均質型機械システムの研究

村田　智

均質型機械システムの研究

平成９年６月

村　田　　智

目　　次

第1章　序　章

1．1　研究の背景

現代は情報化の時代である．いまや家電製品から自動車にいたるまでマイクロプロセッサを内蔵していない工業製品は殆どない．極端な高性能を要求される航空宇宙関係のシステムや，絶対の信頼性が必要な原子力発電システム，効率を追求する高速鉄道システムなどの巨大ハイテクシステムでは，あらゆる部分に情報処理素子が埋め込まれており，全体が大規模で複雑な情報処理ネットワークとして機能するようになっている．

このようにシステムに多様な機能を要求したり，極限的な環境で動作させようとすると，システムそのものの構成が大規模化し，複雑になってゆくのはやむを得ない．構成が複雑になればなるほど，それを設計し，制御することが難しくなってゆく．大規模システムの設計論として，線形動的システム論[1]や熱力学的システム論[2]の立場から理論的なアプローチがいくつか試みられているが，システムの特性が非線形であれば，その挙動を完全に解析することは困難であるし，また，熱力学的・統計力学的アナロジーの成立する（アボガドロ数オーダの）世界と工学の扱うシステムの間にはまだ相当の距離があると言わざるを得ない．結局のところ，解析的な挙動予測はあきらめて，数値シミュレーションによる設計・解析に頼ることになる．しかし，シミュレーションは所詮シミュレーションであって，ひとたびその中の何かの仮定がくずれてしまうと，結果に信頼を置くことができなくなる場合がある．ちょっとした人為的ミスやささいな構造上の欠陥など，わずかな条件の違いが，現実のシステムの挙動とシミュレーション結果を大きく隔ててしまう可能性が常に残っているのである[3][4].

もちろん，われわれがとりうる現実的な方策がシミュレーションのほかにはあまり見あたらない以上，シミュレーションの精度を上げてゆく努力はこれからも続けていかなければならないだろう．しかし，システムの設計の方法論として，大規模かつ複雑なシステムを構築可能でありながら，かつまた，その挙動もある程度予測可能であるというようなものはないだろうか．故障や環境の変化をシミュレーション条件に含

めてゆくといっても，いずれ限界が来るのではなかろうか．そのような方法ではなくて，システムそのものの構造にもっと本質的に工夫をすることでそういう問題を解決することはできないだろうか．極めて複雑なシステムでもその構造の深いところにうまい仕組みを埋め込んでおけば，故障や環境の変化などへの対応が可能になるのではないだろうか．このような考えが，本研究をはじめた動機である．

1．2　大規模システムの抱える問題点

前節で述べた大規模システムは，われわれの工学の到達点を示しているとともに，その限界をも表しているといえる．言い換えれば，これらの大規模システムの抱える問題は，とりもなおさず，われわれの工学のあり方そのものと深く関連している．工学のあり方について，一般的・体系的に議論することは，この小論の範囲を超えるので，そのかわりに，ここでは大規模なシステムの設計論の中に存在する本質的な問題点を抽出し，それについて考察をすすめてみることにしよう．

大規模システムの設計論には，大きく言って3つの問題が指摘できると考える．その第1は耐故障性の問題，第2は拡縮性の問題，そして第3は設計・生産・保守などにかかるコストの問題である．

まず，第1番目の耐故障性の問題であるが，複雑で大規模なシステムは，正常に動作している限りにおいては所期の機能を効率よく発揮するが，いったん一部が故障すると，全体が複雑であるだけにその機能を維持することが困難になる場合があることは既に述べた通りである．当然，これらのシステムに対して耐故障性の観点からさまざまな方策が研究されている．まず，システムに含まれる要素ひとつひとつの信頼性を高めるという方向が考えられる．これは最もオーソドックスな方法であり，最初にとりくむべきことに違いない．ただ，要素の信頼性について述べることはどうしても各論になるので，本研究ではこの方向にはこれ以上言及しない．

耐故障性を高めるもうひとつの方法はシステム内の機能要素の冗長化あるいは多重化である．これは，システムに欠陥が生じても，それが全体に波及して致命的な事態を招かないように設計するという，いわゆる「優美なる機能減退(graceful degradation)」を保証しようとするものである．しかし，このような方法論で対応できる故障の範疇は，機能要素の冗長化・多重化がうまくできる場合に限られる．たとえば，飛行機の

エンジンが複数ある場合，どれかのエンジンが故障しても他のエンジンで飛び続けることができるが，主翼や胴体などの基本的な構造部分が大きく破損した場合には，飛び続けられなくなるというようなことである．

次の拡縮性は，稼働中のシステムを止めずに，サブシステムの追加および削除をおこなう機能のことである．大規模なシステムに対する要求や周囲の環境条件は時々刻々変化するため，これに同一のシステム構成で応えることは困難であることが多い．たとえば，計算機ネットワークのようなシステムではネットワーク内でつねに増設・撤去・移動・保守のための停止などが起こっているため，システム全体の機能を損なわずに機能を維持できる安定性を保証することが，システムの設計者に対する最低限の要件となっている[5]．このような要求に応えるため，これまでとられてきた方策はサブシステムのモジュール化である．全体を複数の機能モジュールに分割し，その組み合わせで要求機能を発現しようとする考え方で，情報処理システムなどの機械的・物理的実体部分の比重の小さなシステムではすでにひろく用いられている．情報だけを扱うシステムでは，モジュール化によってある程度拡縮性を確保できるが，機械部分を含む一般のシステムで拡縮性を保障することはふつうはきわめて困難である．

第３の設計・生産・保守のコストについてみても，大規模システムのかかえる問題の根は深い．システムが複雑になればなるほど，それに含まれる部品の種類は膨大なものとなり，それぞれの仕様を整合させつつ全体の機能を設計する労力は莫大なものである．そのため最近では設計の自動化の研究が急務となっている．このようなシステムに対し，さらに安定性や耐故障性を保証することが要求されるわけだが，システムをとりまくあらゆる状態を想定することは不可能であり，シミュレーションを繰り返して工学的な妥協点を見いだすような設計をおこなわざるを得ないのが現実である．また，そのコストも設計の複雑さに比例して増大する．生産・保守については，システムの構成部品の共通化，品質管理の徹底などの対策がとられているが，抜本的な解決にはなっていない．

１．３　研究の目的

本研究の目的は，前章で述べたような大規模システムの抱える諸問題に対する解決の糸口を示すことにある．ただし，ここでは，すでに実用に供されているような大規

模システムに今すぐに適用できるような方法論を開発するという立場はとらない．そのかわりに，もっと単純なモデルシステムの研究を通して，将来，実用システムにも適用可能な方法論をさぐるという立場をとる．このような立場をあえてとるのには，２つの理由がある．ひとつは，すでに完成された技術が複雑にからまりあった既存のシステムを研究の対象とすると，対症療法的・各論的技術の研究にならざるを得ないという理由であり，もうひとつは問題を一度単純な視点から問い直すことによって，これまで述べてきたような大規模システムの問題を根本的に解決しうるまったく新しい方法論の糸口を得たいという理由である．

１．４　均質型のシステム構成

本研究では，具体的なモデルシステムを設計するという行為を通じて，大規模システムの設計の新しい方法論を探ってゆく．そのときのキーワードとなるのが「均質」という概念である．すなわち，システムの構成要素が「均質」（あるいは互いに同等）であるシステムを扱う．要素が均質であるということは，ある要素を別の要素で置き換えることができるということである．ここで，要素のもつある属性（要素の機能，要素の形態（幾何学的形状），要素の構造（力学的性質），要素のもつ情報（intelligence），要素の意匠など）に注目して，その属性が別の要素で置き換えられるかどうか，という意味での均質性を考えることもできるが，本論文の均質性は，すべての要素が物理的・情報的に全く同じものであることを指すものとする．もちろん，全く均質な要素をいくら集めても，そのままではなんら新しい機能の発現は期待できない．ここで言う要素の均質性とは，たくさんの要素をシステムを構築するための材料として用意したとき，初期時刻においては，すべての要素が同じ状態（特に情報的な意味で）をもっているということであり，その後の構築の段階では，ユニット間であるいはユニットと環境の間にさまざまな相互作用がおこり，次第にユニットの状態は均質でなくなってゆく，ということを考えている．

このような性質を持つシステムは，先に述べた大規模システムの抱える問題を根本的に解決できる可能性をもっている．

具体的に，前述の三つの問題について考えてみよう．まず，第１の問題点，耐故障性については，構成要素の均質性により，故障した要素を，そのまわりに残っている

故障してない要素で代替することができるから，これによって耐故障性をある程度保証できる．さらに，故障要素の置き換えを，外力に頼らずに自動化することができる場合には，故障の自動修復が可能となり，耐故障性の問題は原理的に解決することになる．第2の拡縮性についても，システム全体に追加または削除される部分が，システムそのものと同質であることから，システムの挙動の解析が容易になると考えられる．たとえば，もしもシステムの挙動・特性が要素の数，あるいはシステム全体のスケールに依存しない形で解析できるとすると，どのようにシステムを拡張・または縮小しても，その挙動を予測・保証することができるわけである．第3の設計・生産・保守の問題についてもそれぞれに利点がある．設計については，システムの要素をひとつ設計すれば，あとはその組み合わせでいくらでも大きなシステムが構築できるという利点がある．もちろん，この場合，要素の設計自体が難しくなるのはやむを得ない．また，同種の要素を大量生産すればよいので，生産性の意味でもメリットがある．保守についても要素が徹底して規格化されていることの利点は大きい．一種類の要素について検査・修理などに関する一定の手順を定めておけば，それだけで修復ができるからである．

1．5　情報処理システムとしての大規模システム

大規模システムは，たくさんの構成要素のあいだに張り巡らされた情報交換のネットワークである．システムのあちこちに埋め込まれたマイクロプロセッサは，いわばシステム内の情報処理ハードウェアの基本単位である．これらのプロセッサは単独で機能するわけではなく，センサーやアクチュエータ，他のプロセッサなどと情報をやりとりしながら働くようになっている．したがって，非常におおざっぱに言うと，システムの情報処理ネットワークとしての構造は，これらのプロセッサ群の接続関係を表すグラフで表されると考えられる．

1．6　集中型と分散型の情報処理ネットワーク

大規模システムの情報処理ネットワークには大きくわけて2つの典型的な構造（アーキテクチャ）がある．そのひとつは集中型の構造，もうひとつは分散型の構造である．

集中型のアーキテクチャをもつネットワークでは，全体を統御するひとつの監督プロセッサがあり，他のプロセッサはすべてこのプロセッサによって直接・間接に制御される．したがって，ネットワークの構造は監督プロセッサを根とする樹状（トリー状）構造となる．これを，監督プロセッサを最上位とするプロセッサ群の階層構造と見ることもできる．上位プロセッサは下位プロセッサに命令を伝え，下位プロセッサは上位プロセッサに自分の持っている情報を集約して伝えるという従属関係が基本にある．このようなシステムでは，最上位のプロセッサはシステム全体の情報を総合して最適な判断を下すことができ，また，その判断が速やかに全体に行き渡るという利点がある．すなわち，集中型のシステムでは，その構造は階層的となる場合が多い．現在，実用に供されている大規模システムのほとんどは基本的にはこの構成をとっているといえる．

もうひとつのアーキテクチャ，分散型は網目状構造をもつネットワークである．こちらには全体を統括するようなプロセッサは存在せず，プロセッサ間の従属関係は入り組んでいて，集中型のような階層構造を見いだすことはできない．分散型では，システム全体の機能はこれらのプロセッサの協調によって実現されることになる．

これら集中と分散の構造を組み合わせた折衷様式のネットワーク構造も可能である．ある部分を取ると集中型になっているが，別の部分は分散型になっているとか，集中型ネットワークの一部（サブネットワーク）が分散型で構成されているというような場合である．たとえば，動物の神経系は全体としては脳を中心とする集中型のネットワークであるが，脳そのものは多数の神経細胞の集合体であって，分散型である．

1．7　均質システムにおける集中型と分散型

均質型システムでは各構成要素はすべて同等であり，したがって，システムに含まれるプロセッサも同等である．つまり，均質型システムにおいては，すべてのプロセッサが情報処理ネットワークのどの部分でも担当できる潜在的能力を備えていることが必須となる．これは効率・経済性だけを見た場合，明らかに冗長であるが，先に見てきたように，耐故障性や拡縮性などの面からは本質的に有利な面をもっている．

均質型システムにも，やはり集中型と分散型の構造が考えられ，それぞれに得失がある．集中型の構成を取る場合，先に述べた集中型システムと同様，最適性や即応性

の面では有利である．しかし，階層ごとのプロセッサの処理内容は異なるので，各プロセッサに埋め込むソフトウェアを階層ごとにいくつも用意して，それを適宜切り替えることが必要になる．また，耐故障性の面では，ネットワークの上層が故障した場合にそれ以下の階層全体が機能しなくなる恐れがある．

一方，分散型のネットワークでは，均質なプロセッサ群が特別のリーダーなしに網目状に結合しているので，何らかの方法を使って，このネットワークのなかに望みの機能を埋め込むことが必要である．均質システムではネットワークの構成要素もすべて全く同じものであるので，機能の埋め込みは一層難しくなる．機能の埋め込み，つまり，均質な構造から不均質で意味のある構造を生成するしくみは，集中型のように望みの機能を細かく分解してゆけばおのずと得られるというものではない．しかし，脳の可塑性などに代表されるように，このようなネットワークは耐故障性，環境的応性などきわめて柔軟な能力を秘めている．

以上を簡単にまとめるならば，システムにどのような機能を望むかによって，自ずとその構成が決まってくるということになろう．効率や精度を望むのであれば，集中型の構成がよいが，内部の故障や環境の変化など，不測の事態に対応できることを重視するならば分散型の構成の方が利点が多い．しかし，工学システムとしては，このどちらも重要であるので，設計者はこれらの諸条件をバランスさせながら設計解をさがすことになる．このとき，効率や精度などの数値的な目標は評価関数として単一のパラメータに集約することができ，その最適化として設計問題を定式化できる．ところが，環境適応性や耐故障性などのシステムの構造あるいはトポロジーに関わるような目標はそのような数学的表現が難しいため，それを定量的に評価することは困難である．すなわち，現在はまだシステム設計のすべての目標を一本化し，ひとつの最適化問題に落としこむことができない．したがって，システムをどの程度集中あるいは分散型にするかという全体構成や，どんな指標をもとにシステムを設計するかということは，結局のところ設計者の判断に委ねられているのである．

1．8　均質型システムをめぐる研究の流れ

これまで，均質型機械システムについてやや抽象的な観点から整理を試みたが，この節では，これまでこの分野でどのような研究が行われてきたかを概観する．均質型

システムの研究分野の歴史は以下に示すように，１９５０年代まで遡ることができるが，これまで決して多くの研究がなされてきたとはいえない．しかし，この分野にも，少数ではあるが，非常に重要かつ本質的な研究があるので，それらを中心に均質システムをめぐる研究の流れを考えてみたい．

均質型の発想は，システムが大規模・複雑になって，従来の設計論の壁にあたるところからでてくることはすでに述べたとおりである．アメリカの数学者 von Neumann は，そのような局面に遭遇したことを認識した最初の人ではなかったかと思われる．彼がおこなった仕事は本研究のいわば原点に相当するので，以下，少し詳しく説明する[6].

von Neumann は世界最初のコンピュータENIACの開発に携わったことで有名だが，その設計中に，大規模なシステムでは，素子の信頼性を考慮した設計をしなければならないことに気がついた．当時の演算素子は真空管や遅延線などのアナログ素子で，このような素子を数万点集めたシステムでは，いつもどれかが故障しており，ENIACでは計算の仕事をしている時間より修理（素子の交換）をしている時間のほうが長かったといわれている．彼はこのような経験から，「信頼しえない素子から，信頼しうるシステムを構成するにはどうすればよいか」という問題意識をもつようになり，そこからさらに進んで，複雑なオートマトンの研究，特に自己増殖オートマトンの論理組織を研究するに至った．

自己増殖という機能を考えるにあたって，von Neumann はいくつかのモデルを提案している．第１のモデルは運動学モデル（kinematic model）と呼ばれる．このモデルでは，運動，接触，位置決め，融合，切断等の機能を持つ素子がたくさん空間に浮遊していてものとする．そして，これらの素子を集めてあるシステムを組み立てると，そのシステムはまわりに浮遊している部品を集めて自分と同じシステムを組み立てることができる．そのようなシステムをどうやって設計するか．von Neumann はこの問題を１９４８年頃まで考えていたといわれるが，結局，難しすぎてアプローチを変えることになる．それが第２のモデル，細胞モデル（cellular model）である．これはUlamによって示唆されたもので，このモデルでは全く同じ有限オートマトンがひとつずつ入った細胞が平面上に敷き詰められた数学的な空間を考える．各細胞は，有限種類の状態の中からひとつの状態をとる．また，その状態はまわりの細胞の状態に応じて別の状態に

遷移する．このとき空間上の細胞群にあるパターンの状態をとらせ，それをある遷移規則で遷移させてゆくことにより，パターンの時間空間的発展を観察することができる．von Neumann はあるパターンが自分と同じパターンを別の場所に複製するためには，どのような初期状態と遷移規則が必要かを研究した．これを自己増殖オートマトンと呼ぶ．１９５３年頃まで彼はこの仕事を続けたが，結局完成できずに亡くなった（１９５７年）．後に彼の方針にしたがって完成されたパターンは２９状態のオートマトンを２０万個も使用する巨大なものであった．この自己増殖オートマトンこそ，理論的に均質型システムをあつかった最初の研究であると考えられる．

von Neumann の自己増殖オートマトンが大きな意味を持つのは，このモデルが自己複製の具体的描像を与えるからだけでなく，モデルに進化・適応のメカニズムを埋め込むことができるからである．すなわち，状態パターンとして万能計算機械（チューリング機械）の埋め込みが可能なこと，および複製過程に突然変異を導入することによって，進化する機械を構成する枠組みを提供することができるのである．

このように von Neumann の仕事は本質をついたものであったが，そのモデルが有限オートマトンという数学的・抽象的な世界のモデルであったため，実世界の「もの」には結びつかず，２０数年後にLangtonらによる「人工生命」として再登場するまで，ほとんど省みられることはなかった．（人工生命については後述する．）

ところで，量子力学の建設者の一人 Schlödinger は，「生命とは何か」という著書（１９４４年）の中で，生物といえども物理科学法則に従う分子の機械であり，その核心は繊維状の高分子にあると喝破した[7]．はたして１９５３年，Watson と Crick によってＤＮＡの２重らせん模型が提出されると[8]，すさまじい勢いで生物の分子的機構の解明が始まった．そして研究が進めば進むほど，生物がきわめて精巧な分散システムであることがわかってきた．多細胞生物は，すべての細胞が基本的に同じ情報をもとに協調する「超分散システム」なのである．その根本をになう細胞分裂のしくみ，細胞分化のしくみ，免疫のはたらきなどの分子機構がこれまで次々と明らかになってきている．従来観察や実験だけに頼っていた発生学等の分野でも，たとえば線虫（c.elegance）などの原始的な多細胞生物では，すでに発生の全過程における細胞の系譜が記載されているのみでなく，その過程を支配する遺伝子まで特定されつつある[9]．

こうした生物学の動きに刺激されて，簡単なモデルで生物の代謝や分裂を表現しよ

うとする試みが，Penrose　によっておこなわれた（１９６０年頃）[10]．彼は，特殊な機械ユニットを工夫し，生物の増殖をモデル化した．このユニットは，フックやバネなどの単純な機構の組み合わせによって他のユニットと結合できるようになっている．そして，この結合の仕方によってごく簡単な演算機能を実現した．たとえば，結合するユニット数を３以下に保つという機能をもつユニットは，ユニットを４個つなぐと２個と２個に分かれてしまう．ここで，ユニット２個の組が生物の成体とをあらわすとし，それが餌（別のユニット）を食べて，２体に分裂したと見るのである．このモデルは，たしかに生物の増殖をあらわすにはプリミティブに過ぎるかもしれないが，実体のあるモデルとして提示されているところに，von Neumann のモデルとは違うインパクトがあった．

１９７０年代も後半になると，ＬＳＩ技術の進歩によりコンピュータが普及し，大量の技術計算が誰にでも手軽にできる時代がはじまる．コンピュータの計算力を背景に，さまざまな「生命もどき」のモデルが提案されるようになる．Conwayのライフゲーム[11]はその代表的なものである．これはvon Neumann の自己増殖オートマトンを非常に単純化したもので，各セルは０か１のふたつの状態しかないとし，その遷移則も上下左右のセルの値の和だけによってつぎの状態を定めるようになっている．しかし，セルの初期配列によっては，非常に多様な時空間的パターンに発展する．一定のサイクルでいろいろなかたちを一巡りする発信パターンや，セル空間上を移動するグライダーと呼ばれるパターン，また，特定のパターンを次々に生産するブリーダーと呼ばれるパターンなどが発見されている．

数学計算支援ソフトMathematicaの開発で有名なWolfram はライフゲーム(の１次元版)に現れるパターンを３つに分類した[12]．静的に固定したパターン，または周期的なパターン，全く予測のできないランダムなパターン，そして，その中間の完全な予測はできないが一定の秩序が感じられるパターンである．

Langtonはライフゲームほど単純ではないが　von　Neumannの自己増殖オートマトンほどは複雑でないオートマトンにより自己増殖の過程をあらわす見事なモデルを設計した．そしてこれを「人工生命」と呼んだ．Langtonのモデルそのものは，von Neumannのモデルをうまく単純化したものに過ぎないが，この研究は非常な注目を集め，多くの研究者がセルオートマトンなどの自己組織システムの研究に参加するに至った．そし

て，その中から「カオスの縁（Edge of chaos）」という概念が提出された．カオスの縁とは，先に述べたWolflamの第3の分類に相当するもので，周期解とランダム解の間に，意味のある情報処理体系が存在するのではないかという仮説である．この仮説にしたがって，セルオートマトンのような均質型のシステムには，生物のような柔軟な情報処理能力があるのではないかという期待が生まれ，「複雑系（Complex　Systems）」の研究がひとつのブームとなっている[13].

わが国では，このような流れを背景に，１９８０年代の後半から「自律分散システム」の研究が始まった[14]-[16]．自律分散システムは，セルオートマトンのように純粋な情報処理だけを考えるのではなく，物理的な実体をもつシステムを念頭に置きながら，新しいシステム設計の方法論を探るところにその特徴がある．１９９０年，伊藤は，それをつぎのように定義した[17].

「システムを構成する各要素が個々に自律性を保ちつつ行動しながらお互いに協調し，システム全体としての秩序を生成するシステム」

この時点では，研究の方向性はまだ明確でなく，「概念だけが先行していて，その理論展開はまだほとんど行われておらず，どのような構成原理によれば自律分散システムが実現されるのかわかっていない[18]」状態であった．しかし，自律分散システムという名称が学会等で認知され，次第にまとまった数の発表がなされるようになった．こうして，自律分散システムのイメージはしだいに具体化してきたが，未だその明確な定義，いわんや操作性のある統一的理論が提案されるには至っていない．現時点で最も詳細に自律分散システムを定義したものは，１９９３年，同じ伊藤によるもので，つぎの７つの構成要件があげられている[18].

（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）システムおよびそれを構成する個はエネルギー代謝のある開放系で，常に活性化される．
（e）各個はシステム全体の大域的秩序に関する情報をもち，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）．そして，そこには個間相互作用の場が存在する(相互作用の場)．なお，その場合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，自己組織性）．

（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）．

これを見ると，自律分散システムは，生物という分散システムの取っている構成の特徴を抽出し，そのアナロジーとして定義されていることがわかる．

以上，均質システムをめぐる研究の流れを見てきた．ここからわかることは，要するに，均質システムの研究とは，いかにして生物のメカニズムを人工物で実現するかという研究なのである．生物を究極の手本として，人工物の設計論を根本的に改革することが，均質システムの研究の最終目標である．

1．9　本研究の位置づけ

本研究では，具体的なモデルシステムの構築を通して，均質型機械システムの設計論を展開する．そのモデルシステムとして，2つのシステムを設計・製作する．これら2つのモデルシステムはそれぞれ1．7節で述べた集中型と分散型の均質システムの典型例として扱われる．もちろん，これら2つのモデルの構築事例から得られる知見はそれぞれのモデル特有の条件に制約され，均質型システムの一般的・体系的設計論を展開するのに不十分であることは否めない．しかし，これらのモデルを実際に設計する作業の中から，均質システム設計のどこに本質的な困難さがあるかを明らかにし，また，それを解決する方法を具体的・実際的あるいは詳細に検討してゆくことができると考えられる．われわれが「設計」という行為について論ずるとき，どうしても議論が抽象的になる傾向があるが，工学システムを扱う以上，抽象論に終始するのではなく，具体的な「もの」を相手に具体的な議論を尽くしてこそ，実用に耐えうる方法論が導かれるのではないかと考える．

第1のモデルシステムは，「集中ー均質型」の構成をとるシステムで，「ソフトリ

ンクビークルシステム」と呼ばれる．このシステムは，移動ロボット（ビークル）をその基本要素としてビークル群のシステムを構成し，ビークルが隊列をなして走行する「ソフトリンク」機能を実現するものである．「ソフトリンク」とは，機械的な連結器によらず，通信によってたくさんの車両を仮想的に連結することを意味する．この技術を確立することにより，車間距離の短縮による道路容量の増加，空気抵抗の減少による燃費向上，相対速度減少による安全性の向上，経路情報や事故情報などさまざまな種類の情報伝達による車両群システムの柔軟な運用などが期待される．このモデルシステムにおいては，先頭車に後続車が一定車間距離で追従するための制御システムを，車両のダイナミクスを考慮しながら設計する．ここでは，車両の位置姿勢を推定する技術，車両間の通信技術，車間距離制御則の衝突回避特性などが，重要なポイントとなる．

ソフトリンクビークルシステムは，大規模システムの設計として従来の方法論を適用した場合の事例として位置づけられる．これまで述べてきたように，集中－均質型の構成を取ることにより，理論的見通しの良さ，安定性や制御特性の定量的評価など，従来の制御理論の範囲内でも実用に耐えるシステムが構築可能なことが示される．しかし，その一方で，耐故障性の問題については，集中－均質型の構成は本質的に不利な面をもっていることも明らかになる．そして，この問題により深くアプローチするために，第２のモデルシステムが検討される．

第２のモデルシステムは，「分散－均質型」の機械システムのモデルとしてのユニット型機械システムである．互いの結合を変更する能力をもった自律ユニットをたくさん用意することにより，「自由にかたちを変える」という従来の機械にはない機能を持つシステムを実現できる．ここで提案するユニットは，それぞれが独立したCPUを備えており，自分と隣接ユニットの間の結合を電磁力をもちいた簡単なアクチュエータで自律的に変更する機能を持っている．また，各ユニットは光通信によって隣接ユニットと情報交換ができるようにもなっている．このような機能を利用すると，ユニット群がが互いに協調して望みの構造を組み立てたり（自己組み立て），また，その構造が破損した場合にそれを自動的に修復する（自己修復）ことが実現される．そのようなユニットのハードウェアの構成について，あるいは，このユニットのためのソフトウェアについて詳細に検討することにより，均質ユニット型機械のさまざまな

可能性を示すことができる．本論文ではこのような均質ユニット機械を具体的に設計し，その機能を実験及び計算機シミュレーションで検証してゆく．

1．8節の均質型システムの研究の流れの中にこの第2のモデルシステムを位置づけてみよう．先に述べたように，本研究は，von Neumann の自己増殖オートマトンの研究にその原点をもとめることができるが，ここで提案するユニット型機械システムはある意味で，von Neumann の理論を越える方法論を示しているということができる．すなわち，von Neumann は，当初，運動学モデルという直接的なモデルによって，自己増殖の過程を構築しようとした．しかし，それがあまりにも困難であったために，物理的実体を捨象して純粋に数学的な細胞モデルに移った．本論文のユニット型機械システムは，von Neumann の運動学モデルにもう一度立ち戻って，実体のある機械システムを構築しようとする試みに他ならない．ここで提案されるアプローチは，機械ユニットという実体をもつオートマトンをもちいて，細胞モデルのような均質システムを構成するところに特徴がある．ユニットにユニット同士のつながり方を組み換え能力をもたせたことにより，無限のセル空間を仮定することなく，実体のある要素でシステムを構築することが可能になったわけである．ユニットの実体という点では，Penroseのモデルがあるが，CPUなどの情報処理機構がなかったために，Penroseモデルの機能はきわめて限定されたものであった．これに対し，本論文で提案するユニットはCPUに動作プログラムを与えることにより，自由なソフトウェア設計ができるところが大きな違いである．

もちろん，第3章で詳しく紹介するように，ここで提案するユニット型機械システム以外にも，ユニット型の構成をもつロボットの研究はいくつか行われている．たとえば，福田らはＣＥＢＯＴと呼ばれるセル型のロボットに関してハードウェア，ソフトウェアの両面から非常に広範囲の研究を行っている．また，米国のChirikjianらのリンク型ユニットの研究もある．これらの研究のなかで，本研究の特徴を一言でいうならば，「徹底した均質化と単純化」ということになるであろう．本研究はその根底に，ユニットのハードウェア・ソフトウェアともに，均質化の極限を追求し，「均質」という枠組みの可能性を探るという強い問題意識があり，このような考え方が徹底しているのがその特徴である．それを具体的に見るために，前節で紹介した自律分散システムと対比してみよう．

自律分散システムにおける「個」を機械ユニットと見，「秩序」を全体の形状に対応させれば，自律分散システムの構成要件の多くが，提案する均質型ユニット機械にあてはまる．すなわち，均質な個（ユニット）は自律性をもち（独自のＣＰＵで制御され），全体の大域的秩序に関する情報（形状の記述）にもとづいて，全体の秩序(形状）をつくり出す．そこには局所通信による相互作用の場（拡散場や完成度の分布など）が存在する，というように，自律分散システムの構成要件は均質型ユニットの構成に読み替えることができるのである．自律分散システムという枠組みが提案されて以来，さまざまなシステムが提案されてはきたが，本論文で提案するシステムほど，その本質をよく具体化しているシステムはこれまでなかったのではないかと考えられる．

さて，第１のモデルシステムが集中－均質型の構成によって，実用的なシステムを構築しつつその問題点を明らかにするのに対し，第２のモデルシステムは，分散－均質型の新しいアプローチを導入することにより，自己組み立てや自己修復などの新しい機能の実現可能性を示すものである．しかし，これらの２つのモデルシステムは，集中型か分散型かというひとつの切り口だけを示しているのではない．本研究で取り上げるこれらの２つの事例は，以下にあげるように他にもいくつかの対照的な側面をもっている．

1）幾何学的な拘束の違い：ソフトリンクビークルシステムでは要素（ビークル）同士は物理的に離れており，互いに自由な位置・姿勢をとることができるのに対し，ユニット型機械では，ユニット同士は常に決まった位置関係にあって機械的に結合していなければならない．

2）システムの構造の違い：ソフトリンクビークルシステムが隊列という基本的に１次元のひも状の構成を取るのに対し，ユニット型機械は２次元あるいは３次元のネットワーク構造をもつシステムである．

3）主要な変数の種類：ソフトリンクビークルシステムにおいては主要な変数は，位置，速度，加速度などのアナログ量であるのに対し，ユニット型機械は基本的には有限オートマトンであり，その変数は，どの腕をつかって結合しているかというような離散的な状態量となる．

4）要素の大きさの違い：ソフトリンクビークルシステムが自動車交通への応用を目指しているのに対し，ユニット機械ではユニットをむしろマイクロ化する方向に向かっている．

ここに対比した相違点は，本論の2つのモデルシステムの選び方に直接依存するものであって，これだけで均質システムの設計論のあらゆる側面を検討したことにはならないのはもちろんであるが，注意深くこれらのモデルシステムの設計の過程を検討することによって，均質システムの設計論のいろいろな側面からの知見を得ることできるのではないかと考える．

1．10　論文の構成

この節では，本論文のアウトラインを説明する．本論文は Fig.1.1 に示すように，大きく2つの部分に分けられる．そのひとつが，集中型均質システムの事例である「ソフトリンクビークルシステム」の研究，もう一つが分散型均質システムの事例である「ユニット型機械システム」の研究である．第2章以降で，これら2つのモデルシステムの構築についてそれぞれ論じてゆく．ソフトリンクビークルシステムについては第2章があてられる．この章の中で，ソフトリンクビークルのハードウェア，通信，制御，走行実験などが説明される．ユニット型機械システムについては第3章から第6章までがあてられる．第3章では，おもにユニット型機械のハードウェアについて説明する．ユニット型機械のソフトウェアについては，2つの系統の異なる方法が示される．まず，第4章では第1の系統である小規模な（10ユニット程度）自己組み立てソフトウェアが示さる．次に第5章では第2の系統のもっとユニット数が多い場合の自己組み立てを別のアプローチによって導く．さらに，第6章では第5章の手法を拡張した自己修復手法が示される．最後の第7章では，2つのモデルシステムの構築について振り返り，将来を展望する．

以下，もう少し詳しく各章の内容を説明する．

第2章では，集中型の均質機械システムのモデルとして，たくさんの車両が一定の車間距離をおきながら一群となって走行するソフトリンクビークルシステムの設計と構築を行う．このシステムでは，各車両は前後の車両と光通信によって結ばれており，

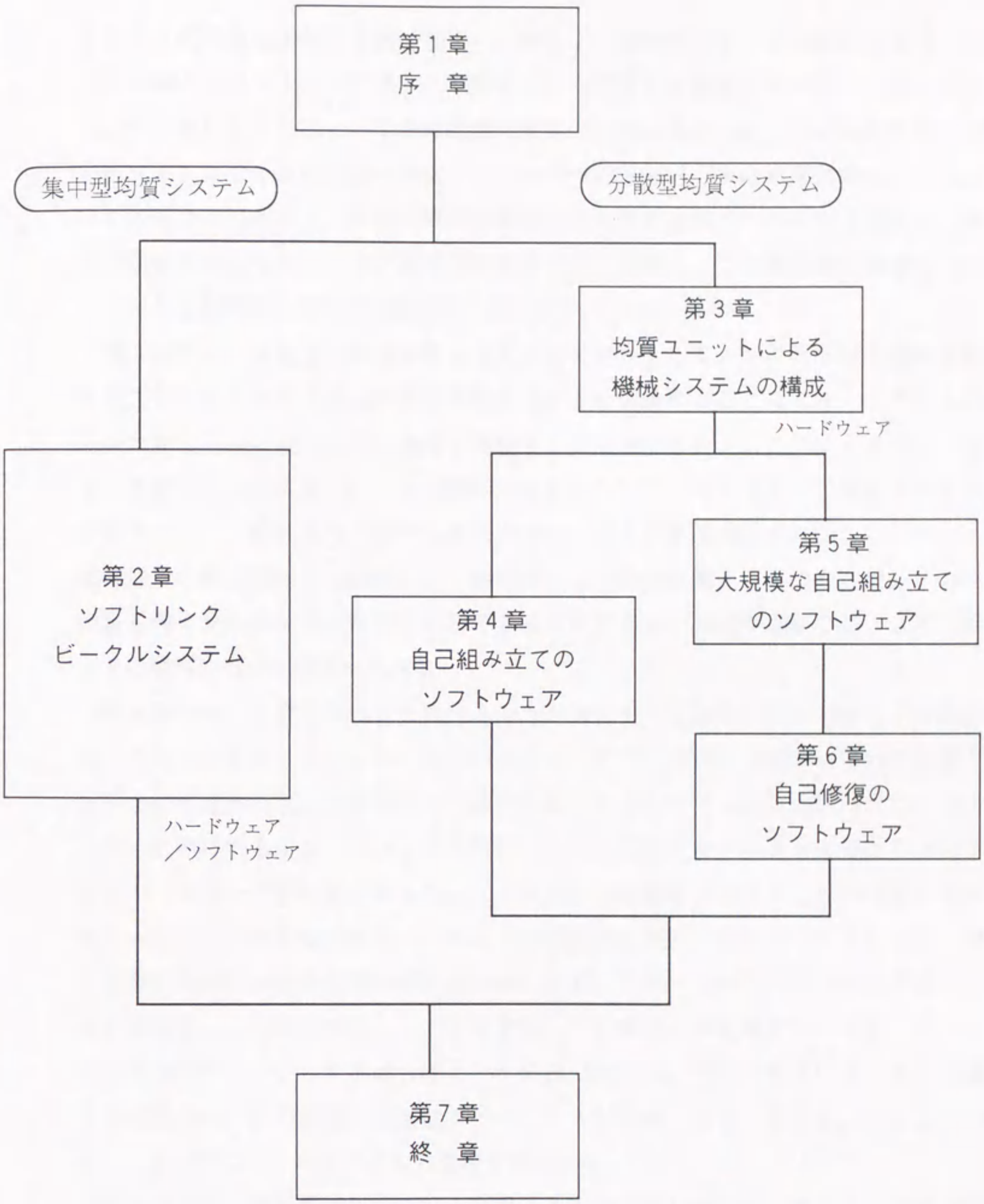


Fig. 1.1 Guide for the reader

あたかも機械的な連結器で結ばれているかのように運動する．車両群は先頭車を中心とする集中管理システムであり，先頭車からの情報を後続車が受け取り，順次後ろにリレーしていくことによって車間距離の制御が実現される．はじめに車両単体の運動方程式およびその位置姿勢の計測システムについてのべ，次に車両間通信の方法について説明する．そして，多数の車両を連結したシステムダイナミクスを構成し，車両間で衝突が起こらないように制御則を設計する．最後に，この制御則を階層化し，もっと大きな車両群に適用する場合の方法を示す．

第３章では，分散型の均質機械システムのモデルとして，ユニット型の機械を設計する．システムをたくさんの同じ自律ユニットで構成することにより，システム全体のかたちを自由に変えられる機械を実現することができる．ここでは，まず，そのような自律ユニットにはどのような機能が必要かというハードウェアに関する設計条件を明らかにし，その上で，設計した２次元ユニットの詳細が示される．このユニットは電磁式の結合機能と光通信による隣接ユニット間通信機能をもつもので，ユニットの組み換えを外部からの助けなしにできるものである．３章の最後では，３次元ユニットの構想について簡単に述べる．

第４章では，３章で述べた２次元ユニットシステムのためのソフトウェアを設計する．このソフトウェアは，たくさんのユニットがでたらめにつながっている状態を，あらかじめ定められた目標形状に収束させるためのもので，「自己組み立てソフトウェア」と呼ばれる．ソフトウェアの設計にあたっては，各ユニットに与えるプログラムはすべて同一でなければならないし，ユニットは隣接ユニットとしか情報交換ができないという制約条件がある．このような制限のもとで，全体のかたちという大域的な目標を達成するための方法が示される．まず，ユニットの局所的な結合状況をあらわす結合型という記号を導入し，それを使って目標形状を記述する．次に，ユニットは目標形状からのずれを不適合度という関数で評価し，ずれの大きいユニットを優先させて動かす．この過程には拡散ダイナミクスが応用される．最後に，シミュレーションによってこのアルゴリズムの機能を検証する．

第５章では，数百程度のユニットを扱える方法を別の観点から考える．この方法は生物が１個の受精卵から分裂を繰り返して段階的に複雑な組織をつくり出すことにヒントを得たもので，タマネギ法と呼ばれる．この方法では，はじめに核と呼ばれる中

心ユニットを選出し，そのユニットのまわりに一層ずつユニットを積み重ねるようにして全体のかたちを作ってゆく．すべてのユニットは，組み立ての過程をあらわす記述行列と呼ばれる特殊な設計図をもっており，自分がその設計図の中のどの段階のどの部分を担当するかを認識することによって，システム内の適当な役割に分化してゆく．ユニットは隣接ユニットとしか情報交換できないので，その制限内でうまく分担を決めるための方法が必要となる．最後にこのアルゴリズムを用いて複雑な形状の自己組み立てができることが，シミュレーション実験で示される．

第６章では，システムが全体のかたちを組み立てるだけでなく，そのかたちがこわれたときに，こわれた部分を自動的に修理するという「自己修復」の方法について述べる．前の章で説明した大規模な自己組み立てのアルゴリズムは組み立てを段階的に行うようになっているので，これを自然に拡張することにより，自己修復のアルゴリズムを得ることができる．すなわち，システムが故障を検出したとき，その故障の程度に応じて，自己組み立ての段階を後戻りさせ，適当なところまで組み立てのプロセスをさかのぼってやり直せばよいのである．そのための故障検出の方法，システム全体を退化させる方法などが導入される．シミュレーション実験によって，このような拡張によって，システムの任意部分の欠落に対応できるようになることを示す．また，この章の最後では，これまで提案されてきたさまざまな自動修復手法を振り返り，ここで提案した手法を位置づける．

最後の章では，ここまでに述べてきた均質型機械システムの２つのモデルシステム，ソフトリンクビークルシステムとユニット型機械システムを振り返り，これらを設計する仕事を通して，何がなされ，そしてどういう問題が残されているかをまとめる．そして，大規模で複雑な人工物を構築するための新しい設計論を展望する．

第2章　ソフトリンクビークルシステム

2．1　はじめに

集中型の均質分散機械システムの典型的なモデルとして，ソフトリンクビークルシステムについて述べる．ソフトリンクビークルシステムは，自動車交通などの実用のトランスポーテションシステムに応用することを最終的な目標として開発した実験システムである[19]．システムの要素としては，工場用の無人搬送車などの移動体（ビークル）を考え，これらのビークル一台一台の位置や速度を車載のコントローラによって制御することにより，ビークル群全体としてのパフォーマンスの向上を図るものである．群としての制御を行うためにはビークル間の情報交換が不可欠であるため，ビークル間のデジタル通信技術が重要である．本論文で提案する「ソフトリンク[註)]」とは，機械的な連結器，すなわち「ハードウェアによるリンク」の対極にあるもので，情報交換だけによって，あたかもビークル間に機械的な連結器が存在するかのような効果をもたせるものである．この車両間通信によって得た情報にもとづいて，各車両ごとに衝突回避のための制御系を構成することになる．

註）多数の車が一定の車間距離を保ちながら隊列となって走行する場合，「プラトゥーニング」と呼ぶのが一般的であるが，ここで提案する「ソフトリンク」もほとんど同じ概念である．ただ，ソフトリンクでは，車間距離をレーダー等で直接で測ることはせず，車両がその絶対位置を知っているとして，情報交換だけで（つまりソフトウェアだけで）車間距離を保つところに特徴があるため，本論文で提案するシステムに限り「ソフトリンク」という言葉を使用することにした．

2．2　ソフトリンクビークルシステムとその背景

たくさんの車両が隊列をなして走行するプラトゥーニング技術の開発は，１９８０年代後半から世界各国ではじまった自動車交通システムの開発の流れの中でおこなわれている．この流れは大きく日米欧の3極にわけることができる．まず，ヨーロッパ諸国では，ＤＲＩＶＥプロジェクトおよびＰＲＯＭＥＴＨＥＵＳとよばれるプロジェクトがＥＵを中心に実行されており，交通管理，経路誘導，運転支援，自動運転，事故通報などの多方面にわたって研究がすすめられている．少し遅れて米国でも１９９

０年からＩＶＨＳプロジェクトを，また，わが国でも１９８０年代なかばから運輸省，警察庁，郵政省，通産省などの主導でいくつかのプロジェクトが推進されている．これらはいずれも道路インフラおよび車両の情報化によって，事故防止，渋滞緩和，環境保全をめざすものである．これらの研究は，まとめてＩＴＳ（Intelligent Transportation Systems）と呼ばれ，一大研究分野を形成している[20],[21].

プラトゥーニングの研究はこのＩＴＳの研究のひとつとして行われている．プラトゥーニングにより期待される効果は3つある．第１は，短い車頭間隔で多くの車両を運行できるため道路容量を増すことができることである．第２は，車両間の距離が一定に保たれること，つまり，車両間の相対速度がゼロになることにより，安全性向上の効果が期待されることである．第３は，車両間隔一定で高速走行した場合の空気抵抗低減効果，つまり燃費の向上である．

これまでのおもなプラトゥーニングの研究としては，まず，ヨーロッパのＰＲＯＭＥＴＨＥＵＳ計画の中で，ドイツのフォルクスワーゲン社が開発したシステム（彼らはＣＯＮＶＯＹシステムとよんでいる）があげられる[22]．これは車上のセンサで先行車との車間距離，速度，加速度などを測定し，車間距離を自動的に保持するシステムで，高速道路上の実験がおこなわれた（この走行実験のビデオは存在するが，詳細については公表されていない）．もうひとつは，米国のＩＶＨＳの一環としてカリフォルニア州でおこなわれたＰＡＴＨプロジェクト（Partners for Advanced Transit and Highways）の中でおこなわれた研究である．こちらは，経路のリファレンスとして磁気ネイルと呼ばれる永久磁石列を路面に埋め込み，車上の磁気センサでコースずれを検知して操舵制御をおこなう．車間距離は上述のフォルクスワーゲンのシステムと同じようにレーダー計測するが，これとは別に車両間の無線通信によっても情報交換ができるようになっている．車間距離の制御モデルとしては車両の運動方程式に基づくスライディングモード制御が試みられている．１９９２年の春にはサンディエゴの高速道路で２台ないし４台の自動車を使った走行実験をおこない，車間距離８～９ｍ，速度８８ｋｍ／ｈで走行することができたと報告されている[23][24].

これと平行して，わが国でも（財）自動車走行電子技術協会などを中心に早い時期から車両間通信技術を利用した自動車交通のシステムが提案されており，経路案内等の実験がおこなわれている[25]-[28]. 本論文で述べるソフトリンクビークルシステムは

これに続くものであり，上述のような海外のプラトゥーニングシステムと相前後して提案されたものである．

ここで提案するソフトリンクビークルシステムでは，車両間個別通信を複数台の自律車両に適用し，車間距離を小さく保った状態で車両群の追従走行を制御する．このシステムは比較的小規模な実験システムであり，車両としては工場用無人搬送車を用いているが，車間距離の計測をおこなわず，車両間通信と車両ごとの速度制御のみによって車間距離を一定に保つ機能を実現しているという特徴をもつ．また，このシステムでもちいている制御則は比較的単純な線形制御則であり，衝突回避特性が定量的に評価できること，通信が一方向でよいこと，多重連結した場合や，さらにそれを階層化した大規模な車群システムを考えた場合の挙動予測が比較的容易にできるなどの利点もある．また，車両間の連結動作をハードウェアとは全く無関係な通信のみを使うために，接続，切り離しの制御が容易であり，将来，もっと大規模なシステムを構成する場合には，柔軟なシステム構成が可能になると期待される．

ソフトリンクビークルシステムの実体は，複数台のソフトリンクビークルと地上に設置されたコントロールセンタである[29][30]．各車両は，自律走行，車両間通信，および路車間通信機能をもつ．コントロールセンタは走行経路状の要所に路車間通信装置を備えており，その前を通る車両と通信することができる．車両間では，車両間通信装置により，ほぼリアルタイムの情報交換ができる．各車両はまず，路車間通信によって，コントロールセンタから目的地情報や作業内容などの運行計画に関する情報を得る．これらの情報に基づいて各車両は単独で自律的に，あるいは車両間通信によって複数の車両がソフトリンクした状態で走行する．車両間では，車間距離を小さく保つために車両の走行制御に関するデータと運行制御に関するデータが継続的に送受される．

以下では，試作したソフトリンクビークルの車載コントローラと車両間通信装置の構成，走行制御アルゴリズム，単独車両の自律航行，ソフトリンク走行実験の結果，アルゴリズムの拡張性などについて説明する．

2．3　ソフトリンクビークル

2．3．1　車両の構成

Fig. 2.1 にソフトリンクビークルを示す．使用した車両は工場内無人搬送車で，大きさは1.3m×0.75m，積載量150 kg，自重 80 kg，最大走行速度 0.7m/sである．操舵方式は差動操舵で，車両の中央左右に駆動輪があり，前後中央に 1 個ずつキャスタがある．車両のトレッドは 0.50m，駆動輪の直径は約 0.25m である．

ソフトリンクビークルの構成を Fig.2.2 に示す．各車両は，デッドレコニングによって自車の位置と方位を車上でリアルタイムで計測し，それに基づいて自律航行を行う．また，赤外線を媒体とした車両間通信装置をもっており，隣接した車両間で 1 対 1 の個別通信を行う．自律航行機能と車両間通信機能はソフトリンクビークルに必須である．この２つの機能は小型コンピュータ（HP9000/310）を用いた車載コントローラで制御される．車載コントローラの制御周期を　655.36msとし，この間に車両の位置と方位の計測，車両間通信によるデータの送受，および操舵と速度の決定を行う．操舵量と速度は，差動操舵方式にあわせて左右輪の速度になおし，ディジタルインターフェース（GPIO），D / A 変換器（左右輪各12 ビット）を経由して車輪駆動装置に送る．

2．3．2　デッドレコニング

－内界センサによる車両の位置方位の計測－

ビークルが外界センサを使用しないで自分の位置や姿勢を推定することをデッドレコニング（推測航法，盲目航法）という．これにはいくつかの方法が考案されている．加速度センサの出力を積分する方法，レートジャイロを用いる方法，車輪の回転角を積分する方法などがあるが，ここでは，簡便・安価で車両に適用しやすい車輪の回転角計測にもとづく方法を採用する[31]-[33].

車両がスリップを生じることなく低速で移動する場合，そのダイナミクスは　Fig.2.3 の地上座標系（x-y 系）で

$$\dot{x} = v \cos\theta \tag{1}$$

$$\dot{y} = v \sin\theta \tag{2}$$

$$\dot{\theta} = \frac{v_r - v_l}{d} \tag{3}$$

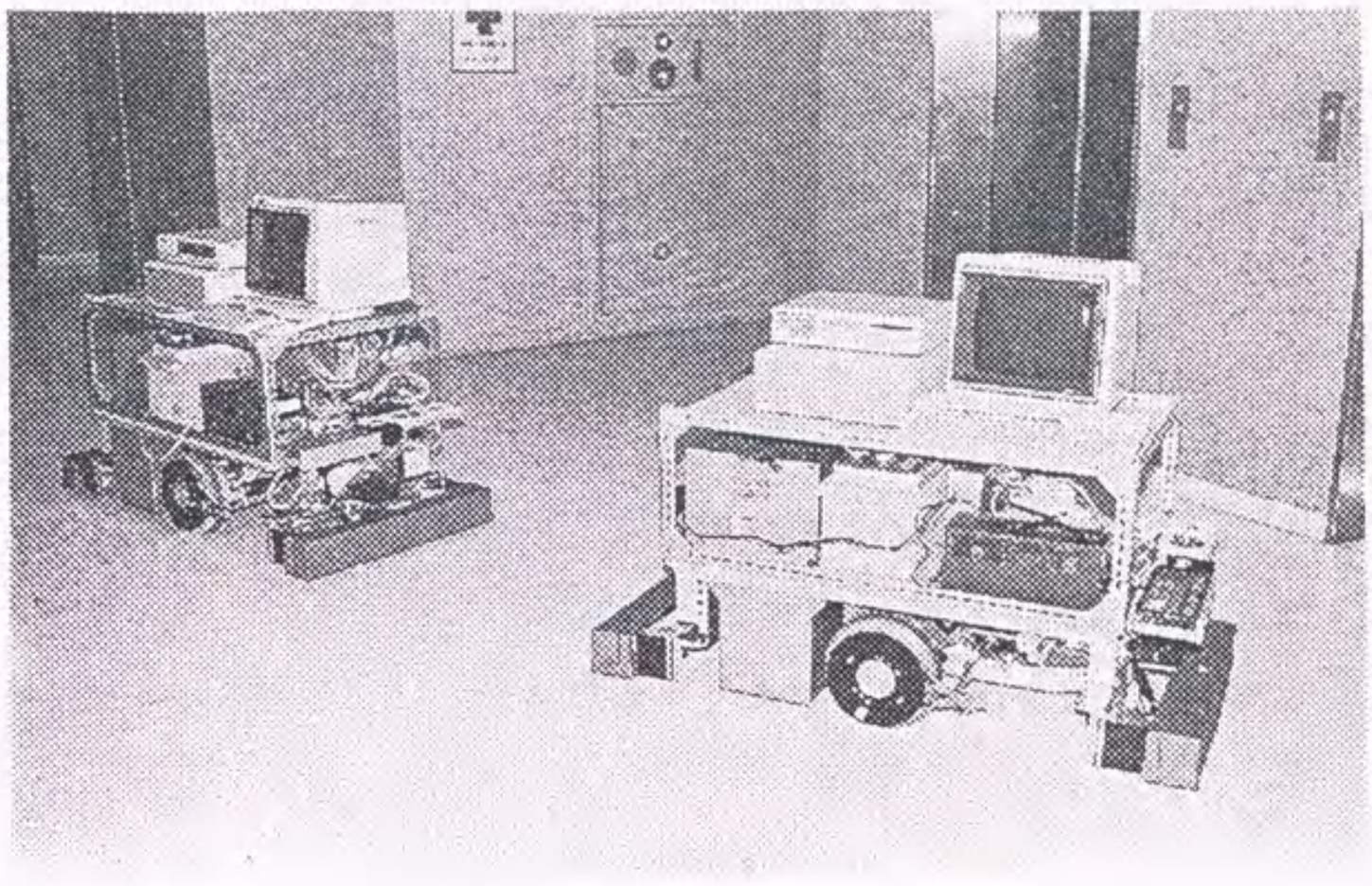


Fig. 2.1 Soft-linked vehicles

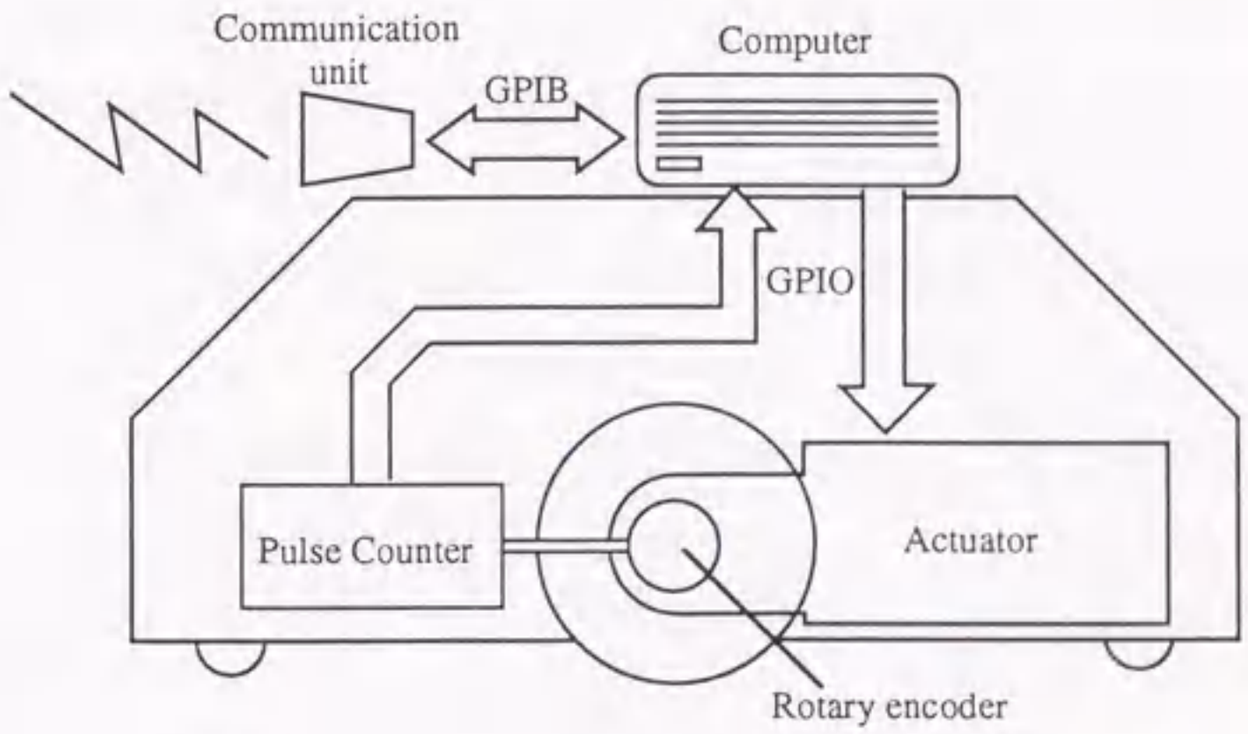


Fig. 2.2 Configuration of a soft-linked vehicle

と表すことができる．

ここで (x,y) は車両の代表点の位置，θ は車両の方位，v は車両の速度，v_r とv_l はそれぞれ右と左の車輪の速度で，$v = (v_r + v_l)/2$ である．したがって左右両輪の速度を測定し，その数値を用いて上式を積分することにより，車両の位置と方位を車上でリアルタイムに推定することが可能になる．

左右輪の速度を計測するために左右の駆動輪にロータリーエンコーダをとりつける．2個のロータリエンコーダからのパルスはカウンタで計測され，ディジタルインターフェース (GPIO) を経由して車載コントローラへ送られる．カウンタの計数周期はカウンタ上で設定し，この周期を車載コントローラの制御周期としている．そのために車載コントローラではカウンタからの割り込みで制御周期を処理している．

2．3．3　デッドレコニングの誤差修正法

車輪の回転角の積分によるデッドレコニングでは，車輪のスリップやタイヤの接地点の間隔 dの微妙な変化などの影響による誤差の累積が問題となる．内界センサーだけでこれを補正することはできないから，なんらかの外界情報を用いる方法が必要である．ここでは，画像処理を用いる方法について述べる[34][35].

この方法では，地上側にあらかじめ位置のわかっているランドマークとして，バーコードを印刷した円筒状のポール（サインポスト）を用意し，これを走行経路に沿って2 本ずつ組にして配置する．車載のテレビカメラでこれらのバーコードを検出することによって三角測量を行い，これを車両の絶対位置の較正に用いる．

ソフトリンクビークルシステムでは，たくさんの車両が車間距離を短く保って走行するため，デッドレコニング誤差の解消のために車両を停止させることができない．したがって，車両の走行中に誤差の修正をおこなうことが必要となる．そのため，Fig. 2.4 に示すような位置推定システムの構成とした．このシステムは a) 車輪回転角計測によるデッドレコニング，b) バーコード検出のための画像処理，c) これらの情報を統合する拡張カルマンフィルタの３つの部分からなる．

a) は前述の通りである．

b) は車載テレビカメラで取り込んだモノクロ画像から路上に置かれたバーコード状

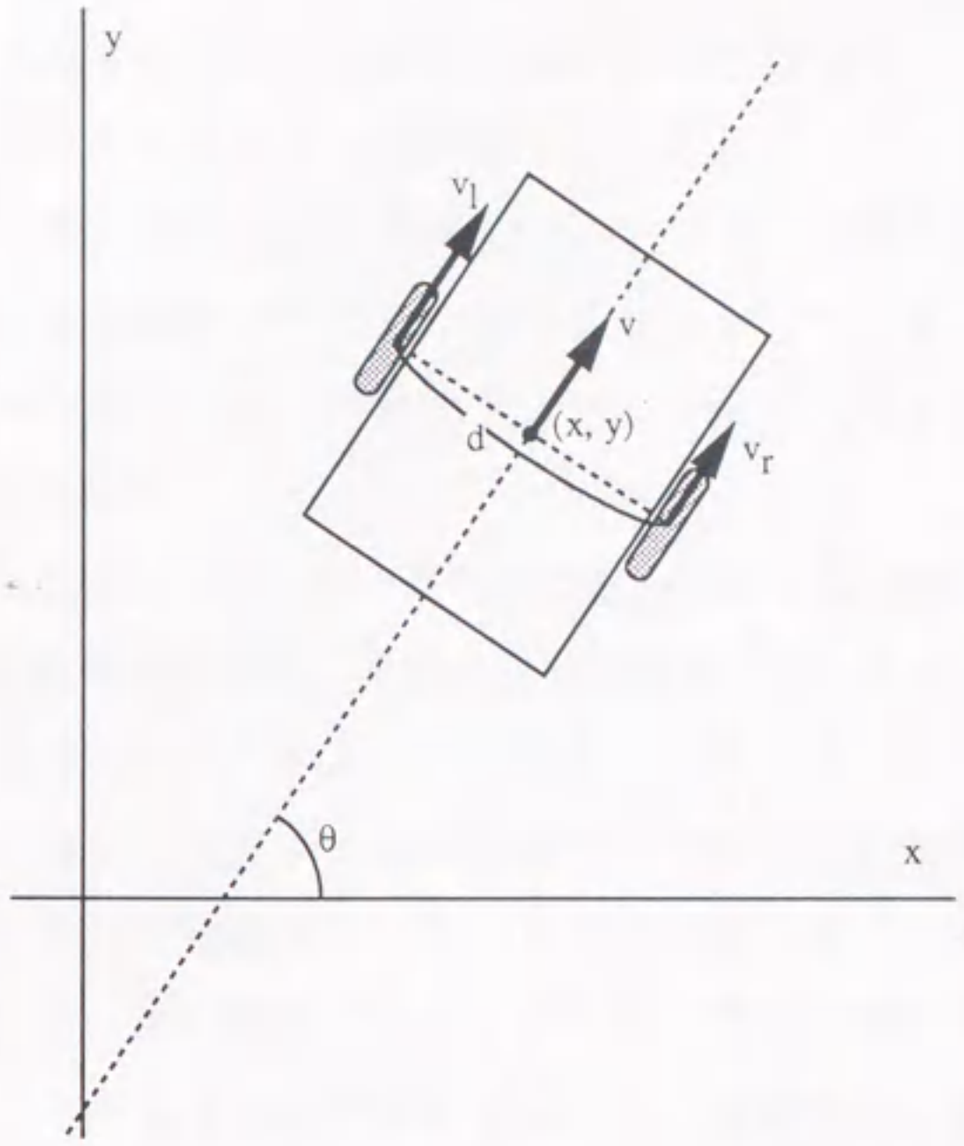


Fig. 2.3 Vehicle dynamics in a fixed reference frame

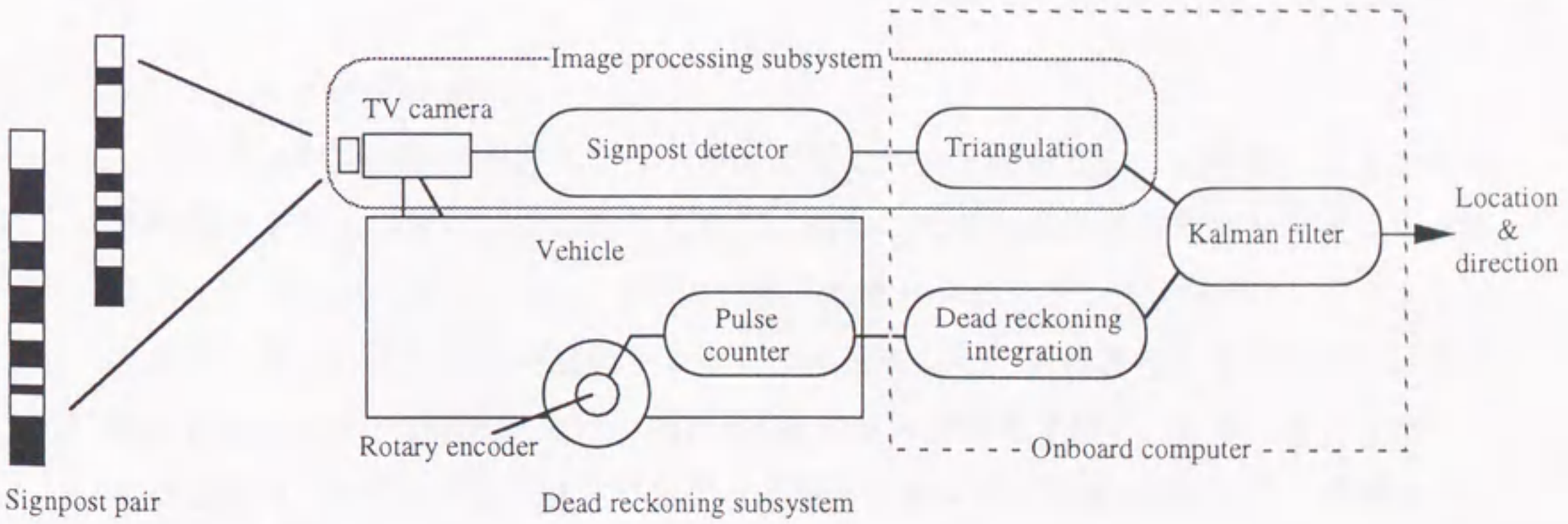


Fig. 2.4 Locating system by onboard image processing and dead reckoning

のサインポストを抽出する回路，および，抽出されたバーコードのコード番号，大きさ，および画像中の位置を測定する回路からなっており，コード番号からサインポストの絶対座標上の位置，大きさから車両とサインポストの間の距離，画像中の位置からサインポストのある方向がそれぞれわかるようになっている．画像中に2本のサインポストが検出された場合には，三角測量によって車両の現在の絶対位置および絶対方位角を知ることができる．

c) は a) b) の情報を組み合わせて，確率的に最も確からしい位置を推定するためのもので，車両の位置姿勢を状態変数，車両の運動方程式 (1) - (3) をフィルタのダイナミクスとし，車輪回転角をこのダイナミクスに対する入力，サインポスト観測による車両の絶対位置を出力とみたシステムに対しカルマンフィルタを構成したものである．これにより，長時間の推定が困難なデッドレコニングと，サインポストが2本抽出できないと観測不能となり，測定精度がサインポストまでの距離に依存してしまうサインポスト観測について，それぞれの欠点を補いつつ，連続的に正確な位置姿勢推定が可能となった．走行実験では，サインポスト対を3組含む複雑な経路（総延長約　50m）を走行して，位置誤差 10cm 以下，姿勢角誤差約1度という結果を得ている．

このような位置姿勢の推定は，ビークルが長時間走行する場合には必要不可欠なものであるが，以下のソフトリンク走行実験では簡単のためデッドレコニングだけを用いている．

2．3．4　車両間通信

車両間通信装置は，通信ユニットと送受信ユニットからなる．近距離にある2台の車両間で1対1の個別通信を行うために，通信のための媒体は指向性を考慮して赤外線（波長 950nm）とした．Fig. 2.5 に交信可能域を示す．この交信領域は，ソフトリンク状態を保ったまま車両が旋回できるように設定した．耐雑音性を増すために，赤外線による光通信の変調方式には，PSKとASKの組み合わせを用いている．また，データ伝送速度は，制御周期と伝送すべきデータ量とに基づいて9600 bpsとした．通信ユニットは車載コントローラとGP-IBで接続され，通信の制御とデータフレームの組み立て，分解を行う．データフレームには標準的な HDLC フレームを用いた．そのために通信ユニットは8 ビットのマイクロプロセッサで制御されている．送受信ユニットは，車両

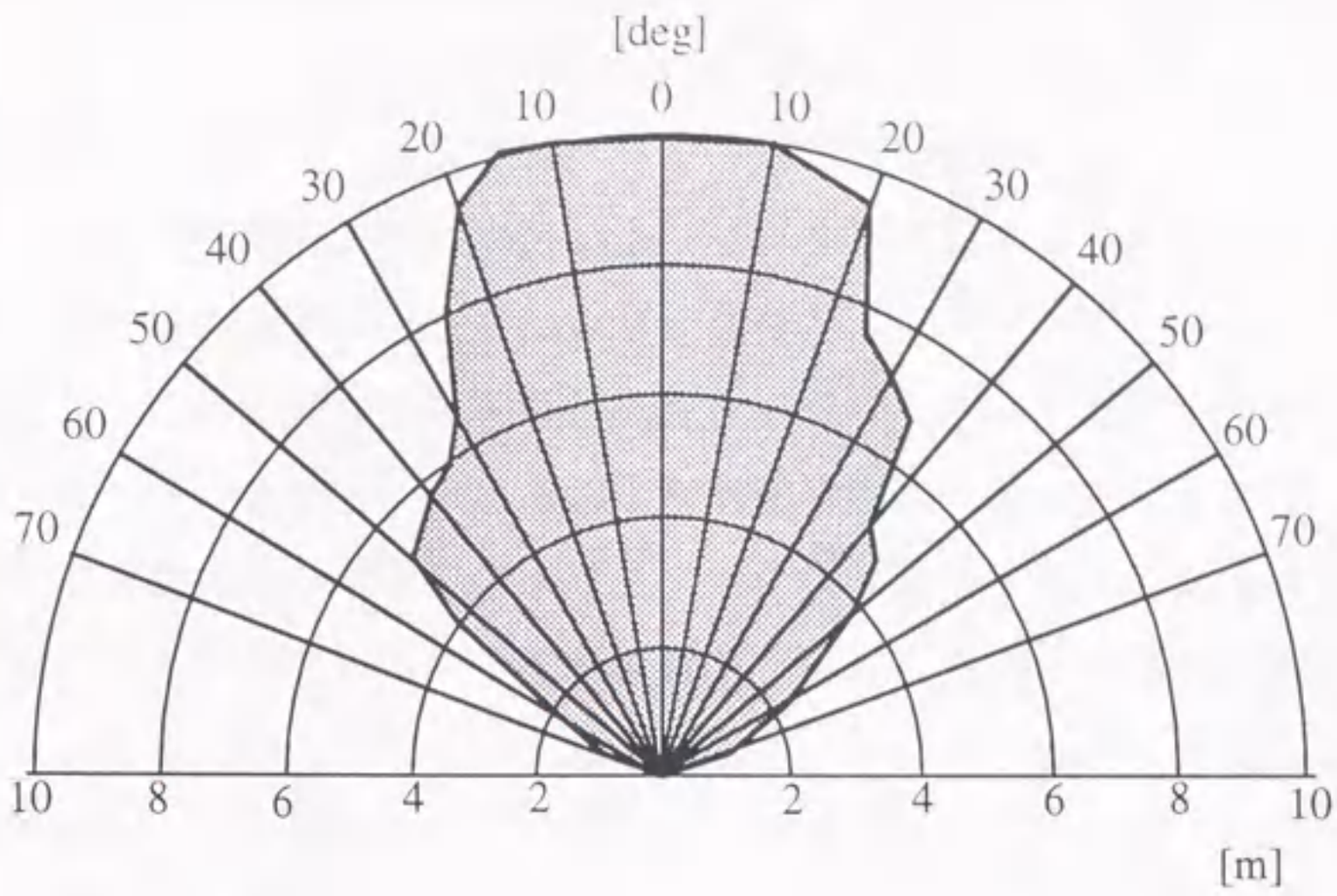


Fig. 2.5 A range of vehicle-to-vehicle communication

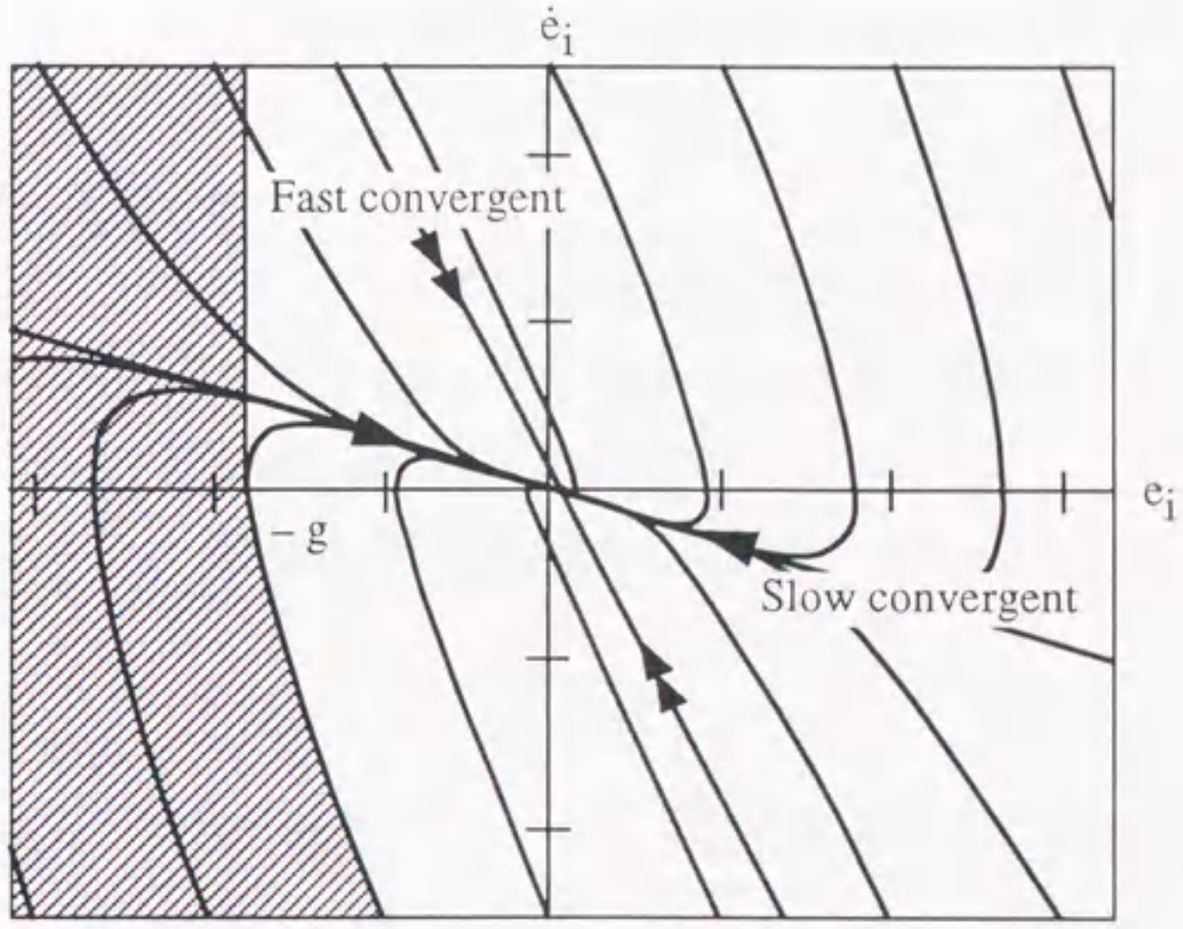


Fig. 2.6 Flows for poles –0.3 and –2 in the phase plane

間通信のために車両の前部と後部に装着する．Fig. 2.5 に示した交信領域を形成するために，送受信ユニットには赤外線発光ダイオードを 4 個用いた．

車両間通信によるソフトリンクは，ソフトリンクを勧誘するステップ，リンク後定常的にデータを送受するステップ，およびリンクを解消するステップからなる．車両間通信の制御のために車載コントローラと通信ユニットとの間のコマンドとレスポンスを用意した．これらのコマンドを用いたソフトリンクの手順はつぎのようになる．まず，ソフトリンクを勧誘するために 1 制御周期内に一度ポーリングフレームを発信する．発信する時刻は，車載コントローラが指定するが，他車からのポーリングフレームとの衝突を避けるために，その時刻はランダムに選ぶ．この間に他車からのポーリングフレームを受信したならば，自車からのポーリングを停止してレスポンスフレームを返信する．この段階からソフトリンクが開始される．以後，先行車と後続車の間で一定周期でデータを送受し，ソフトリンク状態を維持する．通信障害で一定時間内に通信ができない場合には，各通信ユニットを初期状態にして，再びポーリングから始める．ソフトリンクの解消にはコマンドを送受し，解消後，各通信ユニットを初期状態にする．

2．4　走行制御アルゴリズム

ソフトリンクビークルの追従走行制御を，速度制御と操舵制御に分けて考える．

2．4．1　速度制御

速度制御[36]は，ソフトリンク状態にある後続車において重要である．後続車が先行車に小さな一定車間距離で追従するように後続車の速度を制御する必要があるからである．同じダイナミクスをもつ車両群が同一直線上を走行しているものとし，先頭車から1,2,...と付番する．先行車 i の現在位置をx_i，制御入力をu_iとし，そのダイナミクスを

$$\ddot{x}_i = u_i - a\dot{x}_i, \quad i = 1,2,\cdots \tag{4}$$

とする．a は各車両に共通の定数である．設定車間距離 g からの車間距離の偏差を e_i とすると，

$$e_i = (x_i - x_{i+1}) - g, \quad i = 1,2,\cdots \tag{5}$$

となる．(4), (5) 式から位置偏差に関する方程式は

$$\ddot{e}_i = -a\,\dot{e}_i + u_i^*, \quad i = 1,2,\cdots \tag{6}$$

となる．ただし u_i^* は便宜上導入した制御入力で，

$$u_i^* = u_{i-1} - u_i, \quad i = 2,3,\cdots \tag{7}$$

である．先頭車に対する制御入力 u_1 は任意に選ぶことができる．k_1, k_2をフィードバックゲインとして，位置偏差と速度偏差をフィードバックすると，u_i^* は，

$$u_i^* = k_1 e_i + k_2 \dot{e}_i, \quad i = 2,3,\cdots \tag{8}$$

と表される．したがって，各後続車に対する制御入力は先行車に対する制御入力に依存する形で

$$u_i = u_{i-1} - k_1 e_i - k_2 \dot{e}_i, \quad i = 2,3,\cdots \tag{9}$$

となる．すなわち，この制御則を後続車で実行するためには，先行車から後続車への単方向通信でデータを伝送すればよい．ただし，車両間通信や車両上のでの処理のために生じた時間遅れはここでは考慮していない．

u_i (i=1,2...)から速度制御 v_{ci} を求めるためには，(4) 式を書き換えた関係式

$$\dot{v}_{ci} = u_i - a v_{ci}, \quad i = 1,2,\cdots \tag{10}$$

を用いる．

次にフィードバックゲインk_1,k_2を決めよう．車両間の追突を防ぐためには，あらゆる初期状態に対して位置偏差の応答の行き過ぎ量を十分小さくする必要がある．p_1,p_2 を正としてフィードバックゲインk_1, k_2をつぎのように定める：

$$k_1 = -p_1 p_2 \tag{11}$$

$$k_2 = a - p_1 - p_2 \tag{12}$$

このとき，(6)式と(8)式から構成される閉ループシステムの２つの極は，実軸上の $-p_1$ と $-p_2$ に配置される．従って初速度差のないときには追突は起こらない．初速度差が無視できない場合を検討するために e_i— $\dot{e}_i$ の位相面図を考える．Fig. 2.6 は $p_1 = 0.3$, $p_2 = 2.0$ のときの位相面図である．車両の定数 a は，試作した車両に基づいて a = 1.44 と

した．この位相面図の斜線部に初期状態があるとき衝突が生じる．衝突の生じる領域を狭めるためには，この例のようにひとつの極は大きく，他の極は小さく配置する必要がある．

速度制御に必要な偏差や偏差の時間微分（速度差）は，後続車上から直接計測するのではなく，すべて車両間通信で得たデータから求める．すなわち，先行車の位置と速度を (x_i, y_i), v_i：後続車の位置と速度を (x_{i+1}, y_{i+1}), v_{i+1} とおいて，旋回時も含めた偏差 e_i と速度差 $\dot{e}_i$ を

$$e_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} - g \qquad (13)$$

$$\dot{e}_i = v_i - v_{i+1} \qquad (14)$$

と定義する．(13), (14) 式から偏差と速度差を後続車上で計算し，速度制御を行う．ただし，２台の車両が同一直線上にないときは (13), (14) 式は近似式である．

２．４．２　操舵制御

追従走行制御における操舵制御の目的は，先行車の走行軌跡を後続車に辿らせることにある．このために車両の舵角制御アルゴリズムには，単独で走行する場合でも，ソフトリンク状態で走行する場合でも，地点追従法[37]を用いる．このアルゴリズムは地上座標系における現在の車両の位置 (x_0, y_0) と方位 θ_0，および，対象としている目標点の位置 (x_i, y_i) とそこにおける車両の予定方位 θ_1 に基づいて現在車両がとるべき操舵角を与える．したがって各車両が自車の位置と方位のデータを常時もっていることを前提としている．

Fig. 2.7 に示すように，ξ–η 座標系を，現在の車両の位置が原点，方位が０（ξ 軸方向）となるように地上座標系 (x-y 系）を平行移動，回転してつくる．ξ–η 座標系で現在対象としている目標点を（ξ_1, η_1 ）とし，その地点における車両の予定方位を ϕ_1 とする．このとき，操舵のための差動操舵方式の車両の左右輪速度 v_r, v_l は

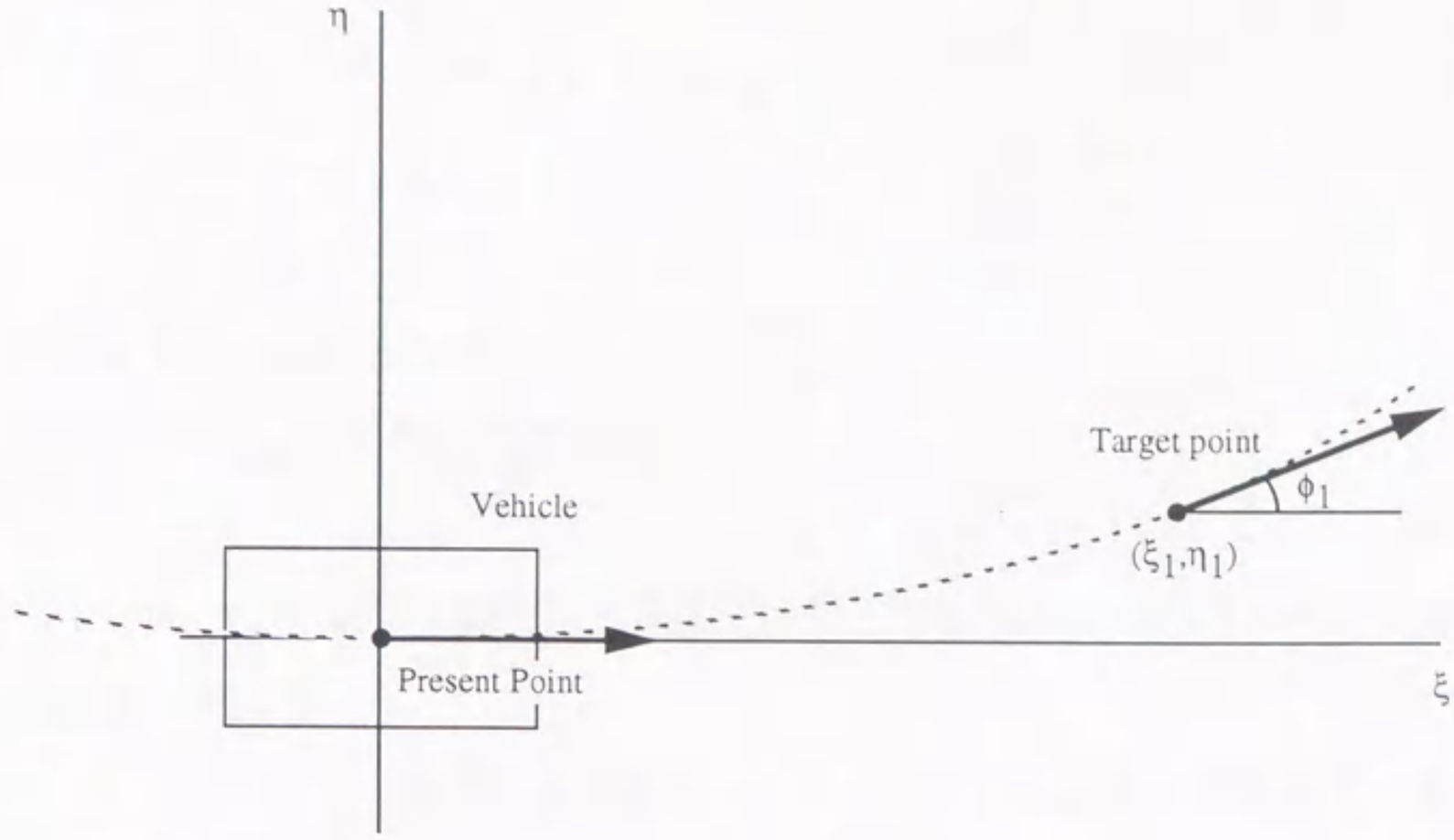


Fig. 2.7 Design of a steering control algorithm

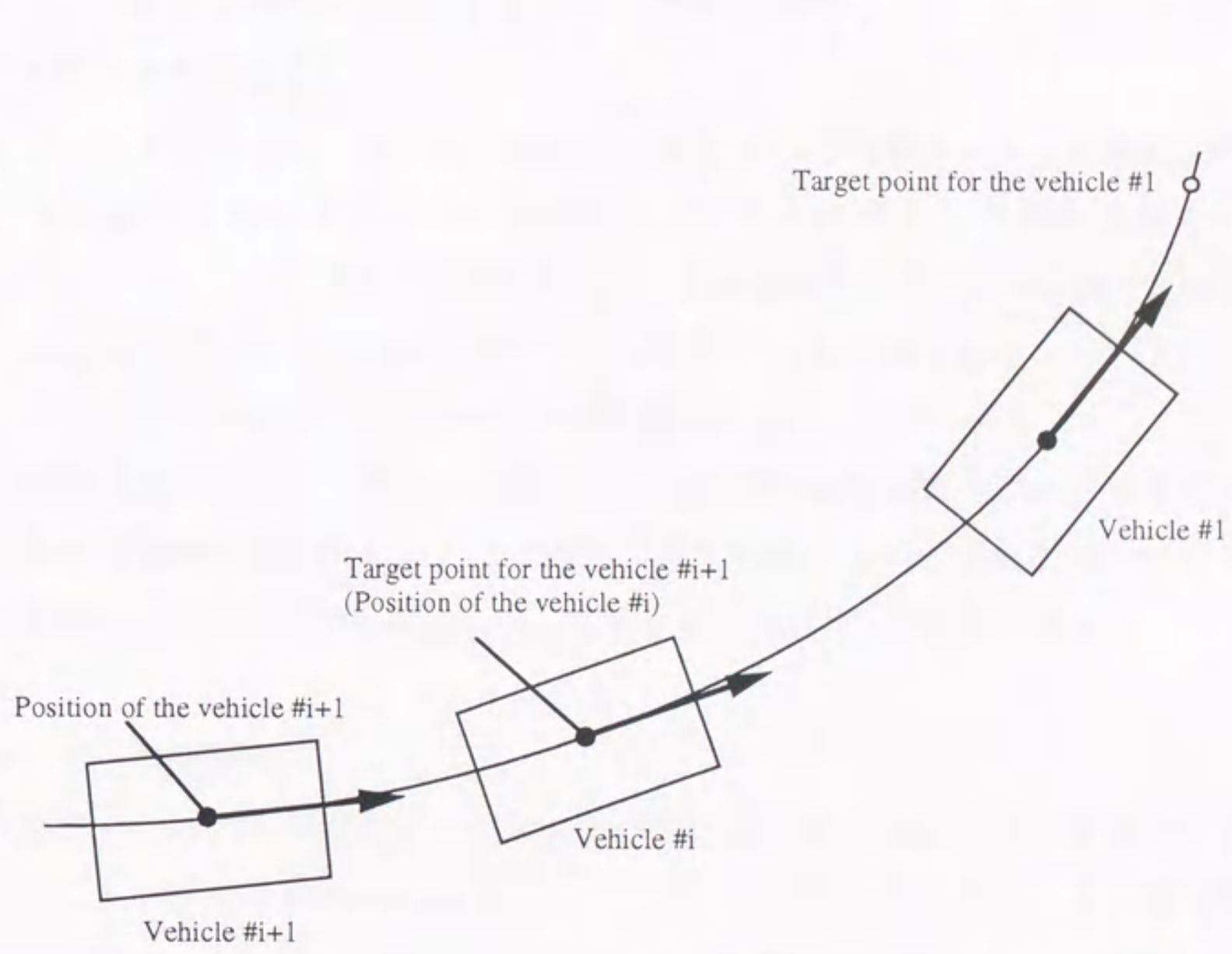


Fig. 2.8 Steering control when vehicles are soft-linked

$$v_r = v_c + \Delta v \tag{15}$$

$$v_l = v_c - \Delta v \tag{16}$$

で与えられる．ここで

$$\Delta v = \frac{v_c(3\eta_1 - \xi_1 \tan \phi_1)\, d}{\xi_1^2} \tag{17}$$

であり，v_c は前節の速度制御で決定された車両の速度，dはトレッド（車輪接地点間の距離）である．

このアルゴリズムでは，車載コントローラはあらかじめ地図をもっていることを前提としている．外部から与えられた目的地に対して決定した経路を目標点列で表すことによって車両は目的地まで自律航行することができる．

各制御周期ごとに車両の現在位置と方位に基づいて，現在対象としている目標点の位置とそこにおける車両の予定方位を ξ–η 座標系に変換し，(15) - (17) 式から操舵のための左右輪速度を計算する．

ソフトリンク状態では，先行車は地図上に設定された目標点によって操舵制御を行うが，後続車は直前の先行車の現在位置と方位を目標点として操舵制御を行う．Fig.2.8　にソフトリンク状態での操舵アルゴリズムを示す．そのために先行車の現在位置と方位は車両間通信で各制御周期ごとに後続車に送る必要がある．このアルゴリズムによって先行車の軌跡を後続車が辿る同轍制御を行うことができる．

走行制御決定の手順は，まず，前節で述べた手順で速度制御を求め，つぎにその結果をもちいて操舵制御を求めることになる．速度制御の決定と操舵制御の決定の間に相互干渉がないことがここに提示した走行制御アルゴリズムの特長である．

2．4．3　車両間通信

走行制御のための車両間通信は，各車両の制御周期(655.36ms）ごとに行う．追従制御のために先行車から後続車へ送られるデータは，上述したように，先行車の現在の位置 (x,y)，方位 θ，速度 v，制御入力 u である．後続車から先行車へは ACK/NACK が返される．各車両の制御周期は必ずしも同期していないために，先行車からのデータを後続車で利用するとき，遅れが生じる可能性がある．

2．5　実　験

車両2台を用いたソフトリンク走行実験を行った．実験を行った場所は，屋内（研究所の実験室と廊下）である．車載コントローラのプログラムは HP9000 の BASICで記述した．以下では，2台の車両を用いて直線走行と車線変更のソフトリンク走行実験について説明する．

（1）車載コントローラでの処理

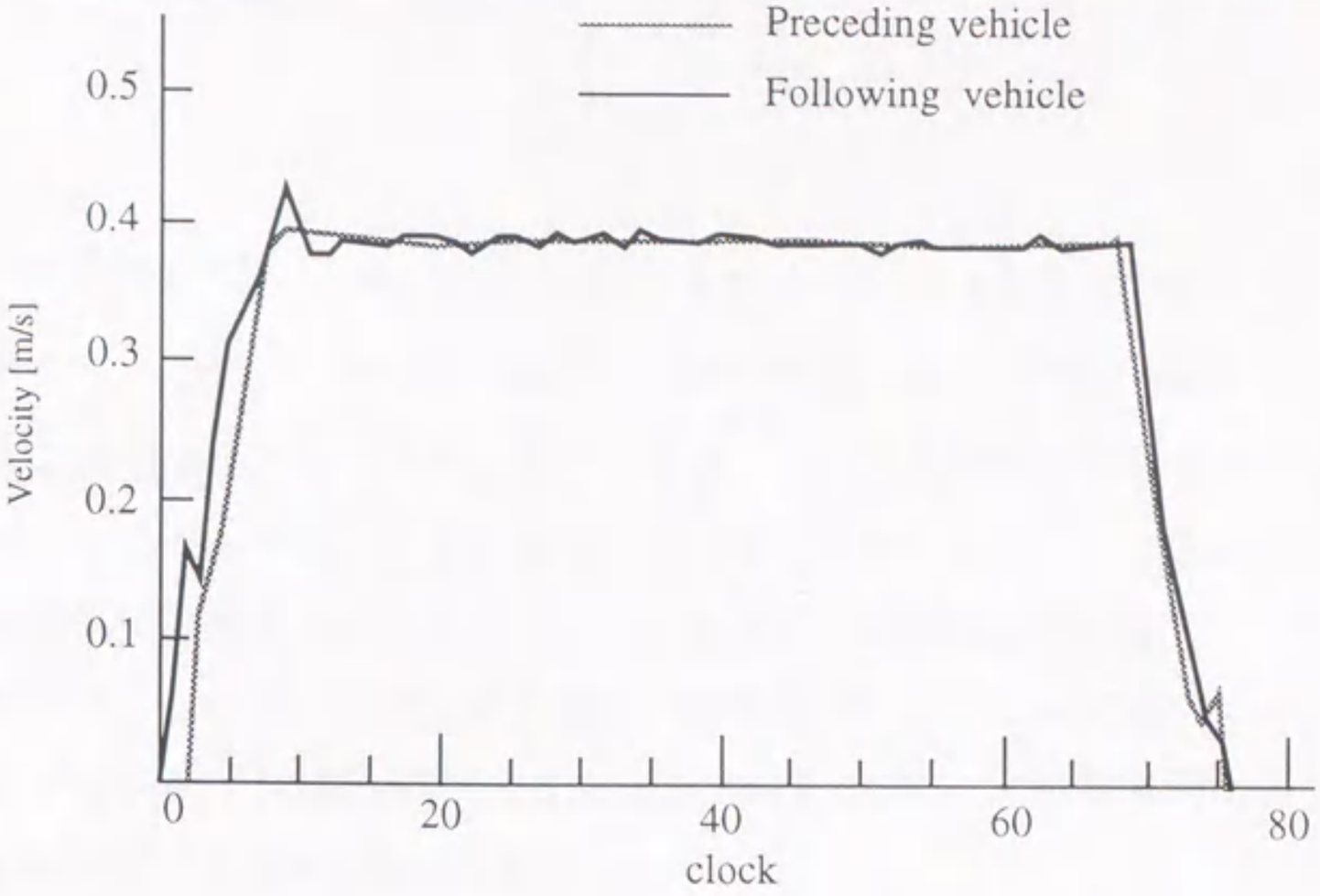
先行車の車載コントローラの処理手順は以下の通りである．2．3．4節で述べた手順で後続車との間の通信をあらかじめ確立しておく．パルスカウンタからの割り込みによって制御周期が決定されるが，制御周期ごとに，まず左右輪の回転数から左右輪速度を求め，(1) - (3) 式を1刻みだけルンゲークッターギル法で積分して自車の位置と方位を求める．これらのデータを後続車に送出し，後続車からの ACK/NACK を待つ．この間に，2．4節で述べたアルゴリズムで速度制御と舵角制御を行う．先行車の制御入力 u_i は，速度パターン v* を与え，(10) 式から決定する．後続車から ACK/NACK を受信したならば，つぎの制御周期に入る．

一方，後続車でのプログラムは，先行車とやや異なり，コンピュータはパルスカウンタからの割り込みと通信ユニットからの割り込みを待っている．パルスカウンタからの割り込みがあれば，先行車と同様に自車の位置と方位を計算し，最新の先行車からのデータを用いて，先行車と同様に速度制御と舵角制御を行う．また，通信ユニットからの割り込みがあれば，先行車からのデータを入力して記憶内容を更新し，ACK を送出する．

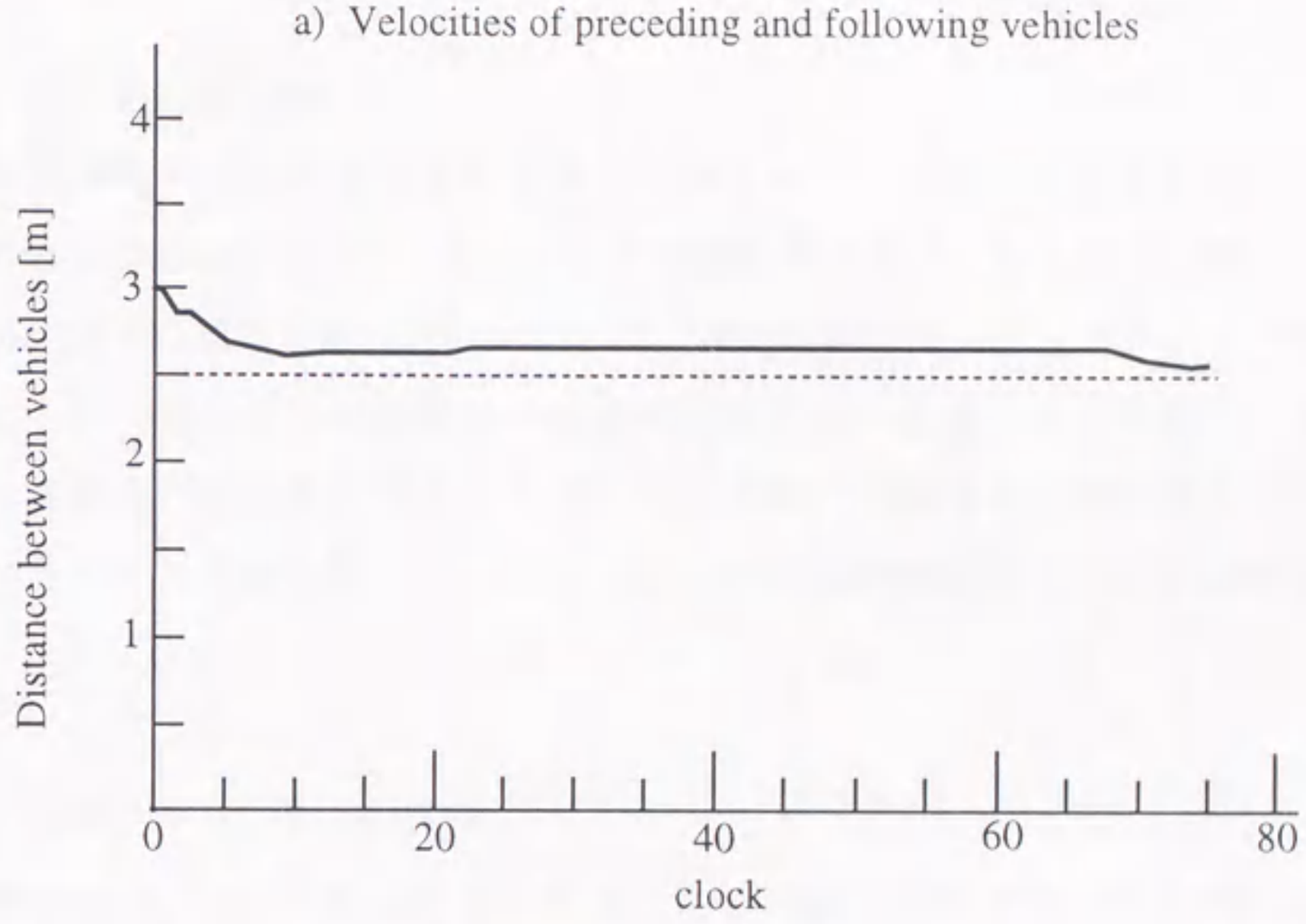
これらの実験でのプログラムでは，NACK　受信や通信エラー時の処理は行わないでつぎの制御周期に入るようにしたが，実験中通信障害はまったく発生しなかった．

（2）直線走行実験

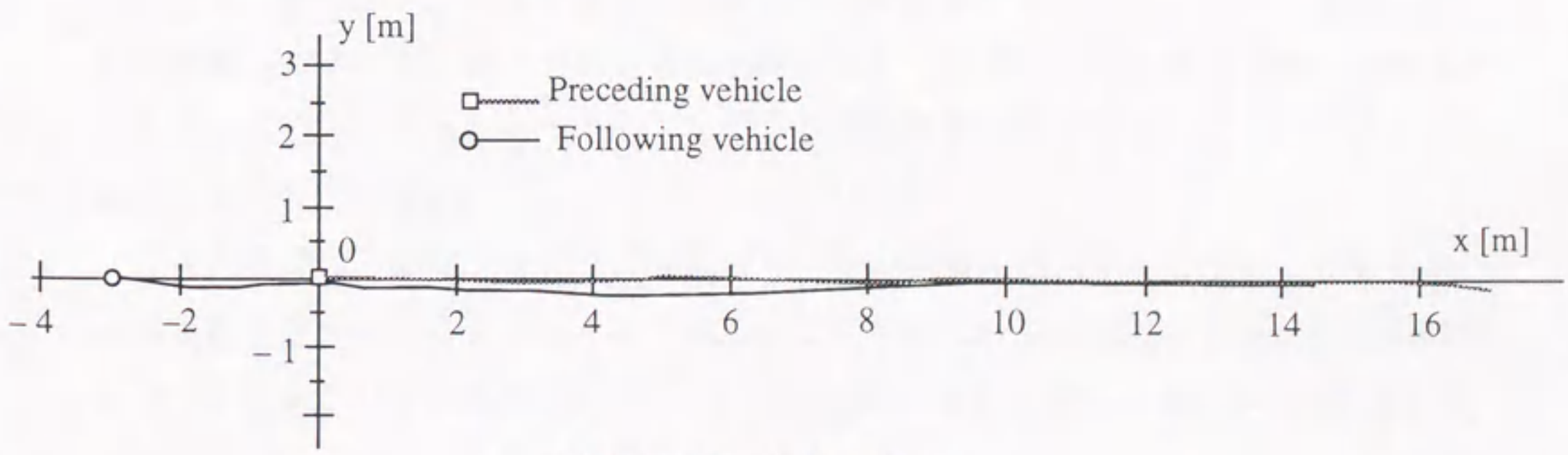
直線走行時の実験結果を Fig.2.9 に示す．これらの結果は，各車両上で計測したデータに基づいて作成した．車両代表転換の設定距離（設定車間距離） g を 2.5m，初期状



a) Velocities of preceding and following vehicles



(b) A distance between the vehicles



(c) Trajectory of the vehicles

Fig. 2.9 Experimental results of driving along a straight line

態での車間距離を 3.0m とし，(11), (12) 式で $p_1 = 0.3$, $p_2 = 2$, $a = 1.44$ としてフィードバックゲイン k_1, k_2 を求めた．ほぼ，一定車間距離で走行しているのがわかる．車間距離に定常状態でオフセットが生じているが，これは先行車と後続車での制御周期が同期していないためである．すなわち，２．４．１節で設計した速度制御アルゴリズムは，連続系に対するものであり，これは各車両の制御周期が同期していることを前提としている．しかし，実際には各車両の制御周期は必ずしも同期していない．各車両の速度が 0.4 m/s で制御周期が 655.36 ms であるから，この間に車両は約 26 cm 走行し，この距離がオフセットの上限値となる．

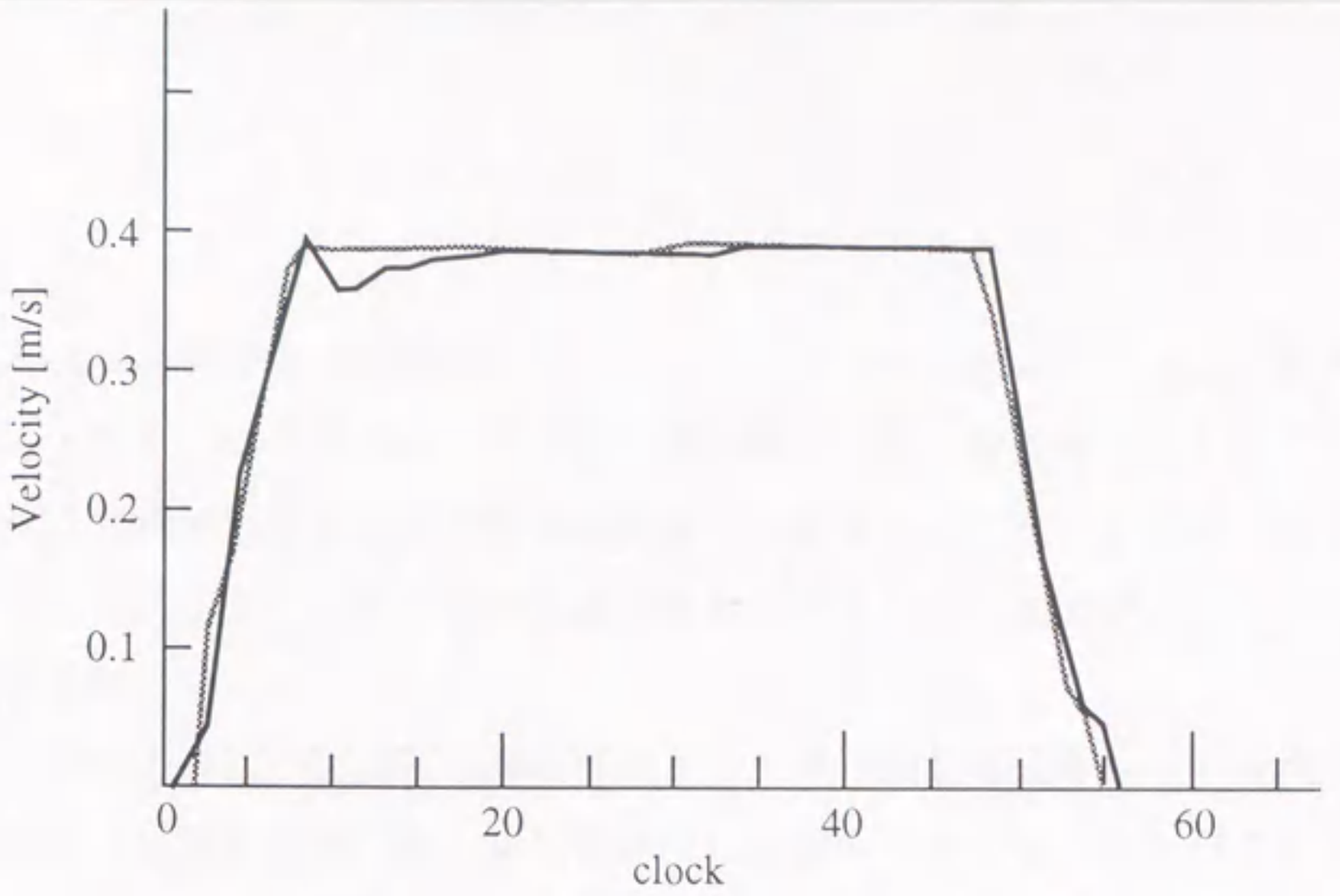
（３）車線変更走行実験

車線変更走行では直線走行と同じ速度制御を行い，先行車の舵角制御のために車線変更経路の目標点列を用いた．Fig.2.10 に実験結果を示す．設定車間距離 g を 2.5 m とし，初期状態での車間距離も 2.5 m とした．走行軌跡はデッドレコニングで測定したものではあるが，後続車が先行車に一定車間距離で同一軌道上を追従走行しているのがわかる．直線走行時の走行軌跡にも生じているが，後続車が出発直後右方に軌跡がはずれているのは，車輪に取り付けたエンコーダの回転開始時にパルス数が正しく計数されないことによる．

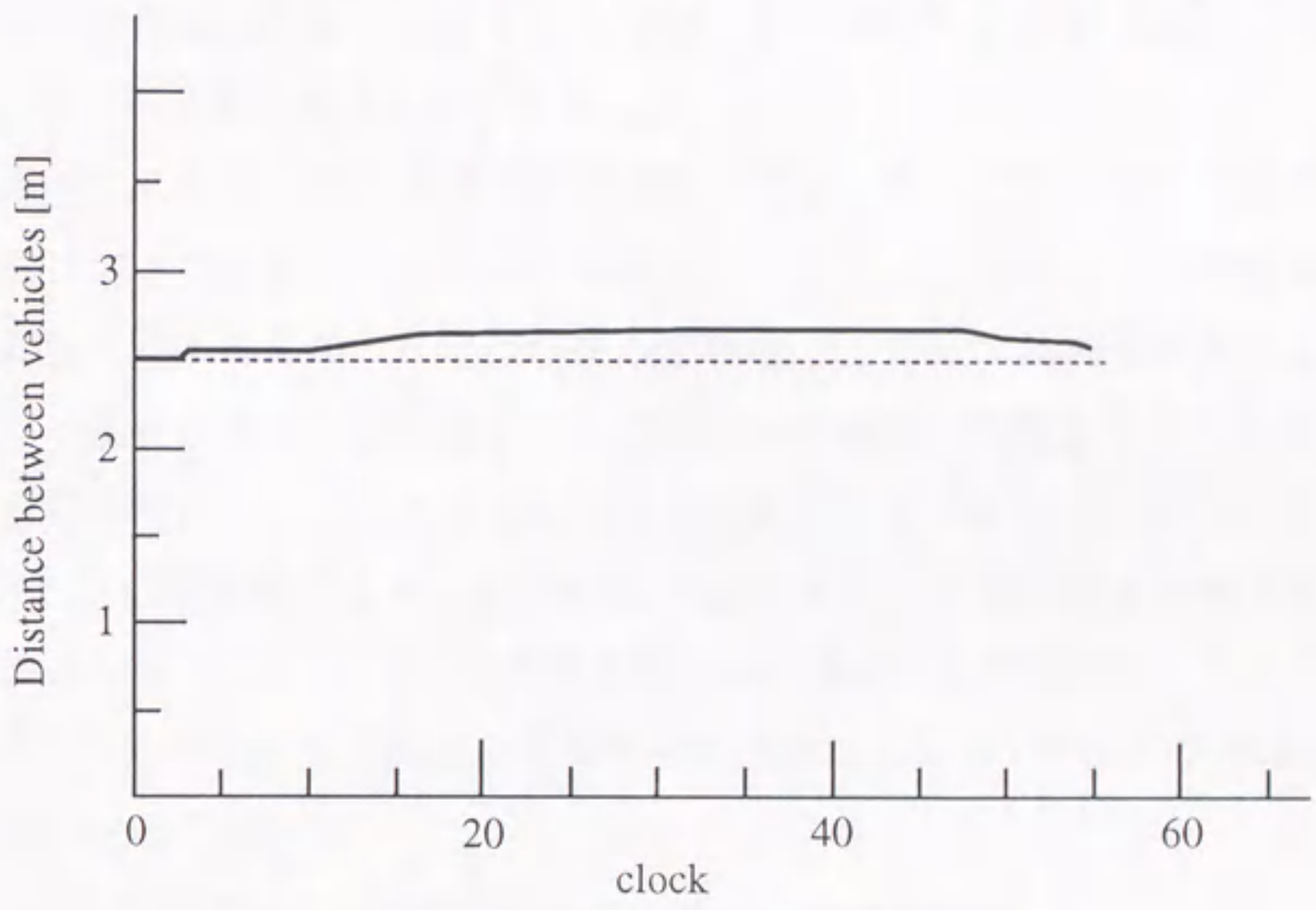
２．６　制御アルゴリズムの拡張

これまでのソフトリンクビークルシステムの設計・解析では，車両間の通信や計算には遅れがなく，すべての車両で同期した制御周期が確保されていることを前提としていた．しかし，システムに含まれる車両の台数が多くなると，これらを実現することは困難になる．言い替えれば，この制御アルゴリズムで扱える車両台数には制限があるということである．そこでもっと大きな車両群を制御するためにアルゴリズムの拡張をおこなう[36][38].

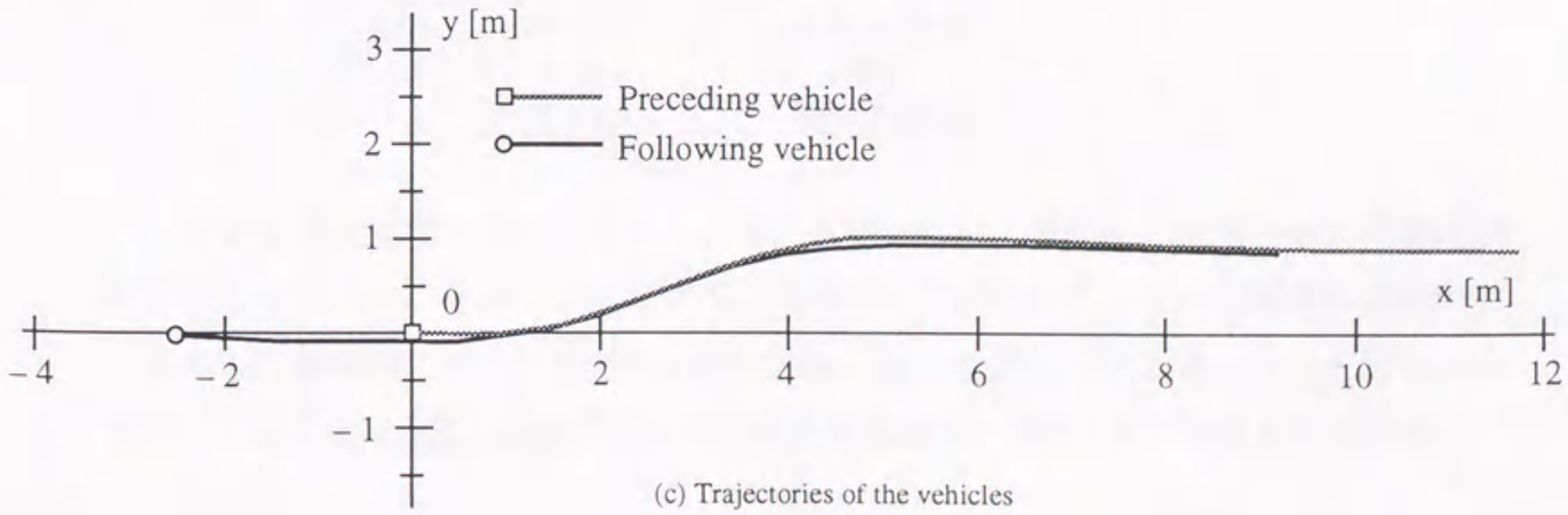
この方法では，制御アルゴリズムにいくつかの階層を導入することによってシステムの規模を拡大しようとするものである．ここではこれらの階層ごとに制御則の時間スケールを変えるマルチタイムスケール法[39][40]の考えをもちいて全体の制御系を設



(a) Velocities of preceding and following vehicles



(b) A distance between the vehicles



(c) Trajectories of the vehicles

Fig. 2.10 Experimental results of a lane changing maneuver

計する．

まず，これまで述べてきた制御則によってソフトリンク状態になっている車両群がいくつかあるとする．これらをまとめて第１層と呼ぶ．第１層に属するそれぞれの車群のなかでは，車間距離収束の速度が比較的速くなるようにゲインを定めておく．このようにすることによって，第１層の各車群はあたかも１台の車両であるかのように扱うことができる．

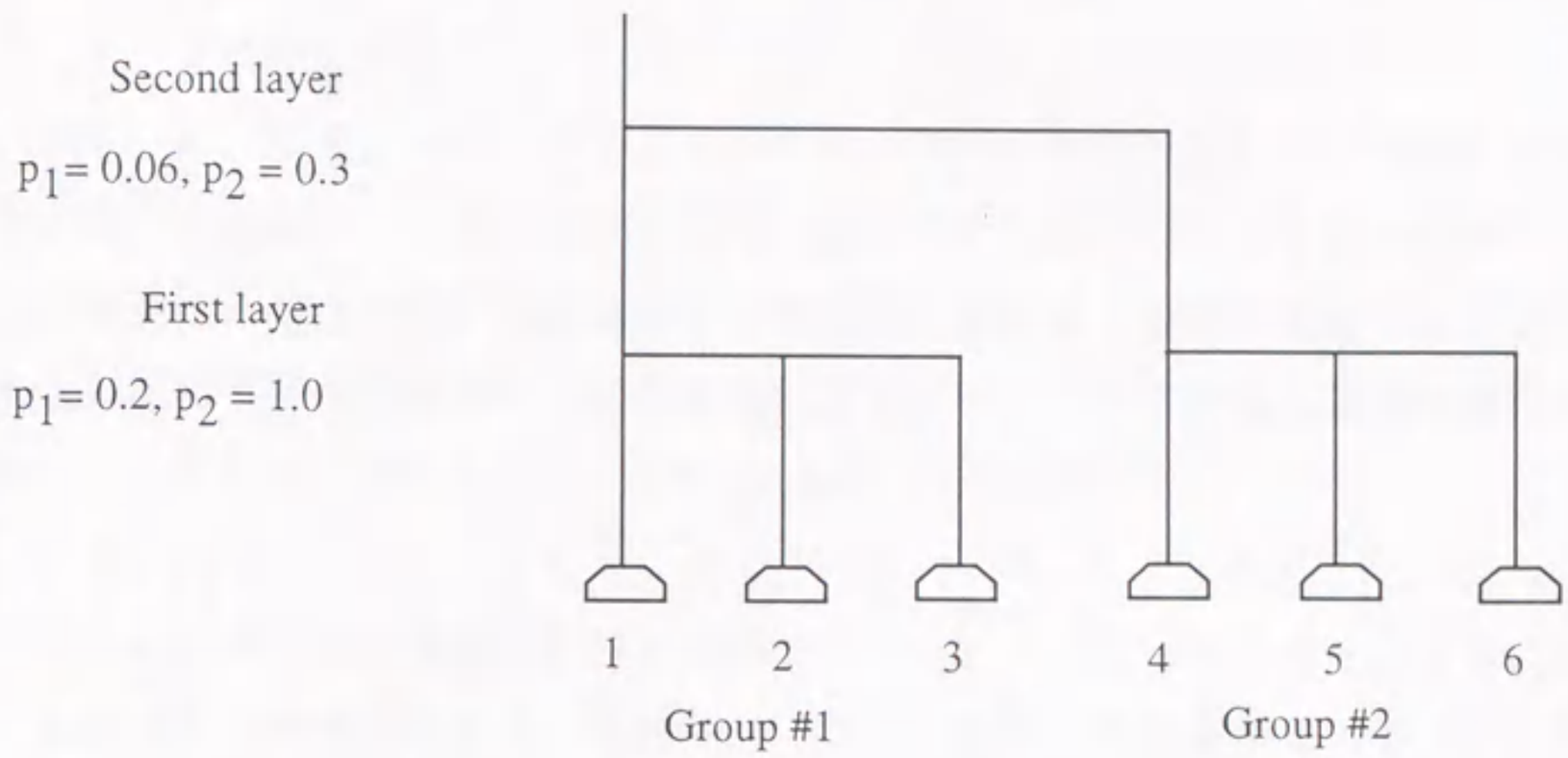
つぎに第１層の各車群の先頭車を構成要素として第２層を考える．この層に対しても同様な制御則を適用すれば，第１層の車群間の距離を一定に保つことができる．ただし，第２層では車群間距離の収束速度は第１層よりも遅くなるように設定され，また，第２層の車群間の距離も第１層のそれよりも大きく設定する必要がある．同様にして，第３層，第４層を考えることもできる．

このような階層化により，大きな車両群の制御が可能になる．ここでは，各階層ごとに異なる通信メディアが利用できることを仮定している．したがって，通信にかかるコストは層が増えるごとに増大するのはやむを得ない．ただ，高次の階層になればなるほど，収束の遅いシステムとなるから，必要となる通信の周期もゆっくりでよい．ただし，第１層以外ではいくつかの車両を飛び越えて通信ができなければならないし，各層の通信はそれぞれ独立に行える必要がある．Table 2.1 に利用可能な通信メディアの例を示す．また，Fig. 2.11 に２つの階層を導入した場合の走行制御のシミュレーション結果を示す．このグラフの縦軸は各車両の絶対位置（上から１号車，２号車の順），横軸は時間をあらわしている．

Table 2.1 Control layer and communication medium

層	サンプリング周期	通信メディアの例	台数
1	500 msec	赤外線通信	～５
2	1 sec	無線	～２０
3	1 min	路車間通信	～１００

このような大規模な車群システムを考える場合には，車両間の通信の遅れの問題が重要になってくる．そのためにはまず，各通信メディアによってどの程度の遅れが生じるのかを実験的に押さえておく必要がある．生じる恐れのある遅れの最大値がわかれば，それと走行速度を勘案して，設定車間距離および階層ごとの設定車群間距離に



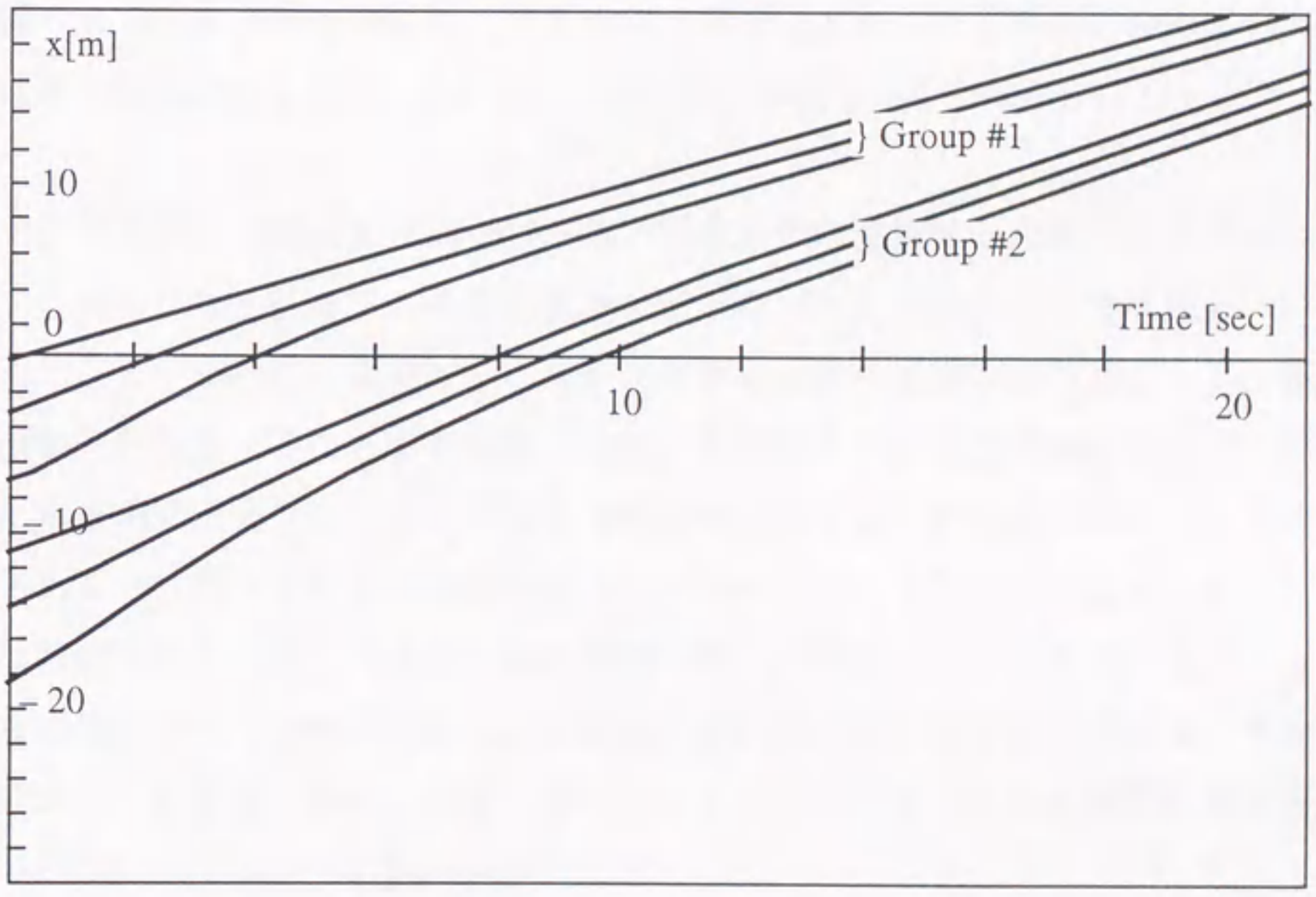


Fig. 2.11 Simulation results of layered control strategy

十分なマージンをあたえることができる.

2．7　まとめと考察

本章では，まず，ソフトリンクビークルシステムの構成について説明し，つぎに速度制御と舵角制御からなる走行制御アルゴリズムを示した．この走行制御アルゴリズムの特長は，速度制御と舵角制御をほぼ独立に求めることができる点にある．そして単独車両の自律走行実験と２台の車両によるソフトリンクの基本走行実験について述べた．これらの実験によって，以下の点を明らかにした．

・ここに提示した速度制御と舵角制御アルゴリズムによってソフトリンクによる車両の追従走行が可能である．実験では，２台の車両が速度制御によって車間距離の初期偏差をなくして一定車間距離を保って走行することができ，また操舵制御によってこれらの車両が同一軌跡上を走行することができた．
・赤外線による車両間通信装置は，車両の停止時だけでなく，直線走行時や旋回時にも安定して個別通信を行うことができ，通信障害が起こることはなかった．

ここで，もう少し一般的な見地から，集中制御型の均質機械システムのモデルとしてのソフトリンクビークルシステムを見直してみたい．このシステムでは地上のコントロールセンターの存在を仮定し，これがシステム全体の運行計画を決定する．決定された情報（目的地とそこまでの経路，到着予定時刻など）は路車間通信によって車両群の先頭車両に伝えられ，残りの車両は車両間通信によって先頭車両に追従する形で運行される．このようにここではコントロールセンタ，あるいは先頭車を起点とする集中型の制御方式，トップダウン的な情報の流れが明瞭である．そこでこのようなシステムの構造が，第１章で指摘した大規模システムの抱える３つの問題点，すなわち，耐故障性，拡縮性，設計・生産・保守のコストについて，どの程度これらを解消できたかについて，若干の考察を加えてみたい．

まず，耐故障性についてであるが，オンライン（走行中）の故障については，本質的な問題の解決をみていないと言わざるを得ない．故障としては，おおきくわけて制御・通信系の故障，センサー系の故障，アクチュエータ系の故障の３つが考えられる．

制御・通信系の故障についてまず考えると，２台の車両をもちいた小規模な実験では，通信の信頼性は十分なものであったが，車両の台数が増えた場合，一部で通信に失敗が生ずることも考えられ，そうした場合は失敗が生じたリンク以降の車両の制御が正常におこなわれなくなることが心配される．通信のとぎれが瞬間的であれば，そこで生じた誤差は３．４．１節の速度制御則の安全領域で吸収されると考えられるが，車両間通信の不具合，または車載計算機の不具合等によって制御・通信系の故障が継続的に生じた場合，その車両より後ろの車両すべてに影響が及ぶことは避けられない．特に，先頭車両に故障が生じた場合は，車両群全体に影響が及ぶ．したがって，こういった走行中の故障に対処するためには，制御・通信系を多重化するとともに，安全に走行を停止するための別の方法，たとえば，レーザー車間距離計などによる車間距離の直接計測などが必要になると思われる．しかし，車両の位置・速度等の計測システム，レーザー車間距離計などのセンサー系の故障に対しても，これらを多重化し，センサーシステムの信頼性をあげておく以外に対処の方法がない．また，車輪の駆動系や操舵系などのアクチュエータ系に関わる故障では，さらに対処が難しい．アクチュエータ系の故障に対しては，これを何らかの方法で検出し，走行を続けることが困難と判断した場合は後続車に緊急停止信号を送って故障車以降の車群を止めるなどの方策をあらかじめ埋め込んでおく必要がある．

ただ，これらの故障が走行に直接影響しない軽微なものであれば，走行後に車両の入れ替えを行えばよく，その意味ではシステムの均質性は全体のパフォーマンス維持に有利である．車両役割の入れ替えをおこなうためには，先頭車と後続車では駆動ソフトウェアが異なるため，各車両にはあらかじめどちらにも対応できるように両方のソフトウェアを搭載しておく必要がある．

次の拡縮性については，速度制御・舵角制御の方式からわかるように，車両群を拡大することはきわめて自然に行われる．すなわち，新しくつけ加わる車両は車両群の最後尾の車両から情報をもらうだけでソフトリンク走行を開始できる．逆に車両群の最後尾車両が車両群から離れることも同様であるが，この場合はソフトリンク状態を解消した後は，独自に目的地までの走行制御を行わなければならなくなる．もし，その車両にさらに後続車がある場合は，車両群は２つの車両群に分断され，それぞれが新しい目的地に向かうことになる．さらに２．６節で述べたような階層を導入すれば，

大規模な車両群の制御が可能となる．このように，ソフトリンクビークルシステムは，おおきな車両群を先頭車の制御という比較的少ない情報処理で柔軟に運用する能力をもっている．

最後の設計・生産・保守の問題については，システムの均質な構造がもつ一般的な有利性を指摘するにとどまるが，交通・輸送システムが高い稼働率とリアルタイム性を要求されることを考えると，同じ車両を多数用意しておけばソフトウェア次第で多様な要求に答えられる本システムの構成はメリットが多いと考えられる．

第3章　均質ユニットによる機械システムの構成

3．1　はじめに

前章では均質な機械システムで集中型の制御をおこなうものの一例としてソフトリンクビークルシステムを考えた．この章からは，均質ユニットによる機械システムの構成手法について考察する．

ソフトリンクビークルシステムとこれから説明する均質ユニット「フラクタム」によるシステムは，1．8節でも述べたとおり，空間的次元，幾何学的拘束条件，要素の大きさ，実用を目指すかどうか，などさまざまな違いがある．しかし，もっとも大きな違いは，その制御方式の考え方の違いである．ソフトリンクビークルシステムが，コントロールセンタあるいは先頭車両が全体を支配する集中制御方式であったのに対して，ユニットシステムではそのような全体を管理監督する部分をまったくもたない完全な分散制御方式をとっている．このため，制御方式の設計は集中型のときほど明示的ではなく，望みの機能をいかにして全体に埋め込むかというところが設計論の主題となる．言い替えれば，多数のユニットが互いに情報交換しながら，全体として協調して，整合性のある動きをさせるための方法論が必要となる．

本研究においては，実際の機械システムを構成することを目的とするため，構成上の徹底的な単純化と自律化にもとづいたアプローチを考える[41][42]．多数のユニットでシステムを構成するため，ユニット単体の機構的な複雑さを極力取り除き，均質な構成要素の協調という本質的な部分に焦点を絞って研究を進めるためである．

この手法の要点は以下のようにまとめられる．

・機械システムの全体をユニットの集合体によって構成する．ユニットは均質，すなわち各ユニットのハードウエア・ソフトウエアは互いに同等な構造とする．
・各ユニットは簡単なアクチュエータを持ち相互の結合位置を変更できる．
・各ユニットは簡単な情報処理機構をもち，自律的に判断を行える．
・各ユニットは簡単な通信機能をもち，結合中の隣接ユニットと情報交換が行える．

このような構成には次のような利点がある.

・機械システムは1種類のユニットだけで成り立つから，故障ユニットを他の任意のユニットで置き換えることができる.
・ユニットの結合を可変とすることで，全体の形状や機能を環境に合わせて変更することができる.
・生産する際には一種類のユニットを大量生産すれば良く，また組立はユニット群が自律的に実行できる.

当然，このような極端な均質化にはいくつか不利な点もあることを認めなければならない.

・すべての部品に情報処理機能やアクチュエータを組み込むことは冗長かつ非効率である.
・均質なユニットだけで，現在利用されているような機械を作ることは難しい.

有用な機能をもつ機械をつくるためには，本質的には非均質な部品が必要であるという議論もあり，それはたしかにその通りである．しかし，だからといって，均質な部品だけで機械を作ることが無意味であるとはいえない．なぜなら，均質な部品をあつめて，その部品よりはずっと大きなサブシステムを構成し，そのサブシステムを部品とするという考え方も成り立つからである．部品は均質でもその集め方をいろいろ変えることにより，いろいろなサブシステムを作ることができるとすれば，これで非均質な部品ができたことになる.

伊藤によれば，生物などの高度に組織化されたシステムは，均質な階層と非均質な階層の積み重ねによってできているという[17]．このようなシステムでは，均質な要素が多数集まっている階層（a層）と，異質なものが集まってできている階層（b層）があり，これらが交互に現れる．たとえば，生命のシステムは，（1）タンパク質・核酸（a層），（2）細胞内組織（b層），（3）細胞（a層），（4）器官（b層），（5）個体（a層），（6）生態系（b層）のような階層の積み重ねであるとみるこ

とができる．人工物の設計論の立場で考えると，従来の工学の重点は圧倒的にｂ層的な発想に依存しており，ａ層的な構成というものはこれまでほとんどかえりみられることがなかったといえる．しかし，人工物で実現することのできない多くの機能を生物がもっているという事実から，ｂ層的な構成にそれらの機能を実現するための鍵があるのではないかと考えることは自然である．

実際に有用なシステムを組み立てるためには，ａ層とｂ層の両方からなるより大きな階層構造を考えなければならないのはもちろんであるが，本論文のユニット機械システムの研究では，あえてｂ層だけを考え，均質な部品だけでどのようなことが可能であるかを考えることにする．それは，ｂ層における設計論を確立してはじめて，次の段階のａｂ両層を含む有機的・包括的な設計論に進むことができるのではないかと考えるからである．

３．２　従来の研究

これまでも様々な立場から均質ユニットに近い構造の機械システムが設計されてきている．実際に機械ユニットの製作まで行った例は，ほとんどが一次元状の機械システムに限られている．たとえば，駆動ソフトウェアは均質ではないが，梅谷・広瀬は相互の結合角を変えられる同種の関節ユニットを直列に連結して，蛇状の移動ロボットを作った[43]．また，市川は磁石によって結合する機械ユニットを用いて「一次元自己増殖機械」を作っている[44]．この機械では，動きは非常に単純化されているが，均質なソフトウェアを実現している．二次元では，小鍛治の「フラクタルマシン」がある[45]．これは伸縮可能なリンク機構をユニットとして組み合わせたもので，いくらでも大きな規模のシステムを構築でき，かつ，同一のソフトウェアで全体の並進運動などが実現できる．ただし，全体の構造は可変ではない．

ユニットの均質性の制約を弛めて，いくつかの異なる種類のユニットをもちいるという立場もある．この立場では，任意の故障に対応することはできなくなるが，ユニットの種類が限定されるため，同じ部品が複数あれば故障部品の置き換えも可能である．このような考えに基づいて，福田らはセルラー・ロボティクスの概念を提案している[46-51]．ここでは，システムの冗長性を保ちつつ，かなり現実的な機能を持ったシステムが構築可能である．セルラー・ロボティクスでは，CEBOTと呼ばれるロボッ

トのシリーズが提案されており，バージョンごとに特色ある機能が実現されている．特に，2次元的な自走式セルが機械的に結合するバージョンが精力的に研究され，伸縮セル，関節セル，ハンドセルなどからなるマニピュレータなどが製作されている[47][48]．また，関節ごとに着脱可能なセルによって3次元的なマニピュレータを構成するバージョンでは，組み立てたマニピュレータが，自分と同じマニピュレータを組み立てることを目標に研究が進められている[49][50]．

均質型のシステムの例としては，やはり福田らのCEBOT のMark IIIというバージョンがあげられる[51]．このユニットは6角形を基本とし，その各面が凹または凸の結合機構を有していて，ユニット同士の結合ができるようになっている．また，ユニットは自走する機能も備えており，さまざまなシステムの構成ができ，また，その構成を自律的に変更することが可能である．

Chirikjianらの提案した，6角形のリンク機構で構成されるユニットが多数集まって任意の構造物を構成するシステムもある[52][53]．このシステムは，これから述べるわれわれのユニットと機能的にはほぼ等価なものであり，システムの形態を自動的に構成することを目的とするなど，本研究に非常に近い研究であるといえる．このユニットのハードウェア・ソフトウェアについては本論文のなかでも何度か触れることになる．

機械システムから少し離れるが，ソフトウェアの世界にもいくつか例がある．第1章で紹介したように，この分野での研究は古く，1950年代には，von Neumann がセル・オートマトンによる自己増殖パターンを研究している[6]. Langtonの自己増殖パターン[54] は Neumann のそれをコンパクトに表現することに成功したもので，近年のいわゆる「人工生命」の研究の嚆矢となった．ただ，先に述べたように，セル・オートマトンは論理空間上の構築物であって，そのままでは実際の機械ユニットとして実現することはできない．

近年，移動ロボット群が協調して作業を行うためのソフトウェアが盛んに研究されている[55][56]．これらも，移動ロボットの一つをユニットと見れば，均質ユニットからなるシステムとみなすことができる．ただし，この場合,ユニットの間の幾何学的・力学的な相互の拘束は比較的弱いものである．ここでは，蟻の社会における作業分担（スウォーム・インテリジェンス）のようなものが目指されている[57]．

以上見てきたように，具体的な機械システムでハードウェアとソフトウェアがとも

に均質であり，さらに構造が可変であるようなシステムを実現した例はこれまで殆ど皆無といってよい．

3．3　機械ユニットの設計条件

第1章で紹介したように，Penrose は同一の部品だけで自己組織的に機械を構成する試みを行なっている[10]．そのモデルは機械と呼ぶには程遠いものであったが，部品形状に課せられる幾何学的制約条件を明らかにしたという意味で重要である．それは，部品形状の自己相補性である．つまり，ジグソーパズルのピースのように，部品の凸部は，同じ部品の凹部にぴったりとはまる必要があるということである．この条件は機械ユニットの設計を著しく制約する．さらに，我々はユニットに必要な最小限度の「機能」として次の2つを考える．

・結合機能：各ユニットは，他の複数（3個以上）のユニットと結合することができる．ユニットが他の2個としか結合できないと，ひも状の一次元機械しか作れない．
・結合の変更機能：各ユニットは，協調してユニット間の結合状態を変更することができる．

第一の機能によって，いろいろな形態の機械システムを構成することができる．各ユニットに3つ以上の結合を許すことで，化学者が分子模型を組み立てるように，任意の三次元構造を作ることができる．第2の機能によって，全体の形状を変えることが可能になる．また，ユニットが単体として移動機能を持たなくても，これら2つの機能をうまく組み合わせて繰り返し適用することによって，ユニットをシステム内で輸送することができる（次節で具体的な例を述べる）．この機能は，例えば故障部分を切り離した後，そこに再生のために必要なユニットを集めるために使うことができる．

これらの機能をもつ機械ユニットを，「機械システムの部分を構成する基本的要素」という意味を込めて，「フラクタム (fractum)」と名付ける．

機械ユニットが自己相補的な形状を持ち，この2つの機能をそなえていれば，それをフラクタムと呼ぶことができるが，実際の設計では，機構の簡潔性，信頼性，交信

機能など，さらに多くの制約条件が課せられる．

3．4　ユニットのハードウェア

この節では，ユニットのハードウェアの具体的な設計例を述べる．

3．4．1　基本ハードウェア

二次元的な機械を構成するための機械要素として，Fig. 3.1 のようなユニットを設計した．このユニットは平面上に置かれ，ボールキャスタによって支持されるが，それ自身に移動する機能はない．全体は円弧を組み合わせた特殊な形状の三層の構造部材からなり，内部に可動部分は全くない．三層とも形状は同じであるが，中間層だけ位相を６０度ずらして固定してある．したがって，このユニットは Penrose の自己相補性を各層ごとに違う位相で満足している．上層と下層には３個づつの永久磁石が，すべて上側が N 極になるように埋め込まれる．中層には，永久磁石のかわりに電磁石が埋め込まれる．各ユニットは独立したCPU を持ち，これが電磁石の極性を制御する．

ユニットの結合原理を Fig.3.2 に示す．電磁石の極性を永久磁石の極性にそろえると，引込力を発生する．これを結合に利用する．この結合を解放するには電磁石を逆向きに励磁すればよい．また，電磁石を引き込んだ状態では，電磁石を回転しても，磁場に何らの変化を生じないから，この状態を軸受けとして使用することができる．

Fig.3.3 はこのユニットが前章の２つの機能を備えていることを示している．まず，このユニットは最大６個までの他のユニットと結合することができ，その結合を変更することができる(a)．さらに，古い結合を解除し(b)，新しく別のユニットと結合する手順を繰り返して，ユニットを輸送することができる(c)．

この設計例では，結合のための力として，電磁力を利用したが，これを静電気力に置き換えてもよく，マイクロ・スケールのユニットも実現可能であろう．

さて，先に述べたように，我々のユニットとよく似た機械ユニットをChirikjianらが提案している[52][53]．そのユニットは６本のリンクからなる輪状可変リンク機構で，リンク間の角度を３個のサーボモータで制御することにより，ユニットを自由に変形させることができるものである．このユニットは自分のリンクと隣りのユニットのリンクを機械的なフックで接合する機構を備えており，リンクの変形と隣接ユニットとの

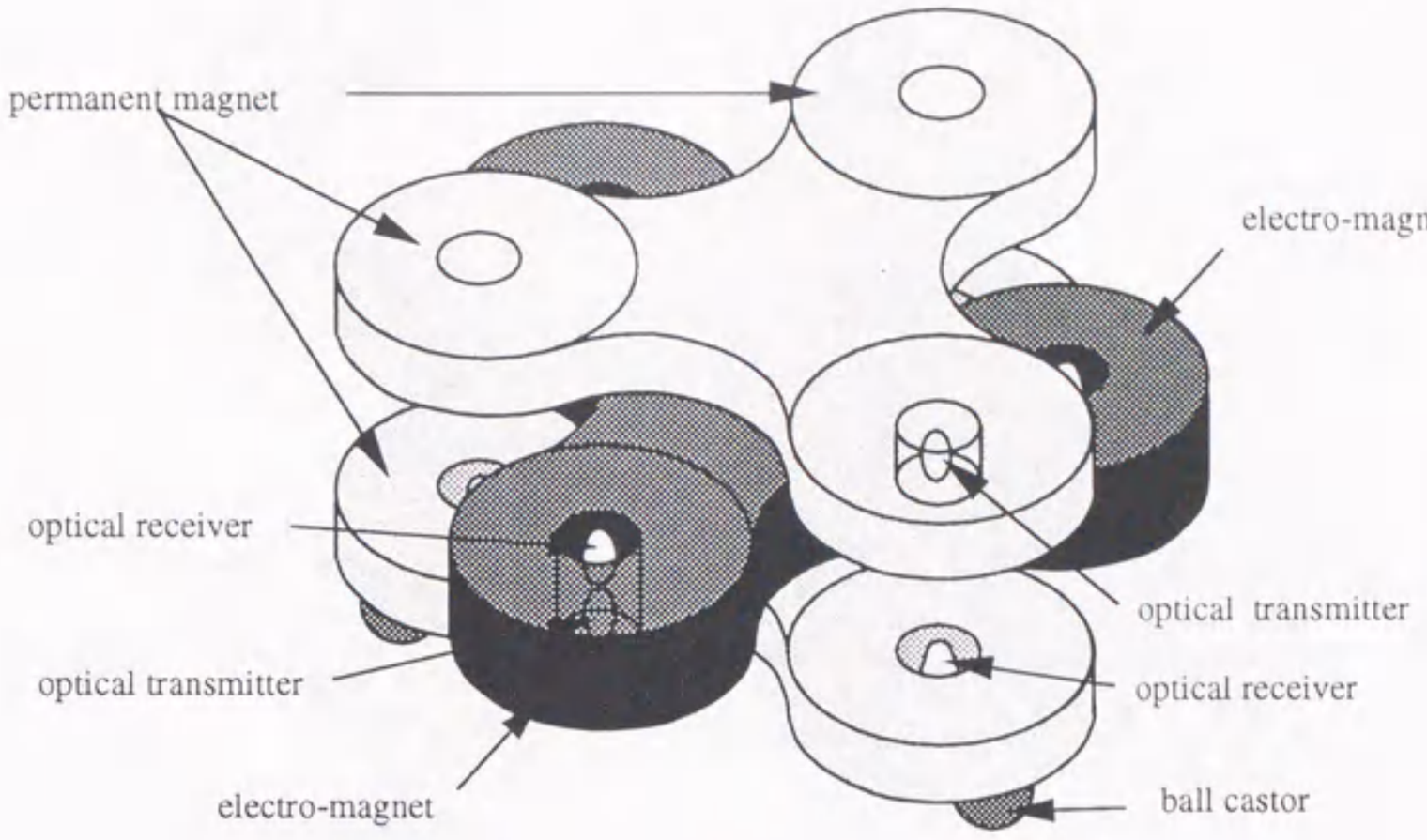
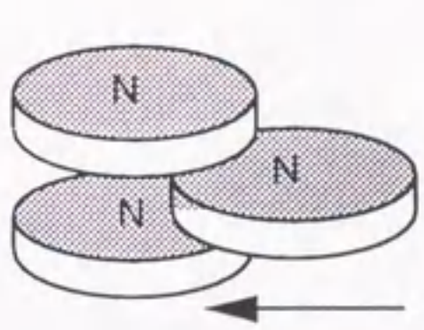


Fig. 3.1 Two-dimensional fractum



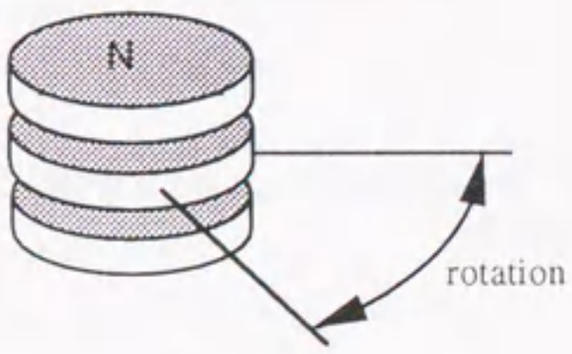
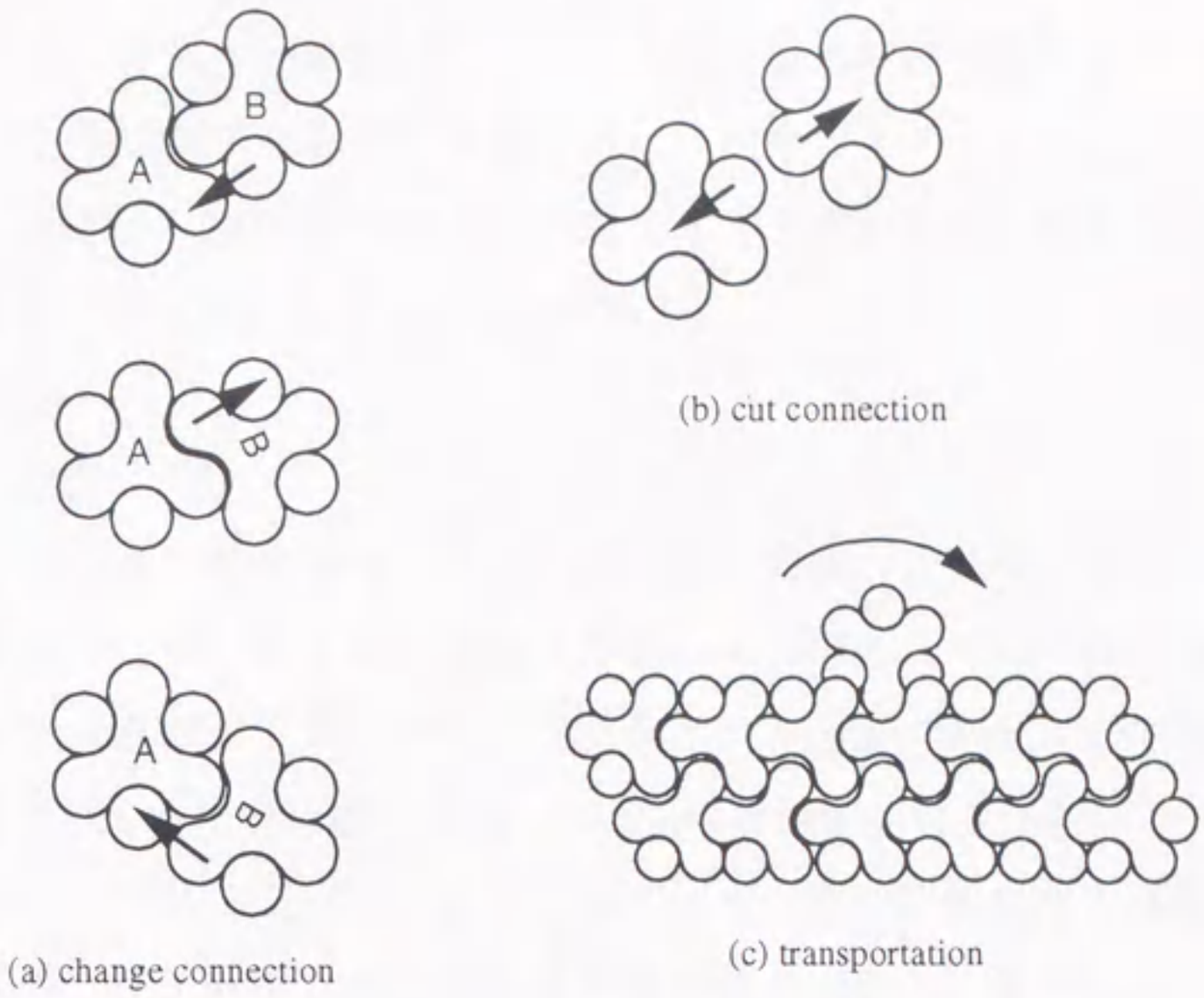


Fig. 3.2 Principle of action



(a) change connection

(b) cut connection

(c) transportation

Fig. 3.3 Basic functions of fracta

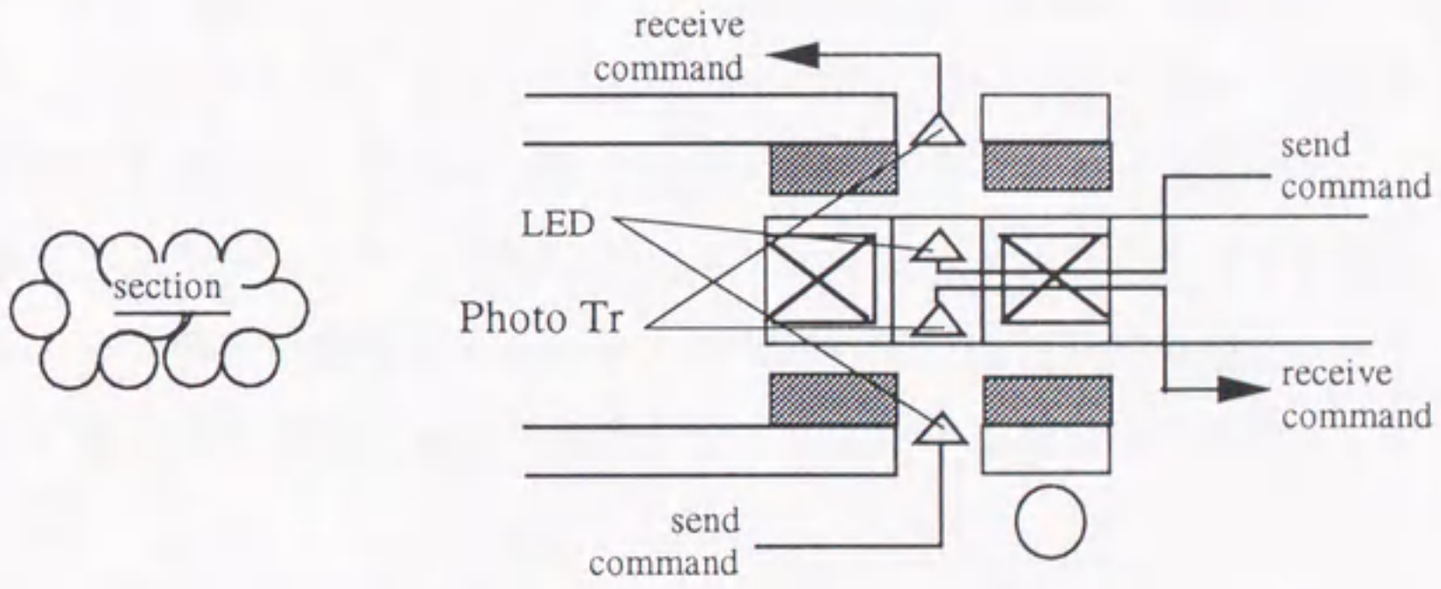


Fig. 3.4 Device for communication

結合離反をうまく組み合わせれば，ユニット間の位置関係を変更できる．ユニット内にサーボモータを搭載するなど，単体の構成は我々のユニットと比べた場合，相当複雑になるが，基本的には同様な動作が可能である．剛性の高い構造体を構成できることや，大きな力を発生できるなどの利点もある．

3．4．2　通信

ユニットの数が多い場合を考えると，放送型の通信（一対多の通信）は通信量が爆発するので適当でない．そこで，隣接するユニット間のみで局所的に通信を行う．この通信は，赤外線通信素子を用いた非接触シリアル通信である．送信素子と受信素子を磁石と電磁石の中心の穴に埋め込む． Fig.3.4 は結合中の２つのユニットの断面を示したものである．これによって，結合中のユニット間では双方向の通信が可能となる．すなわち，右のユニットが上層の送信素子を発光させると，左のユニットは中層の受信素子でこれを受信する．逆に，左のユニットは下層を使って送信し，右のユニットは中層でこれを受ける．

3．5　ハードウェアの基本機能実験

Fig.3.5　に示すように３つのユニットを製作した．このユニットのサイズは，直径12.5cm，高さ16cmで，重さは1.1Kgである．電磁石のコイルの消費電力は１個当りおよそ6Wである．CPUとしては，クロック10MHzのZ80（ROM，RAMともに32Kbyte）を採用した．プログラムはＣ言語で記述した．また，CPU　間の通信プロトコルとしては，調歩同期式を用いた．電力は 12V の直流をケーブルによって直接供給した．

簡単なマスター・スレーブ形式のシーケンシャル・プログラムを作成し，CPU　によるコイルの制御と，隣接するユニットの CPU 間の通信機能を確認し，さらに，結合機能および結合変更機能によって全体の形状を変更する実験を行った．

3．6　３次元ユニットの構想

この章の最後に，ユニットの３次元化について，若干触れておきたい．これまで詳しく述べてきたユニットは２次元であるが，本格的な機械システムを構成するためには，どうしても３次元ユニットが必要である．最近，その実現方法についてひとつの

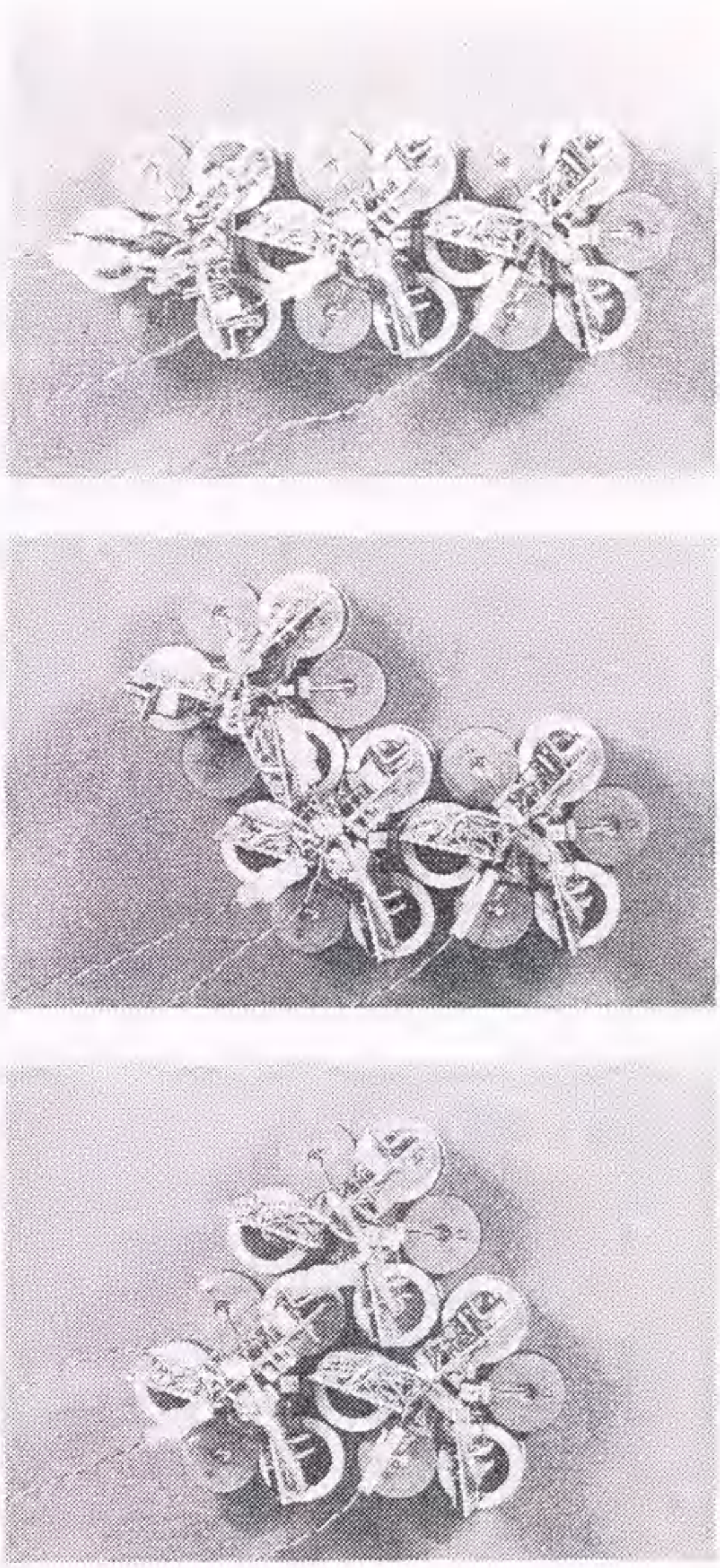


Fig. 3.5 Basic movement of prototype unit

アイデアを得たので，以下でそれを簡単に説明する[58].

均質ユニット機械システムにおいては，各ユニットがまったく相同でなければならない．したがって，設計においては，いわゆる形状の相補性が重要な制約条件となる．ユニットの形状における凸部は同じ形状の凹部に対応していなければならない．これは幾何学的な設計条件としては，大変厳しいものであるが，2次元ユニットの設計では，作動平面上にいくつかの階層をもうけることでこの問題を事実上回避することができた．しかし，3次元ユニットにおいてはこのような「あまり」の次元はもはや存在しないため，設計がより難しくなる．また，これとは別にユニットの駆動力の問題もある．2次元ユニットにおいては，床面との摩擦力に打ち勝つだけの力を発生すればユニットを駆動することができた．しかし，3次元の場合は自重を持ち上げるだけの力が必要であるので，出力重量比の高い機構を設計せねばならない．

これらの条件を念頭に，Fig.3.6　に示すような3次元ユニットを考案した．このユニットは直交する3軸方向に6本の結合腕をもち，かつそれぞれの腕の先端には他のユニットとの結合機構が備えられている．また，各結合腕はそれぞれ独立に回転の自由度をもつ．このようなユニットを多数立方格子状に組み合わせることによって任意の構造物を構成することができる．

ユニットの移動は2個のユニットの組を基本単位として行う．ユニットで敷き詰められた平面（A）を考え，そのうえに2個のユニットの組が載っているとする（Fig.3.7）．このとき，一方のユニットの底面の結合を離し，他方のユニットの底面の結合を軸足として，その軸まわりに回転させることによって，浮いた方の足の結合点を変えることができる．この動作を軸足を適当にスイッチしながら繰り返せば，この2個組は敷き詰められた平面上の任意の位置に移動することができる．また，ある平面（A）上に存在する2個組が，他の平面（B）上に移動するためには，Fig.3.8　に示したような方法を用いて1個ずつ別の平面へ乗り移らせればよい．これらの基本動作を組み合わせて繰り返し実行することによって，多様な構造がユニットの組み替えだけによって実現可能である．

以上のようなアイデアに基づいて，現在ユニットを試作中である．設計したユニットは差し渡しの大きさ約２６ｃｍ，重さ６Ｋｇで，試作個数は１個である．このユニットは中央に一辺１０ｃｍの立方体をもち，この中に出力７ＷのＤＣモーター１個お

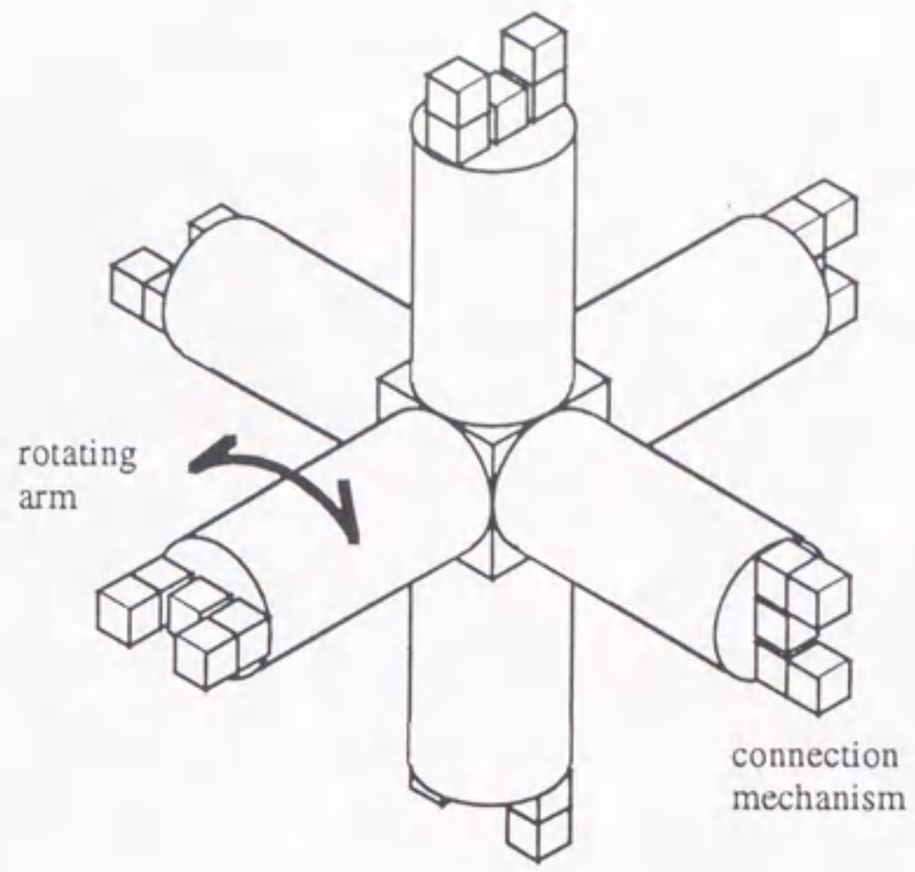


Fig. 3.6 Three-dimensional fractum

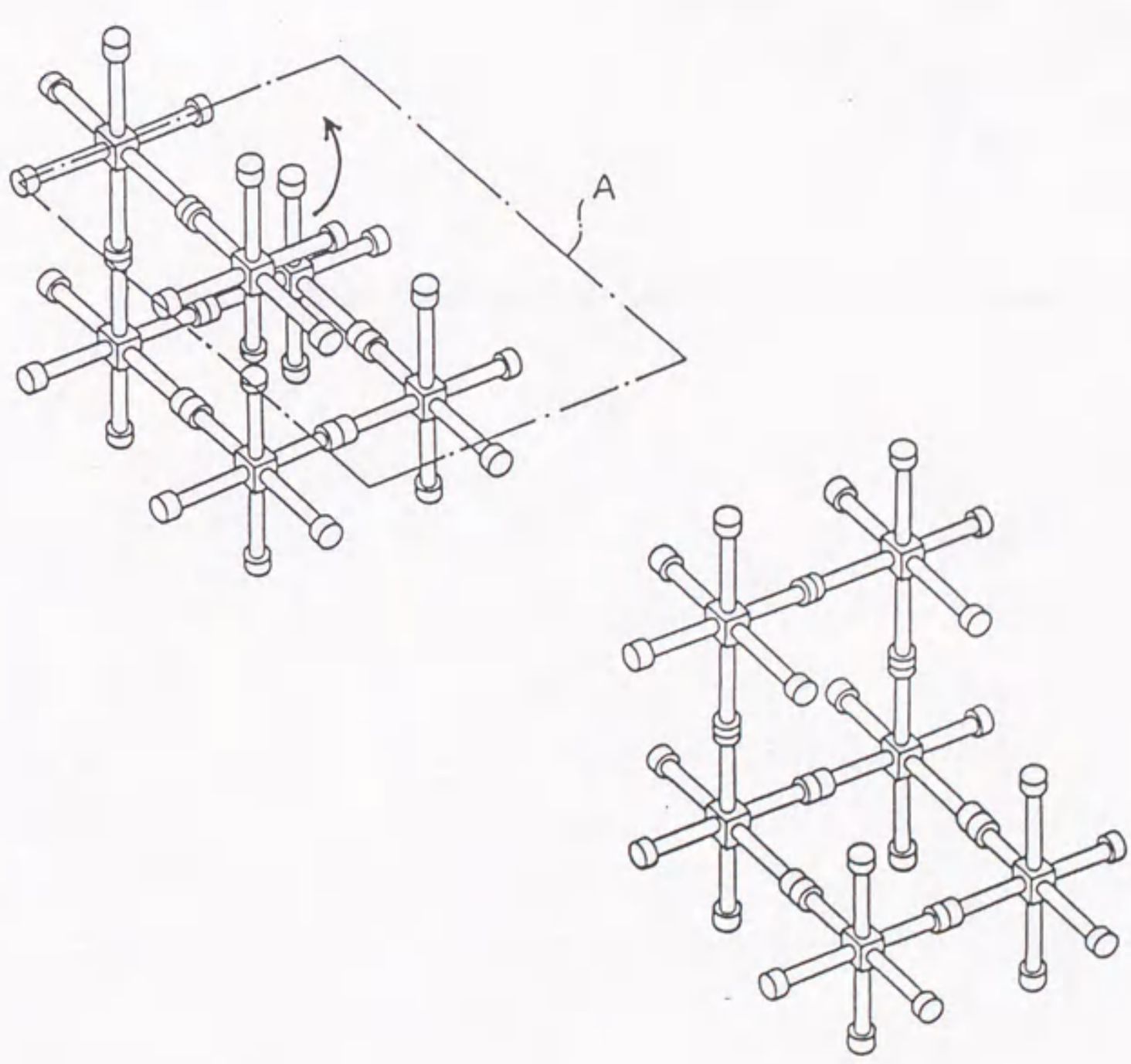


Fig. 3.7 Basic function of 3D fracta: move on a plane

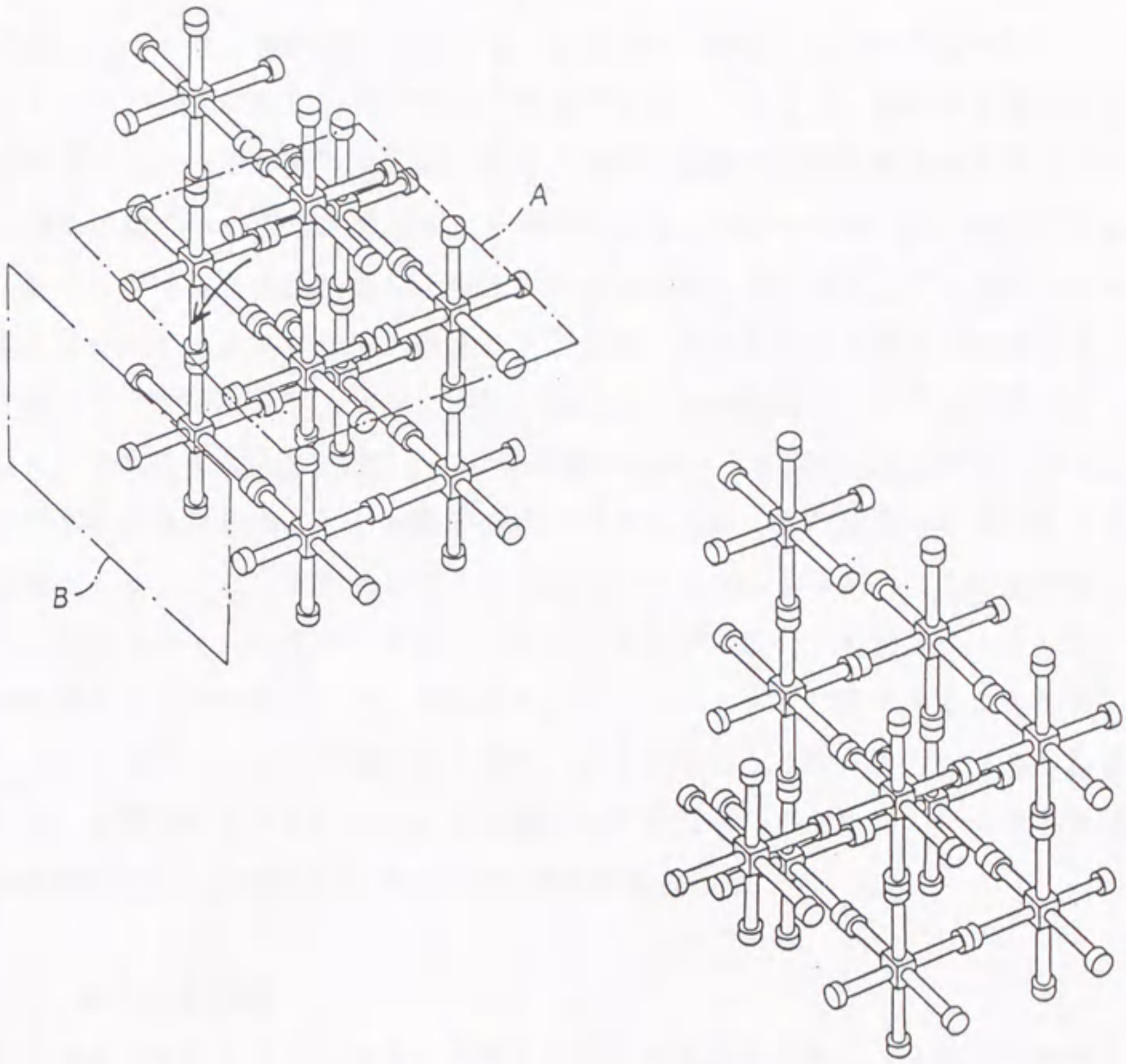


Fig. 3.8 Basic function of 3D fracta: transfer to another plane

よび減速比１００の減速機を内蔵する．立方体の各面に６本の結合腕を取り付け，腕のそれぞれが独立に回転および結合・解放できるようにする．腕の回転機構および結合機構はモーターで駆動されるが，モーターからの動力伝達経路の切り替えによってどの腕を駆動するかの選択を行う．そのため電磁クラッチ６個（腕の結合・解放用）とソレノイド６個（腕回転用）を備えている．予備的な試験として，試作ユニットに重量６Ｋｇのダミーユニットを結合したところ，これを持ち上げる十分なトルクを発生できることが確認されている．今後，腕同士の結合機構についても試験をすすめ，ユニット全体の基本動作を確認してから複数のユニットの製作に進む予定である．

さて，この３次元ユニットの動き方は２次元のユニットの動き方とは本質的に異なる側面をもっている．２次元ユニットではひとつのユニットが，外力の助けなしに，もう一つのユニットのまわりを回ることができたが，この３次元ユニットでは，移動の最小単位はひとつのユニットではなく，２つのユニットの組であることである．もちろん，３次元ユニットを設計する場合，必ずこのような動き方をさせなければならないという訳ではないが，このような動かし方を採用したために，ユニット単体の構成を単純化することができたのではないかと考えられる．

３．７　まとめと考察

ここでは，均質ユニットによる機械システムの構成を提案し，２次元の機械システムを構成するためのユニットのハードウェアを具体的に示し，それを実際に設計・製作して，基本動作を確認した．また，３次元のユニットについてもそのアイデアを示した．

これまでに提案されてきた分散型の機械システムのほとんどは，一定の構成を変えることなく使用するもので，１次元的なひも状のロボット（蛇ロボット，象の鼻マニピュレータ等）がその主流であった．これに対し，機械システム全体の構成そのものが可変であるようなユニット型のシステムは，これまでほとんど提案されておらず，最近になって，われわれのユニットを含めやっと数例を数えるにとどまっている．しかし，全体の構成を変えられることのできるユニットでシステムを構成できれば，多様な要求や環境の変化にあわせて柔軟に構造・機能を変えられる機械システムが実現できることになるのであるから，たとえ実用化の道は遠くとも，この種の研究の重要

性は高いと考えている[59]-[62].

また，機械工学全体の趨勢として，あらゆる機械要素，機械部品のマイクロ化が進んでいることも，均質機械ユニットシステムの実用化に向けて大きな意味を持っている[63][64]．本章で提案した機械ユニットは，内部に可動部分がなく，マイクロ化に適した構造をもっており，十分ユニットを小さくできれば，電磁石のかわりに金属板に帯電させ静電気力によって駆動することができるかもしれない．そうなるとさらに構造は簡単になり，シリコンプロセスだけによってユニットを製造する可能性がでてくる．

一方，ユニットにどうやって電力を供給するか，ユニット間の結合の剛性をもっと高くする，あるいは，剛性を可変にするにはどうすればよいかなど，いくつかの問題が将来に残されている．

第4章　自己組み立てのためのソフトウェア

4．1　はじめに

均質なユニットで構成される機械システムの特徴は，どのユニットがどこに使われてもよいということである．言い換えると，各ユニットには，システムのどの部分にもなれる潜在的な能力が要求される．したがって，各ユニットは同じソフトウェアで駆動され，そのソフトウェアはシステム全体に関する情報を持っていなければならない．これは丁度，我々のからだを構成するすべての細胞が同一の遺伝子を持っているのと相似な構造である．システムに対してこのような要求をすることは，システムのソフトウェアにも厳密な均質性を要求することに他ならない．

もう一つの制約条件として，ユニット間通信に関する制約がある．これは，ユニットのハードウェアの構成上，隣接ユニット間でしか情報交換ができないという制約である．これを「近傍通信の制約」と呼ぶ．ただし，隣接しないユニット間では情報交換がまったくできないわけではなく，その場合は，通信しようとしている2つのユニットの中間に存在するユニットが情報伝達の仲立ち（バケツリレー）をすればよい．こうすれば，ユニットは連結状態にあるシステム内のどのユニットとも原理的には情報交換が可能である．しかし，システムに含まれるユニットの個数が多くなると，システム内の任意のユニット間の通信を前提とするようなソフトウェアでは，通信量が爆発的に増大し，ソフトウェア自体も複雑化してしまう．したがって，近傍通信の制約は，ハードウェア的な制約という意味だけでなく，ソフトウェアの構成上の制約という意味をも持っているのである．

本章では，ソフトウェア設計の具体的な問題として，いわゆる自己組み立て問題を考える．まず，問題を定式化しよう．

［問題］任意の初期配置に置かれたユニット群を考える．ただし，Fig. 4.1 に示したように，各ユニットは少なくともひとつのユニットと結合しているものとする．各ユニットには，あらかじめ全体の目標の形状に関する情報を与えるが，現在の全体の配置に関する情報は与えない．各ユニットは，結合中の隣接ユニットと通信することによ

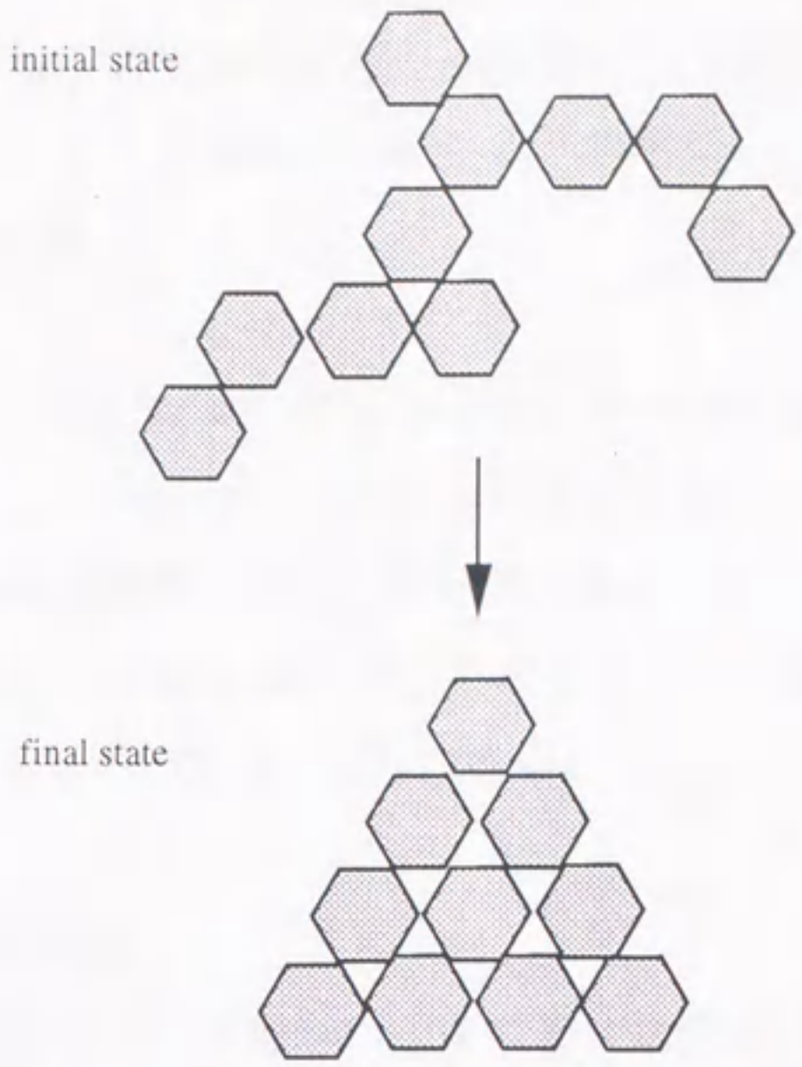


Fig. 4.1 Self-assembly

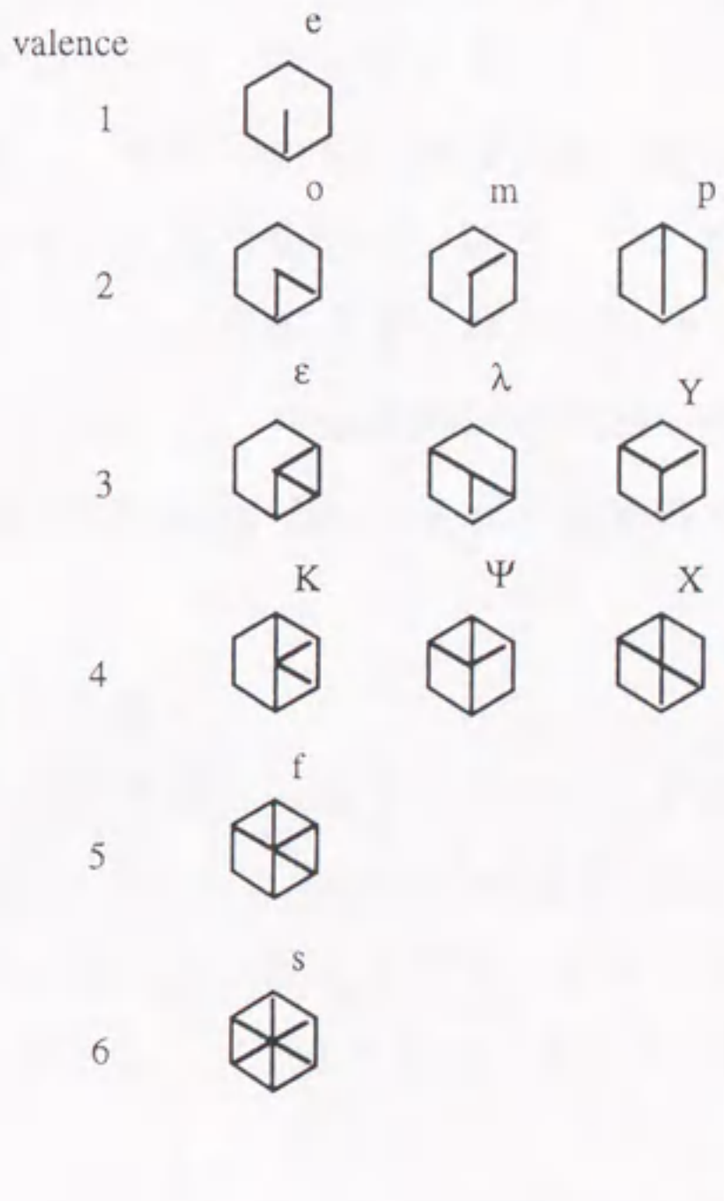


Fig. 4.2 Type of connection

り，自分の周りの局所的な結合状態のみを知ることができるとする．このような制約条件の下で，各ユニットの電磁石をどのように駆動したら，目標の全体形状を形成することができるか？■

初期状態においては，将来，自分が全体の中のどの部分を担当することになるかという「運命」は決まっていない．形態を形成する過程でそれがしだいに決まってゆくという意味で，この問題は「分化」の問題であると言うことができる．また，任意の状態から，全体の形状を形成するわけであるから，これを形状の自己組織（あるいは自己修復）の問題として捕えることもできる．

4．2　全体形状の記述

各ユニットは通信によって自分の近傍の結合状態だけを知ることができる．そこで，全体の形状を局所的な結合関係だけで表現する方法が必要となる．

4．2．1　結合の「型」

一個のユニットが取りうる結合の形式は，回転や鏡像を無視すれば，Fig. 4.2 に示したように全部で１２通りあり，これをそのユニットの「型」と呼ぶ．図中の六角形が一つのユニットを表し，その中の短い棒は，その先に別のユニットが結合していることを示している．結合の本数を結合価と呼ぶ．したがって，各ユニットは１から６の結合価をとる．ただし，この分類では，結合はすべて単結合（一つの電磁石だけで結合している状態）であるとし，二重結合は過渡的にのみ許されるものとして考慮しない．また，λ型は反転すると別の型になるが，簡単のため区別しないこととした．

4．2．2　型の間の距離

異なる二つの型の間に距離を定義するために，次の型遷移ダイアグラムを導入する（ Fig.4.3 ）．一般に，ユニットが結合を変えて移動すると，そのユニットの型が変わるだけでなく，そのユニットに結合していた相手のユニットの型も変化する．このダイアグラムのノードはユニットの型を表し，ユニットの１ステップの移動で，ある型

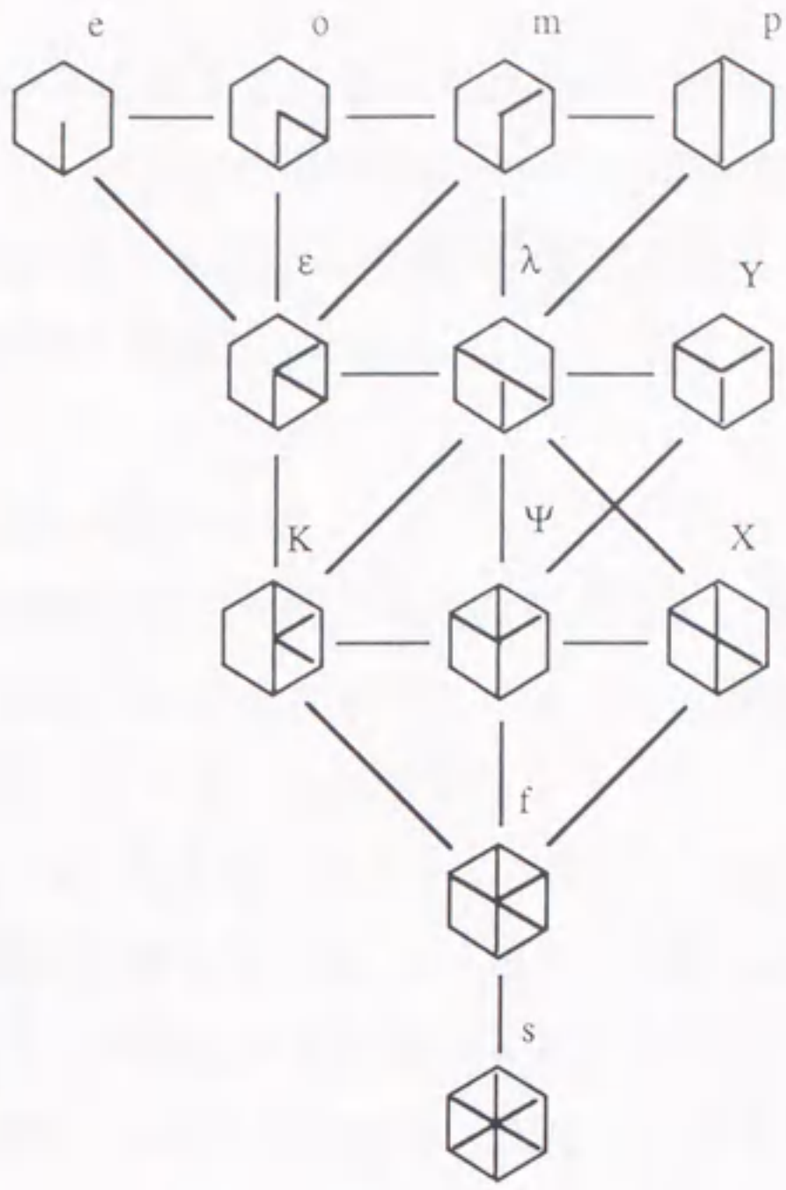
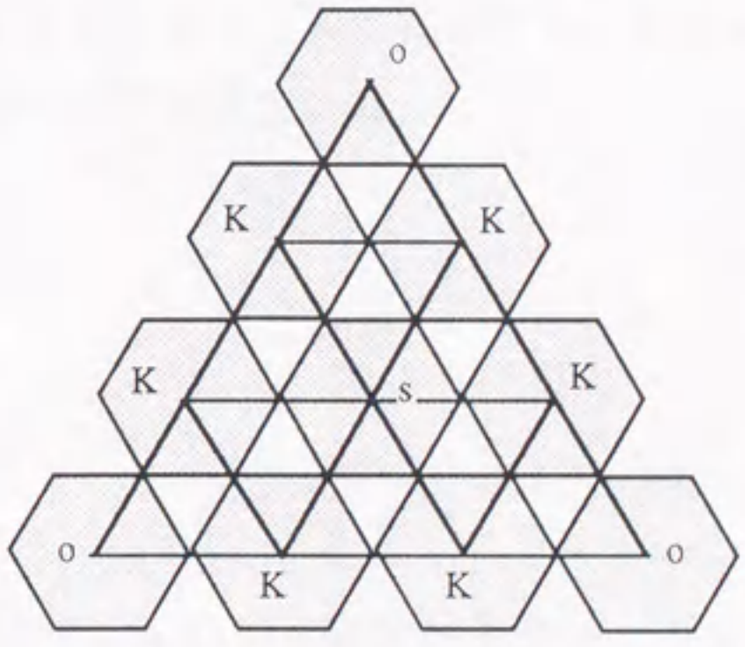


Fig. 4.3 Type transition diagram



o(K,K)

K(o,K,K,s)

s(K,K,K,K,K,K)

Fig. 4.4 An example of target shape and its description

が別の型に変わることがある場合に，それらのノード間にリンクを張ってある．二つの型の間の距離は，ダイアグラムの上の最短パスのリンク数で定義される．例えば，e型と X 型の間の距離は3である．この距離の定義は大雑把ではあるが，ある型が別の型にかわるのに要する手間を反映している．

4．2．3　型による全体形状の記述

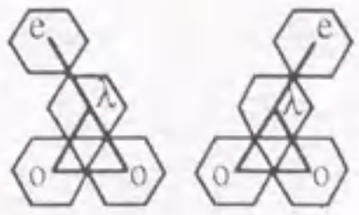
各ユニットは，自分の結合状態を調べることによって，自分の型を知ることができる．また，隣接するユニットと交信することによって，周囲のユニットの型を知ることができる．自分に結合しているユニットの型を並べたリストを「隣接型リスト」と呼ぶ．例題として，Fig.4.4　に示したような１０個のユニットで構成される正三角形の配置を考えよう．この形状には3つの型，o（頂点），K（辺），s（中心）が含まれる．このとき，o型のユニットの隣接型リストは　{K,K}　であり，K型は　{o,K,K,s}，s型は　{K,K,K,K,K,K}　を隣接型リストとして持つ．ただし，リストの中の順序は結合価の昇順で，同じ価数の型が続く場合はFig. 4.2 の順（e,o,m,p,ε,λ,Y,...）とする．

これで，全体の形状を局所的な結合関係だけで記述する準備ができた．目標とする全体形状のなかに含まれる型を「目標型」と呼び，目標型とその隣接型リストの組によって全体形状を記述する．この目標型と隣接型リストの組を「文（statement）」と呼ぶ．同一の文が複数あるときはひとつの文で代表する．このようにすると，Fig. 4.4 の例題の形状は，つぎの3つの文で記述される．

o{K,K}
K{o,K,K,s}
s{K,K,K,K,K,K}

全体の形状がもっと複雑であったり，個数が多かったりすると，この記述は長くなる．また，形状から記述は一意に決まるが，記述から形状は一意に定まらないことがある．例えば，λ型の反転を区別しなかったことから，化学における異性体に似た区別されないいくつかの構造を生じる．例えば，記述：e{λ}, λ{e,o,o}, o{o,λ}からは，下の

ような2種の異性体を生ずる．また，5個以上のユニットが一列に並んだ状態の記述はつねに，e{p}, p{e,p}, p{p,p}であり，この記述では列の長さが不定となる．



しかし，Fig. 4.4 の例題のように，形が単純で対称性の高い場合には，このような不定性はほとんど問題にならない．

4．3　不適合度

形態形成を分散的に実行するためには，各ユニットが目標形状のどの部分に現在の自分の結合状態が最も近いかを知ることが必要である．そこで，現在の結合状態（現在の自分の型と隣接型リスト）と，目標形状を記述した文（複数個ある）の間の近さを表すために，次の不適合度 (difference) を導入する．不適合度は次式で定義される．

$$difference(i) = \min_{j=1}^{M} [\, d(type^d(j), type(i)) + \sum_{k=1}^{6} d(ntype^d(j,k), ntype(i,k)) \,] \qquad (1)$$

ただし，

M　　：目標形状を表す文の数

d(a, b)　：型a と型bの間の距離

type(i)　：i番目のユニットの現在の型

ntype(i,k)：i番目のユニットの現在の隣接型リストのk番目の型

$type^d(j)$　：j番目の目標型

$ntype^d(j,k)$：j番目の目標型の隣接型リストのk番目の型

である．各ユニットは独立してこの式を評価することができる．各ユニットは，それぞれの文について，自分の現在の型とその文の目標型の間の距離を計算する（第1項）．さらに，現在の自分の隣接型リストと各文の隣接型のリストを比べて，リストの要素についてもひとつづつ距離を計算する（第2項）．一般に現在の型の価数と目標型の価数は異なるが，その場合，比較するのに足りない項は便宜的にe型とする．例えば，隣接型リスト{K,K}と{K,Y,s}の違いを評価するときは，{-,K,K}と{K,Y,s}を比較する

（記号 "-"を短い方のリストの先頭に挿入し，これを e 型とみなす）．これらの距離の総和をそれぞれの文について計算し，そのうちの最小の値を，そのユニットの不適合度とする．もし，あるユニットが目標の全体形状の一部になったとすると，そのユニットでは不適合度はゼロになる．当然，全体の形態形成が完了すれば，すべてのユニットで不適合度はゼロになる．不適合度は各ユニットがどの程度目標の形状から離れているかという度合を表す．

4．4　可動性

多数のユニットが結合した状態において，すべてのユニットが移動できるわけではない．たとえば，6 個のユニットはすべてのスロットがすでに埋められているので，動きようがない．ここでは，各ユニットはたかだか自分 1 個を動かすトルクを発生することができるとして，動くことのできる型を限定し，それを「可動型」と呼ぶ．e，o，ε の三つが可動型である．

4．5　移動の戦略

不適合度の大きいユニットを優先的に動かしていれば，いずれは目標形状に到達すると考えられる．しかし，ユニットは局所的な情報交換しかできないので，全体を見渡して不適合度が最悪のものを選ぶのは困難である．そこで，次のような戦略を考える．

［戦略］可動型のユニットで，不適合度がそのユニットの周辺のローカルな平均値より相対的に大きいものを優先的に移動させる．移動の方向は特に決めず，ランダムとする．■

移動方向をランダムとしたのは，どちらの方向に移動したら形態形成が進むかと言うことを局所的に判断することが難しいからである．しかし，移動した結果，もし形態形成が進めば（不適合度が減少すれば），そのユニットの優先度は低下して次には動きにくくなるから，全体としての形態形成は後戻りしにくくなっている．この戦略

では，動くユニットがローカルに選択されるため，大規模なシステムでは複数のユニットが同時に動くことが許されるという利点もある．

この戦略を実行するためには，不適合度のローカルな平均値を各ユニットごとに推定する必要がある．そこで，次の拡散場を導入する．

4．6　拡散場

多数のユニットが隣接するユニットとしか通信できないときに，それぞれのユニットが持つある連続量のパラメータの平均値を推定する問題は，簡単なモデルを用いて解くことができる．すなわち，各ユニットに同じ大きさのタンクを用意して，パラメータの大きさに応じて水を入れ，隣接するユニットのタンクの間を同じパイプで結ぶ．すると，全体の水量は保存され，一定時間の後にはすべてのタンクの水位が同じになるから，それを利用して平均値を知ることができる．この過程を表す方程式は拡散方程式で，

$$\frac{d}{dt}x(i) = K[\sum_{j=1}^{c(i)} x(n(i,j)) - c(i)x(i)] \qquad (2)$$

と書き表せる．ただし，

$x(i)$：i番目のユニットの拡散変数（水位）

$c(i)$：i番目のユニットの価数

$n(i,j)$：i番目のユニットに接続するj番目のフラクタムの番号

K　：拡散係数

である．また，マイナスの水位にならないように，各拡散変数 $x(i)$ はゼロより小さくならないようにしておく．この方程式の変数 $x(i)$ の初期値としてそのユニットの不適合度を与え，適当な時間待つことによって，この変数を不適合度のローカルな平均値として使用することができる．実際には，ユニットの移動によって結合状態が変わるたびに，新しく評価した不適合度を拡散変数に代入しなおすことで，この方程式を連続的に用いる．

しかし，結合状態が変わるたびに，新しい不適合度とそのときの状態量の差が新たに系に注入されることになり，全体の水量が保存されずに増加する傾向を示す．そこ

で，可動型のユニットに限り，一定速度のリーク（漏れ）があるものとする．

$$\frac{d}{dt}x(i) = K[\sum_{j=1}^{c(i)} x(n(i,j)) - c(i)x(i)] - L \qquad (3)$$

（ただし，可動型のユニットにかぎる）

ここに，Lはリーク定数である．こうすることによって状態量の漸増を避けることができ，同時に可動型のユニットがより動き易くなる．

4．7　発火判定

あるユニットの不適合度がローカルな平均より相対的に大きいことを判定するために，次の判定式を用いる．

$$G\,x(i) < \mathrm{difference}(i) \qquad (4)$$

ただし，Gは1より大きい発火定数（比）である．もし，この判定式が満たされ，かつそのユニットが可動型であれば，ただちに1ステップ移動する．（可動型でない場合は何もしない．）このプロセスをユニットの発火と呼ぶ．この判定式では，不適合度がゼロのユニットは決して発火しない．また，形態形成が進むにつれ，不適合度の平均値はゼロに近づくが，判定に比を用いているので，相対的に大きい不適合度をもつユニットはいつかは必ず発火する．したがって，形態形成が完全に完了するまで，どれかのユニットがかならず発火するようになっている．形態形成を完了すると，すべてのユニットの不適合度はゼロとなり，それ以上動かなくなる．

4．8　計算機実験

ソフトウェアの有効性を計算機シミュレーションで確認した．テスト用の目標形状として，Fig. 4.4 のような三角形の配置を用いた．また，初期配置としては，一列にユニットが並んだ配置を用いた．この配置は可動型が両端に一個ずつしかないので，形態形成には不利なものである．シミュレーションに用いたパラメータは次のとおりである．

拡散係数　　K　0.02

発火定数　　G　1.25（e型，o型）20.0（ε型）

リーク定数　L　0.15（e型，o型）0.02（ε型）

拡散係数は式 (2) をそのまま差分方程式と見立てたときの計算きざみである．また，ε 型が他の可動型に比べて発火しにくくしてあるのは，この型が実際によく現れ，この型の発火を簡単に認めると，いつまでも収束しなくなるからである．

１０個のユニットで三角形を形成する実験を1000回行った．ランダムにユニットの移動方向を決めているので，同じ初期配置から始めても，毎回違った経過をたどる．その結果，2000ステップまでに形態形成を完了する確率は 96.8% であった．成功した例を Fig. 4.5 に示す．残りの 3.2% は可動状態のないデッドロック状態である（例えば，６個のユニットがベンゼン分子のようにリング状につながった状態は，可動型を含まないのでデッドロックになる）． Fig. 4.6 は成功した場合の不適合度と拡散変数の総和の時間的な変化を示したものである．形態形成に一応の方向性はあるものの，基本的にはランダムな探索であることが見てとれる．このため，ユニットの個数が増えると，急激に収束性が悪くなる．たとえば，三角形を一列増やして15個のユニットで三角形を形成する実験では，4000ステップまでの成功率でも 73.4% に低下する．この場合残りの 26.6% は収束せずに動き続けていたか，デッドロックに陥っていたか，あるいはその両方であった．

これらのシミュレーションは，すべて同一のパラメータセットのもとで行われている．これらの値は，シミュレーションを繰り返して観察結果をもとに定めたが，どれかのパラメータに鋭く依存してシステムの挙動が変わるようなことはなかった．おおむね，このアルゴリズムはパラメータ依存性が低いといえる．これは，アルゴリズムの根底にある基本ダイナミクスが拡散方程式で，ダイナミクス自体の安定性が極めて高いためであろうと思われる．たとえば，ユニットが移動したあとの新しい不適合度の代入などで，変数 x の値にはかなり頻繁に強制的な跳躍が生じるが，そのような場合にもシステムが数値的に発散するようなことはなかった．ただ，発火定数とリーク定数については，他の可動型に比べてε型の発火率をどの程度落とすかという比率は収

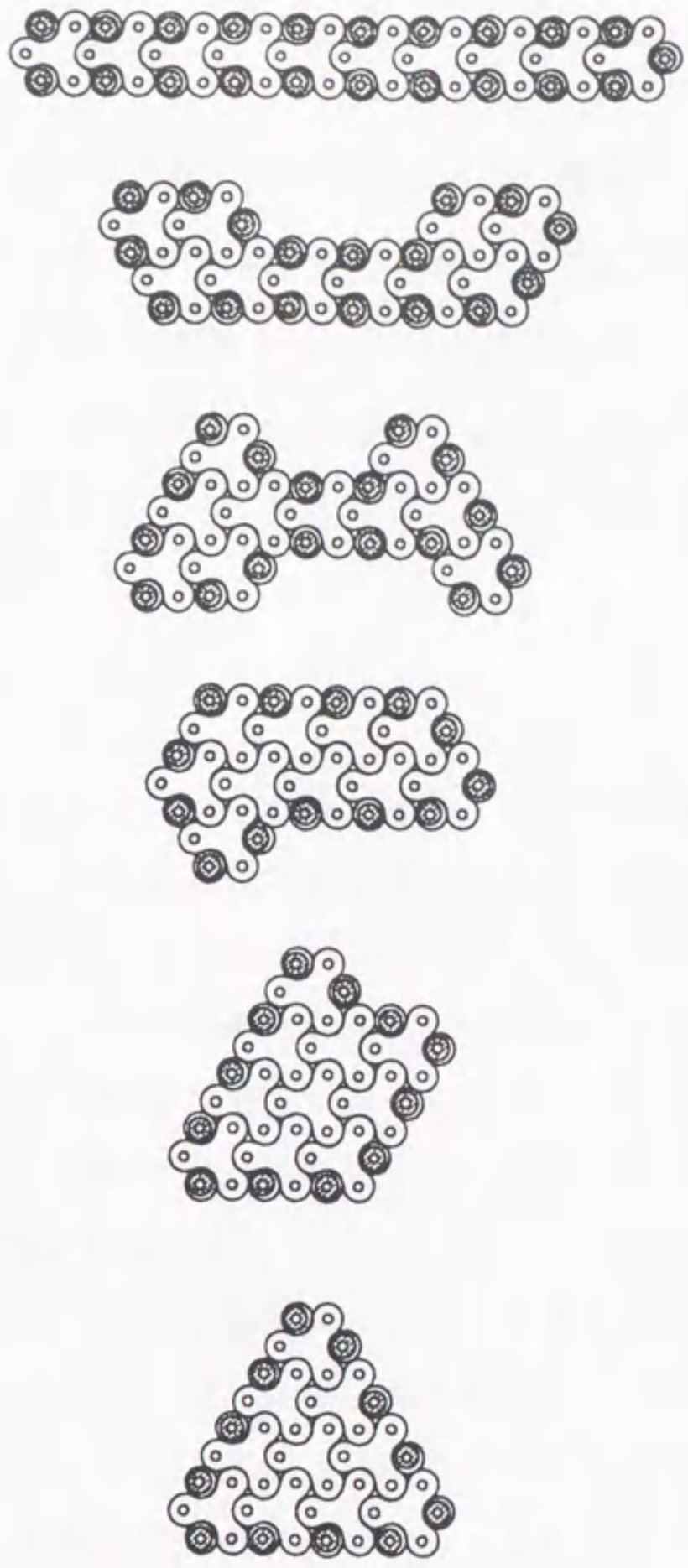


Fig. 4.5 Simulated sequence of self-assembly

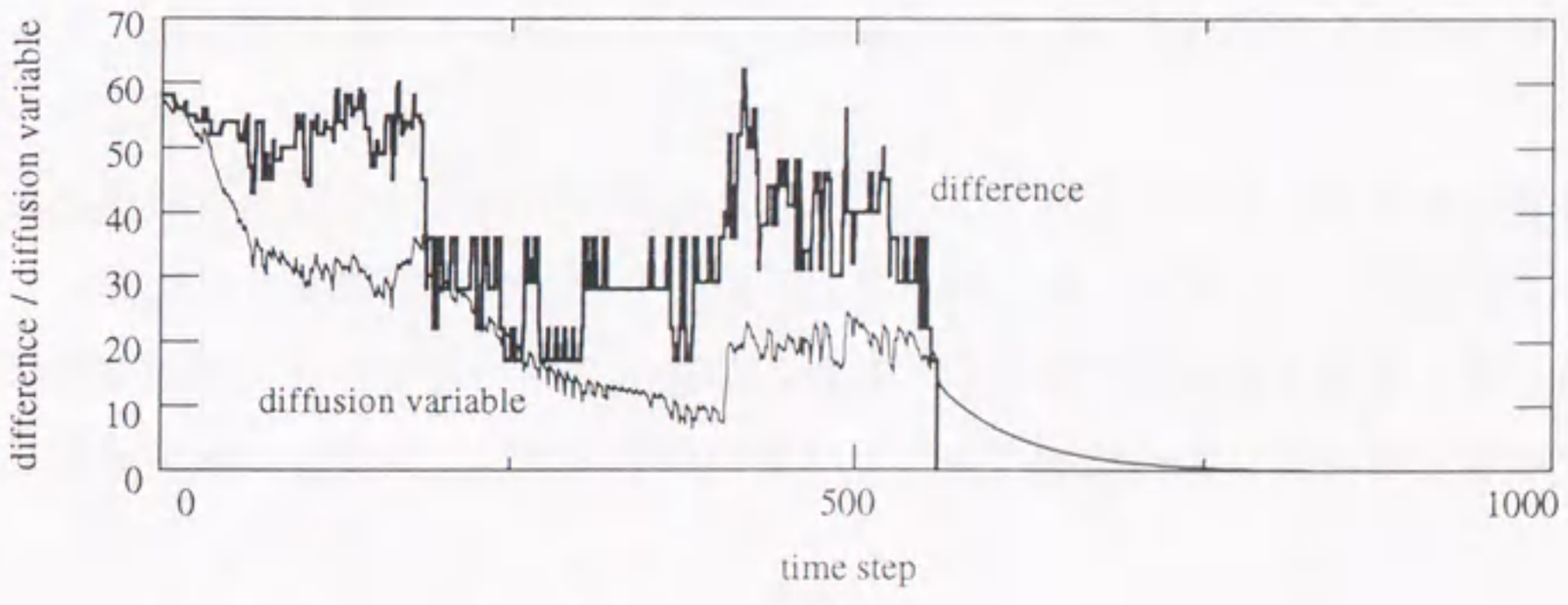


Fig. 4.6 Change of difference and diffusion variable

束の速さに影響する．ε 型の発火率が他の発火率と同程度の大きさのときは，発火するユニット数が増えすぎ，全体がいつまでもうごめいて収束しなくなるが，逆にこれを小さくしすぎるとシステムがε 型で囲まれた6角形状態に落ち込んだときに，ここから抜け出せなくなる．選んだ数値は，シミュレーションの傾向を見ながら決めたものだが，定性的な傾向をみて決めただけであるので，これが最適であると保証することはできない．

4．9　まとめと考察

物理や化学におけるさまざまな自己組織化現象，あるいは生物学における発生および形態形成の問題として，自己組み立ての問題は古くから扱われており，さまざまな仮説が唱えられている[65]-[68]．しかし，それらの多くは，きわめて微細な原子・分子の世界を相手にしていたり，実体を離れてあまりに数学的・抽象的であったり，また，ある生物にだけにあてはまる独特の理論であったりして，ここで対象としている機械ユニットにそれらの仮説・理論を直接適用することは難しかった．したがって，ここではまったく新しい方法論を開発せざるを得なかった．

開発したアルゴリズムは，特殊な機械ユニット「フラクタム」を対象としてはいるが，結合を変えられるユニットシステムの制御の方式として，ある程度の一般性をもっているものである．また，アルゴリズムの細部の手続きがすべて詳細かつ具体的に記述されており，さらに実際のハードウェアユニットをもちいた実験が可能なことが，検証が困難な生物のかたちづくりに関する理論や仮説と大きく異なる点である．もちろん，ここで提案したアルゴリズムによって生物の発生過程が説明できると考えるのはあまりにも早計であろう．ただ，複雑すぎて先のよく見えない生物システムの研究において，全体を見通すある種のアナロジーを与えるという意味での価値はあるのではないかと思っている．

計算機実験の結果からも明らかなように，このアルゴリズムはシステムに含まれるユニット数が１０個程度の小規模な場合はうまく働くが，ユニットの数が増えると，その効率は急激に悪化する．また，形状の複雑さも成功率に大きく影響する．このアルゴリズムを用いた場合，三角形以外の形状を作る実験では１０個に満たない極めて

小規模なものでしか高い成功率は得られなかった．これは，目標形状の対称性が高い場合は，記述はコンパクトにまとめられ，目標状態に達したときのユニットのとりうる状態の数は少ないが，目標形状の対称性が低い場合は，記述に含まれる文の数が増え，すべてのユニットがそのうちのどれかと一致しなければ全体が収束しないためであると考えられる．すなわち，あるユニットが記述を満たせずに動くと，それがすでに記述を満たしている別のユニットの状態をくずしてしまう．全体の形態形成が収束するためにはすべてのユニットが同時に記述を満たす必要があるが，ユニットの数が増えるとそのようなことはきわめて起こりにくくなるのである．つぎの章では，もっとユニット数が多く，目標形状も複雑な場合についても適用が可能な方法を提案する．

さて，この章の最後に，前章で紹介したChirikjianらの研究，とくにそのソフトウェアの部分[69]について，本研究のアプローチとの比較を行いたい．ユニットのハードウェアの違いはすでに述べたとおりであるが，彼らも自己組み立てあるいは自己再編の機能の実現を目指してユニットシステムのためのソフトウェアの研究をすすめている．我々が極端な均質化・分散化の設計を行っているのに対し，彼らは集中化・最適化のアプローチをとっている．すなわち，彼らの研究では，ユニットシステム全体の初期形状と目標とする最終形状が与えられたとき，最小の移動回数でシステムを変形させるにはどうすればよいかということを考えるのである．ここでは，ある特定のユニット（彼らはbase unit と呼んでいる）をコントロールセンターとし，このユニット上にシステムのあらゆる情報を集めることができ，またすべてのユニットの運動はこのユニットが指定できるものと仮定して，アルゴリズムの設計を行っている．彼らの検討によれば，システムがある形状から別の形状になるときのユニットの移動の手順はほとんど無限に考えられ，このなかからひとつの最適な手順を選び出すことは事実上不可能である．そのかわりに，最適手順が存在したとして，そのステップ数の上限及び下限を理論的に評価し，さらにユニット移動のルールにある程度のヒューリスティクスをいれて，準最適な手順を導いている．

以上のようなアプローチは，ここで示した我々のアプローチの対極にあると考えられる．Chirikjianらの集中型のアプローチではシステム内の一点にすべての情報を集めることにより，全体のパフォーマンスの最適化を目指しているわけで，これは，従来の工学の考え方に沿ったものであるといえる．しかし，特定のユニットに情報を集中す

ることは，通信の輻輳，情報処理負荷の集中を意味し，均質型のシステム構成では必ずしも有利な面だけであるとは限らない．全体の情報を集めているにもかかわらず，結局は厳密な最適手順を導くことができないのは，システム内部の自由度が多すぎ，それをそのまま吸い上げて統合してもあまり意味がないことを示していると思われる．もちろん，全体としてみた場合，このような体系的な理論に基づく方法は見通しがよいことは事実で，通信量の爆発をうまく押さえ込むことができれば，非常に多くのユニットから成るシステムの組み立て問題も効率よく解くことができるはずである．

第5章　大規模な自己組み立てソフトウェア

5．1　はじめに

前章で述べたような自己組み立ての難しさは，ユニット群の上に何らかの座標を導入することで軽減できる．ユニット群上に原点を設定し，組み立てをおこなう位置や対称軸の方向をあらかじめ指定することができれば，ユニット群のむだな動き（探索の手間）を大幅に減らすことが可能である[70]．ここでいう座標とは，デカルト座標のように明示的な座標軸をもつものである必要はなく，ユニット間のつながり方をうまく記述できるものであればなんでもよい．

ところで，生物の形態形成，すなわち発生の過程は，単純な形態から複雑な形態へと順を追って変化する段階的な過程である．そこでは受精卵から始まって，自分がどの細胞から分裂したかという情報（系譜）の局所的な受け渡しが，重要な鍵になっているといわれている[71]-[74]．

そこで，これら「組み立ての原点の導入」と，「局所的な情報の受け渡しによる段階的な形態形成」のという2つのアイデアをモティーフとして，大規模で複雑なシステムの組み立てアルゴリズムを構築することを考えた[75]．

アルゴリズムの大筋は以下のようなものである．まず，ユニット群のなかから一個を選んでこれを組み立ての原点とする．このユニットを「核」と呼ぶ．核は生物における受精卵に相当する．次に，核に選ばれたユニットは，ある種の組み立て設計図にしたがって，隣接するユニットの中から適当なものを選んで仮想的な結合のネットワークを形成する．（核の選び方，設計図の書き方については後述．）これを第1段階とする．その次は，第1段階のネットワークに含まれるユニットがそのネットワークの外側のユニットを取り込んで第2段階のネットワークを形成する．この手順を繰り返すことによって次々に新しい段階に移り，全体は次第に大きく複雑に成長してゆく．ネットワークが，一個の核から一枚ずつ層を重ねるように段階的に成長して行くので，このアルゴリズムをタマネギ法（onion　method）と呼ぶ．以降では，順を追ってその詳細を説明する．

5．2　物理型と論理型

われわれは6本の結合腕をもった機械ユニットをモデルとして採用したが，この腕による機械的な結合を物理結合とよぶ．物理結合の様式は，回転や鏡像を無視するとFig.5.1 に示したように１２通りあり，これらを「物理型」と呼ぶ．（4章の小規模な自己組み立てで用いたものとまったく同じものである．ただ，図示の仕方を少し変えてある．）

さて，ユニット間の結合の種類として，物理結合のうえに別の種類の結合を定義する．これを論理結合とよぶ．論理結合はユニット上の CPU が（あとに述べるアルゴリズムにしたがって）設定するもので，物理結合中のユニット間でも論理結合はしていない場合がある．論理結合の型の種類とその名称も物理結合と同様であるが，全く結合がないという型 "n" が加わる．これらを「論理型」とよぶ．タマネギ法では主にこの論理型をもちいてシステムの組み立てをおこなう．

5．3　核の選出

すべてのユニットが完全に同一なソフトウェアで駆動されているときに，ユニット群の中のひとつを選び出す問題が核の選出問題である．これにはソフトウェアの均質性の制約に抵触しないいくつかの方法が考えられるが，以下ではマイクロプロセッサによる実行が容易な，乱数を用いる方法を説明する．

［核の選出法］初期状態において，すべてのユニットは全く同じ状態にあるとする．まず，各ユニットは適当な方法で乱数を振り，その値を変数 m(i) に保存する（m(i) ← random(i)）（i はユニット番号．以下同じ）．次に自分の隣接ユニットすべてと通信して，それぞれの変数を調べ，もしも自分より大きな値をもつものがあれば，その値で m(i) を書き換える．この通信と書き換えの操作を決められた回数 N だけ繰り返す．その結果，もし変数 m(i) が最初に振った乱数 random(i) に一致すれば，自分の N 近傍には自分より大きな値をもったものがいなかったことが分かる．したがって，N　を十分大きく取って（たとえばユニット総数の平方根）ユニット群全体を覆うようにすれば，これで核の選出ができる．■

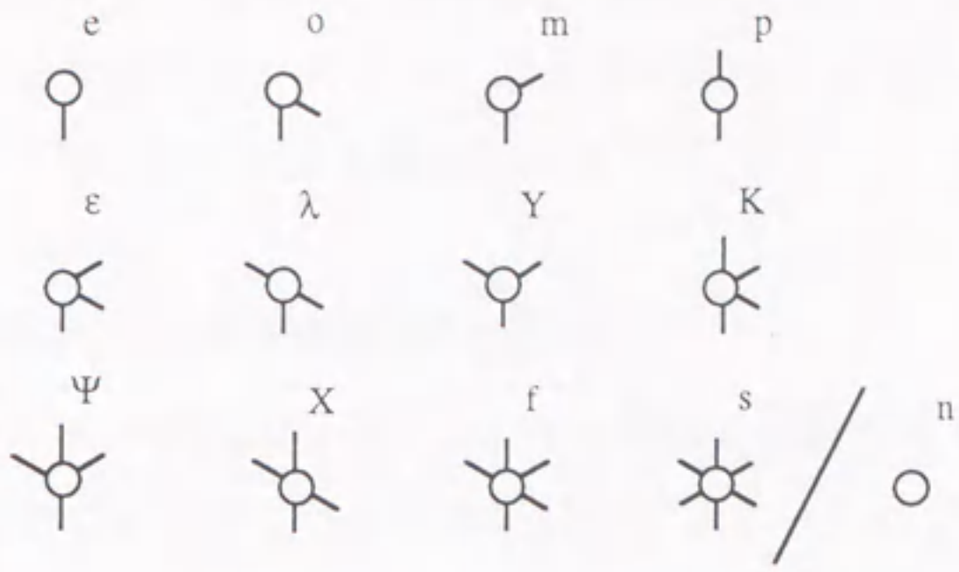


Fig. 5.1 Physical types and logical types

location index

stage	0	1	2	3	4	5	6	
0	(s)							
1	s	ε						
2	-	K	o					
3	-	f	-	o				
4	-	s	K	K	ε			
5	-	-	s	-	K	ε		
6	-	-	-	-	-	K	o	

Fig. 5.2 Discription matrix

乱数の解像度が低い場合は，同じ最大値を取るユニットが複数個存在する可能性が高いが，これを排除するためには，最大値を取ったと思ったユニットが乱数をもう一度振りなおして同じプロセスを繰り返せばよい．

5．4　記述行列　一大規模構築の設計図

すべてのユニットは記述行列とよばれる特殊な設計図をもっている[76]．記述行列には各段階においてシステム内の位置に応じてユニットがどの論理型をとればよいかが書かれている．各ユニットは段階 stage(i) と位置指標 location_index(i) という 2 つの引数をつかってこの行列を参照し，自分の論理型を決定する．記述行列は Fig.5.2 のような下三角行列で，横軸が位置指標，縦軸が段階を表す．

ユニットは初期状態では，段階 = 0，位置指標 = – 1 をもっているが（位置指標 = – 1 のときは，論理型は "n" とする），核になったユニットだけは 段階 = 0，位置指標 = 0 という値をもつことにする．

この例では最終段階で２８個のユニットが三角形を構成する手順を示している．第０段階（第０行）は，核になるユニットだけが孤立した状態を表す．第１段階（第１行）では，核は新しい６つのε型のユニットに囲まれて，論理結合のネットワークが完成する（ネットワークの完成についてはあとで定義する）．第２段階では，前の段階でε型だったユニットは K 型に変化し，新たに o 型のユニットが加わってネットワークをひとまわり大きく拡張する．記号 "-" は，前の論理型を保持するという意味である．以下同様にこの手順を繰り返したときの組み立ての過程を Fig. 5.3 に示す．（この図では，ネットワークの周囲に残っている未分化のユニットは省略してある．）

この記述法では，ある形状が唯一の記述をもつという記述の一意性は保証されない．なぜなら，ある形状を与えたとき，その形状に至るルートはいろいろ考えられ，それぞれが違った記述行列をもつからである．

また，この記述にはある論理型をとるべきユニットが何個あるかという情報は陽に含まれないが，後に述べるネットワークの整合性によって間接的に決まるようになっている．また，論理型の組み合わせや出現の順序を工夫することで，複雑な形状を表現することができる．また，対称性の高い形状では記述をコンパクトにまとめること

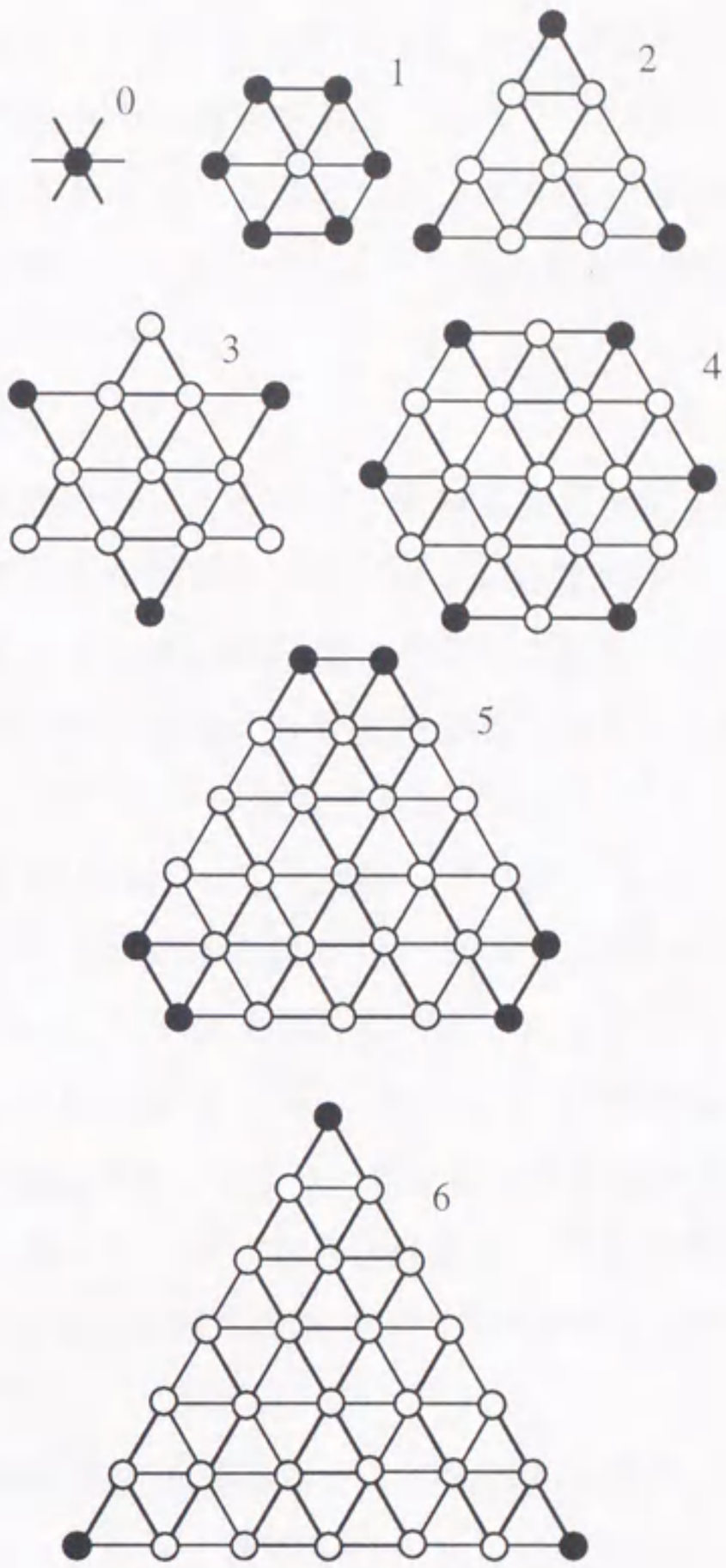


Fig. 5.3 Stages of self-assembly

ができる．さらに，複数の記述行列をポインタで組み合わせて使うこともでき，繰り返し構造などの表現も可能である[77]．Fig. 5.4 は2つの行列を用いた記述の例で，一つの行列が中央の大きな三角形を表し，もう一つが小さな三角形を表す．前者の記述の最後にはポインタがあり，後者にジャンプする．そのおわりにもポインタがあり，自分を再び読み出すようになっている．この繰り返しで図のような形状がつくられる．

5．5　段階の更新

ユニット群は記述行列にしたがって段階ごとに形状を完成して行く．ある段階が完成するとは，そのときの論理ネットワークに含まれるすべてのユニットにおいて相手の論理結合と結ばれていない論理結合がひとつもないことと定義する．そこで，これをユニット群が局所的に判定する手順が必要になる．このために各ユニットに完成度 complete なる s 次元のベクトル変数をもたせる．ただし，s は組み立ての最終段階数，ベクトルの初期値は (0,0,･･･,0) とおく．このベクトルの各要素には各段階においてネットワークがどのくらいできあがったかという（局所的な）評価値がはいる．その評価値が判定スレシホールドTHより大きくなったとき，（そのユニットにおいては）その段階が完成したと判定される．核ユニットだけは特別に初期状態で完成度=(TH+1,0,･･･,0) を設定する．つまり，核だけは第0段階をすでに完成したものとみなすわけである．したがって，核は選出されるとただちに第1段階に移行する．第1段階以降のネットワーク完成の確認は次に述べる方法で行なう．

［最小完成度伝播法[45]］あるユニットの論理結合腕が，隣のユニットの論理結合腕とちょうど対向しているとき，この腕は充足しているという．さらに，そのユニットのすべての論理結合腕が充足していることを完全充足と定義する．各ユニットが完全充足しているとき，隣接するすべてのユニットの同段階の完成度を調べ，そのうちの最小の値に1を加えたもので自分の完成度を置き換える．そうでないときはその段階の完成度を0とおく．すべてのユニットはスレシホールドを越えるまでこのプロセスを繰り返す．■

この方法によれば，ある段階のネットワークが完成すると（論理結合が過不足なく

Fig. 5.4 Example of recursive structure

閉じると），どのユニットの完成度もこのプロセスを繰り返すたびに1ずつ上がって行き，いずれはどのユニットでも完成度が判定スレシホールドを越える．つまり，あるユニットを中心とし，判定スレシホールドを半径とする円内のユニットがすべて完全充足したとき，そのユニットは次の段階に移行する．ただし，すべてのユニットが一斉に移行するわけではなく，完成したものから順に移行してゆく．

5．6　段階および位置指標の決定

ある段階のネットワークが完成すると，論理ネットワークはただちに次の段階の組み立てを開始する．つまりすでに組み立てられた部分に周りの未分化ユニット（段階 = -1 ）をつけ加えてそれを一回り大きく拡大しようとする．このとき，新たに加わるユニットに適切な論理型を取らせる必要がある．これにはそれらのユニットに段階と位置指標を教えてやればよい．（この手続きは生物の発生において，その細胞がどの細胞から由来するかという情報の受け渡しを行うことに相当する．）それは次のようにして行われる．

完成したネットワークに属するユニットは，新しい段階に移行すると，記述行列にしたがって次の論理型になり，新たな論理腕を周囲に延ばす．ネットワークの周りにいたユニットはこの論理腕を向けられると，相手の段階に自分の段階を合わせてその段階に加わろうとする．（ただし，論理腕を出している相手が既に完成したネットワークに含まれていないときは無視する．）このとき位置指標は段階と同じ値でよい．なぜなら，その段階に新たに加わるユニットの論理型は，つねに記述行列の対角部分に書かれているからである．

こうして加わった新たなユニットがネットワークに組み込まれ，この段階が完成すると，以後，そのユニットの位置指標はこのときの値に固定される．

5．7　論理型の方向探索

すべてのユニットが適切な論理型を適切な向きに置いて初めて論理ネットワークが完成するわけだが，記述行列には論理型の方向は書かれていないので，各ユニットは自分で正しい方向を見つけださなければならない．方向の探索は簡単なマッチング演算でできるが，効率の良い探索を行なうために，次のような工夫を施す．

［マッチング］各ユニットは自分の論理型の向きをひとつずつ回転させながら，どの方向が最も完全充足に近いかを判定する．具体的には，自分の論理型を表す配列と相手からの結合要求を表す配列をひとつずつシフトさせながら積和を計算し，最大値を取ったところを自分の方向とする．（もちろんこれらの配列は循環している．）

結合要求：　[0　　0　　1　　1　　0　　0]
新たに加わるユニットの論理型：
　　ずらす→[1　　1　　0　　0　　0　　0]

一般には，最大値を取る場所は複数あるが，そういった場合はそのうちの一つをランダムに選ぶ．■

しかし，全くのランダムでは，計算するたびに方向が変わることになり，その結果，ネットワークが頻繁に途切れて収束が遅くなったり，結合をさぐりあう一種のリミットサイクルに落ちこんだりする．そこで，いつもランダムに書き換えるのではなく，ある一定の確率 P_match で書き換えることにする．

また，詳しくは述べないが，論理結合要求を一方向だけから受けているユニットは，デッドロックを引き起こしやすいので，そういうユニットの論理結合を切り離す手続きも必要になる．そのようなユニットは確率 P_cut でリセットされ，未分化な状態（位置指標＝－１）に戻される．

［強結合と弱結合］すでに完成した論理ネットワークを次の段階を組み立てるときの方向探索によって壊してしまわないために，論理結合に強弱の区別を導入する．すべての論理結合ははじめ弱結合として設定されるが，ある段階のネットワークが完成した時点でそのネットワークに含まれる論理結合を弱結合から　強結合に変更する．すなわち，論理ネットワークはまず弱結合によって構築され，これが完成すると強結合に変更される．■

5．8　ユニット供給のメカニズム

ここまで，論理ネットワークの段階的な組み立ての方法について述べてきたが，段階が進むにつれ，いつかは論理ネットワークがユニット群の縁を越えて先へ延びなければならなくなる．もしもそのような場所にいつでも未分化のユニットを供給することができれば，ネットワークはユニットがある限り成長し続けることができ，ついには目標とする物理的な形状に論理ネットワークの形状が一致するであろう．

二次元機械では，システムがどのような形状になってもその周は一本の閉曲線であるという性質がある．この性質を利用すると，簡単にそのようなユニットの供給を行なうことができる．

［巡回によるユニット供給］ユニットはその物理型にしたがって，可動型と非可動型に分けられる．ここでは，"e","o","ε"を可動型とする．ユニットは自分の物理型が可動型で，かつ未分化である場合に，ある確率 P_move で右に 1 ステップころがって移動する．このようにすると，ユニット群の周上にある未分化のユニットは，群のまわりを反時計周りに巡回する．■

このルールだけでは未分化部分の形状が突起を持ちやすく，ユニットの巡回が効率的に行われない．とくにユニット群に穴ができると，その内部にユニットを供給することが難しくなる．（したがって，目標形状そのものも穴を持っていないものに制限される．）そこで，いくつかの補助ルールをくわえて，ちょうど表面張力によって液滴が収縮するように，未分化部分がなめらかな輪郭を保つようにした．

5．9　自己組み立てのシミュレーション

実際に上述のアルゴリズムを適用する場合は，各ユニットで，

0）核の選出

1）段階と位置指標の決定，論理型の決定

2）論理型の向きの探索

3）完成度の伝播と段階の移行判定

4）未分化ユニットの移動

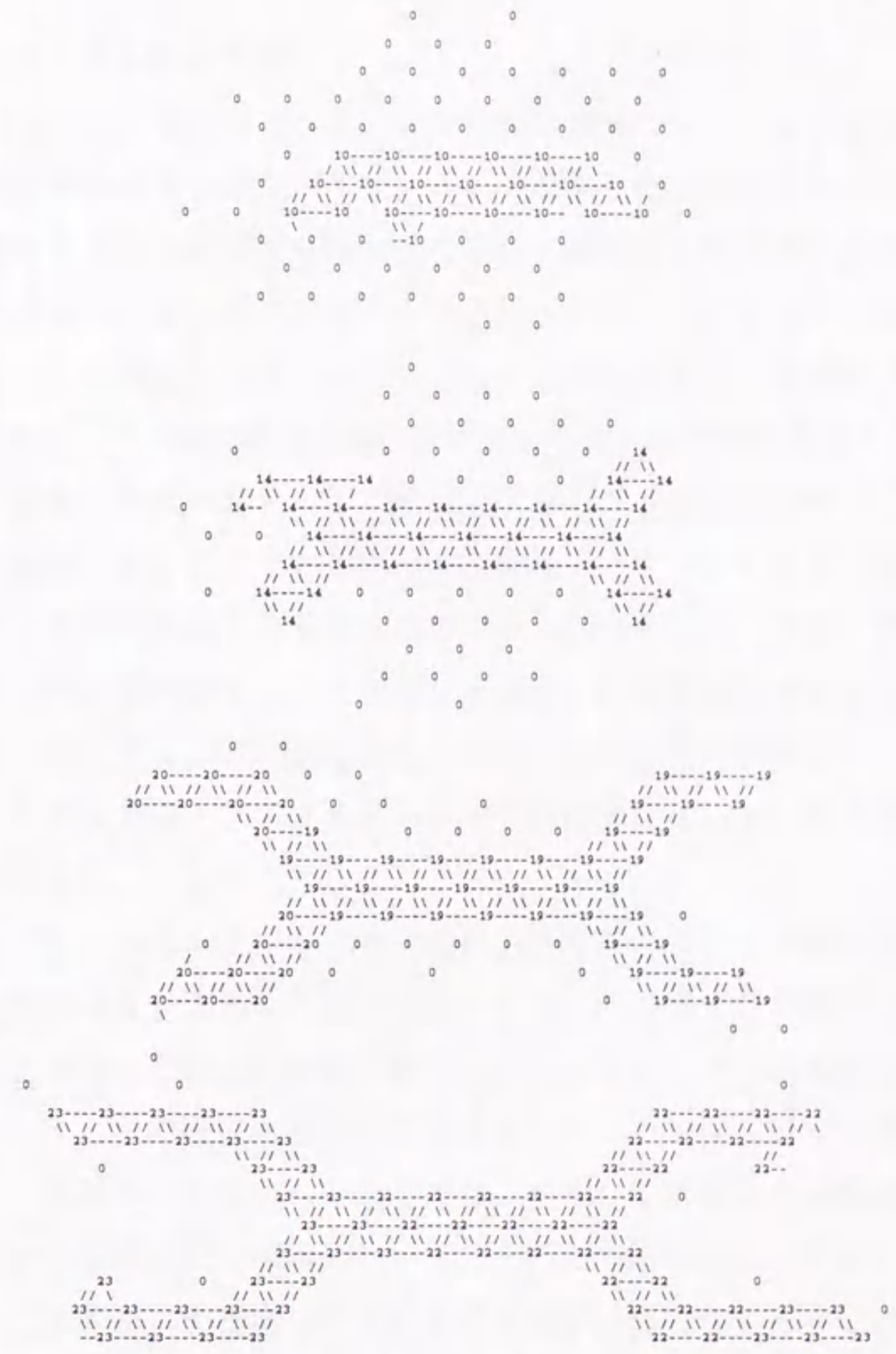
のように手順を進めることになる．各ユニットは0）を終えた後は1）から4）を繰り返すだけであり，完全に分散的，並列的なアルゴリズムとなっている．また，初期状態において，各ユニットのプログラムおよびデータは全く同一である．ユニット間でやり取りされる情報は，核選出用の乱数，物理型，段階，論理結合要求，完成度などで，これらはすべて整数型である．

アルゴリズムのなかでチューニングを必要とするパラメータは P_match, P_cut, P_move と完成度判定の TH だけであるが，これらは互いに関係し合っているので設定には若干の注意が必要である．たとえば，TH を大きく取って広い領域で完成することを要求するときは，P_match や P_move を小さく設定して論理結合の保持時間や未分化ユニットの滞留時間を延ばしてやらないと，いつまでもその段階を完成できなくなる．P_cut と P_match の間にも同様の関係がある．

以上のアルゴリズムを評価するために，いろいろな形状を目標として与え，計算機実験を行なった．まず，Fig. 5.3 の２８ユニットからなる三角形を目標形状として同じ初期状態から1000回の試行を行ったところ，そのすべてのケースで組み立てに成功した．（アルゴリズム中に乱数を用いているので，同じ初期状態からでも毎回違った経過をたどる．）このとき，最終段階に到達するまでに要した時間の平均は 480 ステップ（シミュレーションのサイクル数）であった．

次に，80 個（うち 5 個は予備）のユニットからなる比較的複雑な形状を目標として実験を行った．このときの構築の過程を Fig. 5.5 に示す．図中の数字は段階をあらわし，ひとつの数字がひとつのユニットに対応する．"０"は未分化のユニットを表す．初期状態において中央にあるのが核ユニットである．核選出の過程は省略し，最初から中央に配置してある．ユニット間にある線は論理結合を表している．この例題についても 1000 回の試行を行ったところ，そのうちの 979 回で 5000 ステップ以内の組み立てに成功した．この場合，最終段階（第 25 段階）には平均 2600 ステップで到着している．失敗したケースでは，5000 ステップ以内に組み立てができなかったか，あるいは，シ

initial configuration



final configuration

Fig.5.5 Simulated sequence of self-assembly

ステム内に穴ができたことによるデッドロックに陥っていた．これ以外の形状についても実験を行ったが，ほとんどの場合，高率で組み立てに成功した．このとき目標形状ごとにアルゴリズム中のパラメータをチューニングし直す必要はなかった．

5．10　まとめと考察

この章では，数百個のユニットを含む大規模なシステムの自己組立の方法として，段階的な構築をおこなうアルゴリズムを提案した．このアルゴリズムの主な特徴は，第4章でもちいた結合型の概念を拡張し，物理型と論理型の区別を導入したこと，記述行列によって自己組み立ての全過程をコンパクトに表現したこと，未分化部品を巡回によって供給したこと，の3つである．これにより，前章のアルゴリズムの難点をほぼ解決し，大規模で複雑な形状がきわめて高い成功率で実現できるようになった．

やや抽象的な見方をすると，前章のアルゴリズムでは，すべてのユニットはいつでも同じ手順を繰り返しているだけで自己組み立てがおこるようになっていた，つまり，空間的にも時間的にもまったく均質であったのに対し，本章で述べたアルゴリズムでは，はじめに核の選出という手順があって，その後段階を追って自己組立がおこる．つまり，ここでは，核の選出が起点となって時空間の対称性をくずしているのである．そういう意味では，アルゴリズムの均質性を犠牲にして，組み立ての効率や形状の表現力を取ったことになる．

このアルゴリズムを，もっと大きなシステム，たとえば数千個から数万個のユニットを含むようなシステムに適用することは，このままでは難しいと思われる．なぜなら，ここで用いた記述行列は，厳密にインクリメンタルに組み立てのプロセスを表していて，ひとつの段階もおろそかにできない，いわば「堅い」表現となっているからである．生物のかたちづくりなどでは，おそらく非線形の拡散場などを用いたもっとやわらかな表現方式を採用していて，そのために出来上がりのかたちにはばらつきがでるが，はるかに大規模なシステムの構築が可能となっているのであろう．（たとえば，ヒトのからだは約６０兆個の細胞からできているといわれている．）

また，ユニットの輸送に関して，ここでは2次元の機械システムに適用可能な方法を提案したが，システムが3次元になった場合，このような単純な巡回では必要な場所に必要なだけのユニットを供給することができない．この辺は今後の課題である．

第6章　自己修復のソフトウェア

6．1　はじめに

この章では，第3章以下で述べてきた機械ユニットで構成されるシステムに自己修復機能を導入する．前章まで述べてきた自己組み立てアルゴリズムは，その階層性を利用すると，ごく自然に自己修復アルゴリズムに拡張することができる[78]．ただし，システムに起こりうるさまざまな故障を一般的に扱うことはきわめて困難であるので，システムに起こりうる故障として，任意の複数個のユニットがシステムから欠落するというカテゴリの故障のみを考察する．欠落は任意の時間に任意の場所で何度起こってもよいとするが，残されたユニットはすべて正常に機能すると仮定する．システムはユニットの欠落を検出し，残った部品によって欠落部分を代替する．（すでに組み立てた部分をばらして，でてきた未分化部品を修理に充てることもある．）この過程は組み立ての段階を後戻りすることにより自然に実現できる．つまり，途中まで行なった形状の組み立てを停止し，確実に完成している段階まで後退し，それ以降の組み立てをもう一度やり直せばよいのである．

6．2　欠落の検出

ユニットの欠落の検出はすでに分化を終えて論理ネットワークに組み込まれたユニットが行なう．すなわち，これらのユニットの間で論理結合ができているはずの結合腕において相手からの通信がとだえた場合に，相手ユニットが欠落したとみなすことにする．

6．3　退化信号

システムの段階の後戻りは，欠落を検出したユニットが退化信号をまわりのユニットに送出することにより開始される．退化信号にはどの段階まで退化するかという退化の深さ（退化レベル）が書き込まれている．退化の深さとしては欠落したユニットの位置指標をもちいる．これは残ったシステムに含まれる論理ネットワーク構造のう

ち，最大の段階のものを残すためであるが，これを実行するためには，各ユニットは自分に隣接しているすべてのユニットの位置指標を記憶しておく必要がある．

6．4　段階の後戻り

欠落を検出したユニットからレベル n の退化信号が送信された場合，システム全体を少なくとも段階 n まで後戻りさせる必要がある．以下，Fig. 6.1 の記述行列を使って説明する．後戻りのためにはシステム内のユニットのうち段階が n を越えるもの（領域AとB）が，n まで段階を戻せばよい．このとき，段階 n より後に形状構成に参加したユニット，つまり領域Aは未分化の状態に戻る．領域Bではそれぞれ段階 n の論理型にもどる．また領域Cは欠落箇所が一カ所だけのときは使用されない領域である．この領域は，たとえばあるユニットがあるレベルの退化信号を受け取って退化したあと，システムの別の部分が破損し，それよりも浅いレベルの信号を送信してきた場合，その退化信号を無視して不要な伝播を終止させるために必要となる．

具体的には，この退化手続きは各ユニットが次のような処理を行なうことにより実現できる．各ユニットに対して周囲から受けとった最小の退化信号のレベルを n，自分の位置指標を m，段階を s とする．n＜s であれば，論理的に結合されている周囲のユニットにレベル n の退化信号を送信する．さらに，n と m, s の大小関係に応じて以下を実行する．

・n＜m のとき（領域A）：ユニットは初期化され，未分化の状態に戻る．位置指標と段階も初期値に戻す．
・m≦n＜s のとき（領域B）：ユニットの段階を n に戻す．したがって，論理型も段階 n のものに戻す．
・s≦n のとき（領域C）：退化信号を無視する．

以上の手続きにより，システム全体が適当な段階まで退化することができる．また，退化が終わると退化信号の伝播もとまり，信号だけがいつまでも残ることはない．

この方法では，退化信号の送信を行なったユニットはすぐに形状の組み立てを再開できるので，組み立てと並行して修復を行なうことができる．また，複数個のユニッ

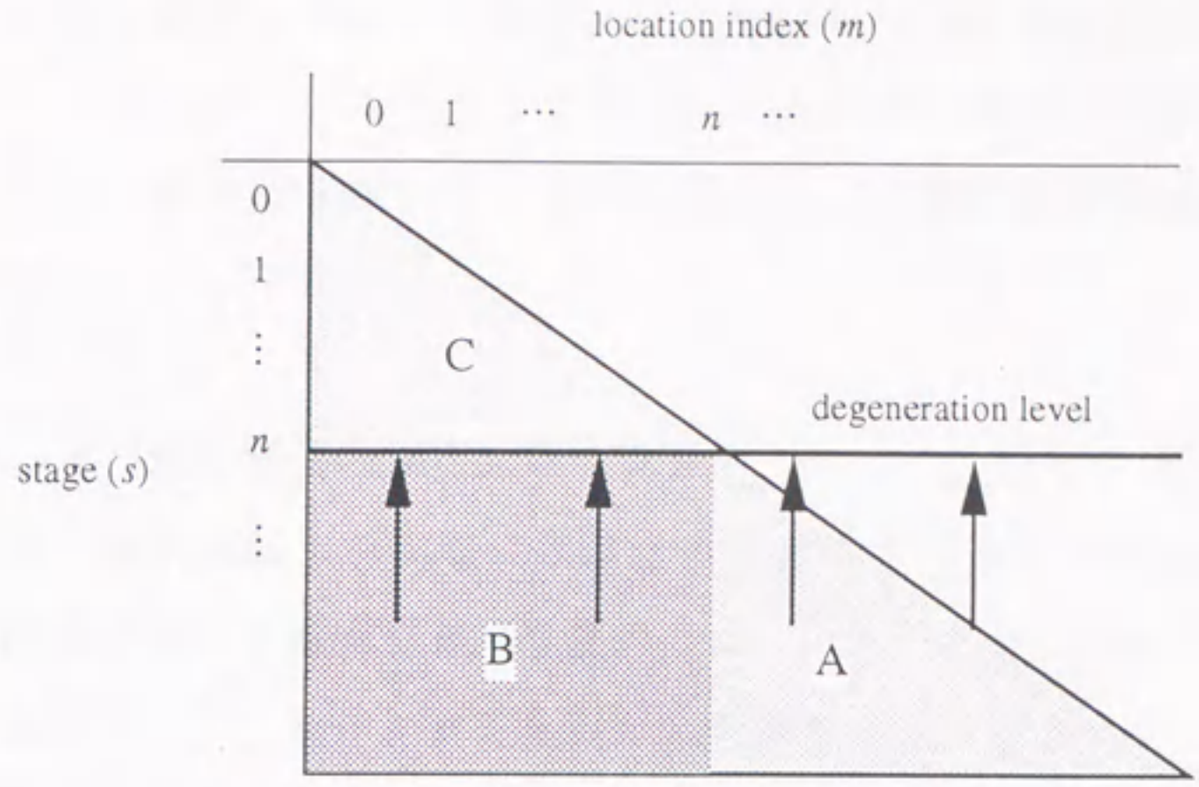


Fig. 6.1 Degeneration process on description matrix

トが立て続けに欠落した場合，いくつかのレベルの退化信号がシステム内を飛び交うことになるが，システムはそのうちの最も深いレベル（最小のレベル）に後戻りする．ただし，核を取り除いてしまった場合は，一番最初の核選出からやり直さなければならない．

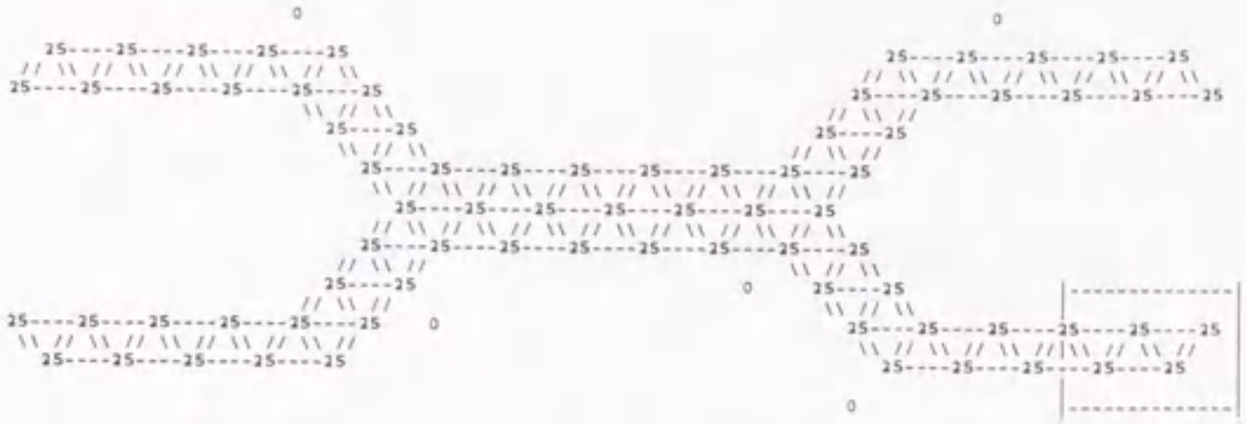
6．5　自己修復のシミュレーション

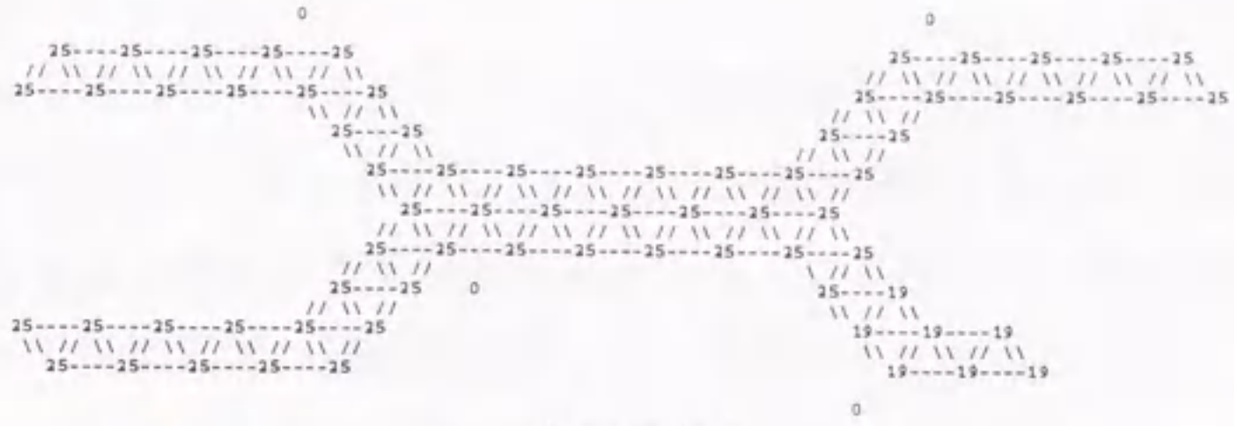
自己修復手続きを実際にユニットにインプリメントするには，欠落の検出および退化信号に関わる処理を5．9節のフローの2）と3）の間で行うようにすればよい．

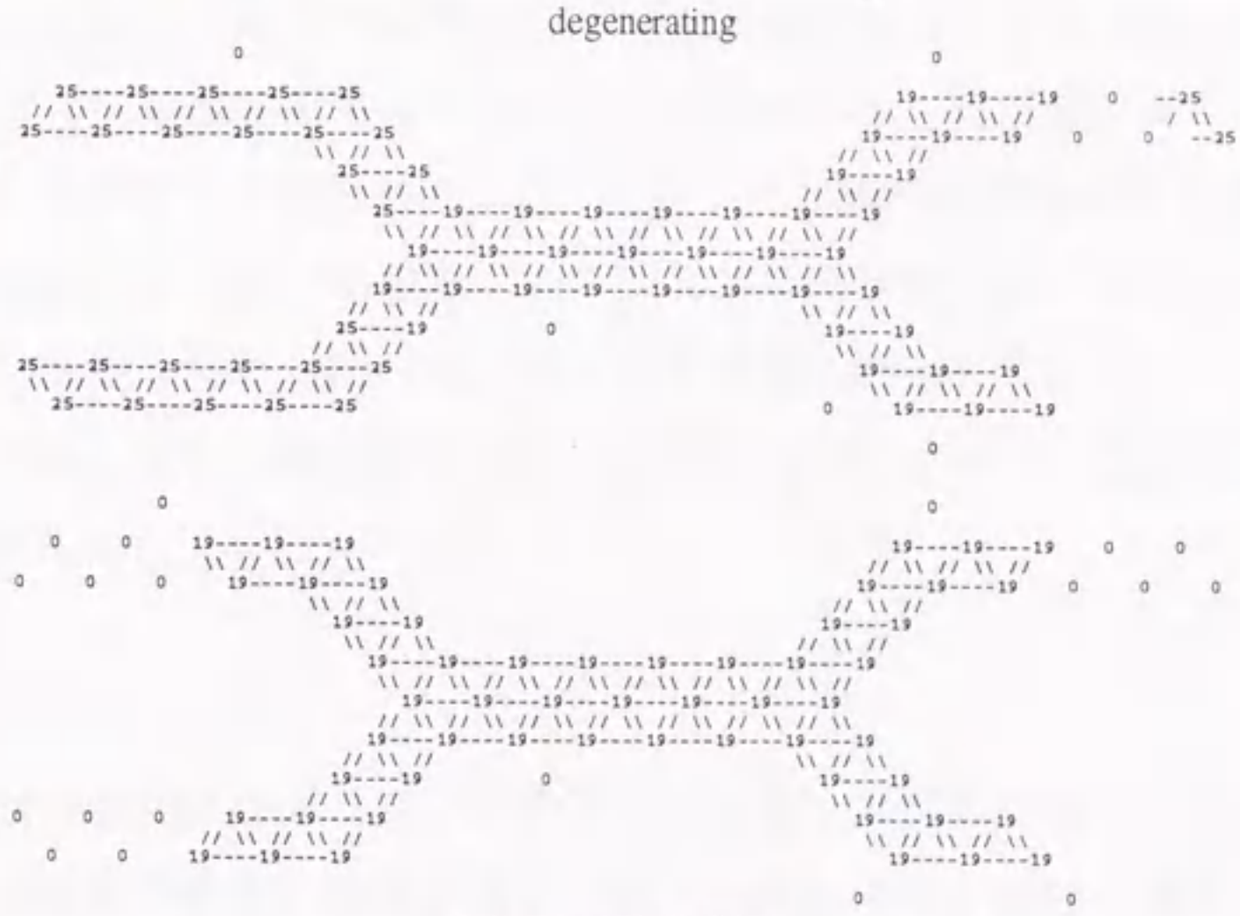
以下にシミュレーションを実行した例を示す．ここでは，前章の Fig.5.5 の形状からある部分を取り除いて自己修復を行なった（ Fig.6.2 ）．まず，一度完成した形状からユニットを取り除く．"0" のユニットは未分化の予備ユニットであり，システムのまわりを巡回している．次の図から4番目の図までは退化信号の伝播の様子を表す．取り除かれたユニットの位置指標 19 にシステム全体が段階を戻そうとしている．退化終了後の形状構成は通常の場合と全く同様に行なわれ，もとどおりの形状が再構築される．

6．7　まとめと考察

本章では，前章までに述べた階層的な自己組み立てアルゴリズムを拡張し，任意箇所の欠落に対応する「自己修復」アルゴリズムを導いた．ここで提案する自己修復は，欠落を検出したユニットの退化信号の発信，および退化信号を受け取ったユニットの退化手続きだけからなるきわめて単純なもので，前章の記述行列の表現をうまく生かして，アルゴリズムをあまり複雑にせずに，任意の箇所の任意回の欠落に対処できるようにしたものである．ただし，ここで扱っているシステムの故障は，ユニットの欠落だけであり，残ったユニットはすべて正常であるという仮定が入っている．もっと一般的な故障に対処するためには，まず，ユニットの故障を自動的に検知・検出することが必要となるが，ユニットに起こりうる故障のパターンはいろいろあって，それを自動的に検出することはきわめて困難である．ただ，ここで対象としているユニットは構成が単純で，機能も結合・離反だけであるから，結合・離反の動作ができるか，また，通信が正常に行えるかどうかだけを判断材料に，かなりの程度故障の診断ができるとも思われる．その結果，故障の可能性ありと判定されたユニットは，すべてシ







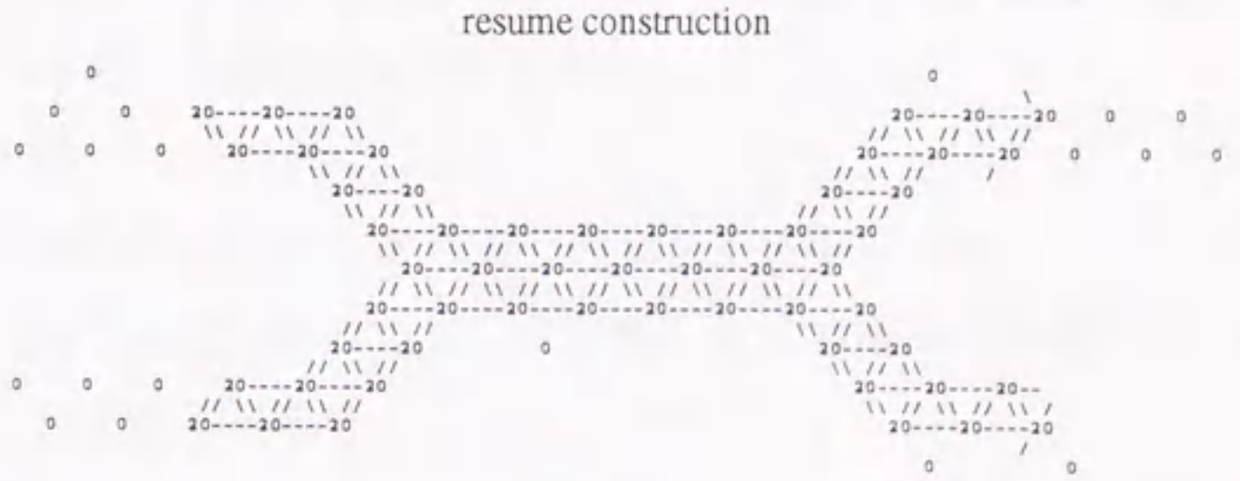


Fig.6.2 Simulated sequence of self-repair

ステムから切り離すという戦略も考えられるだろう．このような疑わしきは全部罰するという戦略は，無駄が多いかもしれないが，現実的・妥協的な解のひとつではあろう．

最後に，ここで提案した自己修復の方法論をこれまで考えられてきた方法論のなかに位置づけてみたい．

高田は現在の機械システムから自己修復機械の実現に至る道のりを順序よく整理して示している[79]．彼によれば，システムの耐故障性を高める必要があるときには，自己修復を考える前に，まず冗長性を高めることを考えるのが普通である．これには大きくいって要素冗長と機能冗長がある．要素冗長は故障が予想される部品をあらかじめ複数個用意しておく方法であり，機能冗長はシステムの一部が故障しても，別の形態によって同じ機能を果たそうとする方法である．航空機は冗長化を徹底的におこなっているシステムの代表例で，たとえば3組の慣性航法装置による多数決（要素冗長）や自動操舵系を手動でも操作できるシステム（機能冗長）などが採用されている．しかし，冗長性を高めただけで，あらゆる故障に対処することはできない．そこで自己修復の登場となる．高田は吉川[80]の議論を参照しながら，自己修復には，つぎの4つの段階があると指摘している．

1）ポテンシャル型

自動調心軸受けのように，ポテンシャル最小の状態で機能を発揮するシステム．何らかの要因で状態に偏差を生じても，自然に正常な状態に復帰する．システムの構造そのものの安定性を利用する方法．

2）制御型

制御可能な状態空間をあらかじめ大きく確保しておくことにより，たとえシステムが劣化しても適当なパラメータ制御によって機能の回復が行える．機能冗長の考え方もこれに含まれる．

3）自動保全型

故障した部品の交換や補修作業を保全ロボットなどが自動的に行う．

4）自己再生型

自分自身をつくり出す機能を有する，真の意味での自己修復システム．

さて，我々のアプローチ，すなわち，均質型ユニットシステムはこの中のどの辺にあるだろうか．

冗長性の観点からは，全体が均質なユニットからなるということは，要素冗長の極限であり，しかも，システムの機能が要素の組み合わせだけで実現されるため，同時に機能冗長にもなっているといえるだろう．また，これまでに実現された工学システムの修復の方法論の段階を考えてみると，そのほとんどが耐故障性を高めるところまでで止まっており，真の自己修復の段階に至っているものは皆無といえる．上の定義で言えば，ポテンシャル型か制御型がせいぜいであり，自動保全型の例も見あたらない．それはある意味で当然のことであって，さまざまな機能を持つ実用システムを相手に自己修復機能の実現することは，現状のテクノロジーではまだ手の届かない夢物語であるということであろう．

一方，本研究のアプローチは，当面の実用性はいったんあきらめはしたが，システム全体を均質なユニットで構成するという根本的に異なる発想に立ったため，修復の段階としては，自己保全型と自己再生型の中間ぐらいに到達していると考えられる．つまり，ユニットがユニットそれ自身をつくり出すこと（自己増殖）はできないが，あらかじめユニットを十分に用意しておけば，任意の構造物の組み立てや修復を自動的におこなうことが実現できるのである．

第7章　終　章

7．1　2つのモデルシステム一反省

本論文では，均質型の機械システムのモデルとして，ソフトリンクビークルシステムとユニット型機械システムの2つを考え，これらの設計の実例から，大規模システムの新しい設計論の糸口をつかむことをもくろんだ．ここで取り上げた2つのモデルシステムは，どちらも同じ均質システムではあるが，その情報処理システムとしての構造は「集中」と「分散」に分かれていた．すなわち，ソフトリンクビークルシステムが均質ではあっても，いわば従来の集中制御システムの延長であったのに対し，3章以降で述べた均質ユニットシステムは完全な分散型システムであった．

ソフトリンクビークルシステムで集中型の制御方式を採用したのは，走行中の車間距離を一定に保つという，きわめて高い速応性を実現するためである．ここに示した集中型の設計は，車間距離や速度の安全領域が定量的に把握でき，車両間の通信も局所的かつ一方向でよいなどの優れた特性を持っている．しかしその一方で，耐故障性の問題に対しては本質的な解決策を見いだすことができなかった．このように，実用システムを視野に入れた開発では，集中型の構成をとることによって，解析が容易になるなどのメリットがあるが，耐故障性の問題には，かえってアプローチがしにくくなるという面もあるのである．

つぎに取り上げたモデル，均質ユニットシステムでは，この耐故障性の問題に新しい方法論によってアプローチすることを目指した．均質ユニットシステムは，その名前通り，完全に均質な構成をもち，情報処理の形態も完全に分散的なシステムである．このようなシステムでいったい何ができるか，という問いかけが，このモデルシステムの設計の動機であり，主題であった．そして，かたちの自己組み立てや自己修復などの機械システムの基本的な機能を実現する新しい方法論が示された．

7．2　残された課題

ここでは，特にユニット型機械システムに関して，今後取り組むべき課題を整理し

てみたい．

本論文に示した方法によって，自己組み立てや自己修復という機能をまがりなりにも実現できたわけであるが，これはあるひとつの方法でもって（強引に）望みの機能を実現したという段階であって，最適性や安定性などの定量的な評価はこれからの仕事である．これをさらに一般化したり，もっと多様な機能を実現しようとすれば，いろいろ多面的に検討を進める必要がでてくることはいうまでもない．

まず，ソフトウェアの面からは，数理的な定式化の必要性があげられる．たとえば，ここで提案した自己組み立てや自己修復のプロセスを一本の微分方程式で表現することができれば，かなり見通しのよい設計ができるようになると思われる．エネルギー代謝の問題などは，システムの上にエネルギーという量を定義しなければならないので，ユニットの運動をもっと別な方法で，より抽象的に表現しなければならないだろう．また，これまで手つかずだったシステム全体に一定の運動をさせるというような機能は，決まったかたちになるだけの自己組み立てや自己修復より一段上の複雑さをもっており，ここでもソフトウェアの数理的・理論的定式化はどうしても必要になるであろう．

このような数式化における主要な困難は，システムの構造（要素間のつながり方）そのものが変化するような問題を記述・解析するための適当な数学的道具がほとんど見あたらないということにある．ユニットの運動によってシステムのトポロジーそのものが時間とともにどんどん変化してしまうという極端な非線形性は，これまでほとんど扱われておらず，したがって，その解析手法もまったくといってよいほど確立されていないのである．では，どのような数理的手法が有効かはまだ何とも言えないが，ユニットの運動を表現する形式の大胆な単純化や統計力学的視点の導入などが必要となるのではないかと思われる．

つぎにハードウェアの面を考えてみると，環境とのインタラクションを実現するために，何らかのセンサーをユニットに搭載することが考えられる．手始めに，各ユニットに接触センサーを取り付けて，システム全体が壁に沿って這う機能であるとか，細い穴をすり抜ける機能であるとかを実現するという問題などが考えられる．また，システム全体が運動を生成するためには，現行のユニットのハードウェアを大幅に設計し直すことも必要になろう．運動を実現するためには，一種類だけのユニットをも

ちいる現在の構成は不適当で，ある程度分化した数種類のユニット（たとえば，収縮する筋肉ユニットと関節ユニット等）の存在を仮定しなければならないのかもしれない．システムの工学的な応用を考えた場合，避けて通ることができないのが，ユニットの3次元化，マイクロ化そしてエネルギー供給の問題である．3次元ユニットのアイデアについては，第3章で簡単に触れたが，マイクロ化やエネルギーの問題についてはまだ十分な検討を行っていない．これらの問題は当面本質的でないとして扱わなかったのであるが，これらはいわばテクニカルな問題であり，さまざまな技術の地道な積み重ねによって解決してゆくほかないであろう．

7．3　新しい設計論の創造に向けて

われわれの身の回りのあらゆる人工物は，われわれの欲する機能を実現するために，自然の法則を応用したさまざまな工夫ーそれを工学と呼ぶーによって，設計されている．そういう意味で「設計」という言葉を使う場合，その言葉の意味は明瞭である．少し抽象的な言い方をするならば，はじめにわれわれがその人工物によって実現したい何らかの「意図」があり，それはまず「仕様」という論理的・抽象的な空間に表象される．さらにそれが細かく分解・還元されて具体的な「もの」の世界，つまり物理空間に投影される．その投影とは，つまり単機能の部品であり，目的とする人工物はここまでの過程をちょうど逆にたどることによってそれらの部品から合成される，というのが設計という言葉の指す内容である．ここでは設計者の意図が，人工物を構成するすべての部分とそれらの間の関係を直接・間接に規定しており，つまるところ人工物は設計者の意図の一方的な反映に他ならないと言える．

ところが，自然物，特に生物においては，設計者の意図のような明示的なプロットはどこにも存在せず，全体の機能はただその無数の部分の絶妙なる配合・協調によって具現されているように感じられる．このような生物と人工物の違いをひとことで言えば，生物がひとりでに「なる」システムであるのに対し，人工物はあくまでも人間がそのように「する」システムであるということになろう．

さて，序章で述べたように，人工物の構造が複雑化の一途をたどり，さまざまな問題点を露呈しつつある今日，人工物とは桁違いの複雑性をもった自然物が，人工物には及びもつかないさまざまな高度な機能を達成しているのをまのあたりにすると，そ

こに新たなものづくりの原理を見いだしたくなるのは当然のことである．生物は自分と同じものを自ら作り出す自己複製の能力を内包しているし，あらゆる環境の変化に適応し，さらに高度に進化してゆくことができる．これまで，生物特有と考えられてきたこれらの機能こそ，これからの人工物の目指しているものではないだろうか．

本論文は，人工物の設計に，生物的な構成を導入することによって「する」設計から「なる」設計への脱皮を図ったものである．多数の均質な要素による構成は，生物が多数の細胞から構成されていることのアナロジーであって，自己組み立てや自己修復の機能はまさしく「なる」システムへの第一歩だったわけである．このような歩みを一歩一歩着実に進めてゆくことにより，いつの日か適応や進化の本質にも迫ることができるのではないかと考える．

参 考 文 献

[1] 田村担之：大規模システム，昭晃堂 (1986)

[2] 深尾　毅：分散システム論－熱力学的システム論－ 昭晃堂 (1987)

[3] 桜井　淳：崩壊する巨大システム，時事通信社 (1992)

[4] 遠藤　浩：飛行機はなぜ落ちるか－設計者から見た航空システムの安全性，講談社 (1994)

[5] G.v. Bochmann: 分散処理システムデザイン，工学社 (1987)

[6] J. von Neumann: Theory of self-reproducing automata, Univ. of Illinois Press (1966)

[7] E.Schlödinger : WHAT IS LIFE — The Physical Aspect of the Living Cell, Cambridge Univ.Press (1944)

[8] J.D.Watson, F.H.C.Crick : Molecular structure of nucleic acids. A structure for deoxyribose nucleic acid, Nature, 171, 737/738 (1953)

[9] B.Alberts et al. editor : Molecular Biology of the Cell (3rd edition), Garland Pub. Inc. (1994)

[10] L.S.Penrose: Self-reproducing, Sci. Amer., 200-6, 105/114 (1959)

[11] W.Poundstone : The Recursive Universe, Cosmic Complexity and the Limits of Scientific Knowledge (1985)（有澤　誠訳：ライフゲイムの宇宙，日本評論社）

[12] S.Wolfram : Cellular Automata and Complexity, Addison-Wesley Pub. Co. (1994)

[13] M.M.Waldrop : Complexity, The Emarging Science at the Edge of Order and Chaos, Sterling Lord Literistic Inc., New York (1992)

[14] 特集：自律分散システム，計測と制御，29-10，877/951 (1990)

[15] 特集：自律分散システムの新たなる展開，計測と制御，32-10，789/861 (1993)

[16] 伊藤，市川，須田編：自律分散宣言－明日を拓くシステムパラダイム，オーム社 (1995)

[17] 伊藤正美：自律分散システムはいかにして構成されるか，計測と制御，29-10，877/881 (1990)

[18] 伊藤正美：自律分散システム研究の課題と将来，計測と制御，32-10，789/796 (1993)

[19] 津川，村田，広瀬，谷田部：車両間通信による自律車両群の走行制御，計測自動制御学会論文集，26-9, 1058/1065 (1990)

[20]津川定之：自動車交通のインテリジェント化－欧米日のプロジェクトの現状，システム／制御／情報，39-5，221/218 (1995)

[21]津川定之：交通運輸システムのインテリジェント化，交通工学，31-6，41/50 (1996)

[22]E.Fiala: Zweckmäßigkeit automatischer Fahrzeugführung, Straße und Autobahn, Heft 3, 113/114 (1990)

[23] K.S.Chang, et al.: Automated Highway System Experiments in the PATH Program, IVHS Journal, 1(1), 63/87 (1993)

[24] J.K.Hedrick, et al.: Control Issues in Automated Highway Systems, IEEE Control Sytems, December 1994, 21/32 (1994)

[25]（財）自動車走行電子技術協会：自動車通信に関する研究－通信方式に関する基礎実験－ (1986)

[26](財)自動車走行電子技術協会：移動体対応型情報システムに関する調査研究－車々間通信技術の応用－ (1989)

[27] 津村俊弘：同輸制御についての一提案，第10回ビークルオートメーションシンポジウム講演論文集，25/28 (1987)

[28] 吉川，岡，吉川，花房：環状線路上の車両群の分散制御，システムと制御，24-10, 690/698 (1980)

[29] S.Tsugawa, S.Murata, T.Yatabe, T.Hirose:Vehicle Following System Using Vehicle-to-Vehicle Communication —Its Concept, Control Algorithm, and Communication System, Proc. ASME/ISCIE USA-JAPAN Symposium on Flexible Automation, 621/628 (1988)

[30] T.Yatabe, T.Hirose, S.Tsugawa, S.Murata: Control of Autonomous Vehicles with Vehicle-to-Vehicle Communication, Proc. IFAC Control, Computers, Communications in Transportation, 553/558 (1989)

[31] 津川ほか：後輪回転数を用いた走行軌跡の自動記録手法，自動車技術会論文集, 23, 91/98 (1981)

[32] S.Tsugawa: Traveling Path Measurement for Navigation of Robot Rover by Precise Counting of Wheel Revolutions, ACTA IMEKO 1985, 5, 41/48 (1986)

[33] 村田，広瀬：差動オドメタによる無人搬送車のデッドレコニング，機械技術研究所所報，47-1, 1/9 (1993)

[34] S.Murata, T.Hirose: Onboard Locating System of Autonomous Vehicle, Proc. IEEE/RSJ Intl. Workshop Intelligent Robots and Systems '89, 228/234

[35] S.Murata, T.Hirose: Onboard Locating System Using Real-Time Image Processing for a Self-Navigating Vehicle, IEEE Trans. Industrial Electronics, 40-1, 145/154

[36] 村田，津川：車両群の速度制御，第 1 回アドバンティシンポジウム講演論文集，23/26 (1986)

[37] 津川，村田：自律車両の操舵アルゴリズム，システム制御情報学会論文誌，2-10, 360/362 (1989)

[38] 津川，村田：車両間通信を用いた車両群のロンジチュージナル制御，機械技術研究所所報，47-2, 8/16 (1993)

[39] 村田，安藤，鈴木：マルチタイムスケール法によるハイゲインレギュレータの設計，計測自動制御学会論文集，Vol.23, No.11, 1158/1164 (1987)

[40] S.Murata, Y.Ando, M.Suzuki: Design of a high gain regulator by the multiple time scale approach, Automatica, Vol.26, No.3, 585/591 (1990)

[41] S.Murata, H.Kurokawa, S.Kokaji: Self-Assembling Machine, Proc. IEEE Robotics Automation, 441/448 (1994)

[42] 村田，黒河，小鍛治：自己修復する機械，計測自動制御学会論文集 31-2，254/262 (1995)

[43] 梅谷，広瀬：ほふく（匍匐）の生物力学的研究，計測自動制御学会論文集，11-1, 20/24 (1975)

[44] 市川芳明：一次元自己増殖機械，日本ロボット学会誌，10-2，266/272 (1992)

[45] 小鍛治繁：多自由度機構と分散制御，精密工学会誌，54-10，1921/1926 (1988)

[46] 川内，稲葉，福田：セル構造化ロボットシステムに関する研究，日本ロボット学会誌，12-1，116/132 (1994)

[47] 福田，中川：動的再構成可能ロボットシステムに関する研究（第 1 報，セル間の自

動接近・結合・分離制御実験），日本機械学会論文集（C編），55-509，114/118 (1989)

[48] 福田，ほか：動的再構成可能ロボットシステムに関する研究（第3報，セル構造化ロボット"CEBOT"の認識，通信システム）日本機械学会論文集（C編），56-523，709/716 (1990)

[49] 福田，ほか：動的再構成可能ロボットシステムに関する研究（第5報，ロボットの自己組織化エンドエフェクタの概念とその機構と制御に関する研究）日本機械学会論文集（C編），57-536，1302/1309 (1991)

[50] 福田，ほか：動的再構成可能ロボットシステムに関する研究（第14報，視覚と力のアクティブセンシングを用いた自己組織型マニピュレータの組立作業誤差修正）日本機械学会論文集（C編），59-565，2788/2795 (1993)

[51] 福田，植山ほか：動的再構成可能ロボットシステムに関する研究（第21報：シリーズIIIによる自動接近・結合・分離実験），ロボティクスメカトロニクス講演会'90講演論文集，345/348 (1990)

[52] G.S.Chirikjian: Kinematics of Metamorphic Robotic System, Proc. IEEE Intl.Conf.Robotics and Automation, 449/455 (1994)

[53] A.Pamecha, G.S.Chirikjian: A Metamorphic Robotic System, IEEE Intl.Conf.Robotics and Automation, Video Proceedings (1995)

[54] C. G. Langton: Self-reproduction in cellular automata, Cellular Automata (1984)

[55] 浅間，ほか：通信を用いた分散的管理に基づく複数の自律型ロボットの協調的作業分担決定法，日本ロボット学会誌，10-7，955/963 (1992)

[56] M. J. Mataric: Minimizing complexity of controlling a mobile robot population, IEEE Int. Conf. of Robotics and Automation, 830/835 (1992)

[57] G. Beni, S. Hackwood: Coherent swarm motion under distributed control, Proc. DARS'92, Wako, Japan, 39/52 (1992)

[58] 村田ほか：分散型機械システムの設計（第6報）－3次元自在結合システム－，第9回自律分散システムシンポジウム資料，21/24 (1997)

[59] 小鍛治，村田，黒河，鈴木：分散アクチュエータ群の協調制御技術，計測と制御，31-11,1131/1136 (1992)

[60] 小鍛治，村田，黒河，鈴木：自律分散機械と情報処理，情報処理，33-6, 665/672 (1992)

[61] 小鍛治，村田，黒河，鈴木：やわらかい機械におけるモジュール化，自己組織化，精密工学会誌，57-12, 33/36 (1991)

[62] 長田　正編著：自律分散をめざすロボットシステム，オーム社 (1995)

[63] 石井威望編：マイクロマシンの衝撃，PHP研究所 (1990)

[64] 藤正　巌：見えない機械－細胞の構造とマイクロマシン，オーム社 (1994)

[65] 池上，斉藤編：生物と協同現象，学会出版センター (1976)

[66] 伏見，野田編：自己組織化，学会出版センター (1977)

[67] 郷，和田編：形態形成，学会出版センター (1977)

[68] 土井洋文：生物のかたちづくり，サイエンス社 (1988)

[69] G.S.Chirikjian, A.Pamecha, I.Ebert-Uphoff: Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots, J.Robotic Systems, 13-5, 317/338 (1996)

[70] 村田ほか：分散機械システムの設計（第2報），第5回自律分散システムシンポジウム資料，19/24 (1994)

[71] 塩川光一郎：分子発生学，東京大学出版会 (1990)

[72] 岡田節人：からだの設計図，岩波新書 (1994)

[73] 柳澤桂子：卵が私になるまで－発生の物語－，新潮選書 (1993)

[74] 塚谷祐一：植物の見かけはどう決まる－遺伝子解析の最前線－，中公新書 (1995)

[75] 村田，富田，黒河，小鍛治：機械の自己組み立てと自己修復，計測自動制御学会論文集（掲載予定）

[76] 村田ほか：分散機械システムの設計（第3報），第6回自律分散システムシンポジウム資料，265/270 (1994)

[77] 村田ほか：分散機械システムの設計（第5報），第8回自律分散システムシンポジウム資料，307/312 (1996)

[78] 富田ほか：分散機械システムの設計（第4報），システム情報関連合同シンポジウム資料，15/20 (1995)

[79] 高田祥三：自己診断・自己修復，精密工学会誌，57-12, 43/46 (1991)

[80] 吉川弘之：信頼性工学，コロナ社 (1979)

論 文 発 表 目 録

No.	章	著 者 名	論 文 題 目	掲載誌名
1	2	津川定之，村田　智，広瀬武志，谷田部照男	車両間通信による自律車両群の走行制御	計測自動制御学会論文集，Vol.26, No.9, 1058/1065, 1990
2	2	S.Murata, T.Hirose	Onboard locating system using real-time image processing for a self-navigating vehicle	IEEE-Trans.IE, Vol.40, No.1, 145/154 1993
3	2	津川定之，村田　智	自律車両の操舵アルゴリズム	システム制御情報学会論文集，Vol.2, No.10, 360/362, 1989
4	3,4	村田　智，黒河治久，小鍛治繁	自己修復する機械－分散ユニット構成による自己組み立て－	計測自動制御学会論文集，Vol.31,No.2, 254/262，1995
5	5,6	村田　智，富田康治，黒河治久，小鍛治繁	機械の自己組み立てと自己修復	計測自動制御学会論文集，Vol.33,No.5, 424/432，1997
6	3,4	S.Murata, H.Kurokawa, S.Kokaji	Self-reconfigurable machine	IEEE-Trans.SMC, 投稿中
7	5,6	村田　智，小鍛治繁	ユニット型機械の可能性	計測と制御，Vol.35, No.7, 550/553, 1996
8	2	富田康治，村田　智，津川定之	マシンビジョンによるビークルのプレビューラテラル制御	システム制御情報学会誌，Vol.7, No.2, 35/41, 1994
9	5	S.Kokaji, S.Murata, H.Kurokawa, K.Tomita	Clock Synchronization Mechanisms for a Distributed Autonomous System	J.Robotics & Mechatronics, Accepted
10	3	小鍛治繁，村田　智，黒河治久，鈴木章夫	分散アクチュエータ群の協調制御	計測と制御，Vol.31, No.11, 1131/1136, 1992
11	3	小鍛治繁，村田　智，黒河治久，鈴木章夫	自律分散機械と情報処理	情報処理，Vo. 33, No.3, 665/672, 1992
12	3	小鍛治繁，村田　智，黒河治久，鈴木章夫	やわらかい機械におけるモジュール化，自己組織化	精密工学会誌，Vol.57, No.12, 2113/2116, 1991
13	3,4	S.Kokaji, S.Murata, H.Kurokawa, A.Suzuki	Self-organization of a module structured machine	J. Robotics and Mechatronics, Vol.4, No.2, 1992
14	2	村田　智，安藤嘉則，鈴木正之	マルチタイムスケール法によるハイゲインレギュレータの設計	計測自動制御学会論文集，Vol.23, No.11, 1158/1164,1987
15	2	S.Murata, Y.Ando, M.Suzuki	Design of a high gain regulator by the multiple time scale approach	Automatica, Vol.26, No.3, 585/591, 1990

国 際 会 議 発 表 目 録

No.	章	著 者 名	論 文 題 目	国際会議名
1	2	S.Tsugawa, S.Murata, T.Yatabe, T.Hirose	Vehicle following system using vehicle-to-vehicle communication	US-Japan Symp. FA, San Fransisco, 621/628, 1989
2	2	T.Yatabe, T.Hirose, S.Tsugawa, S.Murata	Control of autonomous vehicles with vehicle-to-vehicle communication	CCCT, Paris, 553/558, 1989
3	2	S.Tsugawa, S.Murata	Velocity control for vehicle following through vehicle/vehicle communication	ISATA, Italy, 1990
4	2	S.Murata, T.Hirose	Onboard locating system of autonomous vehicle	IEEE-IROS, Tsukuba, 228/234, 1989
5	3,4	S.Murata, H.Kurokawa, S.Kokaji	Self-Assembling Machine	IEEE-ICRA, San Diego, 441/448, 1994
6	3,4 5,6	S.Murata, H.Kurokawa, K.Tomita, S.Kokaji	Self-Assembly method of mechanical structure	AROB, Oita, 55/58, 1996
7	3	H.Kurokawa, S.Murata, S.Kokaji	A decision making network for cell structured machine	DARS, Saitama, 53/58, 1992
8	3,4	S.Kokaji, S.Murata, H.Kurokawa	Self organization of a mechanical system	DARS, Saitama, 237/242, 1994 (Springer Verlag)
9	5,6	K.Tomita, S.Murata, E.Yoshida, H.Kurokawa, S.Kokaji	Reconfiguration method in distrubuted mechanical system	DARS, Saitama, 17/25, 1996 (Springer Verlag)
10	2	K.Tomita, S.Murata, S.Tsugawa, E.Stuck	Visual navigation of an autonomous vehicle along a line	US-Japan Symp. FA, San Fransisco, 1321/1327, 1992

そ の 他 の 資 料

No.	章	著 者 名	資 料 題 目	資 料 名
1	2	村田　智，広瀬武志	差動オドメタによる無人搬送車のデッドレコニング	機械技術研究所所報, Vol.47, No.1, 1/9, 1993
2	2	津川定之，村田　智	車両間通信を用いた車両群のロンジチュージナル制御	機械技術研究所所報,Vol.47, No.2, 38/46, 1993

各章と文献の対応

章	文献番号		
	論文（学会誌）	国際会議	その他
1			
2	1, 2, 3 , 8, 14, 15	1, 2, 3, 4, 10	1, 2
3	4, 6, 10, 11, 12, 13	5, 6, 7, 8	
4	4, 6, 13	5, 6, 8	
5	5, 7, 9,	6, 9	
6	5, 7	6, 9	
7			

謝　　辞

この論文は，私が１９８６年に通商産業省工業技術院機械技術研究所に入所以来，およそ十年の間，従事してきた均質型機械システムに関する研究をまとめたものです．

はじめに，この研究を学位論文としてまとめる機会を与えて下さった名古屋大学工学部の鈴木正之教授にこの場を借りて深い感謝の気持ちを表したいと思います．鈴木先生からは，理論家らしい高い見地から多くの本質的な御教示，御指導を賜りました．

また，名古屋大学工学部の松崎雄嗣教授，福田敏男教授の両先生にも心から感謝いたします．松崎先生には，全体を通して，論文の構成や研究の位置づけなどについて貴重な御意見をいただきました．福田先生は，ロボット工学・マイクロ工学の第一人者として著名ですが，分散ロボットシステムの分野でも早い時期から研究を始められた先駆者のひとりであります．その福田先生に今回直接ご指導していただくチャンスに恵まれたことは大きな喜びでした．先生からもたくさんの建設的なコメントをいただきました．

ソフトリンクビークル関係の研究を指導して下さった，機械技術研究所物理情報部知識工学研究室長の津川定之博士，同谷田部照男主任研究官，広瀬武志博士（現日本大学教授）にも感謝いたします．研究者としての物事の考え方や仕事の進め方など，３人の教えて下さった研究のイロハは，私の大きな財産になっています．

ユニット型機械システム関連の研究は現在の私のメインテーマでありますが，このテーマは，機械技術研究所物理情報部システム工学研究室長の小鍛治繁博士が与えて下さったものです．私は小鍛治博士のフラクタルマシン（多自由度機構）の研究に魅せられて，この道に入りました．以来，７年余，陰に日向にいつもかわらぬ暖かい指導をいただきました．本研究の「ココロザシ」，すなわち，ユニットの厳密な均質性や近傍通信などの制約を守りながら，タネとしてはなるべく少ない情報を入れて複雑なシステムをつくってやろう，という考えは，小鍛治博士の研究理念を受け継いだものに他なりません．私に何ほどのことができるかわかりませんが，これからもこの方針を貫いて研究を進めていきたいと思います．

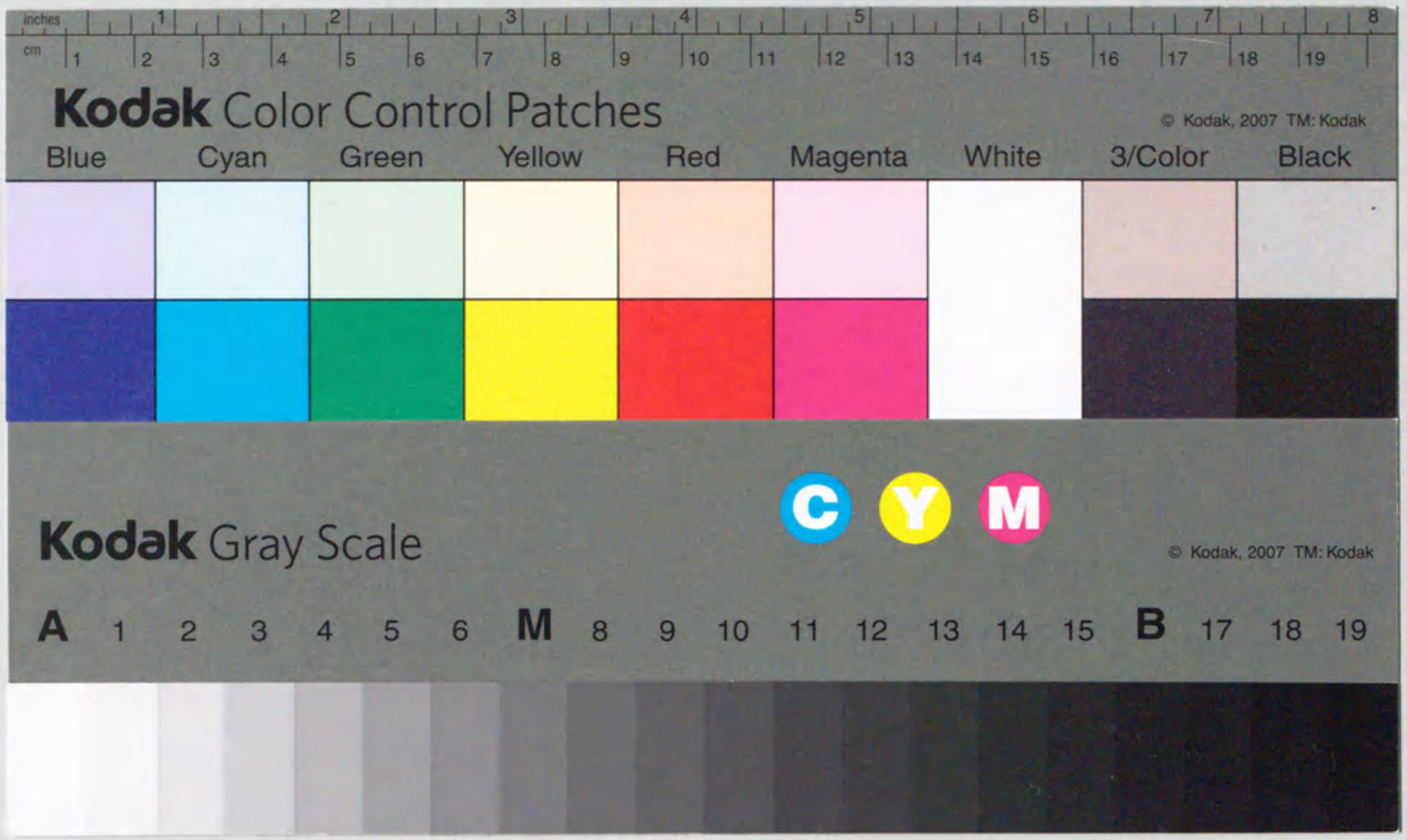
また，この研究の共同研究者として，常に鋭い洞察力と，緻密な論理展開で私を啓

発し続けて下さった機械技術研究所物理情報部計測制御研究室の黒河治久主任研究官と同部知識工学研究室の富田康治主任研究官にお礼を申し上げます．本研究の多くの部分は，お二人の協力がなければとうてい成し得なかったものです．

最後に，私の家族に感謝の気持ちを表します．研究の仕事というものは，ある種世間離れしたところがあるものですが，家族のみなが，私を終始暖かく見守り，励ましてくれました．今は亡き祖父村田新一は，高校の理科教師でしたが，私に生き物の素晴らしさ，学問のおもしろさを教えてくれました．私の進む道は，この少年の頃に決まっていたのかも知れません．

１９９７年6月，つくばにて

村　田　智


inches
cm
Kodak Color Control Patches
© Kodak, 2007 TM: Kodak
Blue
Cyan
Green
Yellow
Red
Magenta
White
3/Color
Black
Kodak Gray Scale
C
Y
M
© Kodak, 2007 TM: Kodak
A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19