An Evaluation of the Fidelity of the Integrate-and-Fire Model for Neural Data

Zeyn Ali Saigol

Master of Science
School of Informatics
University of Edinburgh
2006

Abstract

Finding a simple, one-dimensional neural model that can replicate the input/output behaviour of a complex, conductance based model is a key goal in current neuroscience research. In this work, we aim to match the spike train produced by a Hodgkin-Huxley (HH) model in response to a white noise stimulus using a variant of the integrate-and-fire model, the Spike Response Model (SRM). The parameters of the SRM are fitted to membrane potential data produced by the HH model by finding the optimal linear filter of the stimulus, given the subthreshold potential. We evaluate two different forms of the SRM, where the filter is either independent or dependent on previous spike times. We find both versions of the SRM reproduce the HH data well when fitted using this method, with up to 71% of the spikes produced by the HH model being predicted to within 2ms by the SRM. The spike-time dependent filter is found to perform better, but there are circumstances in which the spike-time independent filter would be a better choice, in particular if data is limited.

Acknowledgements

I would like to thank my supervisors, Chris Williams and David Sterratt, for all their help and enthusiasm.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Zeyn Ali Saigol*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

The response of a neuron to an input stimulus can be predicted very well by conductance-based models such as the classic model developed by Hodgkin and Huxley in the 1950s (Hodgkin and Huxley, 1952). However, such models are usually complex – for example the Hodgkin-Huxley (HH) model, which is one of the more straightforward conductance-based models, consists of 4 coupled differential equations in 4 variables. This makes such models computationally demanding, and hard to grasp intuitively. It is also possible that they capture too accurately the detail of the low-level biophysical processes of neural computation – one suspects that there should be a higher-level abstraction of the computational mechanisms of the brain.

An important goal in computational neuroscience is therefore the development of simple neural models that replicate the behaviour of more complex, conductance-based models. Simple models will be faster, enabling larger networks of neurons to be simulated, and more amenable to analytic examination of their behaviour. In a wider neuroscience context, being able to simulate a large network of biologically-accurate neurons is a key step toward understanding how brains work; by replicating emergent cognitive behaviours in an artificial network, we can understand what produces these behaviours in the cortex.

In this project, a simple model was made to approximate the behaviour of a conductance-based model by explicitly fitting its parameters to data produced by the conductance model. Jolivet et al. (2006) have produced good results using this approach, and we aimed to follow and further explore their techniques.

1.2 Aims

The key goal of this project was to evaluate the performance of a numerical fitting procedure for matching the output of a simple integrate-and-fire model to that of a more complex neuron model, when driven by a white noise input. To achieve this, firstly the fitting procedure has to be optimised, which was done on the basis of how similar the output predicted by the simple model was to the actual output produced by the conductance model. As well as graphical comparisons, quantitative measures were

used to measure the degree of similarity, based on both the membrane voltage and on the sequence of spike times.

Once the procedure is optimised, several extensions are possible. The robustness of the fitted parameters against different types and amplitudes of stimulus current can be explored, or the results with different conductance-based models can be investigated.

1.3 Summary of achievements

Much of the procedure followed was based on work done by Jolivet et al. (2006), using a variant of the integrate-and-fire model known as the Spike Response Model (SRM). In this model, the neuron membrane potential $u(t)$ is predicted as a function of the input current $I(t)$:

$$u(t) = \eta(t - \hat{t}) + \int_0^\infty \kappa(s) I(t - s)\, ds \tag{1.1}$$

where $\hat{t}$ is the time of the last spike, and the model is parametrised by the two kernels η and κ. η models the shape of a spike, which follows a very stereotypical pattern dependent only on the time since the last spike $(t - \hat{t})$. κ is a causal filter, modelling the response of the neuron as a function of the input current history.

The kernels η and κ were found by fitting the SRM to data produced by a complex model, η by an averaging based method, and κ by a least-squares fit, customised to work with this type of filter. Two different forms of κ were used: a Spike-time Independent Kernel (SIK) where κ is a function of s only, and a Spike-time Dependent Kernel (SDK), where κ depends on the time since the last spike as well as s. The SDK allows the model to include adaptation, whereby the response of the neuron to a given stimulus is reduced in the period immediately following an action potential.

The complex model chosen for this project was a HH model, as implemented by the NEURON software package. Time constraints prevented experimentation with other forms of conductance model or input current; instead, the relative performance of the SIK and SDK at predicting the neuron membrane potential was investigated.

Key results were:

- The spike template kernel η was found to be a critical part of the SRM. Despite fine-tuning the method for extracting η, it was not possible to completely isolate the sub-threshold voltage from spike effects.
- The SDK was found to perform better than the SIK at predicting spike times, with up to 71% of the spikes generated by the HH model being correctly predicted by the SRM.
- The SDK was significantly more complex than the SIK, and harder to tune for good performance.
- The SIK was an order of magnitude faster than the SDK. The SDK was no faster than the HH model; however, the implementation of the SDK was not optimised and there is certainly room for some improvement in its efficiency.

- Interestingly, for a fixed duration data vector, the subthreshold potential produced by the SIK matched the HH model more closely than the potential produced by the SDK did. Despite this, the near-threshold behaviour of the SDK was closer to the HH model, which allowed it to predict spike times more accurately.

1.4 Overview of thesis

The remainder of this thesis is organised as follows: in chapter 2, a review of relevant papers in the literature is presented, including a description of the Hodgkin-Huxley model. Chapter 3 describes the models and methods used: firstly the SRM neuron model, the input current, and the specific implementation of the HH model are covered, and secondly, an overview of the methods used to fit the SRM to the input/output traces from the HH simulation is presented.

Chapters 4, 5 and 6 deal with the methods and results obtained, with one chapter focusing on each of the three kernels comprising our model. Firstly, chapter 4 describes how the spike shape kernel, η, was found, and presents the results of extracting this kernel from the membrane voltage. Chapters 5 and 6 deal with finding the κ kernel, and using this to compare the SRM predictions to the actual data. In chapter 5, the spike-time independent kernel (SIK) is explored, and chapter 6 deals with the spike-time dependent kernel (SDK).

Finally, chapter 7 presents conclusions and suggestions for further work building on this project.

Chapter 2

Background

A number of attempts have been made to simplify complex, multi-dimensional models such as the Hodgkin-Huxley model. The dynamics of the HH model become highly nonlinear during an action potential, so a central aspect of most simplified models is that they do not explicitly model spike generation. Instead, they introduce a firing threshold, such that the model is assumed to have emitted a spike if the membrane voltage rises above a certain threshold.

In section 2.1 below we will describe the HH model, and in section 2.2 we review the literature on some simple models that have been created to approximate to the HH model. Then in section 2.3 we will examine previous work on fitting neuronal models to data, and finally in section 2.4 we will review the work of Jolivet, Gerstner and colleagues, which builds on work in the areas of both simple models and fitting models to data. The notation used in equations has been altered in some cases to agree with the notation used by Jolivet et al. (2004), which was the primary reference paper for this thesis.

2.1 Hodgkin-Huxley model

In this section, we will describe the Hodgkin-Huxley (HH) model, which was developed by Hodgkin and Huxley (1952) by performing experiments on the squid giant axon. The model includes spike generation, which is controlled by ion channels that allow currents to flow into and out of the cell body (or *soma*). There are two main ion channels in the model, the sodium and potassium channels, which operate independently from one another. The channels tend to allow either a large current or very little current to flow, so may be thought of as being either open or closed. The open/closed state of each channel is governed by a set of *gating variables*; the Na^+ channel has two gating variables, m and h, and the K^+ channel has a single gating variable, n, which are all dependent on the membrane potential $u(t)$. At rest, the m and n variables are closed, and the h variable is open.

The sequence of events for generating a spike is as follows: as the membrane potential rises toward the firing threshold, the Na^+ activation variable m starts to open. This allows Na^+ to flow into the cell, which increases the membrane potential further and opens m more, and the resulting positive feedback loop causes a spike. However, at higher membrane voltages, the Na^+ inactivation variable h starts to close and causes

Table 2.1: Rate constant functions α and β for the three gating variables m, h, and n, as implemented in NEURON. These differ from the equations in Hodgkin and Huxley (1952) only in that they are offset to a resting potential of -65mV, and the sign of the membrane potential has been reversed.

Gating variable x	$\alpha_x(u)$	$\beta_x(u)$
m	$\frac{-(u+40)}{10\left[e^{-(u+40)/10}-1\right]}$	$4e^{-(u+65)/18}$
h	$0.07e^{-(u+65)/20}$	$\frac{1}{e^{-(u+35)/10}+1}$
n	$\frac{-(u+55)}{100\left[e^{-(u+55)/10}-1\right]}$	$0.125e^{-(u+65)/80}$

the Na^+ current to stop – in fact the spike is only possible because the m variable responds much faster than h, and a spike happens before h can deactivate and stop the inward Na^+ flow. Then at the peak of the spike, the n variable starts to open, which opens the K^+ channel. K^+ then flows out of the cell, causing the membrane potential to drop back down to (slightly below) the resting voltage and ending the spike.

The HH model consists of a main differential equation governing the evolution of the membrane voltage, $u(t)$, and three further differential equations governing the gating variables m, h, and n (which themselves control the Na^+ and K^+ currents). There is also a steady leak current due to the (relatively small) conductance of the membrane in the subthreshold regime. The change in membrane voltage is given by

$$c_m \frac{du(t)}{dt} = -\left[I_{leak}(t) + I_{Na}(t) + I_K(t)\right] + I_{ext}(t)$$

where c_m is the membrane capacitance, and $I_{ext}(t)$ is an external driving current. The remaining currents are defined by

$$\begin{aligned} I_{leak}(t) &= g_{leak}\left[u(t) - u_{leak}^{rev}\right] \\ I_{Na}(t) &= g_{Na} m^3(u,t) h(u,t)\left[u(t) - u_{Na}^{rev}\right] \\ I_K(t) &= g_K n^4(u,t)\left[u(t) - u_K^{rev}\right] \end{aligned}$$

where u_{ion}^{rev} is the voltage at which the *ion* current changes direction, and g_{ion} is the conductance of the *ion* channel. Each of the gating variables m, h, and n is governed by the differential equation

$$\frac{dx(u,t)}{dt} = \alpha_x(u)(1-x) - \beta_x(u)x$$

where x represents m, h, or n. Each gating variable therefore follows an exponential decay to an equilibrium value defined by the 'rate constants' α_x and β_x, with a voltage-dependent time-constant which is also defined by α_x and β_x. The rate constant functions for the gating variables (as used by the NEURON `hh` channel) are shown in table 2.1.

2.2 Simple versus complex models

In FitzHugh (1961); Nagumo and Yoshizawa (1962), a two-dimensional model was developed with the aim of duplicating the HH model's behaviour. In this FitzHugh-

Nagumo (FHN) model, the membrane voltage u is retained as one variable, and a slow variable w, similar to the gating variable n, is introduced. The model is given by the two differential equations

$$\begin{aligned} \frac{du}{dt} &= u - au^3 - w + I \\ \frac{dw}{dt} &= b(cu - dw) \end{aligned}$$

where a, b, c and d are constants, and I is the external current. This model captures much of the behaviour of the HH model (including generating action potentials), and has been widely used due to its relative ease of analysis.

Abbott and Kepler (1990) developed a similar two-dimensional model, but by an analytical reduction of the HH model. Firstly, the gating variable m from the HH equations, which has a very short time-constant, is replaced by its equilibrium value as a function of the membrane voltage. Secondly, a new variable w is introduced, and the gating variables h and n are replaced by their mean values as a function of this new variable. w is carefully chosen to provide a good compromise between the behaviour of h and n, and is defined by a differential equation involving the derivatives of u with respect to h and n. This model is more complex than the FHN model, but reproduces more of the behaviour of the HH model, and is easier to relate to the HH model as the variables were explicitly derived from it.

Kistler et al. (1997) create a simpler one-dimensional model, the Spike Response Model (SRM), which uses a firing threshold to generate spikes. Their aim is to match the Hodgkin-Huxley model as closely as possible in terms of predicting spikes., but they did not explicitly reduce the HH equations. Instead, their starting point was a Volterra series expansion, which provides a general model for a nonlinear time-series:

$$u(t) = \int_0^\infty \kappa_0^{(1)}(s) I(t-s)\, ds + \frac{1}{2!} \int_0^\infty \int_0^\infty \kappa_0^{(2)}(s,s') I(t-s) I(t-s')\, ds\, ds' + \ldots \qquad (2.1)$$

Here $\kappa_0^{(1)}(s)$ models linear response of the neuron to inputs, and $\kappa_0^{(2)}(s,s')$ models quadratic interactions between input current pulses at different times.

In our terminology, the equation 2.1 represents a Spike Independent Kernel (SIK) model. Kistler et al. build on this to create a model that takes into account the time elapsed since the last spike, so the refractory reduction of responsiveness following an action potential can be predicted. This Spike Dependent Kernel (SDK) model is given by

$$\begin{aligned} u(t) &= \eta_1(t-\hat{t}) + \int_0^\infty \kappa_1^{(1)}(t-\hat{t}, s) I(t-s)\, ds \\ &\quad + \frac{1}{2!} \int_0^\infty \int_0^\infty \kappa_1^{(2)}(t-\hat{t}, s, s') I(t-s) I(t-s')\, ds\, ds' + \ldots \end{aligned} \qquad (2.2)$$

where $\hat{t}$ is the time of the most recent spike. They have also introduced the η_1 kernel to represent the highly nonlinear course of an action potential, which is essentially independent of the input (once it has been triggered).

The kernels $\kappa_0^{(1)}$ up to $\kappa_0^{(3)}$ can be found analytically; however, the SDK $\kappa_1^{(1)}$ has to be found numerically, and only the first-order (linear) kernel is used. Kistler et al. explain the numerical derivation of both the SIK and SDK kernels. The first-order SIK kernel is equated with the response of the HH model to a small, sharp current pulse, and higher-order kernels are found by measuring the response to two sharp pulses separated by time δt, and repeating this for different values of δt.

To find the SDK $\kappa_1^{(1)}$, first the model is stimulated with a pulse large enough to initiate a spike, and then a second small pulse is applied at an interval δt after the spike. The kernel is extracted as the difference in membrane response with and without the second current pulse. Finally, η_1 is equated with the course of an action potential that is initiated by a sharp current pulse.

To evaluate how well the SRM performs, Kistler et al. define a *coincidence factor* Γ, which quantitatively measures how well the spike train produced by the SRM matches the HH model spike train. This is given by

$$\Gamma = \frac{N_{coinc} - \langle N_{coinc} \rangle}{\frac{1}{2}(N_{HH} + N_{SRM})} \frac{1}{\mathcal{N}} \tag{2.3}$$

where N_{HH} is the number of spikes produced by the HH model, N_{SRM} is the number of spikes in the test train, N_{coinc} is the number of coincidences in spikes with precision Δ between the two trains, and $\langle N_{coinc} \rangle$ is the number of spikes expected to coincide between the reference train and a homogeneous Poisson process. $\mathcal{N}$ is a normalising factor chosen to bound Γ in the range (-1,1). Note that a random homogeneous Poisson process would have a Γ value of 0.

The spike threshold was found by optimisation with respect to Γ, and results were produced for high firing rates (23-40Hz) and low firing rates (9Hz). For low firing rates, Kistler et al. note that "due to the large inter-spike intervals, effects from spikes in the past are rather weak", and therefore they use the SIK (but with the spike template η_1). Using just first order terms, they achieve $\Gamma = 0.79$, and with second and third order terms this rises to 0.86.

At high firing rates, the SDK with just the η_1 and linear terms is used. This gives very good performance, with Γ of around 0.9 for a variety of input current amplitudes. The SDK predicts approximately 90% of the spikes of the HH model correctly, compared to just 73% for the SIK, even with up to third-order terms.

Feng (2001) also compared the performance of simple models to the Hodgkin-Huxley model. He examined 4 simplified models: the classic integrate-and-fire model (IF) given by

$$\frac{du}{dt} = -\frac{1}{\tau_m}(u(t) - u_{eq}) + \frac{1}{C}I(t) \tag{2.4}$$

where τ_m is the membrane time-constant and C is the membrane capacitance, the IF-FHN model which is a variant of the above where τ_m is made a function of u, the SRM and the Abbott-Kepler model. He examined the behaviour of models presented with correlated inputs, which are expected to be biologically more realistic than uncorrelated inputs as spatially neighbouring cells often fire together (Zohary et al., 1994). IF models exhibited increased firing variability (as measured by the coefficient of variation of the inter-spike interval) with increased input correlation, but HH and IF-FHN

models had a less noisy output with increasing input correlation. The SRM and Abbott-Kepler models also exhibited behaviour close to the HH model, although the author suggested that care is needed to prevent the Abbott-Kepler model from predicting an infinite membrane potential when stimulated with random currents, and the SRM required more computational resources than the IF-FHN model.

2.3 Fitting neural models to data

Early work on fitting simple models to reference data was performed by Brillinger and Segundo (1979), who fitted a linear filter model to data from a real neuron. They assume that the membrane potential is given by

$$u(t) = \int_0^{\hat{t}} \kappa(s) I(t-s)\, ds \tag{2.5}$$

which is very similar to the SIK model proposed by Kistler et al. (1997), except that the integral runs only to the time of the last spike (and there is no zeroth-order term modelling action potentials). However, the potential is regarded as a hidden variable, and the model is fitted on the basis of the efferent spike times only. The model predicts spike times Y_t as the times at which u_t crosses a threshold θ_t, and a refractory period is used to ensure only a single spike is predicted when the model crosses the threshold.

Three different methods are examined for finding the kernel κ, a maximum-likelihood method, a method using cross-spectral analysis, and finally a cross-correlation approach. A maximum likelihood estimate is found by assuming that the threshold consists of a mean value θ with zero-mean, normally-distributed random values added at each time-step. Then the likelihood of the spike train given the input current can be estimated as normal. The cross-spectral method requires two assumptions: firstly that the input current is Gaussian distributed, and secondly that $\kappa(s)$ tends toward zero for high values of s, and therefore the upper limit of the integral in equation 2.5 can be replaced with ∞. It is then possible to derive the following equation relating the correlation of the spike train with the input to the covariance of the input:

$$cov\{Y_t, I_{t-v}\} = L \sum_{s=0}^{\infty} \kappa_s cov\{I_{t-v} I_{t-s}\} \tag{2.6}$$

where L is a constant (note that this is just the general solution for linear regression). This can be solved for κ with the aid of cross-spectral analysis. Finally, a cross-correlation approach can be used if the input is Gaussian white noise. In this case, equation 2.6 reduces to

$$cov\{Y_t, I_{t-v}\} = L\sigma_I^2 \kappa_v$$

giving the kernel κ very efficiently.

The results showed that all three approaches produce very similar κ functions, apart from a slightly faster tailing off to zero exhibited in the cross-spectral and cross-covariance solutions, which was attributed to changing the summation to be to infinity.

Experiments were also tried with a uniformly varying input, and even then the cross-spectral and cross-covariance functions were very close to the maximum-likelihood function, while being much faster to calculate.

Brillinger and Segundo did not explicitly compare the spike train produced by their model with the original spike train; instead, they investigated how the probability of a spike varied with their prediction of membrane potential. They found the probability of a spike rose rapidly from roughly zero to one in a small region around the optimum threshold voltage, which was a significant validation of the concept of using a thresholded model.

The maximum likelihood approach to the parameter fitting problem is also used by Paninski et al. (2004). They adopt a simple model which is based around a stochastic differential equation, using an additive Gaussian noise term, and a fixed threshold θ. Like Brillinger and Segundo, Paninski et al. consider the membrane voltage $u(t)$ to be a hidden variable, and fit model parameters from knowledge of the input stimulus and output spike times only. They show that the parameter space has no local minima for this problem, and present several efficient algorithms for calculating and optimising the log likelihood of the parameters of the model.

2.4 Jolivet and Gerstner work

In Jolivet et al. (2004), a conductance model of a neocortical neuron, as used by Erisir et al. (1999), is compared to two different forms of integrate-and-fire model. The conductance model is based on the HH model, but with an additional fast potassium channel, and with the parameters set to match the behaviour of a fast-spiking cortical neuron. The reduced models, which both introduce a spiking threshold θ, are firstly a non-linear integrate-and-fire model (NLIF), and secondly a Spike Response Model (SRM) as used in Kistler et al. (1997). The NLIF is given by

$$\tau_m \frac{du}{dt} = a_0(u - u_{eq})(u - u_c) + RI \tag{2.7}$$

where τ_m is the membrane time-constant, $u(t)$ is the membrane voltage, R is the resistance, and I is the external current. a_0 is a constant, u_{eq} is the equilibrium potential, and u_c is the critical voltage that will cause the model to spike if presented in a short pulse.

The SRM is given by

$$u(t) = \eta(t - \hat{t}) + \int_0^\infty \kappa(t - \hat{t}, s) I(t - s)\, ds \tag{2.8}$$

where η is a kernel modelling the shape of an action potential, κ is a kernel modelling the integration of the input, and $\hat{t}$ is the time of the last spike. This model can be thought of as a normal integrate-and-fire model with a time-varying time-constant. It also has a time-varying threshold, described by the function

$$\theta(t - \hat{t}) = \begin{cases} \theta^{refr} & \text{for } \hat{t} < t < \hat{t} + \delta_{refr} \\ \theta_0 + \theta_1 \exp\left(-\frac{t - \hat{t} - \delta_{refr}}{\tau_\theta}\right) & \text{for } t \geq \hat{t} + \delta_{refr} \end{cases} \tag{2.9}$$

where θ_0, θ_1 and τ_θ are parameters, and δ_{refr} is a refractory time following a spike, during which neuron is prevented from firing by the very high threshold value θ^{refr}.

An analytical method is used to reduce the conductance model to the single-variable NLIF model, following a similar procedure to Abbott and Kepler (1990). Variables other than $u(t)$ are removed by considering whether they are fast or slow compared to $u(t)$. The slow variables are replaced by constants, and the fast variables are replaced by their equilibrium values, as a function of u. However, the average values for these gating variables depend on the firing regime of the neuron – for example, for a simulation consisting of a single spike the averages are close to their resting values – so a particular regime must be assumed when constructing the NLIF model.

A three-stage numerical technique, in which the parameters of the SRM are fitted to a reference spike train generated from the full model, is used to reduce the conductance model to the SRM. Firstly, the kernel η is extracted by finding the average shape of a spike from the reference train. This is done by aligning all the spikes, using a gradient threshold value to determine the start of each spike. η is then subtracted from the reference potential data, to create a sub-threshold voltage trace which is used in the next stage.

Secondly, κ is determined by converting the SRM equation (equation 2.8) into a summation over discrete measurements, and then using the Wiener-Hopf optimal filtering algorithm, with suitable modifications for this problem as detailed in an appendix in the Jolivet et al. (2004) paper.

Finally, the varying threshold given by equation 2.9 is fitted to the data, using the downhill simplex method from Nelder and Mead (1965) to maximise the similarity of the spike train produced by the model to the reference spike train.

Jolivet et al. performed experiments using four different types of input: a constant input current, a short current pulse, a randomly fluctuating current, and a conductance input composed of many Poisson-spiking synapses. Quantitative analysis of the results was made using the same coincidence factor Γ defined by Kistler et al. (1997), given in equation 2.3.

For current pulses, results were good for inputs significantly above or below the threshold for producing a spike, but only the NLIF model was capable of reproducing the delayed spike behaviour of the full model for inputs close to the spike threshold.

For constant input current, both integrate-and-fire models showed very good agreement with the full model for currents close to the input the parameters were optimised for, matching voltage traces almost exactly for periods covering a few action potentials. However, the SRM model, which includes a variable spike threshold, also reproduced the firing frequency / current curve of the full model over a large range of currents.

For random currents, both reduced models produced a good match to the conductance model over a wide range of parameters, with the SRM generally achieving higher Γ coincidence values than the NLIF. In the case of the SRM, for currents with mean and variance similar to the parameter-fitting input, a Γ value of 0.83 was found, and for most of the mean/variance range tested Γ was above 0.7. For both integrate-and-fire models, the fit was less good when the mean of the input current was much larger than its variance, as then any errors when a spike was missed were compounded over time.

For the conductance-based input, a reference spike train was obtained by stimulating the conductance-based model via 10,000 simulated stochastic synapses, and the

parameters of a modified version of the SRM were fitted to this data. The integrate-and-fire model provided a reasonable match to the full model at the point the parameters had been optimised for, where the Γ factor was 0.7, but for different input rates the correspondence between the two models tailed off rapidly.

In a later paper, Jolivet et al. (2006) applied the SRM to real biological data, using a simplified kernel $\kappa(s)$ which depends on s only, but incorporates a slow adaptation into the threshold:

$$\frac{d\theta}{dt} = -\frac{\theta - \theta_0}{\tau_\theta} + A_\theta \sum_k \delta(t - t_k) \tag{2.10}$$

where t_k are the past firing times of the neuron. This models a slow decay of the firing rate given a constant input, and helped the SRM achieve a very good fit to the biological data, with the Γ factor being about 65% of the best achievable correlation.

Chapter 3

Models and methods

In this chapter, we firstly review the models and data used, and then provide a short description of our methods. We describe the SRM, which is a simple model based on the integrate-and-fire model, in section 3.1, and then describe the data used to fit the parameters of the SRM. These data consist of firstly the input current, described in section 3.2, and secondly the membrane potential produced in response to this input by the HH model, described in section 3.3. Finally we provide a short description of the methods used to fit the data, which largely follow those used by Jolivet et al. (2004). We also derive the mathematical foundation of these methods.

3.1 SRM

The Spike Response Model is an extension of the classic leaky integrate-and-fire model which allows a more complex response to inputs, includes an explicit template for the shape of spikes, and (in the SDK version) allows the response of the neuron to vary at different δt values since the last spike. This is the model for which we have chosen to optimise the parameters, by fitting them to data from the HH model.

The discrete-time version of the SRM (equation 1.1) is

$$u_t^{SRM} = \eta_{t-\hat{t}} + \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} \tag{3.1}$$

when using the SIK (Spike Independent Kernel), and

$$u_t^{SRM} = \eta_{t-\hat{t}} + \sum_{s=0}^{s_{max}-1} \kappa_{t-\hat{t},s} I_{t-s} \tag{3.2}$$

for the SDK (Spike Dependent Kernel).

We assume that κ will decay to zero after a finite time, which is borne out by experimental results. This allows us to sum κ over a lag of 0 to s_{max} steps back in time, rather than having to sum back to infinity.

Note that in future, we will use the term SRM to refer to the model above, rather than other models introduced by Kistler et al. (1997) such as those including higher-order terms. However we refer to both the SIK and SDK versions of the κ kernel by the term SRM.

3.2 Input current

We decided to use a zero-mean, Gaussian-distributed random current, following the type of input used in most experiments in Jolivet et al. (2004). This is a much more biologically realistic regime than a constant input current (which is also used widely in neural modelling), as in-vivo input to a neuron is generated by the irregular firing of tens or hundreds of presynaptic neurons, producing a fluctuating somatic current (Calvin and Stevens, 1968).

A white noise input was generated, where the current value was changed on every time-step. This was the simplest approach, and produced a biologically plausible output voltage trace from the HH model. A more smoothly varying current (either keeping the current constant for several time-steps, or using a "random walk" type input such as an Ornstein-Uhlenbeck process) turned out not to be necessary, but would certainly be of interest for future research.

3.3 Hodgkin-Huxley implementation

The complex neuron model used was a straightforward implementation of a single compartment cell in the NEURON[1] simulation package. Hodgkin-Huxley conductances were added to the compartment using the HH components built in to NEURON, which use the original equations from Hodgkin and Huxley (1952) as described in section 2.1. The output voltage was recorded at a position half-way along the somatic compartment.

3.4 Technical details

The technical architecture consisted of two components: code to produce the input current and HH membrane potential data, and code to fit and evaluate the SRM based on this data.

3.4.1 Input data

MATLAB[2] was used to create a vector of input current values, which was exported to a file. The time-step used was 0.025ms, which is the default setting for simulations in NEURON. For most experiments, data vectors of 2s length were used with the SIK, and 100s vectors with the SDK; however, sometimes the SIK was tested on the 100s data for comparison purposes, and also longer data of lengths 450s and 900s were used when initially developing the SDK code.

MATLAB's implementation of Marsaglia's ziggurat algorithm was used to generate the random sequence used for the input current. The standard deviation of the current was 35nA, which was chosen by finding the minimum current that caused the HH model to fire regularly. The rate of the spike train produced with this input was 33Hz.

[1] http://www.neuron.yale.edu/neuron/
[2] http://www.mathworks.com/products/matlab/

The stimulus file was then read in by NEURON, and used to set the current provided by an `IClamp` object via the `Vector.play` method. When modelling with NEURON, it is common practice to let the membrane voltage settle to a resting level before applying a stimulus current. We chose to let the model settle for 25ms before applying the input current, and also to continue recording the membrane voltage for 75ms after the current had ended. Note that to achieve this, 25ms of zero data had to be generated by the MATLAB code, because setting the delay attribute of NEURON's `IClamp` object just resulted in the first 25ms of the stimulus being zeroed, rather than the stimulus being put back by 25ms. For fitting the model, only data from when the stimulus was active were used.

After the NEURON model had created the HH voltage data, this was exported to a ASCII file, and then converted into MATLAB format, as a significant performance advantage was found in loading `.mat` files into MATLAB as compared to ASCII files.

3.4.2 Fitting procedure

The SRM fit and evaluation was performed entirely in MATLAB. There were two master scripts, one for the SIK and one for the SDK, which delegated the main processing to functions for finding the kernels, plus several utility functions. The output of the system was mainly graphical plots, but also some numerical values for the SMSE and Γ measures (as defined in section 3.5.5). It was not considered worthwhile to write any results to file automatically.

3.5 Overview of the fitting procedure

3.5.1 Action potential template

To find η, the starting points of spikes in the HH voltage data are detected by noting when the voltage gradient rises above a fixed threshold value. The length of the η kernel is chosen to be long enough to include the action potential and following after-hyperpolarisation, until the voltage returns to roughly the resting value. The kernel is then calculated as the average membrane potential over all sets of data starting at a spike and extending the fixed duration chosen.

The subthreshold voltage, $v(t)$, can then be found by subtracting η from $u(t)$:

$$v_t^{data} = u_t^{data} - \eta_{t-\hat{t}}$$

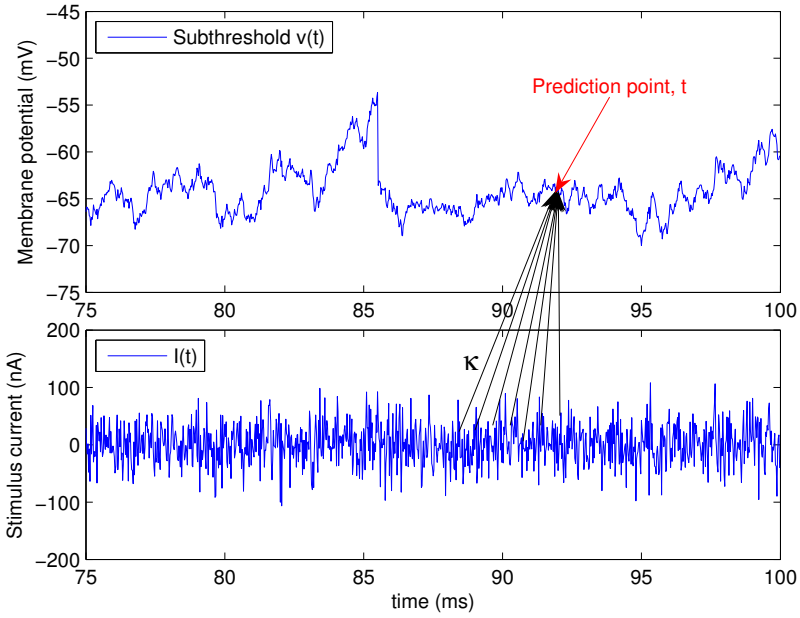
3.5.2 SIK

Once we have the subthreshold voltage v^{data}, finding the weights κ is essentially just a linear regression problem. Figure 3.1 shows a diagram of how this works.

The approach used is to calculate the Wiener-Hopf optimal filter, as described in Jolivet et al. (2004). This is a least-squares method, where the difference between v_t^{data} and the voltage predicted by the SRM,

$$v_t^{SRM} = \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} \tag{3.3}$$

Figure 3.1: A diagram showing how κ is used to predict the membrane voltage at time t based on the recent history of the input current



is minimised. The squared error is therefore given by

$$error = \sum_t \left(v_t^{data} - \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} \right)^2 \tag{3.4}$$

The derivative of the error with respect to element j of the filter κ is

$$\frac{\partial error}{\partial \kappa_j} = 2 \sum_t I_{t-j} \left(v_t^{data} - \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} \right) \tag{3.5}$$

Setting $\frac{\partial error}{\partial \kappa_j} = 0$, we find

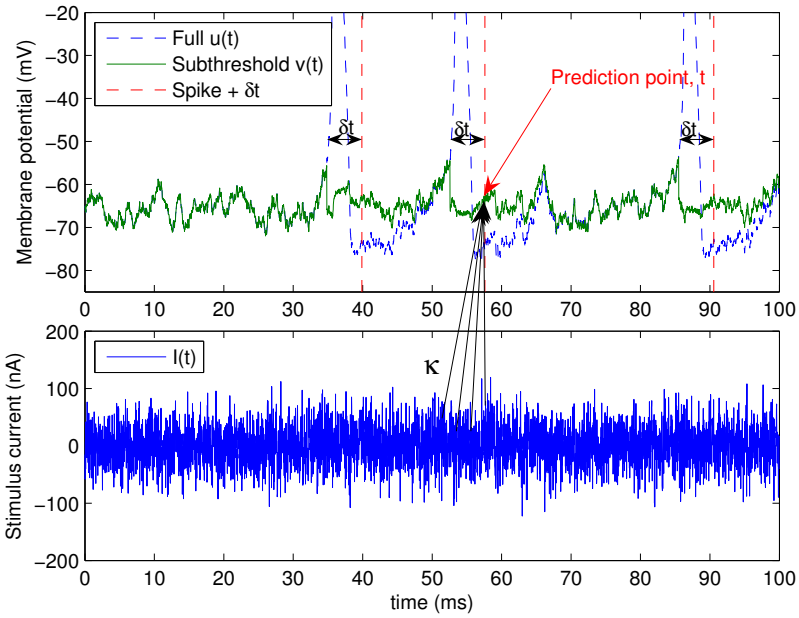
$$\begin{aligned} \sum_t v_t^{data} I_{t-j} &= \sum_t \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} I_{t-j} \\ \sum_{s=0}^{s_{max}-1} \kappa_s \sum_t I_{t-s} I_{t-j} &= \sum_t v_t^{data} I_{t-j} \end{aligned} \tag{3.6}$$

for $j = 0..(s_{max}-1)$. κ is then found by solving this set of simultaneous equations. Note that this is in fact the same result as found by Brillinger and Segundo (1979) (equation 2.6) using covariance analysis, if you equate the correlation of the input and response with $\sum_t v_t^{data} I_{t-j}$, and the covariance of the input at lag $j-s$ with $\sum_t I_{t-s} I_{t-j}$.

3.5.3 SDK

When κ depends on the time since the last spike, $\delta t = t - \hat{t}$, as well as the lag, a different κ will have to be found for each δt value. This means that data for each spike in the HH

Figure 3.2: A diagram showing how the SDK κ is used to predict the membrane voltage at time t based on the recent history of the input current, for a given offset since the last spike δt



voltage train has to be 'aligned' at the start of the spike, and each spike will provide only one data point representing the target v_t value at each value of $t-\hat{t},s$. This is shown diagramatically in figure 3.2.

Mathematically, the SDK is found in a very similar way to the SIK, but with the summations performed over the T spikes, instead of over all data points. So rather than summing over all t values, $\sum_t f(t)$, we sum over spike times offset by δt, $\sum_i f(\hat{t}_i+\delta t)$. This means that the SDK requires a much longer input data vector to find κ, as there will be only as many effective data points as there are spikes in the data. The equations 3.4, 3.5, and 3.6 are all still valid if the summations changed to be over spike times, and so κ for each δt is given by

$$\sum_{s=0}^{s_{max}-1} \kappa_{\delta t,s} \sum_{i=1}^{T} I_{\hat{t}_i+\delta t-s} I_{\hat{t}_i+\delta t-j} = \sum_{i=1}^{T} v^{data}_{\hat{t}_i+\delta t} I_{\hat{t}_i+\delta t-j}$$

3.5.4 Threshold

Jolivet et al. (2006) use a variable firing threshold. Due to time constraints on this project, we did not implement a variable threshold, but instead optimised the fixed threshold θ with respect to the coincidence factor Γ (equation 2.3). However, an absolute refractory period was used, whereby the neuron was prevented from firing at all for 6ms after a spike.

3.5.5 Evaluation

Two measures were used to quantitatively evaluate how well the SRM predicted the response of the HH model. The first is the Standardised Mean Square Error (SMSE), which compares the subthreshold voltage traces, $v(t)$. It is defined by

$$SMSE = \frac{\sum_t \left(v_t^{data} - v_t^{SRM}\right)^2}{\sum_t \left(v_t^{data} - \overline{v_t^{data}}\right)^2}$$

where $\overline{v_t^{data}}$ is the mean value of the subthreshold voltage for the HH model. The SMSE value for the naive approach of predicting the mean for all values of $v(t)$ will be 1, so a good value for the SMSE should be significantly less than 1.

The second measure is the Γ coincidence factor as defined by Kistler et al. (1997), which describes how well the spike times predicted by the SRM match the reference data's spike times. It is given by

$$\Gamma = \frac{N_{coinc} - \langle N_{coinc} \rangle}{\frac{1}{2}(N_{HH} + N_{SRM})} \frac{1}{\mathcal{N}}$$

(Γ was defined in equation 2.3 above, but is repeated here for convenience). Here, N_{coinc} is the number of coincidences in spikes with precision Δ between the two spike trains, where we set Δ to be 2ms for all experiments. $\langle N_{coinc} \rangle$ is the number of spikes expected to coincide between the HH spike train and a homogeneous Poisson process, and it is given by $\langle N_{coinc} \rangle = 2\nu\Delta N_{HH}$, where ν is the spike frequency of the SRM. N_{HH} is the number of spikes produced by the HH model, N_{SRM} is the number of spikes in the test train, and $\mathcal{N} = 1 - 2\nu\Delta$ is a normalising factor chosen to bound Γ in the range (-1,1).

If the model correctly predicted every spike in the HH data, and only those spikes, it would have a perfect Γ value of 1. If it predicted spikes randomly, the Γ value would be 0 (negative Γ values indicate a reverse correlation).

Chapter 4

Fitting the η kernel

In this chapter, we discuss in detail the methods and results obtained for the η-fitting procedure. The η kernel is intended to capture the nonlinear dynamics that occur in the HH model during a spike, so that these effects can be removed from the underlying sub-threshold potential.

4.1 Basic approach

The start of a spike was found by finding the point at which the gradient of the membrane potential, $u(t)$, exceeds a threshold value. This approach worked better than the obvious method of taking the start of a spike to be when a threshold in the potential is exceeded, because the gradient threshold is exceeded at the same point in the progression of the spike for all spikes.

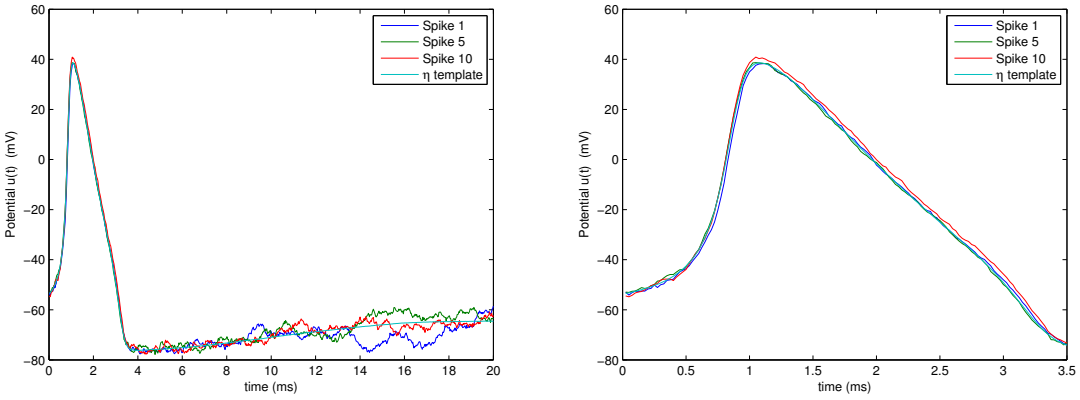
The gradient used was just the instantaneous gradient between adjacent data points, which has a very sharply defined peak coincident with action potentials. Spikes found by this method were found to have almost exactly the same shape, with no rescaling of individual spikes along either the vertical or horizontal axis being required. Similarly, no smoothing of the resulting average action potential function (i.e., the η kernel) was needed. Figure 4.1 shows several spikes found, and the resulting η kernel.

In finding the η kernel, it was necessary to decide upon a duration for the kernel. The aim was to cover a period long enough to include all of the nonlinear effects due to the action potential, but not so long that essentially random subthreshold fluctuations are included in the tail of η. Three possible approaches were considered:

- Choose a fixed duration for η
- Wait until the membrane potential returns to within a small margin of the resting potential
- Wait until the gradient falls below a fixed threshold

We chose to use the first method, because the other two rely on measurements that will vary widely between spikes. As the same duration had to be used for all spikes, so that spikes can be aligned and averaged, it seemed to be an easier option to choose and optimize a fixed length for η.

Figure 4.1: Alignment of spikes, and the resulting η kernel.



(a) Spikes shown over the full kernel duration

(b) Zoom of the first 3.5ms of spikes

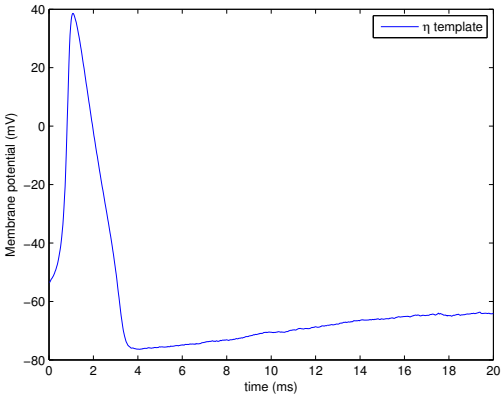
The duration used was 20ms. This produced an averaged spike shape that returned to within 0.4mV of the resting potential. Experiments confirmed that longer or shorter durations gave worse values for the SMSE for $v^{SRM}(t)$ compared to $v^{data}(t)$, using the SIK kernel.

Once the kernel had been found, it was subtracted from $v^{data}(t)$ at all points where there was a spike. To do this, the kernel had to be shifted along the y-axis, as otherwise subtracting the kernel would lead to the subthreshold voltage fluctuating around zero where η had been subtracted, and around the resting potential at other points.

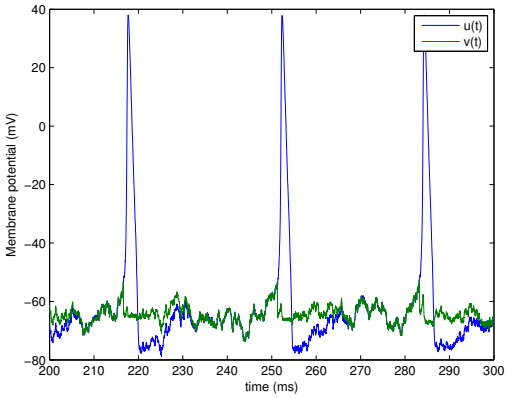
There were several choices for the offset that would be applied to η. One choice was the resting potential of the model, which was -64.97mV, which is approximately the value that the subthreshold voltage fluctuates around. However, there is no reason to expect that the average value for the neuron when driven by a current will be the same as the resting value, just because the current is zero-mean. Therefore it was felt that using the resting voltage could be introducing a small bias into the SRM parameters extracted. Another option was to use the mean value of $u^{data}(t)$, -61.9mV, but this included a significant offset due to spikes and was therefore not felt to be a good value for the subthreshold baseline. Instead, the value chosen was the mean of $u^{data}(t)$ for datapoints that fell outside the 20ms window following spikes, i.e. the mean over data where η would not be extracted. This value was -64.59mV, and experiments confirmed a small improvement in the SMSE using this as compared to using the resting potential as the baseline.

Graphs of the subthreshold voltage $v^{data}(t)$ obtained after subtracting the η template are shown in figure 4.2. Figure 4.2 (c) and (d) show that the variance in the subthreshold voltage is approximately the same in the regions where η has been subtracted as it is in the other regions, which indicates that the procedure was generally successful. Figure 4.2 (c) shows that there is still a sharp upwards trend in the subthreshold voltage before a spike; however, it is important that the subthreshold voltage preserves some pre-spike increase in potential, as we require the SRM's predictions to show a gradual rise in voltage leading up to a spike, in order for it to reach the threshold

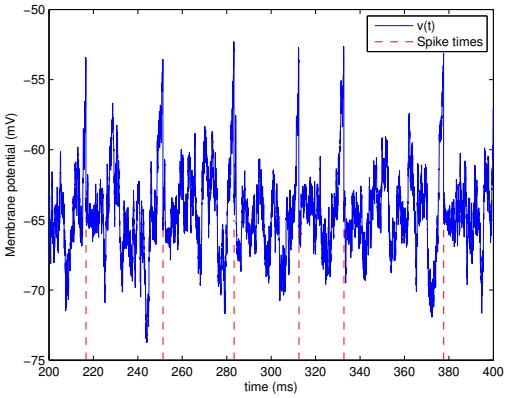
Figure 4.2: Graphs showing the results of the η extraction process.



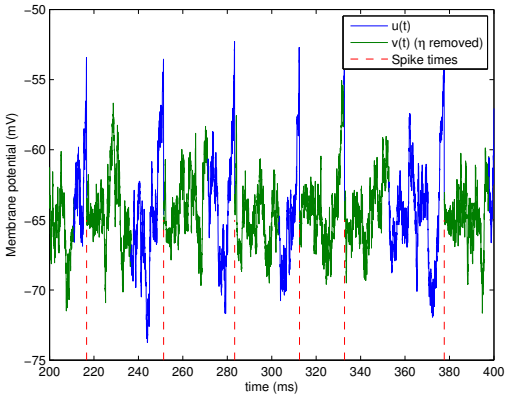
(a) The η kernel



(b) The subthreshold voltage v(t) shown against the full membrane potential



(c) The subthreshold voltage, shown with spike times



(d) The subthreshold voltage, where regions where η has been subtracted are marked in green

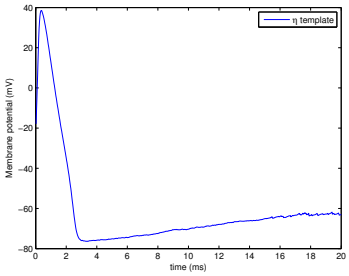
and correctly predict that spike.

4.2 Fine tuning

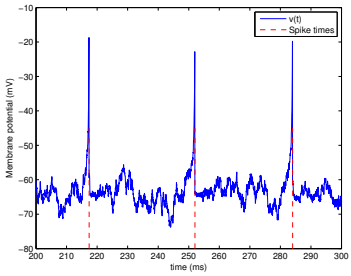
In this section, we discuss some of the issues relating to the η kernel that had to be addressed to obtain reasonable performance from the SRM.

Spikes are detected using a gradient threshold, which is possible because the gradient at the start of a spike is much higher than anywhere else in the data. However, this means that at the point the gradient threshold is exceeded, the spike is already "in progress" and the membrane potential is considerably higher than the baseline level. This means that even after η is removed from the voltage trace, there are "stalks" where the voltage rises sharply coincident with the action potentials, as can be seen in figure

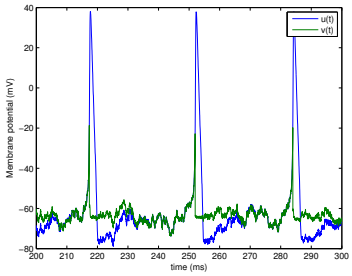
Figure 4.3: Comparison of the η kernels and effects of its removal for voltage gradient thresholds of 200mV/ms and 100mV/ms.



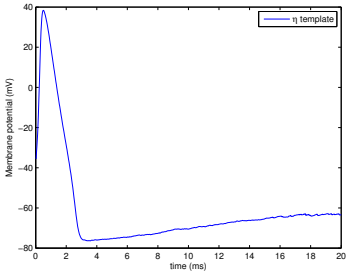
(a) η kernel with threshold 200mV/ms.



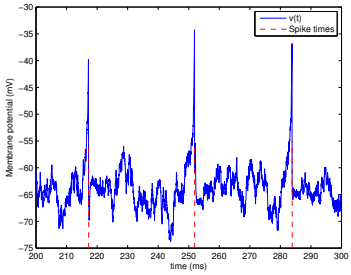
(b) Subthreshold voltage, showing large 'stalks' at spike times



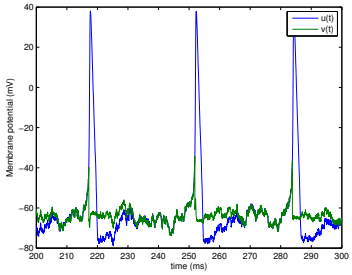
(c) u(t) against v(t) for 200mV/ms gradient threshold.



(d) η kernel with threshold 100mV/ms.



(e) Subthreshold voltage after removing η found with threshold 100mV/ms. Note the voltage scale is much larger than for subfigure (b).



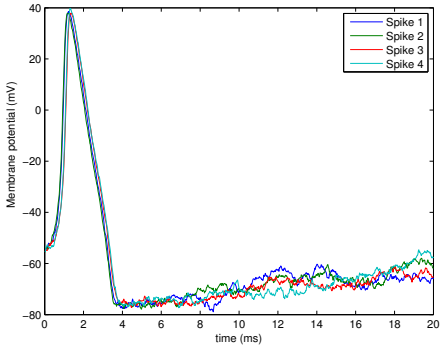
(f) u(t) versus v(t) for 100mV/ms gradient threshold, showing significant 'stalks' remaining in the subthreshold trace.

4.3 (b) and (c).

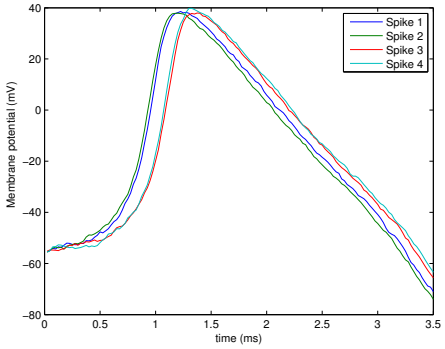
While a rise in voltage leading up to a spike is desirable in the subthreshold trace, as we need the SRM to reach a spike threshold in order to trigger a spike, we suspected that the very sharp stalks of figure 4.3 (b) included some unwanted nonlinear effects. Also, these stalks were high enough and wide enough that we felt they could introduce a significant bias into the κ kernels calculated by the cross-correlation approach. Therefore, we tried using the lowest gradient threshold possible without finding phantom spikes, in order to catch the start of the spike at a lower voltage and therefore reduce the size of the stalks in the subthreshold potential. Figure 4.3 (d) shows the kernel found with the gradient threshold reduced to 100mV/ms, and it starts at a significantly lower voltage than the kernel shown in 4.3 (a), which was found using a threshold of 200mV/ms,. Figure 4.3 (e) and (f) show the subthreshold voltage with this reduced gradient threshold. However, using the lower threshold has failed to remove the stalks – it has just made them slightly shorter.

Two alternative strategies were tried to remove the unwanted stalks: firstly, finding the location of the spikes using a threshold gradient, but then following the trace back

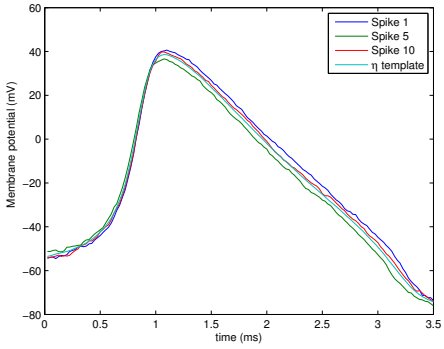
Figure 4.4: Spike alignment with a composite gradient/voltage threshold compared to a gradient threshold and time offset.



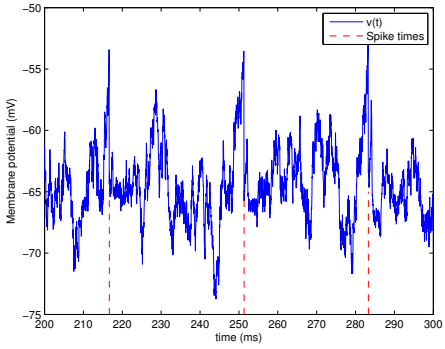
(a) Spike alignment using gradient of 100mV/ms and then a voltage threshold of -55mV.



(b) Zoom on spikes found with a gradient threshold of 100mV/ms, followed by a voltage theshold of -55mV.



(c) Zoom of spike alignment using a voltage gradient of 180mV/ms followed by an offset of 0.7ms.

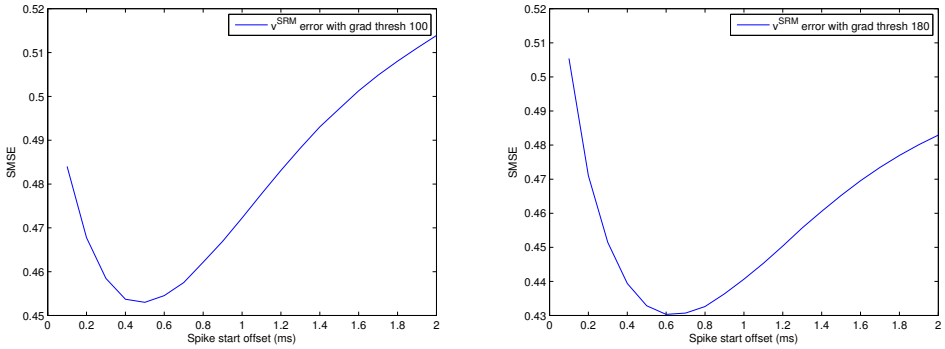


(d) Subthreshold voltage using a voltage gradient of 180mV/ms followed by an offset of 0.7ms

in time until the potential fell below a voltage gradient, and taking the start of the spike to be where that voltage was exceeded. The problem with this was that the alignment of spikes became much worse, as can be seen from figure 4.4. The second idea was to take the start of the spike as a fixed time before the gradient threshold was crossed; this method worked well, preserving spike alignment and considerably reducing the size of the stalks, as shown in figure 4.4 (d). The optimum setting for the offset was found to be 0.7ms, as shown in figure 4.5, and results for the SMSE for different values of the parameters discussed above are given in table 4.1.

Finally, when working with longer data vectors it became apparent that the gradient of 100mV/ms was causing phantom spikes to be detected. Figure 4.6 (a) shows a spurious spike detected due to a sharp, non-spike rise in the membrane potential u^{data}. The problem also resulted in some missed spikes, as shown in figure 4.6 (b). In this case, a rise in the membrane voltage caused a phantom spike to be detected just before

Figure 4.5: Optimisation of the spike start offset using the SMSE measure for v^{SRM} against v^{data}, using the SIK.



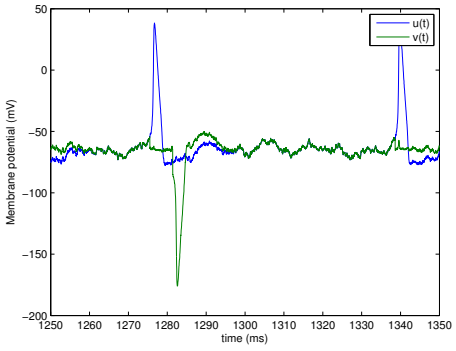
(a) Gradient threshold of 100mV/ms

(b) Gradient threshold of 180mV/ms

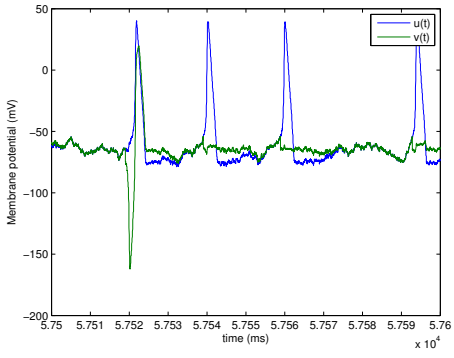
Table 4.1: Results for the SMSE of v^{SRM} against v^{data} for different values of the gradient threshold and spike offset. All results were recorded using the SIK and a 2s data vector.

Grad threshold	Spike offset	SMSE
200	0	0.57
100	0	0.51
100	0.5	0.45
100	0.7	0.46
100	0.9	0.47
180	0.5	0.43
180	0.7	0.43

Figure 4.6: Extra spikes and missed spikes using a gradient threshold of 100mV/ms with the 100s data vector.



(a) A down-spike appears in the subthreshold voltage because a false spike was detected at 1281ms.



(b) Here, a spike was missed because a false spike was detected just before it.

a real spike; as the fitting procedure does not allow a spike to be detected immediately after another spike, the real spike was then missed.

To avoid finding these phantom spikes, we switched to finding the largest gradient that detected all spikes, instead of finding the smallest gradient that did not detect non-spikes. As the spikes now started at an offset from where the gradient threshold was exceeded, this had no detrimental effects on the size of the unwanted subthreshold 'stalks'.

Note that all figures presented in section 4.1 were obtained using the final parameters, a gradient threshold of 180mV/ms and a start time offset of 0.7ms, and the 100s data vector.

Chapter 5

Fitting the κ kernel (SIK)

In this chapter, we discuss the methods used to find the spike independent κ kernel from the voltage data produced by the HH model. We also discuss the results obtained for κ, and for the SRM as a whole when using the SIK.

5.1 Maths and implementation

Firstly, as the stimulus is defined by a zero-mean stationary time series, we note that the model as given by equation 3.3 is only capable of predicting zero-mean $v^{data}(t)$:

$$\begin{aligned} \langle I_{t-s} \rangle &= 0 \\ \left\langle v_t^{SRM} \right\rangle &= \left\langle \sum_{s=0}^{s_{max}-1} \kappa_s I_{t-s} \right\rangle = 0 \end{aligned}$$

This means that the $v^{data}(t)$ vector must be normalised to be zero mean before being used in the fitting procedure to avoid introducing a bias into the resulting κ kernel (this avoids having to use an additional parameter in equation 3.3 to represent a fixed offset).

Mathematically, the SIK implementation is based around equation 3.6. If we define the cross-correlation vector $\mathbf{x}$ by

$$x_j = \sum_t v_t^{data} I_{t-(j-1)} \tag{5.1}$$

for $j = 1..s_{max}$, and the covariance matrix C by

$$C_{sj} = \sum_t I_{t-(s-1)} I_{t-(j-1)} \tag{5.2}$$

where $s, j = 1..s_{max}$, then the vectorised form of equation 3.6 is

$$C\kappa = \mathbf{x} \tag{5.3}$$

(where we have used the fact that $C_{sj} = C_{js}$). In equations 5.1 and 5.2, $(j-1)$ and $(s-1)$ are used in the subscript indices because s and j are matrix indices, and therefore start from 1 instead of 0. This also reflects the implementation more accurately, as MATLAB indices start from 1.

Note that the input current is a stationary time-series, so C_{sj} depends only on $|s-j|$, making C a symmetric Toeplitz matrix. This means that rather than calculate the full covariance matrix C, we can calculate the first row and use the `toeplitz` function in MATLAB to fill in the remaining elements of the matrix. This approach will also be valid for correlated inputs such as an Ornstein-Uhlenbeck process, which still defines a stationary process.

To solve equations of the form of equation 5.3 in MATLAB, rather than invert the matrix, we can use the backslash operator:

```
kappa = C \ x;
```

which is faster and more accurate than matrix inversion. This solution was regularised by adding a small multiple of the identity matrix to C, before applying the backslash operator:

$$C_{reg} = C + \lambda I \tag{5.4}$$

This was to avoid overfitting, and promote numerical stability in the solution.

The cross-correlation vector $\mathbf{x}$ and the first row of the covariance matrix C are calculated by multiplying the elements of $v^{data}(t)$ and $I(t)$ by $I(t-j)$ and then finding the sum of the resulting vector, for all offsets $j = 0..(s_{max}-1)$:

```
x(j) = sum(vData(indices) .* iStim(indices-(j-1)));
C(j) = sum(iStim(indices) .* iStim(indices-(j-1)));
```

The first s_{max} elements of the $v^{data}(t)$ and $I(t)$ vectors are ignored, so that data is available for a lag of s_{max} from all elements (i.e., the range of the variable `indices` in the above code fragment is `sMax:end`).

Finally, the duration of the kernel, as defined by s_{max} (which is the length of κ in data points), had to be chosen. Using s_{max} is actually an approximation, as the summation should be to infinity, so the aim is to choose s_{max} large enough that κ decays to zero (but small enough to be computationally efficient). The output kernel κ was plotted for various values of s_{max} to find the earliest duration for which κ settled to zero. Also, the values of the SMSE and Γ measures were plotted against different durations, to ensure that a length for κ that optimised the overall fitting procedure was chosen. Results for this procedure are given in section 5.3.1.

5.2 Reconstruction

Once κ had been found, the SRM model had to be applied to produce the reconstructed voltage trace, $u^{SRM}(t)$. The SRM is defined by equation 3.1, and we applied it in two stages: firstly the subthreshold potential $v^{SRM}(t)$ was generated, and then the η kernel was added to it at spike times.

To create the subthreshold voltage, κ was convolved with the input current using the MATLAB function `conv`. The formula applied by `conv`[1] is

$$v(t) = \sum_{j=max(1,t+1-len(I))}^{min(t,s_{max})} \kappa(j) I(t+1-j)$$

[1] This is taken from `http://www.mathworks.com/access/helpdesk/help/techdoc/ref/conv.html`, but with the variable names changed to match our notation

This is actually exactly what we want based on equation 3.3, as MATLAB indices run from 1 instead of zero, and the limits on j just ensure that the calculation is performed over all valid indices.

Spikes were added whenever the subthreshold voltage crossed the fixed threshold θ from below, as defined in Jolivet et al. (2004)

$$\text{if } v^{SRM}(t) \geq \theta \text{ and } \frac{d}{dt}v^{SRM}(t) > 0 \Rightarrow \hat{t} = t$$

An absolute refractory period of 6ms was included, so the model could not spike again for 6ms after a spike. The η template was added immediately, so for a period extending the duration of η after a spike, the test for crossing the threshold was carried out on the membrane potential as corrected by η. If the threshold was exceeded during this period, η would be added to the already-corrected potential; however, as η included an afterhyperpolarisation period where it reduced the potential, there were not many spikes within this period, and adding η to the voltage (potentially) several times did not cause any unusual features in the reconstructed $u^{SRM}(t)$ data.

A sanity check of the fitting procedure was conducted by generating a voltage trace using a known kernel (and the same random stimulus), and seeing how well the fitting procedure was able to reproduce the kernel and the voltage trace. With a simple exponential decay kernel and using 200ms of data, an SMSE of 0.000357 was obtained, and the reconstructed voltage trace was indistinguishable from the original.

5.3 Results

5.3.1 Setting s_{max}

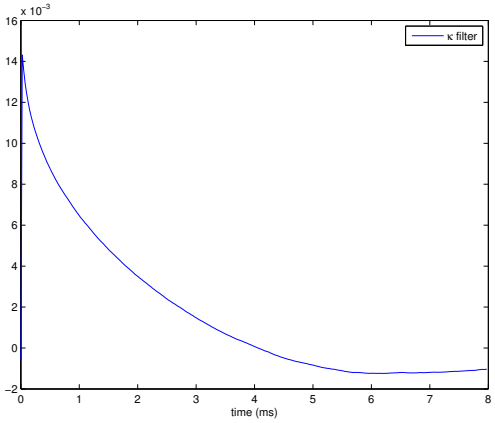
Initially, an 8ms duration was tried for κ, as this was the duration used by Jolivet et al. (2004). The kernel extracted is shown in figure 5.1 (a), and it clearly has not settled to zero after 8ms. Figure 5.1 (b) shows the kernel with a duration of 20ms, which settles to zero after approximately 14ms.

In figure 5.1 (c) we plot the SMSE of the reconstructed subthreshold voltage against κ length, and in 5.1 (d) the Γ spike coincidence factor against κ length. The Γ factor shown was found by optimising the firing threshold θ with respect to Γ (using the `fminbnd` function in MATLAB with a tolerance of 0.1mV in θ). These two graphs show no significant improvement in performance of the SRM with kernels longer than 12ms; in fact, between a length of 8ms and 20ms, the SMSE only improves from 0.430 to 0.422. Therefore, a duration of 12ms was decided upon for the κ filter.

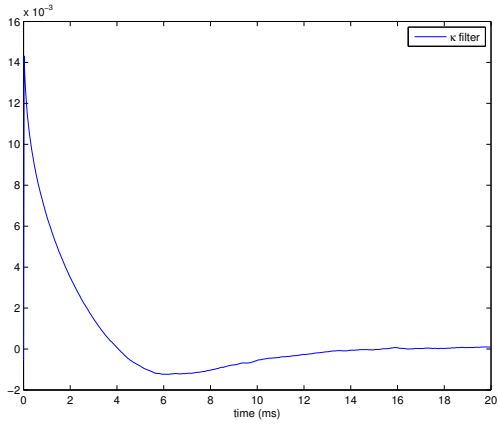
Note that in all plots of κ in this thesis, increasing lag is shown on the x-axis; however, with respect to the input stimulus, this is actually going backwards in time. This is because the lag is measured backwards from the prediction point, so κ graphs show the weight of the kernel for inputs going further back in time with increasing x values.

As can clearly be seen in figure 5.1 (b), the κ kernel is negative between approximately 4 and 12ms. This means the model is showing a resonance with the input — a rise in input current between 4 and 12ms in the past will produce a drop in the output voltage. This is due to activation of the potassium channels, and is expected behaviour for the HH model (Koch, 1999).

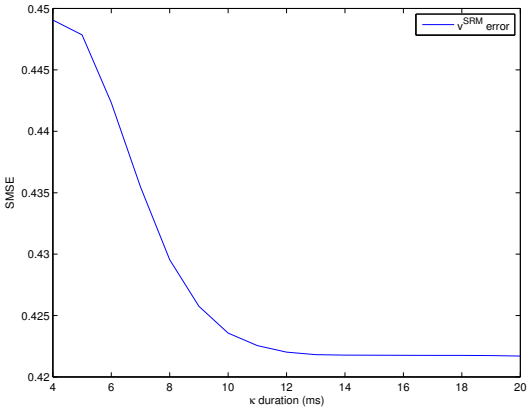
Figure 5.1: Results motivating the choice of 12ms as the duration of κ. These graphs are using the 100s data vector, and no regularisation.



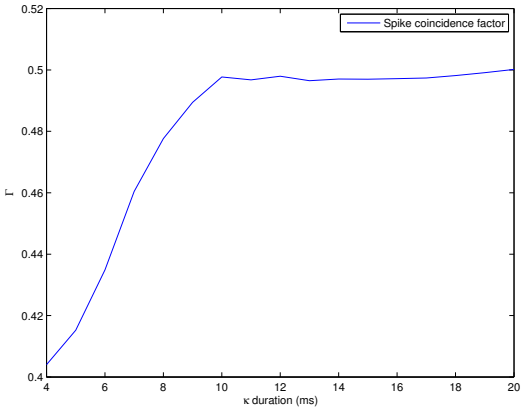
(a) The κ kernel, length 8ms.



(b) The κ kernel, length 20ms.

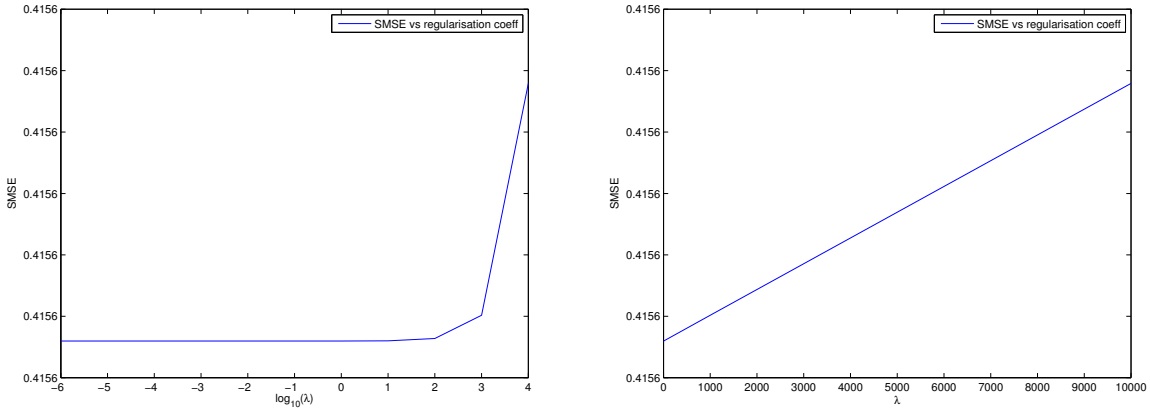


(c) SMSE error versus κ length.



(d) Γ spike coincidence factor versus κ length.

Figure 5.2: Impact of regularisation on the performance of the SRM, using the 100s data vector. Note the values on the y-axes are identical to 4 significant figures.



(a) Logarithmic scale, $\lambda = 10^{-6} : 10^{4}$

(b) Normal scale, $\lambda = 10^{-6} : 10^{4}$

5.3.2 Regularisation

The optimum amount of regularisation to use was found by running the SRM using κ found from a training data set on a separate test data set. The test data was generated using the same parameters for the input current, but a different random seed. The SMSE error of the subthreshold voltage prediction is plotted against the regularisation coefficient (λ in equation 5.4) in figure 5.2, on a logarithmic scale.

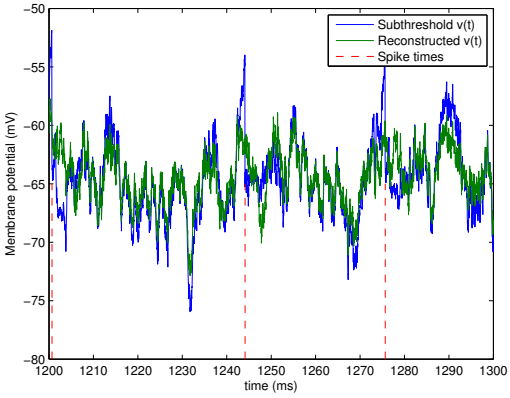
For the SIK, regularisation did not make any difference to the performance, so it was not used. Even using the 2s data, which κ is much more likely to overfit, there was no improvement when regularisation was applied. This is essentially due to the white noise input, which means that the covariance matrix C is very close to being a multiple of the identity matrix. Therefore, adding regularisation just scales the matrix by a constant, which will not improve the fit at all.

5.3.3 Subthreshold reconstruction

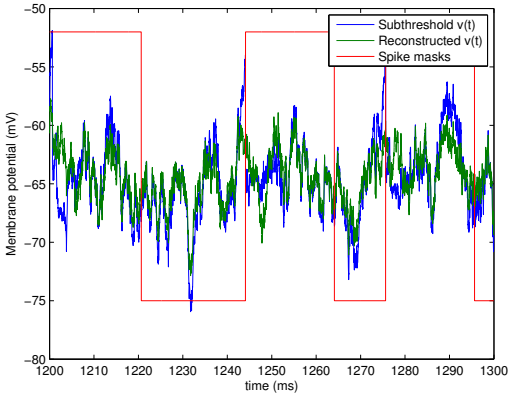
The overall SMSE for $v^{SRM}(t)$ compared to $v^{data}(t)$ was 0.422, which is perhaps not quite as good as could be hoped for. The reconstructed subthreshold potential is plotted against the HH subthreshold potential in figure 5.3 (a) over a 100ms window.

One possible explanation for the average SMSE performance is that the removal of the η kernel at action potentials has not produced the true subthreshold voltage — there are some unwanted nonlinear effects still present in the v^{data} that we have fitted the model to. This would manifest in worse performance in regions where η has been subtracted than in other regions. Figure 5.3 (b) shows the same section of data as 5.3 (a), but with regions where η has been removed shown underneath red 'spike mask' boxes. This graph indicates that the fit of the SRM may be better in the regions where the HH potential was not adjusted, so it was decided to check this quantitatively. The SMSE for data where η was not extracted (which was 35% of the total data) was 0.306,

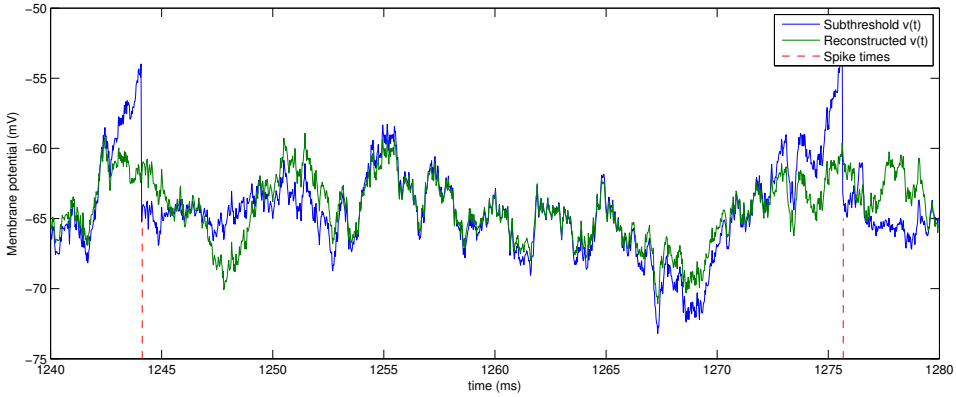
Figure 5.3: The reconstructed v^{SRM} compared to v^{data}. Graphs were created using the 100s data vector, a κ kernel of duration 12ms, and no regularisation.



(a) Subthreshold voltage against reconstruction, with spike times shown by red dashed lines.



(b) Subthreshold voltage against reconstruction. The areas below the red boxes represent regions where the η kernel has been subtracted.



(c) Zoom of a 40ms region.

providing a strong indication that the action potential template was largely to blame for the poor performance.

However, on closer examination of graphs of the reconstructed potential, it can be seen that areas of poor fit tend to be on *either* side of a spike, rather than only after a spike, which would be expected if the η kernel were to blame. Figure 5.3 (c) shows a smaller region of data, and in the middle section, away from the two spikes, the reconstructed trace follows the HH data very closely. Therefore the SMSE outside of a 'window' from 2ms before each spike to 4ms after the spike was calculated. 68% of the data was found to fall outside this window, and the SMSE over that data was 0.258. This is significantly better than the window outside η, suggesting that the SRM is just unable to capture the exact behaviour of the HH model near spikes.

5.3.4 Refractory period

When creating the spiking reconstruction, an absolute refractory period was applied to prevent multiples spikes if $v^{SRM}(t)$ was super-threshold for several consecutive time-steps. Initially, we used the same absolute refractory period as Jolivet et al. (2004), which was 2ms. Figure 5.4 (a) shows the Γ factor against the threshold θ, plotted with the N_{SRM}, N_{HH} and $\langle N_{coinc} \rangle$ values. The figure shows that a remarkably large number of SRM spikes are needed to obtain the optimum Γ – 5406 test spikes compared to 3346 spikes in the original data. This indicates that the SRM is achieving good performance just by predicting spikes at many locations; figure 5.4 (b) shows a region of the $u^{SRM}(t)$ reconstruction where the SRM predicts more spikes than the HH model.

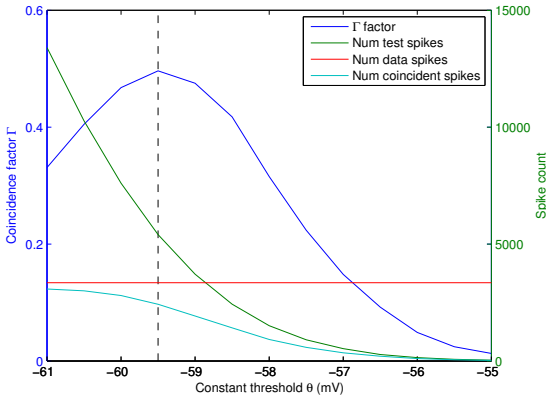
However, 5.4 (b) also suggests that part of the problem is spurious spikes being predicted immediately after a real spike, and the SRM may perform better with a longer refractory period. For Jolivet et al., this was less critical as they used a variable firing threshold, so their model would continue to be less likely to spike for a period following the absolute refractory period.

One approach for choosing a better refractory period is to find the minimum inter-spike interval in the data produced by the HH model. Figure 5.4 (c) shows the ISI for the first 500 spikes from the 100s data vector, and it can be seen that the minimum time between spikes is just below 20ms. In fact, over all the data, the smallest ISI was 13.45ms, so one option was to choose (for example) 13ms as the absolute refractory period. This was not done, as we felt it would be incorporating too specific a characteristic of the particular data into the model.

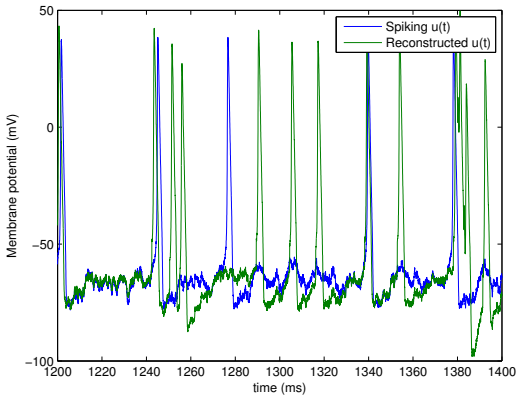
Alternatively, it is possible to estimate the refractory period of the HH model directly in NEURON. To do this, two current pulses are applied to the model, with the first being just large enough to produce an action potential. Then the delay to the second pulse is gradually increased until it generates a second spike; this delay is then the absolute refractory period. The second pulse can have arbitrarily large current (so long as the response is nonlinear, and therefore a genuine spike), which means there will never be a spike within the absolute refractory period, for any type of input, so we are justified in using this refractory period in our model.

The HH model was found to have an absolute refractory period of 6ms, which we adopted in our SRM. Figure 5.4 (c) shows the improvement in the coincidence factor obtained by using this extended refractory period.

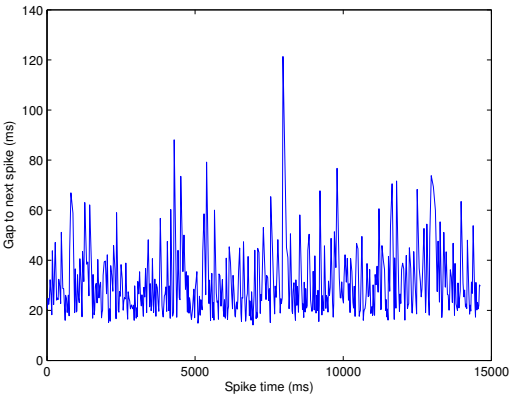
Figure 5.4: Impact of different refractory periods.



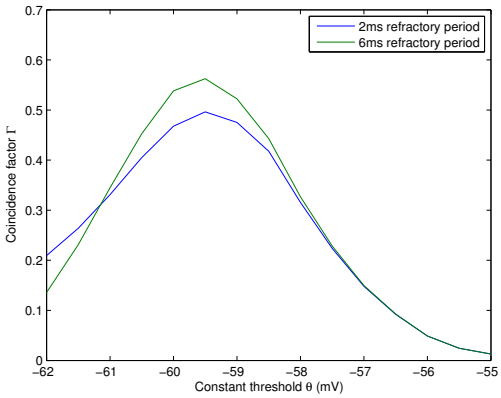
(a) Γ plotted against θ with a 2ms refractory period, also showing the number of test, data and coincident spikes. The black dotted line indicates the θ value where Γ is maximised.



(b) Reconstruction of the spiking voltage $u^{SRM}(t)$ against $u^{data}(t)$, using a threshold of -59.5mV and a refractory period of 2ms.

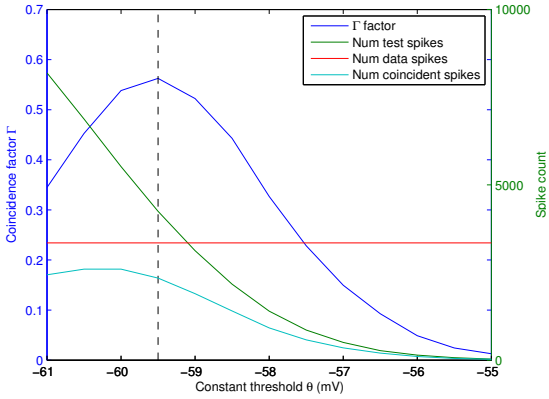


(c) Inter-spike interval for the first 500 spikes

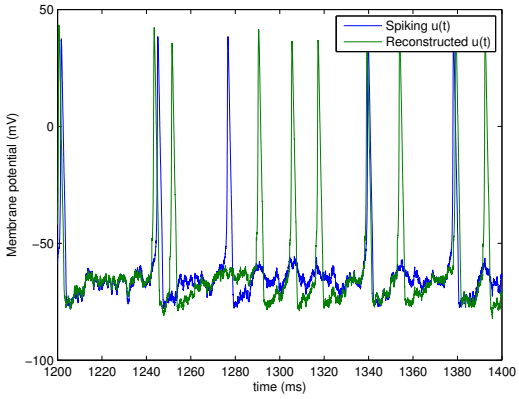


(d) Relative performance of using a 6ms and 2ms refractory period.

Figure 5.5: Setting the firing threshold θ. Results for 100s data, with a refractory period of 6ms.



(a) Γ plotted against θ, also showing the number of test, data and coincident spikes. The black dotted line indicates the θ value where Γ is maximised.



(b) Reconstruction of the spiking voltage $u^{SRM}(t)$ against $u^{data}(t)$, using a threshold of -59.5mV and a refractory period of 6ms.

5.3.5 Firing threshold θ

The value for the spike threshold θ that maximised Γ was found to be -59.5mV. This was quite robust to the duration of κ, the length of the data vector used, and the length of the refractory period. Figure 5.5 (a) shows Γ versus θ, where θ has been evaluated at 0.5mV intervals.

The value of Γ was 0.56 (with 3346 HH spikes, 4247 SRM spikes, and 2341 spikes coincident between the two spike trains). Note that while the number of SRM spikes is close to the number of HH spikes at the optimum Γ value, in some sections of the data (such as figure 5.5 (b)) it still looks as if the SRM is getting good performance by predicting a large number of spikes and having some of them fortuitously coincide with spikes in the HH data. The Γ measure is partly to blame for this; it does not include an explicit penalty for N_{SRM} being much larger than N_{HH}, and a suggestion for improving the measure is given in section 7.2.

5.4 Fitting κ by a function

In Jolivet et al. (2004), once the κ kernel has been found, a function of the form $A\exp(-t/\tau)$ is fitted to it. As well as smoothing the kernel, this allows the SRM to be related back to a traditional Integrate-and-Fire model. The standard Leaky Integrate-and-Fire (LIF) model is given by

$$\tau_m \frac{du}{dt} = -(u - u_{eq}) + RI \tag{5.5}$$

Converting the mathematics given in Jolivet et al. (2004) to the SIK case, equation 5.5 can be integrated to give

$$u(t) = u_{reset} \exp\left(-\frac{t}{\tau_m}\right) + \int_0^\infty \exp\left(-\frac{s}{\tau_m}\right) I(t-s)\, ds \tag{5.6}$$

Then, by matching terms in equation 5.6 with terms from the SRM equation:

$$u(t) = \eta(t-\hat{t}) + \int_0^\infty \kappa(s) I(t-s)\, ds$$

we can equate the η and κ kernels with exponential decays with time-constant τ_m, and therefore reduce the SRM to (nearly) a LIF model.

Jolivet et al. actually use this conversion to run the SRM by using equation 5.5, where τ_m is found by the fitting procedure, which is more computationally efficient than using convolutions to calculate the predictions of the SRM (especially for the SDK case). However, they do not mention how the η kernel fits into this methodology – η would have to be added to $u(t)$ in equation 5.5 somehow.

We follow Jolivet et al.'s method and fit κ by a function of the form $A\exp(-t/\tau)$. Based on our results with the SIK, which show κ having a significant negative portion, we know that such a function cannot be a perfect fit; however, it is still interesting to see how much impact using a function of this form will have.

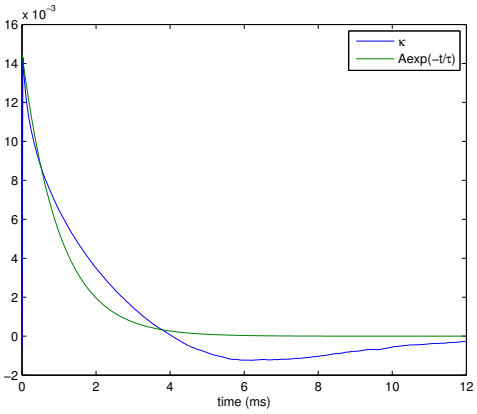
The regression procedure used to fit the parameters A and τ was a conjugate gradient descent algorithm, using a sum-squared error function. This method was selected as it was fast and reliable, and linear regression was not possible because κ had a negative region so could not be transformed using logs. The Netlab[2] procedure `scg` was used for the gradient descent algorithm.

The optimum parameters were found to be $A = 0.0145$ and $\tau = 1.000$, and the fitted function is plotted against the original kernel in figure 5.6 (a). The subthreshold reconstruction using the functional form of κ gives an SMSE value of 0.497, which is worse than the 0.422 obtained with the original kernel, but reflects a surprising amount of robustness to the detail of the kernel. A section of the subthreshold reconstruction is shown in figure 5.6 (b), plotted against the subthreshold potential as found for the HH model.

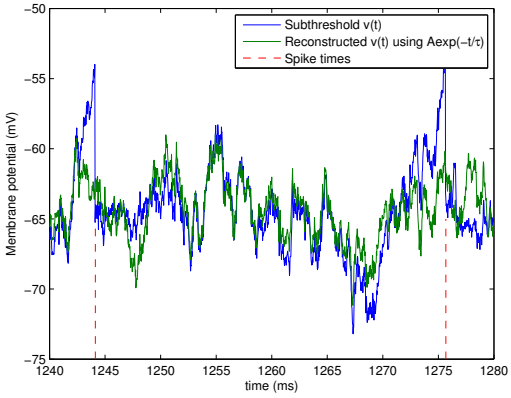
Finally, Γ for the reconstruction using $A\exp(-t/\tau)$ was 0.42, with the optimum θ value again being -59.5mV. This is a more significant drop in performance compared with using the full κ kernel, which managed $\Gamma = 0.56$, and shows that the exact kernel gives better near-threshold performance, which is the critical region when the goal is to predict spike times.

[2] `http://www.ncrg.aston.ac.uk/netlab/index.php`

Figure 5.6: Fitting κ by $A\exp(-t/\tau)$.



(a) The kernel κ plotted with the optimum functional approximation to it



(b) Subthreshold reconstruction using the fitted $A\exp(-t/\tau)$ instead of κ

Chapter 6

Fitting the κ Kernel (SDK)

In this chapter, we describe in detail the methods used to fit the spike-time dependent κ kernel to the HH model data, and methods used to produce the predictions of the SRM using the SDK. We also present results obtained using the SDK, and compare the performance of the SDK with the SIK.

6.1 Maths and implementation

As described in chapter 3, the mathematics behind fitting κ in the SDK case are very similar to the SIK mathematics. The differences are firstly that a different κ kernel must be found for each time δt elapsed since the last spike, and secondly that the summations to find the input covariance and the input/response cross-correlations are done over spike times, rather than over all data points.

The first difference means that the procedure to find κ must be repeated at each δt value. To make it possible to calculate κ in a reasonable amount of time, it was assumed that κ varied slowly with δt, so the same κ could be used for several time-steps. We chose to compute a different κ every 0.5ms (so every 20 data points). Also, the altered dynamics of the model caused by a spike will only persist for a finite time, and after a certain δt value κ will no longer change, so we chose to find κ only for δt values of up to 20ms[1].

The second difference, summing over spikes instead of over all data, has two key consequences:

- The covariance matrix of the stimulus is no longer a Toeplitz matrix.
- For a given length data vector, the summations are done over far fewer points. In fact, given that the data used had a firing frequency of 33Hz and a sampling frequency of 40kHz, there will be approximately three orders of magnitude fewer data points for the SDK.

These factors mean that the implementation of the SDK was slightly different to the SIK implementation.

[1] For these parameters, and using a Linux machine with a 3GHz Intel Xeon processor, finding κ for all δt values took approximately 15 seconds.

As in section 5.1, we find κ is the solution to the linear regression problem $C\kappa = \mathbf{x}$, which we solve using the backslash operator in MATLAB. This time, the covariance matrix will not be just a scaled identity matrix, as it will reflect the increase in current found on average prior to an action potential. This means the covariance matrix will not be as well-conditioned as the SIK covariance matrix, and so regularisation should have more of an effect. Again as in section 5.1, regularisation is applied by adding λI to the covariance matrix.

Where the SDK differs from the SIK is in the method we use to find the covariance matrix C and cross-correlation vector $\mathbf{x}$. The different method is partly necessitated by being unable to take advantage of C being a Toeplitz matrix, and partly because a more efficient approach is made possible by the reduced quantity of data. Firstly we form a $T \times s_{max}$ matrix X, which has the stimulus current values at each spike time as its columns, with the lag j that the current is measured at increasing with the column index:

$$X_{ij} \quad = \quad I_{\hat{t}_i+\delta t-(j-1)}$$

We can then find the covariance matrix efficiently by computing $X^T X$:

$$\left(X^T X\right)_{sj} = \sum_{i=1}^{T} I_{\hat{t}_i+\delta t-(s-1)}\, I_{\hat{t}_i+\delta t-(j-1)} = C_{sj}$$

Note that due to computational constraints, this method for finding C could not be used for the SIK with 100s of data — the matrix would be $4 * 10^6 \times 480$, which would need a machine with much more memory than the 4Gb of RAM in the machine used for these experiments.

Next we use a vector $\mathbf{V}$ containing the voltage values at spike times offset by δt, defined by

$$V_i = v^{data}_{\hat{t}_i+\delta t}$$

to create the cross-correlation vector $\mathbf{x}$ using

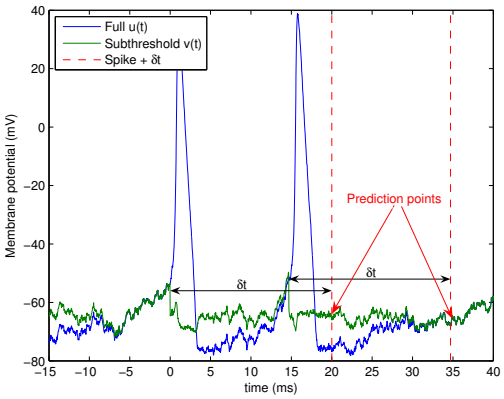
$$\left(X^T V\right)_j = \sum_{i=1}^{T} I_{\hat{t}_i+\delta t-(j-1)}\, v^{data}_{\hat{t}_i+\delta t} = x_j$$

In MATLAB, using this matrix-based formulation to find κ is at least two orders of magnitude faster than creating the covariance matrix by looping over all s and j values.

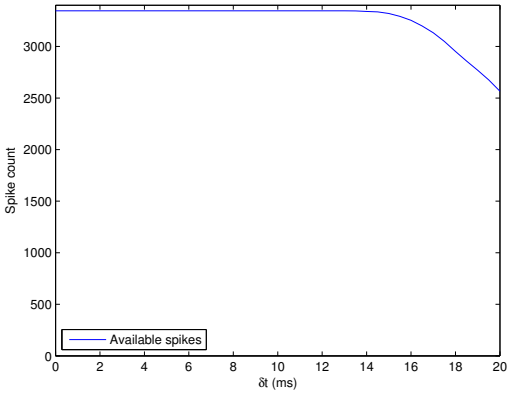
One important detail in the implementation is that, in calculating the covariance and cross-correlation matrices, not all spikes can be used at large δt values. This is because if another spike falls in the interval between $\hat{t}$ and $\hat{t} + \delta t$, the membrane potential at the point $\hat{t} + \delta t$ will be due to the intervening spike rather than the spike at $\hat{t}$. This is shown in figure 6.1 (a); for $\delta t = 20ms$, the data point at 20ms (due to the spike at 0ms) cannot be used, as it is in fact a data point for $\delta t = 5.3ms$ associated with the spike at 14.7ms. Figure 6.1 (b) shows the drop-off in available spikes with δt. There is no reduction in spikes until $\delta t = 15ms$, but from there is a sharp drop off, and at $\delta t = 26ms$ only half of the original spikes are available (not shown on the figure). However, at $\delta t = 20ms$, 77% of the spikes are still available, which was felt to be acceptable.

A similar problem occurs for small values of δt and large values of the lag j that the cross-correlation/covariance is calculated at. If the lag is larger than δt, then κ is

Figure 6.1: For larger values of δt, some spikes cannot be used for training data, as there is a following spike at less than δt afterwards.



(a) For $\delta t = 20ms$, data for the spike shown at 0ms is invalid. Data at the prediction point at 20ms must in fact be associated with the spike at 15ms.



(b) The reduction in spikes available for the fitting procedure with δt.

being found on the basis of the input current before the spike. Biologically, we expect any memory of the stimulus from before an action potential to be erased, as during the spike the membrane time-constants of the neuron will be very short.

Despite this, we note that Jolivet et al. have chosen to allow s_{max}, the upper limit on the lag used, to be larger than δt (see for example figure 4D in Jolivet et al. (2004)). Therefore we have included the handling of s_{max} as a parameter in the code, `sMaxDtOffset`. If this is set to -1, then no restrictions are placed on s_{max}; if it is set to any other value, then for each δt the calculations of the covariance and cross-correlation are made using $s_{max} = \delta t - $ `sMaxDtOffset`.

Finally, we note that although there is an instantaneous drop of around 10mV in the membrane potential at $\hat{t}$, due to the removal of the η template, this does not have any impact on the κ regression as described above. This is because the earliest value of v^{data} used in the fitting process is $v^{data}(\hat{t})$, and this data point is already at the lower potential level.

6.2 Reconstruction

The reconstruction of $v^{SRM}(t)$ and $u^{SRM}(t)$ has to be done completely differently for the SDK than for the SIK. This is because κ depends on the time since the last spike, so we cannot simply apply the filter to the complete stimulus vector without working out the spike times predicted by the model. However, the spike times depend on the predicted subthreshold voltage, so the reconstruction must proceed one time-step at once, where at each step

- The model's prediction for $v^{SRM}(t)$ is calculated, using the $\kappa_{\delta t}$ kernel corre-

sponding to the time elapsed since the last spike.

- $v^{SRM}(t)$ is compared to the threshold, and if the threshold is reached and $v^{SRM}(t)$ is increasing, then a spike is predicted.

The subthreshold potential is calculated by a direct implementation of the formula

$$v_t^{SRM} = \sum_{s=1}^{s_{max}} \kappa_{\delta t,s} I_{t-(s-1)}$$

It may be possible to improve the performance of the reconstruction code by calculating $v^{SRM}(t)$ for all time-steps where a given $\kappa_{\delta t}$ is applicable at once, making use of the MATLAB `conv` function or some other efficient approach. Then the computed $v^{SRM}(t)$ values would have to be checked against the threshold, and if a spike was triggered, values after the spike would be discarded. Each $\kappa_{\delta t}$ is used for 20 time-steps, so there is some scope for an efficiency gain here. The time range that each $\kappa_{\delta t}$ was used for was centred on its δt value, so for example, given a different $\kappa_{\delta t}$ every 0.5ms, $\kappa_{\delta t=0.5}$ would be used for periods of 0.25 to 0.75ms after a spike.

κ is only found for δt values of up to 20ms; when the time elapsed since the last spike is greater than 20ms, the kernel $\kappa_{\delta t=20}$ is used to calculate $v^{SRM}(t)$. The decision to only find κ for δt values of up to 20ms was made because it was expected that κ would not change with δt after this time, and using $\kappa_{\delta t=20}$ would be a very good approximation.

Finally, at the start of the data vectors, there are no previous spikes, and so it is not clear what $\kappa_{\delta t}$ to use. To avoid the SRM missing the first spike in the data, and having the error of using the wrong $\kappa_{\delta t}$ propagate through all the remaining reconstruction, it was decided to start the SDK reconstruction from just after the first spike produced by the HH model. An alternative approach would have been to start the reconstruction using the $\kappa_{\delta t=20}$ kernel, which would avoid giving the SRM information about the spike times in the HH model (albeit only the one spike time). Using $\kappa_{\delta t=20}$ would probably have produced very similar results, based on the ability of the SRM to recover from missed spikes in the data within a few tens of milliseconds at most.

6.3 Features of κ

6.3.1 Overview

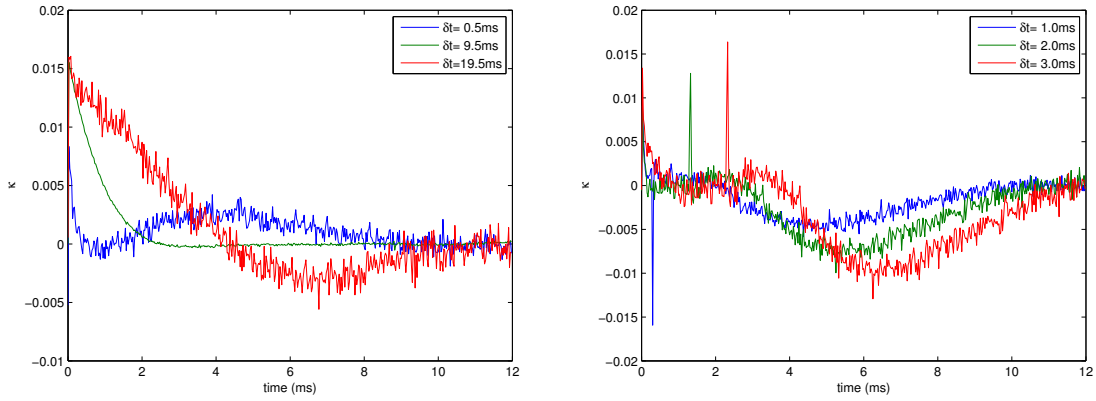
The κ kernels for δt values of 0.5, 9.5, 19.5, 1, 2 and 3 ms are shown in figure 6.2. Further graphs of the kernels at a wider range of δt values are shown in appendix A.1.

There are several interesting aspects to these κ kernels, as compared to the kernel at δt=9.5ms which is considered typical:

1. There is a positive rise in κ for δt =0.5ms, centred around a lag of 4ms.

2. There are pronounced negative dips in the kernels for δt between 1.0 and 3.5ms, at larger lags.

Figure 6.2: κ kernels found at several δt values, using $s_{max} = len(\kappa)$. Note the sharp peaks at approximately $\delta t - 0.7$ms in the kernels at δt=1, 2 and 3 ms.



(a) κ at δt=0.5, 9.5 and 19.5ms

(b) κ at δt=1, 2 and 3ms

3. There are negative curves for large time lags in the kernels for δt greater than 10ms.
4. The fitted κs are very noisy for δt values of 3ms and below.
5. The fitted κs are fairly noisy for δt values of above 15ms.
6. There is a very sharp downward peak at a lag of 0.3ms in the kernel for δt=1.0ms.
7. There are sharp upward peaks at $\delta t - 0.7$ms in κs found for δt of between 1.5 and 3.5ms.

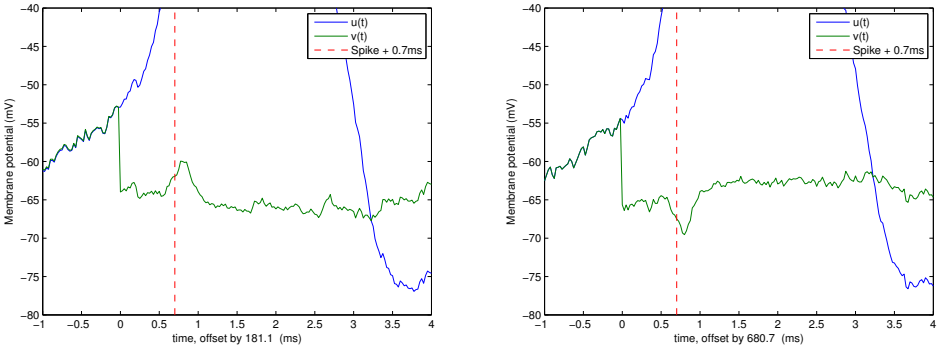
These are discussed in the sections below.

6.3.2 Subthreshold blips issue

To find an explanation for the sharp peaks in κ at $\hat{t} + 0.7$ms, which are found for δt in the range (1.0,3.5) milliseconds, we examined the subthreshold voltage data at $\hat{t} +$ 0.7ms for several spikes. For some spikes, well-defined peaks or troughs can be seen in the subthreshold voltage approximately 0.7ms after the spike. Examples of these 'blips' are shown in figure 6.3, and it was thought likely that these were responsible for the sharp peaks in κ.

The blips are caused by variations in the shape of spikes. If a spike has a sharper rise to its peak than the η spike average does, then when η is removed there will be a residual up-blip in v^{data} where the true spike had larger u^{data} values than η did (and similarly a down-blip will be produced if a spike has a shallower rise than average). We decided to manually classify the first few spikes in the data into 5 different types, based on the shape of the blip in the subthreshold voltage at $\hat{t} + 0.7$ms. The categories were 'strong up-blip', 'marginal up-blip', 'level', 'marginal down-blip', and 'strong

Figure 6.3: Blips in the subthreshold voltage at approximately 0.7ms after the spike.



(a) A spike showing a strong up-blip.

(b) A spike showing a strong down-blip.

down-blip', where the 'level' category was used for v^{data} sections where there was no blip, and the two 'marginal' categories were used where there was a slight blip, but it was not very well defined. Examples of the different categories of spike are shown in the upper plot of each pair in figure 6.4. Figure 6.6 (a) shows that each of the 'level', 'up-blip' and 'down-blip' types occurred approximately equally frequently.

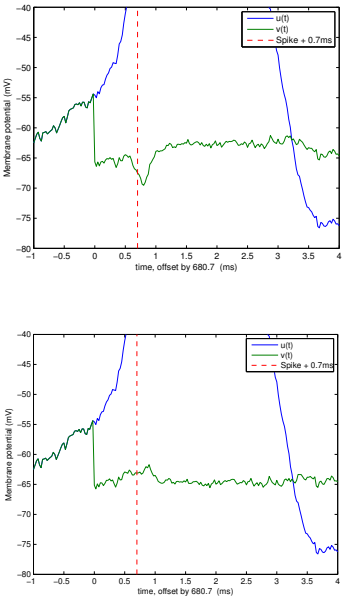
To confirm that the spikes associated with the different types of blip really did have different shapes, we calculated the average spike shape for spikes of each different type. Figure 6.5 shows the resulting blip-category specific templates; figure 6.5 (b) shows only the templates for the 'strong' and 'level' category blips, which have a clearly distinct shape. Although a proportion of the blip effect is probably due to the different spike 'types' being offset from each other on the time axis, the spikes causing strong down-blips clearly have a higher peak than the other spikes, providing some evidence that there are in fact different categories of spikes.

After finding the templates for each type of spike, we were able to subtract the type-specific template from each spike, instead of using η, to get a subthreshold voltage hopefully free of blip artifacts. The results of doing this for example spikes from each category are shown in the lower plot of each pair in figure 6.4.

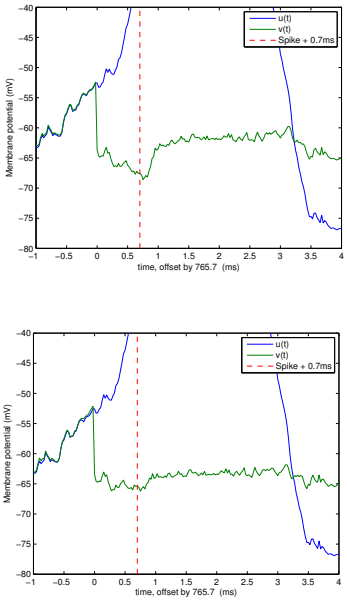
Overall, subtracting the average spikes produced reasonable results for v^{data}, with 79 of the first 100 spikes showing not even a 'marginal' blip in subthreshold voltage after subtraction of the category-specific template. The fairly high number of residual blips suggests that the choice of 5 different types of spike shape was not correct – certainly there is a lot of variation within the subthreshold blips classified to each type. Having said that, the artifacts in the subthreshold voltage would be considerably reduced if even just three different types of spikes were used.

We were also interested in the cause of the different spike shapes, and if there was any pattern to them. One possibility is that the shape of spikes depends on δt, the time elapsed since the previous spike, and figure 6.6 (b) shows the count of the different spike types against δt. The figure shows mostly down-blips at small δt values, and mostly up-blips at large δt values, but this is far from being a rule for where spikes of each type occur. However, there will be some inconsistencies in the classification of

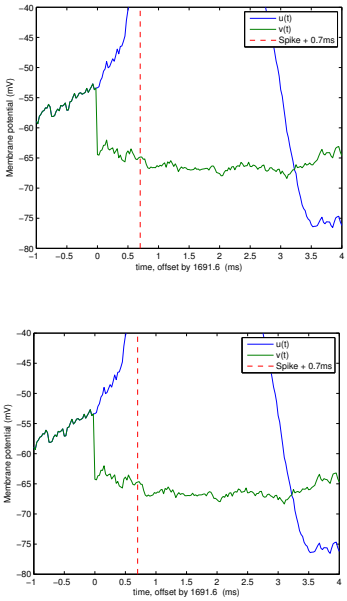
Figure 6.4: Examples of spikes with blips from each of the five categories. Below each example is the same spike showing the subthreshold voltage after the spike average corresponding to that category has been subtracted.



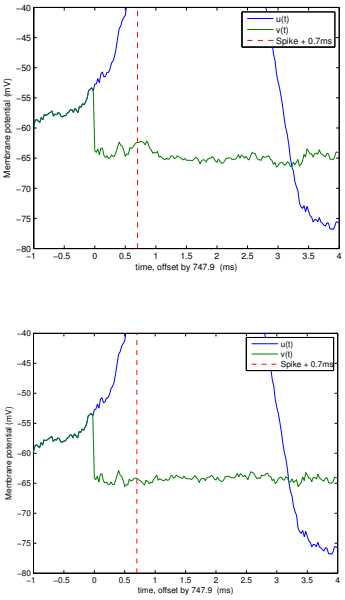
(a) Example of a strong down-blip. Note the removal of the category-specific average has left a slight up-blip in v^{data}.



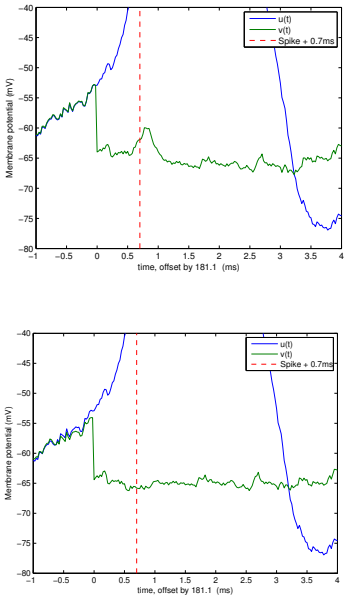
(b) Example of a marginal down-blip.



(c) Example of a level 'blip'.

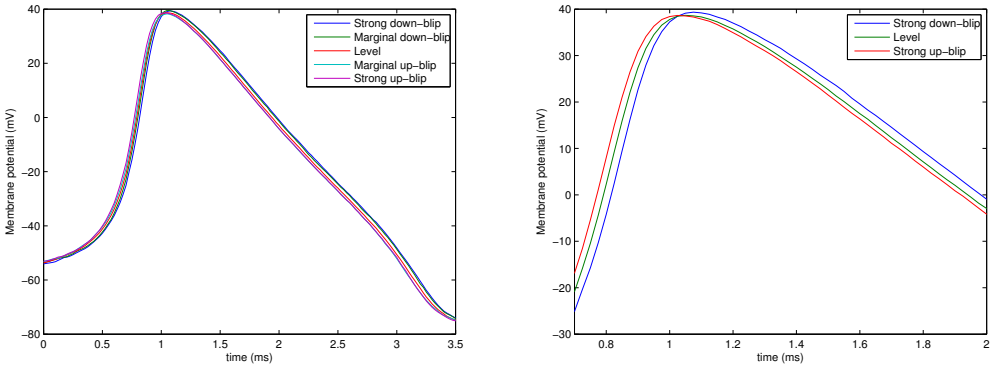


(d) Example of a marginal up-blip.



(e) Example of a strong up-blip.

Figure 6.5: Spike templates found separately for each type of blip, using a total of 150 manually classified spikes.



(a) The average shapes, showing all 5 categories of spike.

(b) The strong up/down and level blip types, shown from 0.7 to 2.0ms after the spike start.

spikes as this was done by hand, and there is a strong possibility that the shape of spikes does vary with δt. It could be that there is a continuum of spike shapes defined by a (noisy) function of δt, and separating spikes into classes is only a rough approximation to this.

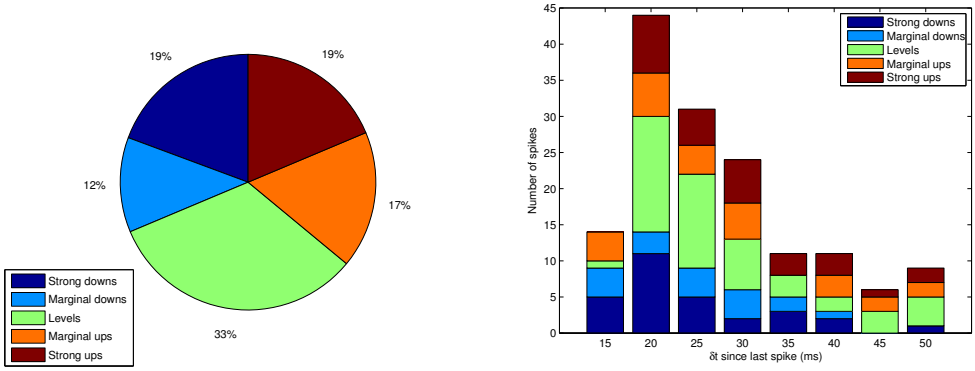
Finally, to confirm experimentally that the blips were the cause of the $\hat{t}+0.7$ms peaks in κ, the κ fitting procedure was applied to the subthreshold data produced by subtracting the spike-type specific templates. To do this, the length of κ was reduced to 2.5ms, so that only 100 parameters needed be fitted, and the fitting procedure was applied first using 70 manually-classified spikes, which was increased to 150 when 70 proved insufficient. Unfortunately, there was still not enough data to produce clean κ kernels.

Instead, we used an approximate method of removing the blips from all of the data to test the theory that they were the cause of the $\hat{t}+0.7$ms peaks. This method was to find the approximate location of the blips in relation to spikes, and join the two points on either side of the probable blip location with a straight line. The blips were found to mostly lie in the region from $\hat{t}+0.45$ to $\hat{t}+1.1$ms, and figure 6.7 (a) shows a spike with the blip erased in this way. The kernels found for δt=1, 2 and 3ms using the data with the blips blanked out like this are shown in figure 6.7 (b) and (c), and provide strong evidence that the blips were *not* the cause of the peaks in κ for δt values between 1.5 and 3.5ms. However they did cause the downward peak in κ at δt=1.0ms, and this downward peak is in fact replaced by an upward peak, bringing this $\kappa_{\delta t}$ into line with its near neighbours.

6.3.3 Features at small δt

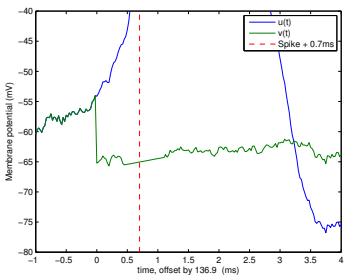
To get an idea of why some κ kernels have a pronounced negative dip at larger lags, it is interesting to look at the Spike Triggered Average (STA) stimulus. This is the input current leading up to a spike, averaged over all spikes. Figure 6.8 (a) shows the

Figure 6.6: (a) Number of each type of subthreshold blip found in the first 150 spikes of the 100s data vector (which corresponds to the first 4254ms of the data). (b) Histogram of the number of each type of blip against the time δt since the previous spike.

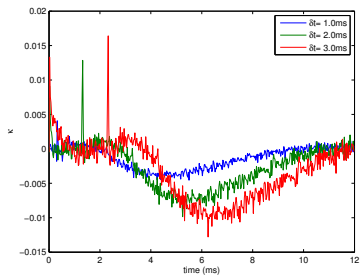


(a) (b)

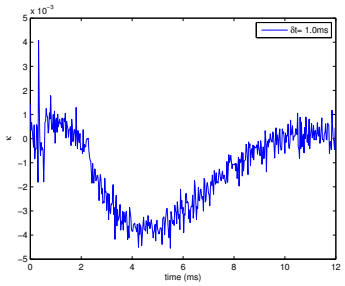
Figure 6.7: Results of an approximate method for removing the blips



(a) A spike with any potential blip replaced by a straight line.

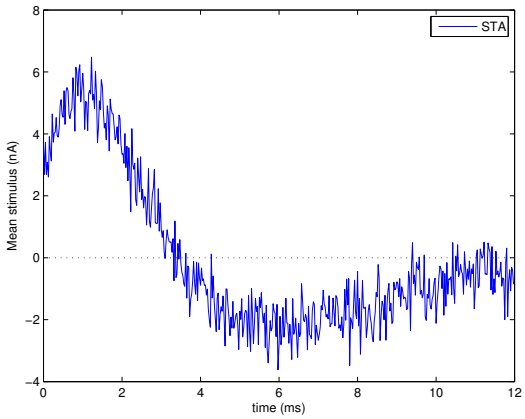


(b) The κ kernels showing the down peak at δt=1.0ms missing, but the upward peaks at δt=2 and 3 ms remaining.

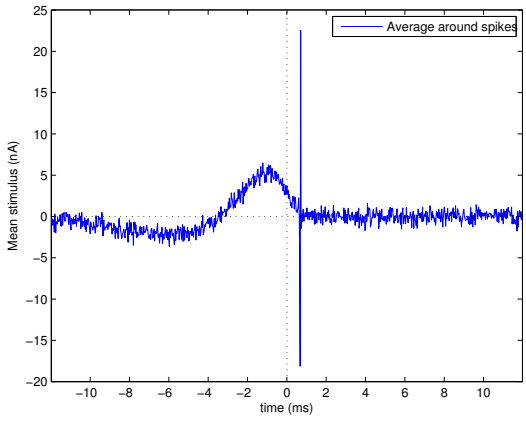


(c) κ for δt=1.0ms. The down peak has actually been replaced by an up peak.

Figure 6.8: The Spike Triggered Average of the stimulus $I(t)$ before a spike, and also extended to the period after a spike.



(a) Spike Triggered Average current, with increasing values on the x-axis reflecting times further in advance of the spike.



(b) STA calculated for both before and after spikes. The x-axis scale is forward, so negative values are times before the spike.

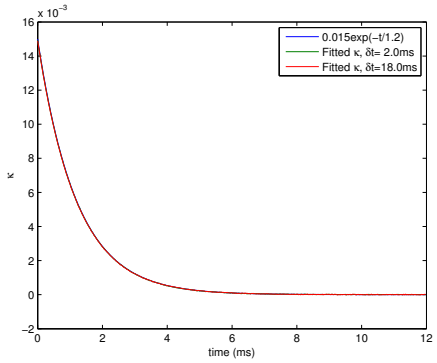
STA, plotted backwards to agree with the convention used for κ plots, and figure 6.8 (b) shows the average stimulus on either side of a spike, with the time-axis forwards.

In figure 6.8 (b), there is a large oscillation at 0.7ms, which is the start of the main body of the spike; as described in section 4.2, the start of the spike (t=0 in the figure) is offset by 0.7ms from the point at which $u(t)$ first exceeds the gradient threshold. It is not clear exactly why this large oscillation in the current is present in the STA – the fitting process does not affect $I(t)$ at all, so it is hard to see how it can be an artifact we have introduced. It is possible that this current surge is just an expected feature of the STA of the HH model, which would make some sense biologically, as the conductance of the membrane increases dramatically during a spike. However, the duration of the oscillation is at most 0.1ms, which is short compared to spike durations which are approximately 1ms at $2/3$ of their peak. Also there are causal problems with this explanation, because the spike is already in progress $\hat{t}+0.7$ms, so it seems unlikely that the pre-determined stimulus vector will happen to have major statistical correlation at this point.

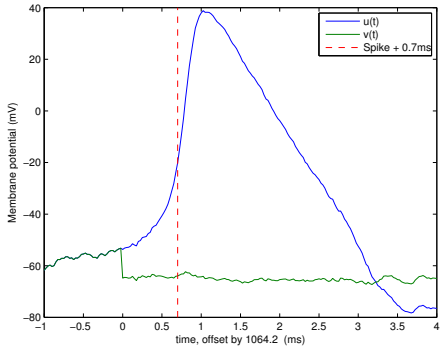
The oscillation was found to still be present when the STA was calculated over small sub-samples of 100 of the 3346 spikes in the data, and so is not caused by rouge data points or localised effects. No such effects are mentioned by Aguera y Arcas et al. (2003), who examine the STA for the HH model, but their graphs end at the spike and so may not show an effect exactly coincident with the spike. Jolivet et al. (2004) do not mention anything in their text, but it is possible to see an oscillation in the κ they find for δt=6ms, corresponding to the spike time (figure 4D in the paper). This phenomenon is worthy of further examination, but this was outside the scope of this project and it will not be discussed further.

In any case, items 1, 2 and 7 from section 6.3.1 are believed to be reflections of the

Figure 6.9: A sample spike.



(a) The kernel used to generate a sequence of artifical v^{data}, and the recovered κ kernels.



(b) A sample spike, where the start of the spike is labelled as t=0.

properties of the HH model as displayed by the STA. The sharp peaks in κ at $\hat{t}+0.7$ms coincide exactly with the location of the sharp oscillation in the STA, and the shape of the κ kernels for δt less than 3.5ms is broadly similar to figure 6.8 (a).

To check that these features found in κ at small δt really did represent properties of the HH model, and not a bias introduced by the fitting procedure due to the highly ordered average input stimulus represented by the STA, the following experiment was performed: we constructed an artificial kernel $\kappa = 0.015\exp(-t/1.2)$ (which is roughly equivalent to the SDK κ at δt=16ms), and used this kernel to generate an artificial v^{data} vector, based on the same stimulus vector used by the HH model. Then the SDK fitting procedure was used to recover the kernel, which means the fit was done using the average input sequence shown in figure 6.8, even though the spike times were not actually related to the v^{data} being used. Figure 6.9 (a) shows the original kernel and the fitted κ kernels from this experiment, which are indistinguishable. This provided evidence that the average stimulus represented by the STA did not introduce a bias into our fitting procedure.

Our current hypothesis for why the similarities to the STA only manifest at small δt values, and also why the κ kernels are very noisy at small δt, is that in the period soon after a spike the response of the HH model is not very strongly correlated with the input. This is probably because nonlinear spike mechanisms, which are mostly independent of the input, still dominate at this point, and also possibly due to the imperfect removal of the η kernel. Figure 6.9 (b) shows the shape of a sample spike, and u^{data} levels out after the spike at 3.5ms, which is exactly the δt value at which the κ kernels switch to the typical form. For larger δt, the normal subthreshold response of the model takes over in shaping κ, and also the ordered region of the STA comprises less of the vector over which the linear regression is performed to find κ.

Table 6.1: Performance measures for the SDK using an unrestricted s_{max}, $s_{max} = \delta t$ and $s_{max} = \delta t$-1. All results were obtained with θ=-57mV, λ=0, a κ length of 12ms and a refractory period of 6ms.

	Full s_{max}	$s_{max} = \delta t$	$s_{max} = \delta t - 1$
Γ	0.652	0.652	0.652
SMSE	0.456	0.444	0.443
SMSE inside $\hat{t} - 2 \rightarrow \hat{t} + 4$	0.944	0.908	0.905

6.3.4 s_{max} versus δt

As mentioned in section 6.1, it is unclear if it makes sense to consider data from before the spike when calculating the κ kernels. A possible advantage of limiting the fitting procedure to using data from after the spike is that it would remove the large negative sweeps at high lags in κ for small δt, as discussed in section 6.3.3. If the data was limited further to 1ms after the spike, that would also remove the peaks found at $\hat{t} +$ 0.7ms for small δt κs.

We investigated the impact of different data limits on the performance of the SRM. Firstly, we limited s_{max} (the maximum lag backwards from the prediction point, measured in data points) to the time of the spike: $s_{max} = \min(\text{len}(\kappa), \delta t)$. κ kernels found using this limitation are shown in figure 6.10 (a) and (b), and a more complete set of kernels is provided in appendix A.2.

Secondly, we set $s_{max} = \min(\text{len}(\kappa), \delta t - 1.0)$. Kernels for this setup are shown in figure 6.10 (c) and (d), and a more complete range of δt values is shown in appendix A.3. Note that here the kernels for δt=0, 0.5 and 1.0ms were limited in size to one data point, and so became instantaneous predictors of the voltage based on the input (and the same is true for κ at δt=0ms for the $s_{max} = \delta t$ case).

The performance of the full SRM in terms of Γ and the SMSE for each of the three s_{max} settings is given in table 6.1. There was no difference at all in the Γ values achieved, but the SMSE was very slightly better for the $s_{max} = \delta t - 1$ setting. In particular, the SMSE over the region around spikes, which is where the small δt kernels are used, was better when using $s_{max} = \delta t - 1$, so we adopted this setting for all future experiments using the SDK.

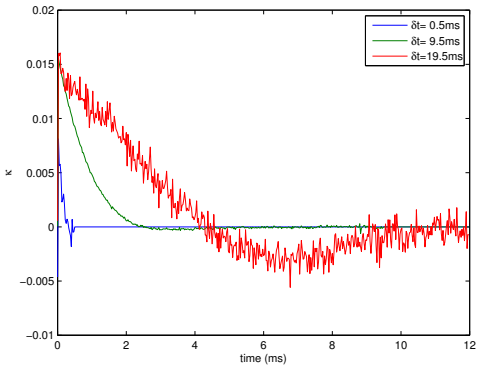
6.3.5 Features at large δt

Finally, we discuss features 3 and 5 from section 6.3.1, the negative dips at δt above 10ms, and the noisy form of κ for δt above 15ms.

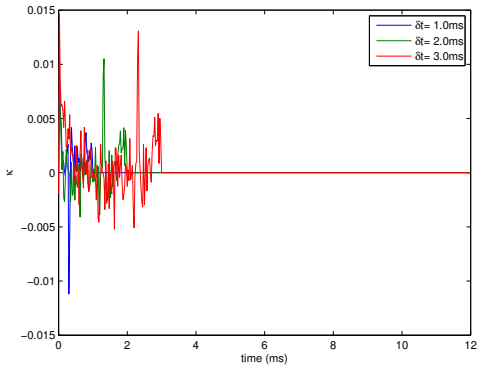
The negative sections of κ are in fact expected at high δt, as further away from spikes, the SDK kernels will tend toward the SIK kernel. As discussed in section 5.3.1 for the SIK, the negative region is simply due to the resonant properties of the model, caused by the potassium channel opening after a rise in membrane potential.

The noise in κ for δt greater than approximately 15ms was at first thought to be due to either limited data, because spikes occurring less than δt apart have to be excluded from the fitting data, or possibly due to effects of the averaged input stimulus as shown in figure 6.8 (b). The mean inter-spike interval (ISI) for the data is only 23ms, and the

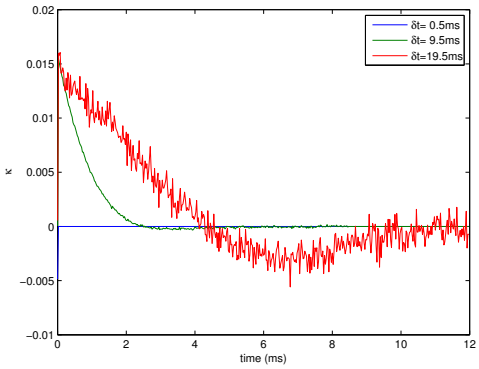
Figure 6.10: κ kernels found using only data from after the spike, and using only data from 1ms after the spike. These results were obtained with θ=-57mV, λ=0 and a refractory period of 6ms.



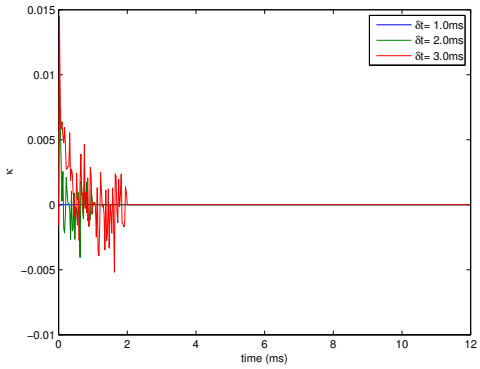
(a) κ found with no contributions from before the spike, for δt=0.5, 9.5 and 19.5ms.



(b) κ found with no contributions from before the spike, for δt=1, 2 and 3ms.

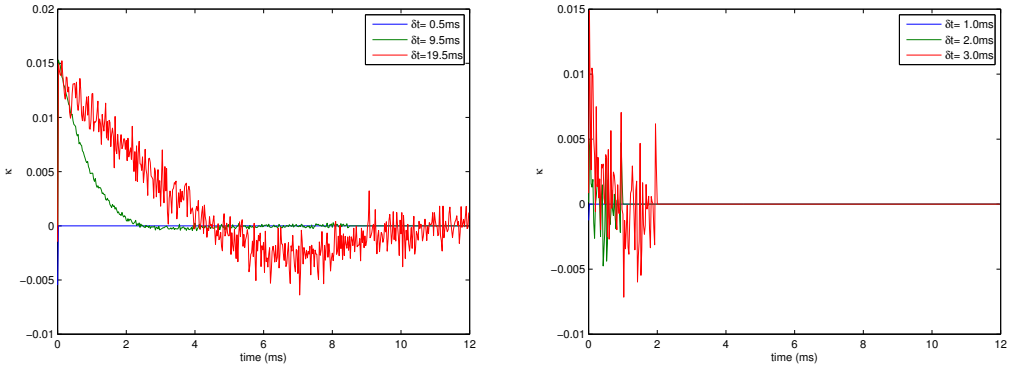


(c) κ found with no contributions from before 1ms after the spike, for δt=0.5, 9.5 and 19.5ms.



(d) κ found with no contributions from before 1ms after the spike, for δt=1, 2 and 3ms.

Figure 6.11: Kernels found using only data for spikes with a gap of at least 30ms to the following spike.



(a) κ for δt=0.5, 9.5 and 19.5ms.

(b) κ for δt=1, 2 and 3ms.

STA shows a build up in mean stimulus prior to a spike that begins some 10ms before the spike.

To test these hypotheses, we ran the fitting procedure using only spikes which had a gap of at least 30ms to the following spike. This should exclude any STA patterns in the input stimulus, even for δt=20ms. Additionally, using the same number of spikes at all δt values will allow us to tell whether or not a reduced amount of data was the cause of the noise in high-δt κs. The kernels found from this experiment, which used 1208 of the 3346 spikes in the 100s data vector, are shown in figure 6.11. As the figure clearly shows that κ for δt=19.5ms is still much noisier than κ for δt=9.5ms, the cause cannot be differences in the amount of data used, and there is less evidence for it being effects of the averaged stimulus prior to following spikes.

We therefore propose that κ is in fact exceptionally well defined for intermediate δt values, between 4 and 14ms, because in this region the response of the HH model to inputs is dominated by refractory effects due to the spike at δt=0. Then the noisiness at high δt values just reflects the natural level of variability in the response of the model; however, we have no direct evidence to support this theory.

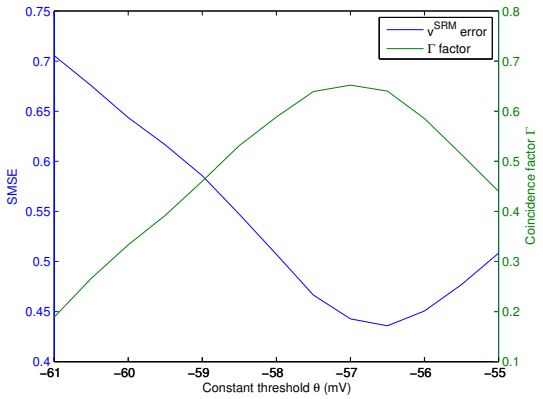
6.4 Results

6.4.1 Basic parameters

As in section 5.3.4, the refractory period used by the SRM was considered to be a property of the HH model, and so was not optimised with respect to the performance of the SRM on the data. The refractory period used was therefore again 6ms.

For the SDK, the results were found to be very dependent on the firing threshold θ. This is because κ is a function of δt, so the future predictions of subthreshold voltage depend critically on the last firing time, which itself depends on θ. Figure 6.12 (a) shows the variation of the SMSE and of Γ with θ, for a typical set of parameters used

Figure 6.12: Graph showing the dependency of Γ and the SMSE on θ (for fixed κ length, and no regularisation)



with the SDK. The optimum Γ is found using a threshold of -57mV, which was the default value adopted for θ in SDK experiments.

Figure 6.13 (b) shows that as well as Γ being very dependent on θ, the optimum threshold θ varied considerably with the duration of κ. In fact, the optimum value of θ was found to change significantly when any of the model's parameters were varied. Therefore, when experimenting with parameters using the SDK, the procedure generally used was to optimise θ separately for each value of the parameter under test. This optimisation was done using MATLAB's `fminbnd` function, which uses the golden section search algorithm to find local minima, and taking $(1-\Gamma)$ as the function to be minimised.

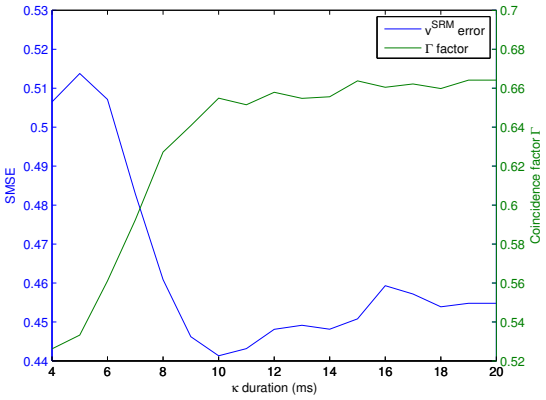
For the duration of κ, as in the SIK case this was determined by finding the length that maximised Γ. Figure 6.13 (a) shows that the peak Γ value is achieved with a κ length of 19ms, but Γ increases only very slightly from 10ms up to 19ms. However, the optimum SMSE is found with a κ duration of 10ms, with a deterioration in performance as measured by the SMSE with durations longer than 10ms. Therefore a duration of 12ms was used, as this was a good compromise between getting the best SMSE and Γ values, and also because it was the same duration used for the SIK, which would make comparisons between the two kernel types easier.

6.4.2 Regularisation

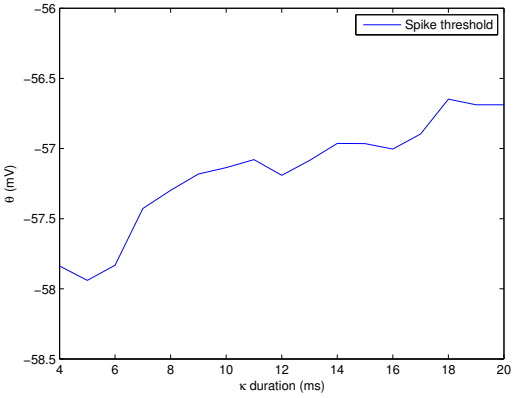
The effects of regularisation were again evaluated by finding κ using a set of training data, and then running the SRM on a separate set of test data to get the subthreshold $v^{SRM}(t)$. The test data was generated using the same parameters for the current and for the HH model, but a different sequence of randomly generated current values. A logarithmic range of values for the regularisation coefficient λ was tested, with initial results shown in figure 6.14, where a constant firing threshold was used.

Figure 6.14 (a) shows that regularisation has no effect until $\lambda = 10^3$, when the SMSE starts to improve, but unexpectedly Γ starts to get worse. Extending the range of λ beyond that shown in 6.14 (a) showed the SMSE rise to 1 and Γ drop to 0 for very large λ.

Figure 6.13: SMSE, Γ and θ shown against the length of κ, where θ was optimised to a tolerance of 0.1mV at every point to maximise Γ.



(a) SMSE and Γ as a function of the length of κ.



(b) The threshold θ that maximises Γ, plotted as a function of the duration of κ.

Figure 6.14: The effect of increasing regularisation on the performance of the SRM on test data, using a constant threshold of -58mV.

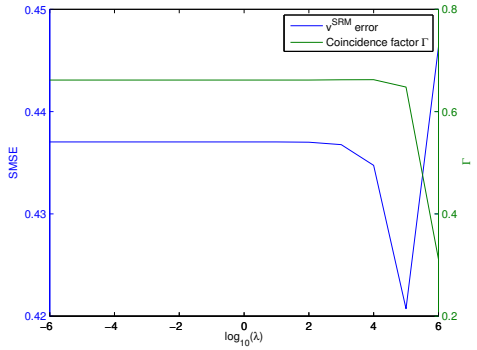
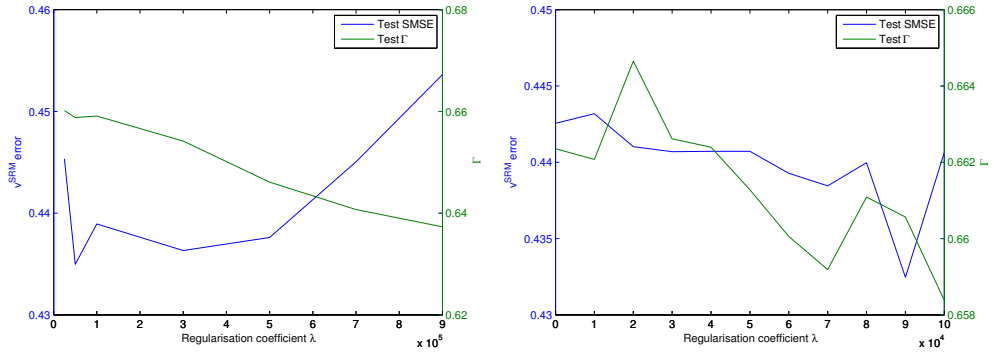


Figure 6.15: SMSE and Γ for different levels of regularisation, where θ is optimised for each value of λ.



(a) Regularisation over a range from $25*10^3$ to $900*10^3$, optimising θ to 0.25mV at each value.

(b) Performance of the SRM with the regularisation factor in the range $0 - 10^5$, where θ was optimised with a tolerance of 0.1mV at each λ value.

Results with $\lambda = 10^{5.5}$, the point where the optimum SMSE value was found, were investigated further using just the training data, to try to ascertain why Γ was poor despite a drop in the SMSE. For this level of regularisation, the number of spikes produced by the SRM was only 2665, as compared to 3346 in the HH model data – suggesting that the threshold may be too high. This proved to be the case.

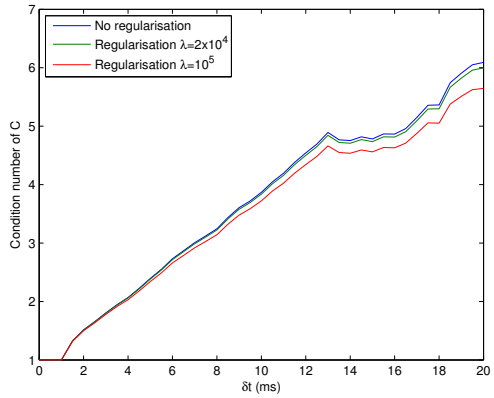
Next, a range of regularisation values was tested where the threshold was optimised separately to maximise Γ at each λ. Figure 6.15 shows the results for the SMSE and Γ on the test data set, using two different ranges of λ. The best results were obtained with a value of $\lambda = 2x10^4$, and the κ kernels found using this value are shown in appendix A.4 (however, these are visually very similar to the kernels found without regularisation).

It is also interesting to examine the condition number of the covariance matrix, which is an indicator of the robustness of the matrix to noisy data. Condition numbers nearer to one indicate a more robust matrix, for which regularisation will be less useful. Figure 6.16 plots the condition number of the SDK covariance matrix as a function of δt, and it shows that the condition number increases with δt, suggesting that possibly more regularisation should have been used when finding κ for higher δt. For comparison, the condition number of the covariance matrix for the SIK was 1.0595.

6.4.3 Subthreshold and spiking reconstructions

The subthreshold reconstruction is shown over a 100ms window in figure 6.17 (a), and it can be seen that the SDK predictions are generally a good fit to $v^{data}(t)$. Figure 6.17 (b) shows a zoom of a region where the SRM has correctly predicted a spike, and for most of this plot $v^{data}(t)$ is indistinguishable from $v^{SRM}(t)$, whereas figure 6.17 (c) shows a section where the SRM has missed a spike and therefore the prediction deviates significantly from the HH data.

Figure 6.16: Condition number of the covariance matrix, plotted against δt, the time elapsed since the last spike. The effect of various levels of regularisation on the condition number is also shown.



A section of the SRM reconstruction of $u(t)$ is plotted in figure 6.17 (d), showing that the SRM predicts some spikes at exactly the correct time, some at approximately the correct time, and some spikes it misses or adds.

The Γ coincidence factor achieved by the SDK was 0.647, and the SMSE was 0.441. Outside of a region starting 2ms before spikes and ending 4ms after, the SMSE recorded was 0.229, which is a fairly good error figure. Part of the reason the SDK performs badly within the window around spikes is that almost a third of the spikes generated by the model are missed or added compared to the HH data, and the subthreshold reconstruction will be systematically wrong in these cases due to the SDK using the wrong $\kappa_{\delta t}$.

6.5 Fitting κ by a function

As with the SIK (section 5.4), we fit the SDK κ by a set of functions, following the procedure used by Jolivet et al. (2004). For κ found at each δt value, the results are fitted by the function

$$\kappa_{\delta t}(t) = A\exp(-t/\tau)$$

as in the SIK case. Additionally, the variation of the time-constants with δt is then fitted by the function

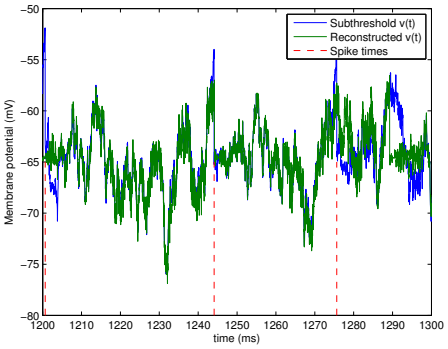
$$\tau(\delta t) = \alpha_1\left(1+\tanh\left[\alpha_2(\delta t-\alpha_3)\right]\right) \tag{6.1}$$

Jolivet et al. then use $\tau(t-\hat{t})$ as defined by equation 6.1 to run the SRM using a version of the integrate-and-fire differential equation with a time-dependent time-constant,

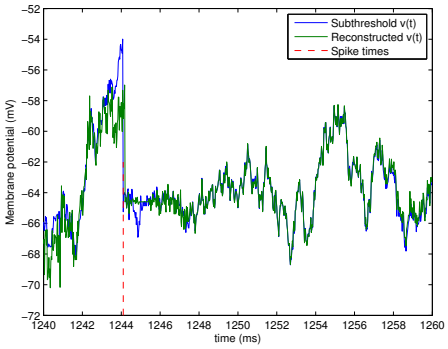
$$\frac{du}{dt} = -\frac{u}{\tau(t-\hat{t})} + \frac{1}{C_m}I(t) \tag{6.2}$$

which is more computationally efficient than using convolutions to generate the SRM predictions (especially when the filter used in the convolution changes every few time-steps, as it does in our case as κ varies with δt).

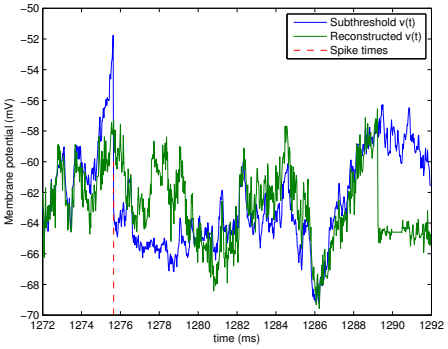
Figure 6.17: Examples of the reconstructed $v^{SRM}(t)$ and $u^{SRM}(t)$ data using the SDK. These results were produced using $\lambda = 2 \times 10^4$, $s_{max} = \min(\text{len}(\kappa), \delta t - 1.0\text{ms})$, and the 100s data vector.



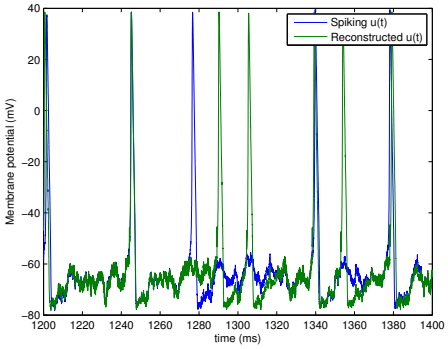
(a) The subthreshold voltage as reconstructed by the SDK over a 100ms region.



(b) Zoom of v^{SRM} around a spike, showing very good match with v^{data} for δt between 4 and 12 ms.

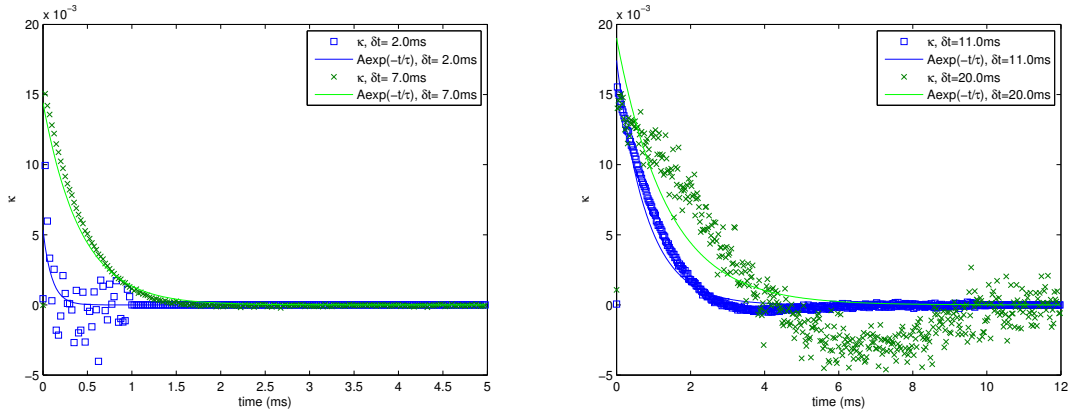


(c) Zoom of the reconstruction of v^{SRM} following a missed spike.



(d) The reconstructed u^{SRM} shown against the full voltage data from the HH model.

Figure 6.18: The fit of κ by $A\exp(-t/\tau)$.



(a) κ fitted by a function at δt of 2 and 7 ms.

(b) κ fitted by a function at δt of 11 and 20 ms.

We do not take this approach for implementing the SRM, but instead use the functional approximations found to generate a 'fitted' version of the κ kernel. This means we also need the scale factors A for the exponential functions at each δt, which are not needed if equation 6.2 is used. We fitted the variance of A with δt by the function

$$A(\delta t) = \alpha_1 \tanh\left(\alpha_2(\delta t - \alpha_3)\right) \tag{6.3}$$

which is very similar to the function used for $\tau(\delta t)$, but allows for the negative values of A found. The regression to find the parameters for all functions was again done using gradient descent on a mean-squared error function and the `scg` algorithm from Netlab.

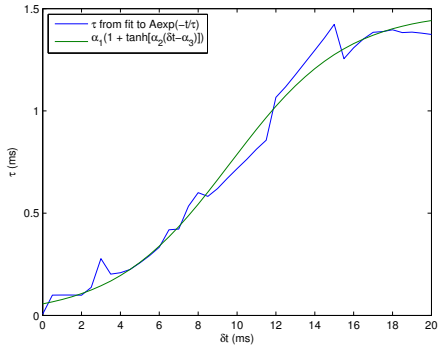
As κ is quite noisy for small δt values, and in particular as it has negative dips for large δt, the exponential decays cannot be expected to fit κ perfectly. However, as figure 6.18 shows, the fit is surprisingly good for most δt values.

τ and A are also very well fitted by the tanh-based functions, as shown in figure 6.19.

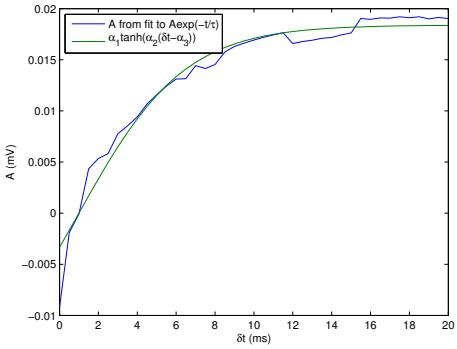
When creating the subthreshold voltage reconstruction using κ as defined by the fitted functions, it was found again that the optimum threshold had changed. The optimum θ value had fallen to -58mV, and as can also be seen from figure 6.20, $v^{SRM}(t)$ does not have enough variance to match $v^{data}(t)$ well. Even in a region where both models have missed a spike, as shown in figure 6.17 (c) and 6.20 (b), the model using the raw κ values produces a subthreshold trace much closer to the target HH data.

Overall, the function-based reconstruction achieved a Γ value of 0.474, and an SMSE of 0.559, both of which are much poorer than the 0.647 and 0.441 managed by the raw κ model. This suggests that the aspects of the kernels that could not be reproduced by the functional version, in particular the negative parts of the curves, provide a key component of the model.

Figure 6.19: The fit of $\tau(\delta t)$ and $A(\delta t)$ by tanh-based functions.

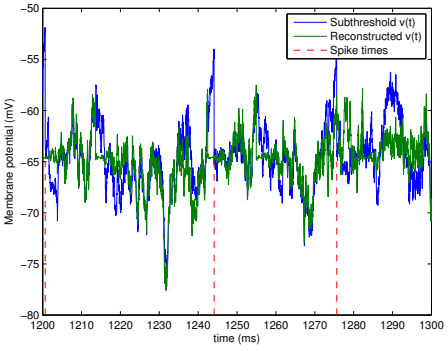


(a) The time-constants of κ fitted by the function $\alpha_1\left(1+\tanh\left(\alpha_2(\delta t-\alpha_3)\right)\right)$.

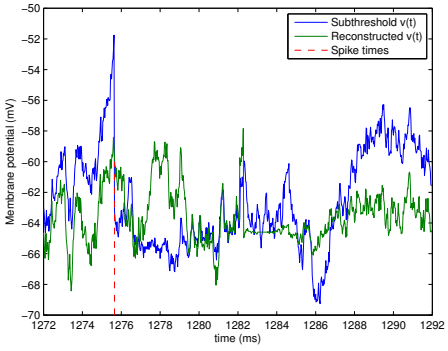


(b) The scale factor A fitted by the function $\alpha_1\tanh\left(\alpha_2(\delta t-\alpha_3)\right)$.

Figure 6.20: Reconstruction of the subthreshold voltage using κ defined by the functions $\tau(\delta t)$, $A(\delta t)$, and $A\exp(-t/\tau)$.

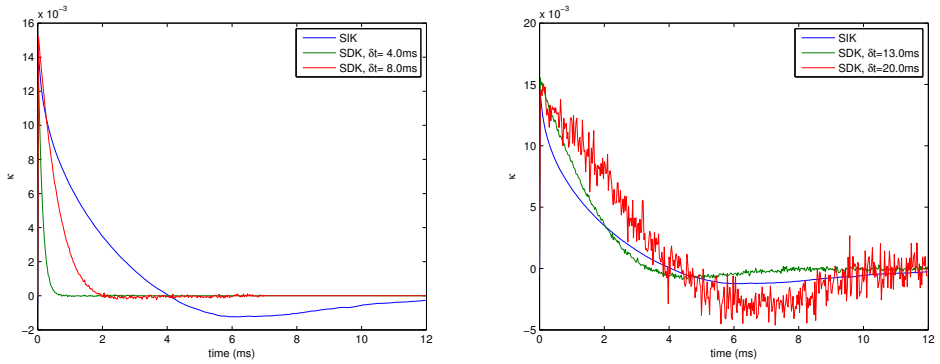


(a) 100ms of the v^{SRM} reconstruction.



(b) Zoom on a 20ms region of the subthreshold reconstruction, where the model has missed a spike at 1275.7ms.

Figure 6.21: Comparison of the κ kernels of the SIK and the SDK.



(a) The SIK κ and SDK κs at δt=4 and 8ms.

(b) The SIK κ and SDK κs at δt=13 and 20ms.

Table 6.2: Summary of performance measures for SIK and SDK

Performance measure	SIK	SDK
Γ	0.562	0.647
Γ counts ($N_{HH} = 3346$)	$N_{SRM} = 4247$, $N_{coinc} = 2341$	$N_{SRM} = 3478$, $N_{coinc} = 2367$
SMSE	0.422	0.441
SMSE outside η window	0.306	0.357
SMSE outside $\hat{t} - 2 \rightarrow \hat{t} + 4$	0.258	0.229

6.6 Comparison of SIK and SDK

A comparison of the κ kernels found using the SIK and SDK methods is shown in figure 6.21. For small δt values, the SDK κ is very different to the SIK κ, with a much smaller time-constant (figure 6.21 (a)). We expected the SDK kernel to be almost identical to the SIK κ at larger δt, as the effects of spikes become negligible, but figure 6.21 (b) shows that although the time-constants become more similar, there is a definite difference in the overall shape between the two types of kernel.

The final SMSE and Γ measures for the SIK and SDK are presented in table 6.2, both using their respective optimal parameters and the same 100s data vector. In terms of spike coincidences, the SDK is decisively superior, but the SIK actually has a better SMSE score than the SDK (although the values are very close). However, outside the region immediately preceding and following a spike, the SMSE for the SDK is better than that for the SIK. Also note that the SIK optimum threshold was -58mV, compared to -57mV for the SDK, which suggests immediately that the SDK is predicting the sharp rises in v^{data} that precede a spike more faithfully than the SIK.

We next compare the subthreshold reconstructions found using the two different types of kernel. The interesting regions are the ones that impact on the Γ score, namely regions where the SRM either misses a spike, or adds an extra spike, and several of these regions are shown in figure 6.22. Note that in order to distinguish the different data traces on these plots, they are on a larger time axis scale than most plots in this

thesis, covering just 10ms.

Firstly, figure 6.22 (a) shows a region where both kernel types have missed a spike. In this case, the kernels perform similarly, with the SDK being slightly closer to v^{data} before the spike, and the SIk slightly closer after it. Next, figure 6.22 (b) shows both kernels adding a spike: here the SIK does much better than the SDK, as the SDK's κ kernels for δt values soon after a spike are very bad predictors of the membrane voltage a long time after a spike.

Figure 6.22 (c) shows a region where the SIK misses a spike, but the SDK predicts it correctly. In this case the SDK is much better than the SIK, before the spike, immediately after it, and especially in the region from δt=3.5ms onwards, where the v^{SRM} prediction of the SDK is indistinguishable from v^{data}. Figure 6.22 (d) shows the SIK adding an extra spike where the SDK does not add a spike. This does not noticeably harm the performance of the SIK, as its dynamics do not change in the aftermath of predicting a spike, and there is nothing to choose between the two models in this plot. Finally, in the case when the SDK misses a spike but the SIK predicts it, neither model match v^{data} particularly well, and in the stretch of data we examined, no cases were found of the SDK adding a spike but not the SIK.

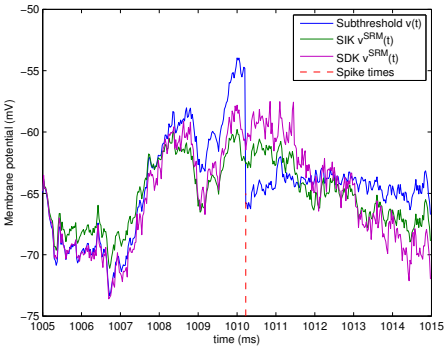
In summary, the differences in performance between the SIK and SDK arise because when the SDK misses or adds a spike, it deviates significantly from the HH subthreshold voltage; this causes its overall SMSE to be slightly worse than that for the SIK. However, the SDK is better than the SIK at predicting the subthreshold voltage away from spikes, and this makes it able to achieve a significantly better Γ value than the SIK.

Lastly, we considered the issue of the amount of data that each kernel requires to make reasonable predictions. With respect to the subthreshold voltage data, the SDK only makes use of one data point per spike (at each δt value). This means it only uses 3346 data points to fit the kernels from the 100s data vector, whereas the SIK has $4x10^6$ data points (the effects of this can easily be seen in the smoothness of the resulting κ kernels, for example in figure 6.21). Therefore we tested the SIK on the equivalent number of data points, which required 85ms of data.

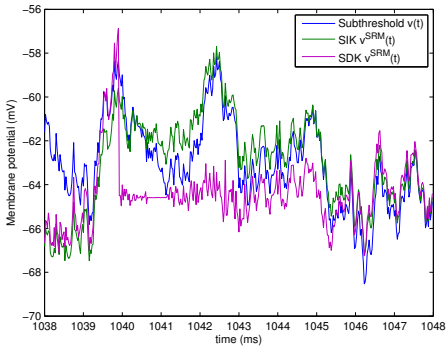
This was done by taking an 85ms section from the 100s data (actually the section starting at 915ms), which contained 3402 data points. The SIK κ kernel fitted using this section of data is shown in figure 6.23, and while it is not nearly as smooth as the kernel found from 100s of data, it has essentially the same shape. Using this kernel to predict the subthreshold and spiking voltages from the full 100s input current, the SIK achieved an SMSE value of 0.477, and a Γ spike coincidence score of 0.587. These values were actually both better than obtained using the kernel fitted from the 100s data, which is an unusual result, as the κ found from the 100s data is extremely smooth and unlikely to be suffering from overfitting.

The SDK requires much more than 85ms of data to find sensible κ kernels; at a length of 12ms and a sampling frequency of 40kHz, there are 480 data points in κ. This means that 480 parameters must be fitted, so it is not possible to produce reasonable results using data with fewer than 500 spikes. With 2s of data (71 spikes), the SDK has an SMSE of over 1, which means it performs worse than if it had predicted the mean voltage everywhere. With 15s of data (512 spikes), it manages an SMSE of 0.72 and Γ of 0.43, and 100s of data is the minimum amount needed for the SDK to make good

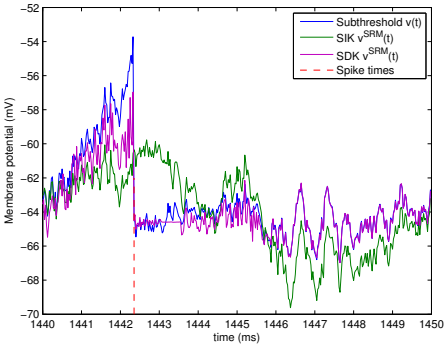
Figure 6.22: Comparison of the subthreshold reconstructions generated by the SIK and the SDK.



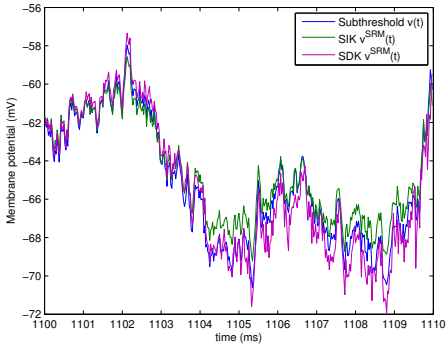
(a) Both models miss a spike.



(b) Both models add an extra spike.

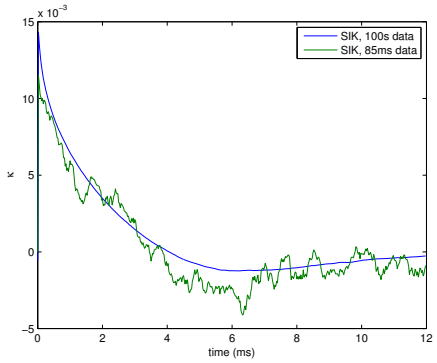


(c) The SIK misses a spike (but the SDK correctly predicts it).

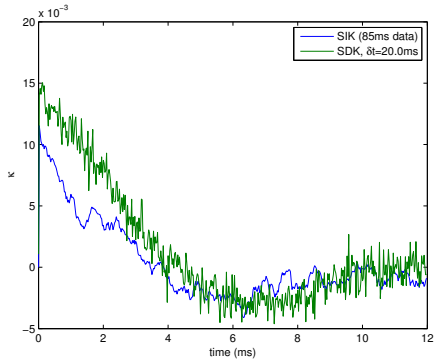


(d) The SIK adds an extra spike.

Figure 6.23: Comparison of κ found using the SIK approach from 85ms of data with the SIK kernel using 100s of data, and with the SDK kernel (found using 100s of data).



(a) SIK kernels found using different amounts of data.



(b) The SIK κ fitted from 85ms of data compared to the SDK κ fitted from 100s of data.

predictions.

Chapter 7

Conclusions and future work

We now summarise the work performed, point out interesting aspects and evaluate the results as a whole, and provide ideas for future work.

7.1 Conclusions

7.1.1 Summary

We have implemented a complete solution for fitting the parameters of an integrate-and-fire based model to neural data, and provided an analysis of the performance of resulting model. The fundamental technique used was linear-regression to find the optimal linear filter for the response of the neuron given the stimulus, based on cross-correlation analysis.

This solution involved firstly extracting a well-defined template for action potentials, and removing this from voltage data from a HH model to create subthreshold membrane potential data that was nearly independent of the action potentials. Two different forms of the Spike Response Model were then fitted to this data; Spike Independent Kernel and Spike Dependent Kernel versions.

For the SIK, we optimised the parameters of the model and were able to obtain a fairly good reproduction of the voltage data and spike times of the HH model, with 55% of the spikes predicted by the SRM matching exactly spikes in the HH data. We reduced the SRM to a normal integrate-and-fire model by fitting the kernel by an exponential decay function, and found the performance of the model dropped markedly.

Using the SDK, we found there were many more factors to consider in setting up the model to work optimally, and several interesting features were discovered. These features were thought to be due to the irregular input/output behaviour of the model during a spike, and possibly also due to imperfect removal of the action potential template. There are several avenues for investigation that may improve results using the SDK (see section 7.2). However, the ultimate performance obtained using the SDK was better than the SIK, with 68% of the spikes predicted by the model coinciding with spikes in the HH data. Finally, we fitted the SDK kernels with a set of functions that allowed us to reduce the SRM to an integrate-and-fire model, but again found that the performance suffered through this procedure.

7.1.2 Evaluation

Compared to results obtained by Jolivet et al. (2004), our SDK model has performed similarly or slightly better than their equivalent model (the constant threshold SRM), based on extrapolating the equivalent input standard deviation in their model from a frequency/current graph they present. Jolivet et al.'s results vary rapidly with stimulus standard deviation, and are presented only graphically, so any comparison can only be very approximate. They achieve a significantly better match to the conductance-based model by including a variable threshold into their SRM, which is something we would consider a high priority for future work on this project.

It is also interesting to compare results in this thesis with results from Kistler et al. (1997), who used a very similar HH model as their reference for fitting an SRM to, but based on impulse responses of the conductance model. The η kernel we found is identical to theirs, and the κ kernels are very similar, showing a flat tail at intermediate δt values, and a negative tail at large δt. For firing rates similar to ours, Kistler et al. obtain values around 0.9 for Γ using an SDK, compared to 0.65 for our SDK. This suggests that the impulse response method is a more accurate method of obtaining κ kernels than the cross-correlation approach taken in this thesis. There are two mitigating factors for our method; firstly, it is applicable to any neural voltage trace, and in particular data from a biological neuron, where there is no model that impulse response tests can be performed on. Secondly, the Γ measures were calculated differently by Kistler et al., in that they used the correct $\kappa_{\delta t}$ kernel at all points based on spike times from the HH data, even when the SRM missed or added a spike, and this is likely to have significantly improved their results.

Due to time constraints for this thesis, it was not possible to evaluate the performance of the model on as many inputs as would have been liked. In particular, the quantitative measures were generated using the same data used to fit the model, and so these measures should be regarded as indicative rather than definitive. A more rigorous approach would have been to use a set of test input vectors for generating the SMSE and Γ measures when optimising parameters, and a separate set of validation input vectors for reporting the final results. This would also have allowed confidence intervals on the SMSE and Γ measures to be determined.

On a related point, it would have been good to test the model on stimuli with a variety of standard deviations and means. This would have enabled the generalisation ability of the model to be evaluated. However, regularisation has been applied to the models developed in this thesis, which will not have improved the results presented, but should allow the model to generalise better to different inputs.

7.1.3 SIK versus SDK

Firstly, as a key goal of a simple neural model is computational tractability, it is relevant that creating voltage predictions using the SIK was 10 times faster than using the SDK. The SDK was no faster than the NEURON model used; however, no attempt was made to optimise the SDK code and there is the potential for some performance gains to be made.

The SDK was found to be very sensitive to the firing threshold used, and this had to be optimised whenever any other parameters for the model were changed. It was also

sensitive to artifacts produced by the subtraction of the action potential template, and in general around spikes the stimulus/response processing of the HH model is at its most nonlinear, so it will always be tricky to fit a linear model to data from these regions. By contrast, the SIK averages over all data, and so is quite robust to irregularities in the data around spikes. In particular, the subthreshold voltage produced by extracting the η kernel has an instantaneous drop of approximately 10mV at spike times, and the SIK was able to perform better than the SDK at reconstructing the subthreshold voltage of the HH model in spite of this. Also, the SIK is able to perform well on a much shorter duration data vector than the SDK, which would be especially relevant if the SRM were to be fitted to hard-to-obtain biological data.

Of course, a simpler model also means there is less room for improvement; we suspect that the SIK implementation is near-optimal, whereas there are still likely to be ways of tuning the SDK to improve its performance.

The SDK should only be expected to be an advantage over the SIK for predicting spikes when the firing rate is high enough to make the ISI shorter than the time it takes κ to reach a steady value. For the data used, which had a median ISI of 25.4ms, the SDK was already able to predict spikes from the HH model more accurately than the SIK, despite having a steady state κ kernel for δt values of more than 20ms since a spike.

The SRM (in both SIK and SDK guises) was found to be capable of capturing more of the behaviour of the HH model than the equivalent IF model, when the IF model's parameters were the found by reduction from the parameters used by the SRM. The SDK proved more powerful than the SIK, but significantly more work was required to get optimum performance from it.

7.2 Future work

7.2.1 Short term

A key task is to improve the procedure for finding and removing η, as the performance of the SRM is quite dependent on this. Two possible approaches that there was insufficient time to attempt will now be described. The first idea is to subtract the η template starting from a point 0.7ms before the gradient threshold is passed, as before, but to keep the spike times used by the κ fitting and v^{SRM} reproduction procedures as the time the threshold was exceeded. This will mean that the κ kernels are found and used on the basis of a time that more accurately reflects the start of the spike.

The second idea is to use a much lower threshold for finding the start of spikes, but use a further check that the point where the threshold is exceeded is actually the start of a spike. This extra check could be, for example, to ensure that the voltage rose above 0mV within 0.5ms of the threshold being exceeded. This may allow the 0.7ms offset to be removed, and η to still start at a low membrane potential.

A minor point is that it would have made most of the plots involving membrane potential easier to interpret if the voltage had been defined to be zero at the resting potential, in terms of the fitting procedure if not the biology.

For some of the parameter optimisations required, especially for the SDK, it may make sense to use a multi-dimensional optimisation algorithm such as the Nelder-

Mead Simplex (Lagarias et al., 1998) to try to ensure that the parameters used are jointly optimal. This was not done as part of this thesis as it is a slightly 'black-box' approach, and a better understanding of the SRM can be gained by examining and plotting the variance of the parameters independently.

There are several areas where interesting results were found, but there was no time to investigate these fully. One such area is the way that the SMSE can improve, but Γ deteriorate, when changing a parameter. This was observed in particular for varying the length of κ and the amount of regularisation for the SDK. Another interesting area is the very sharp oscillation in the spike-triggered average stimulus, as discussed in section 6.3.3. Also, in section 6.6, an unusual result is obtained where the SIK κ found from 85ms of data performs better than the κ found from 100s of data, and the reasons for this deserve further investigation.

To assist with investigating the performance of the SRM, it would be helpful to develop several small tools. The differences in performance between different incarnations of the SRM, for example two different settings for a parameter, normally manifest themselves by differences in the spikes that are missed and/or added between the two models. As there may only be a handful of spikes where the models differ, it would be useful to have a routine to find the differences in missed/added spikes between two SRM spike trains.

Additional tools to provide an insight into the performance of the SRM would be an evaluation of the SMSE in the region of missed/added spikes, to allow the exact impact of missing and adding spikes on the average error to be worked out. An analysis of the SMSE for a range of values of δt would be useful too.

In section 5.3.5, it is pointed out that the Γ measure is often maximised by having the number of spikes predicted by the SRM much larger than the number of spikes in the HH data, in order to get a small increase in the number of coincident spikes. Figure 5.4 (a) shows this especially clearly; although Γ is maximised at θ=-59.5mV, for θ=-59mV the number of coincident spikes is closer to N_{SRM}, and N_{SRM} is also much closer to N_{HH}. Therefore, a better reflection of the desired performance of the SRM may be achieved by adjusting Γ to penalise scenarios where N_{SRM} is much greater than N_{HH}. A basic modification to try would be to include a factor of $\exp\left(-\frac{N_{diff}}{N_{mean}}\right)$ in Γ, where $N_{mean} = \frac{1}{2}(N_{HH} + N_{SRM})$ and $N_{diff} = |N_{HH} - N_{SRM}|$.

Finally, a task involving more significant changes to the framework, but still considered a high priority, is to implement a variable firing threshold, as used either in Jolivet et al. (2004) or in Jolivet et al. (2006).

7.2.2 Medium term

A lot of the time available on this project was spent developing the core framework for fitting the SRM, which left little time for running experiments. There are a wide variety of areas that could be readily investigated using the platform developed, and we will now review a few of these.

Firstly, it would be interesting to use an Ornstein-Uhlenbeck (OU) process input current, which would probably be a better match to *in-vivo* neural inputs. Extending this, the mean and variance of the input current could also be varied, to evaluate the range of validity of SRM, and how robust parameters determined in one input regime

are to different inputs. It would then also be possible to compare the f/I curve showing input current against firing frequency of the SRM to that of the HH model.

In terms of the basic fitting procedure, it would be interesting to investigate the phenomenon of different shaped spikes discussed in section 6.3.2 further. If possible, a small set of types of spike should be identified, and methods to automatically classify spikes developed. Then the correct spike template could be subtracted for each spike, and the resulting subthreshold voltage should be free of 'blip' artifacts. It is also possible that the major difference between the spike 'types' identified in section 6.3.2 is just that they are shifted slightly horizontally, and if this is the case, then it should be possible to eliminate the subthreshold blips by refinements to the spike time identification process.

Another avenue for further enhancements to the basic framework is to smooth the κ kernels by fitting a more suitable function than $A\exp(-t/\tau)$. In particular, a function capable of going below zero would capture much more of the shape of the kernels, for example $Ae^{-t/\tau}(\cos(\omega t+\phi))$.

Finally, it would be very easy to explore the performance of the SRM and the numerical fitting procedure with data from a different conductance based model. There are many readily available models implemented in NEURON, which could be used with the MATLAB framework already developed. This would be especially interesting because, despite being the "canonical" complex model, the HH model is quite different in behaviour to a neocortical pyramidal cell, as it does not have particularly pronounced adaptation effects (Kistler et al., 1997).

7.2.3 Long term

An interesting extension of the framework would be to include dendritic processing in the model. For the conductance model, many NEURON models incorporating dendritic compartments are available. For the SRM, there would be several options for the type of processing to be performed on separate synaptic inputs before these inputs were passed on to the integrating filter component of the model, and these would have to be investigated.

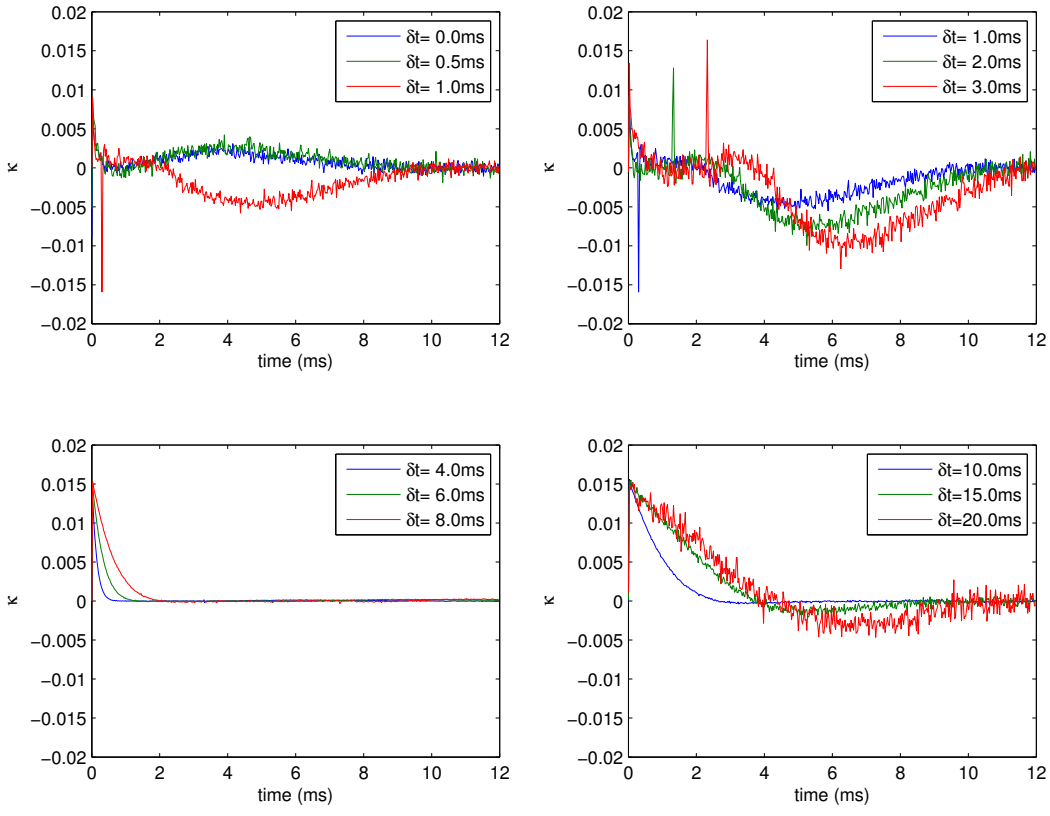
Other longer-term work would be to incorporate a stochastic element into the SRM, and perform a maximum likelihood fit of the parameters to the conductance model data. To do this, the approach of Paninski et al. (2004) and Brillinger and Segundo (1979) would have to be adapted to perform the fit on the basis of the subthreshold voltage, v^{data}, rather than just the spike times.

Appendix A

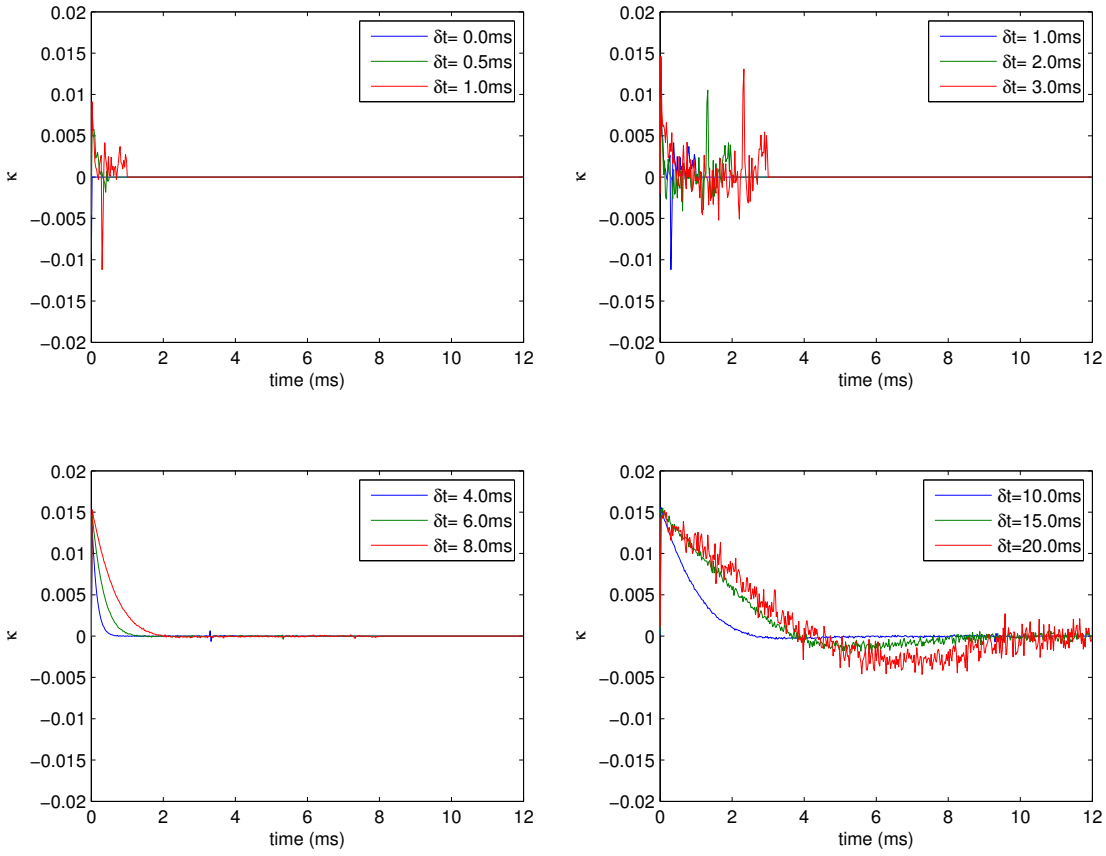
Supplemental figures

This appendix contains additional figures and graphs that were not directly relevant to the main body of the thesis.

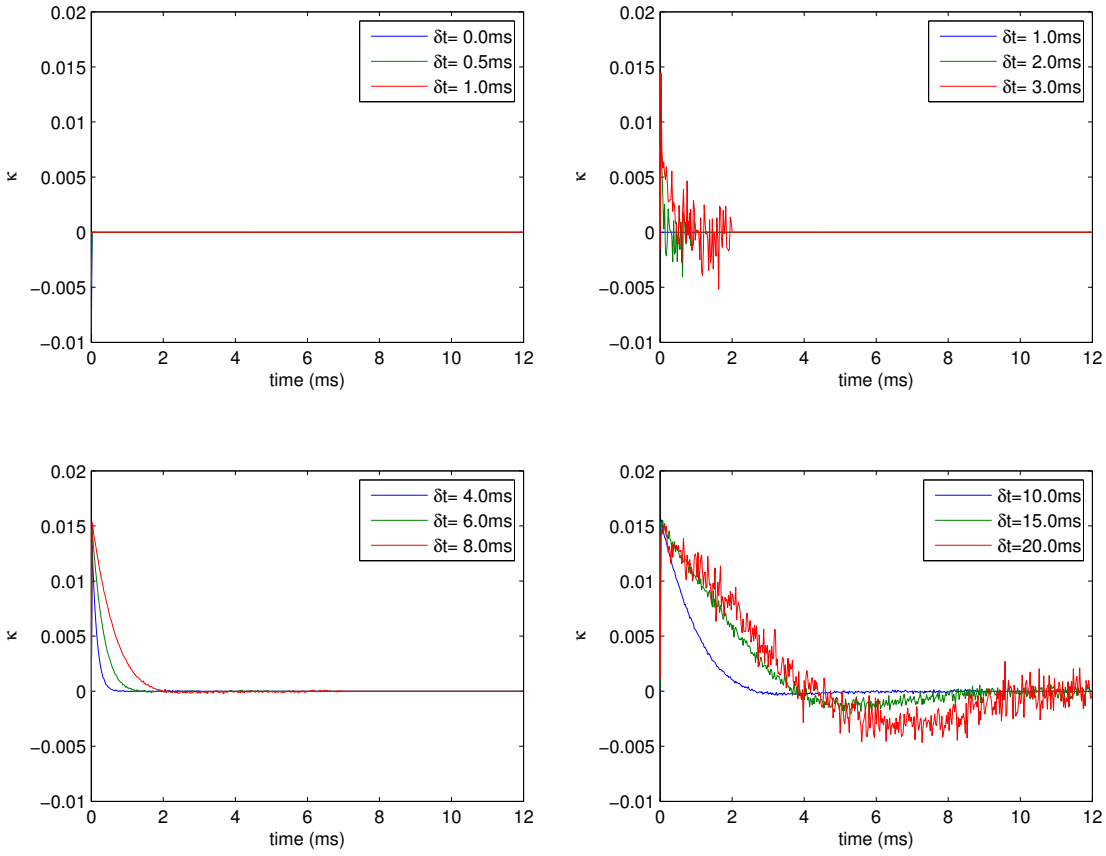
A.1 κ SDK with $s_{max} = \mathbf{len}(\kappa)$



A.2 κ SDK with $s_{max} = \min\left(\mathbf{len}(\kappa), \delta t\right)$

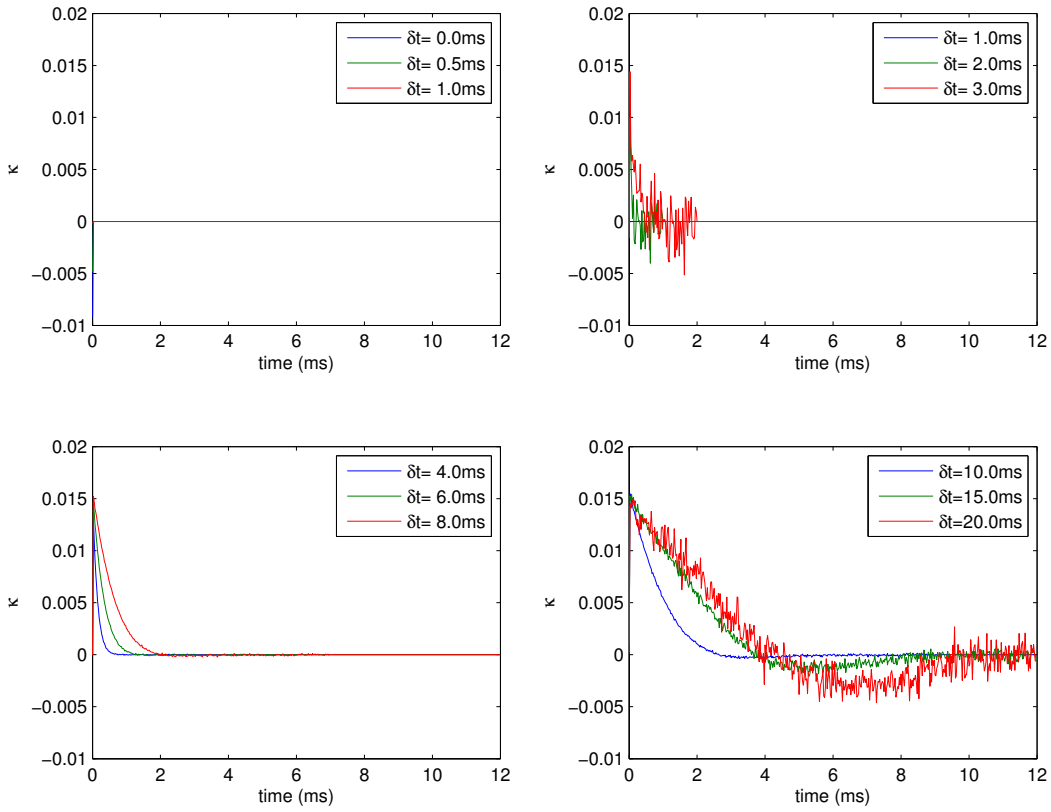


A.3 κ SDK with $s_{max} = \min\left(\mathbf{len}(\kappa), \delta t - 1.0\mathbf{ms}\right)$



A.4 κ SDK with regularisation coefficient $\lambda = 2x10^4$

(Also using $s_{max} = \min(\text{len}(\kappa), \delta t - 1.0\text{ms})$)



Appendix B

Glossary of symbols and terms

C Stimulus covariance matrix, defined by $C_{sj} = \sum_t I_{t-(s-1)} I_{t-(j-1)}$.

δt The time elapsed since the most recent spike, $\delta t = t - \hat{t}$.

η Kernel describing the shape of a spike. Depends only on the time since the last spike: $\eta(t - \hat{t})$.

Γ Spike coincidence factor, $\Gamma = \frac{N_{coinc} - \langle N_{coinc} \rangle}{\frac{1}{2}(N_{HH} + N_{SRM})}$.

$I(t)$ Input current, which was Gaussian white noise.

κ Kernel describing the integration of inputs by the SRM model.

λ Regularisation coefficient, used to regularise the covariance matrix: $C_{reg} = C + \lambda I$.

N_{coinc} Number of spikes that occur at the same time (within a specified tolerance) in the spike train produced by the SRM and the HH model spike train.

N_{HH} Number of spikes in the HH data.

N_{SRM} Number of spikes predicted by the SRM.

s Variable used for the lag back in time from the prediction point, when applying a causal filter (and in particular $\kappa(s)$).

SDK Spike Dependent Kernel. The version of the κ kernel which depends on the time elapsed since the previous spike as well as the lag: $\kappa(\delta t, s)$.

SIK Spike Independent Kernel. The version of the κ kernel which is independent of the time of the previous spike, and depends only on the lag: $\kappa(s)$.

s_{max} Maximum lag considered when creating x and C (so therefore technically measured in number of data points, but sometimes used in a milliseconds context). This defines the length of κ.

SRM Spike Response Model. A one-dimensional neural model which is an extension of the integrate-and-fire model, defined by $u(t) = \eta(t - \hat{t}) + \int_0^\infty \kappa(s) I(t - s)\, ds$.

$\hat{t}$ The time of the most recent spike.

θ Spike threshold. If v^{SRM} reaches θ from below, a spike is predicted by the model.

$v^{data}(t)$ Subthreshold voltage data obtained by subtracting the η template from the output of the HH model.

$v^{SRM}(t)$ Subthreshold voltage data as predicted by the SRM.

x Cross-correlation vector between the stimulus and the membrane voltage, defined by $x_j = \sum_t v_t^{data} I_{t-(j-1)}$.

Bibliography

Abbott, L. and Kepler, T. (1990). Model neurons. from Hodgkin-Huxley to Hopfield. *Lecture Notes in Physics*, (368):5 –.

Aguera y Arcas, B., Fairhall, A., and Bialek, W. (2003). Computation in a single neuron: Hodgkin and huxley revisited. *Neural Computation*, 15(8):1715 – 49.

Brillinger, D. and Segundo, J. (1979). Empirical examination of the threshold model of neuron firing. *Biological Cybernetics*, 35(4):213 – 20.

Calvin, W. H. and Stevens, C. F. (1968). Synaptic noise and other sources of randomness in motoneuron interspike intervals. *J Neurophysiol*, 31(4):574–587.

Erisir, A., Leonard, C., Lau, D., and Rudy, B. (1999). Function of specific K^+ channels in sustained high-frequency firing of fast-spiking neocortical interneurons. *Journal of Neurophysiology*, 82(5):2476–2489.

Feng, J. (2001). Is the integrate-and-fire model good enough? - a review. *Neural Networks*, 14(6-7):955 – 75.

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J*, 1:445466.

Hodgkin, A. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117:500–544.

Jolivet, R., Lewis, T. J., and Gerstner, W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. *J Neurophysiol*, 92(2):959–976.

Jolivet, R., Rauch, A., Lüscher, H.-R., and Gerstner, W. (2006). Integrate-and-fire models with adaptation are good enough. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 595–602. MIT Press, Cambridge, MA.

Kistler, W., Gerstner, W., and Hemmen, J. L. v. (1997). Reduction of the Hogkin-Huxley equations to a single-variable threshold model. *Neural Computation*, 9(5):1015–1045.

Koch, C. (1999). *Biophysics of Computation: Information Processing in Single Neurons*. Oxford University Press, New York.

Lagarias, J., Reeds, J., Wright, M., and Wright, P. (1998). Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1):112–47.

Nagumo, J., A. S. and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proc. IRE*, 50:20612070.

Nelder, J. A. and Mead, R. (1965). A simplex-method for function minimization. *Computer Journal*, 7(4):308–313.

Paninski, L., Pillow, J., and Simoncelli, E. (2004). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Computation*, 16(12):2533 – 61.

Zohary, E., Shadlen, M., and Newsome, W. (1994). Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature*, 370(6485):140 – 3.