IJCNN-SEATTLE-91 Proceedings, vol. II, pp.879–884

A Neural Network Model of Serial Order Recall from Short-Term Memory

Ian Boardman and Daniel Bullock*
Center for Adaptive Systems, Boston University
Boston, Massachusetts 02215

Abstract

We present a neural network that converts a spatial pattern encoding the serial order of a list of items into a corresponding sequence of output events. Performance can be initiated, interrupted and continued under control of a single, independent signal. The network uses recurrent connections to normalize and store node activity in short-term memory, to choose the items in sequence and to modulate the output rate during readout. The latency and duration of output events can model certain empirical trends of human performance in recalling lists of items.

Introduction

Every day, we effortlessly perform tasks of serial order recall from short-term memory, for example when we are told a telephone number and then dial it. As neural modelers, we are interested in the general problem of the representation and control of such serial order processes using neural networks. Work on this problem can contribute to a general understanding of brain mechanisms involved in planned movement tasks such as speaking and to development of artificial systems which perform planned sequential control tasks.

A key paradigm in the psychophysics of human recall from short-term memory (STM) is that of Sternberg, *et al.* (1978, 1980). In this procedure, essentially, the subject quickly memorizes a short list of simple items and then on a signal must accurately recite the list as rapidly as possible. Their experiments revealed dependencies of the latency and duration of output items on list length and on item position within a given list. Both reaction time (time to initiation of readout) and mean item duration (interval between the onset of successive items) grow slower-than-linearly with list length. Within a given list, durations increase with list position, and for longer list lengths, durations may level off or decrease towards the end of the sequence. These trends are observed in speaking one and two syllable words in lists of different or of repeated words, in typing, and in making intentional saccades of the eye to fixed locations (Zingale and Kowler, 1987).

The non-neural "sub-program retrieval" model proposed by Sternberg, *et al.* (1980) to explain these data leaves key aspects of the data unexplained. Under the network paradigm we propose, the empirical trends described above emerge from network dynamics.

*Supported in part by the National Science Foundation (NSF IRI-87-16960).

Model description

One basis for our approach to modeling these data is the neural mechanism for serial order recall described by Grossberg (1978) and Grossberg and Kuperstein (1986). Items are represented by individual nodes, and their sequential order is implicitly encoded by the stored gradient of activation across these nodes: higher activations correspond to earlier items. The field in which the pattern is stored, termed the input field, is a recurrent competitive field (RCF) with an on-center/off-surround connectivity, and it serves as a short-term memory (STM; see for example Grossberg, 1973). Note that there are no adaptive weights in the connections and none are required to store the pattern (see Figure 1).

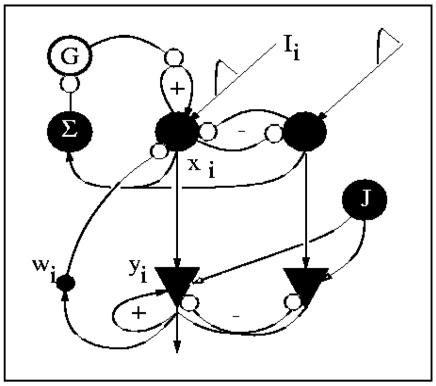


Figure 1: Network representation of the serial order recall model. Excitatory connections have solid arrowheads, inhibitory have hollow circles. Top layer is the input field/STM whose nodes compete with each other and excite their corresponding output nodes, below, gated by arousal J. Input field activity also inhibits a tonic, global inhibitor G which dynamically modulates normalization of the input field. Output nodes feed back strong, persistent inhibition to their source nodes in the input field via interneurons w_i. *cf.* Grossberg and Kuperstein (1986), Figure 9.3.

For the purposes of this article, we shall choose to represent each item of the list, repeated or not, as a separate node. A list of n items, i_j, in the order $i_1, i_2, \ldots, i_n$ is then encoded by the activations x_j in the input field, which are ordered $x_1 > x_2 > \ldots > x_n$. We refer to this pattern of activation as a primacy gradient.[1]

The stored pattern is received by a second RCF that generates the output events and so is called the output field. The output nodes receive specific excitatory inputs in one-to-one correspondence with the nodes in the input field. The pathways mediating the one-to-one mapping are non-specifically gated by a global signal we refer to as arousal, which thus can interrupt or modulate the rate of output. Within the output RCF, faster-than-linear signal functions ensure that a "winner-take-all" competition transpires. The time required for one output node to emerge as the winner is determined jointly by the level of activation of its corresponding input node and the magnitude of the arousal signal. As the winner emerges, its activation becomes sufficient to exceed the (high) threshold for activating a response and for sending a persistent inhibitory feedback signal, via an interneuron, to its own source of excitation in the input field. This process deletes the corresponding item from the stored pattern in the input field, thereby preventing its output from continuing indefinitely. Consequently, the winning node in the output field exhibits a single phase of activity, which is the desired output event. The loss of activation in the input RCF drives this network toward a new equilibrium distribution, and the process of choosing the next winner then begins. This chain of "falling dominos" terminates after the last item is deleted from the input field. The final winner in the output field is left with no competitors and so remains active, holding this "posture" indefinitely.

[1]Other STM patterns are possible; however, our system is capable in general of recalling whatever order is dictated by the order of activation.

Network equations

The serial order recall network of Figure 1 is defined by the following system of equations. The activation of the input nodes x_i change according to:

$$\dot{x}_i = \underbrace{-A_x x_i}_{\text{passive decay}} + \underbrace{(B_x - x_i)[f(x_i - I_i)/S + I_i]}_{\text{shunted excitation}} - \underbrace{(C_x + x_i)[\sum_{i \neq j}^{N} g(x_j - I_j) + h(w_i)]}_{\text{shunted inhibition}} \tag{1}$$

where the excitatory feedback in this field is inhibited by S,

$$S = \begin{cases} G/(\epsilon + \sum_j^N f(x_j)), & J > 0 \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

The output field node activations y_i obey

$$\dot{y}_i = \underbrace{-A_y y_i}_{\text{passive decay}} + \underbrace{(B_y - y_i)(s(y_i) + Jq(x_i))}_{\text{shunted excitation}} - \underbrace{(C_y + y_i)\sum_{i \neq j}^{N} r(y_j)}_{\text{shunted inhibition}} \tag{3}$$

and the inhibitory interneuron activations w_i obey

$$\dot{w}_i = -A_w w_i + (B_w - w_i)t(y_i). \tag{4}$$

The signal functions are specialized according to the pathway as indicated in Table 1. The

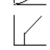
function	pathway description	form
f(x)	on-center (self-excitation)	
g(x)	off-surround (lateral inhibition)	
h(w)	inhibitory feedback	
q(x)	input to y node	
r(y)	off-surround	
s(y)	on-center	
t(y)	input to w node	

Table 1: Signal functions in network equations 1–4. The qualitative input/output characteristic of each signal function is depicted by the miniature graphs in the rightmost column.

forms of the various signal functions are chosen according to the design objectives. The threshold-linear forms are assumed as the simplest first order approximation. They are required for perfect pattern storage in an the RCF input field described by equation 1. We apply faster-than-linear forms (e.g., $f(x) = x^2$) in the output field pathways to ensure "choice" or "winner-take-all" competition (Grossberg, 1973).

There are two methods we chose to generate the serial order encoding. In both cases, we allow the input field to equilibrate such that a stable primacy gradient exists before releasing arousal, although this is not required for correct performance. The first method was analytic and created a set of initial values for the input field. These values are assumed to be the set of activations developed by an external system capable of serial order learning (Grossberg, 1974, 1978); that

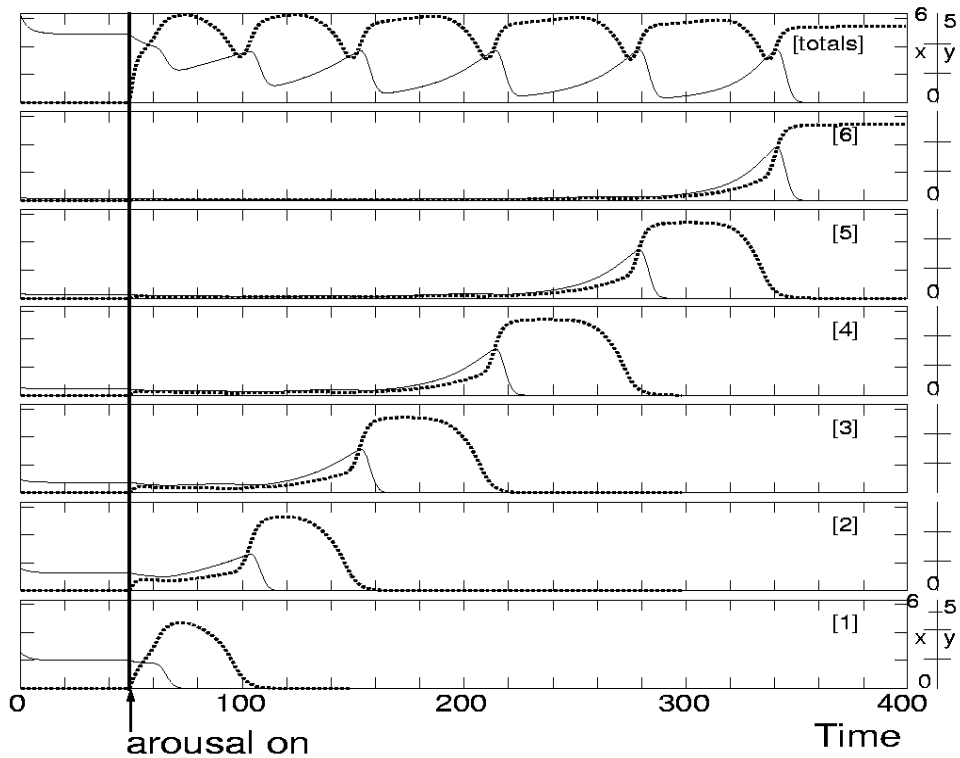


Figure 2: Recall of a 6-item sequence with inhibition of the renormalization of the input field. The solid lines trace over time the input node activities x_i, the dotted lines the output node activities y_i. The uppermost plot shows the total field activity for both fields. From the bottom, the plots show the traces of node activities representing the readout of items. The first plot shows nodes x_1 and y_1, the second shows nodes x_2 and y_2, etc. The input nodes x_i were initialized with a set of values ordered according to primacy. When the x field equilibrated (hatched vertical line, at about $t = 50$), the "go" signal (arousal) was released (set > 0). The values indicated by the ordinate scales on the right are the activities of the nodes, but not typically the values felt by receiving nodes. These node activations would be first transformed by the signal functions of table 1. The output threshold for the nodes y_i was 2.5.

is, by a network that learns the sequence of items and retains the temporal order information (TOI) in its STM. The alternative, procedural method was to successively stimulate the nodes with brief pulses, analogous to a typical list learning paradigm. Primacy in this case is obtained by past-item buffering, implied by the form of the arguments to functions f and g in equation 1. Essentially, this buffering prevents a currently stimulated node from having undue advantage against its previously stimulated competitors, so that later items do not come to dominate over earlier ones. Ensuring a monotone decreasing, or primacy, gradient requires that the stimulus sequence be completed *before* the input RCF reaches its equilibrium potential.

Simulation results

The simulations are carried out by initializing the system, releasing the arousal signal and charting the node activations over time until the last item has been read out. The run can begin with all nodes at rest, or with initial TOI data, and completes with the input STM cleared out, at rest, and one node alone in the output field fully activated. An example simulation appears in Figure 2.

There are several robust properties of the system. The initial pattern is perfectly stored, until the arousal signal is applied, due to the linear signal functions in the input RCF (Grossberg, 1973). There is at any moment only one highly active node in the output field, due to the faster-than-linear signal functions used in that RCF.[2] The input node corresponding to this winner is enduringly inhibited by its associated interneuron, w_i (for clarity, not plotted). After such input node inhibition, a new winner eventually defeats the prior winner in the output RCF because the prior winner has only self-excitation to sustain it, while its biggest competitor has both self-excitation and excitatory input from the input field.

The recall performance can be interrupted and restarted at will using the arousal signal. This is shown by the time course of the input and output field node activities during a paused readout (Figure 3). Note that the output node holds its activation while arousal is off. During the pause,

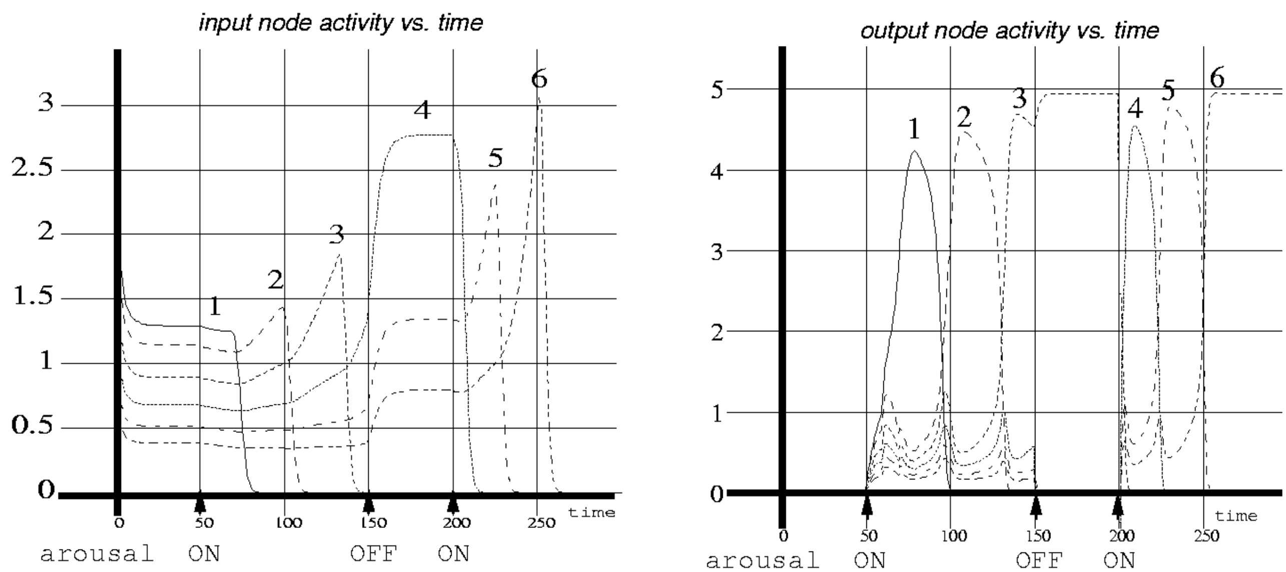


Figure 3: A readout of six items *paused* by turning arousal off in the middle of the sequence. Left: input field. Right: output field.

the input field fully renormalizes, so that when the arousal signal resumes, the sequence continues from where it was interrupted, but at a faster pace.

The goal for these simulations was to model the dependencies of output event latencies and durations on list length and list position found in human subjects. By latency we mean the time from the release of arousal to the onset (above threshold) of the first output event. Mean duration is the difference between onsets of the first and last items, averaged over the number of intervals.

Latency is predicted to increase with list length due to normalization in the input field. The more items, the less activation per node, including the most active. The less activation, the longer it takes to drive the output past threshold. Note, the time required by subjects to process a "go" signal is not modeled by our system. It affects the intercept of the empirical function relating latency to list length.

Latency can also be expected to depend on the onset of arousal, since arousal in our system modulates the feedforward excitation to the output nodes. Many of our simulations apply an abrupt onset of arousal, and consequently, the first winner is chosen quite rapidly, with little

[2] In fact, appropriately biased sigmoidal functions achieve the same objective.

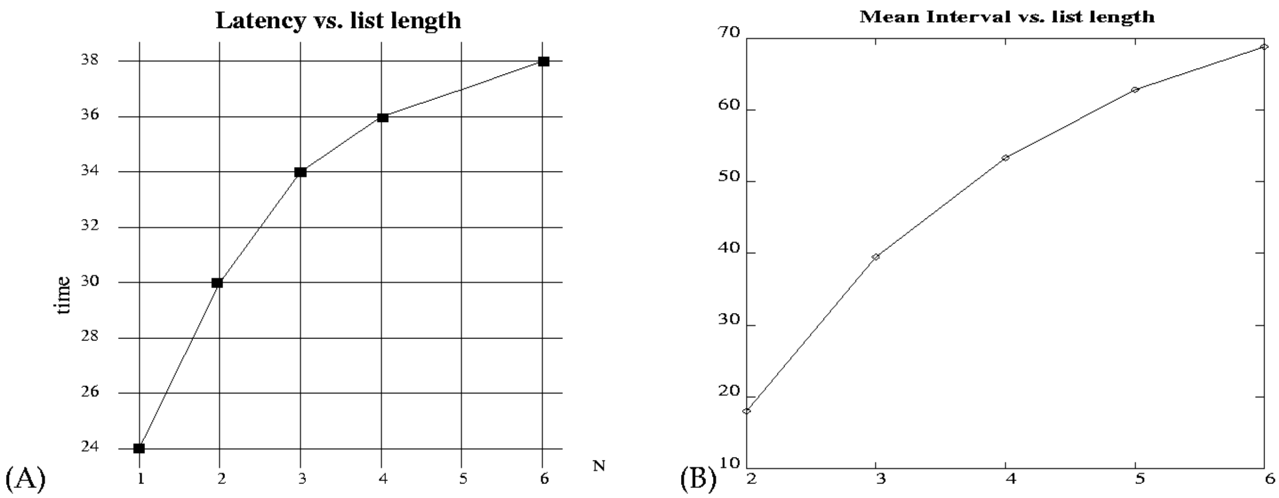


Figure 4: Latency of the first output event (A) and mean item duration (B) vs. list length, using a ramped arousal signal. The trend in (B) is the same if abrupt onset of arousal is used.

dependence on list length. However if the arousal signal ramps up over time, then a latency dependence on list length is revealed, as shown in Figure 4. The slower-than-linear dependence obtained in these simulations is in accord with the Sternberg, *et al.* (1980) and Zingale & Kowler (1987) data. The hypothesis of a ramped arousal signal is natural in a physiological context, and has proven useful in a neural model of variable-speed trajectory formation for elementary movements (Bullock & Grossberg, 1988, 1991).

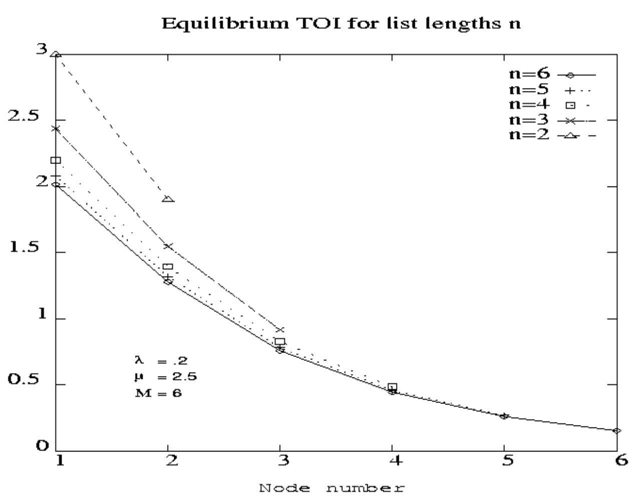


Figure 5: Plots of the equilibrium TOI for various list lengths. The input field was initialized to the values of x_i computed from Grossberg's (1978) eqs. 31-32 (see methods at end of this article), using the parameters shown, and then the input field activations equilibrated to the plotted values. The *ratios* of the initial values do not change.

The dependence of mean item duration on list length also results from normalization of the input field. Due to the storage of a primacy gradient in the input STM, node activity always falls with list position as well as with list length. This can be seen in the graphs of equilibrium TOI patterns in the input STM for each list length (Figure 5A). Since all item durations increase with list length, there is also an increase of the mean inter-item interval; however the converging TOI curves imply that this effect slackens for longer lists. Consequently, mean intervals roll off in a somewhat slower-than-linear manner, as seen in Figure 4B.

We also monitored the system performance by the dependence of the inter-item intervals on list position for each list length. Subjects performing serial order recall are observed to slow down over the course of the list, except at the end of longer lists, when they may begin to speed up. In our system, absolute activity levels in the input field are kept nearly the same throughout readout by suppressing renormalization in inverse proportion to the total field activity. Consequently the time to drive the output over threshold continues to increase over the course of performance. However, the last items of longer lists have greater pressure to reach equilibrium, due to the shunting form of our equations, thus providing the mechanism for accelerating readout in longer lists. These effects are seen in the plots of inter-item intervals vs. list position for each list length (Figure 6).

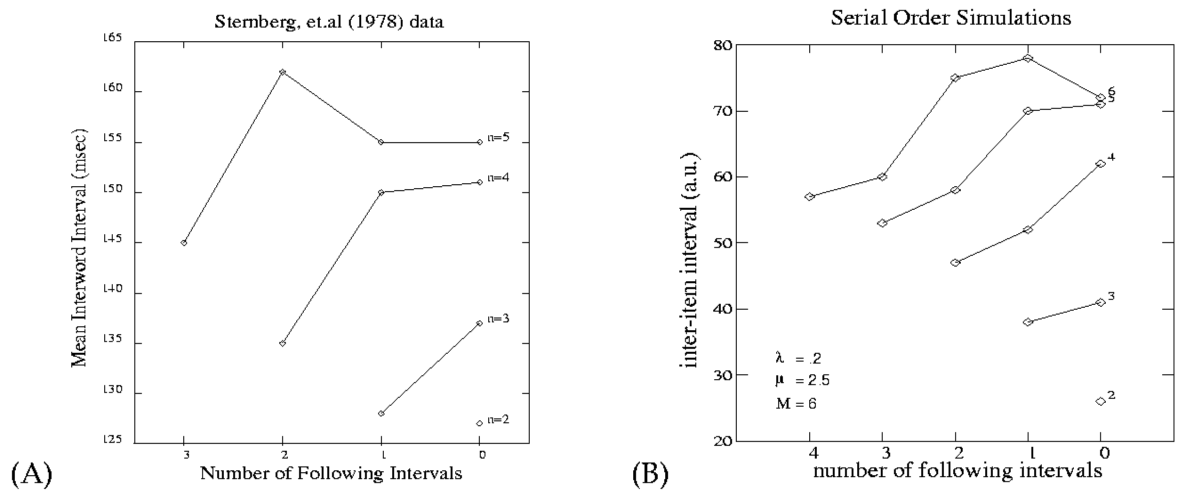


Figure 6: (A) Data from Sternberg, *et al.* (1980). (B) Intervals vs. number of intervals remaining, for various list lengths. The equilibrium TOI input activations at the onset of arousal were those plotted in Fig. 5.

The results compare favorably with those of Sternberg, *et al.* (1980) and Zingale & Kowler (1987). The decline in the last intervals for the longest list length replicates a subtle trend observed in the subject data for recall of planned sequences in speech, typing and saccadic eye movements.

References

[1] Bullock, D. and Grossberg, S. (1988), "Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation," *Psychol. Rev.*, **95**(1): 49–90.

[2] Bullock, D. and Grossberg, S. (1991), "Adaptive neural networks for control of movement trajectories invariant under speed and force rescaling," *Human Movement Science*, **10**(1): 1–51.

[3] Grossberg, S. (1973), "Contour enhancement, short term memory and constancies in reverberating networks," *Studies in Applied Math.*, **52**: 217–257. Reprinted in [6].

[4] Grossberg, S. (1974), "Classical and instrumental learning by neural networks," *Progress in Theoretical Biology*, R. Rosen and F. Snell, eds., pp. 51–141. Reprinted in [6].

[5] Grossberg, S. (1978), "A theory of human memory: self-organization and performance of sensory-motor codes, maps and plans," *Progress in Theoretical Biology* **5** (R. Rosen and F. Snell, eds.), pp. 233–374, Academic Press, New York. Reprinted in [6].

[6] Grossberg, S. (1982), *Studies of Mind and Brain.*, D. Reidel, Boston.

[7] Grossberg, S. and Kuperstein, M. (1986), *Neural Dynamics of Adaptive Sensory-Motor Control: Ballistic Eye Movements*, Elsevier, Amsterdam.

[8] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. (1988), *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, England.

[9] Sternberg, S., Monsell, S., Knoll, R.L., Wright, C.E. (1978), "The latency and duration of rapid movement sequences: comparisons of speech and typewriting," in *Information processing in motor control and learning*, G.E. Stelmach, ed., Academic Press, New York.

[10] Sternberg, S., Wright, C.E., Knoll, R.L., Monsell, S. (1980),"Motor Programs in Rapid Speech: Additional Evidence," in *Perception and Production of Fluent Speech*, R.A. Cole, ed., Lawrence Erlbaum Assoc., Hillsdale, NJ.

[11] Zingale, C.M. and Kowler, E. (1987), "Planning sequences of saccades," *Vision Research* **27**(8): 1327–1341.

Methods addendum

The equations from Grossberg (1978) which we used to create initial conditions for the simulations are reproduced here:

$$\omega_i = \frac{\lambda^{i-1} + R(1 - \lambda^{i-1}) - 1}{\lambda^{i-2} + R(1 - \lambda^{i-2})}, \tag{31}$$

where $R = M/\mu$, and the k STM values in a j length list are

$$x_{kj} = \mu \prod_{m=k+1}^{j} \omega_m. \tag{32}$$