.5)$$

$$\text{sign}(\dot{\theta}_w) = \text{sign}(\dot{\theta}_c) = \text{sign}(\dot{\theta}_p) = \pm 1 \quad (2.6)$$

とする．式 (2.6) の仮定により，クーロン摩擦トルクが全ての軸で同じ方向に働く状況を想定していることになる．ピニオン軸に換算したセルフアライニングトルク T_p は本来タイヤ及び車両の動特性を含む項であるが，低速走行時を除くと一定車速で準静的に操舵する場合にある操舵角の範囲ではヒステリシスを持つ線形特性とみなせる．このとき，ピニオン軸における等価な弾性係数を K_p として，

$$T_p = K_p \theta_p \quad (2.7)$$

とおけ，またヒステリシスはピニオン軸等価部のクーロン摩擦トルク C_p に含めて考えることができる． K_p ， C_p は車速ごとに定まる車両とタイヤの特性を反映したパラメータである．式 (2.1) ~ (2.3) の運動方程式に，式 (2.4) ~ (2.7) の準静的条件の仮定を適用し，それらの式を θ_w ， θ_c ， θ_p について整理し， θ_c ， θ_p を消去して θ_w の式を求めると，

$$\theta_w = \mp \bar{\theta}_w + \frac{T_w}{\bar{K}_w} + \frac{T_a}{\bar{K}_a} \quad (2.8)$$

のような操舵角-トルク特性の関係式が得られる（複号同順）．ただし， $\bar{\theta}_w$ ， \bar{K}_w ， \bar{K}_a は，

$$\bar{\theta}_w \triangleq \frac{C_w}{K_b} + \frac{C_w + C_a}{K_i} + \frac{C_w + C_a + C_p}{K_p} \quad (2.9)$$

$$\bar{K}_w \triangleq \frac{1}{\frac{1}{K_b} + \frac{1}{K_i} + \frac{1}{K_p}}, \quad \bar{K}_a \triangleq \frac{1}{\frac{1}{K_i} + \frac{1}{K_p}} \quad (2.10)$$

により定義される定数である． $\bar{\theta}_w$ は操舵角切片であり，後でその意味を説明する．式 (2.10) より， \bar{K}_w は K_b, K_i, K_p の線形バネの直列接続に等価な剛性であり， \bar{K}_a は K_i, K_p の直列接続に等価な剛性である． \bar{K}_w 及び \bar{K}_a は，それぞれ T_w あるいは T_a を加えたときに θ_w がどれだけ変化するかを表す剛性値である．操舵トルクがゼロの近傍 $T_w \approx 0$ でトルクセンサが不感帯内に入ると検出トルクがゼロ $T_s = 0$ となり，そのときアシストトルクがゼロ $T_a = 0$ とすると，式 (2.8) より，操舵角-トルク特性は操舵角軸 ($T_w = 0$) を横切るとき $\theta_w = \mp \bar{\theta}_w$ となる点を必ず通る．つまり， $\theta_w = -\bar{\theta}_w$ が正転時の操舵角切片であり， $\theta_w = +\bar{\theta}_w$ が逆転時の操舵角切片である．

操舵角切片の大きさは，式 (2.9) のように各部のクーロン摩擦トルクの大きさ及び各部間の弾性係数に依存しており，操舵角切片を望みの位置に移動させるためには，クーロン摩擦及び弾性変形の影響を補償するための補正をアシスト制御に加える必要がある．次の 2.3 節でそのような補正項を加えたアシスト制御式の構成を示す．本論文で提案するアシスト制御は式 (2.8) を基に導出しており準静的な目標操舵角-トルク特性を実現するものであるが，動的特性を解析・補償するためには別途考慮する必要がある．

2.3 アシスト制御式の構成

2.3.1 検出トルク不感帯の設定

EPS を構成する上で，不感帯は操作感に影響する重要な要素であるのでここでその考慮を行う．不感帯は正負対称であるので，中心から片側の境界までの幅でその大きさを表す．

トルクセンサでの検出トルク T_s は，トーションバーの捩れトルクを T_b ，トルクセンサの不感帯の大きさを $\gamma_s (> 0)$ として，

$$T_s = \begin{cases} 0 & \text{if } |T_b| < \gamma_s \\ (|T_b| - \gamma_s) \text{sign}(T_b) & \text{otherwise} \end{cases} \quad (2.11)$$

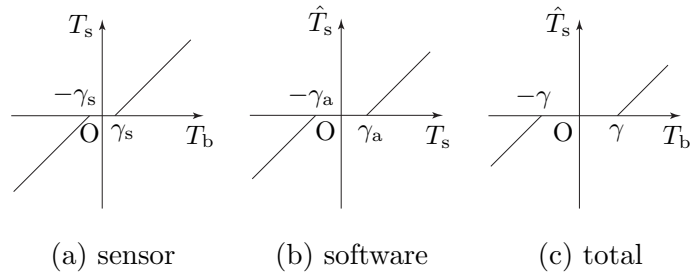


図 2.3 Deadband of torque

と表せ，図示すると図 2.3(a) となる．ソフトウェアで設ける任意の不感帯の大きさを $\gamma_a (\geq 0)$ とすると，アシスト制御を適用して実現される見かけ上のトルクの不感帯の大きさ γ は，

$$\gamma = \gamma_s + \gamma_a \quad (2.12)$$

で与えられる．実現される不感帯の大きさ γ はハードウェアの制約上， $\gamma \geq \gamma_s > 0$ となる．つまり，トルクセンサの不感帯より大きい不感帯しか実現できない．トルクセンサの不感帯を考慮した検出トルク T_s と区別して，以降の節で用いるためにソフトウェアの不感帯を更に考慮した検出トルク \hat{T}_s を，

$$\hat{T}_s = \begin{cases} 0 & \text{if } |T_s| < \gamma_a \\ (|T_s| - \gamma_a) \text{sign}(T_s) & \text{otherwise} \end{cases} \quad (2.13)$$

のように定める．これを図示すると図 2.3(b) となる．このとき， \hat{T}_s は，式 (2.11) ~ (2.13) より， T_b ， γ を用いて，

$$\hat{T}_s = \begin{cases} 0 & \text{if } |T_b| < \gamma \\ (|T_b| - \gamma) \text{sign}(T_b) & \text{otherwise} \end{cases} \quad (2.14)$$

と書き直せ，図示すると図 2.3(c) となる．

不感帯が大き過ぎると余分な操舵トルク及び操舵量が共に必要となり操舵感を悪化させるので，制約の範囲内で不感帯はある程度小さくする必要がある．しかし，逆に実現された不感帯が小さ過ぎると，人間の操舵に対して敏感に反応し過ぎて安定感に欠けて操舵し難くなってしまふ．ソフトウェアにおいて不感帯を適切に調節して実現される不

感帯を最適なものに設定する必要があるが，提案する手法において設定すべきソフトウェアの不感帯については 2.4.2 節で示す．

また，ここで後の節のために，準静的条件を仮定した際に検出トルクが不感帯内となる操舵トルクの条件を示しておく．トーションバーの捩れトルク $T_b (= K_b(\theta_w - \theta_c))$ は，式 (2.1)，及び，式 (2.4) ~ (2.6) の準静的条件の仮定より，回転方向 $\text{sign}(\dot{\theta}_w) = \dots = \pm 1$ と複号同順で，

$$T_b = T_w \mp C_w \quad (2.15)$$

となる．式 (2.14) より検出トルクが不感帯内となるのは $-\gamma < T_b < \gamma$ であり，その不等式に式 (2.15) を代入して整理すると，検出トルクが不感帯内となる操舵トルクの条件が導かれて，

$$\begin{cases} -\gamma + C_w < T_w < \gamma + C_w & \text{if } \dot{\theta}_w > 0 \\ -\gamma - C_w < T_w < \gamma - C_w & \text{if } \dot{\theta}_w < 0 \end{cases} \quad (2.16)$$

となる．

2.3.2 方向依存性アシスト補正項の付加

アシストトルク T_a を定めるために，基本的なアシスト制御として基本アシストマップ $\Phi(\hat{T}_s)$ による制御を考え，それに，2.2.2 節で述べたようなクーロン摩擦と弾性変形の影響を補償する目的で，方向依存性アシスト補正項 $T_c(\hat{T}_s, \dot{T}_s)$ を加えて，

$$T_a = \Phi(\hat{T}_s) + T_c(\hat{T}_s, \dot{T}_s) \quad (2.17)$$

により T_a を定める．ここで $T_c(\hat{T}_s, \dot{T}_s)$ は，

$$T_c(\hat{T}_s, \dot{T}_s) = \Phi_c(\hat{T}_s) \text{sign}(\dot{T}_s) \quad (2.18)$$

のように，検出トルクの変化率 \dot{T}_s の符号に応じて，方向依存性アシスト補正マップ $\Phi_c(\hat{T}_s)$ で決定される量だけアシストトルクを増減する項である． $\hat{T}_s = 0$ における方向依存性アシスト補正マップの値 $\beta_0 (\triangleq \Phi_c(0))$ を原点補正量と呼ぶことにする．

$\Phi(\hat{T}_s)$ 及び $\Phi_c(\hat{T}_s)$ の二つのアシスト制御用マップを適切に設定することで，望みの操舵角-トルク特性を実現することが可能であり，2.4 節でその設定手法を示す．

2.3.3 操舵角切片の移動

ここでは、2.3.2 節で提案した方向依存性アシスト補正項を加えたアシスト制御を適用した場合に、2.2.2 節の最後に述べたような操舵角切片の移動ができることを示す．そのために不感帯内における準静的な挙動の解析を行う．通常、操舵角-トルク特性をプロットする際には操舵特性を測定する領域の両端でのみ切り返しを行い、その間は継続的に操舵を行う．よって、不感帯内では切り返しは行わないものとする．

トーションバーにおける捩りトルク T_b が不感帯の範囲内、つまり $|T_b| < \gamma$ のとき、式 (2.14) より $\hat{T}_s = 0$ となって、基本アシストトルクは基本アシストマップの原点の値 $\Phi(0)$ となる．ここで、基本アシストマップの対称性から原点での値はゼロ $\Phi(0) = 0$ である．切り返しを含まない継続的な操舵を考えているので、 $|T_b| < \gamma_s$ となりトルクセンサ自体の不感帯内に入って、一時的にセンサ出力がゼロ $T_s = 0$ となる状態が続いたときにも、仮定より \hat{T}_s の符号が保存され、角速度の符号と同じとなる．つまり、 $\text{sign}(\hat{T}_s) = \text{sign}(\dot{\theta}_w) = \dots = \pm 1$ が成り立つ．このとき、式 (2.17)、(2.18) よりアシストトルクは $T_a = \pm\beta_0$ となり、式 (2.8) に代入して、不感帯における操舵角-トルク特性の関係式は、

$$\theta_w = \mp(\bar{\theta}_w - \frac{\beta_0}{\bar{K}_a}) + \frac{T_w}{\bar{K}_w} \triangleq \mp\bar{\theta}'_w + \frac{T_w}{\bar{K}_w} \quad (2.19)$$

と表せる．以上より、原点補正量 β_0 を含む方向依存性アシスト補正項を加えることで、操舵角切片を $\mp\bar{\theta}_w$ から $\mp\bar{\theta}'_w = \mp(\bar{\theta}_w - \frac{\beta_0}{\bar{K}_a})$ に変更できることが分かる．

2.4 目標の特性を実現するアシスト制御設定法

2.4.1 目標操舵角-トルク特性

目標とする操舵角-トルク特性を図 2.4 に示すような形で表す．どのような形状が良い操舵感を与えるかは車両特性や感性に関わり、また車速に応じて変わるが、本論文では形状の定め方には触れず、あらかじめ与えられているとする．以下で基本的な制約を

考慮した形状の与え方を定義する．

まず，目標とする操舵角切片 $\bar{\theta}_{wd}$ 及び不感帯幅 γ_d を与える．検出トルクが不感帯内となる操舵トルクの条件は式 (2.16) であるので， $\gamma = \gamma_d$ が実現したとき，正・逆転時に目標とする特性をそれぞれ不感帯で分けると，

$$T_w = \begin{cases} \Psi_{++}(\theta_w) & \text{if (A) } \dot{\theta}_w > 0, T_w > \gamma_d + C_w \\ \Psi_{+-}(\theta_w) & \text{if (B) } \dot{\theta}_w > 0, T_w < -\gamma_d + C_w \\ \Psi_{--}(\theta_w) & \text{if (C) } \dot{\theta}_w < 0, T_w < -\gamma_d - C_w \\ \Psi_{-+}(\theta_w) & \text{if (D) } \dot{\theta}_w < 0, T_w > \gamma_d - C_w \end{cases} \quad (2.20)$$

の4つの部分に分けられる．通常 $C_w < \gamma_d$ であり，そのとき，不感帯より上の部分は $T_w > 0$ に含まれ，下の部分は $T_w < 0$ に含まれる．原点に関して点対称であることを考慮すると，特性の形状としては不感帯より上の部分である (A) と (D) の組，あるいは下の部分である (B) と (C) の組のどちらかの組のみを考えれば良い．すなわち，切り増し・切り戻し時の特性が不感帯より上の領域でそれぞれ $\Psi_+(\theta_w)$ ， $\Psi_-(\theta_w)$ であるとすると，

$$\Psi_{++}(\theta_w) = -\Psi_{--}(-\theta_w) = \Psi_+(\theta_w) \quad (2.21)$$

$$\Psi_{-+}(\theta_w) = -\Psi_{+-}(-\theta_w) = \Psi_-(\theta_w) \quad (2.22)$$

であるので，目標の特性プロファイルとして不感帯より上の領域で $\Psi_+(\theta_w)$ 及び $\Psi_-(\theta_w)$ を与えるものとする．

不感帯の上側境界での操舵トルクは $T_w = \gamma_d \pm C_w$ であり，切り増し・切り戻し時それぞれにおける不感帯上側境界での操舵角 θ_{db+} ， θ_{db-} は，式 (2.19) で $\bar{\theta}'_w = \bar{\theta}_{wd}$ が実現されたとして，境界における操舵トルクを代入して，

$$\theta_{db+} = -\bar{\theta}_{wd} + \frac{\gamma_d + C_w}{\bar{K}_w}, \quad \theta_{db-} = \bar{\theta}_{wd} + \frac{\gamma_d - C_w}{\bar{K}_w} \quad (2.23)$$

となる．不感帯内ではマップの適用はできず，従って形状を考慮する余地はなく，よって $\Psi_+(\theta_w)$ ， $\Psi_-(\theta_w)$ は，それぞれ不感帯境界となる点 $(\theta_{db+}, \gamma_d + C_w)$ ， $(\theta_{db-}, \gamma_d - C_w)$ より右上方に向かって定義する．

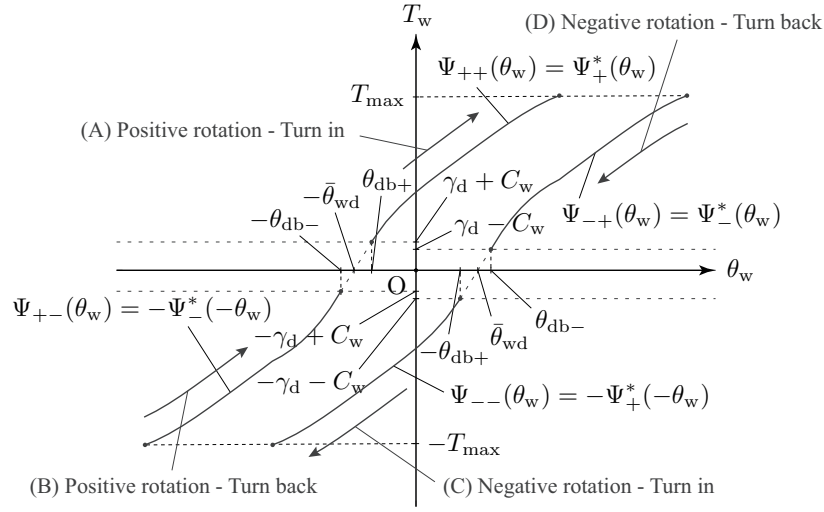


図 2.4 Objective characteristics

2.4.2 アシスト制御用マップ計算式

2.4.1 節で示したように，目標とする操舵角-トルク特性のプロファイルを不感帯より上の領域で $\Psi_+^*(\theta_w)$ と $\Psi_-^*(\theta_w)$ によって与える．ここで目標の特性を実現するのに必要なアシストトルクも $\hat{T}_s = 0$ に対して対称性があることを考慮して，不感帯より上の領域に対応する $\hat{T}_s > 0$ の領域でアシストマップの形状を求めることにする．

特性プロファイルの逆関数について，

$$\Psi_{\pm}^{*-1}(T_w) \triangleq \begin{cases} \Psi_+^{*-1}(T_w) & \text{if } \dot{\theta}_w > 0 \\ \Psi_-^{*-1}(T_w) & \text{if } \dot{\theta}_w < 0 \end{cases} \quad (2.24)$$

のように，正逆回転別に異なる関数形を便宜的に一つにまとめた関数 $\Psi_{\pm}^{*-1}(T_w)$ を定義する．これにより操舵角 θ_w は，

$$\theta_w = \Psi_{\pm}^{*-1}(T_w) \quad (2.25)$$

と表せる．また，式 (2.17) のアシストトルク T_a を計算する右辺の二項 $\Phi(\hat{T}_s) + T_c(\hat{T}_s, \dot{T}_s)$ において，式 (2.18) を考慮して，二項をまとめた関数を，検出トルクの時間変化率の符

号別に $\Phi_+(\hat{T}_s) \triangleq \Phi(\hat{T}_s) + \Phi_c(\hat{T}_s)$, $\Phi_-(\hat{T}_s) \triangleq \Phi(\hat{T}_s) - \Phi_c(\hat{T}_s)$ のように定義したとき, $\text{sign}(\dot{\theta}_w) = \text{sign}(\dot{T}_s) = \pm 1$ を考慮すると, アシストトルク T_a は, 正・逆回転で場合分けして,

$$T_a = \begin{cases} \Phi_+(\hat{T}_s) & \text{if } \dot{\theta}_w > 0 \\ \Phi_-(\hat{T}_s) & \text{if } \dot{\theta}_w < 0 \end{cases} \quad (2.26)$$

とおける. ここで便宜的に,

$$\Phi_{\pm}(\hat{T}_s, \dot{T}_s) \triangleq \begin{cases} \Phi_+(\hat{T}_s) & \text{if } \dot{\theta}_w > 0 \\ \Phi_-(\hat{T}_s) & \text{if } \dot{\theta}_w < 0 \end{cases} \quad (2.27)$$

のような関数 $\Phi_{\pm}(\hat{T}_s, \dot{T}_s)$ を定義すると,

$$T_a = \Phi_{\pm}(\hat{T}_s, \dot{T}_s) \quad (2.28)$$

のように, 正・逆回転の場合分けを一つの式に含む形で書ける.

式 (2.8) に式 (2.28) を代入して, 式 (2.25) と照らし合わせて特性プロファイルの逆関数 $\Psi_{\pm}^*{}^{-1}(T_w)$ を求めると,

$$\theta_w = \Psi_{\pm}^*{}^{-1}(T_w) = \mp \bar{\theta}_w + \frac{T_w}{\bar{K}_w} + \frac{\Phi_{\pm}(\hat{T}_s, \dot{T}_s)}{\bar{K}_a} \quad (2.29)$$

となる. ただし, 式 (2.29) の複号と関数の場合分けは $\text{sign}(\dot{T}_s) = \text{sign}(\dot{\theta}_w) = \pm 1$ の複号と同順である. ここで, 式 (2.14) において, $\hat{T}_s > 0$ の場合を取り, $\gamma = \gamma_d$ 及び式 (2.15) を代入すると,

$$\hat{T}_s = T_b - \gamma_d = T_w \mp C_w - \gamma_d \quad (2.30)$$

である. 式 (2.30) より $T_w = \hat{T}_s \pm C_w + \gamma_d$ であり, 式 (2.29) に代入して, $\Phi_{\pm}(\hat{T}_s, \dot{T}_s)$ について解いて整理すると,

$$\Phi_{\pm}(\hat{T}_s, \dot{T}_s) = \bar{K}_a \left\{ \Psi_{\pm}^*{}^{-1}(\hat{T}_s \pm C_w + \gamma_d) \pm \bar{\theta}_w - \frac{\hat{T}_s \pm C_w + \gamma_d}{\bar{K}_w} \right\} \quad (2.31)$$

となる (複号と関数の場合分けは同順). $\Phi_+(\hat{T}_s)$ と $\Phi_-(\hat{T}_s)$ の定義より, $\Phi(\hat{T}_s)$, 及び,

$\Phi_c(\hat{T}_s)$ はそれぞれ、 $\Phi_+(\hat{T}_s)$ と $\Phi_-(\hat{T}_s)$ の平均、及び、差の二分の一で表せ、

$$\begin{aligned}\Phi(\hat{T}_s) &= \frac{\Phi_+(\hat{T}_s) + \Phi_-(\hat{T}_s)}{2} \\ &= \frac{\bar{K}_a}{2} \left\{ \Psi_+^{*-1}(\hat{T}_s + C_w + \gamma_d) + \Psi_-^{*-1}(\hat{T}_s - C_w + \gamma_d) \right\} \\ &\quad - \frac{\bar{K}_a}{\bar{K}_w} (\hat{T}_s + \gamma_d) \quad (2.32)\end{aligned}$$

$$\begin{aligned}\Phi_c(\hat{T}_s) &= \frac{\Phi_+(\hat{T}_s) - \Phi_-(\hat{T}_s)}{2} \\ &= \frac{\bar{K}_a}{2} \left\{ \Psi_+^{*-1}(\hat{T}_s + C_w + \gamma_d) - \Psi_-^{*-1}(\hat{T}_s - C_w + \gamma_d) \right. \\ &\quad \left. + 2\bar{\theta}_w - \frac{2C_w}{\bar{K}_w} \right\} \quad (2.33)\end{aligned}$$

が導かれる．これらの式によってアシストマップの $\hat{T}_s > 0$ における値を計算することができる．

次に $\hat{T}_s = 0$ でのマップの値について考えるが、基本アシストマップは原点に関して点対称であり、連続性を考慮すると原点でゼロとなる必要がある．また、方向依存性アシスト補正マップの $\hat{T}_s = 0$ における値、つまり原点補正量 β_0 は、式 (2.19) で示される補正適用時の操舵角切片 $\bar{\theta}'_w$ が、目標の操舵角切片 $\bar{\theta}_{wd}$ となるように決めれば良い．よって、検出トルクがゼロ $\hat{T}_s = 0$ でのマップの値は、

$$\Phi(0) = 0, \quad \Phi_c(0) = \beta_0 = \bar{K}_a (\bar{\theta}'_w - \bar{\theta}_{wd}) \quad (2.34)$$

と定められる．式 (2.34) の値は、式 (2.32) 及び式 (2.33) で検出トルクがゼロ $\hat{T}_s = 0$ とした値と等しく、二つのアシストマップは共に原点における連続性が満たされる． $\hat{T}_s < 0$ におけるマップの値に関しては、詳細な説明は紙面の都合上省くが、基本アシストマップは原点に対して点対称、方向依存性アシスト補正マップは $\hat{T}_s = 0$ に対して線対称となるように定めれば良い．また、ソフトウェアによる不感帯の設定では、 $\gamma = \gamma_d$ とするために、式 (2.12) より $\gamma_a = \gamma_d - \gamma_s$ と定めれば良い．

目標操舵角-トルク特性は車速ごとに異なり、また車両やタイヤの特性を反映したパラメータ K_p , C_p も車速ごとに変わるので、アシストマップは車速ごとに求める必要がある．

2.4.3 実現可能な操舵角-トルク特性の制約

提案手法では、式 (2.32) 及び式 (2.33) によってアシストマップの計算を行うが、式中の $\Psi_+^{*-1}(\hat{T}_s + C_w + \gamma_d)$ 及び、 $\Psi_-^{*-1}(\hat{T}_s - C_w + \gamma_d)$ の 2 つの逆関数が計算可能であるためには、目標特性の形状を与える $\Psi_+(\theta_w)$ 及び $\Psi_-(\theta_w)$ が 1 対 1 対応の関数である必要がある。つまり、連続性を考慮すると、 $\Psi_+(\theta_w)$ 及び $\Psi_-(\theta_w)$ は単調増加である必要があり、本提案手法によって実現可能な操舵角-トルク特性の制約として、単調増加なものに限られる。

2.5 シミュレーションと実験による検証

提案手法を用いてアシストマップの生成を行った数値例を示し、また、与えた目標の特性を実現できることをシミュレーション及びベンチ実験装置を用いた実験により検証した結果を示す。ベンチ実験装置は図 2.5 の写真で示すようなものであり、台上試験装置に実車のステアリング装置を備え付けたものである。実験ベンチでは準静的条件下での検証を行うために、負荷として式 (2.7) を実現する板バネを用い、シミュレーションでは提案手法導出の基となった式 (2.1) ~ (2.3) 及び式 (2.7) の妥当性を確認するために実験ベンチの実際の応答との比較を行った。検証では $\pm 80[\text{deg}]$ 程度の範囲で両端での切り返しを除いて準静的に継続して操舵を行い、操舵角-トルク特性を計測した。

Case A 及び Case B の 2 種類の目標操舵特性を想定してマップの生成を行った。Case A は Case B に比べて、目標とする操舵角切片が小さく、また操舵中における操舵トルクが全体的に小さくて済むような目標特性である。図 2.6 の実線及び一点鎖線がそれぞれ、Case A 及び Case B で目標として与えた操舵角-トルク特性であり、それぞれシミュレーション及び実験で同じものを与えた。図 2.7 の実線及び破線がそれぞれ、シミュレーションモデルのパラメータに基づいて生成した基本アシストマップ $\Phi(\hat{T}_s)$ 、及び方向依存性アシスト補正マップ $\Phi_c(\hat{T}_s)$ である。実験においては、簡単な実験により $\bar{\theta}_w$ 、 \bar{K}_w 、 \bar{K}_a の推定を行い、推定値に基づいて生成したマップを用いた。実験のため



図 2.5 An experimental bench of the C-EPS

に生成したマップは紙面の都合上割愛するが図 2.7 と同様のものではあった。シミュレーション及び実験のために生成したアシストマップを用いて、アシスト補償制御を適用した時にそれぞれ実際に得られる操舵角-トルク特性をプロットした結果を図 2.8 に示す。比較のために同じ図上に目標特性を破線及び一点鎖線でプロットしている。シミュレーション及び実験の結果から、生成されたマップを用いたアシスト制御によって目標とする操舵角-トルク特性がよく実現できることが分かり、提案手法の妥当性が示された。また、シミュレーションが実験ベンチの応答を再現していることから本論文で用いたモデルの妥当性が確認された。

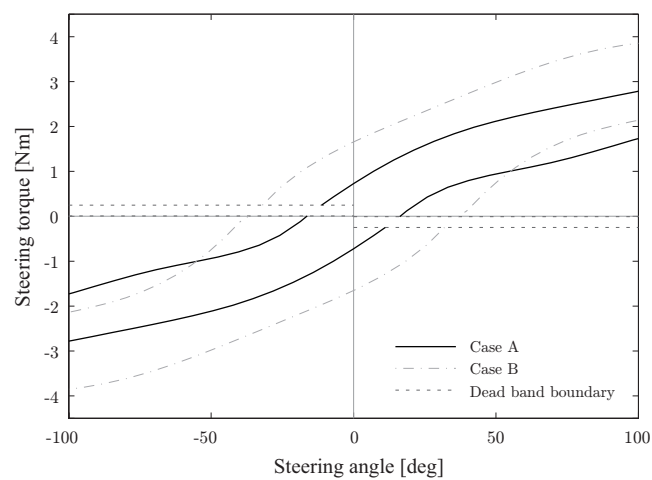


図 2.6 Given profiles of objective characteristics

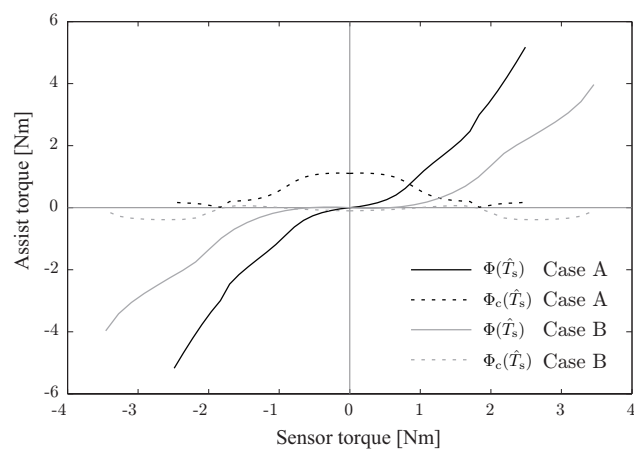
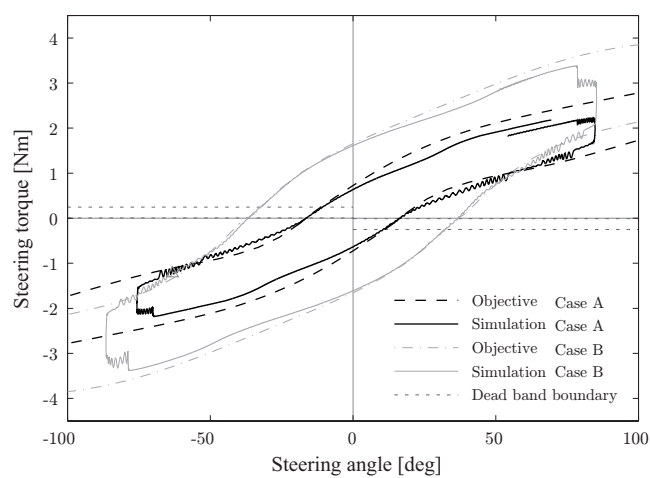


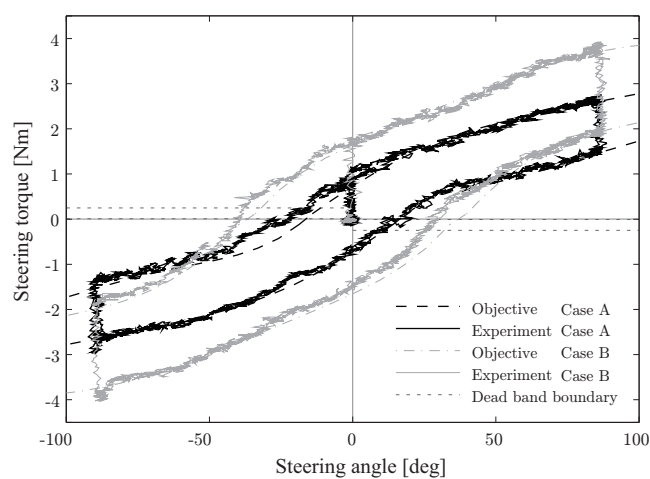
図 2.7 Profiles of the created maps

2.6 結言

本論文では，EPS 中でも今後より重要となる C-EPS を想定して，目標とする準静的な操舵角-トルク特性を実現する系統的手法を提案した．そのために，アシスト制御と



(a) Simulation



(b) Experiment

図 2.8 Simulation and experimental results

して従来より一般的に用いられていた基本アシストマップに加えて，方向依存性アシスト補正項を用いる構成を提案した．C-EPS の等価三慣性モデルを構築し，その準静的解析を基にアシスト制御の構成やそこで用いるアシスト制御用マップ設定法について考察及び導出を行った．また，数値例として実際にアシスト制御用マップの生成を行い，シ

ミュレーション及び実験ベンチにより目標とする操舵角-トルク特性を実現できることを検証し、提案手法の妥当性を示した。

C-EPS に関連した研究における今後の課題としては、動特性を考慮した解析・補償の考案や、中心付近の微小な領域における動作の解析・操作感の改善、操作感の定量的な官能評価に基づく設計・制御・調整手法の研究・開発、操舵角に応じたより高度な制御の検討などが挙げられる。

第 3 章

遭遇型ハプティックデバイスを用いたバーチャル操作パネル上の複数スイッチ提示システム

3.1 緒言

横小路ら [18] が提案した手法は、インパネ上の各スイッチごとに 1 つの代表点とそこでの提示姿勢を定めておき、個々のスイッチの代表点を結んでできる多面体を基に周辺の空間を Voronoi 領域に分け、ユーザの指がその中のどの領域にあるかによって効率の良いデバイスの動作を決定するものであったが、構成した多面体の内部にもスイッチがあるときにデバイスの動作が不連続になるという問題があり、また提示できるスイッチは 1 種類のみであった。本論文は、1 種類のスイッチのみ提示可能であった横小路ら [18] のシステムを拡張し、バーチャル空間内に構築された操作パネル上の複数のスイッチを操作する感覚を 1 つの遭遇型ハプティックデバイスで提示するシステムを構築する。筆者らは既に、重田ら [22] が多指遭遇型ハプティックデバイスで用いるために提案した最近接点情報に基づく動作計画手法を基に、スイッチ提示システムで用いる遭遇型ハプティックデバイスの動作計画手法を提案してきた [24]。本論文では、デバイスの動

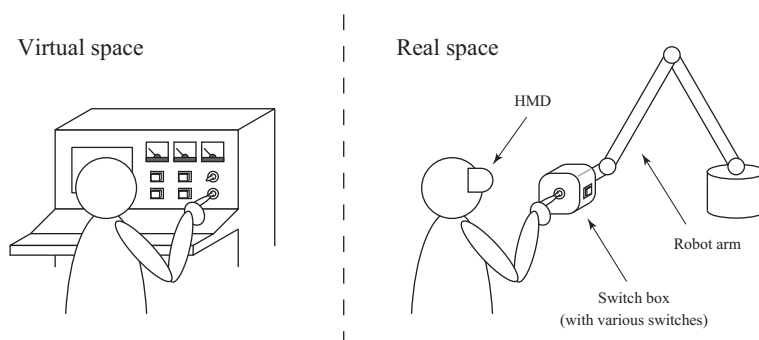


図 3.1 バーチャル操作パネル提示システムの概念図









作負担を軽減するために、動作計画で目標提示位置・姿勢の計算に用いる重み関数の改良を行った。また、バーチャルパネル面座標系を導入して、デバイスがバーチャルパネル面に沿って動くようにすることでユーザとの不用意な衝突が起こらないように動作計画の改善を行った。以下本論文では、3.2 節でデバイスの動作計画手法について詳細に述べ、3.3 節で提案手法の有効性をシミュレーションにより検証した結果を示す。

3.2 バーチャル操作パネル提示システムとデバイスの動作計画

3.2.1 システムの概要

提案するシステムの構成を概念的に示したものが図 3.1 である。6 自由度以上の自由度を有するロボットアームの先端に、複数のスイッチを設置した多面体（以降では「スイッチボックス」と呼ぶ）を取り付け、ユーザがヘッドマウントディスプレイ (HMD) を通して見たバーチャル空間内に構成されたバーチャル操作パネルを操作しようとしたときに、ユーザの指先位置の情報を用いてスイッチボックスの位置・姿勢を制御し、ユーザにバーチャル操作パネルを操作する感覚を提示する。提示したい操作パネル上のスイッチの中に同一種類のものが複数あれば、操作パネル上のスイッチよりも少ない実スイッチをもつ簡略化されたデバイスによりバーチャル操作パネルを提示できることにな

表 3.1 スイッチの分類

categories based on distinct states	categories based on number of orientation candidates			characteristics	
	(i) ∞	(ii) 2, 3, ...	(iii) 1	number of distinct states	multi-renderability by a single real SW
(a) infinite rotary SW		-	-	1	
(b) momentary SW				1	
(c) symmetrical bipolar toggle SW	-	-		2	
(d) other types of SW (push-lock SW, selector SW, etc.)	 *3	 *3		2, 3, ...	×

*1 graphics are examples *2 -:NA *3 push-lock SW

る。また、ユーザの手指に対するデバイスの追従性を確保するために、複数のスイッチ操作を続けて行う作業を模擬する際のデバイスの動作がなるべく少なく済むように、スイッチボックスの設計及び動作計画を決めることが望ましい。以下では、動作計画について考える前に、まず提示するスイッチについて考察する。

3.2.2 スイッチの分類

ここでは、スイッチボックスの設計及びデバイスの動作計画を見通しよく行えるように、提示すべき状態の区別と法線軸回りの配置の任意性という2つの観点からスイッチの分類を行う。

〔提示すべき状態の区別による分類〕

バーチャルスイッチを提示するために必要な状態の区別と、実スイッチが対応できる状態の対称性によってスイッチを分類すると次の4種類に分けられる。

(a) 無限ロータリスイッチ

固有に区別できる状態は定義できず，対称性によりバーチャルスイッチがどの状態であっても一つの実スイッチで提示することが可能である．

(b) モーメンタリスイッチ

指がスイッチから離れている場合には常に off 状態であるので，スイッチの on-off 状態を区別する必要はない．

(c) 2 極対称トグルスイッチ

スイッチには on 状態と off 状態があるが，スイッチの形状に対称性があり，実スイッチを 180 度回転することによりどちらの状態も提示することが可能である．

(d) その他のスイッチ

スイッチには固有に区別できる複数の状態があり，それぞれのバーチャルスイッチを提示するために対応する実スイッチを一つずつ設けて同じ状態を保存する必要がある．

この分類において，(a)，(b)，(c) のスイッチは，バーチャル操作パネル上に同じ種類のスイッチが複数ある場合でもデバイス上の 1 つの実スイッチで模擬することが可能である．

〔法線軸回りの配置による分類〕

スイッチ設置面の法線軸回りに $0 \leq \theta < 2\pi$ 回転させたときに，区別が付かなくなる配置の数でスイッチを分類すると次の 3 種類に分けられる．

- (i) 無限個の配置が存在する
- (ii) 2 つ以上の有限個の配置が存在する
- (iii) 一意な配置が存在する

配置に任意性がある場合には提示姿勢は一意に決まらないが，次の 3.2.3 節では，逆にこの任意性を利用してデバイスの動作負担を軽減する方法について述べる．

種々のスイッチは2つの観点に基づいて分類されるが、表 3.1 は2つの観点に基づくスイッチの分類についてまとめて、実際のスイッチの例を挙げて示したものである。

3.2.3 任意性のある提示姿勢の決定

3.2.2 節で述べたように、スイッチによってはその法線軸回りに配置の任意性を有する場合があります、バーチャルスイッチを提示する際にこの任意性を有効に利用して実スイッチによる提示姿勢を決定することで、デバイスの動作負担を軽減できる可能性がある。以下で配置の任意性を有効に利用した提示姿勢の決定手法について述べる。

ここで、各バーチャルスイッチには個々のスイッチを特定する番号が振られているとし、 S_i はスイッチ番号により特定のバーチャルスイッチを表しているとする。バーチャルスイッチが N 個あった場合に、デバイス上の実スイッチによる提示姿勢を決める順にバーチャルスイッチを S_1, S_2, \dots, S_N とする。この順序は、法線軸回りの配置の任意性の低いものから順に決めていく。すなわち、一意に決まるものは始めに定まっているとし、次に任意性が $2, 3, \dots$ である順に定め、最後に任意性が無限にあるものを定める。任意性が同じスイッチが複数ある場合には、既に定まっている周囲のスイッチとの距離が近いほど、また、その距離が近いスイッチが多いほど、そのスイッチの提示姿勢を定め易いと思見して優先して決める。すなわち、 S_{i-1} まで提示姿勢が決まったときに、次に提示姿勢を決めようとしている任意性が同じスイッチ番号の集合から既に提示姿勢が決まったスイッチ番号を除いた集合を G_i とすると、次に提示姿勢を決めるべきスイッチ S_i は、

$$S_i = \operatorname{argmax}_{j \in G_i} w_{\text{sum } i, j}^s \quad (3.1)$$

により決められる。ここで、 $w_{\text{sum } i, j}^s$ は G_i に属するあるスイッチ番号 j のスイッチに対する既に定まったスイッチ $S_k (k = 1, 2, \dots, i-1)$ についての重み $w_{j, k}^s$ の k についての和、すなわち、

$$w_{\text{sum } i, j}^s = \sum_{k=1}^{i-1} w_{j, k}^s \quad (3.2)$$

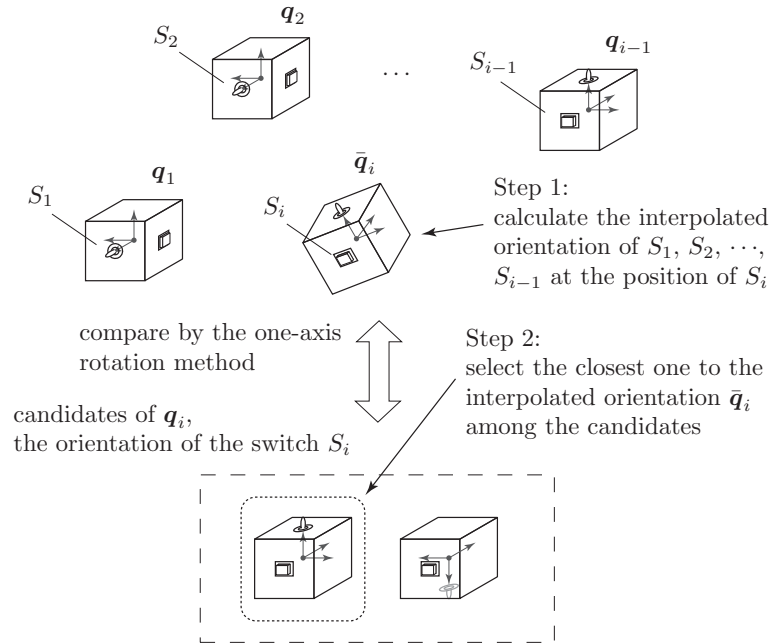


図 3.2 任意性のある提示姿勢の決定

により計算される．重み $w_{j,k}^s$ は後述する提示姿勢の計算で用いるのと同じものである．
 このように，任意性の数と重みの和によって提示姿勢決定の順序が決められたとして，
 次に，法線軸回りに配置の任意性がある場合の提示姿勢の決め方について述べる．

ここで，デバイス上の実スイッチは予めスイッチボックス上に適切に配置されている
 とし，デバイス上の実スイッチによる提示姿勢はデバイスの姿勢（スイッチボックスに
 固定されたデバイス座標系の姿勢）を表すクォータニオン q を用いて表す．バーチャル
 スイッチ S_i に対する実スイッチの提示姿勢 q_i を決める場合，それより以前に決まっ
 ている提示姿勢 q_1, q_2, \dots, q_{i-1} を用いて，バーチャルスイッチ S_i の位置における重み付
 き平均 \bar{q}_i を，

$$\bar{q}_i = \frac{w_{i,1}^s q_1 + w_{i,2}^s q_2 + \dots + w_{i,i-1}^s q_{i-1}}{|w_{i,1}^s q_1 + w_{i,2}^s q_2 + \dots + w_{i,i-1}^s q_{i-1}|} \quad (3.3)$$

のように求め，このバーチャルスイッチ S_i の複数の姿勢の候補の中で，1 軸回転法に
 よってこの重み付き平均姿勢 \bar{q}_i に最も近いものをこのスイッチに対する実スイッチの

提示姿勢 q_i とする．ここで，重み $w_{i,k}^s$ は，

$$w_{i,k}^s = \bar{w}_{i,k}^s \cdot \tilde{w}_{i,k}^s \quad (3.4)$$

$$\bar{w}_{i,k}^s = \frac{1}{|\mathbf{p}_k^v - \mathbf{p}_i^v|^\alpha} \quad (3.5)$$

$$\tilde{w}_{i,k}^s = \sum_{j=1}^{i-1} |\mathbf{p}_k^v - \mathbf{p}_j^v| \quad (3.6)$$

により定める [22]．ただし， \mathbf{p}_k^v はバーチャルスイッチ S_k に指先が接触する位置の代表点であり，スイッチに触れる際の指先位置とする．式 (3.5) の重みは，提示姿勢を決めようとしているスイッチが既に提示姿勢が決まっているスイッチに近いときに値が大きくなるような重みであり， $\alpha (\geq 1)$ は距離による重みの変化の関係を調整するパラメータである． S_i が既に提示姿勢が決まったある S_k から距離が近いほど $\bar{w}_{i,k}^s$ の値が大きくなり， S_i の提示姿勢を決める際に S_k からの影響を大きくする．式 (3.6) の重みは，既に提示姿勢が決まっているスイッチが同じ姿勢で密集しているときに重みが空間的に偏るのを補償するための重みである． S_k が他の $S_j (j = 1, 2, \dots, i-1)$ と離れている場合は， S_k の周辺に S_j がある場合と比べて $\tilde{w}_{i,k}^s$ の値は大きくなり，それによって密集による重みの空間的な偏りを防ぐ．重みの空間的な偏りの補償の有無を比較した例を挙げて補償の意義について後述する．

図 3.2 は，以上で述べた任意性のある提示姿勢の決定法に基づいて， S_i の提示姿勢 q_i を決定する過程を模式的に示している．まず，既に提示姿勢が定まった S_1, S_2, \dots, S_{i-1} から \bar{q}_i を求め，次に，スイッチ S_i の提示姿勢 q_i の複数の候補の中から，1 軸回転法によって \bar{q}_i に最も近いものを選んで実際の q_i とする．

式 (3.6) の重み $\tilde{w}_{i,k}^s$ を用いた場合と用いない場合の比較を図 3.3 に示す．この例ではスイッチは 1 種類で，点 A に 1 個，点 B に 9 個，AB の中点である点 C に 1 個のバーチャルスイッチがある．点 A，点 B のバーチャルスイッチは既に提示姿勢が決まっているとし，それぞれ q_A, q_B であるとする．いま，点 C の位置にある 11 番目のスイッチの提示姿勢を決めるために点 C の位置で平均姿勢 \bar{q}_{11} を計算したとき， $\tilde{w}_{i,k}^s$ を用いずに代わりに定数 1 を用いたとすると，(a) のように点 B にある 9 個のスイッチの影響が大

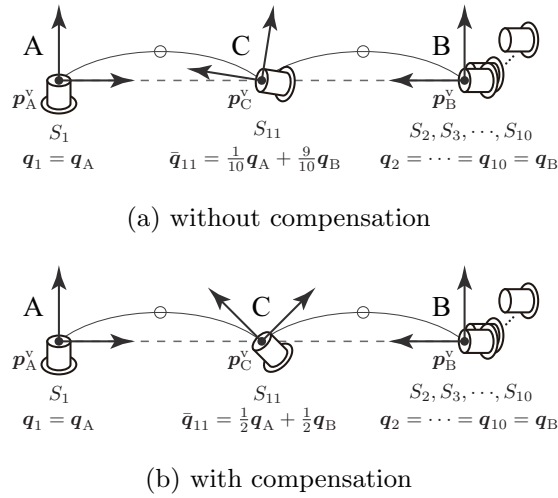


図 3.3 スイッチの密集による重みの空間的な偏りの補償

きく， \bar{q}_{11} はほとんど q_B に近くなってしまふ． $\tilde{w}_{i,k}^s$ を用いてスイッチの密集による重みの空間的偏りを補償した場合，(b) のように点 A，点 B にあるスイッチから距離に応じて均等に影響を受けて， \bar{q}_{11} は q_A と q_B の中間の値となる．このように $\tilde{w}_{i,k}^s$ を用いることによって，同じ姿勢のスイッチが密集している際に，重みの空間的偏りを補償して距離に応じて均等に影響を受けるようにできる．それによって，指先が同じ距離を動いたときのデバイス動作負担を均等化できる．

本節で述べた任意性のある提示姿勢の決定に際して，各バーチャルスイッチの提示姿勢はバーチャルスイッチ及び実スイッチの現在の状態の組合せに対して決めることができ，ユーザの手の位置には依存しない．よって，ユーザがバーチャルスイッチを切り替えるごとに決定アルゴリズムを実行することによって，各バーチャルスイッチに対する提示姿勢を決めることができる．あるいは，バーチャルスイッチ及び実スイッチの全ての状態の組合せについて予めオフライン計算によりデータとして蓄積しておき，状態が変化することによってデータを照会して用いることも可能である．

3.2.4 直交座標系に基づく動作計画

3.2.3 節で述べた方法により，各バーチャルスイッチ S_k ($k = 1, \dots, N$) を提示する際のデバイスの目標姿勢 \mathbf{q}_k が求められる．一旦 \mathbf{q}_k が求めれば，それにより対応するデバイスの目標位置 \mathbf{p}_k も求めることができる．以下では，現在のユーザの指先位置に応じてデバイスをどのように移動させるべきかについて述べる．

ユーザの指先を検出するセンサから現在の指先位置が \mathbf{p}_f であるとき，現時点においてデバイスを待機させるべき目標位置・姿勢 $\mathbf{p}_d, \mathbf{q}_d$ を以下の式により決定する．

$$\mathbf{p}_d = \frac{w_1(\mathbf{p}_f) \mathbf{p}_1 + w_2(\mathbf{p}_f) \mathbf{p}_2 + \dots + w_N(\mathbf{p}_f) \mathbf{p}_N}{w_1(\mathbf{p}_f) + w_2(\mathbf{p}_f) + \dots + w_N(\mathbf{p}_f)} \quad (3.7)$$

$$\mathbf{q}_d = \frac{w_1(\mathbf{p}_f) \mathbf{q}_1 + w_2(\mathbf{p}_f) \mathbf{q}_2 + \dots + w_N(\mathbf{p}_f) \mathbf{q}_N}{|w_1(\mathbf{p}_f) \mathbf{q}_1 + w_2(\mathbf{p}_f) \mathbf{q}_2 + \dots + w_N(\mathbf{p}_f) \mathbf{q}_N|} \quad (3.8)$$

ここに $w_k(\mathbf{p}_f)$ は k 番目のスイッチに対する重みで，次式で与えられる．

$$w_k(\mathbf{p}_f) = \bar{w}_k(\mathbf{p}_f) \cdot \tilde{w}_k \quad (3.9)$$

$$\bar{w}_k(\mathbf{p}_f) = f(r_k) \quad (3.10)$$

$$r_k = |\mathbf{p}_k^V - \mathbf{p}_f| \quad (3.11)$$

$$\tilde{w}_k = \sum_{j=1}^N |\mathbf{p}_k^V - \mathbf{p}_j^V| \quad (3.12)$$

ここで， r_k は指先とバーチャルスイッチ S_k の距離である．式 (3.10) において，指とバーチャルスイッチの距離 $r \geq 0$ に対する重み関数 $f(r)$ は，既報 [24] では式 (3.5) と同様の関数，

$$f(r) = \frac{1}{r^\alpha} \quad (3.13)$$

を用いていた．これは，指がバーチャルスイッチに近づいたとき，そのスイッチに対する重みが大きくなって提示デバイスがそのバーチャルスイッチの位置に近づいていき，指がバーチャルスイッチの位置に到達した場合には，重みが無限大になってバーチャルスイッチと実スイッチの位置が一致して，指が実スイッチに遭遇できるものである．ま

た，式 (3.12) は式 (3.6) と同様に重みの空間的偏りを補償するために導入されたものである．

3.2.5 飽和領域を持った重み関数の導入

動作計画において，指の位置がスイッチ間を移動すると重みが変わりデバイスへの提示位置・姿勢が連続的に変化するが，式 (3.13) の $f(r)$ を用いると，指がバーチャルスイッチに近づいた時に十分に重みが大きくなるように α を大きくしなければならず，それによってスイッチから離れるに従って生じる重みの変化が急になり，その結果デバイスの目標位置・姿勢に急激な変化が起こって動作負担が大きくなるという問題点があった．それを改善するために， $r = 0$ 付近に飽和領域を設けた重み関数，

$$g(r) = \begin{cases} \frac{1}{d^\alpha} & \text{if } r < r_0 + d \\ \frac{1}{(r - r_0)^\alpha} & \text{otherwise} \end{cases} \quad (3.14)$$

を導入し，式 (3.10) で $f(r)$ の代わりに用いる．ここで， r_0 及び d は適当な正の定数である． d を十分小さく取ることにより，指がバーチャルスイッチに $r = r_0 + d$ まで近づいたときに，そのスイッチに対する重みが十分大きくなり，デバイスがそのスイッチの提示位置・姿勢となる．よって， α は本来の目的通りに重みの変化が適切となるように調整すれば良いことになる．

飽和領域を持った重み関数を用いることによってデバイス動作負担がどのように軽減されるのか示すために，1次元の単純な状況を想定したシミュレーションを行い，従来の重み関数 $f(r)$ と提案する重み関数 $g(r)$ を用いた場合についてデバイス動作の比較を示す．ここでは，式 (3.10) の重み関数を変更した効果のみ比較するために，式 (3.12) の重みは用いずに定数 1 とする．また，このシミュレーションにより式 (3.7) によるデバイス位置決定の基本的な動作を確認することができる．スイッチは 1 種類で点として扱い，スイッチ及びデバイスの姿勢は考えないものとし，デバイス上の実スイッチの位置をデバイスの参照点とする．

バーチャルスイッチは 4 つとし， $x = 0, 0.15, 0.25, 0.45$ に配置する．重み関数のパ

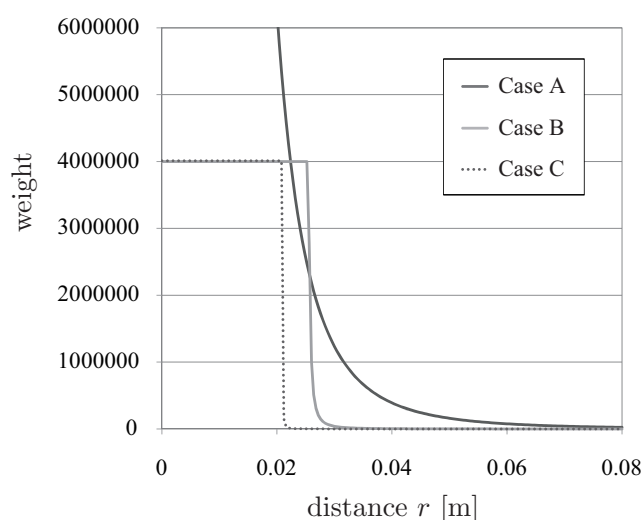
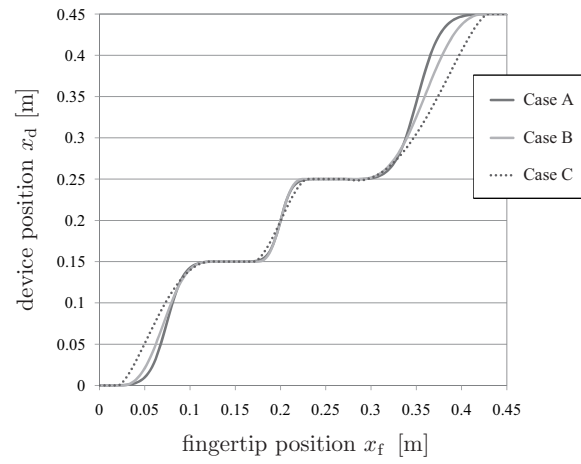


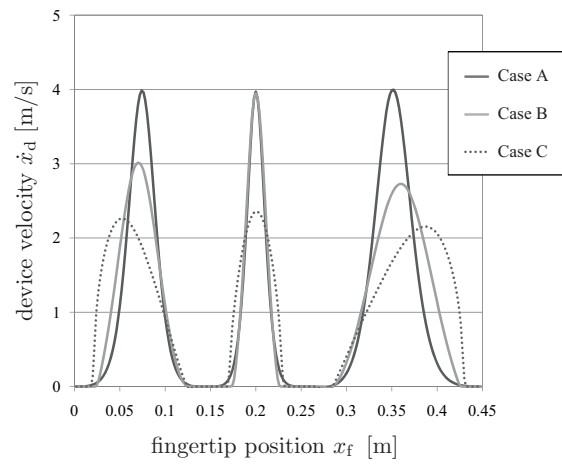
図 3.4 重み関数

ラメータは、スイッチ間の距離が最も近くて 0.1 [m] となる想定する最悪ケースにおいて調整することとし、 $x = 0.15, 0.25$ の間で指先を動かしてデバイス動作が適切となるように調整を行った。Case A ~ C の 3 種類の場合について比較を行う。Case A は $f(r)$ を用いてデバイス動作が最適となるようにパラメータの調整を行う。Case B は $g(r)$ を用いて Case A と同じ最大デバイス速度となるように調整を行う。Case C は $g(r)$ を用いて Case A と同様にデバイス動作が最適となるようにパラメータの調整を行う。調整後の各パラメータは、Case A では $\alpha = 4$ 、Case B では $\alpha = 2, r_0 = 0.025, d = 0.0005$ 、Case C では $\alpha = 1.4, r_0 = 0.021, d = 0.0000192$ となった。Case A ~ C の重み関数を r についてプロットすると図 3.4 のようになる。Case A ~ C について、指先速度 \dot{x}_f を一定速度の 1 [m/s] として $x_f = 0$ から 0.45 まで動かしてシミュレーションを行った。

シミュレーション結果が図 3.5 であり、(a) が指先位置 x_f に対してデバイス位置 x_d をプロットしたものであり、(b) が指先位置 x_f に対してデバイス速度 \dot{x}_d をプロットしたものである。(a) を見ると、指先がバーチャルスイッチの位置から離れるとデバイスが動き始め、指先が別のバーチャルスイッチに近づいていくと、デバイスの位置がそのバーチャルスイッチの位置に収束して待機する様子が分かる。(b) を見ると、Case A で



(a) device position



(b) device velocity

図 3.5 重み関数の比較

はどのスイッチ間においてもデバイスの最大速度は 4 [m/s] であるが, Case B では距離が 0.1 [m] よりも長いスイッチ間では Case A よりもデバイスの最大速度が小さくなっており, 動作負荷が軽減されている. また, Case C ではいずれのスイッチ間においても Case A よりデバイスの最大速度が小さくなっており, 全体的に動作負荷が軽減されている.

まずシミュレーション結果において、指先の移動に応じてデバイスがバーチャルスイッチ間を連続的に移動し、指先がバーチャルスイッチの位置に近づいていくとデバイスが提示位置で待機して、提示位置で指先とデバイスが適切に遭遇できており、式 (3.7) によるデバイス位置決定の基本的な動作が確認できた。また結果から、重み関数 $f(r)$ と $g(r)$ において、最悪ケースとなるスイッチ間の距離が近い状況で同様のパフォーマンスとなるように調整を行った場合でも、スイッチ間の距離が遠い状況では $g(r)$ を用いた方がスイッチ間の移動におけるデバイスの最大速度が小さくなることが分かる。また、 $g(r)$ を用いてデバイス動作が最適となるように調整を行った場合、 $f(r)$ を用いて同様に最適となるように調整した場合に比べて、どのスイッチ間においても最大デバイス速度が小さくなり、全体的に動作負担が軽減されるといえる。

3.2.6 バーチャルパネル面座標系に基づく動作計画

3.2.4 節で述べた動作計画手法を用いたとき、ユーザが指先をスイッチ間で移動させている途中にデバイスと干渉する恐れがあり、既報 [24] ではそのような干渉を避けるために別途衝突回避動作を導入する必要があった。この節では、デバイスの動作をバーチャルパネル面に沿って拘束することにより通常操作中の不用意な衝突を完全に回避する動作計画手法を提案する。

図 3.6 は、バーチャル操作パネルが 2 つの面で構成されて凹み部を持ち、2 つのバーチャルスイッチが別々の面に配置されている場合に、直交座標系に基づく動作計画とバーチャルパネル面座標系に基づく動作計画を用いたとき、デバイスの目標提示位置・姿勢を比べたものである。直交座標系に基づく動作計画の (a) では、現在の指先位置 p_f に対するデバイスの目標位置は、バーチャルスイッチ 1 の提示位置とバーチャルスイッチ 2 の提示位置を線形補間した位置に来るため、操作パネルの凹み部でデバイスがパネル面の手前に出てくることがあり、指とデバイスが干渉する危険性があった。一方、バーチャルパネル面座標系に基づく動作計画の (b) では、デバイスはバーチャルパネル面座標系に沿って動くため、ユーザとデバイスが不用意な衝突を起こすことがない。以下で提案する動作計画手法について詳細を述べる。

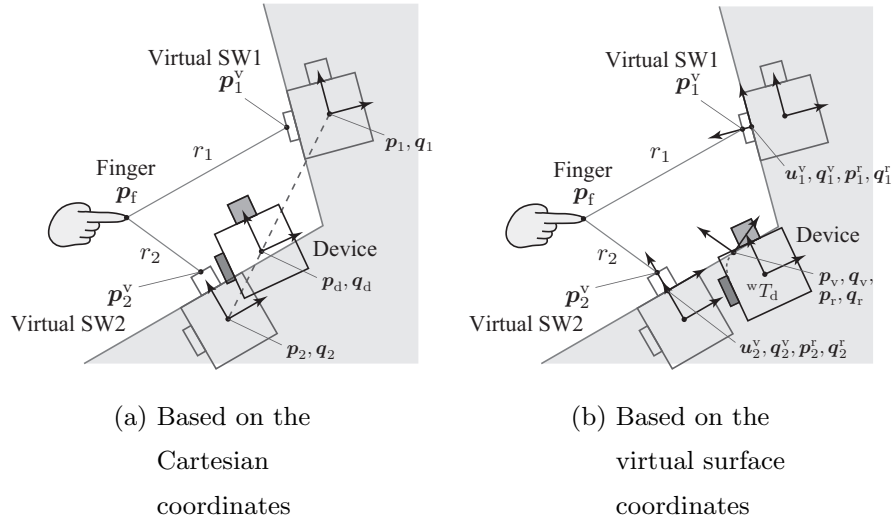


図 3.6 動作計画手法の比較

バーチャルパネル面座標系の 2 次元パラメータ u を用いて、デバイスで提示しようとしているバーチャル操作パネル面を含むように面 $p = \phi_v(u)$ を定義する．このときに、指の位置 p_f を与えて現在のデバイスの目標提示位置・姿勢を計算する方法を以下に示す．ワールド座標系 Σ_w から見たデバイスの位置・姿勢を表す同時変換行列 wT_d を、

$${}^wT_d = {}^wT_s {}^sT_d = {}^wT_s ({}^dT_s)^{-1} \quad (3.15)$$

により求めるものとする．つまり、まず Σ_w から見たバーチャルパネル面上で現在の指先位置に対応する補間されたスイッチの位置・姿勢 wT_s を求めておき、次にデバイス座標系 Σ_d から見てどの位置・姿勢を先に求めた補間位置・姿勢に対応付けするかを dT_s として与え、最終的に式 (3.15) によりデバイスの目標位置・姿勢を得る． wT_s 及び dT_s は、座標系原点から見た位置ベクトル p 及び姿勢を表すクォータニオン q から同時変換行列を生成する操作 $T(p, q)$ を用いて、

$${}^wT_s = T(p_v, q_v) \quad (3.16)$$

$${}^dT_s = T(p_r, q_r) \quad (3.17)$$

のように計算される．ただし，式 (3.16) の $\mathbf{p}_v, \mathbf{q}_v$ は，

$$\mathbf{p}_v = \phi_v(\mathbf{u}_v) \quad (3.18)$$

$$\mathbf{u}_v = \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{u}_i^v \right) / \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \right) \quad (3.19)$$

$$\mathbf{q}_v = \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{q}_i^v \right) / \left| \sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{q}_i^v \right| \quad (3.20)$$

により，また式 (3.17) の $\mathbf{p}_r, \mathbf{q}_r$ は，

$$\mathbf{p}_r = \gamma r_r \mathbf{u}_r \quad (3.21)$$

$$\mathbf{u}_r = \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{u}_i^r \right) / \left| \sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{u}_i^r \right| \quad (3.22)$$

$$r_r = \left(\sum_{i=1}^N w_i(\mathbf{p}_f) |\mathbf{p}_i^r| \right) / \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \right) \quad (3.23)$$

$$\mathbf{q}_r = \left(\sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{q}_i^r \right) / \left| \sum_{i=1}^N w_i(\mathbf{p}_f) \mathbf{q}_i^r \right| \quad (3.24)$$

により計算される． \mathbf{u}_v は各バーチャルスイッチ S_i のバーチャルパネル面座標系における位置 \mathbf{u}_i^v の加重平均であり， \mathbf{p}_v はそれに対応するバーチャルパネル面上の点の 3 次元空間位置ベクトルである． \mathbf{q}_v は Σ_w から見た各バーチャルスイッチ S_i の姿勢を表すクォータニオン \mathbf{q}_i^v を加重平均したのち正規化したものである． \mathbf{u}_r は，各バーチャルスイッチ S_i に対応する実スイッチの Σ_d から見た位置ベクトル \mathbf{p}_i^r を単位化したベクトル $\mathbf{u}_i^r (= \mathbf{p}_i^r / |\mathbf{p}_i^r|)$ を加重平均したのち単位化したベクトルである． r_r は位置ベクトル \mathbf{p}_i^r のノルムを加重平均したものである． \mathbf{q}_r は Σ_d から見た各バーチャルスイッチ S_i に対応する実スイッチの姿勢を表すクォータニオン \mathbf{q}_i^r を加重平均したのち正規化したものである．

また， γ はデバイスの外形がデバイス上の実スイッチの位置を補間した点 $r_r \mathbf{u}_r$ より外に出ているときに，デバイス座標系の位置に係数を掛けることによりデバイス上の提示

位置を外にずらし，デバイスをバーチャルパネル面より後退させるための係数であり，

$$\gamma = \frac{w_{\text{sum}} + a w_0}{w_{\text{sum}} + w_0} \quad (3.25)$$

により計算される．ただし， $w_0 (> 0)$ 及び $a (\geq 1)$ は定数である． w_{sum} は，

$$w_{\text{sum}} = \sum_{i=1}^N w_i(\mathbf{p}_f) \quad (3.26)$$

で計算される重みの総和である． w_{sum} が w_0 に比べて十分大きいときには $\gamma \rightarrow 1$ となり，十分小さいときには $\gamma \rightarrow a$ となる．これにより，指がスイッチに近づいたときに提示位置が実スイッチの位置になり，指がいずれのスイッチからも遠いときにはデバイス座標系から見た提示位置ベクトルを最大 a 倍まで延長することによりバーチャル操作パネル面よりさらにデバイスを後退させる作用を持つ．本節で述べた提案する動作計画手法で用いる重みについては，3.2.4 節及び 3.2.5 節で述べたのと同様のものを用いる．

図 3.7 は提案手法においてデバイスの提示位置・姿勢を決定する様子を示している．(a) がバーチャルパネル面座標系においてバーチャルスイッチの位置・姿勢を補間する様子であり，バーチャルスイッチ 1 の位置からバーチャルスイッチ 2 の位置まで自由空間で指を動かしているとしており，それに対応して①から⑦の位置・姿勢が求まる．そのとき，(b) が，デバイス座標系においてバーチャルスイッチの補間位置・姿勢に対応付けさせるデバイス上の位置・姿勢を与える様子である．図ではスイッチボックスは正方形であるが，提案手法ではスイッチボックスは球座標系で連続な一価関数で表現できる形状であれば良いことに注意されたい．(a)，(b) の対応付けからデバイスの動作計画として得られるデバイスの位置・姿勢が (c) に示されている．

本節で示したバーチャルパネル面座標系に基づく動作計画を用いることによって，バーチャル操作パネルの手前でスイッチを操作するなどの通常の作業を行う場合には完全にユーザとデバイスの衝突を回避することができる．ただし，バーチャルパネル面より奥に手を進入させた場合についてはユーザとデバイスが干渉する余地があり，緊急停止措置などの対策が必要である．また，ベースモジュールとして用いるロボットアーム部分が配置上バーチャル操作パネルより前面に出てくる可能性を排除できない場合は，別途アーム部分に対応した回避動作を導入する必要がある．

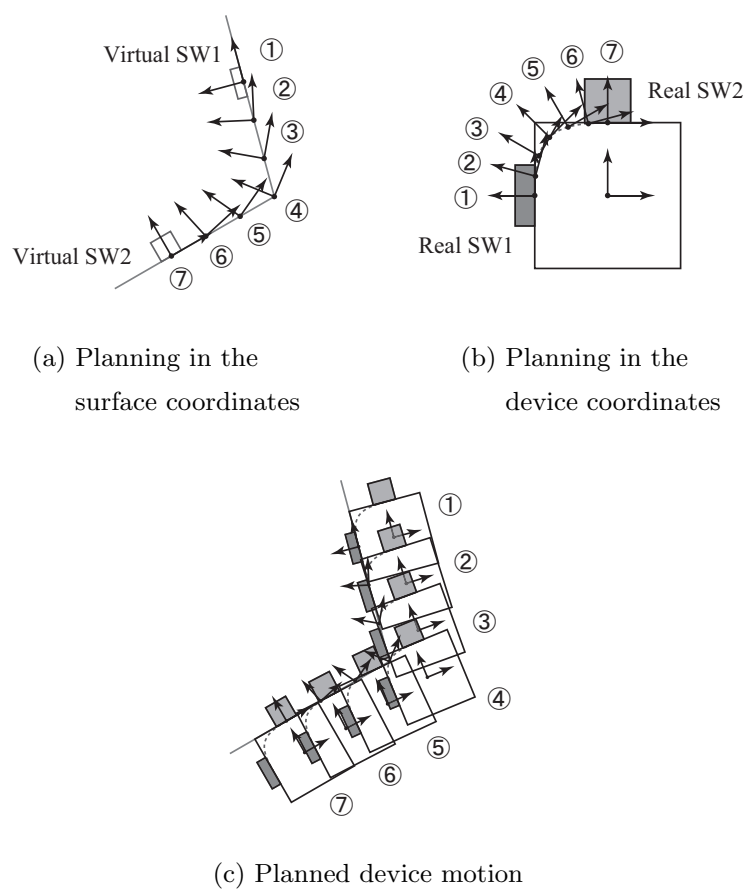


図 3.7 バーチャルパネル面座標系に基づく動作計画

3.3 提案手法の検証

3.3.1 シミュレーション

次に、実際的なバーチャル操作パネルとシステムの構成を想定して、シミュレーションにより提案する動作計画手法の検証を行う。想定した状況は図 3.8 に示すようなものである。バーチャル操作パネルは図 3.8 (a) に示すように、2 枚の平面パネルが 60 度の角度を有して配置されており、正方形モーメンタリスイッチ、2 極対称トグルスイッチ、

無限ロータリスイッチの3種類のスイッチが計12個配置されている。スイッチボックスは図3.8 (b) に示すように、一辺が0.1 [m] の立方体を面取りした形状とし、 $+x$ 方向の面に正方形モーメンタリスイッチを配置し、 $+y$ 方向の面に無限ロータリスイッチを配置し、 $+z$ 方向の面に2極対称トグルスイッチを配置した。スイッチボックスを取り付けるベースモジュールとしては三菱重工業社製汎用ロボットアーム PA10 を想定する。PA10 は本来7自由度であるが第3軸を固定した6自由度ロボットアームとして制御を行う。バーチャル操作パネルと PA10 の位置関係は図3.8 (c) に示すように配置し、矢印形状のオブジェクトによりユーザの指先を表している。

トグルスイッチを切り替えるとバーチャルスイッチの状態と実スイッチの状態が切り替わり、提示姿勢に法線軸回りの任意性がある正方形スイッチとロータリスイッチに関して、3.2.3 節で述べた方法でデバイスの動作負担を考慮して決めている最適な提示姿勢がそれぞれの状態の組合せごとに求められる。全てのバーチャルスイッチの提示位置・姿勢を用いて、3.2.4 節で述べた直交座標系に基づく動作計画、または3.2.6 節で述べたバーチャルパネル面座標系に基づく動作計画によって、デバイスの現在の目標提示位置・姿勢が求められる。重み関数は3.2.5 節で述べた $g(r)$ を用いた。パラメータは動作が最適となるように調整し、 $\alpha = 1.6$, $r_0 = 0.021$, $d = 10^{-12}$, $a = 2.3$, $w_0 = 150$ とした。比較のために $f(r)$ を用いた際には、パラメータは同様に動作が最適となるように調整し、 $\alpha = 4$, $a = 2.5$, $w_0 = 40000$ とした。シミュレーションは OpenGL を用いたグラフィック表示により確認を行った。

実装した動作計画アルゴリズムがユーザ入力に対して適切に動作するかを確認するため、マウスを用いてユーザの指先を移動させ、キーボード入力によりスイッチを切り替えるような、動作検証用インタフェースを PC 上に作成し、手動入力により動作確認を行った。まず、3.2.4 節及び3.2.6 節の動作計画アルゴリズムが適切に機能していることを確認した。指先の移動に連動して滑らかなデバイス動作が生成され、また、全てのバーチャルスイッチの位置で指先がデバイス上の実スイッチと適切に遭遇できることを確認した。また、スイッチを切り替えるごとに3.2.3 節の提示姿勢決定アルゴリズムに基づいて各バーチャルスイッチに対する提示姿勢が適切に再計算されることを確認した。

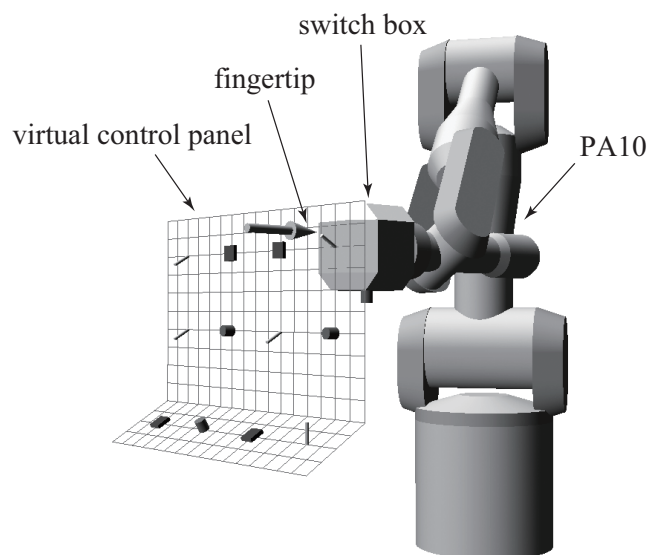
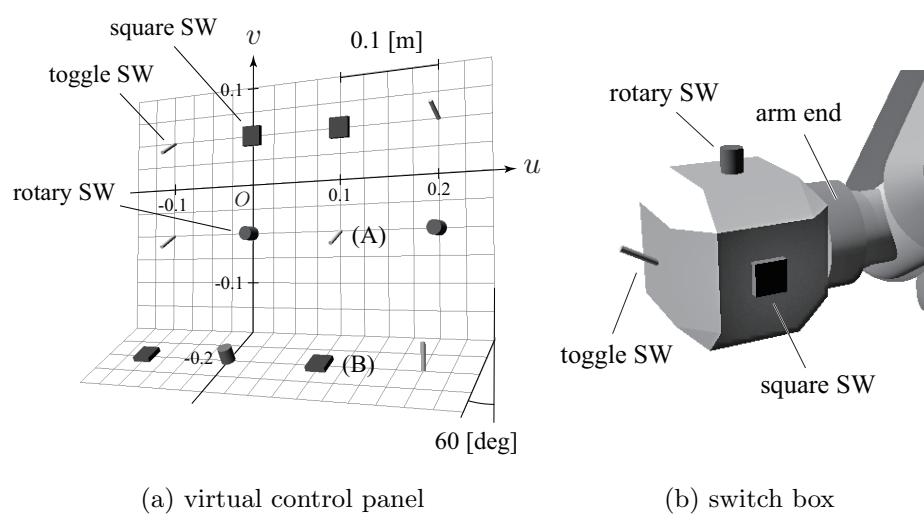


図 3.8 シミュレーションの設定

3.3.2 衝突回避動作の検証

検証を自動化するために、図 3.9 に示したような各バーチャルスイッチへの接触位置を線形補間した指先軌道を計算により与えてシミュレーションを行った。図はバーチャ

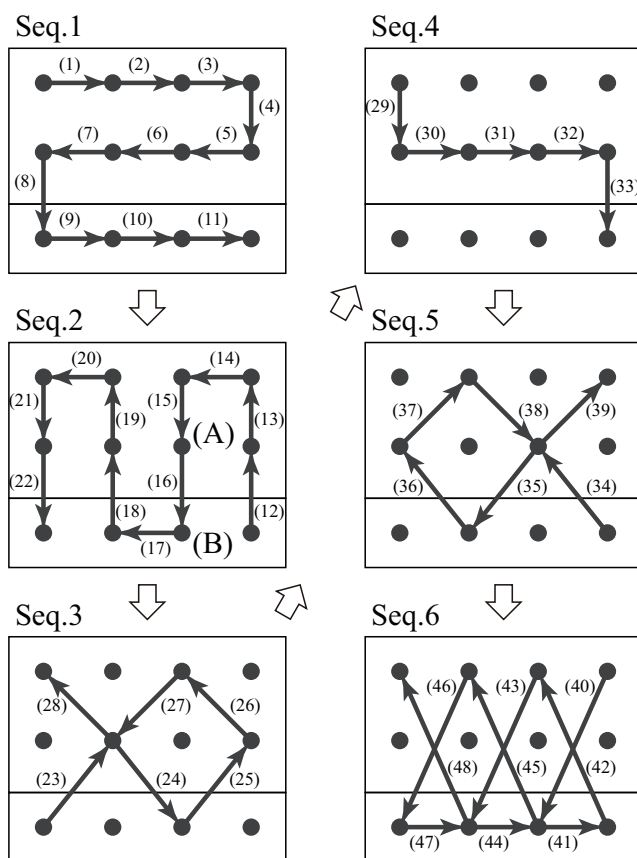
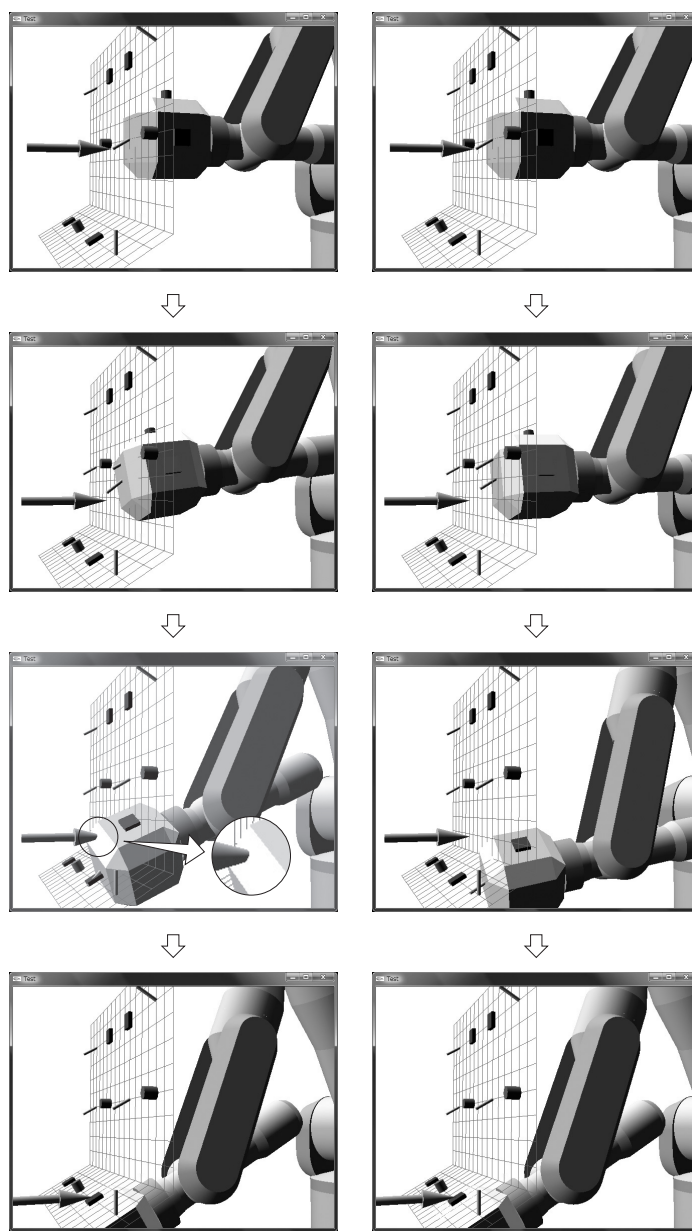


図 3.9 操作の指先軌道

ル操作パネルを正面から見たものであり，黒丸は各バーチャルスイッチを表わし，矢印が指先の動きを表わしている．図 3.9 の指先軌道は，全体を通して実行することで縦・横・斜めに隣接する各スイッチ間を 1 回以上遷移するよう構成されている．直交座標系に基づく動作計画ではデバイスの移動途中で指先と干渉する場面があったが，バーチャルパネル面座標系に基づく動作計画を用いた場合，全てのバーチャルスイッチ間遷移で指先がデバイスと干渉しないことを確認した．

シミュレーション結果の一部を図 3.10 に示す．図 3.9 で示した指先軌道の (16) の部分に相当し，上の画像から順に，(A) のスイッチから (B) のスイッチに指先を移動させ

ている様子を示している。直交座標系に基づく動作計画を用いたシミュレーション結果が (a) であり、バーチャルパネル面座標系に基づく動作計画を用いたシミュレーション結果が (b) である。直交座標系に基づく動作計画を用いた場合、(a) の 3 枚目の画像のようにデバイスが動作中にバーチャル操作パネルの凹部分においてパネル面より突出して指先と干渉を起こしており、動作中にユーザとデバイスの不用意な衝突が発生して好ましくない。それに対して、バーチャルパネル面座標系に基づく動作計画では (b) の画像のようにデバイスはバーチャル操作パネル面に沿って動くので、ユーザが操作パネル面の前面で作業を行っている限りユーザとデバイスが干渉することはない。



(a) Based on the Cartesian coordinates

(b) Based on the virtual surface coordinates

図 3.10 シミュレーションのスクリーンショット

3.3.3 動作負荷の検証

3.3.1 節で設定した実際的なシステム構成を想定したシミュレーションによって、提案する動作計画を適用した際のデバイス動作負荷の検証を行う。動作計画を行う際に、3.2.3 節で述べたような任意性のある提示姿勢の決定を適切に行うことによって、デバイスの動作負荷を軽減できることを検証する。また、3.2.5 節で述べたように、動作計画で用いる重み関数に飽和領域を導入することによってデバイスの動作負荷を軽減できることを検証する。動作計画には 3.2.6 節で述べたバーチャルパネル面座標系に基づく動作計画手法を用いるが、比較のために 3.2.4 節で述べた直交座標系に基づく動作計画手法についても評価を行った。

PA10 の関節角速度を可能な最大値で正規化して、全関節について絶対値の最大値を取って瞬間の負荷率とする。図 3.9 で示した指先軌道を与えてシミュレーションを行った。図 3.9 の指先軌道は、一連の動作を通して実行することで各スイッチ間を 1 回ないしは 2 回遷移し、空間的な偏りがないように構成されている。Seq.1~5 については各スイッチ間を等速で直線的に 1 [sec] かけて指先を移動させ、Seq.6 については同様に 2 [sec] で移動させ、全体で 57 [sec] の一連の指先軌道とした。

FCf, OCf, OCg, FSf, OSf, OSg の全 6 ケースについてシミュレーションを行った。F (fixed) は 3.2.3 節の述べた任意性のある提示姿勢の決定アルゴリズムを用いずに、各種類のスイッチごとに予め決められた 1 つの提示姿勢を用いる場合を表わす。O (optimized) は任意性のある提示姿勢の決定アルゴリズムを用いて提示姿勢を最適化する場合を表わす。C (Cartesian) は直交座標系に基づく動作計画、S (surface) はバーチャルパネル面座標系に基づく動作計画を用いることを表わす。f 及び g は、重み関数としてそれぞれ $f(r)$ 及び $g(r)$ を用いることを表わす。

図 3.11 は、各ケースについて各時刻の負荷率をプロットしたものである。矢印で指している灰色で描かれた区間は、そのスイッチ間遷移で特異姿勢に近づいたことを示している。PA10 の第 6 軸の関節角が $-7 \sim 7$ [deg] 内となったときに特に関節角速度が大きくなっている場合が多かったため特異姿勢に近づいたと見なした。

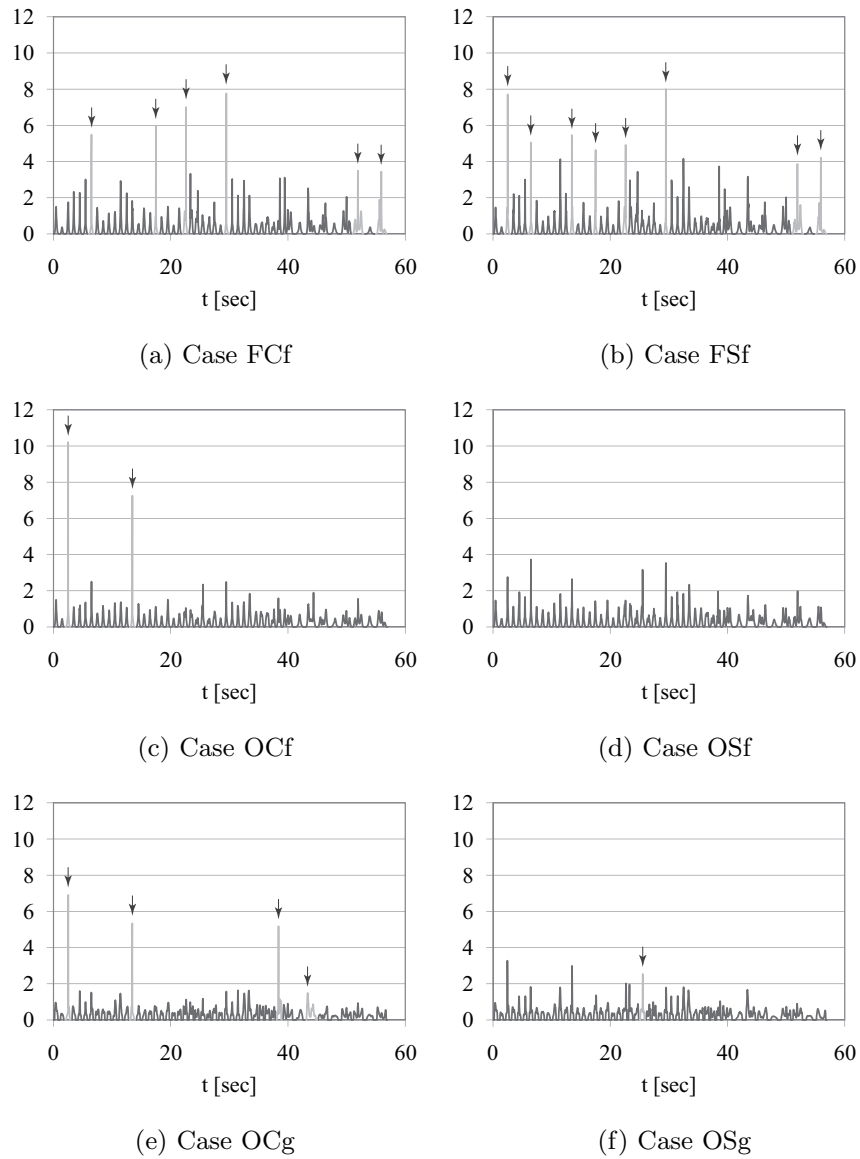


図 3.11 関節角速度負荷率

最大負荷と平均負荷を比較するために、特異姿勢に近づいた区間を除いて、各ケースで最大値と時間平均値を取り、棒グラフを描いたのが図 3.12 である。括弧内はそれぞれ OCf を 100% とした比率を表わしている。図 3.12(a) の最大負荷において、FCf, FSf

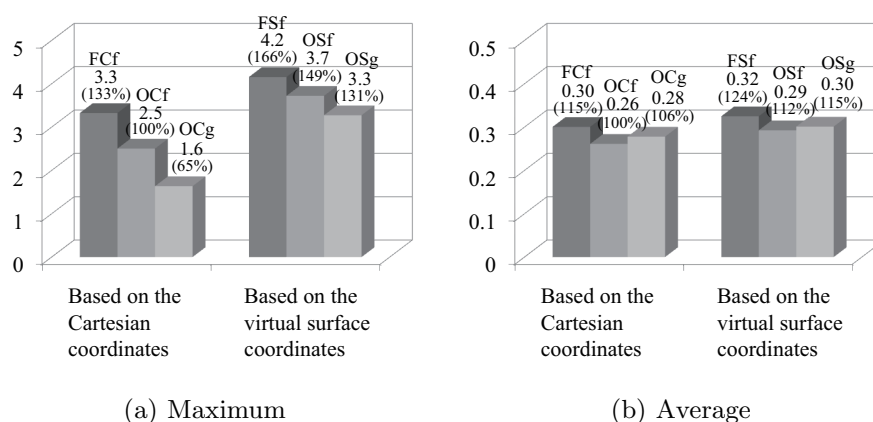


図 3.12 動作負荷の比較

に比べて OCf, OSf の負荷がそれぞれ小さくなっていることから, OCf, OSf においては提示姿勢の任意性を利用することによって動作負荷を軽減できていることが分かり, 3.2.3 節で述べた配置の任意性を利用した提示姿勢決定アルゴリズムの有効性が確かめられた. また, OCf, OSf に比べて OCg, OSg の最大負荷がそれぞれ小さくなっていることから, 重み関数への飽和領域導入によって動作負荷を軽減できていることが確かめられた. また, FCf, OCf, OCg に比べて FSf, OSf, OSg が全体的に大きくなっており, 直交座標系に基づく動作計画よりもバーチャルパネル面座標系に基づく動作計画を用いた方が最大負荷が大きいことが分かる. バーチャルパネル面座標系に基づく動作計画を用いるとバーチャルパネル面に沿ってデバイスが動くので, ユーザに対して衝突回避動作を実現できるが, 直交座標系に基づく動作計画の直線的な動作に比べて余分にデバイスが動くため, 動作負担が大きくなっている. 図 3.12(b) の平均負荷において, FCf に比べて OCf と OCg が, FSf に比べて OSf と OSg がそれぞれ少し小さくなっているが, 全てのケースでほぼ 0.3 前後で大きな差はないと言える.

3.4 結言

本論文では、バーチャル空間内に構成された複数の種類のスイッチを有する操作パネルにおいてスイッチ類を操作する感覚を提示するシステムの構築するために、遭遇型ハプティックデバイスについて動作負担を軽減する効率的な動作計画手法を提案した。デバイスの動作計画は既報 [24] で述べたように、重田ら [22] が提案した最近接点情報を用いた動作計画を基にスイッチ提示システムに合わせた適用を行った。まず、デバイスの設計や動作計画を見通しよく行えるように、提示すべき状態の区別と法線軸回りの配置の任意性という2つの観点からスイッチの分類を行った。次に、各バーチャルスイッチの提示姿勢に法線軸回りの配置の任意性がある場合に、配置の任意性を有効に利用して提示姿勢を決めることでデバイスの動作負担を軽減できるような提示姿勢の決定手法を示した。その次に、基本的なデバイスの動作計画手法として、直交座標系に基づく動作計画手法を示した。デバイスの動作負担を軽減するために動作計画において目標提示位置・姿勢の計算に用いる重み関数に飽和領域を導入することで改良を行った。また、バーチャルパネル面座標系を導入することにより、バーチャルパネル面に沿ってデバイスが動くことでユーザとの不用意な衝突が起こらない動作計画手法を提案した。実際的なシステム構成を想定したシミュレーションにより提案手法を検証し、提案する動作計画手法によってユーザとデバイスの衝突をパネル前面で完全に回避でき、また、デバイス動作負担を軽減できることを確認し、提案手法の有効性を示した。

今後の課題としては、デバイスの動作中にベースモジュールであるロボットアームが特異姿勢に近づいた場合に、目標関節角速度の大きさがが実現不可能な程度に大きくなってしまふという問題が挙げられる。これに対して、デバイスの機構に冗長自由度を持たせて特異姿勢を回避するなどの対策が必要と考えられる。実験による検証や、デバイスの動作負担をより軽減するためにスイッチボックスやベースモジュールについてより良い機構を検討することも必要である。また、本論文のシミュレーションではベースモジュール部分がバーチャル操作パネル面より前面に突出することはなく、ベースモジュールとユーザの干渉は問題とならなかったが、配置上干渉が回避できない場合の為

に、ベースモジュール部分を考慮した衝突回避動作も検討する必要がある。

本論文では、バーチャル操作パネル上の複数スイッチ提示システムを対象とし、1本指でのシステムを考慮したのみであったが、任意形状物体を扱う力触覚 VR システムへの応用や多指システムへの拡張も可能である。また、提案した動作計画手法は、一般のロボットに対して複数の目標位置・姿勢が与えられた場合に、それらの間を滑らかに変化させる動作計画にも応用が可能であり、産業ロボットやサービスロボットへの適用なども考えられる。

第 4 章

最小原理を応用した機械システム の最適軌道計画

4.1 緒言

本章では、デバイスの最適軌道計画問題を最適制御問題として扱い、その一般解法について考える。最適制御問題に Pontryagin の最小原理を適用した場合、一般に常微分方程式の 2 点境界値問題に帰着されるが、本章では随伴変数の初期値を用いて媒介変数表示し、直接解法により厳密に解く手法を提案する。提案手法は従来手法に比べて解の精度や求解に掛かる計算時間の面で優れている。

1.6 節で述べたような従来解法の問題点を解決する最適制御問題の解法として、最小原理により問題のパラメトライゼーションを行う方法を提案する。最適制御入力を、最適性の必要条件である最小原理に基づいて得られる、あるいは動的計画法より求まる偏微分方程式にある仮定を置くことにより得られる常微分方程式、正準方程式を用いて表される形で限定することで、随伴変数の初期時刻における値を最適化変数として最適化を行うことができる。最小原理によれば、最適制御入力は必要条件として各時刻においてハミルトン関数を最小化することを要求するので、ハミルトン関数を各時刻で最小化する入力の範囲で評価関数を最適化する制御入力を探索すれば十分である。

評価関数として最適制御問題の評価関数を用いて直接最適化を行う方法が考えられる。

また，最適制御問題は最小原理によって常微分方程式の2点境界値問題を解くことに帰着されるが，2点境界値問題の解法として，境界条件の誤差ベクトルにより評価関数を定義し，その最適化により境界条件を満たす2点境界値問題の解を求める方法が考えられる．本研究ではこの2つの手法を提案し，その適用例を示して有効性の検討を行う．

次節以降では，まず4.2節において最適制御問題の一般解法として提案する2つの手法について述べる．次に4.3節において，1次元の最適制御問題であるNewtonの最小ドラッグノーズシェイプ問題を対象として，提案手法と従来手法を用いて求解を行い，解の精度と計算時間について比較し，提案手法の有効性について検証を行う．その後，4.4節において機械システム的最適軌道計画問題の設定と随伴変数初期値最適化法の適用例を示す．ここでは機械システムとして特にロボット・多リンク機構系などを対象とし，一般化座標を用いたLagrangeの運動方程式で表現できるシステムを考える．4.4.1節において随伴変数初期値最適化法を実際のロボット・多リンク機構系などの機械システムに適用するための式展開を示し，4.4.2節において1自由度ロボット・リンク系における提案手法を適用した数値例を示し，提案した2つの手法の比較について述べる．4.4.3節において，提案手法を用いることで，随伴変数の初期値をある範囲においてメッシュで切って評価関数の値を計算可能なことを示す．このことは，随伴変数初期値空間における評価関数の形状が分かり，極値や最適解の位置を調べることができ，そのように得られるデータを適切に取り扱えば，擬似的な全探索を行えることを意味している．

4.2 最適制御問題の随伴変数初期値最適化による解法

4.2.1 最適制御問題の定式化

制御対象システムの状態変数を $x = x(t)$ ，入力変数を $u = u(t)$ として，システムの状態方程式が，

$$\dot{x}(t) = f(x(t), u(t)) \quad (4.1)$$

で表されるとし，時刻 $t = 0$ における初期状態 $x(0)$ が，

$$x(0) = x_0 \quad (4.2)$$

であるときに，制御時間区間 $[0, T]$ における評価関数 J を，

$$J = L_f(\mathbf{x}(T)) + \int_0^T L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (4.3)$$

と定義して，許容制御入力を U とするとき， $t \in [0, T]$ における最適制御入力 $\mathbf{u}^*(t)$ は，

$$\min_{\mathbf{u}; \mathbf{u} \in U} J \quad (4.4)$$

の解として定義される．

4.2.2 最小原理による制御入力のパラメトライゼーション

随伴変数 $\mathbf{p} = \mathbf{p}(t)$ を用いて，ハミルトン関数 $H = H(\mathbf{x}, \mathbf{u}, \mathbf{p})$ を，

$$H(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \mathbf{p}^T \mathbf{f}(\mathbf{x}, \mathbf{u}) + L(\mathbf{x}, \mathbf{u}) \quad (4.5)$$

と定義すると，最小原理 [36] により正準方程式は，

$$\dot{\mathbf{x}} = \frac{\partial H}{\partial \mathbf{p}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (4.6)$$

$$\dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{p}) = -\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u})\right)^T \mathbf{p} - \frac{\partial L}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{u}) \quad (4.7)$$

のように導かれる．ただし，制御入力 $\mathbf{u}(t)$ は各時刻 t におけるハミルトン関数最小化問題，

$$\min_{\mathbf{u}; \mathbf{u} \in U} H(\mathbf{x}, \mathbf{u}, \mathbf{p}) \quad (4.8)$$

の解 $\mathbf{u}(t) = \bar{\mathbf{u}}(\mathbf{x}(t), \mathbf{p}(t))$ とする．このとき，随伴変数 $\mathbf{p}(t)$ の時刻 $t = 0$ における初期値を $\mathbf{p}(0) = \mathbf{p}_0$ として，それにより， $\mathbf{u}(t) = \mathbf{u}(\mathbf{p}_0, t)$ ， $\mathbf{x}(t) = \mathbf{x}(\mathbf{p}_0, t)$ ， $\mathbf{p}(t) = \mathbf{p}(\mathbf{p}_0, t)$ とすると，

$$\mathbf{u}(\mathbf{p}_0, t) = \bar{\mathbf{u}}(\mathbf{x}(\mathbf{p}_0, t), \mathbf{p}(\mathbf{p}_0, t)) \quad (4.9)$$

$$\mathbf{x}(\mathbf{p}_0, t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\mathbf{p}_0, \tau), \mathbf{u}(\mathbf{p}_0, \tau)) d\tau \quad (4.10)$$

$$\mathbf{p}(\mathbf{p}_0, t) = \mathbf{p}_0 + \int_0^t \mathbf{g}(\mathbf{x}(\mathbf{p}_0, \tau), \mathbf{u}(\mathbf{p}_0, \tau), \mathbf{p}(\mathbf{p}_0, \tau)) d\tau \quad (4.11)$$

のように，制御入力 $u(p_0, t)$ として随伴変数の初期値 p_0 によりパラメトライズされる．最小原理は最適の必要条件であるので，最適制御入力 $u^*(t)$ は p_0 によりパラメトライズされた制御入力 $u(p_0, t) = \bar{u}(x(p_0, t), p(p_0, t))$ に含まれる．

また，最小原理より導かれる制御区間終端時刻において満たされるべき境界条件は，

$$p(p_0, T) = \frac{\partial L_f}{\partial x}(x(p_0, T)) \quad (4.12)$$

となる．

4.2.3 評価関数の直接最適化による解法（1つ目の提案手法）

評価関数 $J = J(p_0, T)$ を，

$$J(p_0, T) = L_f(x(p_0, T)) + \int_0^T L(x(p_0, t), u(p_0, t)) dt \quad (4.13)$$

として，評価関数 $J(p_0, T)$ を直接最適化する最適化問題，

$$\min_{p_0} J(p_0, T) \quad (4.14)$$

の解 p_0^* ，あるいは終端時間自由の場合は，

$$\min_{p_0, T} J(p_0, T) \quad (4.15)$$

の解 p_0^*, T^* により，最適制御入力 $u^*(t)$ は時間区間 $[0, T]$ あるいは $[0, T^*]$ において，

$$u^*(t) = \bar{u}(x(p_0^*, t), p(p_0^*, t)) \quad (4.16)$$

$$x(p_0^*, t) = x_0 + \int_0^t f(x(p_0^*, \tau), u^*(\tau)) d\tau \quad (4.17)$$

$$p(p_0^*, t) = p_0^* + \int_0^t g(x(p_0^*, \tau), u^*(\tau), p(p_0^*, \tau)) d\tau \quad (4.18)$$

と表される．

また，評価関数 $J(p_0, T)$ を実際に最適化するために， p_0 及び T を微小変化させたときの J への影響を表す偏微分係数 $J_P = \frac{\partial J}{\partial p_0}$ ， $J_T = \frac{\partial J}{\partial T}$ を考える．ここで $U_P = \frac{\partial u}{\partial p_0}$ ，

$\mathbf{X}_P = \frac{\partial \mathbf{x}}{\partial \mathbf{p}_0}$, $\mathbf{P}_P = \frac{\partial \mathbf{p}}{\partial \mathbf{p}_0}$ とすると, 式 (4.16) ~ (4.18) より,

$$\mathbf{U}_P = \frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{x}} \cdot \mathbf{X}_P + \frac{\partial \bar{\mathbf{u}}}{\partial \mathbf{p}} \cdot \mathbf{P}_P \quad (4.19)$$

$$\dot{\mathbf{X}}_P = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \cdot \mathbf{X}_P + \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \cdot \mathbf{U}_P \quad (4.20)$$

$$\dot{\mathbf{P}}_P = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \cdot \mathbf{X}_P + \frac{\partial \mathbf{g}}{\partial \mathbf{u}} \cdot \mathbf{U}_P + \frac{\partial \mathbf{g}}{\partial \mathbf{p}} \cdot \mathbf{P}_P \quad (4.21)$$

となる. \mathbf{X}_P , \mathbf{P}_P の時刻 $t = 0$ における初期値は, 式 (4.17), (4.18) より,

$$\mathbf{X}_P(\mathbf{p}_0, 0) = \mathbf{O} \quad (4.22)$$

$$\mathbf{P}_P(\mathbf{p}_0, 0) = \mathbf{I} \quad (4.23)$$

である. 式 (4.19) ~ (4.23) を用いて, \mathbf{J}_P , \mathbf{J}_T は,

$$\mathbf{J}_P(\mathbf{p}_0, T) = \frac{\partial L_f}{\partial \mathbf{x}}(\mathbf{x}(\mathbf{p}_0, T)) \cdot \mathbf{X}_P(\mathbf{p}_0, T) + \int_0^T \mathbf{L}_P(\mathbf{p}_0, t) dt \quad (4.24)$$

$$\mathbf{J}_T(\mathbf{p}_0, T) = \frac{\partial L_f}{\partial \mathbf{x}}(\mathbf{x}(\mathbf{p}_0, T)) \cdot \dot{\mathbf{x}}(\mathbf{p}_0, T) + L(\mathbf{x}(\mathbf{p}_0, T), \mathbf{u}(\mathbf{p}_0, T)) \quad (4.25)$$

のように計算できる. ただし,

$$\mathbf{L}_P = \frac{\partial L}{\partial \mathbf{x}} \cdot \mathbf{X}_P + \frac{\partial L}{\partial \mathbf{u}} \cdot \mathbf{U}_P \quad (4.26)$$

である. このように求まる \mathbf{J}_P , \mathbf{J}_T を用いて, \mathbf{p}_0 , あるいは, \mathbf{p}_0 , T の値に対して, 非線形計画法により J の直接最適化を行うのが提案手法である.

4.2.4 Shooting Method による 2 点境界値問題の求解

最適制御問題は, 4.2.2 節で示したように最小原理によって, 正準方程式と呼ばれる常微分方程式 [式 (4.6), (4.7)] の 2 点境界値問題を解くことに帰着される. 常微分方程式の 2 点境界値問題の一般的な数値解法として射撃法 (Shooting Method) [41] が知られている. Shooting Method では, 最初にある一方の境界点において全ての従属変数の値を選ぶ. その値は, その境界点での境界条件と矛盾しない限り任意の自由パラメータに依存する. この自由パラメータの値は仮の解を定める値である. 選んだ境界における

初期値を用いて初期値問題の解法で常微分方程式を解き，もう一方の境界点に到達したとき，初期値は仮の解であるので到達点での境界条件との間に誤差が生じる．Shooting Method はその誤差がゼロになるように仮の解を更新して真の解に近付こうというものである．ここで，4.2.2 節で定式化した最適制御問題に対して，Shooting Method を適用して紹介する．

状態変数 $x(t)$ と随伴変数 $p(t)$ を合わせた変数を考えて，

$$\xi(t) = \begin{pmatrix} x(t) \\ p(t) \end{pmatrix} \quad (4.27)$$

としたとき，問題の常微分方程式は，

$$\dot{\xi}(t) = F(\xi(t), t) = \begin{pmatrix} f(x(t), \bar{u}(x(t), p(t))) \\ g(x(t), \bar{u}(x(t), p(t)), p(t)) \end{pmatrix} \quad (4.28)$$

と表され，初期境界条件及び終端境界条件はそれぞれ，

$$B_0(\xi(0)) = x(0) - x_0 = 0 \quad (4.29)$$

$$B_f(\xi(T)) = p(T) - \frac{\partial L_f}{\partial x}(x(T)) = 0 \quad (4.30)$$

と表せる．ただし， ξ の次元を n ， B_0 の次元を n_0 ， B_f の次元を n_f とすると， $n = n_0 + n_f$ である．変数 ξ の時刻 $t = 0$ における値 $\xi(0)$ が時刻 $t = 0$ において境界条件を満たす，つまり式 (4.29) が成り立つように $\xi(0) = \xi_0$ を選ぶが，このときパラメータ $\eta = p_0$ に依存するものとする．すなわち，

$$\xi(0) = \xi_0(\eta) = \begin{pmatrix} x_0 \\ p_0 \end{pmatrix} \quad (4.31)$$

と表せ，初期境界条件〔式 (4.29)〕を満たすとする．通常， η の次元は $n_f = n - n_0$ である．このとき，式 (4.28)，(4.31) により初期値問題として表され， $0 \leq t \leq T$ において，

$$\xi(t) = \xi(\eta, t) = \begin{pmatrix} x(p_0, t) \\ p(p_0, t) \end{pmatrix} \quad (4.32)$$

を数値積分により求めることができ，終端境界条件の誤差ベクトル，

$$E(\eta) = B_f(\xi(\eta, T)) \quad (4.33)$$

を評価することができる。終端境界条件が満たされたときには $E(\eta) = 0$ となり、そのときの η が最適解となる。

文献 [41] は、Shooting Method として、非線形連立方程式 $E(\eta) = 0$ の根 η の求解にニュートン法 (Newton-Raphson 法) を適用することにより、2 点境界値問題を解く方法を示している。いま、解の近傍で $E(\eta + \delta\eta) = 0$ を目指すとして、Taylor 級数の 2 次以上の項を無視して得られる 1 次の近似を 0 とおいた式、

$$E(\eta + \delta\eta) \approx E(\eta) + \frac{\partial E}{\partial \eta} \cdot \delta\eta = 0 \quad (4.34)$$

から次の連立 1 次方程式を得る。

$$\frac{\partial E}{\partial \eta} \cdot \delta\eta = -E \quad (4.35)$$

この式により $\delta\eta$ を求め、 k 回目の反復における解 η^k の更新を、

$$\eta^{k+1} = \eta^k + \delta\eta \quad (4.36)$$

により行うことで解を改善し、求解を行うのが Newton-Raphson 法である。文献 [41] によれば、評価関数の偏微分係数が解析的に求められることはまずないので、むしろ、

$$\frac{\partial E}{\partial \eta} = \frac{E(\eta + \Delta\eta) - E(\eta)}{\Delta\eta} \quad (4.37)$$

のように、差分により求めるということが示されている。

それに対して、本研究では Shooting Method の発展として、評価関数を境界条件の誤差ベクトルの内積により定義し、その偏微分係数を微分方程式の数値積分により求め、評価関数の最適化を準ニュートン法 [43] を用いて行うことにより境界条件を満たす解を求めることを提案する。

4.2.5 境界条件誤差評価関数の最適化による解法 (2 つ目の提案手法)

終端境界条件〔式 (4.12)〕の誤差ベクトル $B_f = B_f(\mathbf{p}(\mathbf{p}_0, T), \mathbf{x}(\mathbf{p}_0, T))$ は、

$$B_f(\mathbf{p}(\mathbf{p}_0, T), \mathbf{x}(\mathbf{p}_0, T)) = \mathbf{p}(\mathbf{p}_0, T) - \frac{\partial L_f}{\partial \mathbf{x}}(\mathbf{x}(\mathbf{p}_0, T)) \quad (4.38)$$

であり，それにより終端境界条件誤差評価関数 $V = V(\mathbf{p}_0, T)$ を，

$$V(\mathbf{p}_0, T) = \frac{1}{2} \left(\mathbf{B}_f(\mathbf{p}(\mathbf{p}_0, T), \mathbf{x}(\mathbf{p}_0, T)) \right)^T \cdot \mathbf{B}_f(\mathbf{p}(\mathbf{p}_0, T), \mathbf{x}(\mathbf{p}_0, T)) \quad (4.39)$$

のように定義する．ここで定めた評価関数 V は境界条件が満たされたときにはゼロとなる．終端境界条件誤差評価関数 $V(\mathbf{p}_0, T)$ を最適化する最適化問題，

$$\min_{\mathbf{p}_0} V(\mathbf{p}_0, T) \quad (4.40)$$

の解 \mathbf{p}_0^* ，あるいは終端時間自由の場合は，

$$\min_{\mathbf{p}_0, T} V(\mathbf{p}_0, T) \quad (4.41)$$

の解 \mathbf{p}_0^*, T^* により，最適制御入力 $\mathbf{u}^*(t)$ は評価関数 J を直接最適化した場合と同様に，時間区間 $[0, T]$ あるいは $[0, T^*]$ において，

$$\mathbf{u}^*(t) = \bar{\mathbf{u}}(\mathbf{x}(\mathbf{p}_0^*, t), \mathbf{p}(\mathbf{p}_0^*, t)) \quad (4.42)$$

$$\mathbf{x}(\mathbf{p}_0^*, t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\mathbf{p}_0^*, \tau), \mathbf{u}^*(\tau)) d\tau \quad (4.43)$$

$$\mathbf{p}(\mathbf{p}_0^*, t) = \mathbf{p}_0^* + \int_0^t \mathbf{g}(\mathbf{x}(\mathbf{p}_0^*, \tau), \mathbf{u}^*(\tau), \mathbf{p}(\mathbf{p}_0^*, \tau)) d\tau \quad (4.44)$$

と表される．このとき，実際に V を最適化するために偏微分係数 $V_P = \frac{\partial V}{\partial \mathbf{p}_0}$ ， $V_T = \frac{\partial V}{\partial T}$ を考えると，

$$\mathbf{V}_P = \left(\mathbf{p} - \frac{\partial L_f}{\partial \mathbf{x}} \right)^T \cdot \left(\mathbf{P}_P - \frac{\partial^2 L_f}{\partial \mathbf{x}^2} \cdot \mathbf{X}_P \right) \quad (4.45)$$

$$V_T = \left(\mathbf{p} - \frac{\partial L_f}{\partial \mathbf{x}} \right)^T \cdot \left(\dot{\mathbf{p}} - \frac{\partial^2 L_f}{\partial \mathbf{x}^2} \cdot \dot{\mathbf{x}} \right) \quad (4.46)$$

である．

準ニュートン法は，最急降下法の大域的収束性とニュートン法の局所的に速い収束性というそれぞれの長所を併せ持つような解法であり，ニュートン法がもつヘッセ行列の正定性に起因する大域的収束性の欠点を補う適切な解法である．文献 [41] では多次元非線形方程式における Newton-Raphson 法の適用に関して，評価関数最適化問題として定式化することについて局所解に陥ることを述べて否定している．Newton-Raphson 法

は根の近傍で速い速度で収束する優れた解法であるが、初期値が解の近傍にない場合には局所的な極小解に陥ってうまく収束しない。最適化問題として定義した場合、非線形計画法として一般に知られている解法の適用が全般的に可能であり、より柔軟に対応できる。この場合、文献 [43] にあるように準ニュートン法と緩和された直線探索を組み合わせ用いるべきである。

4.3 提案手法と従来手法の比較

本節では、1次元の最適制御問題である Newton の最小ドラッグノーズシェイプ問題を対象として、提案手法と従来手法を用いて求解を行い、解の精度と計算時間について比較し、提案手法の有効性について検証を行う。

4.3.1 Newton's Min Drag Nose Shape Problem

Newton の最小ドラッグノーズシェイプ問題は、図 4.1 に示されるように、流体中の軸対象なノーズ形状について、ドラッグ係数が最小となるノーズ形状を求める問題である。

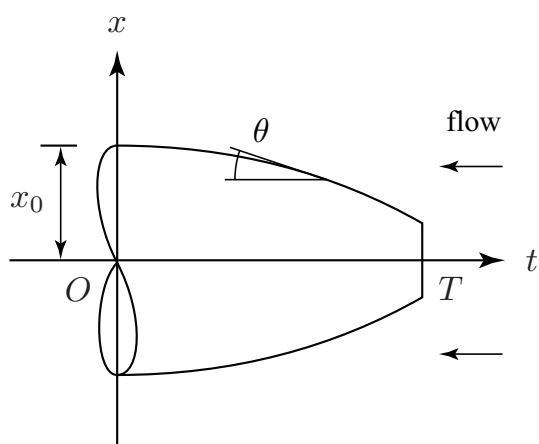


図 4.1 Newton's Min Drag Nose Shape Problem

状態変数 x は位置 t における半径であり，入力 $u = \tan \theta$ として状態方程式は，

$$\frac{dx}{dt} = -u \quad (4.47)$$

である．評価関数 J はドラッグ係数 C_D であり，

$$J = C_D = 2(x(T))^2 + \int_0^T \frac{4xu^3}{1+u^2} dt \quad (4.48)$$

と表わされる．終端を評価する項は，先端の平らな部分 (blunt tip) を評価する項である．本問題では t は時間軸ではなく，形状の回転対称軸方向の位置に取っていることに注意されたい．

4.3.2 比較結果と考察

本節では，提案手法と従来手法合わせて 5 つの手法について，解の精度と計算時間の比較を行った．随伴変数初期値 (Initial value of Adjoint variable) によってパラメトライズして非線形計画法により解く提案手法 (IA method) について，4.2.3 節で述べた J に基づく最適化手法と 4.2.5 節で述べた V に基づく最適化手法の 2 つの提案手法と，従来手法として，Shooting Method， $\frac{\partial H}{\partial \mathbf{u}}(\mathbf{u}) = \mathbf{0}$ を Newton-Raphson 法により解く NR algorithm[33]，常微分方程式の 2 点境界値問題を解く MATLAB ファンクション 'bvp4c'[44] の 3 つを合わせた，合計 5 つの手法について Newton の最小ドラッグノーズシェイプ問題を解くプログラムを MATLAB 上で作成し，それらの手法を用いた求解について比較を行った．'bvp4c' は 3 段 Lobatto III a 公式の選点法 (collocation method) により 4 次の精度の差分を実現し，それにより有限個の t における変数 $\{x(t), u(t), p(t)\}$ について支配方程式と境界条件の残差に対する勾配 (Jacobian) を計算して，ニュートン法 (Newton-Raphson 法) と直線探索を行うように実装されたコードとなっている．

初期条件として $x_0 = 1$ を与え，終端位置は $T = 4$ とした．NR algorithm, Shooting method, IA method においては， t の刻み幅を 4 種類変えて求解を行った．また，'bvp4c' においては， t に対する分割は自動で増減が行われるため，初期解の t の刻み幅

表 4.1 計算時間 t_{calc} [sec]

number of nodes	25	50	100	200
NR algorithm	0.759	2.338	7.535	29.614
'bvp4c'	0.436	0.620	1.211	1.288
Shooting method	0.442	1.614	1.464	2.035
IA method (J)	0.183	0.367	0.715	1.411
IA method (V)	0.190	0.371	0.705	1.404

表 4.2 解の相対誤差 $|\Delta J/\bar{J}|$ [%]

number of nodes	25	50	100	200
NR algorithm	0.53603	0.08005	0.00957	0.00080
'bvp4c'	0.37136	0.05203	0.00494	0.00026
Shooting method	0.05718	0.00871	0.00017	0.00017
IA method (J)	0.02280	0.00360	0.00011	0.00017
IA method (V)	0.05413	0.00952	0.00002	0.00017

を 4 種類変えて求解を行った。4 種類の刻み幅として、 $t = 0 \sim 4$ を 25, 50, 100, 200 分割して与えた。

表 4.1 及び表 4.2 が、各手法を用いたときの計算時間及び解の誤差である。計算時間は 3 回の試行の平均値である。解の誤差は、解の真値を \bar{J} として、

$$\left| \frac{\Delta J}{\bar{J}} \right| = \left| \frac{J - \bar{J}}{\bar{J}} \right| \quad (4.49)$$

により求まる相対誤差をパーセント表示している。真値は十分に細かい刻み幅を用いて求めた。図 4.2 は、各手法について計算時間及び解の誤差を両対数グラフにプロットしたものである。図を見ると、各手法とも計算時間と解の精度におよそトレードオフの関係があることが分かる。IA method では 200 分割を用いた場合に、100 分割より誤差が大きくなっており、適切な分割数の範囲が存在することが分かる。これは、分割数を多

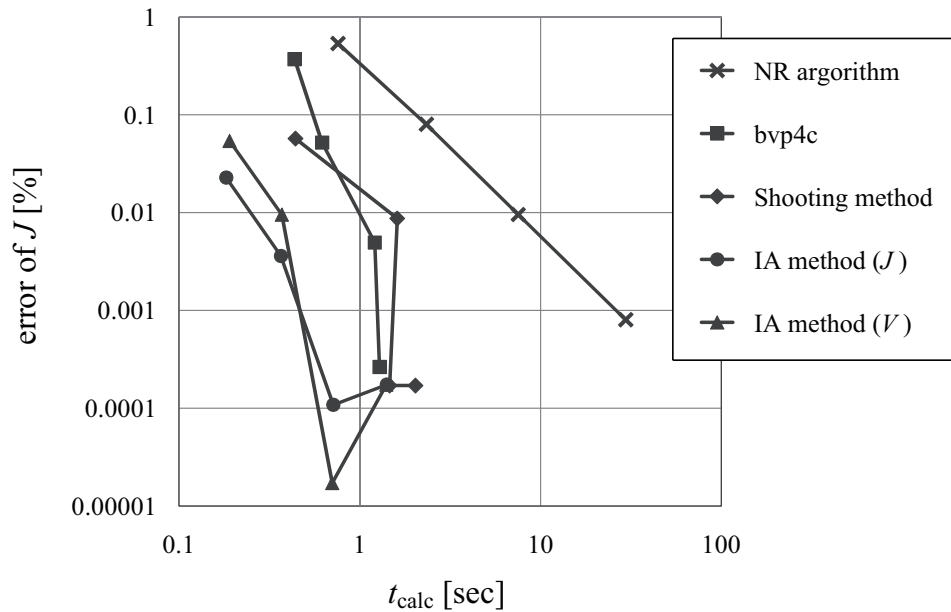


図 4.2 数値解法の比較

くし過ぎると丸め誤差の影響で解の誤差が大きくなるためと考えられる．200 分割の刻み幅を用いたときには，IA method，Shooting method，‘bvp4c’ のいずれを用いても解の精度及び計算時間ともに大きな差はないが，解の誤差及び計算時間のトレードオフ関係から，精度の良い解を最も効率的に得ることができるのは，提案手法の IA method (V) で 100 分割を用いた場合であることが分かる．また，50 分割程度の粗い精度の解を得るために計算時間が最も短くて済むのは，提案手法の IA method (J) であることが分かる．

実用上 J の誤差が 0.01% 程度あったとしてもほとんど真の解と差はないと見なせることは多いと考えられ，その意味で提案手法の IA method (J) を用いるのが最も高速に求解できる手法であると言える．また，より精度の高い解が必要な場合には，提案手法の IA method (V) を用いることで，同じ程度の計算時間を掛けたとしても評価関数の意味でより精度の高い解が得られる．

4.4 機械システムへの随伴変数初期値最適化法の適用

4.3 節では 1 次元の最適制御問題である Newton の最小ドラッグノーズシェイプ問題を対象として提案手法と従来手法を比較した．本節では，機械システムとして特にロボット・多リンク機構系などを対象とし，一般化座標を用いた Lagrange の運動方程式で表現できるシステムについて随伴変数初期値最適化法を適用するための式展開を示す．また，1 自由度ロボット・リンク系における提案手法を適用した数値例を示し，提案した 2 つの手法の比較について述べる．

4.4.1 節で，ロボット・多リンク機構系などの機械システムにおいて最適軌道計画を行うために最適制御問題として定式化を行い，4.2 節で提案した随伴変数初期値最適化法を適用するための式展開を行う．4.4.2 節で重力下における 1 自由度ロボット・リンク系を対象に随伴変数初期値最適化法の適用例を示し，4.2.3 節で提案した評価関数を直接最適化する手法と，4.2.5 節で提案した正準方程式に関する境界条件誤差評価関数を最適化する手法の比較を行う．また，4.4.3 節において随伴変数の初期値をある範囲においてメッシュで切って評価関数の値を計算可能なことを示す．そのことは，評価関数の随伴変数初期値空間における形状が分かり，極値や最適解の位置を調べることができ，そのように得られるデータを適切に取り扱えれば，擬似的な全探索を行えることを意味している．

4.4.1 ロボット・多リンク機構系における状態方程式と評価関数

ここでは機械システムの中でも，一般化座標 q ，一般化力 τ として，Lagrange の運動方程式が，

$$\tau = M(q)\ddot{q} + \tau_N(q, \dot{q}) \quad (4.50)$$

のように書けるシステムを対象とする．このとき，システムの状態方程式は次のように得られる．

$$\dot{x} = f(x, u) = Ax + B(x)(u - u_N(x)) \quad (4.51)$$

ただし,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{pmatrix} \quad (4.52)$$

$$\boldsymbol{A} = \begin{pmatrix} 0 & \boldsymbol{I} \\ 0 & 0 \end{pmatrix} \quad (4.53)$$

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{pmatrix} 0 \\ \{\boldsymbol{M}(\boldsymbol{q})\}^{-1} \end{pmatrix} \quad (4.54)$$

$$\boldsymbol{u} = \boldsymbol{\tau} \quad (4.55)$$

$$\boldsymbol{u}_N(\boldsymbol{x}) = \boldsymbol{\tau}_N(\boldsymbol{q}, \dot{\boldsymbol{q}}) \quad (4.56)$$

である.

システムの評価関数 J を, 状態変数 \boldsymbol{x} , 入力変数 \boldsymbol{u} により,

$$J = L_f(\boldsymbol{x}(T)) + \int_0^T L(\boldsymbol{x}(t), \boldsymbol{u}(t)) dt \quad (4.57)$$

$$L_f(\boldsymbol{x}) = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_d)^T \boldsymbol{Q}_f (\boldsymbol{x} - \boldsymbol{x}_d) \quad (4.58)$$

$$L(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{x}_d)^T \boldsymbol{Q} (\boldsymbol{x} - \boldsymbol{x}_d) + \frac{1}{2} \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u} \\ + \frac{1}{2} (\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}))^T \boldsymbol{Q}_v \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \quad (4.59)$$

のように定義する. ここで, 式 (4.59) における \boldsymbol{Q}_v を含む項は, 状態変数の時間微分 $\dot{\boldsymbol{x}}$ を評価するための項である. このとき, ハミルトン関数 $H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p})$ は,

$$H(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{p}) = \boldsymbol{p}^T \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + L(\boldsymbol{x}, \boldsymbol{u}) \quad (4.60)$$

$$= \frac{1}{2} \boldsymbol{u}^T \boldsymbol{R}' \boldsymbol{u} + \boldsymbol{p}'^T \boldsymbol{B} \boldsymbol{u} + C(\boldsymbol{x}, \boldsymbol{p}) \quad (4.61)$$

のように \boldsymbol{u} の 2 次形式となり, ハミルトン関数最小化問題は許容制御入力 $\boldsymbol{u} \in U$ の制約の下での 2 次計画問題である. ただし, \boldsymbol{R}' 及び \boldsymbol{p}' は,

$$\boldsymbol{R}' = \boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{Q}_v \boldsymbol{B} \quad (4.62)$$

$$\boldsymbol{p}' = \boldsymbol{p} + \boldsymbol{Q}_v (\boldsymbol{A} \boldsymbol{x} - \boldsymbol{B} \boldsymbol{u}_N) \quad (4.63)$$

であり, $C(\boldsymbol{x}, \boldsymbol{p})$ は \boldsymbol{u} を含まない項である.

4.4.2 1自由度ロボット・リンク系における数値例

本節では、1自由度ロボット・リンク系を対象として、4.2節で提案した随伴変数初期値最適化法によって最適軌道計画を行った数値例を示す。また、 Q_v あり・なしの場合を比べて、 \dot{x} を評価する効果についても述べる。

本節で対象とした1自由度ロボット・リンク系は図4.3のようであり、そのパラメータは表4.3とし、評価関数のパラメータは表4.4とした。 Q_v ありの場合には Q_v の値は表の値とし、 Q_v なしの場合には Q_v の要素は全てゼロとして評価関数の計算を行った。

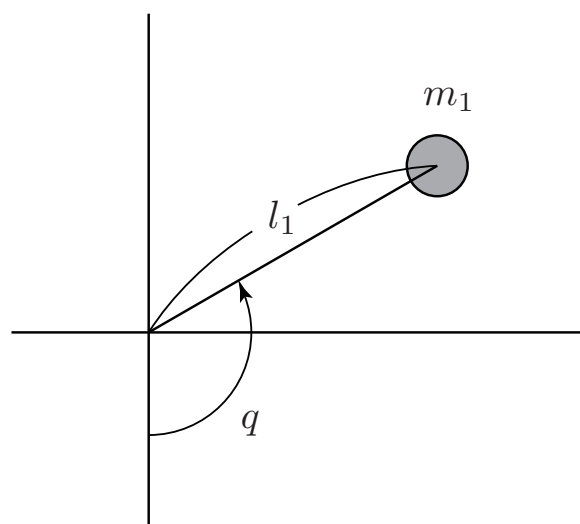


図 4.3 1自由度ロボット・リンク系

表 4.3 1自由度ロボット・リンク系のパラメータ

m_1	0.2	[kg]
l_1	0.3	[m]

表 4.4 評価関数のパラメータ

Q	$\begin{pmatrix} 1 & 0 \\ 0 & 20 \end{pmatrix}$
Q_v	$\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
R	8000
Q_f	$\begin{pmatrix} 4 \times 10^5 & 0 \\ 0 & 8 \times 10^4 \end{pmatrix}$

初期状態と目標状態を，

$$\mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{x}_d = \begin{pmatrix} \pi \\ 0 \end{pmatrix} \quad (4.64)$$

として，提案した手法により終端時刻 $T = 1$ [sec] までの軌道最適化を行った．これは，1 [sec] の間に最下点静止状態から最上点静止状態まで重りを振り上げる動作について，その軌道の最適化を行うものである．いま，制御入力に制約がないとすると，ハミルトン関数を最小化する 2 次計画問題の解より得られる最適制御入力の候補は，

$$\mathbf{u} = -\mathbf{R}'^{-1} \mathbf{B}^T \mathbf{p}' \quad (4.65)$$

となる．

ある p_0 に対して，RKG 法を用いた数値積分により J 及び J_P ，あるいは V 及び V_P を求め，準ニュートン法 [43] を用いて解の更新を行って最適解を探索した．RKG 法による数値積分の時間ステップ幅は 0.01[s] を用いた．ヘッセ行列の近似行列の更新には BFGS 公式を用い，近似行列により探索方向を決定した後，非単調 Armijo 基準を用いた直線探索により解の更新ステップ幅を決定した．用いた計算機環境は，Intel Core 2 Duo CPU T9300 (2.5GHz)，メモリ 4GB を搭載しており，OS は Windows Vista で

表 4.5 Q_v ありの最適化結果

	by J	by V
p_0	$\begin{pmatrix} 360.761 \\ -53.431 \end{pmatrix}$	$\begin{pmatrix} 360.761 \\ -53.431 \end{pmatrix}$
J	766.664	766.664
V	5.7×10^{-10}	1.1×10^{-19}
n_s	77	96
n_e	137	180
t_{calc} [sec]	10.9	14.5

表 4.6 Q_v なしの最適化結果

	by J	by V
p_0	$\begin{pmatrix} 409.587 \\ -36.577 \end{pmatrix}$	$\begin{pmatrix} 409.587 \\ -36.577 \end{pmatrix}$
J	685.056	685.056
V	8.1×10^{-10}	1.9×10^{-17}
n_s	168	285
n_e	269	585
t_{calc} [sec]	21.7	48.9

ある．また，プログラムは Visual C++ 2008 Express Edition を用いてコンパイルを行った．

Q_v ありの場合について最適化を行った結果が表 4.5 であり， Q_v なしの場合について最適化を行った結果が表 4.6 である．それぞれ，評価関数 J の直接最適化による手法と，境界条件により構成した評価関数 V の最適化による手法を用いて最適化を行った．

n_s は解の更新回数であり, n_e は評価関数の評価回数である. また, t_{calc} は最適化に要した計算時間である. 図 4.4 が Q_v ありの場合に最適化により得られた最適軌道と最適入力であり, 図 4.5 が Q_v なしの場合に最適化により得られた最適軌道と最適入力である. (a) が角度, (b) が角速度, (c) がトルク入力を表わしている. それぞれ, J による最適化の結果と V による最適化の結果をプロットしているが, ほぼ同一の最適軌道及び最適入力を得られており, グラフ上では両者の結果は完全に重なっている.

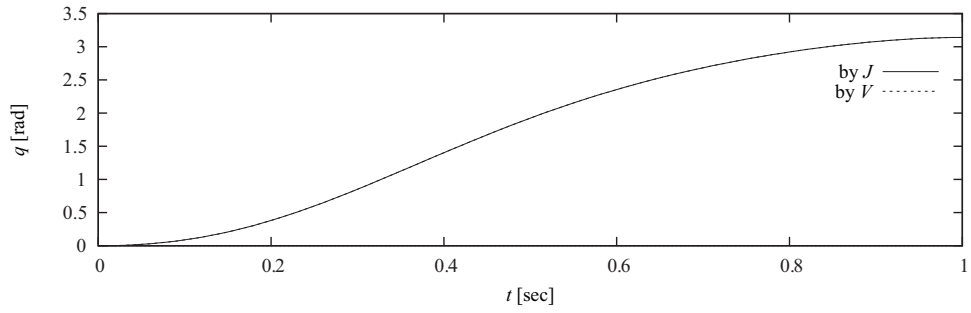
表 4.5 及び表 4.6, あるいは図 4.7 を見ると, V の値が十分に小さくゼロに近くまで収束していて, 境界条件を高い精度で満たしており, 図 4.6, あるいは表 4.5 と表 4.6 を見てわかるように Q_v ありの場合, Q_v なしの場合それぞれについて, J 及び V による最適化のどちらも同じ J の値及び同じ p_0 に到達しており, どちらの最適化によっても同じ最適解に到達していると結論付けられる. J による最適化と V による最適化がどちらも正しい最適解が得られているので, 図 4.4 あるいは図 4.5 において, J の最適化による最適軌道・最適入力と V の最適化による最適軌道・最適入力は一致している.

最適化の過程を見ると, V の値〔図 4.7〕は最適解の近傍において 20~25 桁ぐらい急激に減少しており, 解から離れた領域においては V の傾斜が緩やかで, それにより V による最適化は解から離れた領域で収束速度が遅いことと合致している. このケースでは J による最適化の方が収束が速かった.

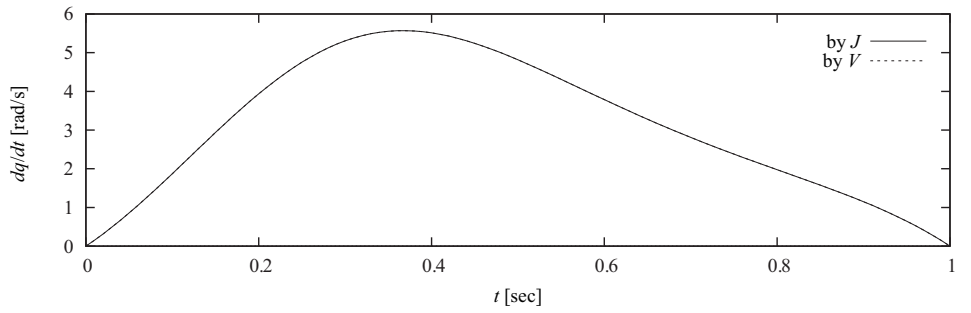
4.3 節で示したノーズシェイプ問題における比較の結果と合わせて考えると, 同程度の精度の解を求めるために J による最適化の方が計算時間は短くて済む傾向があるが, 解の精度や収束性については V による最適化の方が良い場合があるので, 高精度な解を安定して求めたい場合などには V による最適化を用いた方が良い場合もあるといえることができる.

図 4.4 と図 4.5 において, Q_v ありの場合と Q_v なしの場合の軌道を比べると, ありの場合の方が速度の変化が全体的になだらかになっており, Q_v の効果により \dot{q} が抑えられていることがわかる. また, Q_v ありの場合と Q_v なしの場合を比べると, 表 4.5 や表 4.6, あるいは図 4.6 や図 4.7 を見ると分かるように, Q_v ありの場合の方が解の収束が早く, 計算時間が短くて済んでいる. 図 4.4 と図 4.5 を比べて分かるように, Q_v ありの

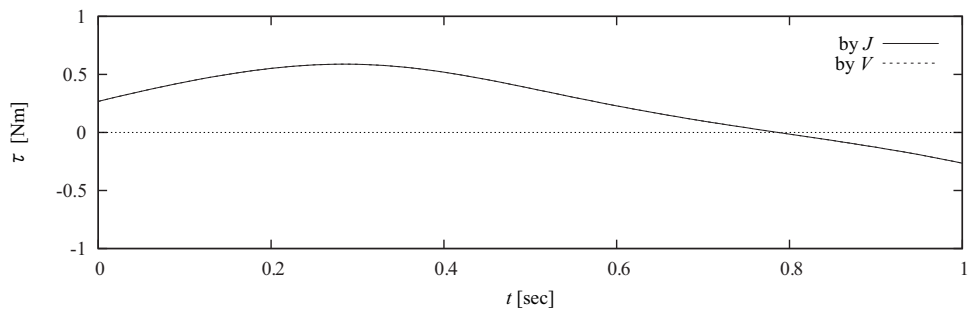
場合となしの場合で実際には最適軌道に大きな差がないとも言えるので、同じような軌道を得られて解の収束が早い Q_v ありを用いる方が計算時間を短縮できて良いと言える。



(a) angle

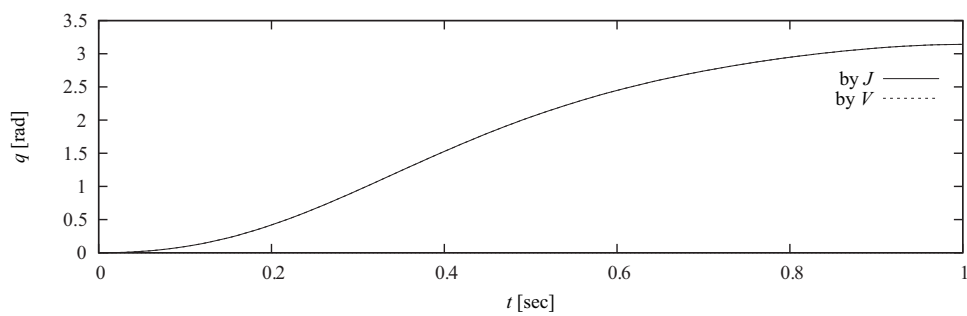


(b) angular velocity

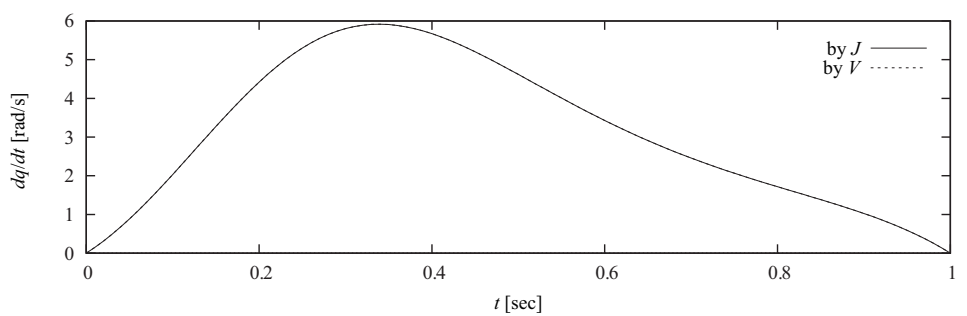


(c) torque input

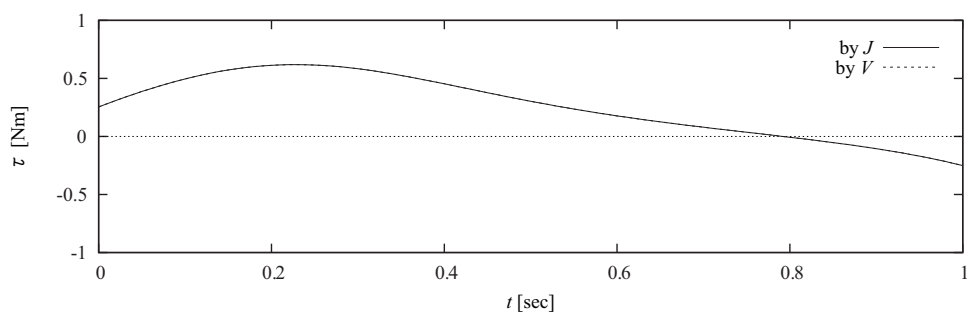
☒ 4.4 Optimization result of the case with Q_v



(a) angle



(b) angular velocity



(c) torque input

☒ 4.5 Optimization result of the case without Q_v

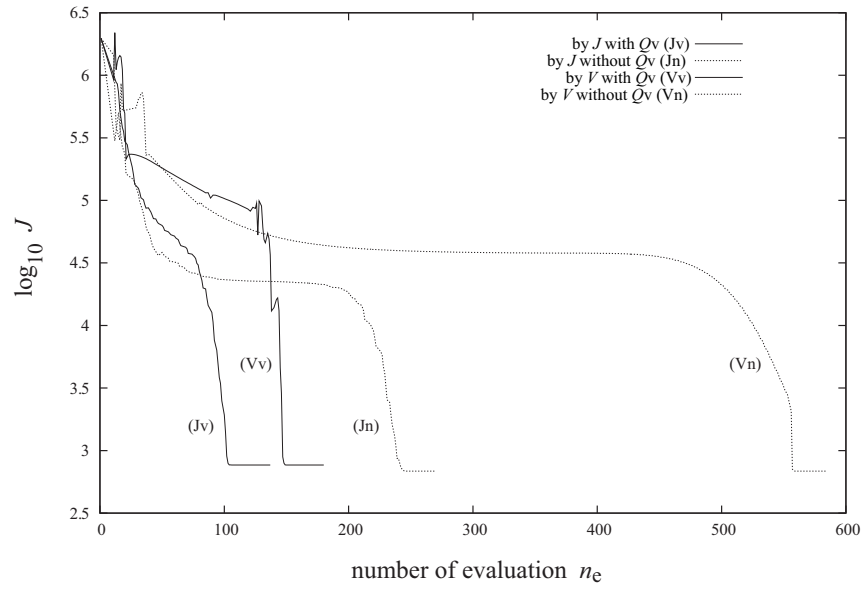


図 4.6 最適化過程における評価関数 J の値

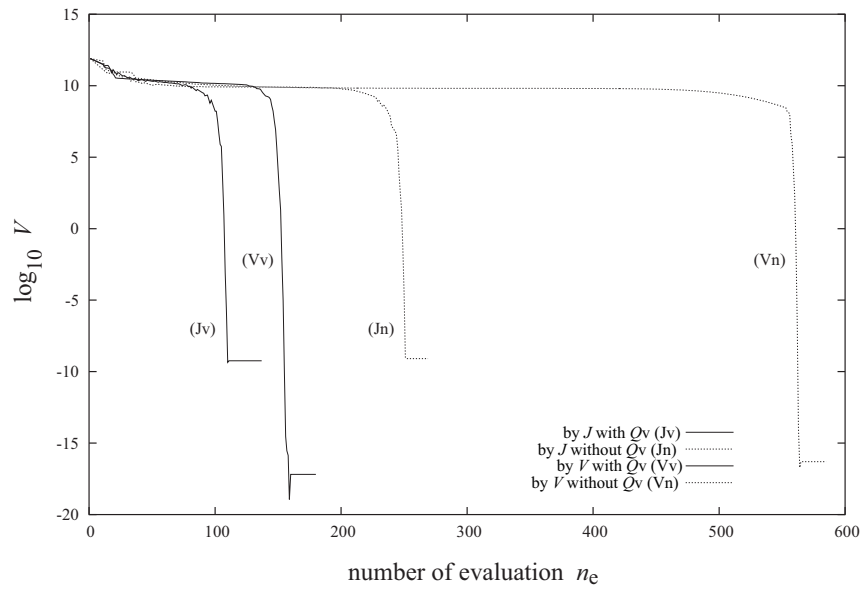


図 4.7 最適化過程における評価関数 V の値

4.4.3 擬似的な全探索に関する考察

本節では、前節と同じ系を対象に R 及び T を変化させて提案手法により最適化を試みた。 R については (8×10^3) と (1×10^4) の 2 種類、 T については 0.8, 1.0, 1.2, 1.4 [sec] の 4 種類を対象とした。初期解を $p_0 = (0, 0)^T$ としたときに、準ニュートン法を用いた勾配法による最適化の成否は、それぞれの R 及び T の組み合わせに対して表 4.7 に示したようになった。 J の最適化あるいは V の最適化のどちらの手法についても同じ結果であった。表 4.7 のように、あるパラメータの組合せに対しては、解法として用いる非線形計画法の性質によって、最適解に到達する前にアルゴリズムが局所的なスポットに陥って求解に失敗するケースがある。

図 4.8 は、表 4.7 の*2 に相当するパラメータを用いた場合に、 p_0 の値を格子状に与えて J の値をプロットしたものである。 J の積分ステップとして $h = 0.1$ [s] を用い、 p_0 の第 1 要素および第 2 要素をそれぞれ [-600, 600] の範囲で 100 等分し、合計 10,000 点において計算を行った。前節と同様の計算機環境を用いて、 $T = 1$ [sec] のときに 10,000 点の計算にかかった時間はおよそ 90 [sec] である。このような粗い計算によって解の全貌が明らかになり、それにより大域的な最適解の存在範囲を十分に狭めて決定できれば、その近傍における値を初期値として勾配法により正しい解を精度よく求めることができるようになる。例えば、*2 のケースでは $p_0 = (0, 0)^T$ から求解を行った場合に、求解に失敗するが、 p_0 の第 1 要素および第 2 要素をそれぞれ [-600, 600] の範囲で 100 等分した 10,000 点から、 J の値が小さい格子点を選んで求解を行うと、3 番目に小さい格子点である $p_0 = (400, -20)^T$ からの求解が成功する。その最適解は $p_0^* = (393.159, -28.044)^T$ であった。

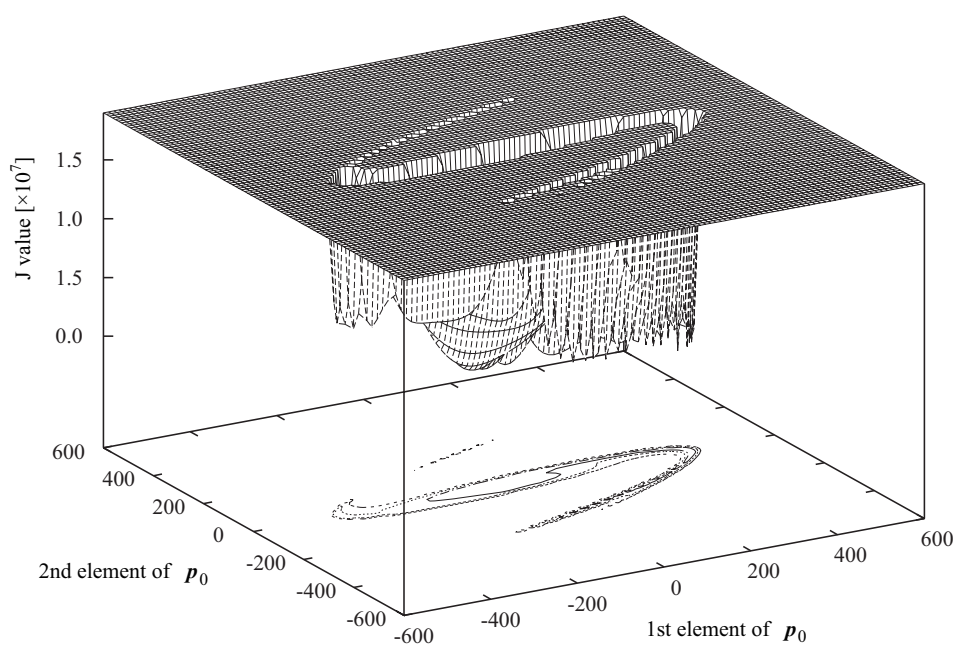
図 4.9, 図 4.10, 図 4.11 はそれぞれ、*1, *2, *3 のケースにおいて、 p_0 空間における J の等高線をプロットしたグラフである。グラフ中の × 印は最適解を表わしている。*1, *3 は、 $p_0 = (0, 0)^T$ から求解を開始しても、等高線が狭い谷をそれほど進まずに最適解に至ることができるが、*2 は狭い谷を進んで $p_0 = (400, 0)^T$ 付近で大きく曲がった先に最適解がある。*2 で $p_0 = (0, 0)^T$ から求解を開始した場合、途中で極値と

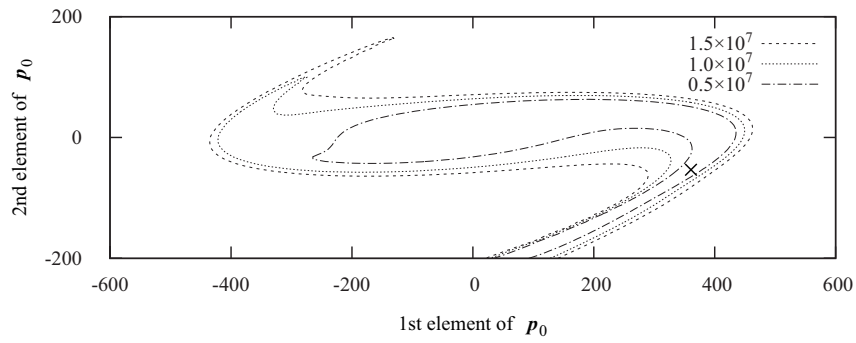
なっているか、あるいは準ニュートン法では2階偏微分係数であるヘッセ行列 (Hessian matrix) が正定となるように近似を行うが、部分的に下に凸でない領域があってヘッセ行列の近似が悪くなっているかしており、最適解に向かう勾配を計算できなくなっていると思われる。このように、 p_0 空間において J を分析することで、解の性質を考慮して解法の考察に役立てることができる。

SCGRA[37][38]、MQA[39][40] に代表されるような、関数空間における非線形計画法など、あるいは問題を離散化して解く従来の手法では、解を表わすパラメータの次元が大きくなり過ぎるため、格子点のようなものを構成して全探索に近いことを行うことは考えられず、擬似的な全探索を行うことはできない。それに対して、提案手法では格子点を構成するなどして擬似的な全探索を行うことができる。ただ、より高次元の問題を考えた場合、次元の大きさに対して指数関数的に計算量が増え、十分に格子を細かく取れなくなるので、その場合には、例えば、ランダムに探索点を決めるような擬似的な全探索手法を用いることで、広い空間を効率的に探索できるようになる。

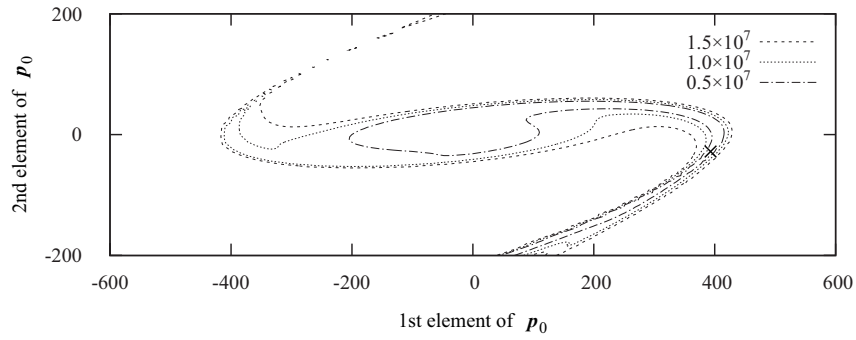
表 4.7 最適化の成否

		T [s]			
		0.8	1.0	1.2	1.4
$R = \left(1 \times 10^4 \right)$	with Q_v			×	
	without Q_v		×		
$R = \left(8 \times 10^3 \right)$	with Q_v		*1	×	*3
	without Q_v				

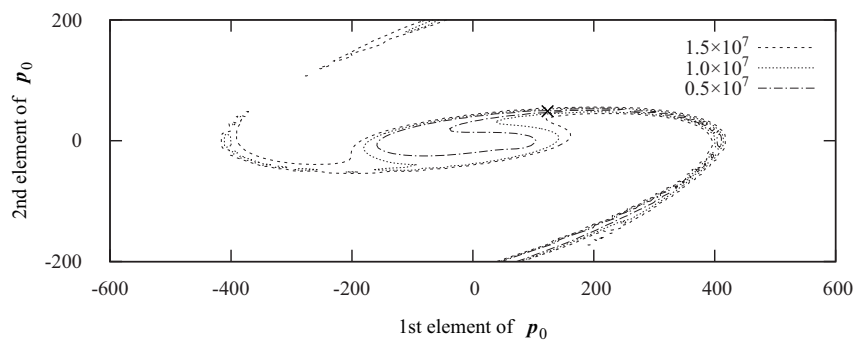
図 4.8 The shape of J in p_0 space in the case of *2



☒ 4.9 The contour plot of J in the case of *1



☒ 4.10 The contour plot of J in the case of *2



☒ 4.11 The contour plot of J in the case of *3

4.5 結言

本研究では、4.2 節において有限時間区間における最適制御問題の一般的な解法を 2 つ提案し、4.4 節において 1 自由度ロボット・リンク系における最適軌道計画について適用例を示し、有効性を検証した。

最適制御理論を応用した機械システムにおける最適軌道計画問題に関する研究や、最適制御問題の一般的な解法について、1.6 節で詳しく述べているが、これまで従来の解法では関数空間における最適化を行っていたので、対象とする系が高次元になると、あるいは計算機内で計算するために必要である時間的離散化に用いる時間ステップの数が大きくなると、それに従い計算量や必要な記憶容量が爆発的に膨大となり、高性能な計算機資源の拘束を長時間必要とした。あるいは時間ステップを粗く取り、精度の悪い解しか得られないなどであった。

提案手法では、最適性の必要条件である最小原理を用いて随伴変数の初期値により解のパラメトライゼーションを行い、問題の評価関数、あるいは最小原理により得られる正準方程式の境界条件誤差評価関数の直接最適化を行うので、解を表すパラメータが解の次元に比べて少数でありながら、解を数値積分的な意味で連続的に扱い、十分滑らかな解を表現でき、それにより、

- 一般の非線形計画法をそのまま適用することができ、非線形計画法における最新の研究成果の恩恵をダイレクトに受けて、最適解への高速な収束が望める。
- 直接最適化を行った場合には計算誤差が少なく、より評価関数の小さい解が得られる。
- 従来の手法では不可能だった擬似的な全探索を行える。

などのメリットが見込める。従来の手法の中でも勾配法に属する逐次更新型アルゴリズムの場合、最適解以外の極値でアルゴリズムが誤停止してしまうという欠点を持っていたが、提案手法では擬似的な全探索を行えるので、ヒューリスティックな方法に比べて正しい最適解に収束する初期値を適切に設定できる可能性を持っている。

提案手法の今後の課題としては、

- 従来の手法との比較
- 擬似的な全探索を行う手法の検討

が挙げられる。

第 5 章

1 次展開によるリアルタイム準最適制御の実現

5.1 緒言

機械システムにおいてリアルタイム準最適制御を行うために，1 次展開最適ゲインと名付けたゲインを導出し，それによりリアルタイム準最適制御を行えることを示す．5.2.1 節において，機械システムにおけるリアルタイム最適制御の可能性について検討し，5.2.2 節において，リアルタイム準最適制御を行うために，1 次展開最適ゲインと名付けたゲインを導出し，また，線形時不変系における定常最適レギュレータでは厳密に最適制御と同値となることを類似例として示す．5.3 節において，4.4.2 節で用いたのと同じ 1 自由度ロボット・リンク系を対象として，1 次展開最適ゲインを求める数値例を示し，1 次展開最適ゲインによるリアルタイム準最適制御の数値シミュレーションを行った例を示す．また，比較例として定常ゲインによる制御を挙げ，白色ガウストルクノイズ及びステップ外乱トルクを加えた場合について，数値シミュレーションによる比較を行い，1 次展開最適ゲインの有効性について検討を行った．

5.2 1次展開によるリアルタイム準最適制御の実現

5.2.1 リアルタイム最適制御について

最適軌道への追従制御を考える前にリアルタイム最適制御の可能性について考える。

厳密なリアルタイム最適制御においては，制御時間区間内の時刻 $t_1 \in [0, T]$ での最適制御入力 $\mathbf{u}^*(t_1)$ は，時間区間 $[t_1, T]$ での評価関数 J_1 を，

$$J_1 = L_f(\mathbf{x}(T)) + \int_{t_1}^T L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (5.1)$$

として，最適化問題，

$$\min_{\mathbf{u}(t; t \in [t_1, T])} J_1 \quad (5.2)$$

の解により定められる。

つまり，随伴変数の時刻 $t = t_1$ における初期値を $\mathbf{p}(t_1) = \mathbf{p}_1$ として，随伴変数初期値最適化の手法により書けば，最適化問題，

$$\min_{\mathbf{p}_1} J_1(\mathbf{p}_1) \quad (5.3)$$

の最適解を \mathbf{p}_1^* として，最適制御入力 $\mathbf{u}^*(t_1)$ は，

$$\mathbf{u}^*(t_1) = \bar{\mathbf{u}}(\mathbf{x}(t_1), \mathbf{p}_1^*) \quad (5.4)$$

と得られる。

具体的な制御問題において制御インターバル Δt で制御入力を更新するときに，時刻 t_1 での解が得られているとして時刻 $t_2 = t_1 + \Delta t$ での求解を考えた場合，時間区間 $[t_2, T]$ での評価関数 J_2 を，

$$J_2 = L_f(\mathbf{x}(T)) + \int_{t_2}^T L(\mathbf{x}(t), \mathbf{u}(t)) dt \quad (5.5)$$

として， J_2 を最小化する最適化問題，

$$\min_{\mathbf{p}_2} J_2(\mathbf{p}_2) \quad (5.6)$$

を解く際に非線形計画法の初期解として時刻 t_1 での解 p_1^* を与えて、それにより時刻 t_2 での最適解 p_2^* を求めることが考えられる。制御時間間隔 $\Delta t = t_2 - t_1$ が短い場合には、 p_1^* と p_2^* は値が近いと考えられ、最適化に要する計算時間は短いと思われる。

しかし、問題が高次元化するとオンライン計算のリアルタイム性を確保するのは困難になると思われ、また、非線形計画法として勾配法を用いた場合には誤って最適値ではない極値で停止することも考えられ、真に最適性を保証することは極めて難しい。また、最終的に辿る軌跡が $t = 0$ において計画される最適軌道とは大きく異なってくる可能性も否定できない。ゆえに、厳密なリアルタイム最適制御を適用することは、対象とするシステム、あるいはシステムに課される制御目的によっては適切でないことがある。

これまでに行ってきた計算機環境 (Core 2 Duo CPU T9300 2.5GHz) による最適化に要する時間を見ても、1次元の問題を対象としていてさえ、精度を大幅に犠牲にし、最適化をある段階で打ち切るようなことをしても、現在普及している計算機の演算性能ではリアルタイム性を確保することは極めて困難と言わざるを得ない。よって、リアルタイム最適制御を適用することは現実的ではない。

5.2.2 1次展開最適ゲインの導出

1次展開最適ゲインとは、ひとことで述べると最適入力最適軌道まわりでの1次の展開により定まるゲインである。本節では1次展開最適ゲインを導出し、線形時不変系における定常最適レギュレータで類似の導出を対照例として示す。

制御時間区間 $t \in [0, T]$ において追従すべき軌道として、理想的な場合における最適軌道 $x_d(t) = x^*(t)$ 、及び最適制御入力 $u_d(t) = u^*(t)$ を考える。このとき随伴変数 $p_d(t) = p^*(t)$ とする。

いま、時刻 $t_1 \in [0, T]$ における状態ベクトル $x(t_1)$ の微小な変化 $x_d(t_1) \rightarrow x_d(t_1) + \Delta x_1$ を考えたときに、 $t \in [t_1, T]$ における評価関数 J_1 を、

$$J_1 = L_f(x(T)) + \int_{t_1}^T L(x(t), u(t)) dt \quad (5.7)$$

として、その微小な変化 ΔJ_1 を最小化する問題を考える。時刻 $t_1 \in [0, T]$ における随

伴変数 $p(t_1)$ の微小な変化 $p_d(t_1) \rightarrow p_d(t_1) + \Delta p_1$ を考えると, ΔJ_1 の最小化問題は随伴変数初期値最適化の手法により,

$$\min_{\Delta p_1} \Delta J_1(\Delta p_1) \quad (5.8)$$

と書ける. この最適化問題の解を Δp_1^* として, それにより $x(t_1)$ の微小な変化 Δx_1 に対して評価関数の微小な変化 ΔJ_1 を最小化する制御入力 $u^{**}(t_1)$ は,

$$u^{**}(\Delta x_1, t_1) = \bar{u}(x_d(t_1) + \Delta x_1, p_d(t_1) + \Delta p_1^*) \quad (5.9)$$

と書ける. ここで, ある Δx_1 に対して Δp_1^* を 1 つ求めることができ,

$$\Delta p_1^* = \Delta p_1^*(\Delta x_1, t_1) \quad (5.10)$$

と書けるので, u^{**} は Δx_1 に依存すると見なすことができ, $x(t_1) = x_d(t_1)$ まわりでのテーラー展開により,

$$u^{**}(\Delta x_1, t_1) = u_d(t_1) + \frac{\partial u^{**}}{\partial \Delta x_1} \cdot \Delta x_1 + (2 \text{ 次以上の項}) \quad (5.11)$$

と書ける. よって, 時変フィードバックゲイン $K^*(t)$ として,

$$K^*(t) = \frac{\partial u^{**}}{\partial \Delta x} \quad (5.12)$$

を定め,

$$u(t) = u_d(t) + K^*(t) \cdot (x(t) - x_d(t)) \quad (5.13)$$

により近似的なリアルタイム最適制御入力を構成することができる. このようなゲイン K^* を 1 次展開最適ゲインと呼ぶことにする.

5.2.3 線形時不変系の定常最適レギュレータ問題における 1 次展開最適ゲイン

システムの状態方程式が,

$$\dot{x} = Ax + Bu \quad (5.14)$$

で表わされ, 時刻 $t = 0$ における初期状態が $x(0) = x_0$ であるとして, 評価関数 J を,

$$J = \int_0^{\infty} \left(\frac{1}{2} x^T Q x + \frac{1}{2} u^T R u \right) dt \quad (5.15)$$

とすると, 定常 Ricatti 方程式

$$A^T P + P A - P B R^{-1} B^T P + Q = O \quad (5.16)$$

の解 P_{∞} を用いて, 定常最適ゲイン K_{∞} を,

$$K_{\infty} = -R^{-1} B^T Q P_{\infty} \quad (5.17)$$

と定めることができる. このとき, 最適軌道は,

$$x_d(t) = \exp\left(t(A + BK_{\infty})\right) \cdot x_0 \quad (5.18)$$

で与えられ, 最適軌道を与える制御入力は,

$$u_d(t) = K_{\infty} x_d(t) \quad (5.19)$$

で与えられる.

ここで, 時刻 $t_1 \in [0, T]$ における状態ベクトル $x(t_1)$ の微小な変化 $x_d(t_1) \rightarrow x_d(t_1) + \Delta x_1$ を考えたときに, $t \in [t_1, T]$ における評価関数 J_1 を,

$$J_1 = \int_{t_1}^{\infty} \left(\frac{1}{2} x^T Q x + \frac{1}{2} u^T R u \right) dt \quad (5.20)$$

とすると, $x(t_1)$ の微小な変化 Δx_1 に対して評価関数の微小な変化 ΔJ_1 を最小化する制御入力 $u^{**}(\Delta x_1, t_1)$ は,

$$u^{**}(\Delta x_1, t_1) = K_{\infty} \left(x_d(t_1) + \Delta x_1 \right) \quad (5.21)$$

となり, 1次展開最適ゲイン K^* は,

$$K^* = \frac{\partial u^{**}}{\partial \Delta x} = K_{\infty} \quad (5.22)$$

と得られ, それにより制御入力を,

$$u(t) = u_d(t) + K^* \left(x(t) - x_d(t) \right) \quad (5.23)$$

と構成できる．これは厳密な最適レギュレータの入力と同じである．すなわち式 (5.19) を考慮すると，最適フィードバック入力 $u^*(t)$ ，

$$u^*(t) = K_{\infty} \cdot x(t) \quad (5.24)$$

に一致する．

つまりこの場合，状態ベクトルの最適軌道からの微小変化に対する評価関数の微小変化を最小化する制御入力は，状態ベクトルの微小変化に対して 1 次の線形項しか持たず，その線形係数（1 次展開最適ゲイン）は定常最適ゲインと同じになる．そして，評価関数の微小変化の最小化は，厳密に評価関数を最適化することと同値である．

5.3 1 次展開最適ゲインの適用と検証

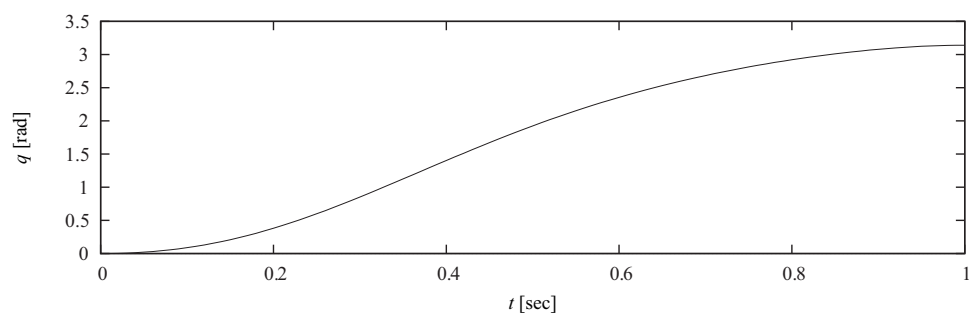
5.3.1 1 次展開最適ゲインの数値例

4.4.2 節において，随伴変数初期値最適化法の適用例として用いた 1 自由度ロボット・リンク系の最適軌道を対象に，その最適軌道への追従においてリアルタイム準最適制御を実現する 1 次展開最適ゲインの計画の数値例を示す．

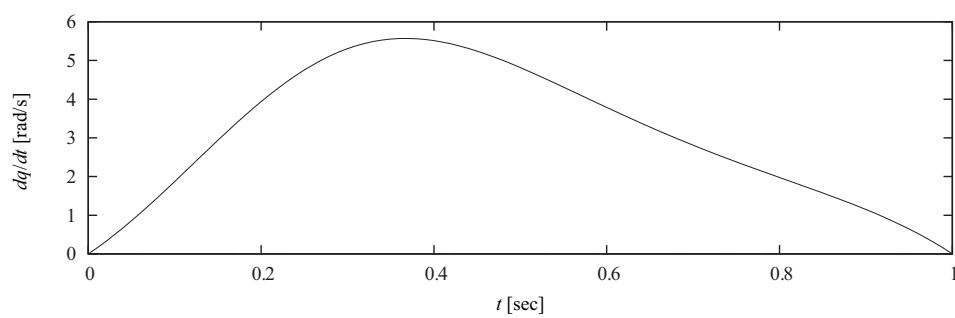
1 自由度ロボット・リンク系は図 4.3 のようであり，1 自由度ロボット・リンク系のパラメータは表 4.3 の値とし，評価関数のパラメータは表 4.4 の値とした．4.2 節で示した随伴変数初期値最適化法により求めた最適軌道及び最適制御入力は図 5.1 のようである．図 5.1 の最適軌道に沿って求めた 1 次展開最適ゲインは図 5.2 のようである．

ここで興味深いのは制御時間区間の始めの方で正のフィードバックゲインとなっている点である．厳密な理由付けには詳細な解析が必要であるが，察するに入力の絶対値を抑えるための項が影響してそうなっているのではないかと思われる．つまり途中の入力に対して制御効率の悪い区間での追従ゲインを小さくし，後半の負荷の小さな区間で制御を効率的に行った方が評価関数を最小化できるということだと推測できる．

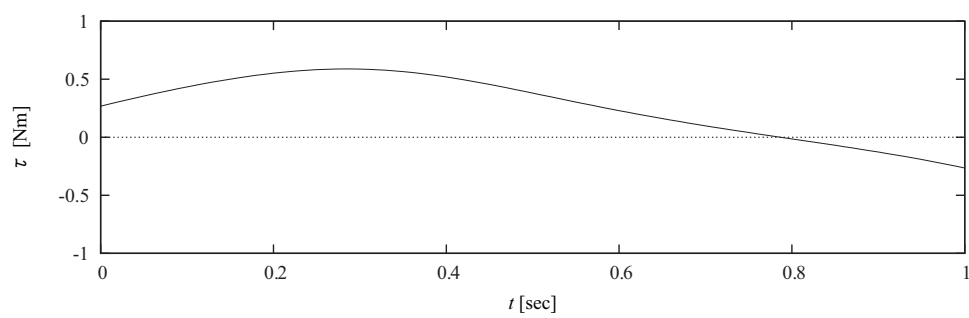
1 次展開最適ゲインを求める際の偏微分は数値微分により行い，その差分を 0.01 と 0.001 の 2 種類試したが得られたゲインは同じであり，2 次以上の項はほとんど利いていないと思われる．



(a) angle

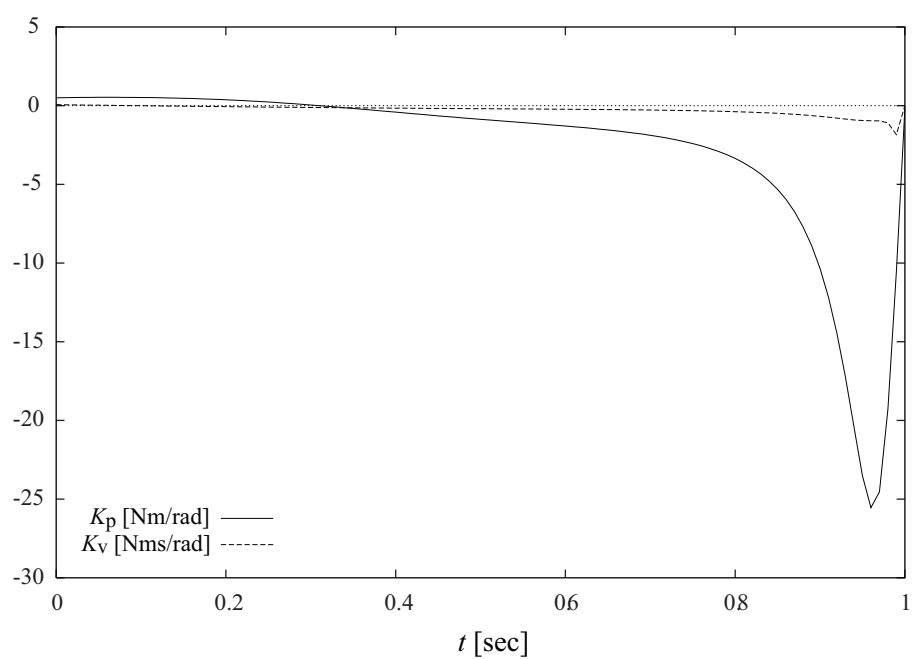


(b) angular velocity



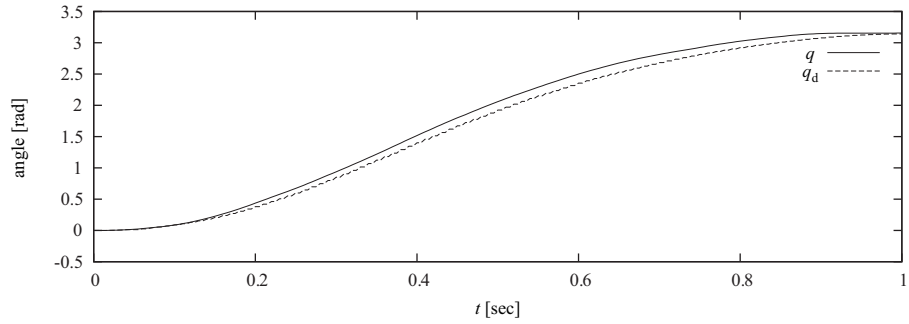
(c) torque input

図 5.1 最適軌道と最適制御入力

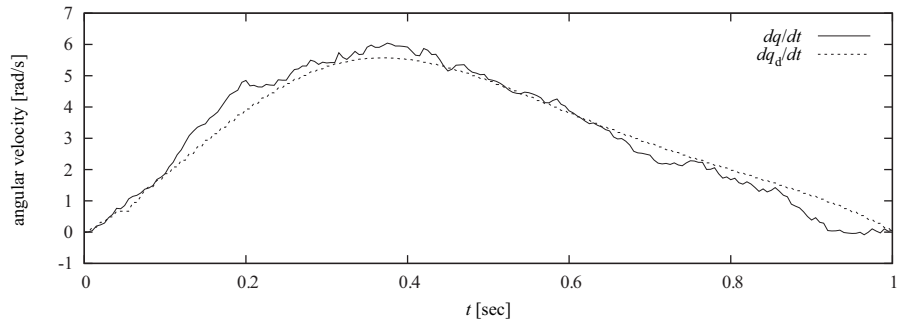
図 5.2 1次展開最適ゲイン K^*

5.3.2 1次展開最適ゲインを用いた制御と定常ゲインを用いた制御の数値例

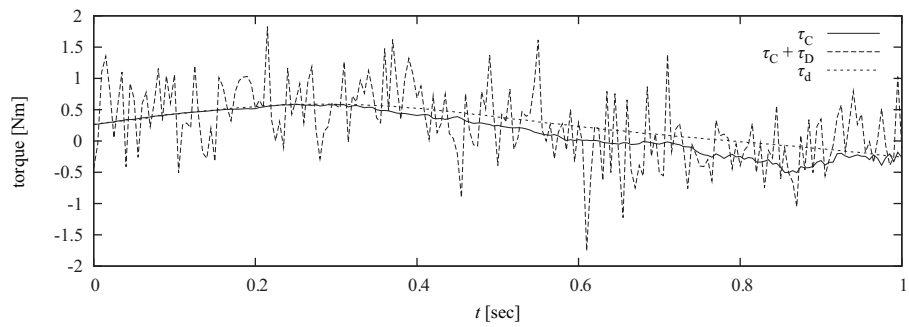
1次展開最適ゲインを用いた制御において、ゲイン及び参照軌道・参照入力は0.01 [sec]ごとに求めた。定常ゲインにおける位置フィードバックゲイン及び、速度フィードバックゲインは、それぞれ $K_p = 10$ 及び、 $K_v = 1$ である。図 5.3 は、1次展開最適ゲインを用いた制御において、標準偏差 $\sigma = 0.5$ [Nm] の白色ガウストルクノイズ及び、 $t = 0.1$ [sec]において +0.1 [Nm] のステップトルクを加えた場合であり、図 5.4 は、1次展開最適ゲインを用いた制御において、標準偏差 $\sigma = 0.5$ [Nm] の白色ガウストルクノイズ及び、 $t = 0.1$ [sec]において -0.1 [Nm] のステップトルクを加えた場合である。また、図 5.5 は、定常ゲインを用いた制御において、標準偏差 $\sigma = 0.5$ [Nm] の白色ガウストルクノイズ及び、 $t = 0.1$ [sec]において +0.1 [Nm] のステップトルクを加えた場合であり、図 5.6 は、定常ゲインを用いた制御において、標準偏差 $\sigma = 0.5$ [Nm] の白色ガウストルクノイズ及び、 $t = 0.1$ [sec]において -0.1 [Nm] のステップトルクを加えた場合である。(a) は角度、(b) は角速度、(c) はトルクのグラフである。



(a) angle

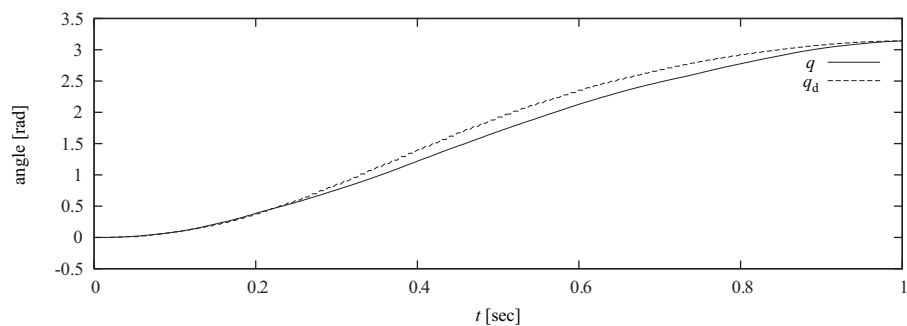


(b) angular velocity

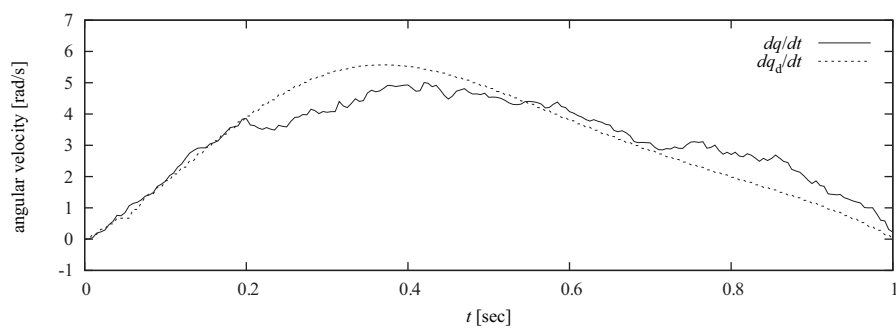


(c) torque

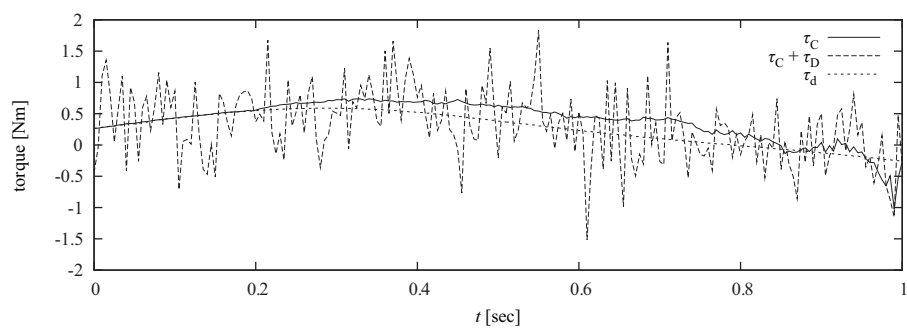
図 5.3 1次展開最適ゲインによる制御（白色ガウストルクノイズと正のステップトルク外乱を加えた場合）



(a) angle

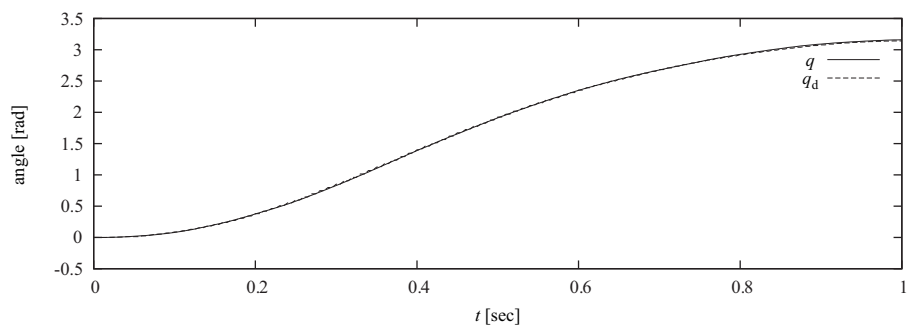


(b) angular velocity

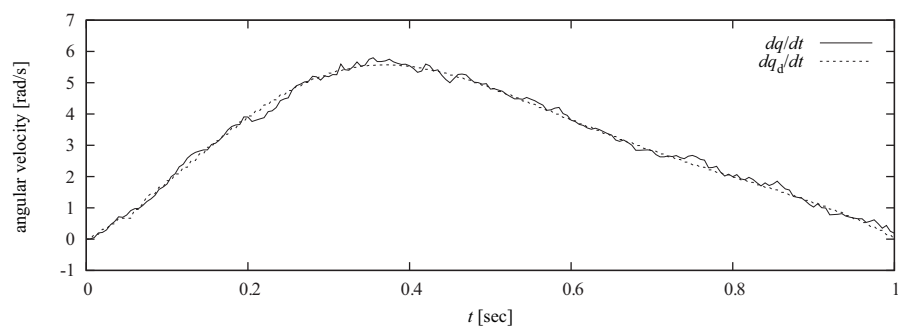


(c) torque

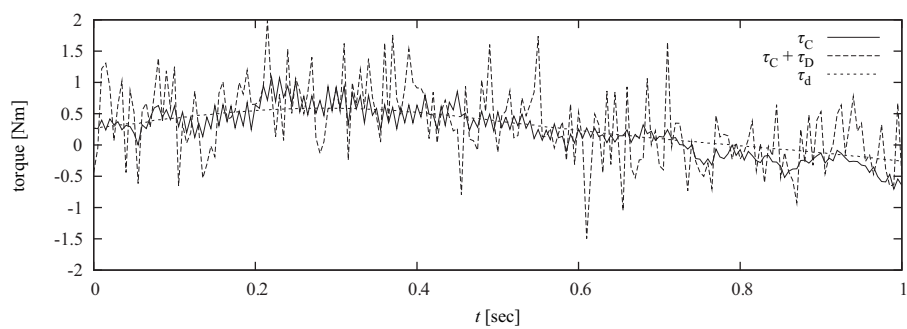
図 5.4 1次展開最適ゲインによる制御（白色ガウストルクノイズと負のステップトルク外乱を加えた場合）



(a) angle

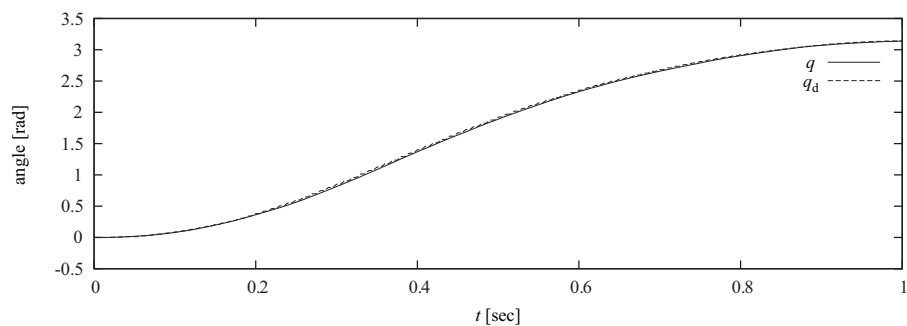


(b) angular velocity

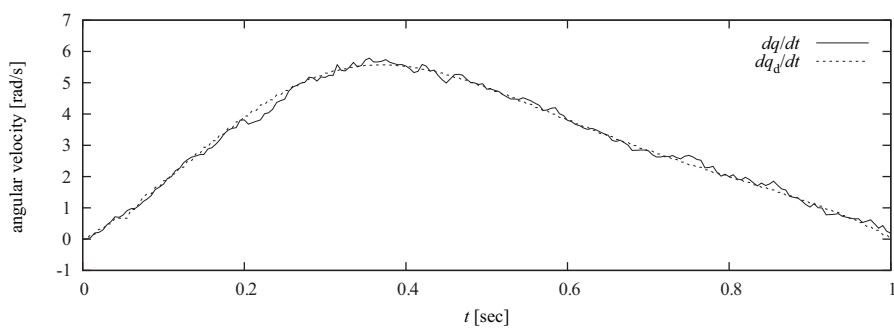


(c) torque

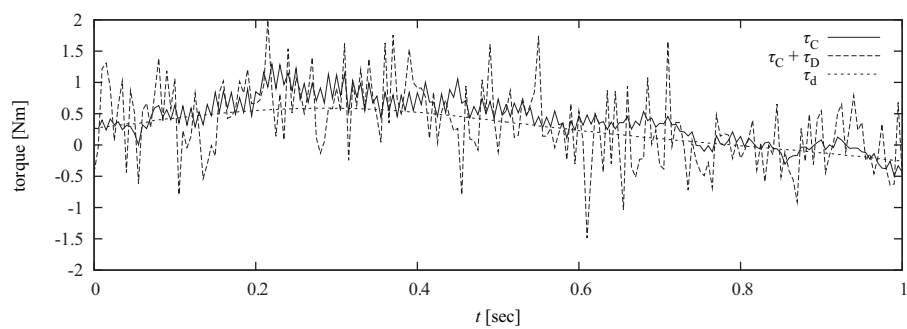
図 5.5 定常ゲインによる制御（白色ガウストルクノイズと正のステップトルク外乱を加えた場合）



(a) angle



(b) angular velocity



(c) torque

図 5.6 定常ゲインによる制御（白色ガウストルクノイズと負のステップトルク外乱を加えた場合）

5.3.3 評価関数の比較

評価関数 J の値を軌道の最適性の評価指標として，1 次展開最適ゲインを用いた制御と定常ゲインを用いた制御について比較を行った．定常ゲインにおける速度フィードバックゲインは $K_v = 1$ で固定とし，位置フィードバックゲイン K_p をいくつか変化させて比較対象とした．図 5.7 はステップトルク外乱を加えた場合における比較である．また，図 5.8 は白色ガウストルクノイズを加えた場合における比較である．図 5.9 は白色ガウストルクノイズ及び正のステップトルク外乱を加えた場合における比較であり，図 5.10 は白色ガウストルクノイズ及び負のステップトルク外乱を加えた場合における比較である．CGC (Constant-Gain-Control) は定常ゲインを用いた制御による結果を示している．FXC (First-order-eXpanded-optimal-gain-Control) は 1 次展開最適ゲインを用いた制御による結果を示している．

ステップトルク外乱を加えた場合に， $K_p = 10$ において定常ゲインによる制御の方が評価関数が小さくなっており， $K_p = 5$ のときも正のステップトルク外乱の場合には定常ゲインの方が評価関数が小さくなっている．白色ガウストルクノイズを加えた場合に，ノイズの標準偏差が小さいときには 1 次展開最適ゲインの方がほぼ評価関数が小さくなっており，また，適切に定常ゲインを選べば 1 次展開最適ゲインと定常ゲインで評価関数の大きさはほとんど変わらない結果となる．ノイズの標準偏差が大きいときには適切に定常ゲインを選べば定常ゲインの方が良い結果となる場合がある．しかし，ノイズの標準偏差によって評価関数を小さくするために適切な定常ゲインの大きさが異なるので，一般的に評価関数を小さくするための適切な定常ゲインというものを決めることはできない．

白色ガウストルクノイズとステップトルク外乱の両方に加えた場合には，正のステップトルク外乱においては 1 次展開最適ゲインの方が良い結果となるが，負のステップトルク外乱においてはノイズの標準偏差が大きくなるほど，定常ゲインの方が良い結果を得られる範囲が大きくなっている．正のステップトルク外乱を加えた場合に 1 次展開最適ゲインが良い結果を示し，負のステップトルク外乱を加えた場合には 1 次展開最適ゲ

インが良くない結果を示すのは、図 5.3(c) と図 5.4(c) を見比べて、正のステップトルク外乱を加えた図 5.3(c) では、制御時間区間の後半で制御トルクがゼロを越えてマイナスとなって絶対値が小さくなり、評価関数は小さくなるのに対し、負のステップトルク外乱を加えた図 5.4(c) では、制御トルクが正の値を大きく維持して絶対値が大きくなっており、それにより評価関数が大きくなることに起因している。

また、定常ゲインによる制御のトルク〔図 5.5(c)、図 5.6(c)〕を見て気になる点を挙げれば、軌道からの誤差を小さく保つために必要な高ゲインが常にかかることとなり、制御入力がかかなり高周波の振動を伴っている点である。実際の制御系を構成する場合にこのことがデメリットとなることが考えられる。

以上のような定常ゲインとの比較から、どんな場合でも定常ゲインより 1 次展開最適ゲインの方が優れているとは言えないが、定常ゲインよりも極端に悪い結果を出すこともなく、大抵の状況で平均的に最適性の悪化の少ない制御を行うことができる。そして、試行錯誤によりどのゲインが優れているか決定する必要がないという点で、1 次展開最適ゲインを用いてリアルタイム準最適制御を行うことはメリットがあり、問題の次元数が大きくなったときには、ゲインの試行錯誤を行うことなく効果的な制御法として適用できると期待される。

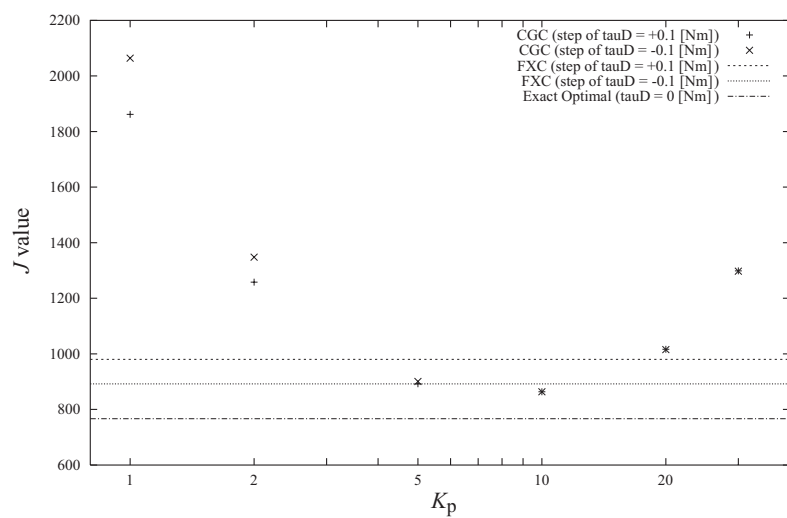


図 5.7 ステップトルク外乱を加えた場合における比較

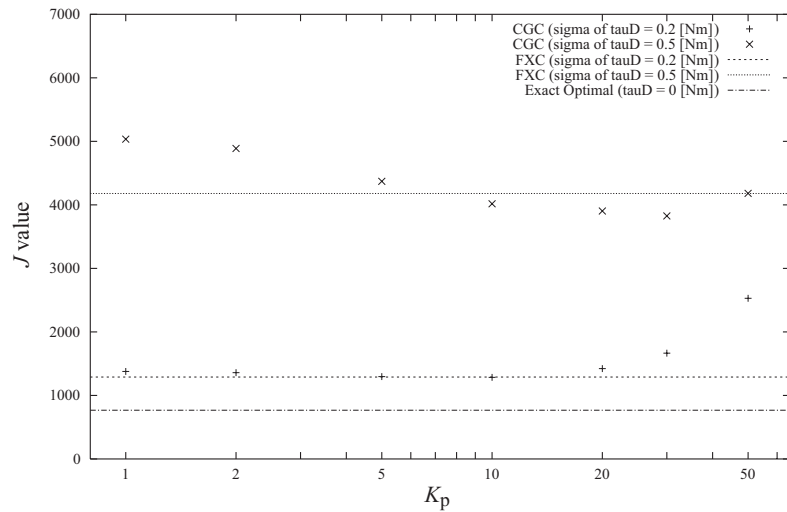


図 5.8 白色ガウストルクノイズを加えた場合における比較

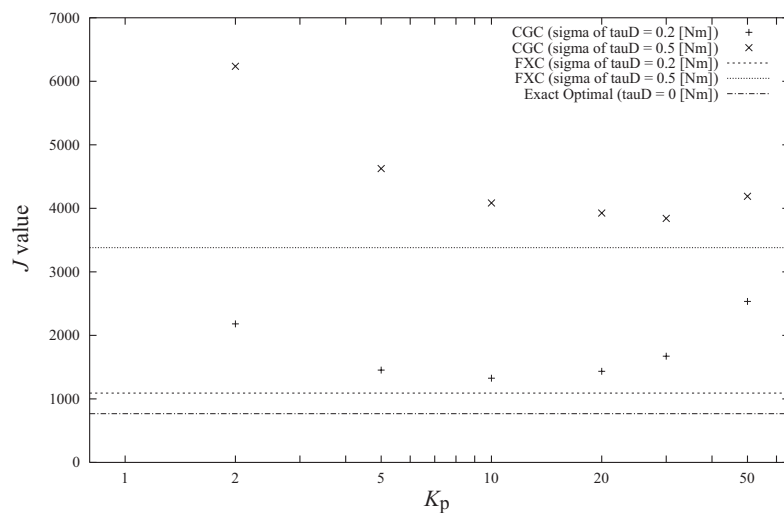


図 5.9 白色ガウストルクノイズ及び正のステップトルク外乱を加えた場合における比較

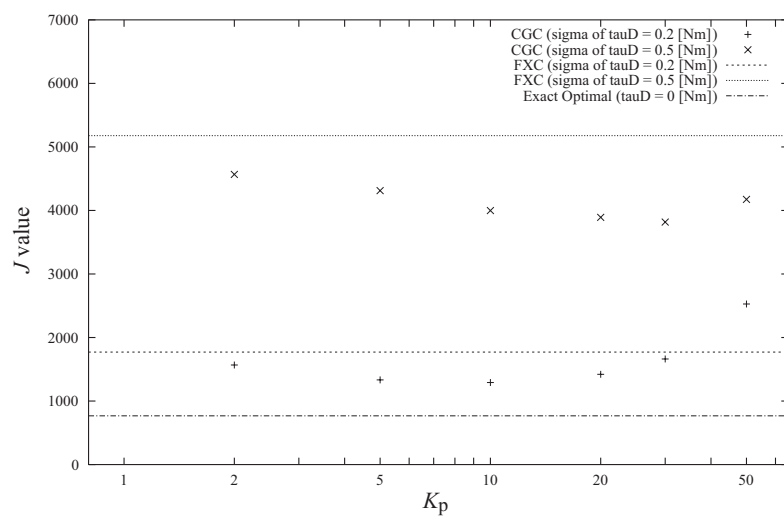


図 5.10 白色ガウストルクノイズ及び負のステップトルク外乱を加えた場合における比較

5.4 結言

5.2 節において機械システムにおいてリアルタイム準最適制御を行うために、1 次展開最適ゲインと名付けたゲインを導出し、それによりリアルタイム準最適制御を行えることを示した。5.2.1 節において、機械システムにおけるリアルタイム最適制御の可能性について検討し、5.2.2 節において、リアルタイム準最適制御を行うために、1 次展開最適ゲインと名付けたゲインを導出し、また、線形時不変系における定常最適レギュレータでは厳密に最適制御と同値となることを示した。

5.3 節において、重力の影響を受ける 1 自由度ロボット・リンク系において適用例を示し、白色ガウスノイズあるいはステップ外乱を加えた場合について、1 次展開最適ゲインを用いた制御と定常ゲインを用いた制御について、問題の評価関数の値を軌道の最適性の評価指標として比較を行った。その結論は、

- どんな場合でも定常ゲインより 1 次展開最適ゲインの方が優れているとは言えないが、定常ゲインよりも極端に悪い結果を出すこともなく、大抵の状況で平均的に最適性の悪化の少ない制御を行うことができる。
- 試行錯誤によりどのゲインが優れているか決定する必要がないという点で、1 次展開最適ゲインを用いてリアルタイム準最適制御を行うことはメリットがある。
- 問題の次元数が大きくなったときには、ゲインの試行錯誤を行うことなく効果的な制御法として適用できると期待される。

ということである。1 次展開最適ゲインによる制御における今後の課題としては、

- 実機実験による有効性の検証
- より高度で複雑な高次元の問題における有効性の検証
- 他のリアルタイム最適制御・リアルタイム準最適制御を行う手法との比較

が挙げられる。

第 6 章

結論

本論文では、機械システムのユーザインタフェースにおいて望ましい操作性を実現することに着目し、2 つの応用的な事例のアプローチを示した。また、機械システムの特性を最適化するために、軌道計画問題を最適制御問題として定式化して、最小原理に基づく新しい数値解法の提案、及び、リアルタイム準最適制御手法の提案を行った。

ハプティックインタフェースの操作感の定量的表現や系統的設計手法を実践的に考えるために、第 2 章で、実際的な対象として自動車のコラム軸アシスト型電動パワーステアリング装置 (C-EPS) を挙げて、C-EPS の準静的な操舵角-トルク特性に着目し、C-EPS において目標とする操舵角-トルク特性を実現するアシスト制御手法を提案した。提案手法により目標の特性を実現するアシスト制御マップを自動的に生成することが可能になる。

また、第 3 章では、様々なスイッチ類が配置された操作パネルを対象として、パーティクル空間内に構築された操作パネル上の複数のスイッチを操作する感覚を 1 つの遭遇型ハプティックデバイスで提示するシステムを考え、最近接点情報に基づく動作計画手法を当該システムに適用した。その中で、スイッチボックスの設計及びデバイスの動作計画を見通しよく行えるようにスイッチの分類を行い、また、スイッチによってはその法線軸回りに配置の任意性を有する場合があります、その任意性を利用してデバイスの動作負担を軽減する提示姿勢の決定手法を提案した。さらに、デバイスの動作負担を軽減す

るために、動作計画で目標提示位置・姿勢の計算に用いる重み関数の改良を行い、また、バーチャルパネル面座標系を導入して、デバイスがバーチャルパネル面に沿って動くようにすることでユーザとの不用意な衝突が起こらないように動作計画の改善を行った。

第4章では、最適制御問題に Pontryagin の最小原理を適用して、問題を随伴変数を用いて媒介変数表示し、直接解法により厳密に解く手法を提案した。提案手法は従来手法に比べて解の精度や求解に掛かる計算時間の面で優れている。提案手法及び幾つかの従来手法を Newton の最小ドラッグノーズシェイプ問題の求解に適用し、各手法による解の精度と計算時間について比較を行った。各手法とも解の精度と計算時間にはトレードオフの関係があり、実用的と思われる解の精度を得るのに掛かる計算時間を比較すると、提案手法は従来手法より計算が3~4倍高速であることを確認した。また、提案手法を実際のロボット・多リンク機構系などの機械システムに適用するための式展開を示し、1自由度ロボット・リンク系において提案手法を適用した数値例を挙げて有効性を示した。

第5章では、機械システムにおいてリアルタイム準最適制御を行うために、1次展開最適ゲインと名付けたゲインを導出し、それによりリアルタイム準最適制御を行えることを示した。第4章と同じ1自由度ロボット・リンク系を対象として、1次展開最適ゲインを求める数値例を示し、1次展開最適ゲインを用いたリアルタイム準最適制御の数値シミュレーションを行って、提案手法の有効性を示した。

今後の課題としては、第2章で述べたような系統的設計手法を発展させて、他の型のステアリング装置や航空機の操縦桿、遠隔操縦のマスターアームに応用することが考えられる。第3章で述べたようなバーチャル操作パネルシステムの実験的検証や応用も期待される。第4章で提案した最適制御問題の新しい一般解法のより高次元の問題への適用や、第5章で提案したリアルタイム準最適制御手法を具体的なデバイスに適用しての検証なども今後の課題である。また、本論文で述べてきたような、系統的設計手法、バーチャルプロトタイピング、軌道計画によるシステム動作の最適化といったインタフェースの操作感向上のためのアプローチを総括して、総合的なアプローチとしてまとめることも今後強く期待される。

参考文献

- [1] F. P. Brooks Jr., M. Ouh-Young, J. J. Batter, P. J. Kilpatrick: “Project GROPE - Haptic Displays for Scientific Visualization,” *Computer Graphics, Proc. of SIGGRAPH '90*, Vol. 24, 1990
- [2] M. M. Minsky, M. Ouh-young, O. W. Steele, F. P. Brooks Jr., M. Behensky: “Feeling and Seeing: Issues in Force Display,” *Computer Graphics, Proc. of SIGGRAPH Symposium on 3D Real-Time Interactive Graphics*, Vol. 24, 1990
- [3] G. Burdea, J. Zhuang, E. Roskos, D. Silver, N. Langrana: “A Portable Dextrous Master With Force Feedback,” *Presence*, 1991
- [4] H. Iwata: “Artificial Reality with Force-Feedback: Development of Desktop Virtual Space with Compact Master Manipulator,” *Computer Graphics, Proc. of SIGGRAPH '90*, Vol. 24, No. 4, 1990
- [5] T. Massie, K. Salisbury: “The PHANToM Haptic Interface: A Device for Probing Virtual Objects,” *Proc. ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1994
- [6] William A. McNeely: “Robotic Graphics: A New Approach to Force Feedback for Virtual Reality,” *Proc. of 1993 IEEE Virtual Reality Annual Int. Symposium (VRAIS'93)*, pp. 336-341, 1993
- [7] 大庭吉裕, 清水康夫: “EPS の慣性モーメント制御における安定性について,” *自動車技術会論文集*, Vol. 38, No. 2, pp. 207-212, 2007
- [8] 久岡雄二, 山本真規, 藤波宏明: “ドライバにとって望ましい車両応答と操舵トルク

- に関する研究,” 自動車技術会論文集, Vol. 28, No. 4, pp. 61-66, 1997
- [9] 栗重正彦, 西原修, 熊本博光: “電動パワーステアリングのオンセンタ特性向上制御方式(擬似的路面反力トルク増大制御方式),” 日本機学会論文集(C編), Vol. 73, No. 733, pp. 2464-2472, 2007
- [10] 毛利宏, 久保田正博, 堀口奈美: “過渡的な操舵力アシスト特性が車両運動に及ぼす影響,” 自動車技術会論文集, Vol. 37, No. 1, pp. 155-160, 2006
- [11] P. Raksincharoensak, 渡邊俊輔, 小竹元基, 永井正夫: “四輪操舵理論に基づくステアバイワイヤの前輪操舵制御,” 自動車技術会論文集, Vol. 36, No. 2, pp. 141-146, 2005
- [12] 葉山良平, 中野史郎, 熊本博光, 西原修: “ステアバイワイヤ車両の転舵応答性評価と運転者-自動車系の閉ループ解析,” 精密工学会誌論文集, Vol. 72, No. 11, pp. 1374-1379, 2006
- [13] 葉山良平, 中野史郎, 熊本博光: “ステアバイワイヤ車両の転舵剛性に関する実証的研究,” 精密工学会誌論文集, Vol. 73, No. 11, pp. 1220-1225, 2007
- [14] 矢船資洋, 横小路泰義: “コラム軸アシスト型電動パワーステアリング装置(C-EPS)における望ましい操舵角力特性の実現,” 日本バーチャルリアリティ学会第12回大会論文集, pp. 19-22, 2007
- [15] 片岡資章: “操舵系「曲がる」...EPS,” よくわかる図解とシミュレーションで学ぶ自動車のモーションコントロール技術入門, トヨタテクニカルディベロップメント, デンソー技研センター編, pp. 161-176, 山海堂, 2006
- [16] Susumu Tachi, Taro Maeda, Ryokichi Hirata, Hiroshi Hoshino: “A Construction Method of Virtual Haptic Space,” Proc. of 4th Int. Conf. on Artificial Reality and Tele-Existence (ICAT'94), pp. 131-138, 1994
- [17] 岩田洋夫, 中泉文孝: “凹凸面を呈示するロコモーションインタフェース GaitMaster,” 日本バーチャルリアリティ学会論文誌, Vol. 5, No. 2, pp. 863-866, 2000
- [18] 横小路泰義, 木下順史, 吉川恒夫: “3次元空間内の複数の仮想物体を提示するための遭遇型ハプティックデバイスの軌道計画,” 計測自動制御学会論文集, Vol. 40,

No. 2, pp. 139-147, 2004

- [19] 大島登志一, 山本裕之, 田村秀行: “実体触知機能を重視した複合現実感システム
自動車インテリア・デザイン検証への応用”, 日本バーチャルリアリティ学会論
文誌, Vol. 9, No. 1, pp. 79-88, 2004
- [20] William R. Provancher, Mark R. Cutkosky, Katherine J. Kuchenbecker, Gunter
Niemeyer: “Contact Location Display for Haptic Perception of Curvature and
Object Motion,” Int. Jour. of Robotics Research, Vol. 24, No. 9, pp. 691-702,
2005
- [21] Yasuyoshi Yokokohji, Norihiko Muramori, Yuji Sato, Tsuneo Yoshikawa: “De-
signing an Encountered-Type Haptic Display for Multiple Fingertip Contacts
based on the Observation of Human Grasping Behavior,” Proc. 12th Sympo-
sium on Haptic Interfaces for Virtual Environment and Teleoperator Systems,
pp. 66-73, 2004
- [22] Ken Shigeta, Yuji Sato, Yasuyoshi Yokokohji: “Motion Planning of
Encountered-type Haptic Device for Multiple Fingertip Based on Minimum
Distance Point Information,” Proc. of World Haptics Conf. 2007, pp.188-193,
2007
- [23] Shuhei Nakagawara, Hiroyuki Kajimoto, Naoki Kawakami, Susumu Tachi,
Ichiro Kawabuchi: “An Encounter-Type Multi-Fingered Master Hand Using
Circuitous Joints,” Proc. of 2005 IEEE Int. Conf. on Robotics and Automation
(ICRA2005), pp. 2667-2672, 2005
- [24] 矢船資洋, 川島達生, 横小路泰義: “仮想空間内の複数スイッチを提示するための遭
遇型ハプティックデバイスの設計と動作計画,” 日本バーチャルリアリティ学会第
13 回大会論文集, 1C2-2, pp. 205-208, 2008
- [25] 栗栖正充, 吉川恒夫: “押し作業における対象物の軌道計画,” 日本ロボット学会誌,
Vol. 13, No. 8, pp. 1115-1121, 1995
- [26] 尾崎弘明, 丘華, 林長軍: “B スプラインを用いるマニピュレータの軌道最適化にお

- けるコンプレックス法の適用,” 日本ロボット学会誌, Vol. 14, No. 4, pp. 560-566, 1996
- [27] 泉照之, Prasanna Boyagoda, 佐藤泰司, 沖田豪: “重力下で PTP 制御されるマニピュレータの消費エネルギーを最小にする最適動作時間,” 日本ロボット学会誌, Vol. 15, No. 7, pp. 1082-1088, 1997
- [28] 岩村誠人, 山本元司, 毛利彰: “非ホロノミック Caplygin System の近似最適軌道計画,” 日本ロボット学会誌, Vol. 17, No. 5, pp. 742-749, 1999
- [29] 岩村誠人, 山本元司, 毛利彰: “複数の拘束条件が課せられる最適軌道計画問題の一次数値解法とその非ホロノミック系への適用,” 日本ロボット学会誌, Vol. 19, No. 2, pp. 260-270, 2001
- [30] 竹内裕喜, 大塚敏之: “Receding Horizon Control を用いた機械的リンク系の最適化計算 連続変形法を用いた数値解に関する検討,” 日本ロボット学会誌, Vol. 17, No. 3, pp. 402-407, 1999
- [31] 竹内裕喜: “脚式ロボットのリアルタイム最適制御 等式拘束条件付 Receding Horizon Control による目標 ZMP 軌道の自動生成,” 日本計算工学会論文集, Vol. 3, Paper No. 20010001, 2001
- [32] 坂和愛幸: “最適システム制御論,” コロナ社, 1972
- [33] Arthur E. Bryson, Jr., Yu-Chi Ho: “Applied Optimal Control,” Hemisphere Pub. Corp., 1975
- [34] 志水清孝: “最適制御の理論と計算法,” コロナ社, 1994
- [35] 加藤寛一郎: “工学的最適制御,” 東京大学出版会, 1988
- [36] 吉川恒夫, 井村順一: “現代制御論,” 昭晃堂, 1994
- [37] J. C. Heideman and A. V. Levy: “Sequential Conjugate-Gradient-Restoration Algorithm for Optimal Control Problems, Part 1, Theory,” Jour. of Optimization Theory and Applications, Vol. 15, No. 2, pp. 203-222, 1975
- [38] A. K. Wu and A. Miele: “Sequential Conjugate-Gradient-Restoration Algorithm for Optimal Control Problems with Non-Differential Constraints and Gen-

- eral Boundary Conditions, Part 1,” *Optimal Control Applications and Methods*, Vol. 1, pp. 69-88, 1980
- [39] A. Miele, A. Mangiavacchi and A. K. Aggarwal: “Modified Quasilinearization Algorithm for Optimal Control Problems with Nondifferential Constraints,” *Jour. of Optimization Theory and Applications*, Vol. 14, No. 5, pp. 529-556, 1974
- [40] S. Gonzalez and S. Rodriguez: “Modified Quasilinearization Algorithm for Optimal Control Problems with Non-Differential Constraints and General Boundary Conditions,” *Jour. of Optimization Theory and Applications*, Vol. 50, No. 1, pp. 109-128, 1986
- [41] William H. Press, 他著 ; 丹慶勝市, 他訳 : “Numerical Recipes in C [日本語版],” 技術評論社, 1993
- [42] 美多勉 : “§10.2.1, 有限要素法 (FEM) とは,” *デジタル制御理論*, pp. 212-216, 昭晃堂, 1984
- [43] 矢部博, 八巻直一 : “非線形計画法,” 朝倉書店, 1999
- [44] Lawrence F. Shampine, Jacek Kierzenka, Mark W. Reichelt: “Solving Boundary Value Problem for Ordinary Differential Equations in MATLAB with bvp4c,” Mathworks (http://www.mathworks.com/bvp_tutorial), 2000

関連業績一覧

論文誌

- [1] 矢船資洋, 横小路泰義: “コラム軸アシスト型電動パワーステアリング装置 (C-EPS) における目標操舵角-トルク特性の実現手法,” 自動車技術会論文集, Vol. 40, No. 1, pp. 9-14, 2009
- [2] 矢船資洋, 横小路泰義: “遭遇型ハプティックデバイスを用いたバーチャル操作パネル上の複数スイッチ提示システム,” 日本バーチャルリアリティ学会論文誌, Vol. 15, No. 1, 11 p., 2010 (印刷中)
- [3] 矢船資洋, 横小路泰義: “最小原理を応用した最適軌道計画問題の数値解法に関する提案,” 計測自動制御学会論文集, (投稿準備中)
- [4] 矢船資洋, 横小路泰義: “1次展開によるリアルタイム準最適制御の実現,” 計測自動制御学会論文集, (投稿準備中)

国内会議

- [1] 矢船資洋, 横小路泰義, 吉川恒夫: “最小原理を応用した機械システムの最適軌道計画の数値計算法に関する提案,” 第6回最適化シンポジウム講演論文集, 116, pp. 91-96, 2004
- [2] 矢船資洋, 横小路泰義: “コラム軸アシスト型電動パワーステアリング装置 (C-EPS) における望ましい操舵角力特性の実現,” 日本バーチャルリアリティ学会第

12 回大会論文集, pp. 19-22, 2007

- [3] 矢船資洋, 川島達生, 横小路泰義: “仮想空間内の複数スイッチを提示するための遭遇型ハプティックデバイスの設計と動作計画,” 日本バーチャルリアリティ学会第 13 回大会論文集, 1C2-2, pp. 205-208, 2008
- [4] 矢船資洋, 川島達生, 横小路泰義: “遭遇型ハプティックデバイスを用いた仮想操作パネル提示システムの実現,” 日本機械学会 ロボティクス・メカトロニクス講演会 2009 講演論文集 DVD-ROM, 1P1-G04, 2009
- [5] 矢船資洋, 横小路泰義: “遭遇型ハプティックデバイスを用いた仮想操作パネル提示システム ~ 仮想パネル面座標系を用いた動作計画 ~,” 日本バーチャルリアリティ学会第 14 回大会論文集, 1A2-6, 2009