

内容梗概

本論文は、自律分散型の制御アルゴリズムによって駆動される、物体搬送システム Magic Carpet の実現を目指し、その設計とシミュレーションによる評価を行った結果を中心に述べ、あわせて、現在試作中の実験装置について経過報告を行ったものである。

マイクロマシーニング技術の発達によって、センサや電子回路を含む超小型モジュールをきわめて多数集積し、個々のモジュールが協力して仕事（たとえば物体搬送）をするという状況が生じてくる。そこで問題となるのがケーブルなど通信路の確保であり、せいぜい隣接モジュールとの交換情報をもとに動作させるアルゴリズムが必要となる。

これに対し、本論文では、まず、'場'の概念を導入して集合体のマクロな動作を決定し、次に、それを、各モジュールのミクロな制御によって実現するという2段構えの設計法を提案している。

- 修士論文 -

自律分散型物体搬送システム Magic
Carpet の設計と試作

指導教官：堀 洋一 助教授

東京大学大学院 工学系研究科 電気工学専攻

76385 及部 七郎斎

1999 年 2 月

目次

第 1 章	序論	1
1.1	自律分散研究の必要性和現状	1
1.2	マイクロマシンに見る自律分散	3
1.3	本論文の構成	4
第 2 章	問題提議	6
2.1	自律分散型物体搬送システム Magic Carpet の提案	6
2.2	Magic Carpet の工学的応用の可能性	8
2.3	自律分散集中複合システム	8
2.4	場の概念 - マクロスケールとミクロスケールの秩序性 -	10
2.5	相互干渉のシステムデザイン - 物理場の適用 -	11
第 3 章	Magic Carpet のマクロスケールでの具体的設計	12
3.1	その 1 - 振動方程式による方法 -	12
3.1.1	トラッキングレギュレータによる各モジュールの完全な制御	13
3.1.2	最適サーボ制御による各モジュールの完全な制御	14
3.1.3	マルチレートサンプリング制御による各モジュールの完全な制御	15
3.2	その 2 - ラプラス場でフィードバックによる方法 -	16
3.2.1	協調駆動のための場の適用	16
3.2.2	位置推定	17
3.2.3	境界条件の生成	20
3.2.4	物体を目標位置まで搬送するための集中系での制御のコントローラ的设计	21
3.3	その 3 - ラプラス場でフィードフォワードによる方法 -	22
3.3.1	拡張ラプラス場と曲面の発現	22
3.3.2	各モジュールに自分の位置を知らせる方法	22
3.3.3	各モジュールに目標位置を知らせる方法	24
3.3.4	拡張ラプラス場としての取り扱い	26
3.3.5	曲面の発現	27
第 4 章	Magic Carpet のミクロスケールの設計具体例 - ソレノイドプランジャの位置制御 -	28
4.1	制作を行ったリアルワールドスケールアクチュエータ	28
4.2	ソレノイドの電磁力学的特性	30
4.3	ロバストな位置制御	32
第 5 章	シミュレーション結果と実験結果	35
5.1	マクロスケール (シミュレーション結果)	35
5.1.1	振動方程式による方法 - 失敗例 -	35
5.1.2	ラプラス場でフィードバックによる方法	42
5.1.3	ラプラス場でフィードフォワードによる方法	45
5.2	ミクロスケール (実験結果)	47

第 6 章 一般的な自律分散システムに関する考察	
– Magic Carpet の設計から得られた知見より–	49
6.1 自律分散集中複合システム	49
6.2 マクロスケールのためのシステムデザイン - 一般的な線形物理場の適用 -	49
第 7 章 結言	54
発表文献	57
参考文献	58
付録 A 付録	60
A.1 自律分散条件	60
A.2 物体の運動の模擬	61
A.3 修正版 位置推定	63
A.4 DA コンバータボードの設計	67

第 1 章

序論

1.1 自律分散研究の必要性和現状

これまで「創発」とか「進化」といえば生物だけが行うことのできる極めて高度な機能と考えられてきたが、20 世紀も終わりに近づき、機械やロボットのような人工物でこのような機能を実現しようとする試みがさまざまなところで行われている。現在私たちがもつ科学技術の範囲内ではたして本当に創発や進化といった高度な機能を実現できるのだろうか。もし、実現可能であるとしたらどんな方法だろう？また、どのような分野で応用できるのだろうか？

さて、現在高度に発展した社会や経済的なシチュエーションの中、さまざまな新しいサービスや新しい商品を提供することが必要とされている。さらにいえば、そういった商品の寿命（ライフサイクル）はだんだん短くなっている。新しい商品が出たと思ったら、またすぐに新製品が発表されるという具合である。そのため工場などの製造ラインでは、プラントの情報だけでなく消費者のニーズに合わせ商品製作が今以上に必要となっている。大量生産から少数他品種への転換、シェアから高付加価値サービスへの流れ、孤立からネットワーク環境への成長、特化からオープン化への転換である [1]。

このような社会の流れの中注目されている技術が「自律分散システム」である。すなわち集中制御から分散制御への移行である。自律分散システムが注目されるのは、社会の流れだけが原因ではないかもしれない。そこには科学者や工学者の希望的観測があるのかもしれない。自律分散システムの学問領域やそこに内蔵される諸問題を考えればその学際性は容易に納得できる。言うまでもなく、自律分散システムの学問領域は、情報・システム・制御工学の各分野を含む。さらに加えて自然科学から社会科学までのさまざまな学問と広く関わりをもっている。また、自律分散システムに関連する学問領域の全て、現代科学の根幹に関する重要な事柄ばかりであり、散逸構造、シナジェティクス、カオス、遺伝子など、各分野でホットになっている話題を幾つかあげることができる。

自律分散システムとはどんな形態のシステム構造を指しているのだろうか？簡単にいえ

ば自律した分散システムといえるのかもしれない。そうなのだろうか？自律したシステムの分散形態かもしれない。また、自律とは何か、分散といってもどの程度の分散なのかと疑問はつるばかりである。この現状は、さまざまな研究者がさまざまなことを主張し自律分散システムの定義に対するコンセンサスがとれていないことに原因がある。そこで、一般的にいわれていることをもとに自律分散システムの定義を付録に示した [2]。

自律分散システムが必要視される理由は、実現に伴うメリットも大きいからである。必要条件的なアプローチに制御の質の向上がある。温度制御を例にあげて考えてみよう。温度制御においては大きな 1 つのヒータで制御するより、分散配置された複数の小さなヒータの方が温度むらを少なくすることができる。また、ヒータのマルチ化に付随して省エネも可能である。ヒータを分散配置をすれば、必要な場所だけ通電すれば良く、不要な場所の温度を上げる必要がない。このような事は 1 つのヒータではどうやってもできない [3]。

次にあげられるものとして、分散化による耐故障率の向上がある。これは、アクチュエータが単数の場合と複数の場合との耐故障率の挙動を比較すれば容易に理解できる。アクチュエータが 1 つの場合、1 つのアクチュエータが系の挙動を決定するエネルギーを発生するわけだから、故障が起これば致命的である。また、1 つのアクチュエータに大きなエネルギーを担わせているわけだから、暴走した場合は悲劇的である。10 トンクラスのトラックが 100km で暴走していることを想像したらその恐ろしい状況が頭に浮かぶであろう。次にアクチュエータがマルチな場合を考えてみよう。例えばこんな事を考えると分かり易い。アクチュエータが故障する確率を 0.01 と仮定しよう。アクチュエータが全部で 3 個ある場合は少なくとも 1 つは故障しない確率は $0.999999(1 - 0.01^3)$ となり、1 つの場合 (0.9) と比べて分散化に伴う耐故障率の向上がよくわかる。

自律分散システムの研究のモチベーションとして次のようなキャッチフレーズがある。「生物に学ぶしなやかなシステムの実現に向けて」生物から学ぶことは非常に大切だと思うがそれがそのまま工学のシステムに適用できるほど甘いものではない。そのことを生物と人工物の違いをもとにして考えてみよう。

生物と人工物では、まず、使用する材料が異なる。生物が蛋白質や核酸など、分子 1 個が機能を持つような生体高分子を材料とするのに対して、人工物では金属やプラスチックなどの均質な材料を一定の形状に成形することで機能を持たせるのが普通である。このため、人工物のスケールの方が生物のそれより何桁も大きくなることはやむを得ない。また、使用する力も違う。生物が分子間に作用する化学的な結合力を用いるのに対して、人工物はもうもう少しマクロなスケールで働く物理的な結合力を用いる。例えば、静電気力、表面張力、電磁気力、機械的なメカニズムによる結合力などである。また、エネルギーの供給方法もことなる。生物では解糖反応により ATP を生成し内部にエネルギー源を持っている。それに対して人工物では外部に電源等の物理的なエネルギー源を求めるのが一般的で

表 1.1: 人工物と生物の違い

	生物	人工物
材料	生体高分子	無機質
作用力	化学的結合	物理的結合
エネルギー	解糖反応により自給	外部から供給
ハード	DNA と蛋白質	論理回路
ソフト	アルゴリズムは未知	集中または分散アルゴリズム
アルゴリズム	複雑な反応ネットワークで、曖昧	決定論的

ある。これらの違いを表 1.1 にまとめてみる [4]。

以上に示したように生物と人工物とは本質的な違いがある。したがって、生物から学んだことをそのまま人工物に適用することはほぼ不可能である。生物から学んだことをベースにして、人工物のための理論体系を構築することが現在必要されているのではないだろうか。

1.2 マイクロマシンに見る自律分散

近年の科学技術の発展により、今まで SF の世界でもてはやされていたことが現実になりつつある。SF 映画の『ミクロの決死圏』で人体の中に超小型の潜水艦で入り、患部を治療しているシーンを憶えていることでしょう。そう、マイクロマシンとはその映画の中で出てくる超小型の潜水艦のイメージなのである [5]。

マイクロマシンの特徴 [6] は、

- マイクロ化
- 多数性
- マイクロエレクトロニクス

の 3 つを挙げることができる。マイクロ化は小さく軽く速い振る舞いを意味し、多数性はマイクロ化により小さく軽くなるため多数で大きな仕事を成し遂げるための多数化である。また、マイクロエレクトロニクスは半導体加工技術に基づくマイクロマシン製法が最も有力であるため含まれる要因である。半導体加工技術を用いれば、マイクロマシンと一緒に演算機能を持つマイクロプロセッサやセンサを組み込むことができる [7] [8] [9]。

マイクロマシンは非常に小さいし、それ一個が持つ能力は非常に小さい。そのためたくさんのマイクロマシンを協調して動かす必要がある。マイクロマシンはプロセッサやセン

サなども一緒に含まれているためそれ自身も自律した振る舞いを行うことが可能である。自律したマイクロマシンをたくさん協調させるためには自律分散の考えが必ず必要になる。

自律分散の考えはまだ始まったばかりであり、現在研究の初期段階といえる。そのため自律分散制御を実現する一般的な手法は全く考えられていない。マイクロマシンに代表される自律分散的なシステムが近い将来簡単に実現できるようになったとしてもその制御手法は全くないと考えられている。そのため自律分散制御実現問題を考えることは非常に急務であり、自律分散制御の一般的な制御アルゴリズムやシステム形態を研究することは非常に重要であり、新しい工学システムを構築する上で避けて通ることができない。

マイクロマシンが内包する問題

1. 通信路確保の難しさ

- マイクロマシンは超小型のため、制御用の信号線でも非常に膨大になってしまふ。そのため非常に近くにあるマイクロマシンとしか通信できない。そのため集中制御は構造上不可能である。

2. 全てのマイクロマシンモジュールはハードもソフトも同質

- マイクロマシンの制作、加工において最も有望視されているのが半導体加工技術に起因するシリコンウエハ上にマイクロマシンを作成する方法である。半導体加工技術を用いているため同じものをたくさん作ることは非常に得意であるが、異なった構造のものをたくさん作ることは向いていない。

以上に示した問題はマイクロマシン特有の問題であり、またマイクロマシン制作上本質的であり決して避けて通ることができない。付録 1 に自律分散条件を示したが、通信路確保の難しさに起因する隣接モジュールのみに通信可能ということは通信の局所性に当てはまる。またマイクロマシンモジュールはハードもソフトも同質であるということは同質性や均質性に当たる。すなわちマイクロマシン用の制御アルゴリズムは自律分散制御の典型といえる。

1.3 本論文の構成

本論文はマイクロマシンに見る自律分散を考え、具体例として自律分散型物体搬送システム Magic Carpet の設計を考える。本論文では主としてこの Magic Carpet のための制御を考えた。第 2 章で本論文の中心問題となる Magic Carpet 問題を提議する。まずはじめに場の概念により Magic Carpet 問題をマクロスケールの秩序とミクロスケールの秩序の

2つに分類する。それぞれについてその具体的な制御法を第3章、第4章に示した。第3では振動方程式による方法、ラプラス場でフィードバックによる方法、ラプラス場でフィードフォワードによる方法の3つの手法により具体的な設計を行った。第4章では、現在製作中の実験装置の一部であるチューブラソレノイドのストロークのロバストな位置制御を実現するためにPDコントローラと外乱オブザーバの複合による方法とPコントローラによる方法を実験でその応答特性を示し制御を行った。第6章には Magic Carpet 問題から得られた知見をもとにして一般的な自律分散システムに関する考察を行った。最後に結言を第7章に示した。

第 2 章

問題提議

2.1 自律分散型物体搬送システム Magic Carpet の提案

本章では序論での問題点をふまえた上で問題の提議を行う。

問題の説明

図 2.1 のような 2 次元平面にたくさん配列したリニアアクチュエータによる物体搬送システム Magic Carpet を作りたい。

搬送物体 (ボール) を運ぶ。

$$(x_{ini}, y_{ini}) \Rightarrow (x_c, y_c)$$

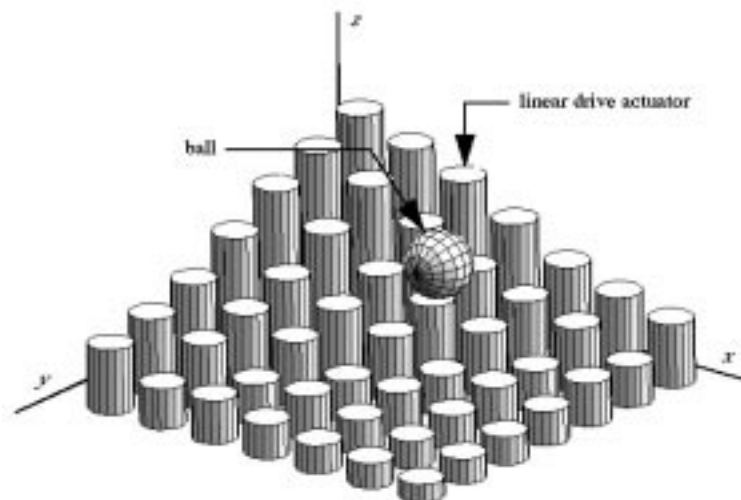


図 2.1: magic carpet のイメージ図

2次元平面 ($x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}$) に x 方向に m 個、 y 方向に n 個配列したアクチュエータ、センサ、多少の計算機能をもつ計算ユニットがひとまとまりになったマイクロマシンモジュールによる物体搬送システムを構成する。問題として以下のようなシステムを設定する。

センサの動作

- モジュール上に物体があるかないかを判別する。また、どの程度の重さがかかっているかもわかる。

アクチュエータの動作

- z 方向にリニアドライブすることができ、針のようなものを出したり引いたりができる。また、その針のようなものの変位は連続的なものである (*on* か *off* のような 2 値ではない。)

計算ユニットの動作

- 非常に単純な計算しかできない (四則演算程度)。また、限られたユニットとして通信できない (隣接するユニット、通信の局所性)。

上記制限のもとで、次の目的を達成する自律分散アルゴリズムを考える。

1. たくさんのモジュールを協調させ駆動させる。
2. 物体を目標位置まで搬送するシステムを構成する。

表 2.1: 問題の補足的説明

各モジュールの条件	設定
通信範囲	となりとだけ
自分の座標	わからない
方向性	わからない
対象物の位置	わからない
アクチュエータの先	滑らか
隣との相互干渉	自由に設定できる。(前もって決めておく)
ストレージ	多少ある
隣との通信速度	速い

2.2 Magic Carpet の工学的応用の可能性

本節では提案した Magic Carpet の工学的応用の可能性について考えてみる。

マイクロマシンでの実現を考えているということ、またマイクロマシン特有の問題をクリアするためのアルゴリズムを考えている。

マイクロマシンの大きな特徴は、**非常に小さい**ということである。そのためマイクロマシンによる物体搬送システム Magic Carpet は高精度の位置決めでの物体搬送を実現できる。

高精度の位置決めができるため、

1. 顕微鏡の対象物体の位置決め
2. マイクロファクトリー内での物体搬送

ができる。顕微鏡での対象物体の位置決めは対象物体が非常に小さく、顕微鏡のピントの合うところが非常に狭いため高精度の位置決めができれば非常に便利である。幼い頃、顕微鏡の対象物体を動かしても上手くピントが合うところに移動ができなくて苦労したのを憶えているだろう。電子顕微鏡など高倍率になればなるほどその必要性は増してくるだろう。また、マイクロファクトリーは被工作物が非常に小さいためその必要があるだろう。

2.3 自律分散集中複合システム

Magic Carpet のシステム形態はどのようなものが考えられるのだろうか？

ひとつの極として完全な集中制御系を考えるのなら、その極と相対する形で自律分散システムが存在するだろう。では、その中間にはどんなシステム形態が存在するのだろうか（図 2.2 参照）。

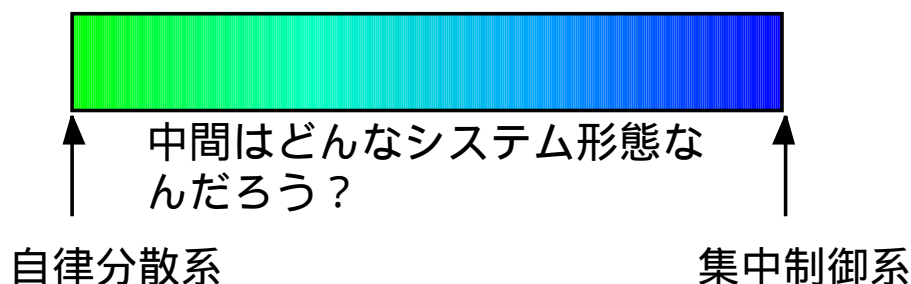


図 2.2: システム形態

集中制御系は今日までに多くの研究がなされたくさんの成果が上げられている。また、方法論も確立されているため制御系の設計は大変行いやすいものになっている。自律分散

系の研究は研究分野として確立されたばかりであり、研究の初期段階といえる。現在高機能を追求するためにたくさんのモジュールが機器に追加され複雑化の一途をたどっている。そういった複雑なものを複雑なものと捉えるのが自律分散的な考え方である。したがって、設計のし易さを考えるならば集中制御系を考えた方がよく、複雑化の流れをくみ取るならば自律分散的な考え方をすべきである。

なんとか上手く Magic Carpet に集中制御系と自律分散制御を複合したものを適用できないだろうか？

ここで提案し Magic Carpet に適用するシステム形態は自律分散系と集中制御系を合わせたものである。集中制御系と自律分散系の良いところをかねそなえたものである。以下に自律分散系と集中系を複合させたシステムの簡単なイメージ図（図 2.3 参照）を示す。

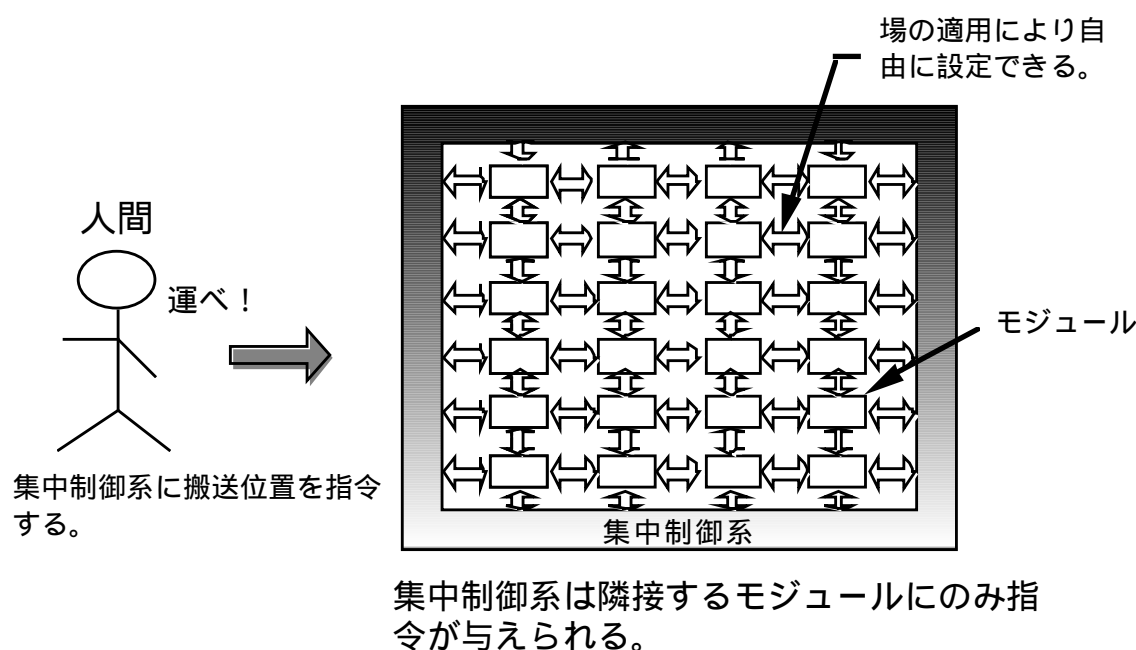


図 2.3: 自律分散集中複合系のイメージ図

Magic Carpet に自律分散集中複合系の適用を考えた場合、大きく分けて3つのことを考えなくてはならない。

1. 集中系での制御（マクロな制御）
2. 各モジュール間での相互干渉
3. 分散モジュールでの振る舞い（ミクロな制御）

以上を簡単にまとめ、着眼する要素として図 2.4 に示した。

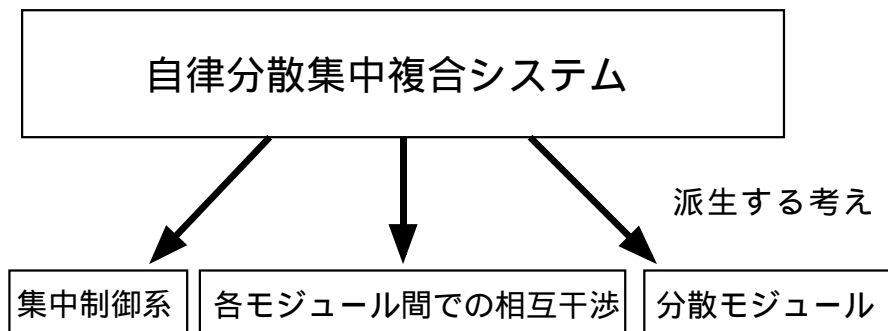


図 2.4: 着眼する 3 つの要素

2.4 場の概念 - マクロスケールとミクロスケールの秩序性 -

前節で着眼する要素 (図 2.4) として、3 つのことを考える必要性について考えた。この 3 つのことを考えるに当たり大きな示唆を与える概念がある。それは『場の概念』 [10] [11] と呼ばれるものである。

場の概念とは、

『自律分散系は一般にたくさんのモジュールにより構成される複雑なシステムである。したがって、各モジュールの振る舞い (ミクロ的振る舞い) とたくさんのモジュールで構成される全体の振る舞い (マクロ的振る舞い) の二つを考えなくてはならない。例えば、社会システムを自律分散系と見るならば、各個人の行動や考えがミクロ的振る舞いに対応し、社会の流行や世論などがマクロ的振る舞いに対応する。社会の流行や世論は "場" の一種である。すなわち場とは全体の秩序性のことである。また、ミクロな振る舞いが比較的短い時間スケールであるのに対して、マクロな振る舞いが長いタイムスケールで進行するのも場の特徴である。』

と言われている。これを Magic Carpet 問題に適用することを考えてみよう。

この概念は着眼する要素の 3 つのうちの集中制御系 (マクロスケールの制御) と各モジュールの振る舞い (ミクロスケールの制御) との関係を示唆するものであり、各モジュールの振る舞いは比較的速い周期で行うべきであり、また集中制御系での制御周期は比較的ゆっくりであり全体の流れを組んでいくような形であるべきであるということについていっている。また、2 つの秩序性を考えながらシステムを設計しなくてはならないことをいっている。

2.5 相互干渉のシステムデザイン - 物理場の適用 -

前節で着眼する 3 つの要素のうち 2 つの関係は場の概念で深く相関があることについて述べた。

各モジュールは各々勝手に振る舞って良いわけであるが、何らかの形で協調していなくては物体搬送の目的は達成できないし、集中制御系からの情報も得ることができない。そのため協調のための場は集中制御系とのやりとりが簡単であり、強度に機能を限定した各モジュールで実現できるものでなくてはならない。また、通信の局所性も満たさなくてはならない。

『場の満たす条件』

1. 集中制御系とのやりとりが簡単で明確
2. 通信の局所性を満足

でなくてはならない。そこでその場として物理場の適用を提案する。

自然界は非常にリーズナブルに構成されている。太陽系の惑星などを例に挙げると分かり易いかもしれない。たくさんの惑星は相互に干渉しながらバランスをとって平衡状態を保っている。そのような自然界に存在するリーズナブルさをモジュールの協調のための場に適用することを考えた。自然界の振る舞いを支配する支配方程式は偏微分方程式で記述されることが多い。偏微分方程式はもちろん偏微分を含んでいる。偏微分は非常に近傍の値が重要になる。偏微分方程式によって支配される場は近傍の値だけをもとにして特徴付けられていることを意味する。ここで、近傍の値だけで特徴づけられているということに注目しよう。場の条件として通信の局所性を考えているからこの近傍の値だけで特徴づけられる場というのは非常に魅力的である。

つぎに集中制御系とのやり取りを考えてみよう。集中制御系とのやり取りは境界条件と形で行うことができるだろう。

第 3 章

Magic Carpet のマクロスケールでの具体的設計

本章では場の概念で示唆されるところのマクロスケールでの制御を考える。集中制御系によるマイクロマシンモジュール群の制御である。

3.1 その 1 - 振動方程式による方法 -

本節では物理場として減衰項を含む膜の振動方程式 [12] によるマイクロマシン群の完全な制御を試みる。振動方程式は時間微分を含み、双曲型に分類される偏微分方程式である。

注意して欲しいのは方程式 (3.1) または式 (3.2) において出現している z, z_{ij} はリアドライブのアクチュエータの変位であり、マイクロマシンモジュール群の並びが x, y 方向であるのに対して z 方向の意味である。

$$\frac{\partial^2 z}{\partial t^2} + k \frac{\partial z}{\partial t} = c^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right) \quad (3.1)$$

このダイナミクスを含む方程式 (3.1) に対して、マイクロマシンの並びを方程式の空間方向に割り当て離散化を施し、状態方程式の形で記述する。そのとき、 ij 番目の状態方程式は、

$$\frac{d}{dt} \begin{bmatrix} z_{ij} \\ \dot{z}_{ij} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c^2 \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} \right) & -k \end{bmatrix} \begin{bmatrix} z_{ij} \\ \dot{z}_{ij} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{c^2}{\Delta x^2} & \frac{c^2}{\Delta x^2} & \frac{c^2}{\Delta y^2} & \frac{c^2}{\Delta y^2} \end{bmatrix} \begin{bmatrix} z_{i+1,j} \\ z_{i-1,j} \\ z_{i,j+1} \\ z_{i,j-1} \end{bmatrix} \quad (3.2)$$

となる。得られた状態方程式を分散モジュール全てでとりまとめると、結局

$$\dot{x} = Ax + Bu \quad (3.3)$$

となる。ただし、ここで x は $[z_{ij} \quad \dot{z}_{ij}]^T$ を含む状態であり、 u は境界条件である。式 (3.3) より全てのモジュールの高さを状態方程式で記述することができる。

得られた状態方程式をもとに全てのモジュールのアクチュエータの高さをコントロールする方法を考えよう。状態方程式に直すことができたため制御工学の道具をそのまま適用できる。

得られた状態方程式をもとにして、集中制御系からの各モジュールの制御（マクロな制御）としてトラッキングレギュレータ、最適サーボ、マルチレートサンプリングの 3 つの手法を試す。

3.1.1 トラッキングレギュレータによる各モジュールの完全な制御

まずはじめに最も簡単な方法として、トラッキングレギュレータ [13] による集中制御系での各モジュールの変位の完全な制御を目指した。

境界条件としての入力に比べ、制御したい量の数が非常に多いため状態方程式のシステム行列 B が縦長になる。正方でないから逆行列が存在しない。そのため疑似逆行列を用いることにする。疑似逆行列として Moore-Penrose の一般逆行列 [14] を用いることにした。 B の疑似逆行列をここでは B^\dagger で示すことにした。

$$x = x_0 + \Delta x \quad (3.4)$$

$$u = u_0 + \Delta u \quad (3.5)$$

ここで、 x_0, u_0 は動作点である。すなわち、

$$\dot{x} = Ax + Bu = 0 \quad (3.6)$$

であり、

$$u_0 = -B^\dagger Ax_0 \quad (3.7)$$

となる。また、図 3.1 の F は極配置法により決定する。

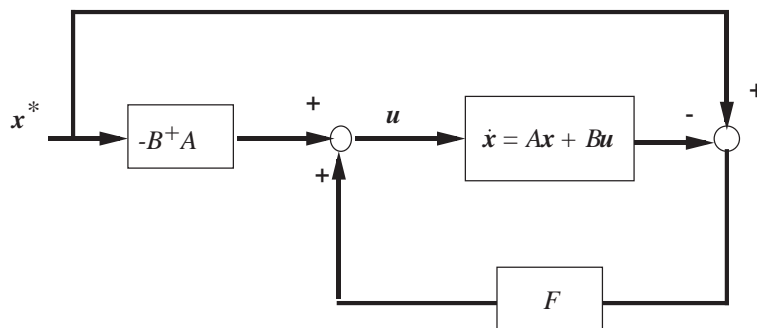


図 3.1: トラッキングレギュレータのブロック線図

3.1.2 最適サーボ制御による各モジュールの完全な制御

つぎに最適サーボ制御 [15] による方法を考える。状態方程式

$$\begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} = C\boldsymbol{x} \end{cases} \quad (3.8)$$

に対して、

$$\boldsymbol{v} = \frac{d}{dt}\boldsymbol{u} \quad (3.9)$$

を定義し、式 (3.8) を書き直す。書き直すと

$$\frac{d}{dt} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \boldsymbol{v} \quad (3.10)$$

$$\boldsymbol{y} = [C \ 0] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{u} \end{bmatrix} \quad (3.11)$$

となる。出力 $\boldsymbol{y} = \boldsymbol{r}$ となれば、 $\boldsymbol{x}, \boldsymbol{u}$ も一定値 $\boldsymbol{x}_s, \boldsymbol{u}_s$ となり、

$$\begin{bmatrix} 0 \\ \boldsymbol{r} \end{bmatrix} = \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{u}_s \end{bmatrix} \stackrel{\text{def}}{=} Z \begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{u}_s \end{bmatrix} \quad (3.12)$$

となる。また、

$$A^\# \stackrel{\text{def}}{=} \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \quad (3.13)$$

$$B^\# \stackrel{\text{def}}{=} \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (3.14)$$

$$C^\# \stackrel{\text{def}}{=} [C \ 0] \quad (3.15)$$

を定義すると、最適サーボ制御のゲイン K_1, K_2 はそれぞれ、

$$\begin{aligned} \boldsymbol{v} &= FZ^{-1} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{y} \end{bmatrix} - FZ^{-1} \begin{bmatrix} 0 \\ \boldsymbol{r} \end{bmatrix} \\ &= -K_1\boldsymbol{x} - K_2(\boldsymbol{y} - \boldsymbol{r}) \end{aligned} \quad (3.16)$$

となる。ここで、 $F = -R^{-1}B^{\#T}P$ で P は

$$A^{\#T}P + PA^\# + Q - PRB^\#R^{-1}B^{\#T}P = 0 \quad (3.17)$$

の Riccati 方程式の解である。また、 Q, R はそれぞれ、 $\delta\boldsymbol{x} = [\boldsymbol{x} - \boldsymbol{x}_s \ \boldsymbol{u} - \boldsymbol{u}_s]^T, \boldsymbol{v}$ に対する評価関数上での重みである。これにより最適サーボにより場を制御できる。ただし、 Z^{-1} を求める際に疑似逆行列を用いたので最適サーボ条件を満たしていない。それ故最適サーボ本来のパフォーマンスは得られないことは特筆すべきことである。

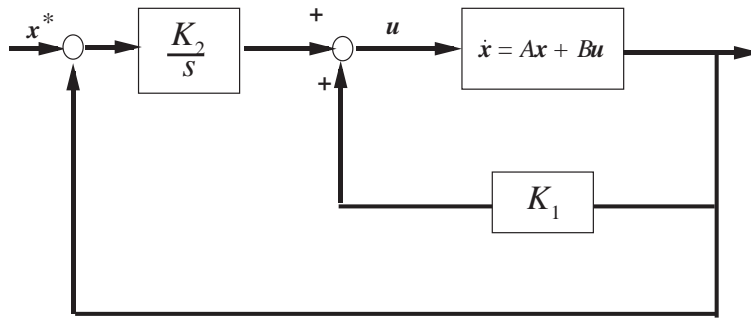


図 3.2: 最適サーボのブロック線図

3.1.3 マルチレートサンプリング制御による各モジュールの完全な制御

最後にマルチレートサンプリング制御 [16] [19] による場の制御を考える。図 3.3 にブロック線図を示した。ここで H_M, S はそれぞれ、マルチレートホールダ、サンブラである。

状態方程式 (3.18) に対して、マルチレートサンプリング時間による離散化を施せば、

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \quad (3.18)$$

は、

$$\begin{cases} \dot{x} = A_m x + B_m u \\ y = C_m x \end{cases} \quad (3.19)$$

と変換できる。ここで、 A_m, B_m, C_m はマルチレートサンプリング時間により離散化されたシステム行列である。これらの行列に対して、

$$F = -B_m^{-1} A_m \quad (3.20)$$

$$K = B_m^{-1} \quad (3.21)$$

を決定し、制御を試みる。ここでマルチレートサンプリング時間により離散変換されたシステム行列の B_m は正方で正則である。

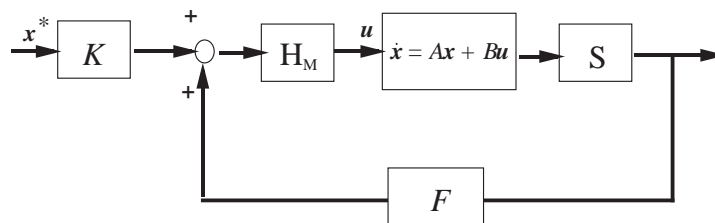


図 3.3: マルチレートサンプリング制御のブロック線図

3.2 その 2 - ラプラス場でフィードバックによる方法 -

本章では物体搬送のためのラプラス場による協調と、ポアソン場による搬送物体の位置の推定の 2 つのことについて説明する。ラプラス場によりモジュール群を協調させておき、協調されたそれらを集中制御系により制御することを目的とする。マクロな制御として相互干渉にラプラス場を適用し、集中制御系でモジュール群の制御を行う。前節の振動方程式による方法との違いは自律分散フィードバック制御を提案し、フィードバック側で位置推定により得られた信号を用い、搬送物体の位置制御を行うことである。図 3.4 に全体の構成図を示した。各ブロックの説明を以下行う。

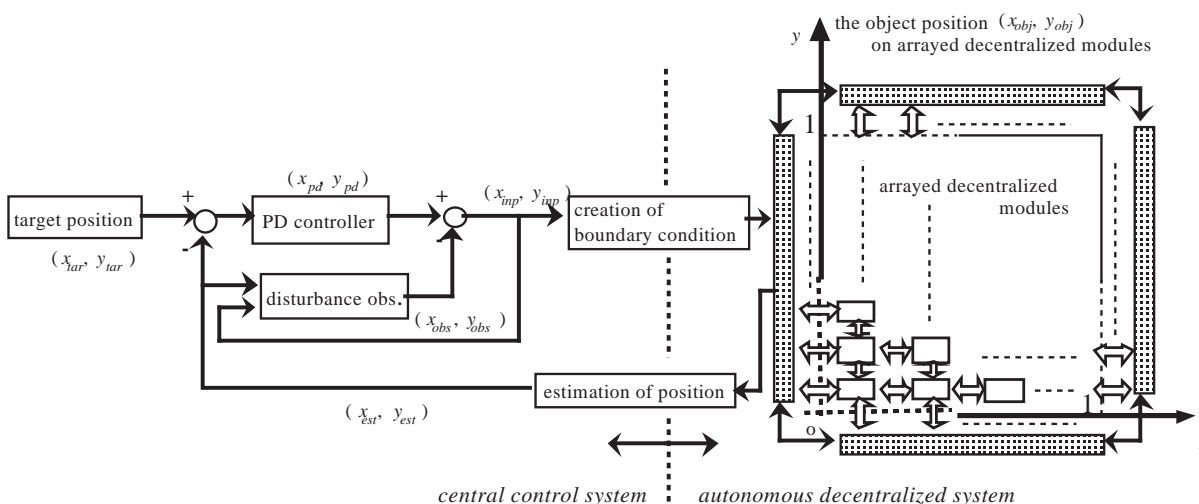


図 3.4: 自律分散フィードバック制御

3.2.1 協調駆動のための場の適用

Magic Carpet のためのシステム形態として自律分散集中複合システムの適用を考えた。またモジュール間の相互干渉としてラプラス場の適用を考える。そのラプラス場により各モジュール群を協調する。なぜラプラス場の適用を考えたかといえば、非常になじみの深い場であるからである。なじみが深いから場の特徴をよくつかめる。また自然界に存在するリーズナブルさをまねしたいと考えたせいもある。

$$\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = 0 \quad (3.22)$$

自然界のリーズナブルさを適用しようとして、式 (3.22) の適用を考えた訳であるが実際に適用する場合は離散化を行う。マイクロマシンモジュールは非常にたくさんあることを

仮定しているが無限にあるわけではない、実際には有限個である。そこで式 (3.22) に差分近似を施した、

$$z_{ij} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (z_{i+1,j} + z_{i-1,j}) + \frac{1}{\Delta y^2} (z_{i,j+1} + z_{i,j-1}) \right\} \quad (3.23)$$

をもちいることにする。境界条件の設定を工夫してやれば、協調駆動しているマイクロマシンモジュールをこれにより上手く動かすことができる。

3.2.2 位置推定

本節では分散的に位置推定を行う方法について提案する。物体の位置推定には協調駆動と同様に場を生成することにより行う。生成する場は協調駆動で用いているラプラス場と非常に類似しているポアソン場である。自律分散条件を満たすため、本論文では一貫して集中系で取り扱える情報は境界での値のみとしている。したがって、位置推定にも境界でも値をもとにして計算を施し、できるだけ正確な位置を導出できるようにしている。

ポアソン方程式 (3.24) は、

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f(x, y) \quad (3.24)$$

であり、強制振動項 $f_{i,j}$ の設定を工夫することにより分散的な方法により位置推定が可能になる。

$f_{i,j}$ の設定アルゴリズムを以下に示す。

$$r_{i,m} = \left[(x_p - x_i)^2 + (y_p - y_m)^2 \right]^{1/2} \quad (3.25)$$

$$r_{j,m} = \left[(x_p - x_j)^2 + (y_p - y_m)^2 \right]^{1/2} \quad (3.26)$$

$$r_{i,n} = \left[(x_p - x_i)^2 + (y_p - y_n)^2 \right]^{1/2} \quad (3.27)$$

$$r_{j,n} = \left[(x_p - x_j)^2 + (y_p - y_n)^2 \right]^{1/2} \quad (3.28)$$

$$f_{i,m} \propto \frac{\beta}{r_{i,m}} \quad (3.29)$$

$$f_{j,m} \propto \frac{\beta}{r_{j,m}} \quad (3.30)$$

$$f_{i,n} \propto \frac{\beta}{r_{i,n}} \quad (3.31)$$

$$f_{j,n} \propto \frac{\beta}{r_{j,n}} \quad (3.32)$$

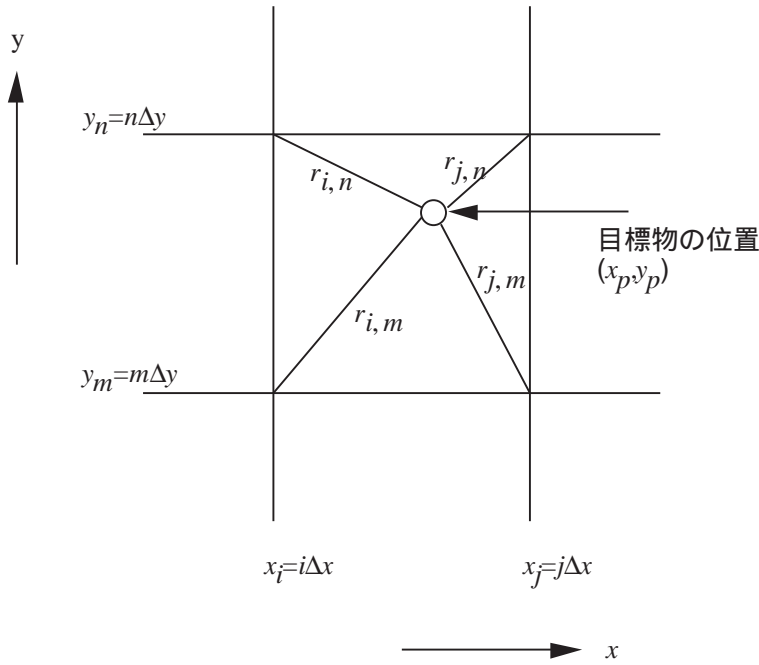


図 3.5: グリット間拡大図

ここで、 β は任意に設定できる正の定数である。これにより場を生成すると図 3.6 のような等高線で示す場が生成できる。

図 3.6 よりわかるように境界で極値を取っている。この極値を取っている点はほぼ搬送物体の位置 (x_p, y_p) である。この点を探し出せば、分散的手法により物体の重心を探し出すことができる。

ここで問題となるのはモジュールが空間的に分散的に配置されているために探し出した場所がモジュールの存在する場所だけであり、位置検出の分解能がモジュールの間隔程度になってしまうことである。それを補うために以下に連続的に位置検出を行う方法を示す。

まずはじめに極値を取るモジュールの場所を検出する。これは単に境界に存在するモジュールでの値を検索していき最小値を探すという非常に簡単なことである。そのあと探し出したモジュールの両脇の座標とその点での生成された値を取り出す。その値をもとにして、その間を 2 次関数を用いて補間を行うことにより本来取るべき極値の位置を連続的に探し出す。以下に数式と図を用いて説明を行う。

$$x_p^{est} = \frac{x_b^2 u_a - x_c^2 u_a - x_a^2 u_b + x_c^2 u_b + x_a^2 u_c - x_b^2 u_c}{2(x_b u_a - x_c u_b - x_c u_b + x_c u_b + x_a u_c - x_b u_c)} \quad (3.33)$$

$$y_p^{est} = \frac{y_b^2 u_a - y_c^2 u_a - y_a^2 u_b + y_c^2 u_b + y_a^2 u_c - y_b^2 u_c}{2(y_b u_a - y_c u_b - y_c u_b + y_c u_b + y_a u_c - y_b u_c)} \quad (3.34)$$

以上の方法により x_p の推定量 x_p^{est} と y_p の推定量 y_p^{est} らを得ることができる。

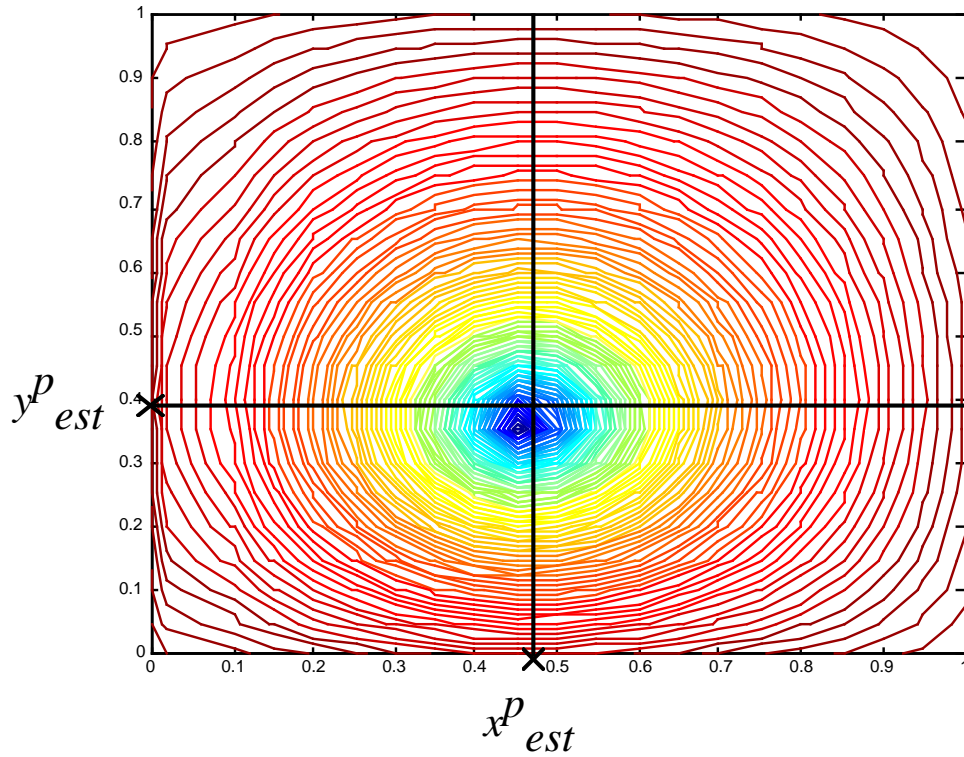


図 3.6: 生成された Poission 場

Poission場により生成された値

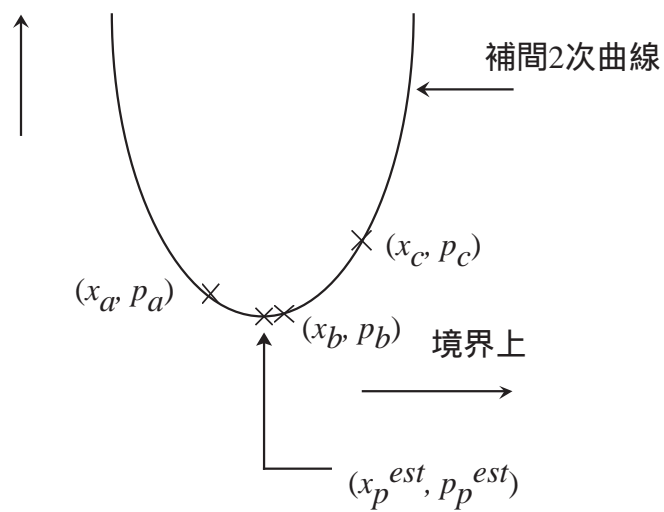


図 3.7: 境界上での値と補間曲線

3.2.3 境界条件の生成

本節ではラプラス場により協調されたモジュール群を駆動するための境界条件の設定法について説明する。

現在位置は前節で説明した位置推定アルゴリズムにより知ることができる。また、目標位置はこちらが任意に設定できる値である。目標位置、現在位置、境界 (1) から (4) を図 3.8 のように仮定しよう。ただし、図 3.8 は図 2.1 を上から見た図である。

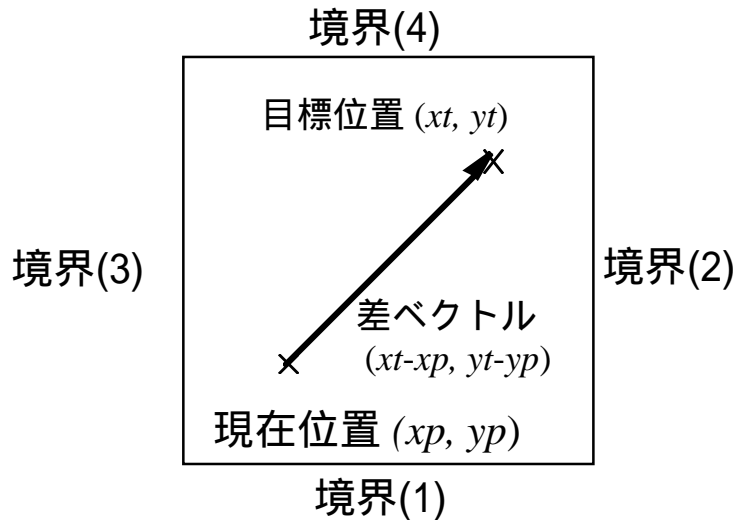


図 3.8: 境界条件の作り方

case1 $x_t - x_p > 0, y_t - y_p > 0$

境界 (3) 上のモジュールの高さを $|x_t - x_p|$ に比例させ上げる。

境界 (1) 上のモジュールの高さを $|y_t - y_p|$ に比例させ上げる。

case2 $x_t - x_p < 0, y_t - y_p > 0$

境界 (2) 上のモジュールの高さを $|x_t - x_p|$ に比例させ上げる。

境界 (1) 上のモジュールの高さを $|y_t - y_p|$ に比例させ上げる。

case3 $x_t - x_p > 0, y_t - y_p < 0$

境界 (3) 上のモジュールの高さを $|x_t - x_p|$ に比例させ上げる。

境界 (4) 上のモジュールの高さを $|y_t - y_p|$ に比例させ上げる。

case4 $x_t - x_p < 0, y_t - y_p < 0$

境界 (4) 上のモジュールの高さを $|x_t - x_p|$ に比例させ上げる。

境界 (2) 上のモジュールの高さを $|y_t - y_p|$ に比例させ上げる。

と 4 通りに場合分けを行い、それぞれの条件にしたがって境界条件を生成する。

なぜこのような 4 通りに場合分けをし、境界条件を生成するのであるがこれはラプラス場の特徴が明確にわかっているからである。ラプラス場は領域内部に極値をもつようなことはできないが、連続な斜面を正確に作るができる。ゆえに以上で示した 4 通りの分類を行い境界条件を生成すれば搬送物体が斜面を転がるというスタイルで物体搬送を行うことができる。

斜面を転がしながら運ぶという方法において、境界を差ベクトルの大きさに比例させて上げるとい比例制御では搬送物体を目標位置に収束させることができない。そこで比例制御を行う代わりに PD コントローラに外乱オブザーバを組み合わせたロバストな搬送物体の位置制御を行うことにする。そのコントローラの設計を次節で説明する。

3.2.4 物体を目標位置まで搬送するための集中系での制御のコントローラ的设计

以上で協調駆動のための場の適用、物体の位置推定について考えた。本節では搬送物体を目標位置まで運ぶための位置制御を行う。

まずはじめに目標位置の入力があり、その目標位置に物体の運動を収束させるためには位置入力から物体の運動までをモデリングしなくてはならない。モデリングを行わなくてはどのような制御を施して良いのかわからないからである。

ゆえにまずはじめにモデリングを行う。

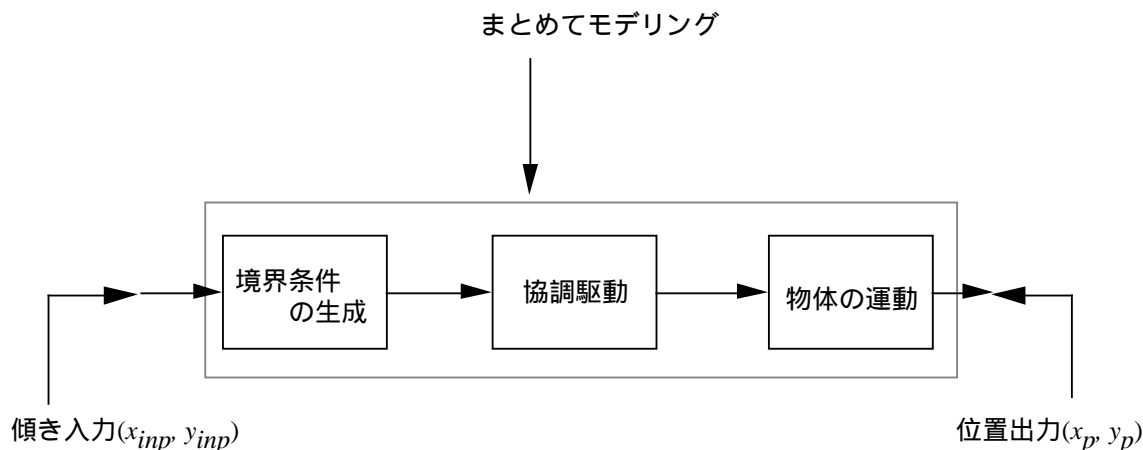


図 3.9: モデリング

図 3.9をもとにして運動方程式を線形近似を行うことにより非常におおざっぱなモデル式 (3.35) を得る。ただし、ここで p_x, p_y は定数である。

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} p_x/s^2 & 0 \\ 0 & p_x/s^2 \end{bmatrix} \begin{bmatrix} x_{inp} \\ y_{inp} \end{bmatrix} \quad (3.35)$$

ただし、

$$p_x = \frac{1 + 2(x_{\max} - x_{\min})}{2(x_{\max} - x_{\min})^2} g \quad (3.36)$$

$$p_y = \frac{1 + 2(y_{\max} - y_{\min})}{2(y_{\max} - y_{\min})^2} g \quad (3.37)$$

であり、 g は重力加速度である。

このモデル式をもとにして位置制御を行うために PD コントローラと外乱オブザーバを複合したロバストな位置制御を実現する。

3.3 その 3 - ラプラス場でフィードフォワードによる方法 -

3.3.1 拡張ラプラス場と曲面の発現

第 2 章で提議した問題を解決するために本節では拡張ラプラス場による曲面の発現による方法を提案する。

本節でのキーワードは『物体を転がして目標位置に運ぶなら、目標位置に穴を作ってやればよい』である。では、目標位置に穴をつくってやるためには各モジュールはどんな情報が必要なんだろう？各モジュールには最低、

1. 自分の座標
2. 目標位置の座標

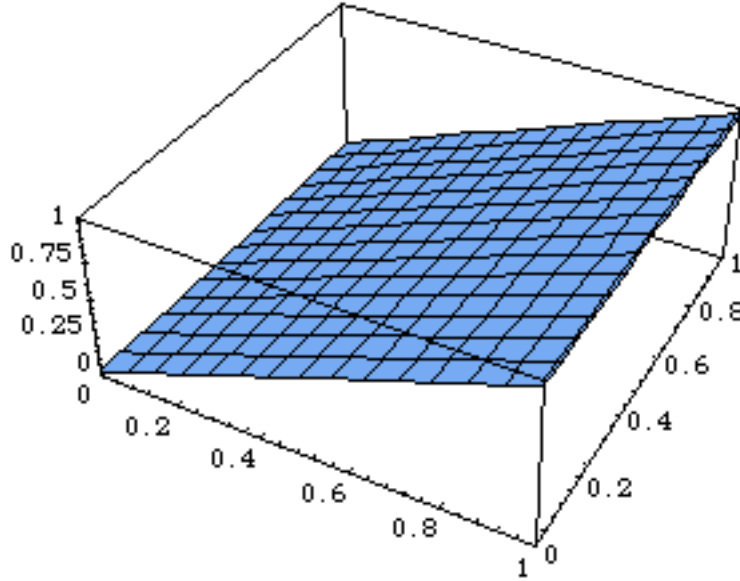
の情報が必要である。自分の座標と目標位置の座標がわかれば、各モジュールは目標位置を原点とする相対座標系を生成することができる。したがって、自分の働きに何らかの重みづけを行うことができる。目標位置から遠いものはあんまり働かなくても良いだろうということである。では、自分の座標と目標位置を知らせるための場と境界条件（集中制御系の働き）はどのようになるのだろうか。自分の位置を知らせる方法と目標位置を知らせる方法を順を追って説明する。

3.3.2 各モジュールに自分の位置を知らせる方法

はじめに各モジュールに自分の座標を設定することを考えよう。各モジュールの x 座標を設定するための場を $u_x^{cd}(x, y)$ としよう。このとき $u_x^{cd}(x, y)$ が解、

$$u^{xcd} = \frac{x_{\max} - x_{\min}}{x_{\max}}x + x_{\min}$$

$$(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}) \quad (3.38)$$



$$x_{\min} = 0, x_{\max} = 1, y_{\min} = 0, y_{\max} = 1$$

 図 3.10: u_x^{cd} の 3D プロット

を持たば、各モジュールに自分の座標を設定することができる。また y 方向も同様に、

$$u^{ycd} = \frac{y_{\max} - y_{\min}}{y_{\max}}y + y_{\min}$$

$$(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}) \quad (3.39)$$

を持たばよい。物理場の適用を考えているから何か偏微分方程式で式 (3.38),(3.39) のような解を持つ方程式を考えたい。幾つかあるだろうが代表的なものとして、

$$\frac{\partial^2 u^{xcd}}{\partial x^2} + \frac{\partial^2 u^{xcd}}{\partial y^2} = 0 \quad (3.40)$$

$$u^{xcd}(x_{\min}, y) = x_{\min}, u^{xcd}(x_{\max}, y) = x_{\max}, u^{xcd}(x, y_{\min}) = u^{xcd}(x, y_{\max}) = x \quad (3.41)$$

を用いよう。式 (3.40) は各モジュール間に適用される相互干渉を示す式であり、式 (3.41) は境界条件として集中制御系で設定される値である。このような場を適用すれば、通信の

局所性を満たしつつ、式 (3.38) を解としてもつような場を生成することができる。これにより各モジュールは自分の座標を知ることができる。

また同様に y 方向も

$$\frac{\partial^2 u^{ycd}}{\partial x^2} + \frac{\partial^2 u^{ycd}}{\partial y^2} = 0 \quad (3.42)$$

$$u^{ycd}(x, y_{\min}) = y_{\min}, u^{ycd}(x, y_{\max}) = y_{\max}, u^{ycd}(x_{\min}, y) = u^{ycd}(x_{\max}, y) = y \quad (3.43)$$

となる。これより x, y の両座標をそれぞれに場の関数として与えることができる。この値をもとにすれば、各モジュールは自分の座標を知ることができる。

具体的に式 (3.40), (3.41), (3.42), (3.43) を離散化すれば、それぞれは、式 (3.44), (3.45), (3.46), (3.46) となる。

$$u_{ij}^{xcd} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (u_{i+1,j}^{xcd} + u_{i-1,j}^{xcd}) + \frac{1}{\Delta y^2} (u_{i,j+1}^{xcd} + u_{i,j-1}^{xcd}) \right\} \quad (3.44)$$

$$u_{0,j}^{xcd} = x_{\min}, u_{i_{\max},j}^{xcd} = x_{\max}, u_{i,0}^{xcd} = u_{i,j_{\max}}^{xcd} = i\Delta x + x_{\min} \quad (3.45)$$

$$u_{ij}^{ycd} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (u_{i+1,j}^{ycd} + u_{i-1,j}^{ycd}) + \frac{1}{\Delta y^2} (u_{i,j+1}^{ycd} + u_{i,j-1}^{ycd}) \right\} \quad (3.46)$$

$$u_{i,0}^{ycd} = y_{\min}, u_{i,j_{\max}}^{ycd} = y_{\max}, u_{0,j}^{ycd} = u_{i_{\max},j}^{ycd} = j\Delta y + y_{\min} \quad (3.47)$$

ただし、

$$\Delta x = \frac{x_{\max} - x_{\min}}{m - 1} \quad (3.48)$$

$$\Delta y = \frac{y_{\max} - y_{\min}}{n - 1} \quad (3.49)$$

$$i_{\max} = m - 1 \quad (3.50)$$

$$j_{\max} = n - 1 \quad (3.51)$$

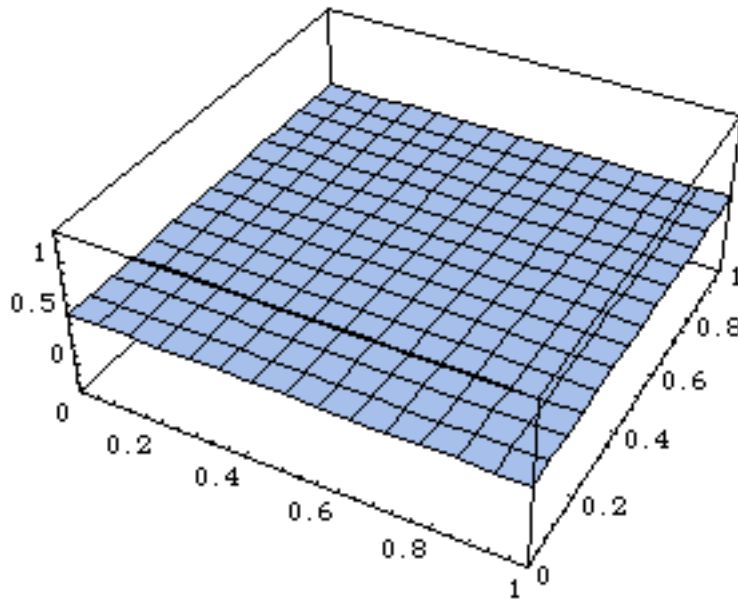
である。

3.3.3 各モジュールに目標位置を知らせる方法

つぎに各モジュールに搬送物体の目標位置を設定することを考えよう。各モジュールに搬送物体の目標位置の x 座標を設定するための場を $u^{xc}(x, y)$ としよう。このとき $u^{xc}(x, y)$ が解、

$$u^{xc} = x_c$$

$$(x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}) \quad (3.52)$$



$$x_{\min} = 0, x_{\max} = 1, y_{\min} = 0, y_{\max} = 1, x_c = 0.5$$

図 3.11: u^{xc} の 3D プロット

を持たば、各モジュールに搬送物体の目標位置を設定することができる。また y 方向も同様に、

$$u^{yc} = y_c \quad (x_{\min} \leq x \leq x_{\max}, y_{\min} \leq y \leq y_{\max}) \quad (3.53)$$

を持たばよい。物理場の適用を考えているから前節同様何か偏微分方程式で式 (3.52), (3.53) のような解を持つ方程式を考えたい。幾つかあるだろうが代表的なものとして、

$$\frac{\partial^2 u^{xc}}{\partial x^2} + \frac{\partial^2 u^{xc}}{\partial y^2} = 0 \quad (3.54)$$

$$u^{xc}(x_{\min}, y) = u^{xc}(x_{\max}, y) = x_c = u^{xc}(x, y_{\min}) = u^{xc}(x, y_{\max}) = x_c \quad (3.55)$$

を用いよう。式 (3.54) は各モジュール間に適用される相互干渉を示す式であり、式 (3.55) は境界条件として集中制御系で設定される値である。このような場を適用すれば、通信の局所性を満たしつつ、式 (3.52) を解としてもつような場を生成することができる。また、定数を各モジュールに設定するためになぜこのような複雑な偏微分方程式を用いるのだろう

と思われるかもしれないが、これは次節で説明する拡張ラプラス場として統一を行うために必要なことである。これにより各モジュールは搬送物体の目標位置を知ることができる。

また同様に y 方向も

$$\frac{\partial^2 u^{yc}}{\partial x^2} + \frac{\partial^2 u^{yc}}{\partial y^2} = 0 \quad (3.56)$$

$$u^{yc}(x, y_{\min}) = u^{yc}(x, y_{\max}) = u^{yc}(x_{\min}, y) = u^{yc}(x_{\max}, y) = y_c \quad (3.57)$$

となる。これより x, y の搬送物体の両目標座標をそれぞれに場として与えることができる。この値をもとにすれば、各モジュールは搬送物体の目標位置を正確で簡単に隣との情報交換だけで知ることができる。

具体的に式 (3.54), (3.55), (3.56), (3.57) を離散化すれば、それぞれは、式 (3.58), (3.59), (3.60), (3.60) となる。

$$u_{ij}^{xc} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (u_{i+1,j}^{xc} + u_{i-1,j}^{xc}) + \frac{1}{\Delta y^2} (u_{i,j+1}^{xc} + u_{i,j-1}^{xc}) \right\} \quad (3.58)$$

$$u_{0,j}^{xc} = u_{i_{\max},j}^{xc} = u_{i,0}^{xc} = u_{i,j_{\max}}^{xc} = x_c \quad (3.59)$$

$$u_{ij}^{yc} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (u_{i+1,j}^{yc} + u_{i-1,j}^{yc}) + \frac{1}{\Delta y^2} (u_{i,j+1}^{yc} + u_{i,j-1}^{yc}) \right\} \quad (3.60)$$

$$u_{i,0}^{yc} = u_{i,j_{\max}}^{yc} = u_{0,j}^{yc} = u_{i_{\max},j}^{yc} = y_c \quad (3.61)$$

3.3.4 拡張ラプラス場としての取り扱い

3.3.2節、3.3.3節で各モジュールの座標と搬送物体の目標位置の座標を各モジュールを知らせる場を提案した。これらは、どちらも式 (3.40), (3.54) という形で同じ 2 階の偏微分方程式のラプラス方程式で記述できる。これは、相互干渉の方法は統一にし、データのみ多重化（ベクトル化）すれば一つの統一的取り扱いが可能であることを示している。ではベクトル化した各モジュールの値をここで定義しよう。

$$\mathbf{u}(x, y) = \left[u^{xc}(x, y) \quad u^{xcd}(x, y) \quad u^{yc}(x, y) \quad u^{ycd}(x, y) \right]^T \quad (3.62)$$

とすれば、普通のラプラス方程式を拡張する形でスマートに書くことができる。

$$\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} = 0 \quad (3.63)$$

$$\mathbf{u}_{ij} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (\mathbf{u}_{i+1,j} + \mathbf{u}_{i-1,j}) + \frac{1}{\Delta y^2} (\mathbf{u}_{i,j+1} + \mathbf{u}_{i,j-1}) \right\} \quad (3.64)$$

3.3.5 曲面の発現

本節では 3.3.2、3.3.3 節の各モジュールの座標と搬送物体の目標位置の座標により、実際に曲面を作ること考える。各モジュールの座標と搬送物体の目標位置がわかれば搬送物体の目標位置を基準とした相対座標系を作ることができる。したがって何らかの重みづけをその相対座標系に基づき行うことができる。

最初に示したように搬送物体を目標位置まで運びたいのなら目標位置に穴を空ければ自然と搬送物体は目標位置に転がるだろうというアナロジーのもとに、具体的に相対座標系を用いて考える。穴を空けたければ穴が空くような関数を用いればよい。岩波の数学公式集の第 1 巻のさまざまな平面曲線が乗っている欄を調べ使えそうな関数がないか調べてみる。いかにも穴を空けるのに都合の良い関数を見つけることができた。見つけた関数は $y = x^2/(x^2 + 1)$ である。この関数をプロットすると、図 3.12 となる。このタイプの関数を用いて、各々のモジュールの z 軸方向の位置指令を作る。その作り方を式 (3.65) に、またその離散系のタイプを式 (3.66) に示す。

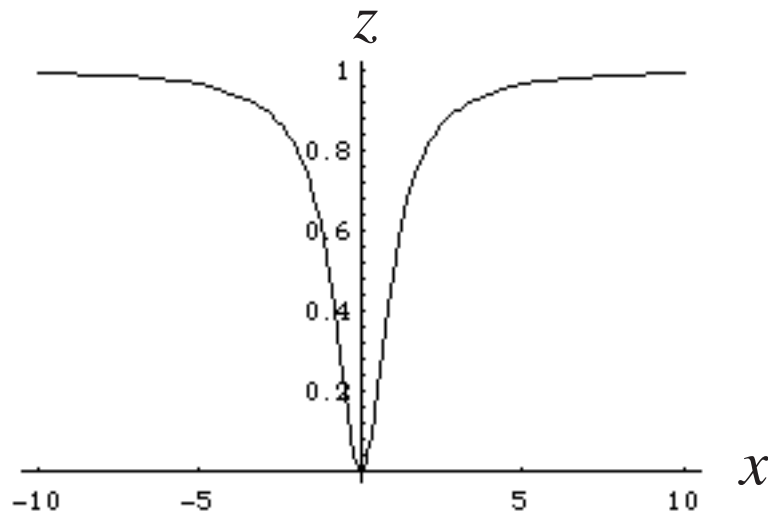


図 3.12: $y = x^2/(x^2 + 1)$ のプロット

$$z(x, y) = f(\mathbf{u}) = f(u^{xc}, u^{xcd}, u^{yc}, u^{ycd}) = \frac{\alpha \left\{ (u^{xcd} - u^{xc})^2 + (u^{ycd} - u^{yc})^2 \right\}}{\alpha \left\{ (u^{xcd} - u^{xc})^2 + (u^{ycd} - u^{yc})^2 \right\} + 1} \quad (3.65)$$

$$z_{ij} = \frac{\alpha \left\{ (u_{ij}^{xcd} - u_{ij}^{xc})^2 + (u_{ij}^{ycd} - u_{ij}^{yc})^2 \right\}}{\alpha \left\{ (u_{ij}^{xcd} - u_{ij}^{xc})^2 + (u_{ij}^{ycd} - u_{ij}^{yc})^2 \right\} + 1} \quad (3.66)$$

曲面を発現させる関数として、式 (3.65) や式 (3.66) を用いれば、搬送物体の目標位置 (x_c, y_c) に簡単に穴を空けることができる。

第 4 章

Magic Carpet のミクロスケールの設計具体例 - ソレノイドプランジャーの位置制御 -

本章では場の概念で示唆されるところのミクロスケールの制御について考える。本来ならば、第 2 章で提議している問題がマイクロマシンによるものであるためマイクロマシンのアクチュエータやプロセッサそしてセンサを用いてミクロスケールの制御について言及できればよいのだがマイクロマシンを制作する能力も装置もない。そこでリアルワールドスケールアクチュエータと題して我々が住んでいる実際の世界のサイズのアクチュエータを用いて実現を行うことにする。

リアルワールドスケールアクチュエータはマイクロマシンと異なり、非常に大きくまた起因する力も異なる。しかしながら、場の概念でいわれるところのミクロスケールには十分な性能を持っている。場の概念でいわれていることで重要なことは異なる 2 つの時間スケールが混在しているがそれらを上手く使い分けることがよいシステムの構築に必要であるということである。したがって、マクロ制御の遅い時間スケールに比べ無視できるくらい速ければ全く問題ない。

マイクロマシンはそれぞれにマイクロプロセッサやセンサを持っている自律モジュールであるということであった。マイクロプロセッサを用いればアクチュエータの位置制御を非常にミクロスケールで行うことができることはいうまでもない。

4.1 制作を行ったリアルワールドスケールアクチュエータ

本来ならマイクロマシンによるアクチュエータの制御を行うことを考えるのだが、マイクロマシンの研究を行っているわけではないし、半導体プロセス技術も持っていない。そのため、大型になってしまうが実際の我々が住んでいる数十 *cm* のスケール (リアルワールドスケール) での実現を考えることにした。

具体的に制作を行ったものを、図 4.1 に示した。一緒にタバコの箱を示したので大きさが解っていただけと思う。



図 4.1: 制作を行ったアクチュエータ

4.2 ソレノイドの電磁学的特性

まずはじめに実験装置に用いたソレノイドプランジャー (図 4.2) の電磁学的特性 [17] について考えてみよう。

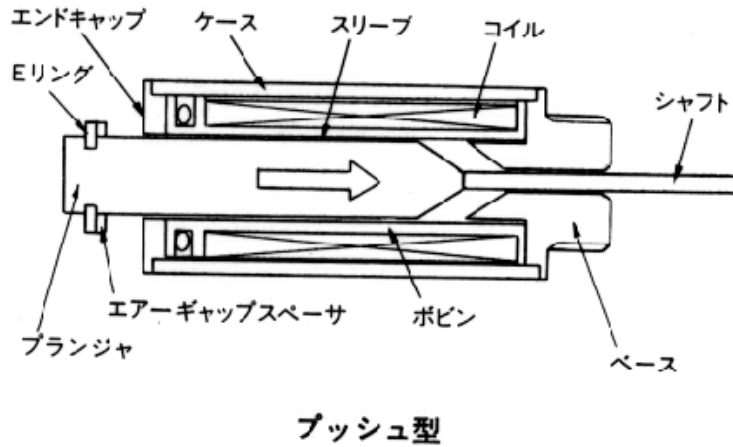


図 4.2: ソレノイドプランジャー

図 4.3のような簡単なモデルを仮定し、ソレノイドプランジャーに発生する力を導出してみる。磁束の連続性とアンペールの周回積分より、

$$\pi R^2 \mu_0 H_1 + 2\pi R \mu_0 H_2 d = 0 \quad (4.1)$$

$$H_1 x - H_2 g = Ni \quad (4.2)$$

の2本の方程式が成り立つ。ただし、ここで μ_0 は真空中の透磁率である。また、プランジャーとフレームの透磁率は真空中のそれと比べ非常に大きいと仮定した。

これよりプランジャーを鎖交する磁束 λ は、

$$\lambda = N\pi R^2 \mu_0 H_1 = \frac{2dN^2 \pi R^2 \mu_0 i}{2dx + gR} \quad (4.3)$$

であり、これより電流 i は

$$i = \frac{2dx + gR}{2d\mu_0 N^2 \pi R^2} \lambda \quad (4.4)$$

となる。したがって、磁界の蓄積エネルギー $W_m(\lambda, x)$ とプランジャーに発生する力 f_e はそれぞれ、

$$W_m = \int_0^\lambda i(\tilde{\lambda}, x) d\tilde{\lambda} = \frac{2dx + gR}{4d\mu_0 N^2 \pi R^2} \lambda^2 \quad (4.5)$$

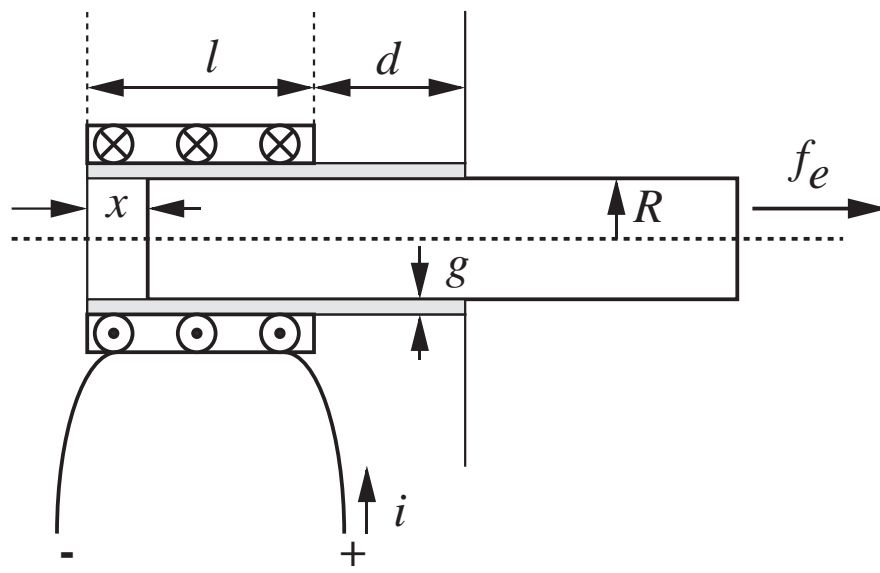


図 4.3: ソレノイドプランジャのモデル

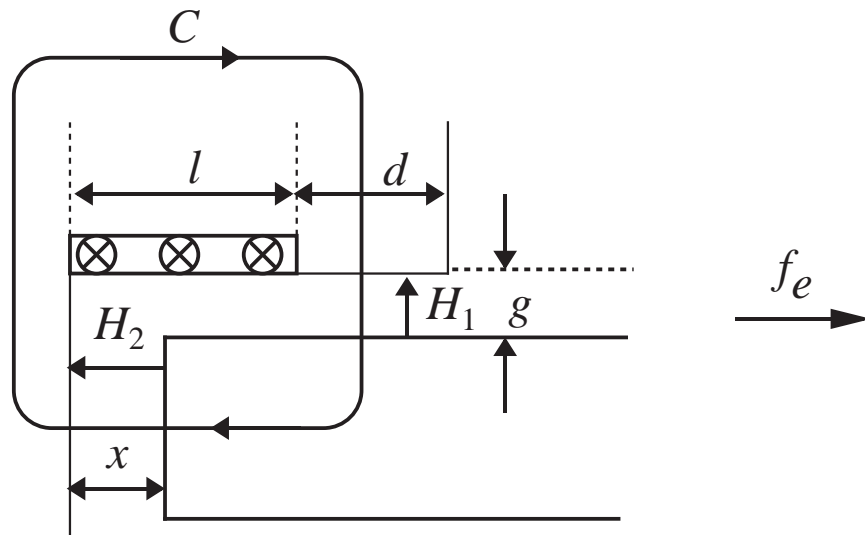


図 4.4: アンペールの周回積分の積分路の取り方

$$f_e = -\frac{\partial W_m}{\partial x} = -\frac{2\mu_0 N^2 \pi R^2}{(2dx + gR)^2} i^2 \quad (4.6)$$

である。式 (4.6) からわかるように発生する電磁気力 f_e は位置 x の 2 乗に逆比例し、電流 i の 2 乗に比例する。すなわち電磁気力 $f_e = f_e(x^{-2}, i^2)$ である。

それでは実際に発生する力を測定してそれぞれのパラメータを決定しよう。発生する電磁気力は $f_e = f_e(x^{-2}, i^2)$ であることが電磁学的な解析によりわかったから、近似関数を x^{-1}, x^{-2}, i, i^2 の多項式としよう。

$$f_e = ax^{-1} + bx^{-2} + ci + di^2 \quad (4.7)$$

これに対して、発生する電磁気力を測定し最小 2 乗法により各パラメータ a, b, c, d を決定すると以下ようになる。

$$f_e = 2.95 \times 10^{-3} x^{-1} + 1.29 \times 10^{-6} x^2 + 0.94i + 0.89 \times 10^{-2} i^2 \quad (4.8)$$

となる。

式 (4.8) をもとにして位置制御を行うコントローラ的设计を行う。

4.3 ロバストな位置制御

ロバストな位置制御を行うためにコントローラとして PD コントローラと外乱オブザーバを複合し用いることにする。図 4.5 のようなブロックのコントローラを設計することにする。

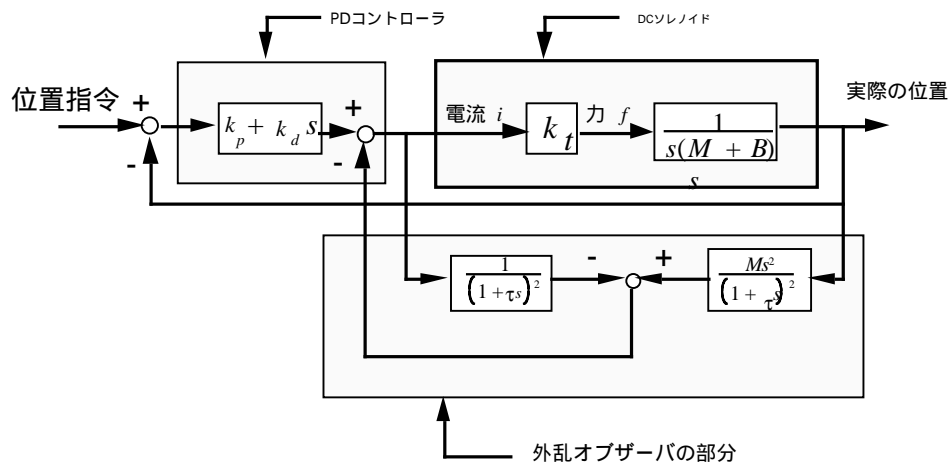


図 4.5: ロバストな位置制御を行うためのコントローラ

この位置制御は場の概念でいうところのミクロスケールで進行することになるので非常に速い応答を期待したい。そこで位置制御を実現するためにアナログコントローラ [18] を組むことにした。その回路図を図 4.6 に示す。

また、現在制作を予定している実験装置に実際組み込むためには、非常にたくさんの数を作らなくてはならないのでできるだけ簡単でなおかつ単価が安く性能が良くななくてはならない。そこで以下に示す P 制御回路 (図 4.7) についても同様に実験を行った。

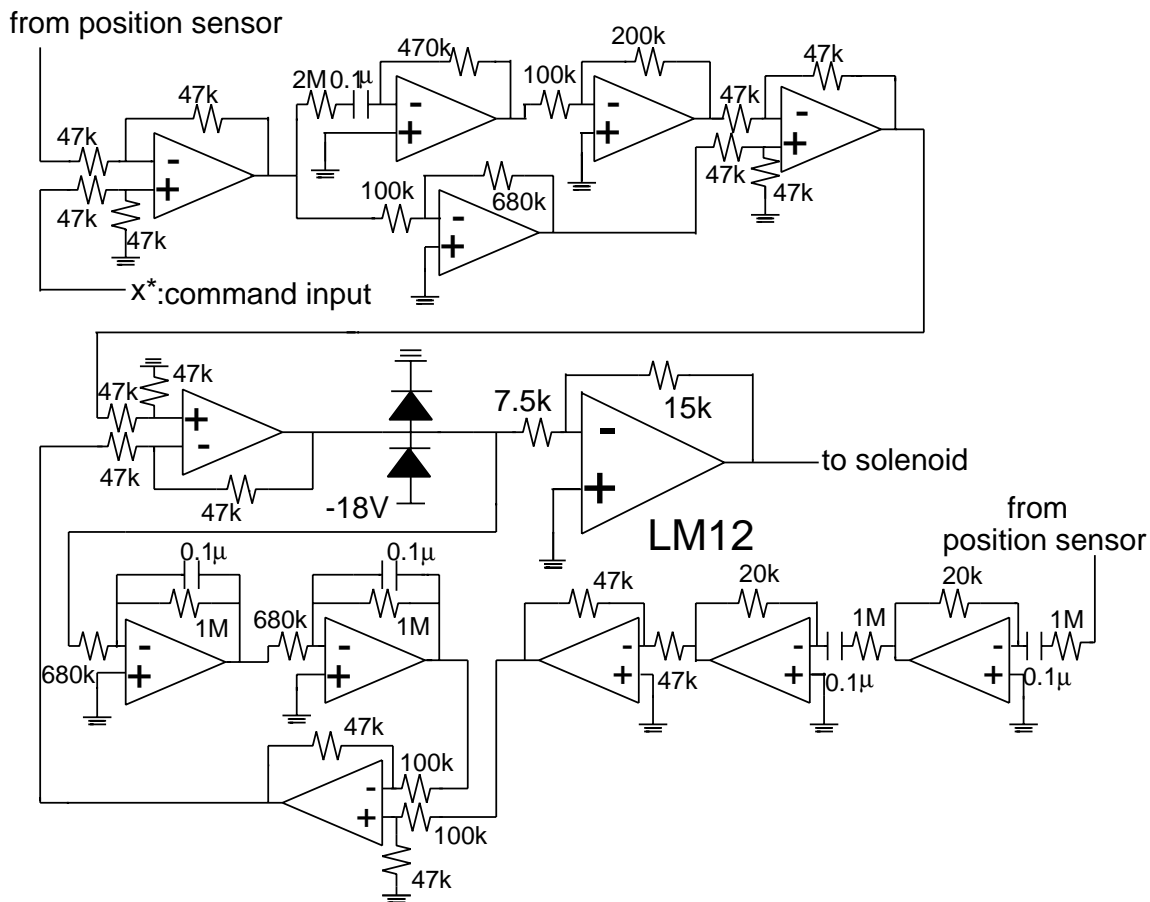


図 4.6: PD コントローラと外乱オブザーバを併用したタイプ

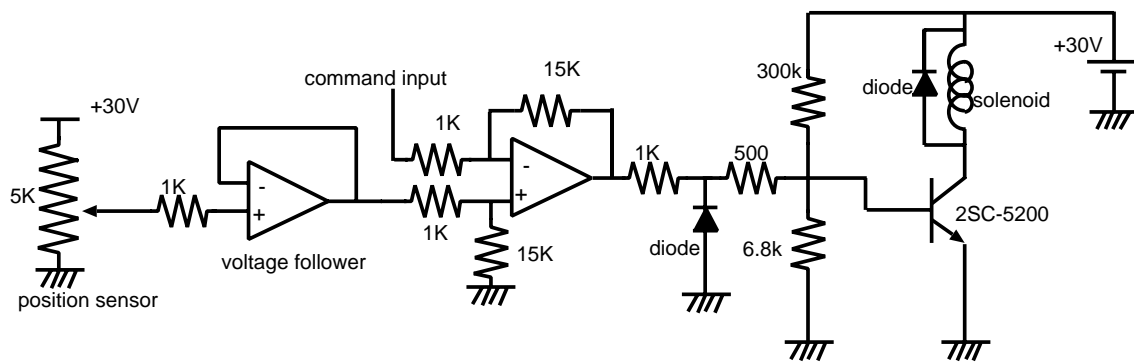


図 4.7: P コントローラのための回路

第 5 章

シミュレーション結果と実験結果

5.1 マクロスケール (シミュレーション結果)

5.1.1 振動方程式による方法 - 失敗例 -

シミュレーション条件

シミュレーションは1次元のモジュールが4個並んでいる場合について行うことにした。本来ならば2次元平面に並べられたモジュール群でシミュレーションをすべきであるが、このシミュレーションは完全な制御を目指しているためまずは少なく簡単なものから試すことにした。図 5.1 にモデル図を示した。このモデルに対しては可制御が完全に成り立っていることを注意しておく。

このモデルに対して以下でトラッキングレギュレータ、最適サーボ、マルチレートサンプリング制御の方法を適用しその結果を示す。

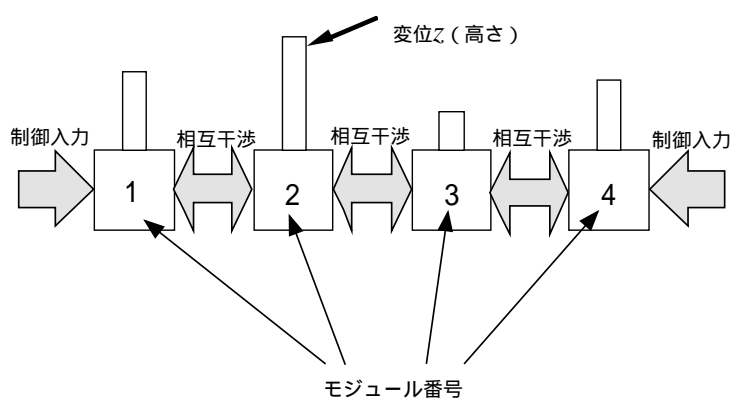


図 5.1: 振動方程式による方法のモデル図

トラッキングレギュレータによる方法

3.1.1節に示したトラッキングレギュレータによる各モジュールの完全な制御の結果を示す。それぞれのモジュールに対して位置指令と位置応答を示した。ここで、モジュール番号は、振動方程式による方法のモデル図 5.1を参考にして欲しい。

結果を見ていただければわかるが、全く制御できていない様子が良くわかる。比較的少ないモジュールで 1 次元の場合でも制御が極めて難しいことがこの結果よりわかる。したがって 2 次元のたくさんのモジュールの場合はよりいっそう制御自体は難しくなるわけであるから、この方法での制御はできないという結論を導くことができる。

制御に用いたパラメータを表 5.1に示す。図 3.1の F は極配置法に決定したが極の配置の仕方は Butterworth の極配置 [20] を用い、左半面に放射状に配置を行った。その大きさを表 5.1に各パラメータ同様示す。

表 5.1: トラッキングレギュレータに用いた各パラメータ

Δx	0.1
c	10
k	1
極の大きさ	80

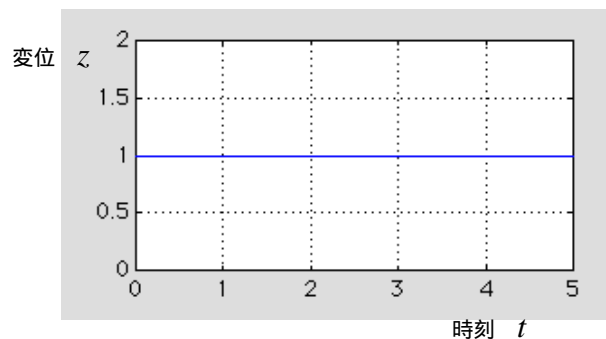


図 5.2-a: モジュール 1 の位置指令

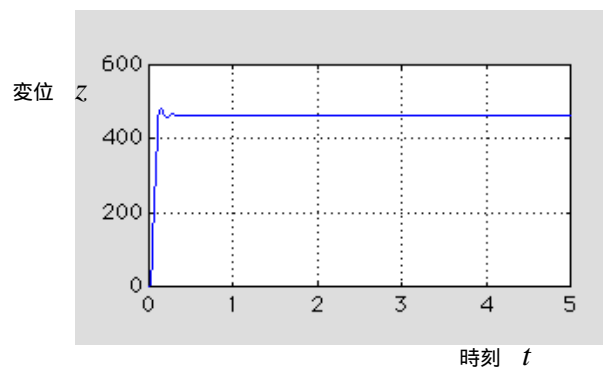


図 5.2-b: モジュール 1 の位置応答

図 5.2: 相互干渉に振動方程式を適用しトラッキングレギュレータで制御を施したときのモジュール 1 の応答

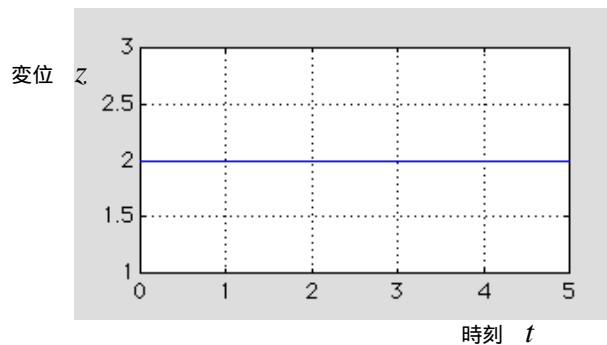


図 5.3-a: モジュール 2 の位置指令

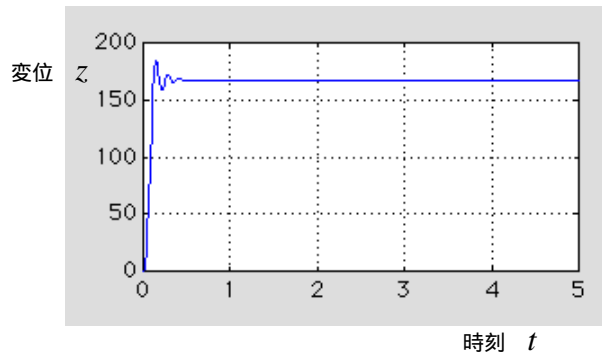


図 5.3-b: モジュール 2 の位置応答

図 5.3: 相互干渉に振動方程式を適用しトラッキングレギュレータで制御を施したときのモジュール 2 の応答

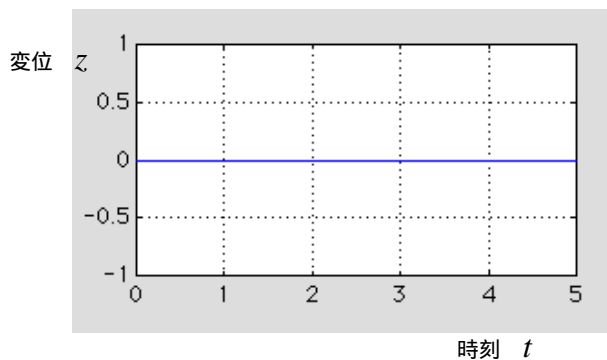


図 5.4-a: モジュール 3 の位置指令

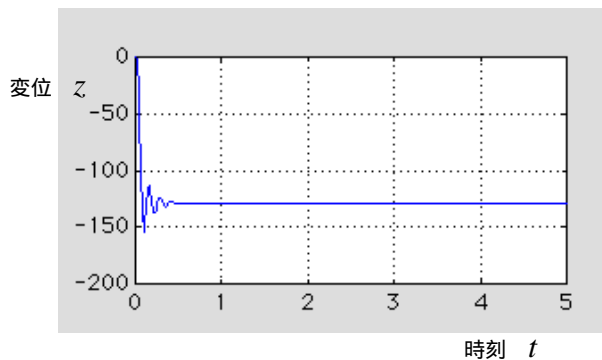


図 5.4-b: モジュール 3 の位置応答

図 5.4: 相互干渉に振動方程式を適用しトラッキングレギュレータで制御を施したときのモジュール 3 の応答

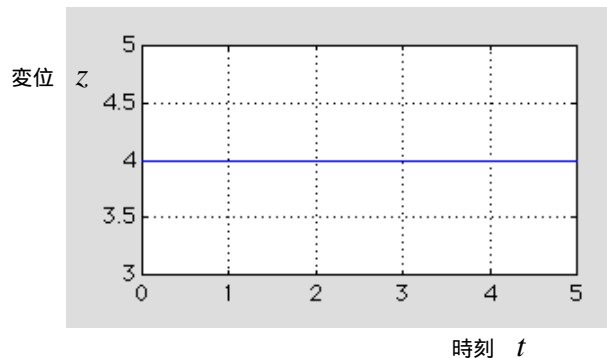


図 5.5-a: モジュール 4 の位置指令

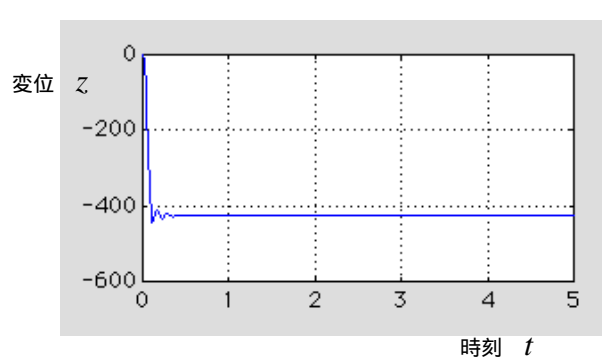


図 5.5-b: モジュール 4 の位置応答

図 5.5: 相互干渉に振動方程式を適用しトラッキングレギュレータで制御を施したときのモジュール 4 の応答

最適サーボ制御による方法

3.1.2節に示した最適サーボ制御による各モジュールの完全な制御の結果を示す。それぞれのモジュールに対して位置指令と位置応答を示した。ここで、モジュール番号は、振動方程式による方法のモデル図 5.1を参考にして欲しい。

トラッキングレギュレータ同様、結果を見ていただければわかるが、全く制御できていない様子が良くわかる。比較的少ないモジュールで 1 次元の場合でも制御が極めて難しいことがこの結果よりわかる。したがって 2 次元のたくさんのモジュールの場合はよりいっそう制御自体は難しくなるわけであるから、この方法での制御はできないという結論を導くことができる。

シミュレーションで用いたパラメータを表 5.2に示す。また最適サーボ制御を行うための重み行列 R, Q はそれぞれ単位行列を用いた。

表 5.2: 最適サーボのシミュレーションに用いた各パラメータ

Δx	0.1
c	10
k	1

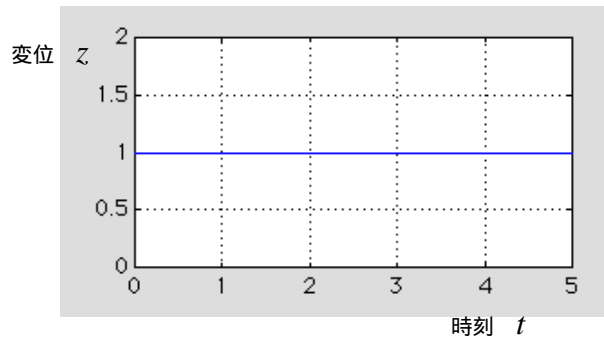


図 5.6-a: モジュール 1 の位置指令

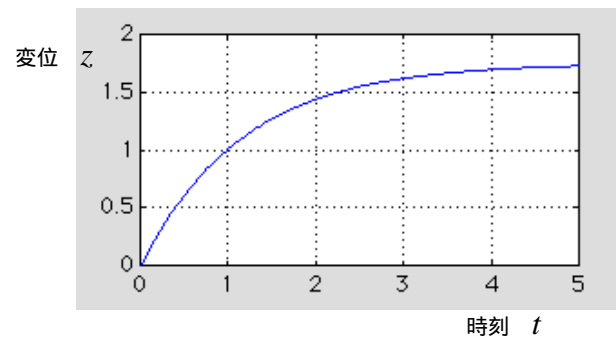


図 5.6-b: モジュール 1 の位置応答

図 5.6: 相互干渉に振動方程式を適用し最適サーボ制御を施したときのモジュール 1 の応答

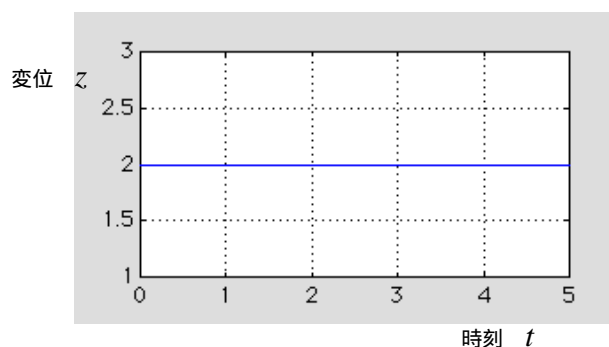


図 5.7-a: モジュール 2 の位置指令

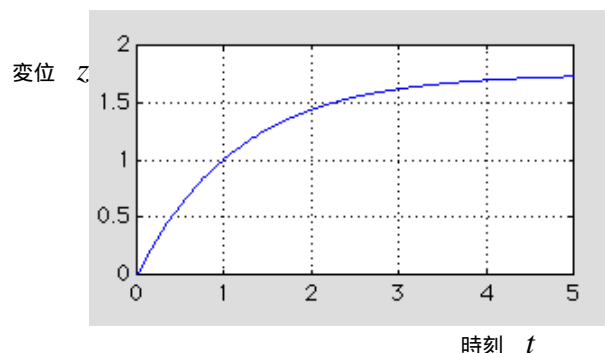


図 5.7-b: モジュール 2 の位置応答

図 5.7: 相互干渉に振動方程式を適用し最適サーボ制御を施したときのモジュール 2 の応答

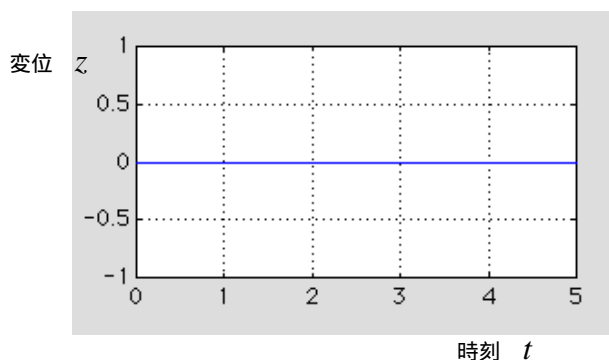


図 5.8-a: モジュール 3 の位置指令

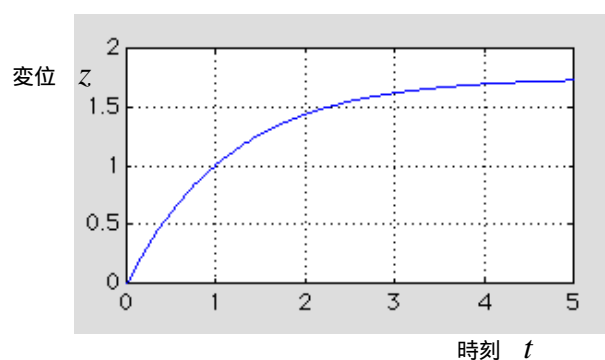


図 5.8-b: モジュール 3 の位置応答

図 5.8: 相互干渉に振動方程式を適用し最適サーボ制御を施したときのモジュール 3 の応答

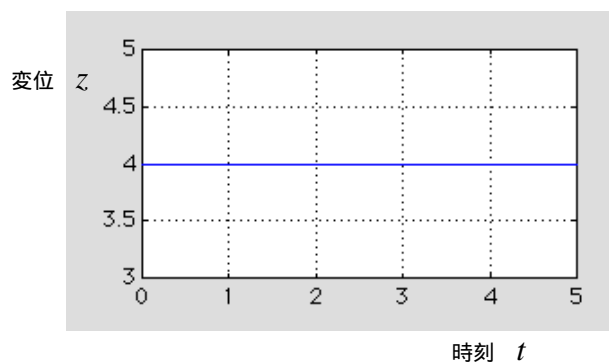


図 5.9-a: モジュール 4 の位置指令

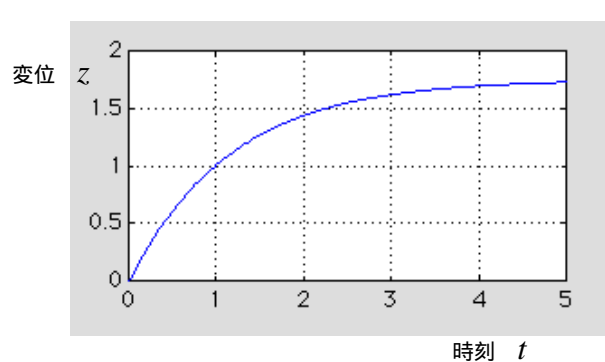


図 5.9-b: モジュール 4 の位置応答

図 5.9: 相互干渉に振動方程式を適用し最適サーボ制御を施したときのモジュール 4 の応答

マルチレートサンプリング制御による方法

3.1.3節に示した最適サーボ制御による各モジュールの完全な制御の結果を示す。それぞれのモジュールに対して位置指令と位置応答を示した。ここで、モジュール番号は、振動方程式による方法のモデル図 5.1を参考にして欲しい。

トラッキングレギュレータ、最適サーボ制御同様に結果を見ていただければわかるが、全く制御できていない様子が良くわかる。比較的少ないモジュールで1次元の場合でも制御が極めて難しいことがこの結果よりわかる。したがって2次元のたくさんのモジュールの場合はよりいっそう制御自体は難しくなるわけであるから、この方法での制御はできないという結論を導くことができる。

ただし、トラッキングレギュレータや最適サーボ制御と同様に制御が上手く行っていないことは事実であるが、トラッキングレギュレータや最適サーボ制御と異なる点はサンプリング点上では制御ができているということである。

シミュレーションで用いたパラメータを表 5.3に示す。

表 5.3: マルチレートサンプリング制御のシミュレーションに用いた各パラメータ

Δx	0.1
c	10
k	1
サンプリングタイム	0.5[s]

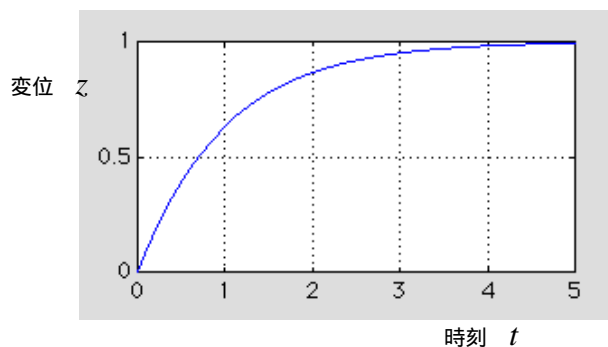


図 5.10-a: モジュール 1 の位置指令

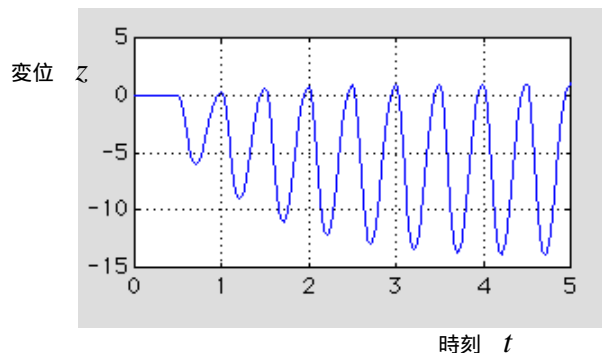


図 5.10-b: モジュール 1 の位置応答

図 5.10: 相互干渉に振動方程式を適用しマルチレートサンプリング制御を施したときのモジュール 1 の応答

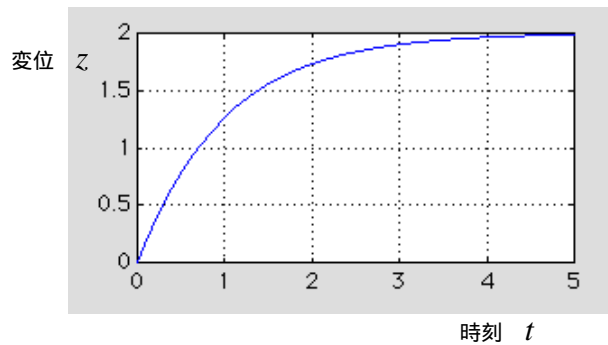


図 5.11-a: モジュール 2 の位置指令

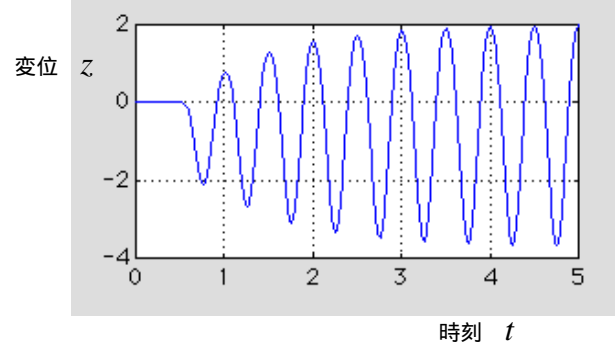


図 5.11-b: モジュール 2 の位置応答

図 5.11: 相互干渉に振動方程式を適用しマルチレートサンプリング制御を施したときのモジュール 2 の応答

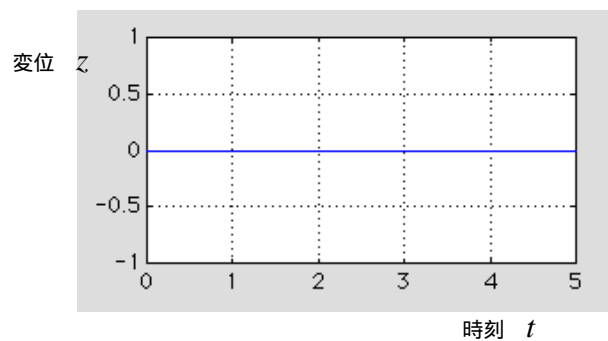


図 5.12-a: モジュール 3 の位置指令

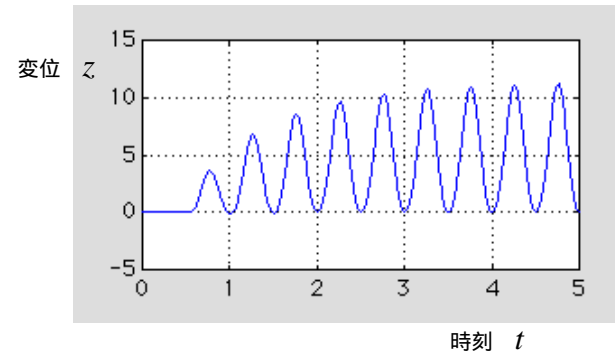


図 5.12-b: モジュール 3 の位置応答

図 5.12: 相互干渉に振動方程式を適用しマルチレートサンプリング制御を施したときのモジュール 3 の応答

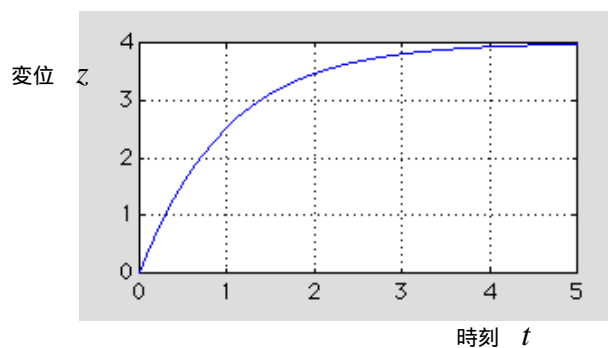


図 5.13-a: モジュール 4 の位置指令

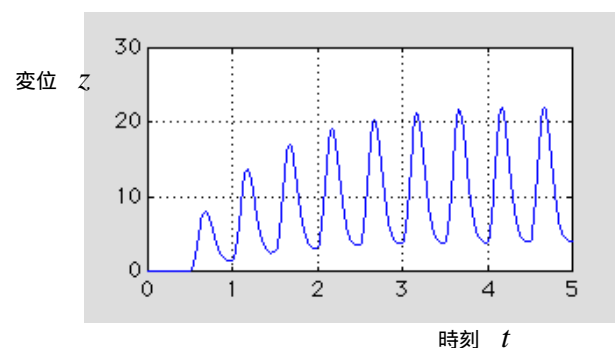


図 5.13-b: モジュール 4 の位置応答

図 5.13: 相互干渉に振動方程式を適用しマルチレートサンプリング制御を施したときのモジュール 4 の応答

表 5.4: ラプラス場でフィードバックのシミュレーションに用いたパラメータ

Δx	0.1
Δy	0.1
x 方向のモジュールの並び	11
y 方向のモジュールの並び	11
(x_{\min}, y_{\min})	(0,0)
(x_{\max}, y_{\max})	(1,1)
(x_{ini}, y_{ini})	(0.6,0.6)
(x_c, y_c)	(0.4,0.4)

5.1.2 ラプラス場でフィードバックによる方法

ラプラス場でフィードバックによる方法のシミュレーション結果を以下に示す。シミュレーションに用いたパラメータを表 5.1.2 に示す。またシミュレーションのイメージをつかんでいただくために図 5.14にモジュールの並びの図を示す。

図 5.15に時間応答を示した。搬送物体の位置がオーバーシュートしながらも目標値に収束しているのがわかる。また、推定位置に誤差が生じているため現在位置の収束値が目標値から微妙にずれていることがわかる。図 5.16-a、図 5.16-b、図 5.16-cに Magic Carpet の挙動の様子を示すために表面曲面を代表的な時刻で示してみた。図 5.16-aはシミュレーションをはじめてすぐの状態であり、偏差が大きいため大きな勾配が生じているのがわかる。また図 5.16-bはオーバーシュートを起こしているために物体を戻すためにシミュレーション初期のころと反対の勾配が生じていることがわかる。図 5.16-cは最終的に収束しているところである。現在位置と目標位置が重なっていることがわかる。

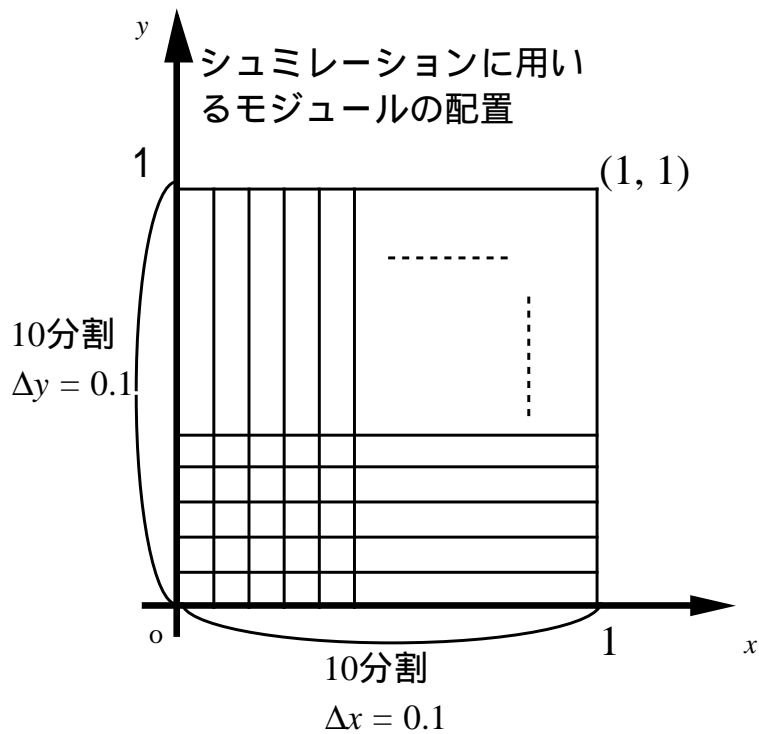


図 5.14: モジュールの並び

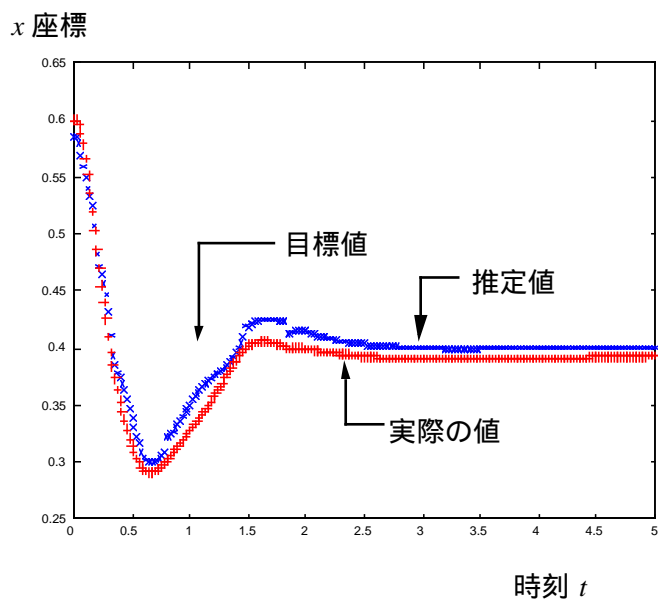


図 5.15: 搬送物体の x 座標の時間応答

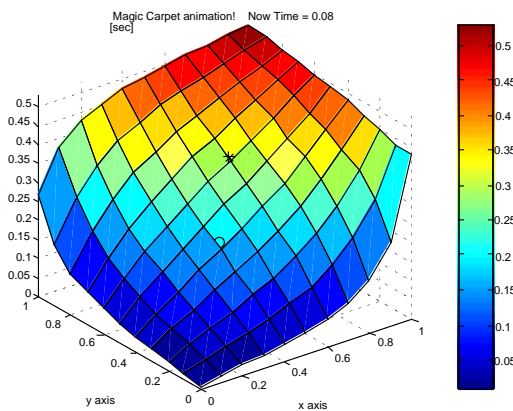


図 5.16-a: 0.08 秒のときの表面曲面

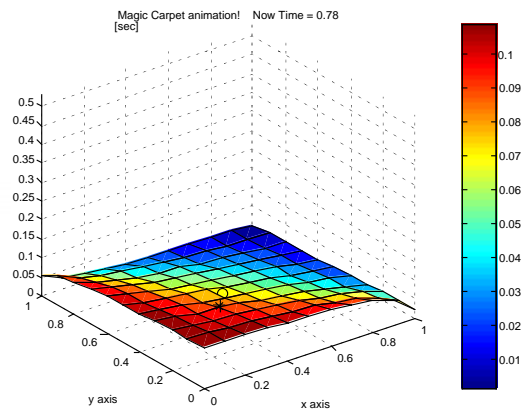


図 5.16-b: 0.78 秒のときの表面曲面

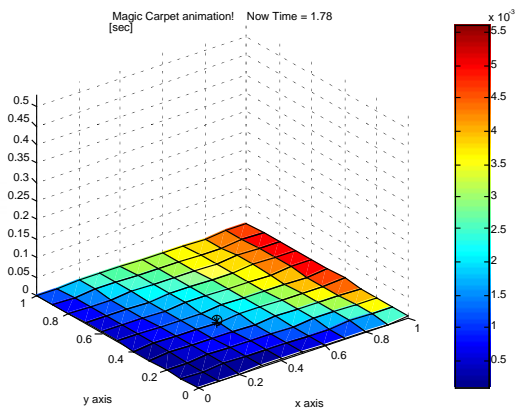


図 5.16-c: 1.78 秒のときの表面曲面

図 5.16: ラプラス場でフィードバックの場合の表面曲面

表 5.5: ラプラス場でフィードフォワードのシミュレーションに用いたパラメータ

Δx	0.1
Δy	0.1
x 方向のモジュールの並び	11
y 方向のモジュールの並び	11
(x_{\min}, y_{\min})	(0,0)
(x_{\max}, y_{\max})	(1,1)
(x_{ini}, y_{ini})	(0.6,0.6)
(x_c, y_c)	(0.4,0.4)
α	5

5.1.3 ラプラス場でフィードフォワードによる方法

ラプラス場でフィードフォワードによる方法のシミュレーション結果を以下に示す。シミュレーションに用いたパラメータは表 5.1.3 に示す。またシミュレーションのイメージをつかんでいただくために図 5.17 にモジュールの並びの図を示す。

図 5.18 に時間応答を示した。これにより搬送対象物の位置が目標値に振動しながら収束しているのがわかる。このシミュレーションでは制定にほぼ 3 秒ほどかかっているがこれは搬送物により大きく異なる。図 5.19-a には 0.21[s] での曲面を示した。搬送対象物が目標値に向かって動いているのがわかる。また、目標値を極小値とする曲面が描けていることも同様にわかる。図 5.19-b には 3.15[s] での曲面を示した。図 5.19-b では完全に目標値に収束していることがわかる。

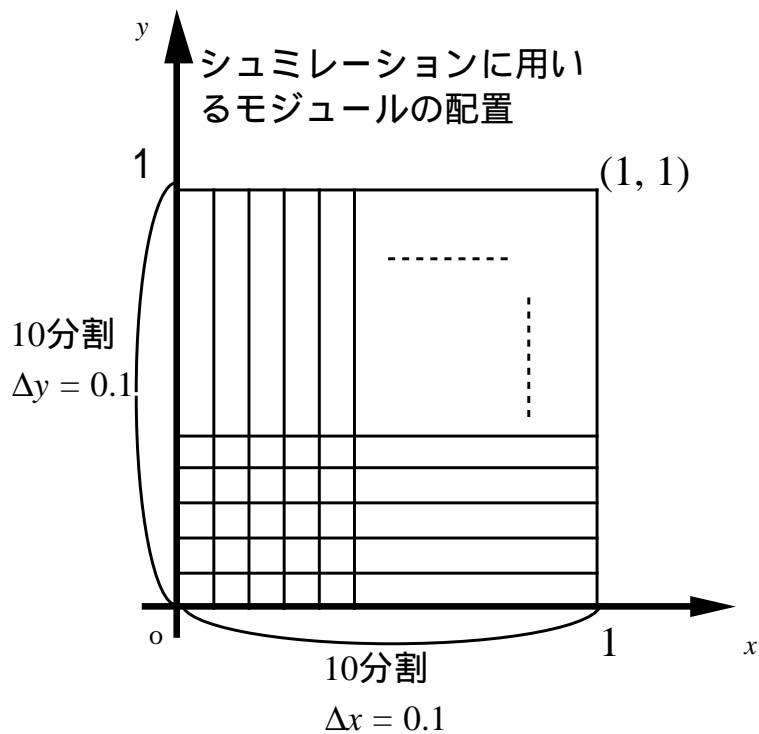


図 5.17: モジュールの並び

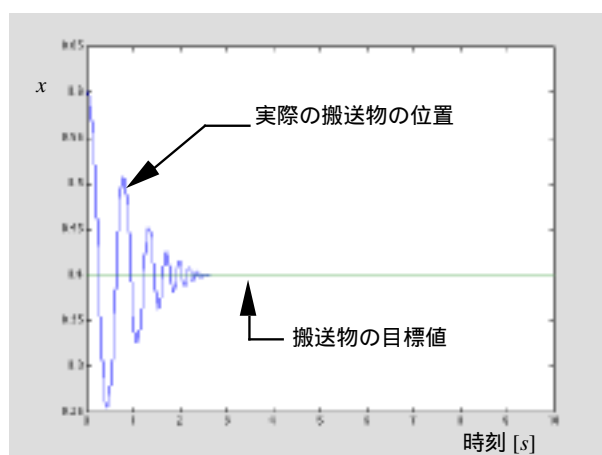


図 5.18: 搬送物体の x 座標の時間応答

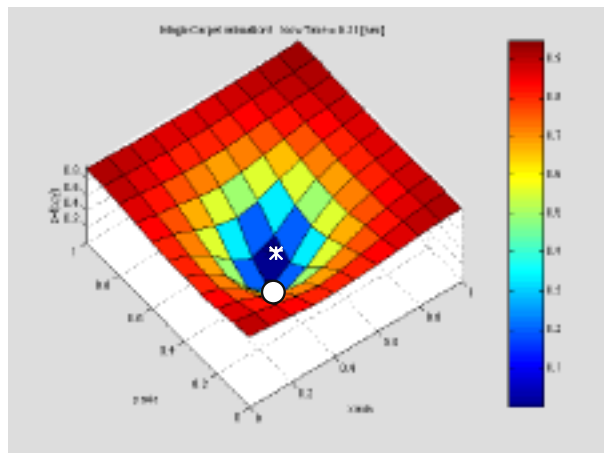


図 5.19-a: 0.21 秒のときの表面曲面

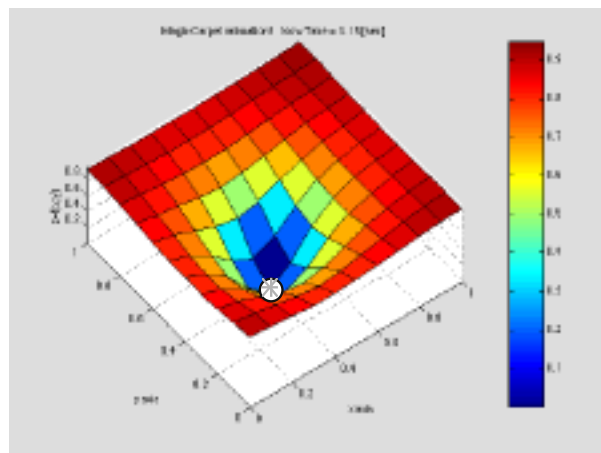


図 5.19-b: 3.15 秒のときの表面曲面

図 5.19: ラプラス場でフィードフォワードの場合の表面曲面

5.2 ミクロスケール（実験結果）

以下にミクロスケールでの秩序性を考えるために第 4 章で提案した方法の実験結果を示す。ロバストな位置制御を実現するために PD コントローラと外乱オブザーバの併用した結果を図 5.20 に示す。非常に良好な結果を得ている。図 5.20-a にそれぞれ位置指令と位置応答を重ねずに示したが、完全追従が行えてることをよくわかっていただくために図 5.20-b にその二つの波形を重ねたものも用意した。非常に良好な結果を得た。

次に実験装置の制作を考えれば非常に簡単でなおかつ性能のよいものが望まれることはいうまでもない。そこで第 4 章提案したのが P コントローラのみの方である。この位置指令と位置応答を図 5.21 に示す。これからもわかって頂けると思うがこれも同様に良好な結果を得た。図 5.21-a にそれぞれ位置指令と位置応答を重ねずに示したが、完全追従が行えていることをわかっていただくために図 5.21-b にその二つを重ねたものを PD コントローラと外乱オブザーバを併用したものの同様に示した。

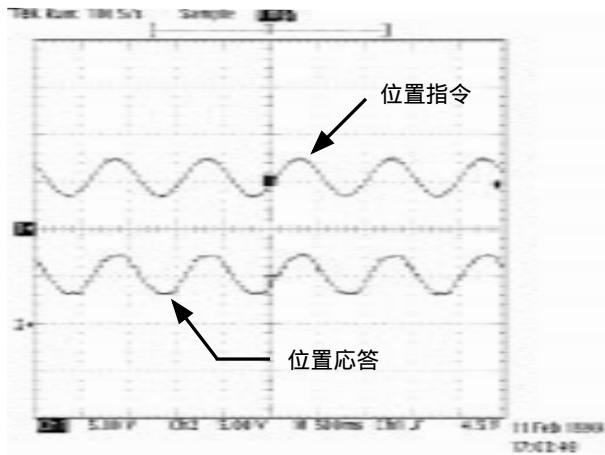


図 5.20-a: 位置指令と位置応答をばらばらにプロットした場合

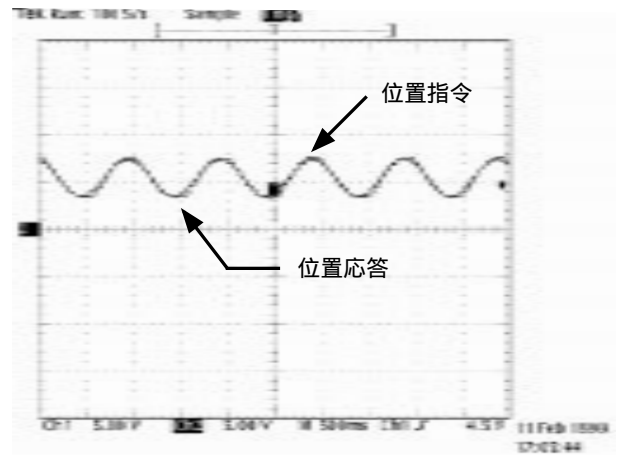


図 5.20-b: 位置指令と位置応答を重ねてプロットした場合

図 5.20: PD コントローラと外乱オブザーバを併用した場合の位置応答

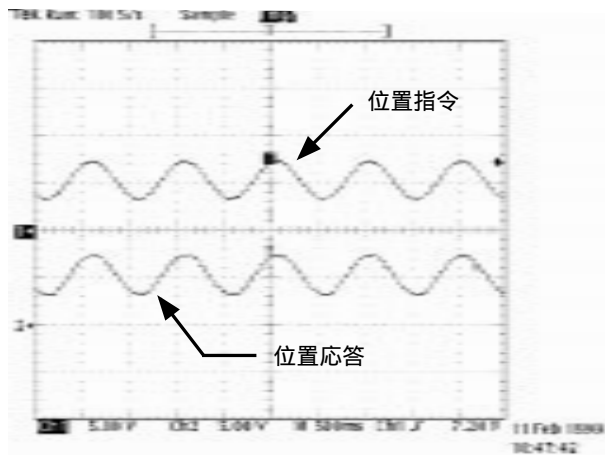


図 5.21-a: 位置指令と位置応答をばらばらにプロットした場合

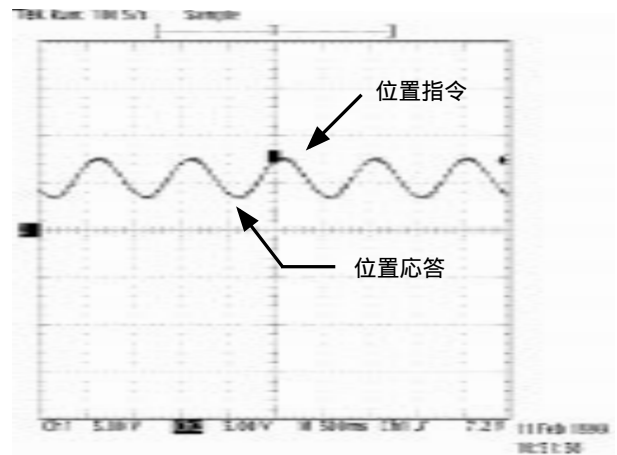


図 5.21-b: 位置指令と位置応答を重ねてプロットした場合

図 5.21: 量産型を目指し簡単な P コントローラのみの場合

第 6 章

一般的な自律分散システムに関する考察 － Magic Carpet の設計から得られた知見より－

6.1 自律分散集中複合システム

自律分散システムのシステム形態は一般にどのようなものが考えられるのだろうか？

今まで Magic Carpet のためのシステム形態を主として考えてきたが自律分散システム一般的なものの一つとして、Magic Carpet のためのシステム形態を考えることはできないだろうか。Magic Carpet のためのシステム形態は分散モジュール群を制御するための集中制御系を置き、必ずしも自律分散システムの形態といいがたいし、階層構造を持っている。しかしながら人とのつながりを考えた場合このような集中制御系をおくことは多くのメリットがある。図 6.1 をここで再掲するが、人との間に一つシステムが入ることにより人とのやり取りは集中制御系が行うという極めてシンプルな形態になっていることがわかる。

6.2 マクロスケールのためのシステムデザイン - 一般的な線形物理場の適用 -

マクロなスケールの制御としてさまざまな方法を提案した。行ったことを表 6.1 で示すと、となる。すべての場合について研究をやったわけでもないし、また限られた場についてのみ研究をおこなったという感が強い。しかしながら実際にさまざまな場をもちいることにより一般的な線形物理場の適用を考えることができるようになった。

各モジュールは各々勝手に振る舞って良いわけであるが、何らかの形で協調していなくては物体搬送の目的は達成できないし、集中制御系からの情報も得ることができないことは先に述べた通りである。

『場の満たす条件』

1. 集中制御系とのやりとりが簡単で明確

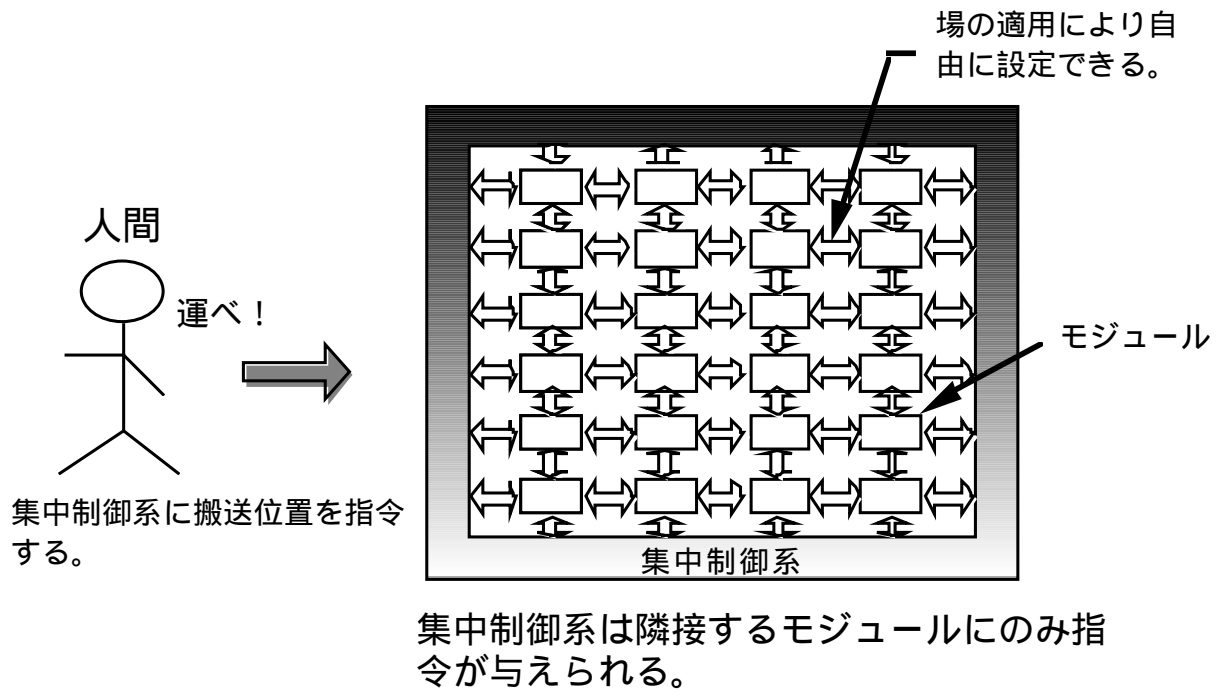


図 6.1: 自律分散集中複合系のイメージ図 (再掲)

表 6.1: 集中制御系の負担による場の分類

場の種類		集中系の負担 (場の概念の非適用度)		
		大	中	小
線形 物理場	静的		3.2にて	3.3にて
	動的	3.1にて		

自律分散制御が目指すところ

2. 通信の局所性を満足

である。そこでその場として線形物理場の適用を提案する。

線形な物理場 [21] の適用を提案する理由について以下で説明する。2 階の偏微分方程式で説明することにしよう。その方程式を

$$a \frac{\partial^2 q}{\partial x^2} + b \frac{\partial^2 q}{\partial y^2} + c \frac{\partial^2 q}{\partial t^2} + d \frac{\partial q}{\partial x} + e \frac{\partial q}{\partial y} + f \frac{\partial q}{\partial t} = 0 \quad (6.1)$$

と書こう。この方程式を空間方向のみ $\Delta x, \Delta y$ で離散化し、 $q_{ij} = q(i\Delta x, j\Delta y)$ で書こう。そうすると、式 (6.1) は、

$$a \frac{q_{i+1,j} + 2q_{ij} - q_{i-1,j}}{\Delta x^2} + b \frac{q_{i,j+1} + 2q_{ij} - q_{i,j-1}}{\Delta y^2} + c \frac{d^2 q_{ij}}{dt^2} + d \frac{q_{i+1,j} - q_{i-1,j}}{2\Delta x} + e \frac{q_{i,j+1} - q_{i,j-1}}{2\Delta y} + f \frac{dq_{ij}}{dt} = 0 \quad (6.2)$$

と書ける。ここで、 x, y 方向の対称性を崩さないように、2 階の空間偏微分は前進、後退差分を複合し、1 階の空間偏微分は両側差分を行った。これを等価式変形を行えば、

$$\frac{d^2 q_{ij}}{dt^2} = \alpha_1 q_{i+1,j} + \alpha_2 q_{i-1,j} + \alpha_3 q_{i,j+1} + \alpha_4 q_{i,j-1} + \alpha_5 q_{i,j} + \alpha_6 \frac{dq_{ij}}{dt} \quad (6.3)$$

となる。ただし、ここで

$$\alpha_1 = -\alpha_2 = -\frac{2a + d\Delta x}{c} \Delta y^2 \quad (6.4)$$

$$\alpha_3 = -\alpha_4 = -\frac{2b + e\Delta x}{c} \Delta y^2 \quad (6.5)$$

$$\alpha_5 = \frac{4}{c} (a\Delta y^2 + b\Delta x^2) \quad (6.6)$$

$$\alpha_6 = -\frac{f}{c} \quad (6.7)$$

である。式 (6.3) は状態方程式での記述の可能性を示している。これは適用した場が線形物理場であるから成り立つものであり、非線形な場ならばそれはできない。

すなわち、図 6.2 のようになる。

ローカルな形での記述を行ったがこれを整理して記述すれば、状態方程式 (式 (6.8)) の形で記述できることが容易に想像できる。すなわち場全てを記述することができる。これはひとえに線形物理場の適用を行ったからである。全てを整理して記述できるため集中制御系での制御が明確な形で制御工学の道具を用いて設計できるわけである。

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{aligned} \quad (6.8)$$

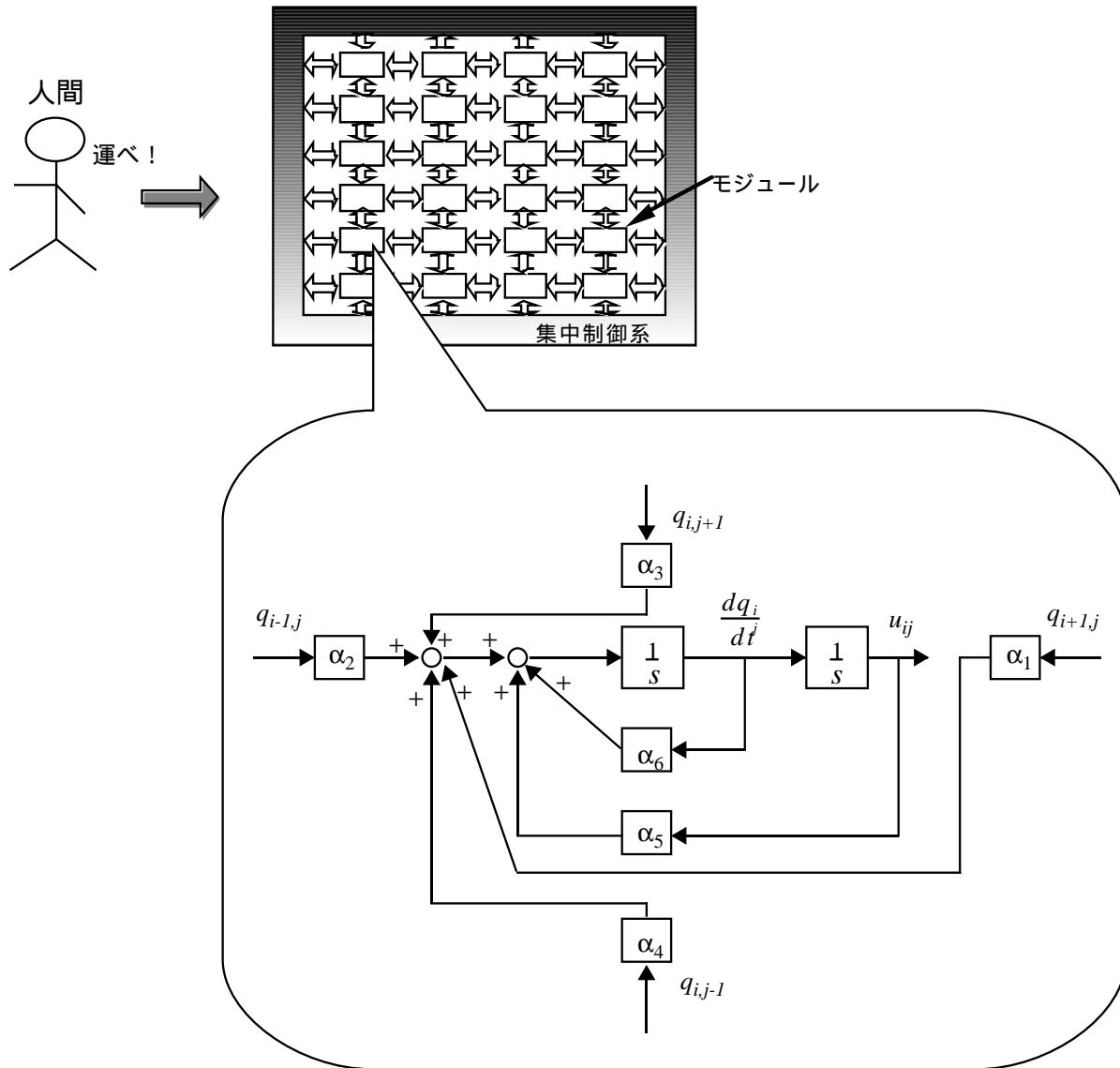


図 6.2: ローカルな形で記述した場合の図、ただし s はラプラス演算子

式 (6.8) において y, u はそれぞれ、 u_{ij} を並べたもの、境界からの入力であり、 A, B, C, D はそれぞれ場の特性によって決まるシステム行列である。 (A, B) により可制御性の議論ができることも非常に意味のあることである。

以上で線形で一般的な場の適用が可能になった。あとは場の設計論について明確に触れることができれば問題ない。これは今後の課題として結言の中でふれることにする。

第 7 章

結言

本論文ではマイクロマシンに代表される自律分散システムを考え、具体例としてマイクロマシンによる自律分散型物体搬送システム Magic Carpet の設計を目指した。

Magic Carpet の設計に当たり主として問題となったことは、

- 小型化に伴う通信路の確保の難しさ（隣接要素のみに通信できる）
- 制作上の問題によりソフト、ハードとも同質

である。

まずはじめに場の概念を適用しモジュール全体のマクロとモジュール単体のミクロの二つの秩序性を考えた。マクロな制御として物理的場の適用を提案した。具体的には振動方程式による方法、ラプラス場でフィードバックによる方法、ラプラス場でフィードフォワードによる方法を提案しシミュレーションによりその有効性を確かめた。また、ミクロな制御としてソレノイドプランジャのロバストな位置制御を行うために PD コントローラと外乱オブザーバの併用と P コントローラによる方法を考え、実験によりその有効性を確かめた。

マクロな制御として振動方程式による方法から得られた知見として、

- 線形でダイナミクスを持つ振動方程式の適用により各モジュールの内部状態量の状態方程式の記述が可能になった。
- 一般に自律分散システムにしばしば存在する入力の数に比べ制御したい量が多い場合、状態方程式での記述が可能になってもその制御は簡単ではない。トラッキングレギュレータ制御、最適サーボ制御、マルチサンプリング制御の 3 つの代表的な手法を用いたがその制御は失敗に終わった。

が挙げられる。

また、ラプラス場でフィードバックによる方法から得られた知見としては、

- 分散的手法により位置推定を行うことにより搬送物体の位置に関するフィードバックループを作ることができた。
- 搬送物体の位置に関するフィードバックにより PD コントローラと外乱オブザーバを組み込むことにより搬送物体のロバストな位置制御が可能になった。

が挙げられる。

最後にラプラス場でフィードフォワードによる方法から得られた知見としては、

- 物体を運びたいところに穴を開ければそこに転がるというシンプルな考えのもとで物体搬送を完了させることができた。
- ラプラス場では表面に穴を開けることができないので場により決定される量を内部状態量としそれから非線形な関数を用いて曲面を発現させることにより、さまざまな曲面が作れるようになった。
- 場の設計を行うに当たり、必要な解を考えその解を持つような場と境界条件を用いることにより場の設計論について多少の示唆を与えることができた。

が挙げられる。

つぎにミクロな制御として、リアルワールドスケールでの Magic Carpet の実現を考えソレノイドプランジャーの位置制御を行った。高速な応答が期待したかったのでアナログ回路による位置のロバストな位置制御を実現した。これによりソレノイドプランジャーは非常に非線形性が強いにもかかわらずロバストな位置制御が可能になった。

問題点としては、マクロな制御が実験によりその有効性を検証することができなかったことが一つに挙げられる。これに関しては現在製作中の実験装置を早期に完成させることにより解決することができると考えている。つぎの問題点として、場の設計論について踏み込むことができなかったことが挙げられる。どのような場でマクロな制御を実現したらよいか全くわからず現状では適用する場によりその集中制御系からの制御の方法を替えなくてはならず、全くもって一般的な手法とはいいいがたい。場としては振動方程式とラプラス方程式の 2 つを用いたがどちらもケーススタディー的要素が非常に強かった。

謝辞

本研究をすすめるにあたり、多くの方々からさまざまなご指導・ご助言をいただきました。堀洋一助教授には、ご多忙にもかかわらずさまざまな視点からの適切なるご指導をいただきました。物わかりの悪くなかなか人のいうことを聞かない私に対して懇切丁寧に指導していただきました。ここに深く感謝いたします。

東京大学生産技術研究所の藤田博之教授にはマイクロマシンの専門家としてさまざまなご意見を頂きました。また自律分散システム・シンポジウム参加の機会を与えて頂きました。ここに深く感謝いたします。

全瑩煥助手に深く感謝いたします。制御理論に関して全く無知だった私に対して丁寧に制御理論のゼミをやっていただきました。

実験装置の制作から研究室の環境整備などさまざまな点で内田利之技術官にはお世話になりました。ここに深く感謝いたします。

博士課程の坂井真一郎さん、鈴木文泰さん、藤本博志さんには研究のことはもちろんのこと、年齢が近かったこともあり私にとってお兄さんのような存在であり本当にお世話になりました。ありがとうございます。

小田井正樹君には学年が同じだったこともあり、一緒に勉強したり、お酒を飲んだり研究室での生活全般においてお世話になりました。彼の存在が多くの安心感を与えてくれました。ありがとうございます。

オンラインセンターの田崎良三助手、西村芳治技術官、山崎一彦技術官には研究室の計算機充実や研究室行事などの面で大変お世話になりました。心より感謝いたします。

呉世訓君、佐渡秀夫君、関弘和君そして卒論生の皆様とともに充実した研究室生活を送れたことに対して感謝いたします。特に北島博晃君とは同じ研究分野に取り組んでいたため、同じ実験部屋で多くのたのしい時間を過ごしました。ありがとうございます。

宇佐美貴代さんには同い年ということもありさまざまな点でお世話になりました。研究が上手く行かず苦しんでいるとき宇佐美さんの笑顔に救われたことを思い出します。心より感謝申し上げます。

最後になりましたが、他にもたくさんの方々にお世話になりました。心より深く感謝申し上げます。

発表文献

- [1] 及部七郎齋、堀洋一：場の概念を用いた自律分散型 magic carpet の提案と設計、電気学会産業応用部門全国大会講演論文集 [III], pp.379-382,1998
- [2] 及部七郎齋、堀洋一：自律分散型 magic carpet の提案と設計、東京大学工学部附属総合試験所年報,Vol.57, pp.221-226,1998
- [3] 及部七郎齋、堀洋一：場の概念を用いた自律分散型 magic carpet の提案と設計手法の分類、第 11 回自律分散システムシンポジウム,pp.267-270,1999

参考文献

- [1] Dragoslav D.Siljak:Large-scale dynamic systems:stability and structure,Elsevier North-Holland(1978)
- [2] 伊藤正美:自律分散システム研究の課題と将来、計測自動制御学会誌, Vol.32, pp.789-796, Oct. 1993
- [3] 新誠一、池田建司、湯浅秀男、藤田博之:自律分散システム、朝倉書店 (1995)
- [4] 村田智:均質ユニットによる機械のかたちづくり、日本ロボット学会誌,Vol.15,No.5,pp.675-679,1997
- [5] 根本泰弘:『マイクロマシン』発刊に際して、日本機械学会誌、Vol.97,No.905,pp.252,1994
- [6] K.J.GABRIEL:アメリカにおける MEMS の研究プロジェクト、日本機械学会誌、Vol.97,No.905,pp.272-275,1994
- [7] 藤田博之:マイクロマシンと自律分散、計測自動制御学会誌, Vol.32, pp.848-853, Oct. 1993
- [8] 藤田博之:自律分散マイクロシステム、日本機械学会誌、Vol.97,No.905,pp.298-301,1994
- [9] 三田吉郎、Andreas Kaiser,Patrick Garda,Maurice Milgram,藤田博之:形状認識を行なう分散型マイクロ搬送システム、第11回自律分散システムシンポジウム,pp.259-262,1999
- [10] 深尾毅:分散システム論、昭晃堂 (1987)
- [11] 石田好輝:自律分散システムの情報処理の特徴、計測自動制御学会誌, Vol.32, pp.830-836, Oct. 1993
- [12] 高原光:振動・波動論入門、現代工学社 (1996)
- [13] 平井一正、羽根田博正、北村新三:システム制御工学、森北出版 (1980)

- [14] 武者利光、岡本良夫:逆問題とその解き方、オーム社 (1992)
- [15] 吉田勝久、佐藤昭:基礎システム理論、コロナ社 (1978)
- [16] 美多勉、原辰次、近藤良:基礎ディジタル制御、コロナ社 (1988)
- [17] 浅野和俊:電気ー力学システムの基礎と応用、養賢堂 (1994)
- [18] 藤井信生:アナログ電子回路ー集積回路化時代のー、昭晃堂 (1984)
- [19] 藤本博志、河村篤男、堀洋一:一般化マルチレートサンプリング制御を用いた高性能完全追従制御法の提案と零点配置法との比較、東京大学工学部附属総合試験所年報, Vol.57, pp.207-214,1998
- [20] 小郷寛、美多勉:システム制御理論入門、実教出版 (1979)
- [21] 大西公平:分散アクチュエータのシステムデザイン、日本機械学会誌、Vol.97, No.905, pp.290-291, 1994
- [22] V.D. バーガー、M.G. オルソン:力学ー新しい視点にたってー、培風館 (1975)
- [23] 加川幸雄:開領域問題のための有限/境界要素法、サイエンス社 (1983)
- [24] 神谷紀生:有限要素法と境界要素法、サイエンス社 (1982)
- [25] 藤井信生:ディジタル電子回路ー集積回路化時代のー、昭晃堂 (1987)
- [26] 伊東規之:ディジタル回路、日本理工出版会 (1986)

付録 A

付録

A.1 自律分散条件

『自律分散システムは、システム全体を統合する管理機構をもたず、システムを構成する各要素がシステムの目的及び環境、他の要素の挙動を認識し、それに基づいて自分の行動を自律的に決定することにより、要素間の協調を図り、全体として大域的秩序を形成または維持するシステムである。』と一般的に定義されている。この定義をかみ砕き基本的な要件に分解してみよう。

(a) システムを構成する個（要素）は多数あり、空間的に分散している（システムの分散性）。

(b) 各個は機能的に代替可能という意味で同質である（個の同質性）。

(c) 各個は自ら主体的に行動する（個の自律性、自発性）。

(d) システム及びそれを構成する個はエネルギー代謝のある開放系で、常に活性化されている（ダイナミクス性）。

(e) 各個はシステム全体の大域的秩序に関する情報を持ち、多数の個の協調（結合、相互作用）によってシステム全体の大域的秩序が形成または維持される（システムの秩序形成）。また、そこには個間相互作用の場が存在する（相互作用の場）。その場合の個間の情報のやりとりは局所的である（通信の局所性）。

(f) 個間の相互作用（個と個の結合）はあらかじめ設定されるのではなく、周囲の状況に応じて随時変化する（相互作用の非決定性、自己組織化）。

(g) 目的や環境に変化があると、自らの構造（相互作用）を変化させて適応する（環境適応性）。

A.2 物体の運動の模擬

実際に協調駆動されたアクチュエータ上で物体の運動はどのようになるのだろうか？
以下で物体の運動を簡単なモデルを仮定し、物体の運動方程式 [22] を導出する。

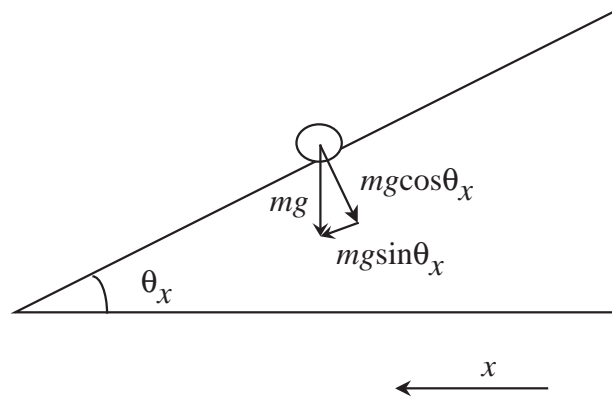


図 A.1: 物体の運動モデル

図 A.1 のようなモデルを考えると、運動方程式は以下の様になる。

$$\begin{cases} m \frac{d^2 x}{dt^2} = mg \sin \theta_x - \mu mg \cos \theta_x \\ m \frac{d^2 y}{dt^2} = mg \cos \theta_y - \mu mg \cos \theta_y \end{cases} \quad (\text{A.1})$$

ただし、ここで m, g, μ はそれぞれ物体の質量、重力加速度、動摩擦係数であり、 y 方向も同様に定式化を行うことにより、2 元の連立微分方程式を得た。

(A.1) からわかるようにこの運動方程式 A.1 を解くためには任意の場所での直角三角形で近似した場合の角度 (任意の場所での微分係数) が必要になる。モジュールは空間的に離散的に配置されているわけではあるが、非常にたくさんあることを仮定しているのでこの間を補間することにより角度を得ることができる。以下にその方法を示す。

四角形の面積座標系を用いることにより一次関数で内部の値を補間することにしよう。

$$u = \sum_{i=1}^4 L_i u_i \quad (\text{A.2})$$

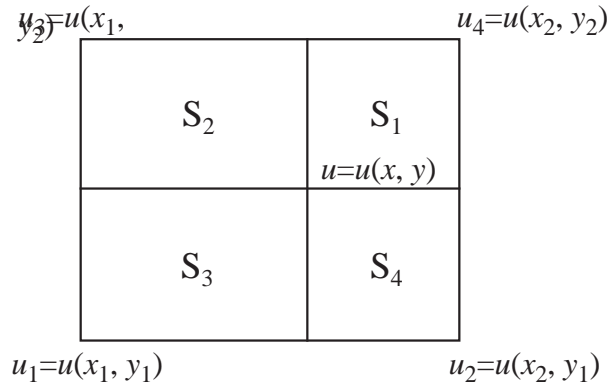


図 A.2: 四角形の面積座標

ただし、 S_i ($i=1,2,3,4$) はその領域の面積

ただし、

$$L_i = \frac{S_i}{S_1 + S_2 + S_3 + S_4} \quad (i = 1, 2, 3, 4) \quad (\text{A.3})$$

すなわち、

$$u = \frac{(x_2 - x)(y_2 - y)u_1 + (x - x_1)(y_2 - y)u_2 + (x - x_1)(y - y_1)u_3 + (x_2 - x)(y - y_1)u_4}{S_1 + S_2 + S_3 + S_4} \quad (\text{A.4})$$

$$\frac{\partial u}{\partial x} = \frac{(y_4 - y)(u_2 - u_1) + (y - y_1)(u_3 - u_4)}{S_1 + S_2 + S_3 + S_4} \quad (\text{A.5})$$

$$\frac{\partial u}{\partial y} = \frac{(x_2 - x)(u_4 - u_1) + (x - x_1)(u_3 - u_2)}{S_1 + S_2 + S_3 + S_4} \quad (\text{A.6})$$

となり、

$$\begin{aligned} \theta_x &= \tan^{-1}\left(\frac{\partial u}{\partial x}\right) \\ &= \frac{(y_4 - y)(u_2 - u_1) + (y - y_1)(u_3 - u_4)}{S_1 + S_2 + S_3 + S_4} \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \theta_y &= \tan^{-1}\left(\frac{\partial u}{\partial y}\right) \\ &= \frac{(x_2 - x)(u_4 - u_1) + (x - x_1)(u_3 - u_2)}{S_1 + S_2 + S_3 + S_4} \end{aligned} \quad (\text{A.8})$$

を得ることができる。以上得られた式を用いて任意の場所で運動方程式を解けば物体の運動を模擬できる。

A.3 修正版 位置推定

従来版の位置推定の方法では差分法を用いて閉領域問題として取り扱っていたためにマイクロマシンモジュール群の並べ方によって位置推定の精度が変わってしまっていた。それはマイクロマシンの並んでいる端の部分が境界となりその内部の閉領域で問題を解いていたためである。

ボールが任意の場所にあるときの位置推定誤差曲面図を図 A.3に示す。

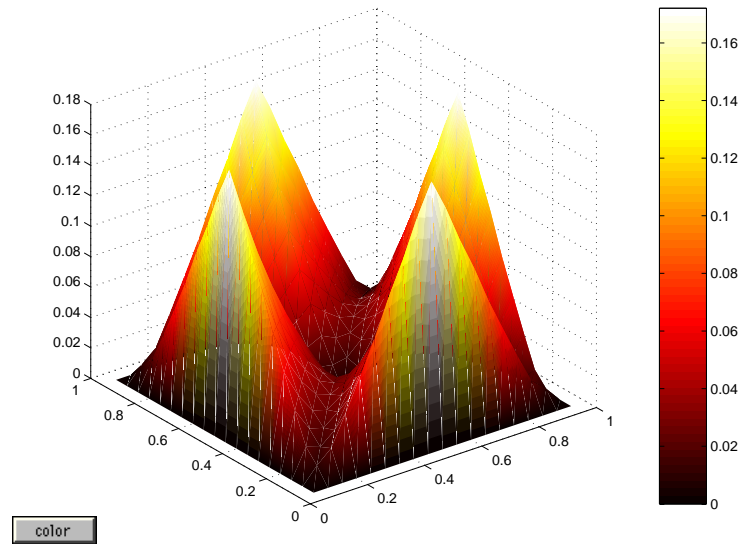


図 A.3: 位置推定誤差曲面図 $z = [(x_p - x_p^{est})^2 + (y_p - y_p^{est})^2]^{1/2}$

それを修正するために境界を無限遠にする半開領域問題 [23] として定式化を行うことにする。従来同様マイクロマシン側では差分法により定式化されたルールにより計算することにし、集中制御系で境界要素法 [24] により疑似的に無限遠に境界があるかのように振る舞うことにする。したがって、マイクロマシン側からはいかにも無限遠までマイクロマシンがぎっちりならんでいるかのような錯覚をさせるアルゴリズムである。

図 A.4の領域に対して、式 (A.9) を解くための正式化が本節の目的である。

$$\begin{cases} \nabla^2 u = f(x, y) & \text{in } \Omega_1 \\ \nabla^2 u = 0 & \text{in } \Omega_2 \end{cases} \quad (\text{A.9})$$

ただし、

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (\text{A.10})$$

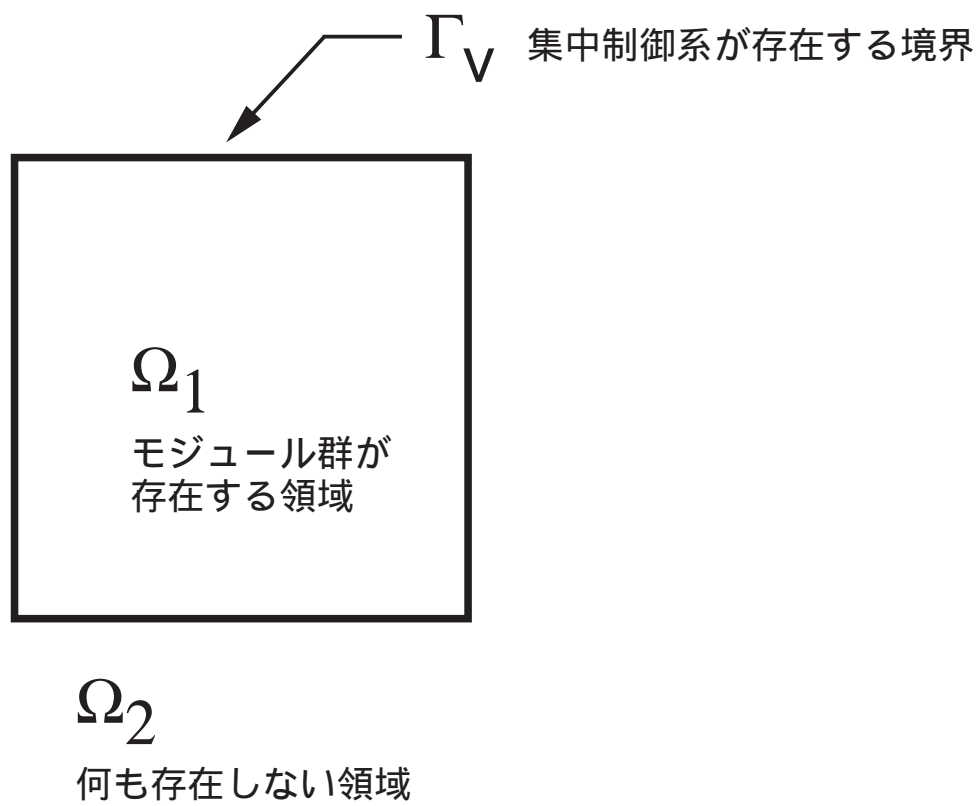


図 A.4: 領域定義図

境界要素法で半開領域問題を解くとき、対象の方程式が楕円型の偏微分方程式の場合は無限遠に境界条件 ($u = 0$) が疑似的に付加される。そのことを利用し半開領域問題を解くことにする。

さて、この問題を解くために領域内部 Ω_1 では差分法 (FDM)、領域外部 Ω_2 では境界要素法 (BEM) を複合する。

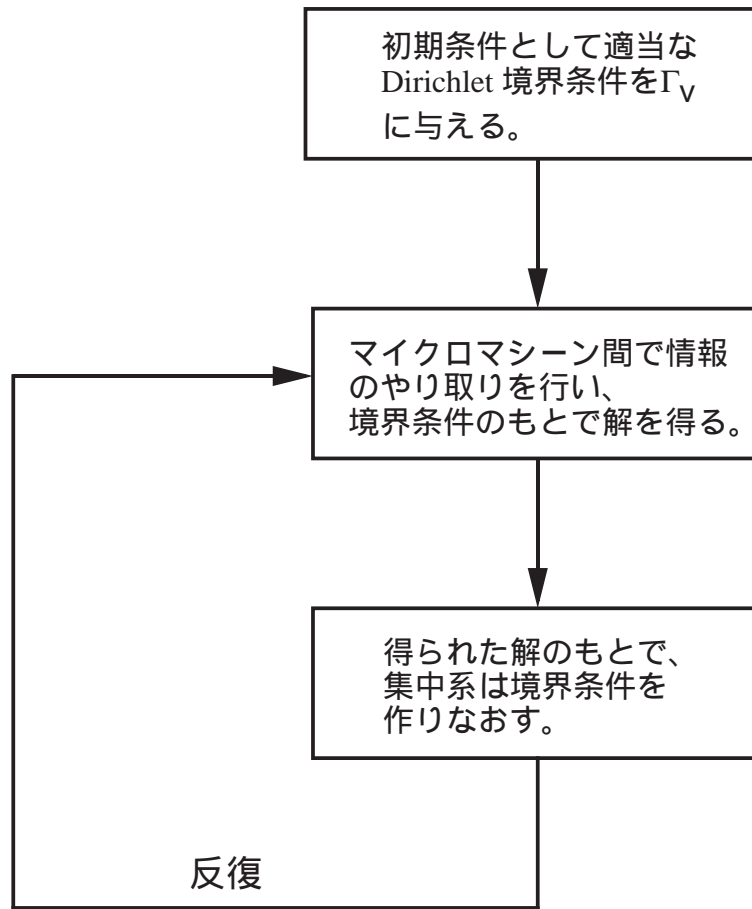


図 A.5: 複合アルゴリズム

複合アルゴリズムを図 A.5に示した。

まず、はじめに領域内部 Ω_1 で式 (A.9) を解くために集中制御系側で適当な境界条件を設定する。その条件をもとにして、マイクロマシン間で式 (A.9) を離散化した式 (A.11) を解く。正確に言えば解くというより情報の交換により収束させるという表現の方が適切である。それにより得られた結果をもとにして集中系では式 (A.12) を解く。集中系で解かれた式 (A.12) の結果をもとにして領域内部のマイクロマシン群で式 (A.9) を解く。その繰り返しにより、無限遠に境界条件 $u = 0$ を設定したような解を得ることができる。

$$u_{ij} = \left[2 \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1} \left\{ \frac{1}{\Delta x^2} (u_{i+1,j} + u_{i-1,j}) + \frac{1}{\Delta y^2} (u_{i,j+1} + u_{i,j-1}) - f_{ij} \right\} \quad (\text{A.11})$$

ただし、ここで $u_{ij} = u(i\Delta x, j\Delta y)$ 、 $f_{ij} = f(i\Delta x, j\Delta y)$ であり、 Δx 、 Δy はそれぞれ x 方向、 y 方向の離散化パラメータである。

$$C(x, y)u(x, y) = \oint_{(\tilde{x}, \tilde{y}) \in \Gamma_v} q(\tilde{x}, \tilde{y})w^*(x, y; \tilde{x}, \tilde{y}) - u(\tilde{x}, \tilde{y})\frac{\partial}{\partial n}w^*(x, y; \tilde{x}, \tilde{y})d\tilde{x}d\tilde{y} \quad (\text{A.12})$$

ただし、ここで $C(x, y)$ 、 $w^*(x, y; \tilde{x}, \tilde{y})$ らはそれぞれ形状関数、グリーン関数と呼ばれるものであり、式 (A.13)、式 (A.14) でそれぞれ定義する。また $\theta(x, y)$ は点 (x, y) での領域形状の角度である。

$$C(x, y) \equiv \begin{cases} 1 & (x, y) \in \Omega_2 \\ \theta(x, y)/2\pi & (x, y) \in \Gamma_v \\ 0 & \text{otherwise} \end{cases} \quad (\text{A.13})$$

$$w^*(x, y; \tilde{x}, \tilde{y}) = -\frac{1}{2\pi} \ln \sqrt{(x - \tilde{x})^2 + (y - \tilde{y})^2} \quad (\text{A.14})$$

集中制御系では具体的には式 (A.12) を離散化した方程式を解くのであるが、煩雑であり非常に一般的であるのでここでは省略することにする。

以上示した方法により位置推定を高精度で、いかにもマイクロマシンが無限に広がっているかのような巧妙なトリックで知ることができる。ただし、本文中では用いなかった手法であるため付録として収録することにした。

A.4 DA コンバータボードの設計

十分に位置制御を施された各プランジャーに位置指令を出し、自律分散制御を施すために PC98 の C-Bus 用の 8bit 64ch の DA コンバータポートを制作した。64ch もの多チャンネルのものは市販されていないため、AnalogDevices 社の 8bit 8ch の DA コンバータ用 IC(AD7228) を 8 個用いて実現した。概略の回路図を図 A.6に示す [25] [26]。

設計を行うに当たり注意した点を幾つか挙げておく。

1. 64ch もの多チャンネルのため書き出し速度の低下を抑えるため、cpu に対して no wait で実現した。
2. データバスが 16bit あるのに対して、8bit の出力のみで良いため上位、下位の 8bit づつにわけ一回の書き出しで 2ch かけるようにした。
3. 他の装置への出力を DA コンバータポートへの出力と勘違いしないように I/O アドレス 16bit をフルデコードするようにした。
4. 省スペースのために、また安定した動作のためにワンボード上に全ての装置を組み込んだ。

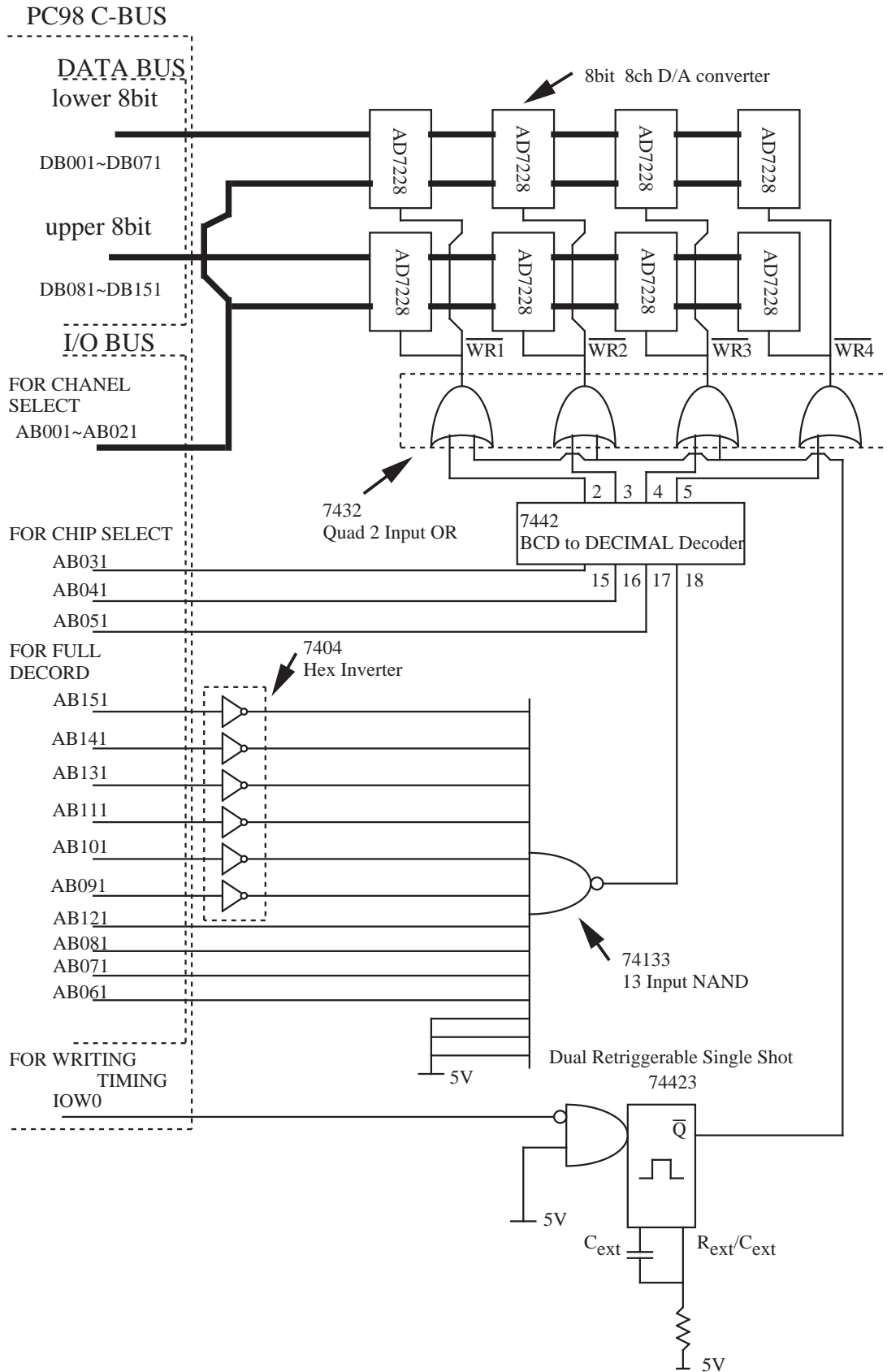


図 A.6: DA コンバータの回路図