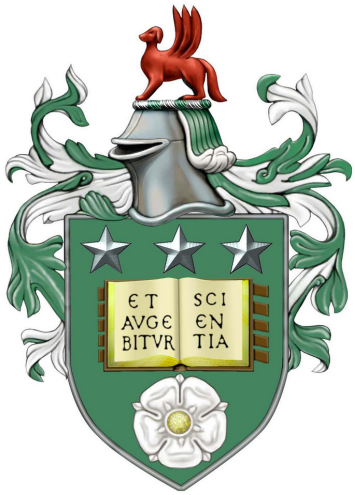
Modelling Decision Making under Uncertainty

Machine Learning and Neural Population Techniques

Elaine Duffin



Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Computing

June 2015

The candidate confirms that the work submitted is his/her own, except where work which has formed part of a jointly authored publication has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Some of the work presented in Chapters 2 and 3 has appeared in publication as follows:

Duffin E, Bland AR, Schaefer A and de Kamps M Differential effects of reward and punishment in decision making under uncertainty: a computational study. Frontiers in Neuroscience. 8:30. 2014.

I carried out the design, implementation and evaluation of all the computational work described in the paper, including creating all the figures. I wrote the majority of the content of the paper. Marc de Kamps provided supervisory advice. Amy Bland and Alexandre Schaefer carried out the original psychological study on which my computational research was based. They made suggestions to help clarify the text for submission and during the review process.

Acknowledgements

Firstly, I would like to thank my supervisor, Marc de Kamps for his support during my PhD. Marc has provided useful guidance and encouragement and has helped me to build confidence in my research. My co-supervisor, Netta Cohen, gave helpful advice, especially in the early stages of my PhD, for which I extend my thanks.

Psychologists who were previously at the University of Leeds, Alexandre Schaefer and Amy Bland, sparked my initial interest in the vast field of human decision making. I thank them for this and for useful discussions.

Many thanks go to Jeanette Hannah for the extremely helpful discussions. Jeanette has helped me to see things in a wider context and to build on my problem solving techniques to help me cope with the demands of PhD research. She helped me to believe that I would be able to succeed. Without her I would not have completed my first year and this thesis would not have existed.

I would like to thank the EPSRC for funding my research through the University of Leeds. I would also like to thank the School of Computing for the opportunity to study here and especially Brandon Bennett for the support he has provided in his role as postgraduate research tutor for the school. I would also like to thank the school for the opportunities to assist with taught courses.

I would like to thank the researchers from room 9.27 who have provided a welcoming and supportive environment where, although we do not share research topics, we have had valuable discussions about other aspects like scientific writing. A special note of thanks goes to Sam for friendship, many discussions and a very careful reading of a draft of this thesis.

Many thanks to my partner, Neal, who has been very supportive during my studies and has also proofread this thesis.

Abstract

This thesis investigates mechanisms of human decision making, building on the fields of psychology and computational neuroscience. I focus on human decision making measured in a psychological task with probabilistic rewards.

I examine the fit of different styles of computational models to human behaviour in the task. I show that my modification to reinforcement learning, using parameters based on whether the previous trial resulted in a win or a loss, is a better fit to behaviour than my Bayesian models. Considering the task from a machine learning perspective, with the goal of gaining as many rewards as possible rather than modelling human behaviour, the performance of my modified reinforcement learning model is similar to that of my Bayesian learner and superior to that of a standard reinforcement learning model.

Using population density techniques to simulate neural interactions, I confirm earlier research that demonstrates conditions which induce oscillations in a system consisting of just two nodes. I extend those findings by showing how the underlying states of the neurons contribute to complex patterns of activity.

The basal ganglia form part of the brain known to be important in decision making. I create a computational model of the basal ganglia to simulate decision making. As oscillatory neural activity is known to occur in the basal ganglia, I add such activity to the model and study its impact on the decisions made. I use the time that activation first falls below a threshold as a criterion for decision making. This alternative approach allows oscillatory activity to have advantages for decision processes.

Having tested my basal ganglia model on individual decisions, I extend the model to incorporate parameters related to my modified reinforcement learning model. I propose a mechanism by which the trial to trial variability observed in human responses could be implemented neurally.

Contents

List of Figures

List of Tables

Abbreviations

BIC Bayesian Information Criterion.
FV feedback validity.
GPe globus pallidus external.
GPi globus pallidus internal.
HMM hidden Markov model.
RL reinforcement learning.
SNc substantia nigra pars compacta.
SNr substantia nigra pars reticulata.
STN subthalamic nucleus.
UNC uncoupled reinforcement learning.
VOL hidden Markov model with volatility.
WL win loss modified reinforcement learning.

Chapter 1

Introduction

1.1 What is a decision?

In order to illustrate the type of decision making I investigate in this thesis, I will start with an example. Consider buying a coffee on the way to work. Suppose there are two coffee shops but you do not know anything about their reputations. How do you decide which one to go to? You could develop an expectation of the quality of the coffee in each coffee shop by trying each one several times.

Even when always frequenting a single coffee shop, there is variation in the quality of the coffee from day to day. After a number of days sampling coffee, you might believe that one coffee shop serves good coffee on 85% of occasions and the other on 70% of occasions, but the actual pattern of good and bad days at each shop is random.

After you have estimated the chance of a good coffee in each shop, you still have to decide which coffee shop to use. Do you stick with always going to the better one, or do you sometimes try the other one to see if things have changed?

Suppose, for example, one of the coffee shops introduces a new coffee machine which affects the quality of the coffee. If you have no indication that there has been a change to the machine, but you get a bad coffee at one coffee shop for a few days, how do you decide whether you should try the other to see if it is now better?

This scenario highlights aspects of the decisions I am interested in. There is a choice of actions, and the action taken can be interpreted as representing the decision made. In the coffee shop example, assuming that you only buy one coffee each morning, the decisions you make can be seen from the actions you take of buying coffee in one of the

two shops each day. I assume that the decision made is not completely random and that there is an underlying motivation to try to make the best decision from the perspective of the individual involved.

Another aspect of the decisions I consider is that there is nobody to teach you what action is best. In the example, you learn from the actual experience of drinking the coffee over a number of days and base your future decisions on those experiences. Learning from experience without direct instruction is called *reinforcement learning*.

Humans are able to learn what rewards to expect from different actions and are also flexible and able to respond to changes in the environment. In the coffee shop example, the change in the environment was an unobserved change to the coffee machine. This type of change to the environment has been described as *unexpected uncertainty* (Yu & Dayan, 2005). As with the original learning, the only way to learn of this change was to taste the coffee over a number of days.

Suppose you have discovered which coffee shop is generally the better of the two and do not experience any overall change in the quality of the coffee. If you decide to always go to the better coffee shop then you would not know whether an improvement had been made to the other coffee shop such that you would now change your opinion as to which is best. Sticking with what you already know is called *exploitation* (Cohen et al., 2007). If on the other hand, you sometimes try the different options to see whether things have changed, this is called *exploration* (Cohen et al., 2007).

In the coffee shop example, the quality of the coffee varies from day to day, but with no predictable pattern to this variation. After time you come to expect random differences in your experience. This probabilistic variation is a natural factor of the environment, there is no action you can take to learn more and remove the variation, this is known as *expected uncertainty* (Payzan-LeNestour & Bossaerts, 2011; Yu & Dayan, 2005).

Situations in which the underlying rules, or regularities in the environment, change unpredictably at different intervals are often called *volatile* (e.g. Behrens et al., 2007; Bland & Schaefer, 2012). The interaction of different forms of uncertainty poses a challenge for successful learning, and recent research has tried to understand how humans adapt to such environments (e.g. Behrens et al., 2007; Hampton et al., 2006).

1.2 Motivation

The decision making in the coffee shop example includes the types of processes that I am interested in, but the real world is highly complex with many different stimuli and possible actions which cannot always be expressed as sequences of separate decisions. Psychological studies of learning and decision making can be carried out in controlled environments, where simple identifiable stimuli can be presented and actions limited to a choice of buttons. Reducing the complexity in this way allows us to consider how different factors influence the decisions people make.

In this thesis, I compare how well different styles of computational model are able to describe human learning under expected uncertainty in a volatile environment as tested by Bland & Schaefer (2011). In the study of Bland & Schaefer (2011), on multiple trials participants pressed one of two buttons in response to a visual stimulus of a triangle presented in one of two colours, red and blue. Participants won and lost points based on their responses but the wins and losses were generated probabilistically based on underlying rules, creating expected uncertainty. Frequent unsignalled switches in the underlying rules created a volatile environment which was contrasted with stable periods with no such switches.

Participants in the task of Bland & Schaefer (2011) were asked to try to win as many points as possible and told that their accumulated points would be converted into a cash payout. From this, I assume that they were motivated to try to maximise the number of points earned and not just press the buttons randomly. The two colours were easily distinguished by the participants who took part and there was a clear decision to be made on each trial as to which of the two buttons to press. The participants were not given instructions as to how to gain points and had to learn from on-screen feedback.

Although the task, as described, seems relatively simple, it is not known what underlying processes are used when people make decisions in these situations. Having clear stimuli and outcomes and only two available actions makes it possible to computationally model the decision making process and to compare observed behaviour to the models.

When looking for models with which to compare human behaviour, it is useful to look at the field of machine learning. Machine learning has the goal of trying to find the optimal solution to problems. If human decisions match those of machine learning

procedures, this would imply that humans are able to behave optimally. This would be important in terms of the ability of evolutionary pressures to direct learning.

Reinforcement learning is a branch of machine learning which is focussed on learning through interaction with the environment and adjusting behaviour according to whether the outcomes of those interactions were favourable or not (Sutton & Barto, 1998). Reinforcement learning does not require any prior knowledge about what actions are good and bad, learning starts with trial and error and can build from there. This learning can allow an agent to favour different actions in different environmental situations.

Bayesian learning represents elements of the environment by probability distributions. Probabilities of the occurrence of different events are computed using relations between different variables in the system and applying probability theory. In the Bayesian learning considered in this thesis, the relations between variables have to be known in advance and using these relations an accurate probability can be calculated for each possible outcome and with these accurate probabilities, optimal decisions can be made.

There has been a recent focus on which style of modelling, reinforcement learning or Bayesian reasoning, is a better approximation of human behaviour in various tasks involving decisions based on probabilistic feedback (e.g. Behrens et al., 2007; Hampton et al., 2006; Payzan-LeNestour & Bossaerts, 2011; Wilson & Niv, 2012). This motivated me to investigate the fit of various computational models to the task of Bland & Schaefer (2011).

Using machine learning models to investigate human behaviour can give us an idea of the rules which underlie behaviour and allow us to spot patterns and inconsistencies in behaviour, but this approach does not give any indication as to how the learning is implemented by neural processes. The basal ganglia, groups of neurons near the centre of the brain, have long been known to be important in decision making (see e.g. Lanciego et al., 2012). Although the basal ganglia form only a small part of the whole brain, there are many interconnections within the basal ganglia, leading to many ideas as to how the basal ganglia support learning and decision making.

One way to investigate proposed models of processing in the basal ganglia is to produce computational models of interacting neural populations. Computational modelling allows selected neural populations to be connected together with different strengths to look at the influence of isolating and changing parts of the basal ganglia circuitry. Such models allow speculation of the functions of different parts of the basal ganglia and

can also allow parameters to be changed to simulate medical conditions. Parkinson's disease is such a condition in which the patterns of neural output in the basal ganglia are thought to play a big role (Lanciego et al., 2012).

Existing neuro-computational models of psychological tasks similar to that of Bland & Schaefer (2011) motivated me to investigate how interactions between neural populations can produce the range of behaviour measured in that task.

1.3 Thesis outline and contributions

In Chapter 2, I give details of the psychological task of Bland & Schaefer (2011) and examine the behaviour of the participants in that task. I relate that task to previous studies in terms of the behaviour observed and computational models of that behaviour.

I have used computational models to describe the human behaviour recorded by Bland & Schaefer (2011), these models are detailed in Chapter 3. Comparing the fit of different models to human behaviour, I found that a reinforcement learning model which had been parameterised such that wins and losses had different impacts on future decisions was a better fit to the human behaviour observed by Bland & Schaefer (2011) than Bayesian style models, even when I allowed action probabilities to differ after wins and losses in the Bayesian models. This contributed to the understanding of the situations in which human learning is better approximated by reinforcement learning models than Bayesian models. Finding, in addition, that my amended reinforcement learning model, when implemented as a machine learning algorithm rather than to match human behaviour, was actually better at the psychological task than my Bayesian models gave a contribution to the understanding of the limitations of techniques for decision making under uncertainty. Much of the work in Chapter 3 has been published as Duffin et al. (2014).

Due to the importance of the basal ganglia in the process of decision making, in Chapter 4, I give an introduction to the biology of the basal ganglia and ideas of how the connected areas of the basal ganglia contribute to information processing. I briefly describe learning in the basal ganglia and the influence of the basal ganglia on Parkinson's disease. I review existing computational models of the basal ganglia focussing on studies which examine potential neural explanations of the range of behaviour exhibited by humans undertaking a single task.

In Chapter 5, I describe the technique of population density modelling of neural systems, a technique I apply in the remainder of this thesis. I describe how I create population density models using the simulator Miind (de Kamps et al., 2008). Population density modelling has much theoretical backing but has not been widely used to model cognitive tasks such as that I consider here. I provide a brief introduction to population density modelling, and then show how simple systems of connected populations of neurons can interact. Firstly, I show that population density modelling leads to the same conclusions as Nevado Holgado et al. (2010) for two connected neural populations. I show more complex patterns of interaction between the neural populations than those of Nevado Holgado et al. (2010) and I show how population density modelling gives a unique insight into how the underlying states of the neurons results in complex output.

I built a computational model of the basal ganglia for decision making using the population density techniques described in Chapter 5. The basal ganglia model built is described in Chapter 6, where I start by showing that a simplified basal ganglia model can correctly make simple decisions where the decisions to be made are motivated by the psychological task described in Chapters 2 and 3. I examine changes in decision making of this model as I simulate increasingly more difficult decisions. Parts of the basal ganglia are known to produce oscillatory activity and I extend the simplified neural system to include such activity. Investigating how decision making is affected by oscillatory input to the system, I give a novel description of the potential benefits of such oscillation.

The basal ganglia model examined in Chapter 6 does not include learning, so in Chapter 7, I build on that model to implement a decision making model with learning. I implement learning in an abstract, not biologically realistic, way and investigate a potential mechanism for trial to trial variation in responses as observed in human behaviour. To do this, I model multiple trials with simulated stimuli and feedback and use the activation of the model to determine the response made from which I calculate changes to be applied for the next trial. This gives simulated behavioural data which I can treat in the same way as in Chapter 3 for the participant data from the psychological task. I suggest that the model with oscillatory activity can produce decisions which vary in line with a theoretical model of action selection. This is a novel contribution as it links a plausible neural mechanism to a theoretical model.

In Chapter 8, I bring together the different styles of modelling presented in this thesis and highlight the contributions made. I consider how the work could be extended in future to overcome some of the limitations which are discussed in previous chapters.

Chapter 2

Behavioural Task

2.1 Introduction

Bland & Schaefer (2011) carried out an investigation into human decision making under uncertainty. Their task, which may seem at first glance to be quite simple, forms the focus of the computational modelling in this thesis. In the task, participants learn from sequential trials of pressing a button in response to a stimulus. Participants win and lose points for correct and incorrect responses respectively. These rewards and punishments through points gained and lost are the only way the participants can learn which response they can expect to be correct, they are not given any direct instruction as to how to respond. Underlying rules, which are not given to the participants, probabilistically determine which responses are correct, giving a situation of uncertainty. Bland & Schaefer (2011) also had a structure of changes to the underlying rules, giving volatility to the environment.

In this chapter, I describe the study of Bland & Schaefer (2011) in more detail and present the behavioural findings from that task. I relate the behaviour observed by Bland & Schaefer (2011) to that of other psychological studies into learning from probabilistic stimuli.

I review computational models to describe behaviour in similar tasks. I describe studies which have considered the ability of alternative styles of learning to match human behaviour. I also examine differences in response to winning and losing. These studies form the background to the computational modelling presented in Chapter 3.

2.2 Details of the task

The study of Bland & Schaefer (2011) involved thirty-one participants (18 female), with a mean age of 24. The study was approved by the local Ethics committee and participants gave written informed consent.

Participants were told that they started with 1000 points and would win or lose points according to each of their responses. On each of 960 trials, participants were shown a red or a blue triangle and had to respond by pressing one of two buttons, described here as button 1 and button 2. They were instructed, "your key press should be a guess about which is the right answer in response to the triangle. You will learn which is the correct answer". After each trial, participants were given immediate on-screen feedback as to whether they were correct and had won 10 points, were wrong and lost 10 points or were too slow to respond (over 1500 ms) and also lost 10 points. Following the feedback, participants were shown a black cross on a white background for a random duration between 1000 and 1500 ms, this formed a separation between the individual trials. Participants were asked to try to win as many points as possible and told that their final points total would be converted to a monetary reward. They were not told about underlying rules regarding rewards and were not given a running total of the points they had accumulated.

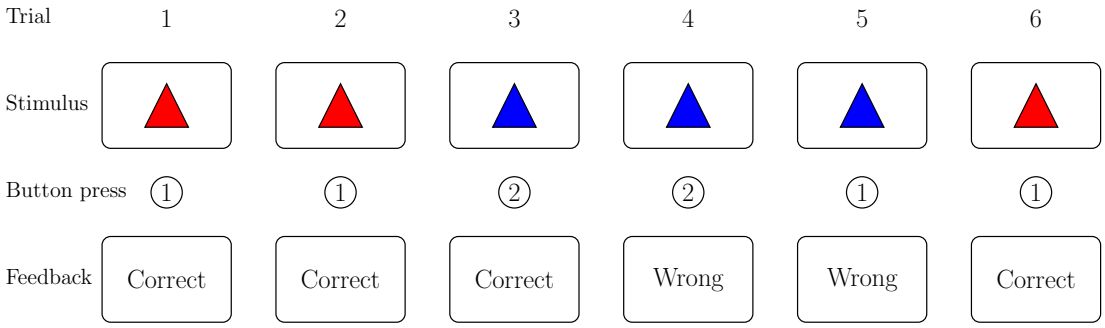


Figure 2.1: Illustration of six trials of the task for a hypothetical participant.

Figure 2.1 illustrates the task by showing a sequence of six trials with the button press and feedback for a hypothetical participant. Each trial consists of a stimulus, a response in terms of a button press and feedback as to whether the response was

correct or not. On the first trial in Figure 2.1, a red triangle is shown, the participant presses button 1 and is told that the response was correct. If this is the very first trial a participant takes, the response is likely to be made by guessing. On the second trial, red is shown again and the participant repeats the behaviour which previously gained a reward, presses button 1 and is again rewarded. On the third trial in Figure 2.1, blue is shown and the hypothetical participant remembering that button 1 was rewarded when red was shown supposes that the opposite response should be made when blue is shown and so presses button 2 and again is rewarded. On the fourth trial, even though the participant has carried out the action which was previously rewarded, pressing button 2 in response to the blue stimulus, the feedback given indicates that the response was wrong. This is because the feedback is based in a probabilistic way on the underlying environment. At this point the participant does not know whether this feedback is due to randomness in the environment or they have misunderstood the situation.

To analyse the behaviour of participants in the task, I encoded each response according to underlying behavioural types, where type 1 behaviour was to press button 1 when a red triangle was shown, and button 2 for blue; type 2 was the opposite of type 1. Using these response types the behaviour can be described without reference to the actual colour shown on each trial, which was randomised by Bland & Schaefer (2011). In terms of these response types, the example shown in Figure 2.1 starts with button 1 pressed in response to a red stimulus, and so this is a type 1 response. The full sequence of response types for the six trials in Figure 2.1 would be 1, 1, 1, 1, 2, 1.

As the environment consisted of two colours of stimuli and two buttons for responses, I assumed that participants knew that the red and blue triangles required opposite button presses. This means that, if feedback shows that one response type is incorrect, then the other response type would have been correct on that trial and vice versa, so regardless of which response is made, feedback lets you know how each response type would have fared. I incorporated this assumption in the above explanation of the responses given in the example shown in Figure 2.1.

The environment could be considered to have a current underlying rule which was manipulated by the experimenters. Rule 1 meant that responses of type 1 were rewarded on the majority of trials. The actual reward on an individual trial was based probabilistically on the underlying rule, giving uncertainty in the environment. Responses were rewarded at two different probabilities or levels of feedback validity (FV) which

remained constant throughout blocks of 120 trials. In high FV blocks, responses in line with the current rule were rewarded on 83.3% of trials, and in low FV blocks this was 73.3% with the actual outcome on individual trials randomised to meet these percentages. Two different numbers of unsignalled rule switches were used, in stable blocks, the environmental rule was constant for all 120 trials. In volatile blocks, the rule switched every 30 trials. Having two rules, two levels of FV and two levels of volatility gave eight conditions which were presented in blocks of 120 trials. All participants experienced all eight conditions but in a randomised order but were not given any indication of changes in experimental condition, having just one break after four blocks (480 trials).

2.3 Behaviour measured

Figure 2.2 illustrates the responses made and feedback given to four individual participants for the first 120 trials. This behaviour is shown in terms of the two response types described above which ignores the actual colour shown to the participant on each trial.

In Figure 2.2, responses which gained a reward of 10 points are shown by green diamonds and those which lost points as red circles. Participants were most likely to switch from one response type to the other after negative feedback, that is a loss of points. We can see this in Figure 2.2 by observing that after a red mark, the next response is often of the opposite type. In Figure 2.2, we can see from the shading that two of the participants shown (the first and third row) started with a stable block of trials, that is no switches in the underlying rule during the 120 trials. For the other two participants shown, the experiment began with volatile blocks having changes in the underlying rule every 30 trials. Although the shading shows which rule was being rewarded with a higher probability, it does not distinguish between the two different levels of probability, or feedback validity, used.

If the underlying rule can be identified from the pattern of rewards, but the result on individual trials cannot be predicted due to the randomness in the environment, then to gain the most rewards possible, one should always respond according to the underlying rule, this is often called *maximising* (see e.g. Yu & Huang, 2014). If one knows that type 1 responses are rewarded mostly, then one should make a type 1 response every trial and ignore occasional losses. In this case the environment could be described as

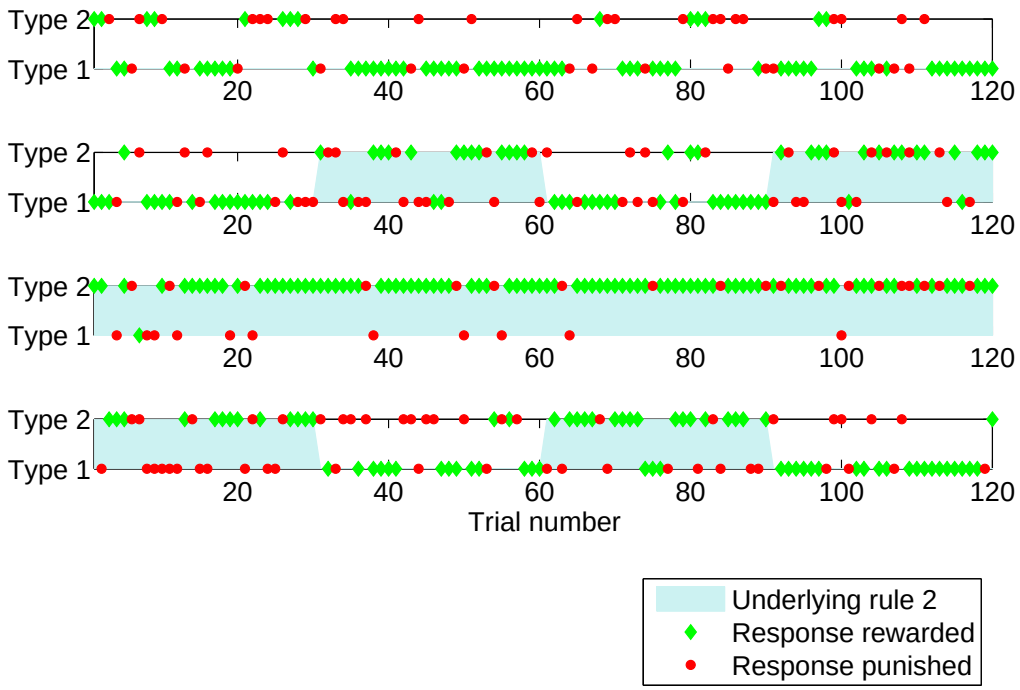


Figure 2.2: Responses made by four individual participants at the beginning of the study by Bland & Schaefer (2011). Type 1 responses are a press of button 1 following a red stimulus and button 2 for blue, with type 2 the opposite. Unshaded areas show that the underlying rule is rule 1, that is that type 1 responses are rewarded with high probability. Shaded areas show that rule 2 applies. Responses which gain (lose) points are shown in green (red).

having expected uncertainty, no more learning about the environment can help you to improve. You cannot avoid the randomness in the environment and so some losses are inevitable. This does not consider how to identify which response is being rewarded most, or how to identify a rule switch.

For each participant, I determined the level of maximising behaviour by calculating the percentage of trials in which the response was of the type which was associated with the underlying experimental rule. Individual differences in responding to the task gave a range of maximising behaviour from 62% to 89% (mean 74.5%, s.d. 6%).

For this and the computational analysis described in Chapter 3, I have excluded one participant whose behavioural performance showed maximising at less than 55% although that participant was not excluded in Bland & Schaefer (2011).

2.4 Research into similar tasks

2.4.1 Behavioural findings

In this thesis, I focus on the task of Bland & Schaefer (2011) which required participants to learn from probabilistic rewards. A related and widely-used task for learning in probabilistic environment asks participants to predict which of two colours will be displayed on each of many successive trials (see e.g. Siegel & Goldstein, 1959; Vulkan, 2000). One colour appears in a higher proportion of trials than the other but the actual colour shown is determined randomly, a probabilistic situation with expected uncertainty. The participants are not told which colour will appear more frequently or with what probability, they have to learn this information over time from the sequence of colours seen. As these studies have a clear outcome on each trial of observing one colour or the other, then it is clear that the two colours are coupled, as in the toss of a biased coin. There is no possibility of neither or both colours being shown in a trial. The colour prediction task differs from the task of Bland & Schaefer (2011) I consider here, which has the additional layer of a stimulus response association, the pairing of a colour to a button. When the task of Bland & Schaefer (2011) is considered as consisting of two possible response types, as described in Section 2.2 above, it becomes a coupled task and equivalent to the prediction of which colour will show next as exactly one of the two response types is rewarded on each trial.

Although the optimal behaviour in these colour prediction studies is always to predict the most likely outcome, it is commonly observed that participants' predictions reflect the probability of occurrence of each colour, behaviour known as *probability matching* (see e.g. Shanks et al., 2002; Vulkan, 2000). Probability matching has been found even when a large number of trials is used and the participants are clearly able to explain which outcome occurs most often (Koehler & James, 2009).

As probability matching is clearly not the optimal behaviour, there have been many attempts to explain this phenomenon (e.g. Gaissmaier & Schooler, 2008; Shanks et al., 2002; Yu & Huang, 2014). One possible explanation is that participants may be trying to find patterns in the sequence of outcomes and using those patterns in their predictions (Gaissmaier & Schooler, 2008; Shanks et al., 2002). Yu & Huang (2014), based on findings from a study into visual decisions, believe that participants are actually maximising, but that participants have an "implicit assumption" that the environment

will change. Probability matching is observed in studies with no switch in the underlying rules, as well as tasks like that considered here which do have a rule switch. For the task I investigate, as described in Section 2.2, there were four blocks of trials with feedback validity (FV) of 73.3% and four with FV of 83.3%, making the overall average FV used in the experiment 78%. In Section 2.3, we saw that the average level of maximising measured from participant responses was 74.5%, so maximising behaviour was slightly below the level of the feedback validity. This maximising measure is based on the underlying rules set by the experimenters without considering how the underlying rules are inferred.

Taylor et al. (2012) compared the performance of two groups of participants who had to predict which of two images would appear on the next trial and the participants knew which image was more likely to appear. One group of the participants was given an additional explanation for why one image was more likely to appear. This group was told that each image showed the result of a toss of a commemorative coin, but that a production mistake meant that one side would come up more often than the other. After the experiment, both groups of participants were asked to estimate the proportion of times the more likely image appeared. Taylor et al. (2012) found that both groups gave similar estimates to the proportion of times the more likely image appeared. The group which was given the additional explanation for the difference showed behaviour which was closer to maximising than the group without the explanation, who showed approximately probability matching behaviour. This showed that the additional explanation influenced behaviour although the explanation was not relevant to the prediction task as both groups knew which image appeared more often.

Researchers have extended the colour prediction tasks to include switches in rules, for example where the colour which is more likely to be rewarded changes during the task, requiring the participants to be flexible in their application of prior experience. One such study is the probabilistic learning task of Behrens et al. (2007), in which participants had to choose between a blue and a green stimulus only one of which would be rewarded. Within each colour on screen was shown a reward value which was the amount that colour would pay out on the next turn if it won. The rewarded colour was determined probabilistically with one colour favoured over the other, this was independent of the varying reward values. Behrens et al. (2007) contrasted a stable period of 120 trials, in which the probability of reward for each colour did not change,

with a period of unsignalled rule changes. Behrens et al. (2007) found that participants were able to react to changes in the underlying situation, showing flexibility. Hampton et al. (2006) also studied behaviour in response to a task with a choice of one from two visual stimuli, but in this case the probabilities of each stimulus giving a monetary reward were independent of each other. With Hampton et al. (2006), one stimulus led to a win on 70% and a loss on 30% of trials while the other stimulus gave 60% losses and 40% wins. The favourable stimulus switched during the task and Hampton et al. (2006) concluded that participants could successfully take account of structure in the environment in terms of these switches.

Flexible learning has also been shown in other probabilistic learning tasks which allow the choice from more than two options, often known as bandit tasks (Dayan & Daw, 2008; Krugel et al., 2009; Payzan-LeNestour & Bossaerts, 2011). Bandit tasks take their name from slot machines in a casino which are also known as one-armed bandits. Bandit tasks can be likened to choosing which slot machine to play and looking at how people decide when to change from playing one machine to another, although without the casino being busy and slot machines being already in use. These bandit tasks can be used to determine the extent to which participants stick with what they already know, exploitation, or try the different options to see whether things have changed, exploration (Cohen et al., 2007). Exploration is not needed in simple two alternative tasks in which it is clear that the alternatives are coupled, for example predicting the next of two colours, which can also be referred to as a one-armed bandit task.

2.4.2 Computational modelling of behaviour

Describing human behaviour by the patterns or proportions of correct responses does not give any indication as to how that behaviour is produced. One way to quantify the behaviour is to examine which computational models best fit that behaviour. This can allow behaviour to be expressed in terms of underlying rules, but without giving a suggestion as to how those rules are implemented in the brain.

Two contrasting machine learning approaches are often used to model human behaviour in similar tasks to that considered here: Bayesian learning and reinforcement learning (e.g. Behrens et al., 2007; Hampton et al., 2006). Bayesian learning requires prior assumptions about the causal structure of an environment, and when those assumptions are correct, the performance is optimal given only the information available

up to a trial. Bayesian learning can theoretically produce behaviour which receives more rewards but, as well as needing knowledge of the environment, requires more computation than the simpler reinforcement learning.

Reinforcement learning does not depend on existing assumptions, but may not be appropriate in changing environments (Sutton & Barto, 1998), for example where there is a switch in the underlying rules that govern an environment. In reinforcement learning (RL), an agent monitors rewards received when testing actions, and uses this information to select future actions in order to try to gain rewards and avoid punishments without having to incorporate the information into a prior model.

Using reinforcement learning, an agent makes trial by trial adjustments to the predicted value of a particular action, that is a prediction of how much reward is expected from that action. When an outcome is received, a *prediction error* is calculated as the difference between the predicted value and the outcome. A positive (negative) value for the prediction error indicates that the outcome was better (worse) than expected. When using reinforcement learning, a *learning rate* controls how much influence this prediction error has in changing the predicted value of an action from its previous value. A higher learning rate gives a high priority to only the most recent outcomes rather than taking account of a long run of trials. If an environment has probabilistic outcomes, or expected uncertainty, then it is better to use a low learning rate and take account of many previous outcomes so as not to be too swayed by single outcomes opposite to that expected. In contrast, if an environment is volatile and changing unpredictably, it would be better to use a higher learning rate as further back in time the environment may have been in a different state. By finding the learning rate which gives the best fit to the behaviour of an individual, the learning rate can be interpreted as a characterisation of that individual.

Reinforcement learning or Bayesian inference can both be used to compute an underlying belief which changes as new information is received but that belief needs to be converted into an action taken on each trial. Behavioural studies have made it clear that, especially when faced with probabilistic outcomes, humans are likely to sometimes guess or try to randomise their responses rather than always respond in alignment with their underlying understanding of a particular task. This behaviour can result in probability matching being observed as described in Section 2.4.1 above. Daw et al. (2006) found a softmax rule to be a good model for the randomness in human decisions and

such a rule is often used to select the action in models of behaviour, (e.g. Frank et al., 2007; Jocham et al., 2009; Payzan-LeNestour & Bossaerts, 2011). A softmax rule varies the amount of randomisation of responses above that of an underlying belief using a *temperature parameter* to control how much randomisation is used. A low temperature gives a high probability of choosing the action with the highest belief, even when the beliefs in each action are quite close. In contrast, a high temperature results in mainly random responses, for models which have only two actions giving probabilities close to 0.5 for each of the actions. In this case, even when there is a big difference between the underlying belief in the different options, the probability of selecting each option will only differ by a small amount. The softmax rule can be applied independently to the model used to calculate the underlying belief. Finding values for the temperature to fit the behaviour of individuals gives another way of characterising those individuals' behaviour.

One aim of computational modelling is to try to understand how decisions vary in different conditions. One way to consider this when using reinforcement learning is by fitting different learning rates to the same individual participant under different experimental conditions. Jocham et al. (2009) used this approach to characterise behaviour in two different conditions of probabilistic feedback. They found a higher learning rate with a higher level of feedback validity.

Several investigations have been made into whether human behaviour is better represented by a Bayesian or reinforcement learning style. Hampton et al. (2006) and Behrens et al. (2007) both found that Bayesian models were a better fit to behaviour than reinforcement learning models in probabilistic tasks with two options and rule switches. Hampton et al. (2006) compared a hidden Markov model to a reinforcement learning model that made no assumptions about the structure of the environment. They concluded that participants make assumptions about the structure of the environment. Nassar et al. (2010) found a Bayesian model to be a better fit in a different probabilistic task with environment changes. In these studies, Hampton et al. (2006) and Nassar et al. (2010) told their participants to expect changes in rule, whereas Behrens et al. (2007) did not. Some studies have found that a Bayesian model is a better fit to human behaviour than simpler models such as reinforcement learning only in conditions when participants have been told to expect changes in rule (Payzan-LeNestour & Bossaerts, 2011; Wilson & Niv, 2012).

Charness & Levin (2005) asked participants to select one of two urns from which a ball would be drawn. There were two types of ball one which gave a reward and one which did not. The participants knew that the environment could be in one of two possible states but did not know which state applied as each trial began. The right urn contained either all valuable or all worthless balls depending on the state. In both states, the left urn contained a mixture of balls but contained a higher proportion of valuable balls when the environmental state was such that the right urn contained all valuable balls. Each trial consisted of two draws from the urns, on some of the trials the participants were required to choose from the urn specified by the experimenters for the first draw. Looking at whether participants chose the same urn on the second draw or switched, Charness & Levin (2005) compared a Bayesian learning model to reinforcement learning and found that participants made errors in circumstances when the prediction of the two models differed. They amended the reward structure of the task and found fewer errors, and concluded that the emotional affect resulting from outcomes plays a big part in reinforcement learning.

Although applying Bayesian inference would lead to better decisions, there is neural evidence that reinforcement learning is strongly implicated in human decision making (see e.g. Niv, 2009). The amount of the neurotransmitter dopamine released in part of the brain is related to reward and punishment. Rather than indicating reward and punishment directly, Schultz and colleagues suggested that dopamine levels signal the difference between an expected reward and that actually received (see e.g. Schultz, 1998). This difference forms the prediction error which is calculated in reinforcement learning, forming a link between theoretical reinforcement learning and neural processes.

In the studies described so far, the participants had to learn from their experience of outcomes which often took the form of rewards and punishments which took the form of gains and losses of money or tokens. Kahneman & Tversky (1984) proposed the concept of loss-aversion, which suggests that behaviour changes more in response to losses than to gains of similar magnitude. The ideas of loss-aversion are often tested in studies of response to risk, that is where participants choose between alternatives with known outcome probabilities. An example (from Kahneman & Tversky, 1984) is a choice between a safe or risky option, where the risky option has an 85% chance of winning \$1000 and a 15% chance of winning nothing and the safe option pays out \$800 with certainty. People tend to prefer the safe option even when the expected reward

from the risky option is higher than that from the safe option, as in this example where the expected reward from the risky option is $850.

As an alternative mechanism to loss-aversion, Yechiam & Hochman (2013b) proposed a loss-attention mechanism in which losses cause participants to attend more closely to a task and so losses decrease the amount of randomisation. In their examination of the loss-attention hypothesis, Yechiam & Hochman (2013a), used several tasks which involved repeated selections between a safe and a risky option where the probabilities had to be learnt from experience. They tested their loss-attention model by fitting a choice sensitivity parameter for each task. This parameter is the inverse of the temperature parameter described in relation to softmax action selection. They found less randomisation of responses in tasks in which losses were possible compared to tasks without losses.

Not all studies find asymmetry in human responses to wins and losses. Pessiglione et al. (2006) required participants to select one of two stimuli on each trial. There were three pairs of stimuli, one pair produced a monetary gain or nothing, one pair produced a monetary loss or nothing and the third pair always gave nothing. These outcomes were given probabilistically and were unknown to the participants at the start of the study. The participants learnt equally quickly to select the stimulus to gain money as to avoid the stimulus to lose money.

Yechiam & Hochman (2013b) noted that few studies comparing alternative computational approaches to learning from experience in dynamic environments have considered separate effects of reward and punishment. Ito & Doya (2009) and Guitart-Masip et al. (2012) are examples of studies which do differentiate learning from rewards and punishments, doing so by fitting different reward values following a win or a loss. Guitart-Masip et al. (2012) had four fractal images which signalled whether participants should respond or not to gain rewards or avoid punishments, these associations had to be learnt from experience and there was no switch in associations. Guitart-Masip et al. (2012) fit a number of different reinforcement learning models to behaviour, the best fit model did not scale rewards and punishments differently. Analysing the decisions of rats in two-stage probabilistic decisions, Ito & Doya (2009) found that a reinforcement learning model with different reward values after a win and a loss was a better fit to the rats' behaviour than reinforcement learning without differentiation between wins and losses.

Although these studies compared alternative learning mechanisms, neither considered Bayesian models which required assumptions about the nature of the environment.

Motivated by neural structures which seem to imply different pathways for learning from a win and a loss, Frank et al. (2007) used separate learning rate parameters following positive and negative feedback when using a reinforcement learning model to analyse human behaviour in a probabilistic task. In their task, participants learnt by selecting one from a pair of images. There were three different image pairs and in each case one of the pair was rewarded with a higher probability than the other, with no switch in the probabilities. Frank et al. (2007) found that when fitting to participants behaviour, the mean learning rate following a win was higher than that after a loss. Frank et al. (2007) were looking at associations between genetics and reinforcement learning parameters and they did not compare alternative models of behaviour.

Niv et al. (2012) used asymmetric learning rates to examine risk sensitivity. In their task, participants had to learn what reward was associated with six different colours in order to determine which colour to select when a pair were presented. Five of the colours had deterministic rewards, one had a probabilistic reward. The main focus of their work was how participants behaved when given a safe option with a constant payout against an option which had double that payout on half the trials and a zero payout otherwise. Niv et al. (2012) suggested that a higher learning rate after a negative prediction error than after a positive prediction error leads to risk aversion, and this was the relationship they found for 81% of their participants.

Cazé & van der Meer (2013) described the benefits of using different learning rates according to the previous outcome. Cazé & van der Meer (2013) considered tasks of learning from feedback with probabilistic outcomes for two uncoupled options in a static environment. Rather than model behaviour, they analysed the performance of reinforcement learning strategies as computational mechanisms for learning. As the alternatives were uncoupled, exploration was necessary in the task and so they used softmax action selection to select responses according to the underlying belief of the model. They described different schemes for probabilistic reinforcement under which it is preferable for learning rates to be asymmetric and which learning rate should be higher than the other. They concluded that it is beneficial to have asymmetric learning rates. Cazé & van der Meer (2013) also make suggestions as to how learning rates might adapt to a situation through meta-learning.

Recently, Gershman (2015) investigated asymmetric learning rates as fit to human behaviour. Gershman (2015) compared a number of reinforcement learning models with softmax action selection to human behaviour in high or low reward situations. In a high reward situation, the probability of reward is greater than 0.5 for two options. In a low reward condition both options have probabilities less than 0.5 of a win. Gershman (2015) found strong support for models with asymmetric learning rates, with the learning rate for a negative prediction error higher than that for a positive prediction error in line with Niv et al. (2012). Gershman (2015) also considered meta-learning methods inspired by Cazé & van der Meer (2013) but found no support for the inclusion of meta-learning into the models.

2.5 Conclusions

In this chapter, I have described the study carried out by Bland & Schaefer (2011) and found that the behaviour observed was in line with that found by other researchers. I reviewed computational models of similar tasks where the computational models try to describe a structure or set of rules on which participants base their decisions. One commonly addressed question is whether humans naturally apply Bayesian reasoning or the more simple to compute reinforcement learning. Research suggests that different situations prompt different learning styles to be applied.

Differences between human responses to wins and losses have been a prominent area of study in psychology building on the work of Kahneman & Tversky (1984). This factor is only recently being included when comparing different underlying models of decision making. When including asymmetries in response to wins and losses, researchers only amend the learning rates in reinforcement learning but not the temperature parameter even though action selection is an important part of decision making.

The lack of consideration of asymmetric responses to wins and losses when comparing different decision making strategies to human behaviour prompted me to model the behaviour reported by Bland & Schaefer (2011) and to allow asymmetric parameters including the temperature parameter. In similar studies, researchers suggested that participants' learning styles adapted to the structure of the environment as shown by a better fit of Bayesian than reinforcement learning models. I proposed to test whether this was also the case in the task of Bland & Schaefer (2011). Many studies (e.g. Behrens

et al., 2007; Hampton et al., 2006) compare the fit of alternative learning models to the group of participants as a whole without much attention to individual differences between participants. As we observed a range of behaviour in the study of Bland & Schaefer (2011), I was also interested in how individual differences are captured by the different models.

In comparing reinforcement learning to Bayesian models, I wanted to consider how much difference there was between the two modelling styles in ability to respond in this particular task when the models are considered in isolation without trying to model human behaviour.

Chapter 3

Modelling the Task

3.1 Introduction

In Chapter 2, I introduced the psychological task of Bland & Schaefer (2011) which forms the focus of this thesis. Participants had to choose one of two buttons in response to a red or blue triangle shown on screen and were given on screen feedback as to whether each response was correct or not. The feedback given was determined probabilistically based on underlying rules of which the participants were not made aware. In this chapter, I present my computational modelling of the task of Bland & Schaefer (2011). Much of the work described in this chapter has been published in Duffin et al. (2014).

Inspired by some of the studies into modelling human behaviour in probabilistic situations described in Chapter 2, I investigate whether reinforcement learning or Bayesian learning is a better fit to the participant behaviour observed during the task of Bland & Schaefer (2011). There has been much research into the differential effects of rewards and punishments on humans in various situations, but not in studies comparing reinforcement learning and Bayesian learning as fits to human behaviour. This led me to implement learning models which have different parameters for the trial after a win compared to the trial following a loss.

When comparing Bayesian inference and reinforcement learning to human behaviour, it is usually on the basis that Bayesian learning leads to decisions which reap more rewards than reinforcement learning due to the ability of Bayesian learning to incorporate the structure of the environment. For the task of Bland & Schaefer (2011), I compare these two approaches when adjusting each model to give the most rewards that model

can produce. In this case the models are not being used to try to replicate human behaviour.

3.2 Overview of the models

To consider whether reinforcement learning or Bayesian inference is a better fit to behaviour, I tested a number of different models. Here I give a brief introduction to the models used, starting with the different reinforcement learning models, full details can be found in the following section.

As described in Chapter 2, reinforcement learning requires the update of a predicted value of an action each time feedback is received. The update depends on a learning rate parameter which is fitted to human behaviour and controls how much more influence recent outcomes have compared to past ones.

To validate the assumption in Chapter 2 Section 2.2 that participants expected the environment to be coupled, an uncoupled reinforcement learning model (UNC) is used which considers the colours seen and the button presses to be independent of each other and a separate predicted value is maintained for each combination of button and colour.

The remaining models tested assume that the environment is fully coupled. In this case, I describe each response as being one of two types, where a type 1 response applies to button 1 being pressed following a red stimulus and button 2 following blue, with type 2 the opposite. Using this description the actual colour presented on each trial is ignored. This uses the assumption that participants expect that each colour requires the opposite button press, that is the environment is coupled.

Two additional reinforcement learning models are used, standard reinforcement learning (RL) and a win loss modified reinforcement learning model (WL). Motivated by findings of asymmetry in human responses to wins and losses as described in Chapter 2, the WL model, unlike the other reinforcement learning models, allows wins and losses to have different influences on learning by allocating two learning rates to each participant, treating trials following a loss and a win separately.

I used two models with Bayesian reasoning, a simple hidden Markov model (HMM) based on the work of Hampton et al. (2006) and a more complex model (VOL) following the work of Behrens et al. (2007). These Bayesian models are based on hidden Markov models which assume that rewards are governed by a hidden environmental state which

cannot be directly observed but can be inferred. Bayesian reasoning is used to determine a probability of reward for each response type. The HMM assumes two hidden states for the environment, with each state determining which response type is rewarded more often. The VOL model assumes an additional level of structure to the environment, volatility, or how quickly the environment is changing. As with Behrens et al. (2007), a hidden state relates directly to the probability of reward for a particular response, in my case representing the probability of response type 2 being rewarded, without the assumption in the HMM of only two states. This gives a flexible model which can respond to any change in state including changes in feedback validity.

In all models, following the calculation of a belief or probability, I apply the softmax action selection rule to determine the probability of making each action on each trial. As described in Chapter 2, the softmax rule is a commonly used way to model the randomness which is present in human responses. Using the softmax rule, I fit temperature parameters which characterise the degree of randomness of a participant's responses given each underlying model. For the HMM, VOL and WL models, I fit two temperature parameters per participant, the parameter being based on whether the previous trial resulted in a win or a loss. This allows differences in response to wins and losses to be a consequence of differences at the action selection stage of the decision.

Given a set of parameters and a model, I calculate a probability for each action on each trial for the outcomes received by the participant. For each model, parameters are fitted to each participant's behaviour by searching possible values to maximise the likelihood of the parameters over all trials. After finding a set of parameters which were the best fit for each participant with a calculated likelihood of those parameters, models were compared by using the Bayesian Information Criterion (BIC) which penalises models which have more parameters (Lewandowsky & Farrell, 2011).

3.3 Details of the models

3.3.1 Reinforcement learning

Uncoupled reinforcement learning (UNC)

I gave an informal description of reinforcement learning in Chapter 2; here I explain fully how this was implemented for this thesis. Reinforcement learning considers the predicted value of, or rewards which will be obtained by taking a particular action. As

described in Section 3.2, one reinforcement learning model tested, referred to as uncoupled reinforcement learning (UNC) does not make the assumption that the environment is coupled. In the UNC model, four separate predicted values are maintained, one for each combination of colour seen and button pressed. The reward value, R, is set from the win or loss feedback, ignoring the actual number of points won and lost, by setting R to 1 if a reward was given on that trial and set to 0 otherwise. At each trial, t a prediction error, $\delta(t)$, is calculated as the difference between the reward and the predicted value of the response made to the colour shown as follows

$$\delta(t) = R(t) - Q_i(t),$$

where i takes four values representing the combination of the button selected in response to the colour shown on trial t and $Q_i(t)$ is the predicted value of that combination represented by i. This prediction error, $\delta(t)$, is used to update the expected value $Q_i(t)$ for the relevant action and colour combination for the next trial, using a learning rate, α, with a value between 0 and 1 as follows

$$Q_i(t+1) = Q_i(t) + \alpha\delta(t).$$

A learning rate parameter, α, was calculated for each participant to fit the behaviour recorded. On each trial, the three $Q_i(t)$ values for colour and button combinations not experienced on that trial are maintained for the next trial, without any forgetting.

Standard reinforcement learning (RL)

In the RL model, I consider the colours and buttons to be opposites, and use the response types as described in Chapter 2 and Section 3.2 above. In this formulation, every trial gives full information about both possible actions and so only one predicted value needs to be maintained whilst still applying the reinforcement learning framework as for the UNC model.

Suppose $Q(t)$ is the predicted value of using response type 1, on trial t and that $R(t)$ is the reward associated with response type 1. In this case the reward, R, is set to 1 or 0 as follows. On trials in which a type 1 response is made, if a reward is given, R is set to 1 otherwise R is set to 0. On trials when response type 2 is carried out, if the response was rewarded R is set to 0 otherwise R is set to 1.

At each trial a prediction error, $\delta(t)$, is calculated and the value $Q(t)$ updated as described for the UNC reinforcement learning model. Using the coupled relationship between the two response types, the expected value for a type 2 response can be calculated as $1 - Q(t)$.

Win loss modified reinforcement learning (WL).

In this model, the predicted values for responding in accordance with each rule are calculated exactly as for standard reinforcement learning but in the WL model, each participant is assumed to have two different learning rates which apply according to whether they received a reward or punishment on the previous trial.

3.3.2 Bayesian models

Hidden Markov model

The hidden Markov model (HMM) used in this work is broadly based on the work of Hampton et al. (2006). This model assumes that the outcome on each trial depends probabilistically on the value of a hidden state at that trial and on nothing else. The hidden state is a random variable which always takes one of two possible values and is equivalent to the two underlying rules set by the experimenters, but unknown to the participants, as described in Chapter 2. In each hidden state one of the two response types is rewarded the majority of the time.

To introduce the model formally, at trial t the hidden state is represented by X_t and has two possible values, denoted by x^i where i can be 0 or 1. Upper case letters are used to denote random variables and lower case to represent a particular value that the random variable can take.

The outcome on trial t in the study is represented by the random variable Y_t and has two possible values for the two response types which can be rewarded. Using the notation $\mathbf{P}$ to denote a set of probabilities, $\mathbf{P}(Y_t)$ represents a probability for each value which Y can take. We can denote the set of probabilities $\mathbf{P}(Y_t)$ by a column vector, where for example for Y, $\binom{a}{b}$ indicates that there is a probability a that a type 1 response is rewarded and b that type 2 is rewarded. I use y_t as shorthand for $Y_t = y$ where y is the known (but arbitrary) value taken by Y on trial t. When an outcome

has been observed, then the probability becomes 1 for one response type and 0 for the other, so y_t can be represented by $\binom{1}{0}$ or $\binom{0}{1}$.

In the HMM, the hidden state only depends on its value at the previous trial and on the set of constant probabilities, $\mathbf{P}(X_t|X_{t-1})$, for staying in the same state or switching. This is equivalent to saying that if rule 1 applies then there is a constant probability of rule 1 applying at the next trial. Given a hidden state, there is assumed to be a constant probability for each possible outcome y, this set of probabilities is written $\mathbf{P}(Y_t|X_t)$. In matrix notation, $\mathbf{P}(X_t|X_{t-1})$ is given by:

$$\begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}$$

where p which is between 0 and 1 represents the probability of being in the same state on trial t as on trial $t-1$. Representing the parameters this way assumes that there is symmetry in the underlying environment. This representation assumes that the probability of a switch from one environmental state to the other is the same whichever of the two states the environment is in initially. By replacing p by another variable, say q, the probabilities $\mathbf{P}(y_t|X_t)$ can be expressed in the same form as for $\mathbf{P}(X_t|X_{t-1})$. This would assume that if one response type is rewarded with a set probability in one environmental state then the other response is rewarded with the same probability in the other state. To fit this model to human behaviour, the probabilities, p and q, are considered to be parameters which are fitted to the observed behaviour.

The HMM assumes that the participants estimate these two sets of probabilities and that they do so quickly enough that they can be considered to be constants. It also assumes that it is not necessary to estimate the two different experimental levels of feedback validity and that participants do not realise that switches only occur at 30 trial intervals.

If we have values, or estimates, for the probabilities of the environment being in each possible state after t trials, $\mathbf{P}(X_t, y_1, ..., y_t)$, then we can incorporate the probability of a switch in state $\mathbf{P}(X_t|X_{t-1})$ to give an estimate for the probabilities for each state at trial $t+1$ which can be used to inform our responses. When an outcome is observed, the probabilities can be updated using the probabilities of the outcome actually observed given each hidden state, $\mathbf{P}(y_t|X_t)$. This gives a process which can be used at each time step and only requires the probability distribution for X to be stored. Figure 3.1 shows the relationship between the variables in the HMM.

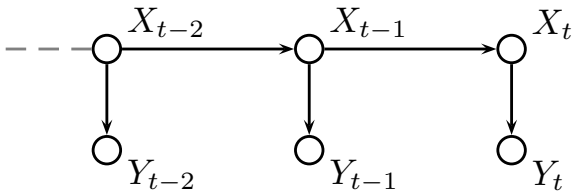


Figure 3.1: Representation of the hidden Markov model, where X_t represents the environmental state at time t and Y_t represents the outcome which only depends on the current state.

To show this process formally, we take the joint probability distribution at trial t for X with all the known observations $y_1, ..., y_t$, written $\mathbf{P}(X_t, y_1, ..., y_{t-1}, y_t)$, and use the definition of conditional probability to write

$$\mathbf{P}(X_t, y_1, ..., y_{t-1}, y_t) = \mathbf{P}(y_t|X_t, y_1, ..., y_{t-1})\mathbf{P}(X_t, y_1, ..., y_{t-1}). \tag{3.1}$$

But y_t depends only on X_t so

$$\mathbf{P}(y_t|X_t, y_1, ..., y_{t-1}) = \mathbf{P}(y_t|X_t). \tag{3.2}$$

Substituting Equation 3.2 into Equation 3.1 gives

$$\mathbf{P}(X_t, y_1, ..., y_{t-1}, y_t) = \mathbf{P}(y_t|X_t)\mathbf{P}(X_t, y_1, ..., y_{t-1}). \tag{3.3}$$

Now we introduce variable X_{t-1} because we know that

$$\mathbf{P}(X_t, y_1, ..., y_{t-1}) = \sum_i \mathbf{P}(X_t, x^i_{t-1}, y_1, ..., y_{t-1}) \tag{3.4}$$

where i takes values 0 and 1. Using the definition of conditional probability, Equation 3.4 can be re-written to give

$$\mathbf{P}(X_t, y_1, ..., y_{t-1}) = \sum_i \mathbf{P}(X_t|x^i_{t-1})P(x^i_{t-1}, y_1, ..., y_{t-1}). \tag{3.5}$$

Now Equation 3.5 can be substituted into Equation 3.3 to give

$$\mathbf{P}(X_t, y_1, ..., y_t) = \mathbf{P}(y_t|X_t)\sum_i \mathbf{P}(X_t|x^i_{t-1})P(x^i_{t-1}, y_1, ..., y_{t-1}). \tag{3.6}$$

The left hand side of Equation 3.6 is now written in terms of its value on the previous trial when combined with known probabilities. To begin the process, it is assumed that there is an equal probability of X being in either of the two possible states.

Hidden Markov model with volatility

The work of Behrens et al. (2007) inspired my hidden Markov model with volatility (VOL) which was presented in Duffin (2011) with the model details being repeated here. Being based on a hidden Markov model, this model shares some features with the HMM described above but makes different assumptions about the nature of the environment.

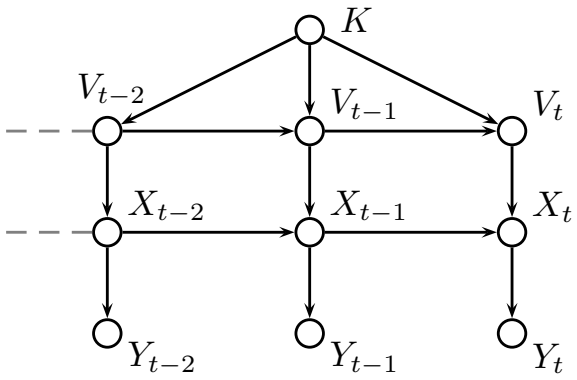


Figure 3.2: Relations between the variables in the hidden Markov model with volatility (VOL). The variables X_t and Y_t are as in the HMM in Figure 3.1 but in this case X_t depends on the volatility V_t as well as on its own previous value.

In the VOL model, as in the HMM, the outcome at a particular trial depends only on the value of a hidden state at that trial. In this case the hidden state, X represents the probability that a type 2 response will be rewarded. As X represents a probability, it must have values between 0 and 1. For computation, X was treated as a discrete random variable by taking 49 equally distributed points in the $(0, 1)$ interval. Responses can be based on the mean, or expected value, of the probability distribution over X, that is the probability that X takes each of its possible values.

As in the HMM, the value of X depends on its previous value, but in the VOL model, X also depends on the value of a second hidden variable, V, representing the volatility of the environment, which was also treated as a discrete random variable taking values between 0 and 1 in the same way as X. The volatility, V, depends on its previous value and that of a parameter, K. The parameter K is a representation of the degree of confidence in the estimate for volatility. The relationships between the variables in the VOL model are shown in Figure 3.2.

The equations for updating probabilities for the VOL model follow in a similar way to those for the HMM. For transitions from one trial to the next, there are sets of probabilities $\mathbf{P}(V_t|V_{t-1}, K)$ and $\mathbf{P}(X_t|X_{t-1}, V_t)$ for transitions of V_t and X_t respectively. In this case the joint probability to consider is $\mathbf{P}(X_t, V_t, K, y_1, ..., y_t)$ and the outcome can be incorporated as follows

$$\mathbf{P}(X_t, V_t, K, y_1, ..., y_t) = \mathbf{P}(y_t|X_t)\mathbf{P}(X_t, V_t, K, y_1, ..., y_{t-1}). \tag{3.7}$$

Now we can introduce and sum out over X_{t-1} as before to give

$$\mathbf{P}(X_t, V_t, K, y_1, ..., y_t) = \mathbf{P}(y_t|X_t) \sum_i \mathbf{P}(X_t|x^i_{t-1}, V_t)\mathbf{P}(x^i_{t-1}, V_t, K, y_1, ..., y_{t-1}). \tag{3.8}$$

Now we need to also sum out over V_{t-1} in a similar way, giving

$$\begin{aligned}&\mathbf{P}(X_t, V_t, K, y_1, ..., y_t) = \\ &\quad \mathbf{P}(y_t|X_t) \sum_i \mathbf{P}(X_t|x^i_{t-1}, V_t) \sum_j \mathbf{P}(V_t|v^j_{t-1}, K)P(x^i_{t-1}, v^j_{t-1}, K, y_1, ..., y_{t-1}).\end{aligned} \tag{3.9}$$

Equation 3.9 gives an expression for $\mathbf{P}(X_t, V_t, K, y_1, ..., y_t)$ in terms of its value on the previous trial.

Following the ideas of Behrens et al. (2007), I used a beta distribution, with a mean of the old value of X, to determine the probability distribution for X at the next time step. Also motivated by Behrens et al. (2007), I used a normal distribution to determine the probability distribution for Y. The actual distributions used for transition matrices and initial distributions were based on my previous investigation (Duffin, 2011) into replicating the behaviour of the model of Behrens et al. (2007). Like Behrens et al. (2007), I assumed that the process for determining the current state and volatility does not vary between participants.

3.3.3 Probabilities for actions

Each learning model gives a predicted value or probability for making a particular response at each trial, this can be considered to be a belief at trial t, denoted $B(t)$. For each model as described above, the value of $B(t)$ will be between 0 and 1. For the UNC model, the belief is a value for making each button press, given the colour that is displayed, for the other models, the belief is based on making a type 1 response.

To model human behaviour, these underlying beliefs have to be used to give an actual probability of each action. This is done using softmax action selection in the following way.

Given a belief $B(t)$ in a type 1 response, the probability, $P(t)$, of making a type 1 response is given by

$$P(t) = \frac{e^{\frac{B(t)}{T}}}{e^{\frac{B(t)}{T}} + e^{\frac{1-B(t)}{T}}},$$

where T is the temperature parameter which is fitted to each participant's responses and controls the amount of randomness above an underlying belief as described in Section 2.4.2.

As this study only considers two possible actions, the probability of making a type 2 response is given by $1 - P(t)$. For the UNC model, the two possible responses relate to the two buttons.

For each of the models, apart from the uncoupled and standard reinforcement learning models (UNC and RL), each participant was assumed to have two different temperature parameters, one applying after positive and one after negative feedback. When a participant failed to respond in the allowed time, I assumed that the previous belief value would be remembered and that the temperature parameter applied would be that used after a loss.

3.3.4 Fitting parameters

Given a set of parameters for a model, the process described above can be used to calculate a probability for each response type at each trial. Assuming that responses are independent given a model, the joint probability of the observed response data given a set of parameters for a participant is given by the product of the probability of each response actually made. This joint probability forms the likelihood of a set of parameters given the response data. For each participant I took the sum of the log likelihood for each response, ignoring trials in which no response was given. Using the search function `fmincon` in MATLAB (2012) to minimise the negative of the log likelihood, I found parameters to maximise the likelihood of each participant's responses for each model separately. Parameters were constrained according to the model and a minimum value had to be specified for the `fmincon` function. As temperature has to take a positive value, for each model I allowed temperature parameters to take any value greater than

or equal to 0.01. For the reinforcement learning models, I allowed the learning rate to take values between 0.0001 and 1 inclusive. For the HMM, the probability parameters took values between 0.00001 and 0.5 inclusive.

The parameter fitting process was done for each participant and model and gave a set of best fit parameters and a log likelihood value for those parameters for that participant and model.

3.3.5 Comparing models

Models with more parameters should be able to show a closer fit to the data so it is customary to penalise models with more free parameters which have been fitted to participants' behaviour (Mars et al., 2012). To do this, I compare the four models described above by calculating the commonly used Bayesian Information Criterion (BIC) for each model which is given by (Lewandowsky & Farrell, 2011).

$$BIC = -2 \log L + k \log N \tag{3.10}$$

where L is the likelihood (calculated as described in the previous section), k is the number of parameters in the model, N is the number of data points and natural logarithms are used. Lower BIC values imply a better fit to the data. For the models used here: RL and UNC each have two parameters, consisting of one learning rate and one temperature; the WL model has four parameters, two of each of learning rate and temperature; the HMM also has four parameters, two for the probabilistic structure and two temperatures, and the VOL model has just two temperatures.

3.4 Results

3.4.1 Comparing model fit

The Bayesian Information Criterion (BIC) was used, as described above, to compare each model giving the calculated BIC values shown in Table 3.1 for all participants combined. As a better model has a lower BIC value, the WL model shows the best overall fit to the data and the UNC model the worst fit. When examining the BIC for each model calculated separately for each participant, rather than just the combined value given in Table 3.1, the UNC model was the worst fit to behaviour compared to the other models for all participants. The finding that the UNC model did not fit

the behaviour supports my assumption that participants expected the red and blue stimuli to require opposite responses and that if button 1 was an incorrect response then button 2 would have been correct.

Model	BIC
RL	773.4
WL	726.3
HMM	742.8
VOL	851.1
UNC	998.0

Table 3.1: The calculated BIC for all models using all participants.

The best fit model was the WL model for 24 of the 30 participants, for four participants the best fit was the RL model and for two the HMM. Of the 24 participants for whom the WL model was the best fit, 23 had HMM as the next best fit. The differences in the BIC between the WL model and each other model were statistically significant, $p < 0.001$ in each case, $t(29) = 5.05$, 4.25 and 7.48 for comparison of WL to RL, HMM and VOL models respectively.

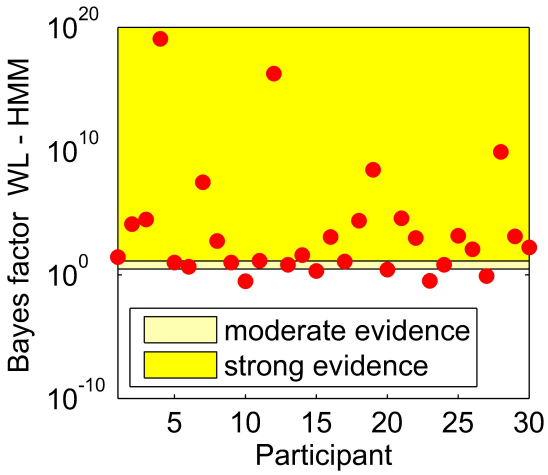


Figure 3.3: Bayes factors for the difference between the WL and HMM models for each participant giving an indication of the amount of evidence in favour of the WL model over the HMM.

The HMM and WL models fit the participants' behaviour better than the other models so I now compare these two models in more detail. Using the process described by Lewandowsky & Farrell (2011), I calculated Bayes factors for the difference between the HMM and WL models for each participant. Bayes factors can give an indication of the size of an effect; Lewandowsky & Farrell (2011) report previously proposed guidelines that a Bayes factor above 10 implies strong evidence for one model over the other, and between 3 and 10 implies moderate evidence. Figure 3.3 shows the Bayes factors for the WL compared to HMM for all participants where the ordering of the participants is based only on the order in which they participated in the study.

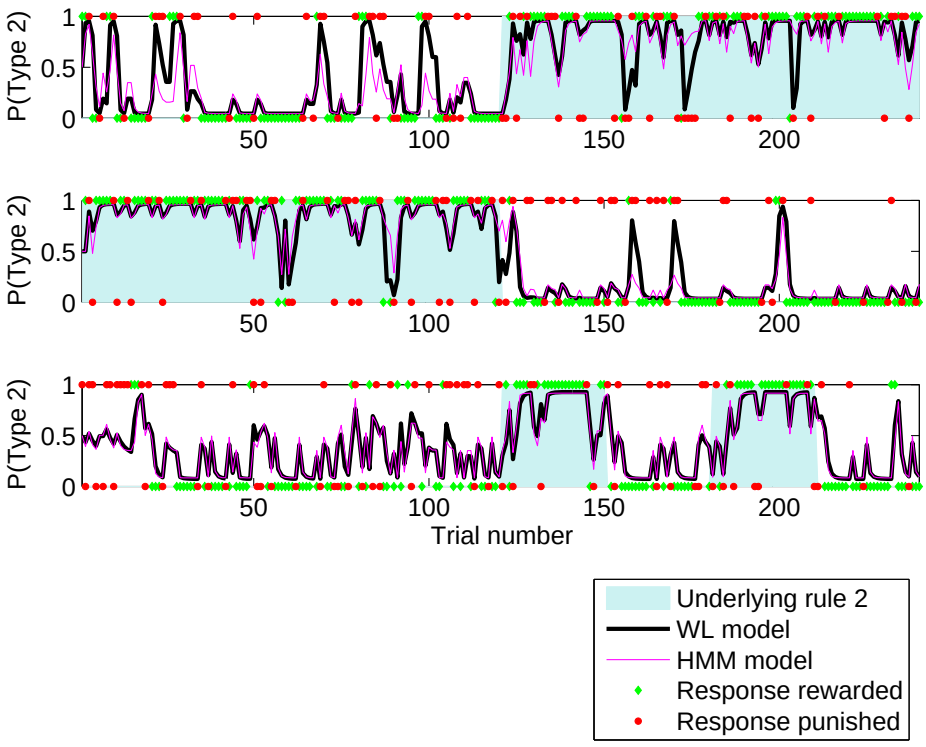


Figure 3.4: Illustration of the calculated trial by trial probabilities for making a type 2 response given by the HMM and WL models using fit parameters for three participants. The actual responses made and feedback given are shown.

Having used all trials to determine the best fit parameters for each participant and model, I used those best fit parameters to calculate a trial by trial probability of making a type 2 response. Figure 3.4 shows these probabilities for the HMM and WL models

for three participants for the first 240 trials of the study. As the log likelihood was used to find the best model and the WL model gave the best fit, where there is a difference between the models, the WL model is usually closer to the actual response made by the participant. Figure 3.4 gives a way to visualise the differences between the models and to look for patterns in the occurrences of the differences.

The RL model used here is most similar to that used by other researchers to characterise behaviour in different conditions (Jocham et al., 2009). Figure 3.5 shows the fit parameters for the RL model. The colouring of the points indicates the percentage of maximising responses of each participant, that is, as described in Chapter 2, the percentage of trials in which the response matched the underlying rule set by the experimenters. Figure 3.5 shows that the most successful participants in the task were fitted with the lowest values for both the learning rate and temperature. A lower temperature suggests less guessing and so would be expected to be a good strategy. A low learning rate means that responses would be slow to respond to negative feedback. Having a low learning rate, a participant is able to ignore randomly occurring losses but this may also mean that responses to changes in the underlying rule are also slow.

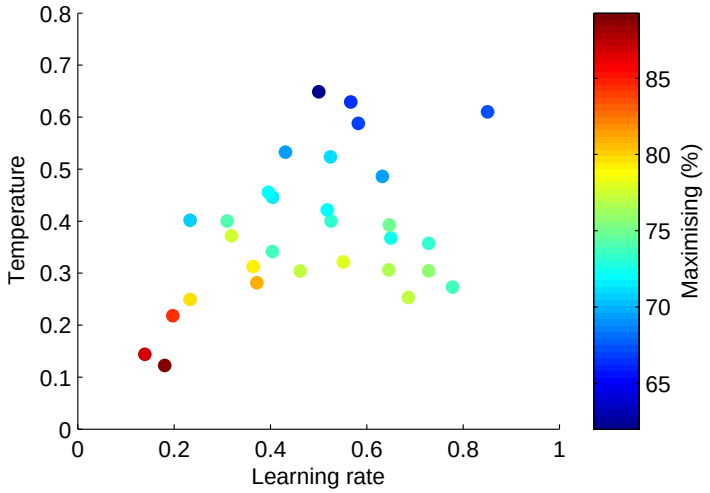


Figure 3.5: Fit parameters for all participants for the RL model coloured to show the level of maximising displayed by each participant.

Figure 3.6 shows the fit parameters for the WL model, the fit temperature was significantly higher after a loss than a win, $t(29) = 5.61, p < 0.0001$ with means of

0.87 and 0.35 after a loss and a win respectively. As a high temperature makes the probabilities for each action closer to each other, according to this model, participants chose more randomly after a loss than a win.

The fit learning rates were significantly higher after a win than after a loss, with means of 0.77 and 0.52 respectively, $t(29) = 4.52, p < 0.0001$. A lower learning rate after a loss implies that losses have less influence on the underlying belief, which allows behaviour to respond slowly to occasional negative feedback. This way people can take advantage of stable periods by not switching to the opposite response type when occasionally losing points when using the most likely response.

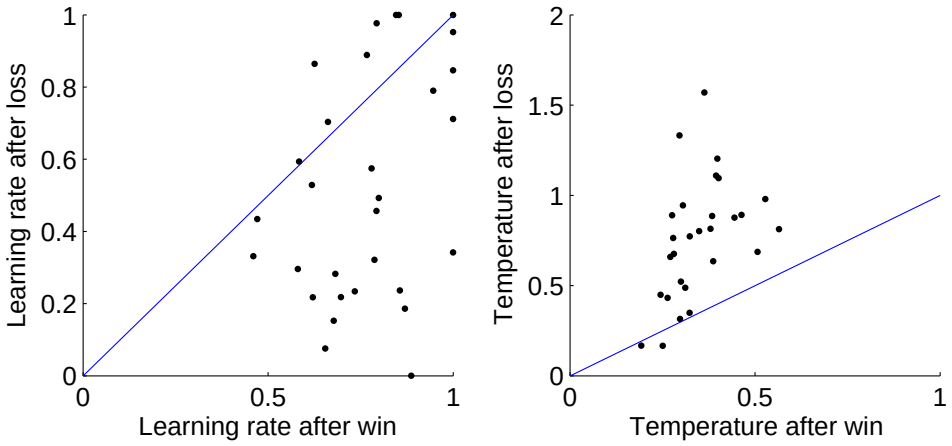


Figure 3.6: Fit parameters for all participants for the WL model with the learning rates on the left and temperatures on the right.

Figure 3.7 shows the relationship between the fit parameter values for the WL model and their equivalent parameters under the RL model. The WL learning rate after a loss is strongly correlated to the RL learning rate ($r(28) = 0.88, p < 0.0001$) and the WL temperature after a win to the RL temperature ($r(28) = 0.88, p < 0.0001$). The correlations between WL learning rate after a win and WL temperature after a loss and their RL equivalents are not so strong. For a given RL learning rate, there is a range of fit WL learning rates after a win, likewise for temperature after a loss. These differences allow the WL model to fit individual differences in human behaviour better than the RL model.

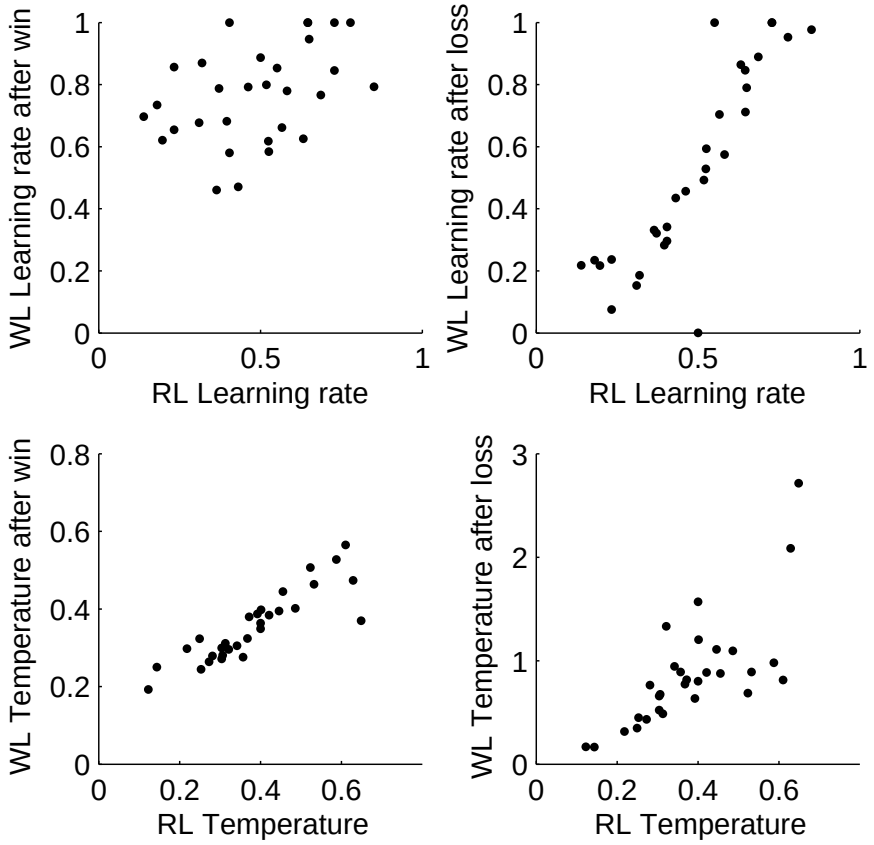


Figure 3.7: Relation between the RL parameters and their equivalents in the WL model for each participant.

3.4.2 Parameter recovery

If the fit parameters are reliable, it should be possible to take simulated data, which has been generated using known parameters, and accurately estimate those parameters (Lewandowsky & Farrell, 2011). For each model, parameters representing 'typical participants' were chosen. Each model's learning rules were used, with a random number generator to convert the probabilities calculated by the model to actual actions at each trial, to generate two sets of simulated responses to each participant's observed outcomes. This gave sets of actions to observed stimuli which were processed in the same way as the original participant responses to give estimated parameters for the simulated responses.

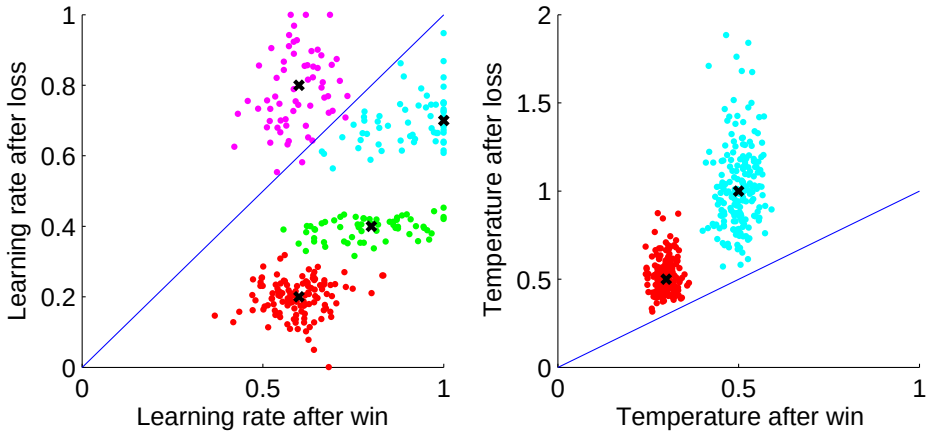


Figure 3.8: Parameters fit to data generated using the parameter values shown by crosses using the WL model.

Figure 3.8 shows that the fit parameters for the WL model are clustered around the parameters used for data generation which are shown by crosses suggesting that the parameters are reliable. For the higher temperature parameters, there is more spread in the fit parameters than for the lower temperature parameters, especially for the temperature after loss.

The left of Figure 3.9 shows the parameters representing probabilities in the structure of the HMM fit to participant behaviour. The error probability is the probability of losing when using the response type associated with the current rule. If the participants had understood the experimental generation of outcomes and were applying that knowledge, I would expect the fit parameters to be close to those approximating the generation of data, indicated by a cross on the left of Figure 3.9. The generative environment had equal numbers of blocks with feedback validity (FV) of 83% and 73%, giving an average probability of 22% of losing when using the response associated with the current underlying rule, the error probability. To approximate the probability of a rule switch, I used a probability of 0.021 based on 5 switches in 240 trials, having switches after 120 or 30 trials, the study having equal numbers of stable and volatile blocks as described in Chapter 2. The HMM assumes that this probability is constant, but the blocks of trials gave structure to the switches in the generative environment.

The right of Figure 3.9 shows the parameters fit to data generated using the HMM

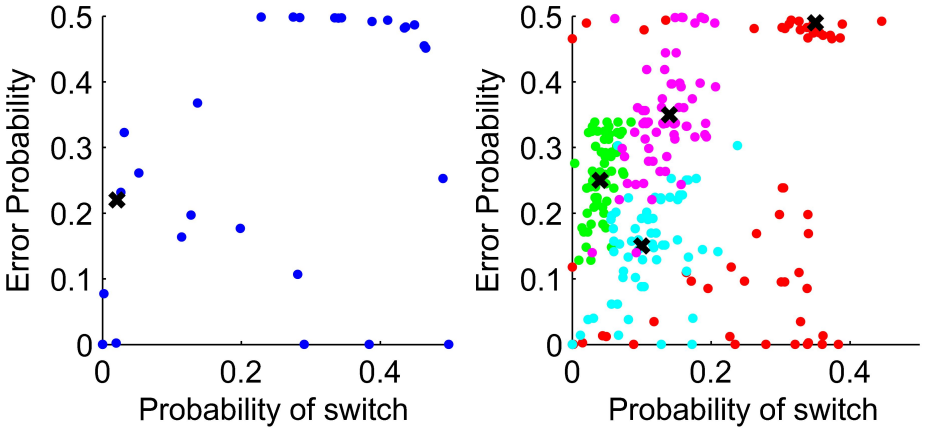


Figure 3.9: Left: Parameters relating to the probabilistic structure of the HMM fit to participant behaviour, with the structure of the generating environment approximated by a cross. Right: Fit parameters for the HMM to data generated with the parameters shown by crosses.

with parameter values shown by crosses. For the HMM, the spread of fit parameters away from the data generation parameters shows that the parameters are not well recovered. In particular, for several participants the fit value for the error probability was 0.49, that is the probability of losing when using the response type associated with the current rule. Fitting parameters to data generated with this parameter value, the estimated parameter values covered the whole range of feasible values. For the data generated with parameters closer to the actual experimental data, the fit parameters are not so widely spread. This suggests that if participants had made their responses in line with a reasonable estimation of the generative structure, this would have been recovered in the data.

3.4.3 Model recovery

Of the models tested, I found that the WL model was the best fit to participant data. If data is generated as described in the previous section and then each model is fitted to the generated data and the model fits are compared, rather than just fitting the model which generated the data, the best fit model should be that which generated the data. Using the simulated data from the previous section, I compared the fit of each model as in the analysis of participant data. Table 3.2 shows the percentage of

simulations using each model which were best fit by each model. The correct model has been identified in most cases for all of the models.

		Simulated model			
		RL	WL	HMM	VOL
Fit model	RL	99.6	8.6	18.8	0
	WL	0.4	86.0	1.6	0
	HMM	0	5.1	78.5	0.8
	VOL	0	0.3	1.1	99.2

Table 3.2: Percentage of best fit models to simulations using each of the models.

The largest incorrect identification was the finding that the RL model was the best fit for 18.8% of the simulations by the HMM. The wrongly identified simulations were those which had the parameter for the error probability set to 0.49, and the probability of a switch set to 0.35. This was also the set of parameters which could not be reliably recovered from the simulated data as described above. A simulation using these parameters always gives probabilities close to 0.5 for each response with slight preference in line with the most recent outcome. Reinforcement learning produces responses in line with the most recent outcome by setting the learning rate to one, and the probabilities remain close to 0.5 by setting a high value for temperature. In this way the same behaviour can be achieved by the HMM and RL models. Using BIC to compare models, RL will be preferred as the RL model has two parameters compared to four for the HMM.

3.4.4 How well can these learning methods do?

Human behaviour in the task of Bland & Schaefer (2011) was best fit by the WL model. Other researchers making similar model comparisons consider Bayesian models to be better in this type of task, (e.g. Behrens et al., 2007; Hampton et al., 2006). I now test which model gives better performance at the task when carried out by an ideal agent. By ideal agent, I mean an agent which always selects the action which the model suggests is most likely to give a reward, and the model parameters are chosen to give the highest number of rewards for the task. I used the sequence of outcomes received

by each participant in the task and then compared the best performance of each model on each participant's trials.

For the RL and WL models, the best parameters were found by a grid search over all possible values of the learning rates at intervals of 0.01. As these ideal agents always choose the preferred action given by the model belief, no temperature parameters are required. For the WL model, a learning rate after a win of 0.48 and after a loss of 0.24 maximised rewards. A learning rate of 0.2 gave maximum rewards for the RL model. The WL model won significantly more rewards than the RL model $t(30) = 3.53, p = 0.0014$.

For the HMM, I searched the parameter space in the region of those parameters approximating the generative environment to find the best performance. The parameters approximating the generative environment are described in Section 3.4.2. The parameters which maximised rewards were 0.021 for the switch probability and 0.2 for the error probability. There was no significant difference between the performance of the WL and HMM models $t(30) = 1, p = 0.33$. The HMM was significantly better than the VOL model, $t(30) = 4.98, p < 0.0001$.

Figure 3.10 shows the maximising behaviour, aligned with the experimental rule, of the ideal WL model in comparison to that of the participants. The percentage of responses in line with the underlying experimental rule were averaged over all participants and the ideal WL model for trials following rule switches, with each of the levels of FV shown separately. The ideal WL model has parameter values which optimise behaviour over all trials, not just volatile blocks. The ideal WL model far outperforms the participants and reaches a steady level of maximising at 100% in the high FV condition.

As well as being able to outperform humans when used by an ideal agent, the WL model can also closely simulate human behaviour. Ten sets of simulated responses were generated using the WL models with the fit parameters and the sequence of outcomes for each individual participant. Figure 3.10 shows that the simulations closely replicate the aggregate performance of the participants. Although only volatile blocks are shown, the parameters used in the simulations were those fit to participant behaviour across all trials regardless of the experimental conditions. Maximising behaviour of participants and simulations quickly adapts to a rule switch and reaches a plateau which is approximately equal to the level of feedback validity (probability matching). For trials 21 to 30 following a switch in the high feedback validity (FV = 83%) condition, participants

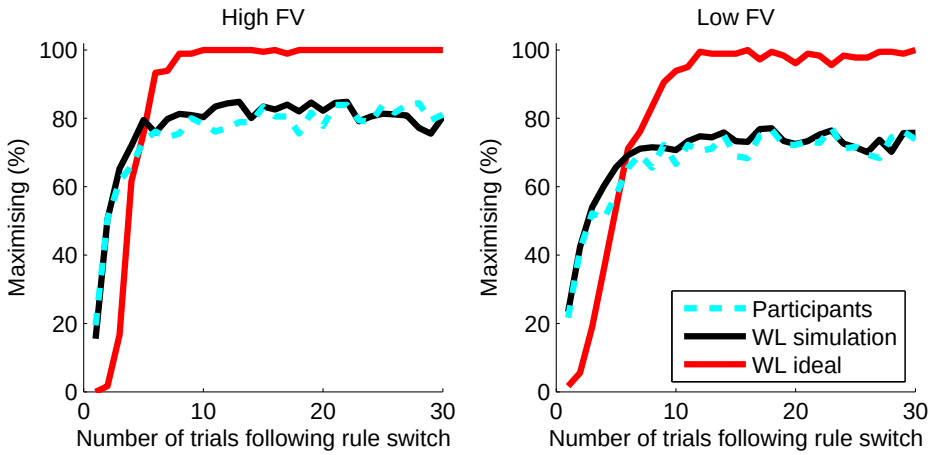


Figure 3.10: Performance of humans and the WL model following a rule switch, aggregated over volatile blocks only. The WL model run by an ideal agent (red) can outperform humans (dashed cyan). The WL model can also simulate human behaviour at this aggregate level when using parameters fit to human behaviour (black)

showed maximising of 82% and the WL simulation 81%. In the low feedback validity condition (FV = 73%), maximising by participants and the WL model was 73%.

3.5 Discussion

3.5.1 Summary of results

I found that a reinforcement learning model with separate parameters according to whether the previous trial resulted in a win or a loss (WL) gave a significantly better description of human behaviour in the two-alternative probabilistic learning task with rule reversals of Bland & Schaefer (2011) than the other models tested. The Bayesian learning models are able to adapt to changes in the environment. However, the WL model was a better fit to the behaviour than the other models although it has constant parameters throughout all trials. The difference between the fit of my WL model and standard reinforcement learning (RL) applied even when the WL model was penalised due to having more parameters than the RL model. In the WL model, the fit learning rate and temperature parameters showed a significant difference between the fit values following a win and a loss.

Comparing the performance of ideal agents on the task, I found no significant difference between the HMM and WL models. Ideal agents have parameters which are chosen to maximise rewards given the model and always choose the option given by the model as the most favourable. Bayesian models are constructed to make optimal decisions, providing that the assumptions underlying the models are correct. Although the assumptions of the Bayesian models are based on the experimental structure used to generate rewards, the HMM or VOL models as implemented and tested in this case did not outperform the WL model on this task although the WL model does not adjust its learning rate to accommodate different levels of unexpected uncertainty and volatility.

Using ideal agents, I found a small but significant improvement in performance of the WL model compared to the RL model on this particular task. All of the models, when used by ideal agents, far outperform human behaviour.

3.5.2 Relation to other work

As the Bayesian and reinforcement learning based models make different assumptions about the environment, comparing the fit of different models to human behaviour can give insights into the assumptions people make about the environment. The HMM I implemented, as with that of Hampton et al. (2006), assumes that there will be rule switches within probabilistic feedback. My VOL model, based on that of Behrens et al. (2007), expects not only rule switches but also that the frequency of switches depends on the level of volatility in the environment.

As described in Chapter 2, in comparing reinforcement learning and Bayesian models to human behaviour, both Hampton et al. (2006) and Behrens et al. (2007) concluded that the Bayesian models were a better fit to behaviour.

When Hampton et al. (2006) compared a hidden Markov model to a reinforcement learning model, the reinforcement learning model made no assumptions about the structure of the environment. This reinforcement learning model would be most similar to my uncoupled reinforcement learning model (UNC) which as explained above was a poor fit to behaviour. As with Hampton et al. (2006), I found that a hidden Markov model (HMM) was a better fit to behaviour than such a reinforcement learning model. My uncoupled reinforcement learning model, however, was not as good a fit as either the RL and WL models. From this I conclude, as did Hampton et al. (2006), that participants made some assumptions about the environment but I have no evidence that

they adjusted their rate of learning according to the structure of volatile and stable periods within a probabilistic environment. The task of Hampton et al. (2006) did not have the two options coupled and so is not directly comparable to the task of Bland & Schaefer (2011). By comparing my UNC and RL models, I found strong evidence that the participants of Bland & Schaefer (2011) did expect the environment to be coupled. It is possible that participants in other studies would also make that assumption when actually it would be incorrect to do so. As the instructions given to participants in tasks can have a strong effect on their behaviour, for example in the findings of Taylor et al. (2012) described in Chapter 2, I expect that instructions would make a difference to assumptions of the structure of an environment.

In my investigations of behaviour, I have focussed on modelling differential responses to losses and gains. Yechiam & Hochman (2013b) proposed a loss-attention mechanism and suggest that losses decrease the amount of randomisation. In the task I consider, unlike that of Yechiam & Hochman (2013a), the participants could not avoid losses as there was no way to predict the outcome on individual trials. I find a higher temperature after individual losses, implying that participants are less likely to follow the underlying belief after a loss. This does not necessarily conflict with the idea of loss-attention, as adding randomness to a response after a loss may be a mechanism for testing an underlying belief without making a large adjustment to that belief.

When modelling the different effects of wins and losses, Ito & Doya (2009) and Guitart-Masip et al. (2012) fit different reward values following a win or a loss. To maintain the symmetry of the task I study in which exactly one response is correct on each trial, I have taken a different approach and fit a separate learning rate, rather than reward value, following wins and losses. This is equivalent to fitting a different learning rate according to the sign of the prediction error, or whether the outcome was better or worse than expected.

Fitting a different learning rate after a win or a loss, I find, as did Frank et al. (2007), that the mean learning rate following a win is higher than that after a loss. Frank et al. (2007) use softmax action selection to model the actual response made, but they used a single constant temperature parameter for each participant which did not vary according to whether a win or a loss had occurred previously. I found that the fit of the model improved by allowing both learning rate and temperature to have parameters which depended on the outcome of the previous trial. I suggest that a combination of

two learning rates and two temperatures can be used to characterise the behaviour of an individual.

Niv et al. (2012) and more recently Gershman (2015) found a higher learning rate after a negative prediction error than a positive prediction error. That is a higher learning rate when the outcome was worse than expected, equivalent to a loss in my modelling, so my fit of a higher learning rate after a win does not align with the findings of Niv et al. (2012) and Gershman (2015).

Bayesian models can optimise the number of rewards received when the assumed structure for the Bayesian inference exactly matches the underlying structure of the task. I examined the performance of the learning models when, rather than being fit to human behaviour, the model parameters were selected to maximise the total number of rewards achieved in the task considered here. I use the term ideal agent to describe this use of the model. In this task, the rewards obtained by an ideal agent using the WL model was not significantly different to that of the ideal HMM. The ideal HMM was set up with parameters to closely resemble the structure of the environment set in Bland & Schaefer (2011), but with the assumption of a small but constant probability of a rule switch. In the experiment, rule switches only occurred at the ends of blocks of 30 or 120 trials. The HMM also assumes that for each environmental state, there is a constant probability of each outcome. However, the experimental data was generated using two levels of FV, of 83.3% and 73.3% as described in Chapter 2, with outcomes randomised to give the correct proportion of outcomes aligned with the underlying rule within a block. I do not believe that these differences between the generative process and the assumptions of the HMM significantly hamper the performance of the HMM. I believe that the ideal agent using the WL model is approaching an optimal level of response in this task. The ideal HMM also performed significantly better than the VOL model, my implementation of the model of Behrens et al. (2007).

The ideal WL model had a small but significant advantage over the ideal RL model in the task. The parameters for the ideal WL model, those which gave the best performance in the task, were learning rates of 0.48 after a win and 0.24 after a loss. This finding that the ideal WL model outperformed the ideal RL model, that is that asymmetric learning rates according to wins and losses, have an advantage over a single learning rate is in accordance with the work of Cazé & van der Meer (2013). Cazé & van der Meer (2013) examined the advantages of asymmetric learning rates more generally with

a range of reward probabilities but in static tasks, that is with no rule switches. I have shown that also in a task with environmental switches, asymmetric learning rates can be advantageous. As my focus is on modelling human behaviour, I have not investigated the impact of different volatility levels or feedback validity on this finding.

The tasks studied by Cazé & van der Meer (2013) do not have coupled outcomes, so exploration is needed and it would not be appropriate to always take the option with the highest belief as I did when testing ideal agents. Cazé & van der Meer (2013) used softmax action selection with a constant temperature parameter to model the action taken. It would be interesting to consider asymmetric temperatures implemented in the work of Cazé & van der Meer (2013).

As with the ideal agent, the participants in the study had significantly higher fit learning rates after a win than a loss, although the parameters fit to human behaviour were generally higher than the ideal parameters, with means for the participants of 0.76 and 0.52 for the learning rates after a win and a loss respectively. When running any of the models as ideal agents, they all far outperform human behaviour. The ideal agents do not have any randomisation of their responses whereas softmax action selection has been included when modelling human behaviour.

3.5.3 Assumptions and limitations

The task considered here, having coupled outcomes in which one or the other response is correct, does not require any exploration, or trying the different alternatives to see if things have changed. The participants were expected to know that if the button press was incorrect, then the other button would have been correct. Exploration is an important feature of learning from experience (see e.g. Cohen et al., 2007). Tasks which have more than two options automatically require exploration, as negative feedback does not show what would have been the correct response. It will accordingly be more difficult to learn when there are more alternatives.

It has been acknowledged that standard reinforcement algorithms are not suitable in complex situations in which there may be many possible states or actions (e.g. Botvinick et al., 2009). The task considered here having been carried out in a psychology laboratory, had two clearly distinguishable stimuli and response buttons with clear feedback immediately after each trial. In real life the actions available are not always clear and it is often the case that many actions may have been taken when a reward is received

so it might not be clear which action or sequence of actions were most useful, this is known as the credit assignment problem (Sutton & Barto, 1998).

Wilson & Niv (2012) compared optimal performance between a Bayesian and non-Bayesian model in their probabilistic learning task and the Bayesian model clearly had superior performance. My finding that my ideal WL reinforcement learning model performs as well as the HMM may be restricted to the case of coupled two alternative tasks. Additionally, the level of feedback validity or volatility might affect the relative performance of the different styles of responding.

The modelling presented here assumes that whatever decision making processes the participants use to make their responses, these remain constant for the whole task. I have not included any modelling which incorporates meta-learning, or the changing of parameters over time as a participant learns a task. Krugel et al. (2009) proposed a mechanism for changing the learning rate during a probabilistic reversal-learning task to allow more flexible response to reversals than would otherwise be the case with reinforcement learning. When adding meta-learning to a model, this often involves adding additional parameters to the model which then means the models are penalised more heavily in comparison, for example when applying BIC as I have done.

I have assumed that the task instructions give participants enough information to form a model. Some studies have found that a Bayesian model is a better fit to human behaviour only in conditions when participants have been told to expect changes in rule (Payzan-LeNestour & Bossaerts, 2011; Wilson & Niv, 2012). In the study I examine, participants were not given such information.

3.6 Conclusions

I found that a reinforcement learning model with separate parameters for a trial following a win to those following a loss gave a better fit to the behaviour in the study of Bland & Schaefer (2011) than the other models tested. For most of the participants, the fit learning rate after a win was higher than that after a loss which also reflected the relationship between the parameters found to be the best for the task when carried out by an ideal agent.

Having found a particular model to be a good fit to human behaviour, it is important to consider how plausible it is that the model could be implemented neurally. In the

next chapter, I describe the basal ganglia, an area of the brain known to be important in learning and action selection. I review existing neuro-computational models of decision making which are based on basal ganglia circuits.

Chapter 4

Basal Ganglia

4.1 Introduction

In Chapter 3, I described algorithmic approaches to modelling human behaviour as recorded in the psychological task of Bland & Schaefer (2011) described in Chapter 2. Such approaches can give insight into the processes carried out, but it is also important to consider how those models might be implemented in the brain (Mars et al., 2012).

The basal ganglia are a set of interconnected brain areas which have long been known to be involved in the selection of motor output (see e.g. Redgrave et al., 2010). It is now known that the basal ganglia also play an important part in reinforcement learning in the brain (see e.g. DeLong & Wichmann, 2010; Redgrave et al., 2011). Through changes to connections in the basal ganglia, stimuli can become associated to rewarding outcomes (see e.g. Redgrave et al., 2011). The basal ganglia are subcortical, near the centre of the brain, roughly shown by the red area in the plastic brain shown in Figure 4.1.

The task of Bland & Schaefer (2011) requires the participants to press one of two buttons to indicate their responses, that is they have to select motor output to produce the movement required to press a button. To succeed in the task, they have to learn associations between the colour of the stimulus shown and the response which is rewarded. This takes the form of reinforcement learning which is driven by feedback in terms of being told whether they won or lost following each button press. As the basal ganglia are involved in these functions, they provide a good focus for investigating how the psychological task may be carried out in the brain.

I describe how understanding has built up on the connections of the basal ganglia

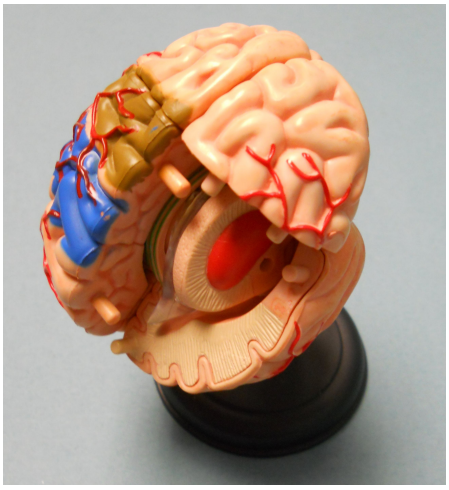


Figure 4.1: Approximate location of the basal ganglia shown in red in the centre of the brain.

nuclei and how those connections allow signals to be processed which contribute to decision making. These findings have become established as models of information flow along pathways in the basal ganglia.

Modelling information flow using a diagram showing how information passes from one region of the basal ganglia to another does not allow the study of the dynamics of the interactions. Computational models of neural processes allow dynamical interactions within isolated circuits to be examined. They also allow experimentation with different connection strengths to see how the outputs change. Such models can be validated by comparing their performance to human behaviour measured in psychological experiments. Computational models can be modified to simulate the effects of conditions such as Parkinson's disease. Computational models can also be used to suggest experiments to be carried out by neuroscientists or psychologists. The results of such experiments can then influence changes to the computational models leading to parallel advances in neuroscience and computational modelling of neural systems.

I describe a number of published computational models of the basal ganglia. It would be intractable to model the biological details of every neuron involved in complex cognitive tasks such as decision making (Cohen & Frank, 2009), so I review models which simulate the interactions between populations of neurons and consider learning and decision making at a systems level.

4.2 Biology of the basal ganglia

4.2.1 Structure of the basal ganglia

The basal ganglia receive neural signals from the cortex and the basal ganglia output projects to another region near the centre of the brain, the thalamus, as well as connections to other brain regions which are not considered here. The basal ganglia are formed from four main structures, the striatum, globus pallidus, substantia nigra and subthalamic nucleus. Figure 4.2 shows the approximate relative positions of these structures within the brain and their relation to the thalamus. Figure 4.2 represents a vertical section cut from ear to ear.

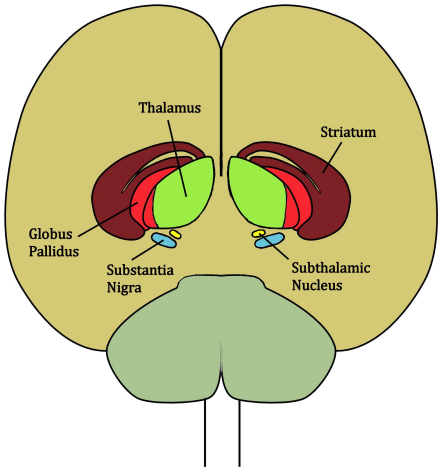


Figure 4.2: Approximate positions of the basal ganglia nuclei on a coronal section, based on Wichmann & DeLong (2009).

Neural connections through the basal ganglia allow signals to be transmitted from a cortical area, through the basal ganglia to the thalamus and back to the same area of the cortex. These connections thus form circuits or loops with separate parallel loops thought to be used for different processes (Alexander et al., 1986). These loops are involved in eye and limb movements, cortical function and emotions.

Input to the basal ganglia from the cortex passes to the striatum. Although the globus pallidus and substantia nigra are identified as single regions within the basal ganglia based on anatomical appearance, both of these are now considered to consist of

two separate areas. The globus pallidus consists of the globus pallidus internal (GPi) and the globus pallidus external (GPe).

The substantia nigra contains the substantia nigra pars compacta (SNc) and the substantia nigra pars reticulata (SNr). Output from the basal ganglia passes to the thalamus from the GPi and SNr, these two regions are often considered to be a single output structure with an internal separation, denoted GPi/SNr (Wichmann & DeLong, 2009).

Within the basal ganglia circuits, Alexander & Crutcher (1990) observed that there are multiple parallel pathways through which signals can pass from the striatum to the GPi/SNr, as shown in Figure 4.3. Using the direct pathway, information flows straight from the striatum to the GPi/SNr. Information travelling via the indirect pathway passes through the GPe and STN in order to reach the GPi/SNr. Traditionally, as suggested in Figure 4.3, it was thought that completely separate areas of the striatum formed part of the direct and indirect pathways. It is now known that some individual neurons in the striatum project to both the GPi and GPe (Nambu, 2008).

Within the indirect pathway, the output from the STN is excitatory, tending to increase the activity in the connected populations. The other nodes of the basal ganglia have inhibitory effects on the later nodes in the system.

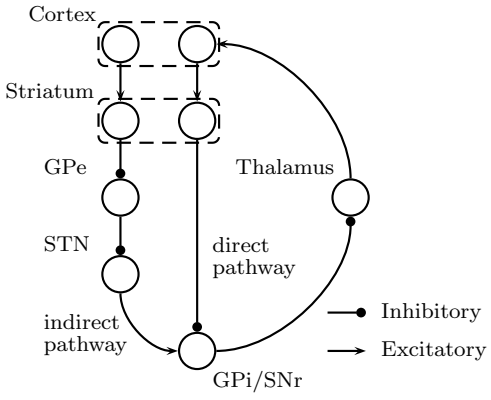


Figure 4.3: Direct and indirect pathways in the basal ganglia as identified by Alexander & Crutcher (1990). Dotted lines indicate multiple nodes in the same region.

Chevalier & Deniau (1990) described the disinhibition mechanism by which information is passed from the striatum to the thalamus. Considering the direct pathway, the normal state for GPi/SNr neurons is to emit spikes at about 100 Hz. This is known as tonic output and inhibits the thalamus thus preventing the production of an action. When the striatum fires, this inhibits GPi/SNr and so prevents the normal inhibition signal from GPi/SNr to the thalamus. The loss of an inhibition signal is known as disinhibition. Like the GPi/SNr, neurons in the GPe and STN are also tonically active.

Smith et al. (1998) added to the understanding of the indirect pathway by including a connection from GPe to GPi/SNr giving the network shown in Figure 4.4.

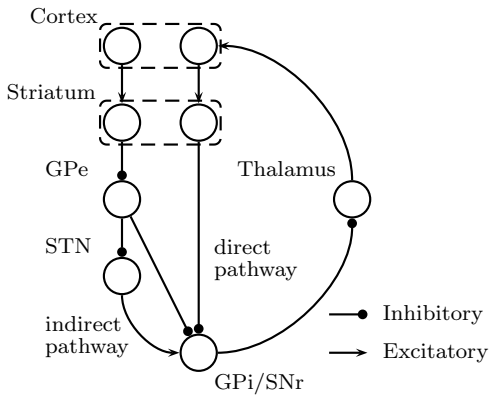


Figure 4.4: Direct and indirect pathways in the basal ganglia as identified by Smith et al. (1998).

Additional connections involving the STN, but not shown in Figures 4.3 and 4.4, were known by Alexander & Crutcher (1990) and Smith et al. (1998) but not highlighted in their descriptions of the direct and indirect pathways. The STN is, along with the striatum, a source of input to the basal ganglia, receiving signals from the cortex. Input to the STN from the cortex is directed from areas involved in motor control, whereas input to the striatum is from many cortical areas (Nelson & Kreitzer, 2014). In addition to receiving input from the GPe as shown in Figure 4.4, the STN projects excitatory output to the GPe, these reciprocal connections form the STN–GPe loop and can lead to complex firing patterns.

Nambu et al. (2002) highlighted the importance of the connections between the cortex and the STN. They gave the name 'hyperdirect' pathway to the connections from the cortex to GPi/SNr through the STN. Figure 4.5 shows this new pathway in addition to the previous two. Nambu et al. (2002) point out that transmission times for signals passing along the hyperdirect pathway is shorter than that for the direct or indirect pathway.

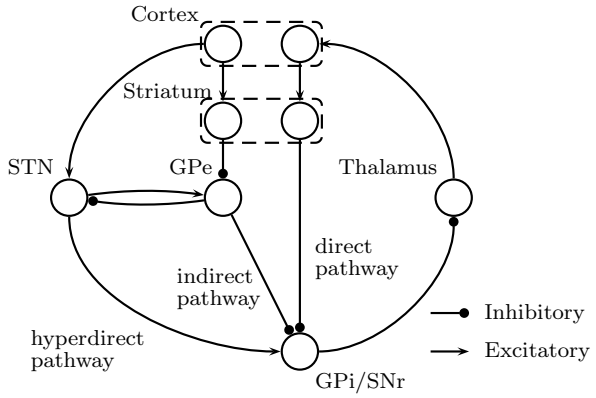


Figure 4.5: Addition of the hyperdirect pathway to the basal ganglia, showing the connections as presented by Nambu et al. (2002).

The connections shown in Figure 4.5 include those which have been the focus of computational models of the basal ganglia. Although this general connection structure has been discovered through experiment, it is still very much a simplification. It is also known, for example that the GPe projects to itself and to the striatum (Nambu, 2008). It is also not clear how much segregation there is between subpopulations which represent different stimuli or different aspects of a task. Nambu (2008) and Calabresi et al. (2014) have described problems with the simple model of direct and indirect pathways. There are different ideas of how the direct and indirect pathways interact in order to produce behaviour. Some explanations assume that the connection from the STN to the GPi/SNr is diffuse in that activation of one subpopulation in the STN increases activity in all, or many, subpopulations in the GPi/SNr. Other researchers suggest that activation in STN passes to corresponding subpopulations in the GPi/SNr.

Calabresi et al. (2014) also point out that both direct and indirect pathways are involved in action selection and that it is now known that many neurons in the striatum connect to neurons in both pathways.

4.2.2 Learning in the basal ganglia

So far the descriptions of the basal ganglia consider how areas of the basal ganglia are connected and how those connections might enable action selection, but do not describe changes over time which enable responses to adapt to the environment.

I included the SNc in the description of the basal ganglia but I have not given any role for this region. Neurons in the SNc fire in response to rewards and release the neurotransmitter dopamine (Schultz, 1992). Schultz (1992) describes how dopamine neurons respond to a reward and that the response can be the same for different rewards. When a cue consistently predicts a reward, then the SNc firing occurs when the cue is presented rather than at the time of the reward. When an animal is overtrained using many thousands of trials, the response is small. A response at the original level occurs when the animal is presented with a new but similar task. Schultz (1992) concludes that dopamine is released in response to novel and unexpected stimuli.

Further research has shown that the dopamine response correlates with the prediction error, or how different the received reward was to the expected reward (see e.g. Niv, 2009; Redgrave et al., 2011; Schultz, 1998). In Chapter 3, we saw that the prediction error was an important feature of reinforcement learning. This forms a link between the algorithmic style of modelling behaviour using reinforcement learning as in Chapter 3 and the underlying neural implementations of such learning. Dopamine signals which occur in response to rewards, or the lack thereof, are termed *phasic* dopamine and the signal is of short duration. This is in contrast to an underlying level of dopamine, known as *tonic* dopamine.

Foerde & Shohamy (2011) give a review of the role of dopamine in reward learning in the basal ganglia. When an expected reward is not received, there is a dip in the firing of the SNc neurons and so a dip in the dopamine level available in other areas. When rewards are delivered probabilistically, the amount of dopamine released at the time of the cue and the outcome depend on the probability of reward.

Dopamine produced in the SNc spreads to many areas of the brain, but the impact of dopamine levels on the striatum has become the focus of much research. Neurons in

the striatum have receptors which respond to dopamine and occur in two main types, D1 and D2 (see e.g. Gerfen, 1992). At the simplified level of considering the direct and indirect pathways as segregated, cells in the direct pathway mainly have D1 receptors and in the indirect pathway D2 receptors. The different responses and distribution of these receptors mean that dopamine received in the striatum has different effects in the direct and indirect pathways and produces different effects in response to rewards and to punishments. Dopamine affects both the excitability of striatal neurons and their plasticity, that is the strength of connections to the striatum. In the direct pathway, high dopamine makes the striatal neurons more likely to fire and promotes the strengthening of connections to those neurons. In the indirect pathway, high dopamine decreases the chance of firing to a given input and encourages the weakening of connections (see e.g. Cohen & Frank, 2009).

4.2.3 The basal ganglia and Parkinson's disease

The basal ganglia have received a lot of attention due to their association with Parkinson's disease (DeLong & Wichmann, 2010; Lanciego et al., 2012; Nelson & Kreitzer, 2014; Redgrave et al., 2010). The most notable symptoms of Parkinson's disease are a lack and slowness of voluntary movements and also tremor. As described in the section on learning, neurons in the SNc produce dopamine which affects other parts of the basal ganglia. In Parkinson's disease, neurons in the SNc die and so there is less dopamine in the rest of the basal ganglia. The impact of this lack of dopamine is often thought to change the balance of activation between the direct and indirect pathways leading to higher than normal activity in the STN leading to higher activity in the GPi/SNr which in turn makes movement less likely (Lanciego et al., 2012). Although there remain many questions about the mechanisms of Parkinson's disease, it is now acknowledged that this description is somewhat simplistic (Obeso et al., 2008; Weinberger & Dostrovsky, 2011). Weinberger & Dostrovsky (2011) describe how patterns of firing, in particular oscillatory activity, are now believed to play an important part in Parkinson's disease. The reciprocal connections between the STN and the GPe are known to allow oscillations which are implicated in Parkinson's disease.

4.3 Computational models of the basal ganglia

4.3.1 Introduction

As described in Section 4.2.1, the basal ganglia can be thought of as comprising many different nodes giving what are frequently described as three separate pathways. It is too complicated to just examine a circuit diagram and work out the behaviour of the system, or to calculate it mathematically. Computational modelling allows us to examine complex interactions between the neural populations, and to investigate the effects of different parameters in the system.

Recent reviews of computational models of the basal ganglia are given by Schroll & Hamker (2013) and Helie et al. (2013). Both reviews consider computational models which simplify the biological details in order to investigate the production of complex behaviour, described as *computational cognitive neuroscience* models by Helie et al. (2013). The reviews of Schroll & Hamker (2013) and Helie et al. (2013) each have a different focus: Helie et al. (2013) concentrate on the use of computational models to investigate the functions of different parts of the basal ganglia; and Schroll & Hamker (2013) look at the cognitive and motor outputs of various models. The different focusses for these reviews serves to highlight the fact that computational modelling of the basal ganglia does not always have the same aims. Different researchers asking different questions include different nodes of the basal ganglia or different additional circuits beyond the basal ganglia.

The studies I consider here nearly all include the direct pathway, but not all include the indirect or hyperdirect pathways. For the models I describe here, Table 4.1 shows which of the main pathways described in Section 4.2.1 are included in the models.

In addition to briefly describing some of the long-standing computational models of the basal ganglia, I will focus on models which include aspects of learning and individual differences in responses.

4.3.2 Review of selected models

Gurney et al. (2001) created a model of action selection which interprets the pathways in the basal ganglia in a different way to the commonly used notions of direct indirect and hyperdirect pathways. Instead they describe a separation between select and control pathways, where the select pathway comprises the pathway from the cortex

	Direct	Indirect	Hyperdirect
Gurney et al. (2001)	✓	✓	✓
Humphries et al. (2006)	✓	✓	✓
Frank (2005)	✓	✓	✗
O'Reilly & Frank (2006)	✓	✓	✗
Frank (2006)	✓	✓	✓
Joseph et al. (2010)	✗	✓	✗
Krishnan et al. (2011)	✓	✓	✗
Kalva et al. (2012)	✓	✓	✗
Humphries et al. (2012)	✓	✓	✗
Stocco (2012)	✓	✓	✗
Schroll et al. (2012)	✓	✗	✓
Chersi et al. (2013)	✓	✓	✓
Guthrie et al. (2013)	✓	✗	✓
Baldassarre et al. (2013)	✓	✗	✓
N'Guyen et al. (2014)	✓	✓	✓
Gurney et al. (2015)	✓	✓	✓

Table 4.1: Summary of the main basal ganglia pathways included in computational models discussed in this chapter.

through D1 neurons in the striatum to the GPi/SNr along with the pathway from the cortex through the STN to the GPi/SNr. The control pathway has a similar pattern but includes D2 neurons in the striatum and the STN, both projecting to the GPe. Control signals are then produced by the GPe and projected to both the STN and the GPi/SNr. This model requires the assumptions that the same signals are transmitted from the cortex to the striatum as to the STN and that the connections from the STN to GPi/SNr are diffuse rather than focussed on a specific target. The model of Gurney et al. (2001) has inspired many further computational models of the basal ganglia. Humphries et al. (2006) added more biological details to the model and compared their model's outputs under different levels of simulated dopamine. They showed that their model could not only select the correct action, but also switch response in response to a change in the input.

Frank (2005) developed a widely used interpretation of the direct and indirect path-

ways. As the direct pathway encourages action via the disinhibition process as described in Section 4.2.1 above, the direct pathway is referred to as the *Go* pathway. The indirect pathway has the opposite effect on the thalamus, and is named the *NoGo* pathway. Within this model stands the idea that the same cortical signal is projected to both the direct and indirect pathway and the balance between the two signals determines whether an action is made or not.

Frank (2005) demonstrates the first computational model to incorporate effects of dopamine to enable learning in the basal ganglia. This model includes the direct and indirect pathways, but not the hyperdirect pathway, or the STN. The indirect pathway is based on the direct connection between the GPe and GPi/SNr as shown in Figure 4.5. The learning mechanism implemented in the models of Frank and colleagues (see e.g. Frank, 2005) requires the model to be run twice for each trial, each time letting the activation at the nodes reach a stable state. The model is firstly run with the input signals and a medium level of dopamine to generate a response, inhibition within the nodes prevents more than one response from being selected. According to the correctness of the response, simulated dopamine is increased or decreased and the model run again. The model is based on the assumption that there is an increase in dopamine which increases the activation in the direct pathway in response to a reward and a dip in dopamine causing increased activation in the indirect pathway in response to negative feedback. This change in dopamine applies to the pathway relating to the action taken. Weights in the model are changed to reduce the difference between the two runs of the model. The model was used to simulate a probabilistic reversal learning task. Frank (2005) simulated the effects of Parkinson's disease by altering the amount of dopamine available. Frank (2005) also investigated changes to the model so that the indirect pathway provided a more global NoGo signal rather than specific to the action taken.

Frank and colleagues have continued to build on this model, adding additional brain areas and simulating more complex tasks. This allowed them to look at the interactions between brain areas and to propose functions for those areas. O'Reilly & Frank (2006) showed how the model could interact with the cortex so that information could be gated to working memory. Frank (2006) included known connections of the STN, thus adding the hyperdirect pathway giving the structure in Figure 4.5. They suggest that the STN provides a signal which will suppress all responses, and refer to this as a global NoGo signal.

As understanding progresses, different questions can be asked. Now I focus on those models which investigate aspects of exploration and exploitation within action selection. Many studies of computational models of the basal ganglia do not consider individual differences within responses. Some, for example Frank (2005) and Frank (2006) consider the effects of Parkinson's disease and the difference between this and a healthy brain. They do not consider the wide range of behaviour within a healthy population and how the variation may be manifested neurally. This does not mean that the models are deterministic with no randomness, but that the randomness is present due, for example, to the initial connection weights being selected randomly, as by Frank (2005).

As softmax action selection is considered to be a good model for the selection of an action given a set of underlying beliefs (Daw et al., 2006) and is the mechanism I used when modelling human behaviour in Chapter 3, I am interested in how this could be implemented in the brain. In particular, I wish to examine possible biological correlates of a temperature parameter as a characteristic of an individual, with the ability for the parameter to have separate values following a win and a loss. Randomness as a result of initially random weights does not give any indication of how temperature could be implemented. I now examine some existing approaches to randomness of responses which could produce a mechanism which could explain the different temperature parameters found to fit human behaviour.

Chakravarthy and colleagues have investigated the role of the STN in exploration (Joseph et al., 2010; Kalva et al., 2012; Krishnan et al., 2011). Although, as indicated in Table 4.1, these models do not include the hyperdirect pathway they do all include the STN as part of the indirect pathway from the cortex to the GPi/SNr via the GPe and the STN. Within this indirect pathway, they also include a recurrent connection from the STN back to the GPe. Expanding the ideas of Frank (2005) of how high and low levels of dopamine enhance the Go and NoGo pathways respectively, Joseph et al. (2010), Krishnan et al. (2011) and Kalva et al. (2012) add the additional concept of medium levels of dopamine promoting exploration. They see the STN–GPe loop as important in allowing this exploratory behaviour. Krishnan et al. (2011) and Kalva et al. (2012) describe models in which both the STN and GPe regions are represented by 2-dimensional layers of neurons. Each STN and GPe node is connected to the corresponding node in the opposite layer by one to one connections in each direction. In addition each layer has lateral connections. Using this arrangement, they find that

the STN–GPe circuit can produce complex oscillatory and chaotic dynamics. Kalva et al. (2012) show that chaotic behaviour is more likely when the connections between the STN back to the GPe and back are relatively strong. Kalva et al. (2012) use the level of simulated dopamine to determine the characteristics of neurons in the striatum, STN and GPe. In the striatum, low dopamine causes the indirect pathway neurons to fire strongly in response to an input, whereas high dopamine gives strong firing in the direct pathway. In the STN–GPe loop the dependence on dopamine is such that the nodes are more active under conditions of low dopamine.

Kalva et al. (2012) compare the output of their model to the behaviour of participants in a two-armed bandit task. They see individual differences between participants as different thresholds which control the balance between the activation of the direct and indirect pathway neurons in the striatum under the influence of dopamine. It is not clear to me whether they keep constant dopamine levels or adjust them according to feedback in the task. They compare the fit threshold parameters to the inverse of the temperature when fitting a behavioural softmax model for the action selection stage. Thus they suggest a possible neural implementation which can give output which has some similarity to softmax action selection.

Humphries et al. (2012), using a network based on that of Gurney et al. (2001) with its select and control pathways, proposed that the basal ganglia output at the SNr can be interpreted as a probability distribution for the actions available. They propose that differences in exploration and exploitation can arise from different levels of tonic dopamine. They model the effects of different levels of tonic dopamine on the excitability of two populations of striatal neurons with two types of dopamine receptors, D1 and D2 for each action. These two populations give different responses to the same input. Within their model, the tonic dopamine level is an individual characteristic of subjects carrying out a task. They modelled a two-alternative task, using reinforcement learning to set the input to the model. In this reinforcement learning, all simulated subjects had the same learning rate. Through changes to striatal excitability, the different tonic dopamine levels gave individual differences in response to the task. Interestingly, these differences would, in other circumstances, have been interpreted as resulting from different underlying learning strategies, but in their simulation the underlying reinforcement learning model was identical for all simulated subjects. Humphries et al. (2012) concluded that high levels of tonic dopamine promote exploration and moderate levels

promote exploitation. They see this effect of tonic dopamine as separable from the effects of phasic dopamine, which as described above is thought to encode a prediction error and be involved in long term learning through changing the strengths of connections. They suggest that tonic dopamine may be set from the prefrontal cortex in response to how uncertain the environment is.

Chersi et al. (2013) also consider the different activity levels of the striatum under different dopamine levels to be important in the generation of randomness. They consider a task of learning which button operates which of three lights. They simulate changes in dopamine levels in response to the operation of the correct button. These dopamine changes affect both the excitability of the striatal neurons and the strengths of connections so that learning occurs. In addition to learning a task, they show that their model can also respond to reversals in the environment.

A different aspect of the striatum is proposed by Stocco (2012) to be important in randomness. Stocco (2012) highlights the inhibitory interneurons which are known to exist within the striatum but are rarely included in computational models. He suggests that the amount of exploration could be controlled by a threshold on the activation of the interneurons which then changes the output from the striatum.

Recent studies of computational models of the basal ganglia have considered other aspects of information processing. One area of interest is how different loops within the basal ganglia interact (Baldassarre et al., 2013; Guthrie et al., 2013; N'Guyen et al., 2014; Schroll et al., 2012). In each of these cases, they compare the output of the models to experimental data, so it is also useful to see how they incorporate exploration into these models. Schroll et al. (2012), Baldassarre et al. (2013) and Guthrie et al. (2013) present models which include the direct and hyperdirect, but not the indirect pathways, whereas N'Guyen et al. (2014) uses all three pathways. In all these models, the STN provides a global signal to the GPi/SNr, however, only the model of N'Guyen et al. (2014) includes recurrent connections between the STN and GPe.

Schroll et al. (2012) present two loops through the prefrontal cortex, basal ganglia and thalamus which learn the task. These loops bias a motor loop which produces the response. Schroll et al. (2012) simulate dopamine and use a three factor learning rule, taking account of pre-synaptic and post-synaptic firing rates and dopamine level, to adjust the weights of connections in the basal ganglia, including the connections from the cortex to both striatum and STN and from the striatum to GPi. They state that

exploration in action selection is supported by random terms in the equations governing each neural population. Schroll et al. (2012) demonstrate that their model can flexibly control working memory.

Baldassarre et al. (2013) simulate three loops through the basal ganglia, controlling arm movement, visual attention and focus on goals. They focus on intrinsically motivated learning, that is where there are no goals but learning occurs due to the agent exploring randomly and observing surprising outcomes, which promote dopamine release. The agent is able to benefit from the learning when goals are introduced.

Guthrie et al. (2013) describe a decision making task as having two distinct stages. The first stage is to decide which cue has greater value and the second stage is to prepare a motor action. They show the striatum as having nodes for the cognitive and motor aspects as well as having nodes which combine the two. They describe the exploration phase of the task as being early on before learning has taken place. In this phase, inherent noise in the system means that there is asymmetry between activation corresponding to different actions and so an action will be chosen and the model can learn from the results of its actions.

N'Guyen et al. (2014) investigate the interactions of subcortical and cortical basal ganglia loops for carrying out tasks. The tasks take simulated visual input representing colour and spatial information and the basal ganglia system learns to generate output representing a saccade to the correct position in the visual field such that the saccade is rewarded. Input to the striatum is biased by a reinforcement learning algorithm which is applied separately to the basal ganglia circuit. Exploration in the system before learning has occurred is due to noise which is added to the simulated visual inputs.

Gurney et al. (2015), building on their previous work (e.g. Gurney et al., 2001; Humphries et al., 2006), propose a new way to incorporate plasticity, or changes of connection strengths between populations, into a model of the basal ganglia. Their model of plasticity takes three factors into account: the firing rates of the pre-synaptic and post-synaptic populations, the dopamine level and the type of dopamine receptor, D1 or D2. The changes at each level of these three factors are based on extensive data based on real neurons. Using their plasticity rules, Gurney et al. (2015) investigate the balance needed between the activity of the D1 and D2 neurons in the striatum in order for a decision to be reached. They find that both D1 and D2 neurons, that is both the direct and indirect pathway, need to be active in order to best select an action. This

study forms an important link between models of neural plasticity and models of action selection.

4.4 Conclusions

Although the basal ganglia make up only a small part of the brain, the known anatomical details are complex. The concept of direct and indirect pathways has provided a useful model, but both Nambu (2008) and Calabresi et al. (2014) have identified problems with this simple model. They point out that the two pathways are not completely separated as some neurons in the striatum have been found to connect to both the GPe and the GPi. Nambu (2008) notes that the hyperdirect pathway forms a faster connection from the cortex to the GPi/SNr.

Until now, the main focus on understanding the role of dopamine in the basal ganglia has been on the effects of dopamine on the cortico-striatal connections. The striatum is not the only point in the basal ganglia which is likely to be influenced by dopamine, as dopamine is also transmitted to the GPi, GPe and STN (Nambu, 2008). Of these, the GPe and STN have recurrent connections which can lead to complex firing patterns, but the influence of these on the basal ganglia function is not known.

Regarding computational models of the basal ganglia more generally, as we have already seen, different research groups address different questions. In order to focus on specific aspects of the basal ganglia, researchers have to choose what level of detail to include in their models. This choice includes deciding which pathways and connections between nodes to include and at what level of detail to model the neurons, their interactions and any changes in the system. Several of the models discussed (e.g. Frank, 2006; Gurney et al., 2001) rely on the assumption that the signal from the STN to GPi/SNr is widely spread or even global. Recent work suggests that the connections between the STN and GPi/SNr are highly focussed, although they could be less so than the connections along the direct pathway (Brodal, 2010; DeLong & Wichmann, 2010).

Having produced the computational model described in Chapter 3, I wanted to explore how that model could be implemented neurally. As highlighted by Cohen & Frank (2009) it is unusual for the same researcher to consider both an abstract computational approach and a neural implementation of the same model. In Chapter 5, I introduce the technique of population density modelling of neural systems that I use for the remainder

of this thesis. I will give some background into how population level modelling is built from simplified models of individual neurons.

Before building a model of the basal ganglia in Chapter 6, I show in Chapter 5 some investigations into the STN–GPe loop. Oscillations in the STN–GPe loop are implicated in Parkinson's disease (Weinberger & Dostrovsky, 2011) leading to studies of computational models of the STN–GPe loop in isolation (e.g. Merrison-Hort & Borisyuk, 2013; Nevado Holgado et al., 2010). I show my results obtained using population density modelling can be related to those using other techniques.

Chapter 5

Population Density Models

5.1 Introduction

In this chapter I introduce the method of population density modelling for neural systems, which I use in the remainder of this thesis. I give a brief outline of the underlying theory for population density modelling and a comparison to other styles of computational modelling of groups of neurons. I show that population density modelling can be used to replicate and build on research based on other modelling styles.

In order to introduce population density modelling, I first give an overview of the underlying simplification of the properties of neurons and their interactions which is used. Neurons produce electrical signals called action potentials, a process also known as firing. The action potential is a large and rapid change in voltage between the inside and the outside of the neuron, the membrane potential. This electrical signal can travel very rapidly between neurons, forming a method of information transfer. A message is passed from one neuron to another at a junction called a synapse. Each individual action potential from a neuron has the same electrical properties, information is transmitted using the timings of the action potentials, that is the pattern of firing. In this thesis, an action potential is taken to be an instantaneous event after which the neuron cannot fire again for a set time period, the refractory period.

5.2 Leaky integrate and fire neurons

Here I introduce a commonly used model of neuronal activity, the leaky integrate and fire model, which also forms the underlying model for the neural simulations I

present. The leaky integrate and fire model was introduced by Lapicque (1907) (available in translation, Brunel & van Rossum (2007)) and allows modelling of the membrane potential of a neuron.

Without giving any details of the chemical properties, a neuron has a membrane which forms a lipid bilayer through which ions can pass. There will be different concentrations of different ions on the inside and the outside of the neuronal cell. At equilibrium, the resting membrane potential, the forces on the ions due to the different concentrations will be balanced by a potential difference across the membrane. Figure 5.1 gives a schematic illustration of the changes in membrane potential, with the initial state being the resting potential.

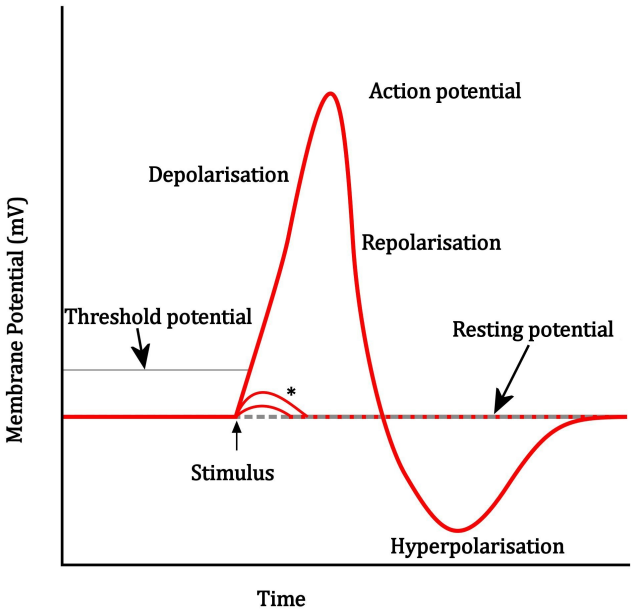


Figure 5.1: Schematic diagram of an individual neuronal action potential.

If a small temporary increase in voltage is externally applied across the membrane, the membrane potential will afterwards decay back to its equilibrium potential. Figure 5.1 shows this with an increase in voltage after the stimulus is applied and the lines shown by * decaying back to the resting potential. When a large enough voltage is applied so that the membrane potential exceeds a threshold, the neuron is said to become active. The voltage rises very quickly with no additional external input. This

occurs due to changes in the properties of the membrane allowing different ions to pass through. The neuron is said to have fired, or an action potential to have been generated. The membrane potential then decreases and dips below the resting potential, this dip is known as hyperpolarisation. These features are shown in Figure 5.1. The resting potential is typically -70 mV, so the increase in membrane potential is called depolarisation, and the peak of the action potential at about 40 mV.

The leaky integrate and fire model simulates changes in membrane potential when below threshold. The action potential can be assumed to be an instantaneous spike after which the membrane potential is reset to a reset potential after a fixed time, the refractory period. For simplicity, I will assume that the reset potential is the same as the resting potential. The potential difference between the inside and outside of a neuron can be considered as forming an electrical circuit containing a resistor and a capacitor, as illustrated in Figure 5.2. In Figure 5.2 the top and bottom represent the outside and inside of the neuron respectively, so the potential difference, V, is that across the cell wall, that is the membrane potential.

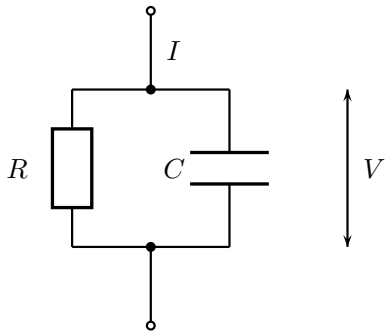


Figure 5.2: RC circuit representation of the membrane of a neuron, used in the leaky integrate and fire model.

To derive the equation for the change in membrane potential of a leaky integrate and fire neuron using the circuit diagram Figure 5.1, consider V to be the membrane potential at an instant in time, then

$$I = I_R + I_C \tag{5.1}$$

where I is the external current applied to the neuron, I_R is the leak current through the resistor which will act to restore the potential to the resting value, V_r, and I_C is the current due to the charging of the capacitor. Using Ohm's Law

$$I_R = \frac{V - V_r}{R} \tag{5.2}$$

From the definition of a capacitor

$$Q = VC \tag{5.3}$$

where Q is the charge, V is the voltage across the capacitor and C is the capacitance. Current is the rate of change of charge so

$$I_C = C\frac{dV}{dt} \tag{5.4}$$

Substituting equations 5.2 and 5.4 into equation 5.1 gives

$$I = \frac{V - V_r}{R} + C\frac{dV}{dt} \tag{5.5}$$

Rearranging and setting $\tau = RC$, this constant being known as the membrane time constant of the neuron, gives

$$\tau\frac{dV}{dt} = -(V - V_r) + RI. \tag{5.6}$$

Equation 5.6 gives the dynamics of the membrane potential of a neuron under the leaky integrate and fire model while the membrane potential is below the threshold. This model does not incorporate the action potential, or spike, itself. The timing of a spike is given by the time at which the membrane potential, V reaches a threshold. When this has occurred, and possibly after a period of delay, the absolute refractory period, the membrane potential, V, is set to the reset potential. If a constant current is applied, which is high enough to cause the neuron to spike, it will continue to do so at regular intervals. Rather than merely recording the timing of each spike, it is usual to calculate the mean firing rate by averaging the number of spikes over a time window.

5.3 Modelling groups of neurons

Neurons do not operate in isolation, a human brain has many billions of neurons each of which may have connections from many thousands of others. There is structure to the organisation of neurons such that local populations of neurons have similar properties.

One way to model connected populations of neurons is to model each neuron individually, including modelling the changes in membrane potential when individual action potentials are propagated from one neuron to another. Modelling many neurons this way is computationally expensive and requires many parameters to be specified. Chersi et al. (2013) and Humphries et al. (2006), discussed in Chapter 4, use this style of modelling. Each neuron in the model produces spikes and it is usual to average the number of spikes over time for a group of similar neurons which form a connected population, as did Chersi et al. (2013).

An alternative to modelling each individual neuron separately is to use a firing rate neuron model. In this case a simulated node represents a population of neurons and the firing rate of the node is modelled directly. One such model is given by

$$\tau \frac{dv}{dt} = -v + F(I), \tag{5.7}$$

where v represents the firing rate of the node, τ is a time constant, I is the total input to the node and $F(I)$ is an activation function (Wilson & Cowan, 1972). The function $F(I)$ is often taken to be a sigmoid function. The constant τ determines the time scale at which the firing rate changes in response to inputs, but this is not the same as the membrane time constant in the leaky integrate and fire model. Although Equation 5.7 looks very similar to Equation 5.6 they are very different. Equation 5.6 describes the membrane potential of an individual neuron and has a discontinuity as the neuron fires and the membrane potential has to be reset. Equation 5.7 describes the continuous dynamics of the firing rate of a population of neurons.

If we assume that the time scale for changes to the inputs is much shorter than the time constant for the firing rate, then the total input I can be described as the weighted sum of the input firing rates from connected nodes.

Examples of computational basal ganglia models, described in Chapter 4, which use this style of modelling of populations of neurons are; Baldassarre et al. (2013), Guthrie et al. (2013) and Gurney et al. (2015).

5.4 An introduction to population density modelling

An alternative approach to modelling populations of neurons is population density modelling which models the states of a collection of neurons using probability distributions, (Knight, 1972; Omurtag et al., 2000; Stein, 1965). Using the approach presented in Omurtag et al. (2000), I describe how a population of leaky integrate and fire neurons is modelled. In this case the state of a neuron is considered to be the membrane potential and population density modelling allows us to consider changes to the distribution of membrane potentials over time.

Before building the population density model, we need to consider some aspects of simulating a population of identical individual neurons in a connected population. The change in membrane potential for each neuron in the system is governed by Equation 5.6. We can rescale the membrane potential, V, so that it is between 0 and 1 where the rescaled membrane potential v is given by

$$v = \frac{V - V_r}{V_T - V_r} \tag{5.8}$$

where V_T is the threshold potential and, as before, V_r is the resting potential. Now Equation 5.6 can be re-expressed as follows

$$\frac{dv}{dt} = -\gamma v + s(t), \quad 0 \leq v \leq 1, \tag{5.9}$$

where

$$\gamma = \frac{1}{\tau}, \quad s(t) = \frac{I}{C(V_T - V_r)} \tag{5.10}$$

We can use $v^j(t)$ to denote the rescaled membrane potential of neuron j of our population at time t. Then each neuron's membrane potential changes over time according to

$$\frac{dv^j}{dt} = -\gamma v^j + s^j(t), \tag{5.11}$$

where there is no difference between each neuron in terms of the leak current, as we have taken them to be identical, but the current to each neuron depends on the other neurons connected to it. We are considering individual spiking neurons, and as described above we can create a list of the firing times for each neuron, so let t_n^l denote the set of firing times of neuron l from the population. Each spike is of identical strength $\hat{s}$ and so the

change in each neuron that l connects to driven by the spikes from l can be expressed as

$$s_l(t) = \hat{s} \sum_n \delta(t - t_n^l), \tag{5.12}$$

where $\delta(t - t_n^l)$ is the Dirac delta function which is zero everywhere except at $t = t_n^l$ where there is a spike. If we know all the connections in the population then we can use Γ^j to denote the set of indices of all the neurons that provide an incoming connection to neuron j. We can then use Equation 5.12 and calculate the change in neuron j due to all the incoming neuronal spikes as

$$s^j(t) = \sum_{l \in \Gamma^j} s_l = \hat{s} \sum_{l \in \Gamma^j} \sum_n \delta(t - t_n^l). \tag{5.13}$$

Suppose that when setting up the population we connect each neuron to G others on average, where the connections are chosen randomly. Suppose also that external input to the population is Poisson distributed and spikes arrive at times t^0.

Now we can start to build the population density model by considering a probability density, $\rho(v, t)$ of membrane potentials over time. Suppose we simulate a population of P connected leaky integrate and fire neurons and do so N times giving N replica simulations. Each time we will generate a new randomly chosen set of connections. This is the approach taken by Omurtag et al. (2000). Nykamp & Tranchina (2000) develop the population density equation using a different approach, using a large population with all to all connections but with randomness in the arrival times of synaptic inputs and in the size of the change in membrane potential when a neuron receives a single spike.

Following the approach of Omurtag et al. (2000), if dv is a small interval of the scaled membrane potential, then we can use $n_k(v, t)dv$ to represent the number of neurons having a membrane potential in dv in the kth replica simulation and define a number density at a membrane potential v as the average number of neurons having membrane potential v over all N of our replica simulations.

$$n(v, t) = \langle n_k \rangle = \lim_{N \to \infty} \frac{1}{N} \sum_k n_k(v, t). \tag{5.14}$$

This number density can be converted into the required probability density, $\rho(v, t)$, by normalising by the number of neurons in each replica population, P, as follows

$$\rho(v, t) = \frac{n(v, t)}{P}. \tag{5.15}$$

As stated above, the average firing rate of a neuron is calculated by summing the spikes emitted over a time window. Using t_n^m to denote the firing times of neuron m, we can calculate the average firing rate per neuron in the population as follows

$$r(t) = \frac{1}{P} \lim_{\Delta t \to 0} \Big\langle \frac{1}{\Delta t} \int_t^{t+\Delta t} dt \sum_{m \neq 0} \sum_n \delta(t - t_n^m) \Big\rangle. \tag{5.16}$$

The part of the incoming firing rate due to external sources, $\sigma^0(t)$, can be similarly expressed as

$$\sigma^0(t) = \lim_{\Delta t \to 0} \Big\langle \frac{1}{\Delta t} \int_t^{t+\Delta t} dt \sum_n \delta(t - t_n^0) \Big\rangle. \tag{5.17}$$

Note that each neuron is connected to G others on average, so the average rate of receipt of incoming impulses, $\sigma(t)$, can be expressed as

$$\sigma(t) = \sigma^0(t) + Gr(t). \tag{5.18}$$

The population density $\rho(v, t)$ gives the distribution of neurons across all possible membrane potentials at time t and $\rho(v, t)dv$ represents the probability of a neuron having a membrane potential in $(v, v + dv)$.

To derive the population density equation we need to consider how $\rho(v, t)$ changes over time. We can use $J(v, t)$ to denote the flux of probability across v at time t. Consider an interval (a, b) of the membrane potentials and the probability within that interval, then $\rho(v, t)dv$ will change due to the flux across a and b.

$$J(a, t) - J(b, t) = \frac{\partial}{\partial t} \int_a^b \rho(v', t) dv'. \tag{5.19}$$

Now let $b = v$ and differentiate by v to give

$$\frac{\partial \rho}{\partial t} = -\frac{\partial J}{\partial v}. \tag{5.20}$$

We are using a rescaled membrane potential such that $v < 1$ with firing occurring when v reaches 1 and the neuron being reset to have $v = 0$, here we use the simplification that the reset potential is the same as the resting potential. Equation 5.20 does not account for the firing and reset of neurons and can be modified as follows

$$\frac{\partial \rho}{\partial t} = -\frac{\partial J}{\partial v} + \delta(v) J(1, t). \tag{5.21}$$

We now need to describe the flux J which is caused by neuronal dynamics due to the leak and to inputs via synapses. Without synaptic inputs, Equation 5.9 becomes as follows

$$\frac{dv}{dt} = -\gamma v \tag{5.22}$$

Now suppose a neuron having membrane potential in $(v, v + \Delta v)$ crosses v during the time interval $(t, t + \Delta t)$ where Δt is short and

$$\Delta v = \frac{dv}{dt}\Delta t + O(\Delta t^2). \tag{5.23}$$

Then

$$\rho(v,t)\Delta v = \rho(v,t)\frac{dv}{dt}\Delta t + O(\Delta t^2). \tag{5.24}$$

The flux of probability is the change over unit time. Using $J_l(v,t)$ to represent the flux due to leak, we can divide Equation 5.24 by Δt to give

$$J_l(v,t) = \rho(v,t)\frac{dv}{dt} + O(\Delta t). \tag{5.25}$$

Let $\Delta t \to 0$ so that we can ignore the $O(\Delta t)$ term and combine with Equation 5.22 to give

$$J_l(v,t) = -\gamma v \rho(v,t). \tag{5.26}$$

Now we need to consider changes to the population density due to neurons receiving input spikes. Each time a neuron receives an input spike its membrane potential will increase by a fixed amount, h.

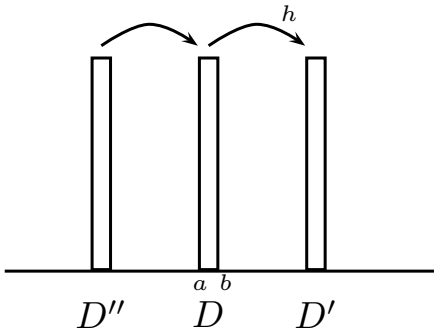


Figure 5.3: Considering the population density as a histogram of membrane potentials, neurons in D will move to D' on receiving an impulse spike.

Consider the part of the membrane potential shown as D in Figure 5.3. Neurons which receive an incoming spike will leave D and move to D'. Also, D will receive those neurons from D'' which receive an impulse. The region D is small but arbitrary and the rate of incoming impulses is $\sigma(t)$ so we can consider the probability flux across a membrane potential v due to synaptic input, $J_s(v,t)$, as

$$J_s(v,t) = -\sigma(t) \int_{v-h}^{v} \rho(v',t)dv'. \tag{5.27}$$

Now we can take Equations 5.25 and 5.27 and insert into Equation 5.21 to give the population density equation as

$$\frac{\partial \rho}{\partial t} = \gamma \frac{\partial v\rho}{\partial v} + \sigma(t)(\rho(v-h) - \rho(v)) + \delta(v)J(1,t) \tag{5.28}$$

with the boundary condition of $\rho(1,t) = 1$ as there cannot be any density above the threshold.

In Equation 5.28, $J(1,t)$ is the flux across the threshold potential of 1 and is equal to the firing rate of the population which is given by

$$r(t) = -\sigma(t) \int_{1-h}^{1} \rho(v',t)dv'. \tag{5.29}$$

The software Miind (de Kamps et al., 2008) provides the functionality to solve Equation 5.28 and allow simulation of large populations of leaky integrate and fire neurons.

5.5 Using population density modelling

The software Miind (de Kamps et al., 2008) allows the specification of populations of neurons and the connections between them in order to simulate neural systems. Incoming spikes are considered to be generated by a Poisson process and so the actual arrival times for spikes are stochastic. It is assumed that within a neural population, each neuron receives a large number of individual synaptic impulses, but that each incoming spike makes only a small change to the membrane potential of the neuron. Given these conditions, the input to a neural population can be simulated using Gaussian white noise (Amit & Brunel, 1997; de Kamps, 2006; de Kamps et al., 2008).

To create simulations using Miind, background input can be specified by a constant or variable firing rate. Populations are connected to each other and to receive background input by specifying a number of effective connections, N, and an efficacy h where

the efficacy represents the increase in membrane potential when an individual neuron receives a single incoming spike. It is important to ensure, when specifying connections, that the efficacy is small in comparison to the threshold potential so that the assumption required to use Gaussian white noise holds. Suppose that an incoming connection with firing rate ν is connected using the parameters N and h to a population which has a membrane time constant τ then the mean μ and standard deviation σ for the Gaussian white noise approximation are then given as follows (Amit & Brunel, 1997):

$$\begin{aligned} \mu &= \tau h N \nu, \\ \sigma^2 &= \tau h^2 N \nu \end{aligned} \tag{5.30}$$

I now show some simple simulations created using Miind (version 0.09). The plots shown in this section use populations with the following settings, the threshold membrane potential is set to 0.02 V, the reset and the resting potential both set to 0 V, the membrane time constant 0.02 s and a refractory period of 0.002 s. Each simulation is started with each entire population having a membrane potential equal to the reset potential, that is 0 V. In each case, efficacies stated refer to a percentage of the difference between the reset and the threshold potential, that is a percentage of 0.02 V.

Firstly, I make some observations from the simplest type of simulation available, that of isolated populations which receive only background input and no connections from other populations. Figure 5.4 shows the results of two separate simulations each of which have a steady background input of 1.4 spk/s which excites a single node using the parameters shown in Table 5.1. Using Equation 5.30 above, it is easy to verify that the mean of the stochastic input is the same in each case, but the top of Figure 5.4 shows a higher variance in the input than the bottom.

Node	Number of Connections	Efficacy
Top	4000	1.6
Bottom	10000	0.64

Table 5.1: Connections from steady background for the two simulations shown in Figure 5.4.

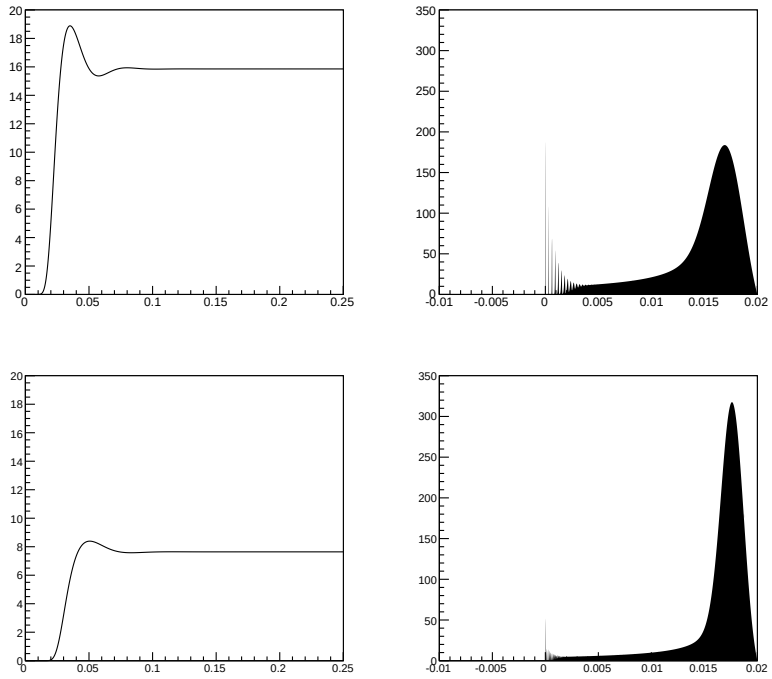


Figure 5.4: Left: Firing rate with time on the x-axis and firing rates in spk/s on the y-axis. Right: Population density at steady state the x-axis shows the membrane potential with the threshold potential at the right. The top and bottom simulations result from the parameters shown in Table 5.1.

Figure 5.4 shows on the left that, for each simulation, the firing rate increases from zero, reaches a maximum, dips and then settles to a steady firing rate. Although the mean input is the same, the steady firing rates are different. When the variance in the stochastic input is higher, the population firing rate at the steady rate is also higher. The population densities on the right of Figure 5.4 show the distribution of membrane potentials in the population with the threshold membrane potential, 0.02 V, at the right of the plot. Where the variance in the connections is greater we also see a wider peak in the membrane potential distribution. The peaks at 0 V show the neurons which have been re-entered into the population after firing at the reset potential.

I now show the output from populations where background input is fed to one neural population which then excites a second neural population as shown in Figure 5.5. In the following simulations, both populations receive input from a steady background of

Figure 5.5: Simple simulation in which node A excites node B in addition to both nodes receiving excitatory background input.

1.5 spk/s, Table 5.2 shows the parameters used to connect the background input to the two nodes. Node A receives higher mean input from the background than node B as node B will also receive excitatory input from node A.

Node	Number of Connections	Efficacy
A	10000	0.64
B	62.5	4

Table 5.2: Connections from steady background for the nodes in Figure 5.5.

Figure 5.6 shows the output of node A from the background connection shown in Table 5.2. Node A starts firing after 0.01 s and the firing rate rapidly rises to reach a peak of approximately 29 spk/s between 0.03 and 0.04 s. The firing rate then dips to approximately 17 spk/s between 0.05 and 0.06 s before settling to a steady firing rate of approximately 20 spk/s. I use spk/s when referring to the firing rate of a population at a particular time and I will use Hz when describing oscillations in the population firing rate. Note that although the specified connections to node A from the background input

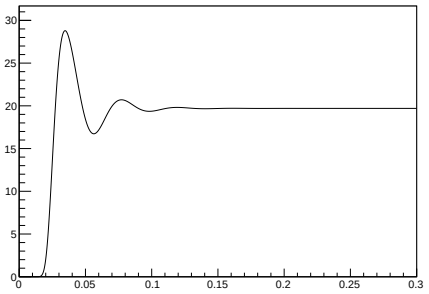


Figure 5.6: Output at node A for the network shown in Figure 5.5.

are the same as for the bottom of Figure 5.4, the output is different as the background firing rate is now higher.

The same output from node A is used for the two tests described below with different connections between A and B. The output from node A excites node B with a delay of 0.1 s. Figure 5.7 shows the output of node B with the two sets of connection parameters shown in Table 5.3.

Test	Number of Connections	Efficacy
1	2500	0.18
2	125	3.6

Table 5.3: Two sets of connection parameters from node A to node B in the simple simulation illustrated in Figure 5.5. The two tests give the same mean input but test 2 has a higher variance in the input.

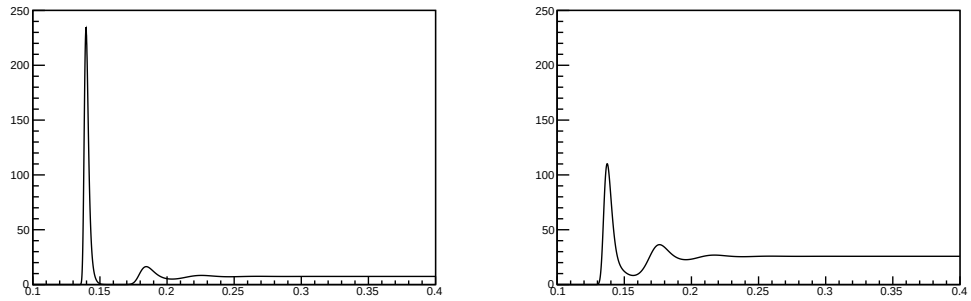


Figure 5.7: Output at node B for the tests in Table 5.3. Left: Test 1. Right: Test 2.

The two sets of connection parameters from node A to node B have the same value for the product of efficacy and number of connections, so as everything else in the simulations are identical and using Equation 5.30, the mean input to node B is identical in each case. The different efficacies mean that test 2 has higher variance in the signal feeding node B. This higher variance gives a lower initial peak firing rate but a higher steady state firing rate as seen in Figure 5.7.

Creating simulations using Miind, I can visualise the population density as it changes over time, not only at the steady state as was shown in Figure 5.4. Figure 5.8 shows

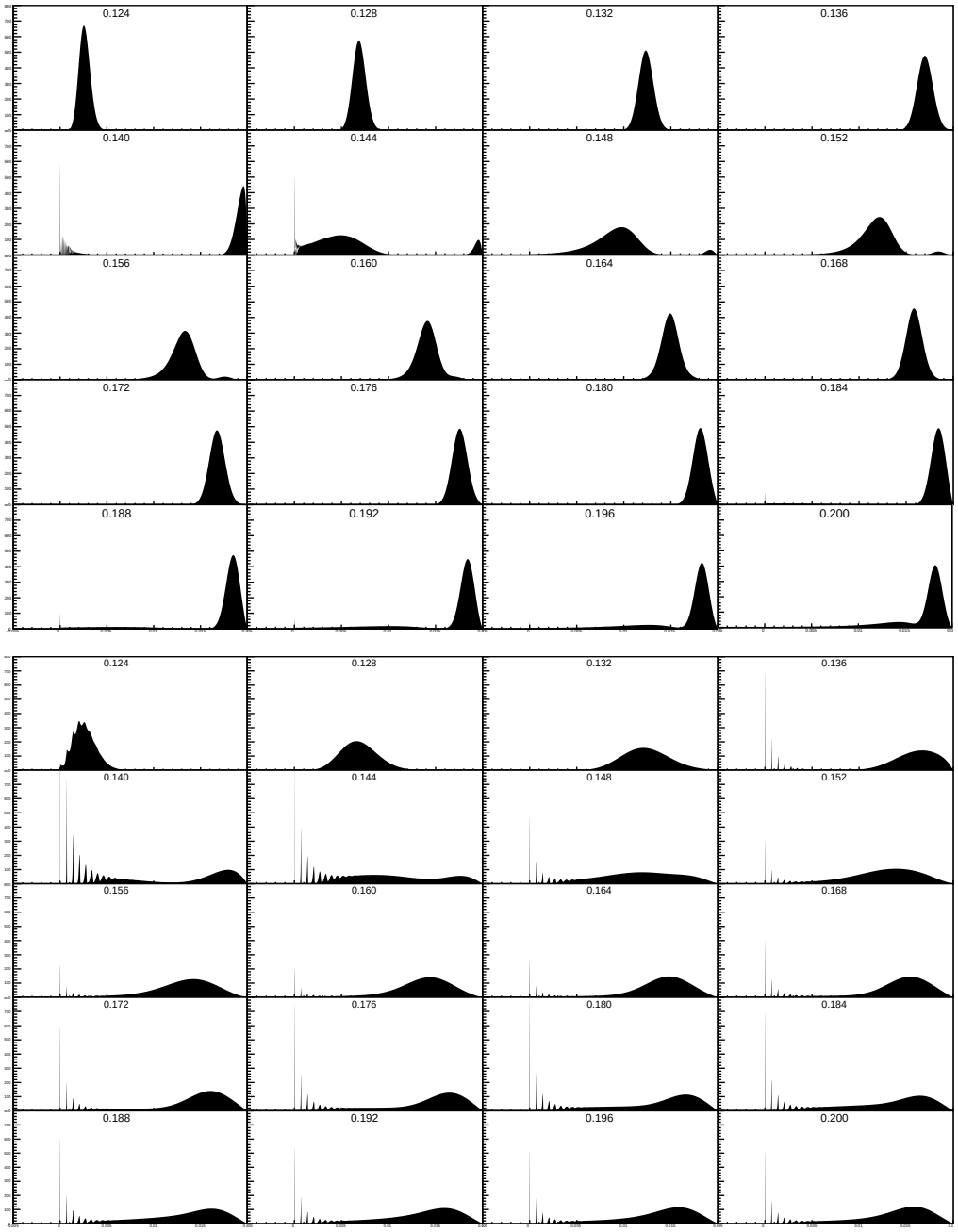


Figure 5.8: Development of population densities over time for node B in Figure 5.5. Top: Test 1. Bottom: Test 2.

the evolution of the population densities for node B in the two tests, corresponding to the firing rates shown in Figure 5.7. The population densities start from 0.124 s of simulated time and are shown at intervals of 0.002 s, reading from left to right and top to bottom with the times shown at the top of each sub-plot. Each sub-plot has the threshold potential on the right hand side, so as the population crosses this threshold, the node will fire and that portion of the population which has fired will be reset after a delay of 0.002 s. These population densities respond to the input from node A with a delay of 0.1 s.

Considering test 1, shown at the top of Figure 5.8, at 0.136 s the membrane potentials of the population have risen towards the threshold in response to the strong input from node A and node B starts to fire. At 0.14 s, some of the population has been re-introduced at the reset potential and there is still a large part of the population being pushed over the threshold. From 0.144 to 0.148 s the membrane potential of that part of the population which has already fired is increasing again and only a small part of the population has not fired, having a membrane potential close to the threshold. By 0.152 s the population has stopped firing, the firing rate of node A has dipped and is no longer enough to push the membrane potential of population B over the threshold. At 0.156 s it can be seen that the part of the population which has not fired now has lower membrane potentials. As the output of node A increases, the membrane potentials of node B increase again so that at 0.176 s node B is firing again.

In test 2 shown at the bottom of Figure 5.8, the population membrane potentials are more spread and at 0.136 s the population has started firing and some of the population has been reset. In this test, once node B has started firing it does not stop firing again, shown by the fact that the population density is always touching the threshold potential.

5.6 Excitatory-inhibitory circuit

I now consider a slightly more complex system again with two neural populations but this time with a recurrent connection, that is each neural population is connected to each other. In this model, one population produces excitatory output and the other inhibitory. The excitatory connection remains as in the simple simulation described above. Now I introduce, in addition, an inhibitory connection back to the excitatory

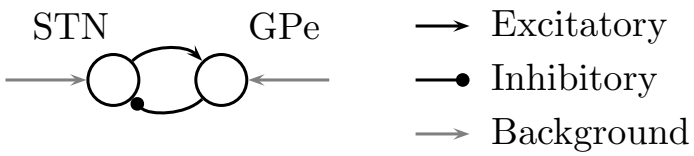


Figure 5.9: Neural system used to investigate the behaviour of an excitatory-inhibitory circuit. Nodes are labelled as parts of the basal ganglia which form such a circuit.

node as shown in Figure 5.9. Although the circuit still appears very simple with just two nodes, adding this additional connection allows complex output to be produced.

This neural circuit can be used to represent the STN–GPe loop within the basal ganglia. As described in Chapter 4, the STN–GPe connections have been included in computational models of the basal ganglia when studying learning and decision making and in connection to randomness of decision making (e.g. Kalva et al., 2012). Neurons in the STN–GPe circuit are known to exhibit unusual oscillatory behaviour in Parkinson's disease prompting several computational studies of the circuit in isolation (e.g. Kumar et al., 2011; Merrison-Hort & Borisyuk, 2013; Nevado Holgado et al., 2010; Pavlides et al., 2012).

Nevado Holgado et al. (2010) analysed a simple mathematical representation of the STN–GPe circuit in order to describe conditions which cause beta oscillations, that is oscillations in the range 13 to 30 Hz. They compared their analytical conclusions to firing rate based neural simulations created using firing rate based modelling techniques as described in Section 5.3. I show how population density modelling allows the same conditions to be observed. Following that, I demonstrate some results which have not been demonstrated using firing rate based models.

Nevado Holgado et al. (2010) set the weights in their model in order to produce responses matching those seen experimentally. They show the results of a simulation of a 'healthy' STN–GPe circuit as having a peak firing rate of approximately 110 spk/s for the STN and 30 spk/s for the GPe. In this simulation the oscillations are quickly damped and steady firing rates of 55 spk/s and 20 spk/s are shown for the STN and GPe respectively. For parameter settings to mimic Parkinson's disease, they show continuing oscillations at a rate of 20 Hz with a larger amplitude for the GPe than for the STN.

Note that the node I refer to as the GPe is labelled GP by Nevado Holgado et al. (2010). The weights used in a rate based model cannot be used directly in a population based model, so I tested various combinations of parameter settings to approximate the output shown by Nevado Holgado et al. (2010).

For the simulations described in this section, the threshold membrane potential, reset potential and refractory period are as used in Section 5.5 above. Now the membrane time constants are set to different values for the two populations, having 6 ms for the excitatory neurons, STN, and for the inhibitory population, GPe 14 ms, these values are as used by Nevado Holgado et al. (2010). The simulations begin with the whole of each population having a membrane potential equal to the reset potential. Background input is applied to each node and after 0.1 s of simulation, additional background input is applied to the STN node, in the presentation of parameters this is shown as having a delay. The first 0.1 s of simulated time are used to allow the population density profiles to approach their steady state before a stimulus is applied.

The first condition given by Nevado Holgado et al. (2010) for oscillations to occur is that the strength of the connections between the two populations have to be strong enough, both from the STN to GPe and the GPe to STN. To demonstrate this, I used the connections shown in Table 5.4 from steady background of 1.8 spk/s each set with 3000 effective connections. As in Section 5.5 above, efficacies are shown as a percentage of the threshold membrane potential.

Node	Efficacy	Delay (s)
STN	2	0
GPe	1	0
STN	0.9	0.1

Table 5.4: Connections from a steady background of 1.8 spk/s to the two nodes.

Figure 5.10 shows the effect of changing the strengths of the connections from the STN to GPe and the GPe to STN when running simulations with background connections as in Table 5.4. In each case the connections between the STN and GPe incur a 6 ms delay in each direction and there are 1250 effective connections. The first 0.1 s of simulation allow the population densities of both nodes to take on a normal distribution

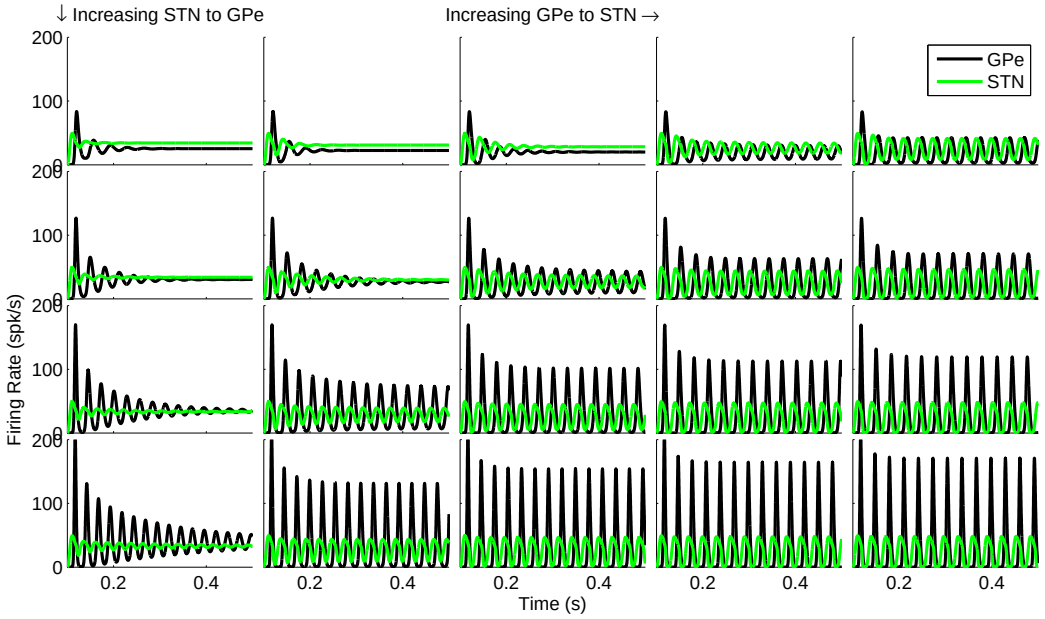


Figure 5.10: The effect of increasing the connection efficacies from the STN to GPe and the GPe to STN. Each row has a single value for the efficacy of the connection from the STN to GPe. Each column represents a single setting for the GPe to STN connection.

using background input which is too low to cause firing. The additional input to the STN at 0.1 s causes the STN to start to fire, starting the interactions in the system. In each case these steady background inputs continue for the remainder of the simulation. Without the first 0.1 s to allow the population densities to develop, all neurons in each population start with the same membrane potential. This is not a biologically realistic situation and in terms of the simulations, causes higher amplitude and longer oscillations in the firing of the STN. This process of the background input being introduced at separate times is used throughout the remainder of this chapter.

The top left plot in Figure 5.10 shows the results of setting an efficacy of 0.05 from the STN to GPe and -0.005 from the GPe to STN, where a negative efficacy indicates an inhibitory connection. Each row in Figure 5.10 shows the effect of keeping the STN to GPe connection constant and increasing the magnitude of the efficacy of the connection GPe to STN by 0.01, as this is an inhibitory connection this gives an efficacy of -0.045 for the rightmost column. Each column shows the effect of increasing the efficacy of the connection from the STN to GPe by 0.01 each time with 0.08 for the bottom row.

In Figure 5.10, the green line shows the activation of the STN and the black line that of the GPe. These colours are used throughout this section but will not be explicitly labelled each time.

Figure 5.10 shows a transition from reaching a steady state to sustained oscillations as the strengths of connections from the STN to GPe and the GPe to STN are both increased. Examining the bottom right plot in Figure 5.10, we can estimate the frequency of the oscillations to be about 33 Hz. Comparing the plots in the bottom row, we see that increasing the strength of connection from the GPe to STN reduces the frequency of oscillations. Looking at the right-hand column, we see that increasing the connection strength from the STN to GPe increases the frequency of oscillations.

The second condition found by Nevado Holgado et al. (2010) is that for oscillations to occur the transmission delay between the STN and GPe nodes has to be high in comparison to the membrane time constants for those nodes. In their mathematical analysis, Nevado Holgado et al. (2010) used the same membrane time constant for each node, whereas I have used the different membrane time constants they used in their simulations.

Connection	Number of Connections	Efficacy
STN to GPe	1250	0.06
GPe to STN	1250	−0.03

Table 5.5: Parameters used to test changes to the delay in transmission between the two nodes, shown in Figure 5.11.

Figure 5.11 shows the effect of increasing the transmission delay between the STN and GPe, keeping the same delay in each direction, from 1 ms and increasing by 1 ms whilst keeping all other parameters constant. The simulations were set up with background input as given in Table 5.4 and connections between the nodes as given in Table 5.5. For small transmission delays, the oscillations die out. As the delay increases, oscillations persist and the amplitude and period of the oscillations increases.

Figure 5.12 shows the effect of altering the efficacies of the connections from the background input to the STN and GPe. I show two levels of background input, described as high and low, to each of the nodes, details of the parameter settings are in Table 5.6.

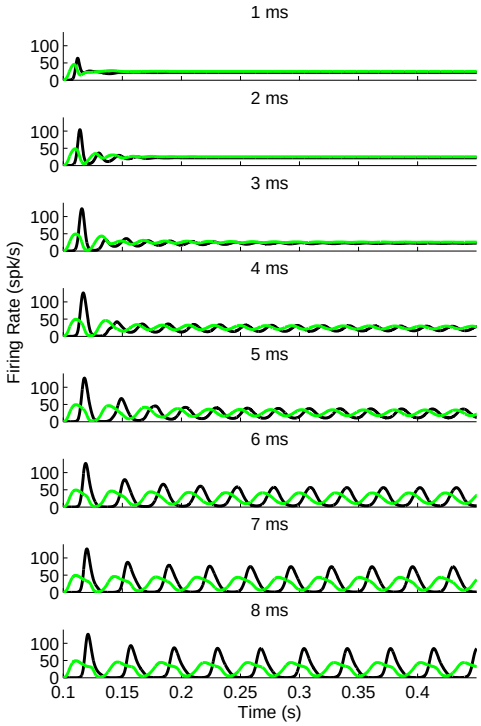


Figure 5.11: Effect of increasing the delay in the connection between the two nodes.

The top plots have the low connection from the background to the STN and the bottom the high connection. The left plots have the low connection to the GPe. Increasing the efficacy of the connection from the background to either the STN or GPe increases the amplitude and the frequency of the oscillations. Nevado Holgado et al. (2010) report that to produce sustained oscillations, the excitatory input from the cortex to the STN must be high in comparison to the inhibitory input from the striatum to the GPe. To keep my model as simple as possible, I have implemented excitatory input from a steady background as the only input external to the STN–GPe circuit. Considering a lower excitatory background input to the GPe as equivalent to a higher inhibitory input, my results are in line with those of Nevado Holgado et al. (2010).

So far, the simulations shown have not included self-inhibition in the GPe node although such self-inhibition is included by Nevado Holgado et al. (2010). I now add that connection and observe the effect of increasing the strength of the self-inhibition of the GPe.

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	3000	2	0
Background to GPe low	3000	0.7	0
Background to STN low	3000	0.8	0.1
Background to GPe high	3000	1	0
Background to STN high	3000	0.9	0.1
STN to GPe	1250	0.1	0.006
GPe to STN	1250	−0.02	0.006

Table 5.6: Parameters used to test the effect of different input connections to the two nodes, giving the results shown in Figure 5.12.

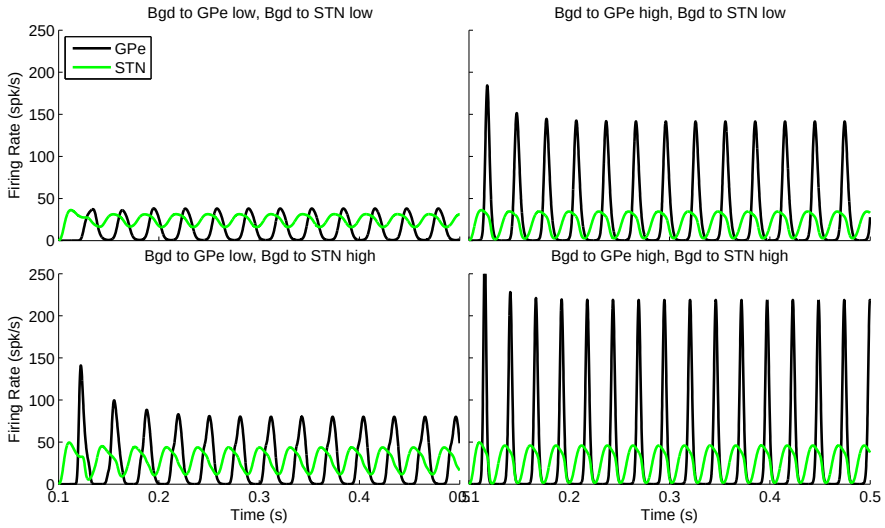


Figure 5.12: Comparison of different connections from the input to the STN and GPe nodes.

Figure 5.13 shows the effect of increasing the strength of an inhibitory connection from the GPe to itself. The top simulation was produced using the parameters given in Table 5.7 the magnitude of the self-inhibition of the GPe was increased by 0.005 for each of the two following simulations whilst keeping all other parameters constant. As would be expected by adding additional inhibition, the maximum firing rate of the GPe node decreases as the self-inhibition increases. In addition, the frequency of the

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	3000	2	0
Background to GPe	3000	1	0
Background to STN	3000	1	0.1
STN to GPe	1250	0.1	0.006
GPe to STN	1250	−0.03	0.006
GPe to GPe	1250	−0.01	0.006

Table 5.7: Parameters for the initial test of adding self-inhibition to GPe, giving the top simulation in Figure 5.13.

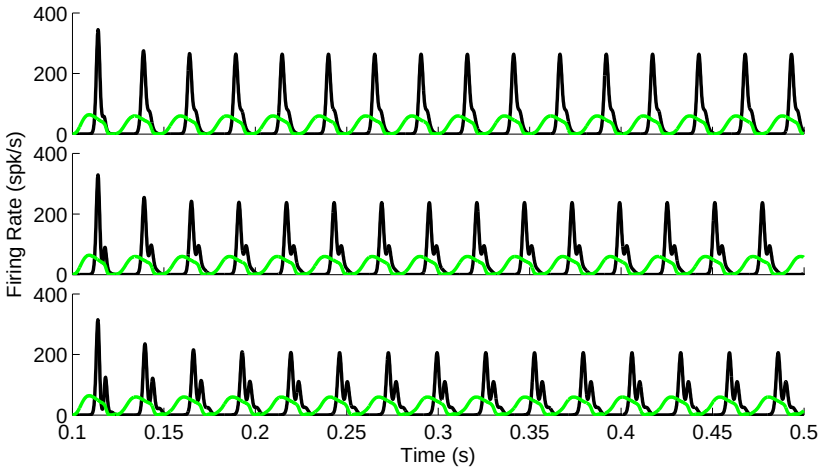


Figure 5.13: Effect of increasing self-inhibition in the GPe node.

oscillations also decreases, as we see fewer peaks in the same time period in the bottom of Figure 5.13 than the top.

Above, I have indicated that the conditions for oscillations to occur as detailed by Nevado Holgado et al. (2010) can be demonstrated using a population density modelling technique.

Figure 5.13 shows that the structure of the oscillations can become more complex as self-inhibition in the GPe node increases as we see that the firing rates of the GPe node do not form a single peak during one period of oscillation. I now describe some investigations into this kind of pattern of firing rate changes.

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	10000	0.55	0
Background to GPe	10000	0.4	0
Background to STN	10000	0.3	0.1
STN to GPe	10000	0.006	0.006
GPe to STN	10000	-0.00325	0.006
GPe to GPe	10000	5×10^{-7}	0.004

Table 5.8: Parameters for Figure 5.14 which give a split in the peak of the oscillating firing of the GPe as shown in Figure 5.14.

Figure 5.14 shows another example of oscillatory behaviour in which there are two peaks in the firing rate of the GPe node within each period of oscillation. These oscillations are generated using the parameters given in Table 5.8 and continue with an unchanging pattern, at least as far as 10 s of simulated time. Note that in this simulation, there is a very small self-excitation of the GPe node, I found by experiment that this allowed a wider range of the other parameters to show splits in the peak firing rates.

Figure 5.15 shows the firing rates of the two nodes, from the same simulation as in Figure 5.14 focussing on one period of oscillation starting from 1 s of simulation time. As I used a population density technique for this simulation, we can examine the distribution of membrane potentials during the simulation. Figure 5.16 shows the

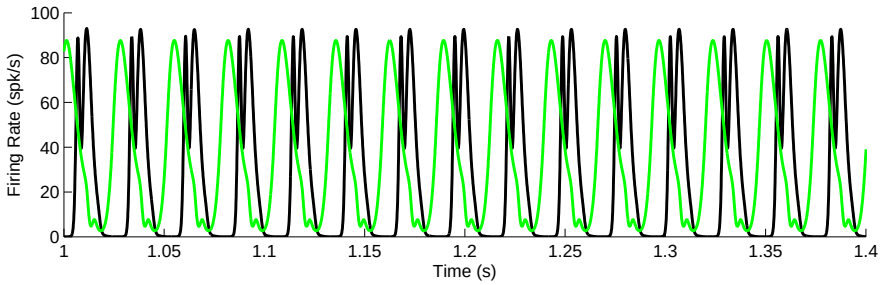


Figure 5.14: Oscillations with a double peak in the firing rate of the GPe node, simulated using the parameters in Table 5.8.

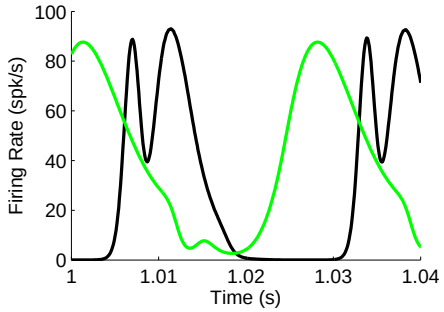


Figure 5.15: Detail of firing rates from Figure 5.14.

change in distribution of membrane potentials for the GPe node starting from 1 s of simulation time as in Figure 5.15.

Looking at Figure 5.16, it is immediately clear that the membrane potentials are not normally distributed. At the start of Figures 5.15 and Figure 5.16, the GPe node is not firing and so none of the population is held in its refractory period. The connection delay between the STN and GPe is 0.006 s. Using the regular pattern of activation shown in Figure 5.15, we see that the STN node is firing at a high rate which is increasing during the 0.006 s up to time 1 s of the simulation, the start of the plot, and so the

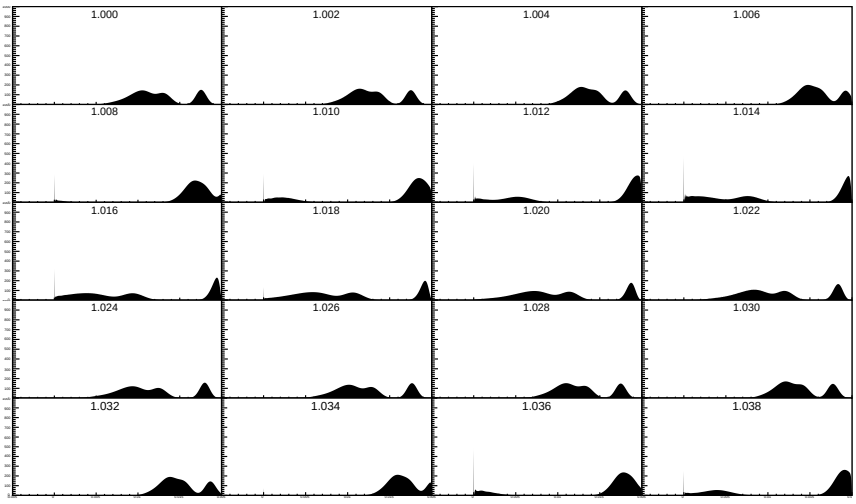


Figure 5.16: Changes in the population density of the GPe for times corresponding to Figure 5.15.

membrane potentials of the GPe neurons are being shifted towards the right, the threshold potential. At 1.006 s, membrane potentials are being pushed over the threshold and so neurons in the node are firing. The distribution of membrane potentials forms two clear peaks. At 1.008 s, some of the neurons which have fired re-enter the distribution at the reset potential. There is a smaller proportion of the neurons being pushed over the threshold potential as those reaching threshold correspond to the dip between the two peaks. The corresponding firing rate of the STN at 1.002 s is still high and so the firing rate rises as the second of the two previous peaks is pushed over the threshold. The firing rate of the STN is falling and by 1.02 s is not strong enough to push the membrane potential of the GPe neurons over the threshold and so the GPe stops firing and the peak close to the threshold potential starts to move back towards the reset potential. There is still a portion of the GPe neurons with a membrane potential close to the threshold as well as a population which has fired and been reset. At 1.026 s the membrane potential distribution looks the same as it did at 1 s.

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	3000	2	0
Background to GPe	3000	1	0
Background to STN	3000	1.1	0.1
STN to GPe	1250	0.15	0.006
GPe to STN	1250	-0.04	0.006
GPe to GPe	10000	5×10^{-7}	0.004

Table 5.9: Parameters for Figure 5.17.

Figure 5.17 shows a simulation, generated using the parameters given in Table 5.9, with a more complex profile to the repeated oscillations, having three peaks in the firing of the GPe node during one period.

Figure 5.18 shows detail from one oscillation during the simulation shown in Figure 5.17, starting from 1.7 s of simulation. From Figure 5.18 we can estimate the period of the oscillations to be 0.028 s, corresponding to a frequency of 36 Hz. The three peaks in the firing rate of the GPe node consist of a double peak which occurs while the STN node is firing and a separate peak which occurs when the firing rate of the STN node is very low.

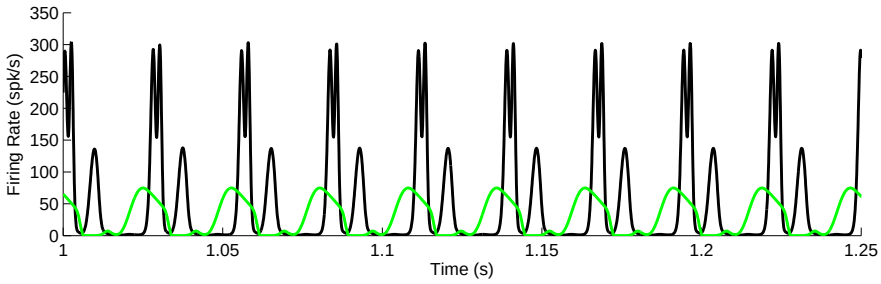


Figure 5.17: Oscillations which show three peaks in the firing rate of the GPe node during one cycle.

Figure 5.19 shows a sequence of snapshots of population density of the GPe node at times corresponding to Figure 5.18 and, as with the previous example, the membrane potentials are not normally distributed during the cycle. In both these examples, we can see that the membrane potentials remain in a split pattern from one cycle to the next, but that does not explain how the split occurs at the beginning of the simulation.

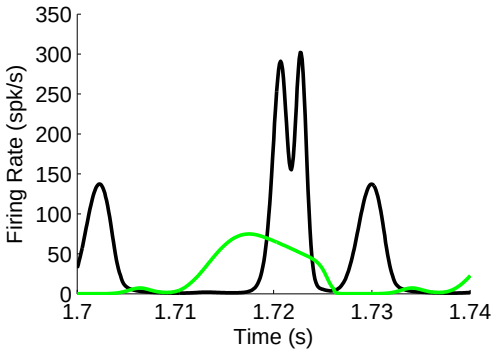


Figure 5.18: Detail of the firing rates for the simulation in Figure 5.17.

Figure 5.20 shows the firing rates at the beginning of the simulation shown in Figures 5.17 to 5.19, from the time that the additional input is fed to the STN node at 0.1 s of simulation. Note that the y-axis is scaled differently to that shown in Figure 5.18 as the first time the GPe node fires, the firing rate reaches above 600 spk/s.

Figure 5.21 shows the population density for the GPe node corresponding to Figure 5.20 when the additional input has been applied to the system. Initially, the membrane potentials in the GPe node are normally distributed. At 0.112 s some of the

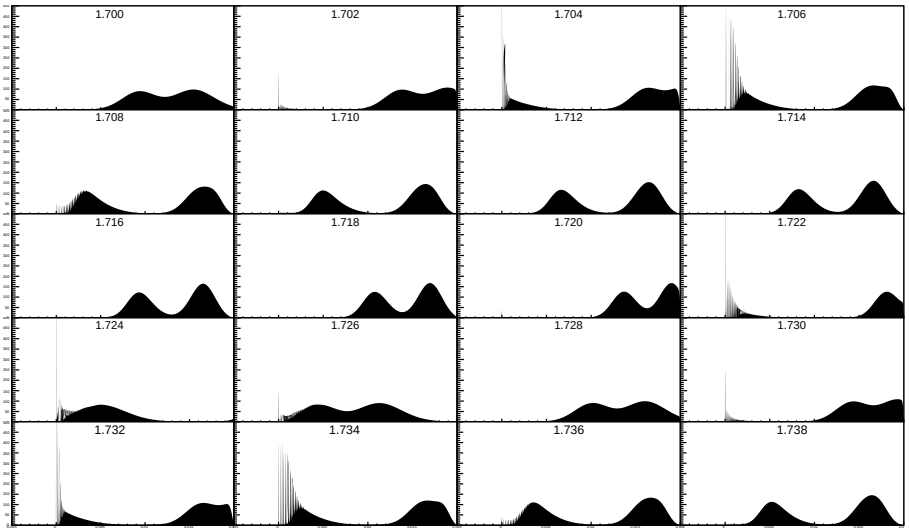


Figure 5.19: Population density for the GPe node corresponding to Figure 5.18.

population has fired and at 0.114 s the whole population has fired and some neurons have been set to the reset potential. This corresponds to a time when the STN is still firing so the membrane potentials of the GPe neurons increase until 0.12 s when the GPe has started firing again. At 0.124 s, not all of the neurons in the GPe population have fired a second time, but taking into account the 0.006 s delay in transmission from the STN to GPe, at 0.118 s the STN node is not firing and so not exciting the GPe node. This has caused a split in the distribution of membrane potentials in the GPe neurons.

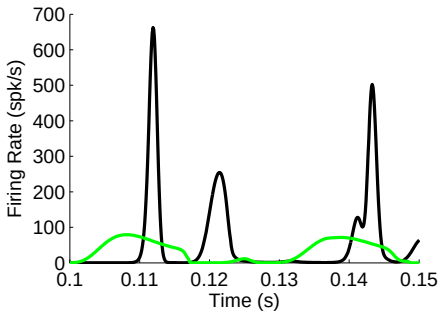


Figure 5.20: Detail from the start of the simulation shown in Figure 5.17, from the time that the additional input is first applied.

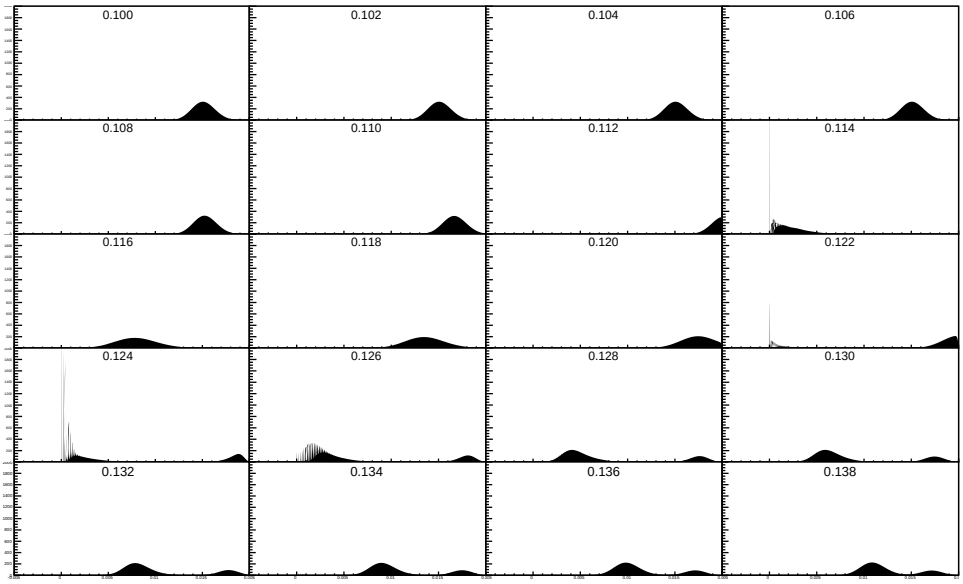


Figure 5.21: Population densities for the GPe node at the start of the simulation, corresponding to the output shown in Figure 5.20.

Varying the simulation parameters still further resulted in the output shown in Figure 5.22 which was based on the parameters given in Table 5.10. Comparing Tables 5.10 and 5.8 which gave a simulation with a double peak in the GPe firing rate, we see that Table 5.10 has stronger connections from the STN to GPe and the GPe to STN in addition to stronger background input to the STN. In Table 5.10 we see that the GPe now has an inhibitory self-connection.

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	10000	0.55	0
Background to GPe	10000	0.325	0
Background to STN	10000	0.4	0.1
STN to GPe	10000	0.007	0.008
GPe to STN	10000	−0.004	0.008
GPe to GPe	10000	−0.005	0.004

Table 5.10: Parameters for Figure 5.22.

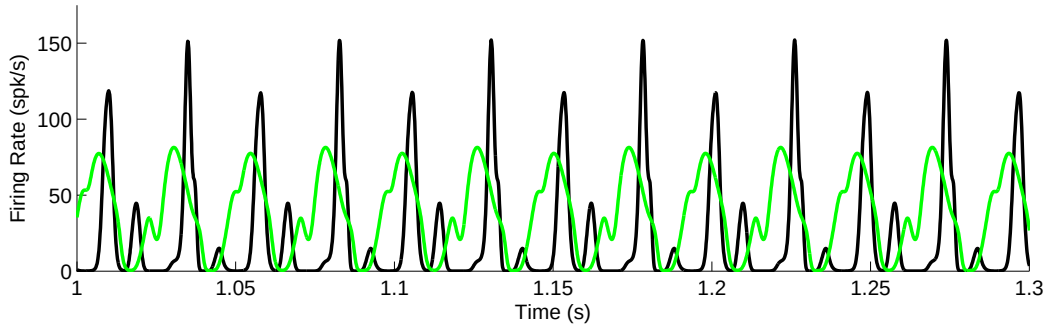


Figure 5.22: Oscillations in which the main peaks of the activation of the GPe alternate in amplitude, produced using the parameters shown in Table 5.10.

In Figure 5.22, the prominent peaks in the firing rate of the GPe node alternate between approximately 120 spk/s and 150 spk/s for the peak firing rate reached. It is also clear that the firing rates of the STN node also do not form regular peaks.

Figure 5.23 shows the activation of the two nodes of the simulation in Figure 5.22 for just over one full period of the oscillations. Using this figure, the elapsed time between the two main peaks in the GPe firing rate is approximately 0.025 s which corresponds to a frequency of 40 Hz. The full period of oscillation at which the pattern of activation of the GPe repeats is approximately 0.048 s corresponding to a frequency of 20.8 Hz.

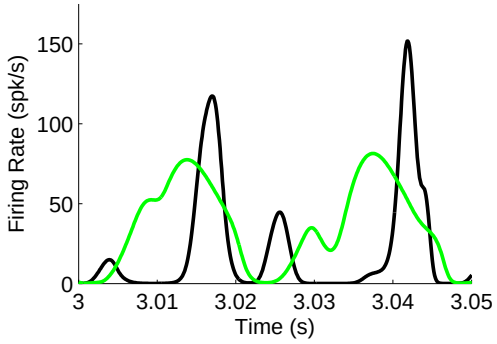


Figure 5.23: Detail of one period of oscillation from Figure 5.22.

Figure 5.24 shows another simulation, based on the parameters in Table 5.11. Those parameters not shown in Table 5.11 were set as in Table 5.10 so we see that the only differences are to decrease the strength of the connection from the STN to GPe but to

Connection	Number of Connections	Efficacy	Delay (s)
STN to GPe	10000	0.006	0.006
GPe to STN	10000	−0.005	0.006

Table 5.11: Parameters for Figure 5.24 which shows complex activity.

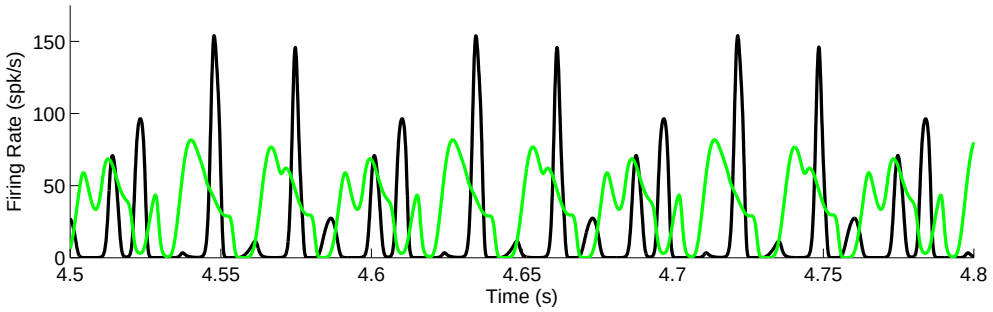


Figure 5.24: A complex but repeating pattern of activation in the two nodes, given by using the parameters in Table 5.11.

increase the magnitude of the inhibitory connection from the GPe to STN. Figure 5.24 shows that the period for the repetition in patterns of firing rates is longer still as the 0.3 s of simulation shown only contain a little over three oscillations based on the peaks of firing of the GPe node.

Having found complex patterns of oscillations of firing rates of the two nodes, I changed the parameters to look for longer periods in the oscillations and the possibility

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	3000	2.0	0
Background to GPe	3000	0.75	0
Background to STN	3000	1.0	0.1
STN to GPe	1250	0.15	0.008
GPe to STN	1250	−0.02	0.008
GPe to GPe	10000	5×10^{-7}	0.004

Table 5.12: Parameters for Figure 5.25 which shows low frequency oscillations.

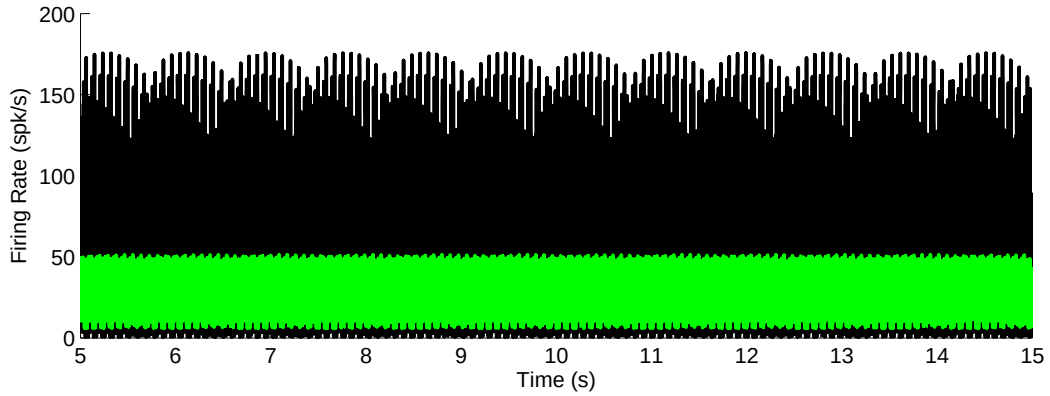


Figure 5.25: Simulation in which there is a very slow repeating pattern of activation.

of patterns which do not repeat. To allow more complex interactions, I increased the delay in transmission of signal from the STN to GPe and back to 0.008 s, the previous simulations had this delay set to 0.006 s.

Figure 5.25 shows a simulation in which there is a very slow repeating pattern in the amplitude of the peaks of the firing of the GPe, showing approximately 12 repetitions in 10 s of simulated time although we also see individual peaks making up the slow pattern. This simulation was produced using the parameters in Table 5.12.

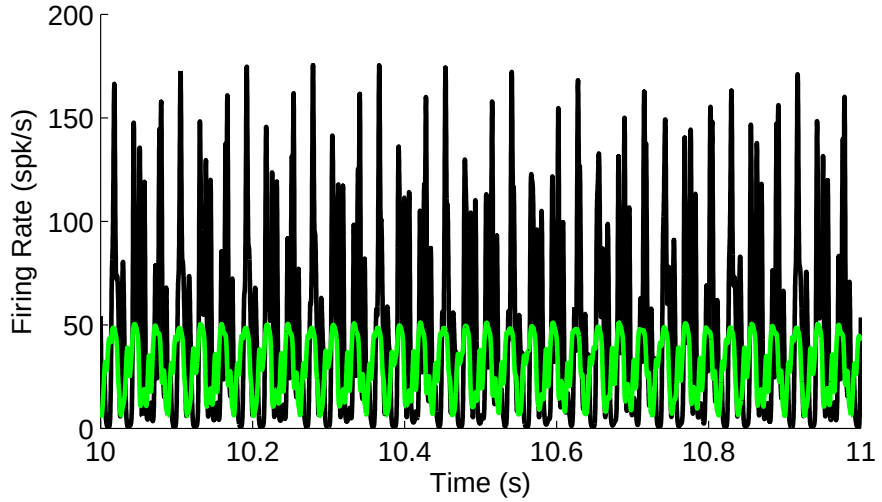


Figure 5.26: Detail of one second of the simulation shown in Figure 5.25.

One second of the same simulation as that shown in Figure 5.25 is shown in Figure 5.26. Now we can see approximately 35 peaks in the firing rate of the STN node although this is only just over one repeat of the slow pattern which is more clearly seen in Figure 5.25.

Connection	Number of Connections	Efficacy	Delay (s)
Background to STN	3000	2.0	0
Background to GPe	3000	1.0	0
Background to STN	3000	1.0	0.1
STN to GPe	1250	0.15	0.008
GPe to STN	1250	-0.03	0.008
GPe to GPe	10000	5×10^{-7}	0.004

Table 5.13: Parameters for Figure 5.27 where the output of the simulation does not appear to repeat.

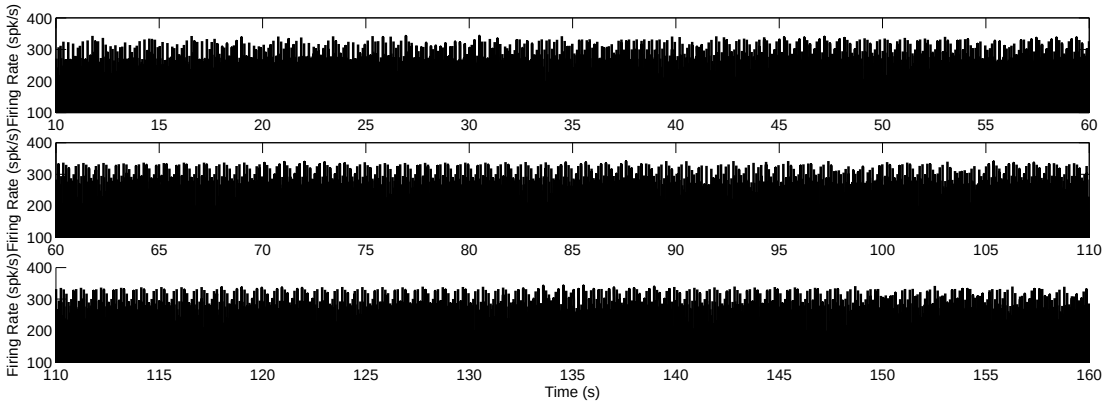


Figure 5.27: Activation of the GPe node using the parameters in Table 5.13 where the output does not appear to repeat over a long time of simulation.

Figure 5.27 shows a simulation, produced using the parameters in Table 5.13, in which there does not appear to be a repeating pattern, even after simulating 160 s. Here only the activation of the GPe node is shown to show the pattern of the maximum firing rates reached over time.

Comparing Tables 5.12 and 5.13, we can see that only two parameters differ between

the simulations shown in Figures 5.25 and 5.27, the background input to the GPe and the strength of the connection from GPe to STN.

An alternative means to examine the relation between the outputs of the two nodes is to plot the activity of one node against that of the other. For a simple periodic system, this will result in a loop. Such plots are shown in Figures 5.28 and 5.29 for the simulations shown in Figures 5.25 and 5.27 respectively. In each of Figures 5.28 and 5.29, the intensity of the colour shows the number of times the same pair of values of the two activations was recorded. The axes are not labelled for visual reasons only. Figure 5.28 clearly shows that there is an oscillatory pattern, as the plot forms a loop, but this loop has a complex structure. In Figure 5.29, it is not clear that the pattern ever repeats exactly, supporting the view shown in Figure 5.27 that this simulation does not repeat.

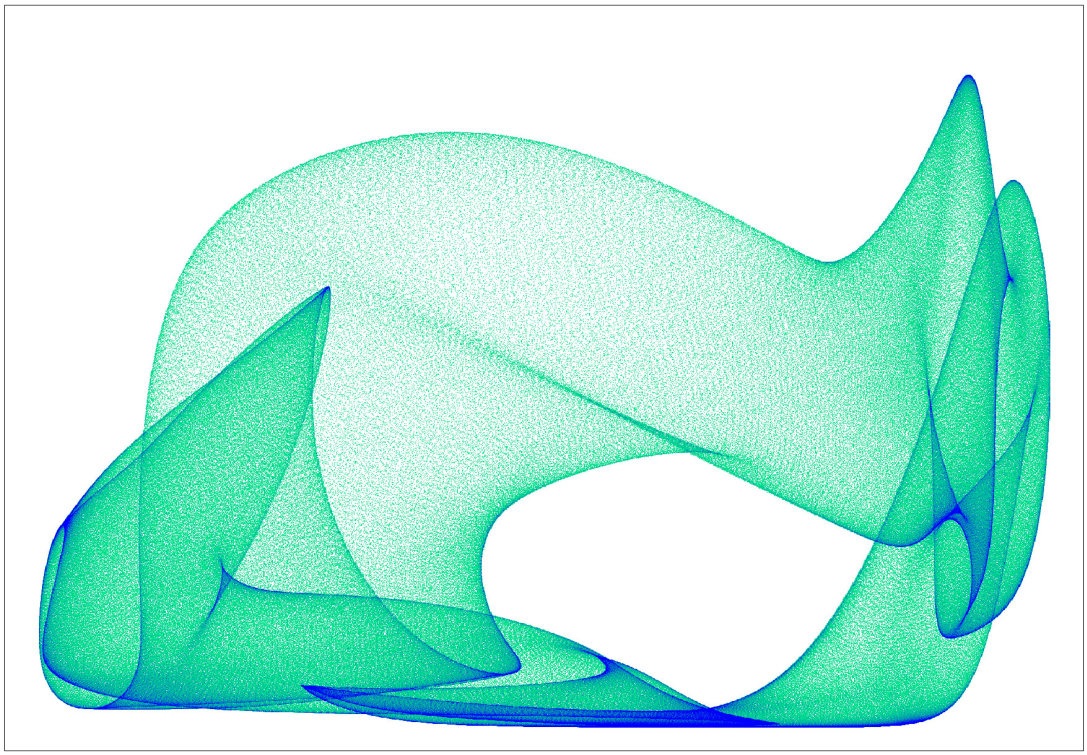


Figure 5.28: Activation of the GPe (y-axis) plotted against activation of the STN (x-axis) for the simulation shown in Figure 5.25.

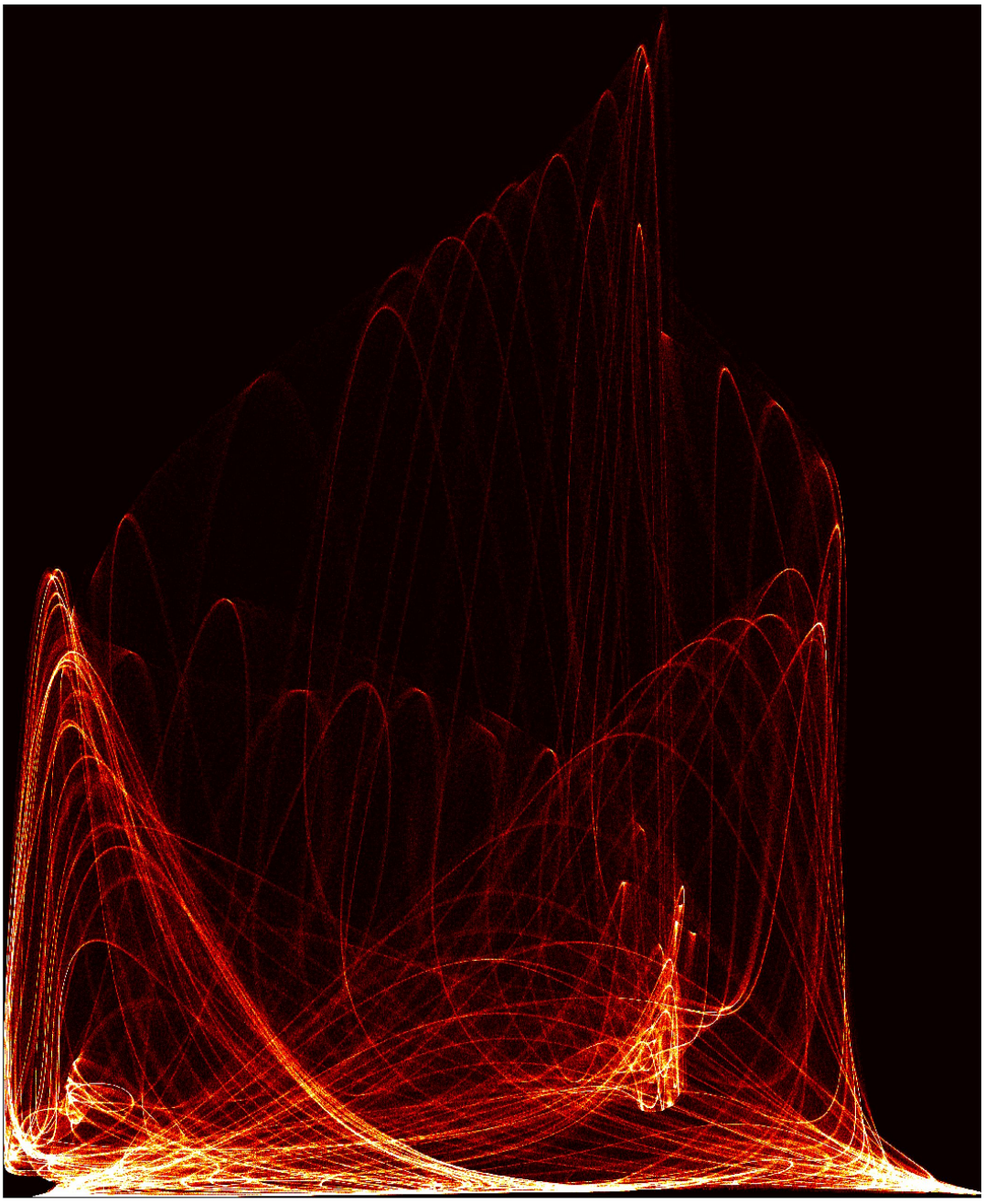


Figure 5.29: Activation of the GPe (y-axis) plotted against activation of the STN (x-axis) for the simulation shown in Figure 5.27.

5.7 Discussion

5.7.1 Summary of results

In this chapter I provided an introduction to the underlying theory behind population density modelling and then showed how neural interactions can be simulated using this technique.

I showed simulations of a simple circuit consisting of an excitatory and an inhibitory node. Within these simulations, I showed how changing some of the parameters of the model allows the simulation to transition from reaching a steady state to oscillatory behaviour, in line with the findings of Nevado Holgado et al. (2010).

I amended the parameter settings for the same underlying neural circuit and showed that as the simulations moved beyond simple oscillations, there became structure in the pattern of activity within each period of oscillation. I used the membrane potential distributions to show how the structure in the firing rate patterns relates to an underlying split in the population density profile.

I showed a simulation which had a pattern of activation which appeared to repeat over a long time scale and one which did not appear to repeat. This shows that complex patterns of behaviour can be produced by a very simple underlying model.

5.7.2 Assumptions and limitations

The simulations in this chapter, by necessity, only cover a very small part of the parameter space of possible simulations which would be possible even for a simple model with only two connected neural populations. In particular, I have only shown a small amount of investigation into the effects of different variances in the connections, in Section 5.5. I have also not varied the membrane time constant or refractory period in the investigations of the excitatory inhibitory circuit.

Although I have estimated the frequency of the oscillations for some of the results shown, I have not included any formal analysis of the frequencies. The final simulation, shown in Figures 5.27 and 5.29, gives the impression that the pattern of activation of the two nodes does not repeat. This has not been formally analysed to test whether this is chaotic behaviour.

It is now known that there are two types of GPe neurons which have different properties which give them different responses to input from STN neurons (Nevado Holgado

et al., 2014). My GPe population assumes that all neurons in the GPe have the same properties.

5.7.3 Relation to other work

I have shown how the conditions for oscillation in the STN–GPe loop as described by Nevado Holgado et al. (2010) can also be demonstrated using population density modelling. Nevado Holgado et al. (2010) show oscillations of 20 Hz when simulating the conditions of Parkinson's disease. These oscillations fall within the range described as beta oscillations, 16 to 31 Hz. I showed oscillations with a higher frequency, for example approximately 33 Hz shown in Figure 5.10, which falls at the low end of the gamma range of oscillations, 32 to 100 Hz. The difference in frequency could be due to the parameters chosen for the simulations, Nevado Holgado et al. (2010) used a firing rate based model and so the model parameters were not directly transferrable to population density techniques. Using a firing rate model required Nevado Holgado et al. (2010) to set sigmoid functions to control the activation of the populations, as described in Section 5.3 above. In population density modelling, this step is replaced by the setting of underlying background input to each node.

Nevado Holgado et al. (2010) showed activation which reached a steady state in both the STN and GP in conditions to represent a normal state as opposed to a Parkinson's disease state. Although enhanced oscillations in the STN and GPe are a feature of Parkinson's disease, oscillations are also a feature of normal activity in these nodes with oscillations recorded at a range of frequencies (Boraud et al., 2005). This normal oscillatory activity may be important in behaviour (Boraud et al., 2005; Kumar et al., 2011).

Humphries et al. (2006) created a spiking neuron model of the basal ganglia in order to investigate action selection. Within their investigations they report oscillatory activity in the STN–GPe loop. They show slow oscillations in conditions of Parkinson's disease and oscillations of 55 Hz, in the gamma range, when simulating normal awake activity in rats. In order to examine the oscillations, Humphries et al. (2006) had to compute a moving average firing rate over groups of neurons. Using population density models saves this step.

Merrison-Hort & Borisyuk (2013) also used a spiking neuron technique, in this case to investigate the STN–GPe loop only. They connected the STN to the GPe so that slow

wave activity in the STN drove activity in the GPe, but did not include a reciprocal connection from the GPe to the STN. They did, however, include lateral inhibition within their GPe neurons. When examining oscillatory activity averaged over groups of neurons, they found that some of the GPe neurons fired in phase with the STN and some out of phase. This pattern was an emergent feature of the system in which all the GPe neurons had the same neuronal properties. Although I am not looking at slow wave activity, the firing of the GPe population at two different times with respect to the phase of the STN output could be similar to I observed in Figure 5.19. Using population density modelling, I can see how the oscillations relate to the split in the population densities. Merrison-Hort & Borisyuk (2013) point out that using their method of analysing groups of spiking neurons, it would be difficult to find such patterns in higher frequency oscillations.

Kumar et al. (2011) simulated a population of 1,000 spiking neurons in the STN and 2,000 in the GPe, in contrast to the 50 STN and 100 GPe neurons modelled by Merrison-Hort & Borisyuk (2013). As with Nevado Holgado et al. (2010), Kumar et al. (2011) were examining conditions for the onset of oscillations but Kumar et al. (2011) did not require changes to the connections between the STN and GPe. Kumar et al. (2011) state that the onset of oscillations can be due to increased excitation of the STN from the cortex, or increased inhibition of the GPe from the striatum. They consider how oscillations can be reduced, in particular by changing the input to the STN which has significance for the use of deep brain stimulation as a treatment for Parkinson's disease.

5.8 Conclusions

I have introduced population density modelling and shown how this technique can replicate results using other computational methods for modelling neurons. In the next chapter, I create a model of the basal ganglia using population density techniques.

Chapter 6

Basal Ganglia Model

6.1 Introduction

This chapter builds on knowledge of the connections and functions of the basal ganglia described in Chapter 4 by using population density modelling techniques described in Chapter 5 to create neural simulations to select the appropriate action in a situation modelled on the behavioural task of Bland & Schaefer (2011) described in Chapter 2. I consider how the task can be represented in a neural system and how the responses of the neural system vary with the difficulty of the task.

To recap, the scenario in the behavioural task is that a red or blue stimulus is shown to a participant on each of successive trials and the participant decides which of two buttons to press. There are two underlying rules that the environment can hold. One possibility, rule 1, is that the correct response is to press button one in response to red and press button two for blue. The second rule, rule 2, is the opposite of rule 1, that is the correct response is to press button two in response to the red stimulus and button one in response to blue. The participants in the task are not introduced to the underlying environmental rules, they have to develop a belief as to which is the current rule from the feedback of whether the response was correct or not. The environmental rules are rewarded in a probabilistic manner making it difficult to identify a rule change.

In this chapter, I do not consider the learning aspect of the task which I return to in Chapter 7, I merely examine a neural structure which can combine input representing a belief as to which underlying rule applies in the environment and input representing one of two colours. The model can produce output representing two buttons in a way

which is appropriate for the belief and colour simulated. Firstly, I implement a simple model representing the direct pathway of the basal ganglia, as described in Chapter 4 Section 4.2.1. I use a simple rule to determine which response is made by the system and consider how the timing of a response changes for different levels of simulated belief in the environmental state.

After examining the response of a simple model, I include the influence of an STN–GPe loop which gives the hyperdirect pathway. As described in Chapter 5, the interactions between the STN and GPe alone can produce complex behaviour. I add the STN–GPe circuit to the basal ganglia model and consider how its influence affects responses.

6.2 Simple model for action selection

6.2.1 Setting up the model

In this section, I consider a model of straightforward decisions which are based on the behavioural task of Bland & Schaefer (2011). By straightforward decisions, I mean where there is a large difference between the evidence supporting two opposing responses. In the simple decision modelled here, evidence for one rule and one colour above the other rule and colour need to be integrated to produce a response.

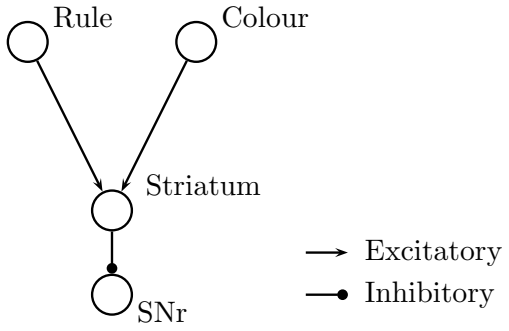


Figure 6.1: Simple model of the basal ganglia used for action selection.

To make these simple decisions, I use the network shown in Figure 6.1 to model the direct pathway of the basal ganglia. In the direct pathway, excitatory cortical input, represented by the nodes labelled Rule and Colour in Figure 6.1, feeds into the striatum,

one of the main input nodes of the basal ganglia. Separate neural populations in the striatum inhibit populations in the SNr, the activity of which forms the output of this model. As described in Chapter 4, the regions GPi and SNr can together be considered as the basal ganglia output, in this chapter I refer simply to the SNr. The normal state of the SNr nodes is to produce activity and a decision is made when the firing rate at the SNr falls. The fall in SNr activity would allow disinhibition of the thalamus as described in Chapter 4, but this step is not included in my model.

Figure 6.1 shows the relationships between the different areas in the model but does not show every individual neural population simulated. The Rule and Colour nodes are each assigned two neural populations to correspond to the two rules and two colours in the psychological task. Making the assumption that the two colours in the task are easily distinguishable, the colour stimulus is implemented by making one of the two cortical colour nodes fire much more strongly than the other. This is achieved in the simulations by setting different efficacies for the connections from a steady background input to each to the two cortical colour nodes. Although two cortical rule nodes are implemented in the model, a difference in belief between the two rules is implemented by setting different efficacies in the connections between the cortical rule nodes and the striatum. I refer to these efficacies as the strengths of beliefs in the two rules, or merely the rule strength. This implementation is based on the ideas of plasticity of the connections between the cortex and the striatum so that learning can take place and rule strengths would be able to change in response to feedback. In this chapter, I provide different connection strengths to represent beliefs in different rules and do not implement learning whilst using a model which can be easily extended to incorporate aspects of learning.

Figure 6.2 shows the full set of neural populations modelled for the striatum and SNr. In this model, I assume that the striatum provides an area where inputs from different cortical areas are combined, shown in Figure 6.2, by the set of four nodes within the striatum which are all connected to each other by inhibitory connections. These nodes will be referred to as the associative striatum based on the nomenclature used by Guthrie et al. (2013) who used an associative striatum to combine motor and cognitive input.

Within the area labelled as striatum in Figure 6.2, the nodes labelled C1, C2, R1 and R2 will be referred to as the specific striatum. The specific striatum nodes re-

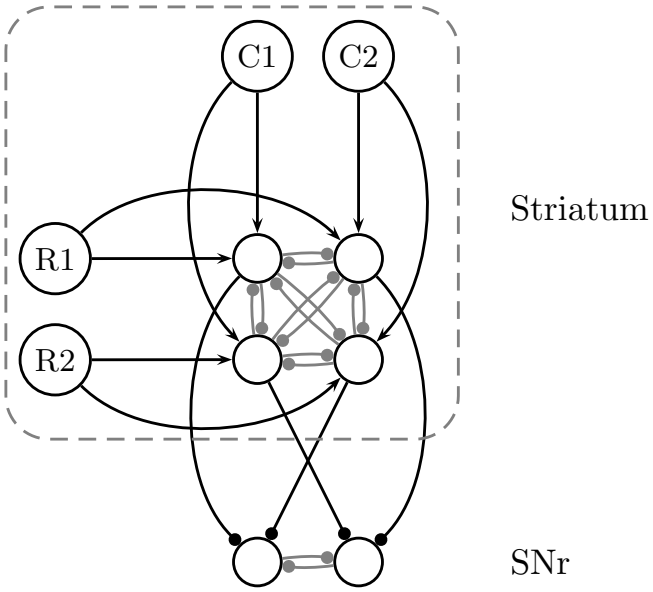


Figure 6.2: All the nodes and their connections for the striatum and SNr in the simple basal ganglia model shown in Figure 6.1.

ceive excitatory input from their corresponding cortical rule or colour nodes, which are omitted from Figure 6.2. For simplicity of modelling, although neurons in the striatum usually give inhibitory output, these nodes are simulated as having excitatory output and merely transfer a signal from the cortex to the rest of the striatum. As described above, for the colour nodes in the cortex, one will be firing more strongly than the other, this difference will be transferred directly to the specific striatum nodes C1 and C2. One of the two specific striatum rule nodes, R1 and R2 will fire more strongly than the other due to differences in the connections from the cortex to specific striatum, or rule strengths. The four associative striatum nodes shown in Figure 6.2 each combine the activation representing one rule and one colour.

The four nodes of the associative striatum, as well as inhibiting each other, send inhibitory output to the two SNr nodes. I consider the two SNr nodes to represent the two response buttons in the task as they would disinhibit distinct populations in the thalamus which would signal the corresponding motor action via the cortex. As shown in Figure 6.2, each of the SNr nodes is inhibited by two of the associative striatum

nodes. This set of connections allows any combination of the two rules and colours to be converted to a response of one button. The two SNr nodes also inhibit each other, this helps to create a difference in firing rates in response to slightly different inputs.

To create the simulations examined here using population density modelling with Miind (de Kamps et al., 2008), each of the populations is set to have a threshold potential of 0.02 V with a minimum potential of -0.02 V. Following firing, there is a refractory period of 2 ms after which neurons are reset at the reset potential equal to the resting potential of 0 V. Different membrane time constants apply to different neural populations as shown in Table 6.1. The membrane time constants for the associative striatum and SNr are as used by Humphries et al. (2006).

Node	Membrane time constant (ms)
Cortex	10
Specific Striatum	10
Associative Striatum	25
SNr	8

Table 6.1: Membrane time constants used in the simple basal ganglia model.

If one of each of the rule and colour nodes at the specific striatum is more active than the other then one of the four nodes in the associative striatum should fire more strongly than the other three. I used trial and error to determine connection parameters to give a large difference in firing rate profile between the appropriate node in the associative striatum for the active rule and colour and the other nodes. As described in Chapter 5 Section 5.5, connecting populations using Miind requires the specification of a number of connections, an efficacy or strength and optionally a delay. For the simple basal ganglia model, the connection strengths and delays are set as shown in Table 6.2 each having 1250 effective connections. All references to connection strengths or efficacies in this chapter are given as a percentage of the threshold potential, which was set to 0.02 V for all nodes, but are shown without percentage signs. Negative values for efficacies indicate inhibitory connections. The delay in transmission for the connections associative striatum–SNr and SNr–SNr shown in Table 6.2 are as used by Thibeault

& Srinivasa (2013). These connections remain constant for all the simulations in this chapter unless otherwise specified.

Connection	Delay (ms)	Efficacy
Associative Striatum–SNr	6	−0.6
Associative–Associative Striatum	1	−0.0008
SNr–SNr	6	−0.06
Specific–Associative Striatum	1	0.0135
Colour–Specific Striatum	10	0.024

Table 6.2: Parameters for connections between nodes in the simple basal ganglia model.

To run the simulations, input has to be provided to the system; I use a steady background input of 1.8 spk/s which is connected to each node with the connection efficacies shown in Table 6.3, each link having 3000 effective connections. A large peak in firing rate can occur when a signal is passed from one node to the next. The background connections to the striatum and cortex were determined in order to try to stop this peak reaching unrealistically high firing rates. The background connections are based on the assumption that a small circuit such as that studied here cannot work in isolation and will always have other inputs which I consider to be noise and not related to this task.

As described in Chapter 4 Section 4.2.1, the SNr is continually firing during its normal state, known as tonic firing. A decision is made when the firing rate at an SNr node is low and has a disinhibitory effect on the thalamus. In the model presented here, tonic firing is achieved by setting the level of background input to the SNr nodes such that the SNr nodes would be firing except when enough inhibitory input is received from the striatum. Based on Humphries et al. (2006), I consider a decision to have been made when the firing rate of the SNr node falls below 5 spk/s. Humphries et al. (2006) show simulations in which the mean firing rates of their SNr populations are between 20 and 40 spk/s before stimuli are applied to the system.

Each simulation is run for 0.05 s of simulated time with the inputs shown in Table 6.3 to allow the populations to reach their equilibrium and represents the time before a rule

or colour stimulus is applied. This results in a low firing rate at the specific striatum which is not high enough to activate the associative striatum.

Node	Efficacy
Specific Striatum	1.45
Associative Striatum	0.575
SNr	2.4
Cortex	1.5

Table 6.3: Strength of connections from a steady background input to nodes in the simple basal ganglia model.

After allowing the populations to settle for 0.05 s, additional background input is fed to the rule and colour nodes in the cortex. These additional input connections represent the onset of the stimulus, that is the time when the red or blue triangle was shown to the participants. An additional connection from the background input to the cortical rule nodes becomes active with 3000 effective connections and an efficacy of 0.45 which is identical for the two rule nodes. Also at 0.05 s of simulation, additional connections become active from the background input to the cortical colour nodes, again with 3000 effective connections, with the efficacies set for different tests. The efficacies of the connections from the cortical rule nodes to the corresponding nodes in the specific striatum are also set for individual tests.

6.2.2 Making a straightforward decision

The model set up as described so far does not give any settings to represent the belief in the underlying rule or the colour stimulus. Table 6.4 shows two sets of parameters to achieve this. A colour stimulus is simulated by setting efficacies for the connections from the background input to the cortical colour nodes, shown as colour 1 and colour 2 in Table 6.4 which are set with 3000 effective connections. The values shown as rule 1 and rule 2 in Table 6.4 give the efficacy for a link from the cortical rule node to the corresponding node in the specific striatum with 1250 effective connections and, as with the connection from cortical colour node to specific striatum, a delay of 0.01 s. These are the connections which represent the strength of belief that one environmental rule

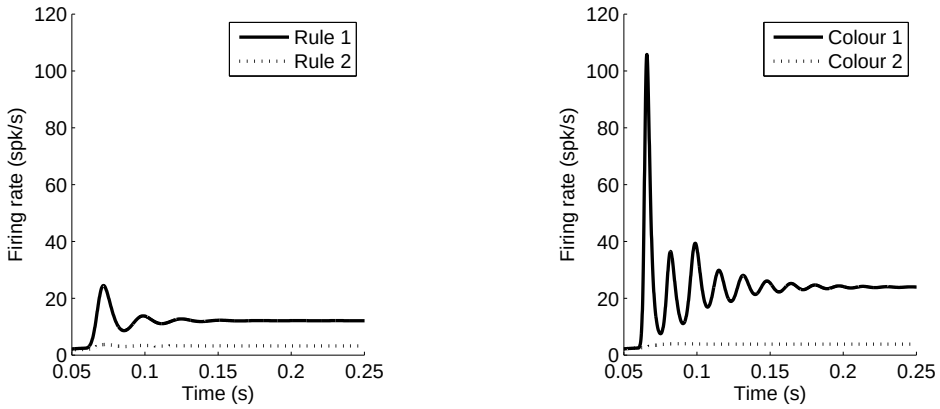
applies and are proposed to change during learning. In this chapter, I examine the impact of setting different values for these strengths on the ability of the network to select the correct action.

For simplicity, efficacies are presented as rule 1 and 2 and colour 1 and 2 as shown in Table 6.4. I reiterate that these efficacies do not refer to the same part of the network for rule and colour. Rule connections are between the cortex and the specific striatum whereas colour connections are between the background input and the cortex.

Firstly, I show that the correct action is achieved when there is a strong difference between the inputs by considering tests 1 and 2 from Table 6.4, which both have rule 1 higher than rule 2 and colour 1 higher than colour 2, and so the correct response would be button 1, with colour 1 as red. For these two tests, I examine not only the firing rates at the SNr nodes during the simulations, but also the firing rates at the striatal nodes to see how the decision is formed.

Test	Rule 1	Rule 2	Colour 1	Colour 2
1	0.0225	0.005	0.9	0.1
2	0.0425	0.005	0.8	0.1

Table 6.4: Parameter settings for tests of a simple decision using the simple basal ganglia model.

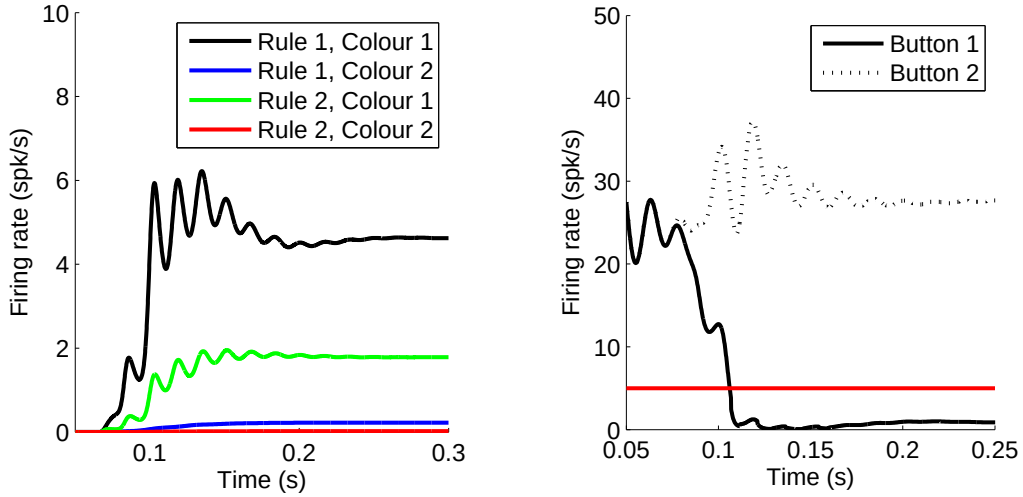


(a) Rule (b) Colour

Figure 6.3: Activations of the specific striatum nodes during simulation test 1.

Figure 6.3 shows the activity at the specific striatum nodes representing the two rules and two colours for the simulation test 1. For both the rule and colour, there is a clear difference between the activation of the two nodes. For the more active colour node, both the initial peak firing rate and the steady firing rate when the oscillations have died down are higher than for the most active rule node. In this section, firing rates are shown from 0.05 s of the simulation as that is when the additional connections become active to represent the time at which the stimulus is applied.

The specific striatum activation shown in Figure 6.3 is fed to the associative striatum resulting in the activation at the four striatal nodes shown in Figure 6.4a. It is clear that, as required, using the connection parameters shown in Tables 6.3 and 6.2 gives much stronger activity in the associative striatum node representing the appropriate combination of colour and rule than in the other nodes.



(a) Associative striatum

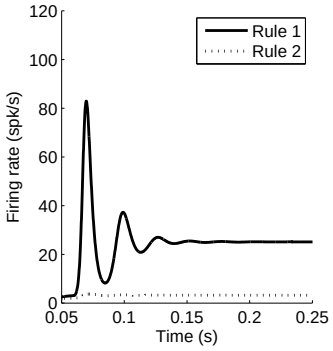
(b) SNr

Figure 6.4: Activations of the associative striatum and SNr nodes in the basal ganglia during simulation test 1.

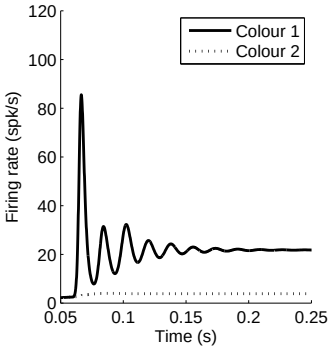
The four nodes of the associative striatum inhibit the two SNr nodes resulting in the output shown in Figure 6.4b which shows that the correct button has been selected, the SNr node corresponding to button 1 having a firing rate below 5 spk/s and that the button remains selected for the rest of the simulation shown. The time at which the firing rate goes below 5 spk/s is 0.106 s from the start of the simulation, I will refer to this as the response time. As the stimulus is applied after 0.05 s of simulations, in

this example the response is made 0.056 s after the stimulus is applied which could be considered to be a reaction time. Timings will be given from the start of the simulation rather than from stimulus onset.

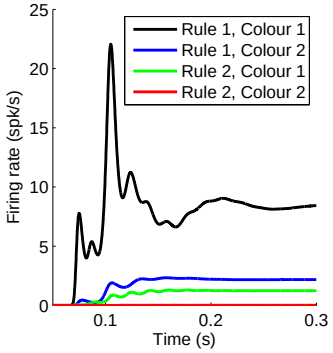
The parameters test 2 from Table 6.4 show an alternative set of inputs in which there is a clear difference between the activation for the two colours and two rules, giving the activation at the specific striatum shown in Figures 6.5a and 6.5b. Test 2 has a higher overall level of activation in the system than test 1. Figure 6.5c shows the activation at the associative striatum for test 2. Note that the maximum firing rate of the most active striatal node is much higher than shown in Figure 6.4a for test 1.



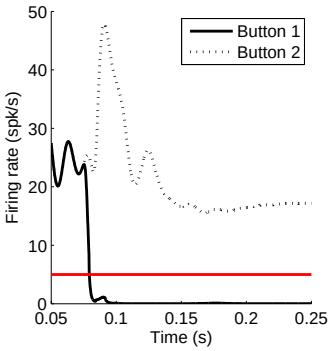
(a) Specific striatum – rule



(b) Specific striatum – colour



(c) Associative striatum



(d) SNr

Figure 6.5: Activations during simulation test 2 for the striatum and SNr nodes in the simple basal ganglia model.

Figure 6.5d shows that for test 2, the correct button is selected and stays selected for the remainder of the simulation. Comparing Figures 6.4b and 6.5d, it is clear that the response is made earlier in test 2 than test 1, the firing rate for the SNr node

corresponding to button 1 falling below the 5 spk/s threshold at 0.079 s of simulation. Also, for test 2 the activation of the selected response remains lower for the remainder of the time shown than in test 1.

6.2.3 Making the decision more difficult

I now examine the effect on the activity of the SNr nodes, and hence the decision made, of making the rule strengths closer to each other. If, based on tests 1 and 2 from Table 6.4, I make rule 2 stronger while keeping rule 1 constant, I will be providing more overall activation to the system. In this case, rather than just seeing the effect of a smaller difference between the two rule strengths, I would also see the effect of having more overall activation. I chose to use tests 1 and 2 from Table 6.4 as starting points and to make rules closer by decreasing rule 1 and increasing rule 2 by the same amount. Using this method, I change the efficacy of the connection between the corresponding rule nodes in the cortex and specific striatum, but not the number of connections. This method of altering rule strengths has the potential that strengths could be set by reinforcement learning as used in Chapter 3. I do not make any changes to the parameters for colour given in tests 1 and 2 as in the psychological task which motivates this neural modelling, it is assumed that the colours are clearly distinguished, but as learning takes place the belief about the underlying environmental state can change.

Using the method described above to amend rule strengths, and starting from test 1 from Table 6.4, I increased rule 2 by 0.001 and decreased rule 1 by the same amount keeping all other parameters constant to give the results shown in Figure 6.6. In Figure 6.6 each successive test is shown in a lighter grey with the lightest grey showing the results for rule 1 set to 0.0145 and rule 2 to 0.013. Here only the SNr activations for button 1 are shown in order to focus on the times at which these first cross the 5 spk/s threshold which represents a decision. In each simulation shown in Figure 6.6, the firing rate of the SNr node corresponding to button 2 was higher than the highest shown for button 1 in Figure 6.6 and so was not close to crossing the 5 spk/s threshold.

Looking at Figure 6.6, it is clear that for the lightest grey plot, the activation of the SNr node does not cross the 5 spk/s threshold at all, so no response is made. All of the plots, with different levels of difference between the two rules, show the same timings in the rise and fall of activation but with different amplitudes. At the first dip after 0.1 s the lightest three lines do not cross the threshold. At the next dip in activation,

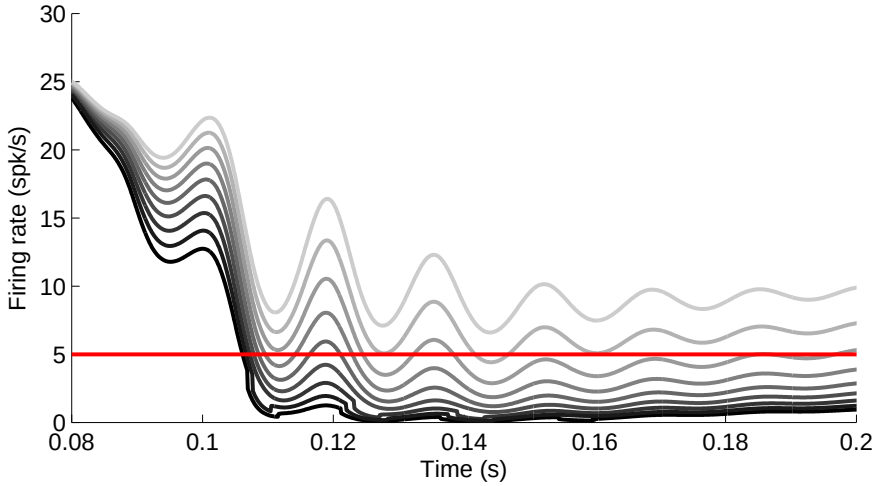


Figure 6.6: Firing rates of the SNr node corresponding to button 1 when making the decision more difficult starting with test 1.

the darkest of those three lines now crosses the threshold, at 0.125 s of simulation. The second lightest line crosses the threshold at 0.142 s. These response times for different stimuli are separated even where the stimuli have only changed by a small amount. These separations in response times relate to the oscillations which occur in the firing rate of the SNr node after a stimulus has been applied and before the oscillations have damped down.

To give more quantitative details to the change in response times for different differences between the rules, additional tests were run starting with the parameters shown as start in Table 6.5 and for each successive test decreasing the strength of rule 1 by 0.000125 and increasing rule 2 by the same amount until reaching the condition shown as finish, giving 33 simulations. This gives smaller differences between successive tests than in Figure 6.6 and now I focus on the first time at which the activation crosses the 5 spk/s threshold and not the pattern of rises and falls. The parameters shown as finish in Table 6.5 were chosen as using this scheme to change the rules and starting from test 1 of Table 6.4, when the rules are closer, no response is made at all.

Figure 6.7 shows the response times against the difference in the strengths of the two rules. For each of the tests shown in Figure 6.7, the activation of the correct SNr node, that corresponding to button 1, crosses the threshold whilst that corresponding

Test	Rule 1	Rule 2	Colour 1	Colour 2
Start	0.019	0.0085	0.9	0.1
Finish	0.01525	0.01225	0.9	0.1

Table 6.5: Start and end parameters for testing the response time at differences between the rules giving the results shown in Figure 6.7.

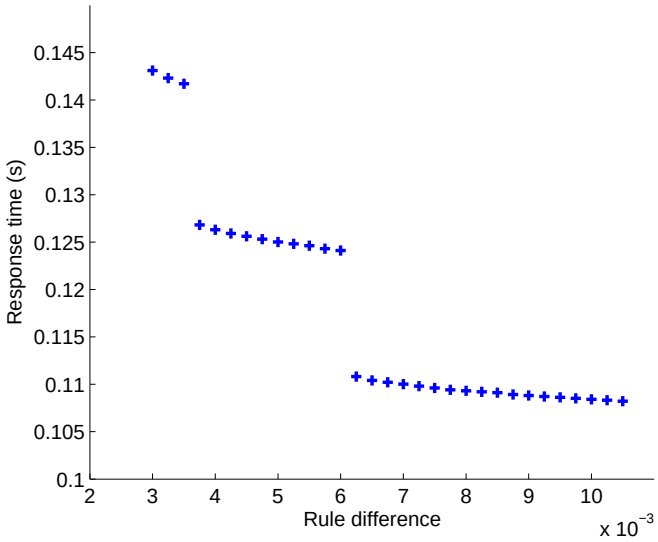


Figure 6.7: Response times against difference in strength between the two rules based on starting from test 1.

to button 2 does not cross the threshold. A decision is expected to be more difficult when there is a small difference between the rule strengths, and these show the longest response times in Figure 6.7. As the decision becomes easier, due to there being a bigger difference between the rules, the response times decrease. The decrease in response times has large steps for some changes in rule difference even though the change in rule strength was the same between each test. For the fastest decisions shown in Figure 6.7, the rate of change of response time between successive tests is very small.

In Section 6.2.1, I described two possible starting points for simulations, test 1 and test 2 given in Table 6.4. Having now examined making the decision more difficult when starting from test 1, I now take a similar process to make the rules closer starting from

test 2. Test 2 has more overall activation in the system and a bigger initial gap between rule 1 and rule 2.

Test	Rule 1	Rule 2	Colour 1	Colour 2
Start	0.0375	0.01	0.8	0.1
Finish	0.0255	0.022	0.8	0.1

Table 6.6: Parameters for making the decision more difficult starting from test 2 and used for the simulations shown in Figure 6.8.

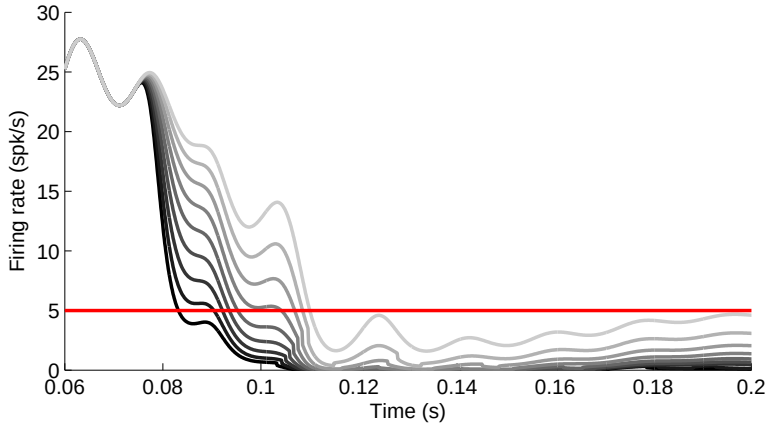


Figure 6.8: Firing rates of the SNr node corresponding to button 1 when making the decision more difficult starting from test 2.

Given test 2 from Table 6.4, we can subtract 0.005 from rule 1 and add the same to rule 2 to give the parameters shown as start in Table 6.6. Figure 6.8 shows simulations which take the start parameters from Table 6.6 and add 0.0015 to rule 2 and subtract the same from rule 1 for each successive test until reaching the parameter values shown as finish. Again it is clear that, although the change to the rules was the same for each successive test, the times at which the activation crosses the 5 spk/s threshold are not evenly spaced.

Continuing to add 0.0015 to rule 2 and subtract the same from rule 1 from the parameters finish in Table 6.6 gives settings for rule 1 and rule 2 of 0.025 and 0.0225 respectively. This process can be repeated twice more to give 0.024 and 0.0235 for rule 1

and rule 2 after which there is a reversal in the dominant rule. Figure 6.9 shows the SNr activation from the three tests defined by such a process. In Figure 6.9 the activation for the SNr node corresponding to button 1, a correct response, is shown by a solid line, and that corresponding to button 2 by a dotted line.

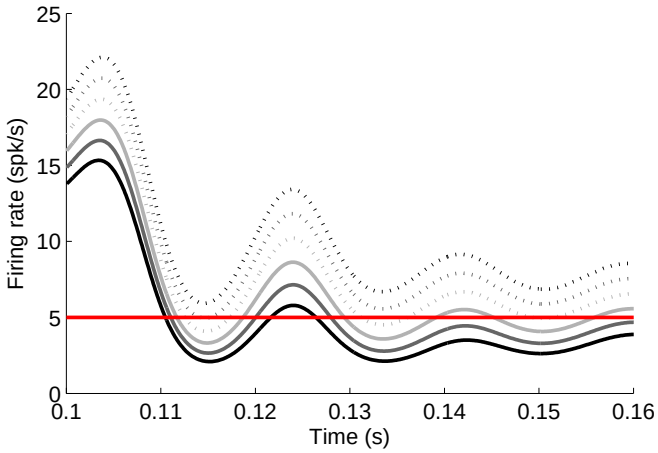


Figure 6.9: Firing rates of the SNr nodes for tests starting from test 2 and making the two rules as close as possible. Solid lines show the activation related to button 1 and dotted lines show activation related to button 2.

The lightest grey lines in Figure 6.9 show the activation of the SNr where rule 1 and rule 2 are set to 0.024 and 0.0235 respectively and the activation of both SNr nodes dips below 5 spk/s. The activation corresponding to button 1, the correct response, crosses the 5 spk/s threshold at 0.112 s and that corresponding to button 2 crosses at 0.113 s. Using the simplification that the response is given by the first SNr node to cross the threshold this test is taken to show a correct response, however the activation corresponding to the incorrect response crosses the threshold only a millisecond later. Depending on how the signal is treated by the next part of the neural system, and not modelled here, this test could also be interpreted as both responses having been selected. This case of the activation of both SNr nodes dipping below the threshold for the smallest difference between the rules starting from test 2 contrasts with the case shown in Figure 6.6 which was based on starting from test 1 and neither response was selected when the difference between the strengths of the two rules was close.

To examine the response times for a range of rule differences at a finer grain than in

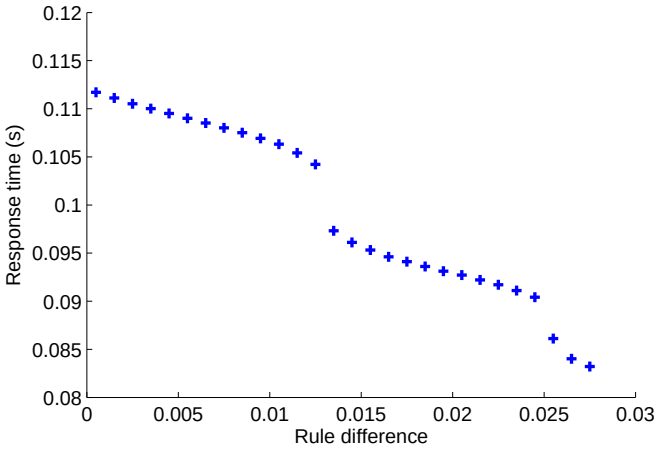


Figure 6.10: Response times against difference in strength between the two rules when starting from test 2.

Figures 6.8 and 6.9, Figure 6.10 shows the response times, starting with the parameters labelled as start in Table 6.6 and altering each rule in steps of 0.0005, until meeting the conditions of the final test in Figure 6.9. Again it is clear that there are steps in response time.

6.2.4 Swapping rule and colour

So far, the simulations shown have all had rule 1 and colour 1 as the dominant ones. I now demonstrate that the correct button is still selected when the dominant rule and colour are reversed.

Figure 6.11 shows the firing rates for the SNr nodes corresponding to each button when the dominant rule and colour are those shown at the top of each plot. The dominant colour is set to 0.8 and the other to 0.1 and with the rule strengths, the dominant one is set to 0.025 and the other to 0.0225 and the simulation run with each combination of these. The simulation in the top left of Figure 6.11 is the same as that shown in black in Figure 6.9. The activation of the correct SNr node crosses the threshold in each case, showing that the simulations work as expected when swapping the rule and colour values. As there is symmetry in the system and it is deterministic, the plots are the same with the response reversed, as required.

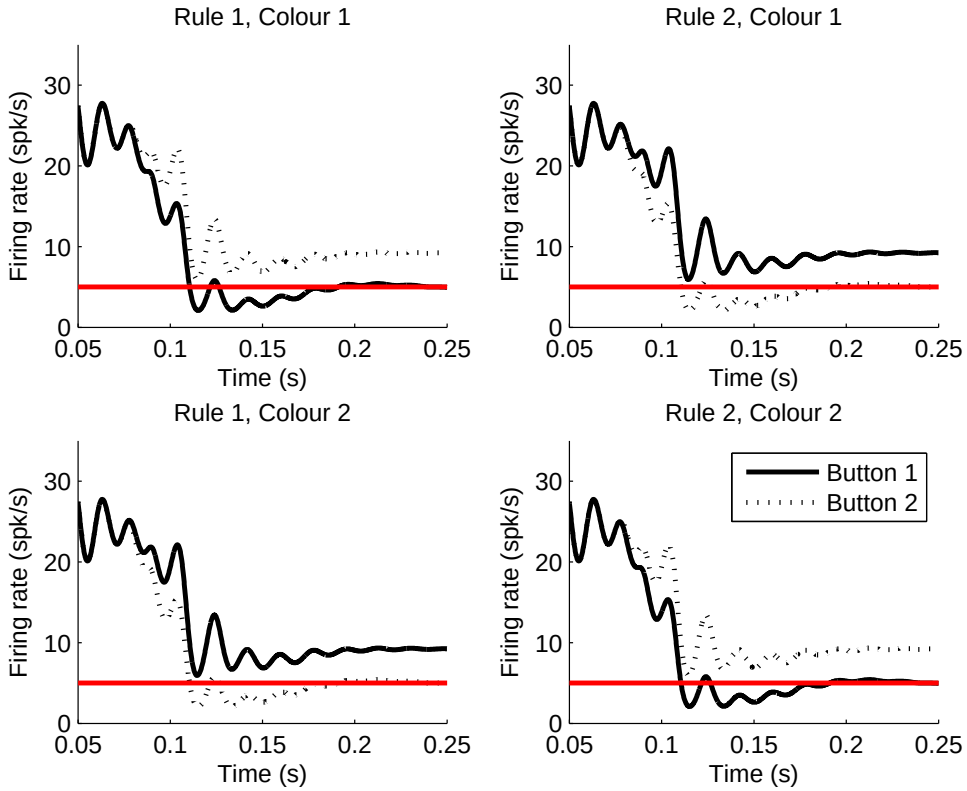


Figure 6.11: Activation of the SNr nodes when reversing the dominant rule and colour inputs to the simple basal ganglia model.

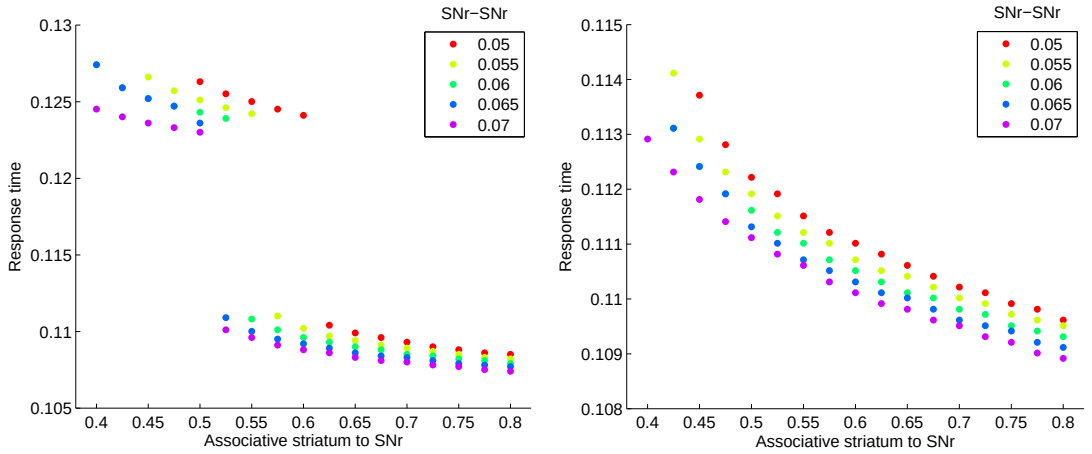
6.2.5 Testing sensitivity to parameters

So far, the simulations have all had the fixed parameters shown in Table 6.2. I now consider how the behaviour of the system changes in response to changes in some of those parameters. In this case, I keep the connections for rule and colour to be the constant values given in Table 6.7. These rule settings were chosen so as to be not the fastest, and most distinct responses of those described earlier with test A based on making the decision more difficult from test 1 and B from test 2. This was done under the assumption that the other parameters in the model would have more effect when considering the response time only when the rules were closer.

Figure 6.12 shows response times for increasing strengths of connection from the associative striatum to the SNr with colours showing different levels of mutual inhibi-

Test	Rule 1	Rule 2	Colour 1	Colour 2
A	0.0175	0.01	0.9	0.1
B	0.025	0.0225	0.8	0.1

Table 6.7: Parameters for rule and colour when testing sensitivity to other parameters in the simple basal ganglia model.



(a) Test A

(b) Test B

Figure 6.12: Response times when changing parameters involving connections to the SNr in the simulation. Test A is a more difficult decision based on test 1, and test B based on test 2.

tion from the SNr to itself. These are all inhibitory connections and the values shown in Figure 6.12 are magnitudes. In previous tests, the connections from the associative striatum to SNr and SNr to SNr have magnitudes of 0.6 and 0.06 respectively. The response times shown are all quite close even when these parameters are changed, especially for Test B shown in Figure 6.12b. Figure 6.12a shows a split in response times depending on the parameters. When the strengths of both connections tested were low enough, no response was made at all. This is not surprising as both inputs inhibit the SNr and a response is made when the output at the SNr decreases below a threshold.

Additional tests, not shown, demonstrated that the strength of the connections from the background input to the striatum and the SNr have a big effect on the model. If the background connection to the associative striatum was too low, then the input from

the cortical nodes was not enough to give a response at the striatum and so there was no response to pass to the SNr. If the connection from the background input to the SNr nodes was too strong then there was no response as the firing rate of the SNr was too high for the inhibitory connection from the associative striatum to push the SNr activation below the threshold.

6.3 Adding the STN–GPe loop to the model

6.3.1 Changing the structure of the model

The model described in Section 6.2.1, which responds correctly to the colour and rule inputs, contains only simple feedforward connections. There are no recurrent connections or other influences which might prevent a correct decision from being made. Now, I add the STN–GPe loop to the model with the STN receiving input from the cortex and projecting to the SNr, thus forming the hyperdirect pathway as described in Chapter 4 Section 4.2.1. As described in Chapter 5, even with just two nodes having recurrent connections, the STN–GPe loop can exhibit a range of behaviour including prolonged oscillations.

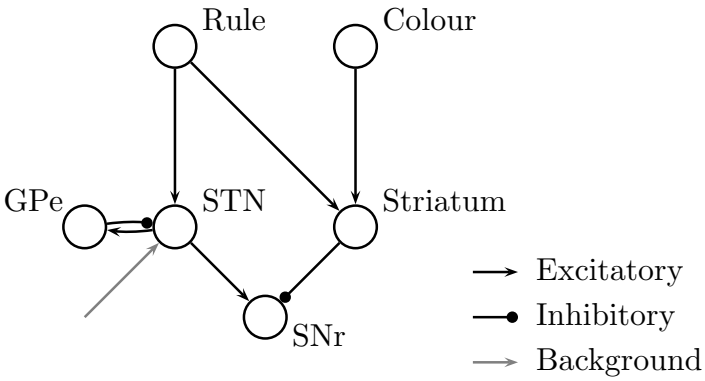


Figure 6.13: Illustration of the basal ganglia network with the GPe and STN nodes included.

Figure 6.13 shows the basal ganglia network with the STN–GPe loop added to the model of Section 6.2.1. Each of the two new regions shown in Figure 6.13 actually

represents two nodes representing the two buttons. To study the impact of the STN–GPe loop in comparison to the simple network presented in Section 6.2.1, the properties of the nodes and connections are kept as described previously for the cortex, striatum and SNr, using the parameters given in Tables 6.1, 6.2 and 6.3. The membrane time constants of the new nodes are shown in Table 6.8 and are the same as used when studying the STN–GPe loop alone in Chapter 5.

Although all the nodes have background input, that to the STN is shown in Figure 6.13 as in this model the background input to the two STN nodes may differ representing a preference for one button over the other. These different strengths of connection from a steady background input represent alternative sources of input to the STN, such as from the motor areas of the cortex.

Node	Membrane time constant (ms)
STN	6
GPe	14

Table 6.8: Membrane time constants used for the STN and GPe nodes within the basal ganglia model.

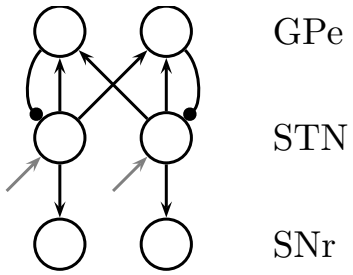


Figure 6.14: Detail of the connections between the STN and SNr as used in the model shown in Figure 6.13. Each STN node projects an excitatory signal to both of the GPe nodes but only to the corresponding SNr node. The background input shown to the STN indicates that these may have different connection strengths.

The full network for the pairs of STN and GPe nodes and their connections to the SNr are shown separately in Figure 6.14, omitting the connections from the striatum

to the SNr. The STN nodes each connect to one of the SNr nodes, which represent the two buttons, so the STN nodes can also be thought of as representing the buttons. In this model, I have used focussed connections from the STN to SNr, that is a one-to-one connection from each STN node to its corresponding SNr node. The rule nodes each connect to one STN node, however, using the network described in Section 6.2.1, the two rule nodes have identical activation as the difference between the rules is introduced by different strengths of connection from the rule to the corresponding rule node in the specific striatum. The rule nodes do, however, have additional input at the time of stimulus onset, so this means that additional input is also passed to the STN at that time.

In Figure 6.14 the connections from a steady background input to the STN are shown, the efficacy of the connection from the background to the STN may be different for the two nodes. This difference represents a predisposition to press one of the two buttons before the stimulus is actually seen. Where there is a difference in the background connections to the STN nodes, the tests are repeated with the connections reversed so that tests are run with bias which separately promotes or impedes the correct response.

Each STN node connects to both the corresponding and opposite GPe nodes. This is based on the idea of both focussed and diffuse connections from the STN to the GPe as implemented by Thibeault & Srinivasa (2013). I will use focussed to describe the connection from the STN to its corresponding GPe node and diffuse to describe the connection from the STN to the opposite GPe node. The connection delay between the STN and GPe is set to 6 ms in each direction and that between the STN and SNr as 2 ms, these delays are as used by Thibeault & Srinivasa (2013). For each of the new links connecting nodes in the system, the number of simulated connections and the delays in those connections are shown in Table 6.9. These parameters will be kept constant for all the simulations described, different strengths of connections will be modelled by using different efficacies which will be stated for individual tests.

6.3.2 Activation of the SNr with influence from the STN

Having set up the model as described above and using rule and colour parameters as for the three tests shown in Figure 6.9, I used the parameters shown in Table 6.10 for the connections relating to the STN and GPe nodes. This gave the results shown in Figure 6.15 where the blue lines show the activation of the SNr nodes when the

Connection	Number of Connections	Delay (ms)
Rule–STN	1250	6
STN–GPe (diffuse)	625	6
STN–GPe (focussed)	625	6
Background–STN	3000	0
GPe–STN	1250	6
Background–GPe	3000	0
STN–SNr	1250	2

Table 6.9: Standard parameters for testing the influence of the STN–GPe loop when added to the basal ganglia model.

background input to the STN is lower for the STN node connected to the SNr node representing the button giving the correct response than to the other STN node, which I refer to as bias in favour of a correct response. The shading of the blue lines corresponds to the shading of the black and grey lines showing the SNr activation in the simulation without input from the STN for three sets of rule strengths.

Connection	Efficacy
STN–GPe (diffuse)	0.05
STN–GPe (focussed)	0.05
Background–STN	3
Background–STN	3.025
GPe–STN	−0.005
Background–GPe	1
Rule–STN	0.05
STN–SNr	0.0175

Table 6.10: Synaptic efficacies for testing the influence of activity from the STN.

The tests shown in Figure 6.9 were repeated with the connection strengths from the background to the STN reversed, however, the qualitative details of the profile of the activation and the times that the activation crossed the 5 spk/s threshold were almost the same as those shown in blue.

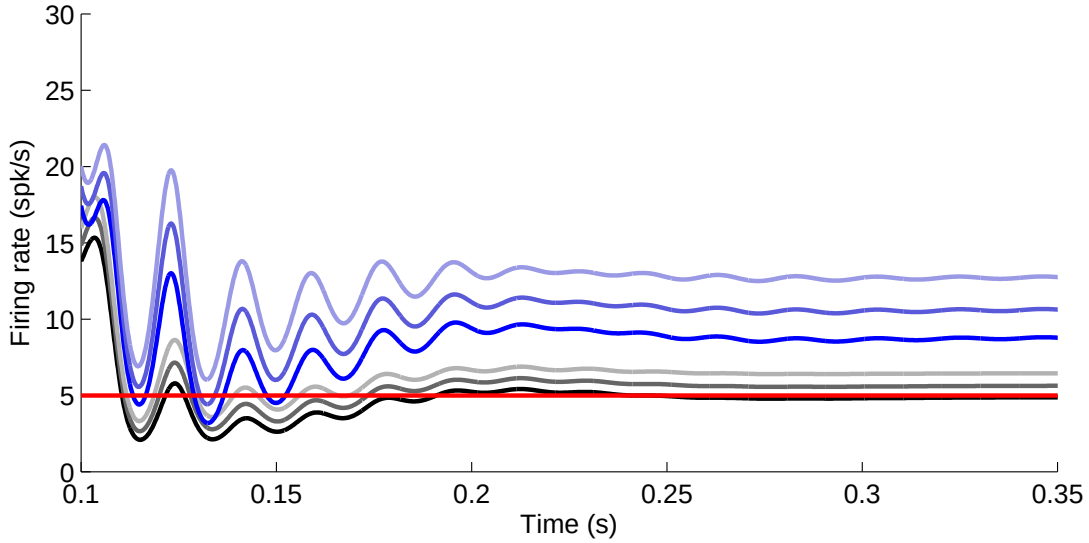


Figure 6.15: Activation of the SNr shown in blue is influenced by excitatory connections from the STN. The activation of the SNr nodes without the influence of the STN shown in black and grey is as in Figure 6.9.

The parameters shown in Table 6.10 give activation of the STN in which the initial oscillations die out. It is clear from the blue lines in Figure 6.15 that in this case with excitatory input from the STN, the responses given by the firing of the SNr nodes are less strong as the minimum firing rate of the SNr node is higher and the time when the firing rate is below the 5 spk/s threshold is shorter. In particular, the palest blue line does not cross the threshold at all, indicating that a response is not made although in the equivalent simulation without the influence of the excitatory input from the STN, a response was made, at 0.112 s. For the middle of the three simulations in Figure 6.15, the addition of the input from the STN delays the response to the next dip in the oscillations in the firing rate of the SNr node. Without the influence of the STN, the response is made at 0.111 s and with this influence, the response is made at 0.131 s. For the darkest simulations shown in Figure 6.15 the difference between the response times with and without the STN is much smaller, 0.111 s and 0.113 s without and with input from the STN respectively. Figure 6.15 shows that both without input from the STN and with input in which the oscillations of the STN die out, the firing rates of the SNr nodes settle to a steady rate after initial oscillations.

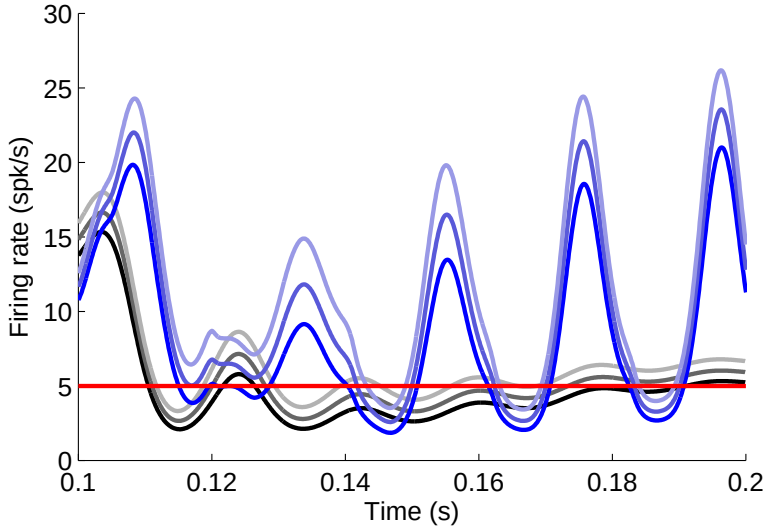


Figure 6.16: Activation of the SNr shown in blue is influenced by oscillatory input from the STN.

As the influence of the input from the STN is to excite the SNr and a decision is made when the output of the SNr dips below a threshold, then it seems intuitive that adding the influence of the STN increase the firing rate at the SNr and thus delay or prevent a response as shown here.

The simulations shown in Figure 6.15 are now repeated with just two changes to the connection parameters in order to make the activation of the STN nodes continue to oscillate. In Chapter 5, I described how oscillations can be maintained in the STN–GPe loop when the connection strengths are higher. Using the parameters in Table 6.10 and changing the STN to GPe focussed connection to 0.105 and the GPe to STN connection to -0.045, results in the simulations shown in Figure 6.16.

In Figure 6.16, we can see that, for each of the rule settings shown, there is a clear response, the activation of the SNr node shown in blue dips below the 5 spk/s threshold. The timings of the responses can be much different from those in the underlying simulation without the influence of the STN. For the lightest of the plots in Figure 6.16 the response was made at 0.112 s without input from the STN and with oscillatory input from the STN the response is now made at 0.143 s.

Figures 6.15 and 6.16 show examples of the possible impact of the STN on responses where the underlying simulations are based on making the decision more difficult from

Connection	Efficacy
STN–GPe (diffuse)	0.05
STN–GPe (focussed)	0.105
Background–STN	3
Background–STN	3.05
GPe–STN	−0.045
Background–GPe	1
Rule–STN	0.05
STN–SNr	0.0175

Table 6.11: Synaptic efficacies for testing the influence of the STN–GPe loop.

test 2. When the rules are made closer starting from test 1, as shown in Figure 6.6, there are situations in which no response was made at all as the firing rate at the SNr did not cross the 5 spk/s threshold. I now look at the impact of input from the STN in this situation using the underlying simulations shown as the lightest three in Figure 6.6. The STN–GPe loop connections were set using the parameters shown in Table 6.11 which, as used for Figure 6.16 above, cause the STN activation to continue to oscillate.

Figure 6.17 shows that with the influence of excitatory oscillatory input from the STN, the activation of the SNr shown in blue clearly dips below the 5 spk/s threshold and so a response is made. This is the case even for the palest blue line, where the corresponding simulation without the effect of input from the STN, shown in the palest grey, does not give a response.

I now look at the change to the population density profiles of the SNr node corresponding to the correct response with and without the influence of input from the STN in the simulations shown in Figure 6.17 to see how the addition of excitatory input from the STN can lead to a response being made when previously none was.

In Figures 6.18a and 6.18b the population densities of the SNr node, respectively without and with the influence of input from the STN, appear similar up to 0.128 s. From 0.13 s to approximately 0.138 s the main peak of population density is further to the right, that is nearer to the threshold potential, in Figure 6.18b than in Figure 6.18a so the firing rate corresponding to Figure 6.18b is higher during this period. From 0.136 s, in Figure 6.18b, we see a portion of the membrane potential distribution which

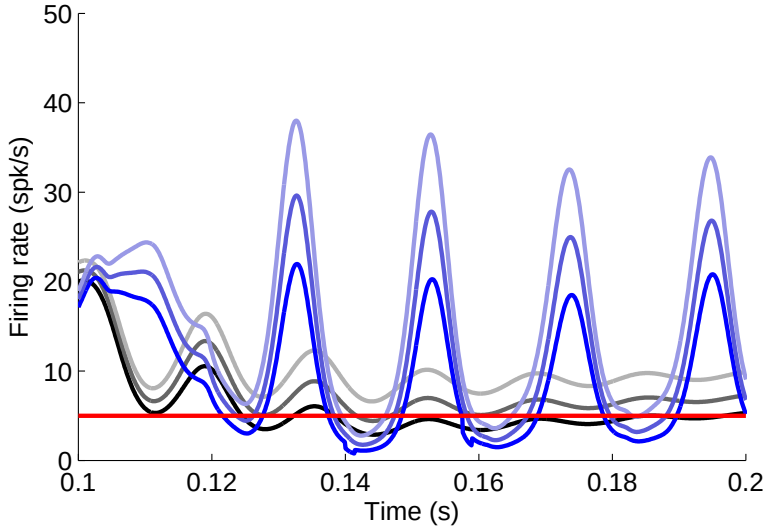
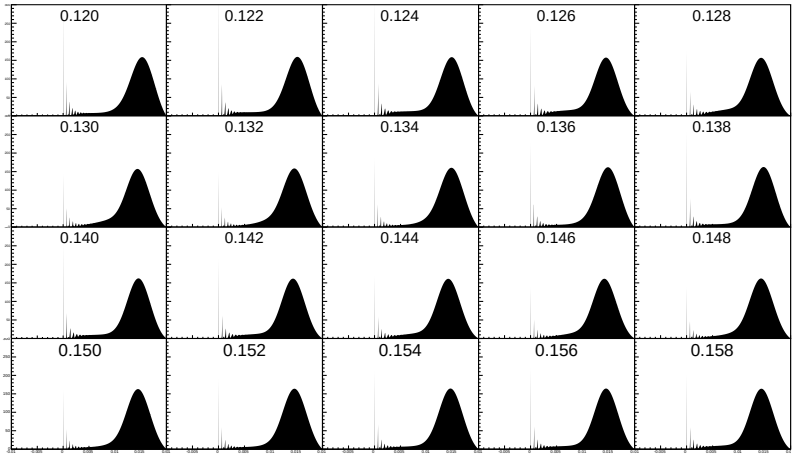


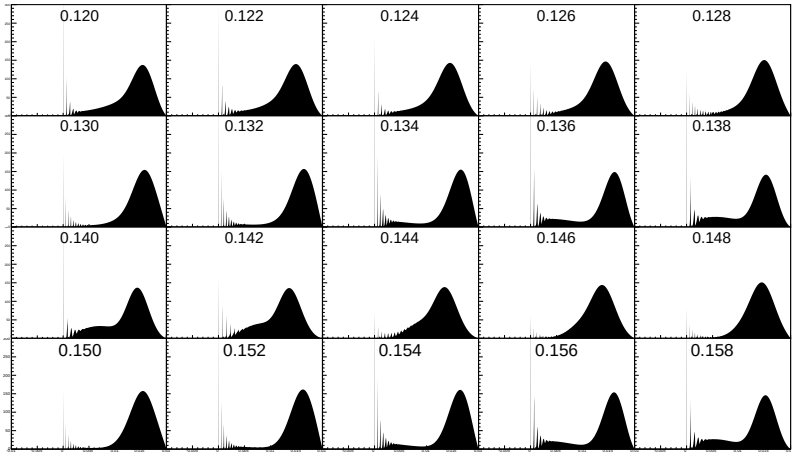
Figure 6.17: Activation of the SNr shown in blue has oscillatory input from the STN. The simulations shown in black and grey do not have input from the STN and are as previously shown in Figure 6.6.

is nearer to the reset potential, representing neurons which have fired and been re-introduced. This means that the population density close to the threshold potential is lower and the firing rate is then lower, becoming low enough to cross the threshold of 5 spk/s.

So far, although I have introduced a bias so that the two STN nodes have different input connections from the steady background and I have run the simulations with those connections both ways round, I have only illustrated the activation of the SNr when the bias at the STN was such as to favour a correct response, shown in blue in Figures 6.15, 6.16 and 6.17. Now, using the parameters shown in Table 6.12, I show a pair of simulations with a larger difference in input between the input to the two STN nodes, and thus a larger difference between the activity of the two STN nodes. In Figure 6.19, as before, the blue lines represent the output of the SNr nodes when the bias at the STN favours the correct response with the activity of the correct SNr node dipping below the threshold. The green lines show the activation of the SNr nodes when the difference between the STN nodes favours the incorrect response. The activity of the SNr shown as a solid line is that representing the correct response, and the dotted line represents an incorrect response.



(a) Without STN influence as the lightest grey plot in 6.17.



(b) With STN influence as the lightest blue plot in 6.17.

Figure 6.18: Population densities for the SNr node at the times indicated for simulations shown in 6.17

Figure 6.19 shows, in black, a simulation in which the activation of the SNr nodes representing both responses drops below the 5 spk/s threshold in the original simulation without any input from the STN to the SNr. This is based on the parameters 0.024 and 0.0235 for rule 1 and rule 2 and 0.8 and 0.1 for colour 1 and colour 2 respectively, shown in light grey in Figure 6.9. In the simulation shown in Figure 6.19, when the bias at the STN favours an incorrect response, the SNr firing rate for the node corresponding to the incorrect response dips below the threshold at 0.112 s of simulation, before that

Connection	Efficacy
STN–GPe (diffuse)	0.05
STN–GPe (focussed)	0.105
Background–STN	2.8
Background–STN	2.95
GPe–STN	−0.045
Background–GPe	1
Rule–STN	0.05
STN–SNr	0.0125

Table 6.12: Synaptic efficacies for testing the influence of input from the STN to the SNr where there is a bigger difference between the input to the two STN nodes than in previous tests.

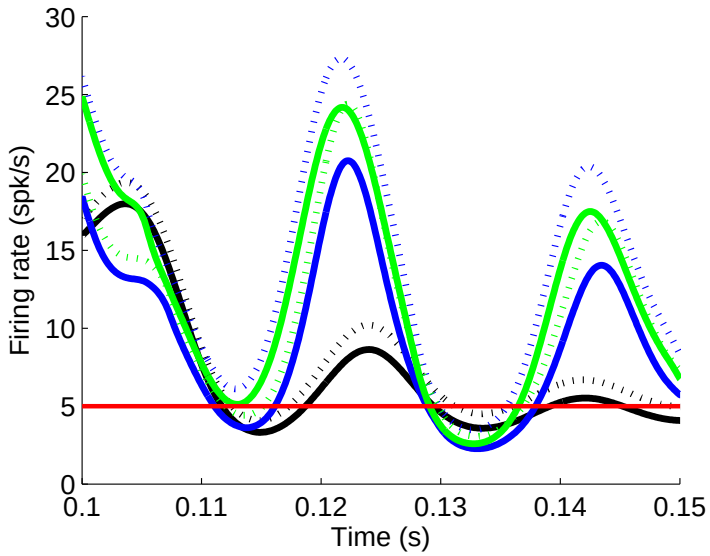


Figure 6.19: Activation of the SNr node corresponding to the correct response is shown by solid lines and that corresponding to the incorrect response is shown as dotted lines. The black lines show a simulation without input from the STN to the SNr, as shown in light grey in Figure 6.9. The blue and green lines show bias at the STN in favour of and against a correct response respectively, using the parameters given in Table 6.12.

for the correct response (0.129 s), so this is taken to be an incorrect response. When the bias at STN favours a correct response then a correct response is made. Figure 6.19 uses the parameters in Table 6.12 where we see that the difference between the inputs to the two STN nodes is larger than for the other tests shown. This difference has been enough to trigger an incorrect response in the case where the bias hinders the correct response.

6.3.3 Response times with input from the STN

Previously, in Section 6.2.3, I examined how the response time varied with the difference between the two rules, finding that there were steps in response times as shown in Figures 6.7 and 6.10. I now look at the response times for the same sets of rule differences, but with the addition of input from the STN to the SNr. To do so, I use two sets of parameters for the connections of the STN and GPe nodes. One set of parameters shown in Table 6.10 allows the oscillations in the STN node to die out and gives the results shown by blue diamonds in Figures 6.20 and 6.21. The other set of parameters has the STN to GPe focussed connection set to 0.105 and the GPe to STN connection set to -0.045 and all other parameters as in Table 6.10. This second set of parameters causes oscillations to be maintained in the firing of the STN and the response times are shown as red circles in Figures 6.20 and 6.21. In each case the timings shown are only for the case where the STN influence is against the correct response. In Figures 6.20 and 6.21, the black crosses indicate the response times presented in Section 6.2.3 without the STN.

Figure 6.20 shows results based on making the decision more difficult starting from test 1, in each case the correct response was made despite there being a bias against the correct response from the STN. Where a response is not shown when the rule difference is small, then this simulation did not give a response as the firing rate of neither SNr node fell below the threshold. The blue diamonds, where the STN output does not continue to oscillate, all show a slower response time than the corresponding response times without input from the STN, shown as black crosses. In addition, a much bigger rule difference is needed in order for a response to be made compared to the simulation without input from the STN. The red circles, with the influence of oscillating output from the STN, show a faster response time for smaller rule differences, but a slower response time for larger rule differences. In addition, this STN influence enables the

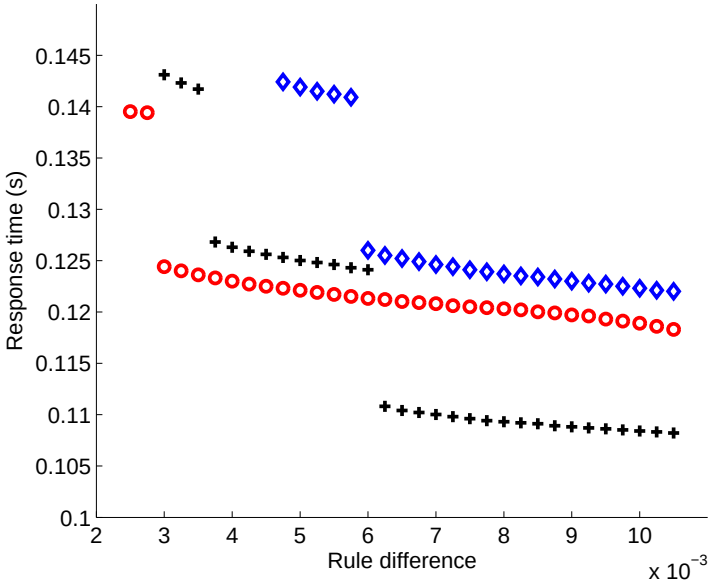


Figure 6.20: Response times against difference in strength between the two rules making the decision more difficult starting from test 1. Black crosses show the simulations without the influence of the STN, blue diamonds with influence of activation of the STN which reaches a steady firing rate and red circles for simulations with continuing oscillations in the activity of the STN.

correct response to be made for some smaller differences in rules than was the case with the basic simulation. In each case there is still a step in response times when changing rule difference.

Figure 6.21 shows the results of the same STN–GPe parameters applied to rule differences based on test 2. These simulations have more activation in the system which results in generally faster response times. Also, the initial test 2 allows for a wider range of rule differences. In Figure 6.21, again the responses with unsustained oscillations in output of the STN, shown as blue diamonds, are slower than the responses without any STN influence. For the smallest rule difference, no response is made in this condition. For the continuing oscillations in the activity of the STN, for some of the simulations the response was slower and for some faster than the basic simulation.

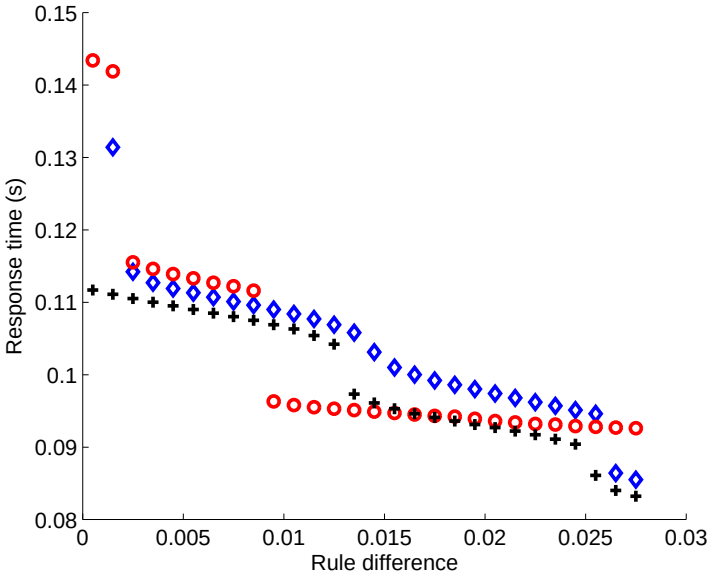


Figure 6.21: Response times against difference in strength between the two rules making the decision more difficult starting from test 2. The symbols and colours are as in Figure 6.20.

6.3.4 Changing the connection from the STN to the GPe

So far, I have only used two different sets of parameters for the connections between the STN and GPe such that the activation of the STN continues to oscillate or reaches a steady state. I have used these parameters with different rule settings and different connections between the background and STN nodes. I now look at how changing the efficacies of the connections from the STN to the GPe affects the activity of the SNr when keeping all other parameters in the simulations constant using the parameters in Table 6.13.

As a basic simulation to test the effect of changing the connections between the STN and the GPe, I use the middle simulation from Figure 6.9 which was also used in Figures 6.15 and 6.16 with the influence of input from the STN to the SNr. The parameters in Table 6.13 were used only with the bias at the STN favouring an incorrect output. For each successive test, the focussed connection from the STN to the GPe is increased by 0.01, until reaching 0.095, and the resulting activation of the SNr node corresponding to the correct response shown in Figure 6.22.

Connection	Efficacy
STN–GPe (diffuse)	0.05
STN–GPe (focussed)	0.055
Background–STN	3
Background–STN	3.025
GPe–STN	−0.035
Background–GPe	1
STN–SNr	0.0175

Table 6.13: Parameters for testing the effect of changing the strength of the STN–GPe connection only.

Figure 6.22 shows that changing the strength of connection from the STN to GPe makes a difference to both the timing and amplitude of the resulting oscillations at the SNr. The correct response was made in each simulation despite the small bias at the STN being towards the incorrect response. The response times, at which the SNr activation dips below 5 spk/s were, for increasing the STN to GPe efficacy, 0.112, 0.128, 0.127, 0.143 and 0.115 s. There is a wide variation in response times which do not follow a simple pattern based on the changes in the STN to GPe connections. This suggests that the model is sensitive to the parameters used and I would expect that it would also be sensitive to other parameters.

6.4 Discussion

6.4.1 Summary of results

In this chapter I have demonstrated a simple model of action selection in the basal ganglia. Firstly, I implemented a feedforward network, representing the direct pathway in the basal ganglia described in Chapter 4. Unlike published computational models of the basal ganglia, I implemented my model using population density techniques as described in Chapter 5.

I used my simulated basal ganglia to make decisions based on the psychological task of Bland & Schaefer (2011) described in Chapter 2. Making a decision in the basal

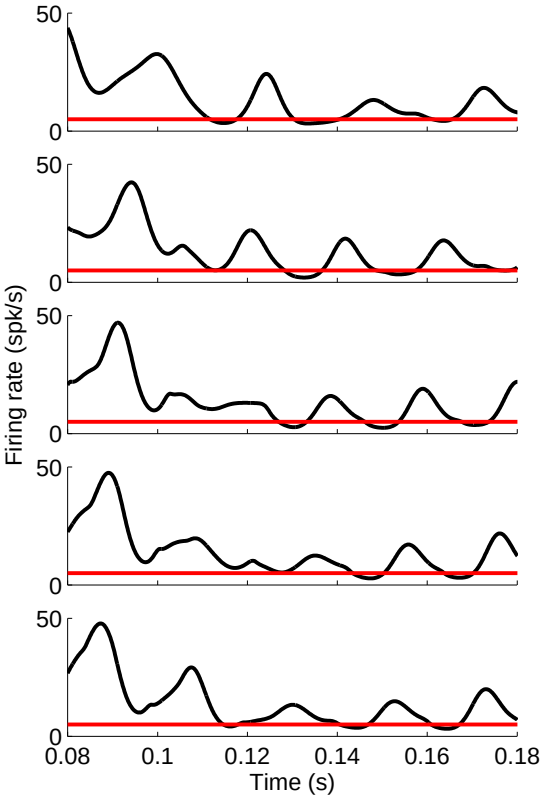


Figure 6.22: Top plot: Activation for the SNr node corresponding to the button for the correct response using the parameters given in Table 6.13. Subsequent plots: Increasing the strength of the focussed connection from the STN to GPe by 0.01 for each successive test.

ganglia model required the combination of a representation of a belief as to environmental state which applied, referred to as a rule strength, and a visual stimulus of a colour. I showed how the timing of the responses made by the model varied according to the difficulty of the decision, modelled as a smaller difference between the belief in each underlying rule. I found that there is not a simple linear relation between rule strengths and response times using my simple model. I have shown how this relates to the oscillations which occur initially as a signal is passed from one node to another in a neural system.

As described in Chapter 4, the hyperdirect pathway forms another channel for information flow through the basal ganglia. I created the hyperdirect pathway by adding an

additional node, the STN, to the simple direct pathway basal ganglia model. The STN takes input from the cortex and is also connected to the GPe forming an excitatory-inhibitory circuit which can display stable or oscillating behaviour, as investigated in Chapter 5. I examined the ability of the model to select the correct action under different conditions of activity of the STN.

I found that if the oscillations in the activity of the STN have subsided before the signal arrives through the direct pathway to the basal ganglia output, the SNr, then adding the excitatory connection from the STN to the SNr can delay or prevent a response from being made, as illustrated in Figure 6.15. When the STN exhibits sustained oscillations then these can allow a decision to be made where one was not made without the influence of input from the STN as shown in Figure 6.17. This gives a possible advantage of oscillatory output from the STN during normal activity.

A bias from the STN towards an incorrect response can produce an incorrect output as shown in Figure 6.19. Whether an incorrect action is produced depends on the difference between the strengths of belief in the two rules and on the strength of the bias at the STN. This will be investigated further in Chapter 7.

6.4.2 Assumptions and limitations

Even though I have only modelled a simplified basal ganglia model for action selection, the parameters chosen for the simulations reported only cover a small fraction of the possible parameter settings. I have only presented two different starting points for considering rule and colour settings. These allow different amounts of excitatory input into the system. For test 2 with a higher maximum rule difference, I lowered the parameter for colour 2 compared to that for test 1 so as to reduce difference in overall activation between the two tests. I have experimented with more parameter settings than those presented here and find the results to be qualitatively similar.

I have tried to focus on just a small number of parameters to look at the influence of specific connections on the behaviour of the model. When changing the connections in the model, I have changed only the efficacy and not the number of effective connections. As described in Chapter 5 Section 5.5, both the number of effective connections and the efficacy should be changed to vary the mean and standard deviation of the input signal. As indicated in Section 6.3.4, the model could be very sensitive to the parameters relating to the STN–GPe loop, but only a small part of this parameter space has

been investigated. Aside from the rule and colour settings and bias at the STN, the parameters are identical for corresponding pathways. An alternative approach would be to add random noise to the input to each node, as done by Guthrie et al. (2013) and Schroll et al. (2012). This approach would have required additional parameters in the model to control the level of noise and additional simulations to see the overall effects.

In all the simulations shown, I have provided additional input to the system at 0.05 s at the time I consider the stimulus to have been presented in the behavioural task. Although the task consists of a colour stimulus which would imply changing the input to the cortical colour node only, additional input is fed to the rule node. The additional signal to the rule nodes represents engagement with the task and an assumption that the appearance of the colour signals that a decision has to be made. The cortical rule nodes excite the STN although excitation of the STN when a decision needs to be made could originate from other brain areas. In addition, both the rule and colour nodes need time for the population densities to develop from the initial state of all neurons having a single membrane potential.

This work only includes some of the known pathways in the basal ganglia. As with Guthrie et al. (2013) I only include the direct and hyperdirect pathways. Unlike Guthrie et al. (2013), I include the GPe node in a loop with the STN. I have used the simple feedforward only model as a baseline upon which to examine the impact of activity of the STN on the system. I have taken the STN–GPe loop to be part of the hyperdirect pathway only rather than including the indirect pathway. In the indirect pathway, the GPe receives inhibitory input from the striatum. It is known that both the direct and indirect pathways are active during movement initiation (Nelson & Kreitzer, 2014). I have simulated the direct and hyperdirect pathway only so that I could keep my model simple and examine the impact of excitatory input from the STN to SNr.

As with Guthrie et al. (2013), I use an association area in the striatum to combine information. N'Guyen et al. (2014) raise a concern that this method does not scale up to cater for large numbers of possible inputs. In the model of N'Guyen et al. (2014), described in Chapter 4, multiple inputs converge on the SNr, they found however, that the model was not able to successfully respond to combinations of inputs. I show that the association area I have implemented can correctly combine signals, the main focus of my investigations is the impact of excitatory input from the STN on the SNr activation. I do not rule out other mechanisms by which signals could be combined.

It is not clear what constitutes a response neurally and computational models involve many simplifications from real biological systems, so each modeller has to choose a plausible mechanism to determine a response. I have followed Humphries et al. (2006) and used a threshold of 5 spk/s as the firing rate below which the firing rate of an SNr node has to fall for a decision to be achieved. I did not implement nodes to receive the tonic inhibitory input from the SNr such that a drop in firing rate from the SNr disinhibits the next node in the system. Whether the next node actually would fire would depend on other factors which are not included in my simulations, such as the delay in transmission of the signal and the membrane time constant of the nodes. This means that it would not always be the case that the following node would always fire when the SNr firing rate falls below the threshold as it may need that dip to last longer to start firing. I have also taken the simplistic approach that the first node to cross the threshold triggers a response even when there is only a very small time difference before the other node falls below the threshold. My assumption is that there could be some other mechanism which can respond very quickly and ensure that only one response is selected but that this mechanism is beyond the scope of my investigations. This is where my approach differs from that of Humphries et al. (2006). Humphries et al. (2006) create populations of individual spiking neurons. They determine the average firing rate for the whole population using a moving time window. This gives them the possibility that more than one response will be selected.

Some computational models (e.g. Frank, 2005) allow a system to reach a stable state and then compare the activation of different nodes when in that stable state to determine whether a decision has been made. My approach of considering the first of the SNr nodes for which the activation falls below 5 spk/s means that the initial oscillations which occur as a signal is passed from one node to another become an important feature of the decision making.

Using my simple basal ganglia model I found that there were steps in what I describe as the response time, that is the time at which the firing rate of one SNr node falls below the threshold of 5 spk/s. A decision having been made at the basal ganglia is only one part of the decision time in a behavioural context. The work presented here does not consider how the decision at the basal ganglia is transformed to a motor output in terms of a button press. Even if a step in response time is seen when the activation at the SNr is considered, I would expect there to be random differences occurring in the rest

of the process to convert the decision into an action so that steps would not be likely to be seen in the behavioural reaction time.

These simulations are deterministic, the results are identical every time they are run. I chose to set the weights in the simulation to fixed values, in most cases identical for each channel so that I could examine the effects of systematically changing some of the weights.

One difference between my model and many other computational models of the basal ganglia is that I have not included diffuse connections from the STN to SNr as included by, for example, Frank (2006) and Gurney et al. (2001) as described in Chapter 4. There is a school of thought that the STN inhibits all actions by increasing the activation of all SNr nodes and this has helped to develop models of interaction between pathways in the basal ganglia giving an explanation for the function of each pathway. Other researchers suggest that the underlying biological connections are topographic in nature, (Brodal, 2010; DeLong & Wichmann, 2010). The computational models of Chakravarthy and colleagues use focussed projections from the STN to SNr (e.g. Kalva et al., 2012).

In my modelling, I have shown that it is possible to produce a lower firing rate at the SNr by the addition of excitatory input from the STN. Oscillatory excitatory input from the STN to the SNr can force the output of the SNr to also oscillate giving periods at which the firing rate of the SNr is lower than it would be without the input from the STN. If, as I consider, short dips in the firing rate of the SNr can result in a decision, this can result in a decision being made when otherwise a decision would not be made. In the psychological task I study, participants lose 10 points if no response is made, so it is better to guess, giving a chance of being correct, than to not answer at all. A mechanism to force a response could be provided by oscillatory input from the STN to the SNr. Although I have used focussed connections from the STN to the SNr and always implemented asymmetric connections from the background input to the two STN nodes, I believe that the effect of producing a response through oscillations in the STN output could also be achieved, in the same way as shown in Figure 6.18, when the STN globally excites the SNr.

In setting the parameters for the simulations, I have always set the additional input to both the rule and colour nodes to be at the same time as each other. This means that the timings of the oscillations which occur as a signal is passed through the basal ganglia nodes always have the same relation to each other. As the STN can have a large

effect on the timing of the response, I would expect that different timings of the signals could also have a big effect on the responses.

In treating each trial as a completely separate event, I have represented stimuli by increasing the background input to the cortical nodes at a particular point in time. This input then remains at the higher level for the remainder of the simulations. Humphries et al. (2006) used an alternative approach in which input to one channel increased and at a later time input to the other channel was increased to a level above that of the first. Humphries et al. (2006) showed that their model could correctly switch response.

As described in Chapter 4, information processing in the basal ganglia is considered to take place in loops so that the decision made at the SNr is fed back to the cortex. I have not implemented such a loop in my model. This is a common approach in models which consider each individual trial as a separate event and do not look at how brain activity is maintained or changes between trials, for example Krishnan et al. (2011) and Humphries et al. (2012).

I have implemented a bias at the STN by means of connections from a background input. This bias could originate in other brain areas or due to changes in other connections.

6.5 Conclusions

I have shown that population density modelling can be used to model cognitive processes. I have shown how the response of the model changes when the inputs are varied and considered how input from the STN changes the responses. I found that the impact of the STN on responses varied according to whether the output from the STN was oscillatory or not. This leads me to suggest a benefit of oscillatory activity from the STN in that responses can be made in situations in which no response would be made without this input. In Chapter 7, I further examine the role of the STN, whilst incorporating learning into the model.

Chapter 7

Learning

7.1 Introduction

In Chapter 6, I examined a simple neural model of the basal ganglia and the addition of the STN–GPe loop to that model. The model responds according to stimuli based on individual trials of the psychological task of Bland & Schaefer (2011) described in Chapter 2. However, the neural model had no mechanism for learning a belief in the underlying state of the environment. I now move towards using the neural model to simulate the reinforcement learning model developed in Chapter 3 to describe human behaviour.

The reinforcement learning model consists of two parts: updates using a learning rate parameter to an underlying belief in which rule currently applies, and probabilistic selection of an action using a temperature parameter used in a softmax rule which controls the randomness of the responses. In this chapter, I take an approach of applying the belief updates given by reinforcement learning directly to the rule strengths in the model, allowing me to focus on how the action selection mechanism might be implemented in the basal ganglia. I propose a mechanism by which connections in the model may implement different temperatures in softmax action selection.

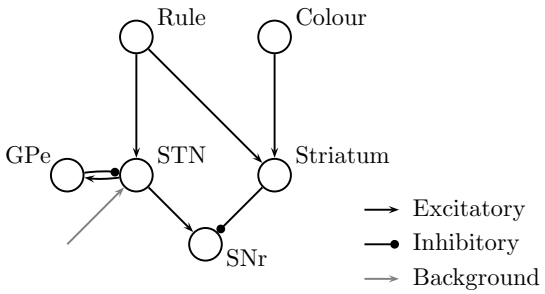
I simulate a sequence of trials and on each trial I use the model's output to determine the response made and use the outcome of whether the response was correct or not to update connections between nodes in the model between trials. I focus on the possibility that the STN–GPe loop acts as a source of randomness, as did Krishnan et al. (2011) described in Chapter 4. Using sequences of trials, I run the model multiple times using

different parameters to represent the learning rate and temperature parameters from the reinforcement learning model. I record the behaviour of the model on each trial, and use these behavioural responses to estimate the reinforcement learning parameters underlying that behaviour in the same way as for the human behavioural data described in Chapter 3.

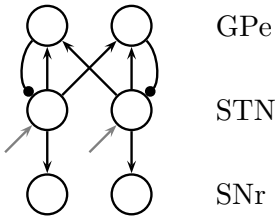
7.2 The STN as a source of randomness

7.2.1 Influence of the STN on decision accuracy

In Chapter 6, I started to examine how the difference between the background input to the two STN nodes in the basal ganglia model could lead to an incorrect response. In Figure 6.19 when the asymmetry between the inputs was against the correct response, the wrong response was made. When the asymmetry at the STN was in favour of the correct response, the correct response was made which I now assume to be true in general and investigate conditions under which asymmetry actually produces an incorrect response. Keeping the input from the background to one of the STN nodes constant, I look at the accuracy of the response when varying the difference in the rule strength and in the difference between the background input to the two STN nodes, which I refer to as the bias at the STN.



(a) Relationship between nodes.



(b) Connections involving the STN.

Figure 7.1: Basal ganglia network used for learning experiments.

The decision making network presented is repeated in Figure 7.1 for convenience. For these investigations, most of the weights for the connections are held constant with the

values used in Chapter 6 in Tables 6.1, 6.2, 6.3, 6.8 and 6.9 with additional connections at 0.05 s of simulation to represent the stimuli as before. The remaining parameter values which are fixed for these experiments are shown in Table 7.1, these parameters produce sustained oscillatory activity in the STN–GPe loop. As in Chapter 6, a response is indicated by the first of the two SNr nodes for which the activation falls below the threshold of 5 spk/s.

Connection	Efficacy
STN–SNr	0.02
STN–GPe (diffuse)	0.05
STN–GPe (focussed)	0.105
GPe–STN	−0.045
Rule–STN	0.05

Table 7.1: Parameters which are kept unchanged for the learning experiments.

Following the investigations into the effects of adding the STN–GPe loop to the decision making model, I chose to use the simulation shown in black in Figure 6.9 as the basic simulation in this chapter and make the rules closer from there. That is, I take the minimum and maximum rule strengths to be 0.0225 and 0.025 respectively and set the colour to 0.8 for the colour shown and 0.1 for the colour not shown. These settings are based on making the decision more difficult from test 2 from Table 6.4 in Chapter 6. I adjust the rule strengths by increasing one and decreasing the other by the same amount. I scale the difference between the two rule strengths so that the maximum difference, which is 0.0025 using the parameters described above, is set to 1.

The connections from the background to the STN nodes give a bias to the response as each STN node connects to just one of the two SNr nodes as shown in Figure 7.1b. The background input to the STN node connected to the SNr node representing the incorrect response is set to 2.5 for each of the tests. The background input to the other STN node is set to various levels up to 3. In general, a higher output at the STN will excite the corresponding SNr node. Under the ideas of Frank (2006) of the STN providing a NoGo signal, this would reduce the chance of the activation of the SNr dipping below a threshold and creating a response. The interactions between the

populations are complex and, as shown in Chapter 6, it is possible for excitatory input to result in a response. As with the difference in rule strengths, I scale the difference between the background input to the two STN nodes to be between 0 and 1 and refer to this scaled difference as the level of scaled bias at the STN.

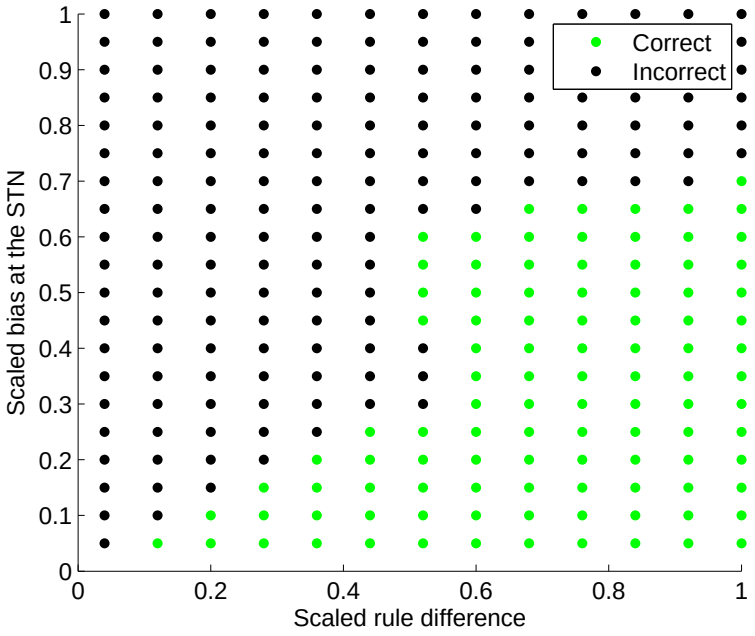


Figure 7.2: Accuracy of the decision made by the model according to the scaled difference between the rules and scaled bias at the STN.

Figure 7.2 shows whether the response was correct or not for combinations of scaled rule differences and scaled bias at the STN as described above. When the rule difference is small, a small bias towards the incorrect response will cause the incorrect response to be taken and when the difference between the rules is large, a large bias is needed to force the incorrect response. For most of the rule differences used, there is a single level of difference in bias such that if the difference is smaller than this the correct answer is given but otherwise the incorrect response is made. This is not the case for the scaled rule difference of 0.52 where for increasing difference in bias, the response changes from correct to incorrect then correct again and incorrect again. To try to understand why this happens, firstly I show in Figure 7.3 the output of the two STN nodes for some different bias settings.

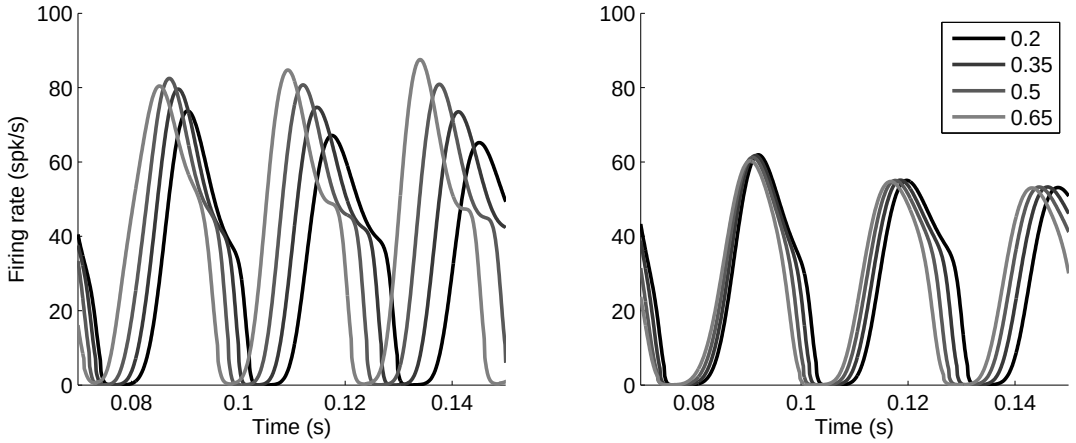


Figure 7.3: Activation of the two STN nodes for increasing bias at the STN. The left output is projected to the SNr node representing the correct response and the right output is projected to the SNr node representing an incorrect response. The grey levels show different levels of scaled bias at the STN.

The asymmetry between the output of the two STN nodes is shown as the difference between the left and right plots in Figure 7.3. The peaks of the activation shown in the left plot, which shows the output which will be passed to the SNr node corresponding to a correct response, are higher than the peaks in the right plot. Noting that the output from the STN is excitatory will therefore tend to increase the firing rate at the corresponding SNr node and that a decision is made when the SNr output falls below a threshold, the higher peaks influencing the correct response will discourage that response. The relationship is not as simple as that, as we saw in Chapter 6 excitatory oscillatory input from the STN to the SNr can encourage a response. The level of complexity in the interactions leads me to examine the output of the simulations for specific parameter values.

The right hand plot in Figure 7.3 shows the output of the STN node which has a constant background input of 2.5 for the different levels of bias. Note that Figure 7.3 shows that the peaks of activation of this node have the same amplitude, but that the timings of these peaks varies according to the level of bias between the two STN nodes, especially later in the simulation. Note that in the left plot of Figure 7.3, it is clear that both the amplitude and the timings of the peaks differ according to the level of bias.

Figure 7.4 shows details of the firing rates of the SNr nodes for simulations with the scaled rule difference set to 0.52 and the four different levels of STN bias shown in Figure 7.3. The grey levels of the four plots in Figure 7.4 match those in Figure 7.3 showing the level of STN bias. The plots in Figure 7.4 each show the activation of the SNr node corresponding to the correct response as a solid line and that corresponding to the incorrect response as a dotted line, focussing on the time period when a response is made. In each of the plots in Figure 7.4, the input from the striatum to the SNr is identical, as this only depends on the colour and rule settings which were unchanged in these simulations.

At the first dip in firing rates of the SNr nodes shown in Figure 7.4, for the lowest level of STN bias, shown in the top left, the firing rate for the SNr node corresponding to the correct answer clearly goes below the 5 spk/s first before that corresponding to the incorrect answer. For the plots shown top right and bottom right, the firing rate for the node corresponding to the incorrect response cross the threshold so the wrong action is made. For the simulation shown in the bottom left, the firing rate dips close to but not below the threshold so no response is made at that time. In each case, when the firing rates dip a second time, the activation corresponding to the correct response clearly crosses the threshold before that for the incorrect response, so the correct response is made in the bottom left plot.

7.2.2 Relating the influence of the STN to the softmax temperature

In order to consider whether the influence of the activity of the STN on decisions made at the SNr in the basal ganglia could produce responses which show similarity to softmax action selection, I first examine softmax action selection more closely than in Chapter 3. I repeat the formula for softmax action selection, for the situation where there are two alternatives. Given a belief, B, that one environmental state applies, where B is between 0 and 1 and the belief in the other response is $1 - B$, and a temperature parameter T which is greater than zero, the probability, P, of response in line with that environmental state is given by

$$P = \frac{e^{\frac{B}{T}}}{e^{\frac{B}{T}} + e^{\frac{1-B}{T}}}.$$

Figure 7.5 shows how the probabilities of an action depend on the underlying belief and the temperature, T. When the beliefs in each option are equal, at 0.5, then the

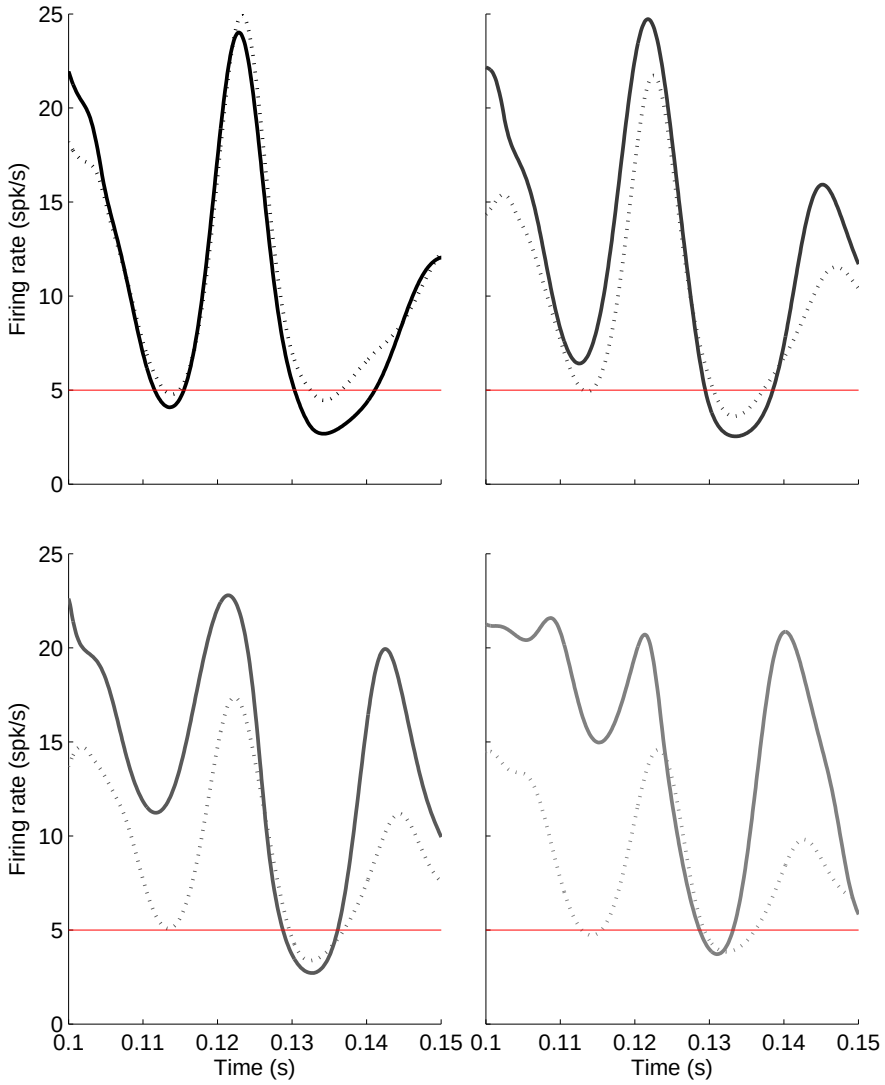


Figure 7.4: Activation of the two SNr nodes for different levels of bias at STN, shown in the same grey levels as in Figure 7.3, for the scaled rule difference of 0.52. Solid lines show the activity of the SNr node representing the correct answer and dotted lines the incorrect answer.

probability of selecting each response is also 0.5 regardless of the value of the temperature. When the temperature is low, 0.05 shown in blue in Figure 7.5, the probability of responding in alignment with the belief quickly becomes 1 as the belief moves away from 0.5. When the temperature is high, even when the belief in rule 1 is 1, thereby

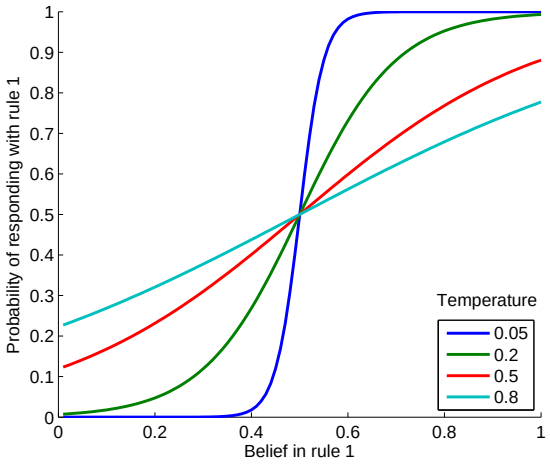


Figure 7.5: Influence of the temperature parameter in softmax on the probability of responding in accordance with the underlying belief

having no belief in rule 2, the probability of selecting rule 1 is still under 80%, having much more randomness in responses.

To potentially relate softmax action selection to the effect of the STN, I first take the approximation that the boundary between correct and incorrect responses shown in Figure 7.2 can be represented by a straight line passing through the origin and the point representing a scaled rule difference of 1 and a scaled bias at the STN of 0.7. Using this simplification to a linear boundary, I created simulations using MATLAB (2012) in which the bias at STN was calculated by sampling the background input to each STN node from a normal distribution with mean zero and standard deviations shown in Figure 7.6. The accuracy of the response was calculated without using the neural simulation, using the assumption that when the bias at the STN is in favour of the correct response a correct response will be made. When the bias at the STN is in favour of the incorrect response, I assume that the response will be correct when the scaled bias is below the linear boundary between correct and incorrect responses for that scaled rule difference. For each of the four standard deviations shown in Figure 7.6, the process of selecting weights and calculating the accuracy was repeated 10,000 times for each of 21 equally spaced scaled rule differences in the (0,1) interval and the percentage of correct responses calculated.

Figure 7.6 considers only the difference between the two rules taken in one direction,

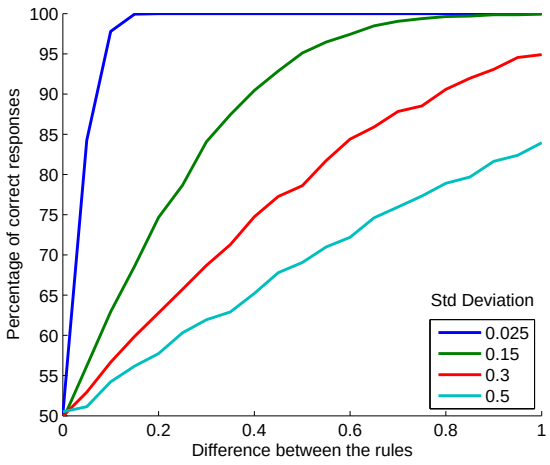


Figure 7.6: Modelling the response made dependent on the difference between the rules and the difference of two values sampled from normal distributions.

so only relates to beliefs from 0.5 to 1 which appear in the top right of Figure 7.5. It is clear that the curves shown in Figures 7.6 and 7.5 have the same shapes. This demonstrates that softmax behaviour can be achieved in this model by randomly selecting the connections to the STN nodes from normal distributions. The temperature in softmax is controlled by the standard deviations of the distributions from which to select the connections. A higher temperature is represented by a higher standard deviation.

7.3 Adding learning to the basal ganglia model

7.3.1 Defining a test process

I have indicated above that it may be possible to use the bias at the STN to implement softmax action selection. This was based on some assumptions; I made a simplification to a linear relation between the rule difference and the STN bias to give the boundary between correct and incorrect decision. As described above, this relation was not linear when using the neural model. Also the relation between accurate and inaccurate responses according to rule difference and the bias at the STN shown in Figure 7.2 was based on keeping the background input to one STN node constant. I assumed that, in the range of connection strengths used, the difference between the

input to the two STN nodes is more important than the absolute values of the two connection strengths.

I now return to the neural population simulations and use reinforcement learning to calculate the rule strengths, with two different learning rate parameters according to whether the previous trial was a win or a loss. I select weights for the background connections to the two STN nodes from normal distributions with a mean of zero and the standard deviation which is also a parameter dependent on whether the previous trial was a win or a loss. This will implement the reinforcement learning model which I found to be the best fit to human behaviour of the machine learning approaches tested in Chapter 3.

A run is formed from a sequence of trials where each trial takes the form of the trials shown in Chapter 6, but in this case weights in the model are amended between each trial. The only weights which are amended are those which connect the rule nodes to the specific striatum and those which supply background input to the STN. The connections from the background to the colour are set to simulate a colour presented on each trial.

7.3.2 Updating weights

Here, I give a recap of reinforcement learning as described in Chapter 3 and explain how it has been applied to the rule strengths in the computational basal ganglia model. The environment is assumed to be coupled so whichever button is correct on a trial, the other button is incorrect. This means that an agent only needs to maintain a predicted value of the outcome for one response type, I use response type 1. Suppose $Q(t)$ is the predicted value of using response type 1, on trial t and that $R(t)$ is the reward associated with response type 1. The reward value can be determined directly from the colour shown and correct response and does not depend on the response actually selected. If colour 1 is shown and button 1 is correct or colour 2 shown and button 2 correct then R is set to 1, indicating that a type 1 response was correct on that trial, in other cases R is set to 0. At each trial a prediction error, $\delta(t)$, is calculated as the difference between the reward and the predicted value of a type 1 response

$$\delta(t) = R(t) - Q(t).$$

This prediction error, $\delta(t)$, is used to update the expected value $Q(t)$ for the next trial, using a learning rate, α, with a value between 0 and 1 as follows

$$Q(t+1) = Q(t) + \alpha\delta(t).$$

As with the best fitting model to human behaviour, I use two learning rates according to whether the previous response was correct or not. The predicted value is converted directly into a rule strength, that is the efficacy for the connection between cortical rule node and the corresponding node in the specific striatum. Keeping rule strengths within the range described above, between 0.0225 and 0.025, connection strengths can be set from the Q value by setting rule 1 to $0.0225 + Q/400$ and rule 2 to $0.025 - Q/400$.

For setting the connection strengths from the background input to the STN, the following process is used. Random numbers are selected from a standard normal distribution for each of the two connections. These random numbers are scaled by a 'temperature' parameter which is set according to whether the previous trial was a win or a loss. The resulting values are added to 2.5 to give the efficacies of the connections from the background to the STN nodes.

7.3.3 Test data

Using aspects of the environment presented to the human participants by Bland & Schaefer (2011), I created test data using MATLAB (2012). Each set of data generated consists of 360 trials in blocks of 120. As with the psychological study, blocks of trials are considered to be stable or volatile. Stable blocks have no switch in underlying rule and volatile blocks have a rule switch every 30 trials. Stable and volatile blocks are alternated to form a set of 360 trials, giving two sets of conditions, starting with stable and volatile. These two conditions were repeated by starting with the opposite underlying rule to give four sets of 360 trials. Each trial consists of a colour shown, coded as 1 or 2 and the number of the correct button for that trial. Each trial has a 50% chance of each colour being shown and a 73.3% chance of feedback being given in alignment with the underlying rule. This data corresponds to the low feedback validity condition used with the human participants.

7.3.4 Simulation

The test data produced was presented to the computational basal ganglia set up as in Section 7.2 and using the sets of parameters shown in Tables 7.2 and 7.3 to determine the weight changes between trials. Note that these temperature parameters are not the same as the temperature in the softmax calculation, but are used to set the standard deviation of the distribution from which the connections from the background to the STN connections are set. Each set of learning rate parameters was paired with each set of temperature parameters and was tested against each of the four sets of test data described in Section 7.3.3 and the process carried out twice with different seeds for the random number generator for the STN weights.

	a	b
Learning rate after loss	0.4	0.25
Learning rate after win	0.8	0.5

Table 7.2: Learning rates used to update rule strengths for simulations of sequences of trials.

	1	2	3	4
Temperature after win	0.06	0.06	0.08	0.08
Temperature after loss	0.13	0.16	0.13	0.16

Table 7.3: Parameters used to set the connections from the background to the STN nodes as described in Section 7.3.2.

7.3.5 Results

Figure 7.7 shows some sequences of responses made by the neural simulations using just three of the possible eight combinations of parameters from Tables 7.2 and 7.3. In Figure 7.7, the actions are represented in terms of responses of one of two types in exactly the same way as for the human participants as shown in Chapter 2. Unshaded areas in Figure 7.7 show that the underlying rule in the environment is rule 1, that is

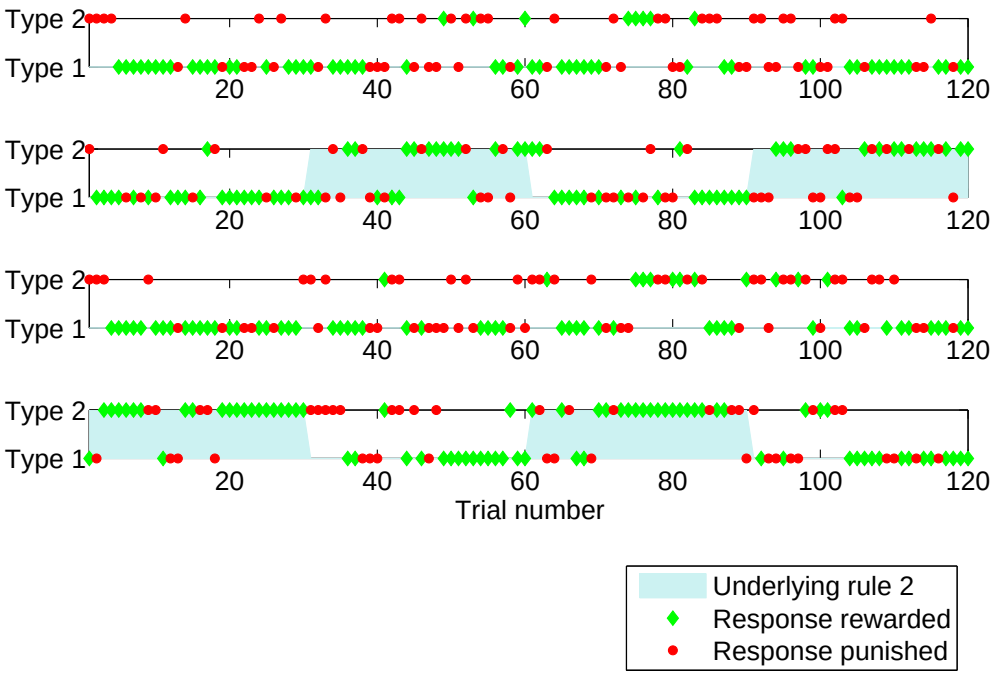


Figure 7.7: Examples of responses generated by neural simulations.

that type 1 responses are rewarded with high probability. Shaded areas show that rule 2 applies. Responses which gain (lose) points are shown in green (red).

The top two simulations in Figure 7.7 use learning rate parameters a from Table 7.2 and the bottom two use b. The second simulation uses parameters 3 from Table 7.3, the others use parameters 4. That is, the bottom two plots have the same parameter sets but different random seeds for the responses and different test data. In Figure 7.7 we can see features which align with responses made by human participants. Most of the occasions where the opposite response type is made to that on the previous trial follow incorrect responses, shown in red. There are, however, trials where an incorrect response is not followed by a switch and trials, for example trial 60 in the top plot of Figure 7.7 where a switch is made although the previous response is correct.

Using the sequence of actions made by the neural simulation to each set of test data, the process described in Chapter 3 was carried out to estimate the model parameters in the same way as for human responses. Figure 7.8 shows the estimated learning rates from the responses, showing the results using all the different temperature parameters.

As the learning rates were used directly to set connection strengths in the model, it would be expected that the responses would show features of those learning rates, but as the responses depend on the activation of nodes in the simulation, this needs to be checked.

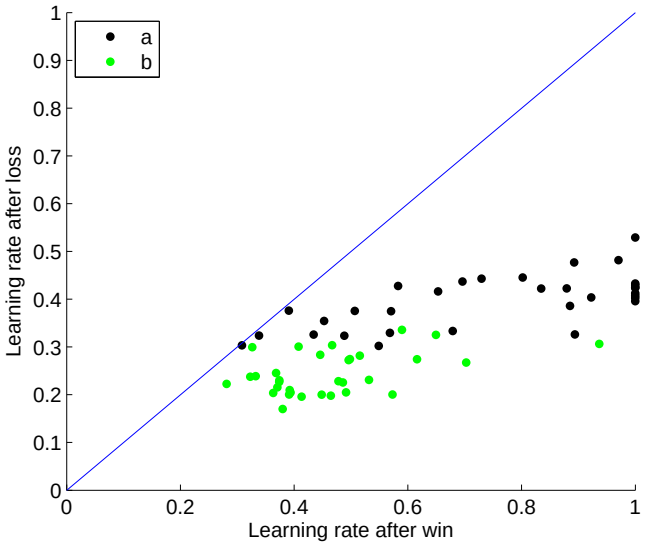


Figure 7.8: Estimated learning rates after a win and a loss for responses generated by neural simulations.

In Figure 7.8, we see that the learning rates have all been correctly estimated as having higher values after a win than a loss and that, on average, the estimated learning rates for test a are lower than for test b, using the parameters shown in Table 7.2. For test b, the estimated learning rates after a win are more spread than after a loss and for test a.

Figure 7.9 shows the estimated temperatures for the neural simulations including both sets of learning rate parameters with each of the combinations of parameters indicated. Tests 1 and 2 in Table 7.2 have a lower temperature after a win than tests 3 and 4. In Figure 7.9, for learning rate parameters a, the fit temperature parameters after a win for tests 1 and 2, shown in black and grey, are lower than for tests 3 and 4 shown in blue. The same applies with learning rate parameters b, comparing red and pink to green. For tests 1 and 2, the equivalent fit temperatures after a win are higher when in conjunction with learning rate parameters b than with learning rate parameters a.

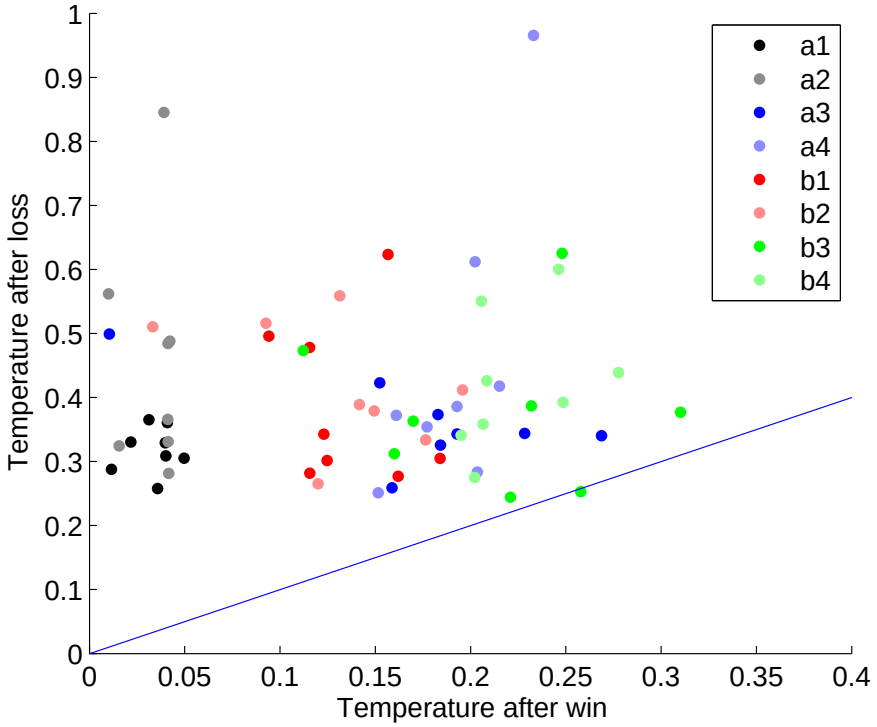


Figure 7.9: Estimated temperatures after a win and a loss for responses generated by neural simulations where the key indicates the combination of parameters from Tables 7.2 and 7.3.

This implies that there is interaction between the effects of the parameters. The effect is less pronounced for tests 3 and 4.

The parameters used for tests 1 and 3 have a lower temperature after a loss than tests 2 and 4. The simulations using these parameters are shown in Figure 7.9 using dark and light markers respectively. In Figure 7.9 there is no apparent difference in the spread of the light and dark markers. This is the case for both sets of learning rate parameters.

So far in this chapter, I have considered only whether a response was considered correct or not based on the first SNr node for which the output fell below the threshold of 5 spk/s. I now look at how the timings of these responses vary according to the underlying belief which was used to set the weights for the connections from the cortical rule node to the specific striatum. I use the higher of the Q values calculated by

reinforcement learning for each response type and consider whether the response was made in accordance with or against that higher Q value and plot the response times separately. As in Chapter 6, I use the total time from the start of the simulation to represent the response time, rather than the time after the stimulus was applied.

Figure 7.10 shows the response times for all the individual trials using the learning rate parameters shown as a in Table 7.2 with responses against the underlying belief shown on the left and those in accordance with the underlying belief on the right.

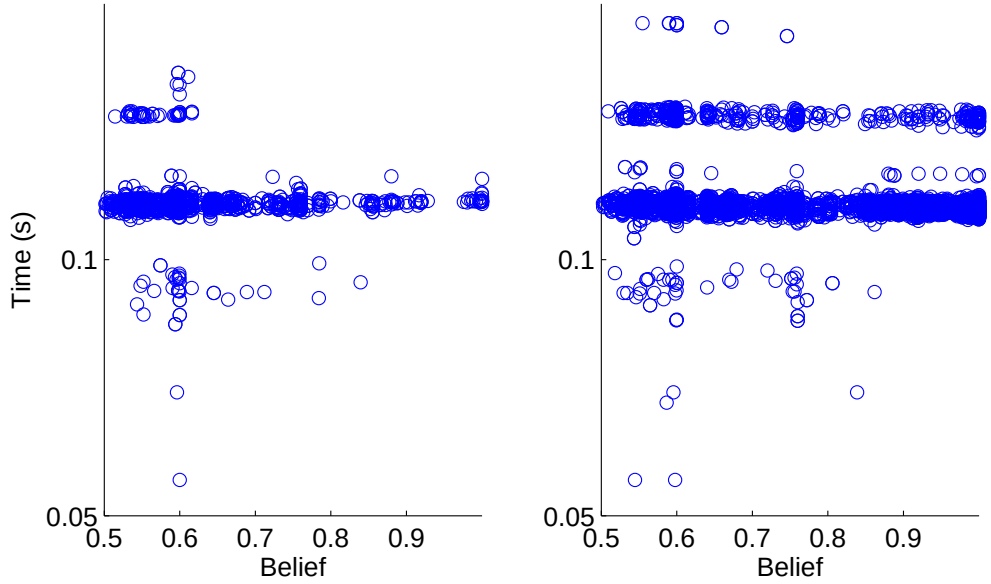


Figure 7.10: Left: response times where response was opposite to the current belief. Right: response times where response was in accordance with the current belief.

In Figure 7.10 it is clear that responses occur at particular bands of time. As all the connection weights are identical for each trial in the simulation apart from those representing the current colour, the rule and the background to the STN, the timing of the responses will depend on the oscillations in the SNr nodes as described in Chapter 6. The distribution of response times is clearly different for responses against the underlying belief shown on the left in Figure 7.10 compared to those in accordance with the underlying belief shown on the right. In particular, when the belief is higher than 0.65, there are no responses against that belief with a response time above 0.12 s, whereas there are responses in accordance with the belief showing these response times.

7.4 Discussion

7.4.1 Summary of results

In this chapter I have proposed a mechanism by which input from the STN to the SNr could give responses which have characteristics of softmax action selection. I have simulated learning tasks which involve sequences of trials, determined the action selected from the output of neural populations and adjusted connection parameters in the model to take account of feedback of whether a response was correct or not. I examine the responses of the model in the same way as I did for human responses in the psychological task.

7.4.2 Relation to other work

I have adjusted the rule strengths exactly as in reinforcement learning by maintaining an abstract Q value which is then converted into weights. This is a similar approach to learning as taken by N'Guyen et al. (2014). It is known that there is plasticity in the connections between the cortex and the striatum and so there is biological motivation for changing those weights through learning. In my model the action taken is determined by the activity of the neural populations and the feedback based on that action determines which learning rate to use on the following trial so there are still aspects of reinforcement learning being determined by the neural simulation.

Applying an abstract update rule in order to change connection weights is not a biologically realistic means of implementing reinforcement learning in the basal ganglia. As described in Chapter 4, levels of the neurotransmitter dopamine are considered to be important in reinforcement learning in the brain. Other researchers simulate dopamine levels within their computational models of the basal ganglia, giving more biological realism (e.g. Gurney et al., 2015; Schroll et al., 2012).

I have focussed on whether the STN could be the source of an exploratory signal. Kalva et al. (2012) also investigate the STN as the basis for exploration. They model the STN–GPe loop as part of the indirect pathway, without providing cortical input to the STN. They suggest that the output of the STN has to be chaotic in order to produce exploratory behaviour. I have suggested that oscillatory activity might be advantageous, but I have not considered possible chaotic behaviour in this chapter. Kalva et al. (2012) describe the effects on their model of varying the level of dopamine in the system, which

changes the characteristics of neurons in the STN, GPe and striatum, and how this can lead to exploration. When describing the ability of their model to produce behaviour which resembles that of participants, however, they only change the characteristics of the D1 neurons in the striatum. They find that they can give realistic overall percentages of exploitation with different thresholds for the striatal D1 neurons.

I have only briefly examined response time of my model and the response time which I take to be the time at which the SNr activation falls below a 5 spk/s threshold is only a part of the behavioural response time. Although I have not analysed the response times of the participants in the psychological task, response times are often considered to be important in such tasks (see e.g. Pleskac & Busemeyer, 2010). Pleskac & Busemeyer (2010) discuss empirical observations which should be able to be explained by models of cognitive processes. One of these observations is that when the decision is easy, mean decision times are shorter for incorrect responses than for correct responses. The results I show in Figure 7.10 suggest that this would be the case with my model.

7.4.3 Assumptions and limitations

In previous chapters all the simulations I have shown have been entirely deterministic, they run the same way every time. In this chapter the learning simulations have randomness at two different points, in the creation of the test data, and in the selection of strengths for the connections from the background to the STN. Fitting parameters to data requires a long sequence of data in order to be reliable. The number of trials used in a run, 360, and the number of runs with different selections of random numbers may not be enough to give a reliable estimate for the reinforcement learning parameters.

I have only considered two different parameter sets for learning rate and four for temperature. This has allowed me to consider whether my ideas are plausible as a mechanism for introducing randomness into decisions. I saw some possible interaction between the parameters, with the estimates for temperature after win varying with the learning rate parameters. The differences between the fit temperatures after a win according to the parameters set give an indication that this method may be able to model softmax action selection. The fit temperatures after a win were low compared to those fit to human behaviour. Different parameters representing temperature after loss would have to be chosen to see if a split could be shown between high and low values. I have not made any suggestion as to how the connection strengths for the input to

the STN nodes may be chosen from a normal distribution on each trial or how changes based on feedback influence the standard deviation of that distribution. The changes would not have to be in the input to the STN, it might be possible to achieve a similar effect through changing the excitability, or ease of response to stimuli, of the STN. This change could be controlled by the level of dopamine, as suggested by Kalva et al. (2012).

I have not considered how the decision making would change with different parameters for the connections between the STN and the GPe. As described in Chapter 5, these parameters can determine whether the output of the STN is oscillatory or not. In Chapter 6, I found that decisions were delayed or prevented when the activity of the STN settled to a steady state. I have not looked at the balance between these two scenarios.

This study treats each individual trial as a separate event with changes to connections occurring only between the trials. Real-life is not split into events in this way. In the psychological task, when feedback was given it was merely in terms of whether the response made was correct or not, there was no reminder as to the colour shown or the button selected. To create a more realistic simulation, these aspects would have to be maintained within the neural activation so that the updates could be applied correctly. One way to allow signals to be maintained would be to implement the loop which feeds back from the basal ganglia to the cortex, as Schroll et al. (2012) point out this could allow memory and allows delayed response tasks and timings so less dependent on artificial nature of time split into individual trials.

7.5 Conclusions

I have given some initial indications that the impact of the input from the STN to the SNr could form the source of softmax action selection. I developed this idea separately from the neural simulations having made many simplifying assumptions on the behaviour of the system. I implemented the idea in my basal ganglia model, getting results which merit further investigations along the same lines.

Chapter 8

Conclusions

8.1 Summary

In this thesis I have taken two contrasting approaches to modelling human behaviour from a psychological task; machine learning and population density techniques for modelling neural systems. As Cohen & Frank (2009) point out, it is unusual for one researcher to combine both these approaches. Using machine learning to model human behaviour in the psychological task, I fit different models to the behaviour of the participants in the study of Bland & Schaefer (2011) and compared how different styles of model were able to fit the behaviour. I also tested the models against each other as ideal agents which tried to gain as many rewards as possible without trying to model human behaviour.

I used population density techniques to investigate interactions between neural populations. I showed models which built up from two interacting populations to a model of the basal ganglia. In my basal ganglia model, a decision was taken to be indicated by the first of two populations for which the activity of that population dipped below a threshold.

8.2 Contributions

Although asymmetric learning rates had been used to model human behaviour (Frank et al., 2007), my published work (Duffin et al., 2014), presented in Chapter 3 of this thesis, was the first to do so when comparing reinforcement learning to Bayesian

models in their ability to fit human behaviour. In addition, unlike other work, my models have asymmetry in the temperature parameters used for action selection.

I made a contribution to understanding the flexibility of reinforcement learning by showing that asymmetric learning rates gave better performance than a single learning rate.

My work in Chapter 5 shows some of the advantages of population density modelling as a technique for neural simulations. In particular, I have shown how the underlying distribution of membrane potentials relates to complex structures in the patterns of firing rates of a simple circuit consisting of two reciprocally connected nodes, one excitatory and one inhibitory.

In Chapter 6, I described a neural simulation of the basal ganglia using a simple network which could make decisions in simple scenarios inspired by the psychological task. These simple decisions allowed me to examine action selection in situations of different levels of difficulty without considering any underlying learning. I found that there were jumps in the response time of my model as I made the decision gradually more difficult

I showed some potential impact of adding the STN–GPe loop to the basal ganglia network. In particular, I showed that if a decision is considered to have been made when the output at the SNr dips below a set threshold, then additional excitatory input from the STN to the SNr can cause the SNr output to dip below that threshold when the STN activity is oscillatory and induces oscillations in the activity of the SNr. This shows a potentially important reason for oscillatory activity to occur at the STN.

In Chapter 7, I gave an indication of how the input from the STN to the SNr could contribute to variation which is observed in human decisions. In particular, I proposed a mechanism by which softmax action selection might be implemented in the brain.

8.3 Ideas for future work

I have made suggestions for the role of the STN within decision making. In Chapter 6, I showed the potential for the oscillating activity of the STN to force a decision to be made in a situation in which none was made otherwise. In Chapter 7, I gave an initial indication that changes to the input to the STN could allow the random selection of responses above an underlying belief in line with the softmax rule for action

selection. Some of these findings were based on modelling which was done separately from the neural system. This allows much scope for further modelling using the neural system. In particular, the simulations should be extended to decisions from more than two options. My proposal that input from the STN to the SNr can produce responses which can be characterised by softmax action select should be tested with more options. The probabilistic relation of the response to the underlying belief should relate correctly for different probabilities across more options.

The mechanism implemented for learning in the neural system was to update the weights, or connection strengths, between trials in an artificial way. An extension to my work would be to implement learning through simulation of neuromodulation. In computational models, these changes are often implemented in the connections between the cortex and striatum, as in the recent work of Gurney et al. (2015). As well as allowing learning through changing the strengths of connections, neuromodulation can also change the excitability of neurons. These changes could be involved in changing the activity of the STN and so altering the activity of the STN from stable to oscillatory for example. My model should be extended to include neuromodulation and to study its impact on decisions.

In Chapter 7 I described how my model related to an observed feature of behavioural resonse times which Pleskac & Busemeyer (2010) state should be able to be explained by models of decision making. One of the other factors listed by Pleskac & Busemeyer (2010) is the speed-accuracy trade-off. This describes the finding that when placing participants under pressure to respond as quickly as possible, the accuracy of their responses decreases (see e.g. Bogacz et al., 2010). Decision making showing the speed-accuracy trade-off is observed in tasks in which the correct response is well known to the participant but there is uncertainty in the environment and the participant has to accumulate sensory evidence to determine the environmental state. A commonly used decision making task is that of motion discrimination, in which participants report the dominant direction of motion of a display of moving dots. I would like to study the speed-accuracy trade off using my basal ganglia model. A commonly held belief is that the STN contributes to this phenomenon by delaying a response and allowing increased accuracy (Bogacz et al., 2010; Frank, 2006). Frank (2006) believes that the STN imparts a global NoGo signal. In my modelling, when its output is oscillatory, the

STN can encourage a response. I would like to take this further to investigate how my interpretation of the contribution of the STN influences speed-accuracy trade-off.

An important aspect of future work to follow from my computational modelling would be to work with neuroscientists to examine how well my predictions match biological reality. If my findings reflect biology, this could make a significant contribution to understanding the value of oscillatory neural activity during normal functioning.

References

Alexander, G.E. & Crutcher, M.D. (1990). Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends in Neurosciences*, **13**, 266–271.

Alexander, G.E., DeLong, M.R. & Strick, P.L. (1986). Parallel organization of functionally segregated circuits linking basal ganglia and cortex. *Annual Review of Neuroscience*, **9**, 357–381.

Amit, D.J. & Brunel, N. (1997). Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cerebral Cortex*, **7**, 237–252.

Baldassarre, G., Mannella, F., Fiore, V.G., Redgrave, P., Gurney, K. & Mirolli, M. (2013). Intrinsically motivated action-outcome learning and goal-based action recall: A system-level bio-constrained computational model. *Neural Networks*, **41**, 168–187.

Behrens, T., Woolrich, M., Walton, M. & Rushworth, M. (2007). Learning the value of information in an uncertain world. *Nature Neuroscience*, **10**, 1214–1221.

Bland, A. & Schaefer, A. (2011). Electrophysiological correlates of decision making under varying levels of uncertainty. *Brain Research*, **1417**, 55–66.

Bland, A. & Schaefer, A. (2012). Different varieties of uncertainty in human decision-making. *Frontiers in Neuroscience*, **6**.

Bogacz, R., Wagenmakers, E., Forstmann, B. & Nieuwenhuis, S. (2010). The neural basis of the speed-accuracy tradeoff. *Trends in Neurosciences*, **33**, 10–16.

Boraud, T., Brown, P., Goldberg, J.A., Graybiel, A.M. & Magill, P.J. (2005). Oscillations in the basal ganglia: The good, the bad, and the unexpected. In J. Bolam, C.A. Ingham & P.J. Magill, eds., *The Basal Ganglia VIII*, vol. 56 of *Advances in Behavioral Biology*, 1–24, Springer US.

Botvinick, M., Niv, Y. & Barto, A. (2009). Hierarchically organized behavior and its neural foundations: a reinforcement learning perspective. *Cognition*, **113**, 262–280.

Brodal, P. (2010). *The central nervous system: structure and function*. Oxford University Press, Oxford.

Brunel, N. & van Rossum, M. (2007). Quantitative investigations of electrical nerve excitation treated as polarization. *Biological Cybernetics*, **97**, 341–349.

Calabresi, P., Picconi, B., Tozzi, A., Ghiglieri, V. & Filippo, M.D. (2014). Direct and indirect pathways of basal ganglia: a critical reappraisal. *Nature Neuroscience*, **17**, 1022–1030.

Cazé, R.D. & van der Meer, M.A. (2013). Adaptive properties of differential learning rates for positive and negative outcomes. *Biological Cybernetics*, **107**, 711–719.

Charness, G. & Levin, D. (2005). When optimal choices feel wrong: A laboratory study of bayesian updating, complexity, and affect. *American Economic Review*, **95**, 1300–1309.

Chersi, F., Mirolli, M., Pezzulo, G. & Baldassarre, G. (2013). A spiking neuron model of the cortico-basal ganglia circuits for goal-directed and habitual action learning. *Neural Networks*, **41**, 212–224.

Chevalier, G. & Deniau, J. (1990). Disinhibition as a basic process in the expression of striatal functions. *Trends in Neurosciences*, **13**, 277–280.

Cohen, J., McClure, S. & Yu, A. (2007). Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Phil. Trans. R. Soc. B*, **362**, 933–942.

Cohen, M. & Frank, M. (2009). Neurocomputational models of basal ganglia function in learning, memory and choice. *Behavioural Brain Research*, **199**, 141–156.

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

Dayan, P. & Daw, N.D. (2008). Decision theory, reinforcement learning, and the brain. *Cognitive, affective and behavioral neuroscience*, **8**, 429–453.

de Kamps, M. (2006). An analytic solution of the reentrant poisson master equation and its application in the simulation of large groups of spiking neurons. *Proceedings of IJCNN2006*, 102–109.

de Kamps, M., Baier, V., Drever, J., Dietz, M., Mosenlechner, L. & van der Velde, F. (2008). The state of MIIND. *Neural Networks*, **21**, 1164–1181.

DeLong, M. & Wichmann, T. (2010). Changing views of basal ganglia circuits and circuit disorders. *Clinical EEG and Neuroscience*, **41**, 61–67.

Duffin, E. (2011). Decision making in uncertain situations. MSc Thesis. University of Leeds.

Duffin, E., Bland, A.R., Schaefer, A. & de Kamps, M. (2014). Differential effects of reward and punishment in decision making under uncertainty: a computational study. *Frontiers in Neuroscience*, **8**.

Foerde, K. & Shohamy, D. (2011). The role of the basal ganglia in learning and memory: Insight from Parkinson's disease. *Neurobiology of Learning and Memory*, **96**, 624 – 636.

Frank, M. (2005). Dynamic dopamine modulation in the basal ganglia: A neurocomputational account of cognitive deficits in medicated and non-medicated Parkinsonism. *Journal of Cognitive Neuroscience*, **17**, 51–72.

Frank, M., Moustafa, A., Haughey, H., Curran, T. & Hutchison, K. (2007). Genetic triple dissociation reveals multiple roles for dopamine in reinforcement learning. *Proceedings of the National Academy of Sciences*, **104**, 16311–16316.

Frank, M.J. (2006). Hold your horses: A dynamic computational role for the subthalamic nucleus in decision making. *Neural Networks*, **19**, 1120–1136.

Gaissmaier, W. & Schooler, L. (2008). The smart potential behind probability matching. *Cognition*, **109**, 416–422.

Gerfen, C. (1992). The neostriatal mosaic: Multiple levels of compartmental organization in the basal ganglia. *Annual Review of Neuroscience*, **15**, 285–320.

Gershman, S. (2015). Do learning rates adapt to the distribution of rewards? *Psychonomic Bulletin and Review*, 1–8.

Guitart-Masip, M., Huys, Q., Fuentemilla, L., Dayan, P., Duzel, E. & Dolan, R. (2012). Go and no-go learning in reward and punishment: interactions between affect and effect. *Neuroimage*, **62**, 154–166.

Gurney, K., Prescott, T.J. & Redgrave, P. (2001). A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biological Cybernetics*, **84**, 401–410.

Gurney, K.N., Humphries, M.D. & Redgrave, P. (2015). A new framework for cortico-striatal plasticity: Behavioural theory meets in vitro data at the reinforcement-action interface. *PLoS Biol*, **13**, e1002034.

Guthrie, M., Leblois, A., Garenne, A. & Boraud, T. (2013). Interaction between cognitive and motor cortico-basal ganglia loops during decision making: A computational study. *Journal of Neurophysiology*.

Hampton, A., Bossaerts, P. & O'Doherty, J. (2006). The role of the ventromedial prefrontal cortex in abstract state-based inference during decision making in humans. *Journal of Neuroscience*, **26**, 8360–8367.

Helie, S., Chakravarthy, S. & Moustafa, A.A. (2013). Exploring the cognitive and motor functions of the basal ganglia: An integrative review of computational cognitive neuroscience models. *Frontiers in Computational Neuroscience*, **7**.

Humphries, M.D., Stewart, R.D. & Gurney, K.N. (2006). A physiologically plausible model of action selection and oscillatory activity in the basal ganglia. *The Journal of Neuroscience*, **26**, 12921–12942.

Humphries, M.D., Khamassi, M. & Gurney, K. (2012). Dopaminergic control of the exploration-exploitation trade-off via the basal ganglia. *Frontiers in Neuroscience*, **6**.

Ito, M. & Doya, K. (2009). Validation of decision-making models and analysis of decision variables in the rat basal ganglia. *The Journal of Neuroscience*, **29**, 9861–9874.

Jocham, G., Neumann, J., Klein, T.A., Danielmeier, C. & Ullsperger, M. (2009). Adaptive coding of action values in the human rostral cingulate zone. *Journal of Neuroscience*, **29**, 7489–7496.

Joseph, D., Gangadhar, G. & Chakravarthy, V.S. (2010). ACE (Actor-Critic-Explorer) paradigm for reinforcement learning in basal ganglia: Highlighting the role of subthalamic and pallidal nuclei. *Neurocomputing*, **74**, 205–218.

Kahneman, D. & Tversky, A. (1984). Choices, values, and frames. *American Psychologist*, **39**, 341–350.

Kalva, S.K., Rengaswamy, M., Chakravarthy, V. & Gupte, N. (2012). On the neural substrates for exploratory dynamics in basal ganglia: A model. *Neural Networks*, **32**, 65–73.

Knight, B.W. (1972). Dynamics of encoding in a population of neurons. *The Journal of General Physiology*, **59**, 734–766.

Koehler, D. & James, G. (2009). Probability matching in choice under uncertainty: Intuition versus deliberation. *Cognition*, **113**, 123–127.

Krishnan, R., Ratnadurai, S., Subramanian, D., Chakravarthy, V. & Rengaswamy, M. (2011). Modeling the role of basal ganglia in saccade generation: Is the indirect pathway the explorer? *Neural Networks*, **24**, 801–813.

Krugel, L., Biele, G., Mohr, P., Li, S. & Heekeren, H. (2009). Genetic variation in dopaminergic neuromodulation influences the ability to rapidly and flexibly adapt decisions. *Proceedings of the National Academy of Sciences*, **106**, 17951–17956.

Kumar, A., Cardanobile, S., Rotter, S. & Aertsen, A. (2011). The role of inhibition in generating and controlling Parkinson's disease oscillations in the basal ganglia. *Frontiers in Systems Neuroscience*, **5**.

Lanciego, J.L., Luquin, N. & Obeso, J.A. (2012). Functional neuroanatomy of the basal ganglia. *Cold Spring Harbor Perspectives in Medicine*, **2**.

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitée comme une polarization. *Journal de Physiologie et de Pathologie Générale*, **9**, 620–635.

Lewandowsky, S. & Farrell, S. (2011). *Computational Modeling in Cognition Principles and Practice*. SAGE Publications, Inc.

Mars, R., Shea, N., Kolling, N. & Rushworth, M. (2012). Model-based analyses: Promises, pitfalls, and example applications to the study of cognitive control. *Quarterly Journal of Experimental Psychology*, **65**, 252–267.

MATLAB (2012). *version 8.0.0.783 (R2012b)*. The MathWorks Inc., Natick, Massachusetts.

Merrison-Hort, R.J. & Borisyuk, R. (2013). The emergence of two anti-phase oscillatory neural populations in a computational model of the Parkinsonian globus pallidus. *Frontiers in Computational Neuroscience*, **7**.

Nambu, A. (2008). Seven problems on the basal ganglia. *Current Opinion in Neurobiology*, **18**, 595–604.

Nambu, A., Tokuno, H. & Takada, M. (2002). Functional significance of the cortico-subthalamo-pallidal 'hyperdirect' pathway. *Neuroscience Research*, **43**, 111–117.

Nassar, M., Wilson, R., Heasly, B. & Gold, J. (2010). An approximately Bayesian delta-rule model explains the dynamics of belief updating in a changing environment. *Journal of Neuroscience*, **30**, 12366–12378.

Nelson, A.B. & Kreitzer, A.C. (2014). Reassessing models of basal ganglia function and dysfunction. *Annual Review of Neuroscience*, **37**, 117–135.

Nevado Holgado, A.J., Terry, J.R. & Bogacz, R. (2010). Conditions for the generation of beta oscillations in the subthalamic nucleus-globus pallidus network. *The Journal of Neuroscience*, **30**, 12340–12352.

Nevado Holgado, A.J., Mallet, N., Magill, P.J. & Bogacz, R. (2014). Effective connectivity of the subthalamic nucleusglobus pallidus network during Parkinsonian oscillations. *The Journal of Physiology*, **592**, 1429–1455.

N'Guyen, S., Thurat, C. & Girard, B. (2014). Saccade learning with concurrent cortical and subcortical basal ganglia loops. *Frontiers in Computational Neuroscience*, **8**.

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, **53**, 139–154.

Niv, Y., Edlund, J., Dayan, P. & O'Doherty, J. (2012). Neural prediction errors reveal a risk-sensitive reinforcement learning process in the human brain. *Journal of Neuroscience*, **32**, 551–562.

Nykamp, D. & Tranchina, D. (2000). A population density approach that facilitates large-scale modeling of neural networks: analysis and an application to orientation tuning. *Journal of Computational Neuroscience*, **8**, 19–50.

Obeso, J.A., Marin, C., Rodriguez-Oroz, C., Blesa, J., Benitez-Temio, B., Mena-Segovia, J., Rodrguez, M. & Olanow, C.W. (2008). The basal ganglia in Parkinson's disease: Current concepts and unexplained observations. *Annals of Neurology*, **64**, S30–S46.

Omurtag, A., Knight, B. & Sirovich, L. (2000). On the simulation of large populations of neurons. *Journal of Computational Neuroscience*, **8**, 51–63.

O'Reilly, R. & Frank, M. (2006). Making working memory work: A computational model of learning in the frontal cortex and basal ganglia. *Neural Computation*, **18**, 283–328.

Pavlides, A., Hogan, S. & Bogacz, R. (2012). Improved conditions for the generation of beta oscillations in the subthalamic nucleus-globus pallidus network. *BMC Neuroscience*, **13**.

Payzan-LeNestour, E. & Bossaerts, P. (2011). Risk, unexpected uncertainty, and estimation uncertainty: Bayesian learning in unstable settings. *PLoS Computational Biology*, **7**.

Pessiglione, M., Seymour, B., Flandin, G., Dolan, R.J. & Frith, C.D. (2006). Dopamine-dependent prediction errors underpin reward-seeking behaviour in humans. *Nature*, **442**, 1042–1045.

Pleskac, T.J. & Busemeyer, J.R. (2010). Two-stage dynamic signal detection: A theory of choice, decision time, and confidence. *Psychological Review*, **117**, 864–901.

Redgrave, P., Rodriguez, M., Smith, Y., Rodriguez-Oroz, M., Lehericy, S., Bergman, H., Agid, Y., DeLong, M. & Obeso, J. (2010). Goal-directed and habitual control in the basal ganglia: implications for Parkinson's disease. *Nature Reviews Neuroscience*, **11**, 760–772.

Redgrave, P., Vautrelle, N. & Reynolds, J. (2011). Functional properties of the basal ganglia's re-entrant loop architecture: selection and reinforcement. *Neuroscience*, **198**, 138–151.

Schroll, H. & Hamker, F.H. (2013). Computational models of basal-ganglia pathway functions: Focus on functional neuroanatomy. *Frontiers in Systems Neuroscience*, **7**.

Schroll, H., Vitay, J. & Hamker, F.H. (2012). Working memory and response selection: A computational account of interactions among cortico-basalganglio-thalamic loops. *Neural Networks*, **26**, 59–74.

Schultz, W. (1992). Activity of dopamine neurons in the behaving primate. *Seminars in Neuroscience*, **4**, 129–138.

Schultz, W. (1998). Predictive reward signal of dopamine neurons. *Journal of Neurophysiology*, **80**, 1–27.

Shanks, D., Tunney, R. & McCarthy, J. (2002). A re-examination of probability matching and rational choice. *Journal of Behavioral Decision Making*, **15**, 233–250.

Siegel, S. & Goldstein, D. (1959). Decision-making behavior in a two-choice uncertain outcome situation. *Journal of Experimental Psychology*, **57**, 37–42.

Smith, Y., Bevan, M., Shink, E. & Bolam, J. (1998). Microcircuitry of the direct and indirect pathways of the basal ganglia. *Neuroscience*, **86**.

Stein, R.B. (1965). A theoretical analysis of neuronal variability. *Biophysical Journal*, **5**, 173–194.

Stocco, A. (2012). Acetylcholine-based entropy in response selection: A model of how striatal interneurons modulate exploration, exploitation, and response variability in decision making. *Frontiers in Neuroscience*, **6**.

Sutton, R. & Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIT Press.

Taylor, E.G., Landy, D.H. & Ross, B.H. (2012). The effect of explanation in simple binary decision tasks. *The Quarterly Journal of Experimental Psychology*, **65**, 1361–1375.

Thibeault, C.M. & Srinivasa, N. (2013). Using a hybrid neuron in physiologically inspired models of the basal ganglia. *Frontiers in Computational Neuroscience*, **7**.

Vulkan, N. (2000). An economist's perspective on probability matching. *Journal of Economic Surveys*, **14**, 101–118.

Weinberger, M. & Dostrovsky, J.O. (2011). A basis for the pathological oscillations in basal ganglia: the crucial role of dopamine. *Neuroreport*, **22**, 151–156.

Wichmann, T. & DeLong, M. (2009). The basal ganglia. In E.R. Kandel, J.H. Schwartz & T.M. Jessell, eds., *Principles of Neural Science*, McGraw-Hill Medical, New York, 5th edn.

Wilson, H.R. & Cowan, J.D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, **12**, 1–24.

Wilson, R. & Niv, Y. (2012). Inferring relevance in a changing world. *Frontiers in Human Neuroscience*, **5**.

Yechiam, E. & Hochman, G. (2013a). Loss-aversion or loss-attention: The impact of losses on cognitive performance. *Cognitive Psychology*, **66**, 212–231.

Yechiam, E. & Hochman, G. (2013b). Losses as modulators of attention: Review and analysis of the unique effects of losses over gains. *Psychological Bulletin*, **139**, 497–518.

Yu, A. & Dayan, P. (2005). Uncertainty, neuromodulation, and attention. *Neuron*, **46**, 681–692.

Yu, A.J. & Huang, H. (2014). Maximizing masquerading as matching in human visual search choice behavior. *Decision*, **1**, 275–287.