obtained by on-line learning and which makes the fluctuation of the CC curve stay within a certain range centred at the dot-dashed line.

The experiments of the model were limited by two factors. One is that the angular velocity obtained is dependant on the error model of the dead-reckoning mechanism in the robot, meaning that the orientation predicted by the HD system drifts faster than that estimated by dead-reckoning. This turns out to be a problem, because the model will not learn the orientation-to-feature association properly if there is too big an error in its prediction of the head direction. Therefore we temporarily use the best available estimation, i.e., the dead-reckoned orientation to limit the drift in the HD system. This is the reason that in Figure 5 (a) the CC curve has in general a tendency to follow the NCC curve. Since the only counterpart in the robot to the vestibular system is gyro which is not available in our experiment setup, this limitation is expected to be removed later by using a gyro to provide angular velocity. The other factor is the continuity of the detected feature. They often stretch a certain angle covering several HD units, as illustrated in Figure 5 (c) where only those features which generate the calibration tics in Figure 5 (a) are shown, and the others are mapped to the feature zero. The continuity of a detected feature makes it difficult to select the right calibration point in the HD units associated with the feature. Tests with ideal odometric parameters have shown that all wrong calibrations result from the wrong selection of the calibration point, suggesting that the selection process of a calibration point from a H-set, see Section 4.2, should be improved.

In summary, the preliminary experiments we have carried out so far show that the suggested model is able to modify the internal orientation of the robot by using learned environmental features. In the future we need to improve the resolution of the model and to examine to which degree the behaviour of the model conforms with the data of existing behavioural experiments in animal navigation.

References

[1] Blair, H. T. A Thalamocortical Circuit for Computing Directional Heading in The Rat. *Advances in Neural Information Processing Systems* 8, ed. D. S. Touretzky, M. C. Mozer and M. E. Hasselmo. pp. 152-158, 1996.

[2] Blair, H. T. and P. Sharp. Anticipatory Head Direction Signals in Anterior Thalamus: Evidence for a Thalamocortical Circuit That Integrates Angular Head Motion to Compute Head Direction. *J Neuroscience* 15(9): 6260-6270, 1995.

[3] Fritzke, B. Growing Cell Structures--A Self-Organizing Network for Unsupervised and Supervised Learning. *Neural Network* 7(9). pp. 1441-1460, 1994.

[4] Knierim, J. J., H. S. Kudrimoti, and B. L. McNaughton. Place Cells, Head Direction Cells, and the Learning of Landmark Stability. *J Neuroscience* 15(3): 1648-1659, 1995.

[5] Konolige, K. http://www.ai.sri.com/~konolige/saphira/, 1996.

[6] Li, G. and B. Svensson. Learning Reactive Navigation Behaviors with Multiple Neural Networks. In *Proceedings of the 5th European Workshop on Learning Robot*, pp. 43-58. June 28, 1996, Bari, Italy, 1996.

[7] Li, G., B. Svensson, and A. Lansner. A Neuro-computational Model for Mobile Robot Self-orientation in Partially Known Environments. In *Proceedings of the 2th International Workshop on Mechatronic Computers for Perception and Action*, pp. 115-124. Feb. 10-12, 1997, Pisa, Italy.

[8] McMaughton, B. L., L. L. Chen, and E. J. Markus. 'Dead reckoning,' Landmark Learning, and The Sense of Direction: A Neurophysilogical and Computational Hypothesis. *J Cognitive Neuroscience* 3: 190-202, 1991.

[9] Mizumori, S. J. Y. and J. D. Williams. Directionally Selective Mnemonic Properties of Neurons in the Lateral Dorsal Nucleus of the Thalamus of Rats. *J Neuroscience* 13(9): 4015-4028, 1993.

[10] Poggio, T. and F. Girosi. Networks for Approximation and Learning. *Proc. of The IEEE*, vol.78, No.9, pp. 1481-1497, 1990.

[11] Redish, A. D., A. N. Elga, and D.S. Touretzky. A coupled attractor model of the rodent head direction system. *Network: Computation in Neural Systems* 7(4): 671-685, 1996.

[12] Roberts, S. and L. Tarassenko. A Probabilistic Resource Allocating Network for Novelty Detection. *Neural Computation* 6, 270-284, 1994.

[13] Skaggs, W. E., Knierim, J. J., H. S. Kudrimoti, and B. L. McNaughton. A Model of The Neural Basis of The Rat's Sense of Direction. *Advances in Neural Information Processing Systems 7*, ed. G. Tesauro, D. S. Touretzky and T. K. Leen. pp. 173-180, 1995.

[14] Taube, J. S. and H. L. Burton. Head Direction Cell Activity Monitored in a Novel Environment and During a Cue Conflict Situation. *J Neurophysiol* 74:1953-1971, 1995.

[15] Taube, J. S. Head Direction Cells Recorded in the Anterior Thalamic Nuclei of Freely Moving Rats. *J Neuroscience* 15(1): 70-86, 1995.

[16] Taube, J. S. and R. U. Muller. Head direction cell activity in the anterior thalamic nuclei, but not the postsubiculum, predicts the animal's future directional heading. *Soc Neurisci Abstr* 21:946, 1995.

[17] Zhang, K. Representation of Spatial Orientation by the Intrinsic Dynamics of the Head-Direction Cell Ensemble: A Theory. *J Neurosci*, 16(6): 2112-2116, 1996.

NCC curve. The other is the difference between the true robot orientation and the dead-reckoned robot orientation calibrated by the model, plotted as the CC curve. The jumping points on the CC curve in Figure 5 (a) is the place where the calibration occurs. Figure 5 (a) is computed from 40000 samples collected from one test run which is equivalent to when a real system keeps rotating 191 rounds for nearly one hour and seven minutes. When the model does not generate any calibration signal, the CC curve drifts along with the NCC curve; otherwise, the calibration will jump towards the direction that the model believes to be the right one. Though this calibration sometimes jumps the CC curve to the wrong direction, in general, the CC curve is better in the sense that it is closer to the true robot orientation than the NCC curve when the latter gradually deviates away. In Figure 5 (b), based on the data collected from ten trials of one hour turning, both the CC and the NCC curves are calculated by averaging the absolute difference between the corresponding curve and the true robot orientation. Note that each of these ten trials started from zero knowledge of the environment. It is also shown here that the CC curve has a batter performance than the NCC curve.

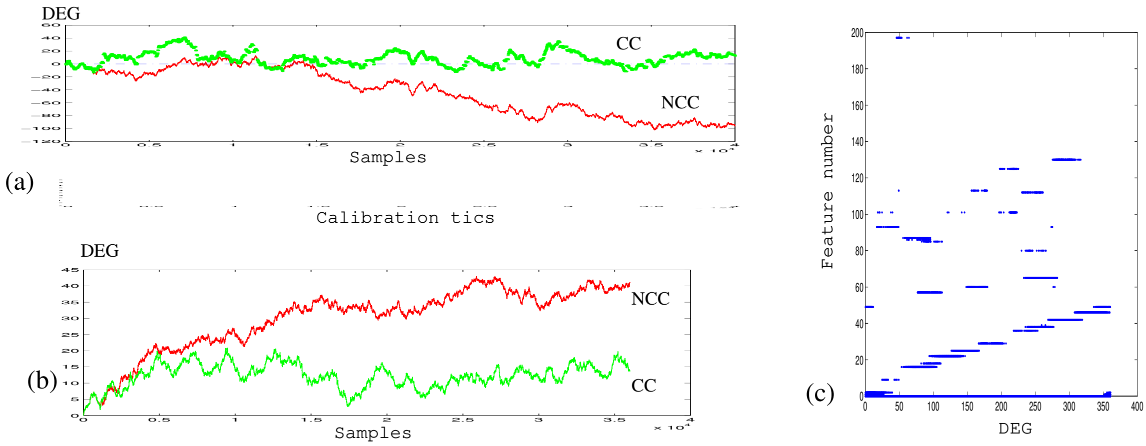

Fig. 5 (a) Observed calibration performance in one trial (b) The average calibration performance of ten trials (c) Feature to orientation distribution which corresponds to (a)

6 Discussion

Some parts of the suggested model are treated in our simulation in their simplified forms without losing the general applicability of the model. First, a head view feature vector taken from sonar readings describes an environmental feature associated with the current robot orientation. Assuming that a certain head view feature is detectable only from one specific head direction, such a choice of feature vector based on the available sensor is functionally equivalent to the feature extracted from camera images. Second, the angular velocity and the angular acceleration given in the model are considered as the result of the angular movement of the robot body, due to the availability of the sensory information in our simulator. Still this is a simplified situation of a mobile platform with a video camera mounted for all direction survey. If the head view feature were extracted from camera images, the angular velocity and the angular acceleration given in the model would have to be calculated by considering at the same time both the angular velocity and the angular acceleration of the body as well as those of the camera. Last, the head motor system in the given model is simplified in this simulation to cover only the effect of the robot body motor system. Such a simplification is also functionally equivalent to the head motor system in the model because the rotation of the robot body changes the robot orientation and hence the perceived environmental feature.

So far we have only tested the calibration performance of the model in the environment illustrated in Figure 4 (a). The environment appears to be quite simple in comparison with those usually used in robot navigation experiments, but it is much more complicated than any cue card tests which are commonly used in neurophysiological experiments with rats, because the range reading patterns do not convey landmark information to the robot as clear as the cue card does to the rats, and the sensor system together with the feature detector in the robot has much lower performance than that of rats. What we intend to investigate is if a model, which works according to a principle similar to its biological counterpart, has any ability to correct the dead-reckoned orientation when it detects a drift-away. Though the observed performance as shown in Figure 5 (a)(b) is poor in the sense that the angular resolution of the calibration is too low, it does show the calibration capability of the model which is

this lower level navigation system that the model is implemented and tested. The robot configuration used in our experiments is a three-wheel vehicle equipped with a bumper, a range-finder with N sonars arranged in a ring, and an odometer for integrating the current robot position and orientation with dead-reckoning. In addition, the current robot speed, and its angular counterpart are also available from the sensor system of the simulated robot. All sensors are sampled 10 times per second. The head view features are taken from sonar readings. A feature vector is simply described by a set of range readings with each of them pointing to a specific direction whose angle to the robot orientation is pre-defined. This vector constitutes a representation of a certain environmental feature which is relevant to the current robot orientation. For each sonar reading, reflection effects are considered, and the jitter used in range reading is about ±8 %.

The robot environment is as illustrated in Figure 4 (a), in which there are some fixed objects detectable by the robot sensor system. In this pre-defined environment, the only manoeuvre that the robot is allowed to perform is turning around. Although this appears to be a situation not very realistic, it simplifies experiments and still provides sufficient information for the model testing purpose, because at the moment the model is concerned only with orientation. Before the experiment starts, the model is in its initial state, i.e., there is no environmental feature associated with any particular direction. The starting orientation of the robot can be arbitrary, but the internal orientation at the starting point is always set to zero. The robot first turns to the left continuously until it returns to the starting point. Next the robot turns to the right continuously until it returns to the starting point. Then the robot turns back to the left again. Such a turning process continues until the experiment is stopped, giving no chance for any pair of a head direction and a perceived head view feature to dominate the learning, because the robot has not been allowed to stay in the same head direction long enough in order to be exposed to the same head view feature intensively. During turning, an association is established by self-learning between every environmental feature which is detected to be stable and a specific direction indicated by the activity of the HD units. This turning behaviour can also be observed from the robot angular velocity, as shown in Figure 4 (b). The angle that the robot turns is not a constant in any interval between two consecutive samples. It is determined by the steering decision of the low level reactive navigation system in response to the sensory information and the goal direction. In simulation, the goal direction is always set 30 degrees left to the robot orientation when the robot turns to the left, and set 30 degrees right to the robot orientation when the robot turns to the right. Therefore the actual turning angle depends only on the response of the low level reactive navigation system to the current sensory information.

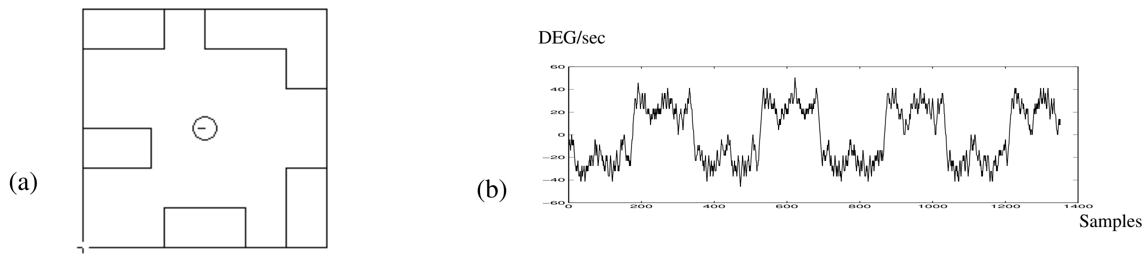

Fig. 4 (a) The environment used for model testing (b) Angular velocity

5.2 Results

In the experiments, 180 HD units were used, limiting the angular resolution to $2°$. The upper limit of the number of HVFM features was set to 1000. The number of sonars used was 12. A valid sonar reading ranges from 29 centimeter to 290 centimeter and was quantized into 10 levels with each of them covering a range of 26.1 centimeter. Therefore the size of the lattice in $LGCS$ was 12×10. The parameters of $LGCS$ are chosen as $\wp = 5$, $\kappa = 0.1$, and $\varphi = 0.2$. In the learning algorithm, parameters were chosen as follows: $c_0 = 1$, $c = 0.05$, $\delta = 10^{-4}$, $\xi = 5 \times 10^{-3}$, and $\vartheta = 5$.

The model was tested with parameters which mimic the uncertainty in a physical robot and are relevant to the orientation problem. A choice of these parameters which generates the result presented in this paper is: the jitter of the wheel encoder being set to 15%, the jitter for heading control set to 50%, and the rate of angular drift chosen to be 40%. These numbers appear to be quite high; however, our experiments show that the drift generated is still much smaller than that observed in human experiments. The system starts from zero knowledge of its environment. The experimental result is illustrated in Figure 5 (a)(b). In order to measure the performance of the model, two differences are compared. One is the difference between the true robot orientation, shown as the straight dot-dashed line in Figure 5 (a), and the dead-reckoned robot orientation without calibration, charted as the

where c is a learning rate, and $x \mathrel{+}= a$ stands for $x = x + a$.

If $f(t_i) \notin E[s(t_i)]$, i.e., the feature perceived is not the one being expected, a mismatch occurs! Three possibilities can account for such a mismatch: one is the instability of the head view feature, to which a solution is suggested in [7]; the other two are the drift of the HD system and the perception of a new feature $f(t_i)$. If $f(t_i)$ is not a new feature, a set of head direction units can be found being associated with it. This set of units suggests a hypothesis on the directions which may be inferred from the perception of $f(t_i)$, hence named as an *H*-set. For the currently perceived feature $f(t_i)$, its *H*-set is defined as

$$H[f(t_i)] = \{s_i | \langle s_i, f(t_i)\rangle \in A \wedge C(\langle s_i, f(t_i)\rangle) \geq \delta\} . \tag{8}$$

The situation $H[f(t_i)] = \varnothing$ suggests that a new feature $f(t_i)$ would be found in a head direction which has been previously exposed to other features. This is the case where (6) should be applied to initialize $C(\langle s(t_i), f(t_i)\rangle)$ and all elements $f_j \in E[s(t_i)]$, should be punished by using

$$C(\langle s(t_i), f_j\rangle) \mathrel{-}= c \cdot c_0 \cdot act(s(t_i)) \cdot act(f_j) , \tag{9}$$

where $x \mathrel{-}= a$ stands for $x = x - a$.

4.2 Drift detection

For the situation $H[f(t_i)] \neq \varnothing$, two methods may be applied. One is to find an element $s_c \in H[f(t_i)]$ closest to $s(t_i)$, i.e., if $d_j = |(s(t_i) - s_j) mod(N)|$ denotes the angular distance between s_j and $s(t_i)$ with $N = \|S\|$,

$$d_c = \min_j \{d_j | s_j \in H[f(t_i)] \wedge d_j = |(s(t_i) - s_j) mod(N)|\} . \tag{10}$$

The other one is to pick up an element $s_c \in H[f(t_i)]$ with the highest confidence:

$$C(\langle s_c, f(t_i)\rangle) = \max_j \{C(\langle s_j, f(t_i)\rangle) | s_j \in H[f(t_i)]\} . \tag{11}$$

A combination of these two methods in the following way may be more powerful than either of them. First,

$$C(\langle s_c, f(t_i)\rangle) \geq \xi \tag{12}$$

is applied to check if the s_c found by using (11) has a confidence in its association with $f(t_i)$ over a pre-defined threshold ξ. If not, s_c should be further determined by applying (10).

In each of these methods, if

$$d_c = |(s(t_i) - s_c) mod(N)| \leq \vartheta , \tag{13}$$

where ϑ is the minimum tolerance of the HD system drift, there is no HD system drift being detected. Otherwise, the internal head direction needs to be calibrated to that indicated by s_c. This can be achieved by using d_c and s_c to generate a calibration term whose influence on the shift of the tuning curve in the HD System depends on the value of $C(\langle s_c, f(t_i)\rangle)$ and $conf(\tilde{x}(t_i), f(t_i))$ calculated from (4) [7].

5 Experimental Result

5.1 Setup

For the time being, experiments on the model are conducted on a modified early version of the Saphira mobile robot simulator and its controller [5]. Its original Fuzzy-logic based control algorithm was replaced by a low level reactive navigation system [6] whose function is mainly obstacle avoidance and goal approaching. It is on top of

mated by the HD system at time t_i always predicts the next head direction [7]. The *processing circle* stands for the perception-decision-action circle. The *time changing point* is the moment when the HD system generates a head direction estimation for the next processing circle. Let $A = S \times F$ and $C: A \to \Re$, where A defines a bidirectional association between S and F, $\Re$ is a set of real numbers, and C determines the synaptic weight for each $a = \langle s(t_i), f(t_i) \rangle \in A$. The strength of a is denoted as $C(a)$. At an arbitrary time instance t_{i-1}, the HD system generates only one predicted direction, resulting in only one cell $s(t_i)$ in S being activated. Correspondingly, HVFM may either add a new cell into F or activate an existing cell in F, in response to the head view input $\tilde{x}(t_i)$. In either case, only one cell $f(t_i)$ in F is activated and all the other cells are silent, constraining that only one feature can be identified at a time.

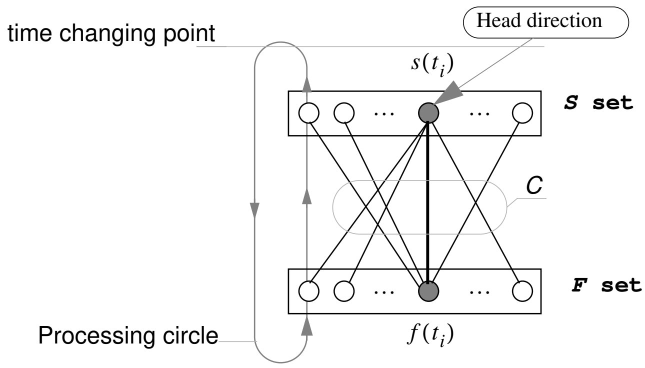

Fig. 3 Head direction to HVFM association

4.1 Learning with expectation

What is of particular interest is how to dynamically construct A and meanwhile identify C by on-line learning, which may be feasible if there exists some sort of dependence between the occurrence of $s(t_i)$ and that of $f(t_i)$. As discussed in [7], if parameters are properly chosen, the activation of $s(t_i)$ can predict the coming of $f(t_i)$, making it possible to perform the learning of C on-line in real time. The interval between the time changing point and the moment when $f(t_i)$ is generated is long enough for the system to estimate what is going to happen with its previous experience coded in C. By finding out all $\langle s(t_i), f_j \rangle$'s with C values above a given threshold δ, an expectation set (hence *E*-set) of $f(t_i)$ can be worked out and expressed as

$$E[s(t_i)] = \{ f_j | \langle s(t_i), f_j \rangle \in A \wedge C(\langle s(t_i), f_j \rangle) \geq \delta \} . \tag{5}$$

This search process may end up with either an empty *E*-set or a non-empty *E*-set. The case that $E[s(t_i)] = \varnothing$ shows that, to the best knowledge of the system, the head direction $s(t_i)$ has never been exposed to any feature before. Therefore, when $f(t_i)$ comes, the confidence in association $\langle s(t_i), f(t_i) \rangle$ should be initialized by using a Hebbian learning rule

$$C(\langle s(t_i), f(t_i) \rangle) = c_0 \cdot act(s(t_i)) \cdot act(f(t_i)) , \tag{6}$$

where act stands for the activation of a unit in either S set or F set, and c_0 is a coefficient.

The case that $E[s(t_i)] \neq \varnothing$ suggests the existence of previous experiences associating $s(t_i)$ with at least one head view feature. Now a matching process should be performed to see if $f(t_i)$ belongs to the *E*-set. If it does, $C(\langle s(t_i), f(t_i) \rangle)$ should be reinforced by using

$$C(\langle s(t_i), f(t_i) \rangle) \mathrel{+}= c \cdot c_0 \cdot act(s(t_i)) \cdot act(f(t_i)) , \tag{7}$$

new cell must be created and inserted into $LDCS$ with its weight vector located at $\aleph$. This cell recruiting process guarantees that none of the existing cells in $LDCS$ has its weight vector being equal to $\aleph$. If, on the other hand, the input $\tilde{x}$ is identified not novel, a matching needs to be performed between $\aleph$ and the weight vector of each cell in $LDCS$. For the ith cell in $LDCS$, where $1 \le i \le K$, its weight vector is $w_i = [w_{i1}, w_{i2}, \ldots, w_{id}]^T$. The distance between w_i and $\aleph$ is denoted as $dist(w_i, \aleph)$. $cell_l$ is said to be the winner of this matching, if

$$dist(w_l, \aleph) = min\left\{dist(w_i, \aleph) = \sum_{j=1}^{d} (w_{ij} - m_j)^2 \middle| 1 \le i, l \le K\right\}. \tag{1}$$

Using $dist(w_l, \aleph)$, we can carry out a second-round of novelty detection by imposing a threshold $\wp$ on $dist(w_l, \aleph)$, which is necessary in case that all elements of λ are non-zero. Those $\aleph$'s with their $dist(w_l, \aleph)$ above $\wp$ incur a new cell insertion in $LDCS$. There is always a winner in the matching between $\tilde{x}$ and $LDCS$, no matter if a new cell is recruited or not. The winner conveys the information about to which category the input $\tilde{x}$ is classified. In practice it is often desirable that the information about the distance between the input $\tilde{x}$ and the cell centre of the winner is available. Given a winner $cell_l$ with its weight vector w_l, we calculate the distance between $\tilde{x}$ and $cell_l$ in the Mahalannobis sense by tracing w_l back to L

$$Dist_M(\tilde{x}, cell_l) = \sqrt{\sum_{i=1}^{d} ((x_i - c_{i, w_{li}})/\sigma_{i, w_{li}})^2} \tag{2}$$

where $c_{i, w_{i,i}}$ and $\sigma_{i, w_{li}}$ are the centre and the semi-width of slot $s_{i, w_{li}}$, respectively. Note that w_{li} is the ith component of $cell_l$'s weight vector w_l. In the context of spatial learning, in this case the head direction calibration, what is of particular interest is the confidence of the closeness between the input $\tilde{x}$ and the centre of the hypercube to which $cell_l$ corresponds, rather than the spatial distance itself. Therefore we assume that each cell in $LDCS$ has an activation function which transforms $Dist_M(\tilde{x}, cell_l)$ into a real value between zero and one. In the choice of this activation function, a Gaussian function is preferable according to [10]. The factorizability of the Gussian makes it possible to calculate a multidimensional variable Gaussian simply by computing the product of a set of one dimensional Gaussians. This feature further justifies the calculation of the variance of each slot s_{ij} from its semi-width σ_{ij} with

$$v_{ij} = \sigma_{ij}/\sqrt{2 \cdot \log(\kappa^{-1})}, \tag{3}$$

where κ defines the confidence value at which a component of $\tilde{x}$ occurs on the boundary between two neighbouring slots. The choice of κ can be between zero and one, preferably to be a small number. Taking $r = Dist_M(\tilde{x}, cell_l)$ and substituting the $\sigma_{i, w_{li}}$ in (2) with $v_{l, w_{li}}$ which is computed with (3), a confidence in that input $\tilde{x}$ is close to the hypercube that $cell_l$ corresponds to can be computed by

$$conf(\tilde{x}, cell_l) = \exp\left(-\frac{r^2}{2}\right). \tag{4}$$

4 The Learning Mechanism

Let S be the set of all cells in the HD system, and F be the set of all possible features available at the moment in HVFM. The size of S is fixed and F is of dynamic size. In the beginning, $F = \varnothing$. Along with the time course t, if a head view input is detected novel, HVFM may add a new unit to F. The dynamic association between them is depicted in Figure 3, where we assume that the HD system is properly tuned so that each head direction $s(t_i)$ esti-

denoting the lower and upper bound, respectively. If each r_i is further quantized into k_i slots with equal or variable width, each slot in r_i can be uniquely identified by $s_{ij} = \langle c_{ij}, \sigma_{ij} \rangle$ with its centre marked by c_{ij} and its semi-width denoted by σ_{ij}, with $j = 1 \ldots k_i$. A lattice structure $L = \{ s_{ij} | (j = 1 \ldots k_i) \wedge (i = 1 \ldots d) \}$ in d-dimensional hyperspace is thus defined. Further we introduce $\lambda = \{ \lambda_{ij} | (j = 1 \ldots k_i) \wedge (i = 1 \ldots d) \}$ as a measurement of the experience of lattice L in such a way that $\lambda_{ij} = 0$ if s_{ij} has not been hit by any $\vec{x}$; and $\lambda_{ij} = n$ if s_{ij} has been hit n times by some $\vec{x}$'s. An s_{ij} is said to be hit by an $\vec{x}$ and hence its experience value is increased by one, if and only if the ith component of $\vec{x}$ falls into s_{ij}. In the case that the ith component of $\vec{x}$ appears at the border of two neighbouring slots, say, s_{ij} and $s_{i, j+1}$, the slot with the higher experience value will win the bid. If it happens that the two slots have equal experience values, a winner has to be selected randomly.

Now we can perform a matching between $\vec{x}$ and L. As shown in Figure 2, the matching process proceeds along each row of the lattice L. The ith component x_i in $\vec{x}$ is compared with every slot in the ith row of L. Among them only one slot is hit, say s_{i, m_i}, with $1 \le m_i \le k_i$. Here m_i marks the slot which is hit by x_i and the change of λ_{i, m_i} sets a novelty detecting variable n_i in such a way that n_i is set to one if λ_{i, m_i} changes from zero to one, otherwise it is set to zero. The results of this matching process are thus stored in a label vector $\aleph = [m_1, m_2, \ldots, m_d]^T$ and a novelty detection vector $N = [n_1, n_2, \ldots, n_d]^T$. Vector $\aleph$ defines, in the d-dimensional feature space, a hypercube that responds to the current input $\vec{x}$. Vector N indicates those dimensions in which some components of $\vec{x}$ are detected to be novel. For an input $\vec{x}$ to be identified as novel, the criterion of choice may depend on specific applications. Two extreme examples are that $\vec{x}$ is classified to be novel if there exists at least one non-zero component in N, and that $\vec{x}$ is taken to be novel if N has no zero component. In the middle a percentage or a specific subset of the components of $\vec{x}$ can be chosen to distinguish the novelty of an input. Another alternative is to measure the Mahalanobis distance between the input $\vec{x}$ and the centre of the hypercube which is defined by vector $\aleph$. If this distance is longer than a pre-defined threshold φ and meanwhile N has non-zero components, we can take $\vec{x}$ as novel. The representation capacity of the lattice introduced here is quite large. In the case that all dimensions have the same number of slots, say $k_i = p$ for all i, altogether p^d different hypercubes can be constructed, with each of them being uniquely determined by its $\aleph$ vector.

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \vdots \\ \rightarrow \end{matrix} \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1k_1} \\ s_{21} & s_{22} & \cdots & s_{2k_2} \\ \vdots & \vdots & \cdots & \vdots \\ s_{d1} & s_{d2} & \cdots & s_{dk_d} \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \vdots \\ \rightarrow \end{matrix} \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_d \end{bmatrix} \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_d \end{bmatrix}$$

Fig. 2 The matching process between $\vec{x}$ and L

3.2 A lattice-based dynamic cell structure

Given a lattice L, a dynamic cell structure DCS is said to be a Lattice-based Dynamic Cell Structure ($LDCS$) if each cell in DCS has its weight vector W located at $\aleph$ when the cell is created in response to the detection of a novel input $\vec{x}$. Since W is located at $\aleph$, a set of slots which defines a hypercube in which the input $\vec{x}$ occurs in the input hyperspace can be uniquely addressed by tracing W back to L.

Suppose that at the moment, there are K cells in a $LDCS$ and a new input $\vec{x}$ is presented. The input $\vec{x}$ is first matched with L, resulting in a label vector $\aleph$ and a novelty detection vector N. Then a novelty detection process is performed by comparing N with a certain novelty detection criterion. If the input $\vec{x}$ is determined to be novel, a

Extraction. This generates a feature description vector $\vec{x}(t)$ which is presented as input to the Head View Feature Map (HVFM). The function of HVFM is to categorize and to classify a sequence of input feature vectors by self-organized learning. HVFM generates an output $f(t-1)$ as a result of feature detection. HD has two groups of input: the detected feature $f(t-1)$ from HVFM gated by a learned weight matrix W, the angular velocity $\dot{\theta}(t-1)$ and the angular acceleration $\ddot{\theta}(t-1)$ which may result from body movement or head rotation. The direction predicted by the HD System can also be used to generate an expectation $E[f(t)]$ of the next feature to be detected. The learning process adapts W by performing a matching between this expectation and the actual feature perceived. We assume that the dynamics of the HD System makes the tuning curve rotate by following $\dot{\theta}(t-1)$ and $\ddot{\theta}(t-1)$, and that the tuning curve can quickly shift in response to the detected head view feature if the feature is perceived stable.

For the time being, we follow [17] to model the HD System. Although this HD system model is less biologically plausible than the model in [11], it takes into account the effect of visual cues on the shift of the tuning curve. We consider this feature important because the mechanism by which the visual cue influences the tuning curve is the main function to be investigated in this work. In addition, there are two critical issues in this model. One is the head view feature detector, denoted in Figure 1 as HVFM. The other is the learning mechanism used.

3 The Head View Feature Detector

The result of head view feature detection should play a major role in determining the shift of the tuning curve, provided that the detected feature is believed to be reliable according to past experience. Considering the dynamic nature of the environment, this head view feature detector must be able to dynamically categorize the input head view information and to classify it. In order to perform such a task, a method which is able to incrementally construct an internal representation R of a partially exposed external feature space F is highly desirable. This puts forward a couple of constraints: (1) since R has to be generated and maintained in a perception-decision-action cycle, the R of choice should be able to capture the distribution of the input data without the help of an off-line training phase; (2) the construction of R is not entirely based on F; instead, the information available for R to be adaptively constructed is the current sample of F and the existing categories in R; (3) the sample from F does not appear randomly; rather, its arrival depends on the task being performed; (4) the chosen R should generate a stable classification result upon the same feature sample regardless of the number of times that other data have appeared.

HVFM is the place where the internal representation of the external world is dynamically constructed and maintained. There are two alternatives for implementing HVFM. One is the Dynamic Cell Growing Structure based on Radial Basis Function kernel [12] and another is the Growing Cell Structure based on Self-organized Feature Map [3]. In spatial reasoning a perceived environmental feature is in general associated with a specific location or a certain orientation. The method proposed in [3] can be problematic in this situation because in this model the existing category cells in a network expand to cover the input feature space by moving the centre of a category which is coded as an input weight vector. Our experiments on these two models have shown that networks of this kind are not ideal carriers of the dynamic internal representation R due to their intrinsic property of moving the kernel centre or the cell position dynamically towards the most recently exposed input data. These methods inherently demand a training phase in which the network can be exposed intensively to the training data so that the network with dynamic cell-growing capability can capture the distribution embedded in the input data. Another presumption of applying these methods is that the input data is typically assumed to show up randomly. This is a condition that is usually not true in robot perception [7]. Therefore, we need a clustering method which can gradually cover the whole known feature space F and still keep the centre of each learned category relatively stable.

3.1 A lattice structure for hypercube construction and novelty detection

We assume that the head view feature can be expressed by a d-dimensional vector $\vec{x} = [x_1, x_2, \ldots, x_d]^T$ and the valid range of each dimension can be pre-determined by $\Re = [r_1, r_2, \ldots, r_d]^T$, where $r_i = [b_i, e_i]$ with b_i and e_i

curve, which is distributedly represented by the activation of the HD units, rotates in response to the signal from the vestibular system, namely, the angular velocity and the angular acceleration; last, the tuning curve presented by the part which emulates the behaviour of anterior thalamic nucleus predicts the future head direction if the head is rotating. The question addressed in these works is how to model biological head direction systems. The results are encouraging, but the mechanism which accounts for the observation that the internal head direction is strongly influenced by visual cues remains not well understood. Experiments on human orientation ability have shown that the estimated direction deviates quickly when a person with both eyes blind-folded is asked to rotate many times. It seems that without the visual cue, the performance of the biological HD system per se is poor. The data from behavioural experiments on freely moving rats also suggest the importance of visual cues to the maintenance of a correct head direction in HD systems. Therefore it is very important to understand how a living animal uses visual cues to calibrate the direction estimated by the HD system, and what kind of learning mechanism is involved in performing orientation tasks.

Determination of orientation is also a part of mobile robot localization which is critical for conducting any form of navigation tasks. In most artificial systems designed for autonomous mobile robots to navigate in indoor environments, localization is performed either by dead-reckoning or by aligning some sort of internal representation of map information with the perceived constellation of a set of pre-defined landmarks. In a partially known environment, methods based on pre-defined landmarks may not always be feasible because the pre-defined landmark in the known part of the environment may not be visible in the unknown part. Though dead-reckoning can be used, it is proved very sensitive to odometric drift. The artificial systems being built so far work quite differently from the biological systems. The performance of the latter is much more robust with regard to partially known or changing environments. Our understanding of the biological system is still in its infancy. Building computational models to emulate the behaviour of biological systems helps to increase our understanding of biological systems, and therefore may help to bridge the gap between artificial systems and biological systems.

Based on the existing observations in biological HD systems, we have proposed a computational model for self-orienting, which incorporates an artificial head direction system, a head view feature detector and a mechanism that associates every detected head view feature with a certain head direction by performing on-line learning [7]. We intend to investigate if such a model, which works according to a principle similar to its biological counterpart, has any ability to correct the drift of the HD system. Since the details about the general aspects of the model and the head direction system are treated in [7], this paper will only briefly describe the model and focus more on the head view feature detector, the learning mechanism and the interaction between them. The rest of the paper is organized as follows. The suggested model will be reviewed in Section 2. The design of the head view feature detector is described in Section 3. The learning mechanism used is discussed in Section 4. Section 5 covers the experiment setup and some preliminary results. Finally we conclude the paper with a discussion about the limitations of the model and the possible further development.

2 A Model for Self-orienting

As shown in Figure 1, a perception-decision-action processing circle which is commonly seen in autonomous mobile robots has been adopted.

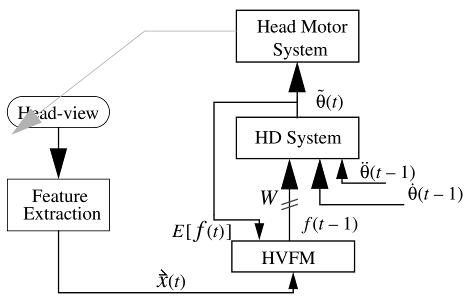

Fig. 1 A neuro-computational model for head orientation control

The HD System emulates the function of the head direction cell ensemble. The output of the HD system is assumed to predict the next preferred head direction $\tilde{\theta}(t)$, which is then used by the Head Motor System for head driving. The Head-view resulting from the activity of the Head Motor System will undergo a process of Feature

Self-orienting with On-line Learning of Environmental Features

Guang Li[1], Bertil Svensson[1], and Anders Lansner[2]

1. Department of Computer Engineering, Chalmers University of Technology, S-412 96 Gothenburg, Sweden
2. Study of Artificial Neural Systems, Royal Institute of Technology, S-100 44 Stockholm, Sweden
email: {guang, svensson}@ce.chalmers.se, ala@sans.kth.se

7/4/97

Abstract

Evidence from recently conducted neurophysiological experiments on freely moving rats has revealed that the firing of the head direction cell ensemble predicts the future head direction in response to the vestibular input, and that visual cues have strong influence on the shift of the tuning curve represented by the firing of the head direction cell ensemble. In this paper we investigate the possibility of using the learned landmark features for the purpose of self-orienting in a partially known environment. A model is suggested that incorporates an artificial head direction system for emulating the behaviour of head direction cell ensembles in biological systems, a lattice-based dynamic cell structure for categorizing and classifying environmental features, and an expectancy-based learning mechanism which learns to associate each head direction with a certain environmental feature. Our preliminary experimental results show that the suggested model helps to correct the drift in the orientation estimated by dead-reckoning.

1 Introduction

Head direction (HD) cells have been found in postsubiculum, anterior thalamic nucleus and some other brain regions being responsible for spatial orientation function [2][9][14][15][16]. The data from recently conducted neurophysiological experiments on freely moving rats has revealed that the firing of the HD cell ensemble in postsubiculum indicates the present head direction, while the firing of the HD cell ensemble in anterior thalamic nucleus predicts the future head direction in response to the angular velocity of the animal's head [2][16]. Each HD cell has a preferred head direction at which the cell discharges maximally, and the firing rate decreases linearly away from this direction. The preferred direction of all HD cells was found distributed over a 360° range [15]. Altogether the firing of these HD cells indicates a particular direction which is relevant to both inertial and landmark information in the environment. It seems that head direction cells preferentially use landmark information from a head view as long as landmarks are perceived stable [2][9][17]. The displacement of a previously exposed landmark may shift the direction represented by the firing of HD cells, indicating that geomagnetic cues play a minor role in affecting the behaviour of biological head direction systems [2][9][15][16].

These observations present interesting insights to the possible mechanisms with which a biological system determines its head direction. Several models have been proposed to account for these observations. In [8] a model was introduced to update the head direction by using angular velocity. The idea of using a circular shift register with attractor dynamics was discussed later in [1][13]. Recently, another attractor model which emulates the tuning curve and predicts the future head direction was proposed in [17]. In addition, a coupled attractor model was suggested in [11] which shows the capability of tracking the biological data recorded from the actual head rotation while maintaining the shape of tuning curves in each of its parts.

There are some features common among these models: first, HD units arranged in circular arrays of a certain size are used to emulate the behaviour of biological head direction cell ensembles; second, a sustained tuning

Proc. AISB workshop on "Spatial Reasoning in Mobile Robots and Animals", Manchester 1997.
Technical Report Series, Department of Computer Science, Manchester University, ISSN 1361 - 6161.
Report number UMCS-97-4-1. http://www.cs.man.ac.uk/csonly/cstechrep/titles97.html