

Hippocampal Models for Simultaneous Localisation and Mapping on an Autonomous Robot

Michael Milford, Gordon Wyeth

University of Queensland

School of Information Technology and Electrical Engineering

St. Lucia 4072, Australia

milford@itee.uq.edu.au, wyeth@itee.uq.edu.au

Abstract

To navigate successfully in a novel environment a robot needs to be able to Simultaneously Localize And Map (SLAM) its surroundings. The most successful solutions to this problem so far have involved probabilistic algorithms, but there has been much promising work involving systems based on the workings of part of the rodent brain known as the hippocampus. In this paper we present a biologically plausible system called RatSLAM that uses competitive attractor networks to carry out SLAM in a probabilistic manner. The system can effectively perform parameter self-calibration and SLAM in one-dimension. Tests in two dimensional environments revealed the inability of the RatSLAM system to maintain multiple pose hypotheses in the face of ambiguous visual input. These results support recent rat experimentation that suggest current competitive attractor models are not a complete solution to the hippocampal modelling problem.

1 Introduction

The objective of this work is to produce a navigation system based on the rodent hippocampus that can perform Simultaneous Localisation And Mapping (SLAM) in medium scale real world environments using only vision and odometry information. The general approach to implementing SLAM has been to model all the aspects of the problem in a probabilistic manner. Various methods have used Kalman filters [Dissanayake *et al.*, 2000] and Markov localisation [Thrun, 2000]. We describe a system of competitive attractor networks that are used to resolve probabilistic interactions between sensor information and internal models of pose and environment. This neural approach known as RatSLAM has the potential to sidestep some of the problems facing existing techniques.

This paper focuses on the RatSLAM system¹ and the advantages it offers over conventional SLAM development to date. Current probabilistic methods for

mapping such as the EM (Expectation Maximisation) approach involve a challenging and computationally expensive algorithm especially when a fine map grid is used. Many attempts at SLAM have used either lasers or sonars as their primary sensors, not vision which is the primary sensor for both the rodents that our RatSLAM system is modelled on and also our human selves. By modelling a real biological system and using a sensor type that is actually used by rodents as their primary sensor, the RatSLAM system should be able to achieve similar navigation and mapping performance as its biological inspiration.

By using a competitive attractor network structure our RatSLAM system is able to take ambiguous visual input and maintain multiple location beliefs simultaneously. Network dynamics allow these beliefs to compete with each other as time progresses until one eventually wins. Visual input during competition can strengthen the belief in one or more of the possible locations. Path integration becomes a neural process rather than an explicit mathematical one.

1.1 Probabilistic Methods

Probabilistic localisation of a robot using the Markov assumption combined with mapping using the expectation-maximisation algorithm has been shown to be an effective way of performing SLAM on a real robot [Thrun, 1997, 1998]. Implemented on a large mobile tour-guide style robot called MINERVA the system was able to perform SLAM using laser and vision sensors in an unprepared museum environment filled with people.

To localize itself the robot attempted to estimate a posterior distribution over the space of its poses conditioned on the available observation and action data. The Markov assumption, that the past is independent of the future given full knowledge of the current state, is used to simplify the problem. By using Bayes rule, the theorem of total probability and by exploiting the Markov assumption twice the current belief state of the robot b_t is obtained.

$$b_t(s_t) = \eta_t p(o_t | s_t, m) \int p(s_t | a_{t-1}, s_{t-1}, m) b_{t-1}(s_{t-1}) ds_{t-1} \quad (1)$$

Equation 1 shows a constant η_t normalizing the product of the desired posterior and integral of the

¹ This research is sponsored in part by an Australian Research Council grant.

posterior over the last pose. The variable m is the map that the robot uses to localize itself. The expectation-maximization algorithm was used to perform mapping for this particular robot. This mapping technique attempts to estimate the *mode* of the posterior based on pose and sensor readings. The algorithm calculates a sequence of maps with higher and higher likelihoods - this forms the *expectation* step. The most likely map is then chosen - the *maximisation* step.

Other mapping techniques have been used - the SLAM algorithms employ Kalman filters but require that features can be uniquely identified in an environment [Castellanos *et al.*, 1999; Leonard and Durrant-Whyte, 1992]. Most real environments contain lots of ambiguity and do not support this assumption restricting this technique to specialized areas.

There are significant implications to how our system might interface with a vision sensor when compared to this robot. By using a laser as a primary sensor the robot received precise geometric information about distances to all objects on its scanning plane (with human body interference) - our vision system gives us no such preciseness. In fact our vision system does not even attempt to extract geometric or shape information from a scene, rather it attempts to remember and identify the *appearance* of a scene as it moves.

The occupational grid maps that are formed by MINERVA are relatively high resolution, strictly cartesian ones, achievable due to the accuracy of the laser scanner but with a high computational load (off-board computers supplemented the robot's processing power). The RatSLAM system uses a coarse representation and also does not necessarily produce a strictly cartesian map (or need to).

1.2 Biological Inspired Localisation

Experimental work has shown that there exist at least two distinct populations of cells in the rodent hippocampus known as *place cells* [McNaughton *et al.*, 1983; O'Keefe and Dostrovsky, 1971] and *head-direction cells* [Ranck, 1984; Taube *et al.*, 1990]. Place cells are associated with certain physical locations of the robot and likewise head-direction cells are associated with certain orientations of the robot.

One specific head direction model involved a population of head angular velocity cells, directional cells, and anticipatory directional cells [Arleo, 2000]. The directional cells were set up so that each was maximally activated when the robot was facing in a certain direction, with a Gaussian drop off in activation as the robot turned away from the preferred direction. The head angular velocity cells' activity was dependent on both the sign and magnitude of the rotational velocity, and the anticipatory cells anticipated the head direction cells' activity by a constant time period. Activity in all three cell types had been observed in experimental work.

In addition a place cell model was developed with each place cell being maximally activated when the robot was at a certain location, the activity dropping off in a 2D Gaussian manner as the robot moved away from that location. Activity in both sets of cells was primarily shifted around by ideothetic input (path integration). Allothetic input (visual cues in this case) was used to relocalize the Khepera robot. Results showed the robot capable of navigating the environment whilst keeping a reasonably accurate idea of its pose.

This research was carried out in a specially designed 800 mm by 800 mm white arena with barcoded walls and an unmoving distant light marker. The robot did not move in real time and offboard processing was used. Exploration and homing behaviours were alternated, with learning deactivated during homing. No evidence was presented of the robot being able to perform global localisation or recover from kidnapping.

This approach is partly the inspiration for the RatSLAM system although we use a coarser place cell network, with an allothetic module that is always turned on. The arena used was highly prepared - ours is much sparser with many ambiguous views, and we have no 'distant' bearing reference. The exploration and foraging stages are replaced in our system by one movement scheme with learning and recalibration 'always on' from initialisation.

2 The RatSLAM System

2.1 The Hippocampus

The hippocampus is generally accepted to be the basis for navigation in rodents and has been studied extensively. Experiments have shown that certain cells respond maximally when a rat is at a certain location (place code cells) and that others respond when it is orientated in a certain direction (head direction cells). Our system uses competitive attractor networks to maintain a population of head direction and place code cells.

2.2 Competitive Attractor Networks

Competitive attractor networks are a type of neural network that usually converge to a stable pattern of activation across their units. The network units can be arranged in many configurations, but generally each unit will excite units close to itself and inhibit those further away, which leads to a clump of activity known as an *activity packet* eventually dominating. Activity injected into the network near this winning packet will tend to move that packet towards it. Activity injected far away from it will create another packet that competes with the original. If enough activity is injected the new packet can 'win' and the old packet disappear. This network structure is used to represent the head direction and place code cells.

2.3 Overall System

Figure 1 shows the basic model. The robot's pose is represented by the activity in the two networks. Wheel encoder information is used to perform path integration by injecting activity into both the networks thereby shifting the current activity packets. Vision information is converted into a local view representation which if familiar, injects activity into the place code and head direction cells that are associated with that specific local view.

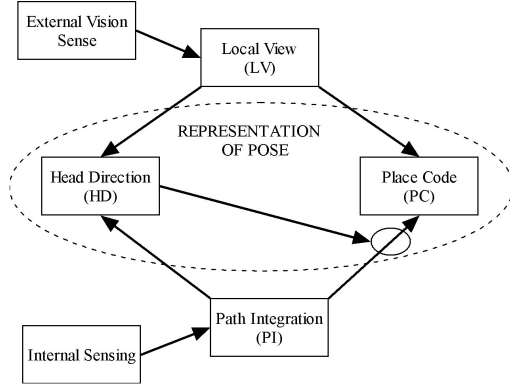


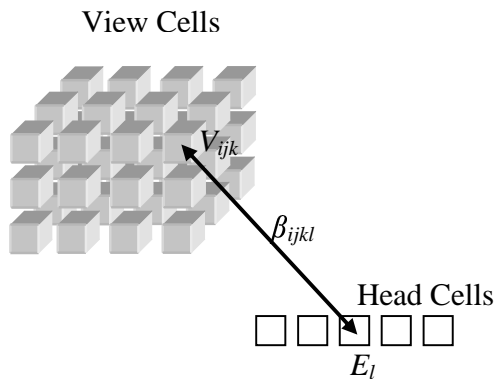
Figure 1 – Pose is represented by activity in the place code and head direction networks. This pose is updated continually by path integration and local view activity input.

2.4 Allothetic Association

The robot's camera and vision processing module can see coloured cylinders and report on their distance and relative bearing from the robot, and associated uncertainties [Prasser and Wyeth, 2003]. A three-dimensional matrix of local vision cells encodes the cylinder colour (type), distance and bearing. Activated local view cells are constantly being associated with the head direction and place code cells that are highly activated at that time through strengthening of weighted connections between them.

The following explains the association process for a head direction cell. The first three subscripts of β reference where in this three dimensional local view cell matrix the connection starts and the fourth subscript the head direction cell to which it goes. Here the weight is increased by an amount proportional to the product of activity in the head direction cell E and view cell V minus some decay constant.

$$\beta_{ijkl} = \beta_{ijkl} + \varsigma(E_i V_{ijk} - \delta_{decay}) \quad (2)$$



2.5 Network to Physical Transformation

Together the head direction and place code networks model the three parameters describing the robot's pose – two cartesian co-ordinates and an orientation variable. Population vector decoding [Arleo, 2000] has been used

to translate network activity into a perceived physical location of the robot. In this case simply picking the most highly activated head direction and place code cells was found to give an adequate network to physical co-ordinates transformation. The direction and position associated with these cells was used as the perceived robot's pose.

3 Head Direction Cells

We model a system of head direction cells, with each cell tuned maximally to a certain preferred robot orientation. The activity of each cell drops off in a Gaussian fashion as the robot orientation moves away from its preferred direction. When the activity of all the head direction cells is viewed as a bar graph one can see an *activity packet* representing the current perceived direction of the robot. This activity packet smears outwards when there is rotational movement but becomes a narrow focused peak in the absence of it.

To achieve this behaviour the cells are fully interconnected by weighted links making the cells part of a type of competitive attractor network. Iterations of the head direction network consist of the following sequence of actions:

1. Activity input from the local vision cells
2. Competitive attractor network dynamics
3. Activity input from the path integration process

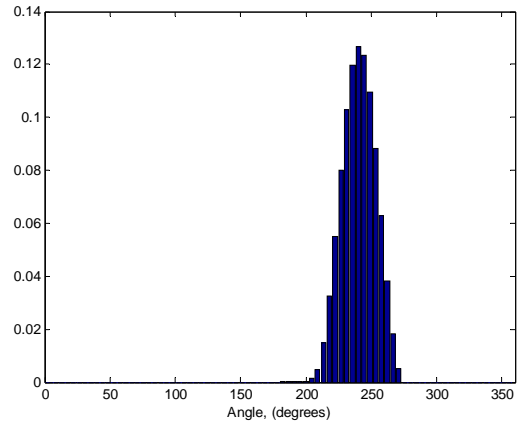


Figure 2 – The head direction network consists of 81 cells with preferred directions spread evenly over 360 degrees. Total activity in the network is normalized.

An example activity packet in the head direction network is shown in Figure 2. This packet shown is for a rotating robot with it smeared over a range of about sixty degrees. When stationary the packet narrows to cover only about fifteen degrees. This smear has a smaller spread than suggested by activity measured in rodent hippocampi but gave us better results.

3.1 Allothetic Orientation Calibration

Path integration in the system must have a companion that compensates for odometry error that compounds over time. One method of keeping the error bounded is to use visual input to calibrate the robot's idea of pose. When the robot sees a scene previously encountered, the current perceived position can be compared with the position

associated with that scene.

In this case whenever a familiar scene is encountered activity is injected into the head direction cells associated with it. However if the robot is rotating the associated activity from the local view must be shifted somewhat according to the rotation speed and direction before being injected into the head direction cells. The amount of shift required is linearly related to the rotational velocity by a constant value.

$$e = k_p w \quad (3)$$

The current view cell activity V is projected through the association weights β to add to the activity injected through a vision influence constant α into the head direction cells E .

$$L_i^t = \sum_{j=0}^X \sum_{k=0}^Y \sum_{l=0}^Z \beta_{ijkl} V_{jkl}^t \quad (4)$$

$$E_i^{t+1} = E_i^t + \alpha L_i^t \quad (5)$$

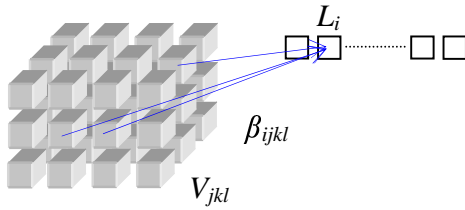


Figure 3 – Schematic showing local view cell activity being projected through association links to a cell L in an array storing allothetic input. Activity in L is injected directly into the head direction cells.

3.2 Competitive Attractor Dynamics

In most environments there exist situations where an agent may exist in multiple ambiguous locations. A competitive attractor network can maintain several peaks of activity representing these positions that can be reinforced by further allothetic input to the system. This type of network allows peaks close together to complement each other whilst peaks far apart compete.

Each cell is connected to every other cell and itself by excitatory weights. The self-connected weight is the largest, with values dropping off in a discrete Gaussian manner for connections to more distant cells. Equation 6 shows a discrete Gaussian distribution of variance s^2 , with normalization across the distribution, being used to form a matrix δ that stores the excitatory weight values between all of the cells.

$$u = \sum_{i=1}^n e^{\left(\frac{i^2}{s^2}\right)} \quad (6)$$

$$\delta_{ij} = \frac{1}{u} e^{\left(\frac{(i-j)^2}{s^2}\right)}$$

The activity in each of the head direction cells is

projected through the appropriate excitatory weight δ to the head direction cell E .

$$E_i^{t+1} = E_i^t + \sum_{j=0}^N \delta_{ij} E_j^t \quad (7)$$

Global inhibition based on the total activity in the network is performed in equation 8 – this ensures that without external input smaller activity packets will gradually ‘die out’.

$$v = \sum_{i=0}^N E_i^t \quad (8)$$

$$E_i^{t+1} = \max(E_i^t + (1.0 - v), 0)$$

3.3 Ideothetic Input

The final step is to input the activity obtained from path integration of the odometry input. The angular velocity is used to find the index shift required for projection of the current head direction activity packet to its anticipated future location. Since the shift is a non-integer but the cells are discrete entities, when projecting from a current head direction cell to a future anticipated one, the activity is split across two cells.

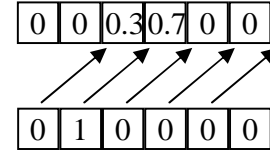


Figure 4 – The discrete nature of the cell representation means activity that is shifted by arbitrary amounts must be redistributed across cells.

Following input of ideothetic activity, one last multiplicative normalisation step is performed.

$$E_i^{t+1} = \frac{1}{\sum_{i=0}^N E_i^t} E_i^t \quad (9)$$

Figure 5 shows a snapshot of activity in the three components of the head direction network, the allothetic input, the ideothetic input activity and the current head direction cell activity. Although the ideothetic input activity packet is only slightly offset from the current head direction activity packet, this is sufficient to produce the desired movement.

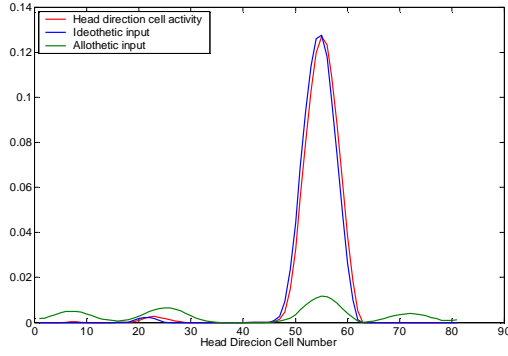


Figure 5 – Ideothetic input is a projection of the current head direction activity forwards in time, causing the current activity packet to shift, in this case to the left. The allothetic input mainly reinforces the head direction peak although there is some visual ambiguity represented by activity input in other locations.

4 Place Code Cells

We model a matrix of place code cells, each cell tuned to be maximally activated when the robot is at a specific location. A coarse representation is used, with each place cell representing a physical area of approximately 250 mm by 250 mm. This is quite different to most previous approaches involving place cells that have used a much finer resolution. To illustrate the coarseness, the robot covers an area roughly represented by only four place cells. The place field size to robot size ratio compares favourably with that observed in nature – the stable place fields formed by our place cells have areas of about 0.3 m², compared to a robot area of about 0.16 m², a ratio of about two. Place fields recorded in healthy rats have an area of about 0.056 m² [Save *et al.*, 1998], compared to a rat ‘area’ of about 0.015 m², a ratio of about three.

To simulate a system of place code cells a competitive attractor network arrangement similar to that used for the head direction network is used. The cells are arranged in a two-dimensional matrix with full excitational interconnectivity between all cells. The excitational weights are created using a similar discrete Gaussian distribution as in (6) but in two dimensions rather than one. The activation of each place cell drops off in a two-dimensional Gaussian fashion as the robot moves away from the cell’s preferred location.

Iterations of the place code network consist of the following sequence of actions:

- Competitive attractor network dynamics
- Activity input from path integration process
- Activity input from allothetic sources

4.1 Competitive Attractor Dynamics

The dynamics were a two-dimensional version of those used for the head direction cells. A two dimensional Gaussian distribution was used to create the excitatory weights ϵ for each cell. These weighted connections were used to project the activity from each cell C in the I by I matrix to the cell being updated.

$$C_{ij}^{t+1} = C_{ij}^t + \sum_{k=0}^I \sum_{l=0}^I \epsilon_{ijkl} C_{kl}^t \quad (10)$$

The main difference between the place code and head direction networks is that there was no wraparound for the place code network – cells at one edge of the matrix did not project strongly to cells at the opposite edge of the matrix. To minimize boundary effects a buffer layer of cells was used around the edges of the cell matrix.

4.2 Ideothetic Input

The current activity packet in the place code network is projected to its anticipated position depending on the velocity of the robot.

Because the amount this anticipated activity is shifted from the current state is not usually a whole number of cells but rather a fraction, each cell’s activity is projected such that it is split between four cells (double the number for the head direction network because we have gone from a 1D to 2D network structure).

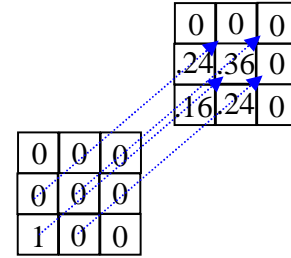


Figure 6 – Projected ideothetic activity is split fractionally between cells.

The robot orientation is not usually known exactly – rather it is represented by a number of highly activated head direction cells. As such the ideothetic input is treated on an individual basis for each head direction cell and its associated activity level.

$$C_{ij}^{t+1} = C_{ij}^t + \sum_{z=0}^{z=N} E_z^t \sum_{x=0}^{x=1} \sum_{y=0}^{y=1} f(v) C_{x+i_i, y+j_j}^t \quad (11)$$

$$v = f(V_{abs}, z)$$

The activity levels in each of four cells C are multiplied by a fractional function dependent on velocity v , which is itself a function of head direction cell index z and the absolute velocity of the robot V_{abs} (the four cells are selected based on i_s and j_s which are functions of v). This is then multiplied by the activity level in each head direction cell E . The sum of this process repeated for all head direction cells is used as ideothetic input and added to the current place code cell activity C .

4.3 Allothetic Input

The need for calibration of the robot’s position using visual input is especially important when considering its location, with the normal problem of slight errors in orientation quickly leading to large positional errors as translation occurs. Unlike for the head direction system, there is no compensation for current movement of the robot when injecting allothetic input into the place code cells. The place code cells are relatively more coarse than the head direction cells and as such are not as sensitive to allothetic input slightly lagging where it should be.

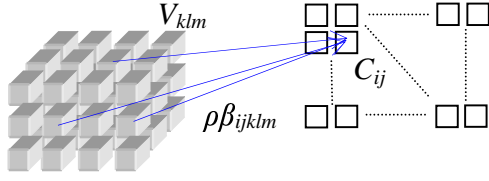


Figure 7 – Activity in the view cells is projected to each place code cell through weighted connections.

The place code activity level C is increased by the activity projected from the view cells V through the association weights β and the overall influence constant ρ .

$$C_{ij} = C_{ij} + \rho \sum_{k=0}^X \sum_{l=0}^Y \sum_{m=0}^Z \beta_{ijklm} V_{klm} \quad (12)$$

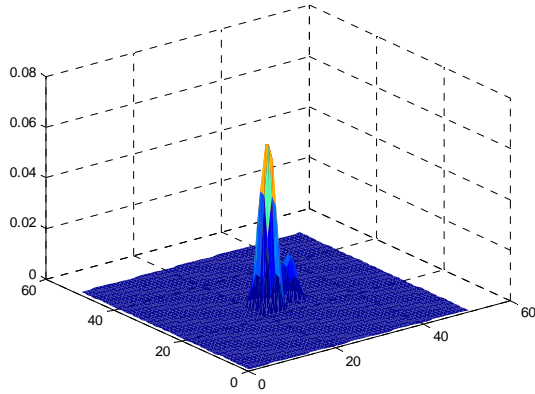


Figure 8 – The place code cells are arranged in a 51 by 51 square grid. Total cell activity is normalized and so cell activity can be thought of as a rudimentary probability of being located at the cell's corresponding physical location. Here there are two possible locations with one being much more probable.

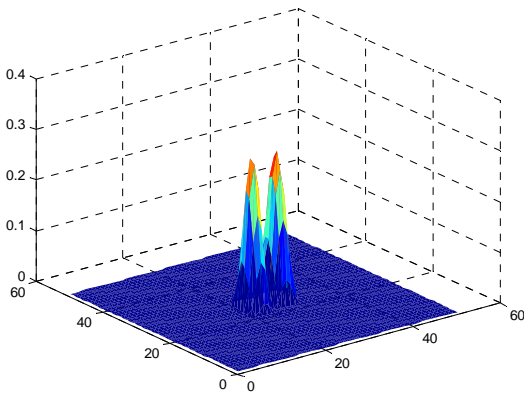


Figure 9 – Allothetic input is projected into the 51 by 51 grid of place code cells whenever visual input is received. This figure shows the corresponding allothetic input for Figure 8, from a vision perspective there is equal probability of being located in two different locations.

Figure 8 shows a sample of the activity levels in

each place cell (it shows activation levels in a 51 by 51 grid of place cells). There is one major peak representing the most probably location of the robot, with another small peak that is starting to grow. This small peak has not been created through ideothetic input, rather by allothetic input as shown in Figure 9. From a purely visual perspective, the robot is equally sure it is in two different size locations, represented by the two peaks of the same size.

In this particular case the small peak in Figure 8 will compete with the larger peak but will not win unless further allothetic input is received supporting the smaller peak and not the larger one. This illustrates the ability of the place code network to deal with ambiguous visual input. Figure 10 shows that view cells through training can end up projecting to more than one location in the place cell network.

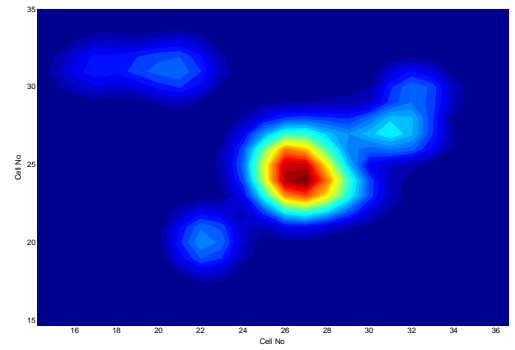


Figure 10 – Weightmap showing strength of connections from one particular view cell to a rectangular section of place cells. The association between this vision cell and the place code network has been trained during exploration such that activity in this view cell projects strongly into one area of place cells and weakly into several other areas.

5 Experimental Setup

The system is implemented both in simulation and on a robot. A Pioneer2-DXE robot from ActivMedia carrying a 400 MHz AMD K6-2 processor performs on board processing of the vision and interfaces with the motors. A 1.1 GHz laptop connected to the robot by an ad hoc wireless link provides the main processing grunt and runs the two networks which take the majority of the processing power. A CCD camera is mounted at the front top of the robot with an effective field of view of about 40 degrees.

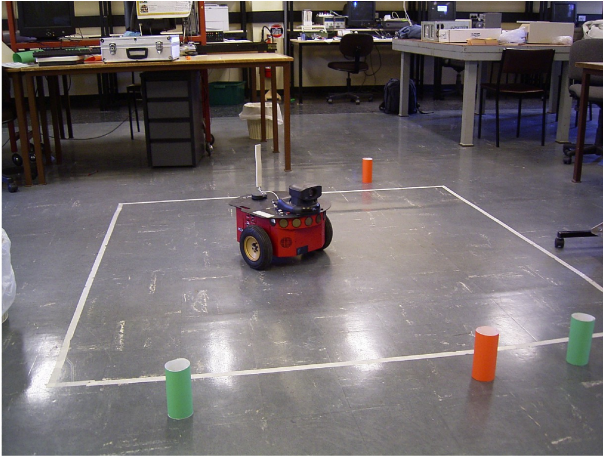


Figure 11 – The Pioneer robot in the two by two metre experimental arena with coloured cylinders arranged around the perimeter.

The initial arena is a flat two by two metre area of linoleum in our lab with masking tape marking the boundaries. Just outside the arena coloured cylinders are placed in various configurations, one such shown in Figure 11. The vision system reports colour, distance and egocentric orientations of cylinders. Because the vision system is trained to recognize rectangular areas of a solid uniform colour it occasionally picks up background objects such as posters and people's socks. The competitive attractor dynamics deal easily with this low intensity 'noise'.

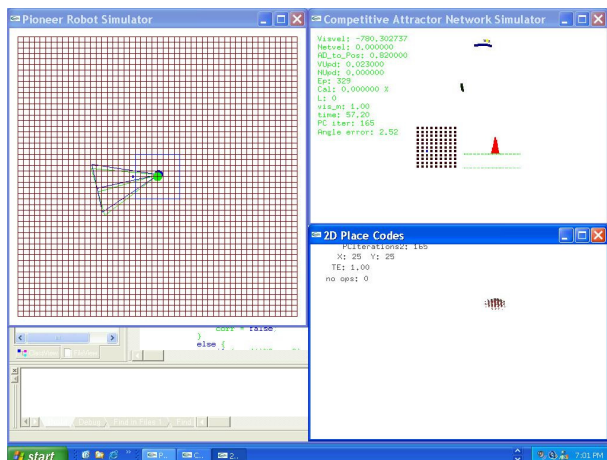


Figure 12 – The simulator displays the simulated and perceived robot positions and orientations, and the activity levels in both the head direction and place code networks.

The simulator mimics the kinematics of the pioneer robot and the vision system. Although no provisions for sensor noise and vagaries of the motor control system have yet been built into the program, it simulates the robot sufficiently well for it to be used in system development. It also models the thresholds for movement on the real robot such as how close the robot tries to get to a specified pose before moving onto the next command.

6 Results

Three stages of testing were carried out. The first involved seeing if the robot could self-calibrate the parameters it used to convert odometry information into path integration in the networks. The second was attempting one dimensional SLAM in a cylinder based world. The last stage involved two dimensional movement and testing the ability of the robot to stay localized.

6.1 Odometry Calibration – 1D

The robot's ability to calibrate the parameters that convert wheel encoder information into path integration was tested both in simulation and on the robot. These tests were run with the robot spinning on one spot in a cylinder world. Extending this to two dimensions and translational movement is harder but can be reasonably expected to be possible if the 1D case is viable.

Simulation

The system reached the stage where it could perfectly self-calibrate in a noiseless environment with perfect odometry using a filtering technique on the vision, as expected.

Robot Platform

Due possibly to control parameters that have not been tuned optimally, rotation of the robot is somewhat jerky although it's average angular velocity over a long time period is very close to that commanded. The best results were achieved when the filter gain K was set to relatively low values to compensate for the variability in the visual velocity.

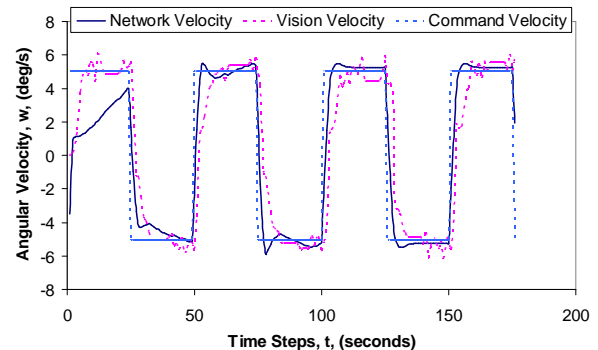


Figure 13 - Odometry calibration was achieved within about fifty seconds of the test beginning. Once corrected the calibration parameters stay unchanged as evidenced by the last two cycles in the graph.

Figure 13 shows the network self-calibrating itself as the robot spins at approximately five degrees per second in 125 degree alternating turns. The actual encoder velocities from the robot vary by about 300% when supposedly doing a constant velocity turn and as such are not reported here. The vision velocity curve is filtered.

Path integration in the head direction network is initially very wrong but corrects itself significantly by twenty five seconds into the trial and is accurate after fifty seconds. The long calibration times are due to the sparsity of the cylinders – only two in this particular trial.

6.2 One-Dimensional SLAM

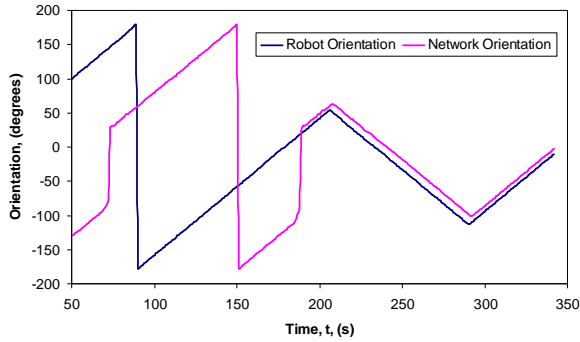


Figure 14 – The robot is able to correct its orientation after a revolution. Note the false correction in orientation that occurs at around 80 seconds.

By spinning the robot through a number of large angular turns, in an arena with two cylinders of the same colour, the robot is able to perform one-dimensional SLAM. Vertical segments of the network curve above (apart from those from -180 to 180 degrees which represent wrapping) occur when the robot is relocalising off visual input. At about 80 seconds into the experiment, the robot corrects its orientation but not to the correct direction – this is due to the ambiguity of having two identically coloured cylinders. A second reorientation jump at about 200 seconds matches up the perceived and actual orientations of the robot, which it maintains for the rest of the experiment.

6.3 Localisation Ability in 2D

Simulation

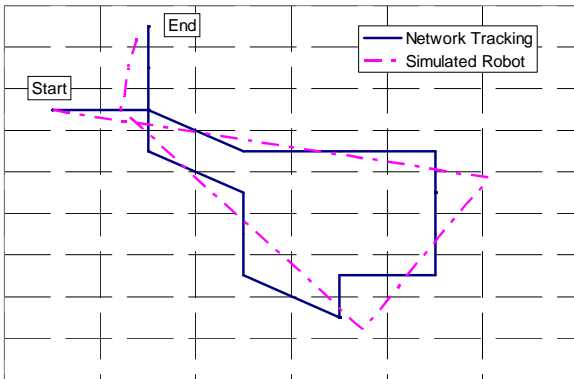


Figure 15 – Over the short term the system is able to track the robot's position effectively. Each grid square is the approximate size of the area represented by one place cell, 250 mm by 250 mm.

In simulation the robots are able to track the robot's position quite well for short periods of time as shown in Figure 15. Each grid square is 250 mm by 250 mm and is represented roughly by one place code cell. The network manages to keep the robot localised to within roughly one place code cell of its actual position for the entirety of the test. The system was also able to recover when kidnapped although successful relocalisation did not always occur, depending mainly on how long it was after kidnapping before a familiar scene was seen.

On longer runs the network's tracking ability proved to be unstable. Over the period of an hour the robot became lost and its perceived location moved well outside its two by two metre arena.

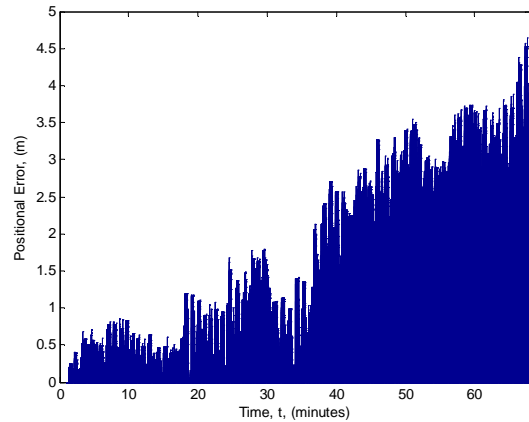


Figure 16 - Positional tracking error remains small for about 20 minutes before starting to grow in an unbounded manner.

Actual Robot

The system was implemented on the real robot and similar results to in simulation were obtained, although the trial could not be run for as long. The RatSLAM system was able to keep the robot localized for short test durations only. Because the robot was moving based on its perceived position, not its actual position, a small error in pose usually compounded quite rapidly and the tests had to be terminated before the robot collided with objects outside the arena.

7 Discussion

7.1 Overview

The results revealed two problems with the RatSLAM system which are discussed here, with a solution presented for one of them. Following this the discussion moves onto biologically plausible ways in which rats may or may not get around the major of these problems. We conclude with where the work can go from here.

7.2 Analysis of Results

There are two flaws in the RatSLAM system developed in this paper, one minor and one very significant. Because the head direction networks and place code networks were separate entities, when the robot relocalised its position through viewing of a familiar scene, it didn't necessarily snap back to the associated robot orientation as well. The opposite occurred too, reorientation but no relocalisation.

Arleo [Arleo, 2000] dealt with this problem by having a place code network which could relocalise from a familiar scene, combined with a head direction network with an extra recalibration constraint. The head direction network not only needed a familiar scene to recalibrate orientation *but also* that the robot perceive itself to be located at a position associated with that scene. This solves the localisation problem and allows the robot to perform position tracking assuming it has a good enough exploration behaviour, and it is thought it would also allow the robot to recover from kidnapping, although no

results were presented. If the robot was kidnapped it would wander until seeing a familiar scene. At that moment the robot would not immediately re-orientate itself since its perceived location would not be the one associated with that scene. However it would relocalise its position, *after which* it could then re-orientate itself since it now had the perceived position associated with the scene. The movement behaviour would need to realize it was relocalising and stop to allow the reorientation to occur.

The major flaw with the system is that it cannot maintain multiple pose hypotheses. Although it can maintain multiple *location* hypotheses, each place code activity packet is associated with the same head direction activity – different place code packets cannot have different orientations associated with them. If ambiguous visual input suggests two possible poses, these poses cannot be verified or denied through further robot movement and visual input because path integration will move *all* place code packets in the same net direction of all possible head directions.

In any environment where the robot regularly has insufficient allothetic information during stages of movement, and if the visual information it does obtain can be ambiguous, this will severely limit its performance.

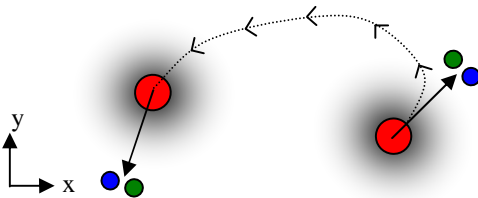


Figure 17 – Shows the overhead view of the robot (red circle) at two moments in time where it has associated the place code activity packet (grey shading) and orientation (arrow) of the robot with an identical view of two cylinders. In Figure 18 the robot re-encounters this scene.

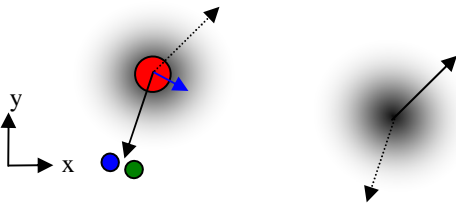


Figure 18 – at some later time the robot encounters the same scene and injects activity supporting the two possible locations and two possible head directions. The robot needs to move to receive further visual input and determine which position and orientation is correct. There is only one head direction network containing both orientation possibilities. Therefore each activity packet in the place cells is associated with *both* head direction possibilities. Ideally to maintain multiple pose hypotheses under further movement the location possibility in the left of diagram would move downwards. However because it now has a second direction associated with it (dotted upwards arrow) that should really only be associated with the other location possibility it will move in the net direction (blue arrow).

7.3 Biological Comparison

Are rats able to maintain multiple pose hypotheses or do they achieve their navigation ability without needing to?

Experiments in which rats are trained to search at the midpoint of two markers but are then given only one in the test show the rat searching in two possible locations relative to the single marker, as if treating that marker as first being one and then the other of the original two [Redish, 1999]. This suggests the possibility of two pose hypotheses being maintained given ambiguous visual input but to confirm one would need to see two activity peaks existing in the place cells - other explanations are possible.

When exposed to environments that have sections that are identical in both appearance and physical orientation, rats seem to be able to construct two distinct hippocampal maps that overlap yet are significantly different [Skaggs and McNaughton, 1998]. When rats were effectively kidnapped from one identical region to another, their place fields initially fired as if the rat had not been kidnapped but then in some of the rats place cell firing changed to represent the rat's true physical location - partial remapping occurred when faced with an ambiguous situation.

When faced with a conflict of distal and local cues the entire spatial representation in the hippocampus can split into opposing representations [Knierim, 2002]. This work shows that several predictions that current competitive attractor models make do not happen, although it points out that there is a wealth of specific evidence and theory that supports them as well.

The author has not yet come across any place cell activity recordings from a real rat faced with ambiguous visual input that show multiple peaks being maintained for any length of time. Although it is possible that a rat can switch between multiple maps maintained by the same set of place cells, no evidence of them concurrently updating multiple maps has been found. Rather rats seem to keep only one map in 'active' memory at any one time, using path integration to keep it updated and possibly switching maps if prompted by visual input. Whether maps remain unchanged between being unselected and reselected is unknown.

7.4 Conclusion

From the robotocist's perspective this RatSLAM system has shown a limitation in the ability of a place code – head direction type system to perform effective navigation and localisation, not to mention SLAM. It shows that rats, although possessing impressive navigation ability are nonetheless probably limited by this inability to maintain *and* update concurrently multiple hypotheses of pose. In the context of developing a neurologically inspired, effective robotic SLAM system, further extensions of the RatSLAM system that overcome this limitation are needed.

8 References

- [Arleo, 2000] Angelo Arleo. Spatial Learning and Navigation in Neuro-Mimetic Systems: Modeling the Rat Hippocampus. *Information Science*. Milan, University of Milan, 198,2000.
- [Castellanos *et al.*, 1999] J.A. Castellanos, J.M.M. Montiel, et al. "The SPmap: A probabilistic framework for simultaneous localization and map building." *IEEE Transactions on Robotics and Automation*, vol. 15(5): 948-953, 1999.

- [Dissanayake *et al.*, 2000] G. Dissanayake, H. Durrant-Whyte, et al. *A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem*. 2000 IEEE International Conference on Robotics and Automation, San Francisco, CA, 2000.
- [Knierim, 2002] JJ Knierim. "Dynamic Interactions between Local Surface Cues, Distal Landmarks, and Intrinsic Circuitry in Hippocampal Place Cells." *Journal of Neuroscience*, vol. 22(14): 6254-6264, 2002.
- [Leonard and Durrant-Whyte, 1992] J.J. Leonard and H. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, Kluwer Academic Publishers, 1992.
- [McNaughton *et al.*, 1983] BL McNaughton, C.A. Barnes, et al. "The contributions of position, direction, and velocity to single unit activity in the hippocampus of freely-moving rats." *Exp. Brain Res.*, vol. 52: 41-49, 1983.
- [O'Keefe and Dostrovsky, 1971] John O'Keefe and J. Dostrovsky. "The hippocampus as a spatial map: preliminary evidence from unit activity in the freely moving rat." *Brain Res.*, vol. 34: 171-175, 1971.
- [Prasser and Wyeth, 2003] David Prasser and Gordon Wyeth. Probabilistic Visual Recognition of Artificial Landmarks for Simultaneous Localization and Mapping. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (to appear)*. Taipei, 2003.
- [Ranck, 1984] Jr. Ranck, J.B. "Head direction cells in the deep cell layer of dorsal presubiculum in freely moving rats." *Society Neuroscience Abs.*, vol. 10: 599, 1984.
- [Redish, 1999] David Redish. *Beyond the Cognitive Map*. Massachusetts, Massachusetts Institute of Technology, 1999.
- [Save *et al.*, 1998] Etienne Save, Arnaud Cressant, et al. "Spatial Firing of Hippocampal Place Cells in Blind Rats." *Journal of Neuroscience*, vol. 18(5): 1818-1826, 1998.
- [Skaggs and McNaughton, 1998] WE Skaggs and BL McNaughton. "Spatial Firing Properties of Hippocampal CA1 Populations in an Environment Containing Two Visually Identical Regions." *Journal of Neuroscience*, vol. 18: 8455-8466, 1998.
- [Taube *et al.*, 1990] J.S. Taube, R.U. Muller, et al. "Head direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis." *Journal of Neuroscience*, vol. 10: 420-435, 1990.
- [Thrun, 1997] Sebastian Thrun. "Learning Maps for Indoor Mobile Robot Navigation." *Artificial Intelligence*, 1997.
- [Thrun, 1998] Sebastian Thrun. "Bayesian Landmark Learning for Mobile Robot Localization." *Machine Learning*, vol. 33: 41-76, 1998.
- [Thrun, 2000] Sebastian Thrun. "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva." *Journal of Robotics Research*, 2000.