

図 2-1 伝達経路の信号波形

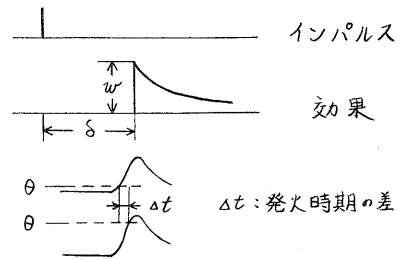


図 2-2 受信細胞に与える効果

そこで、素子の膜電位 $V_i(t)$ と出力インパルス列 $Z_i(t)$ を、各々、内部状態変数、出力変数と考え、これらの状態遷移法則を定義する。

ここで、以下のパラメータを導入する。

素子 i についてのパラメータ

- τ_{Vi} : 膜電位の減衰定数
- R_i : 絶対不応期の時間
- $\theta_i(t)$: 閾値
- $\tau_{\theta i}$: θ の減衰定数
- θ_{0i} : 定常状態の閾値
- θ_{1i} : 発火直後の閾値

素子 j から素子 i への結合 ij についてのパラメータ

- δ_{ij} : 伝達遅れ時間
- $w_{ij}(t)$: 結合荷重
- $\tau_{w_{ij}}$: w の減衰定数
- w_{0ij} : 定常状態の結合荷重
- w_{fij} : 発火直後の結合荷重の増分

なお、 $\theta_0, \theta_1, \tau_{\theta}$ は相対不応期に、 w_0, w_f, τ_w はシナプス強化に対応して導入したパラメータである。

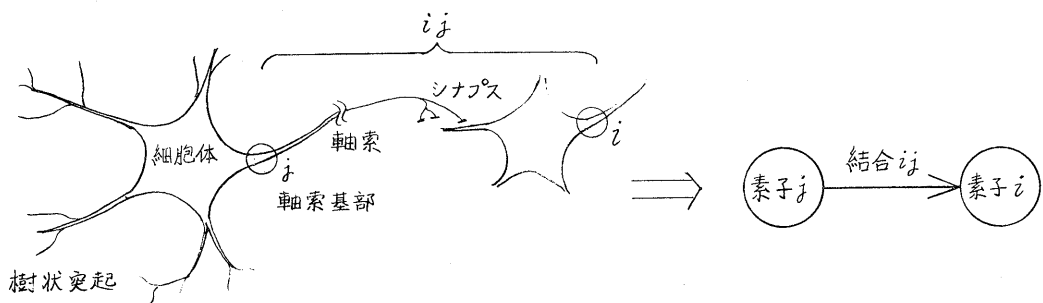


図 2-3 神経回路のモデル化

状態遷移法則

$$\left\{ \begin{array}{l}
 \circ \text{ if 素子が絶対不応期でない and } V_i(t) \leq \theta_i(t) \\
 \text{then} \\
 \frac{dV_i(t)}{dt} = -\gamma_{vi} V_i(t) + \sum_{j=1}^n w_{ij}(t) Z_j(t - \delta_{ij}) \\
 \text{else} \\
 V_i(t) = 0 \\
 \\
 \circ Z_i(t) = \delta(V_i(t) - \theta_i(t)) \quad i = 1, 2, \dots, n
 \end{array} \right.$$

$Z_i(t)$ はデルタ関数の形であり、 $V_i(t)$ を非線形項の第2項がデルタ関数の形であるので、 $V_i(t)$ 、 $Z_i(t)$ 、($i = 1, \dots, n$) についての離散型シミュレーションが可能となる。

シミュレーションに際し、系の初期値と境界が問題となる。 $Z_i(t) = 0$ 、($t \leq 0$) で、 $V_i(0) = 0$ の状態は、このモデルの1つの定常状態ではあるが、“死んだ脳”に対応した無意味な状態である。といって、初期値を乱数で与えるのは、脳の機能面をあまりに無視し過ぎている。また、脳は 10^{10} 個程度の神経素子から成るが、これを一度にシミュレーションするのは現在のところ非現実的である。といって、むやみに脳の一部を切り出してみても、正常な動作は期待できない。

この2つの問題を解決するのに、独立にインパルスを出力する自励発信器を、外部素子として導入する。この外部素子に対し、今までの素子を内部素子と呼ぶ。外部素子の特性は、次のパラメータで決定する。

外部素子 j についてのパラメータ

- T_j : 発信周期
- t_{sj} : 発信開始時刻
- t_{ej} : 発信終了時刻

外部素子の解釈は4通りある。

- 1) シミュレーション対象の系内の素子に結合した系外の素子。
- 2) 膜電位の初期値を与える。
- 3) インパルス列の初期値を与える。(図2-4)
- 4) 脳に挿入した刺激電極。

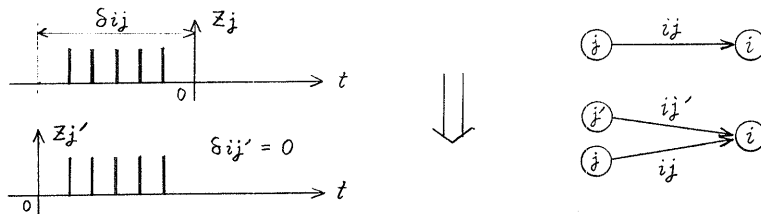


図2-4 インパルス列の初期値の与え方

3. アルゴリズム

シミュレーションプログラムは、3つの独立したプログラムから成る。まず、NETGENが、与えられたパラメータから回路網を構成する。次に、NETSIMが、回路網の動作をシミュレーションし、指定された素子の現象を検出する。最後に、NETREPが、検出された現象に指定された出力処理を施して出力する。これらはいずれも、二次記憶上のファイルを使用して、プログラム間でデータをやり取りする。

3-1. NETGEN

NETGENのフローを図3-1に示す。

網構造入力NETは、システムで用意された網構造記述用サブルーチンを使用して、利用者がFORTRANで作成する。網構造記述用サブルーチンは、内部素子、外部素子、結合を定義するもので、各々に対応して、DC, DX, DSが用意されている。これらのサブルーチンは、引数で与えられたパラメータをチェックして、内部データを作成する。

結合部ソートSORTは、シミュレーションの速度を向上させるため、結合部を S_{ij} についてソートし、相対伝達遅れ時間 S'_{ij} に変換する。

以上によって作成された内部データは、図3-2に示す様なリスト構造を持つ。内部素子と外部素子は、同一の配列CELLで表現され、結合は配列SYNで表現される。これらは、初期対象符行列と伴に、ファイルに出力される。

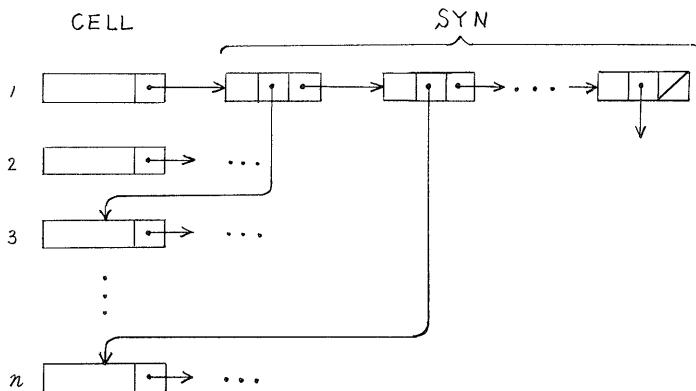


図3-2 回路網の内部データ構造

3-2. NETSIM

NETSIMのフローを図3-3に示す。

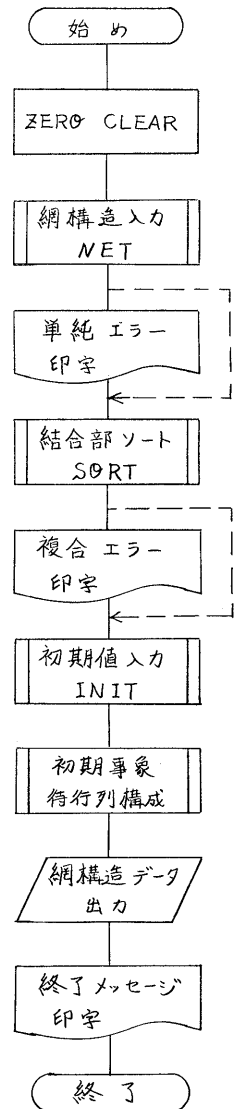


図3-1

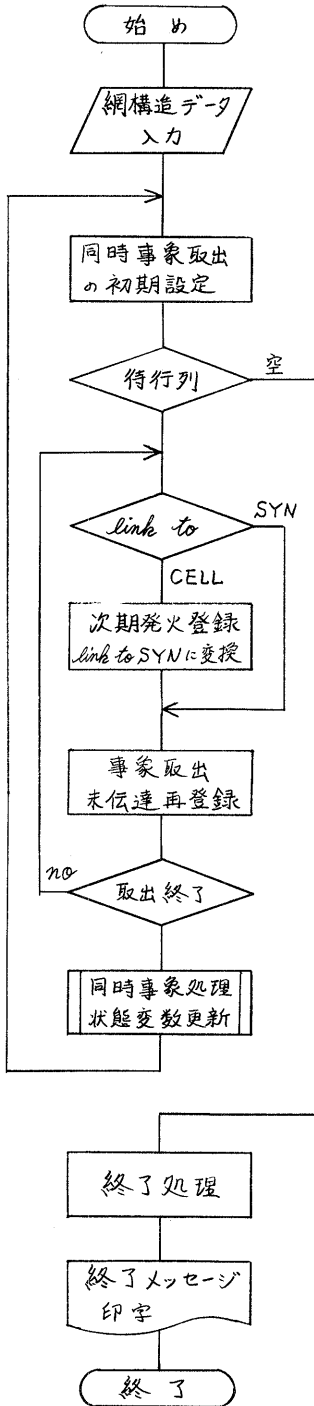


図 3-3

NETSIMは、NETGENで作成された網構造データと初期事象待行列をファイルから入力し、シミュレーション時間と素子の出力指定変更を入力した後、網のシミュレーションを開始する。

シミュレーションされる事象には、インパルスの到着による膜電位の更新と、外部素子のインパルス発射の2種類があり、未処理の事象は1本の優先順位付待行列QUEUEで表現される。QUEUEは、事象表現そのものではなく、リスト構造を持つ外部素子と結合へのポイントの集合である(図3-4)。初期事象待行列は、外部素子のもののみから成る。

事象の処理は、大規模性を考慮して、同時事象をすべて取出してから行う。取出した事象は、一度、配列EVENTに入れて、これを受信素子についてソートして、同一受信素子のものはまとめて処理する。

事象の発生は、事象の取出しに伴うものと、事象の処理によるものがある。事象取出しの後、取出されたQUEUEのポイントが、外部素子と結合のリスト上を1つ進み、新事象が発生したことになり、再びQUEUEに登録される。もし、リストの残りが空ならば、ポイントは削除されたままである。また、外部素子の場合は、次期のインパルス発射も新事象として登録される。事象処理において、更新された内部素子の膜電位がその閾値を越えてインパルスが発射された場合、その内部素子から始まる結合リストが新事象となり、リストの先頭を指すポイントが生成されてQUEUEに登録される。また、このとき、その内部素子は絶対不応期に入り、素子内部の時計を絶対不応期終了まで進める。(CELLとSYNの各要素は、それぞれ、内部に独自の時計を持っている。)さらに、出力指定された素子の場合は、出力ファイルにその現象を出力する。この出力ファイルは、処理終了後、NETREPに渡される。

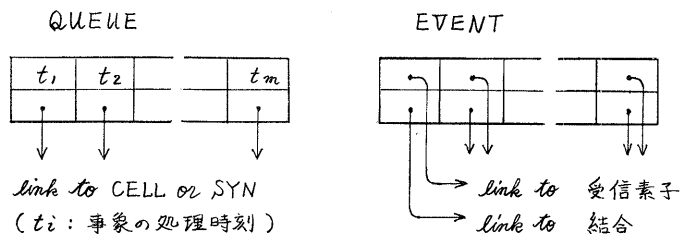


図 3-4 QUEUEとEVENT

4. 優先順位付待行列

シミュレーションにおいて、未処理の事象を優先順位付待行列で表現したが、この待行列のデータ構造と操作アルゴリズムについて述べる。

待行列に対する操作は、次の2つである。

- 1) DELETE : 最小要素を取出す。
- 2) APPEND (t, ℓ) : 要素 (t, ℓ) を新たに登録する。

他に最小要素の参照があるが、これは待行列の内容の変更を伴わない。

さらに、操作列を考慮すると、シミュレーションの終了近くを除いて、DELETE が2回以上連続する部分はDELETE, APPENDと連続する部分よりまれである。また、APPENDが連続する部分もDELETE, APPENDの連続より少量である。そこで、整列2分木 (*heap*) にこの操作列の特徴を生かす改良を加えて、これをデータ構造として採用する。

heap では、最小の要素は常に *root* にある。1回目のDELETEでは、この *root* を取出せばよい。しかし、この操作の後には、DELETEよりAPPENDを行なう確率の方が高いので、*heap* の再構成はせずに *root* を空のままにしておき、続くAPPENDで *root* に新要素を登録して *heap* を再構成する。続く操作がDELETEである場合は、*root* の2つの子のうち小さい方を取出した後で *heap* を再構成し、やはり *root* は空のまま残しておく。*root* の子の比較は *root* が空になった直後に行なって、その結果はポイント HQ に入れておく。*root* が空でないときには $HQ=1$ としておいて、HQ を用いれば常に最小要素を参照できる様にする。

$HQ=1$ のときのAPPENDは、新要素を空の *leaf* に登録して *heap* の再構成を行なう。これは、*root* に登録する場合の半分の比較ですむ。

この改良は、時間計算量の定数を数パーセント改善する。

```

procedure DELETE ;
begin
  NQ ← NQ - 1 ;
  if HQ ≠ 1 then
    begin
      APPEND (key (NQ+2), link (NQ+2)) ;
      NQ ← NQ - 1
    end ;
  if NQ ≥ 2 and key (3) < key (2) then HQ ← 3
  else HQ ← 2
end

procedure APPEND (t, ℓ) ;
begin
  NQ ← NQ + 1 ;
  if HQ ≠ 1 then 要素 (t, ℓ) を root に登録
  else           " leaf " ;
  heap を再構成する ;
  HQ ← 1
end
  
```

図 4-1 待行列操作アルゴリズム

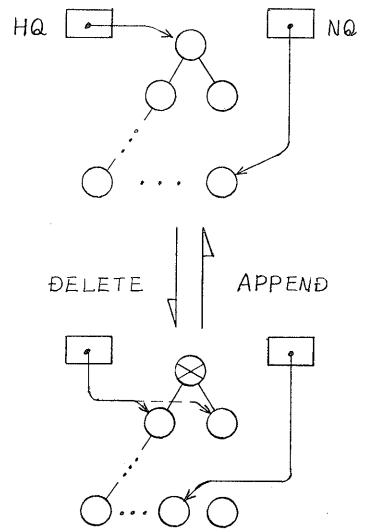


図 4-2 データ構造

5. むすび

神経回路の動作を表現するのに、インパルス列に注目し、これと膜電位の状態遷移法則を定義して離散型シミュレーションが可能であることを示した。そして、回路網を内部素子、外部素子、結合の集まりとしてモデル化した。

シミュレーションにおいては、データの大規模性を考慮し、素子-結合のリスト構造を利用して、事象符行列のしめる計算量のオーダーの改善をはかった。さらに、符行列操作列の特徴を生かしたデータ構造とアルゴリズムを採用して、計算量の定数改善もはかった。

シミュレーション可能な回路網の規模は、320KWのメモリ容量で最大、素子が 2×10^3 個、結合が 2×10^4 個までである。一般に、素子 n 、結合 m とすると、データ領域は $(13m + 19n) \times 4B$ となる。将来の計算機のメモリ容量の飛躍的な増加を期待すれば、実際の脳の一部のシミュレーションも可能となるであろう。

データの大規模性は、また、データ入出力の煩雑さをも招く。入力については、データをパラメータ化してプログラムで与える様にして、これを解決した。出力(NETKEP)については、同様の方針で検討中である。

なお、プログラムの開発は、東京大学大型計算機センターのHITAC 8800/8700を使用した。

参考文献

- 1) J. S. Griffith : 数理神経生物学 産業図書 1973
- 2) A. V. Aho, J. E. Hopcroft, J. D. Ullman : アルゴリズムの設計と解析
サイエンス社 1977
- 3) 佐藤, 倉田 : 神経回路網汎用シミュレーションプログラム
電子通信学会 MBE 77-11
- 4) Larry D. Wittie : Large-scale simulation of brain cortices
Simulation September 1978