

# ストカステック演算によって実現された FIR フィルタにおける内部雑音に関する検討

## Study on Internal Noise of FIR Filters Implemented with Stochastic Computing

○鎌田裕成, 越田俊介, 阿部正英, 川又政征

○Hironari Kamada, Shunsuke Koshita, Masahide Abe, Masayuki Kawamata

東北大学大学院工学研究科電子工学専攻

Department of Electronic Engineering, Graduate School of Engineering,  
Tohoku University

キーワード: ストカステック演算 (stochastic computing), FIR フィルタ (FIR filter), 分散伝達関数 (variance transfer function)

連絡先: 〒 980-8579 仙台市青葉区荒巻字青葉 6-6-05

東北大学 大学院工学研究科 電子工学専攻 川又・阿部 (正) 研究室

鎌田 裕成, Tel.: (022)795-7095, Fax.: (022)263-9169, E-mail: kamada@mk.ecei.tohoku.ac.jp

### 1. まえがき

ストカステック演算 (以下 SC) は, 近似計算法のひとつで, 数値を確率的に表現し演算する手法である. 従来の二進数演算と比較して大幅に回路規模を削減することが可能な手法として, 近年注目を集めている. SC の歴史は古く, 1960年代後半には演算手法が確立された<sup>1)</sup>. 演算回路が小型である, ソフトエラー耐性が高いなどの利点が存在する一方で, 演算結果が確率的である点, 演算時間が長くなってしまふ点などを理由に, SC が使われることはほとんどなかった. その SC が近年, 注目を集めている理由としては, デバイスの小型化や省電力化を目的として回路規模の削減が求められる場合が増えている挙げられる.

SC は画像処理, デジタルフィルタ, 符号化, デイプラーニングなど, 広い分野での応用が検討されている<sup>2-6)</sup>. SC では演算結果が確率的に変動するため, 出力に確率的な変動が含まれていてもそれ

を許容できるアプリケーションへの応用が望ましい. 逆に, 確率的変動が許容できるアプリケーションであれば, SC を用いることで回路規模の大幅な削減を実現することが可能となる.

SC の確率的ふるまいが演算結果に与える影響については, 未だ詳しく明らかにされていない. そこで本稿では, SC をデジタルフィルタに適用した際に, SC による演算結果の確率的変動がフィルタ出力にどのような影響を与えるかについて考察する.

### 2. ストカステック演算

本章では, SC の基礎について説明する. 初めに, SC における数値の表現法について説明する. 次に, SC を構成する演算素子と 2 進数との変換について説明する.

## 2.1 数値の表現

SC では、ビット列内のビット ‘1’ の存在確率によって数値を表現する。例えば、次の2つのビット列、(1 1 0 0) と (0 1 1 0) はどちらも  $1/2$  の確率でビット ‘1’ が存在している。この2つは異なる並びのビット列であるが、SC においては等しい数値を表現する。以下では、前述のように確率によって数値を表現するビット列のことをストカスティック数 (SN) と呼ぶ。

SN の数値表現は、ユニポーラ表現とバイポーラ表現が知られている。以下ではこの2種類の表現法について説明する。

ユニポーラ表現では、SN におけるビット ‘1’ の存在確率を、そのまま数値として扱う。すなわち、語長  $N$  の SN  $X$  がユニポーラ表現によって表す数値  $\hat{x}$  は、以下のように定められる。

$$\hat{x} = \frac{X \text{ 内のビット '1' の数}}{N} \quad (1)$$

ユニポーラ表現では範囲  $[0, 1]$  の数値を表現することが可能であり、数値の刻みは  $1/N$  である。例として、SNA: (1 1 0 0) を考える。4ビットのうち2ビットが ‘1’ であるから、 $A$  がユニポーラ表現で表す数値  $\hat{a}$  は  $\hat{a} = 2/4$  である。

前述のとおり、ユニポーラ表現で表現可能な数値の範囲は  $[0, 1]$  であり、負の値を表現することができない。負の値を扱えるように、表現可能な数値の範囲を拡張した表現法がバイポーラ表現である。SN  $X$  がバイポーラ表現によって表す数値  $\hat{x}'$  は、式 (1) の  $\hat{x}$  を用いて以下のように定められる。

$$\hat{x}' = 2\hat{x} - 1 \quad (2)$$

バイポーラ表現では、範囲  $[-1, 1]$  の数値を表現可能である。また、バイポーラ表現された  $N$  ビット SN の数値の刻みは  $2/N$  である。ユニポーラ表現の場合と同様に、例として SNA: (1 1 0 0) について考える。 $\hat{a} = 2/4$  であり、バイポーラ表現の場合には、 $A$  が表す数値  $\hat{a}'$  は  $\hat{a}' = 2\hat{a} - 1 = 0$  である。デジタルフィルタにおいては、負の値を扱う必要があるため、本稿ではバイポーラ表現を用いる。

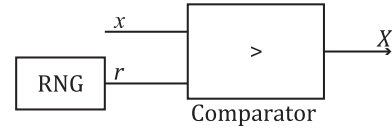


Fig. 1 乱数発生器と比較器を用いた SN の生成

## 2.2 2進表現からストカスティック表現への変換

2進数から SN への変換には、Fig.1 に示すように、乱数発生器 (Random Number Generator) と比較器 (Comparator) からなる SNG (Stochastic Number Generator) を用いる。ここで用いる乱数発生器は、範囲  $[0, 1]$  の一様分布の乱数を発生させる。乱数発生器には、線形フィードバックシフトレジスタ (LFSR: Linear Feedback Shift Register) が多く用いられる。はじめに、ユニポーラ表現への変換について説明する。ユニポーラ表現では、与えられた2進数  $x$  に対して、SN  $X$  の  $i$  番目のビット  $X_i$  を以下の確率  $P(X_i = 1)$  にしたがって発生させる。

$$P(X_i = 1) = x \quad (3)$$

これは、Fig.1 に示した回路を用いて次のように実現できる。一様乱数  $r$  と、変換したい2進数  $x$  を比較器に入力する。 $x$  の範囲は  $[0, 1]$  とする。比較器の出力は以下のように定める。

- $x > r$  のとき、1 を出力
- $x \leq r$  のとき、0 を出力

この処理を  $N$  回繰り返すことで、語長  $N$  の SN が得られる。

一方、バイポーラ表現では、与えられた2進数  $x$  に対して以下の確率  $P'(X_i = 1)$  にしたがって  $X_i$  を発生させる。

$$P'(X_i = 1) = \frac{1}{2}(x + 1) \quad (4)$$

バイポーラ表現に変換する場合は、 $x$  の範囲は  $[-1, 1]$  である。そのため、 $x$  を  $r$  と同じ  $[0, 1]$  の範囲にスケールしたものを、すなわち

$$x' = \frac{1}{2}(x + 1) \quad (5)$$

によって得られる  $x'$  と一様乱数  $r$  との比較をする。

このように、2進数からSNへの変換には乱数を用いるため、変換結果が一意に定まらず、変換前と変換後で表される数値が変化してしまう可能性がある。すなわち、変換前の2進数  $x$  と、 $x$  を変換して得たSNXが表す数値  $\hat{x}$  が、 $\hat{x} \neq x$  となる場合が存在する。

### 2.3 ストカスティック表現から2進表現への変換

SNから2進数への変換には、カウンタ回路が用いられる。カウンタ回路によってSNX内のビット‘1’の数をかぞえ、式(1)と式(2)から数値を求める。2進数からSNへの変換と異なり、あるSNXから得られる値は一意に定まる。

### 2.4 乗算

次に、SN同士の乗算について述べる。乗算を実現する方法はユニポーラ表現とバイポーラ表現とで異なる。ここでは、ユニポーラ表現の場合について述べる。

SN同士の乗算は、2つのビット列におけるビット‘1’の同時生起確率を求めることで実現できる。そこで、ユニポーラ表現されたSN同士の乗算は、2つのSNが互いに無相関であることを前提として、ビット列同士の論理積をとることで求める。互いに無相関であるSNAとBの論理積をCとすると、以下の式が成り立つ。

$$P(C_i = 1) = P(A_i = 1)P(B_i = 1) \quad (6)$$

式(3)と(6)より、以下の式が成り立つ。

$$c = a \times b \quad (7)$$

式(7)より、SN同士の論理積をとることで乗算が実現できることがわかる。例えば、ユニポーラ表現された2つのSN、A:(1100)とB:(0110)の乗算を考える。これを図示したものがFig.2である。それぞれが表す数値  $\hat{a}, \hat{b}$  は  $\hat{a} = \hat{b} = 1/2$  であるから、積  $\hat{a} \times \hat{b}$  は  $1/4$  である。AとBの論理積をCとすると、C:(0100)である。Cが表現する数値  $\hat{c}$  は

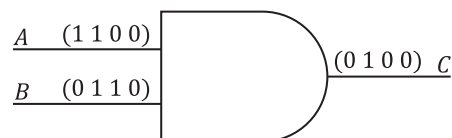


Fig. 2 ユニポーラ表現における乗算の例

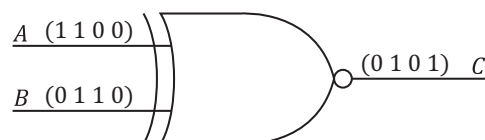


Fig. 3 バイポーラ表現における乗算の例

$\hat{c} = 1/4$  であり、 $\hat{c} = \hat{a} \times \hat{b}$  を満たしていることがわかる。

一方で、バイポーラ表現されたSN同士の乗算は、排他的論理和の否定をとることで計算される。この処理は、回路上でXNORゲートによって実現される。例えば、バイポーラ表現された2つのSNA:(1100)とB:(0110)の乗算を考える。これを図示したものがFig.3である。AとBそれぞれがバイポーラ表現で表す数値  $\hat{a}', \hat{b}'$  は  $\hat{a}' = \hat{b}' = 0$  であるから、積  $\hat{a}' \times \hat{b}'$  は0である。AとBの排他的論理和の否定をCとすると、C:(0101)である。Cが表現する数値  $\hat{c}$  は  $\hat{c} = 0$  であり、 $\hat{c} = \hat{a}' \times \hat{b}'$  を満たしている。

このように、SCでは、乗算を単一の論理ゲートのみで行うことができる。そのため、従来の2進数演算に比べて回路における乗算器の面積を大幅に削減することができる。

### 2.5 加算

本節では、SNの加算について述べる。SNは確率によって数値を表しているため、絶対値が1を超える数値を表すことができない。そのため、数値  $a$  と  $b$  の和  $c$  は、

$$c = \frac{1}{2}(a + b) \quad (8)$$

のように結果の絶対値が1を超えないようにスケールリングされる必要がある。このようなSN同士の加算は、ユニポーラ表現とバイポーラ表現のどちらの場合においてもマルチプレクサによって実現できる。

マルチプレクサの入力を  $A$ ,  $B$ , 出力を  $C$ , 制御信号を  $S$  とする. それぞれの  $i$  番目のビットにおける '1' の存在確率を  $P(A_i = 1)$ ,  $P(B_i = 1)$ ,  $P(C_i = 1)$ ,  $P(S_i = 1)$  とする. マルチプレクサの出力信号の  $i$  番目のビットにおける '1' の存在確率  $P(C_i = 1)$  は以下のように書くことができる.

$$P(C_i = 1) = P(S_i = 1)P(A_i = 1) + (1 - P(S_i = 1))P(B_i = 1) \quad (9)$$

一般に, 2つのSNの加算においては,  $P(S_i = 1) = 1/2$  となるような制御信号  $S$  を用いる. このとき, 式(9)は以下のようになる.

$$P(C_i = 1) = \frac{1}{2}(P(A_i = 1) + P(B_i = 1)) \quad (10)$$

式(10)と, 式(3)もしくは式(4)より, マルチプレクサを用いることによって式(8)に示した加算ができることがわかる. 式(10)からもわかるように, マルチプレクサを用いた加算では, 加算結果の絶対値が1を超えないようにスケーリングされて出力される点に注意が必要である. そのため, SNを2進数に変換したあと, スケーリングの補正が必要となる. 式(10)に示した処理では, 加算結果を1/2倍にスケーリングしたものを出力している. 和を元のスケールで得るためには, 出力を2進数に変換したのち, 2倍する必要がある. また,  $P(S_i = 1) \neq 1/2$  であるとき, マルチプレクサを用いた加算は重み付き加算となる.

マルチプレクサの制御信号  $S$  は,  $P(S_i = 1)$  の確率でビット '1' が出現するビット列である. この信号の生成には, ユニポーラ表現における2進数からSNへの変換器を用いる.

マルチプレクサを用いたユニポーラ表現によるSN同士の加算の例をFig.4に示す. 入力は  $A : (1\ 1\ 0\ 0)$ ,  $B : (0\ 1\ 1\ 0)$ , 制御信号は  $S : (1\ 0\ 1\ 0)$  である.  $\hat{a} = \hat{b} = 2/4$  であるから,  $\hat{a} + \hat{b} = 1$  である. 一方,  $\hat{c} = 2/4$  であるから,  $\hat{c} = \frac{1}{2}(\hat{a} + \hat{b})$  を満たしている. マルチプレクサを用いて正しくSNの加算が実現できていることがわかる.

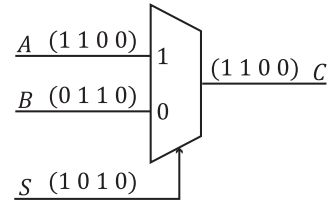


Fig. 4 マルチプレクサを用いた加算の例

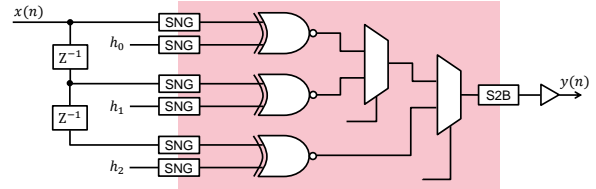


Fig. 5 SCに基づくデジタルフィルタの例

### 3. 内部雑音の発生と伝搬

#### 3.1 内部雑音の原因

入出力を2進数とし, 演算をSCで行うデジタルフィルタの例をFig.5に示す. SCに基づくデジタルフィルタにおける内部雑音の原因としては, 以下の三つが考えられる.

- SNGにおける入力の分散
- 演算における確率的変動
- 加算のスケーリング補正における誤差の拡大

これらの雑音をモデル化したブロック図をFig.6に示す. SNGは確率的プロセスを伴うため, 生成されたSNは確率的変動を持つ. そしてフィルタの乗算, 加算の過程でさらに確率的に変動する. これらの確率的変動が, 内部雑音としてフィルタ出力に生じる. 最後に, 加算によるスケーリングを補正するために数値を定数倍する必要がある. ここで内部雑音が拡大される.

フィルタの内部雑音を解析するためには, SNGで生じる分散と, 演算素子での分散の伝搬について考える必要がある. 以降では, この点について考察する.

#### 3.2 分散伝達関数

分散伝達関数<sup>7)</sup>を用いることで, 入力シーケンスの, 各演算素子における分散の伝搬を計算によって求めることができる. 各演算素子の分散伝達関数は

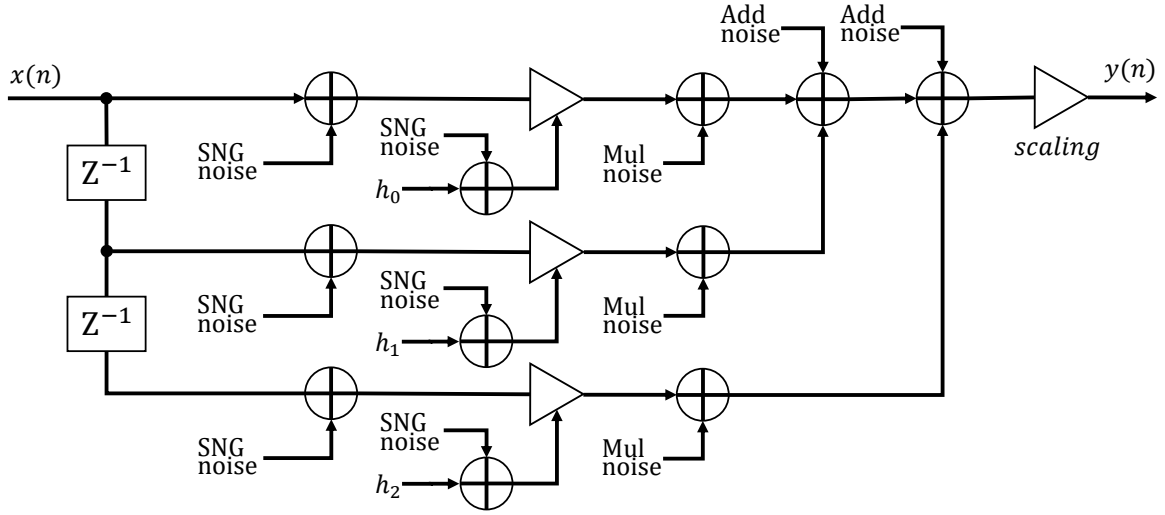


Fig. 6 SCに基づくデジタルフィルタで生じる雑音のモデル

以下の通りである．ここで， $N$  は SN の語長， $a$ ， $b$  は 2 つの入力， $s$  はマルチプレクサの選択信号， $E'_a$  は  $(1 - E_a)$  を表す． $E$  はビットストリーム内の正規化されたビット '1' の数の期待値， $\sigma^2$  は正規化されたビット '1' の数の分散である．

#### AND

$$E_A(a, b) = E_a E_b \quad (11)$$

$$\sigma_A^2(a, b) = \frac{1}{N-1} (E_a - \sigma_a^2 - E_a^2)(E_b - \sigma_b^2 - E_b^2) + \sigma_a^2 E_b^2 + \sigma_b^2 E_a^2 + \sigma_a^2 \sigma_b^2 \quad (12)$$

#### OR

$$E_O(a, b) = E_a + E_b - E_a E_b \quad (13)$$

$$\sigma_O^2(a, b) = \frac{1}{N-1} (E'_a - \sigma_a^2 - E_a'^2)(E'_b - \sigma_b^2 - E_b'^2) + \sigma_a^2 E_b'^2 + \sigma_b^2 E_a'^2 + \sigma_a^2 \sigma_b^2 \quad (14)$$

#### XNOR

$$E_X(a, b) = E_a E_b + E'_a E'_b \quad (15)$$

$$\sigma_X^2(a, b) = \sigma_A^2(a, b) + \sigma_A^2(a', b') + 2\{(E_a - \sigma_a^2 - E_a^2)(E_b - \sigma_b^2 - E_b^2) - E_a E_b E'_a E'_b\} \quad (16)$$

#### MUX

$$E_M(a, b, s) = E_a E_s + E_b E'_s \quad (17)$$

$$\sigma_M^2(a, b, s) = \sigma_A^2(a, s) + \sigma_A^2(b, s') - 2E_a E_b \sigma_s^2 \quad (18)$$

Table 1 実験条件

入力信号	範囲 $[-1, 1]$ , 刻み 0.05
信号長	$10^4$
SN の語長	$2^{12}$

## 4. 単一演算回路の出力の分散

SNG, XNOR による乗算，MUX による加算について，それぞれの回路における出力の分散をビットレベルのシミュレーションによって測定し，分散伝達関数による分散の理論値と比較した．

SNG の出力は二項分布に従う．よって，入力を  $x$  とすると SNG 出力の期待値  $E_{\text{SNG}}(x)$  と分散  $\sigma_{\text{SNG}}^2(x)$  は以下ようになる．

$$E_{\text{SNG}}(x) = \frac{1}{2}(x+1) \quad (19)$$

$$\sigma_{\text{SNG}}^2(x) = \frac{1}{4}(x+1)(1-x) \quad (20)$$

入力する値と出力の分散の関係を調べるため，入力値を  $[-1, 1]$  の範囲で変化させて測定した．入出力は 64bit 浮動小数点数である．シミュレーションには MATLAB を使用した．その他の条件は表 1 に示す．加算と乗算の分散は，SNG の分散も含んでいることに注意が必要である．

結果を Fig.7-11 に示す．いずれの結果も，扱う数値が 0 に近いほど分散が大きくなることを示している．

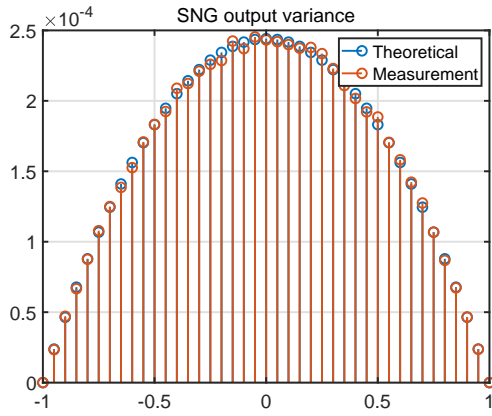


Fig. 7 SNG の分散

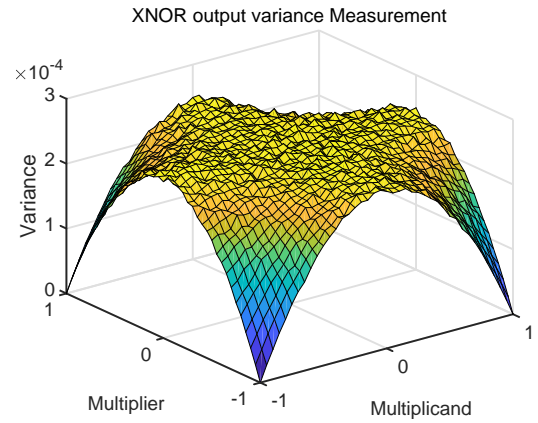


Fig. 8 XNOR の分散の実測値

入力信号	範囲 $[-1, 1]$ の一様乱数から取り出した標本系列
信号長	1000
SN 語長	$2^{12}$
フィルタ係数	範囲 $[-1, 1]$ , 刻み 0.1
試行回数	1000

## 5. FIR フィルタの出力の分散

### 5.1 フィルタ係数と出力の分散

SCに基づく FIR フィルタにおけるフィルタ係数と出力の分散の関係を調べるため、係数を変化させながらフィルタ出力の分散をビットレベルのシミュレーションで測定し、分散伝達関数と比較した。一様乱数から取り出した標本系列を複数回入力し、各時刻における出力の分散を求めた。分散の時間平均を各係数における分散とした。実験条件を表 2 に示した。結果を Fig.12,13 に示した。係数が 0 に近いほど出力の分散が大きくなるのがわかる。これは、SNG における分散が伝搬し出力に影響しているからだと考えられる。

### 5.2 タップ数・語長と出力の分散

SCに基づく FIR フィルタにおけるフィルタのタップ数および SN の語長と出力の分散の関係を調べるため、タップ数と語長を変化させながらフィルタ出力の分散を測定し、分散伝達関数と比較した。分散の求め方は前章と同様である。実験条件を表 3 に示

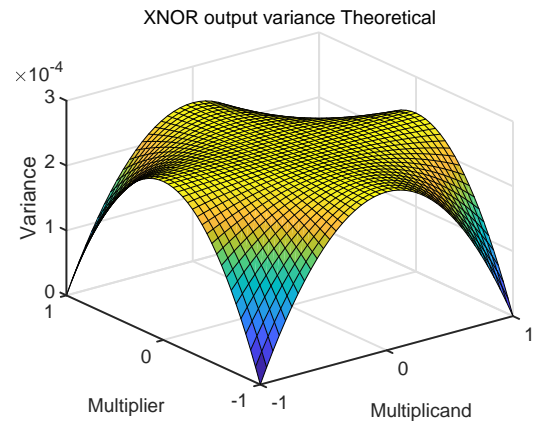


Fig. 9 XNOR の分散の理論値

した。結果を Fig.14,15 に示した。実験結果は以下のことを示している。

- 出力の分散はタップ数の 2 乗に比例する。
- 出力の分散は SN の語長に反比例する。

### 5.3 2 タップフィルタの出力の分散

FIR フィルタの出力の分散を求めるためには、フィルタのブロック図に従って、入力側から順に分散伝達関数を計算していけばよい。最初に求めるのは、SNG における平均と分散である。SNG の出力は二項分布に従う。二項分布  $B(N, p)$  の平均  $\mu$  と分散  $\sigma^2$  は、 $\mu = Np$ ,  $\sigma^2 = Np(1-p)$  である。  $N$  は試行回数、  $p$  は成功確率である。SNG においては、  $N$  は SN の語長、  $p$  はビット 1 の生成確率に相当する。バイポーラ表現の場合、入力実数  $x$  に対して、ビット 1 の生成確率は  $\frac{x+1}{2}$  である。すなわち、SNG 出力の、正規化



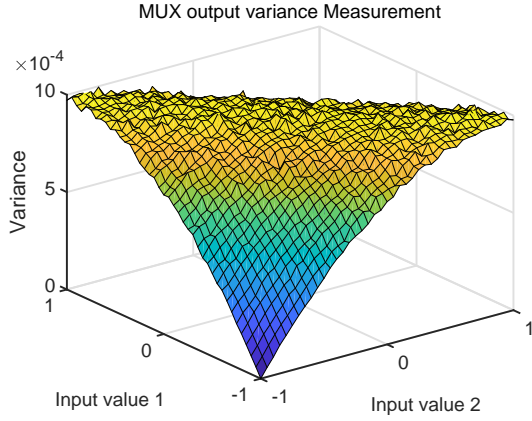


Fig. 10 MUX の分散の実測値

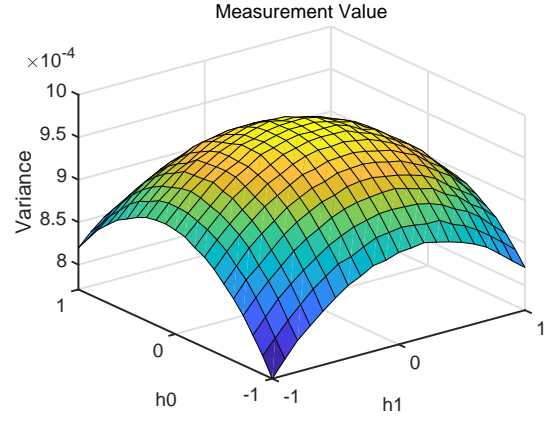


Fig. 12 フィルタ出力の分散の実測値

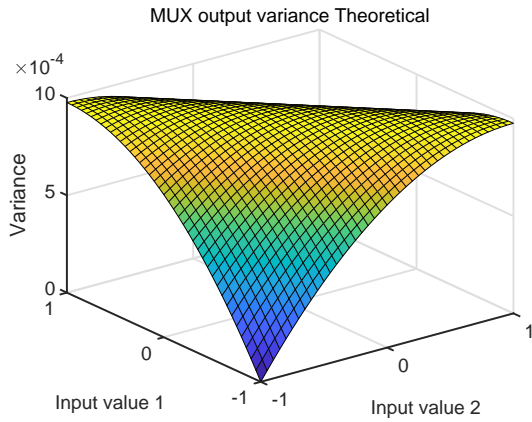


Fig. 11 MUX の分散の理論値

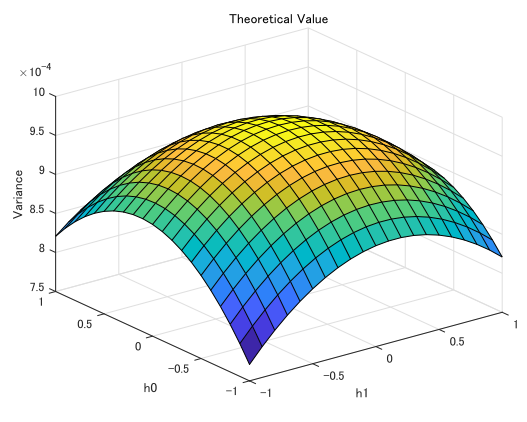


Fig. 13 フィルタ出力の分散の理論値

されたビット1の数の平均  $E_{\text{SNG}}(x)$  と分散  $\sigma_{\text{SNG}}^2(x)$  は、 $E_{\text{SNG}}(x) = \frac{x+1}{2}$ 、 $\sigma_{\text{SNG}}^2(x) = \frac{(x+1)(1-x)}{4}$  である。入力実数の値に応じて、変換後のSNの分散が変化することがわかる。ある時刻の入力  $x(n)$  に対するSNG出力  $X(n)$  の期待値と分散は以下ようになる。ここで  $x(n)$  は範囲  $[-1, 1]$  の一様乱数とする。

$$E_{X(n)} = \frac{x(n) + 1}{2} \quad (21)$$

$$\sigma_{X(n)}^2 = \frac{(x(n) + 1)(1 - x(n))}{4} \quad (22)$$

ここでいう期待値は、同一入力下において、ある特定の時刻の多数の測定値から求めた、集合平均である。また、この期待値と分散は、ビット長で正規化された、ビット列内のビット1の数の期待値と分散を表す。前述の通り、SCでは、SNGにより値が変動し、その変動のパラメータは変換前の値によって決定されることから、このような表現を用いる。次に、入力  $x(n)$  とフィルタ係数  $h_0$  をそれぞれSNへ変換し、乗算したものの  $X(n)H_0$  の期待値と分散を求

める。この処理はXNORゲートにより実装されるので、XNORゲートの分散伝達関数より以下の式が得られる。

$$\begin{aligned} E_{H_0 X(n)} &= E_{X(n)} E_{H_0} + (1 - E_{X(n)})(1 - E_{H_0}) \\ &= \frac{h_0 x(n) + 1}{2} \end{aligned} \quad (23)$$

$$\begin{aligned} \sigma_{H_0 X(n)}^2 &= \frac{1}{N-1} \{ (1 - E_{X(n)}) - \sigma_{X(n)}^2 - (1 - E_{X(n)}^2) \} \\ &\quad \times \{ (1 - E_{H_0}) - \sigma_{H_0}^2 - (1 - E_{H_0}^2) \} \\ &\quad + \sigma_{X(n)}^2 (1 - E_{H_0})^2 + \sigma_{H_0}^2 (1 - E_{X(n)}^2) \\ &\quad + \sigma_{X(n)}^2 \sigma_{H_0}^2 \\ &= \frac{1 - h_0^2 x(n)^2}{4} \end{aligned} \quad (24)$$

同様に、 $x(n)$  の1時刻遅延  $x(n-1)$  と係数  $h_1$  の積については以下ようになる。

$$E_{H_1 X(n-1)} = \frac{h_1 x(n-1) + 1}{2} \quad (25)$$

Table 3 実験条件

入力信号	範囲 $[-1, 1]$ の一様乱数から取り出した標本系列
信号長	250
SN 語長	$[2^1, 2^{12}]$
フィルタタップ数 $N$	$[2^1, 2^8]$
フィルタ係数	遮断周波数 $\omega_c = 0.5\pi$ $h(n) = \frac{\omega_c}{\pi} \text{sinc} \frac{\omega_c}{\pi} n \times \omega(n)$ $n = -\frac{N-1}{2}, \frac{N-1}{2}$ $\omega(n) = 0.54 + 0.46 \cos \frac{2\pi n}{N-1}$ 低域通過フィルタ
試行回数	200

$$\sigma_{H_1 X(n-1)}^2 = \frac{1 - h_1^2 x(n-1)^2}{4} \quad (26)$$

$h_0 x(n) + h_1 x(n-1)$  を表現する  $H_0 X(n) + H_1 X(n-1)$  を実現するマルチプレクサ出力  $Y$  の期待値と分散を考える. マルチプレクサの制御信号  $S$  は確率  $1/2$  でビット  $1$  が出現する系列と仮定する. このとき

$$E_S = 1/2 \quad (27)$$

$$\sigma_S^2 = 1/4 \quad (28)$$

である.  $Y$  の期待値と分散は, 分散伝達関数から以下のように求めることができる..

$$\begin{aligned} E_Y &= E_{H_0 X(n)} E_S + E_{H_1 X(n-1)} (1 - E_S) \\ &= \frac{h_0 x(n) + h_1 x(n-1) + 2}{4} \end{aligned} \quad (29)$$

$$\begin{aligned} \sigma_Y^2 &= \frac{1}{N-1} (E_{H_0 X(n)} - \sigma_{H_0 X(n)}^2 - E_{H_0 X(n)}^2) \\ &\quad \times (E_S - \sigma_S^2 - E_S^2) + \sigma_{H_0 X(n)}^2 E_S^2 \\ &\quad + \sigma_S^2 E_{H_0 X(n)}^2 + \sigma_{H_0 X(n)}^2 \sigma_S^2 \\ &+ \frac{1}{N-1} (E_{H_1 X(n-1)} - \sigma_{H_1 X(n-1)}^2 - E_{H_1 X(n-1)}^2) \\ &\quad \times ((1 - E_S) - \sigma_S^2 - (1 - E_S)^2) \\ &\quad + \sigma_{H_1 X(n-1)}^2 (1 - E_S)^2 \\ &\quad + \sigma_S^2 E_{H_1 X(n-1)}^2 + \sigma_{H_1 X(n-1)}^2 \sigma_S^2 \\ &\quad - 2E_{H_0 X(n)} E_{H_1 X(n-1)} \sigma_S^2 \end{aligned} \quad (30)$$

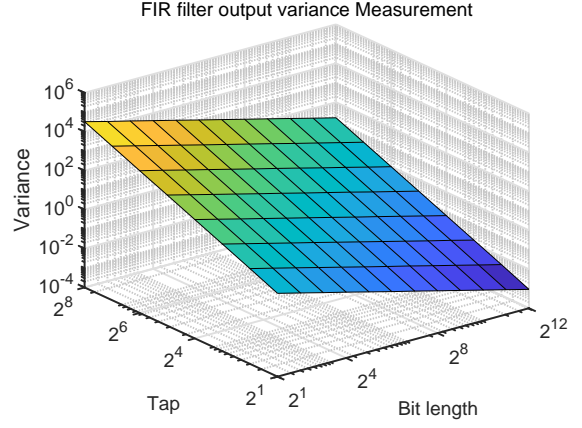


Fig. 14 フィルタ出力の分散の実測値

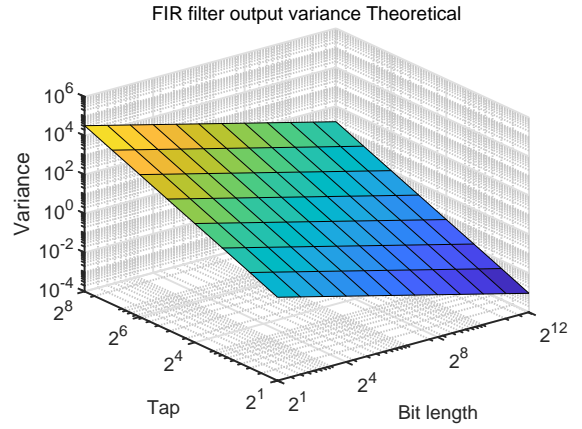


Fig. 15 フィルタ出力の分散の理論値

ここで,

$$E_{H_0 X(n)} - \sigma_{H_0 X(n)}^2 - E_{H_0 X(n)}^2 = 0 \quad (31)$$

$$E_S - \sigma_S^2 - E_S^2 = 0 \quad (32)$$

$$\begin{aligned} &\sigma_{H_0 X(n)}^2 E_S^2 + \sigma_S^2 E_{H_0 X(n)} + \sigma_{H_0 X(n)}^2 \sigma_S^2 \\ &= \frac{1}{16} (1 - h_0^2 x(n)^2) + \frac{1}{16} (h_0 x(n) + 1)^2 \\ &\quad + \frac{1}{16} (1 - h_0^2 x(n)^2) \\ &= -\frac{1}{16} h_0^2 x(n)^2 + \frac{1}{8} h_0 x(n) + \frac{3}{16} \end{aligned} \quad (33)$$

同様に,

$$E_{H_1 X(n-1)} - \sigma_{H_1 X(n-1)}^2 - E_{H_1 X(n-1)}^2 = 0 \quad (34)$$

$$(1 - E_S) - \sigma_S^2 - (1 - E_S)^2 = 0 \quad (35)$$

$$\begin{aligned} &\sigma_{H_1 X(n-1)}^2 E_S^2 + \sigma_S^2 E_{H_1 X(n-1)} + \sigma_{H_1 X(n-1)}^2 \sigma_S^2 \\ &= \frac{1}{16} (1 - h_1^2 x(n-1)^2) + \frac{1}{16} (h_1 x(n-1) + 1)^2 \end{aligned}$$



$$\begin{aligned}
& + \frac{1}{16}(1 - h_1^2 x(n-1)^2) \\
= & -\frac{1}{16}h_1^2 x(n-1)^2 + \frac{1}{8}h_1 x(n-1) + \frac{3}{16} \quad (36)
\end{aligned}$$

また,

$$\begin{aligned}
& -2E_{H_0 X(n)} E_{H_1 X(n-1)} \sigma_S^2 \\
= & -2 \times \frac{1}{2}(h_0 x(n) + 1) \times \frac{1}{2}(h_1 x(n-1) + 1) \times \frac{1}{4} \\
= & -\frac{1}{8}(h_0 x(n) + 1)(h_1 x(n-1) + 1) \quad (37)
\end{aligned}$$

以上より,

$$\begin{aligned}
\sigma_Y^2 & \\
= & -\frac{1}{16}h_0^2 x(n)^2 + \frac{1}{8}h_0 x(n) + \frac{3}{16} \\
& -\frac{1}{16}h_1^2 x(n-1)^2 + \frac{1}{8}h_1 x(n-1) + \frac{3}{16} \\
& -\frac{1}{8}(h_0 x(n) + 1)(h_1 x(n-1) + 1) \\
= & \frac{4 - (h_0 x(n) + h_1 x(n-1))^2}{16} \quad (38)
\end{aligned}$$

入力  $x(n)$  の時間平均をとる.

$$\begin{aligned}
\sigma_Y^2 & \\
= & \frac{1}{4} - \frac{1}{16} \{h_0^2 E[x(n)^2] \\
& + 2h_0 h_1 E[x(n)]E[x(n-1)] \\
& + h_1^2 E[x(n-1)^2]\} \quad (39)
\end{aligned}$$

ここで,  $E[x(n)] = E[x(n-1)] = 0$ ,  $E[x(n)^2] = E[x(n-1)^2] = V[x(n)] = \frac{1}{3}$  より,

$$\begin{aligned}
\sigma_Y^2 & = \frac{1}{4} - \frac{1}{16} \left( \frac{1}{3}h_0^2 + \frac{1}{3}h_1^2 \right) \\
& = \frac{1}{4} - \frac{1}{48}(h_0^2 + h_1^2) \quad (40)
\end{aligned}$$

となる. 式 40 に加算のスケール補正をかけて求めた分散の理論値 ( $\sigma_y^2$ ) を Fig.16 に示す. このグラフは Fig.12 と Fig.13 に概ね一致しており, 分散伝達関数のから FIR フィルタの出力に生じる分散が記述できたことが示された. 一方で, グラフの端の領域 ( $(h_0, h_1) = (-1, -1)$  など) においてシミュレーション結果と誤差が生じている. これについては, 今後の検討が必要である.

#### 5.4 タップ数・語長と分散の関係

次に, フィルタ出力部における S2B 変換直前の SN を  $Y$ ,  $X$  内のビット 1 の数を  $N_1(Y)$ ,  $X$  の語長

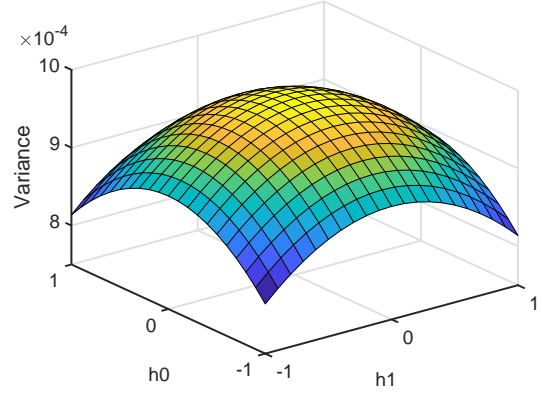


Fig. 16 式の解析によって求めたフィルタ出力の分散の理論値

を  $N_L$ , フィルタのタップ数を  $T$ , スケール補正後の出力を  $y$  のとする. S2B 変換とスケール補正後は以下の式で表すことができる.

$$x = \left( 2 \times \frac{N_1(Y)}{N_L} - 1 \right) \times T \quad (41)$$

これは確率変数  $N_1(Y)$  から  $y$  への変換であるから,  $y$  の期待値と  $N_1(Y)$  の期待値の関係は以下のようになる.

$$E_y = \frac{2T}{N_L} E_{N_1(Y)} - T \quad (42)$$

また,  $y$  の分散と  $N_1(Y)$  の分散の関係は以下のようになる.

$$\sigma_y^2 = \left( \frac{2T}{N_L} \right)^2 \sigma_{N_1(Y)}^2 \quad (43)$$

ここで,  $\sigma_{N_1(Y)}^2$  を, 分散伝達関数の定義に則った分散  $\sigma_Y^2$  に置き換える ( $N_w$  で正規化すると),

$$\sigma_y^2 = \frac{4T^2}{N_w} \sigma_Y^2 \quad (44)$$

となる. この式は, 出力の分散がフィルタのタップ数の二乗に比例し, 語長に反比例することを示している.

## 6. まとめと今後の課題

本稿では, ストカスティック演算によって実現された FIR フィルタの内部で発生する雑音について検討した. 分散伝達関数を用いることで, 2タップ FIR

フィルタで発生する雑音を解析し、式で記述できることを示した。今後は、さらに係数の多い場合について、出力の分散(式 40 の  $\sigma_y^2$ ) の式を導出したいと考えている。

## 参考文献

- 1) B.R. Gaines, “Stochastic computing systems,” *Advances in Information systems Science*, vol.2, pp.37–172, 1969.
- 2) A. Alaghi and J.P. Hayes, “Survey of stochastic computing,” *ACM Trans. Embedd. Comput. Sys.*, vol.12, no.92, pp.1–19, 2013.
- 3) B.D. Brown and H.C. Card, “Stochastic neural computation i: Computational elements,” *IEEE Trans. Comput.*, vol.50, no.9, pp.891–905, 2001.
- 4) H. Chen and J. Han, “Stochastic computational models for accurate reliability evaluation of logic circuits,” *Proc. GLSVLSI*, pp.61–66, 2010.
- 5) Y. Liu and K.K. Parhi, “Architectures for recursive digital filters using stochastic computing,” *IEEE Trans. Signal Process.*, vol.64, no.14, pp.3705–3718, 2016.
- 6) H. Ichihara, T. Sugino, S. Ishii, T. Iwagaki, and T. Inoue, “Compact and accurate digital filters based on stochastic computing,” *IEEE Trans. Emerging Topics in Comp.*, pp.1–12, 2016.
- 7) C. Ma, S. Zhong, and H. Dang, “Understanding variance propagation in stochastic computing systems,” *Proc. ICCD*, pp.213–218, 2012.