REPRODUCING KERNEL HILBERT SPACES FOR POINT PROCESSES, WITH APPLICATIONS TO NEURAL ACTIVITY ANALYSIS

By

ANTÓNIO R. C. PAIVA

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

To my family, for all their love and caring

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Dr. Jose C. Principe, for having accepted me as his student, and for his experienced guidance and advice. His incentive to creativity, breath of knowledge, and critical reaching thinking are, I believe, some of the most valuable lessons I will retain from my doctoral education. Without him, this dissertation would not have been possible.

I also thank Dr. John G. Harris, for serving as my committee member, his interest in my research, and providing an essential practical perspective to much of my work. I also thank Dr. Justin C. Sanchez for his valuable time to read and comment on many of the results shown here. His expertise on neural activity analysis and often complementary perspective can be encountered throughout this dissertation. I also thank Dr. Jianbo Gao for all the advice and interest in serving in my committee.

I am forever indebted to Dr. Francisco Vaz, for first creating the opportunity for me to come to CNEL and for all the help in obtaining funding from FCT. I will never forget that without Dr. Vaz's assistance, I would have missed the wonderful opportunity to get a Ph.D. at the University of Florida.

My friends and colleagues at CNEL deserve credit for many of the joys and for sharing this tortuous path to obtain a doctoral degree. In particular, I thank Il Park (a.k.a., Memming) for many of the contributions to this research, Dr. Yiwen Wang, Ayşegül Gündüz, Shalom Darmanjian, Weifeng Liu, and Dr. Hui Liu for all the fun moments and many discussions about research and life.

Last, but not least, I thank my family for their love and caring, and always being by side, supporting, and cheering me up when I felt down.

TABLE OF CONTENTS

page

APPENDIX

LIST OF FIGURES

Figure page

LIST OF TABLES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

REPRODUCING KERNEL HILBERT SPACES FOR POINT PROCESSES, WITH APPLICATIONS TO NEURAL ACTIVITY ANALYSIS

By

António R. C. Paiva

August 2008

Chair: José C. Príncipe
Major: Electrical and Computer Engineering

Point processes are stochastic random processes, yet a realization consists of a set of randomly distributed event locations. Hence, the peculiar nature of point process has made the application of conventional signal processing methods to their realizations difficult and imprecise to apply from first principles. Statistical descriptors have been extensively studied in the point process literature, and thus provide accurate and well founded methodologies to point process analysis by estimating the distributions necessary to characterize the process. But such methodologies face serious shortcomings when the interactions among multiples point processes need to be considered simultaneously, since they are only practical using an assumption of independence. Nevertheless, processing of multiple point processes is very important for practical applications, such as neural activity analysis, with the widespread use of multielectrode array techniques.

This dissertation presents a general framework based on reproducing kernel Hilbert spaces (RKHS) to mathematically describe and manipulate point processes. The main idea is the definition of inner products (or point process kernels) to allow signal processing with point process from basic principles while incorporating their statistical description. Moreover, because many inner products can be formulated, a particular definition can be crafted to best fit an application. These ideas are illustrated by the definition of a number of inner products for point processes. To further elicit the advantages of the RKHS framework, a family of these inner products, called the cross-intensity (CI) kernels, is

further analyzed in detail. This particular inner product family encapsulates the statistical description from conditional intensity functions of spike trains, therefore bridging the gap between statistical methodologies and the need for operators for signal processing. It is shown that these inner products establish a solid foundation with the necessary mathematical structure for signal processing with point processes. The simplest point process kernel in this family provides an interesting perspective to other works presented in the literature, since the kernel is closely related to cross-correlation.

These theoretical developments also have important practical implications, with several examples shown here. The RKHS framework is of high relevance to the practitioner since it allows the development of point process analysis tools, with the emphasis given here to spike train analysis. The relation between the simplest of the CI kernels and cross-correlation exposes the limitations of current methodologies, but also brings forth the possibility of using the more general CI kernels to cope with general point process models. From a signal processing perspective, since the RKHS is a vector space with an inner product, all the conventional signal processing algorithms that involve inner product computations can be immediately implemented in the RKHS. This is illustrated here for clustering and PCA, but many other applications are possible such as filtering.

CHAPTER 1
INTRODUCTION

1.1 General Motivation

A primal question in any work aspiring for relevance is why such work is worthy of attention. In this section we answer this question. Moreover, answering this question also prepares the reader to understand, and better appreciate, how the problem should be solved, which is done in the next section.

In a very broad sense, one might say that this dissertation was motivated by a desire to understand how the brain works. Or, more specifically, by the desire to understand the basic principles by which the brain represents and computes with information. Nevertheless, this is too broad of a question to tackle. More than simply trying to pose a philosophical question, or for the sake of the interest in fundamental neurophysiology and neuroscience, we were trying to solve an engineering challenge. The goal was do propose a framework for signal processing with neural activity which one could apply to design better (more accurate and reliable) brain-machine interfaces (BMIs).

Naturally, use of this framework for BMI work could greatly benefit from knowledge of the principles of information representation in the brain. More importantly, a framework for signal processing can provide the means to design the necessary tools to search for this understanding. Indeed, this mix of interests will be noticeable throughout.

What do we mean by neural activity in this work? Brain activity can be analyzed using many forms of neural recordings, namely: single-unit activity (SUA), local field potentials (LFP), electro-corticogram (ECoG), electro-encephalogram (EEG), magneto-encephalogram (MEG), just to mention the most commonly used. Each of these signals has specific properties, for example, in terms of spatial resolution and coverage, and signal-to-noise ratio. The general idea is that better properties of the recordings are typically obtained at the expense of greater invasiveness, which is of paramount importance in practical use. Table 1-1 reviews some of the properties.

Table 1-1. Common forms of neural activity recordings and their properties.

Recording	Invasive	Local resolution	Spatial coverage	Spectral range	SNR
SUA	yes	very high	localized	high-frequencies	high
LFP	yes	high	broad	broadband	high
ECoG	yes	high	broad	broadband	high
EEG	no	low	broad	low-frequencies	low
MEG	no	low	broad	low-frequencies	low

In this work only SUA will be considered. This is the most invasive method (together with LFP) with the need to introduce electrodes perforating the cortex. On the other hand, from an engineering perspective, SUA has the best properties, especially in terms of resolution and SNR, and therefore derived BMIs have the potential to achieve the best resolution. Furthermore, BMIs studies based in this form of recording also have the potential to deepen your understanding of how is information represented in the brain and should provide an upper bound on the achievable performance. Finally, this understanding can suggest how to improve the design of BMIs using less invasive neural activity recordings.

SUA-based BMIs are at the forefront of brain decoding for brain-machine interaction. This is understandable since, as stated, this form of recording has the best signal characteristics. However, unlike the other recordings, working with this form of recording presents a challenge of its own since SUA is a recording of the activity of one neuron, and neurons are known to communicate through electrical pulses, called *spikes*. Thus, information is represented not in a voltage waveform as usual but in sequences of spikes, or *spike trains*. The challenge is that spike trains must be modeled as realizations of point processes, for which the basic signal processing operators are not straightforwardly defined. This is the goal of this dissertation.

Notice that although BMIs were the motivation to start this work and are primal applications, they are not the focus of this dissertation. In fact, this work has a substantially broader impact, and for which BMIs are only one application. For example, this may be of great importance in other computation paradigms, such as in liquid state machines (LSM) studies [Maass et al., 2002], or spiking neural network (SNN) models which have recently emerged as a new artificial neural networks paradigm in a way that more closely mimics the brain [Maass and Bishop, 1998; Gerstner and Kistler, 2002]. Moreover, the impact of this work may even go beyond these applications, to whatever problem where processing or analysis of point processes is required.

1.2 Problem Statement

Before moving on, it is beneficial to try to understand the challenge tackled here and why processing with point processes is not as straightforward as for continuous- or discrete-valued random processes.

Point processes are stochastic random processes, yet a realization consists of a set of randomly distributed event locations. Put differently, for a point process the randomness is not contained in the amplitude[1] but when (or where) the event occurs. Consequently, upon observation of a realization of a point process one is not interested in the events themselves but on the mechanism/information underlying the generation of the events.

Point processes play a very important role in statistical modeling and inference in a wide variety of fields, such as: biology, engineering, geography, physics, astronomy [Snyder, 1975, Section 1.1 for application examples]. In general the event space of a point processes can be one-dimensional or multidimensional. However, here we shall deal exclusively with one-dimensional point processes. Often, the event space of one-dimensional point processes is time (as is the case for spike trains). For this reason, we will use "event locations" and "event times" interchangeably to refer to the coordinates of events.

[1] Actually, there are point process models, called *marked point processes*, for which there may be one or more random variables associated with the events. In this case, the amplitude of the event is a result of the randomness in these random variables. Nevertheless, these are a special class of point processes and are not considered in this work.

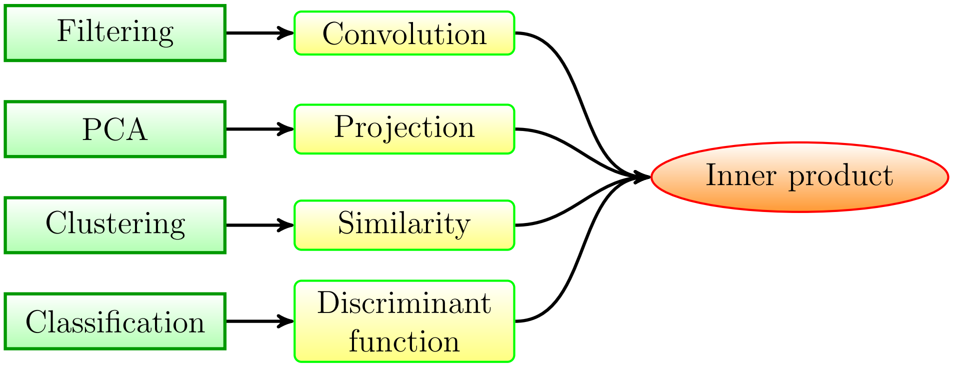


Figure 1-1. An inner product is an elementary operation in signal processing and pattern recognition.

Unfortunately, the peculiar formulation of point processes does not allow for the application of the usual signal processing operations to filter, eigendecompose, classify or cluster point processes or their realizations, which are essential to manipulating these signals and extract the information they convey. From a statistical perspective, point processes can be well characterized and many representations have been developed in the literature [Snyder, 1975; Daley and Vere-Jones, 1988]. Some of these representations and descriptors will be reviewed in Chapter 2. The main limitation of current statistical approaches is that point processes are analyzed independently, and independence need to be typically assumed to avoid handling the high dimensional joint distribution when multiple point processes are considered.

Before attending the question of how to do signal processing with point processes, let us first consider what is necessary for signal processing. For filtering, the output is the convolution of the input with an impulse response; for principal component analysis (PCA), one needs to be able to project the data; for clustering, the concept of similarity (or dissimilarity) between points is needed; and for classification, it is necessary to define discriminant functions that separate the classes. However, careful observation reveals that all of these needed concepts are either implemented directly by an inner product or can be constructed with an inner product. Convolution implements an inner product at each time instant between a shifted version of the input function and the systems' impulse response. Projection is inherently an inner product between two objects. An inner product is also a similarity measure, and dissimilarity measures (such as distances) can be defined given an inner product. Discriminant functions cannot be obtained directly with an inner product but, a neural network can be used to approximate it to the desired precision, with the linear projections in the PEs implemented by some given inner product. In summary, to obtain a general framework for signal processing and pattern recognition with point processes all that it is needed is an inner product definition operating with spike trains.

It must be remarked that it is possible to implement at least some of the aforementioned concepts without defining an inner product. For example, distances between point processes have been defined without explicitly defining an inner product (Section 3.6.3). However, such approaches have limited scope and do not provide a consistent and systematic mathematical framework to do signal processing, and tend to obscure the point process model associated with the operation.

1.3 Main Contributions

Based on the previous considerations, we can state that for signal processing with point processes all that is needed is an appropriate inner product. However, as before, defining an inner product of point processes is not straightforward, but the required mathematical structure follows once one is defined. For this reason, one of the main contributions of this dissertation it to suggest how inner products of point processes can be defined, estimated from realizations, and discuss some of their implications and applications.

Most of the considerations presented here regarding definitions of inner products for point processes are done under the formalism of *reproducing kernel Hilbert space* (RKHS) theory. Due to their equivalence this means that inner products will be defined as kernels.[2] The use of RKHS theory is done to assure that the necessary mathematical structure is well defined even in situations where the inner product is not explicitly defined, thus ensuring generality without sacrifice in rigor. Furthermore, operating with point processes in an RKHS is more convenient since several developments in signal processing and machine learning can be immediately incorporated. Therefore, this provides the framework for the development of a comprehensive set of algorithms for analysis and processing of point processes.

Although frequently overlooked, RKHS theory is a pivotal concept in statistical signal analysis and processing [Parzen, 1967], and machine learning [Schölkopf et al., 1999]. In RKHS theory, kernel operators denote inner product operations in a Hilbert space, which are fundamental for signal processing techniques, thus providing a strong motivation for the use of kernel functions. For instance, the cross-correlation function used throughout statistical analysis and signal processing, including the celebrated Wiener filter [Haykin, 2002], is a valid kernel and induces a RKHS space [Parzen, 1959]. In fact, most (if not all) of our understanding and ease of mathematical treatment of second-order methods can be obtained from the study of the RKHS induced by the cross-correlation.

In this dissertation, several kernels (that is, inner products) for operating with point processes shall be proposed. Notice that their corresponding RKHSs are automatically defined. Two main approaches will be followed. The first derives from ideas in kernel methods, whereas the second defines the inner product in the space of intensity functions of the point processes. Both may play an important role in method developments. We shall mainly focus on the second approach since the use of the conditional intensity functions permits the inner product to encapsulate a complete statistical characterization

[2] Throughout this dissertation we will often refer to ‘kernels’ and ‘inner products’ interchangeably. In our context, unless cautioned otherwise, they should always be understood as the same concept, although the former shall be often preferred since it makes explicit the connection to RKHS theory.

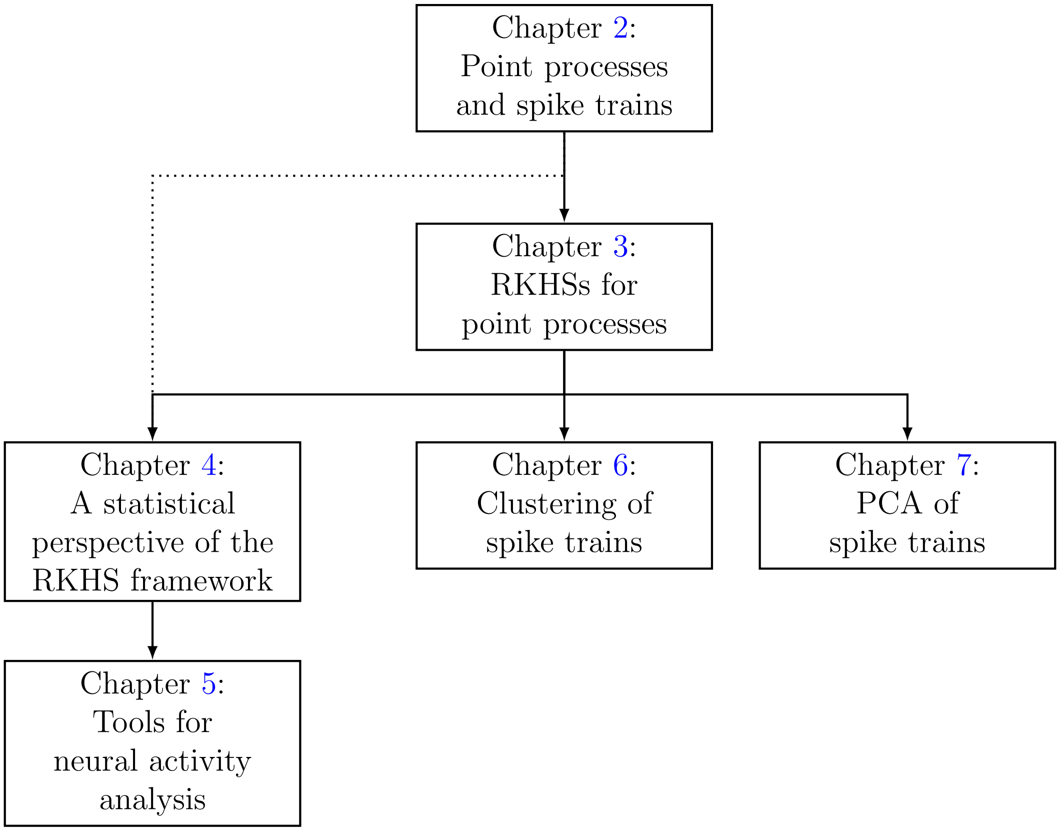


Figure 1-2. Outline of the dissertation.

of the point process, provides a better insight of the properties and limitations of the inner product as a descriptor of the point processes, and because, as will be shown, is closest related to current methodologies. The relevance of these concepts are exemplified in applications, where some of these inner products are utilized.

An important component of this dissertation is also the discussion of implications of this work which, by its generality, provides insightful perspectives in several methodologies described in the literature. Considerations for immediate implications in the state-of-the-art methods for spike train analysis are also presented.

1.4 Outline

This dissertation is organized in roughly four parts. The first comprises of Chapter 1 and Chapter 2 and provides the motivation, establishes the problem from an overall perspective, introduces point processes and spike trains, and reviews previous approches for their analysis. The second part, in Chapter 3, contains the main theoretical contributions and is where the kernels for point processes and the corresponding RKHS are defined and analyzed. The third part explores a more general definition of cross-correlation inspired by an RKHS construction in Chapter 4, and its multiple consequences in terms of new tools for the experimenter in Chapter 5 with several examples of application of these tools in both simulated and real datasets. This part is somewhat independent of the theory in Chapter 3, but in doing so the reader will miss the important connections to the general RKHS framework being presented. Finally, the fourth part shows two application examples of the RKHS framework for machine learning by showing how clustering algorithms for spike trains may be easily derived in Chapter 6, and by deriving from first principles the principal component analysis algorithm for spike trains.

Conclusions and discussion of this work are given in Chapter 8, along with a description of possible ideas for future developments on this work.

CHAPTER 2
INTRODUCTION TO POINT PROCESSES AND SPIKE TRAIN METHODS

In this chapter, we briefly introduce what point processes are and how they arise in a number of problems. This shall be done first in a somewhat informal way, broadly introducing the reader to historical problems that gave rise to the study of point processes in a review manner. Afterwards, the problem of how point processes arise in neurophysiology is discussed to aim on some of the important goals for this work. Then, many of the techniques specifically developed to analyze spike trains are presented, and we discuss the key strategies utilized to handle the particularities of point processes and some of the their limitations. This discussion will, hopefully, allow the reader to have a more general perspective and further appreciate some of the accomplishments of this work.

2.1 History of Point Process Theory

Here, a brief review of some of the historical developments in the theory of point processes is presented. This is done here for two reasons: to introduce the reader to some of the terminology and basic concepts in an informal way, and showcase some of the approaches developed earlier that are still utilized in statistical analysis of point processes. For a more detailed review the reader is referred, for example, to Daley and Vere-Jones [1988, Chapter 1].

Although point processes can be found in a relatively large number of problems, the primordial ideas and developments where been mainly associated with four areas of application, by chronological order:

- life tables and self-renewing aggregates;
- counting problems;
- communications theory; and
- particle physics and population processes.

The first two applications really motivated the initial developments in point process analysis and where developed in parallel with the fundamental ideas of probability (17th century), whereas the remaining two where raised in the previous century. Despite this separation and, as expected, it should be noticed from our presentation that the later topics where strongly affected by the earlier concepts and terminology.

2.1.1 Life Tables and Self-Renewing Aggregates

Life tables are records utilized in demographics studies of a population. Simply put, a life table lists the number of individuals from a population, or their ratio, that survive to a given age. The first known life table is due to John Graunt who in 1662 published the "*Observation on the London Bills of Mortality*" (available at [Graunt, 1662]). This table was analyzed at a later time by Huyghens (1629–1695) who proposed the notion of *expected length of life*. A second life table was constructed in 1693 by Halley using data from the smaller city of Breslau. Compared to Graunt's life table, this table was better since Halley did not have problems with disease, immigration and incomplete data that plagued Graunt's account.

Life tables occupied much of the field of statistics of that time, and was developed parallel to advances in probability theory. There are three basic descriptors (or summary statistics) of a life table: the relative frequency of individuals surviving to a given age or *survivor function*; the relative frequency of individuals that deceased between two ages, called *lifetime distribution function*; and the relative frequency of individuals that die after a certain age, the so-called *hazard function*. These concepts can be written informally in terms of probabilities as:

(i) *Survivor function*: $S(x) = \Pr\{\text{lifetime} > x\}$,

(ii) *Lifetime distribution function*:

$$f(x) = \lim_{dx \to 0} \frac{1}{dx} \Pr\{\text{lifetime terminates between } x \text{ and } x + dx\},$$

(iii) *Hazard function*:

$$q(x) = \lim_{dx \to 0} \frac{1}{dx} \Pr\{\text{lifetime terminates between } x \text{ and } x + dx | \text{lifetime} \geq x\}.$$

Actually these same concepts served as the root for the developments in probability theory by de Moivre, Euler and Laplace. However, it was not until Laplace's work, "*A Philosophical Essay on Probabilities*," that the previous concepts gained a more formal perspective in terms of probabilities. Indeed, although de Moivre had suggested that the survivor function would decrease with constant step for ages between 22 and 86, only after Laplace formal introduction of probabilities the three concepts where connected and more accurately through distributions.

The basic distribution function for lifetime has been the exponential function, $f(x) = \lambda e^{-\lambda x}$, $x > 0$, corresponding to a constant hazard function, $q(x) = \lambda$. That is, the probability of occurrence of an event is independent of previous events. A more accurate fit is usually found by the power-law hazard function with a constant added, $q(x) = B + Ae^{\alpha x}$ $(A > 0, B > 0, \alpha > 0)$, known as Gompertz-Makeham law. It is one of the most widely used functions for fitting a life table. Other commonly used distributions for lifetime modeling are:

- *Gamma*: $f(x) = \frac{\lambda\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}$,
- *Lognormal*: $f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\frac{1}{2\sigma^2}(\log x - \mu)^2}$.

Closely related to the study of life tables were problems in the study of statistical demography, growth, mortality tables and insurance. In the insurance context in particular, the importance of maintaining a stable "portfolio"; that is, a self-regenerating population of individuals, propelled the development of the theory of self-renewing aggregates. Simply put, this problem concerns the study of evolution of the human population and the balance in terms of number of births and deaths. A particularly relevant concept was the idea of *renewal density* characterizing the probability for the need of a replacement in time interval $[t, t + dt)$. In essence, these same ideas served as foundations for renewal theory.

2.1.2 Counting Problems

An alternative representation to statistically describe point process realizations is to count the number of events in intervals or regions of the event space. Unlike other approaches, counting is the only approach that lends itself to extension and systematic use in spaces with more than one dimension. The basic idea is to describe the point process in terms of the distribution of the number of events in a given region of the event space. Since the characterizing element if the "number of events" in the space, discrete distributions play a major role in the statistical analysis of point process under this perspective (even though the space is continuous).

The earliest references of application of a counting approach to point processes seem to be due to Seidel [1876] while studying the occurrence of thunderstorms, and Abbé [1879] which studied the number of blood cells in haemocytometer squares. Notice that these case studies dealt with point processes in two- and three-dimensional spaces, respectively, which justified the need for this approach.

The Poisson distribution is one of the best known examples of discrete distributions, and is particularly important in counting problems of point processes. In 1838, Poisson had included in his monograph, "*Recherches sur la probabilité des jugements en matières criminelles et matière civile*," the derivation of the Poisson distribution as the limit case of the binomial distribution as the interval length (or region volume) approaches zero. Interestingly, the works of Seidel and Abbé occurred after, and apparently in an independent manner, from Poisson's work. In fact, this is understandable since Poisson's result did not get wide attention at the time. Also, the fact that it was not derived in a counting process context may explain why it was unknown or neglected by Seidel and Abbé. Attention was only drawn to Poisson's distribution when in 1898, Von Bortkiewicz used the distribution to fit several phenomena in his monograph "*Das Gesetz der kleinen Zahlen*."

Several advances succeeded Poisson's work, namely on generalization and alternatives to the Poisson distribution. One notable generalization is the negative binomial distribution derived by Greenwood and Yule [1920] to fit accident statistics. However, the negative binomial distribution can be obtained from a mixed Poisson distribution, in which the rate parameter λ is a random variable with a gamma distribution.

2.1.3 Communications and Reliability Theory

Communications and reliability theory are two of the most important application areas of point processes in the past century. Reliability theory developed mainly after World War II and concerned the estimation of the lifetime of connected elements. Naturally, it absorbed much of the terminology and concepts derived earlier for the study of life tables. This application was propelled by the advances and growing industry in electronics in the post WWII period.

Communications theory and, in particular, queueing theory, was the second fundamental application of point processes to engineering, with the advent of telephone trunk lines. The landmark paper in the subject was published by Erlang [1909] on the study of the number of calls in a fixed time interval, for which Erlang derived a distribution. But, at that time, Erlang did not realize his finding corresponds to the Poisson distribution, only making this correction in Erlang [1917]. Actually, the distribution derived by Erlang is a *continuous* probability distribution whereas the Poisson distribution is discrete. Nevertheless, the Erlang distribution is a special case of the gamma distribution where the shape parameter is a natural number, and the gamma distribution had already been derived sometime earlier.

Another fundamental contribution to the field of queueing theory was Palm's thesis work in 1943 on the study of intensity variations in communications traffic [Palm, 1988]. In his work, Palm provided a detailed analysis of particular telephone trunking systems, but also the foundations of a general theory for point process with far reaching impact.

2.1.4 Density Generating Functions and Moment Densities

Point process theory also proved useful for the study of particle scatter in physics. Due to the high-dimensionality of the problem (more than 2 dimensions) the general approach to counting problems was used. However, instead of utilizing discrete distributions directly, the concepts of *generating functionals* and *moment densities* were employed since they provide a more convenient treatment of these problems. These concepts were first developed by Yvon [1935], a physicist looking to characterize the evolution of particle scatter distributions in experimental and theoretical physics studies.

These ideas are related to the *probability generating functional* defined by

$$G[h] = E\left\{\prod_i h(x_i)\right\} = E\left\{\exp\left[\int \log h(x) dN_x\right]\right\}, \tag{2–1}$$

where $h(x)$ is some test function, x_i are the event locations, and N_x is the counting measure. For a finite number of events the probability generating functional allows an expansion in terms of moment density functions (or *product densities*, as are perhaps more commonly known), characterizing the distribution of the number of events and the event locations.

One of the most important results in this regard was obtained by Ramakrishnan [1950], who first derived expressions for the moments of the number of events in terms of product densities and Stirling numbers. These ideas where later considerably extended by Ramakrishnan, Janossy and Srinivasan, among others, and applied to numerous physical problems, such as cosmic ray showers, for example. A review of this approach can be found in Srinivasan and Vijayakumar [2003].

2.1.5 Other Theoretical Developments

The work of Palm in 1943 [Palm, 1988] is one of the landmarks in the theory of point processes in the last century. Even though it had well defined practical context, it established the foundations for a general theory of point processes, and many of the current terminology. There are three major contributions in his work. First, the concept of *regeneration point* after which a point process (or system responsible for the point process) reverts to a given state and evolves independently of the past before the state corresponding to the regeneration point was achieved. Related to this idea, is the idea of a process *aftereffects*, which, simply put, describes the memory property of a process. Thus, Poisson processes are processes without aftereffects, and renewal processes have limited aftereffects. Second, that two distributions are important in describing stationary point processes: the distribution of the time to the next event from an arbitrary origin, and the distribution of the time to the next event from an arbitrary event. These distributions are related by the Palm-Khinchin equations. Third, a partial proof the limit theorem for point processes, which states that superposition of large number of independent point process tends to a Poisson process. Palm's work paved the way for developments by Wold [1948] on processes with Markov dependent intervals, which constitute the next alternative to renewal point processes, and by Khinchin [1960] who greatly extended and refined Palm's work.

The alternative approach was the study of point processes in term of probability measures on abstract spaces. This was motivated by the use of characteristic functionals proposed earlier by Kolmogorov to study random elements in linear spaces. This work allowed for studies on the convergence of measures on metric spaces (which also occur in point processes), and served as the basis for the developments mentioned earlier in generating functionals and moment densities.

Worthy of remark are also the works on the second half of the last century by Cox [1955] (Cox and Isham [1980] for a review) and Bartlett [1963]. These authors were responsible for developments in methods for statistical treatment of data generated by point processes. For example, Cox introduced the important class of *doubly stochastic Poisson processes*, important in the study of inhomogenous Poisson processes, and Bartlett illustrated theoretically how some methods of time series analysis could be adapted to the point process context.

One must must also note the contributions of Moyal [1962] who established the theory point processes in a general state space, providing the relations between product densities and probability generating functionals, as well pointing out several applications this theory.

2.2 Representations and Descriptors of Point Processes

As informally reviewed in the previous section, there are roughly four different approaches to represent and/or describe point processes:

1. Event densities and distributions;
2. Counting processes;
3. Random measures; and
4. Generating functionals and moment densities;

Needless to say that these representations are all closely inter-related and a description in a representation may be converted to another. These are briefly presented next.

2.2.1 Event Densities and Distributions

Point processes can be characterized in terms of the distributions needed to specified the statistics of its events. This is perhaps the most direct approach and relies on the same statistical principles utilize to quantify life tables (Section 2.1.1). Notice that in general multiples distributions may be needed to fully describe a point process.

The two most often used statistics are the density of events, called *rate function* or simply *intensity function*, and the inter-event interval distribution. These specify the expected number of events per space unit and the distribution of the difference between two adjacent events, respectively. The Poisson process is the simplest of the cases, for which the intensity function provide a complete description, and the inter-event interval is inherently specified as the exponential distribution, since this distribution is responsible for the memoryless property of the Poisson process. Another example are the renewal processes, which generalize the Poisson process to general inter-event interval distributions, and therefore both the intensity function and the inter-event interval distribution are needed to characterize the point process. But these two statistics do not suffice in general.

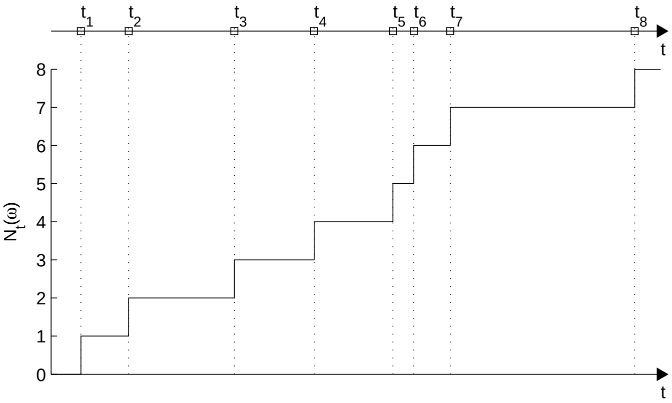


Figure 2-1. A realization of a point process and the corresponding counting process.

A compact description can be attained by using the *conditional intensity function* [Snyder, 1975, Pg. 238], denoted $\lambda(t|H_t)$, where $t \in \mathcal{T}$ ($\mathcal{T}$ is the space of events) and $H_t = \{t_1 < t_2 < \cdots < t_{N(t)}\}$ is the history of the process up to time t (the t_i's denote the event times or locations, with $t_{N(t)}$ denoting the most recent event before t). The conditional intensity function is defined as

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} \Pr[N(t + \Delta t) - N(t) = 1|H_t]. \tag{2–2}$$

For small Δt, $\lambda(t|H_t)\Delta t$ is the probability that an event occurs in the interval $[t, t + \Delta t)$. The conditional intensity function corresponds to the hazard function introduced in Section 2.1.1 since, in survival analysis, it expresses the probability of failure.

The conditional intensity function can be written in terms of the *event probability density function* $f(t|H_t)$ [Daley and Vere-Jones, 1988; Brown et al., 2001a], as

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - \int_{t_{N(t)}}^{t} f(u|H_t)du}. \tag{2–3}$$

Note that $f(t|H_t)$ is related to the distribution of the interval to the next event given all previous, which, as remarked by Palm [1988] (Section 2.1.5), is one of the fundamental distributions in describing a point process.

2.2.2 Counting Processes

Throughout the literature on point processes, the concept of counting process is prevalently used. This is understandable for the reasons presented in Section 2.1.2. The perspective provided by a counting process is easily interpretable and tractable using only the theory of discrete distributions, and is extendible to multidimensional point processes in a systematic manner.

The function $N_t(\omega)$, $t \in \mathcal{T}$, is a *counting process* and is defined as the number of events up to location t for the realization (Figure 2-1). For each $\omega \in \Omega$, $N_t(\omega)$ is a piecewise-constant function of t with unit jumps at the event coordinates. However, notice that, like stochastic random processes, $N_t(\cdot)$ is a functional representation which becomes a well defined function of t only for a fixed ω; that is, a given realization.

A counting process is an attractive formulation also because the derivative of its expectation over ω at a given t may be interpreted as the density of events. Therefore, it provides a way to map the space of events to a density of events, providing an equivalent representation of the statistical distributions mentioned in the previous section without requiring their form explicitly.

2.2.3 Random Probability Measures

Random measures are another way to express point processes. Let the space of events, $\mathcal{T}$, be a locally compact Hausdorff space with a Borel σ-algebra, and $\mathfrak{N}$ the set of locally finite counting measures on $\mathcal{T}$ with σ-algebra $\mathcal{N}$. Then, a point process on $\mathcal{T}$ is a measurable map $\xi : \Omega \leftarrow \mathfrak{N}$, from a probability space $(\Omega, \mathcal{B}, P)$ to the measurable space $(\mathfrak{N}, \mathcal{N})$. That means that for any set $S \in \mathcal{T}$, $\xi(S)$ is a random variable corresponding to the number of events in S.

Typically, the point process random measure is written as

$$\xi(\cdot) = \sum_{n=1}^{N} \delta_{t_n}(\cdot), \tag{2–4}$$

where δ denotes the Dirac measure,

$$\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A, \end{cases} \tag{2–5}$$

N is an integer-valued random variable, and t_n are the events in $\mathcal{T}$.

In essence, random measures formalize mathematically the concepts utilized to build the counting processes. For this reason, counting processes are sometimes called counting measures in the literature.

2.2.4 Generating Functionals and Moment Densities

As mentioned earlier, the foundations for handling stochastic populations of particles had already been set before by Yvon [1935], but there where gaps in the theory. The problem was that of statistically describing a point process characterized by a finite set of points or events, say $X = \{x_1, \ldots, x_N\}$, in a state space χ. A simple probability description can be obtained in terms of the probability mass function (pmf) for the total number of points in the realization, $P_r = \Pr[N = r]$. The pmf can then be utilized to write the joint distribution over a state space of realizations, Π_r, which in turn is specified in terms of the densities $f_r(x_1, \ldots, x_r)$, with properties:

$$\begin{aligned} &\Pr[N(dx) = 1] = f_1(x)dx + o(dx), \\ &\Pr[N(dx) > 1] = o(dx), \\ &\Pr[N(dx) = 0] = 1 - f_1(x)dx + o(d). \end{aligned} \tag{2–6}$$

It must be noted that the density f_1 is *not* a probability density function (pdf). Rather, it is called a *product density function*, to distinguish it from a pdf.

These can be utilized to write the *probability generating functional* in the form,

$$\begin{aligned} G[\zeta] &= E\left\{\prod_{i=1}^{N} \zeta(x_i)\right\} \\ &= P_0 + \sum_{r=1}^{\infty} P_r \int_{\chi^r} \zeta_1(x_1)\ldots\zeta_r(x_r)f_r(x_1, \ldots, x_r)dx_1 \ldots dx_r, \end{aligned} \tag{2–7}$$

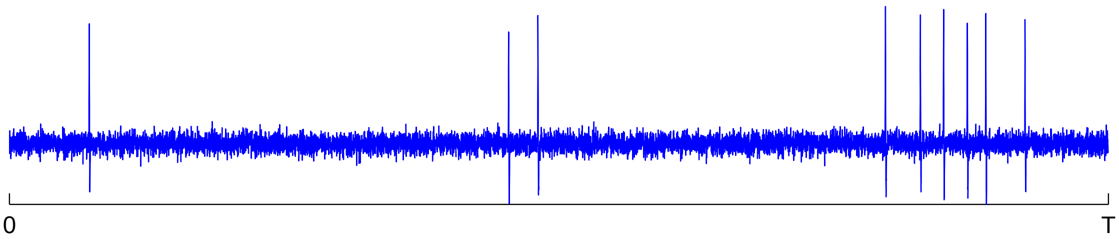


Figure 2-2. An example of a single-neuron extracellular voltage recording showing a few action potentials.

or, equivalently, as

$$G[1+\eta] = 1 + \sum_{k=1}^{\infty} \frac{1}{k!} \int_{\chi^k} \eta_1(x_1)\dots\eta_r(x_k) m_k(x_1,\dots,x_k) dx_1 \dots dx_k, \qquad (2\text{–}8)$$

where the m_k's are the product densities, with

$$m_k(x_1,\dots,x_k)dx_1\dots dx_k \approx E\left\{N(dx_1)\dots N(dx_k)\right\}. \qquad (2\text{–}9)$$

In this regard, Ramakrishnan [1950] was the first to derive explicit expression for the *factorial* moments in terms of product densities and Stirling numbers. For the mth moment it was obtained

$$E\left\{N(dx)^m\right\} = \sum_{k=1}^{m} C_k^m \int_{\chi^k} f_k(x_1,\dots,x_k)dx_1\dots dx_k, \qquad (2\text{–}10)$$

where the weight coefficients C_s^m are the Sterling numbers.

2.3 Spike Trains as Realizations of Point Processes

In neurophysiology it is widely accepted that the fundamental processing units of the brain — the neuron cell — communicate through a discrete pulse-like wave of voltage, called an *action potential* [Dayan and Abbott, 2001]. A neuron receives the action potentials in its, typically, large number of input synapses and produces an output of the same form in the axon, even though internally to the cell membrane the action potential is converted into an analog potential change.

Action potentials, being electrical pulses, can be captured in single-neuron voltage recordings (Figure 2-2), which record the voltage differential to a distant "ground" point, typically the skull. Despite the inevitable noise in these recordings, it is easily observed that action potentials have a very stereotypical shape, with a characteristic fixed amplitude and width associated with a given neuron. That is to say that, from a neurophysiological perspective, the actual shape and magnitude of an action potential is redundant because it contains no information, only the moment it occurs. As explained earlier this kind of phenomena is best described by a point process model. Under this perspective action potentials are simply called "spikes," and these events are responsible for transmitting the information in and out of the neuron only through their occurrence. Correspondingly, a sequence of spikes ordered in time is termed a *spike train*. Since spike trains always correspond to an observation or measurement associated with some underlying point process they are considered to be realizations of a point process from which only the number of spikes and the moments they occur are relevant.

2.4 Analysis and Processing of Spike Trains

Since analysis and processing of spike trains is the primal motivation for this work, for completeness, this section briefly reviews many of the established approaches and methods utilized in their study.

2.4.1 Intensity Estimation

Intensity function estimation is one of the most fundamental problems in spike trains analysis, since an intensity function is a fundamental descriptor of the underlying point process. There are basically three approaches for intensity function of a spike train:

1. Binning;
2. Kernel smoothing; and
3. Nonparametric regression with splines.

2.4.1.1 Binning

Binning is the predominant approach in current spike train analysis and processing methods [Dayan and Abbott, 2001]. Statistically, it is motivated by the counting process representation of a point process. Basically, the binned spike train is obtained

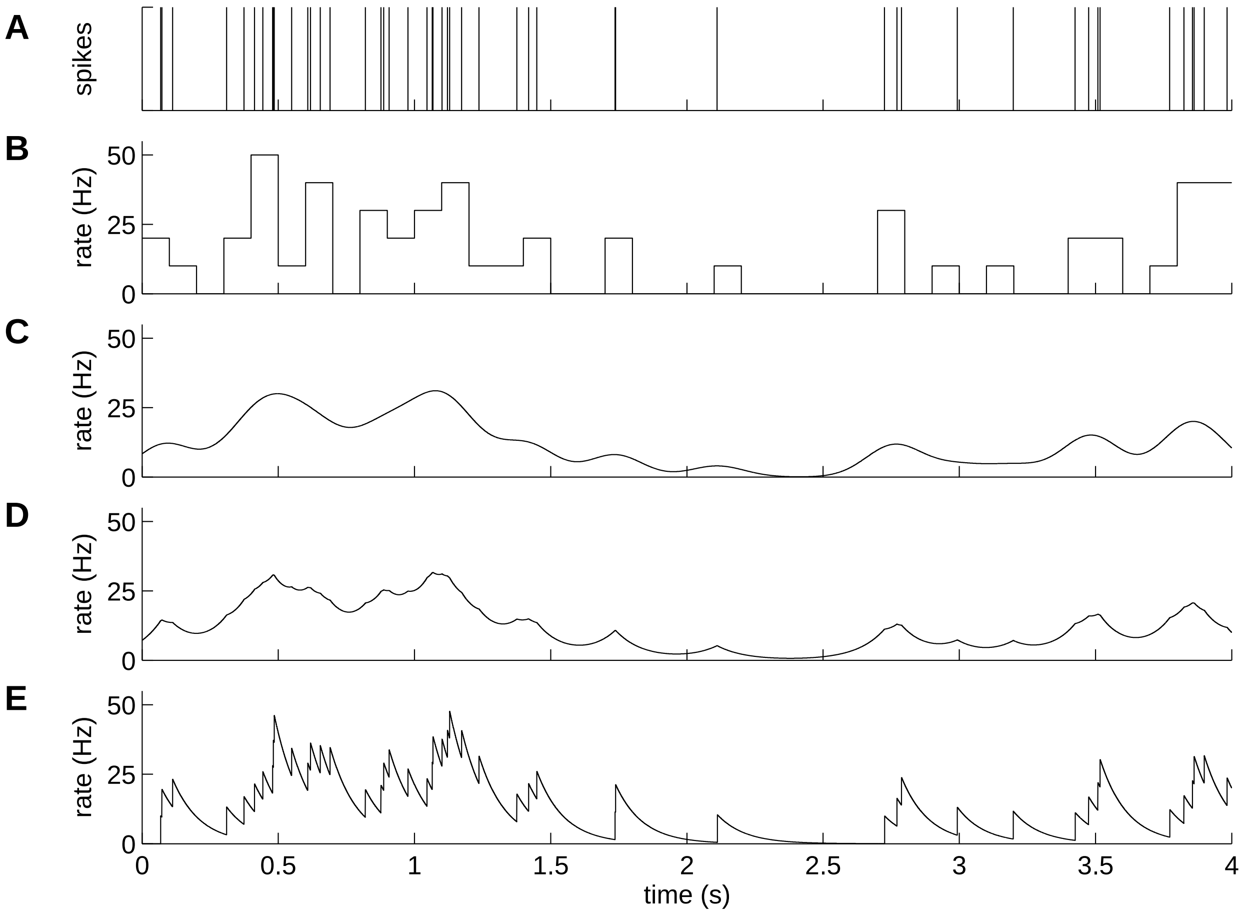


Figure 2-3. Estimation of the intensity function by different procedures. (A) Spike train to estimate the firing rate. (B) Normalized binned spike train, with bin size $\Delta t = 100$ms. (C)–(E) Estimated firing rate by kernel smoothing, for the Gaussian function, Laplacian function and exponential decay function, respectively. The kernel size parameter was 100ms for all three smoothing functions.

by discretizing time and assigning the number of spikes occurring in the time quantization interval (i.e., the bin) to the time instant. If the bin size is large compared to the average inter-spike interval the transformation provides a crude yet effective estimate of the instantaneous firing rate. For consistent intensity function estimation, the binned data is normalized by the bin size (i.e., the size of the quantization interval), although this last step is often skipped.

From a signal processing perspective, binning is a transformation which maps the randomness in the spike train continuous time structure to randomness in the amplitude of the intensity estimation. The use of binning has clear advantages in terms of intuitive understanding and ease of practical use, and the transformation of point processes into discrete random processes allows for the wealth of conventional statistical time series analysis and processing methods to be used. On the other hand, the discretization of time affects the resolution in time of any analysis subsequently performed to the resulting signal. This means that any temporal information in the spikes within and between bins is disregarded, limiting the type of analysis that can be subsequently done. This is especially alarming for neurophysiology use when a number of recent studies suggest that neurons spike timing precision is on the sub-millisecond range [Wagner et al., 2005; Carr and Konishi, 1990], and the actual spike times encode additional information [Hatsopoulos et al., 1998; Vaadia et al., 1995; Mainen and Sejnowski, 1995]. Moreover, a reminiscent problem is at what time-scale to analyze the data. That is, what bin size to choose? Notice that the hard-limiting nature of the rectangular window used make this an even harder task.

2.4.1.2 Kernel smoothing

Kernel smoothing is another approach to intensity estimation, and seems the method of choice in the point processes literature. The main advantage compared to binning is that the precision in the event location is preserved and fully incorporated in the intensity estimation.

Of course, there are other ways to estimate the intensity function of a point process, mainly through smoothing. That is, by convolving some smooth function with the spike train (seen as a sum of time-shifted impulses). Figure 2-3 illustrates some of these methods. More elaborated methods included Bayesian and spline fitting to normalized binned data [Kass et al., 2003; Ventura et al., 2002]. In any case, the improvements in terms of resolution and/or in the estimation of the intensity function these methods might provide are made at the expense of much higher computation complexity. In addition, like for the binned spike trains, any further computation is done without a clear understanding of the mathematical and statistical properties of the space. A realization of a point process can be interpreted as a signal in a continuous parameter space composed as a sum of impulses centered at the event coordinates. Thus, spike trains can be written as,

$$s(t) = \sum_{m=1}^{N} \delta(t - t_m), \tag{2–11}$$

where N is the number of spikes and t_m the spike times in the recording interval, and $\delta(\cdot)$ denotes the Dirac delta. Then, the estimated intensity function is obtained by simply convolving $s(t)$ with the smoothing kernel h, yielding

$$\hat{\lambda}(t) = \sum_{m=1}^{N} h(t - t_m^i). \tag{2–12}$$

Notice that the smoothing function must integrate to 1 so that the estimated intensity function is consistent (integral of the estimated intensity function must equal the number of spikes).

It should be remarked that binning can be posed in terms of kernel smoothing. Specifically, binning of spike trains can be put as a two step procedure:

1. Quantize the spike times to a precision of $\Delta t/2$, where Δt is the bin size;
2. Convolve the sequence of time-shifted impulses centered at the quantized spike times with a rectangular window of width Δt.

This view makes it clear the time discretization in binning. The two approaches are illustrated in Figure 2-3 for several smoothing functions.

2.4.1.3 Nonparametric regression with splines

A recently proposed method for intensity estimation is nonparametric regression with splines [Ventura et al., 2002; Kass et al., 2003]. The basic premise in the use of splines is the smoothness of the intensity functions, which is translated into constraints with regards to which the optimization algorithm finds the estimated intensity function as a weighted combination of splines. In particular, in Kass et al. [2003] the Bayesian adaptive regression splines (BARS) method was utilized since it automatically finds the "knots" where the splines are joined together, thus rendering the methods basically parameter-free. Although this method does not require the choice of bin or kernel size, unlike the previous two approaches, it can only be utilized in offline studies and it has much higher computational complexity due to the Monte Carlo optimization approach utilized by BARS.

Instead of being applied to the data directly spline smoothing can, alternatively, be applied to the binned spike train to smooth the estimated intensity function. Indeed, one of the most important conclusions by Kass et al. [2003] is the much greater data efficiency in intensity estimation by smoothing.

2.4.1.4 Trial averaging

An approach often employed in conjunction with either of the previous methods is trial averaging. This means that if multiple realizations of the experimental trial are available, and stationary is assumes between trials, then one can average the estimated intensity function across trials for improved statistical robustness. The widely used *peri-stimulus time histogram* (PSTH) is an example of trial averaged intensity estimation using binning.

To implement trial averaging the spike trains are first time aligned with respect to the time a stimulus is applied, and then one can first estimate the intensity function for each trial and then average over trials or, conversely, condense the spike trains of all trials together and estimate the intensity function attending for the normalization by the number of trials. One of the advantages of trial averaging is that, from the limit theorem for point processes, the combined spike trains approach a realization of a Poisson process, even if the true underlying point process contains history. Put differently, the estimated intensity function is more likely reflect the true instantaneous firing rate, as intuitively expected.

On the other hand, the main difficulty with trial averaging is the assumption of stationarity between trials. That it, is assumes that the process giving rise to the spike train did not change. However, given the many factors the influence the brain activity (memory, learning, plasticity, attention focus, etc.) this is often a difficult assumption to justify, especially when the number of trials is large.

2.4.2 Methods for Spike Train Analysis

Considering only one neuron in the brain at a time, the information it conveys is expressed through changes in the firing pattern (rate or temporal precision). In this case, one needs to measure the statistics of the observed spike train to infer the neuronal state. The intensity function captures the neuron instantaneous firing rate is therefore of great importance. Any of the methods described in the previous section can be utilized but, as mentioned, binning is the most common method. The peri-stimulus time histogram (PSTH) is often used for spike train analysis [Perkel et al., 1967a; Gerstein and Aertsen, 1985].[1] The PSTH is particularly useful to study and verify the presence of modulations in the neural activity (for example, in terms of the firing rate) with regards to a time-locking stimulus.

Neurons in the brain greatly interact with neighboring neurons through there many (on the order of thousands) synaptic connections. However the previous approach, although well suited statistically, does not scale properly to the simultaneous analysis of multiple spike trains. For this, independence is habitually assumed. Consequently, how to find and measure association or couplings between neurons is another major problem for which several methods have been proposed. The cross-correlation [Perkel et al., 1967b; Dayan and Abbott, 2001] is probably the most widely used technique to measure interactions between two spike trains.

[1] Sometimes the peri-event time histogram (PETH) is mentioned in the literature. Conceptually, the PSTH and PETH are the same thing, although the time mark for alignment of the spike trains is general in the latter case.

If N_A and N_B denotes the binned spike trains, the cross-correlation (or correlogram) is defined as,

$$R_{A,B}[l] = E\left\{N_A[n]N_B[n-l]\right\} \simeq \frac{1}{M}\sum_{n=l}^{M} N_A[n]N_B[n-l]. \qquad (2\text{–}13)$$

where M is the total number of bins and $N_A[n]$, $N_B[n]$ are the number of spikes in the nth bin, respectively.

Cross-correlation as a statistical measure of similarity between spike trains was "imported" from random processes and currently can only be applied to the binned spike trains. Moreover, the expectation implies averaging over time which limits its usefulness as a descriptor of the evolution of correlation as a function of time, and intrinsically requires stationarity and ergodicity over the averaging time interval. To address non-stationary, cross-correlation is averaged over small windows of time which further reduce the time resolution at the sacrifice of statistical reliability.

The limited temporal resolution of cross-correlation lead to the use of other methods. The JPSTH Gerstein and Perkel [1969]; Aertsen et al. [1989]; Gerstein and Perkel [1972] is another widely used tool to characterize the evolution of synchrony over time between two neurons. The fundamental idea is a smoothed two-dimensional scatter diagram of the neuronal firings from one neuron with respect to the other, and time-locked to a stimulus. Although the averaging over time in the JPSTH is removed (apart from smoothing), and thus provides more detailed information about time-dependent cross-correlation with respect to the stimulus, this approach requires trial averaging. Therefore, one needs to assume stationarity between trials which, for the same reasons given previously (Section 2.4.1.4), is an unrealistic assumption. Furthermore, the approach rapidly becomes unmanageable for more than just a few neurons since the analysis is does in pairs (e.g., 16 neurons requires 120 JPSTH plots). The joint interval histogram (JIH) [Rodieck et al., 1962] is a similar tool to identify correlations between inter-spike intervals for which similar considerations may be made.

Other spike train analysis methods in the time domain include unitary events [Grün et al., 2002a,b] and the gravity transform [Gerstein et al., 1985; Gerstein and Aertsen, 1985]. Unitary events is a statistical method to detect coincident spike activity above chance. It does so by comparing the number of coincident spikes with the expected number by chance for the estimated "local" firing rate. However, like other methods, it is sensitive to binning and employs a large moving window analysis for statistical reliability. The gravity transform tackles some of these problems. Mainly because it does not require binning and provides a way to visualize the evolution of synchrony over time. However, it lacks a statistical baseline which limits the knowledge that can be inferred from the analysis.

Several methods for analysis in the frequency-domain have also been proposed. For instance, the partial directed coherence (PDC) [Baccalá and Sameshima, 1999; Sameshima and Baccalá, 1999], and the method by Hurtado et al. [2004]. PDC employs multivariate time series analysis to together with the ideas behind the Granger causality concept to infer inter-dependencies between neurons. But, due to the transformation into the frequency domain, these methods operate over windows of data. Therefore, they require stationarity for the analysis to be valid, and the time resolution is reduced as a consequence.

2.4.3 Processing of Spike Trains

Processing of spike trains is of great interest from a neurophysiological perspective but even more important from an engineering point of view. Mainly because of the tremendous implications for neural prostheses, and in particular for the applications in brain-machine interfaces (BMI).

In the recent years computation with artificial neural networks of spiking neurons has also emerged as an engineering application where tools to do signal processing with spike trains are naturally of great importance. An example is the emerging concept of liquid state machines (LSM) proposed by Maass et al. [2002]. LSMs use the principles of high-dimensional dynamical systems to perform computation with recurrent neural networks of spiking neurons. Another example are spiking neural networks (SNN) currently being studied and applied to large number of problems. Using processing elements that more closely resemble the actual neurons, these networks extend the computation paradigm of artificial neural networks in a way that more closely mimics the brain [Maass and Bishop, 1998; Gerstner and Kistler, 2002].

There are four main approaches currently utilized for processing of spike trains:

1. Linear/nonlinear models;
2. Probabilistic models;
3. State space models; and
4. Volterra/Wiener models.

The literature on these methods will now be quickly reviewed. From the review it will be clearly shown that, as said above, processing of spike trains has been largely motivated and applied to neural prostheses, which is also the (long-term) motivation for this work. In spite of that, we hope that the reader may realize the wide implications of this work beyond this realm of problems.

2.4.3.1 Linear/nonlinear models

These models are the most direct approach towards processing of spike trains, since it utilizes current time series processing techniques. So that these models can be directly applicable the spike train must be transformed into a discrete-time signal, and the standard approach is to utilize binning, since as explained earlier binning implements this mapping. It must be remarked that these cases are predicated on the idea that the information to be extracted is encoded in modulations of the firing rates [Nicolelis, 2003]. Indeed, most results reported in the literature utilize binned spike trains with bin size $\sim$ 100ms, corresponding to firing rate estimation. These models have proven particular important in BMIs since the output is a discrete-time signal of the movement variables.

At the current stage of research, in most of the BMI experimental paradigms the desired response (intended movement) is available. Therefore, these paradigms lend themselves to supervised learning, where the problem is well formulated as a system modeling task. The Wiener filter [Haykin, 2002] is the linear model typically employed, having been extensively utilized in many studies in the literature [Chapin et al., 1999; Wessberg et al., 2000; Carmena et al., 2003]. For nonlinear modeling, neural networks have been used. See Kim et al. [2006]; Sanchez et al. [2003]; Kim [2005]; Sanchez [2004] for a comparison of methods.

It is important to remark that, because for the scenarios envisioned for BMIs the desired response will not be available, some attempts have also been made to move towards the use of unsupervised learning models [Darmanjian et al., 2007].

2.4.3.2 Probabilistic models

Probabilistic models attempt to interpret spike trains content from some probabilistic model, often specific to a given task.

The work by Georgopoulos et al. [1982, 1988] represents a landmark in spike train decoding, when Georgopoulos suggested the concept of *population coding*. Georgopoulos showed in a center-out task that if each neuron is a assigned a "tunning curve," basically denoting the distribution of the movement angle as a function of the neuron's firing rate, and by averaging across a population of neurons a high precision is attained. Perhaps the most important contribution of this work was to provide evidence for the importance of a group of neurons in conveying information in a reliable and effective manner.

Another probabilistic worthy of remark is the Bayesian approach by Shenoy's group (see, for example, Shenoy et al. [2003]). Based on the specific experimental paradigm, a state space model with transitions decoded by maximum a posterior probability, was proposed. This implied the estimation of the marginal distributions for each neuron from data. These distributions were then combined using Bayes' theorem under an independence assumption.

In either of these approaches independence among neurons needs to be assumed. As we had remarked earlier, this is one of the major limitations of statistical methods.

2.4.3.3 State space models

An alternative to the above methods is to utilize state space models together with sequential estimation. These approaches make use of Bayesian tracking to probabilistically infer the evolution of a state sequence over time. Roughly speaking, this is similar to the idea of hidden Markov models (HMMs) but for a continuous state space.

The simplest example of this methodology is the use of Kalman filtering applied to the binned spike trains [Wu et al., 2004]. Kalman filtering applied to spike train processing has numerous limitations: both the model describing the evolution through the state space and the readout are linear, all distributions are assumed to be Gaussian, and is applied to binned spike trains only. Under similar assumptions but applied to the spike trains directly[2] has been also proposed by several groups Eden et al. [2004]; Brown et al. [2001b]. Notice that in the latter the forward (encoding) model is needed and has been assumed to be Gaussian.

To avoid these assumptions, in recent studies particle filters have been used which allow for arbitrary forward models, non-linear evolution through the state space and non-Gaussian distributions. Particle filter creates a probabilistic state space model for the decoder which is recursively and continuously adapted through a Bayesian approach based on the latest observation. However, update of the probabilistic model used Monte Carlo sequential estimation which is not only extremely computational intensive due to the random sampling of the space, but also requires a priori knowledge of properties of the neurons being measured.

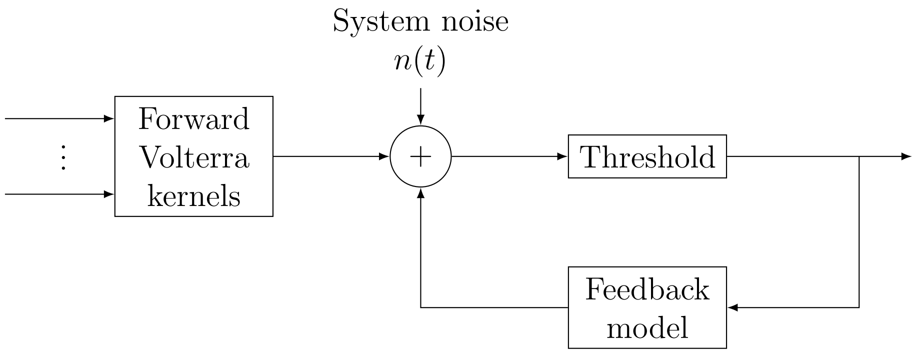


Figure 2-4. Diagram of the Volterra/Wiener model for system identification. A MISO (multiple-input single-output) configuration is shown. For MIMO configurations, basically, the MISO structure is repeated for each output.

2.4.3.4 Volterra and Wiener models

Sometimes the spike train analysis or signal processing problem at hands can formulated as a system identification task. Since the brain is known to be highly nonlinear then a model with nonlinear modeling ability needs to be utilized, such as a neural network. However, if the output is to be a spike train then the Volterra/Wiener models have been utilized [Marmarelis, 2004, 1993; Song et al., 2007]. This is particularly important for the study of specific neural systems by estimating the input-output model from recorded spike trains, or in neural prostheses aiming to replace or aid the functioning of a failing neural structure. Figure 2-4 depicts the architecture of the Volterra/Wiener model.

[2] More correctly said, the sequential methods work with a binary representation of the spike train, equivalent to binning with a very small bin size ($\sim$ 1ms), which corresponds to a Bernoulli random process.

A time-invariant system can be expressed in terms of the Volterra series as

$$
\begin{aligned}
y(t) = h_0 &+ \int_{\mathbb{R}} h_1(\tau_1)x(t-\tau_1)d\tau_1 \\
&+ \int_{\mathbb{R}^2} h_2(\tau_1,\tau_2)x(t-\tau_1)x(t-\tau_2)d\tau_1 d\tau_2 \\
&+ \int_{\mathbb{R}^3} h_3(\tau_1,\tau_2,\tau_3)x(t-\tau_1)x(t-\tau_2)x(t-\tau_3)d\tau_1 d\tau_2 d\tau_3 \\
&+ \cdots \\
= &\sum_{n=0}^{\infty} \int_{\mathbb{R}^n} h_n(\tau_1,\ldots,\tau_n)x(t-\tau_1)\cdots x(t-\tau_n)d\tau_1\cdots d\tau_3 \\
= &\sum_{n=0}^{\infty} H_n[x(t)],
\end{aligned}
\tag{2–14}
$$

where $H_0[x(t)] = h_0$ and

$$
H_n[x(t)] = \int_{\mathbb{R}^n} h_n(\tau_1,\ldots,\tau_n)x(t-\tau_1)\cdots x(t-\tau_n)d\tau_1\ldots d\tau_n \tag{2–15}
$$

is the nth order Volterra functional. One can think of the Volterra series as a Taylor series with memory. The functions h_n are called the Volterra kernels of the system and are causal; that is, $h_n(\tau_1,\ldots,\tau_n) = 0$, if any $\tau_i < 0$, $i = 1,\ldots,n$. In general, these kernels are not unique for a given output. However, if symmetry is imposed with respect to permutations of the τ_i, that is, if $h_n(\ldots,\tau_i,\ldots,\tau_j,\ldots) = h_n(\ldots,\tau_j,\ldots,\tau_i,\ldots)$, for all $i,j = 1,\ldots,n$, then it can be shown that the Volterra series expansion is unique.

In the Volterra series the output of two distinct functionals is not, in general, orthogonal (i.e., uncorrelated). However in the Wiener series the functionals form a *complete orthonormal basis*. In terms of the Wiener series the system can be expressed as

$$
y(t) = \sum_{n=0}^{\infty} G_n[x(t)], \tag{2–16}
$$

where $G_n[x(t)]$ are the Wiener functionals. The characterizing feature of the Wiener functionals is their orthonormality for zero-mean white Gaussian distributed input.

The Wiener and Volterra series are two equivalent approaches to characterize a system. However, the orthogonality of the basis functionals can be used to “isolate” each of the term in the series, this facilitating the estimation of the corresponding kernel. In fact, if the input is zero-mean white Gaussian distributed, the leading Wiener kernels can be obtained directly by cross-correlations between the input (at various lags) and output. Moreover, there is a mathematical relation between the functionals of the two representations. For neurophysiological studies, however, the Volterra series as been more widely used since the estimated kernels provide a better analytical description of the neurophysiological system [Marmarelis, 2004].

This approach has very powerful system modeling capability. However, in practice, it also present several difficulties: a very large number of coefficients need to be estimated, especially for higher order kernels, thus needing large volumes of data for the estimation of cross-correlations, and it estimation requires that the input is zero-mean white Gaussian distributed. Moreover, this approach can only be used if the system is assumed stationary and time-invariant.

CHAPTER 3
INNER PRODUCTS FOR POINT PROCESSES, AND INDUCED REPRODUCING KERNEL HILBERT SPACES

As motivated in Chapter 1, the fundamental operator for signal processing is an inner product definition. In this chapter we introduce several inner products for point process. More important than the inner product definitions themselves we want to illustrate how kernels for point processes can be defined following to different two approaches. Afterwards, we prove several properties of the kernels defined to demonstrate that the kernels are well-posed and each induces a corresponding reproducing kernel Hilbert space (RKHS) for computation. The relation between the RKHS induced by one of these kernels and others is analyzed since it provides insight on the full potential of these kernels may be explored. The problem of how to estimate these kernels from realizations is also considered.

3.1 Inner Product for Event Coordinates

Denote the mth event coordinate in a realization of the point process indexed by $i \in \mathbb{N}$ as $t_m^i \in \mathcal{T}$, with $m \in \{1, 2, \ldots, N_i\}$ and N_i the number of events in the realization. To simplify the notation, however, the explicit reference to the point process index will be omitted if it is not relevant or obvious from the context.

The simplest inner product that can be defined for point processes operates with only two event coordinates at a time. In the general case, such an inner product can be defined in terms of a kernel function defined on $\mathcal{T} \times \mathcal{T}$ into the reals, with $\mathcal{T}$ the event space where the events occur. Let κ denote such a kernel. Conceptually, this kernel operates in the same way as the kernels operating on data samples in machine learning [Schölkopf et al., 1999] and information theoretic learning [Príncipe et al., 2000]. Although it operates only with two events, it will play a major role whenever we operate with complete realizations of point processes. Indeed, the estimator for one of the point process kernels defined next relies on this simple kernel as an elementary operation for computation or composite operations.

To take advantage of the framework for statistical signal processing provided by RKHS theory, κ is required to be a symmetric positive definite function. By the Moore-Aronszajn theorem [Aronszajn, 1950], this ensures that an RKHS $\mathcal{H}_\kappa$ must exist for which κ is a reproducing kernel. The inner product in $\mathcal{H}_\kappa$ is given as

$$\kappa(t_m, t_n) = \langle \kappa(t_m, \cdot), \kappa(t_n, \cdot) \rangle_{\mathcal{H}_\kappa} = \langle \Phi_m, \Phi_n \rangle_{\mathcal{H}_\kappa} . \tag{3–1}$$

where Φ_m is the element in $\mathcal{H}_\kappa$ corresponding to t_m (that is, the transformed event coordinate).

Since the kernel operates directly on event coordinates and, typically, it is undesirable to emphasize events in this space, the kernel κ is further required to be *shift-invariant*. That is, for any $\theta \in \mathbb{R}$,

$$\kappa(t_m, t_n) = \kappa(t_m + \theta, t_n + \theta), \quad \forall t_m, t_n \in \mathcal{T}. \tag{3–2}$$

Hence, the kernel is only sensitive to the difference of the arguments and, consequently, we may write $\kappa(t_m, t_n) = \kappa(t_m - t_n)$.

For any symmetric, shift-invariant, and positive definite kernel, it is known that $\kappa(0) \geq |\kappa(\theta)|$.[1] This is important in establishing κ as a similarity measure between event coordinates since, as usual, an inner product should intuitively measure some form of inter-dependence. However, the conditions posed do not restrict this study to a single kernel. On the contrary, any kernel satisfying the above requirements is theoretically valid and understood under the framework proposed here, although the practical results may vary.

[1] This is a direct consequence of the fact that symmetric positive definite kernels denote inner products that obey the Cauchy-Schwarz inequality.

An example of a family of kernels that can be used (but not limited to) are the radial basis functions [Berg et al., 1984],

$$\kappa(t_m, t_n) = \exp(-|t_m - t_n|^p), \quad t_m, t_n \in \mathcal{T}, \tag{3–3}$$

for any $0 < p \leq 2$. Some well known kernels, such as the widely used Gaussian and Laplacian kernel are special cases of this family for $p = 2$ and $p = 1$, respectively.

Also of interest is to notice that for the natural norm induced by the inner product, shift-invariant kernels have the following property,

$$\|\Phi_m\| = \sqrt{\kappa(0)}, \quad \forall \Phi_m \in \mathcal{H}_\kappa. \tag{3–4}$$

Since the norm in $\mathcal{H}_\kappa$ of the transformed spike times point is constant, all the event coordinates are mapped to the surface of an hypersphere in $\mathcal{H}_\kappa$. The space of transformed event coordinates is called the manifold of $\mathcal{P}(\mathcal{T})$. This provides a different perspective of why the kernel used must be non-negative. Furthermore, the *geodesic distance* corresponding to the length of the smallest path contained within this manifold (in this case, the hypersphere) between two functions in this manifold, Φ_m and Φ_n, is given by

$$\begin{aligned} d(\Phi_m, \Phi_n) &= \|\Phi_m\| \arccos\left(\frac{\langle \Phi_m, \Phi_n \rangle}{\|\Phi_m\| \, \|\Phi_n\|}\right) \\ &= \sqrt{\kappa(0)} \arccos\left[\frac{\kappa(t_m, t_n)}{\kappa(0)}\right]. \end{aligned} \tag{3–5}$$

Put differently, from the geometry of the space of the transformed event coordinates, the kernel function is proportional to the cosine of the angle between two points in this space. Because the kernel is non-negative, the maximum angle is $\pi/2$, which restricts the manifold of transformed spike times to a small area of the surface of the sphere. With the kernel inducing the above metric, the manifold of the transformed points forms a *Riemannian space*. This space is *not* a linear space. Its span however is obviously a linear space. In fact, it equals the RKHS associated with the kernel. Computing with the transformed points will almost surely yield points outside of the manifold of transformed event coordinates. This means that such points cannot be mapped back to the input space directly. This restriction however is generally not a problem since most applications deal exclusively with the projections of points in the space, and if a representation in the input space is desired it may be obtained from the projection to the manifold of transformed input points.

The kernels κ discussed this far operate with only two event coordinates. As in commonly done in kernel methods, kernels on event coordinates can be combined to define kernels that operate with whole realizations of point processes. Suppose that one is interested in defining a kernel on point process realizations to measure similarity in the event patterns [Chi and Margoliash, 2001; Chi et al., 2007]. This kernel could be defined as

$$V(p_i, p_j) = \begin{cases} \max\limits_{l=0,1,\ldots,(N_i-N_j)} \sum\limits_{n=1}^{N_j} \kappa(t^i_{n+l} - t^j_n), & N_i \geq N_j \\ \max\limits_{l=0,1,\ldots,(N_j-N_i)} \sum\limits_{n=1}^{N_i} \kappa(t^i_n - t^j_{n+l}), & N_i < N_j. \end{cases} \tag{3–6}$$

Basically, this kernel measures if the realizations of the point processes have a one-to-one correspondence of the sequence of events. This occurs if the event coordinates occur with high precision and high reliability between the two point processes. Since point processes are defined here in terms of fixed duration, the maximum operation in the definition searches for the best event-to-event correspondence.

3.2 Inner Products for Point Processes

In the end of the previous section it was briefly illustrated how inner products for point processes can be built from kernels for spike times as traditionally done in machine learning. Obviously, many other point process kernels that operate directly from data realizations could be defined for diverse applications in a similar manner. However, in doing so it is often unclear the statistical structure embodied or point process model assumed by the kernel.

Rather than doing this directly, in this section, general inner products for point processes are defined from the intensity functions, which are fundamental statistical descriptors of point processes. This bottom-up construction of the kernels for point processes is unlike the previous approach taken in the previous section and is rarely taken in machine learning, but it provides direct access to the properties of the kernels defined and the RKHS they induce.

There is a great conceptual difference between the two approaches to design inner products for point processes: from kernels on event coordinates and from conditional intensity functions. In the first case, the inner product is defined directly for realizations of point processes and therefore the focus is placed in the estimators from data, whereas in the second case the inner product is primarily a *statistical descriptor* for which the problem of estimation from realizations needs to addressed later. Although both approaches may play a very important role in spike train methods, in this dissertation we focus of the second case, presented in this section, since we feel it is a more principled methodology.

3.2.1 Linear Cross-Intensity Kernels

In general, to completely characterize a point process the conditional intensity function $\lambda(t|H_t)$ is needed, where $t \in \mathcal{T} = [0, T]$ denotes the time coordinate and H_t is the history of the process up to time t. Consider two point processes, $p_i, p_j \in \mathcal{P}(\mathcal{T})$, with $i, j \in \mathbb{N}$, and denote the corresponding conditional intensity functions by $\lambda_{p_i}(t|H_t^i)$ and $\lambda_{p_j}(t|H_t^j)$, respectively. Assuming the point processes are defined in a finite parameter space $\mathcal{T}$, and the boundedness of the conditional intensity functions, we have that

$$\int_{\mathcal{T}} \lambda^2(t|H_t)dt < \infty. \tag{3–7}$$

In words, conditional intensity functions are square integrable functions on $\mathcal{T}$ and, as a consequence, are valid elements of an $L_2(\mathcal{T})$ space. Obviously, the space spanned by the conditional intensity functions, denoted $L_2(\lambda_{p_i}(t|H_t^i), t \in \mathcal{T})$, is contained in $L_2(\mathcal{T})$.

Therefore, we can easily define an inner product of the conditional intensity functions in $L_2(\lambda_{p_i}(t|H_t^i), t \in \mathcal{T})$ as the usual inner product in $L_2(\mathcal{T})$,

$$\begin{aligned} I(p_i, p_j) &= \left\langle \lambda_{p_i}(t|H_t^i), \lambda_{p_j}(t|H_t^j) \right\rangle_{L_2(\mathcal{T})} \\ &= \int_{\mathcal{T}} \lambda_{p_i}(t|H_t^i) \lambda_{p_j}(t|H_t^j) dt. \end{aligned} \tag{3–8}$$

Although we defined the inner product in the space of conditional intensity functions, it is in effect a function of the two point processes, and thus is a kernel function in the space of point processes $\mathcal{P}(\mathcal{T})$. The advantage in defining the inner product in terms of the conditional intensity functions is that the resulting kernel incorporates the statistics of the point processes directly. Moreover, the defined kernel can be utilized with *any* point process model since the conditional intensity function is a complete characterization of the point process [Cox and Isham, 1980]. Notice however that Equation 3–8 denotes a *functional* inner product definition, in the sense that the conditional intensity functions are in general well defined functions of t only for particular realizations of the point processes.

The dependence of the conditional intensity functions on the whole history of the process renders estimation of the previous kernel intractable from finite data, as almost always occurs in applications. A possibility is to consider simplified point process models which reduce the numbers of parameters needed to characterize the conditional intensity functions. One can consider, for example, that

$$\lambda(t|H_t) = \lambda(t, t - t_*), \tag{3–9}$$

where t_* is the spike time immediately preceding t. This restricted form gives rise to inhomogeneous Markov interval (IMI) processes [Kass and Ventura, 2001]. In this way it is possible to estimate the conditional intensity functions from realizations of the point processes, and then utilize the above inner product definition to operate with them. This point process model is very interesting it is simple yet general enough for modeling beyond renewal processes, but since we aim to compare the general principles presented starting from more typical approaches it will not be pursued in this paper. Needless to say, the same principles discussed here can be directly utilized in applications, although at the expense of computational complexity in the computation of the inner product as discussed later.

Another way to deal with the memory dependence is to take the expectation over the history of the process H_t which yields simply the intensity function solely depending on time. That is,

$$\lambda_{p_i}(t) = E_{H_t^i}\left\{\lambda_{p_i}(t|H_t^i)\right\}. \tag{3–10}$$

This expression is a direct consequence of the general limit theorem for point processes [Snyder, 1975] which, as introduced in Chapter 2, states that if multiple point processes are combined they converge towards a Poisson point process. An equivalent but alternate perspective is to merely assume directly Poisson processes to be a reasonable model for the problem at hands. The difference between the two perspectives is that in the second case the intensity functions can be estimated from single realizations in a plausible and simple manner. In either perspective, the kernel becomes simply

$$I(p_i, p_j) = \int_{\mathcal{T}} \lambda_{p_i}(t)\lambda_{p_j}(t)dt. \tag{3–11}$$

Starting from the most general definition of inner product several kernels from constrained forms of conditional intensity functions can be proposed for use in applications. One can think that the definition of Equation 3–8 gives rise to a family of *cross-intensity* (CI) *kernels* defined explicitly as an inner product, as is important for signal processing. Specific kernels are obtained from Equation 3–8 by imposing some particular form on how to account to the dependence on the history of the process and/or allowing for a nonlinear coupling between spike trains. Two fundamental advantages of the construction methodology is that it is possible to obtain a continuous functional space where no binning is necessary and that the generality of the approach allows for inner products to be crafted to fit a particular problem that one is trying to solve.

From the suggested definitions, the *memoryless cross-intensity* (mCI) *kernel* defined in Equation 3–11 clearly adopts the simplest form since the influence of the history of the process is neglected by the kernel. Interestingly, this simple kernel defines an RKHS that is equivalent to cross-correlation analysis so widespread, for example, in spike train analysis [Paiva et al., 2008], but this derivation clearly shows that it is the simplest of the cases. Still, the mCI kernel serves well as an example of the RKHS framework since it provides a broad perspective to several other works presented in the literature and suggests how methods can be reformulated to operate directly with point processes. In any case, as will be shown in Chapters 6 and 7, the derived algorithms are typically applicable for any kernel on point processes.

3.2.2 Nonlinear Cross-Intensity Kernels

The kernels defined in the previous section are linear operators in the space spanned by the conditional intensity functions and are the ones that relate the most with the present analysis methods. However, point process kernels can be made nonlinear by introducing a nonlinear weighting between the conditional intensity functions in the inner product. With this approach additional information can be extracted from the data since the nonlinearity implicitly incorporates in the measurement higher-order couplings between the estimated conditional intensity functions. This is of especial importance for the study of doubly-stochastic point processes, since the nonlinear weighting kernel basically aids the point process kernel to sense the higher-oder moments of the intensity process.

In the example shown we shall consider, for ease of exposition, the intensity functions directly (that is, the memoryless case). However, we remark that the methodology followed can be easily extended to general point process models as above.

By analogy to how the Gaussian kernel is obtained from the Euclidean norm, we can define a similar kernel for spike trains as

$$I^*_\sigma(p_i, p_j) = \exp\left[-\frac{\left\|\lambda_{p_i} - \lambda_{p_j}\right\|^2}{\sigma^2}\right], \tag{3–12}$$

where σ is the nonlinear weighting kernel size parameter and the norm naturally induced by a linear point process kernel, $\left\|\lambda_{p_i} - \lambda_{p_j}\right\| = \sqrt{\left\langle \lambda_{p_i} - \lambda_{p_j}, \lambda_{p_i} - \lambda_{p_j} \right\rangle}$, was used. This kernel is clearly nonlinear on the space of the intensity functions. On the other hand, the nonlinear mapping in this kernel does not operate directly on the intensity functions but on their norm and inner product and thus have reduced descriptive ability on the coupling of their time structure.

An alternate nonlinear CI kernel definition for point processes is

$$I^\dagger_\sigma(p_i, p_j) = \int_{\mathcal{T}} \mathcal{K}_\sigma\left(\lambda_{p_i}(t), \lambda_{p_j}(t)\right) dt, \tag{3–13}$$

where $\mathcal{K}_\sigma$ is a symmetric positive definite kernel with kernel size parameter σ. The advantage of this definition is that the kernel measures the possibly nonlinear coupling between the point process time structure expressed in the intensity functions. To verify this consider as an example that the Gaussian kernel, $\mathcal{K}_\sigma(x) = \exp\left[-x^2/(2\sigma^2)\right]$, was utilized in the computation of the point process kernel. The Gaussian kernel has Taylor series expansion

$$\mathcal{K}_\sigma(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{2^n \sigma^{2n} n!} x^2 = 1 - \frac{x^2}{2\sigma^2} + \frac{x^4}{8\sigma^4} - \ldots \tag{3–14}$$

Thus, this point process kernel depends on the norm of the spike trains (defined throught the mCI kernel) but also on higher-order moments of the difference between the intensity functions. In the remainder of this dissertation we shall refer to the nonlinear CI kernel in Equation 3–13 as the nCI kernel.

3.3 Properties of Cross-Intensity Kernels

3.3.1 Properties of Linear Cross-Intensity Kernels

In this section we present some relevant properties of the linear CI kernels defined in the general form in Equation 3–8. In addition to the knowledge they provide, they are necessary for establishing that the CI kernels are well defined, induce an RKHS with the necessary mathematical structure for computation, and aid in the understanding of the following sections. This section deals exclusively with CI kernels linear in the space of conditional intensity functions, unless explicitly stated, and thus linearity shall be implicit. Nonlinear CI kernels are studied in the next section.

Property 3.1. *The linear CI kernels are symmetric, non-negative and linear operators in the space of the intensity functions.*

Because the CI kernels operate on elements of $L_2(\mathcal{T})$ and correspond to the usual dot product from L_2, this property is a direct consequence of the properties inherited. More specifically, this property guaranties the CI kernels are valid inner products.

Property 3.2. *For any set of $n \geq 1$ point processes, the CI kernel matrix*

$$\mathbf{I} = \begin{bmatrix} I(p_1, p_1) & I(p_1, p_2) & \dots & I(p_1, p_n) \\ I(p_2, p_1) & I(p_2, p_2) & \dots & I(p_2, p_n) \\ \vdots & \vdots & \ddots & \vdots \\ I(p_n, p_1) & I(p_n, p_2) & \dots & I(p_n, p_n) \end{bmatrix},$$

is symmetric and non-negative definite.

Proof. The symmetry of the matrix results immediately from Property 3.1. By definition, a matrix is non-negative definite if and only if $\mathbf{a}^T\mathbf{I}\mathbf{a} \geq 0$, for any $\mathbf{a}^T = [a_1, \dots, a_n]$ with $a_i \in \mathbb{R}$. So, we have that

$$\mathbf{a}^T\mathbf{I}\mathbf{a} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j I(p_i, p_j), \tag{3–15}$$

which, making use of the general definition for CI kernels (Equation 3–8), yields,

$$\begin{aligned}
\mathbf{a}^T \mathbf{I} \mathbf{a} &= \int_{\mathcal{T}} \left(\sum_{i=1}^{n} a_i \lambda_{p_i}(t|H_t^i) \right) \left(\sum_{j=1}^{n} a_j \lambda_{p_j}(t|H_t^j) \right) dt \\
&= \left\langle \sum_{i=1}^{n} a_i \lambda_{p_i}(\cdot|H_t^i), \sum_{j=1}^{n} a_j \lambda_{p_j}(\cdot|H_t^j) \right\rangle_{L_2(\mathcal{T})} \\
&= \left\| \sum_{i=1}^{n} a_i \lambda_{p_i}(\cdot|H_t^i) \right\|_{L_2(\mathcal{T})}^2 \geq 0.
\end{aligned} \tag{3–16}$$

□

Through the work of Moore [1916] and due to the Moore-Aronszajn theorem [Aronszajn, 1950], the following two properties result as corollaries of Property 3.2.

Property 3.3. *CI kernels are symmetric and positive definite kernels. Thus, by definition, for any set of $n \geq 1$ point processes and corresponding n scalars $a_1, a_2, \ldots, a_n \in \mathbb{R}$,*

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j I(p_i, p_j) \geq 0. \tag{3–17}$$

Property 3.4. *There exists an Hilbert space for which a CI kernel is a reproducing kernel.*

Actually, Property 3.3 can be obtained explicitly by verifying that the inequality of Equation 3–17 is implied by Equation 3–15 and Equation 3–16 in the proof of Property 3.2.

Property 3.2, Property 3.3 and Property 3.4 are equivalent in the sense that any of these properties implies the other two. In our case, Property 3.2 is used to establish the other two. The most important consequence of these properties, explicitly stated through Property 3.4, is that a *CI kernel induces a unique RKHS*, denoted in general by $\mathcal{H}_I$. In the particular case of the mCI kernel the RKHS is denoted $\mathcal{H}_I$.

Property 3.5. *The CI kernels verify the Cauchy-Schwarz inequality,*

$$I^2(p_i, p_j) \leq I(p_i, p_i) I(p_j, p_j) \qquad \forall p_i, p_j \in \mathcal{P}(\mathcal{T}). \tag{3–18}$$

Proof. Consider the 2×2 CI kernel matrix,

$$\mathbf{I} = \begin{bmatrix} I(p_i, p_i) & I(p_i, p_j) \\ I(p_j, p_i) & I(p_j, p_j) \end{bmatrix}.$$

From Property 3.2, this matrix is symmetric and non-negative definite. Hence, its determinant is non-negative [Harville, 1997, pg. 245]. Mathematically,

$$\det(\mathbf{I}) = I(p_i, p_i) I(p_j, p_j) - I^2(p_i, p_j) \geq 0,$$

which proves the result of Equation 3–18. □

3.3.2 Properties of Nonlinear Cross-Intensity Kernels

We now prove that the nonlinear CI kernels defined in Section 3.2.2 are well defined. That is, they denote inner products in some RKHS of point processes. The two fundamental requirements are that the point process kernels are symmetric and positive definite in the space of point processes.

Property 3.6. *The point process kernel I^*_σ (defined in Equation 3–12) is a symmetric positive definite kernel of point processes.*

Proof. This function is obviously symmetric as the symmetry is inherited directly from the properties of the norm. In light of Theorem 2.2 in Chapter 3 of Berg et al. [1984], to prove the function is positive definite it suffices to prove that the norm of the difference between two intensity functions is negative definite. By definition, a real function ζ is negative definite if and only if it is symmetric and

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \zeta(x_i, x_j) \leq 0, \tag{3–19}$$

for all $n \geq 2$, $\{x_1, \ldots, x_n\} \subset \mathcal{X}$ and $c_1, \ldots, c_n \in \mathbb{K}$ with $\sum_{i=1}^{n} c_i = 0$. Let $n \geq 2$. For all n, consider the set of point processes $\{p_1, \ldots, p_n\} \subset \mathcal{P}(\mathcal{T})$, and $c_1, \ldots, c_n \in \mathbb{R}$ such that

$\sum_{i=1}^{n} c_i = 0$. Since the function is symmetric it remains to prove that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \left\|\lambda_{p_i} - \lambda_{p_j}\right\|^2 \leq 0. \tag{3–20}$$

Using the norm induced by one of the linear CI kernels, yields

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \left\langle \lambda_{p_i} - \lambda_{p_j}, \lambda_{p_i} - \lambda_{p_j} \right\rangle \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \left[\left\langle \lambda_{p_i}, \lambda_{p_i} \right\rangle - 2\left\langle \lambda_{p_i}, \lambda_{p_j} \right\rangle + \left\langle \lambda_{p_j}, \lambda_{p_j} \right\rangle\right] \\
&= \left(\sum_{i=1}^{n} c_i \left\langle \lambda_{p_i}, \lambda_{p_i} \right\rangle\right) \underbrace{\left(\sum_{j=1}^{n} c_j\right)}_{=0} - 2\left\langle \sum_{i=1}^{n} c_i \lambda_{p_i}, \sum_{j=1}^{n} c_j \lambda_{p_j} \right\rangle \\
&\quad + \underbrace{\left(\sum_{i=1}^{n} c_i\right)}_{=0} \left(\sum_{j=1}^{n} c_j \left\langle \lambda_{p_j}, \lambda_{p_j} \right\rangle\right) \\
&= -2\left\|\sum_{i=1}^{n} c_i \lambda_{p_i}\right\| \leq 0,
\end{aligned} \tag{3–21}$$

since the norm is, by definition, always positive. □

Property 3.7. *For any symmetric positive definite kernel $\mathcal{K}_\sigma$, the nonlinear CI kernel $I_\sigma^\dagger$ (defined in Equation 3–13) is a symmetric positive definite kernel of point processes.*

Proof. The symmetry of $I_\sigma^\dagger$ is a direct consequence of the symmetry of the kernel $\mathcal{K}_\sigma$. Denote by $\{p_1, \ldots, p_n\} \subset \mathcal{P}(\mathcal{T})$ any set of n point processes, with $n \geq 2$, and consider coefficients $a_1, \ldots, a_n \in \mathbb{R}$. To prove that $I_\sigma^\dagger$ is positive define one needs to show that

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j I_\sigma^\dagger(p_i, p_j) \geq 0. \tag{3–22}$$

Substituting the definition of $I_\sigma^\dagger$ in the previous equation yields,

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j I_\sigma^\dagger(p_i, p_j) &= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \int_{\mathcal{T}} \mathcal{K}_\sigma(\lambda_{p_i}(t), \lambda_{p_j}(t)) dt \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \int_{\mathcal{T}} \left\langle \Upsilon_{\lambda_{p_i}(t)}, \Upsilon_{\lambda_{p_j}(t)} \right\rangle_{\mathcal{H}_\mathcal{K}} dt,
\end{aligned} \tag{3–23}$$

where the kernel $\mathcal{K}_\sigma$ was substituted by its inner product in the corresponding RKHS $\mathcal{H}_\mathcal{K}$, and $\Upsilon_{\lambda_{p_i}(t)}$ denotes the transformation of the intensity function value at time t (the argument of $\mathcal{K}_\sigma$) into $\mathcal{H}_\mathcal{K}$. Utilizing the linearity of the integral and the inner product operator leads to

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j I_\sigma^\dagger(p_i, p_j) &= \int_\mathcal{T} \left\langle \sum_{i=1}^{n} a_i \Upsilon_{\lambda_{p_i}(t)}, \sum_{j=1}^{n} a_j \Upsilon_{\lambda_{p_j}(t)} \right\rangle_{\mathcal{H}_\mathcal{K}} dt \\ &= \int_\mathcal{T} \left\| \sum_{i=1}^{n} a_i \Upsilon_{\lambda_{p_i}(t)} \right\|_{\mathcal{H}_\mathcal{K}}^2 dt \geq 0,\end{aligned} \tag{3–24}$$

which proves the property. □

The following property follows immediately as a corollary:

Property 3.8. *The nonlinear CI kernels, I_σ^* and $I_\sigma^\dagger$, each*

(i) Induce an RKHS,
(ii) Give rise to non-negative definite Gram matrices, and
(iii) Verifies the Cauchy-Schwarz inequality.

These are because, as stated in the previous section, Property 3.8(i), Property 3.8(ii) and the positive definiteness of the kernel are equivalent facts. Then, as shown in the proof of Property 3.5, the Cauchy-Schwarz follows.

3.4 Estimation of Cross-Intensity Kernels

The problem of estimating the previously defined cross-intensity kernels is now considered. Recall that this problem only poses itself for kernels defined in terms of the statistical descriptors of point processes, whereas point processes kernels built from kernels on event coordinates are in effect estimators. This will be clear from our presentation and, in fact, the relationship between perspectives will be observed in one of the cases.

3.4.1 Estimation of General Cross-Intensity Kernels

From the point process kernel definitions, is should be clear that for evaluation of CI kernels one needs to estimate first the conditional intensity function from realizations of the point processes. A possible approach is the statistical estimation framework recently proposed by Truccolo et al. [2005] for spike trains. Briefly, it represents the point process realizations as realizations of a Bernoulli random process, and then utilizes a generalized linear model (GLM) to fit a conditional intensity function to the data. This is done by assuming that the logarithm of the conditional intensity function has the form

$$\log \lambda_{p_i}(\hat{t}_n|H_n^i) = \sum_{m=1}^{q} \theta_m g_m(\nu_m(\hat{t}_n)), \qquad (3\text{–}25)$$

where $\hat{t}_n$ is the nth discrete-time instant, g_m's are general transformations of independent functions $\nu_m(\cdot)$, θ_m's are the parameter of the GLM and q is the number of parameters. Thus, GLM estimation can be used under a Poisson distribution with a log link function. The terms $g_m(\nu_m(\hat{t}_n))$ are called the predictor variables in the GLM framework and, if one considers the conditional intensity to depend only linearly on the history of the events then the g_m's can be simply delays. In general the intensity can depend nonlinearly on the history or external factors. Based on the estimated conditional intensity function, any of the inner products introduced in Section 3.2 can be evaluated numerically.

Although quite general, the approach by Truccolo et al. [2005] has a main drawback: since q must be larger that the average inter-spike interval a (very) large number of parameters need to be estimated thus requiring long spike trains ($>$ 10 seconds). Notice that non-parametric estimation of the conditional intensity function without greatly sacrifice the temporal precision requires small time intervals, which means that q and therefore the realizations used for estimation must have longer duration.

In spite of these difficulties, we maintain the importance of the RKHS framework and these point process kernel definitions. For more efficient computation these kernels may make use of developments in conditional intensity function estimation procedures will may expedite their use in practical applications.

3.4.2 Estimation of the mCI Kernel

In the particular case of the mCI kernel, defined in Equation 3–11, a much simpler estimator can be derived. We now focus on this case. Since we are interested in estimating the mCI kernel from single realizations of the point processes, and for the reasons presented before, we will assume that the realizations belong to Poisson processes. Then, using kernel smoothing [Reiss, 1993; Dayan and Abbott, 2001; Richmond et al., 1990] for the estimation of the intensity function we can derive an estimator for the point process kernel. Again, the advantage of this route is that a statistical interpretation is available while simultaneous approaching the problem from a practical point of view. Moreover, in this particular case the connection between the mCI kernel and κ will now become obvious.

According to kernel smoothing intensity estimation, given a realization of point process p_i comprising of event coordinates $\{t_m^i \in \mathcal{T} : m = 1, \ldots, N_i\}$ the estimated intensity function is

$$\hat{\lambda}_{s_i}(t) = \sum_{m=1}^{N_i} h(t - t_m^i), \tag{3–26}$$

where h is the smoothing function. This function must be non-negative and integrate to one over the real line (just like a probability density function (pdf)). Commonly used smoothing functions are the Gaussian, Laplacian and α-function, among others.

From a filtering perspective, Equation 3–26 can be seen as a linear convolution between the filter impulse response given by $h(t)$ and the realization written as a sum of Dirac functionals centered at the event locations. In particular, binning is nothing but a special case of this procedure in which h is a rectangular window and the spike times are first quantized according to the width of the rectangular window (cf. Section 2.4.1.2). Moreover, it is interesting to observe that intensity estimation as shown above is directly related to the problem of pdf estimation with Parzen windows [Parzen, 1962] except for a normalization term, a connection made clear by Diggle and Marron [1988].

Consider realizations of point processes $p_i, p_j \in \mathcal{P}(\mathcal{T})$ with estimated intensity functions $\hat{\lambda}_{p_i}(t)$ and $\hat{\lambda}_{p_j}(t)$ according to Equation 3–26. Substituting the estimated intensity functions in the definition of the mCI kernel (Equation 3–11) yields the estimator,

$$\hat{I}(p_i, p_j) = \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \kappa(t_m^i - t_n^j), \tag{3–27}$$

where κ is the kernel obtained by the autocorrelation of the intensity estimation function h with itself. A well known example for h is the Gaussian function in which case κ is also the Gaussian function (with σ scaled by $\sqrt{2}$). Another example for h is the one-sided exponential function which yields κ as the Laplacian kernel. In general, if a kernel is selected first and h is assumed to be symmetric, then κ equals the autocorrelation of h and thus h can be found by evaluating the inverse Fourier transform of the square root of the Fourier transform of κ.

The accuracy of this point process kernel estimator depends only on the accuracy of the estimated intensity functions. If enough data is available such that the estimation of the intensity functions can be made exact then the mCI kernel estimation error is zero. Despite this direct dependency, the estimator effectively bypasses the estimation of the intensity functions and operates directly on the event coordinates of the whole realization without loss of resolution and in a computationally efficient manner since it takes advantage of the typically sparse occurrence of events.

As Equation 3–27 shows, if κ is chosen such that it satisfies the requirements in Section 3.1, then the mCI kernel ultimately corresponds to a linear combination of κ operating on all pairwise differences of event coordinates, one pair at a time. In other words, the mCI kernel is the expectation of the linear combination of pairwise inner products between event coordinates. Put in this way, we can now clearly see how the mCI inner product estimator builds upon the inner product for event coordinates, κ, presented in Section 3.1.

3.4.3 Estimation of Nonlinear (Memoryless) Cross-Intensity Kernels

As shown in the previous section, the estimator of the mCI kernel results naturally by substituting the kernel intensity function estimator in the mCI kernel definition. For the related nonlinear CI kernels presented in Section 3.2.2 this matter is slightly more complicated due to the nonlinearity introduced in the kernel definition. In this section we briefly suggest how these nonlinear CI kernels can be estimated.

3.4.3.1 Estimation of $\mathcal{I}^*_\sigma$

The nonlinear CI kernel defined by Equation 3–12 lends itself to a very simple estimator. Indeed, this kernel definition relies on the computation of the natural norm induced by the inner product associated with the mCI kernel (but the norm induced by any of the cross-intensity kernels can be considered in general). The induced norm is

$$\begin{aligned}\left\|\lambda_{p_i} - \lambda_{p_j}\right\|^2_{\mathcal{H}_I} &= \left\langle \lambda_{p_i} - \lambda_{p_j}, \lambda_{p_i} - \lambda_{p_j} \right\rangle_{\mathcal{H}_I} \\ &= \left\langle \lambda_{p_i}, \lambda_{p_i} \right\rangle - 2\left\langle \lambda_{p_i}, \lambda_{p_j} \right\rangle + \left\langle \lambda_{p_j}, \lambda_{p_j} \right\rangle \\ &= \left\|\lambda_{p_i}\right\|^2 - 2\left\langle \lambda_{p_i}, \lambda_{p_j} \right\rangle + \left\|\lambda_{p_j}\right\|^2 .\end{aligned} \tag{3–28}$$

Therefore, the computational bottleneck in the evaluation of this kernel is the computation of the three inner products corresponding to the norm. From then on, one only needs to compute the exponential function with this norm (scaled) once. For inhomogeneous Poisson processes, this can be immediately done using the estimator for the mCI kernel described in Section 3.4.2 to first evaluate the norm. Thus, the computational complexity is of the same order.

3.4.3.2 Estimation of the nCI kernel, $\mathcal{I}^\dagger_\sigma$

Evaluation of the nonlinear CI kernel in Equation 3–13, however, does not build on our previous findings. The reason for this is that the kernel $\mathcal{K}_\sigma$ nonlinearly weighs the temporal relationship between the two intensity functions, and therefore we cannot obtain an analytical expressing to the integral on the combination of smoothing functions.

Thus we will propose an estimator which relies on a particular form of the intensity function estimator. The key idea is to simplify the problem by dividing time in intervals during which the interaction among intensity functions is constant. This is achieved simply by utilizing a rectangular pulse as the smoothing function. Again, the focus here in the inhomogeneous Poisson case. In the more general case of conditional intensity function

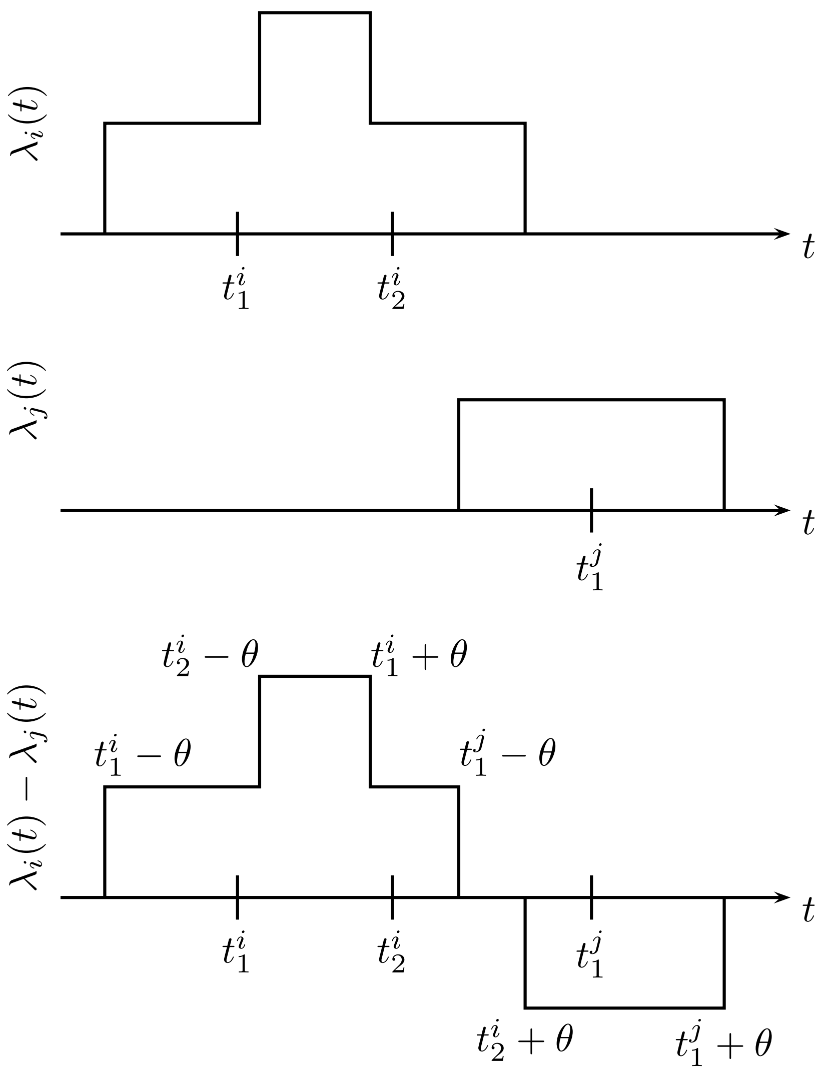


Figure 3-1. Estimation of difference of intensity functions for evalution of nonlinear kernel in Equation 3–13.

estimation using the GLM framework, the point process is made discrete-time which automatically introduces this simplification. Nevertheless, in the estimator presented time needs not to be discretized.

Consider a symmetric, positive definite and shift-invariant kernel $\mathcal{K}_\sigma$ and that a rectangular pulse smoothing function is used for kernel smoothing intensity estimation. If $\mathcal{K}_\sigma$ is taken to be shift-invariant (as occurs for the commonly used kernels), then it is only sensitive to the difference of the arguments. Therefore, it is easy to verify that there exist a small finite number of transitions in the value of the difference between intensity

Table 3-1. Outline of the algorithm for estimation of the nCI kernel, $\mathcal{I}_\sigma^\dagger$.

Step 1 Define,

$$S^\dagger = [t_1^i - \theta, \ldots, t_{N_i}^i - \theta, t_1^i + \theta, \ldots, t_{N_i}^i + \theta, t_1^j - \theta, \ldots, t_{N_j}^j - \theta, t_1^j + \theta, \ldots, t_{N_j}^j + \theta],$$

and the corresponding incremental sequence,

$$\delta = [\underbrace{1, 1, \ldots, 1}_{N_i \text{ times}}, \underbrace{-1, -1, \ldots, -1}_{N_i \text{ times}}, \underbrace{-1, -1, \ldots, -1}_{N_j \text{ times}}, \underbrace{1, 1, \ldots, 1}_{N_j \text{ times}}].$$

Step 2 Sort $S^\dagger$ in ascending order, and apply the same reordering to δ.

Step 3 Set $n = \{\text{Number of negative times in } S^\dagger\}$, $\Delta = \sum_{i=1}^{n-1} \delta_i$, and $\mathcal{I}^\dagger = t_0 = 0$.

```
While $n < 2(N_i + N_j)$ and $S_n^\dagger < T$
    1. $\mathcal{I}^\dagger = \mathcal{I}^\dagger + (S_n^\dagger - t_0) \times \mathcal{K}_\sigma(\Delta/(2\theta))$
    2. $t_0 = S_n^\dagger$
    3. $\Delta = \Delta + \delta_n$
    4. $n = n + 1$
end
$\mathcal{I}^\dagger = \mathcal{I}^\dagger + (T - t_0) \times \mathcal{K}_\sigma(\Delta/(2\theta))$.
```

functions, as depicted in Figure 3-1. For two point processes p_i, p_j with realizations $\{t_1^i, \ldots, t_{N_i}^i\}$ and $\{t_1^j, \ldots, t_{N_j}^j\}$, the changes in the difference of the intensity function occur at times $[t_1^i - \theta, \ldots, t_{N_i}^i - \theta, t_1^i + \theta, \ldots, t_{N_i}^i + \theta, t_1^j - \theta, \ldots, t_{N_j}^j - \theta, t_1^j + \theta, \ldots, t_{N_j}^j + \theta]$. At each of these times, the difference will either increase or decrease by $1/(2\theta)$. To keep track of this we create a sequence with ones for the event times of p_i minus θ and event times of p_j plus θ, and -1 for the event times of p_i plus θ and event times of p_j minus θ (the choice of signs is arbitrary and can be reversed since $\mathcal{K}_\sigma$ is symmetric). By reordering the sequence of times, and applying the same reordering to the sequence of 1 and -1, we obtain a sequence of intervals with constant value on which the integral is simply the interval length times $\mathcal{K}_\sigma(\Delta/(2\theta))$, where Δ is the sum of the difference increments up to that interval. The estimator is summarized in Table 3-1.

The computational bottleneck of this estimator is the sorting operation which has computational complexity with order $\mathcal{O}(N_i N_j \log_2(N_i N_j))$. This means that estimating the nCI kernel is considerably more complex than estimating the mCI kernel, which is

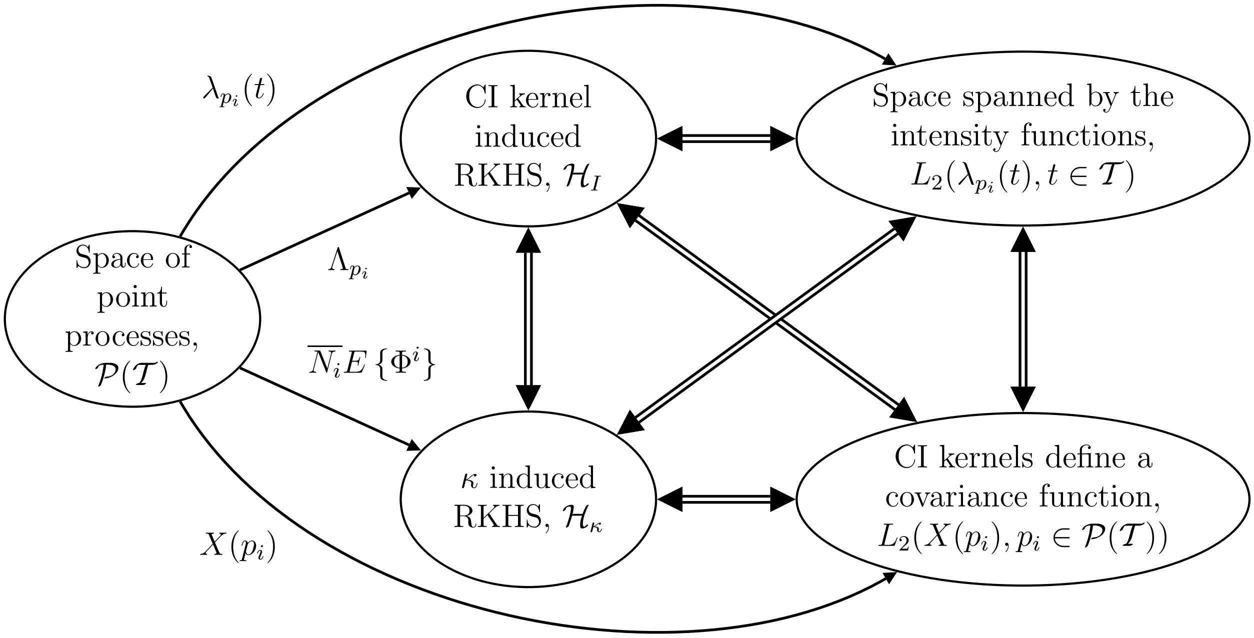

$\lambda_{p_i}(t)$
CI kernel induced RKHS, $\mathcal{H}_I$
Space spanned by the intensity functions, $L_2(\lambda_{p_i}(t), t \in \mathcal{T})$
Space of point processes, $\mathcal{P}(\mathcal{T})$
Λ_{p_i}
$\overline{N}_i E\left\{\Phi^i\right\}$
κ induced RKHS, $\mathcal{H}_\kappa$
CI kernels define a covariance function, $L_2(X(p_i), p_i \in \mathcal{P}(\mathcal{T}))$
$X(p_i)$

Figure 3-2. Relation between the original space of point processes $\mathcal{P}(\mathcal{T})$ and the various Hilbert spaces. The bi-directional double-line connections denote congruence between spaces.

$\mathcal{O}(N_i N_j)$. Nevertheless, the proposed estimator is quite efficient considering that the integral cannot be simplified analytically.

3.5 RKHS Induced by the Memoryless Cross-Intensity Kernel and Congruent Spaces

Some considerations about the RKHS space $\mathcal{H}_I$ induced by the mCI kernel and congruent spaces are made in this section. The relationship between $\mathcal{H}_I$ and its congruent spaces provides alternative perspectives and a better understanding of how the mCI kernel can be utilized for computation with point processes. Figure 3-2 provides a diagram of the relationships among the various spaces discussed next.

Some of these relationships extend directly to more general CI kernels. Therefore, although this section focus on the spaces associated with the mCI kernel, we will mention if similar connections hold for other point process kernels whenever applicable.

3.5.1 Space Spanned by Intensity Functions

In the introduction of the mCI kernel the usual dot product in $L_2(\mathcal{T})$, the space of square integrable intensity functions defined on $\mathcal{T}$, was utilized. The definition of the inner product in this space provides an intuitive understanding to the reasoning involved. $L_2(\lambda_{p_i}(t), t \in \mathcal{T}) \subset L_2(\mathcal{T})$ is clearly an Hilbert space with inner product defined in Equation 3–11, and is obtained from the span of all intensity functions. Notice that this space also contains functions that are not valid intensity functions resulting from the linear span of the space (intensity functions are always non-negative). However, since our interest is mainly on the evaluation of the inner product this is of no consequence. The main limitation is that $L_2(\lambda_{p_i}(t), t \in \mathcal{T})$ is *not* an RKHS. This should be clear because elements in this space are functions defined on $\mathcal{T}$, whereas elements in the RKHS $\mathcal{H}_I$ must be functions defined on $\mathcal{P}(\mathcal{T})$.

Despite the differences, the spaces $L_2(\lambda_{p_i}(t), t \in \mathcal{T})$ and $\mathcal{H}_I$ are closely related. In fact, $L_2(\lambda_{p_i}(t), t \in \mathcal{T})$ and $\mathcal{H}_I$ are congruent. We can verify this congruence explicitly since there is clearly a one-to-one mapping,

$$\lambda_{p_i}(t) \in L_2(\lambda_{p_i}(t), t \in \mathcal{T}) \quad \longleftrightarrow \quad \Lambda_{p_i}(p) \in \mathcal{H}_I,$$

and, by definition of the mCI kernel,

$$I(p_i, p_j) = \left\langle \lambda_{p_i}, \lambda_{p_j} \right\rangle_{L_2(\mathcal{T})} = \left\langle \Lambda_{p_i}, \Lambda_{p_j} \right\rangle_{\mathcal{H}_I}. \tag{3–29}$$

Actually, the congruence between the two space holds for any linear cross-intensity kernel since the inner product is the same in both spaces. For the nonlinear CI kernels, for example, the two space are still closely related but the inner product is not directly available in $L_2(\lambda_{p_i}(t), t \in \mathcal{T})$ and therefore the two spaces are not congruent. A direct consequence of the basic congruence theorem is that the two spaces have the same dimension [Parzen, 1959].

3.5.2 Induced RKHS

In Section 3.3.1 it was shown that the mCI kernel is symmetric and positive definite (Property 3.1 and Property 3.3, respectively). Consequently, by the Moore-Aronszajn theorem [Aronszajn, 1950], there exists an Hilbert space $\mathcal{H}_I$ for which the mCI kernel evaluates the inner product and is a reproducing kernel (Property 3.4). This means that $I(p_i, \cdot) \in \mathcal{H}_I$ for any $p_i \in \mathcal{P}(\mathcal{T})$ and, for any $\zeta \in \mathcal{H}_I$, the reproducing property holds

$$\langle \zeta, I(p_i, \cdot) \rangle_{\mathcal{H}_I} = \zeta(p_i). \tag{3–30}$$

As a result the kernel trick follows,

$$I(p_i, p_j) = \langle I(p_i, \cdot), I(p_j, \cdot) \rangle_{\mathcal{H}_I}. \tag{3–31}$$

Written in this form, it is easy to verify that the point in $\mathcal{H}_I$ corresponding to a spike train $p_i \in \mathcal{P}(\mathcal{T})$ is $I(p_i, \cdot)$. In other words, given any spike train $p_i \in \mathcal{P}(\mathcal{T})$, this spike train is mapped to $\Lambda_{p_i} \in \mathcal{H}_I$, given explicitly (although unknown in closed form) as $\Lambda_{p_i} = I(p_i, \cdot)$. Then Equation 3–31 can be restated in the more usual form as

$$I(p_i, p_j) = \left\langle \Lambda_{p_i}, \Lambda_{p_j} \right\rangle_{\mathcal{H}_I}. \tag{3–32}$$

It must be remarked that $\mathcal{H}_I$ is in fact a functional space. More specifically, that points in $\mathcal{H}_I$ are functions of point processes; that is, they are functions defined on $\mathcal{P}(\mathcal{T})$. This is a key difference between the space of intensity functions $L_2(\mathcal{T})$ explained before and the RKHS $\mathcal{H}_I$, in that the latter allows for statistics of the transformed spike trains to be estimated as functions of spike trains.

In light of Property 3.4 and Property 3.8(i), similar considerations can the drawn for any of the point process kernels presented in this work. Naturally, the functional space and corresponding functional mapping will be different for different kernels, but the same mathematical structure exists. Since the *structure* to perform computation is the same, an algorithm derived in this space can be utilized using any point process kernel.

3.5.3 Memoryless CI Kernel and the RKHS Induced by κ

The mCI kernel estimator in Equation 3–27 shows the evaluation written in terms of elementary kernel operations on event coordinates. This fact alone provides an interesting perspective on how the mCI kernel uses the event statistics. To see this more clearly, consider κ to be chosen according to Section 3.1 as a symmetric positive definite kernel, then it can be substituted by its inner product (Equation 3–1) in the mCI kernel estimator, yielding

$$\begin{aligned}\hat{I}(p_i, p_j) &= \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \left\langle \Phi_m^i, \Phi_n^j \right\rangle_{\mathcal{H}_\kappa} \\ &= \left\langle \sum_{m=1}^{N_i} \Phi_m^i, \sum_{n=1}^{N_j} \Phi_n^j \right\rangle_{\mathcal{H}_\kappa}.\end{aligned} \tag{3–33}$$

When the number of samples approaches infinity (so that the intensity functions and, consequently the mCI kernel, can be estimated exactly) the mean of the transformed event coordinates approaches the expectation. Hence, Equation 3–33 results in

$$I(p_i, p_j) = \overline{N_i}\,\overline{N_j} \left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^j\right\}\right\rangle_{\mathcal{H}_\kappa}, \tag{3–34}$$

where $E\{\Phi^i\}$, $E\{\Phi^i\}$ denotes the expectation of the transformed event coordinates and $\overline{N_i}, \overline{N_j}$ are the expected number of events in realizations from point processes p_i and p_j, respectively.

Equation 3–34 explicitly shows that the mCI kernel can be computed as a (scaled) inner product of the expectation of the transformed event coordinates in the RKHS $\mathcal{H}_\kappa$ induced by κ. In other words, there is a congruence $\mathscr{G}$ between $\mathcal{H}_\kappa$ and $\mathcal{H}_I$ in this case given explicitly in terms of the expectation of the transformed event coordinates as $\mathscr{G}(\Lambda_{p_i}) = \overline{N_i} E\{\Phi^i\}$, such that

$$\left\langle \Lambda_{p_i}, \Lambda_{p_j} \right\rangle_{\mathcal{H}_I} = \left\langle \mathscr{G}(\Lambda_{p_i}), \mathscr{G}(\Lambda_{p_j}) \right\rangle_{\mathcal{H}_\kappa} = \overline{N_i}\,\overline{N_j} \left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^j\right\}\right\rangle_{\mathcal{H}_\kappa}. \tag{3–35}$$

Recall that the transformed event coordinates form a manifold (the subset of an hypersphere) and, since these points have constant norm, the kernel inner product depends only on the angle between points. This is typically not true for the average of these points however. Observe that the circular variance of the transformed event coordinates for point process p_i is [Mardia and Jupp, 2000]

$$\begin{aligned} \mathrm{var}(\Phi^i) &= E\left\{\left\langle \Phi^i_m, \Phi^i_m \right\rangle_{\mathcal{H}_\kappa}\right\} - \left\langle E\left\{\Phi^i\right\}, E\left\{\Phi^i\right\}\right\rangle_{\mathcal{H}_\kappa} \\ &= \kappa(0) - \left\| E\left\{\Phi^i\right\}\right\|^2_{\mathcal{H}_\kappa}. \end{aligned} \tag{3–36}$$

So, the norm of the mean transformed event coordinates is inversely proportional to the variance of the elements in $\mathcal{H}_\kappa$. This means that the inner product between two point processes depends also on the dispersion of these average points. This fact is important because data reduction techniques, for example, heavily rely on optimization with the data variance. For instance, kernel principal component analysis [Schölkopf et al., 1998] directly maximizes the variance expressed by Equation 3–36 [Paiva et al., 2006].

3.5.4 Memoryless CI Kernel as a Covariance Kernel

In Section 3.3.1 it was proved that the mCI kernel is indeed a symmetric positive definite kernel. As reviewed in Appendix A, Parzen [1959] showed that any symmetric and positive definite kernel is also a covariance function of a random process defined in the original space of the kernel (a review of these ideas can be found in Wahba [1990, Chapter 1]). This means that for the mCI, and in general for any of the point process kernels considered, there exists a space of random processes are defined on $\mathcal{P}(\mathcal{T})$ for which the point process kernel is a covariance operator.

Let X denote this random process. Then, for any $p_i \in \mathcal{P}(\mathcal{T})$, $X(p_i)$ is a random variable on a probability space $(\Omega, \mathcal{B}, P)$ with measure P. As proved by Parzen, this random process is Gaussian distributed with zero mean and covariance function

$$I(p_i, p_j) = E_\omega\left\{X(p_i)X(p_j)\right\}. \tag{3–37}$$

Notice that the expectation is over $\omega \in \Omega$ since $X(p_i)$ is a random variable defined on Ω, a situation which can be written explicitly as $X(p_i, \omega)$, $p_i \in \mathcal{P}(\mathcal{T})$, $\omega \in \Omega$. This means that X is actually a doubly stochastic random process. An intriguing perspective is that, for any given ω, $X(p_i, \omega)$ corresponds to an ordered and almost surely non-uniform random sampling of $X(\cdot, \omega)$. The space spanned by these random variables is $L_2(X(p_i), p_i \in \mathcal{P}(\mathcal{T}))$ since X is obviously square integrable (that is, X has finite covariance).

The RKHS $\mathcal{H}_I$ induced by the mCI kernel and the space of random functions $L_2(X(p_i), p_i \in \mathcal{P}(\mathcal{T}))$ are congruent. This fact is obvious since there is clearly a congruence mapping between the two spaces. In light of this theory we can henceforward reason about the mCI kernel also as a covariance function of random variables directly dependent on the spike trains with well defined statistical properties. Allied to our familiarity and intuitive knowledge of the use of covariance (which is nothing but cross-correlation between centered random variables) this concept can be of great importance in the design of optimal learning algorithms that work with spike trains. This is because linear methods are known to be optimal for Gaussian distributed random variables.

As mentioned above, similar considerations can be made for any of the point process kernels, although the Gaussian random processes in the covariance are different for each since they characterize the statistics of the point process model considered by the point process kernel.

3.6 Point Process Distances

The concept of distance is very useful in classification and analysis of data, and point processes are no exception. The main aim of this section is to show that inner products for point processes can be utilized to easily define distances for point processes in a rigorous manner, and indeed naturally induce at least two forms of distances for point processes. This section does not aim at proposing any particular measure but to highlight this natural connection and convey the generality of RKHS framework by suggesting how distances can be formulated from basic principles if needed. Due to their relevance in neurophysiological studies, these ideas are also particularized for the mCI kernel to show that some of the measures proposed in this context are simply special cases of the RKHS framework.

3.6.1 Norm Distance

The fact that $\mathcal{H}_I$ is an Hilbert space and therefore possesses a norm suggests an obvious definition for a distance between point processes. In fact, for the linear cross-intensity kernels, since $L_2(\mathcal{T})$ is also an Hilbert space this fact would have sufficed. The distance between two point processes or, in general, any two points in $\mathcal{H}_I$, is defined as

$$\begin{aligned} d_{ND}(p_i, p_j) &= \left\| \Lambda_{p_i} - \Lambda_{p_j} \right\|_{\mathcal{H}_I} \\ &= \sqrt{\left\langle \Lambda_{p_i} - \Lambda_{p_j}, \Lambda_{p_i} - \Lambda_{p_j} \right\rangle_{\mathcal{H}_I}} \\ &= \sqrt{\left\langle \Lambda_{p_i}, \Lambda_{p_i} \right\rangle - 2\left\langle \Lambda_{p_i}, \Lambda_{p_j} \right\rangle + \left\langle \Lambda_{p_j}, \Lambda_{p_j} \right\rangle} \\ &= \sqrt{I(p_i, p_i) - 2I(p_i, p_j) + I(p_j, p_j)}. \end{aligned} \tag{3–38}$$

where $\Lambda_{p_i}, \Lambda_{p_i} \in \mathcal{H}_I$ denotes the transformed point processes in the RKHS. From the properties of the norm and the Cauchy-Schwarz inequality (Property 3.5) it immediately follows that d_{ND} is a valid distance since, for any spike trains $p_i, p_j, p_k \in \mathcal{P}(\mathcal{T})$, it satisfies the three distance axioms:

(i) Symmetry: $d_{ND}(p_i, p_j) = d_{ND}(p_j, p_i)$;
(ii) Positiveness: $d_{ND}(p_i, p_j) \geq 0$, with equality holding if and only if $p_i = p_j$;
(iii) Triangle inequality: $d_{ND}(p_i, p_j) \leq d_{ND}(p_i, p_k) + d_{ND}(p_k, p_j)$.

This distance is basically a generalization of the idea behind the Euclidean distance in a continuous space of functions.

3.6.2 Cauchy-Schwarz Distance

The previous distance is the natural definition for distance whenever an inner product is available. However, as for other L_2 spaces, alternatives measures for point processes can be defined. In particular, based on the Cauchy-Schwarz inequality (Property 3.5

and Property 3.8) we can define the *Cauchy-Schwarz (CS) distance* between two point processes as

$$d_{CS}(p_i, p_j) = \arccos \frac{I(p_i, p_i)I(p_j, p_j)}{I^2(p_i, p_j)}. \tag{3–39}$$

From the symmetry of the inner products and the Cauchy-Schwarz inequality it follows that d_{CS} is symmetric and always positive, and thus verifies the first two axioms of distance. Moreover, since d_{CS} is the angular distance between points it also verifies the triangle inequality.

The major difference between the normed distance presented in the previous section and the CS distance is that the latter is not an Euclidean measure. Indeed, because it measures the angular distance between the spike trains it is a Riemannian metric. This utilizes the same idea expressed in Equation 3–5 in presenting the geodesic distance associated with any symmetric positive definite kernel.

3.6.3 Spike Train Measures

Several spike train measures have been proposed in the literature [Victor and Purpura, 1997; van Rossum, 2001; Schreiber et al., 2003] and they play an important role in neurophysiological studies. Since spike trains are realizations of point processes the above ideas can also be applied to measure similarity or dissimilarity between spike trains. Indeed, it is insightful to verify that two well established spike train measures can be obtained directly as special cases of the two point process distances presented for the simplest of the point process kernels considered, the mCI kernel.

Since the inner product denotes by the mCI kernel is defined in $L_2(\mathcal{T})$, the norm distance could obviously also be formulated directly and with the same result in $L_2(\mathcal{T})$. Then, if one considers this perspective with a causal decaying exponential function as the smoothing kernel for intensity estimation then we immediately observe that d_{ND} corresponds, in this particular case, to the distance proposed by van Rossum [2001]. Using instead a rectangular smoothing function the distance then resembles the distance proposed by Victor and Purpura [1997], as pointed by Schrauwen and Campenhout [2007], although its definition prevents an exact formulation in terms of the mCI kernel. Finally, using a Gaussian kernel the same distance used by Maass et al. [2002] is obtained. Notice that although it had already been noticed that other cost (i.e. kernel) functions between spike times could be used instead of the initially described [Schrauwen and Campenhout, 2007], the framework given here fully characterizes the class of valid kernels and explains their role in the time domain. Moreover, ultimately the mCI kernel estimator can be utilized for efficient computation using κ to be the Laplacian, triangular, or Gaussian kernel, respectively, for the three cases just described.

The Cauchy-Schwarz distance can also be compared with the "correlation measure" between spike trains proposed by Schreiber et al. [2003]. In fact, it can be observed that the latter corresponds to the argument of the arc cosine and thus denotes the cosine of an angle between spike train, with norm and inner product computed with the mCI kernel estimator using the Gaussian kernel. Notice that Schreiber's et al. "correlation measure" is only a pre-metric since it does not verify the triangle inequality. But, in d_{CS} this is ensured by the arc cosine function.

A more detailed exposition of these inter-relationships can be found in the comparison study in Appendix B.

CHAPTER 4
A STATISTICAL PERSPECTIVE OF THE RKHS FRAMEWORK

This chapter provides an alternative perspective to the RKHS framework, namely, by verifying the construction of an RKHS as obtained from conventional statistical descriptors of interdependence. More specifically, it is shown the relation between cross-correlation and RKHS theory, especially noticeable when its generalized form presented here is compared to the mCI kernel.

The hopefully insightful perspective provided in this chapter has direct consequences for statistical analysis methods. This is exemplified in the second part of the chapter, by showing that the mCI kernel can be utilized to formulate and estimate the cross-intensity function (CIF) [Brillinger, 1976] and spike triggered average (STA) [Dayan and Abbott, 2001] of one point process with regards to the other. More that simply showing the relationship for the mCI kernel, we to aim to explicitly show the limitation of current approaches to the Poisson model, and incite further developments through the perspective provided here.

4.1 Generalized Cross-Correlation and the mCI Kernel

Binned point processes are discrete-time random processes. Therefore, as introduced in Section 2.4.2 for spike trains, the cross-correlation is defined in the usual way as the expectation of the lagged product of the number of events per bin. Hence, assuming ergodicity, the cross-correlation of binned point processes p_i and p_j is habitually estimated with

$$C_{ij}^{bin}[l] = \frac{1}{M} \sum_{n=1}^{M} N_{p_i}[n] N_{p_j}[n+l], \tag{4–1}$$

where M is the number of bins and $N_{p_i}[n]$, $N_{p_j}[n]$ are the number of events in the nth bin for point processes p_i and p_j, respectively. Equation 4–1 clearly shows that C_{ij}^{bin} is an inner product of the binned point processes. In RKHS theory the mapping into the RKHS is often unknown, but in this context it is readily noticeable that binning implements the mapping. However, binning of point processes discretizes the space of events and is therefore undesirable. This raises the question of what is binning actually doing? And, correspondingly, can we utilize a better way to do it?

In essence, binning estimates the density of events at a given time, that is, it attempts to estimate the instantaneous firing rate (apart from a normalization by the bin size) [Dayan and Abbott, 2001]. Hence, in its general form, cross-correlation can be defined directly in terms of the intensity functions of the point processes,

$$\begin{aligned} C_{ij}(\theta) &= E\left\{\lambda_{p_i}(t)\lambda_{p_j}(t+\theta)\right\} \\ &= \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{T} \lambda_{p_i}(t)\lambda_{p_j}(t+\theta)dt, \end{aligned} \tag{4–2}$$

where $\lambda_{p_i}(t)$ and $\lambda_{p_j}(t)$ denotes the intensity functions of point processes p_i and p_j, respectively. This is a *functional* inner product in an infinite dimensional space. We might think that C_{ij}^{bin} is finite dimensional approximation of this functional measure. We shall refer to this definition as the *generalized cross-correlation* (GCC) [Paiva et al., 2008], to distinguish from the binned counterpart.

In the statistical literature the conventional approach for intensity function estimation of point processes is kernel smoothing Reiss [1993], with clear advantages in the estimation Kass et al. [2003]. See Section 2.4.1.2 for a review on intensity estimation with kernel smoothing. So, if the event locations of a point process p_i in the event space $\mathcal{P}(\mathcal{T}) = [0, T]$ are denoted $\{t_m^i : m = 1, \ldots, N_i\}$, where N_i is the number of events of a realization of p_i, the kernel smoothed estimated intensity function is given by

$$\hat{\lambda}_{p_i}(t) = \sum_{m=1}^{N_i} h_\tau(t - t_m^i), \tag{4–3}$$

where h is the smoothing kernel function with size parameter τ. Substituting these intensity estimations in the definition of the generalized cross-correlation (Equation 4–2) and limiting the evaluation to the finite domain of the event space $[0, T]$ yields the estimator

$$\hat{C}_{ij}(\theta) = \frac{1}{T} \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \kappa_\tau \left(t_m^i - t_n^j + \theta\right), \qquad (4\text{–}4)$$

where κ_τ is the kernel obtained by the convolution of the intensity estimation kernel h with itself, and τ is the kernel size (or bandwidth) parameter. Notice that C_{ij}^{bin} is a special case of Equation 4–4 in which the spike times are first quantized and then the GCC evaluated with a rectangular kernel.

From our presentation it should be clear that the so-called GCC equals the mCI kernel, apart from the normalization for the width of the event space. This is clearly observable by comparing the GCC definition in Equation 4–2 with the mCI kernel definition in Equation 3–11, or alternatively from their estimators in Equation 3–27 and Equation 4–4.

Of immediate consequence this perspective suggests a direct replacement for (binned) cross-correlation in point process analysis, with spike train analysis in particular. As an example, this idea has been explored to construct continuous-time cross-correlograms for spike train analysis [Park et al., 2008] which benefit of the direct estimation on the spike times (i.e., the event coordinates), thus providing much higher precision, and in a fraction of the time required by explicitly smoothing.

Most importantly, the observation of this equivalence of the GCC to the mCI kernel reveals the limitations of current methodologies. This means that all *current cross-correlation methods have descriptive power at most equivalent to only the simplest of the cross-intensity kernel definitions given here.* The mCI kernel can accurately quantify at most interactions in the rate functions, equivalent to a inhomogeneous Poisson process model. On the other hand, verifying this close relationship brings forth that cross-intensity kernels are in fact cross-correlation operators for generalized point process models. Therefore, we believe CI kernels represent the future of point process analysis.

For spike train analysis, the kernel size in the mCI kernel has a particular useful interpretation in practice. Notice that Equation 4–3 can be interpreted as the convolution of the spike train with a window given by the smoothing function h, emulating the smoothing process in the neuron cell membrane. Therefore, the size parameter τ determines the smoothing introduced by h and the kernel κ, and thus regulates the scale at which the mCI kernel (or GCC) estimator interprets the neuronal coupling expressed in the intensity function, between the extremes of synchrony in neuron firings (for small kernel size) or firing rate (large kernel size).

4.2 Relevance for Statistical Analysis Methods

4.2.1 Relation to the Cross Intensity Function

The *cross-intensity function* (CIF) is the second-order association moment between two point processes. It was originally proposed by Brillinger [1976] and was applied to spike train analysis by Brillinger and colleagues (Brillinger [1992] and references therein) and others [Hahnloser, 2007].

Statistically, the cross-intensity function (CIF) is the conditional probability of an event occurring at a given location in the event space for a point process p_j given the occurrence of an event in the conditioning point process p_i at some specific location. It is defined as

$$\vartheta_{p_j|p_i}(\theta) = \lim_{\delta\to 0^+} \frac{1}{\delta} \Pr[N_B(\theta + t_k, \theta + t_k + \delta) = 1 | t_k \in p_i], \tag{4–5}$$

where N_B is the counting process associated with p_j and $t_k \in p_i$ expresses that t_k is an event of a realization of p_i.

Naturally, the conditional formulation of CIF means that $\vartheta_{p_j|p_i}(\theta)$ is not a symmetric function. In fact, noting that the (instantaneous) intensity function of an inhomogeneous Poisson process p_j can be written as

$$\lambda_{p_j}(t) = \lim_{\delta\to 0^+} \frac{1}{\delta} \Pr[N_B(t, t+\delta) = 1], \tag{4–6}$$

leads to the observation that the definition of CIF defines a conditional intensity function,

$$\vartheta_{p_j|p_i}(\theta) = \lambda_{p_j}(\theta + t_k | t_k \in p_i), \tag{4–7}$$

where t_k is the "closest" event of p_i to θ.

From Equation 4–7 results that CIF can be written as

$$\begin{aligned} \vartheta_{p_j|p_i}(\theta) &= \lambda_{p_j}(\theta | p_i) \\ &= E_{t_m^A \in p_i}\left\{\lambda_B(\theta + t_m^A)\right\} \\ &\approx \frac{1}{N_A}\sum_{m=1}^{N_A} \lambda_B(\theta + t_m^A). \end{aligned} \tag{4–8}$$

Estimating the intensity function of p_j from a realization with kernel smoothing

$$\hat{\lambda}_B(t) = \sum_{n=1}^{N_B} h(t - t_n^B), \tag{4–9}$$

and substituting this estimate in Equation 4–8, yields an estimator for CIF

$$\hat{\vartheta}_{p_j|p_i}(\theta) = \hat{\lambda}_B(\theta|p_i) = \frac{1}{N_A}\sum_{m=1}^{N_A}\sum_{n=1}^{N_B} h(\theta + t_m^A - t_n^B). \tag{4–10}$$

This equation clearly shows that $\vartheta_{p_j|p_i}(\theta)$ is the intensity function induced in p_j through the occurrence of an event in p_i. Note that the error in this estimator depends only on the estimation of the intensity function of p_j and the expectation over events of p_i. In other words, this estimator is unbiased since both operations can be done exactly for infinite data.

Conversely, similar arguments can be employed to derive that

$$\hat{\vartheta}_{p_i|p_j}(\theta) = \hat{\lambda}_A(\theta|p_j) = \frac{1}{N_B}\sum_{m=1}^{N_B}\sum_{n=1}^{N_A} h(\theta + t_m^B - t_n^A). \tag{4–11}$$

Comparing Equation 4–10 and Equation 4–11 with the mCI kernel estimator in Equation 3–27 it is possible to verify that they are fundamentally the same expect for a scaling (by $1/N_A$), the introduction of a lag parameter, and the use of the smoothing kernel directly (instead of its autocorrelation).

4.2.2 Relation to the Spike Triggered Average

An alternative interpretation that stems from Equation 4–7 is that CIF can be thought of as a event-triggered expectation of the intensity function of p_j around events in p_i. That is,

$$\begin{aligned} \vartheta_{p_j|p_i}(\theta) &= \lambda_{p_j}(\theta + t_k | t_k \in p_i) \\ &= E_{\forall t_k \in p_i}\left\{\lambda_{p_j}(\theta + t_k)\right\} \\ &\approx \frac{1}{N_A}\sum_{m=1}^{N_A} \lambda_{p_j}(\theta + t_m^A). \end{aligned} \tag{4–12}$$

where $E_{\forall t_k \in p_i}\{\cdot\}$ denotes the expectation over all possible events of p_i. This shows the equivalence to the CIF and therefore that the mCI kernel can also be utilized for estimation.

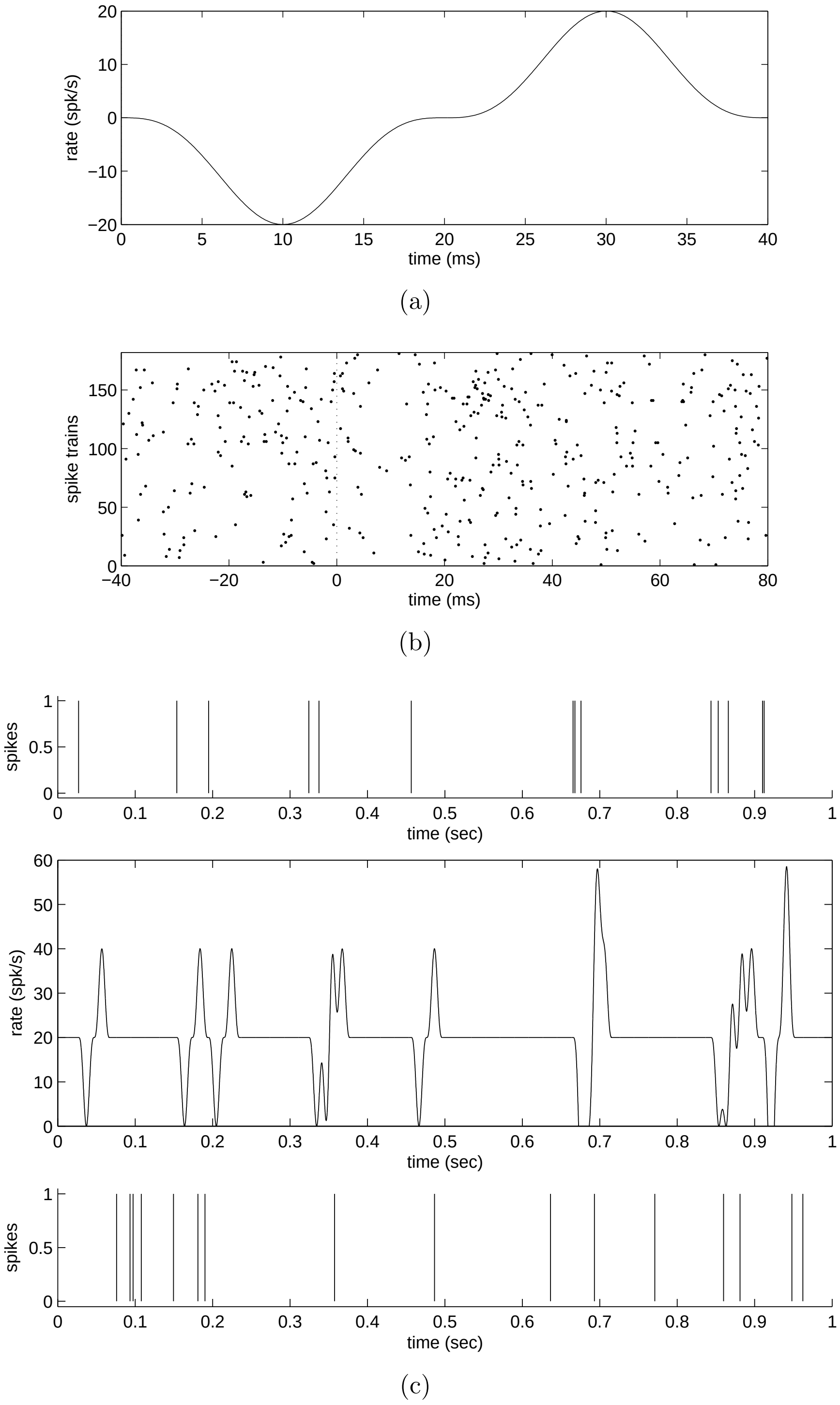
In neurophysiological studies this corresponds to what is called the *spike-triggered average* (STA) [Dayan and Abbott, 2001]. Simply put, STA is a peri-event diagram of a continuous quantity in which the synchronizing events are the firing (i.e., spikes) from a neuron.

It must the remarked that both the CIF and STA are concepts limited for data analysis to the descriptive power of intensity functions, and thus to Poisson models, as can be expected from the close relationship to the mCI kernel. However, as noted earlier about the relationship between the mCI kernel and the GCC, this perspective suggests that the cross-intensity kernels could be utilized to greatly extend these concepts beyond Poisson point processes.

4.2.3 Illustration Example

The relationships just discussed theoretically are now illustrated through a simple simulation. The simulated example was crafted to replicate the dataset of L3 and L10 neurons from experiments with *Aplysia* utilized by Brillinger [1992].

Two 10 second-long spike trains were generated as Poisson processes. p_i was generated as an inhomogeneous Poisson process with rate 20 spk/s, and was used as



(a)

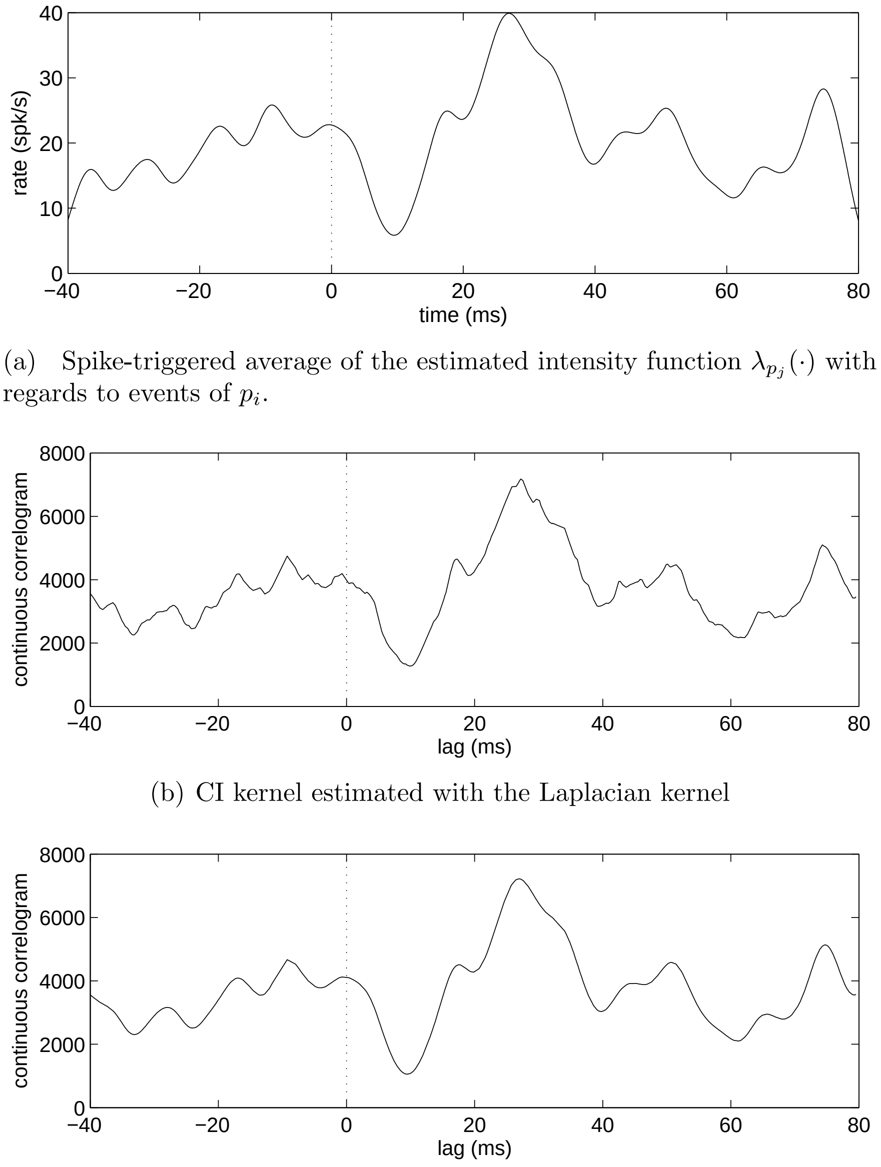
(b)

(c)

Figure 4-1. (a) Modulation in firing induced in p_j through the spikes in p_i. (b) Spikes in time of p_j around the occurence of spikes in p_i (marked by the vertical dotted line). (c) First second of reference spike train, p_i, intensity function of p_j with effects induced by p_i, and corresponding realization of p_j.

the reference spike train (equivalence to a L10 neuron). The goal of the CIF function is to study cross-neuron induced modulations in the intensity function. More specifically, whenever a spike occurs in p_i it introduces a modulation (shown in Figure 4-1(a) for this simulation) in the intensity function of p_j (equivalent to a L3 neuron). Figure 4-1(c) depicts this mechanism, and can be perceived in the resulting spike trains shown in Figure 4-1(b).

As introduced in the previous sections, and shown by Equation 4–7, the CIF corresponds to an average of the intensity function of p_j with respect to the spikes in p_i. This is shown in Figure 4-2(a). Correspondingly, the mCI kernel as a function of the lag evaluated for the two spike trains yields the same (scaled) result. These are shown in Figure 4-2(b)–(c). Notice that if the mCI kernel result is divided by the average number of spikes of p_i in a spike train (≈ 200 spikes $= (20 \text{ spk/s}) \times (10\text{s})$, cf. Equation 4–10) yields an average firing rate of 20 spk/s, as expected. Finally, if one estimates the reverse condition, that is, $\vartheta_{p_i|p_j}(\theta)$, Equation 4–11 proves that from an estimation standpoint this is merely a matter of mirroring the, mCI kernel result, if the intensity estimation smoothing function is symmetric (as is imposed by the elementary kernel used to estimate the mCI kernel). The causal relationship between the neurons is immediately apparent from the causality of the mCI kernel as a function of the lag.



(a) Spike-triggered average of the estimated intensity function $\lambda_{p_j}(\cdot)$ with regards to events of p_i.

(b) CI kernel estimated with the Laplacian kernel

(c) CI kernel estimated with the Gaussian kernel

Figure 4-2. (a) Average intensity function estimated from the "trials" shown in Figure 4-1(b). It corresponds to a spike-triggered average of the intensity function of p_j with regards to the spikes in p_i. (b)–(c) CI kernel as a function of the lag estimated with Laplacian and Gaussian kernels respectively.

CHAPTER 5
APPLICATIONS IN NEURAL ACTIVITY ANALYSIS

As unequivocally stated in this dissertation's title and detailed in chapter 1, this work can be directly applied for neural activity analysis, specifically on spike train analysis. Following the considerations presented in Chapter 4, we now study the application of these ideas for spike train analysis.

In essence, this chapter provides some immediate developments for spike train analysis for use by the practitioner. For ease of direct comparison to current techniques we will base our presentation in the generalized cross-correlation (GCC), but recall that GCC and the mCI kernel are fundamentally the same apart from the normalization. Therefore, the considerations for future improvements are equally applicable, pending on future developments on conditional intensity estimation.

5.1 Generalized Cross-Correlation as a Neural Ensemble Measure

By its very nature, a spike train is realization of a point process (Section 2.3). Therefore it should seem obvious that all the theory presented before can be applied, as a specific application, for spike train analysis. In this section, some ideas regarding the use of GCC for spike train data analysis are put forward. Even in this case, the perspective presented in Chapter 4 allows for developments obscured by the common presentation found in the literature.

Before proceeding, it must be remarked for this application the meaning of the size parameter of the smoothing function, or correspondingly of the kernel κ utilized in the GCC estimator. In this case the size parameter has a well defined physical meaning; it selects the time scale at which the analysis is to be performed. In other words, the size parameter is to be selected according to the firing characteristics known a priori of the neuron and/or the feature of interest for the analysis at hand. An important consequence of the use of kernel smoothing for intensity estimation in this framework is that it seamlessly integrates the differences between spike rates and spike times without discretization of time. Put differently, the use of kernel smoothing makes it easy to zoom into the feature of interest and puts the focus on the time structure of the spike trains as the central parameter been quantified as spike train similarity,

This characteristic can be very useful in spike train analysis, for example, to measure synchrony between spike trains. In computational neuroscience one of the commonly used descriptors of the relation between two spike trains is synchrony. It is obvious that since the information of spike trains is contained in the spike times, synchrony quantifies this relationship somewhat even though there is no metric assigned. However, it is not totally fulfilling as the different definitions in the literature demonstrate: synchrony [Freiwald et al., 2001], synchrony at a lag [Lindsey and Gerstein, 2006], polychronization [Izhikevich, 2006]. For this reason, the definition of synchrony can be substituted by the general concept of similarity as measured by GCC (or the mCI kernel) as proper time-scale. Moreover, the use of point process distances between two spike trains as given in Section 3.6 allows for a full featured metric space if necessary.

To measure similarity between spike trains the GCC estimator in Equation 4–4 is used. Like any estimator, the evaluated value is a random variable which approaches the expected value as more data becomes available. On the other hand, from a practical standpoint the length of the recording is often limited. Anyway, it is desirable to keep the integration interval to a minimum for improved resolution.

We propose to solve this problem through ensemble averaging. If M denotes the number of ensemble spike trains under analysis, the ensemble averaged GCC is,

$$\bar{C}(\theta) = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \hat{C}_{ij}(\theta). \tag{5–1}$$

In this way, the integration interval can be reduced as the number of spike trains increases without sacrifice of the statistical accuracy. In general, the above equation depends on the lag (as the usual cross-correlation), but for the analysis done next the zero lag shall be considered. This corresponds to the situation of synchrony. In practice, one might need to time align the spike trains by first estimating the lag using the continuous cross-correlogram (CCC) [Park et al., 2008], for example.

Of course, an important question that must be considered is which spike trains should be averaged together as constituents of the same ensemble. The clustering algorithm presented in Chapter 6 can be of use to answer this question.

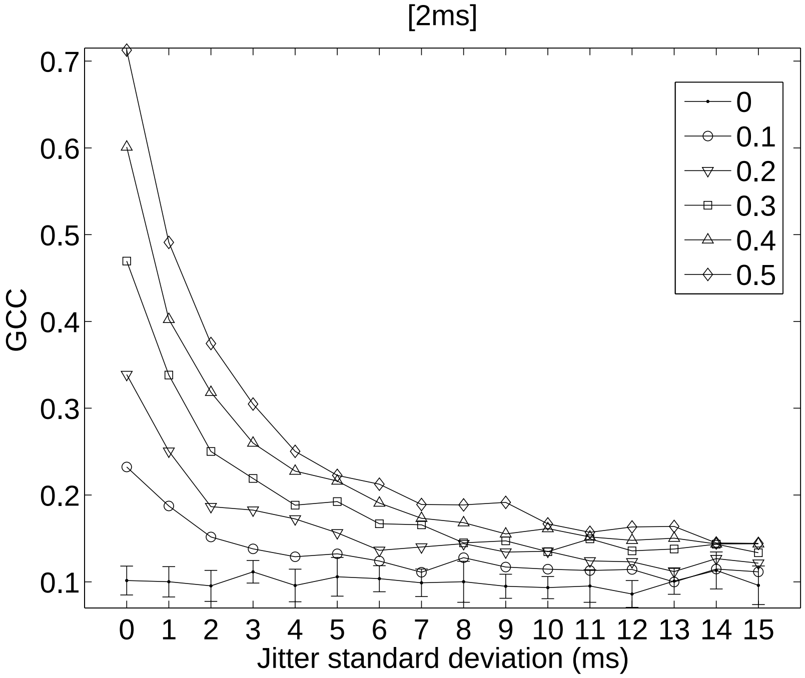
5.2 Empirical Analysis of GCC Statistical Properties

The statistical properties of GCC with regards to jitter in the spike timings and the number of neurons are now analyzed. The behavior of GCC with respect to these two parameters is very important for spike train analysis, especially in synchrony studies.

In the following examples there is the need to generate simulated spike trains under different synchrony (or correlation) conditions. Synchronous spike trains were generated using the multiple interaction process (MIP) proposed by Kuhn et al. [2003, 2002]. In the MIP model an initial spike train is generated as a realization of a Poisson process. All spike trains are derived from this one by copying spikes with a probability ε. The operation is performed independently for each spike and for each spike train. The resulting spike trains are also Poisson processes. If γ was the firing rate of the initial spike train then the derived spikes trains will have firing rate $\varepsilon\gamma$. Furthermore, it can be shown that ε is also the count correlation coefficient [Kuhn et al., 2003]. A different interpretation for ε is that, given a spike in a spike train, it quantifies the probability of a spike co-occurrence in another spike train. In this sense, we shall refer to ε as the synchrony level. Note that an alternative manner of quantifying synchrony could be through the CS distance, in which a distance of zero corresponds to perfect synchrony (i.e., $\varepsilon = 1$).

5.2.1 Robustness to Jitter in the Spike Timings

In a physiological context the idea of precisely synchronous spikes is unlikely to be found. Thus, it is important to characterize the behavior of the GCC estimator when jitter is present in the spike timings. This was done with a modified MIP model where jitter, modeled as zero-mean independent and identically distributed (i.i.d.) Gaussian



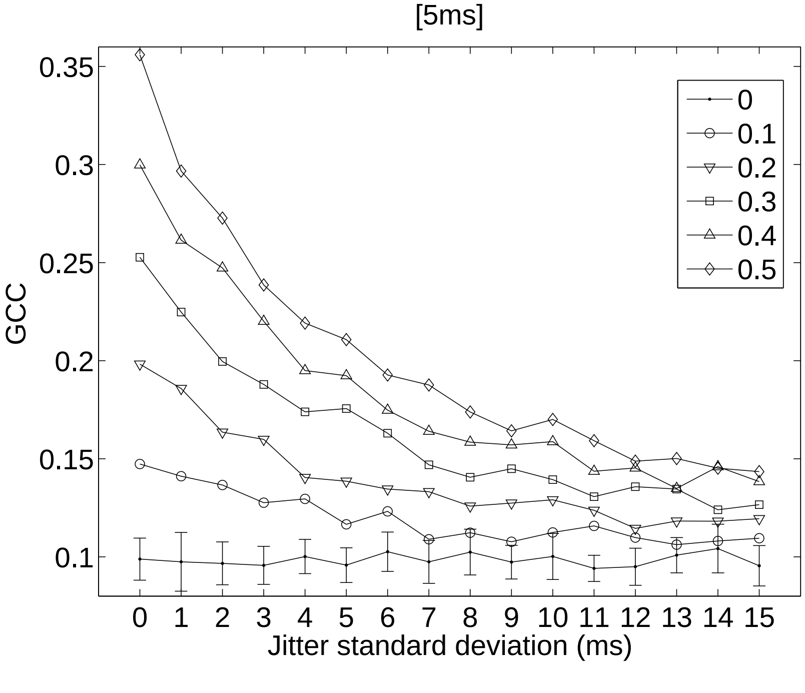


Figure 5-1. Change in CIP versus jitter standard deviation in synchronous spike timings. For the case with independent spike trains, the error bars for one standard deviation are also shown. The estimation kernel κ was the Laplacian kernel with size 2ms (top) and 5ms (bottom).

noise, was added to the individual spike timings. The effect was then studied in terms of the synchrony level and kernel size (of κ). Figure 5-1 shows the average GCC for 10 Monte Carlo runs of two spike trains, 10 seconds long, and with constant firing rate of 20 spikes/s. In the simulation, the synchrony level was varied between 0 (independent) to 0.5 (i.e., 50% of spikes were synchronous), and for a kernel size of 2ms and 5ms. The jitter standard deviation varied between the ideal case (no-jitter) to 15ms.

For a small estimator kernel size the GCC estimator measures the coincidence of the spike timings. As a consequence, the presence of jitter in the spike timings decreases the expected value of GCC. Nevertheless, the results in Figure 5-1 support the statement that the measure is indeed robust to large levels of jitter when compared to the kernel size, and is capable of detecting the existence of synchrony among neurons. Of course, increasing the kernel size decreases the sensitivity of the measure for the same amount of jitter. Furthermore, it is also shown that even small levels of synchrony can be statistically discriminated from the independent case as suggested by the error bars in the figure. (The difference in scale between the figures is a consequence of the normalization of κ, which depends on the kernel size.)

5.2.2 Sensitivity to Number of Neurons

The effect of the number of spike trains used for ensemble averaging is now analyzed. This effect was studied with respect to two main factors: the synchrony level of the spike trains and the kernel size of the GCC estimator κ. In the first case, the kernel size was 2ms, whereas in the second case considered only independent spike trains. The results are shown in Figure 5-2 for the estimated GCC averaged over all pair combinations of neurons. The simulation was repeated for 1000 Monte Carlo runs using 1 second long spike trains simulated as homogeneous Poisson processes with firing rate 20 spikes/s.

As illustrated in the figure, the variance in the GCC estimator decreases dramatically with the increase in the number of spike trains employed in the analysis. Recall that the number of pair combinations over which the averaging is performed increases with

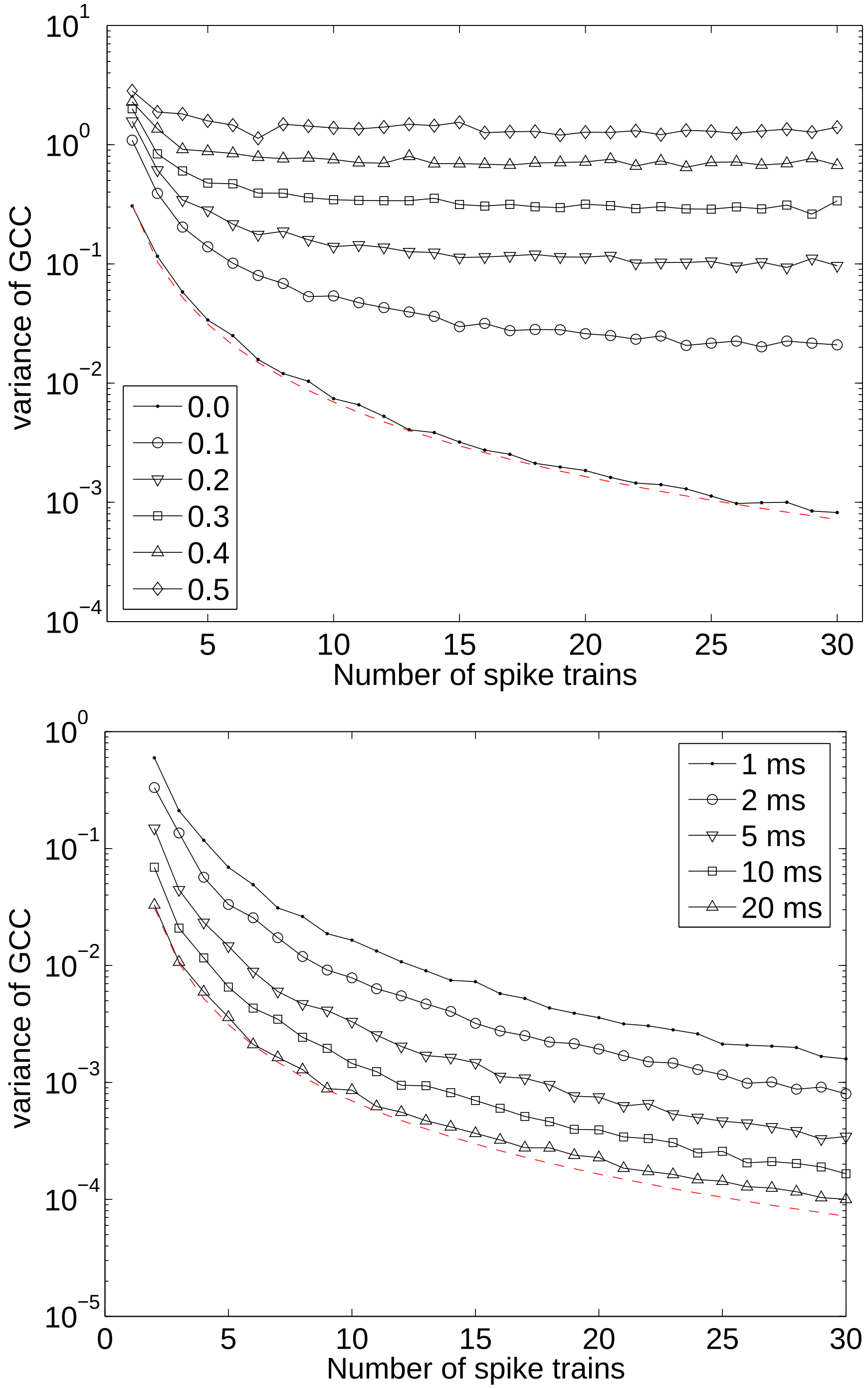


Figure 5-2. Variance (in log scale) of GCC versus the number of spike trains used for spatial averaging. The estimation kernel κ was the Laplacian kernel. (top) The analysis was performed for different levels of synchrony with kernel size 2ms, and (bottom) for different values of the kernel size for independent spike trains. In both situations the theoretical value of GCC for independent spike trains is shown (dashed line).

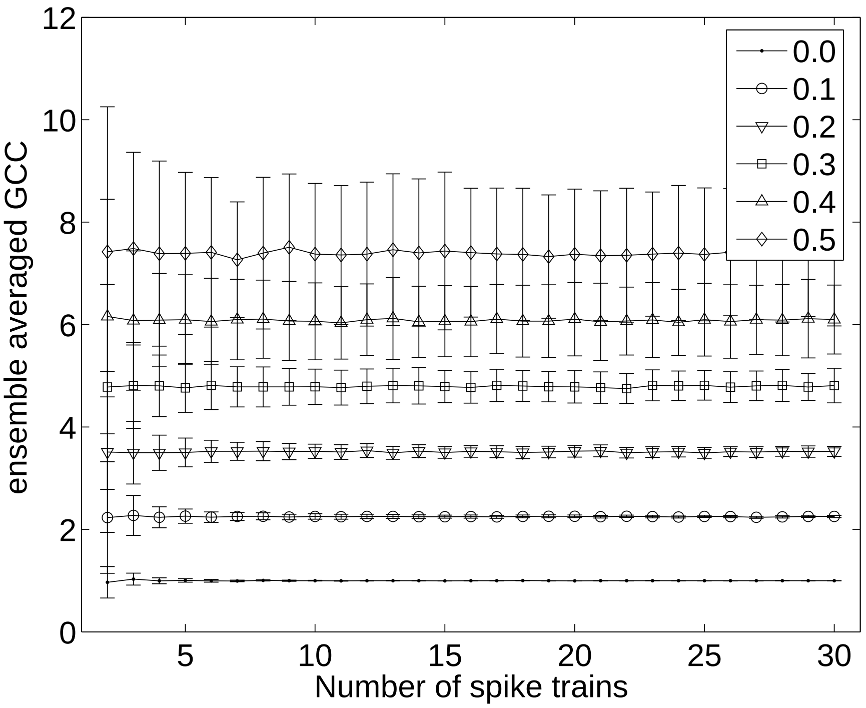


Figure 5-3. Mean and standard deviation of GCC versus the number of spike trains used for spatial averaging for different synchrony levels, corresponding to the first scenario in Figure 5-2.

$M(M-1)$, where M is the number of spike trains. As expected, this improvement is most pronounced in the case of independent spikes trains. In this situation, the variance decreases proportionally to the number of averaged pairs of spike trains. This is shown by the dashed line in the plots of Figure 5-2. Whenever the spike trains are correlated, the improvement on the variance of the estimator is smaller due to a non-ideal averaging situation, reaching a nearly extreme situation for $\varepsilon = 0.5$ where ensemble averaging is almost useless. In any case, such high values of synchrony seem unlikely to be found in neurophysiological experiments. These results support the role and importance of ensemble averaging as a principled method to reduce the variance of the GCC estimator.

Finally, the sensitivity of GCC to the synchrony level should be remarked. In Figure 5-3 the standard deviation was superimposed to the ensemble averaged GCC. It is observable a clear distinction between, at least, the four smaller synchrony levels, i.e., $\varepsilon \in [0, 0.3]$. This means that the GCC estimator has a high degree of accuracy in this interval when averaged over a number of neurons as small as 4, supporting our claim that GCC can be used as a synchrony index.

5.3 Instantaneous Cross-Correlation

The GCC is a more general form of cross-correlation that does not require binning but it still needs a finite interval of data to operate. It is therefore still dependent on an ergoricity assumption. As a function of time, the integrand of the GCC (Equation 4–2), which we shall refer to as the *instantaneous cross-correlation* (ICC), provides a more appropriate representation. ICC is a continuous function of the spike timings and describes temporal structure of the inhomogeneous firings allowing for a direct assessment of similarity in time. One might think of it as a scalar inner product along each of the dimensions indexed by time. Therefore, the ICC is defined as

$$\tilde{c}_{ij}(t,\theta) = \hat{\lambda}_{p_i}(t)\hat{\lambda}_{p_j}(t+\theta), \tag{5–2}$$

where $\hat{\lambda}_{p_i}(t)$, $\hat{\lambda}_{p_j}(t)$ are the estimated intensity functions from spike trains corresponding to point processes p_i and p_j.

For methodologies that can be applied online, only causal intensity estimation smoothing functions can be considered. We propose to use the exponential function,

$$h(t) = (1/\tau)\exp\left[-t/\tau\right] u(t), \tag{5–3}$$

where $u(\cdot)$ is the step function. The exponential function provides both graded interactions and a time scale for the intensity estimation by controlling the time constant τ. Of course the ideas to be presented are not limited to the decaying exponential smoothing function, but it was chosen for its biological plausibility, since it can be interpreted as evoked post-synaptic potentials in a neuron, its wide use throughout neuroscience Dayan and Abbott [2001], and its computational simplicity, since computing the next value depends only on the present value and if a spike occurs in the meantime.

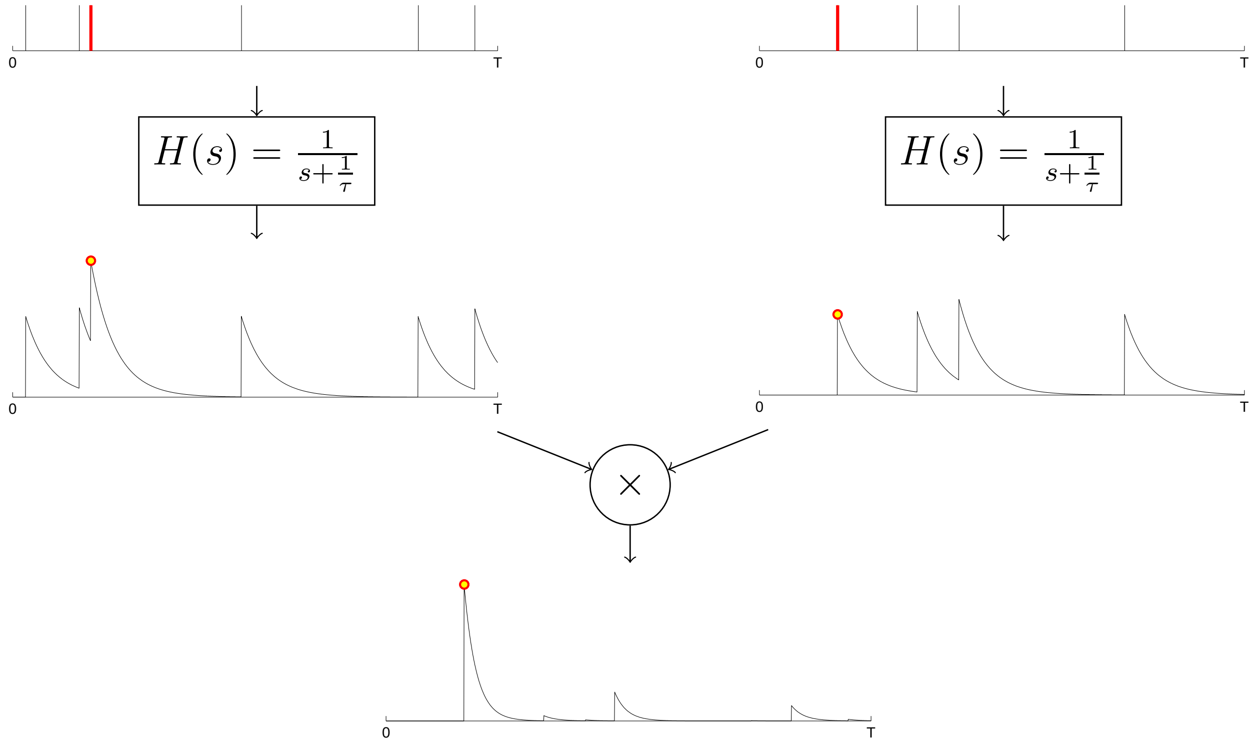

0
T
0
T
$H(s) = \frac{1}{s+\frac{1}{\tau}}$
$H(s) = \frac{1}{s+\frac{1}{\tau}}$
0
T
0
T
$\times$
0
T

Figure 5-4. Diagram outlining the idea and procedure for the computation of the ICC. On top it is shown two spike trains for which the ICC is to be computed, followed by the intensity estimation with the decaying exponential function (represented by $H(s)$). The two estimated intensity functions are then multiplied together to obtain the ICC. The position of synchronous spikes is marked as red circles in the figure.

Using the exponential function, the intensity function at time t estimated from a spike train is

$$\hat{\lambda}_{p_i}(t) = \frac{1}{\tau} \sum_{t_m^i \leq t} \exp\left(-\frac{t - t_m^i}{\tau}\right) u(t - t_m^i). \tag{5–4}$$

This is nothing but the filtering of a spike train by a first order IIR filter. Then, the ICC can be computed by instantaneously multiplying the two estimated intensity functions. Notice that this two layer evaluation process can be computed very easily, and is especially suited for hardware implementation.

For small values of the size parameter τ the ICC quantifies statistically our intuition of synchrony, graded by the decaying exponential function and followed by a coincidence detection operator implemented by the product. When two neurons spike synchronously the product of the estimated intensities at that time will be high, with a maximum if they spike exactly at the same time, but if the spikes are separated by more than $\approx 5\tau$ then the ICC is nearly zero (Figure 5-4). In this respect, the ICC resembles the "gravity force" in the gravity transform framework Gerstein et al. [1985]; Gerstein and Aertsen [1985], but the present work provides a statistical interpretation for the estimator and much broader perspective not available before.

5.3.1 Stochastic Approximation of GCC

As the formulation of ICC suggests, $\tilde{c}_{ij}$ is a stochastic approximation of the GCC under ergodicity. This is easily verified by taking the expectation of Equation 5–2 over time. In particular, the average ICC over a time interval $[0, T]$ with a exponential function results is

$$\frac{1}{T}\int_0^{\infty} \tilde{c}_{ij}(t,\theta)dt = \frac{1}{T\tau^2}\sum_{m=1}^{N_i}\sum_{n=1}^{N_j}\int_{\max(t_m^i, t_n^j)}^{\infty} \exp\left[-\frac{|t_m^A - t_n^B + \theta|}{\tau}\right], \tag{5–5}$$

where the integration goes up to infinity to account for the infinite support of the exponential function but only spike times in the interval $[0, T]$ are included. The evaluation of the integral involves determining which spike firing, t_m^i or t_n^j, occurs later to determine the effective lower integration limit. Solving the integral for both situations (i.e., $t_m^i \leq t_n^j$ or $t_m^i > t_n^j$), however, allows to verify that the difference between the time instants in both situations is positive, which can be summarized in the form of the Laplacian kernel. That is,

$$\begin{aligned} \frac{1}{T}\int_0^{\infty} \tilde{c}_{AB}(t,\theta)dt &= \frac{1}{T}\sum_{m=1}^{N_A}\sum_{n=1}^{N_B}\frac{1}{2\tau}\exp\left(-\frac{|t_m^A - t_n^B + \theta|}{\tau}\right) \\ &= \frac{1}{T}\sum_{m=1}^{N_A}\sum_{n=1}^{N_B}\kappa_\tau\left(t_m^A - t_n^B + \theta\right) \\ &= \hat{C}_{AB}(\theta), \end{aligned} \tag{5–6}$$

where, in this case, κ_τ denotes the Laplacian kernel. Note that the exponential function gives rise to the Laplacian kernel which verifies all the requirements for $\hat{C}_{ij}$ to represent a well defined inner product. If, for example, a Gaussian function of bandwidth σ had been used as the smoothing function for intensity estimation then the resulting kernel κ would also be a Gaussian kernel with bandwidth $\sqrt{2}\sigma$. However, with the Gaussian function we would loose the important advantages of ease of computation and causality.

5.3.2 ICC as a Neural Ensemble Measure

The ICC exploits the temporal nature of the spike trains and enables instantaneous estimation of synchrony because no temporal averaging is done. The price paid is that, for a single pairs of neurons, variability in the spike times is directly translated into the ICC and thus its estimation is quite "noisy" due to events occurring by chance. Instead of averaging ICC over time which yields the GCC in a time interval, an alternative way to reduce the variance of this estimator is to compute the expectation over the neural ensemble,

$$\bar{c}(t,\theta) = E\left\{\tilde{c}_{AB}(t,\theta)\right\}, \tag{5–7}$$

where $E\left\{\cdot\right\}$ denotes the expectation over all pairs of neurons.

The ensemble averaged ICC is a spatio-temporal measure of the ensemble cooperation over time. In this form, and due to the exchange of time for ensemble averaging, the ICC is capable of detecting the presence of dynamic cell assemblies in the ensemble with high temporal resolution. However, as in Section 5.1, it raises the problem of neural selection to evaluate the ensemble average.

5.3.3 Data Examples

Three examples of the application of ICC are now presented. The first two are in simulated paradigms and the third in a recording of motor neurons from the M1 cortex of rat performing a behavioral task. In these examples the analysis is focused on synchrony mainly because it is an application that naturally takes advantage of the high resolution in time of the ICC, but we remark that ICC could also be utilized for studies of correlations in the firing rates in principle.

5.3.3.1 ICC as a synchronization measure

The main goals of this example are: first, to show that the mean value of ICC is sensitive to the synchrony level on that data, second, that this measurement is effective for single-realizations, and, finally, to showcase the use of GCC as a synchrony index; in other words, a descriptor of the synchrony level.

For this example, we generated 10 homogeneous spike trains using the multiple interaction process (MIP) Kuhn et al. [2003]. The MIP model allows for multiple spike trains to be generated according to a selected synchrony level, ε, which is the count correlation coefficient and quantifies the probability of a spike co-occurrence in another spike train.

Figure 5-5 shows *one realization* of the generated spike trains with varying levels of synchrony. All simulated spike trains have average firing rate 20 spikes/s. The figure shows the ICC averaged for each time instant over all pair combinations of spike trains. The time constant, τ, of the exponential for intensity estimation was chosen to be 2ms. To verify Equation 5–6, the bottom plot shows the average value of the mean ICC. This was computed with a causal 250ms long sliding window in 25ms steps. To establish a relevance of the values shown, the expectation and the expectation plus two standard deviations are also shown, assuming independence between spike trains. The mean and standard deviation, assuming independence, are 1 and $\sqrt{\left(\frac{1}{2\tau\lambda}+1\right)^2-1}$, respectively. The expected value of the ICC for a given synchrony level is $1+\varepsilon/(2\tau\lambda)$, with λ the firing rate of the two spike trains, and is also shown in the plot for reference. Finally, the ensemble averaged GCC computed for each second of data is also shown.

It is noticeable from the figure that the ICC estimated synchrony increases as measured by ICC. Moreover, the averaged ICC is very close to the theoretical expected value and is typically below the statistical upper bound under an independence assumption as given by the line indicating the expectation plus two standard deviations. The delayed increase in the averaged ICC is a consequence of the causal averaging of ICC. It is equally

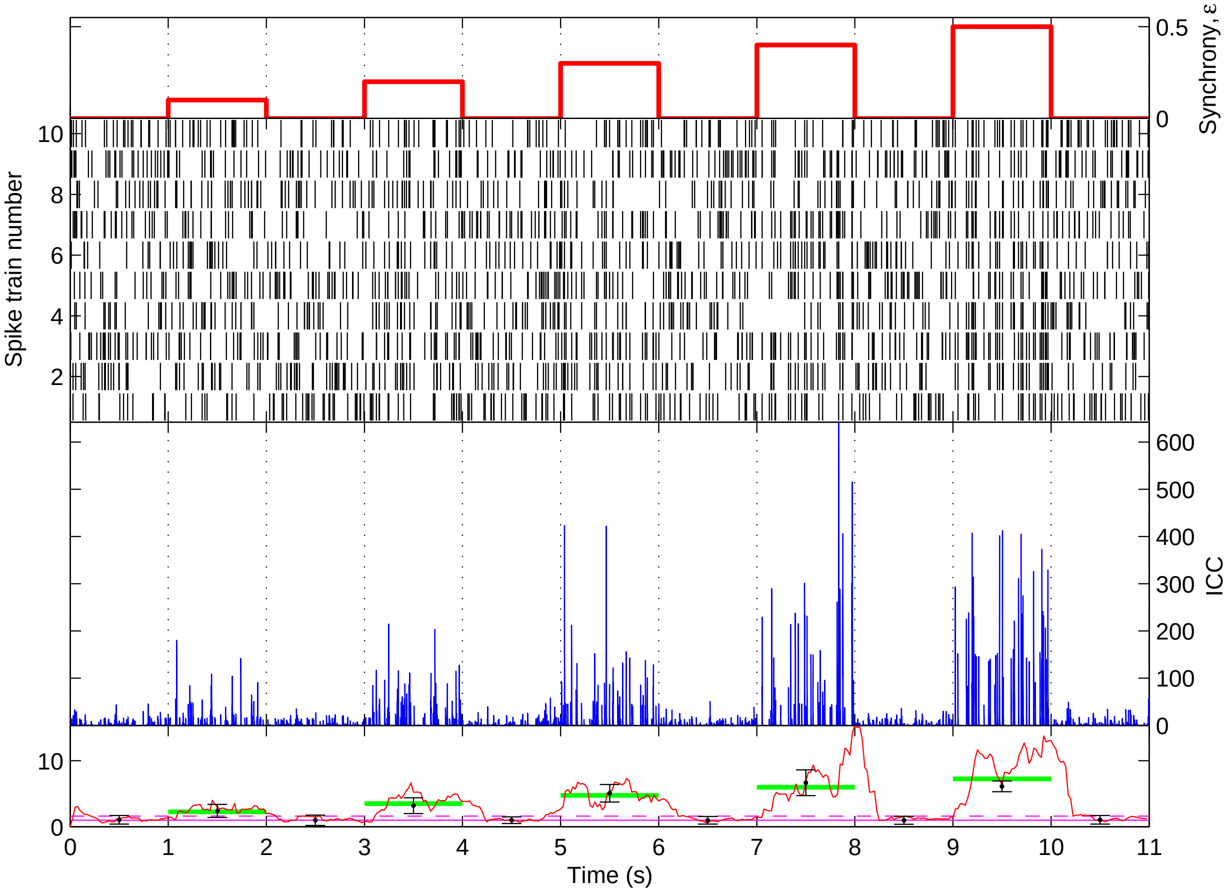


Figure 5-5. Analysis of the behavior of ICC as a function of synchrony in simulated coupled spike trains. (Top) Level of synchrony used in the simulation of spike trains. (Upper middle) Raster plot of firings. (Lower middle) Ensemble averaged ICC. (Bottom) Time average of ICC in the upper plot computed with a causal rectangular window 250ms long in steps of 25ms (dark gray). For reference, it is also displayed the expected value (dashed line) and this value plus two standard deviations (dotted line) for independent neurons, together with the expected value during moments of synchronous activity (thick light gray line), as obtained analytically from the level of synchrony used in the generation of the dataset. Furthermore, the mean and standard deviation of the ensemble averaged GCC scaled by T measured from data in one second intervals is also shown (black).

remarkable to verify that GCC matches precisely the expected values from ICC as given analytically. This shows a significant advantage of the GCC/ICC as it can be used for analysis of data providing not only detection ability but also the possibility to actually measure the synchrony level with a high degree of accuracy.

5.3.3.2 Synchronization of pulse-coupled oscillators

In this example, we show that ICC can quantify synchrony in a spiking neural network of leaky-integrate-and-fire (LIF) neurons designed according to Mirollo and Strogatz Mirollo and Strogatz [1990][1] and the ICC results compare favorably with the extended cross-correlation for multiple neurons. The network is initialized in a random condition and is proven to synchronize over time (Fig. 5-6). The synchronization is essentially due to leakiness and the weak coupling among the oscillatory neurons.

The raster plot of neuron firings is shown in Fig. 5-6. There are two main observations: the progressive synchronization of the firings associated with the global oscillatory behavior of the network, and the local grouping that tends to preserve local synchronizations that either entrain the full network or wash out over time, as expected from theoretical studies of the network behavior Mirollo and Strogatz [1990]. The ICC depicts this behavior precisely: the synchronization increases monotonically, with a period of fast increase in the first second followed by a plateau and slower increase as time advances. Moreover, it is possible to observe in the first 1.5s the formation of a second group of synchronized neurons which slowly merges into the main group. In addition, the envelope of ICC reveals the coherence in the membrane potentials quantified by the information potential (IP). The IP is an information theoretic quantity inversely proportional to

[1] The parameters for the simulation are: 100 neurons, resting and reset membrane potential -60 mV, threshold -45 mV, membrane capacitance 300 nF, membrane resistance 1 MΩ, current injection 50 nA, synaptic weight 100 nV, synaptic time constant 0.1 ms and all to all excitatory connection.

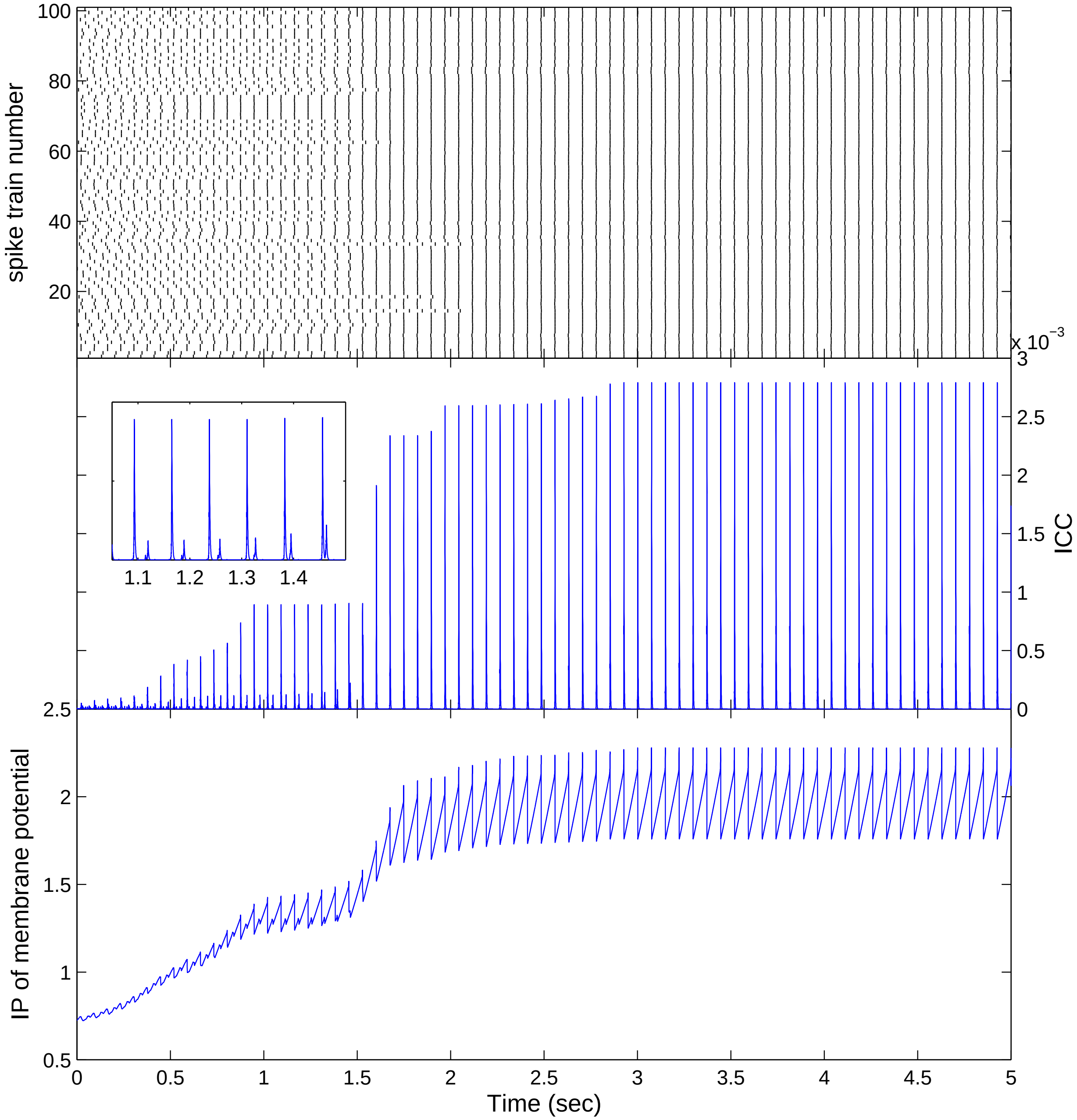


Figure 5-6. Evolution of synchrony in a spiking neural network of pulse-coupled oscillators. (Top) Raster plot of the neuron firings. (Middle) ICC over time. The inset highlights the merging of two synchronous groups. (Bottom) Information potential of the membrane potentials. This is a macroscopic variable describing the synchrony in the *neurons' internal state*.

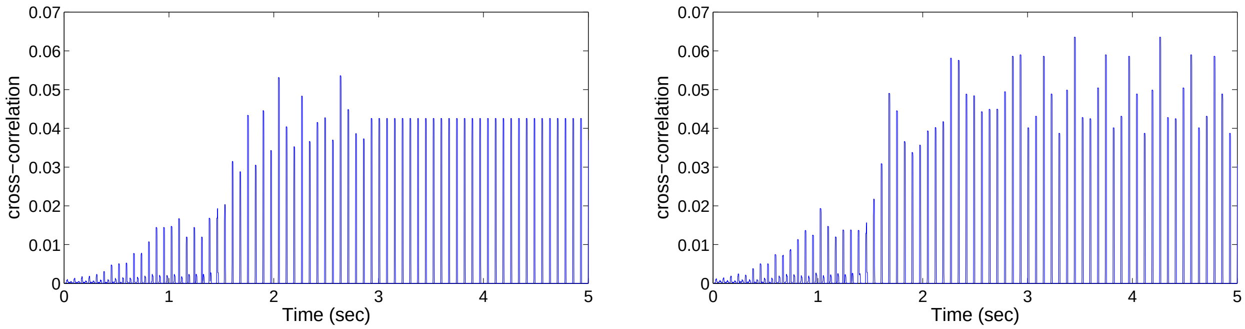


Figure 5-7. Zero-lag cross-correlation computed over time using a sliding window 10 bins long, and bin size 1ms (top) and 1.1ms (bottom).

entropy Príncipe et al. [2000]. It was computed with

$$\mathrm{IP}_\theta = \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\exp(-d(\theta_i,\theta_j)/2\sigma^2) \tag{5–8}$$

with $\sigma = 75\mathrm{mV}$.[2] The IP measures synchrony of the neuron's internal state, which is only available in simulated networks. Yet the results show that ICC was able to successfully and accurately extract such information from the observed spike trains.

In Fig. 5-7 we also present the zero-lag cross-correlation over time, averaged through all pairwise combinations of neurons. The cross-correlation was computed with a sliding window 10 bins long, sliding 1 bin at a time. Results are shown for bin sizes of 1ms and 1.1ms. It is notable that although cross-correlation captures the general trends of synchrony, it masks the plateau and the final synchrony and it is highly sensitive to the bin size as shown in the figure, unlike ICC. In other words, the results for the windowed cross-correlation highlight the importance of working in "continuous" time and without time averaging for robust spike train analysis.

[2] The distance used in the Gaussian kernel was $d(\theta_i,\theta_j) = \min\left(|\theta_i - \theta_j|, 15mV - |\theta_i - \theta_j|\right)$, where θ_i is the membrane potential of the ith neuron. This wrap-around effect expresses the phase proximity of the neurons before and after firing.

5.3.3.3 Analysis of neural synchronous activity in motor neurons

In this last example, the ICC is utilized to analyze the presence of synchronous activity in the motor cortex of a rat's brain. Throughout the literature, synchronous activity has been shown to provide additional information about motor movement when compared to firing rate modulation pattern analysis alone, and including when no firing rate modulations are noticeable [Vaadia et al., 1995; Hatsopoulos et al., 1998; Riehle et al., 1997]. Indeed, synchronous neural activity seems to be an widespread characteristic of the brain and can be found in a number of cortices, such as the auditory [Wagner et al., 2005; Carr and Konishi, 1990] and the visual cortices [Freiwald et al., 2001], for example.

Multichannel neuronal firing times from a male Sprague-Dawley rat were simultaneously recorded during a conditioned behavioral task at the University of Florida McKnight Brain Institute. The rat was chronically implanted with two 2×8 arrays of micro-electrodes placed bilaterally in the forelimb region of the primary motor cortex (1.0mm anterior, 2.5mm lateral of bregma [Donoghue and Wise, 1982]). Neuronal activity was collected with a Tucker-Davis recording rig with sampling frequency of 24414.1Hz and digitized to 16 bits of resolution. The firing times were recorded from individual neurons spike sorted with an online algorithm employing a combination of thresholding and template-based techniques. From sorting, a total of 44 single neurons were recorded, 24 neurons from the left hemisphere and 20 neurons from the right hemisphere. Simultaneously, the rat performed a go no-go lever pressing task in an operant conditioning cage (Med-Associates, St. Albans, VT, USA). The task consisted of choosing and pressing one out of two levers (left or right) depending on a LED visual stimulus to obtain a water reward. The queue and lever press signals were recorded simultaneously with the neural activity with sampling frequency 381.5Hz. See Sanchez et al. [2005] for additional details on the experimental configuration.

ICC was applied to this dataset to investigate for the presence of synchronous neural activity across the ensemble. Figure 5-8 shows some trials with the ensemble ICC. From

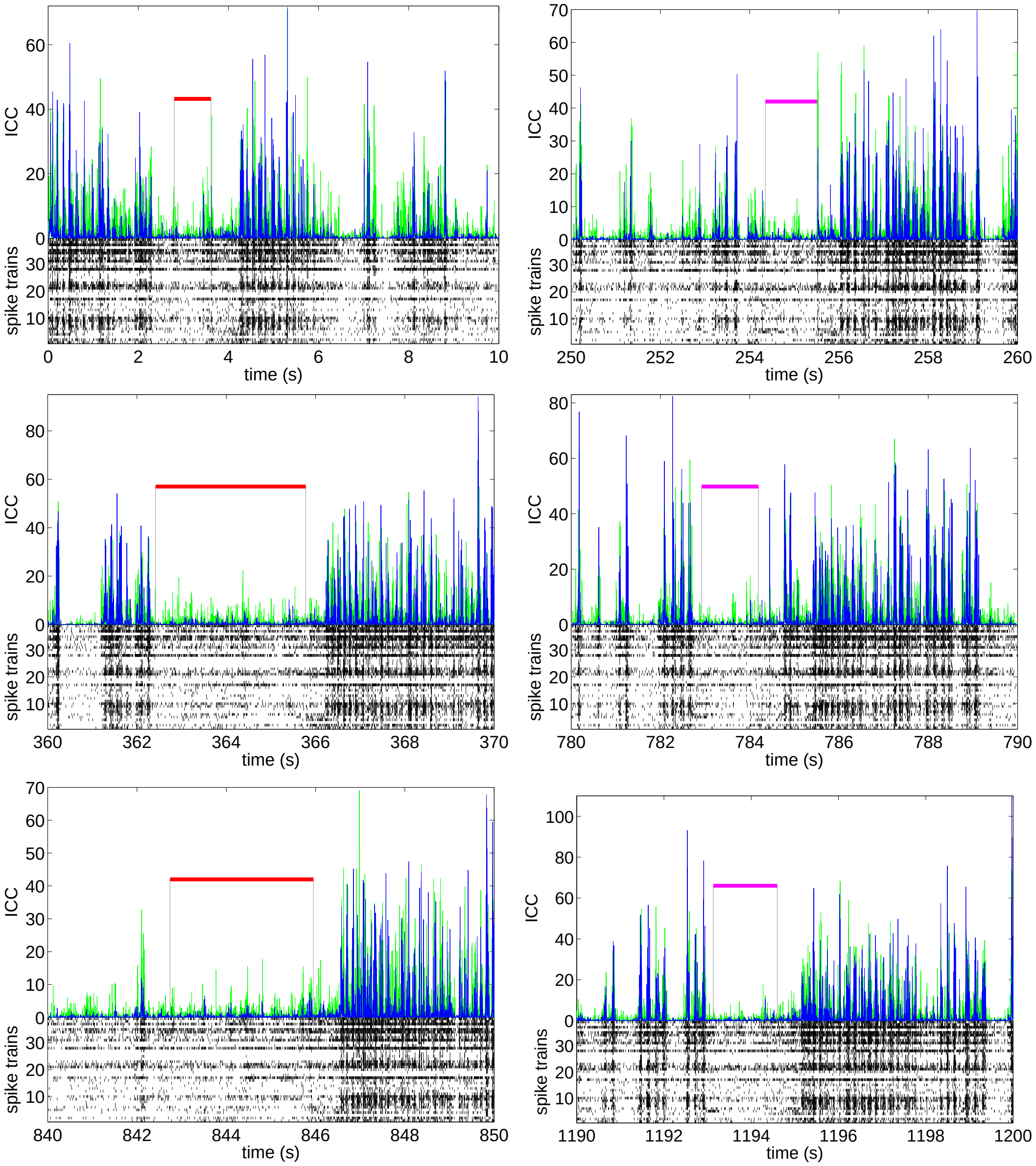


Figure 5-8. ICC and neuron firing raster plot on a single realization, showing the modulation of synchrony around the lever presses. The ICC was averaged throughout the neurons pairs, as given by Equation 5–7, separately for each hemisphere: left (blue) and right (green). The left plots show left lever presses and right plots show right lever presses.

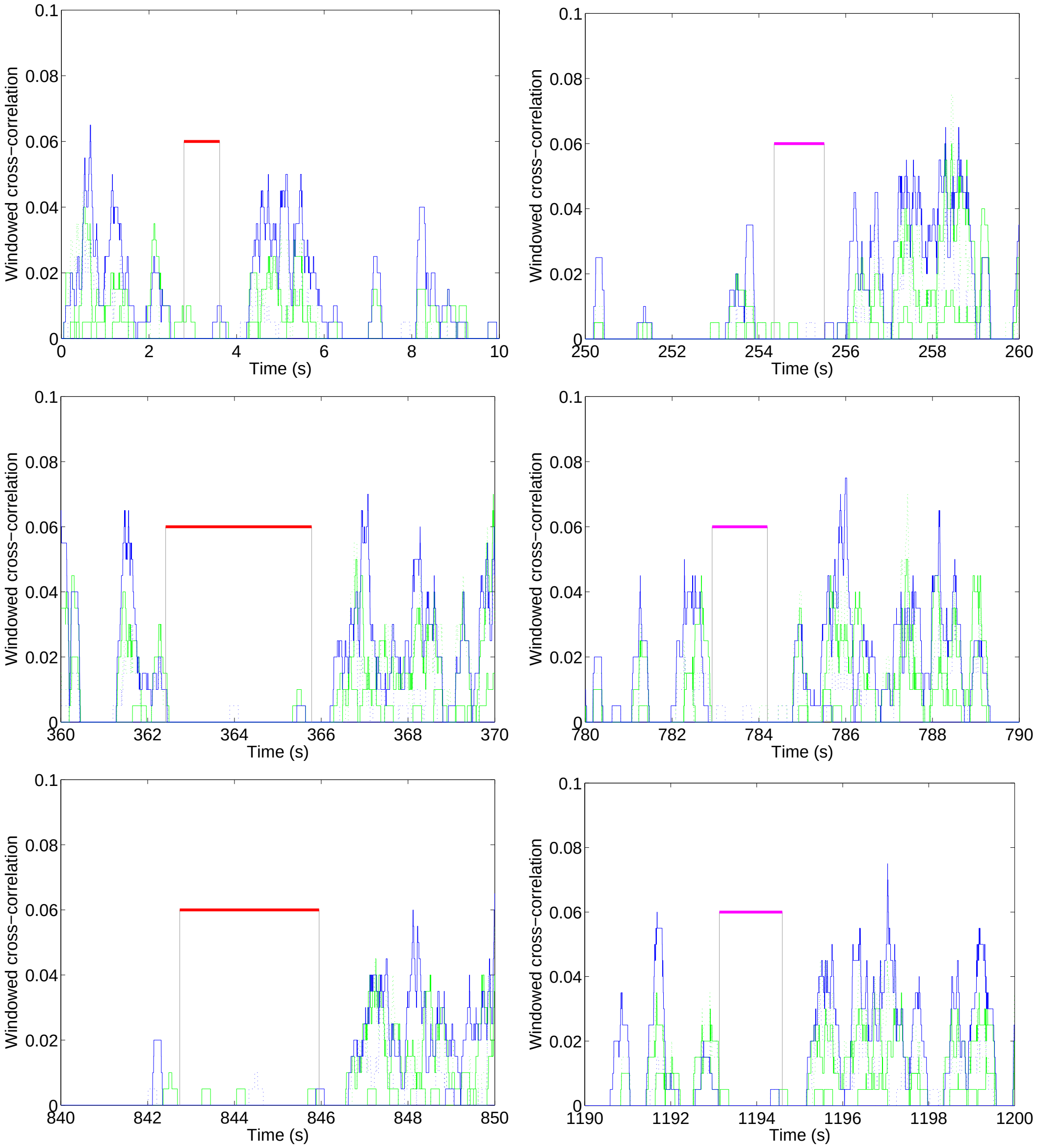


Figure 5-9. Windowed cross-correlation of selected 6 pairs of neurons, for the same segments shown in Figure 5-8. The cross-correlation was computed with a 200ms sliding window over 1ms bins. Four of the neuron pairs, two from each hemisphere, are known to synchronize and are shown in dark gray and light gray solid line for the left and right hemispheres, respectively. The remaining two, one from each hemisphere, do not synchronize strongly and are shown in dotted line. The left plots show left lever presses and right plots show right lever presses.

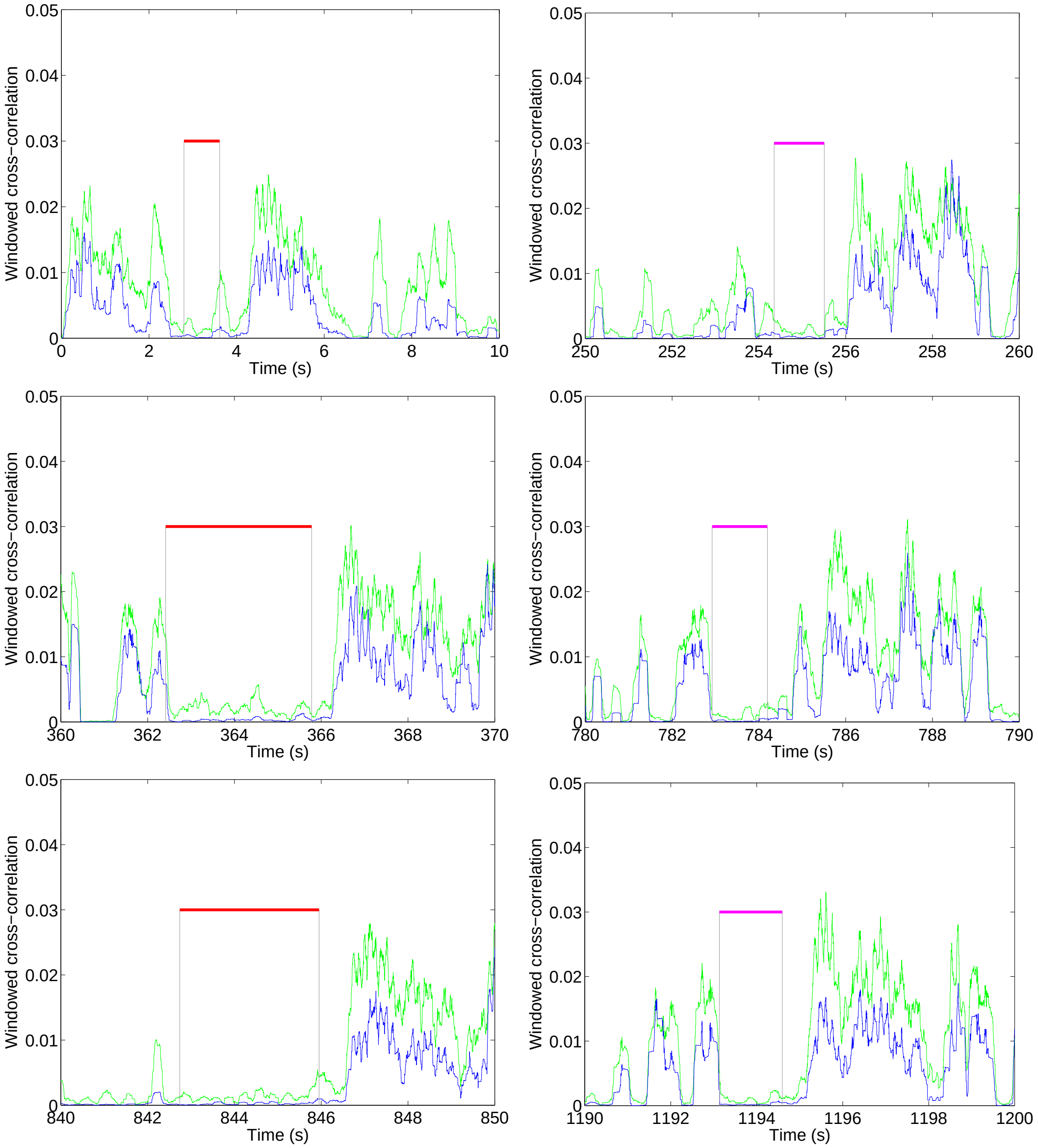


Figure 5-10. *Spatially averaged* windowed cross-correlation, for the same segments shown in Figure 5-8. The cross-correlation for each neuron pair was computed with a 200ms sliding window over 1ms bins. Similar to ICC, the spatial average was done throughout all neuron pair combinations, separately for each hemisphere: left (dark gray) and right (light gray). The left plots show left lever presses and right plots show right lever presses.

the result a systematic increase when the lever is released can be observed. Moreover, while the lever was being pressed the ensemble synchrony was observed to be significantly smaller than synchrony before and after. Actually, in this dataset, visual inspection of the raster plots would yield such conclusions, but ICC provides a quantitative method to translate the visual evaluation. Furthermore, examining the raster plots we can verify the presence of ensemble synchronized activity reoccurring in a periodic manner after the lever is released. Notice that the ensemble ICC captures the presence of this oscillatory synchronized activity, seen in the envelope of ICC in Figure 5-8, directly from a single trial and with high temporal resolution.

For comparison, we show the cross-correlation computed at zero-lag with a 200ms sliding window over 1ms bins [Hatsopoulos et al., 1998]; first, for only some selected pairs of neurons (Figure 5-9), and then spatially averaged (Figure 5-10) as proposed for ICC. Although the presence of synchrony is also successfully captured with cross-correlation, the presence of any periodic modulation in synchrony is not noticeable. This can be expected since the cross-correlation requires stationarity over time and uses time averaging to reduce the randomness of the estimator. These two factors filter out any existing periodicities in the modulation, which may represent a great deal of information. This imposes up front a lower bound on the frequencies that can analyzed. This effect is most visible in Figure 5-10 where spatial averaging greatly improves the estimation as the variance of the estimator is reduced, but the temporal averaging prevents the modulation in synchrony to be clearly noticeable. These figures highlight the importance of the spatial averaging proposed for ICC, in opposition to the time averaging employed in cross-correlation.

The high temporal resolution of the ICC will be wasted in cases where the experimental characteristics do not display high temporal synchrony or the experimental conditions do not allow high precision in temporal measurements. One case is the averaging across trials. Many times, the resolution of the time markers is insufficient with regard to the sampling

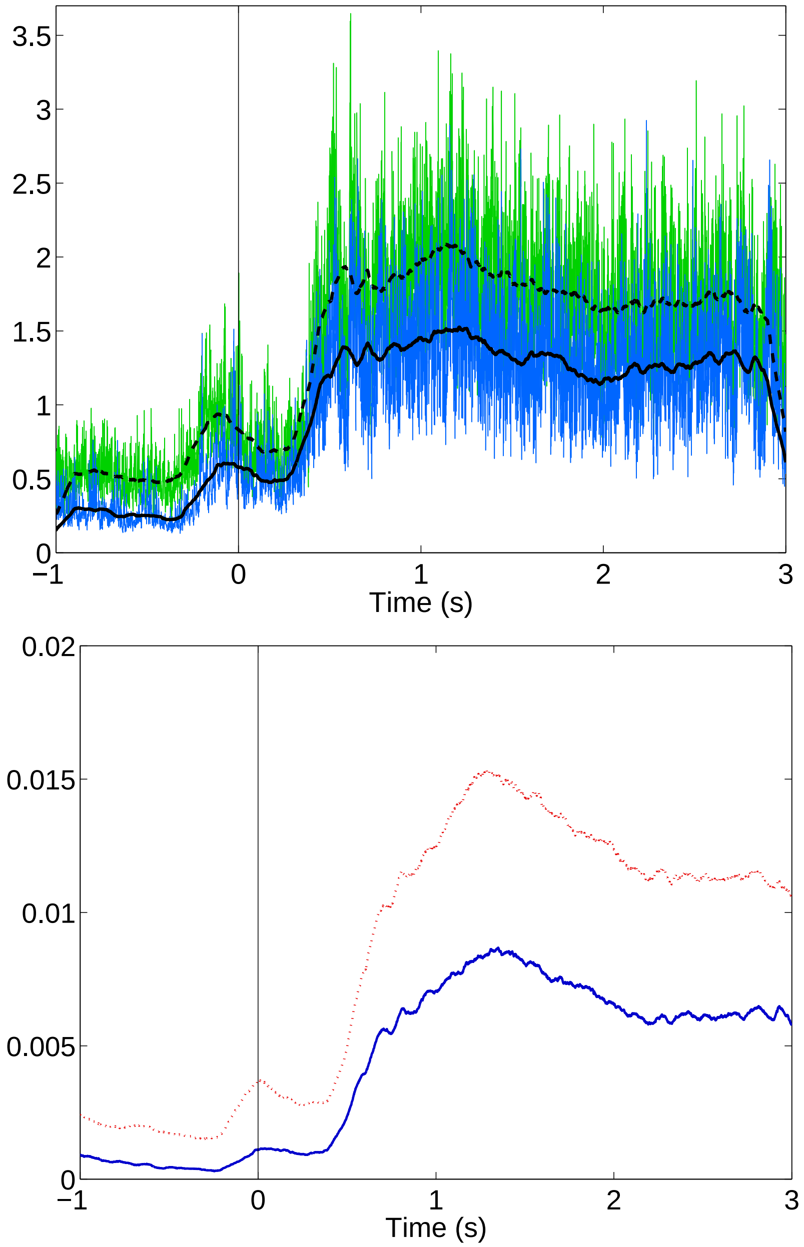


Figure 5-11. Trial averaged ICC (upper plot) and cross-correlation (lower plot) time locked to lever release. The trial averaged ICC is shown for neurons from the left hemisphere (light gray) and right hemisphere (dark gray). Also shown in the plot is the trial averaged ICC smoothed with a 200ms long rectangular window for neurons from the left hemisphere (solid line) and right hemisphere (dashed line). The cross-correlation was computed with a 200ms sliding window over 1ms bins. The triggering event is marked in the figures by time zero.

rate of the neural data collection, or the experimental effects appear asynchronous with the stimulus. However, even in this case the smoothing of the ICC with a lowpass filter will provide results comparable to the cross-correlation function. To illustrate this point, the ICC and its lowpass version (filtered with a rectangular window 200ms long), and the (spatially averaged) cross-correlation were averaged throughout trials synchronized with a lever press. The resulting peri-event plots are shown in Figure 5-11. From the figures, one can conclude that the averaged ICC contains the same information as the cross-correlation where the modulation of synchrony at the lever press is clearly visible as mentioned earlier.

5.4 Peri-Event Cross-Correlation Over Time

The ICC just described is a simple tool to detect and characterize the evolution of correlation with time. Despite the single trial capability of ICC, it is sometimes desired to characterize the interaction among the two neurons as a function of the event onset. Again, averaging over time is not desirable. In this section the *peri-event cross-correlation over time* (PECCOT) is presented. The PECCOT aims to be a tool to analyze and visualize the evolution of synergistic information over time in a convenient way.

5.4.1 Method

The main difficulty in estimating cross-correlation is that in practice only stochastic estimates of the underlying intensity functions are available from spike trains. To obtain statistical reliability, the traditional approach is to average the instantaneous cross-correlation in the argument of expectation over a time interval. The problem with this approach is that it trades time resolution for statistical reliability. A more principled approach is to average over realizations, as expressed in the definition of cross-correlation. There are fundamentally two principled approaches to achieve this:

(i) Average over the neural ensemble; or
(ii) Average over trials.

Each of these approaches implies a particular assumption and provides a specific trade-off. Averaging over the ensemble requires that multiple spike trains are assumed part of the same ensemble, which might have to be found a priori, and one trades “spatial” or ensemble resolution for statistical reliability. Conversely, averaging over trials can only be applied to paradigms where trial repetition is available, and although it quantifies the coupling for each pair of neurons (high “spatial” resolution), it needs to assume stationarity among trials (that is, all trials are realizations of the same underlying process). In spite of that, in both approaches the time resolution is preserved since no integration/averaging over time is involved.

The results presented above where based on averaging over the ensemble. For experimental paradigms with multiple realizations, the second approach is now considered. Instead of averaging over time, the PECCOT averages the instantaneous cross-correlation (ICC) over instances of the event. As a consequence, the PECCOT is able to characterize with high temporal resolution the interactions over time among pairs of neurons. This is conceptually similar to how the peri-event time histogram (PETH) is obtained, but here the quantity expresses neuronal interactions.

Therefore the algorithm for estimation of the PECCOT is as follows:

1. For each realization of the event,
   (a) Estimate the intensity function of each neuron in an time interval around the event onset, $[-T, T]$ (zero corresponding to the event onset), according to Equation 4–3.
   (b) Compute the instantaneous cross-correlation for each pair of neurons. At the kth realization, between neurons i and j, the instantaneous cross-correlation is,
   $$c_{ij}^{(k)}(t) = \hat{\lambda}_{p_i}^{(k)}(t)\hat{\lambda}_{p_j}^{(k)}(t),$$
   where $\hat{\lambda}_{p_i}^{(k)}(t), \hat{\lambda}_{p_i}^{(k)}(t)$ are the estimated intensity functions for the kth realization.
2. Average the instantaneous cross-correlation for each pair of neurons across realizations.

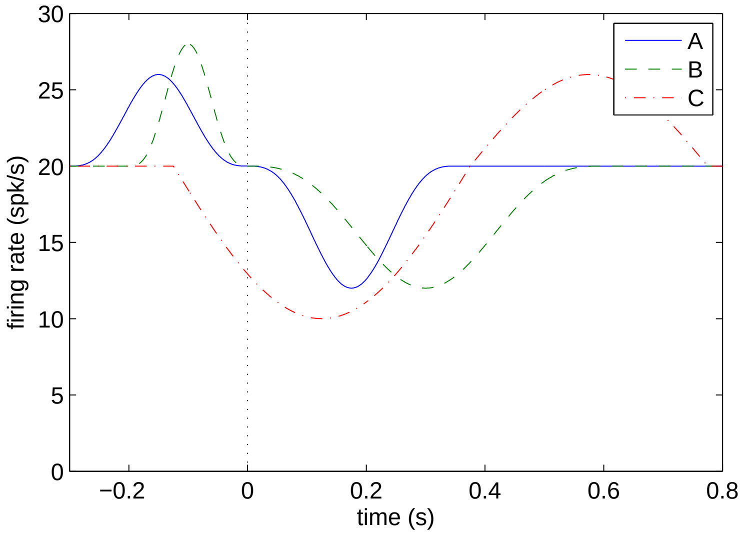


Figure 5-12. Modulation of intensity with the event for each neuron.

Careful examining the algorithm one may recognize the same form that leads to the main diagonal of the JPSTH [Aertsen et al., 1989] which typically expresses the neural interactions. The difference however is that here the computation is done explicitly, and thus much more efficiently. Also, by focusing only on this function, analysis of the overall result is much simpler since the result of all pairs of neurons may be summarized in a single plot. Nevertheless, as for the JPSTH, it is also possible to compute other diagonals by introducing the dependency to a lag between $\hat{\lambda}_{p_i}^{(k)}(t)$ and $\hat{\lambda}_{p_i}^{(k)}(t)$. Moreover, the statistical procedure proposed by Aertsen et al. [1989] for normalization of the JPSTH can be applied for normalization of the PECCOT, with the intensity function estimated by kernel smoothing.

5.4.2 Data Examples

Two data examples of the analysis with PECCOT are now shown. First a simulated dataset is utilized to show the method does capture the desired feature in the data. In the second example the same recording of motor neurons analyzed in Section 5.3.3.3 was analyzed with the PECCOT.

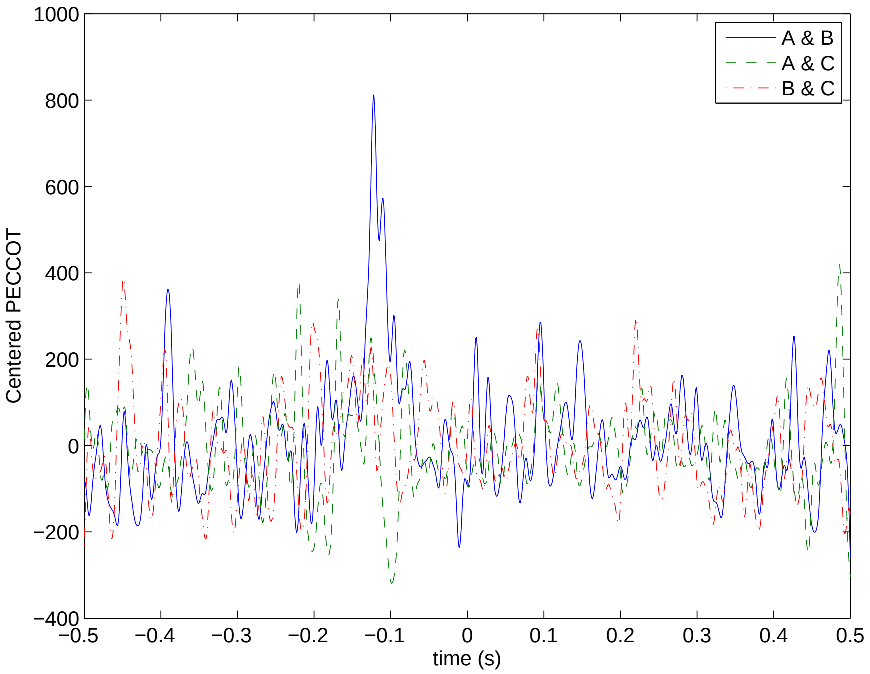


Figure 5-13. Centered PECCOT for the three neuron pairs around the lever.

5.4.2.1 Simulation

To illustrate and validate the method just proposed we consider a simple simulated example. Three neurons with base firing rate 20 spk/s were generated. All of these neurons modulated their firing rate in the time vicinity of the event, as shown in Figure 5-12, and here generated with an inhomogeneous Poisson model. In addition, neurons A and B tended to fire synchronously approximately 0.12s before the event. This coupling was introduced in the generated spike trains by selecting the nearest spike of A to 0.12s before the event as a reference and moving the closest spike in B to the same time (with a 1ms zero mean Gaussian jitter added), if the two spikes differ by less than 50ms (baseline inter-spike interval). Neuron C spiked independently of both A and B. A total of 100 event realizations (trials) where generated.

The constructed dataset was analyzed by PECCOT with a Gaussian smoothing function of width $\sigma = 5$ms. The computed result is shown in Figure 5-13. The result was centered by removing the expected coincidence levels merely due to rate modulations. The

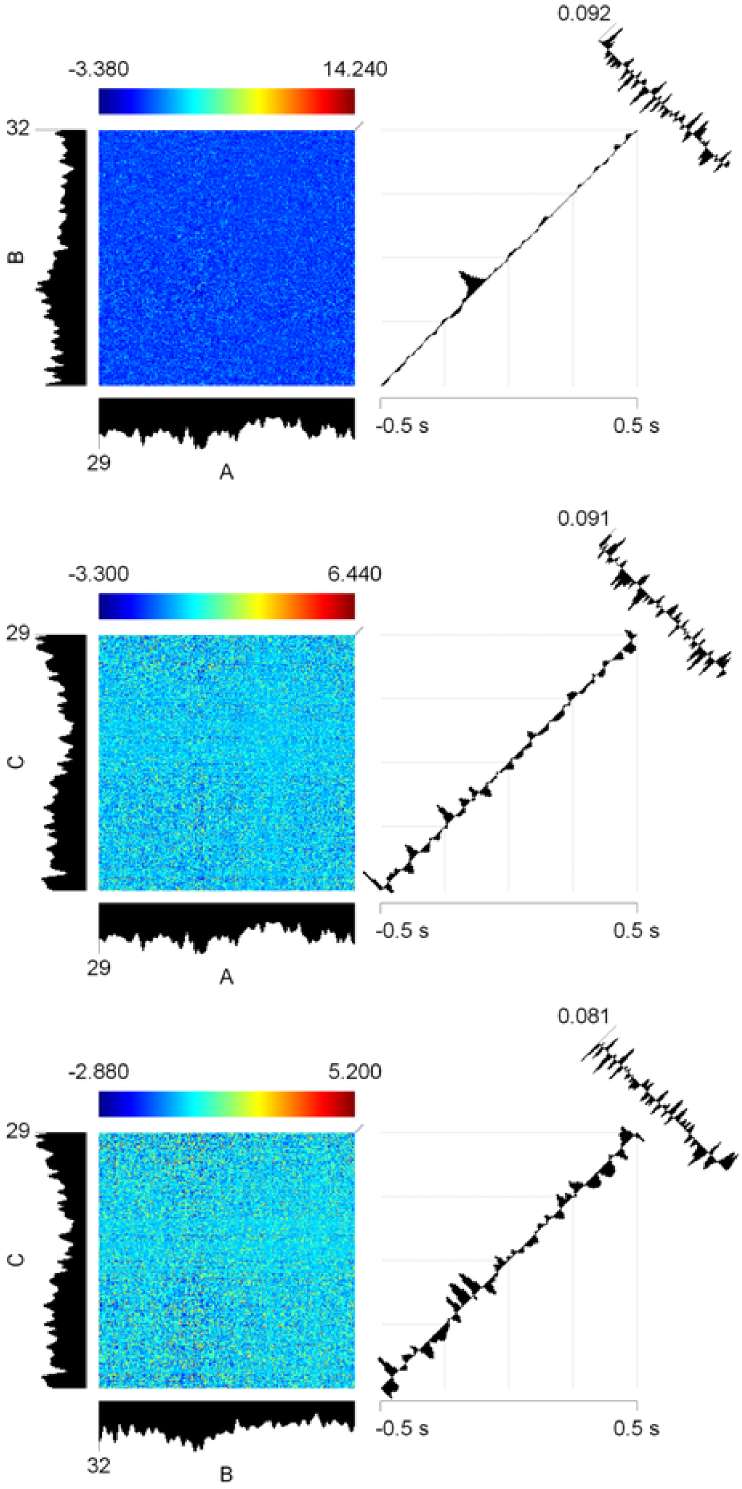


Figure 5-14. Centered JPSTH for each neuron pair.

PECCOT marks the presence of synchronous activity between neurons A and B with a strong peak in the cross-correlation roughly 0.12s before the event onset, as expected given the construction of the dataset. Moreover, the instantaneous cross-correlation between neuron C and others does not show any significant peak, only the effect of firing rate modulations.

For comparison, we also computed the JPSTH for the same neuron pairs (shown in Figure 5-14) using NeuroExplorer (Littleton, MA). For ease of comparison, the bin size was set to 5ms. Again, we observe a strong peak between A and B approximately 0.12s before the event. Several interactions are visible for the other two pairs. However, carefully examining the scales one notices that the peak is about two times higher in the first case. These results highlight the difficulty in analyzing multiple JPSTH plots, especially with an increasing number of neuron pairs. On the other hand, by displaying the result of all neuron pairs in a single plot under the same scale, the PECCOT greatly simplifies this analysis.

5.4.2.2 Event-related modulation of synchronous activity

The PECCOT is now demonstrated for the analysis of couplings in the neuronal firings of neurons in forelimb region of M1 of a rat performing a behavioral task. The same dataset as in Section 5.3.3.3 was utilized. Specifically, we wanted to verify if the neurons' synchronous firing patterns modulated with movement onset.

To test this hypothesis the centered PECCOT[3] was computed in a neighborhood of two seconds before and after the lever presses. The smoothing function for intensity function estimation was a Gaussian function with width $\sigma = 5$ms. For visualization purposes, the centered PECCOT was further smoothed with a Gaussian window of width, $\sigma = 10$ms. To analyze possible differences in synchrony modulation between left and right

[3] Centering was utilized to remove the effect of very different firing rates and their modulations.

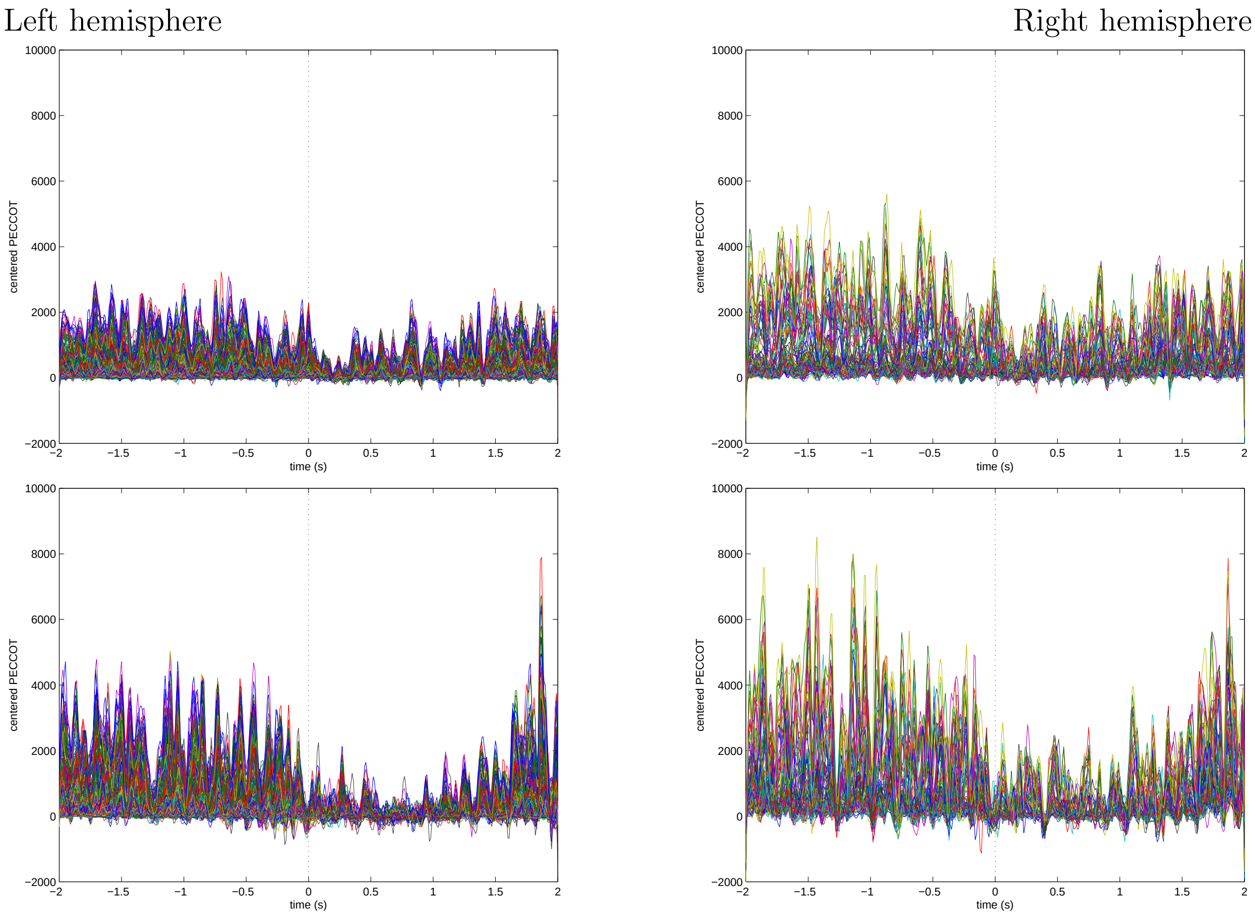
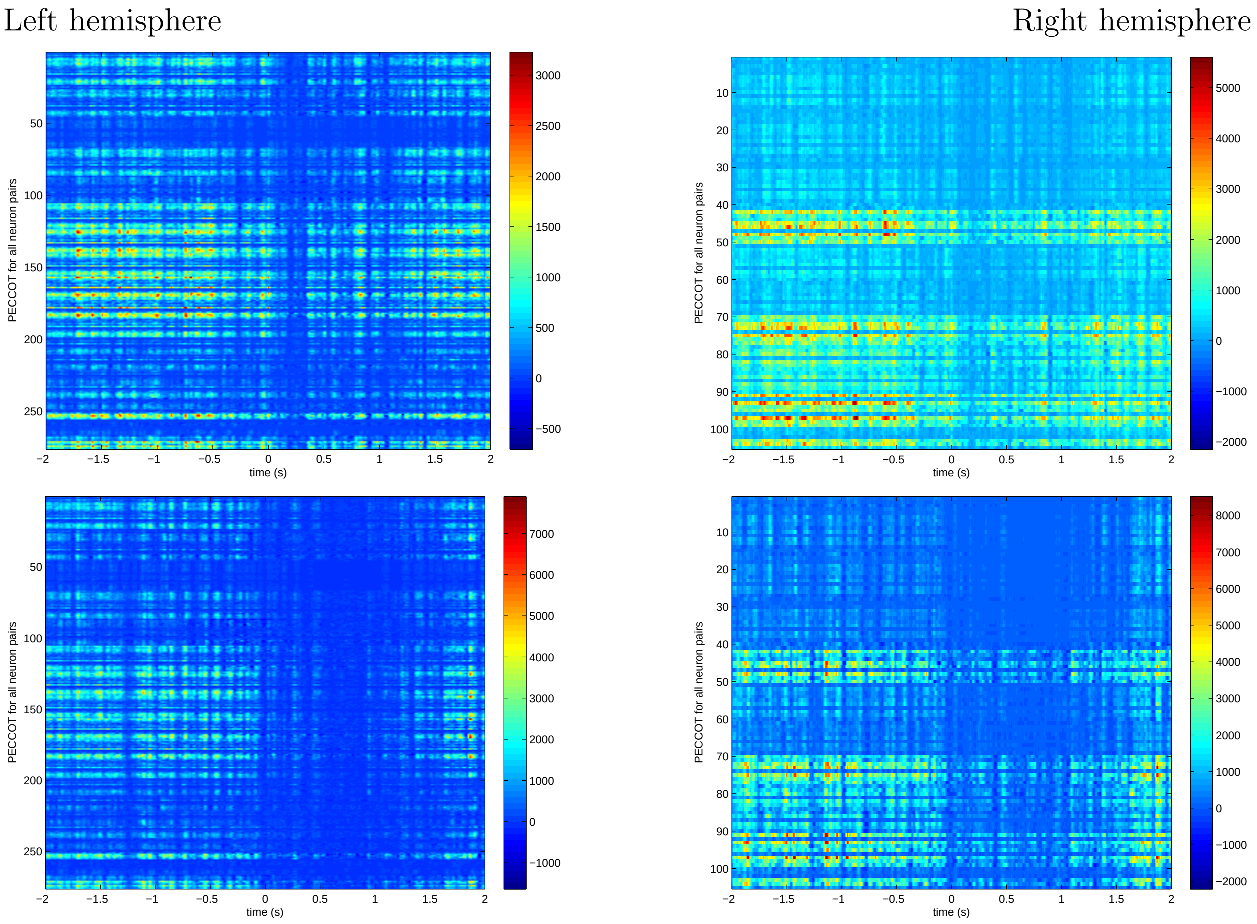


Figure 5-15. Centered PECCOT around the lever press onset. The two columns correspond to neurons from the left and right hemispheres, respectively, and two rows correspond to the situation in which either the left or right lever was pressed, respectively.

lever presses (since the two levers are usually pressed with different paws) and between hemispheres, the situations are considered separately. A total of 93 left lever presses and 45 right lever presses were used for averaging. The results are shown in Figure 5-15 and Figure 5-16. In the first figure PECCOT was shown as in Figure 5-13, while in the second we opted to display the results in the form of a color coded figure due to the large number of neuron pairs, making it easier to visualize the overall modulation and identify the most relevant neuron pairs.

It can be observed that the synchrony among neurons in the left hemisphere is far more widespread than in the right hemisphere, for both left or right level presses. It can


Left hemisphere

Right hemisphere

Figure 5-16. Centered PECCOT around the lever press onset. Like Figure 5-15 but in image form. Each line corresponds to the PECCOT of a pairs of neuron with amplitude color coded.

be clearly observed that in all situations there is considerable interaction among neurons before the lever press instant and that these interactions are almost entirely suppressed immediately after. Approximately one second after the lever press instant the synchrony increases again. Interestingly, it should be remarked that this time interval corresponds approximately to the average duration of a lever press, after which the rat receives a water reward if the correct lever was pressed. Moreover, we notice lever press specific synchrony modulation with depressions around 1.4s, 0.95s, 0.8s, 0.45s and 0.3s before a left lever press, and a major depression around 1.25s before a right lever press. These modulations are present at the same time in both hemispheres. Also, in the images it is apparent that the interactions between neurons tend to be phase locked and have a periodic component in the theta range (3–8Hz). Although we have not investigated the reason for this periodic phase locking of synchrony, these results may provide further evidence on the role of low frequency rhythms commonly found in meso- and macroscopic recordings as “clock signals” for synchronization of multiple brain regions.

CHAPTER 6
CLUSTERING OF SPIKE TRAINS

Having an RKHS framework for point processes is important because it facilitates the development of new methods to operate with point processes, and their realizations. Moreover, all of these methods are developed under the same principles provided by this general theory.

To exemplify the use of point process kernels proposed under the RKHS framework, in the following we show how a clustering algorithm for spike trains can be obtained naturally from any of the point process kernel definitions here presented. Comparing these ideas with previous clustering algorithms for spike trains we find that they result in simpler methods, derived in an integrated manner, with a clear understanding of the features being accounted for, and greater generality. It must be remarked that although this chapter shall consider spike trains, it is immaterial the exact nature of the realizations of point processes to be clustered.

Note that the primary emphasis here is to illustrate the elegance and usefulness of the RKHS framework rather than merely propose another algorithm. In spite of that, it will be shown through multiple simulations that the spike train algorithm presented here performs as good or better than other algorithms in the literature despite its simplicity.

6.1 Algorithm

In the literature a few algorithms have been proposed for clustering of spike trains. Examples are the methods proposed by Paiva et al. [2007] and Fellous et al. [2004]. Both of these algorithms rely on measures between spike trains. Paiva et al. [2007] utilized van Rossum's distance [van Rossum, 2001], but it is pointed out that Victor-Purpura's (VP) distance [Victor and Purpura, 1996, 1997] could be used as well. In turn, Fellous et al. [2004] used instead the "correlation-based measure" proposed by Schreiber et al. [2003]. Nevertheless, as shown in Section 3.6.3, either of the measures used in the previous clustering algorithms can be reformulated in terms of the mCI kernel. More than simply a reformulation of the distances, this raises the question: “Can the RKHS framework be utilized to derive clustering algorithms is an integrated manner?” The answer is yes. For the purpose of this example we will show how spike train kernels defined in the RKHS framework provide the means to do clustering of spike trains. The algorithm will be based on the ideas of spectral clustering, since kernels naturally quantify affinity. Spectral clustering is advantageous for the purpose of this example since the evaluation of the affinity between spike trains by point process kernels and the actual clustering procedure are conceptually distinct. It is possible to extend other clustering algorithms although one must introduce the inner product directly into the computation which slightly complicates matters.

Spectral clustering of spike trains operates in two major steps. First, the *affinity matrix* of the spike trains is computed. Let $\{s_1, s_2, \ldots, s_n\}$ denote the set of n spike trains to be clustered into k clusters. The affinity matrix is an $n \times n$ matrix describing the *similarity* between all pairs of spike trains. The second step of the algorithm is to apply spectral clustering to this affinity matrix to find the actual clustering results. In particular, the spectral clustering algorithm proposed by Ng et al. [2001] was used for its simplicity and minimal use of parameters. The clustering algorithm, presented step-by-step, is presented in Table 6-1. The reader is referred to Ng et al. [2001] for additional details on the spectral clustering algorithm.

Clearly, the defining step for the use of this algorithm is how to evaluate affinity between spike trains. Since inner products inherently quantify similarity, any of the kernels proposed can be used, and in particular the mCI and nCI kernels, for which we provide results. Geometrically, this role of the kernel can be understood since the inner product is sensitive to the norm and angular distance of the two spike trains in the RKHS. In this situation the affinity matrix is simply the Gram matrix of the spike trains computed with the spike train kernel. Note that the cross-correlation (CC) of binned spike trains is in itself an inner product of spike trains and therefore could be used as well. Indeed,

Table 6-1. Step-by-step description of the algorithm for clustering of spike trains. These are basically the steps of the spectral clustering algorithm.

1. Compute the affinity matrix $A \in \mathbb{R}^{n\times n}$ from the n spike trains. The ijth entry of the affinity matrix is given by,

$$a_{ij} = \begin{cases} \hat{\mathcal{I}}(s_i, s_j), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{6–1}$$

   where $\hat{\mathcal{I}}(p_i, p_j)$ denotes the estimator of any point process kernel, evaluated for spike trains s_i and s_j.

2. Construct D as a diagonal matrix with the ith element of the main diagonal equal to the sum of all elements in the ith row of A (or column, since A is symmetric). That is,

$$d_i = \sum_{j=1}^{n} a_{ij}.$$

3. Evaluate the matrix

$$L = (D^{-\frac{1}{2}})A(D^{-\frac{1}{2}}).$$

4. Find $x_1, x_2, \ldots, x_k$, the k eigenvectors of L corresponding to the largest eigenvalues, and form the matrix $X = [x_1, x_2, \ldots, x_k] \in \mathbb{R}^{n\times k}$.

5. Define $Y \in \mathbb{R}^{n\times k}$ as the matrix obtained from X after normalizing each row to unit norm. Consequently,

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}.$$

6. Interpreting Y as a set of n points in $\mathbb{R}^k$, cluster these points into k clusters with k-means or similar algorithm.

7. Assign to the ith spike train the same label of the ith point (row) of Y.

Eggermont [2006] utilized this idea in his analysis. However, binning quantizes the spike times and is therefore introduces boundary artifacts in the analysis, as we will show later.

Compared to the method proposed by Paiva et al. [2007] the algorithm shown here is simpler since no transformation to map the distance evaluation to a similarity measurement and the need to adjust the corresponding parameter is avoided. Since distances are derived concepts and, usually, can be defined in terms of inner products, the approach taken is much more straightforward and principled. Moreover, the algorithm can be generalized merely by using a different point process kernel. Even the simple mCI kernel explicitly unveils a broader potential of the algorithm. In particular, unlike the formulation of Paiva et al. [2007] which was apparently restricted to clustering of spike trains by synchrony, our knowledge based on the mCI kernel reveals this is not true. Rather it is merely a matter of kernel size. Furthermore, there is a close connection between point process kernels and kernels on spike times (i.e., event coordinates), either by construction or in estimation (as Section 3.4.2 elicits), and thus suggests that a multitude of kernels on spike times can be used in place of the Laplacian kernel associated with van Rossum's distance (cf. Section 3.6.1). These ideas shall be illustrated next with some simulation experiments.

6.2 Comparison to Fellous' Clustering Algorithm

The clustering algorithm of spike trains by Fellous et al. [2004] is perhaps the most well established method in the literature. Therefore, this algorithm will now be compared with the clustering algorithm we just described. This allows to assess which algorithm is better, and if clustering ability might have lost in using the RKHS framework.

The algorithm by Fellous et al. [2004] is somewhat similar in principle to the above algorithm, but with important differences. For reference, the clustering algorithm is now given. The algorithm operates in three steps:

1. Compute the similarity matrix using Schreiber's et al. correlation-based measure [Schreiber et al., 2003].

2. Reshape the similarity matrix with a sigmoid function to increase the entropy of the histogram of similarity values.

3. Apply fuzzy C-means (FCM) to the similarity matrix by taking each column (or row) as an input point.

The first step corresponds to the computation of the affinity matrix that we had described earlier. The second step was motivated by the work of Bell and Sejnowski [1995] and, according to the authors, aimed to improve the clustering performance. The last step uses FCM (or fuzzy K-means; they are the same), to obtain the actual clustering. Basically, this uses the idea that neighboring spike trains are reciprocally close and therefore the similarity between two spike trains is small at the same row (or column) of the column of the similarity matrix.

For the comparison, the same surrogate dataset utilized in Fellous et al. [2004] was used. The dataset is available at `http://www.cnl.salk.edu/~fellous/data/JN2004data/data.html`. The dataset includes three scenarios with 2, 3 and 5 clusters. There are a total of 100 situations for each scenario corresponding to multiple levels of extra spikes (non-synchronous spikes aimed to confuse the clustering) and multiple levels of jitter in the synchronous spikes. In each situation, the dataset comprises 30 Monte Carlo runs, each with 35 spike trains to be clustered.

Both clustering algorithms were implemented in Matlab. The results for the algorithm proposed were computed with the mCI kernel estimator using the Gaussian kernel with width 5ms, as indicated in Fellous et al. [2004, pg. 2992]. For reshaping of the similarity matrix the procedure in Fellous et al. [2004, pg. 2999] was followed.

From Figure 6-1, Figure 6-2, and Figure 6-3 one can clearly verify that the method proposed here and using the mCI kernel estimator achieves much better performance. Even though the results are somewhat comparable for the two cluster problem, with a difference smaller than 6%, for a higher number of clusters this improvement is as high

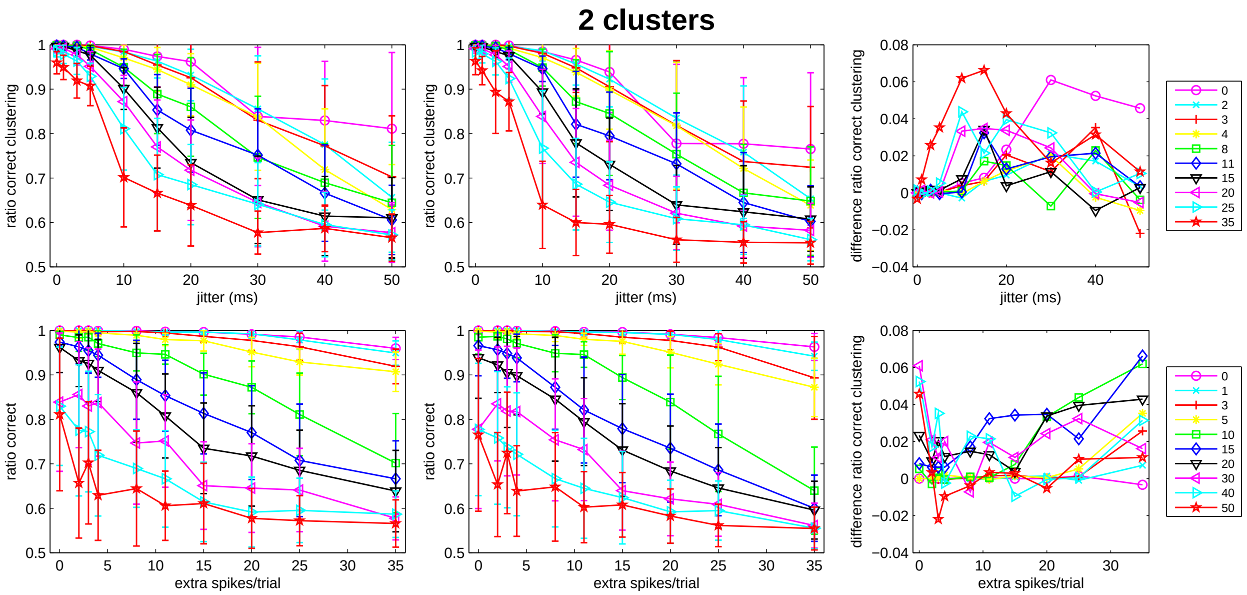


Figure 6-1. Comparison of clustering performance between the clustering algorithm proposed here and Fellous' algorithm for two clusters. In the left column the results for the clustering using the mCI kernel with the Gaussian kernel are shown. In the middle column the results are for Fellous' algorithm. The right column shows the difference between the two methods (first minus second). The upper row shows the results as a function of the jitter standard deviation for multiple number of extra spikes (legend on the right), and the bottom row shows the same results from the reciprocal perspective.

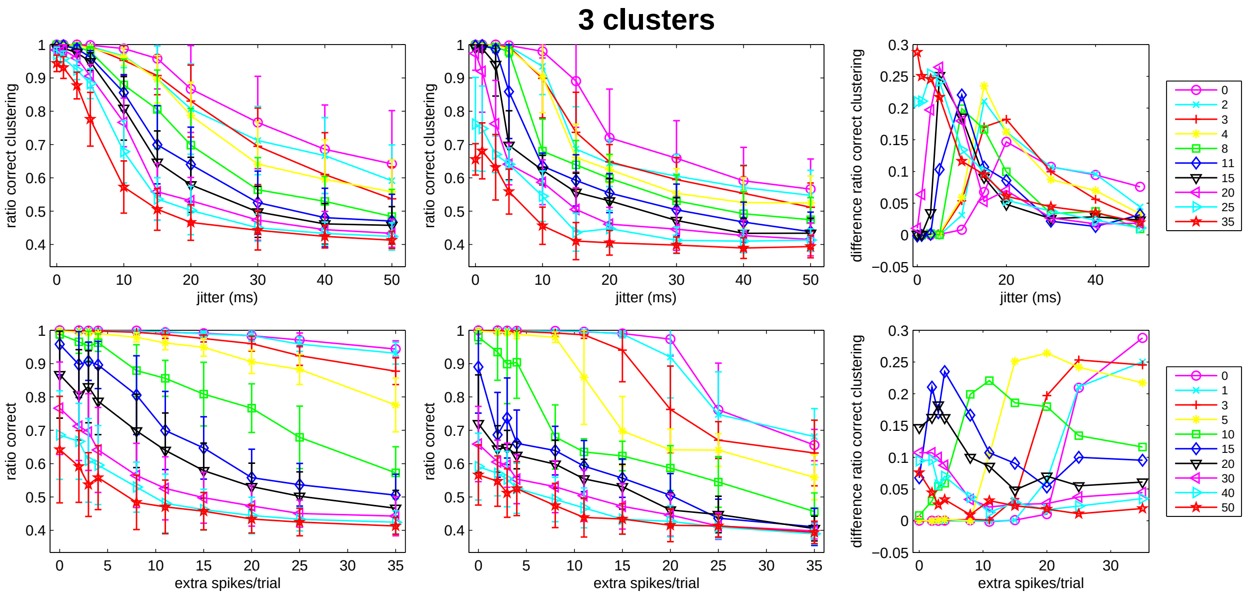


Figure 6-2. Comparison of clustering performance between the clustering algorithm proposed here and Fellous' algorithm, like Figure 6-1, but for three clusters.

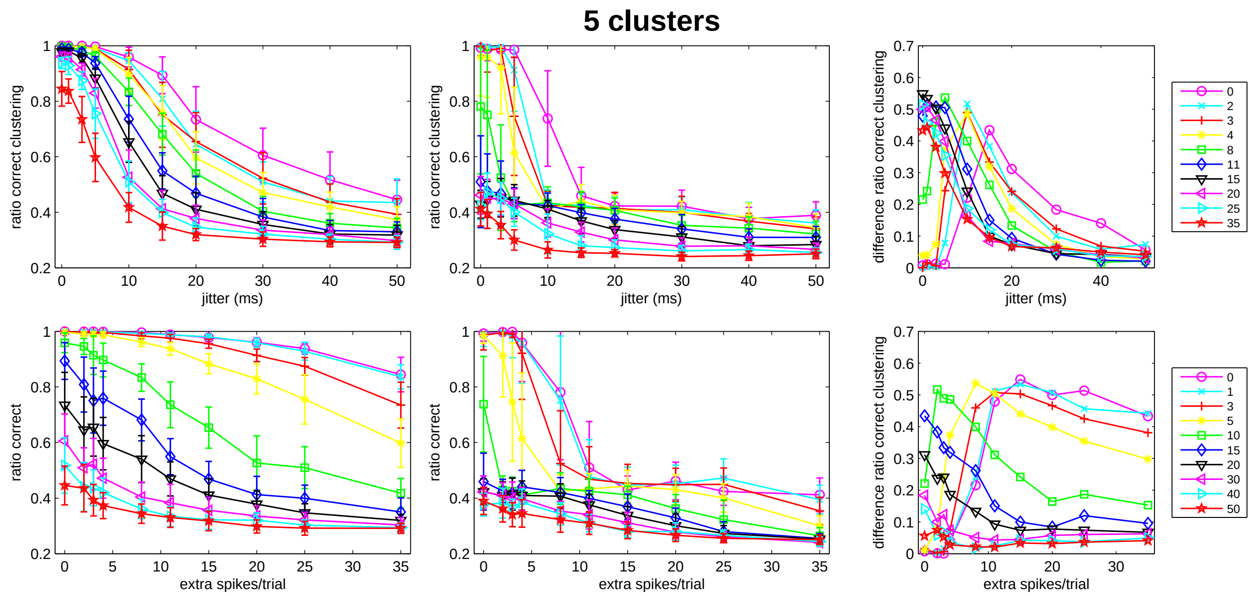


Figure 6-3. Comparison of clustering performance between the clustering algorithm proposed here and Fellous' algorithm, like Figure 6-1, but for five clusters.

as 25% for three clusters and 50% for five clusters! The primary reason for this should the direct use of the whole similarity/affinity matrix in the second method. Note that the implementation embeds the n-dimensional similarity vectors in an n-dimensional space. Consequently, this space is necessarily sparse. Even though it performs acceptably for a small number of clusters, as the number of clusters is increased the sparsity within cluster for the dimensionally of the space greatly hinders the clustering performance. Of course using a larger kernel would mitigate the problem somewhat by introducing correlations among dimensions but would limit the analysis for the problem initially intended by Fellous and colleagues.

The role or relevance of the similarity matrix reshaping always intrigued us. Thus, this was investigated in our simulation, although these results are not shown for conciseness. It was found that this transformation had a minimal impact, and actually the results tended to be slightly better without it.

6.3 Simulations

This section aims primarily to compare the use of multiple point process kernels. First the clustering using the mCI kernel, the nonlinear kernel definition in Equation 3–12, and the (binned) cross-correlation (CC). In Section 6.2, the mCI and nCI kernels are compared for the clustering of renewal point processes.

6.3.1 Clusters Characterized by Firing Rate Modulation

In this simulation example the defining feature of each cluster is similarity in the intensity function underlying each spike train. Specifically, this means that for each cluster an intensity function was generated, in this particular case chosen to be a sinusoidal with 1Hz frequency. These intensity functions were then utilized to generate one second long inhomogeneous Poisson spike trains. Since the spike trains for each cluster were generated according to the same intensity function, ideally, the evaluation of the point process kernels would yield the maximum value for spike trains within cluster and a different value for the remaining spike trains. However, since the data is limited there is some variance in the evaluation of the kernel which leads to clustering errors. Of course, if the spike trains are made longer this variability is decreased and therefore the clustering performance is improved. The clustering performance also depends on how different the two clusters are. In our case the differentiating characteristic between clusters is the phase difference between the two sinusoidal intensity functions.

In the simulation, the clustering performance was measured as the value of the relative phase was varied over the interval $[0°, 180°]$ in steps of 20 degrees. For each value, the clustering performance results were averaged over 100 Monte Carlo runs, each comprising 100 spike trains randomly distributed over the two clusters. Performance results are also given using three different kernel sizes 25ms, 50ms and 100ms, in the estimation of the point process kernels and van Rossum's distance. The kernel sizes were purposely chosen large (that is, on the order of the average inter-spike interval) since by the problem formulation it is known that the distinguishing feature is a smooth intensity

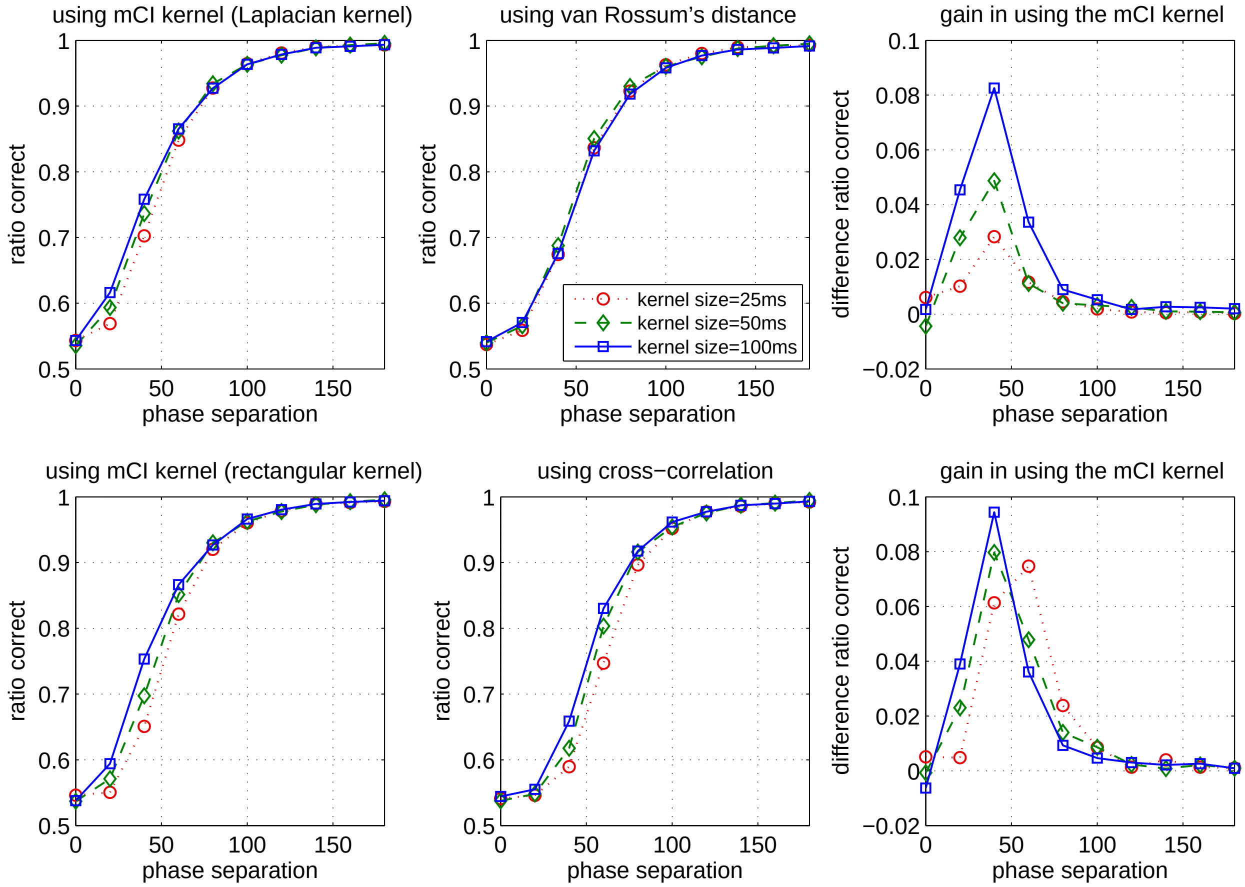


Figure 6-4. Clustering performance as a function of the phase difference in the intensity function. In the left column the results for the clustering using the mCI kernel with the Laplacian and rectangular kernels are shown. Likewise, in the middle column it is shown the results for the algorithm using van Rossum's distance ($\sigma = 10$) and the cross-correlation (top and bottom rows, respectively). The right column shows the difference between the clustering performance using the mCI kernel and the corresponding method on the middle column (of the same row). In each plot, clustering performance results are shown for three kernel sizes specified in the legend (for the cross-correlation interpret "kernel size" as "bin size").

function. Results for point process kernels using a rectangular kernel, with width given by the kernel size, are also shown for comparison with a CC-based inner product. The goal is to illustrate the limitations incurred in a discrete time representation as imposed by binning. Notice that the rectangular "kernel" is not positive definite. Nevertheless, it can be utilized in estimation just like the tanh function is utilized in kernel methods [Schölkopf et al., 1999].

Figure 6-4 shows the clustering performance results using the mCI kernel evaluated with both the Laplacian and rectangular kernels. These results are contrasted with the approach in Paiva et al. [2007], for the optimum size of the Gaussian function ($\sigma = 10$), and utilizing CC as the inner product. Note that the similarity measure utilized in Paiva et al. [2007], corresponds in effect to the nonlinear point process kernel definition in Equation 3–12. The Laplacian and rectangular kernels used to evaluate the point process kernels were selected to approximate the kernel function on spike times implicit in the measure we were comparing against. As shown in the figure, the implementation utilizing the mCI kernel not only is simpler but also outperforms the competing algorithms by up to nearly 10%. This improvement is most noticeable for small phase differences; that is, when discrimination among clusters is the most difficult. Most importantly, the generality of point process kernels allows to experiment with many different kernels on spike times. In this paradigm, between the Laplacian and rectangular kernels, the best results are achieved with the Laplacian kernel. Anyway, it is shown that even utilizing the rectangular kernel the performance can be considerably improved with regards to the use of the CC, only because no binning is utilized.

6.3.2 Clusters Characterized by Synchronous Firings

In contrast to the previous scenario, we now consider the case when clusters are characterized through synchronized spikes among their spike trains. In other words, a dependency is imposed in the underlying process generating spike trains within a cluster such that a spike is added simultaneously into more than one spike train with some

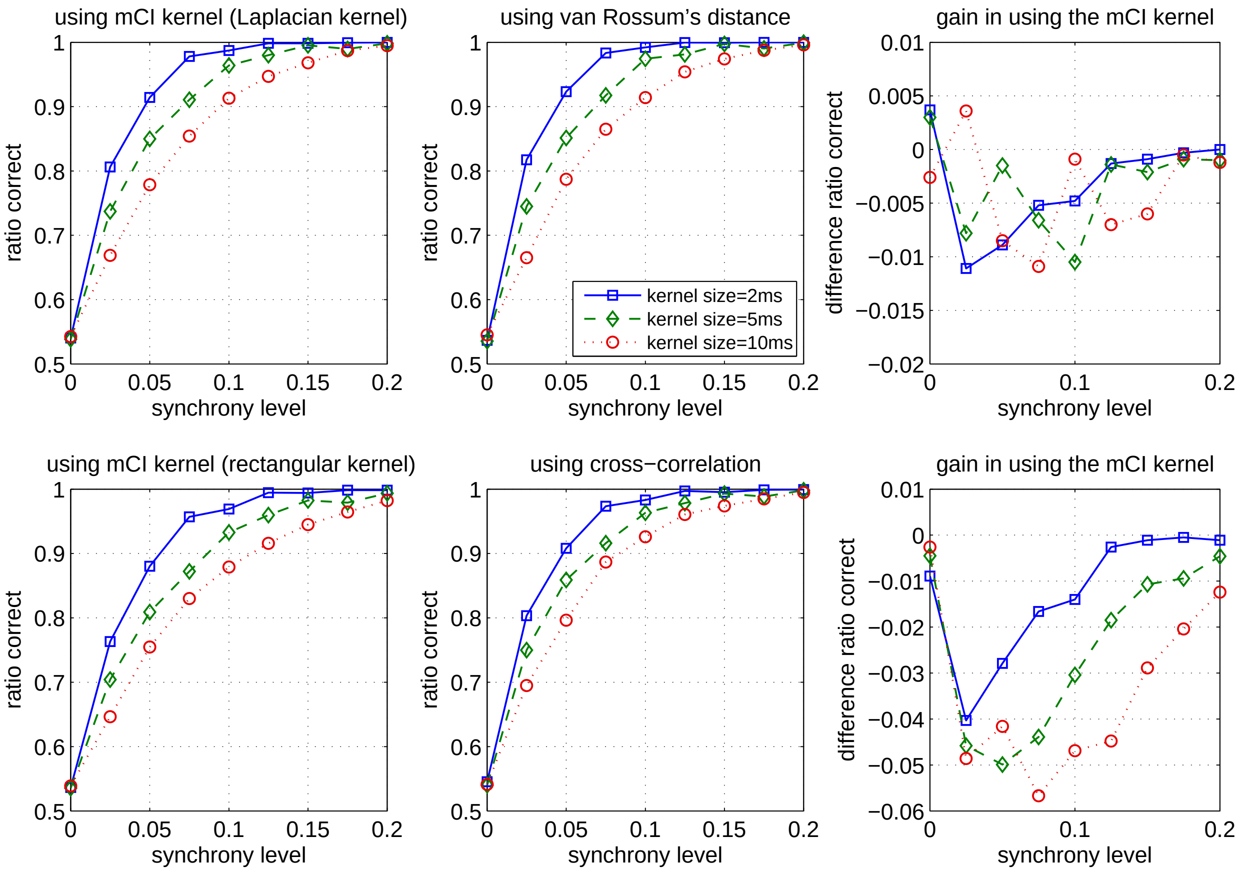


Figure 6-5. Clustering performance as a function of the synchrony level between spike trains within cluster in the jitter-free case. The results are shown in the same form as for Figure 6-4, with results using the mCI kernel in the left column, van Rossum's distance and CC in middle column and difference in performance in the right.

probability. Since each cluster is generated independently so are the resulting spike trains between clusters.

Like the previous simulation, the idea here is to parameterize the synchrony of spike trains within clusters and verify the clustering performance based on this parameter. In our case, this is regulated quite simply by the probability that the generating process introduces a spike into more than one spike train at the same time. In the following we shall refer to this probability as the “synchrony level,” defined as the (expected) ratio of synchronous spikes with regards to the overall spike rate. In our case we modeled this situation through cluster wide synchronous spikes with an average occurrence $\varepsilon\gamma$ spk/s,

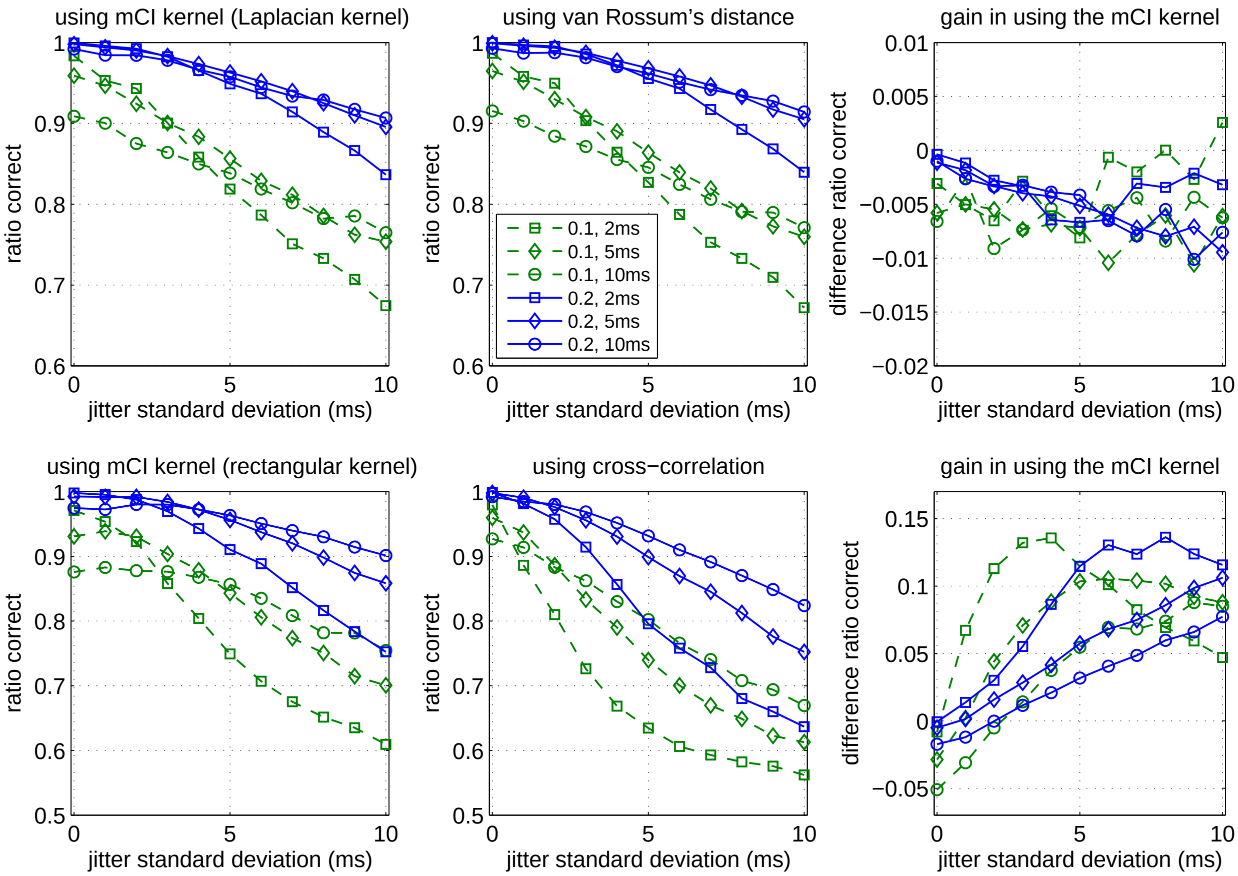


Figure 6-6. Clustering performance as a function of the jitter standard deviation. Again, the results are shown in the same structure as Figure 6-4. However, for each plot in this case, clustering performance results are provided for two values of the synchrony level and, for each synchrony level, for three kernel sizes as indicated in the legend.

where ε and γ are the synchrony level and final average spike rate for the spike trains, respectively. By 'cluster wide synchronous spikes' we mean that synchronous spikes are introduced in all spike trains within a cluster at the same time. This process is then added to an independently generated homogeneous Poisson spike train with average spike rate $(1-\varepsilon)\gamma$. Notice that the resulting spike trains are still Poisson distributed and with average spike rate γ. However, the reader might think that the underlying intensity function for each clusters has most of the time a constant value of $(1-\varepsilon)\gamma$, except at the times of the synchronous wide spikes where scaled impulses are present integrating to $\varepsilon\gamma$. It is worth pointing out that clustering of spike trains corresponding to this paradigm is a common problem. In fact, this was the main motivation and application for the work by Fellous et al. [2004].

Two situations were considered for analysis. In the first, synchronous spikes match perfectly so that the kernel (or bin) size can be made as close to zero as desired. Indeed the best performance is expected as the kernel size is made smaller since there is better discrimination of true synchronous spikes than from spikes that occur by chance. Conversely, as the kernel size is increased more spikes occurring by chance are accounted for, thus increasing the "noise" and variability of the measurement. In the second case, the synchronous spikes were jittered independently with zero-mean Gaussian noise before they were introduced into each spike train. Unlike the first situation which depicts an improvable scenario, this situation aims at understanding how the algorithm perform under some variability in the synchronous spikes as is often encountered in practice.

For the simulation, 100 spike trains were generated at a time according to the process described before and distributed randomly over two clusters. As a result of the process above, spike trains had constant average spike rate of 20 spk/s and were one second long. Results were obtained by averaging over 100 Monte Carlo runs in the first situation (no jitter) and 500 Monte Carlo runs in the second situation (with jitter). This procedure was repeated for each synchrony level and for three different kernel sizes, 2ms, 5ms and 10ms. Compared to the experiment in Section 6.3.1, in this case the kernel sizes were chosen small compared to the average inter-spike interval ($\sim$ 50ms) since in the paradigm formulation it was stated that clusters were characterized by synchrony. Alternatively, this can the thought of in terms of the problem in estimating the intensity function we depicted earlier, which happens to be implicitly taken into consideration by the mCI kernel.

The clustering performance results are given in Figure 6-5 and Figure 6-6, for the jitter-free and with jitter situations, respectively. In both figures it can be observed once again that the clustering results using the point process kernels evaluated with the

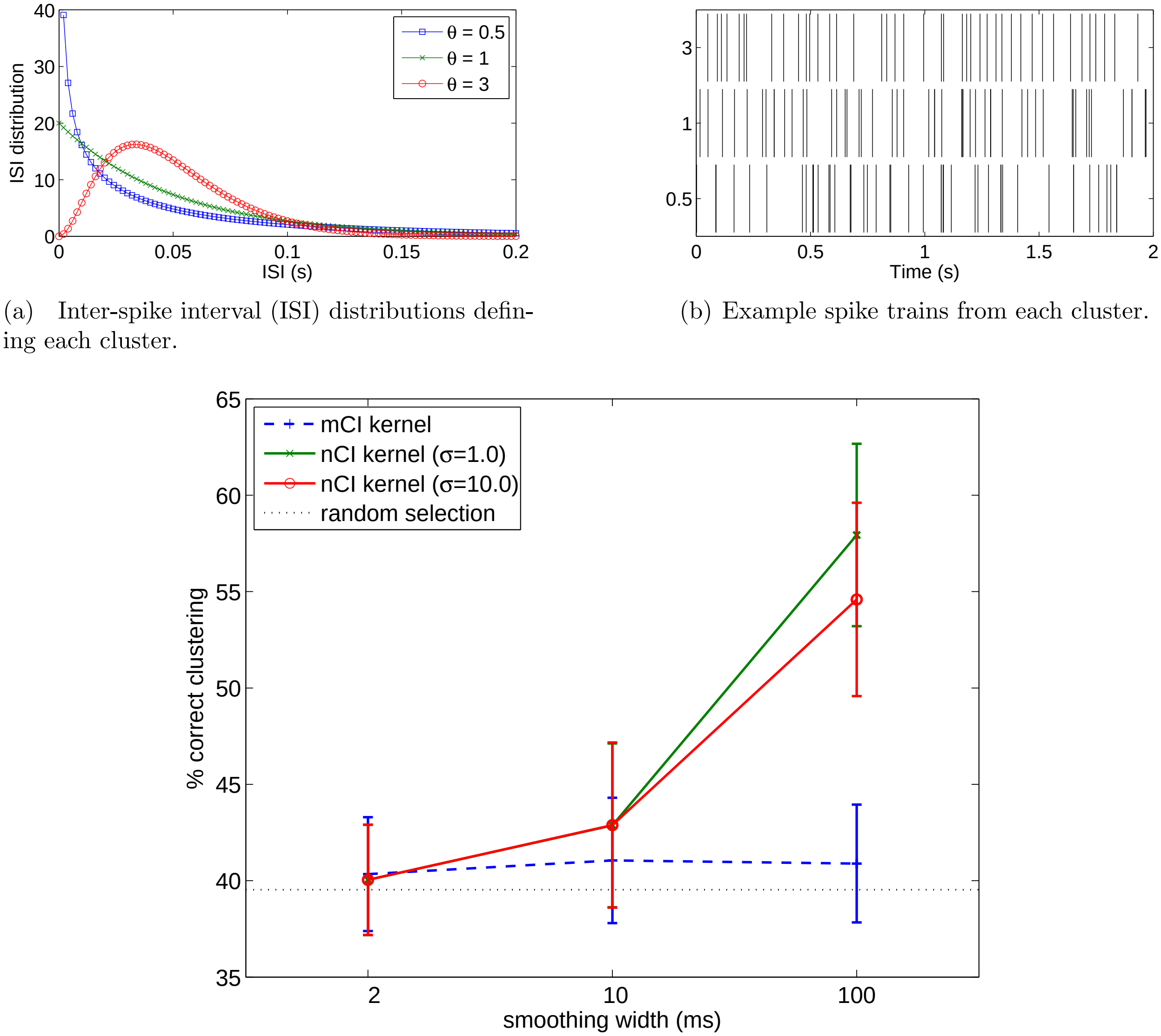
Laplacian kernel are better than with the rectangular kernel. However, the algorithm using the mCI kernel performs similarly to the algorithm based on van Rossum's distance, in either situation. In the case of the comparison with the CC-based algorithm the latter performs better in the noise-free case. However, Figure 6-6 shows that this is only true for small (< 2ms) standard deviations of the jitter noise smaller. As the jitter is incorporated, even small variability in the synchrony in the spikes leads to significant losses in the performance using CC. This shows the particularly significant negative impact of using binned spike trains for synchrony-based clustering under realistic scenarios.

Although, as remarked above, the method by Paiva et al. [2007] utilized one of the nonlinear point process kernels proposed, the results are not better than with the mCI kernel. It must be emphasized that such behavior was expected for the reasons presented in Section 3.2.2. Basically, because the nonlinearity plays a minimal role in extending the capabilities of the mCI kernel this definition, similar to the role of sigmoid function reshaping in Fellous' algorithm. For true nonlinear behavior on the space of intensity functions the nCI kernels needs to be used, with great modeling advantages as shown next.

6.3.3 Clustering of Renewal Processes by mCI and nCI Kernels

The goal of this simulation example is to show the importance of point process kernels that go beyond the first cross-moment (i.e., cross-correlation) between spike trains. For this reason, we applied the algorithm proposed here for clustering of spike trains generated as homogeneous renewal point processes with a gamma inter-spike interval (ISI) distribution. This model was chosen since the Poisson process is a particular case and thus can be directly compared.

A three cluster problem is considered, in which each cluster is defined by the ISI distribution of its spike trains (Figure 6-7(a)). In other words, spike trains within the cluster were generated according to the same point process model. All spike trains were 1s long and with constant firing rate 20 spk/s. For each Monte Carlo run, a total of 100 spike trains randomly assigned to one of the clusters were generated. The results



(a) Inter-spike interval (ISI) distributions defining each cluster.

(b) Example spike trains from each cluster.

(c) Clustering results.

Figure 6-7. Comparison of clustering performance using mCI and nCI kernels for a three cluster problem.

statistics were estimated over 500 Monte Carlo runs. For both the mCI and nCI kernels, the Gaussian function was used as smoothing function with results for three values of the smoothing width, 2, 10 and 100ms. In addition, the Gaussian kernel was utilized for $\mathcal{K}_\sigma$ in the computation of the nCI kernel, with results for kernel sizes $\sigma = 1$ and $\sigma = 10$.

The results of the simulation are shown in Figure 6-7(c). The cluster with shape parameter $\theta = 1$ contained Poisson spike trains, spike trains with shape parameter $\theta = 3$ were more regular, and $\theta = 0.5$ gave rise to more irregular (i.e. "bursty") spike trains. The results with the mCI kernel are at most 1.4% better, on average, than random selection. This low performance is not entirely surprising since all spike trains have the same constant firing rate. Using the nCI kernel with the larger smoothing width yielded an improvement of 14.7% for $\sigma = 10$ and 18% for $\sigma = 1$, on average. Smaller values of σ did not improve the clustering performance ($\sigma = 0.1$ resulted in the same performance as $\sigma = 1$), demonstrating that the selection of kernel size σ for the nCI kernel is not very problematic. But, most importantly, the results show that even though the formulation depends only on the memoryless intensity functions, in practice, the nonlinear kernel $\mathcal{K}_\sigma$ allows for different spike train models to be discriminated. This improvement is due to the fact that $\mathcal{K}_\sigma$ enhances the slight differences in the estimated intensity functions due to the different point process model expressed in the spike trains (Figure 6-7(b)).

6.4 Application for Neural Activity Analysis

To conclude this chapter, we briefly present some results on the application of this algorithm to the neural activity analyzed in Chapter 5. As mentioned then, clustering be coupled with the ICC analysis to determine which neurons to consider as an ensemble so that averaging over the ensemble can be done.

As was observed in Section 5.3.3.3, there is interesting modulation of synchrony in the motor neurons about $0.25 \sim 0.4$ seconds after the lever is released. Therefore, clustering was applied to the set of spike trains (one for each neuron) in the interval $[0.5, 1.5]$ (seconds) after the lever was release, using the mCI kernel with a Laplacian estimation

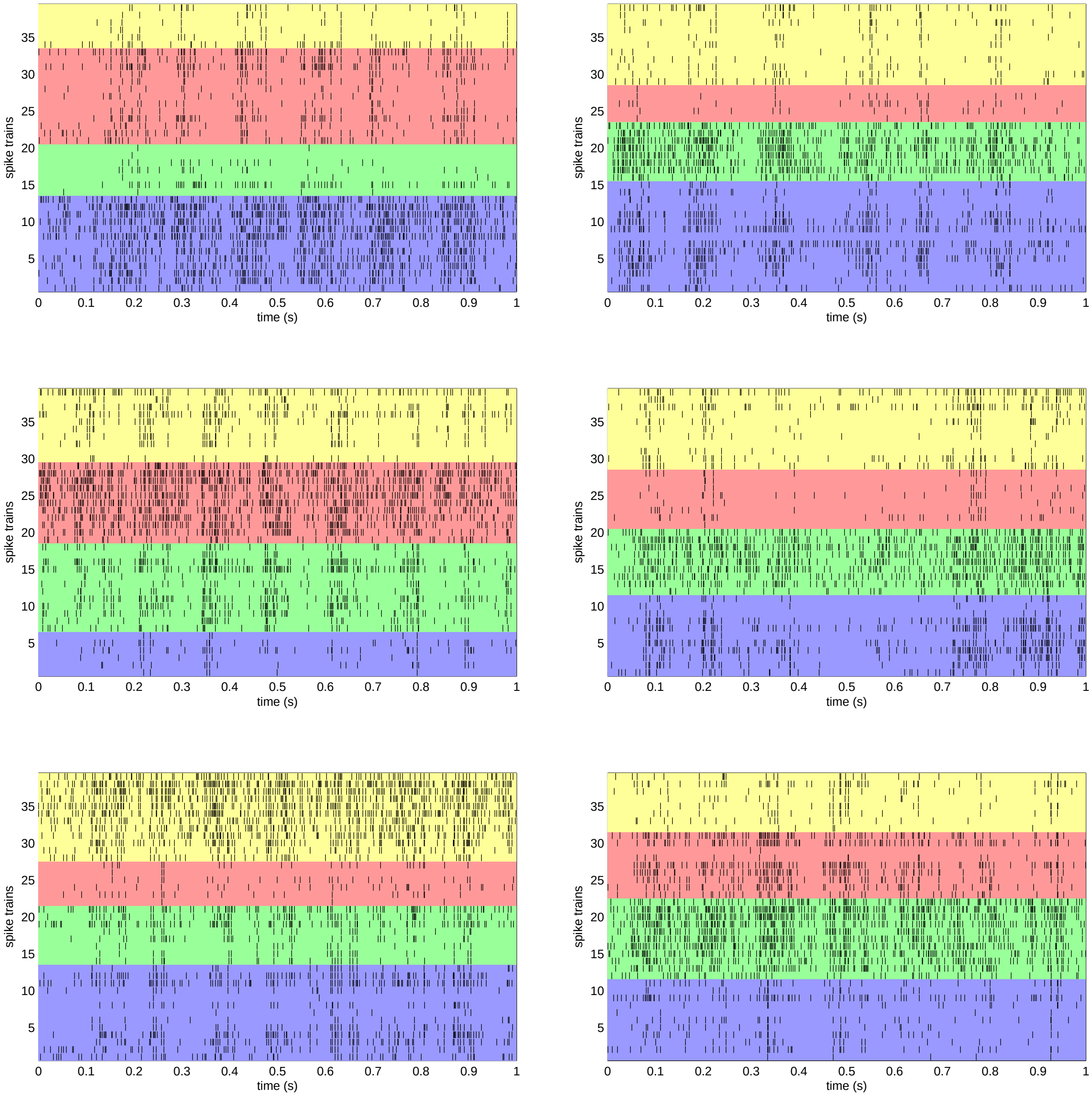


Figure 6-8. Clustering of neural activity following a lever release, assuming 4 clusters. The spike trains correspond to the moments after the lever presses in Figure 5-8.

kernel of width 2ms. The results are shown in Figure 6-8 for the same lever presses shown in Figure 5-8, considering 4 clusters. One of major difficulties when applying clustering to real datasets such as this one is how to choose the number of clusters. This is greatly complicated by the fact that all clusters share some similarity, and thus becomes quite complicated where to place a boundary. In this case, the value was chosen after trying values from 3 to 5. For this dataset, 4 clusters seems to provide a good overall distinction (visually judged from the raster plot) between clusters. (Effectively, we tried to prevent any two clusters from looking quite similar.) The problem with establishing a boundary might signify that fuzzy methods needs to be utilized, maybe simply by replacing K-means by fuzzy K-means in the effective clustering step of the spectral clustering algorithm.

From Figure 6-8 it can be verified that the clustering algorithm separates neurons based on both firing rate and synchrony, despite the small kernel size. This is very important because if ICC is applied to each cluster this results allows for different time-scales to be utilized for each cluster and enhances the various moments when synchrony occurs within cluster and across clusters. For example, in the raster plot it can observed the main synchrony rhythm also shown in the ICC plots in Figure 5-8 (e.g., red cluster in the first plot) but, in addition, it reveals other higher-frequency rhythms (e.g., yellow cluster in the first plot). Together with ICC analysis for each cluster, matching the ICC with LFP activity, and/or simply correlating these findings with the spatial placement of the micro-electrodes might reveal the role this neuronal coupling.

CHAPTER 7
PRINCIPAL COMPONENT ANALYSIS

To further illustrate the importance of the RKHS framework shown here for computation with point processes, in the following we derive the algorithm to perform principal component analysis (PCA) of realizations of point processes, and of spike trains in particular. As in Chapter 6, although we consider spike trains due to main motivation of this work, the ideas are applicable to any one-dimensional point process.

The PCA algorithm will be derived from two different perspectives. First, PCA will be derived directly in the RKHS induced by a point process kernel. This perspective shows the usefulness of the RKHS framework for optimization, and highlights that optimization with realizations of point processes is possible by the definition of an inner product for the point process realizations, and more specifically through the mathematical structure provided by the RKHS. This is also the traditional approach in the functional analysis literature Ramsay and Silverman [1997] and has the advantage of being completely general, regardless of the actual point process kernel definition used. A well known example of discrete PCA done in an RKHS is kernel PCA [Schölkopf et al., 1998].

In the second approach we will derive PCA in the space spanned by the intensity functions utilizing the inner product defined in this space. Thus, this perspective is applicable only for linear CI kernels. The derivation shown here considers the mCI kernel but the same can be derived in terms of the conditional intensity functions for general linear CI kernels. Since for these point process kernels the RKHS is congruent to this space the inner products in the two spaces are isometric, and therefore the outcome will be found to be the same. However, this approach has the advantage that it explicitly makes available the eigenfunctions as (scaled) intensity functions. This is important in many neurophysiological studies since the researcher is often interested in understanding the undergoing process in the neuronal network, as expressed by the intensity functions. Note that, in general, the eigenfunctions are not available in the RKHS because the transformation to the RKHS is unknown. However, this approach is possible here due to the linearity of the space spanned by the intensity functions with the inner product we defined.

7.1 Optimization in the RKHS

Suppose we are given a set of spike trains, $\{s_1, s_2, \ldots, s_N\}$, for which we wish to determine the principal components. Computing the principal components of the spike trains directly is not feasible because we would not know how to define a principal component (PC), however, this is a trivial task in an RKHS.

Let $\{\Lambda_{s_i} \in \mathcal{H}_I, i = 1, \ldots, N\}$ be the set of elements in the RKHS $\mathcal{H}_I$ corresponding to the given spike trains. Note that, correctly speaking, $\Lambda_\cdot$ denotes the transformation for a point process into the RKHS, and for which the inner product is the point process kernel. In spite of that, this chapter deals exclusively with point process realizations and therefore, with some abuse of notation, Λ_{s_i} shall be used to denote the "transformed spike trains." Then, the inner product of Λ_{s_i}'s is in effect the *estimator* of the point process kernel.

Denote the mean of the transformed spike trains as

$$\bar{\Lambda} = \frac{1}{N} \sum_{i=1}^{N} \Lambda_{s_i}, \qquad (7\text{–}1)$$

and the centered transformed spike trains (i.e., with the mean removed) can be obtained as

$$\tilde{\Lambda}_{s_i} = \Lambda_{s_i} - \bar{\Lambda}. \qquad (7\text{–}2)$$

PCA finds an orthonormal transformation providing a compact description of the data. Determining the principal components of spike trains in the RKHS can be formulated as the problem of finding the set of orthonormal vectors in the RKHS such that the projection of the centered transformed spike trains $\{\tilde{\Lambda}_{s_i}\}$ has the *maximum variance*. This means that the principal components can be found by solving the following optimization problem in the RKHS: a function $\xi \in \mathcal{H}_I$ (i.e., $\xi : \mathcal{P}(\mathcal{T}) \longrightarrow \mathbb{R}$) is a principal component if it maximizes the cost function

$$J(\xi) = \sum_{i=1}^{N} \left[\mathrm{Proj}_{\xi}(\tilde{\Lambda}_{s_i})\right]^2 - \rho\left(\|\xi\|^2 - 1\right) \tag{7–3}$$

where $\mathrm{Proj}_{\xi}(\tilde{\Lambda}_{s_i})$ denotes the projection of the ith centered transformed spike train onto ξ, and ρ is the Lagrange multiplier to the constraint $\left(\|\xi\|^2 - 1\right)$ imposing that the principal components have unit norm. To evaluate this cost function one needs to be able to compute the projection and the norm of the principal components. However, in an RKHS, an inner product is the projection operator and the norm is naturally defined. Thus, the above cost function can be expressed as

$$J(\xi) = \sum_{i=1}^{N} \left\langle \tilde{\Lambda}_{s_i}, \xi \right\rangle_{\mathcal{H}_I}^2 - \rho\left(\langle \xi, \xi\rangle_{\mathcal{H}_I} - 1\right), \tag{7–4}$$

Because in practice we always have a finite number of spike trains, ξ is restricted to the subspace spanned by the centered transformed spike trains $\{\tilde{\Lambda}_{s_i}\}$. Consequently, there exist coefficients $b_1, \ldots, b_N \in \mathbb{R}$ such that

$$\xi = \sum_{j=1}^{N} b_j \tilde{\Lambda}_{s_j} = \mathbf{b}^T \tilde{\boldsymbol{\Lambda}} \tag{7–5}$$

where $\mathbf{b}^T = [b_1, \ldots, b_N]$ and $\tilde{\boldsymbol{\Lambda}}(t) = \left[\tilde{\Lambda}_{s_1}(t), \ldots, \tilde{\Lambda}_{s_N}(t)\right]^T$. Substituting in Equation 7–4 yields

$$\begin{aligned} J(\xi) &= \sum_{i=1}^{N} \left(\sum_{j=1}^{N} b_j \left\langle \tilde{\Lambda}_{s_i}, \tilde{\Lambda}_{s_j} \right\rangle\right) \left(\sum_{k=1}^{N} b_k \left\langle \tilde{\Lambda}_{s_i}, \tilde{\Lambda}_{s_k} \right\rangle\right) \\ &\quad + \rho\left(1 - \sum_{j=1}^{N}\sum_{k=1}^{N} b_j b_k \left\langle \tilde{\Lambda}_{s_i}, \tilde{\Lambda}_{s_k} \right\rangle\right) \\ &= \mathbf{b}^T \tilde{\mathbf{I}}^2 \mathbf{b} + \rho\left(1 - \mathbf{b}^T \tilde{\mathbf{I}} \mathbf{b}\right). \end{aligned} \tag{7–6}$$

where $\tilde{\mathbf{I}}$ is the Gram matrix of the centered spike trains; that is, the $N \times N$ matrix with elements

$$\begin{aligned}
\tilde{\mathbf{I}}_{ij} &= \left\langle \tilde{\Lambda}_{s_i}, \tilde{\Lambda}_{s_j} \right\rangle \\
&= \left\langle \Lambda_{s_i} - \bar{\Lambda}, \Lambda_{s_j} - \bar{\Lambda} \right\rangle \\
&= \left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle - \frac{1}{N} \sum_{l=1}^{N} \left\langle \Lambda_{s_i}, \Lambda_{s_l} \right\rangle - \frac{1}{N} \sum_{l=1}^{N} \left\langle \Lambda_{s_l}, \Lambda_{s_j} \right\rangle + \frac{1}{N^2} \sum_{l=1}^{N} \sum_{n=1}^{N} \left\langle \Lambda_{s_l}, \Lambda_{s_n} \right\rangle .
\end{aligned} \tag{7–7}$$

In matrix notation,

$$\tilde{\mathbf{I}} = \mathbf{I} - \frac{1}{N}(\mathbf{1}_N \mathbf{I} + \mathbf{I}\mathbf{1}_N) + \frac{1}{N^2} \mathbf{1}_N \mathbf{I} \mathbf{1}_N, \tag{7–8}$$

where $\mathbf{I}$ is the Gram matrix of the inner product of spike trains $\mathbf{I}_{ij} = \left\langle \Lambda_{s_i}, \Lambda_{s_j} \right\rangle$, and $\mathbf{1}_N$ is the $N \times N$ matrix with all ones. This means that $\tilde{\mathbf{I}}$ can be computed directly in terms of $\mathbf{I}$ without the need to explicitly remove the mean of the transformed spike trains.

From Equation 7–6, finding the principal components simplifies to the problem of estimating the coefficients $\{b_i\}$ that maximize $J(\xi)$. Since $J(\xi)$ is a quadratic function its extrema can be found by equating the gradient to zero. Taking the derivative with regards to $\mathbf{b}$ (which characterizes ξ) and setting it to zero results in

$$\frac{\partial J(\xi)}{\partial \mathbf{b}} = 2\tilde{\mathbf{I}}^2 \mathbf{b} - 2\rho \tilde{\mathbf{I}} \mathbf{b} = 0, \tag{7–9}$$

and thus corresponds to the eigendecomposition problem[1]

$$\tilde{\mathbf{I}} \mathbf{b} = \rho \mathbf{b}. \tag{7–10}$$

This means that any eigenvector of the centered Gram matrix is a solution of Equation 7–9. Thus, the eigenvectors determine the coefficients of Equation 7–5 and characterize the principal components. It is easy to verify that, as expected, the variance of the projections

[1] Note that the simplification in the eigendecomposition problem is valid regardless if the Gram matrix is invertible or not, since $\tilde{\mathbf{I}}^2$ and $\tilde{\mathbf{I}}$ have the same eigenvectors and the eigenvalues of $\tilde{\mathbf{I}}^2$ are the eigenvalues of $\tilde{\mathbf{I}}$ squared.

onto each principal component equals the corresponding eigenvalue squared. So, the ordering of ρ specifies the relevance of the principal components.

To compute the projection of a given input spike train s onto the kth principal component (corresponding to the eigenvector with the kth largest eigenvalue) we need only to compute in the RKHS the inner product of Λ_s with ξ_k. That is,

$$\begin{aligned}\mathrm{Proj}_{\xi_k}(\Lambda_s) &= \langle \Lambda_s, \xi_k \rangle_{\mathcal{H}_I} \\ &= \sum_{i=1}^{N} b_{ki} \left\langle \Lambda_s, \tilde{\Lambda}_{s_i} \right\rangle \\ &= \sum_{i=1}^{N} b_{ki} \left(I(s, s_i) - \frac{1}{N} \sum_{j=1}^{N} I(s, s_j) \right).\end{aligned} \tag{7–11}$$

We emphasize once more that no property specific of a point process kernel was utilized in the derivation. Indeed, it utilizes only the linear vector space structure provided by the RKHS for optimization and computation. Therefore, any of the point process kernels proposed in this dissertation can be utilized.

7.2 Optimization in the Space Spanned by the Intensity Functions

As before, let $\{s_1, s_2, \ldots, s_N\}$ denote the set of spike trains for which we wish to determine the principal components, and $\{\lambda_{s_i}(t), t \in \mathcal{T}, i = 1, \ldots, N\}$ the corresponding intensity functions. The mean intensity function is

$$\bar{\lambda}(t) = \frac{1}{N} \sum_{i=1}^{N} \lambda_{s_i}(t), \tag{7–12}$$

and therefore the centered intensity functions are

$$\tilde{\lambda}_{s_i}(t) = \lambda_{s_i}(t) - \bar{\lambda}(t). \tag{7–13}$$

Again, the problem of finding the principal components of a set of data can be stated as the problem of finding the eigenfunctions of unit norm such that the projections have maximum variance. This can be formulated in terms of the following optimization problem. A function $\zeta(t) \in L_2(\lambda_{s_i}(t), t \in \mathcal{T})$ is a principal component if it maximizes the cost function

$$\begin{aligned} J(\zeta) &= \sum_{i=1}^{N} \left[\mathrm{Proj}_{\zeta}(\tilde{\lambda}_{s_i})\right]^2 - \gamma\left(\|\zeta\|^2 - 1\right) \\ &= \sum_{i=1}^{N} \left\langle \tilde{\lambda}_{s_i}, \zeta \right\rangle_{L_2}^2 - \gamma\left(\|\zeta\|^2 - 1\right), \end{aligned} \tag{7–14}$$

where γ is the Lagrange multiplier constraining ζ to have unit norm. It can be shown that $\zeta(t)$ lies in the subspace spanned by the intensity functions $\{\tilde{\lambda}_{s_i}(t), i = 1, \ldots, N\}$. Therefore, there exist coefficients $b_1, \ldots, b_N \in \mathbb{R}$ such that

$$\zeta(t) = \sum_{j=1}^{N} b_j \tilde{\lambda}_{s_j}(t) = \mathbf{b}^T \tilde{\mathbf{r}}(t). \tag{7–15}$$

with $\mathbf{b}^T = [b_1, \ldots, b_N]$ and $\tilde{\mathbf{r}}(t) = \left[\tilde{\lambda}_{s_1}(t), \ldots, \tilde{\lambda}_{s_N}(t)\right]^T$. Substituting in Equation 7–4 yields

$$\begin{aligned} J(\zeta) = &\sum_{i=1}^{N} \left(\sum_{j=1}^{N} b_j \left\langle \tilde{\lambda}_{s_i}, \tilde{\lambda}_{s_j} \right\rangle\right)\left(\sum_{k=1}^{N} b_k \left\langle \tilde{\lambda}_{s_i}, \tilde{\lambda}_{s_k} \right\rangle\right) \\ &+ \gamma\left(1 - \sum_{j=1}^{N}\sum_{k=1}^{N} b_j b_k \left\langle \tilde{\lambda}_{s_i}, \tilde{\lambda}_{s_k} \right\rangle\right) \\ = &\,\mathbf{b}^T \tilde{\mathbf{I}}^2 \mathbf{b} + \gamma\left(1 - \mathbf{b}^T \tilde{\mathbf{I}} \mathbf{b}\right). \end{aligned} \tag{7–16}$$

where $\tilde{\mathbf{I}}$ is the gram matrix of the centered intensity functions (i.e., $\tilde{\mathrm{I}}_{ij} = \left\langle \tilde{\lambda}_{s_i}, \tilde{\lambda}_{s_j} \right\rangle_{L_2}$). Therefore, this derivation is only valid for point process kernels for which the inner product is explicitly defined in the space of intensity functions (in general, conditional intensity functions).

As expected, since in this case the RKHS and the space of intensity functions are congruent because the inner product produces the same result, this cost function yields the same solution. However, unlike the previous, this presentation has the advantage that it shows the role of the eigenvectors of the gram matrix and, most importantly, how to obtain the principal component functions in the space of intensity functions. From Equation 7–15, the coefficients of the eigenvectors of the gram matrix provide a weighting

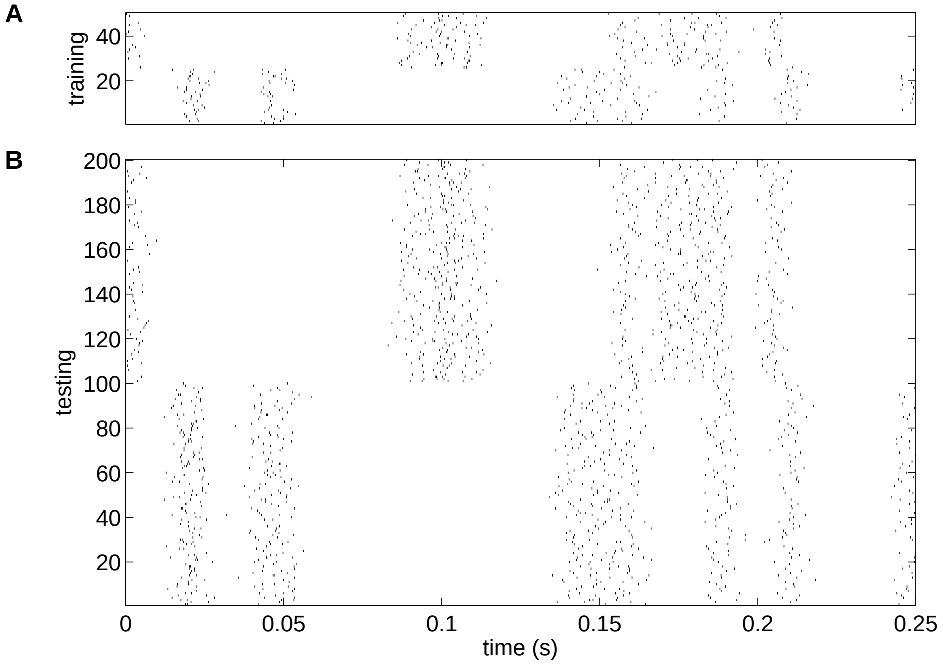


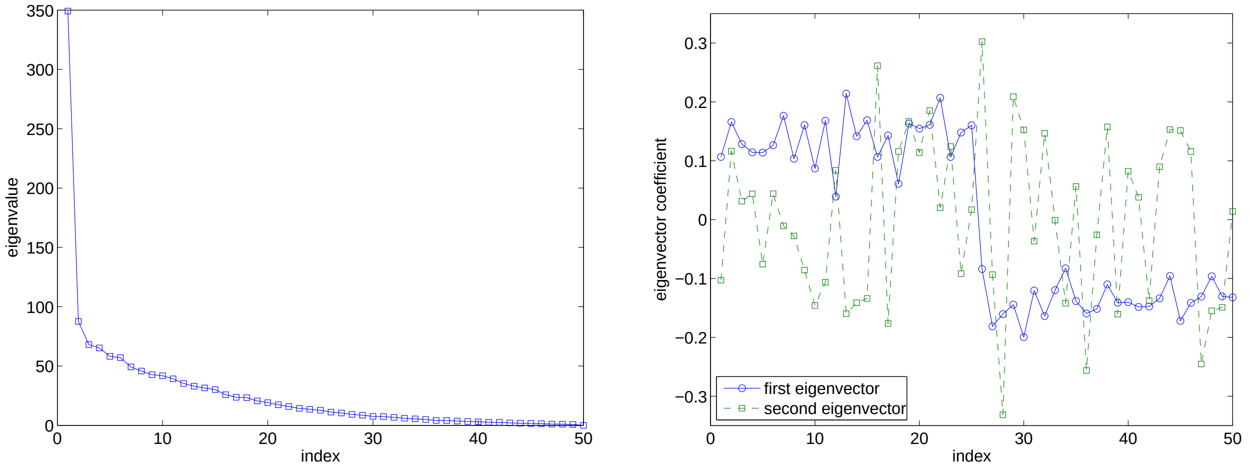
Figure 7-1. Spike trains used for evaluation of the eigendecomposition coefficients of PCA algorithm (A), and for testing of the result (B). In either case, the first half of spike trains corresponds to the first template and the remaining to the second template.

for the intensity functions of each spike trains and therefore expresses how important a spike train is to represent others. In a different perspective, this suggests that the principal component functions should reveal general trends in the intensity functions of the input spike trains.

7.3 Results

7.3.1 Comparison with Binned Cross-Correlation

To illustrate the algorithm just derived, and to compare the use of the mCI kernel with binned cross-correlation (CC) in this task, we performed a simple experiment. We generated two template spike trains comprising of 10 spikes uniformly random distributed over an interval of 0.25s. In a specific application these template spike trains could correspond, for example, to the average response of a culture of neurons to two distinct but fixed input stimuli. For the computation of the coefficients of the eigendecomposition ("training set"), we generated a total of 50 spike trains, half for each template, by



(a) Eigenvalues in decreasing order.

(b) First two eigenvectors of the eigendecomposition of the Gram matrix.

Figure 7-2. Eigendecomposition of the centered Gram matrix $\tilde{\mathbf{I}}$.

randomly copying each spike from the template with probability 0.8 and adding zero mean Gaussian distributed jitter with standard deviation 3ms. For testing of the obtained coefficients, 200 spike trains were generated following the same procedure. The simulated spike trains are shown in Figure 7-1.

According to the PCA algorithm derived previously, we computed the eigendecomposition of the matrix $\tilde{\mathbf{I}}$ as given by Equation 7–8 so that it solves Equation 7–10. The evaluation of the mCI kernel was estimated from the spike trains according to Equation 3–27, and computed with a Gaussian kernel with size 2ms. The eigenvalues $\{\rho_l, l = 1, \ldots, 100\}$ and first two eigenvectors are shown in Figure 7-2. The first eigenvalue alone accounts for more than 26% of the variance of the dataset in the RKHS space. Although this value is not impressive, its importance is clear since it is nearly 4 times higher than the second eigenvalue (6.6%). Furthermore, notice that the first eigenvector clearly shows the separation between spike trains generated from different templates (Fig. 7-2(b)). This again can be seen in the first principal component function, shown in Figure 7-3, which reveals the location of the spike times used to generate the templates while discriminating between them with opposite signs. Around periods of time where the spike from both

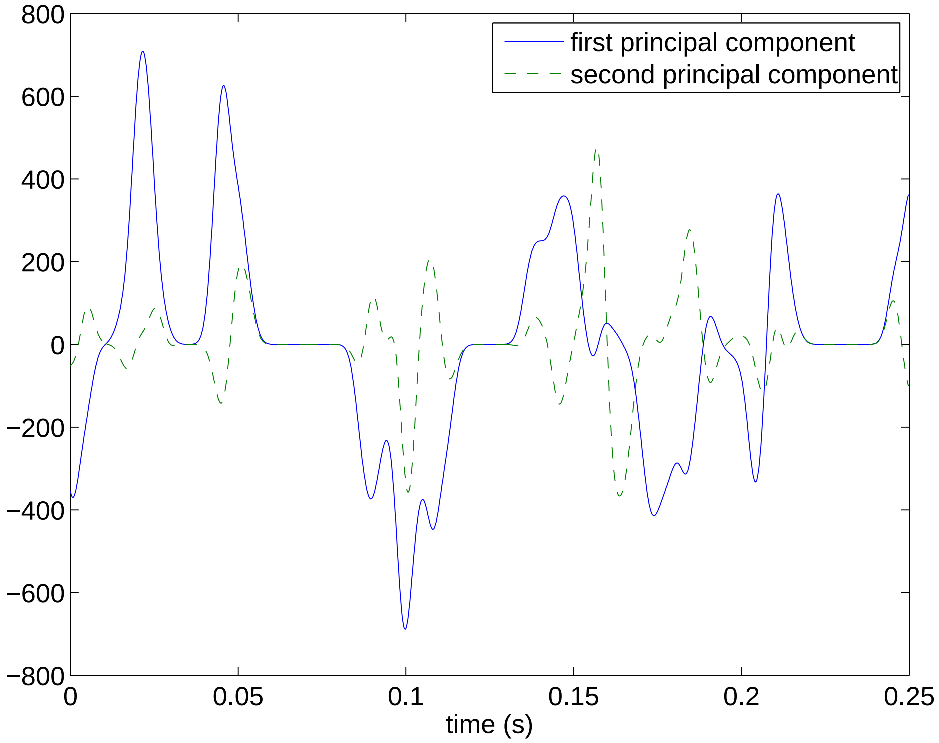
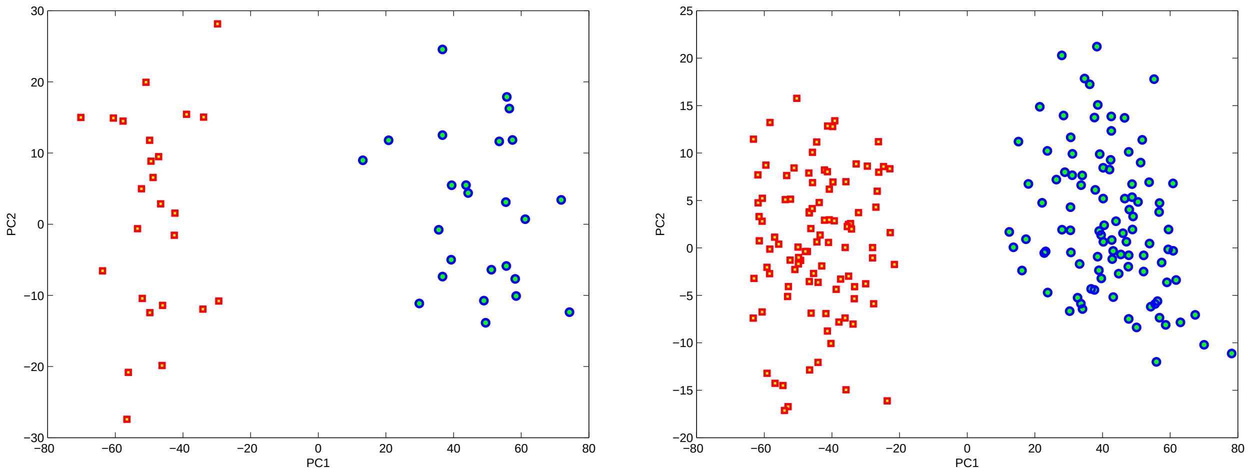


Figure 7-3. First two principal component functions (i.e., eigenfunctions) in the space of intensity functions. They are computed by substituting the coefficients of the first two eigenvectors of the Gram matrix in Equation 7–15.



(a) Projection of the spike trains in the training set.

(b) Projection of the spike trains in the testing set.

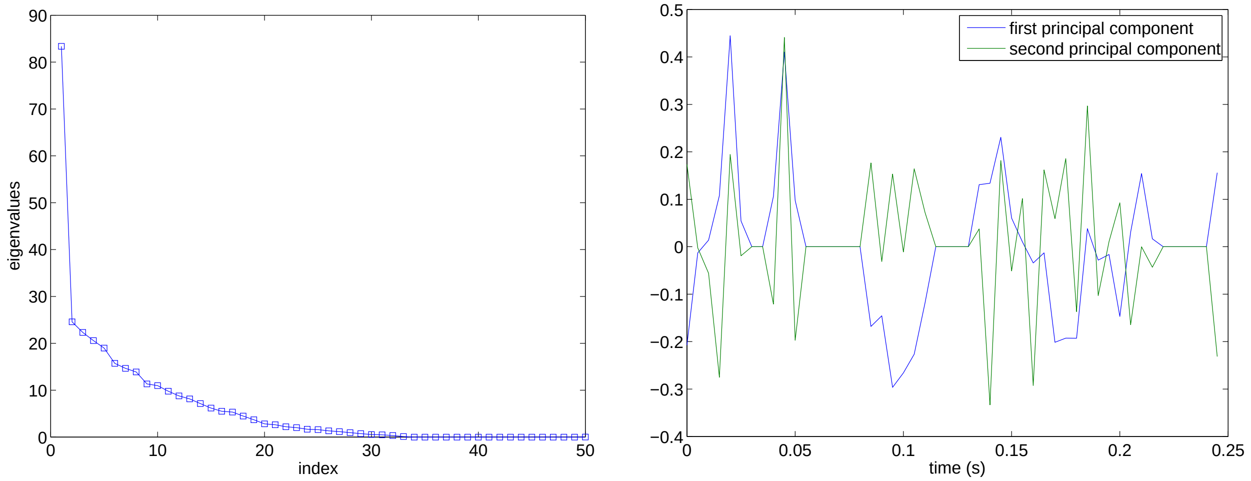
Figure 7-4. Projection of spike trains onto the first two principal components using mCI kernel. The different point marks differentiate between spike trains corresponding to each one of the classes.

templates overlap the first principal component is zero. As can be seen from the second principal component function, the role of the second eigenvector is to account for the dispersion in the data capable of differentiate spike trains generated from different templates, especially around the times where they overlap.

Both datasets, for evaluation and testing, where projected onto the first two principal components. Figure 7-4 shows the projected spike trains. As noted from the difference between the first and second eigenvalues, the first principal component is the main responsible for the dispersion between classes of the projected spike trains. This happens because the direction of maximum variance is the one that passes through both clusters of points in the RKHS due to the small dispersion within class. The second principal component seems to be responsible for dispersion due to the jitter noise introduced in the spike trains, and suggests that other principal components may play a similar role.

A more specific understanding can be obtained from the considerations done in Section 3.5.3. There, the congruence between the RKHS induced by the mCI kernel, $\mathcal{H}_I$, and the RKHS induced by κ, $\mathcal{H}_\kappa$, was utilized to show that the mCI kernel is inversely related to the variance of the transformed spike times in $\mathcal{H}_\kappa$. In this dataset and for the kernel size utilized, this guaranties that the value of the mCI kernel within class is always smaller than inter class. This is a reason why in this scenario the first principal component always suffices to project the data in a way that distinguishes between spike trains generated each of the templates.

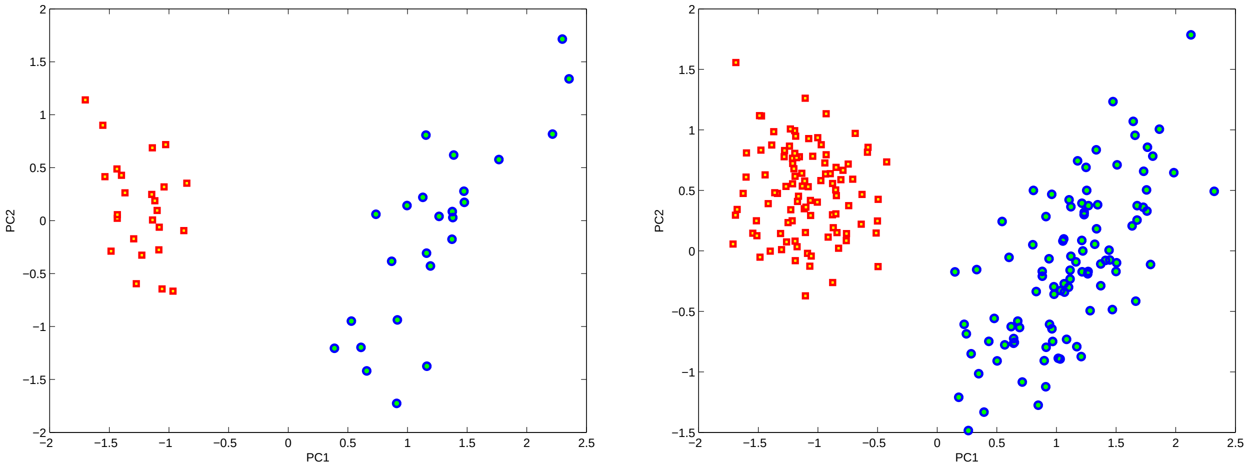
PCA was also applied to this dataset using binned spike trains. Although cross-correlation is an inner product for spike trains and therefore the above algorithm could have been used, for comparison the conventional approach was followed [Richmond and Optican, 1987; McClurkin et al., 1991]. That is, to compute the covariance matrix with each binned spike train taken as a data vector. This means that the dimensionality of the covariance matrix is determined by the number of bins per spike train, which may be problematic if long spike trains are used or small bin sizes are needed for high temporal resolution.



(a) Eigenvalues in decreasing order.

(b) First two eigenvectors/eigenfunctions of the eigendecomposition of the covariance matrix.

Figure 7-5. Eigendecomposition of the covariance matrix.



(a) Projection of the spike trains in the training set.

(b) Projection of the spike trains in the testing set.

Figure 7-6. Projection of spike trains onto the first two principal components of the covariance matrix of binned spike trains. The different point marks differentiate between spike trains corresponding to each one of the templates.

The results of PCA using bin size of 5ms are shown in Figure 7-5 and Figure 7-6. The bin size was chosen to provide a good compromise between temporal resolution and smoothness of the eigenfunctions (important for interpretability). Comparing these results the ones using the mCI kernel, the distribution of the eigenvalues is quite similar and the first eigenfunction does reveals somewhat of the same trend as in Figure 7-3. The same is not true for the second eigenfunction, however, which looks much more jaggy. In fact, as Figure 7-6 shows, in this case the projections along the first two principal directions are not orthogonal. This means that the covariance matrix does not fully express the structure of the spike trains. It is noteworthy that this is not only because the covariance matrix is being estimated with a small number of data vectors. In fact, when the binned cross-correlation was utilized directly in the above algorithm as the inner product the same effect was observed, meaning that the *binned cross-correlation does not characterize the spike train structure in sufficient detail.* Since the binned cross-correlation and the mCI kernel are conceptually equivalent apart from the discretization introduced by binning, this proves the ill effects of this preprocessing step for analysis and computation with spike train, and point process realizations in general.

7.3.2 PCA of Renewal Processes

PCA is, in essence, a filtering operation. Therefore, the mCI and nCI kernels are now compared for PCA of renewal processes. Basically, a paradigm similar to the one utilized in the previous section was employed. Two datasets were generated: for computation of the eigendecomposition (i.e., “training”) with 50 spike trains, and for testing with 200 spike trains. For each dataset, the spike trains were generated from two renewal point process models with gamma distributed inter-spike intervals (shape parameter $\theta = 0.5$ and $\theta = 3$), one half from each model. All spike trains were 1 second long and with mean firing rate 20 spk/s. The simulated spike trains are shown in Figure 7-7.

Then, the algorithm derived in Section 7.1 was applied using both the mCI and nCI kernel. Recall that the PCA algorithm is independent of the point process kernel used,

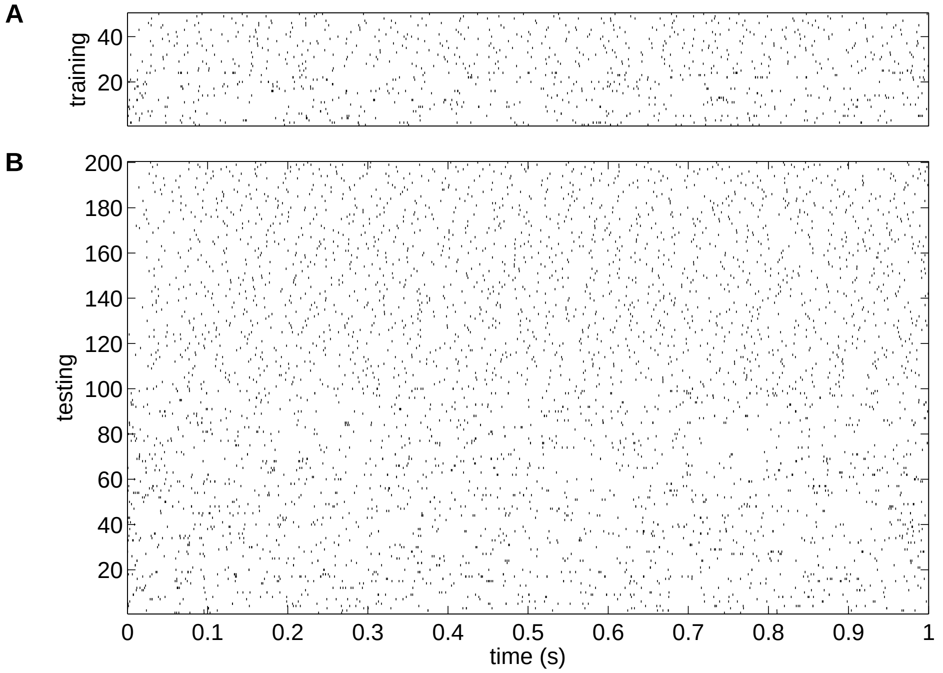
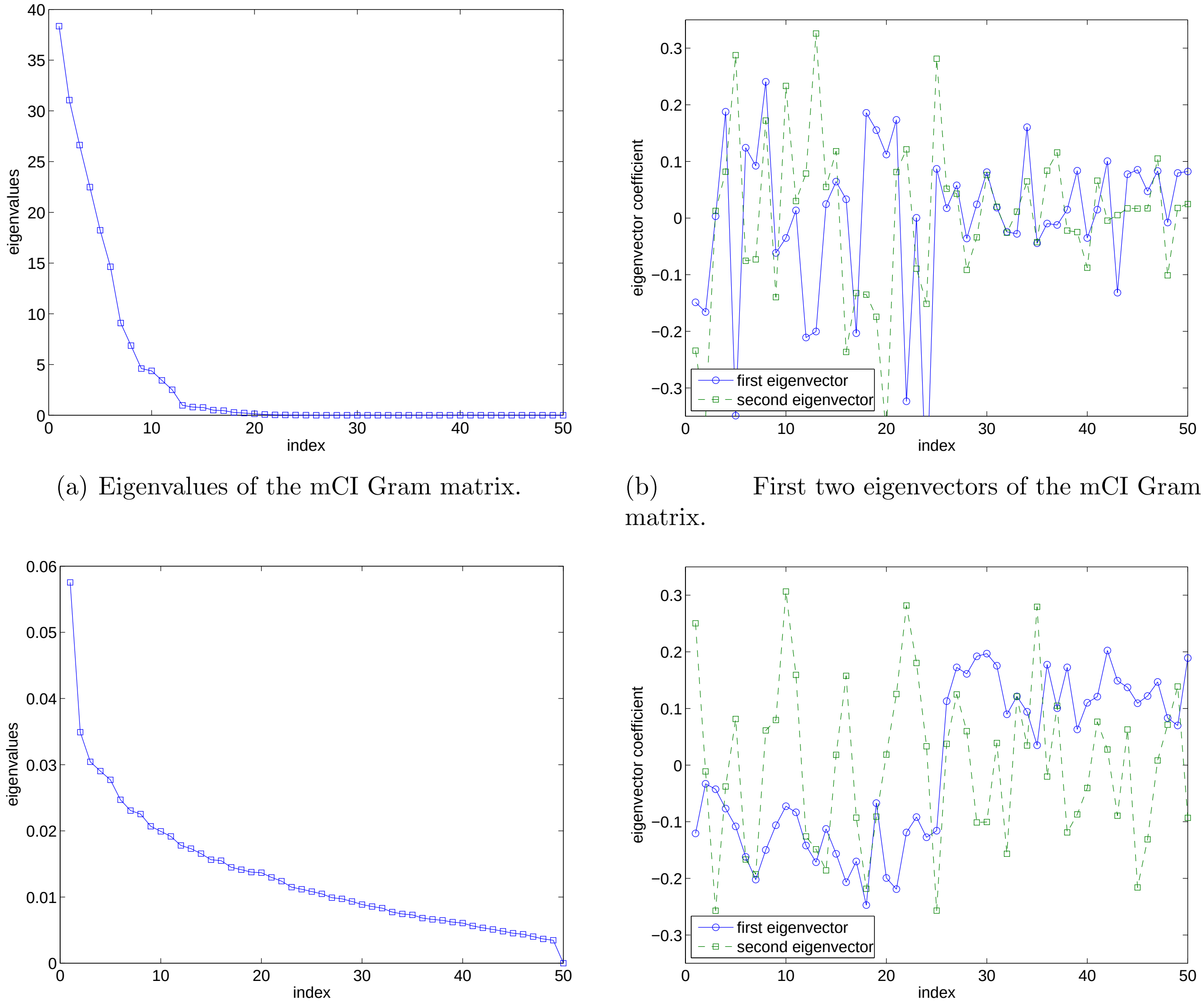


Figure 7-7. Spike trains from renewal point processes for comparison of mCI with nCI kernel. (A) "Training" spike trains for evaluation of the eigendecomposition coefficients of the PCA algorithm, and (B) for testing of the result (B). Each dataset (training and testing) is divided in two halves, each corresponding to one of the renewal point process models.

the only difference is which kernel is used to compute the Gram matrix of the spike trains. The results of the eigendecomposition are shown in Figure 7-8. Although the spike train variability is concentrated in a smaller number of dimensions for the mCI kernel than the nCI kernel, there a clear distinctions between the contribution of the first and second principal components in the latter case (first PC almost twice as important as second PC). This can be judged more easily in the first eigenvector of the eigendecomposition, which in the case of the nCI kernel shows that the first principal component separates spike trains generated from different renewal point process model. The relevance of this observation can asserted in the projections of the dataset, shown in Figure 7-9. For the mCI kernel case, the projections from the two point process models overlap greatly, being only noticeable the higher dispersion of spike trains from the first renewal model ($\theta = 0.5$) due to their more irregular firing. For case using the nCI kernel, however, the



(a) Eigenvalues of the mCI Gram matrix.

(b) First two eigenvectors of the mCI Gram matrix.

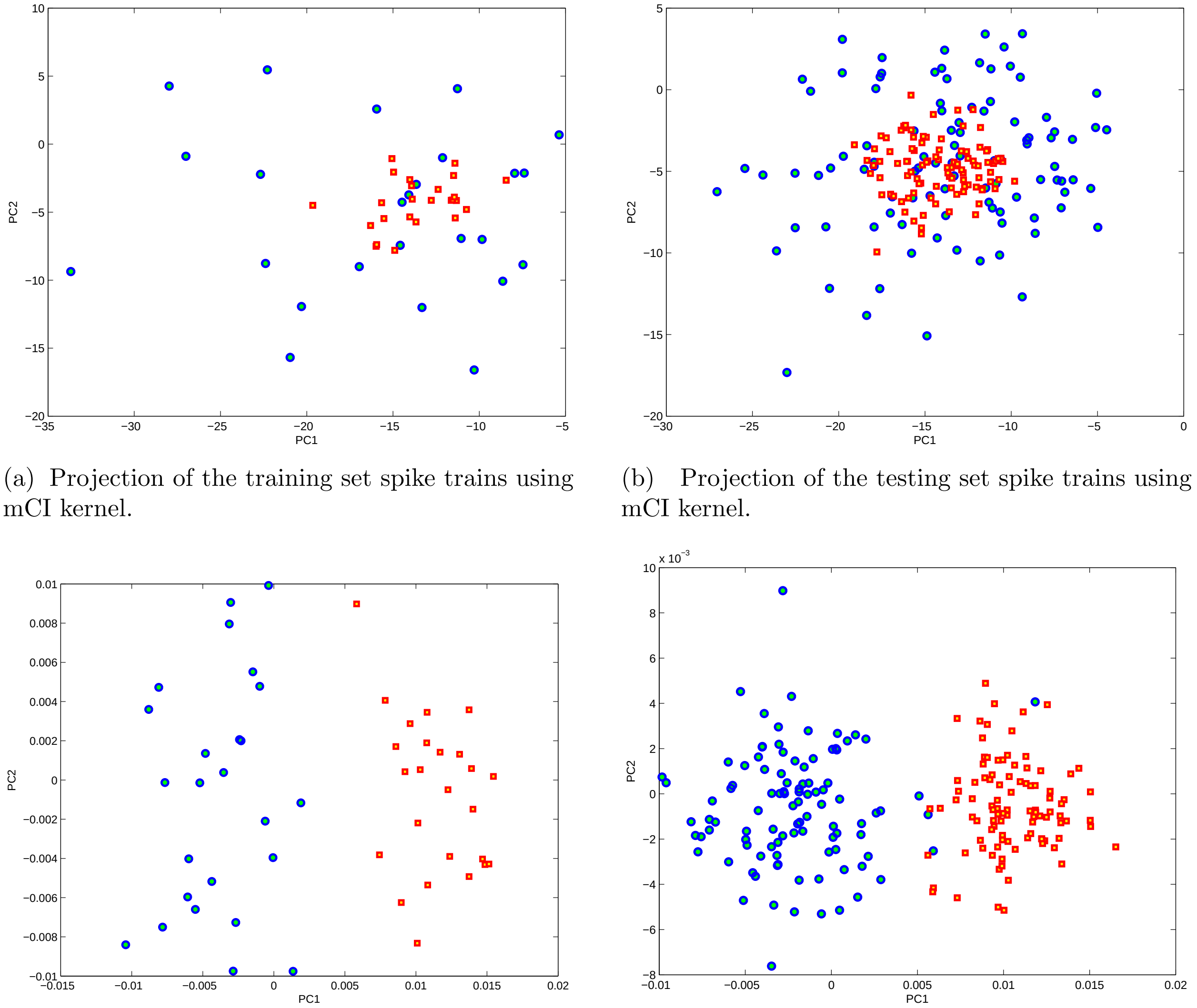
(c) Eigenvalues of the nCI Gram matrix.

(d) First two eigenvectors of the nCI Gram matrix.

Figure 7-8. Eigendecomposition of the Gram matrix, for the mCI and nCI kernels.

first principal component alone is responsible for the separation between spike trains from the two renewal models, as had been noted in Figure 7-8(d).

These results verify once more the generality of the nCI kernel, by being able to quantify and discriminate between renewal point process models. More importantly, the projection results in Figure 7-9 reveal that the use of point process kernels capable of coping with the point process model is very important in ensuring that the data is transformed into the RKHS while preserving the model differences. Put differently, a proper point process kernel for a given model certifies that the RKHS is rich enough so



(a) Projection of the training set spike trains using mCI kernel.

(b) Projection of the testing set spike trains using mCI kernel.

(c) Projection of the training set spike trains using nCI kernel.

(d) Projection of the testing set spike trains using nCI kernel.

Figure 7-9. Projection of renewal spike trains onto the first two principal components using mCI and nCI kernels. The different point marks differentiate between spike trains corresponding to each one of the templates.

that linear suffice in analysing and processing the transformed data in this space. Because, in practice the true underlying point process model is unknown the safest choice is to whenever possible to test using the most general point process kernel and compare with simpler kernels to infer about the complexity of the underlying model.

CHAPTER 8
CONCLUSION AND TOPICS FOR FUTURE DEVELOPMENTS

8.1 Conclusion

The peculiar nature of point process has made the application of conventional signal processing methods to their realizations difficult and imprecise to apply from first principles. In this respect, binning is currently the standard approach since it transforms the point process into a discrete-time random process such that conventional signal processing methods can be used. However, binning is an imprecise mapping since information is irreversibly lost from the point process realizations (see, for example, Section 7.3.1). The most powerful methodologies to point process analysis are based on statistical approaches since the distributions are estimated directly, thus, fully characterizing the point process. But such methodologies face serious shortcomings when multiple point processes and their couplings are considered simultaneously, since they are only practical using an assumption of independence. Nevertheless, processing of multiple point processes is very important for practical applications, such as neural activity analysis, with the widespread use of multielectrode array techniques.

This dissertation presents a reproducing kernel Hilbert space (RKHS) framework for the analysis of point processes that has the potential to improve the set of methods and algorithms that can be developed for point process analysis. The main goal of this dissertation was to present the fundamental theory in order to establish a solid foundation and hopefully entice further work along this line of reasoning. Indeed, the dual role of the dissertation is to elucidate the set of possibilities that are open by the RKHS formulation and to link the theory to methods that are in common use. So further work is needed to bring the possibilities open by RKHS theory to fruition in point process signal analysis.

The core concept of RKHS theory is the concept of inner product which is also the fundamental operator for signal processing with point processes. Therefore much of the contributions of this work at the theoretical level focused on showing how point process kernels could be defined, in terms of kernels on event coordinates or the statistical descriptors of the point processes. The latter approach is, in a sense, an extension of the early work of Parzen [1959] on stochastic processes to point processes by defining bottom-up the structure of the RKHS on the statistics of the point processes; that is, the conditional intensity functions (in general). This result provides a solid foundation for future work both for practical algorithm development but also on a simple way to bring into the analysis more realistic assumptions about the statistics of point processes. Indeed we show that the Poisson statistical model is behind the simplest definition of the RKHS (the memoryless cross-intensity kernel) and that this RKHS provides a linear space for doing signal processing with point processes. However, the same framework can be applied to inhomogeneous Markov interval of even more general point process models which only now are beginning to be explored. We would like to emphasize that building a RKHS bottom-up is a much more principled approach than the conventional way that RKHS are derived in machine learning, where the link to data statistics is only possible at the level of the estimated quantities, not the statistical operators themselves.

Another theoretical contribution is to show the flexibility of the RKHS framework. Indeed it is possible to define alternate, and yet unexplored, RKHS for point process analysis that are not linearly related to the intensity functions. Obviously, this will provide many possible avenues for future research and there is the hope that it will be possible to derive systematic approaches to tailor the RKHS definition to the goal of the data analysis. There are basically two different types of RKHS that mimic exactly the two methodologies being developed in the machine learning and signal processing literatures: kernels that are data independent (κ) and kernels that are data dependent (CI kernels). Specifically for point processes, we show in a specific case how that the former may be used to compose the latter, but they work with the data in very different ways. But what is interesting is that these two types of RKHS provide different features in the transformation to the space of functions. The former is a macroscopic descriptor of the spike time intervals that may be usable in coarse analysis of the data. The latter is a functional descriptor of the data but it is harder to compute. In current methods only the latter is being pursued in the form of binned cross-correlation, but by analogy with the large impact of kernel methods in statistical learning, an equally important impact of the former may be expected. And yet, the theory and the operators presented this far will form the foundations for such future developments.

There are also practical implications of the RKHS methodology presented in this dissertation. Since the RKHS is a vector space with an inner product, all the conventional signal processing algorithms that involve inner product computations can be immediately implemented for point processes in the RKHS. This was illustrated in Chapters 6 and 7, by deriving algorithms for clustering and PCA, but many other applications are possible, such as filtering. Note that the clustering algorithm shown could also be derived using common distances measures that have been defined as has been done before [Paiva et al., 2007]. But we stress the elegance of the proposed formulation that first defines the structure of the space (the inner product) and then leaves for the users the design of their intended algorithm, unlike the approaches presented so far which are specific for the application. The same can be observed in the derivation of the PCA algorithm where the derivation occurs in the RKHS and in a way that is independent of the actual RKHS induced by the point process kernel. This is advantageous as advances in point process kernels may be incorporated in the derived algorithms upon their availability, without the need to restructure the implementation. This was done for both clustering and PCA for the comparison of the mCI and nCI kernels. Indeed, in both cases only the nCI showed sensitivity to the parameters of the renewal point processes. Since in practice the true point process model is unknown, the nCI kernel is preferable as it can accommodate point process models beyond Poisson. The trade-off in doing so is that, in the case of the nonlinear CI kernels defined, another kernel size parameter (of $\mathcal{K}_\sigma$) needs to be selected, even though in our experiments the results depended on this parameter quite coarsely.

The RKHS framework is also of high relevance for development of point process analysis tools. It was shown that the simplest of the CI kernels considered is fundamentally equal to the generalized cross-correlation (GCC) which extends the more common binned cross-correlation. This exposes the limitations of current methodologies as it brings forth the implicit dependence on the Poisson point process model. Therefore, current approaches can accurately quantify at most interactions in the rate functions. The good news are that point process kernels capable of coping with more general point process models were shown here. These kernels are properly defined covariance functions (Section 3.5.4) which current analysis often utilize. Hence, they can replace binned cross-correlation (or GCC) without major changes in current paradigms.

There are still other topics that need to be researched for a fully systematic use of the technique. Perhaps the most important one for practical applications is the kernel size parameter of the kernel function. The theory shows clearly the role of this free parameter; that is, it sets the scale of the transformation by changing the inner product. So it provides flexibility to the researcher, but also suggests the need to find tools to help set this parameter according to the data and the analysis goal. From a neurophysiological perspective, which is particularly important in this work, the kernel size has a biological interpretation. Because the kernel function utilized in the estimation is associated with the filtering of the point process, and the similarity of this step to the spike-to-membrane potential conversion, the kernel size can be interpreted as the time constant of the cell membrane resistive-capacitive network.

8.2 Topics for Future Developments

As said earlier, this dissertation aimed primarily to present the fundamental theory and provide examples for the reproducing kernel Hilbert space (RKHS) framework we propose for processing of point processes. However, there still several topics that need work to further complete this research and better establish the value of the RKHS framework. There are two main topics for future developments:

1. Filtering in the RKHS; and
2. Data efficient CI kernel estimators.

Filtering is the most important signal processing operation for use of the RKHS framework in BMIs, which first motivated this work. As reviewed in Section 2.4.3, the vast majority of current BMIs apply traditional linear/nonlinear filtering methods to binned spike trains but, as shown for the PCA results in Section 7.3.1, the binned cross-correlation does not fully characterize the structure of spike trains. Although conceptually it implements the same idea, the mCI kernel estimator yielded a more consistent outcome and therefore its use in BMIs has the potential to improve current results. These may be improved further by utilizing the nCI kernel. But even if the mCI kernel is used there is a significant improvement over current methodologies as the analysis can be implemented across timescales naturally by incorporating the estimator with multiple kernel sizes. Put differently, the kernel size in the point process kernel estimator can be utilized as a continuous parameter that measures the interactions of the neurons at various timescales.

The difficulties in developing a procedure for filtering in this case are fundamentally the same as with other kernel methods, namely the need for regularization. In this regard, recent developments suggest that this step may avoided explicitly in online implementations if stochastic gradients are utilized (since the gradient regularizes the optimization). Formally, PCA may be utilized since the dimensionally reduction regularizes the Gram matrix.

The second topic is that of developments of the CI point process kernels, and most specifically their estimators. In spite of the successful results using the nCI kernel, this kernel was not truly designed for point processes beyond Poisson and hence its sensitivity is somewhat limited especially as the complexity of the point process model increases. Nevertheless, it is hoped that these results will stem further developments and lead to the design of data efficient CI kernel estimators. A crucial step in deriving an estimator for a CI kernel is the estimation of the conditional intensity function, as can be noticed in Section 3.4. For estimation of the rate function, kernel smoothing can be used quite efficiently. But current methods for estimation of the conditional intensity function are data intensive which prevents a more widespread use of CI kernels capable of coping with general point process models. Therefore, I believe that the solution might involve the use of semi-parametric models of the history evolution in conjunction with the nonlinear kernels ideas to enhance the dimensionality of the point process kernel memory.

APPENDIX A
BRIEF INTRODUCTION TO RKHS THEORY

In this appendix, we briefly review some basic concepts of kernel methods and RKHS theory necessary for the understanding of this dissertation. The presentation here is meant to be as general and introductory as possible, so the notation was purposely chosen to be different from the one used throughout this document.

The fundamental result in RKHS theory is the well-known *Moore-Aronszajn theorem* [Aronszajn, 1950; Moore, 1916]. Let K denote a generic symmetric and positive definite function of two variables defined on some space E. That is, a function $K(\cdot,\cdot) : E \times E \to \mathbb{R}$ which verifies:

(i) Symmetry: $K(x,y) = K(y,x), \quad \forall x, y \in E$.

(ii) Positive definiteness: for any finite number of $l \in \mathbb{N}$ points $x_1, x_2, \ldots, x_l \in E$, and any corresponding l coefficients $c_1, c_2, \ldots, c_l \in \mathbb{R}$,

$$\sum_{m=1}^{l} \sum_{n=1}^{l} c_m c_n K(x_m, x_n) \geq 0. \tag{A–1}$$

These are sometimes called Mercer conditions [Mercer, 1909]. Then the Moore-Aronszajn theorem [Aronszajn, 1950; Moore, 1916] guaranties that there exists a unique Hilbert space $\mathcal{H}$ of real valued functions on E such that, for every $x \in E$,

(i) $K(x,\cdot) \in \mathcal{H}$, and

(ii) for any $f \in \mathcal{H}$

$$f(x) = \langle f(\cdot), K(x,\cdot) \rangle_{\mathcal{H}}. \tag{A–2}$$

The identity on Equation A–2 is called the *reproducing property* of K, and, for this reason, $\mathcal{H}$ is said to be an RKHS with reproducing kernel K.

Two essential corollaries of this theorem can be observed. First, since both $K(x,\cdot)$ and $K(y,\cdot)$ are in $\mathcal{H}$, we get from the reproducing property that

$$K(x,y) = \langle K(x,\cdot), K(y,\cdot) \rangle_{\mathcal{H}}. \tag{A–3}$$

Hence, K evaluates the inner product in this RKHS. This identity is the *kernel trick*, well known in kernel methods, and the main tool for computation in this space. Second, a consequence of the previous properties which can be explicitly seen in the kernel trick is that, given any point $x \in E$, the representer of evaluation in the RKHS is $\Psi_x(\cdot) = K(x, \cdot)$. Notice that the *functional transformation* Ψ from the input space E into the RKHS $\mathcal{H}$ evaluated for a given x, and in general any element of the RKHS, is a real function defined on E.

The seminal work by Parzen [1959] provides a quite interesting perspective to RKHS theory (a review is presented in Wahba [1990, Chapter 1]). In his work, Parzen proved that for *any* symmetric and positive definite function there exists a space of Gaussian distributed random variables defined on the same domain for which this function is the covariance function. Assuming stationarity and ergodicity, this space might just as well be thought of as a space of random processes. In other words, any kernel inducing an RKHS denotes simultaneously an inner product in the RKHS and a covariance operator in another space. Furthermore, it is established that there exists an isometric isomorphism, that is, a one-to-one inner product-preserving mapping, also called a *congruence*, between these two spaces which are thus said to be *congruent*. This is an important result as it sets up a correspondence between the inner product due to a kernel in the RKHS to our intuitive understanding of the covariance function and associated linear statistics. Simply put, due to the congruence between the two spaces an algorithm can be derived and interpreted in any of the spaces.

APPENDIX B
A COMPARISON OF BINLESS SPIKE TRAIN MEASURES

B.1 Introduction

Spike train similarity measures or, conversely, dissimilarity measures are important tools to quantify the relationship among pairs of spike trains. Indeed, the definition of such a measure can be essential for classification, clustering or other forms of spike train analysis. For example, just by using a distance (dissimilarity) measure it is possible to decode the applied stimulus from a spike train [Victor and Purpura, 1996, 1997; Wohlgemuth and Ronacher, 2007]. This is possible because the measure is used to quantify how much the spike train differs from a "template" or sets of reference spike trains for which the input stimulus is known and, hence, classified accordingly (Figure B-1). However, naturally the success of this classification is dependent of the discriminative ability of the measure.

A traditional measure of similarity between two spike trains is to measure the (empirical) cross-correlation of the binned spike trains [Brown et al., 2004]. However, to avoid the difficulties associated with binning and to prevent estimation errors of information when binning is done, binless spike train dissimilarity measures have been proposed. Three well known such measures which we shall consider for comparison are Victor-Purpura's (VP) distance [Victor and Purpura, 1996, 1997][1] , van Rossum's distance [van Rossum, 2001] and the correlation-based measure proposed by Schreiber et al. [2003].

These measures have been utilized in different neurophysiological paradigms (Victor [2005] and references within) and for different tasks, such as classification [Victor and Purpura, 1996, 1997] and clustering of spike trains [Fellous et al., 2004; Paiva et al., 2007;

[1] Actually, in their works, Victor and Purpura [1996, 1997] proposed not one but several spike train distances. Namely, $D^{spike}[q]$, $D^{interval}[q]$, $D^{count}[q]$ and $D^{motif}[q]$. In this study, and as in most references to their works, VP distance refers to $D^{spike}[q]$ for a fair comparison to the other distances considered in this study.

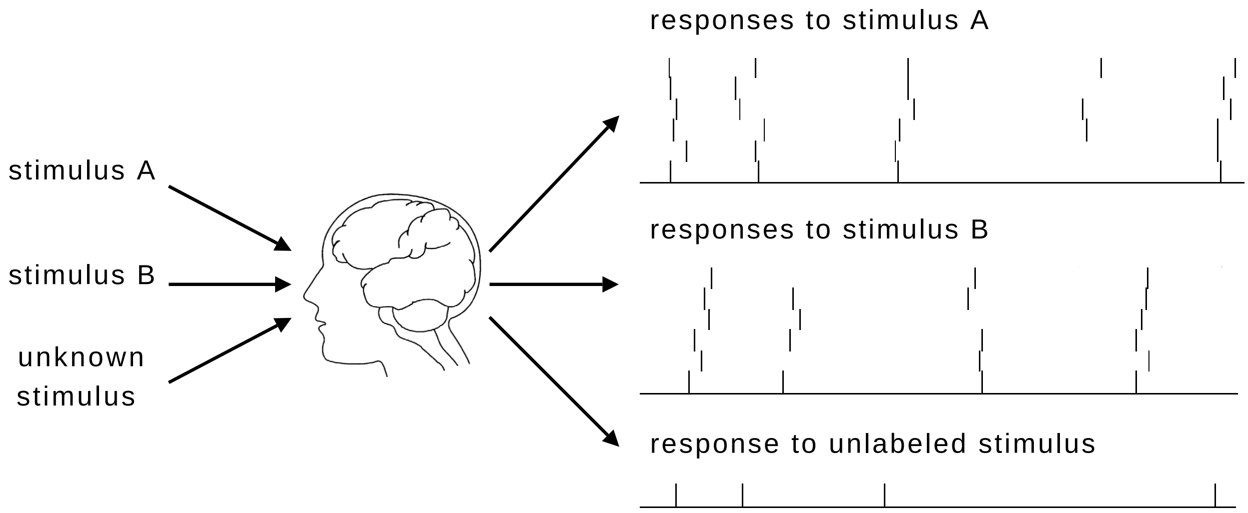


Figure B-1. Typical experimental setup for classification using spike train dissimilarities. In this setup the measure is utilized to quantify the dissimilarity between the new spike train and the reference spike trains for each of the stimulus. Then, the unlabeled stimulus is inferred as the one corresponding to the class for which the new spike train has smaller average dissimilarity.

Toups and Tiesinga, 2006]. However, we feel that in neither of these works was the choice of the measure used have been properly argued versus the candidates. This is perhaps because, to the authors knowledge, a systematic comparison has not yet been attempted in the literature. The work by Kreuz et al. [2007] compares the ISI distance proposed in that paper with several spike train measures, including the ones considered in this work. However, this is done only for synchrony of spike trains generated under a special model with quite strong couplings among neurons. This chapter fills this void by comparing the above mentioned spike train measures in multiple paradigms and under realistic scenarios.

As will be shown from the presentation in Section B.2, each measure implies a given kernel function that measures similarity in terms of a single pair of spike times. Another issue addressed here was to what extent this kernel affects the performance of each measure. Therefore, inspired by the ideas introduced in Chapter 3, the measures are first extended to a set of four kernels and compared for all of these. By evaluating the measures using all of these kernels the comparison is made kernel independent and shows the connection and generality of the principles used in designing the measures.

B.2 Binless Spike Train Dissimilarity Measures

B.2.1 Victor-Purpura's Distance

Historically, Victor-Purpura's (VP) distance [Victor and Purpura, 1996, 1997] was the first binless distance measure proposed in the literature. Two key design considerations in the definition of this distance were that it needed to be sensitive to the absolute spike times and would not correspond to Euclidean distances in a vector space. The first consideration was due to the fact that the distance was initially to be utilized to study temporal coding and its precision in the visual cortex. As stated by the authors, the basic hypothesis is that a neuron is not simply a rate detector but can also function as a coincidence detector. Within this respect the distance is well motivated by neurophysiological ideas. The second consideration is because, in this way it is "not based on assumptions about how responses should be scaled or combined" [Victor and Purpura, 1996].

The VP distance defines the distance between spike trains as the cost in transforming one spike train into the other. Three elementary operations in terms of single spikes are established: moving one spike to perfectly synchronize with the other, deleting a spike, and inserting a spike. Once a sequence of operations is set, the distance is given as the sum of the cost of each operation. The cost in moving a spike at t_m to t_n is $q|t_m - t_n|$, where q is a parameter expressing how costly the operation is. Because a higher q means that the distance increases more when a spike needs to be moved, the distance as a function of q expresses the precision of the spike times. The cost of deleting or inserting a spike is set to one.

Since the transformation cost for the spike trains is not unique, the distance is not yet well defined. Moreover, this criterion needs to guarantee the fundamental axioms of a distance measure for any spike trains s_i, s_j and s_k:

(i) Symmetry: $d(s_i, s_j) = d(s_j, s_i)$
(ii) Positiveness: $d(s_i, s_j) \geq 0$, with equality holding if and only if $s_i = s_j$
(iii) Triangle inequality: $d(s_i, s_j) \leq d(s_i, s_k) + d(s_k, s_j)$.

To ensure the triangle inequality and uniqueness of the distance between any two spike trains, the sequence which yields the *minimum cost* in terms of the operations is used. Therefore, the VP distance between spike trains s_i and s_j is defined as

$$d_{\mathrm{VP}}(s_i, s_j) \triangleq \min_{C(s_i \leftrightarrow s_j)} \sum_{l} K_q\left(t^i_{c_i[l]}, t^j_{c_j[l]}\right), \tag{B–1}$$

where $C(s_i \leftrightarrow s_j)$ is the set of all possible sequences of elementary operations that transform s_i to s_j, or vice-versa, and $c_{(\cdot)}[\cdot] \in C(s_i \leftrightarrow s_j)$. That is, $c_i[l]$ denotes the index of the spike time of s_i manipulated in the lth step of a sequence. $K_q(t^i_{c_i[l]}, t^j_{c_j[l]})$ is the cost associated with the step of mapping the $c_i[l]$th spike of s_i at $t^i_{c_i[l]}$ to $t^j_{c_j[l]}$, corresponding to the $c_j[l]$th spike of s_j, or vice-versa. In other words, K_q is a distance metric between two spikes.

Suppose two spike trains with only one spike each, the mapping between the two spike trains is achieved through the three above mentioned operations and the distance is given by

$$\begin{aligned} K_q(t^i_m, t^j_n) &= \min\left\{q|t^i_m - t^j_n|, 2\right\} \\ &= \begin{cases} q|t^i_m - t^j_n|, & |t^i_m - t^j_n| < 2/q \\ 2, & \text{otherwise.} \end{cases} \end{aligned} \tag{B–2}$$

This means that if the difference between the two spike times is smaller than $2/q$ the cost is linearly proportional to their time difference. However, if the spikes are farther apart it is less costly to simply delete one of the spikes and insert it at the other location. Shown in this way, K_q is nothing but a scaled and inverted triangular kernel applied to the spike times. This perspective of the elementary cost function is key to extend this cost to other kernels, as we will present later.

At first glance it would seem that the computational complexity would be unbearable because the formulation of the algorithm describes the distance in terms of a full search through all allowed sequences of elementary operations. Luckily, efficient dynamic

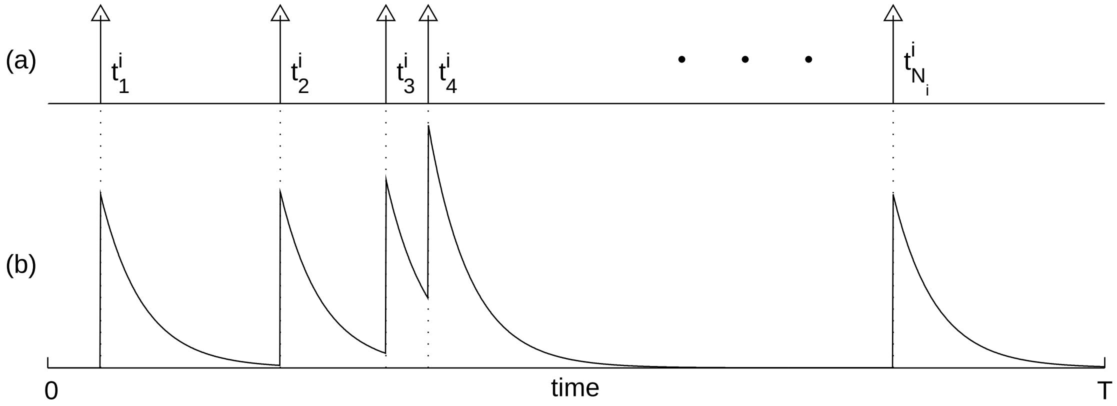


Figure B-2. Spike train and corresponding filtered spike train utilizing a causal exponential function (Equation B–4).

programming algorithms were developed which reduce it to a more manageable level of $\mathcal{O}(N_i N_j)$ [Victor and Purpura, 1996], i.e., the scaled product of the number of spikes in the spike trains whose distance is being computed.

B.2.2 van Rossum's Distance

Similar to the VP distance, the distance proposed by van Rossum [2001] utilizes the full resolution of the spike times. However, the approach taken is conceptually simpler and more intuitive. Simply put, van Rossum's distance [van Rossum, 2001] is the Euclidean distance between the exponentially filtered spike trains.[2]

A spike train s_i defined on the time interval $[0, T]$ and spike times $\{t^i_m : m = 1, \ldots, N_i\}$ can be written as a continuous-time signal as a sum of time-shifted impulses,

$$s_i(t) = \sum_{m=1}^{N_i} \delta(t - t^i_m), \tag{B–3}$$

where N_i is the number of spikes in the recording interval. In this perspective, the filtered spike train is the sum of the time-shifted impulse response of the smoothing filter, $h(t)$,

[2] Filtered spike trains correspond to what is often referred to as "shot noise" in the point processes literature [Papoulis, 1965, Section 16.3].

and can be written as

$$f_i(t) = \sum_{m=1}^{N_i} h(t - t_m^i). \qquad \text{(B–4)}$$

For the smoothing filter, van Rossum [2001] proposed to use a causal decaying exponential function, written mathematically as $h(t) = \exp(-t/\tau)u(t)$, with $u(t)$ being the Heaviside step function (illustrated in Figure B-2). The parameter τ in van Rossum's distance controls the decay rate of the exponential function and, hence, the amount of smoothing that is applied to the spike train. Thus, it determines how much variability in the spike times is allowed and how it is combined into the evaluation of the distance. In essence, τ plays the reciprocal role of the q parameter (Equation B–2) for the VP distance. The choice for the exponential function was due to biological considerations. The idea is that an input spike will evoke a post-synaptic potential at the stimulated neuron which, simplistically, can be approximated through the exponential function [Dayan and Abbott, 2001].

In terms of their filtered counterparts, it is easy to define a distance between the spike trains. An intuitive choice is the usual Euclidean distance, $L^2([0,T])$, between square integrable functions. The distance between spike trains s_i and s_j is therefore defined as

$$d_{\mathrm{vR}}(s_i, s_j) \triangleq \frac{1}{\tau} \int_0^\infty \left[f_i(t) - f_j(t)\right]^2 dt. \qquad \text{(B–5)}$$

van Rossum's distance also seems motivated by the perspective of a neuron as a coincidence detector. This perspective may be induced by the definition. When two spike trains are "close" more of their spikes will be synchronized, which translates into a smaller difference of the filtered spike trains and therefore yields a smaller distance. Despite this formulation, the multi-scale quantification capability of the distance was noticed before by van Rossum [2001]. The behavior transitions smoothly from a count of non-coincidence spikes to a difference in spike count as the kernel size τ is increased. This perspective can be obtained from Equation B–4 if one notices that it corresponds to kernel intensity estimation with function h [Reiss, 1993]. In more broad terms one can thus think of van Rossum's distance as the $L^2([0,\infty))$ distance between the estimated *intensity functions* at time scale τ. Thus, van Rossum's distance can be used to measure the dissimilarity between spike trains at any time scale simply by selecting τ appropriately.

Evaluation of the distance is numerically straightforward, as it directly implements Equation B–5. But explicit computation of the filtered spike trains and integral in a discrete-time simulation is computationally more intensive than evaluating the VP distance which depends only on the number of spikes in the spike trains. Furthermore, the computation burden would increase proportionally to the length of the spike trains and inversely proportional to the simulation step. However, as shown by Paiva et al. and utilized in Paiva et al. [2007], van Rossum's distance can be evaluated in terms of a computationally effective estimator with order $\mathcal{O}(N_i N_j)$, given as

$$d_{\mathrm{vR}}(s_i, s_j) = \frac{1}{2}\left[\sum_{m=1}^{N_i}\sum_{n=1}^{N_i} L_\tau(t_m^i - t_n^i) + \sum_{m=1}^{N_j}\sum_{n=1}^{N_j} L_\tau(t_m^j - t_n^j)\right] + \sum_{m=1}^{N_i}\sum_{n=1}^{N_j} L_\tau(t_m^i - t_n^j), \quad \text{(B–6)}$$

where $L_\tau(\cdot) = \exp(-|\cdot|/\tau)$ is the Laplacian kernel. Thus, this distance can be computed with the same computational complexity as the VP distance.

B.2.3 Schreiber et al. Induced Divergence

The third dissimilarity measure considered in this paper is derived from the correlation-based measure proposed by Schreiber et al. [2003]. Like van Rossum's distance, the correlation measure was also defined in terms of the filtered spike trains. Instead of using the causal exponential function, however, Schreiber and coworkers proposed to utilize the Gaussian kernel. The core idea of this correlation measure is the concept of dot product between the filtered spike trains. Actually, in any space with an inner product two types of quadratic measures are naturally induced: the Euclidean distance, and a correlation coefficient-like measure, due to the Cauchy-Schwarz inequality. The former corresponds to the concept utilized by van Rossum, whereas the latter is conceptually equivalent to the definition proposed by Schreiber and associates. So, in this sense, the two measures are directly related. Nevertheless, it must be emphasized that, like the VP distance, this measure is non-Euclidean since it is an angular metric of filtered spike trains [Paiva et al.].

In defining the measure, write the filtered spike trains as

$$g_i(t) = \sum_{m=1}^{N_i} G_{\sigma/\sqrt{2}}(t - t_m^i), \tag{B–7}$$

where $G_{\sigma/\sqrt{2}}(t) = \exp[-(t)^2/\sigma^2]$ is the Gaussian kernel. Notice the dependence of the filtering on σ which plays in this case the same role as τ in the exponential function in van Rossum's distance, and is inversely related to q in VP distance. Assuming a discrete-time implementation of the measure, then the filtered spike trains can be seen as vectors, for which the usual dot product can be used. Based on this, the Cauchy-Schwarz (CS) inequality guaranties that

$$|\vec{g_i} \cdot \vec{g_j}| \leq \|\vec{g_i}\| \, \|\vec{g_j}\| \,, \tag{B–8}$$

where g_i, g_j are the filtered spike trains in vector notation, and $\vec{g_i} \cdot \vec{g_j}$ and $\|\vec{g_i}\|$, $\|\vec{g_i}\|$ denotes the filtered spike trains dot product and norm, respectively. The norm is given as usual by $\|\vec{g_i}\| = \sqrt{\vec{g_i} \cdot \vec{g_i}}$. Because by construction the filtered spike trains are non-negative functions, the dot product is also non-negative. Consequently, rearranging the Cauchy-Schwarz inequality yields the correlation coefficient-like quantify,

$$r(s_i, s_j) = \frac{\vec{g_i} \cdot \vec{g_j}}{\|\vec{g_i}\| \, \|\vec{g_j}\|}, \tag{B–9}$$

proposed by Schreiber et al. [2003]. Notice that like the absolute value of the correlation coefficient, $0 \leq r(s_i, s_j) \leq 1$. Equation B–9, however, takes the form of a *similarity measure*. Utilizing the upper bound, a dissimilarity can be easily derived,

$$d_{\mathrm{CS}}(s_i, s_j) = 1 - r(s_i, s_j) = 1 - \frac{\vec{g_i} \cdot \vec{g_j}}{\|\vec{g_i}\| \, \|\vec{g_j}\|}. \tag{B–10}$$

In light of the perspective presented here we shall hereafter refer to d_{CS} as the CS dissimilarity measure.

The CS dissimilarity, like the previous two measures, can also be utilized directly to measure dissimilarity in the firing rates of spike trains merely by choosing a large σ. Similar to van Rossum's distance, this is shown explicitly in the formulation of the measure in terms of the inner product of intensity functions, with the time scale specified by σ.

An important difference with regards to the VP and van Rossum's distances needs to be pointed out. d_{CS} is *not* a distance measure. Although it is trivial to prove that it verifies the symmetry and positiveness axioms, the measure does not fulfill the triangle inequality. Nevertheless, since it guaranties the first two axioms it is what is called in the literature a pre-metric [Pekalska and Duin, 2005].

In the definition of the measure and, more importantly, in the utilization of the concept of the dot product the filtered spike trains were considered finite-dimensional vectors [Schreiber et al., 2003]. If this naïve approach is taken, then the computational complexity in evaluating the measure would suffer from the same limitations as the direct implementation of van Rossum's distance. But, like the latter, a data effective method can be obtained in the same way to compute the distance [Paiva et al.],

$$d_{\mathrm{CS}}(s_i, s_j) = 1 - \frac{\sum_{m=1}^{N_i} \sum_{n=1}^{N_j} \exp\left[-\frac{(t_m^i - t_n^j)^2}{2\sigma^2}\right]}{\sqrt{\left(\sum_{m,n=1}^{N_i} \exp\left[-\frac{(t_m^i - t_n^i)^2}{2\sigma^2}\right]\right)\left(\sum_{m,n=1}^{N_j} \exp\left[-\frac{(t_m^j - t_n^j)^2}{2\sigma^2}\right]\right)}}. \tag{B–11}$$

Evaluating the distance using this expression has a computational complexity of order $\mathcal{O}(N_i N_j)$, just like the two previously presented measures.

B.3 Extension of the Measures to Multiple Kernels

From the previous presentation it should be observable that each measure was originally associated with a particular kernel function which measures the similarity between two spike times. Interestingly, the kernel function is found to be different in all three situations. In any case, it is remarkable that the measures are conceptually different irrespective of the differences in the kernel function. To further complete our study we were also interested in verifying the impact of different kernel functions in each measure. In this section we further develop these ideas. In particular, we present the details involved in replacing the default kernel for each dissimilarity measure and, whenever pertinent, intuitively explain how this approach reveals the connections between the measures. It should be remarked that similar considerations have been presented previously by Schrauwen and Campenhout [2007], although under a different analysis paradigm.

In Section B.2.1 the distance between two spikes for the VP distance is defined through the function K_q. This distance represents the minimum cost in transforming a spike into the other in terms of the elementary operations defined by Victor and Purpura. As briefly pointed out, this function is equivalent to having

$$K_q(t_m^i, t_n^j) = 2\left[1 - \kappa_{1/q}(t_m^i - t_n^j)\right], \tag{B–12}$$

where κ_α is the triangular kernel with parameter α,

$$\kappa_\alpha(x) = \begin{cases} 1 - |x|/(2\alpha), & |x| < 2\alpha \\ 0, & |x| \geq 2\alpha, \end{cases} \tag{B–13}$$

which is, in essence, a *similarity* measure of the spike times. Notice that this perspective does not change the non-Euclidean properties of the VP distance since those properties are a result of the condition in Equation B–1. Put in this way, it seems obvious that other kernel functions may be used in place of the triangular kernel, as briefly alluded by Victor and Purpura [1997].

The kernel in the VP distance is not explicit in the definition. Rather, is the cost associated with the three elementary operations. Similarly, in van Rossum's distance and CS dissimilarity measure the perspective of a kernel operating on spike times is not explicit in the definition. The difference however is that the kernel arises naturally as an immediate byproduct of the filtering of the spike trains. This result is noticeable

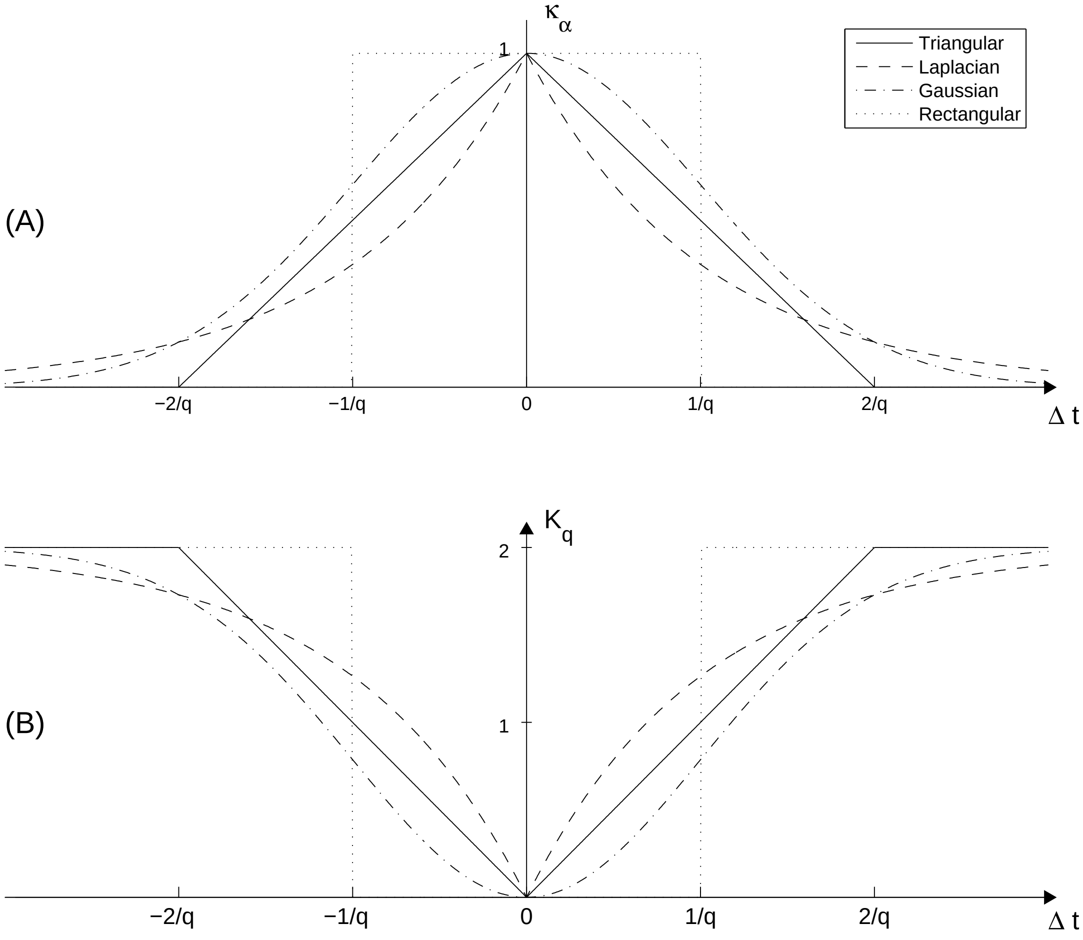


Figure B-3. (a) Kernels utilized in this study and (b) the corresponding K_q function induced by each of the kernels.

in the expressions for computational efficient evaluation given by Equation B–6 and Equation B–11. Again, and just as proposed for the VP distance, alternative kernel functions can be utilized in the evaluation of the dissimilarity measures instead of the proposed kernel by the original construction.

As said earlier, each of the spike train measures considered here was defined with a different kernel function. To provide a systematic comparison, each measure was evaluated with four kernels: the triangular kernel in Equation B–13, and the Laplacian, Gaussian, and rectangular kernels,

$$\text{Laplacian:} \quad \kappa_\tau(x) = \exp\left(-\frac{|x|}{\tau}\right) \tag{B–14}$$

$$\text{Gaussian:} \quad \kappa_\sigma(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \tag{B–15}$$

$$\text{Rectangular:} \quad \kappa_\alpha(x) = \begin{cases} 1, & |x| < \alpha \\ 0, & |x| \geq \alpha, \end{cases} \tag{B–16}$$

For reference, these four kernels and induced distance function K_q in terms of each of the kernels are depicted in Figure B-3. In this way each measure was evaluted for the kernel it was originaly defined and the other kernels for a fair comparison.

Note that if other kernels where to be chosen these would have to be symmetric, maximum at the origin, and always positive, to ensure the symmetry and positiveness of the measure. Additionally, for the VP distance to be well posed, the kernels need to be concave so that the optimization in Equation B–1 garanties the triangle inequality. However, the Gaussian and rectangular kernels are not concave and thus for these kernels the VP measure is a pre-metric. This means that when these kernels are used the resulting dissimilarity is not a well defined distance. Nevertheless, we utilize these kernels here regardless since our aims are to study the effect of the this kernel of the discrimination ability, and also to compare the measures appart this factor.

It is interesting to consider the consequences in terms of the filtered spike trains associated with the choice of each of the four kernels presented. As motivated by van Rossum [2001], the biological inspiration behind the idea in utilizing filtered spike trains is that they can be thought of as post-synaptic potentials evoked at the efferent neuron. In this sense, kernels are mathematical representations of the interactions involved with this idea. As shown before, the Laplacian function results from the autocorrelation of a one-sided exponential function. Likewise, the Gaussian function (with kernel size scaled by $\sqrt{2}$) results from its own autocorrelation. The triangular results from the autocorrelation of the rectangular function. The smoothing function associated with the rectangular

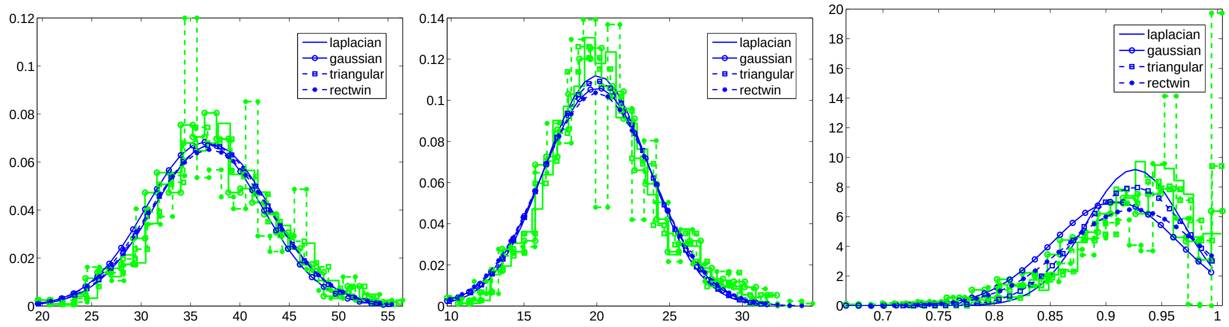


Figure B-4. Estimated pdf of the measures for each kernel considered (green) and corresponding fitted Gaussian pdf (blue). The pdf was estimated by a normalized histogram of the evaluation of the measure with kernel/bin size 2ms for 1000 pairs of uncorrelated spike trains with mean firing rate 20 spk/s and jitter noise of 3ms. (Details can be found in Section B.4.3.)

function corresponds to the inverse of the square root of a sinc function. Based on these observations it seems to us that the Laplacian kernel is, from the four kernels considered, the most biologically plausible.

B.4 Results

In this section results are shown for the three dissimilarity measures introduced in terms of a number of parameters: kernel function, firing rate, kernel size, and, in the last paradigm presented, synchrony and jitter of the absolute spike times.

Three simulation paradigms are studied. In each paradigm we will be interested in verifying how well can the dissimilarity measurements discriminate differences in spike trains with regards to a specific feature. To quantify the discrimination ability of each measure in a *scale-free* manner, the results shall be presented and analyzed in terms of a discriminant index defined as

$$\nu(A,B) = \frac{\bar{d}(A,B) - \bar{d}(A,A)}{\sqrt{\sigma_d^2(A,B) + \sigma_d^2(A,A)}}, \tag{B–17}$$

where $\bar{d}(A,A)$, $\bar{d}(A,B)$ denotes the mean of the dissimilarity measure evaluated between spike trains from the same and different situations, respectively, and $\sigma_d^2(A,A)$, $\sigma_d^2(A,B)$ denotes the corresponding variances. The use of a discriminant index was chosen instead of, for example, ROC plots for ease of display and analysis, and because in this way the conclusions drawn here are classifier-free. $\nu(A, B)$ quantifies how well the outcome of the measure can be used to differentiate the situation A from the situation B. In terms of Figure B-1, think that $\left[\bar{d}(A, A), \sigma_d^2(A, A)\right]$ characterizes the distribution of the dissimilarity measure evaluation for spike trains in response to stimulus A, and $\left[\bar{d}(A, B), \sigma_d^2(A, B)\right]$ characterizes a similar distribution but in which the dissimilarities are evaluated between a spike train evoked by stimulus A and a spike train evoked by stimulus B. This is supported by the fact that the distribution of the evaluation of the measures can be reasonably fitted to a Gaussion pdf (Figure B-4). Therefore, the discriminant index is utilized in the simulated experimental paradigms to compare how well the dissimilarity distinguishes spike trains generated under the same versus different conditions, with regards to a parameter specifying how different spike trains from different stimulus are. The discriminant index ν is conceptually similar to that of the Fisher linear discriminant cost [Duda et al., 2000]. A key difference however is that the absolute value is not used. This is because negative values of the index correspond to unreasonable behavior of the measure; that is, the dissimilarity measure yields smaller values between spike trains generated under difference conditions than spike trains generated for the same condition. Obviously, intuitively the desired behavior is that the dissimilarity measure yields a minimum for spike trains generated similarly.

For contrast to the binless dissimilarity measures considered, results are also presented for a *binned* cross-correlation based dissimilarity measure, denoted d_{CC}. This measure is defined just like the CS dissimilarity through Equation B–10. The difference is that now $\vec{g}_i$ and $\vec{g}_j$ are finite dimensional vectors corresponding to the binned spike trains and, thus, $\vec{g}_i \cdot \vec{g}_j$ is the usual Euclidean dot product between two vectors. Notice that d_{CC} is in essence equivalent to quantize the spike times (with quantization step equal to the bin size) and evaluating d_{CS} using the rectangular kernel, with kernel size equal to half the bin size. Hence, d_{CC} can be alternatively computed utilizing Equation B–11. The former

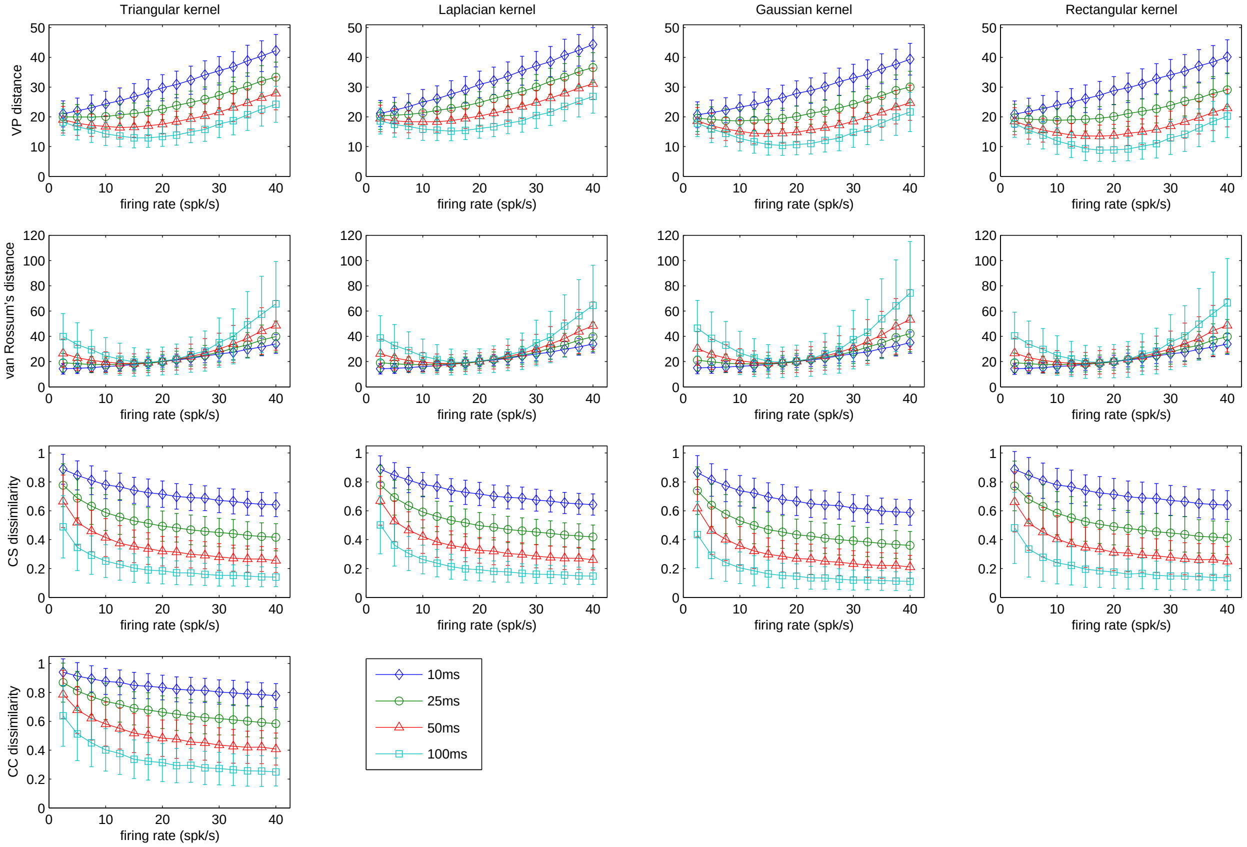


Figure B-5. Value of the dissimilarity measures for each kernel considered as a function of the modulating spike train firing rate. All dissimilarity evaluation are with regards to an homogeneous spike train with average rate 20 spk/s. For each measure and kernel, results are given for four different kernel sizes (shown in the legend) in terms of the measure average value plus or minus one standard deviation. The statistics of the measures were estimated over 1000 randomly generated pairs of spike trains.

approach is more advantageous for large bin size whereas the latter is computationally more effective for smaller bin size (larger number of bins).

B.4.1 Discrimination of Difference in Firing Rate

The first paradigm considered was intended to analyze the characteristics of each measure with regards to the firing rate of one spike train relatively to another of fixed firing rate. The key point was to understand if the measures could be used to differentiate two spike trains of different firing rates. This is important because neurons have been found to often encode information in the spike train firing rates [Adrian, 1928; Dayan

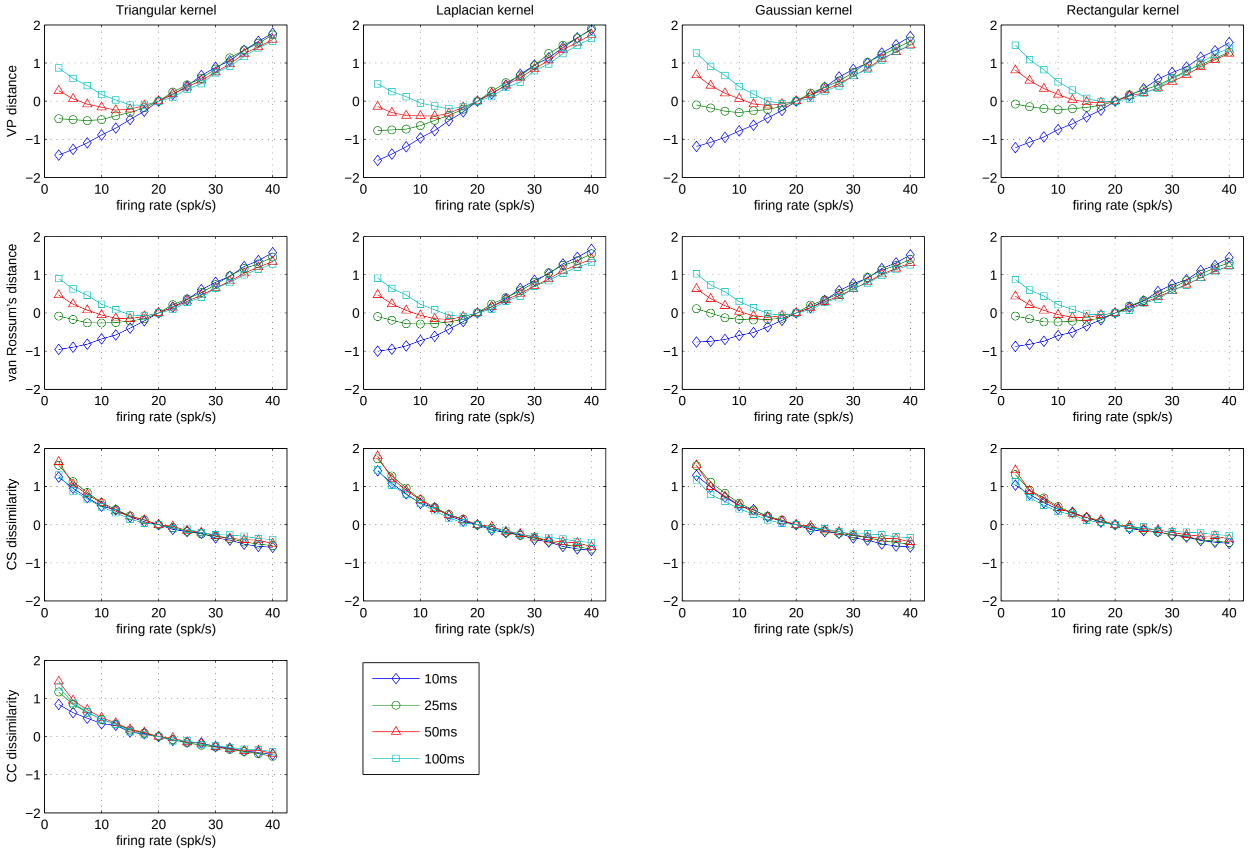


Figure B-6. Discriminant index of the dissimilarity measures for each kernel as a function of the modulating spike train firing rate. See the results in Figure B-5 for reference. The different curves are for different kernel sizes (shown in the legend).

and Abbott, 2001; Rieke et al., 1999]. To simplify matters, all spike trains were simulated as one second long homogeneous Poisson processes. Although this simplification is unrealistic, it allows a first analysis without the introduction of additional effects due to modulation of firing rates in the spike trains. The scenario where the firing rates are modulated over time is considered in the next section. Another important factor in the analysis is the spike train length. Naturally, in this scenario, the discrimination of the measures is expected to improve as the spike train length is increased since more information is available. In practice however this value is often smaller than one second. Thus, the value was chosen as a compromise between a reasonable value for actual data analysis and good statistical illustration of the properties of each measure.

In our study, simulations were made for each dissimilarity measure utilizing each of the four described kernels. In each case, the analysis was repeated for four kernel sizes, 10, 25, 50 and 100 milliseconds. The kernel sizes used were purposely chosen relatively large since firing rate information can only be extracted at a slower time scale. The results are shown in Figure B-5 in terms of mean values plus or minus one standard deviation, as estimated from 1000 randomly generated spike train pairs. For each pair, one of the spike trains was generated at a reference firing rate of 20 spk/s, whereas the firing rate of the other was one of 2.5 to 40 spk/s, in steps of 2.5 spk/s.

Utilizing the estimated statistics, the discrimination provided by the measures was evaluated in terms of the discrimination index ν (Equation B–17) with regards to the results when both spike trains have firing rate 20 spk/s. The results are shown in Figure B-6. The results for VP and van Rossum's distances reflect the importance of the choice of time scale, materialized in the form of the kernel size selection. Only for the largest kernel size (100ms) did these two distances behave as we intuitively expected. This is not surprising since discrimination can only occur if the dissimilarity can incorporate an estimation of the firing rate in its evaluation. Even for this kernel size the discriminant index curve shows a small bias towards smaller firing rates. This is natural since the optimal kernel size is infinity, and smaller kernel size tends to result in bias related to the total number of spikes. The discrimination behavior of the CS dissimilarity however seems nearly insensitive to the choice of the kernel size. On the other hand, when the firing rate is above the reference the outcome is not the desired. For lower firing rates, the positive discrimination index is due to the presence of the norm of the spike train in the denominator of the definition. One of the most remarkable observations is the consistency of the results for each measure throughout the four kernels. Although there are subtle differences in values they seem to be of importance only for small kernel sizes for which, as pointed out, the results are not significant anyway. Comparing with the results for the CC dissimilarity we verify the resemblance with the CS dissimilarity. Like the latter, the CC

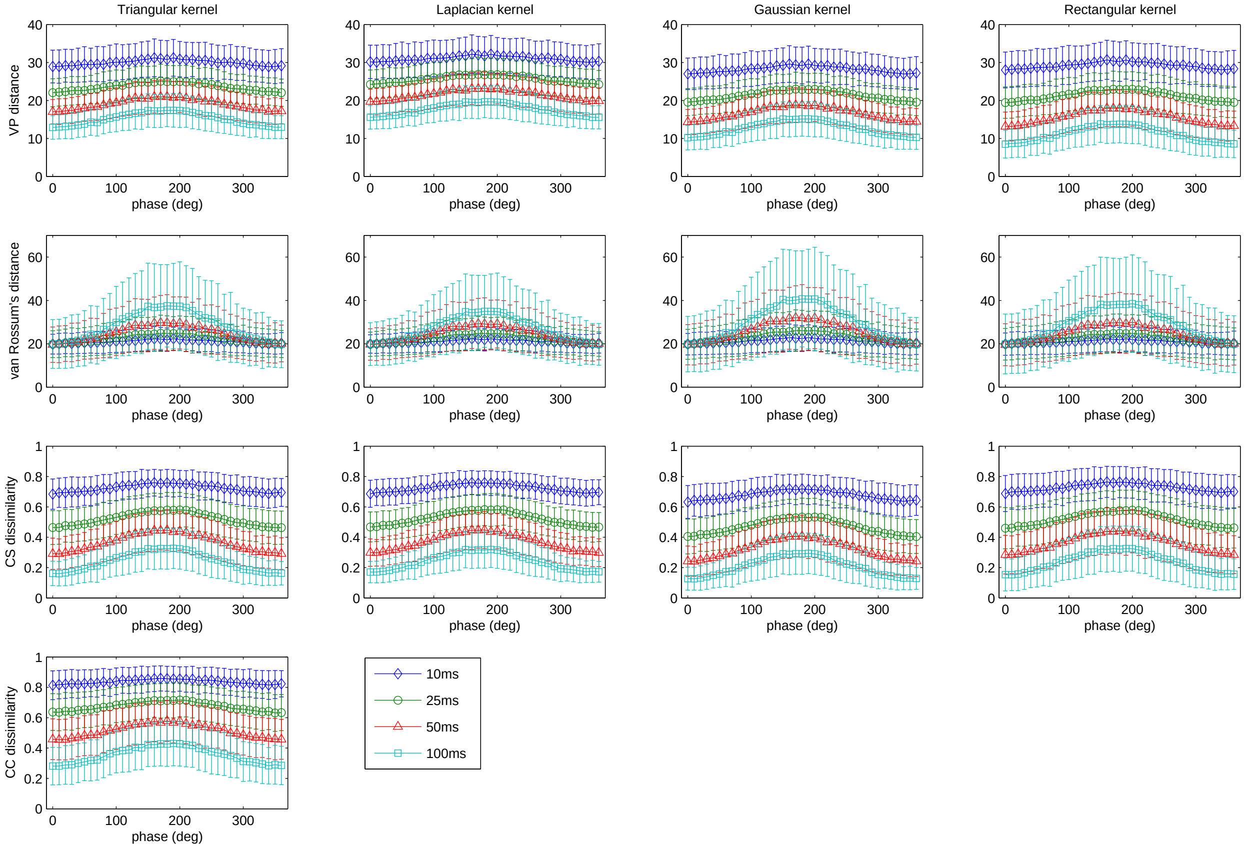


Figure B-7. Value of the dissimilarity measures for each kernel in terms of the phase difference of the firing rate modulation. Like in the previous paradigm, results are shown for each measure, kernel, and four different kernel sizes (shown in the legend) in terms of the measure average value plus or minus one standard deviation. The statistics were estimated over 1000 randomly generated pairs of spike trains.

dissimilarity also is unable to correctly distinguish increases in firing rate of one spike train with respect to the other.

B.4.2 Discrimination of Phase in Firing Rate Modulation

The scenario depicted in the previous paradigm is obviously simplistic. In this case study, an alternative situation is considered in which spike trains must be discriminated through differences in their *instantaneous* firing rates. Spike trains were generated as one second long inhomogeneous Poisson processes with instantaneous firing rate given by sinusoidal waveforms of mean 20 spk/s, amplitude 10 spk/s and frequency 1Hz. A pair of spike trains was generated at a time and the phase difference of the sinusoidal waveforms

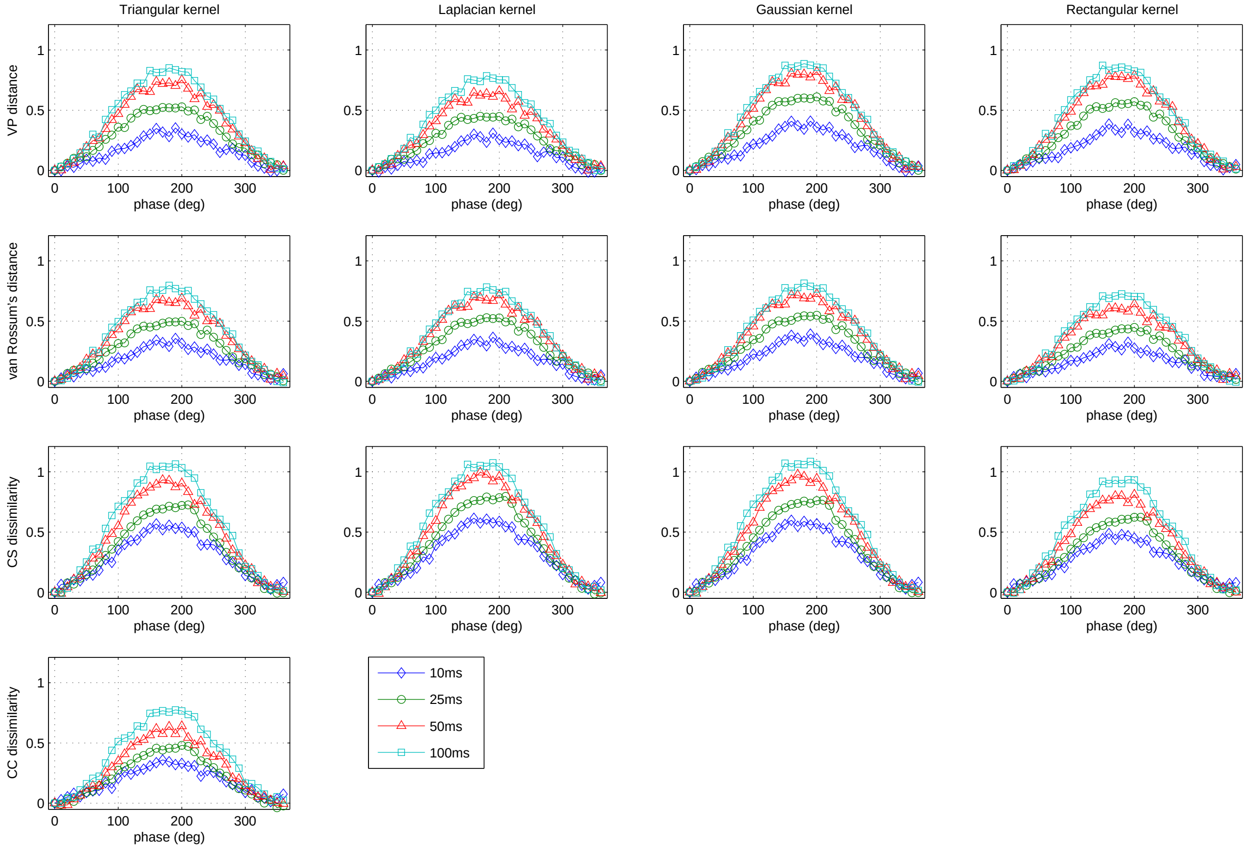


Figure B-8. Discriminant index of the dissimilarity measures for each kernel in terms of the phase of the firing rate modulation as given by Figure B-7. The different curves are for different kernel sizes (shown in the legend).

used to modulate the firing rate of each spike train varied from 0 to 360 degrees. The goal was to verify if the measures were sensitive to instantaneous differences in the firing rate as characterized by the modulation phase difference. This too is a simplification of what is often found in practice where firing rates change abruptly and in an non-period manner. Nevertheless, the paradigm aims at representing a general situation while simultaneously being restricted to allow for a tractable analysis. Obviously, the results are somewhat dependent on our choice of simulation parameters. For example, lower mean firing rates would mean that the dissimilarity measures would be less reliable and, hence, have higher variance. This could be partially compensated by increasing the spike train length. However, the above values are an attempt to approximate real data.

The simulation protocol is similar to that of the case analyzed in the previous section. For each phase difference, we randomly generated 1000 spike train pairs such that the firing rate modulation of the two spike trains differed by the phase difference and applied the dissimilarity measures using each of the four described kernels. As before, the analysis was repeated for four kernel sizes, 10, 25, 50 and 100 milliseconds. Again, the kernel sizes used were chosen large since firing rate information can only be extracted at a slower time scale. The statistics of the dissimilarity measures are shown in Figure B-7.

The analysis of these results with the discrimination index ν with respect to the statistics of each measure at zero phase is depicted in Figure B-8. In this paradigm, the maximum value of the measures was desired to occur at 180°, with a monotonically increasing behavior for phase differences smaller and monotonically decreasing for phase differences greater. As Figure B-8 shows, all measures performed satisfactorily using any of the four kernels and at any kernel size. The CS dissimilarity has the best discrimination with the discrimination index reaching 0.8, compared to a maximum value of 0.65 for the second best. On the other end, overall the CC-based dissimilarity performed the worse. Comparing with the CS dissimilarity (which differs only because the spike times are not quantized) we verify once again the disadvantages of doing binning. With regards to the effect of each kernel, the Gaussian kernel consistently yields the best discrimination for the same kernel size. Conversely, the Laplacian and rectangular kernels seem to perform the worst, although this observation is largely measure dependent. As expected, and similarly to the previous paradigm, the best discrimination is obtained for the largest kernel size since it yields a better estimation of the intensity function. It is noteworthy however that in this paradigm the kernel size cannot be chosen too large, otherwise the intensity function would be over smoothed, thus reducing the differentiation between phases and decreasing the discrimination performance. This phenomenon was observed when we attempted a kernel size of 250ms (not shown).

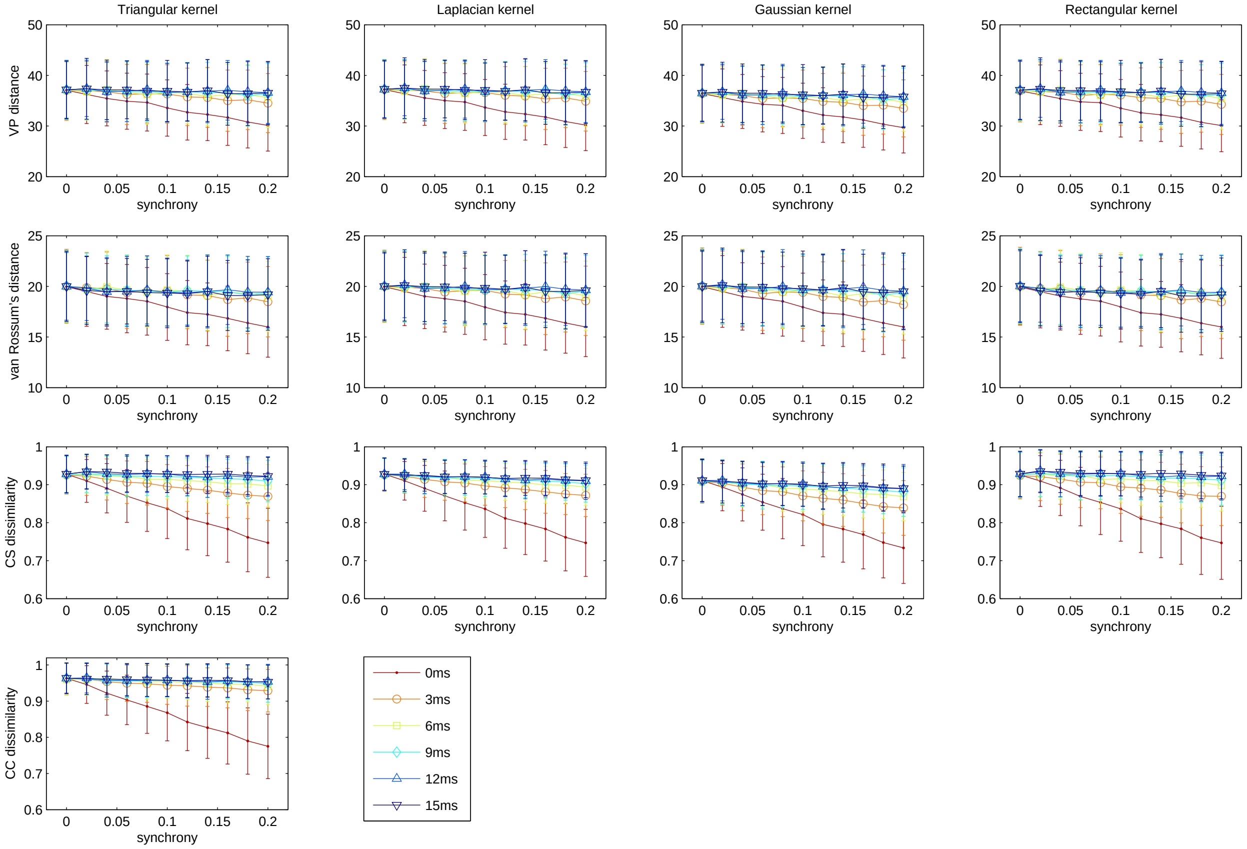


Figure B-9. Value of the dissimilarity measure for each kernel as a function of the synchrony among spike trains. The statistics were estimated over 1000 randomly generated pairs of spike trains simulated with MIP model and average firing rate 20 spk/s. The kernel size was 2ms. The different curves show result under different levels of jitter standard deviation, with the value in the legend.

B.4.3 Discrimination of Synchronous Firings

In this scenario we consider that spike trains are to be differentiated based on the synchrony of neuron firings. More precisely, spike trains are deemed distant (or dissimilar) with regards to the relative number of synchronous spikes. That is, dissimilarity measures are expected to be inversely proportional to the probability of a spike co-occur with a spike in another spike train. This means that, unlike the previous two case studies where differences in firing rate were analyzed, this case puts the emphasis of analysis in the role of each spike. Thus, since the time scale of analysis is much more fine, the precision of a spike time has increased relevance.

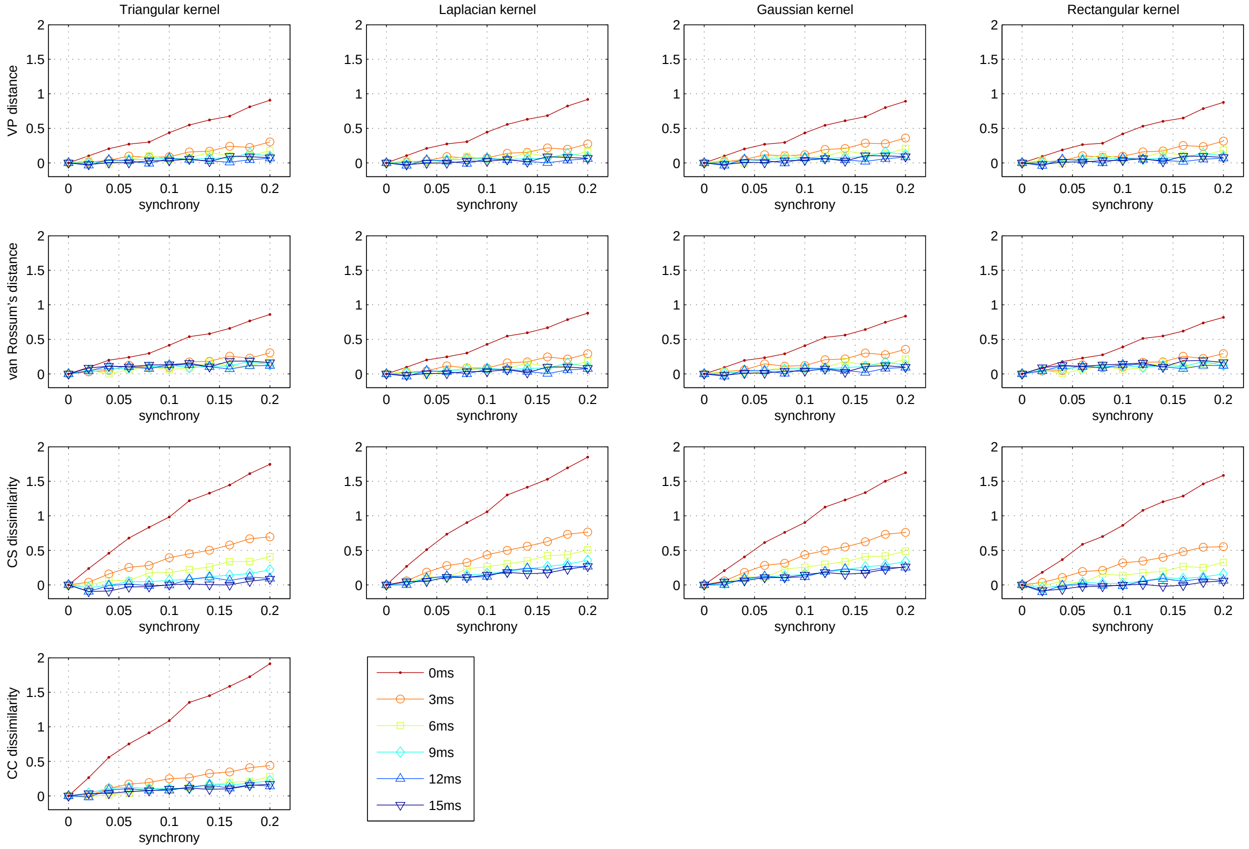


Figure B-10. Discriminant index of the dissimilarity measures for each kernel in terms of the synchrony between the spike trains as given by Figure B-9. The different curves are for different standard deviations (shown in the legend) of the jitter added to the synchronous spikes.

To generate spike trains with a given synchrony the multiple interaction process (MIP) model was used [Kuhn et al., 2003, 2002]. In the MIP model a reference spike train is first generated as a realization of a Poisson process. The spike trains are then derived from this one by copying spikes with probability ε. The operation is performed independently for each spike and for each spike train. Put differently, ε is the probability of a spike co-occurring in another spike train, and therefore controls what we refer to as synchrony. It can also be shown that ε is the count correlation coefficient [Kuhn et al., 2003]. The resulting spike trains are Poisson processes. By generating the reference spike train with firing rate $\varepsilon\lambda$ it is ensured that the derived spikes trains have firing rate λ. To make the simulation more realistic, jitter noise was added to each spike time to recreate the variability in spike times often encountered in practice, thus making the task of finding spikes that are synchronous more challenging. Jitter noise was generated as independent and identically distributed zero-mean Gaussian noise.

For each combination of synchrony and jitter standard deviation, 1000 spike trains pairs were generated, and the dissimilarity measures in terms of the four different kernels were evaluated. All spike trains were one second long and the firing rate 20 spk/s, for similar reasons as in the previous paradigms. The kernel size for the results shown was 2ms. The kernel size was chosen small since in this scenario the characterizing feature is synchronous firings. The results are shown in Figure B-9, and in terms of the discrimination index ν in Figure B-10.

From Figure B-10, the CS and CC dissimilarities have notably better discrimination ability than VP and van Rossum's distance. The results also reveal that the CS dissimilarity is more consistent than the CC dissimilarity since its discrimination decreases in a more graded manner with the presence of variability in the synchronous spike times (even for the same kernel function). The VP and van Rossum's distances have comparable discrimination ability. Comparing the measurements in terms of the kernel functions, it was found that the Laplacian kernel provides the best results, followed by the triangular kernel. Nevertheless, the advantage between different kernels is small.

B.5 Final Remarks

We compared binless spike train measures presented in the literature for their discrimination ability. Given the wide use of these measures in spike trains analysis, classification and clustering, we believe this study is fundamental for understanding the behavior of each measure and deciding which might be more appropriate taking the intended aim into consideration.

Nevertheless, the aim was not just to directly compare the published measures. Here, we extended these measures and provided a broader perspective which, in our opinion, was lacking in the previous presentations. In the review of the measures, it was shown that the measures can be reformulated in terms of elementary kernels on differences of single pairs of spike times. Hence, this kernel can be replaced by any other function able to similarly capture the "closeness" of the spike times. This point of view is important in showing the generality and independence of the measures with regards to the kernel. Moreover, it allows for the comparisons to be done without kernel specific effects. Another, more important perspective presented was that any of the measures considered is a multi-scale quantifier of dissimilarity between spike trains, with scale controlled by the kernel size. This is because, as explicitly verified for the van Rossum's distance and CS dissimilarity, the measures implicitly do intensity function estimation. This observation is key for the understanding of how and why the measures can be utilized to quantify dissimilarity in instantaneous firing rates, despite their formulation aimed at spike timing-based paradigms.

The measures were compared in three experiments with the information for discrimination contained in average firing rates, instantaneous firing rates and synchrony. These were selected to illustrate the concepts discussed and because they were thought to represent the hypothesis to be tested with data analysis. Of course, the simulated paradigms are simplified approximations of the more complex scenarios that may be observed in practice.

Unfortunately, the results reveals that no single measure performs the best or consistently throughout all three paradigms. For instance, if the VP and van Rossum's distances have consistent discrimination in the constant firing rate paradigm they are clearly outperformed in the synchrony-based discrimination task by the CS and CC dissimilarities, but the results of these latter ones are not at all usable in the first paradigm, mostly because their unstability for small number of spikes. Nevertheless, all measures are consistent and comparably perform in the second paradigm, in terms of modulation of the instantaneous firing rates. An intriguing but not entirely surprising result is that, although the VP distance and van Rossum's distance yields quite different results as noticed clearly in Figure B-5 and Figure B-7, their discrimination is the same in all paradigms (Figure B-6, Figure B-8 and Figure B-10).

The results also suggest that the dependence of the measures on a specific kernel is minor. A considerably more relevant issue is the kernel size, as emphasized in the firing rate paradigms. This is because, as mentioned, the measures quantify relations in terms of (implicit) intensity functions. Hence, if the kernel size is not properly selected the estimate of the intensity functions does not account for the desired feature in the spike trains.

Finally, the results depict the importance of *binless* spike train measures. As stated earlier, the only difference between the CS dissimilarity evaluated with the rectangular kernel and the CC dissimilarity is the time quantization incurred with binning. Comparing the results in these two situations in Figure B-8 and Figure B-10 shows that small improvements in discrimination and robustness to jitter noise were achieved in the first and second cases, respectively, by utilizing the spike times directly.

REFERENCES

O. O. Abbé. Über blutkörper-zahlung. *Jena Z. Med. Naturwiss.*, 13:98–105, 1879.

E. D. Adrian. *The Basis of Sensation: the action of the sense organs.* W. W. Norton & Co., New York, 1928.

A. M. Aertsen, G. L. Gerstein, M. K. Habib, and G. Palm. Dynamics of neuronal firing correlation: modulation of "effective connectivity". *Journal of Neurophysiology*, 61(5): 900–917, 1989.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, May 1950.

L. A. Baccalá and K. Sameshima. Directed coherence: a tool for exploring functional interactions among brain structures. In M. A. L. Nicolelis, editor, *Methods for Neural Ensemble Recordings*, pages 179–192. CRC Press, 1999.

M. S. Bartlett. The spectral analysis of point processes. *J. R. Stat. Soc. B*, 25(2):264–296, 1963.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, Nov. 1995.

C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions.* Springer-Verlag, New York, NY, 1984.

D. R. Brillinger. Estimation of the second-order intensities of a bivariate stationary point process. *Journal of the Royal Statistical Society – Series B*, 38:60–66, 1976.

D. R. Brillinger. Nerve cell spike train data analysis: a progression of technique. *Journal of the American Statistical Association*, 87(418):260–271, June 1992.

E. N. Brown, R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14 (2):325–346, 2001a. doi: 10.1162/08997660252741149.

E. N. Brown, D. P. Nguyen, L. M. Frank, M. A. Wilson, and V. Solo. An anlysis of neural receptive field plasticity by point process adaptive filtering. *Proceedings of the National Academy of Science, USA*, 98:12261–12266, 2001b.

E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature Neuroscience*, 7:456–461, 2004. doi: 10.1038/nn1228.

J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biology*, 1(2), Nov. 2003. doi: 10.1371/journal.pbio.0000042.

C. E. Carr and M. Konishi. A circuit for detection of interaural time differences in the brain stem of the barn owl. *Journal of Neuroscience*, 10(10):3227–3246, 1990.

J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature Neuroscience*, 2(7):664–670, July 1999.

Z. Chi and D. Margoliash. Temporal precision and temporal drift in brain and behavior of zebra finch song. *Neuron*, 32(1–20):899–910, Dec. 2001.

Z. Chi, W. Wu, Z. Haga, N. G. Hatsopoulos, and D. Margoliash. Template-based spike pattern identification with linear convolution and dynamic time warping. *Journal of Neurophysiology*, 97(2):1221–1235, Feb. 2007. doi: 10.1152/jn.00448.2006.

D. R. Cox. Some statistical methods connected with series of events. *J. R. Stat. Soc. B*, 17:129–164, 1955.

D. R. Cox and V. Isham. *Point Processes.* Chapman and Hall, 1980.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes.* Springer-Verlag, New York, NY, 1988.

S. Darmanjian, A. R. C. Paiva, and J. C. Príncipe. Hierarchal decomposition of neural data using boosted mixtures of Hidden Markov Chains and its application to a BMI. In *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN-2007*, Orlando, FL, USA, Aug. 2007.

P. Dayan and L. F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems.* MIT Press, Cambridge, MA, USA, 2001.

P. Diggle and J. S. Marron. Equivalence of smoothing parameter selectors in density and intensity estimation. *Journal of the American Statistical Association*, 83(403):793–800, Sept. 1988.

J. P. Donoghue and S. P. Wise. The motor cortex of the rat: cytoarchitecture and microstimulation mapping. *Journal of Comparative Neurology*, 212(12):76–88, 1982.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* Wiley Interscience, 2nd edition, 2000.

U. R. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown. Dynamic analysis of neural encoding by point process adaptive filtering. *Neurocomputing*, 16(5):971–998, May 2004.

J. J. Eggermont. Properties of correlated neural activity clusters in cat auditory cortex resemble those of neural assemblies. *Journal of Neurophysiology*, 96(2):746–764, July 2006. doi: 10.1152/jn.00059.2006.

A. K. Erlang. The theory of probabilities and telephone conversations. *Nyt Tidsskrift for Matematik B*, 20, 1909. Reprinted at http://oldwww.com.dtu.dk/teletraffic/Elang.html.

A. K. Erlang. Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. *Elektroteknikeren*, 13, 1917. Reprinted at http://oldwww.com.dtu.dk/teletraffic/Elang.html.

J.-M. Fellous, P. H. E. Tiesinga, P. J. Thomas, and T. J. Sejnowski. Discovering spike patterns in neuronal responses. *Journal of Neuroscience*, 24(12):2989–3001, Mar. 2004. doi: 10.1523/JNEUROSCI.4649-03.2004.

W. A. Freiwald, A. K. Kreiter, and W. Singer. Synchronization and assembly formation in the visual cortex. In M. A. L. Nicolelis, editor, *Progress in Brain Research*, pages 111–140. Elsevier Science, 2001.

A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2:1527–1537, Nov. 1982.

A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. coding of the direction of movement by a neuronal population. *Journal of Neuroscience*, 8(8):2928–2937, Aug. 1988.

G. L. Gerstein and A. M. Aertsen. Representation of cooperative firing activity among simultaneously recorded neurons. *Journal of Neurophysiology*, 54(6):1513–1528, 1985.

G. L. Gerstein and D. H. Perkel. Simultaneously recorded trains of action potentials: Analysis and functional interpretation. *Science*, 164(3881):828–830, May 1969. doi: 10.1126/science.164.3881.828.

G. L. Gerstein and D. H. Perkel. Mutual temporal relationships among neuronal spike trains. statistical techniques for display and analysis. *Biophysical Journal*, 12(5): 453–473, May 1972.

G. L. Gerstein, D. H. Perkel, and J. E. Dayhoff. Cooperative firing activity in simultaneously recorded populations of neurons: detection and measurement. *Journal of Neuroscience*, 5(4):881–889, 1985.

W. Gerstner and W. Kistler. *Spiking Neuron Models*. MIT Press, 2002.

J. Graunt. Observation of the London Bills of Mortality. http://www.ac.wwu.edu/~stephan/Graunt/graunt.html, 1662.

M. Greenwood and G. U. Yule. An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. *J. R. Statist. Soc. A*, 83:255–279, 1920.

S. Grün, M. Diesmann, and A. Aertsen. Unitary Events in multiple single-neuron activity. I. detection and significance. *Neural Computation*, 14(1):43–80, 2002a.

S. Grün, M. Diesmann, and A. Aertsen. Unitary Events in multiple single-neuron activity. II. nonstationary data. *Neural Computation*, 14(1):43–80, 2002b.

R. H. R. Hahnloser. Cross-intensity functions and the estimate of spike-time jitter. *Biological Cybernetics*, 96(5):497–506, May 2007. doi: 10.1007/s00422-007-0143-7.

D. A. Harville. *Matrix algebra from a statistician's perspective.* Springer, 1997.

N. G. Hatsopoulos, C. L. Ojakangas, L. Paninski, and J. P. Donoghue. Information about movement direction obtained from synchronous activity of motor cortical neurons. *Proceedings of the National Academy of Science, USA*, 95(26):15706–15711, Dec. 1998.

S. Haykin. *Adaptive Filter Processing.* Prentice-Hall, 4nd edition, 2002.

J. M. Hurtado, L. L. Rubchinsky, and K. A. Sigvardt. Statistical method for detection of phase-locking episodes in neural oscillations. *Journal of Neurophysiology*, 91(4): 1883–1898, Apr. 2004.

E. M. Izhikevich. Polychronization: Computation with spikes. *Neural Computation*, 18(2): 245–282, Feb. 2006.

R. E. Kass and V. Ventura. A spike-train probability model. *Neural Computation*, 13(8): 1713–1720, Aug. 2001.

R. E. Kass, V. Ventura, and C. Cai. Statistical smoothing of neuronal data. *Network: Computation in Neural Systems*, 14:5–15, 2003.

A. Y. Khinchin. *Mathematical methods in the theory of queueing.* Griffin, London, 1960. English translation of the Russian original, originally published in 1955.

S.-P. Kim. *Design and analysis of optimal decoding models for Brain-Machine Interfaces.* PhD thesis, University of Florida, May 2005.

S. P. Kim, J. C. Sanchez, Y. N. Rao, D. Erdogmus, J. M. Carmena, M. A. Lebedev, M. A. L. Nicolelis, and J. C. Principe. A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces. *Journal of Neural Engineering*, 3(2): 145–161, 2006. doi: 10.1088/1741-2560/3/2/009.

T. Kreuz, J. S. Haas, A. Morelli, H. D. I. Abarbanel, and A. Politi. Measuring spike train synchrony. *Journal of Neuroscience methods*, 165(1):151–161, Sept. 2007. doi: 10.1016/j.jneumeth.2007.05.031.

A. Kuhn, S. Rotter, and A. Aertsen. Correlated input spike trains and their effects on the response of the leaky integrate-and-fire neuron. *Neurocomputing*, 44–46:121–126, June 2002. doi: 10.1016/S0925-2312(02)00372-7.

A. Kuhn, A. Aertsen, and S. Rotter. Higher-order statistics of input ensembles and the response of simple model neurons. *Neural Computation*, 15(1):67–101, 2003.

B. G. Lindsey and G. L. Gerstein. Two enhancements of the gravity algorithm for multiple spike train analysis. *Journal of Neuroscience methods*, 150(1):116–127, 2006. doi: 10.1016/j.jneumeth.2005.06.019.

W. Maass and C. M. Bishop, editors. *Pulsed Neural Networks*. MIT Press, 1998.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002. doi: 10.1162/089976602760407955.

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–1506, 1995. doi: 10.1126/science.7770778.

K. V. Mardia and P. E. Jupp. *Directional Statistics*. John Wiley & Sons, West Sussex, England, 2000. ISBN 0-471-95333-4.

V. Z. Marmarelis. *Nonlinear Dynamic Modelling of Physiological Systems*. IEEE Press Series in Biomedical Engineering. John Wiley & Sons, 2004.

V. Z. Marmarelis. Identification of nonlinear biological systems using Laguerre expansions of kernels. *Annals of Biomedical Engineering*, 21:573–589, 1993.

J. W. McClurkin, T. J. Gawne, L. M. Optican, and B. J. Richmond. Lateral geniculate neurons in behaving primates. II. Encoding of visual information in the temporal shape of the response. *Journal of Neurophysiology*, 66(3):794–808, Sept. 1991.

J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London – Series A*, 209:415–446, 1909.

R. E. Mirollo and S. H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, Dec. 1990.

E. H. Moore. On properly positive Hermitian matrices. *Bulletin of the American Mathematical Society*, 23:59, 1916.

J. E. Moyal. The general theory of stochastic population processes. *Acta Mathematica*, 108 (1):1–31, dec 1962.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

M. A. L. Nicolelis. Brain-machine interfaces to restore motor function and probe neural circuits. *Nature Reviews Neuroscience*, 4:417–422, 2003. doi: 10.1038/nrn1105.

A. R. C. Paiva, I. Park, and J. C. Príncipe. A reproducing kernel Hilbert space framework for spike trains. Resubmitted.

A. R. C. Paiva, J.-W. Xu, and J. C. Príncipe. Kernel principal components are maximum entropy projections. In *Proceedings of the International Conference on Independent Component Analysis and Blind Source Separation, ICA-2006*, pages 846–853, Charleston, SC, Mar. 2006. doi: 10.1007/11679363_105.

A. R. C. Paiva, S. Rao, I. Park, and J. C. Príncipe. Spectral clustering of synchronous spike trains. In *Proceedings of the IEEE International Joint Conference on Neural Networks, IJCNN-2007*, Orlando, FL, USA, Aug. 2007.

A. R. C. Paiva, I. Park, and J. C. Príncipe. Reproducing kernel hilbert spaces for spike train analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-2008*, Las Vegas, NV, USA, Apr. 2008.

C. Palm. *Intensity variations in telephone traffic*. North Holland, Amsterdam, 1988. English translation of the author's thesis, originally presented in 1943.

A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, New York, 1965.

I. Park, A. R. C. Paiva, T. B. DeMarse, and J. C. Príncipe. An efficient algorithm for continuous-time cross correlation of spike trains. *Journal of Neuroscience methods*, 128 (2):514–523, Mar. 2008. doi: 10.1016/j.jneumeth.2007.10.005.

E. Parzen. *Time Series Analysis Papers*. Holden-Day, San Francisco, CA, 1967.

E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33(2):1065–1076, Sept. 1962.

E. Parzen. Statistical inference on time series by Hilbert space methods. Technical Report 23, Applied Mathematics and Statistics Laboratory, Stanford University, Stanford, California, Jan. 1959.

E. Pekalska and R. P. W. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations And Applications*. World Scientific, 2005. ISBN 9-812-56530-3.

D. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. I. the single spike train. *Biophysical Journal*, 7(4):391–418, July 1967a.

D. H. Perkel, G. L. Gerstein, and G. P. Moore. Neuronal spike trains and stochastic point processes. II. simultaneous spike trains. *Biophysical Journal*, 7(4):419–440, July 1967b.

J. C. Príncipe, D. Xu, and J. W. Fisher. Information theoretic learning. In S. Haykin, editor, *Unsupervised Adaptive Filtering*, volume 2, pages 265–319. John Wiley & Sons, 2000.

A. Ramakrishnan. Stochastic processes relating to particles distributed in a continous infinity of states. *Proceedings of the Cambridge Philosophical Society*, 46(4):595–602, 1950.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer-Verlag, 1997. ISBN 0-387-94956-9.

R.-D. Reiss. *A Course on Point Processes*. Springer-Verlag, New York, NY, 1993.

B. J. Richmond and L. M. Optican. Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex. II. Quantification of response waveform. *Journal of Neurophysiology*, 51(1):147–161, Jan. 1987.

B. J. Richmond, L. M. Optican, and H. Spitzer. Temporal encoding of two-dimensional patterns by single units in primate primary visual cortex. I. Stimulus-response relations. *Journal of Neurophysiology*, 64(2):351–369, Aug. 1990.

A. Riehle, S. Gruñ, M. Diesmann, and A. Aertsen. Spike synchronization and rate modulation differentially involved in motor cortical function. *Science*, 278(5345): 1950–1953, Dec. 1997. doi: 10.1126/science.278.5345.1950.

F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes: exploring the neural code*. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-18174-6.

R. W. Rodieck, N. Y.-S. Kiang, and G. L. Gerstein. Some quantitative methods for the study of spontaneous activity of single neurons. *Biophysical Journal*, 2(4):351–368, July 1962.

K. Sameshima and L. A. Baccalá. Using partial directed coherence to describe neuronal ensemble interactions. *Journal of Neuroscience methods*, 94(1):93–103, Dec. 1999.

J. C. Sanchez. *From cortical neural spike trains to behavior: modeling and analysis*. PhD thesis, University of Florida, May 2004.

J. C. Sanchez, D. Erdogmus, Y. Rao, S.-P. Kim, M. Nicolelis, J. Wessberg, and J. C. Principe. Interpreting neural activity through linear and nonlinear models for brain machine interfaces. In *Proceedings of the International Conference IEEE Engineering in Medicine and Biology Society*, pages 2160–2163, Cancun, Mexico, Sept. 2003.

J. C. Sanchez, J. C. Príncipe, and P. R. Carney. Is neuron discrimination preprocessing necessary for linear and nonlinear brain machine interface models? In *Proceedings of the International Conference on Human-Computer Interaction*, 2005.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors. *Advances in Kernel Methods: Support Vector Learning*. MIT Press, 1999.

B. Schrauwen and J. V. Campenhout. Linking non-binned spike train kernels to several existing spike train distances. *Neurocomputing*, 70(7–8):1247–1253, Mar. 2007. doi: 10.1016/j.neucom.2006.11.017.

S. Schreiber, J. M. Fellous, D. Whitmer, P. Tiesinga, and T. J. Sejnowski. A new correlation-based measure of spike timing reliability. *Neurocomputing*, 52–54:925–931, June 2003. doi: 10.1016/S0925-2312(02)00838-X.

H. Seidel. Über die probabilitäten solcher ereignisse welche nur seiten vorkommen, obgleich sie unbeschränkt oft möglich sind. *Sitzungsber. Math. Phys. Cl. Akad. Wiss. München*, 6:44–50, 1876.

K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen. Neural prosthetic control signals from plan activity. *NeuroReport*, 14(4):591–596, Mar. 2003.

D. L. Snyder. *Random Point Process in Time and Space*. John Viley & Sons, New York, 1975.

D. Song, R. H. M. Chan, V. Z. Marmarelis, R. E. Hampson, S. A. Deadwyler, and T. W. Berger. Nonlinear dynamic modeling of spike train transformations for hippocampal-cortical prostheses. *IEEE Transactions on Biomedical Engineering*, 54 (6):1053–1066, June 2007. doi: 10.1109/TBME.2007.891948.

S. K. Srinivasan and A. Vijayakumar. *Point Processes and Product Densities*. Narosa Publishing House, 2003. ISBN 81-7319-558-7.

J. V. Toups and P. H. E. Tiesinga. Methods for finding and validating neural spike patterns. *Neurocomputing*, 69(10–12), June 2006.

W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93:1074–1089, Feb. 2005. doi: 10.1152/jn.00697.2004.

E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, and A. Aertsen. Dynamics of neuronal interactions in monkey cortex in relation to behavioural events. *Nature*, 373(6514):515–518, Feb. 1995. doi: 10.1038/373515a0.

M. C. W. van Rossum. A novel spike distance. *Neural Computation*, 13(4):751–764, 2001.

V. Ventura, R. Carla, R. E. Kass, S. N. Gettner, and C. R. Olson. Statistical analysis of temporal evolution in single-neuron firing rates. *Biostatistics*, 3(1):1–20, 2002.

J. D. Victor. Spike train metrics. *Current Opinion in Neurobiology*, 15(5):585–592, Sept. 2005. doi: 10.1016/j.conb.2005.08.002.

J. D. Victor and K. P. Purpura. Nature and precision of temporal coding in visual cortex: A metric-space analysis. *Journal of Neurophysiology*, 76(2):1310–1326, Aug. 1996.

J. D. Victor and K. P. Purpura. Metric-space analysis of spike trains: theory, algorithms, and application. *Network: Computation in Neural Systems*, 8:127–164, Oct. 1997.

H. Wagner, S. Brill, R. Kempter, and C. E. Carr. Microsecond precision of phase delay in the auditory system of the barn owl. *Journal of Neurophysiology*, 94(2):1655–1658, 2005. doi: 10.1152/jn.01226.2004.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1990.

J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408(6810): 361–365, Nov. 2000. doi: 10.1038/35042582.

S. Wohlgemuth and B. Ronacher. Auditory discrimination of amplitude modulations based on metric distances of spike trains. *Journal of Neurophysiology*, 97:3082–3092, Feb. 2007. doi: 10.1152/jn.01235.2006.

H. Wold. On stationary point processes and Markov chains. *Skand. Aktuar.*, 31:229–240, 1948.

W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue. Modeling and decoding motor cortical activity using a switching kalman filter. *IEEE Transactions on Biomedical Engineering*, 51(6):933–942, July 2004.

J. Yvon. *La théorie statistique des fluides et l'équation d'état.* Herman, Paris, 1935.

BIOGRAPHICAL SKETCH

António R. C. Paiva was born in Ovar, Portugal, in 1980. In 2003, he received his B.S. degree in electronics and telecommunications engineering from the University of Aveiro, Portugal. During his undergraduate studies, he received four merit scholarships, a Dr. Vale Guimarães award for best district student at the University of Aveiro, and an Eng. Ferreira Pinto Basto award from Alcatel Portugal for top graduating student in the major. After completing his undergraduate studies, he did research in image compression as a research assistant of Dr. Armando Pinho for almost a year.

In the fall of 2004, he joined the Computational NeuroEngineering Laboratory under supervision of Dr. José C. Príncipe for his graduate studies, having obtained the M.S. and Ph.D. degrees in electrical and computer engineering in the fall of 2005 and the summer of 2008, respectively. His doctoral research focused on the development of a reproducing kernel Hilbert spaces framework for analysis and processing of point processes, with applications on single-unit neural spike trains.

His research interests are, broadly, signal and image processing, with special interest in biomedical and biological applications. In particular, these include: kernel methods and information theoretic learning, image processing, brain-inspired computation, principles of information representation and processing in the brain, sensory and motor systems, and development of biological and biomedical data analysis methods.