# Stochastic Resonance and Finite Resolution in a Network of Leaky Integrate-and-Fire Neurons

Nhamo Mtetwa, Leslie S. Smith and Amir Hussain

Department of Computing Science and Mathematics, University of Stirling, Stirling, FK9 4LA, Scotland

**Abstract.** This paper discusses the effect of stochastic resonance in a network of leaky integrate-and-fire (LIF) neurons and investigates its realisation on a Field Programmable Gate Array (FPGA). We report in this study that stochastic resonance which is mainly associated with floating point implementations is possible in both a single LIF neuron and a network of LIF neurons implemented on lower resolution integer based digital hardware. We also report that such a network can improve the signal-to-noise ratio (SNR) of the output over a single LIF neuron.

## 1 Introduction

Stochastic Resonance (SR) occurs when a bistable nonlinear system is driven by a weak periodic signal, and provision of additional noise improves the system's detection of the periodic signal (see [7] for a review). Here "weak" means that the input signal is so small that when applied alone it is undetected [11]. Simulation of SR in "continuous" systems using floating point based models is well established [3], [13] and [12]. Such models are fairly accurate but do not always result in real-time performance. This work aims to show that SR still occurs with lower resolution integer based representations which allow real-time performance on digital hardware such as FPGAs. The FPGA allows us to vary word length for activation and investigate the effect of this on SR. FPGAs are reconfigurable programmable logic devices [4]. The FPGA used here is a Xilinx Virtex XCV1000 [1]. We investigate the effect of varying the resolution of the numbers inside the simulation on SR: we do this by examining SR in a 64 bit floating point model in Java, and in an integer model where the integer length is varied downwards from 16 bits on the FPGA. The SR effect is assessed by applying a subthreshold periodic signal plus noise to the system, and examining its output SNR. For both models we calculate the SNR at different input noise amplitudes. For the integer model we calculate the SNR values at different resolutions. This work extends [10] to a network (see figure 1). Our results show that integer lengths of at least 10 bits produce usable results, and that the network produces a performance improvement for six input neurons at integer length of at least 11 bits.

## 2 The Model

Each neuron is modelled by the LIF neuron model (equation (1)) which idealises the neuron as a capacitor $C$, in parallel with a resistor of resistance $R$. The

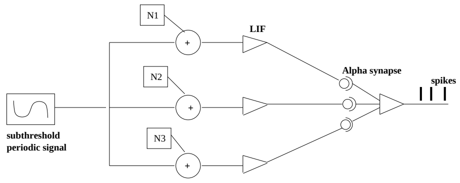



**Fig. 1.** Network model. Layer 1: identical, parallel input LIF neurons receive the same 20Hz subthreshold sinusoidal input plus independent uncorrelated Gaussian white noise (N1, N2, N3). Output: Same LIF neuron model receives spiking input via an Alpha synapse.

effective current $I(t)$ may hyperpolarise or depolarise the membrane.

$$C\frac{dV}{dt} = I(t) - \frac{V(t)}{R} \tag{1}$$

For the input neurons, $I(t)$ is composed of a periodic subthreshold sinusoidal part and a stochastic component made of Gaussian white noise [2]. For the output neuron, $I(t)$ is the input layer's spikes passed through an Alpha synapse [8]. A network model similar to ours using a depressing synapse with 4 input layer neurons was studied in [14]. They report that a depressing synapse improves the SNR at the output. Similar results were also reported in [6] using a rate coding based FitzHugh-Nagumo model network with a similar topology. Both these networks used floating point numbers.

The floating point model uses the Euler approximation of equation (1) given by equation (2) .

$$V(t + \Delta t) = V(t) \times (1 - \frac{\Delta t}{\tau}) + \Delta t \times I(t) \tag{2}$$

where $\Delta t$ is the time step, and $\tau = CR$ is the membrane time constant.

Equation (2) is implemented on an FPGA platform by (i) making all values integers and (ii) making division only by powers of 2. $\frac{\Delta t}{\tau}$ is expressed as $2^{-k}$ (we call $k$ the leakage factor ) and $\Delta t$ is expressed as $2^{-m}$. This is because full integer division is expensive to implement (both in number of gates and speed) whereas bit-shifting is very fast. The synaptic response function was implemented using a look-up table. In integer form equation (2) becomes

$$\bar{V}(t+1) \approx \bar{V}(t) - \bar{V}(t) >> k + \bar{I}(t) >> m \tag{3}$$

where $>>$ denotes bit-shifting to the right.

## 3 Methodology

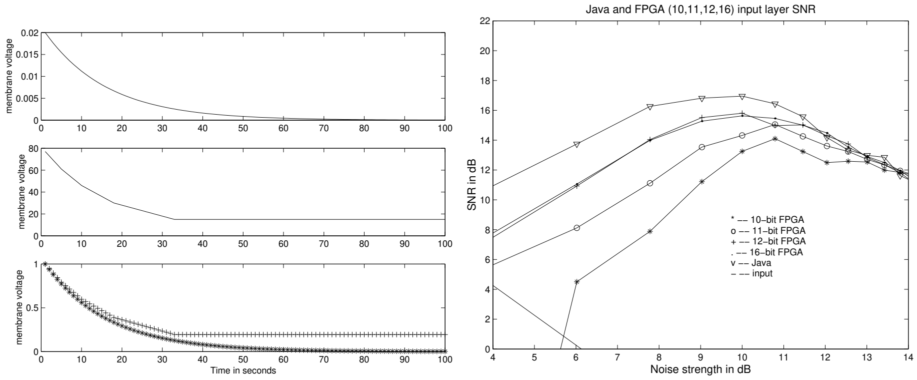
The floating point model can take inputs directly. However the integer model requires integer inputs which means the input signals must be quantised. This is achieved by multiplying by $2^n$ (where $n$ is the resolution i.e. integer length).

The floating point model was implemented in Java and the integer model in Handel-C. Handel-C code is compiled into a netlist file which is converted to a bitmap file by the placing and routing software under the Xilinx Foundation Series 3.1i package.

The main outputs of interest are the spike trains from both the “input” and “output” neurons (see figure 1) from which the power spectra were computed in Matlab using the method of Gabbiani and Koch in chapter 9 of [9]. The SNR is computed using the method of Chapeau-Blondeau in [5] (see equation (4)).

$$SNR = \frac{\log_{10} S(\omega)}{\log_{10} N(\omega)} \tag{4}$$

where $S(\omega)$ is the power spectrum of the spike train at the signal frequency and $N(\omega)$ is the power spectrum of the background noise around the signal frequency. Each SNR value is an average from 10 different spike trains.




(a) Leakage

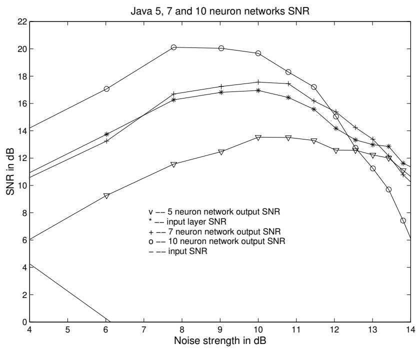
(b) Input layer SNR (Java and FPGA)

**Fig. 2.** (a)Decay due to leakage of membrane from a high start potential without input. Top: Java model decay. Middle: 12-bit FPGA model decay. Bottom: normalised decay for Java (*) and FPGA (+). (b) Input layer SNR for Java and FPGA (resolutions 10,11,12 and 16)

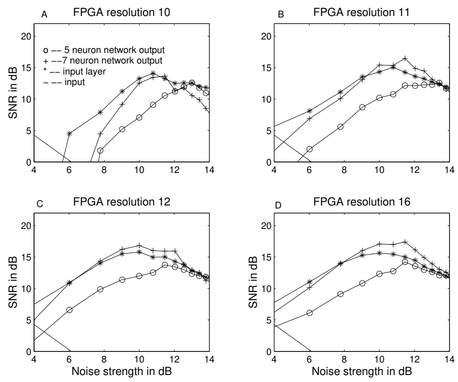
Quantisation affects the evolution of activation and its effect is that the changes in $V(t)$ over $\Delta t$ implemented by equation (3) are forced into integer values and that the integer activation value ceases to decrease when $V(t) >> 4 + I(t) >> 10) < 1$. This leakage problem in the integer model is also coupled to the size of the time step $\Delta t$. Decreasing the time step makes the problem worse as the decrement $\frac{V(t) \times \Delta t}{\tau}$, is proportional to $\Delta t$. Yet increasing $\Delta t$ is not normally an option, as we need $\Delta t \ll T$. For the chosen values of $\tau = 0.02s$ and $\Delta t = 0.001s$, the resolution needs to be at least 10 bits.

# 4 Results

The results are summarised in figures 2 to 4. They show that there is stochastic resonance at the input layer and at the output neuron in both models.




(a) Java: input layer SNR and output neuron SNR (5, 7 and 10 neuron networks)




(b) FPGA (resolution 10, 11, 12 and 16): input layer SNR and output neuron SNR (5 and 7 neuron networks)

**Fig. 3.** (a) 5, 7 and 10 neuron network input layer and output neuron SNR. (b) FPGA (resolutions 10,11,12 and 16 ) input layer and output neuron SNRs for the 5 and 7 neuron network

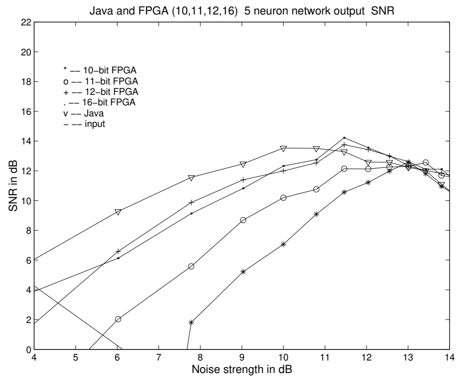
Figure 2(b) shows that at the input layer, Java SNR values are better than all the FPGA resolutions considered. It also shows that SNR values for the integer model improve for resolutions up to 12 bits after which they stay constant as evidenced by the similarity between SNR values for 12 bits and 16 bits resolutions. This result applies to resolutions up to 32 bits [10].

The SNR at the output also increases with increase in resolution (see figure 4) just like it does at the input layer. In both figures the SNR values for the integer model resolutions 12 and 16 bits slightly overtake the Java SNR between 11dB and 13dB noise values. This is interesting because at the input layer SNR values for the Java model are always better than at all the FPGA resolutions considered (see figure 2(b)).

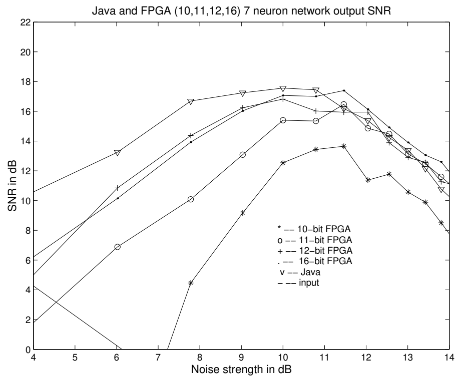
Figure 3(b) shows that the SNR at the output neuron is influenced by both the resolution and number of input neurons. The SNR value at the output increases with an increase in the number of input neurons. The 10-bit resolution SNR result shows that the input layer SNR is always better than than the output neuron result for both 5 and 7 neurons networks. For 11 bits, 12 bits and 16 bits, only the output for the 5 neurons network is less than the input layer neuron

SNR, the output neuron SNR for the 7 neurons network overtakes that of the input layer between 8dB and 13dB noise values. This suggests there is complex interaction between the output neuron SNR, the number of input neurons and the resolution of the integer model.

Figure 3(a) shows that as the network increases in size so does the SNR of the output neuron. This result was only proven on a limited scale on the FPGA (see 3(b)) because we were unable to produce a network with more than 7 neurons due to computing equipment limitations.




(a) 5 neuron network output neuron SNR for Java and FPGA (resolutions 10, 11, 12 and 16)




(b) 7 neuron network output neuron SNR for Java and FPGA (resolutions 10, 11, 12 and 16)

**Fig. 4.** (a): 5 neuron network output neuron SNR ( both Java and FPGA). (b) 7 neuron network output neuron SNR ( both Java and FPGA)

## 5 Discussion

Figures 2(b), 3 and 4 show that the differences in SNR values for the resolutions considered in the low noise regime are much higher than the differences at the high noise end. This applies to both input and output neurons. The difference in SNR in the lower noise levels at the input layer (see figure 2(b)) are as much as 3dB and the same differences for the output layer are as much 6dB (see figure 4(b)). The decrease in the differences in SNR (as noise strength is increased) may suggest that quantisation effects are more pronounced at lower signal plus noise levels. This could be attributed to loss of changes in the activation. The maximum input layer improvement in SNR due to SR on the FPGA platform never reaches the same level as that of the floating point model for lower noise levels.

## 6 Conclusion

The results presented here show that SR in an LIF network can occur in low resolution integer based implementations for both single neuron and the topology used in figure 1. It does not require a continuous system. This is important because such systems can permit digital electronic implementation with real-time performance, for example through FPGA implementations. The amount of SR may not be as high as in high resolution implementations. It has also been shown that in integer systems SNR saturates as we increase the bit length at which activation is computed. For the parameters chosen in this simulation, SNR was not found to improve with an increase in the resolution beyond 12 bits. The improvement in output SNR when the output of a parallel array of neurons converge on one neuron justifies the existence of such networks in the central nervous system confirming the findings of [6]. In addition, we have shown for the first time that this increase in SR continues even with much reduced (integer) resolution.

## References


1. *Virtex 2.5 V Field Programmable Gate Arrays.*
2. M. Barbi, S. Chillemi, and A.D. Garbo. The leaky integrate-and-fire neuron: a useful tool to investigate SR. *Chaos, solutions and fractals*, pages 1273–1275, 2000.
3. R. Benzi, G. Parisi, A. Sutera, and A. Vulpiani. Stochastic resonance in climatic changes. *Tellus*, 1982.
4. G. Bostock. *FPGAs and Programmable LSI.* Butterworth Heinemann, 1996.
5. F. Chapeau-Blondeau, X. Godivier, and N. Chambet. Stochastic resonance in a neuronal model that transmits spike trains. *Physical Review E*, pages 1273–1275, 1996.
6. J.J. Collins, C.C. Chow, and T.T. Imhoff. Stochastic resonance without tuning. *Nature*, 1995.
7. L. Gammaitoni, P. Hanggi, P. Jung, and P. Marchesoni. Stochastic resonance. *Review Modern Physics*, 70:223–287, 1998.
8. C. Koch. *Biophysics of Computation.* Oxford University Press, NY, 1999.
9. C. Koch and I. Segev, editors. *Methods in Neural modeling: from ions to networks.* The MIT Press, 1999.
10. N. Mtetwa, L.S. Smith, and A. Hussain. Stochastic resonance and finite resolutions in a leaky integrate-and-fire neuron. In *in press, proceedings of ESANN2002*, 2002.
11. D. Petracchi, I.C. Gebeshuber, L.J. DeFelice, and A.V. Holden. Stochastic resonance in biologocal systems. *Chaos, Solutions and Fractals*, pages 1819–1822, 2000.
12. M. Stemmler. A single spike suffices: the simplest form of stochastic resonance in model neurons. *Network: Computation in Neural Systems*, 7(4):687–716, November 1996.
13. K. Weisenfeld and F. Moss. Stochastic resonance and the benefits of noise: from ice ages to the crayfish and squids. *Nature*, 373:33–36, 1995.
14. L. Zalanyi, F. Bazso, and P. Erdi. The effect of synaptic depression on stochastic resonance. *Neurocomputing*, pages 459–465, 2001.