IMPULSES AND STOCHASTIC ARITHMETIC FOR SIGNAL PROCESSING

John F. Keane and Les E. Atlas
Department of Electrical Engineering, University of Washington, Box 352500, Seattle, Washington 98195-2500, USA

ABSTRACT

The work described in this paper explores the use of Poisson point processes and stochastic arithmetic to perform signal processing functions. Our work is inspired by the asynchrony and fault tolerance of biological neural systems. The essence of our approach is to code the input signal as the rate parameter of a Poisson point process, perform stochastic computing operations on the signal in the arrival or "pulse" domain, and decode the output signal by estimating the rate of the resulting process. An analysis of the Poisson pulse frequency modulation encoding error is performed. Asynchronous, stochastic computing operations are applied to the impulse stream and analyzed. A special finite impulse response (FIR) filtering scheme is proposed that preserves the Poisson properties and allows filters to be cascaded without compromising the ideal signal statistics.

1. INTRODUCTION

The term "neurocomputation" has typically meant the collective behavior of a group of neuron-like processors, which are interconnected via conventional deterministic discrete-time encoding of levels. However, biological neurons communicate via impulses, where the occurrence of these impulses and the implicit computation at neural junctions is often modeled as stochastic. Biologists and engineers have argued that biological neural systems' fault tolerance, asynchrony, and wide dynamic range is a result of parallelism and pervasive stochastic computation and representation.

This paper describes a discrete signal processing scheme that embraces the notion that stochastic behavior is pervasive, and investigates the implications of a fully random signal processing methodology. The essence of the approach is to code the input signal in the rate parameter of a Poisson point process, perform stochastic computing operations on the signal in the arrival (or, "pulse") domain, and decode the output signal by estimating the rate of the resulting process. Our direct goal is not neural modeling, but is instead to provide a new signal processing architecture. As we will show, our approach provides strong possibilities for asynchronous operation, advantages for fault-tolerance, and other advantages over conventional signal processing architectures.

Poisson arrivals [1][2] and regular pulse streams [3]-[6] have been previously proposed for use in neural networks as the inter-neuron communication format and computational domain. Both periodic and random pulse streams, where the pulse width or duty cycle carries information, have also been used for signal processing [7]-[12]. The approach in this paper uses random impulse-like arrivals to perform signal processing functions. The impulse amplitudes and very short widths contain no information. The contributions in this paper are the use of Poisson arrivals, inspired by neural systems, for the specific task of signal processing; the observation that signal processing operations can be cascaded without corrupting the Poisson properties; and an FIR filter scheme which retains the Poisson properties. As we will show, implementations of this approach obtain simplicity, asynchrony, modularity, and robustness at the expense of coding efficiency and bandwidth.

2. POISSON ENCODING AND DECODING

The foundation of our proposed method is to use a time-varying Poisson arrival process to represent the signal $x(t)$. The value of $x(t)$ sets the rate of a Poisson process that produces arrivals (signed impulse functions). Signal processing operations may be performed on the impulses, described in Section 5. Decoding the impulse signal consists of estimating the rate parameter by counting the arrivals in an interval and dividing by the interval length.

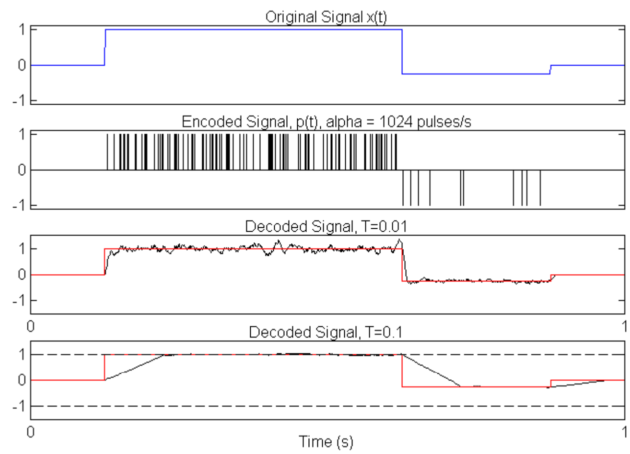

Figure 1. Poisson encoding and decoding. Panels 3 and 4 show the trade-off between bandwidth and SNR. In panel 2, p(t) has been decimated to make individual pulses visible.

The signal to be encoded is a real-valued function, $x(t)$, with an amplitude limit of $\max|x(t)| \le A_{\max}$. Since $x(t)$ can be negative and the rate of a Poisson process cannot, the sign is encoded separately from the magnitude. The signal $x(t)$ is encoded by driving a time-varying Poisson arrival process, P, with a rate parameter

$$\lambda(t) = \alpha \cdot |x(t)| , \quad (1)$$

and marking each arrival with either a positive or negative sign, $I(t) = sign(x(t))$. This variant of pulse frequency modulation produces pulses at random rather than at a fixed interval. Scaling the rate with a constant, α, to set $\lambda_{\max}$, will be shown later to impact the bandwidth and signal-to-noise ratio (SNR). The number of arrivals in an interval, T, assuming $x(t)$ is constant over T, follows a Poisson distribution,

$$P[k \text{ arrivals in interval } T] = (\lambda(t) \cdot T)^k e^{-\lambda(t) \cdot T} / k! . \quad (2)$$

The Poisson nature of the arrivals implies several properties. The arrival times within an interval are conditionally independent of one another, given the number of arrivals in the interval, and conditionally independent of the arrival times outside the interval, given the number of arrivals in the other intervals. The arrival times within an interval are uniformly distributed, given the number of arrivals in the interval [13].

The cumulative number of "positive" and "negative" arrivals up to time t are $N^{+}(t)$ and $N^{-}(t)$, and the net number of arrivals up to time t is $N(t) = N^{+}(t) - N^{-}(t)$. In any interval, T, where $x(t)$ remains constant, $E[N(t) - N(t-T)] = I(t) \cdot \lambda(t) \cdot T$. A realization of the process, P, is shown in Figure 1, where $P(t) = dN(t)/dt$. The decoded signal is

$$\hat{x}(t) = (N(t) - N(t-T))/(T\alpha)\,. \qquad (3)$$

3. SIGNAL-TO-NOISE RATIO AND BANDWIDTH

The impulse rate, bandwidth and SNR are interrelated. The maximum impulse rate, $\lambda_{\max}$, is set by the encoder, but the bandwidth versus SNR trade-off is determined by the decoder. The spectrum of the encoding noise is not bandlimited, even when $x(t)$ is constant. If $x(t)$ is bandlimited, then low-pass filtering can improve the SNR. Choosing a time window, T, for estimating $\lambda(t)$ is equivalent to choosing a low pass filter with $h(t) = rect(t/T)$.

The ensemble average error due to encoding with a Poisson process is zero, since the expected number of arrivals in an interval is exactly the Poisson rate. The noise power over the ensemble is a function of the rate,

$$\sigma_{noise}^2(t) = \lambda(t) \cdot T\,. \qquad (4)$$

Since $\lambda(t)$ is proportional to $x(t)$, the expected encoding noise at any point in time is a function of the signal level. It should be noted that this behavior is much different than conventional PCM computer representations where encoding noise (quantization) is a fixed function of word size.

The highest SNR for a sinusoid using this Poisson encoding scheme occurs when $x(t) = A_{\max} \cdot \sin(\omega \cdot t)$, a full-scale sinusoid. Then, $\lambda(t) = \lambda_{\max} \cdot |\sin(\omega \cdot t)|$ and

$$SNR = \frac{\sigma_{\lambda T}^2}{\sigma_{noise}^2} = \frac{(\lambda_{\max} \cdot T)^2/2}{E[\lambda(t) \cdot T]}\,. \qquad (5)$$

Letting $A_{\max} = 1$, the amplitude probability density of $x(t) = |\sin(\omega \cdot t)|$ over one cycle is

$$f(x) = \frac{2}{\pi \cdot \sqrt{1-x^2}} \qquad (6)$$

where $x \in [0,1]$. The expected value for this distribution is $E[x(t)] = 2/\pi$, $E[\lambda(t) \cdot T] = E[\lambda_{\max} \cdot x(t) \cdot T] = 2 \cdot \lambda_{\max} \cdot T/\pi$, and the resultant SNR comes from substituting these values back into equation (5).

$$SNR_{\lambda T} = \lambda_{\max} \cdot T \cdot \pi/4\,. \qquad (7)$$

If the number of arrivals (up to $\lambda_{\max} \cdot T$) were expressed as a binary number, it would require $\lceil \log_2(\lambda_{\max} \cdot T) \rceil$ bits. One additional bit is required to represent the sign of x, which makes the effective number of bits $B = \lceil \log_2(\lambda_{\max} \cdot T) \rceil + 1$. In decibels (dB), the SNR is therefore

$$SNR_{\lambda T} = 3.01 \cdot B - 1.05\,(dB)\,. \qquad (8)$$

For conventional PCM, $SNR = 6 \cdot B - 1.77\,(dB)$ [14]. The Poisson encoding scheme has a lower slope because the noise increases with the signal level, whereas the PCM encoding noise is essentially constant. Conversely, the Poisson encoding noise diminishes as the signal level falls, and its SNR falls at half the rate that PCM falls. Therefore, a Poisson encoding scheme uses its available dynamic range more effectively than conventional PCM. For example, if the average amplitude of a signal fails to use the most significant 3 bits of PCM encoding, the SNR of PCM encoding drops by 18 dB. Under a Poisson encoding system it is only reduced by 9 dB.

The Poisson encoding noise is a function of signal level, which can be observed in Figure 1. The noise level when $x(t) = 1$ is visibly higher than when $x(t) = -1/4$. Figure 2 illustrates the desirable perceptual effect of this tracking of encoding noise with image signal levels [1].

Figure 2. Poisson encoded images at different peak arrival rates $\lambda_{\max}$. Black is 0, and white is full scale. It should be noted that even with good quality reproduction the bottom two encoded images are virtually indistinguishable from the original top left image.

4. OPERATIONS ON A POISSON PROCESS

Addition, multiplication, and negation have simple structures in the Poisson domain. As shown in Figure 3, addition is performed by interleaving the impulse trains corresponding to the addends. The sum is also a Poisson process. Since the probability of a collision between Poisson arrivals is zero, the rate corresponds to the sum of the two constituent rates. We assume that implementations of this scheme use pulses with a short enough duration to make the non-zero probability of a collision negligible. Positive and negative channels are interleaved separately.

[1] Additional examples may be found at http://www.ee.washington.edu/research/isdl/papers/keane-01-icassp/

Individual impulses of opposite signs do not “cancel” one another, except when they are summed in the decoder. Negation requires re-marking the arrivals with the opposite sign.

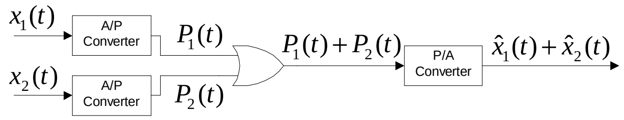

Figure 3. Addition in the Poisson domain. The A/P and P/A converters are analog-to-pulse and pulse-to-analog converters, respectively. The OR gate passes impulses of either polarity.

Multiplication is different from addition and negation in that it requires the creation or destruction of impulses. For the resulting process to retain the Poisson properties, the new or destroyed arrivals must also be Poisson processes. When $x(t)$ is multiplied by a fractional scalar $0 \le a \le 1$, multiplication can be accomplished by “thinning” $P(t)$, as shown in Figure 4 and Figure 5. Thinning is the random removal of each impulse by a Bernouilli trial with a retention probability of a and a removal probability of $1-a$ [13]. This form of multiplication introduces no SNR degradation.

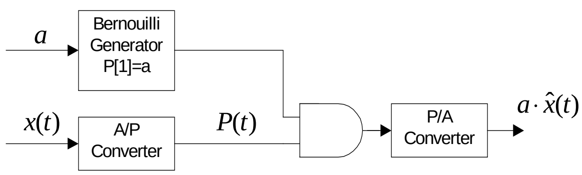

Figure 4. Multiplication in the Poisson domain. The AND gate, in this case, gates impulses of either polarity based on the Bernouilli generator input.

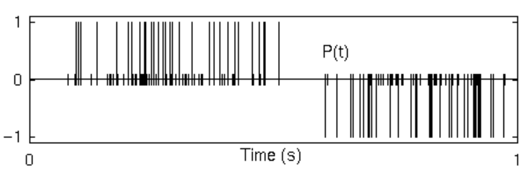

Figure 5. Multiplication in the Poisson domain. The original arrivals are shown as tick marks, and the ones remaining after thinning are unit spikes.

Multiplication by a larger scalar $a > 1$ requires the creation of Poisson impulses at the rate $(a-1)\cdot\lambda(t)$. Since the true $\lambda(t)$ is, in practice, unknown, the observed estimate would instead be encoded, introducing SNR degradation. We conclude that limiting scalar operations to $0 \le a \le 1$ is more practical.

Delay in the pulse domain corresponds to delay in the input signal’s time domain.

None of these operations (addition, negation, multiplication delay) affects the SNR of the resulting signal. The same SNR output would be obtained whether the operations occurred before or after the Poisson encoding process. One exception to this claim is the addition of signals with opposite signs. In this case, the output signal (mean) is reduced, but the noise components (variances) add.

5. FILTERING

Finite (FIR) and infinite impulse response (IIR) filtering can be implemented using the operations listed above. In addition to introducing SNR issues, filtering will also impact the statistical properties of the output signal. Before filtering, the arrivals in $P(t)$ occur independently at the applied rate. Filtering inherently creates correlation between the number of pulses in an interval and the pulse count in neighboring intervals. But, passing $P(t)$ through an IIR filter or a traditional FIR filter using the operations described in Section 4 would also introduce correlation in the arrival of pulses, at integer multiples of the delay time. For example, a stochastic filter that uses a traditional direct form FIR filter topology (Figure 6, left) will deliver, with probability $b_i \cdot b_j$, the same impulse to the output stream after the i^{th} delay and after the j^{th} delay, causing two arrivals to be spaced exactly (*j-i*) delays apart. This causes a statistical dependence between impulses. By having such a correlation between pairs of impulses, the independent increment property of the Poisson process described in Section 2 is violated.

However, if pulses are not re-used at successive taps, the Poisson properties can be preserved. One way to avoid this spike in the autocorrelation of the filter output is to use a specialized FIR filter structure that does not re-introduce arrivals at multiple lags. We call this topology a “successive removal” FIR filter (Figure 6, right). It makes the probability of an arrival in some arbitrarily small interval Δt independent of other arrivals, given the pulse rate at time t.

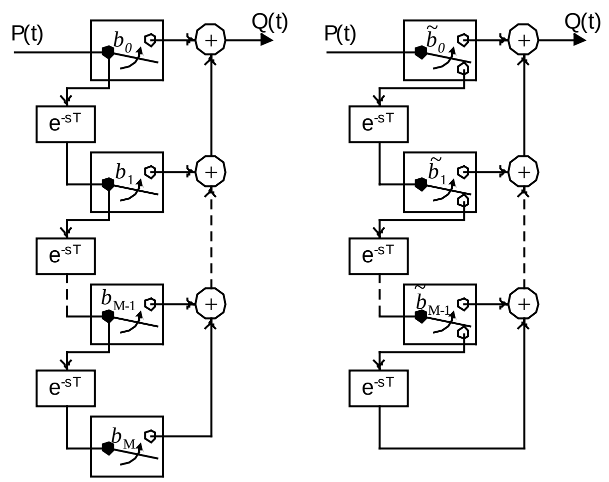

Figure 6. Stochastic FIR filter topologies: Direct form (left) and successive removal (right).

At each node of the successive removal filter, the input signal is stochastically multiplied by its appropriate weight. However, none of the pulses that are delivered to the output are also passed down the delay line. The multiplication operation splits the incoming impulse stream into two paths: impulses that go immediately to the output, and impulses that are put to the next delay element. To implement a filter $Q(t) = P(t)\cdot(b_0 + b_1 e^{-sT} + \cdots + b_M e^{-sMT})$, new coefficients must be calculated for the structure in Figure 6. The coefficients must be normalized, so that

$$\sum |b_i| = 1 . \quad (9)$$

To accommodate the structure of Figure 6, set $\tilde{b}_1$ such that $b_1 = (1 - b_0) \cdot \tilde{b}_1$, or $\tilde{b}_1 = b_1 / (1 - b_0)$. Likewise, the higher order coefficients are calculated by

$$\tilde{b}_m = \frac{b_m}{(1 - b_0) \cdot (1 - \tilde{b}_1) \cdots (1 - \tilde{b}_{m-1})} . \quad (10)$$

The response of any filter implemented with random operations on the impulse stream will be random. Essentially, the performance is equivalent to the response of a filter with noisy, time-varying coefficients. As a result, the filter, which has the advantages mentioned previously, has a degraded response (see Figure 7).

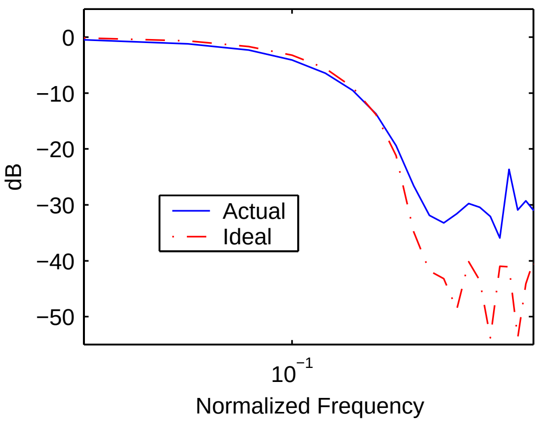

Figure 7. Successive removal FIR filter, mean performance of the magnitude response (M=10).

6. CONCLUSIONS

Stochastic computing can be used with Poisson processes to perform signal processing functions. Encoding a signal into a Poisson process brings the benefits of pulse-based computing: calculations are asynchronous and much easier to implement than standard arithmetic. This approach is robust because every pulse is like a least significant bit and isolated errors are predictably insignificant. Moreover, when signal amplitudes fall below full-scale, the SNR decreases less than it would in a conventional PCM binary encoding scheme. If pulse arrivals correspond to energy consumed, then the total energy scales with signal level.

By judicious selection of the topology, FIR filtering can be performed that preserves the Poisson nature of the encoded signals. This allows filters to be cascaded without losing the required independence among impulse arrivals. These filters, given appropriate circuit technology, could be low power, dense, and robust to localized errors.

Acknowledgement

The authors would like to thank Prof. Chris Diorio of the University of Washington Department of Computer Science and Engineering for inspiring discussions throughout this work.

The suggestions of A. Michael Dougherty and Bradford W. Gillespie of the University of Washington Department of Electrical Engineering for the improvement of the manuscript are highly appreciated.

7. REFERENCES

[1] A.S. French and A.V. Holden, "Alias-free sampling of neuronal spike trains," *Kybernetik*, **8**(5), pp. 165-171, May 1971

[2] H. Card, "Doubly Stochastic Poisson Processes in Artificial Neural Learning," *IEEE Trans. On Neural Networks*, **9**(1), pp. 230-231, January 1998

[3] A. Murray and A.V.W. Smith, "Asynchronous VLSI Neural Networks Using Pulse-Stream Arithmetic," *IEEE Journal of Solid-State Circuits*, **23**(3), pp. 688-697, June 1988.

[4] A.F. Murray, A. Hamilton, H.M. Reekie, and L. Tarassenko, "Pulse-stream arithmetic in programmable neural networks," *IEEE International Symposium on Circuits and Systems*, **2**, pp. 1210-1212, 1989.

[5] A.F.Murray, "Pulse Arithmetic in VLSI Neural Networks," *IEEE Micro*, pp. 64-74, December, 1989.

[6] A. Murray, "Pulse Techniques in Neural VLSI: A Review," *Proc. of the 1992 IEEE International Conf. On Circuits and Systems*, **5**, pp. 2204-2207, 1992.

[7] C.L. Janer, J.M. Quero, J.G. Ortega, and L.G. Franquelo, "Fully Parallel Stochastic Computation Architecture," *IEEE Trans. On Signal Processing*, **44**(8), pp 2110-2117, August 1996.

[8] A. Astaras, R. Dalzell, A. Murray, and M. Reekie, "Pulse-based circuits and methods for probabilistic neural computation," *Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, pp. 96-102, 1999.

[9] J.M. Quero, S.L Toral, J.G. Ortega, and L.G. Franquelo, "Continuous Time Filter Design Using Stochastic Logic," *42nd Midwest Symposium on Circuits and Systems*, **1**, pp. 113-116, 2000.

[10] S.L. Toral, J.M. Quero, and L.G. Franquelo, "Stochastic Pulse Coded Arithmetic," *IEEE International Symposium On Circuits and Systems*, **1**, pp. 599-602, August 1999.

[11] A. Hamilton and K. Papathanasiou, "Reconfigurable analogue systems: the pulse-based approach," *IEE Proceedings on Computers and Digital Techniques*, **147**(3), pp. 203–207, May 2000.

[12] M. Nagata and A. Iwata, "A PWM Signal Processing Core Circuit Based on a Switched Current Integration Technique," *IEEE J. Solid State Circuits*, **33**(1), pp. 53-60, January 1998.

[13] D. R. Cox and P.A.W. Lewis, *The Statistical Analysis of Series of Events*, John Wiley & Sons, Inc., New York, 1966.

[14] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1992.