Communicated by Christof Koch

The Storage of Time Intervals Using Oscillating Neurons

Christopher Miall
*King's College Research Centre, King's College,
Cambridge, CB2 1ST, England*

A mechanism to store and recall time intervals ranging from hundreds of milliseconds to tens of seconds is described. The principle is based on beat frequencies between oscillating elements; any small group of oscillators codes specifically for an interval equal to the lowest common multiple of their oscillation periods. This mechanism could be realized in the nervous system by an output neuron, excited by a group of pacemaker neurons, and able to select via a Hebbian rule a subgroup of pacemaker cells to encode any given interval, or small number of intervals (for example, a pattern of pulses). Recall could be achieved by resetting the pacemaker cells and setting a threshold for activation of the output unit. A simulation is described and the main features of such an encoding scheme are discussed.

1 Introduction

How do we perceive and recall intervals ranging from tenths of seconds to seconds? It is these sorts of time intervals that underlie much of our behavior, as well as our appreciation of rhythms and music, and yet the way in which our nervous system encodes and stores time is unknown. There must be ways that time can be mapped onto neurons that allow them to be activated at specific intervals, or for specific durations, in response to internal or external time cues.

Several neuronal mechanisms have been suggested over the years but they seem most suitable for intervals an order of magnitude or more shorter than those of interest here. For example Licklider (1951) suggested that a chain of neurons could form a simple delay line, with each synapse adding a discrete delay to a transmitted signal. Each neuron in the chain might contribute only a few milliseconds delay, however, so intervals of the order of seconds would necessitate unwieldy chains hundreds of cells long.

Another suggestion was to make use of the conduction delays in axons, so that time is mapped as distance along an axon (Jeffress 1948). Such a mechanism is indeed found to underlie detection of interaural time differences (Carr and Konishi 1988), but the delays achieved are in the range of 200–400 μsec. Braitenberg (1961) suggested that conduction delays in the parallel fibers of the cerebellum could be used to cross-correlate their activity patterns, and Longuet-Higgins (1989) has proposed that a similar scheme could be used to store temporal correlations between pulse trains. Again, it is unlikely that axons will be found of sufficient length and low conduction velocities to encode intervals of more than a few tens of milliseconds.

A third possibility is that oscillatory pacemaker neurons could be used directly to encode time, either by choosing pacemakers with suitable oscillation periods from a population of cells, or through plastic changes to the period of oscillation of each cell (Torras 1986). An interesting development of this idea is that a group of cells could combine to encode a temporal waveform, by forming its Fourier series. However, this scheme may have limits to the intervals (or frequencies) that it could store. Choosing among a population of pacemakers would require a very wide distribution of oscillation frequencies to be selected from, while changing the oscillation period would require an individual cell to adapt its frequency over a wide range. The idea of using oscillators to store an arbitrary temporal sequence was also fundamental to Longuet-Higgins' holophone (Longuet-Higgins 1968). However, it required a bank of oscillators with a wide range of frequencies represented, and also required these frequencies to be more-or-less equally spaced. Another difficulty with the scheme was that in order to store long intervals, an oscillator was required with an oscillation period equal to the interval to be stored. [Longuet-Higgins (1968) suggested a possible solution here, in that a long temporal pattern could be stored by a mechanism of reentrant pathways, effectively setting up a chain of shorter output sequences.] Schemes based on oscillators might also suffer a trade-off between accuracy and period, as cells with very long oscillation periods would have slowly changing membrane potentials, and hence one would expect some scatter of the moment at which spiking initiates in each cycle.

In this paper a related mechanism that seems to avoid these pitfalls is presented. One can make use of "beating" between pairs or groups of oscillators to store time intervals. Thus, a group of pacemaker cells, even if they have quite similar oscillation frequencies, can be used to encode a wide range of time intervals, and to recall the interval at a later time. Some of the possibilities of this scheme are described here, and the relationships between numbers of cells, their inherent accuracy, and the accuracy with which they can encode time are discussed.

2 Oscillators and Beating

Consider a group of oscillators (pacemaker neurons), each with a slightly different frequency of oscillation, and each spiking for a brief part of each cycle (Fig. 1A). The beat frequency of any pair of these oscillators is then the frequency at which they spike simultaneously: looking at oscillators 5 and 6 in Figure 1B, it can be seen that they spike together only three times, as indicted by the stars. Thus, their beat frequency is much lower than their intrinsic oscillation frequency; it is given by the difference between the frequencies of the two cells. So if this difference is small, the beat frequency is low and the beat period long. Considering a group of cells, rather than just a pair, the beat frequency is given by the lowest common multiple of the periods of their oscillations. Although this statement is strictly true for oscillators active for only one instant in each cycle, partial coincidence of activity may occur at shorter intervals if the pacemakers are active for some significant portion of each cycle, as the group then drifts in and out of synchrony with a time course given by the lowest common multiple.

Now, to encode any particular time interval, for example, the interval between time t_0 and time t_1 indicated by the open bars in Figure 1B, one could select the subgroup of cells that is active at both times, in this case cells 1, 3, and 6. The chosen set of oscillators fits the time interval specifically: they spike together only twice after t_0, at t_1 and t_2. So to recover this interval at some later point, one need only reset all the oscillators, and wait for the selected group to spike simultaneously. One way this could be achieved would be to activate "Hebb" synapses between the selected pacemakers and a postsynaptic unit, and then set a threshold for activation of that unit equal to the number of cells selected. As will be shown later, with larger populations of oscillatory cells, the numbers of cells selected to encode any given interval need not be known accurately to set a suitable threshold.

In the example shown, the three selected cells are again almost synchronous at time t_2, which is exactly double the original period $t_0 - t_1$. Thus, the same mechanism can be used to oscillate with a given period. The three are not exactly synchronous at time t_2 because they were selected imprecisely: the diagram has a limited resolution and is not accurate enough to distinguish the marginal difference in phase between cell 6 and cells 1 and 3 at the selection time t_1.

A final point to make about this example is that the pacemakers were all started simultaneously at time t_0. This is only a convenience, and the same procedure can be used in a larger group of pacemakers that is not synchronized. For instance, if we were to select cells at times t_1 and t_2 without resetting the population, we would end up with a similar result, having in this case chosen only units 1 and 3 (again given the limits of resolution imposed in the diagram). However, resetting the whole population before the encoding process significantly reduces the numbers of cells needed to accurately code any interval, and is vital if more than one interval is to be stored (see later).

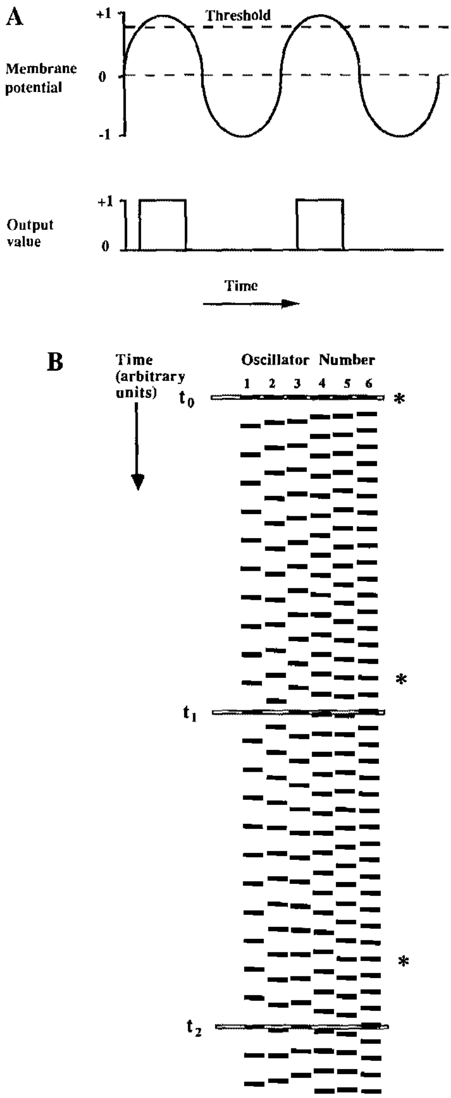

Figure 1: (A) A schematic diagram of the behavior of a single pacemaker. The membrane potential oscillates sinusoidally (upper record), and the pacemaker is considered active ("spiking") if a threshold level is exceeded. In the simulations presented in this paper, the output of such a pacemaker was taken to be +1 or zero, as shown in the lower record. (B) Beating among a group of six oscillators. Activity in each oscillator is indicated by a short black bar. The stars on the right indicate times when units 5 and 6 are synchronous. The open bars highlight those units active at selection times t_0 and t_1; t_2 is twice the interval $t_0 - t_1$.

3 A Simulation

To demonstrate the potential and the limits of this idea, the basic mechanism indicated in Figure 1 was simulated on a computer. A population of up to 500 pacemaking units was defined, with oscillation frequencies between 5 and 15 Hz chosen using a random number generator to give an average frequency of 10 Hz (Llinás 1988) and a standard deviation about the mean of 1.6 Hz. The iteration rate for the model was 60 Hz, allowing good resolution of even the highest frequency pacemakers. Each oscillator's membrane potential was considered to be a pure cosine function ranging in amplitude ±1, and an activity threshold was set to delimit the "active" (spiking) and "inactive" parts of its cycle (Fig. 1A). Thus, if the threshold was set at 0.0, the unit was considered to be active for half its cycle, and if the threshold was set at +0.9, the unit was active for 15% of its cycle. The output of each pacemaker was zero if the membrane potential was below the activity threshold, and +1 if above. This represents continuous "spiking" activity above threshold, which is obviously somewhat unrealistic. A more realistic simulation in which spike rate is proportional to membrane potential above the activity threshold could be envisaged. However, the scheme is not dependent on this property, and can be realized even if each pacemaker emits a single spike at each cycle (see below).

All pacemaking units synapsed onto a single output unit via Hebbian synapses taking values between +1 and −1; hence the output unit received input of zero or between ±1 from each pacemaker at each time step. The total input from the pacemakers was summed, and displayed as a time histogram (see Figs. 2–5). Now, to store any specific interval, $t_0 - t_1$, first all pacemaker units were synchronized at t_0 (as in Fig. 1B). Those above threshold at time t_1 were noted, and the strength of their synapses onto the output unit set accordingly. In most of the simulations reported here, the selected synapses were set to a strength of +1 and all others set to zero; exceptions to this general rule will be pointed out later. To test the specificity with which the selected units stored each interval, all the pacemakers were again resynchronized, and the activity of the output unit monitored (Fig. 2). The model was tested with a population of between 10 and 500 pacemakers, with intervals ranging from 200 msec to 10 sec, and with the activity threshold of the pacemakers ranging from 0 to +0.999.

3.1 Timing Specificity. Figure 2 shows the total pacemaker output activity received by the output unit when 250 pacemakers were used at three activity thresholds. In each graph the x axis has been scaled to the stored interval and the y axis to the number of pacemakers selected to encode that interval. Hence the output unit is maximally activated at the extreme right of each graph. Comparing the maximum output (that is,

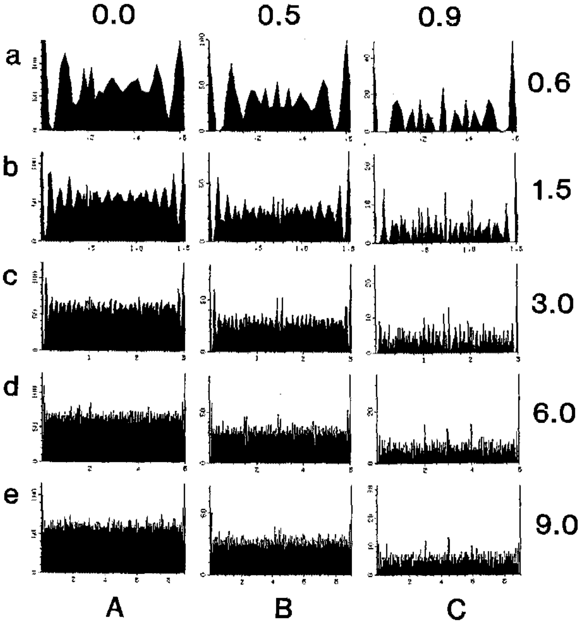

Figure 2: Records of the output unit's activity in recall of intervals. Each graph shows the total synaptic *input* received by the output unit at each iteration of the model. In each the x axis has been scaled to the stored interval and the y axis to the number of pacemakers selected to encode that interval. Hence the output unit is maximally activated at the extreme right of each graph. The exact numbers of cells selected are not important; only the relative height of the peak at the extreme right of each graph to the next highest peak is critical for discrimination of the correct interval. Hence, a threshold for the output unit would be needed to discriminate activity at this time (t_1) from all other times, but is not indicated here. Recall is shown of 5 intervals from 0.6 sec (Row a) to 9 sec (Row e); the figures to the right indicate the interval stored (in seconds). The simulation included 250 pacemakers, with activity threshold α = 0.0, 0.5, or 0.9 (columns A–C; see figures across top).

the number of pacemakers active at time t_1) to the next highest peak of activity gave a measure of the reliability of the output, its specificity for the stored interval. The next highest peak was generally found at half of the stored interval, as in the bottom right of figure 2, or was separated from the main peak by 100 msec, as in the left column of figure 2. The latter is because the population had an average frequency of 10 Hz; thus, while all the selected cells are active at time t_1, many will also be active 100 msec earlier. Three things should be pointed out in figure 2. First, specificity increased as the activity threshold increased (in columns from left to right). Second, specificity increased as the time interval increased (in rows from top to bottom), although in general the specificity was good for all intervals above 1 sec. Third, the output was elevated at some fractions of the specified interval; for example, peaks can be seen at one-third, half, and two-thirds the stored interval.

As discussed in relation to figure 1, the accuracy with which units are selected (given by the activity threshold level) also determines the reliability of further output activity at multiples of the stored interval (that is, t_2, t_3 etc.). In these simulations the activity threshold needed to be at least 0.5 to allow reliable discrimination of a second peak, and even higher if multiple outputs were to be reliable (Fig. 3). If the threshold was too high, however (0.99 or more), very few units were selected for any

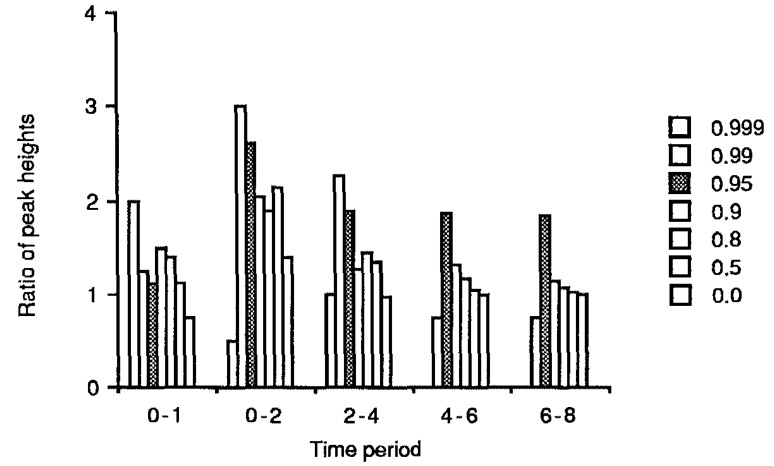

Figure 3: The specificity for output at multiples of the original interval. A population of 250 pacemakers was used to store an interval of 2 sec, with activity thresholds varied from $\alpha = 0.999$ to 0.0. The ratio of output activity at multiples of 2 sec (1, 2, 4, 6 etc.) to the next highest output level (that is, the ratio of peaks, or "output specificity") was then calculated and is displayed; only values above 1 allow discrimination of the desired interval. When $\alpha = 0.999$, too few units were selected and discrimination of even the original interval was poor. At $\alpha = 0.95$ (stippled bars), even activity at cycle # 4 (8 sec) could be accurately detected. As α dropped further, discrimination of repeated cycles fell off.

one interval. Discrimination of later peaks was best if the threshold was set as high as possible while still allowing about 20 units to be selected.

3.2 Timing Resolution. The average frequency of the pacemakers was 10 Hz, and the activity threshold directly determined the resolution of the model. For example, if the threshold was set at zero, the basic resolution was $\approx$ 50 msec (50% of the cycle; Fig. 2, left column). In other words at this threshold the model could not be used to distinguish intervals that differed by less than 50 msec. If the activity threshold was high (0.9 or above; Fig. 2, right column) resolution was limited only by the iteration rate of the model (16 msec).

The shortest interval that could be stored was of course equal to the shortest period of any pacemaker. However, unless the activity threshold was high (above 0.5), the output signaling intervals under 1 sec could not be reliably distinguished from peaks 100 msec before or afterward (Fig. 2, top row). Thus, the temporal resolution at low activity thresholds was ±100 msec, but was arbitrarily high for high thresholds.

The longest interval that can be stored is difficult to specify. It depends on the number of pacemakers, their activity thresholds, and the precise distribution of oscillation frequencies within the group. For any given group, there must be some interval that cannot be distinguished from another shorter interval that is a fraction of the first. For example, if we consider just units 5 and 6 in figure 1B, it would be impossible to code for the interval indicated by the lowest star, since these two units also code for half that interval, indicated by the middle star. In simulations of 250 or 500 pacemakers, the upper time limit seems to be at least 20 sec, although it has not been possible to test every interval below that value.

3.3 Number of Pacemakers. The accuracy and specificity with which each interval can be stored are also related to the number of units used to encode it. The proportion of the whole population of pacemakers expected to be selected at any time t_1 is equal to the proportion of time each unit is active. Thus, if the units are active for 15% of each cycle (an activity threshold of 0.9), then about one-sixth of the population will be selected to store any one interval.

The simulations indicate that the best results are achieved when at least 15–20 units are selected to encode an interval. If this number drops below 5 or 6, the specificity becomes less predictable: some intervals are resolved clearly, others are coded ambiguously, and some intervals cannot be encoded at all, because no units are active at time t_1. Thus, the minimum population size should be approximately $20/p$, where p is the proportion of time each pacemaker is active. For these units, with sinusoidal membrane potential, $p = \cos^{-1}(\alpha)/\pi$, where α is the activity threshold. At the highest threshold used ($\alpha = 0.999$), each pacemaker was active for less than 1.5% of the cycle. Hence the output of each could be thought of as a single spike, and although the scheme was still valid (not shown), the population of pacemakers needed to reliably encode any arbitrary interval is high (about 1400, by the equations given above).

3.4 Setting a Threshold for the Output Unit. As indicated in figure 2, the output unit is continuously barraged by inputs, and an threshold must be defined to distinguish the input at t_1 from all others. An appropriate threshold can be set without need to know exactly how many units are selected at any one time. In general, about twice the number of units are active at t_1 than at the next highest peak, and as long as the population of pacemakers is large (for example, 100–200), then the number selected at any time t_1 will be roughly constant (as discussed above). A suitable value for the output threshold would be $\beta = 0.75m \cdot \cos^{-1}(\alpha)/\pi$, where m is the total population size and α the activity threshold.

3.5 Inhibitory Pacemakers. Another way to increase the specificity for each interval is to include a number of pacemakers that make inhibitory synapses onto the output unit. If inhibitory units can be selected that are inactive at times t_0 and t_1 then as a group they will tend to be active at all other times, partially inhibiting the output unit and increasing its specificity. Because of the asymmetry between the percentage time each pacemaker is active or inactive, this mechanism will select a set of inhibitory units with a different distribution of oscillation frequencies from the excitatory set. It is therefore not equivalent to merely changing the synaptic strengths of the excitatory units, or to altering the output unit's threshold.

Figure 4 shows a simulation in which half the pacemakers were inhibitory. The selection rule was modified so that all excitatory units active at times t_0 and t_1 had, as before, synapse strengths of +1; all inhibitory units inactive at time t_1 had synapse strengths of -1 and all others had synapse strengths of zero. It can be seen that the inhibitory units had two effects. First, the output unit was inhibited at most times between t_0 and t_1; this was most pronounced when the activity threshold was high (Fig. 4d and e) because then the inhibitory units inactive at t_1 greatly outnumbered the excitatory units active at that time. Thus, the threshold for the output unit could be set at zero without regard for the numbers of cells in the population. Second, activity at higher harmonics (at half or one-third of the original interval) was diminished because of the different distribution of frequencies in the excitatory and inhibitory groups.

3.6 Storing Multiple Intervals. The basic mechanism can be used to store more than one time interval simply by selecting additional pacemakers that encode for each additional time (Fig. 5). The threshold for the output unit, β, need not be altered since approximately the same number of units will be chosen for each interval. Figure 5 has been scaled according to the total number of cells selected, so to make the output threshold clear a line has been drawn at $\beta = 70$ in each graph, which is the average number of units chosen for each interval. However, since the background activity is proportional to the total number of units, and not the number active at each selected time, only three or four intervals can be stored before the peaks can no longer be distinguished. Further-

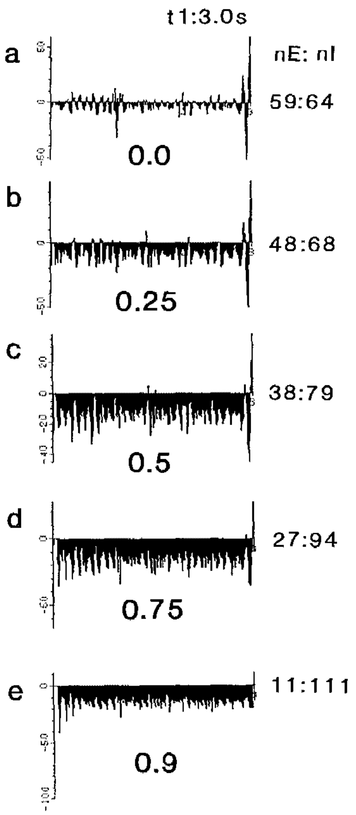

Figure 4: Records of the output unit's input activity in recall of a 3-sec interval when inhibitory units were included in the simulation. The format used is the same as in figure 2. Half of the 250 pacemakers were inhibitory; the numbers of excitatory (nE) and inhibitory units (nI) selected are given on the right of each graph. The activity threshold, α, was varied from 0.0 to 0.9, as indicated below each graph.

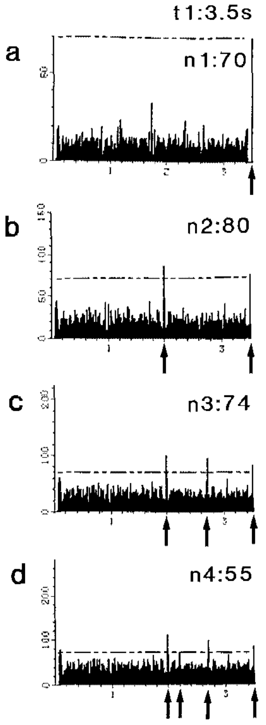

Figure 5: Records of the output unit's input activity in recall of multiple intervals. The format used is the same as in figure 2. In graph **a**, an interval t_1 of 3.5 sec was stored; in each subsequent graph one extra interval was added (at 2.0, 2.7, and 2.2 sec respectively, as indicated by the arrows). Five hundred excitatory pacemakers were simulated with activity threshold $\alpha = 0.9$; the numbers selected at each interval are given above each graph. A threshold for the output unit at $\beta = 70$ has been indicated in each graph.

more, recover of multiple inputs can be achieved only by resetting all pacemakers, and so a pattern of pulses cannot easily be repeated several times.

3.7 Synapse Strengths. The simulations discussed so far have used integer synapses taking values 0 and 1, or 0, +1, and −1. In a biological framework it might be more realistic to assume that synapses could take values between 0 and 1, and would move only gradually between these two limits on repeated presentations of a particular interval. In other words, a learning rule might increase the strength of selected synapses relative to the others only gradually. Temporal specificity is rather poor in such circumstances. If the number of selected pacemakers is small relative to the whole population (that is, the activity threshold is high), then the output unit receives a barrage of input from the unwanted units. It is therefore difficult to distinguish the peak at time t_1 unless the selected units have synapse strengths considerably greater than the others. Alternatively, the activity threshold must be low, so that the number of selected units is of the same order as the number of unwanted units, but that introduces limits to performance which were discussed previously.

However, it might be possible to stop the unwanted pacemakers oscillating at all, perhaps by temporarily altering their membrane properties, leaving only the selected group active. This is functionally equivalent to setting the synapse strengths to zero as in these simulations.

4 Conclusions

The scheme presented here allows neurons with relatively high oscillation frequencies to encode intervals much greater than their own oscillation period. By selecting groups of pacemakers from a population, intervals between pairs of pulses, or short patterns of pulses, can be stored and recalled. The scheme is surprisingly robust, and even when individual pacemakers are active for 50% of each cycle, temporal resolution is good.

Three weaknesses of the scheme are apparent. First, the synapses onto the output unit need to be considerably stronger from the selected group than from other pacemakers. Second, the selected units need to be reset synchronously to most accurately recall the stored interval. Third, the pacemakers need to maintain a stable oscillation frequency for the duration of the stored interval. Actually, this is not strictly so, as the scheme would still work if the pacemakers *reliably* drifted or swept through a frequency range after being reset; the critical point is that their behavior must be repeatable in storage and in recall. None of these problems seems insurmountable, but since we are ignorant of the site of timing operations in the brain, it remains to be seen whether suitable populations of pacemakers will be found to realize this scheme. Pacemaking units are found in a number of brain sites, and the figure of 10 Hz chosen for these simulations is biologically reasonable (Llinás 1988). It is therefore interesting to see that the temporal resolution of the simulations is of the same order as that found in psychophysical tests (that is, about 50 msec, see review by Pöppel 1978). Pöppel also reviews evidence for a qualitative difference in the perception of short intervals to those greater than about 2–3 sec. It would be interesting to find the longest interval that could be unambiguously stored by the proposed scheme. However, with the numbers and threshold levels chosen here, the limiting interval seems to be at least 10 sec.

Finally, it should be mentioned that the fundamental principle here is of beating between oscillators, and is not dependent on the underlying properties of the pacemakers units. The oscillatory components could therefore be realized as individual pacemaker neurons, as simulated, or by entrained groups of pacemakers, or by reverbatory circuits containing several neurons. In the latter cases, any one cell in the group or circuit could make a synaptic connection to the output neuron.

Acknowledgments

This work was supported by King's College Research Centre. I thank Graeme Mitchison for interesting discussions about these ideas, and the reviewers for their useful suggestions about the manuscript.

References

Braitenberg, V. 1961. Functional interpretation of cerebellar histology. *Nature (London)* **190**, 539.

Carr, C.E., and Konishi, M. 1988. Axonal delay lines for time measurement in the owl's brainstem. *Proc. Nat. Acad. Sci. U.S.A.* **85**, 8311–8316.

Jeffress, L.A. 1948. A place theory of sound localization. *J. Comp. Physiol. Psychol.* **41**, 35–39.

Licklider, J.C.R. 1951. A duplex theory of pitch perception. *Experientia* **7**, 128–134.

Llinás, R.R. 1988. The intrinsic electrophysiological properties of mammalian neurons: Insights into central nervous system function. *Science* **242**, 1654–1664.

Lonquet-Higgins, H.C. 1968. Holographic model of temporal recall. *Nature (London)* **217**, 104.

Lonquet-Higgins, H.C. 1989. A mechanism for the storage of temporal correlations. In *The Computing Neuron*, R. Durbin, C. Miall, and G. Mitchison, eds. Addison-Wesley, Wokingham, England, pp. 99–104.

Pöppel, E. 1978. Time perception. In *The Handbook of Sensory Physiology*. Vol. VIII, pp. 713–729. Springer-Verlag: Berlin, Heidelberg, New York.

Torras, C.I.G. 1986. Neural network model with rhythm-assimilation capacity. *IEEE Syst. Man, Cybernet.* **16**, 680–693.

Received 1 March 1989; accepted 10 April 1989.