

# 大規模 MOSFET リザーバーの高速シミュレーション

村田 寛也<sup>1,a)</sup> 久米 祐貴<sup>1</sup> 辺 松<sup>1</sup> 栗野 皓光<sup>1</sup> 佐藤 高史<sup>1</sup>

**概要:** エコーステートネットワーク (ESN) は再帰的ニューラルネットワーク (RNN) の一種であり、時系列データの処理に適している。入力層-中間層間および中間層-中間層間の重みは学習前にランダムな値に固定し、中間層-出力層間の重みのみを学習することが大きな特徴である。この特徴から、学習コストが低く、高速な学習が可能である。また、入力層、中間層はリザーバーとも呼ばれ、その単純さからリザーバーをハードウェアや物理現象で表現する様々な研究がある。MOSESN はリザーバーを MOSFET クロスバアレイ回路としたものであり、重みの生成にトランジスタの微細化に伴うしきい値電圧のばらつきを利用しており、集積回路化可能なリザーバーとして知られている。しかし、使用される素子数の多さから、MOSESN のシミュレーションには多大な時間を要する。本稿では、リザーバーの疎性を利用して、オフ状態の MOSFET に流れる電流を縮約し、MOSESN の回路シミュレーション高速化を試みた。提案する手法により、回路シミュレーション時間は最大で 94% 短縮された。

## Fast Simulation of Large-scale MOSFET Reservoir

HIROYA MURATA<sup>1,a)</sup> YUKI KUME<sup>1</sup> SONG BIAN<sup>1</sup> HIROMITSU AWANO<sup>1</sup> TAKASHI SATO<sup>1</sup>

**Abstract:** Echo State Network (ESN) is a kind of recurrent neural network (RNN). RNN is suited for processing time series data. Only the output, or the readout, weights are trained in ESN. Therefore, the major advantage of ESN is fast learning, resulting in low training cost. Due to its simplicity, hardware implementation of the input and middle layers in ESN (also known as the reservoir) has attracted attention. Hardware reservoirs are expected to achieve lower power consumption and faster operation than software-based reservoir. MOSESN is a hardware implementation of ESN and it consists of MOSFET double crossbar array circuits. The weights are generated using the variation of the threshold voltage due to the miniaturization of transistors in MOSESN. It is expected that the reservoir of MOSESN can be implemented as an integrated circuit. However, the circuit simulation of MOSESN requires a large amount of time because of the large number of elements. In this paper, we attempt to speed up the circuit simulation of MOSESN by using the sparsity of the reservoir to replace the currents of off-state MOSFETs. The proposed method reduces the circuit simulation time by up to 94%.

### 1. はじめに

エコーステートネットワーク (Echo State Network: ESN) [1] は回帰型ニューラルネットワーク (Recurrent Neural Network: RNN) の一種であり、時系列データの処理に適しているが、学習の速さを特徴の 1 つとしている。一般に、RNN は入力層、中間層、出力層の 3 層と入力層から中間層、中間層から中間層、中間層から出力層への 3 種類の重みを持つ。他の RNN モデルとして知られる

Long Short Term Memory (LSTM) [2] ではこれらの重みすべてを学習するが、ESN では、入力-中間層間および中間-中間層間の重みを学習前にランダムに固定し、中間-出力層間の重みのみを線形回帰などの単純なアルゴリズムで学習させる。この単純で高速な学習過程により、ESN はあまり計算リソースを必要としない [1,3]。また、上記のような構造から、中間層はリザーバー、出力層はリードアウトとも呼ばれ、同様の構造をもつ Liquid State Machine [4] と ESN をまとめてリザーバーコンピューティング (Reservoir Computing: RC) と呼ぶ [5]。

ESN の構造の単純さから、リザーバー部を物理現象で表現する、ハードウェアリザーバーを用いた ESN に関する研

<sup>1</sup> 京都大学 大学院情報学専攻 通信情報システム専攻  
〒 606-8501 京都府京都市左京区吉田本町

<sup>a)</sup> paper@easter.kuee.kyoto-u.ac.jp

究が注目を浴びている [3]. ハードウェアリザバーを使用するメリットとしては、低消費電力性、高速な動作が期待されることが挙げられる. MOSESN [6] はハードウェアリザバーを用いた ESN の 1 つである. MOSESN の優れた点として、MOSFET の微細化に伴うしきい値電圧の製造ばらつきを利用するため、他のハードウェアリザバー [7, 8] と異なり、入力-中間層間および中間-中間層間の重みを事前に生成し、書き込む必要がないことが挙げられる. さらに、MOSFET は製造コストが低く、回路面積も小さくすることができるという利点もある. MOSESN はまだ完全にはチップ化されておらず、時系列データに関するタスクが MOSESN で解決可能かどうかは、回路シミュレーションを通して確認する. しかし、MOSESN における MOSFET の素子数は数万個におよび、回路シミュレーションには少なくとも数日を要する. ESN の中間層は疎であることから、シミュレーションを簡略化する方法として、オフ状態で使用する MOSFET をネットリストに記述せずにシミュレーションを行うという方法が考えられる. しかし、MOSFET のリーク電流が無視されることから大規模な MOSESN では不正確な結果を与え、また高温環境での動作を確認したい場合等に対応できない. そこで本稿では、MOSESN における MOSFET のリーク電流の振る舞いを確認し、それらを縮約し電流源に置換することで、シミュレーションの高速化を試みた.

## 2. Echo State Network

本章では、ESN の基本構造と制約について述べる.

### 2.1 ESN の基本構造

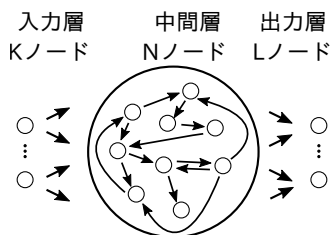


図 1: ESN の基本構造

図 1 は ESN の基本構造である. ESN は入力層、中間層、出力層の 3 層で構成される. 入力-中間層間および中間-中間層間の重み  $\mathbf{W}$ ,  $\mathbf{W}_{in}$  は、学習開始前にランダムに決定し、固定する. また、中間-出力層間の重み  $\mathbf{W}_{out}$  は学習によって決定する. 高次元空間の中間層はリザバーと呼ばれ、出力層はリードアウトと呼ばれる. 入力層、中間層、出力層のノード数をそれぞれ  $K$ ,  $N$ ,  $L$  と表し、時刻  $t$  における入力を  $\mathbf{u}(t) \in \mathbb{R}^K$ , 中間層のノード状態を  $\mathbf{x}(t) \in \mathbb{R}^N$ , 出力を  $\mathbf{y}(t) \in \mathbb{R}^L$  と表し、各層間の重みをそれぞれ  $\mathbf{W}_{in} \in \mathbb{R}^{N \times K}$ ,  $\mathbf{W} \in \mathbb{R}^{N \times N}$ ,  $\mathbf{W}_{out} \in \mathbb{R}^{L \times N}$  と表すと、 $\mathbf{x}(t)$ ,  $\mathbf{y}(t)$  は以下

の式によって更新される.

$$\mathbf{x}(t) = f[\mathbf{W}_{in} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1)] \quad (1)$$

$$\mathbf{y}(t) = g[\mathbf{W}_{out} \cdot \mathbf{x}(t)] \quad (2)$$

関数  $f(\cdot)$  には非線形の活性化関数、例えば、双曲線正接関数  $\tanh(\cdot)$  が、関数  $g(\cdot)$  には恒等関数が用いられることが多い.

### 2.2 ランダム重み $\mathbf{W}_{in}$ , $\mathbf{W}$ の制約

リザバーの状態は、近い過去の入力に強く依存し、遠い過去の入力への依存性を小さくする必要がある. そのため、 $\mathbf{W}_{in}$ ,  $\mathbf{W}$  は、以下の条件を満たさなければならない [1].

- 要素の平均が 0 であること.

また  $\mathbf{W}$  のみについて、

- 疎行列であること.
- スペクトル半径 (固有値の絶対値の最大値) が 1 程度または 1 未満であること.

$\mathbf{W}$  のスペクトル半径に関する条件を満たすとき、リザバー内部の過去の状態は漸近的に小さくなる.

## 3. MOSESN

本章では MOSESN の構造と動作を説明する. MOSESN [6] は、ハードウェアリザバーを用いた ESN の 1 つであり、ランダムな重みの生成に、MOSFET の閾値電圧ばらつきを利用している. メモリスタクロスバレイ回路に基づく ESN [7] から着想を得ている. なお、メモリスタとは、過去に流れた電流を記憶する抵抗器である. このメモリスタクロスバレイ回路による ESN を memESN (memristor ESN) と呼ぶこととする. memESN では、外部で重み  $\mathbf{W}_{in}$ ,  $\mathbf{W}$  をランダムに生成し、メモリスタに抵抗値として書き込む必要があったが、MOSFET クロスバレイ回路では、重みを生成、書き込み、調整する必要がない. さらに、MOSFET はメモリスタに比べ製造も容易であり、低コスト化、低消費電力化および回路の小型化が期待できる. これらの特性から、ハードウェアリザバーを用いた ESN の中でも特に、MOSFET リザバーを用いた ESN はセンサや組み込みシステムなどと相性が良いと言える. 本稿では、MOSESN における MOSFET クロスバレイ回路および周辺回路を MOSFET リザバーと呼ぶこととする.

### 3.1 クロスバレイ回路の基本動作

図 2 に MOSFET クロスバレイ回路の構造を示す. 横軸方向に電圧を入力とする配線が複数あり、縦軸方向に電流を出力とする配線が複数ある. 横軸を行、縦軸を列と呼ぶ. 入力と出力は格子状に重ねられ、その各交点は、n 型 MOSFET のドレインとソースで結ばれる. 基本的な動作を次に示す.

- (1) それぞれの行に入力電圧を印加する.
- (2) 交点に接続された MOSFET のドレイン・ソース間に電流が流れる.
- (3) それぞれの列に流れる電流を出力とする. 出力は列上の各 MOSFET のドレイン・ソース間を流れる電流の和である.

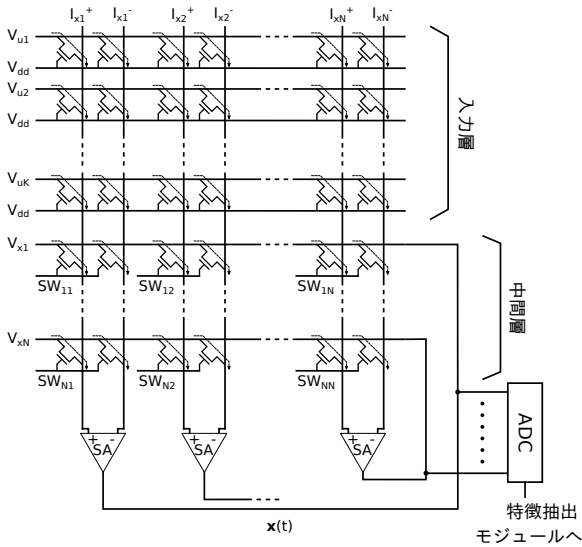


図 2: MOSFET クロスバアレイ回路

図 2 において, 入力信号は各行左端から電圧として印加され, 出力信号は各列下端から電流として得られる.

線形領域で動作させる MOSFET をコンダクタンスのように考える. サイズが  $A \times B$  のアレイの第  $i$  行と第  $j$  列を結ぶ MOSFET のオンコンダクタンスを  $G_{i,j}$  とすると, 第  $j$  列に流れる電流の和  $I_j$  は  $\sum_{i=0}^{B-1} V_i G_{i,j}$  と表すことができ,  $\mathbf{V} = [V_0, V_1, \dots, V_{B-1}]$  とすると, 次式が成り立つ.

$$\begin{aligned} \mathbf{I} &= [I_0, I_1, \dots, I_{A-1}] \\ &= \left[ \sum_{i=0}^{B-1} V_i G_{i,0}, \sum_{i=0}^{B-1} V_i G_{i,1}, \dots, \sum_{i=0}^{B-1} V_i G_{i,A-1} \right] \\ &= \mathbf{V}\mathbf{G} \end{aligned} \quad (3)$$

ここに,  $\mathbf{G} = (G_{i,j})_{i=0,1,\dots,A-1, j=0,1,\dots,B-1}$  である. 入力電圧, 出力電流であるという制限はあるものの, クロスバアレイ回路は行列とベクトルの乗算を行う回路とみなすことができる.

ESN では式 (1) のような演算, すなわち, 行列とベクトルの乗算を頻繁に行うため, クロスバアレイ回路は ESN の実装に適している. 具体的には,  $\mathbf{W}_{in} \cdot \mathbf{u}(t) + \mathbf{W} \cdot \mathbf{x}(t-1)$  の加算, 乗算をクロスバアレイ回路を用いて行う. しかし, 通常クロスバアレイ回路には, 負の重みを表現できないという欠点がある. memESN および MOSESN では, 2つのクロスバアレイ回路を使用することにより, 負の重みも表現している.

### 3.2 クロスバアレイ回路における負の重みの表現

正の値のみを用いて負の値を表現する方法として, 正の値同士の減算がある. 文献 [6,7] では, 1つの重みの表現に素子を2つ使い, 電流の差をとることにより, 疑似的に負のコンダクタンスを表現している.  $\mathbf{W}_{i,j}$  を  $G_{i,j} \equiv G_{i,j}^+ - G_{i,j}^-$ , すなわち, 2つのコンダクタンスの差で表現する. ただし,  $I_j^+ = \sum_{i=0}^{B-1} V_i G_{i,j}^+$ ,  $I_j^- = \sum_{i=0}^{B-1} V_i G_{i,j}^-$  である. 式 (3) において,  $I_j = I_j^+ - I_j^-$  とすると, 次のように変形できる.

$$\begin{aligned} I_j &= \sum_{i=0}^{B-1} V_i G_{i,j}^+ - \sum_{i=0}^{B-1} V_i G_{i,j}^- = \sum_{i=0}^{B-1} V_i (G_{i,j}^+ - G_{i,j}^-) \\ &= \sum_{i=0}^{B-1} V_i G_{i,j} \end{aligned} \quad (4)$$

### 3.3 電流減算および電流電圧変換回路

負の重みを用いるクロスバアレイ回路では, 列ごとに電流の減算を行い, さらに, 電流を電圧に変換する回路が必要となる. 電流減算および電流電圧変換回路を以後, SA (summation amplifier) と呼ぶこととする. SA はオペアンプを用いた図 3 のような回路で実装できる [6,7].

オペアンプや抵抗は理想的であるとする. オペアンプの入力抵抗は無限大であるため, 図 3 において,  $I^+$  はすべて抵抗  $R_0$  に流れ込む. さらに, 仮想接地により, 抵抗  $R_2$  に流れる電流は  $-(R_0/R_1)I^+ + I^-$  となる.  $R_0 = R_1$  と設定すると, SA の出力電圧  $V_o$  は,

$$V_o = f_{SA} \left[ -R_2 \left( -\frac{R_0}{R_1} I^+ + I^- \right) \right] = f_{SA} [R_2 (I^+ - I^-)] \quad (5)$$

と表せる. ここで  $f_{SA}$  はオペアンプの出力特性を表す関数であり, 電源電圧を  $v_e$ , 入力電圧を  $v$  とすると,  $-v_e \leq v \leq v_e$  の範囲では入力に対し線形の出力をし,  $v < -v_e$  の範囲では  $-v_e$ ,  $v > v_e$  の範囲では  $v_e$  を出力する.  $\tanh(\cdot)$  に近い振る舞いをするため, SA において電流の減算および電圧への変換と同時に  $\tanh(\cdot)$  の近似計算が可能であるといえる.

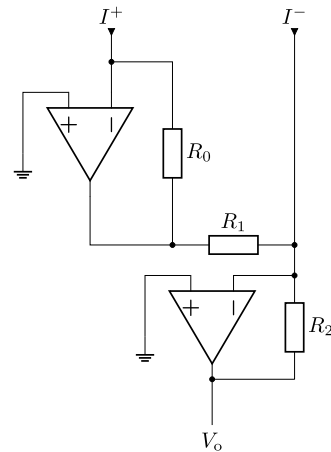


図 3: memESN および MOSESN で使用する減算器 (SA)

### 3.4 MOESN の周辺回路

MOESN には、クロスバアレイ回路や SA 以外にも、いくつかの周辺回路が必要である。その周辺回路の説明を表 1 に示す。

## 4. MOSFET クロスバアレイのシミュレーション高速化のための回路縮約

MOESN による分類が可能かどうかを確認するために、回路シミュレータを用いたシミュレーションを行う必要がある。しかし、MOESN は、クロスバアレイ回路において  $W$  を表現する部分だけでも  $2N^2$  個 ( $N$  は中間層のノード数であり、少なくとも 100 程度は必要) の MOSFET を用いるため、大規模な回路シミュレーションの実行が必要となる。そこで、MOSFET クロスバアレイ回路のシミュレーション時間を短縮するために、クロスバアレイ回路の大部分を占めるオフ状態の MOSFET に流れる電流を電流源に置換して簡略化する。本章では、MOESN のリーク電流について議論する。

### 4.1 $W$ の非零位置と分類性能の関係

本節では、MOSFET クロスバアレイ回路におけるオフ状態の MOSFET の必要性和その振る舞いを述べるとともに、シミュレーションを簡略化かつ高速化する方法を示す。2.2 節で述べたように、中間-中間層間の重み  $W$  は疎行列である必要がある。そのため、MOESN においても、オン状態で動作させる MOSFET は  $2N^2C$  個 ( $C$  は 0.025 ~ 0.15) 程度とする。  $2N^2$  個の MOSFET すべてを接続するのが本来の回路構成であり、この場合は回路使用時にゲート電圧を調整することで 1 つのアレイで様々な  $W$  を実現できる。ここでは MOSFET クロスバアレイ回路において、オフとする MOSFET を接続せず、オン状態の  $2N^2C$  個の MOSFET のみを接続してシミュレーションを高速化することを考える。ただし、実際にはオフの MOSFET は無視できず、ランダムに生成する  $W$  についても、非零位置により回帰や分類の性能が変動すると考えられることから、まず、予備的なシミュレーションを行った。

Python により実装した ESN において、密行列  $W_{\text{original}}$  を 1 つ生成し、二項乱数により、0 と 1 の割合が、 $1-C : C$  となるような  $W_{\text{mask}}$  を生成する。  $W$  は  $W_{\text{original}}$  と  $W_{\text{mask}}$  の要素積により与えられる。接続率  $C$  をある値に固定し、同一の  $W_{\text{original}}$  に対し、非零位置のみを変えて 10 回、分類のシミュレーションを行う。接続率を 0.025 ~ 0.15 の範囲で変化させながら、3 つのデータセットに対して行った結果を図 4 に示す。非零位置により分類性能にばらつきがあり、また、最適な接続率もデータセットにより異なることが分かる。シミュレーションでは、オフ状態の MOSFET も考慮して  $W$  を変えられるようにすべきである。

MOSFET はオフ状態でもドレインソース間にサブス

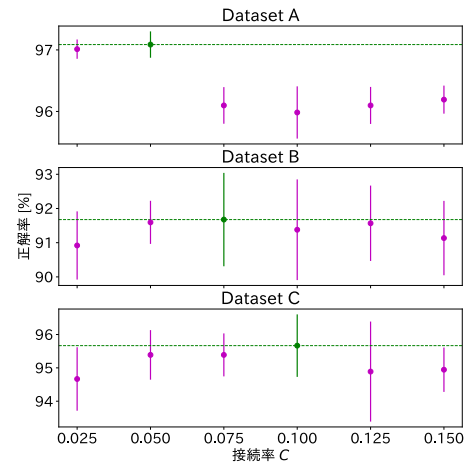


図 4: 非零位置の変化に伴う分類性能のばらつき。点は 10 回の平均を、エラーバーは標準偏差を示し、緑は一番高い正解率を表す。

レッシュヨルドリーク電流が流れる。室温ではリーク電流はオン電流に比べ極めて小さいが、回路の温度上昇に伴い指数的に増加する。またリーク電流の寄与は、アレイサイズに比例して大きくなる。このため、オフ状態の MOSFET を単に省略するだけでは実際の回路動作を再現できない可能性がある。

### 4.2 MOSFET のサブスレッシュヨルドリーク電流

ドレインソース間のリーク電流は、閾値電圧や温度をパラメータとする簡易的なモデルでは次式で表される [9,10].

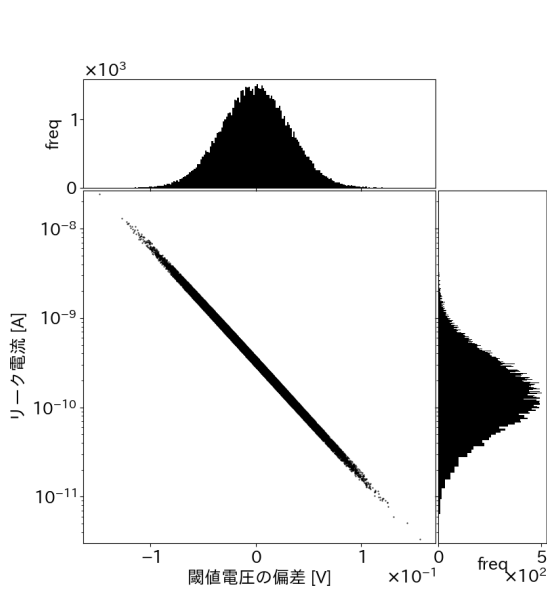
$$I_{\text{leak}} = I_0 \times \exp \{ (V_{\text{GS}} - V_{\text{th}}) / S \} \quad (6)$$

ここに、 $V_{\text{GS}}$ ,  $V_{\text{th}}$ ,  $I_0$ ,  $S$  はそれぞれ、ゲートソース間電圧、閾値電圧、 $V_{\text{GS}} = V_{\text{th}}$  の時の  $I_{\text{DS}}$ , サブスレッシュヨルドスロープを表す。サブスレッシュヨルドスロープは温度依存性をもつが、本稿では温度が一定、すなわち、 $S$  を定数として扱う。MOSFET の閾値電圧のばらつきは正規分布で表される [11]. そのため、式 (6) から、リーク電流の分布は対数正規分布に従う。室温 (25 °C) で回路シミュレーションを実行し、MOSFET クロスバアレイ回路内の MOSFET に流れるリーク電流を測定したところ、図 5(a) のようになった。図 5(a) より、閾値電圧とリーク電流の間には極めて強い相関がある。完全に直線状とならないのは、回路シミュレーションでは  $V_{\text{DS}}$  にも依存する、より厳密なモデル式を用いているためである。

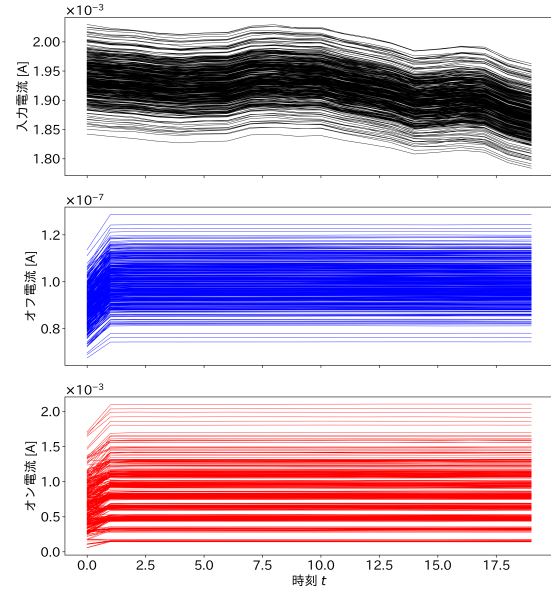
次に、MOESN におけるリーク電流の作用を考える。クロスバアレイ回路では、同じ列の MOSFET に流れる電流は加算され、MOESN では負の重みを表現するために減算を行う。各列で観測される電流値の時間変化は例えば図 5(b) のようである。図 5(b) は上から順に、データを入力する部分の MOSFET に流れる電流、 $W$  を表現する MOSFET のうち、オフ状態のものに流れる電流の列ごとの和、 $W$  を表現する MOSFET のうち、オン状態のものに

表 1: MOSESN の周辺回路

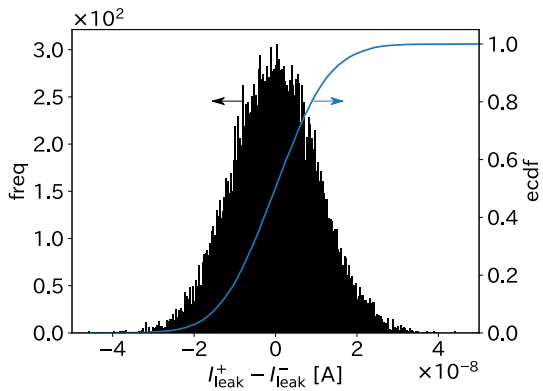
同期回路 (synchronous circuit: Sync)	離散信号を制御する
電圧線形変換回路 (voltage linear converter: VLC)	MOSFET への入力電圧範囲を調整する
電圧制限回路 (voltage restriction: VR)	SC の関数 $f_{SC}$ の電圧制限を調整する
アナログ・デジタル変換回路 (analog digital converter: ADC)	学習層へ信号を読み出す



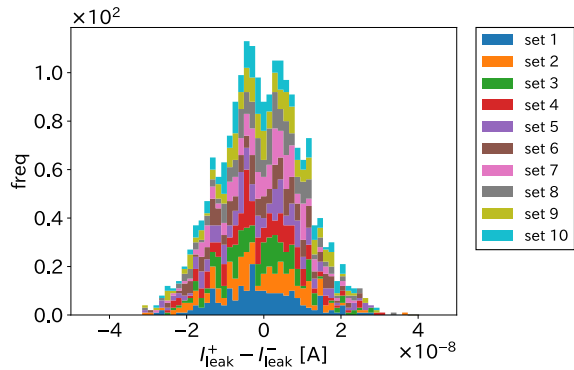
(a) 閾値電圧の偏差とリーク電流の散布図



(b) 各列に流れる電流の時間変化. 上から順に  $W_{in}$ ,  $W$  の零要素,  $W$  の非零要素に対応する MOSFET に流れる電流を列ごとに足し合わせたもの.



(c) シミュレーションにより得た  $n_j = 195$  におけるリーク電流  $I_{leak}^+(j) - I_{leak}^-(j)$  のヒストグラムと経験累積分布関数



(d) 乱数を用いて擬似的に生成した  $I_{leak}^+(j) - I_{leak}^-(j)$  の分布.  $N = 200, C = 0.025$  の MOSFET リザバーに対して 10 回分生成したものの.

図 5: MOSESN におけるリーク電流の振る舞いと乱数による生成結果

流れる電流の列ごとの和を示す. リーク電流の時間的変化はあるが, 変動の絶対値は小さく, 無視できると仮定する.

いま, クロスバレイ回路の第  $j$  列の MOSFET のうち, オフ状態のもの個数を  $n_j$ , 2つのアレイの第  $j$  列に流れるリーク電流の列ごとの和を  $I_{leak}^+(j)$ ,  $I_{leak}^-(j)$  と表すこととする. 母集団から抽出した標本の部分和について, 中心極限定理が成り立つ [12]. 中心極限定理により,  $n_j$  が十分大きいとき,  $I_{leak}^+(j) - I_{leak}^-(j)$  は, 平均 0, 分散  $2n_j V(I_{leak})$  の正規分布に従う. なお,  $V(I_{leak})$  は, 1つの

MOSFET に流れるリーク電流の分散であり, 回路シミュレータの出力から求めることができる. 図 5(c) は, 全ての列でちょうど 5 個だけの MOSFET がオンになるように設定した MOSESN のシミュレーションを 100 回行って得た  $I_{leak}^+(j) - I_{leak}^-(j)$  のヒストグラムである. なお, 中間層のノード数, 接続率はそれぞれ  $N = 200, C = 0.025$  とした. 得られた結果に Kolmogorov-Smirnov 検定 [13] を行ったところ, 有意水準 5% で,  $I_{leak}^+(j) - I_{leak}^-(j)$  は正規分布に従っているとみなすことができる.

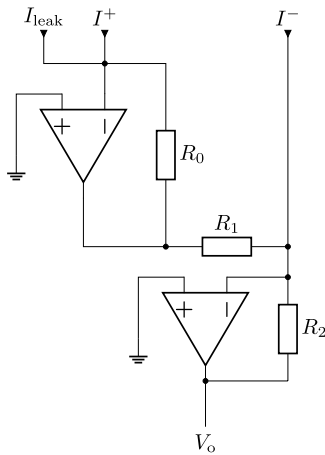


図 6: リーク電流を電流源で代替したときの SA 内部. 擬似的に生成した  $I_{\text{leak}} = I_{\text{leak}}^+(j) - I_{\text{leak}}^-(j)$  を入力する.

以上の検討から, 本稿では, 次の手順でオフの MOSFET を電流源に置き換える.

- (1)  $W$  の非零要素の位置, すなわち, オン状態で使用する MOSFET をランダムに決める.
- (2) すべての列について平均 0, 分散  $2n_j V(I_{\text{leak}})$  の正規分布に従う乱数を用い, 各列で減算を行ったあとのリーク電流の値  $I_{\text{leak}}^+(j) - I_{\text{leak}}^-(j)$  をランダムに求める.
- (3) 2 で求めた値に相当する電流源を各列の SA 内部に接続する.

実際, この手順で擬似的に作成した  $I_{\text{leak}}^+(j) - I_{\text{leak}}^-(j)$  は図 5(d) のようになった.

## 5. リーク電流の縮約の評価

第 4 章では, MOSFET クロスバアレイ回路のシミュレーションを簡略化する手法の提案を行った. 本章では, その手法の評価を行う.  $N = 200$ ,  $C = 0.025$  として, 1 次元および 12 次元の時系列データを入力したときの回路シミュレーションを比較する. なお, 電流源の接続は, 3.3 節で述べた SA の内部で, 図 6 に示すように行った. 回路シミュレーションには, Synopsis 社の HSPICE を使用した. time コマンドで実行時間を測定した結果が表 2 である. すべての MOSFET を接続する手法の最大 16.6 倍の速度で回路シミュレーションが実行できた. 接続率, すなわち, オンの MOSFET の割合  $C$  が 2.5 % 程度であることから, MOSFET の削減数に応じた高速化が可能となっている. なお, 入力層の重み  $W_{\text{in}}$  を表現する部分の MOSFET は  $NK$  個をすべてオン状態で接続しているため,  $NC$  を一定に保ったまま  $N$  を大きくしたときも, シミュレーション時間は長くなる.

また, 2 に示したシミュレーション時間は, 1 時刻分の入力時系列に対して 1 時刻分のリザーバーの内部状態を求める, すなわち,  $u(t)$  から  $x(t)$  を求めるのにかかる時間である. 分類タスクのデータセットを扱う際は, この計算

表 2:  $K = 1, 12$  における回路シミュレーションの実行時間の比較

手法	実行時間 [s]	
	1 次元	12 次元
従来手法 (すべての MOSFET を接続)	50.534	55.017
提案手法	3.043	4.127

を, 小さいデータセットでは 10,000 回程度, 大きいデータセットでは 700,000 回以上行うことになる. 並列化を行っても表 2 に示す実行時間比は保存される. データの前処理や学習にかかる時間はシミュレーション時間に比べ十分短いため, 回路シミュレーションの高速化により MOSESN 全体のシミュレーション時間を短縮できる.

## 6. まとめ

本稿では, リザーバー部に MOSFET クロスバアレイ回路を用いたハードウェア ESN(MOSESN) のシミュレーションの高速化を試みた. MOSESN では, クロスバアレイ回路の素子数の大きさから, 回路解析に多大な時間を要する. そこで, オフ状態の MOSFET のサブスレッショルドリーク電流の振る舞いを確認し, 電流源に置換することで, シミュレーション時間を大幅に短縮することに成功した.

今後の課題としては, MOSFET リザーバーの集積回路化に向けた周辺回路の設計や, 回路の温度上昇などといった環境変動に対するロバスト性の確保が挙げられる. MOSFET クロスバアレイ回路は, 最小スケールで設計可能であり, 比較的小さい面積かつ低消費電力での動作が可能であると考えられる. 一方で, 本稿で扱った構成の MOSFET リザーバーは, 数百個のオペアンプを常時駆動させる必要があり, 回路面積や消費電力の点で問題がある. これらの問題を解決し, ソフトウェア実装より低消費電力かつ高速に動作する集積回路として実装できる ESN に関する検討が必要である.

## 謝辞

本研究の一部は JSPS 科研費基盤研究 (B)20H04156 の助成を受けた.

## 参考文献

- [1] H. Jaeger, “Adaptive nonlinear system identification with echo state networks,” *Advances in neural information processing systems*, vol. 15, pp. 609–616, 2002.
- [2] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [3] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, “Recent advances in physical reservoir computing: A review,” *Neural Networks*, vol. 115, pp. 100–123, 2019.
- [4] T. Natschläger, W. Maass, and H. Markram, “The” liquid computer”: A novel strategy for real-time computing on time series,” *Special issue on Foundations of Infor-*

- mation Processing of TELEMATIK*, vol. 8, no. 1, pp. 39–43, 2002.
- [5] G. Tanaka, “リザバーコンピューティングの概念と最近の動向,” *The journal of the Institute of Electronics, Information and Communication Engineers*, vol. 102, no. 2, pp. 108–113, 2019.
- [6] Y. Kume, S. Bian, and T. Sato, “A tuning-free hardware reservoir based on mosfet crossbar array for practical echo state network implementation,” in *Asia and South Pacific Design Automation Conference*, 2020, pp. 458–463.
- [7] A. M. Hassan, H. H. Li, and Y. Chen, “Hardware implementation of echo state networks using memristor double crossbar arrays,” in *International Joint Conference on Neural Networks*, 2017, pp. 2171–2177.
- [8] J. Moon, W. Ma, J. H. Shin, F. Cai, C. Du, S. H. Lee, and W. D. Lu, “Temporal data classification and forecasting using a memristor-based reservoir computing system,” *Nature Electronics*, vol. 2, no. 10, pp. 480–487, 2019.
- [9] A. O. Adan and K. Higashi, “Off-state leakage current mechanisms in bulksi and soi mosfets and their impact on cmos ulsis standby current,” *IEEE Transactions on Electron Devices*, vol. 48, no. 9, pp. 2050–2057, 2001.
- [10] J. Kao, S. Narendra, and A. Chandrakasan, “Subthreshold leakage modeling and reduction techniques,” in *International conference on Computer-aided design*, 2002, pp. 141–148.
- [11] 平本, 竹内, and 西田, “MOS トランジスタのスケールアップに伴う特性ばらつき,” *The journal of the Institute of Electronics, Information and Communication Engineers*, vol. 92, no. 6, pp. 416–426, Jun 2009.
- [12] F. J. Anscombe, “Large-sample theory of sequential estimation,” in *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 48, no. 4, 1952, pp. 600–607.
- [13] F. J. Massey Jr, “The kolmogorov-smirnov test for goodness of fit,” *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.