

Title	神経回路網ダイナミクスを用いた適応的情報処理に関する研究
Author(s)	太田, 智
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/729
Rights	
Description	Supervisor: 櫻井 彰人, 知識科学研究科, 修士

修 士 論 文

指導教官 櫻井 彰人 教授

北陸先端科学技術大学院大学
知識科学研究科 知識システム基礎学専攻

950017 太田 智

審査委員： 櫻井 彰人 教授（主査）
林 幸雄 助教授
橋本 敬 助教授

2001 年 2 月

目次

1.	1
1.1 はじめに	1
1.2 背景と指針	1
1.3 目的	3
1.4 手順	3
2.	4
2.1 モデルの選定と設計	4
2.1.1 スパイキングニューロンモデル	4
2.1.1.1 作成したモデルと実際のニューロン	5
2.1.1.2 モデルの数理的表現	5
2.1.2 シナプス可塑性のモデルと更新規則	8
2.1.3 ネットワークモデルについて	9
2.2 ネットワーク構造の設定	9
2.3 入力の設定	9
2.4 プログラムの実装方法	10
3.	12
3.1 実験1 スパイキングニューロンの挙動	12
3.1.1 ネットワーク構造	12
3.1.2 結果とモデルの性質	12
3.2 実験2 単純なネットワーク	13
3.2.1 ネットワーク構造	13
3.2.2 結果とモデルの性質	13

3.3 実験3 周期性をもったダイナミクス	14
3.3.1 ネットワーク構造と入力パターン	14
3.3.2 結果	15
3.3.3 考察	17
3.4 実験4 非周期性をもったダイナミクス	19
3.4.1 ネットワーク構造と入力パターン	19
3.4.2 結果	19
3.4.3 考察	22
3.5 実験5 周期性をもつ原因	24
3.5.1 ネットワーク構造と入力パターン	24
3.5.2 結果	24
3.5.3 考察	25
3.6 実験6 複雑な入力と周期性をもったダイナミクス	26
3.6.1 ネットワーク構造と入力パターン	26
3.6.2 結果	29
4.	30
4.1 実験結果からの考察	30
4.1.1 スパイキングニューラルネットダイナミクスとパラメータ	30
4.1.2 スパイキングニューラルネットダイナミクスと情報処理	30
4.1.2.1 周期性	30
4.1.2.2 周期性と情報処理性	31
4.1.2.3 パラメータと情報処理性	31
4.1.3 現象の数学的解釈	32
4.1.4 モデルのウェイト更新規則と実際の細胞のシナプス可塑性	32
4.1.4.1 モデルのウェイト更新規則	32
4.1.4.2 ウェイト更新可能範囲	32
4.1.4.3 ウェイト更新程度	33
4.2 Vaadiaらの実験と本研究の接点	33

4.2.1	Vaadiaらの実験内容	34
4.2.2	Vaadiaらの実験と本研究のモデル	35
5.		38
6.		39
		40
		42

1. 研究背景と目的

1.1 はじめに

本研究では、ニューロンモデルを利用して研究を行うが、それを研究していく先に何を期待しているのかをここで明確にしておく。

脳で起こっている現象は様々な計器を利用して事細かに調べられてきている。また、コンピュータの処理能力が上がり、細胞周りの分子レベルシミュレーションすら可能となってきた。しかし、そうした観測技術の進歩があり、微細なシミュレーションが可能となった今でも、脳の働きは暗かに閉ざされたままで、情報処理がどのようにおこなわれているか、全くと言って良いほどわかっていない。

一方、脳の情報処理方法はコンピュータの情報処理方法と随分異なっていて、コンピュータが苦手とする作業を簡単にやってのけている場面が多々あったりする。そのため、脳の情報処理機構の解明が、コンピュータの苦手とする作業に新たな処理アルゴリズムを与えてくれるという期待が持たれ続けている。ニューラルネットワークを研究する目的の多くは、この脳独特の情報処理性を明確にし、その結果をコンピュータにフィードバックさせることと考えている。

なお、この研究を通じて期待するものは、コンピュータへのフィードバックを兼ね、純粋に脳の働きを理解したいためである。

特に脳の働きの中で不思議なのは、神経細胞が遺伝子で記述された構造を使い、単に新陳代謝を行っているだけにもかかわらず、その影響が情報処理となっている事実である。つまり、はじめから情報処理を行う目的で神経回路が発現しているわけではないのに、どうして情報処理が可能になってしまうか？それが核心的謎である。恐らく細胞自体は大域的にどういった影響を及ぼしているかなど知る由もない。単に自らの状態を維持すべく、細胞個々は細胞なりの活動を永遠続けているだけである。それにもかかわらず、細胞の集団はコンピュータのような情報処理性を有してしまう。これが謎でなくて何が謎だろうか。本研究の大目的は、こうした情報処理性が細胞のどんな活動によって実現されているかを探ることである。そして、この研究を通じ、そうした活動の「何か」でも探り当てればと考えておこなった。ところで、本文中で「情報処理性」という言葉を使用する。これは、そもそも情報処理とは別の現象や処理にみえるものが、情報処理をともなってしまう性質を指して「性」を付加している。神経細胞は情報処理をしているのかもしれない。しかし、細胞自身が情報処理を意識（そもそも細胞が意識をもつのかかわらないか）していないと認めるのなら、神経回路網が情報処理をしていると断言するよりは、情報処理性を有していると表現するほうが妥当と考えられる。

1.2 背景と指針

近年、スパイクニューラルネット(パルスニューラルネットとも言われる)に関心が集まっている。コンピュータの性能が飛躍的に上がり、シミュレーションしやすくなったのも要因のひとつであろうが、肝心なことは、高次機能を担う皮質において、スパイクタイミングの相対的時間差に依存した情報処理性が見出され[1]、シナプス可塑性[2][3]の学習則においても、シナプス前ニューロンとシナプス後ニューロンの発火タイミングに依存してシナプスウェイトがダイナミックに変化しているという知見が得られたことである[4][6][7]。

スパイクニューラルネットは人工的に作られたニューロンの一種である。ニューロンモデルには細

胞のどんな働きを重視するかによって多数のモデルが存在しているが、このスパイクニューロンモデルは、実際の細胞から観測されている発火現象とその細胞間で伝播しているスパイクに注目して作成されている。一方、従来の学習や記憶の研究に用いられてきたニューロンモデルは、スパイクの概念を省いて組み立てられてきたものが大半である。そのため、脳の大脳皮質における情報処理性がスパイクに依存的で、かつ、スパイクの時空間的構造が本質的情報をコードしている可能性を示唆した近年の研究結果にそぐわないモデルとみられる場合もある。そして、こうした従来のモデルを用いていると高次機能を説明できない可能性も考えられている。もちろん、スパイクタイミングのズレやスパイクタイミング依存シナプス可塑性が、本質的情報処理の副次的現象として観測されているだけかも知れないので、簡単には結論付けられない。しかし、スパイクレベルの話が本質的でないと即座に断言できないのなら、このスパイクによる情報処理性について検討するのは、本質的な情報処理を探る上でも重要な方向性と考えられる。

こうした方向性と、スパイクタイミングは大切な情報処理性を担っているのではないかと言う私自身の直感も手伝って、本研究はスパイクニューラルネットワークを計算機上に実装し、シミュレーション実験する方針をとった。

また、従来のニューラルネットワークで用いられている学習のあり方についても考えをめぐらし、新たな学習の可能性について検討する。本研究において導入した学習規則は、Markram[4]らが実際の脳にて発見した「スパイクタイミング依存シナプス可塑性 (spike-timing-dependent synaptic plasticity: STDPと略す)」に相当するものである。この可塑性を簡単に説明すると、発火に携わった入力を通過させたシナプスウェイトを上げ、発火以降の応答不可能な時間帯 (不応期と呼ぶ) に入力をもたらしたシナプスウェイトを下げるという規則 [5][6][7][26] である。

実験では上記のシナプスウェイト更新規則を半永久に適応し、周期的な時空間パターンを入力した場合で生じたダイナミクスを、諸所のパラメータを変えながら調べていく。なお、こうしたスパイクニューロンにウェイト更新規則を付加させてネットワークを作成すると、同じ初期ウェイト値の回路であっても、絶えずウェイトが変化する関係で、異なった入力スパイクパターンに対しては異なったネットワークダイナミクスが発生すると予想される。本研究ではそうした現象が、本当に生じるかどうかを調べてゆく。

また、そうしたダイナミクスが現れたのなら、このメカニズムを利用してVaadia[1]らがサル運動前野で発見した、ニューロン間の発火タイミングのズレが起こる原因を説明できないかを試みる。ちなみに、本研究の出発点は、Vaadiaらの発見した不可思議な2つのニューロン[1]にある。これらのニューロンはタスクのおおの動作に対して影響を与えたり受けたりせず、あたかもタスクの内容に依存して同期的な発火タイミングを変えているように見える細胞である。現在のところこのメカニズムがどのように実現されているのか明らかにされていない。そこで、実際に得られている知見からニューラルネットワークモデルを構築し、同じような挙動を示すニューロンモデルを組み立てることで、Vaadiaが発見した2つのニューロンの発火タイミング変更メカニズムを理解しようと試みたのである。

ところで、従来のニューラルネットの研究と比べて、本研究の趣は異なっている。それは、学習後や学習中の出力、シナプスウェイトを対象として回路性能を評価せず、すなわち、ネットワークの学習能力や記憶力を入出力の正確さや冗長性を調査して評価せず、ただただ、ウェイト変更規則を適用させながら出力時系列を観測し、その時系列から観測されるネットワークのダイナミクスを調べるのみだからである。こうしてダイナミクスを評価し、学習や記憶を考える方向性は、記憶や学習のメカニズムが回路網ダイナミクスと深い関係があるという仮説を根拠に据えている。そしてまた、ニューラルネットワークが有するダイナミクスを解明し、学習や記憶のメカニズムに迫ろうとする方向性もある。

Aertsen[8]らは、2つのニューロンの同期的発火タイミングを変えるような現象を説明するモデルとしてパルスニューロンネットワークモデルを提案し、考察しているが(たとえば似たようなものに[26]も)、本研究で取り扱うような入力パターンに対する網羅的なネットワークダイナミクスには言及していない。また、本研究と似たような入力を想定してダイナミクスに言及しているパルスニューロンモデルの研究もたくさん

あるが、1つのニューロンの性質について研究している場合がほとんどである（たとえば[9]のような）。

1.3 目的

研究の目的はスパイクニューロンのネットワークを作成し、そこにSTDPを実装させた場合、どのようなダイナミクスが生じるか、それを極単純な時空間パターンを入力としてネットワークに与えることにより調べる[31]。また、修士論文の企画書で述べたVaadiaらの実験結果に対する解釈も、上記のモデルとシミュレーションの結果を踏まえて行う。

以上が本研究の目的である。

1.4 手順

(1) まずはニューロンモデルを選定する

後の「2.1モデルの選定と設計」で選定を行う理由を述べるが、要は従来一般的に使われてきたニューロンモデルではスパイクのシミュレーションができないため、ニューロンモデルの選定が不可欠である。モデルの構造とその数理的表現に関しては2章以降で述べる。

(2) 選定を終えたら、それを計算機に実装する

実装する際はMicrosoft Visual C++を使用した。計算機シミュレーションなので、速度面を考慮してC系のプログラム言語を、そして開発しやすさという面を考慮してMicrosoftの開発環境Visual Studioを選択した。尚、実装したプログラムに関しては、付録Aに綴った。

(3) 単純なネットワークを作成してネットワーク全体の挙動を観測してみる（3章）

モデルが構築したとおりの挙動を示すかを調べるフェーズである。「3. 実験」の3.1節と3.2節の手続きは、この方針を具体化したもの。

(4) STDPによるシナプス変化を追加し、実験する（3章）

はじめはウエイトを変化させない場合の基本的なニューロンモデルとネットワークを完成させる。そのときにバースト(発火しっぱなし)したり、無発火状態に陥らない範囲のパラメータ設定や入力規模をあらかじめ把握しておき、そうした上でウエイト変更規則の実装を試みる。パラメータ設定の際、神経生理学的知見として得られているものに関しては極力それに近くなる形でパラメータを設定した。知見のないものや、分からぬものに関しては、自分でいろいろ変更させてみた。

(5) パラメータを変えながらシミュレーションし、結果を収集する（3章）

(6) 結果を考察する（4章）

(7) 結果を踏まえてVaadiaらの検出したデータのモデルを考える

Vaadiaらの実験については「4.2 Vaadiaらの実験と本研究との接点」の部分でまとめてある。

2. スパイキングニューラルネットワーク

2.1 モデルの選定と設計

スパイクのシミュレーションとなると、軸索上のパルス位置、樹上突起上の信号の位置、ひいては信号の厳密な生起タイミングや細胞の内部状態など、諸所の変数を厳密な時間を踏まえてシミュレートする必要がある。だが、従来のニューラルネットモデルの多くは、そもそもそうした状況が無視しても構わないとの理由により、スパイクや細かい時間の概念が組み込まれていない場合が多かった。

今回の研究では、スパイクの位置やパルスの生起タイミングなどをシミュレーションできるスパイキングニューロンをベースとしたネットワークモデルを対象に据える。つまり、これを実践するのにふさわしいモデルを作成する。

しかしながら、諸所の神経生理学の文献を調査し、全くの無から組み立てるのは困難である。そのため、既存のモデルを流用する方針を採った。今回、そのために注目したモデルはGerstner[10][11]が作成したintegrate-and fire neuron modelである。彼の提案したモデルは、スパイクが細胞に入力されてから軸索へ発火シグナルが伝達するまでの一連のスパイク位置が時間を追って厳密に記述されているので、パルスの位置、発生タイミングを記述する上で有効なモデルである。また、このモデルの原点はHodgkin-Huxley方程式[12]に置かれているため、ある程度内部電位の厳密性も考慮されている。こうした点から、本研究ではGerstnerのモデルをベースに実験用のモデルを作成することにした。ちなみに、そうして組み立てたモデルは以下のような性質を持つ。

- ・ 内部電位に対してははっきりとした発火条件が備わっている（閾値の概念）
- ・ 入力される信号の大きさは、接続されているものによって程度が違う（ウエイトの概念）
- ・ ウエイトはSTDPに従って更新される（ウエイト変更の概念）
- ・ 複数の入力によってもたらされる影響は、線形和をとる（加算部分の概念）
- ・ シナプス後から、加算部分の距離（遅延の概念1）
- ・ 加算部から軸索を通過してシナプス前に達するまでの距離（遅延の概念2）
- ・ 伝播するスパイクはデルタ関数であらわされる（スパイクの概念）

細かい話になるが、今回は大規模なニューラルネットワークを想定していないため、ネットワークに独特な層構造を持たせてシミュレートさせなかった。つまり、脳の局所的な接続状況に見られるような相互結合性を再現するという意味で、ニューロンモデルが準相互結合を成すように設定した。ここで「準相互結合」という表現をしたのは、完全に相互結合しているわけではないという意味で「準」とした。なお、完全結合にしなかった理由は特にない。

2.1.1 スパイキングニューロンモデル

ここからはニューロンモデルについて説明する。説明の順序としては、まず、モデルを組み立てる際に根拠とした事柄を述べる。そしてそうした根拠で作ったニューロンモデルひとつと、それに対するのひとつの入力に関する内部変化の様子を数理的に説明し、ついで複数の入力になされた場合の内部変化について数理的に述べる。

2.1.1.1 作成したモデルと実際のニューロン

神経細胞がスパイクのような鋭い波形を出力するのは、細胞の内部電圧が急速に変化するからである。そして、そうした鋭い変化は、細胞のイオンチャンネルが一斉に、かつ一瞬だけ開閉するために生じている。しかしながら、本研究で想定しているスパイクングニューロンモデルは、イオンチャンネルのレベルまで具体化しないまま、イオンチャンネルの働きによって生じるスパイクの存在はモデルに取り入れる。つまり、あるニューロンがスパイクを受け取ると、伝達遅延を伴った後に内部にスパイクの影響がじわりと広がり、そしてその影響は入力の数に応じて増大し、いずれその影響が限界に達すると自らもスパイクを発生させる。こうした現象は真似るようにする。

これらの、スパイク、遅延、内部への影響、入力の数に応じて増大する影響は、実際のニューロンの表面的性質である。ただ、さらに細部の、たとえばレセプターやチャンネルや分子1つ1つの位置関係を考慮しないのは、細胞ネットワークの情報処理性を考えるときには必要ないと考え、切り捨てたからである。イオン交換などは、細胞の代謝にとって必要不可欠な機能かもしれないが、細胞が情報処理のためだと把握しているようには思えない。つまり代謝の二次的現象として情報処理が生じていると考えるのが自然だろうと直感した。もっとも、イオン交換を引き起こすことが細胞にとって具体的にどういった交換をもたらしているかは知らないが、細胞が情報処理をするためだけにイオン交換をしているとはやはり考えにくい。こうした理由から、細胞の新陳代謝や維持活動の一環とみなせるイオン云々は実装せず、イオンがチャンネルを通じてイオンが行き来した結果として生じる二次的現象を模倣することにした。

また、今回使用したモデルは、正確なスパイクの位置関係がシミュレーションできなければならない関係から、軸索のどの位置にスパイクがあるか、樹上突起のどの位置まで信号が伝播しているかもモデルに取り入れる必要がある。だから、軸索、樹上突起に伝播距離の概念を設け、スパイク位置を正確に表現できるようも設計した。ところで、この実装の裏返しは伝達遅延の存在であり、接続長が異なればユニークな遅延を持たせられる仕組みである。

以上の構築根拠から、ニューロンモデルを概念的に示すと、図2.1のように表現できる。

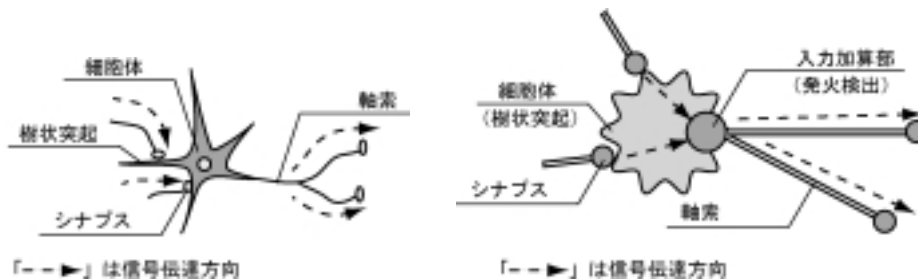


図2.1 実際の細胞の概略図（左）とモデルの構造（右）

モデルでも実際のニューロンでも、シナプス、樹上突起、軸索を伝播する信号には伝達遅延が伴う。シナプスは図上で小さい楕円もしくは円として描かれているが、実際には軸索の先と樹上突起の表面とその隙間を指してシナプスと呼んでいる。また、実際の細胞では、入力の影響が加算される仕組みが備わっていて、加算された電圧値がある値（閾値）を超えると発火する傾向を持っていることが知られている。

2.1.1.2 モデルの数理的表現

ニューロンの物理的構造を決定したので、これらを数理的に表現しなおすことにする。そのために、1

つのニューロンにおける1つのシナプスからの入力の影響を数理的に説明し、そののちに複数シナプスからの入力を考える。

シナプス i において、ある時刻 t_j にシナプス後へ到着した信号は、遅延時間 d の後、細胞に内部変化をもたらす。その内部電位の変化は、以下の式であらわされる。

$$V_{ij}(t) = w_{ij} \frac{t - t_j - d}{\tau} e^{-\frac{t - t_j - d}{\tau}} H(t - t_j - d) \quad (\text{式2.1})$$

t はある時刻を指し、 w_{ij} はシナプス i における時刻 t_j のウエイトで、内部電位の電圧方向の強弱を決定付けている。 τ は膜時定数で、これによって内部変化の時間軸方向の度合いを調整している。関数 $H(\cdot)$ はヘビサイドステップ関数である。実験では、 w_{ij} に初期値として2~4を与え、 τ は0.01に設定した。

ちなみに(式2.1)を「内部電位と時刻」の関係によりおおよその内部変化を示した形を示す(図2.2)。

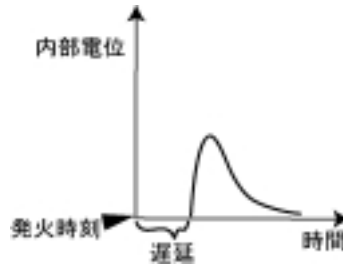


図2.2 入力後の内部電位の様子

シナプス後に達して直ぐ内部電位は変化せず、遅延の後、急速に上昇し、徐々に減退する。

以上が、ひとつの入力に関する内部変化であるが、本来の挙動として複数の入力を考慮する必要があるので、ここからは複数の入力の場合について内部電位を考える。今回の場合、複数の入力に対しては、Gerstner[5]らが行った処理と同様、入力個々に対する内部電位を単に線形的な和をとって総内部電位を記述する方針を採った。

つまり、シナプス i における入力の集合(パルスの時系列)を $\{t_{ij}; j=0 \dots n\}$ と定義し、それらの時刻におけるそれぞれのウエイトを w_{ij} であらわすと、ある細胞に接続されているニューロンの内部電位は、(式2.2)であらわすことができる。

$$V_i(t) = \sum_{j=0}^n w_{ij} \frac{t - t_{ij} - d}{\tau} e^{-\frac{t - t_{ij} - d}{\tau}} H(t - t_{ij} - d) \quad (\text{式2.2})$$

そして、シナプス i の入力集合を $\{j_i = 0 \dots n_i\}$ と定義しなおして、すべてのシナプス $\{i = 0 \dots m\}$ からの入力を考慮して細胞の内部電位を求めると、それぞれの入力からの影響を線形和で表せる(図2.3)ので、

$$V(t) = \sum_{i=0}^m \left\{ \sum_{j_i=0}^{n_i} w_{t_{ij}} \frac{t - t_{ij} - d}{\tau} e^{-\frac{t - t_{ij} - d}{\tau}} H(t - t_{ij} - d) \right\} \quad (\text{式2.3})$$

となる。

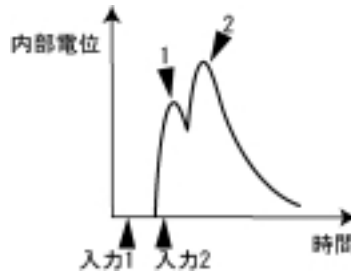


図2.3 2つの入力に対する内部電位の反応例

入力を検知しても内部電位が直ぐに変化しないのは、伝達遅延が考慮されているからである。入力が複数ある場合、それぞれの入力に対する内部電位時系列を計算し、それらの線形和をとることによってニューロンモデルの内部電位時系列としている。

ところで、シナプスによって遅延 d が異なっていたり、 d が以前の内部電位に依存した関数 $d(t)$ として表されている可能性も十分考えられる。そして、 $d(t)$ が入力に対して何等かの情報処理性を与えているかもしれない。だが、今回のモデルではその影響を考慮していない。

内部電位の定義を終えたので、次は発火するタイミングについて説明する。

発火は内部電位が閾値 θ を超えた時刻を算出すれば、それが発火した時刻になるようにした。つまり(式2.4)を描き、これを時刻 t について解けば、その時刻が発火時刻となる。

$$V(t) \geq \theta \quad (\text{式2.4})$$

実際のニューロンでは、スパイクと内部電位の区別がない。発火スパイクは内部電位のとりわけ高いピーク(活動電位)として表現されるだけである。しかし今回のモデルではこの両者を引き離し、その引き離しを担うものとして固定閾値を導入している。これは、より簡単な理論で発火現象を説明するために設けられた便宜上の概念で、昔から使われている。しかしながら、もし閾値が内部でダイナミックに動いているという事実と、それによって本質的情報処理がなされている証拠が得られたのであれば、固定型閾値を導入しているモデルは再考を求められる。ちなみに、Gerstnerはダイナミック閾値モデルという概念を導入して、ニューロン集団発火現象をいくつか説明している[11]。

本研究で使用したモデルも活動電位を表現するところを省き、固定型閾値を導入して発火するタイミングを算出した。つまりスパイクである活動電位を厳密にシミュレーションせず、スパイクをデルタ関数と見立てている。

それから、実際のニューロンは発火の後に不応期を持っている。不応期とは、入力にたいして内部電位を変えないような状態が数ミリ秒続くことである。今回のモデルでは、こうした不応期は実装した(図2.4)。

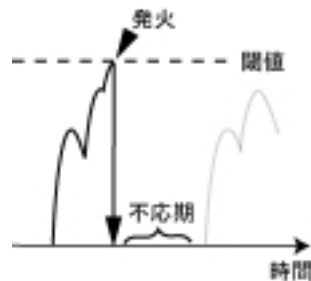


図2.4 発火と不应期

発火の直後、不应期と呼ばれている時間帯があり、この時間帯に入力されたスパイクに対して、内部電位は変化しない。

2.1.2 シナプス可塑性のモデルと更新規則

シナプスウェイト(以降は単にウェイトとあらわす)更新規則の概要を「はじめに」で示したが、具体的な規則を数式であらわすと(式2.5)ようになる。ちなみにこれは実際に得られた知見[4][6][7]を基に組み立てられている。

$$\Delta W = \begin{cases} Ae^{\frac{\Delta t}{\tau}} & \text{if } (\Delta t < 0) \\ -Ae^{-\frac{\Delta t}{\tau}} & \text{if } (\Delta t \geq 0) \end{cases} \quad (\text{式2.5})$$

Δt は { (発火した時刻) - (信号入力時刻) } で、 ΔW はウェイトを更新する程度。 τ はシナプスの強化や退化が起こるときのISI (interspike interval: 図2.5) の範囲を決定する定数で、 A は更新程度を決める定数である。実験では、 τ を0.01とした。

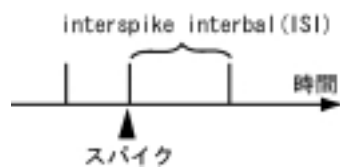


図2.5 inter-spike interval

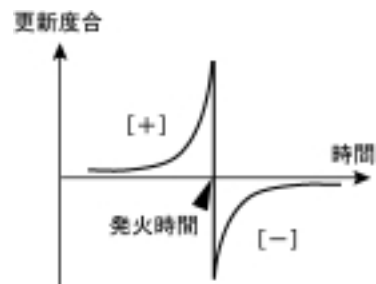


図2.6 ウェイト更新度と発火と入力の関係

図2.6は(式2.5)を概略図化したものである。発火に携わった入力に関してはシナプス可塑性による強化が起こり(ウェイトが増し)、逆に発火した後に入力されたものに関しては退化させられる(ウェイトが減る)。しかも、どちらの場合も発火した時間に近ければ近いほどその程度が大きいう具合になっている。ちなみに、こうした現象を説明するにも、代謝の話と絡めるとうまくできそうである。つまりHebb[13]

が示したように、発火に関与した場合の接続を強め、そうでないものを弱めるのは、細胞の代謝効率が係わっているように見えるのである。なぜなら、活動電位を発生させるような大きな電位変化が起きれば、たくさんのイオン分子が内外で交換されると考えられる。だからそうした機会が多く得られれば、細胞にとって内部に蓄積した無駄なものを交換する機会が増えて有益なのかもしれない。よってその機会を与えてくれる部位（シナプス）を敏感にし、利益をもたらさない部位は弱められるのかもしれない。

また、今回はウエイト更新が永遠に行われるものの、ウエイトは無限に上昇したり、下降したりしない仕組みを採用した。これはウエイトが永久に大きくなりつづけたり、小さくなりつづけたりするのは不自然と思われるので、こうした方針をとった（4.1.4節）。

ところで、最近の知見では、図2.6に示されるような変化のみならず、様々なバリエーションのウエイト更新関数が見つけられている[7]。

2.1.3 ネットワークモデルについて

ネットワークは2.2.2節で述べたニューロンモデルをくみあわせて作成する。今回の実験の場合、一様乱数を利用して接続パターンを決めた。また、それらの接続はニューロンの位置で距離が決まり、そしてその距離に応じて諸所のニューロン間の伝達時間が設定されるようになっている。よって、接続関係が同じでも細胞が配置された位置関係が異なっていれば、伝達距離も変わってくるので、同じ時空間パターンに対する反応も微妙ながらも変わってくる。

2.2 ネットワーク構造の設定

ネットワークの規模は、シミュレーション環境の制約によって決定付けられた。なぜなら、規模が大きければ大きくなるほど、シミュレーション時間は長くなり、それに乗じて採集しなければいけないデータも指数関数的に多くなって、解析しにくくなるからである。

こうした大量長期の厄介さを避けるため、ニューロンモデル数とそれらのコネクショ数を極力小さい状況を想定し、それでも尚、面白そうな結果が出るような規模を選ぶこととした。しかしながら具体的にどのくらいの規模だと面白い結果が出るようになるのか。それらは経験的に知ったという以外に説明できない。

そして、その経験則として得られた規模は以下の通りである。

- ・ ニューロン数 16から20くらい（うち、入力素子が半分弱くらい）
- ・ 興奮性ニューロン数 : 抑制性ニューロン数 4 : 1
- ・ 興奮性ニューロンの投射数 : 抑制性ニューロンの投射数 2 : 3

なお、ネットワークにおける興奮性ニューロンと抑制性ニューロンの存在割合や、投射の割合については解剖学的知見を参考に設定した[14][15]。

2.3 入力の設定

本研究では、皮質の局所的な神経回路網をモデルしている。

一方、大脳皮質で観測されるISIは、ランダム性の強いスパイクインターバルを持った過程とみなされている。これは観測されるスパイクインターバルの平均値と標準偏差の比が1に近い過程である[16]。また、一定刺激を与えた場合、Poisson分布に従うISIが観測されるという知見もある[17]。

こうした事実から、本研究ではニューラルネットワークへの入力として、Poisson過程（指数分布をとる過程）に準ずるスパイクインターバルを有した入力パターンを想定した。そのスパイクインターバルの生

成方法は、以下のような手続きを踏まえておこなう。

- (1) 一様乱数 $\{ x: 0 < x < 1 \}$ を発生させる。
- (2) x を以下の(式6)に代入しISIを導出する。

$$ISI(x) = -\lambda \ln(x) \quad (\text{式6})$$

は分散の平均値を変えるパラメータ。

- (3) もし、ISIが不応期より短いならば、(1)に戻ってやりなおす。

以上の操作を直感的に理解しやすいよう図示したのが、図2.7である。

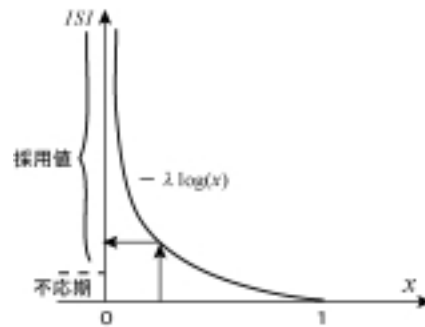


図2.7 ランダム数とISIの関係

2.4 プログラムの実装方法

プログラムは少しばかりオブジェクト指向で組み立てた。

ニューロンオブジェクトは、樹上突起棘オブジェクト、軸索オブジェクト、細胞体オブジェクトという具合の3つのオブジェクトからなる。また、これらのオブジェクトに関連をもたせ、パルスニューラルネットワークオブジェクトを構成するようになっている。これ以外にも様々なオブジェクトをもうけたが、ここではモデルと本質的つながりのあるオブジェクトのみに注目し、簡単にその役割を説明する。

・ ネットワークオブジェクト

このオブジェクトはスパイクニューラルネットワーク全体を管理している。ちなみにネットワークはニューロンオブジェクトの集合からなる(図2.7)。また、ネットワークに課せられるグローバルな時間や、どの軸策がどの樹上突起の突起棘に到達しているかを管理するのも、ニューロンの内部状態やウエイトパルスの時系列データを保持するのも、このオブジェクトである。

単位時間辺りの実行もこのオブジェクトのメンバ関数が行う。この関数は、このオブジェクトで管理される総てのオブジェクトにける単位時間シミュレーションを連続して行う。

- ・ ニューロンオブジェクト

ニューロンオブジェクトは主に3つのパートからなる。次に示す3つは、それらを簡単に説明したものである。

- ・ 樹上突起棘オブジェクト

ウエイトの値を保持し、ウエイトの上下限を管理する。パルス到達時刻も管理し、遅延時間と見比べて細胞の加算部にいつ信号が到達するかを計算する。

- ・ 細胞体オブジェクト

内部電位の総和を求め、閾値と比較して発火するか否かを定める。発火の履歴、入力履歴、ウエイトの履歴はこの部分で保持される。これらの履歴は、発火が起こればすれてクリアされる。不応期の計算もこのオブジェクトが行う。

- ・ 軸索オブジェクト

接続先のニューロンオブジェクトまでの物理的な距離を保持している。発火を確認して、相手の樹上突起棘にいつ到達するかを計算する。

- ・ パラメータオブジェクト

ニューラルネットワークで使用されているおおよそのパラメータは、このオブジェクトで管理されている。このオブジェクトのメンバ関数によって変更されたパラメータを即座にネットワークへと反映させられるメンバ関数を持つ。入力パターンなどもこのオブジェクトが管理する。

なお、これらのオブジェクトの構造を概念的に示すと、図2.8ようになる。

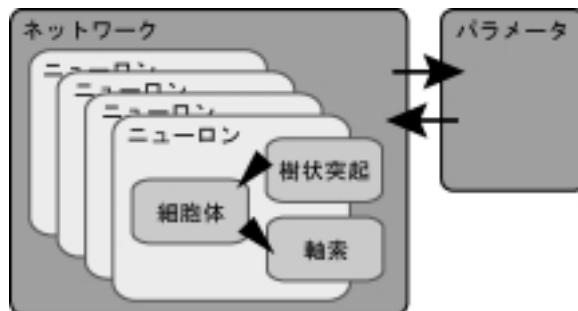


図2.8 各オブジェクトの関係図

3. 実験

作成したモデルの挙動を確かめる目的も兼ね、まずは簡単なネットワークの挙動を提示する。そののちに規模大きくして、ネットワークダイナミクスを観測する。ちなみに、実験ために変更したパラメータは、主に以下に示すものである。

- 入力生成用のシード
- ネットワーク生成用のシード
- STDPのシナプス更新程度およびシナプス更新可能範囲

3.1 実験1 スパイキングニューロンの挙動

3.1.1 ネットワーク構造

まずは回路の特性を示すため、もっとも単純な接続の場合を考えてシミュレーションする(表3.1)。用意したニューロンは2つで、一方が入力専用、もう一方がその入力専用細胞からのスパイクを受け取るニューロンとした(図3.1)。

	入力専用	興奮性	抑制性	合計
ニューロンの数	1	1	-	2
1ニューロンあたりの投射数	1	0	-	1

表3.1 実験1で用意したネットワーク構造

3.1.2 結果とモデルの性質

まずは結果を図示する(図3.2 - 図3.3)。



図3.1 使用した構造
ニューロン0からニューロン1への投射がある。1に隣接する小さい丸はシナプスを表現している。

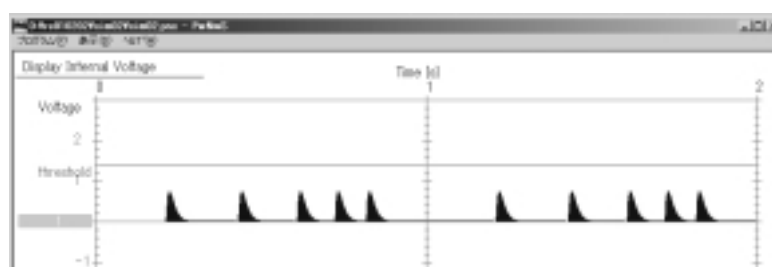


図3.2 ニューロン1から観測された内部電圧時系列
単一の入力のみでは、閾値を上回らないので発火しない。

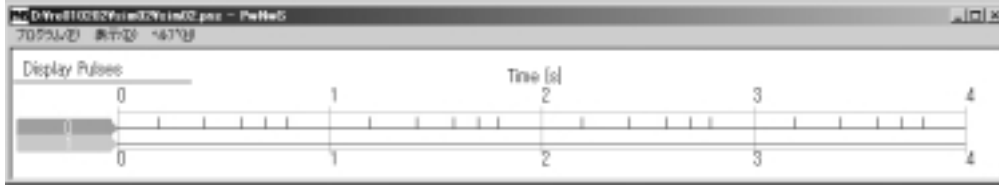


図3.3 観測されたパルスパターン

ニューロン0が発火しているにもかかわらず、投射を受けているニューロン1は発火しない。図3.2でも示したが、ニューロン1の内部電圧が変化してもその変化が閾値を上回っていないので発火しない。

入力した時空間パターンは、図3.3のニューロン0(ラベル0)の部分に記した時系列である。ニューロン0には、1秒ごとに同じパターンを入力しているため、1秒ごとに同じスパイクパターンが繰り返し観測される。一方、ニューロン1は投射を受けているにもかかわらず一切発火しない。これは、単独の入力によって発火せず、少なくとも2つ以上の入力がほぼ同時に入力されなければならない同時刻検出性がニューロンに付与されているためである。ニューロンはひとつの入力に対する内部電位の反応として、急速に上昇しその後ゆっくりと下降する性質を持つので、ひとつの入力にたいする影響よりも閾値を高くすれば、こうした同時刻検出性を実装できることになる。ただし、内部電圧変化を規定している時定数が大きく、立ちあがった電圧が時間をかけて下降するように変えると、同時刻検出性は失われる。

3.2 実験2 単純なネットワーク

3.2.1 ネットワーク構造

同時刻検出性を概観するため、もう1つ簡単な例を紹介する。今回用意した回路は表3.2と図3.4に示すような構造である。

	入力専用	興奮性	抑制性	合計
ニューロンの数	2	1	-	3
1ニューロンあたりの投射数	1	0	-	2

表3.2 実験2で用意したネットワーク構造

3.2.2 結果とモデルの性質

異なる2つのニューロンから投射を受けているニューロンが発火するには、それぞれの入力からほぼ同時にスパイクが到着しなければならない(図3.5、図3.6)。これが同時刻検出性である。今回の場合は、入力として設定した2つの細胞の発火パターンに相関性がさほど無いため、その入力を受けているニューロンはなかなか同時に入力を受け取れていない。

しかしながら、複雑なネットワーク構造をもつようになると1つのニューロンに投射される数が増すため、ほぼ同時に到着するパルス量は格段に上昇する。そのため、それなりに発火してくれるようになる。

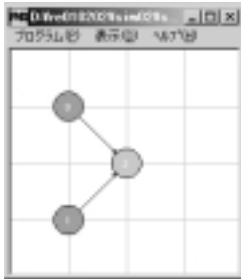


図3.4 使用した構造
ニューロン0とニューロン1から
ニューロン2への投射がある。

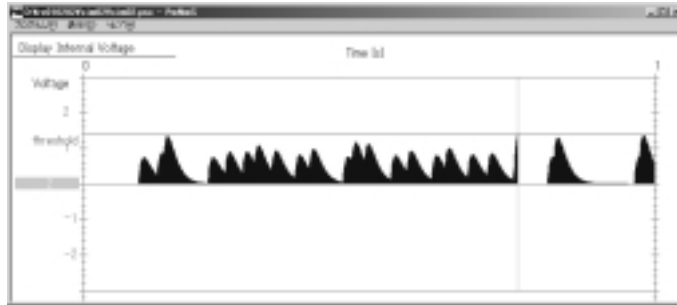


図3.5 観測された内部電圧時系列
入力されるタイミングが合わなければ、なかなか発火しない。
閾値を超えて発火すると、その時刻に縦線が描かれる。

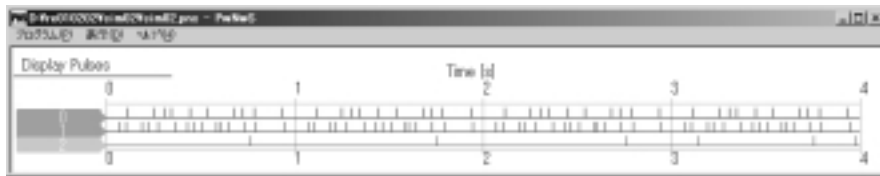


図3.6 観測されたパルスパターン
ニューロン0とニューロン1は盛んに発火しているが、投射を受けているニューロン2はなかなか発火しない。

3.3 実験3 周期性をもったダイナミクス

3.3.1 ネットワーク構造と入力パターン

今回は先ほどの実験1、実験2よりも大きな回路を用いた。ネットワークの内容を表3.3、図3.7に示す。ネットワークを作成する際、接続対象と、入力以外のニューロンの配置(2×4)はランダムに決定した。また、今回のネットワークに入力として与えた時空間パターンを図3.8に示す。図3.8からもわかるように同じ時空間構造をもったスパイク列を、1秒ごとに繰り返し入力した(2.4節参照)。そして、この時空間パターンを1000秒間入力しつづけ、最初の数秒間を除いて、ダイナミクスを調べた。

	入力専用	興奮性	抑制性	合計
ニューロンの数	8	6	2	16
1ニューロンあたりの投射数	3	3	6	54

表3.3 実験3で用意したネットワーク構造

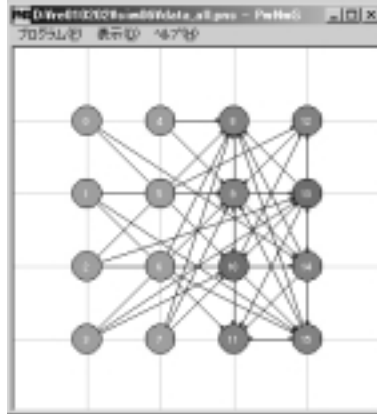


図3.7 実験3で用いたネットワーク

3.3.2 結果

まず、入力ニューロン以外のニューロンから観測されるスパイクを見ると、それは必ずしも1秒ごとに同じ時空間パターンが出力されているようでもなかった。しかしながら、どの1秒間と比較しても異なっているわけでもなかった(図3.8)。また、こうした現象は、2.1.2節で紹介したウエイト更新規則を付与しない場合では起こらなかった。つまり1秒ごとに同じスパイク時空間パターンを持つ入力と同じく、入力以外のニューロンからも1秒ごとに同じ時空間パターンが繰り返されていた。こうしたことから、各1秒ごとに異なった時空間パターンが入力専用ニューロン以外から観測されるようになった原因を、今回の場合、ウエイト更新規則が適応され続けられたためと考えられる。

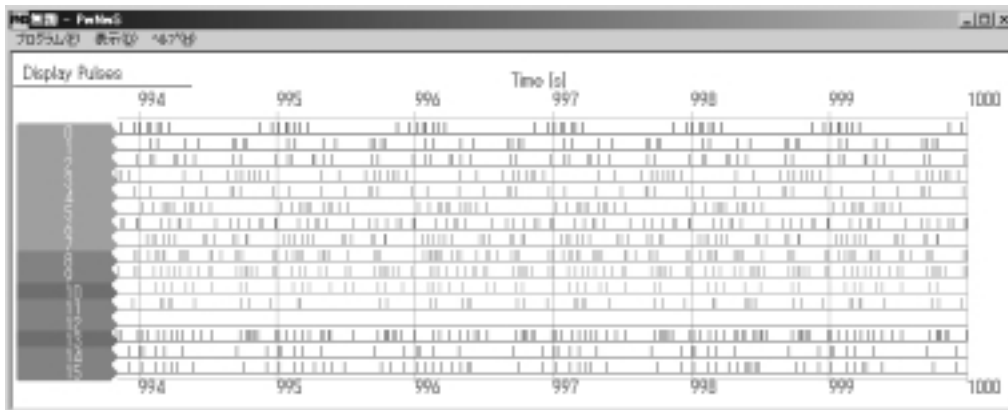


図3.8 入力と出力のスパイク時空間パターン(部分:994-1000秒)
 入力(ニューロン0~ニューロン7)の時空間パターンは1秒ごとに同じ物が繰り返されている。それにもかかわらず、入力以外(ニューロン8~ニューロン15)の時空間パターンは微妙に異なっている。ただ、ニューロン11を観ると、994-995秒の時空間パターンと、999-1000秒の時空間パターンは同じようである。

これらの現象をさらに詳しく調べるため、ウエイトのデータ(図3.9 図3.10)や、内部電圧のデータ(図3.11)の解析を行った。するとウエイト時系列が周期的に変遷していると示されていた。そしてその変遷が、5秒ごとに繰り返されていると分かった。また、内部電圧のリターンマップを作成したところ、それが周期5秒であることが示されていた(図3.12)。

今回の場合、1秒ごとに同じ入力繰り返し与えられていると分かっているので、500秒から1000秒までの500秒間のそれぞれ1秒間を(図3.13)に示す形でサンプリングし、そのうちのある値とその1つ前(1秒前)の値を2次元情報と見立てて、2次元平面にプロットしたものをリターンマップとして用いた。

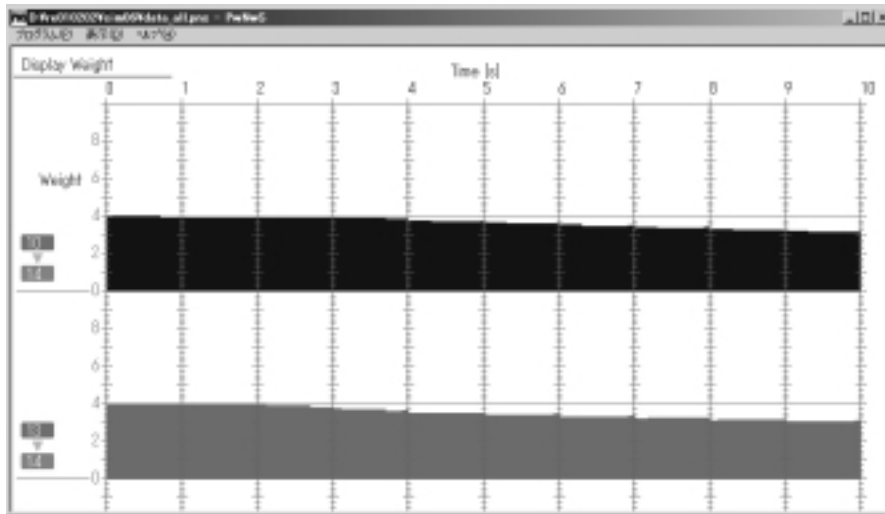


図3.9 ウエイトの時系列データ(部分:0-10秒)

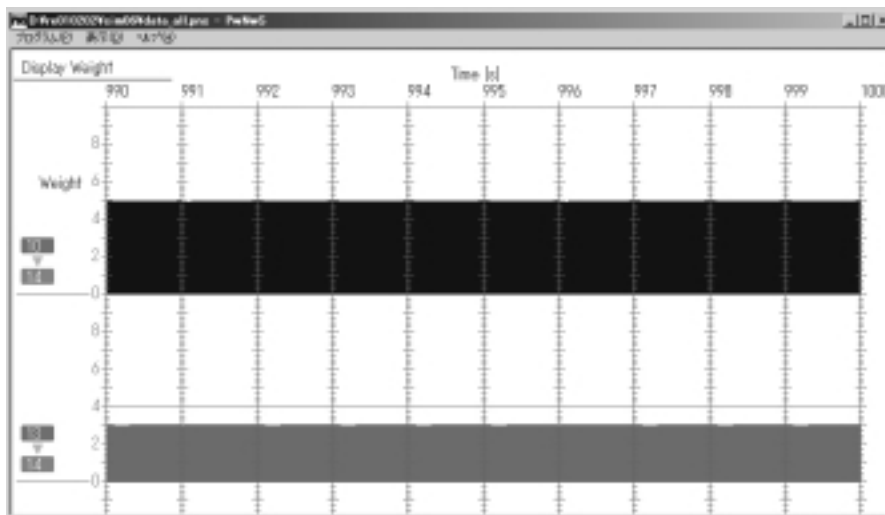


図3.10 ウエイトの時系列データ(部分:990-1000秒)

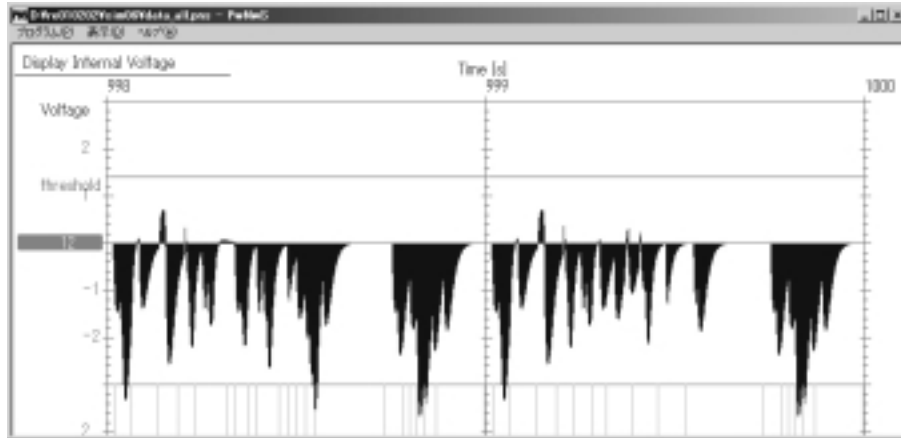


図3.11 内部電圧の時系列データ部分（部分：998-1000秒）

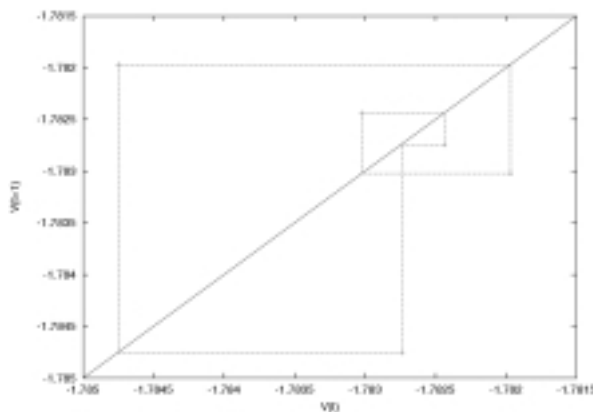


図3.12 内部電圧時系列から作成したらターンマップ

横軸の時間を t 秒目の電圧値とすると、縦軸はその $t + 1$ 秒の電圧値。それを1秒ずつずらしながらプロットしたもの。ほかにも $V(t) = V(t+1)$ のライン（図中の斜線）とそれらの点を時系列に沿うように線で結んだもの（500-1000秒）がある。時系列に沿ったラインが閉じているので、この系が周期的出力をしているとわかる。なお、分析対象としたニューロン12は一切発火していない。このニューロンを分析対象としたのは、発火が少ないために電位が0になる場合が少なく、正確に周囲からの影響を反映していると考えられたからである。

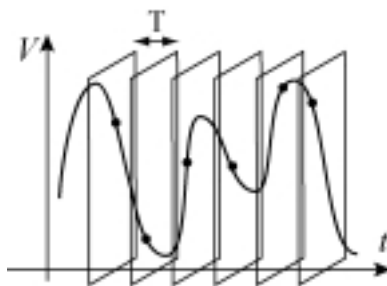


図3.13 リターンマップの作り方

リターンマップは図中のように時系列を等時間間隔にスライス ($n=1,2,3,\dots,N$) し、ある時刻(t)の値とそれより前かもしくは後($t \pm 1$)の値を利用して2次元情報とし、これを平面にプロットする。極簡単なコンテキスト情報を載せているので、異なった値ばかりになるとばらばらに散らばり、周期的な場合はそのうち同じ点を周回ようになる。図3.12を作成するときには入力の周期と同じ $T=1$ で行った。

3.3.3 考察

シナプス可塑性に相当しているウエイトは、はじめ急速に変化していき(図3.9)、やがて周期的な変化

を繰り返す状態へ落ち着く(図3.10)。しかもその変化はシミュレーション開始時と比べて小さい変化におさまっている。また、その現象と同じく、内部電位も周期的に変化するようになり、その周期は入力スパイク時空間パターンの周期と比べて5倍になっていた(図3.12)。こうした現象はウエイト更新規則を適応しないと生じないことから、ウエイト更新規則によってもたらされたと考えられる。

また、ウエイトのグラフ(図3.10)を見ると、各々のウエイトは最大ウエイトや最小ウエイトに達しており、そうした中で、微妙な変化を周期的に繰り返している。だから、より詳しくこれらの周期現象の発生原因を考えると、こうしたウエイトの上下限とウエイト更新規則が関係して生じていると想像される。実際、この上下限の幅を非常に広くした上で、図3.7と同じ構造を用い、図3.8で示したようなスパイク時空間パターンを同様に入力すると、周期的な現象はなくなり(図3.14)、かわりに非周期的な挙動が現れていた。たとえば、図3.12で示したのと同じく、ニューロン12を対象にリターンマップを作成したところ、図3.15のようになったので、非周期的といえる。

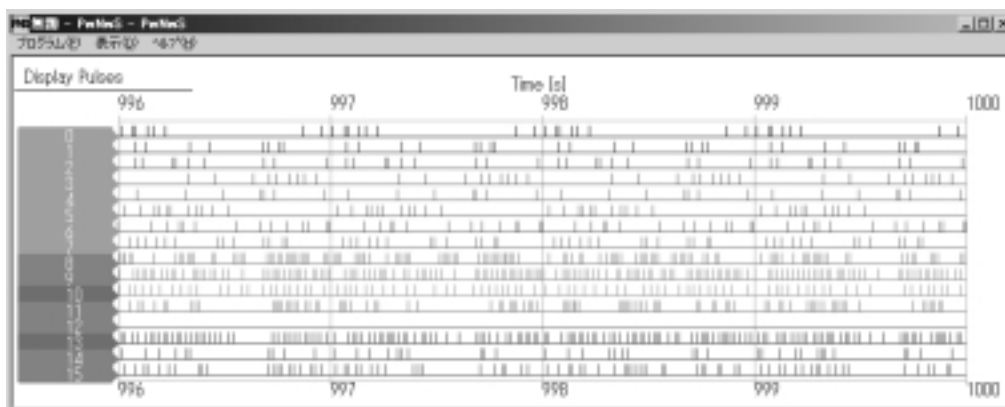


図3.14 周期性を失った時空間パターン(部分:996-1000秒)

ものすごい勢いで発火する細胞が増えているのは、高いウエイトの結合が増えたからである。でもニューロン12は発火していないのには驚きである。

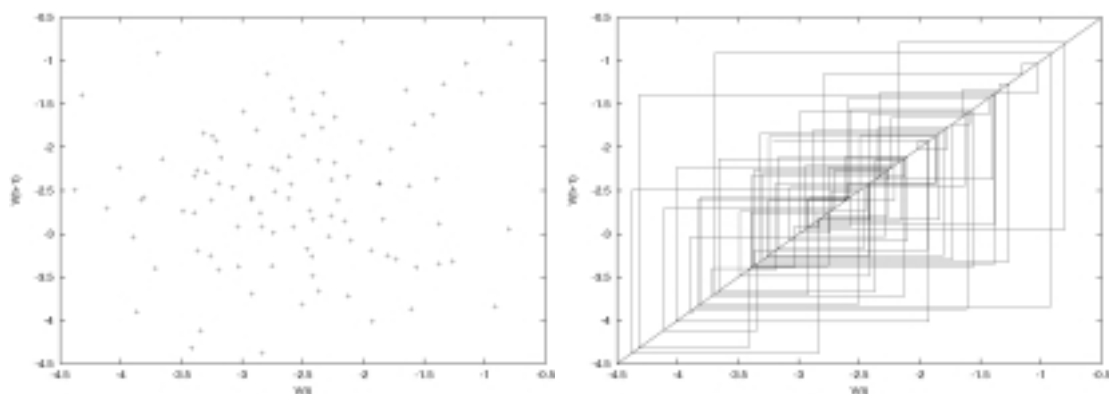


図3.15 ウエイト更新可能範囲を広げた場合の内部電位の挙動

ニューロン12の内部電位時系列を用い、図3.14と同じ時間帯とサンプリング幅でリターンマップを作成した。左は値を素直にプロットしたグラフで、右はそれらの点にたいし時系列を踏まえて線で結んだグラフ。図3.12の実験時と比べウエイト更新可能範囲は約2倍にした。これで周期性を見出すのは難しい。

こうしたことから、今回の実験3の回路と入力のくみあわせによって生じた周期的現象は、ウエイト更新可能範囲の上下限と、ウエイト更新規則の存在によってもたらされたと考えられる。

3.4 実験4 非周期性をもったダイナミクス

3.4.1 ネットワーク構造と入力パターン

回路は実験3と同じ初期構造（入力時空間パターンを除く総てが同じパラメータ）を用いたが、実験3とは異なったスパイク時空間パターンを入力として用いた。そして実験3と同様に、1秒ずつ連続して同じ時空間パターンを与えつづけた（図3.16）。つまり入力の実験3と異なっているが、入力の与え方は実験3と同じである。

3.4.2 結果

実験4では、実験3と同じ条件下のウエイト更新規則が適応されてウエイト更新可能範囲に上下限が設定されていても、必ずしも周期的なダイナミクスを示すに至らない例である。つまり、入力として与えられる時空間パターンが異なっていれば、観測されるダイナミクスが異なっている一例である。

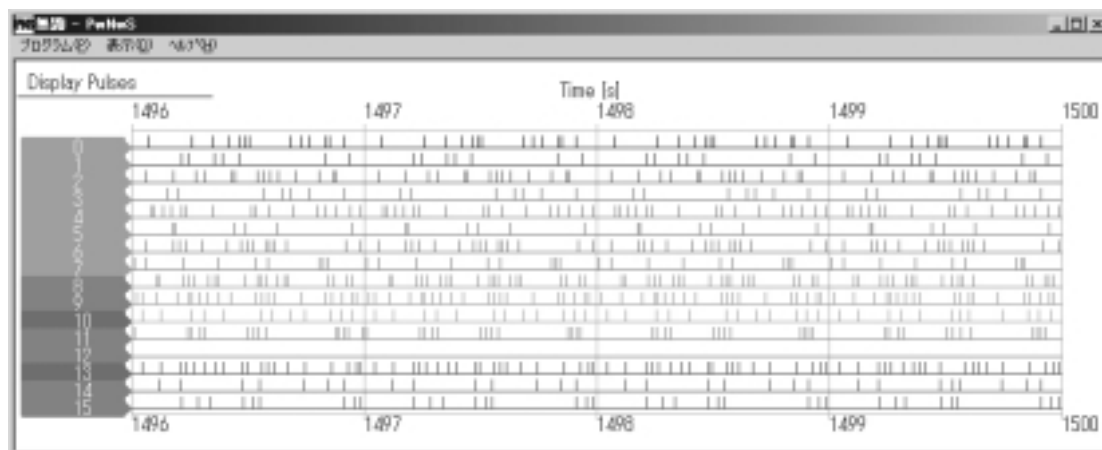


図3.16 実験4で入力した時空間パターンとそれに対する反応（部分：1496-1500秒）

入力はニューロン0からニューロン7に対して与えられている。よってそれらのニューロンのスパイク時系列は1秒ごとに同じ時空間パターンを呈している。これに対し、ニューロン8からニューロン15はそれらの入力と、それぞれが互いに影響しあって発火している。これらのニューロンはおおよそ同じ時空間パターンを出力しているように見えるが、所々で違う時空間パターンが現れている。たとえば、ニューロン10からは1496-1499秒において同様な時空間パターンが1秒ごとに観測できるが、1499-1500秒では、それらとは異なった時空間パターンが現れている。

スパイク時空間パターンを見る限り（図3.6）では、それが周期的であるかどうかはわかりにくい。そこで、ウエイトの変化と内部電位時系列を見る。まず、ウエイトの時系列では、大概は上下限に達していたが、そうした上下限に達していても不規則に変遷する時系列がいくつか存在していた（図3.17）。

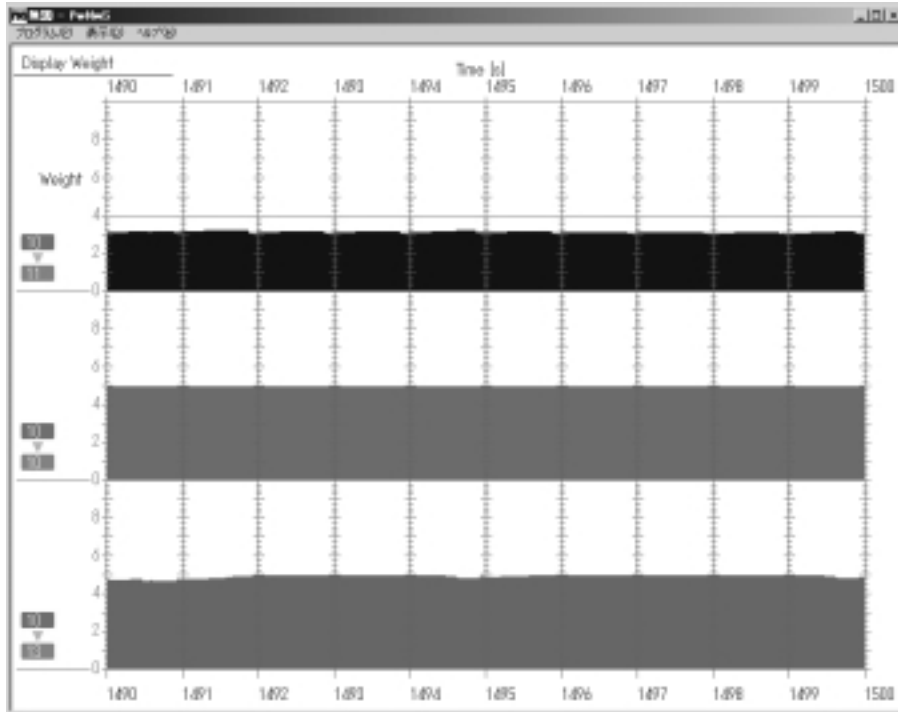


図3.17 実験4で得られたウエイト時系列（部分1490-1500秒）

ニューロン10からニューロン11やニューロン10からニューロン13への投射に課せられたウエイトの時系列は、図中の最下部の時系列である。周期的になりそうでならない現象が確認されている。

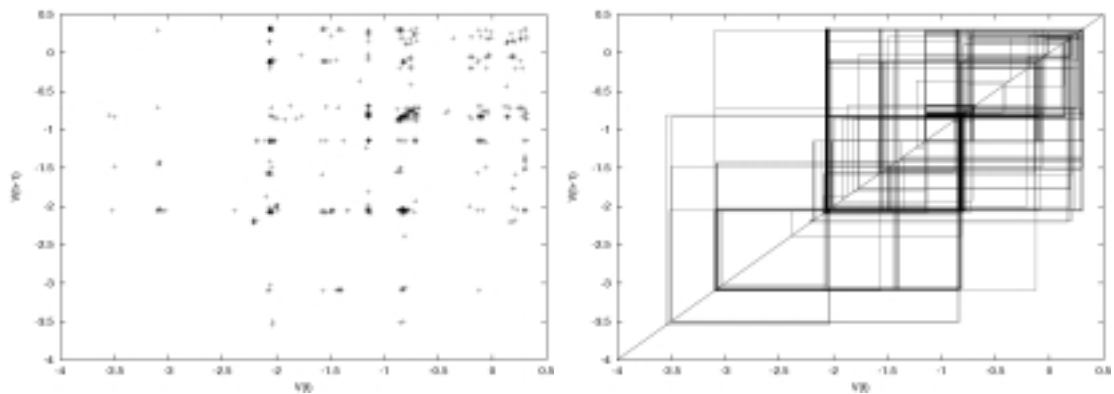


図3.18 実験4におけるニューロン12で観測された内部電位時系列から作成したリターンマップ

左図は観測された値をそのままプロットしたもの。右図はそれらの点にたいし時系列を踏まえ、線で結んだもの500-1000秒間の内部電位を1秒ごとにサンプリングした。時々ウエイトが上下限に達して決まった値をとるせいか、プロットされる点はそれとなく格子状に並ぶ傾向がある。しかしながら、それらの点を経る時系列に周期性を見出すのは難しい。

ニューロン12は投射を受けていても発火しない。だから発火したら消えてしまう別のニューロンから与えられる入力の影響を、内部電圧時系列に反映し続けている(このような特徴をもつニューロンをモニタリングニューロンと呼ぶことにする)。つまり、周囲の影響を長い間保持する性質を持っているので、ネットワークのダイナミクスを調べるうえで、適切な時系列を内部に保持していると考えられる。よって、モニタリングニューロンの内部電位時系列からリターンマップを作成すれば、ネットワークのおおよそのダイナミクスが解析できると考えられる。今回の実験でも、こうしたモニタリングニューロンの特性を踏まえ、ニューロン12の内部電圧時系列からリターンマップを作成し、その周期性を調べた。すると周期を見出すのは極めて難しい状況にあると判明した(図3.18)。

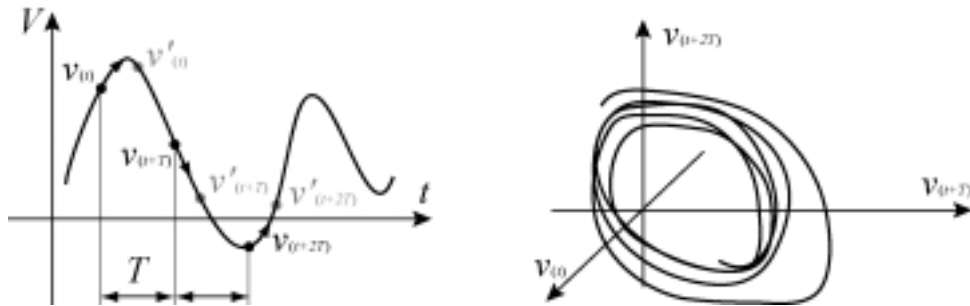


図3.19 アトラクタの再構成とアトラクタ

アトラクタの再構成[18]は、まず時系列データを数点サンプリングし(左図)、その各々のサンプリング値を別の座標系にプロットすることで成される(右図)。ちなみに、このサンプリングは時系列が呈している平均周期と比べて、大凡数分の1程度にするのが良いと知られている。左の図は等間隔Tで3点 $v(t), v(t+T), v(t+2T)$ にわたってサンプリングしており、右図はこれを $v(t)-v(t+T)-v(t+2T)$ 空間にプロットしている。これを時系列に沿って並行移動させながら行くと、アトラクタ(右図のぐるぐる)ができる。なお、サンプリングする点の数は(例では3つ)埋め込み次元と呼ばれ、正確にアトラクタを構成する際は埋め込み次元を徐々に大きくしていきながら調べる方法がとられる。

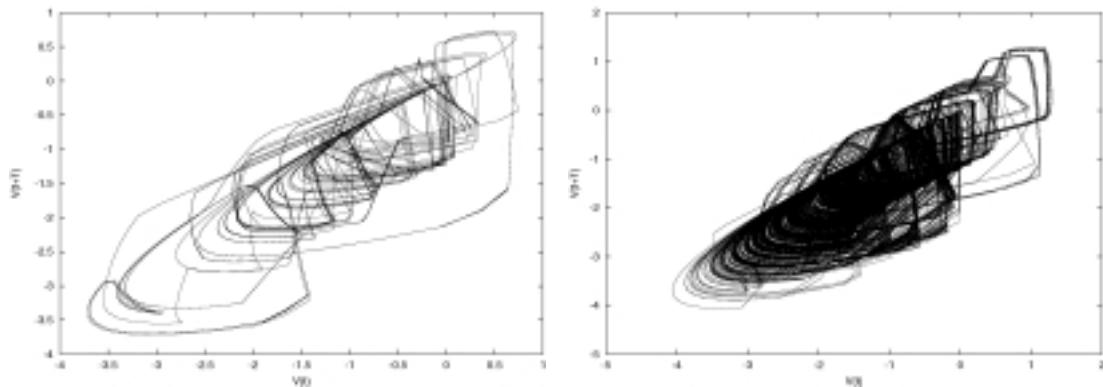


図3.20 実験3と実験4のモニタリングニューロン内部電位時系列から作成したアトラクタ

モニタリングニューロン12から観測された内部電圧時系列を使ってアトラクタを再構成した。観測対象は900から1000秒。サンプリングの幅は0.005秒。このアトラクタは埋め込み次元を調査してから再構成していないため、的確に再構成されたアトラクタではない。しかしながら、時系列が示す内部電位の変遷は、決まった山形をした時系列の高々線形和なので、複雑な構造を持っていないことが想像出来た。そこで、あえて埋め込み次元2で再構成した。左は実験3でウェイト更新規則を適応した時に生じた周期5へ収束しているアトラクタ。右は実験4でウェイト更新規則を適応した場合のアトラクタ。実験4の場合では周期性はなくなり、稠密構造が現れている。

それから、このモニタリングニューロンの内部電位時系列から、強引にアトラクタを作成し、その周期性を概観してみた(図3.19、図3.20)。本研究の目的はカオスか否かを語るといふ点まで突っ込んで議論しないので、あえてリアプノフ指数を計算しないが、カオスでの可能性は非常に高いようである。なぜなら、稠密構造がアトラクタの中にうかがえ、所々に折りたたみと引き伸ばし構造があるように見えるからである。

3.4.3 考察

実験3と違う点は、入力スパイク時空間パターンが異なっている点だけである。しかし、実験3では周期に収束したが、実験4では周期でないか、もしくは非常に長い周期という結果になった。しかもそれらは一例だけではない。実験3、実験4と同じ初期構造でも、入力時空間パターンを変えると、異なる周期性を示す例がいくつか見つかっている(図3.21、図3.22、図3.23)。こうしたことから、入力時空間パターンを変えることが、異なった周期性をともなうダイナミクスを発生させることに相当していると考えられる。

一方、実験3、実験4と同様の初期条件で入力させつづけた場合、周期性を持たないダイナミクスになったが、ウェイト更新可能範囲を狭めて同様にダイナミクスを調べると、入力周期(1秒)に比べ1.7倍の周期に収束する入力時空間パターンもあった。また、実験3の考察で述べたように、もともと周期性を有していても、重み更新可能範囲を広げることによって周期性が喪失する場合があることも示された。それから、本研究の実験として取り上げないが、ウェイト更新程度(式2.5のAの値)を変化させても異なる周期性ダイナミクスが発生することがわかっている。

こうしたことから、今回用いたモデルは、入力の時空間パターンを変えてダイナミクスを変えることもできるが、入力パターンを変えずとも、ウェイト更新可能範囲やウェイト更新程度を変えてダイナミクスの周期性を変えることもできると断言できる。

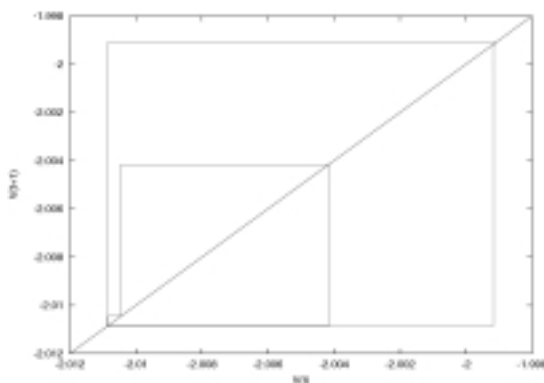


図3.21 入力周期の27倍周期のダイナミクス

実験3と実験4と同じ初期状態(入力時空間パターン以外は総て同じ)を用いて、実験3、実験4とは別の入力パターンを入れた場合の一例。図は、モニタリングニューロンの内部電位から作成したリターンマップ。500-1000秒間を1秒毎にサンプリングした。ほとんど同じ内部電位を行き来しているが、入力周期を1秒とすると、この周期は入力の27倍周期になっている。周期の内訳として、6秒に1回同じ電位を訪れるパターンを4回繰り返し、その3秒後にそれとまた同じ内部電位になる。つまり27周期といっても、周期 $6 \times 4 + 3 \times 1$ という構成を繰り返している。

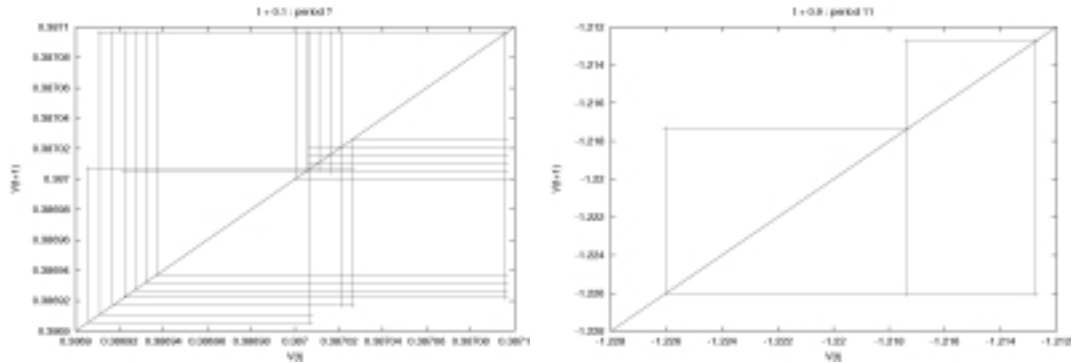


図3.22 テンポラルな周期性ダイナミクス

一時的に周期状態になる例もある。この例も、実験3 実験4と同じ構造であり、実験3 実験4とは異なる1秒単位の時空間パターンを繰り返し入力し、モニタリングニューロンの内部電位時系列(500-1000秒)から1秒ごとに電位をサンプリングしてリターンマップを作成した。右図のリターンマップと、左図のリターンマップは、サンプリングする時間が0.8秒ずれている。左図は周期性を見出すのは難しいが、右図は11周期になっていることがデータからわかっている。しかも右図の周期構成は、入力の周期を周期1とすると、周期 $1 \times 10 +$ 周期1になっている。こうしたことから、周期性が一時的に発生していると考えられる。

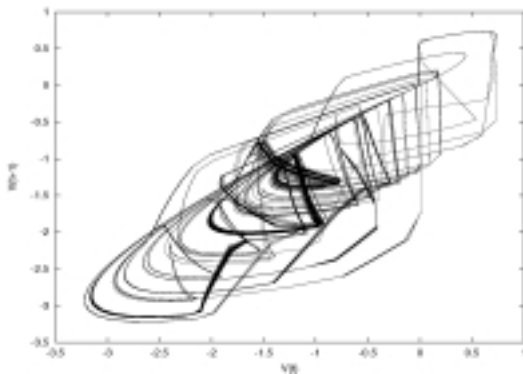


図3.23 図3.22の補足図

図3.24でサンプリングの対象としたモニタリングニューロンの内部電位時系列から、アトラクタを構成した。観測時間は800-1000秒で、0.005秒のサンプリング幅を持たせ、埋め込み次元2で再構成した。

稠密となる軌道部もあれば、稠密でない軌道部もある。こうした現象は、一時的に非周期性や一時的に周期性を繰り返す系で見られると考えられる。時系列全体で非周期的ならば、図3.20(右図)のように稠密構造が全体に現れる。

ちなみに、どのくらいの割合で周期性が生じるかを本研究だけで論ずることはできない。なぜなら莫大な探索空間が存在し、それらをすべて走査できないからである。たとえば、入力時空間パターンの違いや物理構造の違いで異なったダイナミクスが生じるのにくわえ、実験3の結果からもわかるように、ウエイト更新可能範囲を変えただけでダイナミクスが変わる。さらに、観測する時間によっても違うダイナミクスが生じている例すらあり(図2.22、図2.23)また、ウエイト更新程度が異なっても異なる周期性ダイナミクスが生るとわかっている。すなわちこれらの順列組合せをすべてテストするのは、不可能に近い。とって、すぐさま周期性の生じる条件を提示できない。

しかしながら、そうした莫大な探索空間が広がっているにもかかわらず、適当に設定したウエイト更新可能範囲や入力時空間パターンでも周期的な現象を誘発できることから、周期性ダイナミクスが生じる割合は0に近いわけではないと考える。

3.5 実験5 周期性をもつ原因

続く実験では、最も簡単な周期現象に着目し、周期性が生じる原因について考察していく。

3.5.1 ネットワーク構造と入力パターン

回路には、実験3 実験4で利用した物と同じ構造（1秒単位の入力時空間パターンのみが違い、他の条件は総て同じ）を用いた。ただし入力の実験3や実験4とは異なった時空間パターンを用いた（図3.26）。

3.5.2 結果

まずは、観測された結果を示す（図3.24）。しかし図3.24では周期性があるかどうか判別できないので、先の実験と同様にモニタリングニューロン（ニューロン12）の内部電位時系列からリターンマップを作成し、状態どのように遷移しているかをみた。すると、ほぼ入力周期と比べて2倍（正確には6倍）の周期が発生しているとわかった（図3.25）。

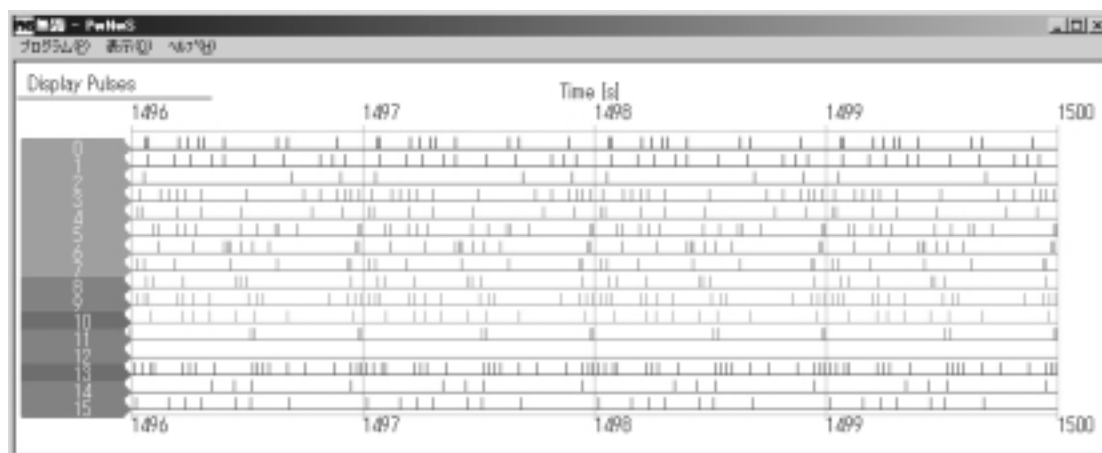


図3.24 実験5の入力時空間パターンとそれに対する反応（部分：1496-1500秒）

入力ニューロンはニューロン0からニューロン7である。入力の実験3と実験4と同様、1秒ごとに同じ時空間パターンを繰り返し与えた。スパイク時空間パターンを見る限り、入力周期と同じ周期のダイナミクスが生じているように見える。しかし、たとえばニューロン8の1496.5秒付近にある3つの連続したスパイクは、1497.5秒付近にある3つのスパイクと違うスパイクインターバルを持っているように見える。実際、データ上では違うスパイクインターバルになっていた。

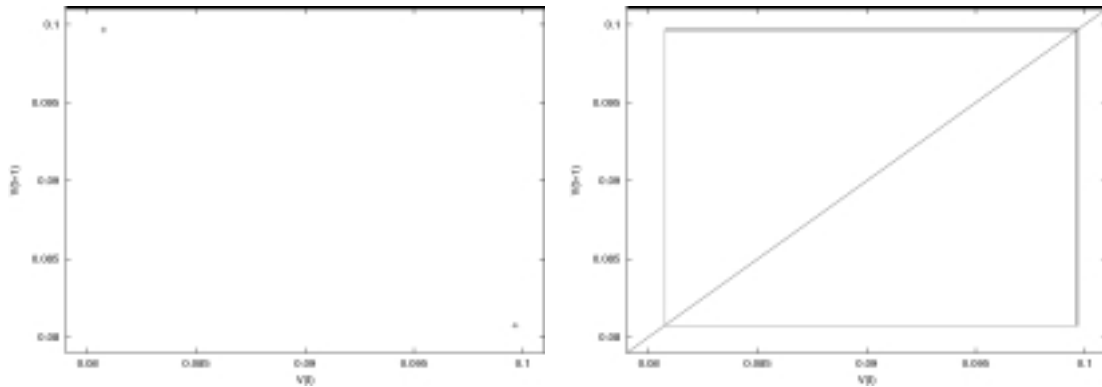


図3.25 実験5におけるモニタリングニューロン内部電位時系列から作成したリターンマップ
 実験3、実験4と同様にモニタリングニューロンの時系列からリターンマップを作成した。すると、ほぼ2状態を交互に訪れているとわかった。なお、正確には周期6になっている。

3.5.3 考察

入力のほぼ2倍周期という、もっとも簡単な周期現象に近いカタチが得られたので、どのようなメカニズムで周期性が生じるかを、この短い周期の例から考察してみる。

今回用いたモデルは、入力時空間パターンが異なっていればそれぞれのニューロンに達するスパイクタイミングが変わり、それに乗じて、それぞれのニューロンで同時刻検出されるスパイク時空間パターンも変わってくる。また、発火タイミングはスパイクの同時検出性と密接な関係があり、同時検出性が変化すれば、発火タイミングも異なり、この差異によって発火に依存して変化するウエイトの時系列パターンも異なってくる。

こうしたことから、もし入力の与えられている周期内でウエイトが変化し、その変化がつぎの周期に影響を及ぼす(図3.26下)のであれば、たちまち系全体の時空間パターンに影響を与えてしまうと想像できる。

ところで、そうしたウエイト時系列で、ある一定期間で元に戻ることを保証されるとなると、そこに周期性が発生する可能性が考えられる。そして、ウエイトの値が元に戻ることを保証する最も簡単な方法は、上下限を設けることである。なぜなら、ある値から移動して上限に達したり、下限に達する場合には、複数のケースが考えられ、それゆえ、それら複数のケースを1点に集約する役目を果たしえるからである。また、こうしたことから、ウエイト限界に達する現象が生じるウエイト時系列が回路にあれば、個々に走り始めた回路上のニューロンの発火タイミング時系列や別のシナプスのウエイト時系列を、ある程度の範囲内ならば、1つの状態へセットしてしまう可能性が想像できる。

実験5では、図3.26(上図)のように、ウエイト下限に達してトグルスイッチ的に変化するウエイトがある。また、上限に達したあるシナプスのウエイト時系列(図3.26(下図))は、2秒周期で変化している。



図3.26 周期性ダイナミクスとウエイトの限界（部分：1490-1500秒）

上図はウエイト下限（上限値約5、下限値約3）に達したウエイト時系列。下図は、上限（上限値約2.5、下限値約1.5）に達した時系列。下図の時系列では、1490-1491で変化したウエイトが1491-1492のフェーズに影響を与えている。

ウエイト時系列とスパイク時空間パターンが、ある条件下（その条件は不明だが）で周期性の状態へ落ちると、内部電圧は周期的時系列を刻むようになると考えられる。

3.6 実験6 複雑な入力と周期性をもったダイナミクス

実験6では少しばかり複数な入力パターンに対する周期性を考える。

また、周期性を誘発させることを優先したので、今までの実験3から実験4とは違うパラメータの組合せで実験した。

3.6.1 ネットワーク構造と入力パターン

用意した回路を、表3.4に示す。具体的な構造を、図3.27に示す。

入力パターンは図3.30のように、周期 $4 \times 1 + 周期1 \times 1$ を用いた。こうした入力パターンをわかりやすくするため、以降は1秒単位の時空間パターンをひとつの大文字アルファベットであらわすことにし、たとえば、周期 $4 \times 1 + 周期1 \times 1$ をAAAABA...（もしくは(4A1B)*）の時空間パター

ンと表現する。

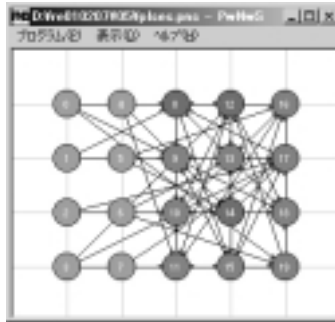


図3.27 実験6のネットワーク

	入力専用	興奮性	抑制性	合計
ニューロンの数	8	9	3	20
1ニューロンあたりの投射数	3	3	5	66

表3.4 実験3で用意した回路構造



図3.38 入力したスパイク時空間パターン

今回用いた入力パターンは今まで実験のように同じ時空間パターンを1秒ごとに繰り返し入力していない。5回に1度、異なった時空間パターンを1秒間入力している。図では、1495 1496 1496 1497 1498 1499秒では同じ時空間パターン(A)だが、1499 1500秒では異なった時空間パターン(B)が入力されている。この時空間パターンシーケンスをAAAABA... (もしくは(4A1B)^{*})と定める。



図3.29 実験6で観測されて時空間パターン（部分：1480-1500秒）

ニューロン8からニューロン19は、入力専用ニューロンではない。観測された時空間パターンは、目視で確認する限り1480-1485と1490-1495は同じ時空間パターンを持っているし、1485-1490と1495-1500の時空間パターンは同じ時空間パターンのように見える。ちなみに、ニューロン19は5秒おきに同じ時空間パターンを繰り返しているようだ。

ところで、パルス時空間パターンが周期的か否かを分別する作業は、単に目視のみで行っている。スパイク時系列を図のように表示させ、それを一定期間ごとにスクロールしていきながら連続的に表示させると、それが周期的になっているかそうでないかが、残像によって確かめられるのである。

3.6.2 結果

図3.27にあるようなネットワークに、図3.28で示したような(4A1B)*の入力を1500秒与えると、図3.29に示すような周期的なスパイク時空間パターンが観測された。これを入力(4A1B)*と同じように表現すると(CC'C''C'''DEE'E''E'''F)*という時空間パターンの繰り返しとして表現できる。

また、モニタリングニューロンとしてニューロン13を選択し、このニューロンの内部電位時系列からリターンマップを作成すると、10状態を繰り返し変遷していた(図4.8)。

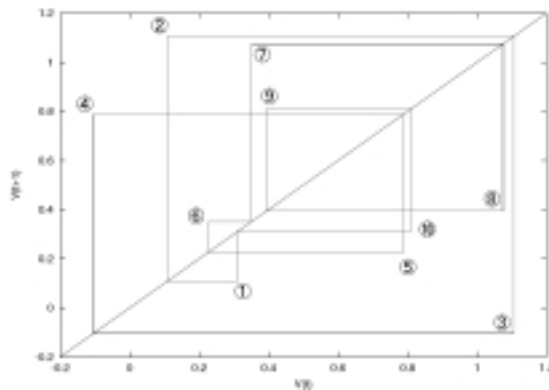


図3.30 モニタリングニューロンとしてニューロン13の内部電位時系列から作成したリターンマップ
ニューロン13の内部電圧時系列から作成したリターンマップ。ダイナミクスとして、周期10があることを物語っている。ただ、 から へ変遷する場合で微妙ながらぶれが存在している。

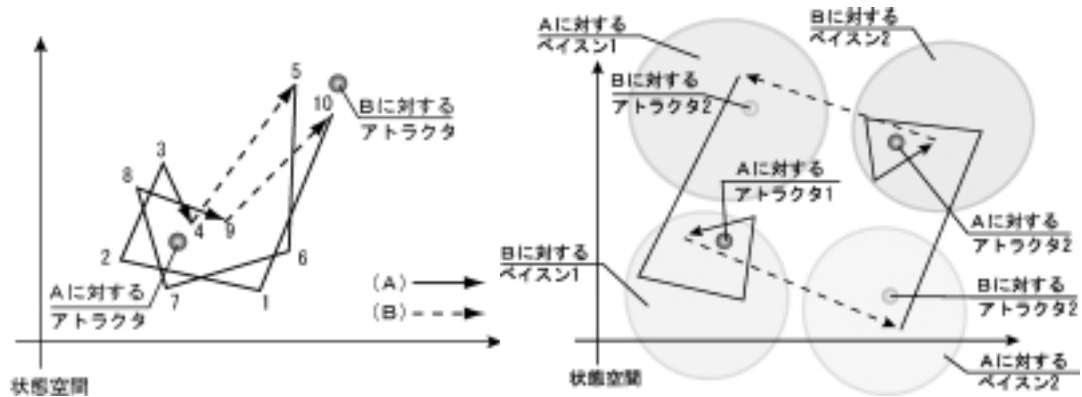


図3.31 アトラクタの概念を使った実験6の考察

もし、Aのみが入力されているなら、Aに対するアトラクタに収束しようとする。一方、Bのみが入力されたらBのアトラクタへ収束しようとする。こうしたことから、都合よく左図のようなループ構造(アトラクタを変遷するのアトラクタ)が得られれば、今回実験6のような結果になると考えられる。

もしくは、アトラクタやそれに対するベイシスが複数存在していて、右図のような変遷を経て、アトラクタを変遷しているかもしれない。

4. 考察

考察では、3章で扱った実験のまとめと実験全体から得られた考察を行い、この研究の出発点である Vaadiaの研究[1]と本研究の関係について述べる。

4.1 実験結果からの考察

4.1.1 スパイキングニューラルネットダイナミクスとパラメータ

本研究で用意したモデルを用いて計算機シミュレーションすると、たとえ全く同じパラメータを有したネットワーク構造を用意しても、入力する時空間パターンが異なっていれば、異なった周期性のダイナミクスを発生させられると示めせた(3.3節 3.4節)。また、周期性のダイナミクスは、入力パターンを固定して、ウエイト更新可能範囲の幅やウエイト更新程度を変えることによって変化した(3.3節 3.4節)。そして、ダイナミクスが時間軸方向でも異なっていて、それが周期的に繰り返されている場合も存在していた(3.4節、図3.24、図3.25)。

こうした事実から、今回のモデルは様々なパラメータもち、これらを調節することによって様々な周期性や非周期性を持ったダイナミクスを発生させられると考えられる。

一方、ウエイト更新範囲を広げると、周期性がなくなって非周期的挙動になり、逆に非周期的なものはウエイト更新範囲を狭めると周期性をもったダイナミクスになる傾向があることから、そもそも、周期性ダイナミクスに至る原因が、ウエイト更新可能範囲の限界性にある可能性が非常に高い(3.5節)と考えられる。

しかしながら、任意の周期を発生させるためにどのパラメータをどれだけ動かせば良いのか、これはわからないままである。

4.1.2 スパイキングニューラルネットダイナミクスと情報処理

4.1.2.1 周期性

パラメータを調節して任意の周期を発生させることもできないが、どれだけの種類の周期を発生させられるか、それを今回の研究から探り当てるのもまた、難しい。だが、図4.1のようなことを考えると、今回使用したモデルからあらゆる周期を発生させられる可能性を示せる。

本研究のモデルは単純な写像モデルではないが、ウエイトを離散的に変化させている点で写像的要素がある。そこで、1秒単位の時空間パターンをネットワークに入力する手続きを、写像 F を1回だけ適用することに対応させ、これを $F_{(1sec)}$ などと表せば、3.3節の実験3で観測された周期5の現象を $F_{(1sec)} \cdot F_{(1sec)} \cdot F_{(1sec)} \cdot F_{(1sec)} \cdot F_{(1sec)}$ で一周する周期軌道持つと説明できる。なお、この写像の定義域と値域は、共にウエイト変更可能範囲の上下限内である。

実際には、時間軸に沿ってダイナミクスそのものを周期的に変えるような系があるので、単純に写像の概念を導入するのは誤りだが、ウエイト時系列のどの時間からサンプリングを開始しても(サンプリングする時間幅はいかなる場合で同じ)、同じ周期性が見出せるのであれば、単純な写像の話へ落とし込んで差し支えないと考える。

3	5	7	9	11	$2n+1$...			
$2 \cdot 3$	$2 \cdot 5$	$2 \cdot 7$			$2(2n+1)$...			
$2^2 \cdot 3$	$2^2 \cdot 5$	$2^2 \cdot 7$			$2^2(2n+1)$...			
$2^m \cdot 3$	$2^m \cdot 5$	$2^m \cdot 7$			$2^m(2n+1)$...			
.....	2^k	2^{k-1}			16	8	4	2	1

図4.1 シャルコフスキーの定理

今、写像 f において、 n 周期点を持てば必然的に m 周期点を持つようになっているとき、この関係を $n \ m$ と表現すと、シャルコフスキーの定理は図のような周期点の関係を持つと主張している。つまり、シャルコフスキーの定理によれば、写像 f が周期3を有すると、その写像は総ての周期軌道を持つことになる。

このように、今回使用したモデルにおける写像性と、シャルコフスキーの定理（図4.1）からすれば、3章実験3の周期5の存在は、非常に多くの周期を発生させられる可能性を示唆している。

一方、3章実験4で観測されたダイナミクスがカオスの可能性も示唆されている。よって、この系の時系列から正確にアトラクタを再構成[18]するか、時系列に摂動を与えるなりして、リアプノフスペクトルを計算し、スペクトルの最大値が正を上回っているかどうかを調べ、カオスが否かを判定し、その結果からあらゆる周期を発生させられるか否かを厳密に示せると考えられる。

4.1.2.2 周期性と情報処理性

仮に、あらゆる周期を誘発できるなら、いくつかの情報処理性が思いつく。たとえば、数え上げるとか、 n 進数の桁上がりなどである。それから、ある入力を与えられたときに決まった時空間パターンを出力する（周期が1であったような場合）なら、単なる記憶として利用できるかもしれないし、周期的に出力が変遷するのであれば、時間的前後関係の情報をその周期分だけ反映させられる事ができるので、ごく短い過去の単純な情報（モデルではネットワークのすべてのウエイト状態とスパイクの位置）を表現できるかもしれない。

また、こうした桁上がりやカウンタのような機能を使ってコンピュータが動作している（コンピュータは2進数を利用している）ことを考えると、コンピュータ相應の機能が、こうしたスパイクングニューラルネットを使って実現できる可能性があるとも考えられる。

4.1.2.3 パラメータと情報処理性

入力される時空間パターンに依存してシステムの周期性が変化することを外部依存性ダイナミクスとして位置付けるのなら、ウエイト更新可能範囲やウエイト更新程度の変化によって周期性が変わる現象を内部依存性ダイナミクスとして位置付けられるかもしれない。ちなみに、ここで示した外部、内部というのはスパイクングニューラルネットの視点から見た場合に生じる区別である。

もし、生物の神経回路網において、本研究のモデルに付与したようなウエイト更新程度やウエイト更新可能範囲に相当するパラメータが存在し（4.1.4節参照）、これらパラメータが生物内部の何等かの要因で変化するとしたなら、外部からの信号によってもたらされる生物の内部変化や出力を、ある程度生物自身が調節できることに対応付けられる。解剖学的に見出された事実ではないが、今回のモデルで表

せるような外部依存性ダイナミクスと内部依存的ダイナミクスの相互作用性は、外部からの入力に依存し、生物が有効な出力を決定するために、何らかの形で内部依存性ダイナミクスのパラメータを変え（4.1.4節参照）、行動の調節や試行錯誤を行っていると思像できるからである（類似的な現象と考察として[28][29]）。

4.1.3 現象の数学的解釈

スパイクニューラルネットワークは、スパイク時空間パターンを受け取っては別の時空間パターンを吐き出している。こうした現象を数学的に考えるため、例えばネットワークの状態 a にウエイトの写像要因でもある時空間パターン F を代入して、別のネットワークの状態 b になり、写像要因 G （これもスパイク時空間パターン）を出力していると思なす。そして、そうした変換がネットワーク内部に実装された重み更新規則 X とネットワーク構造 N によってもたらされているとし、時間についてはとりあえず考えないとする。すると、スパイクニューラルネットワークの入出力を（式4.1）のように表現できると考えられる。

$$(a, F) \xrightarrow{X, N} (b, G) \quad (\text{式4.1})$$

（式4.1）からもわかるように、今回観測されたさまざまな現象を数学的に説明するには、写像の写像（関数の関数）といった写像や関数の存在と性質を考えなければならない。こうした数学は非常に難しいので、今回観測したダイナミクスを数学的に解析するのは当然難しいと考えられる。

4.1.4 モデルのウエイト更新規則と実際の細胞のシナプス可塑性

4.1.4.1 モデルのウエイト更新規則

実際のニューロンの活動として、極めて動的にシナプスウエイトを変更していることが示唆されている[19][20][21]（[21]は海馬からの知見である）。これらの知見によって、動的シナプス可塑性を利用した本研究の妥当性が保証されている。

ダイナミックに変化すればするほど、すなわちシナプスウエイト更新程度が大きければ大きいほど、状態の急激な変化が観測されるようになり、より周期的ダイナミクスが発生しやすいようになっている傾向があった。

4.1.4.2 ウエイト更新可能範囲

ウエイトに上下限が存在している事実は、生物の立場からして現実的設定である。なぜなら、無限に大きくなる細胞がないのと同様、無限に大きくなるシナプスウエイトは存在しないと考えられるからである。しかしながら、モデルに付加したウエイト上限に関してはこうした直感的考察からそれではないかもしれないが、ウエイト下限には疑問がのこる。何故なら、今回下限として設定した値は切断されない、すなわちゼロにならないからである。つまり1度接続されたシナプスは永遠につながれたままと設定されており、発火に対して無駄な入力をどれだけ受け取ろうとも、ある一定以下にはウエイトが弱められないと仮定されているのである。だから、発火とは無縁の入力を伝播させてくるシナプスが容赦なく消去されると、解剖学的示唆があれば、今回設定した条件は誤っていることになる。

ただ、こうした周期性ダイナミクスの生じる原因が、なんらかの形でウエイト更新可能範囲の限界に（上限のみでも）達していれば、今回のようなあらゆる周期を生み出すようなダイナミクスが生じると考えられるので、仮説としてつぎを立てられる。

「神経回路において、ある細胞へ周期的な時空間パターンを与えたときに、それとは別の細胞から周期的時空間パターンを観測したならば、それらを接続しているシナプスのなかに、シナプスウェイトに相当する値が上限もしくは下限に達しているものがある。周期性は、それらシナプスウェイト限界値とそうでない値を交互、もしくは周期的に変遷しているために生じる」

なお、老朽化という観点を除いて、下限が非ゼロであるという知見が得られるとするなら、非ゼロになる理由として以下に列挙する原因が考えられる。

- ・ 細胞は、発火をもたらすものだけを必ずしも待遇しているわけではなく、発火機会を与えてくれるものにたいしても友好的である。つまり発火と無縁の入力が多くても、チャンネルを開ける機会を与えてくれることには変わらないので、ある程度のシナプス可塑性を残しておく。
- ・ シナプスを弱めるものと、シナプスを強める要因は絶えず何等かの形で働いていて、ウェイトの値はそうした要因がせめぎあった結果生じている。だから、そうした強弱要因がどちらともゼロでない限り、非ゼロの値でウェイトが平衡する。
- ・ 細胞自身が生産できる細胞の部品量に限りがあると考えれば、管理できるレセプターやチャンネル数に限りがあるかもしれない。そして、もしこうした状況で過剰な投射を受けているとなると、細胞は発火しやすさという効率を求めて、部品や資源配分という非線形計画問題を解決しなければならないようになる。そうした結果、強めたままでありたいものでも、相対的に弱めなければならないものが生じえる。それが下限性（上限性もふくめて限界性）の役割を果たすかもしれない。

4.1.4.3 ウェイト更新程度

本研究では、ウェイト更新程度とネットワークダイナミクスとの関係を詳しく研究していない。だが、生物学的観点からすれば、シナプスウェイト更新程度がシナプス周辺や細胞の状況に応じてダイナミックに変化している可能性は十分考えられる。たとえば、細胞の活動が何らかの形で抑制されているなら、ウェイト増減に使われる物質が細胞中心部のほうからスムーズ運ばれてこない関係で更新程度が鈍くなると想像できるし、逆に細胞の活動が何らかの原因で活性化されているのなら、抑制されている場合とは逆で、シナプス可塑性の強化や退化作業が効率的に行われるかもしれない。

いずれにしても、神経伝達物質やホルモン[22]によって神経細胞の活動が抑制されたり、興奮させられたりする場合は頻繁にある。それにもかかわらず、シナプスウェイトの変化程度がまったく変化しないとは考えにくい（つまり、シナプス可塑性が変わりえる）。また、ホルモンは身体からの情報を伝達してくる役割を持っていて、そうしたものが神経回路網に影響し、さまざまな感情を生み出していると考えられる立場がある。たとえばDamasio[18]が考えているような身体と脳との関係がそのひとつである。

こうしたことから、ウェイト更新程度をパラメータとして据えた場合のダイナミクスについて考えることは肝心なことかも知れない。

4.2 Vaadiaらの実験と本研究の接点

Vaadiaらはサルにある仕事を学習させた後、前頭葉にある神経細胞を生きのまま観測した。その結果、

ある独特な発火現象が起こっているのを発見する。この節では、本研究に関する話題を一旦隅に置き、Vaadiaらの実験について簡単に述べ、その後、本研究の実験と彼らが行った実験の関係について述べる。

4.2.1 Vaadiaらの実験内容

Vaadiaらは、ある2種類のタスクを繰り返し連続して行えるようにサルに学習させた。説明の都合上、その2種類の一方を「GOパラダイム」、もう一方を「NO-GOパラダイム」と呼び、それぞれを区別しない場合を単にパラダイムと呼ぶことにする。

ところで、これらのパラダイムは図4.2のような構成をしている。

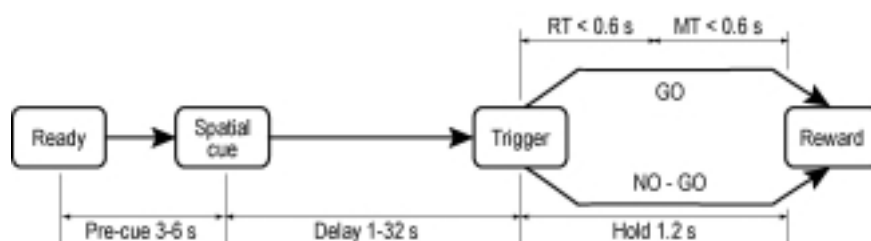


図4.2 Vaadiaの実験パラダイム

サルは自発的にセンターキーを押す（レディーランプが点灯する）。その3～6秒後、実験実行者側から「GO」タスクをするか、「NO GO」タスクをするかが目の前に設置したタスクランプによって知らされる。そのタスクの内容が提示されてから更に、1秒から32秒して実行ランプ（トリガー）が点灯すると、提示されたタスクの内容にしたがってサルはタスクをこなす。GOならばセンターキーから0.6秒以内に手を離し、つぎの0.6秒内でターゲットキーを押す。NO-GOならばセンターキーを1.2秒間押したままている。これらのタスクが成功すれば、報酬がサルに与えられる。

また、サルはパラダイムをある秩序だたくみあわせで複数回実行するようにも学習させられる。そのパラダイムのくみあわせとは図4.3に示すようなパターンである。



図4.3 Vaadiaの実験の連続パラダイム

図の四角はパラダイムの単位。GOとNO-GOパラダイムのうち一方を4回連続して成功した場合、次から別のパラダイムを4回連続成功するようにし、そして、またもう一方のパラダイムを連続して4回目成功するようにして……という具合に、連続交互4回ずつ行うよう学習させられる。

ところで、連続4回パラダイムを成功させた場合にのみ「パラダイム変更」のシグナルを、図のようなタイミング（special peripheral lights）で提示されるようになっている。

学習を終えたサルの運動前野から適当に2つのニューロンを選び、Vaadiaらは発火パターンを観測した。測定データはJPSTH(joint peri-stimulus time histogram[24][30])にかけられ、2つのニューロンの発火タイミングがそれぞれどのような相対的時間差を持っているか調べられた。その結果、GOパラダイムの場合とNO-GOパラダイムの場合とで、発火数の平均値に優位な差は認められなかったが、2

つのニューロン間の発火タイミングがずれているニューロンの組合せが発見された（図4.4、図4.5）。

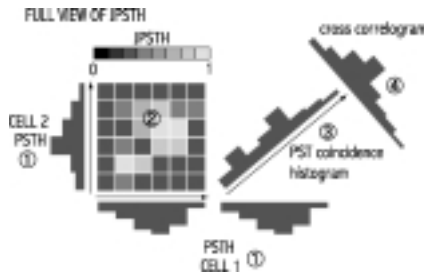


図4.4 JPSTHの概要

JPSTHは、2つのニューロンから観測されるスパイク時系列から、発火タイミングによる同期性、発火平均、ニューロン間の相関がどのようになっているかを調べる時に使う。

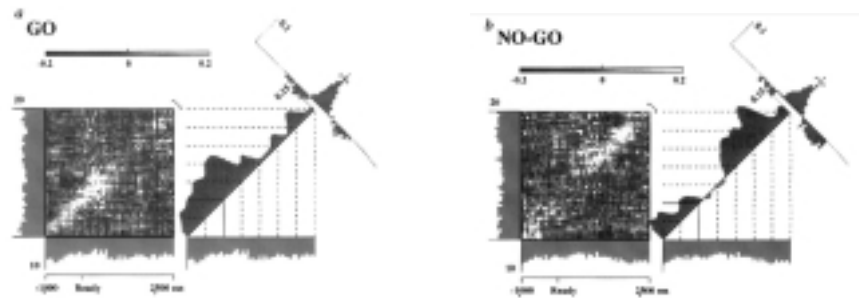


図4.5 2つのニューロンの発火タイミングとVaadiaの実験

Vaadiaらは、タスクができるようになったサルの前野から2つのニューロンを選択し、それぞれのスパイク時系列をJPSTHにかけた。観測対象となった時間帯は、レディーの1秒前からレディー後2.5秒の計3.5秒間である。各値は、数100回の試行を重ねて平均されている。ちなみに、2つのニューロン間の発火同期性を、図4.4に示すのマトリックスと、のヒストグラムから読み取ることができる。のヒストグラムは、高ければ高いほど、2つのニューロンが同期的に発火したことを示している。

GOの場合は観測時間帯の前半で同期性が高まっているが、NO-GOの場合は、観測時間帯の後半の方で高い同期性が現れている。平均発火率に関しては図4.4におけるに相当する部分で示されているが、GO、NO-GOで区別するのは難しい。

このVaadiaらの論文が参考文献として取り上げられる訳は、発火の平均値では情報処理の切り分け（今回の場合はGOかNO-GOかを分けること）ができなくとも、2つのニューロンの同期的発火パルスタイミングを考慮すると切り分けできるという具体的な現象を取り上げたからである。

4.2.2 Vaadiaらの実験と本研究のモデル

まず、Vaadiaらの実験でサルが獲得したタスクの流れを捕らえなおす。そのために、図4.2、図4.3で示した作業手順を誤りなくこなせるようになったサルの脳で何が生じているかを考える。

タスクを習得しているならば、サルはほとんどのパラダイムで失敗しなかったと思われる。そして、タスクを遂行しているときは、タスクにかかわる脳の各部位で同様なスパイク時空間パターン繰り返されていたと想像できる。

また、サルはGO NO GOのパラダイムの内容のみならず、パラダイム変更信号の意味も獲得していたと仮定する。つまり図4.3で示した特別ライトが点灯するのは、つぎからタスクのパラダイムが変わるからだ、サルが理解していると考える。

こうなると、サルの脳で観測される時空間パターンは、近似的に図4.6のような流れになる。

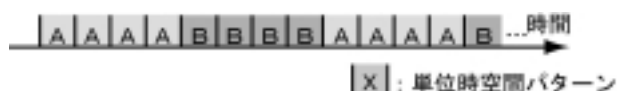


図4.6 学習したサルから観測されると思われる近似的な時空間パターンシーケンス
運動前野や運動野を観測していて、同じような時空間パターンが観測されたら同じような動作をしていると仮定するなら、Vaadiaの実験で学習したサルは、決まった仕事を繰り返していることになるので、図のようなシーケンスが近似的に観測されるはずである。つまり4回連続交互に同じような時空間パターンが繰り返されていると考えられる。

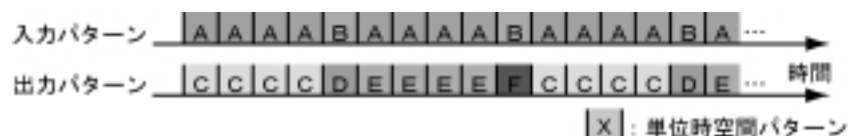


図4.7 special peripheral lightsを考慮した近似的時空間パターンのシーケンス
Vaadiaらが観測した時間帯は、GOパラダイム、NO-GOパラダイムとも同じ(レディーをする)動作を行っているときなので、入力として得られる時空間パターンに大きな区別はない。だから入力されてくる情報に対して特異的に反応する部位は、同じような時空間パターンが繰り返し観測されると思われる(図中Aのシーケンス)。しかしながらAに反応する部位が、special peripheral lights(図中Bの時空間パターン)にたいしても特異的に反応するならば、その出力パターンは下図のような出力パターンが現れるはずである。要するに、special peripheral lightsが点灯するたびに、CからD、DからCというように出力される時空間パターンがスイッチングするはずである。こうして入力の影響を受けて出力パターンが変わることを、モードがシフトと呼ぶ。モードとは、ある入力値aに対してある出力値bを定めている「システムの状態」を指す。モードがシフトするとは、その入力aが変わってなくても、出力値を変え、たとえば出力値cになる「システムの状態遷移」を指している。

また、Vaadiaが観測した時間帯を考えると、同期的なスパイクタイミングがずれる原因が、トリガー後のGOやNO-GOの作業へ分岐したためでなければ、GOやNO GOの指示信号(図4.2のspecial cue)が与えられたためでもなく、サルがそうした指示信号を受け取る前から、行うタスクの内容がGOかNO-GOのいずれかがわかっていたことによって生じていた現象と考えられる。そして、そうした予見は、パラダイム変更信号の存在(図4.3のspecial peripheral lights)によってスイッチされている可能性が一番高いと考えられる。

一方、脳には独特の入力に対して異なる部位が反応する反応の局在性がある。とりわけ視覚情報に関してよく研究されている[25]。今、この反応の局在性踏まえ、たとえば図4.3のspecial peripheral lightsに特異的に反応する部位から投射を受けているようなネットワークを想定すると、図4.7のような時空間パターンの変遷が想像できる。

ところで、図4.7のシーケンスは3.6節の実験6の内容に似ていることが読み取れる(図4.8)。こうしたことから、学習したサルの脳内で、実験6で提示したようなネットワークの存在が考えられと同様の現象が発現していた可能性がある。

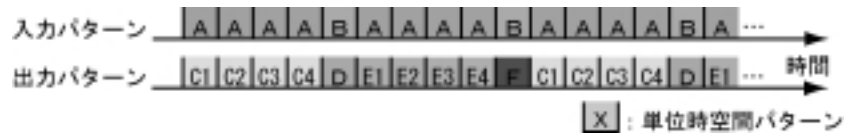


図4.8 実験6で観測された時空間パターンシーケンス

3.6節の実験6で観測された入力専用ニューロン以外のもから観測された時空間パターンは図中「出力パターン」で表した時空間パターンシーケンスをもつ。これは、厳密にいえば図中4.7の出力パターンと同じでないが、モード（図4.7）を区別できる時空間パターン持つ意味で似た状況である。

ちなみに、今回のモデルだと、モード（図4.7で説明）に依存して同期的に発火するタイミングを変える（発火の平均値で差は生じない）ニューロンの組合について議論できない。なぜなら、今回使用したモデルでは出力される時空間パターンに分散が生じないため、試行結果の平均を取ること、スパイクのカウント数にたいして平均を取ることにも意味がないからである。よって、Vaadiaらの研究結果とこの結果を比較し、発火の平均値と同期的発火のどちらが、より情報処理性を反映しやすいか評価できないのである。

しかしながら、図3.29の示されているように、モードによって異なるスパイク時空間パターンを出力するような近似的なモデルとしてなら、本研究のモデルを位置付けることはできる。だから、もしこうした現象が脳の各所で存在しているなら、当然、そうした部位Aから投射を受けている別の部位Bは、異なる時空間パターンに反応する可能性があるわけなので、部位Bからも異なる時空間パターンが出力されるようになる。そして、そうした時空間パターンを出力するニューロンの中には、実験5で示したようなメカニズムによって、平均発火率に差はなくとも、同期するタイミングを一時的に変えるようなニューロンのくみあわせが存在するかもしれない。

5. 結 論

本研究では、スパイクニューロンモデルにスパイクタイミング依存シナプス可塑性を付与し、ニューロンモデルを相互結合させてネットワークモデルを作成した。そしてこのネットワークモデルを計算機に実装し、シミュレーション実験した。このときネットワークモデルに与えた入力は、脳の皮質で観測されているスパイク時空間パターンを考慮したものとした。

スパイクニューラルネットワークを計算機に実装してシミュレーションしたところ、同じパラメータに設定したネットワーク構造（ウエイト更新規則のパラメータも変わらない構造）であっても、異なった時空間パターンを入力すると、異なった周期性を持ったダイナミクスが観測された。そして、入力時空間パターン別で異なっていた周期性ダイナミクスのなかには、さまざまな周期を有するダイナミクスや非周期を持ったダイナミクスがあった。よって、本研究で用いたモデルは、たとえ同じネットワーク構造を初期に有していても、入力によって様々な周期を有するダイナミクスを発生させられるモデルと位置付けられる。そして、異なる周期性が得られる現象は、ウエイト更新規則を適用しない場合では生じなかったことから、ウエイト更新規則の存在によってもたらされた現象と結論付けられる。それから、シナプスウエイトの限界性に依存して生じていることも示した。

また、同じネットワーク構造（ウエイト更新規則のパラメータは変わる構造）で、同じ時空間パターンを入力していても、シナプスウエイト変更範囲やシナプスウエイト変更程度が異なっていれば、異なった周期性ダイナミクスが生じるともわかった。よって、ウエイト更新規則が持っているパラメータを変更させても、ダイナミクスを変えられるモデルとしても位置付けられると示めた。

しかしながら、任意の周期を発生させるための具体的なパラメータについて言及することができなかった。現在のところ、周期を有したダイナミクスが生じるか否かを実験して確認するだけであり、これらのダイナミクスを数理的に説明できない。今後はこうした現象を数理的に解析されることが望まれる。なぜなら、周期を任意に発生させることが、本研究で使用したスパイクニューラルネットワークを制御することに対応すると考えられるためである。

一方、Vaadiaらがサルの大脳皮質で発見した神経細胞の発火現象を、本研究で作成したモデルを使って説明を試みたが、本研究で用いた理想状態のモデルとVaadiaらが観測した皮質の状況とに落差があったため比較ができなかった。しかしながら、本研究で使用したモデルを用い、Vaadiaらが発見した発火現象の原因については言及した。

謝 辞

指導教官であられます櫻井彰人教授からは、研究内容に対する鋭い指摘を賜り、より洗練した研究内容にする機会を与えてくださいました。別の研究室に所属しておられます橋本敬助教授からは、複雑系解析の立場から解析に関する助言をいただきました。また、櫻井研究室に所属されている皆様や、他の研究室に配属となっている方からも、さまざまな助言をいただき、議論もさせていただきました。それから、櫻井研究室の荒木修助手は、多角度からの助言や資料、多くの知識を与えてくださいました。とくに、本研究の出発点でありますVaadiaらの論文も、スパイクニューロンモデルの教科書も荒木助手から紹介していただいたものでした。

本当に大勢の方からさまざまなモノをいただきました。皆様にこの場を借りて御礼申し上げます。最後に、大学院在学中の生活支援をしてくださった家族に感謝を表します。

参考文献

- [1] E. Vaadia, et. al., Dynamics of neuronal interactions in monkey cortex in relation to behavioural events, *Nature*, 373, 515-518 (1995).
- [2] M. F. Bear et. al., Synaptic plasticity: LTP and LTD, *Current opinion in neurobiology*, 4, 389-399 (1994).
- [3] T. V. P. Bliss et. al., S synaptic model of memory: long-term potentiation in the hippocampus, *Nature*, 361, 31-39 (1993).
- [4] H. Markram, et. al., Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275, 213-215 (1997).
- [5] J. C. Magee et. al., A synaptically controlled, associative signal for hebbian plasticity in hippocampal neurons, *Science*, 275, 209-213 (1997).
- [6] S. Song, et. al., Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3, 919-925 (2000).
- [7] L. F. Abbott et. al., Synaptic plasticity: taming the beast. *Nature Neuroscience supplement*, 3, 1178-1183 (2000).
- [8] A. Aertsen et. al., Dynamics of function coupling in the cerebral cortex: an attempt at a model-based interpretation, *Physica D*, 75, 103-128 (1994).
- [9] M. Adachi et. al., Associative Dynamics in chaotic neural network, 10,83-93 (1997).
- [10] W. Gerstner, Population dynamics of spiking neurons: first transients, asynchronous states, and locking, *Neural computation*, 12, 43-89 (2000).
- [11] W. Gerstner, Pulsed neural networks , chapter 1,10, 3-53, The MIT Press (1998).
- [12] A. L. Hodgikin et. al., A quantitative description of ion current and its applications to conduction and excitation in nerve membranes. *J. physiol. (London)* 117: 500-544 (1952).
- [13] D. O. Hebb, *The Organization of behavior*, Wiley (1949); 白井(訳):行動の機構, 岩波書店 (1957).
- [14] R. Douglas et. al., *The synaptic organization of the brain*, 459-510, Oxford university press (1998).
- [15] M. Abeles, *Corticonics; neural circuits of the cerebral cortex*, 52-56, Cambridge university press (1991).
- [16] W. R. Softky et. al., Cortical cells should fire regularly, but do not, *Neural computation*, 4, 643-646 (1992).
- [17] L. P. O'Keefe, Response variability of MT neurons in macaque monkey. *Society for Neuroscience abstracts*, 23, 1125(1997).
- [18] S. Sato et. al., *カオス入門*, 108, 培風館(1992)
- [19] Guo-qiang Bi & Mu-ming Poo., Distributed synaptic modification in neural networks induced by patterned stimulation. *Nature* 401 792-796 (1999) .
- [20] M. V. Tsodyks, et. al. ,The Neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proc. Natl. Acad. Sci. USA*, 719-723 (1994) .
- [21] F. Crick, *The astonishing hypothesis*, Touchstone, DNAに魂はあるか 驚異の仮説, 中原英臣他訳, 講談社, 323-326 (1995).
- [22] 伊藤眞次, *脳のホルモンと記憶*, 朝倉書店(1991).

- [23] A. Damasio, *Descartes' Error; Emotion, Reason, and the human brain*; 生存する脳, 田中三彦訳, 講談社(1994).
- [24] Aertsen, A. M., Gerstein, G. L, Habib, M. K. & Plam, G. J. *Neurophysiol.* 61, 900-917(1989).
- [25] 酒田英夫 他, 岩波講座・現代医学の基礎; 脳・神経の科学 脳の高次機能, 岩波書店 (1999).
- [26] W. Gerstner, A neuronal learning rule for sub-millisecond temporal coding, *Nature*, 383, 76-78 (1996).
- [27] D.J. Amit et. al. Dynamics of a recurrent network of spiking neurons before and following learning, *Network: computation neural system*, 8, 373-404 (1997)
- [28] 荒木修 他, 再帰的スパイクニューラルネットワークモデルにおけるデュアル情報コーディング, 電子情報通信学会 信学技報, 5, 49-56(2000).
- [29] O. Araki et. al., Dual information representation with stable firing rates and chaotic spatiotemporal spikes in a neural network model, (2000 submitted).
- [30] 伊藤浩之, 多細胞同時記録データの統計解析法, 日本神経回路学会誌, 7, 8-19(2000).
- [31] S. Ohta, O. Araki, A. Sakurai, The effects of spike-timing-dependent synaptic plasticity on neural network dynamics, (2001 submitted).

付録

A.	43
A.1 シミュレータについて	43
A.1.1 動作環境	43
A.1.2 開発環境	43
A.2 操作説明について	43
A.2.1 シミュレータの主な機能	43
A.2.2 ザ・バグ	44
A.2.3 操作方法	44
A.2.3.1 物理構造の用意	44
A.2.3.2 パラメータの設定と確認	47
A.2.3.3 シミュレーションの仕方	49
A.2.3.4 データの読み込み	51
A.2.3.5 データの保存	51
A.2.3.6 スパイク時系列を見る	51
A.2.3.7 シナプスウエイト時系列を見る	52
A.2.3.8 内部電位時系列を見る	53
A.2.3.9 表示移動距離を設定する	53
A.2.3.10 その他	54
	55

付録 A

A.1 シミュレータについて

このシミュレータは、実験用に作成しました。それから、スパイクニューラルネットをもっと身近に知ってもらってもらうよう、気楽に動かせるようなものがいいかなと思いましたので、あえてGUIを付加して作成しました。と、いうのはただの言いわけで、GUIは全くの個人的趣味です。ちなみにこのプログラムとソースはhttp://www.jaist.ac.jp/~ootasato/main_theme/PwNwS/におきました。

A.1.1 動作環境

OS	CPU	Memory	HDD
Windows 98	Pentium 400MHz	320Mbyte	36Gbyte

表A.1 シミュレーションの動作環境
データをたくさん保持するのでメモリ空間は広いほうがよいと思われます

A.1.2 開発環境

Microsoft Visual Studio、 Microsoft Visual C++

VC++を素早く学ぶため、[A1],[A2],[A3],[A4]のような本を参考にしました。ことに[A1]はWindowsの基本的な概念を学ぶ上で、非常にありがたい本でした。

A.2 操作説明

まずはシミュレータの主な機能について述べ、その機能を利用するための操作方法をそれに続けて述べます。アプリケーション名はPwNwSで、Pulse Neural Networks Simulation System(仮名)の擬似省略形から来ています。

また、諸所で具体的な画像を提示しますが、それらで使用するプログラムのバージョンが以前のものですので、表示が現在と異なっている所もあります。ご了承ください。

A.2.1 シミュレータの主な機能

- 簡単なスパイクニューロンを組み合わせ、その挙動をシミュレーション
- 構造をラフに出力(画面に)
- シミュレーション結果として、スパイクの時系列、ウエイトの時系列、内部電位の時系列を出力(画面に)
- データ(構造データ、スパイク時系列データ、ウエイト時系列データ)を読み書き

A.2.2 ザ・バグ

現在明確になっているバグを述べます。

- ・ 保存したファイルの構造と読み込んだ後の構造が多分違うというバグ

このバグは、作成したネットワークを保存し、その保存したファイルを読み込むと、その物理構造が作成したネットワークと異なったものになるバグです。

このバグの原因は明らかになっていますが、これを正すための時間的余裕が無かったためそのままになっています。ごめんなさい。

物理構造データの差異が生じないようにするには、物理構造を構築後、一旦保存し、その保存したファイルを読み込み、それを使ってシミュレーションするようにすれば、たえず同じ物理構造と保証されます。

以上の方法で回避できない場合は、あらかじめ、構築した時に利用したパラメータをメモしておき、毎回パラメータを入れなおして構造を作成し、シミュレーションしていただくことをおすすめします。

- ・ 範囲外指定をすると落ちる場合がある

多くの箇所(ダイアログボックス等)で範囲指定を伴う作業がありますが、大概の場合、条件から逸脱した値を入力すると、セグメンテーションフォルトなどで落ちてしまいます。これはユーザー側が慎重に値を入力すると前提しているわけではなくて、単に個人的に使用するなら指定範囲調査を踏まえなくてもいいという悪しき思想に甘えてしまったからです。ごめんなさい。

だから、プログラムを使用される際はぜひ指定するパラメータの値に注意を払っていただきたいと存じます。

- ・ JPSTHは未実装

JPSTHは未実装のため、使用しないでください。一度だけなら動作しますが、これも範囲指定によってはプログラムが落ちかもしれないので、なるべく使用を避けてください。

A.2.3 操作方法

これからは簡単な操作方法について説明します。プログラムにヘルプを実装していないので(一生実装しないと思われます)この場を借りて解説いたします。

A.2.3.1 物理構造の用意

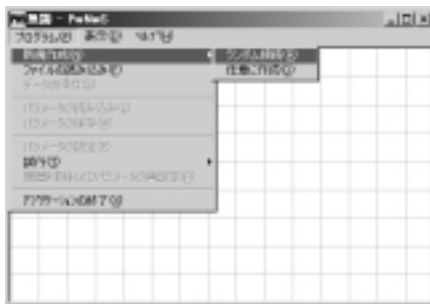
物理構造を準備するには3通りの方法があります。それは、ランダムに物理構造を設定する方法、ファイルから読み込む方法、任意に準備する方法です。以降では、これらの準備方法について説明いたします。

ランダムに物理構造を作成する

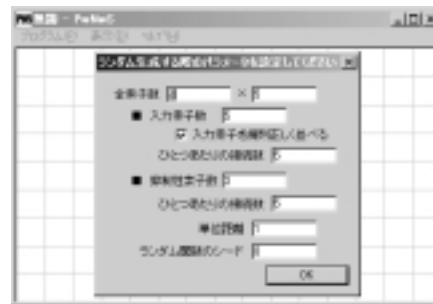
まず図A.1のように「新規作成」の中の「ランダムに接続」の項目を選択します。すると図A.2のよう

に各種パラメータの設定ダイアログが現れるので、ここで物理構造の規模に関するそれぞれのパラメータを設定します。設定の規模に関しては本文中の2.2節にあったネットワーク規模を参考にさせていただければ幸いです。

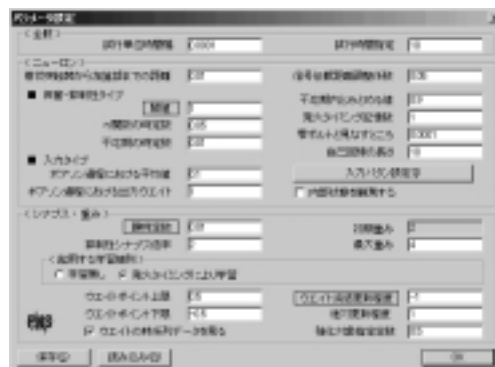
そうしてネットワーク規模の設定が終了すれば、こんどは図A.3のようなパラメータの設定画面が現れるので、ここでシミュレーションを行う際のパラメータを設定（確認）します。設定方法のちょっと詳しい解説に関しては、「パラメータの設定と確認」を参照していただければと思います。



図A.1 ネットワークの新規作成



図A.2 ランダムネットワーク作成時のパラメータ設定



図A.3 ニューラルネットワークのパラメータ設定ダイアログ

「OK」ボタンを押してパラメータの設定が終了したら作成された構造が表示されるはずですが。

ファイルから読み込む

ファイルから読み込む場合、以前に保存した物理構造データを利用する場合と、の「テキストで作成」したデータを読み込む場合のいずれが該当しますが、そのどちらでも次のことが言えます。

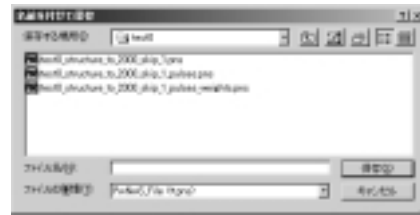
ファイルの中に物理構造のみが保存されているのであれば、この読み込み作業を行うだけでシミュレーション準備が完了します。逆に物理構造以外の余計なデータがファイルに内在していれば、読み込みに失敗するか、読み込んだ後で多少の操作をし、物理構造のみを準備したように見せなければなりません。これは仕様なのです。申し訳ありません。

もし、物理構造のみのデータを読み込むのであれば、次から述べる手続きによりシミュレーションできます。まず図A.4のようにメニューを選択します。すると図A.5のようなダイアログボックスが現れ

るので、ここで物理構造のファイルを選択します。選択後は、例によってシミュレーションの際のパラメータ設定(確認)ダイアログ(図A.3)が現れるので、パラメータ設定を必要とする場合はここで設定し、設定の必要ない場合は何もせず、「OK」を選択してシミュレーションの準備を終えます。



図A.4 ファイルから物理構造の読み込み



図A.5 読み込みファイルの指定

もし、物理構造のみでなく物理構造とシミュレーションの時系列データが保存されているデータを選んだ場合、以上の操作(この場合、パラメータ設定は無い)に加え、次に示す理由と方法で、シミュレーション準備を非正規に行わなければなりません。

今回のバージョンだと、読み込みの際にどのデータ(物理構造やシミュレーションの時系列データがある)をロードするかを選択できないのです。ファイルに書かれている総てのデータを自動的に読み込んでしまうのです。そのため時系列データを含むファイルを読み込んだ場合、アプリケーションは単に閲覧モードになり、すなわち、シミュレーションが終了した状態と同じになり、直ぐにはシミュレーションできなくなってしまいます。だから、時系列データが含まれていても、指定したファイルの物理構造を利用してシミュレーションしたい場合は、読み込み後に「履歴の削除およびパラメータの再設定」を選択し、「パラメータの設定と確認」で示すような操作で、パラメータを設定してください。そうすることで、データファイルがすべてクリアされ、シミュレーションの準備を終えた状態と同じになります。

任意に物理構造を作成する

任意に作成する場合は、2通りのやり方があります。ひとつはテキストを直接打ち込んで作成する方法。もうひとつはグラフィカルに作成する方法です。しかしながら、グラフィカルに作成する方法は今回の発表までに準備できなかったため、結局、任意に作成する場合はテキストで書くしかありません。ごめんなさい。ちなみにその書き方は、図A.6に述べてある通りです。

物理構造をテキストで書く場合、はじめは単にテキストファイルとして書き(図A.6のXMLチックな文法を守って)、保存する際に拡張子「pns」をつければ、上記の「ファイルから読み込む」の手続きによって任意に作成したネットワークをPwNwS上に用意することができます。

ちなみに、もし間違った書式で構造が表現されていたならば、「ファイルから読み込む」操作の際に、間違っている箇所(行番号)とその凡その理由が表示されるようになっています。たとえば図A.7のようにです。

```

##This_File_is_PwNwS_Program_File##
#   はじめの1行は以上の文字列を必ず書く
#   クロスハッチはコメントアウト
#   ネットワーク構造は以下のタグ及び構造を持たせれば、任意にネットワークを作成できる。
#   もし、タグスペルや構造に誤りがあれば、読み込みの際に間違いが指摘される。
#   以下のタグは大文字のみ使用しているが、実際は小文字大文字の区別はない。

<FILE> # ファイル構造
<BOXCREATE> # 細胞の作成
<CREATE><X>0</X><Y>0</Y><Z>0</Z><TYPE>1</TYPE><MODE>4</MODE><OBSERVE>1</OBSERVE></CREATE>
<CREATE><X>0</X><Y>1</Y><Z>0</Z><TYPE>1</TYPE><MODE>4</MODE><OBSERVE>1</OBSERVE></CREATE>
#   ..... 省略
#   CREATE   - - - -   ひとつのニューロンを作成する。
#   X,Y,Z    - - - -   位置を指定する。整数で指定。
#   TYPE     - - - -   入力ニューロンの場合「1」その他は「0」
#   MODE     - - - -   入力パタンの設定、通常は「4」を指定。
#   OBSERVE  - - - -   通常は「1」
</BOXCREATE> # 細胞の作成終了
<BOXPROJECT> # 投射の作成
<PROJECT><FROM>0</FROM><TO>1</TO><TYPE>0</TYPE></PROJECT>
<PROJECT><FROM>1</FROM><TO>2</TO><TYPE>0</TYPE></PROJECT>
#   ..... 省略
#   PROJECT  - - - -   一本の投射を作成する。
#   FROM     - - - -   投射元の細胞番号。細胞番号は<CREATE>した順番に同じ。
#   TO       - - - -   投射先の番号。
#   TYPE     - - - -   投射元が抑制性ニューロンの場合「1」。その他は「0」。
</BOXPROJECT> # 投射の作成終了
</FILE> # ファイル作成終了

```

図A.6 構造ファイルの作成方法と書式



図A.7 読み込みエラー表示

A.2.3.2 パラメータの設定と確認

パラメータを設定(確認)するケースは3箇所あります。まずは、物理構造を用意する際に設定(以降は確認を省いて単に設定と表現します)、そして、シミュレーションが終了してからの再設定、最後に、シミュレーション前の再設定です。

何れの場合も、パラメータ設定ダイアログ(図A.3)を表示してからパラメータ設定を行います。尚、に関しては意図的な操作を伴わないで、自動的に設定ダイアログボックスが表示されますが、他のふたつ、に関しては、メニュー「プログラム(P)」の「パラメータの設定(P)」か「履歴の削除およびパラメータの再設定(R)」を選択して、パラメータ設定ダイアログを表示し、設定します。パラメータを設定すると、物理構造のデータを除き、総ての時系列データが削除されますから、もし、重要な結果が得られた場合は、パラメータ設定を行う前に、それらの結果を保存しておかなければなりません。ちなみに、時系列データの保存に関しては「データの保存」の項目に述べてあります。

一方、パラメータを変更しない場合や時系列データを消したくない場合、すなわち確認のみでいい

場合は、どの設定以降目にも触れず、右上の「x」（閉じる）を押して設定ダイアログを閉じていただきます。

設定画面が現れたら、諸所のパラメータを設定します。設定する際は、表A.1のパラメータ設定ダイアログの簡単な説明を参考にされるのがいいと思われます。

パラメータ設定ダイアログの項目	説明
試行単位時間幅	ネットワークの処理単位時間を決めるパラメータ。0.00005秒に設定しているのをすすめる。
試行時間指定	ネットワークをどれだけの時間実行させるかを決めるパラメータ。規模にもよっては非常にシミュレーション時間がかかるので、まずは短い時間を設定して試し、その後で正式に設定する。
樹上突起棘から加算部までの距離	デフォルト値0.01くらいがいい。この値を大きくすると、入力されてから内部電位が変化し始めるまでの時間が長くなる。
信号伝播距離調整係数	これもデフォルトのままでもいい。なお、この値を大きくすると発火してから次の細胞に到達するまでの時間が長くなる。
閾値	低いと発火しっ放しになり、同時刻検出性を喪失してしまう。高すぎると、たくさん物が同時刻に入力されなければ発火しなくなる。大体1.2から1.5がよい。
関数の時定数	今回のモデルでは無意味な定数。
不応期の時定数	不応期の長さを決定する時定数。0.01くらいがいいが、大きくすれば、ネットワーク全体の発火率を下げる事に値する。
不応期内と認める値	今回のモデルでは無意味な定数。
発火タイミングの記憶数	今回のモデルでは無意味な定数。でもデフォルトのままにしておく。
零ボルトと見なすところ	入力により変化した内部電圧が何処まで下がれば0ボルトと見なしていいかを定める定数。大きい値であればその大きい分だけ過去の入力情報が早く取り除かれるので、計算回数が減る。
自己回帰の長さ	自己回帰が存在するときの長さを決める定数。
Poisson過程における平均値	ランダム生成する入力の平均ISIを決定するあたり。0.1くらいがいいと思われる。大きくすれば、ISIが大きくなり発火がまばらになる。
Poisson過程における出力ウエイト	Poisson過程を変形するパラメータ。1が通常のPoisson過程。
入力パタンの設定	A.2.3.10 その他の「入力パタンの設定」を参照
内部状態を観測する膜時定数	チェックを入れると内部電位時系列を閲覧できるようになる。通常0.01くらい。色々試して、発火パターンを見るのもいいかと。
抑制性シナプス倍率	抑制性は興奮性のものに比べて影響が強いので、それなりの値を設定する。デフォルトでは、興奮性よりも2倍ウエイトが高くなっている。
最大重み	興奮性ニューロンの仮想最大重み値を設定する。
起用する学習規則	学習の有り無しを設定する。
ウエイトポイント上、下限	ウエイトの移動上限と、移動下限を決める
ウエイト減退更新程度	ウエイトを更新するときどれほどの割合で行うかを定める定数。通常は1。
絶対更新程度	ウエイトを更新するときどれほどの割合で行うかを定める定数。
強化対象指定定数	通常は1。
ウエイトの時系列データを見る	チェックを入れるとウエイトの時系列を閲覧できるようにする。

表A.1 パラメータ設定ダイアログの簡単な説明

設定が終了すれば「OK」を押して設定ダイアログボックスを閉じます。もし観測データを大量に予約するのであれば、このとき配列準備に時間を要することがありますので、その辺りご考慮ください。

ところで、これらのパラメータは読み書きできるようになっています。

もし、以前に保存したパラメータファイルを読み込むのであれば、パラメータ設定ダイアログボックスの左下にある「読み込み」を押してファイル(拡張子が「ppf」)を読み込みます。もし、現在表示されているパラメータを保存したい場合は、同じく、ダイアログボックスの左下にある「保存」ボタンを押し、ファイル名を指定し、保存します。

A.2.3.3 シミュレーションの仕方

シミュレーションを行うためには、物理構造が用意されていて、パラメータが設定されていなければなりません。それらを済ませてない場合は、先にそれらの準備します。準備が終わっている場合は、以下の手続きを踏まえてシミュレーションを行います。

シミュレーションには3つの方針があります。それらは、試しに単位時間ずつ動かしてスパイクの様子をうかがいつつシミュレーションする方式、そして、指定した時間分を一気にシミュレーションする方式、最後に、ある特定のパラメータのみを任意の幅で移動させつつ、それらのパラメータに対して連続的にシミュレーションする方式です。(2000年1月現在で にバグがあることが分かっているので、使用を控えてください。)

以降では、これら3つの方式について説明していきます。

スパイクを見ながら、徐々にシミュレーションする

これは、凡そのスパイクの様子をうかがって、このままシミュレーションすべきかどうかを決定する際などに使用するシミュレーション方式です。実行するには、図A.8にあるメニューを選びます。すると図A.9のような表示が現れるので、「次の単位(N)」ボタンを押せば単位時間(現在のバージョンではこの数値を任意に指定できません)だけ実行して一旦停止します。もしシミュレーションを、最後まで続行してもらいたい場合は「残り全部(R)」を押し、このシミュレーションを中断(終了)したい場合は「中断(I)」を押します。



図A.8 徐々にシミュレーションする



図A.9 徐々にシミュレーションダイアログ

パラメータ設定ダイアログボックスで指定した時間分だけシミュレーションする

この場合は、図A.10のようにメニューを選択する。するとパラメータ指定ダイアログボックスで指定した時間分だけシミュレーションします。



図A.10 指定した時間分だけシミュレーションするとき

ある特定のパラメータを移動させつつシミュレーションする
(2000年1月現在、バグが残留しているので注意が必要 「A.2 ザ・バグ」節を参照してください)

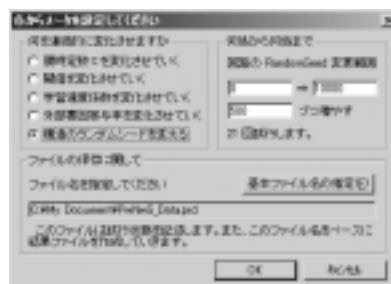
この方式は、ある特定のパラメータに関して連続的に変化させながら行います。そしてそれらの結果は、ログファイルと共に、ログファイルと同じフォルダ内にデータファイルとして保存されます。

この方式でシミュレーションをするには図A.11にあるようにメニューを選びます。すると、図A.12のような設定画面が現れるので、どのパラメータをどの値からどの値まで変化させ、どれだけ区切りでインクリメントさせながら連続的にシミュレーションするかを指定し、どのフォルダのどんなファイル名で保存するかを決定します。決めて「OK」を押すと、指定された連続シミュレーションを行い、それらのデータを指定したフォルダへ保存していきます。

保存されたファイルを開覧する為には「ファイルの読み込み」を参照していただきたく思います。



図A.11 あるパラメータ変えながら試行



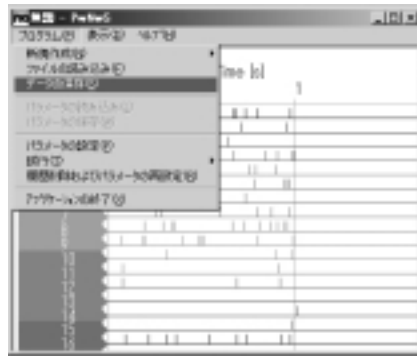
図A.12 変更させるパラメータの設定と保存先の指定

A.2.3.4 データの読み込み

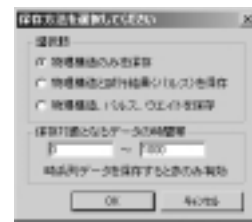
データを読み込むには、あらかじめ、構造データのみなのか、スパイクやウエイトの時系列データも含むものなのか、その何れであるかを知っていなければなりません。何はともあれ、これらが分かっているものとして話を進めさせていただきます。

もし物理構造のみのデータをロードするなら、「物理構造の準備」で記してあるように行います。時系列データを閲覧するためにロードするなら、時系列データを含むファイルを読み込むだけでよいようになっていますので特に作業は必要ありません。ただし時系列データを含むデータを読み込んだ場合、基本的にその構造を使ってシミュレーションできないので、その点をご注意願います。

A.2.3.5 データの保存



図A.13 保存する



図A.14 データの保存方法

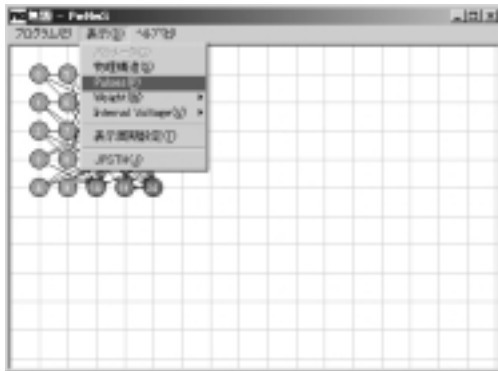
データを保存する時は、図A.13に示すようにメニューを選択します。すると図A.14のようなダイアログボックスが現れるので、ここでどのデータセットを保存するかを選択します。もし、時系列データも含めるようデータセットを選択するのであれば、その時系列の保存開始時間と終了時間も設定します。時間指定のデフォルト値は、シミュレーション開始から終了の全時系列分です。

A.2.3.6 スパイク時系列を見る

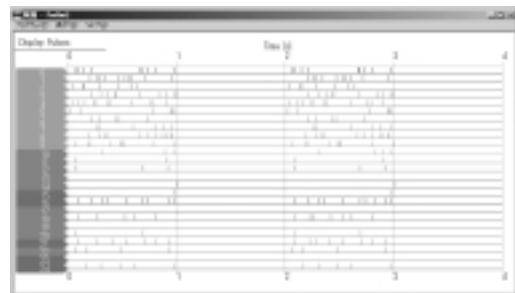
シミュレーションが終了すると、スパイク時系列が表示されるようになっていますが、もし、別の表示モードに切り替えた後で、再びスパイクの時系列を閲覧するときは、図A.15にあるメニューを選択すればスパイク時系列を見ることができます。

なお、これらの時系列(図A.16)は「**↑**」や「**↓**」を使って表示位置を移動することができます。シフトを押しながら矢印を押すと、移動距離単位を増やして移動します。そして、コントロールキーを押しながら矢印を押すと、「表示移動距離を設定する」の設定値に従って移動します。

時系列のスケールを変えるには、コントロールキーとシフトキーを押しながら、「**+**」、もしくは「**-**」のボタンを押します。



図A.15 スパイクの時系列を表示させる



図A.16 スパイク時系列のラスタ表示

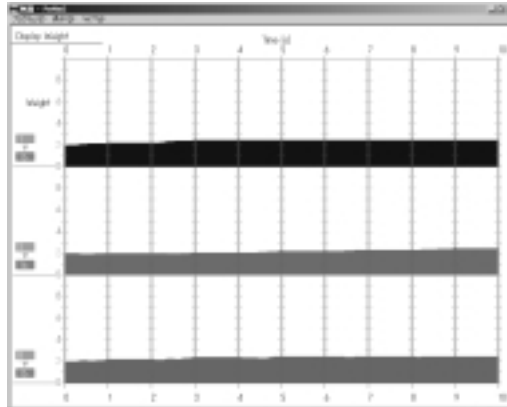
A.2.3.7 シナプスウェイト時系列を見る



図A.17 ウェイトの時系列を表示させる 図A.18 ひとつの細胞へのウェイト時系列を観る

シミュレーションが終了するとシナプスウェイトの時系列（概略）を観ることができます。この時系列を閲覧するためには図A.17にあるメニューを選択します。すると図A.19に示すような表示モードに切り替わります。なお、時系列の移動、スケールの変更に関しては「スパイクの時系列を見る」で示したものと同じです。

また、一つ細胞に対するウェイトを見るには図A.18で示すメニューを選択します。

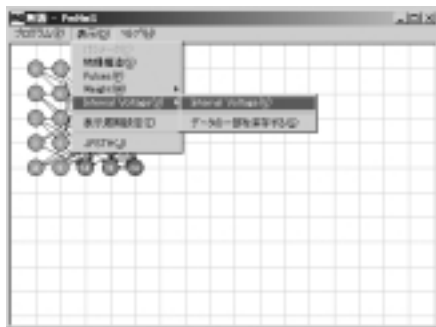


図A.19 ウエイト時系列

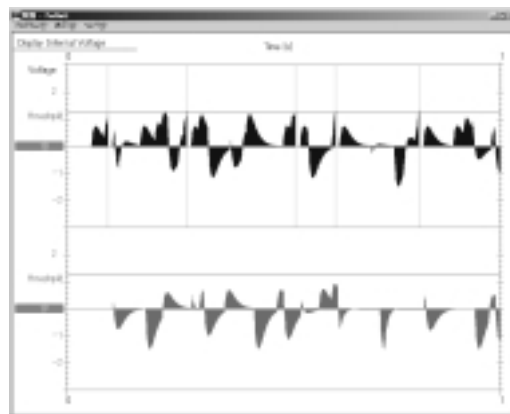
A.2.3.8 内部電位時系列を見る

この時系列を閲覧するためには図A.20にあるメニューを選択します。すると図A.21に示すような内部電位表示モードに切り替わります。時系列の移動、スケールの変更に関しては「スパイクの時系列を見る」で示したものと同様の操作で行います。

また、内部電位の時系列分析する手助けとなるよう、時系列のデータを様々な形で、出力できるようにしました。こうして出力したデータは、GNU PLOTで使用できるようなデータ構造になっています。



図A.20 内部電位時系列を閲覧する



図A.21 内部電位時系列

A.2.3.9 表示移動距離を設定する

スパイク時系列、ウエイト時系列、内部電圧時系列の何れが表示されている場合でも、コントロールキーを押しながら「 \leftarrow 」や「 \rightarrow 」を押すと任意の距離ずつ移動させられます。その移動距離を設定するには、図A.23にあるメニューを選択すると、図A.24のようなダイアログが表示されるので、ここでコントロールキーを押しながら矢印を押した場合、どれだけ移動させるかを決められます。



図A.23 表示移動距離を指定する



図A.24 移動距離指定ダイアログ

A.2.3.10 その他

- ・ 終了

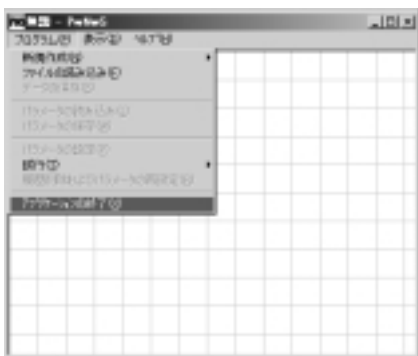
図A.25のメニューを選択するか、PwNwSアプリケーションウィンドウの右上の「x」をクリックするか、ウィンドウ左上のアイコンをダブルクリックします。

- ・ 入力パタンの設定

パラメータ設定ダイアログの「入力パタンの設定」でPoisson過程のスパイク時系列を作るときに使用するシードと、そのシードを適応する時間幅を指定できます（図A.26）。

実行順...の列の番号を指定し、削除を押すとそのシードを削除できます。

追加を押すと、どのシードをどれだけの時間で、何回実行するかを質問されます。このとき、実行回数として「-1」を指定すると、そのシードは指定された時間分の時空間パターンを永遠に繰り返すように設定されます。



図A.25 終了するときのメニュー



図A.26 シードのシーケンスを作成するダイアログ

参考文献

- [A1] 柏原正三, C++ユーザーのためのWindowsプログラミングの基礎知識, アスキー出版 (1999).
- [A2] 林晴比古, 新Visual C++6.0入門 ビギナー・シニア編, SOFT BANK (1998).
- [A3] 柴田望洋, プログラミング講義C++, SOFT BANK (1996).
- [A4] Microsoft Corporation著 マイクロソフト(株)訳, Microsoft Visual C++6.0 MFCライブラリリファレンス, Microsoft PRESS (1998).