Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Robotics and Autonomous Systems 45 (2003) 249–263

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Analog computation with rings of quasiperiodic oscillators: the microdynamics of cognition in living machines

Edward A. Rietman [a,*], Mark W. Tilden [b], Manor Askenazi [c]

[a] *Triton Systems Inc., 200 Turnpike Road, Chelmsford, MA 01824, USA*
[b] *Institute for Physical Sciences, Los Alamos, NM, USA*
[c] *Microbia Inc., Cambridge, MA, USA*

Received 17 October 2002; received in revised form 2 June 2003; accepted 15 August 2003

Abstract

We describe experimental results to demonstrate the wide-ranging computational ability of quasiperiodic oscillators built from rings of differentiating Schmitt triggers. We describe a theoretical model based on necklace functions to compute the number of states supportable by a ring circuit of a given size. Experimental results are presented to demonstrate that probabilistic state machines can be built from these ring circuits. Other experimental results are given to demonstrate that the rings can model spiking neural network circuits.

Keywords: Quasiperiodic oscillators; Microdynamics; Schmitt trigger

1. Practical and theoretical foundations

Hasslacher and Tilden [1] describe a special type of ring oscillator – called a nervous network – and describe its use in controlling the dynamics of walking robots (see Fig. 1). They describe that the circuit cannot only control the gait dynamics but that it also assists the robot in adapting to a rugged landscape (a fractal landscape). Obviously this circuit is a primitive analog computer taking inputs from the external world and transforming those signals to electrical pulses which drive motors controlling the gait of a robot. There are several overall variations in which the input and output signals are fed into and out of the main 'ring-core' and these have been described by Hasslacher and Tilden [1]. Here we focus our attention on the 'ring-core' and demonstrate how these ring circuits can be used as analog computational machines. We describe the microdynamics, or the 'bit-level' dynamics, of these circuits, and we describe some theoretical and simulation results.

1.1. The Nv-neuron

The most popular artificial neuron is a spiking neuron. When it receives voltage signals it produces a pulse train. As the voltage increases the frequency or firing rate increases. Fig. 2(A) shows a simplified neuron of the type investigated by Murray and Tarassenko [9] among others. As seen from the graph, the input voltage increases and the output frequency also increases. Basically it is a voltage-controlled oscillator. The *RC* components determine the frequency behavior.

* Corresponding author. Fax: +1-978 250 4533.
E-mail address: erietman@tritonsys.com (E.A. Rietman).

0921-8890/$ – see front matter
doi:10.1016/j.robot.2003.08.002

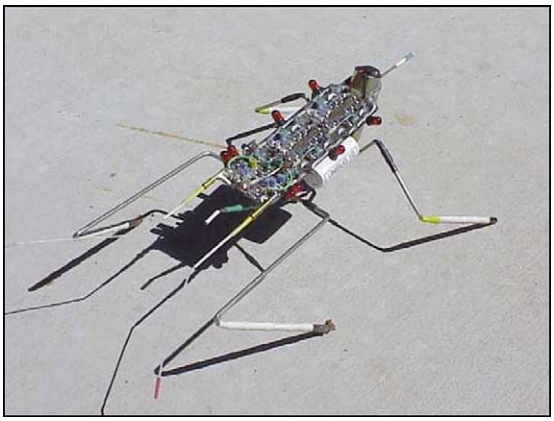


Fig. 1. Simple insect-like robotic organism – a living machine.

Fig. 2(B) shows a second type of neuron that also acts as a voltage-controlled oscillator, but requires a threshold and saturates. The sigmoidal curve approximates the dynamics of biological neurons. The operational voltage range for which the neuron fires is a narrow voltage band given by the threshold of the Schmitt trigger. Below some input voltage-threshold the neuron is not firing. Above a threshold the neuron will saturate and not increase its firing rate. The frequency at the inflection point of the sigmoid is given by the *RC* time constant. The Schmitt trigger is the key component making up this neuron. It is also the key component in another type of artificial neuron called the nervous neuron (Nv-neuron).

The overall focus of this paper is to describe the computational dynamics of small networks built from

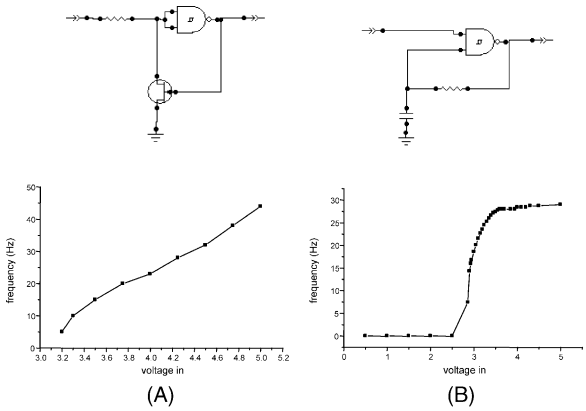


Fig. 2. Two artificial neurons compared in this study.

the Nv-neuron. The basic neuron circuit has been described by Tilden [14]. We made certain modifications to this circuit to enable us to experiment with it in larger systems and make quick changes in the basic architecture of the computational nodes. The basic circuit is an inverting Schmitt trigger with a differentiating input. Our circuits were built with the NAND Schmitt trigger, like those used in the circuits of Fig. 2. This allows us to quickly modify the neuron architecture with a minimum of disruption to the network. A pure inverting Schmitt trigger could be substituted for the NAND Schmitt trigger.

Fig. 3 shows the basic Nv-neuron. The differentiating circuit is well known and Horowitz and Hill [5] Jones [6] are excellent discussions of the behavior. Since the circuit uses an inverting device, when the input to the neuron (position A, in Fig. 3) is high the output will be low, and the neuron is 'firing'. When node A is low the neuron is in a quiescent state and the output is high. This is shown in the figure as the input signal and the output signal. The input to the neuron can be expressed as

$$V_{\text{in}}(t) = V_{\text{h}}[-u(t-t_0) + u(t-t_1)], \tag{1}$$

where V_{h} is the high-output-voltage and $u(t)$ is the step function given as

$$u(t) = \begin{cases} 0 & t < 0\,, \\ 1 & t \geq 0. \end{cases} \tag{2}$$

The behavior at node A is much more interesting. Mathematically it is given by

$$V_{\text{A}}(t) = \begin{cases} 0 & t < t_0, \\ -V_{\text{h}}\left[-\dfrac{\exp(-(t-t_0))}{RC}\right] & t_1 > t > t_0, \\ [V_{\text{h}} + V_{\text{A}}(t_1^-)]\dfrac{\exp(-(t-t_1))}{RC} & t > t_1. \end{cases} \tag{3}$$

While the input is high the voltage at node A will be low (zero volts). If the input goes low, the capacitor initially follows the leading edge of the input (negative) and discharges during the time $t_1 - t_0$, whereupon it forces node A again to zero volts. So node A goes negative and discharges until the voltage is zero. If the input again goes high the capacitor will again initially follow the input, but will start to charge. The charging

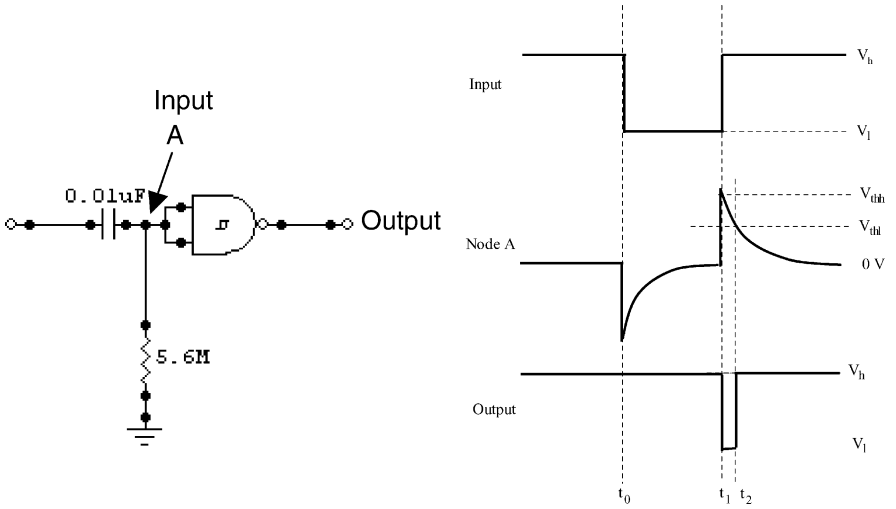


Fig. 3. The basic circuit of the Nv-neuron and its input/output response.

time is of course given by the product of the capacitor value (F) and the resistor value (Ω) $\tau = RC$. Throughout the paper we will simply refer to this as tau.

The output of the circuit, V_{out}, is given by influences from the above dynamics at the input to the Schmitt trigger (node A) but also by the dynamics of the Schmitt trigger itself. As shown in Fig. 4, the Schmitt trigger exhibits a threshold and hysteresis in the voltage input/output curve. This means the output is dependent not only on the input but also the recent history input. The neuron fires only when the input goes from a low to high transition and when the voltage at node A reaches V_{thh}. It then fires for a time, tau, determined by the *RC* components. When the voltage at node A exceeds the high threshold, V_{thh}, the output is driven low. When the voltage at node A drops below the low threshold, V_{thl}, the output is driven high again. The neuron will fire while the voltage at node A is between the high threshold and the low threshold.

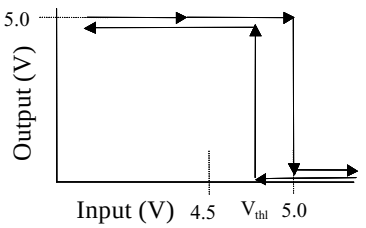


Fig. 4. Input/output hysteresis curve for the Schmitt trigger.

The following relation gives the firing time of the neuron:

$$t_2 - t_1 = -RC \ln\left[\frac{V_{\text{thl}}}{V_{\text{h}}(1 - \exp(-(t_1 - t_0)/RC))}\right]. \quad (4)$$

This equation says that the firing time is dependent on the low threshold voltage, the magnitude of the input, and the *RC* constant. So a long duration input pulse will cause a long output pulse and a short input pulse will cause a short output pulse. This is really an important point. Short *duration* pulses will propagate through a network faster than long duration pulses. However, if the frequency of input is too high (relative to tau) the neuron stays in the resting state (logic high) and does not fire (logic low). In practice the variation between *RC* components will often result in-phase noise and the oscillator will not enter an oscillator death state. But oscillator death can be observed under certain high frequency conditions depending on tau. The actual relation between oscillator death caused by high frequency and tau has not been measured.

In summary the Nv-neuron has some desirable properties not observed in networks built with inverters. First the Nv-neuron output pulse has a variable duration proportional to the input pulse. Second the end of the input pulse triggers the output; so shorter pulses

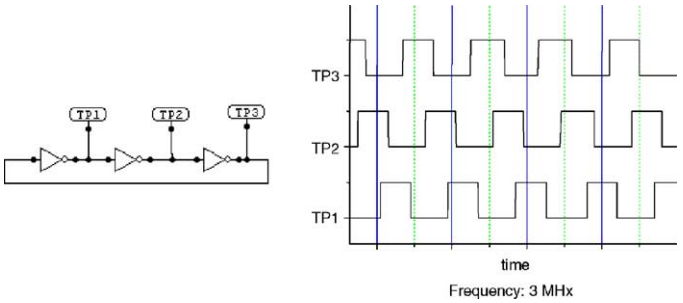


Fig. 5. Classical ring oscillator built from CMOS inverters.

will travel faster in the ring. Third there is a maximum length of the output pulse, and fourth if the input pulse is too short relative to the *RC* time constant than it will not trigger an output pulse. This also results in the observation that trailing pulses that may be too short will not cause an output. This results in the device entering a 'refractory' period. How this analog behavior relates to real neurons is a question for further study.

1.2. Ring oscillators

The analog processing elements onboard our autonomous robots are rings of the Nv-neurons. As a baseline to our study of rings of Nv-neurons we will first examine the behavior of ring oscillators built from CMOS 4049 inverters. By definition an oscillator is an unstable circuit, or at best, a metastable circuit. When constructing oscillators from CMOS inverters, the charge time, tau, is determined by the physics of the transistors making up the inverter. Fig. 5 shows the relevant circuit and an actual oscilloscope trace for all three nodes in a 4049 ring circuit. If this was an ideal circuit built from ideal inverters with instantaneous time response then we could expect that when one pulse is high the next pulse in line would be low. So ideally we would expect two adjacent inverters to be oscillating 180° out-of-phase. But as seen in the figure, these real-world inverters are in-phase a fraction of the time. Furthermore, the degree of out-of-phase change does not appear to be constant (though that is not indicated in the figure), and is related to the phase noise in the ring oscillator.

Rings of the Nv-neuron (see Fig. 6) oscillate. Any number of neurons $(1, 2, \ldots, N)$ can be connected into a ring configuration and obtain oscillations. In general, older and slower chip types (40XX) can be used to build Nv-ring oscillators, because the threshold input voltage is wider. The LS04 is fast and does not have a wide threshold voltage. The HCT240 has a wide input voltage and is fast. However, it tends to generate more phase noise than the 40XX chips. So all our studies were done with the 40XX family, and in particular the 4093.

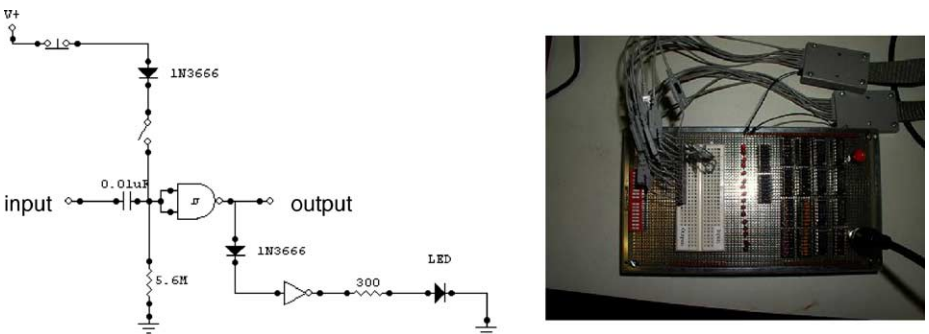


Fig. 6. The experimental Nv-neuron, and the rapid prototype breadboard.

Besides the main Nv-components, shown in Fig. 6, there are other components that enabled input and output without disturbing pulse trains in the networks. The diodes are used to buffer signals into and out of the Nv-networks, and prevent the *RC* waveform at input (point A in Fig. 3) from being clipped at $V_h + 0.7$ V. The 7404, the 300 Ω current limiting resistor and the LED allows observations of the pulsing at each node. Sixteen nodes like this were wired onto the circuit board shown in Fig. 6 and a small solder-less breadboard was used for quick configuration of various types of networks. The switch shown in the schematic diagram was part of a bank of DIP switches, and the diodes were all connected to one pushbutton switch. The DIP switches allowed presetting the input for each node and the pushbutton switch enabled and then allowed the network to settle to some limit-cycle or attractor point.

1.3. Nv-rings and complexity

In this section we will briefly review some earlier papers describing the dynamics of Nv-networks [1,2,13]. This will be followed by a discussion of totient functions and necklace machines and how they model the dynamics of Nv-rings.

Still and Tilden [13] describe in detail how pulses from the Nv-ring drive the leg dynamics for small walking robots. In their 1995 paper Hasslacher and Tilden describe, at a rather high-level of abstraction, the dynamics for several sizes of Nv-rings, and describe how input disturbances (e.g. pulses from an antenna) automatically change the gait dynamics of the robot. But more interesting is feedback from torque sensors on the motors driving the legs. When signals from the torque sensors enter the Nv-ring they not only disturb the gait dynamics, but they disturb it in such a way as to essentially mimic the landscape the robot is walking on. If the landscape is particularly rugged it will automatically adapt to this landscape. If the robot is attempting to walk up a sand hill it will automatically adjust its gait accordingly to keep from slipping. None of the cognitive behavior is the result of A/D conversion and digital processing (more on this later). This cognitive behavior comes about essentially for free from the rich input/output dynamics of the Nv-rings and Nv-networks. The neural core is not only the central pattern generator for the robot, but also the cognitive processor.

Because of the memory effect of hysteresis in the Schmitt triggers the robot essentially builds up, in a short-term memory, a dynamic model, or internal representation, of the world. As we will see later, with larger rings more complex pulse patterns are sustainable. And this suggests that complex arrays of Nv-rings could be used to build more advanced brains for robots. None of these systems require programming in the conventional sense.

One of the goals of our research is to define the limits and capabilities for small networks, in order to build a ‘library’ of ‘components’ to build biomorphic robotic brains. The computational elements in networks and arrays of Nv-rings are *the rings*. The individual Nv-neurons are only components that build up the computational elements – the rings themselves are the computational elements for networks. For the long-term goal of building brains, we must keep in mind that highly connected networks will not result in complex behavior but rather oscillator death. Like all complex networks, the connectivity must be near a threshold – at the edge of chaos – in order to exploit emergent behavior. We need to define that threshold, which we know is related to the *RC* time constant. Once we have that threshold and a ‘library’ of ‘components’ we will be able to use a genetic algorithm to design complex networks for specific behaviors. The adaptability of networks of these rings is not yet known and needs to be explored.

Continuing with the review by Hasslacher and Tilden [1], if we represent 1s as ‘processes’ in the Nv-rings then Fig. 7 shows several distinct process patterns for small rings. Individual processes will remain independent of each other as long as they are separated by two or more tau – the time constant for the neurons. In other words they must be separated by two neurons. Because the circuits are rings, patterns such as 100100, 010010, and 001001 are identical. In the following we convert the lowest Boolean equivalent to decimal and use that decimal number for further discussions.

The pattern, for example, 101000, will cycle in synchronization forever provided one neuron in the chain has a shorter tau than the others. If the tau are nearly equal, then they will mode lock into 100100. In a saturated pattern, 101010, the entire process chain ro-

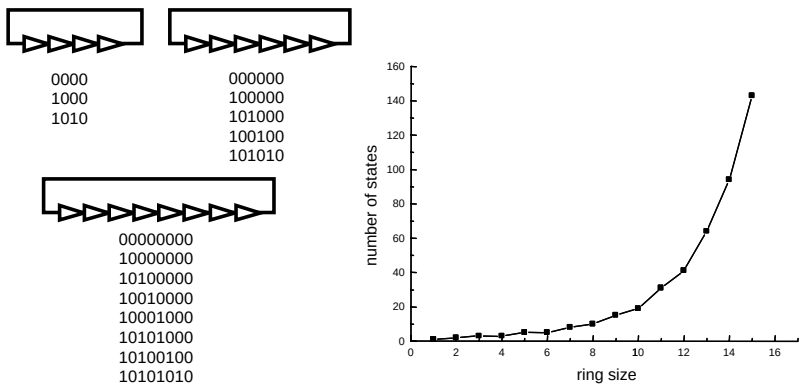


Fig. 7. Some distinct process patterns of small Nv-rings and the relation between the ring size and the number of distinct process patterns.

tates at the speed of the shortest Nv-neuron. This is determined by the lowest tau and is given in Eq. (4). As more processes are introduced to an unbalanced Nv-ring, they will travel faster. In the next section we will see results for a number of these.

The number of distinct processes sustainable in a loop of a given size $(1, 2, 3, 4, \ldots, N)$ follows the sequence $(1, 2, 2, 3, 3, 5, 5, 8, 10, 15, 19, 31, 41, 64, 94, 143, \ldots)$. This sequence follows a function known as a *2-ary necklace* [10,15–17] and is given by

$$N(n, 2) = \frac{1}{n}\sum_{i=1}^{\nu(n)} \phi(d_i)\,[F(d_i - 1) + F(d_i + 1)]\,, \qquad (5)$$

where d_i are the divisors of n with $d_1 \equiv 1, d_2, \ldots, d_{\nu(n)} = n$; $\nu(n)$ is the number of divisors of n; $\phi(n)$ the totient function, and $F(\cdot)$ the Fibonacci sequence. The totient function is also called the Euler totient function and is given as the number of positive integers less than n, which are relatively prime to n [16,17].

2. Experimental results

In this section we will discuss the dynamics of simple Nv-rings and we will discuss experimental results for coupling between small rings and some observations on a large ring of 48-nodes.

2.1. Basic dynamics

The apparatus shown in Fig. 6 was used for numerous experiments in exploring the dynamics of small rings. In addition we built two other systems to enable us to explore coupling of rings and pulse storage in larger rings, and these will be discussed subsequently. Our first experiments were to confirm the theoretical conjectures described above. We wired rings with 4-, 5- and 6-nodes to which we initialized the nodes and allowed the oscillator to settle to a limit-cycle. Each ring was initialized with binary equivalents of $0, 1, \ldots,$ up to the maximum for that ring (e.g. 15, 31, and 63 for 4-, 5-, and 6-nodes, respectively). After letting the system settle to an attractor or limit-cycle, the output of each node was captured on a digital oscilloscope. Fig. 8 shows the results. These graphs are read from left to right and top to bottom. The first graph of each set is initialization of 0, the second is initialization of 1, etc.

The first observation from Fig. 8 is that the ring with 5-nodes clearly looks chaotic and likely does not support the necklace theory. To understand the results for the 4- and 6-node rings it is necessary to first recall from Fig. 3 that a neuron fires when the output goes from high to low. So it is clear from Fig. 8 that the 4-node exhibits essentially only two limit-cycles (0101, 0001). (Limit-cycles are regular repeating patterns.) Of course (0000) is also a stable state. So the necklace theory clearly supports the observations for this ring.

The 6-node ring supports four distinct limit-cycles (010101, 100100, 101000, and 100000) and one attractor point (0000). This is, again, in complete agreement with the necklace theory.

Because of space limitations we have not presented the results for 8-nodes, but the necklace theory again is supported. But at 5-nodes (shown in Fig. 8) and

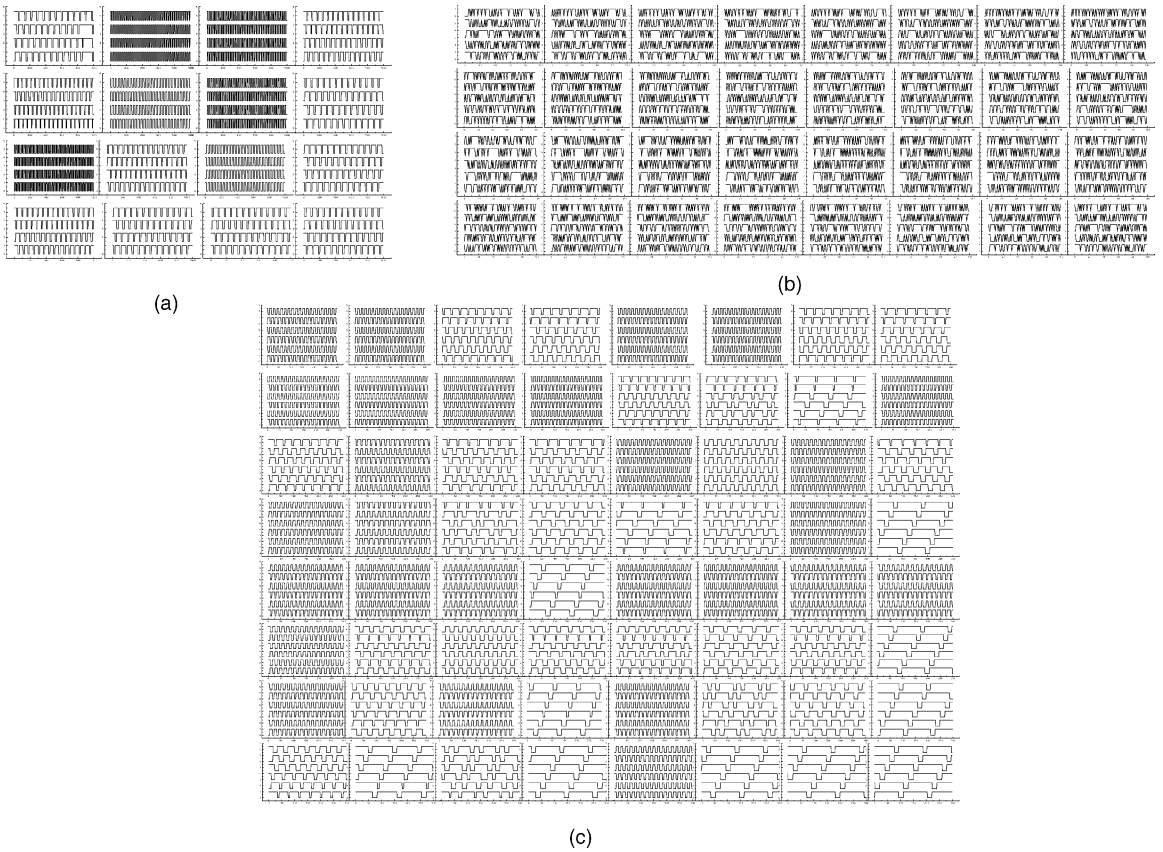


Fig. 8. Scope tracing for each node in 4-, 5-, and 6-node ring experiments.

7-nodes (not shown) the limit-cycles appear to be "chaotic".

In order to test if the 5-node data stream is random we compared it with a random data stream of 0s and 1s. We generated a random 0, 1 stream of the same length as the data stream. Then we computed the run-length for 0s and 1s for both the random stream and the 5-node data stream, and compared the autocorrelation function for these two time streams.

The autocorrelation function (ACF) is easily described. Consider N observations of a discrete time series. We can form $N-1$ pairs of observations of the run-length of the type $(z_1, z_2), \ldots, (z_{N-1}, z_N)$. The correlation coefficient can be written by reading the first observation in each pair as another variable. We can write the autocorrelation function as

$$\rho_k = \frac{\mathrm{Cov}(Z_t, Z_{t+k})}{\sqrt{\mathrm{Var}(Z_t - \hat{Z}_t)}\sqrt{\mathrm{Var}(Z_{t+k} - \hat{Z}_{t+k})}}, \qquad (6)$$

where Cov is the covariance and Var is the variance between two samples in time.

As seen in Fig. 9, the autocorrelation is significantly different for each of these time streams. The random and 5-node data show similar behavior to time lag of about 25 after that the 5-node time stream shows increased correlation from about lag 25 to 32. In this lag-region the maximum amplitude of the correlation is 0.25. A correlation of 0.25 is significant albeit not large. We can conclude that the 5-node ring is not random but we cannot conclude that the 5-node is chaotic (deterministic chaos). It remains to be determined how other odd numbers of nodes will behave in ring configurations.

Naturally, in order to use these Nv-rings as computational elements in larger networks and arrays we must map their input/output behavior. Fig. 10 shows the I/O behavior for the 4-, 6-, 8- and 10-node

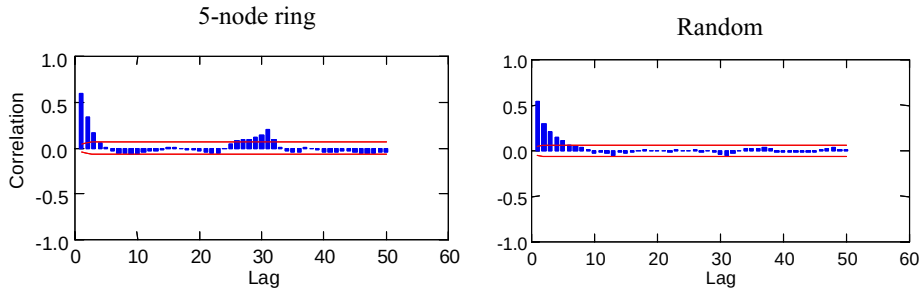


Fig. 9. ACF plots for 5-node Nv-ring and random. The lag represents the time–distance for correlation.

Nv-rings. The numbers are all in decimal equivalents (or deceq). Considering the necklace function, and Fig. 7, we can work out the stable limit-cycles for any size ring and of course we can write that as a smallest decimal equivalent. Examining the 4-node machine attractor diagram of Fig. 10 there are two stable limit-cycles (1, 5) and one attractor point (0). The figure shows the effects from initializing the machine with any state and then shows the attracting limit-cycle. For example, initializing with (0, 3, 4, 7, 9, 11, 12, 13, 14, 15) will result in the 4-node machine settling to the (1) limit-cycle; while initializing with (1, 2, 6, 8, 10, 5) will result in it settling to the (5) limit-cycle. Of significance is that many of the unstable initializations settle to one of the attracting limit-cycles. But also note that some of the initializations at a 'stable' limit-cycle settle to a different limit-cycle. We see that initialization (1) settles to (5) and (5) settles to (5). Limit-cycle (5) for the 4-node machine is saturated with the maximum number of processes.

Fig. 10 also shows the attractor diagrams for the 6-, 8- and 10-node machines without the leaf nodes – the outer most nodes that feed into other nodes. Only the 'stable' limit-cycles are shown and the effects from initialization by these limit-cycles. In each case one of the strongest attractors is the saturated limit-cycle. In the 6-node machine limit-cycles (9) and (21) couple

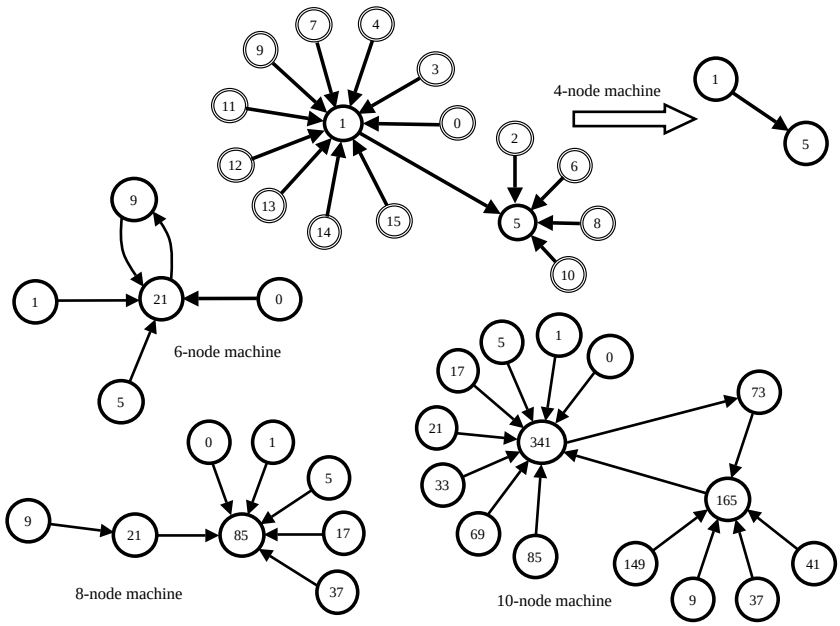


Fig. 10. Input/output behavior (attractor diagrams) for the 4-, 6-, 8-, and 10-node Nv-rings.

together as a super oscillator, in this case a two-cycle. And the 10-node machine limit-cycles (73), (165) and (341) couple together as a super-three-cycle. Larger systems exhibit even more complex dynamics.

We should add that the 'states' in these attractor diagrams are individually stable limit-cycles. But when they receive some noise (perhaps phase noise) they will jump to the indicated state with some probability. We have not determined that probability. So it should be realized that each of the attractor diagrams in Fig. 10 should have a probability attached to the edges in the attractor networks. Consequently a realistic model for these machines is a bounded probabilistic state machine. Of course this probability must be measured for intelligently exploiting these machines for computation. Once we have precise measurements on the probability it may be possible to simulate them with hidden Markov models.

2.2. Coupling experiments

Fig. 11 shows a system with rings of 4-, 6- and 8-nodes. This was designed to allow us to couple the rings through diodes and to send external pulse trains into the rings to study the dynamics. All the connections were made through (3N666) diodes between the input/output points shown in the schematic diagram in Fig. 6. The system was designed to allow us to connect, via the DIP switches, the three rings in any linear chain and to enter external pulse trains into any of the rings, via a diode and a BNC connector. In addition, we could monitor the output of any of the rings with an oscilloscope connected to a BNC connector again via a diode. Though physically larger than the capacitors shown in Fig. 6, the capacitors were the same value, 0.01uF. The *RC* components were mounted on DIP headers to allow quick changes in the tau for the nodes.

Though we did numerous coupling experiments, we will discuss only a small number of them here. In one experiment, we connected the 4- and 6-node ring to feed into the 8-node ring. All the rings were essentially running at their 'power-on' limit-cycle. There are three signals traces shown in Fig. 11. The first upper trace is the pulses coming out of the 6-node ring. The middle trace is the pulses out of the 4-node ring and the bottom trace shows the interaction between the 4-, 6- and 8-node within the 8-node ring. Essentially the 4- and 6-node rings are

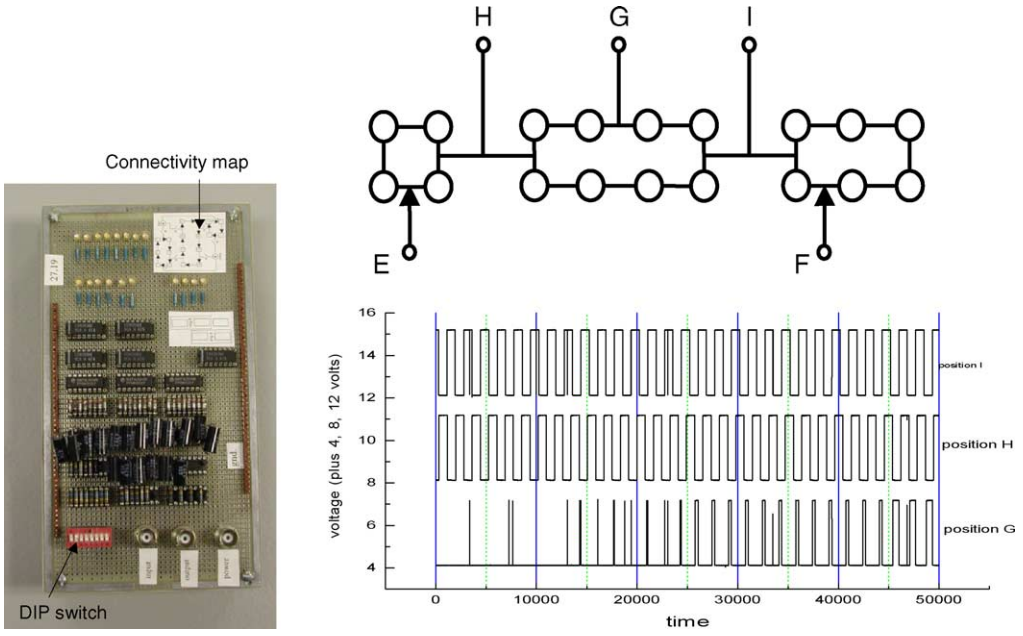


Fig. 11. Prototype circuit to study coupling, schematic diagram of connectivity for one experiment, and oscilloscope trace for one experiment.

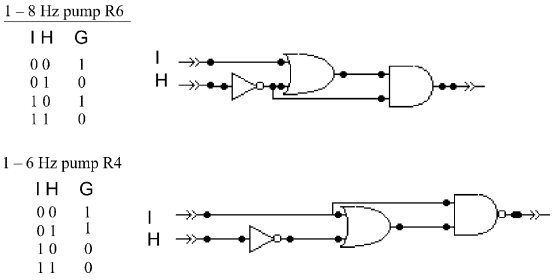


Fig. 12. Results from external pulse injection and Boolean equivalent.

180° out of phase. The bottom trace shows pulses of varying widths depending on the constructive and destructive interference among the three pulse trains. The same pattern is repeated over and over. Though the figure only shows one period of the repeated signal.

In another series of experiments we injected external pulse trains into the rings and observed the results. When injecting pulse trains into the Nv-rings it is important to remember that above a frequency near 1/tau the oscillator will settle to a deep attractor (00000. . .). If the frequency is too high, with respect to the tau the neuron will stay in the resting state and remain in logic high.

In one set of experiments, using the same connectivity as the previous experiment (i.e. 4-node → 8-node, 6-node → 8-node) we also injected pulses into the 6-node ring (or 4-node ring) and observed the effects in the 8-node ring. Varying the injected frequency (position E or F in Fig. 11) and allowing the other rings to run freely, we observed the dynamics (at positions I, H, and G, respectively, see Fig. 11) shown in Fig. 12. The pulses were monitored at position H. The figure shows the results as if it was a truth table and the Boolean equivalent circuit. In other words, the figure shows that we can construct Boolean logic circuits with from the constructive and destructive interference of pulses.

Using the connectivity shown in Fig. 11 we set out to explore the interference of pulses in rings. We let the 4-node ring run freely and connected it to feed into the 8-node ring. We also connected the 6-node ring to feed into the 8-node ring. However rather than let the 6-node ring run freely we injected external pulses into it. If the external pulse train was higher than 8 Hz it resulted in oscillator death. In this case the 6- and 8-node rings stabilized to (000000) and (00000000), respectively. This occurs when the Nv-neurons saturate and stay in the logic high state – essentially a resting state. In practice it is quite possible to use *RC* components to produce faster oscillators. We limited our study to this range since the robots were operating at these frequencies – the walking speed.

Though the nodes within the rings are differentiators, when the rings are coupled together with resistors the rings act as integrators. This allows us to essentially assemble networks of integrators like the spiking neurons described by Maas and Bishop [7]. But unlike the classical spiking neuron, our Nv-rings are pulse train generators and frequency controlled oscillators. So networks of the Nv-rings can be assembled for *spike-train* processing. The networks, or arrays, of Nv-rings are essentially hybrid computers. Analog information can be coded as frequency of pulses and time between pulse trains. The presence or absence of pulses represent digital information processing and therefore provides some robustness to noise. This hybrid computational approach has been described at length by Sarpeshkar [11] and Sarpeshkar and O'Halloran [12] with respect to spiking neurons and carries over to computational networks of Nv-rings.

However, unlike the model systems described by Sarpeshkar et al. and Maas and Bishop [7] our system is slightly more complicated in that arrays of Nv-rings would essentially be processing whole trains of spikes. So the time information imbedded in the spike-train itself (the gaps between pulse packets and the frequency of the pulse packet) is also important. In order to observe some of this computation we coupled two 6-node rings together as shown in Fig. 13. Our results are shown in Fig. 14.

Each of the small graphs shown in Fig. 14 is the result of capturing the trace from each node in each ring on a digital oscilloscope. The experiments were started by first initializing ring number 1, the driving ring, with (000001), and initializing ring number 2 by a decimal equivalent exactly like the graphs shown in Fig. 8. Like the previous set of graphs (Fig. 8) the upper-most left graph is the result of initializing ring number 2 with zero. The graph to its right was initialized with 1, etc. for the other graphs, consecutively reading across and down.

Each individual graph shows 16 oscilloscope traces. The bottom eight traces, in each graph, show the output for each of the six nodes in ring number 1.

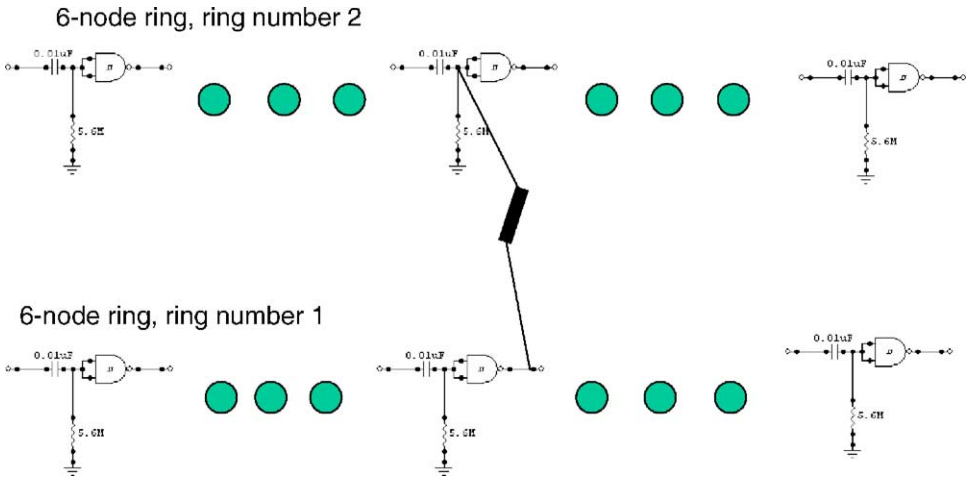


Fig. 13. Schematic diagram of coupling of two 6-node Nv-rings.

In addition it shows two additional traces that are simply dummy lines from the graphics processing. The upper set of eight traces is from the six nodes in ring number 2 and again two dummy lines. In each experiment ring number 1 was initialized with (000001). The observed effects are from coupling with a 1 M resistor as shown in the schematic diagram in Fig. 13. As could be expected the rings affect each other and the dynamics is detailed in the graphs of Fig. 14. Constructive and destructive interference effects are clearly seen. Also seen in this figure is the 6-node, two-cycle suggested in Fig. 10 between (9) and (21).

With a resistor connection link between the two Nv-rings (see Fig. 13) there are three possible coupling conditions we call: subcritical, critical and supercoupled. Consider the resistor (Rb) to ground in the individual differentiators within the loops. If the loops have, essentially identically components each loop, when not coupled, will behave essentially the same way. Now when coupling them with a resistor (Rc) we can get different types of behavior depending on the relation between Rb and Rc. If Rc $<$ Rb (the condition we used) we will have subcritical coupling. As can be seen in Fig. 14 processes are preserved in some cases but in many cases the loops saturate to the maximum decimal equivalent. This is exactly what we also observed in the attractor diagram of Fig. 10. The results shown in Fig. 14 are the results of a subcritical coupling (Rb $=$ 5.6 M, Rc $=$ 1 M).

Critical coupling is where Rc $=$ Rb. In this case processes are preserved but fall into ‘traps’ and can stay essentially in that limit-cycle forever unless perturbed by another process or an external signal. Loops of critically coupled Nv-rings can be used as counters, gates, latches, and shift registers. But in large arrays the stability is marginal; as the rings acquire more and more pulses they tend to saturate (again as shown in Fig. 10). Because of space limitations we do not show results from critical coupling. This will be a key element in a later publication.

The third condition, supercoupling occurs when Rc $>$ Rb. Here processes (e.g. patterns) can be created and destroyed, and there are only a few stable topologies that can support these phenomena. Arrays of supercoupled Nv-rings need to have an odd number of loops to stabilize dying/saturating processes. Again this will be the focus in a later publication. An interesting observation is that, while Nv-rings act internally as differentiators to the signals, when coupled into loops they act as integrators to the signals.

2.3. Threshold experiments on a large ring

We built a large Nv-ring of 48-nodes. Our rationale in using such a large ring is that we could actually inject small pulse trains into the ring and more carefully observe the dynamics. Though each node had the same *RC* values (tau $\sim$ 70 Hz), subtle variations between them resulted in short duration pulses at

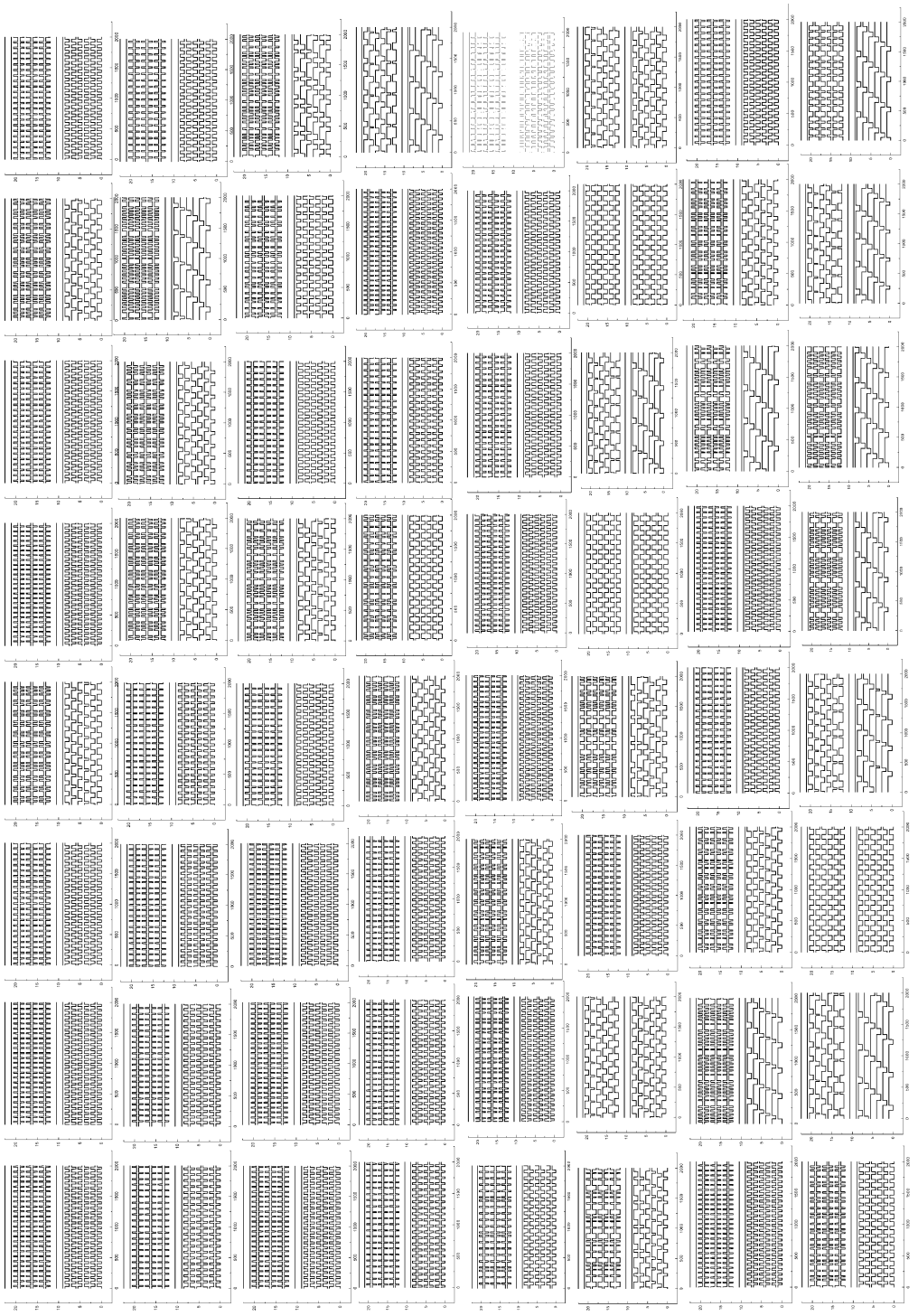


Fig. 14. Two-ring coupling experiment (see text for discussion).

several nodes. As the pulse train traveled around the loop, the short duration pulses dominated the dynamics and caused interference with other slightly longer duration pulses and with the phase noise. The end result was that the pulse train transformed itself into a stable pulse train able to continuously circle around the Nv-ring. For example in the 48-node ring the following would be a stable pulse train (000000000000000000000000000000001010 101 010101). Many other possibilities exist. By the necklace function there are an enormous number of stable limit-cycles or states possible.

Hopfield [3] and Hopfield and Hertz [4] suggested that a subthreshold neural cell potential might always be present as an oscillation of some frequency. More recently this has been also described in some detail by Gerstner [18]. If a neuron receives a pulsing or sinusoidal input in addition to some, perhaps sinusoidal (or otherwise) subthreshold input, then a neuron may fire only at some threshold value or when the input is in-phase with the subthreshold oscillations. From the behavior of the Nv-neurons outlined in Eqs. (3) and (4) it should be obvious that we can emulate the same type of behavior in our Nv-rings. To test this conjecture we applied a d.c.-offset voltage and an a.c. signal to the ring of 48-nodes. The results are shown in Fig. 15. The first figure shows the effects from 1.0 V offset and a 25 Hz 1.3 V peak-to-peak sine wave signal as input to the ring. The second shows the effects of 1.0 V offset with an a.c.-modulated signal of 100 Hz carrier wave and 15 Hz modulations. In both cases the neurons in the Nv-ring fire during the same phase in the a.c. signal. This is exactly the same type of behavior described by Hopfield et al. for integrate and fire neurons.

3. Results and discussion

In the first part of the paper we focused on the behavior of Nv-neuron circuits. Several important points were described. When the neuron is in the resting state, it is logic high. When the neuron input goes from high to low logic there will be no response on the output. But during this time the capacitor will charge. When the capacitor is charged it will produce a spike at the input to the inverting Schmitt trigger. At some threshold in that spike (depending on the type of logic utilized) it will force the capacitor to discharge and induce logic low at the output of the neuron. At this point the neuron is firing. The pulse width will be determined by the discharge time of the capacitor. When the capacitor has discharged, the neuron output will again go to logic high and return to a resting state. As a consequence of this behavior, short duration pulses will propagate through a network or ring faster than long duration pulses. So the neuron with the shortest tau will have a significant impact on the overall network or ring dynamics. In the case of rings these short duration pulses will interfere with longer duration pulses and produce phase noise that will eventu-

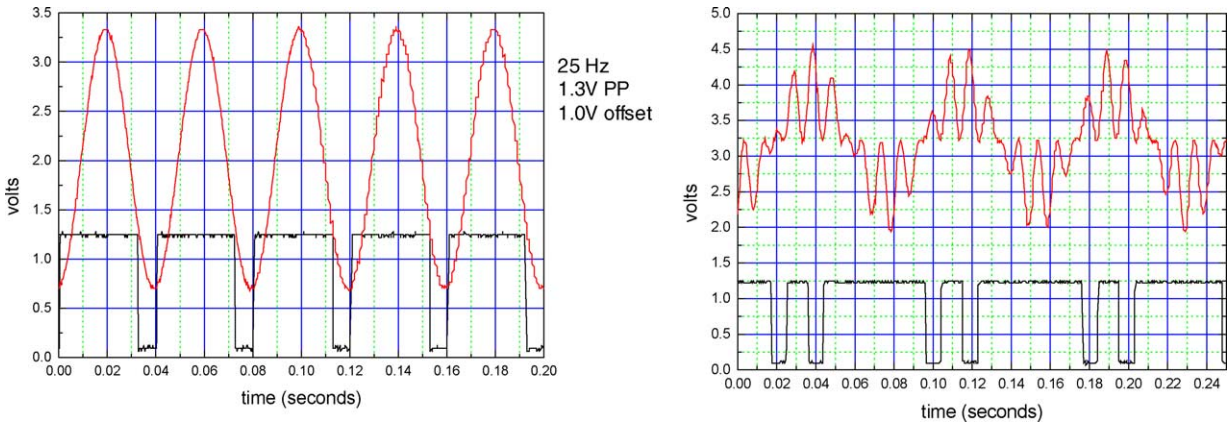


Fig. 15. Experimental results demonstrating threshold behavior.

ally result in the ring circuit settling to a stable limit-cycle. But new pulses injected into the ring can upset this limit-cycle and force the dynamics to a new limit-cycle.

We found that the necklace function adequately describes the dynamics of rings of even numbers of Nv-neurons. The necklace theory is not supported by rings of odd numbers of Nv-neurons. This non-support is likely due to typical phase noise generated in ring oscillators of odd numbers of nodes.

The full dynamics has been explicitly described for the 4-, 5- and 6-node Nv-rings. In addition we have described the attractor diagrams (Fig. 10) for 4-, 6-, 8- and 10-node machines. The attractor diagrams represent the limit-cycles for different initial conditions. Except for the leaf-nodes in the diagrams the indicated limit-cycles are stable unless perturbed by external pulses. As the size of the rings increase we find that the perturbations can result in further limit-cycles. For example, a 10-node Nv-ring will oscillate between the three cycle (73–165–341) but only when perturbed, will it change to a new limit-cycle within this super-cycle. The probability of change has not been measured, but from our observations it is clear that probabilistic automata could be used to model the dynamics of larger systems.

Looking at the attractor diagrams we could begin to assemble larger systems with more complex dynamics. A 6- and 8-node machine could be coupled together such that when the 6-node machine enters state (9) or (21) it will force the 8-node machine to enter state (85). If this was further connected to a 10-node machine it could force it to then enter state (341).

These systems can also be utilized as hybrid analog-digital information processors. If we initialize the 4-node machine with (0, 3, 4, 7, 9, 11, 12, 13, 14, 15) it will settle to the limit-cycle (1) and if we initialize the same machine with (1, 2, 6, 8, 10) it will settle to the limit-cycle (5). Unlike the limit-cycle to limit-cycle transitions, these transitions do not require perturbations so the transition probability is 1. Each limit-cycle is metastable. In the 6-node machine, for example, the limit-cycle (1) may be the stable state but reinitializing or injecting pulses will not necessarily force it to state (21). It may in fact go to state (9) but only via state (21). As pointed out above, we do not have a mapping of the input to output in probability, except to say that the leaf-nodes will, with probability one, settle to the indicated limit-cycles. This settling time is very fast and we did not measure it. It is reasonable to see that by the necklace theory the ‘processes’ will ‘bounce’ away from each other as they travel around the ring. So the entire settling time would be on the order of the number of nodes times the *RC* time constant.

The ‘settled’ limit-cycle is stable and this stability is based on the necklace theory and the number of ‘processes’ in the loop. This is not dependent on the specific chip. For example we constructed the same type of loop, e.g. 10-node, with different 4093 packages and observed the same limit-cycle and super-limit-cycle behavior. The only variation, which we did not attempt to measure, could be the time variation between *RC* elements in the loop. This would result in subtle variations in the settling time for the system to enter a limit-cycle.

Since the limit-cycles are stable this suggests memory storage, where the memory states are the pulse patterns in the Nv-ring. Large rings can support larger numbers of stable limit-cycles and therefore a larger number of memory states. Using the integrating behavior of coupled Nv-rings it is possible to generate new and unusual stable patterns (Fig. 14) by constructive and destructive interference of pulses within the rings and it is possible to build networks of Boolean computation circuits (Fig. 12).

Lastly, we have demonstrated that we can exploit the threshold behavior outlined in Eq. (4). Nv-rings can be subjected to a underlying bias and lower voltage signals or amplitude modulated signals can result in firing of the neurons in-phase with the input signal or the underlying bias. This and the integrating dynamics are the same type of behavior attributed to spiking neuron models.

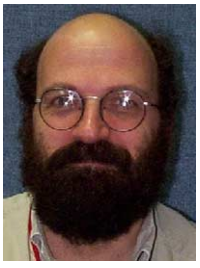
Nv-rings have all the necessary properties to span most computational functions, minimally, elegantly, and robustly, and the field is wide-open for investigation.

Acknowledgements

We thank, Matt Mosis for sharing with us an unpublished manuscript [8], and an anonymous reviewer who improved the readability of the paper.

References

[1] B. Hasslacher, M.W. Tilden, Living machines, Robotics and Autonomous Systems 15 (1995) 143–169.
[2] B. Hasslacher, M.W. Tilden, Theoretical foundations for nervous networks and the design of living machines, in: Proceedings of the Workshop on Control Mechanisms for Complex Systems: Issues of Measurement and Semiotic Analysis, SFI, 1997.
[3] J.J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation, Nature 376 (1995) 33–36.
[4] J.J. Hopfield, A.V.M. Hertz, Rapid local synchronization of action potentials: toward computation with coupled integrate-and-fire neurons, Proceedings of the National Academy of Sciences 92 (1995) 6655–6662.
[5] P. Horowitz, W. Hill, The Art of Electronics, 2nd ed., Cambridge University, New York, 1989.
[6] M.H. Jones, A Practical Introduction to Electronic Circuits, 3rd Ed., Cambridge University, New York, NY, 1995.
[7] W. Maas, C.M. Bishop, Pulsed Neural Networks, MIT, Cambridge, MA, 1999.
[8] M. Moses, Minimalist approach to design of walking robots, unpublished manuscript, 2001.
[9] A. Murray, L. Tarassenko, Analogue Neural VLSI: A Pulse Stream Approach, Chapman and Hall, London, 1994.
[10] F. Ruskey, Information on Necklaces, Lyndon Words, De Bruijn Sequences, 1995. http://www.theory.cs.uvic.ca/~cos/inf/neck/NecklaceInfo.html.
[11] R. Sarpeshkar, Analog versus digital: extrapolating from electronics to neurobiology, Neural Computation 10 (1998) 1601–1638.
[12] R. Sarpeshkar, M. O'Halloran, Scalable hybrid computation with spikes, Neural Computation 14 (2002) 2003–2038.
[13] S. Still, M.W. Tilden, Controller for a four legged walking machine, in: Smith, Hamilton (Eds.), Neuromorphic Systems Engineering Silicon from Neurobiology, World Scientific, Singapore, 1998, pp. 138–148.
[14] Tilden, US Patent 5 325 031 (28 June 1994).
[15] J.H. van Lint, R.M. Wilson, A Course in Combinatorics, Cambridge University, Cambridge, 1992.
[16] E. Weisstein, Necklace, 1999. http://mathworld.wolfram.com/Necklace.html.
[17] E. Weisstein, Totient Function, 1999. http://mathworld.wolfram.com/TotientFunction.html.
[18] W. Gerstner, Spiking neurons, in: Mass and Bishop, Pulsed Neural Networks, MIT, Cambridge, MA, 1999, pp. 3–53.



Edward A. Rietman has B.S. degrees in physics, chemistry and a BA in philosophy. He has an M.S. degree in materials science and a Ph.D. in physics. He is currently working on an M.S. degree in computational and theoretical biology.

He spent 19 years at Bell Labs working on solid-state physics, neural network hardware and AI applications for control in CMOS manufacturing. He has published several books on neural networks, chaos, parallel computing, artificial life and nanotechnology. In addition he has authored over 100 technical papers and dozens of patents. He is currently running a research group on adaptive optics and artificial intelligence. His current research interests are on methods to exploit the physics of materials for computation, optical computing, and cellular automata.

Mark W. Tilden was educated in Systems Engineering at the University of Waterloo (Canada) and did a Masters in Media Electronics. He built and ran his own TV studio during the 1980s and has been involved in many projects including industrial machining, engineering education, computer electronics, autonomous mine removal, virtual reality systems, satellite and space systems design, artificial intelligence and neuromorphic initiatives, and is the founder of the International BEAM Robot Olympic Games. Until recently he was a senior physicist at Los Alamos National Laboratory developing analog nervous network control applications for social computation studies. Currently he is a board member of his own non-profit science startup (Institute for Physical Sciences, Washington, DC), licenses technology to an on-line robot education business (http://www.solarbotics.com), but spends most of his time in Hong Kong as head of Research and Development for WowWee (Hasbro) toys. He has multiple patents, books, and publications and several dozen web sites dedicated to his efforts. He has numerous awards for his community education and research work (including an honorary Ph.D.), and has held several world records in machine design. Hundreds of his "living" robots are on constant display in museums around North America, and in syndicated reruns on documentary TV.

Manor Askenazi received an M.S. in computer engineering from Imperial College, London. He has pursued a career in computational research and development, working in a wide variety of domains: from supercomputer manufacturing at Thinking Machines Corporation in Cambridge, where he developed the Performance Analysis Library for the Computational Science Group, through complexity science at the Santa Fe Institute, where he helped develop the Swarm simulation package, through to his current position as a bioinformatics scientist at Microbia Inc. in Cambridge, where he is responsible for the integration and analysis of transcriptional, metabolic and phenotypic profiles generated by Microbia's drug discovery and development programs.