DEPARTMENT OF COMPUTER SCIENCE

Analysis of neural activity during spatial decision making

Thomas Jahans-Price

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

September 2010 CSMSC-10

Declaration

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Thomas Jahans-Price, September 2010

Abstract

The project has two main components, the first is the introduction of a neuroscience software toolkit MQL (Maze Query Language). MQL is a MATLAB written tool that allows the querying of positional data recorded from rats undertaking spatial tasks set in mazes. The user defined queries created allow specific trajectories to be picked out from a recording session and specific timestamps to be selected from the trajectories. The querying system of MQL is designed to aid the speed of analysis and to allow the analysis of complex trajectories. MQL also corrects for lost signal during recording by carrying out interpolation. Software fulfilling these functions could not be found and therefore MQL fills this gap in software for neuroscientists. MQL is for use on data from any type of maze and is downloadable with documentation from
http://www.cs.bristol.ac.uk/research/machinelearning/mql

The second component of this project was the use of MQL to analyse data from a specific spatial decision making task carried out by rats in a maze. The use of MQL on this data allowed analysis not otherwise possible. Activity at the choice turn in the maze was analysed in order to look at neural activity and investigate why errors occur in this task. Findings indicated that errors in the maze were not due to exploration by the rat, with no significant change in prefrontal activity. These results also showed rats took significantly longer to turn during error trials. Activity during all the turns in the maze was analysed using MQL queries in order to validate a prediction of a computational model of the task. The prediction that there should be neurons selective for turn directions was able to be verified as results showed the discovered of these turn selective neurons in the medial prefrontal cortex. Using MQL the activity of these neurons was analysed just before the choice turn of the maze and showed that turn selective neurons increase their activity before making the turn they select for. Furthermore when activity during correct and error trials were analysed in this area of the maze, turn selective neurons were shown to be significantly less selective during error trials than during correct trials.

- This project produced a new software toolkit MQL, which interpolated position data to correct for signal loss and uses user defined queries to return selected trajectories from position data (See MQL section).
- The MQL software was used in analysis of data from the Jones and Wilson (2005) T-maze task to answer research questions leading to the following discoveries:
    - Analysis of timing during choice turns showed rat significantly slower during errors
    - Analysis of activity during choice turns showed similar levels activity casting doubt on a theory of exploration to explain errors.
    - Analysis of activity during turns led to the discovery of turn selective neurons in the prefrontal cortex of the brain, something we are unaware of being reported previously.
    - Analysis of turn selective neurons leading to choice turn showed an increase in activity.
    - Turn selective neurons were less selective on error trials during the choice turn.

Acknowledgements

I would like to express my thanks to Rafal Bogacz, for his invaluable advice, guidance and encouragement throughout this project.

Thanks also to Matt Jones and Nadine Becker for their help, feedback, for providing data and piloting MQL.

Thanks to Angela Onslow for giving feedback on the MQL software, MQL website and for help proof reading.

I would like to greatly thank my family for their constant support and encouragement.

Contents

1 – Introduction

This section will explain the objectives of the project and then will go on to explain the structure of this document.

1.1 - Aims and Objectives

This project has two main aims firstly to develop a software tool that can query positional data collected from the recording of spatial maze based tasks carried out by rats. Such a tool would be developed in order to aid analysis of these spatial tasks by providing a method of selecting desired trajectories. Secondly to utilise such software in order to perform analyses on a data from a specific maze based decision making task in order to answer research questions.

1.1.1 - Software Objectives

- Implementation of system to interpolate position data to correct for signal loss
- Querying system based on user defined lines, returning trajectories that intersect them
- Precision and flexibility in the querying system, many different queries possible, specific trajectories able to be returned
- Graphical User Interface used to create queries
- Simple system for exporting data
- Software able to be used for a wide variety of mazes

1.1.2 – Analysis Objectives

- To address the following research questions:
    - Why do errors occur in the Jones and Wilson (2005) T-maze task? Analysis of timing and activity during the choice turn using the developed software tool to compare correct and error trajectories.
    - Are turn selective neurons found? Analysis of turns in maze to compare activity during different turn directions.
    - If turn selective neurons are found, does their activity increase before a decision? Analysis of maze trajectories leading to choice turn.
    - If turn neurons are found, what role do they play in errors. Comparison of activity of turn selective neurons during correct and error trials.
- To address other research questions that arise as a result of analyses.

1.2 - Structure

This report consists of 5 main sections, firstly time is spend explaining the background of decision making through early experiments to recent computational models. The Jones and Wilson T-maze task is explained along with the reasons for the research questions mentioned in the analysis objectives. Requirements for a software querying tool are also explained in the background section. The Methods section explains the development of the software tool and its use in creating queries to form the basis of analysis. Next the results section explains in detail the research questions answered along with describing the results and whether these answer the initial questions. The 4th section is a discussion of the results found and what they may mean to in relation to ideas discussed in the background, there is also an evaluation of this project to determine if it has met its objectives and if it has been a success. Finally future work is discussed in terms of both further extensions to the MQL querying tool and further analysis that could be carried out to answer questions raised by the findings of this project.

2 – Background

2.1 – Neural Basis of Decision Making

How from simple neurons do complexity and cognition like the ability to make decisions arise? It is a question that has been studied and modelled extensively. This section will examine the neurophysiology of decision making and experimental data in order to describe some fundamental assumptions required for the modelling of decision making.

2.1.1 – Random Dot Test

Due to the complexity of the brain, we are unable to model general decision making. Instead behaviour and neuronal activity are studied in a situation where a choice between two alternatives is presented and a decision made. This is a very basic form of many real scenarios but much research has been done with this type of experiment. The most prevalent decision making experiment is the Random Dot (RD) test (Britten et al 1993). It consists of showing a subject (human or animal) a number of dots flashing, a subset of which move coherently from left to right or right to left, the rest moving randomly (Shadlen & Gold 2004). The subject decides whether the coherent dots moved to the left or right with either an eye movement in the choice direction (saccade) or by the press of a button.

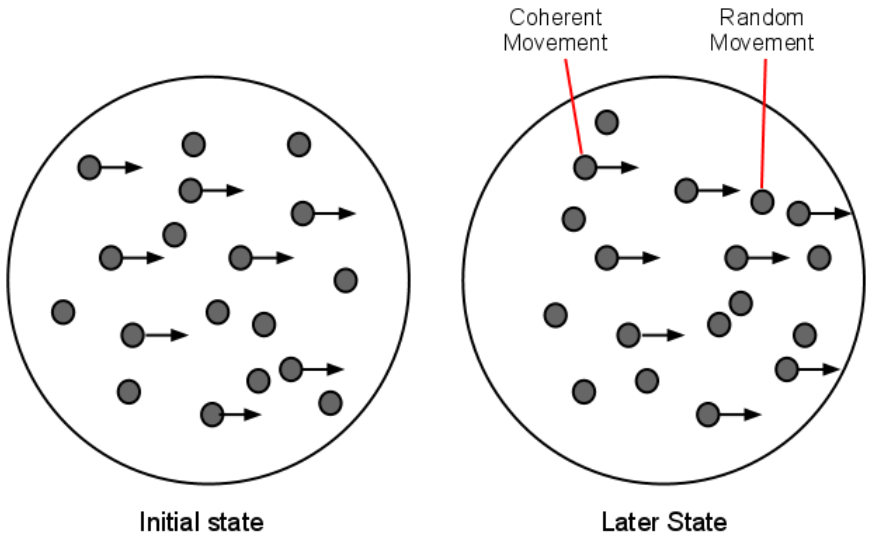

Figure 2.1: Random Dot test (Shadlen & Gold 2004) showing two states in the RD test, the second of which shows coherent movement from some dots and random movement from the rest.

The flashing dots provide the test subject with noisy data to process, this causes a relatively slow speed response from neurons, enabling the decision process to be studied (Shadlen & Gold 2004). Another strength of the RD experiment is that it allows for variation of both viewing time and difficulty (the higher the percentage of dots moving coherently, the easier the task becomes) which enables it to ascertain a variety of interesting data. When the viewing time is left to the control of the subject, effectively giving them a free choice, reaction times can be calculated. The varying the difficulty of the task and the viewing time shows the speed accuracy trade off that we see in nature.

2.1.2 – Accumulation of Evidence

Accumulation of Evidence is a fundamental assumption in the understanding and modelling of decision making. It is believed that groups of neurons represent evidence for alternatives in a choice, accumulation of evidence is the concept that the activity in these neurons (measured by the rate that they produce spikes, referred to as the firing rate) accumulates during the decision making process. Accumulation of evidence is proposed by Stone (1960) and advocated by Vickers (1970). It is shown experimentally that firing rates do accumulate in the lateral intraparietal (LIP) by Schall (2001) and by Shadlen and Newsome (2001). This accumulation provides an explanation for the speed accuracy trade off, as data is presented in the RD task firing of neurons increases, accumulating evidence for each alternative over time. The firing rate is affected by the difficult of the task, the easier it is, the quicker the firing rate increases and evidence is accumulated. This demonstrates the speed accuracy trade off as with longer periods of choice time more difficult tasks can eventually be solved. This is possible as over time the evidence can accumulate to an equivalent level, albeit at a slower rate. Conversely for a simple version of the test, high accuracy is achieved within a smaller timeframe.

2.1.3 – Threshold

Roitman and Shadlen (2002) show that varying the difficultly of the task does not change the neural activity level required to execute a decision and that this implies a common threshold of some kind is reached.

2.1.4 – Noise

Models have been proposed that work on these assumptions, however in order for there to be a percentage of erroneous choices the evidence provided for each alternative has to be noisy (Ratcliff 2001). This was also shown experimentally (Britten et al 1993) where we see noise in the firing in MT, a visual input region of the brain responsible for direction detection. Noise in the evidence explains error, it also allows the prediction of accuracy and of the speed accuracy trade off. This also explains the need for accumulation as without noise in the firing rates from MT, if an alternative with a higher firing rate for its evidence is known, the correct choice is also known. The addition of noisy evidence to our understanding of decision makes this problem non trivial and requires that the evidence for all alternatives be sampled over time in order to try and ascertain the correct choice (Bogacz et al 2007).

2.1.5 – Summary

We can represent decision making in the RD test and similar choice experiments with the following 3 properties: evidence for alternatives is accumulated over time, that the evidence is noisy and that decisions are made when a threshold is reached. These three assumptions are important foundations of decision making that computational models use.

2.2 – Computational Models of Decision Making

Models attempt to recreate and potentially explain experimental data. As a consequence they often propose changes to our physiological understanding of the brain. This can be seen in Vickers(1970) where no experiment was carried out, simply the evaluation of different mathematical models, the conclusion of this paper was the validity of the aforementioned accumulation concept. This became a fundamental concept in decision making, which was later supported by experimental results (Shadlen 2004). Through creating computational models, we aim not only to understand the current state of the art, but to further it.

This section will explain some models for decision making starting with experimental data models must explain, mentioning a number of mathematical and computational models before focusing on the Leaky Competing Accumulator (Usher and McClelland 2001) which provides a framework in which models for the rat T-maze spatial task (Jones and Wilson 2005) have been developed in.

2.2.1 – Error Rate and Reaction Time

Models of neurophysiology need to explain experimental data, the measurements in experiments most pertinent to decision making e.g. random dot test, are reaction times and error rate. Reaction times and error rate are linked, we see this in the free choice version of the experiment where the choice task is increased in difficulty, reaction times increase as does the error rate. The converse is also true, as when the difficulty is reduced we see a drop in both the error rate and reaction time. This is intuitive as easier tasks should be completed quickly and accurately, more difficult tasks take longer and are prone to mistakes.

- **Lower** Difficulty
    - **Lower** Reaction Time
    - **Lower** Error Rate
- **Higher** Difficultly
    - **Higher** Reaction Time
    - **Higher** Error Rate

2.2.2 – Speed Accuracy Trade Off

In the forced choice experiments we see the formalisation of the speed accuracy trade off. It can be seen by maintaining the difficulty level of the test and varying the amount of viewing time the subject is allowed. This gives results that show with more reaction time allowed the error rate is lowered and with less reaction time error rate is higher. This follows what one would expect as greater time allowed to make a decision should yield greater accuracy and rushing a decision without consideration would lead to error.

Fixed difficulty

- **Lower** Reaction Time - **Higher** Error Rate
- **Higher** Reaction Time - **Lower** Error Rate

2.2.3 – Diffusion Model

A simple mathematical model of choice is the diffusion model (Ratcliff 1978) which models the decision process as the integration of the difference in supporting evidence for 2 alternatives. This is represented as a point, shown in Figure 2.2 as Y, which fluctuates in a manner similar to a particle floating on water with movement up and down on a current over time. Positive and negative thresholds of difference between the 2 alternatives are set and once the level of evidence for either alternative becomes great enough the threshold is breached and the decision made. The 'current' can be thought of as the way the level of difference in evidence change as time progresses and that it will tend toward the correct decision. This mean that the particle is floating, fluctuating due to noise in the evidence levels but will be 'pulled by the current' toward the correct alternative, however due to noise in the evidence it is possible that the particle can be pushed off course giving an incorrect decision and therefore an error rate.

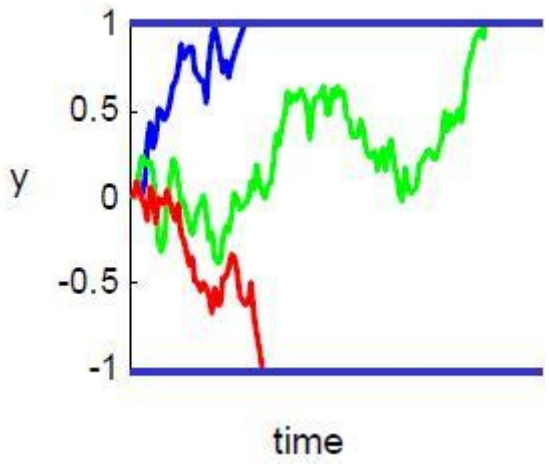

Figure 2.2: Diffusion model from Bogacz et al (2006)

The diffusion model is important as it is implements the optimal test for decision tasks, the sequential probability ratio test (SPRT) (Wald 1947). SPRT is optimal as for the decision task it maintains a fixed error rate and produces the lowest reaction time, this is a benchmark of performance that other models aim to match.

2.2.4 – Race Model

The race model which was proposed by stone (1960) and advocated by Vickers (1970) is a mathematical model for decision making based on the principal of accumulation. It sets a threshold and whichever alternative accumulates the most supporting evidence and breaches the threshold first, wins the 'race' and the decision is made in favour of that alternative.

A benefit of this model is that it is multi-dimensional with support for more than 2 alternatives as evidence for many alternatives can be accumulated independently. This is a benefit over the diffusion model, which represents the difference in accumulation with a single dimension. An issue with both this model and the diffusion model is that the threshold must be altered depending on the difficulty of the task in order for the model to act optimally. Setting the threshold higher allows more time to sample the input giving higher accuracy and setting the threshold lower makes a decision based on less information, giving lower accuracy but faster decision. Therefore depending on the difficulty, to maintain an error rate and minimise reaction time the threshold must be lowered for an

easier task to minimise reaction time and raised for a more difficult task in order to maintain accuracy.

2.2.5 – Biologically Inspired Connectionist Models

Biologically inspired models describe decision making in the following way: They represent the mean input from sensory neurons for alternative *i* as I_i and the accumulation of this supporting evidence in areas like LIP by integrator neurons represented with y_i. A threshold level of evidence is set and once the level of activity in an integrator y_i reaches the threshold the decision is made in favour of alternative *i*. The longer we sample from the average activity of input neurons the more the effect of noise is reduced and the greater degree of certainty we have about the correct choice. This effectively describes a biological version of the race model, which still has the problem that the difference between alternatives may be obvious at an early stage requiring the modification of the threshold dependant on the difficulty. To overcome this problem, biologically inspired models introduce further biological concepts.

2.2.6 – Inhibition and Excitation

Connections between neurons in the brain can be excitatory or inhibitory, meaning neurons can either increase or reduce firing of other neurons. Up until now the only neurons mentioned have been excitatory neurons that represent evidence and excite integrator neurons. These neurons are shown in diagrams as a line with an arrow to show an excitatory relationship between two groups of neurons. The biologically inspired computational models also use inhibition, where the increase in firing in one group of neurons will reduce firing in another group, shown on the diagram as a line with a dot.

2.2.7 – Leak

The firing rate of neurons is dependent upon the level of current that is input into either a single neuron or a population. A key feature of neurons is the leak of information they hold over time, this is due to an intrinsic decay in the input current (Usher and McClelland 2001). This leak of current in neurons and therefore drop in firing rate means that without further neurons would eventually lose all information. Leak is build into computational models as a rate of decay at which the level of accumulated evidence in integrator neurons falls over time.

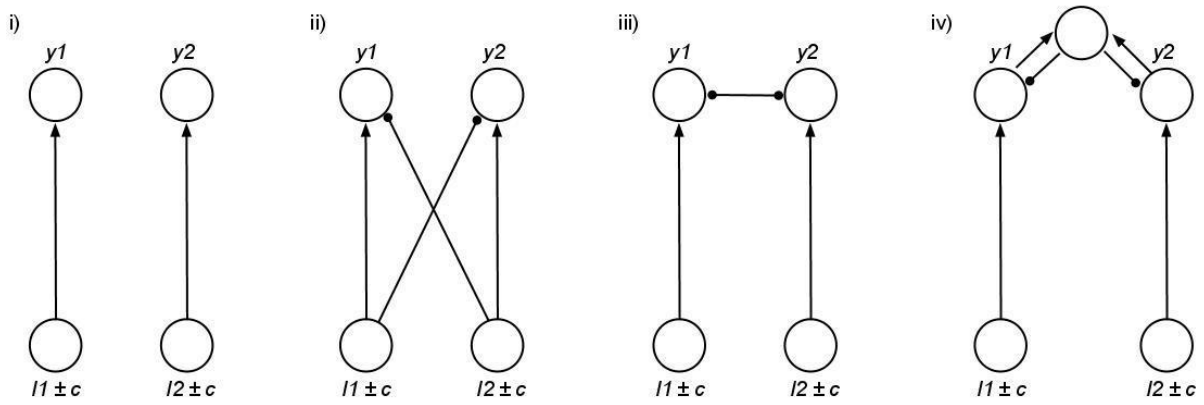

Figure 2.3 shows a number of models of decision making and their structures:

I. The race model (Vickers 1970)
II. Feed forward inhibition model (Mazurek et al. 2003)
III. LCA model with lateral inhibition (Usher and McClelland 2001)
IV. Pool inhibition model (Wang 2002)

Circles represent groups of neurons, lines with dots represent inhibitory connections, lines with arrows represent excitatory connections.

2.2.8 – The Pool Inhibition Model

This computational model, shown in the above Figure as part iv) features two integrators trying to reach a threshold, this is in keeping with the other models. However these integrators are not inhibited directly by the other alternative, as in the LCA model and the feed forward model. The Pool model's distinguishing feature is a separate integrator that both alternatives activate through excitatory connections, this group of integrator neurons simultaneously inhibits the y_i integrators.

2.2.9 – Leaky Competing Accumulator Model (LCA)

The LCA is a biologically inspired model, as its name suggests its primary features are leaking accumulators that sum supporting evidence and compete with each other to determine the alternative to be chosen. For *i* number of alternative the LCA model uses y_i competing integrators similarly to the race model, the integrators have a decay rate (*k*) which states how fast the summated evidence in them will leak out. The integrators use lateral inhibition to inhibit each other. This is in contrast with other models of inhibition such as feed forward in which the integrators are inhibited by the input neurons for the opposite alternative and pooling where a single separate population of neurons is simultaneously excited by, and in turn inhibits all integrators y_i. Usher and McClelland (2001) used local lateral inhibition and they state neurophysiological evidence supporting the idea that lateral inhibition is more likely to occur in a single area of the brain and excitation more likely to occur between brain areas.

The effect of adding leak and inhibition to the computational model is that inhibition prevents evidence levels becoming high simultaneously for both alternatives. For example in a difficult free choice where evidence for both alternatives (in a 2 alternative task) is at a similar level the integrators inhibit each other and are both stopped from reaching the threshold, at the same time they are both leaking evidence and continuing to accumulate evidence sampled from the input

neuron population. In this difficult task inhibition and leak results in an increased reaction time order to improve accuracy. In an easy task, where evidence for one alternative is much greater than the other, the effect of inhibition is to greatly inhibit the weaker alternative allowing the stronger to quickly reach the threshold. This helps the model replicate the reaction time data we see experimentally.

In an average difficulty task we would see a combination of these two effects, an initial move toward similar levels of evidence before one alternative growing and increasingly inhibiting the other allowing the model to move toward the correct choice.

2.2.9.1 - Equation

To simplify the equation, the assumption can be made that the integrators y_1 and y_2 start at 0 (Bogacz et al 2006)

$y_1(0) = y_2(0) = 0$

Noise is required to be added to the evidence sampled by the integrators, this noise is generated using wiener process, noise is shown as fluctuations dW_i of amplitude $\pm c$. Decay and inhibition are represented by k and w respectively.

Using leaky competing accumulator model with these parameters, levels of evidence for alternatives $i = 1$ and $i = 2$ are represented by the following equations (Usher and McClelland 2001):

$$\begin{cases} dy_1 = (-ky_1 - wy_2 + I_1)\,dt + c_1 dW_1 \\ dy_2 = (-ky_2 - wy_1 + I_2)\,dt + c_2 dW_2 \end{cases}$$

This 2 choice model can be extended to include more than 2 alternatives, the equation to describe the level of evidence for alternative i from Usher and McClelland (2001) is:

$$dy_i = \left(-ky_i - w \sum_{\substack{j=1 \\ j \neq i}}^{N} y_j + I_i \right) dt + c_i dW_i, y_i(0) = 0$$

The addition of noise creates the probability of incorrect choices and allows over many trials an average error rate to be calculated for the LCA model. In the forced choice variant of decision tests the LCA model can compute average response times.

2.2.9.2 – Effect of parameters

The inhibition and inability for y_1 and y_2 to simultaneously be high, leads to a plane of attraction, if we imagine the LCA model as a two dimensional state space with y_1 and y_2 as the axises and the state of the decision process as a point with its position dependant on the levels of evidence in y_1 and y_2. In this state space, the plane of attraction is a diagonal line, starting and finishing where one alternative is at the threshold and the other at 0, the midpoint occurs where $y_1 = y_2$. How the parameters decay (k) and inhibition (w) are set affects attraction within this state space, creating

areas of increased attraction or repulsion. These areas of modified attraction act similarly to the diffusion model as a current to pull the levels of evidence in y_1 and y_2 in various directions. Depending on which is greater, decay or inhibition the differing currents in state space that are created have interesting effects (Bogacz et al 2007). These effects are described by both Usher and McClellend(2001) and Busemeyer and Townsend (1993) and shown below in Figure 2.4:

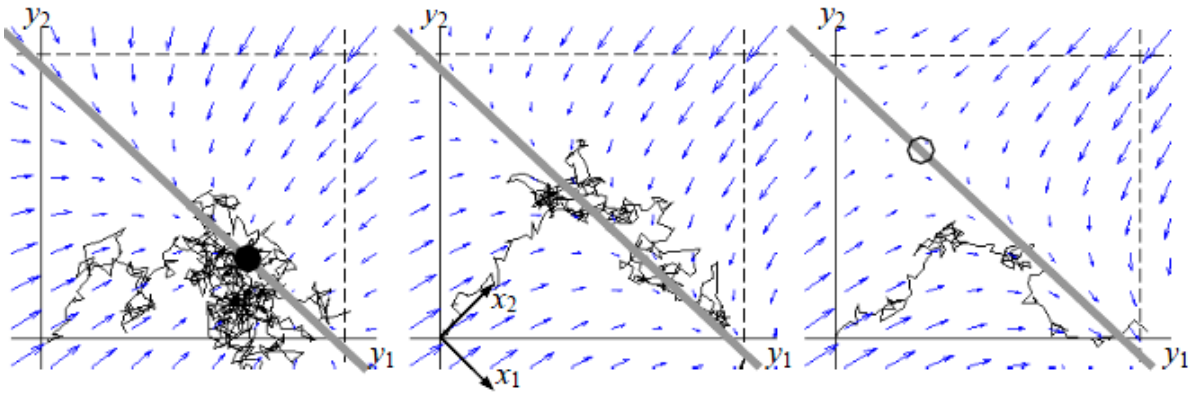

Figure 2.4 from Bogacz et al (2007): shows the different state spaces for different parameter combinations. Decay > Inhibition (left), (b) Decay = Inhibition (centre), Decay < Inhibition (right)

- Primacy (Figure 2.4 right) - When inhibition is greater than decay a point of repulsion is created on the plane of attraction. This has the effect of primacy whereby evidence presented early in decision making has more impact on the outcome. This occurs as once the state of the decision process is toward one threshold past the point of repulsion, it is less likely to be able to go back.

- Recency (Figure 2.4 left) - When decay is greater than inhibition an area of attraction is created, this has the opposite effect to primacy, in that evidence presented later in the decision process become more responsible for the outcome. This effect is called recency and comes about because the attracting point makes it likely that evidence seen later has the ability to push the state from one alternative to another through the pull of the attraction point, as initial evidence will decay away faster.

- Balanced (Figure 2.4 centre) - When inhibition and decay are equal the a pull toward the plane of attraction is present but no points of high attraction or repulsion, this is referred to as the 'balanced' state of the model whereby evidence presenting throughout the decision process has equal weight.

These effects seen in the changing of parameters, were shown to occur in humans by User and McClelland (2001), they observed all 3 states present during their experiments. When in the balanced state LCA approximates the optimality of the diffusion model with respect to minimising reaction time when error rate is fixed. When decay and inhibition are set to 0, the LCA is equivalent to the race model (Vickers 1970) as there is no longer any leak or inhibition, simply the accumulation of evidence.

2.2.10 – Summary

This section described a number of computational models of decision making, how the introduction of biologically inspired features such as leak and accumulation are important in overcoming problems of previous models. The LCA model which was focused on in this section is used as a

foundation for a model of the T-maze task, a spatial decision making task both of which are described in the following section.

2.3 – T-maze Spatial Decision Task

This section will examine a specific decision making experiment and the current computational model that describes it. Before mentioning the task, its purpose and findings, it is important to explain a key neuroscience concept which underlies this research. Therefore this section will begin by giving an overview on the concept of theta rhythms, and how some people believe they could be a crucial idea in our understanding of how brains function. This will lead to an explanation of the Jones and Wilson (2005) T-maze task followed by a description of the Gorochowski (2009) computational model. The last part of this section will describe predictions of the model, analysis of data from the T-maze task and the research questions they raise.

2.3.1 – Theta Rhythms

Some neuroscientists believe that oscillations are responsible for the coordination of neural firing in the hippocampus by acting as a clocking mechanism (Mehta et al 2002). This would allow the coordination of functional interactions in this particular area of the brain, it is also thought by some that oscillations could be used to coordinate functions between separate brain areas that are linked either functionally or anatomically(Siapas et al 2005, Jones and Wilson 2005). Two areas of the brain that are linked functionally are the hippocampus and the medial prefrontal cortex (mPFC). The hippocampus is associated with memory and spatial tasks, the mPFC is thought to be involved in decision making and guiding behaviour relating to rewards (Öngür and Price 2000).

A number of types of oscillations exist, theta rhythms are oscillations of neural firing in the 4-12 hz range and are linked with the aiding and facilitation of tasks such as spatial exploration in rodents (Vanderwolf 1969). They are observed in the rat hippocampus (Buzsaki 2002) and their existence comes from the relationships between interconnected inhibitory and excitatory neurons in the hippocampus which can give rise to oscillatory firing. The firing of some excitatory spatial neurons in the hippocampus referred to as ‘place cells’ become locked to the local theta cycle, the effects of this is the actions potentials in place cells are raised with particular phases of the cycles, making them more likely to fire in time (Jones and Wilson 2005).

There is some evidence to suggest that the other areas of the brain have neuronal firing locked to the hippocampal theta rhythm. Several studies look at the medial prefrontal cortex, it is suggested by Siapas et al (2005) and Jones and Wilson (2005) that the mPFC is phase locked to the hippocampus. Jones and Wilson (2005) suggest that this could be related to the ability to form long term memories and could aid in transferring information from hippocampus to neocortex. The phase locking between these two areas of the brain was examined using a decision making spatial task with rats, carrying out neural recordings to find out if theta rhythms are the method by which these different parts of the brain coordinate their activity.

2.3.2 – The T-maze Task

The experiment conducted by Jones and Wilson (2005) was designed to investigate the possibility of theta rhythms playing a role in coordinating brain activity during decision making in rats undertaking a spatial task. The experiment consists of a T-maze task made up from a central arm with 4 other arms leading to 4 reward points.

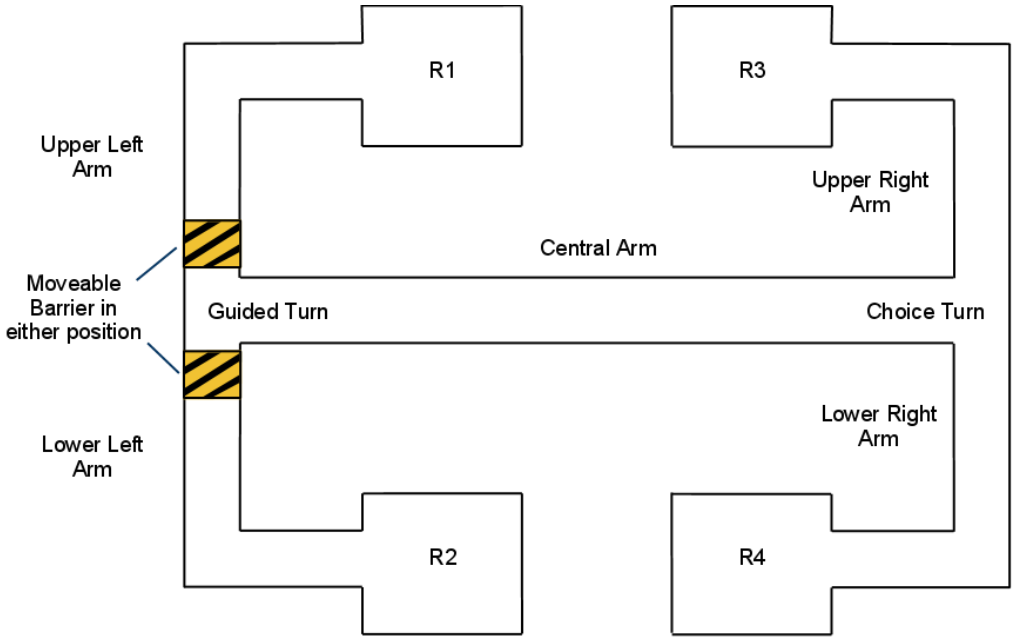

Figure 2.5 – Structure of Jones and Wilson (2005) T-maze, R1-4 are reward points, barrier is only ever in one of the positions shown.

The maze is designed in order to provide two general types of trial. A 'choice trial' provides the rat with a choice between two alternatives, to turn either left or right. A 'guided trial' is a control in order to obtain comparison data, in this case one choice is blocked using a moveable barrier and the rat must take the only available turn.

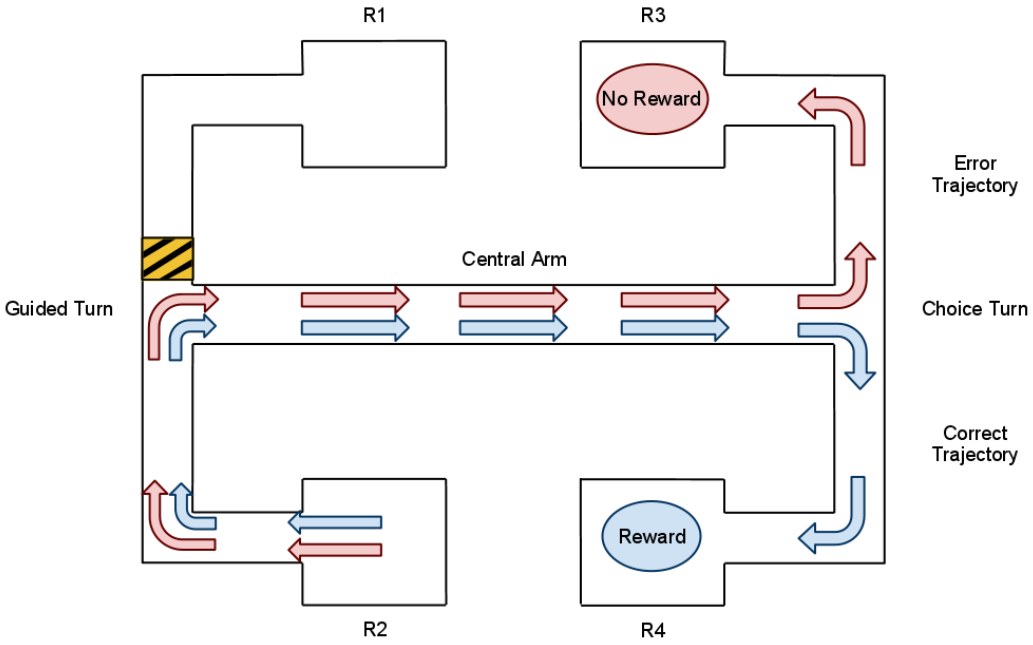

Figure 2.6: Visualisation of choice trial starting at the R2 reward point. Blue arrows demonstrate the trajectory of the 'correct trial' consisting of two turns in the same direction resulting in a reward at reward point R4. Red arrows show the trajectory of the 'error trial' of two different turns resulting in no reward at reward point R3.

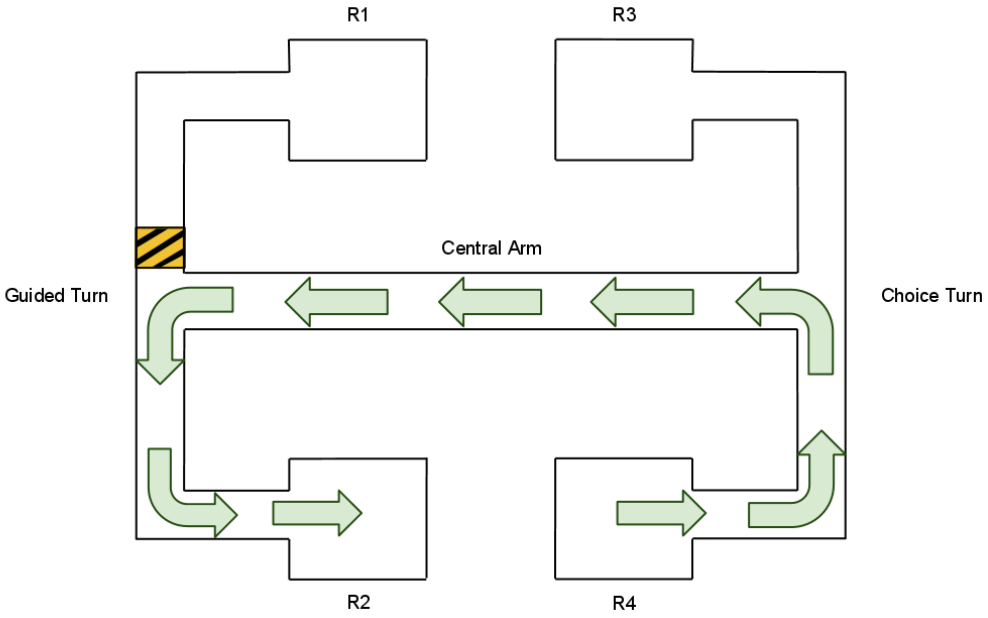

Figure 2.7: Guided trial starting at R4. The barrier placed in the upper left arm, shows that a trajectory from right to left starting at reward point R4 would result in a guided left turn.

In total the T-maze experiment produces six trials, four 'choice trials' and two 'guided trials'.

Choice trials:

- Correct trials starting at lower left reward point R2 finishing at R4
- Error trials starting at lower left reward point R2 finishing at R3
- Correct trials starting at upper left reward point R1 finishing at R3
- Error trials starting at upper left reward point R1 finishing at R4

Guided trials:

- Guided trials starting at lower right reward point R4 finishing at R2
- Guided trials starting at upper right reward point R3 finishing at R1

Choice trials always start on the left side of the maze and finish at the right, the guided trials start from the right and finish to the left. On the choice trial the rat starts at either of the left side reward points, runs along the connected arm it must then make an initial guided turn enforced by the barrier into the central arm, run along the central arm from left to right toward the choice turn where it must make the same turn again in order to get a reward. For example: if the rat starts at the lower left reward point R2 it would make a guided turn right into the central arm, on exit of the central arm the rat is presented with the choice turn, the correct alternative being a right turn (shown in Figure 2.6). If the rat makes the turn choice right it is rewarded at reward point R4 with sugar solution, incorrect choices resulting in the rat running to R3 are not reinforced. The same is true for a trial starting at R1 which is rewarded at R3 but not reinforced at R4. The data gained from this task showed significant coherence in the theta rhythm range 4-12hz between the CA1 and mPFC regions of the rodents brain. The coherence was significantly higher during correct trials than both error trials and guided trials (Jones and Wilson 2005). This supports the idea of communication and coordination between areas of the brain.

2.3.3 – Existing Computational Model

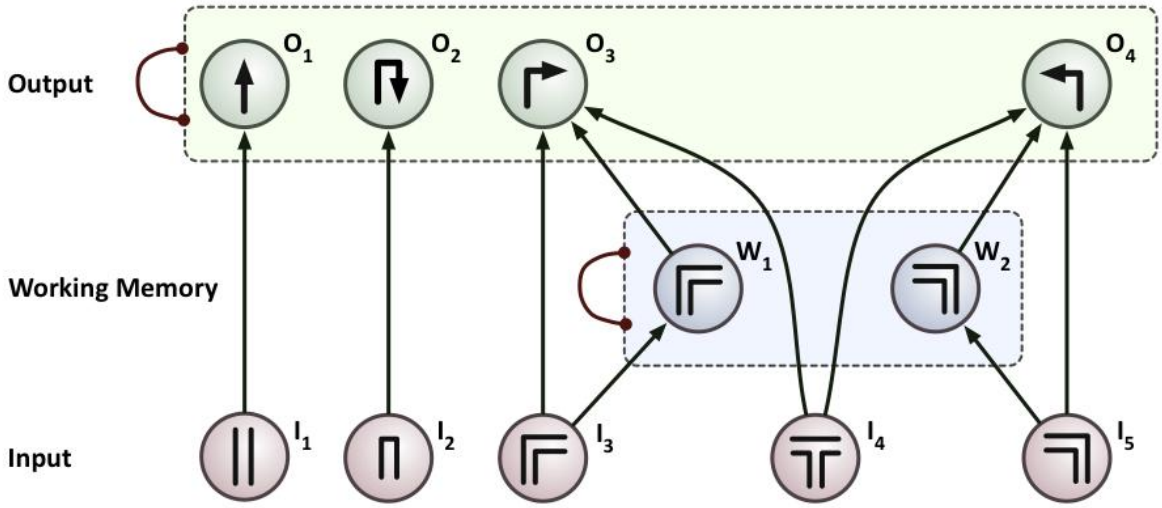

Figure 2.8 – Computation model of T-maze task, taken from Gorochowski (2009), the output layer shows a double inhibitory connection at the group level, this represents the fact that the integrators laterally inhibit each other, equivalent to lateral inhibitory connections between all of the outputs. The same is applied to the working memory layer.

Output and working memory integrators Gorochowski (2009):

$$
\begin{aligned}
O_1 &= (I_1 - kO_1 - w\,(O_2 + O_3 + O_4))\,dt + cdN \\
O_2 &= (I_2 - kO_2 - w\,(O_1 + O_3 + O_4))\,dt + cdN \\
O_3 &= (I_3 + I_4 + W_1 - kO_3 - w\,(O_1 + O_2 + O_4))\,dt + cdN \\
O_4 &= (I_4 + I_5 + W_2 - kO_4 - w\,(O_1 + O_2 + O_3))\,dt + cdN \\
W_1 &= (I_3 - kW_1 - wW_2)\,dt + cdN \\
W_2 &= (I_5 - kW_2 - wW_1)\,dt + cdN
\end{aligned}
$$

Figure 2.8 shows the initial computational model of the Jones and Wilson (2005) T-maze task developed by Gorochowski (2009). It is a connectionist model based on LCA model. It shows inputs for all possible stimuli the maze provides, a straight path, dead end, forced turn and choice turn. The inputs are linked to outputs for all the potential alternatives available to the rat, to run straight, turn around, turn right and turn left. It includes an extra set of integrators in the model which represent working memory, there are lateral inhibitory connections between the integrators in both working memory and output as in the LCA. Working memory is required for the model as on a choice run, the rats are required to keep the last turn in mind during their run down the central arm until they make the choice turn. The relevant working memory integrators accumulate evidence when a turn is made, then when the choice turn stimulus is present which ever turn has the highest evidence in working memory will affect the decision.

A wiener process was used to add noise to the model, in order to ensure that the noise scaled correctly over time, the distribution of noise was based on the time step using the following equation where dN is noise output and dt is timestep (Gorochowski 2009):

$$
dN \sim Normal\left(0, \sqrt{dt}\right)
$$

The model described above did not account for theta oscillations which are found in the data of the experiment. Modifications were made, based on the fact that oscillations result from inhibitory and excitatory connections and that this may be through a group of neurons shared by integrator neurons (Gorochowski 2009). Therefore the addition of an extra group of integrators was made to both working memory and output integrators. These integrators would in turn inhibit the neuronal populations that excite them in a similar manner to the Pool model (See The Pool Inhibition Model). This alone did not create the required oscillations, but by adding a delay in the connections to and from the new integrators, oscillations arise with a delay greater than 0 (Gorochowski 2009).

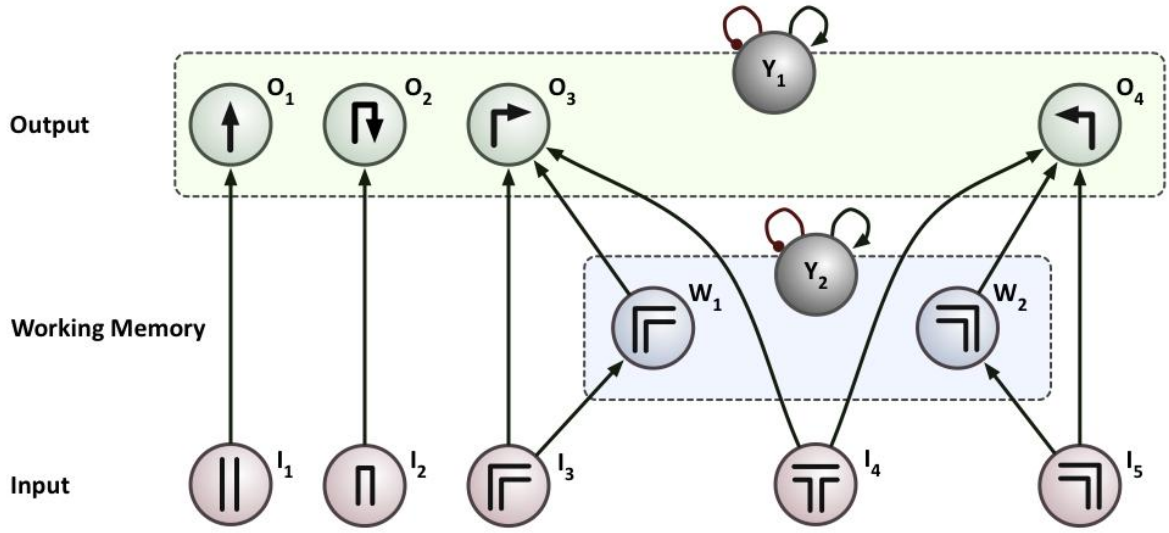

Figure 2.9 – Modified computation model of T-maze task which includes additional pool integrators, taken from Gorochowski (2009)

This new model represents the integrators with the following amended equations (Gorochowski 2009):

$$\dot{O}_1(t) = [I_1(t) - kO_1(t) - wY_1(t_d)]\,dt + cdN$$
$$\dot{O}_2(t) = [I_2(t) - kO_2(t) - wY_2(t_d)]\,dt + cdN$$
$$\dot{O}_3(t) = [I_3(t) + W_1(t) + I_4(t) + kO_3(t) - wY_1(_d)]\,dt + cdN$$
$$\dot{O}_4(t) = [I_4(t) + W_2(t) + I_5(t) + kO_4(t) - wY_1(_d)]\,dt + cdN$$
$$\dot{W}_1(t) = [I_3(t) - kW_1(t) - wY_2(t_d)]\,dt + cdN$$
$$\dot{W}_2(t) = [I_5(t) - kW_2(t) - wY_2(t_d)]\,dt + cdN$$
$$\dot{Y}_1(t) = [a_1[O_1(t_d) + O_2(t_d) + O_3(t_d) + O_4(t_d)] - kY_1(t)]\,dt + cdN$$
$$\dot{Y}_2(t) = [a_2[W_1(t_d) + W_2(t_d) - kY_2(t)]\,dt + cdN$$

2.3.4 – Predictions of Model and Further Analysis of T-maze Data

The Gorochowski model of the T-maze task makes a number of predictions about neural activity and behaviour of the rat. Work has been carried out to look for evidence in the data that supports this model and its implications (Bogacz 2010). This work was based on data from the Jones and Wilson (2005) experiment and it verified a number of the Gorochowski model predictions, including showing that neurons in working memory do appear to be selective for the direction of the initial turn made into the central arm. This was shown by neurons in the prefrontal cortex firing significantly differently depending on the direction of the initial turn for choice trials. The analysis by Bogacz (2010) also looked at the cause of errors during incorrect trials and strategies rats may use to be successful in the task. The length of time rats took to run along the central arm of the maze was analysed in order to compare correct and error trials (we will refer to this type of analysis as timing). This analysis showed that rats spend significantly longer in the central arm on error trials when compared with correct trials. Also analysed during choice trials in the central arm was the neural activity in prefrontal and hippocampal neurons, by counting spikes average rates of firing were calculated, this showed no significant difference in neural activity between correct and error trials for either brain region. It is possible that what we refer to as error trials during the T-maze task, rather than being an error made at the choice turn is in fact an exploratory move. The rats, despite

having learnt the T-maze task may still explore the maze occasionally. Daw et al (2006) showed in humans that during decisions classed as exploratory, the prefrontal cortex is more active and suggest that the prefrontal cortex controls exploritory behaviour. In order to answer this question of whether error turns are exploratory analysis of activity in the pfc during the choice turn is required.

Another question raise by the model is that of neurons selective for turn direction, a prediction of the model is that these neurons should be found (shown in Figures 2.8 and 2.9 in the output layer receiving input from working memory neurons). Neurons selective for turns and trajectories have been found in the parietal cortex (Whitlock et al 2008) an area of the brain thought to be responsible for spatial navigation. In order to verify the existence of these neurons in the T-maze task, to investigate their involvement in decision making and the question of the cause of errors, the development of a tool to query neural data recordings is proposed(Bogacz 2010). It would the aid in the analysis of the T-maze data and provide the ability to include and exclude parts of the maze from the trials recorded (Figure 2.10). It is stated that such a tool would be useful as it would help facilitate the investigation of turn neurons, as we would need to isolate activity at all turns and corners in the maze.

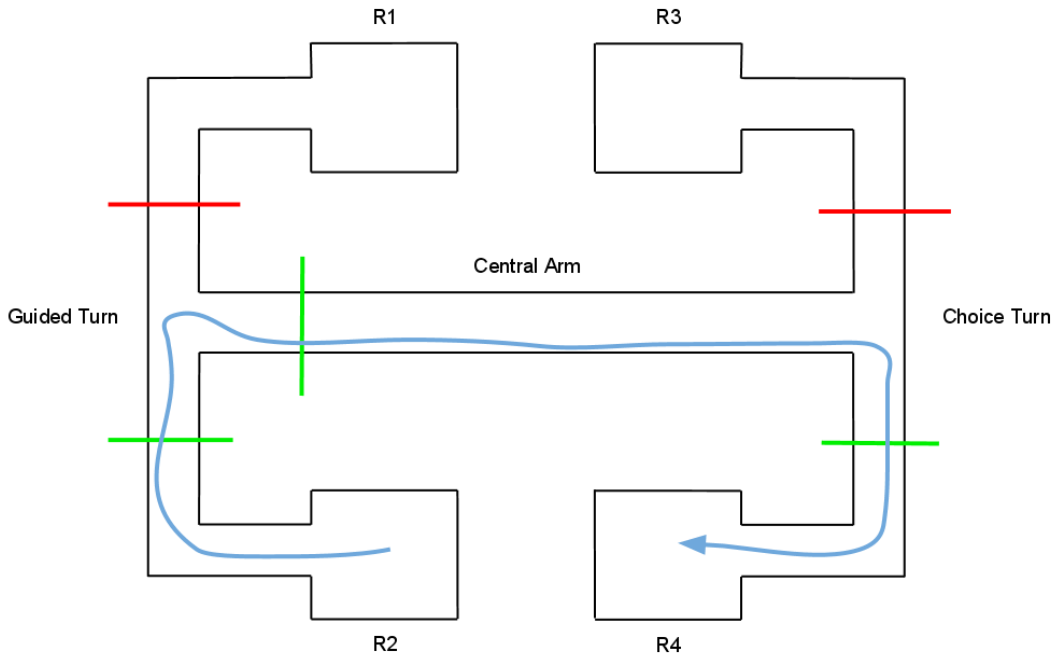

Figure 2.10: Potential query generated by a maze query tool, excluding trials involving both upper arms with red lines, including correct trials from R2 to R4 using green line. Example run shown in blue.

2.3.4 Summary

This section explained the details of the T-maze task explaining the classification of different trials. The existing computational model was explained, as were research questions posed by some of the model's predictions and by analysis of T-maze experiment data. In order to address these research questions a software tool was proposed to query the position data recorded from spatial tasks such as the T-maze. The following section examines available neuroscience software to find if this requirement currently is met.

2.4 – State of the Art - Existing Software

A number of software suites exist to manage data from neural recordings, this section will examine the different packages available, their functions and focus. It will attempt to ascertain if the need for a querying tool proposed by Bogacz(2010) can be met by currently existing systems.

2.4.1 – Neuralynx

Neuralynx is a company providing solutions for neuroscience research. The focus of their software and hardware components is the acquisition of electrophysiology data through the recording systems and they produce, indeed as mentioned, this is the experimental software and hardware used in the recording of the T-maze experiment. They also provide software utilities to aid in the recording, to convert between file formats and visualise data Neuralynx(2010). The latter of these tools allows the visualisation of data from recordings and has the ability to select a subset of the data through a graphical interface. The selection involves drawing a square over a subset of the data to view it alone, this method however doesn't give the flexibility of the proposed maze query tool as it only allows a single section of the maze to be selected in one go. The proposed maze querying tool by comparison would allow the drawing various different points of intersection and non-intersection to construct a potentially complex query and return a subset of trajectories. Without this level of precision it would not be possible to investigate the activity at the choice junction in the maze.

2.4.2 – Neuroshare

The aims of Neuroshare are stated in their mission statement webpage (Neuroshare 2010), the primary goal of Neuroshare was the creation of an open source single file format with to enable the user the ability to conduct their own analysis in many different environments. They list tools to conduct spike detection and playback of data but this is general analysis and again, not tailored to the problem the proposed maze querying tool aims to address.

2.4.3 – FIND

Find (Meier et al 2007) is a software suite that aims to bring the many different data formats together for identical analysis in one place, this would allow the data from many different experiments to be compared and analysed together. It uses the Neuroshare format and provides tools to analyse many types of neural activity data including imaging data, spikes and continuous recording (Meier et al 2008). However it does not appear that these tools include the functionality required for the querying of spatial task position data.

2.4.4 – Sigtool

Sigtool is has been developed to run within MATLAB, the software allows analysis of neural data. It is intended for spike detection, spike train analysis and waveform analysis, not the querying of spatial tasks. (Sigtool 2010)

2.4.5 – NeuroDB

Is a database based common data format from neuroshare it allows storage of neural data and the ability to import a subset of the data into MATLAB using SQL to query the database (NeuroDB 2010).

2.4.6 – Summary

The majority of neuroscience software suites and projects available are concerned with broader aims than this project, and generally they are more concerned with initial experimental data extraction than analysis. Their aims are large, to create platforms for general neuroscience data through standard file formats, databases and data importing tools. The additional tools available were largely aimed at sorting and importing data, such as spike clustering in order to retrieve spike data from neural recordings. There are data visualisation and analysis tools included with some of these projects, however they are again general in approach and scope and did not meet the requirements that this project aimed to address. From the literature available no mention of a tool that could query the data from spatial tasks by excluding and include paths run in a precise, flexible and visual way was found.

2.5 Background Summary

This background section has explained the background of decision making by describing experimental data and computational models. It has explained the decision making experiment which this project is concerned with (the T-maze task), described the current computation model and outlined open research questions. Finally this section has explained the requirement for a software toolkit to aid in the answering of these questions and shown that such software was unable to be found currently available. The following methods section describes the developed software solution, how it was used to aid analysis in answering research questions and the data used for analysis.

3 – Methods

The methods section is divided into three subsections, initially there will be a brief section describing the data recorded from the T-maze experiment (see T-maze Task for details) as methods of this project heavily relied upon it. The two main subsections are divided in order to show the methods of the two separate aims of the project. Firstly the software toolbox developed MQL (Maze Query Language) will be described, secondly the use of MQL in analysis of data in order to answer research questions will be explained.

3.1 - Data

Data was collected from 6 rats with between 18-35 minutes of recording per rat (average of 21 minutes) using neuralynx software. The rats are trained to correctly complete a pattern of turns in the T-maze task (for details of the task see The T-maze Task) for 14 days before implantation of a head electrode, in this time rats achieved an average success rate on the task of 83% ± 5% standard error. The data consisted of two types: position data recorded as x and y coordinates of the head implanted electrode, sampled 30 times a second, and neural spike train data in the form of timestamps when spikes occurred. Spikes were recorded from a total of 77 neurons in the hippocampus and 78 neurons in the prefrontal cortex (for more details see Jones and Wilson 2005).

3.2 - MQL

The MQL toolbox was written in MATLAB and is downloadable with additional documentation, screenshots and tutorials, from http://www.cs.bristol.ac.uk/research/machinelearning/mql

3.2.1 - Interpolation

Signal loss during recording resulted in some erroneous data points. When the signal is lost by the position tracker the position coordinates recorded are usually set close to (0,0) resulting in points outside of the maze close to axes. In some cases signal loss also results in positional data set to invalid areas within the maze in the space above and below the central arm (see Figure 3.2).

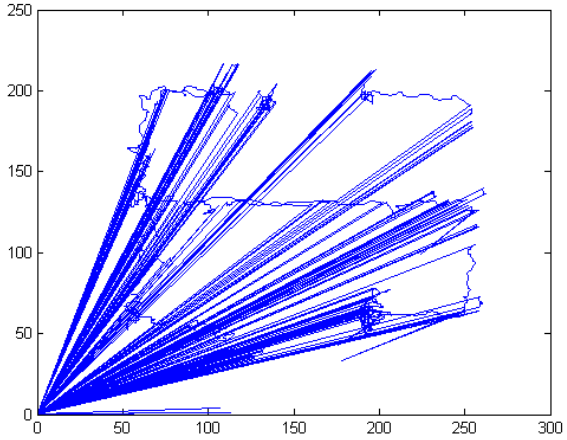

Figure 3.1: Plot of uninterpolated position data. This is a section of the data from approximately 3 minutes of recording, it shows that when signal is lost the position coordinates default to zero and that this happens frequently during a single trial.

MQL provides interpolated position data in these two cases. Data outside of the maze is identified using a user specified square perimeter of the maze (shown in black in Figure 3.2). Points outside of this will be interpolated using what we will refer to as “box interpolation”

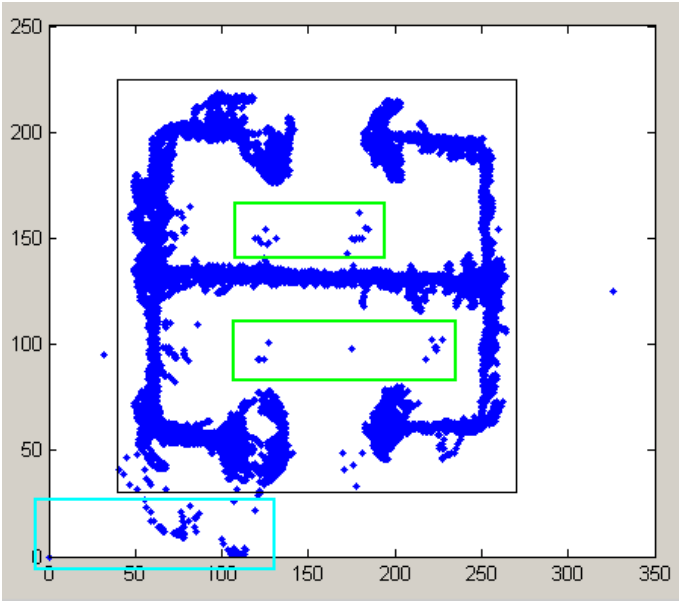

Figure 3.2: Groups of candidate invalid points shown on the maze, points outside the maze are shown in the blue box, points inside are shown in the green box

Figure 3.2 shows raw positional data before interpolation. The light blue and green boxes are not part of MQL and have been added to the image to highlight example points which demonstrate the two cases in which interpolation is required to correct data. Points in the light blue box resulting from signal loss will be identified for interpolation by the box interpolation method. The points in the green boxes show points in the spaces between arms that also require interpolation. To identify signal loss or error inside the perimeter of the maze such as the points shown within a green box in Figure 3.2, a user defined distance parameter is used, this will be referred to as “distance

interpolation". Distance interpolation looks for changes in position between two points that are greater than a user defined distance parameter representing the furthest the rat would be expected to move. In order to determine which data points require interpolation by this distance parameter two consecutive excessive movements need to be observed, the movement before the point and the movement following it. This is required because if we find only a single excessive movement between two points A and B, we cannot classify either as invalid. Both A and B are within acceptable distances of their previous and next points respectively (see Figure 3.3) which suggests that they could be valid. If (as in Figure 3.3) the point previous to A is valid position data and A is within acceptable distance of this point it is possible that A is a continuation of valid recording. The same is true for the data point following B, if it is within the distance criteria of B and is valid then we cannot be confident that B is not a valid precursor to this following valid neighbour.

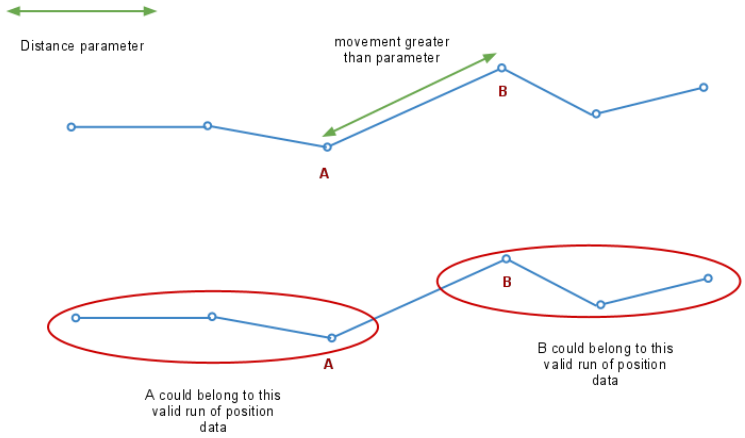

Figure 3.3: A single excessive change in position is not sufficient to select either point A or B for correction as they could both potentially be valid.

However with two excessive movements A -> B and B -> C, as shown in Figure 3.4, point B can be classified as erroneous and identified for interpolation. Figure 3.4 shows that point B is not within an acceptable distance of either of its neighbouring points, highlighting it as outside of valid recording and requiring interpolation.

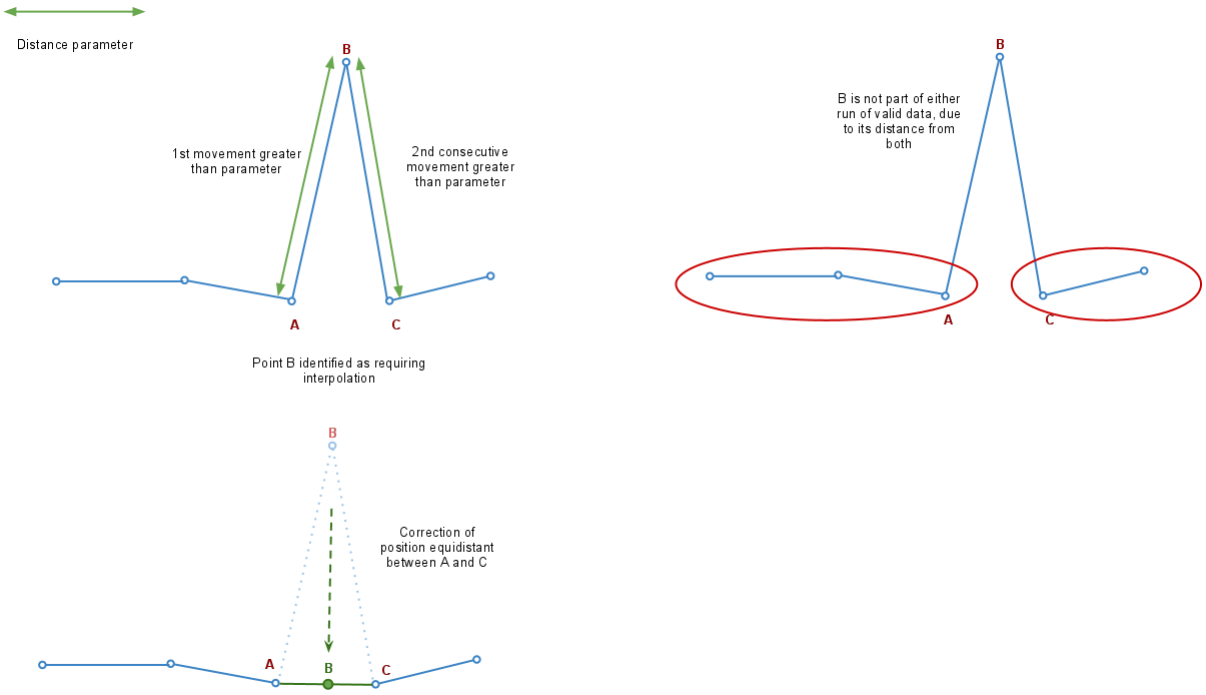

Figure 3.4: When two consecutive excessive changes in position are observed the point between them (point B) should not be classed as valid.

During the interpolation process, the corrected values between the previous and next valid point are set using the assumption of straight movement with constant running speed between the two valid points. Erroneous points outside of the maze perimeter (points identified by box interpolation) are interpolated before erroneous points within the perimeter (points identified by distance interpolation). This is to ensure that the distance constraint is not applied to points outside the maze which it is not intended for.

3.2.2 - Validity of Interpolated Data Points

MQL additionally classifies interpolated data into valid and invalid interpolations. An interpolated position is classified as valid if it satisfies two constraints, both of which are based on user defined parameters: timeout and distance. Validation is required due to the assumptions of constant speed and straight movement between valid points made during interpolation.

The timeout constraint warns against excessive signal loss. In the case of signal loss for very short periods of time interpolation is able to fill small gaps in a trial with reasonable accuracy. However the longer the signal is lost and the more consecutive position data points that must be corrected, the less confidence we can have in the accuracy of the resulting interpolation. The longer the period of time we consider, the more unlikely it is that a rat will maintain a constant running speed and straight trajectory. The timeout constraint is the maximum length of time that signal can be lost for and is expressed in terms of the maximum number of samples that can be consecutively interpolated. If the number of samples in a period of signal loss exceeds the timeout parameter the entire signal loss period is marked as invalid.

The second validation constraint applies to interpolation carried out under the distance parameter. It marks as invalid any points which once interpolated still maintain distances to their neighbouring points which exceed the distance parameter. The logic to this validity check is that if a data point was flagged as erroneous due to the distance parameter, then after interpolation has been carried out if it still violates that same constraint it should again be marked as invalid.

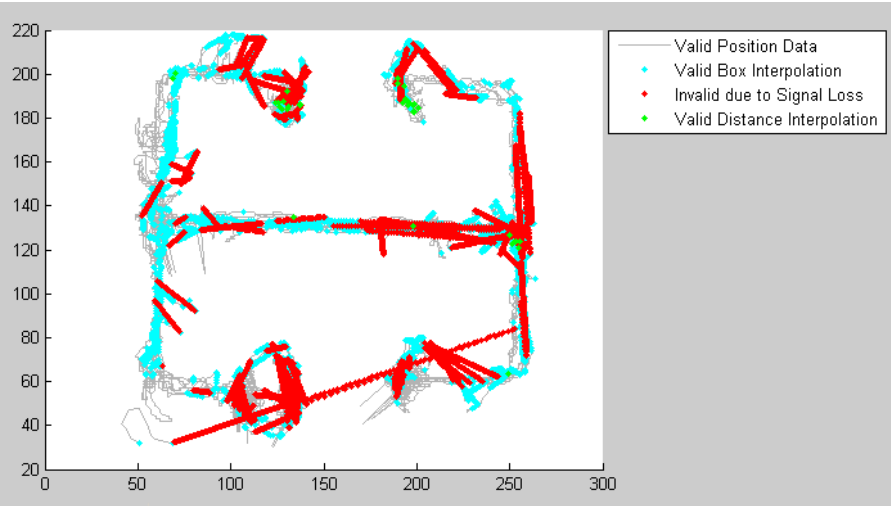

Figure 3.5: Plot of interpolated data. MQL visualises interpolation on a single figure, with the option to switch off types of valid or interpolated points using the MQL GUI.

Figure 3.5 gives an example of how interpolation is visualised in MQL, it shows the interpolation of the data used in Figure 3.2.

The validity of data is calculated in order to enable an MQL user to consider the accuracy of the interpolation and to choose to exclude certain types of invalid points dependant on their particular research questions. The parameters used for analysis were a timeout of 30 samples equivalent to 0.999 seconds and a change in (x,y) coordinate distance of 30.

3.2.3 - Querying

In order to allow analysis of data from any given type of T-maze trial and from any section of the maze a system was developed to enable the flexible selection of position data on the basis of user specified constraints called queries.

3.2.4 - Query Lines

Queries created by MQL consist of a number of constraints and return the subset of position data which meets all of them. Each constraint is a horizontal or vertical “query line” that intersects with part of the maze. An MQL created query uses these lines to specify a desired trajectory. When the query is run MQL collects series of chronological position data which cross all the query lines in the order they are defined, i.e. the line defined by query line 01 must be crossed first. Additional “avoid”

constraints can be added. These are query lines used to remove trajectories that intersect them in order to fine tune queries.

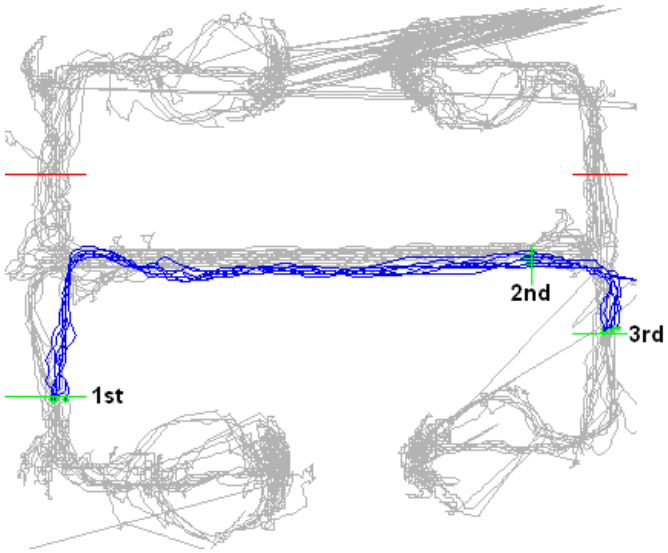

Figure 3.6: Query to return correct trials which begin at the lower left reward point. Query lines are shown in green, avoid lines in red (see section on Avoid Lines), all position data is shown in grey and trajectories which satisfy the MQL query are shown in blue. Numbers do not appear in MQL software they have been added to the Figure for additional clarity. In MQL queries are numbered in a listbox and can be selected and amended using the GUI, for details on this process see the tutorial on the MQL website.

The query in Figure 3.6 illustrates how queries are constructed to return a specific trajectory. In this case the trajectories we want to retrieve are all the correct choice trials (see The T-maze Task for details) which start from lower left reward point R2, run up the lower left arm, make a guided turn enforced by the barrier into the central arm, run along the central arm from left to right toward the choice turn and then make the correct right turn down into the lower right arm and into to reward point R4.

The first query line, intersecting with the lower left arm, ensures that correctly selected trajectories start in that arm and come from the lower left reward point R2. The second query line is a vertical line placed in the central arm just before the choice turn to make sure that the trajectory passes through the central arm. The third and final query line passing through the lower right arm ensures the run is a correct trial, finishing with the rat running to the lower right reward point R4. The second query line could be placed anywhere in the central arm and return the exact same trajectories. The third query line could also have been positioned anywhere along the bottom right arm. However the reason for the precise placement of the second and third query lines is due to the research question being asked. As this query was created with the investigation of the choice turn in mind, the choice of query lines 2 and 3 gives the timestamps for the beginning and end of the choice turn on a correct choice trial. This allows the calculation of the time taken to make the correct turn and the firing rate of neurons during the turn for each of the satisfactory trials MQL finds.

MQL uses query lines to export the timestamps at which they are crossed, providing data to calculate timings and firing rates occurring between query lines. The timestamps are exported as an *n×m* matrix where *n* is the number of trajectories and *m* the number of query lines. Also exported from MQL are the coordinates of the query lines, avoid lines and the parameters used for interpolation. This is to ensure that users of MQL can reuse interpolation parameters and query line coordinates for consistency in analysis of a previously used dataset. Also exported is the validity of the position data during the point at which the trajectory crosses the query line (see Validity at query lines for details).

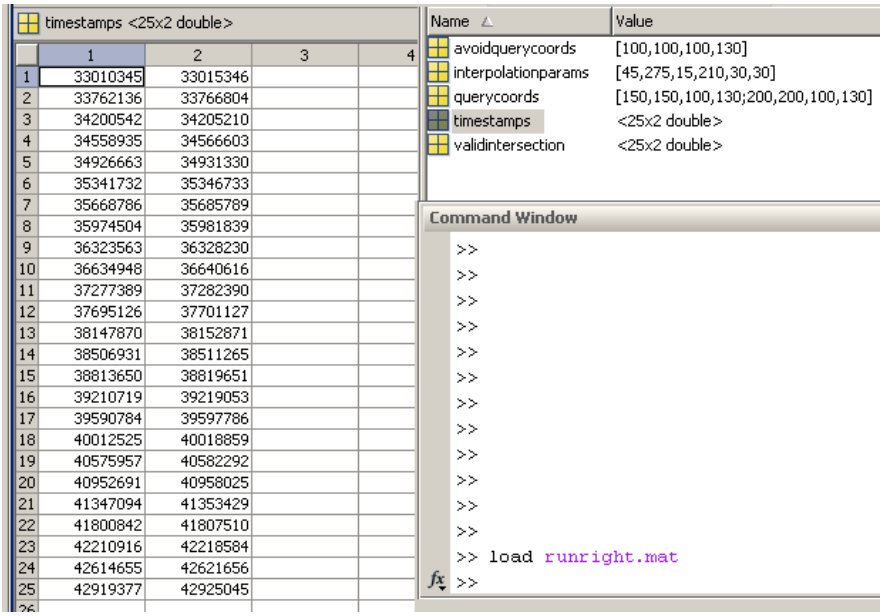

Figure 3.7: Exported timestamp data from a query with two query lines in the same manner as the query in Figure 3.10, (Exact details the data is available in the MQL tutorial http://www.cs.bristol.ac.uk/research/machinelearning/mql). The first column contains timestamps of valid trajectories which crossed the first query line (Query 01) the second contains timestamps at which the second query line (Query 02) was crossed. The first query line must be crossed before the second for the trajectory to be valid. The variables shown in the top right hand box are the variables output by the MQL export process. avoidquerycoords and querycoords are the coordinates of the avoid line and query lines in the format [xmin, xmax, ymin, ymax] lines are seperated by semicolons. Interpolationparams are the parameters of the interpolation process in the format [box xmin, box xmax, box ymin, box ymax, timeout number of samples, distance parameter]. Valid intersection is the validity at the crossing of each of the query lines for each trajectory returned (for more details see Validity at Query Lines)

3.2.5 - Avoid Lines

The red avoid lines shown in Figure 3.6 are required for some queries as in some cases the constraints specified by query lines could return trajectories other than the desired ones. A good example is the counter part to the query in Figure 3.6 an error trial starting at the lower left reward point R2. The only change required in terms of query lines, from the trajectory of the correct trial starting at R2 is to move the horizontal third query line to the upper right arm of the maze, in order

to now specify trajectories that start at the lower left reward point R2, make an error in turning left at the choice turn and finish at the upper right reward point R3. While the new query will return the trajectory of the error trial starting at R2 it also returns many additional trajectories (shown in Figure 3.8). In order to only obtain the error trial trajectories an avoid line must be added to the lower right arm to remove any trajectories that pass through it.

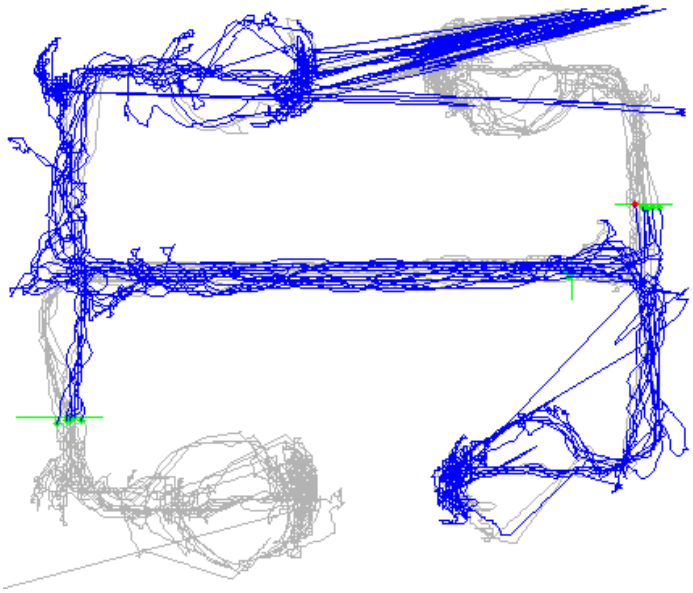

Figure 3.8: Query to select error trial trajectories starting from lower left reward point R2 and finishing in upper right reward point R3. Query is constructed using only query lines and returns many unwanted trajectories in addition to the desired ones. Trajectories in the upper left arm, reward point R1, lower right arm and reward point R4 are returned. The trajectories desired should only have position data in the lower left arm, central arm and upper right arm.

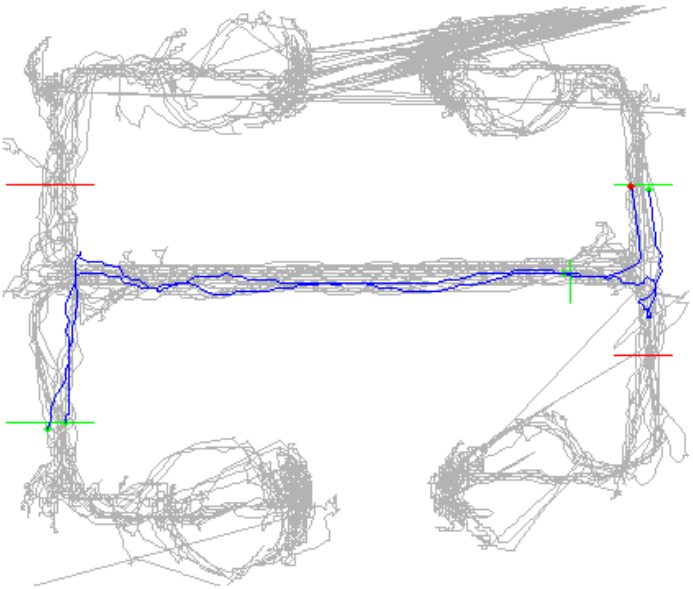

Figure 3.9: Query to select trajectories starting from lower left reward point R2 and finishing in upper right reward point R3. Two Avoid lines have been used in addition to the query lines, they are shown in red and cross the upper left and lower right arms to remove unwanted trajectories that intersect them in these arms. Using query lines and avoid lines leaves only the desired trajectories.

3.2.6 - Validity at Query Lines

Figures 3.8 and 3.9 show a red dot at the point of intersection with the third query line for one of the trials. This is a visual representation of validity of data at query lines, a green dot if data is valid, red if data is invalid. Validity of all points is calculated during interpolation (see Validity of Interpolated Data Points). Initially in its development MQL excluded trajectories with invalid data but this led to very few valid trajectories being returned. MQL was modified to identify all trials but, additional to the timestamps output, to return the validity of the two data points that cross each query line so that users of MQL can decide which trajectories to consider valid for their analysis. Validity is exported with timestamps in the same format, a $n \times m$ matrix where n is the number of trajectories and m the number of query lines. Rather than timestamps the matrix contains boolean values, 1 if data is valid at the query line and 0 if not valid. This allows trials where the position data is unreliable at places of importance in the maze to be discarded during analysis. When analysing the firing rate of neurons, the position of the rat is not important during the choice turn, what matters is that the timestamps for entering and exiting the turn are reliable. The same is true for all other positional data in the trajectories returned, invalid data during the initial guided turn, lower left arm or central arm does not affect the analysis. Validity at query line data also allows other areas in the maze to have signal loss if they are not required for the current analysis. For the analysis of timing and firing in the choice turn, unreliable position data due to signal loss is permitted during the turn. This does not influence the accuracy of any analysis of how long rats take to run the turn so long as the points at which the rat started and finished the turn (the query lines) are accurate. All of the analyses carried out here used validity at the point of query lines as a way of deciding which trajectories to class as invalid and therefore not to include in analysis.

3.3 - Analysis Using MQL

Data analysis was carried out using the MQL software toolbox to define queries used to obtain timestamps from trajectories of interest in order to answer research questions.

3.3.1 - Choice Turn Queries

The four queries used in the analysis of behaviour during the choice turn consisted of those shown in Figures 3.7 and 3.9 plus additional queries shown in Figure 3.10 below. These four queries select the trajectories of the four choice trials, two correct and two error (see T-maze Task for details of trials). Using the trajectories returned by these queries to export the timestamps of the beginning and end of choice turn in these four cases, it was possible to compare timing and neural activity on correct versus error trials.

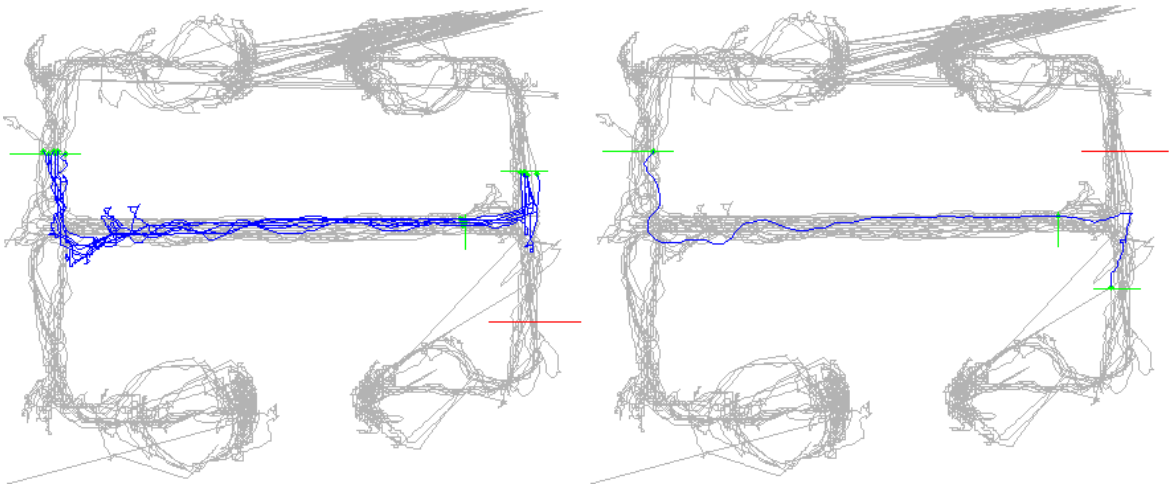

Figure 3.10: Query line defined trajectories used for the analysis of timing information and firing rates occurring during correct choice turns beginning from the upper reward point (left) and error choice turns beginning from the upper reward point (right). Query lines in green, avoid lines in red, all recorded data in grey, MQL returned trajectories in blue.

3.3.2 - Turn Neuron Queries

A prediction of the Gorochowski model (2009) that neurons selective for turns should exist was investigated. This verification was performed by analysing activity from individual neurons during all the turns in the maze. In order to achieve this MQL was used to create 16 queries for each maze, a left turn and right turn query for each of the possible 8 turns in the maze. The trajectories collected for left turns were concatenated to give an *n×2* array (where *n* is the total number of trajectories) of timestamp pairs that represent the start(left column) and finish (right column) of each left turn trajectory in the maze. The same was done for right turns enabling a comparison of average firing rates left versus right turns.

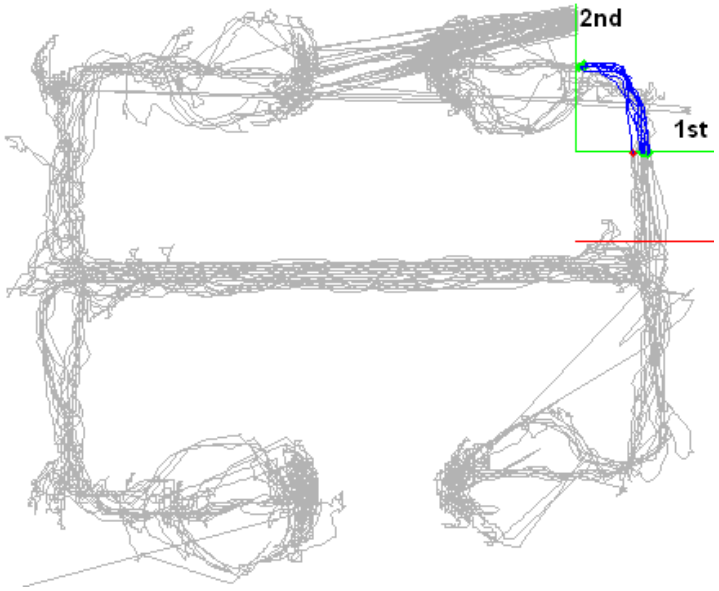

Figure 3.11: One of the 16 turn queries for the maze. This query returns trajectories from a left turn to the top right reward point. Query lines shown in green, avoid lines red, returned trajectories in

blue. Numbers show the order the query lines must be crossed in and that therefore this query selects a left turn, reversing this order would give the query for turning right away from the reward point.

3.3.3 - Turn Approaching Queries

To look at the activity of turn selective neurons leading up to and during the choice turn queries similar to those described in the Choice Turn Queries section were used. There were four queries which select the trajectories of the four types of choice trial: the correct and error trials starting at the upper left reward point and the correct and error trials starting from the lower left reward point. In order to analyse firing rates as the choice turn was approached along the central arm addition vertical query lines were added. This created a number of intervals along the central arm at which timestamps could be obtained.

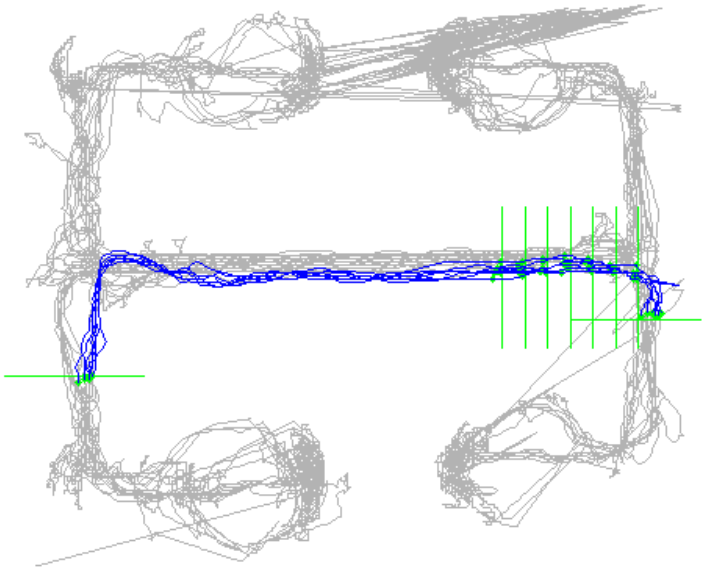

Figure 3.12: The turn approaching query which selects trajectories of correct trials starting at the lower left reward point, note the additional vertical query lines in the central arm that return the timestamps of intervals leading up to the choice turn. Query lines are shown in green, returned trajectories in blue, all other recorded position data in grey.

3.3.4 - Calculation of Timing and Firing Rate

In order to analyse the timestamps obtained from the aforementioned queries to answer research questions, custom MATLAB functions were written outside of MQL. These custom functions were used to load the positional and neural activity data and calculate average time and average firing rates for individual or pooled neurons between timestamps. Average time was calculated as the difference between two timestamps averaged over the number of trials. Average firing rates for neurons we calculated by counting the number of spikes occurring during an interval and dividing by the interval length. A number of statistical tests were built into these functions in order to compare results for statistical significance details of which are found in the following results section.

4 – Results

This section details the results of the analyses carried out in the previous methods section, where required details of statistical tests are provided. Following the results of some analyses further questions were posed and therefore additional analyses were carried out, this is the case for the question of turn selective neurons which prompted additional analyses such as the analysis of activity leading up to and including the choice turn. All queries for the analyses in this section are described in the Analysis Using MQL subsection, addition statistical tests are however described alongside results.

4.1 - Choice Turn Analysis

This analysis is based on initial analysis by Bogacz(2010) where the central arm of the maze was analysed to look for difference between correct and error trials(details in Prediction of Model and Further Analysis of T-maze Data). The analysis showed that rats spend significantly longer in the central arm on error trials and suggested that the analysis of the choice turn could provide addition insight. It is possible that errors are due to an exploratory decision and that we might expect to see increased neural activity in the pfc on error trials if this is the case (Daw et al (2006)). Using timestamps obtained from the MQL queries described in the Choice Turn Queries section analysis of timing and average firing rates were carried out.

4.1.1 - Timing

Do rats take longer to make the choice turn on error trials than on correct trials?

4.1.1.1 – Results

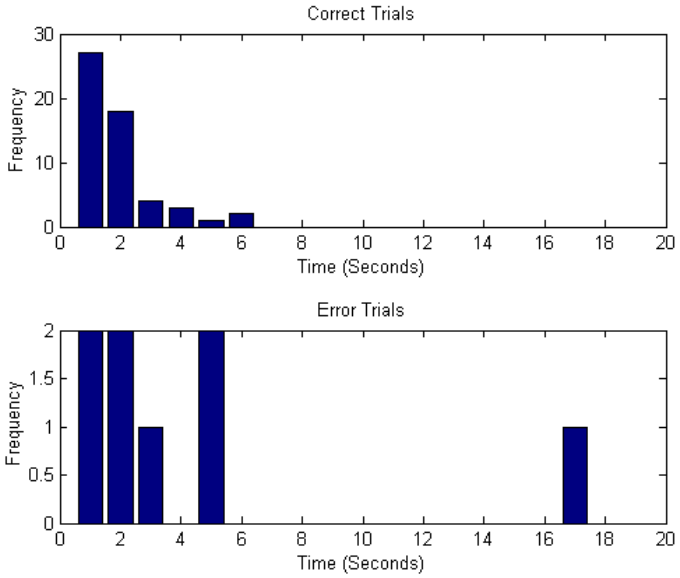

Figure 4.1: Histogram showing the distribution of time taken to make the choice turn for correct and error trials

Trial	Median time taken to make choice turn
Error	2.08 seconds
Correct	1.06 seconds

Table 4.1 – showing the median time in the choice turn for correct and error trials

P value of Wilcoxon rank sum test of correct versus error times: 0.02132

The Wilcoxon rank sum test was used on the two samples of times (time take for the choice turn on correct and error trials) to test for a significant difference. It does so by calculating the probability of observing the two samples if they came from distributions with the same median. The rank sum test was used rather than a standard paired ttest, as the timings are not normally distributed, an assumption made by the ttest and not by rank sum. The result of the ranksum shows that there is a probability of 2.132% that the samples are from the same distribution. A probability of less than 5% suggests that this is not due to chance and that therefore the timings on correct and error trials are significantly different. The median times taken during choice turns show that the significant difference is due to increased time during turns on error trials with a median of nearly double that of correct trials. This can also be seen in the histograms in Figure 4.1 where correct trials are found most densely around 1-2 seconds and error trials more evenly distributed between 1-5 seconds. Median values were used so as to not allow the outlying error result (shown in Figure 4.1) to overly influence results.

4.1.1.2 - Summary

Results suggest the rats do take longer to make the choice turn during error trials, showing a significant difference between times for correct and error trials, with error trials taking longer on average.

4.1.2 - Firing Rate

Is there a difference in neural activity in the hippocampus or prefrontal cortex during the choice turn when comparing correct and error trials?

4.1.2.1 – Wilcoxon signed rank test

Significance values could not be calculated for individual neurons due to low numbers of error trials, therefore the significance was calculated for each brain region. This was done by taking the average firing rate of each pfc and hpc neuron and pooling them to give a large sample of firing rates during the choice turn on correct and error trials for each brain region. A Wilcoxon signed rank test (signrank) was used to compare firing rates during correct and error trials for statistical significance. Signrank test was chosen as the individual neurons have highly variable firing rates. This is shown in the histograms where some neurons fire at average of around 30 times a second and others fire less than once a second, therefore considering them as samples from the same distribution is an incorrect assumption. Signrank accounts for this variability by taking the difference of two samples to give a single distribution, signrank then calculates the probability that the mean of this distribution is 0. This subtraction removes the fact of a general high or low firing rate of a neuron and allows only the difference in firing to be compared.

4.1.2.2 - Results

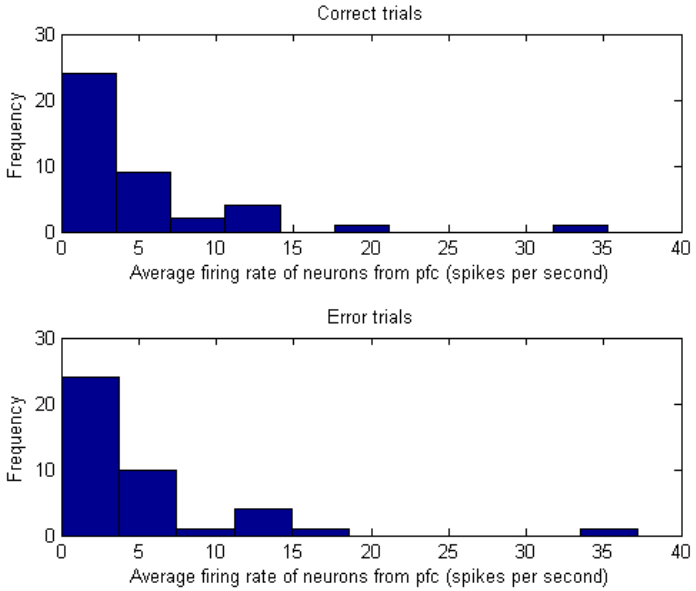

Figure 4.2: Histogram showing distribution of average firing rates from neurons in the prefrontal cortex during the choice turn of correct and error trials.

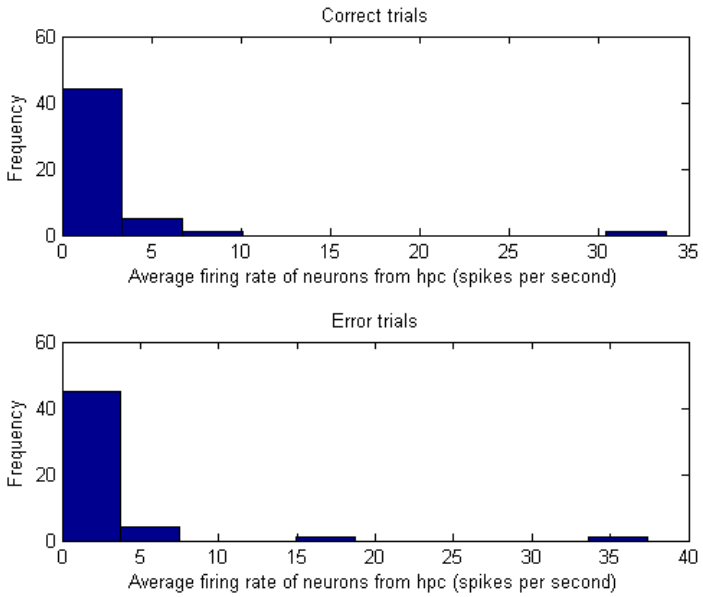

Figure 4.3: Histogram showing distribution of average firing rates from neurons in the hippocampus during the choice turn of correct and error trials.

	Correct trials	Error trials	p value
Hippocampal neurons:	1.51 spikes per second	1.66 spikes per second	0.62
Prefrontal neurons:	4.82 spikes per second	4.89 spikes per second	0.98

Table 4.2: Shows the average firing rate on correct and error trials of all the neurons in each region as well as significance values of the Wilcoxon signed rank comparison between the two trials.

The firing rates shown in histograms are the average firing rates during the choice turn for correct and error trials for each neuron. Neurons are shown from four of the six rats totalling 41 from the prefrontal cortex and 51 from the hippocampus, data from only four rats was used as the others did not have recordings of error trials.

P values returned by the signrank test of 62% for hpc and 98% for pfc are much higher than the significance threshold of 5% showing that there is not a significant difference. The histograms and average firing rates support this showing similar firing rates for correct and error trials in both hippocampal and prefrontal neurons. The average values shown in the table are an average of the average firing rates shown in the histograms. This is to ensure that a rat with more trials has the same influence on the results as a rat with fewer trials.

4.1.2.3 - Summary

Statistical tests show no significant difference in activity between correct and error trials for either regions of the brain. Results show that not only is the activity of the pfc not significantly different during choice trials but that the activity is at very similar levels for correct and error trials with activity during error trials being greater by 0.07 spikes per second.

4.2 - Turn Selective Neurons

Do turn selective neurons exist and if so which area of the brain are they found in?

The Gorochowski model predicts that in the spatial task we should expect to find neurons selective for turns, neurons that are active for a particular direction turn and significantly less active during the opposite direction of turn.

4.2.1 - Two Way Analysis of Variance

In order to detect turn selective neurons a two way analysis of variance (anova2) statistical test was employed. Given the average firing rate of a particular neuron during each of the eight left turns and each of the right turns rather than calculating if the average firing rate during one turn is significantly different to the other turns, the anova2 is able to calculate if average firing rates during one set of turns are significantly different to average firing rates during the set of turns in the opposing direction. This allows neurons significantly more active in one turn direction than the other direction to be found.

4.2.2 - Results

Neuron	Rat	Region	P value	Turn direction selected for (preferred turn direction)
1	5	PFC	0.0012	Left
2	5	PFC	0.0101	Left
3	5	PFC	0.0019	Right
5	5	PFC	0.0015	Left
10	5	PFC	0.0274	Left
14	5	PFC	0.0231	Right
17	5	PFC	0.0147	Right
18	5	PFC	0.0293	Left
1	6	PFC	0.0084	Right
2	6	PFC	0.0005	Left
3	6	PFC	0.0024	Left
4	6	PFC	0.0136	Right
5	6	PFC	0.0074	Left
6	6	PFC	0.0188	Right
8	6	PFC	0.0253	Right
12	6	PFC	0.0154	Right
14	6	PFC	0.0088	Left
15	6	PFC	0.0364	Right
16	6	PFC	0.0012	Left

Table 3: list of the turn selective neurons identified by two way analysis of variance. Columns (left to right): Rat number is which of the 6 rats the neuron was recorded from, each neuron recorded from has a number this is shown in the neuron column, region shows whether the neuron is from the

prefrontal cortex (pfc) or hippocampus (hpc), P value is generated by anova2, preferred turn specifies which turn direction the neuron selects for, i.e. which direction it has higher activity during.

The anova2 highlighted 19 neurons in two rats which were selective for a particular turn direction. 10 are selective for left turns and 9 for right turns, this direction with higher activity will be referred to as the 'preferred turn' with the opposite turn direction referred to as the 'unpreferred turn'.

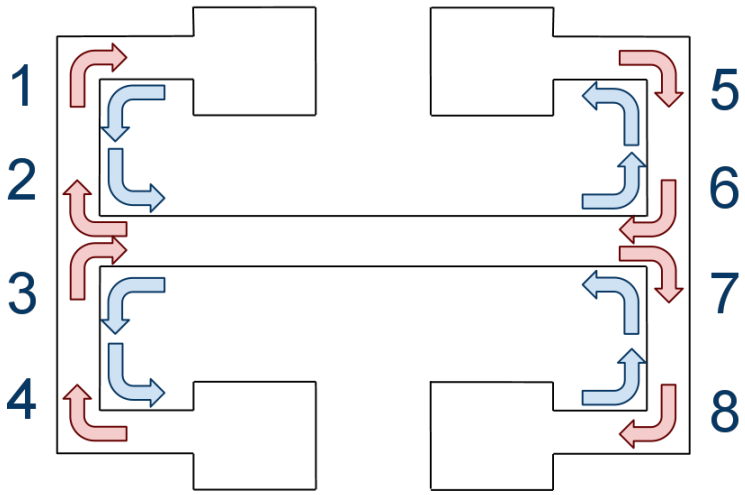

Figure *4.4: Visualisation of all the left and right turns in the maze, left turns shown with blue arrows, right turns shown with red arrows.*

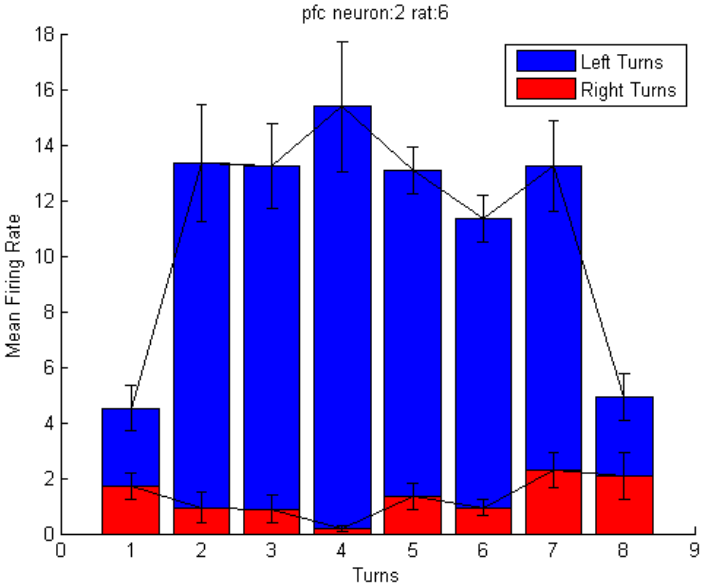

Figure 4.5: Example of a neuron selective for left turns. It shows higher average firing rates during all 8 left turns than during all 8 right turns.

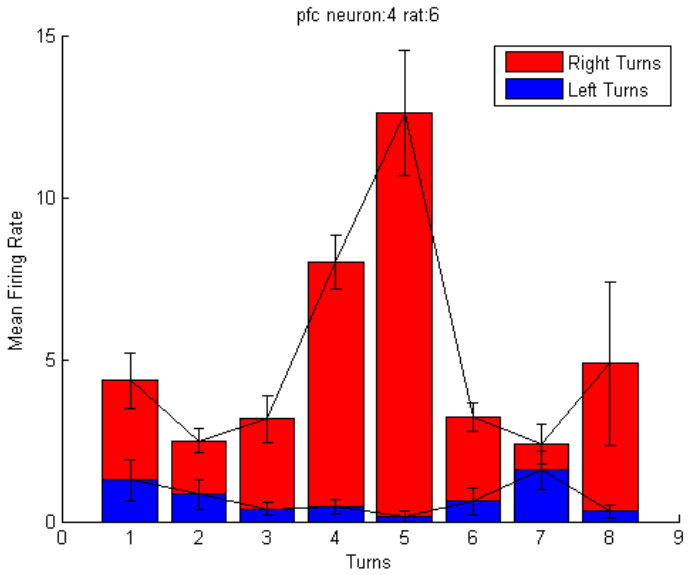

Figure 4.6: Example of a neuron selective for right turns. It shows higher average firing rates during all 8 right turns than during all 8 right turns.

The Figures show that these neurons are selective for left and right turns respectively as they have consistently far higher activity during their preferred turn direction compared to during their unpreferred turn. This difference in firing is reflected in their low anova2 p values of 0.0005 for neuron 2 from rat 6 and 0.0136 for neuron 4 from rat 6. Some firing occurs during unpreferred turns, however this is to be expected due to the significant amount noise present in data from the brain.

4.2.3 - Summary

19 turn selective neurons were found in the prefrontal cortex of 2 rats using a two way analysis of variance, they show significantly more activity during all 8 turns in one direction than during the 8 turns in the opposite direction.

4.3 Comparing All Turns

Are turn neurons selective for a particular turn or turns rather than all turns?

As the function of the prefrontal cortex is thought to be the learning of rules and strategies that guide behaviour (Wallis et al 2001) it is possible that the turn neurons found could simply be selective for a single turn that is important to the task such as the choice turn and that this the reason they appear significant.

4.3.1 – Results

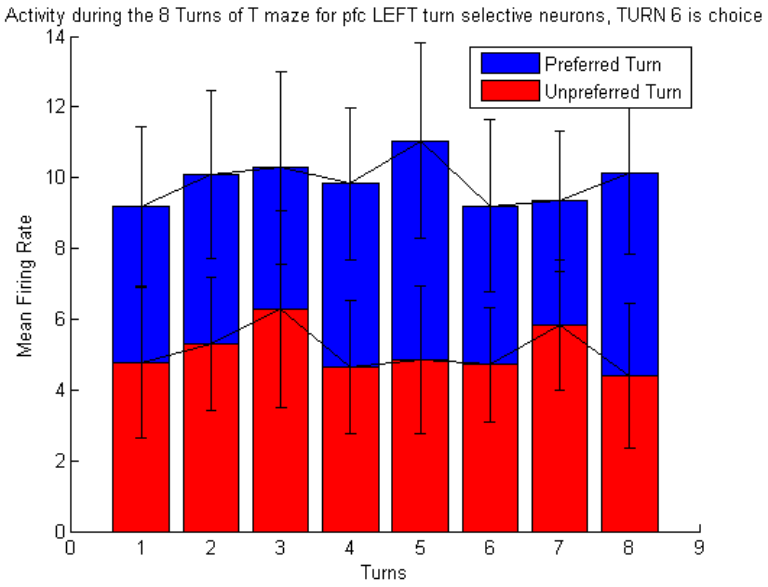

Figure 4.7: The average activity at each of the turns for left turn preferring neurons

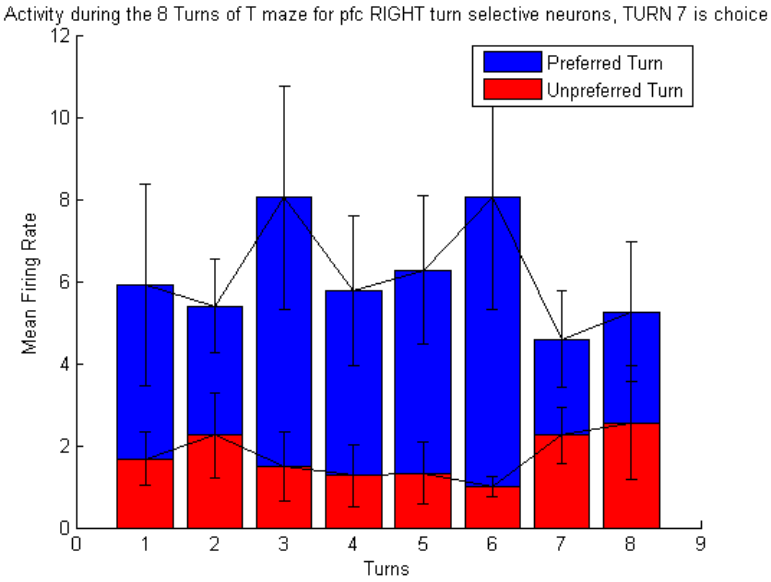

Figure 4.8: The average activity at each of the turn for right turn preferring neurons

The Figures illustrate that average firing rates averaged over all 19 turn selective neurons identified by the anova2 shows similar firing rates across all 8 turns during the preferred turn direction. The same is true for average firing rates during the unpreferred turn direction which are consistently lower than during the preferred turn. If for example firing rates during the choice turns were affecting the previous analysis and the turn neurons were selective for the choice turn, we might expect to see higher average firing rates for left preferred neurons at turn 6 and right preferred neurons at turn 7.

The reason for separating the right and left preferring turn selective neurons is due to the numbering of turns. When analysing firing rates at the turns of the maze the context of the turn is

important, turns 6 and 7 are only choice turn when turning out of the central arm into the lower or upper right arm, therefore a preferred right turn on turn 6 is not a choice turn but the first turn made in a guided trial. Averaging both sets of neurons would remove the context of the turn by having two different preferred turns at each of the maze turns.

4.3.2 - Summary

It does not appear that a single turn is influencing the two way analysis of variance. Results show similar average firing rates during all preferred turns adding to the conclusion that the turn selective neurons found select for all turns in the maze regardless of context.

4.4 - Firing rates of turn neurons before and during the choice turn

Do turn neurons increase their firing rate before the choice turn when making a preferred turn? Additionally, do turn neurons decrease their firing rate on non-preferred turns?

The Gorochowski computational model suggests that turn selective neurons are fed into by working memory neurons and that as part of the decision making process should increase their firing rate before a decision is made if the turn made is preferred. Conversely the neurons selective for the unpreferred turn should decrease their firing rate. This is due to the model being based on the LCA model (described in Leaky Competing Accumulator Model) where evidence is accumulated over time before a decision with the winning choice inhibiting the alternative over time. This effect of an increase in firing rate has been observed experimentally by Shadlen and Newsome (2001).

4.4.1 – Results

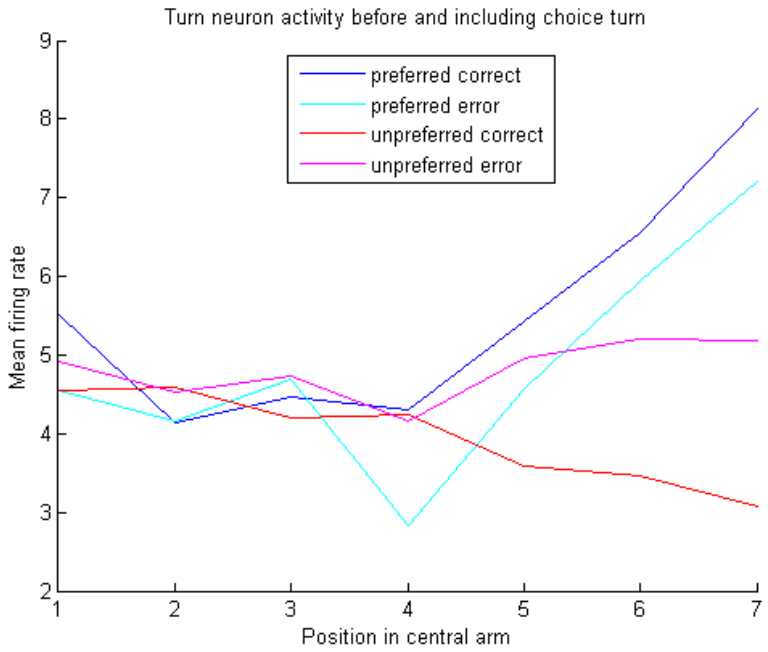

Figure 4.9: Average firing rate of all turn selective neurons leading to and during the choice turn. Positions 1 to 6 are average firing rates during intervals leading to the choice turn (position 1 is further from the choice turn than position 2), position 7 is the average firing rate of all the turn selective neurons during the choice turn. Preferred and unpreferred terminology is the direction of the choice turn in relation to the selectivity of the turn selective neurons and is used in order to enable both sets of turn selective neurons to be included on one plot.

Each position represents from left to right the intervals between query lines in Figure 3.12 with the last position the interval where the rat makes the choice turn. The activity value on the x axis is an average firing rate calculated from all 19 turn neurons from their individual average activity during the given interval. The four lines plotted on the above figure represent the average firing rate during the intervals on the four types of choice trial (two correct trials and two error trials described in the T-maze section). The preferred and unpreferred terms define the direction of the choice turn and whether it is the direction a turn neurons selects for. For example a correct preferred trial For a right turn selective neuron this is a choice trial starting from the bottom left reward point (R2) with a right choice turn, for a left turn selective neuron it is during a trial starting at the top left reward point (R1) involving a left choice turn. Using the preferred and unpreferred terms allows firing rates from both sets of turn selective neurons to be combined.

Figure 4.9 shows that firing rate of turn selective neurons during the four types of choice trial is at a similar level on the approach to the turn, however after interval 4 as the turn is neared activity for the preferred trials (blue and light blue) increases showing selectivity for the preferred turn that is about to be made. The unpreferred correct trial (red) shows a decrease in average firing rate showing selectivity for the preferred turn that will be made. The unpreferred error (pink) increases at a slower rate. In order to determine whether these changes in average firing rate for turn selective neurons during the 4 choice trials between the intervals 4 and 7 was significant, statistical

tests were used. Due to the variable firing rates of neurons signrank was again used to test for statistical significance between intervals 4 and 7. It used the samples of the average firing rate from each of the 19 neurons during the intervals 4 and 7 for each trial.

Trial	P value from signrank comparison of activity at intervals 4 and 7
preferred correct (Blue)	0.002902
preferred error (Light Blue)	0.008607
unpreferred correct (Red)	0.084017
unpreferred error (Pink)	0.407966

Table 4.3: P values for comparison of average firing rates of turn selective neurons during intervals 4 and 7. This test looks for a significant change in the average firing rate between the approach of the choice turn and during the choice turn.

Both of the preferred turns (blue and light blue) show a significant increase in average firing rate of turn selective neurons over the last 3 intervals with p values much lower than 5%. The unpreferred correct trial (red) shows a decrease in firing rate between intervals 4 and 7, but with a p value of 7.7% it does not decrease enough to be significant. The unpreferred error trial (pink) which shows an increase in the firing of turn neurons during it does not change significant either.

The increase in activity of turn selective neurons between intervals 4 and 7 during the unpreferred error trial (pink) might be expected. In a trial starting at the upper left reward point where the correct turn to make is left, if the rat makes an error and turns right, left turn selective neurons may increase activity before the turn as they are selective for the correct turn that should be made. Therefore as the left selective neurons are firing during a right turn they have an increased firing rate during an unpreferred turn on an error trial.

4.4.2 - Summary

For both correct and error trials, during preferred choice turns the turn selective neurons do increase their average firing rate before and during the choice turn as predicted and the increase is a significant one. During trials with an unpreferred choice turn average firng rate is shown to decrease leading up to the choice turn though not by a significant level. The average firing rate of turn selective neurons during the unpreferred error trial increases however this could show that the turn selective neurons are selective for the correct choice turn direction.

4.5 – Turn Selective Neurons Activity: Correct versus Error Trials

Are turn selective neurons less selective on error trials?

During the investigation of the previous research question the average firing rate of turn selective neurons increased during correct preferred trials and decreased during the correct unpreferred trial showing selectivity for the choice turn made. This was not the case during error trials where the average firing rate increased for the unpreferred turn direction. This is interesting as it highlights

that for the sections of the maze before and during the choice turn, turn neurons appear less selective during error trials compared to during correct trials.

4.5.1 - Results

Signrank was used to test for statistical significance between turn selective neurons firing rate during different trials at given intervals. As with the previous research question the samples compared were the average firing rate from each of the 19 neurons during the intervals for the trials shown in the tables (intervals defined in Figure 3.12 and shown in Figure 4.9).

Trial 1	Trial 2	Interval	P value
preferred correct (Blue)	unpreferred correct (Red)	6	0.022231
preferred correct (Blue)	unpreferred correct (Red)	7	0.000293
unpreferred error (Pink)	unpreferred correct (Red)	7	0.039474
preferred error (Light Blue)	unpreferred error (Pink)	7	0.077767
preferred correct (Blue)	preferred error (Light Blue)	7	0.314389

Table 4.4: P values for comparisons of average firing rates of the 19 turn selective neurons during pairs of trials at a given interval. P value show if the activity is significantly different between trial 1 and 2 during the chosen interval (colours from Figure 4.9)

Average firing rates during the correct trials (blue and red) showed a significant difference at the last intervals 6 and 7. Error trials (light blue and pink) did not show a significant difference neither did the preferred trials (blue and light blue). However the unpreferred trials (pink and red) did show a significant difference during the final turn interval.

A significant difference in average firing rate was shown for the correct trials during the choice turn but no significant difference was shown between the two error trials. In order to determine if the difference in selectivity was significant the signrank test was used to compare the difference in average firing rate between preferred and unprefered correct trials with the difference in average firing rate between preferred and unpreferred error trials. In the choice turn interval 7 signrank gives a p value of 0.006210 showing that there is a significant difference and that turn neurons are more selective for the choice turn made during correct trials than during error trials.

4.5.2 - Summary

Turn neurons are significantly more selective for the choice turn direction that is made during correct trials than during error trials in the section of the maze where the choice turn occurs.

4.6 - Reward neurons

Are there neurons selective for the behaviour of running to reward points?

During analysis to identify turn selective neurons the average firing rate during turns of all individual neurons was investigated. Through this process two neurons of particular interest were discovered that appear to be selective for a particular behaviour.

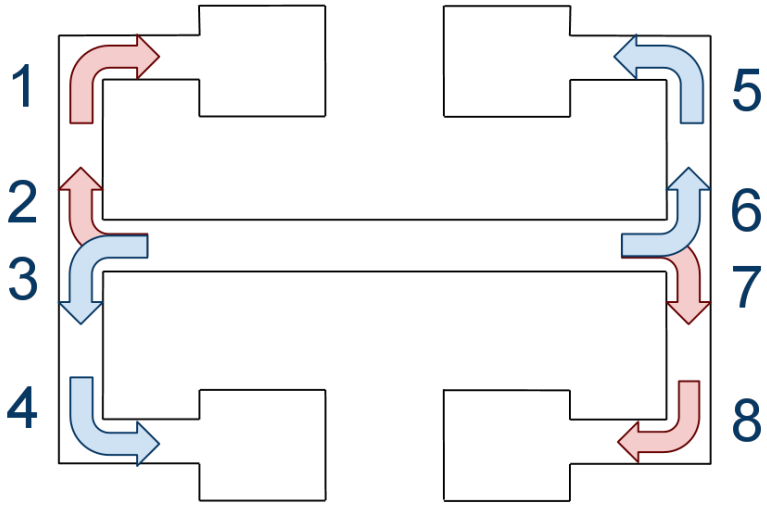

Figure 4.10: Diagram showing 'reward turns' turns that lead away from the central arm to the reward points where rats receive a reward. Left turns are shown with blue arrows, right turns with red arrows.

4.6.1 - Results

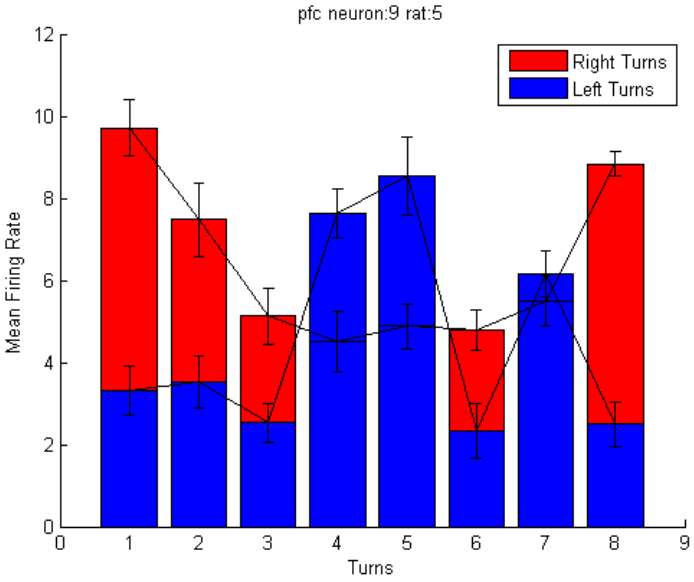

Figure 4.11: Reward selective neuron showing higher firing rates during the last 4 reward turns, right on 1 and 8 left on 4 and 5

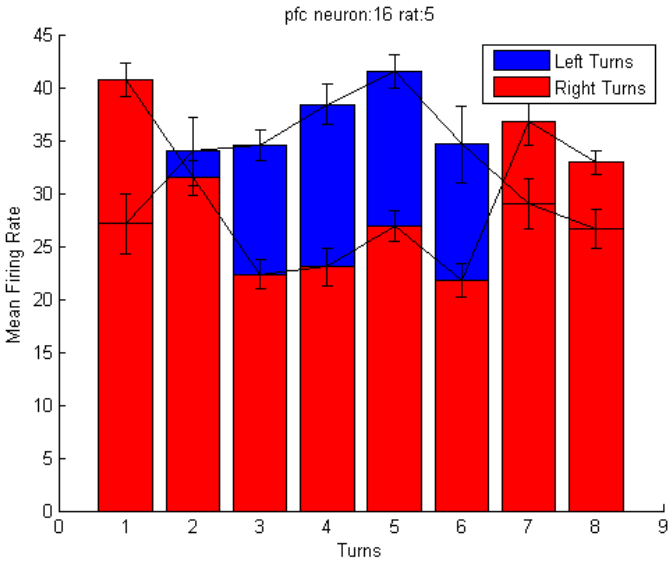

Figure 4.12: Reward selective neuron showing higher firing rates during 7 of the 8 reward turns, right on 1,2,7 and 8 - left on 3,4,5 and 6

Both neurons 9 and 16 from rat 5 appear to exhibit selectivity for similar paths in the maze, they have higher activity on the turns to the reward points. Neuron 9 shows higher activity on the 4 turns directly into the reward box, right on 1 and 8, left on 4 and 5 with lower activity on turns 3, 6 and 7. Neuron 9 shows higher activity on the two consecutive turns made to each reward point, right on turns 1, 7 and 8; left on turns 3, 4, 5 and 6. The exception with both of these neurons is turn 2 which has higher activity from neuron 9 than the 3 other equivalent turns and on neuron 16 where it the average firing rate for left and right turns are very similar. However given the levels of noise in neurons and the consistency of the 7 other turns it appears these neurons are selective for specific behaviour in the T-Maze task.

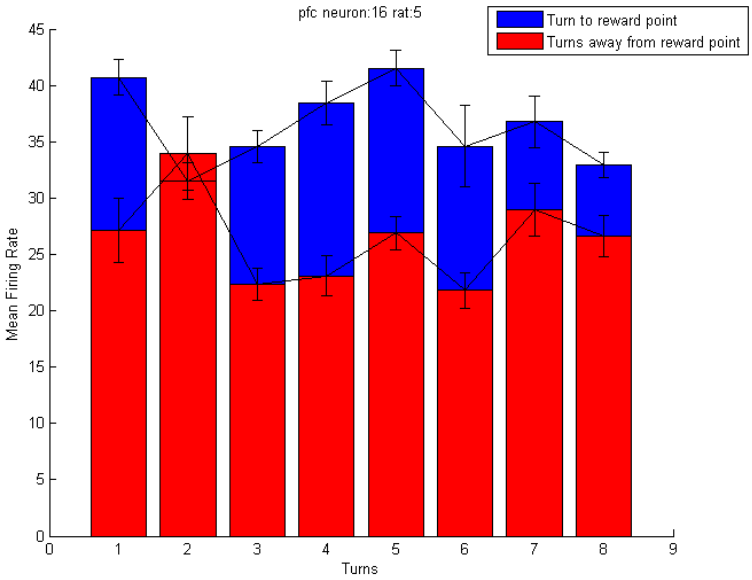

Figure 4.13: Activity during turns for neurons 16 comparing turns toward reward point with those away from the reward point.

Using a two way analysis of variance to compare activity during turns toward the reward point (right 1,2,7,8, left 3,4,5,6) with turns away from the reward point (left 1,2,7,8 right 3,4,5,6) for neurons 16 from rat 5 gave a p value of 0.0021 showing a significant difference in levels of activity. Neuron 9 from rat 5 gave a p value of 0.13 from this comparison.

4.6.2 - Summary

Neuron 16 from rat 5 appears to be selective for the behaviour of running to the reward point, it shows a significant difference in activity when comparing turns toward the reward box, with those away from it. Neuron 9 from rat 5 did not show a significant difference when comparing these turns suggesting that although it initially appears to be selective that this may not be the case.

5 - Discussion

This section will discuss the observed results and whether or not the research questions have been answered. It will compare findings to the current literature where appropriate and finally will review the aims and objectives of the project to judge if they have been completed and if so how successfully.

5.1 - Choice Turn Analysis

The results of analysis of the choice turn area of the maze were consistent with the previous analysis of the central arm (Bogacz 2010) which also showed rats significantly slower on average for error trials and no significant difference in average firing rates between correct and error trials. This result supports the error rates and reactions times for decision making tasks (described in Error Rate and Reaction Time) as when rats take longer to make the decision they are more likely to make errors. The timing result is also consistent with human studies (Ratcliffe and Rounder 1998) where humans are sometimes quicker but on average slower during errors compared to when correct. The exploration hypothesis proposed as an explanation for errors is not supported by the similar levels in overall neural activity in both brain regions. If it were the case we might expect to see some neurons with greater activity on error trials as Daw et al (2006) found the prefrontal cortex to be significantly more active during exploratory decisions. However due to the high accuracy of rats and time periods of recording, there was a lack of error trials recorded in the data. This resulted in the firing rates from individual neurons needing to be pooled in order for statistical comparisons to be made between firing rates on correct and error trials. Therefore firing rates during correct and error trials from each neuron were not compared separately which would be preferable in order to investigate this question further and determine if specific neurons are selective for exploratory turns during error trials.

5.2 - Turn Selective Neurons

All but four of the turn selective neurons identified were the same neurons as the trajectory selective identified by Bogacz(2010). This is likely a product of the way data is encoded in the pfc, whereby neurons are used to encode multiple different concepts (Wallis et al 2001) and therefore it is proposed that the neurons responsible for remembering the direction of the last turn could also be selective for the direction of the current turn. The turn selective neurons were only found in rats 5 and 6, a possible explanation for this is the number of neurons recording from in each rat with 18 and 16 pfc neurons recorded from in rats 5 and 6 respectively. This was a higher number of pfc neurons than rats 1, 2 and 3 which had 6, 8 and 7 neurons recorded from. Rat 4 had 23 neurons recorded from it but did not have positional data recorded for all of the turns in the maze and so was not used for this analysis. A greater number of neurons recorded from increases the chances of finding particular types of neuron.

We are not aware of any previous reports of turn selective neurons in the prefrontal cortex, only neurons selective for parts of a maze associated with particular behaviours (Jung et al 1998). The parietal cortex is a region more likely to contain turn selective neurons as it is the area of the brain thought to be involved in spatial navigation (Nitz 2009, Whitlock et al 2008). A hypothesis for the

presence of turn selective neurons being found in the prefrontal cortex is that combinations of turns are important for success in the T-Maze task and therefore the ability to remember the turns made everywhere in the maze are part of a strategy that develops over time as the rat learns the task. Therefore turn neurons may be present in the parietal cortex but as rats learn the task and develop strategies we may see turn neurons begin to appear in the prefrontal cortex. As all data collected was from rats already trained in the T-maze task this analysis was not possible but with recording from rats during learning stages and from electrodes in the parietal cortex this question could be investigated.

5.3 – Activity Leading to and During Choice Turn

When analysing the firing rates of turn selective neurons, activity during preferred trials was shown to increase. This is consistent with the predictions of accumulation based computational models (described in section Computational Models of Decision Making) and with the experimental results of Shadlen and Newsome (2001) whereby before a decision can be made neurons representing evidence for a choice increase their activity. If these neurons encode a particular turn direction then it follows that when a decision is being made between left and right they should follow this pattern of increasing their activity before and this is shown to be the case.

The firing rate of turn selective neurons during the correct trials also appears to support the LCA model as during the correct trials unpreferred neurons do not increase their firing rate but appear to show inhibition from the preferred neurons to a level where during the choice turn there is a significant difference in their activity. This is an example of the LCA model where evidence for one choice is much greater and the neurons representing the other choice are greatly inhibited. This makes the turn neurons highly selective for the correct choice.

On error trials the difference in activity between the two groups of neurons is not significant and the turn neurons are significantly less selective than during correct trials. This is a potential explanation for errors consistent with the LCA model as similar activity on preferred and unpreferred error trials indicates a much closer competition between the two choices and similar levels of evidence for each choice. When the choice is more difficult, according to the LCA model evidence is integrated more slowly with eventually one choice taking over and inhibiting the other. This is demonstrated firstly with error trials taking longer on average, and secondly by Figure 4.9. Figure 4.9 shows the light blue and pink lines representing preferred and unpreferred neurons, both increase their firing rate but while the light blue line continues to increase the pink slows and eventually drops during the choice turn indicating increased inhibition from the opposing turn selective neurons.

This difference in selectivity when comparing correct and error trials is however in contrast with the findings of Roitman and Shadlen (2002). Their experiment showed that in a decision making task, similar levels of activity were recorded during correct and error trials from neurons in the parietal cortex. This could be due to the different brain regions neurons were recorded from and analysis of the parietal cortex during the T-maze task would be required to confirm this.

5.4 - Evaluation

The first objective of this project was the creation of a software toolbox for the querying of position data. This was completed with the development of the MQL toolbox and therefore this objective can be considered a success as it met the requirements of enabling the querying of spatial position data and through that the analysis of neural recording data to answer research questions. Furthermore it allowed analyses not previously possible to be carried out, such as the analysis of the choice turn, analysis before the choice turn and analysis of all turns in the maze. The flexibility of the software in the way that the query system used query lines, allowed a wide variety of queries to be created quickly and efficiently. Specifying the coordinates of query lines gave precision allowing, for example the analysis of firing between intervals of a specific and consistent size along the central arm. The exporting of timestamps allowed many different types of custom analysis such as average firing rates from many pooled neurons. Using timestamps also allowed the concatenation of multiple results in order to compare firing rates during all turns of the maze. Choices were made during the development of MQL to ensure that it can be as broadly used as possible. Initially it was planned that there may be custom features due to the analysis being primarily of the T-maze data, however all choices such as the methods of interpolation, exporting of timestamps, loading of data were deliberately chosen in order to make them available for general use. The file format for example, initially required Neuralynx format position data, this was changed to a standard MATLAB .mat file allowing data from multiple sources. Likewise timestamps and validity of data are exported is in a way so as to enable a user to use the data in any way so as to best answer their specific research question. At the time of writing this software is currently being used by neuroscience researchers in the medical school at Bristol University for analysis of data from a new version of the T-maze experiment.

In addition to the success of the first objective of this project, the second objective: to use MQL to conduct analyses of neural data from rats in the T-maze task in order to answer research questions has also been completed. The questions posed by earlier research (Bogacz 2010) about why errors occur was investigated through analysis of the choice turn and gave additional insight into the cause of errors indicating that exploration may not be an explanation and showing timings consistent with the literature. The second question to verify predictions of the existing computational model (Gorochowski 2009) by investigating the existence of turn neurons also proved fruitful with 19 neurons shown to be selective for left or right turns. Analysing turn selective neurons leading up to the choice turn showed them to adhere to computation models and experimental results of decision making. The choice turn analysis also showed a significant difference between in the selectivity of these neurons during correct trials and error trials providing further information about the potential cause of errors. Some analyses were limited by lack of particular trials being recorded, however this was beyond the control of the project and where this occurred more general analyses were carried out meaning that no questions were unanswered. Given that all the research questions were able to be addressed to some level and that some (such as the discovery of turn selective neurons) led to further questions and analyses, the research objective of this project can be a considered to have been successful.

A key contributing factor to the success of this project was the splitting of the two objectives. In order to allow enough time for both to be completed one month was allocated to the development

of MQL and the remaining time to addressing research questions (with a small amount of time for changes to MQL where required). Stopping development after this time was crucial to allow sufficient attention to be paid to data analysis. Without this limit to development, the ability to address the further research questions that came from results would have been limited.

This project dealt with the decision making area of neuroscience, it showed how we can attempt to understand decision making and the brain in general through the analysis of experimental data and through using the predictions of computational models. It also showed how software tools can be invaluable in simplifying otherwise prohibitively complex analysis and that in the interdisciplinary field of neuroscience, computer scientists have an important role to play.

6 - Future work

The use of MQL has highlighted potential modifications that could be of benefit, likewise results of analyses posed further questions that could be answered in the future. This section lists future work in both these areas.

6.1 – Extension of MQL

The MQL software was sufficient to carry out the analyses described however extensions to the functionality of the software could increase its potential use.

Allowing the deletion of selected query or avoid lines would be of use in correcting mistakes or in amending existing queries where currently all query lines must be removed. The addition of the ability to create diagonal lines could be of benefit in creating queries for circular mazes. Creation of queries for circular mazes is currently possible however diagonal query lines could give additional precision to such a query.

6.2 - Further Analysis

In a few of the analyses carried out neurons had to be pooled together or data from certain rats ignored. This was a result of a lack of error trials recorded and prevented some questions being fully investigated. Carrying out these analyses again with data record from more rats over larger periods of time would be of interest so as to fully explore the questions. Similarly it would be interesting to carry out some of the analyses on data recorded from the parietal cortex during the T-maze task. This could help answer whether turn selective neurons exist in the parietal cortex and if so whether they exhibit similar selectivity on correct and error trials to that shown of turn selective neurons in the pfc. Recording from the prefrontal cortex while the T-maze task is being learnt by rats would be interesting to analysis as it would be possible to see if turn neurons do develop over time as hypothesised.

7 – References

Bogacz, R., Brown, E., Moehlis, J., Holmes, P., & Cohen, J. D. (2006). The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced choice tasks. *Psychological Review [0033-295X]*, *113*(4), 700.

Bogacz, R., Usher, M., Zhang, J., & McClelland, J. L. (2007). Extending a biologically inspired model of choice: multi-alternatives, nonlinearity and value-based multidimensional choice. *Philosophical Transactions of the Royal Society, B.*, *362*(1485), 1655-1670.

Bogacz, R. (2010). Why do the rats make errors in the Jones and Wilson (2005) task? Analysis of behavioural and spiking data. *Bristol University, Unpublished*.

Britten, K. H., Shadlen, M. N., Newsome, W. T., & Movshon, J. A. (1993). Responses of neurons in macaque MT to stochastic motion signals. *Visual Neuroscience*, *10*(6), 1157-1169.

Busemeyer, J. R., & Townsend, J. T. (1993). Decision field theory: A dynamic-cognitive approach to decision making in uncertain environment. *Psychological Review*, *100*(3), 432-459.

Buzsáki, G. (2002). Theta oscillations in the hippocampus. *Neuron*, *33*(3), 325-340.

Daw, N., O'Doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature, 441*, 876-879.

Gold, J. I., & Shadlen, M. N. (2004). The neurophysiology of decision-making as a window on cognition. *The Cognitive Neurosciences*, *Ed. M S Gazzaniga (Cambridge, MA: MIT Press)*, 1229-1241 .

Gorochowski, T. E. (2009). Cross-Frequency Coupling of Neuronal Oscillations During Cognition. *Bristol University, Unpublished Paper*.

Jones, M. W., & Wilson, M. A. (2005). Theta rhythms coordinate hippocampal-prefrontal interactions in a spatial memory task. *PLoS Biol*, *3*(12), e402.

Jung, M., Qin, Y., McNaughton, B., & Barnes, C. (1998). Firing characteristics of deep layer neurons in prefrontal cortex in rats performing spatial working memory tasks. *Cereb Cortex, 8*(5), 437-450.

Mazurek, M. E., Roitman, J. D., Ditterich, J., & Shadlen, M. N. (2003). A role for neural integrators in perceptual decision making. *Cerebral Cortex*, *13*, 1257-1269.

Meier, R., Boven, K., Aertsen, A., & Egert, U. (2007). FIND - Finding Information in Neuronal Data An open-source analysis toolbox for multiple-neuron recordings and network simulations. *Proc. 7th German Neurosci Meeting*, 1212.

Meier, R., Egert, U., Aertsen, A., & Nawrot, M. P. (2008). FIND - A unified framework for neural data analysis. *Neural Networks*, *21*(8), 1085-1093.

Neuralynx. (2010). *Software Downloads*. Retrieved May 7, 2010, from http://www.neuralynx.com/Software

NeuroDB. (2010). *NeuroDB the neurophysiology database*. Retrieved May 7, 2010, from http://neurodb.dertech.com/

Neuroshare. (2010). *neuroshare.org - Home Page*. Retrieved May 7, 2010, from http://neuroshare.sourceforge.net/index.shtml

Nitz, D. (2009). Parietal cortex, navigation, and the construction of arbitrary reference frames for spatial information. *Neurobiology of Learning and Memory, 91*(2), 179-185.

Öngür, D., & Price, J. L. (2000). The organization of networks within the orbital and medial prefrontal cortex of rats, monkeys and humans. *Cerebral Cortex*, *10*(3), 206-219.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *83*, 59-108.

Ratcliff, R., & Rouder, J. N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, *9*(5), 347-356.

Ratcliff, R. (2001). Putting noise into neurophysiological models of simple decision making. *Nature Neuroscience*, *3*, 336.

Roitman, J. D., & Shadlen, M. N. (2002). Response of Neurons in the Lateral Intraparietal Area during a Combined Visual Discrimination Reaction Time Task. *The Journal of Neuroscience*, *22*(21), 9475-9489.

Schall, J. D. (2001). Neural basis of deciding, choosing and acting.. *Nature Reviews Neuroscience*, *2*, 33-42.

Shadlen, M. N., & Newsome, W. T. (2001). Neural Basis of a Perceptual Decision in the Parietal Cortex (Area LIP) of the Rhesus Monkey. *The Journal of Neurophysiology, 86*(4), 1916-1936.

Shneiderman, B. (1998). *Designing the User Interface - Strategies for Effective Human-Computer Interaction* . Boston, MA, USA: Addison-Wesley.

Siapas, A., Lubenov, E., & Wilson, M. (2005). Prefrontal phase locking to hippocampal theta oscillations. *Neuron*, *46*(1), 141-151.

Sigtool. (2010). *Sigtool*. Retrieved May 7, 2010, from http://sigtool.sourceforge.net/

Stone, M. (1960). Models for choice reaction time. *Psychometrika*, *25*(3), 251-260.

Usher, M., & McClelland, J. L. (2001). On the time course of perceptual choice: the leaky competing accumulator model. *Psychological Review*, *108*, 550-592.

Vanderwolf, C. H. (1969). Hippocampal electrical activity and voluntary movement in the rat. *Electroencephalography and Clinical Neurophysiology*, *26*(4), 407-418.

Vickers, D. (1970). Evidence for an accumulator model of psychophysical discrimination. *Ergonomics*, *13*(1), 37-58.

Wald, A. (1947). *Sequential Analysis*. New York: Wiley.

Wallis, J. D., Anderson, K. C., & Miller, E. K. (2001). Single neurons in prefrontal cortex encode abstract rules. *Nature*, 411, 953-956.

Wang, X. (2002). Probabilistic decision making by slow reverberation in cortical circuits. *Neuron*, *36*(5), 955-968

Whitlock, J., Sutherland, R., Witter, M., Moser, M., & Moser, E. (2008). Navigating from hippocampus to parietal cortex. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(39), 14755-14762.

8 – Appendix: Source code

Interpolation Function

```
function [pos_interp] = interpolation (data, mazeBox, timeout, maxDist, boxI, distI, invalidD, invalidT)

% function [pos_interp] = interpolation (data, mazeBox, ...
%                      {timeout, maxDist, boxI, distI, invalidD, invalidT})
%
% Interpolates rat position data. Data is interpolated if either:
%       1 - A data point's position is outside of a defined box (mazeBox)
%       2 - [OPTIONAL] Two consequtive data points show movement greater
%           than a given distance (maxDist)
%
%
% Inputs:
%  data - behavioural data with positions in the form of a t-by-3 vector
%         where t is the number of timestamps. The 3 columns of the data
%         are: timestamps, x coordinates of the rat at timestamp, y
%         coordinate of the rat at timestamp.
%           [timestamps, x coordinates, y coordinates]
%  mazeBox - coordinates of box, values outside of which are interpolated
%           [x1,x2,y1,y2] i.e. mazeBox = [30, 300, 1, 260];
%
% Optional Inputs:
%  timeout - minimal number of timesteps by which continual interpolation
%            invalidD is not considered valid, data will be marked as
%            invalid.
%            Default: 1000000 in order to disable timeout for signal loss
%            less 1000000 timesteps
%  maxDist - maximum valid distance travelled in 1 timestep, longer
%            distances are interpolated, if they are still longer after
%            interpolation they are marked as invalid.
%            Default : 1000000 - unless mazes are very large no
%            interpolation based on distance moved will be carried out
%  boxI - if 1 show points interpolated using mazeBox criteria, default: 1
%  distI - if 1 show points interpolated using maxDist criteria default: 1
%  invalidD - if 1 show points invalid due to distance criteria default: 1
%  invalidT - if 1 show points invalid due to timeout criteria default: 1
%
% Outputs:
%  pos_interp - data with interpolated missing positions
%
%  contents of pos_interp:
%  pos_interp (:,1) - time stamp
%  pos_interp (:,2) - x position of electrode
%  pos_interp (:,3) - y position of electrode
%  pos_interp (:,4) - v validity of datapoint
%
%  v - Vector showing if data is valid(1) or invalid(0), validiy is defined
%      by either loss of signal for greater than a given number of
%      timesteps (timeout) or data points still showing movement of greater
%      than maxDist after interpolation

%set up variables
t = data (:,1);     %time stamp
x = data (:,2);     %position of head electrode
y = data (:,3);
N = length (t);     %number of time points
invalid = zeros(N, 1); %timeout points
invalid2 = zeros(N,1); %distance points
invalid3 = zeros(N,1); %timeout from distance points
pos_was_interped = zeros(N,1);
v = ones(N,1); %overall validity

if nargin < 3
    timeout = 1000000;
end

if nargin < 4
    maxDist = 1000000;
end

if nargin < 5
    boxI = 1;
end

if nargin < 6
    distI = 1;
end

if nargin < 7
    invalidD = 1;
end
if nargin < 8
    invalidT = 1;
```

```
end


figure
hold on

%check to see if in box, if not interpolate, if signal lost for too long
%not valid interpolation.
if isvalid(x(1), y(1), mazeBox)
     ok = find(x > mazeBox(1) & x < mazeBox(2) & y > mazeBox(3) & y < mazeBox(4));
     first = ok(1);
     x(1:first-1) = x(first);%all points up to first non zero = first non zero position
     y(1:first-1) = y(first);
     pos_was_interped(1:first-1) = 1;
     first_interp = 2;
end

for i=2:N
   pos_was_interped(i) = isvalid(x(i), y(i), mazeBox);

   %first pos in run of interp points, prev pos valid
   if pos_was_interped(i) && ~pos_was_interped(i-1)
       first_interp = i;
   %end of run of interp point, prev pos invalid
   elseif ~pos_was_interped(i) && pos_was_interped(i-1)
       if (i - first_interp + 2) > timeout
           invalid(first_interp + 1:i-1) = 1;
           v(first_interp + 1:i-1) = 0;
       end
       for j = first_interp:i-1
           %current pos in relation to run
           diff = (j - (first_interp-1))/(i- (first_interp-1));
           x(j) = x(i)*diff + x(first_interp-1)*(1-diff);
           y(j) = y(i)*diff + y(first_interp-1)*(1-diff);
       end
   end
end

%calculate invalid distance
invalid(1) = 0;
v(1) = 1;
for i=2:N-1

   %find distance between current point and last good point if greater than
   %threshold interp not valid
   %check distance
   invalid2(i) = check_dist2(x(i-1), x(i), x(i+1), y(i-1), y(i), y(i+1), maxDist);

   %rules:
   %1 - if two points are close together they are both valid
   %2 - if after interp, points still far apart, mark invalid
   %3 - if a run of invalid position points, interp all

   %current point invalid, previous point valid
   if invalid2(i) && ~invalid2(i-1)
       first_dist = i;
   %previous point invalid, current point valid
   elseif ~invalid2(i) && invalid2(i-1)
       if (i - first_dist + 2) > timeout
           invalid3(first_dist + 1:i-1) = 1;
           v(first_dist + 1:i-1) = 0;
       end
       for j = first_dist:i-1
           %current pos in relation to run
           diff = (j - (first_dist-1))/(i- (first_dist-1));
           x(j) = x(i)*diff + x(first_dist-1)*(1-diff);
           y(j) = y(i)*diff + y(first_dist-1)*(1-diff);
           %check to see if still invalid
           if check_dist (x(j), x(j-1), maxDist)
               invalid3(j) = 1;
               v(j) = 0;
           end
       end
   end

end

%v = valid interpolation
pos_interp = [t, x, y, v];

%show result of interpolation
 plot(x, y, 'Color', [.7 .7 .7]);

 leg = {'Valid Position Data'};
 if boxI
    interpedints = find(pos_was_interped);
    if ~isempty(interpedints)
```

```
        plot(x(interpedints), y(interpedints), '.c');
        leg = [leg,'Valid Box Interpolation'];
    end
end

if invalidT
    invalidints = find(invalid);
    if ~isempty(invalidints)
        plot(x(invalidints), y(invalidints), '.r');
        leg = [leg,'Invalid due to Signal Loss'];
    end
end

if distI
    invalidposints = find(invalid2);
    if ~isempty(invalidposints)
        plot(x(invalidposints), y(invalidposints), '.g');
        leg = [leg,'Valid Distance Interpolation'];
    end
end

if invalidD
    invalidposints2 = find(invalid3);
    if ~isempty(invalidposints2)
        plot(x(invalidposints2), y(invalidposints2), '.m');
        leg = [leg,'Invalid After Distance Interpolation'];
    end
end

legend(leg);

%%FUNCTIONS%%

%%%%returns true (1) if valid is invalid, 0 if valid, hence return is
%%%%called interp_needed
%%params%%
function interp_needed = isvalid(xpos, ypos, mazeBox)

%if outside x and y coords of box
if xpos < mazeBox(1) || xpos > mazeBox(2) || ypos < mazeBox(3) || ypos > mazeBox(4)
    interp_needed = 1;
else
    interp_needed = 0;
end

%%%%checks for 2 consequtive invalid moves%%%%
%%params%%
function invalid2moves = check_dist2(xp, xc, xn, yp, yc, yn, dist)
if check_dist([xp, yp],[xc, yc], dist) && check_dist([xc, yc], [xn, yn], dist)
    invalid2moves = 1;
else
    invalid2moves = 0;
end

%%%%check if distance between coords is greater than threshold%%%%
%%params%%
function invalid_move = check_dist(p1, p2, maxDist)
invalid_move = 0;

dist = pdist([p1; p2]);
if dist > maxDist
    invalid_move = 1;
end
```

Query function

```
function [timestamps, valid] = run_query(interp_data, query, avoid)

% function [startpoints, valid, endpoints, timestamps] =...
%                                   query (interp_data, query, {avoid})
%
% Queries rat position data using lines that must be crossed in order by
% positional data. Avoid queries can be optionally used, they discount runs
% which cross them.
%
% Inputs:
%  interp_data - behavioural data with positions (timestamps,x,y)
%  query - coordinates of query, querys must be either horizontal or
%          vertical lines of the form [x1,x2,y1,y2;x1,x2,y1,y2] at least 2
%          query lines must be specified.
%            i.e. query = [25,75,90,90;220,220,90,140;270,300,90,90];
% Optional Input:
%  avoid - coordinates of lines used to discount runs that intersect them,
%          defined in the same manner as query lines, though there is no
%          minimal limit
```

```
%           i.e. avoid = []
%                avoid = [25,75,160,160]
%          Default = []
%
% Outputs:
%  timestamps - timestamps of point on or before intersection for each
%               query i.e. 5 query lines will return 5 columns of
%               timestamps
%  valid - for each timestamp 1 if both of the points crossing the query
%          line are valid, 0 if either are invalid.

t = interp_data (:,1);     %time stamp
x = interp_data (:,2);     %position of head electrode
y = interp_data (:,3);
v = interp_data (:,4);     %validity of data point
N = length (t);     %number of time points
timestamps = []; %timestamps at points of intersections
startpoints = [];
endpoints = [];
runTimeStamps = []; %hold timestamps of intersections during querying
runPosition = [];
valid = [];
runvalid = [];

if nargin == 1
    avoid = [];
end
%plot background maze
plot(x,y, 'Color', [.7 .7 .7])

%loop through and set up queries
maxQ = size(query);
for queryCount = 1: maxQ(1)
     queryp1x(queryCount) = query(queryCount, 1);
     queryp1y(queryCount) = query(queryCount, 3);
     queryp2x(queryCount) = query(queryCount, 2);
     queryp2y(queryCount) = query(queryCount, 4);
     %direction(queryCount) = query(queryCount, 5);
    plot ([queryp1x(queryCount), queryp2x(queryCount)], [queryp1y(queryCount), queryp2y(queryCount)], 'g');
end

maxA = size(avoid);
for queryCount = 1: maxA(1)
     avoidp1x(queryCount) = avoid(queryCount, 1);
     avoidp1y(queryCount) = avoid(queryCount, 3);
     avoidp2x(queryCount) = avoid(queryCount, 2);
     avoidp2y(queryCount) = avoid(queryCount, 4);
     %direction(queryCount) = query(queryCount, 5);
    plot ([avoidp1x(queryCount), avoidp2x(queryCount)], [avoidp1y(queryCount), avoidp2y(queryCount)], 'r');
end

activeQuery = 1;
for i = 1:N-1

    %invalid point reset query - depricated in version 1.4
    %if v(i)
        %activeQuery = 1;
    %end
    if checkavoid(x(i), x(i+1), y(i), y(i+1), avoid)
        activeQuery = 1;
    elseif intersection(x(i), x(i+1), y(i), y(i+1), query, 1)
        %check doesn't meet first query, if does reset query count
        activeQuery = 2;
        startOfRun = i;
        runTimeStamps = [t(i)];
        runPosition = i;
        if v(i) || v(i+1)
            runvalid = [1];
        else
            runvalid = [0];
        end
    elseif intersection(x(i), x(i+1), y(i), y(i+1), query, activeQuery)
        %check meets current query

        %check if points that meet query are valid
        if v(i) || v(i+1)
            runvalid = [runvalid, 1];
        else
            runvalid = [runvalid, 0];
        end
        runTimeStamps = [runTimeStamps, t(i)];
        runPosition = [runPosition,i];

        %if current query is final query record run and reset query
        if activeQuery == maxQ(1)
            activeQuery = 1;
            startpoints = [startpoints, t(startOfRun)];
            endpoints = [endpoints, t(i)];
```

```
            timestamps = [timestamps; runTimeStamps];
            valid = [valid; runvalid];
            plotpoints(runPosition, x, y, v);
            runTimestamps = [];
            runPosition = [];
        else
            if activeQuery == 1
                startOfRun = i;
            end
            activeQuery = activeQuery + 1;
        end
    end

end

%%FUCNTIONS%%

function avoidmet = checkavoid(xc, xn, yc, yn, avoid)
avoidmet = 0;
for i = 1:size(avoid,1)
    if intersection(xc, xn, yc, yn, avoid, i)
        avoidmet = 1;
    end
end

function querymet = intersection(xc, xn, yc, yn, query, queryNo)
%query indexes
x1 = 1;
x2 = 2;
y1 = 3;
y2 = 4;

querymet = 0;
%horizonal
if query(queryNo, y1) == query(queryNo, y2)
    %xcoord in query range
    if query(queryNo, x1) <= xc && xc <= query(queryNo, x2)
    if query(queryNo, x1) <= xn && xn <= query(queryNo, x2)
    %y coords go through y pos of query
    if (yc <= query(queryNo, y2) && query(queryNo, y2) <= yn) || ...
            yc >= query(queryNo, y2) && query(queryNo, y2) >= yn
              querymet = 1;
    end
    end
    end
elseif query(queryNo, x1) == query(queryNo, x2)
%vertical

    %xcoord in query range
    if query(queryNo, y1) <= yc && yc <= query(queryNo, y2)
    if query(queryNo, y1) <= yn && yn <= query(queryNo, y2)
    %y coords go through y pos of query
    if (xc <= query(queryNo, x2) && query(queryNo, x2) <= xn) || ...
            xc >= query(queryNo, x2) && query(queryNo, x2) >= xn
              querymet = 1;
    end
    end
    end
end

function plotpoints(runpoints, x, y, v)
numPoints = size(runpoints);
for i = 1:numPoints(2)
    if v(runpoints(i)) || v(runpoints(i) + 1)
        plot(x(runpoints(i)), y(runpoints(i)), 'g.');
    else
        plot(x(runpoints(i)), y(runpoints(i)), 'r.');
    end
end
```