Negative Correlation in Neural Systems

Simon Durrant

Submitted for the degree of DPhil

University of Sussex

September 2008

Declaration

I hereby declare that this thesis has not been and will not be, submitted in whole or in part to another University for the award of any other degree.

Signature:

Acknowledgements

Firstly, I would like to thank my supervisor, Professor Jianfeng Feng, for the support and inspiration that made this work possible. The central theme of negative correlation first appeared in a discussion with him, and he helped point me in the right direction on countless occasions. He was also stoically understanding of the decidedly variable pace of progress at times.

Many others were also helpful in the formation of ideas and understanding and in providing useful feedback on existing work at various times, including Keith Kendrick, Graham Tattersall, Andy Philippides, Nigel Stocks and of course my colleagues in the Bioinformatics Lab at Sussex.

I am grateful to the Department of Informatics (or Cognitive and Computer Science as it was when I joined) for a Graduate Teaching Assistantship which financially supported me throughout this work, as well as providing me with highly valuable teaching experience.

Finally, many thanks to my friends and family who made life enjoyable during what could have otherwise been a difficult and lonely experience. I am grateful in particular to Eleanor Wye for many interesting and distracting conversations throughout, and my partner Alicja Knast for moral support and a shining example of how to actually finish writing a thesis. The last word of thanks must go to my good friend Paul Horton, who provided superb company and an invaluable sounding board throughout the thesis. While talking and drinking with Paul, things literally got clearer with every pint. So Paul: Cheers!

The large brain, like large government, may not be able to do simple things in a simple way.

- Donald Hebb

University of Sussex

Simon Durrant

Submitted for the Degree of DPhil

Negative Correlation in Neural Systems

Summary

In our attempt to understand neural systems, it is useful to identify statistical principles that may be beneficial in neural information processing, outline how these principles may work in theory, and demonstrate the benefits through computational modelling and simulation. Negative correlation is one such principle, and is the subject of this work.

The main body of the work falls into three parts. The first part demonstrates the space filling and accelerated central limit convergence benefits of negative correlation, both generally and in the specific neural context of V1 receptive fields. I outline two new algorithms combining traditional ICA with a correlation objective function. Correlated component analysis seeks components with a given correlation matrix, while correlated basis analysis seeks basis functions with a given correlation matrix. The benefits of recovering components and basis functions with negative correlations are shown.

The second part looks at the functional role of negative correlation for integrate-and-fire neurons in the context of suprathreshold stochastic resonance, for neurons receiving Poisson inputs modelled by a diffusion approximation. I show how the SSR effect can be seen in networks of spiking neurons, and further show how correlation can be used to control the noise level, and that optimal information transmission occurs for negatively correlated inputs when parameters take biophysically plausible values.

The final part examines the question of how negative correlation may be implemented in the context of small networks of spiking neurons. Networks of integrate-and-fire neurons with and without lateral inhibitory connections are tested, and the networks with the inhibitory connections are found to perform better and show negatively correlated firing patterns. This result is extended to more biophysically detailed neuron and synapse models, highlighting the robust nature of the mechanism. Finally, the mechanism is explained as a threshold-unit approximation to non-threshold maximum likelihood signal/noise decomposition.

Contents

List of Figures

Chapter 1

Introduction

To fully understand human thought and behaviour, the formal goal of psychologists and the informal goal of countless others, we are faced with the unenviable task of trying to understand how the human brain works. In recent years, movement towards this goal in the form of computational neuroscience has led to a new understanding of brain function. More than that, though, it has led to a better understanding of the type of knowledge that is required, and the type of approach that is needed. The use of an information processing perspective, which essentially means to try to understand particular neural structures in terms of how they contribute to information processing, has been an important development. In this thesis, we adopt this approach, and examine neural systems from the perspective of one particular principle: negative correlation.

1.1 Overview

The information processing power and flexibility of the human brain cannot be overstated; the sheer complexity of the system is daunting. Understanding the functioning of the human brain in its many forms has been the goal of neuroscientists for over a century, and great advances have been made in this time, in both the core knowledge about the system itself, and in techniques developed to accelerate the advancement of this knowledge. One of those techniques, made possible by the rapid development of affordable high-powered computing, is electronic simulation of biological systems at many different levels.

At one extreme are highly detailed biophysical models of particular aspects of the brain, such as the axon cable of a single neuron (Rall, 1977), or aspects of neural function such as sleep consolidation (Walker, Stickgold, Alsop, Gaab, & Schlaug, 2005; Rasch, Büchel, Gais, & Born, 2007). At the other extreme are models that are much less interested in accurately simulating biophysical systems, and instead focus on either using the general approach of a distributed network of small processing units (artificial neural networks) to solve abstract problems that may or may not be related to brain function (Rumelhard & McClelland, 1986; McClelland & Rumelhard, 1986; Haykin, 1999). Alternatively, abstract mathematical techniques may be used to understand real details, as in the case of ICA revealing the principle behind receptive fields in the primary visual cortex (Olshausen & Field, 1996; Hyvärinen, Karhunen, & Oja, 2001; Hoyer, 2002).

The use of abstract mathematical techniques highlights something of central importance: the extent of fine detail that needs to be understood, which may be loosely described as the extent to which neural systems are chaotic, or more colloquially as the extent to which there are bottlenecks between levels of detail in neural systems such that we can drop off a level without losing the essence of the behaviour, is still a matter for debate. Nevertheless, it is clear that to understand the human brain is as much about understanding the principles upon which it works as it about understanding the fine details of the wetware that implements those principles. If our goal is to understand how the brain performs so well in everyday problem-solving, we need both of these components, linked together. We need to understand what principles are likely to be of use to the brain, and then understand how these principles may operate given the known facts of the biophysical substrate.

This thesis is about one such principle: negative correlation. Negative correlation is a relatively simple high level principle, but also quite specific and measurable, and easily understood. As we will see in section 1.2 it can also be clearly beneficial in several rather abstract ways. The question we seek to address in this thesis is this: Is negative correlation used in the brain to assist neural information processing, and if so, how does it arise?

To answer this question, we look at the brain in three distinct ways, operating at different levels. The reason for this is to demonstrate just how pervasive the principle of negative correlation is in the brain. It does not just operate at the level of the individual neuron, or at a population of neurons, although its presence here is of central importance, but also in quite abstract and high level ways. It is our belief that negative correlation is one of the guiding principles in neural systems at a variety of levels.

1.2 Negative Correlation

In this section (and subsections) we introduce the central concept of negative correlation, giving first a straightforward definition and example. We then outline the central limit theorem in a slightly more formal way, and give a short discussion on convergence rates, which is key to understanding one of the crucial benefits of negative correlation. Finally, we give a simple demonstration of two of the benefits of negative correlation: centre-surround space filling, and accelerated central limit convergence. This material is introduced at this very early stage to highlight its central importance in all that follows.

Correlation simply describes the linear relationship between two variables, giving a limited indication as to how much a change in one variable corresponds to a similar change in another variable. The value is measured by the correlation coefficient, given for random variables i and j as follows:-

$$\begin{aligned} R_{i,j} &= \frac{c_{i,j}}{\sqrt{c_{i,i}c_{j,j}}} \\ c_{i,j} &= \mathrm{E}\{(x_i - \mu_i)(x_j - \mu_j)\} \end{aligned} \tag{1.1}$$

For a set of more than two variables, a correlation matrix C is defined such that the i, jth element is given by $R_{i,j}$ as defined by equation 1.1, which is called the Pearson product-moment correlation. This is the most common definition, although other definitions also exist (popularly including $C = \mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$ for a set of random variables $\mathbf{x}$).

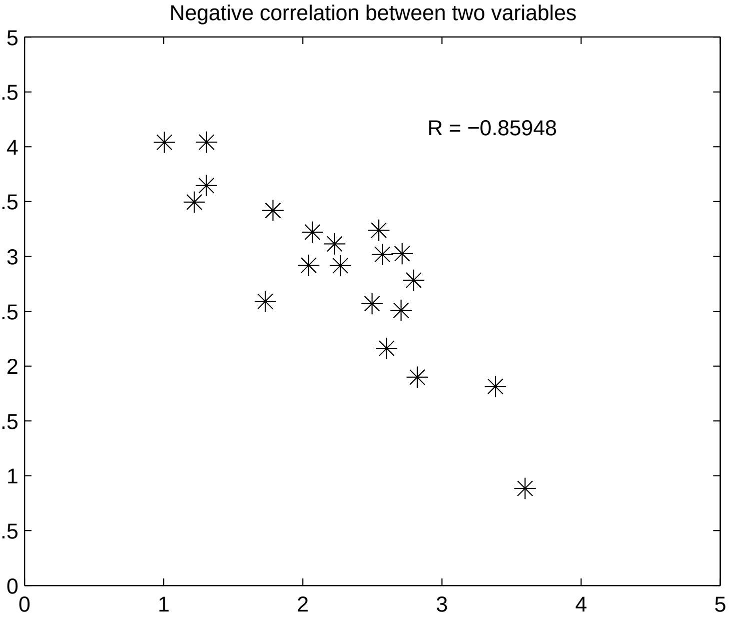


Figure 1.1: Negative correlation between two variables.

Informally, negative correlation simply refers to a situation where one variable tends to decrease when another variable tends to increase, and vice versa. Figure 1.1 shows a simple example for two random variables, represented here by a limited sample of data points. Negatively correlated variables have two interesting and related properties: centre-surround space filling and enhanced noise reduction. We will briefly examine each of these here.

1.2.1 Space Filling

Negative correlation creates opposites: as one variable moves above its mean value, another variable negatively correlated with the first tends to move below. As a result of this, they tend to be on opposite sides of their mean values at any one time. If they share the same mean, this will result in centre-surround space-filling. This can be displayed by treating records (element indices within variables) as dimensions, and plotting each N-element discrete sampled variable as a point in N-dimensional space. If, for example, we have eight standardised variables, each sampled twice providing two records, we can plot each variable as a point on two-dimensional axes using the records as the dimensions (and therefore the sampled values as the coordinates); figure 1.2 gives an example of this. For negatively correlated variables, the

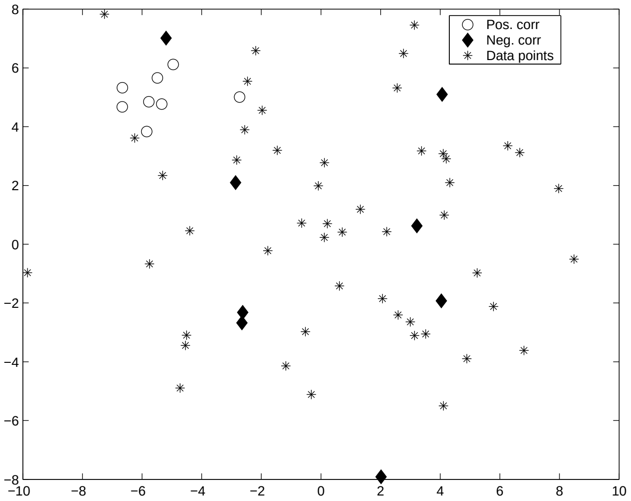


Figure 1.2: Efficacy of negatively correlated basis functions. The negatively correlated basis functions (black diamonds) are more widely distributed than the positively correlated basis functions (white circles), and offer a more useful basis for representing the data points (crosses). The non-negative least-squares error for representing the data is 0 for the negatively correlated bases, but 12.294 for the positively correlated bases. In this example, the positively correlated basis functions have a correlation with each other of 0.9 , whilst the negatively correlated basis functions have the lowest possible correlation for eight basis functions, -0.14286.

values will tend to be on different sides of the mean, which will force the variables to occupy more of the space. Positively-correlated variables, by contrast, will tend to have similar values on the same records, and hence will tend to cluster together.

This is true not just of random variables, but of any ordered set of numbers, including vectors and correspondingly basis functions, and here the benefit of the centre-surround space-filling becomes apparent. Basis functions can be used to reduce the dimensionality of data by representing it as a set of coefficients of existing basis functions. In situations where the basis functions must be specified before the data is known, or must be fixed for a number of different data sets, existing techniques such as PCA, ICA or even Cholesky decomposition (which has a different type of centre-surround space-filling not based on explicit correlation measures) cannot be used. In that case, the question becomes: what basis set is most likely to offer an efficient representation of the data (as measured by reconstruction error, i.e. the difference between a reconstructed data set and the original before compression). Space filling tells us that negatively-correlated basis functions span a data space better than positively-correlated ones, with less redundancy in a potential representation. If the

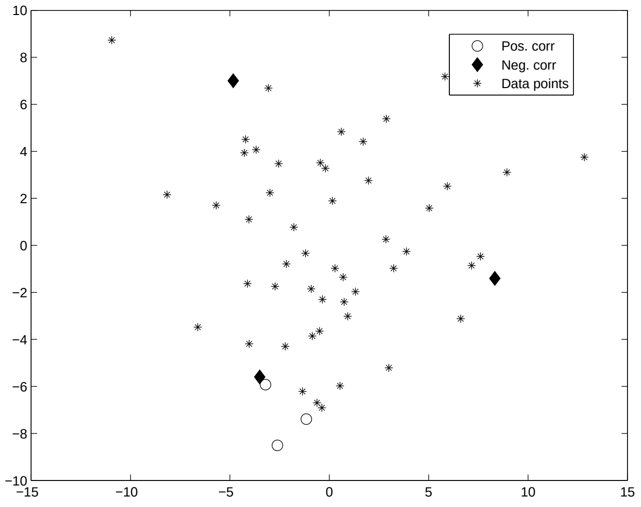


Figure 1.3: Benefit of negatively correlated basis functions in compressing data. The negatively correlated basis functions (black diamonds) are more widely distributed than the positively correlated basis functions (white circles), and offer a more useful basis for representing the data points (crosses). The non-negative least-squares error for representing the data is 10.4 for the negatively correlated bases, but 14.8 for the positively correlated bases. In this example, the positively correlated basis functions have a correlation with each other of 0.98, whilst the negatively correlated basis functions have almost the lowest possible correlation for three basis functions, -0.49. Note that only the first two dimensions are shown here for clarity; the data is actually five-dimensional.

coefficients of the basis set are restricted to positive real numbers, as is the case for many natural quantities, then this space-filling becomes important. Figure 1.3 shows an example where original 5-D data has been compressed using just three basis functions (only first two dimensions in original data space are shown for clarity). The positive basis functions (with an average correlation of 0.98 between them) cluster together because they each have similar values on a given dimension in the original data space; the negative basis functions (with an average correlation of -0.49 between them) are more widely spread because they tend to have opposite values on a given dimension. The reconstruction SSE when restricted to non-negative coefficients is much lower for the negative basis functions (10.4) than the positive ones (14.8), averaged over ten sets of randomly generated basis functions and data (and lower in every run).

1.2.2 Enhanced Noise Reduction

The enhanced noise reduction effect, shown in figure 1.4, is that negatively correlated (Gaussian) noise will tend to reduce to zero more rapidly than independent and

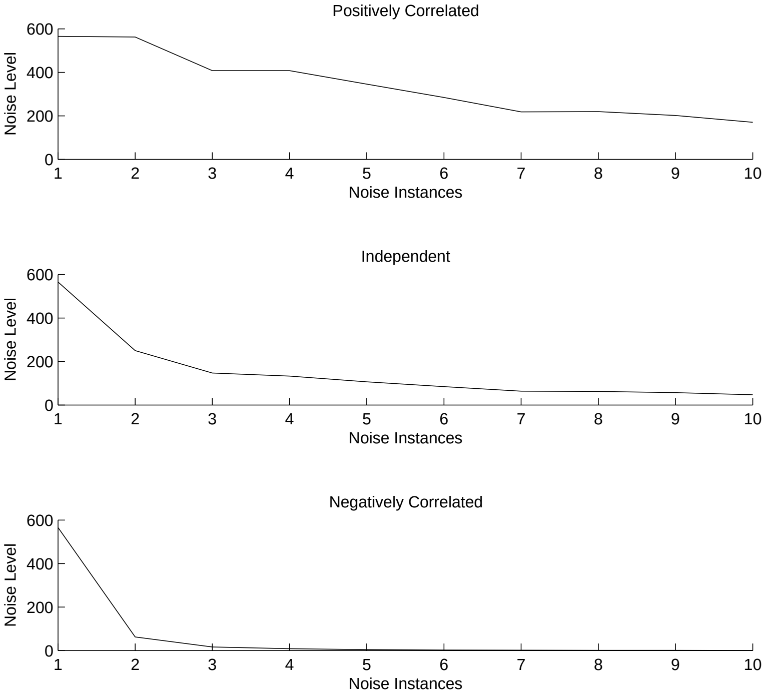


Figure 1.4: Central limit shrinkage of negatively correlated noise. As the number of noise instances (samples) increases, negatively correlated noise shrinks to zero quickly, whereas independent and positively correlated noise require more instances for their values to decrease, with positively correlated noise potentially having a non-zero asymptote. This shows how negative correlation can eliminate noise both more quickly, and more completely. The positively correlated noise here has a correlation of 0.1 between each of the ten instances of noise, whilst the negatively correlated noise has the opposite value of -0.1.

positively correlated Gaussian noise as the number of instances of this noise increase. The utility of this result is shown in figure 1.5, where Gaussian noise is added to a number of replications of the same image. Where the noise is positively correlated the image is almost completely obscured. Independent noise also results in an image in which the detail has been largely lost to the noise. Only when the noise (of the same strength as the previous two cases) is negatively correlated, does the image emerge, as the noise effectively cancels itself out. For a better understanding and slightly more formal demonstration of the important enhanced noise reduction effect in negatively correlated variables, we need to look at the central limit theorem for a set of random variables, and how convergence rates are affected by the correlation of the variables.

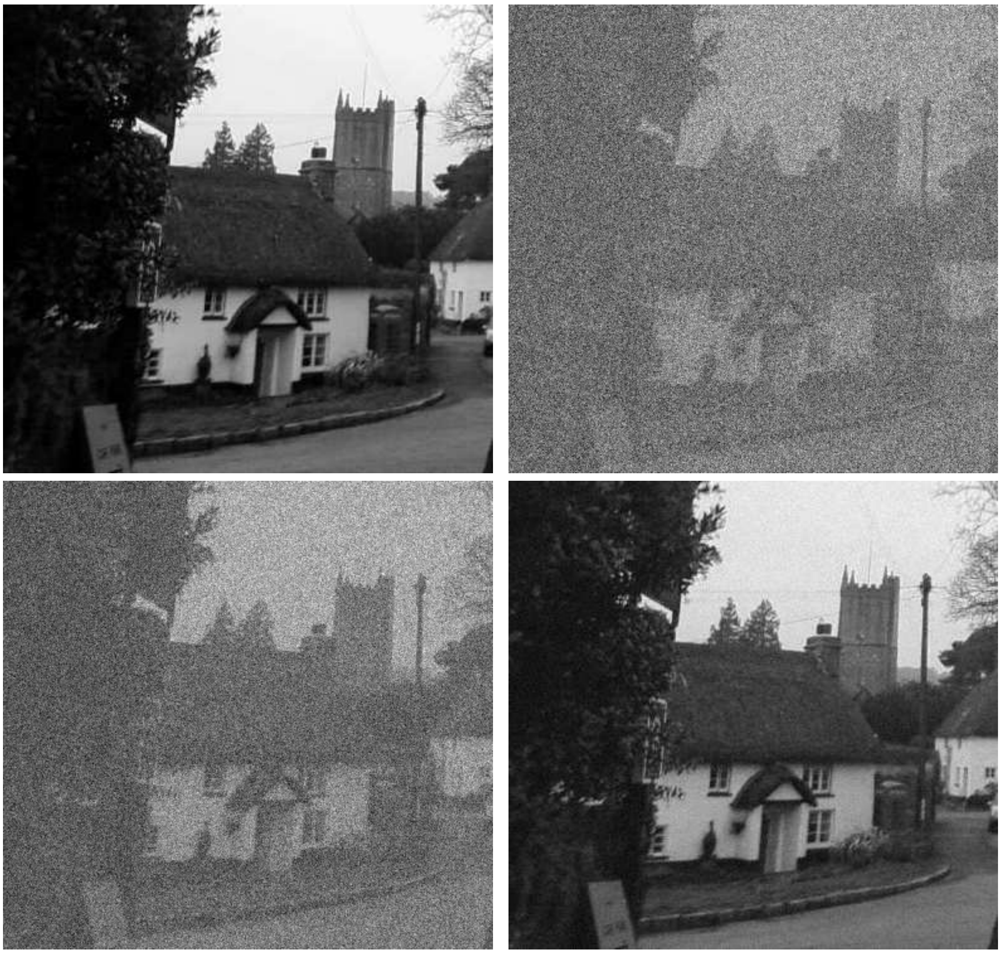


Figure 1.5: Benefit of negatively correlated noise: *Top Left:* Original Image *Top Right:* Positively Correlated Noise. *Bottom Left:* Uncorrelated Noise. *Bottom Right:* Negatively Correlated Noise. With ten separate samples of noise added to the original image, the differing effects of the correlation of the noise can clearly be seen here. In particular, negatively correlated noise largely disappears leaving original image clearly visible. Here, the positively correlated noise again has a correlation of 0.1 between each of the ten instances of noise, whilst the negatively correlated noise has the opposite value of -0.1.

1.2.3 Central Limit Theorem

The central limit theorem describes how the sum of a set of variables approaches a Gaussian variable. For the purposes of this demonstration, in which we aim to produce an intuitive understanding of the central limit theorem for negatively correlated variables, we will use identically-distributed (i.d.) variables, but not exclusively independent identically-distributed (i.i.d.) variables, since we want to be able to handle the more general case where variables are correlated, of which uncorrelated (and therefore possibly, although not necessarily, independent) is a special case (subset).

We start by defining a variable, ξ, which can have any distribution, although here we assume a Gaussian distribution in order that it can be uniquely specified by its first two moments. We define it as having a mean $\mu_\xi = \mathrm{E}\{\xi\} = \int_{-\infty}^{\infty} \xi p(\xi)$, a standard deviation $\sigma_\xi = \mathrm{E}\{\sqrt{(\xi-\mu)^2}\}$ and a variance $\sigma_\xi^2 = \mathrm{E}\{(\xi-\mu)^2\}$.

Let us first suppose that we have N independent variables like this. The sum of these variables is:-

$$R_N = \sum_{n=1}^{N} \xi_n \tag{1.2}$$

The mean of the summed variable R_N is $\mu_R = N\mu_\xi$; in other words, the sum of the means of each variable, which as they are all the same, is the mean of one of the variables multiplied by the number of variables. The standard deviation of the summed variable R_N is $\sigma_R = \sqrt{N}\sigma_\xi$; in other words, the square root of the number of variables multiplied by the standard deviation of the individual variable. The variance of the summed variable R_N is $\sigma_R^2 = N\sigma_\xi^2$; in other words, the sum of the variances of each variable. With sufficient N, R_N also takes on a Gaussian form, the importance of which we will see shortly.

Now let us consider the case for averaging the N independent variables instead of just summing them:-

$$\begin{aligned} S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} \\ \mu_S &= \frac{N\mu_\xi}{N} = \mu_\xi \\ \sigma_S &= \frac{\sqrt{N}\sigma_\xi}{N} = \frac{\sqrt{N}\sigma_\xi}{\sqrt{N}\sqrt{N}} = \frac{\sigma_\xi}{\sqrt{N}} \\ \sigma_S^2 &= \frac{N\sigma_\xi^2}{N^2} = \frac{\sigma_\xi^2}{N} \end{aligned} \tag{1.3}$$

Here we see that as N increases, the standard deviation (and so the variance) correspondingly decrease; this is convergence to the central limit. The mean is the same as the mean of an individual variable.

Given the Gaussian form of the averaged variable, and given that we know the first and second (i.e. all) moments for this, we can write this variable in terms of a

Gaussian variable $Norm(\mu, \sigma^2)$ with mean μ and variance σ^2 (note that the $Norm$ here stands for Normal; N is the standard way of referring to Gaussian variables but we did not use this to avoid confusion with the N in the equations on its own, with refers to the number of variables).

$$\begin{aligned}
S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = Norm(\mu_\xi, \frac{\sigma_\xi^2}{N}) = \mu_\xi + Norm(0, \frac{\sigma_\xi^2}{N}) = \mu_\xi + \phi \\
\phi &= Norm(0, \frac{\sigma_\xi^2}{N}) \\
\mu_\phi &= 0 \\
\sigma_\phi &= \frac{\sigma_\xi}{\sqrt{N}} \\
\sigma_\phi^2 &= \frac{\sigma_\xi^2}{N}
\end{aligned} \tag{1.4}$$

Here, ϕ is a random variable which takes a Gaussian form, has zero mean and a variance given by the central limit theorem for i.i.d. variables.

Now consider the case where we have N identically-distributed variables that may be correlated (i.e. not independent). The first two moments (mean and variance) of the average of these variables are again given by the central limit theorem (or the law of large numbers in this more general case), and we can write it in a similar form as the previous example, but with two important differences:-

$$\begin{aligned}
S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = \mu_\xi + \phi \\
\mu_\phi &= 0 \\
\sigma_\phi^2 &= \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{cov}(\xi_i, \xi_j)\right) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mathrm{cov}(\xi_i, \xi_j)
\end{aligned} \tag{1.5}$$

The first difference is that, because the variables may be positively correlated, the tendency of the distribution shape to converge on a Gaussian may be lowered, to the point that for a given N, the shape may not be significantly Gaussian; this will be true in the limit of full positive correlation for any N and a non-Gaussian individual variable distribution. As such, we do not specify a parametric form for ϕ here, simply describing it in terms of the first two moments instead. The second difference is that the variance of the summed variable S_N now includes all the covariance terms, rather than just the (self-)variance terms, i.e. the whole covariance matrix rather than just the principal diagonal. It is apparent that the variance is simply the average of the covariance matrix, i.e. the sum of all N^2 elements divided by N^2.

The i.i.d. case can be seen in this context, where the variance of one of the variables (being the same for each one) is the obviously the same as the variance of all the variables summed and divided by the number of the variables, and all the non-self covariance terms (i.e. all elements of the covariance matrix outside of the principal diagonal) are zero (as the variables are uncorrelated), hence the part in brackets of the above variance expression is simply equal to the variance of one of the variables, and the whole thing equal equal to that variance divided by the number of variables.

Note that if ϕ has a Gaussian form, we can rewrite the above as follows:-

$$\begin{aligned} S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = Norm(\mu_\xi, \frac{\sigma_\xi^2}{N}) = \mu_\xi + Norm(0, \frac{\sigma_\xi^2}{N}) = \mu_\xi + \frac{Norm(0, \sigma_\xi^2)}{\sqrt{N}} \\ \mu_\phi &= 0 \\ \sigma_\phi^2 &= \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{cov}(\xi_i, \xi_j)\right) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathrm{cov}(\xi_i, \xi_j) \end{aligned} \tag{1.6}$$

Now consider the case for correlated i.d. variables, but where all the correlations are the same, i.e. variables ξ_1 and ξ_2 have the same correlation as ξ_2 and ξ_3. This is obviously a subset of the previous example, but we are able to simplify the expressions slightly as a result of this condition:-

$$\begin{aligned} S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = \mu_\xi + \phi \\ \mu_\phi &= 0 \\ \sigma_\phi^2 &= \frac{1}{N}\left(\sum_{j=1}^{N}\mathrm{cov}(\xi_1, \xi_j)\right) = \frac{1}{N}\left(\sigma_\xi^2 + \sum_{j=2}^{N}\mathrm{cov}(\xi_1, \xi_j)\right) \end{aligned} \tag{1.7}$$

Noting that a variable ϕ with variance $\frac{\sigma_\phi^2}{N}$ is equivalent to $\frac{\phi}{\sqrt{N}}$ where ϕ has a variance of σ_ϕ^2 (see the Gaussian case above where this is shown explicitly), we can rewrite the above as:-

$$\begin{aligned} S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = \mu_\xi + \frac{\phi}{\sqrt{N}} \\ \mu_\phi &= 0 \\ \sigma_\phi^2 &= \sum_{j=1}^{N}\mathrm{cov}(\xi_1, \xi_j) = \sigma_\xi^2 + \sum_{j=2}^{N}\mathrm{cov}(\xi_1, \xi_j) \end{aligned} \tag{1.8}$$

It is worth noting that as well as ϕ being a variable, which, when divided by $\sqrt{N}$

gives the variance term for the average of N variables, it is also the variance term if not divided by anything, of the sum of N variables.

1.2.4 Convergence Rates

The central limit theorem is concerned amongst other things with the rate of convergence to the central limit. Here, we can look at the rate of this convergence under different correlation conditions. The three possible conditions in general are positively correlated, uncorrelated and negatively correlated. We first remind ourselves of the equations:-

$$\begin{aligned} S_N &= \frac{\sum_{n=1}^{N} \xi_n}{N} = \mu_\xi + \frac{\phi}{\sqrt{N}} \\ \mu_\phi &= 0 \\ \sigma_\phi^2 &= \sum_{j=1}^{N} \text{cov}(\xi_1, \xi_j) = \sigma_\xi^2 + \sum_{j=2}^{N} \text{cov}(\xi_1, \xi_j) \end{aligned} \tag{1.9}$$

We must be clear about what we mean by convergence and the rate of convergence here. We can define the reliability of a variable in terms of its variance, i.e. how far a typical value will be from the central (mean) value. Slightly more formally, this is therefore the expected value of the difference between the variance and the expected value (mean) of the summed variable, $\text{E}\{(S_N - \text{E}\{S_N\})^2\}$. When we talk about this value converging, we are referring to the fact that it can be written as a function of N, the number of variables, and that as the number of variables increases, the central limit theorem tells us that this value will decrease towards zero in the uncorrelated case, at a certain rate. Here, we are interested in what this rate is, and how it compares with the rate (and a lower bound limit, which may not necessarily be zero) for the case where the variables have positive or negative correlation. We start by rewriting the previous expectation in terms of the N and the variance of the summed variable, for our particular case of i.d. variables:-

$$\begin{aligned} \text{E}\{(S_N - E\{S_N\})^2\} &= \text{E}\{(S_N - \mu_\xi)^2\} = \text{E}\{(S_N - E\{\xi_1\})^2\} \\ \text{E}\{(S_N - \mu_\xi)^2\} &= \text{E}\{\frac{\phi^2}{N}\} = \frac{\sigma_\phi^2}{N} \\ \sigma_\phi^2 &= \sigma_\xi^2 + \sum_{j=2}^{N} \text{cov}(\xi_1, \xi_j) \end{aligned} \tag{1.10}$$

The first of these follows because the variables are i.d., and so the expected value of the sum is the same as the expected value of any individual one, hence we can replace $\mathrm{E}\{S_N\}$ with μ_ξ or $\mathrm{E}\{\xi_1\}$; the second equality here is included for completeness and equivalence with our results shown later in chapter 5. The second equation follows directly from the definition of S_N given in the previous set of equations and the definition of variance (which for a zero-mean variable is simply the expected squared value); the third equation is simply σ_ϕ^2 repeated from earlier for convenience.

It is easy to see that for a given value of the numerator (σ_ϕ^2) the convergence will be at a linear rate given by N, and this applies to all situations. We may therefore say that convergence is at a rate of N/h, where h controls the effective rate. As h gets bigger, the rate of convergence is slower, i.e. there will be less convergence for a given N. Conversely, as h gets smaller, the convergence will be faster. We can see from our expression for (σ_ϕ^2) how h changes as a function of the correlation, and can thus characterise the three convergence scenarios as follows:-

$$\mathrm{E}\{(S_N - \mu_\xi)^2\} = \begin{cases} \dfrac{\sigma_\xi^2 + \sum\limits_{j=2}^{N} |\mathrm{cov}(\xi_1, \xi_j)|}{N} & \text{if } \mathrm{cov}(\xi_1, \xi_j) > 0 \\ \dfrac{\sigma_\xi^2}{N} & \text{if } \mathrm{cov}(\xi_1, \xi_j) = 0 \\ \dfrac{\sigma_\xi^2 - \sum\limits_{j=2}^{N} |\mathrm{cov}(\xi_1, \xi_j)|}{N} & \text{if } \mathrm{cov}(\xi_1, \xi_j) < 0 \end{cases} \tag{1.11}$$

$$h \approx \begin{cases} \sum\limits_{j=2}^{N} |\mathrm{cov}(\xi_1, \xi_j)| & \text{if } \mathrm{cov}(\xi_1, \xi_j) > 0 \\ \sigma_\xi^2 & \text{if } \mathrm{cov}(\xi_1, \xi_j) = 0 \\ \sigma_\xi^2 - \sum\limits_{j=2}^{N} |\mathrm{cov}(\xi_1, \xi_j)| & \text{if } \mathrm{cov}(\xi_1, \xi_j) < 0 \end{cases} \tag{1.12}$$

We can describe these results as follows:-

1. With positive correlation, h is potentially large, convergence is slow, and there is a fixed lower bound for any value of N below which we cannot go, which is given by the average covariance. In the case of i.d. variables with an average positive correlation of c, we therefore find as $N \to \infty$, $\mathrm{E}\{(S_N - \mu_\xi)^2\} \to c\sigma_\xi^2$.

2. When the variables are uncorrelated, the accuracy tends to zero with a rate of $\frac{N}{\sigma_\xi^2}$. As N is larger, accuracy gets correspondingly better. There is no finite lower bound here, but improving accuracy is dependent on being able to

increase N, which in a real system such as a neural system is not something that can be readily increased without limits.

3. When the variables are negatively correlated, for any value of N we can converge to an arbitrarily low value provided that the correlation is sufficiently negative (this is achievable only in a limiting sense), because h can take on an infinitesimal value, and so the rate of convergence can grow without bound. We no longer need to increase N to increase accuracy, although increasing N will of course still have this effect as well.

This highlights the clear benefit of negative correlation for a group of random variables in terms of reliability and reduction of noise. We will refer to this subsequently as the accelerated or enhanced central limit convergence of negatively correlated variables. We will see that this important property, although intuitively obvious, has significant consequences for neural systems.

1.3 Thesis Organisation

This thesis is about a principle, not a model. As such, we dispense with the more usual prototype of model description, simulation results and discussion in successive chapters, and instead adopt that as a prototype within each of our three core research chapters, which examine the principle in different ways, and at different levels. It is organised in the following way.

This chapter gives a description of the central topic of this work, provides a motivation for it and gives an outline of how it fits together. It also includes detail of any previous publications containing parts of this work, and any other sources or materials related to it in this way.

Chapter 2 gives the background material, in particular describing the tools and techniques that were used in the subsequent chapters. It gives a very brief description of biological neurons, before going on to show how these can be artificially modelled as integrate and fire neurons and Hodgkin-Huxley neurons, with kinetic synapse models. Following this, neural information processing and information theory as it relates to this is briefly outlined, followed by a short elementary account of stochastic resonance, including suprathreshold stochastic resonance, and finally independent component analysis and sparse coding, in particular with relation to receptive fields in the primary visual cortex (V1).

Chapter 3 describes two new and closely related algorithms for the application of negative correlation to independent component analysis and sparse coding. The purpose of this chapter, which is quite different from the two chapters that follow it, is twofold: it acts as both a practical demonstration of the benefits of negative correlation in the context of an information processing problem, and also to show that even in rather high level and abstract ways, negative correlation can still be beneficial to neural processing in particular. This chapter first describes the background to the algorithms, and in particular the relationship between ICA/sparse coding algorithms and V1 receptive fields, then describes the algorithms themselves, and then provides some simple demonstrations of their application to natural visual images in terms of how efficiently the images are represented, and the correlation of the components or basis functions obtained.

Chapter 4 examines the application of negative correlation in neural systems via the phenomenon of suprathreshold stochastic resonance. It first replicates and describes the basic central result in SSR from the original papers by Stocks (Stocks, 2000a, 2001a), before modifying the approach to use integrate and fire neurons. This it turn developed by the use of Poisson spike train diffusion approximations to show that the optimal noise level for information transmission in a population of identical neurons is achieved when the noise is negatively correlated.

Chapter 5 builds on the results from chapter 4 by examining ways in which negative correlation between neurons can be achieved, and asking how beneficial this can be. A simple visual tracking task is used as one way into answering this question, and both simple and more biophysically realistic neuron and synapse models are tested, in order to see how robust the effect might be.

Chapter 6 gives a final summary of the work, broadly linking together the central conclusions, describing the main contribution, and pointing out any limitations. Directions for future work are also discussed.

1.4 Publications, Previous Work and Sources

Most of the work in this thesis appears here for the first time anywhere in print. However, one part has been previously published; specifically, chapter 5 has been published in a modified form as "Negatively Correlated Firing: The Functional Meaning of Lateral Inhibition Within Cortical Columns" in *Biological Cybernetics* volume 95 no.5, pages 431-453 (Durrant & Feng, 2006).

Section 2.4 of the background chapter 2 stems from the author's previous MSc dissertation (University of Sussex, 2003; "Natural Image Statistics and the Early Visual System: Independent Component Analysis and Sparse Coding Approaches" (Durrant, 2003), and uses one figure from that work, although it must be emphasised that this section itself has been written entirely from scratch for this thesis. The algorithms described in the ICA chapter 2 are also subject to an international patent: WO 2006/125960 A2 (with Professor Keith Kendrick of the Babraham Institute, Cambridge, and Professor Jianfeng Feng of the University of Warwick) - *Signal Processing, Transmission, Data Storage and Representation.*

The experiments and demonstrations in section 2.4 of the background chapter 2, and the ICA chapter 3, have been performed using software written specifically by the author for this purposes. However, this software was inspired by, and initially partly modelled upon, the software packages *fastica* (Hyvärinen & Oja, 1997; Hyvärinen et al., 2001) *sparesnet* (Olshausen & Field, 1996) and *nnscpack* (Hoyer, 2002), which we we acknowledge here as sources. The software for all other experiments, demonstrations and figures throughout the work was written exclusively by the author for the express purpose of this thesis. All source code can be supplied upon request.

Chapter 2

Background: Modelling and Analysing Neural Systems

In this chapter, we introduce the tools and models that will be used in the subsequent chapters, giving the background material and some relevant early results necessary to underpin the later core research, starting here with individual neuron models and their biological inspiration, then examining how these can be combined and evaluated from an information theroetic perspective, then moving up a level to look at the issue of noise in a population of neurons in the context of stochastic resonance and suprathreshold stochastic resonance, and finally moving to a more abstract level to look at how the principles of independent component analysis and sparse coding are employed in a neural context, taking the popular example of receptive fields in the primary visual cortex. In the core research chapters that follow (chapters 3, 4 and 5), we call upon this material in essentially the opposite direction, to emphasise that the relationship between these levels is bidirectional.

Most of this chapter describes standard results and approaches in the field, and as such, while presenting the it in our own way, we nevertheless follow and recommend a number of sources for this material, including the excellent texts of Dayan and Abbott (Dayan & Abbott, 2001) and Gerstner and Kistler (Gerstner & Kistler, 2002) for neuron modelling and neural information processing, the widely cited articles of Destexhe, Mainen and Sejnowski (Destexhe, Mainen, & Sejnowski, 1994, 1998) for synaptic modelling, the text of Hyvärinen, Karhunen and Oja (Hyvärinen et al., 2001) for independent component analysis, the review article of Gammaitoni, Hänggi, Jung and Marchesoni for stochastic resonance (Gammaitoni, Hanggi, Jung, & Marchesoni, 1998) and the seminal articles of Stocks (Stocks, 2000a, 2001a) on suprathreshold stochastic resonance. In addition, we present some original demonstrations and results which help to lead in to the main research work of the subsequent chapters.

2.1 Neurons and Synapses

2.1.1 Biological Neurons

In the human brain, there are principally two types of cells. The more numerous of the two, glial cells, are generally believed to be support structures only, not playing an active role in neural information processing. As such, most interest has focused on neurons and for more than a century a serious research effort has been undertaken in order to describe, characterise and understand these building blocks of the central nervous system. This has ranged from the early drawings of Otto Dieters, through the Golgi-staining work of Ramón y Cajal, to electroscopy, single- and more recently multi-electrode physiological recording techniques. A summary of much of this research can be found in the text by Dowling (Dowling, 1992).

Figure 2.1 from Kandel (Kandel, 2000) shows a number of different types of neurons that are found within the human brain. While differing in many of their details, these neurons all have the same basic functional units. They have a dendrite tree which acts as an input area, a cell body (soma) which integrates these inputs and controls the central behaviour of the neuron and an axon which projects possible outputs. Functionally, neurons operate on the principles of electronic circuits (see section 2.1.2 below), with the membrane potential (Koester & Siegelbaum, 2000) as the perhaps the most important defining feature of the state of a neuron at any given time. However, although some neurons have direct electrical signal propagation to each other, for the most part the connections between neurons are mediated by chemical processes in small gaps between them (synapses), the vast majority of which are between the axons of presynaptic neurons and the dendrites of the postsynaptic neuron.

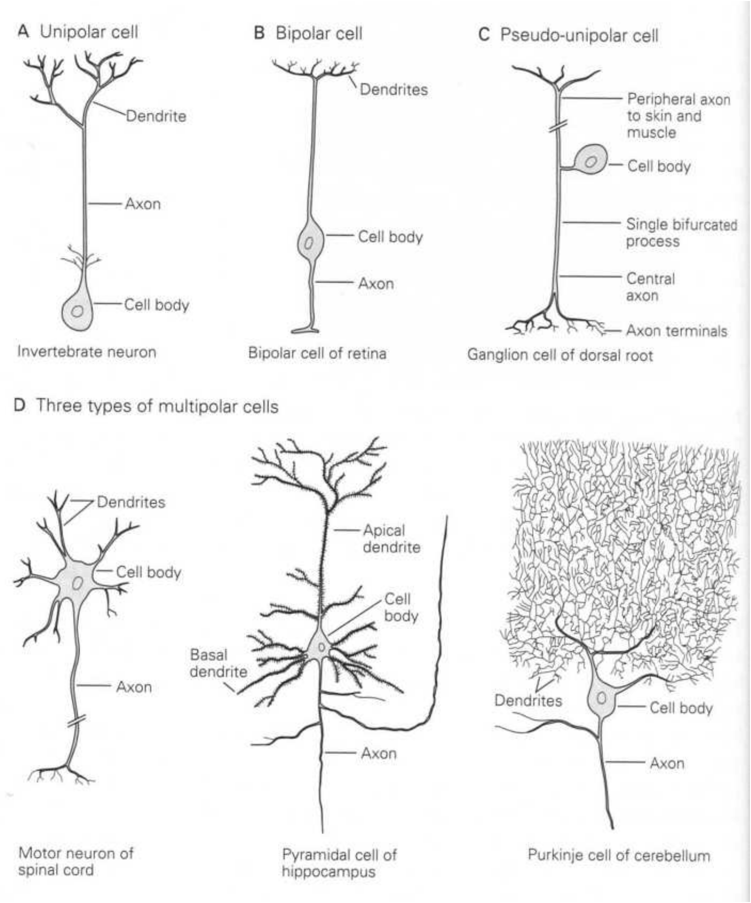


Figure 2.1: A number of different types of neurons found in the human brain. They have certain features in common, including the presence of axons, dendrites and cell bodies. From (Kandel, 2000) ©MIT.

Signal propagation occurs when a current travelling down the axon of a presynaptic neuron reaches the synapse. There, synaptic vesicles move to the edge of the synaptic membrane and release molecules of neurotransmitter into the synaptic cleft (gap). Different neurotransmitter types bind with specific receptor types on the other side of the synaptic cleft (during a short period of time before any remaining molecules are taken back into the presynaptic terminal). The binding of neurotransmitter molecules to postsynaptic receptor sites causes a chemical chain reaction which essentially propagates an electrical current through the dendrite, where it may be combined with the current coming from other synapses on other branches of the dendritic tree. This current makes it way to the cell body, where it has an impact on the membrane potential. A positive postsynaptic current has a depolarising effect on the membrane potential, while a negative postsynaptic current has a hyperpolarising effect. The extent of this impact depends on a number of factors, including the existing membrane potential and the size of the postsynaptic current. The cell body is relatively insulated against the direct effect of external currents, but the postsynaptic current exerts its influence through specific ion channels (Siegelbaum & Koester, 2000) that exist in the cell membrane, which may open or close in response to a current, again depending on the existing state of the membrane potential. If the change in the membrane potential is high enough, some types of ion channels give positive feedback, further rapidly driving up the membrane potential before being finally suppressed. This chain of events is known as an action potential, and is believed to be the fundamental unit of neural information transfer (see section 2.2). This action potential results in a presynaptic current in this neuron, which then travels down the axon to the synaptic terminals, where the process continues on to another neuron.

The above highly simplified account leaves out a myriad of important details, but nonetheless captures the very basic essence of core neuronal behaviour. The core elements - the membrane potential, presynaptic and postsynaptic currents, ion channels, synaptic transmission - have been taken as the building blocks for simulated neuron models in the burgeoning field of computational neuroscience (Dayan & Abbott, 2001). These models, which form the basis for much of the work presented in

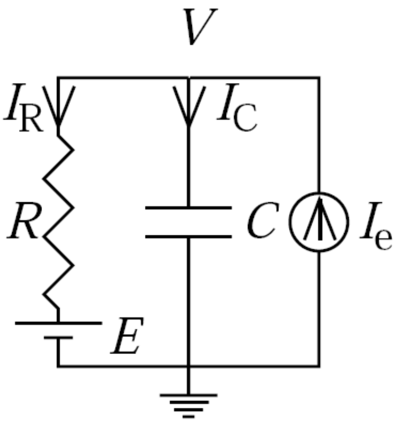


Figure 2.2: Electronic circuit equivalent of an integrate-and-fire neuron. See text for details. From (Dayan & Abbott, 2001) ©MIT.

chapters 4 and 5, are described in the following sections.

2.1.2 Integrate and Fire Neurons

As we have seen, biological neurons are characterised by a voltage (the membrane potential) V, which acts as a state variable, and a number of external input currents, which affect that state variable. The neural membrane acts as both a capacitor (with capacitance C_M) and a resistor (with resistance R_M), and a neuron has a characteristic resting potential sometimes known as the leakage potential, E_L, to which it will revert in the absence of any external driving force, and a surface area A_M, both of which are more or less constant for a given neuron. In the simplest form we can combine all of the external input currents into a single current, I_E, and we therefore have the simple electronic circuit shown in figure 2.2.

Given an excess charge across the cell membrane of Q_M, we have

$$\begin{aligned} Q_M &= C_M V \\ I_C &= \frac{dQ_M}{dt} \\ \Rightarrow I_C &= C_M \frac{dV}{dt} \end{aligned} \tag{2.1}$$

Kirchoff's laws require that the currents in our circuit sum to zero. In the absence of an external input, we can set I_E to zero and hence we find $I_C = -I_R$. We also know from Ohm's law that $I_R = V/R_M$. Temporarily rescaling the voltage to set the resting potential E_L to zero, we can combine these facts to allow us to rewrite the equation of our zero-input circuit as:-

$$C_M \frac{dV}{dt} = -\frac{V}{R_M} \tag{2.2}$$

Adding the resting potential (which is not time dependent in this formulation) back in (which for human cortical neurons which we are exclusively concerned with in this work most commonly takes a value of $E_L = -64.996$ mV) gives us the more general form:-

$$C_M \frac{dV}{dt} = \frac{E - V}{R_M} \tag{2.3}$$

Finally, we can consider the case where there is an external driving current, I_E (which may represent a set of external currents in practice, such as post-synaptic currents on dendrites triggered by action potentials in connected neurons). Here, Kirchoff's laws require that $I_E - I_C - I_R = 0$, which also means $I_C = I_E - I_R$. Hence our circuit equation becomes:-

$$C_M \frac{dV}{dt} = \frac{E_L - V}{R_M} + I_E \tag{2.4}$$

The capacitance C_M and resistance R_M given above are in fact dependent on the membrane surface area A_M, which is not generally desirable. For capacitance, this can be factored out by dividing by the membrane area, leaving an area-independent measure called the specific capacitance $c_M = C_M/A_M$, which takes a typical value of 10 nF/mm^2. Membrane conductance G_M can be similarly divided by the membrane area to give a specific membrane conductance g_M with a typical value of 1 μS/mm^2, the inverse of which is the specific membrane resistance, which is therefore obtained by multiplying by the membrane area $r_M = R_M A_M$ and takes a typical value of 1 MΩmm^2. These two membrane specific properties are often combined into a single parameter called the membrane time constant $c_M r_M = \tau_M$, which essentially controls the timescale on which the neuron operates, and takes a typical value of around 10 ms. Dividing by the membrane surface area (typically around 0.1mm) allows us to rewrite out basic integrate-and-fire (IF) neuron model as:

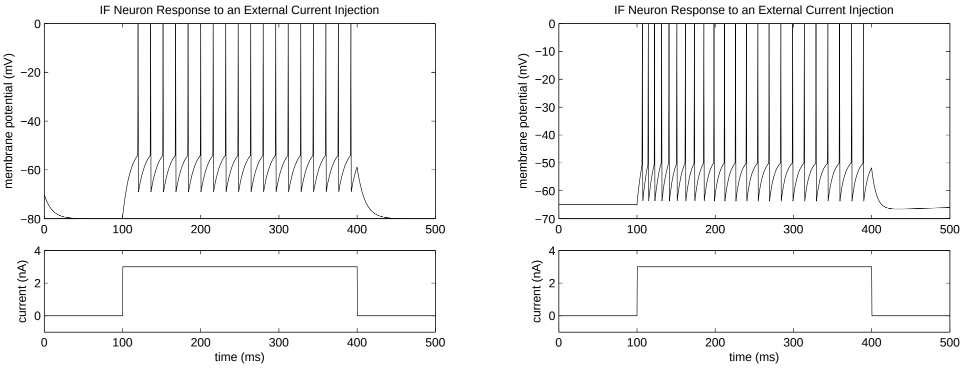


Figure 2.3: Spiking behaviour of an integrate and fire neuron. *Left:* Basic leaky integrator model. *Right:* Leaky integrator with an adaptive conductance term.

$$\begin{aligned} \frac{C_M}{A_M}\frac{dV}{dt} &= \frac{E_L - V}{R_M A_M} + \frac{I_E}{A_M} \\ \Rightarrow c_M\frac{dV}{dt} &= \frac{E_L - V}{r_M} + \frac{I_E}{A_M} \\ \Rightarrow \tau_M\frac{dV}{dt} &= E_L - V + I_E R_M \end{aligned} \tag{2.5}$$

Equation 2.5 is the central equation for the leaky integrator neuron model which characterises integrate-and-fire models. It controls the evolution of the membrane potential, except when this value reaches a predetermined firing threshold V_{thre} (a constant). At that time, the neuron is said to have fired (emitted a spike), and the membrane threshold is instantaneously reset to the resting potential (or in some models an alternative value). A demonstration of its basic behaviour is shown in figure 2.3. Rather than combining the specific membrane capacitance and resistance in a single time constant, it is also possible to view describe this in terms of a (separated) specific membrane capacitance and conductance:-

$$c_M\frac{dV}{dt} = g_M(E_L - V) + \frac{I_E}{A_M} \tag{2.6}$$

This alternative view is useful to highlight the fact that the membrane current can be described in terms of the membrane potential (a variable), reversal potential (constant) and conductance specific to the cell membrane (constant). It suggests the mechanism for including additional inputs that might be useful to model known neural behaviour, which is to include the membrane potential (a variable), a reversal potential to modulate the relevant behaviour and a conductance to scale the effect. One popular example is the addition of a term to model the known adaptive quality of neural spike trains, whereby the firing rate relaxes after an initial peak before

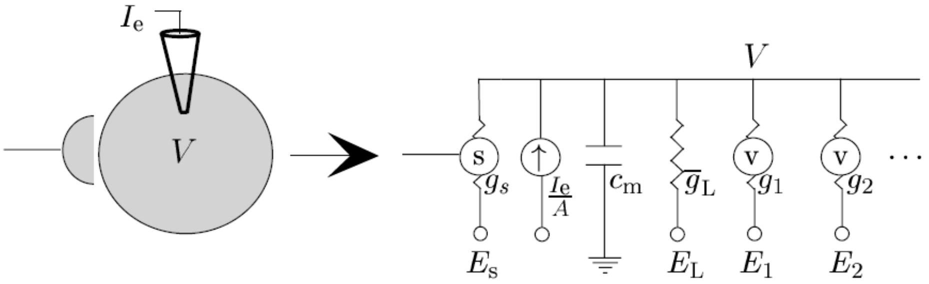


Figure 2.4: Electronic circuit equivalent of a neuron modelled with individual channel conductances, such as the Hodgkin-Huxley model. Here g_s represents the conductance from an external synaptic input, $\frac{I_e}{A}$ gives an external current input, c_m represents the specific membrane capacitance, g_L gives the leakage conductance, and g_1 and g_2 represent other channel conductances, such as from a potassium channel and a sodium channel. From Dayan and Abbott (2001) ©MIT.

settling into a steady lower rate, in response to a constant input. This can be modelled as shown in equation 2.7, where g_A is a conductance dedicated to the adaptive mechanism and E_A is the adaptive reversal potential. The behaviour of this type of model is also shown in figure 2.3.

$$\begin{aligned} c_M \frac{dV}{dt} &= g_M(E_L - V) + g_A(E_A - V) + \frac{I_E}{A_M} \\ \Rightarrow \tau_M \frac{dV}{dt} &= E_L - V + g_A(E_A - V) r_M + I_E R_M \end{aligned} \tag{2.7}$$

The leaky-integrator-plus-conductances approach to modelling neural behaviour derives from the more detailed neural models that include specific channel conductances. Originally characterised by Hodgkin and Huxley (Hodgkin & Huxley, 1952), these more detailed models are briefly described in the next section.

2.1.3 Hodgkin-Huxley Neurons

The integrate and fire model outlined in the previous section describes neural firing dynamics in terms of the evolution of the membrane potential with a single conductance corresponding to a leakage channel, along with an instant-reset threshold. This serves as a good approximation of neural behaviour insofar as the generation of neural spike trains is concerned; it does not, however, capture the more detailed behaviour of the membrane potential, and in particular it does not characterise the membrane potential changes during and after the neuron fires an action potential. Hodgkin and Huxley (Hodgkin & Huxley, 1952), in a now classic paper, produced a set of equations that did just this. Their work concentrated on the axon of the giant squid, but it has since been applied in all sorts of scenarios, including a rather generic description of human cortical pyramidal neurons (although these are in fact somewhat different in their detailed dynamics).

From section 2.1.1, we may recall that, in addition to a generic leakage conductance, neurons have a number of ion channels which are involved in the depolarisation of a cell. These channels can be grouped according to the type of ion, and as each channel is either open or closed at a given time, the proportion of open channels at any given time can be characterised by the opening/closing probability for that type of channel. Hodgkin and Huxley modelled the two principal types of ion channels: potassium and sodium. Potassium channels are modelled with a single voltage-dependent gate as shown in equation 2.8; while the membrane potential is high, the gate will remain open. Sodium channels (equation 2.9) are slightly more complex, having an activation gate that, like the potassium channel gate, has an increasing probability of opening as the membrane potential increases, but also an inactivation gate which blocks the channel when the membrane potential becomes too high. Each channel type has an associated conductance value ($g_K = 0.36$ mS/mm^2 and $g_{Na} = 1.2$ mS/mm^2) and reversal potential ($E_K = -77$ mV and $E_{Na} = 50$ mV) following the model described at the end of the previous section), so altogether there is a current for each channel type (also shown in equations 2.8 and 2.9) that contributes to the overall membrane current (I_M). This current is combined with the basic capacitor-resistor circuit for a neuron (as described earlier, and shown in figure 2.4), to give the overall model for a Hodgkin-Huxley neuron, shown in equation 2.10.

$$\begin{aligned}
i_K &= g_K p_K (V - E_K) \\
p_K &= n^4 \\
\frac{dn}{dt} &= \alpha_n (1-n) - \beta_n n \\
\alpha_n &= \frac{(0.01(V+55))}{(1-\exp(-0.1(V+55)))} \\
\beta_n &= 0.125\exp(-0.0125(V+65)
\end{aligned} \tag{2.8}$$

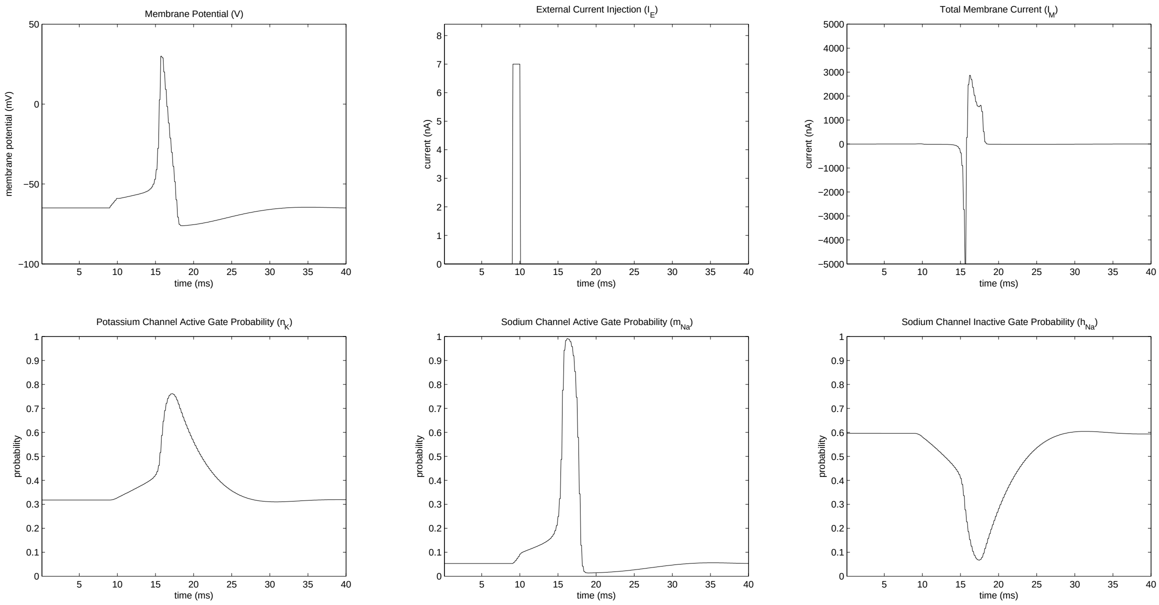


Figure 2.5: Behaviour of a Hodgkin-Huxley neuron during an action potential. *Top Left:* Membrane potential. *Top Middle:* External current injection. *Top Right:* Total membrane current. *Bottom Left:* Potassium channel active gate opening probability. *Bottom Middle:* Sodium channel active gate opening probability. *Bottom Right:* Sodium channel inactive gate opening probability.

$$\begin{aligned}
i_{Na} &= g_{Na} p_{Na} (V - E_{Na}) \\
p_{Na} &= m^3 h \\
\frac{dm}{dt} &= \alpha_m (1 - m) - \beta_m m \\
\frac{dh}{dt} &= \alpha_h (1 - h) - \beta_h h \\
\alpha_m &= \frac{(0.1(V + 40))}{(1 - \exp(-0.1(V + 40)))} \\
\alpha_h &= 0.07 \exp(-0.05(V + 65) \\
\beta_m &= 4 \exp(-0.0556(V + 65)) \\
\beta_h &= \frac{1}{(1 + \exp(-0.1(V + 35)))}
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
c_M \frac{dV}{dt} &= I_M + \frac{I_E}{A_M} \\
I_M &= g_L (E_L - V) + g_K p_K (E_K - V) + g_{Na} p_{Na} (E_{Na} - V)
\end{aligned} \tag{2.10}$$

The basic dynamics of a Hodgkin-Huxley (HH) neuron are shown in figure 2.5. This example shows it receiving a fixed current injection from an external electrode. In reality, the *in vivo* behaviour of neurons is driven by synaptic input from other neurons rather than steady current injections. As well as modelling the basic internal dynamics of neurons through the IF and HH models, we therefore also need to model synaptic inputs.

2.1.4 Synapses

Synapses are labelled by the neurotransmitters used to generate them, and can be broadly divided into two types: fast (ionotropic), which have an almost instantaneous rise followed by a slower decay, and slow (metabotropic), which rise more slowly. There are many types of synapses found in the human brain and the modelling of synapses is a major topic in computational neuroscience but our discussion here will be limited to four of the most commonly found (and modelled) neurotransmitters: AMPA (fast excitatory), NMDA (slow excitatory), $\mathrm{GABA_A}$ (fast inhibitory) and $\mathrm{GABA_B}$ (slow inhibitory); readers are referred to the excellent review article by Destexhe for a more in-depth discussion; (Destexhe et al., 1994). There are broadly three approaches to modelling synapses, which cover an increasing level of biophysical detail, and we will briefly look at each of these in turn.

The simplest type of synapse gives an instantaneous current injection; we call these delta synapses by analogy to the Dirac delta function which they invoke. If we recall equation 2.2 from earlier (the leaky integrate and fire model rescaled to zero resting potential), add in an external current injection I_E and divide through by the total membrane capacitance C_M, we may add a delta synapse term to this model by the addition of a new term, I_S, which represents a synaptic current injection. I_S is a sum of delta functions, each of which represents the input from a single presynaptic spike s occurring at time m^s, multiplied by a parameter w_S which reflects the strength of the connectivity (as well as representing the efficacy of the connection, this effectively incorporates the synaptic conductance g_S and specific membrane capacitance c_M). In practice, g_S is usually parameterised by indices representing the particular pair of neurons being connected, $g_S^{i,j}$, allowing for different connection strengths between different neurons, but this is omitted here for clarity.

$$\begin{aligned}\frac{dV}{dt} &= -\frac{V}{\tau_M} + \frac{I_E}{C_M} + I_S \\ I_S &= \sum_{m^s<t} \delta(t - m^s) w_S\end{aligned} \tag{2.11}$$

Equation 2.11 defines Stein's model (Stein, 1965, 1967) which will be revisited in more detail in chapter 5. The simplicity of a delta synapse comes from the fact that it acts as an instantaneous external current injection, immediately having its

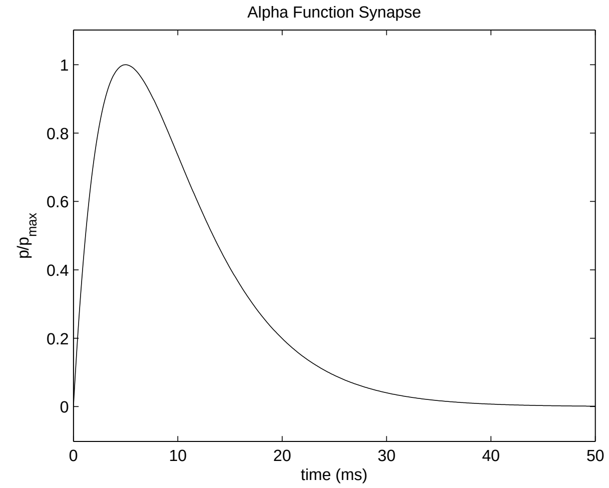


Figure 2.6: Vesicle opening probability of an alpha function synapse, showing its temporal evolution.

full impact. The temporal dynamics related to the effect of the synaptic input on the membrane potential are thus controlled entirely by the neuron's intrinsic temporal properties, i.e. the membrane time constant τ_M, and follow a simple exponential decay. This time-independence (other than the time of the pre-synaptic spikes) renders it useful for both efficient computation and tractability, but it is also inherently unrealistic. We have already seen that biophysically plausible synapses are characterised by different time scales, which emphasises the need for the temporal dynamics of the synapse to be taken into account. One popular way to do this is to use an alpha function (shown in equation 2.12), which describes the synapse in terms of the time of the peak and the subsequent delay, which are both controlled by a single parameter τ_S. This means that the rise time and decay time are directly related, allowing this to be used to model both faster and slower acting synapse types where the the ratio of the rise and decay times remains relatively constant.

$$\begin{aligned} P_S &= \frac{P_{max} t}{\tau_S} \exp(1 - \frac{t}{\tau_S}) \\ I_S &= g_S P_S \end{aligned} \tag{2.12}$$

The general approach in an alpha function is related to that of channel conductances in HH neurons seen earlier: there is an opening probability P_S (here for post-synaptic vesicles; there for ion channels) which in practice represents a proportion of open units when taken to the macro scale at which these models operate (representing many individual ion channels or individual vesicles), which translates into a synaptic current by a simple product with a fixed conductance (g_S here); for this reason the opening probability is generally rescaled to allow the time derivative to equal unity by dividing by a predetermined maximum probability, so that the current injection due to the synapse is simply g_S, as in a delta synapse, but now spread over time. As pointed out by Destexhe et al (Destexhe et al., 1998, 1994), a significant weakness of alpha synapses is that they have no obvious way to integrate multiple spikes. This means that the highly probably scenario, especially for slower synapses, of a second spike arriving before the current due to the first spike has decayed to an insignificant level, will cause problems for this model. Possible workarounds for this include truncating the effect of an existing spike when a new one arrives, or summing the effect of the two, or some more general function which can be applied to the opening probability in the event of two spikes occurring within some proportion of the synaptic time constant τ_S, but these are all difficult to generalise to an unknown number of spikes occurring within a reasonable time period.

Instead, Destexhe et al (Destexhe et al., 1998, 1994) have proposed more closely following the analogy with ion channels in HH neurons by adopting kinetic markov models. Here each type of synapse, AMPA (fast excitatory), NMDA (slow excitatory), $\mathrm{GABA_A}$ (fast inhibitory) and $\mathrm{GABA_B}$ (slow inhibitory), are characterised by a synaptic conductance specific to the synapse type, analogous with a conductance specific to a channel type, a synaptic reversal potential directly analagous to an ion channel reversal potential (also specific to the synapse type) reflecting the relationship between the effect of the synaptic current and the membrane potential, and a vesicle opening probability (proportion) analagous to a channel opening probability (proportion), which like the latter is controlled by kinetic equations. Given a current time t and a presynaptic spike at time t', we have the synaptic equations given as follows.

N represents a square pulse of neurotransmitter release that extends for a period of T ms after the presynaptic spike:-

$$\begin{cases} N &= 1 \quad \text{for} \quad (t-t') \leq T \\ N &= 0 \quad \text{for} \quad (t-t') > T \end{cases} \tag{2.13}$$

An AMPA synapse is fast (due to the high opening and closing probabilities), and excitatory (due to the high reversal potential):-

$$\begin{aligned} I_{AMPA} &= g_{AMPA}P_{AMPA}(V - E_{AMPA}) \\ \frac{dP_{AMPA}}{dt} &= N\alpha_{AMPA}(1 - P_{AMPA}) - \beta_{AMPA}P_{AMPA} \\ \alpha_{AMPA} &= 1.1 \text{ mM}^{-1}\text{ms}^{-1} \\ \beta_{AMPA} &= 0.19 \text{ ms}^{-1} \\ g_{AMPA} &= 0.7 \text{ nS} \\ E_{AMPA} &= 0 \text{ mV} \end{aligned} \tag{2.14}$$

An NMDA synapse is slow (due to the much lower opening and closing probabilities), but still excitatory (due to the same high reversal potential):-

$$\begin{aligned} I_{NMDA} &= g_{NMDA}P_{NMDA}(V - E_{NMDA}) \\ \frac{dP_{NMDA}}{dt} &= N\alpha_{NMDA}(1 - P_{NMDA}) - \beta_{NMDA}P_{NMDA} \\ \alpha_{NMDA} &= 0.072 \text{ mM}^{-1}\text{ms}^{-1} \\ \beta_{NMDA} &= 0.0012 \text{ ms}^{-1} \\ g_{NMDA} &= 0.3 \text{ nS} \\ E_{NMDA} &= 0 \text{ mV} \end{aligned} \tag{2.15}$$

A GABA$_{\mathrm{A}}$ synapse is fast (due to the much high opening and closing probabilities), but now inhibitory (due to the low reversal potential):-

$$\begin{aligned} I_{GABA_A} &= g_{GABA_A}P_{GABA_A}(V - E_{GABA_A}) \\ \frac{dP_{GABA_A}}{dt} &= N\alpha_{GABA_A}(1 - P_{GABA_A}) - \beta_{GABA_A}P_{GABA_A} \\ \alpha_{GABA_A} &= 5 \text{ mM}^{-1}\text{ms}^{-1} \\ \beta_{GABA_A} &= 0.19 \text{ ms}^{-1} \\ g_{GABA_A} &= 0.8 \text{ nS} \\ E_{GABA_A} &= -80 \text{ mV} \end{aligned} \tag{2.16}$$

A GABA$_{\mathrm{B}}$ synapse is slow (due to the much lower opening and closing probabilities), and still inhibitory (due to another low reversal potential):-

$$\begin{aligned} I_{GABA_B} &= g_{GABA_B} P_{GABA_B} (V - E_{GABA_B}) \\ \frac{dP_{GABA_B}}{dt} &= N\alpha_{GABA_B}(1 - P_{GABA_B}) - \beta_{GABA_B} P_{GABA_B} \\ \alpha_{GABA_B} &= 0.09 \text{ mM}^{-1}\text{ms}^{-1} \\ \beta_{GABA_B} &= 0.19 \text{ ms}^{-1} \\ g_{GABA_B} &= 0.06 \text{ nS} \\ E_{GABA_B} &= -77 \text{ mV} \end{aligned} \tag{2.17}$$

These synaptic conductances can be added to the IF neuron model of equation 2.6 as additional conductances:-

$$\begin{aligned} c_M \frac{dV}{dt} &= I_M + \frac{I_E}{A_M} + I_S \\ I_M &= g_L(E_L - V) \\ I_S &= I_{AMPA} + I_{NMDA} + I_{GABA_B} + I_{GABA_B} \end{aligned} \tag{2.18}$$

They can be added to the HH neuron model of equation 2.10 in the same way, which gives a highly detailed and biophysically plausible model of neuronal activity when receiving inputs from other connected spiking neurons:-

$$\begin{aligned} c_M \frac{dV}{dt} &= I_M + \frac{I_E}{A_M} + I_S \\ I_M &= g_L(E_L - V) + g_K p_K (E_K - V) + g_{Na} p_{Na} (E_{Na} - V) \\ I_S &= I_{AMPA} + I_{NMDA} + I_{GABA_B} + I_{GABA_B} \end{aligned} \tag{2.19}$$

This demonstrates the unified neuron and synapse modelling approach which is adopted later in chapter 5. The core is a differential equation representing the change in the membrane potential over time with the rate is modulated by the specific membrane capacitance. The membrane potential heads towards a resting potential given in general leakage term at a rate specified by the leakage conductance, in the absence of any external input. An external current injection will depolarise the neuron to an extent modulated by the surface area of the neuron. In an IF model, the neuron will either reach a pre-specified threshold V_{thre}, fire and reset, or it will simply decay back to the resting potential after the external input is switched off. In an HH model, more complex feedback dynamics involving potassium and sodium ion channels with activity and inactivity gates give the neuron an implicit soft threshold instead, but in most instances whether or not an action potential is elicited remains the same. As well as an external current injection, neurons can

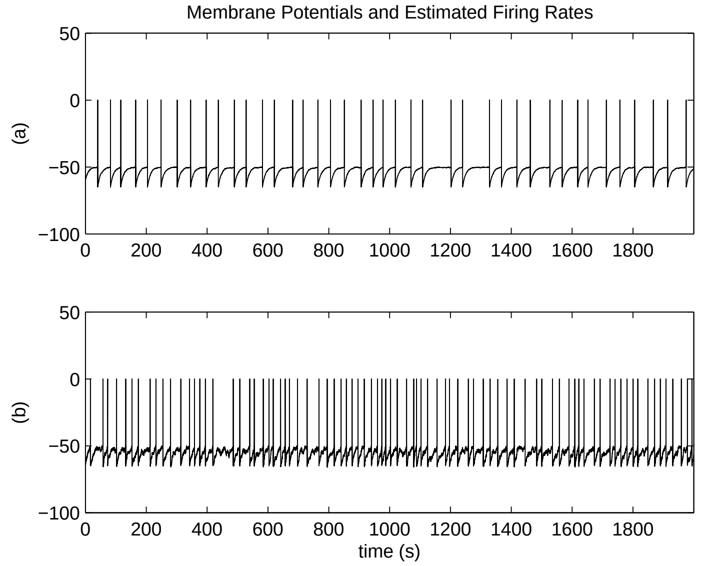


Figure 2.7: Membrane potential for two integrate and fire neurons, effectively showing the spike trains.

receive synaptic input from other neurons (or more accurately presynaptic spike trains of unspecified origin). Like ion channels, these are governed by kinetic equations that control the proportion of maximal conductance at any one time, which have an opening and closing probabilities which determine their speed of action, and include a reversal potential which determines their direction of action (excitatory or inhibitory). The tools of conductances, reversal potentials, opening and closing probabilities and kinetic Markov models provide a convenient unified framework for neuronal modelling which has proved highly popular and successful to date.

2.2 Neural Information Processing

2.2.1 Neural Firing: Coding and Statistics

There are a wide variety of possibly meaningful modes of communication between neurons in the brain, but the one that is by far the most prevalent, widely studied and believed to be most important, is the action potential. We have seen in section 2.1 how action potentials are generated, and how this aspect of neural behaviour can be captured in modelling. A series of actions potentials, or spikes as they are sometimes called, generated by a single neuron over time, is a spike train. An example of two spike trains arising from two different neurons is shown in figure 2.7. The spike train is the basic building block of neural information transmission. There are essentially two ways that a spike train can carry information, which is either in the relative timing of individual spikes (temporal coding) or simply how frequently spikes are occurring (rate coding). Although these two measures are in fact related (see the discussion by Theunissen and Miller (Theunissen & Miller, 1995) for a more detailed discussion of this), the rate coding model dominates in computational neuroscience, and is certainly more compatible with the concept of the brain as a noisy stochastic information processing system, so we will focus exclusively on rate coding from here on.

The estimation of neural firing rates can be done in a number of ways. If a spike can be defined as occurring at a given point in time by reaching a predefined threshold, then it can be characterised by a delta function $\delta(t - t_m)$, where m labels a given spike. The sum of these delta functions describes a spike train, and the integral of this sum over time will give the total number of spikes in the spike train. Given integration over a time window T, we can therefore estimate the firing rate during this window as a simple time average of the number of spikes:

$$\begin{aligned} r &= \frac{1}{T}\int_0^T \sum_{m=1}^{n} \delta(t - t^m) dt \\ \Rightarrow r &= \frac{n}{T} \end{aligned} \tag{2.20}$$

An obvious problem with this scheme is that it gives just a single firing rate for an entire neural spike train. Whilst for relatively short spike trains this may be acceptable, a more general solution to cover spike trains of all lengths is desirable. The obvious extension of the previous approach is to divide a longer spike train into relatively short segments and estimate the firing rate for each segment separately. However, this procedure is unacceptably sensitive to the potentially arbitrary placement of the window boundaries. To overcome this, an alternative approach is to use a sliding window, and divide by the window area (or more efficiently, use a window with an integral of 1). For smooth estimation, the window should not be rectangular, and so the use of a more continuous window, typically a Gaussian, is usually adopted. The only significant disadvantage to this approach is that the window has a width parameter (the variance in the case of a Gaussian window), the value of which determined the relative smoothness of the firing rate. Choosing a value for

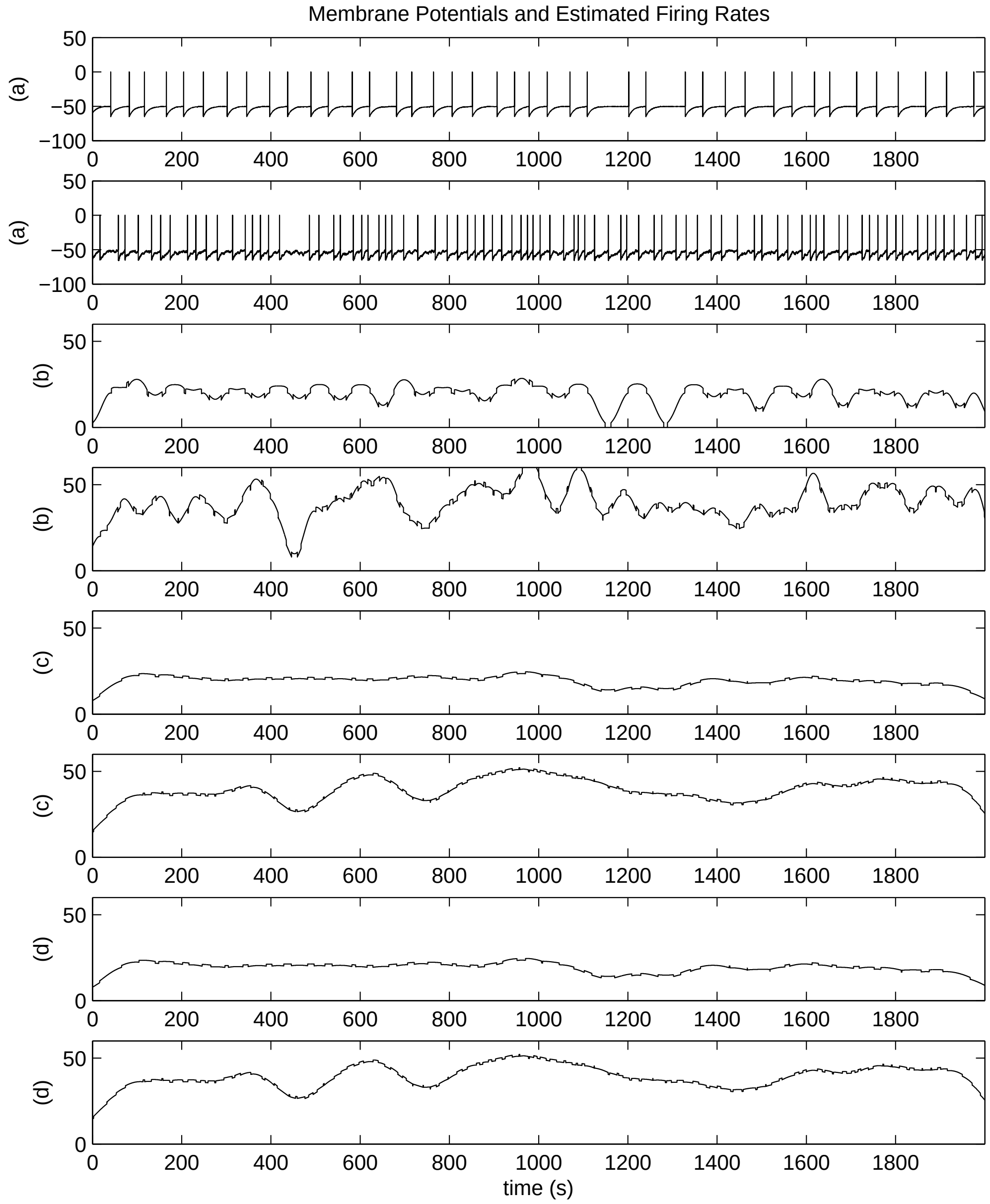


Figure 2.8: *a*: Membrane potential for two integrate and fire neurons (same as in figure 2.7, effectively showing the spike trains. *b*: Firing rates from convolution with a Gaussian smoothing kernel with a small variance. *c*: Firing rates from convolution with a Gaussian smoothing kernel with a larger variance. *d*: Firing rates from kernel density estimation with a Gaussian smoothing kernel with the same variance as *c*.

this parameter is typically a matter of heuristics. The spike trains previously shown in figure 2.7 are shown again in figure 2.8, but now also with firing rates estimated using a Gaussian sliding window with three different widths. Apart from producing a smoother estimate, the key advantage of this sliding window approach is that it produces a continuous firing rate estimate that constantly varies over time, rather than only after each fixed window.

Given a spike train defined as a series of delta functions, and if we interchange the time axis for a probability axis, then the Gaussian sliding window approach to estimating neural firing rates is in fact equivalent to kernel density estimation (KDE) with a Gaussian kernel (see section A for an intuitive introduction to kernel density estimation for functions of one and two variables), centred at the mid-time point, where the x-axis is labelled as a time axis but treated as a probability axis, and the resulting firing rate estimate is identical to that using a Gaussian sliding window. This is due both to the fact that the function describing the probability of obtaining a particular value in a continuous random variable can be also be given as a delta function, and the fact that a Gaussian kernel is symmetric. A spike that is exactly at the centre of a Gaussian sliding window will contribute the peak value of that window to that particular location. A spike located somewhere else within the window will contribute a lower value to that location, and these values will be summed to give the total value. As the window moves along to the other location the situation will be reversed. For a Gaussian kernel that is explicitly centred only at each spike, but for which the contribution is added to all time points covered by the kernel while at that location, will similarly have a full contribution from the first spike while at that location, and a secondary contribution from the other spike when the kernel is later centred at other the spike, and vice versa again. The result is that the estimated function will look identical (this can be seen empirically in figure 2.8), and the consequence of this is that neural firing rates may be efficiently estimated using a kernel density estimation approach, and in particular, that the number of kernels need only be equivalent to the number of spikes regardless of the sample rate, making this approach considerably more efficient and scalable. This is the approach used to estimate firing rates for the remainder of this work. [1]

[1] This equivalent is mentioned here partly because this KDE approach is not explicitly reported

In addition to estimating the firing rate from the spike train of a single neuron, we may wish to estimate the mean firing rate of a population of neurons. Gerstner and Kistler (Gerstner & Kistler, 2002) suggest an approach to measuring what they call the population activity, and which is sometimes also loosely described as the mean field potential, based on counting the number of neurons in the population that emit a spike within a small time window (n_{spikes}):-

$$\begin{aligned} A(t) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{n_{spikes}(t, t+\Delta t)}{N} \\ \Rightarrow A(t) &= \frac{1}{N} \sum_{i=1}^{N} \sum_{s} \delta(t - m_i^s) \end{aligned} \tag{2.21}$$

It can clearly by seen from equation 2.21 that the population activity is effectively representing a collection of spike trains each characterised as a sum of delta functions. The obvious parallel with the single neuron approach allows us to calculate the firing rate as a simple time derivative, which effectively becomes a simple average of the individual firing rates:-

$$\begin{aligned} r &= \frac{1}{T} \int_0^T \frac{1}{N} \sum_{i=1}^{N} \sum_{s} \delta(t - m_i^s) dt \\ \Rightarrow r &= \frac{n_{spikes}}{NT} \end{aligned} \tag{2.22}$$

where $n_{spikes} = \sum_{i=1}^{N} n_i$ sums all of the spikes within the time window across all of the neurons in the population. This provides a formal justification of the intuitively satisfying approach of simply averaging the activation of all neurons within a given population in order to obtain the mean population firing rate. In this way, the more powerful KDE approach to estimating individual neuron firing rates outlined earlier can also be easily adapted to calculate population averages, by simply averaging the individual firing rates at any given time. In practice, however, any form of smoothing has two less desirable side effects: sensitivity to endpoints and artificial infusion of information. The former exists because at the beginning and the end of a time series the smoothing kernel/window will cross over into undefined time points. The usual approach of zero-padding results in lower estimates for these areas. This issue is particularly important when the time series is relatively short in comparison to the full-width half-maximum (FWHM) or variance of the kernel/window; this effect

anywhere in the literature to our knowledge.

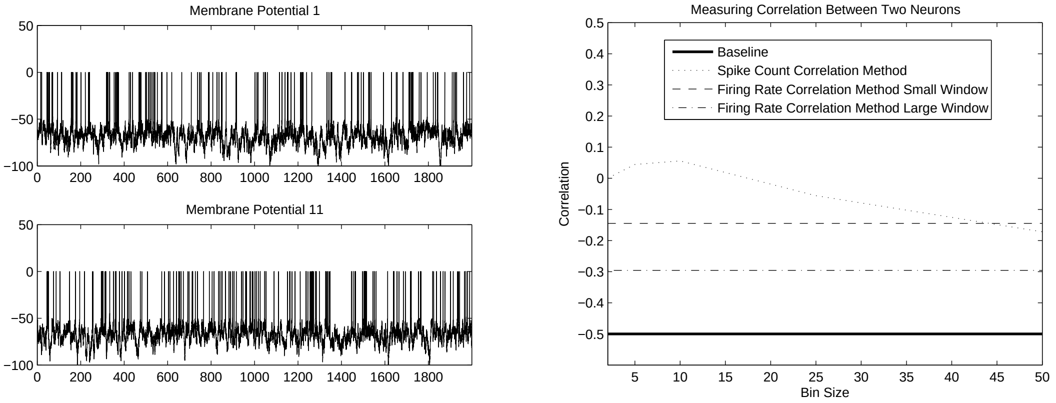


Figure 2.9: Different methods for estimating the correlation between two neural spike trains. *Left*: The spike trains. *Right*: Correlation curves from spike counts using different bin sizes (dotted line), correlation values (independent of bin size) for smoothed firing rates, for a small smoothing kernel and a larger smoothing kernel.

is described for the particular case of spike train simulation later in this section. The artificial boost to the signal information content comes from the fact that smoothing inherently leads to greater predictability (lower entropy), i.e. two close time points are more likely to be similar. This is obviously relevant in cases where information theoretic measures are being used, including the simulations shown in chapter 4.

The mean firing rate of a group of neurons is probably the most commonly required statistic of population activity, but in this thesis we are centrally concerned with how the firing of one neuron affects other neurons, and in particular, if the spike trains of connected neurons show any pattern of being positively or negatively correlated with each other. We therefore need a means of measuring correlation between spike trains. For determining correlations between the spike trains of different groups of neurons, the correlation multivariate analysis of variance procedure developed by Wu, Kendrick and Feng (Wu, Kendrick, & Feng, 2007) can be used. However, in this work we are exclusively concerned with correlations between individual spike trains, and there are essentially two ways of approaching this measurement, which reflect the two approaches to estimating firing rates. One is to take the continuous firing rate curves estimated by a sliding Gaussian window or equivalently KDE, and measure the Pearson correlation coefficient of these (and in the more general case, the complete cross-correlation can be measured).

For long spike trains or relatively short windows/kernels, this approach is reliable and straightforward. However, if the spike train is too short relative to the window size, which is often the case in spike train simulations, then the firing rate estimate during the rise and fall periods (the start and end of the spike train for which a significant part of the kernel falls outside the maximum or minimum time periods), which is highly unreliable and will always take roughly the same shape (an increase at the start and a decrease at the end), will dominate and make the correlation value artificially positive (because the similarity in shape means that the estimated firing rates are highly correlated in these periods). A popular alternative approach, which makes the sensitivity to different window sizes more explicit, is to use a simple windowed spike count approach instead, but with different window sizes, resulting in correlation curves (correlation values plotted against bin size). Figure 2.9 shows the results of the different correlation measures for a pair of neurons with an inhibitory GABA$_\mathrm{A}$ connection between them. It can be seen that there is some sensitivity to bin size, and correlation values for a relatively short spike train and just two neurons is rather unreliable, but the measures taken together give a clear indication of negatively correlated spike trains. This demonstration also gives a clear flavour of this work, and in particular foreshadows the results outlined in chapter 5.

We saw previously in equation 2.20 that a set of spikes can be idealised as a sum of Dirac Functions. If we accept that information transmission from one neuron to another is exclusively by means of spikes, rather than sub-threshold membrane potential variability (a biophysically plausible simplification for the neural systems discussed here), then we can treat a neural spike train as a point process. If we further assume that the spikes in a given spike train are independent from one another (another biophysically plausible, if not entirely justified, assumption), then we can characterise a spike train as a Poisson process, where the probability of obtaining n spikes in a time window T is given by:-

$$P(n) = \frac{(rT)^n}{n!}\exp(-rT) \tag{2.23}$$

The Poisson approximation to a neural spike train has been widely used in computational neuroscience (Rieke, Warland, Steveninck, & Bialek, 1997), and we make use of it ourselves in chapters 4 and 5. The Poisson distribution has the unusual property of having a mean and variance equal to each other, and fully describing the density function which therefore takes a single parameter; as shown in equation 2.23 for a Poisson spike train this is the firing rate r. In the homogeneous case, the firing rate is treated as constant over time, but in the more general inhomogeneous case, the firing rate of a Poisson process can vary over time. In section 4.4 in chapter 4 we will see how the mean and variance of a neural Poisson process can be separated out as signal and noise using a diffusion approximation (Feng & Tirozzi, 2000). Altogether, the Poisson spike train is fundamental to the statistical modelling of neuronal response.

The classical homoegeneous Poisson process is a rewnewal process, meaning that all events are independent. The probably density for interspike intervals in a Poisson process reflects this fact:

$$P_{ISI}(\tau) \quad = \quad r \exp(-r\tau) \tag{2.24}$$

where $P_{ISI}(\tau)$ is the density for an interspike interval τ and r is the firing rate as before. The independence of the ISIs in this renewal process ensures that the autocorrelation over a sustained period will tend towards zero, i.e. the spike train from a single Poisson neuron cannot benefit from negative correlation on its own. However, neurons are not propelry Poisson renewals processes. In a series of studies (Longtin, 1993; Chacron, Longtin, & Maler, 2001; Chacron, Lindner, & Longtin, 2004; Lindner, Longtin, & Chacron, 2005; Middleton, Chacron, Lindner, & Longtin, 2008), Chacron, Longtin, Lindner and colleages have examined the firing dynamics of the classic leaky integrate-and-fire neuron (but with a randomly varying firing threshold), and have found that the fixed threshold reset behaviour of a neuron results in negatively-correlated ISIs. Using band-limited white noise, Chacron et al (Chacron et al., 2004) examined the information transfer capabilities of neurons with fixed and random resets, the former of which have negatively-correlated ISIs typical of real neurons while the the latter have uncorrelated ISIs characteristic of a renewal process. They found that although the SNR decreased overall as a result of the fixed reset, information transmission at low frequencies actually improved. This suggests a potentially beneficial role for negative correlation within individual neurons, as well as across neurons. Although not discussed in detail by Chacron et al, the situation in biological neurons is likely to be even more biased towards negatively-correlated ISIs due to cumulative refractory effects. In other words, if a neuron has fired twice in quick succession (hence giving a short ISI), the additional refractory effect is likely to lead to a comparatively longer ISI to follow. Although negative correlations within individual spike trains is not the subject of the studies outlined in later chapters here, and will only be briefly touched upon in chapter 5, we believe that potentially benefits for information processing use broadly the same mechanism - decreased sensitivity to noise due to decreased fluctuations in the firing rate over a sustained period, in comparison to that which would be obtained if the ISIs were independent. It effectively acts as a damper on system sensitivity, and as we will see in chapter 4 in the context of correlations across a network of neurons, can even be used to tune the sensitivity to a level optimal for information transfer.

2.2.2 Information Theory

The measures of correlation described in the previous section give an indication of how the spike trains of different neurons are related. Those neurons are not required to be connected in any way (though they typically will be) and nor are they assigned to any particular functional role with regard to the neural encoding or decoding of a stimulus. In short, spike count or firing rate correlation can tell us two or more neurons relate to each other, but nothing about how they relate to anything external to themselves, or how they process information per se. In order to do this, we need to define a neural information channel (one or more neurons in this case), a stimulus s, and a response r, the latter of which we will designate here as a firing rate. We can then invoke the concepts of information theory created by Shannon (Shannon & Weaver, 1949) (see also the useful introduction by Pierce (Pierce, 1980)) in order to quantify the amount of information that passes through the channel in a rigorous fashion. We will give an outline and derivation of the basic concepts and equations of this theory for discrete random variable only here, for clarity, but it should be noted at the outset that the theory applies equally well to continuous variables (such as a firing rate that can take any real value) by making some simple adjustments. Information theory is based on the probability distributions of both the stimulus and response. The central concept is to measure how much uncertainty about something is removed after receiving a communication of some sort. This in turn invokes concepts of the prior probability $P_{prior}(x)$ and the posterior probability $P_{post}(x)$, and expressed in units of bits, which are obtained by taking the base-2 logarithm, we have:-

$$I = \sum_x P_{post}(x) \log_2 \frac{P_{post}(x)}{P_{prior}(x)} \tag{2.25}$$

This states simply that the information given by a communication event is the difference of the probability before and afterwards, for all possible events (commonly termed the alphabet) x, weighted by their relative importance which is determined also by their probability of occurrence. Where the communication event is a member of the alphabet and thus determines the posterior probability with a certainty of 1 for that event and a certainty of 0 for all other events, which will generally be the case for situations considered in this work, equation 2.25 reduces to:-

$$I = -\log_2 P_{prior}(x) \tag{2.26}$$

Equation 2.26 tells us the information gained from an event, which relates directly to how probable that event was anyway (the more probable it was, the less information was gained by it actually happening). If we average across all responses, weighted by the probability of that response occurring, we get the entropy, which is therefore a single measure that applies to the entire response variable x:-

$$H_x = -\sum_x P_{prior}(x) \log_2 P_{prior}(x) \tag{2.27}$$

This tells us the average predictability of the prior probability of x, which gives a measure of how much information it is possible to obtain from x actually taking a known value. If x had only one value that occurred with any significant likelihood, then we would expect the entropy to be rather low (since although there are high information events, these are very rare). In terms of a neuron as an information channel, and dropping the subscript which is assumed to be clear by the context for clarity, we have an entropy for both the stimulus and the response:-

$$\begin{aligned} H_s &= -\sum_s P(s) \log_2 P(s) \\ H_r &= -\sum_r P(r) \log_2 P(r) \end{aligned} \tag{2.28}$$

To measure neural information processing, it is not enough to be able to characterise the stimulus and the response separately. We would like to know to what extent they are related, that is to what extent a given stimulus value is associated with a given response value, across all of the possible stimuli and responses. In order to do this, we first need to invoke the joint probability of getting any give stimulus-response pair, $P(s, r)$. This can be characterised in terms of their conditional and marginal probabilities:-

$$P(r, s) \quad = \quad P(r|s)P(s) = P(s|r)P(r) \tag{2.29}$$

Using Bayes theorem, we can relate the two conditional probabilities to each other their respective marginal probabilities (and thus calculate one from the other without the need to know the joint probability):-

$$P(r|s) \quad = \quad \frac{P(s|r)P(r)}{P(s)} \tag{2.30}$$

Conditional probabilities are often given the role of posterior probabilities in a Bayesian context, and the marginal probabilities as prior probabilities. Hence $P(r)$ may be regarded as all we know about the probability of obtaining a given response r_i before anything happens, but after stimulus event s_j that probability can be updated to take into account this new information. In this way, we can characterise the information using the formula from equation 2.25 but now with the new probabilities:-

$$I(r) \quad = \quad \sum_s P(s|r) \log_2 \frac{P(s|r)}{P(s)} \tag{2.31}$$

If the stimulus and response are unrelated, then the average stimulus probability will be the same whether or not the response is known, $P(s|r) = P(s)$, and therefore the information measure in equation 2.31 will be equal to zero. In other words, if the response and stimulus are unrelated, no information is obtained from knowing the response. The same argument can be made in parallel for the stimulus, by invoking Bayes theorem again as in equation 2.30 by switching the variables. Equation 2.31 gives us the information for any single given response, averaged across all possible stimuli. This can be generalised in one further step, noting the parallel argument for the stimulus, by averaging across all of the responses:-

$$MI = \sum_r P(r) \sum_s P(s|r) \log_2 \frac{P(s|r)}{P(s)} \tag{2.32}$$

Equation 2.32 is called the mutual information of the channel, and gives a direct measure of how much the response can tell us about the stimulus on average, or vice-versa. If the neural response is firing rate, then we are essentially looking at how much firing rates vary as a function of a given input stimulus, versus how much they vary for other reasons unrelated to the stimulus. We can reformulate the mutual information to more explicitly reflect this intuitive understanding by invoking the concept of the noise entropy H_{noise} which is the entropy due to things other than the stimulus. Starting with the mutual information expressed above, but in the parallel form (switching the stimulus and response variables), we have:-

$$\begin{aligned}
MI &= \sum_{r,s} P(s)P(r|s) \log_2 \frac{P(r|s)}{P(r)} \\
\Rightarrow MI &= -\sum_r \sum_s P(s)P(r|s) \log_2 P(r) + \sum_{r,s} P(s)P(r|s) \log_2 P(r|s) \\
\Rightarrow MI &= -\sum_r P(r) \log_2 P(r) + \sum_{r,s} P(s)P(r|s) \log_2 P(r|s) \\
\Rightarrow MI &= -\sum_r P(r) \log_2 P(r) + \sum_{r,s} P(s)P(r|s) \log_2 P(r|s) \qquad (2.33) \\
\Rightarrow MI &= H_r - H_{noise} \\
H_r &= -\sum_r P(r) \log_2 P(r) \\
H_{noise} &= -\sum_{r,s} P(s)P(r|s) \log_2 P(r|s)
\end{aligned}$$

This measure has three key benefits. The first is that, unlike other direct measures of information flow such as the correlation or the signal-to-noise ratio, this uses all orders of statistical information rather than just the first two, and hence makes no assumption about a Gaussian form of relevant variables or does not higher order information. Secondly is that the measure is sufficiently general that it is in no way limited to neuron modelling scenarios. It will be seen in section and chapter 3 that these same information theoretic measures can be applied as principles to help understand the known characteristics of visual neurons. Finally, when used in the context of modelling neural response with artificial neuron models such as those outlined in sections 2.1.3 and 2.1.2, it is entirely ambivalent as regards the number of neurons (it may be one or an entire population), how many layers of neurons are involved, and what the nature of the stimulus is (it could be synaptic inputs from other modelled neurons, externally modelled spike trains simulating the output from other modelled neurons, or an external electrode current injection). It is finally worth noting that there is a hard limit on the amount of information that can be transmitted. Recalling equations 2.28 and 2.33, we see that the response variability and the stimulus variability impose an upper bound on the amount of information that can be transmitted through the channel. Even if the noise entropy is minimal (in other words, if changes in the response are associated with changes in the stimulus), which represents a minimal information loss due to the properties of the channel itself, then a low response entropy will result in a low mutual information, simply because there is not much information possible at the output regardless of how well related it is to anything else. Similarly, it the stimulus has a low entropy, then there is little opportunity for the information transmission properties of the channel to be used. This issue is of particular importance in the phenomenon of suprathreshold stochastic resonance, as we will see in the next section and in chapter 4.

2.3 Stochastic Resonance

The presence of noise in a biological system that has carefully evolved over millions of years raises the obvious question of why it remains present. There are two broad possibilities: (1) it is an unfortunate but necessary by-product of a natural system made with less than total precision; (2) it serves a functional role. (1) has historically been the prevailing view in neuroscience, and there is undoubtedly some support for this. Complex patterns of synaptic connectivity (Shadlen & Newsome, 1998; Kistler & De Zeeuw, 2002), the considerable variation in *in vivo* firing patterns and even firing rates under the same stimulus conditions (Zador, 1998), the stochastic nature of biophysical synapse operations, and the presence of Johnson noise (Manwani & Koch, 1999), all contribute to the belief that noise in neural systems is inevitable. Recently, however, support has grown for the second possibility, at least in addition to the first.

The theory of stochastic resonance became mainstream in the early 1990s (see (Gammaitoni et al., 1998), for a comprehensive review and early history of stochastic resonance), and although originally proposed for bistable physical systems driven by

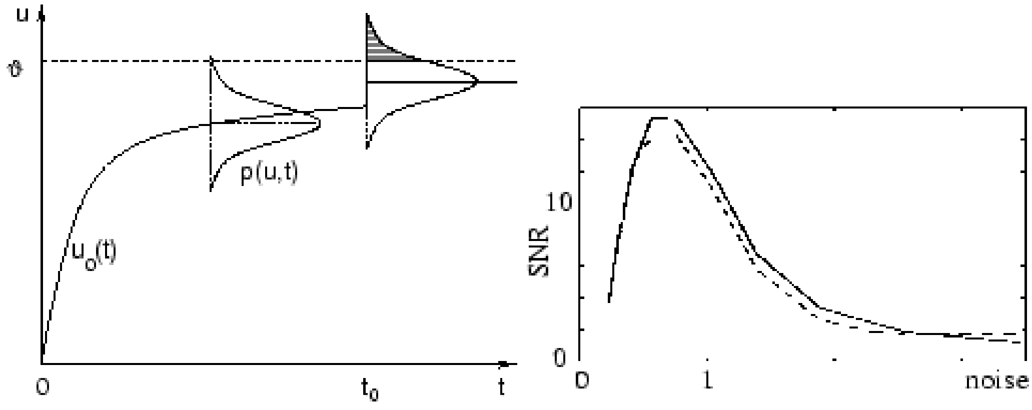


Figure 2.10: *Left*: The mechanism of stochastic resonance, formalised as escape noise. *Right*: Mutual information as a function of noise. It is apparent that the presence of some noise aids signal transmission. See text for details. From (Gerstner & Kistler, 2002) ©CUP.

periodic stimuli, was soon generalised to the wider group of threshold dynamical systems driven by some time-varying (possibly aperiodic) stimulus (see (Collins, Chow, & Imhoff, 1995; Collins, Chow, Capela, & Imhoff, 1996)) for the early development in this area). Stochastic resonance was soon applied to spiking neuron models, including the Fitzhugh-Nagumo model and the leaky integrate-and-fire model (Jung, 1995; Kosko & Mitaim, 2003), and rapidly became widespread ((Moss, Pierson, & O'Gorman, 1994; K. & F., 1995) give a review of the early development of these models).

The mechanism for traditional stochastic resonance in neural systems (Wiesenfeld & Jaramillo, 1998) is most widely interpreted in the framework of probabilistic threshold crossing given a subthreshold time-varying driving input. The central effect is shown in figure 2.10, adapted from (Gerstner & Kistler, 2002) and (Plesser & Gerstner, 2000), which highlights both the mechanism and the information processing benefits. Stochastic resonance works in this context due to the fact that neural thresholds (in common with most others) are unidirectional. If a noise-free membrane potential is constantly below threshold due to insufficient input, then no output spikes will ever be emitted, and so no information can be transmitted (in this extreme case). If zero-mean noise is added to the system, then sometimes the momentary noise will push the signal below the true value, which will have no direct effect on the output. Sometimes, however, the momentary noise will be positive and push the signal higher; this has the possibility to push the signal over the threshold.

How likely this is depends on far the true signal is subthreshold and the variance of the noise distribution (assuming a Gaussian form); this is the mechanism shown in the left side of figure 2.10. If the noise strength (equivalent to variance for zero-mean Gaussian noise) is too high, the firing rate will slowly start to saturate and information transmission will decrease; if the noise strength is too low, information transmission will fall off rapidly because the system does not reach threshold often enough (effectively creating a sampling problem that gets progressively worse for weaker input signals). There is an optimal point in between that maximises information transmission.

Although the characteristic curve of stochastic resonance has been observed in recordings from biological neural systems (K. & F., 1995), the biophysical plausibility of stochastic resonance remains open to debate. It is known that some neurons have the ability to adjust their threshold levels relative to the input scale which means that they are capable of adjusting their firing thresholds to some extent in response to different signal levels. Retinal Ganglion cells, for example, are know to be able to adjust their thresholds in response to different light levels (Manookin & Demb, 2006). Traditional stochastic resonance requires that the mean input level is consistently some way below the firing threshold, and indeed that the threshold is therefore set at a suboptimal level; that is, that there is an alternative fixed threshold value that would allow better information transmission to occur. While it is possible that neural thresholds are suboptimal for certain types of stimulus, this does not seem to be a likely property on the whole for a highly effective and long evolved information processing organ, and any threshold adaptation would further undermine it.

An important development in stochastic resonance theory was the introduction of the concept of suprathreshold stochastic resonance (Stocks, 2000a, 2000b, 2001a, 2001b). Instead of a subthreshold regime being required for the effect to be present, this relaxes that restriction and adopts a quite different approach based on the population activity of a number of similar units. Originally demonstrated for a comparator array of identical threshold processing units, the central mechanism is that noise in the inputs gives rise to a greater variety of possible summed output

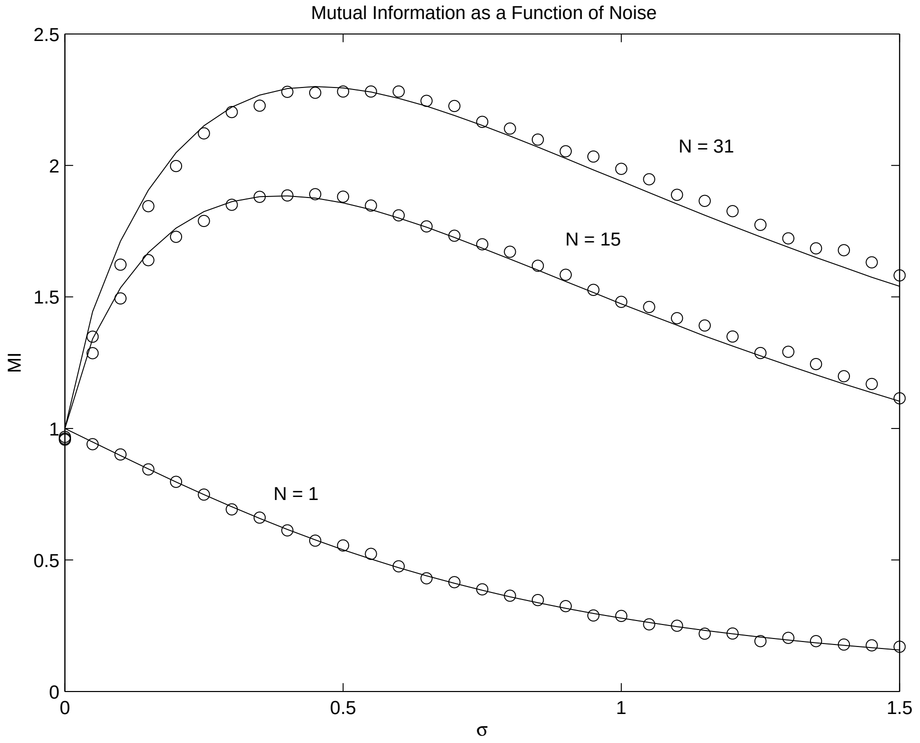


Figure 2.11: Mutual information shown as a function of the noise strength parameter. The SSR effect, information transmission is optimised by a non-zero noise level, is clearly evident. The effect becomes more pronounced for a greater number of processing units. In this figure, the circles represent values from the digital simulation and the lines represent numerical evaluation of eq.4.8; in all subsequent figures, the circles still represent data points but the lines are simply visual aides.

states, which reduces the otherwise severe output entropy restriction on information transmission in the system. Put simply, greater possible variety in the output of the system allows greater possibility of these outputs providing information about the inputs. The result is an information curve (as a function of noise level), shown in figure 2.11 for a replication of the comparator array (this will be discussed in more detail in chapter 4 that looks broadly similar to the curve of traditional (subthreshold) stochastic resonance.

In addition to the discrete time comparator array and, SSR research has also taken place for the continuous time Fitzhugh-Nagumo models (Stocks & Mannella, 2001), and the ubiquitous integrate-and-fire and Hodgkin-Huxley neuron models (Hoch, Wenning, & K., 2003b, 2003a). It has also been studied previously in the specific context of correlation (Hoch et al., 2005),. These previous results, together with our new developments, are discussed in detail in chapter 4.

2.4 Independent Component Analysis and Sparse Coding

In addition to modelling neurons and neural information processing explicitly, this thesis tackles the problem from the other direction as well by looking at the information in natural stimuli as they pertain to neural systems using algorithms that describe the stimuli in terms of a set of representative components. In particular, we are using developments of independent component analysis and sparse coding in the context of receptive fields in the primary visual cortex. This approach was first adopted by Olshausen and Field (Olshausen & Field, 1996) and has since been adopted by a wide variety of other researchers (Hateren & Schaaf, 1998; Hyvärinen & Hoyer, 2000; Hoyer, 2002, 2003; Hurri & Hyvrinen, 2003).

$$\begin{aligned} \mathbf{x} &= \mathbf{A}\mathbf{s} \\ \mathbf{y} &= \mathbf{W}\mathbf{x} \end{aligned} \tag{2.34}$$

Independent component analysis (ICA) and sparse coding seek to represent a set of variables **x** as a linear combination **A** of components **s**, without knowing either the form of the components or the coefficients in **A**, as shown in equation 2.34. The only constraint is that the components are as independent as possible. Under the additional constraint that not more than one of the components is Gaussian in form, ICA algorithms seek to maximise the nongaussianity of the estimated components, typically by optimising a nonquadratic objective function such as $\log(\cosh(x))$. This can also be shown to be the equivalent of trying to maximise the mutual information between the data and the decomposed representation (Hyvärinen et al., 2001). Sparse coding imposes the additional constraint of a sparsity term that seeks to push the representation to be sparse (have as few significantly active components as possible for a given data variable), with a parameter controlling the trade-off. A non-negative sparse coding algorithm has also been developed (Hoyer, 2002), which is similar to sparse coding but with the components separated into positive and negative channels, all of which have positive coefficients. The independent component algorithm is described more fully in chapter 3, and a detailed description and comparison of all three algorithms, especially in the context of receptive fields and natural images, is given in (Durrant, 2003).

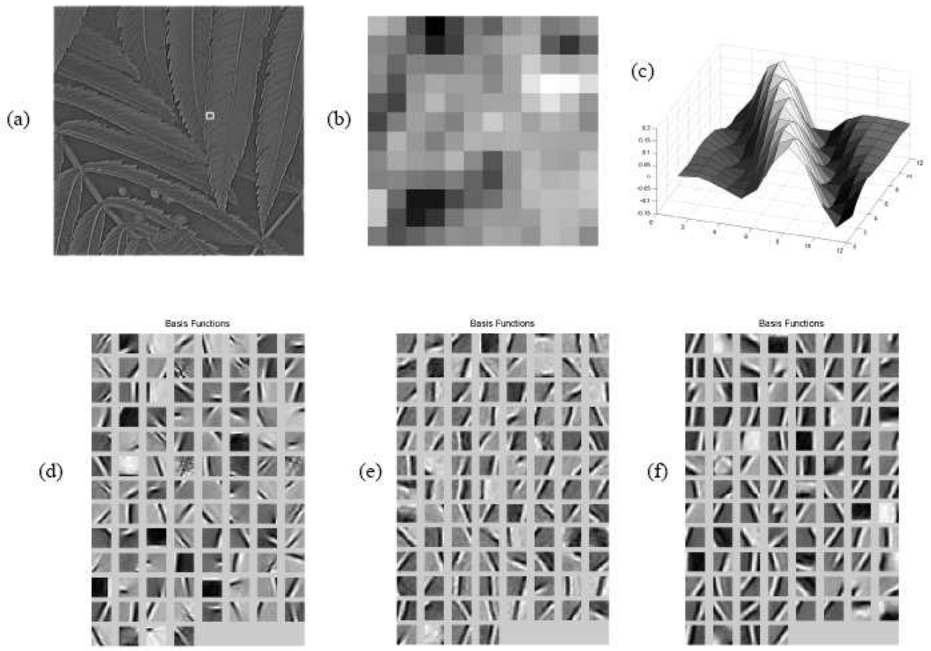


Figure 2.12: *(a)* training image; *(b)* image patch (corresponding to white square from *(a)*); *(c)* close-up of one basis function; *(d)* ICA basis functions; *(e)* Sparse coding basis functions; *(f)* NNSC basis functions

The use of ICA and sparse coding for generating basis functions that resemble V1 receptive fields has its roots in the original redundancy reduction hypothesis of the early visual system by Attneave (Attneave, 1954) and Barlow (Barlow, 1961), which was subsequently revised by the latter as hypothesis that redundancy is not reduced by rather made more explicit (Barlow, 2000; Barlow & Gardner-Medwin, 2001), which also means that coding becomes more sparse. The development of ICA and sparse coding algorithms in the 1990s made investigation of this hypothesis possible, and the application of those algorithms to natural visual images that the visual system has to handle constantly (Simoncelli & Olshausen, 2001; Rolls & Deco, 2001), resulted in basis functions that looked like the V1 receptive fields originally described by Hubel and Wiesel (Hubel & Wiesel, 1959, 1979).

An example of this is shown in figure 2.12, which also illustrates the basis procedure. An image, which has been subject to some basic preprocessing (a), is divided into small patches (one of which is shown in (b). Each of these patches provides a data sample for the algorithms, which are then run on the data. They produce basis functions such as those in (d) for ICA, (e) for sparse coding and (f) for non-negative sparse coding. A surface representation of one of the basis functions (the first of the NNSC set) highlights the classic gabor-like shape that these basis functions can take, and their similarity to V1 receptive fields is also clear from this figure. The precise form of the basis functions varies somewhat across images, and is highly sensitive to parameter settings and image preprocessing as shown in (Durrant, 2003), but nevertheless this result shows that the early visual system appears to have a set of processing units whose response properties are optimised to handle natural images. In the ICA framework, the components can be seen as coefficients of the basis functions, and the activation of a number of V1 neurons, each characterised by one basis function, can therefore represent an image. Given our interest in the functional benefits of negative correlation, an obvious question that is raised about this representation scheme is whether or not negatively correlated components (neural activations) offer an even better way than components that are necessarily forced by ICA to be independent (and therefore uncorrelated). This question is addressed in the next chapter.

2.5 Summary

In this chapter we have presented the background material that lays the foundation for the core research chapters that follow. In particular the tools and techniques used in those chapters have been outlined here. We have seen a very basic overview of biological neurons, and how they can be modelled as integrate and fire neurons which captures the basic electrodynamic behaviour. More detailed behaviour can be modelled using the extra ion channels that make up Hodgkin-Huxley neurons. Similarly, for synapses we have seen models that range from the simplest synapses based on Dirac functions, through to kinetic synapses for several different neurotransmitter types that give a high level of biophysically plausibility.

Neural information processing has been very briefly described, including methods for estimating firing rates, methods for evaluating the correlation between neural spike trains, Poisson approximations for spiking neurons, and information theory as it pertains to neural information transmission. Following these, the topic of stochastic resonance and suprathreshold stochastic resonance was outlined, emphasising the potential importance of noise in neural systems and highlighting the fact that it is not necessarily something to be overcome or removed but may actually be beneficial from a signal processing perspective. Finally, independent component analysis and sparse coding were described in the context of the early visual system and V1 receptive fields in particular. All of the material is necessarily brief and introductory, and limited only to those topics and techniques that are directly relevant to the subsequent chapters.

Chapter 3

Algorithms for Exploiting Negative Correlation

In this chapter, we first explore the benefits of negative correlation in a practical way, through the development of two new algorithms based on independent component analysis. ICA has been used to demonstrate that V1 receptive fields can arise as a set of basis functions that are best able to represent natural images in terms of a simple linear combination (see section 2.4 for an introduction to this). As the component coefficients represent activations of V1 simple cells in this formulation, and we are concerned in later chapters with negatively correlated neural activity, it makes sense to first ask in this more abstract framework whether or not negatively correlated components can offer any benefits in terms of neural image representation. This is the motivation for the algorithms that we present here.

3.1 Overview

3.1.1 Introduction

Since the development of algorithms for Independent Component Analysis (ICA) little more than a decade ago (see (Hyvärinen et al., 2001) for a brief history), it has seen a number of applications, notably in cases of blind source separation (Lee, Lewicki, & Sejnowski, 2000), in models which relate natural image statistics to the properties of the early visual system (Olshausen & Field, 1996, 1997; Hyvärinen & Hoyer, 2000), and for visual recognition more generally (Moghaddam, 2002; Balakrishnan, Hariharakrishnan, & Schonfeld, 2005). In addition, more general applications which would previously have used PCA have been able to utilise ICA as a higher-order version of this long-standing technique (Yang, Zhang, Frangi, & Yang, 2004).

In ICA, components are by definition assumed to be statistically independent, that is:-

$$\mathrm{E}\{g(x)h(y)\} = \mathrm{E}\{g(x)\}\mathrm{E}\{h(y)\} \tag{3.1}$$

This also means that components must be uncorrelated, since this is a weaker condition subsumed by independence, where h(x) and g(x) are simply identity functions:-

$$\mathrm{E}\{xy\} = \mathrm{E}\{x\}\mathrm{E}\{y\} \tag{3.2}$$

This can be a useful working assumption, although it is widely accepted that, where applied in practice, the underlying components are often not strictly independent, necessitating a degree of approximation in the technique, or else requiring specific workarounds (Bressan & Vitria, 2003). More generally, however, independence is often statistically not the optimal condition for a set of variables. In particular, negatively correlated variables have some properties which make them preferable to positively correlated and independent variables, especially in cases of noisy systems where negative correlation can help reduce the noise (Feng & Tirozzi, 2000) and increase the storage capacity (space filling).

3.1.2 Benefits of Negative Correlation

Two specific benefits of negative correlation have been described in section 1.2. The first is that negatively correlated Gaussian noise will tend to reduce to zero more rapidly than independent and positively correlated Gaussian noise as the number of instances of this noise increases. For noisy image representation, the utility of this result was shown in figure 1.5, where Gaussian noise is added to a number of replications of the same image. The second benefit of negatively correlated variables is their ability to fill a space better than positively correlated or independent variables, because of their tendency to push each other away. This was shown in figure 1.5. The space filling benefit is only apparent when the independent components (coefficients of the basis functions) are restricted to positive values.

The need for the non-negative coefficient restriction for the space filling benefit arises because where basis functions are negatively correlated with each other, by definition the most negatively correlated function for basis function **a**, will be simply -**a**. It is clear from this that if coefficients to basis function **a** are allowed to take negative values, then there is no meaningful distinction between **a** and -**a** in the model, which effectively means that whenever **a** is present, -**a** is also implied as present. This means that a set of independent components will best cover the space under these circumstances, with their implied negatively correlated components also being present; actual positively or negatively correlated basis functions under these circumstances will be to some extent redundant and suboptimal. However, where only non-negative coefficients are allowed, basis function **a** no longer implies -**a** as well, meaning that there is now a benefit to having real negatively correlated basis functions, as the implies ones are no longer present. This non-negative coefficient restriction is increasingly popular in more recent work (Lee & Seung, 2001; Hoyer, 2002, 2003) for other, principled, reasons, such as the fact that natural quantities cannot be negative, images cannot consist of negative amounts of constituent objects, neural firing rates cannot be negative etc., and so the non-negative constraint should not be regarded as a weakness of this approach.

3.1.3 Two different approaches for negative correlation and ICA

We saw earlier that the basic ICA model, $\mathbf{X} = \mathbf{AS}$, has two quite different sets of variables to estimate: the components themselves, which form the matrix $\mathbf{S}$, and the basis functions, which together form the mixing matrix **A**. Both of these are candidates for negative correlation, and as such two complementary algorithms, correlated component analysis (CCA) and correlated basis analysis (CBA), have been developed and are presented in this chapter. Both of them are generalisations of ICA, which can be seen as special case of either algorithm.

The next section will outline the theoretical framework for CCA and CBA, show the basic steps in the implementation of it, and highlight the benefits of the specific approach taken here. Following that is a section containing some simple examples demonstrating CCA's ability to recover negatively correlated components, and then a section showing examples of how CBA takes full advantage of the benefits of negative correlation.

3.2 Algorithm

3.2.1 General Form

Both CCA and CBA use the same fundamental approach, which is to have an ICA core to find a set of components or basis functions, along with a Lagrangian constraint term to encourage those components or basis functions to be negatively correlated.

We therefore start with the basic ICA model, as given previous in section 2.4:-

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{3.3}$$

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{3.4}$$

($\mathbf{x}$ are the mixed components, $\mathbf{A}$ is the mixing matrix, $\mathbf{S}$ are the original source components, $\mathbf{y}$ are the recovered source components, $\mathbf{W}$ is the demixing matrix.)

As stated above, we can find negatively correlated components or basis functions by maximising independence, as under ICA, with an additional constraint to minimise the correlations (maximise the negative of the correlations) of the recovered components or basis functions using the technique of Lagrange multipliers.

The difference between the marginal distributions $f_{y_i}(y_i, \mathbf{W})$ and the joint distribution $f_{\mathbf{y}}(\mathbf{y}, \mathbf{W})$ of the independent components $\mathbf{y}$, can be expressed as the difference between the marginal differential entropies $\sum_i^m H(y_i)$ and the joint differential entropy $H(\mathbf{y})$ of these components. This in turn can be given by the Kullback-Leibler (K-L) divergence:-

$$\begin{aligned} D_{f\|\tilde{f}}(\mathbf{W}) &= \sum_i^m H(y_i) - H(\mathbf{y}) && (3.5)\\ H(\mathbf{y}) &= H(\mathbf{Wx}) = H(\mathbf{x}) + \log \mid \det(\mathbf{W}) \mid && (3.6)\\ \Rightarrow D_{f\|\tilde{f}}(\mathbf{W}) &= \sum_i^m H(y_i) - H(\mathbf{x}) - \log \mid \det(\mathbf{W}) \mid && (3.7) \end{aligned}$$

The correlation constraint term, including the different versions for the two different algorithms, will be outlined in the next section. For now, it will be represented by a lagrangian placeholder function $\phi(\mathbf{W})$, and the standard lagrangian coefficient, λ. This gives us the correlation constraint term:-

$$\lambda(\phi(\mathbf{W})) \tag{3.8}$$

This can be added to the K-L divergence to give a complete objective function to be minimized:-

$$D_{f\|\tilde{f}}(\mathbf{W}) = \sum_i^m H(y_i) - H(\mathbf{x}) - \log \mid \det(\mathbf{W}) \mid + \lambda(\phi(\mathbf{W})) \tag{3.9}$$

It is important to note that in an iterative algorithmic implementation (which is the case for almost all practical ICA algorithms), this function is equivalent to the following two-step procedure:

$$\bar{D}_{f\|\tilde{f}}(\mathbf{W}) = \sum_{i}^{m} H(y_i) - H(\mathbf{x}) - \log |\det(\mathbf{W})| \tag{3.10}$$

$$D_{f\|\tilde{f}}(\mathbf{W}) = \bar{D}_{f\|\tilde{f}}(\mathbf{W}) + \lambda(\phi(\mathbf{W})) \tag{3.11}$$

This means that the standard K-L divergence function can be calculated in the first step, and the negative correlation constraint term can be applied in the second step, without the optimization technique employed for both steps having to be the same. The result of this is that existing algorithms for the ICA core can be imported without any significant modification for the first step, and a simple gradient approach used to reduce the correlation between the derived components or basis functions for the second step.

The activation functions forming the update steps in an iterative algorithm for the two equations above can be formed by taking the gradient of the objective function with respect to the demixing matrix $\mathbf{W}$. For the various terms in the equations, this gradient is computed as follows:-

$\sum_{i}^{m} H(y_i)$ The marginal distributions are the most problematic, as the formation of the gradient requires a parametric estimation of the distributions. This can be achieved with reasonable accuracy using the Gram-Charlier expansion. However, ICA algorithms typically take advantage of a computationally much simpler approximation, where the objective function is simply given by an appropriate nonquadratic function. A popular specific choice is $\log(\cosh(\mathbf{Wx}))$, which yields $\mathbf{x}\tanh(\mathbf{Wx})$ as the derivative term; more generally, the derivative can be given as $\mathbf{x}\varphi(\mathbf{Wx})$.

$H(\mathbf{x})$ The first of the two terms which together make up the joint distribution is a function only of the mixture variables $\mathbf{x}$, which means that this term is a constant, not dependent on $\mathbf{W}$. It thus drops out of the gradient altogether.

$\log \mid \det(\mathbf{W}) \mid$ The second of the joint distribution terms clearly is dependent on $\mathbf{W}$. Some fixed-point ICA algorithms also drop this term, by pre-whitening the data (thus assuming zero correlation), which results in this term also being a constant. However, this is clearly not appropriate when the components are encouraged to have non-zero correlation values, and so the gradient of this term must be included. This is given by $\mathbf{W}^{-T}$ (the inverse transpose of the demixing matrix).

$\lambda(\phi(\mathbf{W})$ The abstract form of the correlation constraint term has a similarly abstract gradient: $\lambda\frac{d\phi(\mathbf{W})}{d\mathbf{W}}$. The detailed form of these functions is outlined in the next section.

Putting these gradient terms together, we have the complete gradient activation functions to be used in the iterative algorithm:-

$$\frac{d\bar{D}_{f\|\tilde{f}}(\mathbf{W})}{d\mathbf{W}} = \mathbf{x}\varphi(\mathbf{W}\mathbf{x}) - \mathbf{W}^{-T} \tag{3.12}$$

$$\frac{dD_{f\|\tilde{f}}(\mathbf{W})}{d\mathbf{W}} = \frac{d\bar{D}_{f\|\tilde{f}}(\mathbf{W})}{d\mathbf{W}} + \lambda\frac{d\phi(\mathbf{W})}{d\mathbf{W}} \tag{3.13}$$

This finally gives us iterative update steps for estimating the demixing matrix $\mathbf{W}$ based on maximising the negative gradient (also including here a learning rate η, as required by many gradient algorithms):-

$$\Delta\bar{\mathbf{W}} = \eta[\mathbf{W}^{-T} - \mathbf{x}\varphi(\mathbf{W}\mathbf{x})] \tag{3.14}$$

$$\Delta\mathbf{W} = \Delta\bar{\mathbf{W}} - \lambda\frac{d\phi(\mathbf{W})}{d\mathbf{W}} \tag{3.15}$$

These provide the central weight update steps in the most general form. Specific implementation involved the use of a chosen existing ICA technique for the first update step; several have been tested for use with the algorithms presented here, including a simple generic gradient method developed for testing these algorithms, the Bell-Sejnowski algorithm (Bell & Sejnowski, 1995), Amari's natural gradient version of the Bell-Sejnowski algorithm (Amari, 1999), and Hyvärinen's FastICA algorithm (Hyvärinen & Oja, 1997), with the important caveat that the orthogonalization step in a whitened domain must be removed (in order to allow components to be correlated at all), when the implementation and testing of the algorithms is described in more detail. It should be emphasised that the particular choice of ICA algorithm is unimportant with regard to our new contribution, the negative correlation aspect. We chose these particular algorithms because they remain widely known, and represent three quite distinct approaches. In all of our demonstrations, however, the comparison is with the ICA algorithm alone (the control condition), and the same algorithm with our additional negatively correlation term, so the particular performance of a given ICA algorithm itself does not in any way bias the demonstrations in favour of our model.

Specific implementation of the second update step involves a choice of negative correlation constraint function $F(\mathbf{W})$, and a method for optimizing this function with respect to $\mathbf{W}$. This is the subject of the next section.

3.2.2 Correlation Constraint Function

In to gain the benefits of negative correlation, we need to ensure that the estimated components or basis functions, which the ICA core aims to make independent (and therefore uncorrelated), are negatively correlated. This combination of ICA core and correlation constraint are effectively aiming to make the components or basis functions as unrelated as possible in higher order statistics, while attaining a particular value for second order statistics. We discuss three possible approaches to adding a correlation constraint, in each case consisting of a coefficient element of some sort, and a correlation function whose gradient controls the direction for the updated component or basis function estimate such that the correlation moves in the desired direction. In the following we assume for simplicity of notation that all variables have been normalised to zero mean, which is a standard preprocessing step in most ICA techniques.

Given a random vector **a**, we first define a matrix-valued function $F(\mathbf{a})$ which takes a vector **a** of N components or basis functions, and returns the correlation matrix of these:-

$$F(\mathbf{a}) = E\{\mathbf{a}\mathbf{a}^T\} = \begin{bmatrix} E\{a_1a_1\} & E\{a_1a_2\} & \dots & E\{a_1a_i\} \\ E\{a_2a_1\} & E\{a_2a_2\} & \dots & E\{a_2a_i\} \\ \vdots & \vdots & \ddots & \vdots \\ E\{a_ia_1\} & E\{a_ia_2\} & \dots & E\{a_ia_i\} \end{bmatrix} \tag{3.16}$$

$$[F(\mathbf{a})]_{i,j} = E\{a_ia_j\} \tag{3.17}$$

The vector derivative of $F(\mathbf{a})$ gives a third order tensor, with the (i,j,k)th element representing the partial derivative of the correlation between the ith and jth variables with respect to the kth variable:-

$$\nabla F(\mathbf{a}) = \frac{dF(\mathbf{a})}{\mathbf{a}} \tag{3.18}$$

$$[\nabla F(\mathbf{a})]_{i,j,k} = \frac{\partial E\{a_ia_j\}}{a_k} \tag{3.19}$$

$$\tag{3.20}$$

The simplest case is to have a correlation function that gives a scalar measure of the overall correlation (with the vector-valued gradient of this providing the direction for updating the estimate of **a**), and a free parameter to control the direction and extent of this influence relative to the ICA term. The scalar correlation measure is obtained by summing the elements of the correlation matrix; the gradient is therefore simply obtained by summing $\nabla F(\mathbf{a})$ over the first two dimensions, where it can be shown that

$$[\psi(\mathbf{a})]_k = \sum_{i=1}^{N}\sum_{j=1}^{N}[\nabla F(\mathbf{a})]_{i,j,k} = \sum_{i=1}^{N} 2a_i \tag{3.21}$$

$$\tag{3.22}$$

In this case, the correlation constraint term in the estimate update is simply $\lambda\psi(\mathbf{a})$, where λ is a coefficient that represents the free parameter as discussed above.

A second possibility is to specify a mean correlation target value c_t. It is possible that the mean correlation of the expected underlying components in CCA may be known. Alternatively, we may have in mind a particular mean correlation value for the basis functions in CBA (in particular, we may want them to have a certain negative value, given the benefits of negatively correlated basis functions in data representation). In these circumstances, it is clearly desirable for the algorithm to be able to yield components (CCA) or basis functions (CBA) with a specified correlation level. In this case, the scalar-valued correlation function remains as defined in eq.3.22, but the coefficient is

$$\lambda = c_t - c_a \tag{3.23}$$

$$c_a = \frac{1}{N^2 - N} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} E\{a_i a_j\} \tag{3.24}$$

$$\tag{3.25}$$

λ gives the difference (including direction) between the target mean correlation and the current mean correlation (we use by convention non-identical variables only here, removing the trivial self-correlations; hence zero is uncorrelated, a positive value is positively correlated and a negative value is negatively correlated), and acts as an adaptive learning rate.

It may be the case that we actually know something about the correlation structure of the components or basis functions that goes beyond simply knowing the mean correlation. As well as scalar-valued correlation measures, therefore, we should be able to specify a full target correlation matrix. In this case, the adaptive learning variable becomes a matrix of coefficients; $\lambda = \mathbf{C}_t - \mathbf{C}_a$ where $\mathbf{C}_t$ is the target correlation matrix and $\mathbf{C}_a$ is the correlation matrix of the current estimate of $\mathbf{a}$. It is no longer possible to sum over all the (i,j)th elements of the correlation function of eq.3.20, as was done for the scalar-valued correlation measure (eq.3.22), because the vector of partial derivatives for each pairwise correlation has a different coefficient, specified by the corresponding element of λ, and so are no longer identical. Instead, we have the following vector-valued estimate update function:-

$$
\begin{aligned}
[\psi(\mathbf{a})]_k &= \sum_{i=1}^{N}\sum_{j=1}^{N} \lambda^{(i,j)} \left[\nabla F(\mathbf{a})\right]_{i,j,k} && (3.26)\\
\lambda^{(i,j)} &= (\mathbf{C}_t^{(i,j)} - \mathbf{C}_a^{(i,j)}) && (3.27)
\end{aligned}
$$

The adaptive learning rate coefficient approach used in the second and third approaches described above also has the added benefit of making the algorithms highly stable with respect to the correlation constraint, as the size of changes to the demixing matrix related to the correlation constraint term are made in direct proportion to the distance from the target correlation, which acts as a global attractor. Our tests show show that the algorithms can reliably attain any possible scalar correlation target (that is, any mean correlation between 1 and -1/(N-1), where N is the number of components/bases), or any valid correlation matrix. It should also be noted that it is trivial to extend this approach to incorporate partially-specified correlation matrices by combining the scalar and matrix approaches, where the sums in eq.3.27 are limited to those specified components rather than the entire set.

This general gradient algorithm for reducing the correlation between a set of variables is used in both CBA and CCA. For CBA, the variables whose correlation is specified are the set of basis functions, which are the columns of the mixing matrix $\mathbf{A}$, which therefore means $\mathbf{A}^T$ gives the sampled variables in rows to be used in the above equations (as random vector $\mathbf{a}$). For CCA, the components which are the rows of $\mathbf{S}$ are the variables to be used.

One further step is required in order to employ this gradient approach in our CBA and CCA algorithms. These algorithms, as seen in the earlier equations, require the update steps to be for the separating matrix $\mathbf{W}$ (although in practice, some ICA algorithms use the mixing matrix $\mathbf{A}$ in their update step). We therefore need to be able to give the negative correlation step update matrix, expressed above in terms of either $\mathbf{A}^T$ for CBA or $\mathbf{S}$ for CCA, in terms of $\mathbf{W}$. To do this we can note the fact that the least-squares error inverse for a non-square matrix $\mathbf{A}^T$ is given by the Moore-Penrose pseudoinverse, $(\mathbf{A}^T)^+$. This therefore gives us the best estimate of $\mathbf{W}$ to be used directly in the update step, and has the added benefit of being simple and relatively efficient to calculate. Conversely, in order to first enter the $\mathbf{A}$ domain in order to calculate the gradient update step, the pseudoinverse can be used in the other direction, on the $\mathbf{W}$ separating matrix yielded by the first update step. We thus have a translation from $\mathbf{W}$ into $\mathbf{A}$ for the gradient update step, and then back again into $\mathbf{W}$ to yield the final updated $\mathbf{W}$ matrix for the current iterative pass. It should be noted that it is not possible to perform the gradient update directly in the $\mathbf{W}$ domain because minimising the correlation of $\mathbf{A}^T$ is equivalent to maximising the correlation of $\mathbf{W}$, which is unstable because the fixed point of perfect correlation does not invert to perfect negative correlation back in the $\mathbf{A}$ domain.

For the CCA algorithm, the update translation operations are slightly different. Given the $\mathbf{W}$ matrix from the first update step, the components $\mathbf{S}$ can be easily calculated by noting that $\mathbf{S} = \mathbf{WX}$. Once the negative correlation update has been calculated in the $\mathbf{S}$ domain, the conversion back to the $\mathbf{W}$ domain is given by another simple calculation: $\mathbf{W} = \mathbf{SX}^+$. It should be noted that where S is constrained to be non-negative, which is not the inherently the case for CCA as it is for CBA but may be adopted for some purposes nonetheless, the calculation of S is more complex, and typically found using a constrained optimization technique, which will generally be much slower than the methods given here.

3.2.3 Benefits of the Current Approach

By adopting a two-step update procedure, where the separating matrix is first calculated using an ICA update step, and then the resultant components or basis functions are made more negatively correlated using the new gradient step given in the previous section, there are a number of particular benefits:-

- The two update steps do not need to use the same, gradient-based optimization procedure. This is especially important as the negative correlation gradient algorithm is not stable in the $\mathbf{W}$ domain in which ICA update steps typically operate.
- By having the ICA update step separately, existing ICA update steps can be

used with almost no modification required. The existing algorithms do not even have to use a gradient optimization approach to be usable; multiplicative or quadratic programming algorithms can also be used. The only constraint on ICA algorithm is that it must not contain an orthogonalization step. This is obviously necessary in order to allow components to be anything other than uncorrelated.

- By translating into the **A** domain for CBA and the **S** domain for CCA where necessary, the ICA step can operate in either the **A** or the **W** domain, and still be compatible with these algorithms.
- Existing ICA algorithms do not need to estimate the components **S** in order to be used with these algorithms, although obviously ones that do are also compatible.
- By using a separate negative correlation update step, the effect of the negative correlation constraint term is both easy to assess, and easy to control through the strength of the parameter λ.
- The separate negative correlation update step ensures the stability of the algorithm, as the stability of existing ICA steps is not altered within the first update step, and the second update step also has guaranteed stability for a sufficiently low learning rate.

It can be seen that the current algorithms are an extension of, and in a real sense a generalization of, ICA, combining the benefits of existing ICA algorithms with the the benefits of negative correlation. The following two sections give some brief demonstrations of how these combined benefits allow these two new algorithms to outperform ICA.

3.3 Examples of CCA

The examples in this section show the CCA algorithm in operation. As CCA is designed to find negatively correlated components, the demonstrations here focus on its ability to accurately recover source signals that are negatively correlated. Its performance is contrasted with that of ICA on the same tasks.

3.3.1 Example 1: Basic performance

The first example (figure 3.1) clearly demonstrates the most important feature of CCA - its superior ability to recover the original, negatively correlated signals. While ICA has recovered signals that remain quite significantly mixed, and are not the original source signals, CCA has successfully recovered the original, negatively correlated source signals to a much greater extent. This can be quantified by measuring the standardized Euclidean distance between the original and respective recovered signals for both ICA and CCA. Because ICA and CCA are linear operations, they can only recover signals up to a scalar constant; as such, signals are each standardized to unit variance prior to the measurement, to allow a fair comparison. For the first signal, the ICA distance is 12.607, while the CCA is only 0.11358. For the second signal, the difference is even more extreme: 1976.2 for the ICA distance, while only 0.16399 for the CCA distance.

3.3.2 Example 2: ICA recovers original independent signals, CCA recovers negatively correlated source signals

The example here (figure 3.2) visibly demonstrates the difference between the independence goal of ICA and the negatively correlated components goal of CCA. The negatively correlated source signals are recovered by CCA, whilst ICA recovers independent signals. These results can be quantified in the same way as those in the previous example, through standardized Euclidean distance measurements. For the first signal, the ICA-recovered signal has a distance of 60.043 from the original, whilst the CCA-recovered signal is closer at 17.193. The difference in performance is greater for the second signal, where ICA yields a distance of 15.248 whilst CCA gives a signal just 0.83118 from the original.

The signals recovered by ICA are actually closely related to those from CCA, and can be explained in terms of the method used for generating the source signals. This

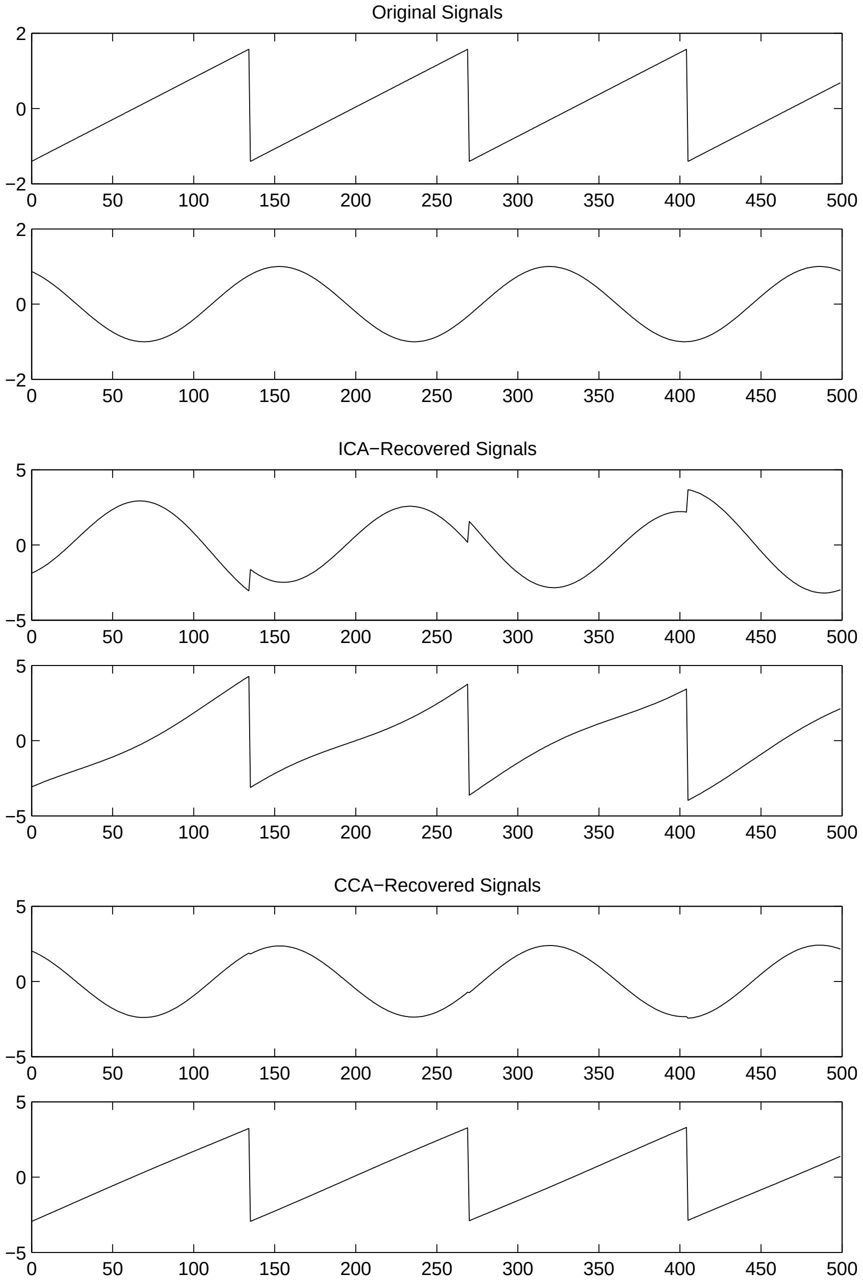


Figure 3.1: CCA vs ICA for signal recovery. This shows the central benefit of the CCA algorithm, which gives a better recovery of the original, negatively correlated signals than ICA (quantitative measure given in the text). The correlation of the ICA components was 0, whereas that of the CCA components was -0.34805, much closer to the original signals correlation of -0.35089.

was a standard technique of starting with independent source signals (such as a sine wave and a sawtooth function for the two-component example), and pre-mixing them with a negative correlation matrix to establish the original source signals for the algorithms to recover. After this, the pre-mixed source signals are mixed together with the mixing matrix to produce the mixed data. Because both mixing and pre-mixing are linear operations, they can in fact be described by just a single mixing operation, as though the original independent signals were mixed together just once to produce the mixed data. Because of this, it is not surprising that ICA finds this combined mixing matrix and original independent source signals. It is important to note that this does not at all invalidate this test; on the contrary, it points to a specific weakness in this ICA algorithm when faced with correlated signals (which it is not designed for). It is desirable however, also to test the algorithms without this pre-mixing stage leading to this phenomenon. This is addressed in the next section.

3.3.3 Two methods for generating negatively correlated test signals

The most common method for generating negatively correlated test signals is to first generate independent signals, and then to pre-mix them with a negative correlation matrix. An advantage of this method is that it allows easy and precise control of the correlation relationship between any number of components. However, it was seen in the previous example that under these circumstances, ICA will tend to recover the independent signals prior to pre-mixing, rather than negatively correlated source signals. An alternative method for creating negatively correlated signals without pre-mixing by a correlation matrix is to use phase control. By adjusting the relative phase of two periodic signals, their correlation can be altered. figure 3.3 shows two periodic signals along with a graph that shows how the correlation changes with phase shift. It is straightforward using this approach to set the correlation to a desired value, including a particular negative correlation, or alternatively simply to set the phase to the point of maximally negative correlation. The advantage of this approach is that the source signals remain in their original form, without being

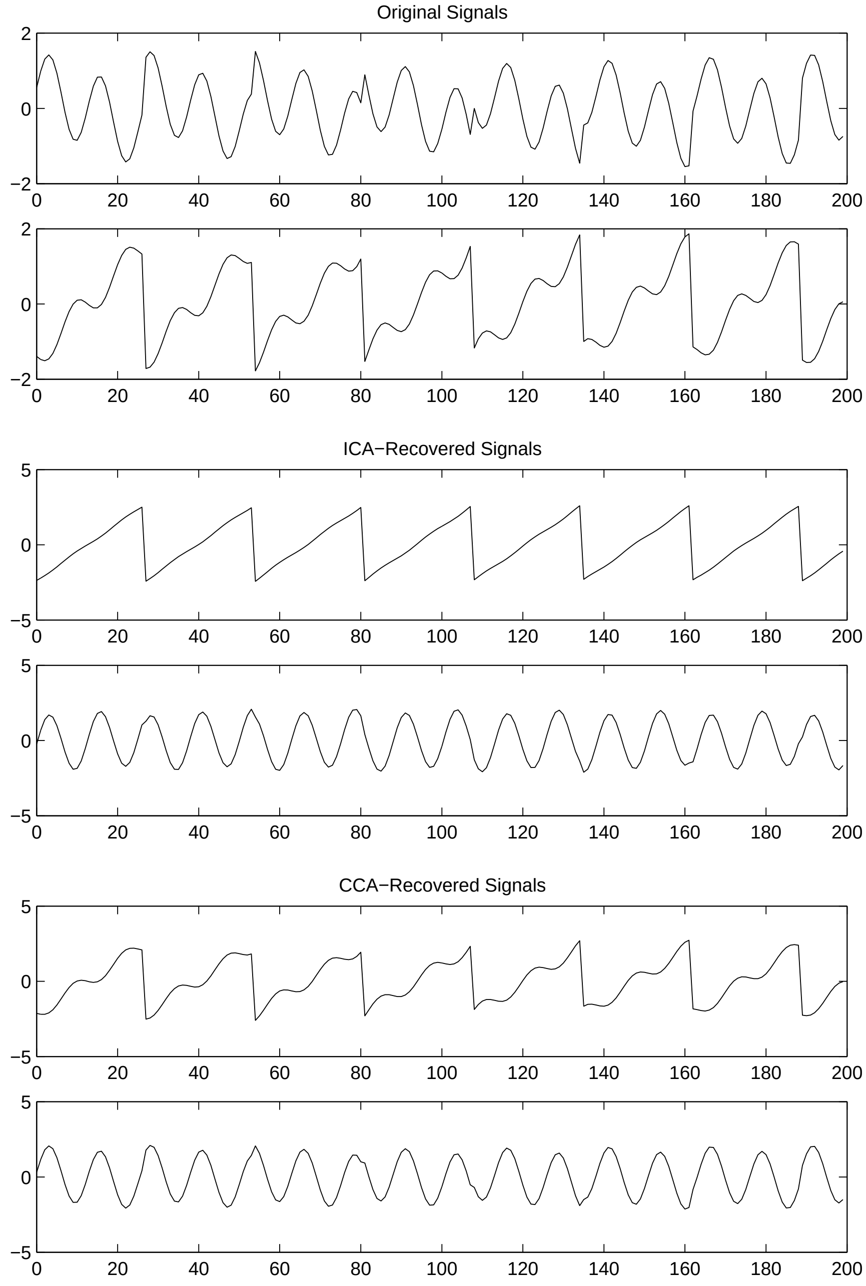


Figure 3.2: ICA recovering uncorrelated signals. Where the signals are pre-mixed to be negatively correlated, before the main mixing stage to produce the mixed data, ICA recovers the first, uncorrelated versions of the signals, whereas CCA recovers the desired, negatively correlated signals. The correlation of the ICA components was 0, whereas that of the CCA components was -0.4, the same as that of the original signals (-0.4).

pre-mixed. Clean signals with a negative correlation provide a useful way of further testing the CCA algorithm, and this is the method that is used in the remaining two examples.

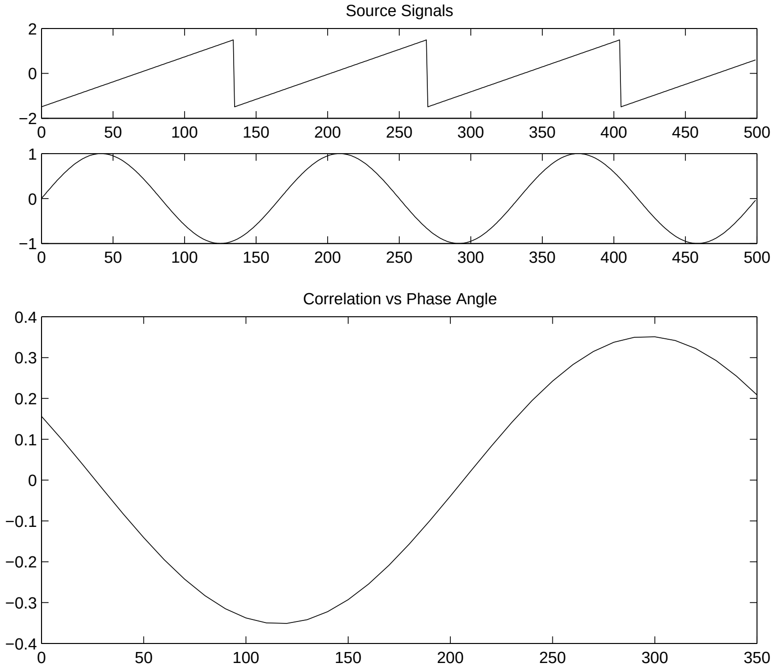


Figure 3.3: Correlation control using phase shift. Sinusoidal and saw-tooth source signals are shown, along with the correlation between these two signals as a function of the phase between them.

3.3.4 Example 3: ICA recovers independent ”mixtures”, CCA recovers negatively correlated clean signals

Using the technique of phase shift in generating the negatively correlated source signals, this example (figure 3.1, shown earlier) demonstrates the superior performance of CCA in recovering the original source signals. It is notable that ICA recovers signals that are statistically independent, but that do not take the precise shape of the original source signals. In finding independent rather than negatively correlated signals, ICA is forced to find slight mixtures of the original signals, rather than the pure signals themselves.

3.3.5 Example 4: Assessing the correlation constraint coefficient (λ)

The final example in this section looks at the role of the correlation constraint coefficient (λ) in the absence of a correlation target. In the earlier description, it was shown that the algorithm is capable of yielding components with a desired correlation value through the use of an adaptive learning rate. This also increases the stability of the algorithm. In order to evaluate the behaviour of the correlation constraint term, however, it is useful to instead adopt the approach that treats the coefficient as a free parameter, without any specified correlation target.

The value of the coefficient was systematically varied whilst the other experimental parameters (learning rate, epochs etc.) remained constant. It can be seen in figure 3.4 that the correlation of the derived components changes smoothly with the value of λ, which shows both the stability of the algorithm under changes to this value, and demonstrates that the negative correlation step offers a way to systematically control the correlation of the components found by CCA (including even making them positively correlated if so desired).

3.4 Examples of CBA

The CCA algorithm has been shown to be effective in recovering components that are negatively correlated. The CBA algorithm has a complementary purpose, which is to utilize the noise-reduction and space filling benefits of negative correlation. It was seen in earlier sections how negatively correlated basis functions could offer a theoretical advantage over positively correlated and independent basis functions in representing data with non-negative coefficients. This section contains two practical examples of this advantage in operation, inspired by the widespread use of ICA on natural image processing.

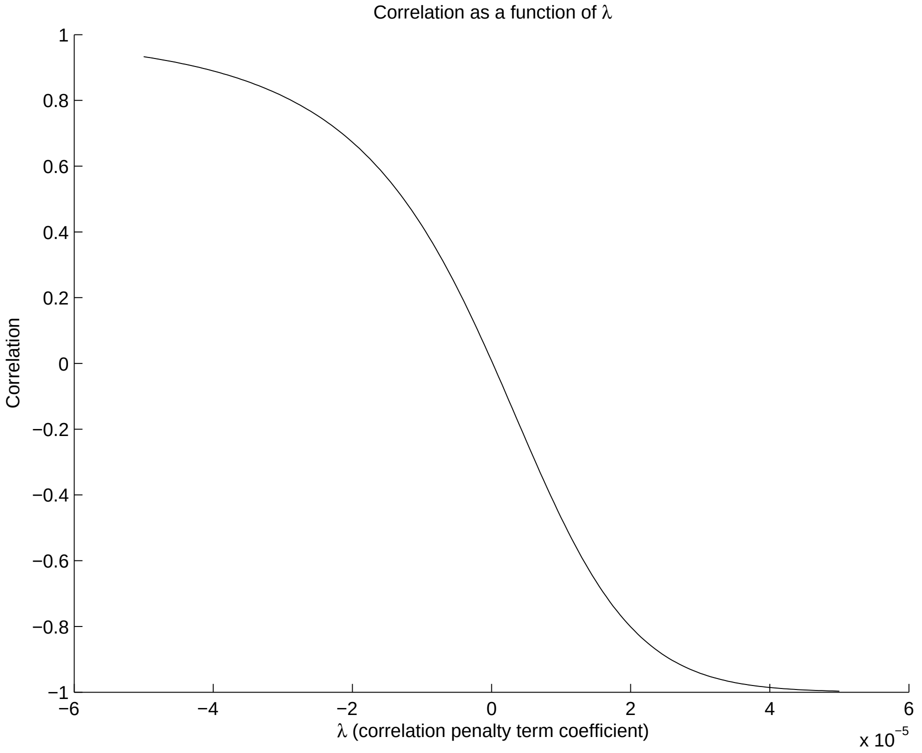


Figure 3.4: Correlation as a function of the constraint coefficient λ. The sigmoid curve in this graph highlights the robust and stable nature of the CCA and CBA algorithms, with correlation varying smoothly with the strength of the negative correlation constraint.

3.4.1 Example 1: A pre-whitened natural image

It can be seen that the original image has been preprocessed with a low-pass whitening filter. This image is actually one that has been used in examples of ICA, where such filtering is common to assist the ICA algorithm in useful basis functions. In order to give a fair trial to ICA, this pre-whitened image is used in the test here. Three different conditions were tested: positive correlation (where λ was given a negative value), independent (ICA) and negative correlation (CBA, where λ was given a positive value). figure 3.4.1 shows the basis functions found in the three conditions. It is immediately apparent that the positive correlation condition has obtained perfectly correlated basis functions, which is catastrophic for representing data points, as it is equivalent to only having one basis function. The independent and negatively correlated conditions have found ten different basis functions. The correlation values are given for these, which show that the algorithm has indeed found positively correlated, uncorrelated, and negatively correlated basis functions respectively.

Figure 3.4.1 also shows the image reproduced by representing each 3x3 image patch as a non-negative linear combination of the basis functions for each of the three conditions, and placed in its appropriate position in the overall image. This technique allows an immediate evaluation of the performance of the algorithms. It is clear that the positively correlated basis functions have allowed only a very poor representation of the image, not surprising in view of the perfect correlation between the basis functions. More significantly, however, the independent basis functions have also resulted in a rather noisy image reproduction, suggesting that they are suboptimal for this task. Only the negatively correlated basis functions allow for a perfect reproduction. The reproduction error values are given for all three conditions, corroborating the visual evidence.

3.4.2 Example 2: An unpreprocessed natural image

Whilst ICA algorithms prefer the data, in this case a natural image, to be preprocessed, in particular pre-whitened, it is worth investigating whether or not the CBA algorithm performs any worse on an image which has not been preprocessed at all.

This example follows the same procedure as the previous one, with positively correlated, independent, and negatively correlated conditions. Figure 3.6 shows that once again, when positive correlation is encouraged, perfectly correlated basis functions are found, whereas the uncorrelated and negatively correlated conditions find ten different basis functions.

The image reproductions in figure 3.6 also follow the pattern of the previous example, with the positively correlated basis functions allowing the worst image reproduction, followed by the independent basis functions which still give a very noisy reproduction, and then the negatively correlated basis functions which give a perfect, noise-free, reproduction. It can be seen from the error values as well that the lack of preprocessing of the image did not damage the performance of the algorithm at all, in contrast to that of the ICA algorithm, whose relative performance here was worse than in the previous example.

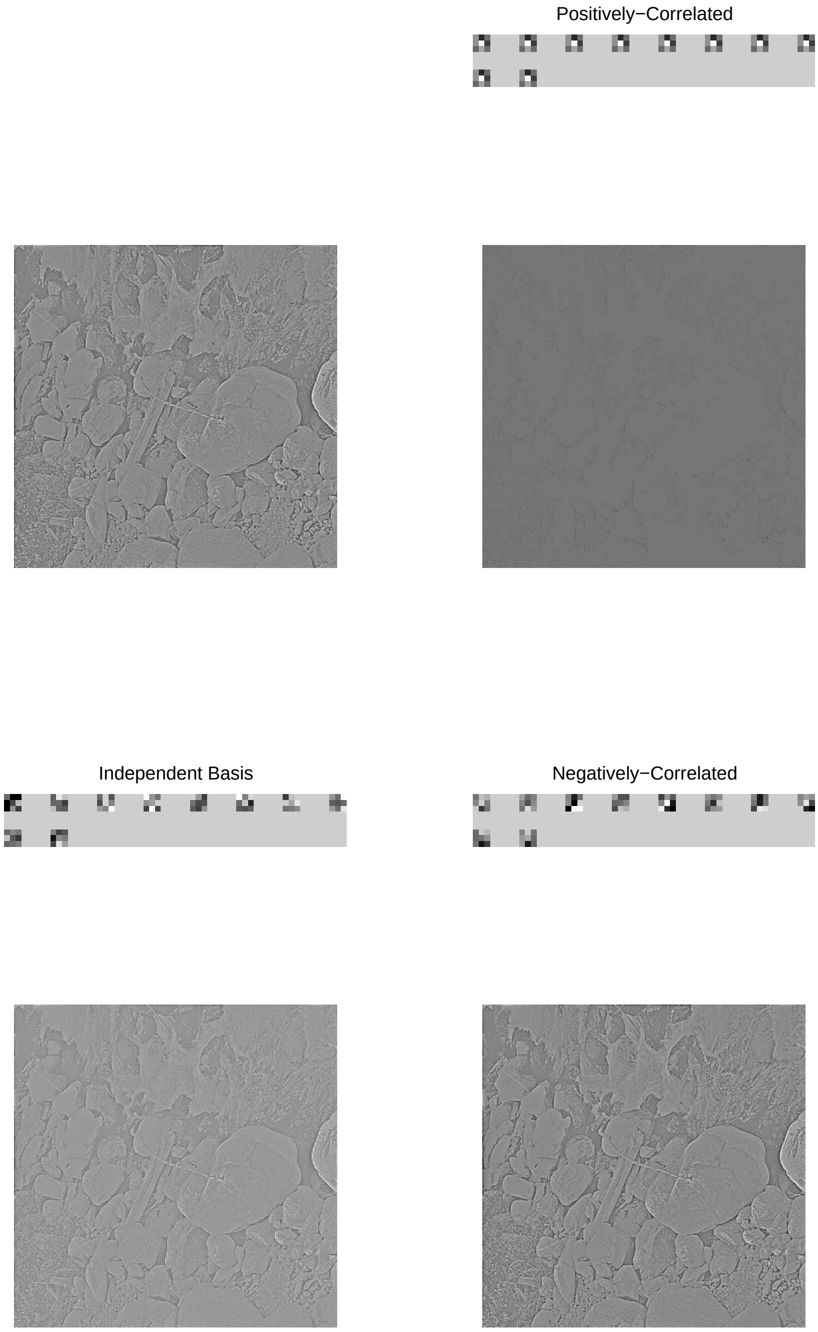


Figure 3.5: Basis functions and image representation. This figure shows the positively correlated, independent, and negatively correlated basis functions recovered by using the CBA algorithm (with a negative constraint coefficient to obtain positively correlated basis functions and a zero coefficient to obtain the independent basis functions, which is therefore ICA in effect). Image data has also been represented using these basis functions, and the clear benefit of negative correlation is apparent. The correlation values for positively correlated, independent and negatively correlated basis functions respectively are 1 (the maximum), -0.03666 (close to uncorrelated), and -0.10717 (near to the lowest possible of -0.11111. The respective image representation LSE values are 1.6625, 0.36347 and 0.

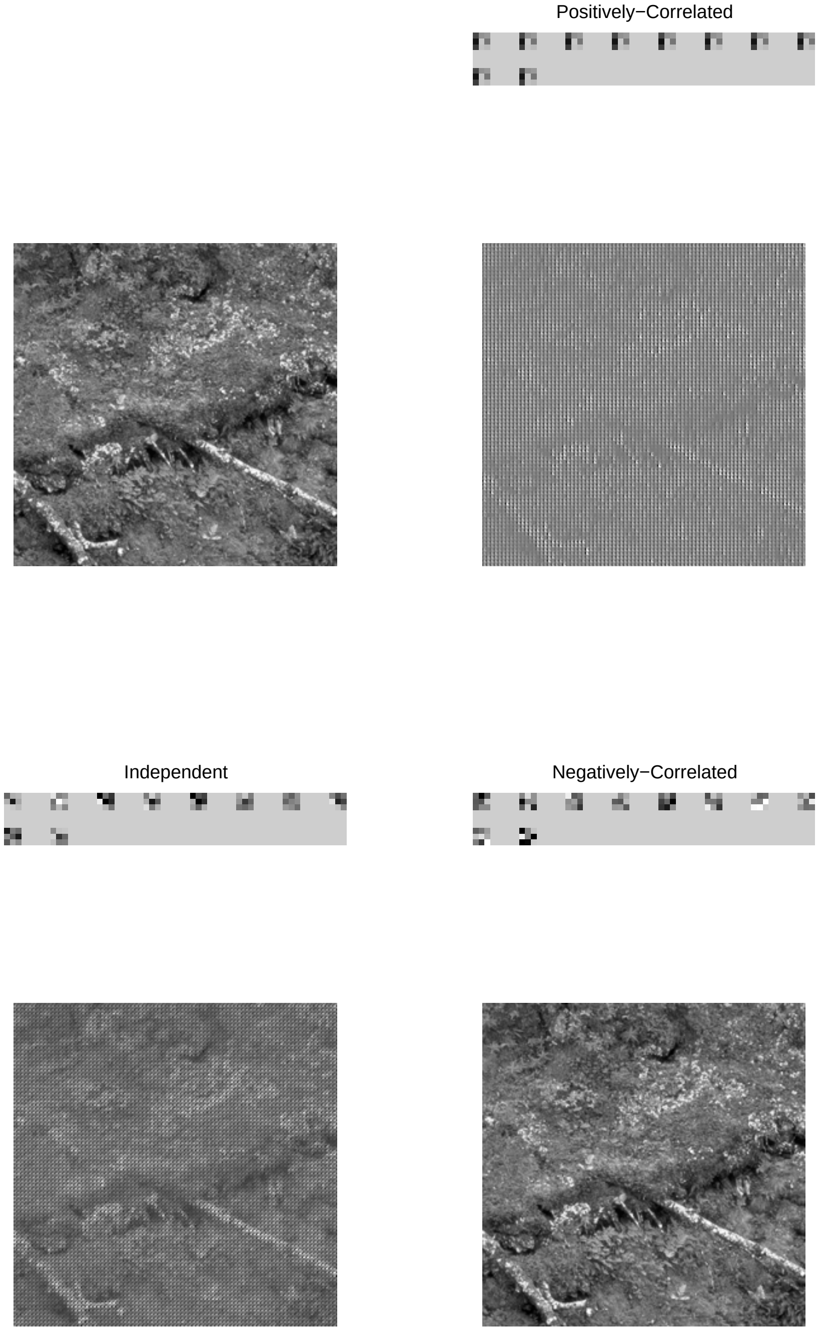


Figure 3.6: Basis functions and image representation for non-preprocessed image data. Similar to figure 8, except for the fact that in this case the image data has not been subject to any preprocessing. Again, negative correlation offers by far the best basis functions for representing the image data. The correlation values for positively correlated, independent and negatively correlated basis functions respectively are 1 (the maximum), 0.20188 (slightly positively correlated), and -0.098868 (near to the lowest possible of -0.11111. The respective image representation LSE values are 1.7583, 1.1481 and 0.

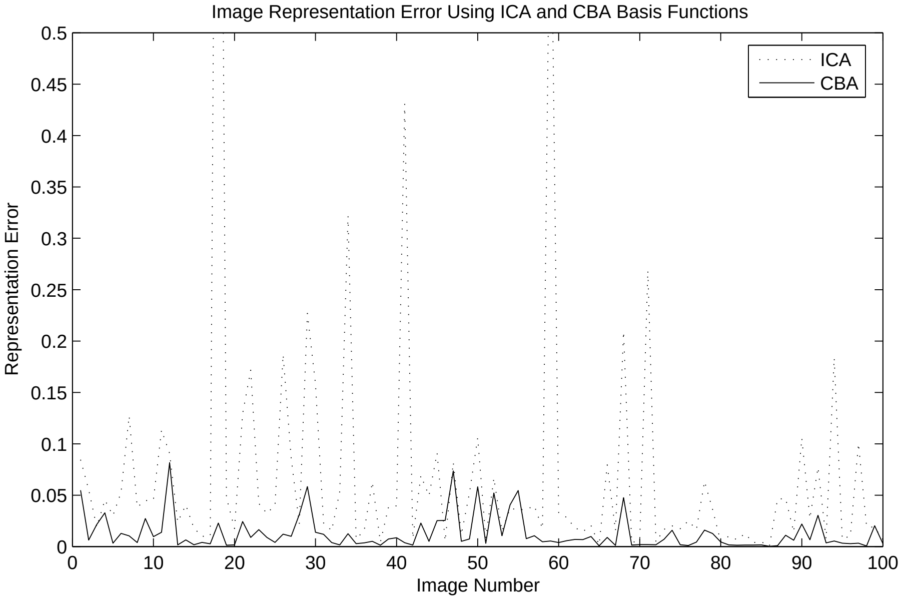
3.4.3 Example 3: Performance across a set of unpreprocessed natural images

In order to demonstrate the reliability of the algorithm, it was tested against ICA across a set of 100 images taken from the van Hateren natural stimuli collection (Hateren & Schaaf, 1998). The results are shown in figure 3.7. As expected, the correlation of the basis functions for CBA is much lower than that of ICA; the target was to have as low a correlation as possible. It can also be seen that the representation error (the error between the reproduced image when represented using a particular set of basis functions with non-negative coefficients) is much lower for CBA than ICA; this result is very statistically significant ($p<0.0002$), even for this relatively small sample. The performance is also more reliable than the ICA performance, where the error variance is much greater. These results further confirm the usefulness of negatively correlated basis functions, and hence the CBA algorithm which can obtain them, in representing data where natural (non-negative) quantities are required for the coefficients (the actual components).

3.5 Conclusions

Negative correlation has several benefits which can result in systems with lower noise, or more accurate representation of information with a limited set of resources. In particular, it has been shown that negatively correlated noise is reduced in accordance with the central limit theorem much more effectively than independent or positively correlated noise. It has also been shown that negatively correlated basis functions allow a more accurate representation of a set of data with non-negative coefficients than the same number of independent or positively correlated bases.

In this chapter, we have outlined two algorithms to exploit these statistical benefits of negative correlation, both of which are developments of the relatively new ICA approach. CCA finds components which are negatively correlated, whilst CBA finds negatively correlated basis functions. Both algorithms are based on an ICA core with a Lagrangian constraint term encouraging negative correlation, but the algorithms make use of a number of special techniques in order to allow the con-



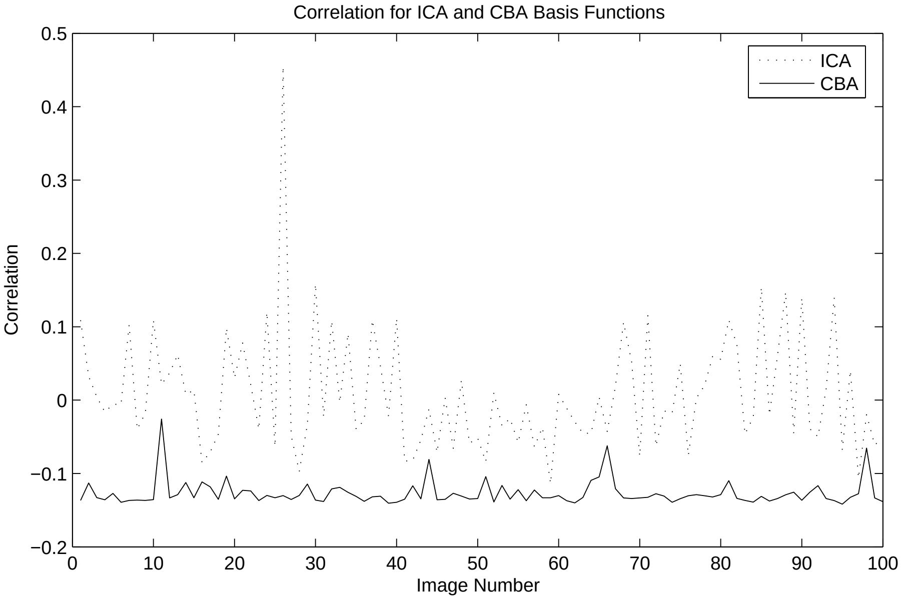


Figure 3.7: Image representation using basis functions obtained from ICA and CBA respectively. It can be seen that representation error is much lower when using the CBA basis functions; the mean error for ICA is 0.073078 while that for CBA is much lower at 0.012704; the difference is highly significant ($p<0.0002$). The mean correlation for ICA is around zero as expected (0.0080254), while CBA finds basis functions which are almost as negatively correlated as possible (-0.12701), a difference which is again very highly significant ($p\approx 0$).

straint term to be applied separately, and in a different domain, to the main ICA update step. A number of advantages to this have been outlined, emphasizing in particular the compatibility of these new algorithms with a wide variety of existing ICA approaches, as well as their relative efficiency and stability.

Several simple demonstration examples of CCA and CBA have been given here, each chosen to demonstrate a particular feature of the algorithms. These examples show that:-

- CCA offers superior performance to ICA in recovering negatively correlated signals.
- ICA recovers uncorrelated versions of the signals, whilst CCA recovers the actual negatively correlated signals.
- When clean, negatively correlated source signals are generated using a phase-shift technique, ICA tends to recover uncorrelated mixtures of these, whereas CCA recovers the negatively correlated clean original signals.
- CBA gives basis functions which allow more accurate representation of data (image data in the examples given here), allowing better recovery of that data, than ICA.
- CBA appears to be less demanding in terms of required preprocessing of data for than ICA.
- For both algorithms, correlation of components/basis functions varies smoothly as a function of λ, the negative correlation constraint coefficient (shown as a CCA example in this chapter, but equally valid for CBA also).

The examples presented in this chapter are just small demonstrations of what CCA and CBA can do. In particular, although the CBA examples were in this case given for image reproduction, it is important to note that there is nothing special about image data in this regard, and the result is equally applicable to any data whatsoever, including data in variables that are not themselves negatively correlated.

When non-negative coefficients are used, negative correlated basis functions will always be on average at least as effective as independent basis functions, and usually more so, at representing any set of data whatsoever.

There are a number of possible further developments for the algorithms presented here. One possibility is to explore the effects of negatively-related higher-order moments, particularly in view of the higher-order, non-Gaussian nature of ICA which is an important part of these algorithms. Whether or not the same benefits, perhaps to an ever greater extent, could exist for negative higher-order moments remains to be seen.

Another as-of-yet unexploited potential advantage of the CBA algorithm also requires further development. This concerns the space filling benefit of negative correlation that was outlined in section 1.2.1. It can be shown that at present, the advantage conferred by the space filling property of negatively correlated basis functions is actually the result not of space filling per se, but of the increased probability that the basis functions will surround the mean of the data, which therefore allows a more accurate non-negative coefficient representation. When all the basis functions lie in a similar direction from the data mean, as is more likely to happen with positively correlated and uncorrelated basis functions because they are more closely tied together, this will result in the suboptimal representation that is seen in the examples. What this means is that the actual space filling itself, which results in negatively correlated basis functions being on average closer to the data points they are representing and hence require on average lower coefficient values, is not yet being exploited by the algorithm. In fact, in systems where resources (which means coefficient values) are costly (including biological systems), this space filling benefit is likely to be important. For example, in neural systems it may result in lower firing rates being needed because individual neurons may be more accurately attuned to individual stimuli. This intriguing idea requires further investigation.

Also related to biological systems, the notion of how negative correlation is actually implemented in such systems is another subject for research. For example, data from the olfactory bulb suggests that neural firing is negatively correlated (Nicol, Feng,

& Kendrick, (in revision)). Whilst this result may be seen as supporting the above hypothesis that natural systems will exploit the benefits of negative correlation, it also raises the question as to what neural mechanisms can give rise to it. The issue is the subject of the next two chapters.

Chapter 4

Suprathreshold Stochastic Resonance in Neural Processing Tuned by Correlation

The benefits of negative correlation, based on accelerated central limit convergence and space filling, described in section 1.2, have been seen in the previous chapter to be present in algorithms that are concerned with image decomposition and representation. These suggest that neural activation in the early visual system may benefit from being negatively correlated and raises the question of just how the benefits of negative correlation arise in the brain. In this chapter we explore suprathreshold stochastic resonance, which offers one possibility for this mechanism at a neural level.

4.1 Introduction

One of the more striking facts in research into neural processing is that neurons *in vitro* fire with considerable regularity in response to a constant stimulus, while neurons *in vivo* exhibit much greater irregularity in response to the same stimulus (Mainen & Sejnowski, 1995). This irregularity has generally characterised as noise without this necessarily implying a lack of functional utility. A number of possible sources for neural noise *in vivo* have been proposed, including intrinsic channel noise (White, Rubinstein, & Kay, 2000) and Johnson electrical noise (Manwani & Koch, 1999), but the source widely regarded as the most important, especially in view of the clear difference between the *in vivo* and *in vitro* cases, is network noise. This argues that noise can arise from the pattern of spiking inputs arriving at synapses to a given neuron, which itself may arise due to the pre-synaptic neurons themselves having irregular firing patterns, or by their having a particular pattern of connectivity (Shadlen & Newsome, 1998; Kistler & De Zeeuw, 2002). The presence of irregularity has led to neural spike trains being treated as stochastic processes, and in particular Poisson processes. We will use such an approach ourselves in section 4.3.

Given the prominence of neural noise *in vivo*, it is natural to question what the functional role of such noise might be. Although the debate about temporal and rate coding continues (Gautrais & Thorpe, 1998), it is the case that rate coding tends to dominate in computational neuroscience research. From this perspective, noise would seem to be detrimental to neural processing by creating local fluctuations of the rate code, centred around the "true" (signal) value. However, one well-documented functional role of noise in rate coding is stochastic resonance (Gammaitoni et al., 1998; Longtin, 1993), in which noise can help to expose the temporal structure in a predominantly subthreshold signal. Although this has been found in some specific situations in neural coding (e.g. (Levin & Miller, 1996)), in general it suffers from the problem that neural systems are adaptive in a number of ways, including a limited ability to independently vary firing thresholds, and so an effect which relies on thresholds set considerable above a signal DC level is inevitably limited in scope.

Recently, an important new form of stochastic resonance has been discovered which does not require subthreshold signals, but instead operates in a quite different way in the context of a population of simple processing elements. Termed suprathreshold stochastic resonance (SSR), it it is a phenomenon whereby some noise in the system actually aids information processing rather than hinders it. Unlike traditional stochastic resonance, however, the system does not need to be subthreshold, i.e. it does not need to be fundamentally mistuned in that way. Instead, it exploits the fact that a set of processing units with similar or identical response properties are to some extent redundant, and this limits the possible variability of their response. If the input is highly varied, it is possible that the aggregated output of a set of similar units will not be able to capture the full variability of that input, because the number of possible output states is relatively limited. In that situation, noise that applies to the input to each unit separately effectively acts as a way to give greater potential variability in the output, with a consequent increase in the mutual information. The conditions for SSR to be possible are simply that the response is aggregated across a set of processing units with at least partially similar/overlapping outputs which are individually limited in their response capabilities (most obviously a threshold unit, but not necessarily). This suggests a functional role for noise in neural information processing, and one that is likely to be much wider in scope than traditional stochastic resonance. This has been demonstrated in simple, discrete time, models of threshold units, the specific context of the bifurcation point of neurons modelled by the Fitzhugh-Nagumo equations (Stocks, 2000a, 2000b, 2001a, 2001b; Stocks & Mannella, 2001), and for the more common models of integrate-and-fire neurons and Hodgkin-Huxley neurons (Hoch et al., 2003b, 2003a). In this chapter, we present a demonstration for integrate-and-fire neurons when presented with inputs modelled by Poisson processes (the most common paradigm). In particular, we demonstrate that improved information processing shown in SSR may be achieved in neural systems by tuning the correlations in neuronal firing, in a way that conforms to known biophysical properties.

4.2 Simple Threshold Model

We begin with a replication and extension of the basic SSR model outlined in (Stocks, 2000a, 2001a), in order to both verify the performance of the model and highlight one particular aspect not made explicit in the existing literature; most of this section closely follows the analysis given in (Stocks, 2000a, 2001a). The model consists of N independent threshold units (Heaviside functions), which receive a noisy input at each time point, and may possibly "fire" in response, depending on whether or not they reach the firing threshold. The input to the system consists of a signal component given by a Gaussian random variable $x(t)$ with mean μ_x and variance σ_x, which is the same for all units at a given time point, and a random Gaussian noise component $\eta_i(t)$ with zero mean and variance σ_η, which is independent for each unit. The output $y(t)$ is simply the number of units, n, that have reached their firing threshold, θ_i. Here we consider only the case where all units have the same firing threshold, which is set to the mean input value (the signal DC component). Hence we have:-

$$\begin{aligned}
y_i(t) &= \begin{cases} 1 \text{ if } x(t) + \eta_i(t) > \theta_i \\ 0 \text{ if } x(t) + \eta_i(t) < \theta_i \end{cases} \\
y(t) &= \sum_i y_i(t) \\
\theta_i &= \theta = \mu_x \text{ for all } i \\
P_{0|x} &= \int_{-\infty}^{\theta - x} P_\eta(\eta) d\eta \\
P_{1|x} &= \int_{\theta - x}^{\infty} P_\eta(\eta) d\eta \\
P_\eta(\eta) &= \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp \frac{-\eta^2}{2\sigma_\eta^2} \\
P_x(x) &= \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp \frac{-(x - \mu_x)^2}{2\sigma_x^2}
\end{aligned} \tag{4.1}$$

In keeping with (Stocks, 2000a, 2001a), our performance measurement is given by the mutual information of the system, which characterises the information processing capabilities of the system. This is a probability-based approach which essentially measures the extent to which the inputs and outputs can be predicted from each other. In general, noisier inputs will lead to lower mutual information all other things being equal, but if the information processing capability of the system is actually enhanced by the presence of a certain level of noise, then we would expect to see the mutual information increase in these circumstances. Hence mutual information is a very appropriate way to test for SSR. Recalling the material from section 2.2.2, the mutual information is given by:-

$$\begin{aligned}
MI &= H_y - H_{noise} \\
H_y &= -\sum_{n=0}^{N} P_y(n) \log_2 P_y(n) \\
H_{noise} &= -\int_{-\infty}^{\infty} P_x(x) dx \sum_{n=0}^{N} P_y(n|x) \log_2 P_y(n|x)
\end{aligned} \tag{4.2}$$

Here, H_y is the entropy of the output, and H_{noise} gives the noise entropy, which essentially represents how much of the output entropy is due to noise rather than the input signal x. The difference between them is the mutual information MI, which characterises how well the output y can be predicted given knowledge of the input signal x. P_y is the output probability function and P_x is the input density function, giving a semi-continuous information channel.

For this relatively simple model, it has been shown to be possible to obtain a closed form expression for the mutual information in terms of the probabilities conditioned on the input value, which can be evaluated numerically without the need to perform any simulation. However, this will not be the case for the more complex model which forms the main part of this chapter, and so we run a simulation for this simple model in addition to evaluating eq.4.8, both to be consistent in our replication and in order to verify that the simulation approach produces reliable results. Loosely following the analysis in (Stocks, 2000a, 2001a), to obtain the expression of mutual information for this simple model, we first observe that

$$P_y(y) = -\int_{-\infty}^{\infty} P_y(n|x) P_x(x) dx \tag{4.3}$$

and

$$P_y(n|x) = C_n^N P_{0|x}^{N-n} P_{1|x}^{n} \tag{4.4}$$

which means that the noise entropy can be rewritten as

$$H_{noise} = -\int_{-\infty}^{\infty} P_x(x) dx \sum_{n=0}^{N} P_y(n|x) \log_2 C_n^N P_{0|x}^{N-n} P_{1|x}^{n} \tag{4.5}$$

Observing that

$$\begin{aligned} \sum_{n=0}^{N} (N-n) P_y(n|x) &= N P_{0|x} \\ \sum_{n=0}^{N} n P_y(n|x) &= N P_{1|x} \end{aligned} \tag{4.6}$$

we can further simplify the noise entropy to

$$H_{noise} = -\int_{-\infty}^{\infty} P_x(x) dx \left(\sum_{n=0}^{N} P_y(n|x) \log_2 C_n^N + N(P_{0|x} \log_2 P_{0|x} + P_{1|x} \log_2 P_{1|x}) \right) \tag{4.7}$$

which finally gives us the mutual information

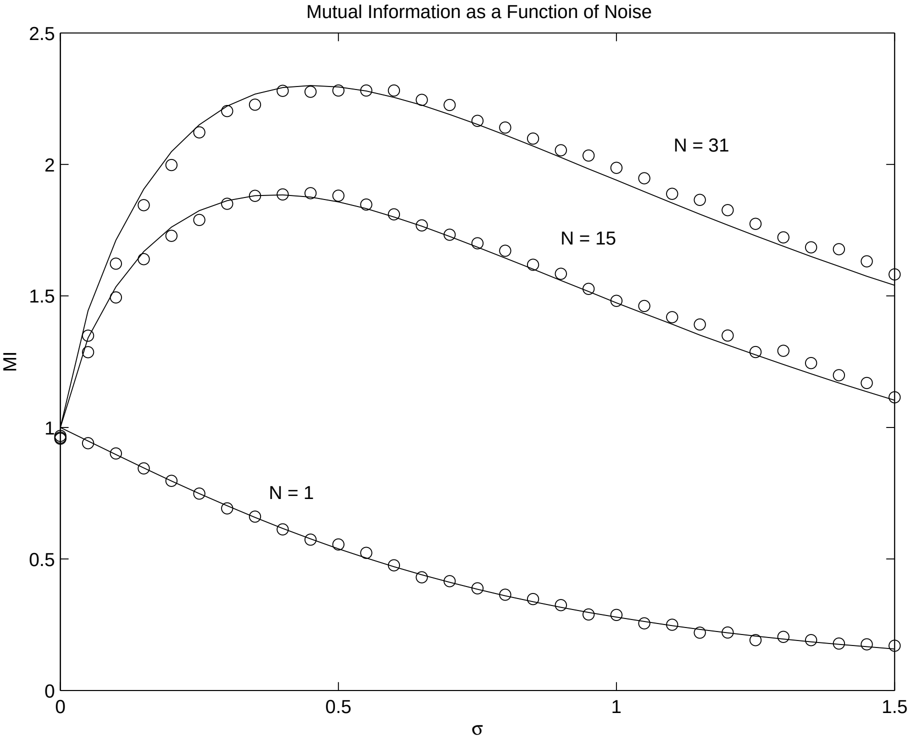


Figure 4.1: Mutual information shown as a function of the noise strength parameter. The SSR effect, information transmission is optimised by a non-zero noise level, is clearly evident. The effect becomes more pronounced for a greater number of processing units. In this figure, the circles represent values from the digital simulation and the lines represent numerical evaluation of eq.4.8; in all subsequent figures, the circles still represent data points but the lines are simply visual aides.

$$\begin{aligned} MI &= -\sum_{n=0}^{N} P_y(n)\log_2 P_y(n) \quad - \\ & \left[-\int_{-\infty}^{\infty} P_x(x)dx\left(\sum_{n=0}^{N} P_y(n|x)\log_2 C_n^N + N(P_{0|x}\log_2 P_{0|x} + P_{1|x}\log_2 P_{1|x})\right)\right] \end{aligned} \tag{4.8}$$

The results for both evaluating the expression and running a simulation are shown in fig.4.1. The SSR effect can clearly be seen, with a certain (non-zero) level of noise producing optimal information transfer when there is more than one unit present. The results from the two methods (numerical evaluation and simulation) show close agreement, and this figure also clearly replicates fig.2 in (Stocks, 2000a). We may reasonably conclude that the current approach is therefore valid for the task at hand.

Stocks (Stocks, 2000a, 2000b, 2001a, 2001b) has suggested that this SSR effect is quite different from traditional stochastic resonance, and explained the mechanism for this SSR effect in terms of the noise giving access to more information in the

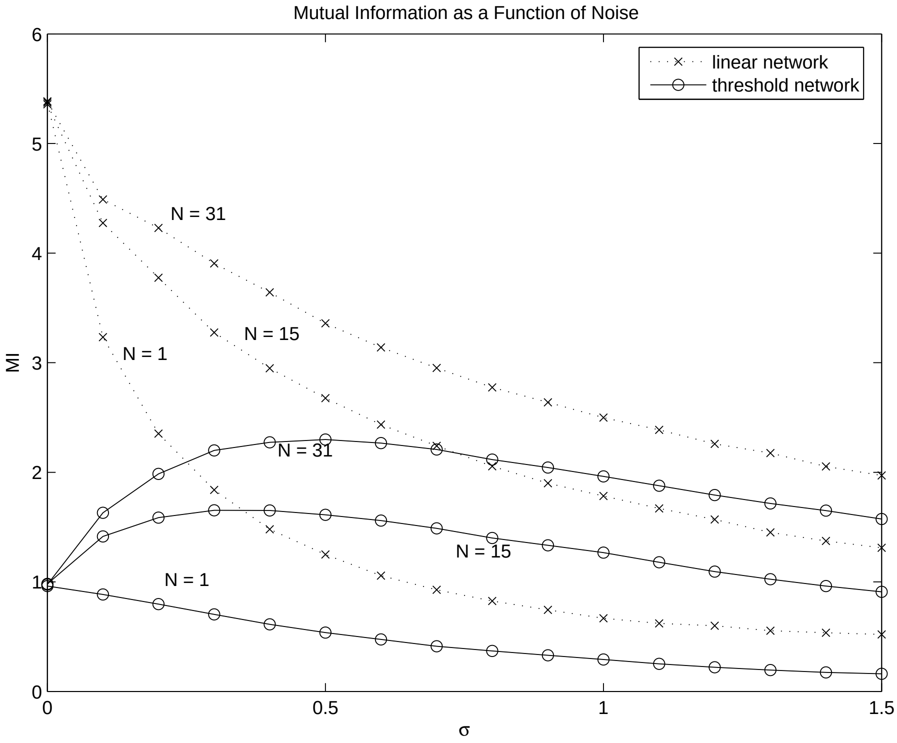


Figure 4.2: Mutual information shown as a function of the noise strength parameter for two different network types. The SSR effect is evident for the network with Heaviside functions, but notably absent from the network with linear functions.

original signal by effectively varying the firing thresholds across the units, which will clearly give a better performance than if all units behave identically (a situation of maximum redundancy). One way to test this explicitly is to compare the network of Heaviside functions with one of linear functions; both networks are the identical to the one before except for the output neurons, and hence both still process instantly in discrete time (i.e. each input is handled completely and independently from every other input; there are no residual network state dynamics in these cases). The inputs are the same as before (a random Gaussian signal which is the same for all units at any one time, and an additional random noise component which is independent for each unit), and the network of Heaviside functions is also the same as the one described previously; hence for that network we have

$$\begin{aligned}
P_\eta(\eta) &= \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp \frac{-\eta^2}{2\sigma_\eta^2} \\
P_x(x) &= \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp \frac{-(x-\mu_x)^2}{2\sigma_x^2} \\
y_i(t) &= \begin{cases} 1 \text{ if } x(t) + \eta_i(t) > \theta_i \\ 0 \text{ if } x(t) + \eta_i(t) < \theta_i \end{cases} \\
\theta_i &= \theta = \mu_x \text{ for all } i \\
y(t) &= \sum_i y_i(t)
\end{aligned} \tag{4.9}$$

The network of linear functions is the same as the Heaviside network in all regards except for the individual unit output, which is simply the input passed through unchanged (a simple linear function); hence for that network we have

$$\begin{aligned}
P_\eta(\eta) &= \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp \frac{-\eta^2}{2\sigma_\eta^2} \\
P_x(x) &= \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp \frac{-(x-\mu_x)^2}{2\sigma_x^2} \\
y_i(t) &= x(t) + \eta_i(t) \\
y(t) &= \sum_i y_i(t)
\end{aligned} \tag{4.10}$$

Fig.4.2 shows the mutual information and output entropy for the two networks, across a range of noise levels and network sizes. The SSR effect can be seen in the network with Heaviside functions, but is completely absent in the network with linear functions, something which is consistent with Stocks' explanation. It highlights the fact, however, that in order for an SSR effect to be possible, each individual unit must be suboptimal, otherwise there is no extra information for the noise to give access to, as shown in the case of the linear activation functions. In addition, it makes it clear that although multiple units are needed for the SSR effect, and that the effect is larger for a greater number of units, this is not (solely) due to a central limit effect (which exists in both networks, including the linear function network where there is no SSR).

4.3 Integrate-and-Fire Model

We first describe a simplified but nevertheless sufficiently realistic neuron model that is widely used in the computational neuroscience community - the leaky integrate-

and-fire model (Feng, 2004), which we previously described in more detail in section 2.1.2. We use the version with no reversal potentials, and receiving spike train inputs (modelled as Poisson processes that can be nonhomogeneous in the general case) which represent the effect on the membrane potential of inputs from other neurons; is therefore Stein's model, shown previously in equation 2.11. This has been used previously to examine traditional stochastic resonance (Feng & Tirozzi, 2000), though our treatment here is somewhat different. We start by repeating equation 2.11 from section 2.1.2:-

$$\frac{dV}{dt} = -\frac{V}{\tau_M} + \frac{I_E}{C_M} + I_S \tag{4.11}$$

With no external electrode input current ($I_E = 0$), explicitly making the membrane potential a function of time, dropping the subscript from synaptic inputs for clarity (as they are now the only form of input) and using Poisson process approximations for the presynaptic spike trains, equation 4.11 can be rewritten as:-

$$\begin{aligned} dV(t) &= -\frac{V(t)}{\tau_M}dt + dI(t) \\ dI(t) &= a\sum_{i=1}^{p} dE_i(t) - b\sum_{j=1}^{q} dI_j(t) \end{aligned} \tag{4.12}$$

where $V(t)$ is the membrane potential and is a function of (continuous) time; τ_M is the membrane time constant which controls the speed of decay of the membrane potential. $dI(t)$ is the input to the neuron and consists of: $E_i(t)$ are the p excitatory postsynaptic potentials (EPSPs) with rate $\lambda_E(t)$ and a magnitude of a; the q inhibitory postsynaptic potentials $I_i(t)$ have a rate $\lambda_I(t)$ and magnitude of b; see section 4.4 for more details of this Poisson approximation. A spike is recorded when $V(t)$ crosses the threshold V_{thre}, at which point the neuron is reset to the resting potential, which gives us a set of firing times m^s:-

$$\begin{cases} m^s &= \inf\{t > m^{s-1} : V(t) \geq V_{thre} | V(m^{s-1}) = V_{rest}\} \\ m^0 &= 0 \end{cases} \tag{4.13}$$

For all experiments presented here, $\tau_M = 20$ms, $V(t)$ was rescaled to have a zero resting potential, $V_{thre} = 20$mV, $p = q = 50$ and $a = b = 0.5$. These values were selected both as being in a plausible biophysical range, and being of the same order of magnitude as those used in (Feng & Tirozzi, 2000; Feng & Ding, 2004) for ease of comparison; it should however be noted that in general, the central behaviour of the model is similar across a wide parameter space.

Our model consists of N integrate-and-fire neurons, each of which receive a continuous noisy input, and may possibly fire in response, depending on whether or not they reach the firing threshold. It is useful when first testing for an SSR effect in integrate-and-fire neurons to adopt a similar approach to that used in the basic SSR model (Stocks, 2000a, 2000b, 2001a) and control the input signal and noise directly, so the inputs $dI(t)$ were changed in this experiment from the usual Poisson processes, to ones directly characterised by independent signal μ_x and noise σ_x parameters which can be manipulated as part of the experiment. The input to the system consists of a stepped signal component (although any signal could in principle be used; the input signal is regarded as continuous in the treatment here) given by a variable $x(t)$ which changes value every 250ms with the same value for all units at a given time point, and a random Gaussian noise component $\eta_i(t)$ with zero mean and variance σ_x, which is independent for each neuron. The output of the system $y(t)$ is the mean firing rate of the population, estimated using a non-causal Gaussian filter with a constant width equal to the membrane time constant (20ms); the period for each input stimulus step (250ms) is much higher than this, giving the neurons an opportunity to adapt to a change in the inputs and stabilise in every case. Hence we have:-

$$\begin{aligned}
y(t) &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \frac{n_{spikes}(t, t+\Delta t)}{N} = \frac{1}{N} \sum_{i=1}^{N} \sum_{s} \delta(t - m_i^s) \\
x(t) &= \sum_{k} \xi_k \chi[t \in \{(k-1)T_W, kT_W\}] \\
\xi_k &= \{10, 20, 30, 40\} \\
P_\eta(\eta) &= \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp \frac{-\eta^2}{2\sigma_x^2}
\end{aligned} \tag{4.14}$$

Here n_{spikes} is the total number of spikes, and $y(t)$ is the average population activity, each occurring over the short time interval, Δt. ξ_k is the input firing rate which steps from 10Hz to 40Hz in increments of 10Hz successively (these values were used to reflect typical firing rates), χ is the indicator function denoting set membership and T_W is the length of time the signal remains at one value, which is 250ms here.

In keeping with previous work on SSR (Stocks, 2000a, 2000b, 2001a), our perfor-

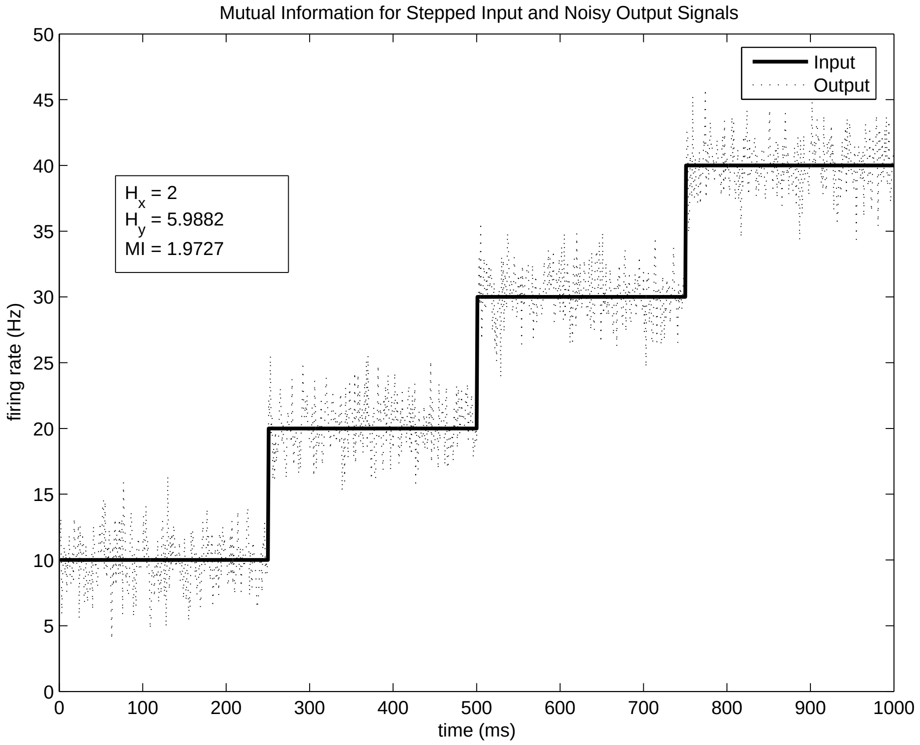


Figure 4.3: Entropy and mutual information with a noise-free stepped input and a simulated noisy output. The mutual information nearly matches the input entropy, reflecting the fact that in spite of the noisy output, the input can still be strongly predicted on the basis of a given output value.

mance measurement is given by the mutual information of the system, which characterises the information processing capability of the system. This is a probability-based approach which essentially measures the extent to which the inputs and outputs can be predicted from each other. It is given by:-

$$\begin{aligned} MI &= H_y - H_{noise} \\ H_y &= -\int_{-\infty}^{\infty} dy\ P_y(y) \log_2 P_y(y) \\ H_{noise} &= -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dx\ dy\ P_x(x)P_y(y|x) \log_2 P_y(y|x) \end{aligned} \qquad (4.15)$$

Here, H_y is the entropy of the output, and H_{noise} gives the noise entropy, which essentially represents how much of the output entropy is due to noise rather than the input signal x. The difference between them is the mutual information MI, which characterises how well the output y can be predicted given knowledge of the input signal x. The mutual information is evaluated numerically throughout. It is important to note that in the continuous time models used here, the integration stepsize was chosen to be small enough to ensure that the system was never allowed to relax (which would effectively be a discrete time model). Note also that the expressions for the output and noise entropies should also contain a term for the logarithm of the measurement resolution with which they were estimated, $\log_2 \Delta$. However, as the measurement resolution was held constant across those two expressions in each case, these terms drop out of the expression for the mutual information and so have been omitted for convenience.

In general, noisier inputs will lead to lower mutual information all other things being equal, but if the information processing capability of the system is actually enhanced by the presence of a certain level of noise, then we would expect to see the mutual information increase in these circumstances. Hence mutual information is a very appropriate way to test for SSR. In our model, if the population of neurons have an input that is completely predictable on the basis of the firing rate output, even if the output itself is not smooth (as we would expect for a neural population subject to Gaussian noise) or in the same range as the input, then the mutual information will match the output entropy, and the noise entropy will be zero. An example of the type of input signal, output and mutual information measurement used in our model, here simply using randomly generated data, is shown in fig.4.3.

The network was presented with the input signal shown in fig.4.3, a stepped signal changing every 250ms, giving four distinct input values in total. The result for networks of different numbers of neurons, in terms of mutual information as a function of the noise parameter σ_x, is shown in fig.4.4. This clearly shows that the SSR effect does exist for integrate-and-fire neurons operating in continuous time, and has a strong similarity to the results of the basic model. Networks with more than one neuron improved performance with some noise present, with greater improvement for a greater number of neurons.

4.4 Poisson Inputs and Diffusion Approximation

The experiment in the previous section demonstrates the existence of SSR in networks of integrate-and-fire neurons operating in continuous time, but does so by directly specifying the noise level. In practice, neurons are subject to spiking inputs from other neurons that are most commonly modelled as Poisson processes,

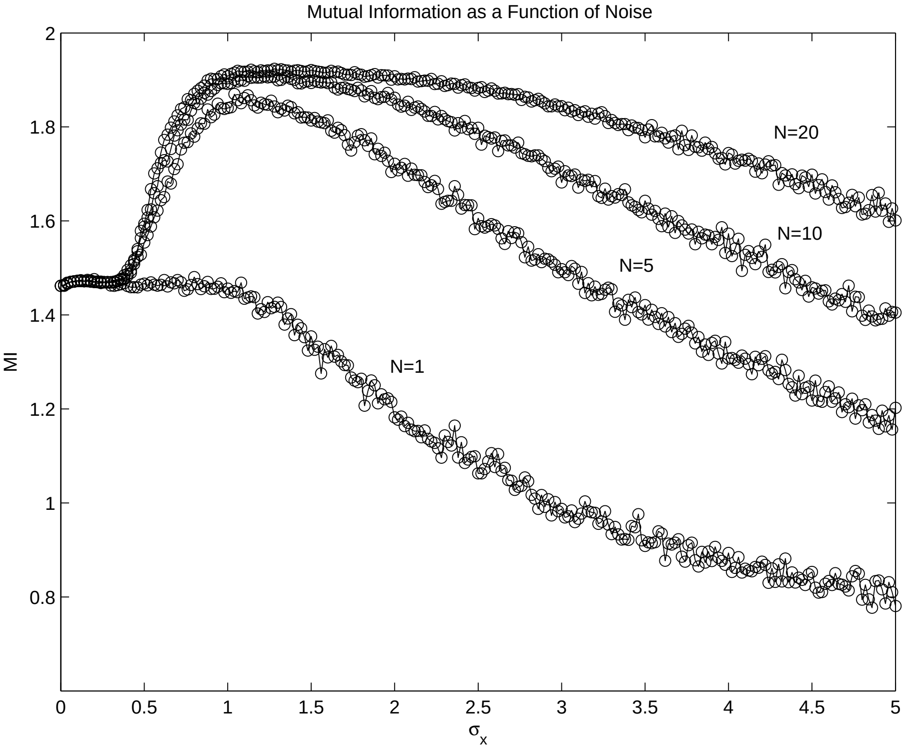


Figure 4.4: Mutual information shown as a function of the noise strength. Networks containing more than one neuron benefit from some noise as predicted by SSR theory.

and where the noise level scales with the signal; in other words, we are studying a system with signal-dependent noise.

An integrate-and-fire neuron receiving Poisson inputs is described by eqs.4.12 and 4.13, where $dI_{syn}(t)$ gives the Poisson inputs. Following a previous analysis (Feng & Tirozzi, 2000), invoking a diffusion approximation and thereby separating out the signal (first order) and noise (second order) terms, we can rewrite these inputs as

$$dI_{syn}(t) = a\sum_{i=1}^{p}\lambda_E(t)dt - b\sum_{j=1}^{q}\lambda_I(t)dt + a\sum_{i=1}^{p}\sqrt{\lambda_E(t)}dB(t) - b\sum_{j=1}^{q}\sqrt{\lambda_I(t)}dB(t) \quad (4.16)$$

where $\lambda_E(t)$ is the excitatory Poisson process parameter, $\lambda_I(t)$ is the inhibitory Poisson process parameter, $B(t)$ is a standard Brownian motion and all other variables are as previously specified. Noting that the standard deviation of a sum or difference of N (not generally independent) Gaussian processes with mean individual variance σ^2 and mean correlation c is given by $\sqrt{N(\sigma^2 + c(N-1)\sigma^2)}$, we can further simplify this to

$$\begin{aligned} dI_{syn}(t) &= \left(ap\lambda_E(t) - bq\lambda_I(t)\right)dt + \\ &\left(\sqrt{a^2\lambda_E(t)p + b^2\lambda_I(t)q + a^2\lambda_E(t)p(p-1)c + b^2\lambda_I(t)q(q-1)c}\right)dB(t) \end{aligned} \tag{4.17}$$

in which the first term represents the signal, and the second term describes the noise. In order to place this in the context of SSR, it is necessary to be able to express the input as a single variable with a mean value which matches the neural firing threshold. In the context of integrate-and-fire neurons receiving excitatory and inhibitory Poisson inputs, this is usually referred to as the case of balanced inputs (Feng & Ding, 2004; Rossoni & Feng, 2007), where the input is characterised by a single rate parameter λ_x, and a ratio of excitatory to inhibitory input strengths r, which is adjusted to ensure that the inputs are balanced. For all the simulations presented, we consider the *weakly* balanced condition, that is one in which the inputs are balanced on average across the entire simulation period, this ensures that our simulation is both consistent with our previous experiments, and also much more realistic than the implied changes of connectivity in mid-simulation as is the case for the *strongly* balanced condition in which the input ratio is adjusted whenever the stimulus changes value.

The analysis given by Feng and Tirrozi (Feng & Tirozzi, 2000) focuses on the special case of q=0 only and focuses on the SNR equation. Here we examine the more general case where q can take any real value, which means we must first rewrite the signal and noise terms described in terms of separate mean and variance

$$\begin{aligned} dI_{syn}(t) &= \mu(t)dt + \sigma(t)dt \\ \mu(t) &= ap(\lambda_E(t) - \lambda_I(t)) \\ \sigma^2(t) &= a^2p(\lambda_E(t) + \lambda_I(t) + c(p-1)\lambda_E(t) + c(p-1)\lambda_I(t)) \end{aligned} \tag{4.18}$$

Balanced inputs imply that the average synaptic input matches the firing threshold, i.e. $E\{\frac{dV(t)}{dt}\} = 0$ and $E\{V(t)\} = V_{thre}$, which means

$$\begin{aligned} 0 &= E\{\frac{1}{\tau_M}V_{thre} + I_{syn}(t)\} \\ \Rightarrow \frac{V_{thre}}{\tau_M} &= \mu(t) \end{aligned} \tag{4.19}$$

because $\mu(t)$ is the expected signal value and the expected noise value is zero by definition. Expressing this in terms of our rate parameter λ and ratio r

$$\begin{aligned}\lambda(t) &= \sum_{i=1}^{p} \lambda_E(t) \\ r(t)\lambda(t) &= \sum_{j=1}^{q} \lambda_I(t)\end{aligned} \tag{4.20}$$

which gives us the general result

$$\begin{aligned}r(t) &= \frac{a}{b} - \frac{V_{thre}}{b\lambda(t)\tau_M} \\ \mu(t) &= \frac{V_{thre}}{\tau_M} \\ \sigma^2(t) &= a^2\lambda(t)(1+c(p-1)) + (ab\lambda(t) - \frac{bV_{thre}}{\tau_M})(1+c(q-1))\end{aligned} \tag{4.21}$$

For the current experiment, $a = b = 1$, $p = q = 50$ and $c = 0$ (uncorrelated inputs), and we also need to be able to separately control the noise level in order to determine whether or not any non-zero noise level gives improved performance. For this experiment we use a noise coefficient σ_x prepended to the synaptic input expression given in eq.4.21 that could reflect the variability of noise levels in biophysical systems. These facts allow us to simplify eq.4.21 for this particular experiment to:-

$$\begin{aligned}r(t) &= 1 - \frac{V_{thre}}{a\lambda(t)\tau_M} \\ \mu(t) &= \frac{V_{thre}}{\tau_M} \\ \sigma^2(t) &= \sigma_x(2a^2\lambda(t) - \frac{aV_{thre}}{\tau_M})\end{aligned} \tag{4.22}$$

Presenting the network with the same stepped input stimulus as in the previous experiment, and with the input rate parameter $\lambda = 50$ we obtained the results shown in fig.4.5. The SSR effect is once again clearly present. These experiments are an important development from existing work because they here use neuron models with biophysically plausible parameters and in receipt not of continuous driving currents but of presynaptic spike trains modelled as Poisson processes. Altogether this represents a dynamically realistic (albeit highly simplified) neural scenario, and shows clearly that SSR can play a beneficial role in neural information processing in such a scenario.

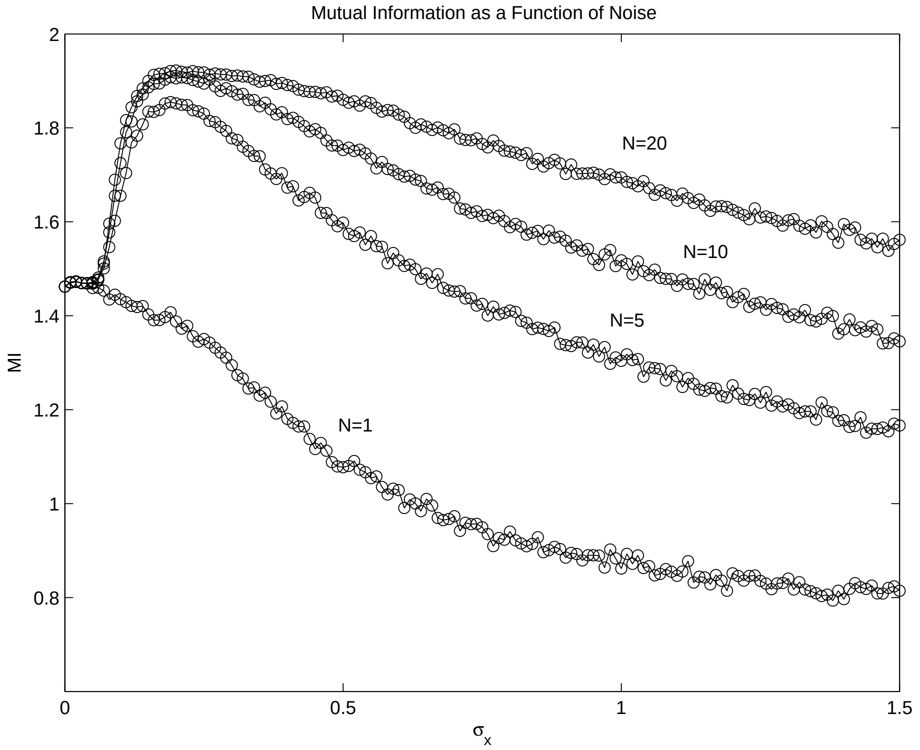


Figure 4.5: Mutual information shown as a function of the noise strength. Once again networks containing more than one neuron benefit from some noise.

4.5 Correlation Control

In the previous experiment, the Poisson inputs were uncorrelated, that is $c = 0$ in eq.4.21. This may not be true in the general case, so it is interesting to see how the correlation between inputs affects SSR. Correlation has previously been studied in the context of SSR (Hoch et al., 2005), where it was found that positive correlations in the background noise decrease information transmission, and decrease the benefits of population coding. The authors studied a population of IF neurons in the context of an identical aperiodic Gaussian stimulus input to each neuron, plus zero-mean Gaussian noise. Importantly for our current work, the correlation of the noise input to each neuron was explicitly controlled in order to study the influence of noise correlation on both performance and the SSR effect. Figure 4.7 shows the central results of this research. It is clear that the SSR effect is still present even in the case of correlated noise; this is an important and useful result. It is also clear, however, that the presence of positive correlations in the noise reduces the overall performance of the system. This is not at all surprising, as the central limit theorem (see section 1.2.3) demonstrates that a component of a correlated noise system is

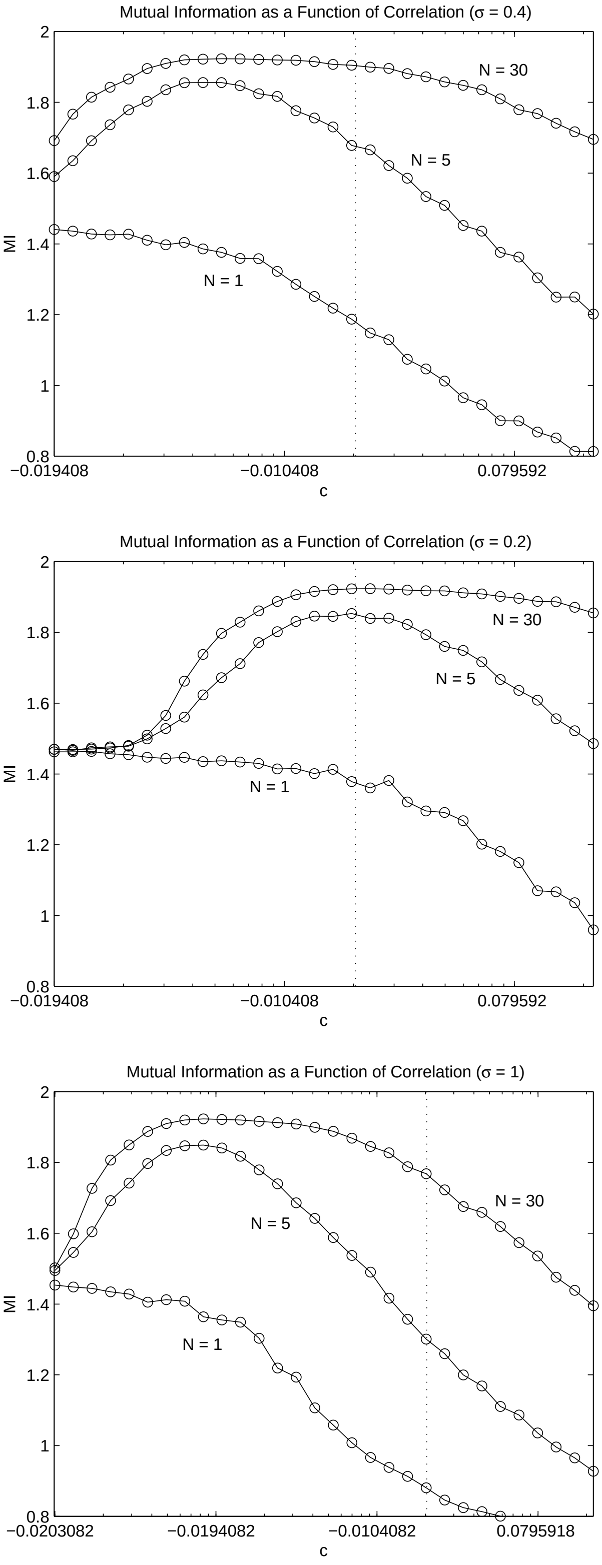


Figure 4.6: Mutual information as function of correlation. The vertical dotted line gives the zero correlation position; logarithmic spacing is used for visual clarity. *Top:* $\sigma_x = 0.4$; negatively correlated inputs give optimal performance *Middle:* $\sigma_x = 0.2$; uncorrelated inputs give optimal performance at this noise level *Bottom:* $\sigma_x = 1$; negative correlated inputs give optimal performance in the absence of a special noise parameter

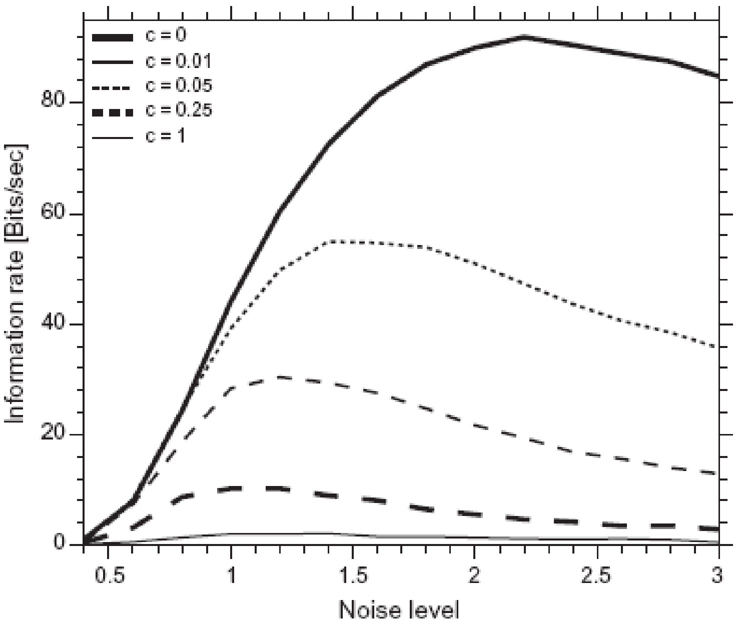


Figure 4.7: Mutual information shown as a function of noise strength for several different correlation levels. From (Hoch et al., 2005).

equivalent to a component of an uncorrelated noise system of a higher magnitude, and higher noise generally leads to lower performance. This is also the reason that the maximum mutual information point is at increasingly lower values of the noise magnitude parameter as the correlation parameter value increases as shown in the graph (although this explanation is not present in the paper).

Although the Hoch, Wenning and Obermayer work is clearly relevant to the work we present in this chapter, there are three important differences. Firstly, as in previous cases such as (Zohary, Shadlen, & Newsome, 1994), the authors focus exclusively on positive correlation. This is unfortunate, especially given that they found the optimal coding strategy in terms of maximising mutual information was obtained at the minimum correlation value examined (c=0). Secondly, they were looking at the effects of correlation in the noise, rather than in neural firing; this is also unfortunate as noise of a given level is most usefully assumed to be uncorrelated, since correlated (zero-mean) noise is simply equivalent to uncorrelated noise of a lower level. Finally, they do not attempt to explain how the correlations come to be present in the neural system; this is understandable as it is beyond the scope of their paper, but nevertheless such an explanation would be welcome. In the model that we outline here we seek to address these three points, studying the more bio-

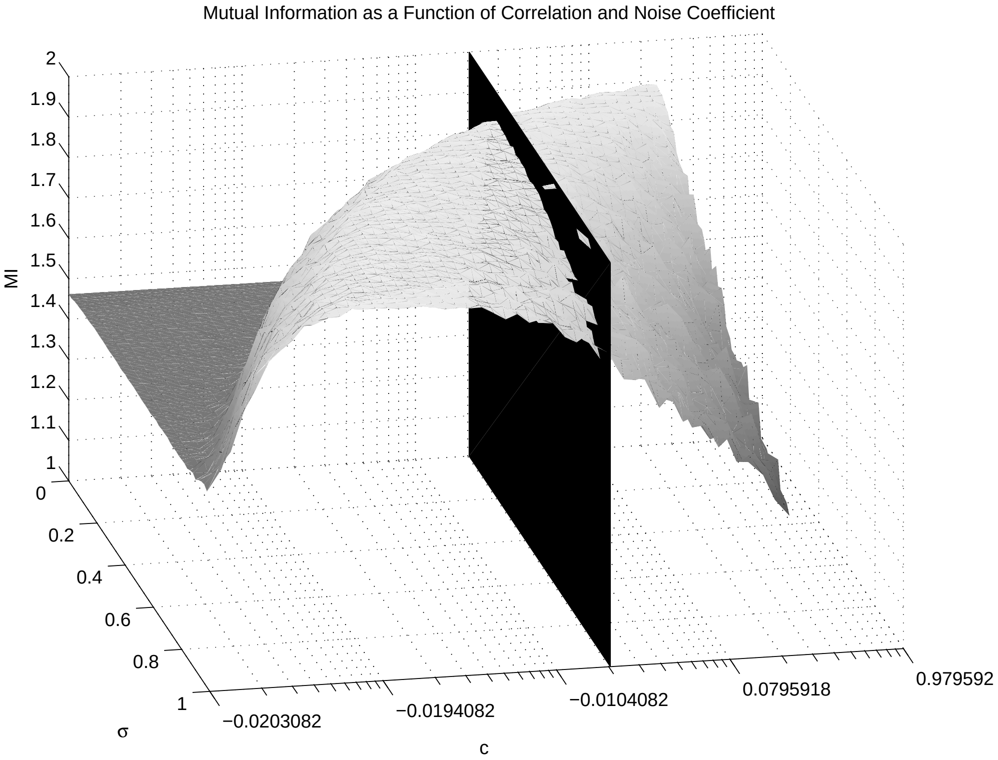


Figure 4.8: Mutual information shown as a function of correlation and noise strength. The vertical plane shows the position of zero correlation.

physically relevant case of the correlations in neuronal firing, rather than separating out the noise component, and place this research in the context of the benefits of negative correlation.

We first fixed inherent input noise parameter σ_x at a chosen value (shown in the figures), and then measured mutual information as a function of correlation instead.

It can be seen in fig.4.6 that for a typical low value (< 1) of the input noise parameter ($\sigma_x = 0.4$), negatively correlated inputs give optimal performance. In fact, σ_x must be approximately 0.2 for small numbers of neurons N to have optimal performance from uncorrelated inputs. In the particular case of $\sigma_x = 1.0$, which is equivalent to having no special noise parameter, optimal performance comes from negatively correlated inputs. The more general result across a range of σ_x and correlation values can be seen in fig.4.8. The ridge of highest mutual information values shows that as the noise parameter becomes larger, the correlation must become more negative to compensate and retain optimal information transmission.

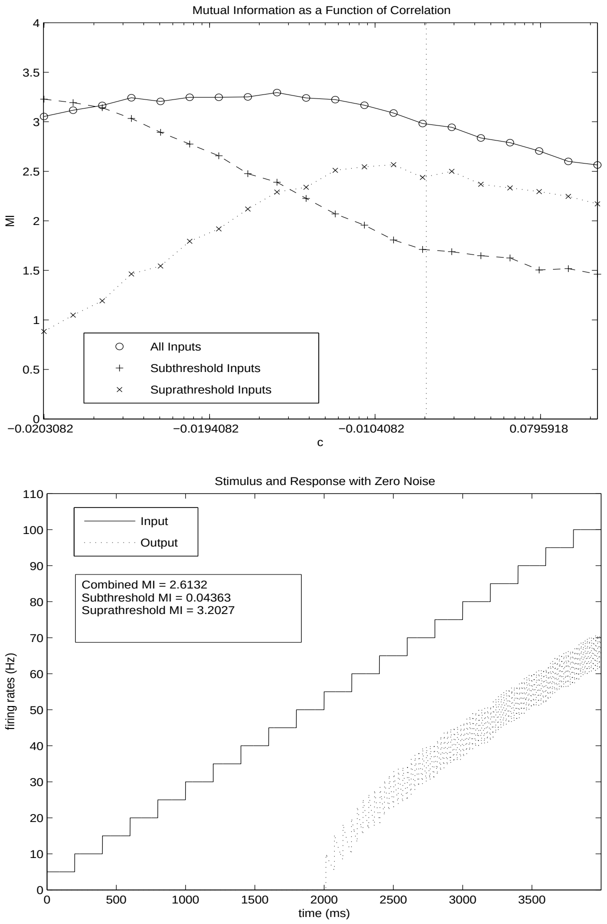


Figure 4.9: *Top:* Mutual information as a function of correlation for all inputs, subthreshold inputs and suprathreshold inputs. The input firing rate threshold to stimulate any output at zero correlation is 50Hz. This figure shows that increased noise is beneficial for the subthreshold part of the signal, and detrimental to the suprathreshold part of the signal. *Bottom:* Stimulus and response for a zero-noise case. The flat response in the subthreshold region results in low mutual information in the absence of noise.

When testing the network with weakly balanced inputs, it is instructive to examine the mutual information separately for the subthreshold and suprathreshold regions of the input space, as well the total (combined) mutual information. The top of fig.4.9 shows these values for a network with 20 neurons, using a stepped signal with ten different equally-spaced input values, half of which are subthreshold (<50Hz) and half of which are suprathreshold (>50Hz) by definition. It is clearly apparent that as correlation becomes more positive, thereby increasing noise, the suprathreshold signal loses MI, while the subthrehold signal gains MI. This provides an interesting link between SSR and classical SR; we are seeing a classical SR effect for the subthreshold component of the signal which up to a certain level outweighs the detrimental effect on the suprathreshold signal, giving an overall benefit. The bottom of fig.4.9 clearly shows the reason why; the subthreshold signal has no response at all, because the signal is not strong enough to reach threshold, and there is no noise to assist it. This is both traditional SR, because noise would clearly help to sometimes push the input over threshold, and more often the closer the signal is to the threshold, and SSR because varied thresholds in the neurons would allow some to respond to a subthreshold signal because it their individual thresholds would also be below the signal DC level. This is the theoretical link between SR and SSR.

The fact that noise is required to boost subthreshold signals, but potentially damaging for suprathreshold, suggests that an adaptive correlation control procedure could be highly beneficial in optimising information transmission. A relatively high correlation (according to a model-specific scale; in this case relatively high essentially means uncorrelated) at low signal levels could generate allow a high level of noise that would boost the signal into the threshold region, and as the signal level becomes higher, the correlation should become more negative, reducing noise and thus limiting the detrimental effect on the suprathreshold signal region. An example of this is shown in fig.4.10; the top gives the piecemeal adaptive correlation function shown, and the bottom gives the simulation of the same network shown in fig.4.9 but now with adaptive correlation control. It can be seen that overall information transmission is improved by using the adaptive procedure, controlling noise level in response to prevailing stimulus conditions. This example is certainly not optimal in any sense, but is given as a proof of concept. The question of the optimal

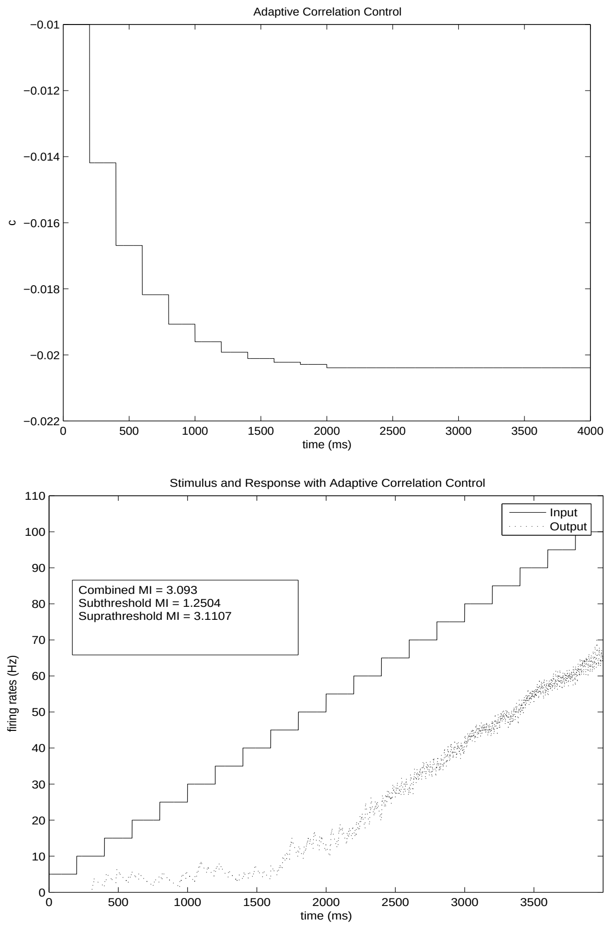


Figure 4.10: *Top:* Adaptive correlation control. The correlation decreases as a function of input, using an piecemeal step function as shown. *Bottom:* Stimulus and response with adaptive correlation control. The input firing rate threshold to stimulate any output at zero correlation is 50Hz. The subthreshold region now has some response, though not optimal, and the suprathreshold region remains largely free of the detrimental effect of noise.

adaptive technique to use here remains open to research, as does the question of how, if at all, the brain implements such a feature. Adaptive correlation control requires some knowledge of the input stimulus, such as the range, and it may be supposed that more precise characterisation of the input signal will allow for a more accurate adaptive mechanism. Biophysical constraints in the brain, however, make advance knowledge of at least the approximate range of the presynaptic signal quite plausible, so neural adaptive correlation control is a serious possibility that requires further investigation. On this final point, it is interesting to note some very recent evidence coming out of multi-electrode array data (Horton, Bonny, Nicol, Kendrick, & Feng, 2005) that suggests that neural firing is in fact negatively correlated during stimulus conditions when firing rates are higher, and largely uncorrelated during rest conditions with lower firing rates, which is entirely consistent with our findings here. This will be discussed more in the next two chapters.

4.6 Conclusions

Suprathreshold stochastic resonance is an important new phenomenon that offers a functional explanation of noise in information processing systems. We have developed a framework for SSR in continuous time in the context of integrate-and-fire neural models, both with free noise and with Poisson process spiking inputs modelled by a diffusion approximation. In particular, we have shown that correlation between the inputs can be used to tune the noise, thus providing a biophysically plausible mechanism, and demonstrated how adaptive correlation control can be used to further improve information transmission, something which preliminary evidence suggests may actually be the case for *in vivo* neural systems; it is notable that neural firing within the local population should be negatively correlated to provide optimal information transmission in response to a strong stimulus. Finally we have shown an interesting theoretical link between SSR and traditional SR. It is too early to say for certain that SSR exists in real neural systems, but the demonstration and explanation outlined here suggest that this is very likely to be the case, and the evidence to date is encouraging.

Chapter 5

Lateral Inhibition Within Cortical Columns

We have seen in the previous chapter that negatively correlated inputs can control the noise level that a neuron receives, which in turn aids information transmission through the mechanism of suprathreshold stochastic resonance. In that chapter, we used a Brownian diffusion approximation to a set of Poisson inputs that would act as presynaptic spike trains, allowing us to directly control the correlation of the presynaptic inputs. In this chapter, we shift the focus to examine those inputs more closely, and simulate the presynaptic neurons directly rather than through a collective approximation. As such, we are now concerned with the noise level in the output of these neurons (which would act as an input to the next layer of connected neurons in a larger model), and how it may aid information transmission. In addition, we now allow correlation to arise naturally out of the interconnections between neurons rather than simply setting a correlation parameter, a major step in the direction of biophysical plausibility. We also use more biophysically realistic neurons and synapses, and test the model with a specific visual tracking task. The result is a model of much greater complexity and biophysical plausibility, and we aim to see whether or not the benefits of negative correlation, seen in the previous chapter, still exist in this more demanding scenario.

5.1 Overview

5.1.1 Introduction

Understanding the functional meaning of particular aspects of neural architecture is a central objective of neuroscience. Inhibitory interneurons are very common in the neocortex, and lateral inhibition has been shown to play an important role in sharpening the distinctions between similar inputs, where such inputs would otherwise invoke nearly the same response in neurons that have only slightly different response properties. However, it is our belief that inhibition also plays an important role in population coding (Tate et al., 2005; Nicol et al., 2005), stabilising the mean field potential (see equation 5.1) and greatly improving the ability of a group of neurons to accurately represent a given stimulus, by creating negatively correlated firing patterns. In this chapter, we will describe the principle on which this improvement is based, showing how it extends the benefit of population coding. We will then present a model and a set of experiments using this model, which demonstrates that pools of neurons operating with inhibitory connections perform better on a stimulus-tracking task than the same model with no inhibitory connections. The model is designed to show how this can work in principle in neural sensory coding, incorporating a number of realistic aspects such as the use of spiking neurons, online estimation and a simple filtering mechanism. To extend our results to a biophysically realistic model is straightforward and we will include a brief discussion on how our model is related to the circuits found in the primary visual cortex. Furthermore, we want to point out here that negatively correlated firing is opposite to the scenario of synchronised firing which is still a hot topic discussed in the literature (Palanca & DeAngelis, 2005; Gray, Knig, Engel, & Singer, 1989).

5.1.2 Inhibitory Mechanisms in Neocortex

Sensory neural processing is most often thought of in terms of many interconnected circuits using excitatory connections to propagate signals through layers of increasingly abstract representation. Whilst there is some truth in this necessarily simplified image, it is also the case that a substantial fraction of neurons in the brain are inhibitory. Inhibitory neurons can operate in a variety of different ways, such as feedforward inhibition, where excitatory and inhibitory connections project from the same area to areas that are typically opposite in function, and feedback inhibition, where excitatory neurons suppress the activity of other neurons in the same area through local inhibitory interneurons. Much of the existing work on inhibitory mechanisms in the context of cortical columns has focused on the role of lateral inhibition in sharpening distinctions between neurons with slightly different response properties (Ratliff, 1972; Martin, 1984). However, the existence of local inhibitory circuits (Tucker & Katz, 2003b, 2003a) and the fact that horizontal connections lead to inhibitory postsynaptic potentials (Hirsch & Gilbert, 1991), suggest that inhibitory circuits are likely to operate within cortical columns, as well as across them. Clearly, inhibitory connections within cortical columns do not have the same role as those that operate across different columns. Lateral inhibition within columns means that neurons with the same response properties are inhibiting each other. What could be the functional role for this local inhibitory mechanism? In this chapter, we propose that this local inhibition improves the performance of pooled neurons, by exploiting one particular aspect of the law of large numbers applicable to population coding.

5.1.3 Population Coding and the Law of Large Numbers

Many areas of the neocortex show a columnar structure, in which all of the neurons in a given layer within the column (sometimes called a pool of neurons) have essentially the same response properties. Somatosensory cortex and primary visual cortex (V1) are two prominent examples of this. Cortical columns are an important structural unit in the brain, and operate on the principle of population coding, where the activity of the pool as a whole is taken to be the signal, rather than the firing rates of the individual neurons. This mean population activity (Gerstner & Kistler, 2002) was given by equation 2.2.1, repeated here for convenience:-

$$A(t) =^{\lim}_{\Delta t \rightarrow 0} \frac{1}{\Delta t}\frac{n_{spikes}(t, t+\Delta t)}{N} = \frac{1}{N}\sum_{j=1}^{N}\sum_{s}\delta(t - m_j^s) \tag{5.1}$$

In this equation, n_{spikes} is the number of spikes, and $A(t)$ is the average population activity, each occurring over the short time interval, Δt. The advantage of this scheme for neural processing is that it allows variations in firing rates of individual neurons, due to extrinsic or intrinsic noise, whilst maintaining the signal-to-noise ratio through the cancellation of the noise when the firing rates are pooled together. This is an effect of the central limit theorem as described in section 1.2.3 (and shown in figure 1.2). When a group of random elements, for example neural firing rates, are summed together, they give a mean population activity (mean field potential). For a system in which the random elements consist of signal plus zero-mean noise (including but not limited to Gaussian noise), which is a typical assumption for most noisy systems including neurons, the mean value will tend to be closer to the signal value because the noise cancels out. More elements pooled together results in greater noise cancellation. There is, however, a way to accelerate this effect, and this is to have a noise component that is negatively correlated. In order to take advantage of this fact, however, a system has to have a way to influence the correlation of the noise component, or at least the effect of the noise component.

In neural systems (and more generally in all threshold systems), a centrally important fact is that a neuron which spikes will be on average more likely to have a positive noise component than a negative one (because positive noise components lead to an increased membrane potential, which in turn increases the probability of spiking). In order to simulate the effect of having negatively correlated noise (without actually effecting the extrinsic or intrinsic noise directly, which is by definition beyond control), this neuron must reduce the membrane potential of the other neurons in the pool. This can be achieved through the inhibitory connections identified in section 5.1.2, and the result is that neurons within a pool should have firing patterns that are negatively correlated with each other. We believe that this will result in a more stable system, with an estimated signal value closer to that of the underlying signal, which will give an improved performance on whatever task the neural pool undertakes. In the following sections, we test this hypothesis on the task of tracking a moving stimulus using groups of pooled neurons.

5.2 Methods

5.2.1 Network Model

Our model [1] consists of a set of $N_c = 10$ cortical columns, each containing $N_n = 100$ integrate-and-fire neurons (see section 2.1.2), which are either unconnected (control condition; an effective synaptic strength of zero), or have a full set of inhibitory connections to all other neurons in the column (there were no recurrent connections back to the originating neuron itself). There are no connections across columns. Each neuron is parameterised by v_{thre} (firing threshold), v_{rest} (the resting and reset potential), τ_M (the membrane time constant), C_M (the total membrane capacitance) and x_{col} (the stimulus position to which neurons in the column respond most strongly). In the experiments presented here, v_{rest} was always set to 0 (using a standard rescaling approach from realistic values, for ease of analysis), C_M was always set to 1nF and τ_M was always set to 20ms. The value of v_{thre} was varied in the experiments (details are given in Sections 5.2.4 and 5.3.3). x_{col} took evenly spaced values between 0 and 10 (inclusive), which were chosen to coincide with the limits of the stimulus range (in order that the model could handle all stimulus positions, but also that the full range of stimulus positions that could be handled by the model were available to be tested).

The model used here, and defined by equation 5.2, is Stein's model (Stein, 1965, 1967), previously introduced in section 2.1.4, and here rewritten to include column, neuron and where appropriate synapse indices. This is a classical leaky integrator model with simple Dirac synapses. Our motivation for choosing this particular model, as in the previous chapter also, is that this model has been widely used in studies of stochastic noise (starting with Stein's own work, and including more recent work e.g. (Feng & Tirozzi, 2000)), it maintains an equivalent approach for the spiking neuron inputs and the external stimulus inputs, and in particular it allows a direct mapping of the relationship between the spiking neuron input current and the effect on the membrane potential, without being complicated by the effect of the time trajectory and the existing state of the membrane potential. However, we

[1] A software tool for experimentation with our model is available at www.sussex.ac.uk/Users/sjd23/lateral_inhibition.html

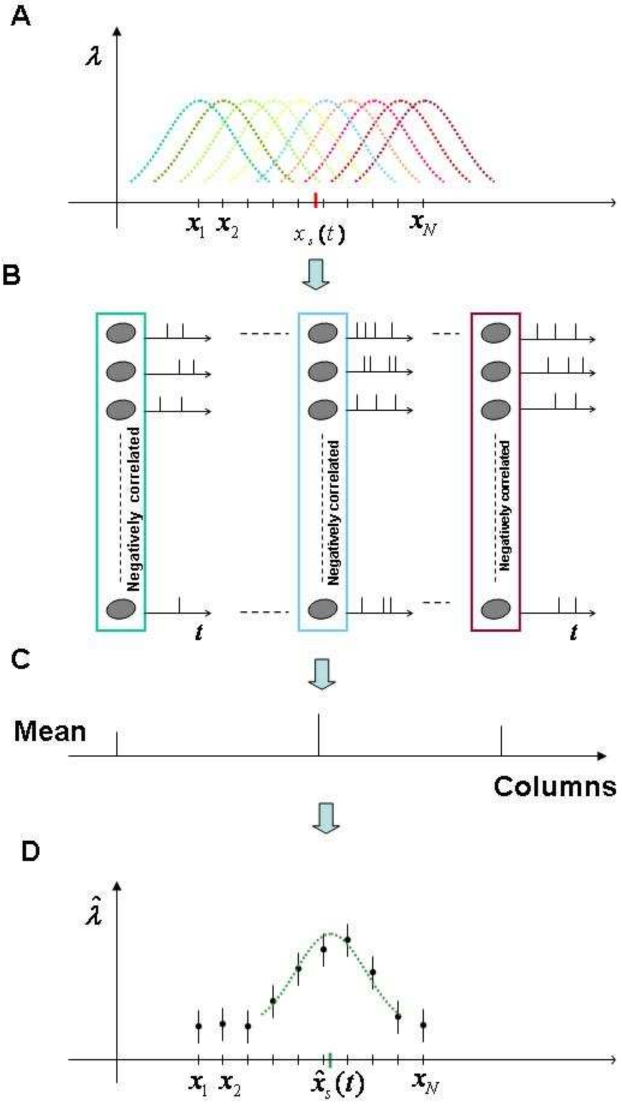


Figure 5.1: Tracking a moving target: schematic drawing of the setup. A: Tuning curves for cortical columns (coloured dotted lines), centred on different parts of the input space. The red line on the x-axis represents an example input $x_s(t)$. B: Neurons are grouped into columns (boxes) with each column response tuned to specific locations along the input space. Columns whose tuning curve is centred near the example input have a higher firing rate (more spikes) than columns that are tuned to respond more strongly to stimuli in other locations. C: The position of a moving target $x_s(t)$ is inferred from the ensemble spike activity sampled during a short time interval by aggregating the responses within each column, shown here, which each act as a vote for their optimal response position (the centre of their tuning curve) in estimating the input position ($\hat{\lambda}$), with the activation acting as a weight for that vote. D. The column responses are filtered using a simple Gaussian filter (see text for more details) in order to sharpen the response and give a final estimate of the input position.

are aware of the need to make the link with more biophysically realistic models as well, and present the results of simulations with such models in section 5.4.

At each simulation step, external inputs (determined by stimulus position and a random element; see section 5.2.2) were presented to each of the neurons, and the membrane potential $v^{(i,j)}$ of the jth neuron in the ith column was updated according to the following leaky integrator equation:

$$\frac{dv^{(i,j)}(t)}{dt} = -\frac{v^{(i,j)}(t)}{\tau_M} + \frac{I_E^{(i,j)}(t)}{C_M} + I_S^{(i,j)}(t) \tag{5.2}$$

where we have

$$\left\{ \begin{array}{rcl} I_E^{(i,j)}(t) & = & \mu(\lambda^{(i,j)}(t)) + \sigma(\lambda^{(i,j)}(t))B_t \\ \mu(\lambda^{(i,j)}(t)) & = & a\lambda^{(i,j)}(t)(1-r) \\ \sigma(\lambda^{(i,j)}(t)) & = & a\sqrt{\lambda^{(i,j)}(t)(1+r)} \\ r(\lambda^{(i,j)}(t)) & = & 1 - \frac{v_{thre}}{\lambda(t)a\tau_M} \\ I_S^{(i,j)}(t) & = & \sum\limits_{k=1,k\neq j}^{N_n} \sum\limits_{t_m^{(i,k)}<t} \delta(t-t_m^{(i,k)})w^{(i,j,k)} \end{array} \right. \tag{5.3}$$

Here, $I_E^{(i,j)}(t)$ is the external input from the stimulus with an input rate $\lambda^{(i,j)}(t)$ (see section 5.2.2 for more details); B_t is the Brownian motion; a is the magnitude of excitatory postsynaptic potentials (taking a fixed value of 0.5 mV); $r(\lambda^{(i,j)}(t))$ is the ratio between inhibitory and excitatory inputs (see below for more details). $I_S^{(i,j)}(t)$ is the spiking input from other neurons with connection strengths $w^{(j,k)}$ between the jth neuron and the kth neuron, where $t_m^{(i,k)}$ is the mth spike generated from the kth neuron in the ith column. In the control condition, $I_S^{(i,j)}(t)$ will always be zero. The synapse model (as shown in equation 5.3) simply injects a fixed hyperpolarising current into neurons which have inhibitory connections from the spiking neuron (as shown in equation 5.3). In order to account for the limit of shunting inhibition (where the hyperpolarising effect of inhibitory input is related to the degree of depolarisation caused by excitatory inputs, especially when a neuron is already hyperpolarised to a significant extent; see (Andersen, Dingledine, Gjerstad, Langmoen, & Laursen, 1980)), we have also included a half-wave rectification which ensures that neurons do not become hyperpolarised beyond resting potential. Neurons which reached the threshold level v_{thre} produced a spike and had their membrane potential set to v_{rest}

for the remainder of the simulation step.

It is important to understand the dynamics of the input behaviour. The balancing condition, $r(\lambda^{(i,j)}(t))$ means that the neurons are assumed to have approximately an even ratio of excitatory and inhibitory inputs (note that this does not include the inputs from the inhibitory connections added explicitly by our model), resulting in a zero mean input. This condition highlights the fact that neural noise in vivo tends to scale with input (because as the input stimulus strength is increased, the neural noise level increases, rather than a simple linear increase in the synaptic input), and is required in order to allow an analytic expression of neural firing rates to be possible (as outlined in (Feng & Ding, 2004)). Given the balancing condition, μ provides a constant input to all neurons irrespective of stimulus position and σ effectively sets the variance of the Brownian motion according to how close the stimulus is to the column centre of a given neuron. The result of this is that the magnitude of the input to neurons for which the stimulus is closer to their maximal response position will be greater than that for other neurons.

5.2.2 Input Stimuli

In order to facilitate comparison with the optimal statistical approach outlined in (Rossoni & Feng, 2007), we adopt the same approach to input stimuli. In the experiments presented in section 5.2.4, the stimulus position was either held constant throughout, or moved instantaneously every 100ms, creating a steplike signal. As the former case can be contained within the latter allowing the updated position to be the same at each step, the stimulus position is given as (Rossoni & Feng, 2007):-

$$x(t) = \sum_k \xi_k \chi[t \in \{(k-1)T_W, kT_W\}] \tag{5.4}$$

Here, ξ_k are independent, uniformly-distributed random variables, between limits $[X, Y]$, χ is the indicator function denoting set membership and T_W is the length of time window used for estimating the firing rate. For a constant input position, $X = Y = x$. For the step inputs, the limits are $[0, L]$, where L is the highest maximal-response position for any neuron in the network model. Altogether, equation 5.4

describes an input that remains in a fixed position between 0 and L for T_W time, before moving on to another position chosen at random from between those same limits, where it remains for the same period of time again, and so on until the end of the simulation. Given the stimulus position, $x(t)$, a Gaussian input rate λ is created for the jth neuron in the ith column as follows:-

$$\lambda^{i,j}(t) = \lambda_{core} + c\lambda_{core}\exp(-\frac{(x(t) - x^i_{pos})^2}{2\sigma^2}) \tag{5.5}$$

Here, x^i_{pos} represents the centre of the tuning curve for neurons in the ith column, where their response is at its maximum. There are also three constants in this stimulus creation. σ is the spatial resolution of each neuron, and was set to 1 for all neurons across all experiments. λ_{core} provides both a constant input component, and effectively scales the random element in the balanced input; this was fixed at 3 for all of the experiments. Finally, c scales the stimulus intensity, effectively controlling the extent to which the input is position dependent; this was fixed to 10 for all of the experiments. These parameter values were chosen in order to rescale the input to ensure that the model used realistic neural firing rates.

5.2.3 Decoding Strategy

In order to effectively evaluate the performance of a network model on a given task, it is necessary to be able to convert the network's output back into the domain of the task. In the case of our model using spiking neurons to perform the tracking task, this means we have to address the issue of neural decoding. In their model which performed the same tracking task, (Rossoni & Feng, 2007) explored two statistical decoding methods, one based on moment estimate, and an optimal unbiased strategy using a censored maximum likelihood approach. The latter of these provides a useful benchmark on the best decoding performance possible given a set of particular conditions (including balanced inputs). For our model, however, we prefer to use a decoding strategy that can be interpreted in terms of a neural model. There are two aspects to our decoding strategy:-

1. All of the spikes in the estimation window for each column are simply summed and normalised, and multiplied by their respective column position, to give a value which represents the column's contribution to the position estimate.

2. The resultant set of outputs (one for each column) are further weighted using a simple Gaussian filter (explained below). The stimulus position is estimated to be the mean of the filtered output. This is essentially a centre of mass approach.

The first stage represents the projection of activity from the columns to the next stage of cortical processing. This activity represents a vote for the estimated position of the input to be at the centre of that column's tuning curve; the amount of activity represents a weight on that vote. In other words, a column which is more active will have a greater influence on the position estimate and pull it towards the centre of its tuning curve while less active columns will exert less influence on the estimate; this makes it a centre of mass approach. Given N neurons in each column defined by equations 5.2 and 5.3 outlined earlier, the activation for the ith column will simply be the sum of all the spikes m_j^s from each neuron j in the column in response to input $\lambda^{i,j}(t)$ (defined in the previous section), given by

$$A_i^{raw}(t) = \frac{1}{N} \sum_{j=1}^{N} \sum_{s} \delta(t - m_j^s) \tag{5.6}$$

which is simply the raw population activation equation 5.1.

The second stage represents a relative heightening of the more active outputs compared to the other outputs. Lateral inhibition across columns has been widely interpreted as having this effect; feedback connections, which are prominent between V1 and the LGN, could also act as this sort of sharpening mechanism. The function, which is set up in advance, and is identical for all experiments and conditions outlined in this chapter, is simply a Gaussian centred on the winning column w which therefore has the maximum value; other columns have decreasing values with greater distance from the winning column in accordance with the Gaussian shape. These values are used to weight the column activations, before using them to find final position estimates. The effect of this weighting function is to compensate for artificial edge-effects which exist as a result of a finite number of columns being combined through a centre of mass approach, and results in a more accurate conversion of the neural output back into the domain of the original signal. The Gaussian weighting function for the activation of the ith column gives new weighted outputs

$$A_i(t) = \frac{A_i^{raw}(t)}{\sqrt{2\pi\sigma^2}} \exp(\frac{-(x_{pos}^i - x_{pos}^w(t))^2}{2\sigma^2}) \tag{5.7}$$

where σ is the standard deviation of the filter, which was fixed at a value of 1.5, determined by numerical experiments to give a reliable performance. These weighted outputs are used in turn to provide weighted votes for the position estimate of the stimulus, which are then aggregated to provide the final position estimate as follows:

$$\hat{x}_i(t) = \sum_i A_i(t) x_{pos}^i \tag{5.8}$$

It can be seen that both stages in our decoding strategy have biological plausibility and do not rely upon any abstract, complex statistical calculation. Also in keeping with biological plausibility, estimation was performed online (while the simulation was running), which means that the position estimate was updated and available at each simulation step (rather than being unavailable until the end of each estimation period), allowing us to see the way in which the estimate varied over time. This was achieved using a sliding estimation window representing the memory of the estimation system; only spikes within this window were included. This means that we counted the number of spikes in a certain time window ending at the current time point (and starting at a time point equal to the window size subtracted from current time), and this window slides along the time axis as the simulation progresses, so that it always ranges between the current time point and a previous time point given by the window size subtracted from the current time. This is the standard approach when calculating causal moving averages, and is equivalent to convolution with a low-pass finite impulse response filter. A rectangular sliding window was used in the experiments presented here, which was found to work better than alternative schemes such as KDE for the reasons outlined in section 2.2.1, although different estimation window types could be implemented in the model. The size of the estimation window is a parameter that was varied during the experiments (details given in the next section), in order to assess the interaction of estimation time and inhibition.

5.2.4 Experiments

A number of experiments have been conducted, to examine the effect of inhibitory connections in the network on performance on the stimulus tracking task. Basic performance was measured using the mean squared error (MSE), of the position estimate and the actual stimulus position, at the end of each estimation window period. Although a sliding estimation window was used to produce a position estimate at the end of every simulation step, in calculating the MSE we use only the position at the end of a stimulus period (the period of time during which the stimulus has been in one location), in order to ensure that there are no contamination effects from previous stimulus position adversely affecting the current position estimate. In addition to examining basic performance, correlation measures (both across and within columns, and estimate autocorrelation), and firing rates have been shown where appropriate. We also vary a number of other parameters in the experiments, including:-

- The stimulus type: both constant and steplike signals are used.

- The size of the estimation window: values from 100ms right down to just 10ms are tested.

- The number of neurons in each column: set to 100 by default, but varied in one experiment to examine the effect on central limit convergence of neuron numbers.

- The firing threshold V_{thre}: set to 5 by default for the network with lateral inhibitory input, and 20 for the network without, to ensure similar firing rates and therefore a balanced input for both, but varied in one experiment to have the same threshold value of 20 for each.

- The inhibitory connection strength $w^{(i,j,k)}$ between the jth and kth neurons in the ith column. (set to -1 by default, but all values from 0 to -1 tested in one experiment).

The results of all of these experiments are presented in the next section.

5.3 Results

5.3.1 Basic Performance

The performance of the network was first tested on a stimulus with a constant position of 4.5. It can be seen in figure 5.2 that when lateral inhibitory connections are used, the MSE is much lower, and the network's online position estimate is much more stable. As the estimation window size becomes smaller, the stimulus tracking task becomes more difficult, and this is reflected in the increased MSE for the smaller window sizes. It is notable, however, that even for small window sizes, the network with inhibitory connections performs quite well, giving a performance with a 10ms estimation window slightly better than that of the network with no inhibitory connections and a 50ms estimation window. This highlights the very significant benefits that inhibitory connections within a pool of neurons can bring in situations where fast estimation is required, which would be expected for an evolved sensory processing system. Figure 5.3 shows the firing rates for all of the neurons in the network (in this case with inhibitory connections), demonstrating that neurons in the columns whose maximal response properties are closest to the stimulus position have the highest firing rates, but also emphasising the noisy nature of neural firing while performing the tracking task.

Figures 5.4 and 5.5 show the same results for a random steplike signal which moves position instantaneously every 100ms. As for the constant stimulus, the network with inhibition significantly outperforms the network with no lateral connections, and the difference, not that apparent at 100ms, becomes increasingly obvious as the estimation window size decreases. The relationship between estimation window size and MSE for networks with and without inhibitory connections is shown explicitly in figure 5.6. It is worth noting from the results shown in these figures, that while lateral inhibition leads to considerably improved estimation performance, it does also seemingly slow down estimation. This is because negative correlation acts as a damper on the system, while the absence of it, or even positive correlation, acts as a reinforcement of an existing pattern of activation, thereby providing stronger inputs which cause the neurons to overcome the inertia of their membrane time constant more rapidly. The better estimation of a dampened feedback system takes

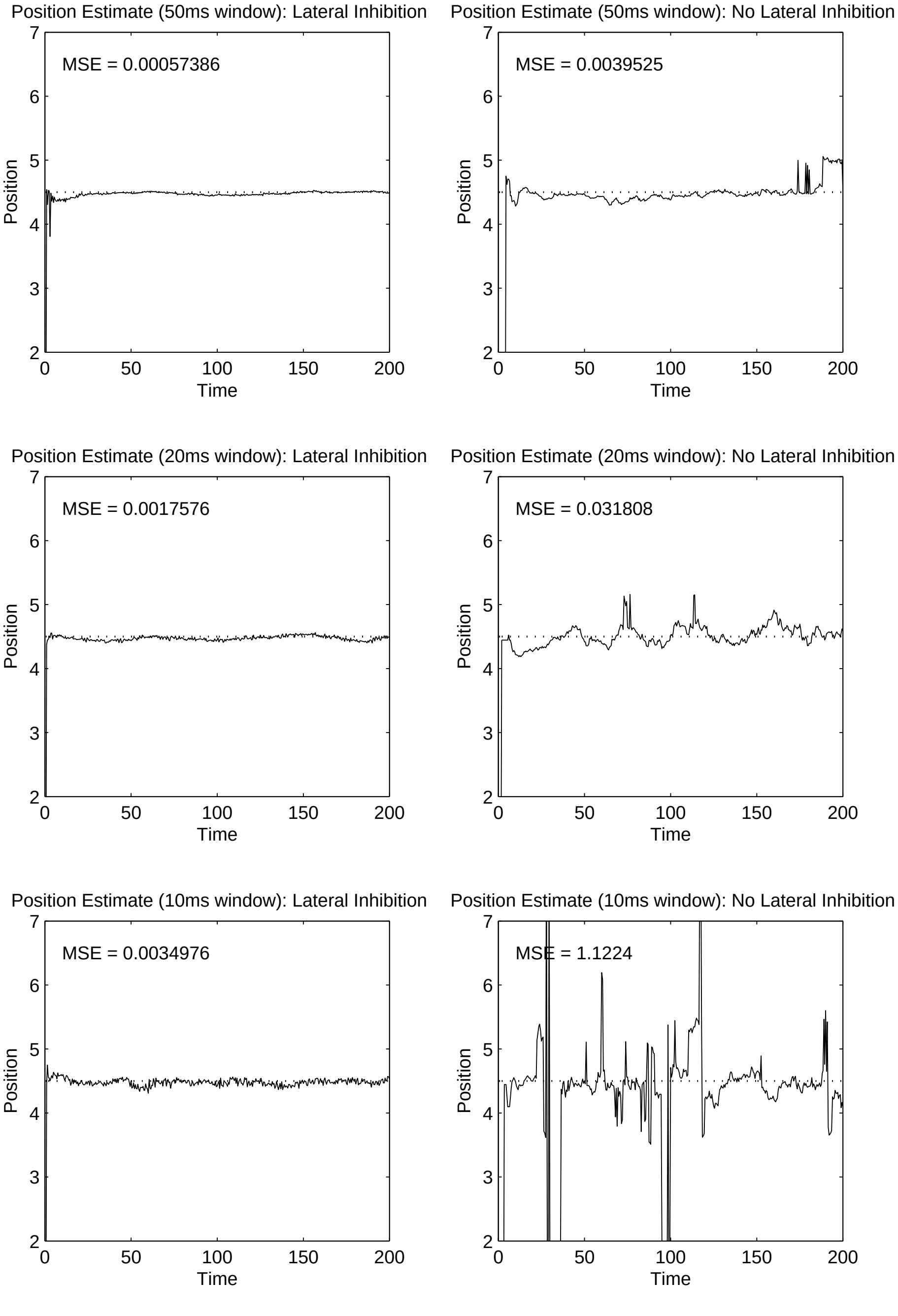


Figure 5.2: Position targets (dotted line) and estimates (solid line) for a constant signal, across 50ms, 20ms and 10ms estimate window sizes (shown here in different rows). *Left column*: Lateral inhibition used. *Right column*: Lateral inhibition not used. It can be seen that the MSE is lower when lateral inhibition is present, across all window sizes. The greater reliability of the position estimate when neurons have inhibitory connections is particularly apparent for smaller estimation window sizes, when the estimation task becomes more difficult.

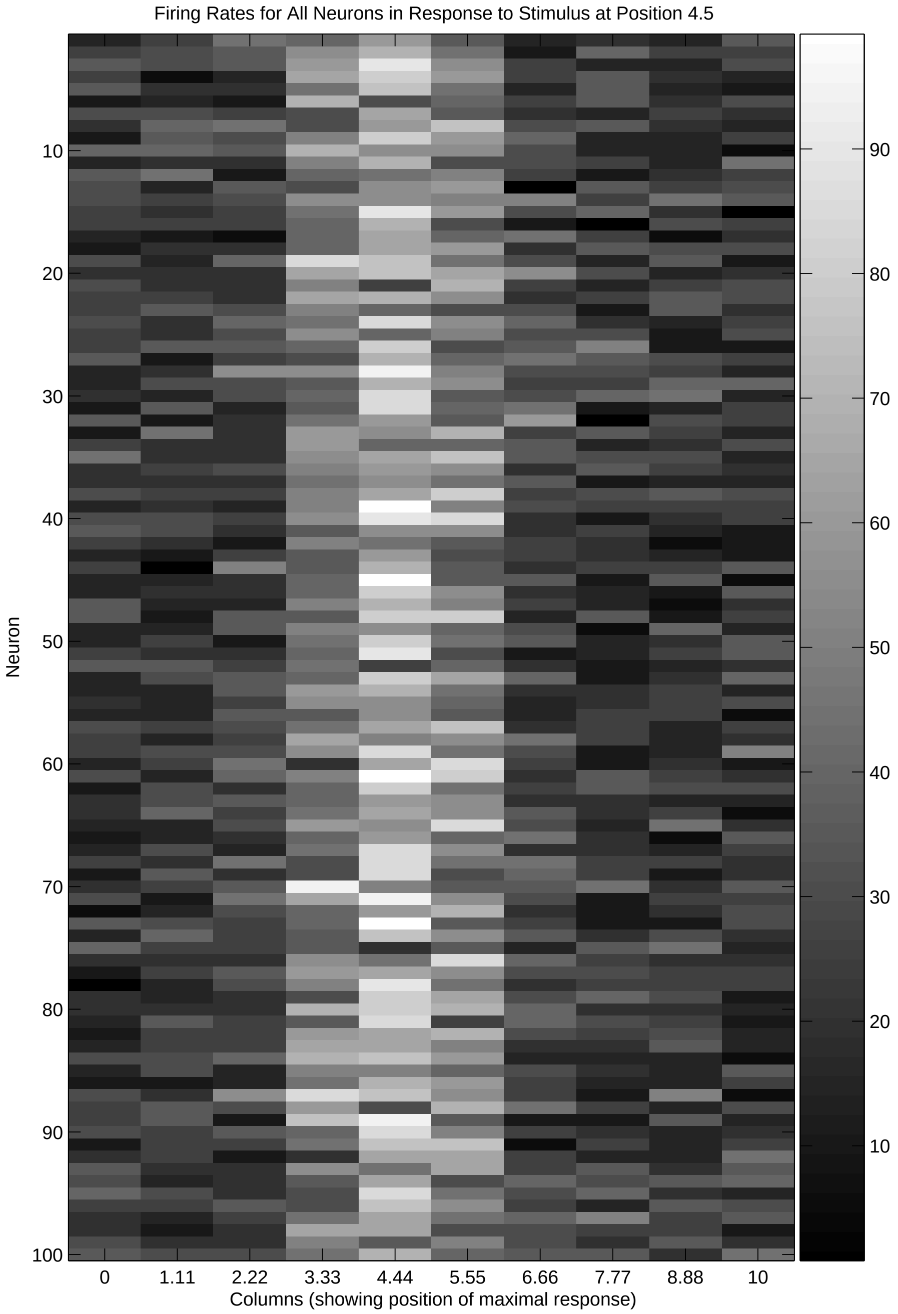


Figure 5.3: The mean firing rate for each neuron, shown here using a grayscale index, with higher firing rates shown by lighter shades. The neurons in the columns which have maximal response near to the constant stimulus position of 4.5 show higher firing rates, as expected. The variability in individual firing rates even within the same column is also apparent here.

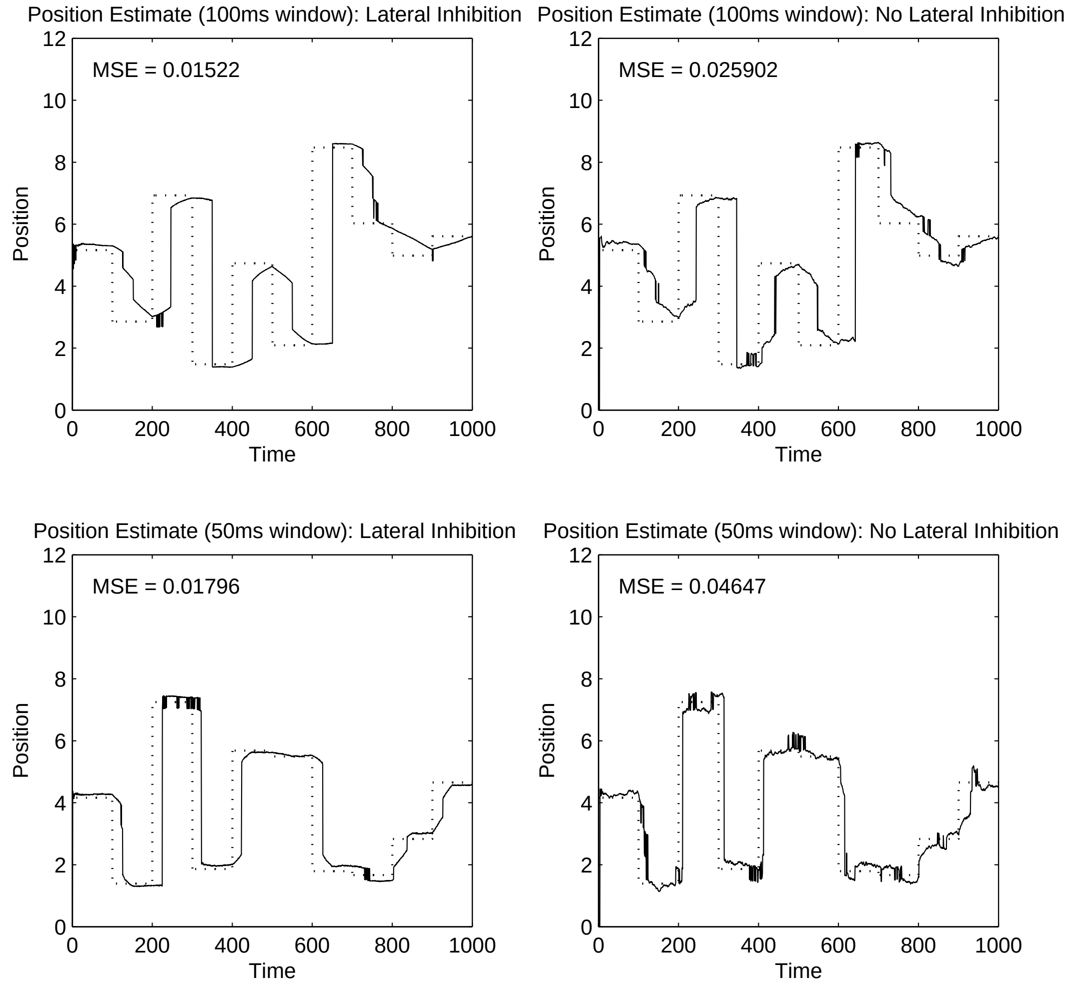


Figure 5.4: Position targets (dotted line) and estimates (solid line) for a random steplike signal, across 100ms and 50ms estimate window sizes (shown here in different rows). *Left column*: Lateral inhibition used. *Right column*: Lateral inhibition not used. It can be seen that the MSE is lower when lateral inhibition is present, across all window sizes.

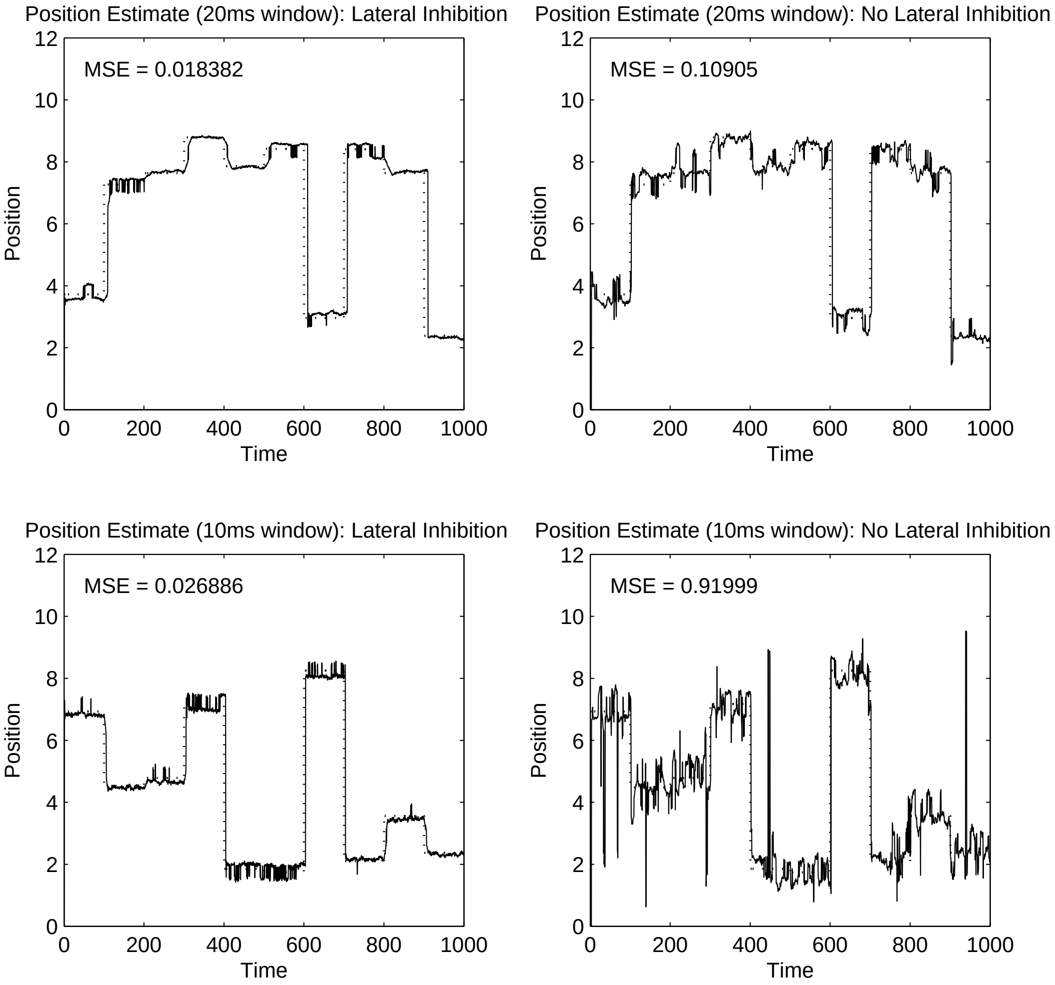


Figure 5.5: Position targets (dotted line) and estimates (solid line) for a random steplike signal, across 20ms and 10ms estimate window sizes (shown here in different rows). *Left column*: Lateral inhibition used. *Right column*: Lateral inhibition not used. It can be seen that the MSE is lower when lateral inhibition is present, across all window sizes. The difference is greater for these smaller estimation windows than for the larger estimation windows in the previous figure.

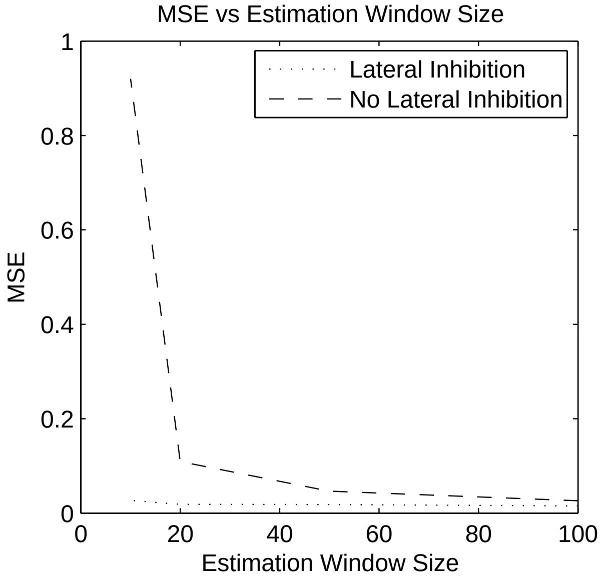


Figure 5.6: The effect on the MSE of estimation window size. The increased difficulty of the estimation task given less time can be seen here. Performance for the network without inhibitory connections drops quite considerably for small window sizes, while the presence of inhibitory connections helps the network to maintain a robust performance at these smaller sizes.

more time. Nevertheless, as the results in figure 5.6 show clearly, the beneficial effect scales very well with decreased estimation window sizes (because there is less inertia to overcome), so the longer estimation time seems unlikely to pose any serious problem for the information processing capabilities of the system.

5.3.2 Correlation Measures

It is useful to look more closely at one specific simulation, and in particular several different measures of correlation. Figure 5.7 gives the results of a simulation (of both a network with inhibitory connections, and one without) using default values and a 50ms estimation window. The top row of figure 5.7 shows that, in keeping with the results of the previous section, the MSE for a network with inhibition is much lower than that of a network with no inhibition. The difference is qualitatively characterised by a smoother estimation curve for the inhibitory network, reflecting its greater reliability. This difference can be measured quantitatively by the autocorrelation at low lags of the position estimates for each network, which are shown in the left-middle graph; the greater autocorrelation for the inhibitory network reflects the smoother, more reliable, curve obtained.

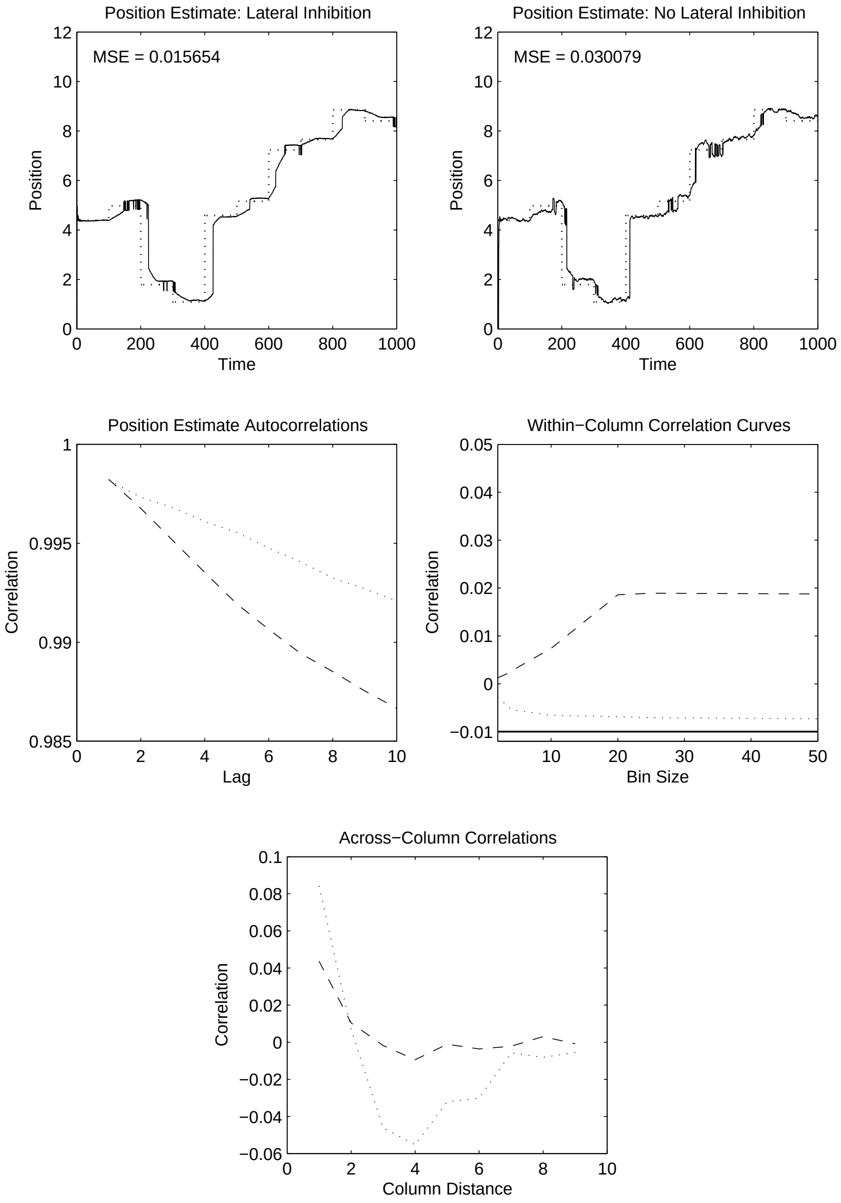


Figure 5.7: A detailed example using a 50ms estimate window. *Top*: Position targets (dotted line) and estimates (solid line) for a random steplike signal. The better performance with lateral inhibition present can be seen. *Middle left*: Position estimate autocorrelations. These give an indication of the reliability of the signal by measuring short-term fluctuations in the estimate, and show the greater reliability of the estimate when lateral inhibition is used (dotted line). *Middle right*: Correlation curves of firing patterns for neurons in the same column. The use of inhibitory connections results in negatively correlated firing patterns (dotted line). The solid line at the bottom represents the lowest possible correlation for a network of that size. *Bottom*: Mean correlation across columns as a function of column distance. This shows that neurons in columns close to each other will tend to respond in a similar way as expected. Neurons in columns at medium distances tend to respond in quite different ways, giving a 'Mexican hat' shape, which is accentuated by the presence of inhibitory connections (dotted line).

Of central importance to the understanding of lateral inhibitory connections in a column of neurons is the concept of firing rate correlation as introduced previously in section 2.2.1. This is a measure of the average correlation between any two neurons within the column, essentially reflecting whether they fire together (synchronisation), avoid firing together (anti-synchronisation), or operate independently (no synchronisation). It is calculated by first taking the number of spikes emitted by a neuron in a period of time given by the bin size, for all such periods during the simulation, which gives a series of momentary firing rate values for each neuron in the column. The correlation between these values is then measured, giving a correlation matrix which shows the correlation between all pairs of neurons in the column, and the average value is calculated to give a mean correlation value for that column. This procedure is repeated for each column, and finally the overall average correlation value is taken from these.

We have previously shown that when the firing rates of neurons are negatively correlated (that is, when one neuron fired more and other neurons correspondingly tended to fire less), the benefit of convergence to a central limit of pooled activity is enhanced. One mechanism for these firing rates to be achieved is if the spike trains of the neurons themselves are negatively correlated, which means that when one neuron fires, it reduces the probability of other neurons firing. Clearly, lateral inhibition might be expected to lead to this behaviour. The right-middle graph of figure 5.7 shows correlation curves for the networks with and without lateral inhibition. These are calculated by measuring the number of spikes within short time bins (and shown across a range of bin sizes in order to ensure that the results are not an artefact of bin size), and evaluating the correlation of these with each other as described above; the mean correlation value is shown. The graph also shows a baseline value (solid line), which is the lowest possible mean correlation value, a constraint resulting from the number of neurons in a column. It is clear from the graph that inhibitory connections have resulted in negatively correlated firing patterns emerging from the network, and without these, the firing patterns are slightly positively correlated.

In addition to measuring the correlation of neurons within columns, it is of interest to evaluate the correlation across columns, given that column location is directly related to neural response properties (neurons in columns close to each other have similar response properties). The mean correlation between columns is shown at the bottom of figure 5.7. It is clear from this that neurons in columns close to each other tend to respond in similar ways as expected, and also that neurons in very distant columns tend to be independent of each other. Interestingly, neurons a medium distance away show a negative correlation, indicating that they tend to respond at different times. We believe that this is because neurons at longer distances actually have a negative correlation component when either are particularly active, but a positive correlation component when columns centrally in between them are more active and as a result they are both less active at the same time. Columns a middle distance apart do not have as strong a positive correlation component (since there are fewer columns in between that could become active), and as a result appear to be more negatively correlated overall. It is also interesting to observe that the correlation curve is more accentuated for the network with inhibitory connections than the one without, indicating a smoother harmonic behaviour by that network.

It can be seen from the above discussion of the mechanism of lateral inhibition within the columns in the model that the focus of interest is on negative correlation in the firing patterns between neurons. As we saw in section 2.2, as well as these instantaneous network correlations across different spike trains, there are potentially correlations within spike trains that may also aid information transmission. If this is the case, it should show up in the autocovariance of the ISIs at low lags (especially lag 1, i.e. between successive intervals). However, our results averaged over 100 simulations revealed no consistent pattern of ISI autocorrelation for any of the neurons in the network. This is in fact not surprising given that the majority of neurons each fire at most once during the estimation period of the sliding spike count window. Furthermore, there was no significant difference in the autocovariance of the ISIs between the two networks. In combination with the very short estimatation window used (rendering longer-term spike train statistics irrelevant in this context), this shows clearly that the improved performance of the network with lateral inhibitory neurons is not due to any systematic changes in the spike train statistics of individual neurons, and in particular is not due to negatively correlated ISIs in the way that (Chacron et al., 2004; Lindner et al., 2005) have discussed. Rather, our results appear to be entirely due to the lateral inhibitory feedback causing neurons to have negatively correlated firing patterns relative to each other, rather than relative to their own past or future firing.

5.3.3 Threshold, Firing Rate and Inhibitory Connection Strength

As described in section 5.2.2, the inputs have been balanced, in order to allow comparison with (Rossoni & Feng, 2007). However, this is not in itself a requirement for our network, since we are not using the analytical expression for neural firing rates dependent upon this, given in (Feng & Ding, 2004). It is also the case that the network with inhibitory connections obviously has an additional source of input to each neuron (lateral inhibitory inputs from other spiking neurons) beyond the external stimulus input. As a result, given the same input strength and firing threshold, neurons in the inhibitory network will have a lower firing rate. This is shown in the top row of figure 5.8. It should be noted that in this situation, the improved performance of the network with inhibitory connections is evident. In order to rule out the difference in firing rates as a possible factor, and to simulate balanced inputs, the firing rate threshold of neurons in the inhibitory network was reduced to a level that would allow a firing rate approximately the same as that of the non-inhibitory network (the default situation). It should be noted that this is simply a scaling operation, equivalent to increasing the scale of the inputs or simulating background excitatory connections, for example, and in no way changed the dynamics of the model. The middle row of figure 5.8 confirms that the inhibitory network still has a much better performance than the non-inhibitory network, showing that irrespective of whether the firing threshold or the firing rates are the same, the inhibitory connections still allow the network to perform better.

The relationship between inhibitory connections and within-column correlations is closely linked to the relationship between inhibitory connections and firing rates. We have already seen in section 5.3.2 that inhibitory connections lead to negatively correlated firing patterns. The bottom of figure 5.8 shows that as the connections become more inhibitory, the correlation duly becomes more negative, as we would expect if the inhibitory connections were responsible for the negative correlation.

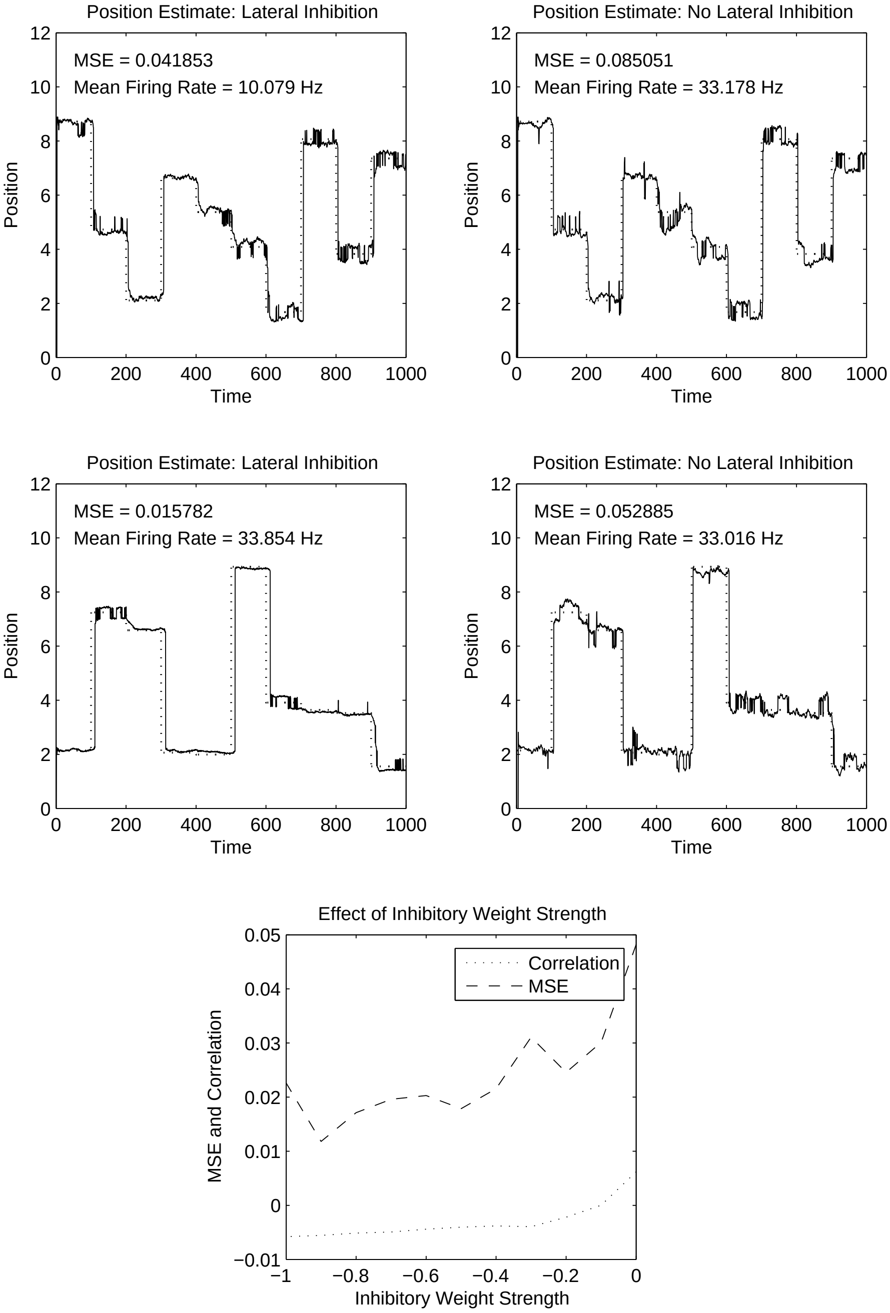


Figure 5.8: The effects of threshold and firing rate. *Top*: Position targets (dotted line) and estimates (solid line) for a random steplike signal using a 20ms estimate window, with the same firing threshold for each ($v_{thre} = 20$). *Middle*: Position targets (dotted line) and estimates (solid line) for a random steplike signal using a 20ms estimate window, with approximately the same firing rate. *Bottom*: The effect of changing the inhibitory connection strength on MSE and correlation. Increasing the strength of the inhibitory connections decreases the MSE, and reduces the correlation in the firing patterns of neurons within columns.

It also shows that the MSE tends to decrease as the connections become more inhibitory, which conforms with our earlier results.

5.3.4 Number of Neurons

Our hypothesis, described earlier in section 5.1.3, is that population coding benefits from the central limit theorem, and that by having negatively correlated firing patterns, that benefit is enhanced. We have already seen that negatively correlated firing patterns arise from inhibitory connections and improve performance. In order to assess the initial central limit effect, we tested our model with different numbers of neurons. Figure 5.9 top row shows the performance of networks with and without inhibitory connections, with 100 neurons in each column. The middle row shows the performance with just 10 neurons per column. It can clearly be seen that with fewer neurons, the performance is much worse. The bottom graph shows the central limit effect clearly across a range of neuron population sizes. It also shows that across all sizes, the presence of inhibitory connections, with the associated negative correlation, improves performance.

5.4 Biophysical Models

5.4.1 Synapse Model

The model we presented in the previous section is very simple, in order to allow a useful evaluation of the lateral inhibitory mechanism in a controlled manner, without the additional complexity of more realistic neural dynamics. In particular, we have used a Dirac function for our synapse model, with no reversal potential, in order to be able to control the inhibitory input by making it free of temporal dynamics and independent of the state of the neuron. However, it is also necessary to use a much more biophysically realistic synapse model in order to ensure that the results presented so far are relevant for real neural systems. In the experiments described in this section, we use kinetic synapses implemented as Markov models, with the biophysically-realistic parameter values given in (Destexhe et al., 1998, 1994), as previously introduced in section 2.1.4. The excitatory synapses used in section 5.4.4 are fast AMPA synapses, and the inhibitory synapses used in all the biophysical

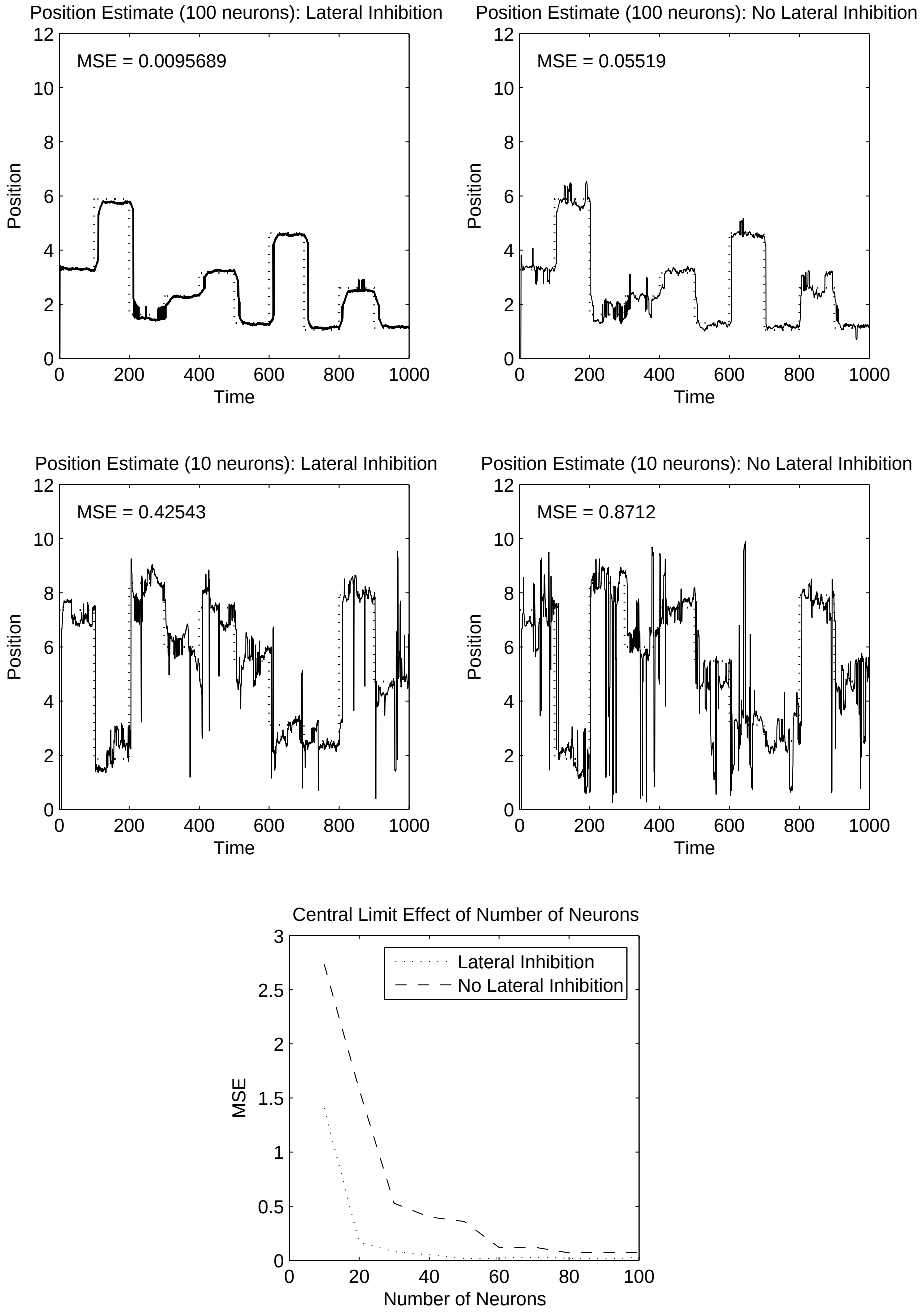


Figure 5.9: The effect of neuron numbers. *Top*: Position targets (dotted line) and estimates (solid line) for a random steplike signal using a 25ms estimate window, with 100 neurons in each column. *Middle*: Position targets (dotted line) and estimates (solid line) for a random steplike signal using a 25ms estimate window, with 10 neurons in each column. *Bottom*: The effect of changing the number of neurons on the MSE when inhibitory connections are present (dotted line) and not present (dashed line). The central limit effect can be seen clearly in both cases, where increasing the number of neurons reduces the MSE. In addition the MSE is consistently lower when inhibitory connections are present irrespective of the number of neurons.

model experiments are fast GABA$_A$ synapses; we repeat the equations here for convenience. These are combined with the standard integrate-and-fire neuron model (Feng, 2004), also shown again here, giving us the following:-

$$c_M \frac{dv^{(i,j)}(t)}{dt} = -g_L(v^{(i,j)}(t) - E_L) + \frac{I_E^{(i,j)}(t)}{A_M} - g_{AM}^{(i,j)}(t)(v^{(i,j)}(t) - E_{AM}) - g_{GA}^{(i,j)}(t)(v^{(i,j)}(t) - E_{GA}) \tag{5.9}$$

where we have

$$\begin{cases} g_{AM}^{(i,j)}(t) &= \sum_{k=1,k\neq j}^{N_n} \bar{g}_{AM} P_{AM}^{(i,j,k)}(t) w^{(i,j,k)} \\ g_{GA}^{(i,j)}(t) &= \sum_{k=1,k\neq j}^{N_n} \bar{g}_{GA} P_{GA}^{(i,j,k)}(t) w^{(i,j,k)} \\ \frac{dP_{AM}^{(i,j,k)}(t)}{dt} &= \alpha_{AM}(T)(1 - P_{AM}^{(i,j,k)}(t)) - \beta_{AM} P_{AM}^{(i,j,k)}(t) \\ \frac{dP_{GA}^{(i,j,k)}(t)}{dt} &= \alpha_{GA}(T)(1 - P_{GA}^{(i,j,k)}(t)) - \beta_{GA} P_{GA}^{(i,j,k)}(t) \\ T &= 1 \quad \text{for} \quad (t - t') \leq 1 \\ T &= 0 \quad \text{for} \quad (t - t') > 1 \end{cases} \tag{5.10}$$

Here, $g_{AM}^{(i,j)}(t)$ is the total AMPA (AM) synaptic current going into the jth neuron in the ith column at time t, $P_{AM}^{(i,j,k)}(t)$ is the fraction of AMPA receptors open for the synapse between the jth and kth neurons in the ith column at time t, and analogously for the GABA$_A$ synapses (GA). T represents a square pulse of transmitter concentration in the cleft, which occurs from the time of the most recent presynaptic spike, t', for a duration of 1ms. This causes receptors to open in accordance with the kinetic equations shown, with opening rates $\alpha_{AM} = 1.1$ mM^{-1}ms^{-1} and $\alpha_{GA} = 5$ mM^{-1}ms^{-1}. Open receptors are continually closing at rates of $\beta_{AM} = 0.19$ ms^{-1} and $\beta_{GA} = 0.18$ ms^{-1}. The contribution to the total postsynaptic current from each synapse consists of three components: the maximum conductance values ($\bar{g}_{AM} = 0.7$nS and $\bar{g}_{GA} = 0.8$nS), the proportion of open receptors as discussed previously, and the synaptic efficacy $w^{(i,j,k)}$, which is set to an identical value for every synapse ($w^{(i,j,k)} = 50$ for the experiments here; this value was chosen to allow neurons to fire with realistic rates, and reflects the fact that there are only a limited number of synaptic inputs in the model; it can be regarded as a scaling parameter). Finally, the total synaptic current for each synapse type is modulated by the distance between the current membrane potential and the reversal potential for the synapse ($E_{AM} =$

0mV and E_{GA} = -80mV; in practice these were rescaled to E_{AM} = 65mV and E_{GA} = -15mV in order to bring them in line with the zero resting potential approach adopted throughout this work). See section 2.1.4 for more details of these synapse models and parameters.

The leaky integrate-and-fire neuron of eq.5.9 has a resting potential of $V_{rest} = -65mV$, rescaled to V_{thre} = 0mV, a firing threshold of V_{thre} = -50mV, rescaled to V_{thre} = 15mV, a leakage channel with a reversal potential of E_L = -65mV rescaled to 0mV, a specific membrane capacitance c_M = 10nf/mm^2, a specific membrane resistance of r_M = 2MOhms mm^2, and a membrane area of A_M = 0.1 mm^2. This gives a membrane time constant of τ_M = 20ms and a total membrane capacitance of C_M = 1nF, and allows us to rewrite eq.5.9 in order to more clearly show the relationship between this biophysical model (including the rescaled threshold potential) and Stein's model outlined earlier (compare with eqs.5.2 and 5.3):-

$$\frac{dv^{(i,j)}(t)}{dt} = -\frac{v^{(i,j)}(t)}{\tau_M} + \frac{I_E^{(i,j)}(t)}{C_M} + I_S^{(i,j)}(t) \tag{5.11}$$

$$\begin{cases} I_E^{(i,j)}(t) &= \mu(\lambda^{(i,j)}(t)) + \sigma(\lambda^{(i,j)}(t))B_t \\ \mu(\lambda^{(i,j)}(t)) &= a\lambda^{(i,j)}(t)(1-r) \\ \sigma(\lambda^{(i,j)}(t)) &= a\sqrt{\lambda^{(i,j)}(t)(1+r)} \\ r(\lambda^{(i,j)}(t)) &= 1 - \frac{v_{thre}}{\lambda(t)a\tau_M} \\ I_S^{(i,j)}(t) &= \frac{(-g_{AM}^{(i,j)}(t)(v^{(i,j)}(t) - E_{AM}) - g_{GA}^{(i,j)}(t)(v^{(i,j)}(t) - E_{GA}))}{c_M} \end{cases} \tag{5.12}$$

All of the stimulus inputs and parameters are identical to the model presented in section 5.2. The membrane equation is also the same; the only difference is in the form of the synaptic inputs $I_S^{(i,j)}(t)$, which are now based on the more biophysically realistic kinetic models as described above. In the following sections, we present the results of some simulations based on this model.

5.4.2 Basic Performance

We first examine the basic performance of the model with more biophysically realistic synapses as described in section 5.4. Figure 5.10 shows the results of a typical

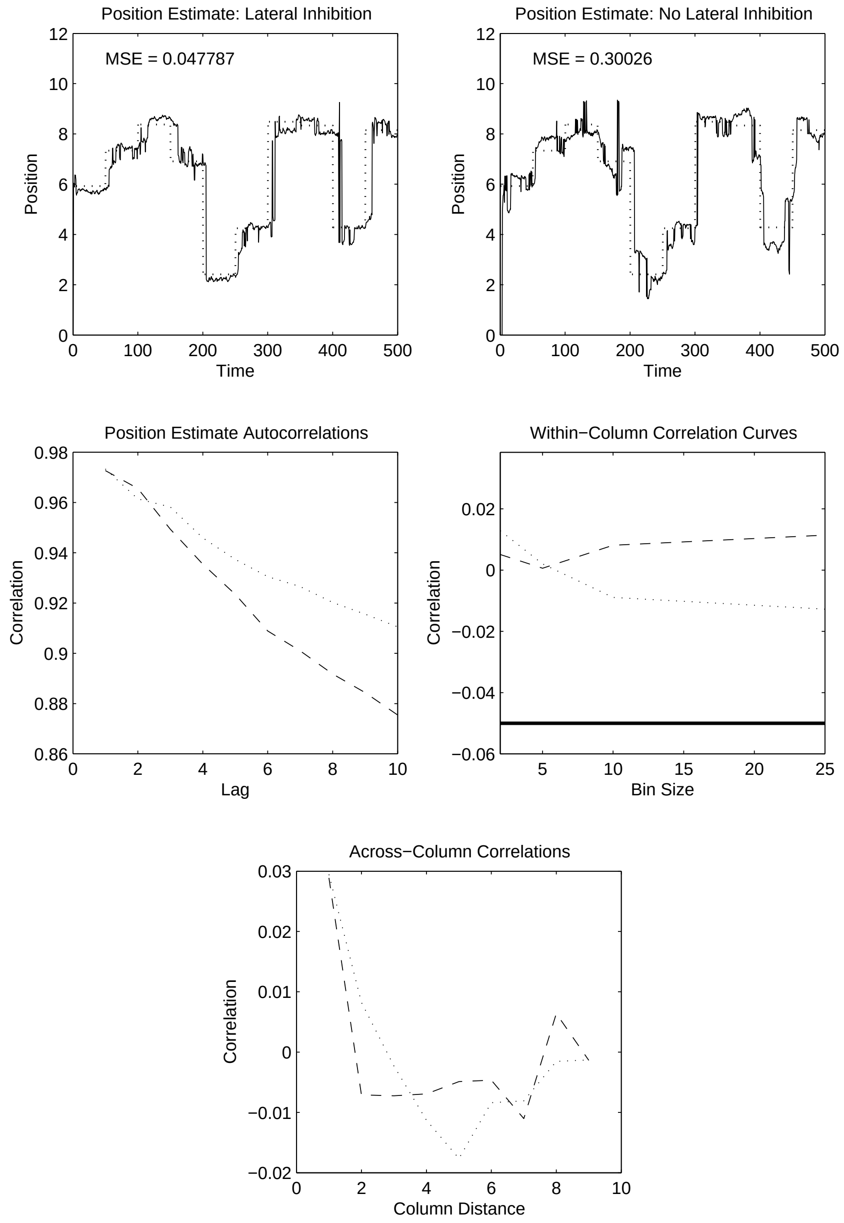


Figure 5.10: Basic performance of the biophysical model using a 50ms estimation window. *Top*: Position targets (dotted line) and estimates (solid line) for a random steplike signal. The better performance with lateral inhibition present can be clearly seen. *Middle left*: Position estimate autocorrelations. These give an indication of the reliability of the signal by measuring short-term fluctuations in the estimate, and show the greater reliability of the estimate when lateral inhibition is used (dotted line). *Middle right*: Correlation curves of firing patterns for neurons in the same column. The use of inhibitory connections results in negatively correlated firing patterns (dotted line). The solid line at the bottom represents the lowest possible correlation for a network of that size. *Bottom*: Mean correlation across columns as a function of column distance. This shows that neurons in columns close to each other will tend to respond in a similar way as expected. Neurons in columns at medium distances tend to respond in quite different ways, giving a 'Mexican hat' shape, which is accentuated by the presence of inhibitory connections (dotted line).

run, and can be considered analagous to figure 5.7. The graphs in the top row of this figure give the central result: the MSE of the network with lateral inhibitory connections is considerably lower than that of the network with no lateral connections. The networks are identical in every other regard. This result is in keeping with the results obtained using the simpler synapse model. The autocorrelation graph also confirms this result, showing more rapid changes (high frequency noise) for the network with no inhibitory connections, reflecting the greater variability of the mean firing rate while the same stimulus is being presented. Similarly, we observe the same negative correlation between the spike trains of neurons with inhibitory connections, but not in those without such connections. Similarly, we observe the same shape in the across-column correlation graphs for this new model as for the previous model, because the same stimulus-response dynamics apply to both models. Overall, we can see that the performance of the two models is very similar; it appears that the biophysically realistic synapses do not change the fundamental dynamics of the lateral inhibitory mechanism.

It should also be emphasised that this behaviour is completely typical and can be reproduced with high reliability. Figure 5.11 shows the MSE and correlation figures for 100 simulation runs. It can be seen that in every single run, for the network with lateral inhibitory connections the MSE is lower and the mean correlation (for the largest bin size) is negative, while the network without lateral inhibitory connections always demonstrates worse performance (higher MSE) and a higher mean correlation value.

5.4.3 Membrane Time Constant

As outlined in sections 2.1.4 and 5.4.1, the central difference between the kinetic synaptic model used in the biophysical model and the instantaneous current injection of the simple model is that the former has a time trajectory (the effect of which is subject to further variation dependent on the state of the postsynaptic membrane potential). The effect of the synapse is not limited to the time at which the presynaptic spike takes place, but increases to a peak over a short period of time , and decays over a longer period. The time course of the AMPA and GABA$_A$ synaptic currents is shown on the left of figure 5.12. It is apparent that most of

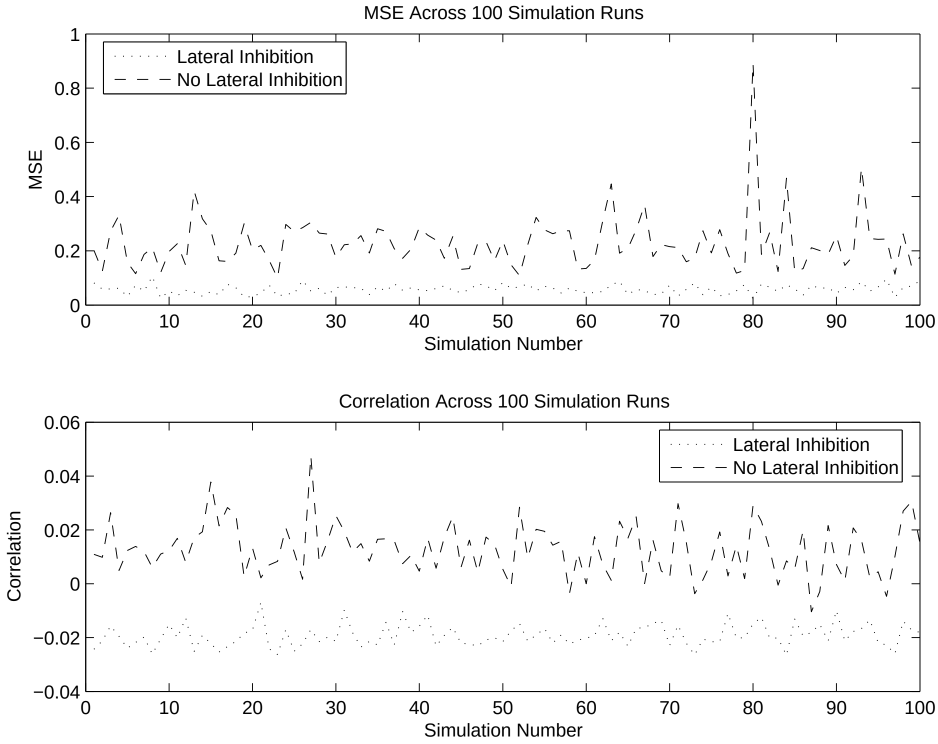


Figure 5.11: Results from the biophysical model across 100 simulation runs. *Top*: The MSE for the network with (dotted line) and without (dashed line) lateral inhibitory connections. In all cases, the MSE was lower for the network when lateral inhibitory connections were included. *Bottom* The correlation for the network with (dotted line) and without (dashed line) lateral inhibitory connections. In all cases, the correlation was negative when lateral inhibitory connections were included, and was always lower than the network with no lateral connections.

the activity takes place over the first 20ms, for both synapse types. However, this is also modulated by the rate at which external inputs affect the trajectory of the membrane potential. By using integrate-and-fire neurons, we have the opportunity to vary this rate - the membrane time constant - and thus examine the effect it has on the performance of the networks when using realistic synapse models.

The right-hand side of figure 5.12 shows the MSE as a function of the membrane time constant for two different estimation window sizes. It is immediately apparent that lower membrane time constants give better performance. This result is not surprising; a lower membrane time constant allows both inhibitory lateral inputs (where present), and the stimulus inputs, to be incorporated into the membrane potential more rapidly, and thus allows a more accurate reflection of these inputs in the firing rate within a short period of time. The better performance at 50ms than 50ms shown in this graph reflects the general improvement in performance given a

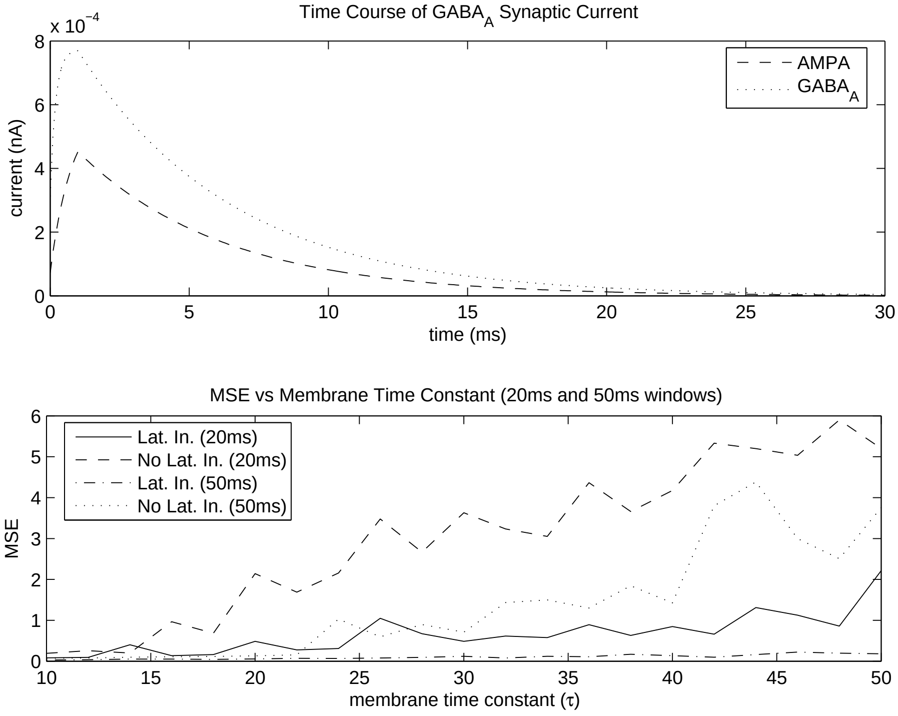


Figure 5.12: *Top*: Time course of AMPA and GABA$_A$ synaptic currents, given a single presynaptic spike at time=0. Most of the activity takes place over the first 20ms. *Bottom*: MSE from simulation runs of the biophysical model with varying membrane time constants. Lower membrane time constants allow synaptic inputs to be assimilated into the membrane potential more rapidly, and thus enhance performance on this time-critical task. More time is available when the estimation window is 50ms rather 20ms, and which has correspondingly better performance.

longer period over which the momentary firing rate is estimated (shown previously in figure 5.6).

5.4.4 Mixing Excitatory and Inhibitory Connections

All of the experiments outlined so far have focused on the difference between a network in which neurons in each column have lateral inhibitory connections to all other neurons within the column, and a network with no lateral connections. In this section, we examine the effect of replacing some of the lateral inhibitory connections with excitatory connections (the replacement rather than addition is in order to avoid the biophysically implausible situation of the same neurons giving both excitatory and inhibitory connections). With 100 neurons per column, there are therefore 99 lateral connections per neuron (our neurons do not have recurrent connections to themselves). The results shown in the left panel of figure 5.13 show an interesting effect: as the number of excitatory connections increases, there is no

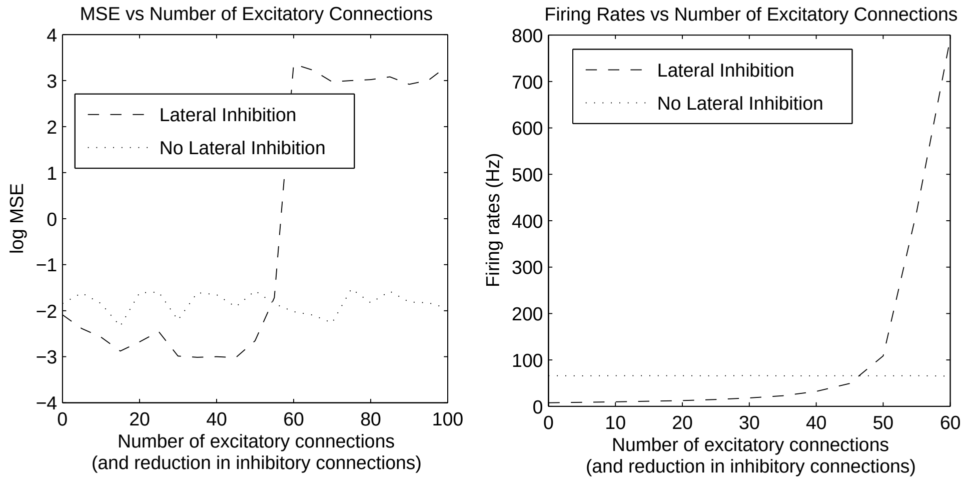


Figure 5.13: Adding excitatory lateral connections. *Left*: MSE as a function of the number of excitatory connections (and a corresponding decrease in the number of inhibitory connections); shown here using a logarithmic scale for clarity of display. Performance is relatively static until the point where the number of excitatory connections exceeds the number of inhibitory connections, after which the MSE rapidly rises. *Right*: Firing rate as a function of the number of excitatory connections (and a corresponding decrease in the number of inhibitory connections). The firing rate is relatively low (and in a biophysically plausible range) until the number of excitatory connections exceeds the number of inhibitory connections, after which it rapidly grows without bound.

obvious decrease in performance, which remains better than that of the network with no lateral connections, until a critical point is reached when the number of excitatory connections begins to exceed the number of inhibitory connections, at which time performance catastrophically falls off. There are two main contributing factors in this behaviour. The improved performance of the network with a greater number of inhibitory than excitatory connections is due to the beneficial effect of lateral inhibition, which is the main topic of this chapter and has been seen in all of our previous experiments. On its own, we would expect this to lead to a somewhat more smoothly linear function than we see in the figure. The second major factor is the feedback effect from the excitatory connections, whereby a spiking neuron adds an excitatory input to other neurons in the same column, which are in turn more likely to spike and send further excitatory inputs to the neurons in the column, including the original neuron. This feedback mechanism is inherently unstable, and can lead to large increases in the firing rates. By contrast, the lateral inhibitory connections are inherently stable, since they seek to suppress the very activity that triggers them. While there are significantly more inhibitory than excitatory connections, the stabilising effect of the inhibitory connections allows the network to benefit from the higher firing rates caused by the excitatory connections, which is why performance actually slightly improves rather than linearly decreasing, without spiralling out of control. After reach the critical point, however, the system breaks, firing rates start to rise without bound, and performance catastrophically drops off. The right panel of figure 5.13 confirms the firing rate effect. These results confirm that excitatory lateral connections can be combined with inhibitory connections, without undermining their benefit, as long as the inhibitory connections remain in the majority, ensuring that the firing rate does not grow too much.

5.4.5 Hodgkin-Huxley Neurons

In addition to using more biophysically realistic synapse models, it is also possible to use a more biophysically realistic neuron model. The integrate-and-fire model is widely used because it is both computationally straightforward, allows specific control over some parameters such as firing threshold and membrane time constant, and and is analytically tractable. However, this comes at a certain price of realism, and although it offers a reasonable approximation to real neurons, there can be significant differences. It is therefore useful, where possible, to test the central aspect of a model that is based on integrate-and-fire neurons on a more biophysically realistic neuron model, such as the Hodgkin-Huxley model. We tested our model using Hodgkin-Huxley neurons, as introduced in section 2.1.3. We rescaled the equations to zero-resting potential in this version, and present this slightly modified version here for convenience:-

$$\left\{\begin{array}{rcl} c_M \dfrac{dv^{(i,j)}(t)}{dt} & = & -I_M^{(i,j)}(t) + \dfrac{I_E^{(i,j)}(t)}{A_M} - I_S^{(i,j)}(t) \\ I_M^{(i,j)}(t) & = & \bar{g}_L(v^{(i,j)}(t) - E_L) + g_K^{(i,j)}(t)(v^{(i,j)}(t) - E_K) + g_{Na}^{(i,j)}(t)(v^{(i,j)}(t) - E_{Na}) \\ I_E^{(i,j)}(t) & = & \mu(\lambda^{(i,j)}(t)) + \sigma(\lambda^{(i,j)}(t))B_t \\ I_S^{(i,j)}(t) & = & g_{AM}^{(i,j)}(t)(v^{(i,j)}(t) - E_{AM}) + g_{GA}^{(i,j)}(t)(v^{(i,j)}(t) - E_{GA}) \end{array}\right. \tag{5.13}$$

where we have

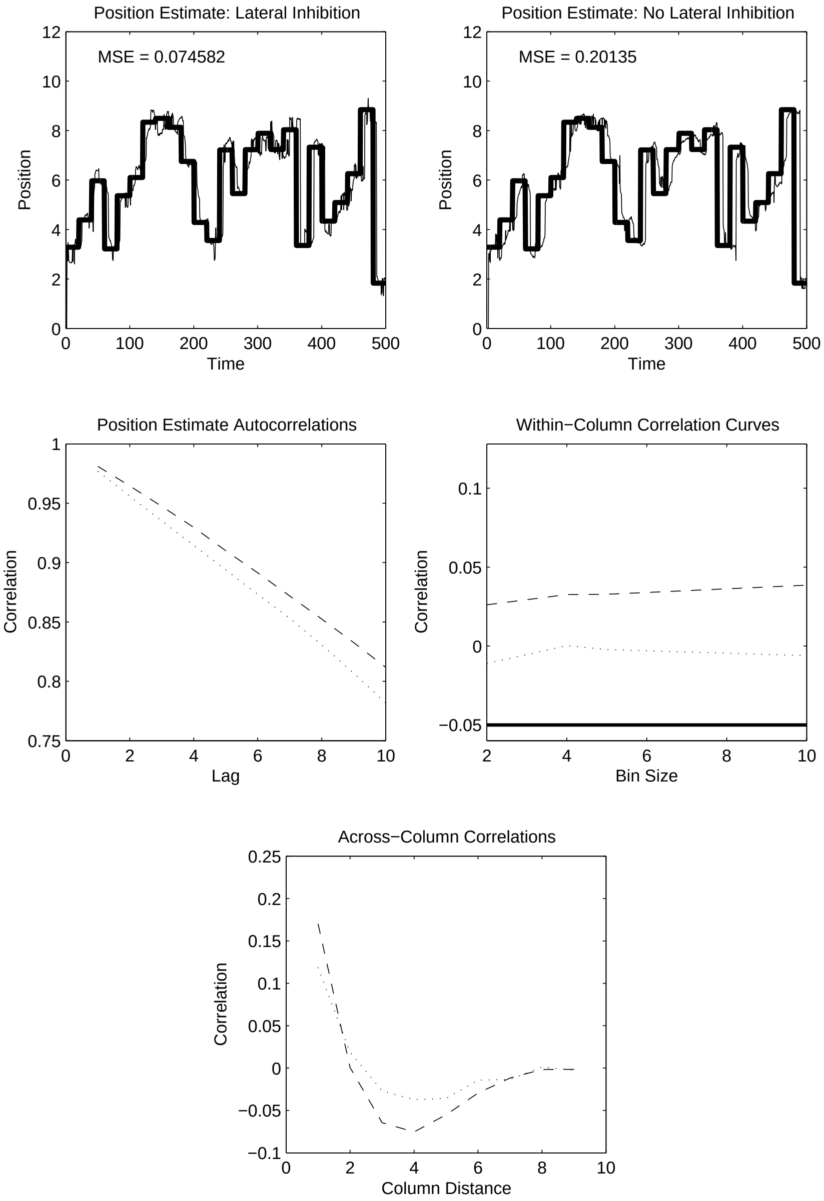


Figure 5.14: A detailed example using a 50ms estimate window. *Top*: Position targets (dotted line) and estimates (solid line) for a random steplike signal. The better performance with lateral inhibition present can be seen, although the effect is not as clearcut as for the model based on integrate-and-fire neurons. *Middle left*: Position estimate autocorrelations. These show little significant difference between the two networks, and suggest similar levels of high frequency noise in the position estimates for both. *Middle right*: Correlation curves of firing patterns for neurons in the same column. The use of inhibitory connections results in slightly negatively correlated firing patterns (dotted line). while the neurons without inhibitory connections show a slightly positive correlation. The solid line at the bottom represents the lowest possible correlation for a network of that size. *Bottom*: Mean correlation across columns as a function of column distance. This shows that neurons in columns close to each other will tend to respond in a similar way as expected. Neurons in columns at medium distances tend to respond in quite different ways, giving a 'Mexican hat' shape for both networks.

$$\left\{\begin{array}{rcl} g_K^{(i,j)}(t) & = & \bar{g}_K n^{(i,j)4}(t) \\ g_{Na}^{(i,j)}(t) & = & \bar{g}_{Na} m^{(i,j)3}(t) h^{(i,j)}(t) \\ \dfrac{dn^{(i,j)}(t)}{dt} & = & \alpha_n(v(t))(1 - n^{(i,j)}(t)) - \beta_n n^{(i,j)}(t) \\ \dfrac{dm^{(i,j)}(t)}{dt} & = & \alpha_m(v(t))(1 - m^{(i,j)}(t)) - \beta_m m^{(i,j)}(t) \\ \dfrac{dh^{(i,j)}(t)}{dt} & = & \alpha_k(v(t))(1 - h^{(i,j)}(t)) - \beta_k h^{(i,j)}(t) \end{array}\right. \tag{5.14}$$

Here $I_E^{(i,j)}(t)$ and $I_S^{(i,j)}(t)$ (including all referenced variables) are as previously defined in eqs.5.10 and 5.12. The specific membrane capacitance, c_M = 10nf/mm^2, and the membrane area, A_M = 0.1 mm^2, are also the same as before. The reversal potentials for the leakage channel, potassium channel and sodium channel respectively, here rescaled to zero resting potential in order to remain consistent with the previous models, are E_L = 11.387mV, E_K = -12mV and E_{Na} = 115mV, while their respective maximum conductance values are $\bar{g}_L$ = 0.003mS/mm^2, $\bar{g}_K$ = 0.36mS/mm^2 and $\bar{g}_{Na}$ = 1.2mS/mm^2. The opening rates of the specific channel activation and inactivation gating probabilities are $\alpha_n = \frac{(0.01(v(t)-10))}{(1-exp(-0.1(v(t)-10)))}$, $\alpha_m = \frac{(0.1(v(t)-25))}{(1-exp(-0.1(v(t)-25)))}$ and $\alpha_h = 0.07exp(-0.05(v(t)))$, while the closing rates are $\beta_n = 0.125exp(-0.0125(v(t))$, $\beta_m = 4exp(-0.0556(V(t)))$ and $\beta_h = \frac{1}{(1+exp(-0.1(v(t)-30)))}$. See section 2.1.3 for a more detailed description of these parameters.

We present the results of a typical run of our model using Hodgkin-Huxley neurons in figure 5.14. The MSE is once again lower for the network with lateral inhibitory connections, though the difference is not as pronounced as for the integrate-and-fire models, something also reflected in the autocorrelation curves. We also find that the correlation is negative for the model with inhibitory connections, although again this effect is not as strong here as for the integrate-and-fire models. There is a general trend (reflected in this typical simulation being shown) towards a slightly more positive correlation for the model both with and without lateral connections. The across-column correlation curves are very similar to those of the previous models. Altogether, these results can be seen to vindicate the earlier models, suggesting that the benefits of lateral inhibition exist across different neuron models, and different synapse models, and as such are robust and reliable.

5.5 Discussion

5.5.1 Basic Mechanism

Current theories of inhibitory mechanisms in the sensory cortex focus on lateral inhibition between neurons with slightly different response properties. These theories do not give an explanation of lateral inhibition between neurons with the same response properties, such as neurons in the same cortical columns in the primary visual cortex. We have presented a model here which demonstrates that one possible role of these local inhibitory mechanisms is that they improve the accuracy of the mean field potential by reducing the effects of noise. The explanation is as follows:-

1. Inhibition between neurons in the same column ensures that when one neuron fires, the probability of other neurons firing is decreased. This gives negatively correlated firing patterns.

2. Negatively correlated firing patterns mean that one neuron with a higher than average firing rate will result in reduced firing rates of other neurons in the pool. This is in effect the centre-surround space filling property of negatively correlated data.

3. The combined activity of all neurons in the pool (mean field potential) is stabilised by the centre-surround property, ensuring that the effects of random noise are minimised in comparison to the same pool of neurons operating without inhibitory connections.

4. The reduced-noise pooled activity is carried through decoding and leads directly to an improved performance on the given task of the pool of neurons.

In the case of our model, the improved performance of the network on the stimulus tracking task was evident in each experiment outlined. By effectively making the noise negatively correlated, we are reducing the level of noise beyond that which could be obtained by pooling uncorrelated or positively correlated noisy variables. An interpretation of this mechanism in terms of probabilistic noise reduction, which we believe is right at the heart of negatively correlated noise benefits, is offered in section 5.6.

5.5.2 Multi-Electrode Data

Given that our model has produced negatively correlated firing patterns, and shown the benefit of these, we might expect actual neural firing in the brain to be negatively correlated, at least in areas where the type of population coding we have used is in force. Historically, data taken from single-electrode recordings has not been able to give the firing patterns of neurons close to each other during the same time period, meaning that it has not been possible to accurately assess whether or not neural firing in the brain is in fact negatively correlated. Recently, however, data from the olfactory bulb of the rat and inferior temporal cortex of the sheep obtained using a multi-electrode array, coupled with an advance in spike sorting (Horton, Nicol, Kendrick, & Feng, 2007; Tate et al., 2005; Nicol et al., 2005), has allowed us to examine this question, and significantly it appears that neural firing patterns are indeed negatively correlated, as our model predicts.

As mentioned before, our theoretical results, together with our experimental findings, could contribute to the contemporary neurobiological on-going debate concerning the hypothesis of the temporal correlation advanced to solve the perceptual problem of linking different features in a unitary object or visual scene. Although fascinating and grounded on simulations and brain models, in addition to important electrophysiological findings on the sensory systems, this hypothesis is regarded as not conclusive, and it still excites numerous critical observations from different approaches. Negatively correlated firing patterns (of which anti-phase firing is a special case) are opposite to positively correlated firing patterns (of which synchronized or in-phase firing is a special case), but both of them can produce rhythmic firing patterns. As such, our findings here do not in any way depend upon the absence of rhythmicity in neural firing patterns, but can apply equally whether or not rhythms are present. Furthermore, negatively correlated neural firing patterns have been reported in some earlier published results, such as (Zohary et al., 1994). Nevertheless, in the literature to date, little attention has been given to its functional role.

5.5.3 Further Developments

Further exploration of the implications of the current model in terms of other theories could yield fruitful results. For example, it has been suggested that the brain employs sparse codes (Olshausen & Field, 1997; Field, 1995). Sparse coding actually arises naturally as a consequence of negatively correlated neural firing patterns, where the negative correlation ensures that only a small number of neurons are actively involved in representing a given input at any one time. Could the important benefits of local inhibitory mechanisms outlined here be responsible for the brain's choice of neural code? This is one of the questions for continued investigations into the benefits of local inhibitory mechanisms.

5.6 Probabilistic Noise Reduction

Finally, we attempt to give a simple mechanistic and intuitively satisfying explanation of how lateral inhibition can lead to an improved performance which is a generalisation of our previous more specific explanations, to place them in a proper context.

5.6.1 Threshold Noise Reduction

We start with two observations from the results in this chapter:-

- Combining replicated noisy signals reduces the overall noise level, due to the central limit theorem.
- In an artificial neural model (a threshold system), adding lateral inhibition between processing elements with the same response properties result in better performance.

It is important to understand the relationship between these two, and in particular, why adding lateral inhibition in threshold systems leads to improved performance. The effect of a lateral inhibitory connection is that when one element reaches a certain upper limit (the threshold), it sends a negative input into the other elements with the same response properties (neurons in the same column in our model), which results in those elements having lower overall input at that time point than would be the case if no lateral inhibition was present. This results in the firing pattern in the column being negatively correlated. However, it is not the negatively correlated firing pattern itself that is directly responsible for the improved performance. Rather, it is a second, more subtle negative correlation also introduced, that leads to the better performance. This is negatively correlated noise (hence the connection between our two initial observations). It arises because, all things being equal, and given zero-mean noise (a standard assumption) a processing element that reaches a high value (the threshold) is more likely to have a positive noise component (hence we also incorporate subthreshold stochastic resonance into our explanation), than a negative one, since all elements with positive noise will on average have higher values than all elements with negative noise. The inhibitory feedback sent from the element that has reached threshold (with likely positive noise) simulates negative noise added into the other elements in the same column, hence we have negative noise created to balance the positive noise, creating a negative correlation in the noise, and resulting in faster noise cancellation and better performance. In summary:-

- Element reaches threshold; this element is more likely to contain positive than negative noise
- Negative input is consequently sent to the replication elements (other elements with the same response properties; neurons in the same column), effectively simulating negative noise. Hence negative correlation is created between the positive noise of the firing element, and the negative noise of the replication elements.
- When the replications are combined (averaged), the negatively correlated noise gives better performance than an equivalent system without lateral inhibitory connections (and thus without the introduction of negatively correlated noise).

This mechanism is fine for systems that use upper thresholds (and it is trivial to create an equivalent system using lower thresholds; in this case, positive lateral connections would be required to simulate positive noise, and thus introduce negative correlation into the noise). However, for systems which are not naturally threshold systems, or for tackling problems which are normally handled with non-threshold systems, there are significant drawbacks for doing it this way, such as:-

- Threshold systems using columns of replicated elements with specific response properties, inevitably have a finite resolution, determined by the number of columns and the range of values required to be covered. For responses in between specific columns, an interpolated response mechanism of some sort must be used. Some simple systems, such as centre of mass, can give a reasonable performance on some tasks, but if the function to be approximated is very nonlinear, interpolation becomes increasingly problematic.

- Any interpolated response technique will be subject to edge-effects (where a response which is near one end of the set of columns, i.e. near the limit of values that the system can output, suffers from being unable to receive a contribution by columns beyond the end of the system, as such columns by definition don't exist). This leads to a central tendency in the output, which needs to be counterbalanced by some sort of filtering, which is problematic and requires tuning.

- A threshold system requires appropriate threshold values to be set, and appropriate interpretation of the output firing rates/counts of the system to be possible. Without a good choice of threshold value, the system may not fire at all, or will fire constantly with no difference between columns. The choice of threshold requires some knowledge of the statistics of the input variable. It is important to realise that without knowledge of the range of inputs, for example, it is not possible to reliably configure a threshold system to operate correctly, irrespective of whether or not it has lateral connections within columns.

- Similarly, a system using lateral connection between elements requires a weight for the strength of the connections to be set, which again required some knowledge of the statistics of the system in order for an appropriate value to be set.

While neural systems are threshold systems on the whole, for a more general understanding of this mechanism we therefore need to consider the equivalent for non-threshold systems.

5.6.2 Non-Threshold Noise Reduction

Given the difficulties of using threshold systems in situations where they may not be the natural choice, and given the desire to understand the mechanism in a way that is not dependent on the presence or absence of a threshold, it is useful to find an equivalent technique for introducing negatively correlated noise in non-threshold systems. We can use our understanding of the way in which inhibitory connections introduce negatively correlated noise in threshold systems to achieve this. In general, the situation for a signal with additive noise may be stated as

$$\mathbf{X} = \mathbf{S} + \mathbf{N} \tag{5.15}$$

where $\mathbf{X}$ is the variable that we have access to, $\mathbf{S}$ is the original signal and the $\mathbf{N}$ is the additive noise). Our objective is to reduce the noise as much as possible, thereby recovering the original signal. One way in which this may be done is to introduce negatively correlated noise, effectively cancelling out the original noise by adding opposite noise (an accelerated central limit effect as described in section 1.2.3). This is also equivalent to subtracting out our best estimate of the noise. The task is therefore to decompose the variable, $\mathbf{X}$, into a signal component $\mathbf{S}$ and a noise component $\mathbf{N}$. In this interpretation, the threshold approach described in the previous section undertakes the decomposition by estimating that the noise component is positive when the variable values have resulted in threshold being reached, and adds negatively correlated noise into the other elements in the same column as a result. The non-threshold equivalent of this would be to simply choose a value, above which the noise component of the variable would be regarded as likely to be positive, and add an element of negative noise (thus introducing negative correlation into the noise) to the replications of this element of the variable. There are several ways upon which this basic scheme can be improved:-

1. Both upper and lower thresholds can be employed, to allow both positive and negative noise to be subtracted out.

2. A more sophisticated probability model can be employed, depending upon how much knowledge of the noise and signal distributions is present. This is simply a more accurate generalisation of the threshold approach.

3. Addition of negatively correlated noise can be applied to all replications of a variable, including the one in which the noise estimation takes place. This also means that noise reduction can take place even where there is only once instance of a variable (no replications).

5.6.3 Noise Reduction Based on Probabilities

The general approach is to use our knowledge of the distributions from which the signal and noise are drawn, without knowing anything about the actual signal or noise values, to produce an estimate of the most likely noise value at each time/sample point, which allows us to counteract this by the introduction of negatively correlated noise.

An illustration of probabilistic noise reduction

In order to understand the principle, we can demonstrate the approach with a very simple example. We have an original signal which is a stochastic process described by a uniform distribution between 5 and 15, and additive noise, which is given by a Gaussian variable with zero mean and a variance (effective noise strength) of 5:-

$$\begin{cases} \pi_s(\mathbf{S}(t)) = 0.1 \quad \text{where} \quad 5 \leq S(t) \leq 15, \\ \pi_s(\mathbf{S}(t)) = 0 \quad \text{elsewhere.} \end{cases} \tag{5.16}$$

$$\pi_n(\mathbf{N}(t)) = N(0,5) \tag{5.17}$$

The variable to which we have access, $\mathbf{X}$(t), is the sum of the signal and noise (see equation 1). Our task is to estimate the separate signal and noise components of the variable, so that we can reduce the noise through the introduction of negatively correlated noise, using our knowledge of the relative distributions of the signal and the noise.

In order to illustrate this process, let us take a specific variable value: $\mathbf{X}(1) = 12$. There are obviously an infinite number of possible ways to decompose this into constituent signal and component parts. Even if we only considered integer values, there are still numerous combinations. For example, $\mathbf{S}(1) = 10$ and $\mathbf{N}(1) = 2$ is possible, as is $\mathbf{S}(1) = 13$ and $\mathbf{N}(1) =$ -1, $\mathbf{S}(1) = 5$ and $\mathbf{N}(1) = 7$ etc.. However, some of these combinations are more probable than others.

In order to determine this, we first need to consider the signal distribution. Although all values (within the limits) are equally probable in our current signal distribution, we can approach the question in a different way and ask if, given a particular candidate value, the signal value is more likely to be higher or lower than this value. For example, if our first guess is that signal component is 12, we may ask if this is really an unbiased estimate of the signal component. In fact, the cumulative distribution function shows us that all things being equal, there is a greater chance of the signal value being lower than this, than higher than this. An unbiased estimate of the signal value is always given by the mean, which in this case is 10. The further a candidate value is from this unbiased estimate, the less likely it is to have occurred. In the extreme case, we know for certain that signal values cannot have gone outside the limits, and our cumulative probability calculation confirms this.

On the face of it, this would suggest that the most probably signal value is always simply going to be given by the signal distribution mean, irrespective of what the variable value is. However, in addition to estimating the probability of a candidate signal value, we must also estimate the probability of the corresponding noise value (which together make up the variable value that we are given). In our example, we have a variable value of 12. Although we know 10 is the most likely signal value, we also know that 2, which would have to be the noise component if the signal had a value of 10 here, is not the most probable noise value (that is zero, given again by the mean of the distribution). Hence we need to balance the signal and noise probabilities for given candidate values, which in reality means calculating the joint probability. For additive noise, it is reasonable to assume that the signal and noise components are independent, hence the joint probability may be calculated by the product of the marginal distributions:-

$$P(\bar{\mathbf{S}}(t), \bar{\mathbf{N}}(t)) \quad = \quad P_s(\bar{\mathbf{S}}(t)) P_n(\bar{\mathbf{N}}(t)) \tag{5.18}$$

$$P_s(\bar{\mathbf{S}}(t)) = \int_{-\infty}^{\bar{\mathbf{S}}(t)} \pi_s(\bar{\mathbf{S}}(t))\, d\bar{\mathbf{S}}(t) \tag{5.19}$$

$$P_n(\bar{\mathbf{N}}(t)) = \int_{-\infty}^{\bar{\mathbf{N}}(t)} \pi_n(\bar{\mathbf{N}}(t))\, d\bar{\mathbf{N}}(t) \tag{5.20}$$

$$\bar{\mathbf{N}}(t) = \mathbf{X}(t) - \bar{\mathbf{S}}(t) \tag{5.21}$$

(where $\mathbf{X}$(t) is the variable value, $\bar{\mathbf{S}}$(t) is our candidate signal value and $\bar{\mathbf{N}}$(t) is our candidate noise value; note that a one-tailed cumulative distribution is required to calculate probabilities from the centre, so the integrals in equations 5 and 6 will change direction when the candidate value is below the distribution mean)

The most likely signal value $\bar{\mathbf{S}}$(t) (and hence corresponding noise value $\bar{\mathbf{N}}$(t)) at time t, given a variable value $\mathbf{X}$(t) at that time, is given by the maximum of equation 4, which can be solved numerically:-

$$\bar{\mathbf{S}}(t) = \arg\max_{\bar{\mathbf{S}}(t)} P(\bar{\mathbf{S}}(t), \bar{\mathbf{N}}(t)) \tag{5.22}$$

For our example value of 12 given earlier, the highest marginal probability signal value is 10 (the mean value), the highest marginal probability noise value is 0 (the mean value), but taken together, the highest joint probability occurs when the signal value is 10.2 and the noise value is 1.8. We now have the most likely signal and noise estimates. The final step is to add a new noise variable which is simply the negative of the noise estimate; this has the effect of subtracting out our best estimate of the existing noise, and leaving us with our best estimate of the original signal. It is notable that in test cases where we know the real noise values, this additional noise variable is shown to have the effect of making the noise negatively correlated overall, as predicted by our theoretical basis for the procedure.

In order to evaluate the performance of this method, a variable covering 1000 time/sample points was created using the signal and noise distributions described earlier. Two replications of the variable were generated (that is, the signal was created, one copy was made, and additive noise was generated and added separately to both copies), and these were averaged at the output, in order to exploit the standard central limit noise cancellation. This provides a benchmark for performance without using the technique outlined earlier. In addition, simple noise cancellation was tested, which involved upper and lower thresholds being used (above/below which an opposite noise element would be introduced to create negatively correlated noise); this is the non-threshold equivalent of the threshold lateral connection approach, but improved by having both upper and lower thresholds present rather than just an upper one. Finally, the full probabilistic noise cancellation method was also tested, finding the most probably signal and noise values at each time point, and adding negatively correlated noise to cancel this out. In each case, the estimated, noise-cancelled signal was compared to the actual original signal, to see how effective the noise cancellation was. The results are as follows:-

Condition	**MSE**	**Correlation**
No cancellation	13.3375	0.04466
Simple cancellation	7.7440	-0.31595
Probabilistic cancellation	5.9443	-0.35632

It is clear from this that the noise cancellation method is effective, and that using a more rigorous probability model results in improved performance. It is also notable that the noise correlation is most negative for the probabilistic cancellation. This provides further confirmation that the introduction of negatively correlated results in more effective noise cancellation. These results are fully replicable with a variety of different signals, including both periodic and non-periodic signals.

5.6.4 Conclusions

We know that negatively correlated noise cancels more rapidly than independent noise, providing an enhanced central limit noise cancellation technique. The challenge is to find a way of introducing negatively correlated noise into a system which inherently contains independent noise. We have previously developed a technique for doing this in threshold systems, through the use of lateral inhibitory connections; here we have generalised this technique to non-threshold systems, and at the same time refined it to further improve performance.

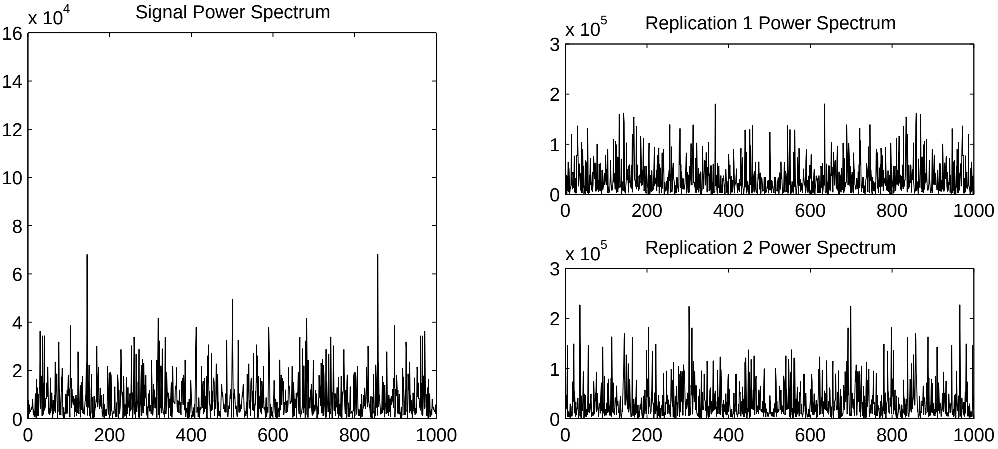


Figure 5.15: *Left*: Power spectrum of the original signal. *Right*: Power spectrum of two replications of the original signal with additive noise. It can be seen that the signal and noise cover the same frequency range.

One important point to note about this technique is that it operates entirely in the signal domain, and is not in any way concerned with the frequency characteristics of any signal that may exist. Figure 5.15 shows that for our previous example, the signal and the noise entirely overlap in the frequency domain. This means that existing noise-reduction techniques based on bandwidth filtering of any sort, such as Wiener filtering, will not be of any use in this situation. As such, our technique is complementary to most existing noise reduction techniques, and in particular may be applied in tandem with them. A signal with white noise, for example, may first have noise outside the signal frequency range removed by bandpass filtering, then have the remaining noise within the signal frequency range reduced using our technique, possibly followed by other filtering such as magnitude filtering.

In summary, the probabilistic noise reduction technique, of which negatively correlated firing patterns created by lateral inhibitory connections in neural systems is just one example, has the following properties and benefits:-

- It provides better performance than simple central limit theorem noise reduction.
- It can be used in threshold and non-threshold situations alike.
- It gives an optimal noise reduction for an additive noise model, based on the information at hand. This claim can be made because all of the information

that is known about the signal and noise is characterised by their probability distributions, and this information is used to give the most probable signal and noise estimate given a particular data value. Any other estimate is by definition more unlikely to be the actual signal value, and therefore sub-optimal.

- It can be applied for both periodic and non-periodic signals, including simple random variables as shown in our demonstration. All that is required is a knowledge of the signal range, which is available in many real applications, and an assumption (or knowledge) about the noise distribution, which is common practice.
- It can exist in tandem with other noise reduction techniques, to provide even better performance than they yield on their own.

Overall, we believe that lateral inhibitory connections in cortical columns give rise to negatively correlated firing patterns, but also more subtly to a crude form of probabilistic noise reduction that is designed to control the noise level, and in particular to reduce it from that which would be obtained from neurons that did not suppress the output of other neurons with similar response properties when firing. In other words, based on our simulation results and postulated general mechanism, we believe the brain is performing probabilistic noise reduction, with the density estimation setup in some sense controlled by patterns of local connectivity. Our example shows only a very simple and sub-optimal form of noise reduction, but it may be the case that the more complex patterns of connectivity found in the biological brain may setup to perform better density estimation. This is clearly an important area for future research.

Chapter 6

Conclusions

This work has been about how negative correlation can be beneficial in information processing, and how it can arise and be utilised in neural systems. We have demonstrated how it can operate at a quite abstract level, or how it can operate in a much more specifically neural way in noise correlation of a population of neurons, or even the correlation of neural spike trains. We have also shown how this may be achieved in practice with lateral inhibitory connections using biophysically detailed models. In this final chapter, we present an overall summary of the work described in the previous chapters and the contribution this makes to the field, as well as the limitations of the work as it stands which could be overcome in future developments.

6.1 Summary and Contribution

We initially presented (in section 1.2) an outline of the benefits of negative correlation in general. Although these may seem obvious, it is surprising how little these benefits are reported or utilised in information processing applications. The two key related benefits for negatively correlated variables over uncorrelated and especially positively correlated variables are space filling and accelerated central limit shrinkage. In chapter 2, we outlined the tools and methods used throughout the subsequent chapters, and gave a few demonstrations of some key properties.

In chapter 3 two different algorithms utilising these benefits were outlined and applied to natural images to demonstrate their benefits. Correlated component analysis adopts the ICA framework, but allows specification of a target mean correlation value or an entire correlation matrix which the activations (coefficients) of the derived components are required to match, while correlated basis analysis, adopting the same framework, allows specification of a target correlation value or matrix for the basis functions. Previous work has shown how this framework gives rise to basis functions that resemble V1 simple cell (Olshausen & Field, 1996; Hyvärinen & Hoyer, 2000) and complex cell (Hoyer, 2002) receptive fields, with the coefficients therefore representing activation of V1 neurons for both static (Durrant, 2003) and moving (Hurri & Hyvrinen, 2003) images. This framework adopts the principle of efficient coding based on redundancy reduction (Barlow, 1961, 2000; Barlow & Gardner-Medwin, 2001), and we have therefore demonstrated that the principle of negative correlation actually assists in representing images both more efficiently and more accurately when their coefficients are negatively correlated (CBA), as well as showing that negatively correlated basis functions themselves can be recovered more effectively using CCA than ICA.

The benefit of accelerated central limit noise cancellation in correlated variables was shown in chapter 4 adopting a different framework: that of suprathreshold stochastic resonance. This relatively recent and important concept (Stocks, 2000a, 2001a) shows that a certain level of noise greater than zero can allow for improved information transmission in a set of processing units with similar response properties. We have shown that this phenomenon exists when using integrate and fire neurons for the processing units, and further that the noise level can be controlled by adjusting the correlation between the inputs when they are other neurons approximated as Poisson processes. This optimal noise level is shown to be achieved with a negative correlation between the inputs.

Having demonstrated that negative correlation between the inputs to a neuron improves information transmission, we looked at the problem from the other side in chapter 5, showing how negative correlation between neural spike trains (outputs) is also beneficial. A visual tracking task was used to give a practical demonstration of how negatively correlated outputs in a small population of neurons gives rise to improved performance over uncorrelated outputs. Furthermore, we showed that this negative correlation can arise naturally from local inhibitory connections, a type of connectivity which is known to actually exist in the brain (Gardner & Martin, 2000), using not only integrate and fire neurons, but also Hodgkin-Huxley neurons and kinetic synapses, highlighting the fact that this is a biophysically realistic result. This is further reinforced by the results of some recent multi-electrode array studies (Nicol et al., 2005). Overall, we have seen that negative correlation between neurons is clearly useful in optimising information flow within the brain.

6.2 Limitations and Future Work

The limitations described here relate very much to the road not travelled. Although this work may be described as a tour through computational neuroscience guided by the principle of negative correlation, it is in fact a tour through only a tiny subset of that vast field. As such, the biggest limitation of this thesis, as for any thesis in this area, may be described simply as scope. A number of massive and hugely important areas have not been touched upon, including the whole subfield of learning and plasticity. The absence of these areas from this work should not be taken to indicate the absence of negative correlation from these areas, but rather simply the need to focus on a limited number of areas in order to give proper demonstration of negative correlation in operation.

On the other hand, the work is also limited from the opposite direction, in the sense that the need to demonstrate negative correlation in operation in several different (if complementary) ways in the brain has resulted in the need for brevity in each of these demonstrations, with the inevitable consequent lack of exhaustive investigation in any of these areas. There are many more ICA algorithms that could have been tested, or a much more exhaustive and rigorous examination of the new CCA and CBA algorithms, or the inclusion of a comparative study of V1 receptive fields for the new negative correlation algorithms, perhaps in the context of different images. Similarly, the role of noise correlation in SSR could have been examined for more biophysically realistic neurons, and different patterns of connectivity could have been examined for the lateral inhibitory network, or it could have been tested on different tasks.

In addition to these smaller branches that could have been taken, a whole range of even bigger issues could loosely be described as limitations, and certainly represent an outline sketch for possible future developments. These include the following:-

1. Both SSR and lateral inhibitory network approaches could be tested with neurons that do not have identical response properties. It may be hypothesised that rather than requiring this to be a binary state of either benefiting from negative correlation if the neurons have identical response properties, or not benefiting if they do not, there is a continuum here, and both the SSR effect, and the benefits in lateral inhibitory connected networks, gracefully degrade as the response properties between the units become increasingly dissimilar. This is an issue of obvious relevance for models of the early visual system (Rolls & Deco, 2001), but likely to be applicable in a wide variety of neural contexts.

2. The relationship between probability estimation and patterns of connectivity needs to be investigated. Although the principle of probability has been widely used to understand the brain (Rao, Olshausen, & Lewicki, 2002), and probabilistic networks have been used in the domain of artificial neural networks for some time (Haykin, 1999) so the relationship between probability and patterns of connectivity may be said to exist as an issue, these has not to our knowledge been a very explicit attempt to match up patterns of connectivity with probability density estimation. This is a potentially huge area, and in our view is the one the key concepts for understanding neural systems. The principle on which populations of laterally connected neurons may work, outlined in chapter 5, strongly suggests that simple lateral inhibitory connections are beneficial due to a crude form of density estimation, and it may be hypothesised that a more complex pattern of connectivity may give a better density estimation and hence better noise reduction.

3. Related to the previous issue, an organism in a dynamic environment requires

an adaptive information system in order to prosper (Pfeiffer & Scheier, 1999; Nolfi & Floreano, 2000). The question of how a pattern of connectivity can arise that is suitable for density estimation in a given environment, and how it can adapt to changes in that environment, is firmly in the domain of neural learning and plasticity. How far Hebbian learning (Hebb, 1949; Hertz, Krogh, & Palmer, 1991), and modern variants of it such as spike timing dependent plasticity (Song, Miller, & Abbott, 2000; Roberts & Bell, 2002; Rao & Lewicki, 2002), can go towards achieving this remains an open question.

4. The relationship between correlation in local populations of neurons and synchronisation at a more global level requires elucidation. Synchronised neural activity is believed to be important in neural coding (Eckhorn, Frien, Bauer, Woelbern, & Kehr, 1993; Singer, 1994), and as it is a subset of correlation (in the sense that synchronised neurons are necessarily correlated, although the reverse does not need to be true; see (Wu et al., 2007) for a brief discussion of this), it would suggest that correlation should have some effect on synchronisation. Negative correlation implies anti-synchronisation; how groups of locally negatively correlated neurons can give rise to bursts of globally correlated activity is another key question in computational neuroscience that is brought squarely into view by our work here.

6.3 Final Comments

Our results highlight the importance of negative correlation in neural systems. We have shown that the space filling and in particular accelerated central limit noise cancellation properties of negative correlation are beneficial in neural processing in a number of ways, and our detailed biophysical modelling results suggest that this is very likely to be widely used in the biological brain. We have given some indication as to how it works in principle and in practice, but clearly this work is, we hope, the start of a very much larger body of research, yet to be undertaken in the field, as to precisely how noise in the brain is controlled with negative correlation, how it interacts with the variability of neural response properties, how it interacts with patterns of connectivity, and how it develops and adapts to a changing environment.

We have started to chip away at the mystery of negative correlation in the brain, and an exciting new pathway into understanding the brain is opening up as a result.

References

Amari, S. (1999). Natural gradient learning for vver- and under-complete bases in ica. *Neural Computation*, *11*, 1875-1883.

Andersen, P., Dingledine, R., Gjerstad, L., Langmoen, I. A., & Laursen, A. M. (1980). Two different responses of hippocampal pyramidal cells to application of gamma-amino butyric acid. *Journal of Physiology*, *305*, 279-296.

Attneave, F. (1954). Informational aspects of vision. *Psychological Review*, *61*, 183-193.

Balakrishnan, N., Hariharakrishnan, K., & Schonfeld, D. (2005). A new image represesntation algorithm inspired by image submodality models, redundancy reduction, and learning in biological vision. *IEEE TPAMI*, *27*(9), 1367-1378.

Barlow, H. (1961). The coding of sensory messages. In W. Thorpe & O. Zangwill (Eds.), *Current problems in animal behaviour* (p. 331-360). CUP.

Barlow, H. (2000). Redundancy reduction revisited. *Network: Computation in Neural Systems*, *12*, 241-253.

Barlow, H., & Gardner-Medwin. (2001). Localist representation can improve efficiency for detection and counting. *Behavioral and Brain Sciences*, *23*, 467-468.

Bell, A., & Sejnowski, T. (1995). An information maximization algorithm that performs blind separation. *Advances in Neural Information Processing Systems*, *7*, 456-474.

Bressan, M., & Vitria, J. (2003). On the selection and classification of independent features. *IEEE TPAMI*, *25*(10), 1312-1317.

Chacron, M. J., Lindner, B., & Longtin, A. (2004). Noise shaping by interval correlations increases information transfer. *Phys Rev Lett*, *93(5)*, 080601.

Chacron, M. J., Longtin, A., & Maler, L. (2001). Negative interspike interval correlations increase the neuronal capacity for encoding time-varying stimuli. *J Neurosci*, *21*, 5328-5343.

Collins, J., Chow, C., Capela, A., & Imhoff, T. (1996). Aperiodic stochastic resonance. *Physical Review E*, *54*(5), 5575-5584.

Collins, J., Chow, C., & Imhoff, T. (1995). Stochastic resonance without tuning. *Nature*, *376*, 236-238.

Dayan, P., & Abbott, L. (2001). *Theoretical neuroscience: Computational and mathematical modelling of neural systems.* MIT Press.

Destexhe, A., Mainen, Z., & Sejnowski, T. (1994). An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neural Computation*, *6*, 14-18.

Destexhe, A., Mainen, Z., & Sejnowski, T. (1998). Kinetic models of synaptic transmission. In C. Koch & I. Segev (Eds.), *Methods in neuronal modeling* (2nd ed., p. 1-25). MIT Press.

Dowling, J. E. (1992). *Neurons and networks.* Belknap.

Durrant, S., & Feng, J. (2006). Negatively-correlated firing: The functional meaning of lateral inhibition within cortical columns. *Biological Cybernetics*, *95*, 431-453.

Durrant, S. J. (2003). *Natural image statistics and the early visual system: Independent component analysis and sparse coding approaches.* Unpublished master's thesis, University of Sussex.

Eckhorn, R., Frien, A., Bauer, R., Woelbern, T., & Kehr, H. (1993). High frequency (60-90 hz) oscillations in primary visual cortex of awake monkey. *Neuroreport*, *4*, 243-246.

Feng, J. (Ed.). (2004). *Computational neuroscience: A comprehensive approach.* Chapman and Hall/CRC Press.

Feng, J., & Ding, M. (2004, June). Decoding spikes in a spiking neuronal network. *Journal of Physics A: Mathematical and General*, *37*(22), 5713-5728.

Feng, J., & Tirozzi, B. (2000). Stochastic resonance tuned by correlations in neuronal models. *Phys. Rev. E.*, *61*, 4207-4211.

Field, D. J. (1995). Visual coding, redundancy and feature detection. In M. J. Arbib (Ed.), *The handbook of brain theory and neural networks* (p. 1012-1016). MIT Press.

Gammaitoni, L., Hanggi, P., Jung, P., & Marchesoni, F. (1998). Stochastic resonance. *Rev.Mod.Phys*, *70*, 223-287.

Gardner, E., & Martin, J. (2000). Coding of sensory information. In E. Kandel, J. Schwartz, & T. Jessell (Eds.), *Principles of neural science* (p. 411-429). McGraw-Hill.

Gautrais, J., & Thorpe, S. (1998). Rate coding vs temporal order coding: A theoretical approach. *Biosystems*, *48*, 57-65.

Gerstner, W., & Kistler, W. (2002). *Spiking neuron models.* Cambridge University Press.

Gray, C. M., Knig, P., Engel, A. K., & Singer, W. (1989, March). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, *338*, 334-337.

Hateren, J. van, & Schaaf, A. van der. (1998). Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc.R.Soc.Lond.*, *B*(265), 359-366.

Haykin, S. (1999). *Neural networks: A comprehensive foundation.* Prentice Hall.

Hebb, D. (1949). *The organization of behaviour: A neuropsychological theory.* Wiley.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation.* Addison-Wesley.

Hirsch, J. A., & Gilbert, C. D. (1991). Synaptic physiology of horizontal connections in the cat's visual cortex. *Journal of Neuroscience*, *11*, 1800-1809.

Hoch, T., Wenning, G., & K., O. (2003a). Adaptation using local information for maximizing the global cost. *Neurocomputing*, *52-54*, 541-546.

Hoch, T., Wenning, G., & K., O. (2003b). Optimal noise-aided signal transmission through populations of neurons. *Phys.Rev.E*, *68*, 011911.

Hoch, T., Wenning, G., & K., O. (2005). The effect of correlations in the background activity on the information transmission properties of neural populations. *Neurocomputing*, *65-66*, 365-370.

Hodgkin, A., & Huxley, A. (1952). A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. *J.Physiol.(Lond.)*, *117*, 500-544.

Horton, P., Bonny, L., Nicol, A., Kendrick, K., & Feng, J. (2005). Applictions of multi-variate analysis of variances (manova) to multi-electrode array data. *J.Neurosci.Meth.*, *146*, 22-41.

Horton, P., Nicol, A., Kendrick, K., & Feng, J. (2007). Spike sorting based upon machine learning algorithms (soma). *J. of Neuoroscience Methods*, *160*(1), 52-68.

Hoyer, P. (2002). Non-negative sparse coding feature subspaces. *Neural Networks for Signal Processing*, *12*, 557-565.

Hoyer, P. (2003). Modeling receptive fields with non-negative sparse coding. *Neurocomputing*, *52-54*, 547-552.

Hubel, D., & Wiesel, T. (1959). Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, *148*, 574-591.

Hubel, D., & Wiesel, T. (1979). Brain mechanisms of vision. *Scientific American*, *241*(3), 150-162.

Hurri, J., & Hyvrinen, A. (2003). Simple-cell-like receptive fields maximize temporal coherence in natural video. *Neural Computation*, *15*(3), 663-691.

Hyvärinen, A., & Hoyer, P. (2000). Emergence of phase- and shift- invariantfeatures by decomposition of natural images into independent feature subspaces. *Neural Computation*, *12*(7), 1705-1720.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis.* Wiley.

Hyvärinen, A., & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, *9*(7), 1483-1492.

Jung, P. (1995). Stochastic resonance and optimal design of threshold detectors. *Physics Letters A*, *207*, 93-104.

K., W., & F., M. (1995). Stochastic resonance and the benefits of noise: from ice ages to crayfish and squids. *Nature*, *373*, 3336.

Kandel, E. (2000). Nerve cells and behaviour. In E. Kandel, J. Schwartz, & T. Jessell (Eds.), *Principles of neural science* (p. 19-35). McGraw-Hill.

Kistler, W., & De Zeeuw, C. (2002). Dynamical working memory and timed responses: The role of reverberating loops in the olivo-cerebellar system. *Neural Computation*, *14*, 2597-2626.

Koester, J., & Siegelbaum, S. (2000). Membrane potential. In E. Kandel, J. Schwartz, & T. Jessell (Eds.), *Principles of neural science* (p. 125-139). McGraw-Hill.

Kosko, B., & Mitaim, S. (2003). Stochastic resonance in noisy threshold neurons. *Neural Networks*, *16*(5-6), 755-761.

Lee, D., & Seung, H. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing*, *13*, 556-562.

Lee, T.-W., Lewicki, M., & Sejnowski, T. (2000). Ica mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation. *IEEE TPAMI*, *22*(10), 1078-1089.

Levin, J., & Miller, J. (1996). Broadband neural encoding in the cricket cercal sensory system enhanced by stochastic resonance. *Nature*, *380*, 165-168.

Lindner, B., Longtin, A., & Chacron, M. J. (2005). Integrate-and-fire neurons with threshold noise: a tractable model of how interspike interval correlations affect neuronal signal transmission. *Phys Rev E*, *72(2)*, 021911.

Longtin, A. (1993). Stochastic resonance in neuron models. *J.Stat.Phys.*, *70*, 309-327.

Mainen, Z., & Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. *Science*, *268*, 1503-1506.

Manookin, M. ., & Demb, J. . (2006). Presynaptic mechanism for slow contrast adaptation in mammalian retinal ganglion cells. *Neuron*, *50,3*, 453 - 464.

Manwani, A., & Koch, C. (1999). Detecting and estimating signals in noisy cable structures, i: Neuronal noise sources. *Neural Computation*, *11*, 1797-1829.

Martin, K. A. C. (1984). Neuronal circuits in cat striate cortex. In E. Jones & A. Peters (Eds.), *Cerebral cortex: Functional properties of cortical cells* (Vol. 2, p. 241-284). Plenum.

McClelland, J., & Rumelhard, D. (Eds.). (1986). *Parallel distributed processing: Psychological and biological models* (Vol. 2). MIT Press.

Middleton, J. W., Chacron, M. J., Lindner, B., & Longtin, A. (2008). Correlated noise and memory effects in neural firing statistics. *www.scientificcommons.org*, 42418344.

Moghaddam, B. (2002). Principal manifolds and probabilistic subspaces for visual recognition. *IEEE TPAMI*, *24*(6), 780-788.

Moss, F., Pierson, D., & O'Gorman, D. (1994). Stochastic resonance: Tutorial and update. *International Journal of Bifurcation and Chaos*, *4*, 1383-1397.

Nicol, A., Feng, J., & Kendrick, K. ((in revision)). Negative correlation yields computational vigour in a mammalian sensory system.

Nicol, A. U., Magnusson, M. S., Segonds-Pichon, A., Tate, A., Feng, J., & Kendrick,

K. M. (2005). *Local and global encoding of odor stimuli by olfactory bulb neural networks.*

Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics.* MIT Press.

Olshausen, B., & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607-609.

Olshausen, B., & Field, D. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, *37*, 3311-3325.

Palanca, B. J. A., & DeAngelis, G. C. (2005). Does neuronal synchrony underlie visual feature grouping? *Neuron*, *46*, 333-346.

Pfeiffer, R., & Scheier, C. (1999). *Understanding intelligence.* MIT Press.

Pierce, J. R. (1980). *An introduction to information theory: Symbols, signals and noise.* Dover.

Plesser, H., & Gerstner, W. (2000). Noise in integrate-and-fire neurons: from stochastic input to escape rates. *Neural Computation*, *12*, 367-384.

Rall, W. (1977). Core conductor theory and cable properties of neurons. In E. Kandel (Ed.), *Handbook of physiology* (Vol. 1, p. 39-97). American Physiology Society.

Rao, R., & Lewicki, M. (2002). Predictive coding, cortical feedback and spike timing dependent plasticity. In R. Rao, B. Olshausen, & M. Lewicki (Eds.), *Probabilistic models of the brain.* MIT Press.

Rao, R. P. N., Olshausen, B. A., & Lewicki, M. S. (2002). *Probabilistic models of the brain: Perception and neural function.* MIT Press.

Rasch, B., Büchel, C., Gais, S., & Born, J. (2007). Odor cues during slow-wave sleep prompt declarative memory consolidation. *Science*, *315*, 1426-1429.

Ratliff, F. (1972). Contour and contrast. *Scientific American*, *226*(6), 901-910.

Rieke, F., Warland, D., Steveninck, R. de Ruyter van, & Bialek, W. (1997). *Spikes: Exploring the neural code.* MIT Press.

Roberts, P., & Bell, C. (2002). Spike-timing dependent synaptic plasticity in biological systems. *Biological Cybernetics*, *87*, 392-403.

Rolls, E., & Deco, G. (2001). *The computational neuroscience of vision.* OUP.

Rossoni, E., & Feng, J. (2007). Decoding spike ensembles: Tracking a moving stimulus. *Biological Cybernetics*, *96*(1), 99-112.

Rumelhard, D., & McClelland, J. (Eds.). (1986). *Parallel distributed processing: Foundations* (Vol. 1). MIT Press.

Shadlen, M., & Newsome, W. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *J. Neurosci*, *18*, 3870-3896.

Shannon, C., & Weaver, W. (1949). *The mathematical theory of communication.* University of Illinois Press.

Sheather, S. J., & Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B, 53*, 683690.

Siegelbaum, S., & Koester, J. (2000). Ion channels. In E. Kandel, J. Schwartz, & T. Jessell (Eds.), *Principles of neural science* (p. 105-124). McGraw-Hill.

Simoncelli, E., & Olshausen, B. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscicence 2001*, *24*, 1193-1216.

Singer, W. (1994). The role of synchrony in neocortical processing and synaptic plasticity. In E. Domany, J. van Hemmen, & K. Schulten (Eds.), *Models of neural networks ii.* Springer-Verlag.

Song, S., Miller, K., & Abbott, L. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, *3*(9), 91926.

Stein, R. (1965). A theoretical analysis of neuronal variability. *Biophys.J.*, *5*, 173-194.

Stein, R. (1967). Some models of neuronal variability. *Biophys.J.*, *7*, 37-68.

Stocks, N. (2000a). Optimising information transmission in model neuronal ensembles *Stochastic Processes in Physics, Chemistry and Biology.* In J. Freund & T. Poschel (Eds.), *Lecture notes in physics* (p. 150-159). Springer-Verlag.

Stocks, N. (2000b). Suprathreshold stochastic resonance in multilevel threshold systems. *Phys.Rev.Lett.*, *84*, 2310-2314.

Stocks, N. (2001a). Information transmission in parallel arrays of threshold elements: suprathreshold stochastic resonance. *Phys.Rev.E*, *63*, 1-9.

Stocks, N. (2001b). Suprathreshold stochastic resonance: an exact result for uniformly distributed signal and noise. *Phys.Lett.A*, *279*, 308-312.

Stocks, N., & Mannella, R. (2001). Generic noise-enhanced coding in neuronal arrays. *Phys.Rev.E*, *64*, 1-4.

Tate, A. J., Nicol, A. U., Fischer, H., Segonds-Pichon, A., Feng, J., Magnusson, M. S., & Kendrick, K. M. (2005). *Lateralised local and global encoding of face* *stimuli by neural networks in the temporal cortex.*

Theunissen, F., & Miller, J. P. (1995). Temporal encoding in nervous systems: a rigorous definition. *J. Comput. Neurosci*, *2*, 149-162.

Tucker, T. R., & Katz, L. C. (2003a). Recruitment of local inhibitory networks by horizontal connections in layer 2/3 of ferret visual cortex. *Journal of Neurophysiology*, *89*, 501-512.

Tucker, T. R., & Katz, L. C. (2003b). Spatiotemporal patterns of excitation and inhibition evoked by the horizontal network in layer 2/3 of ferret visual cortex. *Journal of Neurophysiology*, *89*, 488-500.

Walker, M., Stickgold, R., Alsop, D., Gaab, B., & Schlaug, G. (2005). Sleep-dependent motor memory plasticity in the human brain. *Neuroscience*, *133*, 911-917.

White, J., Rubinstein, J., & Kay, A. (2000). Channel noise in neurons. *Trends In Neuroscience*, *23*, 131-137.

Wiesenfeld, K., & Jaramillo, F. (1998). Minireview of stochastic resonance. *Chaos*, *8*, 539-548.

Wu, J., Kendrick, K., & Feng, J. (2007). Detecting correlation changes in electrophysiological data. *J. Neuroscience Methods*, *161*(1), 155-165.

Yang, J., Zhang, D., Frangi, A., & Yang, J.-Y. (2004). Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE TPAMI*, *26*(1), 131-137.

Zador, A. (1998). Impact of synaptic unreliability on the information transmitted by spiking neuron. *J. Neurophysiol*, *79*, 1219-1229.

Zohary, E., Shadlen, M. N., & Newsome, W. T. (1994, July). Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature*, *370*, 140-143.

Appendix A

Kernel Density Estimation

Here we introduce methods for kernel density estimation in the case of probability density functions of one or two variables only. The approach taken is an informal introduction and outline of the basic method, with an emphasis on intuitive understanding, rather than a rigorous proof, which could be found in statistical methods textbooks.

A.1 Univariate Kernel Density Estimation

A.1.1 Sampling Distribution and the Histogram Approach

We start by defining a continuous random variable, x, for which we have a set of n samples X_i, for $i = 1$ to n. We want to estimate the probability density function (pdf) of x, without using any functional form, i.e. we want to do nonparametric density estimation.

Our first option is to simply use the sampling distribution, assigning a probability $p(x_i)$ of $1/n$ to each sample value, and a probability of zero for all other possible values. For an adequately-sampled discrete random variable, this approach is fine. For a continuous random variable such as we have here, however, we have a problem in that the values in the sample represent just one finite set out of an infinite number of possibilities. As such, we cannot expect to get the same values back if we were to take another sample, and so the probability density we have using this approach is not useful to us; there are too many gaps no matter how large our sample. Slightly more rigorously, it can be said that probability of obtaining any given value for X_i out of an infinite set of possibilities (the set of continuous numbers between fixed bounds) is infinitely small, and hence any approach based on this is futile.

The sampling distribution does, however, tell us something about how regularly, when moving along the axis of values for variable x, we can expect to find sampled events, i.e. the density of events. This leads us to the histogram approach[1], where we look at the density of events within a given window, and declare that to represent the relative probability across that entire window. In other words, we put the samples X_i into a set of discrete bins which each have a width b, and count the number in each bin. The probability of finding $x = X_0$ is determined by the proportion of the total responses (n, the number of samples) that are found in the bin to which X_0 belongs.

There are three problems with this approach:-

1. The density $p(x)$ is not continuous (it jumps when moving from one bin to the next).
2. The density $p(x)$ is dependent on the bin edge locations.
3. The density $p(x)$ is dependent on the width of the bins, b.

A.1.2 Kernel Density Estimation

We can overcome problems one and two by using kernel density estimation. Here, instead of placing points within pre-determined finite-length bins, we evaluate a kernel (itself a pdf of some form, which acts as a distance measurement) centred at each sample position, and average the kernels (sum over them, and divide by the number) to get the pdf. A kernel is characterised by a bandwidth, b, so problem three of the histogram approach unfortunately remains (and will be discussed briefly at the end).

Given the form of the kernel, we first define a grid of values (covering at least the entire range of sampled values, plus a bit extra on either side), with a small but

[1] Alternatively, it may lead us to a convolution approach; see section 2.2.1 for a demonstration of this equivalence

finite resolution. For a kernel of unit variance at a specific sample point, we first divide the distance between all the sample points and the kernel centre (which form the essential input to the kernel) by the bandwidth b. This effectively makes the distances smaller by a factor of b, which makes the kernel values for these distances correspondingly larger by the same factor (subject to the particular shape of the kernel). We then sum over all the kernels to get an overall kernel, divide by the number of kernels to get the average of the kernels, and also divide by the bandwidth to cancel out the effect earlier of increasing the kernel values by reducing the distances by the bandwidth b. This means that the bandwidth does not affect the overall size of the average kernel (which is therefore just the same as any of the individual kernels from which it has been averaged), but it does affect the reach of each kernel in effect. In other words, it simulates an increased variance for the kernel.

For a Gaussian kernel (the most common choice), we can see this explicitly, since if $f(X/b) = N(0,1)/b$, then $f(X) = N(0,\sqrt{b})$. Hence for kernel density estimation with a Gaussian kernel $k(x)$, we have:-

$$\left\{ \begin{array}{rcl} \hat{f}_0(x) & = & \frac{1}{nb}\sum_{i=1}^{n} k_0(\frac{x-X_i}{b}) \\ k_0(x) & = & N(0,1) \end{array} \right. \tag{A.1}$$

which is equivalent to

$$\left\{ \begin{array}{rcl} \hat{f}(x) & = & \frac{1}{n}\sum_{i=1}^{n} k(x-X_i) \\ k(x) & = & N(0,\sqrt{b}) \end{array} \right. \tag{A.2}$$

Here, $\hat{f}(x)$ gives us the estimated probability density for variable x. There are various somewhat involved methods for the choice of the optimal value for the bandwidth parameter b, but a simple rule of thumb proposed by Sheather and Jones (Sheather & Jones, 1991) that works reasonably well for Gaussian densities is b=0.90 min(σ_X,IQR/1.34)$n^{-1/5}$, where IQR is the interquartile range of the sample data and σ_X is the standard deviation of the sample data .

The general approach of kernel density estimation as shown here is based on treating each data sample as having its own probability distribution, and finding the expected value of these distributions (this effectively means averaging them, aka summing and dividing by the number of them, since the probability for the occurrence of each distribution is the same, as they each represent a single and therefore equiprobable sample). In the limit where the kernel is a Dirac function, the probability of getting the actual sample value is 1, and the probability for all others are zero. However, other kernel functions such as the Gaussian function, make the assumption instead that the sample simply represents the most probable value of a range of possibilities, and accordingly gives a probability to all the possible values (defined by our grid in practice), with the probability typically declining the further away from the actual sample value we are (hence a Gaussian function works well as a typical kernel function; this also explains why the kernel functions are required to be densities themselves, i.e. integrate to one).

A.2 Multivariate Kernel Density Estimation

A.2.1 Different Types of Probability Distributions

When more than one variable is involved, we have multivariate sample data, and need to estimate a multivariate pdf. In this section, we will look at the case where there are two variables, x and y, and we have a set of paired data points (X_i, Y_i). We first define some basic distribution types:-

- $p(x)$ The marginal density of x (the probability of $x = X$, without knowing anything else); one-dimensional.

- $p(y)$ The marginal density of y (the probability of $y = Y$, without knowing anything else); one-dimensional.

- $p(x, y)$ The joint density of x and y (the probability of a particular pair of values $(x = X, y = Y)$ together); two-dimensional (grid across all possible x and y combinations)

- $p(x|y)$ The conditional density of x (the probability of $x = X$ given $y = Y_0$); one-dimensional (slice through $p(x, y)$ at $y = Y_0$).

- $p(y|x)$ The conditional density of y (the probability of $y = Y$ given $x = X_0$); one-dimensional (slice through $p(x, y)$ at $x = X_0$).

We can also define some basic relationships amongst the different distribution types, following Bayes theorem:-

$$\begin{aligned} p(x,y) &= p(x|y)p(y) = p(y|x)p(x) \\ p(x|y) &= \frac{p(x,y)}{p(y)} = \frac{p(y|x)p(x)}{p(y)} \\ p(x) &= \sum_y p(x|y)p(y) = \sum_y p(x,y) \end{aligned} \tag{A.3}$$

A.2.2 Bivariate Kernel Density Estimation

Looking initially at the sampling distribution again, it should be apparent that the same problem, and the same potential solution, applies for multivariate (bivariate in this example) as for univariate. The kernel principle requires a little more explanation, however, in view of the new distribution types outlined above. We first choose a sample data point (X_i, Y_i), which amounts to selecting a point based on either a particular X_i value or a particular Y_i value. These two are equivalent ($p(x|y)p(y) = p(y|x)p(x)$), so for the purposes of illustration we will consider the case where Y_i value is chosen. We have a kernel density $k(Y_i)$ for this Y_i value, following the same principle as in the univariate case, which in the limit is a Dirac function (probability of one for the sample value and zero for all others). For the corresponding X_i value in the pair, we have a conditional density, conditioned on the Y_i value; this is given by the kernel density $k(X_i)$. The joint probability of (X_i, Y_i) is the product of the two kernels, since $p(x,y) = p(x|y)p(y) = p(y|x)p(x)$, in the general case using different bandwidth parameters a and b for the two kernels. Just as for the univariate case, the overall estimate of the density function was simply the average of all such kernels (normalised for bandwidth), so in estimating the multivariate function we have:-

$$\left\{ \begin{aligned} \hat{f}(x,y) &= \frac{1}{nab}\sum_{i=1}^{n} k(\frac{x-X_i}{a})k(\frac{y-Y_i}{b}) \\ k(x) &= N(0,1) \end{aligned} \right. \tag{A.4}$$

This gives us a two-dimensional joint density $p(x,y)$. If we want the conditional density $p(x|y)$, we can see from the previous equations that we should divide this by the marginal density $p(y)$, which is of course calculated using simple univariate kernel density estimation, hence altogether we get:-

$$\hat{f}(x|y) \quad = \quad \frac{\frac{1}{nab}\sum_{i=1}^{n} k(\frac{x-X_i}{a})k(\frac{y-Y_i}{b})}{\frac{1}{nb}\sum_{i=1}^{n} k(\frac{y-Y_i}{b})} \tag{A.5}$$

We can use this by setting a conditioning value, $y = Y_0$, and finding the conditional density $p(x|y = Y_0)$:-

$$\hat{f}(x|y = Y_0) \quad = \quad \frac{\frac{1}{nab}\sum_{i=1}^{n} k(\frac{x-X_i}{a})k(\frac{Y_0-Y_i}{b})}{\frac{1}{nb}\sum_{i=1}^{n} k(\frac{Y_0-Y_i}{b})} \tag{A.6}$$

which can be rewritten as:-

$$\begin{cases} \hat{f}(x|y = Y_0) \quad = \quad \frac{1}{nab}\sum_{i=1}^{n} w_i(y)k(\frac{x-X_i}{a}) \\ w_i(y) \quad = \quad \frac{k(\frac{Y_0-Y_i}{b})}{\frac{1}{nb}\sum_{i=1}^{n} k(\frac{Y_0-Y_i}{b})} \end{cases} \tag{A.7}$$

Here we have deliberately not algebraically cancelled the coefficients $\frac{1}{nab}$ and $\frac{1}{nb}$, as this allows us to have an intuitive understanding of these equations. The main conditional density equation shows that we sum over the contribution to the conditional density of x from each sample point X_i, just as we would for the marginal density $p(x)$. However, these contributions are now weighted by the distance the corresponding Y_i value is from the conditioning value Y_0, measured by the same kernel function. In the limit of a Dirac function, we would have the situation where only X values whose corresponding Y value exactly matched the conditioning value would have any influence over the density; this is the standard approach for discrete random variables. For the continuous variables we have here, however, we allow all X values to contribute, but weighted by their corresponding Y value distance from the conditioning value. The weight $w_i(y)$ has two components. The main one (the numerator) is the kernel distance measurement required for the weight to function as just described. The other one (the denominator) represents the marginal probability $P(Y_0)$, which needs to be factored out in the conditional density (because the contributions of the the Y distance measurement should not be reduced by the probability $P(Y_0)$, as this is given as a conditioning variable, i.e. it has actually occurred; this is effectively just a normalising, scaling constant).

There are no straightforward methods for selecting values of the bandwidth parameters a and b; the situation is more complex than for the univariate case because in addition to variance, we now need to consider covariance as well, i.e. there is direction as well as distance of the variance component of the joint density. Nevertheless, the univariate parameter selected independently for each variable may provide a reasonable rule of thumb for low dimensionalities (such as the bivariate case considered here) and especially where the variables are believed to be relatively independent.