(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 February 2005 (24.02.2005)

PCT

(10) International Publication Number
WO 2005/017825 A2

(51) International Patent Classification[7]: G06N 3/02

(21) International Application Number:
PCT/GB2004/003160

(22) International Filing Date: 22 July 2004 (22.07.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
0319285.3 15 August 2003 (15.08.2003) GB

(71) Applicant *(for all designated States except US)*: **ISIS INNOVATION LIMITED** [GB/GB]; Ewert House, Ewert Place, Summertown, Oxford OX2 7SG (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **STRINGER, Simon, Maitland** [GB/GB]; Department of Experimental Psychology, University of Oxford, South Parks Road, Oxford OX1 3UD (GB).

(74) Agents: **STRACHAN, Victoria, Jane** et al.; Urquhart-Dykes & Lord, Alexandra House, 1 Alexandra Road, Swansea SA1 5ED (GB).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: THREE-DIMENSIONAL MODELLING

(57) Abstract: Apparatus for representing the spatial structure of subject's three- dimensional spatial environment using a continuous attractor neural network. Information in multiple spatial dimensions can be represented simultaneously in a single continuous attractor network, for example permitting the representation of the full 3D spatial structure of an agent's environment, where each space represented by the network would correspond to the egocentric location space of a particular type of spatial feature in the agent's environment. The different spaces can be encoded and represented within a single network.

WO 2005/017825 A2

Three-Dimensional Modelling

This invention relates generally to three-dimensional modelling techniques and, more particularly, to a method and apparatus which employs a continuous attractor neural network (CANN) to represent the full three-dimensional spatial structure of a user's environment.

Continuous attractor neural networks (CANNs) are neural networks with recurrent weights which have been trained to reflect the topology of a continuous n-dimensional space. The space is typically a space of possible states of a user or agent situated in an environment, and might be, for example, a one-dimensional space of head directions from 0 to 360 degrees, or a two-dimensional space of (x,y) Cartesian coordinate locations in an environment.

Continuous attractor neural networks can maintain a localised packet of neuronal activity representing the current state of the agent in the continuous space without external sensory input.

To date, CANNs have been used for applications such as the representation of an agent's head direction or location in the environment. In these sorts of application, the network supports a single packet of activity representing a single point in the n-dimensional space represented by the continuous attractor network. However, such a spatial representation has been found to be extremely inadequate, in that it does not provide information about the full 3D spatial structure of the agent's environment, i.e. the egocentric locations of all of the obstacles and targets for action, for example, in the environment. The limited spatial representation currently provided by existing continuous attractor networks does not provide a robust informational basis for movement and navigation through complex, cluttered, real-world environments.

We have now devised an arrangement which overcomes these problems.

In accordance with the present invention, there is provided apparatus for representing the spatial structure of subject's three-dimensional spatial environment, the apparatus comprising a continuous attractor neural network including a plurality of nodes representing said three-

dimensional spatial environment, the nodes having connections therebetween, which connections have assigned thereto weights trained to encode a number of feature spaces, whereby each feature space represents the egocentric location of a different kind of feature in said environment, a subset of said nodes being assigned to represent each feature space and individual nodes of said subset being assigned to represent different locations in said feature space, the weights assigned to the connections between the nodes of each subset being trained to capture the topology of the respective feature space, wherein each feature in the environment is capable of stimulating a neuronal activity packet, each activity packet representing the egocentric location of a respective feature, and wherein a plurality of said activity packets may be simultaneously active within said network so as to maintain different types of feature simultaneously active in the correct location within said representation of said three-dimensional spatial environment.

In accordance with the invention, a technique has been developed, in which information in multiple spatial dimensions can be represented simultaneously in a single continuous attractor network. One application of this would be the representation of the full 3D spatial structure of an agent's environment, where each space represented by the network would correspond to the egocentric location space of a particular type of spatial feature in the agent's environment. The different spaces can be encoded and represented within a single network through the following steps:

(1) The recurrent synaptic weights of the continuous attractor network are trained to encode a number of different feature spaces, where each feature space represents the egocentric location of a different kind of spatial feature in the agent's environment.

(2) Each feature space is encoded in the continuous attractor network in the following way. First, a subset of neurons are randomly selected from the network to represent the feature space. These neurons are then individually assigned to represent different random locations in the feature space. When all the neurons have been assigned to their respective locations in the feature space, the recurrent weights between the neurons are trained to capture the topology of the feature space. This can be done through the use of an associative learning rule, which ensures that the synaptic weight

increments reflect how near the pre- and post-synaptic neurons are in the feature space. This procedure is repeated for all of the different feature spaces.

(3) During operation of the network, each spatial feature in the environment stimulates a packet of neuronal activity at the appropriate location in its feature space in the continuous attractor network. These activity packets are simultaneously active, with each packet representing the egocentric location of a particular kind of spatial feature. In numerical simulations, we have found that these activity packets can be stably maintained by the network even after the visual cue has been removed. Furthermore, as the agent moves the activity packets can be correctly shifted through their respective feature spaces, for example, by an idiothetic velocity signal. It has been theoretically and experimentally that in large networks, the activity packets are able to move through their respective feature spaces without interfering with each other.

(4) An additional feature of the model which helps to stabilise the activity packets is a nonlinearity in the transfer function, in which the firing threshold is lowered for those neurons in the network which are already highly activated. This has the effect of reinforcing the firing of those neurons in the network which are already active, and so helps to stabilise individual activity packets.

Also in accordance with the present invention, there is provided a method for representing the spatial structure within a subject's three-dimensional spatial environment, the method comprising the steps of providing a continuous attractor neural network including a plurality of nodes representing said three-dimensional spatial environment, the nodes having connections therebetween, which connections have assigned thereto weights trained to encode a number of feature spaces, whereby each feature space represents the egocentric location of a different kind of feature in said environment, a subset of said nodes being assigned to represent each feature space and individual nodes of said subset being assigned to represent different locations in said feature space, the weights assigned to the representing connections between the nodes of each subset being trained to capture the topology of the respective features space, wherein each feature in the environment is capable of stimulating a neuronal activity packet, each activity packet representing the egocentric location of a respective feature, and wherein a

plurality of said activity packets may be simultaneously active within said network so as to maintain different types of feature simultaneously active in the correct location within said representation of said three-dimensional spatial environment.

The ability of the multi-packet continuous attractor networks described above to represent the egocentric locations of multiple spatial features in the agent's environment allows the network to represent the full 3D spatial structure of the agent's environment, and so provide a rich base of information for movement and navigation within complex, cluttered environments. The new method described above for enabling a continuous attractor neural network to represent the full 3D spatial structure of an agent's environment is likely to have important applications in robotics with respect to planning and executing movements of all types within complex, cluttered, real-world environments. For example, algorithms based on the new multi-packet network will offer more flexible and adaptive movement of robot manipulators, and provide more robust navigation for mobile robots. Furthermore, the spatial representations within the network can be simultaneously updated using velocity (e.g. idiothetic) signals, which permits the network to perform dead reckoning. The rich spatial representation provided by the multi-packet neural network would provide a useful compliment to other navigational aids such as visual landmark recognition.

However, the multiple spaces encoded by the new continuous attractor networks described above need not necessarily correspond to the physical world, but could instead represent more abstract spaces. For example, in speech recognition and natural language processing there may need to be multiple (acoustic and semantic) dimensions represented simultaneously by the network. This would permit these representations to interact with each other during processing of speech input or generation of speech output.

These and other aspects of the present invention be apparent from, and elucidated with reference to, the embodiment described hereinafter.

An embodiment of the present invention will now be described by way of example only and with reference to the accompanying drawings, in which:

Figure 1: Simulation of model 1 with two activity packets active in two different features spaces, x^{μ} and x^{μ}, in the same continuous attractor network of feature cells. The figure shows the steady firing rates of the feature cells within the two feature spaces after the external visual input has been removed and the activity packets have been allowed to settle. In this experiment the two feature spaces significantly overlap, i.e. they have feature cells in common. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. In the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and in the plot on the right the feature cells have been ordered according to the order they occur in the second feature space. In each plot there is a contiguous block of active cells which represents the activity packet within that feature space. In addition, in each plot there is also noise from the activity packet which is active in the other feature space.

Figure 2: Network architecture for continuous attractor network model 1, including idiothetic inputs. The network is composed of two sets of cells: (i) a continuous attractor network of feature cells which encode the position and orientation of the features in the environment with respect to the agent, and (ii) a population of idiothetic cells which fire when the agent moves within the environment. When the agent is in the light, the feature cells are stimulated by visual input I^{V}. In model 1 there are two types of modifiable connection: (i) recurrent connections (w^{RC}) within the layer of feature cells, and (ii) Idiothetic Sigma-Pi connections (w^{ID}) to feature cells from combinations of idiothetic cells (clockwise rotation cells for the simulations presented here) and feature cells.

Figure 3: Experiment 2. Simulations of model 1 with two activity packets active at different locations in the same feature space x^{μ} in the continuous attractor network of feature cells. The network thus represents the presence of the same feature at different locations in the environment relative to the agent. The figure shows the firing rates (with high rates represented by black) of the feature cells through time, where the feature cells have been ordered according to the order they occur in the feature space x^{μ}. The plot shows the two activity packets moving through the feature space x^{μ}.

Figure 4: Experiment 3. Simulation of model 1 with two activity packets active in two different features spaces x^{μ} and x^{ν}, in the same continuous attractor network of feature cells which has global inhibition. The network thus represents the presence of two different types of feature, μ and ν, in the environment. In this experiment the two feature spaces do not have any feature cells in common. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. Furthermore, in the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and in the plot on the right the feature cells have been ordered according to the order they occur in the second feature space. Thus, the left and right plots show the two activity packets moving within their respective feature spaces.

Figure 5: Experiment 4a. Simulation of model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. Conventions as in Figure 4. In this experiment the two feature spaces significantly overlap, i.e. they have feature cells in common so that there is some interference between the activity packets. Nevertheless, path integration in each of the spaces is demonstrated.

Figure 6: Learned recurrent synaptic weights between feature cells in experiment 4a. The left plot shows recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the recurrent weights from feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying symmetric weight profile about feature cell 99, which is necessary for the recurrent weights to stably support an activity packet at different locations in the space x^{μ}. However, in this experiment feature cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space, between feature cell 99 and other feature cells encoding the second feature space x^{ν}. The right plot shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The

plot shows the recurrent weights from feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

Figure 7: Learned idiothetic synaptic weights between the idiothetic (rotation) cells and feature cells in experiment 4a. The left plot shows the idiothetic weights w_{ijk}^{RC} between the rotation cell k and the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the idiothetic weights from the rotation cell and feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying asymmetric weight profile about cell 99, which is necessary for the idiothetic weights to shift an activity packet through the space x^{μ}. However, in experiment 4a feature cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space. The right plot shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the idiothetic weights from the rotation cell and feature cell 97 to the other feature cells in the subset Ω^{ν}. The right plot for the second feature space shows similar characteristics to the left plot.

Figure 8: Experiment 4b. Simulation of model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}. Experiment 4b was similar to experiment 4a, except that for experiment 4b the network contained five times as many neurons as in experiment 4a. For experiment 4b the network was composed of 5000 feature cells, with each of the feature spaces represented by 1000 feature cells chosen randomly. As the number of neurons in the network increases the movement of the activity packets through their respective spaces is much smoother, which can be seen by comparing the results shown here with those shown in Figure 5 for the smaller network.

Figure 9. General network architecture for continuous attractor network model 2. The network architecture of model 2 is similar to model 1, being composed of continuous attractor network of feature cells, and a population of idiothetic cells. However, model 2 uses Sigma-Pi

recurrent synaptic connections w^{RC} within the continuous attractor network, and higher order Sigma-Pi idiothetic synaptic connections w^{ID} to feature cells from combinations of idiothetic cells and feature cells.

Figure 10: Experiment 5. Simulation of model 2 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. In this experiment the two feature spaces significantly overlap, and have the same degree of overlap as in experiment 4. This experiment is similar to experiment 4, except here we implement model 2 with higher order synapses instead of model 1. The results presented in this figure should be compared to those shown in Figure 5 for model 1. It can be seen that with the higher order synapses used by model 2, there is much less interference between the representations in the two separate feature spaces.

Figure 11: Network architecture for model 1 augmented with an additional decoder network of e.g. motor cells. The network is composed of three sets of cells: (i) a continuous attractor network of feature cells, (ii) a population of idiothetic cells which fire when the agent moves within the environment, and (iii) a population of motor cells which represent the motor activity of the agent. During the initial motor training phase in the light, the feature cells are stimulated by visual input I^{V}, and the motor cells are driven by a training signal t. There are three types of modifiable synaptic connection: (i) Recurrent connections (w^{RC}) within the layer of feature cells, (ii) Idiothetic Sigma-Pi connections (w^{ID}) to feature cells from combinations of idiothetic cells (clockwise rotation cells for the simulations presented here) and feature cells, and (iii) Associative connections (w^{M}) to the motor cells from feature cells.

Figure 12: Experiment 6. Simulation of the network architecture of model 1 augmented the addition of a network of motor cells. The simulation of the continuous attractor network of feature cells is performed in an identical manner to experiment 4, with two activity packets active in two different overlapping feature spaces, x^{μ} and x^{ν}. This figure shows the firing rates of the linked network of motor cells through time, where the motor cells have been ordered according to the order they occur in the motor space y. There is a single activity packet in the motor network which tracks the location of the activity packet in the feature space x^{μ} represented by the continuous attractor network of feature cells. However, the pattern of

activity in the motor network does not contain the noise that is present in the feature space x^{μ} due to the representation of feature v.

As explained above, "Continuous attractor" neural networks can maintain a localised packet of neuronal activity representing the current state of an agent in a continuous space without external sensory input. In applications, such as the representation of head direction or location in the environment, only one packet of activity is needed. For some spatial computations a number of different locations, each with its own features, must be held in memory. Previous approaches to continuous attractor networks (in which one packet of activity is maintained active) are extended herein to show that a single continuous attractor network can maintain multiple packets of activity simultaneously, if each packet is in a different state space or map. It is also shown how such a network could, by learning, self-organize to enable the packets in each space to be moved continuously in that space by idiothetic (motion) inputs. Such multi-packet continuous attractor networks could be used to maintain different types of feature (such as form vs colour) simultaneously active in the correct location in a spatial representation. Further, high-order synapses can improve the performance of these networks, and the location of a packet could be read by motor networks. The multiple packet continuous attractor networks described here may, for example, be used for spatial representations in brain areas such as the parietal cortex and hippocampus.

In the first instance, model 1 will be presented, in which is a continuous attractor network that is able to stably maintain simultaneously active the representations of multiple features each one of which is in its own location. The model allows the relative spatial location of each feature to be fixed relative to each other, in which case the agent can be thought of as moving through a fixed environment. The model also allows for the case where each feature can move to different locations independently. What characterises a packet of neuronal activity is a set of active neurons which together represent a feature in a location. A set of simultaneously firing activity packets might be initiated by a set of visual cues in the environment. Later, it will be shown how these representations may be updated by idiothetic (self-motion, e.g. velocity) signals as the agent moves within its environment in the absence of the visual cues. Model 1 is able to display these properties using relatively low order synapses, which are self-organized through local, biologically plausible learning rules. The

architecture of the network described below is that shown in Figure 2. The weights learned in the network are different from those that have been considered previously in that more than one topological space is trained into the synapses of the network, as will be illustrated in Figures 6 and 7.

First, it will be demonstrated how Model 1 is able to stably maintain the representations of multiple features after the visual input has been removed, with the agent remaining stationary within its environment. A reduced version of Model 1 used to evaluate this contains a network of feature cells, which receive inputs from initiating signals such as visual cues in the environment and are connected by the recurrent synaptic weights w^{RC}. In the light, individual feature cells i are stimulated by visual input I_i^V from particular features μ in the environment, with each feature in a particular position with respect to the agent. Then, when the visual input is removed, the continued firing of the feature cells continues to reflect the position of the features in the environment with respect to the agent. The spaces represented in the attractor are continuous in that a combination of neurons represents a feature, and the combination can move continuously through the space bringing into activity other neurons responding to the same feature but in different locations in the space. The connectivity that provides for this continuity in the spaces is implemented by the synaptic weights of the connections between the neurons in the continuous attractor.

The behaviour of the feature cells is governed during testing by the following "leaky-integrator" dynamical equations. The activation h_i^F of a feature cell i is governed by the equation

$$\tau \frac{dh_i^F(t)}{dt} = -h_i^F(t) + \frac{\phi_0}{C^F}\sum_j (w_{ij}^{RC} - w^{INH}) r_j^F(t) + I_i^V, \quad (1)$$

including the following terms:

The first term, $-h_i^F(t)$, on the right of Equation (1) is a decay term. The second term on the right of Equation (1) describes the effects of the recurrent connections in the continuous attractor, where r_j^F is the firing rate of feature cell j, w_{ij}^{RC} is the recurrent excitatory (positive)

synaptic weight from feature cell j and w^{INH} is a global constant describing the effect of inhibitory interneurons. The scaling factor (ϕ_0 / C^F) controls the overall strength of the recurrent inputs to the continuous attractor network of feature cells, where ϕ_0 is a constant and C^F is the number of synaptic connections received by each feature cell from other feature cells. The third term, I_i^V, on the right of Equation (1) represents a visual input to feature cell i. When the agent is in the dark, then the term I_i^V is set to zero. Lastly, τ is the time constant of the system.

The firing rate r_i^F of feature cell i is determined from the activation h_i^F and the sigmoid transfer function

$$r_i^F(t) = \frac{1}{1+e^{-2\beta(h_i^F(t)-\alpha)}} \qquad (2)$$

where α and β are the sigmoid threshold and slope, respectively.

It can be assumed that, during the initial learning phase, the feature cells respond to visual input from particular features in the environment in particular locations. For each feature μ, there is a subset of feature cells that respond to visual input from that feature. The subset of feature cells that respond (whatever the location) to a feature μ is denoted by Ω^μ. The different subsets Ω^μ may have many cells in common and so significantly overlap with each other, or may not have cells in common in which case any particular feature cell will belong to only one subset Ω^μ. (Note that a feature cell might also be termed a feature-location cell, in that it responds to a feature only when the feature is in a given location).

After each feature μ has been assigned a subset Ω^μ of feature cells, the subset Ω^μ of feature cells is evenly distributed through the space x^μ. That is, the feature cells in the subset Ω^μ are mapped onto a regular grid of different locations in the space x^μ, where the feature cells are stimulated maximally by visual input from feature μ. However, crucially, in real nervous systems the visual cues for which individual feature cells fire maximally would be determined

randomly by processes of lateral inhibition and competition between neurons within the network of feature cells. Thus, for each feature μ the mapping of feature cells through the space x^{μ} is performed randomly, and so the topological relationships between the feature cells within each space x^{μ} are unique. The unique set of topological relationships that exist between the subset Ω^{μ} of feature cells that encode for a feature space x^{μ} is a map. For each feature μ there is a unique map, i.e. arrangement of the feature cells in the subset Ω^{μ} throughout the location space x^{μ} where the feature cells respond maximally to visual input from that feature.

The recurrent synaptic connection strengths or weights w_{ij}^{RC} from feature cell j to feature cell i in the continuous attractor are set up by associative learning as the agent moves through the space as follows:

$$\delta w_{ij}^{RC} = k^{RC} r_i^F r_j^F \tag{3}$$

where δw_{ij}^{RC} is the change of synaptic weight and k^{RC} is the learning rate constant. This rule operates by associating together feature cells that tend to be co-active, and this leads to cells which respond to the particular feature μ in nearby locations developing stronger synaptic connections.

In the simulations performed below, the learning phase consists of a number of training epochs, where each training epoch involves the agent rotating with a single feature μ present in the environment during that epoch. During the learning phase, the agent performs one training epoch for each feature μ in turn. Each training epoch with a separate feature μ builds a new map into the recurrent synaptic weights of the continuous attractor network of feature cells, corresponding to the location space x^{μ} for the particular feature.

The following addition to the model is not an essential feature, but can help to stabilize activity packets. The recurrent synaptic weights within the continuous attractor network will be corrupted by a certain amount of noise from the learning regime. This in turn can lead to drift of an activity packet within the continuous attractor network. It has been proposed that in real

nervous systems this problem may be solved by enhancing the firing of neurons that are already firing. This might be implemented through mechanisms for short term synaptic enhancement, or through the effects of voltage dependent ion channels in the brain such as NMDA receptors. In the models presented here, an approach is adopted in which these effects have been simulated by adjusting the sigmoid threshold α_i for each feature cell i as follows. At each timestep $t + \delta t$ in the numerical simulation, the following may be set:

$$\alpha_i = \begin{cases} \alpha^{\text{HIGH}} & \text{if } r_i^{\text{F}}(t) < \gamma \\ \alpha^{\text{LOW}} & \text{if } r_i^{\text{F}}(t) \geq \gamma \end{cases} \tag{4}$$

where γ is a firing rate threshold. This helps to reinforce the current positions of the activity packets within the continuous attractor network of feature cells. The sigmoid slopes are set to a constant value, β, for all cells i. The above form of non-linearity described by equation (4) can be employed to stabilize each activity packet in the presence of noise from irregular learning, and to reduce the effects of interactions between simultaneously active activity packets.

In the simulations described herein, the agent is simulated rotating clockwise, and the position of each feature μ with respect to the agent in the egocentric location space x^{μ} is in the range 0 to 360 degrees.

The dynamical equations (1) and (2) given above describe the behaviour of the feature cells during testing. However, we assume that when visual cues are available, the visual inputs I_i^{V} dominate all other excitatory inputs driving the feature cells in equation (1). Therefore, in the simulations presented in this paper we employ the following modelling simplification during the initial learning phase. During the learning phase, rather than implementing the dynamical equations (1) and (2), we train the network with a single feature μ at a time, and set the firing rates of the feature cells within the subset Ω^{μ} according to Gaussian response profiles as follows. During training with a feature μ each feature cell I in the subset Ω^{μ} is randomly assigned a unique location $x_i^{\mu} \in [0,360]$ in the space x^{μ}, at which the feature cell is stimulated

maximally by the visual input from the feature. Then, during training with each different feature μ, the firing rate r_i^F of each feature cell i in the subset Ω^μ is set according to the following Gaussian response profile.

$$r_i^F = e^{-(sF_i)^2/2(\sigma^F)^2} \quad (5)$$

where s_i^F is the distance between the current egocentric location of the feature x^μ and the location at which cell i fires maximally x_i^μ, and σ^F is the standard deviation. For each feature cell i in the subset Ω^μ, s_i^F is given by

$$s_i^F = \text{MIN}(|x_i^\mu - x^\mu|, 360 - |x_i^\mu - x^\mu|). \quad (6)$$

In experiment 1 we used the following parameter values. The parameter governing the response properties of the feature cells during learning was $\sigma^F = 10^o$. A further parameter governing the learning was $k^{RC} = 0.001$. The parameters governing the "leaky-integrator" dynamical equations (1) and (2) were $\tau = 1.0, \phi_0 = 300000$ and $w^{INH} = 0.0131$. The parameters governing the sigmoid activation function were as follows: $\alpha^{HIGH} = 0.0$, $\alpha^{LOW} = 20.0$, $\gamma = 0.5$ and $\beta = 0.1$. Finally, for the numerical simulations of the "leaky-integrator" dynamical equations (1) we employed a Forward Euler finite difference method with a timestep of 0.2.

Experiment 1: The stable representation of two different features in the environment with a stationary agent.

The aim of experiment 1 is to demonstrate how a single continuous attractor network can stably support the representations of two different types of feature after the visual input has been removed, with the agent remaining stationary within its environment. This is done by performing simulations of model 1 with two activity packets active in two different feature spaces, x^μ and x^ν, in the same continuous attractor network of feature cells. The network of feature cells thus represents the presence of two different types of feature, μ and ν, in the

environment.

For experiment 1 the network is trained with two features, μ and v. The continuous attractor network is composed of 1000 feature cells. In the simulations presented here, 200 of these cells are stimulated during the learning phase by visual input from feature μ. This subset of 200 feature cells is denoted Ω^{μ} and it is this subset of cells that is used to encode the location space for feature μ. Similarly, a further 200 feature cells are stimulated during the learning phase by visual input from feature v. This subset of 200 feature cells is denoted Ω^{v}, and it is this subset of cells that encodes the location space for feature v. For experiment 1 the two subsets, Ω^{μ} and Ω^{v} are chosen randomly from the total network of 1000 feature cells, and so the subsets significantly overlap. During the learning phase, the subset Ω^{μ} of feature cells is evenly distributed along the 1-dimensional space x^{μ} (and correspondingly for the Ω^{v} cells in the x^{v} space). The training is performed separately with 10 revolutions for each of the two spaces.

After the training phase is completed, the agent is simulated (by numerical solution of equations (1) and (2)) for 500 timesteps with visual input available, with the agent remaining stationary, and with features μ and v present in the environment. There is one occurrence of feature μ at $x^{\mu} = 72^{\circ}$, and one occurrence of feature v at $x^{v} = 252^{\circ}$. Next the visual input was removed by setting the I_i^{V} terms in equation (1) to zero, and the agent was allowed to remain in the same state for another 500 timesteps. This process leads to a stable packet of activity at $x^{\mu} = 72^{\circ}$ represented by the feature cells in the subset Ω^{μ} (Figure 1 left) and a stable packet of activity at $x^{v} = 252^{\circ}$ represented by the feature cells in the subset Ω^{v} (Figure 1 right). In the plot on the left the feature cells have been ordered according to the order they occur in the first feature space, and a stable activity packet in this space is demonstrated. In the plot on the right the feature cells have been ordered according to the order they occur in the second feature space, and a stable activity packet in this second space is confirmed. It can be seen that the two activity packets were perfectly stable in their respective spatial feature spaces, with no change even over much longer simulations.

In the model described above, it is considered only how the continuous attractor network of

feature cells might stably maintain the representations of the locations of features as the agent remained stationary. The issue of path integration will now be addressed. That is, we show how the representations of the locations of the features within the network might be updated by idiothetic (self-motion) signals as the agent moves within its environment. This is an important problem to solve in order to explain how animals can perform path integration in the absence of visual input. The issue also emphasises the continuity of each of the spaces in the continuous attractor, by showing how each packet of activity can be moved continuously.

The full network architecture of model 1, now including idiothetic inputs, is shown in Figure 2. The network is composed of two sets of cells: (i) a continuous attractor network of feature cells which encode the position and orientation of the features in the environment with respect to the agent, and (ii) a population of idiothetic cells which fire when the agent moves within the environment. (In the simulations performed below, the idiothetic cells are in fact a population of clockwise rotation cells). For model 1, the Sigma-Pi synapses connecting the idiothetic cells to the continuous attractor network use relatively low order combinations of only two pre-synaptic cells.

The network of feature cells receives Sigma-Pi connections from combinations of feature cells and idiothetic cells, where the idiothetic cells respond to velocity cues produced during movement of the agent, such as herein clockwise rotation. (The velocity cues could represent vestibular and proprioceptive inputs produced by movements, or could reflect motor commands).

The behaviour of the feature cells within the continuous attractor network is governed during testing by the following "leaky-integrator" dynamical equations. The activation h_i^{F} of a feature cell i is governed by the equation

$$\tau \frac{dh_i^{\mathrm{F}}(t)}{dt} = -h_i^{\mathrm{F}}(t) + \frac{\phi_0}{C^{\mathrm{F}}} \sum_j (w_{ij}^{\mathrm{RC}} - w^{\mathrm{INH}}) r_j^{\mathrm{F}}(t) + I_i^{\mathrm{V}}$$

$$+ \frac{\phi_1}{C^{\mathrm{FxID}}} \sum_{j,k} w_{ijk}^{\mathrm{ID}} r_j^{\mathrm{F}} r_k^{\mathrm{ID}} \qquad (7)$$

The last term on the right of equation (7) represents the input from Sigma-Pi combinations of feature cells and idiothetic cells, where r_k^{ID} is the firing rate of idiothetic cell k, and w_{ijk}^{ID} is the corresponding overall effective connection strength.

Equation (7) is a general equation describing how the activity within a network of feature cells may be updated using inputs from various kinds of idiothetic cells. In the simulations presented later, the only movement performed by the agent is clockwise rotation, and in principle only a single idiothetic cell is needed in the model to represent this movement (although in the brain such a movement would be represented by a population of cells). However, the general formulation of equation (7) can be used to incorporate inputs from various other kinds of idiothetic (self-motion) cells, for example forward velocity cells. These cells fire as an animal moves forward, with a firing rate that increases monotonically with the forward velocity of the animal. Whole body motion cells have been described in primates. In each case, however, the idiothetic signal must represent a velocity signal (speed and direction of movement) rather than say acceleration.

At the start of the learning phase the synaptic weights w_{ijk}^{ID} may be set to zero. Then the learning phase continues with the agent rotating with the feature cells and idiothetic cells firing according to the response properties described above. The synaptic weights w_{ijk}^{ID} are updated at each timestep according to a trace learning rule

$$\delta w_{ijk}^{ID} = k^{ID} r_i^F \quad \bar{r}_j^F r_k^{ID} \tag{8}$$

where δw_{ijk}^{ID} is the change of synaptic weight, r_i^F is the instantaneous firing rate of feature cell i, $\bar{r}_j^F$ is the trace value (temporal average) of the firing rate of feature cell j, r_k^{ID} is the firing rate of idiothetic cell k and k^{ID} is the learning rate. The trace value $\bar{r}^F$ of the firing rate of a feature cell is a form of temporal average of recent cell activity given by

$$\bar{r}^F(t+\delta t) = (1-\eta) r^F(t+\delta t) + \eta \bar{r}^F(t) \tag{9}$$

where η is a parameter set in the interval [0,1] which determines the contribution of the current firing and the previous trace. The trace learning rule (8) involves a product of three firing rate terms on the right hand side of the equation.

During a training epoch with a feature μ the trace learning rule (8) operates as follows. As the agent rotates, learning rule (8) associates an earlier activity pattern within the network of feature cells (representing an earlier location of the feature with respect to the agent), and the co-firing of the idiothetic cells (representing the fact the agent is rotating clockwise), with the current pattern of activity among the feature cells (representing the current location of the feature with respect to the agent). The effect of the trace learning rule (8) for the synaptic weights w_{ijk}^{ID} is to generate a synaptic connectivity such that, during testing without visual input, the co-firing of a feature cell j and the idiothetic cells, will stimulate feature cell i where feature cell i represents a location that is a small translation in the appropriate direction from the location represented by feature cell j. Thus, the co-firing of a set of feature cells representing a particular feature in a particular location, and the idiothetic cells, will stimulate the firing of further feature cells such that the pattern of activity within the feature cell network that represents that feature evolves continuously to track the true location of the feature in the environment.

It can be shown that a continuous attractor network of the same form as implemented here can perform path integration over a range of velocities, where the speed of movement of the activity packet in the continuous attractor network rises approximately linearly with the firing rate of the idiothetic cells.

Numerical simulations of model 1 will now be presented in respect of a moving agent, in which the locations of the activity packets within the network of feature cells must be updated by idiothetic signals. The simulations are for a case where the idiothetic training signal is the same for the different feature spaces represented in the network. This achieves the result that the different features move together as the agent moves, providing one solution to the binding problem, and indeed showing how the features can remain bound even despite a transform such as spatial translation through the space.

Experiment 2: Moving the representation of two identical features at different locations in the environment as an agent moves

It is well known to a person skilled in the art that the representation of two identical objects is a major issue in models of vision. The aim of experiment 2 is to demonstrate how a single continuous attractor network can represent two identical features at different locations in the environment, and update these representations as the agent rotates. This is done by performing simulations of model 1 with two activity packets active at different locations in the same feature space x^{μ} in the continuous attractor network of feature cells. In this situation the network of feature cells represents the presence of the same feature at different locations in the environment relative to the agent.

For this experiment the network is trained with only a single feature μ. The continuous attractor network is composed of 1000 feature cells. In this experiment a subset of 200 feature cells, denoted Ω^{μ}, is stimulated during training in order to encode the location space for feature μ. For each feature cell i in the subset Ω^{μ} there is a unique location of the feature μ within its space x^{μ} for which the feature cell is stimulated maximally. During the learning phase, the agent rotates clockwise for 10 complete revolutions with visual input available from feature μ present in the environment. The learning phase establishes a set of recurrent synaptic weights between the feature cells in the subset Ω^{μ} that allows these cells to stably support activity packets in the feature space x^{μ} represented by these cells.

After the training phase is completed, the agent may be simulated (by numerical solution of equations (7) and (2)) for 500 timesteps with visual input available, with the agent remaining stationary, and with two occurrences of feature μ in the environment. There was one occurrence of feature μ at $x^{\mu} = 72^{\circ}$, and another occurrence of feature μ at $x^{\mu} = 252^{\circ}$. While the agent remained in this position, the visual input terms I_i^{V} for each feature cell i in equation (7) were set to a Gaussian response profile (except for a constant scaling) to that used for the feature cells during the learning phase given by equation (5). (When there is more than one feature present in the environment, the term I_i^{V} is set to the maximum input from any one of

the features). The visual input was then removed by setting the I_i^V terms in equation (7) to zero, and the agent was allowed to remain in the same direction for another 500 timesteps. The activity for the next 200 timesteps is shown at the start of Figure 3, and it is clear that two stable packets of activity were maintained in this memory condition at the locations (of $x^\mu = 72°$ and $x^\mu = 252°$) where they were started. Next, in the period 201-1050 timesteps in Figure 3 the agent rotated clockwise (for a little less than one revolution), and the firing of the idiothetic clockwise rotation cells (set to 1) drove the two activity packets through the feature space x^μ within the continuous attractor network. From timestep 1051 the agent was again stationary and the two activity packets stopped moving.

From these results we see that the continuous attractor network of feature cells is able to maintain two activity packets active at different locations in the same feature space x^μ. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals.

However, it may be seen from Figure 3 that when the two packets begin to move, one activity packet grew a little in size while the other activity packet shrank. In other simulations it was found that during movement one activity packet can die away altogether, leaving only a single activity packet remaining. This effect was only seen during movement, and was due to the global inhibition operating between the two activity packets. Thus the normal situation was that the network remained firing stably in the state into which it was placed by an external cue; but when the idiothetic inputs were driving the system, some of the noise introduced by this was able to alter the packet size.

The shape of the activity packets shown in Figure 3 are relatively binary, with the neurons either not firing or firing fast. The degree to which the firing rates are binary vs graded is largely determined by the parameter w^{INH} which controls the level of lateral inhibition between the neurons. When the level of lateral inhibition is relatively high, the activity packets assume a somewhat Gaussian shape. However, as the level of lateral inhibition is reduced, the activity packets grow larger and assume a more step-like profile. Furthermore, the non-linearity in the activation function shown in equation (4) also tends to make the firing rates of the neurons

somewhat binarized. The contributions of both factors have been previously examined. In the simulations described herein, a relatively low level of inhibition was used in conjunction with the non-linear activation function, and this combination led to step-like profiles for the activity packets. However, in further simulations we have shown that the network can support multiple activity packets when the firing rates are graded, although keeping the network in regime where the firing rates are relatively binary does contribute to enabling the network to keep different activity packets equally alive. Although the network operates best with a relatively binary firing rate distribution, we note that the network is nevertheless a continuous attractor in that all locations in the state space are equally stable, and the activity packet can be moved continuously throughout the state space.

Experiment 3: Updating the representation of two different features in the environment using non-overlapping feature spaces

The aim of experiment 3 is to demonstrate how a single continuous attractor network can represent two different types of feature in the environment, and update these representations as the agent rotates. This is done by performing simulations of model 1 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. The network of feature cells thus represents the presence of two different types of feature, μ and ν in the environment.

The whole experiment can be run similarly to experiment 2, except that the network was trained with two features, with 200 of the cells assigned to the subset Ω^{μ} that represents feature μ, and 200 of the cells assigned to the subset Ω^{ν} that represents feature ν. For experiment 3 the two subsets, Ω^{μ} and Ω^{ν} did not overlap, that is, the two subsets did not have any cells in common. During the first learning stage the network was trained with feature μ, and then during the second learning stage the network was trained with feature ν.

The results from experiment 3 are shown in Figure 4. The left plot shows the firing rates of the subset of feature cells Ω^{μ} that encode the location space x^{μ} for feature μ and the right plot shows the firing rates of the subset of feature cells Ω^{ν} that encode the location space x^{ν} for feature ν. Furthermore, in the plot on the left the feature cells have been ordered according to

the order they occurred in the feature space x^{μ}, and in the plot on the right the feature cells have been ordered according to the order they occurred in the second feature space x^{ν}. Thus, the left and right plots show the two activity packets moving within their respective feature spaces. From timesteps 1 to 200 the agent is stationary and the two activity packets do not move. From timesteps 201 to 1050, the agent rotates clockwise and the idiothetic inputs from the clockwise rotation cells drives the two activity packets through their respective feature spaces within the continuous attractor network. From timestep 1051 the agent is again stationary and the two activity packets stop moving. From these results we see that the continuous attractor network of feature cells is able to maintain activity packets in two different feature spaces, x^{μ} and x^{ν}. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals.

Experiment 4: Updating the representation of two different features in the environment using overlapping feature spaces

In experiment 4 we demonstrate how a continuous attractor network can represent two different features in the environment using two different overlapping feature spaces, and update these representations as the agent rotates. In this case the continuous attractor network stores the feature spaces of two different features μ and ν, where the subsets of feature cells used to encode the two spaces x^{μ} and x^{ν} have a number of cells in common. This is the most difficult test case, since using overlapping feature spaces leads to significant interference between coactive representations in these different spaces. Experiment 4 was composed of two parts, 4a and 4b. In experiment 4a we used the same size network as was used for experiment 3, whereas for experiment 4b the network was five times larger in order to investigate the effects of increasing the number of neurons.

Experiment 4a was run similarly to experiment 3, with a network of 1000 feature cells, and where each of the subsets Ω^{μ} and Ω^{ν} contained 200 cells. However, for experiment 4a the two subsets Ω^{μ} and Ω^{ν} were chosen randomly from the total network of 1000 feature cells, and so the subsets significant overlapped. That is, the two subsets had approximately 40 cells in common.

The results from experiment 4a are shown in Figure 5. The left plot shows the firing rates of the subset of feature cells Ω^{μ} that encode the location space x^{μ} for feature μ, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} that encode the location space x^{ν} for feature ν. From these results we see that the continuous attractor network of feature cells is able to maintain activity packets in two different feature spaces, x^{μ} and x^{ν}. Furthermore, as the agent moves, the network representations of the egocentric locations of the features may be updated by idiothetic signals. However, experiment 4a showed two effects that were not present in experiment 3. Firstly, because the two feature spaces have cells in common, each feature space contains noise from the firing of cells in the activity packet present in the other feature space. This shows as random cell firings in each of the two spaces. Secondly, the activity packets in each of the two feature spaces are both distorted due to the interference between the two spaces. The gross distortion of the two packets was only seen during movement. However, although the two packets were able to influence each other through global inhibition, the distortion of the two activity packets was primarily due to excitatory connections that existed between the neurons in the two packets.

Figure 6 shows the learned recurrent synaptic weights w_{ij}^{RC} between feature cells in experiment 4a. The left plot of Figure 6 shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{μ} which encodes the first feature space x^{μ}. For this plot the 200 feature cells in the subset Ω^{μ} are ordered according to their location in the space x^{μ}. The plot shows the recurrent weights from feature cell 99 to the other feature cells in the subset Ω^{μ}. The graph shows an underlying symmetric weight profile about feature cell 99, which is necessary for the recurrent weights to stably support an activity packet at different locations in the space x^{μ}. However, in this experiment cell 99 was also contained in the subset Ω^{ν} which encoded the second feature space x^{ν}. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space, between feature cell 99 and other feature cells encoding the second feature space x^{ν}. The right plot of Figure 6 shows the recurrent weights w_{ij}^{RC} between the feature cells in the subset Ω^{ν} which encodes the second feature space x^{ν}. For this plot the 200 feature cells in the subset Ω^{ν} are ordered according to their location in the space x^{ν}. The plot shows the recurrent weights from feature cell 97 to the

other feature cells in the subset Ω^v. The right plot for the second feature space shows similar characteristics to the left plot.

Figure 7 shows the learned idiothetic synaptic weights w_{ijk}^{ID} between the idiothetic (rotation) cells and feature cells in experiment 4a. The left plot of Figure 7 shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^μ which encodes the first feature space x^μ. For this plot the 200 feature cells in the subset Ω^μ are ordered according to their location in the space x^μ. The plot shows the idiothetic weights from the rotation cell and feature cell 99 to the other feature cells in the subset Ω^μ. The graph shows an underlying asymmetric weight profile about cell 99, which is necessary for the idiothetic weights to shift an activity packet through the space x^μ. From the idiothetic weight profile shown in the left plot of Figure 7, it can be seen that the co-firing of the rotation cell k and feature cell 99 will lead to stimulation of other feature cells that are a small distance away from feature cell 99 in the space x^μ. This will lead to a shift of an activity packet located at feature cell 99 in the appropriate direction in the space x^μ. In this way, the asymmetry in the idiothetic weights is able to shift an activity packet through the space x^μ when the agent is rotating in the absence of visual input. However, in experiment 4a feature cell 99 was also contained in the subset Ω^v which encoded the second feature space x^v. Thus, there is additional noise in the weight profile due to the synaptic weight updates associated with the second feature space. The right plot of Figure 7 shows the idiothetic weights w_{ijk}^{ID} between the rotation cell k and the feature cells in the subset Ω^v which encodes the second feature space x^v. For this plot the 200 feature cells in the subset Ω^v are ordered according to their location in the space x^v. The plot shows the idiothetic weights from the rotation cell and feature cell 97 to the other feature cells in the subset Ω^v. The right plot for the second feature space shows similar characteristics to the left plot.

In experiment 4b it is investigated how the network performed as the number of neurons in the network increased. This is an important issue given that recurrent networks in the brain, such as the CA3 region of the hippocampus, may contain neurons with many thousands of recurrent connections from other neurons in the same network. Experiment 4b was similar to

experiment 4a, except that for experiment 4b the network contained five times as many neurons as in experiment 4a. For experiment 4b the network was composed of 5000 feature cells, with each of the feature spaces represented by 1000 feature cells chosen randomly. It was found that as the number of neurons in the network increased there was less interference between the activity packets, and the movement of the activity packets through their respective spaces was much smoother. This can be seen by comparing the results shown in Figure 8 for the large network with those shown in Figure 5 for the smaller network. It can be seen that, in the small network, the size of the activity packets varied continuously through time. In further simulations (not shown) this could lead to the ultimate extinction of one of the packets. However, in the large network, the activity packets were stable. That is, the size of the activity packets remained constant as they moved through their respective feature spaces. The simulation result of experiment 4b supports the hypothesis that large recurrent networks in the brain are able to maintain multiple activity packets, perhaps representing different features in different locations in the environment.

From experiment 4 it can be seen that one way to reduce interference between activity packets in different spaces is to increase the size of the network. Another way of reducing the interference between simultaneously active packets in different feature spaces, using higher order synapses will now be described.

In model 2 described herein, the recurrent connections within the continuous attractor network of feature cells employ Sigma-Pi synapses to compute a weighted sum of the products of inputs from other neurons in the continuous attractor network. In addition, in model 2 the Sigma-Pi synapses connecting the idiothetic cells to the continuous attractor network use even higher order combinations of pre-synaptic cells.

The general network architecture of model 2 is shown in Figure 9. The network architecture of model 2 is similar to model 1, being composed of a continuous attractor network of feature cells, and a population of idiothetic cells. However, model 2 combines two presynaptic inputs from other cells in the attractor into a single synapse w^{RC}; and for the idiothetic update synapses combines two presynaptic inputs from other cells in the continuous attractor with an idiothetic input in synapses w^{ID}. The synaptic connections within model 2 are self-organized

during an initial learning phase in a similar manner to that described above for model 1.

The behaviour of the feature cells within the continuous attractor network is governed during testing by the following "leaky-integrator" dynamical equations. Model 2 is introduced with synapses that are only a single order greater than the synapses used in model 1, but in principle the order of the synapses can be increased. In model 2 the activation h_i^{F} of a feature cell i is governed by the equation

$$\begin{aligned}\tau \frac{dh_i^{\mathrm{F}}(t)}{dt} &= -h_i^{\mathrm{F}}(t) + I_i^{\mathrm{V}} \\ &+ \frac{\phi_0}{C^{\mathrm{F}}} \sum_{j,m} w_{ijm}^{\mathrm{RC}} (r_j^{\mathrm{F}}(t) r_m^{\mathrm{F}}(t)) \\ &- \frac{\phi_0}{C^{\mathrm{F}}} \sum_{j} w^{\mathrm{INH}} r_j^{\mathrm{F}}(t) \\ &+ \frac{\phi_1}{C^{\mathrm{FxID}}} \sum_{j,m,k} w_{ijmk}^{\mathrm{ID}} (r_j^{\mathrm{F}} r_m^{\mathrm{F}} r_k^{\mathrm{ID}})\end{aligned} \qquad (10)$$

The effect of the higher order synapses between the neurons in the continuous attractor is to make the recurrent synapses more selective than in model 1. That is, the synapse w_{ijm}^{RC} will only be able to stimulate feature cell i when both of the feature cells j,m are co-active. Each idiothetic connection also involves a high order Sigma-Pi combination of 2 pre-synaptic continuous attractor cells and one idiothetic input cell. The effect of this is to make the idiothetic synapses more selective than in model 1. That is, the synapse w_{ijmk}^{ID} will only be able to stimulate feature cell i when both of the feature cells j,m and the idiothetic cell k are co-active. The firing rate r_i^{F} of feature cell i is determined from the activation h_i^{F} and the sigmoid function (2).

The recurrent synaptic weights within the continuous attractor network of feature cells are self-organized during an initial learning phase in a similar manner to that described above for model 1. For model 2 the recurrent weights w_{ijm}^{RC}, from feature cells j,m to feature cell i may be

updated according to the associative rule

$$\delta w_{ijm}^{RC} = k^{RC} r_i^F r_j^F r_m^F \quad (11)$$

where δw_{ijm}^{RC} is the change of synaptic weight and k^{RC} is the learning rate constant. This rule operates by associating the co-firing of feature cells j and m with the firing of feature cell i. This learning rule allows the recurrent synapses to operate highly selectively in that, after training, the synapse w_{ijm}^{RC} will only be able to stimulate feature cell i when both of the feature cells j,m are co-active.

The synaptic connections to the continuous attractor network of feature cells from the Sigma-Pi combinations of idiothetic (or motor) cells and feature cells are self-organized during an initial learning phase in a similar manner to that described above for model 1. However, for model 2 the idiothetic weight w_{ijmk}^{ID} may be updated according to the associative rule

$$\delta w_{ijmk}^{ID} = k^{ID} r_i^F \bar{r}_j^F r_m r_k^{ID} \quad (12)$$

where δw_{ijmk}^{ID} is the change of synaptic weight r_i^F is the instantaneous firing rate of feature cell $i, \bar{r}_j^F$ is the trace value (temporal average) of the firing rate of feature cell j, etc., r_k^{ID} is the firing rate of idiothetic cell k, and k^{ID} is the learning rate. The trace value $\bar{r}^F$ of the firing rate of a feature cell is given by equation (9). During a training epoch with a feature μ, the trace learning rule (12) operates to associate the co-firing of feature cells j,m and idiothetic cell k, with the firing of feature cell i. Thus, learning rule (12) operates somewhat similarly to learning rule (8) for model 1 in that, as the agent rotates, learning rule (12) associates an earlier activity pattern within the network of feature cells (representing an earlier location of the feature with respect to the agent), and the co-firing of the idiothetic cells (representing the fact the agent is rotating clockwise), with the current pattern of activity among the feature cells (representing the current location of the feature with respect to the agent). However, learning rule (12) allows the idiothetic synapses to operate highly selectively in that after training, the

synapse w^{ID}_{ijmk} will only be able to stimulate feature cell i when both of the feature cells j,m and idiothetic cell k are co-active.

Experiment 5: Representing overlapping feature spaces with higher order synapses

The aim of experiment 5 is to demonstrate that the higher order synapses implemented in model 2 are able to reduce the interference between activity packets which are simultaneously active in different spaces. Experiment 5 is run similarly to experiment 4. That is, experiment 5 involves the simulation of model 2 with two activity packets active in two different feature spaces, x^{μ} and x^{ν}, in the same continuous attractor network of feature cells. In this experiment the two feature spaces have the same degree of overlap as was the case in experiment 4. For experiment 5, due to the increased computational cost of the higher order synapses of model 2, the network was simulated with only 360 feature cells. Each of the two feature spaces, x^{μ} and x^{ν}, was represented by a separate subset of 200 feature cells, where two subsets were chosen such that the two feature spaces had 40 feature cells in common. This overlap between the two feature spaces was the expected size of the overlap in experiment 4, where there were a total of 1000 feature cells, and each of the two feature spaces recruited a random set of 200 cells from this total.

The results of experiment 5 are presented in Figure 10, and these results can be compared to those shown in Figure 5 for model 1. The left plot shows the firing rates of the subset of feature cells Ω^{μ} belonging to the first feature space x^{μ}, and the right plot shows the firing rates of the subset of feature cells Ω^{ν} belonging to the second feature space x^{ν}. It can be seen that with the higher order synapses used by model 2, the activity packets in the two separate feature spaces are far less deformed. In particular, over the course of the simulation, the activity packets maintain their original sizes. This is in contrast to experiment 4, where one packet became larger while the other packet became smaller. Hence, with the higher order synapses of model 2, there is much less interference between the representations in the two separate feature spaces.

It will now be considered how subsequent, for example motor, systems in the brain are able to respond to the representations of multiple features supported by a continuous attractor network

of feature cells. The execution of motor sequences by the motor system may depend on exactly which features are present in the environment, and where the features are located with respect to the agent. However, for both models 1 and 2 presented in this paper, if multiple activity packets are active within the continuous attractor network of feature cells, then the representation of each feature will be masked by the "noise" from the other active representations of other features present in the environment, as shown in Figure 5. In this situation, how can subsequent motor systems detect the representations of individual features? It is proposed herein that a pattern associator would be able to decode the representations in the continuous attractor network and would have the benefit of reducing noise in the representation.

The way in which the decoding could work is shown in Figure 11, which shows the network architecture for model 1 augmented with a pattern associator in which the neuronal firing could represent motor commands. During an initial motor training phase in the light, the feature cells in the continuous attractor are stimulated by visual input I^{V}, the motor cells are driven by a training signal t, and the synapses w^{M} are modified by associative learning. Then, after the motor training is completed, the connections w^{M} are able to drive the motor cells to perform the appropriate motor actions. During the learning, the synaptic weights w_{ij}^{M} from feature cells j to motor cells I are updated at each timestep according to

$$\delta w_{ij}^{M} = k^{M} r_{i}^{M} r_{j}^{F} \tag{13}$$

The motor activity of the agent may be characterised by an idealised motor space y. We defined the motor space y of the agent as a toroidal 1-dimensional space from $y = 0$ to $y = 360$. This allowed a simple correspondence between the motor space y of the agent and the feature spaces. Next, we assumed that each motor cell fired maximally for a particular location in the motor space y, and that the motor cells are distributed evenly throughout the motor space y.

Experiment 6: How motor cells respond to individual representations within the network of feature cells

In experiment 6 it can be demonstrated that the motor network is able to respond appropriately to the representation of a particular feature μ in the continuous attractor network of feature

cells, even when the representation of feature μ is somewhat masked by the presence of noise due to the representation of another feature ν in the same continuous attractor network. Experiment 6 was similar to experiment 4, except here we augment the model 1 network architecture to include a network of motor cells, as described above.

For experiment 6, the continuous attractor network of feature cells is trained with two features μ and ν in an identical manner to that described above for experiment 4. The motor network was trained as follows. The motor network contains 200 motor cells. During the first stage of learning, while the continuous attractor network of feature cells was being trained with feature μ, the network of motor neurons was trained to perform a particular motor sequence. The learned motor sequence was simply $y = x^{\mu}$. That is, the motor neurons learned to fire so that the activity packet within the motor network mirrors the location of the activity packet in the feature space x^{μ}. While the agent ran through the motor sequence associated with feature μ during the first stage of the training phase, the synaptic weights w_{ij}^{M} are updated according to equation (13). During the second stage of training, in which the feature cells were trained with feature ν, the motor cells did not fire, and so the synaptic weights w_{ij}^{M} do not change. Then, after training, the goal was for the network to repeat the motor sequence with $y = x^{\mu}$, even when there is a representation of the second feature ν also present in the network of feature cells. In this case, the network of motor cells must be able to ignore the irrelevant representation in the second feature space x^{ν}.

Results from experiment 6 are shown in Figures 5 and 12, which show the firing rates of the feature cells and motor cells through time. In Figure 12 the motor cells have been ordered according to the order they occur in the motor space y. There is a single activity packet in the motor network which tracks the location of the activity packet in the feature space x^{μ} represented by the continuous attractor network of feature cells. This may be seen by comparing the results in Figure 12 with those shown in the left plot of Figure 5. However, the pattern of activity in the motor network does not contain the noise that is present in the feature space x^{μ} due to the presentation of feature ν. Thus, the motor network is able to filter out the noise due to the representations of irrelevant features. This means that the motor network performs the motor sequence correctly in that, at each stage of the sequence, the correct motor

neurons fire, but with no firing of the other neurons in the motor network.

Experiment 6 demonstrates that the motor system is able to detect and respond appropriately to the representation of a particular feature μ in the continuous attractor network of feature cells, even when the representation of feature μ is somewhat masked by the presence of noise due to the representations of other features in the same continuous attractor network. This is because, if the firing threshold is set to a relatively high value for the motor cells, the motor cells will ignore random "salt and pepper" noise (due to activity packets in overlapping feature spaces) in the feature space x^{μ} they have learned to respond to, and will only respond to the presence of a genuine contiguous activity packet in the space x^{μ}.

Finally, in further simulations (not shown here) it can be demonstrated that the performance of the motor network could be improved further by implementing even higher order synapses w^{M}, where the pre-synaptic input terms involved a product consisting of the firing rates of two feature cells.

In the above, it has been explored how continuous attractor neural networks can be used to support multiple packets of neuronal activity, and how these separate representations may be simultaneously updated by external inputs to the continuous attractor network, where such external inputs might represent, for example, idiothetic inputs or motor efference copy. To achieve this, the continuous attractor networks presented here make use of (i) a recurrent synaptic connectivity that encodes each activity packet on a separate "map", and (ii) higher order Sigma-Pi synapses to enable the idiothetic input to move the activity packets. The networks also benefit from a non-linear neuronal activation function that could be implemented by NMDA receptors. An important property of the models presented in this paper is that they are self-organizing with biologically plausible learning rules. However, the framework developed here is quite flexible. For example, although in the simulations presented above the features moved in tandem, it is possible for the external signals to learn to drive some but not other activity packets, and to drive the activity packets at differing speeds from one another. Hence, we propose that such multi-packet continuous attractor networks may play an important role in various aspects of brain function.

The simulations described above showed that when more than one activity packet is active in the network, the activity packets may under some conditions be stable. One of the conditions is a bounded non-linear transfer function is used. Considering stationary activity packets, two packets of activity in a single feature space remain separate and stable if the activity packets are far enough apart, as described above. This is true even when the activity packets are of unequal size. Further, as demonstrated above, two stationary activity packets can remain stable even if they share active neurons, but the activity packets are in different feature spaces. Next the situation will be considered when the agent is moving and path integration is being performed with moving activity packets. In the case of two activity packets moving in one feature space, the activity packets may interfere with each other, with one activity packet growing while the other packet shrinks. These effects were observed in experiment 2, the results of which are shown in Figure 3. When the activity packets are moving in different overlapping feature spaces, then the packets may interfere more severely. This effect was observed in experiment 4a, the results of which are shown in Figure 5. As the overlap between two feature spaces increases, the activity packets in the two spaces interfere with each other more and more. This can be seen by comparing Figures 4 and 5, which show results with zero overlap and an overlap of approximately 40 cells between the feature spaces, respectively. The results of the simulations performed in respect of experiment 4b, supports the fact that increasing the size of the network reduces the interference between the activity packets in the two different feature spaces, as shown in Figure 8. Thus it has been shown that two activity packets in different feature spaces can both be moved successfully by path integration, and result in persistent separate non-moving activity packets when the idiothetic movement-inducing signal is removed. In simulations of continuous attractor networks with the sigmoidal transfer function (2), the level of activity in the network due to both the size and number of activity packets could be controlled by the level of lateral inhibition between the neurons $w^{INH.}$

The importance of a bounded non-linear transfer function for the findings just summarized is supported by the following results. In attractor networks governed by equations (1), but trained with a small set of random binarised activity patterns, it was found that with the sigmoidal transfer function (2), which remains bounded as $h \rightarrow \infty$, stable multiple activity patterns were supported. However, when the sigmoid function was replaced with a threshold

linear function, the network was unable to support multiple activity patterns. With the threshold linear transfer function, the firing rates of the neurons in a single pattern always grew large enough to suppress the other neurons in the network. Hence, only a single pattern could be supported by the network. However, this limitation of the threshold linear transfer function could be remedied by introducing an upper bound on the neuronal firing rates. When the firing rates were clipped to a maximum of 1, the network was once more able to support multiple patterns. Thus, these simulations suggest that in order for the network to support multiple patterns, the form of the transfer function must ensure that the neuronal firing rates are bounded as the activation h_i increases. Providing the transfer function was bounded, the number of activity patterns stably supported by the network could be controlled by the level of lateral inhibition between the neurons $w^{INH.}$

The concept of storing multiple maps in a single continuous attractor network has been investigated previously. The concept of a chart comes from neurophysiological studies of place cells in rats, which respond when the rat is in a particular place. When moved to a different environment, the relative spatial locations in which neurons fire appear to be different, and hence the concept arises that these hippocampal place cells code for different charts, where each chart applies in a different spatial environment. However, within any one environment or "chart", there would be only one activity packet representing the current location of the animal. In contrast, in the continuous attractor model according to the invention, the concept is that multiple maps can have simultaneously active packets of activity. Indeed, a more abstract viewpoint is adopted, in which different maps may be thought of as distinct spaces in which a representation specific to that space may move continuously. Thus, in the models presented herein, rather than using different charts to represent different environments, we use each map to represent the space through which the representation of an individual feature may move continuously to represent the changing position of the feature with respect to the agent. The system we describe can maintain several packets of activity simultaneously because each activity packet receives support from the other neurons in the same space in the continuous attractor. As shown in the simulations, individual neurons can be in more than one of the spaces.

A key question is how many feature spaces may be learned by a continuous attractor network

before the network reaches its loading capacity. This question has been investigated previously. The maximum number of feature spaces (charts) that may be stored is

$$N_{feature\ spaces} \sim - C/log\ (a_m) \quad (14)$$

where C is the average number of recurrent synaptic contacts per cell, and a_m is the map sparseness which is related to the size of a typical activity packet relative to the size of the entire feature space.

The system we describe enables the representations in the different maps to be moved together or separately by the same or different idiothetic inputs. This provides one solution to the binding problem, in that features in different classes (and hence maps) can be moved together by for example a single idiothetic input.

In the above the theory of multi-packet continuous attractor networks was developed in the context of how a brain might represent the 3-dimensional structure of an animal's environment. This involves the representation of many independent features and their individual spatial relationships to the agent. After the agent has learned an alphabet of features through early visual experience, when the agent is exposed to a single snapshot view of a new environment (not encountered during training), a full 3-dimensional representation of the new environment is initiated in the network of feature cells. Then, when the visual input is removed, the representation can be maintained and updated as the agent moves through its environment in the absence of visual input. Hence, the models presented in this paper can perform path integration using only a single view of a new environment. Another application of this class of model is to the situation when more than one spatial location must be simultaneously and independently represented, as occurs for example when one moving object may collide with or miss another moving object. To the extent that this can be represented in allocentric space, this is likely to involve the hippocampus (or structures that receive from it), and the hippocampus does not use topological mapping in the brain, so that the individual representations of allocentric space would overlap.

The network described here is able to learn how to move the activity packets in their egocentric feature spaces given any kind of vestibular velocity signal (e.g. clockwise or anti-

clockwise rotation, forward velocity etc, or perhaps some form of motor efference copy). It should not matter, for example, that rotations in 3-D space do not commute, unlike rotations in one-dimensional space. To see this, consider the following situation. If a clockwise rotation of the agent results in the relative change in position of a feature μ from egocentric location A to location B1, and a further upward turn of the agent results in the relative movement of the feature μ from location B1 to location C1, then these transitions are what the network would learn. Furthermore, if an upward turn of the agent results in the relative change in position of feature μ from egocentric location A to location B2, and a further clockwise rotation of the agent results in the relative movement of feature μ from location B2 to location C2, then these transitions would also be learned by the network. The network would be capable of learning both transition sequences, and replaying either sequence using only the relevant vestibular signals.

A key problem with current models of hippocampal place cells is the inability of the representations supported by these neural networks to provide a basis for planning novel routes through complex environments full of obstacles. Current models of place cells assume a single activity packet in a 2-dimensional layer of place cells, where the cells are simply mapped onto the floor of the containment area. However, such a representation merely locates the agent in a 2-dimensional space, and cannot provide information about the full 3-dimensional structure of the surrounding environment, which would be necessary for planning a novel route along paths and past obstacles, etc. However, we have addressed herein how the full 3-dimensional structure of the surrounding environment might be represented in a continuous attractor network, and how this representation may be updated through idiothetic signals or motor efference copy. Only such a representation of the full 3-dimensional structure of the agent's environment will provide a robust basis for planning novel routes in complex, cluttered environments.

The concept introduced herein may be relevant to understanding the visuo-spatial scratchpad thought to implement a representation of spatial positions of several objects in a scene. Consider the output of the inferior temporal cortext (IT), which provides a distributed representation of an object close to the fovea under natural viewing conditions. The representation of this object in different positions in egocentric space would be learned by a

continuous attractor network of the type described in this paper by combining this output of IT with a signal (present in the parietal cortex) about the position of the eyes (and head etc). For each egocentric position, the network would have an arbitrary set of neurons active that would represent the object in that position. As the agent moved, the relation between the idiothetic self-motion signals and the object input would be learned as described here. Each object would be trained in this way, with a separate "object" space or chart for each object. After training, eye movements around the scene would establish the relative positions of objects in the scene, and after this, any idiothetic self-motion would update the positions of all the objects in egocentric space in the scene.

There is some evidence from cue rotation experiments that different representations can be simultaneously active in the rat hippocampus, and if so, the simultaneously active representations could be based on processes of the type discussed herein. There is also evidence for multiple representations in the rat hippocampus from experiments in which individual visual cues in the rat's environment are moved. In such experiments, the activity of many cells followed either the distal or local cue sets, while other cells encoded specific subsets of the cues. In particular, it was demonstrated that some hippocampal place cells encode the egocentric location of a number of different subsets of environmental stimuli with respect to the rat. These experiments suggest that the rat hippocampus may support multiple independent activity packets that represent different aspects of the spatial structure of the environment, with individual place cells taking part in more than one representation. Moreover, the fact that spatial cells in the hippocampus are not arranged in a topographic map suggests that the feature spaces are encoded by different random orderings of the cells. Whether it is the hippocampus or some other brain region that maintains a dynamical representation of the full 3-dimensional structure of the agent's environment to provide a robust basis for planning novel routes in complex, cluttered environments remains to be shown.

An embodiment of the present invention has been described above by way of example only, and it will be apparent to a person skilled in the art that modifications and variations can be made to the described embodiments without departing from the scope of the invention as defined by the appended claims.

CLAIMS:

1. Apparatus for representing the spatial structure within a subject's three-dimensional spatial environment, the apparatus comprising a continuous attractor neural network including a plurality of nodes representing said three-dimensional spatial environment, the nodes having connections therebetween, which connections have assigned thereto weights trained to encode a number of feature spaces, whereby each feature space represents the egocentric location of a different kind of feature in said environment, a subset of said nodes being assigned to represent each feature space and individual nodes of said subset being assigned to represent different locations in said feature space, the weights assigned to the connections between the nodes of each subset being trained to capture the topology of the respective feature space, wherein each feature in the environment is capable of stimulating a neuronal activity packet, each activity packet representing the egocentric location of a respective feature, and wherein a plurality of said activity packets may be simultaneously active within said network so as to maintain different types of feature simultaneously active in the correct location within said representation of said three-dimensional spatial environment.

2. Apparatus according to claim 1, wherein recurrent synaptic weights of the continuous attractor network are trained to encode a number of different feature spaces, where each feature space represents the egocentric location of a different kind of feature in the agent's environment.

3. Apparatus according to claim 2, wherein each feature space is encoded in the continuous attractor network by means of the following steps:
   (a) a subset of neurons are randomly selected from the network to represent the feature space;
   (b) said subset of neurons are then individually assigned to represent different random locations in the feature space; and
   (c) when all the neurons have been assigned to their respective locations in the feature space, the recurrent weights between the neurons are trained to capture the topology of the feature space.

4. Apparatus according to any one of claims 1 to 3, wherein nonlinearity in the transfer function is employed to stabilise the activity packets, in which the firing threshold is lowered for those neurons in the network which are already highly activated.

5. A method for representing the spatial structure within a subject's three-dimensional spatial environment, the method comprising the steps of providing a continuous attractor neural network including a plurality of nodes representing said three-dimensional spatial environment, the nodes having connections therebetween, which connections have assigned thereto weights trained to encode a number of feature spaces, whereby each feature space represents the egocentric location of a different kind of feature in said environment, a subset of said nodes being assigned to represent each feature space and individual nodes of said subset being assigned to represent different locations in said feature space, the weights assigned to the c onnections between the nodes of each subset being trained to capture the topology of the respective features space, wherein each feature in the environment is capable of stimulating a neuronal activity packet, each activity packet representing the egocentric location of a respective feature, and wherein a plurality of said activity packets may be simultaneously active within said network so as to maintain different types of feature simultaneously active in the correct location within said representation of said three-dimensional spatial environment.

6. A method according to claim 5, including the step of feature spaces as said agent moves through said three dimensional spatial environment.

7. Apparatus substantially as herein described with reference to the accompanying drawings.

8. A method substantially as herein described with reference to the accompanying drawings.

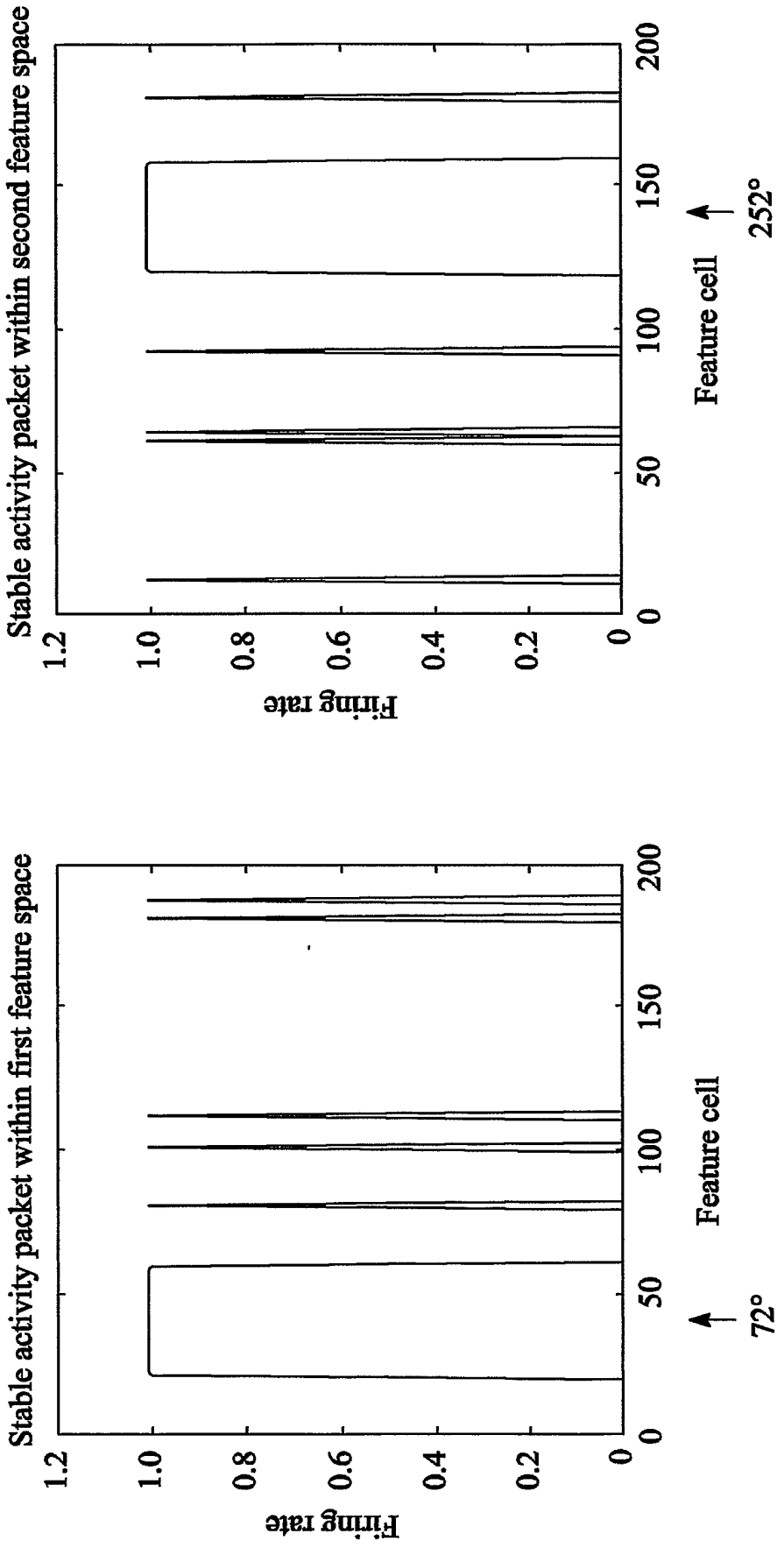
Stable activity packet within first feature space
Firing rate
1.2
1.0
0.8
0.6
0.4
0.2
0
0
50
100
150
200
Feature cell
72°
Stable activity packet within second feature space
Firing rate
1.2
1.0
0.8
0.6
0.4
0.2
0
0
50
100
150
200
Feature cell
252°

FIG. 1

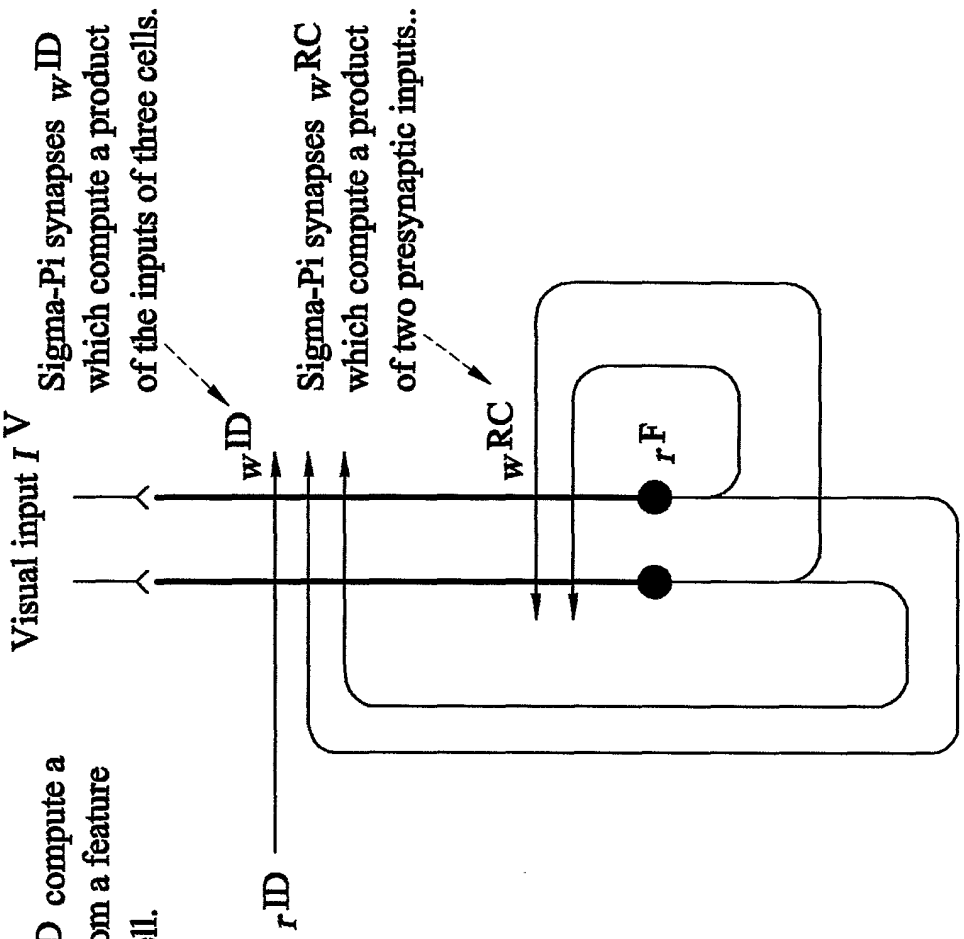
Visual input I V
Sigma-Pi synapses w^{ID} which compute a product of the inputs of three cells.
Sigma-Pi synapses w^{RC} which compute a product of two presynaptic inputs..
w^{ID}
w^{RC}
r^{F}
r^{ID}

FIG. 9

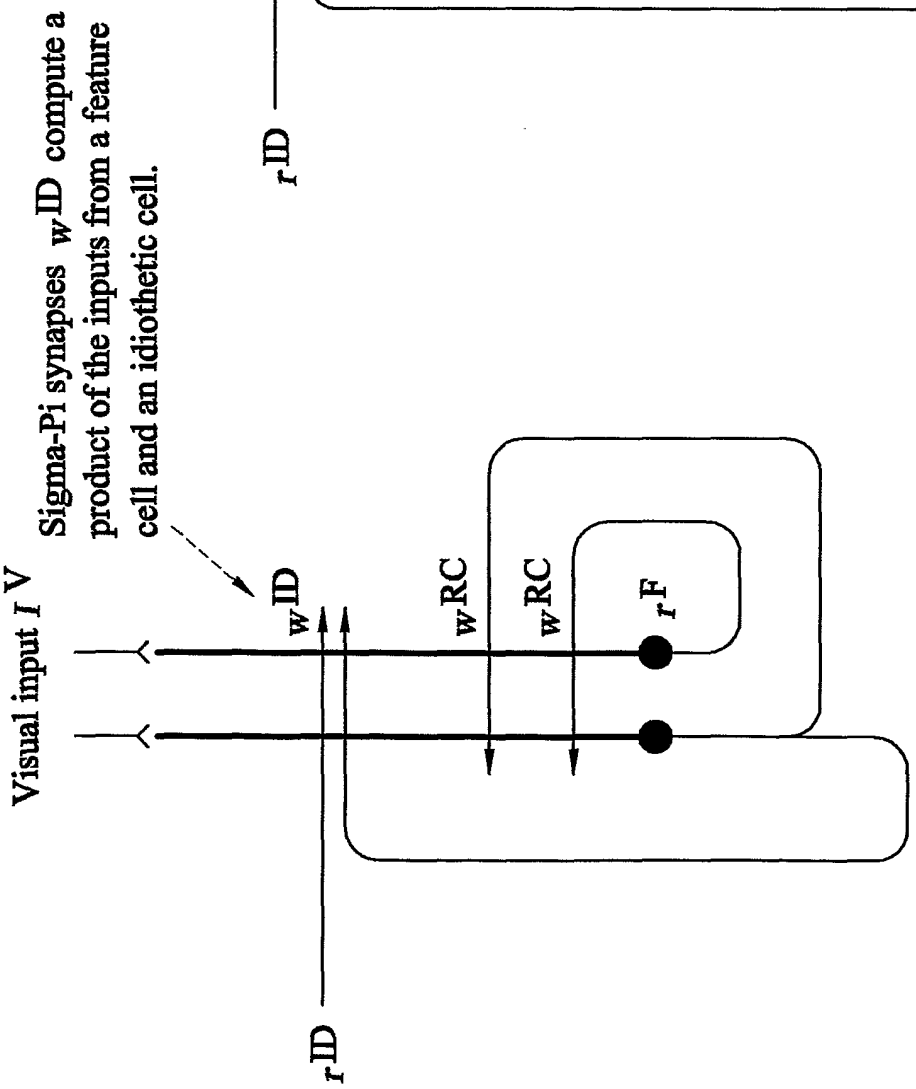
Visual input I V
Sigma-Pi synapses w^{ID} compute a product of the inputs from a feature cell and an idiothetic cell.
w^{ID}
w^{RC}
w^{RC}
r^{F}
r^{ID}

FIG. 2

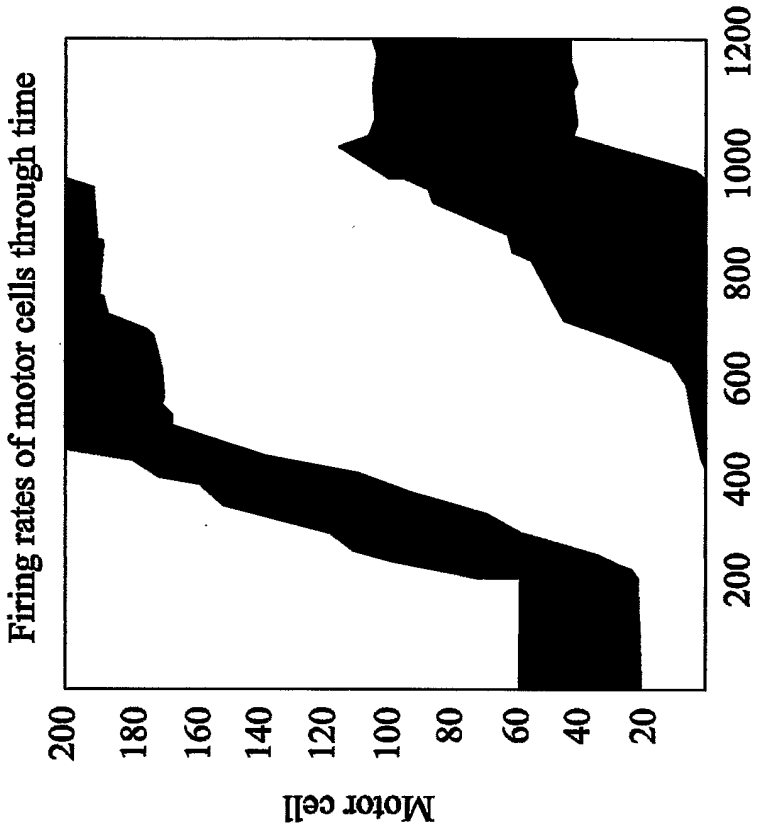
Firing rates of motor cells through time
Motor cell
200
180
160
140
120
100
80
60
40
20
200
400
600
800
1000
1200
Timestep

FIG. 12

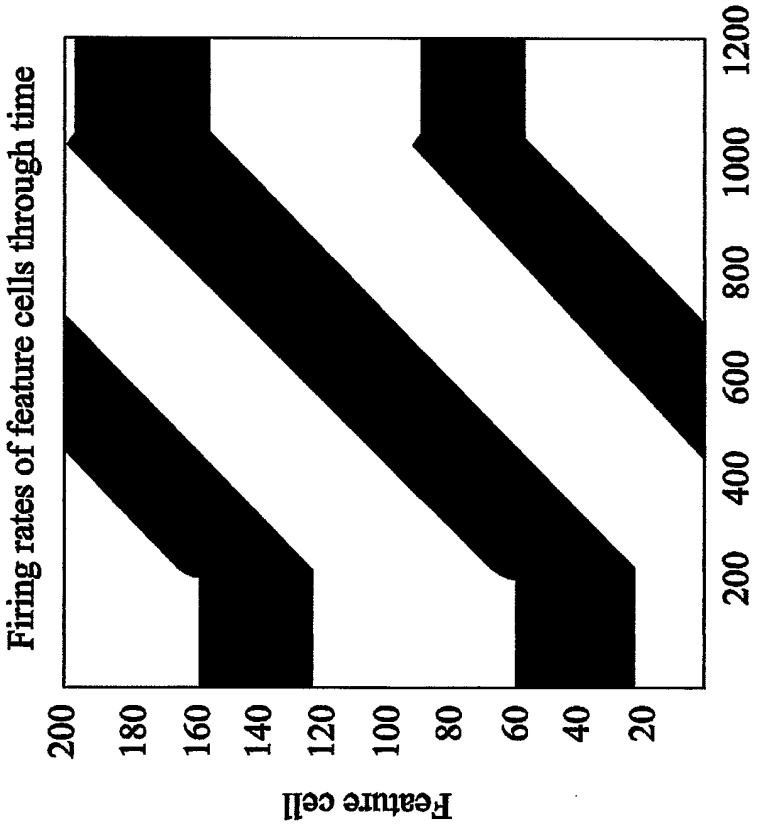
Firing rates of feature cells through time
Feature cell
200
180
160
140
120
100
80
60
40
20
200
400
600
800
1000
1200
Timestep

FIG. 3

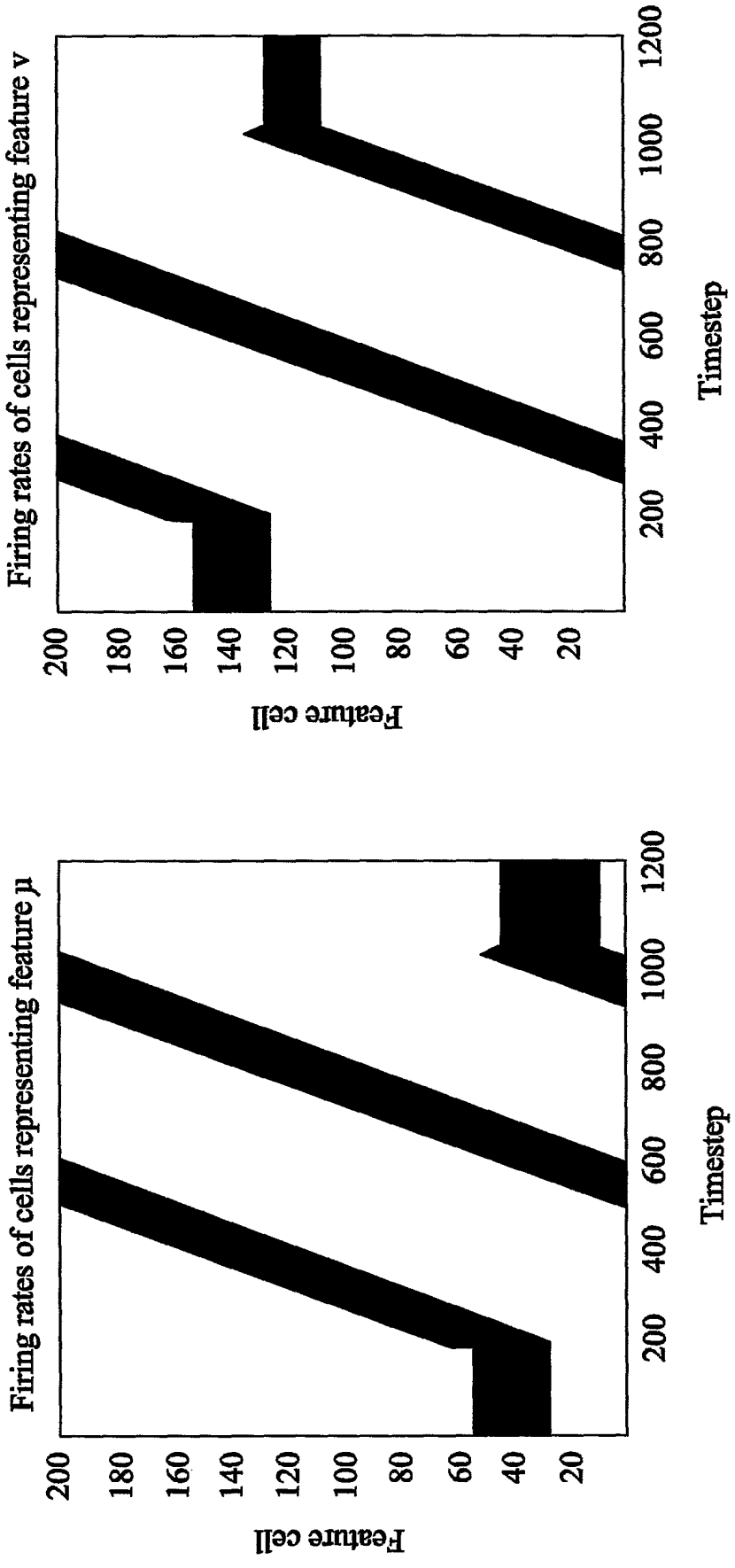
Firing rates of cells representing feature v
Feature cell
200
180
160
140
120
100
80
60
40
20
200
400
600
800
1000
1200
Timestep
Firing rates of cells representing feature μ
Feature cell
200
180
160
140
120
100
80
60
40
20
200
400
600
800
1000
1200
Timestep

FIG. 4

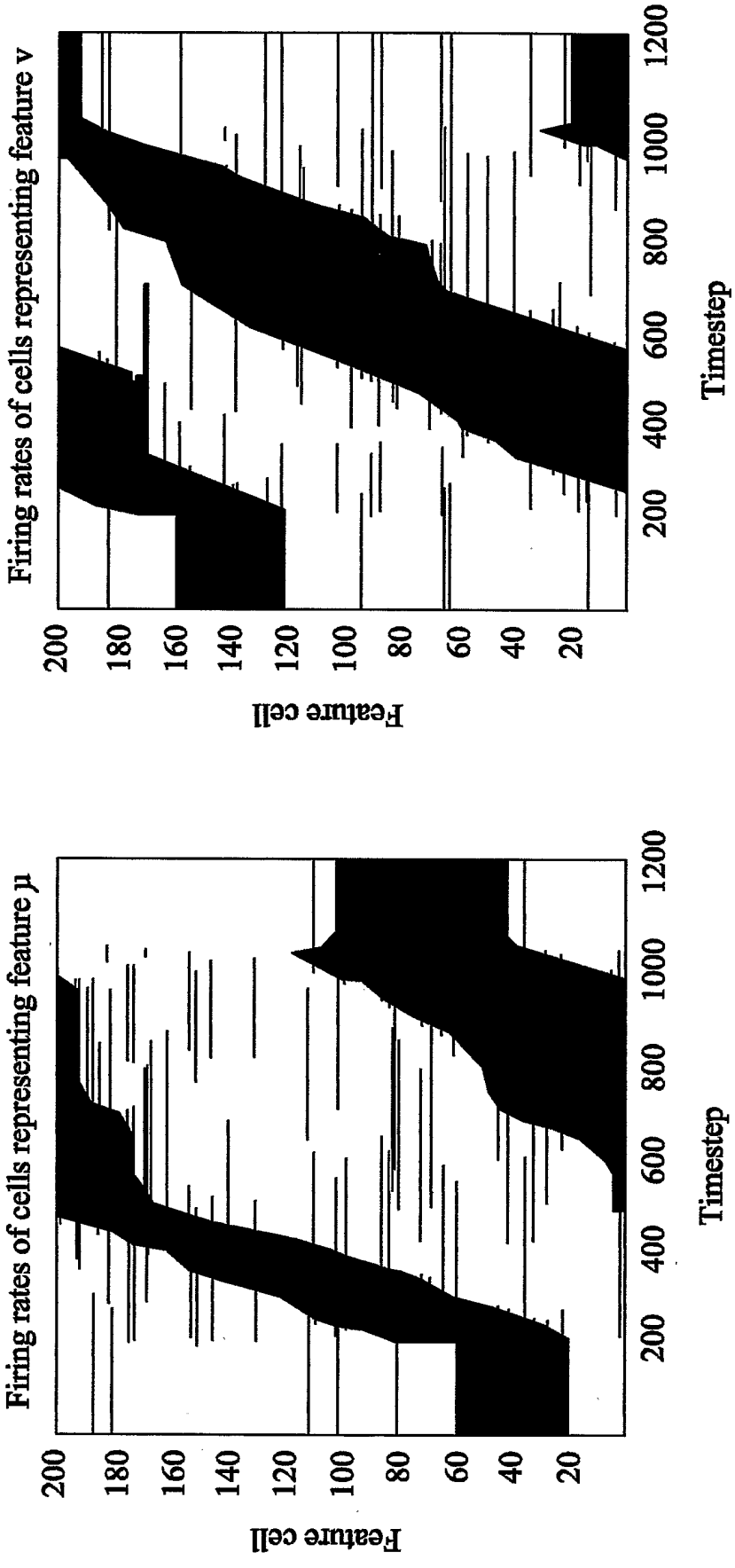
Firing rates of cells representing feature v
Feature cell
200
180
160
140
120
100
80
60
40
20
Timestep
200
400
600
800
1000
1200
Firing rates of cells representing feature μ
Feature cell
200
180
160
140
120
100
80
60
40
20
Timestep
200
400
600
800
1000
1200

FIG. 5

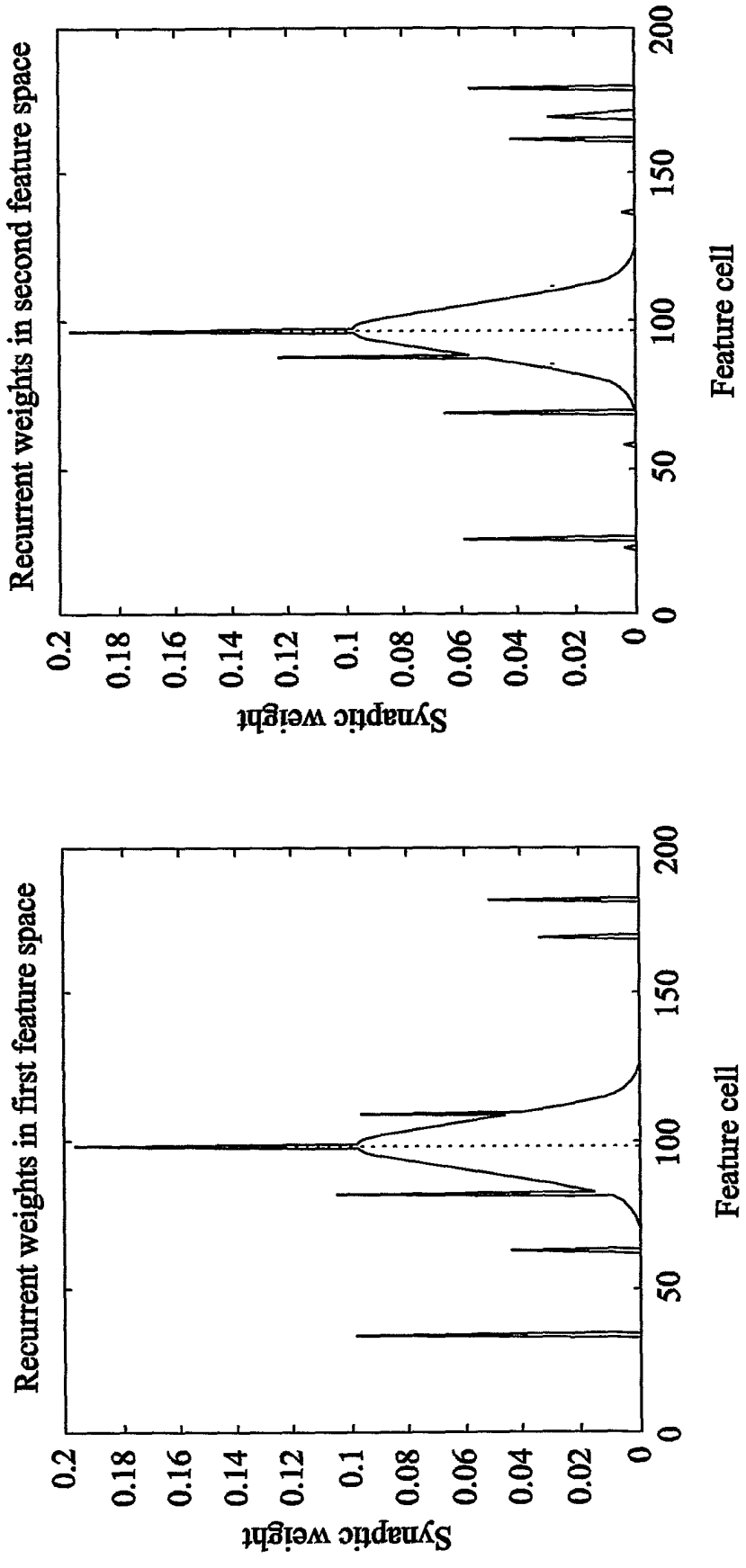
Recurrent weights in first feature space
Synaptic weight
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
Feature cell
Recurrent weights in second feature space
Synaptic weight
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
Feature cell

FIG. 6

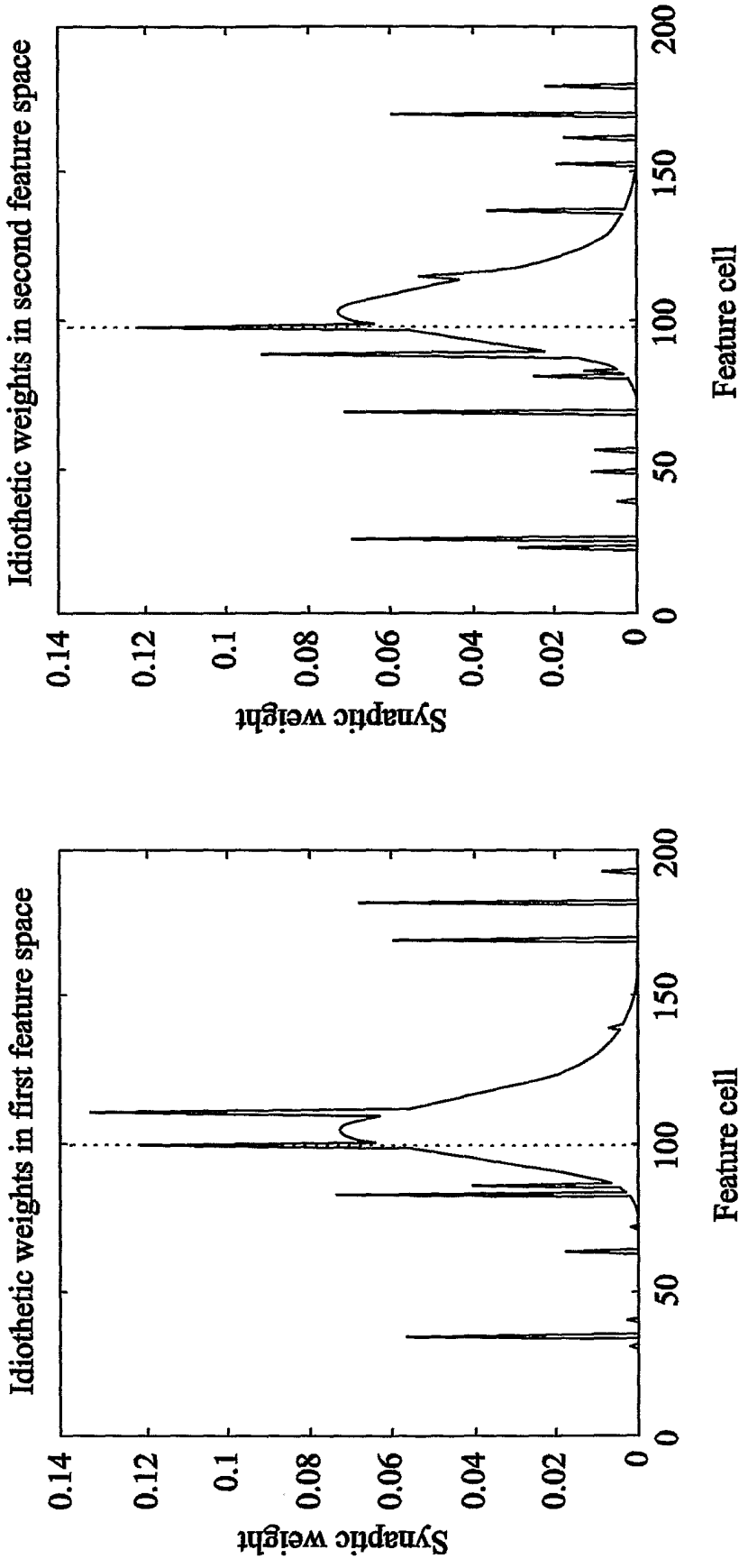
Idiothetic weights in first feature space
Synaptic weight
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
Feature cell
Idiothetic weights in second feature space
Synaptic weight
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
50
100
150
200
Feature cell

FIG. 7

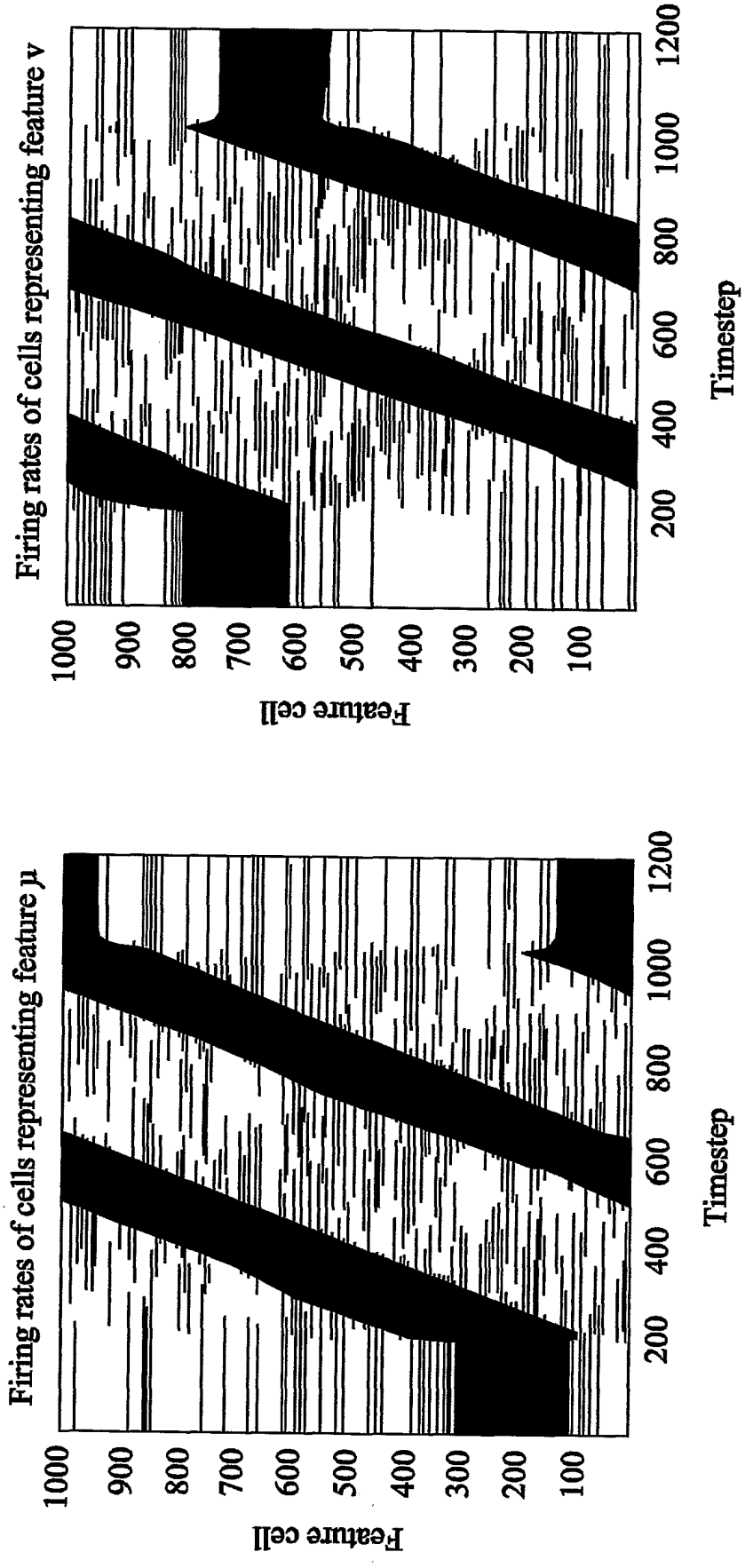
Firing rates of cells representing feature v
Feature cell
1000
900
800
700
600
500
400
300
200
100
200
400
600
800
1000
1200
Timestep
Firing rates of cells representing feature μ
Feature cell
1000
900
800
700
600
500
400
300
200
100
200
400
600
800
1000
1200
Timestep

FIG. 8

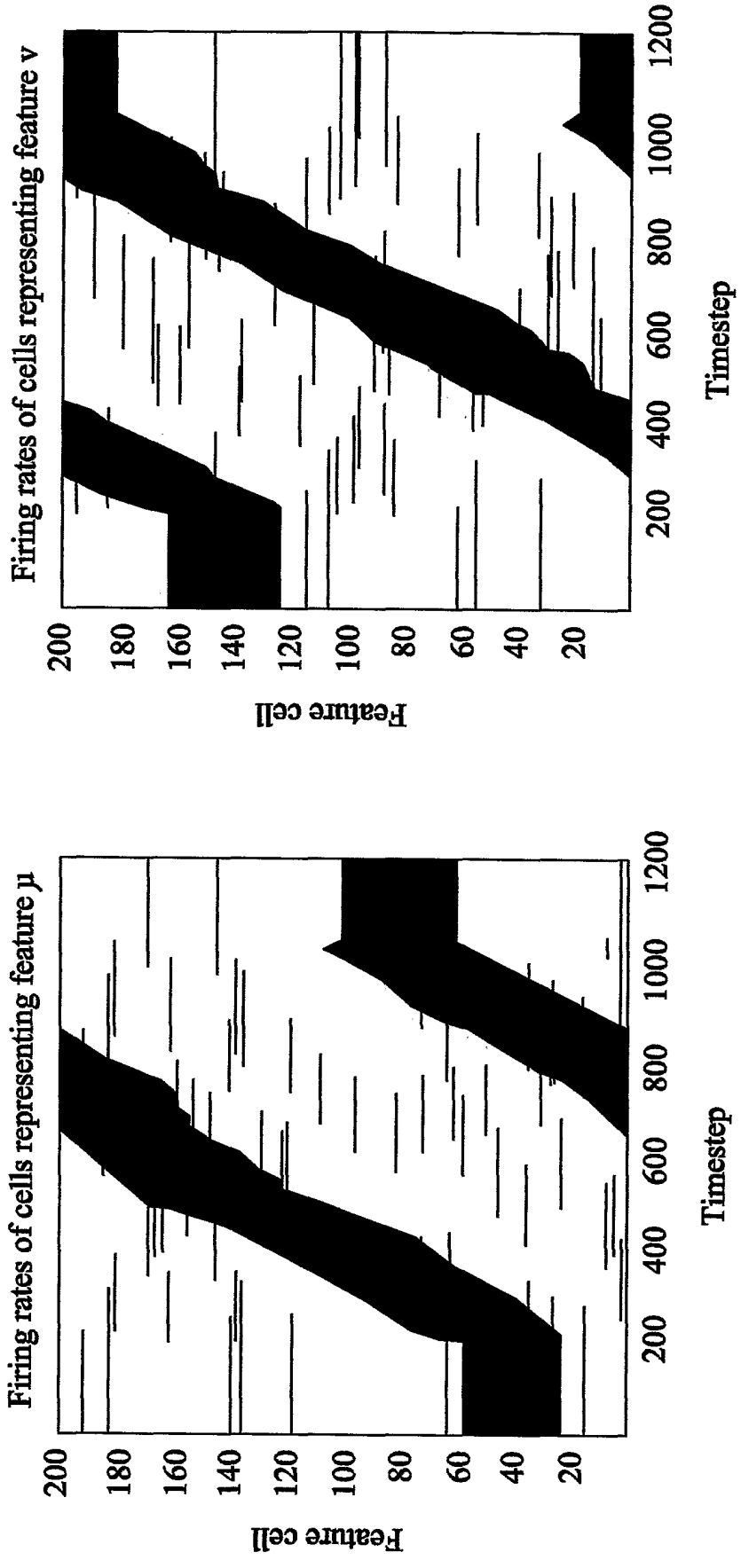
Firing rates of cells representing feature v
Feature cell
Timestep
200
180
160
140
120
100
80
60
40
20
200
400
600
800
1000
1200
Firing rates of cells representing feature μ
Feature cell
Timestep

FIG. 10

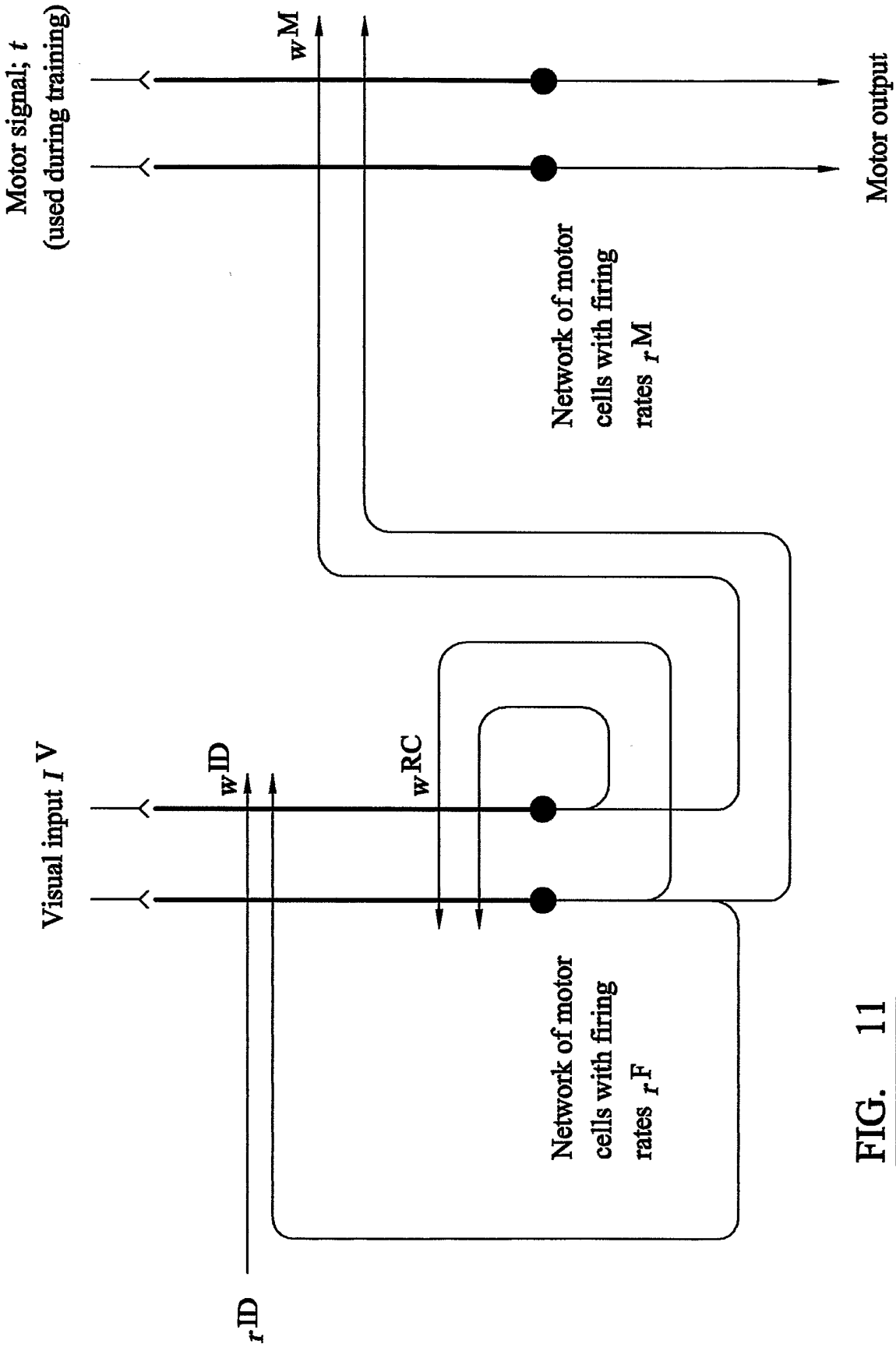
Motor signal; t
(used during training)
wM
Motor output
Network of motor
cells with firing
rates rM
Visual input I V
wID
wRC
Network of motor
cells with firing
rates rF
rID

FIG. 11