セルフアセンブリシステムにおける

"形の計算"の実験的考察

慶應義塾大学 環境情報学部 升森 敦士

要旨

通信や計算はデジタル化によって、飛躍的な発展を遂げているが、生産手法においても、設 計情報をコード可能なデジタルマテリアルを用いたデジタル化を実現することで、飛躍的な発 展を遂げる可能性がある.デジタルマテリアルを用いた生産手法では、生物をはじめとした自 然におけるもののつくられ方と同様に、自己組織的に形を構成(セルフアセンブリ)させること が重要になってくる.その際、デジタルマテリアルや外部環境の、形やパターンの相互作用が 重要な要素の一つになってくると考えられる.そこで、本研究では、セルフアセンブリと形、 パターンに関する研究を行う.まず、人工のセルフアセンブリシステム、及び、物理シミュレ ーションや実空間での実験環境の設計・実装を行い、それらを用いて、主に、"形の計算"と も言える、形やパターンの相互作用とセルフアセンブリシステムのふるまいとの関係を、実験 を通して考察する.その結果、形やパターンと、セルフアセンブリシステムの効率性、要素モ ジュールの相互作用の頻度や、アセンブリされる順序などのセルフアセンブリシステムのふる まいとの関係性が見いだされた.

1. はじめに	1
1.1. 研究背景	1
1.2. 関連研究	2
1.3. 目的・意義	2
2. 設計・開発	4
2.1. 概要	4
2.2. セルフアセンブリシステム	4
2.3. モジュール	5
2.4. パターン最適化ツール	7
2.4.1. 遺伝的アルゴリズム	8
2.4.2. 遺伝型・表現型	8
2.4.3. 評価関数	9
2.4.4. 選択・変位・交叉	9
2.4.5. 実装	10
2.5. 物理シミュレーション環境	12
2.5.1 設計	12
2.5.2. 実装	13
2.6. 実空間での実験装置	14
2.6.1 設計	14
2.5.2. 実装	15
3. 実験 1: 凹凸パターン最適化	17
3.1. 目的	17
3.2. 方法	17
3.3. 結果	22
3.4. 考察	22
4. 実験 2:形,パターンの相互作用 (物理シミュレーション)	23
4.1. 概要	23
4.2 実験 2-1	25
4.2.1. 目的	25
4.2.2 方法	25
4.2.3. 結果	25
4.2.4. 考察	29

	4.3. 実験 2-2	29
	4.3.1. 目的	29
	4.3.2. 方法	30
	4.3.3 結果	32
	4.3.4. 考察	46
	4.4. 実験 2-3	47
	4.4.1. 目的	47
	4.4.2. 方法	47
	4.4.3 結果	49
	4.4.4. 考察	50
5.	結論・展望	51
6.	謝辞	53
7.	参考文献	54

1. はじめに

1.1. 研究背景

自然は、原子、分子の結合からはじまり、鉱物、雪の結晶、渦の生成など多くの形やパターン をつくる.このような自然におけるもののつくられ方は、主に要素間の相互作用によって秩序が 形成される自己組織化現象によって形成される.生物は、このような自己組織化現象の最も高度 な例と言えるだろう.また、生物自体も、タンパク質の合成というかたちでものの生産を行って いると言える.生物はDNAというかたちで設計図を持っており、その設計図をもとに、リボソ ームなどの高分子のはたらきによって、様々な機能をもつタンパク質を合成する.生物は、この ように設計図をもとにタンパク質の合成を行うが、リボソームが設計図を元に全て組み立てるの ではなく、リボソームによって行われるのはアミノ酸の結合といった比較的単純な作業である. 結合されたアミノ酸は、その化学的性質によって、自ら折りたたまれていき、立体構造をとる. このように、生物をはじめとして、自然におけるものの作られ方は要素間の相互作用による自己 組織化現象を通して行われる.

一方,従来の人工物の生産手法はこうした自然におけるもののつくられ方とは大きく異なる. 例えば,最先端の工場では,細かい部品から複雑なものを作り出しているが,それを行っている のは,命令に忠実に従う精密な組み立て機械である.これは,いわば人の手作業によるものづく りからのアナロジーであると言え,自然におけるものの作られ方とは異なる.例えば,DNAの 複製のエラー修正機構は非常に優れたものであるが,こうした組み立て機械が命令に従って行う トップダウン的な生産手法では,エラーの修正は困難となる.

Gershenfeldは、アナログ通信からデジタル通信、アナログ計算からデジタル計算の流れに、 アナログのものづくりから未来のデジタルファブリケーションへの流れを重ねて論じている[1]. Gershenfeldの言うように、アナログ通信やアナログ計算は、デジタル化することで、アナログ 方式におけるエラーの蓄積を克服したのと同様に、工作機械を用いてアナログな加工を行うので はなく、ファブリケーションをデジタル化し、プログラム可能なデジタルマテリアルを用いるこ とで、ものづくりにおいてもデジタル革命が起こる可能性があるだろう.このようなデジタルマ テリアルを用いたものづくりでは、デジタルマテリアルの集合自体が設計図となっており、例え ば、デジタルマテリアルによって構成されたものの一部が欠損していることが分かっているが、 どこが欠損しているか分からない場合でも、材料としてのデジタルマテリアルを容器に入れて振 るだけで、修復が行われるような自己修復的なエラーの修正を行うことも可能になるかもしれな い.

また,従来の人工物の生産手法においては,より複雑で精密なものを作るために,より精密な 組み立て機械が必要となるため,いずれ頭打ちの状況になると考えられる.このような状況を克 服するためには,生物を初めとした自然におけるものの作られ方が参考になるだろう.実際に, 人間は約60兆個の細胞によって構成されており,その細胞一つ一つが複雑なシステムとなってい るが,従来の人工物の生産手法によって,これほど複雑なものをつくることは不可能であるだろ

1

う.よって,設計情報をコード可能なデジタルマテリアルを用いた未来のデジタルファブリケー ションの研究を進めていく必要がある.こうした研究は,生物や自然の理解にも繋がるだろう.

1.2. 関連研究

自己組織的な生産手法の研究は、主にセルフアセンブリ(自己集合)システムの研究として行われてきている。特に、分子レベルでのセルフアセンブリの研究が盛んであり、DNAを用いて計算を行うDNAコンピュータ[2]や、DNA分子を利用した形状構成の手法などが研究されている。[3][4]主に、DNA分子を用いたセルフアセンブリシステムでは、Wangの抽象的なタイルモデル[5]を元にして発展している。分子レベルよりはスケール大きなナノスケールレベルでのセルフアセンブリシステムの研究もなされている[6].

一方,それらと比較して,よりマクロなスケールにおいても,セルフアセンブリシステムとし て捉えることのできる研究がいくつもなされている.例えば,早くも1958年にはPenroseらが, マクロスケールでのメカニカルな自己複製可能なモデルを提案している[7][8]. 他にも,マクロ スケールの物質を要素モジュールとして用い,モジュールの形状や,結合に用いる磁石などの素 材,外部環境をデザインすることで自己複製を実現させている例がいくつもある[9][10]. 例え ば,Virgoらは,要素モジュールの形を適切にデザインすることで,自己複製における分裂の段 階で,外部から特別な力を加えることなく,そのモジュール同士の形の相互作用によって促され るモジュールのふるまいによって自然に分裂が生じるモデルを実現している[11].

また,自己複製に限らず,マクロスケールの要素モジュールを用いて任意の目的形状を構成する手法[12][13]や,自己組織的にパターンをつくる研究も行われている[14].

こうしたマクロスケールでのセルフアセンブリ研究では要素モジュール同士の結合に磁石が用いられる場合が多いが、一方で、磁石を用いることなく、対流を利用し、要素モジュールの凹凸を噛み合わせるといった手法の研究も行われている[15][16].

近年では、小型のロボットモジュールを用いたセルフアセンブリシステムの研究も多数行われ てきている[17].

また, Miyashitaらは, 要素モジュールの形と, セルフアセンブリに関する研究を行っている. その研究によると, バイブレータを搭載した円や四角や三角の形をした要素モジュールを水に浮 かべそのふるまいを観察した結果, モジュールの形状などにより, 自己組織的に生じてくるパタ ーンに違いが出てくることが分かっている[18][19].

1.3. 目的·意義

セルフアセンブリシステムのように新しい生産手法においては、これまでの人工物の生産手法 とはシステムの制御の仕方が大きく異なる.こうした新しい生産手法では、デジタルマテリアル にいかに設計情報をコードするか、いかにしてデジタルマテリアルの自己組織的な秩序化を促す かといったことが重要になってくる.その際に,モジュールの形と外部環境からのエネルギーの パターンなどの相互作用の理解が重要になってくるだろう.

実際,自然におけるモノのつくられ方においては,形が重要な役割を果たす.例えば,立体構 造におりたたまれたタンパク質はその形に生じる化学的性質によって生体内で様々な機能をはた す.また,生物の免疫に関しても抗原と抗体の形が重要な役割をもつことが分かっている.自然 は,原子,分子の結合から惑星にいたるまで様々なものを作り出しているが,化学反応を含め, 自然のもののつくられ方は形の相互作用によっているとも言えるだろう.また,このような形の 相互作用をある種の計算として捉えることもできるだろう.未来のデジタルファブリケーション の実現のためには,このような形の計算ともいえる形の相互作用について理解を深める必要があ ると考えられる.

また,先述した,Miyashitaらの研究のような例もあるが,特に目的形状が明確なセルフアセ ンブリシステムにおいては,形の計算に重きをおいた研究はほとんどなされていない.そこで, 本研究では,まず,目的形状が明確な,セルフアセンブリシステムの設計・実装を行い,そのセ ルフアセンブリシステムにおける形やパターンとシステムのふるまいの関係について実験を通し て考察する.また,こうしたセルフアセンブリと形の計算の研究があまりなされていないため, 手探りで実験を行いながら,こうした研究のフレームワークについて考えることも大きな目的の 一つである.

また,形の計算に関する研究は,未来のデジタルファブリケーションの実現につながるととも に,生物や自然の理解を深めることにも繋がっていくと考えられる.

3

2. 設計·開発

2.1. 概要

本研究のように比較的マクロなスケールにおける人工のセルフアセンブリシステムは大まかに は図1のようなものとして捉えられる.このようなセルフアセンブリシステムを設計する場合, 主に,システム全体の設計,モジュールの設計,アセンブリ環境の設計が必要となる.本研究で は,これらに加えて,パターン最適化ツール,物理シミュレーション環境,実空間での実験装置 の設計・実装を行う.

本章の以下の節で各項目の設計について見ていく.



図1. セルフアセンブリシステム概念図

2.2. セルフアセンブリシステム

人工のセルフアセンブリシステムは、概ね以下の4つの指標によって分類することができる. 一つ目は、スケールである.主に、分子レベルのものか、比較的マクロスケールなものかで分け られる.二つ目は、エネルギーである.セルフアセンブリシステムでは、要素モジュールの個々 の相互作用によるため、個々モジュールに何らかのエネルギーを与える必要がある.大きく分け ると、要素モジュールがなにかしらのアクチュエーターをもっている内部駆動型(能動型)か、 要素モジュールの存在する外部環境から振動や風圧という形でエネルギーをあたえる外部駆動型 (受動型)に分けられる.また、内部駆動型、外部駆動型の中でもそれぞれどのような方式をと るか、エネルギーの段階数で分類できる.三つ目は、目的形状の有無である.明確な目的形状が あってその形状を構成するものと、自己組織的に形状やパターンを形成するものに分けられる. 4つ目は、次元である.これは、主に、二次元的な形状を形成するか、三次元的な形状を構成す るかで分けることができる. 本研究では、(1)マクロスケール、(2)外部駆動型(振動)、(3)目的形状あり、(4)二次元的 形状の特徴を持つ人工のセルフアセンブリシステムの設計及び実装を行う.

また,このような分類に基づくシステムを用いて目的形状をセルフアセンブリするまでの手順 としては、目的形状を構成するためのモジュールセットの生成、モジュールにコードするパター ンの生成、パターンをコードしたモジュールの制作、アセンブリ環境でアセンブリを実行、とい う流れとなる.

そのため、本来は、本研究で設計実装するパターン生成ツールに加えて、目的形状を構成する ためのモジュールセットを生成するツールが必要となる.こうしたツールを用いて、より少ない 構成のセットを生成することで、効率のよいセルフアセンブリが可能となる.しかし、本研究で 用いるような2次元的な形状を構成するための最小のモジュールセットの生成手法は、鎖状、ま たは木構造の形状に関しては提案されているが、任意の形状における最小のモジュールセットの 生成に関しては、NP困難問題であることが分かっている.[20]また、本研究では、セルフアセ ンブリシステムのふるまいを観察することが主な目的であることから、現段階ではモジュールセ ットの生成ツールに関しては、設計を行わなかった.また、目的形状を構成するための最小のモ ジュールセットの厳密な最適解ではなくより優れた解を求める手法としてヒューリスティックな 最適化手法を用いた手法も提案されている[13].

本研究では、上記のような構成となるセルフアセンブリシステムを用いて、セルフアセンブリ システムのふるまいと、形やパターンとの関係を、実験を通して考察する.そのため、上記の基 本的なシステムに加えて、実験用の物理シミュレーション環境や、アセンブリ環境の設計・実装 も合わせて行う.

以下の節において, 各項目の設計について見ていく

2.3. モジュール

本節では、アセンブリシステムの内、モジュールの設計について述べる.

本研究で用いるアセンブリシステムは、DNAのようにモジュールの種類が少ないDNA型では なく、モジュールのパターンの組み合わせが多く必要なタンパク質型に相当する.こうした方式 は、DNA型と比較し、目的形状によって必要なモジュールの種類が大きく変動するためデメリ ットが大きいと考えられるが、実用に際しては、こうした多種類の高分子型のモジュールを、よ り小さなDNA型のデジタルマテリアルにおいて構成することも考えられる.つまり、本研究に おける、デジタルマテリアルは、より小さなDNA型によって構成された集合体を抽象化した中 間子として捉えることもできるのである.

本研究では、図2にあるように、立方体を基本形状とし、その立方体の上下を除いた四つの側面に、磁石と凹凸からなるパターンを配置する設計とする. (本研究では、磁石には、ネオジム磁石(φ2×1mm)を用いる) 3×3のグリッドの中央のセルに磁石を配置し、残りの8つのセルに

凹, 凸, 凹凸なしのいずれかを配置する設計としている. これらの凹凸パターン一面の可能な組 み合わせは, 2·3⁸ (13, 122)通りである.

Bhallaらのモデルのように、凹凸ではなく、複数の磁石の配置によって、パターンを作る方法 もあるが、この手法では、本来結合すべきではない、マテリアル同士が比較的弱い結合をする準 安定状態に陥る場合が多くなることから、本研究では、凹凸パターンを採用する.

また, DNA型によって構成される中間子として捉えるのではなく, 側面の8つのセルが全て凹 になっており, そこに凸, または平のサブモジュールを配置することで, 凹凸パターンをプログ ラム可能なデジタルマテリアルとして利用することも可能である.



図2. モジュールデザイン 一辺が10mmの立方体を基本形状とし、その側面に凹凸パターン と磁石を一つずつ配した設計. 左は3D-CADによるモデリング. 右は、3Dプリンタで出力し た実際のモジュール.

2.4. パターン最適化ツール

目的の形状を構成するのに必要なモジュールセットが求められた場合,それらのモジュールセットを,実際にモジュールにコードするための凹凸のパターンとして生成する必要がある.その 際,アセンブリエラーが生じないように凹凸パターンの組み合わせ最適化を行う必要がある.この凹凸パターン一面の組み合わせは上記のように13,122通りであるため,n種類の凹凸パターンの探索空間は13,122のn乗となる.これは,例えば,nが3の場合で282,429,536,481通りとなり,組み合わせが爆発的に増えていく.そのため,こうした問題を近似的に解くことに適したヒューリスティックな最適化の手法の中から遺伝的アルゴリズムを用いて最適化を行う.

また,前節で設計した要素モジュールで予備実験を行ったところ,目的形状のアセンブリは達成されたが,正しいモジュールの組み合わせによる結合よりも拘束力の弱い,準安定状態と呼べる状態に陥るケースが多く見られた.(図3)

セルフアセンブリさせる際に容器に与える振動の強さを調整することで,安定状態の結合を分 裂させずに,このような準安定状態の結合のみを分裂させることも可能であると考えられるが, 準安定状態に陥ってから準安定状態が解消されるまでに多少の時間を要するため,セルフアセン ブリが完了するまでに余分に時間がかかってしまう.そこで,通常のアセンブリエラーを回避す るためのパターン最適化と同時に,可能な限り,準安定状態に陥る組み合わせが少なくなるよう なパターンの最適化が行えるように設計する.



図3. 準安定状態に陥ったモジュールの様子 相補的でない 側面の凹凸パターン同士の場合でも凹凸のうちのいくつかが 噛み合うことで,画像にあるよう比較的弱い結合をする場合 が観察された.この比較的弱い結合をここでは進安定状態と

2.4.1. 遺伝的アルゴリズム

遺伝的アルゴリズムは、生物の進化を参考にして Holland らによって考案された解探索手法 の一つである[21]. Hollandらの研究に続いて、Goldbergらの研究[22]など、多くの研究によっ て発展し、 その有効性が示されたことから、現在では工学的な応用を含め幅広い分野で用いら れている.

遺伝的アルゴリズムを構成するのは、主に、遺伝型、表現型と、それらの操作を担う、選択、 交叉、突然変異である.遺伝的アルゴリズムでは、対象となる遺伝群の表現型を評価し、その適 応度に基づき遺伝群から規定数を選択し、その選択された親個体を交叉し、突然変異を与えるこ とで子個体を生成する.そして、その子個体群を評価し選択するという流れで繰り返し処理され る.

以下の節では、本研究で用いた遺伝型、表現型、評価関数、選択、交叉・変異について見ていき、最後に、本研究で実装した、インタラクティブに突然変異率などのパラメータの変更が可能なIPGAについて見ていく.

2.4.2. 遺伝型·表現型

遺伝型は、01200012のように3種類の記号からなる8桁の文字列を一つの遺伝子とし、ユーザ ーが必要とする凹凸パターンの数に応じて、遺伝子の数が決まる.例えば、2種類の凹凸パター ンが必要な場合は、8桁の文字列からなる遺伝子が2個、計16文字の文字列となる.例えば、図4 にある2種類の凹凸パターンは102201012、000001122という文字列で表現される.

表現型は,遺伝子となる8桁の文字列を左上から順に3×3のマトリックスに配置していき,0 を凹凸なし,1を凸,2を凹とする.これを一つの凹凸パターンとし,それぞれの遺伝子に対して 凹凸パターンが表現される.





図4. 凹凸パターンの遺伝型 ●が凹, ○が凸を表している. これ ら2種類の凹凸パターンは,本研究で用いる遺伝型では 102201012,000001122という文字列として表現される. 0が凹凸 なし,1が凸,2が凹に対応しており,文字列は左から順に,グリ ッドの左上から右下の順に対応している

2.4.3. 評価関数

遺伝的アルゴリズムでは,遺伝型を元に生成された表現型を何らかの方法で評価し,その表現型の適応値が決定される必要がある.本研究では,式(1)のような形式からなる評価関数をあらかじめ用意しておき,この評価関数に従って,表現型の評価が決定されるように設計されている.

$$f(x) = \sum_{k=0}^{n} w_{i} f_{i}(x)$$
(1)

ここで、 $f_i(x)$ は、各指標の評価値を算出する関数であり、wにあたる部分がその各指標の値に掛け合わせる重みとなっており、評価関数は各指標に、重みを掛け合わせたものの線形結合となる.今回は、f(x)の値がより小さいものほど選択率が高まるように設計されている.

以下で、今回用いた各評価指標について述べる.今回は、単純なアセンブリエラーだけでなく、 準安定状態も減らすことが必要となる.準安定状態に陥る主な原因は、凹凸パターンが完全に相 補的ではないが、一部のセルが相補的であることであると考えられる.そこで、評価関数は、な るべく部分的にも相補的にならないパターンの組み合わせを高く評価するように設計する必要が ある.以下で、具体的な評価指標を示す.

大きく分けて二段階で評価している. (1) 0-1(平-凸), もしくは1-1(凸-凸)の関係になるセルが 多い組み合わせを高く評価. 1-1 (凸-凸)の関係が多いものをより高く評価. (2) 0-1, もしくは 1-1の関係にあるセルが3組以上ある場合, それらのセルからなるConvex-hullを求めその面積が より大きなものを高く評価する. (便宜上は加点方式として説明しているが, 実際には減点方式 として実装している.)

(1)は単純に、0-1、もしくは1-1の関係にあるセルが多い方が、準安定状態に陥る可能性が低く なることが期待できるために設定している.(2)は、0-1(平-凸)、もしくは1-1(凸-凸)の関係が3つ 以上ある場合でも、準安定状態に陥る可能性があるため、Convex-Hullの面積を比較することで、 より準安定状態に陥る可能性が少ないパターンの組み合わせに最適化されることを狙っている.

2.4.4. 選択・変位・交叉

まず,選択について述べる.遺伝的アルゴリズムで,一般的に用いられる選択のアルゴリズム として,ルーレット方式,ランク方式,トーナメント方式などがあるが,本研究では,トーナメ ント方式を採用する.トーナメント方式では,まず個体群の中から,指定した数の個体をランダ ムに選び出し,その選び出した個体同士の適応度を比較し,その中で最も高い適応度を示すもの を選択する.そして,これを指定した回数繰り返し次世代に残す親を選択する. 遺伝的アルゴリズムで用いられる交叉のアルゴリズムとして、一点交叉、二点交叉、一様交 又などが挙げられる.一点、二点、一様の順により、効率よく解を探索できることが主張されて いる[23][24]. そのため、本研究では一様交叉を採用した.一様交叉は、すべての遺伝子について、 2つの親個体のどちらの個体から引き継がれるかがランダムに決定されるというものである.

突然変異に関しては、点突然変位を採用した.具体的には、事象ごとに予め設定された確率で 各セルの値がランダムに変更される.

2.4.5. 実装

本研究では、遺伝的アルゴリズムの実行中に、突然変異率、交叉率、評価に関するパラメータ をリアルタイムに変更できるようソフトウェア(IPGA)の実装を行った(図5).これにより、 人為的な操作で、局所解から抜け出すことを促すことができ、最初の段階では変異率・交叉率を 高めにし、徐々に変異率・交叉率を下げて行くことで効果的に解探索を行える等のメリットがあ る.所謂インタラクティブなGAは、評価をあらかじめ用意しておいた評価関数のみで評価する のではなく、ユーザーの好みによって選択していくといったものである.一方、本研究で開発し たIPGAでは、変異率・交叉率等のパラメータをユーザーがインタラクティブに操作するといっ たものである.なお、本研究では実装を行わなかったが、ファジイ推論を用いた動的パラメトリ ックGA等、より効率の良い手法の提案もされている[25].



図5. IPGA実行中のスクリーンショット 最適化実行中に,左上のスライダーで,突然変異率, 交叉率などをリアルタイムで変更することができる. 実際に、凹凸パターンを生成し、その凹凸パターンに基づいてモジュールを制作し、磁石は埋め込み試行実験を行った.その結果、図6にあるように、目的の形状にセルフアセンブリすることが確認できた.¹



図6. セルフアセンブリ実行結果

¹ 以下の動画で,セルフアセンブリされる様子を確認できる. http://vimeo.com/56069584

2.5. 物理シミュレーション環境

2.5.1 設計

セルフアセンブリシステムの研究において、物理シミュレーションは、生成したモジュールセットで目的形状がアセンブリ可能かどうかの確認や、マテリアルの形状やモジュールセットの遺伝的アルゴリズムなどの進化的アルゴリズムによる最適化を行う際の評価としても使用できる等、様々なメリットがあると考えられる.

本研究では、特に、モジュール、領域サイズ、領域形状、領域の運動パターンを変えた際のセ ルフアセンブリシステムのふるまいを観察することを目的としている.しかし、実空間での実装 を目的としたセルフアセンブリシステムの研究においては、上で述べたような、準安定状態のよ うな想定外のエラーなどが起こることが考えられるため、セルフアセンブリシステムのふるまい の観察においても物理シミュレーションで閉じるのではなく、実空間での実験が重要になり、物 理シミュレーションはあくまで予備的な扱いにすべきであるだろう.



図7. 物理シミュレーション実行中のスナップショット1 正方形型の領域に9個の立方 体型のモジュールを入れて振動を与えている様子.一部のモジュールが結合している様 子が確認できる.本物理シミュレーション環境では,このように,演算結果をリアルタ イムで描画し,アセンブリの経過を確認することが可能となっている.

本物理シミュレーション環境は、入力として、モジュールパターンのセット、各モジュールの 個数、結合ルール、モジュールサイズ、モジュール形状、領域サイズ、領域形状、領域の運動パ ターンなどを与えて実行することで、セルフアセンブリのシミュレーションが可能となっている. 図7にあるように演算結果をリアルタイムで描画しながらセルフアセンブリの経過を観察するこ とも可能となっている.

実際に、必要項目を入力し、シミュレーションを実行したところ、図8にあるように目的の形 状がアセンブリされることが確認できた.



図8. 物理シミュレーション実行中のスナップショット2 シミュレーション実行の結果,アルファベットの"S"を模 した目的形状がセルフアセンブリされた様子.

2.5.2. 実装

実装にあたって、オープンソースの物理演算エンジンであるBullet Physics²を用いた. Bullet Physicsは、リアルタイムで描画可能な物理演算エンジンの中では、衝突判定が比較的優秀であるとされており[26]、今回のセルフアセンブリのシミュレーションに適していると考えられる. 本来リアルタイムの物理シミュレーションでは、処理速度を優先し、精度を犠牲にしている場合があり、正確性を重視する目的には不向きである.しかし、本研究では、セルフアセンブリシス

² Erwin Coumans らが開発するオープンソースの物理演算エンジン. http://bulletphysics.org/

テムのふるまいを観察しながら、インタラクティブに領域形状や運動パターンなどの条件を変更 していくことも想定し、リアルタイムの物理演算エンジンであるBullet Physicsを採用した.精 度を重視した物理シミュレーション環境を構築することは今後の課題である.

以下では、要素モジュール、結合、領域形状、領域の運動パターンの実装に関して述べる.

本物理シミュレーション環境は、領域形状や領域の運動パターンと、目的形状の出力時間など セルフアセンブリシステムのふるまいとの関わりを見ることが、主な目的の一つである.そのた め、処理効率を優先し、モジュールの基本形状は立方体としており、凹凸パターンは表現してい ない.立方体以外にも、基本形状として、円筒型、三角柱型を選択できるよう実装している.ま た、ユーザーが3Dモデリングソフトで制作した、任意の形状の3Dモデルのobjファイルを読み込 みモジュールとして使用することも可能である.

磁力のシミュレーションは処理コストが大きくなるため、本シミュレータでは、近似的に、一 定距離内にあるモジュールが結合ルールに適合するか調べ、結合ルールに適合する場合は、その 結合ルールに基づきモジュール同士が結合されるように設計されている.また、実空間での結合 と異なり、一度結合されたモジュール同士は、強い力が加わっても分解されない.

領域形状は、3Dモデリングソフトで制作し、objファイルとしてインポートする仕様になって いる. その際、領域形状は剛体か柔軟体のいずれかで表現するかを選択することが可能である.

容器の運動パターンは,現段階では,予め用意したパターン(例:平行運動,旋回運動)の組 み合わせによってデザインする仕様になっている.

2.6. 実空間での実験装置

2.6.1 設計

実空間でのセルフアセンブリシステムの実験環装置として用いることを目的として,任意の運動パターンで容器を振ることを可能にするための振盪器(ShakerBot)と,ShakerBot専用ソフトウェアの設計・開発を行った.本節では,そのShakerBotと専用ソフトウェアに関して述べる.

ShakerBotは、3軸のロボットアームを、専用ソフトウェア上で、ユーザーがデザインした運動パターンに基づき制御する構成になっている.運動パターンに関しては、あらかじめ決められたパターンを組み合わせることに加え、ベジェ曲線や直線の描画ツールを使用して任意のパターンをデザインすることが可能となっている.

2.5.2. 実装

ShakerBot のハードウェアの基本構造は DFRobot社製のロボットアームを使用している. そのロボットアームを構成する3つのサーボモータをArduinoと専用ソフトウェアによって制御している.

ロボットアームは使用するにあたって、位置情報などのキャリブレーションが必要になるが、 本研究で実装した専用のソフトウェアでは、画面上のスライダーを使用してリアルタイムに調整 できるようになっている.また、専用ソフトウェアでは、デザインした運動パターンをロボット アームが再現した際のアームの動きを画面上で確認できる仕様になっている.

図9のように、ソフトウェア上で、運動パターンのデザインを行い、図10のようにシャーレを せっとして、予備的に実験を行ったところ、目的形状を形成することが確認できた.



図9. shakerBot専用ソフトウェア実行中のスクリーンショット ユーザー は、このソフトウェアを用いて運動パターンのデザインを行い、実行ボタ ンを押すことで、ロボットアームを命令を送ることができる.現段階で は、直線ツールと、ベジェ曲線ツールが実装されており、それらをもちい て任意の運動パターンをデザインすることができる.画像は運動パターン をベジェ曲線で描いている様子.



図10. ShakerBotでシャーレを振る様子

3. 実験 1:凹凸パターン最適化

3.1. 目的

理論的には、アセンブリエラーが生じないモジュールのセットが得られた場合でも、実空間で アセンブリを実行させる際に、想定外のエラーが生じる場合がある。例えば、本研究において設 計した要素モジュールでは、結合すべきではない要素モジュール同士の凹凸の一部が噛み合って 比較的弱い結合をする準安定状態といえるような状態に陥る場合がある。

実験1では、こうした準安定状態が減少するように遺伝的アルゴリズムで最適化したモジュールと最適化していないモジュールのアセンブリ出力時間を比較する.

これにより、本研究で設計・実装した遺伝的アルゴリズムによるパターン最適化の手法及びソフトウェアの有効性を確認するとともに、準安定状態のようなエラーを減少させることで、セルフアセンブリシステムの出力時間に変化が見られるか検証する.

3.2. 方法

本節では,実験1の手法について述べる.セルフアセンブリシステムは,2章で述べた設計・実装したものを用いる.

実験の手順としては、まず、本研究で設計・実装した遺伝的アルゴリズムによる凹凸パターン 最適化のソフトウェアを用いて、要素モジュールの凹凸パターンに準安定状態が生じにくくなる ように最適化する.その最適化後の凹凸パターンを配したモジュールセットと、最適化前のモジ ュールセットとでそれぞれアセンブリ実験を実行し、目的形状がアセンブリされるまでの時間を 計測し比較する.

セルフアセンブリの目的形状として、アルファベットの"S"、"F"、"C"を模した形状と、モジ ュール3つからなる棒状の形状の4種類を用いる(図11). これらの目的形状を構成するモジュー ルセットの詳細を表1、2に示す.表1の左がモジュールの番号で、右がその番号のモジュールの 側面にある凹凸パターン名である.凹凸パターンは左から順にモジュールの側面に時計回りに対 応している.表2が、結合ルールとなっており、例えば、"A+B→A·B"はAとBの凹凸パター ンが互いに相補的な組になっており、接近した際に結合することを示している.最適化前と最適 化後の凹凸パターンを図12に示す.

本研究で用いるセルフアセンブリシステムにおける環境は、容器と、容器に振動を与える外部 装置からなる、容器として、ポリスチレン製のシャーレ(φ90×15mm)を用いる、容器へ外部か ら振動を与える装置として、AS ONE社製TRIO TM-2Fを用い、振動のレベルは、10段階中の8 に設定する(図13).

上記の構成の要素モジュールのセットを,目的形状ごとに,それぞれ10回ずつ試行し,目的形状のアセンブリ完了までの時間を計測する.この際,用意するモジュールのセットは, "S", "F", "C"の3種類に関しては,それぞれ,目的形状一つ分とするが,棒状の形状に関しては,目的形状4個分のモジュールセットを用意しておき,目的形状が4個中3個アセンブリされ

た時点での時間を計測する.これは、アセンブリすべき結合が残り一つの場合は、準安定状態に 陥ることはなく、最適化前と最適化後で最後のモジュールが結合するまでに要する時間には差が 生じないと考えられることから、最適化前と最適化後の効率性を正確に測れなくなることを避け るための条件である.



図11. "S", "F", "C"型の目的形状を構成する要素モジュール一覧 セル内の数字はモジュールの 番号.棒状の形状は2:1個,3:1個,4:1個の計3個のモジュールからなる. "S"型形状は1:2 個,2:2個,3:2個,4:2個,5:2個,6:1個の計11個のモジュールからなる. "F"型形状 は,1:1個,2:1個,3:3個,4:3個,5:1個の計9個のモジュールからなる. "C"型形状は, 1:1個,2:1個,3:2個,4:2個,8:1個,9:1個,10:1個の計9個のモジュールからなる.

module ID	concave-convex pattern distribution
1	A - E -
2	ВІ
3	J - C -
4	D
5	F G
6	H · H ·
7	F - I C
8	F - K -
9	L - A -
10	B I

表1. モジュールと凹凸パターンの対応表

表2. 結合ルールー覧
connection rule set
$A + G \rightarrow A - G$
$B + H \rightarrow B \cdot H$
$C + I \rightarrow C - I$
$D + J \rightarrow D - J$
$E + K \rightarrow E - K$
$F + L \rightarrow F - L$



S

J

Ν

L

Ν

Κ

N

図12. 凹凸パターン一覧 セル内の●が凹, 〇が凸, S, Nは磁石の磁極に対応している. A: 遺伝 的アルゴリズムによる最適化前の凹凸パターン.上段と下段がそれぞれ相補関係にある.B:遺伝 的アルゴリズムによる最適化後の凹凸パターン. 準安定状態が生じにくくなるように凹凸パターン が最適化されている.A同様に上段,下段が相補関係になっている.

I

Ν

G

Ν

н

Ν

20



図13. 実験環境概観 外部から振動を与える装置としてAS ONE 社製 TRIO TM-2F を使用している. TM-2Fにシャーレを固定す るためのアタッチメントは、本研究用に、3次元CADソフトでモ デリングし、3Dプリンタで出力したものを使用. 容器はポリスチ レン製シャーレ(Φ90×15).

3.3. 結果

実験1の結果について見ていく.4種類の目的形状ごとに,計測した出力時間の最適化前,最適 化後それぞれの平均値の比較を図14に示す.この図から,全ての目的形状において最適化前と比 較し,最適化後の方が目的形状の出力時間が大幅に短縮されたことが確認できる.この結果から, 遺伝的アルゴリズムによる凹凸パターンの最適化によって,準安定状態に陥る頻度が減少し,ア センブリの効率性が増したことが考えられる.



図14. 凹凸パターン最適化前と最適化後のモジュールセットによるセルフアセンブリ完了まで の経過時間 遺伝的アルゴリズムを用いて最適化した凹凸パターンと,最適化前の凹凸パター ンで,それぞれの目的形状のアセンブリが完了までの経過時間をグラフ化したもの. グラフ は,それぞれ,左から棒状形状,S,F,Cの文字を模した形状を目的形状とした場合のグラ フ.各目的形状ともに,最適化後の方が,目的形状のアセンブリ完了までの経過時間が短いこ とが分かる.

3.4. 考察

実験1によって,モジュールの凹凸パターンを最適化することで,セルフアセンブリの効率性 を高めることができることが確認された.

また,最適化前の凹凸パターンからなるモジュールセットにおいては,準安定状態が連なって 閉じた形状に陥る場合があった.この準安定状態が連なった形状に陥った場合,モジュール二つ からなる準安定状態よりもより高い結合状態となり,分解されるまでにより時間を要した.この 準安定状態の複合体に陥る頻度については計測を行っていないが,"C"型の目的形状の最適化前 の出力時間が最も遅かったのは,このような状態に陥る頻度が高かったことによると考えられ る. このように、あらかじめ想定していた準安定状態とも異なる準安定複合体に陥る様子が何度も 確認された.本研究で用いたような、目的形状があらかじめ決まっているセルフアセンブリシス テムにおいては、こうした準安定状態のようなエラーの頻度を減らしていくことが課題となるだ ろう.

また、今回は、モジュールの凹凸パターンの最適化のみを試したが、今後、評価関数を検討す ることで、モジュールの形や、凹凸の形を遺伝的アルゴリズムなどの最適化手法で最適化し、セ ルフアセンブリシステムの効率性を高めることも可能になると考えられる.

4. 実験2:形,パターンの相互作用(物理シミュレーション)

4.1. 概要

セルフアセンブリによって得られる結果は、モジュール自体のデザインだけでなく、アセンブ リを行う領域(ここでは、シャーレ様の容器)の形、又、その外部から与えられるエネルギーの パターン(ここでは容器の運動パターン)によっても変化すると考えられる.そこで、本章では、 そうしたセルフアセンブリシステムと形、パターンの関係を、実験を通して考察する.

本章における実験では、本研究のために開発したセルフアセンブリシステム用の物理シミュレ ーションを用いる.物理シミュレーション環境上で、複数の領域形状と領域の運動パターンの組 み合わせを適用させ、セルフアセンブリシステムのシミュレーションを実行し、その結果から、 領域形状や領域サイズ、運動パターンと、出力時間を含めたアセンブリシステムの挙動との関係 を考察する.

本章では、大きく分けて3つの実験を行う.

一つ目の実験では、セルフアセンブリシステムの出力時間と、形、パターンの関係を探る目的 で、複数の領域の形状、運動パターンの組み合わせを適用させ、それぞれの組み合わせにおける 目的形状が出力されるまでの時間を計測する.

二つ目の実験では、容器内のモジュールの挙動と、形とパターンとの関係を探る目的で、複数 のモジュール形状、複数の領域形状、運動パターンの組み合わせを適用させ、それぞれの組み合 わせにおけるモジュールの相互作用係数、速度、角速度を測定する.

三つ目の実験では、アセンブリ順序と形、パターンの関係を探る目的で、複数の領域の形状、 運動パターンの組み合わせを適用させ、それぞれの組み合わせにおける、目的形状が出力される までのアセンブリの順序を記録する.

以下では、本章の実験で用いる基本的な条件となる、領域形状、運動パターンについて述べた 後、次節から、個々の実験について詳しく説明していく.

領域の形状として,円型,正方形型,非対称型を使用する(図15).それぞれ大,中,小の三 つのサイズを用意する.それぞれのサイズの面積は形状に依らず一定としている. 運動パターンとして,平行型,環状型,ランダム型を使用する(図16). それぞれ運動パタ ーンの速度は一定である.ここでのランダム型は,一定の範囲内で目的位置がランダムに決定さ れ,その目的位置に向けて一定の速度で移動し,目的位置に達した時点で次の目的位置がランダ ムに決定されるといったものである.



図15. 領域形状 左から円型,正方形型,非対称型. それぞれ大,中,小の3つ のサイズを用いる. それぞれのサイズにおいて,領域によらず面積は一定として いる.



図16. 領域の運動パターン 左から,平行型,環状型,ランダム型. それぞれ 速度は一定.

4.2 実験 2-1

4.2.1. 目的

本研究の実験で用いるセルフアセンブリシステムのように目的形状があらかじめ決められてセ ルフアセンブリシステムにおいては、出力結果としての形状はアセンブリエラーが生じない限り は一定である.一方、セルフアセンブリシステムは確率的なふるまいをするシステムであり、そ の出力に要する時間は一定ではない.

そこで、複数の領域サイズ、領域形状や領域の運動パターンの組み合わせでシミュレーション を実行し、目的形状の出力時間と、形、パターンとの間にどのような関係があるかを探索する.

4.2.2 方法

実験2·1では、先に述べた通り、複数の領域の形状、運動パターンの組み合わせを適用させ、 それぞれの組み合わせにおける目的形状が出力されるまでの時間を計測する.

領域の形状,運動パターンは,前節で述べたように,領域形状3種,領域サイズ3種,運動パタ ーン3種による計27種類の組み合わせを用いる.

目的形状は,実験1で用いたものと同様の,モジュール3個が平行に連なった棒状の形状とする. 目的形状を構成するためのモジュールセットに関しても,実験1で用いた棒状の形状と同様である(表1,2).

上記の,条件からなるセルフアセンブリシステムにおいて,目的形状4個分のモジュールセットを適用し,目的形状が3個アセンブリされた時点での時間を記録する.27組の組み合わせごとに,50回ずつ試行する.

4.2.3. 結果

本節では、実験2-1の結果について見ていく.まず、領域形状3種、領域サイズ3種、領域の運動パターン3種からなる27組の、目的形状3個分のアセンブリ完了までの経過時間の平均時間を図17に示す.最も出力時間が短かったのは小型の非対称型領域形状とランダム型の運動パターンの組み合わせで7.2秒、最も経過時間が長かったのは、中型の正方形型形状と平行運動パターンの組み合わせで22.5秒であり、組み合わせによっては出力時間に有意な差があることが確認できた.

さらに、環状型の運動パターンは、小型の領域形状では、出力時間が比較的長い傾向にあるが、 その他のサイズでは、出力時間は比較的短い傾向にあるといった点や、平行型の運動パターンは、 正方形型の領域形状と組み合わさると、出力時間が大幅に長くなるといった点からも、組み合わ せによる相性のようなものがあることが確認できる.

25



図17. 各領域サイズ・形状・運動パターンにおけるアセンブリ経過時間の比較 領 域の形状3種類(円型,正方形型,非対称型),及び,運動パターン3種類(平行型, 環状型,ランダム型)の組み合わせにおける目的形状の出力時間の平均値を指名し ている.上から領域サイズ:小,中,大.左から,領域形状:円,正方形,非対称 多角形.棒グラフの左から,領域の運動パターン:平行型,環状型,ランダム型. 最も出力時間が短かったのは小型の非対称型領域形状とランダム型の運動パターン の組み合わせで7.2秒,最も経過時間が長かったのは,中型の正方形型形状と平行運

次に、サイズ、領域形状、運動パターンごとにまとめて平均値を求めた結果を図18に示す. この図から、サイズでは小型の領域が、比較的、出力時間が長く、中型、大型に関しては、平 均値自体は中型の方が大型よりも出力時間は短かった. 誤差もあるため、さらなる検討が必要で あるが、領域のサイズ、つまり、領域内のモジュールの密度には適した値があることが予想され る. 今後、領域のサイズに関しては、さらに複数のサイズの領域による追加実験を行う必要があ るだろう.



図18. 各領域サイズ,領域形状,運動パターンの出力時間の平均値の比較

領域形状は,正方形型が,大幅に出力時間が長く,次いで,円型,非対称型となるが,円型, 非対称型に関しては,それほど大きな差が見られなかった.しかし,これらの形状を,正方形型, 円型,非対称型の順に,1~3の値をつけ,その値と,各領域が適用されたそれぞれ9組の出力時 間の平均値との相関係数は約0.43であり,ある程度の相関があることが分かった(表3,図19). その他の順に値付けをした場合も同様に試したが,正方形型,円型,非対称型の順に値付けをし た場合が最も相関係数が高い結果となった.

領域の運動パターンに関しては、図18からも平行型、環状型、ランダム型の順に出力時間が短 くなることが確認できる.領域形状に関しても同様に相関係数を求めたところ、約0.44であり、 ある程度の相関があることが分かった(表3,図19).同様に、その他の順に値付けをした場合 も試したが、平行型、環状型、ランダム型の順に値付けをした場合が最も相関係数が高い結果と なった.



図19. 各領域形状,運動パターンと出力時間の散布図 A: 各領域形状と出力時間の散布図. 縦軸は上から順に,正方形型,円型,非対称型.この順番に並べた場合に,出力時間との相関係数が最も高かった. B: 各運動パターンと出力時間の散布図. 縦軸は上から順に平行型,環状型, ランダム型.この順番で並べた場合の出力時間との相関係数が最も高かった.

表3. 領域形状,領域の運動パターンと出力時間との相関係数

attribute	correlation coefficient
container shape	0.44
kinetic pattern	0.43

4.2.4. 考察

平均値を見ると,領域形状では正方形型,運動パターンでは平行運動型と,領域形状,運動パ ターンともに,平行直線的なパターンが,最も出力時間が長く,次いで,円型,環状型のように 曲線的なパターン,非対称型,ランダム型のように,ランダムなパターンという結果となった. 直感的には領域形状,運動パターンの傾向には関連があるように思えるが,曲線的なパターンと ランダムなパターンの差は誤差も大きかった.

しかし、領域形状では、正方形型、円型、非対称型、運動パターンでは、平行型、環状型、ラ ンダム型の順に値つけをした場合の出力時間との相関係数が最も高く、それぞれ、約0.43、約 0.44であり、ある程度の関連がある可能性が考えられる。今後さらなる検討が必要であるが、例 えば、図形の複雑さや、曲率の変化などの、形の性質と、その形が生み出す領域内でのモジュー ルのふるまいの傾向に関連がある可能性が考えられる。今後、形自体の性質との比較を行ってい く必要があるだろう。

4.3. 実験 2-2

4.3.1. 目的

実験2・2では、複数の領域形状、運動パターンの組み合わせを適用させ、それぞれの組み合わ せにおける各モジュール同士の相互作用係数(詳細は後述する),速さ、角速度の大きさ(ここ では、対象とするモジュールのZ軸周りの角速度の大きさとする)、軌道パターンを記録する.そ の結果から、各組み合わせとモジュールの相互作用との関係を考察する.また、計測した諸要素 の傾向と、実験2・1で計測した出力時間とを比較することで、出力時間に関係する要素について 検討することを目的とする.

4.3.2. 方法

領域の形状,サイズ,運動パターンに関しては,実験2-1と同様に基本的な形状,パターンを 用い,各領域形状,サイズ,運動パターンにおける,各モジュール同士の相互作用係数,モジュ ールの速さ,角速度の大きさ(ここでは,対象とするモジュールのZ軸周りの角速度の大きさとす る)を記録する.

ここで、相互作用係数とは、試行ごとに、対象となるモジュールをランダムに一つ選び、その 対象とするモジュールが、その他の各モジュールの一定範囲内に存在していた時間をそれぞれ個 別に記録しておき、それらの経過時間の平均値を、それぞれの経過時間の分散で割ったものとす る(式2). これは、対象とするモジュールが、その他のモジュールの一定範囲内に存在してい た時間の平均時間がより長く、かつ、その他のモジュールの一定範囲内に存在していた時間のば らつきが少ないほど、モジュール間の相互作用の頻度は高くなると考えられることから設定した. 平均時間と分散はそれぞれ、20回のシミュレーションを実行した結果の平均値を、27組すべて の値で標準化したものを用いることとする.

$InteractionCoefficient = \frac{InteractionAverageTime}{InteractionVariance}$ (2)

実験2-2においては、主に、各組み合わせにおける、容器内でのモジュールの相互作用の傾向 を観察することが目的であるため、目的形状、結合ルールは設けない.そこで、実時間に換算し て10秒間のシミュレーションを行った時点で完了とする.その間の相互作用時間、速度、角速度 を1フレームごとに記録する.領域形状、サイズ、運動パターンの組み合わせごとに、20回ずつ 試行する.

また,モジュールの形状は実験2-1での立方体型に加え,図20にあるような円筒型,三角柱型 においても,同様に,相互作用係数,速さ,角速度の大きさの記録を行う.これら3種類のモジ ュールの質量,体積は同じ値に設定してある.モジュールの数はそれぞれ,実験2-1と同じ12個 とし,1試行ごとに領域内でランダムに配置する.



図20. 円筒型, 三角柱型のモジュール 左が円筒型, 右が三角柱型. それぞれ実験2-1で用いた 立方体型のモジュールと体積, 質量, 高さを同じ値に設定している.

4.3.3 結果

実験2・2の結果を見ていく.実験2・2では、複数のモジュール形状、領域形状、運動パターンの 組み合わせを適用させ、シミュレーションを実行し、それぞれの組み合わせにおける各モジュー ル同士の相互作用係数、速度、角速度、軌道パターンを記録した.これらの値について、実験 2・1の結果と比較しながら結果を見ていく.

まず,相互作用係数,速さの平均値,速さの分散,角速度の大きさの平均値,角速度の大きさの分散でそれぞれ,実験2-1の出力時間との相関係数を求めたところ,相互作用係数と,角速度の平均値で,比較的相関係数が高く,その他の項目は相関係数が低いことが分かった(表4).

表4. 相関係数の比較			
attribute	correlation coefficient		
variance of speed	0.213		
average of speed	0.049		
variance of angular speed	-0.218		
average of angular speed	-0.632		
interaction coefficient	-0.644		

そこで、相互作用係数と、角速度の大きさの平均値を二つの従属変数とし、実験2-1で記録した出力時間を独立変数として重回帰分析を行った.その結果、得られた線型方程式が以下の(式3)となる.(*IC*は相互作用係数の平均値,*AS*は角速度の大きさの平均値)

$3.2447 - 1.669 \cdot IC - 1.593 \cdot AS$ (3)

この線型方程式によって、予測される出力時間と、実験2-1において得られた実測値との散布 図を、図21に示す.これらの相関係数は約0.75であり強い正の相関関係にあることが分かった.



図21. 出力時間の予測値と実測値との散布図

(式3)よって求められる,予測出力時間の平均値を,領域サイズ,領域形状,運動パターンご とにまとめた場合の平均値を図22に示す.この図から,実験2-1の結果で示した,出力時間を領 域サイズ,領域形状,運動パターンごとにまとめた平均値と比較すると,平均値自体は,領域形 状では円型より非対称型の方が高く,運動パターンでは環状型がランダム型より高いなど,出力 時間の場合との多少の差異も見られたが,概ね似た傾向であることが分かる.



図22. 領域サイズ,領域形状,運動パターンごとにまとめた予測出力時間の平均値の比較

これらの結果と、先ほどの回帰分析によって求められた線型方程式から、出力時間には、相互 作用係数と、角速度の大きさの平均値が同程度に影響していることが予想される.これは、相互 作用係数のみでは、接近したモジュール同士のどの側面が向かい合っているかが測れず、相互作 用係数が高い場合でも、決まった側面同士しか接近しておらず、出力時間の効率化に影響しない 可能性が考えられるが、角速度の平均値が高くなることで、そうした偏りが減少すると考えられ ることから、この二つの変数の組み合わせによって、より正確な相互作用の度合いが測れると考 えられる.

そこで、ここでは仮に、(式4)のように相互作用係数と角速度の大きさの平均値からなる線型 方程式によって求められる値を、相互作用度数と呼ぶこととする.(ILは相互作用度数、ICは相 互作用係数、ASは角速度の大きさの平均値とする)この相互作用度数が高い組み合わせほど、セ ルフアセンブリを実行した際の出力時間は短くなることが予想される.また、(式4)における、 n, mは、後述するように、回帰分析によって求められる値を用いることにするが、適用させ るモジュール形状、領域形状、サイズ、運動パターンの種類によって異なってくる.

$IL = n \cdot IC - m \cdot AS \tag{4}$

実験2·1の条件に、円筒型、三角柱型モジュールを加えた場合の、n, mの値はそれぞれ、 -12.87と-2.42となる.これらの値は、立方体型、円筒型、三角柱型の相互作用係数、角速度平均 値をまとめて標準化し、その内、立方体型の場合の標準化した相互作用係数と角速度の大きさの 平均値を従属変数とし、出力時間を独立変数として、先ほどと同様に回帰分析を行い、得られた 線型方程式の相互作用係数と角速度平均値の係数として求められた値を用いた.この場合、出力 時間は、立方体型のモジュールの場合のみ適用しているため、この相互作用度数から、円筒型、 三角柱型の出力時間の予測をすることはできないが、円筒型、三角柱型の相互作用度数から出力 時間の相対的な傾向を予想することはできると考えられる.

立方体型,円筒型,三角柱型それぞれの相互作用度数の平均値の比較を図23に示す.この図から,三角柱型,立方体型,円筒型の順に相互作用度数が高くなることが確認できる.実際に,相 互作用度数が高いほど,出力時間が短くなる傾向があるとすると,三角柱型,立方体型,円筒型 の順に,出力時間が短くなると予想される.

34





続いて、立方体型、円筒型、三角柱型の相互作用度数の、領域サイズ、領域形状、運動パター ンでの平均値の比較を図24に示す.立方体型では、相互作用度数から予想される出力時間の傾向 は、実験2・1で計測した出力時間の平均値と概ね似た傾向であることが分かる.平均値自体を見 ると、実際の出力時間の傾向と比較し、領域形状では、円型、非対称型、運動パターンでは環状 型、ランダム型がそれぞれ入れ替わる結果となったが、誤差の範囲内であると考えられる.

円筒型のモジュールでは、領域サイズに関しては、実験2・1で測定した出力時間の傾向と類似 の傾向が確認できたが、領域形状では、円型、非対称型、運動パターンでは、環状型、ランダム 型がそれぞれ入れ替わる結果となった.立方体型の場合と比較し、こちらの場合は有意な差があ ると考えられる.

三角柱型のモジュールでは、立方体型とほぼ同様の傾向であった.

35



図24. 領域サイズ,領域形状,運動パターンごとにまとめた相互作用度数の平均値 の比較 A:立方体型モジュールにおける領域サイズ,領域形状,運動パターンごと にまとめた相互作用度数の平均値の比較 B:円筒型モジュールにおける,領域サイ ズ,領域形状,運動パターンごとにまとめた相互作用度数の平均値の比較 C:三角 柱型モジュールにおける領域サイズ,領域形状,運動パターンごとにまとめた相互 作用度数の平均値の比較

次に、立方体型、円筒型、三角柱型をまとめた、領域サイズ、領域形状、運動パターンごとの 平均値を図25に示す.領域サイズに関しては、小型の領域の相互作用度数が低く、中型、大型に 関しては有意な差が見られなかった.領域形状では、正方形型、非対称型、円型、運動パターン では、並行型、ランダム型、環状型の順に、相互作用度数が高いという結果となった.この結果 から予想される出力時間の傾向は、実験2-1の結果から得られた結果と比較して、領域形状では 円型と非対称型、運動パターンでは環状型とランダム型の値が逆転しており、差も大きくなって いる.こうした傾向は、詳しくは後述するが、円筒型のモジュールの角速度の大きさの平均値が 非常に高かっために生じていると考えられる.



図25. 立方体型,円筒型,三角柱型をまとめた場合の領域サイズ,領域形状,運動パターン ごとの平均値の比較

続いて,立方体型,円筒型,三角柱型のモジュールのそれぞれにおける速さ,角速度の大きさの確率分布について見ていく.まず,立方体型モジュールの場合について確認し,続いて,円筒型,三角柱型の場合と比較する.

立方体型の場合の、相互作用係数、速さ、角速度の大きさに関してまとめたグラフを図26, 27に示す.図26にあるように、速さをヒストグラムにして、領域サイズ、領域形状、運動パタ ーンの各組み合わせで比較したところ、モジュールの速さの分布には、運動パターンが大きく影 響していることが分かった.領域形状やサイズによる影響は比較的少ないが、領域サイズでは、 サイズが大きくなるほど、速さ0付近の確率が低くなるといった傾向が確認でき、領域形状では、 環状型の運動パターンが、影響を受けやすいといった傾向が確認できる.また、大型の円型形状 と環状型の運動パターンの組み合わせの場合だけ、他とは大きく違う傾向を示した. 角速度の大きさも同様にヒストグラムにして比較したところ、領域サイズによる確率分布の差 異はほとんど見られず、領域形状と、運動パターンによる影響が強いことが分かった.傾向とし ては、正方形型形状では角速度の大きさ0付近での確率が高くなる、円型領域形状と環状型運動 パターンの組み合わせでは、角速度の大きさ0付近の確率が低くなるといったことが確認でき た.



図26. 実験2-2における対象モジュールの速さの確率分布 上から順に領域サイズ,大,中, 小. 左から順に領域形状,円型,正方形型,非対称型.運動パターンは,平行型,環状型,ラ ンダム型ごとに色分けをして表示している.この確率分布から,モジュールの速さには,領域 の運動パターンが最も強い影響を与えていることが分かる.



図27. 実験2-2における対象モジュールの角速度の大きさの確率分布 上から順に領域サイズ,大, 中,小. 左から順に領域形状,円型,正方形型,非対称型.運動パターンは,平行型,環状型,ラ ンダム型ごとに色分けをして表示している.この確率分布から,モジュールの角速度の大きさに は,角速度の大きさの分布には,組み合わせごとの差異はあまり見られないが,正方形型形状では 角速度の大きさ0付近での確率が高くなる傾向や,円型領域形状と環状型運動パターンの組み合わ せでは,0付近の確率が低くなる傾向などが確認できる.

続いて,円筒型,三角柱型の場合について図28~31に示す.先ほどの立方体型の場合と比較 して,三角柱型はほぼ同じような傾向であった.一方,円筒型の場合では,円型形状と環状型の 運動パターンの組み合わせの分布が他の場合と大きく異なる結果となった.

この円型形状と環状型の運動パターンの組み合わせのうち中型の領域サイズの確率分布を三種類のモジュール型で比較したものを図32に示す.この図からも明確に分かるように、同サイズ、同形状、同型の運動パターンであっても、モジュールの形によって、容器内でのモジュールのふるまいに差が生じる場合があることが確認できた.



図28. 円筒型モジュールの速さの確率分布



図29. 円筒型モジュールの角速度の大きさの確率分布



図30. 三角柱型モジュールの速さの確率分布



図31. 三角柱型モジュールの角速度の大きさの確率分布



図32. 速さと角速度の大きさの確率分布の比較 A:立方体型,円筒型,三角柱型の3種類 のモジュールの,中型の円型領域形状,環状型の運動パターンの組み合わせの場合の速さ の確率分布の比較. B:Aの場合と同条件による速さの確率分布の比較.速さ,角速度の 大きさともに,立方体型,三角柱型は分布の傾向に大きな差異はないが,円筒型は分布の 傾向が大きくことなることが確認できる.

実験2-2では容器内でのモジュールの軌道の記録も行った.このうち,立方体型モジュールの 小型の領域の場合における軌道を図33に示す.これらの軌道の記録から定性的に判断すると,特 に,領域の運動パターンが軌道のパターンに傾向が強いと捉えることができるだろう.



図33. 各領域,運動パターンにおける領域内のモジュールの軌道の比較

4.3.4. 考察

実験2-2の結果から、領域形状や運動パターンなどの環境が変わることで、領域内でのモジュ ールのふるまいに様々な差異が生じることが確認できた.

また,出力時間は,相互作用係数と角速度の平均値からなる相互作用度数と強く関連があるこ と可能性が高いことが示された.この相互作用度数は,領域内で,モジュールがいかにランダム に振る舞っているかを示す指標ともなり得ると考えられ,モジュールがランダムに移動するほど, アセンブリの出力時間は短くなることが予想される.今後,容器内のモジュールのふるまいの乱 雑度自体を別の尺度で解析し,相互作用度数や出力時間との関連を調べる必要があるだろう.

また,相互作用度数は,領域サイズ,領域形状,領域の運動パターンの種類の違いにおいて, それぞれ有意な差異が見られた.領域サイズでは,小型で最も低く,中型,大型では大きな差異 は見られなかった.この結果からも,領域サイズ,つまり,領域内でのモジュールの密度として 適した値があることが予想される.今回の実験で用いた中型,大型の領域では大きな差異が見ら れなかったため,今後,より大きなサイズの領域を用いて同様の実験を行う必要があるだろう.

また、相互作用度数を領域形状でまとめた平均値を見てみると、正方形型、非対称型、円型の 順、運動パターンで平均値をとると、平行型、ランダム型、環状型の順に大きな値を示した.こ れは、現段階では、定性的な判断となるが、領域形状、運動パターンともに、形のパターンによ って類似の傾向があると捉えることもできる。例えば、領域内でモジュールが乱雑にふるまう傾 向が、領域形状、運動パターンともに、円に近づくほど高くなる可能性などが考えられる。今後、 より多くの種類の形状や運動パターンを適用して実験をするとともに、形状自体の解析を行って いくことで、形の複雑さと、その形が生み出す乱雑さとの関係を見いだして行くことができるか もしれない.

また、3種類のモジュール型で、速さの確率分布と、角速度の大きさの確率分布をそれぞれ比較したところ、同サイズ、同形状、同型の運動パターンであっても、モジュールの形によって、容器内でのモジュールのふるまいに差が生じる場合があることが確認された。特に、他との違いが顕著だったのは、円筒型モジュールと、円型領域形状、環状型の運動パターンという同系統の形、パターンの組み合わせであり、形やパターンの組み合わせと、容器内でのモジュールのふるまいにある種の規則性が見いだせる可能性が示唆された。

46

4.4. 実験 2-3

4.4.1. 目的

実験2·3では、複数の領域の形状、運動パターンの組み合わせを適用させ、それぞれの組み合わせにおける、目的形状が出力されるまでのアセンブリの順序を記録する.この実験によって、領域形状、サイズ、運動パターンの組み合わせによって、アセンブリの順序に差異が生じうるかを調べ、これまでの実験と同様に、セルフアセンブリシステムのふるまいと形、パターンの関係を考察する.

4.4.2. 方法

領域形状,運動パターンに関しては,基本となる形,パターンを用い,領域サイズは中型のみ で行う.目的形状は、小文字の"f"を模した形状とする(図35).この目的形状を構成するモ ジュールセットと結合ルールは、表5,6にあるように、モジュール8個,結合パターン14種類で ある.アセンブリ順序を記録することが目的であるため、構成するモジュールの種類は全て異な るように設定した.

領域サイズ,領域形状,運動パターンの組み合わせごとに,100回ずつ試行し,そのアセンブ リされる順序を記録する.これは,あらかじめ各結合に番号をふっておき,結合が生じる度にそ の番号を記録するという方法で行う.

目的形状である"f"型形状を構成するのに必要な結合は7カ所であるため、目的形状のアセンブリが完了した時点で、1~7からなる7桁の数字が得られる.この得られた、7桁の数字からアセンブリ順序が一意に決まる.



図34. "f"型形状がアセンブリされた様子

module ID	concave-convex pattern distribution
11	M
12	T N
13	U - O -
14	V S P R
15	W - G -
16	Х
17	Z
18	Y

表5. モジュールと凹凸パターンの対応表

表6. 結合ルール一覧
reaction rule set
$M + T \rightarrow M^-T$
$N + U \rightarrow N^-U$
$O + V \rightarrow O - V$
$P + W \rightarrow Q - X$
$R + Y \rightarrow R$ -Y
$S + Z \to S^{\text{-}}Z$

4.4.3 結果

実験2·3では、複数の領域の形状、運動パターンの組み合わせを適用させ、それぞれの組み合わせにおける、目的形状が出力されるまでのアセンブリの順序を記録した.

今回は、アセンブリ順序の組み合わせ数に対して、試行回数が少ないため、アセンブリ順序の 全ての段階で比較するのではなく、最後に行われた結合の番号をもとに比較をした.その結果、 図35にあるように、アセンブリの最後に結合される結合箇所に偏りが見られた.最も偏りの多か ったもので、組み合わせで3倍以上の開きがあった.

また、図36にあるように、各領域形状、運動パターンごとに最後に結合された結合箇所の番号 の分散の平均値を出し、比較した.その結果、領域形状では非対称型が、運動パターンではラン ダム型の分散の平均値が低く、アセンブリ順序のパターンに偏りが少ない結果となった.



図35. アセンブリ順序のパターン比較 一列目から順に円型,正方形型,非対称型の形状,一行目 から順に平行型,環状型,ランダム型の運動パターン.各領域形状,運動パターンの組み合わせご とに,100回ずつシミュレーションを行い"f"型形状がアセンブリされる順序を記録し,最後に結合 された箇所の数を合計し,比較したもの.横軸が,最後に結合された箇所の番号.縦軸がその数の 合計値.右下の非対称型形状,例えば,ランダム型運動パターンでは,アセンブリ順序に偏りがな いが,円型形状,平行型運動パターンなどは,最も少なかった順序パターンと最も多かった順序パ ターンで3倍もの差があることが確認できる.



図36. アセンブリ順序の分散の平均値の比較 左が領域形状に関してまとめたもの, 右が領域の運動パターンに関してまとめたもの.

4.4.4. 考察

この結果からアセンブリの順序には,領域の形状や運動パターンの組み合わせによって偏りが 生じる可能性があることが分かった.

また,特に偏りが少なかったのはランダム型の運動パターンであった.実験2・2で記録したラ ンダム型の運動パターンの軌道を見てみると,他の運動パターンの軌道に,ある種のパターンが 見られるのに対して,ランダム型の運動パターンでは,比較的そのようなパターンが見られな い.

このように、領域内でのモジュールのふるまいのパターンがアセンブリ順序の偏りに繋がって いる可能性がある.領域形状や運動パターンとアセンブリ順序の偏りの関係がより詳しく分かれ ば、領域形状や運動パターンによって、アセンブリシステムのふるまいをゆるやかに制御するこ とが可能となるかもしれない.

ただし、この結果から、単純に、領域の形と運動パターンによってアセンブリ順序にばらつき が出ると断言はできないだろう.今後、試行回数を増やし、様々な目的形状を試すとともに、実 空間での実験を行う必要がある.

5. 結論·展望

本研究では、人工のセルフアセンブリシステムの設計及び実装を行い、そのセルフアセンブリ システムを用いて、形の計算に関するいくつかの実験を行った.その結果、各モジュールの形と、 各領域サイズ、領域形状、領域の運動パターンなどの環境との相互作用によって、それぞれの組 み合わせでのセルフアセンブリシステムのふるまいに変化が見られた.また、その変化の傾向と、 形やパターン自体の性質に関連性がある可能性が見いだされた.今後、形やパターン自体の性質 とセルフアセンブリシステムのふるまいとの関係を深く探っていく必要があるだろう.

本研究では、手探りで様々な実験を行ってきたが、物理シミュレーション環境では、様々な形 やパターンを簡単に試せることから、こうした研究においても有効な手段の一つとなるだろうと いう可能性があると感じた.しかし、実空間における形の相互作用では、まったく別の結果が得 られる場合もあり、基本的には、実空間での実験に重きを置くべきだろう.今後、実空間で同様 の実験をして、物理シミュレーションの結果と比較し、物理シミュレーションの結果がどの程度 信頼できるかといったことも検証していく必要がある.

また、このような、形の計算ともいえる、形の相互作用について研究することで、セルフアセンブリシステムの理解が進み、未来のデジタルファブリケーション技術としてのセルフアセンブリシステムの実現に近づいていくことが期待される.

例えば、現在までに、モジュールの種類に制限のないモデルにおいて任意の三次元形状を構成 するためのセルフアセンブリシステムの研究はなされてきているが[13]、DNAのようにモジュー ルの種類が少数に限定されているモデルにおいて任意の三次元形状をつくる手法の研究はほとん どなされていない.そこで、現在、複数段階のエネルギーを用いることで、タンパク質の合成の ように、一度鎖状の構造に結合し、その後、折りたたまれることで、立体構造を形成するという セルフアセンブリシステムの実現に向けた研究を進めている(図37).このように複数段階のエ ネルギーを用いるなど、より複雑なセルフアセンブリシステムにおいては、環境とモジュールと の相互作用の制御が重要になってくると考えられ、形の計算的考察が重要な役割を果たしてくる と考えられる.



図37. フォールディング型のセルフアセンブリシステム概念図

また,形の計算の研究は,生命をはじめとした自然現象の理解にも繋がっていく可能性がある. その際,形の計算の理論をつくっていくことが重要な課題となるだろう.

Turingが提案したチューリングマシンのモデルは、人が算術をする際に紙に数字を書く、消す、 置き換えるという操作を抽象化することで構想したとされているが、これを人の手仕事によるも のづくりと、従来の人工物の生産手法との関係と重ねて考えることはできないだろうか.未来の デジタルファブリケーション手法と同様、人間の身体のアナロジーから抜け出し、自然のあり方 を参照することで、新たな計算パラダイムに繋がって行くかもしれない.それは、いわゆる計算 とは異なるものであるかもしれないが、そのような計算では得られない結果を得ることができる だろう.

6. 謝辞

本研究を進めるにあたり,指導教員である田中浩也准教授には,研究のテーマ設定をはじめ, 毎週のゼミでの長時間の議論などを通して多くの指導をしていただきました.大変感謝いたして おります.また,3DプリンタやCNCレーザーカッターなどの様々な工作機械の整った素晴らし い研究環境を与えてくださったことで,充実した研究活動を行うことができ,大変感謝いたして おります.

学部二年時より所属していた先端生命科学研究会の内藤泰宏准教授には,先端生命科学研究会 での研究だけでなく,本研究をはじめとした田中浩也研究会での研究を含め,様々な議論を通し て指導して頂きました.大変感謝いたしております.

遡って考えると、「シミュレーションデザイン」という授業を履修したことが、今の研究にも 繋がっていると思います.授業を担当してくださった、井庭崇准教授、古川園先生に感謝いたし ます.

また,毎週のゼミでの長時間の議論や,日々のものづくりや研究など,刺激的で,楽しい時間 を共に過ごした田中浩也研究会のメンバーに感謝します.

7. 参考文献

- [1] N. Gershenfeld, Fab: The Coming Revolution on Your Desktop--from Personal Computers to Personal Fabrication. Basic Books.
- [2] L. Adleman, "Molecular computation of solutions to combinatorial problems," *SCIENCE-NEW YORK THEN WASHINGTON-*, 1994.
- P. W. K. Rothemund, "Design of DNA origami," ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005., pp. 471–478.
- P. W. K. Rothemund, "Folding DNA to create nanoscale shapes and patterns Supplementary Notes 1 – 11 Supplementary Note S1: Design of DNA origami, "pp. 1–82.
- [5] B. T. Laboratories and M. Hill, "Proving Theorems by Pattern Recognition I," vol. I, 1959.
- [6] G. M. Whitesides and M. Boncheva, "Beyond molecules: self-assembly of mesoscopic and macroscopic components.," *Proceedings of the National Academy of Sciences of the United States* of America, vol. 99, no. 8, pp. 4769–74, Apr. 2002.
- [7] L. Penrose, "MECHANICS OF SELF-REPRODUCTION," Annals of human genetics, 1958.
- [8] L. Penrose, "Self-reproducing machines."
- [9] J. Breivik, I. Life, and N. R. Hospital, "Self-Organization of Template-Replicating Polymers and the Spontaneous Rise of Genetic Information," pp. 273–279, 2001.
- [10] M. Matsumoto and S. Hashimoto, "Passive Self-Replication of Millimeter-Scale Parts," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 2, pp. 385–391, Apr. 2009.
- [11] N. Virgo, C. Fernando, B. Bigge, and P. Husbands, "Evolvable physical self-replicators.," *Artificial life*, vol. 18, no. 2, pp. 129–42, Jan. 2012.
- [12] N. Bhalla and P. Bentley, "Staging the Self-Assembly Process Using Morphological Information," *Proceedings of European*..., 2011.
- [13] N. Bhalla, P. J. Bentley, P. D. Vize, and C. Jacob, "Programming and evolving physical self-assembling systems in three dimensions," *Natural Computing*, pp. 475–498, Nov. 2011.
- [14] B. A. Grzybowski, H. A. Stone, and G. M. Whitesides, "Dynamic self-assembly of objects rotating at a liquid ± air interface," vol. 405, no. June, 2000.
- [15] M. T. Tolley and H. Lipson, "Fluidic Manipulation for Scalable Stochastic 3D Assembly of Modular Robots," pp. 3–8, 2009.
- [16] M. Krishnan, M. T. Tolley, H. Lipson, and D. Erickson, "Hydrodynamically tunable affinities for fluidic assembly.," *Langmuir : the ACS journal of surfaces and colloids*, vol. 25, no. 6, pp. 3769–74, Apr. 2009.
- S. Griffith, D. Goldwater, and J. M. Jacobson, "Robotics: self-replication from random parts.," *Nature*, vol. 437, no. 7059, p. 636, Sep. 2005.

- S. Miyashita and R. Pfeifer, "Attributes of two-dimensional magnetic self-assembly," *Adaptive Behavior*, vol. 20, no. 2, pp. 117–130, Dec. 2011.
- K. Nakajima, A. M. T. Ngouabeu, S. Miyashita, M. Göldi, R. M. Füchslin, and R. Pfeifer,
 "Morphology-induced collective behaviors: dynamic pattern formation in water-floating elements.," *PloS one*, vol. 7, no. 6, p. e37805, Jan. 2012.
- [20] L. Adleman, Q. Cheng, and A. Goel, "Combinatorial optimization problems in self-assembly," *Proceedings of the thiry ...*, p. 23, 2002.
- [21] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. A Bradford Book.
- [22] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional.
- [23] D. H. Ackley, "A connectionist machine for genetic hillclimbing," Jul. 1987.
- [24] G. Syswerda, "Uniform Crossover in Genetic Algorithms," pp. 2–9, Jun. 1989.
- [25] T. MASUDA and T. UEDA, "A Novel Method for Distribution System Loss Minimization Problem Based on Dynamic Parametric GA Using Fuzzy Reasoning Rules," ... Japan Society for Fuzzy ..., 1999.
- [26] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," vol. 1, no. 212, 2007.