ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 343-348

Stochastic Resonance and Finite Resolution in a Leaky Integrate-and-Fire Neuron

Nhamo Mtetwa, Leslie S. Smith and Amir Hussain
Computing Science and Mathematics Department, University of Stirling,
Stirling, FK9 4LA, Scotland

Abstract. This paper discusses the effect of stochastic resonance (SR) in a leaky integrate-and-fire (LIF) neuron and investigates its realisation on low resolution digitally implemented systems. We report in this new study that stochastic resonance which is mainly associated with floating point implementations is possible on lower resolution integer based representations which result in real-time performance on digital hardware.

1 Introduction

Adding noise to a telephone line does not improve transmission of information. Yet, under certain circumstances, adding the right amount of noise can enhance rather than diminish the detection of a weak periodic signal. This phenomenon is known as 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 [6] for a review). In this context "weak" means that the input signal is so small that when applied alone it is undetected [10]. Simulation of SR in "continuous" systems using floating point based models is well established [3], [12] and [11]. These floating point models are fairly accurate but they do not always result in real-time performance. We aim to show that SR still occurs with lower resolution integer based representations which allow real-time performance on digital hardware, especially Field Programmable Gate Array (FPGA) based implementations. The FPGA allows us to vary the 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]. Limited resolution experiments have previously been tried for back-propagation networks [7]. We discuss first the realisation of SR in a LIF neuron model and then 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 integer models where the integer length is varied downwards from 32 bits.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 343-348

The SR effect is assessed by applying a subthreshold periodic signal plus noise to an LIF neuron, and examining the 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. Our results show that both models exhibit SR. SR is stronger on the floating point model than on the integer model. For the integer model we found that only bit lengths above 10 produced reasonable results, and that results improve from 10 bits up to 12 bits after which they stay constant up to 32 bits.

2 The Model

The model is based on the LIF neuron model which is described by equation (1) which idealises the neuron as a capacitor C, in parallel with a resistor R. The effective current $I(t)$ may hyperpolarise or depolarise the membrane.

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

In the SR context, $I(t)$ is composed of a periodic subthreshold sinusoidal part and a stochastic component made of Gaussian white noise so that $I(t) = I_0 + m\cos(\omega t) + \sigma\xi(t)$, where $\omega = 2\pi/T$, T being the period of the signal, and $\xi(t)$ represents Gaussian white noise of zero mean and amplitude σ [2]. The LIF is widely used in neuronal modelling because it captures most of the important subthreshold dynamics of real neurons [8].

In Euler form equation (1) becomes

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

where Δt is the time step, and $\tau = CR$ is the membrane time constant. The floating point model is a Java simulation of equation (2) .

Equation (2) is made suitable for implementation 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 of the FPGA) and Δt 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. Here a full integer division resulted in the LIF requiring 87 236 gates compared to 10 718 gates using bit-shifting. For a neuron with $\tau = 20ms$ and time step $\Delta t = 1ms$, we have the following approximation using equation (2)

$$V(t + \Delta(t)) \approx V(t)(1 - \frac{1}{2^4}) + \frac{I(t)}{2^{10}}$$

In integer form the above equation becomes

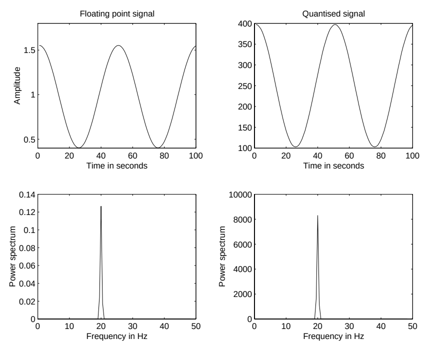
$$\bar{V}(t+1) \approx \bar{V}(t) - \bar{V}(t) >> 4 + \bar{I}(t) >> 10 \tag{3}$$

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

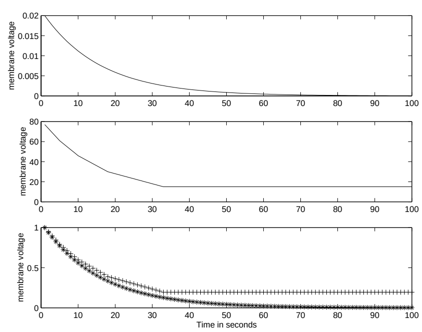
ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 343-348

3 Methodology

The floating point model can take inputs directly. However the integer model requires integer inputs which means the signals need to be quantised: see figure 1(a). The floating point subthreshold sinusoidal signal and the Gaussian



(a) Signal Quantisation



(b) Leakage

Figure 1: (a) 20 Hz input subthreshold signal and power spectra plots. Top row left to right: Floating point input signal and integer input signal. Bottom row: Power spectra of floating point signal(left) and integer signal (right). The quantised signal preserves the frequency spectrum of the signal. (b) 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 (+).

white noise used in Java are quantised by multiplying them by 2^n (where n is the resolution i.e. the integer length). The same is done for the threshold. Quantisation is a nonlinear process which means we cannot be sure that we are giving the same signal to both models. Since we are investigating SR, we need to check that the frequency spectrum of the signal is preserved. To check this we apply a Fourier Transform to both the integer and floating point signals. Figure 1(a) shows that the integer model preserves the frequency spectrum of the floating point signal.

The floating point model (see equation 2) was implemented in Java. The integer model (see equation 3) was implemented in Handel-C. The 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 simulation is run as follows. A noise stream is generated and then with the same noise stream the simulation is run with different noise amplitudes across the resolutions of choice. The main output of interest are the spike trains. The power spectra of the spike trains are computed, and used to compute the

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 343-348

SNR. The power spectrum S of each spike train is computed using the method of Gabbiani and Koch in chapter 9 of [9]. The SNR is computed using the method of Chapeau-Blondeau in [5] captured in equation (4).

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

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.

To investigate the effect of quantisation on the activation we looked at the leakage. The membrane potentials for both models were initialised to high start values, close to the threshold values and allowed to decay. We plot the decaying potentials for both floating point and FPGA with limited resolution. We also normalise them and plot them on the same axis (see figure 1(b)).

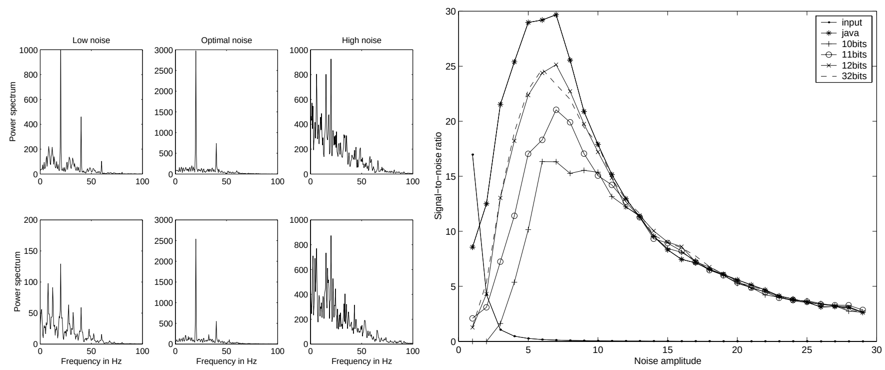
Figure 1(b) shows that the membrane potential for the floating point model decays to almost zero in the absence of input but that of the integer model decays to $2^k - 1$ where k is the leakage factor. This means if the leakage factor is 4 the integer neuron will not leak below $2^4 - 1 = 15$. We were therefore limited to resolutions which result in threshold values which are above 15 when quantised. As a result only resolutions which start from 10 bits were considered for the chosen values of $\tau = 0.020s$ and $\Delta t = 0.001s$. The effect of activation on quantisation is that the changes in $V(t)$ over Δt implemented by equation (3) are forced into integer values and that this integer 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 Δt. Decreasing the time step makes the problem worse as $\frac{V(t)\times\Delta t}{\tau}$, the decrement, is proportional to Δt. Yet increasing Δt is not an option, as we need $\Delta t \ll T$.

4 Results

The results are summarised in figure 2. They show that both the Java model and the FPGA model exhibit stochastic resonance. In figure 2(a) the power spectra go through a maximum as noise strength is increased for both models. The SNR exhibits the same behaviour (see figure 2(b)) and this is the signature for stochastic resonance [2]. Figure 2(b) also shows that the SNR for the Java model is higher than any of the different resolutions for the FPGA. For the FPGA model we have useful results starting from a resolution of 10 bits. SNR values improve from 10 bits up to 12 bits after which they remain constant independent of further increase in the resolution as shown by the similarity of the 12 bit and 32 bit SNR plots in figure 2(b).

5 Discussion and Conclusions

Figure 2(a) (left most column) shows that the low noise power spectrum of the FPGA model is more noisy and has less pronounced peaks than that of



(a) Power spectrum (b) Stochastic Resonance

Figure 2: (a) SR in the power spectrum. Top: shows the typical resonance effect for the Java model as we move from low noise to high noise values. Bottom: resonance effect in the integer model (12 bits resolution) for the same noise values as the Java model. (b)20 Hz subthreshold signal SNR plots for both Java (Floating point) and FPGA (different integer resolutions: 10, 11, 12 and 32 bits as shown on graph) compared with a common input SNR.

the Java model. This suggests that the effect of quantisation is strong at low noise levels. This effect is confirmed by the differences in SNR values between the discrete and continuous models for low noise values shown in figure 2(b). On the high noise side the difference in power spectra is much smaller, but the power spectra for both models are very noisy.

The maximum improvement in SNR due to SR on the FPGA platform never reaches the same level as that for the floating point model for lower noise levels. The difference in SNR in the lower noise levels for the different plots in figure 2(b) are as high as 10dB. This could be attributed to loss of changes in the activation. The differences in SNR decrease to about 5dB at the optimal noise level and finally fall to zero in the high noise region. The decrease in the differences in SNR (as noise strength is increased) suggests that quantisation effects are more pronounced at lower signal plus noise levels. For future work, we intend to investigate any underlying nonlinear quantisation effects using e.g. Higher Order Spectra (HOS) analysis techniques.

The results presented here show that SR in an LIF can occur in low resolution integer based implementations. It does not require a continuous system. This is important because such systems can permit 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 discrete systems SNR saturates as we increase the bit length at which activation

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 343-348

is computed. For the parameters chosen in this simulation, SNR was not found to improve with an increase in the resolution from 12 up to 32 bits.

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] L. Gammaitoni, P. Hanggi, P. Jung, and P. Marchesoni. Stochastic resonance. *Review Modern Physics*, 70:223–287, 1998.

[7] M. Hohfeld and S.E Fahlman. Probabilistic rounding in neural network learning with limited precision. *Neurocomputing*, 1992.

[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] 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.

[11] 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.

[12] 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.