

植物生長のモデリングに関する
基礎的研究

平藤 進之

①

植物生長のモデリングに関する
基礎的研究

平藤 雅之

- 目 次 -

1 序論.....	5
1. 1 研究の背景と目的.....	6
1. 1. 1 コンピュータの進歩.....	6
1. 1. 2 大規模プログラミングとフレーム問題.....	7
1. 1. 3 エキスパートシステムへの期待と限界.....	10
1. 2 モデリングにおける考え方.....	15
1. 2. 1 基礎的研究としての目的.....	15
1. 2. 2 生物を模倣したアルゴリズム.....	16
1. 2. 3 複雑なシステムの再帰的理解.....	18
1. 2. 4 データに関する情報理論的な問題.....	19
1. 2. 5 人工生命研究との関係.....	21
2 形態的变化のモデル化.....	23
2. 1 形態モデルの問題点.....	24
2. 2 形態モデルのアルゴリズム.....	26
2. 2. 1 Lシステム.....	26
2. 2. 2 Lシステムを使ったサイズのモデル.....	27
2. 2. 3 ファジィ・ニューラルネット.....	30
2. 2. 4 マクロネットワークによるメタルールの構築.....	35
2. 2. 5 サイズのメタルールの構築.....	38
2. 3 考察.....	43
2. 3. 1 プログラミング手法としての発展性.....	43
2. 3. 2 形態モデルの逆問題とその応用方法の考察.....	44
3 量的変化のモデル化.....	47

3. 1	従来のモデル化手法の問題点	48
3. 1. 1	コンピュータプログラムとしての必要条件	48
3. 1. 2	回帰的モデルにおけるモデルの良否評価	49
3. 1. 3	生長パターンの「ゆらぎ」	51
3. 2	器官間の干渉を考慮した生長モデル	53
3. 2. 1	多体系としてみた植物の生長変化	53
3. 2. 2	器官生長モデルの定式化	53
3. 2. 3	器官ネットワークモデル	56
3. 2. 4	Lotka-Volterra 方程式の数値解との比較	58
3. 2. 5	器官生長モデルの拡張	60
3. 3	モデルのシステム同定手法	63
3. 3. 1	同定アルゴリズム	63
3. 3. 2	システム同定例	69
3. 4	考察	72
3. 4. 1	推定されたパラメータの意味	72
3. 4. 2	ニューラルネットとの比較	72
3. 5	結論	75
4	生長の複雑さに関する実験と予測モデル	76
4. 1	実験の目的と概要	77
4. 2	カオス性の判定手法	79
4. 2. 1	観測データに基づく判定手法	79
4. 2. 2	相関次元の計算法	80
4. 3	実験	83
4. 3. 1	実験方法	83
4. 3. 2	実験結果	87
4. 4	カオス性の解析	91
4. 4. 1	ストレンジアトラクタの観察	91
4. 4. 2	相関次元によるカオス性の判定	92

4. 4. 3 非線形モデルの同定による検証	93
4. 5 カオス性の実験的検証	97
4. 5. 1 実験方法	97
4. 5. 2 実験結果	99
4. 5. 3 光合成速度の予測モデル	104
4. 6 考察	107
4. 6. 1 光合成速度のカオス	107
4. 6. 2 カオスの発生と協調的なメカニズムの存在の可能性	108
4. 6. 3 カオス性の応用	108
4. 6. 4 複雑さとあいまい性	110
4. 7 結論	111
5 議論 —モデルの一般化について—	112
5. 1 器官ネットワークモデルの一般化	113
5. 1. 1 モデルの汎用性	113
5. 1. 2 ハードウェア化の可能性	115
5. 2 カオス系のモデルのシステム同定	119
5. 3 生物の階層性を考慮した生物系の一般モデル	122
5. 4 マクロ系とメソスコピック系の特徴を備えたモデル	125
5. 4. 1 メソスコピック系のモデル	125
5. 4. 2 メソスコピック系の特徴を持つ生物モデル	126
5. 4. 3 数値解法	129
5. 4. 4 巨大分子系による最適化計算と定向突然変異のモデル	134
5. 4. 5 生物現象と非生物現象の定義	136
6 結言	138
謝辞	140

付録.....	142
A MACNETのソースコード.....	142
A-1 引数と戻り値.....	142
A-2 ソースコード.....	145
B ハードウェアLVモデルの回路図.....	156
参考文献.....	158

1 序論

近年の情報化の進展は産業構造や社会構造を非常に速いペースで変化させている。その結果、かつて繁栄を誇った産業のいくつかは没落し、新たに情報関連産業が急速に発展している。自動車産業のように現在も繁栄を維持している産業では、CIM や工業用ロボットの導入に見られるように、コンピュータ技術の進歩が生産性の向上に寄与する仕組みを備えている。

農林水産業においても生産性を向上させるために、統計モデル、線形計画法、システムダイナミクス・モデルなどから始まってエキスパート・システム、ニューラルネットの応用など、コンピュータを生産管理等に利用するための研究が行われて来た¹²⁾。また、農業用BBS(パソコン通信システム)や電子産直システム³⁾などは、かなり早い時期から実用化されている。コンピュータの利用に対する関係者の努力は他産業と比較して劣っているとはいえ、むしろ先進的ですからある。しかしながら、最も重要な生産場面においてはコンピュータはほとんど利用されていない。これが、他産業に比べて生産性を低くしている大きな要因の一つとなっている。

ここでは、この問題を「生物を含む複雑なシステム」のモデル化の観点から論じるとともに、本研究での取り組み方の概要⁴⁾を述べる。また、本研究と深いつながりを有する人工知能や人工生命の研究との関連、モデル構築に不可欠なデータについて情報量などについて述べる。

1. 1 研究の背景と目的

1. 1. 1 コンピュータの進歩

コンピュータの能力は2～3年で倍増するといわれており、この10年間で約100倍になった。西暦2000年には2,000Mipsもの計算パワーを持つパソコンが出現すると予測され(図1-1)、このトレンドは少なくとも2010年あたりまで続くことが予想されている。このような大きな変化が数十年にわたって継続した結果、情報機器の性能向上が生産性の向上に直結する仕組みを備えている産業と、そうでない産業とでは、生産性に非常に大きな格差が生じて来ている。

近年、農業分野でも生産の効率化の切り札として「情報化」が叫ばれ、現場では種々の事業の下で、コンピュータの導入が大がかりに推進されている⁵⁾。

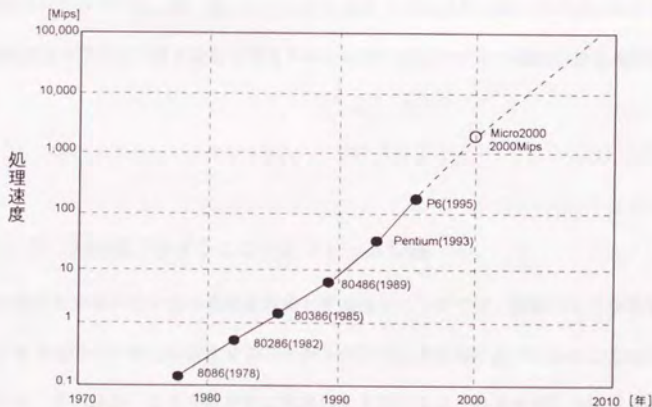


図1-1 80系CPUの処理速度の変化 (Intelの計画より作成)

しかし、農林水産業など生物を対象とした産業では、コンピュータの利用は一部に止まっている。生産性向上に重要な生産管理、生産制御、作業管理などの場面では、コンピュータはほとんど利用されていない。この原因を列挙すると、

- ①それぞれの業務ですぐに利用できるソフトが少ない
- ②利用したときの効果が少ない
- ③個々の問題解決に必要なデータが十分得られない
- ④操作が難しい
- ⑤導入に要する費用が高い

などが挙げられる。

④、⑤は、計算機のハードウェアやOSの進歩に伴って自動的に解消する問題である。しかし、①、②、③には、以下のような生物が関与するプログラムに特有の難しい問題が内包されており、根本的な解決策が必要である。

本研究の目的の一つは、①、②、③に内在する種々の原因を情報科学の観点から考察し、その解決法の一つとして最も重要と考えられた植物の生長モデルの構築方法を見いだすことである。

1. 1. 2 大規模プログラミングとフレーム問題

複雑な構造と非線形性を持つ現象を対象とするモデリングでは、影響の小さな要素であっても、できるだけモデルに反映させることがモデルをより現実になづけることになると考えられている。その結果、モデルは非常に大規模なものとなる。対象の構造の複雑さと非線形性のため、モデルを構成する方程式系も複雑にならざるを得ない。たとえば、最近の植物生態系の数理モデル⁸⁾は、非常に複雑な多数の方程式から構成されている。これほど、方程式系

が複雑になって来ると、モデル中のパラメータや数式を追加や削除したとき、その操作によってモデルがどの程度改善されたかを検証することは困難であると思われる。

大規模モデルの構築とは大規模プログラムの構築に他ならない。大規模プログラムの構築では、プログラムのステップ数が増大するにつれてプログラムの複雑さが指数関数的に増大し、プログラム全体の一貫性を維持することが困難になる「大規模プログラミング」の問題が存在している。たとえば、プログラムが大規模になるに従って完璧なデバッグは困難となり、ソフトウェアとしての品質維持が難しくなってくる。そのため、現実には無制限にモデルを複雑にすることは不可能である。

複雑なシステムでは、関連する要因がさらに別の要因の影響を受けるという関係が限りなく連鎖し、そのため、モデルの大きさは爆発的に増大する。このパラドックスの存在は人工知能研究では古くから知られており、「フレーム問題」と呼ばれている（「あるルールを実行するためには関連する別のルールが必要になり、さらにそのルールを実行するためには、また別のルールが必要になる、という連鎖が無限に続き、非常に簡単と思える問題が解けなくなる」というパラドックスである。近年、フレーム問題が、人工知能の本質に関わる重要な問題として認識されるようになった⁷⁾）。たとえば、医療診断を行うエキスパート・システムではフレーム問題のため、知識ベースが大規模なものとならざるを得ないことが多い⁸⁾。

複雑なシステムであっても各変数間の関係が線形な場合は、影響の小さな変数を無視することで、この連鎖の分岐を打ち切ることができる（たとえば、「枝刈り」と呼ばれるプログラミング技法がある）。チェスや将棋などのようにドメインが限定されていれば、枝刈りをうまく行うことでフレーム問題は回避できる。しかし、生物を含むシステムでは、生物に伴う非線形性により、非常に小さな変化であってもシステム全体に大きな影響を与える可能性があるため、枝刈りが難しくフレーム問題は避け難い。

むしろ生物を扱う研究分野ではフレーム問題の存在などは当然であり、フレーム問題にこ

だわるとモデリングを行うこと自体がナンセンスになってしまうということもあるだろう。また、複雑な現象を扱う研究分野では、モデルを線形化（準線形化）し多変量解析など使つてデータの中に埋もれたわずかな規則性を見つけたことに傾注して来たという経緯がある。線形モデルの世界では、フレーム問題の存在を考える必要はないため、フレーム問題への認識は希薄にならざるを得ない。

そのため、フレーム問題に対しては、ある種の思考停止状態にあり、大規模モデルに対するソフトウェア科学的な議論は極めて少ない。そのため、初期のエキスパートシステムに似て、モデルに盛り込んだ知識や数式の多さを誇る傾向がある一方で、重回帰モデルのような単純なモデルに対する過大評価がある。

人工知能研究以外の研究者の場合、興味が主としてそれぞれの問題にあるため、モデリングにおけるフレーム問題への認識は一般に乏しいように思われる。生物の非線形性や複雑性をモデルに反映させ、そのモデルをコンピュータ・プログラムにしようとしたとき、フレーム問題は大規模プログラミングの問題として顕在化して来る。大規模プログラミングの困難さに対する認識が低いと、大規模モデルが構築できるかどうかの問題はプログラミング能力の問題とみなされることになる（大規模プログラムの開発が失敗した場合の原因をプログラマの能力不足に帰着させると、この問題はますます解決が困難になる。）。冒頭で列举した「利用できるソフトが少ない」などといった問題は、プログラマやソフトハウス、あるいは行政や研究者の「能力不足」や「怠慢」として片づけられがちであるが、大規模プログラミングやフレーム問題という難問が存在しているためであることが一つの大きな原因である。

ここで、植物生長のモデリングに限定すると、列举したいいくつかの問題は以下のようなソフトウェア上の問題から派生していると考えることができる。

(1) 植物の複雑さ

植物の複雑さを処理する手法が確立されていない。サンプル数や説明変数をいくら増やしても予測精度が向上しないことが多く、情報（データや知識）の蓄積に応じて少しずつモデルを改善して行くことも困難である。

（2）必要な情報の不足

対象とする問題の複雑さに比して、十分な情報が得られるケースはまれである。

（3）情報の形態の多様性

入手可能な情報には、あいまいなデータ、論理的知識、主観的知識、事例などが混在している。

（4）モジュール化の欠如

従来のモデルの多くはブラックボックス的な利用が困難であり、他のプログラムに簡単にリンクできない（大規模なモデル構築では、比較的小さなモデルを結合して大規模にする必要がある）。

従来の植物生長モデルは、他分野で成功したモデル化手法をそのまま植物に適用しただけのものがほとんどである（たとえば、重回帰分析などの統計モデルやシステム・ダイナミクスに基づいたものなど）。あいまいな情報の活用や多様な情報の統合化は人工知能研究において盛んに研究されていることから分かるように⁹⁾、既存のモデル化手法の枠組みだけで問題を解決しようとするのはもともと難しいと考えられる。

1. 1. 3 エキスパートシステムへの期待と限界

植物を含む複雑なシステムに対する問題を解決する新しい方法として、「専門家や篤農家のあいまいな知識を利用する」という方法が大きな期待を持って迎えられ、世界各国でエキ

スパートシステムの開発が行われた^{10,11)}。我が国でもほぼ同時に多数のエキスパートシステムの開発が試みられ¹²⁾、筆者らも農業用のエキスパートシステムをいくつか構築した^{13,14)}。また、異なるエキスパートシステム構築ツールを用いて複数のエキスパートシステムの構築を実験的にを行い、エキスパートシステムの構築におけるツールや手法の違いの問題についても詳しく調べた¹⁵⁾。これらの結果からは、コンバインの故障診断のような生物との関係が比較的希薄な問題ではドメインの限定が容易であり、エキスパートシステムの開発は非生物系のエキスパートシステムと同様にそれほど難しくないことが判明した。しかし、作業管理のように植物の生育や病気の発生に深く関わる問題では、知識ベースの大規模化が避けられないことが明らかとなった。

知識ベースシステムのメリットは、本来、複雑な問題を複雑なまま表現することである。そのための具体的なプログラミング方法として知識とプログラムを分離し、大量の知識をデータベース的に管理する。このようにして作られたプログラムが知識ベースシステムであり、通常のプログラミング手法に比べて、大規模で複雑な問題を少ないステップ数で表現できるという利点がある。ただし、この利点は相対的なものであって、知識ベースが大規模になるにつれてプログラミングは、当然、難しくなってくる。たとえば、通常のエキスパートシステム構築ツールでは、ルール間の論理関係を、プロダクションルールや述語論理式で表現する（内部での実体は、Lisp や Prolog などを用いた文字列のパターンマッチングと関数の実行）。ルールを構成するシンボル（概念）の一つ一つに対しては、フレームを用いてシンボル間の階層性や依存関係などの定義を行う。このような方法によって、入力するコード数は圧縮されるが、その反面、一つ一つのルールを入力する際には細心の注意が必要であるため、通常のプログラミング言語によるコーディングよりも1ステップ当たりについて見ると大きな手間がかかる。

筆者らの構築実験では、ルール数とフレーム数の合計が数十個を超えると知識ベースの構

築作業は急速に複雑となり、数 100 個に達すると知識ベース全体の論理的整合性を維持することが極めて困難になった。生物に関わる問題では関連する知識が爆発的に増大するため、既存のエキパートシステム構築ツールが提供する知識ベース入力機能では、生物に関わる問題に必要な大量のルールを入力することは、到底、不可能と判断された。

そこで、筆者らは自然言語処理によって文書ファイルから自動的に大規模知識ベースを構築し、異なる知識ベースシステム間で知識の共有化を可能にする手法の研究を行った^{16,17)}。これは、自然言語（英文）で書かれた知識を、あらかじめ入力しておいた生成文法に関する知識と辞書データに基づいて構文解析し、自動的に記号論理式に変換するという方法である（文法知識と辞書データを変えることで日本語にも対応できる）。

この方法では、フルテキストが入力された文献データベースを充実させることで、自動的に知識ベースが増大するという利点がある。文法構造に基づいた記号論理式への変換には、多義性などから生ずるあいまい性があるが、少なくともキーワードの文字列マッチングを用いた全文検索よりも、目的とする情報を探し出し易い。最近では、大規模知識ベースの構築を行うために、インターネットを使って蓄積された大規模な文書情報を利用するというアプローチが提案されている¹⁸⁾。そのため、大規模な文書情報を大規模知識として利用する技術はますます重要となり、次第に実用性が高まっている。実際には、ある程度、制約された文法（前後の文脈に依存しない文脈自由文法）で書かれたテキストしか知識ベース化できないという制約があるが、原理的には知識ベースの入力に関する問題は解決したとして議論を先に進めることができる。

ところが、植物生長に関する知識を表現するためには形態の構造や大きさの変化の複雑さを扱う必要があり、これらは記号論理的な知識表現形式では記述が困難であった。また、環境変化に対する植物生長の変化に関する知識も記述が難しいものであった。

そこで、本研究では植物生長における形態変化のモデル化を行った（「2 形態的変化の

モデル化」)。ここでは、あいまいかつ多様な情報の総合的な利用、モデル(知識)のモジュール化を行う方法について検討し、植物生長のモデリングを多様かつあいまいな情報を表現する問題として考えた。ここで開発したモデルは、植物生長に関する様々な情報(知識、実験データなど)を統合化する、一種の知識ベースシステムである(このモデル化の方法はフレーム問題を本質的に解決するものではなく、フレーム問題の困難を「改善」することを目的としたものである)。

さて、あいまい性があるルールが多数連鎖していると、推論を重ねるにつれて推論結果のあいまい性が増大し、結局、ほとんど意味のない結論しか得られないという問題がある。あいまい性の表現手法には、通常、確率論を始め、ファジィ理論、ベイズ確率論、Depster & Shafer の理論など様々な方法が提案されており、いずれの方法でもルールのあいまい性をうまくチューニングできれば、推定結果のあいまい性を減少させることができる。

しかし、現実には、多数のルールのあいまい性のチューニングを行うのは非常に難しく、ルール数が増大するにつれて、この手間は爆発的に増大するという問題がある。筆者らの試作例では、数10個のルールからなるエキスパートシステムでも、チューニングを行うことが非常に難しくなった。このエキスパートシステムにおいては、複雑さやあいまいさを生じさせている原因の多くが植物の生理的な挙動に関わるものであった。

人工知能研究がフレーム問題やあいまい性のチューニングの問題の壁に直面し始めたとき、ニューラルネットがこれらの問題を解決する新たな手法として登場した。エキスパートシステムとニューラルネットを組み合わせたコネクショニスト・エキスパートシステム¹⁹⁾が開発され、学習によるルールのチューニングの自動化が図られた。また、農林水産業分野において、多数の要素が複雑に絡み合う農業生産地域の総合評価問題などにニューラルネットがいち早く適用され、事例データに基づく学習によって知識を獲得したニューラルネットにより、良い結果が得られた²⁰⁾。

このような経緯から、生物に関する複雑かつあいまいな問題ではニューラルネットが有効であると考えられた。そこで、第2章では、人工知能的な方法とニューラルネットを組み合わせさせて植物生長の形態変化のモデリング手法を開発した。

ニューラルネットのアルゴリズムや現実の神経回路網を単純化してモデル化するプロセスには、植物生長のモデル化において参考とすべき優れたものがある。第3章では、植物の構成要素である器官に着目し、多数の器官から構成されたシステムとしてモデルを構築した。また、ニューラルネットと同様、このモデルを観測データからシステム同定する手法を開発した。

1. 2 モデリングにおける考え方

1. 2. 1 基礎的研究としての目的

植物は、器官-組織-細胞の各階層において、各構成要素が環境変化の影響を受けながら自己組織化が行われる非常に複雑なシステムである。植物の生長は、非常に多くの非線形要素が結合した複雑なシステム（複雑系）の時間的変化と見ることができ、線形システムの場合、モデル構築の際に影響の小さい要素を無視することは大きな誤差にはならないが、非線形システムでは些細な効果でも無視すれば時間発展の結果として非常に大きな誤差を生み出す可能性がある。しかも、ある効果が最終的にどの程度の影響をシステム全体に及ぼすことになるのかをアприオリに判断することは困難である。

第2章では、できるだけ多くの情報をモデルに反映させることを中心テーマとして取り組んだが、この接近方法で上記の問題をクリアできる保証は存在しない。そのため、これとは全く異なる接近方法についても考える必要がある。そこで、第3章ではできるだけ単純な数学的構造を持つモデルで複雑な現象を扱うという考え方に基づいて、植物の生長に見られる複雑な変化のモデル化を試みた。

これらのモデルでは、いずれも生長過程における複雑さの表現手法が重要なテーマになっている。複雑さの実体が何であるのかを調べるため、第4章では光合成速度、及び生長速度の環境変化に対する複雑な応答についての実験を行い、その複雑さの特徴についてカオス性の観点から解析した。

植物は生長に際して自己の形態を環境変化に応じて適応させるシステムであるが、環境に対する適応性を持つシステムは自己組織系や協同現象と呼ばれ、基礎的な研究はかなり古くから行われて来た^{21,22)}。しかし、これらのほとんどの研究では、対象を非常に単純化して定式化し、安定性などの数学的特性の解析、一般解の導出などを行っている。そのため、これらの方法を、植物の生長という具体的な現象に対して、そのまま適用することは難しい。一方、

生物学分野でのこれまでのモデル化手法は、前節でも述べたようにほとんどが古いタイプのモデルリング手法の枠組みの範囲に収まっている。

最近では、生物系（脳の機能、生物の自発的形成、免疫ネットワーク、多種の生態系ネットワークなど）や非生物系（流体、多モードレーザー、スピン系、固体物理の非線形現象など）、社会現象などに対する「複雑系の科学」の研究が活発に行われている²⁹⁾。3、4章においては、最近の複雑系の科学の理論や考え方を応用して植物の生長のモデル化や実験データの解析に取り組んだ。

後で詳しく議論するように、ある生物の機能を模倣して作られたアルゴリズムは、他の生物の現象にも適用できる場合がある。第5章では、植物の生長モデルを一般化して、他の対象に利用する可能性について考察した。さらに、この一般化モデルにマイクロ系（メソスコピック系）に特有の量子論的性質を表現する簡単なモデルを追加して、生物や生命の特徴を備えた生物モデルとする方法について考察した。

1. 2. 2 生物を模倣したアルゴリズム

前節で述べたように、生物のモデルを構築する際には、大規模プログラミングや複雑系のモデリングという大きな問題がある。これらの問題に対しては、生物のモデリングの研究から生み出された「生物を模倣したアルゴリズム」が役に立つと考えられる。ここで、「生物のモデリング」と「生物を模倣したアルゴリズム」には以下のような相互依存的な入れ子関係（再帰的關係）があり、これがモデリングや現象の解明に際して非常に大きな利点を備えていると思われる。

一般に、生物系は非線形性のある多数の部分系から構成された適応複雑系であり、部分系全体を協調的に機能させて自己組織化を行う上位機能があると考えられる。そのため、生物

系のモデリングでは部分系に関するモデルだけではなく、部分系間のダイナミクスに関する上位レベルのモデルが必要である。部分系間の関係が上位レベルのメカニズムで記述できる場合、部分系のモデルを非常にラフにして部分系間のモデルを作ることができる。

このような方法によるモデル化の例としては、ニューラルネット²⁴⁾（人工神経回路網）と遺伝的アルゴリズム^{25,26)}（GA: Genetic Algorithm）がある。これらは、システムを構成する部分系（ニューロンや種）の機能を単純化し、システム全体を支配するルール（学習や適応）をプログラムで表現する。これらは、最適化問題全般に適用できる汎用アルゴリズムであると同時に、全体の機能（脳の機能や生物進化）をコンピュータ上の計算過程として表現するモデル（計算モデル）でもある。

ニューラルネットやGAには、

①サブ・モデルをできる限り単純化する

②対象システムの複雑さは、サブ・モデルの数を増やすことで表現する

という特徴があり、これは複雑な系のモデルを構築する際に不可欠な要素であると考えられる。たとえば、ニューラルネットではニューロンのモデルをシグモイド関数（S字型関数）で表された入出力特性を持つユニットで表している（実際には、ニューロンはマイクロプロセッサ以上の複雑な特性を有しているといわれる）。全体構造も、ニューロンが階層のあるいは相互結合するという単純なものである。このモデルでは、実際の神経回路網が有する複雑さは、ニューロンの数、及びシナプス結合（ニューロン間で信号を伝える経路の重み係数）などのパラメータの多さで表現されている。

ニューラルネットとGAでは、さらにシステム同定（未知パラメータの推定）にも生物の「学習」や「進化」の機能を模倣したアルゴリズムを使うという接近方法が採られている。すなわち、

③生物の機能を模倣したアルゴリズムを使って生物のモデルを同定する

という特徴がある(③はいわば「毒をもって毒を制する」接近方法である)。

第3章では、①、②、③の接近方法に沿って、器官生長の単純なモデルを用いて器官間の複雑な相互関係のモデリングを行っている。

1. 2. 3 複雑なシステムの再帰的理解

モデリング手法とアルゴリズムの間の再帰的關係は、複雑なシステムを理解する方法にも利用することができる。

我々が無理なく理解できる上限は、数学の定理の証明であれば数ページまで、プログラムであれば数百行がほぼ限界と言われている。これ以上に複雑な定理やプログラムでは、問題の分割と構造化によって対象を部分的に理解し、個々の理解を積み重ねることによって全体を「論理的に理解」せざるを得ない(数学やプログラミング言語では、論理的に厳密な論理体系があるため、このようなことが可能である訳である)。

しかし、非線形な特性を持つ複雑なシステムでは、個々の部分をすべてサブ・システム(部分系)に分割したとしても、全体を「理解」できるとは限らない。線形システムであれば理解した部分に比例して、全体を理解できるが(回帰分析という寄与度)、非線形なシステムではすべてのサブ・システムを個別に理解したとしても、システム全体を理解したことにはならない。

秩序を協調的に生成する複雑系では、個々の要素の「理解」を積み重ねるよりも、むしろシステム全体を支配する法則(ルール)が重要であり、本質的である。すなわち、線形システムのモデル化では、考慮する要素や要素間のメカニズムを詳細にモデルに組み込むことが全体を理解する効率的な方法となり、複雑系のモデル化では、システム全体のラフなモデルを最初に作り、そのモデルを実際の現象に近づける方法が効率的である。

これを実現するためには、ラフな記述が可能であると同時に観測された情報からシステム同定が可能であり、かついくらかでも精密化が可能なポテンシャルを備えた表現手法（数学的な枠組み）が必要である。たとえば、ニューラルネットには、隠れユニットを増やせば任意の一価連続関数が近似できるという特性を備えている^{27,28)}。植物生長のモデルでこのような特性を備えたモデルは、どのように作れば良いのだろうか？（第3章では、このようなモデルの構築に取り組んだ。）

ちなみに、適応複雑系のモデルの多くは非線形システムの最適化問題に帰着されるため（たとえばニューラルネットにおける記憶や学習は誤差関数やエネルギー関数の最小化、GAは適応度の最大化）、ある分野で開発された手法がそのまま他の分野でも利用できることが多い。すなわち、複雑適応系のモデル化において得られた研究成果は、モデル化の際の対象を超えて非常に広い用途が期待できる。そのため、従来のモデル化の接近方法と比べると大きな波及効果が期待でき、さらに最適化手法の観点からは、生命現象の本質に関する横断的な議論を行うことができる^{29,30,31)}。

また、ニューラルネットのシステム同定に GA を組み合わせるということも可能であり、実際に行われている^{32,33,34)}。生物を模倣したアルゴリズムには、「役に立つ手法は何でも利用する」という工学的精神を発揮しやすいというメリットがある。

1. 2. 4 データに関する情報理論的な問題

一般に複雑なモデルほど、構築するためには多くの情報量が必要である。このため、環境変化に対する植物の挙動の複雑さを解析するためには非常に大量の情報が必要であると予想される。大量の情報を得るためには、実験の徹底的な自動化と高速化、そして一回の実験で得られる情報量の最大化などを図る必要がある。植物生長を対象とした実験では一回の実験

に長い時間がかかるため、多数のサンプルを得るには複数の植物個体を用意して並列的に実験を行うのが普通である。ところが植物には個体差があるため、2つの異なる個体間のデータだけでは厳密な比較ができない。そのため通常は、平均値や分散値などを用いて統計的な比較が行われているが、そうすると今度は1サンプルの価値（含まれる情報量）が極端に小さくなってしまふ。

一般に、実験において効率よく情報を収集するためには、1回の試行で得られるサンプルに含まれる「正味の情報量」をできるだけ多くする必要がある。1サンプルに含まれる「正味の情報量」は情報理論における相互情報量 $I(a; b)$ として次式で定義される³⁵⁾。

$$I(a; b) = -\log P(a) + \log P(a|b) \quad (1-1)$$

ここで、 $P(a)$ は事象 a が発生する確率、 $P(a|b)$ は事象 b が起きたときに事象 a が発生する確率である。また、 $-\log P(a)$ は事象 a が有する情報量、 $-\log P(a|b)$ は事象 b から事象 a が発生する予測の情報量である。事象 b をモデルの予測値、事象 a を実験による観測値とすると、滅多に起こらない現象が予測に反して起こる実験ほど、多くの相互情報量が獲得されることになる。

ちなみに (1-1) 式では、完全にランダムな現象が最も多くの情報を持つことになるが、これは「完全にランダムな情報は、それ以上圧縮できない」という意味であり、ランダム・データが実験者にとって価値があるかどうかとは無関係である。観察された現象がランダム現象でない場合には、予想できない現象（一見ランダムに見える決定論的な現象）ほど大きな情報量を持っている（我々にとって価値がある）。このことから、実験で得た情報を使ってモデルを修正し、修正されたモデルの予測結果と大きく異なる実験を集中的に行うようにすれば、多くの有益な情報と高い予測精度を持ったモデルが自動的に得られると考えられる。

以上のような考え方から、第4章では、1個体の植物を使って、できるだけ多くの情報量を持つ測定サンプルを得るための実験を行った。

1. 2. 5 人工生命研究との関係

人工生命の研究³⁶⁾では、人工知能の研究と同様、非常に単純化した生物・生命現象をできるだけ簡単なコンピュータプログラムで記述することが試みられている。また、「生命と非生命の定義」や「生命の基本的メカニズム」を論理的な枠組みで扱う高度に抽象化された理論的研究も行われている^{37,38,39)}。これらのアプローチは、生物の現象面の複雑さを直接扱わないという点でユニークであり、計算機実験によって生命の本質に迫る一つの可能性を持っている。計算機実験に用いられている生物モデルは、単純な論理で動作するプログラムに生命的な特徴を持たせようとするものである。

人工生命研究が出現する以前では、生物モデルの研究は測定データとの一致や生理的メカニズムとの関係が主な興味の対象であり、コンピュータプログラムとしてのアルゴリズムなどに関する研究はほとんど行われていなかった。これに対して人工生命の研究では、どのようなアルゴリズムが生物的な現象を生み出かということに大きな関心がある。

しかし、人工生命研究では、現実の生物から得られた生の測定データに対する関心は希薄である⁴⁰⁾。人工生命研究は、人工知能研究における「強い人工知能」と「弱い人工知能」に対応して、「強い人工生命 (Strong AL)」と「弱い人工生命 (Weak AL)」に分けられる。強い人工生命は文字通り生命を持つプログラムであり、生命現象の解明に迫ることが目的である。弱い人工生命は、人工的な仮想空間の中で生命に似た振る舞いをするプログラムを生み出すことが目標である。弱い人工生命にはコンピュータ・グラフィックなどの実用的用途があるため、最近の人工生命研究ではこちらが圧倒的に多い⁴¹⁾。

第2, 3章におけるモデリングの目標は弱い人工生命と逆のアプローチであり、現実の植物に関するデータなどの情報をできるだけモデルに反映させることで、リアリティーを持たせることを目指している。そして、モデル内部に測定データや既知の知識を集積して、必要

な時（予測など）に取り出すことができるようにすることを目的としている。

第5章では、第3章で提案したモデルを生命のモデルへと拡張する方法について考察したが、これは強い人工生命の立場にある。

2 形態的変化のモデル化

ここでは植物生長のモデル化の出発点として、形態変化のモデル化を行う⁴⁾。ここでの目的は、①植物生長に係わる種々の情報（観測データや知見）をできるだけ活用して大規模モデルを構築する方法を開発すること、②植物生長に関するデータベースあるいは知識ベースシステムとして様々な角度から情報を引き出せるようにすること、③実際の現象に合うように自己修正する機能をモデルに持たせること、などである。

2. 1 形態モデルの問題点

植物の枝や根は非常に複雑な構造を持っているが、空間的構造には自己相似性などの規則性がある。植物以外の生物にも同様の規則性が見られ、Lindenmayer は書き換え規則（プロダクションルール）で生物の形態を表現する方法を提案した⁴³⁾。この方法は提案者の頭文字をとってLシステムと呼ばれている。Lシステムでは、形態の空間的構造をリスト（記号列）、形態変化をリストの書き換え規則で記述する。

Lシステムは再帰的プログラミングとの相性が良く、C 言語や Lisp などのスタックを有する言語処理系を用いると容易にプログラム化することができる。そのため、CG（コンピュータグラフィックス）では植物のリアルなイメージを生成するのに良く利用されている⁴⁴⁾。また記号論理的な側面から、Lシステムは人工知能を構成する諸理論（プロダクション・システム、オートマトン理論、Chomski の生成文法など）と共通点があり、Lシステムのルール群は形態に関する知識ベースとしても利用できる。

このようなLシステムの特徴から、ここではLシステムを植物生長における形態変化モデルの中核として用いることにする。

現実の植物には個体ごとに「ばらつき」がある。このばらつきを表現する方法として、Lシステムにランダムなゆらぎを加えた確率Lシステムが提案されている⁴⁵⁾。渋沢は根系の土壌密度が根の伸張に与える影響をメタルール（「Lシステムのルール」を制御するルール）として組み込んだ確率Lシステムで根の複雑な変化の再現を試みている^{46,47)}。

しかし、植物の形態変化は環境への適応の結果であり、植物個体ごとの形態の「ばらつき」の本質が確率的なランダム過程であるとは考え難い。確率LシステムはばらつきのあるCG画像の生成を行うための便宜的な方法と考えられる。

形態に、一見、ランダムに見えるばらつきがある一方で、植物は全体の形態を環境に適応させるために全体としてはあたかも高度な知能をもっているかのように振る舞っている。こ

2 形態的変化のモデル化

の面に着目すると、植物生長の行動としての合目的性は何らかの最適化問題として定式化が可能である。たとえば Sakai は、限られた生長資源の下で植物は栽植密度と個体間の距離に応じて受光量を最大化するとしてモデルを作って双子葉植物の形態変化を説明している⁴⁵⁾。

ただし、植物の最適戦略が人間にとって必ずしも理解しやすいものであるとは限らず、形態を決定するルールが多変数の非線形関数となって非常に複雑な場合や、単純なルールでも適用結果がカオスになる場合には、形態を決定するルールを見つけだすことは非常に難しい。たとえ合目的なある決定論的ルールによって形態が決定されていたとしても、その適用結果が極端に複雑であれば、あたかもランダムに変化して見えることになる。

もともと形態変化に関わる生理的なメカニズムは、葉齢や環境に影響する複雑なものであるため、そのルールの生成にはその複雑さに対応した多くの情報が必要であることが予想される（ルールが単純である場合でも、「単純」であることを検証するためには、ある程度のデータが必要となる。具体的には「4 生長の複雑さに関する実験と予測モデル」を参照のこと。）。

しかし、我々が通常、入手可能な情報は決して十分ではなく、データ形式がばらばらであったり、欠測や測定誤差などによるあいまい性が伴うなどの問題がある。また、新たに多少異なった実験条件で得られた新しい情報が追加されることもある。モデルを作る際に一貫性がある正確なデータが大量に得られないという現実の下では、あいまいな情報や整合性のない情報でも、できるだけ無駄にすることなく利用する必要がある。

そこで以下では、このような情報を用いてモデルを構築するため、環境に依存する形態変化を表現するルールをシステムに組み込む方法を考える。

2. 2 形態モデルのアルゴリズム

ここでは、Lシステムで表された植物の形態モデルを対象にして、植物生長に与える環境変化の影響をモデル上で統合化するために必要なアルゴリズムを開発する。

まず植物の形態をLシステムのルールとして表し、Lシステムのルール、及びそれを制御するメタルールの表現法と構築法について考える⁴⁹⁾。具体的には、数値データや論理的関係を記述する知識など種々の形態で存在し、さらにあいまい性をも伴った情報を、ニューラルネットと人工知能的技法を使って統合化する手法を開発し、この手法を用いて植物生長における形態変化モデルを開発する。

2. 2. 1 Lシステム

Lシステムを用いると、ある構造を持った形態を記号列で表現し、記号をプロダクションルールで書き換えることで形態変化を表現することができる。

植物は種や品種の違いによって生育時の形態変化パターンが大きく異なり、Lシステムを用いたモデルでは、それぞれ異なったルールを用いて表現する必要がある。たとえば、開花によって栄養生長の状態から生殖生長の状態に劇的に変化し、新たな器官形成が行われなくなる種がある。このような違いをすべてLシステムのルールで表そうとすると、ルールが非常に複雑になる。また、同一の種や品種内に見られる構造の多様性は、生理的要因や環境要因によってLシステムのルールが大きく変化することを示しており、このような違いを表現するためには、さらに膨大な数のルールが必要となってしまう。

このように、Lシステムのルールは、種の性質、時間、環境条件によって左右されるが、このような多数のルールの制御を行う方法としては、それぞれのルールが利用される（発火する）頻度を確率変数によって表現する方法^{50,51)}（Stochastic L-system）や、器官間の養分や植

2 形態的変化のモデル化

物ホルモンの移動を考慮してルールの選択を周囲の状況に依存させる方法 (Context-sensitive L-system) が提案されている^{52,53)}。

形態変化は、ホメオティック遺伝子による制御、及び光形態形成にみられるように光感応性イオンチャンネルやフィトクロム等が環境から受ける反応に基づいている。したがって、Lシステムのルールの発火制御を行う過程は、本質的には確率事象ではなく、複雑さ(あいまいさ)の伴った決定論的現象であると考えられ、ルールの制御を確率的に行うことは不適當である(もちろん、CGにおいて、多様な形態の画像を生成するという目的であれば有用である)。また、生長の状況に応じてルールを制御するのは、実際には、その制御ルール(メタルール)を決定することが重要な問題となって来る。

そこで、Lシステムのルールの発火を、ファジィ知識を埋め込んだニューラルネットで表されたメタルールで制御するようにした。メタルールにニューラルネットを利用することで現実の事例を学習させることができるようになり、事例データの蓄積に従って、少しずつモデルを現実近づけることが可能となる。

2. 2. 2 Lシステムを使ったタイズのモデル

以下では議論を具体的に進めるために、タイズのLシステムを考えることにする。

節をA,B,D,E, 莖をS, 葉をLで表すと、分枝のパターンとして、たとえば図2-1のルールグループ a, a' のようなルールが得られる。図2-1のルールは、圃場及びポットを用いて様々な環境条件下で栽培したタイズ(品種: エンレイ)の形態を調べ、それらの形態を生成するのに必要と考えられたルールである。

ここで、()で囲まれた式は左側に分枝した状態を表し、[]で囲まれた式は右側に分枝した状態を表すと定義する。ただし、タイズの分枝や出葉は90°間隔で互生であり、方向は

2 形態的変化のモデル化

4状態ある。そのため、分枝方向を表すには4種の括弧記号が必要である。しかし、90°周期が正確に守られていると仮定すれば、括弧記号を区別しなくても括弧の数からこの4種の状態は自動的に決定できる。

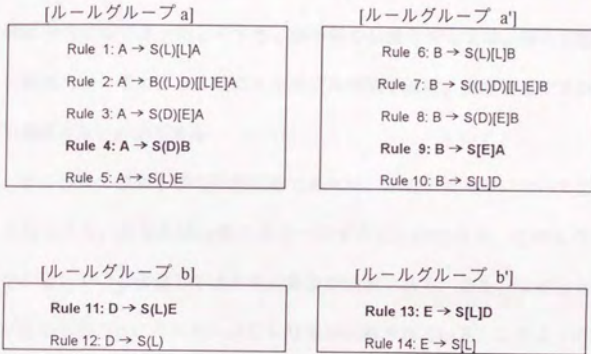
ここでは、式の見やすさと2次元での図形表示を容易にするために、左右の分岐を示すカッコだけを用いることにする。このとき、Lシステムの式または2次元の図形表示では、1節離れた分枝は直交していると解釈する。

さて、この記号を使ってダイズの形態を記号化しようとするとき、図2-1のルール(Rule 1~10)に示すように、1つの状態に対して複数の分枝ルールが必要となる。

ダイズの分枝は生育の前半で停止し、残りの期間は出葉だけである。即ち、分枝が停止し出葉のみの場合(図2-1のルールグループb, b')と、分枝がさらに継続する2つの場合がある(図2-1のルールグループa, a')。ルールグループa, a'からb, b'への遷移は、a, a', b, b'内でのルールの選択によって決定される。

ルールグループa, b (a', b')の中ではルールはそれぞれ一つだけが選択されるが、これは生長資源(主に光合成が活発かどうかなどの生理的要因)に依存している。たとえば、強光下で光合成が盛んに行われる条件下では分枝が活発になる。弱光下では分枝が少なくなり極端な場合には蔓化する。また、出葉周期は播種期、葉齢、気温などによっても変化する。

このような複雑なメカニズムを、Lシステムのルールだけで表すことは難しい。そこで、どのルールが適用されるかをさらに上位のルール(メタルール)で決定することにする。このメタルールは様々な要因が非線形的に影響を与える非常に複雑なものになるため、次に提案するファジィ・ニューラルネットを用いて構築する。



↓ メタルール

[メタルールによって選択されたルールセット]

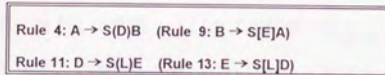


図2-1 メタルールによるLシステムのルールセットの選択

2. 2. 3 ファジィ・ニューラルネット

個別の品種に関して、分枝がどのような条件でどのように起こるかという詳細なデータが手に入るということはほとんど期待できない。また、例えそのようなデータを実験によって得て、特殊なモデルを作ったとしても、他の種や品種に対しては、同じ実験を再び行う必要がある。現状ではこのような方法によるモデル構築手法は、実験に要するコストの点でほとんど実用価値のないものである。

他方、ある程度の経験を積んだ栽培者であれば、ある環境条件においてどのような形態に生長して行くかを、かなりはっきりイメージすることができる。このようなイメージは、データが不足している状況下では非常に重要な情報である。また、文献としては、断片的ではあるが種や品種ごとにそれぞれ有益な情報が記載されている。このように比較的簡単に入手可能な情報だけでモデルを構築できれば、モデルの実用価値（対費用効果）を高くすることができる。

ここでは、ニューラルネットを用いてメタルールを作るため、ノイズの含まれる数値データやあいまいな知識情報を総合的に利用する手法を考える。

ニューラルネットは、入力パターンと出力パターンからなるデータセットが与えられると、それを一つのパターンとして学習し、シナプス荷重の値として分散的に記憶する。未知の入力パターンに対しては、記憶したパターンに基づいた内挿的な出力パターンを出力する。記憶するパターンにあいまい性がある場合、あいまい性に応じて学習する強度を変化させれば、あいまい性の大きい情報とあいまい性の小さい情報を共に組み合わせて活用できるだろう。このような学習は、次のような方法で行うことができる。

ある特定の条件 x_0 における関数の値 $f(x_0)$ が、ある知識から g_0 として得られたとしよう。この x_0, g_0 の値にあいまい性がない場合、

$$\{g_0 - f(x_0)\}^2 \equiv 0 \quad (2-1)$$

2 形態的変化のモデル化

となる。これは、ニューラルネットの入力層のユニットの値が x_0 のとき、出力ユニットの値が必ず g_0 になるように制約条件となる。すなわち、入出力層のユニットを座標軸とした位相空間中において、 $f(x)$ がこの点を通るように拘束されることを示す。

x_0, g_0 の値にあいまい性がある場合は、広がりのある領域を「なるべく通る」ように要請する緩い条件として考えることができる。この広がりが確率分布で与えられると利用しやすいが、多くの場合、これは主観によって示されるあいまいな情報である。この情報をここではファジィ理論において定義されるメンバーシップ関数で表現することにする。

入力層のユニットが n_{in} 個、出力層のユニットが n_{out} 個あるとし、入力層における入力値が $x=(x_1, x_2, \dots, x_{n_{in}})$ のときの出力ベクトルを $y=(y_1, y_2, \dots, y_{n_{out}})$ 、教師ベクトルを $T=(t_1, t_2, \dots, t_{n_{out}})$ とする。

学習サンプルは入力ベクトル x と教師ベクトル T の組から構成されるが、あいまい性は入力ベクトル x と教師ベクトル T の両者において考えられる。ところが、入力ベクトルのあいまいさは、あいまいさのない入力ベクトルを持つ複数の学習サンプルによって代替できる。なぜならば、教師ベクトルが同じ T で入力ベクトルが多少異なる x, x', x'', \dots のような学習サンプルが複数個存在すれば入力ベクトルのあいまい性が表現されるからである。したがって、入力ベクトルにはあいまい性がないものとして扱うことができる。そこで、以下では入力ベクトルにはあいまい性がないものとして扱う。

このような学習サンプルを使って学習するため、バックプロパゲーション (Back Propagation: BP) ・アルゴリズムを以下のように拡張する。

方法1

学習サンプルが m 個あるとき、 i 番目の学習サンプルにおいて、出力層のユニット j の教師値が t_j ($0 \leq t_j \leq 1$) である可能性をメンバーシップ関数 $\lambda_j(t_j)$ で表すとす

$(0 \leq \lambda_y(t_j) \leq 1)$. このとき, Zadeh が提案したファジィ事象の確率の定義に従うと⁵⁰, 出力層における2乗平均誤差 E は, 2乗誤差を $\lambda_y(t_j)$ で重みづけして,

$$E = \sum_{i=1}^m \sum_{j=1}^{n_{out}} \frac{1}{N_{ij}} \int_0^1 \lambda_{ij}(t_j) (t_j - y_j)^2 dt_j \quad (2-2)$$

として定義される. ここで, N_{ij} は, ファジィ集合の大ききさで,

$$N_{ij} = \int_0^1 \lambda_{ij}(t_j) dt_j \quad (2-3)$$

である ($N_{ij} = 0$ の場合を除く) . そして, この E を BP アルゴリズムで最小化する.

(2-3) 式の中にある積分計算は, BP による学習の1ステップごとに行う必要があり, 学習時間を長くする要因の一つとなる.

ところが, 主観的なデータの場合, 学習サンプル1個ごとのあいまいさは指定できても, 各出力ユニットごとに教師値のあいまいさを細かく指定することは困難であり, メンバシップ関数を含む (2-3) 式を厳密に計算する意義はほとんど認められない. しかも, 実際には, メンバシップ関数の形の微妙な違いが有する情報量は小さい. この違いを検出できるほどの観測データが得られる場合には, ファジィ理論的方法を適用する必要性は少なくなってくる.

そこで, ここでは十分な観測データがない場合を想定する. この場合, メンバシップ関数の形状を, ある程度, 簡略化しても, 大きな誤差は生じないと考えられる. 具体的には, 方法 I を以下のように簡略化する.

方法 II

サンプル i の教師値 t_j が提示されたときの2乗誤差 e_i は,

$$e_i = \sum_{j=1}^{n_{out}} (t_j - y_j)^2 \quad (2-4)$$

である。ここで、サンプル i 全体の正しさをメンバーシップ関数 $\xi_i(e_i)$ で表す（ただし、 $0 \leq \xi_i(e_i) \leq 1$ ）。観測データの回帰式に対する 2 乗誤差の分布に関する正確な事前情報が得られないとして、このメンバーシップ関数の形状を矩形とすると、

$$\begin{cases} \xi_i(e_i) = 1 & (e_i \leq d_i) \\ \xi_i(e_i) = 0 & (e_i > d_i) \end{cases} \quad (2-5)$$

である。ここで、 d_i は矩形のメンバーシップ関数の幅である。このとき、2 乗平均誤差 E を、

$$E = \sum_{j=1}^{n_{\text{out}}} \{1 - \xi_i(e_i)\} e_i \quad (2-6)$$

と定義する。この定義に基づいた BP による学習は、

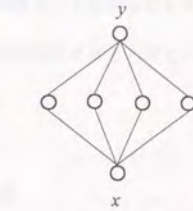
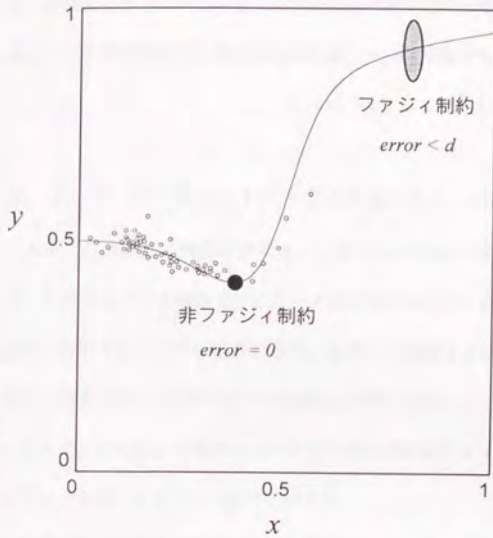
① $e_i > d_i \rightarrow$ シナプス荷重を修正

② $e_i \leq d_i \rightarrow$ 何もしない

となる。

方法 II による学習は、方法 I よりもはるかに簡便かつ高速である。方法 II のアルゴリズムでは、 d_i が大きいほど緩い制約になり、 d_i が 0 であると回帰曲線（超曲面）が必ずサンプル i の示す点を通過しなければならない強い制約になる。

観測されたデータ量が不十分である場合、先験的な知識から絶対通過すべき点と、あいまいな知識から通過することが望ましい範囲が与えられていれば、この学習アルゴリズムでは、たとえば図 2-2 に示すような回帰曲線が得られることになる。図 2-2 は、観測されたデータ (x_j, y_j) が取りうる変域のうちある範囲に偏って得られている場合を示している。



左図の滑らかな関数を生成するニューラルネット

図 2-2 ファジィ制約として利用されるあいまいなデータ (概念図)

図 2-2 の例では $x > 0.7$ の領域で観測データが全く得られてないため、この領域における非線形回帰式の予測精度は非常に低い。ここでは、縦長の楕円で示した領域を通過するというあいまいな知識（ファジィ制約）が得られているため、この情報によって図で示したような $y=1$ に漸近する回帰曲線になり、 $x > 0.7$ における回帰式の予測精度を向上させることができると考えられる。この場合、もし観測データだけから非線形回帰式を作ると下に凸の曲線が求められ、図 2-2 の回帰曲線とはかなり異なった曲線となるだろう。逆に、図中に●で示すような点を必ず通るという知識（非ファジィ制約）が得られたとすると、●の周辺の観測データは少なくとも良い。さらに、 $x=0$ 付近でのファジィ制約が与えられれば、観測データが全く無くても図 2-2 の回帰曲線に近い曲線を得ることができるはずである。

このように、ファジィ・ニューラルネットの学習アルゴリズム（方法 I または方法 II）に

より観測データとあいまいな知識を組み合わせて回帰式の予測精度を向上させることができ、観測データが全く得られない場合でもあいまいな知識から回帰式を推定できると考えられる。

2. 2. 4 マクロネットワークによるメタルールの構築

ダイズの形態は品種間差が大きく、メタルールは品種によって違ったものになる。メタルールを作るのには品種ごとにデータが必要であるが、実験によってそのデータを集めるのは長い年月を要するため困難である。しかも、文献などから得られる情報は断片的なものであり、特定の品種に関するデータとなると僅かである。

そこで、あいまいな情報を元にダイズの生長に関するメタルールを、ファジィ・ニューラルネットワークを用いて以下の方法で作成する。

生長に変化を及ぼす環境要因が m 種あるとき、これをベクトル $x = (x_1, x_2, \dots, x_m)$ で表す。また、環境要因によって、 L システムのルールが n 通りに変化するとき、ルールが発現した状態を 1、発現していない状態を 0 とし、ルールの状態ベクトルを $r = (r_1, r_2, \dots, r_n)$ で表すことにする。

メタルールは、環境要因から、どのルールが発現するかを決定するものである。すなわち、メタルールは、 m 次元の環境要因ベクトル x から n 次元のルールの状態ベクトル r を推定する関数である。このメタルールは、入力ユニットが m 個、出力ユニットが n 個のファジィ・ニューラルネットワークで実現できる。

しかし、 m が多いと学習に用いる学習用サンプルが膨大な数になる。たとえば、各入力ユニットに $\{0, 0.25, 0.5, 0.75, 1\}$ の 5 つの値のうちから一つの値を選んで、その値を入力ユニットの一つに与えるとしたときの組み合わせは 5^m 通りである。 $m=5$ とすれば 3,125 個もの教師パターンが必要であり、この程度の規模のニューラルネットワークでさえも必要なデータを

2 形態的変化のモデル化

全て集めるのは難しいだろう。また、メタルールの内部には比較的明確な論理をいくつか含んでいる場合があり、これを他のデータと一緒にして単一のファジィ・ニューラルネットに学習させようとするのは、いたずらに学習を難しくするだけである。

そこで、メタルールをその論理的関係を元に分割し、それぞれを比較的小規模なファジィ・ニューラルネットのモジュール（以下、ニューラルネット・モジュールと呼ぶ。）の集合で表現することにする（図 2-3）。分割することで学習用サンプルは少なくすることができる。例えば、上記の例で示したファジィ・ニューラルネットを入力ユニットが 3 個と 2 個の 2 つのファジィ・ニューラルネットに分割できれば、学習に必要なデータのサンプル数は $150 (=5^3 + 5^2)$ 個で済むことになり、必要なサンプル数が約 1/20 にまで削減される。また、分割することで、たまたま入手できた断片的な情報を使うことが容易となる。

しかし、ニューラルネット・モジュールが多数になると、ニューラルネット・モジュールの追加や削除、ニューラルネット・モジュール間の接続などの手続きが煩雑になるため、これをうまく管理する方法が必要である。

そこで、得られた情報の内容に応じてメタルールを自動生成する以下の方法を考案した。

得られる情報は定量的なデータや定性的知識、曖昧な情報など様々である。これらの情報を、それぞれファジィ・ニューラルネットに学習させ（ファジィ・ニューラルネットのフォーマットに変換）、以下の手順で組み合わせる。

- ① 知識情報や数値データなど種々のデータセットを学習によってニューラルネット上の記憶に変換する。
- ② ニューラルネット・モジュールの入出力ユニットにタグネーム（名称）を定義し、ニューラルネット・モジュールを 1 つのルール（または関数）として利用できるようにする。
- ③ ニューラルネット・モジュールの集合をデータベースとして管理する。
- ④ ニューラルネット・モジュールが接続してできるマクロ・ネットワークの入出力ユニットのタグネームを定義する。

⑤マクロ・ネットワークとニューラルネット・モジュールのタグネームの文字列パターンマッチングにより、マクロ・ネットワークの構築に必要なニューラルネット・モジュール（サブネットワーク）の自動接続を行う。

マクロ・ネットワークを構築する情報は、マクロ・ネットワークを構成するニューラルネット・モジュールの名称とマクロ・ネットワークの入手カユニットのタグネームだけで十分である。マクロ・ネットワークの自動構築、ニューラルネット・モジュールのデータベース管理、マクロ・ネットワークの実行を行うためのツール MacNet (Macro Network) を開発した (MacNet のソースコードを付録 A で示す)。

MacNet は、C 言語の一つの関数である。あらかじめ MacNet に model 文で指定された複数のニューラルネット・モジュールが記録されたファイルを読み込ませておく (このファイルのフォーマットは、ニューラルネットを作成するのに用いるニューラルネット・シミュレータに依存する。) 、次に MacNet が呼び出されたときには、MacNet は仮想的に一つのマクロ・ネットワークとして振る舞う。MacNet の内部では、ニューラルネット・モジュールの入出力ユニットのタグネームの文字列パターンマッチングを行い、ニューラルネット・モジュールの論理的接続を前向き推論によって見つけ出す。そして、この論理的接続に従って、マクロ・ネットワークの入力ユニットの入力値を使って各ニューラルネット・モジュールの計算を順次行う。マクロ・ネットワークの出力ユニットのすべての値が決定されるまで、これを繰り返し、計算結果をマクロ・ネットワークの出力ユニットの値として出力する。

MacNet の構造上、マクロ・ネットワークを構成するニューラルネット・モジュールに、他のマクロ・ネットワークや自分自身を利用するように拡張するのは容易である。また、ニューラルネット・モジュールの代わりに、通常のプログラム (関数) を実行するように拡張することも容易である。このような拡張を行った MacNet は、ニューラルネットを基礎とした汎用言語として利用できると思われる。

2. 2. 5 ダイズのメタルールの構築

図 2-1 で示した L システムのルールを制御するメタルールを、MacNet を使って作成した。

ダイズの形態は、栽植密度、日射量、気温、播種日などによって様々な形態をとる。栽植密度と日射量、気温などの要素と形態変化に関するデータは得難く、ここでは光合成速度が活発かどうかというあいまいな指標（0~1 のファジィ数で表す）でまとめて表現することにする。

播種日の違いによる変化は出葉周期の違いとして知られており、出葉周期は播種日と播種後経過日数（播種してから経過した日時）に依存している。このデータは実験的に比較的測定し易く、文献から集めることが可能である。そこで、播種日及び播種後経過日数と出葉周期の関係を、一つのニューラルネット・モジュール（cycle）で表すことにする。

出葉後、ある程度の日数が経過すると、出葉と分枝が起こる（以下、まとめて「分枝開始」と呼ぶ）。ある葉が発生してから経過した時間（以下、「発生後経過日数」と呼ぶ）が、出葉周期の日数を超えると次の葉が出葉することになる。発生後経過日数は植物体の各部分で異なっているため、各部分で独立して計算する必要がある。この計算はあいまい性のない単純な論理演算であるが、これもニューラルネット・モジュールで表現することができる。そこで、この論理的関係を一つのニューラルネット・モジュール（branch）で表すことにする。

十分な時間が経過して分枝発生が可能な条件が整ったとき、出葉と分枝が起こる。このとき、環境条件（ファジィ数で表された「光合成速度」の状態）と、その時点で主茎にある節数（以下、「主茎節数」と呼ぶ）に応じて、図 2-1 の L システムのルールが異なってくる。このルールに関する詳細な情報（データ）は入手できなかったため、以下の方法で主観的な知識情報を使うことにする。

光合成速度と主茎節数の数の関係を直感的に植物の生長状態の「勢い」として捉え、勢い

2 形態的変化のモデル化

に応じてLシステムのルールが選択されるものとして表現することにする（勢いが弱いと Rule_5, 勢いがやや強いと Rule_3,4, 勢いが強いと Rule_1,2 が選択される.）。この勢いに関するデータは経験的知識に基づいて作成し、これも一つのニューラルネット・モジュール（photo）で表すことにする。

各ニューラルネット・モジュールを文献^{55,56)}から得られた情報や経験的知識を使って学習させ、MacNetで結合した（図2-3, 表2-1）。このソースファイルを図2-4に示す。

これをメタルールとして実行させたファジィLシステムの結果の例を図2-5に示す。寄せ集めのデータや経験的知識を元に作成したにも係わらず、このモデルが生成した形態変化の様子は、弱光下において光合成速度が不活発な状態で生育したダイズと比較的強い光強度の下で生育したダイズの実際の形態（図2-6）をうまく再現することができている。このモデルでは節間長や葉のサイズ、分枝の角度などの定量的な形態は扱っていないが、形態の直感的特徴は、このモデルに含まれている情報でかなり表現されていると判断される。

図2-5の環境条件と図2-6の環境条件は多少異なっているため、完全に形態が一致していないのは当然であるが、もし環境条件が同じであってもメタルールが寄せ集めの情報から作られた不完全なものであるため完全に形態が一致するとは限らない。しかし、形態変化と環境条件の測定データが得られれば、メタルールを構成する各ニューラルネット・モジュールを再学習させることで修正することができる（ただし、形態変化を長期間測定するのは非常に多くの手間がかかるため、実際には難しい。この具体的な解決方法については、形態モデルの逆問題として「2.3.2 形態モデルの逆問題とその応用方法の考察」で考察する。）。

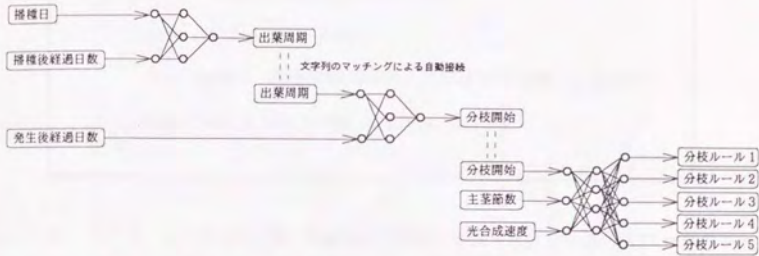


図2-3 タグネーム（入出力定義名）のマッチングに基づいた前向き推論による複数のニューラルネット・モジュールの自動接続

表2-1 各ニューラルネット・モジュールの入出力ユニットの定義

モジュール名	cycle	branch	photo
モジュールの入力タグネーム	播種日 播種後経過日数	出葉周期 出葉後経過日数	主莖節数 出葉後経過日数 分枝開始
モジュールの出力タグネーム	出葉周期	分枝開始	rule_1 rule_2 rule_3 rule_4 rule_5

```

/* An example of source list of MacNet */
model { cycle, branch, photo }
input { 播種日, 播種後経過日数, 出葉後経過日数, 主茎節数 }
output { rule_1, rule_2, rule_3, rule_4, rule_5 }
    
```

図2-4 マクロ・ネットワークを MacNet で記述した例（マクロ・ネットワークのソースコード）。

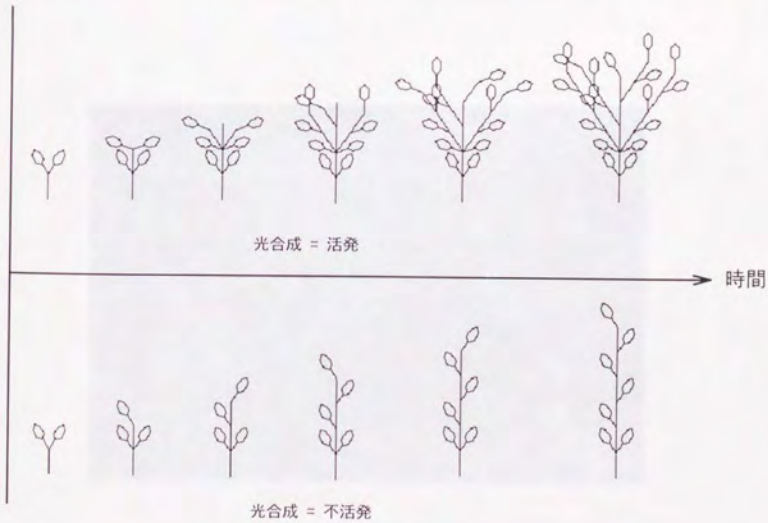


図2-5 形態モデルの実行例（ダイズ）

2 形態的変化のモデル化



(a) 光合成がやや活発な場合



(b) 光合成が不活発な場合

図 2-6 光環境の違いによるダイズの形態の違い

2.3 考察

2.3.1 プログラミング手法としての発展性

ここで開発したモデリング手法には、以下のような特長がある。

- ①あいまいな情報の活用と過去の情報の再利用が容易にできる。
- ②得られた情報量に応じた精度でモデルが構築できる。
- ③必要に応じて大規模かつ複雑なモデルが容易に構築できる。
- ④実測データによるモデルの修正が自動的にできる。

これらの特長はすべて、フォーマットがばらばらなファクトデータや、あいまいな経験的知識を、ニューラルネットのモジュールに変換し統合化したことに起因している。ニューラルネット・モジュールにしたことにより、データ構造に関する細かな情報は完全に隠蔽され、モデル構築を行うプログラマは、入出力の関係だけを考えれば良い。

MacNet で構築したマクロ・ネットワークは、大規模モデルの構築以外に、複数のデータベースを管理するデータベース管理システムとしての機能も持っている。データベースの選択は、タグネームのマッチングにより行われる。データベースに記録されたレコードの検索は、ニューラルネットの連想記憶機能を使って行うことができる。しかもこの検索では、不完全なキーの組み合わせやあいまいな数値の組み合わせからでも、必要なデータを引き出すことができるという利点がある。データベースへのレコードの追加や削除は、そのレコードに相当するデータを学習データファイルへ追加または削除した後、再学習によって行うことができる。

ニューラルネット・モジュールの代わりに既存のプログラムをマクロ・ネットワークに組み込むと（ただし、入出力変数をタグネームで定義しておく必要がある。）、マクロ・ネットワークにニューラルネットでは表現できない画面へのグラフィック出力や機械の制御など

2 形態的変化のモデル化

の機能を持たせることができる。マクロ・ネットワークに通信機能（TCP/IP や PPP など）を持ったプログラム組み込みば、マクロ・ネットワークの入力ユニットから与えたコマンドによってインターネットを経由し離れたワークステーションにあるプログラムを起動したり、そこで得た計算結果を持って来ることも可能である。この計算結果を、さらに他のニューラルネット・モジュールで加工し、マクロ・ネットワークの出力ユニットから出力するようにすれば、MacNet の上で非常に大規模な分散システムを構築することができると思われる。

しかし、非常に大規模なシステムの構築を行う場合、タグ・ネームの文字列パターン・マッチングだけで巨大なネットワークを作るとタグ・ネームの管理が煩雑になって来る。また、タグ・ネームの付け方にミスが発生しやすくなる。このような問題を避けるため、次のような方法が考えられる。

- ①タグ・ネームにスコープの制限を設けて、ある範囲のモジュールとだけマッチングを行うようにする。
- ②マッチングによって生じたモジュール間のリンクによって形成されたネットワークを一つのニューラルネットと考えて、正しい動作を教師パターンとして学習させることでリンクの重みを自己修正させる。
- ③マクロ・ネットワーク自体もモジュール化して、マクロ・ネットワーク内で使用できるようにする。

2. 3. 2 形態モデルの逆問題とその応用方法の考察

Lシステムは形態を表現するための言語体系の一種である。Lシステムが生物の発生モデルでしばしば利用されるのは、この言語が生物の発生の過程を表現しやすいというのが最大の理由である（Lisp や Prolog が人工知能研究において推論モデルの表現に適しているというのと同じ理由である）。Lシステムは、ある初期状態から複雑なパターンを自動的に作り出

2 形態的変化のモデル化

すという機能は有しているが、その逆に、実際の生物の形態変化から、その形態変化を生成するLシステムのルールを見つけだすことはできない（これは、形態パターンからルールを発見する逆問題である。）。もし、CCD カメラを使って長期間連続的に撮影した画像データから、Lシステムのモデルが自動的に生成可能になれば非常に便利である。そこで、本章で提案した形態モデルの発展方向として、この問題の解決手法について考察してみよう。

形態をLシステムの記号列に変換する問題は、自然言語処理において形態素解析により日本語文の句構造を見つけだすのと同じ問題である。そして、形態の変化からLシステムのルールを見つけだすのは、文章から文法を見つけ出す問題に相当する。

これを行う方法としては、機械学習的な方法が考えられる。しかし、画像として得られる形態パターンは複雑であり、ノイズが重畳している可能性もある。そのため、常に正しい記号列が得られる保証はない。形態を表現する記号列に誤りが混入している可能性がある場合、機械学習によって記号列の変化の事例データから、記号列変化を記述する変換規則（Lシステムのルール）を発見するのは非常に難しい問題になる。

もう一つの方法としては、目的とする形態を生成するようにLシステムのルールを適応的に変化させる方法が考えられる。

この方法は既に、「利己的遺伝子」の研究で有名な Dawkins が実現している³⁷⁾。彼は、Lシステムのルールセットを遺伝的アルゴリズムで最適化するプログラム *Blind Watchmaker* を開発した。このプログラムでは、自然淘汰と突然変異の進化メカニズムに「累積淘汰」という概念を導入することで、現実の生物界にみられるような進化が可能であることを実験的に示し³⁸⁾、アルファベットの世界パターンなど様々な形態を持った疑似生物 *Biomorph* を作り出している（ただし、このプログラムの目的は、上記で述べた「形態モデルの逆問題」を行うことではなく、トートロジー的な議論が多い進化論争³⁹⁾において計算機実験に基づく議論を行うことにある。）。

2 形態的変化のモデル化

Dawkins の方法と本章で提案したモデリング手法を組み合わせることで、植物の生長パターン事例（画像データベース）から、形態モデル（形態の記号列と形態変化を表現するLシステムのルール、及びメタルール）を自動的に生成できると考えられる。特に最近のマルチメディア技術を組み合わせると、これは意外と簡単に実現できると思われ、具体的には次のような方法が考えられる。

- ①圃場にビデオカメラを設置する。
- ②植物体を1[frame/hour]程度の速度で撮影し（微速度撮影）、動画ファイル（たとえば、Video for Windows や QuickTime）として記録する。
- ③各画像を2値化し、形態モデルが生成したパターンとの一致を適当な評価関数（相関係数など）で評価する。
- ④遺伝的アルゴリズムを用いて評価関数を最大にするLシステムのルールを探索する（Dawkinsの方法）。
- ⑤Lシステムのルールの変化を表現するメタルールを、ニューラルネットを用いて学習させる（本章の方法）。

①～⑤は、すべて市販のビデオカメラやパソコン、ビデオキャプチャ・ボードなどを用いて簡単に実現可能である。このようなセットを日本各地に設置すると、植物の生長に関する非常に重要なデータを自動的に得ることができると考えられる。生長パターンの動画像データをインターネットなどを介して共有すれば、複数の地域における栽培事例のファクトベース化や遠隔地に分散した農場のリモート管理などができるだろう。このようなデータがそれぞれ地域で大量に蓄積され、相互利用できるようになると、環境に対する植物の形態変化に関する新たな一般的知見を得ることができると思われる。

また、生長過程の動画像ファクトベースのデータ圧縮、形態モデルのモデルベース（モデルのデータベース）や生長パターンの検索などにも利用できるだろう。

3 量的変化のモデル化

前章では、植物生長における形態変化のモデルを扱ったが、本章では器官の量的な変化に対するモデル化を試みる。

前章では、「情報が不十分である」ということから、知識などの既存の情報を活用するモデル化手法を考えた。そこでのモデル化の考え方は、保有する情報をできるだけ活用して、複雑な構造を有するモデルを作ることにあった。この考え方とは逆に、単純な構造を持つモデルによって複雑な変化を表現できないだろうか？ また、測定がしやすいデータだけを用いて、モデルを自動的に生成できないだろうか？

本章では、簡単な数学的構造を持つ方程式を用いて、多様かつ複雑な変化をする植物の生長パターンの定量的に表現できるモデルの構築手法を開発する。また、器官の重量（または大きさ）の時系列データから、このモデルに含まれるパラメータをシステム同定する手法を開発する。

3. 1 従来のモデル化手法の問題点

3. 1. 1 コンピュータプログラムとしての必要条件

植物は環境変化に適応して、自らの形態をダイナミックに変えながら生長している。マクロ的に見ると、これは植物個体を構成する複数の器官の量的（空間的）時間変化として捉えられる。この変化は非常に複雑ではあるが、計算機モデルとしてうまく記述できれば特定の器官の生長を制御するための手段になると考えられる。これまでに植物の生長モデルは数多く提案されてきたが、植物個体の空間的変化と時間的変化を定量的に扱い、しかも全ての未知パラメータが観測データから推定可能なモデルとなると極めて少ない。

植物生長のモデリング手法としては、環境条件や生理的要因間の関係を方程式や論理式（以下、これらをまとめて知識と呼ぶ）で記述し、知識を増やすことで複雑な現象のモデルを構築するボトムアップ的な方法がある。この方法では知識の増大に伴うプログラム・コード量とプログラムの複雑さの爆発的増大という大規模ソフトウェア開発に共通の困難がある。この困難は人工知能やオブジェクト指向プログラミングなどの技法を導入することで緩和されるが、根本的解決策にはならない。また、モデルに含まれる未知パラメータの最小2乗推定は困難である。

一方、パラメータの最小2乗推定を行うためモデルを単純化し、線形または準線形システムで表現するトップダウン的な方法がある。環境条件と植物生長を扱うモデルではロジスティック重回帰^{60,61)}やべき級数展開多項式による方法⁶²⁾が試みられている。しかし、複雑な現象を記述するためには非常に多くのパラメータ（項）が必要になるという問題がある。

さて、流体系や非平衡化学反応系など多くの非線形力学系において、単純な支配方程式が非常に複雑な現象を生み出すことが知られている（線形系では複雑な現象が発生するには支配方程式も複雑でなければならない）。植物体を構成する多くの要素には非線形性があり、植物個体の多様かつ複雑な生長パターンも単純な構造の方程式で表すことができる可能性が

ある。そこで、ここでは植物生長パターンの複雑な変化をできるだけ単純な構造を持つ非線形方程式で記述することを試みる。次に方程式に含まれる未知パラメータを推定するアルゴリズムを提案する。そして小規模なモデルと観測データにより、これらを検証する。また本研究と同じ発想の下で開発された他の生物系におけるモデル（神経回路網モデル）と比較しながら、提案したモデルの意味について議論する。

3. 1. 2 回帰的モデルにおけるモデルの良否評価

非線形性のある対象をモデル化するには、モデルに付与する非線形性が重要となる。そのため、あらかじめ対象の特性が分かっている場合には任意の非線形関数が近似できる非線形回帰的なモデルが便利である。ニューラルネットは非線形回帰的なモデルであり、2乗誤差最小化の学習過程は非線形重回帰分析あるいは非線形主成分分析としての機能を有している^{53,64,65)}。

学習によって中間層のユニットに対象の有する特徴が非線形な主成分として獲得された場合には、学習に使用したサンプルとは異なったケースに対しても、そのニューラルネットは正しい推定・予測・認識などを行うことが可能である。たとえば、片山らは入出力層が同一で第3層のユニットが1つしかない5層のニューラルネットワークを用いて2次元の折れ曲がった分布や円形の分布を学習できることを示し、第3層のユニットに対象の一般的特徴が獲得されていることを確認している⁶⁶⁾。

しかし、ノイズと本来の非線形な特性を区別することが困難であるため、ノイズ成分まで学習してしまい識別率が低下することがある。これは「過剰学習」と呼ばれている。このような問題があるものの、生物的な現象に対してニューラルネットを適用するとうまく行くケースが多い（この理由の一つは、生物の生長曲線とニューラルネットのユニットの特性関数に用いられているシグモイド関数が全く同じ形の曲線であることにある。詳しくは、後

で考察する.)。

たとえば、森本らは、ニューラルネットを使った光合成速度のモデルを実験データから作り、最適条件の探索に利用している⁶⁷⁾(このような用途では、ニューラルネットに対象の特性が十分獲得されている必要がある)。Murase *et al.* はカルマンフィルタを用いたニューラルネットの学習アルゴリズムを開発し、これを使って植物生長モデルを作っている⁶⁸⁾。このモデルでは入力ユニットに時間、養液濃度、気温を与え、出力ユニットには長さと同重量が定義されている。実測データ(カイワレダイコン)を使って、予測精度の比較を行った結果、このモデルは重回帰モデルやバックプロパゲーション・アルゴリズムで学習したモデルよりも高い精度で予測できたという。

ニューラルネットでは、この他にも様々な学習アルゴリズムが提案されており、それらのほとんどは2乗誤差を最小化するものであるが、しばしばバックプロパゲーション・アルゴリズムよりも高い汎化能力を示すことが報告されている(ただし、たまたま良い結果が得られたケースだけが論文となって報告されるという「バイアス」がかかっている可能性があることは否定できない.)。

いずれにせよ、どの学習アルゴリズムでも、それが2乗誤差を最小化するものである場合には、学習終了時の2乗誤差が同じ大きさであれば同定されたモデルに本質的な違いはないはずである。予測精度の違いは学習アルゴリズムと適用した問題の相性により、たまたま良いモデルが生成される場合があるために生ずると考えられる。

汎化能力の本質的な違いはモデルの良否に起因している。ニューラルネットのような非線形モデルの場合、線形モデルの場合と異なって良否評価は非常に難しいが、AIC (Akaike's Information Criteria: 赤池の情報量基準) や MDL (Minimum Description Length)^{69,70)}などの統計的モデル選択基準の適用が試みられている。いずれも誤差(残差)の大きさとモデルの自由度の関数であり、誤差が小さいと同時に自由度も小さいことを良いモデルの条件としている。

すなわち、誤差が同じ大きさであれば、ニューラルネットを構成するユニットを少なくするほど良いモデルになると考えられている。

3. 1. 3 生長パターンの「ゆらぎ」

観測したデータのサンプル数が少ないと、AIC などのモデル評価基準で選んだ最適モデルは直線に近いめらかな曲線（曲面）を描く関数になる。線形回帰モデルでは、2つの変量間に直線的な比例関係があるという前提があり、直線から大きくずれたゆらぎ成分は全てノイズとして扱われる。そのため、環境条件の違いなどによって生じる生長パターンの変化もノイズとして扱われることになる。この生長パターンの違いが植物の適応的行動によって生み出されている場合、そのメカニズムがモデルに獲得される可能性は全くなってしまう。

複雑な変化をモデルで表現できるようにモデルの自由度を大きくすると、今度は非常に多くのデータを集める必要が生じてくる。自然環境下では、正確なデータを多数得るのはほとんど不可能である（「4 生長の複雑さに関する実験と予測モデル」では、人工環境下で精密な測定を試みる。）。このような場合、先験的な知見をモデルに加えることによって、少ないサンプルでも良いモデルを見つけたことはできないだろうか。

非線形系や相互作用のある3体以上の力学系ではカオス的な「ゆらぎ」が生じる場合がある。植物体を構成する器官には強い非線形性があり、構成要素も多数あるため、これが全体の変化を複雑にしている可能性がある（これは、従来の回帰的なモデルでは全く扱うことができない。）。

多数の微分方程式で記述されたシステムダイナミクス・モデルは、このようなゆらぎを表現できるが、植物の器官の間の定量的関係に関する知見は少ないため、ボトムアップ式にメカニスティックなモデルを作ることは困難である。既知の部分だけでモデルを作ったとして

3 量的変化のモデル化

も、今度はそのモデルに含まれる多数の未知パラメータをいかにチューニングするかが大きな問題となって来る。

以上の議論から、植物生長などの非線形回帰的モデルが具備すべき条件として以下のよう
に考えることができる。

①線形の回帰的モデルは直線関係しか扱えない非常に「堅い」モデルである。一方、ニューラルネットは任意の非線形関係が記述できるため、「柔らか」すぎる。アプリアリに決定された「堅い」非線形関係を備えると同時に、未知の要素に適應できる「柔らかさ」を持ったモデルが望ましい。

②カオス的な複雑さを扱うことができる必要がある。

③線形回帰モデルと同様、観測データだけから構築できる使いやすさと、解釈のしやすさを備えている必要がある。

そこで以下では、これらの条件を満たすモデル化手法を考える⁷⁾。

3. 2 器官間の干渉を考慮した生長モデル

3. 2. 1 多体系としてみた植物の生長変化

植物の生長は、器官の伸長、肥大及び新たな器官の発生によって行われる。発生時期や空間的位置、環境変化に対する応答特性などの違いから、それぞれの器官は異なったパターンで生長すると考えられる。また、植物体の内部では、器官の間で生長に必要な物質や生長に必要な空間など（以下、生長資源と呼ぶ）の配分によって相互に影響を受けあっている。すなわち、器官は生長資源を相互に奪い合う「競合」関係にある一方、根や葉の生長のように他器官の生長を促す「共生」関係が存在する。このとき、生長資源の配分量に対する器官生長の応答に非線形性があると、全体の挙動はかなり複雑になるはずである。

個体を構成する器官の量的生長パターンが異なっているのはなぜだろうか。もしマイクロレベルのゆらぎが拡大したものであるならば、モデリングではマイクロ系のゆらぎを主な対象として扱う必要が生じて来る。

他方、「3体問題」として良く知られているように、相互作用のある3体以上の力学系の運動はマクロレベルで非常に複雑になるという現象がある。植物でも個体を構成する細胞—組織—器官の各階層の要素間では相互に影響を及ぼし合っているため、マイクロレベルのゆらぎがなくても全体の変化は複雑になるはずである。しかし、各要素間の定量的関係に関する知見は少ないため要素間の個々のメカニズムからボトムアップ式にモデルを作ることは難しい。本章では後者の立場に立ち、モデルの内容を未知として観測データからシステム同定するモデル化手法を開発した⁷⁹⁾。

3. 2. 2 器官生長モデルの定式化

器官生長を非線形（2次式）、器官が相互に及ぼす影響を線形（1次式）として扱い、器

官生長のモデルを,

$$\frac{dg_i}{dt} = g_i \left(r_i - \sum_{j=1}^m \mu_{ij} g_j \right) \quad (i=1,2,\dots,m) \quad (3-1)$$

と定式化しよう。ただし、 m は器官の数、 r_i は器官 i の生長率、 μ_{ij} は器官 i から器官 j への干渉の程度を表す重み係数（以下、干渉作用係数と呼ぶことにする）である。器官 i の生長が器官 j の生長を阻害するとき μ_{ij} は正、器官 i の生長が器官 j の生長を促すとき μ_{ij} は負である。 g_i は生体重や乾物重、大きさ、体積など、植物生長に関して指標となる量である。ここでは g_i の内容を特に限定しないで扱うことにし、正規化された無次元量とする。以下では g_i を器官の現存量と呼ぶことにする。

(3-1)式は、生態学において Lotka-Volterra 方程式として知られる式を一般化した式（一般化 Lotka-Volterra 方程式と呼ばれる。以下、LV 方程式と略す。）として知られるものと同じになる。LV 方程式の解析解を求めることは困難であるが、数値計算からカオスなどの非常に複雑な解があることが知られている⁷³⁾。すなわち、(3-1)式は、植物の生長パターンの複雑さのかなりの部分を表現できる可能性を有している。

さて、生長は比較的緩やかな変化であり最終状態（収穫期）において、 g_i ($i=1,2,\dots,n$) は安定しなければならない。Shigesada *et al.* により、(3-1)式には、

$$\mu_{ii} > 0, \quad \mu_{ii} > \mu_{ij} \quad (i \neq j) \quad (3-2)$$

のとき Lyapunov 関数が存在し、大域的安定平衡点が存在することが証明されている⁷⁴⁾。すなわち、植物の生長を (3-1) 式を用いて表現する場合、(3-2)式が制約条件として必要であり、このことから現実の器官間の干渉は常に弱いものと推定される。以下では、さらに強い制約である、

$$|\mu_{ii}| \gg |\mu_{ij}| \quad (i \neq j) \quad (3-3)$$

を仮定する（この制約が適用できない場合、以下のモデルは (3-1) 式の良い近似にならない

3 量的変化のモデル化

が、(3-1)式のモデルとは別のモデルとして利用可能である。) . 器官 i に対する他の器官からの影響を、

$$z_i = \sum_{j(i \neq j)} \mu_{ij} g_j \quad (3-4)$$

とすると、(3-1)式は、

$$\frac{dg_i}{dt} = g_i (r_i - z_i - \mu_{ii} g_i) \quad (3-5)$$

となる。(3-5)式において、 g_i の変化が他器官への干渉を經由して z_i に及ぼす影響は $\mu_{ij}^2 \times g_i^2$ のオーダーで微小である。そのため、 z_i の時間変化を無視し、また、 r_i, μ_{ii} の時間変化は穏やかであると想定し短期間内ではこれらを一定とみなすことにしよう。これらの条件の下で(3-5)式を積分すると、

$$g_i = \frac{r_i - z_i}{\mu_{ii}} \frac{1}{1 + \exp(-(r_i - z_i)t + \theta_i)} \quad (3-6)$$

という近似式が得られる。ここで、 θ_i は器官現存量の初期値 $g_i|_{t=0}$ から決まる初期定数で、

$$\theta_i = \ln \left(\frac{r_i - z_i}{g_i|_{t=0} \mu_{ii}} - 1 \right) \quad (3-7)$$

である。最初の仮定より g_i は正規化された量であるから、(3-6)式中の右辺の係数 $(r_i - z_i) / \mu_{ii}$ は常に1でなければならない。すなわち、 μ の対角要素は自由パラメータではなく、

$$\mu_{ii} = r_i - z_i \quad (3-8)$$

である。これを(3-7)式に代入すると、

$$\theta_i = \ln \left(\frac{1}{g_i|_{t=0}} - 1 \right) \quad (3-9)$$

が得られる。したがって、(3-6)式は、

$$\begin{cases} g_i = \frac{1}{1 + \exp(-p_i t + \theta_i)} \\ p_i = r_i - z_i \end{cases} \quad (3-10)$$

となる。以下では簡単化のため、時間 t も $[0,1]$ の区間で正規化して考えることにする。

(3-10)式はニュートン・ラフソン法などにより数値的に解くことができるが、ここでは(3-10)式を器官が構成するネットワークとして表現し、ニューラルネット（人工神経回路網）に似たダイナミクスを付与して解く方法を提案する。この方法を採用する理由はいくつかあるが、一つはニューラルネットのために開発された種々の理論や解法、専用ハードウェアなどが応用できることである。また両モデルの共通性を考えることは、複雑な生物現象における単純な支配原理を見いだす手がかりになるとも考えられる。

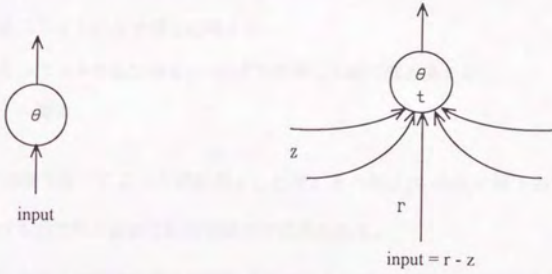
3. 2. 3 器官ネットワークモデル

(3-10)式より、単一器官の生長モデルを図 3-1(b)に示す非線形変換ユニット（以下、器官ユニットあるいは単にユニットと呼ぶ）として表すことにする。このユニットの入力は生長率から他器官からの干渉分を減じた値であり、出力は現存量である。ニューロンのモデル（図 3-1(a)）とは、ユニット内で時間 τ の乗算を行うところが異なっている。

器官ユニットを使うと、(3-4)、(3-10)式は図 3-2 に示すような器官ユニットが相互に結合したネットワークで表される（以下、これを単に器官ネットワークと呼ぶ）。図 3-2 においてユニット間を結ぶ矢印付きの線は、ユニットの出力値に作用係数を乗じた値を他方のユニットの入力値から減じることを意味している。器官生長率 r は、定数（=1）を常時出力するユニットから器官ユニットへの作用（重み係数）として表現する。

$$\text{output} = \frac{1}{1 + \exp(-\text{input} + \theta)}$$

$$\text{output} = \frac{1}{1 + \exp(-\text{input} \times t + \theta)}$$



a. ニューロンのモデル

b. 器官生長のモデル

図3-1 ユニットで表されたニューロンのモデルと器官生長のモデル

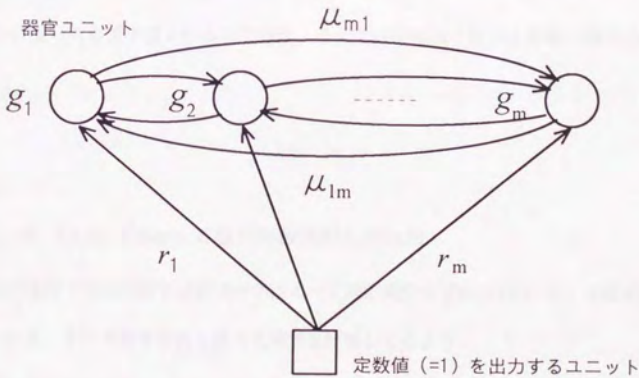


図3-2 器官間の干渉による影響を考慮した生長モデル

3 量的変化のモデル化

器官ネットワークにおいて器官現存量の初期状態を乱数で設定した後、(3-10)式をユニットの状態遷移規則として次のように動作させて見ることにしよう。

- I 各ユニットに対する器官間干渉量を計算する。
- II 各ユニットの入力値を計算する。
- III 各ユニットの出力値を(3-10)式で計算した値に置き換える。
- IV Iへ戻る。

I～IVの繰り返しによってが収束したとき、その値は(3-10)式の解である。すなわち、時間 t における器官現存量は上記の手続きで求められる。

器官間干渉が十分弱ければ g_i は明らかに収束し、(3-10)式には実数解がある。逆に(3-10)式が実数解を持たない場合は収束しない。器官間干渉が大きい場合でも(3-10)式の実数解は存在し、たとえばホップフィールドモデル⁷⁹⁾の研究から μ が実対称行列の場合は必ず収束することが知られている。しかし、器官ネットワークではむしろ反対称行列(要素が $\mu_{ij} = -\mu_{ji}$ となる行列)に近いことが予想されることから、ここでは器官間干渉が十分弱い場合だけを対象とする。

3. 2. 4 Lotka-Volterra 方程式の数値解との比較

(3-3)式の条件下では器官ネットワークによって数値積分せずに(3-1)式(LV方程式)の近似解が得られる。そこで簡単な例を使って両者を比較してみよう。

器官を2種として器官ネットワークモデルを、

$$\begin{cases} g_1 = \frac{1}{1 + \exp(-(20 - 0.5g_2)t + 10)} \\ g_2 = \frac{1}{1 + \exp(-(10 - g_1)t + 3)} \end{cases} \quad (3-11)$$

とする。このとき、生長率 r 、作用係数 μ 、初期定数 θ は、

$$r = \begin{pmatrix} 20 \\ 10 \end{pmatrix}, \quad \mu = \begin{pmatrix} - & 0.5 \\ 1 & - \end{pmatrix}, \quad \theta = \begin{pmatrix} 10 \\ 3 \end{pmatrix} \quad (3-12)$$

である。 μ の対角要素 μ_{ii} は(3-8)式で与えられる変数であるが、器官間干渉量が小さいため簡単に、

$$\mu_{11} = r_1, \quad \mu_{22} = r_2 \quad (3-13)$$

とする。このとき、対応するLV方程式は、

$$\begin{cases} \frac{dg_1}{dt} = g_1(20 - 20g_1 - 0.5g_2) \\ \frac{dg_2}{dt} = g_2(10 - 10g_2 - g_1) \\ g_1|_{t=0} = \frac{1}{1+e^{10}} \\ g_2|_{t=0} = \frac{1}{1+e^3} \end{cases} \quad (3-14)$$

となる。図3-3にLV方程式の数値積分解(実線)と器官ネットワークによる解(点線)を示す。両者の解を比較すると、器官ネットワークモデルがLV方程式の良い近似になっていることが分かる。後半部分では誤差が大きくなっているが、これは器官現存量の増大に伴い器官間干渉量が増えて、(3-8)式の正規化条件が満たされなくなるためである。そこでこの対角要素を(3-8)式の通りに行ってみると、器官ネットワークによる解と非常に近くなった(グラフが重なるため表示していない)。

このことから、器官現存量が小さいうちは器官ネットワークは制約のないLV方程式と良く一致し、器官現存量が大きくなると(3-8)式の制約の効果が現れるといえる。

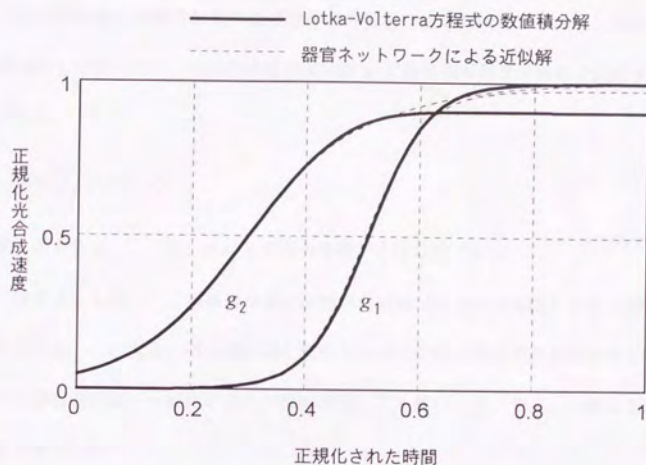


図3-3 Lotka-Volterra 方程式の数値積分解と器官ネットワークモデルによる解の比較

3. 2. 5 器官生長モデルの拡張

器官の生長は環境条件の影響を受けている。器官 i の生長率 r_i が環境要因 x_k ($k = 1, 2, \dots, n$) の影響を受けるとき、 r_i を、

$$r_i = \sum_{k=1}^n v_{ik} x_k \quad (3-15)$$

で表すことにする。ここで v_{ik} は環境要因 x_k の重み係数である。このとき器官現存量 g_i は、

$$\begin{cases} g_i = \frac{1}{1 + \exp(-q_i t + \theta_i)} \\ q_i = \sum_{k=1}^n v_{ik} x_k - z_i \end{cases} \quad (3-16)$$

である。これは図 3-4(a) に示すネットワークになる。ネットワーク上の \square は時刻 t における値 $x_k(t)$ を出力するユニットである。

3 量的変化のモデル化

個体全体の現存量は全器官の現存量の総和から容易に求められる。しかし、実際には全器官が観測可能とは限らない。一部の器官の現存量 g と個体現存量 Ψ の関係を記述する最も単純なモデルは、

$$\Psi = \sum_{i=1}^m w_i g_i + \xi \quad (3-17)$$

という式で表される。ここで、 w_i は g_i の重み係数、 ξ は定数である。

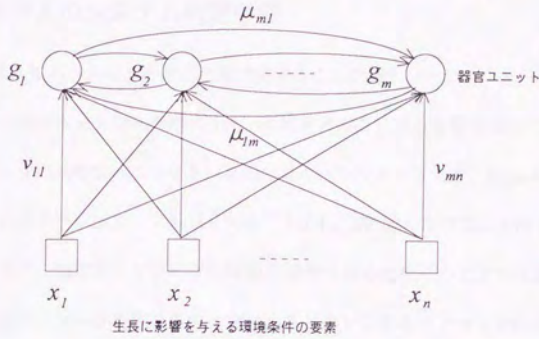
また、器官ネットワークに個体現存量を仮想的な器官 (ψ 番目の器官) として追加する方法も考えられる。この場合、個体現存量と器官現存量の間には相互結合ができるが、個体現存量から器官現存量への結合に具体的意味は見いだし難い。そこで、この結合を除去すると個体現存量 Ψ は単に、

$$\begin{cases} \Psi = \frac{1}{1 + \exp(-st + \xi)} \\ s = \sum_{k=1}^m w_{\psi k} g_k \end{cases} \quad (3-18)$$

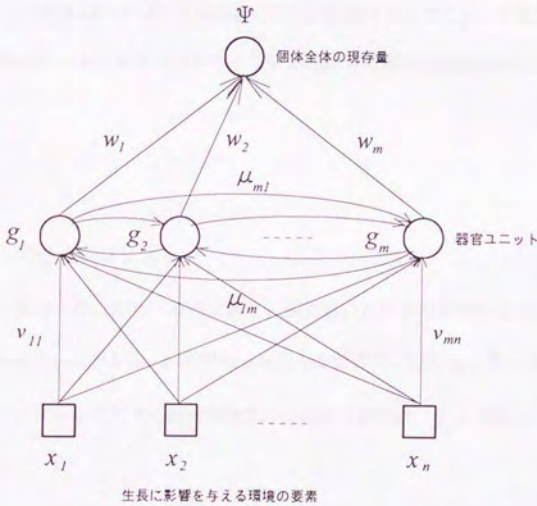
と書くことができる。ここで、 $w_{k\psi} = 0$ である。

(3-18) 式によって個体現存量を表現する方法は、全体の系を LV 方程式だけで表現できるというメリットがあり、専用のハードウェアによって LV 方程式を計算する場合に有用である (「5. 1. 2 ハードウェア化の可能性」を参照)。

未知パラメータの数は(3-17)、(3-18)式のどちらの方法でも同じであり、(3-17)または(3-18)式の特性的なユニットで個体現存量を表すと、いずれも図 3-4(b)示す構造のネットワークになる。



(a) 器官，環境変化を変数として含むモデル



(b) 器官，個体全体，環境変化を変数として含むモデル

図3-4 環境変化の影響を考慮した器官ネットワーク

3. 3 モデルのシステム同定手法

LV 方程式は未知パラメータを最小 2 乗推定することが難しいため、ニューラルネットの学習理論を参考に器官ネットワークのパラメータ推定アルゴリズムを考える。

隠れユニット（拘束のないユニット）を持つニューラルネットでは、Rumelhart *et al.* が提案したバックプロパゲーション・アルゴリズム⁷⁹⁾（以下、BP アルゴリズムと略す）が良く利用される。ところが、器官ネットワークには相互結合があるためそのままでは適用できない。近年、相互結合型ニューラルネットの一種であるリカレントネットワークにおいて、微分方程式で表現された誤差伝搬を利用して BP 学習を行うアルゴリズムがいくつか提案されている^{77,78)}。しかし、これらはパラメータの修正ごとに数値積分を必要とし、計算量が膨大となる。

以下では、器官ネットワークモデルのシステム同定を行うための実用的なアルゴリズムを提案する。

3. 3. 1 同定アルゴリズム

観測により、環境条件、器官（または個体）現存量、及びその時間からなるサンプルが多数得られたとしよう。このとき、 p 番目のサンプルの器官現存量を g_p 、個体現存量を Ψ_p 、時間を t_p とし、サンプル p に対する器官現存量の観測値（教師値） g_{pi} と器官ユニットの出力値 g_i の 2 乗誤差を、

$$E_p = \frac{1}{2} \sum_i (g_{pi} - g_i)^2 \quad (3-19)$$

とする。ただし観測されていない器官は隠れユニットとして扱い、(3-19)式には含めない。また個体現存量 Ψ だけが観測されている場合は、

$$E_p = \frac{1}{2} (\Psi_{pi} - \Psi_i)^2 \quad (3-19)$$

とする。このとき全サンプルに対する2乗誤差の和 E_{sum} は、

$$E_{sum} = \sum_p E_p \quad (3-20)$$

である。この器官ネットワークにおける未知パラメータを最小2乗法によって求めるには、 E_{sum} を最小にする値を求めれば良い。Amari の確率降下法⁷⁹⁾の考え方によれば、個々の E_p を等確率に逐次減少させることで最小化される。ユニット間の結合が一方方向のネットワークでは、確率降下法的一种であるBPアルゴリズムにより E_{sum} を最小化できる。すなわち器官ネットワークでは、器官間干渉の作用係数以外のパラメータがBPアルゴリズムにより推定できる。したがって、以下ではBPアルゴリズムが適用できない器官間干渉を求める方法だけを示す。

さて、 E_p は $-\partial E_p / \partial \mu_{ij}$ に微小定数を乗じた値を μ_{ij} に繰り返し加算することで極小化される(最急降下法)。 $\partial E_p / \partial \mu_{ij}$ は(3-4),(3-10)式から、

$$\frac{\partial E_p}{\partial \mu_{ij}} = \frac{\partial E_p}{\partial p_i} \frac{\partial p_i}{\partial \mu_{ij}} = \frac{\partial E_p}{\partial p_i} g_j = g_j \delta_i \quad (3-21)$$

となる。ここで、 $\delta_i = \partial E_p / \partial p_i$ とした。 δ_i は、さらに微分の合成法則から、

$$\delta_i = \frac{\partial E_p}{\partial p_i} = \frac{\partial E_p}{\partial g_i} \frac{\partial g_i}{\partial p_i} = g'_i \frac{\partial E_p}{\partial g_i} \quad (3-22)$$

となる。ここで、 $g'_i (= \partial g_i / \partial p_i)$ の値は(3-10)式から計算できる。しかし、 $\partial E_p / \partial g_i$ の計算では、 g_i が(3-4)式を介して他のユニットを変化させる影響を考慮しなければならない。

前章では(3-10)式を状態遷移規則としてダイナミックに変化する器官ネットワークを利用したが、ここでも $\partial E_p / \partial g_i$ を計算するため同じ変化をする器官ネットワークを考える。

Rumelhart *et al.* は相互結合を持つネットワークを階層型ネットワークに置き換える方法を提

3 量的変化のモデル化

案しているが⁸⁰⁾、これは $\partial E_p / \partial g_i$ の計算方法を考える際の参考になる。そこで、以下では器官ネットワークと等価な階層型ネットワーク（以下、等価ネットワークと略す）とを対比しながら考える。ただし、以下で示される等価ネットワークと Rumelhart らの方法とは異なっている。

等価ネットワークは全ての層のユニット数が等しい階層型ネットワークである。各ユニットは真上と真下のユニットを除く上層と下層の全てのユニットに接続されている。ある層の左から i 番目のユニットについて見ると、真上のユニットを除く全ての層のユニットと結合し、その結合係数は μ_{ki} ($k=1, 2, \dots, m$) である。

器官ネットワークでは、全ての器官ユニットが(3-10)式に基づいて一斉に状態遷移する。等価ネットワークにおいて下層のユニットの出力値が状態遷移前の状態と等しいとき、上層のユニットの出力値は状態遷移後の状態と等しい。したがって、器官ネットワークにおいて相互結合を通じて影響が無限に循環する過程（図 3-5(a)）は、階層数が無限大の等価ネットワークとして表現できる（図 3-5(b)）。

この等価ネットワークの入力層に器官ネットワークと同じ初期値を与えたとしよう。入力層の1段上層における出力値のパターンは器官ネットワークで状態遷移が1回起こった時の出力値のパターンに等しくなる。器官ネットワークが状態遷移を繰り返し行い器官ユニットの出力値が収束したとき、等価ネットワークの出力層付近の層でも同じ値に収束している。

さて、(3-22)式の $\partial E_p / \partial g_i$ を等価ネットワーク上で考えてみよう。まず器官ネットワークにおいて相互結合が無いとしたときの δ_i ($\delta_i^{(0)}$ と書くことにする) は等価ネットワークの出力層における値に等しい。このときの $\partial E_p / \partial g_i$ は(3-19)、(3-19)式から、それぞれ、

$$\frac{\partial E_p}{\partial g_i} = g_{pi} - g_i \quad (3-23)$$

または、

$$\frac{\partial E_p}{\partial g_i} = \frac{\partial \Psi}{\partial g_i} (\Psi_p - \Psi) \quad (3-23)'$$

である。したがって、

$$\delta_i^{(0)} = g_i' (g_{pi} - g_i) \quad (3-24)$$

または、

$$\delta_i^{(0)} = g_i' \frac{\partial \Psi}{\partial g_i} (\Psi_p - \Psi) \quad (3-24)'$$

である。器官ユニット i が隠れユニットの場合は、

$$\delta_i^{(0)} = 0 \quad (3-24)''$$

となる。

次に器官ネットワークにおいて、直接、接続している器官ユニットの影響分 $\delta_i^{(1)}$ を考えてみる。等価ネットワークにおいて $\delta_i^{(1)}$ は出力層のユニットへ1段下層のユニットが与える影響のことである。そこで $\partial E_p / \partial g_i$ を下層のユニット k ($\neq i$) の入力値 p_k による多変数微分として考えると、

$$\frac{\partial E_p}{\partial g_i} = \sum_{k(n_i)} \frac{\partial E_p}{\partial p_k} \frac{\partial p_k}{\partial g_i} \quad (3-25)$$

であるから $\delta_i^{(1)}$ は、

$$\delta_i^{(1)} = g_i' \sum_{k(n_i)} \frac{\partial E_p}{\partial p_k} \frac{\partial p_k}{\partial g_i} \quad (3-26)$$

となる。この式中の $\partial E_p / \partial p_k$ は上記で求められた $\delta_i^{(0)}$ であり、また、(3-4)式から $\partial p_k / \partial g_i = -\mu_{ki}$ である。したがって、(3-26)式は、

$$\delta_i^{(1)} = -g_i' \sum_{k(n_i)} \mu_{ki} \delta_k^{(0)} \quad (3-27)$$

となる。

さらに、下層からの影響についても上記と同様であり、s 段下層のユニットからの影響は、

$$\delta_i^{(s)} = -g'_i \sum_{k \in (s)} \mu_{ik} \delta_k^{(s-1)} \quad (3-28)$$

となる。

δ_i を誤差信号と呼ぶことにすると、等価ネットワークにおいて(3-28)式は出力層側から入力層側への誤差信号の伝搬ルールと解釈できる。この誤差信号の伝搬は器官ネットワークにおいて相互結合上を無限に循環する誤差信号の伝搬と解釈できる。等価ネットワークでは誤差信号は出力層側から入力層側へ伝搬するが、器官ネットワークでは同じユニットを何度も通過して伝搬するため器官ネットワークのユニット i では誤差信号が積算される。したがって、 δ_i は、

$$\delta_i = \sum_{s=0}^{\infty} \delta_i^{(s)} \quad (3-29)$$

である。器官間干渉の非対角要素の絶対値が十分小さければ $\delta_i^{(s)}$ はすぐに減衰し(3-29)式は収束する。収束しないときには(3-3)式の条件を満たさないとみなし、初期条件を変えてやり直すことにする。

(3-21)、(3-29)式から $\partial E_p / \partial \mu_{ij}$ は計算できるが実際のプログラムでは修正量 $\Delta \mu_{ij}$ を、

$$\Delta \mu_{ij} = \alpha \Delta \mu_{ij}^{old} - \eta \frac{\partial E_p}{\partial \mu_{ij}} \quad (3-30)$$

η : 学習係数, α : 安定化係数, $\Delta \mu_{ij}^{old}$: 前回の修正量

とする。学習係数 η と安定化係数 α は、BP アルゴリズムと同様、修正量の振動を抑えるための経験的パラメータである。全てのサンプル p に対して $\Delta \mu_{ij}$ を計算し、 μ_{ij} を $\mu_{ij} + \Delta \mu_{ij}$ に修正する。この修正を $\Delta \mu_{ij} = 0$ になるまで繰り返すと、 E_{sum} を極小にする μ が得られる。具体的には、以下のような手続きで行う。

<Step 1> {ユニット出力値, 干渉作用係数, しきい値をランダムに初期化}

- <Step 2> {学習用観測サンプルを1つ読み込む}
 <Step 3> {ユニット出力値 $\{g_k, g_i, \dots\}$ を計算}
 <Step 4> {観測値とユニット出力値との2乗誤差の計算}
 <Step 6> *if* (2乗誤差 < 打ち切り誤差) *end* ;
 <Step 5> {2乗誤差の微分から誤差信号を計算}
 <Step 6> *while* (誤差信号 > ϵ)
 {ネットワーク上で誤差信号を逆方向に伝搬させ、各ユニット上で誤差信号を
 累積加算}
 <Step 7> {誤差信号の累積値を用いて干渉作用係数、しきい値の修正}
 <Step 8> *go to* <Step 2> ;

このアルゴリズムを検証するため、(3-11)式の例を使って試した。図3-3のグラフにおいて等間隔に (t, g_k, g_i) の値をサンプリングした41個のサンプルを使用し、 r, μ が未知であるとして、その値を上記のアルゴリズムにより推定した。(30)式のパラメータは $\eta=4.0, \alpha=0.4$ とし、 r, μ は $[0,0.1]$ の一様乱数で初期化した。約1万回の修正で平均2乗誤差(= E_{sum} / 器官ユニット数 / サンプル数)は 10^{-4} 以下になり、 r, μ の値は(3-12)式にはほぼ一致した。その際、相互結合上を伝搬する誤差信号は、ほとんどの場合、4~5回以内の循環で十分に減衰した。

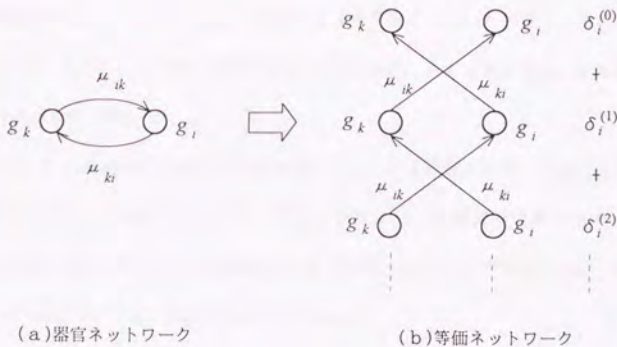


図3-5 器官ネットワークの相互結合による信号の循環を表現する等価ネットワーク

3. 3. 2 システム同定例

器官ネットワークに含まれるパラメータには強い制約があるため、表現可能な関数関係は限られている。したがって、もし器官ネットワークモデルが現実から遊離したものであれば、システム同定の際に E_{sum} を小さくできないと考えられる。そこで、器官ネットワークモデルの妥当性を評価するため、大豆（品種：エンレイ）の生長に関する観測データ⁸¹⁾を使ってシステム同定を行った。

器官ユニットは葉、莖、根、子実の4種（乾物量）とし、モデルに組み込む環境要因は正規化した温度と日射量の2種とした。ただし、これらの環境要因は時間的変動が大きい。そこで器官生長はその時点までの累積された影響を受けていると考えて、温度、日射量の値は観測開始時からデータのサンプリング時までの平均値とした。この器官ネットワークを図3-6に示す。

同定には、日本の3地域（刈和野、塩尻、熊本）において1978年から1981年の4年間に観測された12種の時系列パターン（70個のサンプル）を使用した。4種の乾物重と2種の環境要因、時間（日数）の各値は、それぞれの最大値で除算して正規化した。

器官ネットワーク中の各パラメータを[0,0.1]の一様乱数で初期化した後、学習係数 η を0.05、安定係数 α を0.2として約10万回のパラメータ修正を行ったところ、平均2乗誤差は0.01になった。ただし、(3-3)式の制約を満たす解に収束しやすくするため、 μ の場合だけ学習係数を $\eta/20$ (=0.0025)にした。

図3-7に、ある気象条件（1979年の刈和野）における計算結果を示す。各器官の生長パターンは全体として正しく再現されていた。また、子実について観測値（●で示す）と計算値（実線）を比較すると定量的にも比較的精度良く再現されていると判断される。他の年度や地域のサンプルについても、ほぼ同様の傾向であった。

この解析例では全栽培期間にわたるデータを使ったため、 μ には器官間干渉の平均的特性

3 量的変化のモデル化

が同定されていると考えられる。そこで、 μ の値を調べてみた。図 3-8 は μ を器官ユニットごとに分けて表示したもので、作用係数が正の場合は実線、負の場合は破線で示した。葉から他の器官への作用は全て負になっているが、これは葉の生長が他の器官の生長に寄与することを示している。一方、子実から他の器官への作用は全て正であるが、これは子実が他器官の生長を抑制しながら生長していることを意味している。これらのことから、主として器官間のソース-シンク関係が器官間干渉の作用係数として同定されていると考えられる。

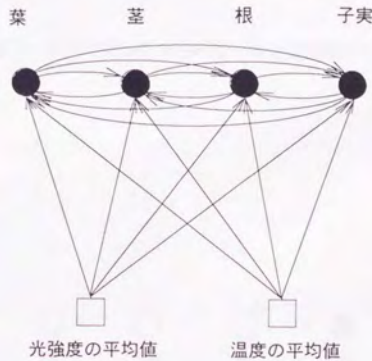


図 3-6 ダイズの簡単な器官ネットワークモデルの例

3 量的変化のモデル化

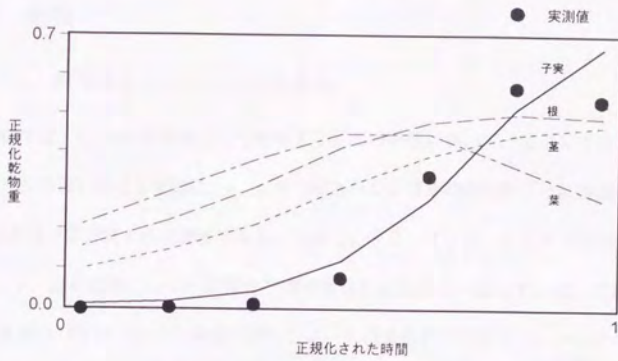


図3-7 モデルの出力パターンと観測値の例

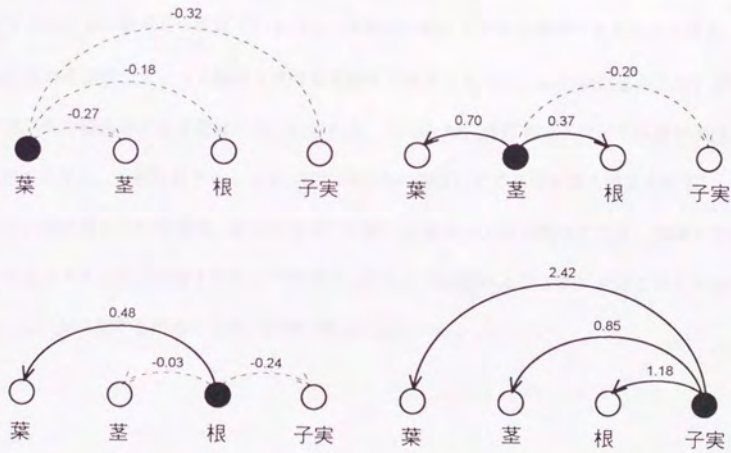


図3-8 推定された干渉作用係数

3. 4 考察

3. 4. 1 推定されたパラメータの意味

上記の例では、 r 、 μ の時間変化を考慮せずに全ての時刻において一定として計算したが、生育ステージの違いによる変動は、 r 、 μ を一定とみなし得る時間区間ごとに観測データを分け、 r 、 μ を別々に計算すれば評価できる。しかし、「3. 3. 2 システム同定例」で示したように、 r 、 μ を定数としても計算値と観測値は比較的良く一致している。これは生長パターンの多様なバリエーションが個々のパラメータの変化だけではなく、パラメータ全体の静的構造によっても生み出されることを示している。

システムの構造の特徴のほとんどは μ に含まれるが、 μ が張る空間 $M \subset R^{m(m-1)}$ は競合(共生)に関する特徴空間である。観測データから μ を最小2乗推定するプロセスは M への射影(フィルタ)と解釈され、システムの特徴は M における μ の位置で表される。このとき E_{sum} が小さく μ が原点から離れているほど、変数間の競合で現象が説明できたことになる。

(3-1)式の導出時にユニット間の干渉作用を線形と仮定したため、 μ の絶対値が大きい要素から強い依存関係のある2変数が見い出される。そのため、器官ネットワークは植物個体生長だけではなく、一般の複雑なシステムの相互関係の解析に広く適用可能と考えられる。

また、根粒菌との共生関係、雑草や害虫の影響、収穫等の人為効果などでは、関連する各要因をユニットとして定義するだけで器官ネットワークに組み込むことが可能と考えられる(ただし、 μ に関する制約のため、適用できない場合がある。)

3. 4. 2 ニューラルネットとの比較

生態系と器官ネットワーク、器官ネットワークとニューラルネット(神経系のモデル)には、(3-1)式を接点として様々な共通性がある。これは多様な生物系における統一モデル

3 量的変化のモデル化

を得るための手がかりになるかもしれない。そこで、現実の生物系としてはかなり遠縁関係にある器官ネットワークとニューラルネットワークとの関係を詳しく考えてみよう。両者には、まず次のような相違点がある。

- 1) ユニットの特性：ニューラルネットワークでは神経細胞の刺激に対する応答をロジスティック関数で近似しているが、器官ネットワークのユニットが表現するロジスティック関数は LV 方程式から導き出されたものである。
- 2) ユニット間の結合：ニューラルネットワークの場合はシナプス結合という実態があるが、器官ネットワークでは物質移動、空間の占有、光の遮蔽、植物ホルモンによる調節、老化などの種々の要因が複合したものである。
- 3) モデルの機能：ニューラルネットワークはユニット数が極端に多く任意の一価連続関数を近似できるが²⁷⁾、器官ネットワークはユニット数が比較的少なく近似可能な関数は限定されている（そのため、強いノイズを含むデータからでもパラメータ推定を行いやすい）。

このような相違点を踏まえた上で両者の類似性から器官ネットワーク（LV系）を情報処理機械として解釈してみよう。器官間干渉はニューラルネットワークのシナプス結合に対応し『分散記憶』とみなすことができる。このとき、環境条件の変化によって起こる生長パターンの変化は『パターン認識』または『連想』の機能に対応し、環境条件によって『分散記憶』の一つが引き出され特定の生長パターン（器官ごとの時間変化）が生成されると解釈される。器官が多数あることは『分散記憶』が冗長であることを意味する。一部の器官の破壊や環境条件のゆらぎに対して植物生長は強いロバストネスを有するが、このモデルにおいては『分散記憶』の冗長性が、その理由の一つとして挙げられる。これらのアナロジーは生長における最適制御法を考える際にも役立つと考えられる。

3 量的変化のモデル化

器官ネットワークのユニットには時間変数が含まれているが、ニューラルネット（ホップフィールドモデル，BP モデル）にはない。そこで器官ネットワークの時間変数を定数にする
とホップフィールドモデルと同じになる。さらに，結合の一部を切断すれば BP モデルと同じになる。すなわち，器官ネットワークは，これらのモデルを特殊なケースとして包含している。したがって，器官ネットワークとニューラルネットは LV 系の特殊形態と考えられる。

ちなみに，このことからニューラルネットを器官ネットワークに埋め込むことができることが分かる。たとえば，環境条件と光合成速度の関係式などをニューラルネットで表現し，これを器官ネットワークに接続すれば大規模なメカニスティック・モデルが構成できる。

3. 5 結論

環境条件に対して複雑な応答をする植物個体生長を、非線形特性を有する器官を単位としたシステムとして表現するモデルを提案した。また、このモデルを少ない計算量で近似的に解くアルゴリズム、及び未知パラメータを最小2乗推定するアルゴリズムを提案した。これらは簡単な例題と観測データを使ったテストにより妥当性が確認された。提案されたモデリング手法は植物の個体生長だけでなく、多数の変数が相互に干渉しながら時間発展する一般的な複雑系において適用可能と考えられる。

4 生長の複雑さに関する実験と予測モデル

第2, 3章では, 植物の比較的長期的な生長パターンの多様な変化をモデル化する方法について考察した. 植物生長における最も重要なパラメータは光合成速度と生長速度であるが, 第2章の形態モデルでは光合成速度を外部変数として与えた. 第3章の量的変化のモデルでは生長速度(生長率)を観測した各器官の現存量(乾物重)から求めた. そして, 環境変化に対して個々の器官の応答を非常に単純な式で表現し, しかも生長率を一定とした場合でも, 器官が複数存在することによって複雑で多様な生長パターンが生成されるを見た.

環境条件が複雑に変動したとき, 当然, 光合成速度などの生理的パラメータも複雑に変動しているはずである. その結果, 植物の生長パターンは, さらに複雑な変化をすることになるだろう. しかし, 生長パターンがあまりに複雑すぎると, 環境変化に対して植物は適応的に生長することが困難となってくると思われる. それでは, 短時間で変化する環境の変動に対して, 植物はどのように応答しているのだろうか?

ここでは, 人工環境下で環境を時間的に変化させ, 光合成速度や生長速度がどのように応答するかを実験的に解析する.

4. 1 実験の目的と概要

植物は、通常、複雑に変化する環境の下で生長している。植物生長が環境変動に対して受動的に揺さぶられているとすれば、これは生長を複雑にする非常に大きな要因になっているはずである。自然環境のように極端に大きな外乱がある環境下で、植物はどのようにして適応的な生長を実現しているのだろうか？ また、このような複雑な変化は、どうすればモデルに組み込むことができるだろうか？

植物生長を計測する重要なパラメータには、光合成速度や生長速度がある。そのため、これらは植物の生育を制御する際の評価関数としても、しばしば利用されている。環境変化に対するこれらの変化を予測するモデルがあれば、植物の生長制御も容易になる。そこで、ここでは光合成速度と生長速度について考えることにする。

光合成などの生体内代謝には非常に多くの要素が関係し、未知の要素も多く残されている。これらのメカニズムに基づいてボトムアップ式に光合成モデルを作ろうとすると、モデルは非常に複雑になり、未知のパラメータを数多く含むものになる^{83,84}。そのため、植物の生長制御を行う際には、光合成モデルを用いずに、光合成速度を直接計測して最適条件を探索する方法が試みられている⁸⁵。しかし、探索空間が非常に広く、しかも探索速度は植物の生理的応答時間に律速されているため、大域的最適状態を限られた時間内で発見することが難しいという問題がある。さらに別の方法として、測定データを使ってモデルをシステム同定しながら最適条件を探索する方法が考えられる。しかし、環境条件と植物の光合成速度と生長速度の関係には複雑な非線形関係があるため、モデルのシステム同定には多くのデータが必要である。この場合、短時間にどれだけ多くの情報量を得ることができるかがキーとなる。

そこで、種々の環境条件下で出来る限り多くのサンプルを得るための実験を試みた。その結果、得られたデータには種々の不規則な変化が多数観測された。

非線形系では、構造が複雑でも全体としては単純な挙動を示す場合や、逆にカオスのよう

4 生長の複雑さに関する実験と予測モデル

に単純なシステムがランダムに見えるほど複雑な変化を生成する場合がある⁸⁶⁾。そこで、観測された不規則な変化がカオスであるかどうかを解析した。

その結果、この複雑な変化はカオスであると判定された。さらに光合成速度については、非線形モデルによる予測結果と線形モデルの予測結果を比較し、カオス性の検証とモデルの予測可能性の検討を行った^{87,88)}。

以上の実験では、光合成速度の変動のカオス性を全く予期せずに行っていたため、実験条件が非常に複雑なものとなっていた。そこで、環境条件の変化を単純なものにした実験をデザインし、カオス性の検証と発生条件を実験的に調べた。その実験においても光合成速度の複雑なゆらぎが観測され、そのゆらぎはカオスであった。

4. 2 カオス性の判定手法

4. 2. 1 観測データに基づく判定手法

気象⁸⁹⁾ (Navier-Stoks 方程式) や生態系での個体数変化⁹⁰⁾ (Logistic 写像, Lotka-Volterra 方程式) のように支配方程式が既知の場合, カオス性の判定は容易であり, これらがカオスであることはかなり以前から知られている. 最近では時系列データに対しても適用可能なカオス性の判定方法がいくつか提案され, 脳波⁹¹⁾, 心臓の心拍数のゆらぎ⁹²⁾, 経済循環や株価変動⁹³⁾ などの現象もカオスであることが判明している.

観測によって得た時系列データに対するカオスの判定方法にはいくつかあり, 主として次のような方法が利用されている⁹⁴⁾.

3次元以下のカオスでは,

①時系列データを適当な空間に埋め込んでストレンジアトラクタの存在を調べる方法

②ポアンカレリターンマップの自己相似性を調べる方法

などがある. 高次元のカオスの場合はこれらの方法は適用し難いことが多いため,

③相関次元

④最大リアプノフ数を用いる方法

がしばしば利用されている. また,

⑤カオスを生成するメカニズムのモデルを観測データから同定して決定論的生成過程の存在証明を行う

という方法がある.

観測によって得られた時系列データを使った判定では, 常に統計学的な不確実性が伴っている. そのため異なる判定法を併用して, 判定の信頼性を高める必要がある. ここでは, 手法的に大きく異なる①, ③, ⑤をカオス性の判定に用いることにした.

4. 2. 2 相関次元の計算法

Takens⁸⁵⁾によると、高次元力学系の軌道は任意に選んだ一つの次元の情報だけから再構成できることが示されている。たとえば光合成系が多数の独立な変数から構成されていても、測定が容易な一つのパラメータ（たとえば光合成速度）に関するデータだけから光合成系全体の特性を知ることができる。

元の高次元系の特性を1次元の時系列データ（図4-1）から解析するために、まず時刻 t における観測値 $x(t)$ 、及び観測時刻が $\tau, 2\tau, \dots, (d-1)\tau$ だけずれた $d-1$ 個のサンプルを用いて d 次元ベクトル、

$$\begin{aligned} & \{\psi_d(1), \psi_d(2), \dots\} \\ & = \left\{ \left(x(0), x(\tau), \dots, x((d-1)\tau) \right), \left(x(\tau), x(2\tau), \dots, x(d\tau) \right), \dots \right\} \end{aligned} \quad (4-1)$$

を作成する（ d は埋め込み次元と呼ばれる）。このとき元の系の相関次元 D は、このベクトル集合を使って、以下の方法で求めることができる⁸⁶⁾。

まず d 次元データの D_d を次式によって計算する。

$$D_d = \lim_{r \rightarrow 0} \frac{\log C(r)}{\log r} \quad (4-2)$$

ここで $C(r)$ は、

$$C(r) = \frac{1}{n^2} \sum_{i,j(0 \neq j)} H(r - |\psi_d(i) - \psi_d(j)|) \quad (4-3)$$

で定義され、相関積分と呼ばれている。 H は Heviside の階段関数で、

$$H(\eta) = \begin{cases} 1 & (\eta \geq 0) \\ 0 & (\eta < 0) \end{cases} \quad (4-4)$$

である。異なる d について D_d が一定値 D になったとき、 D が相関次元である。ただし、常に $D_d \leq d$ になるため、 D よりも十分大きな d で D_d が一定になる必要がある。具体的には、 r と $C(r)$ を座標軸とした両対数グラフ上で十分大きな d についてをプロットしたとき、グラフが

4 生長の複雑さに関する実験と予測モデル

直線領域を有し、異なる d のグラフが平行であれば相関次元が存在する。そして、相関次元 D はグラフの傾きから求められる。

$C(r)$ は $\{\psi_d(1), \psi_d(2), \dots\}$ が分布する d 次元空間において、2 点間の距離が r 以内にある点の個数 (図 4-2) の累積分布関数であり、 D_d はその分布の特徴を表すパラメータである。 d と r の変化に対して D_d が一定であれば、元の高次元系にはスケール変換に対する自己相似性が存在する。したがって、相関次元が存在すれば、その現象はランダムではなくカオスであると判定される。

τ の選び方は任意であるが、 τ があまりに小さいと要素間の相関が大きくなってしまう。逆に大きすぎるとノイズの影響が出てくる。ここでは、自己相関関数の絶対値を極小にする値のうち最小の値を τ として用いることにする。

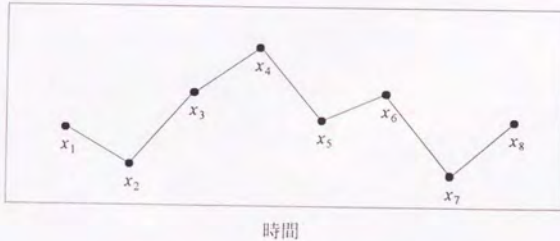


図 4-1 1次元の時系列データ

4 生長の複雑さに関する実験と予測モデル

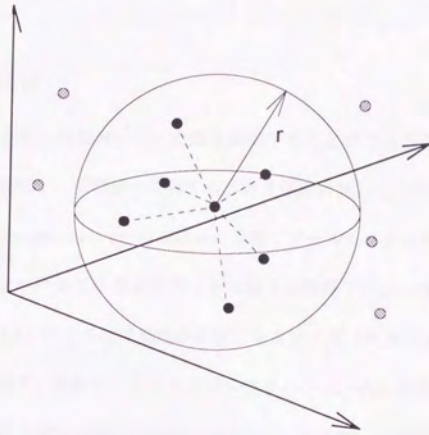


図4-2 ある点から距離が r より小さい点の総数

4. 3 実験

4. 3. 1 実験方法

温度、相対湿度、気圧、光強度、CO₂濃度を制御することができるグロースチャンバーを用いて植物を長期間栽培し、環境変化に対する応答を連続計測した(図4-3, 図4-4を参照)。供試植物はメロン (*Cucumis melo* var. *reticulatus*, 品種: アールセイヌ夏II) である。

まずグロースチャンバー外で小型水耕ポットに種子を播種し、人工光下で育成した。レーザー外径測定器(図4-5)による莖径測定が可能になるサイズ(本葉が2枚程度になった時)にまで生長したところで、水耕ポットごとグロースチャンバー内に設置し、以後、本実験を含む約3カ月の間、完全密閉状態で連続的に栽培した。その間、水耕液はチャンバー外から約3~7日間隔で密閉状態を維持したまま交換した。

測定項目は、CO₂の消費量から求めた見かけの光合成速度、レーザー外径測定器で計測した莖の根元直径、熱電対で計測した葉裏面の表面温度などである。

グロースチャンバー内に設置した日を基準日として、24日目から55日目までの31日間、温度、相対湿度、CO₂濃度、気圧の順に変化させ、そのときの植物の生理的応答を測定した。これらのパラメータは、表4-1の範囲で変化させた。

環境条件を変化させると植物の状態が変化するため、その影響を受けてグロースチャンバー内のCO₂濃度、相対湿度、気温も多少変化する。非線形系では、この過渡応答にも重要な情報が含まれている可能性が高いが、これには実験系の過渡特性も影響していると考えられるため、本実験では利用しない。このような過渡応答の影響を減らすと同時に限られた期間に出来るだけ多数のサンプルを得るため、次のように実験系を制御した。

- ① 環境条件を変える。
- ② CO₂濃度、相対湿度、気温が安定するのを待つ(CO₂濃度の変動幅が5.0ppm以内、露点温度及び気温の変動幅が0.5°C以内になったとき安定したとみなす)。

4 生長の複雑さに関する実験と予測モデル

③ 10分間同じ環境条件を維持し、最後の5分間の計測値の平均を記録する。200分を経過しても安定しない場合は最後の5分間の平均値を記録する。

④ ①から繰り返す。

茎径の時系列データから生長速度に相当する値を計算するため、

$$\dot{R}(t) = \frac{R(t) - \bar{R}(t)}{\bar{R}(t)} \quad (4-5)$$

$$\bar{R}(t) = \frac{1}{10} \sum_{i=0}^9 R(t-i) \quad (4-6)$$

とした。さらに $\dot{R}(t)$ が0~1の範囲になるように正規化し、この値を生長速度 $R_g(t)$ として用いることにした。

ただし、この $R_g(t)$ には、茎の水分状態の変化による変動分が含まれているが、生長による茎径変化と水分状態の変化による茎径変化とは分離することが難しい。しかし、以下の実験では水耕によって植物を栽培したため、茎における水分欠乏による茎径変化の影響は比較的少ないと考えられるため、ここでは水分状態による茎径の変化も生長変化の一種として扱い、以下では両者を分けずに $R_g(t)$ をそのまま生長速度として呼ぶことにする。

4 生長の複雑さに関する実験と予測モデル

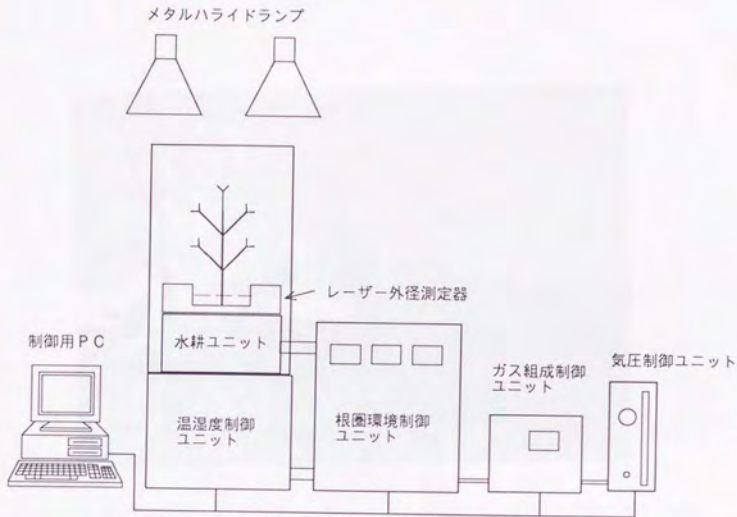


図4-3 実験システムの構成

表4-1 実験における各環境因子の値の変化範囲

環境因子	変化の範囲
気温	13 - 40 [°C]
相対湿度	35 - 90 [%]
CO ₂ 濃度	30 - 1000 [ppm]
気圧	0.1 - 0.13 [MPa]

4 生長の複雑さに関する実験と予測モデル



図4-4 実験システムの全体の様子

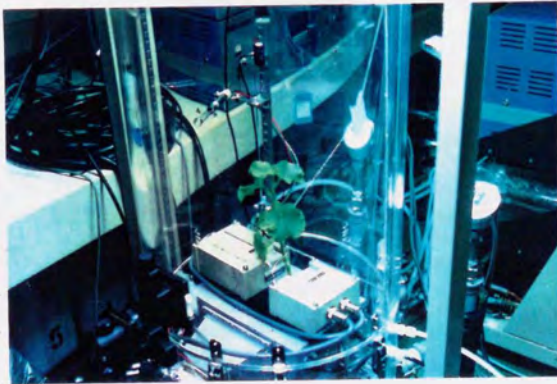


図4-5 レーザー外径測定器による莖径の計測

4.3.2 実験結果

各環境パラメータは、ほとんどの場合で 20~120 分以内で安定に達し、最頻値として約 30 分ごとに 1 つのサンプルが得られていた。水耕液の交換等のメンテナンス時には、その影響が完全に無くなったと見なせるまで同一環境条件を長時間継続した。

図 4-6 に測定された各変数の変化の様子を示す。このグラフの横軸は時系列データのサンプル番号である。環境条件に対する光合成速度の応答は不規則的であり、なぜかグラフの前半部分が後半部分よりも複雑である。

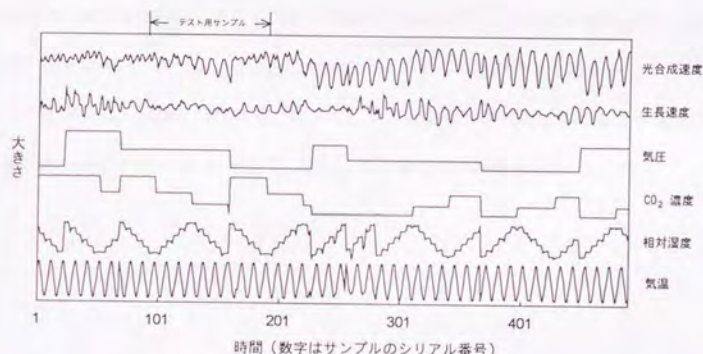


図 4-6 環境変化に対する光合成速度及び生長速度の応答 (横軸は実験回数番号、間隔は約 30 分。)

図 4-7(a)は、測定で得られたデータ (図 4-6 で示した時系列データ) を温度-相対光合成速度の平面でプロットした軌道図である。また、図 4-7(b)は温度-茎径の生長速度の平面でプロットした軌道図である。図 4-7(a),(b)を見ると、環境変化に対する植物個体の応答はどれも複雑であり、グラフから明確な法則性を見いだすことは困難である。

図 4-8 は観測された光合成速度の値を気圧ごとにプロットしたグラフである。図 4-6 で示し

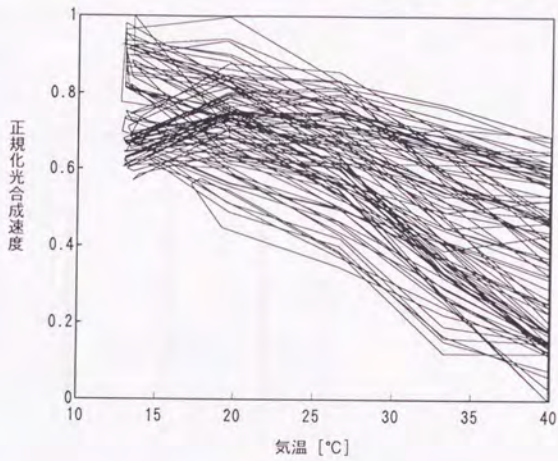
4 生長の複雑さに関する実験と予測モデル

たように、この実験では5種類の異なった気圧が選択されたため、それぞれの気圧における光合成速度の値が縦方向に分散してプロットされている。各気圧において気温、相対湿度等の違いによって光合成速度が異なるため光合成速度の値が縦方向に分布している訳であるが、気圧の違いによる分布のばらつきは非常に大きく、気圧-光合成速度の間に単純な傾向は全く見いだせない。

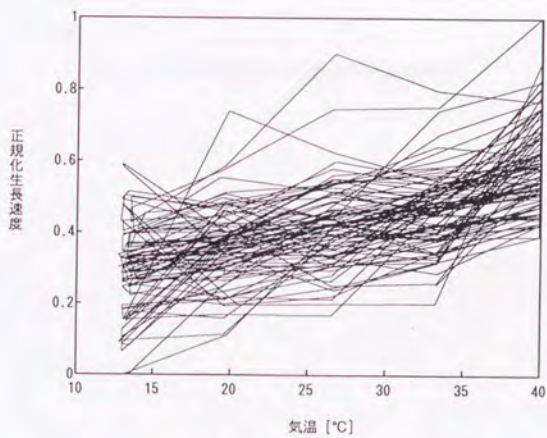
測定値の平均から大まかな傾向を見出す方法がしばしば用いられるが、図 4-7(a)では一部に上に凸の放物線状のパターンがあり、この部分に着目して平均化すれば温度-光合成の特性に関する既往の知見⁹⁷⁾と同様の規則が見い出される。しかし、この方法では平均的傾向から外れた変化(ゆらぎ)をノイズとして扱うことになり⁹⁸⁾、また図 4-8 のように単純な傾向が見い出せないデータでは、ほとんど全てをノイズとして扱わざるを得ない。

この実験では、観測されたゆらぎが全てノイズに帰着できるほど実験条件や測定精度が不安定であったとは考え難い。そこで、次に、このゆらぎの解析を行う。

4 生長の複雑さに関する実験と予測モデル



(a) 温度-光合成速度



(b) 気温-生長速度

図4-7 気温と光合成速度及び生長速度の平面における軌道

4 生長の複雑さに関する実験と予測モデル

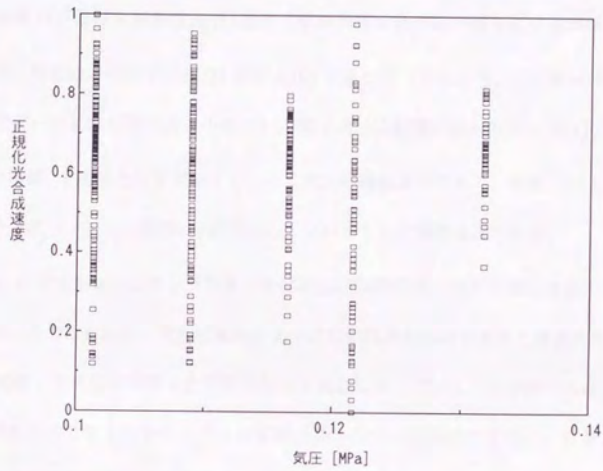


図 4-8 気圧の変化と光合成速度の関係

4. 4 カオス性の解析

4. 4. 1 ストレンジアトラクタの観察

光合成速度 $P(t)$ 及び生長速度 $R_g(t)$ の自己相関関数の絶対値が極小になる点とともに $\tau = 3$ のときで、相関値はそれぞれ 0.028 及び 0.105 であった (すなわち、3 ステップだけシフトした時系列データとの相関が最も小さい)。図 4-9(a) は相関が最も小さい $P(t)$, $P(t+3)$ をそれぞれ x 座標, y 座標としてプロットした 2 次元の相軌道図である。また、図 4-9(b) は、同様に $R_g(t)$, $R_g(t+3)$ を x 座標, y 座標としてプロットした相軌道図である。

図 4-9(a) にはストレンジアトラクタと呼ばれる空間的に偏った分布構造を持つパターンが現れている。このことから、光合成速度の変化は比較的低次元のカオスと推測される。一方、図 4-9(b) では、中央部が密集した対称的な分布構造になっている。この図からは、生長速度のゆらぎがカオスであるのかランダムな変動であるのかの区別はできない。しかし、光合成速度がカオスであるとするれば生長速度もカオスである可能性が高く、生長速度のゆらぎは高次元のカオスである可能性が高い。



(a) 光合成速度



(b) 生長速度

図 4-9 光合成及び生長速度の相軌道図

4. 4. 2 相関次元によるカオス性の判定

次に、 $P_g(t)$ 及び $R_g(t)$ を用いて相関次元によるカオス性の判定を行った。

サンプル数 N が多いほど相関次元の推定値 D の信頼性は高まる。しかし、 D が大きいと非常に多くのサンプルが必要とされ、その目安として、

$$D \leq 2 \log_{10} N \quad (4-7)$$

という条件がある⁹⁹⁾。以下の解析に用いたサンプル数（光合成速度：498 個，生長速度：489 個）では、 $D < 6$ 程度までの D が計算できる（ただし、 D が大きいと計算結果の信頼性は低くなる）。

図 4-10(a) は、光合成速度の時系列データを用いて、 $\tau = 3$ 、 $d = 2 \sim 20$ のときの相関積分のグラフである。 d が異なる曲線の全てに直線領域が存在し、 d が大きくなるにつれて傾きが収束している。このことから相関次元が存在すると考えられ、直線部分の傾きから計算した相関次元 D は $2.7(\pm 0.4)$ であった。したがって、光合成速度のゆらぎは $D = 2.7$ 程度のストレンジアトラクタを持つカオスである可能性が高い。

図 4-10(b) は 莖径の生長速度の時系列データから求めた、 $\tau = 3$ 、 $d = 2 \sim 20$ のときの相関積分のグラフである。光合成速度の場合よりも傾きが大きく、直線領域は非常に狭くなっている。この部分から相関次元を計算すると $D > 4.5$ となるが、489 個のサンプルでは信頼性の高い判定は困難である。

これらの結果は、ストレンジアトラクタの観察に基づく判定と一致している。

4 生長の複雑さに関する実験と予測モデル

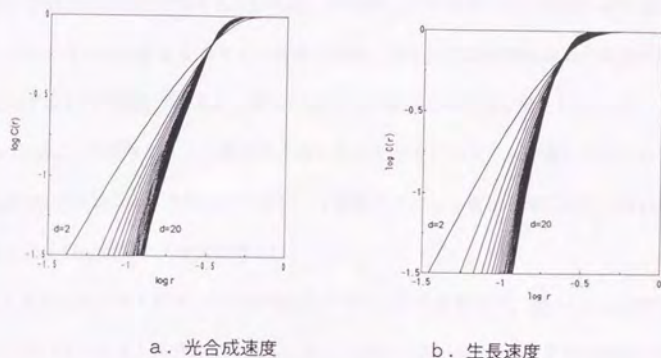


図4-10 埋め込み次元 $d=2\sim 20$ のときの r と相関積分 $C(r)$ の関係

4. 4. 3 非線形モデルの同定による検証

上記の解析結果が示す通り、光合成速度のゆらぎはカオスである可能性が非常に高い。もしカオスであるとすれば、そのカオスを生成するもとになる非線形モデルが存在する。相関次元の計算によって光合成速度が比較的低次元のカオスと推定されたため、短期予測であれば高い精度で予測できなければならない。そこで次に、このモデルの存在を示すことにより、カオス性を検証することにする。

光合成速度を予測する非線形モデルには、図4-11(a)に示す3層のニューラルネットを用いた。入力層のユニットは温度、相対湿度、 CO_2 濃度、気圧の環境状態に関する4変数、そして1ステップ前の光合成速度 $P(t-1)$ からなる5個とし、出力層のユニットは現在の光合成速度 $P(t)$ とした。中間層のユニットは5個である。

このモデルは、出力層のユニットの出力値が入力層の一つのユニットに時間遅れを伴って入力するリカレントネットワーク (Jordanのシーケンシャルネットワーク¹⁰⁰⁾) になっている。

4 生長の複雑さに関する実験と予測モデル

ある時点での光合成速度が得られていれば、再帰的にこのモデルを n 回動作させることで n ステップ先の光合成速度を予測することができる。カオスには初期値の微小な差が指数関数的に増大するという特性があるが、図 4-11(a)のモデルはこの特性を満たしている。

図 4-11(b)は、このモデルと比較するために作成したモデルで、入力層に 1 ステップ前の光合成速度 $P(t-1)$ のユニットを持っていない。中間層のユニット数は 6 個にした（自由パラメータの数は図 4-11(a)のモデルとほぼ等しい）。

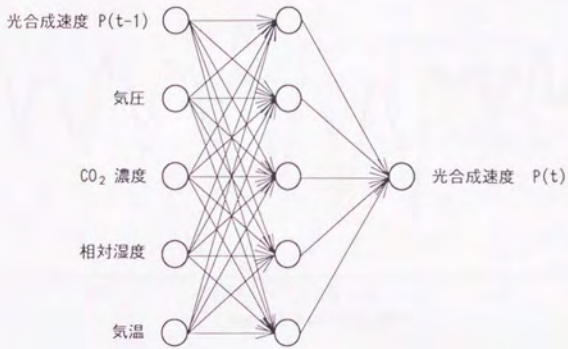
図 4-6 で示したテスト用サンプル領域を除いたサンプルを使って、図 4-11(a),(b)のモデルをバックプロパゲーション・アルゴリズムによって同定した。(a),(b)のモデルの比較のため、2乗誤差が 0.0001 になったところで両者とも学習を打ち切った。

図 4-12 は、図 4-7(a),(b)のモデルによる図 4-6 のテスト用データに対する予測結果（現時点の環境状態と 1 ステップ前の光合成速度の値を使った予測）である。図 4-13 は重回帰モデルを用いて、図 4-10 と同様に行った予測結果である。説明変数は図 4-11(a),(b)のモデルと同じで、4 つの環境因子と 1 ステップ前の光合成速度を含むモデル（予測値-a）と含まないモデル（予測値-b）である。

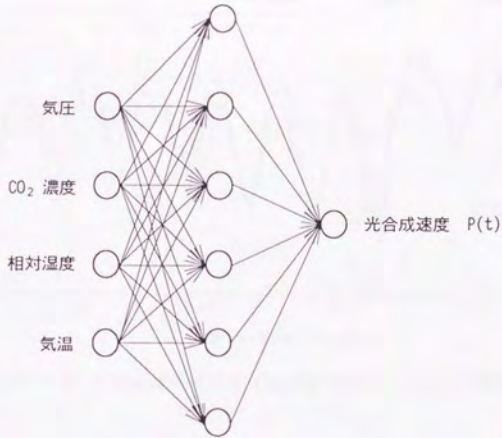
図 4-12 の(a),(b)を比較すると、(a)では小さな変化パターンのほとんどが正しく予測されているのに対し、(b)では予測できていない。重回帰モデルではどちらのモデルも（図 4-13(a),(b)）、不規則パターンは全く予測されていない。

これらの結果から、光合成速度の複雑な変化は、光合成系の履歴依存性と非線形性によって生み出されたカオスであると考えられる。

4 生長の複雑さに関する実験と予測モデル



(a) 入力：4種的环境因子と1ステップ前の光合成速度



(b) 入力：4種的环境因子

図4-11 ニューラルネットを使った2種のモデル

4 生長の複雑さに関する実験と予測モデル

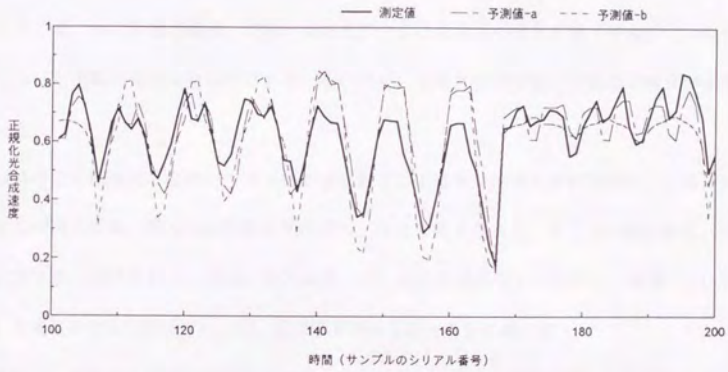


図4-1-2 2種の非線形モデル（ニューラルネット）による予測結果

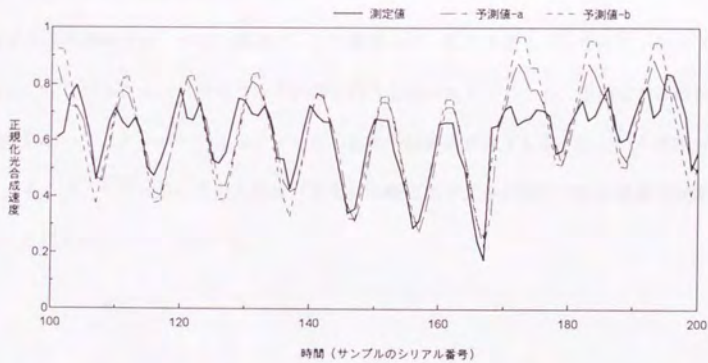


図4-1-3 2種の線形モデル（重回帰モデル）による予測結果

4. 5 カオス性の実験的検証

「4. 4 カオス性の解析」で用いられたデータはカオスの発生を全く予期していない段階で行った実験で得られたものである。そのため、カオス性の評価には複雑な解析が必要であった。

植物の光合成速度の変動がカオスになる原因として植物体内部の要因以外に、①環境条件の変化間隔の変動、②水耕液温度変化の遅れ、なども考えられる。そこで水耕液温度、光強度、気圧を一定に維持し、温度、相対湿度、CO₂濃度を規則正しいステップ間隔（30分、60分）で変化させる実験を行い、①、②の影響があるかどうかを調べた。

最初に、単純な3種類の環境変化パターンに対して、光合成速度のゆらぎがどのように発生するかを実験的に調べた。また、各実験において観測されたゆらぎを予測するモデルを作成し、それぞれの予測精度を比較した。

カオス系の制御では、少ない観測データで精度の高い短期予測モデルを逐次、生成（または修正）しながらフィードフォワード制御を行う必要がある。しかし、予測モデルのシステム同定において、オーバーフィッティングのため予測精度が低下しやすいという問題が見られた。そこで、モデルのシステム同定（学習）の際にカオス系の特性である軌道不安定性を考慮した同定法について考察する。

4. 5. 1 実験方法

光合成速度のカオスを観測するためには、一個体の植物だけを用いて計測する必要がある。また、光合成速度に影響を与える環境因子が計測されていなかったり、ゆらいでいたりすると、光合成速度のカオス性の判定や区別が難しくなってくる。そこで、気温、相対湿度、CO₂濃度、光強度、気圧、水耕液温度、水耕液濃度、水耕液水位を独立に制御でき、様々な

4 生長の複雑さに関する実験と予測モデル

環境変化パターン下で植物1個体の光合成速度（みかけの光合成速度）、生長速度（莖径の増加速度）、葉温などの変化を計測する実験システムを開発し、以下のような実験を行った。実験に用いた植物は前節で用いたものと同品種のメロン苗（草丈：約30cm）である。

【実験1】

温度、相対湿度、CO₂濃度を周期的に変化させ、光合成速度などを連続的に計測した。このとき、光合成有効量子束密度（PPFD: Photosynthetic Photon Flux Density = 1300 μ mol/m²/sec）、気圧（0.1MPa）、水耕液温度（21°C）、風速（約0.2m/sec）は一定に保った。水耕液は、常時、一定温度を保ちながら、一定流量で循環させた。水位は、蒸散に伴って減少するため、水位が一定になるように水耕液供給量を制御した。同時に、水耕液のEC（電気伝導度）、pHを計測し、これらも一定の範囲に収まるよう、適宜、水耕液を交換した。水耕ユニット部分では、チャンバー内の空気を一定量の流量で多孔質フィルター通して微細な気泡にし、水耕液と攪拌して酸素を供給した。環境変化のパターンは離散的な三角波とし、表4-2に示す周期で変化させた。全実験時間は約500時間（測定したサンプル数：499個。60分間隔で環境条件をステップ状に変化させ、1サンプルを得た。）である。

表4-2 環境変化のパターン

環境要因	周期[hour]	変化範囲
温度	5	18 ~ 35 [°C]
相対湿度	20	35 ~ 85 [%]
CO ₂ 濃度	80	300 ~ 1800 [ppm]

【実験2】

3種の環境要因に対して実験1の1/2の周期で変化させた。全実験時間は約230時間（サンプル数：447個。30分間隔で環境条件をステップ状に変化させ、1サンプルを得た。）である。

【実験3】

実験2と同様の条件で相対湿度を40%、CO₂濃度を380ppmに固定し、温度のみを周期的に変化させる実験を行った。ただし、一部の光源ランプの断線事故のため、PPFDは、実験1、2と異なる450 μmol/m²/secで一定にした。全実験時間は約50時間（サンプル数：101個、30分間隔で環境条件をステップ状に変化させ、1サンプルを得た。）である。

4. 5. 2 実験結果

実験1、2を比較すると、ゆっくりした環境変化（実験1）の方が、速い環境変化（実験2）よりも光合成速度の変化が複雑であり（図4-14(a), (b)）、アトラクタも複雑であった（図4-15(a), (b)）。温度だけを変化させた実験3でも光合成速度の複雑なゆらぎが観察された（図4-15(c)）。測定されたサンプルから計算した実験1、2、3のアトラクタの相関積分を図4-16に示す（ $d=2, 3, \dots, 10$ ）。

図4-16の実験1のグラフ(a)、実験2のグラフ(b)では、埋め込み次元 d の各グラフにはどれも直線に近い領域があり、その領域のグラフは平行に近い（ d が大きくなるにつれてグラフの間隔が狭まり、傾きの大きさが近づいている）。ただし、サンプル数の制約とデータに重畳するノイズのため d を極端に大きくして傾きの取束を直接確認することは困難である。図4-16のグラフ(a)では $r < 0.2$ 、グラフ(b)では $r < 0.1$ の領域でグラフが折れ曲がり傾きが広がっているが、これはノイズの影響と考えられる。

また、実験3のグラフ(c)はサンプル数が少ないため直線部が、多少、曲がっているが、全体として直線とみなすことができ、グラフはほぼ平行になっている。図4-16のグラフの直線部分の傾きから求めた相関次元は、それぞれ 2.7 ± 0.9 、 1.7 ± 0.4 、 1.5 ± 0.5 であった。

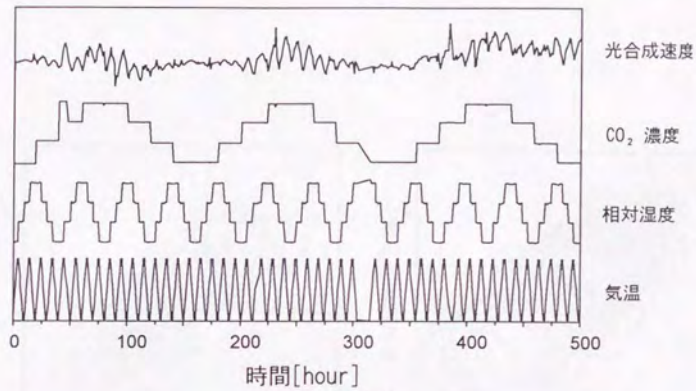
環境変化が遅い方が応答が複雑になるというのは、一見、不自然に思えるが、これは環境

4 生長の複雑さに関する実験と予測モデル

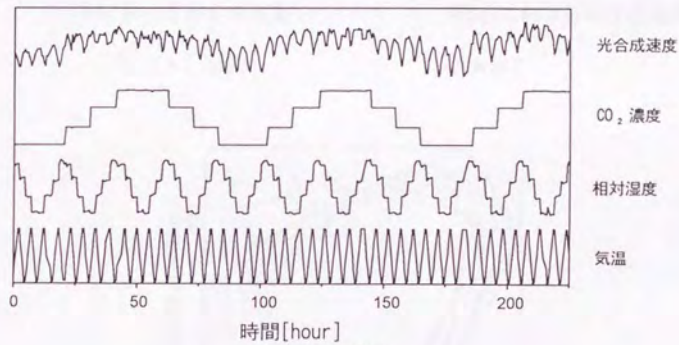
条件が変化していない期間（実験 1 では 1 時間，同じ環境が持続している）でも，光合成速度は変動しているためと考えられる．環境変化が遅いと，環境変化に追従しようとする応答の成分以外のゆらぎの影響が相対的に大きくなると解釈できる．自然環境では，環境変化パターンの周波数スペクトルは非常に広いレンジにわたるため，光合成速度は常時，非常に複雑なカオス状態にあると推測される．

カオスの発生要因として，植物体内部のメカニズム以外に根圏部における温度勾配による遅延や，根周囲における水耕液成分や溶存酸素などの濃度勾配による遅延が考えられる．しかし，1～3の実験では，水耕液は常時循環し，エアレーションによって水耕液には酸素が供給されている．そのため，根圏部での温度や酸素濃度の勾配や変動は少なく，これらの影響はほとんどないと考えられる．したがって，光合成速度のカオスは，植物体内部での光合成産物の蓄積や転流などの生理的メカニズムによって発生していると推測される．

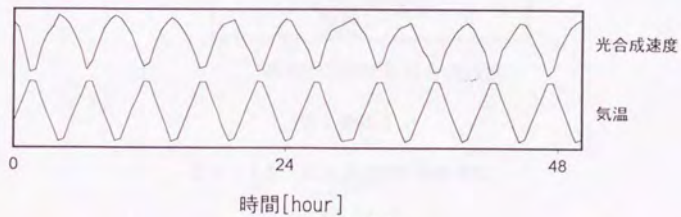
4 生長の複雑さに関する実験と予測モデル



(a) 実験 1



(b) 実験 2



(c) 実験 3

図 4-1-4 周期的環境変化パターンに対する光合成速度の応答

4 生長の複雑さに関する実験と予測モデル

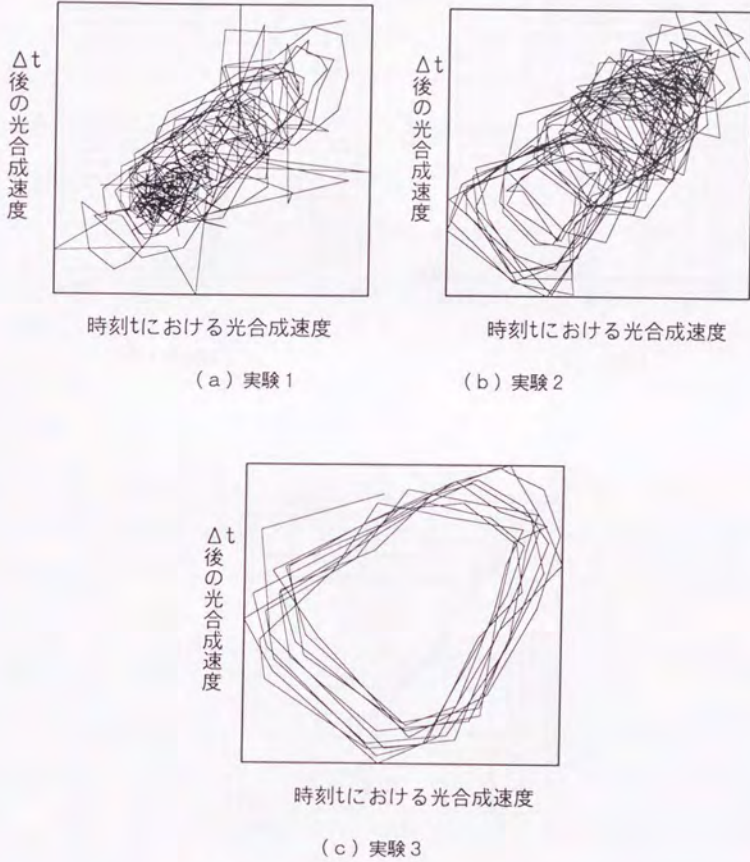
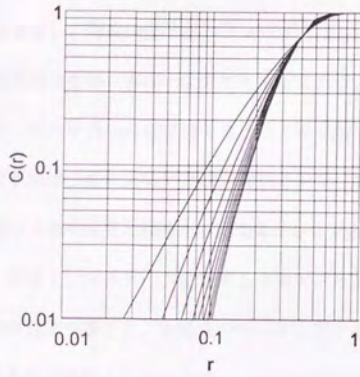
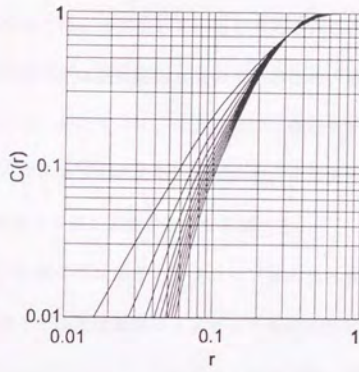


図 4-15 光合成速度の相軌道図

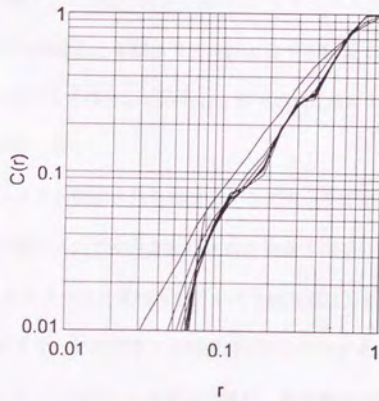
4 生長の複雑さに関する実験と予測モデル



(a) 実験 1



(b) 実験 2



(c) 実験 3

図 4-16 光合成速度の相関積分

4. 5. 3 光合成速度の予測モデル

光合成速度のカオスは、環境変化に対する応答の非線形性と過去の履歴依存性（時間遅れの影響）、個体内部の諸パラメータの相互依存性などによって生じていると考えられる。光合成速度のゆらぎがカオスであるとすれば、そのゆらぎは決定論的なものであるため、適当なモデルを用いれば予測できることを示唆している。そこで、4. 4と同様に、線形モデル（重回帰分析による）及び非線形モデル（ニューラルネットによる）を用いて、1ステップ前の光合成速度と現時点の環境条件から光合成速度の予測が可能かどうかを調べた。

実験1, 2のデータに対する予測モデルでは、3種の環境因子と1ステップ前の光合成速度を説明変数とし、実験3のデータに対する予測モデルでは温度と1ステップ前の光合成速度を説明変数とした。実験1, 2の非線形モデルに用いたニューラルネットは入力層-中間層-出力層のユニット数が4-8-1の3層、実験3のモデルは2-8-1の3層の階層型ネットワークである。

図4-17(a), (b)は、実験1, 2の後半の100個のサンプルをテスト用として除外し、残りのサンプルを用いてシステム同定した2種のモデルによる予測結果である。また(c)は、実験3の後半50個のサンプルをテスト用として除外し、前半の51個のデータを用いて同定した2種のモデルによる予測結果である。

相関次元の低いアトラクタが観察された実験2, 3のデータではニューラルネットの予測精度は高く、相関次元の高いアトラクタが観察された実験1のデータでは予測精度が低い。

図4-17(a)では、ニューラルネットと重回帰モデルの予測誤差はどちらも大きい。ある部分ではニューラルネットが小さく、別の部分では重回帰モデルの方が小さい。光合成速度の変化幅が大きい部分では、ニューラルネットの推定が優れ、変化幅が小さく平坦な部分では重回帰モデルが優れているように見える。

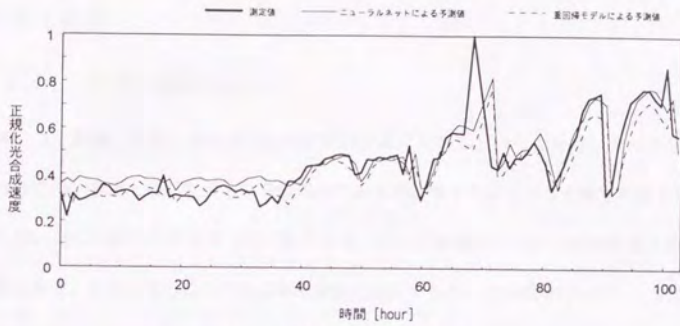
実験2で測定された光合成速度の変化には、4. 3. 1の実験（図4-6）で見られたものと

4 生長の複雑さに関する実験と予測モデル

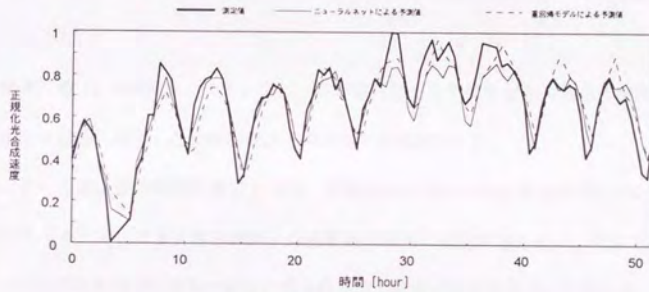
同様、複雑な変動が見られるが、4. 4. 3で行った予測結果の比較(図 4-12, 図 4-13)と同様、この光合成速度のゆらぎを重回帰モデルは予測できていない(図 4-17 (b))。このことから、このゆらぎはカオスと考えられる。

実験 1 に対するモデルのシステム同定では、学習の打ち切り誤差に対して非常に敏感であり、過学習によるオーバーフィッティングの悪影響が大きく見られた。カオス系では推定誤差が急激に増大するため、最小 2 乗法によるモデルのシステム同定では、この性質を考慮する必要があると考えられる。

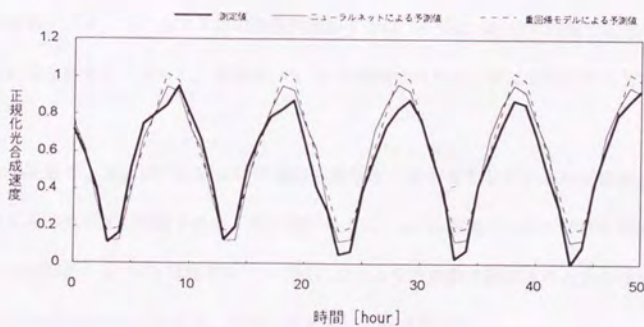
4 生長の複雑さに関する実験と予測モデル



(a) 実験1のデータの予測



(b) 実験2のデータの予測



(c) 実験3のデータの予測

図4-17 線形モデル及び非線形モデルによる光合成速度の予測

4. 6 考察

4. 6. 1 光合成速度のカオス

「4. 3 実験」では、光合成速度の応答がカオスであるということは、全く予想していなかったため、実験を通して得られる情報をできるだけ多くするという方針で実験を行った。そのため、同じ実験時間でできるだけ多くの測定サンプルを得るため、環境状態や光合成速度がある程度、安定したところで次の環境条件に設定するという実験を行った。したがって、環境条件を数 10 分間隔（最頻値で 30 分）で変化させたときの光合成速度の後半 10 分間の、ほぼ安定した状態での平均値が観測されている（それより短い時間スケールでの応答成分は無視している）。

その結果、数 10 分間隔を 1 ステップとした時間スケールで見たとき、光合成速度には環境条件と過去の状態に依存した比較的低次元のカオスが観測された。

「4. 5 カオス性の実験的検証」では、環境変化に対する光合成速度の応答がカオスであるという「4. 4 カオス性の解析」の結果を実験的に確認するために、できるだけ単的なバターン的环境変化下に植物を置き、光合成速度の応答の複雑さについて調べた。

「4. 3 実験」では、実時間としては 30 分前後の間隔で環境条件をステップ状に変化させたのに対し、「4. 5 カオス性の実験的検証」では 30 分と 60 分の間隔で環境条件をステップ状に変化させた。さらに、気温だけを 30 分間隔でステップ状に変化させる実験も行った。

いずれの実験でも光合成の応答には非常に複雑なゆらぎが発生しているのが観測され、それらのゆらぎはカオスと判断された。興味深いことに、60 分間隔でステップ状に環境条件を変化させた実験よりも 30 分間隔でステップ状に変化させた実験で観測された光合成速度のゆらぎの方が低次元のカオスとなり、予測しやすいことが分かった。

「4. 3 実験」では、たまたま、ほぼ 30 分間隔でステップ状に変化していたために比較

4 生長の複雑さに関する実験と予測モデル

の単純なカオスが発生し、カオスの存在が確認できたと考えられる。

4. 6. 2 カオスの発生と協調的なメカニズムの存在の可能性

光合成に関与するメカニズムの実体は非常に複雑であることから、物理的な自由度は非常に大きいはずである。ところが観測された光合成速度のカオスはいずれも低次元であり、システムとしての自由度が極めて小さくなっていることを示している。これは、植物内部の機能全体が協調的に動作した結果、自由度が小さくなっているためと考えられる。

一方、「4. 3 実験」における生長速度の変動は低次元のカオスではなかった。これは、生長が光合成とは独立した要因（たとえば光合成産物の転流など）に影響されているためと考えられる。

4. 6. 3 カオス性の応用

一般に、カオス現象は初期値の微小な違いが指数関数的に増大するため長期予測が難しい（もちろん、その大きさはその変数の変域内で頭打ちになる。）。そのためカオス現象では、正確な長期予測を前提とした制御や計画は原理的に困難であることが知られている。

環境変動に対する光合成速度や生長速度の応答がカオスであるということは、いくら精度の高いモデルを用意しても精度の高い長期予測が不可能であることを意味している。すなわち、環境制御操作を数ステップ行った際の先まで見越して光合成速度を厳密に最大化することは困難である。図 4-13 において示したように、線形モデルで予測可能な程度の精度で光合成速度の予測は可能であるが、「光合成速度を増大させようとして行った環境制御操作が逆に光合成速度を低下させてしまう」という事態が時々発生することは避け難い。

4 生長の複雑さに関する実験と予測モデル

しかしその反面、カオスには次のような利点もある。カオス系では軌道が空間的に遍在し、外乱によって元の軌道を外れてもストレンジアトラクタに急速に引き込まれるという特徴がある。このとき、カオス系の一般的性質として、非常に小さな制御量を系に与えるだけで大きな変化を生み出すことができる（非カオス系では大きな変化を生成するために大きな制御量が必要である）。

植物は環境変化に対する適応の手段として、この性質を利用しているとも考えられる。工学的には、カオス性を次のように利用することができる。

(1) 生育動態のモニタリング

植物個体内部の制御系によってカオス性は維持されているため、もしその制御特性が変化すれば相関次元の値も変化することになる。そのため相関次元は、生育ステージによる違い、品種間差、適応変化などによる制御特性の変化を定量的に測定する指標として利用できると考えられる。

(2) 少ない情報量による同定

カオス系ではカオスを生み出すモデルが比較的単純であるため、同定に必要な情報はかなり少なく済む。そして短期予測に限れば（本論文の「4.4.3 非線形モデルの同定による検証」や「4.5.3 光合成速度の予測モデル」の例では約30分後）、同定したモデルを使って精度の高い予測を行うことができる。

(3) 微小な制御量による生育環境制御

微小外力によるカオス系の制御は倒立振り子、熱対流、心臓組織の拍動などで実験的に検証されているが^{101,102)}、植物でも同様に環境条件の小さな変化によって生育状態を制御できると考えられる。

4 生長の複雑さに関する実験と予測モデル

植物生長は長期間にわたる変化であるため、極端に微小な刺激でも大きな影響が見られる場合もあると思われる。その下限としては、熱ノイズレベル以下の微弱電場が細胞に影響を与えるという報告があり¹⁰³⁾、非常に弱い刺激が植物の生長パターンに影響を与える可能性がある。ただし、カオスの性質上、積極的な制御を行わずに電磁場を印加した場合には、効果が大きくばらつくと予想される。電磁場が植物生長に与える影響は古くから報告されているが、実際、各報告ごとにその効果がまちまちであることが指摘されている¹⁰⁴⁾。

4. 6. 4 複雑さとあいまい性

植物の挙動は複雑で捉えにくいあいまいなものと直感的に考えられているが、このあいまいさはカオスとランダムネスに分けて考える必要がある（決定論的な系においては、ランダムネスは獲得情報の不足に他ならない）。

カオス時系列データの分布は特異な構造を持っているが、局所的に観察すると単純かつ稠密であるため、ランダムデータと区別し難い。そのため、もしカオス現象をランダム現象と見誤ると、カオスを発生させている決定論的メカニズムに関する十分な情報が獲得されているにも係らず、「得られたサンプルには未知の要因が影響している」とか「測定時に大きなノイズが混入して十分な情報が得られなかった」などと判断されることになる。

一般に生物現象に関するデータは測定時に大きなノイズが混入することが多く、測定した現象がカオスであったとしても、測定したデータには大きなノイズが重畳している可能性は常にある。しかし、そのカオス現象をランダム現象と決めつけると、サンプルに付随するあいまい性を不必要に過大評価することになる。また、カオスを生み出している単純なメカニズムを発見するチャンスを失うことになる。

4. 7 結論

環境条件が変化したときに見られる植物個体の光合成速度の複雑な変化がカオスであることを明らかにした。さらに、光合成速度の複雑な変化が決定論的な非線形モデルによって再現及び予測できることを示し、そのカオス性を検証した。

従来、「植物の挙動にはあいまい性がある」とか「栽培プロセスにはあいまい性がある」と言われて来たが、このあいまい性の一部は植物の応答のカオス性に起因すると考えられる。ここでは、その表現方法及び応用方法を考察した。

5 議論 -モデルの一般化について-

ニューラルネットは脳のモデルを超えて広く非線形系のモデルとして利用することができる。同様に、本研究で提案したモデルについても植物の生長モデルを超えた用途の可能性がある。ここでは、本研究で開発した生長モデルの一般化と生物系の一般モデルについて議論する。

5. 1 器官ネットワークモデルの一般化

5. 1. 1 モデルの汎用性

器官ネットワークモデルは多数の隠れユニット（「観測値とユニットの出力値が一致する」ように拘束されないユニット）を持つことを許容すると、ニューラルネットと同様、任意の連続関数を近似する能力を有している。これは器官ネットワークモデルの時間パラメータを定数にすればリカレントネットワークに一致し、さらに接続を一部切断するとフィードフォワード型のニューラルネットに一致することからいえる。

また、器官ネットワークモデルの元になった Lotka-Volterra 方程式も任意の連続関数を近似する能力を有している。図 5-1 に示すようなパラメータを持つ 2 つのユニットのペアからは、 \sin 関数と \cos 関数を主要な成分とする周期関数が出力される。このようなユニットが多数隠れユニットとして存在すればその和が入力されるユニットの入力値はフーリエ級数展開となるため、任意の連続関数が合成されることになる。したがって、隠れユニットを持つ Lotka-Volterra モデル (LV モデル) は、非線形系の一般化モデルとして利用できる。

この LV モデルはニューラルネットのアナログモデルよりも簡単な電子回路で構成することができるといって特徴を持っている。そのため大規模化や高速化が不可欠である自然条件における植物群落の生長モデルや生物系の一般モデルに適していると考えられる。

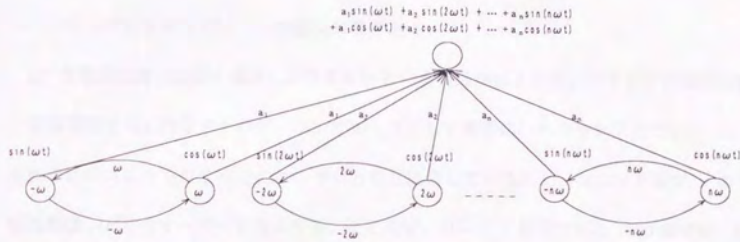


図 5-1 sin 関数と cos 関数を発生するユニット出力値の重み付け総和による任意連続一価関数の生成 (フーリエ級数展開)

5. 1. 2 ハードウェア化の可能性

LV モデルを広く利用するためには実行速度の高速性が問題となる。高速化するためにはハードウェア化を行うのが一つの解決方法である。

LV 方程式は積和演算と乗算しか含まないという特徴があるため、アナログの積和演算回路と乗算回路を用いればアナログ・コンピュータとして簡単にハードウェア化できる。しかし、それぞれのユニットがすべてのユニットと相互結合しているため、ユニット数が n のとき、結線数は $o(n^2)$ のオーダーで増大する。たとえば、ユニット数がたった 1000 個でも、結線用の配線が 200 万本にもなってしまう。ハードウェア・ニューロコンピュータの場合と同様、結線数の爆発が最も重大なボトルネックとなっている。

配線数を減らすためには、CATV などの有線通信で行われているように信号の多重化が利用できると考えられる。各ユニットの出力を交流にして各ユニットの値を異なる周波数に割り振ると、1本の線（信号バス）にすべての信号を載せることができる。すなわち、ユニット i の出力値を $x_i \sin \omega_i t$ としたとき、1本の信号バスに、

$$e = x_1 \sin \omega_1 t + x_2 \sin \omega_2 t + \dots + x_n \sin \omega_n t \quad (5-1)$$

の信号を載せて同時に伝送することができる。ここで、ユニット i への他のユニットからの作用係数も同様に、

$$W_i = w_{1i} \sin \omega_1 t + w_{2i} \sin \omega_2 t + \dots + w_{mi} \sin \omega_m t \quad (5-2)$$

の信号で供給すると、結線数の増大は $o(n)$ のオーダーに抑えることができる。しかも、 e と W_i を乗算回路で乗算すると、直流成分（搬送波 $\sin \omega_i t$ よりもゆっくりと変化する成分）は、

$$e W_i = w_{1i} x_1 + w_{2i} x_2 + \dots + w_{mi} x_m \quad (5-3)$$

となり、積和演算がリアルタイムで一度にできる。

ちなみに、異なる周波数を用いて結線数を減らす方法はアナログ多重通信の基本技術の一つであり、ニューラルネット（学習しきい素子）のハードウェアに対しては、横井らが上記

と似た方法を試みている¹⁰⁵⁾。ただし、これは LV モデルに比べて回路が非常に複雑であり、適用用途も LV モデルよりも限られている。そのため、ハードウェア化を行うメリットは LV モデルの方が大きいと考えられる。

LV モデルを、この方法でハードウェア化した時のブロック図を図 5-2 に示す。また、具体的な回路の設計例を付録 B に示す。

各ユニットを一つのチップ（またはワン・ボード）で作成すると、ユニット出力値信号を出力するピンをバスに接続するだけで LV モデルが構成できる。各チップには、それぞれユニットへの重み係数信号を供給する必要がある。これは一定の波形パターンを持った信号であり、任意波形発生器などを使って外部から供給すれば良く、他のチップと接続する必要はない（チップの面積に余裕がある場合は、チップ内部で発生させても良い）。

この方法が通常の数値積分と同じ結果を得られるかどうかを調べるため、30 個のユニットからなる LV モデル（諸パラメータは乱数で設定）を用いて比較した。図 5-3 のシミュレーション結果が示すように、両者の結果は完全に一致している。

ユニット1への重み係数係数

$$w_{1, \sin(\omega_1 t)} + w_{2, \sin(\omega_2 t)} + \dots + w_{n, \sin(\omega_n t)}$$

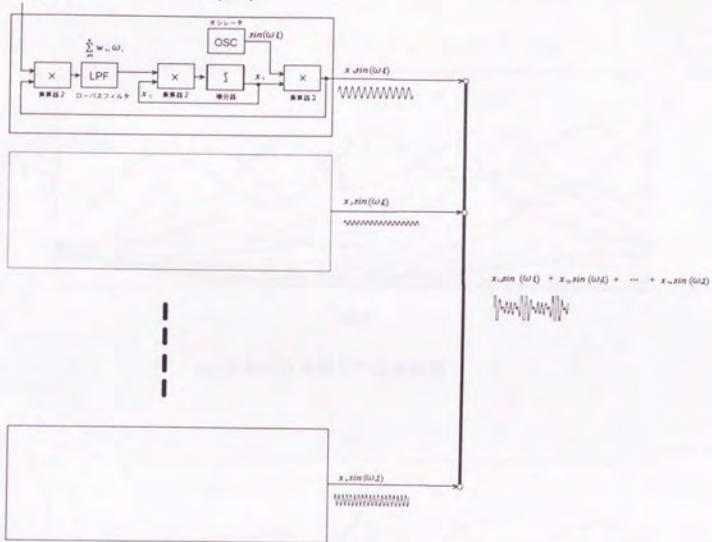
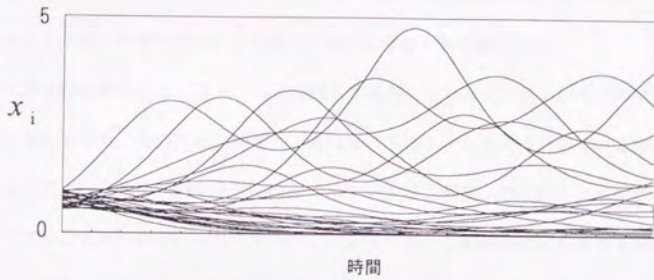
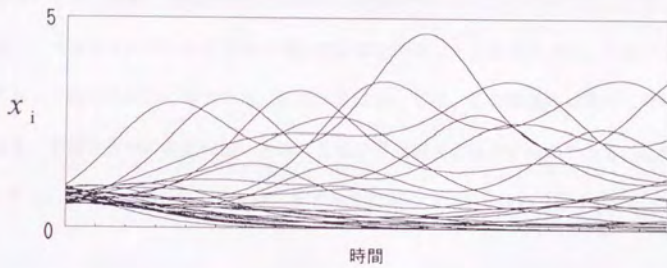


図5-2 LV方程式を計算するアナログ回路のブロック図



(a) 通常の数値積分による計算



(b) 提案したハードウェアによる計算 (シミュレーション)

図5-3 通常の数値積分による結果と提案した方法の結果の比較

(ユニット数: 30, w の範囲: 0~10, r の範囲: -10~10)

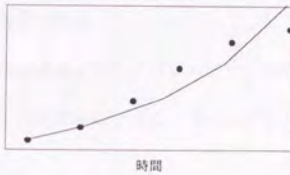
5. 2 カオス系のモデルのシステム同定

植物の生長において、最も重要なパラメータである光合成速度のゆらぎがカオスによるものであることから、植物生長モデルではこの効果を考慮する必要がある。

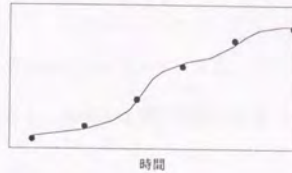
カオス系では時間とともにファジネスが増大するが、これは真に正しいモデルであっても不可避なものである。そのため、時間とともに増大するファジネスは許容すべきであると考えられる。従来のシステム同定手法ではシミュレーション時間全体にわたる誤差の総和を最小化するアルゴリズムが用いられてきたが、上記の考察から初期値設定時点から離れるほど小さいペナルティーを与える評価関数を用いる必要がある。そこで、以下のようなシステム同定方法を考案した。

植物のサイズ（重量）の時系列データが不規則な間隔でサンプリングされているとき、カオス系では、1つ前のサンプルの状態から現在の状態を予測したときの誤差は、予測する時間の間隔に対して指数関数的に増大する。カオス系においては、この誤差は必然的に発生するものであり、予測モデルの精度が悪いために生じる予測誤差とは別のものである。この誤差を減らすためにモデルを修正することは、モデルの改善につながらない（図5-4）。

○ 短期間の予測誤差を最小化



× 全期間の予測誤差を最小化



◎ 短期間の予測誤差を最小化

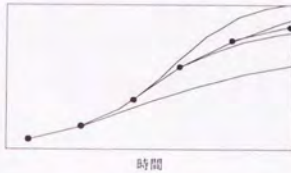


図5-4 カオス系モデルに対する最小2乗法によるシステム同定の考え方

たとえば、第4章の光合成速度の予測モデルにおいて、相関次元の低いアトラクタが観察された実験2, 3のデータではニューラルネットの予測精度は高く、相関次元の高いアトラクタが観察された実験1のデータでは予測精度が低かった。この結果からも分かるように、カオス系では推定誤差が急激に増大するため、最小2乗法によるモデルのシステム同定では、この性質を考慮する必要があると考えられる。

そこで、ある時刻 (t_1, t_2, \dots) における観測値 (x_1, x_2, \dots) と、その予測値 ($f(x_1), f(x_2), \dots$) に対して、上記の考え方を適用すると、以下のような関数の最小化を行うリカレント・ニューラルネットの学習アルゴリズムが考えられる。

$$E^2 = \sum_{h=1}^j \sum_{i=j+1}^n e^{-\lambda(t_i - t_j)} (f_i(x_j) - x_i)^2 \quad (5-4)$$

ここで λ は定数(最大リアプノフ数), $f_i(x_j)$ は時刻 j における観測値 x_j を用いた時刻 i の予測値である.

(5-4)式は, バックプロパゲーション・アルゴリズムで容易に利用できる. また, 「3. 3.

1 同定アルゴリズム」で示したアルゴリズムにおいても, (5-4)式は適用可能である.

5.3 生物の階層性を考慮した生物系の一般モデル

生物の生長に関わる現象は、一般的な生物の階層性から以下のように分けることができる。
ミクロレベルでは、

- ・細胞内外の拡散的分子運動（分子の拡散的移動，速度論的な化学反応など）
- ・分子レベルの量子力学的運動（DNA 転写，タンパク質合成，イオンチャネルの開閉，酵素反応など巨大分子系の量子力学的運動）

これよりも，もう少しマクロのレベルでは、

- ・古典力学的運動（細胞内外での物質交換など）
- ・量子力学的運動（細胞のくびれ運動，膜と細胞内骨格間の秩序的運動など）

これらのミクロ的な運動が融合して，さらにマクロレベルでは、

- ・細胞伸張
- ・細胞分裂
- ・組織の形態形成
- ・器官分化

さらに群落レベルでは、

- ・生態系の動的変化

がある。

植物の生長では，上記のミクロレベルの運動とマクロ的な変化が相互に影響を及ぼし合いながら時間発展し，個体の自己組織化が行われていると考えられる。当然のことながら生態系における種の個体数変化は LV モデルで表現できるが（元々，LV モデルは生態系のモデルである。），本章の始めに考察したように物質移動に伴う任意の量的な変化や化学反応なども LV モデルの枠組みの中で表現できる。

したがって，ミクロレベルでの分子動力学，量子力学的運動を除くと，ほとんどの変化は、

原理的に LV モデルで表現できると考えられる。すなわち、LV モデルは生物の持つ生物らしさの挙動を簡略化して表現し、コンピュータ上で生物現象に関する計算実験を行うための『モデル生物』的モデルとして利用可能である。

さて、生態系の動的変化にはカオス性があり、また「4 生長の複雑性に関する実験と予測モデル」で見たように個体レベルでもカオス性がある。組織や細胞の増殖のレベルにおいても、環境に対する応答には同様にカオス的な複雑さがあると予想される。そのため、生物系には各階層においてそれぞれの法則によって生成されたカオスがあると考えられる。

細胞、組織、器官、個体、生態系の各階層は相互に大きな影響を与えているが、微小な摂動が急速に拡大するというカオスの性質から、生物系は各階層で秩序を保ちながら上下の階層の変動の影響を増幅して伝達するシステムになっている。各階層におけるカオスの相互作用がシステムとして、どのような意味をもっているのかを調べるためには、単純化された各階層のモデルが必要である。LV モデルの汎用性から各階層はすべて LV モデルで表現することができるため、原理的には全体を一つの LV モデルで表現することができる。

各階層のモデルを LV モデルで表現すると、図 5-5 のようなモデルになる。下の階層ほど構成要素のサイズは小さく、時間的な変化は速くなる。

細胞の層の下には DNA やイオンチャンネルなどの巨大分子が生成する秩序を持った層がある。このような巨大分子系の単純なモデルにはスピン系の Ising モデル¹⁰⁰⁾がある (Ising モデルは Hopfield モデルと多くの共通点を持ち、LV モデルの特殊なケースとして表すことができる。)

ところが、このスケールは物理学においては量子論的效果が現れ始める世界である。0.1~1 μ m 程度以下のスケールの系は物理学においてはメソスコピック系と呼ばれ、量子力学的な効果が顕著に現れてくることが知られている。生物のモデルでは最下層におけるメソスコピック系の量子論的なメカニズムを考慮したモデルが重要であると考えられる。この効果を

考慮するためには巨大分子系に対する Ising モデルや LV モデルは利用できない。

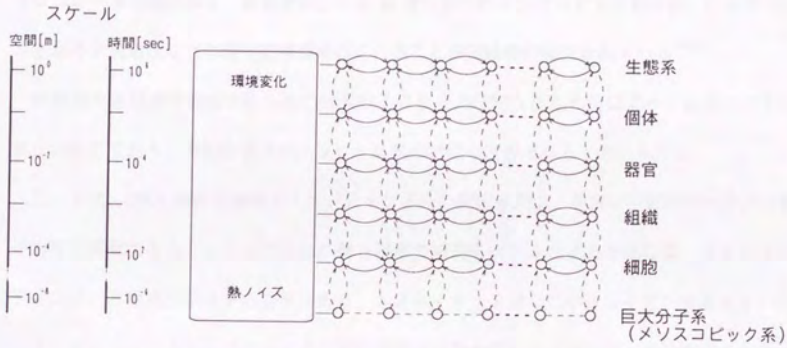


図 5-5 LV 方程式を使った生物系の階層モデル

5. 4 マクロ系とメソスコピック系の特徴を備えたモデル

5. 4. 1 メソスコピック系のモデル

細胞の内部では、DNA やイオンチャネルなどの巨大分子が生み出す秩序（情報）を持ったミクロレベルの層がある。最近では、 $1\ \mu\text{m}$ オーダーのメソスコピック系でも、ジョセフソン効果や永久電流などの量子効果が現れていることが実験的に確認されている^{107,108}。

細胞内や細胞膜の表面では少数の巨大分子の量子力学的な相互作用によって非常に大きな変化が生じており、典型的なメソスコピック系の物理の世界であると考えられる。

たとえば、DNA の転写過程やイオンチャネルの開閉運動は、非常に少数の巨大分子の量子力学的運動であり、これらの現象に伴う特有の機能をモデル化するためには、系をシュレディンガー方程式で記述する必要がある。大きなイオンを通して小さなイオンを通さないというイオンチャネルに特有のイオン選択機構は古典力学的なメカニズムでは説明できないものであり、分子軌道法を用いた計算で初めて理解できる¹⁰⁹。また、1分子のDNAから情報を読みとる機構や遺伝子の発現過程は、速度論的な議論（化学反応における質量作用の法則）が適用できない量子力学的分子運動である。

ちなみに、多数の分子が関与する化学反応であっても、その本質は量子力学的運動である。たとえば、適当な複数の波長のレーザー光を組み合わせると、古典的化学反应論では説明不可能な経路の化学反応を発生させることができる^{110,111,112}。

メソスコピック系の層で発生した量子的ゆらぎは、たとえ微少な変化であっても上層がカオス的な変化をしていると、増幅されて大きな影響をマクロ系に与えると考えられる。また、上層（マクロ系）からメソスコピック系への影響も、当然、存在すると考えられる。これがDNAの配列の変化を引き起こすと、再びマクロ系の大きな変化を発生させる。このメソスコピック系とマクロ系の相互作用による再帰的ループは、何らかの自己組織的な現象を生じさせていると考えられる。

ミクロレベル（メソスコピックレベル）での量子的効果が生命現象の本質に関わっているという考え方は、多くの研究で指摘されている¹¹³⁾。生物系においては、代謝系におけるポーズ凝縮^{114,115)}や、タンパク質のチェイン形成¹¹⁶⁾、細胞間の相互作用¹¹⁷⁾、血液細胞表面の粘性¹¹⁸⁾、DNAからの生物フォトン放射^{119,120)}、意識¹²¹⁾や記憶^{122,123)}などの脳機能に対して、量子論を適用した議論がある。

さらに、マクロレベルの生物現象一般に対する量子論的な研究や論説は、古くはボーア、ハイゼンベルグ、シュレディンガー、ノイマンといった量子力学の創設者らの考察から始まって、最近ではホイーラーやペンローズの『人間原理』に至るまで多数ある^{124,125,126)}。しかし、これらの多くは理論的な議論に留まり、数値実験すら困難なものが多い。

そこでここでは、これまでのモデリングに関する議論に基づいて、コンピュータ上で数値実験可能な生物系のモデルを考えてみる。

5. 4. 2 メソスコピック系の特徴を持つ生物モデル

生物系を構成する階層の最下層が量子的ゆらぎを持った運動を行い、その上の層がカオスによる古典力学的運動を行う世界であるとしたとき、生物のモデルはこの両者を結合して与えられる。

DNAのある領域の情報の読み出しは複数の巨大分子の相互作用の結果によるものである（以下の議論は、DNAの情報発現やイオンチャンネルの開閉でも同じである。そのため、以下では「DNAの情報の読み出し」で代表する。）。そこで、巨大分子系を構成する粒子の座標を $x_i (i=1,2,\dots,n)$ とすると、この変化は、シュレディンガー方程式を用いて以下のように表される。

$$i\hbar \frac{\partial}{\partial t} \psi(t) = (H(t, x_1, x_2, \dots, x_n) + V(t, \psi(t))) \psi(t) \quad (5-5)$$

ここで、 H は巨大分子系を構成する粒子の相互作用ハミルトニアン、 V は巨大分子系が上層のマクロ系から受ける影響を表現するポテンシャル・エネルギー、 ψ は巨大分子系の状態を表す波動関数である。ただし、 V は巨大分子系の影響を受けて変化するため、 ψ の関数でもある。

この最も単純なモデルとしては、ポテンシャルエネルギーに複数の極小値を有する系の量子モデルと LV モデルが結合したモデルを考えることができる (図 5-6)。これは、量子モデルの状態が LV モデルの干渉係数 w_{ij} に影響を与え、同時に、LV モデルの状態が量子モデルのポテンシャルエネルギーに影響を与えるというものである。このモデルには、以下のような非常に興味深い性質がある。

$H+V$ を、たとえば図 5-7 に示すような多峰性の曲線 (一般的には超曲面) とし、図の A の近傍を「読み出される」状態 (以下、A 状態)、それ以外の部分を「読み出されない」状態 (以下、非 A 状態) とする。波動関数の確率解釈から、A 状態を時刻 t に観測する確率 $P_A(t)$ は

$$P_A(t) = \int_A \psi(x,t) \psi^*(x,t) dx \quad (5-6)$$

である。

波動関数の確率解釈とは、何度か繰り返すことができる場合、 $P_A(t)$ の確率で A 状態が観測され、 $P_B(t)=1-P_A(t)$ の確率で非 A 状態が観測されるということである。繰り返しが無い場合 (1 度しか観測しない場合) は、A 状態と非 A 状態が同時に存在すると考える必要がある¹²⁷⁾。日常的感觉では A 状態と非 A 状態は背反事象であるから、同時に存在するということは常識的には理解に難い概念である。これは、『シュレディンガーの猫』として有名なパラドクス^{128,129)} (「量子的事象で生死が決定される猫は、生きている状態と死んだ状態が重ね合わせになる。」というパラドクス) であり、波動関数の確率解釈に関して、非常に長い間、論争があった。

しかし、ベルの定理とアスぺの実験¹³⁰⁾から、例え常識的には理解し難いとしても、現実存在する実体はA状態と非A状態の2つの重ね合わせ状態として考えなければならないことが明らかとなっている。もちろん、通常は、 $P_A(t) \gg P_B(t)$ または $P_A(t) \ll P_B(t)$ であるため、『シュレディンガーの猫』のパラドクスを目の当たりにする確率は極端に小さい。

ところが、ここで議論している生物系のモデルでは、マイクロ系の影響を受けて時間発展する上層のマクロ系がカオス性を持っていると、他方（Aまたは非A）の影響が小さいといっても無視できなくなってくる。この効果は、(5-5)式における波動関数 ψ と H 及び V が同時に変化するときの解から調べる事ができる。

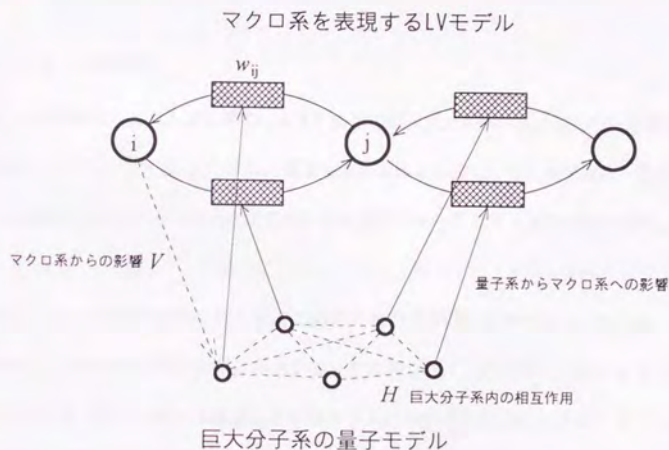


図5-6 量子系を含むLVモデル

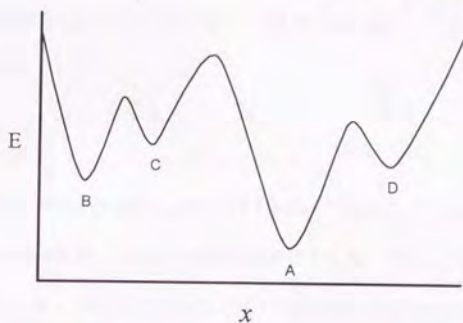


図5-7 量子系のポテンシャルエネルギーの例 (概念図)

5.4.3 数値解法

H と V が時間的に変化しない場合、 $H+V$ の最小値付近で時間とともに $P_A(t)$ が増大する。そのため、このプロセスは、 $H+V$ の大域最小解を求める方法として利用できる。たとえば、Tryba *et al.* は、シュレディンガー方程式のこの性質をニューラルネットの学習に利用している¹³⁾。

ただし、(5.5) 式を数値的に解くためには膨大な計算時間が必要である。Tryba *et al.* は、シュレディンガー方程式の計算が並列プロセッサに容易にインプリメントできることから、逆にシュレディンガー方程式を最適化アルゴリズムに利用する方法の大きなメリットとして挙げている。実際、この方法は未知パラメータ数が少数 (2, 3 個まで) の場合には、実用的な水準にある。同様に、LV 系の未知パラメータの数が少ない場合には、そのパラメータ推定に (5.5) 式を用いることができる。そこで、この過程を数値計算で調べてみよう。

例題として、2つのユニットを持つ LV モデルの1つの作用係数 w を未知として、この作用係数がそれぞれのユニットの出力が正弦波 ($0.5+0.5 \sin t$) と余弦波 ($0.5+0.5 \cos t$)

になるようにする問題を考えてみる (図 5-8)。(5-5)式において、ハミルトニアン H を 1 粒子の運動エネルギーとして、

$$H = -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial w^2} \quad (5-7)$$

とする。ここで、 \hbar はプランク定数 (1.055×10^{-34} J·sec) である。ここで、未知の w の値を (5-5) 式を使って求めるため、 w を (5-5) 式の空間座標とした。また、上層のマクロ系からの影響を表すポテンシャル・エネルギー V をユニットの出力値と目標値との 2 乗誤差として、

$$V(w) = \int_0^{2\pi} (0.5 + 0.5 \sin t - x_1(w, t))^2 + (0.5 + 0.5 \cos t - x_2(w, t))^2 dt \quad (5-8)$$

とする。 $m = 1.66 \times 10^{-24}$ kg, 時間の単位を [μ sec], 長さの単位を [nm], エネルギーの単位を [eV] で表現すると、(5-5) 式は、

$$i \frac{\partial \psi(w, t)}{\partial t} = (-32 \frac{\partial^2}{\partial w^2} + 1.5 \times 10^9 V(w)) \psi(w, t) \quad (5-9)$$

となる。この式を数値計算で解くと、正解の値 ($w=2$) のところで波動関数の絶対値 (確率密度) が急速に増大することが確認できる (図 5-9)。

この方法には、エネルギーの局所最小状態 (局所最適解) でトラップされていてもトンネル効果によってバリアーを乗り越えてエネルギー最初状態 (大域最適解) を見つけ出すことができるというメリットがある。トンネル効果による大域最適解の探索の例として、図 5-10 にポテンシャル・エネルギーの形が多峰性の場合の計算結果を示す。

ところが、求めたいパラメータ数が増えると、(5-5), (5-7) 式が高次元の偏微分方程式となるため、数値計算では計算量が極端に多くなりすぎて解くことができない。

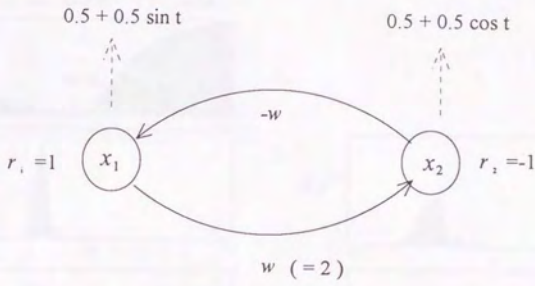


図5-8 w の値が未知の2ユニットのLVモデル

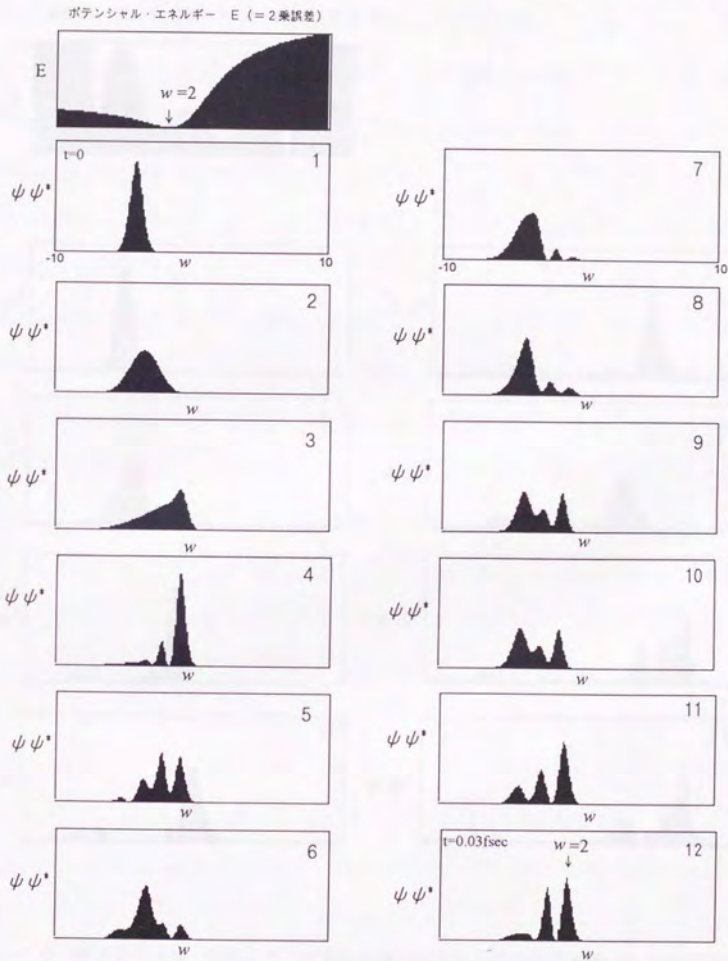


図5-9 シュレディンガー方程式の数値計算によるLVモデルのパラメータ推定

複数の極小値を持つポテンシャル・エネルギー

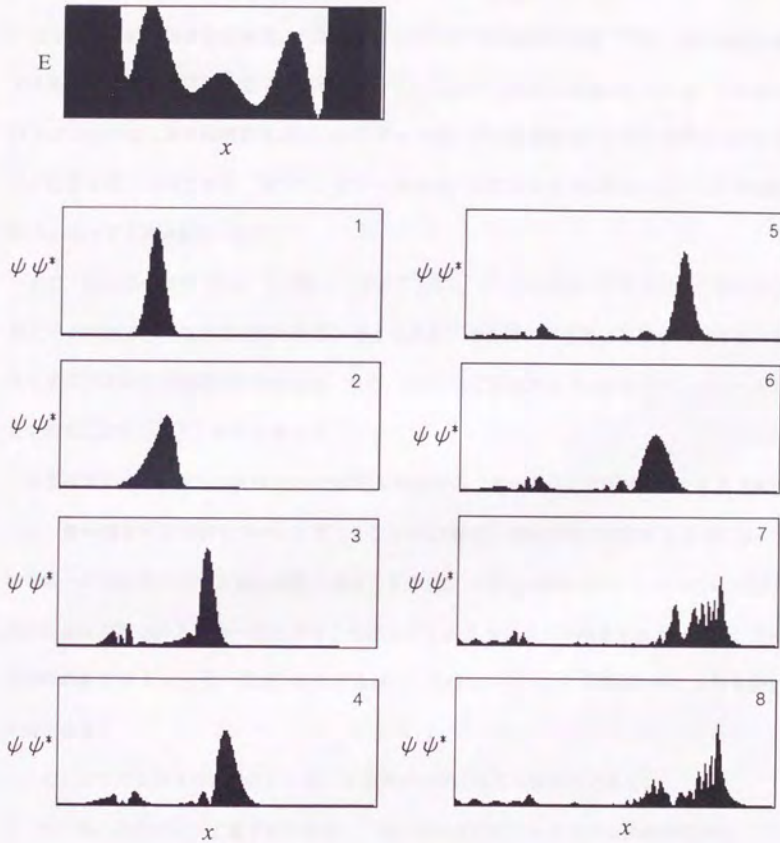


図5-10 ポテンシャル・エネルギーが多数の極小値を持つ場合の計算例。左上の最初のグラフ(1)は初期状態を表す。ポテンシャル・エネルギーの山を乗り越えて波動関数の分布が左に移動し、最後の状態(8番目のグラフ)ではピークがエネルギー最小状態にきている。ただし、これ以降も複雑なカオス的運動を続ける。最小値を正確に求めるには、運動エネルギーを散逸させて定常状態にすれば良い。

5. 4. 4 巨大分子系による最適化計算と定向突然変異のモデル

近年、量子の非局在性を利用した量子コンピュータの理論的可能性^{132,133)}や、弱い結合のある多数の量子系を利用した量子コンピュータ¹³⁴⁾の技術的可能性が議論されている。この量子コンピュータは、波動関数の重ね合わせによってあたかも無限個のプロセッサがあるかのよう計算を行うものである。量子コンピュータを用いれば大域最適化問題のような大規模計算でも実行できる可能性がある。

さて、巨大分子系が(5-5)式に従って運動する際に $H+V$ を最小化する過程は、量子コンピュータの計算とみなすことができると考えられる。そこで、「生物は複雑な環境に適応するために不可欠な大域最適化問題を解くため、DNA などの巨大分子系を量子コンピュータとして利用している。」と考えると見よう。

ある環境条件で、ある特定の DNA の情報が発現することが生長に有利であるとする（あるいは、他の種よりも生存しやすとする。）。この場合、環境条件に適応するためには、 V が環境への適応度に対する減少関数（必ずしも単調減少する必要はない）になっていれば、 $P_A(t)$ は適応度を高める方向へ変化する。ただし、このメカニズムが機能するためには、 $P_A(t)$ の時間的変化が V の変化（環境の変化する速さ）の時間スケールと同程度かそれよりも遅い必要がある。

このモデルで表現されたシステムは、次の点で非常に特異な物理系である。

- ①マクロ系の変化によって量子系が変化し、量子系の変化によってマクロ系が変化する。
- ②量子系がマクロ系の時間的スケールと同程度、あるいはマクロ系よりもゆっくり時間発展する。

通常の物理現象ではマイクロ系のゆらぎはマイクロ系が多数集まった段階で消失し、平均値によって代表される状態（マクロ変数）に収束する¹³⁵⁾。そのため、日常的な現象（マクロ系）では量子系のゆらぎは観測されない。上で述べたような生物系を除くと、①のような量子系

—マクロ系間の相互依存的な現象を見いだすことは難しい。また、通常の量子的現象ではフェムト秒オーダーで変化する非常に速い成分が主要な役割を果たしており、②のように量子系のゆっくりした時間変化の成分によって発生する現象も見あたらない。

さて、上記では H を遺伝子発現として説明したが、巨大分子系のより大きな変化が起こる場合には、DNA 分子自体の構造変化が発生する可能性がある。この場合、(5-5) 式は、環境に適応する方向に DNA 分子の構造が変化する運動のモデルになる。この運動は、生物の進化を表現している。もし、この運動が生じているとすれば、「突然変異が完全にはランダムではなく、環境に対して大域最適の方向に、やや大きな確率を持つ」という現象が存在することが予想される。

このような現象を探してみると、進化生物学の分野において、Cairns *et al.*^{136,137) が発見した定向性突然変異 (Directed Mutation) が非常に似た現象として見いだされる。}

定向性突然変異の存在は、「突然変異は無方向のランダムな変化である」とする伝統的な進化理論 (総合説) に抵触しているため批判もある。しかし、その後、それらの批判に対応した多数の追試や研究が行われ、定向性突然変異の存在は確認されつつある¹³⁸⁾。定向性突然変異は、近年、ますます研究が盛んになっているが^{139,140)}、依然としてそのメカニズムの説明はできていない¹⁴¹⁾。ここで主張する量子論的メカニズムによって定向性突然変異が発生しているとすれば、従来のような古典的メカニズムに還元して説明することは当然難しい。

たとえば、大腸菌は通常の増殖状態で、2つの突然変異 X (β -グルコシドを利用する際に必要な遺伝子 *bgl* の中の介在配列の切り出しを行う突然変異) と、突然変異 Y (*bgl* 調節遺伝子の抑止を解除する突然変異) をそれぞれ 4×10^{-8} と 2×10^{-12} の確率で発生する。突然変異 X, Y を同時に起こす確率は理論的には 8×10^{-20} であるが、突然変異 X, Y を同時に起こさないと生存できないようにした大腸菌は、非増殖状態で X, Y の突然変異を 10^{-8} という高い確率で同時発生させてしまう。突然変異 X が起こった段階では、まだ β -グルコシドを全く利

用できないにも関わらず、突然変異Xを起こす種が1~10%の割合で発生する。これは、図5-7においてエネルギー極小状態(B, C, D)からエネルギー最小状態(A)へ移る際に、目前にあるポテンシャルエネルギーのバリアを乗り越えるというトンネル効果で説明できるだろう。

ただし、ここで提案したモデルは、マクロ系を量子力学的に厳密に扱うことを避けている。厳密には細胞以上の層や環境などのマクロ系も多粒子系の運動として表現し、波束の収縮過程やエネルギーの散逸(摩擦)、マクロ系の非線形性なども全て量子論の枠組みで表現する必要がある。しかし、これらの現象は観測問題などの理論的に難しい問題^{142,143,144}を多数含むため、ここでは量子論的な定式化を行わなかった。

5. 4. 5 生物現象と非生物現象の定義

量子論的運動とマクロ的運動が相互に影響を与えながら環境に適応する時間発展する自己組織系は、非常に特異な物理現象である。逆に、これが生命現象と非生命現象を区別する境界線と考えられる。

量子的現象がマクロ系に一方的に影響を与える系には、たとえば霧箱(荷電粒子の通過を霧の発生で検出する装置)があるが、これは到底「生命現象」には見えない。また、気象現象や原子炉の内部の粒子の運動のようにマクロまたはミクロのレベルだけで起こる現象は、どんなに複雑であっても「生命」があるようには見えない。しかし、(5-5)式に従って運動する巨大分子系とマクロ系は、環境に対して大域最適化を行う自己組織的現象を生じさせるため、極めて生物的な性質を備えている。

ここで示した生命現象の定義が適当かどうかは、DNAやRNAなどの物質を使わずに、(5-5)式で定義した系をコンピュータ上で人工的に作り、その系の振る舞いに対して「生命的な

5 議論 —モデルの一般化について—

振る舞いを感じるかどうか」(チューリング・テスト)で、間接的に調べることができると考えられる。

6 結言

(1) 植物生長の複雑さに関するモデル化

植物の生長では、DNA や細胞レベルでの複雑さとは別に、比較的単純なマクロ的なメカニズムによって生成される複雑さが存在する。本研究で開発した計算モデルは、植物の生長に見られるマクロレベルの複雑さを表現できる。

(2) 環境変化に対する応答のカオス性

数十分程度のオーダーの環境条件の変化に対する光合成速度や生長速度の応答にはゆらぎがあり、このゆらぎの多くの部分はカオスである。短期的な予測であれば、このゆらぎは適当な非線形モデルによって高い精度で予測可能である。

(3) 環境変動に対するカオス的応答のモデル化

本研究で開発された諸モデルは、いずれもカオス的な複雑さを表現することができる。

(4) 計算モデルのプログラムとしての特長

開発された計算モデルはプログラムとして、

- ・獲得した情報に応じて利用できる
- ・ブラックボックス的に利用できる
- ・他プログラムへ簡単にリンクできる

などの特長があり、農業情報ネットワークのホストプログラムなどに簡単に組み込むことができる。

(5) 開発したモデルの発展性

現実の植物ではマイクロレベルとマクロレベルの複雑さが絡み合っていると考えられる。本研究で開発した LV モデルは、マクロレベルの複雑さを表現する汎用のモデルとして利用できる。この LV モデルにマイクロレベルに特有の量子力学的効果を付加することで、生物/生命系を単純化して表現するモデルとなる。これは量子論的運動とマクロ系の運動が相互に影響を与えながら環境に適応する生命現象特有の運動を表現できる。そして、このメカニズムを生物現象と非生物現象を区別する基準とすることを提案した。

謝辞

本研究のかなりの部分は、科学技術庁の振興調整費・総合研究「ファジィシステムとその人間・自然系への適用に関する研究」における「植物生長のモデル化」（第Ⅰ期）、「植物生長のモデリングと改善」（第Ⅱ期）の研究の一環として行われたものである。東京大学農学部の高倉直教授には本研究の実施及びとりまとめに際して御指導を頂くとともに、当プロジェクトの推進においても大変なお骨折りを賜った。また、本研究を開始する前の段階では重要なヒントを頂いた。心から深く感謝する次第である。

蔵田憲次助教授（東京大学）と古在豊樹教授（千葉大学）には叱咤激励と御指導を賜った。深く感謝の意を表する。

本研究の実施に際しては、当時の企画課長である西尾道徳企画連絡部長、桑名隆（前）機械作業部長、小野良孝（前）畑農作業研究室長（現中国農業試験場、地域基盤部長）、唐橋需機械作業部長の御尽力を頂いた。窪田哲夫畑農作業研究室長、小林恭主任研究官には、議論の相手となって頂くと同時に御助言を頂いた。深く感謝する次第である。

4章での前半の実験では、メロンの栽培において小野誠氏（熊本県農業研究センター）に御助力頂き感謝する。

上記プロジェクトのリーダーである浅居喜代治教授（当時大阪工業大学、現福山平成大学）には、御指導・御助言を頂くとともに、放送大学の講義において紹介（放映）する機会や「ファジィ科学」（海文堂）において本研究の成果の一部を発表させて頂く機会を賜り感謝する。

本研究の成果の一部は IFAC のハンガリー及び中国の国際会議において発表したが、この2つの国際会議で御一緒した橋本康教授（愛媛大学）には御助言を賜ると同時に、大変、御

世話になった。深く感謝する次第である。

本論文の取りまとめにあたっては、農業生物学の立場から石井龍一教授（東京大学）、農業工学の立場から瀬尾康久教授（東京大学）、大下誠一助教授（東京大学）に、多くの御助言・御指導を賜り深く感謝する。

最後に、農業研究センター機械作業部の皆様には様々な面で御世話になった。心から御礼を述べたい。

付録

A MacNet のソースコード

ANSI-C で書かれた MacNet のソースコードを以下に示す。MacNet のすべての機能は、関数 `macro_neuro()` の引数 `read_file` に以下に示す値をセットして、`macro_neuro()` を実行することで利用できる。

ニューラルネット・モジュールは、別に用意したニューラルネット・シミュレータを用いてあらかじめファイルとして用意しておく必要がある。そのため、本コードはニューラルネット・シミュレータに依存している。以下に示すコードはニューラルネットシミュレータには RHINE Ver.3.1 (株)CRC 総合研究所) に対応している。`neuro.h` は、このシミュレータにおけるニューラルネットの構造体の定義を行うインクルードファイルである。

A-1 引数と戻り値

(1) マクロネットの計算結果の画面への出力制御

`output_on=0` 画面表示なし

`output_on=1` 画面表示あり

(2) 関数機能の制御

`macro_neuro()` の機能は、`read_file` の内容によって異なる。

0 実行

1 初期化 (実行しない)

ニューラルネット・モジュール（ネットワークデータ）の読み込み

- 2 初期化（実行しない）
ニューラルネット・モジュールの入出力ユニットのタグネーム（文字列データ）の読み込み
- 3 初期化（実行しない）
ニューラルネット・モジュール用メモリ領域確保&ニューラルネット・モジュール、
ニューラルネット・モジュールのタグネームの読み込み
- 4 初期化（実行しない）
ニューラルネット・モジュール用メモリ領域確保&ニューラルネット・モジュール
のタグネームの読み込み
- 1 ニューラルネット・モジュール用メモリ領域の解放
- 2 タグネーム記録用メモリ領域の解放
- 3 ニューラルネット・モジュール&タグネーム記録用メモリ領域の解放

(3) 関数の戻り値

- 1 システム定義ファイルがない
- 2 システム定義ファイルのバージョンが違って利用できない
- 3 ネットワーク定義ファイルがない
- 4 ネットワーク定義ファイルのバージョンが違って利用できない
- 5 メモリが確保できない
- 1 正常終了

(4) MacNet 定義ファイル (*.mnn) のコマンドとフォーマット

MacNet は、model{ }, input{ }, output{ } 文からなるテキストファイルを読み込み、これらの内容に従って自動的に必要なファイルを探し出してメモリに読み込み、実行する。

《model{ } 文》

model{ } で使用するニューラルネット・モジュール名（ファイル名と同一）を宣言する。

記法 model{action, safety} {第1サブネット名, 第2サブネット名 …}

《input{ }, output{ }文》

input{ }, output{ }文で、マクロネットの入出力ユニットのタグネームを定義する。

記法 input{ 入力名1, 入力名2, 入力名3 }

 output{ 出力名1, 出力名2, 出力名3 }

A-2 ソースコード

```

#include <stdio.h>
#include <dos.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <malloc.h>
#include <neuro.h>

#define MAXBUFF 1024

macro_neuro(fname,net_input,net_output,read_file,macro,output_on)
char *fname;
float *net_input,*net_output;
int read_file,output_on;
MACRO_NEURO *macro;

{
int cond,ans,i,j,k,L,m,count;
FILE *fopen(),*fpp;

if(read_file>0){
read_macro_file(fname,read_file,macro,output_on);
return 1;
}

if(read_file==0){
/* 初期設定 */
for(i=0;i<macro->n_subnet;i++){
for(j=0;j<macro->model[i].ncell[0];j++){
*(macro->in+i)*j)-0.0;
*(macro->in_set+i)*j)-0;
}
}
for(i=0;i<macro->n_subnet;i++){
macro->calculated[i]=0;
macro->set_value[i]=0;
}
/* in に質問項目の情報を入力する */
/* マクロ入力層と結合しているサブネットのサーチ */
for(j=0;j<macro->n_connect;j++){
if(macro->connect[j].model_out==0){
k=macro->connect[j].unit_out;
/* l=0 はマクロ入力層 */
l=macro->connect[j].model_in-1;
m=macro->connect[j].unit_in;
*(macro->in+l)*m)=net_input[k];
*(macro->in_set+l)*m)=1;
}
}
/* 全入力層データが入力されたサブネットの判定 */
i=search_set_model(macro,output_on);

```

```

/* サブネットの認識を実行 */
for(count=0;count<macro->n_subnet;count++){
    /* 入力層にデータがセットされたサブネットを探す */
    i=0;
    while(1){
        if(macro->set_value[i]==1
            && macro->calculated[i]==0)break;
        if(i==macro->n_subnet){
            if(output_on>0){
                printf("n/n/a 入力されたサブネットワークがありません。 \n");
            }
            exit(0);
        }
        i++;
    }
    /* 入力層にデータがセットされたサブネットを実行 */
    if(output_on>0){
        printf("nSUBNET [%d] is ACTIVE.\n\n",
            *(macro->name+i));
    }
    neuro(fname,*(macro->in+i),*(macro->out+i),
        0,&macro->model[i]);
    if(output_on>0){
        write_result(*(macro->in+i),
            *(macro->out+i),&macro->model[i]);
    }
    macro->calculated[i]=1;
    /* 推論 */
    for(j=0;j<macro->n_connect;j++){
        if(macro->connect[j].model_out==i+1){
            k=macro->connect[j].unit_out;
            /* 【注意】 */
            /* connect[j].model_* の 0 はマクロ入力層 */
            /* macro->in_set の 0 は最初のネットワーク */
            l=macro->connect[j].model_in-1;
            m=macro->connect[j].unit_in;
            *(*(macro->in+l)+m)=*(*(macro->out+i)+k);
            *(*(macro->in_set+l)+m)=1;
        }
    }
    /* 全入力層データが入力されたサブネットの判定 */
    search_set_model(macro,output_on);
}
/* 結果の処理 */
if(output_on>0){
    printf("n 【結果】 \n");
}
/* マクロ出力層と結合しているサブネットのサーチ */
for(j=0;j<macro->n_connect;j++){
    if(macro->connect[j].model_in==0){
        k=macro->connect[j].unit_in;
        l=macro->connect[j].model_out;
        m=macro->connect[j].unit_out;
        net_output[k]=*(*(macro->out+l)+m);
        if(output_on>0){

```



```

        printf("%6s : ",macro->mes_out[k]);
        printf("%6f\n",*(macro->out+l+m));
    }
}

return 1;
}

if(read_file<0){
    /* メモリ領域を解放 */
    for(i=0;i<macro->n_subnet;++){
        neuro(fname,macro->in_macro->out,
            read_file,macro->model+i);
    }
    if(read_file<2)free_macro(macro);
}

int search_set_model(macro,output_on)
int    output_on;
MACRO_NEURO    *macro;
{
    int    i,j,k,l;
    i=0;
    /* 全入力層データが入力されたサブネットの判定 */
    for(j=0;j<macro->n_subnet;j++){
        if(macro->calculated[j]==0){
            l=1;
            for(k=0;k<macro->model[j].ncell[0];k++){
                if(*(macro->in_set+j+k)-1)=0;
            }
            if(l==1){
                macro->set_value[j]=1;
                i++;
            }
        }
    }
    if(output_on>0){
        printf("n/n?????になった????ニューラルネットワーク数 : %d\n",i);
    }
    return    i;
}

write_result(in,out,model)
float    *in,*out;
NEURO    *model;
{
    int    i;
    printf("《%6s》 \n",model->system_name);
    printf("【入力】 \n");
    for(i=0;i<model->ncell[0];i++){
        printf("t%6s : ",model->input_mes[i]);
        printf("%6g\n",in[i]);
    }
}

```

```

    }
    printf("【出力】\n");
    for(i=0;i<model->nccell[model->layer-1];i++){
        printf("t%s: ",model->output_mes[i]);
        printf("%g\n",out[i]);
    }
}

no_file(fname,ext)
char *fname,*ext;
{
    printf("\n\n%s.%s\n",fname,ext);
    printf("\n\nファイルがありませんので終了します。 \n");
    exit(0);
}

free_macro(macro)
MACRO_NEURO *macro;
{
    int i;
    /* マクロネットの入力数 */
    free(macro->mes_in);
    free(macro->mes_out);
    free(macro->set_value);
    free(macro->calculated);
    /* マクロネット内サブネット結合データ保存領域の解放 */
    free(macro->connect);
    for(i=0;i<macro->n_subnet;i++){
        free(*(macro->in+i));
        free(*(macro->out+i));
        free(*(macro->in_set+i));
    }
    /* 各サブネットの入力層データ設定済み判別フラグ用領域の解放 */
    free(macro->in_set);
    /* 各サブネットの入出力用 float 変数領域の解放 */
    free(macro->in);
    free(macro->out);
    /* マクロネットを構成するサブネットの定義用 NEURO 構造体領域の解放 */
    free(macro->model);
}

get_command(fpp,buff1,buff2)
char *buff1,*buff2;
FILE *fpp;
{
    char c;
    int i;
    i=0;
    /* コマンド string の抽出 */
    /* 文字列"{"まで読む */
    if(read_string_to_char(fpp,buff1,('{')<0)return -1;

```

```

/* 文字列"j"まで読む */
if(read_string_to_char(fpp,buff2,'j')<0)return -1;
return 1;
}

read_string_to_char(fpp,buff,ch)
FILE *fpp;
char *buff,ch;
{
int i;
char c,d;
i=0;
retry:
while(1){
/* *, tab, CRLF の除去 */
do{
c=fgetc(fpp);
if(c==EOF)return -1;
/* コメントの読み飛ばし */
if(c=='/'){
d=fgetc(fpp);
if(d=='*'){
/* コメント終わりまで読む */
do{
d=c;
c=fgetc(fpp);
if(c==EOF){
printf("a\n/nノット文にエラーがあります。 \n/n");
return -1;
}
}while(!(d=='*' && c=='/'));
c=fgetc(fpp);
}
else{
buff[i]=c;
i++;
c=d;
}
}
}while(c==' ' || c=='\t' || c=='\n');
if(c==ch)break;
buff[i]=c;
i++;
if(i>=MAXBUFF)return -1;
}
buff[i]=0;
}

token_string(buff1,buff2,ch,n)
/* buff1 内文字列の n 番目トークン ch までの文字列を buff2 にコピー */
char *buff1,*buff2,*ch;
int n;
{

```



```

int      i,j,k,l;
k=strlen(buff1);
l=strlen(ch);
i=j=0;
/* トークンを n 回読み飛ばす */
while(i<n){
    /* 読み終わりの判定 */
    if(j+1+l>k){
        buff2[0]=NULL;
        return;
    }
    /* トークンのサーチ */
    if(strncmp(buff1+j,ch,l)==0){
        i++;
        j+=l-1;
    }
    j++;
}
/* 次のトークンまたは NULL までコピー */
i=0;
while(1){
    /* トークンのサーチ */
    if(j+1>k)break;
    if(strncmp(buff1+j,ch,l)==0)break;
    buff2[i]=buff1[j];
    i++;
    j++;
}
buff2[i]=NULL;
}

str_number(buff)
/* buff="###(X)" の x の値を返す */
char      *buff;
{
char      n[10];
int       i,j,k;
i=0;
k=strlen(buff);
do{
    if(buff[i]!='(')break;
    i++;
}while(i<k);
if(i==k)return -1;
j=0;
i++;
do{
    if(buff[i]==')')break;
    n[j]=buff[i];
    i++;
    j++;
}while(i<k);
if(i==k)return -1;
i++;

```

```

n[i]=NULL;
return  atoi(n)-1;
}

search_model_no(macro, buff)
/* buff で指定された名称のモデルの番号のサーチ */
/* No.1 からスタートする. */
MACRO_NEURO  *macro;
char          *buff;
{
    int      i,k;
    if(strcmp("MODEL",buff,strlen("MODEL"))==0)return 0;
    k=strlen(buff);
    for(i=0;i<strlen(buff);i++){
        if(buff[i]!='\')k=i;
    }

    i=0;
    do{
        if(strcmp(*(macro->name+i),buff,k)==0)return i+1;
        i++;
    }while(i<macro->n_subnet);
    return -1;
}

read_subnet(macro, buff2, buff1, read_file, output_on)
/* buff1 : {}内文字列
   buff2 : コマンド */
MACRO_NEURO  *macro;
char          *buff1,*buff2;
int          read_file,output_on;
{
    int      i,j;
    /* 利用するサブネットの個数のカウント ( buff1 内の ':' の数を数える) */
    macro->n_subnet=count_char(buff1,':')+1;
    /* マクロネットを構成するサブネットの定義用 NEURO 構造体領域の確保 */
    macro->model=(NEURO *)malloc(
        macro->n_subnet*sizeof(NEURO));

    /* 各サブネットの入出力用 float 変数領域の確保 */
    macro->in =(float **)malloc(macro->n_subnet*sizeof(float *));
    macro->out=(float **)malloc(macro->n_subnet*sizeof(float *));
    macro->model=(NEURO *)malloc(macro->n_subnet*sizeof(NEURO));

    /* サブネットの計算終了判定用ワークエリア */
    macro->set_value=(char *)malloc(macro->n_subnet*sizeof(char));
    macro->calculated=(char *)malloc(macro->n_subnet*sizeof(char));

    /* 各サブネットの入出力用 float 変数領域の確保 */
    macro->in =(float **)malloc(macro->n_subnet*sizeof(float *));
    macro->out=(float **)malloc(macro->n_subnet*sizeof(float *));

    /* 各サブネットの入力層データ設定済み判別フラグ用領域の確保 */

```

```

macro->in_set=(char **)malloc(macro->n_subnet*sizeof(char *));

/* サブネットの名称文字列領域の確保 */
macro->name=(char **)malloc(macro->n_subnet*sizeof(char *));

/* サブネットを読み込ませる */
for(i=0;i<macro->n_subnet;i++){
    /* トークン ;までの文字列を buff2 にコピー */
    token_string(buff1,buff2,";",i);
    if(output_on>0)printf("読み込み中 [%s]n",buff2);
    j=neuro(buff2,macro->in,macro->out,read_file,
             macro->model+i);
    if(j<0)return j;
    *(macro->name +i)=(char *)malloc((strlen(buff2)+1)*sizeof(char));
    strcpy(*(macro->name +i),buff2);
    *(macro->in +i)=(float *)malloc((macro->model[i].ncell[0]+1)
                                   *sizeof(float));
    *(macro->out+i)=(float *)malloc(
        (macro->model[i].ncell[macro->model[i].layer-1]+1)
        *sizeof(float));
    *(macro->in_set+i)=(char *)malloc((macro->model[i].ncell[0]+1)*
                                       sizeof(char));
}
return 1;
}

count_char(buff,c)
char      *buff,c;
{
    int      i,j;
    j=0;
    for(i=0;i<strlen(buff);i++){
        if(buff[i]==c)j++;
    }
    return j;
}

read_macro_file(fname,read_file,macro,output_on)
char      *fname;
MACRO_NEURO *macro;
int      output_on;
{
    char      *buff1,*buff2;
    int      i,j,k,l,connect_cn;
    FILE      *fopen(),*fpp;

    buff1=(char *)malloc(MAXBUFF);
    if(buff1==NULL){
        printf("a'n'n メモリが不足しています。 \n'n");
        return -1;
    }
    buff2=(char *)malloc(MAXBUFF);
    if(buff2==NULL){

```



```

    free(buff1);
    printf("a\n/n メモリが不足しています. \n/n");
    return -1;
}
strcpy(buff1,fname);
strcpy(buff1,"mnn");
if((fpp=fopen(buff1,"r"))==0){
    no_file(buff1,"");
}

/* ファイル中にある connect の文字列の数を数える */
i=j=k=l=connect_cn=0;
while(get_command(fpp,buff2,buff1)>0){
    if(strcmp(buff2,"connect")==0)i++;
    if(strcmp(buff2,"input")==0){
        j=read_inout_mes(macro,buff2,buff1);
        strcpy(buff2,"");
    }
    if(strcmp(buff2,"output")==0){
        k=read_inout_mes(macro,buff2,buff1);
        strcpy(buff2,"");
    }
    if(strcmp(buff2,"model")==0){
        l=read_subnet(macro,buff2,buff1,read_file,output_on);
    }
}
if(j!=1 || k!=1 || l!=1){
    printf("a\n/n?ネットワーク定義ファイルの内容に誤りがあります. \n/n");
    if(j==0)printf("input が定義されていません. \n/n");
    if(k==0)printf("output が定義されていません. \n/n");
    if(l==0)printf("model が定義されていません. \n/n");
    free(buff1);
    free(buff2);
    return -1;
}
macro->n_connect=i;

/* マクロネット内サブネット結合データ保存領域の確保 */
macro->connect=(CONNECT *)malloc(macro->n_connect*sizeof(CONNECT));
fclose(fpp);

strcpy(buff1,fname);
strcpy(buff1,"mnn");
fpp=fopen(buff1,"r");

/* 文字列"j"まで読む (buff1 に {} 内文字列 buff2 にコマンドが入る) */
while(get_command(fpp,buff2,buff1)>0){
    if(strcmp(buff2,"connect")==0){
        /* connect コマンド サブネット間の結合
           connect(action(4)->safety(3))
           : [??' ネット番号][出力エント番号]
           : - [??' ネット番号][入力エント番号]
           :
           buff2="connect" buff1="action(4)->safety(3)"
        */

```

```

token_string(buff1,buff2,"->",0);
k=search_model_no(macro,buff2)-1;
if(k<-1){
    printf("a\n?ネットワーク出力結合定義が不正です。 \n\n");
    return -1;
}
l=str_number(buff2);
if(k>=0 && (l<0 || l>macro->model[k]
            ncell[macro->model[k].layer-1])){
    printf("a\n?ネットワーク出力結合定義が不正です。 \n\n");
    return -1;
}
macro->connect[connect_cn].model_out=k+1;
macro->connect[connect_cn].unit_out=l;
token_string(buff1,buff2,"->",1);
k=search_model_no(macro,buff2)-1;
if(k<-1){
    printf("a\n?ネットワーク出力結合定義が不正です。 \n\n");
    return -1;
}
l=str_number(buff2);
if(k>=0 && (l<0 || l>macro->model[k].ncell[0])){
    printf("a\n?ネットワーク入力結合定義が不正です。 \n\n");
    return -1;
}
macro->connect[connect_cn].model_in=k+1;
macro->connect[connect_cn].unit_in=l;
connect_cn++;
}
}
fclose(fpp);
free(buff1);
free(buff2);
}

```

```

read_inout_mes(macro,buff1,buff2)
MACRO_NEURO *macro;
char *buff1, /* コマンド */
      *buff2; /* {内文字列} */
{
int ij;

/* マクロネットの入力数 ( buff1 内の ; の数を数える ) */
j=count_char(buff2,';')+1;

/* input コマンド */
if(strcmp(buff1,"input")==0){
macro->n_in=j;
macro->mes_in=(char **)malloc(macro->n_in*sizeof(char *));
/* トークン ; までの文字列を buff1 にコピー */
for(i=0;i<j;i++){
token_string(buff2,buff1,";",i);
*(macro->mes_in+i)=(char *)malloc((strlen(buff1)+1)
                                *sizeof(char));

```

```
        if(*(macro->mes_in+i)==NULL){
            printf("a\nメモリが不足しています。 \n\n");
            return -1;
        }
        strcpy(*(macro->mes_in+i),buff1);
    }
    return 1;
}

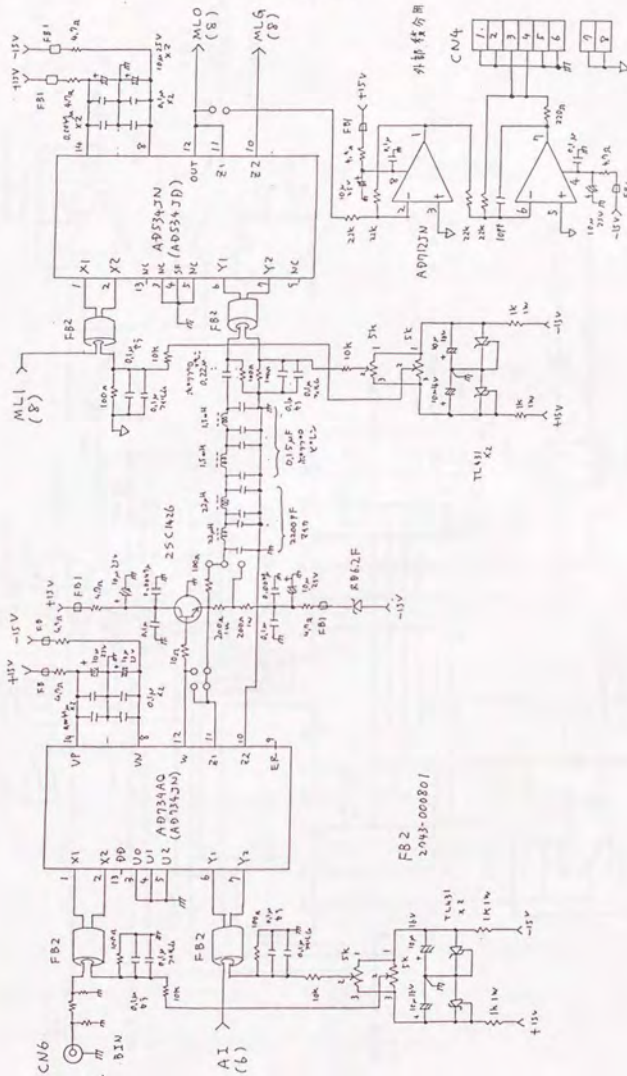
if(strcmp(buff1,"output")==0){
    macro->n_out=j;
    macro->mes_out=(char **)malloc(macro->n_out*sizeof(char *));
    /* トークン:までの文字列を buff1 にコピー */
    for(i=0;i<j;i++){
        token_string(buff2,buff1,",",i);
        *(macro->mes_out+i)=(char *)malloc((strlen(buff1)+1)
            *sizeof(char));
        if(*(macro->mes_out+i)==NULL){
            printf("a\nメモリが不足しています。 \n\n");
            return -1;
        }
        strcpy(*(macro->mes_out+i),buff1);
    }
    return 1;
}
}
```


B ハードウェア LV モデルの回路図

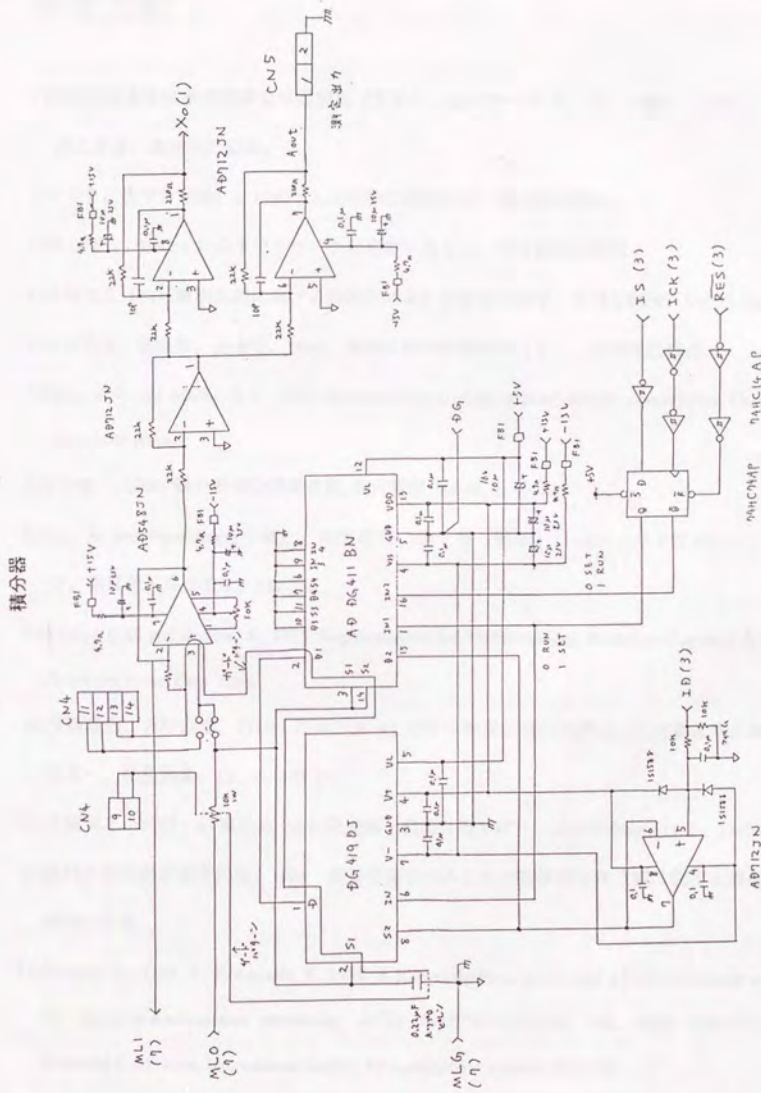
乗算器 2

LPF (ローパスフィルタ)

乗算器 1



図A-1 演算処理部 (図5-2の乗算器1, LPF, 乗算器2まで)



図A-2 積分器周辺主要部 (図5-2の積分器と初期状態設定用アナログスイッチ)

参考文献

- 1) 農林水産省農林水産技術会議事務局（監修），旭リサーチセンター（編），1986：人工知能と農業，農林統計協会。
- 2) システム農学会（編），1990：人工知能と農業情報，農林統計協会。
- 3) 田上隆一，1992：村のネットワークが農業を変える，日本経済新聞社。
- 4) 平藤雅之，1984：植物生産における知能化技術と複雑性の科学，計測と制御，Vol.33, No.10。
- 5) 町田武美，塩光輝，山中守，1994：地域農業の情報戦略（I），農林統計協会。
- 6) Pakes, A.G. and Maller, R.A., 1990: *Mathematical ecology of plant species competition*, Cambridge University press.
- 7) 甘利俊一，1990：脳の曖昧な情報表現，現代思想，vol.18, 7, 40-49。
- 8) Barr, A. and Feigenbaum（編），田中幸吉，淵一博（監訳），1983：人工知能ハンドブック，第II巻，共立出版，234-300。
- 9) Bobrow, D.G. and Collins, A., 1975: Representation and Understanding; *Studies in Cognitive Science*, Academic Press, New York.
- 10) 平藤雅之，古川嗣彦，1986：アメリカ AI 紀行－農業研究における人工知能応用に関する調査－，農業気象，42, 2, 171-176。
- 11) 平藤雅之，1987：AI 事始め Q&A 展望編－農業分野では－，*AI Spectrum*, 11, 2, 25-26。
- 12) 農林水産技術会議事務局，1988：農業生産管理のための新情報処理技術に関する開発調査研究成果集。
- 13) Hirafuji, M., Ono, Y., Kobayashi, K., 1988: A Knowledgebase system and a Neuro-computer system for agricultural information processing, *ACTA HORTICULTURAE*, 230, High Technology in Protected Cultivation, International Society for horticultural Science, 253-259.
- 14) システム農学会（編），1989：人工知能と新しい農業技術，農林統計協会。

- 15) 平藤雅之, 吉田智一, 小野良孝, 1987: 複数のエキスパートシェルによる各種エキスパートシステムの試作と評価, 農作業研究, 第22巻別号1, 41-42.
- 16) 平藤雅之, 1985: 自然言語処理による大規模知識の知識ベース化, 情報処理学会第31回全国大会講演論文集II.
- 17) 平藤雅之, 1987: パソコン用エキスパートシェル/AI 言語 (Prolog) の現状と利用方法, OHM 10月別冊, 「パソコンエキスパートシステム—低価格ツールによるエキスパートシステムの構築手順—」, オーム社, 4-15.
- 18) Whitehead, S.D., 1994: Auto-FAQ; An experiment in cyberspace leveraging, *Mosaic and Web Advance Proceedings*, Vol. 1, WWW Conference '94, 25-38.
- 19) Gallant, S.I., 1988: Connectionist Expert System, *Communications of the ACM*, Vol.31, No.2, 152-169.
- 20) 平藤雅之, 小野良孝, 1989: 作業エネルギーからみた耕草地生産システムの最適化, 農林水産省グリーンエネルギー計画成果シリーズ (V-2系), 農林水産省農林水産技術会議事務局.
- 21) Nicolis, G. and Pligogine, I, 1977: *Self-Organization in Nonequilibrium Systems*, Jhon Wiley & Sons, New York.
- 22) Haken, H., 1978: *Synergetics*, Springer-Verlag, Berlin Heidelberg. (H.ハーゲン著, 小森尚志, 牧島邦夫共訳, 1980: 協同現象の数理, 東海大学出版会).
- 23) 金子邦彦, 1991: 複雑系の理解に向けて, 数理科学, No.336, 5-8.
- 24) 平藤雅之, 1990: ニューロコンピューティング, バイオエキスパートシステムズ (星岳彦, 平藤雅之, 本條毅編著), コロナ社, 105-144.
- 25) Holland, J.H., 1945: *Adaptation in Natural and Artificial Systems*, Ann Arbor, The Univ. of Michigan Press.
- 26) Goldberg, D.E., 1989: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison

- Wesley Publishing Company.
- 27)入江文平, 川人光男, 1990: 多層パーセプトロンによる内部表現の獲得, 信学論, Vol. J 73-D-II, 8, 1173-1178.
- 28)Irie, B. and S. Miyake, S., 1988: Capabilities of Three-layered Perceptrons, *Transaction of ICNN'88*.
- 29)平藤雅之, 1990: 生物システムの最適化アルゴリズム, オペレーションズ・リサーチ, 日本オペレーションズ・リサーチ学会, Vol.36, No.10, 485-489.
- 30)Wang, Q., 1987: Optimization by Simulated Molecular Evolution, *Biological Cybernetics*, 57,95-101.
- 31)King, R.M. and Banerjee, P., 1986: ESP:Placement by Simulated Evolution, *IEEE Trans. CAD-8-3*, 245-256.
- 32)Harp, S. A., Samad, T. and Guha, A., 1990: Designing Application-Specific Neural Networks Using the Genetic Algorithm, *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann Publishers, 447-454.
- 33)Rogers, D., 1990: Predicting Weather Using a Genetic Memory: a Combination of Kanerva's Sparse Distribution Memory with Holland's Genetic Algorithms, *Advances in Neural Information Processing Systems 2*, Morgan Kaufmann Publishers, 455-464.
- 34)Keesing, R., Stork, D.G., 1991: Evolution and learning in neural networks: The number and distribution of learning trials affect the rate of evolution, *Advances in Neural Information Processing Systems 3*, Morgan Kaufmann Publishers, 804-810.
- 35)瀧保夫, 1978: 情報論 I -情報伝送の理論-, 岩波書店, 29-37.
- 36) Langton, C.G. (Ed.), 1987: *ARTIFICIAL LIFE*, The proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems, held september in Los Alamos, New Mexico, Addison-Wesley Publishing Company, Inc.
- 37)高橋磐郎, 早迫亮一, 1990: 自己増殖オートマトンの設計と試作-生命の数学モデル-, 情報処理学会論文誌, Vol.31, No.2, 238-248.

- 38) 郡司幸夫, 長島知正, 1990: 生物的自律性とその形式化, 計測と制御, Vol.29, No.10, 893-898.
- 39) 郡司幸夫, 今野紀雄, 1991: 自己言及と自律的境界, 数理科学, No.336, JUNE, 54-58.
- 40) 稲吉宏明, 1993: 人工生命への招待, 情報処理, Vol.34, No.7, 884-891.
- 41) Thro, E, 1993.: *Artificial Life Explorer's Kit*, SAMS Publishing.
- 42) 平藤雅之, 1991: ファジィニューラルネットによる植物生長のモデル, 情報処理学会研究報告, **32**, 5.
- 43) Lindenmayer, A., 1968: Mathematical Models for Cellular Interactions in Development, *J. Theoret. Biol.*, **18**, 280-299.
- 44) Prusinkiewicz, P, Lindenmayer, A., 1990: *The Algorithmic Beauty of Plants*, Springer-Verlag New York Inc., 1-3.
- 45) 土居洋文, 1968: 生物のかたちづくり, サイエンス社, 55-84.
- 46) 渋沢栄, 藤浦建史, 竹山光一, 岩尾俊男, 1990: トウモロコシ根系分布のべき法則とLシステムによるシミュレーション, 農機北海道支部報, **31**, 27-31.
- 47) 渋沢栄, 藤浦建史, 岩尾俊男, 竹山光一, 1993: トウモロコシ根系パターン形成の階層的モデリング, 農業機械学会誌, **55**, 2, 101-108.
- 48) Sakai, S., 1991: A Model Analysis for the Adaptive Architecture of Herbaceous Plants, *J. theor. Biol.*, **148**, 535-544.
- 49) Hirafuji, M. 1990: A Plant Growth Model by Neural Networks and L-systems, *Proc. IFAC 9th Symposium on Identification and System Parameter Estimation.*, Pergamon Press, Oxford, 605-609.
- 50) Yokomori, Y., 1980: Stochastic characterization of EOL languages, *Information and Control*, **45**, 26-33.
- 51) Eichhorst, P and Savitch, W.J., 1980: Growth functions of stochastic Lindenmayer systems, *Information and Control*, **45**, 217-228.
- 52) Herman, G. T. and Rozenberg, G., 1975: *Developmental systems and languages*, North-Holland,

- Amsterdam.
- 53) Lindenmayer, A. and Rosenberg, G. (Ed.), 1976: *Automata, Languages, development*, North-Holland, Amsterdam.
- 54) Zadeh, I.A., 1968: Probability Measure of Fuzzy Events, *J. of Math. Anal. and Appl.*, **22**, 421-427.
- 55) 加藤一郎: 大豆の生育, 作物大系第4編豆類, 養賢堂, 13-37.
- 56) 御小柴公人(編著), 1975: *ダイズの作り方*, 農山漁村文化協会, 38-73.
- 57) Dawkis, R., 1988: The Evolution of Evolvability, in *ARTIFICIAL LIFE*, The proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems, held september in Los Alamos, New Mexico, Addison-Wesley Publishing Company, Inc., 201-220.
- 58) Dawkis, R., 1986: *The Blind Watchmaker*, Harlow, Longman. (リチャード・ドーキンス著, 中嶋康裕, 遠藤彰, 遠藤知二, 足田努訳, 日高敏隆監修: ブラインド・ウォッチメイカー—自然淘汰は偶然か?—, 早川書房)
- 59) 池田清彦, 1991: 構造主義科学論からみた進化論史, 講座進化①—進化論とは—, 東京大学出版会, 79-120.
- 60) Hosmer, D.W. and Lemeshow, S., 1989: *Applied Logistic Regression*. A WILEY-INTERSCIENCE PUBLICATION, a division of JHON WILEY & SONS, Inc., New York.
- 61) Hsu, F.H., Nelson, C.J. and Chow, W.S., 1984: A Mathematical Model to Utilize the Logistic Function in Germination and Seedling Growth, *Jornal of Experimental Botany*, Vol.35, No.160, 1629-1640.
- 62) 易佑林, 橋本康, 1988: 光強度とCO₂ガス濃度に応答するサラダナ光合成速度の非線形システム同定, *生物環境調節*, **26**, 4, 171-174.
- 63) 平藤雅之, 小野良孝, 小林恭, 1989: ニューラルネットによる多変量解析とエキスパートシステム構築方法, *日本ソフトウェア科学会第5回大会論文集*, 113-116.
- 64) Bourland, Y and Kamp, Y., 1988: Auto-Association by Multilayer Perceptrons and Singular Value

- Decomposition, *Biological Cybernetics*, 59, 291-294.
- 65) Asoh, H. and Otsu, N., 1988: Nonlinear Data Analysis and Multilayer Perceptrons, *Proc. of IJCNN 1989 in Washington D.C.*, II, 411-415.
- 66) 片山泰男, 1988: ニューラルネットによる画像符号化, *PCSJ* 88, 3-5, 37-38.
- 67) 森本哲夫, 竹内利信, 橋本康, 1993: 遺伝アルゴリズムおよび人工ニューラルネットワークを利用した水耕の最適制御法, *生物環境調節*, 31, 1, 21-27.
- 68) Murase, H. and Honami, N., 1991: Kalman Filter Neuro-Computing for Biological System Models Using Neural Networks, *Proc. IFAC/ISHS 1st Workshop on Mathematical and Control Applications in Agriculture and Horticulture*, IFAC Workshop series, Vol. I, Pergamon Press, Oxford, 343-348.
- 69) 栗田多喜夫, 1989: ニューラルネットにおけるモデル選択の試み, *電子通信情報学会技術研究報告*, PRU89-16, 17-22.
- 70) 小長谷明彦, 1993: GA の遺伝子情報処理への応用, 計測と制御, Vol.32, No.1, 76-81.
- 71) 平藤雅之, 1992: 器官間干渉を基礎とした植物生長モデルとその同定法, *農業気象*, 48, 3, 285-293.
- 72) Hirafuji, M., 1991: A Time-continuous Quantitative Plant Growth Model based on Neural Networks and Lotka-Volterra Equations, *Proc. IFAC/ISHS 1st Workshop on Mathematical and Control Applications in Agriculture and Horticulture*, IFAC Workshop series, Vol. I, Pergamon Press, Oxford, 289-294.
- 73) Hofbauer, H., Sigmund, K., 1984: *Evolutions Theorie und Dynamische Systeme*, Berlin und Hamburg, (ホッフバウアー, シグムンド著, 竹内康博訳, 生物の進化と微分方程式, 現代数学社, 68-82.)
- 74) Shigesada, N., Kawasaki, K. and Teramoto, E., 1984: The effects of interference competition on stability, structure and invasion of a multispecies system, *Journal of Mathematical Biology*, 21, 97-113.

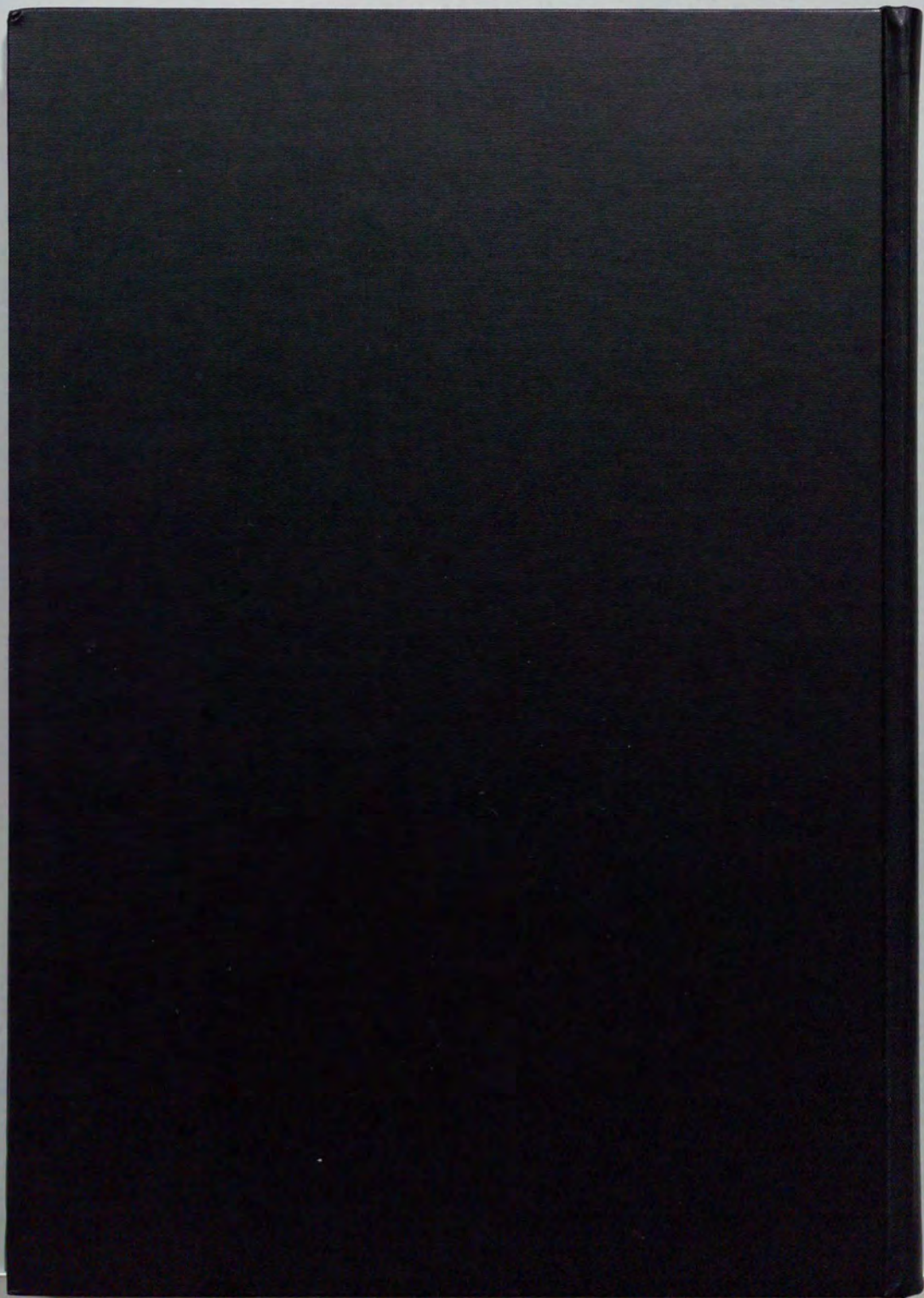
- 75) Hopfield, J.J., Tank, D.W., 1986: Computing with Neural Circuits: A Model, *Science*, Vol.233, 625-633.
- 76)Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986: Learning Internal Representations by Error Propagation, in *Parallel Distributed Processing* (Eds. by McClelland, J.L., Rumelhart, D.E. and the PDP Research Group), Vol.1, 354-361, MIT Press.
- 77)Pineda, F.J., 1987: Generalization of Back-Propagation to Recurrent Neural Networks. *Physical Review Letters*, 59, 2229-2232.
- 78)Pearlmutter, B.A., 1989: Learning State Space Trajectories in Recurrent Neural Networks, *Proceedings of IJCNN 1989* in Washington D.C., II, 365-372.
- 79)Amari, S., 1967: A theory of adaptive pattern classifiers, *IEEE Trans. EC-16*, 279-307.
- 80)Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986: Learning representations by back-propagating errors, *Nature*, Vol.323, 533-536.
- 81)農林水産技術会議事務局, 1987: C3, C4, CAM作物個体群における太陽エネルギー利用(資料集), グリーンエナジー計画成果シリーズ, II系(物質固定), No.13.
- 82)Irie, B. and Miyake, S., 1988: Capabilities of Three-layered Perceptrons, *Transaction of ICNN'88*.
- 83)棟方研, 1980: 葉緑体, 理工学社.
- 84)和田野晃, 1993: 生体内代謝経路のコンピュータシミュレーション, *生物物理*, Vol.33, No.2, 39-44.
- 85)Takakura, T., Kozai, T., Tachibana, K. and Jordan, K.A., 1974: Direct digital control of plant growth. I. Design and operation of the system. *Trans. ASAE*, 17, 1150-1154.
- 86)May, R.M., 1976: Simple mathematical models with very complicated dynamics. *Nature*, 261, 459-467.
- 87)Hirafuji, M., 1992: Neural shell for biological complex systems, *Proc. of IFAC workshop on expert systems in agriculture*, International Academic Publishers, 75-82.
- 88)平藤雅之, 窪田哲夫, 1994: 変動環境下における植物生長のカオス性, *生物環境調節*, 32, 1,

- 31-39.
- 89)Lorenz, E.N., 1983: Deterministic nonperiodic flow, *J. Atmos. Sci.*, **20**, 130-141.
- 90)May, R.M., 1974: Biological populations with non overlapping generation: stable points, stable cycles and chaos, *Science*, **186**, 645-647.
- 91)品川嘉也, 瀬野裕美, 1991: 脳波とフラクタル, 生物物理, Vol.31, No.6, 38-42.
- 92)長島知正, 永井喜則他, 1990: 時系列データ解析とカオス, 計測と制御, Vol.29, No.9, 53-60.
- 93)Goodwin, R.M., 1990: *Chaotic Economic Dynamics*, Oxford University Press.
- 94)下條隆嗣, 1992: カオス力学入門, 近代科学社.
- 95)Takens, S., 1981: Detecting Strange Attractors in Turbulence, *Lecture Notes in Mathematics*, **898**, 366-381.
- 96)Grassberger, P. and Procaccia, I., 1983: Estimation of Kolmogorov Entropy from Chaotic Signal, *Physical Review Letters*, **50**, 346-349.
- 97)日本農業気象学会, 1977: 農業気象ハンドブック 285-311, 養賢堂.
- 98)今井勝, 1981: 個体, 群落の物質生産の測定と解析, 「光合成研究法」(加藤栄, 宮地重遠, 村田吉男編集), 59-73, 共立出版.
- 99)Ruelle, R., 1990: Deterministic chaos: the science and fiction, *Proc. of R. Soc., London, A*, **427**, 244-248.
- 100)Jordan, M.I., 1986: Attractor dynamics and parallelism in a connectionist sequential machine. *Proc. of the Eighth Annual Meeting of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum.
- 101)Garfinkel, A., Spano, M. L., Ditto, W. L. and Weiss, J. N., 1992: Controlling Cardiac Chaos, *Science*, Vol. 257, 1230-1235.
- 102)Shinbrot, T., Grebogi, C., Ott, E. and Yorke, J.A., 1993: Using small perturbation to control chaos, *Nature*, Vol. 363, 411-417.
- 103)Weaver, J.C. and Astumian, R.D., 1990: The Response of Living Cells to Very Weak Electric Fields: The Thermal Noise Limit. *Science*, Vol. 247, 459-462.

- 104)河村広巳, 重光司, 菅沼浩敏, 遠藤織太郎, 1981: 電界の生物学的影響に関する文献調査
-植物に対する影響-。電力中央研究所報告, 調査報告:480020.
- 105)横井博一, 斉藤正男, 1986: 新しい学習素子-Foulethret -, 電子通信学会論文誌,
Vol.J69-A, No. 6, 1173-1175.
- 106)高山一, 1991: スピングラス, 丸善, 東京.
- 107)Buttiker, M., Imry, Y. and Landauer, R., 1983: Josephson behavior in small normal one-dimensional
rings, *Physics letters*, Vol.96A, No.7, 365-367.
- 108)Ambegaokar, V. and Eckern U., 1990: Coherence and Persistent Currents in Mesoscopic Rings,
Physical Review Letters, Vol.65, No.3, 381-384.
- 109)Suenobu, K., Tani, S. and Imamura, A., 1993: An *ab initio* Molecular Orbital Study on Effect of
Electric Field Strength on Ionic Selectivity of the Sodium Channel, *J. theor. Biol.*, **161**, 395-403.
- 110)Rice, A., 1992: New Ideas for Guiding the Evolution of a Quantum System, *Science*, Vol.258, 412-
413.
- 111)Paul, B. and Shapiro, M., 1989: Coherence Chemistry: Controlling Chemical Reactions with Lasers,
Account of chemical research, Vol.22, No.12, 407-413.
- 112)Judson, R.S. and Rabitz, H., 1992: Teaching Lasers to Control Molecules, *Physical Review Letters*,
Vol.68, No.10, 1500-1503.
- 113)永田親義, 1986: 量子生物学入門, 学会出版センター, 東京.
- 114)Frohlich, H., 1970: Long Range Coherence and Action of Enzymes, *Nature*, **228**, 1093.
- 115)Wu, T.M. and Austin, S., 1978: Bose-Einstein Condensation in Biological Systems, *J. theor. Biol.*,
70, 209-214.
- 116)Lauck, L., Vasconcellos, A.R. and Luzzi, R., 1992: On Frohlich Coherent Effects in Biological
Systems: Influence of Carriers and High Order Dissipative Effect, *J. theor. Biol.*, **158**, 1-13.
- 117)Paul, R., Chatterjee, R., Tsuzynski, J.A. and Fritz, O.G., 1983: Theory of Long-range Coherence in
Biological Systems I. The anomalous Behaviour of Human Erythrocytes, *J. theor. Biol.*, **104**, 169-

- 185.
- 118) Pokorny, J., 1983: Fröhlich Electromagnetic Radiation from Human Leukocytes: Implication for Leukocyte Adherence Inhibition Test, *J. theor. Biol.*, **102**, 295-305.
- 119) Popp, F.A. and Nagl, W., 1983: A physical (electromagnetic) model of differentiation: 2 Applications and examples, *Cytobios*, **37**, 71-83.
- 120) Chwirot, W.B., 1986: New Indication of Possible Role of DNA in Ultraweak Photon Emission from Biological Systems, *J. Plant Physiol.*, Vol.122, 81-86.
- 121) Wolf, F.A., 1989: On the Quantum Physical Theory of Subjective Antedating, *J. theor. Biol.*, **136**, 13-19.
- 122) Stuart, C.I.J.M., Takahashi, Y. and Umezawa, H., 1978: On the Stability and Non-local Properties of Memory, *J. theor. Biol.*, **71**, 605-618.
- 123) Sivakami, S. and Srinivasan, V., 1983: A Model for Memory, *J. theor. Biol.*, **102**, 287-294.
- 124) Lockwood, M., 1989: *Mind, Brain and the Quantum*, The Compound 'I', Basil Blackwell Limited.
(マイケル・ロックウッド著, 奥田栄訳, 1992: 心身問題と量子力学, 産業図書, 東京)
- 125) 松野孝一郎, 1991: プロトバイオロジー, 東京図書, 東京.
- 126) 大木幸介, 1984: こころの量子論, 日経サイエンス社, 東京.
- 127) 町田茂, 1994: 量子力学の反乱—自然は実在するの—, 学習研究社.
- 128) 並木美喜雄, 1985: マクロ系の量子力学と観測問題, 物理学最前線, **10**, 共立出版, 140-216.
- 129) Herbert, N., 1985: *Quantum Reality*, John Brockman Associates Inc., New York. (ニック・ハーバート著, はやし・はじめ訳, 1990: 量子と実在, 白揚社)
- 130) 江沢洋, 1987: 量子論の発展とパラドクス, 日本物理学会編「量子力学と新技術」, 培風館, 204-242.
- 131) Tryba, V. and Gosser, K., 1991: A Modified Algorithm for Self-Organizing Maps Based on The Schr

- ödinger Equation, *Artificial Neural Networks*, Lecture Note in Computer Science, **540**, Springer-Verlag, 33-45.
- 132) Deutsch, D. and Jazsa, R., 1982: Rapid solution of problems by quantum computation, *Proc. R. Soc. Lond.*, A **439**, 553-558.
- 133) Deutsch, D., 1985: Quantum theory, the Church-Turing principle and universal quantum computer, *Proc. R. Soc. Lond.*, A **400**, 97-117.
- 134) Lloyd, S., 1993: A Potentially Realizable Quantum Computer, *Science*, Vol. 261, 1569-1571.
- 135) 福田礼次郎, 1990: マクロ系の量子力学, 丸善, 9-40.
- 136) Cairns, J., Overbaugh J. and Miller, S., 1988: The origin of mutants, *Nature*, Vol. 335, 142-145.
- 137) Cairns, J., 1990: Causes of mutation and Mu excision, *Nature*, Vol. 345, 213.
- 138) 柴谷篤弘, 1991: 現代進化生物学の波乱, 講座進化④ 進化論とは, 東京大学出版会, 207-243.
- 139) Culotta, E., 1994: A boost for "Adaptive Mutation", *Science*, Vol. 265, 318-319.
- 140) Rosenberg, S. M., Longrich, S., Gee, P. and Harris, R. S., 1994: Adaptive Mutation by Deletion in Small Mononucleotide Repeats, *Science*, Vol. 265, 405-407.
- 141) Lenski, R. E. and Mittler, J. E., 1993: The Directed Mutation Controversy and Neo-Darwinism, *Science*, Vol. 259, 188-194.
- 142) Neumann, J.V., 1932: *Die Mathematische Grundlagen der Quantenmechanik*. (井上健, 広重徹, 恒藤敏彦訳, J.V.ノイマン著, 1957, 量子力学の数学的基礎, みすず書房, 332-352)
- 143) Jammer, M., 1974: *The Philosophy of Quantum Mechanics*, John Wiley & Sons, Inc. (マックス・ヤンマー著, 井上健訳, 1983: 量子力学の哲学, 紀伊国屋書店)
- 144) Bastin, T. (Ed.), 1971: *Quantum Theory and Beyond*, Cambridge University Press, London. (柳瀬睦夫, 村上陽一郎, 黒崎宏, 丹治信春訳, 1973: 量子力学は越えられるか, 東京図書)



inches
cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Kodak Color Control Patches

© Kodak, 2007 TM: Kodak



Kodak Gray Scale



© Kodak, 2007 TM: Kodak

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

