Functional interactions in basal ganglia studied by multiple single unit recordings sorted by an unsupervised template matching algorithm

Olga Chibirova

Interactions fonctionnelles dans les ganglions de la base étudiées par l'enregistrement simultané des activités unitaires discriminées par un algorithme non supervisé de tri de potentiels d'action

Functional interactions in basal ganglia studied by multiple single unit recordings sorted by an unsupervised template matching algorithm

Thèse de doctorat

Présentée à

l'Université Grenoble Joseph Fourier Neurosciences – Neurobiologie

par

Olga Chibirova

Directeur de thèse : Prof. Alessandro E.P. Villa

Jury

Prof. François Berger, Président
Dr. Jean-Pierrre Rospars, Rapporteur
Dr. Tatyana Aksyonova, Rapporteur
Prof. Alessandro E.P. Villa, Directeur de thèse

Grenoble 2006

Adresses

Président du jury : Pr. François BERGER
UJF Grenoble 1
Inserm U318, Pav B Anc Maternité
CHU Micallon, BP 217
38043 Grenoble Cédex 9
Tel. +33-4-76.76.56.25
Email: Francois.Berger@ujf-grenoble.fr

Rapporteur 1 : Dr. Jean-Pierre ROSPARS
Directeur de recherches
UMR 1272 UPMC-INRA-INAPG Physiologie de l'insecte
& Unité Mathématique et Informatique Appliquées
INRA Centre de Versailles-Grignon, RD 10,
78026 Versaille Cédex, France
Tél. +33-1-30.83.33.55
Email: rospars@versailles.inra.fr

Rapporteur 1 : Dr. Tatyana AKSYONOVA
Research Director
Laboratory of Applied Nonlinear Analyses
Institute for Applied System Analyses,
National Academy of Scinces of Ukraine
37 Prospect Peremogy,
Kiev 56, 03056, Ukraine
Tél. +38-044-241.87.05
Email : aks@consy.ms.kiev.ua

Directeur de Thèse : Pr. Alessandro VILLA
UJF Grenoble 1
Inserm U318, Pav B Anc Maternité
CHU Micallon, BP 217
38043 Grenoble Cédex 9
Tél. +33-4-76.76.56.25
Email: Alessandro.Villa@ujf-grenoble.fr

Interactions fonctionnelles dans les ganglions de la base étudiées par l'enregistrement simultané des activités unitaires discriminées par un algorithme non supervisé de tri de potentiels d'action

Olga Chibirova

Laboratoire de neurosciences précliniques, CHU, UJF, Grenoble

La thèse porte sur une nouvelle méthode de tri non supervisé de potentiels d'action et sur son application à l'analyse de l'activité neuronale des ganglions de la base. Le développement de nouvelles approches au tri de potentiels d'action est actuel en vue de nouveaux outils nécessaires à l'électrophysiologie effectuée pendent la neurochirurgie fonctionnelle, autant que pour l'efficacité des l'expériences électrophysiologiques en temps réel.

La méthode présentée dans la première partie de cette thèse est une nouvelle approche à ce problème qui décrit les potentiels d'action à l'aide des équations différentielles avec perturbation caractérisant la variation interne de leur forme. Le logiciel permettant le tri de potentiels d'action non supervisé développé à partir de cette méthode comprends un algorithme automatique d'évaluation d'étalons de classes et de leurs rayons.

La seconde partie présente l'application de la méthode à l'analyse de l'activité neuronale des ganglions de la base. Les donnés pour les analyses ont été recueillis au bloque chirurgical du département de neurochirurgie de l'Hôpital Universitaire de Grenoble pendent l'électrophysiologie intra chirurgicale et représentent le STN (950 enregistrements), le GPI (183) et le SNR (105) de 13 patients parkinsoniens et 2 patients souffrant de dystonie. Les analyses sont destinées à définir les formes typiques de potentiel d'action et à révéler un parallèle entre la nature de l'activité neuronale et la gravité des symptômes de la maladie de Parkinson.

Functional interactions in basal ganglia studied by multiple single unit recordings sorted by an unsupervised template matching algorithm

Olga Chibirova

Laboratoire de neurosciences précliniques, CHU, UJF, Grenoble

The present thesis is devoted to the development of a new unsupervised spike sorting method and its application to the investigation of neuronal activity. The development of new approaches to spike sorting is crucial both for the intrasurgical electrophusiology and for the efficiency of real time electrophysiological experiences.

The method presented in the first part of this thesis is a novel approach to the problem which describes action potential by means of differential equations with perturbation characterizing the internal variation of their forms. The unsupervised spike sorting software implemented on the basis of the presented method comprises an automatic algorithm of estimation of class centers and their radiuses.

The second part presents the application of the method to the investigation of neuronal activity in basal ganglia. The data for the analyses were acquired in the surgical room of the department of neurosurgery of the University Hospital, Grenoble, and represent the STN (950 recordings), the GPI (183) and the SNR (105) of 13 Parkinsonian patients and 2 dystonia patients. The analyses are aimed to define typical forms of action potential and to reveal a parallel between the nature of neural activity and the gravity of the Parkinsonian disease symptoms.

List of publications

1. Chibirova O.K., Larouche S., Rouat J., Villa A.E.P., Aksenova T. I., Computer realization and testing of an algorithm for spike sorting with use of non linear dynamical equations and its application to neuronal activity investigations. "Journal of Automation and Information Science", 2005 - № 1 – p126-136.
2. Chibirova O.K., Villa A.E.P., Aksenova T. I., Informational technologies in neurophysiology: real-time spike sorting. (In Ukrainian: Информационные технологии в нейрофизиологии: классификация в реальном времени импульсов нейронов. Обзор проблем и методов. Системні дослідження та інформаційні технології) 2005 Т- №1 – p.58-68.
3. Chibirova O.K., Aksenova T. I., Benabid A.-L., Chabardes S., Larouche S., Rouat J., Villa A.E.P., Unsupervised Spike Sorting of extracellular electrophysiological recording in subthalamic nucleus Parkinsonian patients. "Biosystems" 79, 2005 –p.159-171.
4. Aksenova, T.I.; Chibirova O.K., Villa,A.E.P., Nonlinear oscillation models for the spike sorting of single units recorded extracellularly, International Joint Conference on Neural Networks 1IJCNN 2004, 25-29 July 2004, Budapest, Hungary, p.3029-3035.
5. Aksenova, T. I.; Chibirova O.K., Dryga, O.A., Tetko, I. V.; Benabid A.-L.; Villa,A. E. P., An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals. "Methods " 30, 2003 – C.178-187.
6. Aksenova T.I., Chibirova O.K., Benabid A.-L., Villa A.E.P., Nonlinear Oscillation Models for Spike Separation. "Lecture Notes in Computer Science (LNCS)", 2002. v. 2526 – p.61-70.
7. Aksenova, T.I., Chibirova O.K., Tetko I.V., Driga A.A., Villa,A.E.P., Recognition of neuron impulses based on non linear dynamic equations. SICPRO'2000, Moscow 25-30 sept 2000. (in Russian: Распознавание импульсов нейронов на основе нелинейных уравнений динамики. - Тез. Докл. Междунар. Конф. «Идентификация систем и задачи управления")
8. Aksenova T.I., Tetko I.V., Dryga O.A., Chibirova O. K., Villa A. E. P. Detection and separation of extracellular neuronal discharges, "Smart Engineering System Design", Proc. ANNIE'01, November 3-8, St. Louise, USA, ASME Press, New York, 2001, p.557-562.

9. Aksenova, T.I., Tetko I.V., Chibirova O.K., Villa,A.E.P., Recognition of neuron impulses by means of non linear equations. “Journal of Automation and Information Science”, 2001 - №3 - P.16-27.

List of conferences and presentations

1. 8P^{thP} International Congress “Parkinson Disease and Movement disorders” (Italy, Rome, July a2004).
2. 4P^{thP} European Forum of Neurology FENS'04 (Portugal, Lisbon, July 2004)
3. International Joined Conference of Neural Networks and Fuzzy Systems IJCNN, FUZZ-IEEE (Hunghary, Budapest, July 2004)
4. 5-ая International Working Conference “Neurocoding - 2003” (Italy, Aula, Faculty of Mathematics, Universit of Turin, August 2003)
5. 5th Conference European Society of Mathematical and Theoretical Biology ECMTB2002 (Italy, Milan, July 2002)
6. 3d International Symposium “Medical Data Analysis” ISMDA’2002 (Italy, Rome, University La Sapienza, 2002)
7. International Conference “Artificial intellect in engineering” ANNIE’01 (USA, St-Luis, Laboratory of Artificial Intelligence, University Missouri Rolla, 2001)
8. International Conference “System Identification and Control Theory Problems”, SICPRO'2000 (Russia, Moscow, Institute de Control Theory Problems, RAS, august 2000)

9.18P^{thP} International Conference IEEE, “Technologies in medicine and biology.” (Netherlands, Amsterdam, 1996)

Aknowlegmetns

Prof. Alessandro E.P. Villa, my thesis director who made me discover the world of the electrophysiology, for guidind and advising me in this world, for his patience and for forgiving me my bad caracter.

Dr. Tatyana Aksyonova, jury member, for having been my guid and teacher for several years, for giving the taste of the scientific work, for her wonderful ideas without wich this research would never have been possible. And for being here and helping me when I have doubts and hesitations.

Dr. Jean-Pierre Rospars, jury member, for taking time to read and anylize my work in very little details, for his very interesting remarks and for his advises for the further research.

Prof. François Berger, the jury president, for the attention to my work for accepting to spend time to review and comment it inspite of his tight schedule.

The presented work is a transdisciplinary research, which wuold have never be possible without support of my daer colleagues. I would like to thank neurosurgeants **Dr. Napoleon Torres**, **Dr. Keyumars Ashkan**, **Dr. Stephan Chabardes** and **Dr. Eric Signoré** for helping with the data aquision during surgeries and with the results interpretation and for answering with patience to all my questions, neurophysiologist **Dr. Brigitte Pialat** and neurolog **Dr. Valérie** for taking time to discuss with me my research and explaine me biological and neurological purposes conserning my work.

Je voudrais remercier du fond de cœur nos merveilleuses secretaires, **Mme Annick Abbadi** et **Mme Chantale Baume** qui non seulement m'ont aidé à survivre dans le monde des demarches administratives mais aussi pour être toujours prêtes à trouver du temps pour écouter mes problèmes et pour savoire à chaque fois comment les resoudre.

Table of contenets

List of figures

List of tables

List of abbreviations

PD - Parkinsonian Disease

STN – Sub Thalamus Nucleus

SNr – Substancia Nigra pars reticulata

GPi – Globus Pallidus internal

GPe – Globus Pallidus external

PPN – Peduncolopontine

HF DBS – High Frequency Deep Brain Stimulation

VIM - Thalamus Ventral Intermediate N ucleus

MPTP - 1-Methyl-4-Phenyl-1,2,3,6-tTtrahydroPyridine

UPDRS - Unified Parkinson Disease Rating Scale

USS – Unsupervised Spike Sorting

TSA – Time and Spatial Adaptation

TW – Typical Waveform

TWL – Typical Waveform Library

PNN – Polynomial Neural Network

GMDH – Group Method of Data Handling

UNIVERSITE JOSEPH FOURIER
SCIENCES TECHNOLOGIE SANTÉ

Interactions fonctionnelles dans les ganglions de la base étudiées par l'enregistrement simultané des activités unitaires discriminées par un algorithme non supervisé de tri de potentiels d'action

Functional interactions in basal ganglia studied by multiple single unit recordings sorted by an unsupervised template matching algorithm

Thèse de doctorat

Présentée à

l'Université Grenoble Joseph Fourier Neurosciences – Neurobiologie

par

Olga Chibirova

Directeur de thèse : Prof. Alessandro E.P. Villa

Jury

Prof. François Berger, Président
Dr. Jean-Pierrre Rospars, Rapporteur
Dr. Tatyana Aksyonova, Rapporteur
Prof. Alessandro E.P. Villa, Directeur de thèse

Grenoble 2006

Table of contenets

List of figures

List of tables

List of abbreviations

PD - Parkinsonian Disease

STN – Sub Thalamus Nucleus

SNr – Substancia Nigra pars reticulata

GPi – Globus Pallidus internal

GPe – Globus Pallidus external

PPN – Peduncolopontine

HF DBS – High Frequency Deep Brain Stimulation

VIM - Thalamus Ventral Intermediate N ucleus

MPTP - 1-Methyl-4-Phenyl-1,2,3,6-tTtrahydroPyridine

UPDRS - Unified Parkinson Disease Rating Scale

USS – Unsupervised Spike Sorting

TSA – Time and Spatial Adaptation

TW – Typical Waveform

TWL – Typical Waveform Library

PNN – Polynomial Neural Network

GMDH – Group Method of Data Handling

Chapter 1

Introduction

Ce chapitre s'articule autour des deux parties principales du présent document. La première partie et dédiée à une nouvelle méthode de tri des potentiels d'actions. La deuxième partie application de cette méthode pour des études de l'activité neuronale dans les ganglions de la base.

The present work is devoted to the investigation of neuronal activity in basal ganglia by means of multiunit electrophysiological recording analyses. The first part of the work presents a novel method of spike sorting. The method is used for the extraction of single units from multiunit records from STN, SNr and GPi of Parkinsonian and dystonia patients. The second part of the thesis studies the neuronal activity in STN, SNr and GPi using the obtained single unit spike trains.

1.1 Multiunit recording and spike sorting. Introduction to Part I

Since the early decades of the electrophysiology in the years 1940s and 1950s, the "single unit" concept referred to the possibility to record the activity of a single neuron. The detection of all spikes from a single neuron is a difficult task when the experiment consists of extracellular recordings of action potentials. The basic hypothesis is that all spikes generated by one specific neuron are characterized by a similar shape and this shape is unique and conservative for each distinct neuron during a stationary recording (i.e., in absence of any movement of the electrode tip with respect to the neural tissue and in absence of any transient electronic and electric noise). The waveform of extracellularly recorded spike depends on the electric properties of the microelectrode, on its relative position with respect to the recorded neurons and on the electric properties of neuronal membrane.

The recording of extracellular neuronal activity in noisy situations (Musial et al., 2002), such as chronically implanted freely-moving animals (Villa et al., 1999) as well as the

neurosurgical electrophysiology (Ashkan et al, 2004), represents a crucial step because it provides unique information about the pattern of neuronal activity of the regions explored during the electrode penetrations. However, the quality of the information gained during the advancement of the electrode depends on the spike sorting, i.e. separation from the background noise of few action potentials (spikes) from the same electrode.

Among different methods used in neurophysiology for spike sorting (Schmidt, 1984; Lewicki, 1998) template-matching has become one of the most popular. This technique is based on templates that represent some typical waveform shapes of neuronal discharges in time domain. The classification of a candidate spike is performed by comparing the electric signal recorded from the microelectrode to all available templates and then by selecting the best matching template. Recent developments of this technique have been determined by the availability of fast computers at cheap price and include computationally intensive methods such as neural networks (Chandra et Optican, 1997; Kim et Kim, 2003) and wavelet transforms (Letelier et Weber, 2000; Hulata et Segev, 2002).

These and many other algorithms proceed the spike waveform classification in real time automatically way. The learning stage of the algorithms is still delicate due to a number of difficulties caused by the nature of the neuronal activity signal. Among these difficulties are high level noise, presence of artifacts, variability of individual neuron's spike waveforms and finally the ignorance of the number of observed neurons.

A fundamental problem of the template-matching technique is represented by the number of distinct waveforms that may be separated from one microelectrode signal. The usual practice is to use a "supervisor", i.e. an experienced human operator, who can provide a preliminary classification of the waveforms following a selection of templates corresponding to distinct single units. Both extracellular and intracellular noise may affect the shape of the action potential (Fee et Mitra, 1996) and the task of spike sorting is even more difficult when the recordings are performed from freely-moving animals, due to the presence of noise at lower frequencies than the signals of interest. The extracellular noise is usually taken into account by most of models as an additive noise. The intracellular noise may produce intrinsic variations in the spike waveform and it is more difficult to account for.

Distributed neuronal activity across cell assemblies may generate synchronized firing across many neurons (Singer, 1999) and complex spatio-temporal interactions (Abeles, 1991; Villa, 2000). It is then necessary to record multiple spike trains simultaneously in order to

gain access to distributed brain processes. Here appears the problem of a decomposition of such records into single unit spike trains. Each spike train is assumed to represent the unique time series of single neuron action potentials. The accuracy of this assumption is particularly important with the increase of microelectrode recordings in humans as guidance to the localization of the optimal site for deep brain stimulation (DBS). The success of DBS for treatment of motor disorders, in particular Parkinson's disease opens new perspectives to the development of electrophysiological techniques in contemporary neurosurgery (Limousin et al., 1998; Krack et al. 1999).

A method for spike sorting is presented in Part I of the thesis. The mathematical background of the method was developed in 1999-2000 in the Institute of System Analysis, Kiev, Ukraine, with participation of the author (Aksenova et al. 2000; Aksenova et al. 2001). The project was directed by Dr. Aksyonova. The method is based on the use of the inverse methods of nonlinear oscillation theory (Chertoprud et Gudzenko, 1976) and considered the problem of spike sorting in phase space of a dynamical self-oscillating system. That makes possible to account the intracellular noise and to reduce the problem to separation of a mixture of asymptotically normal distributions. The spike waveform is described as an ordinary differential equation with perturbation. This mathematical formulation allows us to characterize the signal distortions in both amplitude and phase. Moreover, an unsupervised learning algorithm for automatic selection of representative spike templates is developed. On the basis of the developed algorithm a prototype software for the Unsupervised Spike Sorting (USS) is created. The algorithms are validated and the USS software is tested on a set of simulated signals.

1.2 Functional interactions in basal ganglia. Introduction to Part II

Models of basal ganglia dysfunctions are used to explain the pathophysiological symptoms that characterize PD. According to the generally accepted model the appearance of the symptoms associated to Parkinson's disease (PD) is due, at last partly, by the rise in firing rate of the STN neurons that in turn increase the rate in the neurons of the output basal ganglia nuclei –SNr and Gpi, which exert mainly an inhibitory effect via GABA release. The validity of such models rests upon the correlation between the neuronal activity in the basal ganglia and the degree of clinical symptoms due to PD (Asai et al., 2003; Abe et al., 2003; Niktarash,

2003). The investigation of the activity patterns of subthalamic neurons (Magarinos-Ascone et al., 2000; Liu et al.,2002) represents an important objective for better understanding the mechanisms that subserve the regulatory loops of the basal ganglia. The STN plays a key role in the regulation of the output pathway of basal ganglia. The inactivation of STN in patients affected by PD dramatically reduces much of the clinical symptoms and its reversible inactivation by deep brain stimulation is one of most valuable techniques of present functional neurosurgery (Benabid et al., 1994; Limousin et al., 1998).

It was recently demonstrated that both firing rate and pattern of neuronal activity in the STN and GPi/SNr are altered profoundly by chronic dopamine depletion that underlies the pathology of PD (Magill et al., 2001). High and low-frequency oscillatory patterns in basal ganglia and their modifications in response to behavioral events have been recently studied (Cassidy et al., 2002; Bevan et al., 2002; Levy et al., 2002b). It has been proposed that an increase in synchronization between neuronal discharges in the basal ganglia contributes to generate the appearance of several clinical symptoms typical of PD. Synchronization of the activity patterns (Levy et al., 2000, 2002) and tremor-related neuronal firing (Rodrigues et al., 1998; Hurtado et al., 1999; Hutchison et al., 1997) were examined for Parkinsonian patients. Changes of neuronal firing patterns have been associated to improvement of clinical symptoms typical for PD during DBS (Benazzouz et al., 2000; Hashimoto et al., 2003). Different changes in firing rate, firing patterns, oscillatory processes, and synchronization of activity in the basal ganglia nuclei were revealed by single electrode recordings performed in human PD patients during surgical intervention aimed to relieve parkinsonian symptoms (Bergman et al., 1998a; Filion, 1991; Hurtado et al., 1999; Hutchison et al., 1994, 1997; Merello et al., 1999; Nini et al., 1995).

The present work is a contribution to STN and its output structures SNr and GPi neuronal activity investigation. The study was carried out in tree directions: definition and investigation of activity patterns in STN, definition of typical waveforms of neuronal discharges and their correlation with the defined activity patterns and, finally, the investigation of dependences between the activity patterns and clinical symptoms of PD.

Part I

Unsupervised Spike Sorting (USS) algorithm and software

Chapter 2
Spike Sorting

Ce chapitre est une introduction dans le problème de trie de potentiel d'action (Spike Sorting). Les potentiels d'action sont de rapide perturbations du champ électrique produites des cellules nerveuses. Ils peuvent être enregistrés d'une manière extracellulaire à l'aide d'électrodes implantées dans les tissues nerveux. L'électrode enregistre l'activité des cellules à proximité de sa pointe. Les enregistrements contiennent de pics étroits (spikes) correspondant aux potentiels d'action. La tache du trie de potentiel d'action est de séparer les spikes appartenants aux différentes cellules sur la base de leurs formes.

A ce jour, il existe de nombreuses méthodes du tri de potentiel d'action, dont une brève revue est présentée dans ce chapitre. Le problème du développement de nouvelles approches est néanmoins actuel en vue de nouveaux outils nécessaires à l'électrophysiologie effectuée pendent la neurochirurgie fonctionnelle, autant que pour l'efficacité des l'expériences électrophysiologiques en temps réel.

La variabilité de la forme des potentiels d'action, ainsi que la présence d'artefacts et du bruit non stationnaire, complique le développement des algorithmes non supervisées de trie de potentiels d'action. La méthode présentée dans la première partie de cette thèse décrit une nouvelle approche à ce problème.

For a great number of neurophysiological studies it is necessary to isolate single neural cell activity. Examples of such studies are investigation and comparison of neuronal activity in various brain structures, neuronal activity characterization for pathologies, animal behavioral experiments. During extracellular recording the cells detected by a single electrode are those lying within tens of microns of the electrode tip and extracellular recording is usually multineuronal, often referred to multiunit analysis. It is sometimes possible to isolate a single unit by manipulating the electrode, but these manipulations are, however, difficult or impossible to execute.

The electric signal recorded by the electrode usually includes the activity of many cells. It is characterized by occurrences in random times of short impulses of a particular form, called spikes (figure 2.1). Spike sorting is the signal processing applied to sort spikes

into groups according to their waveforms, each group being presumed to represent a single cell. Moreover, the spike sorting approach can provide information that is extremely difficult to obtain using one-cell-one-electrode approach.

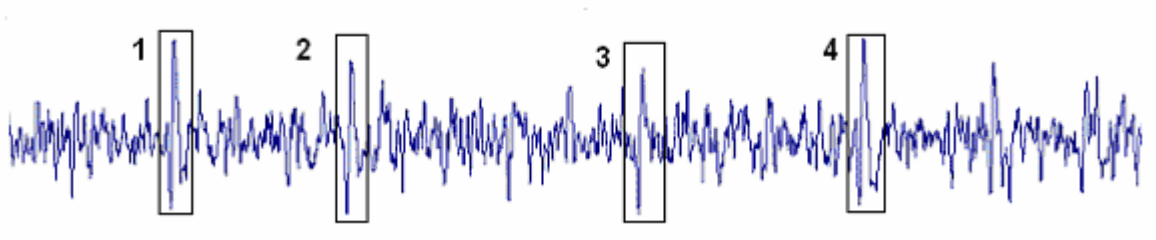


Figure 2.1
Electrophysiological signal recorded extracellulary during a functional neurosurgery of a Parkinsonian patient. 1-4 – spikes, corresponding to neural action potentials. Spikes 1 and 2 are similar in waveform and may be single unit spikes.

The aim of the present work is to apply a spike sorting technique that can be used during functional neurosurgery to increase the efficiency of the intrasurgery electrophysiology. For that purpose the spike classification algorithm had to be in real-time. Moreover it was preferable to develop an unsupervised spike sorting to avoid the necessity of the presence of an experienced electrophysiologist during surgery. The time factor in functional neurosurgery is crucial, so the learning algorithm for the spike classification had to be efficient and not time consuming. In the same time the surgery room spike sorting software was to be convenient and simple, not requiring special computer skills. The efficiency and simplicity of a spike sorting software is also important for on-line experiments with animals. For the off-line electrophysiological data treatment the mentioned features are useful as well, since it permits to treat easily large amounts of data. Thus, despite of a great number of spike sorting methods available to date it appears necessary to create new and more efficient tools, requiring less user efforts, getting closer to a fully automatic procedure.

In this chapter the nature of neuron action potentials and the principles of exracellular electrophysiological recordings are briefly exposed. Then, a general scheme of spike sorting algorithms is presented as well as a short review of existing spike sorting methods.

2.1 Action potential

The neurons communicate by means of short local perturbation of the electrical potentials across the cell membrane, called action potentials. The mechanism of action potential was discovered and described by Hodgkin et Haxley (1952).

The membrane voltage changes during an action potential result from changes in the permeability of the membrane to specific ions (particularly sodium Na^+ and potassium K^+), the internal and external concentrations of which is maintained in an imbalance. This imbalance makes it possible to generate action potentials but also the resting membrane potential. Although the concentrations of the different ions attempt to balance out on both sides of the membrane, they cannot because the cell membrane allows only some ions to pass through ion channels. At rest, potassium ions (K^+) can cross through the membrane easily through the potassium leak channels, sodium ions outside and negatively charged protein molecules inside the neuron cannot cross the membrane. The resting membrane potential of a neuron is about -70 mV. At rest, there are more sodium ions outside and more potassium ions inside that neuron.

. In a simplified model of the action potential, the resting potential of a patch of membrane is maintained by a potassium leak channel. The rising phase (figure. 2.2) of the action potential occurs when the voltage-dependent sodium channels open causing the sodium permeability to greatly exceed the potassium permeability. This critical opening of the voltage dependent sodium channels occurs when the membrane potential reaches a critical level, reffered to as the “threshold potential”. After a short delay, the voltage-dependent potassium channels opens and the voltage-gated sodium channel become inactiv. As a consequence, the membrane potential is driven back toward the resting potential, resulting in the action potential's falling phase. As more potassium channels are open than sodium channels, the potassium permeability is now larger than it was before the action potential was generated (at rest only the potassium leak channel is open). As a result, the membrane potential undershoots the resting potential level. The delayed-rectifier potassium channel, being voltage-dependent, is closed by the hyperpolarized voltage, and the cell returns to its resting potential.

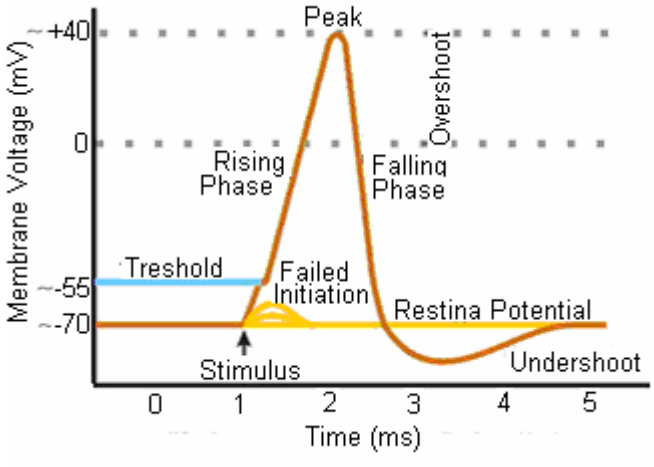


Figure 2.2
Schematic action potential with aresting potential at -70mV and a thresholde at -55mV.

Where membrane has undergone an action potential, a refractory period follows. This period arises primarily because of the voltage-dependent inactivation of sodium channels. In addition to the voltage-dependent opening of sodium channels, these channels are also inactivated in a voltage-dependent manner. Immediately after an action potential, during the absolute refractory period, virtually all sodium channels are inactivated and thus it is impossible to fire another action potential in that segment of membrane. With time, sodium channels are reactivated in a stochastic manner and as they become available, it becomes possible to fire an action potential, albeit one with a much higher threshold. This is the relative refractory period and together with the absolute refractory period, lasts approximately five milliseconds.

In many cases a cell fires a group of action potentials spaced by a little more then the absolute action potential. Such a group is call “burst”. In general, such bursts are not driven entirely by synaptic input, but rather by the biophysics of the membrane. For example, extremely long voltage sensitive calcium channels are found in some neurons: the first action potential in a burst provokes the opening of some of these channels, but they neither close nor inactivated rapidly. Ca++, which is concentrated outside the cell by the ion pumps, flows in through these open channels. As a result, as soon as the first action potential is over and the potassium channels closed, the depolarizing calcium channels can provoke the next action potential. The current that flow in this and subsequent action potentials may be lower then in the initial one.

2.2 Extracellular recording

The mechanism of action potential, as well as many others neuronal phenomena, have been understood through measurements taken using an intracellular electrode, that is one which penetrates the cell. It is difficult to record with such an electrode in intact animal and even more difficult in an awaken one.

Extracellular recording is used to measure the extracellular field potentials outside the neurons. In its most simple form, extracellular recording can be performed by placing a single wire electrode in the brain that has insulation covering all but its very tip. Fluctuations in the voltage between this wire and a neutral reference wire (e.g. a wire attached to a skull screw) can then be measured. Since the fluctuations in the local field potential that occur in the brain are commonly in the frequency range of about 1 kHz and less than 1 mV, the signal must be amplified so that it may be detected and recorded. In the process of amplifying the signal, it is useful to filter it to remove very low (<1Hz) and very high (>3kHz) frequencies.

In intracellular recording the voltage fluctuations across the resistor of the cell's membrane are mesured. In extracellular recording, the recording electrode is outside the cell (fig. 2.3).

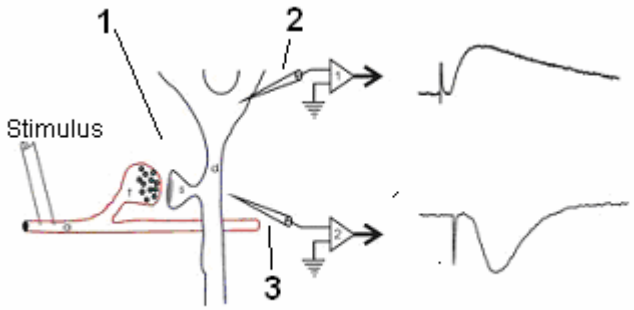


Figure 2.3
A schematic diagram, showing an extracellular field potential recording from rat hippocampus. At the left is a schematic diagram of a presynaptic terminal and postsynaptic neuron (1). Then the synapse releases glutamate the net flow of current is inward, so a current sink is generated. An extracellular electrode (3) detects this as negativity. An intracellular electrode placed inside the cell body (2) records the change in membrane potential that the incoming current causes.

The resistor in extracellular recording is the tip of the electrode itself. Extracellular electrodes can record transient changes in the local balance of positive and negative charges. Since the inside of the electrode is electro neutral, and the tip has a resistance, a voltage can develop

across the electrode tip between the electro neutral interior and the exterior local change in charge balance.

Using this basic strategy, if the wire is placed near to a neuronal cell body (less than 140 microns according to Henze et al. (2000)), action potentials fired by that cell may be recorded. This is possible because to fire an action potential the neuron transiently opens sodium channels allowing positively charged sodium ions to rush down the voltage gradient into the cell. This movement of ions into the cell creates a negative fluctuation in voltage in the immediately surrounding area relative to distant locations. This leads to a transient change in voltage between the extracellular recording electrode and the distant reference wire.

Many cell membranes might lie close to the electrode tip so that many spikes from many cells are recorded. To isolate a single cell activity the experimenter may move the electrode so that its tip lies very close to the cell body, and thus the spikes from this cell appear far larger in amplitude than those from the other cells. Due to such difference in amplitude a simple hardware device can be used than to record the spike train of the single cell. However, optimizing the single cell isolation by moving the electrode is difficult if possible. Instead, spike sorting algorithms can be applied to obtain the single unit isolation.

2.3 Spike Sorting methods review

Single unit activity detection on multiunit extracellulary recorded signal has been an object of intensive research since last 2-3 decades. During this period many different approaches to the problem solution were suggested. In general all on-line spike sorting methods deal in one or another way with the following sub problems:

- Spike event detection –extraction of spikes from the electrical signal.
- Definition of features for the classification, which may be as simple as maximal amplitude feature, or more sophisticated, like wavelet transform coefficients.
- Learning of the classification algorithm, which requires the number of classes estimation, as well as the estimation of class's parameters on a learning set.
- On line classification.

In different approaches each of these problems may be solved on- or off-line, automatically, semiatomatically or manually. In many developed approaches the first stage – spike detection, and the last one – spike classification - are realized on-line and in an automatic way (Gadike et Albus, 1995; Zouridakis et Tam, 1997; Kreiter et al., 1997; Kim et Kim, 2003). The most complicated problem is the problem of automatic learning of the classification algorithm, i.e. automatic estimation of number of classes and of their parameters. Nevertheless the development of automatic methods of classification is crucial since the manual classification is not only a time-consuming but also subjective, operator-depending procedure and the probability of error is high (Harris et al., 2000; Wood et al., 2004). Many authors suggest an automatic learning procedure which is not robust to outliers and requires manual collection of the learning set (Forster et Handwerker, 1990; Bergman et DeLong, 1992). A learning set collected automatically unavoidably contains outliers. Often the learning procedure results are controlled and corrected by an operator (Gadike et Albus, 1995). In some approaches certain stages may be omitted if a sub problem is not considered and its solution is considered to be known a priori. For example, in most of technics the number of classes is assumed to be defined. Some algorithms put together the learning and the classification stages, providing an off-line clustering on a large data set (Fee et al., 1996; Sahani, 1999; Quian Quiroga et al., 2004; Shoham et al., 2003).

A number of factors that increase the difficulty of the single unit activity detection should be taken into account when choosing a method of spike sorting. Most of sophisticated algorithms provide excellent results on “good” signals where the negative factors are not important, but these algorithms may be inapplicable in the opposite case. The use of simplest methods based on threshold crossing is often preferable.

1.3.1 Threshold crossing

The simplest solution of the single unit activity detection problem is the threshold crossing method (Schmidt, 1984). This method uses one of the principal spike waveform characteristics – its amplitude, which mostly depends on the distance from the cell to the electrode. Whenever the signal crosses a threshold set by user, a spike event is recorded. The obvious advantage of the method is the simplicity and the minimum requirements to the software and equipment. However by means of the threshold crossing it’s impossible to distinguish cells with slightly different large spike amplitudes. Moreover this method is

inapplicable in presence of high amplitude artifacts. All detected spikes, as well as mistaken artifacts are considered to belong to the same single unit and the firing rate is overestimated.

In the case of threshold crossing only one sub problem of spike sorting is solved - event detection.

1.3.2 Template Matching and Clustering in Feature Space

All spike sorting algorithm based on spike waveform analysis can be conventionally divided into two groups: Template Matching (Abeles et Goldstein, 1977; Gadike et Albus, 1995; Forster et Handwerker, 1995; Peterson et Merzenich, 1995; Okada et Maruyama, 1982; Zouridakis et Tam, 2000; Simon, 1965) and Clustering in Feature Space (Fee et al., 1996; Sahani, 1999; Quian Quiroga et al., 2004; Shoham et al., 2003; etc).

Template Matching is a group of classification algorithms based on matching spike waveform to the previously defined templates. Usually each template corresponds to a single unit. All sample points of the signal on the segment corresponding to a spike are usually used as features for the classification. In earlier applications the templates were defined manually by operator (Friedman, 1968; Bergman et DeLong, 1992; Okada et Maruyama, 1982) which is a time consuming process. In later approaches the templates are estimated as the mean spike waveform of a group belonging to a signal unit. For that purpose a clustering is necessary and the mean is estimated for each cluster. Estimation of the class center as the mean of the cluster has its drawback – the mean is sensible to statistical errors. To make the estimation more robust the clustering algorithm must take into account the presence of outliers. An example of such a sophisticated algorithm is a fuzzy clustering (Zouridakis et Tam, 2000). It is assumed that each spike belongs to all clusters with some probability according to Euclidian distance to the center of the cluster. The clusters are constructed iteratively. The iterative procedure requires a large amount of calculations.

Clustering in Feature Space uses as features some spike waveform characteristics such as maximal and minimal amplitudes, the width of spike etc.) Single unit spikes form a cluster in the feature space. The aim of the Clustering in Feature Space is to find clusters corresponding to each observed single unit and to define its boundaries for further classification. The simplest earlier methods required a manual clustering. Later methods based on some probabilistic models appeared. Considering the distribution in clusters as

Gaussian, it is possible to provide a Bayes clustering as it is described in Cheesman et Shutz, (1988). Moreover, these authors suggest a method of rejecting artifacts by creating a special class with big radius and small wait. The number of classes is estimated here by maximal likelihood method for mixtures of different number of Gaussian distribution (Snider et Bond, 1998; Glaser et Marks, 1998). Nevertheless, this method fails if the real distributions differ significantly from the Gaussian. That is why the later works suggest clustering methods which are not based on this assumption (Fee et al., 1996; Glaser et Marks, 1998). In the first work the clustering is realized on the basis of recursive bisection. The set is divided into a large number of small clusters which are merged then if the probability density between clusters exceeds a threshold. The second method is analogous and based on the assumption that the spike waveform variation is continues in time. The small clusters are merged according to this assumption. Both methods require a large amount of calculations.

Spike Detection

The first stage of Template Matching and Clustering in Feature Space algorithms is the detection spike occurrences within a multiunit signal. The threshold for spike detection is usually defined according to statistical characteristics of signal. Segments of the signal around threshold crossings are considered as potential spikes of one of the observed neuron cells. One of major problems making difficult a further classification of these signal segments is that noise is usually nonstationary and its amplitude sometimes can be of the same order of the signal amplitude. Thus, segments of noise and artifacts are detected together with spikes.

Feature Space

For Clustering in Feature Space some characteristics of spike waveform are used as features, often it is the extrema amplitudes, time between the local extrema etc. (Feldman et Roberge, 1971; Dinning, 1981; Lewichki, 1994). Template Matching methods take into account the entire waveform, here all time samples are considered as features. Feature space can be optimized using, for example, the principal components method (Glaser, 1971; Gerstein et al., 1983; Salganicoff et al., 1988) which selects from the initial feature set several features providing the best classification results to the reduced feature space method (Kreiter et al., 1989; Lewicki, 1994) which reduces the number of features to minimum necessary for the calculations. Sometimes the optimization is carried out manually. For example in Kreiter et al., 1989 the operator selects 8 features as 8 among 64 sample points with maximal

variance on a learning set. Then, learning and classification are carried out in 8-dimentional feature space.

Synchronization of spikes

If the signal samples are considered as features then the variability of the spike waveform leads to the problem of the spike segments synchronization. If the internal noise is absent spikes can be synchronized by a time shift, for example, by matching the spike mass centers (Forster et Handwerker, 1990; Bergman et DeLong, 1992). But the spike waveform may vary because of internal process in the nervous cell (Fee et al., 1996; Quirk et Wilson, 1999; Oweiss et Anderson, 2002). That leads to nonlinear spike waveform variability and spikes can be no more synchronized by a time shift. In order to take into account the nonlinear deformation spikes may be dividedinto several segments which are synchronized independently. The synchronization may be performed, for example, following the principle of minimal surface between two segments (Kreiter et al., 1989) or by finding the optimal synchronization iteratively, shifting spikes by one sample point on each step and maximizing the correlation between them. These procedures increase the time of calculations.

Learning procedure

This is the most complicated stage of spike sorting. It is necessary to construct templates for Template Matching or to define cluster boundaries in case of Clustering in Feature Space. The number of classes corresponding to the number of neurons being observed is not known a priori and must be estimated during the learning phase. Moreover, if the learning set is formed automatically, it contains artifacts and misdetected noise which must be rejected during the learning process. Finally, if signal samples are taken as classification features, spike distributions in such feature space in general are not Gaussian (Fee et al., 1996; Shoham et al., 2003) and the methods based on the assumption of Gaussian distributions in feature space may not be efficient. Other clustering methods based on more complicated probabilistic models, for example clustering with gradient of probability density (Fee et al., 1996), hierarchical model of Gaussian distributions (Sahani, 1999), Student's distribution model (Shoham et al., 2003) give better results but often demand more complicated calculations.

In the simplest case an operator carries out the templates selection based on visual inspection or on manual tracing of cluster boudaries (Gadike et Albus, 1995; Bergman H.,

DeLong, 1992; Okada et Maruyama, 1982). This is a time consuming, difficult and subjective process. For example in Gadike et Albus, (1995) the operator has to estimate boundaries for 27 features used for the clustering. The constraints mentioned above – high amplitude noise, artifacts, spike waveform nonlinear variability – make the implementation of an automatic learning procedure rather sophisticated. Often, the learning set must still be inspected by an operator to reject noise and artifacts. As for the number of classes estimation, sometimes the automatic clustering is carried out for several possible number of classes. The final number of classes is selected according to the best clustering by visual inspection (Zouridakis et Tam, 2000; Snider et Bond A.B., 1998; Glaser et Marks, 1968) or by some criteria, for example the criteria of maximal likelihood (Cheesman et Shutz, 1988).

Classification

This stage of spike sorting is rather simple and consists either in matching a detected spike with one of the templates defined during the learning stage or by verifying the cluster boundaries. Many of modern application provide automatic real-time classification (Lewicki, 1994).

In some works the learning and the classification stages are joined together (Oweiss et Anderson, 2002; Fee et al., 1996; Sahani, 1999; Quian Quiroga et al., 2004; Shoham et al., 2003). In this case the spike sorting is provided by off-line clustering of a large set of detected spikes (e.g. 50 000 to 100 000 spikes (Fee et al., 1990)).

1.3.3 Other spike sorting algorithms

The methods described above use the spike waveform characteristics for the feature space definition. However, there are also other approaches to the problem. Some of them are based on frequency methods, such as optimal linear filtration (Roberts et Hartline, 1975; Gozani et Miller, 1994). The method is based on the optimal filter construction for each of the template spikes selected manually. This approach shows worse results of the classification but its advantage is that it allows detection of overlapping spikes. Another approach, allowing as well the detection of overlapping spikes is the neural network (Yamada et al., 1992; Mirfakhraei et Horch, 1994). The realization of these algorithms requires a manual learning set collecting and a prior class number estimation. The most popular modern methods are based on the wavelet transform (Quian Quiroga et al., 2004; Zouridakis et Tam, 1997; Letelier

et Weber, 2000). The coefficients of the wavelet decomposition of template spikes are considered as classification features. Clustering methods in this feature space are analogous to the methods used in time domain for Clustering in Feature Space and Template Matching and usually they can be realized automatically in off-line mode. The quality of the classification in the feature space of wavelet transform coefficients is very high and the latest algorithms work in a very efficient way. However, a preliminary basis construction is required for these algorithms.

Chapter 3

Mathematical purposes

Le tri de potentiels d'action doit considérer deux types de variations de la forme des potentiels : le bruit externe et la variation intrinsèque d'origine cellulaire. Ce fait implique le problème lié à l'observation de potentiels d'action synchronisés. Il a été démontré que la distribution des potentiels d'action dans le domaine temporel n'est pas Gaussien. La classification des potentiels d'action doit être effectuée en présence d'erreurs d'artéfacts et au bruit non stationnaire

Le logiciel de Unsupervised Spike Sorting (tri de potentiel d'action non supervisé) est basé sur une nouvelle méthode de tri de potentiels d'action. Cette méthode décrit les potentiels d'action à l'aide des équations différentielles avec perturbation qui caractérisent la variation interne de leur forme. Ces équations décrivent un système dynamique oscillatoire, dont les propriétés permettent de réduire la tâche du tri de potentiels d'action à la tâche de séparation d'un mélange de distributions normales asymptotiquement dans l'espace de phases du système dynamique.

Un algorithme automatique d'évaluation d'étalons de classes et de leurs rayons a été développé. A chaque itération l'algorithme examine le groupe d'apprentissage pour trouver le potentiel d'action avec la densité de probabilité maximale dans leur voisinage. A la dernière itération les potentiels d'action sont considérés comme étalons des leurs classes respectives.

This chapter presents the mathematical description of our novel method for spike sorting. The method belongs to the wide class of template matching algorithms for spike sorting. Among different methods used in neurophysiology for spike sorting (Schmidt, 1984; Lewicki, 1998) template matching is one of the most popular. This technique is based on construction of templates that represent the typical waveform of neuron (Bergman and DeLong, 1992; Forster and Handwerker, 1995; Gadike and Albus, 1995). The algorithms of this class compare the candidate spike waveforms with all available templates and select the

best matching one. Most of the algorithms realize the spike matching in time domain, i.e. using the sample points of digitized signal. The drawback of this method is that spike waveforms could be slightly distorted not only in amplitude, but also along the time axis. As a consequence, classes of spikes in time domain may not form clusters and the distributions inside the classes may not be Gaussian (Fee et al., 1996).

The present approach is based on the use of the inverse methods of nonlinear oscillation theory (Chertoprud et Gudzenko, 1976). Spike waveform is described as an ordinary differential equation with perturbation. This mathematical formulation allows the characterization of signal distortions in both amplitude and phase. Spike sorting is considered as a problem of the classification of trajectories in phase space of the dynamical system corresponding to the differential equations. In this feature space the problem of spike classification may be reduced to the separation of a mixture of normal distributions.

The implementation of this method includes several steps. Firstly, a procedure for detecting spike occurrences out of a noisy signal must be performed. Secondly, the estimation of the trajectories in the phase space must be calculated, by the appropriate numerical algorithms. Finally, clustering and classification algorithms should be realized in the transformed feature space. The unsupervised learning algorithm that has been developed here allows the automatic selection of representative spike templates.

3.1 Model

3.1.1 Dynamical system with perturbation

We suppose that an electrical signal $\tilde{x}(t) = x(t) + \xi(t)$ is observed at discrete times; $x(t)$ is the neuronal activity signal without noise; $\xi(t)$ is a sequence of independent uniformely distributed random variables with zero mean and finite variance ($\sigma_\xi^2 < \infty$); $x(t)$ is characterized by the occurrences of spikes. Spikes $x_i(t_i^* + t)$, $0 < t \le T^i$ appear at random times t_i^* and have duration T^i. All other time signal $x(t)$ equals zero. Each spike is assumed to correspond to a neuronal discharge generated by one of p observed neurons. $\mathbf{X}_j$ denotes the general population of spikes generated by a single neuron j, $0 \le j < p < \infty$;

$\mathbf{X}_j = \{x_i(t_i^* + t), 0 < t < T^i\}$. Since the spikes of q single neuron are assumed to be similar, each general population contains spikes with similar waveform. The general assumption on which is based the approach is that all spikes of general population $\mathbf{X}_j$ are solutions of the same ordinary differential equation with perturbation

$$\frac{d^n x}{dt^n} = f^j\left(x,...,\frac{d^{n-1}x}{dt^{n-1}}\right) + F(x,...,t), \tag{3.1}$$

where *n* is the order of the equation, $F(x,...,t)$ is a perturbation function and the equation

$$\frac{d^n x}{dt^n} = f^j\left(x,...,\frac{d^{n-1}x}{dt^{n-1}}\right) \tag{3.2}$$

describes a self-oscillating system with a stable limit cycle $\mathbf{x}^0(t) = (x_1^0(t),..., x_n^0(t))'$ in phase space with coordinates

$$x_1 = x, x_2 = \frac{dx}{dt},..., x_n = \frac{d^{n-1}x}{dt^{n-1}}. \tag{3.3}$$

The period of stable oscillations is T. The perturbation function $F(x,...,t)$, bounded by a small value, is a random process with zero mean and small correlation time $\tau << T$. f^j is twice continuously differentiable on all its arguments.

In case of stable oscillations (Eq. 3.2) the trajectory of the signal continuously tends to the limit cycle $\mathbf{x}^0(t)$ whenever it is found in its neighborhood (Bogoljubov et Mitropolsky, 1961). The perturbation function $F(x,...,t)$ in Eq. (3.1) tends to displace the trajectories of the signal out of the limit trajectory. However, if the perturbation is small enough the trajectories stay in neighborhood of the limit cycle $\mathbf{x}^0(t)$, *i.e.* the solutions of Eq. (3.1) are similar but do not coincide.

We introduce local coordinates $(\mathbf{n}(\theta),\theta)$ in neighborhood of the limit cycle (Chertoprud et Gudzenko, 1961; Gudzenko, 1962). Let us fix an arbitrary point on the limit cycle P_0 as the starting point (figure 3.1). The position of any arbitrary point P on the limit trajectory can be described by its phase θ, which is the time of movement along the limit cycle from the starting point P_0 to point P. Phase θ unambiguously characterizes all points of the limit trajectory. Since f^j in Eq.(3.2) is twice continuously differentiable on all its arguments, it is possible to construct a hyperplane (and only one) that is normal to the limit cycle at point P with phase θ. Point $M(\theta)$ of intersection of this hyperplane and an arbitrary trajectory of Eq.(3.1) corresponds to phase θ. The zero point on this trajectory is $M_0 = M(0)$. Any trajectory in the vicinity of the limit cycle can be described by variables $(\mathbf{n}(\theta),\theta)$, where $\mathbf{n}(\theta)$ is vector PM in the phase space Eq.(3.3). $t(\theta)$ is the time of movement along the trajectory from an initial point $M(0)$ to the analyzed point $M(\theta)$. Thus, the limit trajectory is defined by $\mathbf{n}(\theta) \equiv \mathbf{0}$ and $t(\theta) \equiv \theta$, where $\mathbf{0}$ is a vector with all components equal to *0*. So, for the stable limit cycle the phase equals the time of motion.

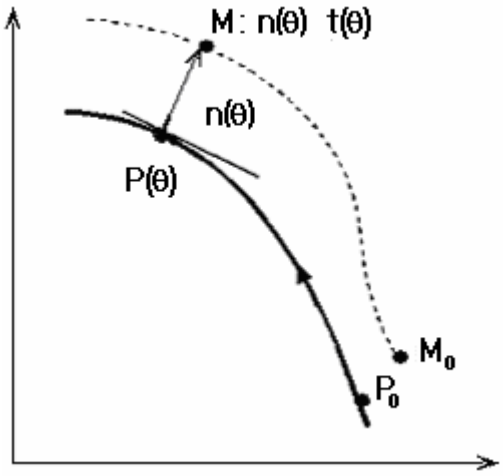


Figure 3.1

New variables, phase θ, time motion $t(\theta)$ from $M_0 = M(0)$ to $M(\theta)$ and normal deviation $\mathbf{n}(\theta)$ are introduced to describe the trajectories of the analyzed signal in phase space. The thick line is the limit trajectory. The length of vector $|\mathbf{n}(\theta)|$ corresponds to the minimal distance between the signal and the limit trajectory.

From the point of view of the described model the spikes of each general population $\mathbf{X}_j$ represent cycles in the neighborhood of the stable limit cycle of the appropriate equation.

The limit trajectory $\mathbf{x}_j^0(\theta)$, $0 < \theta < T$, corresponds to an ideal spike without noise, referred to as an "undisturbed spike" (Figure 3.2).

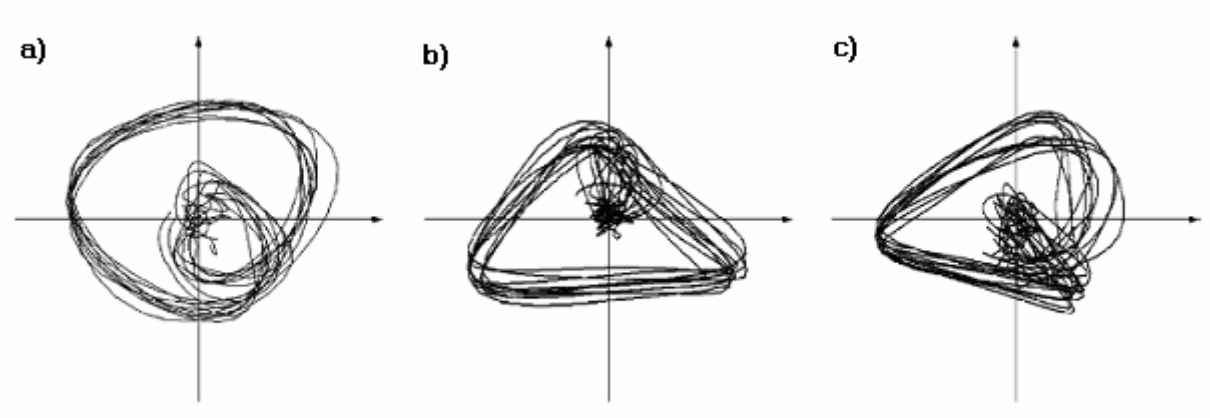


Figure 3.2
Trajectories in phase space Eq.(3.3) of spikes of three single neurons recorded simultaneously from the same microelectrode. Trajectories stay in neighborhood of the limit cycle. They are similar but do not coincide.

Subsequently, the spike sorting method rests upon the following fundamental assumptions:

- spike waveforms are considered as solutions of differential equations with perturbation according to the mathematical model introduced in Eq. (3.1) and Eq. (3.2). The perturbation represents the internal variation of neuronal discharges and is introduced directly in the dynamical system.
- The general population $\mathbf{X}_j$ representing a single unit activity consists of signals described by the same differential equation, so a single unit activity can be described by one dynamical system.

3.1.2 Feature space

The problem of spike sorting can be reduced to the problem of separation of a mixture of normal distributions taking into account some important properties of the self oscillating system with perturbation, Eq. (3.1) and Eq. (3.2) (Gudzenko, 1962):

i) Vectors of normal deviations from limit cycle $\mathbf{n}_j(\theta)$ in phase space Eq.(3.3) have an asymptotically Gaussian distribution for any θ in case of weakly correlated noise $F(x,...,t)$ (correlation time $\tau << T$, T - the period of the oscillating system).

$$\mathbf{n}(\theta) \sim N(0, \sigma^2) \tag{3.4}$$

ii) The mean trajectory of signals in phase space Eq.(3.3) converges to the limit cycle in linear approximation if the number of averaged trajectories increases infinitely. Thus, the mean trajectory represents an estimation of the limit cycle in first approximation: $n^*(\theta) \approx 0$, $t^*(\theta) \approx \theta$ (Chertoprud et Gudzenko, 1976; Gudzenko, 1962).

$$n^*(\theta) = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} n_i(\theta), \quad t^*(\theta) = \lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} t_i(\theta), \tag{3.5}$$

These properties allow the estimation of the limit cycle by calculating the mean trajectory of the signal in the phase space and the estimation of the "undisturbed spike" (the ideal spike in phase space) that can be used as a template for spike sorting.

The standard feature space for template matching algorithms is usually formed by signal samples: $x_i = x_i(t_i)$ t_i, $i = 1,...,M$. This feature space of dimension M will be denoted R^M. A transform of this feature space is used for the approach being described in here. The values of the spike trajectories in phase space Eq.(3.3) at points θ_i, $0 < \theta_1 < \theta_2 < ... < \theta_M = T$, $\Delta\theta = 1$ are considered as new features for classification:

$$\mathbf{X} = \left(\mathbf{x}'(t(\theta_1)) \middle| \mathbf{x}'(t(\theta_2)) \middle| ... \middle| \mathbf{x}'(t(\theta_M))\right)', \tag{3.6}$$

where

$$\mathbf{x}(t(\theta_i)) = \left(x(t(\theta_i)), \frac{dx(t(\theta_i))}{dt}, \ldots, \frac{dx^{n-1}(t(\theta_i))}{dt^{n-1}} \right)' .$$

The transformed feature space, which will be denoted Ω has dimension $n \times M$, n – order of the differential equation. It is important to note that the partition of the interval of spike observation becomes generally irregular in time, *i.e.* $t(\theta_{i+1}) - t(\theta_i) \neq 1$. The Euclidean norm in this feature space is

$$\|\mathbf{x}\|_{\Omega} = \sqrt{\sum_{k=1}^{M} \sum_{i=0}^{n-1} \left(\frac{d^i x}{dt^i}(t(\theta_k)) \right)^2} \tag{3.7}$$

According to the self oscillating system properties Eq.(3.5), Eq.(3.6) for each general population $\mathbf{X}_j$, $0 \le j < p$, the vector $n(\theta) = x^0(\theta) - x^i(t(\theta))$ has an asymptotically normal distribution with mathematical expectation close to zero for any θ. In the new feature space $\mathbf{\Omega}$ we have obtained *p* normally distributed classes corresponding to *p* general populations, *i.e.* to *p* observed neurons. So, the problem of classification in the standard feature space with generally non-Gaussian distribution (Fee et al., 1996) was replaced by the problem of separation of a mixture of asymptotically Gaussian distributions in the transformed feature space. An undisturbed "ideal" spike $\mathbf{x}_j^0$ corresponds to center of class $\mathbf{X}_j$ and can be estimated by averaging of the spike trajectories of the general population $\mathbf{X}_j$ in phase space. It is important to note that the transformed feature space dimension is n times greater than the dimension of the standard feature space, n – order of the differential equation Eq.(3.1). Nevertheless, the advantage to have normal distributions in the transformed feature space allows the implementation of efficient algorithms.

3.2 Algorithms

3.2.1 Estimation of the trajectory in the phase space

The method based on the described model requires the estimation of the spike trajectories in phase space Eq.(3.3). Higher-order derivatives of the signal should be calculated in presence of noise that seriously affects the calculations. Thus, following (Aksenova et Shelekhova, 1994) the following integral operator with a regularization parameter was used for the derivative estimation:

$$D_\alpha^k x(t) = \int_R \omega_\alpha^{(k)}(\tau - t) x(t) dt \,, \tag{3.8}$$

where kernel function ω_α satisfies the conditions:

a) $\omega_\alpha = 0$, when $|t| > \alpha$,

b) $\int_R \omega_\alpha(\tau - t) dt = 1$,

c) ω_α has k continuous derivatives.

D_α^k with parameter of regularization α estimates the smoothed k-order derivative of the signal (Aksenova et Shelekhova, 1994). A computationally efficient algorithm of derivative estimation (Aksenova et Shelekhova, 1995; Aksenova et Shelekhova, 1997) is used to calculate D_α^k.

In phase space spike trajectories are described now in coordinates:

$$D_\alpha^0 \tilde{x}(t), D_\alpha^1 \tilde{x}(t), ..., D_\alpha^{n-1} \tilde{x}(t)$$

instead of the original coordinates $x, \frac{dx}{dt}, \cdots, \frac{d^{n-1}x}{dt^{n-1}}$. Moreover D_α^k is a linear operator. Thus, Eq.(3.1) stays the same in linear approximation in coordinates (3.9):

$$D_\alpha^n x(t) = \hat{f}\left(D_\alpha^0 x(t), ..., D_\alpha^{n-1} x(t)\right) + F\left(D_\alpha^0 x(t), ..., t\right)$$

The application of the integral operators replaces the complex problem of derivative calculation with a simpler calculation of integrals. Any function satisfying the condition set (a)-(c), Eq.(3.8) can be used as a kernel function. Selection of both the kernel function and the value of parameter α depend on the order of the derivative to be calculated, on the level of the additive noise, and on the required smoothness of data.

3.2.2 Detection of spike occurrences

The procedure of spike detection is necessary to determine the time intervals of the signal that correspond to spike occurrences. Spikes are usually characterized by an amplitude which is significantly larger than the level of background noise and their occurrence may be detected by threshold crossing. Threshold crossing is applied here to the estimated first derivative of the observed signal. The advantage of this approach is the use of the filtering features of the operator Eq.(3.8). Spikes are detected according to the procedure described below.

The observed signal is considered as a mixture of noise $\xi(t)$ and a signal containing spikes $x(t)$. If the probability of spike occurrence is small enough, then the parametric values of the mixture are close to the same values of the noise itself. In such case, the values of the mixture can be used instead of the noise characteristics even if the estimation is biased.

The same consideration is true for the derivatives since its calculation is reduced to the summation:

$$D_\alpha^1 \xi(t) = \sum_{-\alpha}^{\alpha} A_i \xi(t) .$$

For the signal with additive noise $\tilde{x}(t) = x(t) + \xi(t)$ the first derivative variance is

$$\sigma_{D\xi}^2 = D_\alpha^1 \xi(t) = \sum_{-\alpha}^{\alpha} A_i \xi(t) = \sigma_\xi^2 \sum_{-\alpha}^{\alpha} A_i^2$$

So, $D_\alpha^1 \xi(t)$ is a random variable with zero mean and variance $\sigma_{D\xi}^2$.

Choosing an appropriate level of confidence, *e.g.* p=0.99, it is possible to find the threshold for spike detection $\boldsymbol{R}_{detect}$ by considering the distribution of $D_\alpha^1 \xi(t)$ as normal,

according to the central limit theorem. The obtained threshold is used to detect spike occurrences.

3.2.3 Automatic learning algorithm for spike recognition

An unsupervised learning algorithm for spike sorting is necessary in order to provide a fast and user-friendly selection tool to use during a real-time experiment or during human neurosurgery.

The traditional approach to construct templates for classification matching is to estimate the mean trajectories of clusters. However, the estimation of the mean is not robust to outliers that unavoidably appear in the learning set collected without user supervision in presence of noise and artifacts. Instead of averaging the observed trajectories, the learning algorithm selects one of them to be considered as a template representing the class. To evaluate this trajectory the algorithm analyzes the learning set to find the trajectories with maximal probability density in their neighborhood. The rationale is that for the Gaussian distribution with mathematical expectation E the value $x^* = Ex$ provides the maximum of $P(|x-x^*|<R)$ for any given parameter R.

The distribution of the squared distances from the center to all other spikes belonging to a given class was used to estimate the class radius. If the normally distributed vectors of trajectory normal deviation $\mathbf{n}(\theta)$ are independent the square of the distance from each spike to the center of its class follows a χ^2 distribution (figure 3.1).

3.2.3.1 Iterative procedure for number of classes and class centers estimation

Let us suppose that the learning set L contains only one class trajectories (figure 3.3), on the first step the initial estimation of the class center is an arbitrary spike trajectory x_0. The subset R_0 is the R - neighborhood of x_0: $R_0 = \left\{x : \left\|x - x^0\right\|_\Omega < R\right\}$.

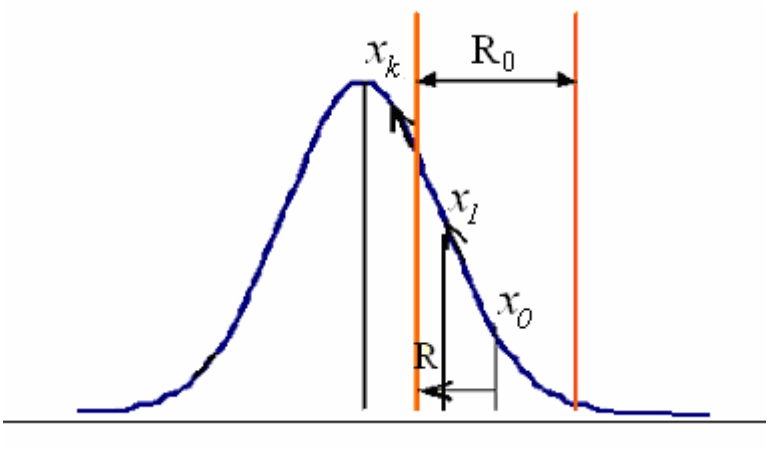


Figure 3.3.
Illustration of the iterative algorithm if the learning set consists of only one class.

The next estimation x^1 is the element of the subset R_0 minimizing the sum of distances between all R_0 elements: $x^1 = \arg\min_{x^* \in R_0} \sum_{x \in R_0} \left\| x^* - x \right\|_\Omega$. This element realizes the maximum of the probability density on R_0. On the next step, the subset R_1: $R_1 = \left\{ x : \left\| x - x^1 \right\|_\Omega < R \right\}$ is considered, and so on. Due to the symmetry and unimodality of the normal distribution, this procedure converges to the mean value on any choice of the parameter R. However, a larger training set is required for smaller values of R.

For the simultaneous search of centers of p classes (p is unknown), it is assumed that the classes are separated enough to consider that the maxima of the density of joint distributions are near the centers of the classes. The maxima of the density are the stationary points of the iterative procedure described above. Thus, it is sufficient to select the initial estimate points in the neighborhood of each maximum in order to detect all centers of the classes. The initial estimates are calculated iteratively following the next procedure described below and illustrated by figure 3.4:

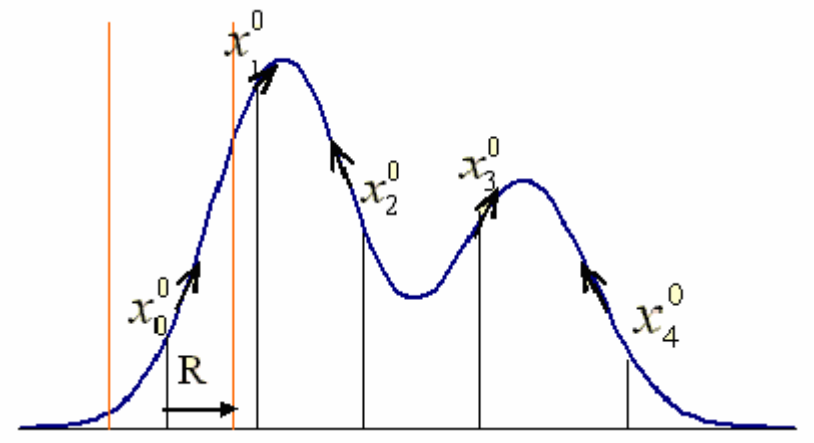


Figure 3.4
Illustration of the iterative algorithm if the learning set consists of several classes.

Firstly, an initial estimated subset L_0 of the learning set L is constructed as follows. any spike x_1^0 is considered to be the first initial estimate: then the learning set is scanned for any element x_2^0: $\left\|x_1^0 - x_2^0\right\|_\Omega > R$, where R is a threshold value. For each next element x_i^0 of the learning set the distance between this element and all previously found elements of L_0 $\left\|x_{i-j}^0 - x_i^0\right\|_\Omega > R$ for all $j = 1,...i-1$. The number p_0 of L_0 elements in the end of this process is the first estimate of the number of classes.

First iteration: after all L_0 elements were found, the learning set is scanned in the neighborhood of each of them for the elements $x_i^1 \in R_{i0}$, such that:

$$R_{i0} = \left\{x : x \in L, \left\|x_i^0 - x\right\|_\Omega < R\right\},$$

minimizing the sum of the distances to all other elements of the learning set:

$$x_i^1 = \arg\min_{x^* \in R_{i0}} \sum_{x \in R_{i0}} \left\|x^* - x\right\|_\Omega ,$$

i.e. the elements maximizing the probability density on each R_{i0}. The elements x_i^1, $i = 1,..., p_1$ form the set of the next estimates L_1.

In an analogous way, the set L_k of k-th estimates $x_i^k, i = 1,..., p_k$ is constructed at k-th iteration. At each iteration it is possible that $x_i^k = x_j^k$, i.e. the algorithm converges to the same stationary points from different first estimates. p_k is the k-th estimate of number of classes p.

Since the set of first estimates is chosen in a random way it is possible to loose one of the stationary points. Therefore, on several first iterations the L_k is completed by new elements following the same subroutine:

element $x_{p_k+1}^k \in L \setminus L_k : \left\| x_{p_k+1}^k - x_i^k \right\|_\Omega < R$, $i = 1,.., p_k$,

element $x_{p_k+2}^k \in L \setminus L_k : \left\| x_{p_k+2}^k - x_i^k \right\|_\Omega < R$, $i = 1,.., p_k + 1$, etc.

The numeric tests showed that the described iterative algorithm converges in 5-7 iterations. The stationary points of the algorithm are considered as class templates (or class centers).

3.2.3.2 Estimation of class radius

The distribution of the distances from the center to all spikes of the learning set L containing N elements is used to estimate the class radius. According to the model properties Eq.(3.4), Eq.(3.5), for the spike trajectories of class X, for any θ, the vector of normal deviations $n(\theta) = x^0(\theta) - x^i(t(\theta))$ has a distribution close to the Gaussian distribution with zero mean. We will consider the distribution of the random variable

$$\xi = \left\| x^0 - x^i \right\|_\Omega^2, \; i = 1,..., N$$

in the case where the spikes x^i are the elements of the class with the center x^0 and in the case where they are not. We will assume that the learning consists of two classes $Z = \{z^i\}$ and $Y = \{y^i\}$ with centers in z^0 and y^0. We will also assume that the variance in the both classes is the same for any θ : σ_z^2 and σ_y^2. For each θ we will consider:

$$z^0(\theta) - z^i(t(\theta)) \sim N(0, \sigma_z^2),$$

$$y^0(\theta) - y^i(t(\theta)) \sim N(0, \sigma_y^2),$$

$$z^0(\theta) - y^i(t(\theta)) \sim N(z^0(\theta) - y^0(\theta), \sigma_y^2) \text{ and}$$

$$y^0(\theta) - z^i(t(\theta)) \sim N(y^0(\theta) - z^0(\theta), \sigma_z^2).$$

If the random variables $n(\theta)$ are independent on different θ_k, $1 \le k \le M$, M - number of fragmentation points on interval $0 < \theta < T$, then the random variables

$$\zeta_z = \frac{1}{\sigma_z^2} \left\| z^i - z^0 \right\|_\Omega^2$$

and

$$\zeta_y = \frac{1}{\sigma_y^2} \left\| y^i - y^0 \right\|_\Omega^2$$

have a χ^2 distribution with mean value M and variance $2M$. The density of the χ^2 distribution has a maximum in $M - 2$, where M - degree of freedom, which is the same as the number of considered points of the spike trajectory in phase space on the interval $0 < \theta < T$. The random variables

$$\zeta_{zy} = \frac{1}{\sigma_z^2} \left\| z^i - y^0 \right\|_\Omega^2 \text{ and } \zeta_{yz} = \frac{1}{\sigma_y^2} \left\| y^i - z^0 \right\|_\Omega^2$$

have noncentral χ^2 distribution with noncentrality parameters

$$m_z = \frac{1}{\sigma_z^2} \left\| z^0 - y^0 \right\|_\Omega^2 \text{ and } m_y = \frac{1}{\sigma_y^2} \left\| x^0 - y^0 \right\|_\Omega^2,$$

respectively.

We consider the random variable $\xi = \left\| x^0 - x^i \right\|_\Omega^2$ as a mixture of random variables ζ, ζ_y, ζ_{xy} and ζ_{yx}. Let us assume that impulses of the first and the second classes appear with the same probability; then, the distribution density of $\xi = \left\| x^0 - x^i \right\|_\Omega^2$ is a half-sum of the probability densities of variables ζ_x and ζ_{xy}. If the noncentrality parameter is large

enough this density is bimodal and the first maximum is defined by the maximum of the distribution of the value

$$\zeta = \frac{1}{\sigma_x^{\,2}} \left\| \mathbf{x}^i - \mathbf{x}^0 \right\|_\Omega^2 .$$

Thus, for the random variable:

$$\frac{\sigma_x^{\,2}}{M} \xi = \frac{1}{M} \left\| \mathbf{z}^i - \mathbf{x}^0 \right\|_\Omega^2$$

the first maximum of the distribution density is approximately in the point

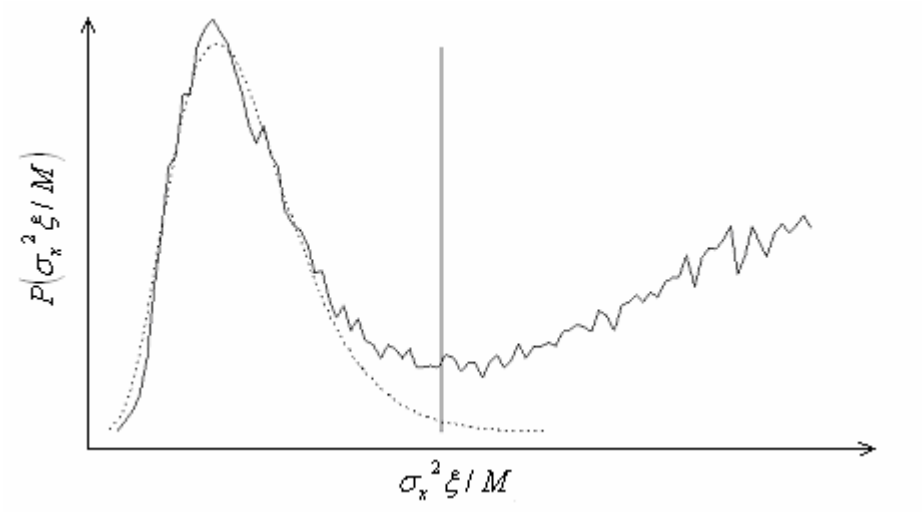
$$\sigma_x^{\,2} - \frac{2\sigma_x^{\,2}}{M} .$$

The argmax estimates the variance if we neglect the second number. The result does not significantly change for a greater number of classes.

Until now we have assumed the independency of the random variables $n(\theta)$ for different θ, which in general does not correspond to the problem statement. The random variable $\xi = \left\| x^0 - x^i \right\|_\Omega^{\,2}$ is a sum of squares of dependent Gaussian random variables $n(\theta)$, and it can be represented as a square form of some independent Gaussian random variables. Such square form has an ω^2 distribution (Martynov, 1978), with parameters depending on the covariation matrix of original dependent variables $n(\theta)$. Since this matrix is unknown, the empirical distribution of the random variable

$$\frac{\sigma_x^{\,2}}{M} \xi = \frac{1}{M} \left\| x^0 - x^i \right\|_\Omega^{\,2}$$

is approximated by a theoretical χ^2 distribution (Press et al, 1994). Quantiles of the χ^2 distribution may be used to estimate the class radiuses, corresponding to an appropriate confidential level (Fig.3.5).



Fugure. 3.5

Histogram of random variable $\frac{\sigma_x^2}{M}\xi = \frac{1}{M}\left\|x^0 - x^i\right\|_\Omega^2$ and the graph of the approximating χ^2 distribution (dot line).

3.2.3.3 Choice of the parameter for the learning procedure

The learning procedure is rather robust to the value of parameter R, but it must be bounded by the minimum interclass distance. In addition the value of the parameter R should not be too small since, in this case, we need a larger learning set.

The choice of the parameter R is based on an approximate estimate of intraclass variances σ given the assumption that they are similar for all classes. A similar assumption was made above for the class radiuses estimation. Here we consider the distribution of the random value

$$\xi = \left\|x^i - x^j\right\|_\Omega^2,\ i \neq j; i = 1,..., N;\ j = 1,..., N$$

in the case where the elements of the learning set x^i and x^j belong to the same class and in the case where they do not. Firstly, we assume that the vectors $n^{ij}(\theta) = x^i(t(\theta)) - x^j(t(\theta))$ are independent for different θ.

If in addition we assume that impulses of the first and the second class appear with the same probability, the distribution of variable $\xi = \left\|x^i - x^j\right\|_\Omega^2$ is a mixture of two χ^2 distribution and two noncentral χ^2 distributions. The first maximum is realized

approximately at point $2\sigma^2 - \frac{4}{M}\sigma^2$. Even if the class variances are different, we can estimate the minimal one.

For the parameter R we take the value

$$\sqrt{2\sigma^2 - \frac{4\sigma^2}{M}}.$$

If M is large enough, i.e. $R \sim \sqrt{2}\sigma$, which corresponds to the confidential level 0.84 for normal distribution.

Since the assumption of independency of random variables $n^{ij}(\theta)$ for different θ in general is not true, the empirical distribution of $\frac{1}{M}\left\|x^i - x^j\right\|_\Omega^2$ was approximated with a theoretical χ^2 distribution (Press et al., 1994). The argmax of the theoretical distribution was used as R, i.e. the learning procedure parameter.

3.2.4 Real-time classification

The decision function $d_i(x) = p_i p\left(x \middle| X_i\right)$ (Gonsales, 1978) was used as classifier for class X_i. p_i is the class probability:

$$p_i = \frac{\left|\left\{x : x \in L, \left\|x^i - x\right\|_{\Omega'} < R^i\right\}\right|}{\sum_{j=1\ldots p}\left|\left\{x : x \in L, \left\|x^j - x\right\|_{\Omega'} < R^j\right\}\right|},$$

where $|*|$ - number of set elements, p - number of classes, L - learning set. Given the assumption that random variables $n(\theta)$ are independent for different θ, and the variance does not depend on θ, the classifier is Bayesian (Gonsales, 1978):

$$\ln p_i - \frac{1}{2}\ln\sigma_i - \frac{1}{2\sigma_i^2}\left\|x^i - x\right\|_{\Omega'},$$

where σ_i is the variance of class X_i, x^i is the class center. Since the assumption of independency of random variables $n(\theta)$ for different θ does not correspond to the problem statement, for estimation of $p(x|X_i)$ we used the quantiles of χ^2 distribution, approximating the empirical distribution of variable $\frac{1}{M}\left\|x^0 - x^i\right\|_\Omega^2$ on the learning set.

Spike $x(t)$ is classified into X_i, if X_i realizes the maximum of the decision function. The probability of erroneously detected noise on the spike detection stage is non zero, such that it is necessary that the distance between the trajectory of the spike and the class center satisfied also the condition $\left\|x^i - x\right\|_{\Omega'} < R^i$, where R^i is the class radius estimated while learning. Otherwise the spike $x(t)$ is rejected.

3.2.5 Choice of the algorithm parameters

For the realization of the described algorithm a number of parameters should be specified: the order n of the differential equation (3.1), the kernel of operator D_α^k for derivatives calculation and the regularization parameter α.

3.2.5.1 Form and order of the differential equation

The use of equations of a high order decreases the speed of the algorithms since it increases the feature space dimension. An equation of second order provides satisfactory results (the efficiency evaluation of the algorithms is described in chapter 4). Then, because of the high level of noise we used the equation:

$$\frac{d^3x}{dt^3} = \tilde{f}\left(\frac{dx}{dt}, \frac{d^2x}{dt^2}\right)$$

instead of the equation:

$$\frac{d^2x}{dt^2} = f\left(x, \frac{dx}{dt}\right).$$

3.2.5.2 The kernel of the integral operator

The kernel of the operator D_α^k Eq.(3.8) used for the numeric differentiation may be any function satisfying the conditions: $\omega_\alpha = 0$, then $|t| > \alpha$; $\int_R \omega_\alpha(\tau - t)dt = 1$; ω_α has k continuous derivatives. Using certain piecewise polynomial kernels (Aksenova et Shelekhova, 1997; Aksenova et Shelekhova 1995) it is possible to provide an efficient numerical differentiation of noisy signal, which is particularly important for the real-time algorithms.

For the piecewise polynomial kernels the construction of operators D_α^k, the carrier $|t| < \alpha$, is divided into fragments. On each fragment the kernel is represented by a polynomial function of $k+1$ order. The kernel must be a k times continuously differentiable function and the fragments must be multiple to the digitalization fragments. Thus the kernel is unambiguously defined by the order of the differentiation and by the fragmentation of the kernel carrier. For the selected model we need the first and the second derivatives estimation. For kernels definition a regular fragmentation was considered the following kernel $\omega_\alpha(t)$ on $t \in [-\alpha, 0]$ has differentiation of second order:

$$\omega_\alpha(t) = \begin{cases} \frac{16}{3\alpha}\left(\frac{1}{\alpha}t + 1\right)^3, & t \in [-\alpha, -\frac{3}{4}\alpha] \\ -\frac{16}{3\alpha}\left(\frac{1}{\alpha}t + \frac{1}{2}\right)^3 + \frac{2}{\alpha}t + \frac{3}{2}, & t \in [-\frac{3}{4}\alpha, -\frac{1}{4}\alpha] \\ \frac{16}{3\alpha}\left(\frac{1}{\alpha}t\right)^3 + 1, & t \in [-\frac{3}{4}\alpha,] \end{cases}$$

$\omega_\alpha(t)$ on $t \in (0, \alpha]$ is symmetrical. The kernel for the first and the second derivative estimation are the first and the second derivatives of $\omega_\alpha(t)$ respectively, where the kernel for the first derivative estimation is a second order polynomial. To simplify the procedure we constructed another kernel function $\omega_\alpha^*(t)$ which assures only first order derivative estimation, but the $\omega_\alpha^{*(1)}(t)$ is a polynomial of first order:

$$\omega_\alpha^*(t)=\begin{cases}\frac{2}{\alpha}(\frac{t}{\alpha}+1)^2, t\in[-\alpha,-\alpha/2]\\ -\frac{2}{\alpha}\left(\frac{t}{\alpha}\right)^2+\frac{1}{\alpha}, t\in[-\alpha/2,\alpha/2]\\ \frac{2}{\alpha}(\frac{t}{\alpha}-1)^2, t\in[\alpha/2,\alpha]\end{cases}$$

Thus we used kernels $\omega_\alpha^{*(1)}(t)$ and $\omega_\alpha^{(2)}(t)$ for the derivatives estimation (Fig. 3.4).

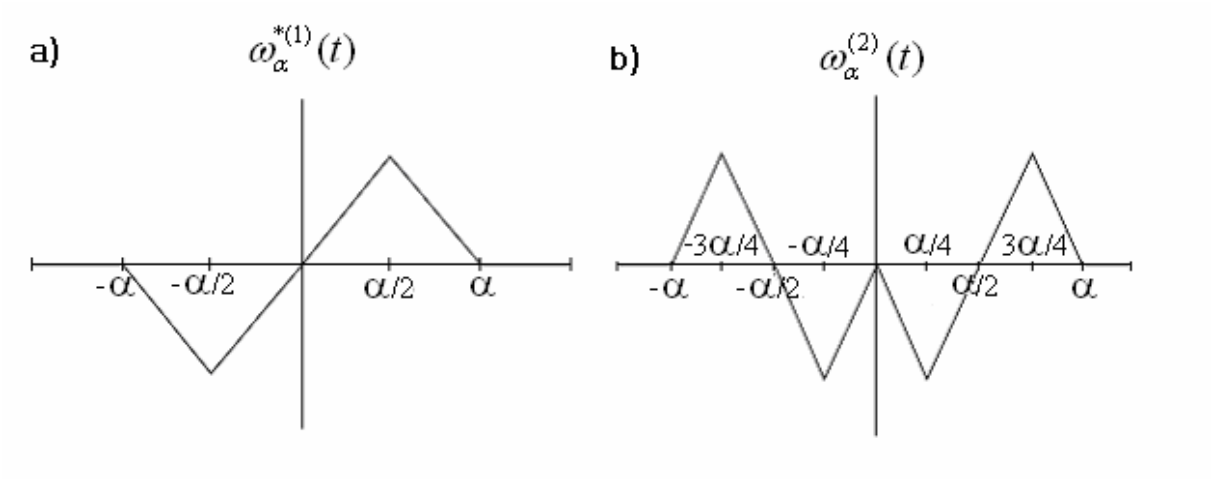


Figure 3.6:

Kernel functions $\omega_\alpha^{*(1)}(t)$ (a) and $\omega_\alpha^{(2)}(t)$ (b) used for the first and the second derivative estimation.

3.2.5.3 Regularization parameter of the integral operator

For the realization of fast numerical differentiation (Aksenova et Shelekhova, 1997; Aksenova et Shelekhova 1995) the regularization parameter α must be a multiple of the digitalization fragments Δt, *i.e.* for the selected kernel function parameter α must by multiple of $2\Delta t$ for the first derivative and multiple to $4\Delta t$ for the second derivative.

We considered the integral operator as a band filter

$$y(t)=\int_R h(t-\tau)x(\tau)d\tau \ .$$

The spectral densities of the input and the output signal are relied as following (Korn et Korn, 1984):

$$\Phi_{yy}(w) = \left|\mathrm{H}(iw)\right|^2 \Phi_{xx}(w),$$

where

$$\mathrm{H}(iw) = \int_R h(\varsigma) e^{-iw\varsigma} d\varsigma \ .$$

For the applied operators of numerical differentiation

$$h(\varsigma) = (\omega_\alpha(t))^{(k)} = \omega_\alpha^{(k)}(t),$$

where k is the derivative order. Using the kernel presented on figure 3.6 (a) for the estimation of the first derivative if $\alpha = 1$ (fig.3.7):

$$\left|H_1^1(iw)\right|^2 = c_1^1(w) = \frac{64}{w^4}\left(\sin w - 2\sin(0.5w)\right)^2 .$$

The transition function $c_1^1(w)$ reaches its maximum for $w \approx 2.7$. For an arbitrary α:

$$\left|H_\alpha^1(iw)\right|^2 = c_\alpha^1(w) = \frac{1}{\alpha^2} c_1^1(\alpha w),$$

and the maximum is obtained for $w \approx 2.7/\alpha$.

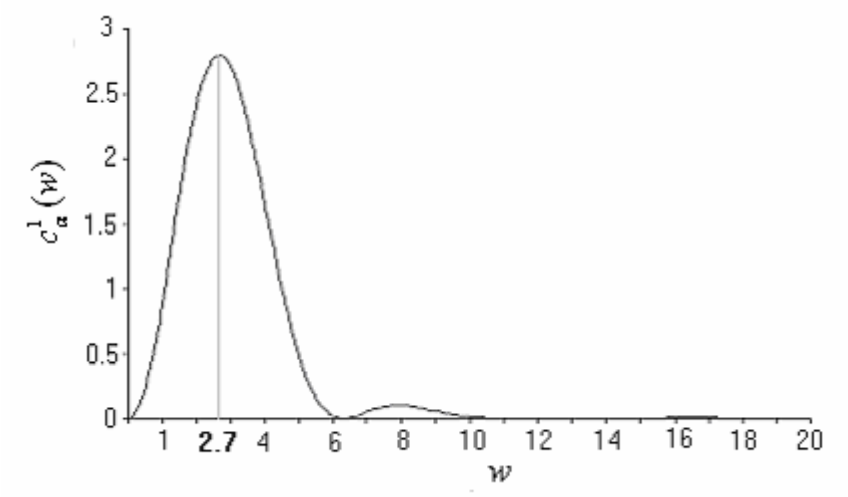


Figure 3.7:

Transition function $c_1^1(w)$ for kernel $\omega_\alpha^{*(1)}(t)$ for the first derivative estimation.

Normally, the spike duration oscillates near 1 ms. In the frequency domain it corresponds to frequency $\nu = 1\,\text{kHz}$. The value of the regularization parameter was chosen so that the argmax of the transition function $c_\alpha^1(2\pi\nu)$ coincided with this frequency, *i.e.* $\alpha \approx 2.7/2\pi\nu$. If $\nu = 1$ then $\alpha \approx 0{,}43$. Thus, for the first derivative estimation for parameter α we used a value which is near $0{,}43$ and multiple to two digitalization fragments.

A similar layout can be carried out for the kernel $\omega_\alpha^{(2)}(t)$ used for the second derivative estimation (fig.3.6 (b)). If $\alpha = 1$:

$$\left|H_1^2(iw)\right|^2 = c_1^{(2)}(w) = \frac{64^2}{w^4}\left(-1 + 2\cos(0.25w) - 2\cos(0.75w) + \cos w\right)^2.$$

The maximum of the transition function is reached for $w \approx 3.8$ (figure 3.8). The regularization parameter should be chosen near the value 0.6, and it must be multiple to four digitalization fragments.

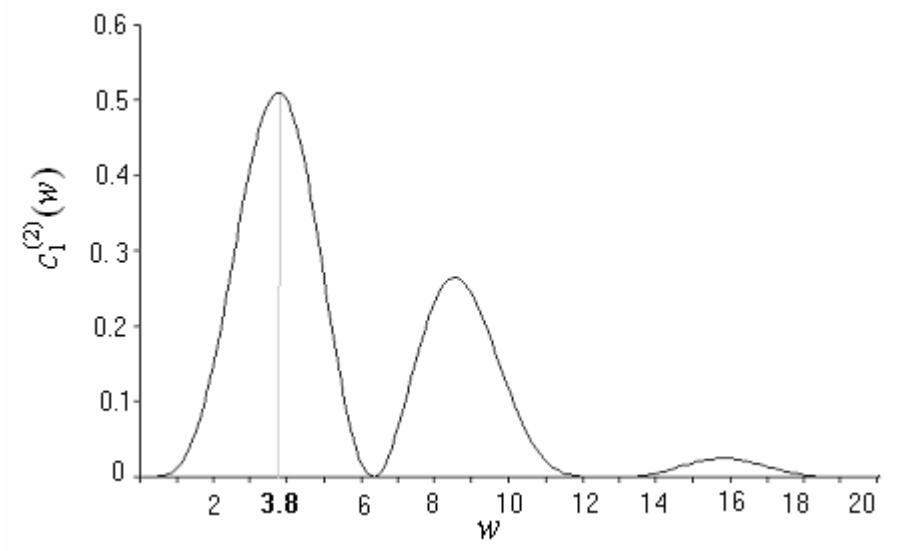


Figure 3.8:
Transition function $c_1^1(w)$ for kernel $\omega_\alpha^{(2)}(t)$ for the second derivative estimation.

Chapter 4
USS Software

Un logiciel permettant le tri de potentiels d'action non supervisé a été développé à partir des algorithmes décrits dans le chapitre précédent. Ce chapitre expose la structure générale du logiciel, le traitement du signal électrophysiologique, les procédures principales et l'interface utilisateur.

Le logiciel se compose de trois bloques : les module de calcule, le bloque de lecture de donné et l'interface utilisateur. Il est nécessaire d'appliquer une procédure d'apprentissage avant le tri proprement dit des potentiels d'action. La procédure d'apprentissage construit de manière automatique les étalons représentatifs de potentiels d'action. L'apprentissage est rapide, 6-15 secondes selon la puissance de l'ordinateur, à partir de la fin de l'enregistrement des signaux.

Pour procéder au tri de potentiels d'action l'utilisateur peut visualiser le signal, détecter les potentiels d'action et effectuer le tri en temps réel. A la fin de la procédure l'utilisateur peut néanmoins intervenir pour rectifier les résultats d'apprentissage.

On the basis of the algorithms described above a crossplatform software was developed. It permits an unsupervised spike sorting (USS) with minimal interaction with the user on the learning stage. The user is nevertheless allowed to interact in any stage of the signal processing. The USS software consists of three blocks: Data Acquisition Block, Calculation Modules and User Interface (figure 4.1). The Calculation Modules are developed using the C++ programming language. Their code can be compiled by Microsoft Visual Studio 6.0 (MicroSoft Corp., Redmont, WA) running on Microsoft Windows 2000/XP (MicroSoft Corp., Redmont, WA) operating system and compiled as dynamic libraries by CodeWarrior (Metrowerks, Austin, TX) for Macintosh (Apple Computer Inc., Cupertino, CA) platforms with MacOS 9.x (Apple Computer Inc., Cupertino, CA) operating system. The user Interface was developed and compiled with LabView 6.0 (National Instruments, Austin, TX). The output files corresponding to the spike trains are in standard ASCII format for multivariate time series, originally proposed by Professor M. Abeles at the Department of Neurophysiology of the Hebrew University, Jerusalem.

The software allows off-line analysis of prerecorded data. Analog files corresponding to the electrophysiological signals in standard formats like WAV and AIFF, as well as customized formats with defined sampling rates and bits resolution, are readable by the program. The Calculation Modules process the data stream independently of the data source. The spike sorting speed corresponds to a real-time process.

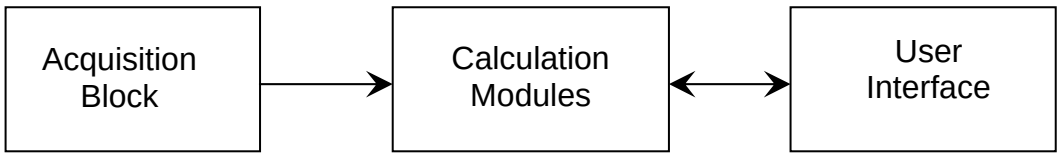


Figure 4.1
USS software main blocks

4.1 Software schemes

The program offers three operating modes: signal visualization without any processing, learning set collection and classification. The data stream is recorded in a buffer and the buffer is processed according to the operating mode. The signal visualization operating mode displays the raw signal and the first order derivative of the signal. This visual inspection can be useful for modifying the default values of the level of spike detection threshold and regularization parameter for the derivative calculations.

The user may determine whether the learning phase starts after a minimum interval of acquisition or after a minimum number of suprathreshold events has been collected. Learning procedure is lunched automatically after the learning set is collected. For a learning set of 300 spikes the learning process takes about 10-15 seconds on a personal computer equipped with a Pentium3 500 MHz (Intel Corp., Santa Clara, CA) and about 6-8 seconds on a Macintosh PowerPC G4 400 MHz (Apple Computer Inc., Cupertino, CA). At completion of the learning procedure the program is back into the visualization operating mode and ready for spike sorting.

The program works in real-time in signal visualization mode, learning set collection mode and classification operating mode. The learning procedure is an off-line procedure. Here is a flowchart of the software.

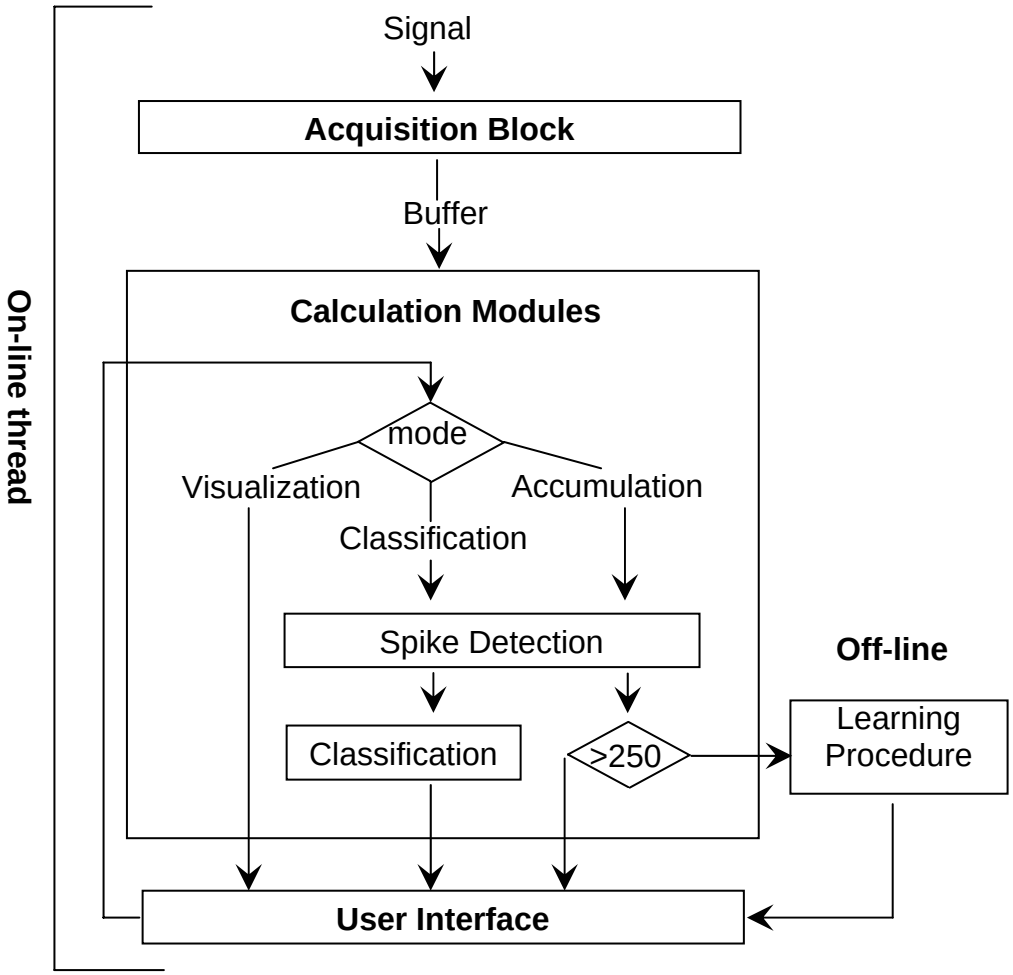


Figure 4.2
Flowchart of the electrophysiological signal processing by the USS. The user switches between operating modes of "signal visualization", "learning set accumulation" and "spike sorting" via the user interface block. While the software works in one of these modes the data coming from the acquisition block is processed on line. The only off line module of the USS is the learning procedure which is launched automatically then the learning set accumulation is over.

The iterative learning procedure used in USS converges to the local maximums of probability density not always corresponding to distinct classes. Then not all the templates represent a cluster and several templates may represent one class. Some post processing is necessary, starting with the elimination of overlapped classes. Classes are formed by all spike trajectories in *R*-vicinity of the template trajectory. If more then *p* percents of the trajectories of one class belong in the same time to another class, both classes are considered to represent the same neuron with high probability and the less representative class is eliminated. By

default p = 80% but the user is allowed to change this parameter. Some low amplitude classes may appear due to background noise, and non-representative classes due to artifacts. To eliminate them automatically we used the criterion (a/A) × (n/N) < k, where a is the amplitude of the class template, A is the maximal template amplitude, n is the number of class members, N is the learning set volume, k is a parameter. The parameter k is by default equal to 0.12; this value can be changed by the user.

After the post-processing the results are displayed graphically to offers the possibility to select only specific classes. All the information about selected templates is saved into a file, so that the templates could be loaded for later usage. After the selection of templates the program is ready to perform the on-line classification of spikes. The suprathreshold events are either classified in one of the selected classes and displayed in a classification window (figure 1(b)) or rejected. The epochs of the events of all classes are recorded as a multivariate time series. By default the time resolution of time series is set to 1 ms.

4.2 User interface

4.2.1 Main window

When launched, the USS loads the main window with menu. The files containing the recorded signals can be loaded from the menu Files. The software can process a signal recorded from one electrode channel. The commands for data processing are available from the menu item Operate. All the parameters for the processing are set by default or calculated automatically. The user is also allowed to fit them, if necessary, from the menu Operate->Parameters or directly via the main window controls. From the menu item View it is possible to adjust the signal amplitude to the output graph dimensions. From the menu item Help the information about the current version is available.

4.2.1.1 Main window controls

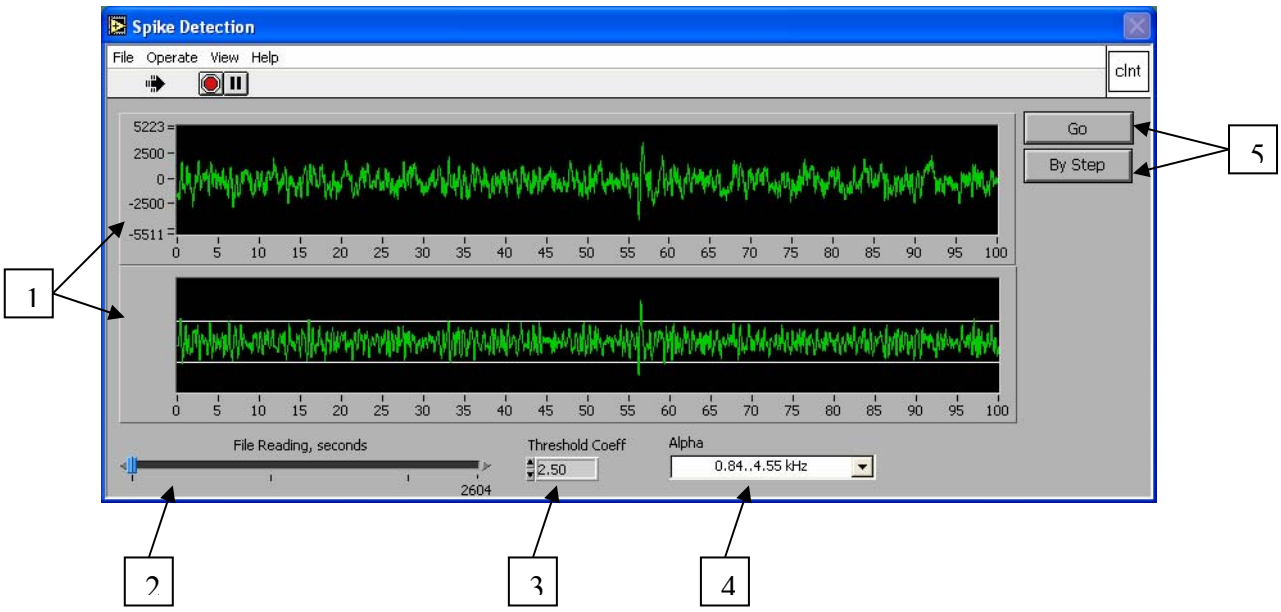


Figure 4.3
USS main window

Here is a short description of the main window controls (figure 4.3):

1. Display of the signal (above) and its derivative (below). The signal is visualized by buffers of 100 ms by default. User may change the buffer length.
2. This control allows moving over the file. The scale is from 0 to the length of the recording in ms.
3. Spike detection threshold adjusting control. A spike is detected if the signal derivative crosses a threshold value calculated as k*σ, where σ is the standard deviation of the derivative, k is a coefficient equal to 2.5 by default. The user can change this coefficient. The lines on the derivative graph correspond to the current threshold levels.
4. Band pass filtering. The operator used for derivative calculation acts as a band pass filter. The regularization parameter of the operator α is set to cut off the frequencies over the selected band (0.84-4.55kHz is set by default in order to filter below and above spike frequencies). The user can choose another filtering from the pop-up menu.

5. “Stop/Go” and “By Step” buttons. The button “Go” starts the visualization and its caption switches to “Stop”. By pressing it again the user can suspend the visualization (the button caption changes again to “Go”). Once suspended, the visualization can be performed buffer by buffer by pressing the button “By step”.

4.2.1.2 Main window menu

Loading Files. Data files, as well as some additional files, are loaded from the main window menu File:

File->Open; File->Close

Data can be read from files of the following formats: AIFF, VAW, MAP and from two byte integer binary files. The binary files are supposed to correspond to one channel recording with a sampling rate of 48000 Hz. For the other formats the information about the count of channels and the sampling rate is read from the file header. If more then one channel is recorded in the file, a select channel dialog appears on the monitor (figure 4.4). After the channel has been selected the visualization is started. Before opening the next file the current one should be closed.

Figure 4.4
Channel selection dialog.

Open additional files. Some files for additional data processing or files keeping the information of previous processing can be loaded**.**

File->Template files

To proceed with the unsupervised spike sorting the learning phase should have been completed. This step can be skipped by loading class templates from a pre-recorded file.

The template files are internal USS files. The format is described in the following sections. They are created automatically then the user starts the detection after the learning procedure. After loading a template file the program is ready for spike sorting.

File->Superimpose

This option was added because it could be necessary to superimpose noise or artifacts on the analyzed signal to create test files. Noise can be added to obtain signals with different signal-to-noise ratio. Artifacts may be superimposed to a signal file to simulate a signal with stimulation artifacts. The file for the superposition should be a two bite integer binary file.

Operating. From this menu item the user operates the software and switches between the operating modes.

Operate->Parameters

After loading all necessary files the user can change the parameters and start the learning phase. Some of the parameters can be changed through the main window (see main window controls). The other parameters can be changed through the Parameters dialog box which is accessible from the menu Operate->Parameters. It is recommended to set all the parameters before starting the signal processing. From this dialog box the user can change the following settings (figure 4.5):

1. “Training Start Time” fixes the time from the beginning of the file to start the learning set accumulation. By definition it is 0 and the leaning set accumulation is started as soon as the user selects the menu item Operate->Training.
2. “Detection Start Time” fixes the time from the beginning of the file to start spike sorting. By definition it is 0 and the spike sorting is started as soon as the user selects the menu item Operate->Detection.
3. “ms Par Bf” is the length of the buffer visualized on the main window in ms.
4. “Spike duration”. The spike duration may vary slightly such that the length of the segment to be extracted as spike could be fitted by the user.
5. “Accumulation Spike Amount” is amount of spikes to accumulate for the learning set. It is not recommended to set this value over 350 spikes because the time for the

learning procedure increases significantly. The learning procedure works properly on 250-300 spikes learning set.

6. "Detection Radius Coeff" is the coefficient to fit the class radiuses during classification. The learning procedure calculates class radiuses automatically. If the user is not satisfied with the spike sorting results, the radiuses fit can be adjusted by this coefficient. The radiuses are multiplied by the coefficient.

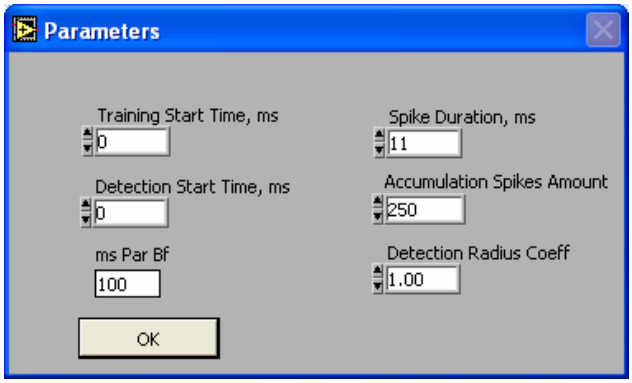


Figure 4.5
Parameters dialog box

Operate->Training

Starts the collection of spikes detected from the current signal and launches the learning procedure.

Operate->Detection

Starts the spike sorting if the templates are already defined either by the previous learning or loaded from a file. Otherwise the system stays in the visualization mode.

4.2.2 Learning procedure results window

To start the learning set accumulation the user selects the Operate->Training menu item. Then the learning set collection is started. After completion of the learning set accumulation the learning procedure starts automatically. The results of the procedure are visualized by the results window (figure 4.6).

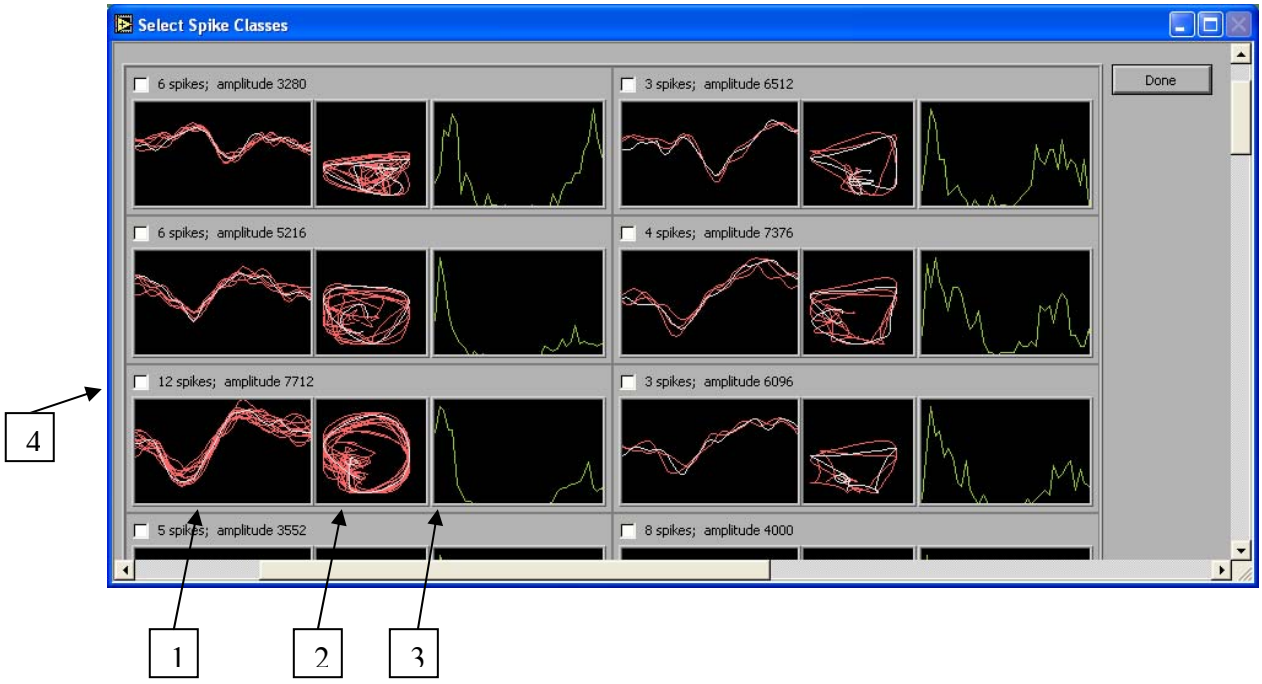


Figure 4.6
Learning window

This window presents all detected classes. For each class there is a graph (1) for the visualization of the spikes extracted from the signal, and a graph (2) representing the phase portraits of the spikes. On the third graph (3) the histogram of distances between the center of the class and all other spikes from the learning set is shown.

The user selects the classes to analyze by marking the corresponding check boxes (4). After selecting the classes the user presses the Done button. If the user is not satisfied with the result, then the training may be computed again by selecting the Operate->Training menu item. The user should consider the following particularities of the USS software with respect to the learning procedure:

1) The results of the training procedure are only saved when the spike sorting is started. If the training is not followed by the spike sorting the selected templates are not saved.

2) The training procedure is tuned to show rather several templates that could correspond to the same observed neuron rather then losing separate neurons by merging templates too far. Among the templates proposed at the end of the training procedure there may

be several templates corresponding to one observed neuron. This feature can be fitted in the program code (see Appendix A).

3) If the results of the training procedure represent too many templates corresponding to one observed neuron, it is recommended to increase the "Detection Radius Coefficient" on the parameter window before starting the spike sorting.

4.2.3 Spike sorting progress window

Then user selects the menu item Operate->Detection and the spike sorting window appears (figure 4.7).

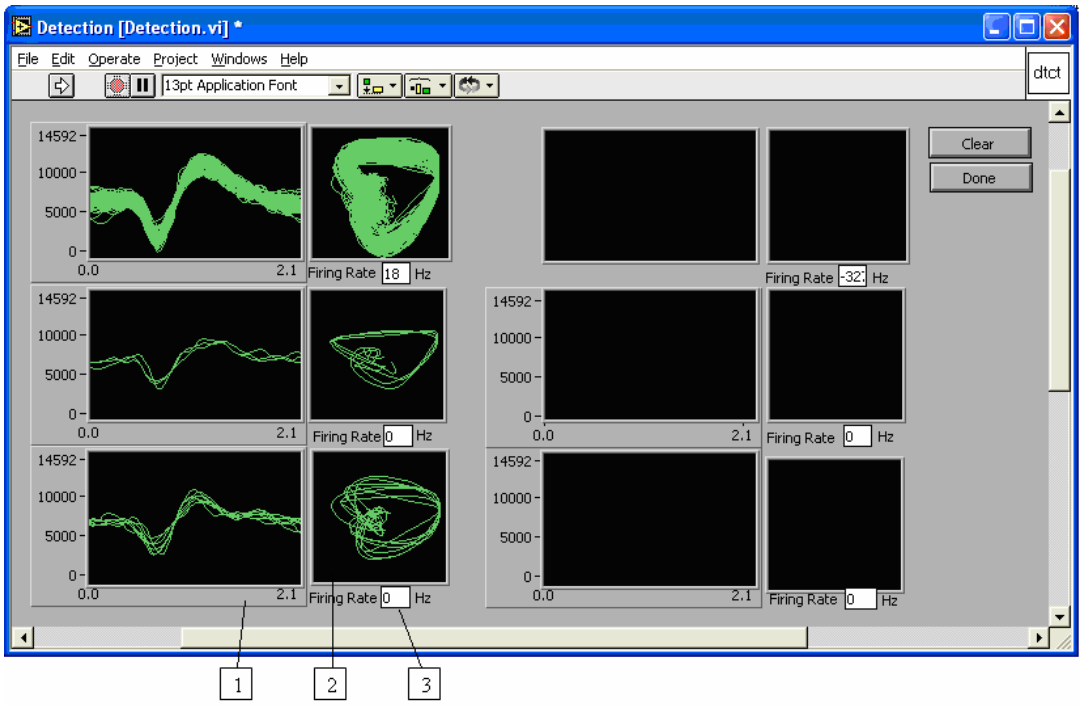


Figure 4.7
Spike Sorting window

This window displays the classification process. While spikes extracted from the signal are being classified they are visualized on the spike sorting window on the appropriate graph (1) as well as their phase portraits (2) and their firing rate (3). The user can visualize also the rejected (non classified spikes) in a separate window. If the spike sorting results do not seem

satisfactory, the user can stop the classification, change the parameters through the main window and through the parameter dialog box (figure 4.5) or repeat the learning and then restart the classification at any time.

Spike sorting starts either after the learning or after providing the templates loaded from a file. The “select file” dialog box appears and the user selects a file for output. The extension ABL is added to the file. In the same time another file with the same name and extension TMPL is created and all the templates are saved in this files (such file is not created if the templates were loaded from an external file).

4.3 File Formats

4.3.1 USS input files

USS input files are:

- Standard audio formats AIFF and WAV
- Format MAP used in Alpha Omega AlphaMap Software (see Appendix B).
- Two byte integer binary files containing data samples only.

4.3.2 USS output files

The output files corresponding to spike trains are in standard ASCII format for multivariate time series, originally proposed by Professor M. Abeles at the Department of Neurophysiology of the Hebrew University, Jerusalem. The files have extension ‘abl’ and starts with optional headings included into “” containing information of treated data. The data part of abeles files consists of series of triples divided by inline space.

<event type>, <event number>, <time from previous event>

<event type> used by USS: 1 - for spikes, 51 – for seconds counter

<event number> - template number to which belongs the spikes or zero if the spike is not identified; always 1 for second counter

<time from previous event> - appears in ms.

Each series begins with 0,1,0 and ends with 0,2,0. 0,FFFF,0 means end of file.

Example:

```
"Channel 1"
"start time 13300"
0,1,0
1,1,52
1,3,88
…
1,3,30
51,1,65
1,4,91
…
1,3,16
1,2,4
0,2,0
0,FFFF,0
```

4.3.3 USS internal files

To save the results of learning the USS software creates files with extension ‘tmpl’. These files are used to store the information about signals already processed and can be reused in off-line mode. The data in ‘tmpl’ files is organized as following

```
int32 Number of templates
int32 Template length
```

For each template:

single (2bytes)	radius (value used for classification)
int32	extention1 (coefficient of normalization for first derivative)
int32	extention2 (for second derivative)
int16[Template length]	row signal
int32[Template length]	first derivative
int32[Template length]	second derivative

Chapter 5

Validation

La validation et l'évaluation de la performance de l'USS sont effectuées en trois étapes. La première étape est la vérification de conformité au modèle utilisé. Le test χ^2 de Pearson est utilisé pour la vérification de l'hypothèse de normalité des distributions des trajectoires de potentiels d'action dans l'espace de phase Eq.(3.3).
A la deuxième étape la performance de la méthode est évaluée sur des signaux simulés avec différentes proportions signal/bruit générés a partir de 3 signaux connus. Trois types d'erreur ont été considérés : non détection, classification erronée et fausse alarme. La même procédure est appliquée aux signaux précédemment traités par un algorithme de suppression de bruit.
La troisième étape est la comparaison de l'USS avec d'autres méthodes de tri de potentiels d'action.

5.1 Model verification

If the neuron firing follows the assumptions made in chapter 2 Eqs.(3.1-3.3), the distribution of spike trajectories in phase space must be close to Gaussian. For the model verification it is sufficient to apply a test of normality of distribution for real single unit spike trajectories. The real extracellular signal, selected for the verification, represented a well isolated single unit activity. For all detected spikes the trajectories in phase space Eq.(3.3) were estimated as it is described in chapter 3, section 3.2.1. Then, one of the trajectories was selected as the class template $x^0\left(t(\theta)\right)$ by means of the iterative procedure developed for USS, described in section 3.2.3. On this trajectory an arbitrary point $x^0\left(t(\theta^0)\right)$ was fixed and the normal to trajectory x^0 hyperplane was constructed to this point. Then the points of intersection of all spike trajectories with the normal hyperplane were found. The template trajectory is considered as an estimation of the mathematical expectation. Thus, in case of good conformity of the model the distribution of vector $\mathbf{n}(\theta^0) = x^i\left(t(\theta^0)\right) - x^0\left(t(\theta^0)\right)$ should be close to the normal with zero mathematical expectation. For the verification of the

hypothesis of normality of vector $\mathbf{n}(\theta^0)$ distribution the Pearson χ^2 test of likelihood with 10 degrees of freedom was applied to a set of 60 observations. The test showed that the hypothesis can be accepted with significance level S*.

5.2 Evaluation of performance

5.2.1 Test files

Analog recordings distributed by Alpha-Omega Inc. (Nazareth, Israel) corresponding to a mixture of three single units discharges were digitized at a 50 kHz sampling rate and used as raw files. USS was applied to these recordings and extracted three spike templates (Fig.5.1).

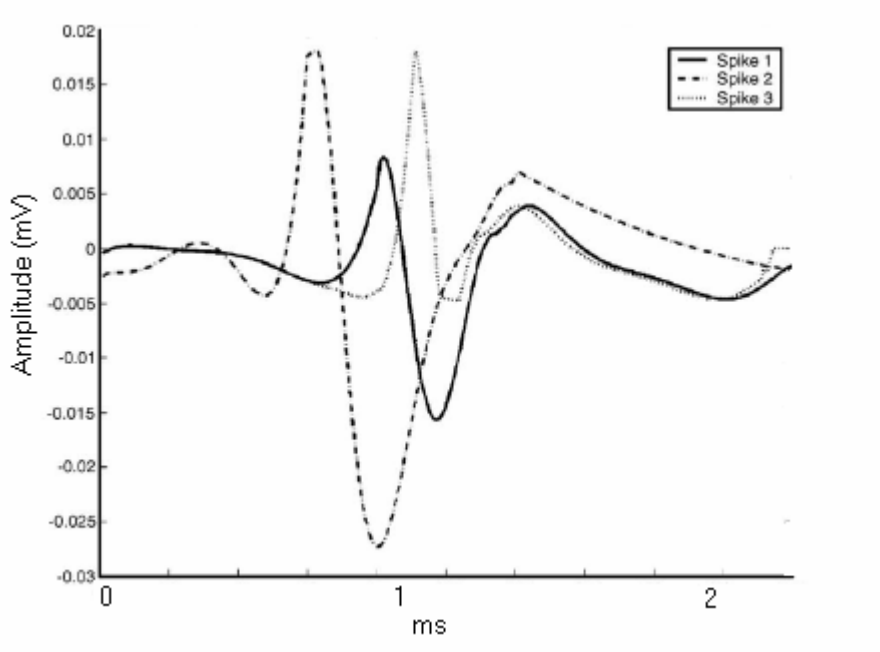


Figure 5.1
Spikes of the three single units used for the simulation of the electrophysiological signal.

Test files were obtained by random insertion of these three templates in files with different levels of background noise. The noise was extracted from the electrophysiological recordings performed in the STN during surgery on a Parkinsonian patient at the University Hospital of

Grenoble. Two levels of signal-to-noise ratio (SNR) were used for testing: SNR=2.5 dB and SNR=3.55 dB. For each level of noise the classification scores of USS were computed for two sets of data. The first set is the test file without pre-processing, while the second set is the test file processed with a Time and Spatial Adaptation (TSA) denoising procedure before undergoing USS analysis. The TSA denoising procedure is a wavelet based technique that has been originally proposed for denoising speech signals (Bahoura and Rouat, 2001). It is based on the principle of time and spatial adaptive thresholding of the wavelet coefficients. TSA was selected for the tests because it requires neither an explicit estimation of the noise level nor the a priori knowledge of the SNR, which is usually needed in most of popular enhancement methods. The originality of TSA resides in its ability to modulate in time and scale the wavelet thresholds depending on the noise and on the signal time evolution.

5.2.2 Tests

A raw file formed by 1003 events was generated after introducing three template spikes at known random times. Two levels of additional noise (SNR = 2.5 dB, SNR = 3.55 dB) were superimposed to the raw file in order to obtain two test files. The test files were also tested after pre-processing by a TSA denoising procedure described above. The performance of USS with different test files was determined according to three types of error due to an incorrect classification. The first type of error is “false alarms”, i.e. the detection of events that were absent in the original unprocessed file. The rate of false alarms can be computed exactly as we know the exact epochs of all spikes in the test file without noise and without applying denoising pre-processing. The second type of error is defined as “non-detection”, which corresponds to missing an event present in the original unprocessed file. The third error is “misclassification”. It means that a spike was detected but incorrectlyg classified.

The comparison of USS performance without any noise and with a background noise typical for real data recordings in surgery room (SNR 2.5 - 3.55 dB) showed that about 3% - 8% of spike occurrences were missed but no misclassification was observed (table 5.1). The application of TSA, a wavelet based technique for denoising, did not improve USS performance. This result was rather surprising but can be understood as the TSA denoising should also be applied to the prototype’s shape. These results suggest that TSA modifies spatiotemporal cues of the signal that are used by the USS procedure to perform the spike sorting. With or without preprocessing the number of false alarms generated by USS is very

low (equal to 0 for SNR of 3.5 dB and 1.5% for recording conditions close to those of the surgery room). Conversely, denoising increased the error due to missed detections up to nearly 10%.

Error type	Test signal without noise	SNR=2.5 dB	SNR=3.55 dB	SNR=3.55 dB with denoising
Misclassification	0	0%	0%	0%
Faulse alarme	0	1%	0%	0%
Nondetection	0	8%	3%	10%
Total	0	9%	3%	10%

Table 5.1
Results of the test spike sorting on the simulated signals with different SNR, with and without a previous denoising.

5.3 Comparison with other algorithms

The results of the performance evaluation showed that the USS provides a rather efficient single unit separation. In addition to that evaluation a comparison of the USS with some other available algorithms was carried out. The signals for the performance comparison were simulated as proposed by Letelier et Weber, 2000. Inverse Fourier transform was used to generate a trial lasting 1000 seconds with an artificial noise at the same level as indicated originally (Letelier et Weber, 2000). For spike imitation the same three spike templates were used (fig. 5.1). 10,000 instances of each template were added in a random way to the background noise avoiding template overlapping. The resulting data set mimicked three neurons firing independently at an average rate of 10 spikes per second.

The algorithm was trained using the simulated spike train and 3 classes were extracted. After spike detection and sorting we compared the epochs of the detected spikes with those of the actual simulated spike train. The following error index was used to estimate a generalized error:

$$I = \sqrt{\sum_{i=1}^{N} (e_1^i)^2 + \sum_{i=1}^{N} (e_2^i)^2} \ . \qquad (5.1)$$

Here $N = 3$ - number of classes, e_1 and e_2 – number of nonclassified and misclassified spikes for each class. The procedure of defining the templates and spike classification were repeated five times and the results from five realizations were averaged.

The comparison of the performance of our method with the Wavelet Transform Coefficients method and with Principal Component Analysis and Reduced Feature Set is presented in table 5.2. These methods are wildly used for neurophysiologic studies. For each of them the error index was calculated in the same way. All the three methods showed an error index greater then the USS.

Algorithm	Error index	
	Mean value	Standard deviation
Wavelet Transform Coefficients	36	7
Principal Component Analysis	138	2
Reduced Feature Set	89	3
USS algorithm	13	4

Table 5.2
Mean error indexes (formula 5.1) and its standard deviation for spike classification of a simulated test signal, containing spikes of three kinds. The values were obtained after 5 repetitions of the test spike sorting for each algorithm.

During the evaluation the classification of overlapped spikes was not considered. The test signals did not contain overlapped spikes. However the problem is rose how much spike overlapping decreases the performance of the USS method. The percentage of the overlapping spikes depends on the firing rate of the neurons and on the time duration of the spikes. Given a firing rate of 30 spikes/second and spike duration equal to 2 ms (Letelier et Weber, 2000) one may expect 18% of overlapping spikes. This means that the error index (table 5.2) for our method would increase up to 30% if all overlapping spikes were not sorted correctly by our program, but even in this case the USS algorithm would perform better than the other tested methods.

It should be also mentioned that the test protocol used in Letelier et Weber, 2000 was not aimed to detect spikes but only to sort them. This may have lead to overestimate the performance of other methods compared to USS. This procedure does not take into account the number of spikes that could be lost during the spike detection procedure as well as the number of noise fluctuations that could be detected as spikes. Different spike detection procedures may be associated to spike sorting. It is therefore difficult to estimate their impact on the error index reported in other studies.

5.4 Discussion

The extracellular recording of neuronal activity is the most popular method to study brain functions. It is usually necessary to separate single unit activity from the multiunit record and spike sorting algorithms must be applied.

A novel method for detection and classification of neuronal discharges based on non linear modeling has been presented here. Neuronal discharges are considered as solutions of non linear dynamical equations. Each neuron is assumed to have its proper equation. The classification of spike waveforms is then considered as a pattern recognition problem in the phase space of dynamical systems described by the equations. Similar approaches based on signal processing in phase space were applied to other fields, like cardiology (Aksenova et Chibirova, 1996), and analytical chemistry for processing chromatograms assessing drug quality control (Aksenova et al., 1999). In these studies the signal processing could be performed off-line and the recognition was achieved by supervised algorithms because the training set was available and the number of classes was known. The new application of the approach in neurophysiology has required new algorithms and methods. The electrophysiological signal is usually characterized by a high level of background noise. Moreover the number of classes—number of single units—is not known with anticipation.

The USS system was developed with aims of application in experimental and clinical neurophysiological studies. The USS comprises spike detection and spike classification algorithms, as well as an iteration-learning algorithm that estimates the number of classes and their centers according to the distance between spike trajectories in phase space is developed. This algorithm scans the learning set in order to evaluate spikes trajectories with maximal probability density in their neighborhood. The estimation of neuronal discharges trajectories

in phase space required the calculation of the signal derivatives. The integral operators with piece-wise polynomial kernels were used to calculate the first and second order derivatives. This provided computational efficiency of the developed approach for the real time application. The described method provides good results for simulated spike trains (Letelier et Weber, 2000) and it favorably compares with other approaches.

The comparison with some other spike sorting algorithms without spike overlapping demonstrated a high level of performance of the developed method. Although some general hints show that even in case of overlapping the algorithm will still be competitive with the others some additions should be done in perspective to resolve the overlapping problem and, thus, improve the performance of the algorithm.

The USS algorithm demonstrated its performance in separating single unit signals with amplitude barely greater than the amplitude of background noise. A denoising procedure such as TSA, very efficient for improving the recognition of speech signals in a noisy environment (Bahoura and Rouat, 2001), could not improve at all, even worsened, the performance of USS. The USS showed a high performance without any pre-processing, the performance obtained by USS with SNR as low as 2.5 dB provides evidence that USS may be used for medical application with considerable confidence.

Part II

Neuronal Activity Analysis

Chapter 6

Basal Ganglia and Pathophysiology of Parkinson's Disease

Ce chapitre introduit briévement la physiologie desganglions de la base et du noyau sousthalamique en particulier et la physiopathophysiologie de la maladie de Parkinson. L'introduction dans la pathophysiologie de la maladie de Parkinson, du point de vue du modèle d'Albin-DeLong.
La chirurgie fonctionnelle et ses cibles chirurgicales pour le traitement de la maladie de Parkinson sont également abordées. La stimulation profonde du noyau sousthalamique est présentée, ainsi que le rôle de l'électrophysiologie intrachirurgical pour le positionnement final de l'électrode de stimulation.

6.1 Basal Ganglia

The basal ganglia comprise the following main nuclei (Yelnik, 2002):

- the striatum, consisting of the caudate nucleus and the putamen;
- the thalamus;
- the globus pallidus with its internal (GPi) and external (GPe) segments;
- the subthalamic nucleus (STN);
- the substantia nigra subdivided into pars compacta (SNc) and pars reticulata (SNr)
- the peduncolopontine (PPN).

Anatomically, all these structures are located subcortically adjacent to the ventricular system. These nuclei range in size and shape and have extensive afferent and efferent connections to the cerebral cortex, cerebellum and the sensory nuclei. Several neurotransmitter systems are active in the basal ganglia. Gamma-aminobutyric acid (GABA) and glutamate are the most common neurotransmitters. Dopamine also plays an important role in the regulation of the basal ganglia functions. The known dopamine receptors can be separated into two distinct families: D1and D2 receptors (Calne et al., 1993). The effect of dopamine and dopamine agonists on D1 receptors is excitatory, whereas their action on D2 receptors is inhibitory.

Based on neuroanatomical and neurophysiological studies in animal models of Parkinson's disease (PD) and electrophysiological studies in PD patients undergoing functional brain surgery a model of basal ganglia function and physiopathopysiology has been proposed (Albin et al., 1989; DeLong, 1990) (figure 6.1). According to this model, the connections between the striatum and GPi/SNr can be divided into direct and indirect pathways.

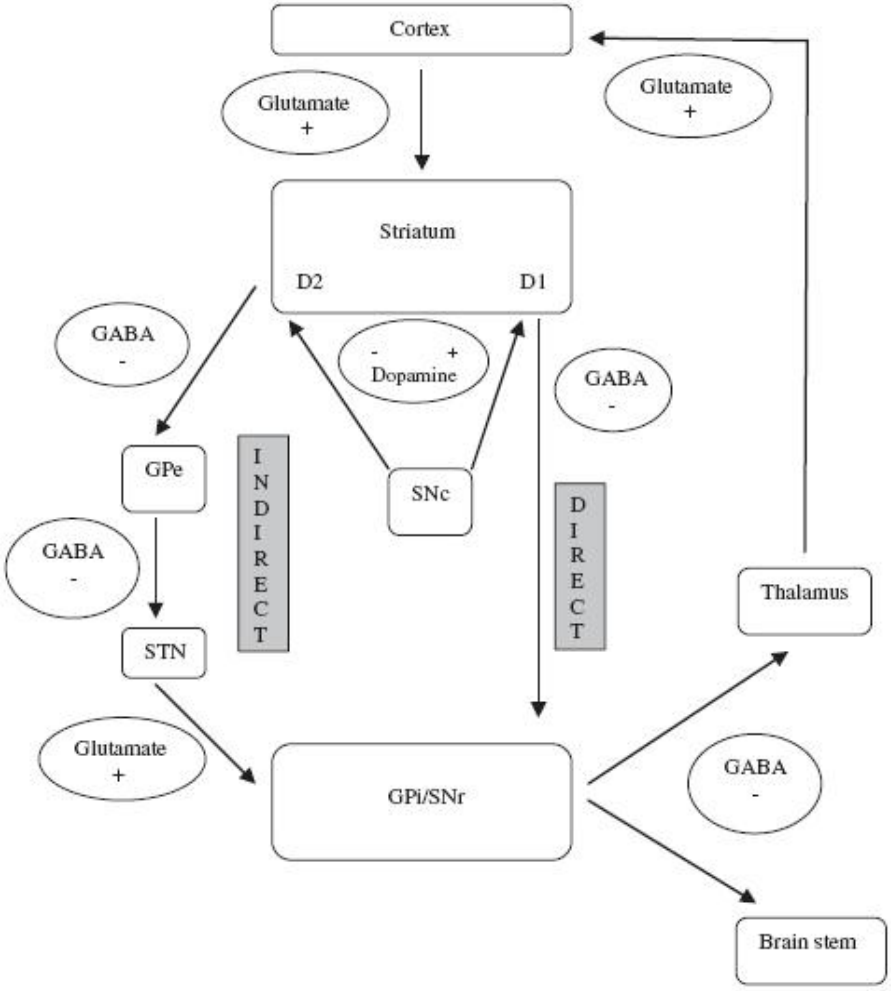


Figure 6.1
Simplified schematic diagram showing internal basal ganglia connections.

The direct pathway consists of monosynaptic neurons containing D1 receptors that project from the striatum (caudate and putamen) to GPi/SNr. This is an inhibitory circuit whose excitation reduces the output of the basal ganglia. Since the GPi/SNr employs the inhibitory neurotransmitter GABA, the excitation of the direct pathway reduces the inhibitory

effect on the thalamus and cortex, thus facilitating motor function.

In the indirect pathway, the striatal neurons containing the D2 receptors project to GPi/SNr via the GPe and STN. This is an excitatory circuit whose activation increases the GABAergic output of the basal ganglia thus suppressing cortical activity (Russchen et al., 1985; Lang et Lozano, 1998). The striatum and STN are the main input nuclei of the basal ganglia, receiving afferents mainly from the cortex but also from the amygdala (Russchen et al., 1985). The output nuclei, GPi and SNr, project back to the cortex via the thalamus, as well as to several brain stem structures including the pedunculopontine nucleus (PPN) (Wichmann et al.,2003; Nandi et al., 2002).

This network is composed of several circuits that remain segregated throughout their subcortical course. Each of these circuits originates in specific cortical areas, passes through a distinct portion of the basal ganglia and thalamus, and project back to the frontal cortical area of origin. The cortical sites of origin of these circuits define the presumed function of the circuits as "motor", "oculomotor", "associative" and "limbic". In each of basal ganglia-thalamocortical circuits, the striatum and STN serve as the input stage of the basal ganglia, and the GPi and the SNr as output station (Hazrati et al., 1990; Parent et Hazrati, 1995).

Computer simulation studies and mathematical analysis of models of the basal ganglia are being used increasingly to explore theories of basal ganglia function (Gillies et Arbuthnott, 2000). In (Gurney et al, 2000, Gurney at al 2004) proposed a computational model of basal ganglia describing anatomically defined connections between basal ganglia. The models are used to describe basal ganglia interactions during behavioral, cognitive and selection tasks.

6.2 Parkinsonian Disease

PD is a progressive neurodegenerative disorder of the central nervous system. It affects about 1% of the population over 65 years old. The PD is characterized by muscle rigidity, bradykinesia, tremor at rest, and postural instability. PD affects the basal ganglia and is unique in that the symptoms result from a single transmitter deficit due to the loss of dopamine in the substantia nigra, pars compacta. The etiology of the disease is unknown (idiopathic), but epidemiological studies have suggested that exposure to toxins and viruses

may predispose to the disease along with life events and the ageing process. It is likely that about 70-80% of the nigrostriatal neurones have to cease functioning or die before the symptoms appear (Kessler, 1972).

PD is characterized by degeneration of the dopaminergic neurons that therefore results in inhibition of movement, and appearance of the clinical hallmarks of akinesia and rigidity. Under normal conditions, dopaminergic output from the SNc to the striatum facilitates movement by exciting the direct pathway (via D1 receptors) and inhibiting the indirect pathway (via D2 receptors) (Wichmann et DeLong, 2003). The depletion of dopamine acting through D1 receptors excitatory from the nigrostriatal projection zones in the striatum, results in the direct pathway being underactive. This leads to decreased firing of thalamic neurones and hence inhibition of initiation of movement. The depletion of dopamine (acting through D2 receptors - inhibitory) from the nigrostriatal projections in the striatum, results in the indirect pathway being overactive (figure 6.2). Hence, the combined hyperinhibitory outflow from GPi may account for the negative symptoms of PD (rigidity and bradykinesia).

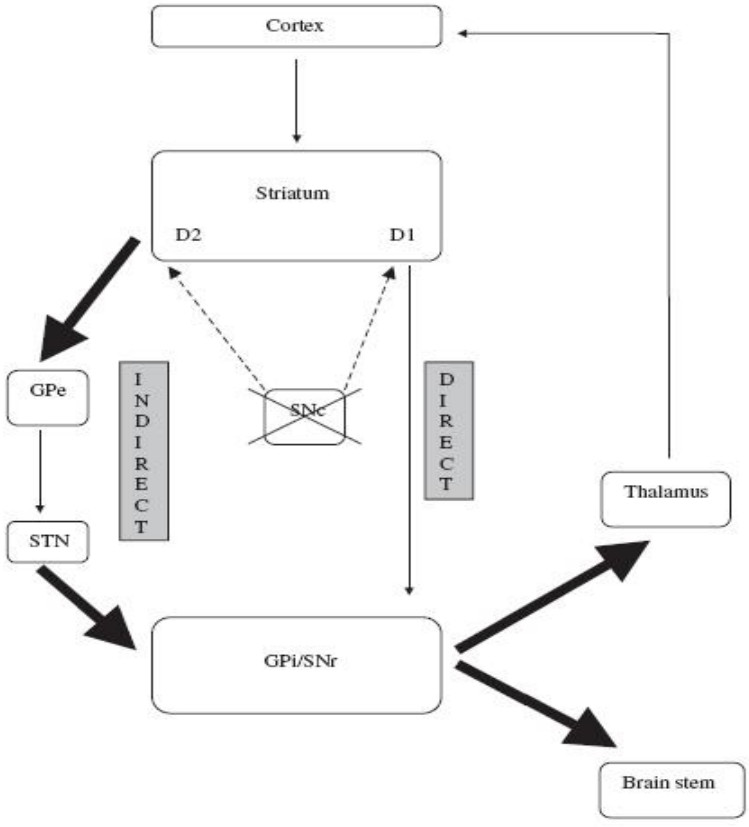


Figure 6.2
Simplified schematic diagram of basal ganglia in PD. Arrow sizes indicate the degree of the activity of the pathways.

Hyperkinetic symptoms of PD, such as tremor and involuntary contractions, are more difficult to explain in this model. With regard to tremor, studies on primate models of PD have shown increased coupling of neighbouring neurons in the STN, GPi and SNr, both in terms of oscillatory and non- oscillatory burst activities compared with normal (Bergman et Wichmann, 1994; Fillion et Tremblay, 1991). Lack of dopamine in the basal ganglia therefore appears to result in the loss of the normal parallel activity of the subcircuits and encourage abnormal synchronicity of part or all of the basal ganglia system which then manifests as tremor (Bergman et Deuschl, 2002). Other involuntary movements seen in PD have been explained by the 'action selection model' (Alexander et al, 1990). In this hypothesis, when a voluntary movement is generated, the basal ganglia facilitate this by inhibiting antagonist actions. Loss of this ability due to dopamine depletion in PD leads to abnormal recruitment and subsequent involuntary muscle activity.

Moreover, the basal ganglia disorder in PD may account for both typical motor disturbances and also cognitive-emotional deficits often encountered in the advanced stages of the disease. It can be explained by the fact that associative-cognitive and emotional basal ganglia circuits, as well as motor circuit receive an input from STN, which is abnormally hyperactive in PD.

6.3 The Subthalamic nucleus: anatomy and neurophysiology

The abnormal activity in the STN is an important feature in the pathogenesis of movement disorders, such as Parkinsonism. *In vitro* recordings of STN neurons suggest that these cells have at least two different discharge modes, depending on the membrane potential (Beurrier et al., 2000). In a relatively depolarized state, these neurons discharge in a "single spike mode", at frequencies in the 10-20 Hz range, most likely driven by slowly inactivating voltage gated sodium channels, and modulated by calcium dependent potassium currents (Beurrier et al., 1999; Bevan et Wilson, 1999). At more hyperpolarized potentials, the cells discharge in a burst mode, often with regular recurring bursts. The various excitatory and inhibitory inputs to the STN utilize and modify these intrinsic discharge properties to produce the rich variety of discharge patterns recorded from the STN under *in vivo* conditions (Wichmann et al. 1994).

In human, the STN is of diencephalic origin arising from the lateral hypothalamic

cell column, and is located between the thalamus and the mid-brain. Human STN is lens-shaped measuring 3x5x12 mm and lying obliquely in all three planes (Yelnik et Percheron 1979). In rodents, the STN's dendritic tree extends beyond the limits of the nucleus, hence receiving inputs that terminate in the adjacent structures, but in primates the STN is a 'closed nucleus' implying limitation of the dendrites to the nucleus (Mannen 1960). Thus, the STN is both anatomically and physiologically well delineated.

The STN receives glutamatergic afferents from the cortex (Kitai et Deniau, 1981) and the parafascicular nucleus of the thalamus (Mouroux et al., 1995), GABAergic input from the GPe (Alexander et Crutcher, 1990), dopaminergic projections from the SNc (Campbell et al., 1985) and cholinergic/glutamatergic afferents from the PPN (Parent et Hazrati, 1995). Its efferents are excitatory glutamatergic and are mainly directed at GPi and SNr (Lang et Lozano, 1998).

6.4 Functional neurosurgery

To improve motor functions, PD patients are usually treated with levodopa, which replaces dopamine in brain. However, as the disease progresses, the levodopa treatment loses its efficacy, and at the same time drug-related side effects appear (especially psychosis, motor fluctuations, and dyskinesias). At this stage, the best alternative is surgery.

Discrete surgical brain lesions to treat the symptoms of PD had been widely used during the 1950s and 1960s. The low success rate and unacceptable side effects and complications of that surgery as well as the introduction of levodopa seemed to indicate to abandon surgery as treatment for PD (Giladi et Melamed, 2000; Honey et al, 1999). For several reasons, however, there has been a tremendous resurgence in the surgical therapy for PD. This has been prompted by a combination of factors. Firstly, conventional medical therapies can loose their effectiveness over time, have been ineffective in preventing long term decline and can in many cases be associated with unacceptable side effects including dopa induced involuntary movements (dyskinesia) and psychiatric complications (Marsden et al., 1977; Nutt et al., 1987). Secondly, improved neurosurgical techniques have allowed for more consistent results with fewer complications. Thirdly, increased understanding of basal ganglia physiology has provided models which account for some of the pathophysiology of Parkinsonian features and a scientific rationale for surgical intervention. There are tree

categories of surgical treatment of PD: lesions, High Frequency Deep Brain Stimulation (HF DBS) and transplantation. Surgical targets for PD are Thalamus - ventral intermediate nucleus (VIM), GPi and STN.

VIM nucleus thalamotomy is quite effective in relieving tremor (Tasker, 1990). Its effects on the other clinical features of PD are less prominent and more variable. For this reason, thalamotomy is restricted to patients who have predominantly drug resistant tremor. This represents a small proportion of the PD population which means that for most patients, the thalamus is not the most appropriate surgical target. Chronic DBS of the thalamus, which has similar indications, is a nonlesional alternative to thalamotomy (Guiot et al., in 1962). Chronic VIM stimulation is highly effective for tremor, with over 85% of patients having a very good or excellent response with little or no tremor evident in the contralateral arm (Benabid et all, 1996). While effective for tremor, VIM stimulation did not influence bradykinesia or rigidity (Koller et al., 1997).

Surgical lesioning of GPi (Pallidotomy) as well as HF DBS is mainly used in patients in whom rigidity and bradykinesia/dyskinesia are the major reasons for disability. DBS of GPi improves all major Parkinsonian features and drug induced dyskinesias (Gross et al., 1997; Tronnier et al., 1997). The clinical effects are dependent on which parts of the pallidum are stimulated and which stimulating parameters are used (Krack P et al., 1998; Bejjani et al., 1997).

STN lessioning have been used in monkeys with surprisingly good results. Such a lesion, however, carries a great risk of persistent contralateral hemiballism. STN lessioning has so far not been routinely used in humans. In opposite, STN DBS represents an exciting new development in PD surgery.

6.4.1 High Frequency Deep Brain Stimulation (HF DBS) of STN

STN is glutamatergic and drives both GPi and the SNr, the two nuclei which constitute the collective output of the basal ganglia (figure 6.2). It is therefore strategically situated to exert a powerful influence on motor function. Reducing STN activity would diminish the driving of GPi and SNr thereby lessening the inhibition of thalamocortical projections and the motor cortical system. Such an intervention would thus be expected to facilitate movement. This has been supported by experimental data from the animal models of PD.

Electrophysiological results in monkeys made Parkinsonian by the injection of 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine (MPTP) showed a significant increase, as well as alterations in the firing pattern of the STN neurons (Bergman et al., 1994; Miller et DeLong, 1987). It therefore followed that interruption of transmission from the STN might have the potential to improve Parkinsonian symptoms. Indeed, Aziz (Aziz et al., 1991), Bergman (Bergman et al., 1994) and Guide (Guide et all., 1994) independently demonstrated this through lesioning of the STN in the MPTP primate models. The problem encountered with these studies, however, was the development of hemiballismus and dyskinetic movements. As a result, interest was diverted away from anatomical lesioning of the STN and towards modulating its function. HFS (frequency > 100 Hz) of the STN presented a promising alternative.

HFS was known to produce controllable and reversible modulation of the function of deep brain nuclei. HFS of the ventral intermediate nucleus of thalamus had already been shown to improve Parkinsonian tremor without the adverse effects of thalamotomy (Benabid et al, 1987; Benabid et al, 1991; Benabid et al, 1996; Ohye et al., 1989; Tasker et al., 1983). On this basis, STN HFS was applied to the MPTP monkey model of PD with significant improvement in rigidity and motor scores without hemiballismus (Benazzouz et al., 1993). Furthermore, the improvements were comparable with those obtained after levodopa treatment (Benazzouz et al., 1996).

The science was turned into clinical practice in 1993, when bilateral HFS STN electrodes were implanted in PD patients. This was associated with improvements in all cardinal motor symptoms of PD, as well as a reduction in drug-induced dyskinesias, and improvements in the independence and quality of life of the patients (Benabid et al., 1994). The three first patients stimulated with implanted quadripolar electrodes showed striking improvements in all motor disabilities in PD. These patients were evaluated using the UPDRS scale three months after surgery: the activity of daily living scores improved by 58-88%, and motor scores by 42-84% (Limousin et al., 1995). This has been confirmed later on larger series (Limousin et al., 1998; Kumar et al., 1998).

6.4.2 Intraoperative electrophysiology

Neurons within each basal ganglia nucleus have specific patterns of neuronal

discharges. Using single cell neuronal recording, these neuronal discharge patterns can be recorded and analyzed. Neuronal firing patterns have specific "bursting" and "oscillation" patterns in addition to specific discharge rates. These patterns of activity can be useful to map specific structures for functional surgery.

The first operative step is the definition of the anatomical target. The localization of the deep brain structures - prior to surgery for PD - is usually achieved using magnetic resonance or axial computed tomography-guided stereotactic targeting in concert with stereotactic frames, which is then refined by intraoperative electrophysiological techniques. It is a 'theoretical' target because, firstly, even with the modern neuroimaging techniques and stereotactic atlases the location of the STN cannot be defined with 100% accuracy in all patients and, secondly, because notwithstanding its small size, the STN appears to have a complex internal topography with motor and non-motor areas, which can only be recognized by physiological means (Rodriguez-Orzoz et al., 2001).

Hence, some teams employ intraoperative electrophysiology to increase the accuracy of targeting. Intraoperative electrophysiological analysis, including both electrical stimulation and electrical recording is aimed at providing functional information and improve the confidence in neurosurgery for PD. The microelectrode technique probably yields the highest-quality localizing information and despite of the extra time, effort, and expense of performing intraoperative electrophysiological testing its usefulness for clinical applications has been confirmed.

Electrical recording allows assessment of multi-unit neuronal activities, thus identifying the signature pattern of various deep brain nuclei. The thalamus can be recognized by a bursting pattern of discharge. Under the thalamus there is then an area of decreased activity - the zona incerta and the Fields of Forel. Then STN is usually characterized by a sudden increase in the background noise reflecting the high cellular density of the area and neurons that show bursting activity at frequencies of 40 – 50 Hz (Benazzouz et al., 2002; Pralong et al., 2002; Pralong et al., 2002). Location within the motor area of the STN can be verified by the response of neurons (increased firing audible through signal transduction and amplification) to passive movement of limbs or muscular palpation (motor response). The motor area is predominantly located in the dorsal portion of the STN (Sterio et al., 2002). Entrance into the SNr is identifiable by its high, but regular discharge rate of about 70 spikes/s.

Electrostimulation is usually performed at several levels. The same parameters as those for chronic stimulation (frequency of 130 Hz and pulse width of 0.06 ms) at various amplitudes are applied on each level. During the stimulation, the patient is examined clinically by a movement disorder neurologist. Clearly patient's cooperation at this stage is crucial. The examination has two purposes, to identify clinical improvement and discover the side effects (Pollak et al., 2002 ; Houeto et al., 2003; Krack et al., 1999; Benabid et al., 2000).

Chapter 7

Data Acquisition

Les donnés pour les analyses ont été acquises au bloque chirurgical du département de neurochirurgie de l'Hôpital Universitaire de Grenoble pendent l'électrophysiologie intra chirurgicale. Ce chapitre est destiné à exposer les conditions sous lesquelles l'acquisition des donnés a été effectuée. Ce chapitre décrit la procédure chirurgicale de stimulation profonde, ainsi que quelques détails techniques d'enregistrement des signaux électrophysiologiques.

7.1 Surgical procedure and intrasurgical electrophysiology

The following is a description of the procedure as it has evolved in the Departments of Neurosurgery and Clinical Neurosciences of the University Hospital of Grenoble. The surgical operation for chronic deep brain stimulation of STN is performed under local anesthesia. Pharmacological PD medication is stopped 12 hours before surgery. The bilateral procedures were performed consecutively during the same operative session. Patients were placed in a stereotaxic frame and bilateral STN were targeted (Benabid et al., 2002).

The first operative step is the definition of the anatomical target, which is then refined by intraoperative electrophysiological techniques. The location of each STN is determined using ventriculography landmarks, i.e. midline of the third ventricule and the anterior (AC) and posterior (PC) commissures. The antero-posterior coordinates of STN calculated according with atlases (Schaltenbrand and Wahren W, 1977; Talairach et al., 1957) is then used to build a Guiot scheme (Guiot et al., 1968) based on the AC-PC line and the lateral coordinates are set at 12 mm from the middle of the third ventricule (figure 7.1).

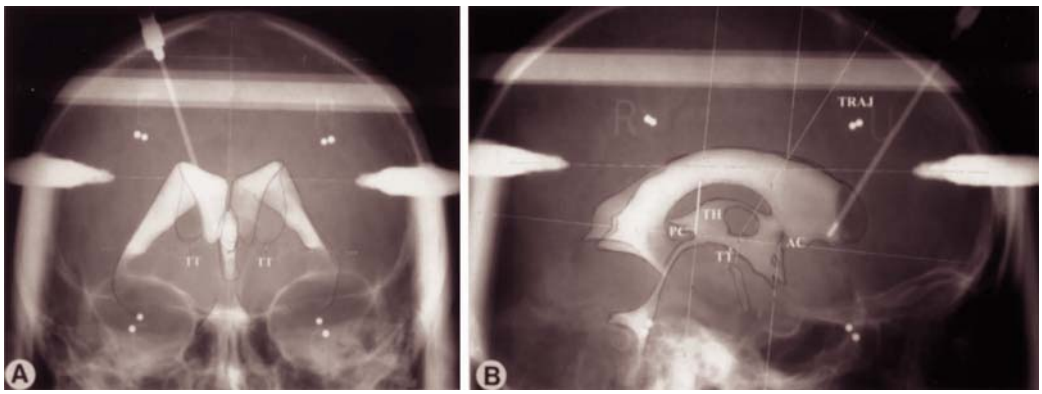


Figure 7.1
Anteroposterior (A) and lateral (B) ventriculograms showing the theoretical subthalamic nucleus target (TT), anterior commissure (AC), posterior commissure (PC), thalamic height (TH) and initial target trajectory (TRAJ).

A small skin flap is made under local anesthetic and a burr hole is fashioned. To approach the target a device containing five parallel tubes is used. The configuration of the five electrode guidance tubes is with one at the centre (aimed at the theoretical target) and four concentrically in the periphery, each 2 mm apart. This allows insertion of five microelectrodes (Tungsten bipolar, impedance 2–6 MΩ, FHC, Bowdoinham, USA) simultaneously, and thus enables probing of a larger area to map the STN volume in detail and locate the optimal target.

To locate the functionally optimal target, microrecording and microstimulation are used through the same 1-mm tip microelectrodes. Microrecording allows assessment of single and multi-unit neuronal activities, thus identifying the signature pattern of various deep brain nuclei. Recording is commenced at the AC – PC line. The five microelectrodes are moved simultaneously by a microdriver at 0.2-mm intervals. At various stages of the electrode advance the position of the microelectrodes is confirmed by intraoperative X-ray radiography. During a typical track, the first structure encountered is the thalamus, recognized by a bursting pattern of discharge. There is then an area of decreased activity as the electrode tip enters the zona incerta and the Fields of Forel. The STN is usually encountered about 3 mm below the AC – PC line and is characterized by a sudden increase in the background noise reflecting the high cellular density of the area and neurons that show bursting activity at frequencies of 40 – 50 spikes/s (Benazzouz et al., 2002; Pralong et al., 2002; Sterio et al., 2002). Location within the motor area of the STN can be verified by the neuronal response of

neurons (increased firing audible through signal transduction and amplification) to passive movement of limbs or muscular palpation (motor response). In the series of Sterio et al., this area was predominantly located in the dorsal portion of the STN. The STN is usually left 7 – 8 mm below the AC – PC line. Entrance into the SNr is identifiable by its high, but regular discharge rate of about 70 spikes/s.

Microstimulation then follows during the ascent back out of the brain. Microstimulation is usually performed at three levels, typically 7, 5 and 3 mm below the AC – PC line. Monopolar stimulation is utilized with patient/frame used as the reference. The same parameters as those for chronic stimulation (frequency of 130 Hz and pulse width of 0.06 ms) are applied sequentially to each of the five microelectrodes at various amplitudes, starting at 0.5 μA and increasing up to 5 μA. During the stimulation, the patient is examined clinically by a movement disorder neurologist to define the final trajectory and depth for the definitive electrode.

On the basis of this clinical assessment and of the electrophysiological recordings the surgical team selects the most appropriate site for the chronic implantation of the stimulating macroelectrode (Medtronic 3389, Minneapolis, USA). During and in the end of the surgical procedure, a teleradiological X-ray control allowed for identification of the position of the tip of each micro-electrode. Five days after surgery, an MRI examination was performed in order to assess the final position of the tip of the macroelectrode within the STN and the lack of bleeding around it. The definitive electrode is then inserted under fluoroscopic guidance to its final location.

7.2 Data acquisition

The electrophysiological data were collected from 13 Parkinsonian patients mostly suffering from severe akinesia with disabling motor fluctuation and rigidity, and, for some of them, from tremors and from 2 distonia patients at the Department of Neurosurgery of the University Hospital of Grenoble. A surgical microdrive allowing five simultaneous parallel trajectories, with four around each 2 mm apart from the central one, was used. Tungsten bipolar microelectrodes (Frederick Haer and Co., Brunswick, ME), with impedances ranging from 2 to 6 MΩ were used for recording neuronal activity.

Recordings were performed at different sites along the trajectory, using the MicroGuide™ Intraoperative Microelectrode Recording System for Functional Neurosurgery of AlphaOmega, Nazareth, Israel. MicroGuide™ is a modular multi-channel FDA approved system which allows manual and computer controlled electrode manipulations. It comprises a data acquisition system "Alpha Map" that allows acquiring of a large number of analog and digital signals simultaneously in various modes. The Alpha Map can be used as a data logging system to save the analog and digital inputs continuously at a high rate to a disk in data acquisition internal system files. The MicroGuide™ system comprises also other features that can be useful during functional neurosurgery (figure 7.2). The system installed the University Hospital Grenoble comprises five isolated preamplifiers that provide five simultaneous recordings on each site. Using this system, signals with distinctive neuronal activity were selected and digitalized during 90-150 seconds at sampling rate 48 kHz and saved in Alpha Map data acquisition files.

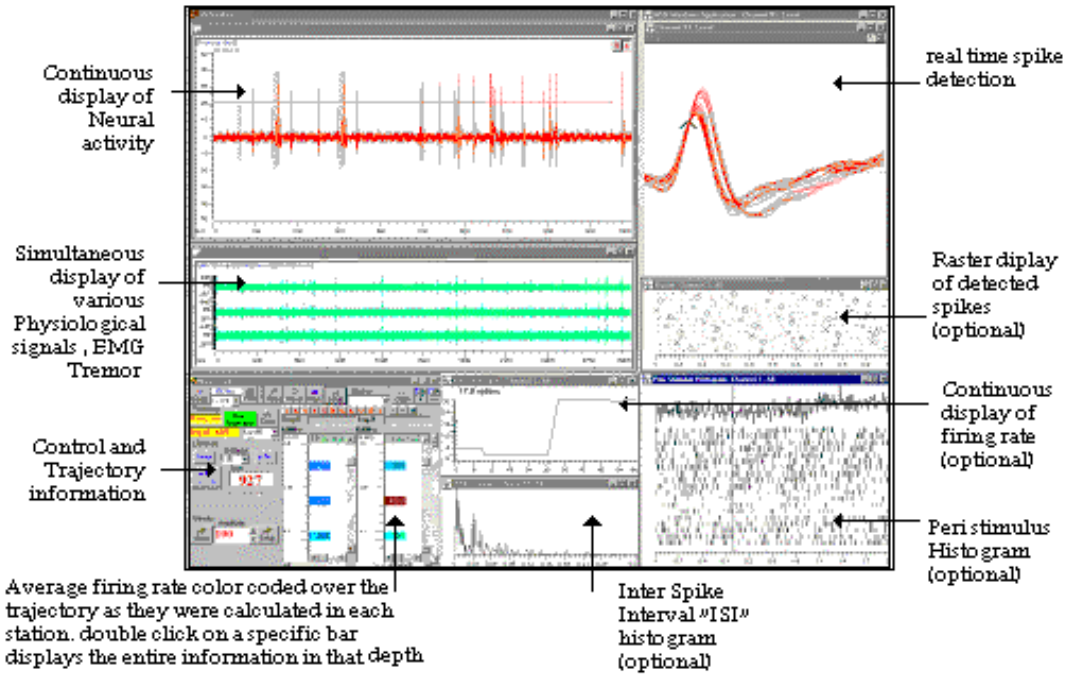


Figure 7.2

AlphaMap software. Visualisation of the data acquisition, some data post processing and the MicroGuide trajectory control.

7.3 File format

The Alpha-Map data acquisition files have a specific rather sophisticated binary format. These files have extension 'map' and contain a set of consistent blocks. A detailed description of the data format is given in Appendix B.

For reading Alpha-Map data files a shared dynamic library was created and included into the USS. The dynamic library is described in Appendix B.

Chapter 8

Spike Train Analysis

L'analyse des trains de potentiel d'action a été effectuée sur les données recueillies sur 13 patients parkinsoniens et 2 patients souffrants de dystonie. 950 trains de potentiel d'action unitaires provenant du NST, 183 trains provenant du GPi et 105 trains provenant du SNr ont été extraits de ces enregistrements. Les patterns d'activité neuronale dans les structure de NST, GPi et SNr ont été comparés en appliquant des test statistiques. Les résultats principaux sont suivants : les patterns d'activité neuronal chez les patients individuels se diffèrent significativement ; Nous pouvons observer également des différences entre les patterns d'activité dans les structures de NST, GPi et SNr, en particulier les différences de l'activité bursté dans GPi et de l'activité oscillatoire en fréquences de tremblement dans NST.

8.1 Spike sorting

The first step of the analysis was the separation of single unit spike trains for all the electrophysiological data set acquired at the neurosurgery theatre. For this purpose we used the USS software described in Part I of this document. The quality of spike train separation by means the USS is discussed in Chapter 4. The threshold level for spike detection was set, by default, equal to 3 times the variance of the signal derivative in order to exclude most of the noise that could interfere. Segments of 1.2-1.4 ms assumed to represent spikes were extracted from records. For the learning, sets of 300 spikes were collected. The results of the learning procedure were inspected by an expert operator (figure 3.5) in order to select the final templates for the spike sorting.

We analyzed in this way the electrophysiological data of 13 Parkinsonian patients and 2 dystonia patients who underwent DBS surgeries. About 350 single channel recordings characterized by steady state recording conditions and distinguishable neuronal activity were

selected. 950 single unit spike trains from STN, 186 spike trains from SNr and 105 spike trains from GPi were extracted, which makes in average 3-4 single units per record, up to 8 single units from 1 channel.

8.2 Spike train analysis

The Poissonian process is usually considered as a reference model of point process describing the neuronal activity. The usual analyses of spike trains include auto- and crosscorrelation analysis as well as spectrum analysis. For the Poissonian process the autocorrelation trace, as well as the crosscorrelation trace of two independent Poissonian processes, are flat and stay within 99% confidential levels (figure 8.1). In this study neuronal spike trains were analyzed following the usual analysis with the addition of specific analyses described later in this chapter. Different kinds of deviations from the Poissonian process were classified into several groups of particular neuronal activity patterns.

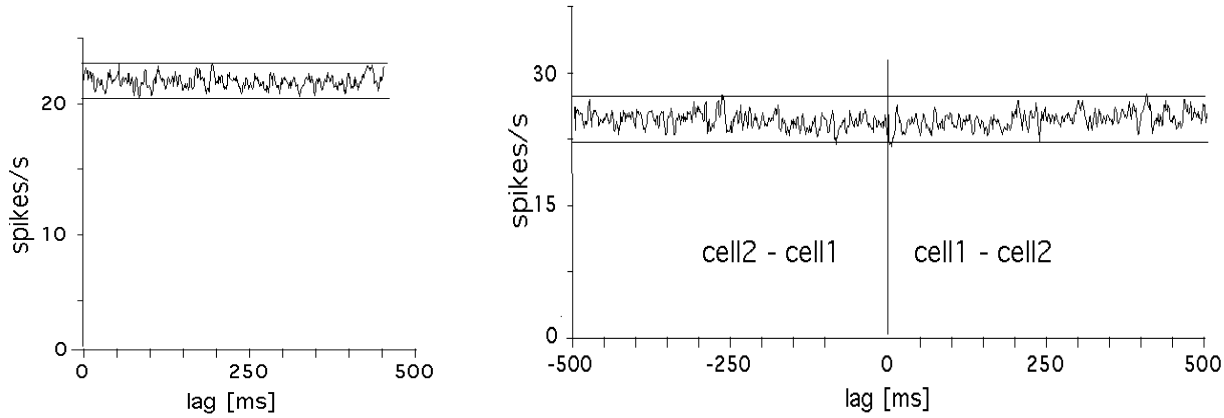


Figure 8.1
Autocorrelation trace (right) and crosscorrelation traces for cells (left), firing of which corresponds to the Poissonian process.

8.2.1 Time domain analysis of spike train

Time domain analyses included calculation of autocorrelations for spike trains and of crosscorrelations for pairs of them recorded from the same microelectrode. These computations were performed with the DAN software accessible on public domain at the website http://www.openAdap.net. The calculation of correlations and of the threshold levels

corresponding to the Poissonian process is realized following the method described in (Abeles, 1982). The auto- and crosscorrelations were calculated twice with bins 2 ms and 8 ms. The epoch was set to 500 ms for crosscorrelations and 1000 for autocorrelations. Then they were analyzed according to the method described in (Villa et Lorenzana, 1997). For the units with auto- and crosscorrelations graphics crossing the Poissonian process threshold levels the following patterns were defined:

1) Bursting cells ('Burst'): the autocorrelation of spike trains of these cells have a peak that crosses the upper threshold level close to the Y-axis (figures 8.2, 8.3). The burst is characterized by the location of its peak and its duration (Villa et Lorenzana, 1997). The duration of the burst is defined as interval with the autocorrelation curve above the threshold.

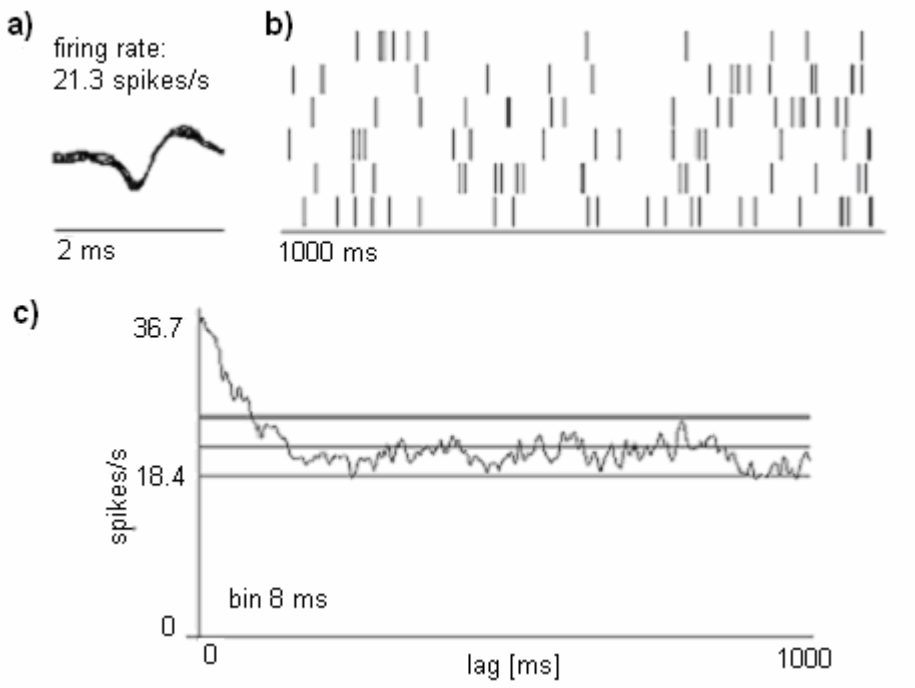


Figure 8.2
"Long" burst: (a) – bursting cell spike waveforms, (b) – raster and (c) – autocorrelation trace. The duration of the burst is about 80 ms.

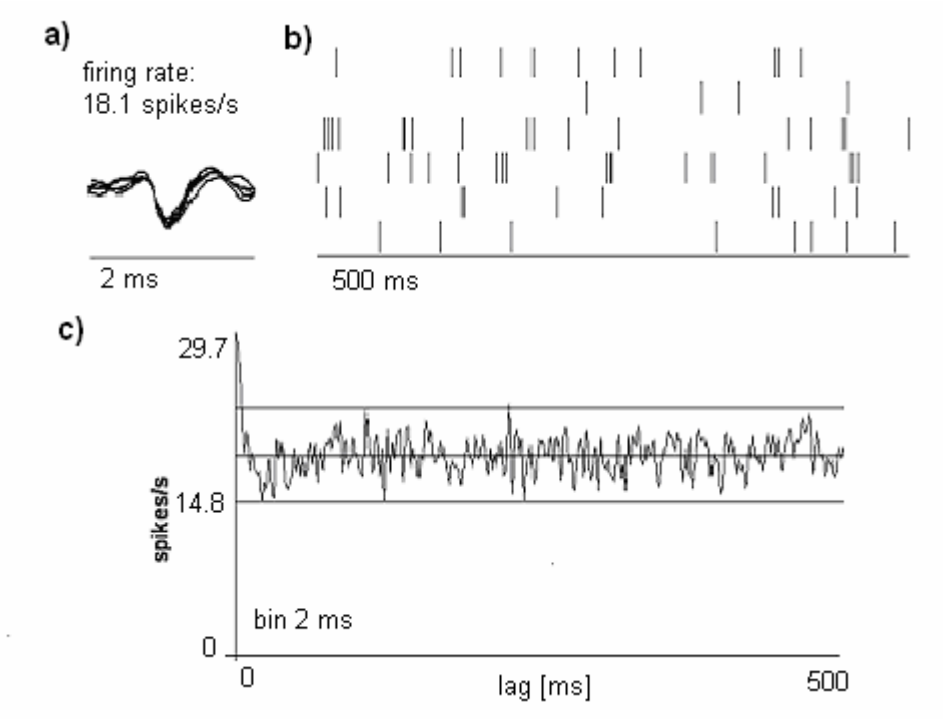


Figure 8.3
"Short" burst: (a) – bursting cell spike waveforms, (b) – raster and (c) – autocorrelation trace. The duration of the burst is about 10-15 ms.

For the analyzed cells the burst duration varies between 3 and 170 ms. Figure 8.4(a) represents the histogram of burst durations of all analyzed cells followed the burst pattern. The histogram shows that the distributions between 25 and 40 ms are represented less frequently than the others and the histogram seems to correspond to a mixture of populations. An additional analysis was applied to distinguish bursts of different duration corresponding probably to a different kind of bursting activity. It was assumed that the histogram may be approximated by a mixture of two normal distributions. The parameters of the mixture were found by the maximal likelihood method. Figure 8.4 (a) represents the histogram and the approximating mixture graph, figure 8.4 (b) shows the two distributions of the mixture separately.

Thus, the burst pattern was divided into two independent patterns represented by one of the populations. The point separating the populations was selected as the point equalizing the mistake of loss for each population due to their overlapping (figure 8.4 (b)). The patterns were named "Short Burst", the duration of which is less then 30 ms, and "Long Burst", the duration of which is greater then 30ms.

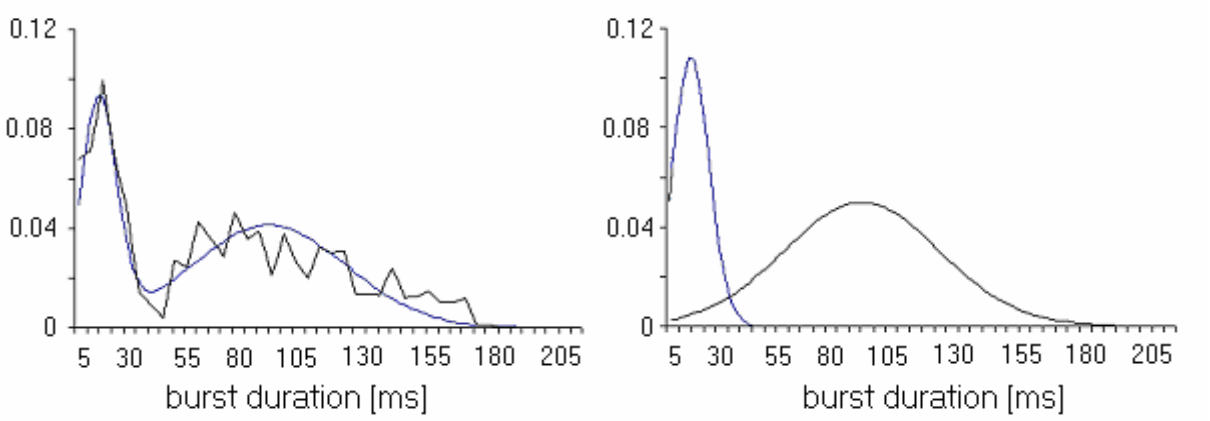


Figure 8.4

(a) Histogram of burst durations of all analyzed cells. The smooth line is a graph of the mixture of two normal distributions with parameters defined by the maximal likelihood approximating the histogram.
(b) Two normal distribution of the mixture. The point separating them was defined as the point which equalize the squares of two shaded areas i.e. the mistake of misclassification.

2) Cells with a significant refractory period ('Refractory Period'): the autocorrelation of spike trains that starts with a gap that crosses the lower threshold level (figure 8.5). The duration of the refractory period, is the interval with the autocorrelation trace below the threshold. The histogram of the refractory period is shown at figure 8.6.

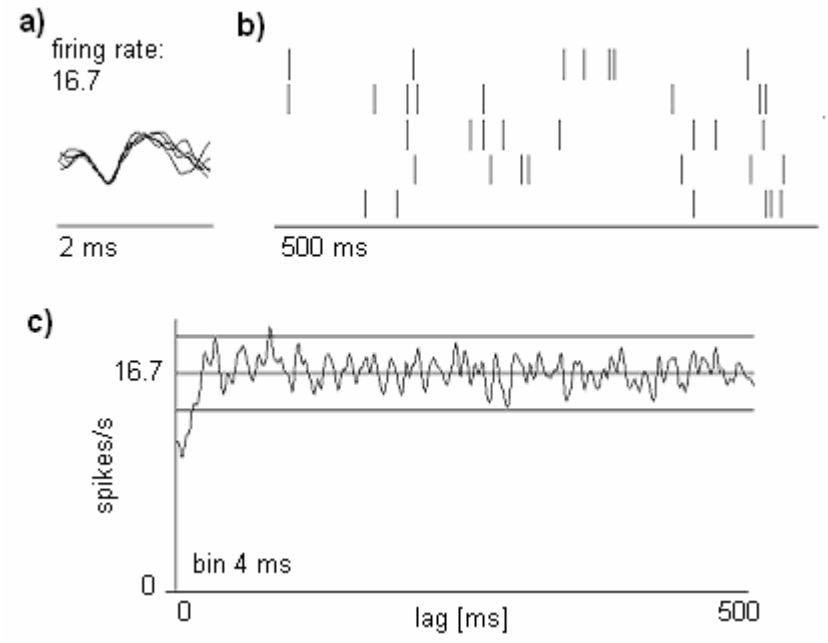


Figure 8.5

Cell with refractory period: (a) –spike waveforms, (b) – raster and (c) – autocorrelation trace. The refractory period is about 25-30 ms.

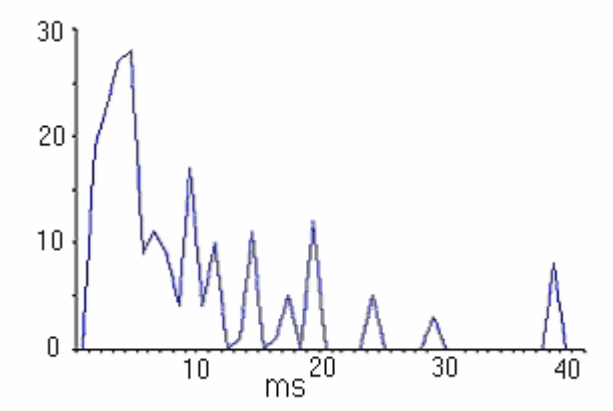


Figure 8.6
Histogram of refractory period durations of the analyzed cells.

3) Correlated cells ('Correlation'): the crosscorrelation graph of spike trains of these cells crosses the upper or the lower threshold level near the Y-axis. An example is presented on figures 8.7 and 8.8. On figure 8.7 is presented a correlation trace crossing the threshold level. Figure 8.8 shows rasters, spike waveforms and autocorrelation traces of the cell pair.

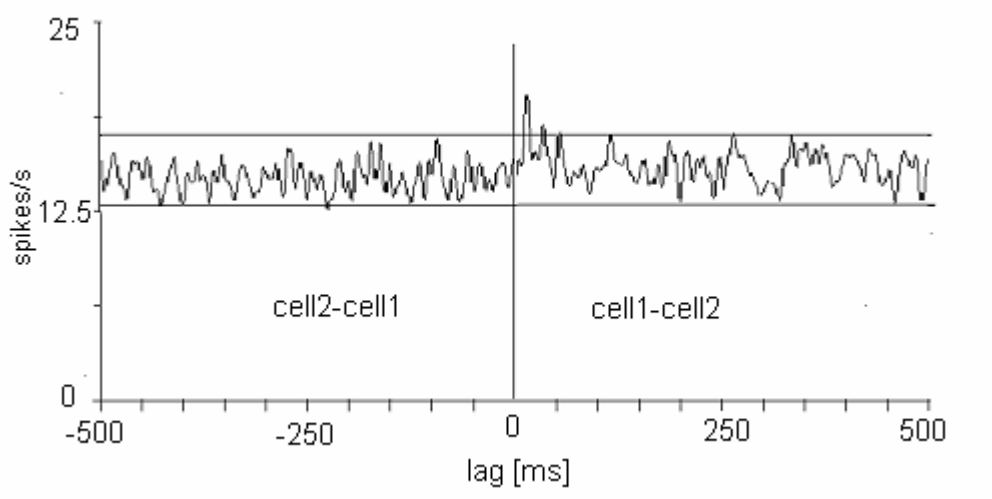


Figure 8.7
Crosscorrelation trace of a pair of correlated cells.

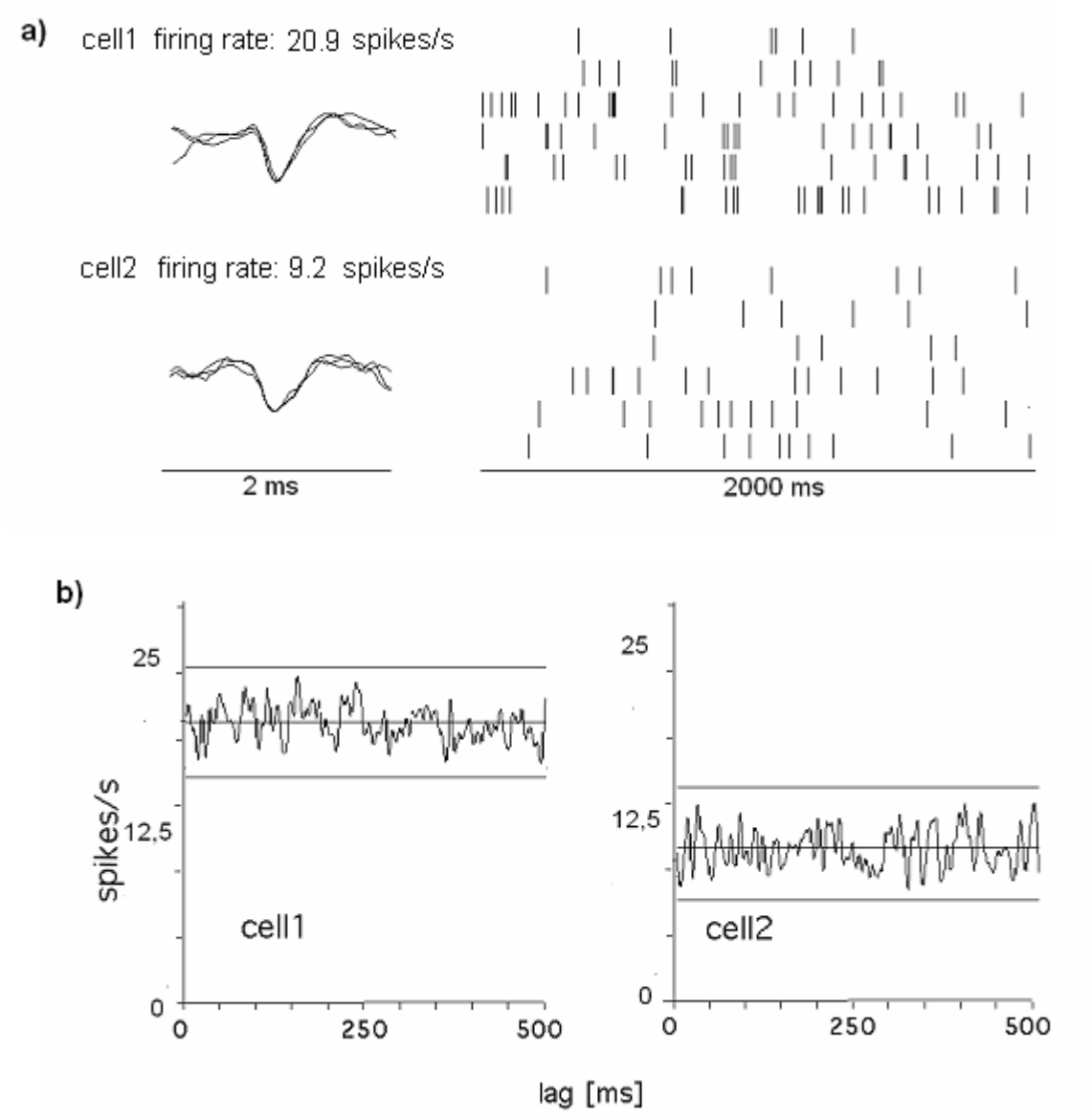


Figure 8.8
Rasters and spike waveforms (a) of two 2 corellated and their autocorellation graphs (b) (see fig. 8.7).

8.2.2 Frequency domain analysis of spike train

The analysis of the autocorrelations clearly showed highly distinctive oscillatory patterns of different frequencies (figures 8.9, 8.10). Notice that the multiunit channel recordings defined during surgery as “tremor cells containing” correspond always to spike trains with characteristic oscillation frequency components at about 4–6 Hz (figure 8.9). To better reveal the oscillating cells for each single unit time series we calculated the power

spectrum by applying the Fast Fourier Transform to the autocorrelogram traces with epochs equal to 500, 1000, 2000 ms. The cell was considered oscillating on some frequency if at least one of the three power spectrum traces calculated for the three epochs had a significant peak on this frequency. The significant peaks of the power spectrums were detected assuming that the neuronal firing is a Poissinian process and the confidential level is equal to 0.99 (figures 8.9(c), 8.10(f)).

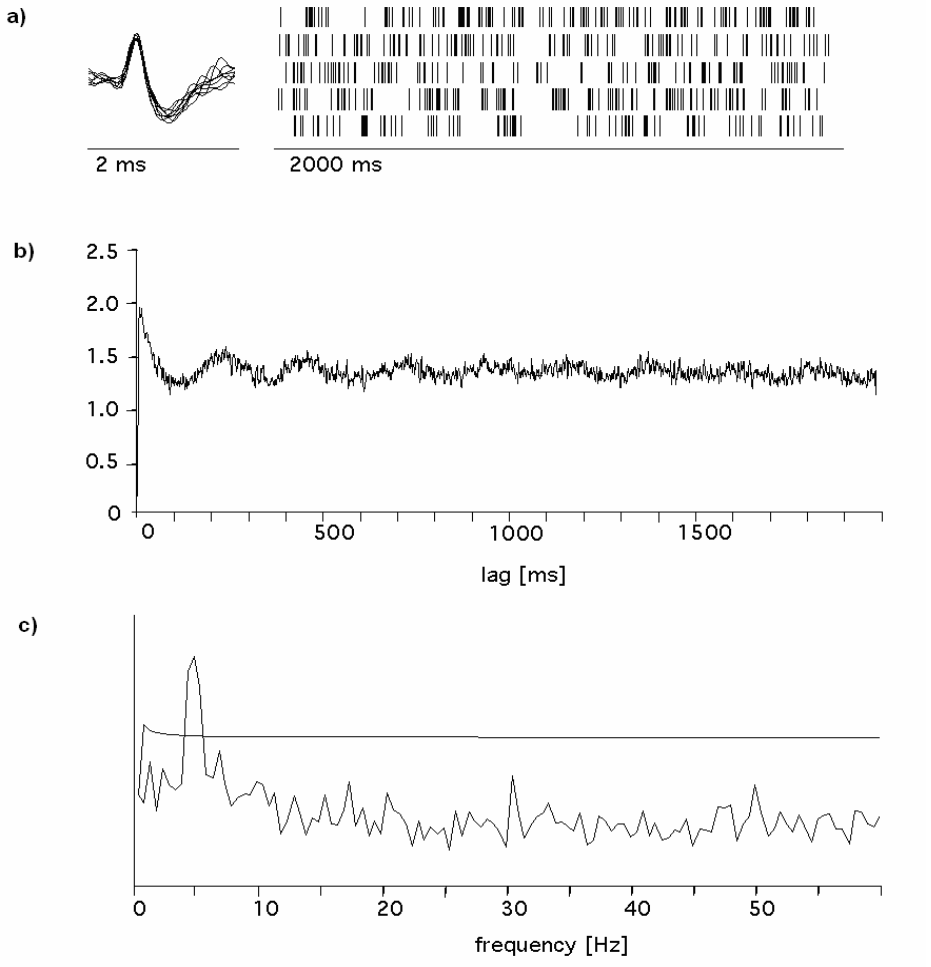


Figure 8.9

Single unit characterized by an oscillatory firing pattern with oscillation rates about 5 Hz.. The figure represents raster and spike waveform of the oscillating cell (a), its autocorrelogram (b) and power spectrums (c) with the confident level.

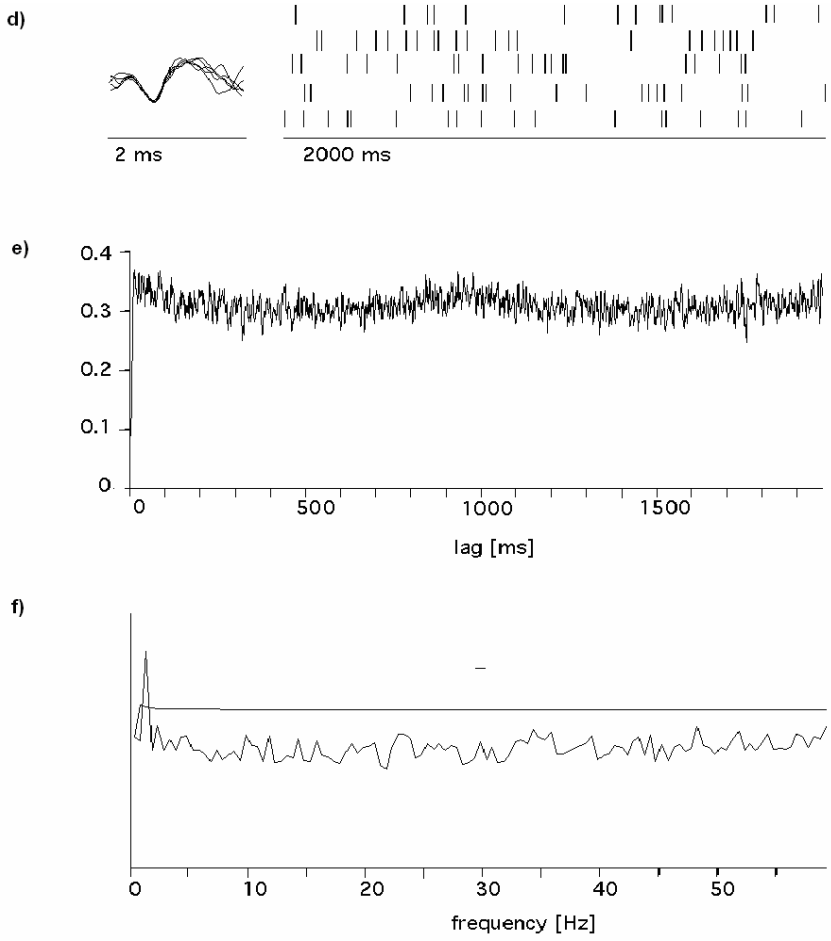


Figure 8.10

Single unit characterized by an oscillatory firing pattern with oscillation rates about 1 Hz.. The figure represents raster and spike waveform of the oscillating cell (d), its autocorrelogram (e) and power spectrums (f) with the confident level.

To define the most representative frequencies the following method was applied: the sum of all occurrences of power spectrum threshold crossings for each spike train were computed for frequency values ω in the range [1,50] Hz (figure 8.11). Let us assign $a(\omega)$ the number of threshold crossings for each $\omega = 1,...,50$. In case of a Poissonian process the probability of the threshold crossing by spectrum is $p = 0.01$. That means that $a(\omega)$ has a binomial distribution. The binomial distribution may be approximated by the normal distribution if the number of statistical observation n satisfies the condition $np(1-p) \geq 9$ (Huber, 1984), *i.e.* if $p = 0.01$ then n must be $n \geq 101$. Thus $a(\omega)$ can be considered as normal distributed with mean $\bar{a}$ and variance σ_a^2, which depends on n. In presence of significant peaks we have a mixture of two distributions: the "basic" distribution $N(\bar{a},\sigma_a^2)$

and a distribution that corresponds to significant peaks. Let us assume that the normalized random value

$$\xi_i = (a - \bar{a}) / \sigma_a$$

follows the distribution

$$P_\delta(\xi) = (1-\delta)\varphi(\xi) + \delta h(\xi),$$

where $\varphi(\xi)$ is a standard normal distribution density and $h(\xi)$ is a distribution density with large variance $\sigma_h >> 1$. In other words this means that we assume to observe a random value that follows either the "basic" distribution or $h(\xi)$ that forms the δ part of the sample. The threshold for detecting significant peaks is estimated by the parameters of the "basic" distribution considering the model of mixture. For any $\delta < 0.2$, the maximum likelihood estimate can be used for the mean and variance of the basic distribution (Ivchenko et Medvedev, 1984). We used a threshold corresponding to the confidential level 0.95 estimated for the reference hemisphere (side 1) to define the frequency bands with high probability density and considered as clusters the areas where the traces were above the threshold line. Three frequency bands were defined: [0–2] Hz, [4–6] Hz, [8–10] Hz.

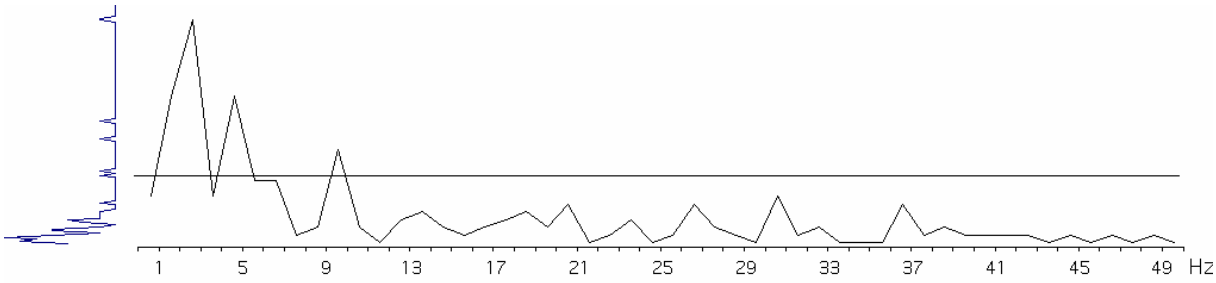


Figure 8.11

Cumulated distribution of significant peaks detected in the spectrum of single units characterized by oscillatory patterns. All occurrences of significant peaks in the frequency range [0–50] Hz were added together. The trace represents the probabilities of occurrence of the peaks at a resolution of 1 Hz. The horizontal line represents a 95% level of significance. The following clusters appeared: [0–2] Hz, [4–6] Hz, [8–10] Hz.

8.3 Results

The single unit spike train analysis was aimed to reveal for each single unit if it follows one or more of the defined patterns:

- Short Burst
- Long Burst
- Refractory Period
- Oscillation [0–2] Hz
- Oscillation [4–6] Hz
- Oscillation [8–10] Hz
- Correlation

Tables 8.1, 8.2 and 8.3 show the results grouped by patients for STN, SNr, and GPi respectively. Values in the tables show the absolute number of cells on which the given activity pattern was recorded. Values in parentheses present the percentage of cells of the given activity pattern with respect to the number of analyzed cells and pairs of cells. Table 8.4 recapitulates the results obtained in STN, SNr and GPi.

patient	cell count	pair count	mean firing rate	Long burst	short burst	refractory period	0-2Hz	4-6 Hz	10Hz	correlation
				cell number (%)						pair num. (%)
JC	81	98	18.6	9 (11%)	5 (6%)	5 (6%)	14 (17%)	12 (15%)	12 (15%)	12 (15%)
AG	131	257	13.4	36 (27%)	0 (0%)	26 (20%)	53 (40%)	6 (5%)	10 (8%)	44 (17%)
BS	100	196	18.5	28 (28%)	6 (6%)	18 (18%)	43 (43%)	21 (21%)	12 (12%)	42 (21%)
CB	107	299	23.0	13 (12%)	1 (1%)	28 (26%)	23 (21%)	3 (3%)	5 (5%)	21 (7%)
GB	66	71	14.9	4 (6%)	6 (9%)	9 (14%)	9 (14%)	4 (6%)	6 (9%)	19 (27%)
LS	60	82	20.6	7 (12%)	6 (10%)	13 (22%)	7 (12%)	7 (12%)	0 (0%)	18 (22%)
DS	123	195	23.5	1 (1%)	21 (17%)	3 (2%)	17 (14%)	16 (13%)	3 (2%)	34 (17%)
AT	118	205	28.9	2 (2%)	1 (1%)	5 (4%)	8 (7%)	4 (3%)	6 (5%)	89 (43%)
RW	64	117	21.8	6 (9%)	3 (5%)	6 (9%)	4 (6%)	2 (3%)	4 (6%)	16 (14%)
CG	60	90	20.7	7 (12%)	7 (12%)	3 (5%)	12 (20%)	2 (3%)	1 (2%)	14 (16%)
SR	40	72	22.5	0 (0%)	13 (33%)	1 (3%)	4 (10%)	0 (0%)	4 (10%)	20 (28%)
Total	950	1682	20.54	113(12%)	69 (7%)	117 (12%)	194(20%)	77 (8%)	63 (7%)	329 (20%)

Table 8.1

Neuronal activity in STN. Each row corresponds to an individual patient. Values in the tables show the absolute number of cells on which the given activity pattern was recorded. Values in parentheses present the percentage of cells of the given activity pattern respectively to the number of analyzed cells and pairs of cells.

patient	cell count	pair count	mean firing rate	long burst	short burst	refractory period	0-2Hz	4-6 Hz	10Hz	correlation
				cell number (%)						pair num. (%)
AG	25	44	22.6	5 (20%)	0 (0%)	9 (36%)	10 (40%)	1 (4%)	9 (0%)	5 (11%)
BS	13	19	14.5	4 (33%)	1 (4%)	5 (38%)	7 (50%)	0 (0%)	5 (8%)	3 (16%)
CB	49	105	19.9	1 (2%)	3 (6%)	5 (10%)	3 (6%)	0 (0%)	5 (10%)	5 (5%)
AC	35	48	14.2	3 (9%)	8 (23%)	13 (37%)	20 (57%)	2 (6%)	13 (11%)	14 (29%)
EC	28	32	14.4	0 (0%)	2 (7%)	6 (21%)	8 (29%)	0 (0%)	6 (4%)	10 (31%)
BM	33	46	26.9	3 (9%)	7 (21%)	15 (45%)	4 (12%)	0 (0%)	15 (9%)	13 (28%)
Total	183	294	19.27	16 (9%)	21 (11%)	53 (29%)	52 (28%)	3 (2%)	53 (29%)	50 (17%)

Table 8.2

Neuronal activity per patient in SNr. Each row corresponds to an individual patient. Values in the tables show the absolute number of cells on which the given activity pattern was recorded. Values in parentheses present the percentage of cells of the given activity pattern respectively to the number of analyzed cells and pairs of cells.

patient	cell count	pair count	mean firing rate	long burst	short burst	refractory period	0-2Hz	4-6 Hz	10Hz	correlation
				cell number (%)						pair num. (%)
AV	25	44	8.8	2 (8%)	2 (8%)	1 (4%)	1 (4%)	1 (4%)	2 (8%)	4 (9%)
MG	80	324	8.3	27 (34%)	3 (4%)	27 (34%)	24 (30%)	0 (0%)	5 (6%)	23 (7%)
Total	105	368	8.66	29 (28%)	5 (5%)	28 (27%)	25 (24%)	1 (1%)	7 (7%)	27 (7%)

Table 8.3

Neuronal activity per patient in GPi. Each row corresponds to an individual patient. Values in the tables show the absolute number of cells on which the given activity pattern was recorded. Values in parentheses present the percentage of cells of the given activity pattern respectively to the number of analyzed cells and pairs of cells.

zone	cell count	pair count	mean firing rate	long burst	short burst	refractory period	0-2Hz	4-6 Hz	10Hz	correlation
				cell number (%)						pair num. (%)
STN	950	1682	18.68	113(12%)	69 (7%)	117 (12%)	194(20%)	77 (8%)	63 (7%)	329 (20%)
SNr	183	294	15.27	16 (9%)	21 (11%)	53 (29%)	52 (28%)	3 (2%)	53 (29%)	50 (17%)
GPi	105	368	8.66	29 (28%)	5 (5%)	28 (27%)	25 (24%)	1 (1%)	7 (7%)	27 (7%)

Table 8.4

Summary table of neuronal activity in STN, SNr, GPi.

In addition for 7 Parkinsonian patients the results obtained for STN are grouped according to the initially operated hemisphere (side I) and the other hemisphere (side II) (Table 8.5).

Patient	side	cell count	pair count	mean firing rate	long burst	short burst	refractory period	0-2Hz	4-6 Hz	10Hz	correlation
					cell number (%)						pair num. (%)
JC	I	60	98	19.3	6 (10%)	0 (0%)	4 (7%)	8 (13%)	5 (8%)	8 (13%)	9 (9%)
	II	21	20	16.5	3 (14%)	5 (24%)	1 (5%)	6 (29%)	7 (33%)	4 (19%)	3 (15%)
LS	I	39	38	21.0	7 (18%)	4 (10%)	10 (26%)	7 (18%)	7 (18%)	0 (0%)	5 (13%)
	II	21	44	20.2	0 (0%)	2 (10%)	3 (14%)	0 (0%)	0 (0%)	0 (0%)	13 (30%)
DS	I	72	112	21.5	1 (1%)	13 (18%)	3 (4%)	13 (18%)	14 (19%)	0 (0%)	24 (21%)
	II	51	83	25.5	0 (0%)	8 (16%)	0 (0%)	4 (8%)	2 (4%)	3 (6%)	10 (12%)
AT	I	46	68	28.6	2 (4%)	0 (0%)	3 (7%)	6 (13%)	3 (7%)	1 (2%)	9 (13%)
	II	72	137	28.2	0 (0%)	1 (1%)	2 (3%)	2 (3%)	1 (1%)	5 (7%)	80 (58%)
RW	I	39	73	22.9	5 (12%)	2 (4%)	3 (8%)	2 (4%)	2 (4%)	3 (8%)	9 (13%)
	II	25	44	20.7	1 (4%)	1 (4%)	2 (8%)	2 (8%)	0 (0%)	1 (4%)	6 (14%)
CG	I	33	46	19.9	7 (21%)	6 (18%)	2 (6%)	12 (36%)	2 (6%)	1 (3%)	11 (24%)
	II	27	44	21.5	0 (0%)	1 (4%)	1 (4%)	0 (0%)	0 (0%)	0 (0%)	3 (7%)
RS	I	10	25	26.7	0 (0%)	5 (50%)	1 (10%)	3 (30%)	0 (0%)	0 (0%)	0 (0%)
	II	30	47	18.2	0 (0%)	8 (27%)	0 (0%)	1 (3%)	0 (0%)	4 (13%)	20 (43%)

Table 8.5

Activity on the initially operated hemisphere (I) and the second operated hemisphere (II) in STN. Each row corresponds to a brain hemisphere of an individual patient. Values in the tables show the absolute number of cells on which the given activity pattern was recorded. Values in parentheses present the percentage of cells of the given activity pattern respectively to the number of analyzed cells and pairs of cells.

8.4 Statistical tests

A statistical test was applied to evaluate the variations of the activity pattern representation on different patients and in different brain structures. Let us denote θ^1 the probability, that neuron activity recorded under condition I corresponds to type A and θ^2 – the same probability for the activity of a neuron recorded under condition II (by condition we mean a patient or a brain structure or a brain hemisphere). For evaluation of the statistical likelihood of the assumption about difference of neuronal activity under different condition let

us verify the hypothesis: $H^0 : \theta^1 = \theta^2$. For a neuronal activity type A we have a set of random values $\{x_i^1\}$, observed under condition I and a set of random values $\{x_i^2\}$ observed under condition II. Sets $\{x_i^1\}$ and $\{x_i^2\}$ follow a binomial distribution: $p(x_i^j) = B(\theta^j, \theta^j(1-\theta^j)n_i^j)$, i=1,2; n_i – sets lengths.

Since statistical samples on the patients and on the structures have different sizes the bootstrap method was applied. The histograms for each activity pattern were compared using the Kolmogorov-Smirnov test of uniformity of two distributions (Ivchenko, press. 1984). The advantage of this test is that it doesn't suppose any assumption about the data distribution.

Statistical tests were applied for 4 patients with sample set longer then 100 units (table 8.1). On each bootstrap simulation their sets for each activity pattern were resampled by means of bootstrap method. The simulation was repeated 10 times. On each simulation a 10 sample data set were generated. Table 8.6 shows the confidence levels of the hypothesis of difference of frequencies of each activity patterns for different patients $H^0 : \theta^1 \neq \theta^2$.

	long burst			**short burst**			**refractory period**		
Patients	1	2	3	1	2	3	1	2	3
1									
2									
3	**S****	**S****		**S****	**S****		S*	S*	
4	**S****	**S****				**S****	**S****	**S****	
	oscillations [0;2]Hz			**oscillations [4;6]Hz**			**oscillations 10Hz**		
Patients	1	2	3	1	2	3	1	2	3
1									
2									
3									
4	S*		S*		S*			S*	

Table 8.6

Confidence levels of the hypothesis that the frequencies of an activity pattern are different on individual patients $\left(H^0 : \theta^1 \neq \theta^2\right)$, estimated by means of the Kolmogorov-Smirnov test. The results are grouped by neuronal activity patterns. For the analysis the data recorded on the following 4 patients were used (see table 8.1):
Patient 1 – AG ; Patient 2 – CB ; Patient 3 – DS ; Patient 4 – AT.

The same procedure was performed for the data grouped by brain structures and the results are shown in table 8.8.

	STN-SNr	SNr-Gpi	STN-Gpi
long burst			
short burst		S*	
refractory period			S*
Oscillations 0-2Hz			
Oscillations 4-6Hz	S*		**S****
Oscillations 10Hz			

Table 8.7
Confidence levels of the hypothesis that the frequencies of an activity pattern are different in brain structures of STN, GPi and SNr $\left(H^0 : \theta^1 \neq \theta^2\right)$, estimated by means of the Kolmogorov-Smirnov test.

The results were also compared for the initially operated and the secondly operated brain hemispheres. No statistical significant differences were found. However The confidential level of the hypothesis $H^0 : \theta^1 \neq \theta^2$ equal to 0.7 was obtained for the long burst and oscillations of [1-2]Hz.

We applied also a more sophisticated Fisher test to compare some of observed samples. Here we hade to take into account that statistical data is represented by samples of different size. For example only 183 cells were observed in SNr and 950 of them in STN. Some mathematical hints (Appendix C) allowed using the Fisher test as a test of uniformity allowing comparison of differently represented sets without a loss of information and provides more accurate results. For a correct application of the test there is a constraint on the sample size (see Appendix C):

$$n \geq 1.8 \frac{1}{\theta(1-\theta)},$$

where θ is the mathematical expectation of the analyzed activity pattern. The mathematical expectation may be estimated by the mean value and can be taken from tables 8.1-8.5. According with this formula rather large samples are needed and the test can't be applied here

only for a comparison of probabilities of Refractory Period, Oscillations [0,2]Hz and Oscillations [4,6]Hz on STN and SNr. The results of the test are the following:

	STN-SNr
refractory period	>0.92
Oscillations 0-2Hz	very low
Oscillations 4-6Hz	>0.96

Table 8.8
Results of the Fisher test estimation of confidential levels of the hypothesis that the frequencies of an activity pattern are different in brain structures of STN and SNr $(H^0 : \theta^1 \neq \theta^2)$.

8.5 Discussion

The presence of bursting and oscillatory activity in the STN is typical for Parkinsonian patients and MPTP treated animals (Bergman, 1994; Rodriguez-Oroz et al, 2001). Here was considered only low frequency band, namely [0-10]Hz. The analyses showed that the frequencies of these bands are not represented equally. Thus this band was divided into 3 bands, one of them is the tremor frequency band - [4-6] Hz. Oscillations of different frequency bands were considered independently. The same procedure was applied for the analysis of burst duration. Short burst should correspond to the usual understanding of this term while the long burst corresponds to long (>30ms) fluctuations in firing intensity, which is represented by condensations and on the cell raster. If the condensations appear regularly the cell oscillates at the same time on a low frequency.

The mean firing rate of STN cells was 21+/-13 spikes/s. This result is distinct of those previously described. According to other data in PD patients, single STN neurons have a mean firing rate of 42.30 +/- 22.00 spikes/s (Benazzouz 2002), or 33+/-17 spikes/s (Hutchison et al., 1998). We do not exclude that this discrepancy is the result of difference in separation of spike trains of individual neurons.

The mean number of separated individual neurons during the analysis was 4-5 per single channel record. Thus this value varied significantly. On some records only 2-3 (and

more rare only one) individual neurons were found, on others up to 7-8 neurons could be distinguished.

Table 8.6 shows that the neuronal activity profile varies from patient to patient. The statistically significant differences between patients may be due to the physiological particularities of each patient or/and to the etiology and progression of the disease. This second assumption was studied and the results of the study are presented in chapter 9 of the present thesis.

The comparison of brain structures neuronal activity patterns (table 8.7) showed that GPi neuronal activity differs from SNr and STN activity. From the other side, the statistical material for GPi analysis was insufficient to draw any a conclusion (105 cells from only 2 patients). STN and SNr, in turn, differ significantly only in probability of refractory period ($P>0.8$) and of 4-6 Hz oscillations ($P>0.95$). STN has more 4-6 Hz oscillations than SNr and GPi. This frequency often corresponds to tremor.

The results of these analyses show also that the percentage of bursting and low frequency oscillating cells has a tendency to be more abundant in the first operated side, even though this difference was not statistically significant.

The application of the Fisher test where it was possible confirmed the results the Kolmogorov-Smirnov test (table 8.8). This fact can be considered as a validation of the applied statistical methods and of the results presented here.

Chapter 9

Spike waveform analysis

Les études de l'activité neuronale basées sur le tri de potentiels d'action à l'aide de l'USS ont révélé que les formes des potentiels sont sensiblement les même pour tous les patients. Ce chapitre décrit les analyses effectuées pour définir les formes typiques de la décharge neuronale et leur répartition NST, SNr et GPi. Cinq types de formes ont été définis suite à l'application de la procédure de clustering du USS ont été considérés.

Dans ce chapitre nous analysons les relation entre les différentes formes de potentiel d'action et les de patterns d'activité neuronale. Quelques différences de patterns ont été trouvées entre deux de ses types. L'étude de répartition de ses types dans NST, SNr et GPi a montré une différence de représentation de certains types dans ses structures.

9.1 Typical Spike Waveforms (TW)

The studies of neuronal activity based on spike sorting by means of the USS revealed that the waveform of some neuronal discharges seems to be reproducible on electrophysiological records from different patients. We assumed then that there existed several TW and that the spike's waveform of a firing neuron usually corresponds to one of those, thus suggesting that a neuron may be classified according to spike's waveform.

The USS is a template matching spike sorting technique and saves template waveforms for each detected single unit at the learning stage. The single unit template waveforms provided by the USS are denoted as "templates". All templates generated during the spike train analysis were stored in USS internal files. Since the spike amplitude depends, among other factors, on the distance between the recording electrode and the cell membrane, the collected templates were normalized according to the amplitude. A clustering procedure is applied to the cumulated a set of templates and a representative waveform is constructed for each cluster. The representative waveforms will form a TW set such that a TW is the center of

a cluster of templates, i.e. common for several of single units. All typical waveforms form a typical spike waveform library (TWL).

The same routine may be carried out on two independent sets of templates S1 and S2, forming a TWL for each of them (TWL1 and TWL2) to validate the results of clustering. Then, the templates from each set should be classified according to TWL1 and to TWL2. The results of all four classifications should be compared. The final TWL may be constructed of waveforms of TWL1 and TWL2 taking into account the classes overlapping of classes.

The general scheme of the analysis carried out for TW definition was the following:

1. Single units were extracted with USS from multiunit records from STN GPi and SNr.
2. The template single unit spikes generated by USS for each single unit were normalized and stored.
3. All stored template spikes were organized in 4 sets: 2 independent sets S1 and S2 from STN, a GPi set and an SNr set.
4. A TWL was generated for each set of templates.
5. Each TWL was used to classify the corresponding training and test data sets. The percentages of templates in the classes and an amount of non-classified templates were calculated and compared.
6. To compare the result of the classifications the percentages of overlapping were also calculated between the classes belonging to different TWLs.
7. A final common TWL is formed by the waveforms of all 4 TWL.

A further analysis studied the relation between the spike's waveforms and the activity patterns defined by bursting, frequency oscillatory activity and refractory period.

9.2 Results

709 STN, 294 Gpi and 186 SNr single unit templates were stored. The STN template set was split into 2 sets STN1 and STN2 of 453 and 256 templates. The appropriate procedures of the USS were applied for clustering and classification. The clustering procedure of USS estimates the cluster center and the radius for each cluster. The cluster centers were

considered as typical waveforms and radiuses were used for further classification. The USS parameters for clustering and classification were fit once and stayed unchanged through the analysis.

After clustering the following results were obtained: 10 clusters for STN1; 6 clusters for STN2; 9 clusters for GPi; 10 clusters for SNr. Clusters constructed by USS are shown on figures 9.1-9.4. Clusters are numbered through all 4 sets with a unique cardinal. In each cluster a TW was defined. The TWs are shown on the figures as bold lines inside each cluster. Images in each frame of the figures correspond to the cluster with the number marked in the upper left corner and represent both spike waveforms in time domain and spike trajectory in the pase space (2.3) used by USS for clustering and classification procedures.

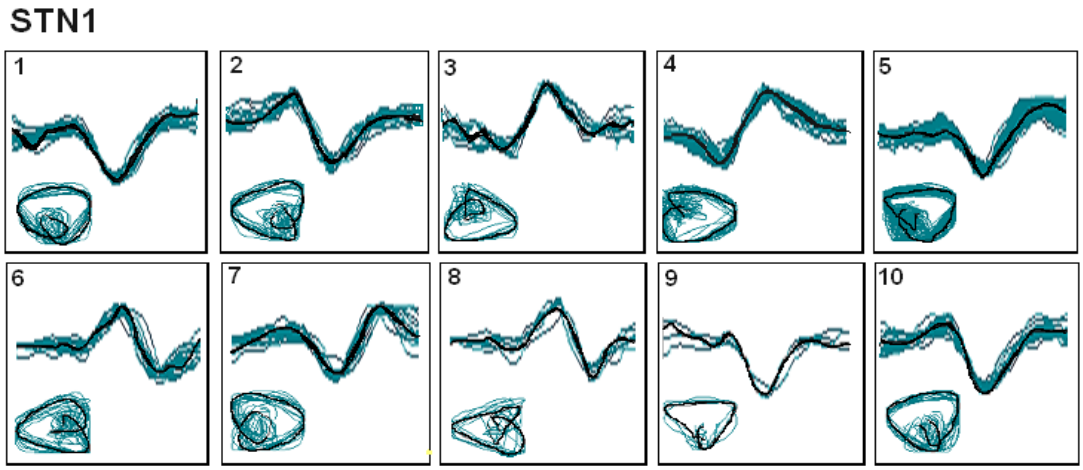


Figure 9.1
Clusters and TWs obtained from the GPi data set.

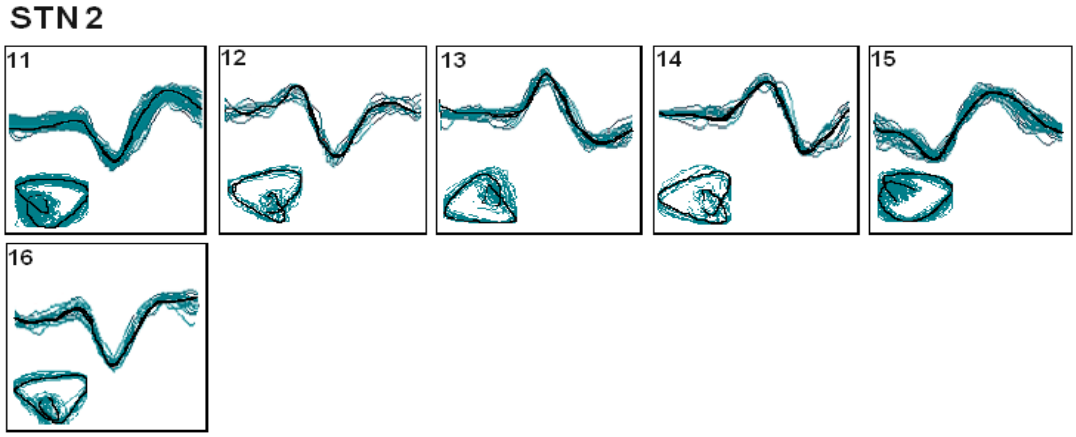


Figure 9.2
Clusters and TWs obtained from the second independent data set in STN.

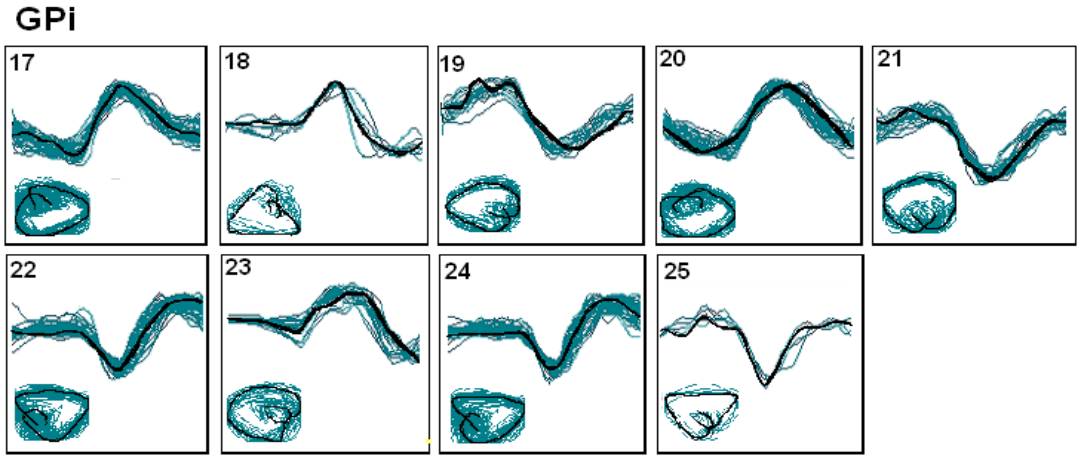


Figure 9.4
Clusters and TWs obtained from the GPi data set.

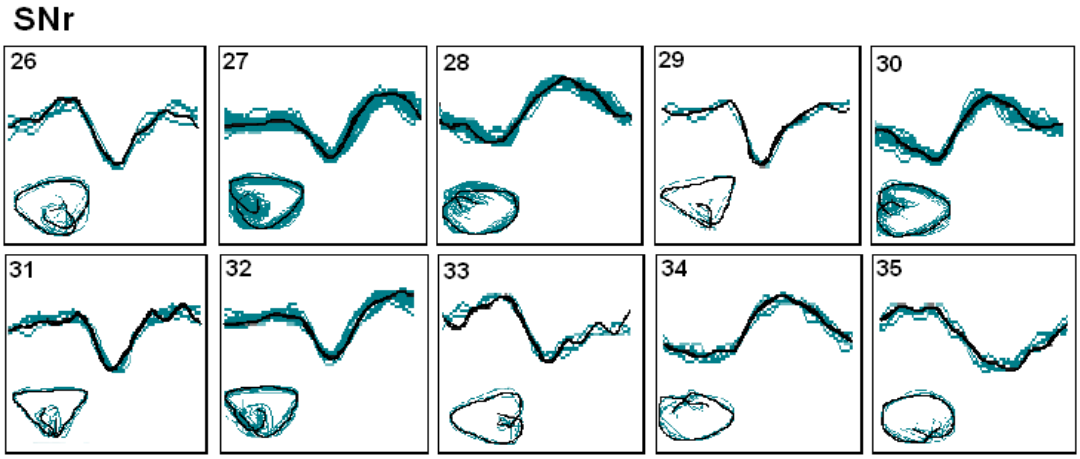


Figure 9.4
Clusters and TWs obtained from the SNr data set.

All template sets were classified then according to all constructed TWL, i.e STN1 set was classified 4 times according to its own STN1 TWL and also according to ST2 TWL, GPi TWL and SNr TWL. The same was done with STN2, GPi and SNr sets. The classes' overlapping is shown in table 9.1. The meaning of table values x_{ij} is following: x_{ij} % of templates of i-th class obtained after classification of a template set on its own TWL belonging also to the j-th class after classification on an another template set TWL. The table values greater the 80% are marked with dark gray and the values between 40% and 80% are marked with light grey.

		STN1										
		0	1	2	3	4	5	6	7	8	9	10
STN1	0											
	1											
	2											
	3											
	4											
	5											
	6											
	7											
	8											
	9											
	10											
STN2	0	33%		17%	24%							
	11						68%		30%			
	12			44%								53%
	13				30%			60%				
	14							87%				
	15				30%	67%						
	16		55%				28%					17%
Gpi	0	73%										
	17				30%	70%						
	18							73%		18%		
	19			87%								
	20				68%	22%						
	21		35%	37%								16%
	22						85%					
	23	64%						29%				
	24						80%					
	25			73%							18%	
SNr	0	31%					15%	23%				
	26			63%								25%
	27						92%					
	28					100%						
	29			43%							43%	
	30					100%						
	31		38%								63%	
	32						86%					
	33			100%								
	34				77%	23%						
	35	50%		50%								

		STN2						
		0	11	12	13	14	15	16
STN1	0	80%					20%	
	1							31%
	2			97%				
	3				33%		52%	
	4						93%	
	5		74%					26%
	6				47%	53%		
	7		92%					
	8				57%	43%		
	9	20%	30%	20%				30%
	10			92%				
STN2	0							
	11							
	12							
	13							
	14							
	15							
	16							
Gpi	0	100%						
	17	38%					61%	
	18				64%	27%		
	19	39%		61%				
	20	22%					65%	
	21			63%				26%
	22		54%					38%
	23	29%			57%			
	24	24%	76%					
	25			100%				
SNr	0	46%			15%	19%		
	26			100%				
	27		95%					
	28	20%					80%	
	29			100%				
	30	50%					50%	
	31							75%
	32		29%					71%
	33			100%				
	34	38%					54%	
	35	67%		33%				

		Gpi									
		0	17	18	19	20	21	22	23	24	25
STN1	0	80%				20%					
	1						91%				
	2				52%		28%				21%
	3		67%								
	4		90%								
	5							56%		37%	
	6								20%		
	7							42%		58%	
	8			86%							
	9						20%			30%	50%
	10						92%				
STN2	0	64%									
	11										
	12	67%									62%
	13			46%							
	14				19%						
	15										
	16				63%						
Gpi	0										
	17										
	18										
	19										
	20										
	21										
	22										
	23										
	24										
	25										
SNr	0	19%		19%				19%	23%		
	26						63%				25%
	27							40%		59%	
	28		80%			20%					
	29										100%
	30		95%								
	31										88%
	32							79%			
	33				83%						17%
	34					92%					
	35				33%		67%				

		SNr										
		0	26	27	28	29	30	31	32	33	34	35
STN1	0	100%										
	1	64%							27%			
	2		38%							41%		
	3	67%									26%	
	4						83%					
	5			74%					23%			
	6	100%										
	7	50%		50%								
	8	100%										
	9	20%		30%		20%						
	10	25%	67%									
STN2	0	30%										
	11	37%		44%					51%			
	12											
	13											
	14		23%									
	15				42%							
	16											
Gpi	0	73%										
	17						63%				17%	
	18	100%										
	19									71%		19%
	20				13%						68%	
	21		51%						16%			26%
	22								33%			
	23	93%										
	24	16%		80%								
	25					55%		18%		27%		
SNr	0											
	26											
	27											
	28											
	29											
	30											
	31											
	32											
	33											
	34											
	35											

Table 9.1

Overlapping of crossclassification of each data set respectively to the classification on the own data set's TWL. Table values X_{ij} mean that X_{ij} % of templates of i-th class obtained after classification of a template set on its own TWL belong also to the j-th class after classification on an another template set TWL. The table values greater the 80% are marked with dark gray and the values between 40% and 80% are marked with light grey.

The same information is presented by graphs on figure 9.2. For each pair of data sets the overlapping between classes obtained on the own data set's TWL and on the TWL of another data set was considered. If more then 40% of the elements of class A are members of class B the classes are connected with a dotted arrow directed from class A to class B. If the overlapping is greater then 80% the classes are connected with a solid arrow.

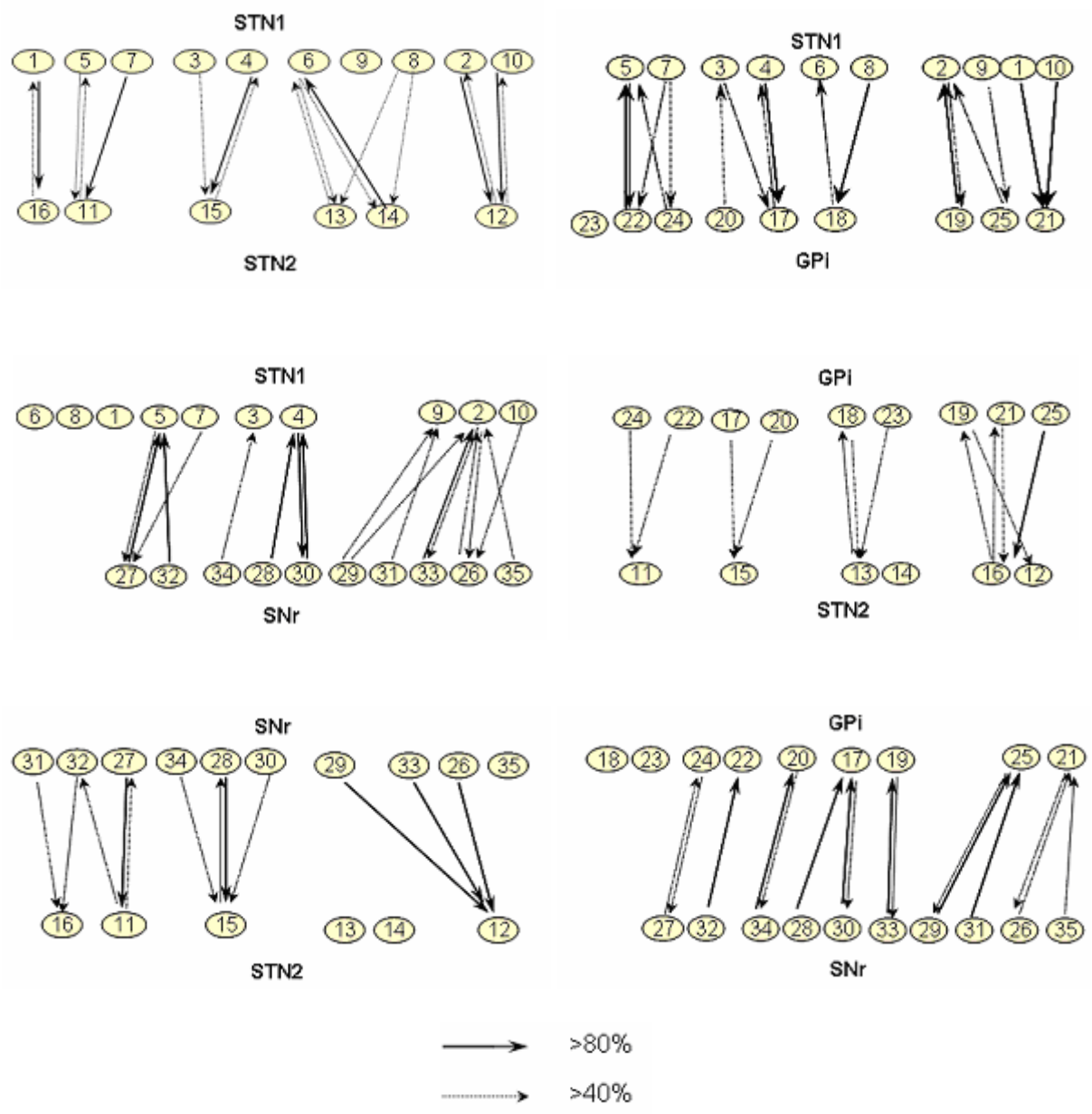


Figure 9.5

Overlapping of crossclassification of each set respectively to the classification on its own TWL. If overlapping is greater than 80% the classes connected with a solid arrow. If the overlapping is greater than 40% and less or equal to 80% the classes are connected with a dotted arrow.

Table 9.1 and figure 9.5 show that all classes corresponding to the 35 TWs can be split into 5 nearly non overlapped groups of classes:

Type 1: 3 4 15 19 21 25 27 30
Type 2: 1 9 16 22
Type 3: 2 10 12 17 20 24 26 29 31 35
Type 4: 6 8 13 14 28 33
Type 5: 5 7 11 18 23 32 34

The distributions of these 5 types in different patients and in different brain structures were calculated using 609 STN cells from 7 Parkinsonian patients, 183 SNr cells from 6 Parkinsonian patients and 105 GPi cells from 2 dystonia patients. Table 9.2 shows how the types are distributed in the STN of 11 patients. The analysis per patient in SNr and GPi were not carried out because of statistical insignificance of the data. The distributions of the 5 Types in STN, SNr and GPi were studied and compared overall (tables 9.3, 9.4).

name	cell count	non identified	type 1	type 2	type 3	type 4	type 5
JC	81 (100%)	2 (2%)	29 (36%)	12 (16%)	13 (15%)	9 (11%)	16 (20%)
AG	131 (100%)	1 (1%)	49 (37%)	5 (4%)	28 (21%)	22 (17%)	26 (20%)
BS	100 (100%)	5 (5%)	18 (18%)	8 (8%)	14 (14%)	11 (11%)	44 (44%)
CB	107 (100%)	7 (7%)	52 (48%)	3 (3%)	29 (27%)	3 (3%)	13 (12%)
GB	66 (100%)	2 (3%)	42 (63%)	4 (8%)	12 (18%)	3 (55%)	3 (3%)
RW	64 (100%)	4 (6%)	23 (37%)	4 (6%)	12 (19%)	6 (9%)	15 (23%)
CG	60 (100%)	2 (3%)	19 (32%)	5 (8%)	7 (12%)	8 (13%)	19 (32%)
Total	**609 (100%)**	**23 (3%)**	**232 (39%)**	**41 (7%)**	**115 (19%)**	**62 (10%)**	**136 (22%)**

Table 9.2
Distribution of neurons of five Types in STN. Each row corresponds to an individual patient. Cell values represent the number of cells' Type units, values in parenthesis present the percentage of cells of the given Type respectively to the number of analyzed cells.

structure	cell count	non identified	type 1	type 2	type 3	type 4	type 5
STN	609 (100%)	19 (3%)	232 (38%)	49 (8%)	103 (17%)	64 (10%)	144 (24%)
SNr	183 (100%)	5 (3%)	62 (34%)	7 (4%)	30 (16%)	10 (5%)	69 (38%)
Gpi	105 (100%)	3 (3%)	43 (40%)	3 (3%)	21 (20%)	5 5(%)	30 (29%)
Total	**897** (100%)	**27** (3%)	**337** (37%)	**59** (7%)	**154** (17%)	**77** (9%)	**243** (27%)

Table 9.3

Distribution of neurons of five types on brain structures of STN, SNr and GPi. Cell values represent the number of cells' Type units, values in parenthesis present the percentage of cells of the given Type respectively to the number of analyzed cells.

The comparison of the "Type" distribution between structures and patients was done by assuming θ^1 the probability of type A on structure(patient) I and θ^2 – the same probability on structure(patient) II. For evaluation of the statistical likelihood of the assumption about the difference of type A probabilities the zero hypothesis is $H^0 : \theta^1 = \theta^2$. The bootstrap method and the Kolmogorov-Smirnov test were employed as it was described in chapter 7. Tables 9.3 and 9.4 show the confidential levels of the hypothesis for brain structures and individual patients respectively.

	STN-SNr	SNr-Gpi	STN-Gpi
Type 1			
Type 2	S*		**S****
Type 3		S*	
Type 4	**S****		S*
Type 5			

Table 9.4

Confidence levels of the hypothesis of difference of frequencies of activity patterns $(H^0 : \theta^1 \neq \theta^2)$ estimated with the Kolmogorov-Smirnov test.

Application of the Fisher test (Appendix D) was possible only for comparison of probabilities of Type 1 and Type 5 on STN and SNr. The confidential level obtained for Type 1 was very low and for Type 2 it was equal to 0.8 which confirms the results obtain with application of the Kolmogorov-Smirnov test.

Table 9.4 shows the probability of activity patterns defined in chapter 7 of each type, i.e. the conditional probability $P(pattern|type)$. To have a better statistics we took into account all cells recorded during DBS surgeries of Parkinsonian patients independently of the brain structure. Thus, we assume here that "WT-activity pattern" dependency doesn't change in different brain structures. 1282 cells of 14 Parkinsonian patients were analyzed. After waveform classification 50 cells were rejected as unclassified. Figure 9.6 represents the same information in graphical form for the 3 most representative types: Type 1, Type 3 and Type 5.

type	cell count	long burst	short burst	ref. per.	0-2Hz	4-6 Hz	10Hz
1	466	47 (10%)	33 (7%)	61 (13%)	93 (20%)	37 (8%)	33 (7%)
2	77	3 (4%)	9 (12%)	6 (8%)	17 (22%)	8 (10%)	10 (13%)
3	250	18 (7%)	13 (5%)	5 (2%)	23 (9%)	15 (6%)	15 (6%)
4	106	16 (15%)	2 (2%)	5 (5%)	24 (23%)	8 (8%)	6 (6%)
5	333	63 (19%)	37 (11%)	77 (23%)	103 (31%)	37 (11%)	20 (6%)
total	**1232**	**148 (12%)**	**99 (8%)**	**160 (13%)**	**271 (22%)**	**111 (9%)**	**86 (7%)**

Table 9.5
Number of cells having one of the neuronal activity patterns among cells of a given WT. In parenthesis - the percentage of such cells or the conditional probability $P(pattern|type)$.

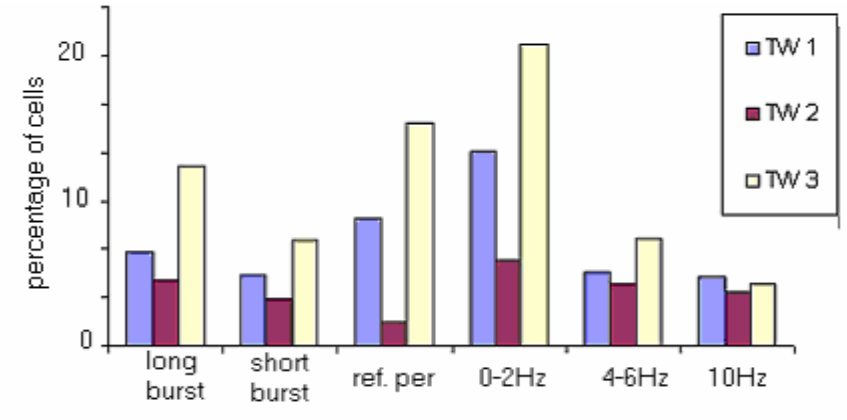


Figure 9.6
Histograms for the neuronal activity patterns for waveform Types 1, 3 and 5 (table 9.5).

The comparison of neuronal activity pattern for these 3 types using the Kolmogorov-Smirnov test gave the following results (table 9.6):

	TW1-TW3	TW3-TW5	TW1-TW5
long burst	S*		
short burst			
refractory period	S**	S**	
Oscillations 0-2Hz	S*		
Oscillations 4-6Hz			
Oscillations 10Hz			

Table 9.6
Confidence levels of the hypothesis of difference of frequencies of activity patterns $(H^0 : \theta^1 \neq \theta^2)$ estimated using the Kolmogorov-Smirnov test.

9.3 Discussion

Neuronal spikes usually have a typical waveform consisting of a rapid positive going component that lasts a few hundred microseconds, followed by a slower negative going component that may last 2–3 times as long (Fee et al, 1996). This waveform nevertheless varies according to the distance and the characteristics of the recording microelectrode. This makes it possible to separate spikes belonging to different units from a multiunit recording. If the microelectrode moves, then the spike wave form should change continuously. Let us assume that neuronal discharges vary also as a function of the neural cell type or of the systemic state or other circumstances. We assume that this variation is more significant than the variation of the form due to the microelectrode position and that it should be possible to group spike waveforms. If we assume that the recording conditions are standardized it is possible to postulate that similar waveforms could be identified in different patterns.

We should notice here that the neuron firing is a non-linear process depending of a great number of parameters including physiological type of the cell. We have not suggested a model describing the spike waveform variation as a function of neuron physiological type. Thus, it is impossible so far to find correspondences between known physiological types and

defined here waveform types. The proposed study is an attempt to show that spike waveforms of different neurons recorded on different patients can be grouped and the groups don't correspond to individual patients. This study can not answer to the question about the nature of the tendency of spike waveforms to form cluster.

Clustering and classification carried out for two independent homogenous sets of STN revealed five groups whose overlapping did not exceed 40%. Two more data sets were added, one from SNr and one from GPi. The classes obtained could still be divided into five non overlapping groups (figure 9.2).

The distribution of waveform types was studied on individual patients. The results of the study showed that all defined waveform types are present in all patients, although the types may be represented differently (table 9.2). Thus, the difference in spike waveforms can not be considered as patient individual physiology. The study of the distribution of waveform types in brain structures of STN, SNr and GPi showed that in spite of the presence of all types in all structures, there is some difference between structures. We have to notice here that the GPi statistical data was insignificant for final conclusions. The difference between STN and SNr was verified statistically with the Kolmogorov-Smirnov test and then verified by the Fisher test where it was applicable. The statistical verification showed a significant difference in probability of some waveform types.

A very impotent part of the study is the correspondence between waveform types and neuronal activity patterns. For this analysis the three most significant Types were selected. The analysis showed no difference in firing patterns for two of these Types. The third of Type was significantly different with respect to Long Burst, Refractory Period and Oscillations [0,2]Hz.

Chapter 10

PD symptoms and neuronal activity and patterns

Les études de l'activité neuronale exposées dans le chapitre 7 ont démontré que les patterns d'activité peuvent apparaître différemment chez les patients individuels. Dans ce chapitre nous examinons comment l'activité neuronale varie en fonction de la gravité des symptômes parkinsoniens et vice versa. Un algorithme basé sur la méthode de réseau de neurones artificiels est appliqué pour chercher des dépendances entre les niveaux de gravité des symptômes cliniques relevés au cours des suivis neurologiques des patients et les probabilités d'apparition des patterns d'activité neuronale définis et estimés dans le chapitre 7. Les résultats des calcules montrent les dépendances positives entre les patterns d'activité et les symptômes cliniques suivent : Régidité - Burst Court ; Trémor – Période Réfractaire ; Trémor – Oscillations de 4-6Hz ; Akinesia - Oscillations de 4-6Hz ; Akinésie – Corrélation ; Bradikyneésie - Corrélation.

The results of the neuronal activity patterns analysis in STN showed a significant difference in probability of certain activity patterns on individual patients. This difference may be due to the individual physiology and to the intensity of the movement disorder in each patient. Parkinson Disease is indeed characterized by a number of symptoms which may be more or less expressed and even absent on different patients. Here is described an attempt to correlate the symptomatic characteristics from patient's monitoring and the frequencies of neuronal activity patterns defined in chapter 7.

10.1 PD symptoms

The information about the disease evaluation was taken from the department of Neurology of the University hospital of Grenoble which followed the Parkinsonian patients

operated in the Department of Neurosurgery of the same hospital. During the monitoring of the evaluation of PD the symptoms are scored under different conditions: on/off medications and on/off stimulation on the pre- and postoperative stages. To estimate the gravity of PD symptoms the neurologists of the department use the standard Unified Parkinson Disease Rating Scale (UPDRS). The symptoms are scored from 0 to 4:

0=Absent
1=Slight
2= Mild
3=Moderate
4=Marked

Four dysfunction types are scored: tremor, rigidity, repetitive movement dysfunction and bradikinesia. The evaluation of these dysfunction types, except bradikinesia, includes several scorings presented in table 10.1 estimated on the left and the right side separately. The official UPDRS contains only 5 steps, nevertheless neurologist often use "+" notation to increase the scale resolution (for example 3+ means that the symptom gravity is between 3 and 4). Values in the table adapt this additional notation by adding 0.5 to the symptom score if it was "+" marked by neurophysiologist. Repetitive movements' dysfunction shows the akinesia gravity.

Tremor			**Rigidity**			**Repetitive movements**			**Bradikinesia**
	right	**left**		**right**	**left**		**right**	**left**	
upper limb	0.0	0.0	upper limb	3.0	1.5	index-thumb	3.0	2.0	2.0
inferior limb	1.0	0.0	inferior limb	2.0	0.0	hands	3.0	1.0	
face	0.0		neck	3.0		"marionette"	3.5	2.0	
action	1.0	1.0				feet	3.5	1.5	
sum	2.0	1.0		8.0	4.5		13.0	6.5	2.0

Table 10.1
Example of a symptom scores sample according with the sample methodic used at the department of Neurology of the Grenoble University Hospital.

For the present study the motor symptoms scores were sampled before the surgical intervention and off-medication. The scores were added together by symptoms and hemispheres as it shown in table 10.1. The dysfunctions scores of neck regidity as well as

bradikenesia were included in the two side's sums. The neuronal activity characteristics were taken from the results of the neuronal activity patterns analysis described above in Chapter 7 of the present memoir (Tables 7.1, 7.4). Thus, for each patient we have a vector of activity patterns:

{ burst long,
burst short,
refractory period,
oscillations 0-2Hz,
oscillations 4-6Hz,
oscillations 10Hz,
firing rate,
correlation }

and a vector of symptoms scores:

{ tremor,
rigidity,
akinesia,
bradikinesia }.

The symptom scores on the left and the right hemispheres are different and the neuronal activity was considered separately for each hemisphere. The symptoms and the activity patterns were associated contralaterally, i.e. the scores evaluated for the left limbs movement dysfunctions were associated with the data recorded from the right brain hemisphere.

10.2 Modeling with Polynomial Artificial Network (PNN) algorithm

Iterative GMDH-type (Group Method of Data Handling) Robust PNN algorithm (Aksenova, 2005) has been employed to provide the analysis of the dependences between clinical symptoms and the types of neuronal activity that were found out in STN of PD patients. The GMDH approach (Modala, 1994; Yurachkovsky, 1981) for complex data sets analysis is aimed at determining internal data relationships, extracting relevant variables and presenting knowledge of these relationships in the parametric form of linear and nonlinear polynomial regression equations. Robust PNN provides robust linear and nonlinear polynomial modeling in the presence of outliers or/and correlated and irrelative variables. For a set of independent variables $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ and one dependent variable y the

purpose of the algorithm is to find a subset of relevant variables $\{x_{i1}, x_{i2}, ..., x_{ik}\}$ and a model $\mathbf{Y} = f(x_{i1}, x_{i2}, ..., x_{ik})$, where $f(x_{i1}, x_{i2}, ..., x_{ik})$ is a polynomial that fitted data the best way in sense of some criterion.

The main features of the PNN algorithm are: fast learning, results in the parametric form of the polynomial equation, complexity control and twice-hierarchical neural net structure. A common problem is that the power of polynomials increases too fast in the traditional GMDH algorithms. Twice-hierarchical neural net structure provides a wide search without an increase in complexity. This structure provides the convergence of the coefficients. As different from the traditional GMDH algorithms Robust PNN (Aksenova, 2003b) provides robust (M-regression) model identification that implements the algorithm in the presence of large errors (outliers) in vector **Y**.

For PNN analysis the data is presented as a matrix **X** of observations which correspond to the independent variables and a vector of observations **Y** corresponding to the dependent variable *y*. The fitness of models quality may be estimated by the coefficient r^2 of correlation between the observed and the values predicted by the model. The hypothesis is that **Y** depends on some components of matrix **X**. Thus, if for some model "**Y** predicted" and "**Y** observed" are correlated then we will consider that there is a dependences described by this model. If the error in vector **Y** has equal dispersions for each component and there is no error in matrix **X** the correlation coefficient r^2 may be used as a test verifying the hypothesis $H_0 : a_0 \times a_1 \times ... \times a_m \neq 0$, i.e. all the coefficients of the polynomial are not equal zero and thus the variable y is dependent. The value $r^2=0.43$ corresponds the hypothesis confidential level $p=0.95$.

10.3 Results

10.3.1 Data matrices

All patient data was organized as two matrixes. Matrix **Y** (table 10.2) recapitulated all the data related to patterns of temporal neuronal activity with variables:

Y = { y_1 – percent of long burst cells,
y_2 – percent of short burst cells,
y_3 – percent of cells with refractory period,
y_4 – percent of oscillating cells (0-2 Hz),
y_5 – percent of oscillating cells (4-6 Hz),
y_6 – percent of oscillating cells (10 Hz),
y_7 – mean firing rate,
y_8 – percent of correlated pairs}

The clinical symptoms formed matrix **X** (table 10.3), namely variables:

X = { x_1 – tremor,
x_2 - rigidity,
x_3 – akinesia,
x_4 – bradikinesia }

name	side	tremor	rigidity	akinesia	bradikinesia
AG	left	2.5	2	3	2.5
BS	right	6	8	12	3
CB	right	6.5	4	7	1.5
GB	right	2	8	13	2
LS	left	8	5	9.5	3
LS	right	5	4	8.5	3
DS	right	8	5	12	2
DS	left	6	4	8	2
AT	left	2	4	8	1
AT	right	2	2	8	1
RW	left	3	4	1.5	2
RW	right	3	5	1	2
CG	left	1.5	10	6.5	2
CG	right	0.5	9	5.5	2
DS	left	3	7.5	6.5	2.5
DS	right	0	8.5	10.5	2.5

Table 10.2
Matrix **X** of symptom scores sampled on left and right sides for each patient.

Nom	Site				Oscillations				
		burst long	burst short	refractory period	0-2Hz	4-6 Hz	10Hz	firing rate	correlation
AG	left	0.27	0.00	0.20	0.40	0.05	0.08	13.45	0.17
BS	right	0.28	0.06	0.18	0.43	0.21	0.12	18.51	0.43
CB	right	0.12	0.01	0.26	0.21	0.03	0.05	12.97	0.07
GB	right	0.06	0.09	0.13	0.13	0.06	0.09	14.95	0.26
LS	left	0.18	0.10	0.26	0.18	0.18	0.00	21.01	0.13
LS	right	0.00	0.10	0.14	0.00	0.00	0.00	20.19	0.30
DS	right	0.01	0.18	0.04	0.18	0.19	0.00	21.54	0.21
DS	left	0.00	0.16	0.00	0.08	0.04	0.06	25.48	0.12
AT	left	0.04	0.00	0.07	0.13	0.07	0.02	27.55	0.13
AT	right	0.00	0.01	0.03	0.03	0.01	0.07	28.23	0.59
RW	left	0.12	0.04	0.08	0.04	0.04	0.08	22.95	0.13
RW	right	0.03	0.03	0.08	0.05	0.00	0.03	20.74	0.14
CG	left	0.21	0.18	0.06	0.36	0.06	0.03	19.89	0.23
CG	right	0.00	0.04	0.04	0.00	0.00	0.00	21.45	0.06
DS	left	0.00	0.50	0.10	0.30	0.00	0.00	26.70	0.0
DS	right	0.00	0.27	0.00	0.03	0.00	0.13	18.20	0.44

Table 10.3

Matrix **Y** of neuronal activity. Values x_{ij} of this matrix are equal to $\lambda/\sqrt{n_j}$, where λ is the frequency of the pattern and n_j - number of the observed cells on the set, corresponding to a brain hemisphere of a patient.

10.3.2 Dependent symptoms

Matrix **X** of symptom scores is considered to be error free. Conversly, the matrix **Y** of activity patterns contains an error. Since the patient's statistical samples are of different size the dispersion of error for the **Y** components is different. This error is corrected by normalization of the matrix **Y**: each matrix component was divided by $\sqrt{n_j}$, n_j is the number of observations for any *j*-th patient.

The Robust PNN algorithm software of PNN Discovery Client 1.3 (http://www.pnn.com.ua/) was used to construct models of dependences between variables of **X** and **Y**. The first part of the analysis included estimation of all dependences $y_i = f(x_j,\ldots,x_k)$, where y_i is the i-th activity pattern vector for one brain hemisphere of a patient. The PNN Discovery Client estimates a few variants of polynomial models in each case. We considered the 3 best models with criteria $r^2 > 0.43$. This coefficient threshold correspond to the

confidential level 0.95 for the hypothesis of existence of a dependency between vector y_i and matrix $\mathbf{X}$. Table 10.4 presents selected models for each considered activity pattern.

activity pattern	models	r^2	dependent symptoms
Y1 Long Burst	0.015799*X1+0.001159*(X2**2)-0.000713*(X3**2)	0.58	**X1 X2 -x3**
	0.018488*X1-0.009174*X3+0.001395*(X2**2)	0.56	
	0.011853*X2+0.001757*(X1**2)-0.00075*(X3**2)	0.55	
Y2 Short Burst	-0.055195+0.01321*X1+0.013959*X2	0.65	**X1 X2**
	0.008383*X1+0.008144*X2	0.65	
	0.012114*X1+0.011507*X2-0.016602*X4	0.63	
Y3 Refractory Period	0.013454*X1+0.026145*X4 0.00528	0.55	**X1 X4**
	0.010942*X1-0.006215*X2+0.046188*X4	0.57	
	0.015125*X1-0.003684*X3+0.035658*X4	0.57	
Y4 Oscillation 1-2Hz	0.030248*X1+0.049463*X2-0.112388*X4	0.79	**X1 X2**
	-193425+0.024914*X1+0.043958*X2	0.75	
	0.009577*X1+0.022959*X2	0.74	
Y5 Oscillation 4-6Hz	-0.053833+0.015631*X1+0.007268*X3	0.75	**X1 x3**
	0.016738*X1	0.76	
	0.012552*X1+0.002719*X3	0.73	
Y6 Oscillation 10Hz	0.05803-0.008746*X1+0.003422*X3	0.5	**-X1 X3 X4**
	0.079064-0.007412*X1	0.42	
	0.058119-0.008297*X1+0.011435*X4	0.45	
Y7 Firing Rate	31.104465-0.958834*X1-1.064893*X2	0.57	**-X1 -X2**
	28.427741-1.017846*X1-0.084013*(X2**2)	0.55	
	27.693659-0.814379*X2-0.075934*(X1**2)	0.46	
Y8 Correlation	-0.017489*X1+0.017338*X3+0.058796*X4	0.7	**X3 X4 –x1**
	0.013881*X3+0.040535*X4	0.63	
	-0.101291+0.015843*X3+0.077487*X4	0.63	

Table 10.4

Results of $y_i = f(x_j,\ldots,x_k)$ modeling by Robust PNN. The three best models with criteria $r^2 > 0.43$ were taken into consideration. The dependent symptoms are given in upper case if they appear more then once and with a significant coefficient in the models. Dependent symptom variables are given in the last column in lower case either if this variable's coefficients in the model are much smaller than other variables coefficients or if the variable appears only once in the 3 selected models. Otherwise the dependent symptom variables are given in upper case.

10.3.3 Dependent activity patterns

The Robust PNN is able to treat the presence of outliers in the dependent variable. Let us consider the dependences $x_i=f(y_j,\ldots,y_k)$, where $\mathbf{x}_i$ is the i-th symptom scores vector. Here the independent variable contains an error and the criteria r^2 can't be applied. Nevertheless the algorithm may still be used and these results may cross verify the first part analysis result. Like in the first part we considered the 3 best models suggested by the PNN Discovery Client, which are presented in table 10.5.

symptoms	models	$\mathbf{r^2}$	dependent activity patterns
X1 Tremor	(-)14.892275*Y1+25.485394*Y3+30.422389*Y5	0.82	**Y3 Y5 -y1 y7**
	11.174682*Y3+19.229979*Y5+0.068835*Y7	0.74	
	.56068+9.25755*Y3+19.936073*Y5	0.72	
X2 Rigidity	7.708413+10.27329*Y2-0.154016*Y7	0.54	**Y2 -y7 y1**
	4.648178+8.882124*Y2	0.46	
	4.149816+4.596205*Y1+9.937242*Y2	0.49	
X3 Akinesy	3.978743+21.148461*Y5+11.809496*Y8	0.67	**Y5 Y8 -y1 y7**
	6.551802-15.792603*Y1+38.189547*Y5	0.66	
	20.912704*Y5+0.164459*Y7+14.811012*Y8	0.66	
X4 Bradikinesy	1.556933+2.338241*Y3+2.068313*Y8	0.56	**Y8 y2 y3**
	1.679161+1.259394*Y2+1.990788*Y8	0.52	
	1.835763+1.937433*Y8	0.44	

Table 10.5

Results of $x_i=f(y_j,\ldots,y_k)$ modeling by Robust PNN. The 3 best models were taken into consideration. The dependent activity patterns variables are given in the last column in lower case either if this variable's coefficients in the model are much smaller than other variables coefficients or if the variable appears only once in the 3 selected models. Otherwise the dependent symptom variables are given in upper case.

The dependences were considered as "strong" if they appear more than once in the 3 selected models with significant coefficients, which corresponds to the upper case notation in tables 10.4, 10.5. Otherwise, the dependences were considered as "weak". They correspond to the lower case notation in tables 10.4, 10.5. Figure 10.1 summarizes the significant dependences. The dependences found during the first part analysis were taken into account only if they were confirmed by a reciprocal dependency found in the second part of the

analysis. ”Weak” direct and reciprocal dependences are displayed as dotted arrows in Figure 10.1. They are shown as double arrows on the schema. If the both dependences are “strong” they are represented by a solid arrow.

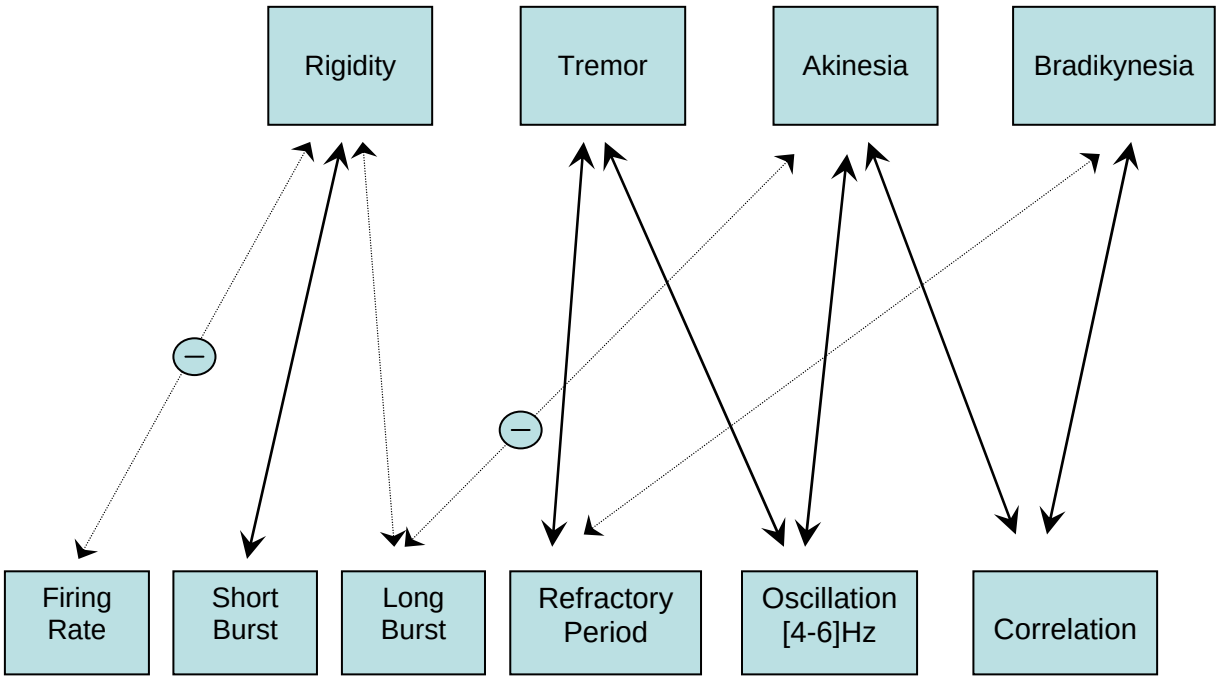


Figure 10.1
Schema of “symptom – activity pattern” dependences. Symptoms and patterns are connected with arrows if dependencies were found during modeling in both directions, i.e. Symptom_A=*f*(pattern_B) and Pattern_B=*f*(Symptom_A). The arrows are solid if the both dependences are “strong” otherwise the arrows are dotted. Negative dependences are marked with sighns “-” in round frams.

10.4Discussion

Increased firing rate in STN and, consequently, in the STN target nuclei, i.e. GPi/SNr, influences the activity of their ultimate target structures i.e. the thalamus, neocortex and brainstem. These changes may impair the normal functioning of the motor system and may underlie the Parkinsonian symptoms such as tremor, rigidity, akinesia, gait, and postural disturbances (Hamani, 2003).

Using PNN modeling several dependences were found between the severity of PD symptoms and the neuronal activity in STN. It is worth noting a strong dependency between rest tremor and the oscillations at [4,6]Hz. This dependency is well known and illustrates the

validity of our approach. Conversely, the association of tremor and refractory period is new and less obvious. Oscillations at [4,6]Hz contribute not only to tremor but also to akinesia. Bursting activity, in particular “short burst” was associated with rigidity. Correlated activity contributes to both akinesia and bradikinesia.

The overall picture drawn by these associations offers a new and challenging scheme of the physiopathology of PD symptoms with respect to the STN activity. Further analysis aimed at studying the association of the same symptoms with the activity of the other main nuclei of the basal ganglia, i.e. GPi/SNr and Thalamus (VIM) should eventually lead to a thorough interpretation of the complex symptomology observed in PD patients. Another perspective is the interpretation of the results according with the existing models of basal ganglia (Gurney et al, 1998). From the other side the results could be used for precising and developing such models.

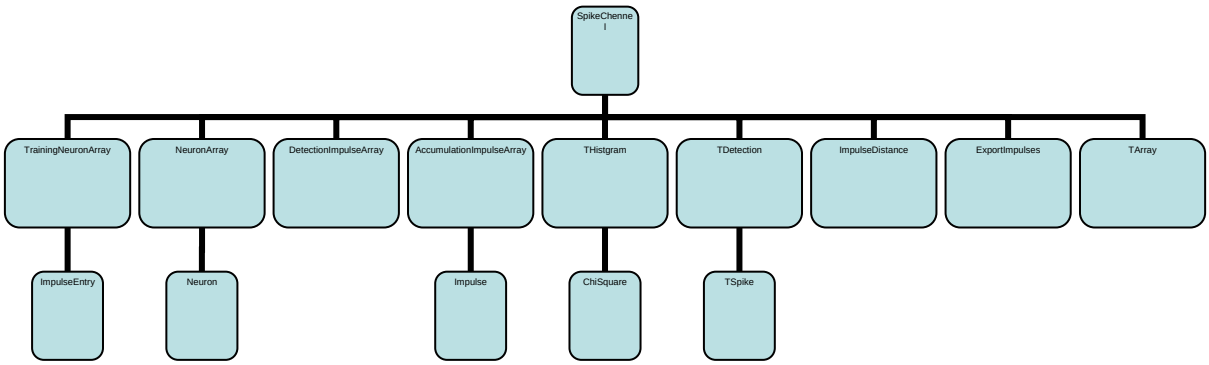
Appendix A

USS software implementation

Dynamic link library main classes

1. SpikeChannel – This class includes all the procedures necessary to process a signal from one microelectrode (one channel). USS processes the channels consequently, so only one object of this class is created. If several channels are supposed to be processed simultaneously one object of SpikeChannel must be created for each of them.
2. TArray – Template class to facilitate manipulations with arrays
3. ImpulseDistance – Includes the procedures to calculate distances between impulses.
4. AccumulationImpulseArray – Includes a dynamic array of impulses of type Impulse (see class 14 "Impulse") which to be filled during accumulation of the learning set for the learning procedure. There are also functions to manipulate this array.
5. DetectionImpulseArray – During the classification of impulses of type Impulse (see class 14 "Impulse") all candidate impulses extracted from the current buffer are added to the dynamic array of this class. Includes also functions to manipulate this array.
6. TrainingNeuronArray – The class is to manipulate the results of the learning procedure and to do the post-processing of the results. Includes an array of all found templates of type Neuron (see class 13 "Neuron").
7. NeuronArray – A class for manipulation with a dynamic array of type Neuron (see class 13 "Neuron").
8. THistogram – A class for histograms calculation.
9. ExportImpulses – Accumulates the classification results as two arrays: neuron numbers and occurrence times during the classification and allows copying them to a memory buffer. (This functionality is not used in the present version of USS).
10. TDetection – Includes functions necessary to process the classification.
11. ChiSquare – A class for χ^2 distribution estimation.
12. TSpike – Keeps detected impulses and their derivatives.
13. Neuron – Includes the information about impulse classes such as class template and class radius.
14. Impulse - Keeps detected impulses and the information about it. The functions of this class allow normalizing and denormalizing an impulse.
15. ImpulseEntry – a class to process entries (and non entries) of impulses in detected classes during learning procedure and its post-processing.
16. TNeuronCenter – structure to define the center of a class.

Classes Diagram



List of external library functions

EXPORT void all_Init(int channelCount, int bufferSize, int* error)

Description:
Creates channelCount objects of class SpikeChannel and initializes their parameters. Sets the global variable corresponding to the number of channels to analyze simultaneously to channelCount.

Parameters:
const int channel –number of channels to anywise simultaneously
int bufferSize
int * error – set to 1 if error, set to 0 if successful.

EXPORT void all_Done(int* error)

Description:
Deletes all the objects of class SpikeChannel created before. Sets the global variable corresponding to the number of channels to analyze simultaneously to 0.

Parameters:
int * error – set to 1 if error, set to 0 if successful.

EXPORT int all_Ready(int* error);

Description:
Returns 0 if all channels are ready to be analyzed; 0 – otherwise.
Parameters:
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SpikeMain(const int channel, short * Buffer, int bufSize, int * error)

Description:
Processes next buffer of data signal according to the current regime of the program:
Visualization regime - signal derivative estimation;
Accumulation regime - signal derivative estimation, detection of spikes and accumulation of the learning set;
Spike sorting regime - signal derivative estimation, detection of spikes and their classification.

Parameters:
const int channel – channel number
short * Buffer – pointer to a buffer of data
int bufSize – size of the buffer
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_StartAccumulation(const int channel, const int accumulate Noise,
int* accumulateMask, int* keepMask, int size, int * error)

Description:
Sets the regime of accumulation for the specified channel set accumulation and creates an object of class AccumulationNeuronArray and initializes it.

Parameters:
const int channel – channel number
const int accumulate Noise -
int* accumulateMask -
int* keepMask -
int size – learning set size (number of spikes to accumulate)

int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_StartDetection(const int channel, int * error)

Description:
Sets the classification regime for the specified channel.
Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_StopDetection(const int channel, int * error)

Description:
Resets the current regime to the visualization for the specified channel.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void all_StartAccumulation(int * error)

Description:
Sets the regime of accumulation for all the channels and creates an object of class AccumulationNeuronArray for each of them.

Parameters:
int * error – set to 1 if error, set to 0 if successful.

EXPORT void all_StartTraining(int* error)

Description:
Stops the on-line flow of the program. Stars the learning procedure for all the channels for which a learning set is accumulated. Creates an object of class TrainingNeuronArray with the results of the learning procedure for each of them. Resets the on-line flow.

Parameters:
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingIteration(const int channel, int * error)

Description:
Starts the next iteration of the learning procedure for the specified channel.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingMarkSmall(const int channel, int * array, const int size, int * error)

Description:
Learning results post-processing function. Finds the non-representative classes and puts information about in the array "array". (Class is considered to be non-representative if it has less then 3 members.)

Parameters:
const int channel – channel number
int * array – array[i] is 1 if the class i is considered to be non-representative; 0 – otherwise.
const int size – size of the array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingRemoveSmall(const int channel, int * error)

Description:
Learning results post-processing function. Finds and removes the non-representative classes. (Class is considered to be non-representative if it has less then 3 members.)

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingRemoveArtifacts(const int channel, int * error)

Description:
Learning results post-processing function. Finds and removes the classes, formed by artifacts of large amplitude. (Class is considered to be formed by artifacts if it has less then 4 members.)

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingMarkOverlapped(const int channel, int * array, const int size, int * error)

Description:
Learning results post-processing function. Finds the overlapped classes and puts information about in the array "array". (Classes are considered overlapped if one of them includes more then 80% of members of the another one.)

Parameters:
const int channel – channel number
int * array – array[i] is 1 if the class i is the less representative of two overlapped classes; 0 – otherwise.
const int size – size of the array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingViewOverlapped
(const int channel, const int neuron, int * array, const int size, int * error)

Description:
Learning results post-processing function. Finds all the classes overlapped with class "neuron" and puts information about in the array "array". (Classes are considered overlapped if one of them includes more then 80% of members of another one.)

Parameters:
const int channel – channel number
const int neuron – class number
int * array – array[i] is 2 if the class i is overlapped with classes "nuron"; 0 – otherwise
const int size – size of the array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingRemoveOverlapped(const int channel, int * error)

Description:
Learning results post-processing function. Finds and removes the overlapped classes and puts information about in the array "array". (Classes are considered overlapped if one of them includes more then 80% of members of the another one.)

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_DelTrainingNeuron(const int channel, int neuron, int * error)

Description:
Learning results post-processing function. Deletes class "neuron" from the resalts of the learning.
Parameters:
const int channel – channel number
int neuron – class number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_Accept(const int channel, int * error)

Description:
Creates an object of class "DetectionImpulseArray" for the channel "channel", and prepares everything for classification according to the accepted templates.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void all_Accept(int * error)

Description:
Finds all the channel for which the learning is finished and creates an object of class "DetectionImpulseArray" for each of them. Prepares everything for classification according to the accepted templates.

Parameters:
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_memcpyD0(const int channel, short * DestBuf, int Size, int * error)

Description:
Copies the currently processed buffer of row signal.

Parameters:
const int channel – channel number
short * DestBuf – pointer to the destination array
int Size – size of the destination array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_memcpyD1(const int channel, int * DestBuf, int Size, int * error)

Description:
Copies the derivative of row signal estimated for the currently processed buffer.

Parameters:
const int channel – channel number
short * DestBuf – pointer to the destination array
int Size – size of the destination array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_memcpyD2(const int channel, int * DestBuf, int Size, int * error)

Description:
Copies the second derivative of row signal estimated for the currently processed buffer.

Parameters:

const int channel – channel number
short * DestBuf – pointer to the destination array
int Size – size of the destination array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetAlpha(const int channel, int * alpha, int * error)

Description:
Gets the current regularization parameter for the first derivative estimation.

Parameters:
const int channel – channel number
int * alpha – pointer to the regularization parameter value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetAlpha(const int channel, int * alpha, int * error)

Description:
Sets the current regularization parameter for the firs derivative estimation.

Parameters:
const int channel – channel number
int * alpha – pointer to the regularization parameter value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetBeta(const int channel, int * beta, int * error)

Description:
Gets the current regularization parameter for the seconf derivative estimation.

Parameters:
const int channel – channel number
int * beta - pointer to the regularization parameter value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetBeta(const int channel, int * beta, int * error)

Description:
Sets the current regularization parameter for the second derivative estimation.

Parameters:
const int channel – channel number
int * beta - pointer to the regularization parameter value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetImpCount(const int channel, int * cnt, int * error)

Description:
Gets the count of impulses in the learning set.

Parameters:
const int channel – channel number
int * cnt – pointer tho the value of impulses count
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetImpCount(const int channel, int * cnt, int * error)

Description:
Sets the count of impulses in the learning set.

Parameters:
const int channel – channel number
int * cnt – pointer tho the value of impulses count
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTbefore(const int channel, int * Tbefore, int * error)

Description:
Gets the current parameter of number of point to extract before the threshold crossing during detection of spikes in row signal.

Parameters:
const int channel – channel number
int * Tbefore – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetTbefore(const int channel, int * Tbefore, int * error)

Description:
Sets the current parameter of number of point to extract before the threshold crossing during detection of spikes in row signal.

Parameters:
const int channel – channel number
int * Tbefore – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTafter(const int channel, int * Tafter, int * error)

Description:

Gets the current parameter of number of point to extract after the threshold crossing during detection of spikes in row signal.

Parameters:
const int channel – channel number
int * Tafter – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetTafter(const int channel, int * Tafter, int * error)

Description:
Sets the current parameter of number of point to extract after the threshold crossing during detection of spikes in row signal.

Parameters:
const int channel – channel number
int * Tafter – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetImpulseSize(const int channel, int * ImpulseSize, int * error)

Description:
Gets the size of the arrays in which the spike and its derivatives are kept.

Parameters:
const int channel – channel number
int * ImpulseSize – pointer to the value of arrays size
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetAutoThreshold(const int channel, int * autoThreshold, int * error)

Description:
Gets the Boolean value of the automatic threshold parameter. If the value is true, the threshold for detection of spikes from the signal is recalculated automatically for each buffer of signal. Otherwise the threshold is set to a fixed value.

Parameters:
const int channel – channel number
int * autoThreshold – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetAutoThreshold(const int channel, int * autoThreshold, int * error)
Description:
Gets the Boolean value of the automatic threshold parameter. If the value is true, the threshold for detection of spikes from the signal is recalculated automatically for each buffer of signal. Otherwise the threshold is set to a fixed value.

Parameters:
const int channel – channel number
int * autoThreshold – pointer to the value of the parameter
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetThreshold(const int channel, int * threshold, int * error)

Description:
Gets the value of the current threshold for detection of spikes from the.

Parameters:
const int channel – channel number
int * threshold – pointer to the value of the threshold
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetThreshold(const int channel, int * threshold, int * error)

Description:
Sets the value of the current threshold for detection of spikes from the signal.

Parameters:
const int channel – channel number
int * threshold – pointer to the value of the threshold
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetThreshCoeff(const int channel, float * threshCoeff, int * error)

Description:
Gets the value of the parameter of coefficient to calculate the threshold for detection of spikes from the signal. To calculate the threshold value the standard diviation of the signal is multiplied by the coefficient.

Parameters:
const int channel – channel number
int * threshold – pointer to the value of the coeffitient
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetThreshCoeff(const int channel, float * threshCoeff, int * error)

Description:
Sets the value of the parameter of coefficient to calculate the threshold for detection of spikes from the

signal. To calculate the threshold value the standard diviation of the signal is multiplied by the coefficient.

Parameters:
const int channel – channel number
int * threshold – pointer to the value of the coeffitient
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionRadiusCoeff(const int channel, float * Coeff, int * error)

Description:
Gets the value of the parameter of coefficient to fit the detected classes' radiuses for the classification. After the learning the classes' radiuses are calculated automatically and then they are multiplied by the coefficient.

Parameters:
const int channel – channel number
int * Coeff – pointer to the value of the coeffitient
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetDetectionRadiusCoeff(const int channel, float * Coeff, int * error)

Description:
Sets the value of the parameter of coefficient to fit the detected classes' radiuses for the classification. After the learning the classes' radiuses are calculated automatically and then they are multiplied by the coefficient.

Parameters:
const int channel – channel number
int * Coeff – pointer to the value of the coeffitient
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetSigma(const int channel, float * sigma, int * error)

Description:
Gets the value of the standard deviation of the signal.

Parameters:
const int channel – channel number
int * sigma – pointer to the value of the standard deviation
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetNeuronCount(const int channel, int * count, int * error)

Description:
Gets the count of classes, detected during the learning stage.

Parameters:
const int channel – channel number
int * count – pointer to the value of the count
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetNeuron(const int channel, const int neuron, int * Center, float * Radius, int * ImpulseCount, int* changeable, int * error)

Description:
Gets information (see parameters) about class "neuron".

Parameters:
const int channel – channel number
const int neuron – class number
int * Center – number of the impulse, representing the class center
float * Radius – radius of the class
int * ImpulseCount – number of impulses-members of the class
int* changeable – 1- if the changes of class parameters are allowed, 0- otherwise
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetIsIn(const int channel, const int neuron, char* IsIn, int * error)

Description:
Gets the array containing information about belonging of the impulses from the learning set to class "neuron".

Parameters:
const int channel – channel number
const int neuron – class number
char* IsIn – IsIn[i] is 1 if impulse i belongs to the class, 0 - otherwise
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetMinMaxNeuron(const int channel, short* min, short* max, int* error)

Description:
Gets numbers of classes of maximal and minimal amplitudes.

Parameters:
const int channel – channel number
short* min – pointer to the minimal amplitude class number

short* max - pointer to the maximal amplitude class number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetImpulse
(const int channel, const int impulse, short * data0, int * data1, int * data2, int * error)

Description:
Gets impulse "impulse" from the learning set.

Parameters:
const int channel – channel number
const int impulse – impulse number
short * data0 – pointer to the array of signal values, corresponding to the impulse
int * data1 - pointer to the array of signal first derivative values
int * data2 - pointer to the array 0f signal second derivative values
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetImpulseMinMax
(const int channel, const int impulse, short * min, short * max, int * error)

Description:
Gets maximal and minimal amplitude values of impulse "impulse" from the learning set.

Parameters:
const int channel – channel number
const int impulse – impulse number
short * min – pointer to the minimal amplitude value
short * max – pointer to the maximal amplitude value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetImpulseTime (const int channel, const int impulse, int * time, int * error)

Description:
Gets the time of the impulse "impulse" occurrence.

Parameters:
const int channel – channel number
const int impulse – impulse number
int * time – pointer to the occurrence time
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetMeanSpike(const int channel, const int neuron, short* data0, int* error)

Description:
Calls SpikeChannel class methods to calculate mean impulse fore class "neuron".

Parameters:
const int channel – channel number
const int neuron – class number
short* data0 – pointer to the array, containing the mean impulse
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_Distr1toAll
(const int channel, const int impulse, int * Distr, int DistrSize, float Step, int * error)

Description:
Calls SpikeChannel class methods to calculate distribution histogram of distances from impulse "impulse" to all other impulses of the learning set.

Parameters:
const int channel – channel number
const int impulse – impulse number in the learnn set
int * Distr – pointer to the array, containing the calculated distribution
int DistrSize – size of the array
float Step – histogram bin
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTrainingRadius(const int channel, int neuron, float * radius, int * error)

Description:
Gets class radius.

Parameters:
const int channel – channel number
int neuron – class number
float * radius – pointer to the radius value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTrainingDistances(const int channel, float * dist, int * error)

Description:
Gets the array of distances between all the spikes from learning set.

Parameters:
const int channel – channel number
float * dist – pointer to the distances array
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetTrainingRadius(const int channel, int neuron, float * radius, int * error)

Description:
Sets the class radius for learning.

Parameters:
const int channel – channel number
int neuron – class number
float * radius – pointer to the radius value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_TrainingCalculateRadius(const int channel, int* error)

Description:
Initiates the (re)calculation of detected classes radiuses.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTemplateCount(const int channel, int * count, int * error)

Description:
Gets the count of detected classes.

Parameters:
const int channel – channel number
int * count – pointer to detected classes count
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetTemplate(const int channel, const int impulse,
short* data0, int* data1, int* data2, int* ext1, int* ext2, int* error)

Description:
Gets a detected template. Each template is represented by a spike detected from the raw signal. For each spike the first and second derivative calculated and may be normalized.

Parameters:
const int channel – channel number
const int impulse – number of the template to get
short* data0 – pointer to the array of the spike from raw signal
int* data1 - pointer to the array containing the first derivative of the spike
int* data2 - pointer to the array containing the second derivative of the spike
int* ext1 – pointer to the value of the normalization coefficient for the first derivative
int* ext2 - pointer to the value of the normalization coefficient for the second derivative
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetMinMaxTemplate(const int channel, short* min, short* max, int* error)

Description:
Gets values of minimal and maximal amplitudes on all the detected templates.

Parameters:
const int channel – channel number
short* min – pointer to the value of minimal amplitude
short* max – pointer to the value of maximal amplitude
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetExpoImpsCount(const int channel, int * count, int * error)

Description:
Gets count of exported templates.

Parameters:
const int channel – channel number
int * count – pointer to the value of template count
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetHistogramSize(const int channel, int* size, int* error)

Description:
Gets the size of a histogram.

Parameters:
const int channel – channel number
int * size – pointer to the value of the histogram size
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_ResetHistogram(const int channel, int* error)

Description:
Resets the members of class THistogram to initial values.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetHistogram
(const int channel, const int neuron, int* good, int* bad, int* chi, int size,
float* e1, float* e2, int* error)

Description:
Gets histogram information.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionRadius(const int channel, const int neuron, float* radius, int* error)

Description:
Gets the class radius used for spike sorting.

Parameters:
const int channel – channel number
int neuron – class number
float * radius – pointer to the radius value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_SetDetectionRadius(const int channel, const int neuron, float* radius, int* error)

Description:
The class radius is calculated automatically by learning procedure are used for spike sorting. This function allows setting the class radius.

Parameters:
const int channel – channel number
int neuron – class number
float * radius – pointer to the radius value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionSpike(const int channel, int spikeNumber, int* nrn, double* spikeX, double* spikeY,
double* portretX, double* portretY, int* match, int* time, int* error)

Description:
Gets a spike from the member array of impulses of class DetectionImpulseArray.

Parameters:
const int channel – channel number
int spikeNumber – number of the spike in the array
int* nrn – pointer to the class number value
double* spikeX – pointer to the array of X coordinates of the spike graph in time domain
double* spikeY – pointer to the array of Y coordinates of the spike graph in time domain
double* portretX – pointer to the array of X coordinates of the spike graph in phase space
double* portretY – pointer to the array of Y coordinates of the spike graph in phase space
int* match
int* time – pointer to the value of the spike occurrence time
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionAmplitude(const int channel, double* ampl, int* error)

Description:
Gets maximal amplitude value for graphical output of sorted spikes.

Parameters:
const int channel – channel number
double* ampl – pointer to the amplitude value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionFiringRate(const int channel, int neuron, double* firingRate, int* error)

Description:
Gets firing rate of a sorted spike.

Parameters:
const int channel – channel number
int neuron – class number
double* firingRate – pointer to the firing rate value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_GetDetectionSpikeCount(const int channel, int* spikeCount, int* error)

Description:
Gets number of spikes detected from the current buffer of signal.

Parameters:
const int channel – channel number
double* spikeCount – pointer to the count value
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_AddTemplates(const int channel, int templLength, float Radius, short* spike, int* der1, int* der2, int ext1, int ext2, int* error)

Description:
Adds a class template to the member array of type Neuron in NeuronArray class.

Parameters:
const int channel – channel number
int templLength – size of the arrays Spike, der1, der2
float Radius – radius of the class, represented by the template
short* spike – pointer to the array of the spike from raw signal
int* der1 - pointer to the array containing the first derivative of the spike

int* der2 - pointer to the array containing the second derivative of the spike
int* ext1 – pointer to the value of the normalization coefficient for the first derivative
int* ext2 - pointer to the value of the normalization coefficient for the second derivative

int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_ArrangeEverythingForDetection(const int channel, int* error)

Description:
Creates and initializes all necessary objects for spike sorting. This function should be used if the templates are exported and the learning stage is skipped.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_UpdateTemplates(const int channel, int* error)

Description:
Should update templates according to the changes in the activity. This function is not used in USS.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void all_updateChi(int* error)

Description:
Should update histograms after the update of the templates. This function is not used in USS.

Parameters:
const int channel – channel number
int * error – set to 1 if error, set to 0 if successful.

EXPORT void chl_AccumulationDone(const int channel, int* done, int* error)

Description:
Verifies if the accumulation of the learning set is over.
Parameters:
const int channel – channel number
int* done - set to 0 if the accumulation of the learning set is finished, non 0 value if not
int * error – set to 1 if error, set to 0 if successful.

EXPORT int all_Ready(int* error)

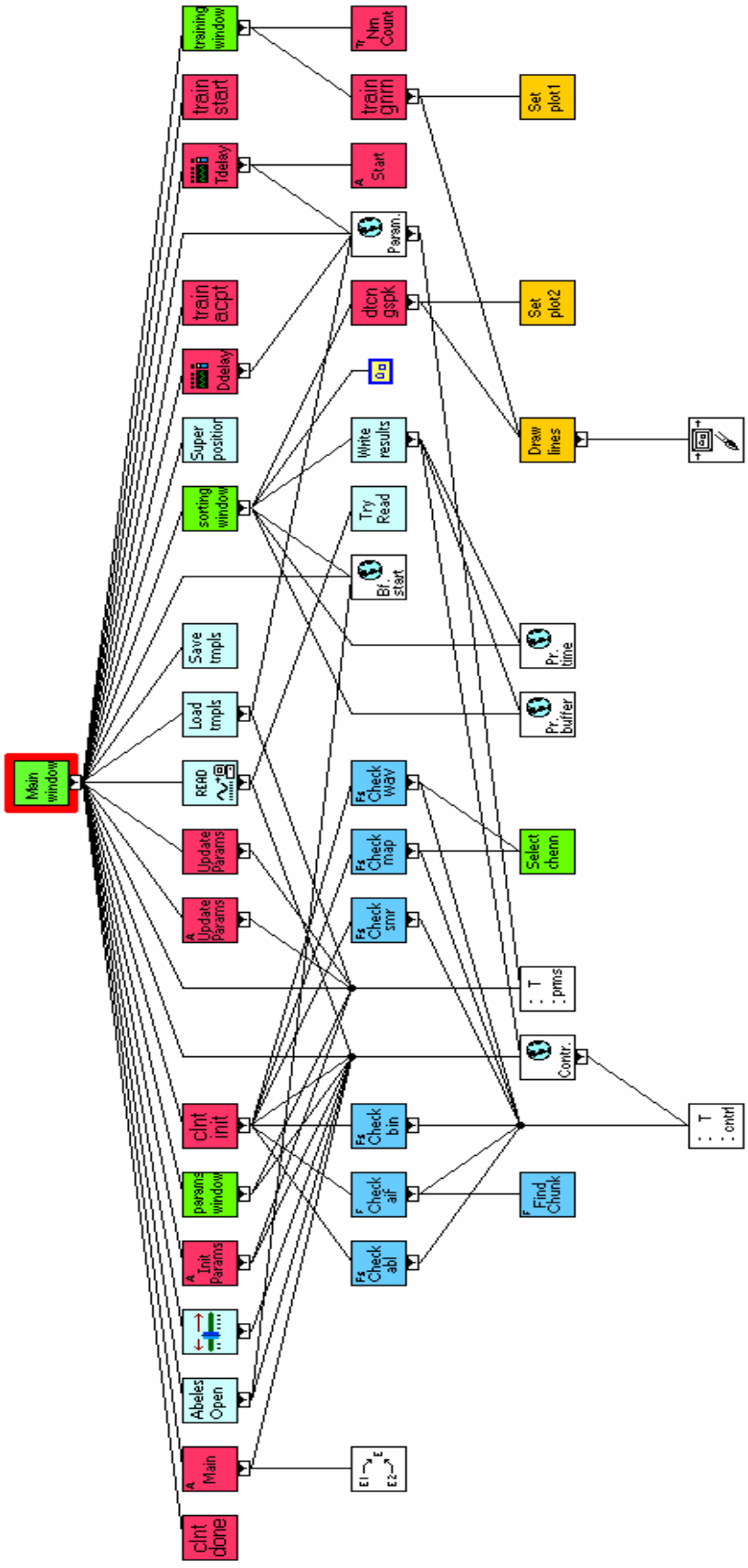
Description:
Returns number of channels ready for on-line processing, i.e. they are not on the off-line stage of learning.
Parameters:

Appendix A:

USS LabView modules documentation

USS data reading and manipulation, as well as user interface is realized on National Instruments LabView. The realization contains about 50 operating modules which called Visual Instruments. They permit interaction with user by means of windows menu and LabView controls, data file reading, data and analysis visualization, operating which includes switching between application modes (simple visualization, learning set collection, spike sorting, running of the learning procedure) and library function calls. All these modules can be grouped into User Interface Modules, File Manipulation Modules, Data Format Checking Modules, Operating Modules. The schema below represents hierarchical view of modules, user defined data types and global variables which is followed by descriptions of each of them.USS can be lunched directly from LabView environment by starting the main window module SpikeDetection.vi. The spike sorting window module should have been opened previously to avoid blinking during spike sorting,

Appendix A:

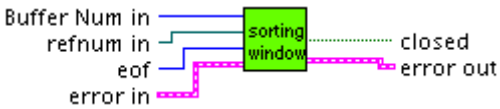

training window
train start
Tdelay
train acpt
Ddelay
Super position
sorting window
Save tmpls
Load tmpls
READ
Update Params
A Update Params
clnt init
params window
A Init Params
Abeles Open
A Main
clnt done
Main window
Nm Count
train gnrn
A Start
Param.
dtcn gspk
Go
Write results
Try Read
Bf. start
Pr. time
Pr. buffer
F5 Check wav
F5 Check map
F5 Check smr
F5 Check bin
F Check air
F5 Check abl
Set plot1
Set plot2
Draw lines
Select chenn
: T : pms
Contr.
: T : cntrl
F Find Chunk

User Interface Modules

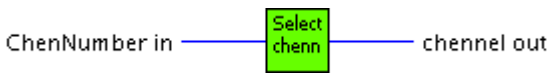
File name: SpikeDetection.vi
Functions:
- menu handling
- signal visualisation
- parameters fitting
- switcheing between Visualisation, Learning and Spike Sorting modes

---

File name: Training.vi
Functions:
- visualissation of the results of the learning proocedure;
- user template selection feed back;
Input
- error handling structure;
Output
- error handling structure;



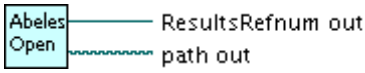
File name: Detection.vi
Functions:
- spike sorting visualisation;
Input:
- number of already readed data buffers fron the current signal;
- current signal file reference;
- eof (0-no, 1-yes)
- error handling structure;
Output:
- closed (boolean)
- error handling structure;



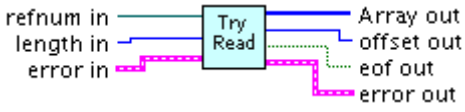
File name: SelectChannel.vi
Functions:
- channel selection for multy chanel recordings
Input
- number of channels in the recording;
Output
- selected channel number;

File name: DialogUpdateParams.vi
Functions:
- parameters fitting;
Input
- global parameters structure;
Output
- global parameters structure;

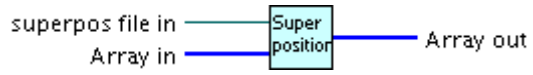
File Handling Modules



File name: AbelesOpen.vi
Functions:
- opens and prepares output file;
- writes heading information;
Outputs:
- LabView file reference;
- file path;

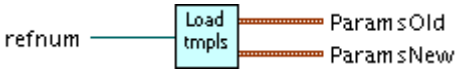


File name: TryRead.vi
Functions:
- reads next portion of data of specified length from the file to a data buffer;
- checks eof;
Inputs:
- LabVieew file reference;
- length of data to read;
- error handling structure;
Outputs;
- a buffer containing read data;
- eof (bool)
- new offset;
- error handling structure;

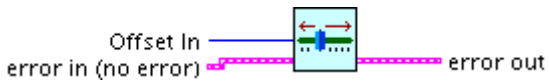


File name: Superposition.vi
Functions:
- superimposes by addition a specified file data and a data buffer;
Inputs:
- data buffer;
- LabView file reference;
Outputs;
- resulting buffer;

File name: SaveTemplates.vi
Functions:
- creats a new template file;
- saves templates and templates info resulted from the last training procedure;
Inputs:
- folder path to save the file;



File name: LoadTemplates.vi
Functions:
- loads templates and template info from a template file;
- modifies spike sorting parameters according to the loaded info.
Inputs:
- LabView template file reference;
Outputs;
- parameters structure containing old parameters;
- parameters structure containing newly loaded parameters;

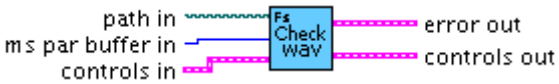


File name: File Set Offset.vi
Functions:
- allows moving along the current record by means of main window slide control;
Inputs:
- current offset;
- error handling structure;
Outputs;
- new offset;
- error handling structure;

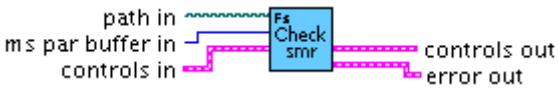
File name: ResultsWrite.vi
Functions:
- writes detected spike record to the output time series file during spike sorting;
Inputs:
- current data buffer number;
- LabView file reference;
- current channel number;
- template neuron number;
- time in sample points in current data buffer;

File name: Data Read.vi
Functions:
- Manages moving along the file and reading of data corresponding to the selected channel;
Inputs:
- error handling structure;
Outputs;
- eof (bool);
- filled buffer;
- new offset (current position)
- new offset;
- error handling structure;

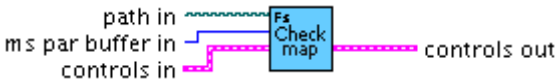
Format Checking Modules



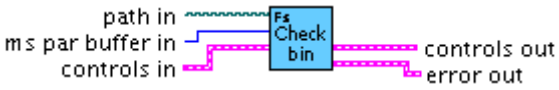
File name: CheckFormatWAV.vi
Functions:
- fills the data read control structure from headings of wav file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;



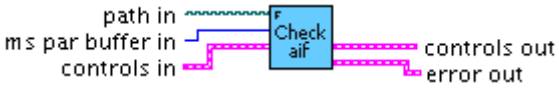
File name: CheckFormatSMR.vi
Functions:
- fills the data read control structure from headings of smr file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;



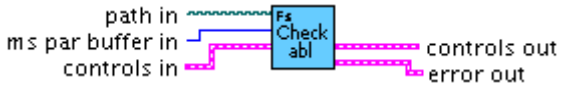
File name: CheckFormatMAP.vi
Functions:
- fills the data read control structure from headings of map file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;



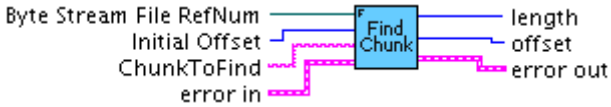
File name: CheckFormatBin.vi
Functions:
- fills the data read control structure from headings of binary file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;



File name: CheckFormatAIF.vi
Functions:
- fills the data read control structure from headings of aif file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;

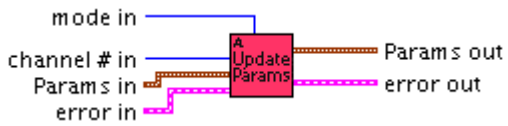


File name: CheckFormatAbl.vi
Functions:
- fills the data read control structure from headings of abeles file;
Inputs:
- file path;
- data buffer length in ms;
- file read controls structure;
Outputs;
- file read controls structure;
- error handling structure;

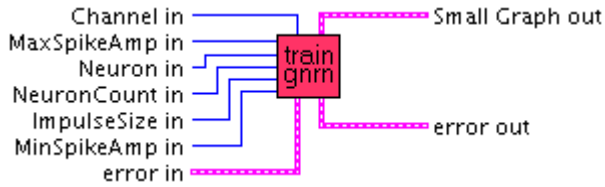


File name: LoadTemplates.vi
Functions:
- loads templates and template info from a template file;
- modifies spike sorting parameters according to the loaded info.
Inputs:
- LabView template file reference;
Outputs;
- parameters structure containing old parameters;
- parameters structure containing newly loaded parameters;
FindChunkAIF.vi
Functions:
- finds a specified chunk of aiff file;
Inputs:
- aiff file reference;
- current offset;
- chunk specification;
- error handling structure;
Outputs;
- read data length;
- new offset;
- error handling structure;

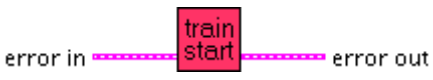
Operating Modules



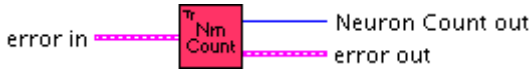
File name: UpdateAllParameters.vi
Functions:
- updates all the algorithm parameters;
Input:
- mode (get/set);
- current channel number;
- parameters structure;
- error handling structure;
Outputs;
- parameters structure;
- error handling structure;



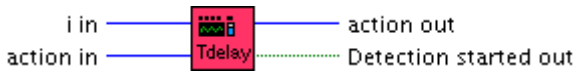
File name: TrainingGetNuron.vi
Functions:
- Get a specified template after leaning and draws all graphs to show on the training window;
Inputs:
- current channel number;
- maximal amplitude in learning set;
- minimal amplitude in learning set;
- number of templates;
- number of template to show;
- spike duration in sample points;
- error handling structure;
Outputs:
- graphs structure; // contains time domain graph, space phase graph, and the histogram of distances from the template to all other learning set spikes.
- error handling structure;



File name: Training Start.vi
Functions:
- starts learning procedure;
Inputs:
- error handling structure;
Outputs;
- error handling structure;

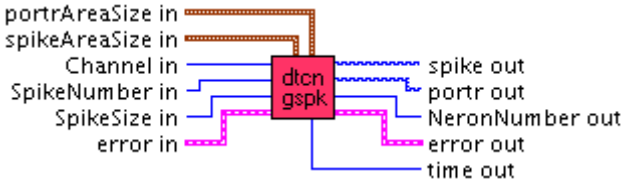


File name: Training Get Neuron Count.vi
Functions:
- gets number of templates;
Inputs:
- error handling structure;
Outputs;
- number of templats;
- error handling structure;



File name: Training Delay.vi
Functions:
- starts learning at the specified time (if time is not specified starts spike sorting immediately);
Input:
- current buffer number;
- current execution mode; //0-visualisation only; 1-learning; 2-spike sorting;
Outputs;
- new execution mode;
- learning started (bool);

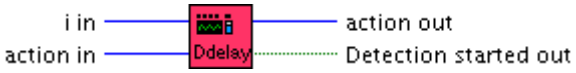
File name: Training Accept.vi
Functions:
- terminates learning procedure;
- initialises templates;
Inputs:
- error handling structure;
Outputs;
- error handling structure;



File name: Detection Get Spike.vi
Functions:
- gets next detected spike and sends it to the appropriate plotter on the spike sorting;
Inputs:

- current channel number;
- time domain plotter parameters structure;
- phase space plotter parameters structure;
- spike number;
- spike duration in sample points;
- error handling structure;

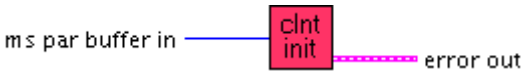
Outputs:
- data to plot in time domain plotter;
- data to plot in phase space plotter;
- template number; //0 - if the spike were not classified
- spike occurrence time in ms;
- error handling structure;



File name: Detection Delay.vi
Functions:
- starts spike sorting at the specified time (if time is not specified starts spike sorting immidiately);

Input:
- current buffer number;
- current execution mode; //0-visualisation only; 1-learning; 2-spike sorting;

Outputs;
- new execution mode;
- spike sorting started (bool);



File name: Client Init.vi
Functions:
- initialises session;

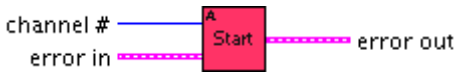
Inputs:
- error handling structure;

Outputs;
- error handling structure;

Client Done.vi
Functions:
- terminates session;

Inputs:
- error handling structure;

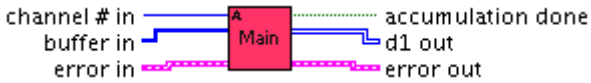
Outputs;
- error handling structure;



File name: A Start.vi
Functions:
- starts learning set collection;

Inputs:
- current channel number;
- error handling structure;

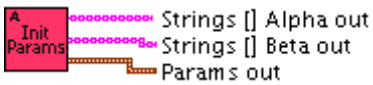
Outputs:
- error handling structure;



File name: A Main.vi
Functions:
- controls data flow;
- controls switches between execution modes;
- calls the derivative calculations;

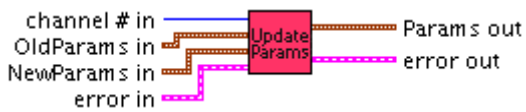
Inputs:
- current channel number;
- current data buffer
- error handling structure;

Outputs:
- accumulation done (bool);
- buffer containing derivative of current data buffer;
- error handling structure;



File name: A Init Parameters.vi
Functions:
- initializes algorithm parameters;

Outputs;
- string array for the first derivative smoothing parameter choice;
- string array for the second derivative smoothing parameter choice;
- parameters structure;



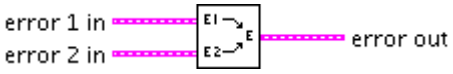
File name: UpdateChangedParameters.vi
Functions:
- updates all the algorithm parameters;

Input:

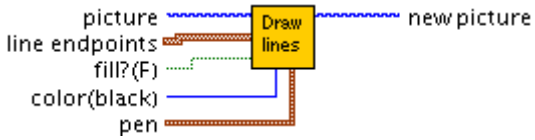
- current channel number;
- new parameters structure;
- old parameters structure;
- error handling structure;

Outputs;
- parameters structure;
- error handling structure;

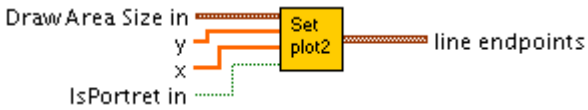
Additional Modules



File name: Error Merge.vi
Functions;
- takes two possible errors and passes through either the "error in 1" if the error boolean is true or "error 2 in".



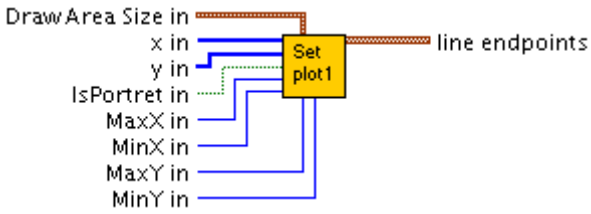
Draw Multiple Lines.vi
Draws multiple connected lines into a picture.



File name: Training Plot Conv.vi
Functions;
- constructs scaled data structures ready to draw in plotters for spikes and their phase portrets.

Input:
- plot area parameters structure;
- data array X;
- data array Y;
- portret (bool)

Output:
- data structure for input to the LabView drawing moduls.



File name: TrainingDraw.vi
Functions;
- constructs scaled data structures ready to draw in plotters for spikes and their phase portrets.

Input:
- plot area parameters structure;
- data array X;
- data array Y;
- portret (bool);
- maximum X on the learning set;
- minimum X on the learning set;
- maximum Y on the learning set;
- minimum X on the learning set;

Output:
- data structure for input to the LabView drawing moduls.

User Defined Types

File name: Controls.ctl
File reading controls structure
-Occurence
-ScanRate //Sampling rate
-BufferSize //Size of the buffer to read data from file
-FileRefNum //LabView file rreference
-DataOffset //Offset from begining of the file to read data
-DataPeriod //
-DataLength //Total amount of data in bytes
-BlockLength //Length of block for data formats organized by blocks
-BlockMarker //Marker of blocks to read for data formats organized by blocks
-BlockFirstRead //Offset fron the beginning of block
-channel //Channel number
-swap (bool) //Swap bytes while reading
- byBlocks (bool) //Data organized by blocks
-Ext //file extension

File name: Params.ctl
Algorithm parameters structure:
- Alpha //first derivative smoothing parameter
- Beta //second derivative smoothing parameter
- SpikeDuration // Duration of extracted spikes in sample points
- SpikeAmount //Amount of spikes to collect for learning
- Training StartTime //Time in ms from the record beginning to start training
- DetectionStartTime //Time in ms from the record beginning to start training
- DetectionRadiusCoeff //Coefficient to fit spike sorting classes radiuses

File name: Params.vi
Params: params //algorithm parameters

Global Variables

File name: BuffStartDet.vi
int: BuffStartDet //first buffer for spike sorting, variable for current time estimation

---

File name: Controles.vi
Controls: controls //file reading parameters

File name: PrevBuff.vi
int: PrevBuf //Previous buffer number, variable for current time estimation

---

File name: PrevTime.vi
int: PrevTime //Previous spike time, variable for current time estimation

Appendix B

MAP file format

Each Alpha-Map file with extension 'map' contains a set of consistent blocks. There are two types of blocks, Definition blocks and Data blocks. Both block types comprise header and a data part.

The data part of the definition blocks describes the workspace at the time the data is saved. The data part of the data blocks contains digitized signals or information was receiving in the time of acquisition. The file structure uses standard 'C' programming terms. All fields in the blocks are aligned to 2 bytes.

Each 'map' file starts with a number of definition blocks. The first block of the file is a special type "h" - header block. All other block types are specified using numeric symbols from 0 to 7 (ASCII characters); figures 0, 1, 2, 3, 4, 6 indicate definition blocks, while figures 5 and 7 specify data blocks.

The structure of each type of block is shown here below. All block types are written using standard 'C' terms (for 16-byte applications, using 2-byte-alignment). The data parts of the various blocks are shown in frames. If the meaning of a variable is not self evident from the name, it will be clarified in the comments under the frame.

Each block starts with a header. The header of any block consists of two items:

```
- int m_length              // length of the block (in bytes)
- char m_TypeBlock          // lock's identifier
```

For Definition blocks:

```
Type h - Header (m_TypeBlock='h'):
```

Data part:

```
short m_version;           // the program version number
_dostime_t m_time;         // the start time of data saving
_dosdate_t m_date;         // the date of data saving
```

Type 0 - Set boards (`m_TypeBlock='0'`):

Data part:

```
long m_nextBlock;          // the distance (in bytes) from the beginning of the file
                           // to the next block
int m_BoardCount;          // the number of boards
int m_GroupCount;          // the number of channel groups
WINDOWPLACEMENT m_placeMainWindow;
```

Type 1 - Boards (m_TypeBlock='1'):

The Data part of the board definition block contains information concerning three topics: Channels specification for this board (from the "HardWare" dialog box),Default values

(common for all channels of the same mode) and Auto scaling parameters generated in RMS-mode,
where:
`m_TrgMode = 0` - Down, 1 - Up, 2 - Up and Down.
`m_nameBoard =` the model of the board (like DAP3200);

This is the last part of the block. For more details see the corresponding dialog boxes in the Alpha-MAPsoftware.

Note: The "Board" block is usually followed by the number of definition blocks for channels and triggers concerning the corresponding board.

Type 2 - Definition blocks for channels (m_TypeBlock='2').
This block type contains the common data part for all types of channels:
Comment: m_isAnalog = 1 for Analog channels or 0 for Digital channels
m_isInput = 1 for Input channels or 0 for Output channel

This information should also include the following contents. Each type of channel has different contents, as shown below:

- For Continuous channels:

```
float m_Amplitude;          //scale
float m_SampleRate;
int m_ContBlkSize;          //not used up to ver. 1.9
UINT m_ModeSpike            // shape
float m_Duration;
BOOL m_bAutoScale;
char m_Name;                //channel name
```

Comment: m_ModeSpike - uses different values for different drawing modes:
Normal bars, Colored bars (see Initialize dialog box).

- For Level channels:

```
float m_SampleRate;
int m_nSpikeCount;
UINT m_ModeSpike ;
float m_nPreTrigmSec;
float m_nPostTrigmSec;
int m_LevelValue;
BOOL m_TrgMode;             //positive,negative
BOOL m_YesRms;
BOOL m_bAutoScale;
char m_Name;
```

Comment: `m_YesRms = 1` if automatic level calculation, based on RMS, is used `m_bAutoScale = 1` if scale for this channel drawing is also adjusted automatically from RMS calculation (vice versa it is zero (0)).

- For External Trigger channels:

```
float m_Amplitude;
float m_SampleRate;
int m_nSpikeCount;
UINT m_ModeSpike ;
float m_nPreTrigmSec;
float m_nPostTrigmSec;
UINT m_TriggerNumber;
char m_Name;
```

- For Digital channels:

```
float m_SampleRate;
BOOL m_SaveTrigger;
float m_Duration;
char m_Name;
```

Comment: `m_SaveTrigger = 1` if saving trigger is used for this Digital channel (this option is not available in the current Alpha-MAP version, but has been included as a reserve options).

Type 6 - Definition block for a trigger (`m_TypeBlock='6'`):

Comment: Variable m_StateChannels saves the Trigger Mode for all eight (8) channels of this trigger using bit couples (from right to left).Each pair of bits defines for each channel its Trigger Mode: 0 - Down, 1 - Up, 2 -Up and Down.

Note: After completing the definition blocks for all boards (including their channels and triggers) a set of definition blocks for signals displaying windows (groups and subgroups of channels) should follow.

Type 3 - Definition block for a group (`m_TypeBlock='3'`);
Data part:

```
long m_nextBlock;
short m_Number;
short m_Z_Order;
int m_countSubGroups;
WINDOWPLACEMENT m_placeGroupWindow;
18 bytes;                                   //reserve field
char m_nameGroup;
```

Comment: short m_Z_Order defines the order of appearance of the Windows on the screen. (18 bytes form a reserve field -not for user).
The Definition block "Group" is followed by definition blocks for the subgroups that are included in this group

Type 4 - Definition block for a subgroup (m_TypeBlock='4'):
Data blocks appear after all definition blocks have passed. Each file may obtain data blocks of two types: type 5 and type 7. Type 5 block contains data of a single signal (channel) only. A type 7 block contains data of several channels as well as some additional service information from the DAP.

Comment: m_TypeOverlap =1 if the channels of this subgroup are non-overlapped or Zero if they are overlapped – m_bisMaximized =1 if this subgroup window is in Maximized mode or Zero (0) if it is in Normal mode.

Type 5 - Data block for one channel (m_TypeBlock='5'):

The data part of this block starts with field -UINT m_Number; which is the identifier of the channel, after which actual data of this signal will be in he "short" format type.

Type 7 - Data block for number of channels (m_TypeBlock='7');
The data part of this block starts with field - UINT m_Number_Board, which is the identifier of the board, after which the actual data from the DAP is entered. This data has a specific structure and in addition to digitized signal values it also contains special service items.

The first integer (2 bytes) specifies the data that follows next. In decimal notation it has four digits. We will mark it as FINT for possible use as a whole or parts thereof, as separate numbers. The FINT can be used for:

1. Channel specification. Then F=0 for analog input, F=1 for digital input. The rest of the FINT is used for the channel's pin number; NT={channel's pin number}+1.

2. Information about DAP buffers filling: FINT= -111.The length of his data - 2 bytes (for Replay is not used).

3. DAP returns RMS-value: FINT= -222. The length of the RMS (Root Mean Sauare) data is 2 bytes.

4. Restart of the DAP: FINT= -444.The length of the following data (channel number) - 2 bytes.

5. Information about data loss: FINT= -333. The following data consist of 10 bytes: 2 bytes for channel number, 4 bytes for the first lost Sample and 4 bytes for the last lost Sample.

Now we should return to the Item 1 to watch the main data flow. The channel number identifier FINT is followed by a set of Sample data. For each Sample data 2 bytes (integer) are used. The length of this set (in Samples) depends on the channel type (continuous, level, external trigger and digital).

The block size for the continuous channels is: m_ContBlkSize, which the user can take from

Type 2 - "Definition blocks for channels" for continuous channels

For software versions up to 1.9, the user should calculate the block size as follows:

For a continuous channel the amount of Samples in that set is approximately equal to the product of Sampling Rate and Window Duration (with restriction: not more then 800 and not less than 3). To receive the precise value of the Samples set length the appropriate part of the 'C' program is presented.

Text Box:

```
long Samples =m_SampleRate*m_Duration;    // Samples in Window
if(Samples < 3)
Samples = 3;

// to get the biggest Block as possible
m_BlockSize = 800;
if(Samples <= m_BlockSize)
{
m_NumBlocks = 1;
m_BlockSize = Samples;
}
else
{
m_NumBlocks = Samples/m_BlockSize + 1;

// if less then 4 block, then a bad screen view
if(m_NumBlocks < 4)
{
m_NumBlocks = 4;
}
m_BlockSize = Samples/m_NumBlocks+1;
}
```

Comment: The value of variable m_BlockSize is a Samples set length.
Samples= (int)((Pre-trigger time) x (S.Rate)) + (int)((Posttriggeer time) x (S.Rate))
For the level and external trigger channels to receive the length of Samples set (in Samples) it is necessary to use the next procedure.
Comment: 2 integers are added to cancel losses because of integer multiplication and 2 integers (4 byte) for timestamp; this means that we have 2 at the end of the formula.

For digital channels, after FINT only 3 integers follow, the first 4 bytes for time of event (in Samples) and the last 2 bytes for presence of event (1 - Yes, 0 - No)

List of external library functions

```
EXPORT FILE* Open( const char* fn);
```

Description:
Opens the specified file and returns the files handle.

Parameters:
`const char* fn` - file name.

//

```
EXPORT void Close ();
```

Description:
Closes the opened file.

//

```
EXPORT int ReadMAP( int BufferSize, short Marker, short* Buffer);
```

Description:
Creates channelCount objects of class SpikeChannel and initializes their parameters. Sets the global variable corresponding to the number of channels to analyze simultaneously to channelCount.

Parameters:
`int BufferSize - size` of the buffer to fill with read data
`short Marker - channel` marker (10002 - 1st channel, 10003 - 2d channel etc.)
`short* Buffer` - data buffer handle.

//

```
EXPORT float GetSamplingRate(short Marker);
```

Description:
Returns recording sampling rate of the specified channel.

Parameters:
`short Marker - channel` marker (10002 - 1st channel, 10003 - 2d channel etc.)

//

```
EXPORT float GetAmplitude(short Marker);
```

Description:
Returns maximal signal amplitude of the specified channel.

Parameters:
`short Marker - channel` marker (10002 - 1st channel, 10003 - 2d channel etc.)

Appendix C

Hypothesis of parallelism of regression lines

Lets denote θ^1 the probability, that neuron activity registered under condition I corresponds to type A and θ^2 – the same probability for the activity of a neuron registered under condition II. The zero hypothesis is: $H^0 : \theta^1 = \theta^2$. For neuronal activity type A we have a set of random values $\{x_i^1\}$, observed under condition I and a set of random values $\{x_i^2\}$ observed under condition II. Sets $\{x_i^1\}$ and $\{x_i^2\}$ are binomial distribution: $p(x_i^j) = B(\theta^j, \theta^j(1-\theta^j)n_i^j)$, i=1,2; n_i – sets lengths.

Lets consider random value $z_i^j = \frac{x_i^j}{\sqrt{n_i^j}}$; The distribution of random value is binomial of the following type: $p(z_i^j) = B(\theta^j\sqrt{n_i^j}, \theta^j(1-\theta^j))$, i.e we obtained a mathematical expectation depending on $\sqrt{n_i^j}$, but the variance is equal for the both sets $\{z_i^1\}$ and $\{z_i^2\}$, which is equal $\theta^j(1-\theta^j)$.

Представим z_i^j в виде: $z_i^j = \theta^j\sqrt{n_i^j} + \xi$, где ξ – случайная величина с распределением $p(\xi) = B(0, \theta^j(1-\theta^j))$. Таким образом, z_i^j может быть представлена в виде линейной регрессии относительно $\sqrt{n_i^j}$ с коэффициентом θ^j. Тогда гипотеза $H^0 : \theta^1 = \theta^2$ является гипотезой о параллельности линий регрессии. The hypothesis may be verified with the Fisher test (Ivchenko et Medvedev, press).

To apply the Fisher test for the comparison of two datasets $\{z_i^1\}$ and $\{z_i^2\}$ each of them must contain at least 5 samples. More over, Fisher test supposes some the assumption of normality of analyzed distributions. Binomial distributions $p(z_i^j) = B(\theta^j\sqrt{n_i^j}, \theta^j(1-\theta^j))$ converge to normal distribution and may be considered as normal if

$$n_i^j \geq \frac{9}{\theta^j(1-\theta^j)}.$$

Bibliography

Abe, K., Asai, Y., Matsuo, Y., Nomura, T., Sato, S., Inoue, S., Mizukura, I., Sakoda, S., 2003. Classifying lower limb dynamics in Parkinson's disease. Brain Res. Bull. 61, 219–226.

Abeles, M., 1991. Corticotronics: Neural circuits of the cerebral cortex, Cambridge University Press, New York.

Abeles, M., Goldstein, M.H.Jr., 1977. Multispike train analysis. Proc. IEEE 65, 762-773

Abeles, M., 1982. Quantification smoothing and confidence limits for single units histograms. J. of Neurosc. Methods 5, 317-325.

Aksenova, T.I., Chibirova, O.K., Tetko, I.V., Driga, A.A., Villa, A.E.P., 2000, Recognition of neuron impulses based on non linear dynamic equations. SICPRO'2000, Moscow 25-30 sept 2000. (in Russian: Распознавание импульсов нейронов на основе нелинейных уравнений динамики. - Тез. Докл. Междунар. Конф. «Идентификация систем и задачи управления")

Aksenova, T.I., Tetko I.V., Chibirova O.K., Villa,A.E.P., Recognition of neuron impulses by means of non linear equations. "Journal of Automation and Information Science", 2001 - №3 - P.16-27.

Aksenova, T., Chibirova, O., Benabid, A.-L., Villa, A., 2002. Nonlin- ear oscillation models for spike separation. Lecture Notes Com- put. Sci. 2526, 61–70.

Aksenova, T., Chibirova, O., Dryga, A., Tetko, I., Benabid, A.-L., Villa, A., 2003. An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals. Methods, A Companion Methods Enzymol. 30, 178–187.

Aksenova, T.I., and Shelekhova, V. Y., 1994. Cybernetics and Computing Technology 103, 79-83.

Aksenova, T.I., and Shelekhova, V. Y., 1997. Cybernetics and Computing Technology 107, 62-71.

Aksyonova, T.I., Shelekhova, V.Y., 1995. Efficient algorithms of derivative estimation for noisy observations. SAMS 18, 159–163.

Aksenova, T.I., and Chibirova, O. K., 1996. *18th Ann. Int. Conf. of IEEE*, Amsterdam.

Aksenova, T.I., Tetko, I. V., Ivakhnenko, A. G., Villa, A. E. P., Welsh, W. J., and Zielinski, W. L., 1999. *Anal. Chem.* **71,** 2423-2430

Albin, R.L., Young, A.B., Penney, J.B., 1989. The functional anatomy of basal ganglia disorders. Trends Neurosci 12, 366 – 75.

Alexander, G.E., Crutcher, M.D., DeLong, M.R., 1990. Basal ganglia- thalamocortical circuits: parallel substrates for motor, oculo- motor, 'prefrontal' and 'limbic' functions. Prog Brain Res 85, 119 – 46.

Alexander. G.E., Crutcher. M.D., Functional architecture of basal ganglia circuits: neural substrates of parallel processing. Trends Neurosci 13, 266 – 71.

Ashkan, K., Wallace, B, Bell, B.A., Benabid, A.L., 2004. Deep brain stimulation of the subthalamic nucleus in Parkinson's Disease 1993 – 2003: where are we 10 years on? British Journal of Neurosurgery, 18(1), 19 – 34

Aziz, T.Z., Peggs, D., Sambrook, M.A., Crossman, A.R., 1991. Lesion of the subthalamic nucleus for the alleviation of 1-methyl-4- phenyl-1,2,3,6-tetrahydropyridine (MPTP)-induced Parkin- sonism in the primate. Mov Disord 6(4), 288 – 92.

Bahoura, M., Rouat, J., 2001. A new approach for wavelet speech enhancement. IEEE Signal Process. Lett. 8 (1), 10–12.

Bejjani, B., Damier, P., Arnulf,I., 1997. Pallidal stimulation forParkinson's disease. Two targets? Neurology 49(6), 1564-1569.

Bejjani, B.P., Dormont, D., Pidoux, B., Yelnik, J., Damier P., Arnulf .I, 2000. Bilateral subthalamic stimulation for Parkin- son's disease by using three-dimensional stereotactic mag- netic resonance imaging and electrophysiological guidance. J Neurosurg 92, 615 – 25.

Benabid, A.L., Benazzouz, A., Limousin, P., Koudsie A., Krack P., Piallat B., 2000. Dyskinesias and the subthalamic nucleus. Ann Neurol 47(Suppl 1), 189 – 92.

Benabid, A.L., Koudsie A., Benazzouz, A., Le Bas, J.F., Pollak P., 2002. Imaging of subthalamic nucleus and ventralis intermedius of the thalamus. Mov Disord 17(Suppl 3), 123.

Benabid, A.L., Pollak, P., Gao, D.M, Hoffmann, D, Limousin, P., Gay E., 1996. Chronic electrical stimulation of the ventralis intermedius nucleus of the thalamus as a treatment of movement disorders. J Neurosurg 84, 203 – 14.

Benabid, A.L., Pollak, P., Gervason,C, Hoffmann, D., Gao D.M., Hommel M., 1991. Long-term suppression of tremor by chronic stimulation of the ventral intermediate thalamic nucleus. Lancet 337(8738), 403 – 6.

Benabid, A.L., Pollak P., Louveau, A., Henry, S., de Rougemont, J., 1987. Combined (thalamotomy and stimulation) stereotactic sur- gery of the VIM thalamic nucleus for bilateral Parkinson disease. Appl Neurophysiol 50(1 – 6), 344 – 6.

Benabid, A.L., Pollak, P., Gross, C., Hoffmann, D., Benazzouz, A., Gao, D.M., Laurent, A., Gentil, M., Perret, J., 1994. Acute and long-term effects of subthalamic nucleus stimulation in parkin- son's disease. Stereotactic Func. Neurosurg. 62, 76–84.

Benazzouz, A., Boraud, T., Feger, J., Urbaud, P., Bioulac, B., Ross C., 1996. Alleviation of experimental hemiParkinsonism by high- frequency stimulation of the subthalamic nucleus in primates: a comparison with L-dopa treatment. Mov Disord 11, 627 – 32.

Benazzouz, A., Breit, S., Koudsie, A, Pollak, P., Krack, P., Benabid, A.L., 2002. Intraoperative microrecordings of the subthalamic nucleus in Parkinson's disease. Mov Disord 17(Suppl 3), 145 – 9.

Benazzouz, A., Gross, C., Feger, J., Boraud, T, Bioulac, B., 1993. Reversal of rigidity and improvement in motor performance by subthalamic high-frequency stimulation in MPTP-treated monkeys. Eur J Neurosci 5, 382 – 9.

Benazzouz, A., Gao, D.M., Ni, Z.G., Piallat, B., Bouali-Benazzouz, R., Benabid, A.-L., 2000. Effect of high-frequency stimulation of the subthalamic nucleus on the neuronal activities of the substan- tia nigra pars reticulata and ventrolateral nucleus of the thalamus in the rat. Neuroscience 99, 289–295.

Bergman, H., Deuschl, G., 2002. Pathophysiology of Parkinson's disease: from clinical neurology to basic neuroscience and back. Mov Disord 17(Suppl 3), 28 – 40.

Bergman, H., Wichmann, T., Karmon, B., DeLong, M.R., 1994. The primate subthalamic nucleus. II. Neuronal activity in the MPTP model of Parkinsonism. J Neurophysiol 72, 507 –20.

Bergman, H., DeLong, M.R., 1992. A personal computer-based spike detector and sorter: implementation and evaluation. J. Neurosci. Methods 41, 187–197.

Beurrier, C., Bioulac, B., Hammond C., 2000. Slowly inactivating sodium current (I(NaP)) underlies single-spike activity in rat subthalamic neurons.J Neurophysiol. 83(4), 1951-7.

Beurrier, C., Congar, P., Bioulac, B., Hammond, C., 1999. Subthalamic nucleus neurons switch from single-spike activity to burst-firing mode.J Neurosci. 19(2), 599-609.

Bevan, M.D., Wilson, C.J., 1999. Mechanisms underlying spontaneous oscillation and rhythmic firing in rat subthalamic neurons. J Neurosci. 19(17), 7617-28.

Bevan, M.D., Magill, P.J., Terman, D., Bolam, J.P., Wilson, C.J., 2002. Move to the rhythm: oscillation in the subthalamic nucleus- external globus pallidus network. Trends Neurosci. 25, 525–531.

Bogoljubov, N.N., Mitropolsky, Y. A., 1961. Asymptotic methods in the theory of non-linear oscillations. Gordon and Breach, New York.

Calne, D., Schulzer, M., Mak, E., Stoessl, A.J., 1993. Treatment of Parkinson's disease. New England Journal of Medicine 329: 1021-1024.

Campbell. G.A., Eckardt. M.J., Weight F.F., 1985. Dopaminergic mechanisms in subthalamic nucleus of rat: analysis using horseradish peroxidase and microiontophoresis. Brain Res. 333, 261 – 70.

Cassidy, M., Mazzone, P., Oliviero, A., Insola, A., Tonali, P., Laz- zaro, V.D., Brown, P., 2002. Movement-related changes in syn- chronization in the human basal ganglia. Brain 125, 1235–1246. Donoho, D.L., Johnstone, I.M., 1994. Ideal spatial adaptation by wavelet shrinkage. Biometrika 81 (3), 425–455.

Chandra, R., and Optican, L. M., 1997. *IEEE Transactions on Biomedical Engineering* **44**, 403-412

Cheesman. P., Shutz. J. A., 1988. Bayesian Classification system. 5th Intern. Conf. Of machine learning (San Francisco, 1988). - C.54-64.

Chertoprud. A.G., Gudzenko, L.I., Basov. N.G. (Ed.), 1976. Kinetics of Simple Models of Oscillating Theory, Nauka, Moscow.

Chikering. E.H., Orban. G.A., 1997. Efficient approximation for the marginal likelihood of Bayesian network with hidden variables. Machine Learning 29, 279-284.

Bibliography

Cuny. E., Guehl. D., Burbaud. P., Gross. C., Dousset. V., Rougier. A., 2002. Lack of agreement between direct magnetic resonance imaging and statistical determination of a subthalamic target: the role of electrophysiological guidance. J Neurosurg 97:591 – 7.

DeLong M.R. Activityof pallidal neurons during movement // Journal neurophysiology. - 1997. - №78. - C.199-213.

DeLong MR. Primate models of movement disorders of basal ganglia origin. Trends Neurosci 1990;13(7):281 – 5.

Dinning G.J. Real-time classification of multiunit neural signals using reduced feature set // IEEE Transaction Biomedical Engineering. - 1981. - №28. - C.804-812.

Egidi M, Rampini P, Locatelli M, Farabola M, Priori A, Pesenti A et al. Visualisation of the subthalamic nucleus: a multiple sequential image fusion (MuSIF) technique for direct stereotaxic localisation and postoperative control. Neurol Sci 2002;23(Suppl 2):S71 – 2.

Fee M. S., Mitra M.P., Klenfeld D. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-Gaussian variability // Journal of Neuroscience Methods. - 1996. - №69. - C.175-188.

Fee M. S., Mitra M.P., Klenfeld D. Variability of extracellular spike waveforms of cortical neurons // Journal of Neurophysiology. - 1996. - №6. – т.76 - C. 3823-3832.

Feldman J.F., Roberge F.A. Computer detection and analysis of neuronal spike sequences // Informatics. - 1971. - №9. - C.185-197.

Fillion M, Tremblay L. Abnormal spontaneous activity of globus pallidus neurons in monkeys with MPTP-induced Parkinsonism. Brain Res 1991;547: 142 – 51.

Forster, C., Handwerker, H., 1995. Automatic classification and anal- ysis of microneurographic spike data using a PC/AT. J. Neurosci. Methods 31, 109–118.

Friedman, D.H., 1968. Detection of signals by template matching. Johns Hopkins Press, Baltimor.

Gadike, R., Albus, K., 1995. Real-time separation of muktineuron recordings with a DSP32Csignal processor. J. Neurosci. Methods 57, 187–193.

Gerstain Design of laboratory for multineuron studies / Gerstain G.L., Bloom M.J., Espinosa I.E и др. // IEEE Trans. System. Man. Cybern. - 1983. - №13. - C 668-676.

Gillies. A, Arbuthnott. G., (2000) Computational models of the basal ganglia. Mov Disord. Sep;15(5):762-70

Glaser E.H., Ruchkin D.S. Principles of neurobiological signal analysis // New-York, "Academic press", 1976 – 471C.

Glaser E.M. Separation of neuronal activity by waveform analysis // Advances in Biomedical Engineering. - 1971. - №1. - C.77-136.

Glaser E.M., Marks W.B. On-line separation of interleaved neuronal puls sequences // Data Acquisition Process. Biol Med. - 1998. - №5. - C.137-156.

Gonsales Ту Дж., Гонсалес Р. Принципы распознавания образов. – М.:«Мир», 1978. – 411с.

Gozani S.N., Miller J.P. Optimal discrimination and classification of neuronal action potential waveforms from multiunit multichennel recording using software-based linear filters // IEEE Transactions in Biomedical Engineering. - 1994. - №41. – т.4. - C.358-372.

Gross C, Rougier A, Guehl D, et al. High-frequency stimulation of the globus pallidus internalis in Parkinson's disease: a study of seven cases. J Neurosurg 1997; 87(4): 491-498.

Gudzenko, L.I. (1962) Izvestiia Vuzov Radiophysics 5, 573-587

Gudzenko, L.I., 1962. The statistical method of determination of characteristics of a non-control autooscillating system. Izvestiia Vuzov Radiophisics 5, 572–587 (in Russian).

Guiot G, Arfel G, Derome P. La chirurgie stereotaxique des tremblements de repos et d'attitude. Gaz Med France1968; 75:4029 – 56.

Guiot G, Hardy J, Albe-Fessard D. Delimitation precise des struc- tures sous-corticales et identification de noyaux thalamiques chez l'homme par l'electrophysiologie stereotaxique. Neurochirurgia (Stutt) 1962; 51: 1-18.

Guridi J, Herrero MT, Luquin R, Guillen J, Obeso JA. Subthalamotomy improves MPTP-induced Parkinsonism in monkeys. Stereotact Funct Neurosurg 1994;62(1 – 4):98 – 102

Gurney, K., Humphries, M., Wood, R., Prescott, T., and Redgrave, P. (2004). Testing computational hypotheses of brain systems function: a case study with the basal ganglia. *Network: Computation in Neural Systems*, 15:263–290.

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

84:401–410.

Harris K.D Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements / Harris K.D., Henze D.A., Csicsvari J. и др. // Journal of Neurophysiology. - 2000. -N. 84. - C. 401-414.

Hashimoto, T., Elder, C.M., Okun, M.S., Patrick, S.K., Vitek, J.L.,2003. Stimulation of the subthalamic nucleus changes the firing rate pattern of pallidal neurons. J. Neurosci. 23, 1916–1923.

Hazrati LN, Parent A, Mitchell S, Haber SN. Evidence for interconnections between the two segments of the globus pallidus in primates: a PHA-L anterograde tracing Brain Res. 1990 Nov 12;533(1):171-5.study.

Henze, DA, Borhegyi, Z, Csicsvari, J, Mamiya, A, Harris, KD, & Buzsaki, G (2000). Intracellular features predicted by extracellular recording in the hipocampus in vivo. *J Neurophysiol* 84:390-400.

Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol. 1952 Aug;117(4):500-44.

Honey C., Gross, R.E., Lozano A.M., 1999. New Developments in the Surgery for Parkinson's Disease Can. J. Neurol. Sci. 26: Suppl. 2, 45-52.

Houeto JL, Welter ML, Bejjani BP, Tezenas du Montcel S, Bonnet AM, Mesnage V et al. Subthalamic stimulation in Parkinson disease: intraoperative predictive factors. Arch Neurol 2003;60:690 – 4.

Huber P. Хьюбер П., Робастность в статистике. - М.:Наука, 1984. – 304с.

Hulata, E., Segev. R., and Ben-Jacob, E. (2002) *Journal of Neuroscience Methods* **117**, 1-12

Humphrey D.R., Schmidt E.M. Extracellular single-unit recording methods // Neurophisiologcal Techniques, Neuromethods. - 1990. - №15. – С.138-145.

Hurtado, J.M., Gray, C.M., Tamas, L.B., Sigvardt, K.A., 1999. Dy- namic of tremor related oscillation in the human globus pallidus: a single case study. Proc. Natl. Acad. Sci. U.S.A. 16, 1674–1679.

Hutchison, W.D., Lozano, A.M., Tasker, R.R., Lang, A.E., Dostro- vsky, J.O., 1997. Identification and characterization of neurons with tremor-frequency activity in human globus pallidus. Exp. Brain Res. 113, 557–563.

Ivchenko Ивченко Г.И., Медведев Ю. И. Математическая статистика. – М.: Высшая школа, 1984. - 248с.

Kessler. Epidemiologic studies of Parkinson's disease [Abstract]. Americal Journal of Epidemiology 1972; 96: 242-254.

Kim K.H., Kim S.J. Method for unsupervised Classification of multiunit neural signal recording under low signal-to-noise ratio // IEEE Transactions on biomedical engineering. - 2003. - №4. – Vol.50. – C.421-431.

Kitai ST, Deniau JM. Cortical inputs to the subthalamus: intracellular analysis. Brain Res 1981;214:411 – 5.

Koller W, Pahwa R, Busenbark K, et al. High-frequency unilateral thalamic stimulation in the treatment of essential and parkinson- ian tremor. Ann Neurol 1997; 42: 292-299.

Kondziolka et al. A comparison between magnetic resonance imaging and computer tomography for stereotactic coordinate determination. Neurosurgery 1992; 30(3): 402-407.

Korn G., Korn T., 1984 (in Russian): Корн Г., Корн Т. Справочник по математике для научных работников и инженеров. - М.:«Наука», p831.

Krack P, Pollak P, Limousin P, Benazzouz A, Deuschl G, Benabid AL. From off-period dystonia to peak-dose chorea. The clinical spectrum of varying subthalamic nucleus activity. Brain 1999;122:1133 – 46.

Krack P, Pollak P, Limousin P, et al. Opposite motor effects of pallidal stimulation in Parkinson’s disease. Ann Neurol 1998; 43(2): 180-192.

Kreiter A.K., Aertsen A.J., Gerstein G.L. A low coast single board solution for real time unsupervised waveform classification of multineuron recording // Journal of neuroscience methods. - 1989. - №30. - C.59-69.

Kumar R, Lozano AM, Kim YJ, Het al. Double-blind evaluation of subthalamic nucleus deep brain stimulation in advanced Parkinson’s disease. Neurology 1998; 51: 850-855.

Lang AE, Lozano AM. Parkinson's disease. Second of two parts. N Engl J Med 1998;339:1130 – 43.

Lanotte MM, Rizzone M, Bergamasco B, Faccani G, Melcarne A, Lopiano L. Deep brain stimulation of the subthalamic nucleus: anatomical, neurophysiological, and outcome correlations with the effects of stimulation. J Neurol Neurosurg Psychiatry 2002;72:53 –8.

Letelier J.C., Weber P.P. Spike sorting based on discrete wavelet transform coefficients // Journal of Neuroscience Methods. - 2000. - №101. - C.93-106.

Levy, R., Ashby, P., Hutchison, W.D., Lang, A.E., Lozano, A.M., Dostrovsky, J.O., 2002. Dependence of subthalamic nucleus os- cillations on movement and dopamine in Parkinson's Disease. Brain 125, 1196–1209.

Levy, R., Hutchison, W.D., Lozano, A.M., Dostrovsky, J.O., 2000. High-frequency synchronizzation of neuronal activity in the sub- thalamic nucleus of parkinsonian patients with limb tremor. J. Neurosci. 20, 7766–7775.

Levy, R., Hutchison, W.D., Lozano, A.M., Dostrovsky, J.O., 2002. Synchronized neuronal discharge in basal ganglia of parkinso- nian patients in li;ited ascillatory activity. J. Neurosci. 22, 2855–2861.

Lewichki M.S. Bayesian modeling and classification of neural signals // Neural Comput. - 1994. - №6. - C.1005-1030.

Lewicki, M.S., 1998. A review of methods for spike sorting: the detection and classification of neural action potentials. Network Comput. Neural Syst. 9, 53–78.

Limousin P, Pollak P, Benazzouz A, Hoffmann D, Broussolle E, Perret JE et al. Bilateral subthalamic nucleus stimulation for severe Parkinson's disease. Mov Disord 1995;10:672 – 4.

Limousin, P., Krack, P., Pollak, P., Benazzouz, A., Ardouin, C., Hoff- mann, D., Benabid, A.-L., 1998. Electrical stimulation of the subthalamic nucleus in advanced parkinson's disease. N. Engl. J. Med. 339, 1105–1111.

Liu, X., Ford-Dunn, H.L., Hayward, G.N., Nandi, D., Miall, R.C., Aziz, T.Z., Stein, J.F., 2002. The oscillatory activity in the parkinsonian subthalamic nucleus 1667–1672.investigated using the macro- electrodes for deep brain stimulation. Clin. Neurophysiol. 113,

Magarinos-Ascone, C.M., Figueras-Mendez, R., Riva-Meana, C., Cordoba-Fernandez, A., 2000. Subthalamic neuron activity re- lated to tremor and movement in Parkinson's disease. Eur. J. Neurosci. 12, 2597–2607.

Marsden et al. Success and problems of long-term levodopa therapy in Parkinson's disease. Lancet 1977; 1: 345-349.

Martynov G.V., 1978 (in Russian): Мартынов Г.В. Критерий омега-квадрат. - М.:Наука, p78.

Matsuo, Y., Mizukura, I., Abe, K., 2003. A coupled oscillator model of disordered interlimb coordination in patients with Parkinson's disease. Biol. Cybern. 88, 152–162.

Miller WC, DeLong MR. Altered tonic activity of neurons in the globus pallidus and subthalamic nucleus in the primate MPTP model of Parkinsonism. In: Carpenter MBJA, ed., The basal ganglia 2. New York: Plenum Press, 1987:415 – 27.

Mirfakhraei K., Horch K. Classification of action potentials in multiunit intrafascicular recordings using neural network pattern-recognition techniques // IEEE Transactions in Biomedical Engineering. – 1994. - №1 - C.89-91.

Mouroux M, Hassani OK, Feger J. Electrophysiological study of the excitatory parafascicular projection to the subthalamic nucleus and evidence for ipsi- and contralateral controls. Neuroscience 1995;67:399 – 407.

Musial, P.G., Baker, S.N., Gerstein, G.L., King, E.A., and Keating, J.G. (2002) *Journal of Neuroscience Methods* **115**, 29-43

Nandi D, Aziz TZ, Liu X, Stein JF. Brainstem motor loops in the control of movement. Mov Disord 2002;17(Suppl3):S22 – 7.

Niktarash, A.H., 2003. Transmission of the subthalamic nucleus os- cillatory activity to the cortex: a computational approach. J. Com- put. Neurosci. 15, 223–232.

Nir Giladi and Eldad Melamed The Role of Functional Neurosurgery in Parkinson's Disease Functional Neurosurgery in Parkinsonism, IMAJ • Vol 2 • June 2000, 455-461 .

Nutt et al. On-off phenomenon: relation to levodopa pharmacokinetics and pharmacodynamics. Annals of Neurology 1987; 22: 535-540.

Ohye C, Shibazaki T, Hirai T, Wada H, Hirato M, Kawashima Y. Further physiological observations on the ventralis intermedius neurons in the human thalamus. J Neurophysiol 1989;61:488 – 500.

Okada M., Maruyama N. Software system for real-time discrimination of multy-unit nerve impulses // Computer programming in biomedicine. - 1982. - №14. – Vol.2. - C.57-64.

Oweiss G.K., Anderson D.J. Spike sorting: a novel shift and amplitude invariant technique // Neurocomputing. – 2002. - №44. – C.1133-1139.

Parent A, Hazrati LN. Functional anatomy of the basal ganglia. I. The cortico-basal ganglia-thalamo-cortical loop. Brain Res Brain Res Rev. 1995 Jan;20(1):91-127. Review.

Peterson B.E., Merzenich M.M. Macintosh program for automating data acquisition and analysis applied to neurophysiology // Journal of neuroscience method. - 1995. - №57. - C.121-131.

Plenz, D., Kitai, S.T., 1999. A basal ganglia pacemaker formed by the subthalamic nucleus and external globus pallidus. Nature 400, 677–682.

Pollak P, Krack P, Fraix V, Mendes A, Moro E, Chabardes S et al. Intraoperative micro- and macrostimulation of the subthalamic nucleus in Parkinson's disease. Mov Disord 2002;17(Suppl 3): S155-S161.

Pralong E, Ghika J, Temperli P, Pollo C, Vingerhoets F, Villemure JG. Electrophysiological localization of the sub- thalamic nucleus in Parkinsonian patients. Neurosci Lett 2002;325:144 – 6.

Press W.H., Teukolsky S.A., Vetterling W.T., Flannerly B.P. Numerical recipes in C, the art of scientific computing / - Cambridge:"University press", 1994. – 994c.

Quian Quiroga R, Nadazy Z, Ben-Saul Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering // Neural Computation. - 2004. - №16. C.1661-1687.

Quirk M.C., Wilson M.A., Interaction between spike waveform classification and temporal sequence detection // Journal of neuroscience methods. – 1999. - №94. – C.41-52.

Roberts W.M., Hartline D.K. Separation of multiunit nerve impulse trains by a multichannel linear filter algorithm // Brain Researches. - 1975. - №94. - C.141-149.

Rodrigues, M.C., Guiridi, O.J., Alvarez, L., Mewes, K., Macias, R., Vitek, J., DeLong, M.R., Obeso, J., 1998. The subthalamic nu- cleus and tremor in Parkinson's disease. Movement Disord. 13(Suppl.), 111–118.

Rodriguez-Orzoz MC, Rodriguez M, Guridi J, Mewes K, Chockkman V, Vitek J et al. subthalamic nucleus in Parkinson's disease: somatotopic organization and physiological characteristics. Brain 2001;124:1777 – 90.

Russchen FT, Bakst I, Amaral DG, Price JL. The amygda- lostriatal projections in the monkey. An anterograde tracing study. Brain Res 1985;329:241 – 57.

Sahani M. Latent Variable Models for neural data analysis // Thesis for the degree of philosophy doctor. - California institute of technology, 1999.

Salganicoff Unsupervised waveform classification for multi-neural recordings / Salganicoff M., Sarna M., Sax L., Gerstein G.L. // Journal of Neuroscience Methods. - 1988. - №25. - C.181-187.

Schaltenbrand G, Wahren W. Atlas for stereotaxy of the human brain, 2nd edn. Stuttgart: Georg Thieme Verlag, 1977.

Schmidt, E.M., 1984. Computer separation of multi-unit neuroelec- tric data: a review. J. Neurosci. Methods 12, 95–111.

Shoham S., Fellows M.R., Normann R.A. Robust automatic sorting using mixture of multivariate t-distribution // Journal of neuroscience methods. – 2003. N 127. – C.111-122.

Simon W. The real-tome sorting of neuroelectric action potentials in multiple unit studies // Electrophysiology and Clinical Neurophysiology. - 1965. - №18. – C.192-195.

Singer, W. (1999) *Neuron* **24**, 49-65

Snider R.K., Bond A.B. Classification of non-stationary neural signals // Journal of neuroscience methods. - 1998. - №84. - C.155-166.

Starr PA, Christine CW, Theodosopoulos PV, Lindsey N, Byrd D, Mosley A et al. Implantation of deep brain stimulators into the subthalamic nucleus: technical approach and magnetic resonance imaging-verified lead locations. J Neurosurg 2002;97:370 – 87.

Sterio D, Zonenshayn M, Mogilner AY, Rezai AR, Kiprovski K, Kelly PJ et al. Neurophysiological refinement of sub- thalamic nucleus targeting. Neurosurgery 2002;50:58 – 69.

Stern and Hurtig (eds) (1999). The medical clinics of North America: Parkinson's disease and Parkinsonian syndromes. W.B. Saunders Company.

Talairach J, David M, Tournoux P, Corredor H, Kvasina T. Atlas d'anatomie stereotaxique des noyaux gris centraux. Paris: Masson, 1957.

Tasker RR, Siqueira J, Hawrylyshyn P, Organ LW. What happened to VIM thalamotomy for Parkinson's disease? Appl Neurophysiol 1983;46(1 – 4):68 – 83.

Tasker, RR. Thalamotomy. Neurosurg Clin North Am 1990; 1: 841-864 The primate subthalamic nucleus. III. Changes in motor behavior and neuronal activity in the internal pallidum induced by subthalamic inactivation in the MPTP model of parkinsonism. J Neurophysiol. 1994 Aug;72(2):521-30.

Teukolsky S.A., Vetterling W.T., Flannerly B.P. Numerical recipes in C, the art of scientific computing / Press W.H., - Cambridge:"University press", 1994. – 994c.

Tronnier VM, Fogel W, Kronenbuerger M, Steinvorth S. Pallidal stimulation: an alternative to pallidotomy? J Neurosurg 1997; 87(5): 700-705.

Villa, A.E.P., Lorenzana, V.M.B., 1997. Ketamine modulation of the temporal pattern of discharges and spike train interactions in the rat substantia nigra pars reticulata. Brain Res. Bull. 43, 525–535.

Villa, A.E.P., Tetko, I.V., Hyland, B. and Najem, A. (1999) *Proceedings of the National Academy of Sciences of the United States of America* **96**, 1106-1111

Villa, A.E.P. (2000) *in* Time and the Brain (R. Miller, Ed), *Conceptual advances in brain research* **2**, 1-51, Harwood Academic Publishers.

Wheler B.S., Heetderks W.J. A comparison of techniques for classification of multiple neuron signals // IEEE Transactions on biomedical engineering. - 1982. - №12. - C.752-759.

Wichmann T, Bergman H, DeLong MR. The primate subthalamic nucleus. III. Changes in motor behavior and neuronal activity in the internal pallidum induced by J Neurophysiol. 1994 Aug;72(2):521-30.subthalamic inactivation in the MPTP model of parkinsonism.

Wichmann, T., DeLong, M.R. (2003) Functional neuroanatomy of the basal ganglia in Parkinson's disease. Advances in neurology, Vol. 91:9 – 18.

Wood On the variability of manual spike sorting / Wood F., Black M.J., Vargas-Irwim C., Donoghue J.P. // IEEE Transaction on biomedical engineering. – 2004. - №6. – Vol.51. – C.912-918.

Yamada Data processing for multichennel optical recordings: action potential detection by neural network / Yamada S., Kage H., Nakashima M. и др. // Journal of Neuroscience Method. - 1992. - №43. - C.23-33.

Yelnik J., 2002. Functional anatomy of the basal ganglia. Mov Disord 17(Suppl 3), 15 – 21.

Zonenshayn M., Rezai A.R., Mogilner A.Y., Beric A., Sterio D., Kelly P.J., 2000. Comparison of anatomic and neurophysiological methods for subthalamic nucleus targeting. Neurosurgery 47, 282 – 94.

Zouridakis G., Tam D.C., 2000. Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering. Computer methods and programs in biomedicine 61, 91-98.