IIR Filters Using Stochastic Arithmetic

Naman Saraf, Kia Bazargan, David J. Lilja, and Marc D. Riedel
Department of Electrical and Computer Engineering
University of Minnesota, Twin Cities
Minneapolis, MN, USA
{saraf012, kia, lilja, mriedel} @umn.edu

Abstract— **We consider the design of IIR filters operating on oversampled sigma-delta modulated bit streams using stochastic arithmetic. Conventional digital filters process multi-bit data at the Nyquist rate using multi-bit multipliers and adders. High resolution ADCs based on the sigma-delta modulation generate random bits at an oversampled rate as intermediate data. We propose to filter the sigma-delta modulated bit streams directly and present first and second order low pass IIR filters based on the stochastic integrator. Experimental results show a significant reduction in hardware area by using stochastic filters.**

Keywords—Stochastic computing; Oversampling; Sigma-delta modulation; Stochastic integrator; IIR filters

I. INTRODUCTION

Filtering is a fundamental operation in signal processing to modify the spectral characteristics of signals, with numerous applications in image and speech processing. Due to the proliferation of VLSI technology, bulk of the filtering today is performed in the digital domain. Digital filtering outperforms analog signal processing by offering better noise immunity, repeatability and a superior tolerance to process variations.

Conventional digital filtering is based on *Nyquist*-rate data convertors and multi-bit digital filters as depicted in Fig.1. The analog input is converted to a multi-bit digital representation by a Nyquist rate analog-to-digital convertor (ADC), and processed by a multi-bit digital filter. The filtered digital signal is then converted back to the analog domain by a digital-to-analog convertor (DAC). The key attributes of a digital filtering architecture are the data processing rate and the resolution. Conventional digital filtering processes data at the Nyquist rate f_N, which is equal to twice the maximum frequency in the analog input signal (sampling theorem) [1] while the resolution is determined by the ADC resolution.

Despite the pervasiveness of conventional digital filtering, the architecture suffers from a number of limitations. The modest resolution of modern Nyquist rate ADCs limits the resolution of the architecture to 8-12 bits [2], [3]. In addition, the multi-bit digital filters require multi-bit multipliers and adders, leading to a large area and power dissipation.

A significant improvement over the conventional Nyquist-rate digital filtering is possible by using oversampled *sigma-delta modulator* (SDM) based data convertors [4]. An SDM operates at a processing rate much higher than the Nyquist rate while converting the analog input to a much lower resolution one bit representation. The SDM therefore trades amplitude resolution in favor of resolution in time.

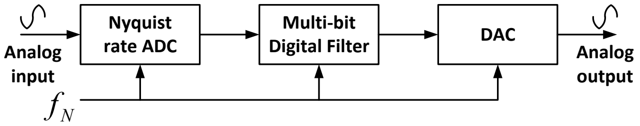

Fig.1 Conventional digital filtering using Nyquist rate ADCs/DACs

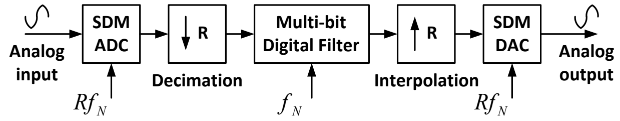

Fig.2 Conventional digital filtering based on SDM ADCs/DACs

SDM based ADCs offer a high resolution (upto 24 bits) at a fraction of the power dissipated by Nyquist rate ADCs [3]. Moreover, SDM based ADCs are built using fewer analog components leading to a lower implementation cost [3] and better integration with the standard CMOS technology.

The SDM bit stream at the *oversampled* rate Rf_N $(R \gg 1)$ is processed by a decimation filter to obtain a multi-bit digital representation of the analog input at the Nyquist rate. Therefore, Nyquist-rate data convertors in the conventional digital filtering architecture may be replaced by SDM based data convertors to improve the resolution. However, the resulting architecture (Fig.2) still requires multi-bit digital filters, and more importantly additional decimation and interpolation filters to interface the SDM at the oversampled rate.

We propose to filter the oversampled SDM bit streams directly to avoid the interface filters and simplify the digital filters to process bit stream data. The proposition of filtering analog signals encoded as SDM bit streams by digital filters constructed using one bit adders and multipliers promises major savings in area.

The idea of filtering oversampled SDM bit streams directly has been explored in literature. Numerous filter designs have demonstrated the feasibility of the architecture in Fig.3 such as the SDM based bit stream FIR filters [5-7], IIR filters [8-10] and LMS adaptive filters [11], [12]. Tutorial papers explaining the approach with potential applications have also appeared [13], [14].

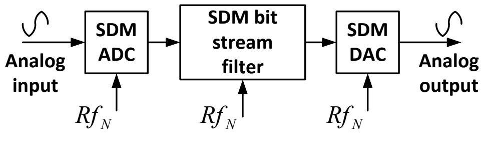

Fig.3 SDM based oversampled bit stream filtering

978-3-9815370-2-4/DATE14/©2014 EDAA/

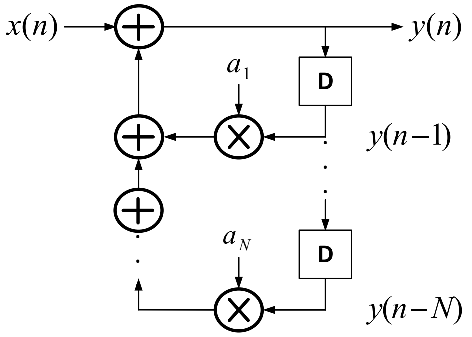

Fig.4 The conventional IIR filter structure. Existing SDM based bit stream IIR filters encode either the input signal $x(n)$ or the coefficients a_i in the bit stream format to simplify the multipliers.

Digital filters are classified into FIR or IIR filters based on the filter transfer function. IIR filters meet a given set of filter specifications using fewer memory and logic resources than FIR filters and are the focus of the current work. Existing SDM based bit stream IIR filters [8-10] make limited use of the oversampled bit streams, encoding either the input signal or the filter coefficients in the bit stream format, while representing the other in a conventional multi-bit format leading to simpler multipliers. Moreover, the number of SDMs required in the filters increases linearly with the filter order.

We propose an IIR filter structure based on the *stochastic integrator* [15] to process analog inputs encoded as oversampled SDM bit streams. Our approach differs from the conventional IIR filter structure (Fig.4) by avoiding explicit multipliers and adders. We develop an original z-transform analysis of the stochastic integrator and present a modified stochastic integrator with a gain parameter K. We discuss the design of first and second order low pass IIR filters based on the modified stochastic integrator and present frequency domain simulation results. We compare the hardware costs of our proposed filters to the conventional IIR filters and conclude with a discussion on the advantages and trade-offs of SDM based bit stream filtering. In the following sections, we assume that the oversampled SDM bit streams are available for processing and do not discuss the design of SDMs.

II. Stochastic Integrator

Consider an m-bit binary up/down counter C_N with N states labelled as $\{0,1, \dots N-1\}$. Let $count(n)$ denote the m-bit binary number stored in the counter at time n. The counter inputs up and $down$ update the counter state every clock cycle according to the rule in Table I. Let the counter inputs be driven by independent Bernoulli random bit streams such that,

$$Prob(up=1)=p_1(n),\ Prob(down=1)=p_2(n) \qquad (1)$$

TABLE I. Counter state update rule

$\boldsymbol{up}$	$\boldsymbol{down}$	$\boldsymbol{count(n+1)}$
1	0	$\boldsymbol{count(n)+1}$
0	1	$\boldsymbol{count(n)-1}$
0	0	$\boldsymbol{count(n)}$
1	1	$\boldsymbol{count(n)}$

The next counter state $count(n+1)$ is a random variable with a probability distribution function shown in Table II.

TABLE II. Probability distribution function of the counter state

$\boldsymbol{count(n+1)}$	$\boldsymbol{Probability}$
$\boldsymbol{count(n)+1}$	$\boldsymbol{p_1(n)(1-p_2(n))}$
$\boldsymbol{count(n)-1}$	$\boldsymbol{p_2(n)(1-p_1(n))}$
$\boldsymbol{count(n)}$	$\boldsymbol{1-p_1(n)-p_2(n)+2p_1(n)p_2(n)}$

The *expected* value of the next counter state is computed as,

$$\begin{aligned} E[count(n+1)] &= (count(n)+1)(p_1(n)-p_1(n)p_2(n)) \\ &+ (count(n)-1)(p_2(n)-p_1(n)p_2(n)) \\ &+ count(n)(1+2p_1(n)p_2(n)-p_1(n)-p_2(n)) \end{aligned} \qquad (2)$$

which simplifies to,

$$E[count(n+1)] = count(n)+p_1(n)-p_2(n) \qquad (3)$$

In addition, the counter state is compared with an m-bit uniformly distributed random integer using a comparator. The comparator output is logic 1 if the counter state exceeds the random integer, else logic 0. Let $P(n)$ denote the probability of the comparator output to be logic 1 when the counter state is $count(n)$. Then,

$$Prob(comparator\ output = 1\ at\ time\ n) \triangleq P(n)$$

$$\Rightarrow P(n) = \frac{count(n)}{N} \qquad (4)$$

The probability of the comparator output to be logic 1 in the next clock cycle, $P(n+1)$ is computed from $P(n)$ and the input probability values $p_1(n)$ and $p_2(n)$ using the law of total probability as,

$$\begin{aligned} P(n+1) &= \left(P(n)+\frac{1}{N}\right)(p_1(n)-p_1(n)p_2(n)) \\ &+ \left(P(n)-\frac{1}{N}\right)(p_2(n)-p_1(n)p_2(n)) \\ &+ P(n)(1-p_1(n)-p_2(n)+2p_1(n)p_2(n)) \end{aligned}$$

$$\Rightarrow P(n+1) = P(n)+\frac{p_1(n)}{N}-\frac{p_2(n)}{N} \qquad (5)$$

We notice that (3) and (5) differ only by a factor of $1/N$. In fact, a simpler method to arrive at (5) is to compute the expected value of the next counter state as a function of the present state and the input probability values, and dividing the expression by the number of states N. Intuitively, the output of the system is logic 1 with a probability that is equal to the expected value of the counter state scaled by $1/N$. We will use the observation later to derive equivalent expressions for the modified stochastic integrator.

We interpret the probability variables in (5) as real values in the interval $[-1,1]$ by applying the *bipolar* transformation,

$$\begin{aligned} y(n) &= 2P(n)-1 \\ x_i(n) &= 2p_i(n)-1, \qquad i=\{1,2\} \end{aligned} \qquad (6)$$

Equation (5) is now expressed as,

$$\boldsymbol{y(n+1) = y(n)+\frac{x_1(n)-x_2(n)}{N}} \qquad (7)$$

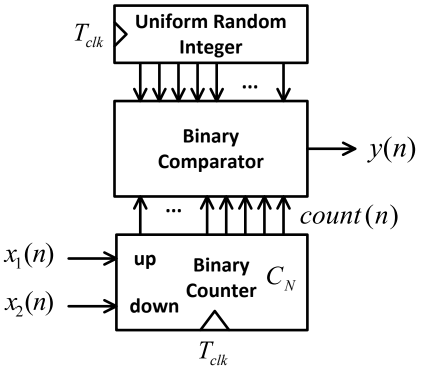

Fig.5 Stochastic integrator

We have thus constructed a dynamical system described by the recurrence relation in (7) with real valued inputs and outputs in the interval $[-1,1]$. We refer to the system as the *stochastic integrator* depicted in Fig.5.

To verify the analogy with a discrete-time integrator, we consider the frequency domain representation of (7) using the z-transform. Let the z-transform pairs of the variables be,

$$y(n) \leftrightarrow Y(z), x_1(n) \leftrightarrow X_1(z), x_2(n) \leftrightarrow X_2(z) \tag{8}$$

$$\Rightarrow zY(z) = Y(z) + \frac{X_1(z) - X_2(z)}{N} \tag{9}$$

Therefore, the transfer functions of the system become,

$$\left.\frac{Y(z)}{X_1(z)}\right|_{X_{2=0}} = \frac{1}{N(z-1)}, \qquad \left.\frac{Y(z)}{X_2(z)}\right|_{X_{1=0}} = \frac{-1}{N(z-1)} \tag{10}$$

The transfer function of a discrete-time integrator is given by,

$$H(z) = \frac{1}{z-1} \tag{11}$$

The system therefore performs discrete time integration of the input random bit streams $x_1(n)$ and $x_2(n)$ with a gain of $1/N$ to produce the output random bit stream $y(n)$. The input $x_1(n)$ is connected to the *non-inverting* input of the stochastic integrator while $x_2(n)$ is connected to the *inverting* input.

A stochastic integrator is characterized by the number of states N and the state update rule. Equation (7) describes a stochastic integrator where the state is updated every clock cycle and the next state takes on one of three possible values shown in Table I. Let us now consider an integrator with N states where the state is updated every K clock cycles. The integrator observes K $(K < N)$ random bits at each of the two inputs before transitioning to the next state. We assume that the input probability values do not vary significantly over the K clock cycles and remain equal to $p_1(n)$ and $p_2(n)$.

Let the K random bits at the two inputs be denoted by,

$$X_j^i, \quad j = \{1,2\}, \quad i = \{1,2, \ldots K\} \tag{12}$$

with expected values $\forall i$,

$$E\left[X_1^i\right] = p_1(n) \times 1 + \left(1 - p_1(n)\right) \times 0 = p_1(n)$$
$$E\left[X_2^i\right] = p_2(n) \times 1 + \left(1 - p_2(n)\right) \times 0 = p_2(n) \tag{13}$$

The integrator state is now updated according to the rule,

$$count(n+1) = count(n) + \sum_{i=1}^{K} X_1^i - \sum_{i=1}^{K} X_2^i \tag{14}$$

The expected value of the next state is therefore given by,

$$\begin{aligned} E[count(n+1)] &= E[count(n)] + E[\sum_{i=1}^{K} X_1^i] - E[\sum_{i=1}^{K} X_2^i] \\ &= count(n) + \sum_{i=1}^{K} E\left[X_1^i\right] - \sum_{i=1}^{K} E\left[X_2^i\right] \\ &= count(n) + Kp_1(n) - Kp_2(n) \end{aligned} \tag{15}$$

$$\Rightarrow P(n+1) = P(n) + \frac{K(p_1(n) - p_2(n))}{N} \tag{16}$$

Applying the bipolar transformation yields,

$$\boldsymbol{y(n+1) = y(n) + \frac{K(x_1(n) - x_2(n))}{N}} \tag{17}$$

Equation (17) describes a stochastic integrator with a gain of K/N that operates at a rate K times lower than the clock rate with transfer functions given by,

$$\left.\frac{Y(z)}{X_1(z)}\right|_{X_{2=0}} = \frac{K}{N(z-1)}, \qquad \left.\frac{Y(z)}{X_2(z)}\right|_{X_{1=0}} = \frac{-K}{N(z-1)} \tag{18}$$

A hardware realization of the parameterized stochastic integrator is shown in Fig.6. An up/down counter C_K with $2K+1$ states $\{-K, \ldots 0, \ldots K\}$, computes the difference in the number of logic 1 bits observed at the two integrator inputs over the K clock cycles. The signed binary count in C_K is added to the integrator state stored in C_N at the end of the K^{th} clock and the counter C_K is reset. Note that a stochastic integrator with $K = 1$ is identical to the structure in Fig.5. In the following sections, we will represent a stochastic integrator by the symbol shown in Fig.7 with the inverting input labeled as **"−"**.

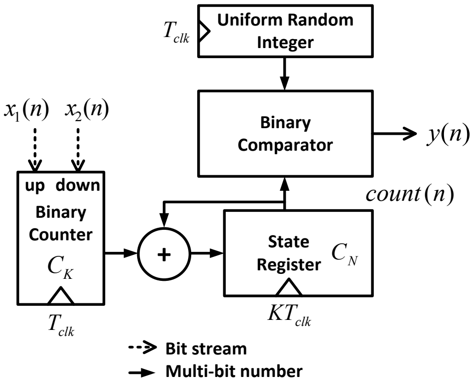

Fig.6 Modified stochastic integrator with parameter K

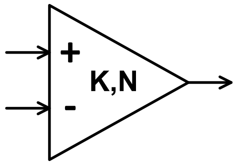

Fig.7 Stochastic integrator symbol with parameter K

III. Stochastic IIR Filters

We now discuss bit stream IIR filters based on the stochastic integrator. A general N^{th} order IIR filter is described by the difference equation,

$$y(n) = \sum_{i=1}^{N} a_i y(n-i) + \sum_{j=0}^{M} b_j x(n-j) \qquad (19)$$

with a frequency domain representation,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{j=0}^{M} b_j z^{-j}}{1 + \sum_{i=1}^{N} a_i z^{-i}} \qquad (20)$$

where $x(n)$ and $y(n)$are the filter input and output values at time n. The constant coefficients a_i and b_j determine the *zeros* and *poles* of the filter. The zeros and poles govern the filter frequency response and are obtained from the filter transfer function $H(z)$ as roots of the numerator and the denominator polynomial. Poles near $z = 1$ in the complex z-plane realize low pass digital filters. Moreover, every pole of a stable filter must lie inside the unit circle $|z| = 1$.

The input and output values for a conventional digital IIR filter are deterministic multi-bit numbers. However, in the context of our stochastic integrator based bit stream IIR filters, $x(n)$ and $y(n)$ represent the instantaneous probabilities of the input and output bit streams. The input bit streams to our stochastic IIR filters are always generated by SDM based ADCs. Autocorrelation studies have revealed that the bit stream generated by an SDM is a Bernoulli process [13]. Moreover, bit streams generated by distinct sources are uncorrelated.

Consider the stochastic integrator configuration depicted in Fig.8 where the input bit stream $x(n)$ is derived from an SDM based ADC. The recurrence relation of the filter is given by,

$$\begin{aligned} y(n+1) &= y(n) + \frac{K\big(x(n) - y(n)\big)}{N} \\ &= y(n)\left(1 - \frac{K}{N}\right) + \frac{K}{N}x(n) \end{aligned} \qquad (21)$$

Thus, the transfer function becomes,

$$\begin{aligned} zY(z) &= Y(z)\left(1 - \frac{K}{N}\right) + \frac{K}{N}X(z) \\ \Rightarrow H(z) &= \frac{Y(z)}{X(z)} = \frac{K}{Nz + K - N} \end{aligned} \qquad (22)$$

The only pole p_1 of the filter is located at $z = 1 - K/N$. Since,

$$0 < K < N \Longrightarrow 0 < p_1 = 1 - K/N < 1 \qquad (23)$$

and the system is a stable first order low pass IIR filter.

A second order low pass stochastic IIR filter is realized by the system in Fig.9 with the recurrence relations,

$$\begin{aligned} y_1(n+1) &= y_1(n) + \frac{K_1}{N_1}(x(n) - y(n)) \\ y(n+1) &= y(n) + \frac{K_2}{N_2}(y_1(n) - y(n)) \end{aligned} \qquad (24)$$

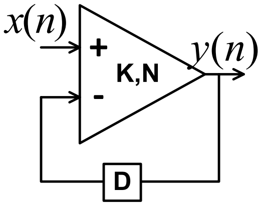

Fig.8 First order low pass stochastic IIR filter

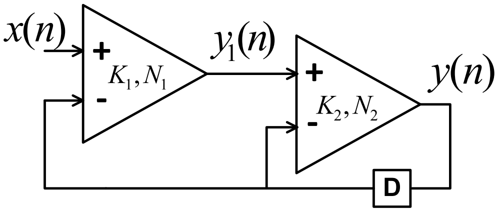

Fig.9 Second order low pass stochastic IIR filter

The transfer function of the system is obtained by solving the following pair of equations,

$$\begin{aligned} zY_1(z) &= Y_1(z) + \frac{K_1}{N_1}(X(z) - Y(z)) \\ zY(z) &= Y(z) + \frac{K_2}{N_2}(Y_1(z) - Y(z)) \end{aligned} \qquad (25)$$

which yields $H(z) =$

$$\frac{K_1 K_2}{N_1 N_2 z^2 + (N_1 K_2 - 2N_1 N_2)z + (N_1 N_2 - N_1 K_2 + K_1 K_2)} \qquad (26)$$

The poles of the second order stochastic IIR filter are given by,

$$p_1, p_2 = 1 - \frac{K_2}{2N_2} \pm \sqrt{\frac{N_1^2 K_2^2 - 4N_1 N_2 K_1 K_2}{4N_1^2 N_2^2}} \qquad (27)$$

while the zeros are located at $z = \infty$.

The second order IIR filter may possess two real valued poles or a pair of complex conjugate poles depending on the stochastic integrator parameters. Second order systems, thus, exhibit a broader range of behavior than first order systems. Since $K_2 < N_2$,

$$0 \le 1 - \frac{K_2}{2N_2} \le 1 \qquad (28)$$

the system is a stable second order low pass IIR filter.

IV. Experimental Results

In this section we present experimental results on stochastic filters based on the first and second order IIR filter structures described in the previous section. Table III lists the stochastic integrator parameters for the filters used in our experiments. The parameter values were selected to demonstrate a diverse set of filter responses. The sampling frequency f_s and the oversampling ratio R were set at $f_s = 1$ and $R = 256$. Therefore, the highest input frequency f_{in} processed by the stochastic IIR filters is given by,

$$f_{in} = \frac{f_s}{2R} = \frac{1}{512} \Longrightarrow f_N = 2f_{in} = \frac{1}{256} \qquad (29)$$

TABLE III. Stochastic integrator parameters used in the experiments

Filter	*Order*	K_1	N_1	K_2	N_2
LPF_1A	1	1	64	-	-
LPF_1B	1	1	256	-	-
LPF_2A	2	1	64	1	256
LPF_2B	2	1	256	1	64

We measure the magnitude responses of the stochastic filters in Table III by applying a sinusoidal test input of frequency f to an SDM based ADC and filtering the resulting bit stream. The multi-bit stochastic integrator state corresponding to the output bit stream is the filter output. The filter response at frequency f is determined from the FFT of the filter output. The test input frequency is swept from DC to f_N to generate the curves shown in Fig.10 and Fig.11. Each plot shows the simulated magnitude response of the stochastic IIR filter with the expected response of a conventional IIR filter.

Fig.10 depicts the magnitude responses of the first order stochastic IIR filters. Recall that the pole of a first order IIR filter is located at $z = 1 - K/N$. The pole of LPF_1B is much closer to $z = 1$ than the pole of LPF_1A, leading to a sharper roll-off and lower bandwidth. The magnitude responses of the second order stochastic IIR filters are shown in Fig.11. Filter LPF_2B has two simple real poles while LPF_2A has a pair of complex conjugate poles, which is evident from the overshoot in the magnitude response.

The output resolution of a stochastic filter is determined by the number of integrator states N. A conventional IIR filter with an r-bit resolution is equivalent to a stochastic IIR filter having an integrator with $N = 2^r$ states. However, the pole location of a stochastic filter is strongly affected by the value of N, unlike conventional filters where the resolution has no impact on the pole location. This requires a careful selection of values for K and N while designing stochastic IIR filters to simultaneously meet the requirements on the output resolution and the pole locations.

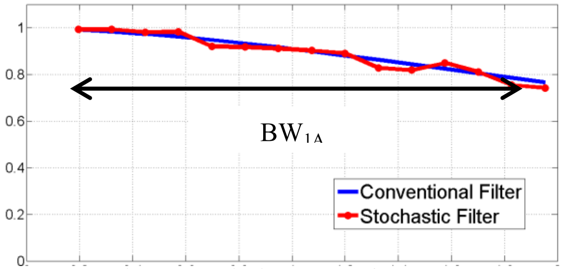

Fig.10a Magnitude response of LPF_1A

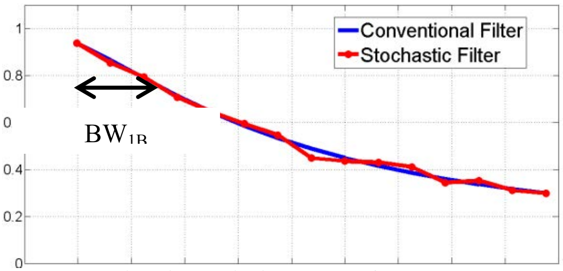

Fig.10b Magnitude response of LPF_1B

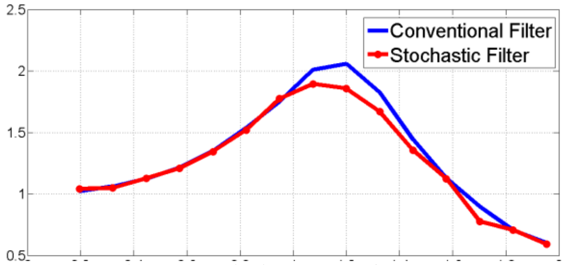

Fig.11a Magnitude response of LPF_2A

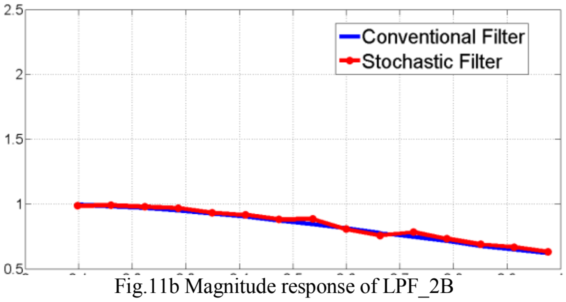

Fig.11b Magnitude response of LPF_2B

We compare the hardware costs of the stochastic IIR filters with the conventional IIR filters having an equivalent output resolution for a 45nm CMOS technology. The results in Table IV clearly indicate that stochastic filters occupy a smaller area than the conventional filters. The random number generator in the stochastic integrators was constructed using an LFSR.

TABLE IV. Comparison of the hardware area (library units)

Filter	*Conventional IIR Filter*	*Stochastic IIR Filter*
LPF_1A	554	118.4
LPF_1B	897.2	156.1
LPF_2A	1067.4 ***	270.0
LPF_2B	1248.3 ***	270.0

*** considering different resolution in the two stages to make a comparison

V. A Design Example

First order IIR filters are of considerable interest in audio signal processing due to their low implementation cost. We consider the design of first order low pass IIR filters for audio signals known as *treble-cut* filters. The specifications for an example treble-cut audio filter are summarized in Table V.

TABLE V. Specifications for a treble-cut audio filter

Nyquist frequency $\boldsymbol{f_N}$ (Hz)	**44100**
Oversampling ratio $\boldsymbol{R}$	**64**
Sampling frequency $\boldsymbol{f_s} = \boldsymbol{Rf_N}$ (Hz)	**2822400**
Computation resolution (bits)	**9**
Analog cut-off frequency $\boldsymbol{f_c}$ (Hz)	**800**

The design of a digital first order low pass IIR filter begins with an analog first order low pass transfer function given by,

$$H(s) = \frac{\omega_c}{s + \omega_c} \qquad (30)$$

where $\omega_c = 2\pi f_c$ is the analog cut-off frequency in rad/s.

The analog transfer function pole at $s = -\omega_c$ is mapped to a z-plane pole using the impulse invariance method [16] as,

$$z = e^{-\omega_c/f_s} = e^{-2\pi\times 800/2822400} = 0.998 \qquad (31)$$

Based on the location of the z-plane pole in (31) and the filter specifications, we select $K = 1$ and $N = 512$ as the parameters for a first order stochastic treble-cut audio filter. We test the stochastic filter by applying a mixture of sinusoidal signals at frequencies $f_{low} = 100Hz$, $f_{med} = 800Hz$ and $f_{high} = 2500Hz$. The input and output signal spectrums in Fig.12 and Fig.13 verify the operation of the stochastic treble-cut audio filter. The signal at frequency f_{med} is at the filter cut-off frequency and is attenuated by 3dB. The signal at f_{high} is removed from the output while the signal at f_{low} is unaffected. Table VI compares the area of the first order stochastic treble-cut audio filter with a conventional implementation. We observe that the stochastic implementation requires a much smaller area than the conventional IIR filter.

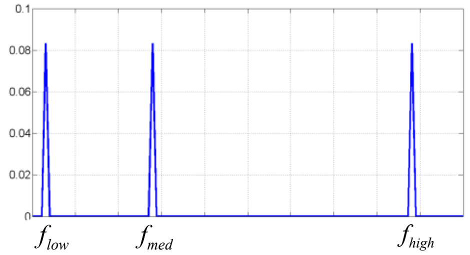

Fig.12 Input signal spectrum

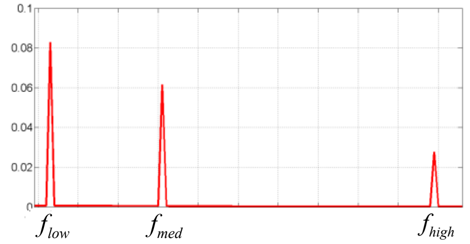

Fig.13 Filtered output signal spectrum

TABLE VI. Comparison of the hardware area for the audio filter (library units)

Filter	***Conventional IIR Filter***	***Stochastic IIR Filter***
Treble-cut filter	918.2	175.2

VI. Conclusions And Future Work

We have presented an approach to filter analog signals encoded as SDM bit streams by IIR filters constructed using the stochastic integrator. The proposed architecture simplifies the arithmetic operations in the conventional IIR filters and offers significant savings in area. However, unlike conventional digital filters, stochastic filters operate on probabilistic data and are susceptible to random noise.

Future work on stochastic IIR filters involves extending the low pass structures to design band pass and high pass filters based on the stochastic integrator and deriving additional performance metrics such as the power dissipation. Developing an accurate noise model to predict the SNR at the output of the stochastic filters is a challenging task. The final validation of the approach would require designing stochastic filters to solve complex filtering problems with an acceptable performance.

VII. Acknowledgment

This work was funded in part by the National Science Foundation under Grant CCF-1241987.

References

[1] C.E Shannon, “Communication in the presence of noise,” *Proceedings of the IEEE*, vol.86, no.2, pp.447-457, 1998.

[2] R.H Walden, “Analog-to-digital converter survey and analysis,” *IEEE Journal on Selected Areas in Communications*, vol.17, no.4, pp.539-550, 1999.

[3] Bin Le, T.W Rondeau, J.H Reed, C.W Bostian, “Analog-to-digital converters,” *IEEE Signal Processing Magazine*, vol.22, no.6, pp.69-77, 2005.

[4] J.M de la Rosa, “Sigma-Delta Modulators: Tutorial overview, design guide, and state-of-the-art survey,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol.58, no.1, pp.1-21, 2011.

[5] P.W Wong, R.M Gray, “FIR filters with sigma-delta modulation encoding,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.38, no.6, pp.979-990, 1990.

[6] P.W Wong, “Fully sigma-delta modulation encoded FIR filters,” *IEEE Transactions on Signal Processing*, vol.40, no.6, pp.1605-1610, 1992.

[7] S.M Kershaw, S. Summerfield, M.B Sandler, M. Anderson, “Realisation and implementation of a sigma-delta bitstream FIR filter,” *IEE Proceedings on Circuits, Devices and Systems*, vol.143, no.5, pp.267-273, 1996.

[8] D.A Johns, D.M Lewis, “Design and analysis of delta-sigma based IIR filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol.40, no.4, pp.233-240, 1999.

[9] D.A Johns, D.M Lewis, D Cherepacha, “Highly selective ‘analog’ filters using ΔΣ based IIR filtering,” *IEEE International Symposium on Circuits and Systems*, pp.1302-1305, vol.2, 1993.

[10] B.R Owen, D.A Johns, “A single-column structure for delta-sigma based IIR filters,” *IEEE International Symposium on Circuits and Systems*, vol.5, pp.2413-416, 1992.

[11] E Pfann, R.W Stewart, M.W Hoffman, “Oversampled sigma-delta LMS adaptive FIR filters,” *IEE Proceedings on Vision, Image and Signal Processing*, vol.147, no.5, pp.385-392, 2000.

[12] Q Huang, G.S Moschytz, “Analog multiplierless LMS adaptive FIR filter structures,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*,vol.40, no.12, pp.790-794, 1993.

[13] F Maloberti, “Non-conventional signal processing by the use of sigma delta technique: a tutorial introduction,” *IEEE International Symposium on Circuits and Systems*, vol.6,pp.2645-2648 vol.6, 1992.

[14] V da Fonte Dias, “Sigma-delta signal processing," *IEEE International Symposium on Circuits and Systems*, vol.5, pp.421-424, 1994.

[15] B.R Gaines, “Stochastic Computing Systems,” *Advances in Information Systems Science*, J.F.Tou, vol.2, chapter 2, pp.37-172, New York: Plenum, 1969.

[16] J.G Proakis, “Digital Signal Processing,” 4th Edition.