Neuronal Calcium Imaging Signals
Modeling and Analysis

Thomas I Beka

Norwegian University of Life Sciences

Department of Mathematical Sciences and Technology

Master Thesis 30 credits 2011

This page intentionally left blank

Acknowledgment

This thesis is submitted in partial fulfillment for a master's degree at the Norwegian University of Life Sciences, UMB.

I would like to use this occasion to thank my inspiring supervisor Professor Gaute T. Einevoll, who has kindly given me the opportunity to work on this delighting project, and made many critical comments during our weekly discussions.

I owe my deepest gratitude and admiration to my supervisor Dr. Szymon Łęski (UMB/Nencki Institute of Experimental Biology), who has tirelessly shown his commitment to answer my questions and spotted numerous mistakes. Without his support this thesis would not have been possible.

It is an honor for me to express my gratitude to Associate Professor Anna Devor (Harvard/UCSD) for initiating this thesis project, while I was a visiting student in her lab at UCSD, and Dr. Lydia Reznichenko for providing me with experimental data.

I owe a special thanks to Professor Anders M. Dale (UCSD) for making himself available for constructive discussions and contribution to the project.

I am indebted to my many of my fellow students, friends and family members for their attention and cheerfulness.

Ås, 13^{th} of May, 2011

Thomas I. Beka

thomab00@gmail.com

This page intentionally left blank

Abstract

The advent of two-photon calcium imaging in vivo has presented a new arena to detect neuronal action potentials and identify neuron types based on their fluorescence signatures. However, despite the growing popularity, reconstructing spike patterns from the fluorescence traces still remains a major challenge. Also, not much is usually said about how the calcium waveforms corresponding to a spike (calcium kernel) should be estimated. In this thesis, we present a novel approach for calcium kernel estimation from slopes of a fluorescence trace by combining the Savitzky-Golay filter with an iterative algorithm for fitting a nonlinear model (Levenberg-Marquardt). We also present a new method for spike detection, which employs deconvolution and greedy optimization. First we test these methods on synthesized calcium signals, and then we apply them to experimental traces from wild-type and transgenic mice expressing human α- synuclein (model of Parkinson's disease). We show longer calcium response in the somatosensory cortex neurons of the transgenic mice, read-out both spontaneous and evoked activities as well as follow the hierarchy in fluorescence transient elevation arrivals when mice whiskers were stimulated electrically.

This page intentionally left blank

Contents

I **12**

1 Introduction **12**

1.1 Organization of the thesis 15

2 Biological Background **16**

2.1 Neurons 16

2.2 Action potentials 17

2.3 Parkinson's disease 18

2.4 Ca^{2+} dynamics in neurons 18

3 Experimental Background **20**

3.1 Two-photon microscopy 20

3.2 Experimental methods 21

3.3 Experimental data 24

3.3.1 Data processing 24

3.3.2 Fluorescence traces of calcium imaging 25

3.3.3 Experimental data examples 26

4 Mathematical Background **28**

4.1 A mathematical model of Ca^{2+} signals 28

4.2 Formulation of the problem 30

4.3 Existing algorithms 31

5 Methods: Modeling **33**

5.1 Generating calcium signals 33
5.1.1 Spike generation 33
5.1.2 Kernel generation 35
5.1.3 Output signal generation 36
5.1.4 Adding Noise 39
5.2 Estimating kernels from data 42
5.2.1 Kernel amplitude 42
5.2.2 Kernel clearance time constant 44
5.3 Estimating spike trains 47
5.4 How to supplement deconvolution 50
5.4.1 Median Filtering 50
5.4.2 Optimizing in the greedy approach 51

6 Results: Modeled data **53**

6.1 How well can we estimate a kernel? 53
6.1.1 h_e and its error term when threshold is imposed based on data 54
6.1.2 Imposed threshold via the S-Gf algorithm 59
6.1.3 Kernel time constant 61
6.2 How well can we reconstruct spikes? 72
6.2.1 Spike train reconstruction with estimated kernels: DDA 72
6.2.2 Median-filtering when kenel is estimated 78
6.2.3 Optimizing the DDA when kernel is estimated 81
6.2.4 How better can we reconstruct spikes if the kernel is known exactly? 84

7 Application **87**

7.1 Threshold parameters and the error term 87

7.1.1 Three threshold parameters 87

7.1.2 Error term 88

7.2 Traces of two-photon calcium imaging in vivo 89

7.2.1 Selected examples 89

7.2.2 How do 1- and 3-pulses traces differ? 91

7.3 Estimating calcium kernels 92

7.3.1 1-pulse wild-type trace: Parameters as for model data 92

7.3.2 1-pulse wild-type: $k = 5, 7, 9, 11$ and $\alpha = 1$ 93

7.3.3 Fitting functions to calcium slopes when k is large . . 96

7.3.4 1-pulse transgenic 99

7.3.5 3-pulses transgenic 100

7.3.6 3-pulses wild-type 102

7.4 Summarizing kernel parameter estimation 103

7.5 Spike train reconstruction and calcium trace fit 104

7.5.1 1-pulse wild-type 104

7.5.2 1-pulse transgenic 107

7.5.3 3-pulses wild-type 109

7.5.4 3-pulse transgenic 112

8 Discussion **116**

8.1 Modeled Ca^{2+} signals 116

8.1.1 Kernel amplitude 116

8.1.2 Kernel time constant 117

8.1.3 AP reconstruction 118

8.2 Experimental signals 119

8.2.1 Amplitude and time constant 119

8.2.2 AP 119

8.2.3 Conclusion 120

II Appendix **122**

9 Abbreviations **122**

10 Parameter Symbols **123**

11 MATLAB scripts **124**

Part I

1 Introduction

In recent years, there have been significant progresses in understanding the molecular pathway and synaptic pathology of Parkinson's disease, a progressive neurological disorder involving the basal ganglia [24].

However, translation of these results into clinically effective treatments remain a major challenge. Limited understanding of the key molecular events that evoke neurodegeneration is the main obstacle stopping scientists from developing neuroprotective therapies [6]. On a molecular level, it is now well established to consider a protein known as α-synuclein (α–syn) as a key component of the disease [26].

We have had an interest to investigate if α-syn perturbs Ca^{2+} homeostasis in neurons (section 2.4) [9]. For this purpose, in vivo two-photon calcium imaging is performed on neurons of wild-type (control) and transgenic (expressing human $\alpha-$syn) mice (section 3.2).

Two-photon calcium imaging in vivo is gaining recognition in the scientific community. This state-of-the-art imaging technique is favored due to the enormous possibilities it provides to image deeper (0.5-1 mm) in the brain with good sub-micron visibility [16], high spatial (up to 1 μm) and temporal (10 -10^4 Hz) resolution [21, 18]. In addition, the technique allows simultaneous and prolonged imaging for a population of neurons [16].

Calcium imaging signals are often used to detect firing of action potentials. However, reconstructing a spiking pattern from the fluorescence calcium signatures is not yet well resolved. Since the second half of the last decade, a number of algorithms have been proposed to sort out spikes from the fluctuating fluorescence traces [44, 35, 42, 14, 20]. Some of these algorithms have

addressed the problem from image processing point of view [42], others have approached the problem with methods used for machine learning [35], while some groups have tackled the problem from signal processing point of view or combined the image and signal approaches [44, 12]. In this thesis, we address the problem from a signal processing point of view.

As it is described in more detail in section 4, most of the groups which have worked with the problem have put forward different forms for deconvolution as a spike detecting algorithm: For instance, Yaksi and Friedrich proposed temporal deconvolution [44], Holekamp et al. considered optimal linear deconvolution [20], whereas Vogelstein et al. have suggested non-negative deconvolution [42]. In addition, some of the groups have supplemented deconvolution with data filtering (for instance, Wiener [20], Butterworth low-pass [44]) as a pre-processing procedure, as well as with data on measured activity by techniques other than multiphoton imaging (for instance, loose-patch clamp recordings [35]).

Deconvolution is favored under the assumption that each AP evokes a unitary, usually exponentially decaying, Ca^{2+} transient [12, 44]. As a result, the time-varying fluorescence trace can be approximated by a convolution of a time series of instantaneous events of AP with a kernel expressing the Ca^{2+} transient plus noise [44] (see section 4.1).

Nevertheless, none of these deconvolution algorithms have clearly shown how the calcium transient kernel should be constructed and how its parameters should be estimated (see section 4.3).

In this thesis we attempt to develop and test algorithms that can estimate calcium clearance time constant, sort out the activity pattern (evoked and spontaneous) from the traces and make a suggestion for the feasiblity of identifying neuronal cell types based on their unique fluorescence signatures.

A novel algorithm for kernel parameter estimation from slopes of a given calcium trace will be presented in this thesis. We will develop this algorithm by combining the Savitzky-Golay filter with an iterative algorithm for fitting a nonlinear model (Levenberg - Marquardt).

Once the kernel is estimated, APs can be deterministically sorted out from a trace with deconvolution. An alternative approach than the conventional Fourier domain method will be presented for this deconvolution.

We improve the AP estimations of deconvolution with an optimization algorithm in the greedy approach.

We model a family of calcium signals mimicking experimental data.

The modeled signals will play crucial roles to understand the dynamics of calcium signals. Designed algorithms will be fine-tuned on the modeled signals.

In additon, the modeled signals will be used as a test data for comparing algorithms as well as fit for parameters that we require for later use in the experimental data application and analyses.

The fine-tuned algorithms on the modeled data will be applied on selected two-photon calcium imaging signals in vivo at the end of the text.

The theme of this thesis is motivated by an ongoing research on Parkinson's disease (see section 2.3) at the laboratory of Dr. A. Devor at the University of California in San Diego.

1.1 Organization of the thesis

This thesis is organized in such a way that: In part I, section 2 starts by giving the necessary biological background, in section 3 we briefly present two-photon microscopy, our experimental methods as well as how the data is processed. In section 4, we do the mathematical modeling and survey existing algorithms in the field. Section 5 addresses the procedures for Ca^{2+} signal modeling and algorithms design. In section 6, we use the modeled signals to test and fine-tune algorithms as well as compare results. In section 6, algorithms will be applied on selected two-photon Ca^{2+} imaging signals in vivo. Section 8, will summarize, discuss and conclude on the major results of the thesis. Part II is dedicated to Appendix, where in section 9 and 10 the most important abbreviations and symbols of parameters are listed respectively. In section 11A to E, we have presented the major MATLAB scripts used for the modeling process in this thesis.

Parts (red), sections (red), Table of contents (red), Figures (red), Equations (red), footnotes (red), citation-forward (green) and citation-backward (red) are all cross-referenced throughout the text. In addition, hyperlinks are available in the electronic version of the text.

2 Biological Background

2.1 Neurons

Neurons are cells in the central nervous system that are connected to each other in an intricate pattern [11]. Neurons are specialized in integrating, processing and transmitting information [4]. For instance, these cells configure muscle movements as well as brain activities, such as learning and speech. These tasks of high coordination are possible due to the neuronal capability to conduct electrical impulses at high speed and over long distances.

There are about 10^{11} neurons in a human brain [23]. Morphologically, a neuron is divided into three distinct parts (Figure 2.1) : A cell body or soma, dendrites and an axon. In the soma, nucleus and the majority of cytosolic organelles are contained [19], serving as a central processing unit [11]. Dendrites are branchy in structure and emanate from the soma [19]. These structures are specialized in bringing signals from other neurons to the soma. Their branchy structure enlarges the receptive volume for increased connection to other neurons [4]. Typically, a neuron can receive inputs from more than $10,000$ other neurons via its dendritic synapses [23]. An electrical input signal that is above a certain threshold will be taken away from the soma by an axon, a tubular process extended towards target cells [4, 11].

Due to the presence of a barrier in the form of cell membrane, ions cannot freely wander between the intracellular and the extracellular medium of a neuron. Na^+, K^+, Ca^{2+} and $Cl^-$are the four predominant ion types involved in trans-membrane electric potential transaction [7]. However, these ions can cross the barrier in response to electrical stimulation.

2.2 Action potentials

The active electrical response of a neuronal membrane to a stimulus is called an action potential, also called a spike [4]. Action Potentials (APs) are the prime means of communication between neurons [7].

The process of transmitting information from one neuron to the next is called AP firing. When a neuron is at rest, its potential inside the membrane is about -65 to -70 mV [11] relative to the medium outside of the membrane. Neurons fire when incoming current from other neurons become large enough to depolarize the membrane potential above a certain threshold of resting potential [23, 4].

In general one AP can differ from the other in its width, amplitude and shape; nevertheless, they are stereotypically treated as being similar. In neuronal encoding, when information conveyed by APs studied, one is interested in the number and the timing of spikes, the form is usually irrelevant [7]. Typically, the potential density of an AP can reach about 100 mV and remain high for about $1 - 2$ ms.

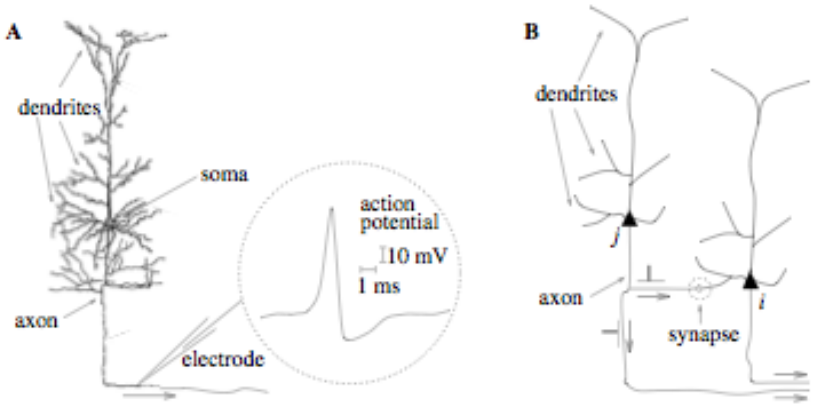


Figure 2.1: **A**. *Ramón y Cajal's drawing of a single neuron showing a soma, an axon and dendrites. In the inset, a typical action potential of amplitude* 100 *mV and duration about* 2 *ms is displayed.* **B.** *A presynaptic neuron* j *transfering a signal to a postsynaptic neuron* i *through the synapse (marked by the dashed circle). Adopted from Gerstner and Kistler,* 2002.

When a chain of APs is released by a single neuron, the sequence at (regular or irregular) intervals is called a *spike train* [11].

2.3 Parkinson's disease

According to a London based PD research group VIARTIS, more than 6 million people were diagnosed with PD in 2009, worldwide [41]. Other statistical data have indicated that nearly 2% of the aging population ($\geq$ 65 years) suffers from movement impairments due to PD [4]. In most cases, early (30-40s years of age) PD-onset is associated with heredity, whereas onset after 60 is considered as sporadic [26].

The four main symptoms of PD are tremor, slowness of movement (bradykinesia), rigidity, and postural instability [4]. Other clinical symptoms of the disease include fixed facial expression, changes in speech and handwriting, excessive sweating, psychological problems such as depression, and in late stages dementia [34]. The motor impairments are primarily due to degenerating dopaminergic neurons in the substantia nigra and the loss of their dopaminergic projections to the striatum [6]. The non-motor symptoms of Parkinson disease are gaining much more attention than before since they contribute greatly to disability as the disease progresses. In addition, non-motor impairments do not respond to dopaminergic drugs as motor impairments do [4]. So far there is no definitive agent known to slow down disease progression at the cellular level in Parkinson disease [3].

It is argued that the basic pathology of PD is the death of dopaminergic neurons in the pars compacta of substantia nigra, part of the brain that sends dopaminergic projections to the basal ganglia [6]. In the midbrain substantia nigra covers a large area darkly pigmented with melanin, a metabolic by product of dopamine breakdown [15].

Recently, it has been speculated that the development of PD to have a positive link with the rate at which α-syn, a small acidic synaptic terminal protein of 140 amino acids, forms aggregates [6]. It is also indicated that inability to regulate membrane permeability for Ca^{2+} as one feature of neurodegenrative diseases [22].

2.4 Ca^{2+} dynamics in neurons

The electrical activity of neurons is sustained via ionic currents that propagate through neuronal membranes [7, 23]. Calcium ions are one of the ionic species involved in this transmembrane current transaction [23].

In general, there are two isolated calcium storages in the brain: The extracellular environment and the intracellular regions. The former stores a large amount of calcium supply, whereas the endoplasmic reticulum (ER), mitochondria and the lysosome of the intracellular regions also serve as Ca^{2+} reservoir [31, 2].

The storage capacity of the intracellular region is affected by Ca^{2+} buffering proteins, such as the α-syn nucline, however, the entire dynamics of this is not yet well resolved [2].

Mainly, two phenomena cause the cytosolic calcium level to fluctuate: Each AP increases the membrane Ca^{2+} conductance [23] leading to a brief Ca^{2+} influx (typically lasting for 1 ms) through membrane channels [27, 5]. The brief influx causes transient elevation, usually denoted by $[\Delta Ca^{2+}]$, in intracellular calcium concentration [5, 12]. The excess intracellular calcium is typically cleared out from a cell in an exponentially decaying manner [12]. Secondly, Ca^{2+} can emerge from the intracellular stores via channels operated by special receptors [39]. For instance, mitochondria takes up Ca^{2+} during cytosolic surplus and releases Ca^{2+} during cytosolic deficiency [2]. In this thesis we consider only APs as the causes of predominant intracellular calcium elevations.

In the case of PD, it is assumed that, cells might be able to adjust to Ca^{2+} irregularities in early stages of the disease by clearing excess calcium into intracellular stores and out of the cell. The ongoing research at the laboratory of Dr. A. Devor at UCSD, which motivated this thesis, focuses on how α-syn oligomers perturb neuronal calcium homeostasis in vivo. The objective of the study is detecting differences in Ca^{2+} dynamics in neurons of wild-type and transgenic (α-syn) mice. For this purpose, we have employed two-photon calcium imaging in vivo. In this thesis, we attempt to estimate exponential calcium kernel parameters and reconstruct the spike trains of the two-photon imaging signals.

3 Experimental Background

3.1 Two-photon microscopy

Two-photon microscopy is a relatively new imaging technique that employs laser beams [16]. The physics behind the technique is exciting a fluorescent target molecule by two lower energy photons [28]. However, to achieve the necessary amount of exitation, the two photons should be absorbed almost simultaneously (typically within 1 fs) by the target [16]. In this case, the energy increase achieved by the target molecule is equivalent to the sum of the individual photon energies [30]. During relaxation, an exited target molecule emits fluorescence and the traces of the fluorescence are recorded as an image signal.

Biological tissues scatter ultraviolet (UV) and visible light radiations, which is the operating wavelength range for single-photon (confocal) microscopes [28]. On the other hand, tissues are transparent to near-infrared (n-IR) radiation, ranging from 700 nm to 1000 nm [30]. Two-photon microscopes operate in the n-IR region; therefore, they are more suitable for in-depth brain imaging in vivo. In ideal situation, 500 μm up to 1 mm penetration depth can be reached with two-photon laser scanning in the cortex [17]. In addition, since n-IR stores lower energy, the technique is less phototoxic to a tissue [28].

A spatial resolution as high as 1 μm as well as a 10 Hz to 10 kHz temporal resolution can be reached with two-photon imaging [21]. In Figure 3.1, we have illustrated a spatiotemporal resolution (in logarithmic scale) of various imaging techniques available in neuroscience. In this case, two-photon calcium imaging is categorized under optical imaging.

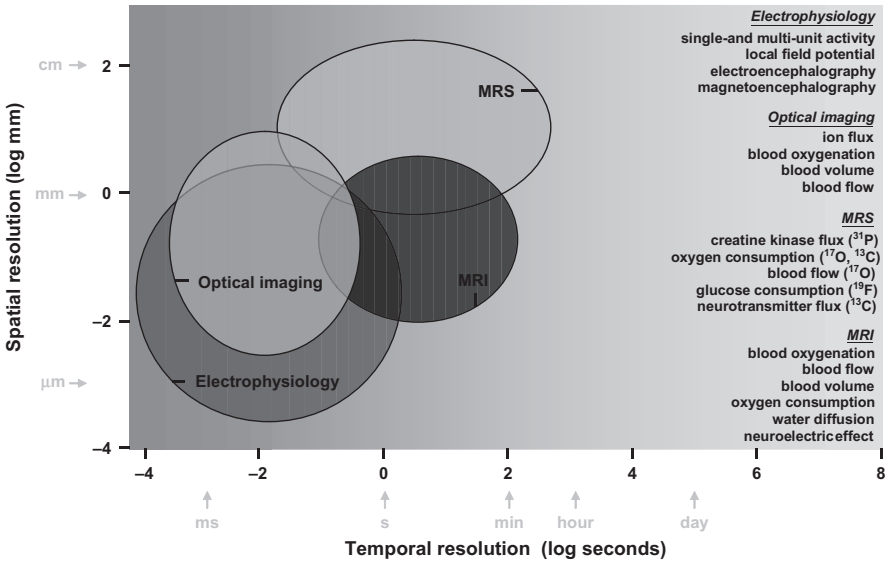


Figure 3.1: *Comparison of spatiotemporal resolutions of various imaging techniques available in neuroscience. In this case, two-photon calcium imaging can be classified under the* Optical imaging *in the illustration. Adopted from F. Hyder, 2009*

3.2 Experimental methods

The experiment is designed to investigate the role of Ca^{2+} in PD neurodegeneration in vivo. For this purpose, traces of emitted fluorescence in the primary somatosensory cortex (SI) are recorded through two-photon microscope for a group of 12-month old transgenic mice expressing human α-syn under the mThy-1 promoter, and their age-matched wild-type littermates (Controls) under a similar condition. The transgenic mice used for the study have fully developed PD. Actual imaging is made in the barell cortex across the depth down to 300 μm. The barrel cortex is chosen for this purpose due to the well-resolved mechanism between each barrel and its functional connection to a single whisker [4].

Mice were anesthetized with isoflourane and ventilated with a mixture of air and oxygen during a surgical procedure prior to imaging. Mice heart rate, blood pressure, expired CO_2 and body temperature were continuously monitored throughout the experiment. Respiration was aimed to maintain

pCO_2 between 35 and 45 mmHg. During the surgery cannulas were inserted in the femoral artery and vein, Isofluorane was discontinued and anesthesia was maintained.

An area of the skull overlying SI was exposed, the skull and dura mater were removed, and the space between the exposed brain surface and the cover glass was filled with agarose (Sigma), for further detail on the the practical procedures refer to Devor et al., 2008 [8].

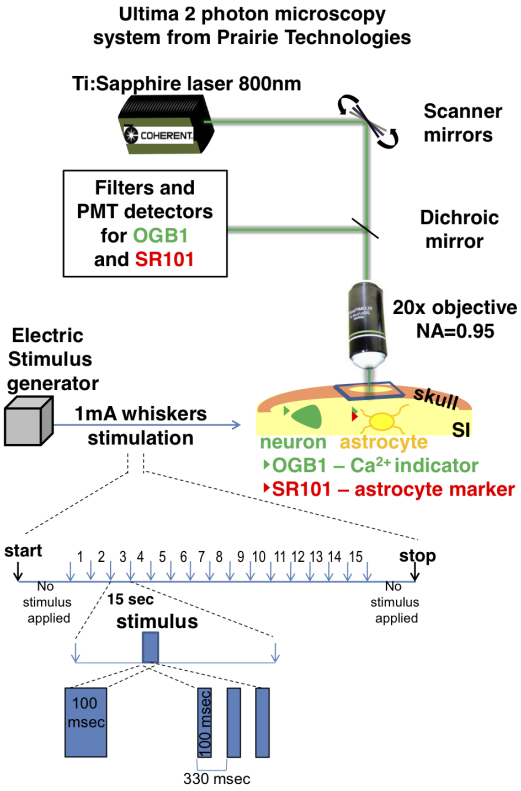


Figure 3.2: *Top: An Ultima two-photon laser microscope system set-up from Prairie Technologies. The microscope uses a titanium (Ti) sapphire laser of* 800 *nm. PMT is a short form for photomultiplier tubes (vacuum), NA abbreviates object side numerical aperture (opening). A Dichroic mirror shows different colors when viewed from different directions due to different absorption coefficients for light polarized in different directions. SI refers to the primary somatosensory cortex. Bottom: Experimental flow. blue arros denote individual stimulus, in total 15. Stimulation is given through whiskers, either as a single electric shock in* 100 *ms (bottom left blue rectangle) or as a train of 3-pulsess (three assembled rectangles at bottom right). Adopted from Dr. Devor's Lab (L. Reznichenko) Sfn 2010 poster.*

The center of neuronal response was mapped with a silver ball electrode prior to the two-photon imaging. This is done by measuring the surface potential recording of the so-called multi unit activity (MUA). After mapping the neuronal response using surface potentials, calcium indicator Oregon Green BAPTA-1 AM (OGB1) and glial marker SR101 were micro-co-injected in the center of the responsive area about 300 μm below the cortical surface. In addition to exposing the skull, two minor surgeries are performed. The first one for catheterizing femoral artery for blood pressure monitoring and applying anesthesia, and the second one is tracheotomy surgery for planting supportive breathing apparatus. All surgeries are performed according to the US National Institutes of Health (NIH) guidelines for the care and safety of laboratory animals. In addition, all experimental procedures were approved by the University of California at San Diego Institutional Animal Care and Use Committee.

Images were obtained using 4-channel Ultima 2-photon microscopy system from Prairie Technologies, microscope details are shown in Figure 3.2 or can be accessed at Prairie-technologies website [1].

For practical reasons, calcium activities are imaged at three separate cortical layer frames: i) Cortical layer I, about 100 μm below the surface ii) cortical layer II, about 150 μm below the surface; and iii) cortical border layer II-III, about 250 μm below the surface. In each layer, on average about 10 neuronal cell bodies are imaged within approximately 50 X 100 μm field of view (FOV) at acquisition rate of 10 Hz. Whiskers are electrically stimulated with 0.3 mA to 1 mA, in a single pulse during 100 ms or in a 3 Hz train of pulses during 1 s. There are 15 stimulus trials for each FOV with an interstimulus interval (ISI) of 15 s, details for the flow of the experiment are shown in Figure 3.2. In addition, for each FOV both (1 Hz and 3 Hz pulses) the stimulus conditions are presented. Besides stimulated trials, some stimulus free (blank) trials are collected for the sake of estimating spontaneous occurrence and duration.

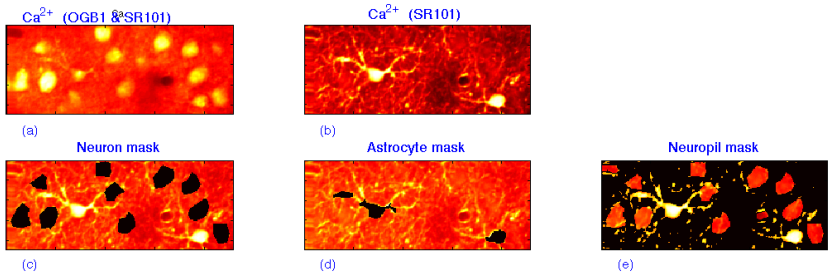
3.3 Experimental data

3.3.1 Data processing

Raw experimental data is processed and transformed into easily accessible and manipulable formats with MATLAB. Regions occupied by neurons, astrocytes and vessels in a raw image are identified and masked as regions of

interest (ROI). A typical example of experimental image with masked ROIs is shown in Figure 3.3.

Figure 3.3: *Regions of neurons, astrocytes and neuropiles will be identified from the two-photon imaging data and masked as shown in the figure.*



In Figure 3.3: (a) Due to high calcium concentration, some spots look brighter. The brighter spots can represent either neuronal calcium as marked by OGB1 or astrosytic calcium marked by both OGB1 and SR101. On the other hand, the dark spots represent locations of vessels. These locations look darker because they lack of calcium, therefore, unmarked neither by OGB1 nor by SR101. (b) Astrosytic spots are sorted out from the brighter spots of (a) due to their unique SR101 content. (c) Black spots denote neuron masks, i.e. the brighter spots in (a) minus the astrosytic spots of (b). (d) Black spots denote astrosytic locations. (e) Black region denote a neuropile, i.e. the entire ROI minus vessels, neurons and astrocytes. Or, in other words, a neuropile is the unmarked ROI as neurons, astrocytes and vessels.

3.3.2 Fluorescence traces of calcium imaging

Change in calcium level can be read out from the fluorescence emission made by a stained cell

$$\Delta[Ca] \quad \alpha \quad \frac{\Delta F}{F} \tag{3.1}$$

where $\Delta[Ca]$ implies the change in calcium concentration, α implies proportionality and $\frac{\Delta F}{F}$ implies the fractional fluorescence change [12]. $\frac{\Delta F}{F}$ is the

change in the amount of emitted light intensity by a cell, an astrocyte or a neuropile [16]. $\frac{\Delta F}{F}$ increases with increasing calcium concentration in the presence of (unsaturated) calcium dye [4].

3.3.3 Experimental data examples

Pixels of the masked regions serve as a basis to construct calcium imaging fluoresence traces. In Figure 3.4, fluorescence traces of a neuropile, astrocytes and neurons are illustrated in blue, red and green respectively. Each trace lasts for 285 s as indicated in section 3.2. The first and the last 20 second intervals of a given experimental run are blank (non-stimulated), therefore, dominant elevations observed in these intervals must have been caused by spontaneous activities.

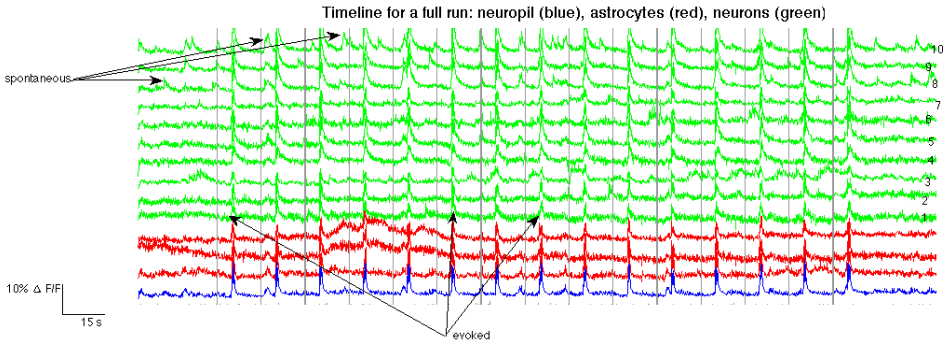


Figure 3.4: *Fluorescence traces of a full two-photon Ca^{2+} imaging experimental run for a transgenic (α-syn) mouse: Neurons (green), astrocytes (red) and a neuropil (blue). Examples of evoked and spontaneous events are indicated with arrows. Stimulus arrives to the region of interest (neurons, astrosytes and neuropil) about 5 s after the gray vertical bars, therefore, elevations in that vicinity are evoked activities. A small sized masked region is more likely to give noisier signal due to lower picture element inclusion. This is one of the reasons why the neuropil trace (blue) seems less noisier than the green (neuron) traces.*

In Figure 3.5, we have illustrated a full run fluorescence trace ($\frac{\Delta F}{F}$) for a single neuron as a function of time. In this thesis, we will attempt to model such a trace. Later on we exploit the models as a test data to develop algorithms that can be used to estimate some parameters and reconstruct spike trains of the real data.

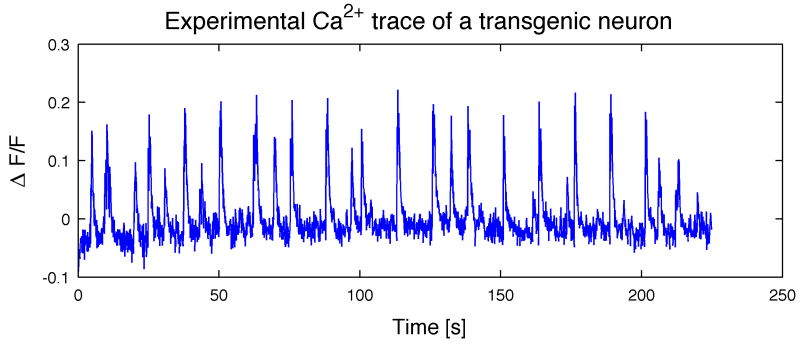


Figure 3.5: *Experimental trace for a transgenic trace.*

4 Mathematical Background

Detecting the level of intracellular calcium concentration by means of a fluorescence trace from two-photon imaging is gaining popularity in neuroscience. Excited fluorophore, used to stain neurons and astrocytes as a calcium indicator, emits light while relaxing to a lower energy state. The amount of emitted fluorescence, however, depends on cells activity. In calcium imaging, one records the trend of fluorescence fluctuation caused by a changingcalcium level. Fluctuation in calcium in turn is a result of increased membrane Ca^{2+} conductance in course of action potential positive feedback effect.

Nowadays, two-photon calcium imaging serves as an alternative and a complimentary technique to neuronal electrical recordings. In fact, two-photon imaging is sometimes preferred due to its convincing results in intact brain studies in vivo [44].

Reconstructing spiking patterns from a fluorescent trace, however, is not a straightforward process. This is mostly due to various measurement errors and artifacts. In recent years, various spike-train-inferring algorithm for calcium imaging fluorescence traces have been proposed. Nevertheless, non of the existing algorithms has managed to fully resolve the quest for spike inference from fluorescent traces.

4.1 A mathematical model of Ca^{2+} signals

In this text we deal with three signal types: *i*) A spike output made by a neuron, *ii*) A broadening signal, and *iii*) A wave-like calcium response. Type

i is the one that we detect as an input signal in our system, we simply refer to it as a *spike* (pl. *spike train*). Type *ii* will be referred as a broadening (calcium) *kernel*, whereas, type *iii* will be recognized as a calcium *output* signal.

Suppose any spike output made by a neuron gives a characteristic wave-like calcium response. Therefore, a calcium output signal is composed of many repetitions of the same wave-like calcium response. Mathematically this is convolution. Roughly speaking, convolution is a sort of signal blending. In signal processing, convolution is perhaps one of the most important algorithms in terms of relating three signals of interest [38]: Such as a spike, a broadening kernel and a calcium output signal mentioned above.

In mathematical terms, convolution is written in the form of

$$(\kappa * \eta)(t) = \psi(t) \tag{4.1}$$

where $\kappa(t)$, $\eta(t)$ and $\psi(t)$ represent a broadening kernel, a spike train, and an output signal respectively, whereas $*$ denotes the convolution operator. In addition, the output signal $\psi(t)$ is explicitly defined as

$$\psi(t) = \sum_{t'=0}^{\infty} \kappa[t'] \cdot \eta[t - t'] \tag{4.2}$$

The above expression is useful when convolution is applied on discrete input signals. Note that, convolution can also be performed on continuous impulse signals. However, the mathematical justification of the procedure is a bit more complicated. One of the reasons for that is, the mathematical abstraction of the continuous delta function used to approximate continous impulse signals [38]. The spike train we are going to be looking at in this thesis is discrete, hence, extending the discussion of convolution to contiuous signals will not be necessary here.

The calcium kernel $\kappa(t)$ used in the convolution will have an exponentially decaying form. This is due to neuronal excess somatic calcium clearing mechanism discussed in section 2.4.

4.2 Formulation of the problem

In section 4.1 we have surveyed how to construct output signals from two known signals, namely a spike train and a kernel. In this subsection we will briefly present a strategy to invert the convolution in order to estimate spike trains.

Basically, the core of this thesis is making an attempt to estimate input signals (spikes and kernels) from calcium output signals as accurately as possible.

We attempt to reverse the convolution that brought forth calcium output signals with *deconvolution*. We perform the deconvolution *deterministically* in a sense that we acknowledge the exponential form of the unknown broadening kernel. Deconvolution is commonly performed in the Fourier domain [38]. However, in this thesis, we will develop an alternative approach to the conventional method of deconvolution. This issue will be revisited in section 5.

4.3 Existing algorithms

Several computational techniques have been proposed to infer spike trains for a population of neurons from a noisy multiphoton calcium imaging signals. Some of these techniques are worked out through direct analysis of images generated by the multiphoton fluorescence microscopes, while others (including us) have analyzed the 1D time-series fluorescence transient traces constructed based on the images. However, in both approaches dominant elevations observed in fluorescence transient signals are accounted for action potential firings [12, 35, 42, 44].

In the following paragraphs, we will briefly survey some of the proposed algorithms for the purpose of reconstructing spike trains for a population of neurons.

One of the earliest works in this field has suggested extracting event onset time from the first derivative of fluorescence traces with imposed threshold on the fractional fluorescence change $\frac{\Delta F}{F}$ [33, 36, 42].

While studying a hippocampal slice, Sasaki et al. have introduced a spike detection algorithm by combining principal-component analysis (PCA) and support vector machine (SVM) [35]. Sasaki et al. have argued that their machine learning based PCA-SVM algorithm to be better suited to reconstructing spike trains than earlier semi-manual calcium transient threshold procedures. However, their algorithm is heavily reliant on the existence of high signal quality, in addition to requiring hundreds of data examples (both image and physiological) to train the algorithm on how fluorescence react to AP [42].

Yaksi and Rainer [44] have suggested a technique that makes use of temporal deconvolution, while studying odor-response pattern in populations of zebra fish olfactory bulb neurons. They have argued that due to cells ionic content, AP opens voltage-gated Ca^{2+} channels causing a unitary Ca^{2+} transients with a fast rise followed by exponential tailing. One of the challenges this study has raised is that during trains of APs individual transients summate causing a complex temporal variation in Ca^{2+} concentration. Furthermore, this study have proposed the possibility of supplementing cell type identification through spectral separation of Ca^{2+} transients and genetically encoded cell-type markers.

Another suggestion for spike detection in a fluorescence trace comes from the work of Göbel and Helmchen [12]. In their study, Göbel and Helmchen have approximated each fluorescence trace by the convolution of a spike train with a single AP evoked calcium transient plus some additive noise. However, the main focus of this study was attempting to precisely estimate and improve signal quality based on dye saturation analysis.

Greenberg and his colleagues [14] have employed template-matching algorithm to study firing rates and identify individual spikes for populations of neurons in the visual cortex of awake rat.

Holekamp and colleagues [20] employed optimal linear deconvolution, a signal processing approach that involves Wiener filtering the fluorescence image data [42].

Recently, Vogelstein et al. [42] have proposed a non-negative deconvolution algorithm for spike inference from fluorescence calcium imaging data. This algorithm involves filtering the image data to enhance the image quality and infers the approximately most likely spike train.

In addition to multiphoton fluorescence data, most of the groups mentioned above have employed other activity measurements such as the patch clamp recordings [35] as a supportive procedure in their spike reconstruction algorithm.

Even though most of the proposed algorithm in the literature employ deconvolution with an inverted kernels as a spike detection algorithm, non of them have clearly specified how these kernels should be constructed from the fluorescence transient signals.

In this thesis we model a family of calcium signals.

Based on the modeled signals, we develop and test a novel algorithm that enables to estimate calcium kernels from slopes of a calcium transient.

The estimated kernels will be used to develop a new optimized deconvolution algorithm in the greedy approach for detecting spikes.

5 Methods: Modeling

In this section we present the modeling and data analysis process in more detail, and show how models are implemented and visualized in MATLAB.

The section is organized as follows: In section 5.1 some assumptions regarding action potentials will be presented. Thereafter, based on assumptions on experimental data set; spike trains, kernels and model signals will be generated. In section 5.2, we present how we can estimate kernels from noisy modeled calcium signals. Our prime spike estimation algorithm will be introduced in section 5.3. Section 5.4 addresses techniques that can improve the spike estimation algorithm discussed in section 5.3.

All computational task is performed with MATLAB 7.9.0 (R2009b) on Mac OS X, 2.26 GHz intel core 2 Duo processor.

5.1 Generating calcium signals

5.1.1 Spike generation

Assume that a typical neuron is capable of firing a number of spikes (action potentials) during a trial duration spanning over several seconds.

All spike occurrences during the trial will be detected and the arrival timings will be listed as follows:

$$\mathcal{F} = \left\{ t^{(1)}, ..., t^{(i)} \right\}, \quad i = 1, 2, 3... \tag{5.1}$$

Where $\mathcal{F}$ is a set containing all spike arrival times and $t^{(i)}$ denotes the i^{th} action potential fired. Thus we can express the resulting spike train as

$$\eta(t) = \sum_{i=1}^{N} \eta_i(t), \quad \begin{matrix} i = 1, 2, 3... \\ N \leq +\infty \end{matrix} \tag{5.2}$$

Where $\eta_i(t)$ is the i^{th} spike occurance, $\eta(t)$ a representation of a spike train as a time series and N is the number of spike occurrences during the trial.

We assume that a spike and the one preceding are statistically independent. Furthermore, Similar to Vogelstein's and his associates' method [42], our model considers merely positive spikes. For this reason, spikes are generated stochastically with a Poissonian generator. Such a generator provides with discretely distributed nonnegative integer spikes. The practical generation of spikes will be made in MATLAB[1].

$$\eta_i(t) = \begin{cases} \mathbb{Z}^+, & t = t^{(i)} \\ 0, & otherwise \end{cases} \tag{5.3}$$

Where $\mathbb{Z}^+$ denotes positive integers.

The firing rate $r(t)$ is another parameter that needs to be assumed before setting up the spike generation process. Throughout this thesis, based on experimental data, we assume a fixed firing rate r held constant at 0.3 Hz.

Since experimental data is recorded with 10 Hz temporal resolution, the sampling time step dt in our model is also held at 0.1 s.

Trial duration for signal generation is limited to 200 s (Experimental traces are a bit longer than 220 s, for instance the trace in Figure 3.5).

[1] MATLAB scripts are included in the Appendix.

Now, we space the duration for signal generation (200 s) by the sampling rate (0.1 s). Doing so gives us a time vector that contains 2001 elements. We call this vector $[0 : dt : 200]$ as a *time vector* of a full signal generation.

In Figure 5.1 a typical Poissonian spike train generated in MATLAB is presented. According to the figure, it is less likely to detect more than one spike occurrences at a time.

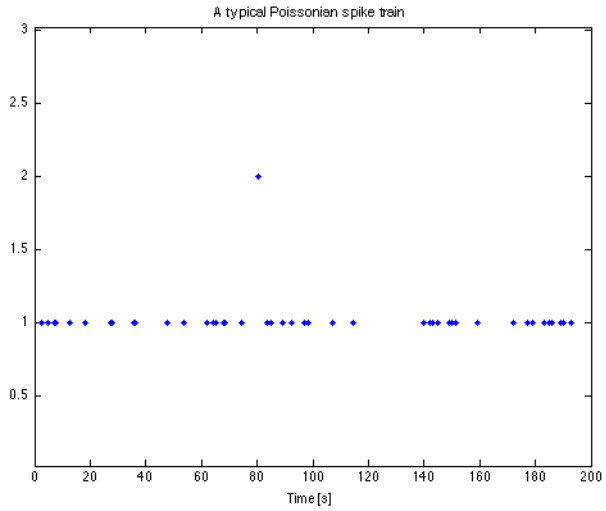


Figure 5.1: *A typical Poissonian train. This train will be later on used to generate the entire model calcium data.*

5.1.2 Kernel generation

Earlier groups who have worked with calcium imaging signals have indicated the capability of an action potential to produce a unitary calcium transient with a sharp rise followed by an exponential tailing [44, 35]. Today, the fact that excess intracellular calcium (resulting from an action potential firing) being cleared out from a neuron in an exponential manner is well established in the scientific community [12, 42]. In the coming paragraphs, we present how we model and generate exponentially decaying calcium kernels that mimic these phenomena.

In mathematical terms we model the mechanism by the following formula:

$$\kappa(t) = h_0[\exp(\frac{-t}{\tau_{c_0}})] \tag{5.4}$$

Where κ is a fixed (known) kernel, τ_{c_0} is a fixed intracellular calcium clearance time constant, and h_0 denotes a fixed kernel amplitude.

Through assumption made based on experimental data set τ_{c_0} and h_0 are held fixed at 0.5 s and 0.044 respectively. In fact, 0.5 s is in accordance with previous studies made in the field, for instance, Göbel and Helmchen have observed $\tau_c = 0.5 - 1$ s while studying a neuronal somata [12].

We also assume that a typical neuron uses three seconds to clear out all the excess somatic calcium ion that has resulted from an action potential firing. In our model, 3 s means $6\tau_{c_0}$. Subsequently, the kernel will be truncated to 31 sampling time steps.

5.1.3 Output signal generation

In our model it is assumed that the level of calcium shots upwards at an instant a spike is detected. To model the phenomena, we combining the signals we have worked on in section 5.1.1 and 5.1.2

Customarily, convolution is performed as a polynomial multiplication in Fourier domain [25]. This method of multiplying in the frequency domain and inverse transforming the result back to the time domain arises from the *convolution theorem*. Nevertheless, the standard method of convolution is less convenient for our purpose. This is because later on when we start to vary kernel shapes, this method will not provide the flexibility that we may require. Rather, we develop a method that performs convolution by scaling and injecting a kernel at an instant of spike detection. Thus the convolution for a fixed kernel due to the i^{th} spike can be written as

$$\psi_i(t) = (\kappa * \eta_i)(t) \tag{5.5}$$

where $\psi_i(t)$ is the convolved calcium output signal due to the i^{th} spike, κ denotes a fixed kernel with τ_{c_0} and h_0, whereas η_i represents the i^{th} spike.

As an example, we look at the convolution occurring due to the second spike $\eta_2(t)$

$$\psi_2(t) = (\kappa * \eta_2)(t) = \sum_{t'=0}^{\infty} \kappa(t')\eta_2(t - t') \tag{5.6}$$

where $t - t' = t^{(2)}$, i.e. the arrival time of the second spike. The above equation can also simply rewritten in terms of the second spike

$$\psi_2(t) = \eta_2(t)[\kappa(t - t^{(2)})] \tag{5.7}$$

where $\eta_2(t)$ is the argument of the second spike.

Typically, a neuron will fire several action potentials in a trial duration (200 s). Thus we extend the convolution made for the second spike in Equation 5.6 to include all incoming spikes in the trial duration.

$$\sum_{i=1}^{N} \psi_i(t) = \sum_{i=1}^{N} \sum_{t'=0}^{\infty} (\kappa[t']\eta_i[t - t']) \tag{5.8}$$

Where N is number of detected spikes in a trial and the left side of the equality sign is the output signal. Furthermore, when it is appropriate, Equation 5.8 can be simplified in terms of the unitary spike

$$\sum_{i=1}^{N} \psi_i(t) = \sum_{i=1}^{N} [c_i(\kappa_i[t - t^{(i)}])] \tag{5.9}$$

where c_i is the i^{th} scaling constant or simply the i^{th} spike argument. The way the output signal is expressed in Equation 5.8 and 5.9 is cumbersome. Thus for simplicity we drop the summation sign and index i

$$\psi(t) = \sum_{i=1}^{N} \psi_i(t) \tag{5.10}$$

where $\psi(t)$ is the calcium output signal constructed based on contributions from all detected spikes ($\eta_i(t)$) in a given trial.

So far the output trace $\psi(t)$ has been modeled only for a fixed kernel amplitude and constant clearance time constant. Now, we consider the fixed τ_{c_0} and h_0 as the mean parameters of the known kernel. Thereafter, we extend the fixed mean values to include other parameter values around them. We denote the new parameter extensions by τ_c and h. Making such an extension will enable us to produce different combinations of output signals made-up of varying hs and τ_cs with common family mean parameters (h_0 and τ_{c_0}).

Extensions are made as small increments to the left and the right of the mean values. We denote the increments with Δh and $\Delta\tau_c$ to the mean amplitude and the mean clearance time constant respectively. Thus we obtain a range of values where we can choose out candidate parameters from.

$$\begin{aligned} h &= h_0 + \Delta h \quad where \;\; |\Delta h| \leq dh \\ \tau_c &= \tau_{c_0} + \Delta\tau_c \quad where \;\; |\Delta\tau_c| \leq d\tau_c \end{aligned} \tag{5.11}$$

Where h_0 and τ_{c_0} are 0.044 and $0.5s$ respectively as indicated earlier and where

$$dh = \{0, 0.010, 0.015, 0.020\}$$
$$d\tau_c = \{0, 0.10, 0.20\}$$

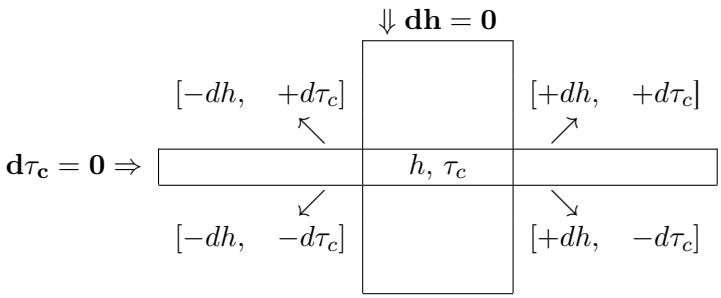


Table 5.1: *Possible parameter combinations for calcium signal generation (before noise introduction).*

Note that a zero increment is another way of saying a kernel parameter is just the one given by the mean.

The candidate parameters (h and τ_c) are uniformly distributed within a given interval (Equation 5.11). Therefore, it is equally likely for any value to be chosen by the parameter selection algorithm as long as it is within a specific interval. Therefore, when the parameter selection algorithm loops through the different intervals (demonstrated in Table 5.1), it will have the freedom to make various combinations of h and τ_c by selecting them out randomly from the interval at disposal.

As we will show later on , calcium output traces generated in this fashion (with unknown kernel parameters within a known interval range) will play a crucial role when we investigate the effectiveness of our spike and kernel estimation algorithms.

In MATLAB, we inject a kernel when we detect any incoming spike. As an example, a segment of a typical output calcium signal generated with varying kernels is illustrated Figure 5.2.

5.1.4 Adding Noise

The following paragraphs present how the final model signal of calcium imaging data is prepared from $\psi(t)$.

We prepare the model calcium output signals by corrupting $\psi(t)$ with *uncorrelated* additive noise. Uncorrelated in a sense that knowing one noise sample does

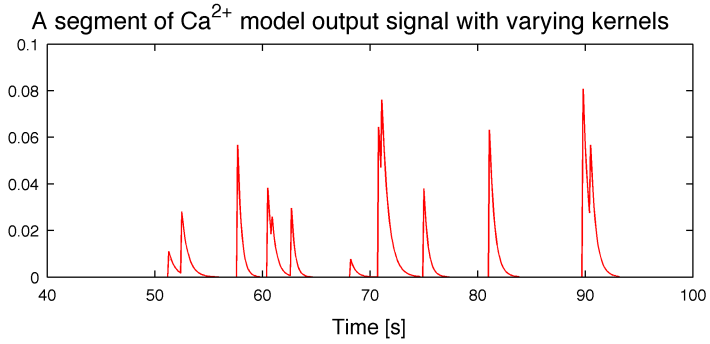


Figure 5.2: *A segment of a typical output signal (without noise).*

not help to predict other noise samples [37]. We do so, by adding a *white Gaussian noise* of *unit variance* and *zero mean*, which we simply denote by ϵ. MATLAB has integrated functions that can generate white noise (for instance *randn* and *awgn*).

However, the amount of noise encountered in experimental data set is not well known. In fact, the imaging instrument employed for experiment does not specify the signal to noise ratio (SNR) of the recorded signals. Therefore, it is plausible to have a varying degree of SNR in our models. For this reason, we corrupt $\psi(t)$ with varying amount of noise

$$\xi = \{1, 2, 3, 4, 8, 16, 32\} \tag{5.12}$$

where ξ denotes the SNR. In this way, values chosen from ξ will decide the amount of noise corrupting $\psi(t)$. In general, choosing $\xi = 1$ means a signal is corrupted by very powerful noise whereas the power of corrupting noise is minimal when the chosen $\xi = 32$.

In most cases, what is considered as a noise in experimental data has displayed slow drift. Similarly, we prepare additive noise with varying drift level. For this reason, additive noise will be low-pass filtered by the *Moving Average Smoothing* (MAS). Smoothing is done through replacing each noise data point with average of the neighboring noise data, within a certain *span*

or *filter width* [40]. This way, the MAS will smooth out short-term fluctuations with a moving window while preserving long-term trends. Hence, an array of raw additive noise signal ϵ data $[\epsilon(t_1), \epsilon(t_2), \ldots, \epsilon(t_N)]$ will be transformed to a new smoothed ϵ_s array

$$\epsilon_s(t_j) = (2s-1)^{-1}[\epsilon(t_j+s) + \epsilon(t_j+s-1) + ...\epsilon(t_j-s)] \tag{5.13}$$

where $\epsilon_s(t_j)$ is the smoothed noise signal for the j^{th} data point, s is the number of neighboring noise data points on either side of $\epsilon_s(t_j)$, and $\omega = 2s+1$ is the filter window.

$$\omega = \{1, 2, 3, 4, 5, 7, 9, 11\} \tag{5.14}$$

Where ω contains candidate numbers of neighboring data points for MAS application. For simplicity, we denote the smoothed additive noise simply by ϵ_s.

Now, we corrupt the output signal $\psi(t)$ with a family of smoothed and scaled version of additive noise.

$$\widetilde{\psi}_s(t) = \psi(t) + \frac{h}{\xi}\epsilon_s \tag{5.15}$$

where $\widetilde{\psi}_s(t)$ is corrupted output signal, the smoothing index s is chosen from the ω, h is kernel amplitude and ξ is the chosen SNR value. In the rest of this text, we will refer to the corrupted output signal $\widetilde{\psi}_s(t)$ as the modeled calcium signal.

In Figure 5.3, we have illustrated a segment of a typical calcium signal (varying kernel) with added noise.

Finally, we will generate a total of 672 model signals by selecting out parameters from h, τ_c, ξ and ω ranges in different sequential steps and combinations. Each of the 672 model signals will be stored labeledwith their respective h, τ_c, ξ and ω parameters.

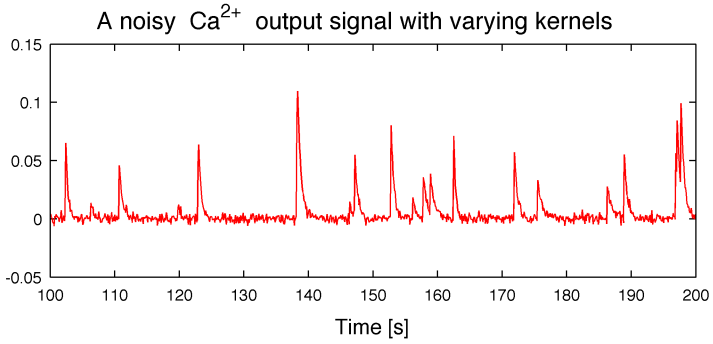


Figure 5.3: *A segment of a noisy typical Ca^{2+} signal generated with varying kernels.*

5.2 Estimating kernels from data

5.2.1 Kernel amplitude

In this subsection we show how kernel amplitudes are estimated from a calcium signal $\widetilde{\psi}_s(t)$, which can be either model or experimental signal.

We employ a peak detection strategy to estimate kernel amplitudes. Peak occurrences, in both experimental and model signals, are sporadic. In most signals variable peak shapes have been encountered. In fact, plateaued-peaks are not uncommon to encounter.

We start out the algorithm by differentiating the model data with respect to time, which we simply denote by $\widetilde{\psi}'_s(t)$. This is simply done by calculating differences between adjacent data points of $\widetilde{\psi}_s(t)$.

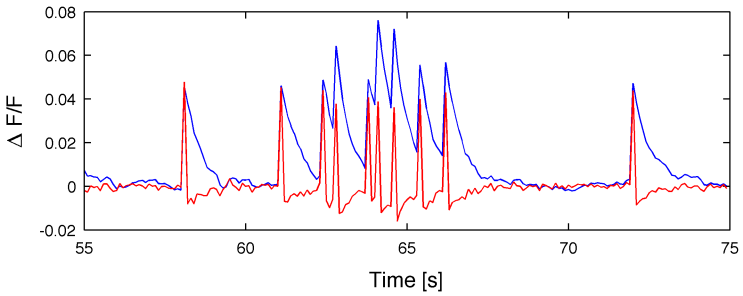
$$\widetilde{\psi}'_s(t) = [(\widetilde{\psi}_s(t_2) - \widetilde{\psi}_s(t_1)), (\widetilde{\psi}_s(t_3) - \widetilde{\psi}_s(t_2)), ..., (\widetilde{\psi}_s(t_N) - \widetilde{\psi}_s(t_{N-1}))] \quad (5.16)$$

Where N is the number of data points in the model signal.

When incoming spikes arrive almost simultaneously, their individual kernel response summates revealing wider and taller peaks. These phenomena lead to complex temporal variation with higher elevations than normally

expected. We have employed the differentiated signal for kernel height estimation in order to resolve this potential bias. In a differentiated model signal superimposed elevations are distinguished and attenuated. In other words, peak height in a differentiated signal has been observed as being inversely proportional with peak width in the non-differentiated (original) model singal.

Figure 5.4: *Differentiating reveals the underlying situations of a signal .Blue: A typical modeled signal. Red: Blue signal when differentiated.*



Next, we apply *Savitzky-Golay filter* (*S-Gf*) on $\widetilde{\psi}'_s(t)$. The *S-Gf* least square fits the differentiated model data segment by segment to a polynomial. Thereafter, central points of the fitted polynomial will be calculated and used to smooth the model data [10].

$$[\widetilde{\psi}'_s(t_j)]_{sg} = \frac{\sum_{k=-n}^{n} A_k \widetilde{\psi}'_s(t_{j+k})}{\sum_{k=-n}^{n} A_k}, \quad k = 5, 7, 9, 11, 13... \tag{5.17}$$

Where $[\widetilde{\psi}'_s(t_j)]_{sg}$ is the *S-Gf* $\widetilde{\psi}'_s(t)$, A_k is the weighting coefficient, and the odd number k defines the smoothing window size. In this case, the smoothness of $\widetilde{\psi}'_s(t_j)$ can be adjusted by varying the parameter k.

We use the maximum of $[\widetilde{\psi}'_s(t_j)]_{sg}$ to threshold the differentiated model data $\widetilde{\psi}'_s(t)$.

$$[\widetilde{\psi}'_s(t)]_{th} = [\widetilde{\psi}'_s(t) - \max([\widetilde{\psi}'_s(t_j)]_{sg})] \tag{5.18}$$

Where $[\widetilde{\psi}'_s(t)]_{th}$ is the threshold amplitude of the differentiated calcium signal.

Next, we compute the average of the differentiated data that lie above the amplitude threshold ($\widetilde{\psi}'_s(t) > [\widetilde{\psi}'_s(t)]_{th}$). The calculated average will be denoted by h_e and represent the estimated height of a kernel.

5.2.2 Kernel clearance time constant

This subsection shows how the kernel time constant τ_c is estimated from the model data $\widetilde{\psi}_s(t)$.

For a given $\widetilde{\psi}_s(t)$, parts of the signal containing exponentially decaying tails specified by a minimum peak height and peak separation (kernel width) will be selected. Specification of threshold peak height is made based on the kernel amplitude estimation made in section 5.2.1.

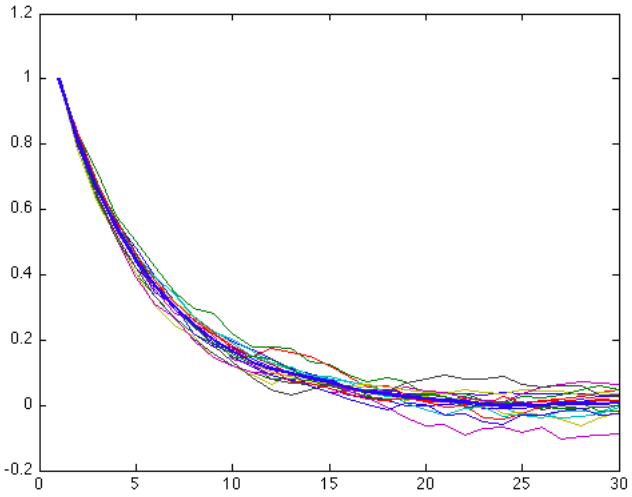
$$[h_e]_{th} = \alpha h_e \tag{5.19}$$

Where $[h_e]_{th}$ is the threshold kernel amplitude, h_e estimated average kernel height in a given trace and α is threshold scaling factor (held fixed at 0.6). In this case, only elevations that are above the threshold height are associated with spike incident. When such a peak is detected, we follow its exponential tail up to $6\tau_{c_0}$ sample steps (3 s). In the mean time, if another elevation satisfying the amplitude requirement is detected, the exponential kernel under investigation will be discarded. We list all the exponential kernels satisfying the two requirements and compute their average

$$\bar{\kappa}(t) = \frac{1}{N}\sum_{k=1}^{N}[\kappa_k(t)] \tag{5.20}$$

where $\bar{\kappa}(t)$ is kernel average of a given model data and $\kappa_k(t)$ is given by Equation 5.4. Next, we iteratively curve fit the non-linear (slope) data in $\bar{\kappa}(t)$. The time constant τ_c will be extracted from an exponential function that will be fitted to the averaged data of the decaying slopes.

Figure 5.5: *Decay slopes of a typical modeled signal. The average is shown in thick blue.*



As a starting point we transform the non-linear $\bar{\kappa}(t)$ to a linear model. We perform the linear transformation through modifying Eqution 5.4 and taking its logarithms on both sides.

$$log(\bar{\kappa}) = log(h_e) - \frac{t_j}{\tau_c} \tag{5.21}$$

Where $\bar{\kappa}$ is the time dependent data to be fitted, h_e is the estimated kernel height and τ_c is unknown by assumption. In the log domain, we search for the best fit to which the above log model converges. Thus if we assume $log(k_j)$ as the best fit

$$S = \sum_{j=1}^{n} [log(\kappa_j) - (log(h_e) - \frac{1}{\tau_c} t_j)]^2 \tag{5.22}$$

where S is the summed square of the residuals between data points and the fitted model. To find the coefficient that minimizes S, we differentiate the above equation with respect to the unknown parameter τ_c.

$$\frac{\partial S}{\partial \tau_c} = 2 \sum_{j=1}^{n} \frac{t_j}{\tau_c^2} [log(\kappa_j) - (log(h_e) - \frac{1}{\tau_c} t_j)] = 0 \tag{5.23}$$

If we solve the equation for τ_c and represent the solution by τ_{c_e}

$$\frac{1}{\tau_{c_e}} = \frac{\sum_{j=1}^{n} [t_j log(\kappa_j)] - \sum_{j=1}^{n} [t_j log(h_e]}{\sum_{j=1}^{n} t_j^2} \tag{5.24}$$

where τ_{c_e} is the best estimate for τ_c.

In practice, the fitting is done in MATLAB by writing a function that transforms the decay slopes (nonlinear) system into a linear (logarithmic) model and superimposing a fit on the data.

However, as pointed out in MATLAB documentation (Pitfalls in Fitting Nonlinear Models by Transforming to Linearity), linear curve fitting may be susceptible to extreme outliers in the data [40]. Therefore, we perform an iterative non-linear curve fitting of the averaged kernel in the least square sense (Levenberg - Marquardt). Unlike the linear curve fitting, it is difficult to give an explicit expression for the parameter to be estimated through the non-linear model [43]. We employ the MATLAB built-in function called $nlinfit$ to do the fitting in the time domain. Basically, in our algorithm, non-linear model iteratively refines the parameters that are estimated by the linear model.

5.3 Estimating spike trains

In this subsection we discuss our starting algorithm to estimate input spikes from a smoothed model signal $\widetilde{\psi}_s(\mathrm{t})$.

We start by developing a spike estimation algorithm for a known broadening kernel. For this purpose, we employ the kernel with amplitude h_0 and time constant τ_{c_0} (discussed in section 5.2). The algorithm will be designed to estimate one unknown input parameter (spike count) based on two known parameters (kernel and $\widetilde{\psi}_s(\mathrm{t})$).

Per definition the length of a kernel is six times the clearance time constant. Therefore, the fixed kernel employed for the algorithm will have 31 sampling time steps or $6\tau_{c_0}$. Whereas, the sparse spike η and the model signal $\widetilde{\boldsymbol{\psi}}_{\boldsymbol{s}}$ have each the length of the time vector T (2001 sampling time steps). Basically, we would like to construct a linear system of the three parameters of interest (unknown spike, known kernel and known model signal). Once a linear system is constructed, it can be solved with the MATALB *back -slash* (\) operator in the least square sense. In practice, the back-slash operator has a dimensional compatibility requirement. The operator will be forced to seek for approximate solutions as long as the two matrices under operation are of the same number of rows, whereas, if one of the matrices is square having the same number as the rows of the second matrix, the operator will be forced to seek for exact solutions [40].

To construct a kernel matrix, we replicate the kernel by flipping it several times. Thereafter, we construct a square matrix that is sparse and diagonal by placing the replication along the diagonal. We denote this matrix by $\boldsymbol{\kappa}$.[2]

Now, we express the linear system we have mentioned above in terms of the matrix $\boldsymbol{\kappa}$

$$\boldsymbol{\kappa} \cdot \eta = \widetilde{\psi}_s \tag{5.25}$$

[2]Bold faces in this text represent vectors and matrices.

where $\boldsymbol{\kappa}$ and $\widetilde{\psi}_s$ are the known diagonal kernel matrix and model signal respectively, whereas η is the unknown spike coefficients that we would like to estimate and $\cdot$ denotes the dot product.

We estimate the unknown spike coefficients in the least square sense by solving the linear system with the back-slash operator

$$\eta_e = \boldsymbol{\kappa} \backslash \widetilde{\psi}_s \tag{5.26}$$

where $\backslash$ denotes the MATLAB *back -slash* operator and η_e denotes the spike estimate. The spike estimate η_e gets the same number of columns and rows as $\widetilde{\psi}_s$ and $\boldsymbol{\kappa}$ respectively. This means, η_e will have exactly the size of the spike we have generated and employed for the convolution in section 5.2. Strictly speaking, the spike estimation algorithm developed here can be seen as an alternative method for the standard deconvolution in the Fourier domain.

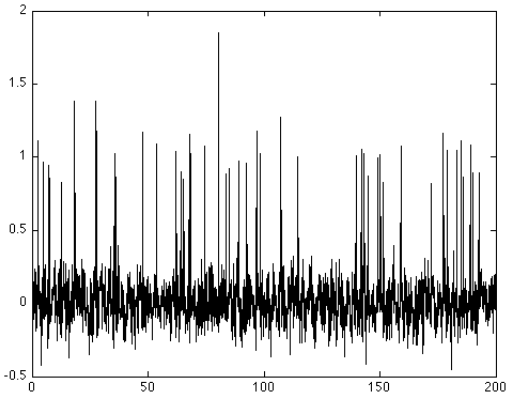


Figure 5.6: *A reconstructed spike train*

The spike argument values reported by the estimator (η_e) can be non-integer as well as negative. This is inconsistent with the spike assumption made in section 5.1.2. Therefore, all the estimation below zero will be excluded. The remaining elements of η_e will be rounded to the nearest integer.

$$\bar{\eta}_e = round(\eta_e \geq 0) \tag{5.27}$$

Where $\bar{\eta}_e$ $(= 0, 1, 2...)$ is the rounded spike estimation. In practice, the rounding can be easily performed in MATLAB. A typical example of estimated and rounded spikes are demonstrated in Figure 5.7.

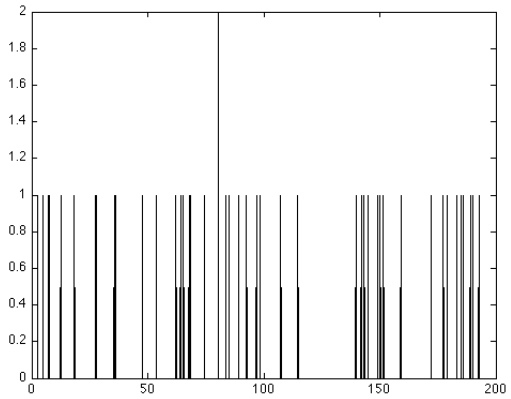


Figure 5.7: *The estimated spike train where its counts are rounded to the nearest poitive integer.*

5.4 How to supplement deconvolution

Theoretically, deconvolution is a noise sensitive algorithm [38, 40]. That means, noise might be reported as spikes by the spike estimation algorithm developed in section 5.3.

In the coming subsections we present techniques that can contribute to enhance the accuracy of the deconvolution algorithm.

5.4.1 Median Filtering

One possible way to improve the accuracy of the spike estimation algorithm is to enhance the quality of the model trace $\widetilde{\psi}_s(t)$.

We employ *median filter* to remove some of the noise characteristics in the model signal. The technique replaces center value with a median value of all points within a given window [32]. MATLAB's built-in function *medfilt*1 can do the task by applying a sliding window to a sequence of model data points [40].

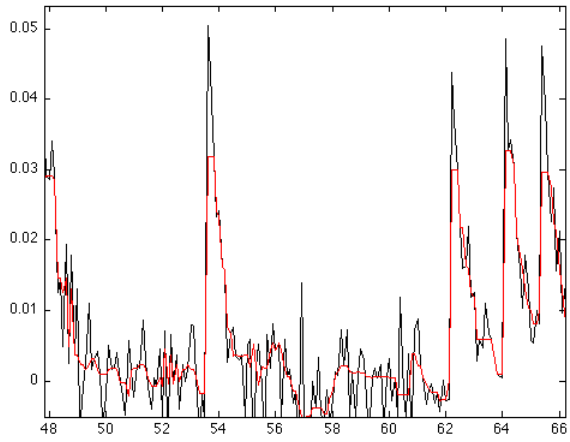


Figure 5.8: *Curve shapes are preserved by the median-filter.*

The median filter is employed as a pre-processing to deconvolution. The filter attenuates and flattens out sharp peaks observed in $\widetilde{\psi}_s(t)$ without distorting the characteristic sharp rises in model signals. Due to model signal attenuation, kernels also need some adjustment before deconvolution. To mimic the

level of attenuation in $\widetilde{\psi}_s(t)$, we adjust (truncate) kernels with median filter of the same window length as that of the one used for the model signal. A segment of a median filtered model signal is demonstrated in Figure 5.8 as an example.

5.4.2 Optimizing in the greedy approach

As an alternative to the median filter, we develop an optimization algorithm in this subsection.

The optimization algorithm starts by convolving the estimated and rounded spikes with kernel in matrix form (discussed in section 5.3).

$$\widetilde{\psi}_e(t) = \bar{\eta}_e \cdot \boldsymbol{\kappa} \tag{5.28}$$

Where $\widetilde{\psi}_e(t)$ is the estimated calcium trace, $\boldsymbol{\kappa}$ is the kernel in matrix form , $\bar{\eta}_e$ is the estimated integer spike and $\cdot$ denotes the dot product. Next, the algorithm quantifies the residual between $\widetilde{\psi}_s(t)$ (the the original signal we started with) and $\widetilde{\psi}_e(t)$

$$\gamma_i = \widetilde{\psi}_s(t = i) - \widetilde{\psi}_e(t = i) \tag{5.29}$$

where γ_i denotes the i^{th} (residual) error. The mean square error (MSE) will be calculated.

$$E[\gamma] = \frac{1}{N}\sum_{i=1}^{N}\gamma_i^2 \tag{5.30}$$

Where the MSE is denoted by $E[\gamma]$, the expected (mean) value of γ^2. The MSE will serve as a learning rule to determine whether there is a need to make a correction or not. Correction is made in the form of adding a missing spike or removing unwanted spike. In this case, if the MSE is zero, then the calcium traces coincide exactly.

We set a criterion for possibly missing or unwanted spikes, which the algorithm employ while investigating the MSE locally for individual data points.

$$\gamma_i > \theta \tag{5.31}$$

Where γ_i is the norm of the i^{th} data point MSE and θ is a threshold criterion. When the i^{th} MSE norm is larger than θ, the algorithm makes a locally optimal decision either to remove or add a spike depending on the sign of γ_i. Such an error minimization algorithm by making a locally optimal decision at the immediate iteration step is called *greedy optimization algorithm* (GOA) [13]. Upon completion of iteration for the i^{th} error, the GOA performs convolution and measures the MSE norm once again. Then it will continue performing minimization for the $(i+1)^{th}$ error if it is available.

The amount of correction that can be made by the GOA is defined by the number of allowed maximum iteration steps denoted by I . For instance, if $I = 50$, then it is possible to add or remove up to 50 spikes. In the GOA I is determined by the total number of estimated spikes (sum($\bar{\eta}_e$)) (the $\bar{\eta}_e$ estimated in section 5.3). Doing so enables to double-check each spike estimated by the deconvolution algorithm for its accuracy. However, the GOA may converge before reaching its maximum iteration step if optimal value (convergence) is arrived. The GOA assumes that convergence is achieved when it is no longer possible to lower the MSE. Finally, we denote the optimized calcium model signal by $[\widetilde{\psi}_e(t)]_{opt}$.

6 Results: Modeled data

In this section, we apply the algorithms designed in section 5 to model calcium imaging data and analyze results.

As test data, we have generated a family of 672 model signals through the procedures discussed in section 5.1. All model traces will be generated from a single known Poissonian spike train that is constructed with different h, τ_c, ω and ξ parameters. The different combinations of the four parameters will serve as indexes of model signals when we present analyses of the modeled data. In this case, the true (Poissonian) spike train as well as h_0 and τ_{c_0} (see Equation 5.11) will serve as controls when the accuracy of algorithms to estimate kernels as well as reconstruct spike trains and firing rates is measured.

In section 6.1 the efficiency of kernel estimation algorithms will be tested on the modeled data. While doing so, we fit for some threshold parameters that are necessary to know when the procedure is applied on experimental data later in section 7.3.

Based on the estimated kernels, algorithms will be asked in section 6.2 to reconstruct spike trains and fit for firing rates. There, the different algorithms designed for spike train reconstruction in section 5.3 and 5.4 will be compared and contrasted.

6.1 How well can we estimate a kernel?

As it is pointed out in section 5.2, we would like to estimate both the average amplitude and the clearance time constant for kernels of a calcium signal. First we investigate how well we can estimate the kernel amplitude h_e in section 6.1.1. In section 6.1.3, clearance time constant τ_{c_e} will be extracted from a fitted curve through a linear transformation as well as an iterative nonlinear least square model.

6.1.1 h_e and its error term when threshold is imposed based on data

To recapitulate, we have estimated kernel amplitudes by differentiating the model signal and imposing a threshold condition on the differentiated signal. As a starting point, we assume the threshold based on experimental data. This means, the threshold will be held constant for the entire 672 model signals that we have generated. Later on, the threshold will be defined via the maximum S-Gf of the differentiated signal (see Equation 5.17). Consequently, the average amplitude threshold will be potentially variable from one model data signal to another.

We start out by assuming a threshold $[\widetilde{\psi}'_s(t)]_{th}$ to be 0.01 based on experimental data. Results of amplitude estimations with this threshold for the entire model data set is illustrated in Figure 6.1.

Figure 6.1 displays the underlying four parameters (h, τ_c, ω and ξ) of a given model trace together with the respective amplitude estimation. In total, 4 x 3 panels are presented in the figure. Each panel row represents the interval from which h is selected out, as shown in Equation 5.11. In that case, the first row denotes $h = h_0$, since $|\Delta h| \leq dh(= 0)$, whereas for the 4^{th} row $|h| \leq (h_0 + 0.020)$. This means, the spread from h_0 increases when moving downwards from the 1^{st} to the 4^{th} panel row. In the same way, moving from the 1^{st} to the 3^{rd} panel column-wise will increase the spread in τ_c from τ_{c_0}, as shown in equation 5.11.

In addition, each of the 12 panels contains 8 x 7 pixels. Each pixel represents one signal. The SNR ξ (see Equation 5.12) is constant within panel columns with smallest ($\xi = 1$) and largest ($\xi = 32$) signal qualities represented by the 1^{st} and the 7^{th} column respectively. On the other hand, the MAS window size ω (see Equation 5.14) is constant within panel rows. In this case, the largest ($\omega = 1$) and smallest ($\omega = 11$) noise frequency components given at the top and bottom row respectively.

The correct value of h_e ($= 0.044$) corresponds to yellow color in Figure 6.1. Smaller amplitudes are estimated in the blue/green pixels, where the smallest h_e among them being 0.019. Larger amplitudes are estimated in the red pixels, among them the marked pixel in Figure 6.1 has given the largest h_e ($= 0.069$). The signal for this largest estimated h_e is illustrated in Figure 6.2. Almost all of the large h_e are estimated in the two top-left pixels. In those pixels both the ξ and the ω are at their lowest.

Figure 6.1: *Estimated average amplitudes for the entire model signals (=* 672*), where the imposed threshold is set to* 0.01 *based on data. In the figure, there are* 4 *x* 3 *panels, row- and column-wise increase denote spread from* h_0 *and* τ_{c_0} *respectively (see Equation 5.11). Each panel contains* 8 *x* 7 *pixels, rows and colums denote* ω *(MAS window as in Equation 5.14) and* ξ *(SNR as in Equation 5.12). According to our model, yellow pixels indicate good estimations (about* $h_0 = 0.044$*), whereas red and blue denote over and under estimated amplitudes respectively. The largest* h_e *in this case is* 0.069*, on the other hand,* $h_e = 0.0436$ *is among the best estimates. The signals behind these good and over estimation are presented in Figure 6.2 as an example.*

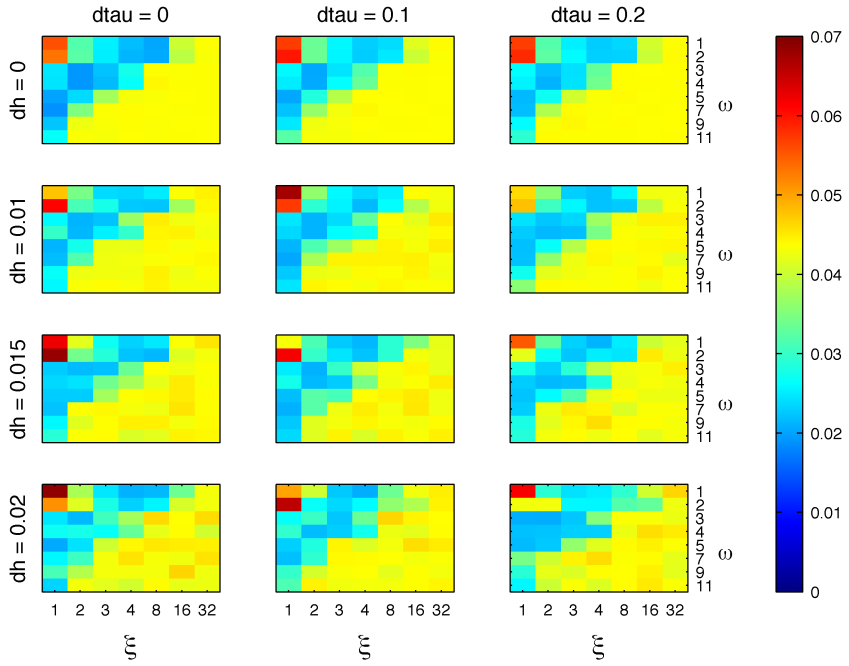
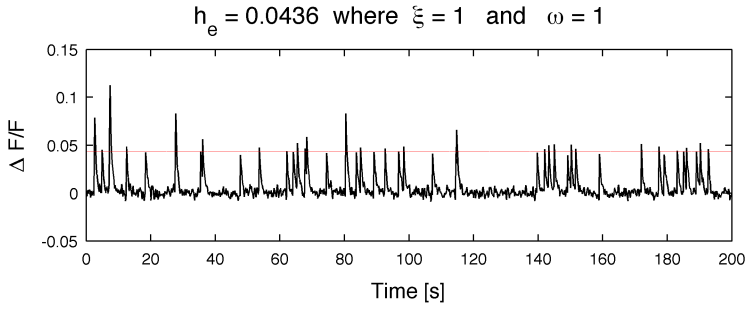
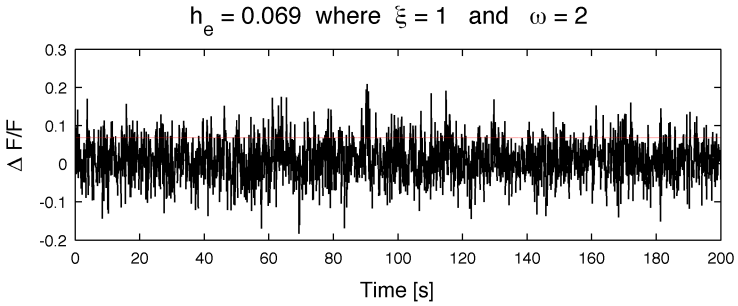


Figure 6.2: *Examples of a well- and an over-estimated average amplitude in Figure 6.1's illustration. h_e is marked by the red line. (a) Well estimated according to expectaion (h_0). $h_e = 0.0436$. In Figure 6.1, the signal is represented by $\omega = 3$ and $\xi = 8$ pixel in the 1^{st} row and 1^{st} column panel. (b) Over estimated h_e (=0.069). Representation in Figure 6.1: $\omega = 2$ and $\xi = 1$ pixel in the 3^{rd} row and 1^{st} column panel.*

(a)



(b)



To illustrate the deviation of estimated kernel amplitudes from the true height h_0, we calculate the absolute value of difference between estimated and true kernel height

$$\delta_h = |h_e - h_0| \tag{6.1}$$

wehere δ_h represents the absolute kernel amplitude estimation error, in this case, when 0.01 is imposed as a threshold.

δ_h for the entire family of model data is illustrated in Figure 6.3.

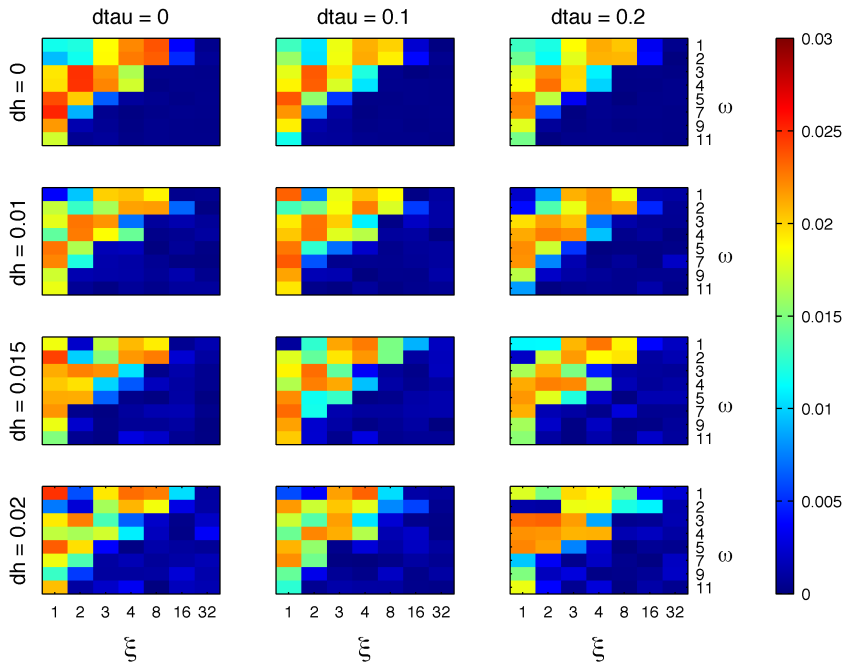
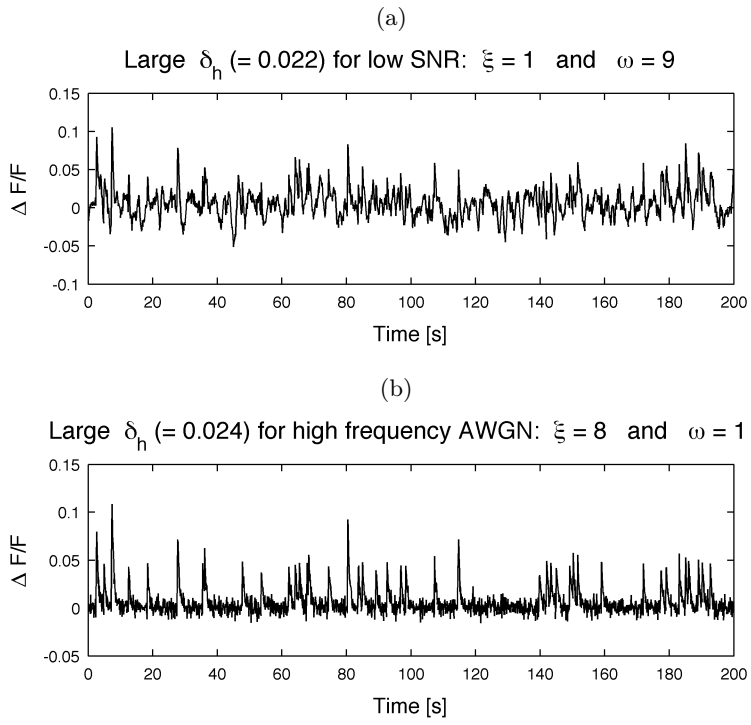


Figure 6.3: *Absolute average amplitude estimation error* δ_h *for the* 672 *model signals. Larger* δ_h *(non blue pixels) are obtained for the smaller amplitude estimations (blue/green pixels in Figure 6.1). The largest* δ_h= 0.025.

As demonstrated in Figure 6.3, higher errors are registered for the region with the smaller estimated kernel amplitudes (blue/green pixels in Figure 6.1). The largest δ_h obtained is 0.025, which means its h_e is about 56% below the true kernel amplitude h_0. The pixels with largest δ_h (red/yellow) are observed almost in all of the 12 panels stretching diagonally from lower bottom to middle top. Signals underlying the pixels with high δ_h are of either very low SNR (the outer left pixel columns) or their AWGN contains high frequency (the top pixel rows).

In Figure 6.4 underlying signals for two pixels with large δ_h that are selected out from the panel with h_0 and τ_{c_0} are demonstrated. The signal in Figure 6.4a has very low SNR ($\xi = 1$) but it is well filtered with MAS ($\omega = 9$), whereas in Figure 6.4b, the signal has relatively higher signal quality ($\xi = 8$), nevertheless, there is a high drift in the AWGN of the signal ($\omega = 1$). The two signals have comparable δ_h ($\delta_h = 0.022$ and $\delta_h = 0.024$ for Figure 6.4a and b respectively) in spite of the sufficiently large difference in their ξ and

Figure 6.4: *Examples for signals behind large δ_h. Signals are selected from two pixels in the 1^{st} row and 1^{st} column panel of Figure 6.3. (a) Low SNR ($\xi = 1$) but well smoothed AWGN ($\omega = 9$). (b) Relatively good SNR ($\xi = 8$) but high frequency AWGN ($\omega = 1$).*



ω.

If the imposed amplitude threshold $[\widetilde{\psi}'_s(t)]_{th}$ is increased from 0.01 to 0.025, δ_h will be much smaller the non-blue pixels of Figure 6.3. δ_h for the new imposed threshold (0.025) are displayed in Figure 6.5. In this case, the signals illustrated in Figure 6.4a and b get δ_h about $2 \cdot 10^{-5}$ and $7 \cdot 10^{-4}$ respectively. In addition, the maximum $\delta_h (= 0.025)$ in Figure 6.3 is reduced by an order of magnitude to $\delta_h = 0.004$ in Figure 6.5.

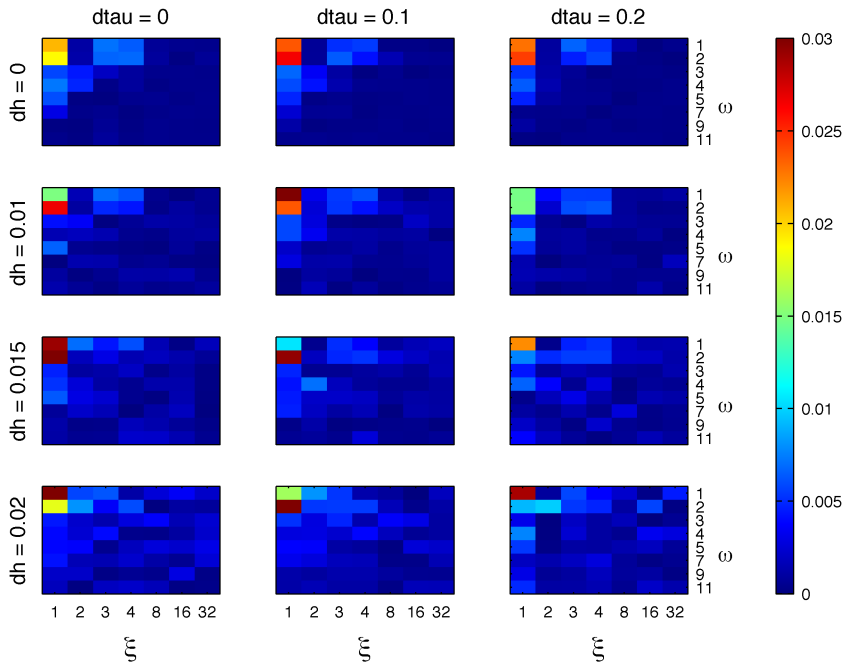


Figure 6.5: δ_h *when imposed amplitude threshold is increased to* $[\widetilde{\psi}'_s(t)]_{th}$*=0.025. Most of the non-blue pixels (large estimation errors) in Figure 6.3 are replaced with blue (small* δ_h*) pixels in this Figure.*

6.1.2 Imposed threshold via the S-Gf algorithm

Now we employ the S-Gf of the differentiated signal to decide on the kernel amplitude threshold to be imposed. In this case, it is potentially possible that each of the 672 model signals get a distinct threshold condition.

Figure 6.6 displays h_e obtained through imposing the maximum of $[\widetilde{\psi}'_s(t_j)]_{sg}$ as a threshold condintion (see Equation 5.17).

Figure 6.6 has similar properties as that of Figure 6.1 (when 0.01 was imposed as amp threshold). The yellow pixels are still the better estimates with respect to the true kernel amplitude h_0. In fact, some of the blue/green pixels in Figure 6.1 are replaced by yellow pixels in Figure 6.6. This means some smaller amplitude estimates in Figure 6.1 have been adjusted to more correct values with respect to expectation (h_0).

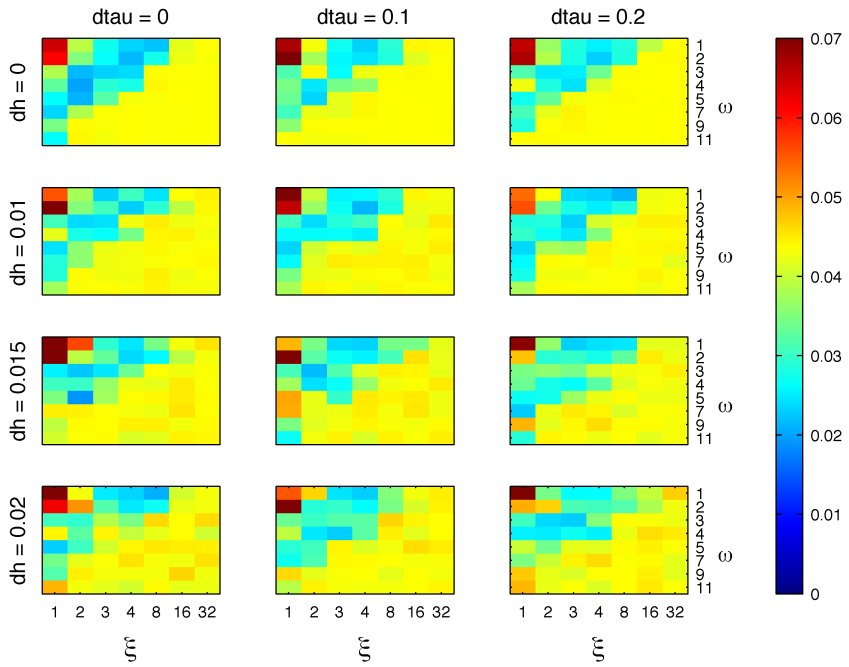


Figure 6.6: *h_e obtained when the imposed threshold is the maximum of the S-Gf differentiated signal, $[\widetilde{\psi}'_s(t_j)]_{sg}$. Yellow pixels denote well estimated amplitudes. Some of the green/blue (under estimated amplitudes) pixels of Figure 6.1 are replaced by yellow pixels in this figure.*

δ_h (Equation 6.1) for the entire data when threshold is imposed through the S-Gf procedure are displayed in Figure 6.7. In the figure, the red/yellow pixels contain the largest estimation errors than the rest. In comparison to the errors seen in Figure 6.3, imposing the S-Gf threshold condition has in particular enabled to decrease δ_h for bottom left pixels. In these pixels, the underlying signals contain very low signal quality ($\xi = 1, 2$), even though the AWGN is well smoothed ($\omega = 7, 9, 11$) . In addition, most of the red pixels seen in Figure 6.3 are replaced either by blue or yellow pixels in Figure 6.7. For instance, the percentage error of the h_e which led to the largest δ_h in Figure 6.3 is reduced by 5% in Figure 6.7. This pixel is placed in the 4^{th} row 1^{st} column panel with the pixel parameters of $\xi = 1$ and $\omega = 1$ in the figure.

In Figure 6.7 the estimation errors for the signals illustrated in Figure 6.4a and b are about $1 \cdot 10^{-2}$ and $2 \cdot 10^{-2}$ respectively. As already mentioned above, the δ_h of these signals were about $2 \cdot 10^{-2}$ and $2.4 \cdot 10^{-2}$ respectively

when the threshold was set to 0.01.

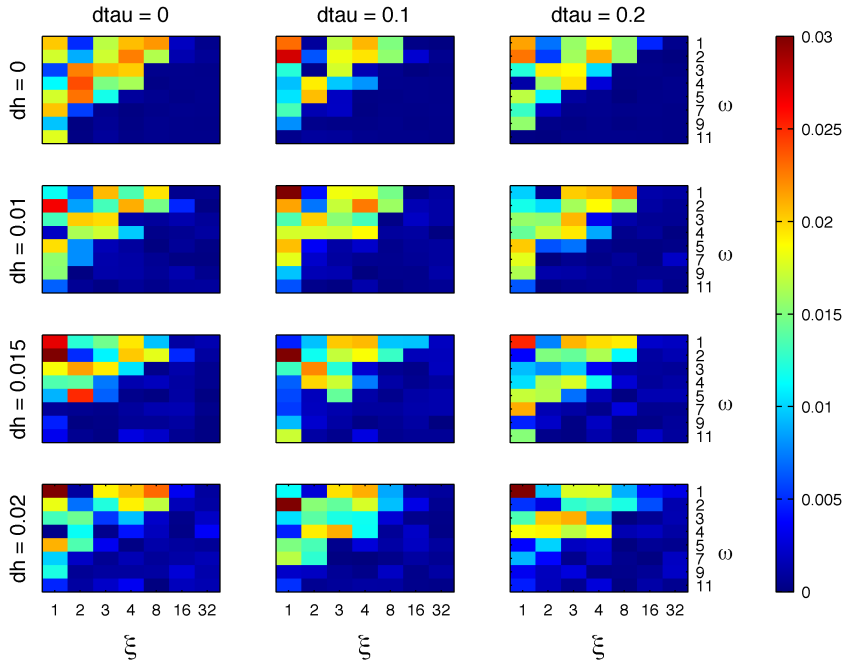


Figure 6.7: δ_h *when amplitude threshold is imposed through the maximum S-Gf procedure (Equation 5.17). Red/yellow pixels denote large estimation errors.* δ_h *has decreased in the bottom left pixels compared to Figure 6.3.*

As a conclusion to this subsection, we present a typical signal whose amplitude is estimated with a 5% error by our S-Gf based algorithm in Figure 6.8. In Figure 6.7, this signal is represented by a pixel in the 3^{rd} row/ 2^{nd} column panel with the pixel parameters of $\xi = 4$ and $\omega = 3$.

6.1.3 Kernel time constant

As pointed out in section 5.2.2, from a given calcium signal only parts that satisfy a minimum peak height and peak separation (kernel width) criteria are selected. The selected slopes will be averaged and an exponential curve will be fitted to the data. Finally, the calcium clearance time constant τ_{ce} will be extracted from the fitted curve.

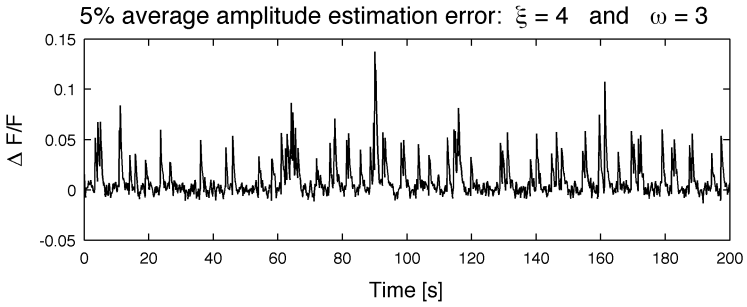


Figure 6.8: *A typical model signal whose amplitude is estimated with* 5% *error. The signal contains a relatively low signal quality and a high frequency AWGN. This signal is taken from a pixel with* $\xi = 4$ *and* $\omega = 3$ *in the* 3^{rd} *row/*2^{nd} *column panel of Figure 6.7*

The minimum peak height selection criterion makes use of the estimated average kernel amplitude, h_e. As a kernel amplitude estimator we employ the S-Gf based algorithm (see section 5.2.1). According to Equation 5.19, the minimum peak height criterion for calcium clearance time constant estimation is specified by equation 5.19 as

$$[h_e]_{th} = \alpha h_e$$

where $[h_e]_{th}$ is the minimum peak height requirement for exponential decay selection, h_e is the estimated kernel amplitude and α is the threshold scalar.

Per definition, the minimum kernel width is $6\tau_{c_0}$ as discussed in section 5.1.2.

Kernel time constants estimated from a curve fitted to data through a linear (logarithmic) model are displayed in Figure 6.9. In this case, α is held constant at 0.6. This implies that a peak is selected only if it is tall enough to reach the top 40% of h_e. According to the figure, the τ_{c_e} of signals underlying the yellow pixels are estimated with good precision, while signals underlying brown and blue pixels are over and underestimated respectively. Typically, the panels are dominated by brown or red pixels, implying signals with overestimated time constants.

The deviation of τ_{c_e} from the expected clearance time constant τ_{c_0} can be expressed with the following formula

$$\delta_{\tau_c} = |\tau_{c_e} - \tau_{c_0}| \tag{6.2}$$

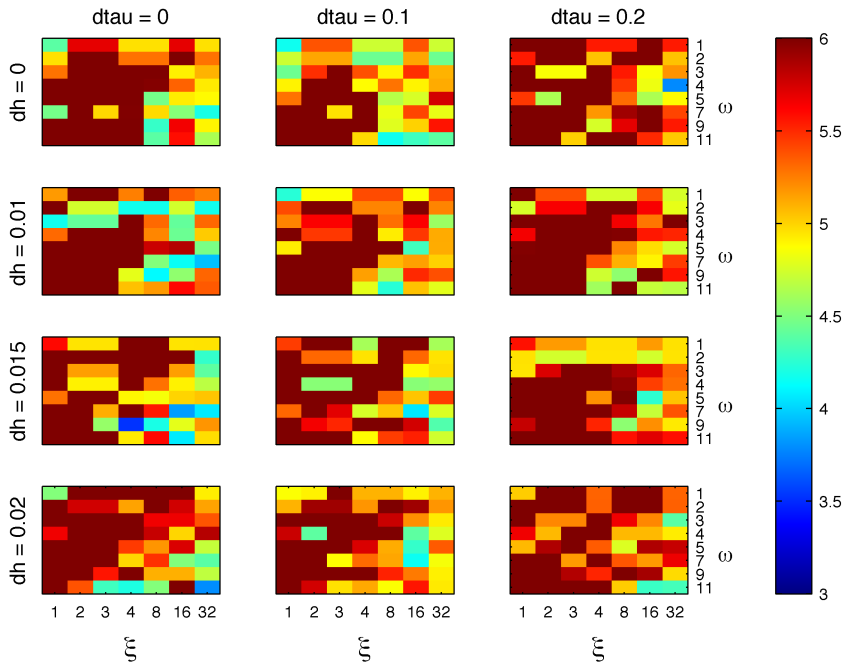


Figure 6.9: *Calcium kernel time constants estimated from a curve fitted to data through a linear (logarithmic) model, where the threshold scalar α is set to* 0.6. *Yellow pixels represent well estimated signals according to expectation (τ_{c_0}). The colorbar is scaled in time steps (for instance, 6 mean 0.6 seconds). The calcium time constants of signals underlying the yellow pixels are well estimated.*

where δ_{τ_c} denotes the absolute calcium clearance time constant estimation error.

δ_{τ_c} obtained from the linear curve fitting model ,where $\alpha = 0.6$, are displayed in Figure 6.9.

Blue pixels in Figure 6.10 indicate lower estimation error, whereas brown pixels imply relatively larger estimation error. According to the figure, most of the pixels with small δ_{τ_c} are concentrated in the three right most columns of each panel, where ξ is large. However, unlike the δ_h patterns shown in Figure 6.7, the appearance of large δ_{τ_c} are a bit more scattered or with less pattern in Figure 6.10.

Now we trace back a typical signal underlying a pixel in 1^{st} row/ 1^{st} column

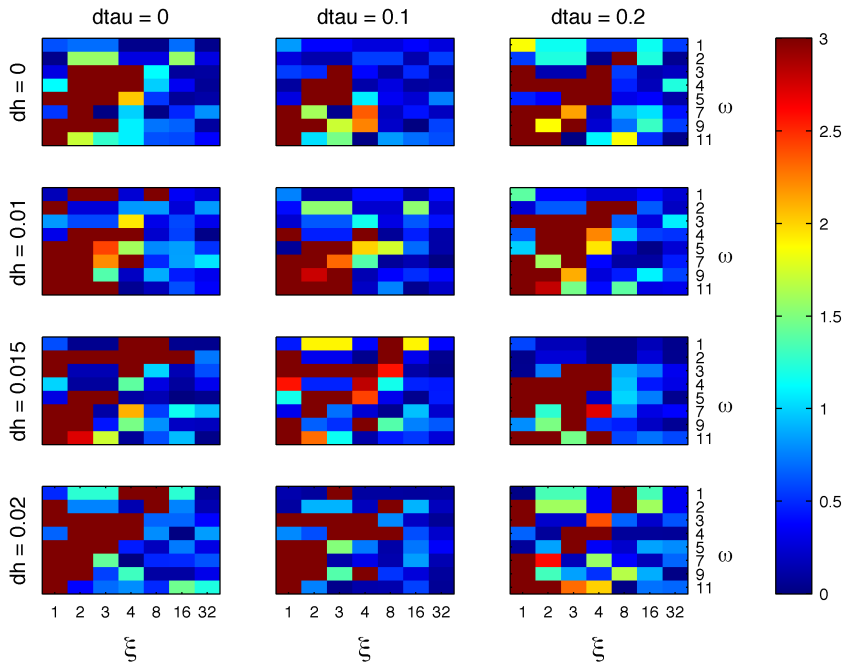


Figure 6.10: *Calcium clreance time constant estimation error (δ_{τ_c}), when the parameter is extracted from a curve fitted to data through a linear model and $\alpha = 0.6$. Blue pixels denote small errors while browns represent large δ_{τ_c}. In the high noise ($\xi = 1, 2, 3$) pixel columns, the top rows ($\omega = 1, 2$) are estimated with less errors. This is the opposite of what was seen for δ_h in Figure 6.7.*

panel with the parameters of $\xi = 8$ and $\omega = 3$. This signal is displayed in Figure 6.11. According to Figure 6.10, the signal's time constant is estimated with δ_{τ_c}=0.11 s.

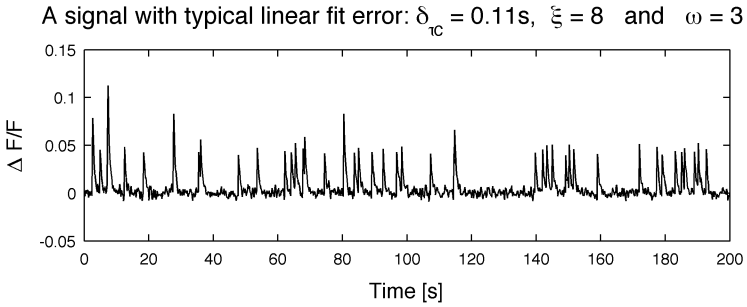
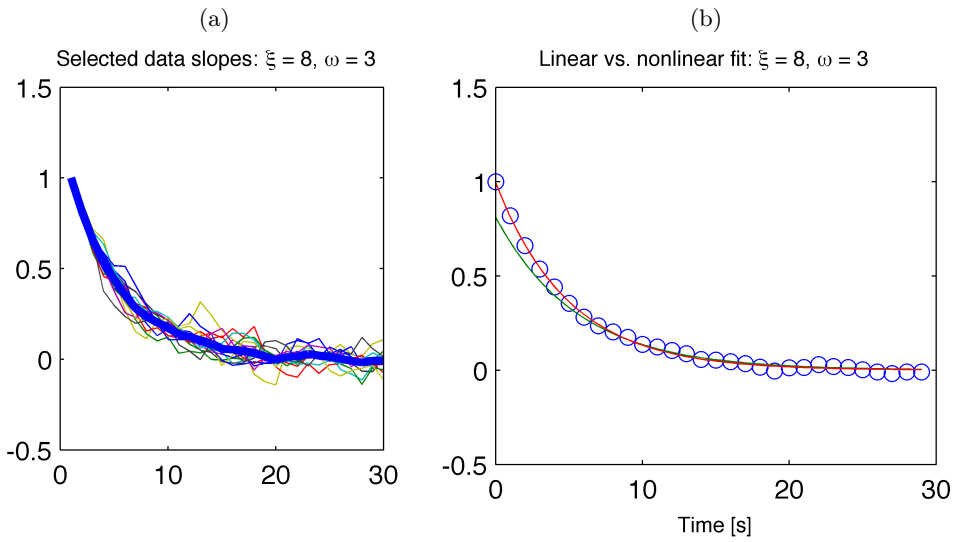


Figure 6.11: *A typical signal selected from* 1^{st}*row/* 1^{st} *column panel with pixel parameters* $\xi = 8$ *and* $\omega = 3$*. According to Figure 6.10, the signal's calcium time constant is estimated with* $\delta_{\tau_c} = 0.11s$ *from an exponential curve fitted to its data through the linear model.*

Figure 6.12: *Comparison of how the linear and the iterative nonlinear models have fitted an exponential to the averaged slope data of the signal in Figure 6.11. (a) Selected signal slopes with their average (thick blue) . (b) Averaged data (blue dots), linear fit (green), nonlinear fit (red).*



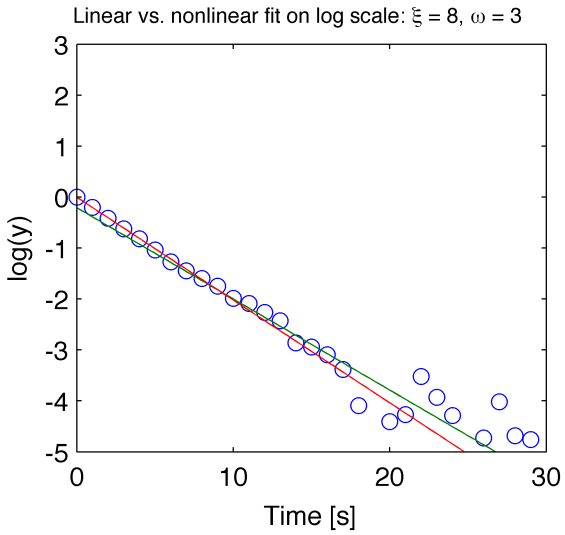


Figure 6.13: *Curves fitted to the slope average (Figure 6.12b) of the typical signal (Figure 6.11) through the linear (green) and nonlinear (red) model plotted together in logarithmic scale. The distance between data point outliers is magnified here than in Figure 6.12b.*

When a curve is fitted to the data through the linear transformation method, the signal's (Figure 6.11) $\tau_{c_e} = 0.612s$, which is 22% off (above) the expected time constant (τ_{c_0}). Let us now investigate how well the time constant of the signal could have been estimated if iterative nonlinear least square model was employed to fit a curve to signal's data instead.

First we illustrate the slopes of the signal with their average in Figure 6.12a, and in Figure 6.12b, we superimpose the linear and nonlinear fit on the averaged data and plot them together.

The linear fit model (green curve) shown in Figure 6.12b fails to follow the trend of the averaged data, but the nonlinear fit model (red curve) does follow the trend. For instance, the peak of the linear fit model is nearly 0.2 units lower than the peak averaged data point in the figure.

In order to see how data point outliers are treated by the logarithmic transformation, the linear and the nonlinear fits of the averaged slope data in Figure 6.12a are plotted together in the logarithmic scale in Figure 6.13.

The logarithmic scale plot exhibits some extreme outliers. This means the

logarithmic transformation has significantly magnified the small distances between data points that were closer to zero in the original scale [40].

In Figure 6.14, calcium kernel time constants extracted from a curve fitted to data iteratively through the nonlinear model when $\alpha = 0.6$ are presented.

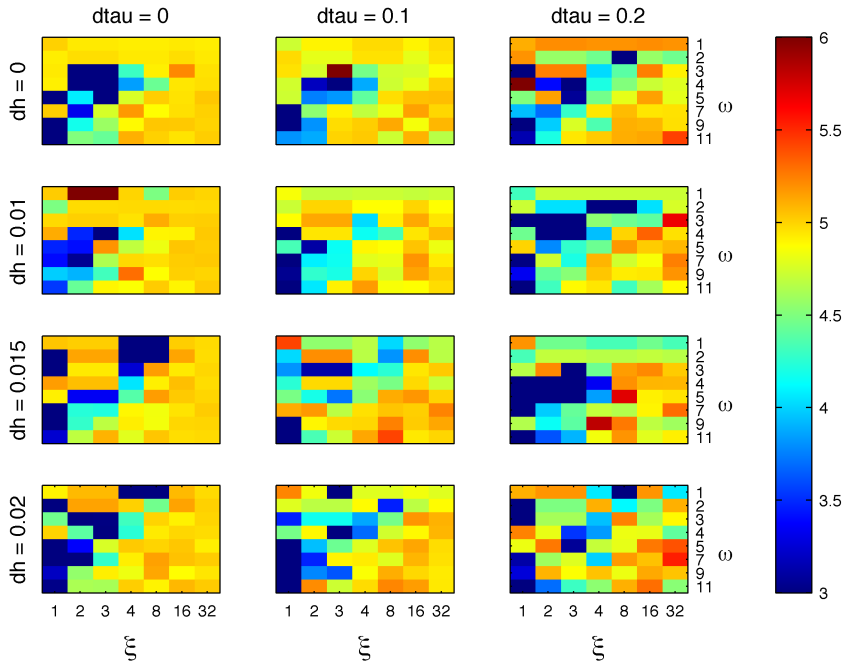


Figure 6.14: *Estimated time constants through the nonlinear model when* $\alpha = 0.6$. *Yellow pixels denote good estimates with respect to expectation* (τ_{c_0}). *Colorbar is coded in time steps (*6 *means* 0.6 *seconds). Compared to the linear estimation in Figure 6.9 are replaced by blue pixels here, implying underestimation but closer to expectation.*

Figure 6.14 is the nonlinear counterpart of the linear model estimates demonstrated in Figure 6.9. Some of the brown pixels (estimated time constants larger than τ_{c_0}) seen in Figure 6.9 are replaced by blue pixels (smaller estimates than τ_{c_0}) in Figure 6.14. Calculating the absolute estimation difference between Figure 6.9 and 6.14 will clarify how the two curve fitting models have impacted the clearance time constant estimation.

$$\delta_{fit} = |(\tau_{c_e})_{lin} - (\tau_{c_e})_{nlin}| \tag{6.3}$$

Where δ_{fit} is the absolute estimation difference between a linear and nonlinear fit models, whereas $(\tau_{c_e})_{lin}$ and $(\tau_{c_e})_{nlin}$ are estimated time constants through the linear and the nonlinear method respectively. In Figure 6.15, δ_{fit} when $\alpha = 0.6$ are illustrated.

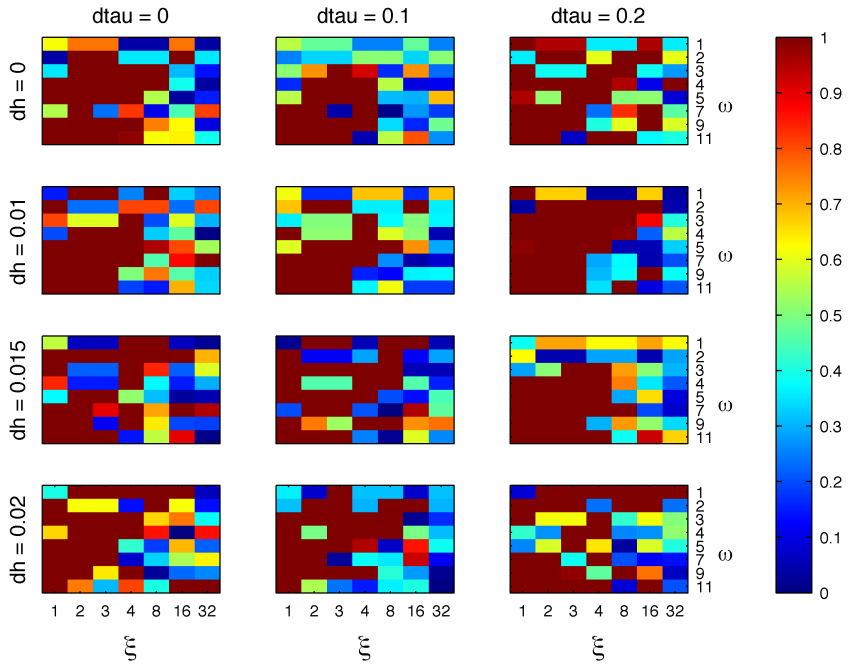


Figure 6.15: *Absolute difference showing how time constants are estimated by the linear and the nonlinear algorithms. In this case, the linear and the nonlinear models have estimated different time constants for signals underlying the non blue pixels. The largest δ_{fit} are represented by brown pixels where $\xi = 1, 2, 3, 4$ and 8.*

Figure 6.15 shows that the time constants in the non-blue pixels are estimated differently by the linear and the nonlinear model. In fact, for the brown pixels the ratio of δ_{fit} to τ_{c_0} is more than 20%. This large ratio (20%) in those pixels coincides with our earlier observation, i.e. more yellow pixels (good estimates) in Figure 6.14 for those pixels than we saw in Figure 6.9 with the linear model for the brown pixel regions of Figure 6.16.

We investigate what δ_{fit} imply, by plotting δ_{τ_c} of the nonlinear model in

Figure 6.16.

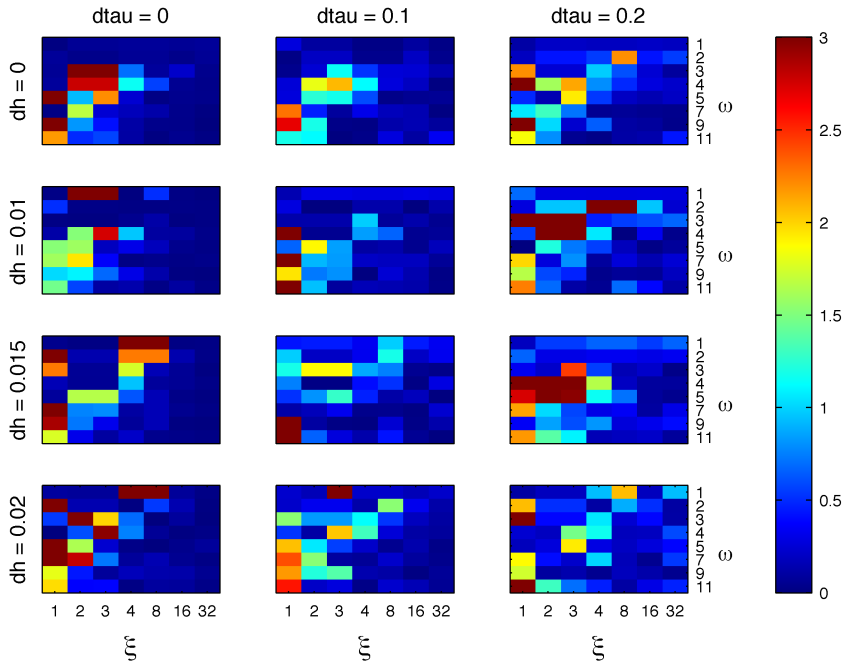


Figure 6.16: δ_{τ_c} *for clearance time constants estimated through the nonlinear model. The panels are dominated by blue pixels, implying a* 20% *or less deviation from expectation.*

Clearly, Figure 6.16 is dominated by blue pixels, implying estimation errors that are much less than 0.1 s (1 time step on the colorbar), or less than 20% of the expected time constant τ_{c_0}. For instance, the time constant of the typical signal we illustrated in Figure 6.11 is estimated with 1.6% error from τ_{c_0} through the nonlinear model, compared to a 22% error obtained through the linear model.

Next, we investigate how the minimum peak criterion impact time constant estimation. For this purpose, we estimate the time constant through the nonlinear model with different amplitude threshold scalars, α. In Figure 6.17, τ_{c_e} when $\alpha = 0.4$ are presented. Comparison of Figure 6.17 with Figure 6.14 ($\alpha = 0.6$) shows that, some of the blue and green pixels seen in the 1^{st} panel column of Figure 6.14 being replaced by yellow (good τ_{c_e}) pixels in Figure 6.17, when α is reduced from 0.6 to 0.4. However, the converse phenomenon is seen in the 3^{rd} panel columns of Figure 6.14 and Figure 6.17.

One reason for this contrary phenomenon can be the way underlying signals are generated. In that case, signals of the 1^{st} panel column are generated with zero spread from the mean clearance time constant as it is explained by Equation 5.11 and in the 3^{rd} paragraph of section 6.1.1.

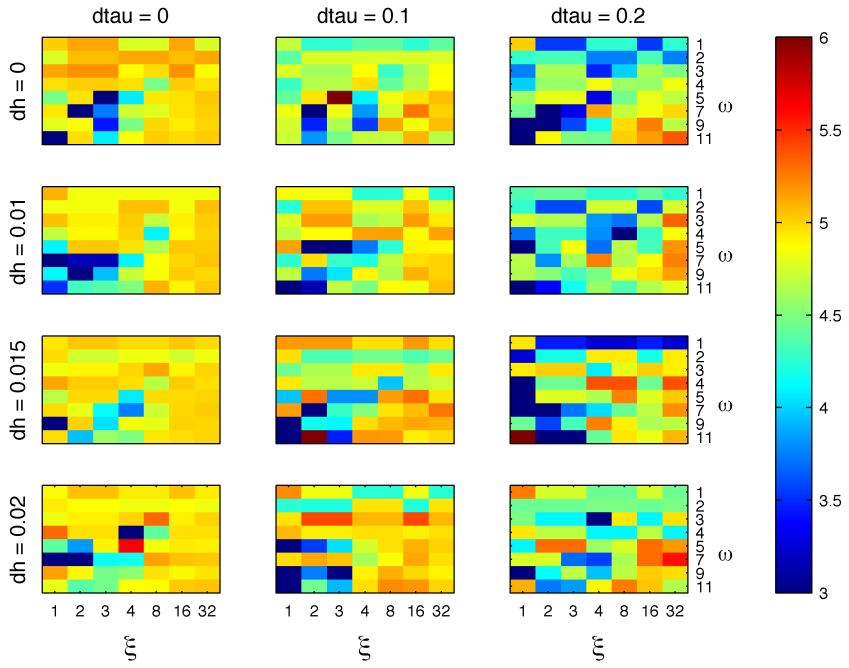


Figure 6.17: *Calcium kernel time constants estimated through the nonlinear model when* $\alpha = 0.4$. . *Yellow pixels are well estimated. More yellow pixels are seen in the first panel column due to the reason described by Equation 5.11.*

We investigate the impact of α on time constant estimation by calculating the absolute error for different α as follows

$$\delta_{\alpha_\tau} = [\,|\,(\tau_{c_e})_{\alpha_1} - \tau_{c_e}|\,-\,|\,(\tau_{c_e})_{\alpha_2} - \tau_{c_e}|\,] \tag{6.4}$$

where δ_{α_τ} is the absolute time constant estimation difference arising from unequal minimum peak criterion scalar α, whereas α_1 and α_2 are the first and the second threshold scalars respectively.

δ_{α_τ} computed when α_1 and α_2 are 0.6 and 0.4 respectively are displayed in Figure 6.18.

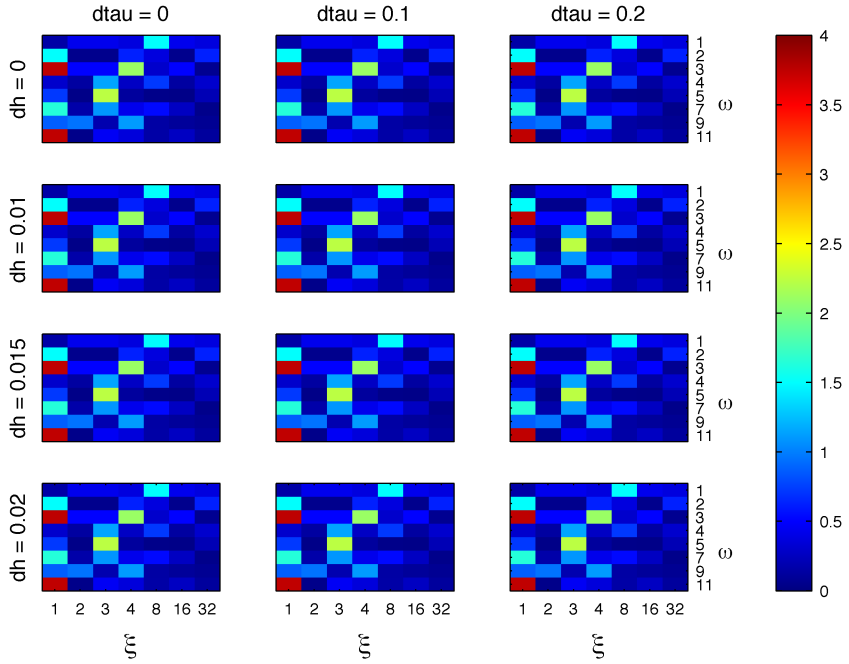


Figure 6.18: *The impact of the minimum peak criterion scalar, α, on time constant estimation, as measured by Equation 6.4. In this case, α_1 and α_2 are* 0.6 *and* 0.4 *respectively. All the panels display similar features. Two pixels of the 1^{st} column (where $\xi = 1$ and $\omega = 3, 11$) in each panel have shown the largest $\delta_{\alpha_\tau} (= 0.38$ s) due to the use of different α. In the blue pixels, δ_{α_τ} is either zero or below* 0.2 *seconds (i.e, about* 40% *of the the true time constant, τ_{c_0}).*

The display in Figure 6.18 tells us that using different α may or may not affect the calcium clearance time constant estimation. The figure has also illustrated that the effect of α is larger for signals with low quality ($\xi = 1$), despite the AWGN frequency.

6.2 How well can we reconstruct spikes?

In this subsection, spike train reconstruction algorithms (GOA, MF and GOA), designed in section 5.3 and 5.4, will be applied on model data.

In section 6.2.1, kernels that are estimated from data will be employed. For clarity, we denote estimated kernels by κ_e

$$\kappa_e = h_e[\exp(\frac{-t}{\tau_{c_e}})] \tag{6.5}$$

where h_e and τ_{c_e} are the average kernel amplitude and calcium clearance time constant that are estimated in section 6.1 respectively. Furthermore, κ_e will be written in its matrix form and denoted by $\boldsymbol{\kappa}_e$.

Employing estimated kernels could be a source of error. In order to investigate the effect of kernel estimation errors more closer, algorithms will be tested with a fixed kernel $\boldsymbol{\kappa}$ in section 6.2.4.

6.2.1 Spike train reconstruction with estimated kernels: DDA

The entire test data is generated from a single Poissonian spike train, which we have defined in Equation 5.2 in the form of

$$\eta(t) = \sum_{i=1}^{N} \eta_i(t), \quad \begin{matrix} i = 1, 2, 3... \\ N \leq +\infty \end{matrix} \tag{6.6}$$

where $\eta_i(t)$ is the i^{th} spike occurrence, N is the number of spike occurrences during a trial duration and η represents the Poissonian spike train which have generated the entire test data. The goal of the spike reconstruction algorithms is to recover the original spike train from noisy calcium traces.

To control how well we have estimated spikes, we sum up the estimated and rounded spikes overtime and compare it with the number of original spikes that have generated the model data

$$\delta_{recon} = \sum_{i=1}^{N} \eta_i(t) - \sum_{j=k}^{M} \overline{\eta}_e(t_k) \tag{6.7}$$

where δ_{recon} is the spike train reconstruction error, $\overline{\eta}_e$ is the rounded spike estimate (see Equation 5.27) and t_k denotes the timing for k^{th} estimated spike arrival. In this case, what we check by δ_{recon} is how the total number of spikes in the original and estimated spike train coincides. In other words, the equation does not control the spike arrival timings between the two spike trains. However, results (for instance, the illustration in Figure 6.21) will show that, when δ_{recon} is small or zero, the timings between original and estimated spikes coincide well with each other . On the other hand, if δ_{recon} is negative, algorithms have overestimated the number of spikes present in the model trace and vice versa for positive δ_{recon}.

In Figure 6.19, we have illustrated the spike estimation error δ_{recon} arising from the DDA for the entire family (672) of test data.

In the figure, the DDA has wrongly estimated number of spikes in the upper left (blue) pixels of each panel. The underlying signals of these badly estimated pixels contain low SNR ($\xi = 1, 2, 3, 4, 8$) and high frequency AWGN ($\omega = 1, 2, 3, 4, 5, 7$). On the other hand, δ_{recon} is either very low or zero in the red pixel regions.

Another main feature of Figure 6.19 is sharp jump from the blue (large negative δ_{recon}) to the red pixels where δ_{recon} is zero. In many cases, these jumps take place in hundreds of spikes. For instance, δ_{recon} is -94 in the blue pixel of the 2^{nd} row/ 1^{st} column panel with the parameters $\xi = 8$ and $\omega = 2$, whereas the δ_{recon} of the neighboring red pixel with equal SNR as the blue but with lower frequency AWGN ($\omega = 3$) is zero.

We will visualize this characteristic sharp jump in the DDAs spike reconstruction error with background signals of two pixels chosen from the blue and red pixel regions of the 1^{st} row/ 1^{st} column panel in Figure 6.20 and 6.21.

In Figure 6.20, the black trace on the top is a model calcium trace with $\xi = 4$ and $\omega = 1$ that we are attempting to fit with the DDA. In the second plot, the original spike train (η) that has generated the trace is shown in black, whereas the DDA reconstructed spike train ($\overline{\eta}_e$) is printed in magenta.

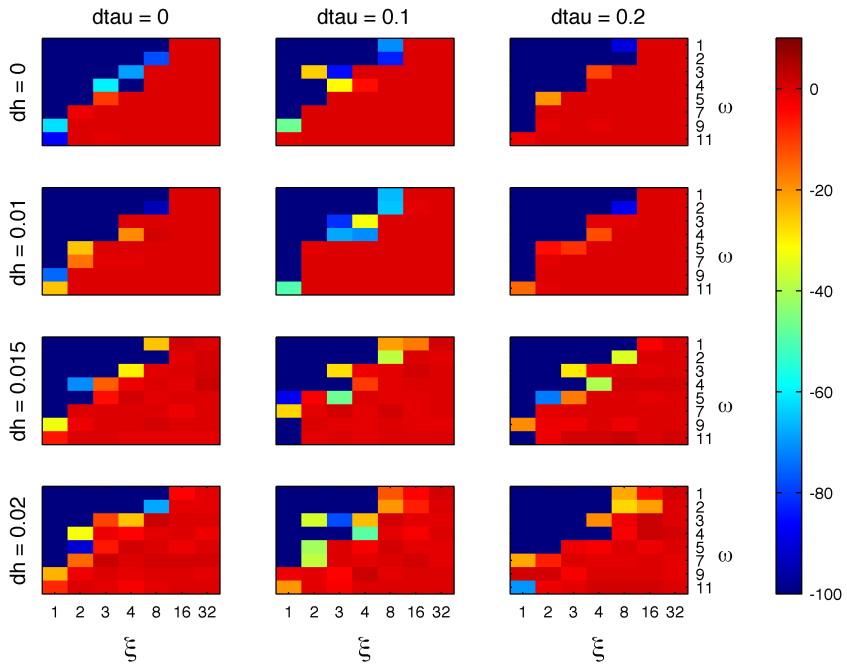


Figure 6.19: δ_{recon} *for the entire test data. Non-red pixels denote model signals whose spike numbers are overestimated by the DDA. There is a sharp jump in spike estimation error between the blue and the red pixels. The largest number of spike estimation errors are seen in the pixels with* $\xi = 1$ *and* 2*, where the DDA has overestimated spike trains by about 600.*

Deconvolution has overestimated both the number and size of the spikes present in the black trace. As a result, the deconvolution spike train contains 482 ($\delta_{recon} = -482$) false spikes. The third plot (magenta) in the figure illustrates the deconvolution fitted trace. This trace is noisier and present features that do not exist in the original trace.

Now, we may ask why do deconvolution estimate so many false spikes? The answer for this question is given in Figure 6.21.

In Figure 6.21, the original calcium trace and its deconvolution fit are shown in blue and magenta in the first and last plots of the figure respectively. The second plot of the figure illustrates the original spike train (blue) and the spike train reconstructed by deconvolution. Spikes in the two trains coincide entirely both in time and size, as it was also verified by the red

pixel ($\delta_{recon} = 0$) in Figure 6.19. In addition, the pattern, sizes and shapes of the peaks in the fitted trace (last plot of Figure 6.21) and those in the original trace match very well.

The only difference between the model traces in Figure 6.20 and Figure 6.21 is their signal quality. The SNR (ξ) of the former is 4, while the later had SNR = 16. This can mean, low SNR (for instance, 4) is not good enough for the DDA to extract spikes from a trace accurately. In such cases, deconvolution will interpret peaks in the noise factor as if they are initiated by APs and will report them. In fact, this finding demonstrates what is written about deconvolution in the literature, its extreme sensitivity to high noise [40, 38].

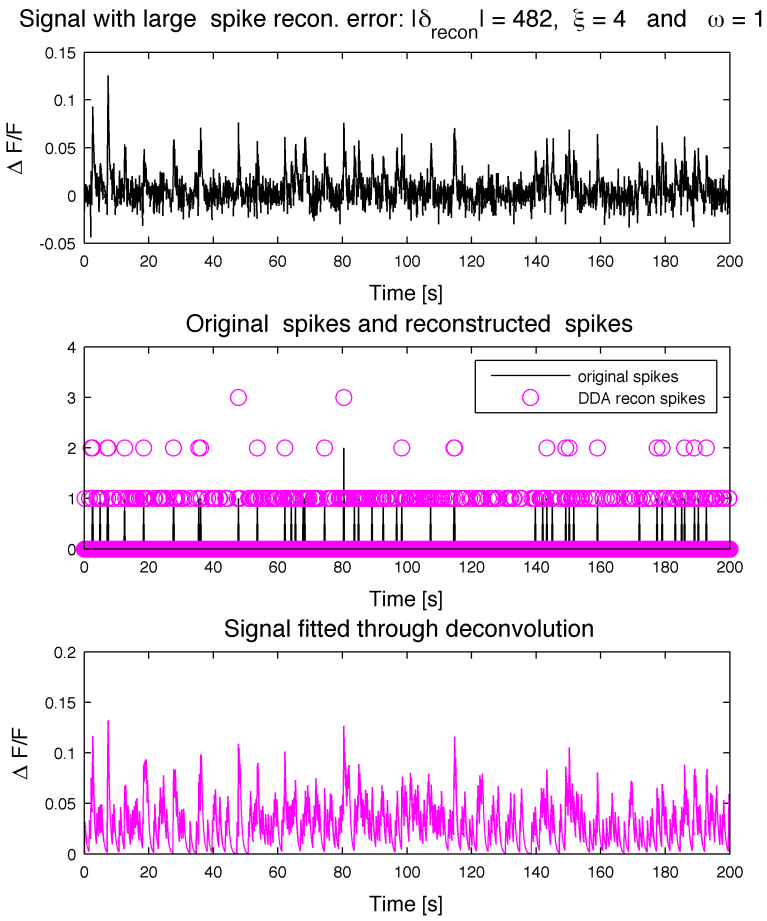


Figure 6.20: *A calcium model signal (top plot) for which the DDA has estimated 482 false spikes (magenta middle plot). The signal has the SNR of 4.*

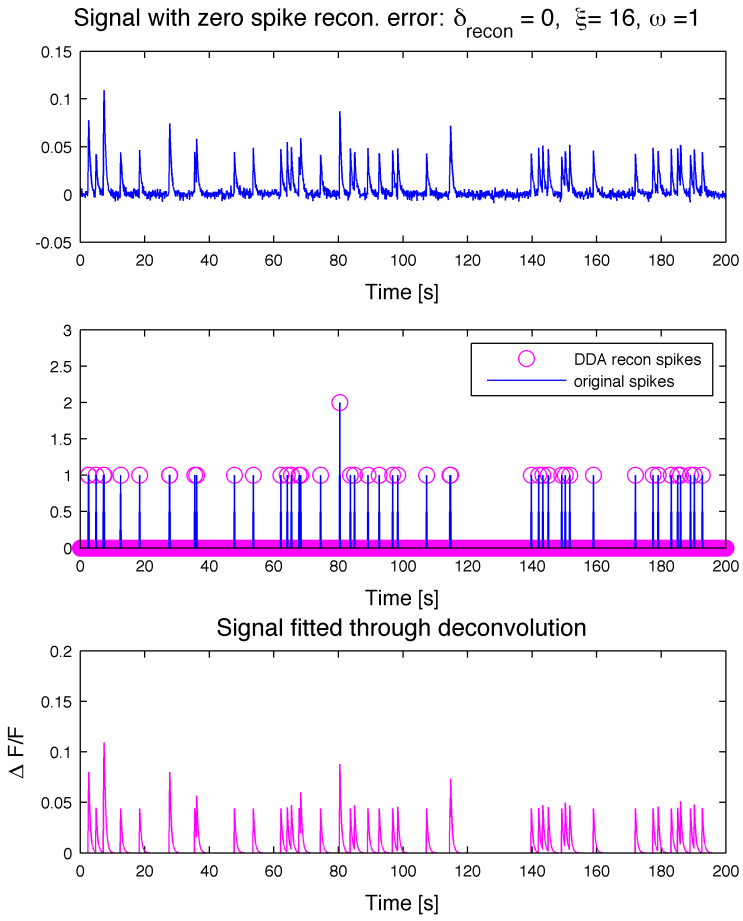


Figure 6.21: *A model signal that its spike train is fitted thoroughly by the DDA. The SNR of the signal (top plot) is 16.*

6.2.2 Median-filtering when kenel is estimated

As an attempt to improve the precision of deconvolution's spike train estimation, we apply the median-filter (MF) on the original test data $\widetilde{\psi}_s$ as a pre-processing to deconvolution (section 5.4). With the MF, we especially aim to reduce the estimation errors of deconvolution in the blue pixels ($\xi = 1, 2, 3, 4, 8$) of Figure 6.19.

We test the MF with varying sliding windows on the model data. For clarity, we denote the size of the sliding window with β. Consequently, we denote the spike reconstruction errors of MF enhanced deconvolution by $(\delta_{recon})_{\beta}$.

In Figure 6.22, we have presented $(\delta_{recon})_{\beta=11}$ for the entire test data. Comparison between Figure 6.19 and this figure shows that the MF enhancement has enabled to replace some of the dark blue pixels of low SNR in the former with light blue or other colors. This means, according to the colorbar, spike trains of signals underlying those pixels are reconstructed with lesser errors. However, there is no such a clear pattern on which of the dark blue pixels in Figure 6.19 are transformed to other colors due to the MF enhancement.

On the other hand, the red pixels with little or zero δ_{recon} in Figure 6.19 are transformed into brown pixels in Figure 6.22 due to the MF enhancement. According to the color bar and Equation 6.7, brown pixels imply at the minimum 10 missing spikes from the estimation. For instance, the δ_{recon} of the background signal illustrated in Figure 6.21 was zero before the MF enhancement, however, when the signal (top plot in the figure) is median-filtered, deconvolution has lost 10 of its spikes.

The sliding window used for the MF smoothing so far was large in size ($\beta = 11$). In order to investigate if the size of the slide window has to do with the large errors seen, we apply MF on $\widetilde{\psi}_s$ with smaller window (β =5) .

The spike reconstruction errors of MF enhanced deconvolution when $\beta = 5$ are presented in Figure 6.23.

$(\delta_{recon})_{\beta=5} \geq (\delta_{recon})_{\beta=11}$ where SNR is very low ($\xi = 1, 2, 3, 4, 8$) and vice versa where$\xi = 16, 32$; according to Figure 6.22 and Figure 6.23. For example, the spike train of the signal we illustrated in Figure 6.22 was reconstructed precisely by the DDA ($\delta_{recon} = 0$), with 10 missing spikes by 11 sliding windows MF ($(\delta_{recon})_{\beta=11} = 10$) and now 3 spikes are missed out

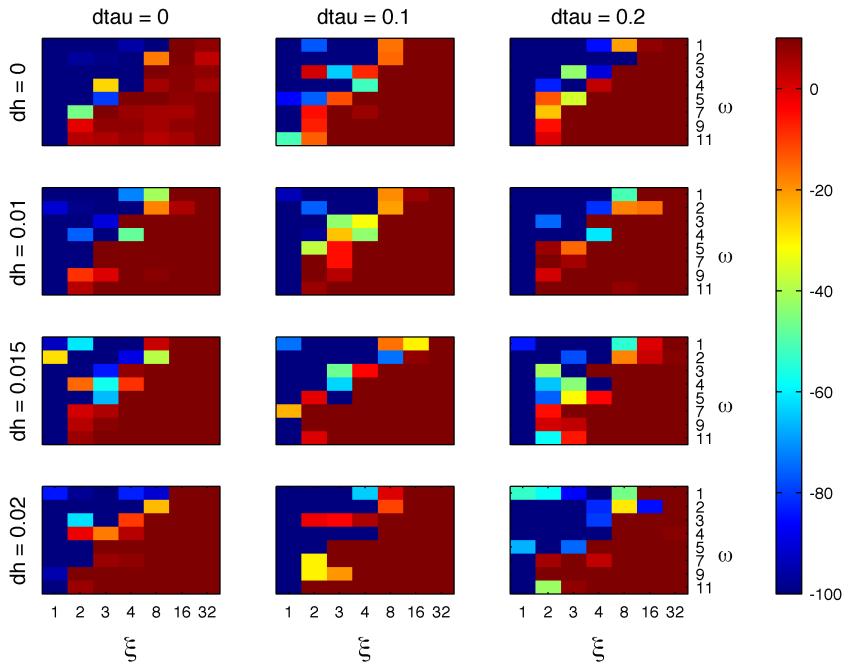


Figure 6.22: *Spike train estimation errors of MF enhanced deconvolution with* 11 *sliding windows.* $\beta = 11$ *MF enhancement has transformed some of the dark blue pixels seen in Figure 6.19 into other pixel colors of lesser* δ_{recon}*. However, the good deconvolution estimations in the red pixels of the same Figure 6.19 are deteriorated by the MF enhancement.*

by the 5 sliding windows MF enhanced deconvolution. This means, we can reduce the additional errors arising due to the MF enhancement when we decrease β, nevertheless, MF still add more errors to the already existing good deconvolution spike train estimates.

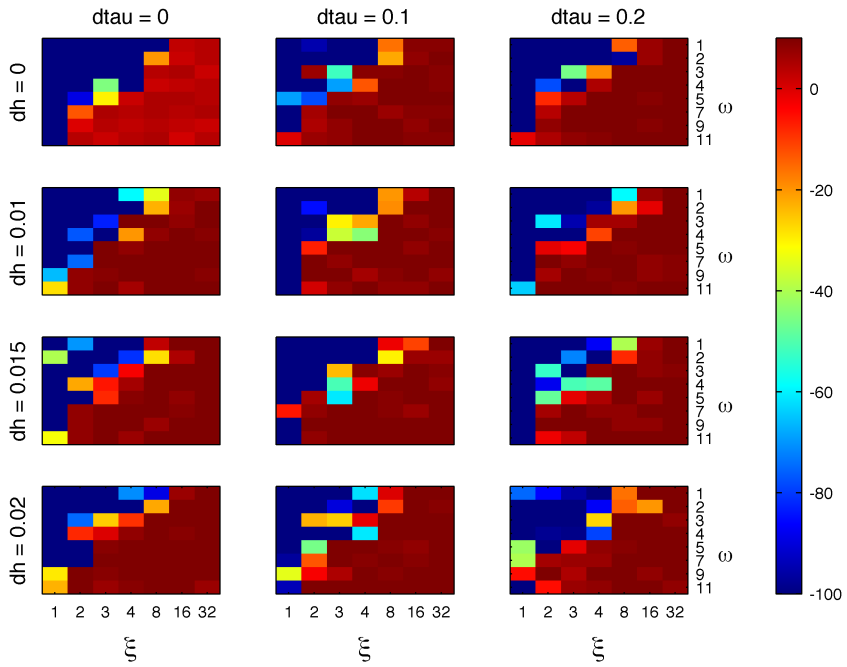


Figure 6.23: *Spike train estimation errors of MF enhanced deconvolution with* 5 *sliding windows.Compared to the display for* $\beta = 11$ *in Figure 6.22,* $\beta = 5$ *MF enhancement misses fewer spikes in the brown pixel regions. Nevertheless,* $\beta = 5$ *also still adds errors to the good spike train reconstructions that are already made by DDA (Figure 6.19).*

6.2.3 Optimizing the DDA when kernel is estimated

In the following paragraphs, we will investigate the possiblity to enhance deconvolution's estimation without deteriorating the already existing good estimations (red pixels of Figure 6.19). For this purpose we apply the GOA.

As it is described by Equation 5.31, the GOA makes the optimal decision either to add or remove a spike locally based on a comparizon made between i^{th} mean square error (MSE) and the threshold criterion θ

$$\gamma_i > \theta$$

where γ_i is the norm of the i^{th} data point MSE and where θ is defined via the estimated kernel amplitude

$$\theta = \vartheta h_e \tag{6.8}$$

where ϑ is the amplitude threshold scalar and h_e is the estimated average kernel amplitude. In addition, the maximum allowed iteration steps I for the GOA is defined from the number of spikes estimated by the DDA. For clarity, we denote the spike reconstruction errors committed by the optimized deconvolution by $(\delta_{recon})_{\vartheta}$.

In Figure 6.24, we have presented the spike reconstruction errors of $(\delta_{recon})_{\vartheta=0.2}$. In comparizon to the non optimized deconvolution in Figure 6.19, several dark blue pixels are transformed into light blue or other colors of lesser spike estimation error. This means, optimizing has enabled to remove tens of wrongly estimated spikes from the pixels where $\xi = 1, 2, 3, 4$ and 8. On the other hand, optimizing with $\vartheta = 0.2$ has missed some few spikes from the red pixels of Figure 6.19, where the DDA by itself had managed to estimate the spike numbers precisely.

In addition, the GOA has impaired the good estimations of deconvolution in the $\xi = 1$, $\omega = 9$ and 11 pixels of Figure 6.19 panels. This can mean, GOA

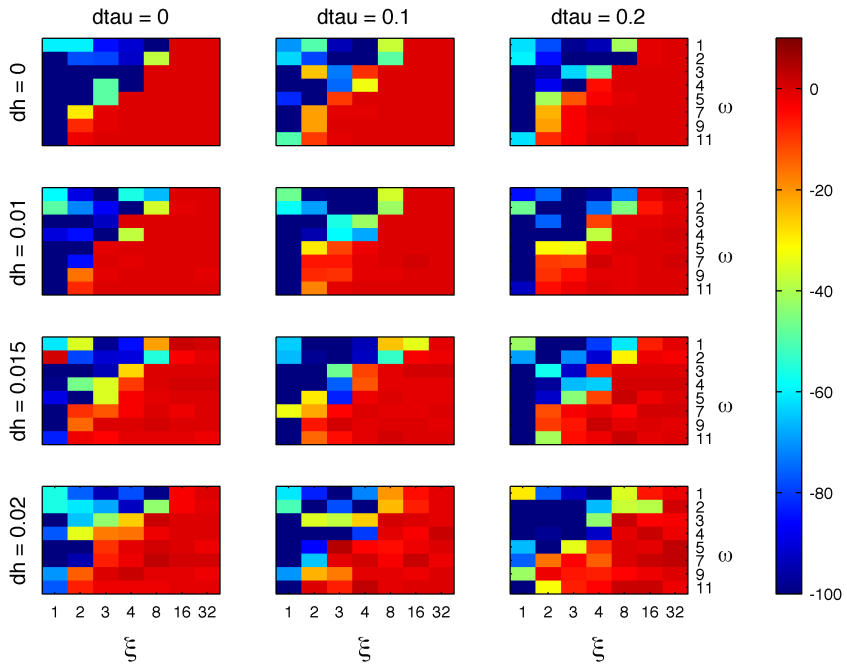


Figure 6.24: *The spike reconstruction errors of* $(\delta_{recon})_{\vartheta=0.2}$.

with $\vartheta = 0.2$ is not well suited for a signal with very low SNR containing a slow drift AWGN. In the next paragraphs, we will investigate whether using other ϑ value can avoid the shortcomings of the GOA.

In Figure 6.25, we have presented $(\delta_{recon})_{\vartheta=0.5}$ for the entire family of model data. It can be seen from the figure that the false spikes observed in figure 6.24, in the low SNR ($\xi = 1, 2, 3, 4$ and 8), as well as very low SNR and slow drift AWGN ($\xi = 1$ and $\omega = 9, 11$) pixels are reduced by about ten spikes. However, for the high frequency AWGN regions (the dark blue pixels of Figure 6.25), employing $\vartheta = 0.2$ is found to be the better alternative in removing more false spikes, as indicated by the brightness of the red pixels in Figure 6.24. This means, decreasing the amplitude threshold scalar, ϑ, will enable to remove more false spikes from our spike train estimation when it comes to noisy signals of $\xi = 1, 2, 3, 4$,8 containing high drift AWGN of $\omega = 1, 2, 3, 4, 5$ and 7.

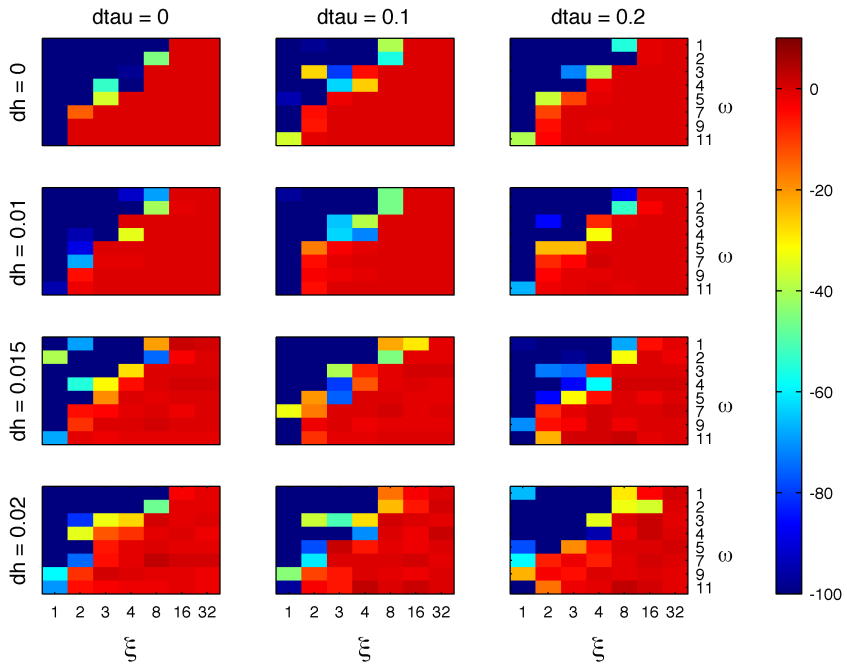


Figure 6.25: $(\delta_{recon})_{\vartheta=0.5}$ *for the entire family of model data.*

6.2.4 How better can we reconstruct spikes if the kernel is known exactly?

In this subsection, we present how well spike trains could have been reconstructed if the kernel estimation bias was avoided from the procedure. For this purpose, we employ the fixed kernel given in Equation 5.4 in its matrix form. We denote this kernel by $\boldsymbol{\kappa}$. In this case, the kernel will have h_0 and τ_{c_0} as its parameters.

In section 6.2.3 we have seen that the combination of DDA and GOA as being the best alternative among the different algorithms tested for the purpose of spike reconstruction. Consequently, in this subsection we only discuss what happens when deconvolution's estimation is optimized.

We start out by reconstructing spike trains with the deterministic deconvolution. δ_{recon} when kernel is fixed are presented in Figure 6.26. Primarily, this figure is compared with Figure 6.19 since the two are similar except for the nature of their kernels. The amount of observed red pixels is significantly higher here, compared to Figure 6.19. This means we can reconstruct spikes better when we know the kernel. In addition, if the SNR of a signal is known, then one can predict how well the spike trains will be reconstructed with the DDA.

In the same way as for the $\boldsymbol{\kappa}_e$ DDA spike train reconstruction, the blue pixels in Figure 6.26 denote pixels that contain tens of false spikes. However, unlike those in Figure 6.19, the pixels follow a characteristic pattern here. In fact, they are limited to the upper left corner of each panel, covering either the lowest SNR ($\xi = 1$) with less filtered AWGN ($\omega = 1, 2, 3, 4$) or the lower SNR ($\xi = 2, 3, 4$) with unfiltered AWGN ($\omega = 1, 2$). This means the false spikes in the first row and first column of panels arise from deconvolution, not from kernel estimation bias. In other words, deconvolution requires the SNR> 8 in order to reconstruct spike trains well, when the AWGN is unfiltered.

Let us now investigate if optimization can further improve the spike train estimation of the DDA, when kernel is known.

$(\delta_{recon})_{\vartheta=0.2}$ when kernel is known are displayed in Figure 6.27. The amount of pixels with many false spike estimates is notably reduced in this figure in comparison to Figure 6.26for the non optimized DDA. In fact, only two pixels contain many (≈ 100) false spikes in each panel. In both pixels of

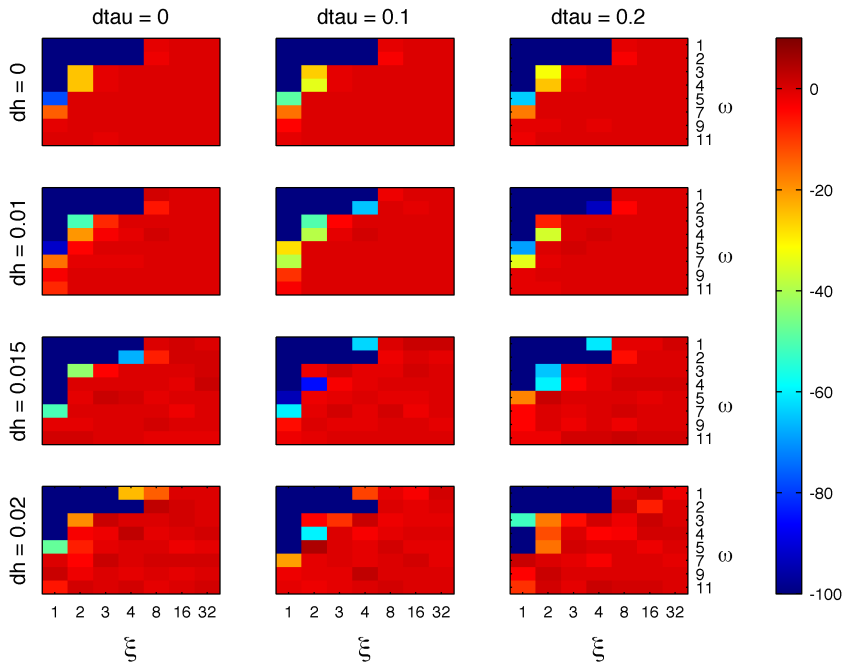


Figure 6.26: δ_{recon} *of DDA when kernel is fixed*

these types, the SNR (ξ) is one. This means, the GOA corrects the errors of the DDA by removing hundreds of spikes. For instance, about 857 false spikes are removed from the DDA's spike train estimation by the GOA for the noisiest signal with $\xi = 1$ and $\omega = 1$.

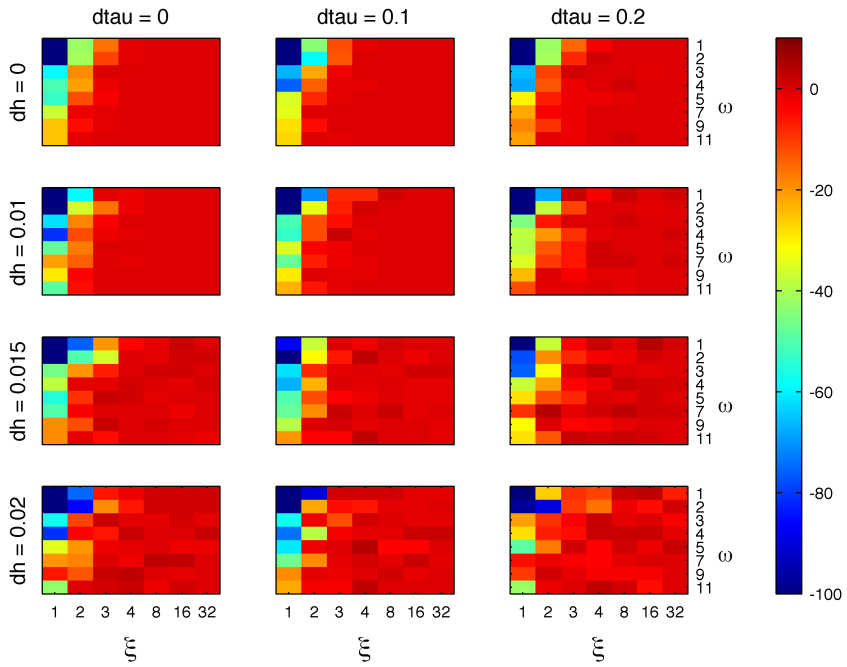


Figure 6.27: $(\delta_{recon})_{\vartheta=0.2}$ *when kernel is known.*

7 Application

7.1 Threshold parameters and the error term

7.1.1 Three threshold parameters

In our algorithms, there are three parameters (denoted by k, α and ϑ) that need careful thought.

The S-Gf smoothing window size k The algorithm for kernel amplitude estimation is developed in section 5.2.1 and tested in section 6.1. The sequence of the algorithm is as follows: First differentiate a calcium trace, filter the differentiated trace with Savitsky-Golay smoothing filter and denote the filtered signal by S-Gf, compute the maximum of the S-Gf signal (see Equation 5.17), and finally consider the average of the data above the S-Gf maximum as the average calcium kernel amplitude h_e.

The parameter k in Equation 5.17 is *critical* for the kernel amplitude estimation algorithm. For the test data in section 6.1.1, employing $k = 11$ was the best alternative. However, we need to investigate if it is reasonable to use $k = 11$ the two-photon imaging data.

The minimum peak height scalar α: In the kernel time constant estimation algorithm, modeled and tested in section 5.2.2 and 5.2.2 respectively, we use a threshold value defined as

$$[h_e]_{th} = \alpha h_e \tag{7.1}$$

where h_e is the estimated kernel amplitude, the threshold $[h_e]_{th}$ is a minimum height requirement imposed on a calcium trace to be considered as being instigated by an action potential.

α is a *critical* parameter when we estimate a kernel time constant τ_{c_e}. For the test data in section 6.1.3, we have demonstrated $\alpha = 0.6$ as the best alternative. For the two-photon data, we need to investigate if this α value is reasonable.

The optimization threshold scalar ϑ ϑ is the third *critical* parameter that relates h_e and the optimization threshold θ (see Equation 6.8 and 5.31). In section 6.2.1, we found out $\vartheta = 0.2$ being the best alternative for calcium traces with different signal quality.

7.1.2 Error term

The kernel time constant is extracted from an exponential function fitted through an iterative nonlinear fitting algorithm as shown in section 5.2.2.

We employ the standard deviation of τ_{c_e} as an error estimate, which we denote by σ_τ. We compute σ_τ from the diagonal *covariance matrix* estimated in the process of fitting the exponential function. . The MATLAB implementation of this procedure is included in the Appendix (Section 11C).

7.2 Traces of two-photon calcium imaging in vivo

7.2.1 Selected examples

Algorithms are applied on four selected two-photon calcium imaging fluorescence transient signals in vivo. Images are taken from different cortical depths in the barrel cortex.

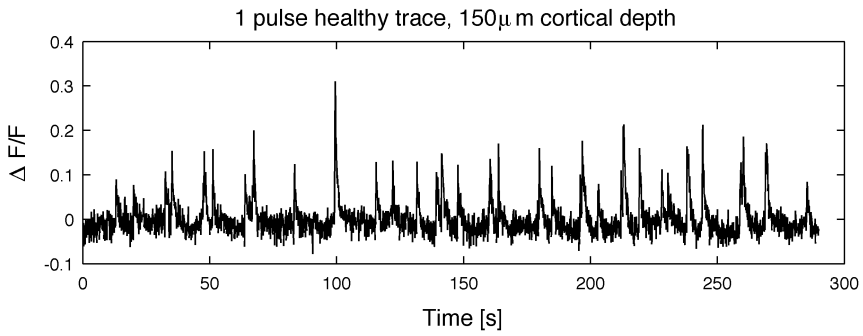
Mouse type and (Id. nr.)	Whisker stimulation	Cortical depth
wild-type (0723R2ne2)	1-pulse	$150\mu m$
transgenic (0728R15ne4)	1-pulse	$100\mu m$
wild-type (0723R4ne10)	3-pulses	$235\mu m$
transgenic (0728R8ne8)	3-pulses	$250\mu m$

Table 7.1: *Experimental calcium traces selected for algorithm application. Stimulations are given through whiskers. Mice were imaged at the Laboratory of Dr. A. Devor, University of California at San Diego.*

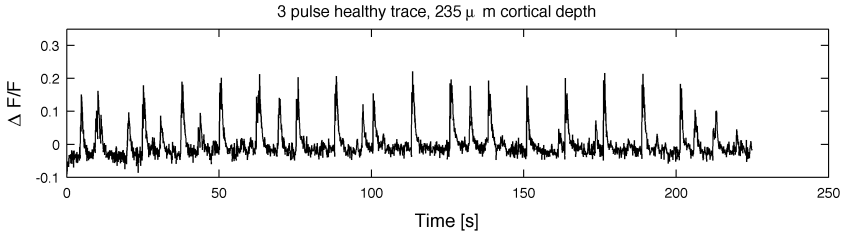
The traces are illustrated in Figure 7.1. The first two examples illustrated in Figure 7.1a and 7.1b are taken from 1 and 3-pulses wild-type (healthy) mouse respectively, whereas those presented in Figure 7.1c and 7.1d are taken from 1 and 3-pulses transgenic (Parkinson's) mouse respectively. The details of the animal models are summarized in Table 7.1.

Figure 7.1: *Typical two-photon calcium imaging traces in vivo, selected for algorithm application. (a) 1-pulse wild-type (b) 3-pulses wild-type (c) 1-pulse transgenic (d) 3-pulses transgenic*

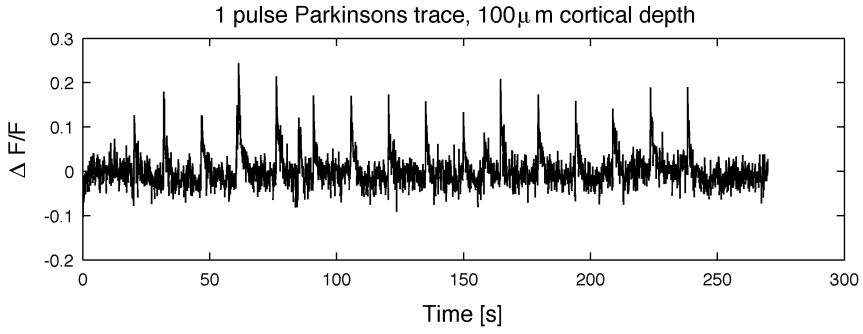
(a)



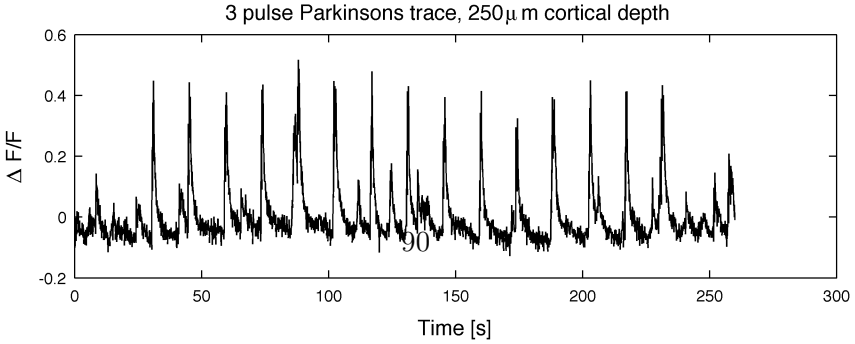
(b)



(c)



(d)



7.2.2 How do 1- and 3-pulses traces differ?

In general 1-pulse traces contain isolated elevations, whereas a train of three consecutive elevations characterizes most 3-pulses traces. In Figure 7.2, we have illustrated these characteristic differences. Figure 7.2a and 7.2c show typical trace responses of the wild-type and the transgenic neuron for 1-pulse stimulation respectively. Figure 7.2b and 7.2d display how the peaks of a wild-type and a transgenic neuron have responded to 3-pulses stimulation in a decreasing and an increasing order respectively.

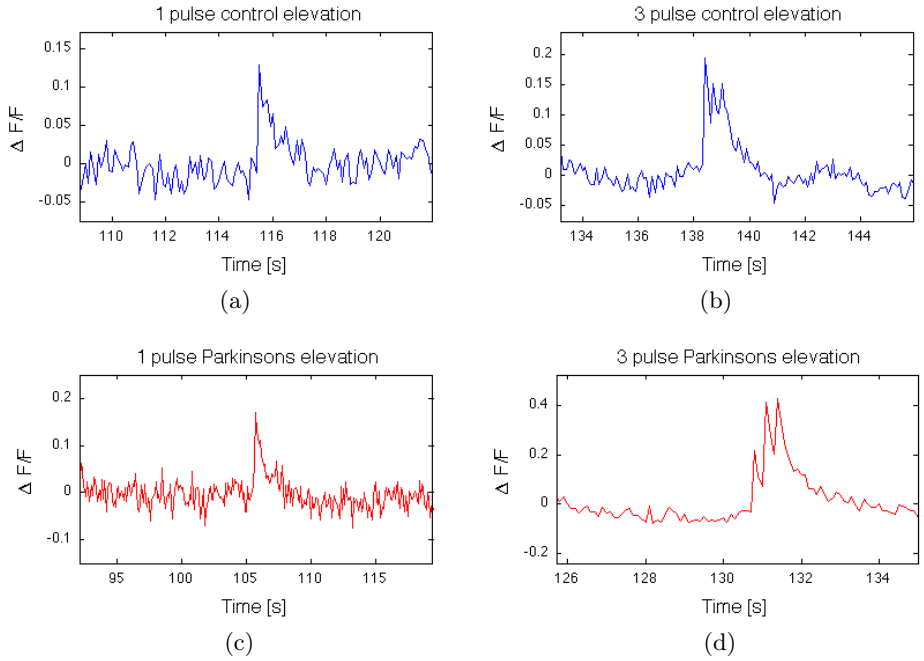


Figure 7.2: *Zoomed in traces for a closer look on characteristic differences between wild-type and transgenic neuron response to the nature of stimulation. (a) 1-pulse wild-type (b) 3-pulses wild-type. Peaks in decreasing order (c) 1-pulse transgenic (d) 3-pulses transgenic. Peaks in increasing manner.*

7.3 Estimating calcium kernels

7.3.1 1-pulse wild-type trace: Parameters as for model data

As a starting point, we attempt to fit for h_e using the parameter value $k = 11$. In Figure 7.3, h_e fitting made for the 1-pulse wild-type trace is illustrated. This fitting has reported $h_e = 0.082$. In order to investigate the goodness of this estimate, we perform the iterative nonlinear fit using $\alpha = 0.6$ and compute σ_τ.

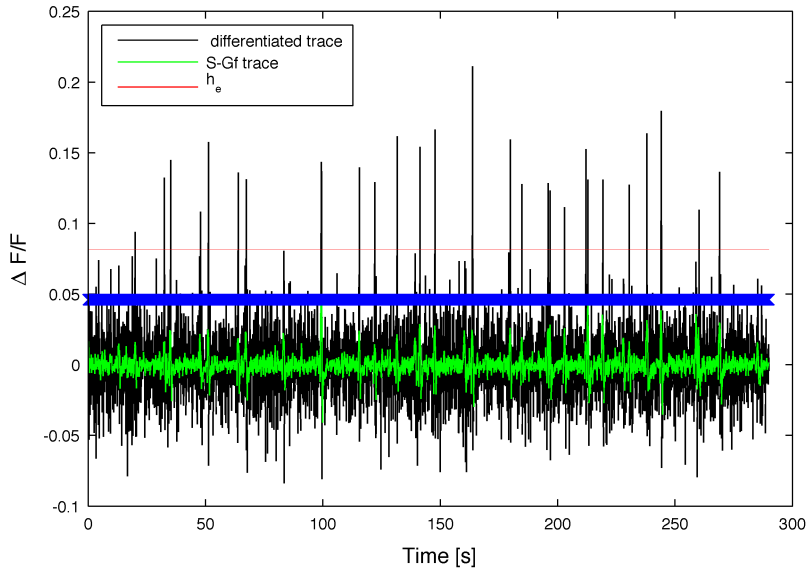


Figure 7.3: *Fitting for h_e with the best test data k (=11) threshold parameter value. Trace type: 1-pulse wild-type (0723R2ne2). The green trace is the S-Gf of the differentiated black trace. The blue line marks the roof of the S-Gf trace. The red line marks the estimated average calcium kernel amplitude,* $h_e = 0.082$.

From the 1-pulse wild-type example, only one calcium slope has fulfilled the minimum criteria to be included in the time constant estimation slopes category (illustrated in Figure 7.4). An exponential function is fitted to the data points through an iterative nonlinear least square method (red in Figure 7.4).

From the fitted exponential, we have obtained $\tau_{c_e} = 0.272s$, with large uncertainty ($\sigma_\tau = 0.137s$). This means, we need different values of k and α for experimental data.

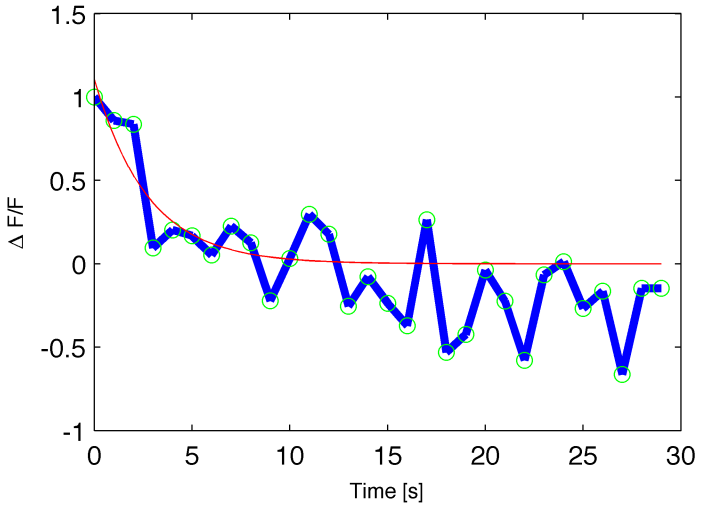
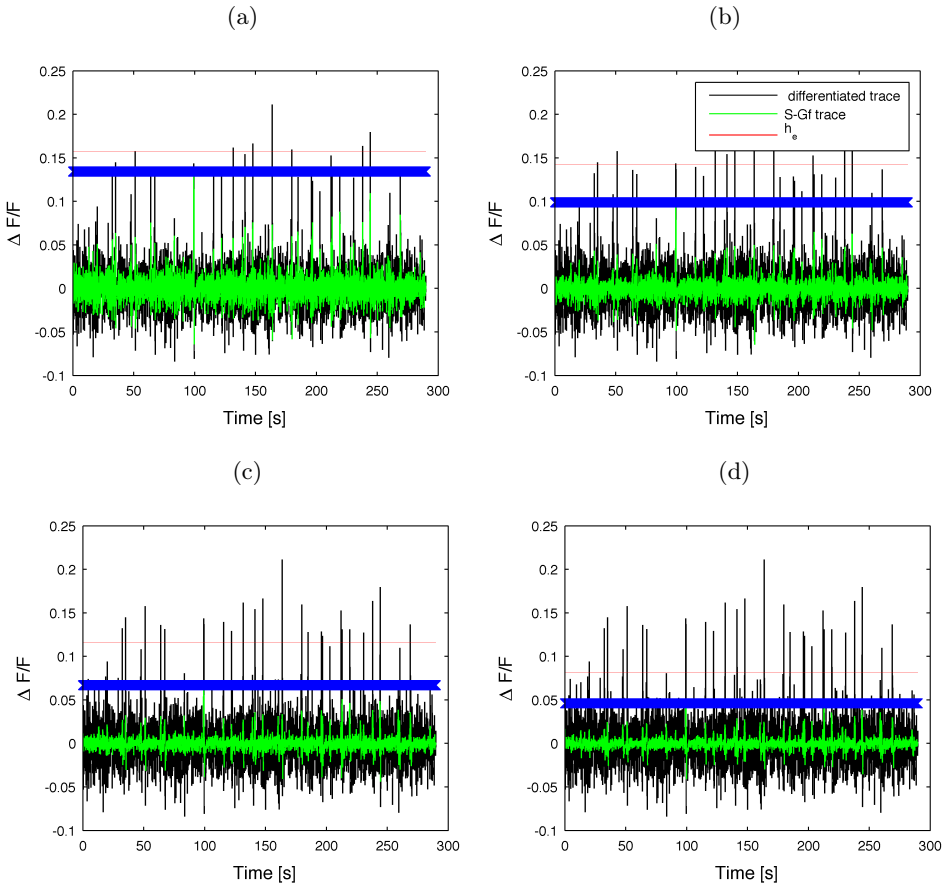


Figure 7.4: *(a) 1-pulse wild-type trace slopes. Only one slope has fulfilled the minimum criteria to be considered as action potential initiated, when model data threshold parameters ($k = 11$ and $\alpha = 0.6$) are applied. b) Fitted exponential (red) gave $\tau_{c_e} = 0.272s$. The covariance matrix gave $\sigma_\tau = 0.137s$.*

7.3.2 1-pulse wild-type: $k = 5, 7, 9, 11$ and $\alpha = 1$

The kernel amplitude fits for the different k are illustrated in Figure 7.5. In the figure, black denotes the differentiated experimental trace, green the S-Gf of the black trace, blue the roof of the the green trace and red marks h_e.

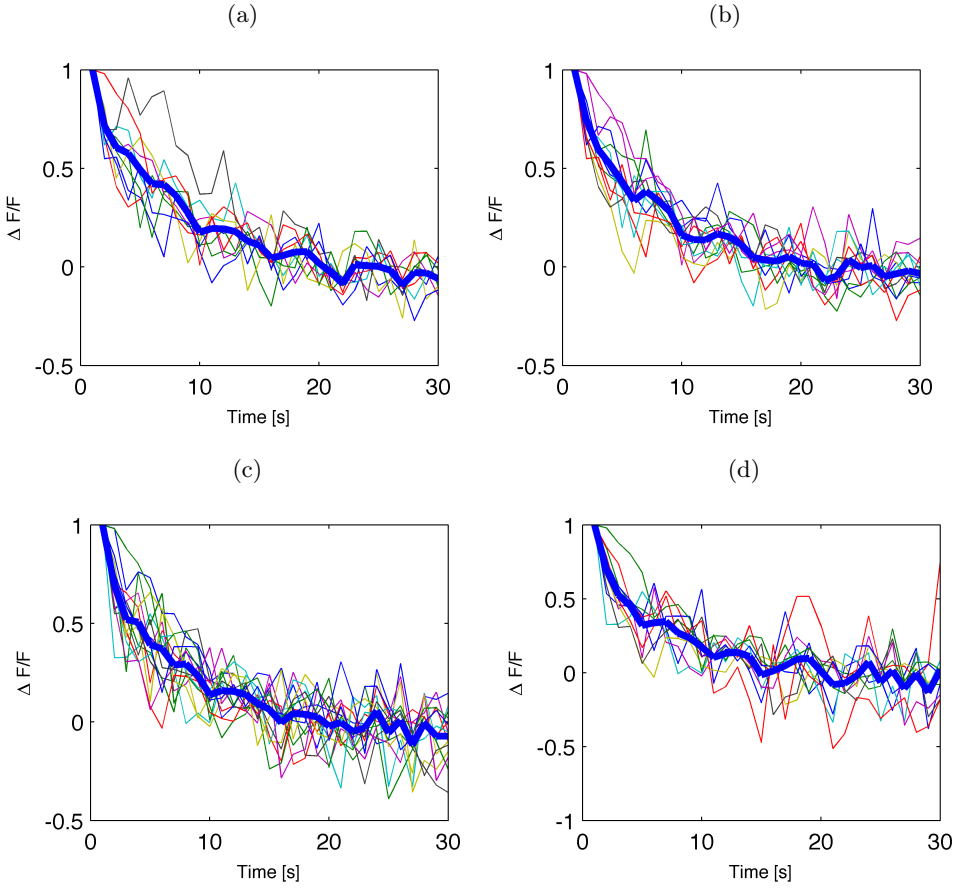
Figure 7.5: h_e *fitted with different* k *when* $\alpha = 1$ *for 1-pulse wild-type trace. The blue line marks the roof of the S-Gf trace. The red line marks* h_e*. (a)* $k = 5$*:* $h_e = 0.1577$ *(b)* $k = 7$*:* $h_e = 0.1419$ *(c)* $k = 7$*:* $h_e = 0.116$ *(d)* $k = 11$*:* $h_e = 0.0817$



According to Figure 7.5, decreasing k increases the amplitude and the thickness of the S-Gf trace. This pushes the amplitude threshold (max S-Gf) upwards. Consequently, the smallest k ($= 5$) has given the largest h_e ($= 0.158$). Conversely, the smallest h_e ($= 0.082$) is obtained when $k = 11$.

In Figure 7.6, trace slopes that have satisfied the inclusion criteria (see section 5.2.2 and Equation 7.1) when $\alpha = 1$ are presented.

Figure 7.6: *1pulse wilde type trace slopes that have satisfied the minimum inclusion criteria. Slope averages are marked by thick blue. (a)* $k = 5$ *(b)* $k =7$ *(c)* $k = 9$ *(d)* $k = 11$.



We fit exponential functions to the slope averages (marked in thick blue in Figure 7.6). The exponential fits are presented in red in Figure 7.7.

From the fitted exponentials in Figure 7.7, we have extracted τ_{c_e} =0.619s, 0.565s, 0.516s and 0.496s when $k = 5$, 7, 9 and 11 respectively. Like for h_e, the smallest k (= 5) has given the largest time constant estimate.

In Table 7.2 the fitted kernel parameters for the 1-pulse wild-type trace are summarized.

k	h_e	τ_{c_e} [s]	σ_τ [s]
5	0.158	0.619	0.0093
7	0.142	0.565	0.0094
9	0.116	0.516	0.0128
11	0.082	0.496	0.0167

Table 7.2: *Fitted kernel parameters and error terms for the 1-pulse wild-type trace (0723R2ne2) when* $\alpha = 1$

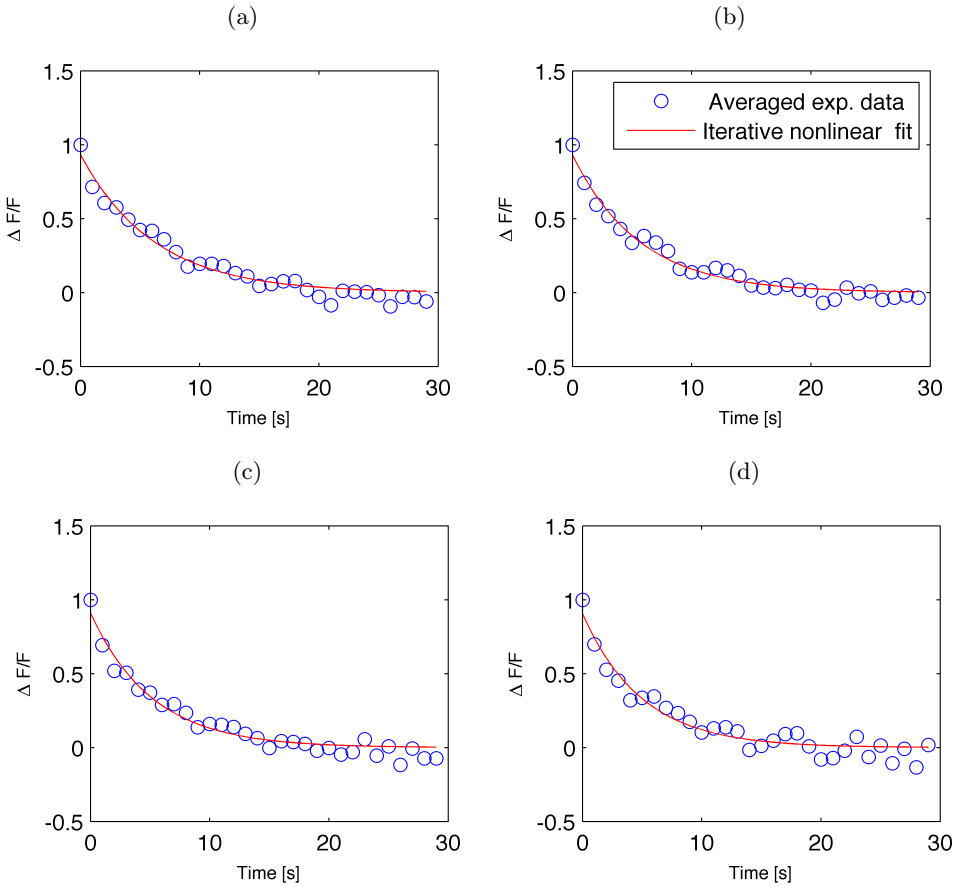
According to Table 7.2, both $k = 5$ and $k = 7$ (when $\alpha = 1$) are well suited when the kernel parameters of this particular trace are fitted.

7.3.3 Fitting functions to calcium slopes when k is large

To check whether we can use large k, we consider $k = 13$ and $k = 19$. The trace decays and the fitted exponential functions for these k are illustrated in Figure 7.8

In Figure 7.8, when k is large, fewer slopes are selected than when k was small in Figure 7.6. In addition, the fitted exponentials have flat bottom and contain several data point outliers. This means, increasing k too much affects the quality of fitted kernel parameters negatively, in a similar way as scaling with $\alpha = 0.6$ did in Figure 7.4. The large error terms (σ_τ listed in Table 7.3) are also indicative of bad estimates due to increased k.

Figure 7.7: *Exponential functions fitted to the calcium trace slope averages (the thick blue curves in Figure 7.6) . The Y-label* $\frac{\Delta F}{F}$ *is normalized. (a)* $k = 5$*:* τ_{c_e} =0.619 *and* σ_τ =0.0093 *(b)* $k = 7$*:* τ_{c_e} =0.565 *and* $\sigma_\tau = 0.0094$ *(c)* $k = 9$*,* τ_{c_e} =0.516 *and* σ_τ =0.0128 *(d)* $k = 11$*:* τ_{c_e} =0.496 *and* σ_τ =0.0167



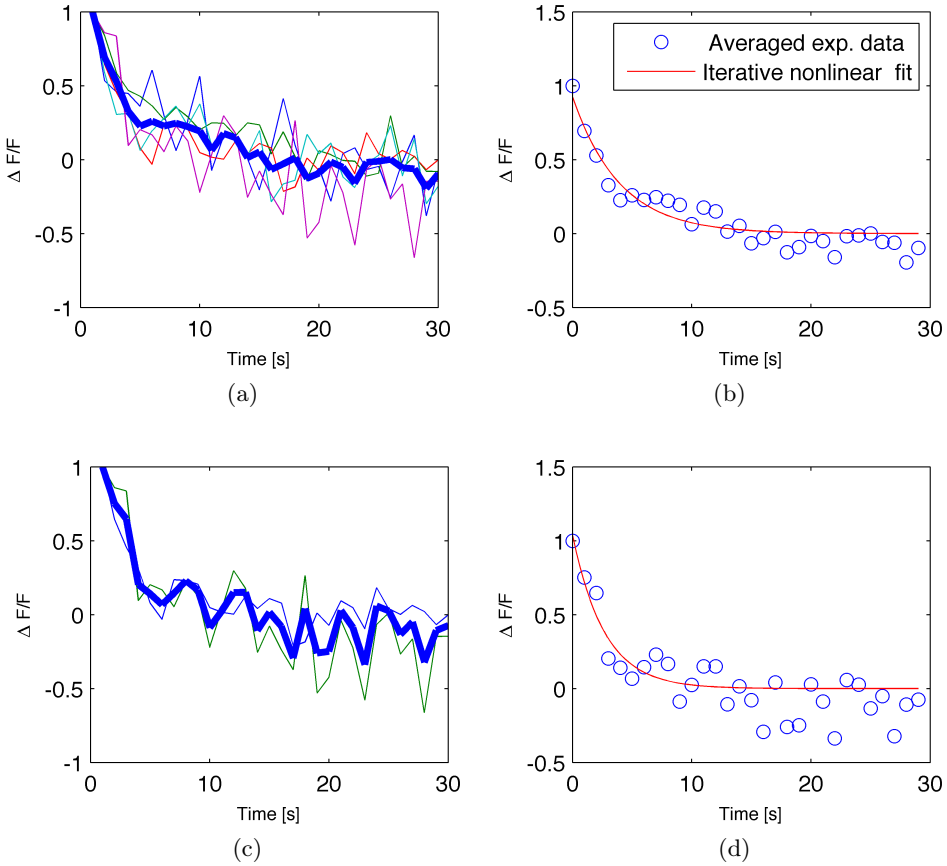


Figure 7.8: *Exponential functions fitted to the 1-pulse wild-type calcium trace slope averages , when k is large. (a) $k = 13$: Slopes (b) $k = 13$: Fitted exponential (c) $k = 19$. (d) $k = 19$*

k	h_e	τ_{c_e} [s]	σ_τ [s]
13	0.062	0.394	0.0302
19	0.044	0.268	0.0820

Table 7.3: *Summary of fitted parameters and their error term for k=13 and k=19. Large k has resulted in large uncertainty.*

7.3.4 1-pulse transgenic

The parameters of the kernel fit and their uncertainty for the 1-pulse transgenic trace obtained when $k = 5, 7, 9$ and 11 are summarized in Table 7.4.

k	h_e	τ_{c_e}[s]	σ of τ_{c_e}[s]
5	0.156	0.759	0.0078
7	0.127	0.768	0.0089
9	0.080	0.683	0.0188
11	0.063	0.099	0.5853

Table 7.4: *Fitted kernel parameters and their error terms for 1-pulse transgenic trace (0728R15ne4).* .

According to Table 7.4, the algorithm has estimated the time constant to 0.76 s (for $k = 5$ and 7) with a standard error 0.008 s. In this case, the ratio of σ_τ to τ_{c_e} is about 1%. h_e is oppositely related to increasing k but not necessarily τ_{c_e}.

7.3.5 3-pulses transgenic

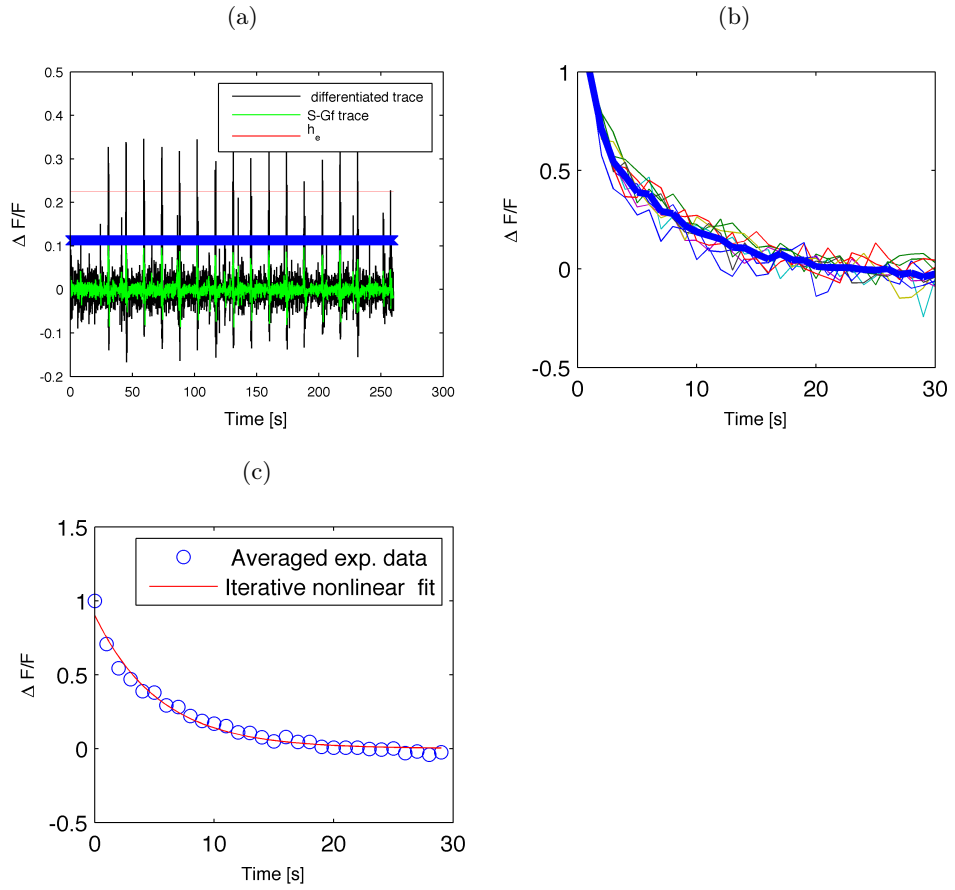
For the 3-pulses transgenic trace, the kernel fitting procedure when $k = 7$ are illustrated in Figure 7.9. In this case, $h_e = 0.226$. A time constant of 0.536 s with 0.0081 s standard error is extracted from the fitted exponential function to the slope average of the trace.

In Table 7.5, we have extended k 's range up to 17 and listed the fitted kernel parameters. The table shows an interesting relationship between h_e and τ_{c_e}. When the signal quality is good, as it is the case for this particular trace, a stable time constant can be estimated by the algorithm regardless of changing h_e and k. In other words, τ_{c_e} does not always decrease due to increasing k like h_e. In addition, the stablity of τ_{c_e}, together with the small standard errors (σ_τ), in the interval $5 \leq k \leq 15$ indicates the likelihood of the fitted time constant to be the most probable estimate.

k	h_e	τ_{c_e}[s]	σ of τ_{c_e}[s]
5	0.248	0.536	0.0081
7	0.226	0.536	0.0081
9	0.215	0.539	0.0079
11	0.199	0.539	0.0079
13	0.191	0.539	0.0079
15	0.173	0.540	0.0080
17	0.146	0.474	0.0120

Table 7.5: *Fitted kernel parameters and error terms for 3-pulses transgenic trace. Stable* $\tau_{c_e}(= 0.54)$ *with small standard error* $(= 0.008)$ *is estimated for the interval* $5 \leq k \leq 15$*, regardless of decreasing* h_e *due to increasing* k*.*

Figure 7.9: *Calcium kernel fitting procedure for 3-pulses transgenic trace, when* $k = 7$. *(a) Black: Differentiated trace. Green: S-Gf of the black trace. Blue: The roof of the green trace. Red:* $h_e = 0.226$. *(b) Calcium slopes satisfying the minimum inclusion criteria. Thick blue: Slope average. (c) Red: Fitted exponential function slope average data points.* $\tau_{c_e} = 0.536$ *s and* $\sigma_\tau = 0.0081$ *s.*



7.3.6 3-pulses wild-type

In Table 7.6, fitted kernel parameters and uncertainties for the 3-pulses wild-type trace are listed.

k	h_e	τ_{c_e}[s]	σ_τ of τ_{c_e}[s]
5	0.142	0.476	0.0152
7	0.117	0.417	0.0184
9	0.103	0.425	0.0203
11	0.100	0.401	0.0233

Table 7.6: *Fitted kernel parameters and error terms for the 3-pulses wild-type trace .*

According to Table 7.6, the standard deviation to time constant ratio (σ_τ : τ_{c_e}) for k =5 and 7 are smaller than for other k (about 3% and 4.5% respectively). This implies, $k = 5$ or $k = 7$, as for the other trace types studied, being the well suited S-Gf threshold parameters for this particular trace.

7.4 Summarizing kernel parameter estimation

We conclude the section about calcium kernel parameters by giving a brief summary of the main results in Table 7.7.

k	Stimulus in pulses	Mouse type	Cortical depth	h_e	τ_{c_e} [s]	σ_τ [s]	$\sigma_\tau : \tau_{c_e}$ [%]
5	1	wild-type	$150\mu m$	0.158	0.619	0.0093	1.5
		transgenic	$100\mu m$	0.156	0.759	0.0078	1
7	1	wild-type	$150\mu m$	0.142	0.565	0.0094	1.7
		transgenic	$100\mu m$	0.127	0.768	0.0089	1.2
5	3	wild-type	$235\mu m$	0.142	0.476	0.0152	3.2
		transgenic	$250\mu m$	0.248	0.536	0.0081	1.5
7	3	wild-type	$235\mu m$	0.117	0.417	0.0184	4.4
		transgenic	$250\mu m$	0.226	0.536	0.0081	1.5

Table 7.7: *Summary of calcium kernel fitting for the four experimental traces (Figure 7.1). k=5 and k=7 (when $\alpha = 1$) are well suited as S-Gf smoothing threshold parameters in the algorithm. Except for the 3-pulses wild-type trace (the noisiest among the four), time constants are estimated with <2% normalized error ($\frac{\sigma_\tau}{\tau_{c_e}}$). Larger time constants are estimated for transgenic traces both for in 1 and 3-pulses stimulation cases. h_e has decreased with increasing k for all cases. Among traces, the largest amplitude is estimated for the 3-pulses transgenic. The h_e of the 3-pulses transgenic is twice as large as its 3-pulses wild-type. In the case of transgenic traces, h_e is larger for larger cortical depth and vice versa for τ_{c_e}.*

7.5 Spike train reconstruction and calcium trace fit

We can summarize the spike train reconstruction algorithms as follows: A spike train will be reconstructed with deconvolution deterministically with the estimated kernel (see section 5.3 and 6.2 for its design and testing respectively). Then we calculate the calcium trace corresponding the reconstructed spike train. We optimize the positions of spikes by comparing the fitted calcium trace with experimental trace.

The entire procedure will be applied on the four traces (Figure 7.1). In this subsection, experimental traces of the 1-pulse and 3-pulses will be denoted by blue and red respectively. For DDA calcium trace fits we use magenta and green will denote the optimized trace fits. For the sake of comparison, the corresponding spike train for each (DDA and GOA) calcium trace fit will be shown by dots underneath.

The parameters $k = 5$ and $\vartheta = 0.5$ will be employed for the illustrations and discussion in this subsection.

7.5.1 1-pulse wild-type

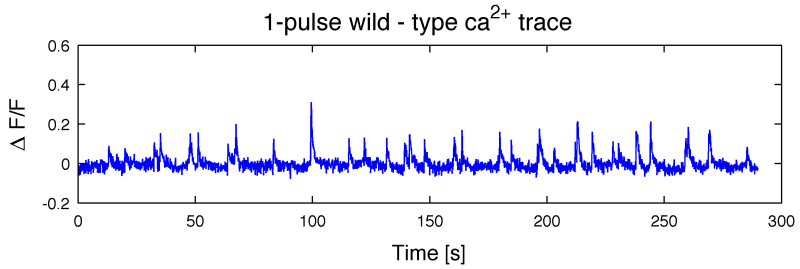
The kernel employed for 1-pulse wild-type trace is shown in Figure 7.7b.

In Figure 7.10, the experimental Ca $^{2+}$ trace, the DDA Ca^{2+} trace fit with its underlying spike train and the optimized Ca^{2+} trace fit with its underlying spike train are illustrated in (a), (b) and (c) respectively.

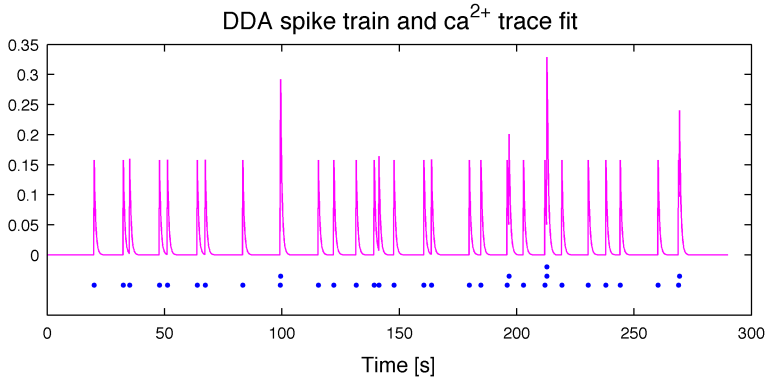
In this case, the calcium trace is fitted very well both with and without optimization, as it can be seen from the illustrations. Dominant elevations in the experimental trace have coincided with the optimized calcium trace fit both in time and size. However, some dominant elevations seen in the experimental trace are missing when it comes to the DDA trace fit.

Both the DDA and the optimized spike trains have indicated only individual spike arrivals. For instance in Figure 7.10b and c, the two blue dots on top of each other at about 100 s on the spike train indicate that the two spikes have arrived close to each other . In this case, the first spike has arrived at about 99.3 s, whereas the second spike (the one on the top) has arrived at about 99.4 s.

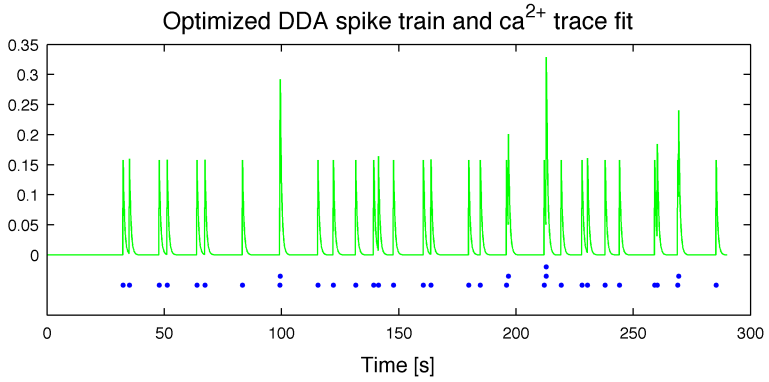
In total, 33 and 34 spikes are estimated by the DDA and the GOA respectively. However, by comparing the blue spike dots in Figure 7.10b and c, one can see some removed 'false' spikes, for instance at about 45 s and 205 s on the time axis. In addition, some spikes are also added to the spike train when deconvolution is greedy optimized, for instance, the spike at about 280 s in Figure 7.10c.



(a)



(b)



(c)

Figure 7.10: *Reconstructed spike trains for the 1-pulse wilde-type trace. The blue dots underneath the trace fits in (a) and (b) represent the corresponding spike trains for the DDA and the GOA respectively. In both spike trains spikes have arrived individually, but sometimes very close to one another (for instance, at about* 100 *s). Some spike are removed and some added to the train by when the deconvolution spike train is optimized.*

7.5.2 1-pulse transgenic

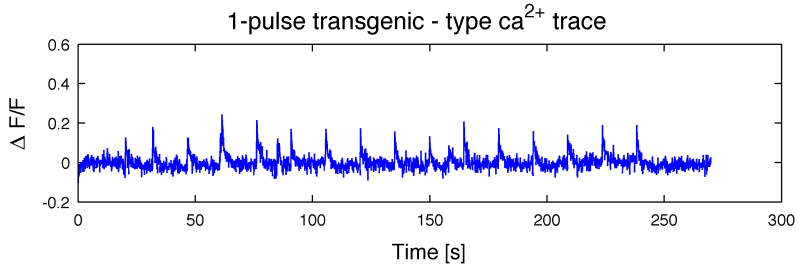
In this subsection, we will demonstrated the impact of the parameter k and ϑ on spike reconstruction. In Table 7.8, we have summarized the number of estimated spikes ($\sum_{i=1}^{N}\eta_i$, where the i^{th} spike is denoted by η_i as in Equation 6.6) through deconvolution (with different k) and optimization (with different k and ϑ).

DDA has estimated 54 spikes for the 1-pulse transgenic trace, when $k = 7$, where as only 31 spike detection were made when $k = 5$. The larger number of spikes are detected when $k = 7$ because deconvolution has detected 2 spikes arriving at same time in total 6 times. Non of these 2 spike arrival at the same time is detected when $k = 5$. This means, spike detection can be determined by the size of k. And k is oppositely related with estimated h_e (see Table 7.7).

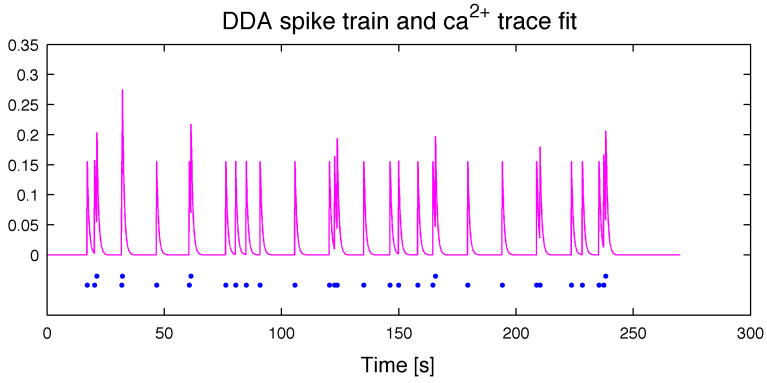
We have illustrated the spike trains and the Ca^{2+} trace fits of the 1-pulse transgenic-type trace in Figure 7.11. There are only individual spike arrivals in the deconvolved and optimized spike trains.

	Estimated	number	of spikes	=	$\sum_{i=1}^{N}\eta_i$
k	DDA	$\vartheta = 0.2$	$\vartheta = 0.5$	$\vartheta = 0.8$	$\vartheta = 0.9$
5	31	21	24	40	48
7	54	21	20	20	20

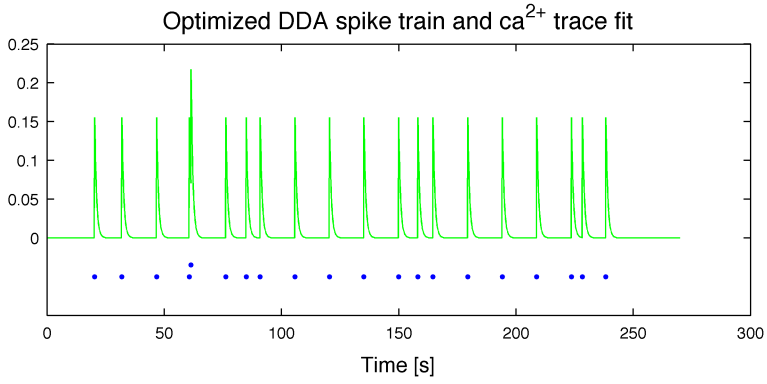
Table 7.8: *The table illustrates the impact of the parameter k and ϑ on spike train reconstruction. For $k = 7$, DDA has detected multiple (6) times when two spikes arrive at the same time, therefore, 54 spike detections. This means k (oppositely related with h_e Table 7.7) can increase the number of spikes detected by the DDA. Optimization is influenced to a lesser degree by k, as long as ϑ is kept at about* 0.5.



(a)



(b)



(c)

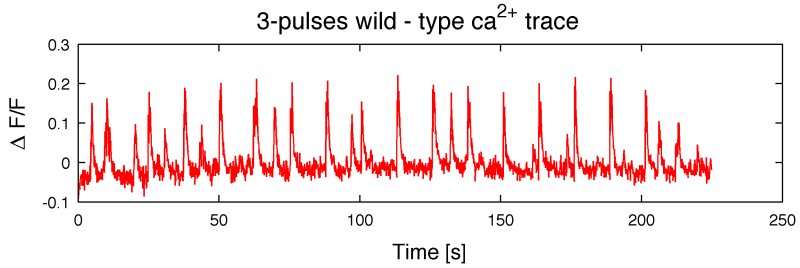
Figure 7.11: *Spike train and Ca^{2+} fit for the 1-pulse transgenic-type trace. Here $k = 5$ and $\vartheta = 0.5$. In total deconvolution has estimated 31 spikes. Optimization has removed 11 of the 31 DDA spikes. Spikes have arrived individually in both the DDA and GOA spike trains.*

7.5.3 3-pulses wild-type

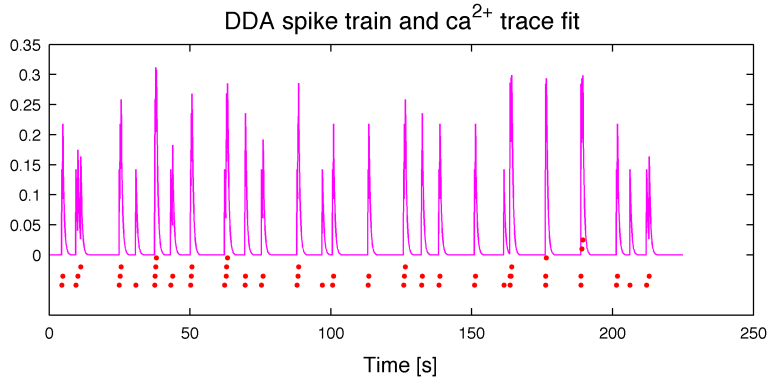
In Figure 7.12, the reconstructed spike trains and the calcium trace fits for the 3-pulses wilde-type trace are illustrated. DDA has reported 2 spikes arriving together three times, whereas only single spike arrivals are reported when the deconvolution is optimized.

For this experimental trace, 60 spikes are estimated by the DDA, optimization has removed 15 of those spikes.

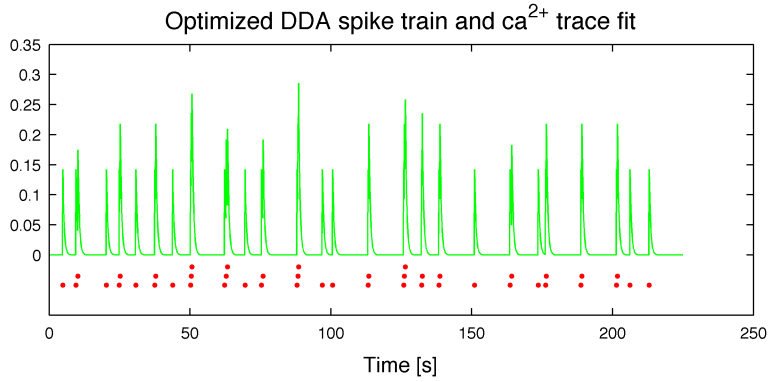
We investigate the plots a bit more closer, we zoom into the segment of Figure 7.12 (between 6 and 25 seconds) in Figure 7.13. The zoomed experimental Ca^{2+} trace is shown in (a). As it is illustrated in Figure 3.2 and 3.4 (experimental flow), no stimulation is given to the mouse during the first 20 seconds of the experiment (or the first stimulus arrives at about 25 s) . Therefore, the fluorescence elevations (Figure 7.13a) at about 10 s (spike assembly) and 20 s must have occured due to spontaneous activities. For the spike assembly at $\sim$ 10 s, DDA has detected 3 spontaneous events; only 2 of these have remained after optimization. The GOA considers the last arriving event in the assembly as a sub-threshold activity. On the other hand, the spontaneous event at $\sim$ 20 is neglected by the DDA but detected and added to the spike train by the GOA as shown in Figure 7.13c.



(a)

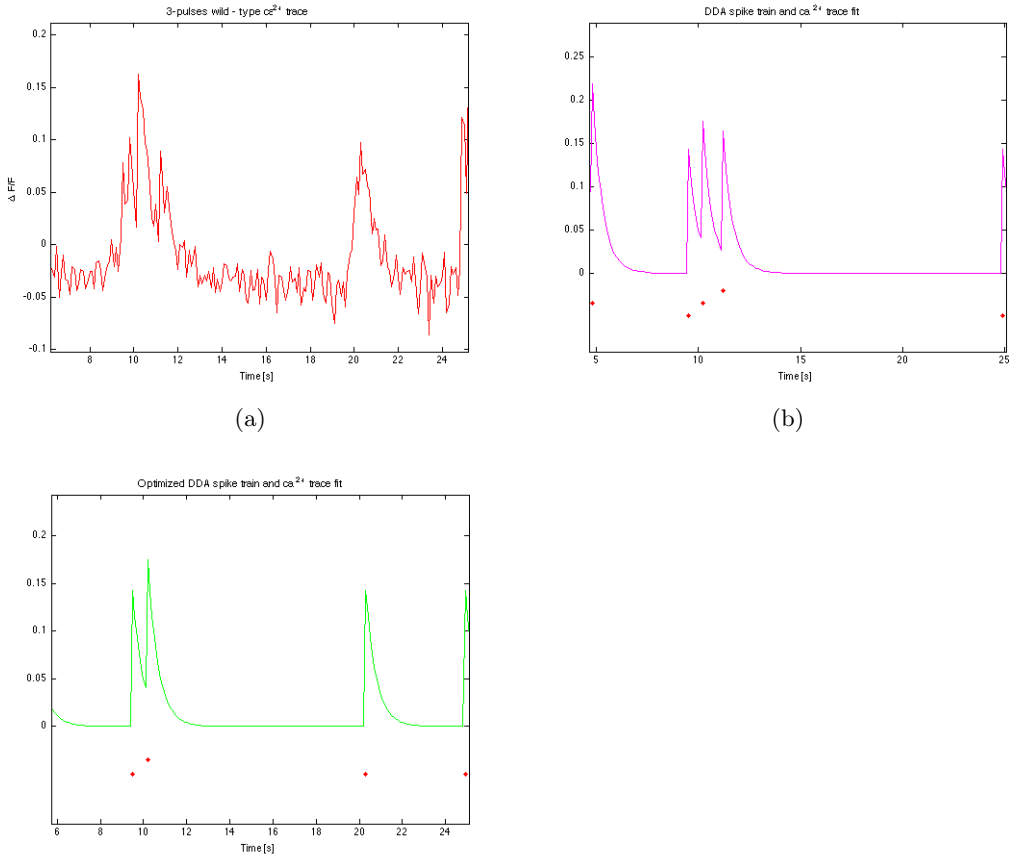


(b)



(c)

Figure 7.12: *Reconstructed spike trains and their corresponding* Ca^{2+} *trace fits. DDA and GOA have detected* 60 *and* 45 *spikes for the 3-pulse wilde-type trace respectively. The* Ca^{2+} *fits have larger amplitudes than those in Figure 7.10 and 7.11 due to dense spike arrivals resulting from the 3-pulse whisker stimulation. DDA has detected* 2 *spikes arriving together* 3 *times, non of these are detected when the train is optimized.*



(a) (b)

(c)

Figure 7.13: *Figure 7.12 zoomed in. The spontaneous fluorescence elevations (see Figure 3.2 and 3.4) at* ~ 10 *s and* ~ 20 *s in (a) are detected by the optimized deconvolution as illustrated by (c).At* ~ 10 *s, DDA has reported three spontaneous events; however, the* 3^{rd} *event in the assembly is considered as arising from sub-threshold activity and removed during optimization. The spontaneous event at* ~ 20 *is neglected by deconvolution as illustrated in (b).*

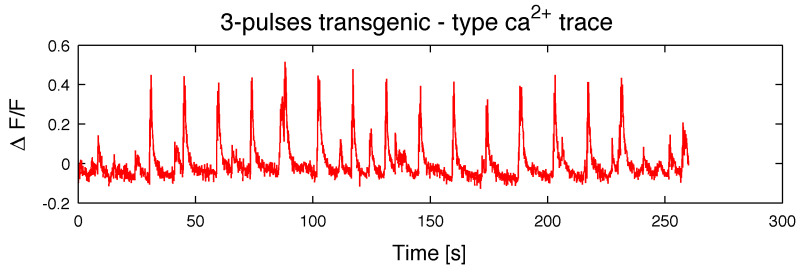
7.5.4 3-pulse transgenic

The reconstructed spike trains and the fitted Ca^{2+} traces for the 3-pulse transgenic-type trace are shown in Figure 7.14.

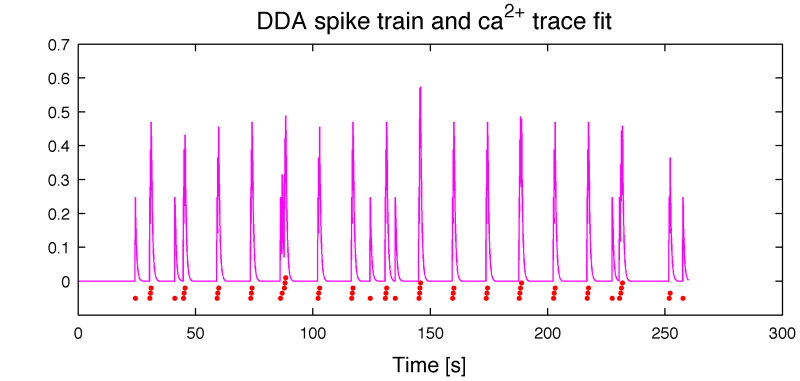
In total, deconvolution and optimization have detected 58 and 54 spikes in the experimental trace respectively. In this case, the GOA has reported 2 spikes arriving together at $\sim$ 102 s, no such case is reported by the DDA.

In Figure 7.15, we have zoomed into a segment (between $\sim$ 130 and $\sim$ 155 s) of Figure 7.14.

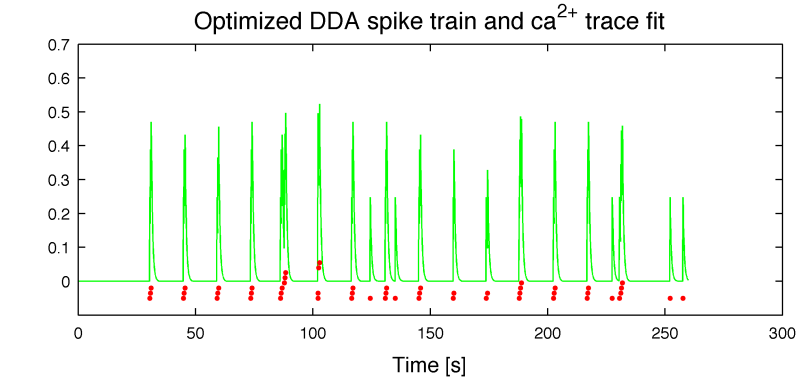
As it is explained in section 3.2 and illustrated in Figure 3.2 and 3.4, whiskers are given a train of 3 electrical stimulations in 15 seconds interval, the first one arriving to the neuron at $\sim$ 25 second after the experiment run is started. This means the hierarchically assembled fluorescence elevations at $\sim$ 132 and $\sim$ 147 s in Figure 7.15a must have been due to evoked APs, whereas the elevation coming after the 1^{st} assembly at $\sim$ 135 s should be due to a spontaneous event. The spontaneous elevation is recognized both by the DDA and the GOA. Deconvolution has detected 4 underlying spikes for the 2^{nd} spike assembly at $\sim$ 147 s. One of these four spikes is removed when deconvolution's spike train is optimized. If we compare the hierarchy of the 2^{nd} elevation assembly in (a) and (b) with (c), we acknowledge the optimized fit as being more coinciding with experimental fluorescence transient elevation hierarchy.



(a)

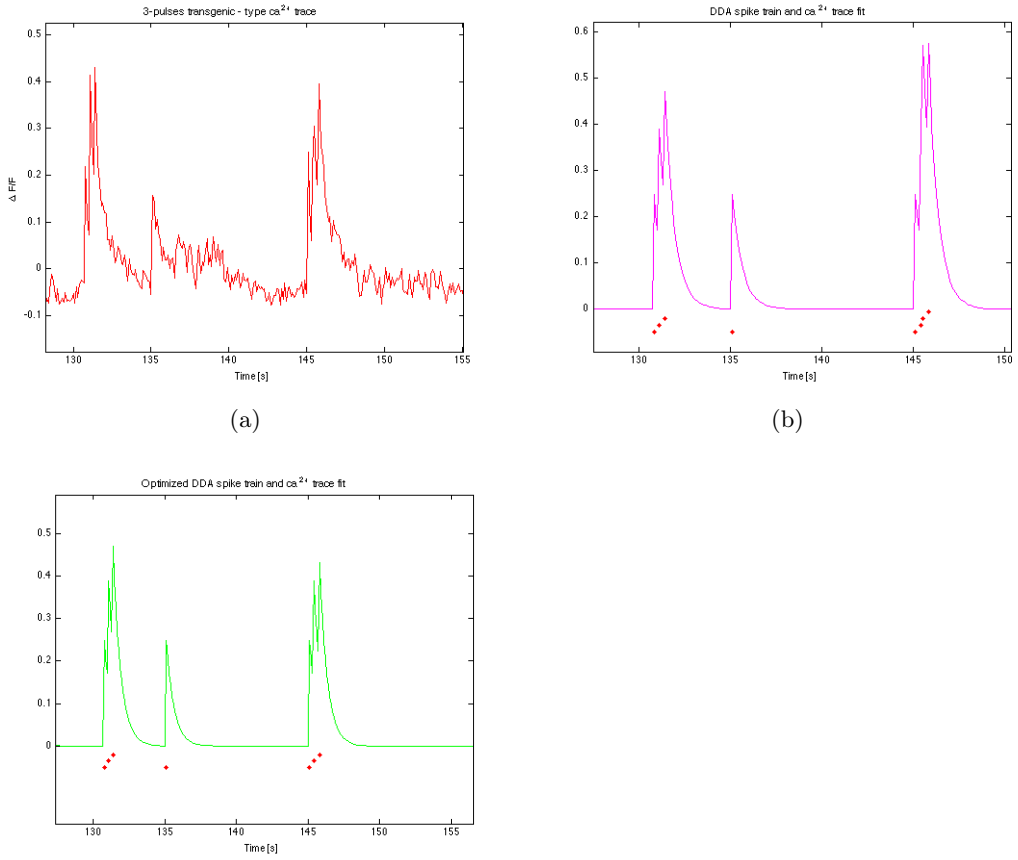


(b)



(c)

Figure 7.14: *Reconstructed spike trains and Ca^{2+} trace fits for the 3-pulse transgenic-type trace. 58 and 54 spikes are detected in the trace by the DDA and the GOA in the trace respectively. Dense spike arrivals are observed in ∼15 seconds intervals, reflecting stimulus (in 3-pulses) onsets (see Figure 3.2). Optimization has reported one situation (∼ 102 s) where two spikes have arrived together. No such a situation is reported by the GOA.*



(a) (b)

(c)

Figure 7.15: *A segment (∼ 130 s to ∼ 155 s) of Figure 7.14 zoomed in. The assembled fluorescence transient elevations at ∼ 132 s and ∼ 147 s in the experimental trace in (a) are evoked (see Figure 3.2 and 3.4). Optimization (c) has managed to follow both the fluorescence transient elevations hierarchy (for the evoked case) as well as the spontaneous event at ∼135 s .*

8 Discussion

We have modeled a family of calcium imaging fluorescence signals.

A novel algorithm is developed to estimate a calcium kernel from a given fluorescence trace. We have designed the algorithm by combining differentiation, Savitzky-Golay filter with and an iterative algorithm for fitting a nonlinear model (Levenberg - Marquardt).

For spike detection we have developed a deconvolution algorithm that employs greedy optimization.

The modeled signals have played significant roles in fine-tuning the designed algorithms.

In vivo two-photon calcium imaging signals are analyzed with these algorithms.

8.1 Modeled Ca^{2+} signals

We have employed an alternative algorithm for convolution than the standard MATLAB function *conv.m* for the purpose of varying kernel types when we modeled the calcium signals.

8.1.1 Kernel amplitude

The average amplitude of a signal with low SNR ($\xi \leq 8$) is estimated with the largest error (δ_h), as demonstrated in Figure 6.1, 6.3 and 6.7.

However, estimation errors were significantly reduced when the AGWN of a poor quality signal is filtered with MAS, as in shown by the δ_h in Figure 6.4a and b. This means, the algorithm is sensitive for high frequency noise component.

Our method is effective in taking the inherent noise level in a signal into consideration when setting the amplitude threshold criteria. For instance, as illustrated in Figure 6.9, it was possible to estimate the average amplitude of the typical noisy signal with 5% error from expectation (h_0).

Estimating the kernel amplitude from the differentiated signal rather than from the original signal is advantageous. Wider (plateaued) calcium transients, for instance due to closely arriving spikes, will be attenuated without losing the underlying information.

Employing the S-Gf algorithm to exclude peaks associated with the noise component is appropriate. For instance, unlike MAS, peaks that are narrow compared to the filter width (k) will not be damaged when the filter passes through the peaks [10]. In addition, S-Gf is optimal, in a sense that it minimizes the least-squares error in fitting a polynomial to each frame of the noisy calcium trace. This makes the S-Gf a better choice than other finite impulse response (FIR) filtering methods for our purpose. In most cases, FIR tends to filter out a significant portion of the signal's high frequency content along with the noise [29].

8.1.2 Kernel time constant

Combining αh_e (Equation 5.19) and the kernel width $6\tau_{c_0}$(section 5.1.2) as a calcium slope selection criteria has worked on all the 672 modeled signals. For the modeled data $\alpha = 0.6$ has shown good results. If necessary, the threshold scalar α can be tuned for any signal.

Varying α had, especially, an implication for signals with $\xi < 4$, as shown in Figure 6.18. Therefore, one needs to be cautious in choosing its value.

Time constant is extracted from an exponential function fitted to a slope average. Fitting through linear transformation gave larger errors particularly for signals with poor quality ($\xi = 1, 2, 3, 4$) as in Figure 6.11.

Fitting exponential functions to calcium slopes through the iterative nonlinear model is advantageous over the linear model.

Time constant estimation error is significantly reduced for the noisiest signals with respect to expectation (τ_{c_0}) with the nonlinear model, as shown in

Figure 6.16 and 6.17. For instance, the time constant of the signal in Figure 6.12 was off only by 1.6% from expectation when estimated by the nonlinear model compared to a 22% error of the linear model.

8.1.3 AP reconstruction

DDA: Deconvolution has estimated spike trains precisely for signals of good quality ($\xi > 8$), even when the AWGN is unfiltered (for all ω) as shown in Figure 6.19 and 6.21.

For signals of low quality ($\xi \leq 8$), estimation quality is determined by the size of ω, as shown in Figure 6.20.

We have noticed a sharp jump in estimation quality (from hundreds of false spikes to zero) with increasing ξ and ω, as shown in by the blue and red pixels of panels in Figure 6.19.

The spike inference quality for the DDA is significantly reduced due to to kernel estimation bias, as demonstrated in Figure 6.19 and Figure 6.26. According to the later, deconvolution is inadequate only if $\xi \leq 4$ and the drift in the AWGN is high ($\omega = 1$ and 2).

MF: Smoothing a signal by MF as a pre-processing to the DDA, impaired the performance of the DDA for some signals of good quality ($\xi > 8$), as in Figure 6.23.

On the other hand, MF has removed wrongly estimated spikes by deconvolution for poor quality signals. However this issue needs more invesitgation, since our model could not identify pattern for the improvement.

GOA: Optimizing the DDA has enabled to remove hundreds of false spikes for signals with $\xi \leq 8$.

$\vartheta = 0.2$ (Equation 6.8) was more suited for signals of higher drift AWGN ($\omega = 1, 2, 3, 4, 5, 7$) and $\xi = 1, 2, 3, 4, 8$, as shown in Figure 6.24 and 6.25.

In general, an iterative optimization in the greedy approach was time consuming, taking up to 6 hours of CPU time for the entire family of model data (672 signals each lasting for 200s).

8.2 Experimental signals

We have found that, $k = 5$ and $\vartheta = 0.5$ are the best suited parameters for the purpose of application.

8.2.1 Amplitude and time constant

One of the main results from experimental data analyses is that, larger time constants are estimated for the transgenic traces compared to their wild-type counterparts for both 1- and 3-pulse stimulation cases, as listed in Table 7.7 .

Among the transgenic traces, we have obtained larger time constant ($\tau_{c_e} = 0.77$) for the neuron imaged at cortical layer I (100 μm) and stimulated with a single pulse. We have not encountered direct correlation between h_e and τ_{c_e}, such as larger h_e leading to larger τ_{c_e}. Cortical depth may have implication on this issue; therefore, it needs further investigation.

Nearly twice as large h_e is estimated for the transgenic trace stimulated with 3-pulses compared to its single pulse counterpart.

For the 1-pulse traces, nearly equal kernel amplitude is estimated for both wild-type (cortical layer II) and transgenic traces (cortical layer I) traces.

h_e of the transgenic trace is twice as large as the wild-type's h_e for the 3-pulses traces.

For all the traces investigated, estimation error ($\sigma_\tau : \tau_{c_e}$) was $< 5\%$, where σ_τ is obtained from the covariance matrix of the nonlinear model fit.

8.2.2 AP

DDA can infer about the same number of spikes as the GOA, as seen from the 1-pulse wild-type trace in Figure 7.10b and c. But this does not mean there is the same spike pattern in the two.

More frequent spike arrivals are observed for the traces stimulated with 3-pulses.

Both spontaneous and evoked events are fitted by the GOA method as demonstrated for the 3-pulses traces in Figure 7.13 and 7.15.

The hierarchical order of Ca^{2+} transient elevations of traces stimulated with 3-pulses (Figure 7.2b and 7.2d) is detected by the GOA calcium trace fit, as shown in Figure 7.13 and 7.15.

Our greedy optimization approach is iterative, where another spike is added or removed from the DDA spike train during each iteration. We believe that the spike trains obtained after the optimization are the most likely one.

8.2.3 Conclusion

The major findings of the thesis can be summarized as follows:

• We have modeled a family of calcium imaging fluorescence signals, which we have employed for algorithm design, test and analysis.

• Using a novel algorithm, we have shown slower calcium response in somatosensory neurons of α-syn transgenic mice.

• We could read-out both spontaneous and evoked activities of a transgenic and wild-type mice neurons, as well as follow the hierarchy in fluorescence transient arrivals when mouse is given a train of electric stimulations with the new GOA (spike inference) algorithm.

We believe that our methods have the potential to identify neuronal types based on their fluorescence signatures. This could be an extension of the present work and would require analysis of a larger sample of experimental data.

Part II

Appendix

9 Abbreviations

AP	Action potential
MAS	Moving-average smoothing filter
S-Gf	Savitzky-Golay smoothing filter
MF	Median filter
AWGN	Additive white Gaussian noise
SNR	Signal to noise ratio
DDA	Deterministic deconvolution algorithm
GOA	Greedy optimization algorithm
PD	Parkinson's disease
MSE	Mean square error

10 Parameter Symbols

ξ	SNR of model signals
ω	Sliding-window size of the MAS filter
k	Sliding-window size of the S-Gf
γ_i	Difference between Ca^{2+} signals at $t = i$
α	Calcium kernel amplitude threshold scalar
θ	Threshold for MSE criterion
ϑ	Amplitude threshold scalar for θ
δ_h	Absolute kernel amplitude estimation error
δ_{τ_c}	Absolute clearance time constant estimation error
δ_{α_τ}	Absolute estimation difference due to different α
δ_{recon}	Spike train reconstruction error
σ_τ	Error term for estimated time constant

11 MATLAB scripts

Variable names employed in the scripts coincide with the names in section 5, 6 and 7; therefore, they are cross-referenced accordingly, when there is a need.

A. Model Ca^{2+} trace generator

Code name: Ca_model_signal.m

```
dh = [0.00,0.01,0.015,0.02];      %      Equation 5.11
dtau_c = [0.00,0.10,0.20];        %      Equation 5.11
xi = [1,2,3,4, 8, 16,32];         %      Equation 5.12
omega = [1, 2, 3, 4, 5, 7, 9,11];%      Equation 5.14

h0 = 0.044;                       % expected amplitude
tau0 = 0.5;                       % expected time constant
firing_rate = 0.31;               % firing rate
T = 200;                          % assumed trial duration
dt = 0.1;                         % sampling time step
t_vec = [0:dt:T];                 % time vector

% Construct a Poissonian spike train

eta = random('poiss',firing_rate
              *dt,length(t_vec),1);% Equation 5.3

% Generate a family of calcium traces (672 in total)

for h = dh;
   for tau_c = dtau_c;
        for SNR = xi;
             for MAS = omega;
                  Ca_trace = zeros((length(t_vec)+60),1);
                  firing_times = find(eta~=0);
                   for t = firing_times'
```

```
                    kappa_amp = (h0-h) + ((h0+h)
                                - (h0-h))
                                * rand(1,1);
                     kappa_tau = (tau0-tau_c)
                                 + ((tau0+tau_c)
                                  - (tau0-tau_c))
                                 * rand(1,1);

                     kappa_width = t_vec(1:max(find
                                 (t_vec<=6*kappa_tau)));
            %   Equation 5.4 with varying parameters
                     kappa = kappa_amp*exp(-kappa_width
                                          /kappa_tau)';

            % Convolve calcium kernels with the
                   Poissonian spike train
                     Ca_trace(t:(t+length(kappa)-1)) =
                             Ca_trace(t:(t+length(kappa)-1))
                             + eta(t)*kappa; % Equation 5.5

                 end

            % raw band-limited white Gaussian noise
              epsilon = kappa_amp/SNR*randn(size(t_vec'));

              epsilon_s = smooth(epsilon,MAS,
                                 'moving'); % Equation 5.13

              Ca_trace(length(eta)+1:end)=[];
              Ca_trace = Ca_trace + epsilon_s;% Ca model signal

            % To visualize traces while constructed
             figure(1); clf; plot((0:length(Ca_trace)
                                  -1),Ca_trace);
          end
       end
    end
end
```

B. Ca^{2+} trace amplitude estimator

Code name: h_estimate.m

```
%load Ca_trace

% differentiated Ca_trace (Equation 5.16)
df_Ca_trace = diff(Ca_trace);

% k is the smoothing window parameter in Equation 5.18
% k=11 is chosen for modeling & k=5 for application
% (see section 7 and 8 for  details on k)
SGf = smooth(df_Ca_trace,k,'sgolay'); % Equation 5.17

% Estimated average kernel amplitude
h_e = mean(df_Ca_trace(df_Ca_trace>max(SGf)));
```

C. Ca^{2+} clearance time constant estimator

Code name: estimate_tau_c.m

```
% load Ca_trace

h_estimate.m   % call the amplitude  estimator code


min_h_1 = alpha*h_e; % alpha in Equation 5.19
%We have used alpha=0.6 for the modeled data
%and alpha=1 for the experimetnal data

min_h_2 = alpha*min_h_1;

all_slopes = []; % A matrix to be filled by the trace
%decay slopes

% Find  peak values and location, 30 decay slope width
[min_peak, peak_locs] = findpeaks(Ca_trace,
      'minpeakheight',  min_h_1, 'minpeakdistance',30);

for j = 1:length(min_peak);

   if peak_locs(j) < length(Ca_trace) - 30;

% follow the trace 30 time steps after loc peak
      t_vec_decay = peak_locs(j)-1 : peak_locs(j)+30;

% exclude a localized peaks if there is another peak
% reaching (alph^2)*h_e within the 30 time steps
       min_peak_filt = findpeaks(Ca_trace(t_vec_decay),
                                 'minpeakheight',min_h_2);

        if length(min_peak_filt)==1
           %plot(Ca_trace(t_vec_decay)/min_peak(j));
```

```
% matrix containing all decay slops
          all_slopes = [all_slopes, Ca_trace(t_vec_decay)
                        /min_peak(j)];

      end
    end
end

% slopes average
av_slopes = mean(all_slopes(2:end-1,:),2);
%figure(10); plot(all_slopes(2:end-1,:));
hold; plot(av_slopes,'linewidth', 2); hold off;
avslope = av_slopes;

%  avoids from crashing avslope is empty
%(may happen for  very! noisy signals)

if not(isempty(avslope));
x =( t_vec_decay(2:end-1) - t_vec_decay(2))';

% Fits exp. function to avslope through the linear model
modelFun = @(p,x) p(1)*exp(p(2)*x); % exp. decay function
paramEstsLin = real([ones(size(x)),x]\log(avslope));
paramEstsLin(1)=exp(paramEstsLin(1));
xx = linspace(min(x), max(x));
avslope_Lin = modelFun(paramEstsLin, xx);
%Ltau_e = (-1/paramEstsLin(2)); %  linear fit estimate

% Improve linear fit ( Levenberg - Marquardt )

paramEsts = nlinfit(x, avslope, modelFun, paramEstsLin);
avslope_NLin = modelFun(paramEsts,xx);
tau_e = (-1/paramEsts(2))*0.1;

% calculate error term
[ParamEsts,r,J,COVNlfit,mse] = nlinfit(x, avslope,
                               modelFun, paramEstsLin);

% COVNlinfit is  the estimated covariance matrix
```

```
%for the fitted coefficients
Sigma_tau = sqrt(diag(COVNlfit(2,2))); % tau_e error term
end
```

D. DDA spike trains

Code name: DDA_spiketrain.m

```
% Run Ca_model_signal.m first or
% add its variables in Workspace

h_estimate      % call h_estimate.m
kappa_amp = h_e;

estim_tau_c     % call estim_tau_c.m
tau_c = tau_e;

% kernel in matrix form like in Equation 5.25
kappa_mat = spdiags(repmat(fliplr(kappa'),[length(t_vec) 1]),
            (-length(kappa')+1:0),length(t_vec),
            length(t_vec));

% DDA (Equation 5.26)
eta_dec=kappa_mat\Ca_trace;


eta_decrnd = max(0,round(eta_dec)); % Equation 5.27

delta_recon = sum(eta)-sum(eta_decrnd); % Equation 6.7
```

E. GOA spike trains

Code name: GOA_spiketrain.m

```
% Either call DDA_spiketrain.m or add variables in
% Workspace

vartheta; % Equation  6.8, we used
%for modeled data we used vartheta=0.2 and 0.5
%for experimental data
I = sum(eta_DDAr); % Maximum number of iterations
%see after Equation 6.8 or 5.31

cost_vec = NaN(size(eta_DDAr));
deta_vec = zeros(size(eta_DDAr));
for iter = 1:I
    Ca_traceDDA = kappa_mat*eta_DDAr; % Equation 5.28

% Calculate residue error
   Ca_trace_res = Ca_trace - Ca_traceDDA;

% MSE
   cost_best = mean((Ca_trace_res).^2); % Equation 5.30
   fprintf(1,'iter=%d: cost_best=%f\n',iter,cost_best);

% potentially "missing" spikes
   deta_vec(:) = 0;
   cost_vec(:) = NaN;
   for i = 1:length(deta_vec)
% Equation 5.31 and 6.8
      if abs(Ca_trace_res(i))> vartheta * kappa_amp %
        eta_GOA = eta_DDAr;
% Correct mistakenly forgotten spikes, Equation 5.31
        eta_GOA(i) = max(0,eta_GOA(i)
                    + 1*sign(Ca_trace_res(i)));
% Convolve the GOA spikes, Equation 5.28
        Ca_traceDDAop =   kappa_mat*eta_GOA;

% MSE after greedy optimization
        cost = mean((Ca_trace-Ca_traceGOA).^2);
```

```
        cost_vec(i) = cost; %
        deta_vec(i) = 1*sign(Ca_trace_res(i));

      end
    end
    [mv,mi] = min(cost_vec);
% Convergence criteria
    if ~isfinite(mv) || mv >= cost_best, break;
    end;
% Correct DDA spike estimation
    eta_DDAr(mi) = eta_DDAr(mi) + deta_vec(mi);
end
```

References

[1] 2010. 3.2

[2] R Bertram, JL Greenstein, R Hinch, E Pate, J Reisert, MJ Sanderson, TR Shannon, J Sneyd, and RL Wilson. *Tutorials in Mathematical Biosciences II: Mathematical Modeling of Calcium Dynamics and Signal Trasduction*. Springer, 2003. 2.4

[3] KM Biglan and B Ravina. Neuroprotection in parkinson's disease: An elusive goal. *Semin Neurol*, 27(2):106–112, 2007. 2.3

[4] Marc D Binder, Nobutaka Hirokawa, and Uwe Windhorst, editors. *Encyclopedia of Neuroscience*. Springer, 2009. 2.1, 2.2, 2.3, 3.2, 3.3.2

[5] JG Borst and F Helmchen. Calcium influx during an action potential. *methods enzymol*, (293):352–371, 1998. 2.4

[6] W Dauer and S Przedborski. Parkinson's disease: Mechanisms and models. *Neuron*, 39:889–909, 2003. 1, 2.3

[7] P Dayan and LF Abbott. *Theoretical Neuroscience: Computational and mathematical modeling of neural Systems*. MIT Press, 2001. 2.1, 2.2, 2.4

[8] A Devor, EMC Hillman, P Tian, C Waeber, and IC Teng. Stimulus-induced changes in blood flow and 2-deoxyglucose uptake dissociate in ipsilateral somatosensory cortex. *JNeurosci*, 23(53):14347–14357, 2008. 3.2

[9] Anna Devor. Projects: Laboratory of dr. anna devor. UCSD Neurosciences : http://nil.ucsd.edu/index.php?menu=project, 2011. 1

[10] Constantinos E Efstathiou. Signal smoothing algorithms. 5.2.1, 8.1.1

[11] Wulfram Gerstner and Werner Kistler. *Spikin Neuron Models: Single Neurons, Populations, Plasticity*. CAMBRIDGE, 2002. 2.1, 2.2, 2.2

[12] W Göbel and F Helmchen. In vivo calcium imaging of neural network function. *physiology*, 22(10.1152):358–365, 2007. 1, 2.4, 3.3.2, 4.3, 5.1.2, 5.1.2

[13] Gaston H Gonnet. Greedy algorithms for optimization: an example with synteny, January 2006. 5.4.2

[14] D S Greenberg, A R Houweling, and J N D Kerr. Population imaging of ongoing neuronal activity in the visual cortex of awake rats. *Nature Neurosci*, 11(7):749–751, June 2008. 1, 4.3

[15] B Greenstein and A Greenstein. *Color Atlas of Neuroscience, Neuroanatomy and Neurophysiology*. Thieme, 2000. 2.3

[16] F Helmchen. *Two-photon Functional Imaging of Neuronal Activity*, chapter 2, pages 37–55. In Vivo Optical Imaging of Brain Function. CRC Press, 2nd edition, 2009. 1, 3.1, 3.3.2

[17] F Helmchen and W Denk. Deep tissue two-photon microscopy. *Nature Methods 2, 932-940*, (2):932–940, 2005. 3.1

[18] F Helmchen and D Kleinfeld. In vivo measurements of blood flow and glial cell function with two-photon laser scanning microscopy. *Methods in enzymology*, 444:231–254, 2008. 1

[19] Ptrick R. Hof, Bruce D. Trapp, Jean de Vellis, Luz Claudio, and David R. Colman. *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience*, chapter 1. Elsevier Science, 2004. 2.1

[20] TF Holekamp, D turaga, and TE Holy. Fast three-dimensional fluorescence imaging of activity in neural populations by objective-coupled planar illumination microscopy. *Neuron*, 57(5):661–667, 2008. 1, 4.3

[21] R Homma, BJ Baker, L Jin, O Garaschuk, and A Konnerth. *Wide-Field and Two-Photon Imaging of Brain Activity with Voltage- and Calcium-Sensitive Dyes*, chapter 3. Dynamic Brain Imaging Multi-Modal Methods and In vivo Applications. Humana Press, 2009. 1, 3.1

[22] S M Hurtley. Astrocytes in alzheimer mice. *Sci. Signal.*, 2(ec82), 2009. 2.3

[23] EM Izhikevich. *Dynamical systems in Neuroscience: The geometry of Excitability and Bursting*. MIT Press, 2007. 2.1, 2.2, 2.4

[24] S Jacobson and EM Marcus. *Neuroanatomy for the Neuroscientist*. Springer, 2008. 1

[25] Erwin Kreyszig. *Advanced Engineering Mathematics*. Wiley, 2006. 5.1.3

[26] MP Mattson and T Magnus. Aging and neuronal vulnerability. *Nature*, 2006. 1, 2.3

[27] RJ Miller. Multiple calcium channels and neuronal function. *science*, 235(4784):46–52, 1987. 2.4

[28] M Oheim, DJ Michael, M Geisbauer, D Madsen, and RH Chow. Principles of two-photon excitation fluorescence microscopy and other nonlinear imaging approaches. *Advanced Drug Delivery Reviews*, 58:788–808, 2006. 3.1

[29] S J Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Englewood Cliffs,NJ, 1996. 8.1.1

[30] R Paschotta. *Two-photon fluorescence microscopy*. Encyclopedia. Wiley-VCH, Berlin, 2008. 3.1

[31] OH Peteresen, A Tepikin, and MK Park. The endoplasmic reticulum: one contiuous or several separate ca2+ stores? *Trends Neurosci*, 24:271–276, 2001. 2.4

[32] W K Pratt. *Digital Image Processing*. New York: John Wiley and Sons, 1991. 5.4.1

[33] P Ramdya, B Reiter, and F Engert. Reverse correlation of rapid calcium signals in the zebrafish optic tectum in vivo. *J Neurosci Methods*, 157(230-237), 2006. 4.3

[34] L Reznichenko. *Neurorescue Mechanisms of the Green Tea Polyphenol: Epigallocatechin-3-Gallate (egcg) in Cell Culture and Animal Models of Neurodegeneration: Combination of Neuroprotective Drugs*. PhD thesis, Israel Institue of Technology, 2008. 2.3

[35] T Sasaki, N Takahashi, N Matsuki, and Y Ikegaya. Fast and accurate detection of action potentials from somatic calcium fluctuations. *J Neurophysiol*, 100:1668–1676, 2008. 1, 4.3, 5.1.2

[36] D Smetters, A Majewska, and R Yuste. Detecting action potentials in neuronal populations with calcium imaging. *Methods*, 18:215–221, 1999. 4.3

[37] Julius O Smith. Spectral audio signal processing, october 2008 draft. 5.1.4

[38] Steven W Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub, 1st ed edition edition, 1997. 4.1, 4.1, 4.2, 5.4, 6.2.1

[39] Grace E. Stutzmann, Ian Smith, Antonella Caccamo, Salvatore Oddo, Frank M. LaFerla, and Ian Parker. Enhanced ryanodine receptor recruitment contributes to ca2+ disruptions in young, adult, and aged alzheimer's disease mice. *neuroscience*, 20(19):5180–5189, 2006. 2.4

[40] Inc The MatWorks. *MATLAB Help*, 1984-2009. 5.1.4, 5.2.2, 5.3, 5.4, 5.4.1, 6.1.3, 6.2.1

[41] Viartis. Parkinson's disease, december 2010. 2.3

[42] JT Vogelstein, AM Packer, TA Machado, T Sippy, B Babadi, R Yuste, and L Paninski. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *J Neurophysiol*, 104:3691–3704, 2010. 1, 4.3, 5.1.1, 5.1.2

[43] Eric W Weisstein. Least squares fitting, 2011. 5.2.2

[44] Emre Yaksi and Rainer W Friedrich. Reconstruction of firing rate changes across neuronal populations by temporally deconvolved ca2+ imaging. *Nature Methods*, 3(5):377–383, 2006. 1, 4, 4.3, 5.1.2