Uppsala University
Signals and Systems

A SIGNAL PROCESSING APPROACH TO PRACTICAL NEUROPHYSIOLOGY

A Search for Improved Methods in Clinical Routine and Research

Björn Hammarberg

UPPSALA UNIVERSITY 2002

Dissertation for the degree of Doctor of Philosophy
in Signal Processing at Uppsala University, 2002

ABSTRACT
Hammarberg, B., 2002. A Signal Processing Approach to Practical Neurophysiology: A Search for Improved Methods in Clinical Routine and Research, 222 pp. Uppsala. ISBN 91-506-1551-3.

Signal processing within the neurophysiological field is challenging and requires short processing time and reliable results. In this thesis, three main problems are considered.

First, a modified line source model for simulation of muscle action potentials (APs) is presented. It is formulated in continuous-time as a convolution of a muscle-fiber dependent transmembrane current and an electrode dependent weighting (impedance) function. In the discretization of the model, the Nyquist criterion is addressed. By applying anti-aliasing filtering, it is possible to decrease the discretization frequency while retaining the accuracy. Finite length muscle fibers are incorporated in the model through a simple transformation of the weighting function. The presented model is suitable for modeling large motor units.

Second, the possibility of discerning the individual AP components of the concentric needle electromyogram (EMG) is explored. Simulated motor unit APs (MUAPs) are prefiltered using Wiener filtering. The mean fiber concentration (MFC) and jitter are estimated from the prefiltered MUAPs. The results indicate that the assessment of the MFC may well benefit from the presented approach and that the jitter may be estimated from the concentric needle EMG with an accuracy comparable with traditional single fiber EMG.

Third, automatic, rather than manual, detection and discrimination of recorded C-fiber APs is addressed. The algorithm, detects the APs reliably using a matched filter. Then, the detected APs are discriminated using multiple hypothesis tracking combined with Kalman filtering which identifies the APs originating from the same C-fiber. To improve the performance, an amplitude estimate is incorporated into the tracking algorithm. Several years of use show that the performance of the algorithm is excellent with minimal need for audit.

Keywords: matched filter, asynchronous detection, Kalman filter, initialization, MHT, Wiener deconvolution, line source model, electromyography, needle EMG, motor unit potential, MUAP, mean fiber concentration, jitter, microneurography, C-fiber, spike sorting

Björn Hammarberg, Signals and Systems, Uppsala University, PO Box 528, SE-751 20 Uppsala, Sweden. Email: Bjorn.Hammarberg@signal.uu.se.

ISBN 91-506-1551-3
Printed in Sweden by Elanders Gotab AB, Stockholm 2002
Distributed by Signals and Systems, Uppsala University, Uppsala, Sweden

To everyone who finds it useful

CONTENTS

LIST OF TABLES

The answer is out there, Neo.
It's looking for you and it will find you,
if you want it to.

Trinity, in *The Matrix*.

Preface and acknowledgments

My main purpose when setting out on this journey six years ago was to develop algorithms that may actually be used in practice. Consequently, the applications presented herein are tailored to work reliably in a practical situation and, when being in conflict, optimality has been traded for applicability.

From time to time, challenging obstacles have crossed my path and I have often asked myself if this really was what I wanted. Obviously, I have been able to surmount the barriers sufficiently often, and in retrospect, I may proudly notice that things have worked out remarkably well.

Hopefully, this work may contribute in making this world a slightly better place; not necessarily for mankind but maybe in the everyday life of the physician examinating his patient, the patient being diagnosed, the researcher analyzing his experiments, or the engineer challenged with a signal processing problem.

Regarding the thesis, it is intended for readers ranging from an engineer with a biomedical interest to a physician interested in engineering. To comply with this intention, I have put an effort in providing background information both from medicine and engineering.

To facilitate the applicability of the algorithms presented, I have as far as possible provided all details and reasonings necessary to re-implement and apply the algorithms in reality. In the literature, this type of information is often omitted which may cause a lot of frustration.

The thesis consists of two main portions of which the second part, the applications, makes up the main contribution of this work. The first part provides the signal processing foundation on which the applications are built. Hence, the knowledgeable reader, familiar with the signal processing concepts presented therein, may benefit the most from using it as a reference.

The introduction of the thesis provides a comprehensive overview that is primarily intended for persons non-familiar with neurophysiology. It provides the basic knowledge needed to fully appreciate the rest.

Several persons have had a part in this work. First of all, I would like to express my deepest gratitude to my supervisors Anders Ahlén, Mikael Sternad, Erik Stålberg, and Erik Torebjörk for their unconditional support and tutorship during these years.

There are also a number of people without whom I probably wouldn't have done this at all. I hereby gratefully acknowledge: Lars Antoni, who advised me to call Erik Stålberg and apply for a job; Karin Rouleveld, for introducing me to her supervisor; Dick Stegeman, for arranging a job for me at his department in Nijmegen; all his staff for providing an excellent atmosphere; and especially, the PhD students who thrilled me to become a PhD student myself upon coming home.

During the course as a research engineer and, in time, a PhD student I have learnt to know many wonderful persons. I would like to thank all my colleagues at the Department of Clinical Neurophysiology for providing such a friendly and supportive environment. I especially mention the other engineers Peo Fällmar, Lars Karlsson, Stefan Stålberg, David Ullström, Per Ytterberg, and Mats Åström for all enjoyable moments at work and at the traditional Svenssons events. Special thanks goes to David who helped me with one of the images in the introduction.

I would like to acknowledge Lena Eriksson and Maggan Grindlund for always being so cheerful. Many thanks to Lena for providing me with EMG signals.

I express my gratitude towards Clemens Forster, Martin Schmelz, Christian Weidner, and all the others at the Erlangen lab for great company and partnership.

I thank all my colleagues at the Signals and Systems group for all memorable moments and events. Thanks goes to the other PhD students and especially my roommates Torbjörn Ekman and Nilo Casimiro-Ericsson. Special thanks goes to Claes Tidestav for all invaluable help with various LaTeX issues, to Jonas Öhr for being who he his and for all interesting discussions, and Ove Ewerlid for providing me with the basic linux setup and teaching me how to improve it further.

Anders Svärdström deserves a special acknowledgment for his extensive experience in putting theory into practice and for his invaluable help in this regard.

Most importantly, I would like to thank my wonderful wife Anna-Carin for your love and support. You have been completely invaluable and the most important reason for my success. Thank you for your excellent help with proofreading the manuscript. Moreover, you deserve a special acknowledgment for taking care of all things I have neglected the last several weeks.

Finally, I thank my precious and joyful son Fredrik for all unforgettable moments you give; I have really missed your company during this busy time. With this put to an end, I am looking forward to all the time we will spend together.

Björn Hammarberg
Uppsala, 19 March 2002.

CHAPTER 1
Introduction

UNDERSTANDING how nerve and muscle cells work is an ancient interest. There are written records from 1700 B.C. that claims to be a transcription of an even older document describing a thorough neurophysiological case study [18] [31]. Modern neurophysiology, however, did not begin to formalize until Luigi Galvani, around 1780-1790 A.D., caused the famous twitches of a frog leg during a dissection. He discovered that the twiches were provoked by touching the nerve of the leg with a metal knife.

Upon trying to explain Galvanis observation, Alessandro Volta had discovered that a current (later known as a Galvanic current) is induced if two metal plates are connected via an electrolytic medium.[1] Based on this finding, Volta stated in 1792 that the twitches in the frog leg were provoked by an electrical current between the steel knife and the tin plate upon which the leg was lying. Later experiments proved Volta right.

This was the beginning of an important paradigm shift that established the base of modern neurophysiology. Prior to Galvani's discovery, Descartes and others claimed that the nerves were hollow tubes carrying "vital spirits" but now it was clear that the nerves were electrical conductors of some kind. Further experiments showed that even the muscles had many similarities with the nerves in this respect.

Progressively, new discoveries adding to the knowledge of the functions of muscles and nerves were being made. Due to technical difficulties, however, most advances during the 19th century were on the anatomical level. The mystery of whether the nerves were similar to metal wire conductors or whether the situation was more complex remained.

[1] This discovery was later used by Volta to invent the first electric battery, the Voltaic pile.

Owing to instrumentation improvements, measurements that provided new insights became possible to carry out and, as the end of the 19th century was approaching, the functional knowledge of nerves and muscles increased rapidly. To mention a few achievements, Hermann von Helmholtz was in 1852 able to measure the conduction velocity of a nerve signal, realizing it was much too small to be explained by simple conduction as in a metal wire.

Further indications along this line was obtained when Sidney Ringer around 1880 discovered that in order for an isolated frog heart to continue beating, salts needed to be present in the surrounding solution. Specifically, sodium (Na), calcium (Ca), and potassium (K) salts were needed and had to be in special concentrations relative to each other.

A clear hypothesis for the conduction of the nerve signal was not available until Julius Bernstein in 1902 proposed that the inactive nerve or muscle fiber is normally electrically polarized (negative inside) and that the *action potential* (AP), as he called it, is a self perpetuating depolarization and repolarization of the cell membrane. He also made the first real theoretical contribution in assuming a semi-permeable cell membrane that could help explain the resting potential through differences in potassium ion (K+) concentrations on the inside and outside of nerve and muscle cells.

With the invention of the cathode ray oscilloscope in 1897, the measurements were significantly improved, both in accuracy and in feasibility. In 1913, the Nobel Prize winner, Edgar Douglas Adrian determined that the signal being transmitted is pulse-code modulated, i.e., rather than being conducted as continuous signals, the information is conducted as pulses in all-or-nothing responses with the pulse repetition frequency proportional to the strength of the signal.

1.1 The constitution and function of nerves

One of the most annoying and irrefutable findings against the hollow-nerve theory of Descartes was the repeated anatomical reports of nerves actually being anything but hollow. Further anatomical studies revealed that the nerve trunks are actually constituted by nerve fibers, or *axons*. These are responsible for conducting the AP from the nerve cell body to its destination at the *nerve terminal*, see Figure 1.1. The input to the cell is acquired through the so-called *dendrites*. With these and other findings, a profound neurophysiological knowledge of the nerves was beginning to formalize.

Still, the key principle behind the nerve conduction was unknown. To get further knowledge of the electrochemical events taking place in the nerve cell during the AP conduction, John Z. Young was in 1937 one of the first to make use of the

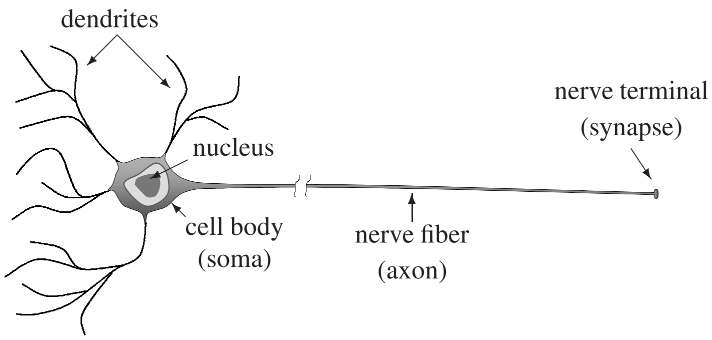


Figure 1.1: A nerve cell with the most important parts indicated.

giant axon in squid. This axon has a diameter of about 0.8 mm which is approximately 100 to 1000 times larger than other animal neurons; hence the attribute "giant".

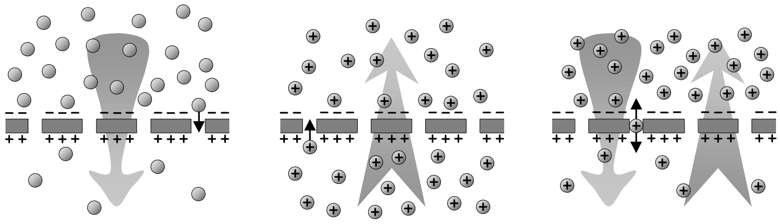
The ease of working with large neurons made important experiments possible for the first time. This included the first intracellular recordings of the nerve cell AP as well as the first measurements of the underlying ionic currents which produce them.

1.1.1 The electrical properties of the cell membrane

Around 1940, Alan L. Hodgkin together with Andrew F. Huxley and, independently, Kenneth S. Cole along with H. J. Curtis made the first measurements of the actual membrane voltage during an AP. The results were somewhat unexpected. From Bernstein's hypothesis, the membrane voltage was anticipated to increase from its negative value about -65 mV to zero during the depolarization of the membrane. Instead, the potential continued to increase and peaked near +50 mV.

It was suggested that this observation might arise from the membrane becoming selectively permeable to sodium ions (Na^+) and this was later confirmed by Hodgkin and Katz in 1949.

To explain why the permeability of specific ions and the peak value of the membrane voltage are important factors, we note that there are two forces which act on the ions over the membrane, see Figure 1.2. First, if the *concentration* of a certain ion is different on the two sides of the membrane, this *concentration gradient* asserts a force on the ion towards the side with the *lowest* concentration to level out the difference. Second, if the *potential* is different on the two sides of the membrane, this *voltage gradient* asserts a force on a positive ion towards the side with *negative* potential and vice versa for negative ions.



(a) (b) (c)

Figure 1.2: The effect on particles of concentration and voltage gradients over a cell membrane. In all diagrams, the compartment above the membrane has a lower potential than the compartment below the membrane. (a) The concentration of neutral particles above the membrane is higher than below which causes a downward net flow. (b) The concentration of the positively charged particles are the same on both sides of the membrane and, hence, there is no concentration gradient. Due to the potential difference over the membrane, however, there is an upward net flow of positively charged particles. (c) The concentration of particles is higher above the membrane than below which tends to push particles downwards. The positive charge of the particles in combination with the voltage gradient, however, tends to push particles upwards. For certain potential and concentration differences, these two effects cancel each other which results in a zero net flow.

At the *equilibrium potential* for a particular ion, the concentration and voltage gradients just balance and there is no net flow of that ion across the membrane. This potential is given by the *Nernst equation* that for a univalent ion at 20 °C is

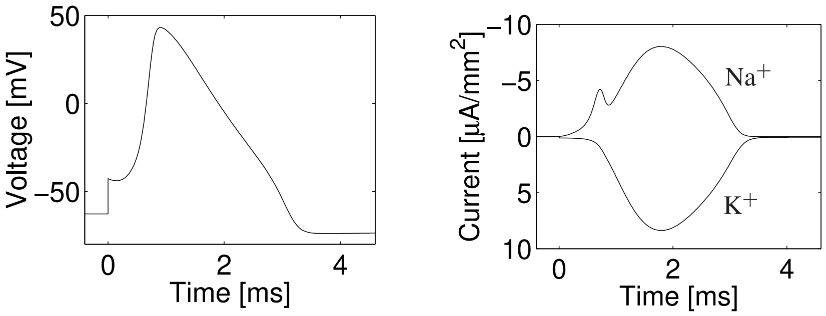
$$V = 58 \log_{10} \frac{C_o}{C_i} \quad [\mathrm{mV}]$$

where V is the equilibrium potential (internal minus external), and C_o and C_i are the outside and inside concentrations of the ion, respectively.

If we insert the concentrations found in the squid axon of the potassium (K^+), sodium (Na^+), and chloride (Cl^-) ions, we obtain the following equilibrium potentials [67, p. 44]

$$\begin{aligned} V_K &= -75\,mV \\ V_{Na} &= +55\,mV \\ V_{Cl} &\approx -65\,mV \quad . \end{aligned}$$

Knowing that the axon's resting potential is about -65 mV, the following three things can be noted. First, the chloride ions are in equilibrium and no net flow over



(a) (b)

Figure 1.3: The change of the membrane potential and the ion currents during the conduction of an AP. (The current is defined to be positive if it flows out of the cell.) At $t = 0$, the membrane is depolarized by a current impulse that increases the membrane potential with $+20$ mV. (a) The depolarization causes a quick increase of the membrane potential to about $+50$ mV. This is followed by a slower reduction phase where the lowest point is actually below the initial, resting potential (hyperpolarization). (b) The depolarization causes an increased permeability to sodium ions (Na^+) which is manifested by a pronounced influx (negative current) of these ions because the concentration and voltage gradients both asserts a push into the cell. Above a certain membrane potential, voltage-gated potassium channels open which increases the permeability to potassium ions (K^+). This is manifested by an efflux (positive current) of these ions because the concentration and voltage gradients both assert a push out of the cell.

the membrane occurs. Second, there will be an *influx* of sodium ions according to its equilibrium potential being *above* the resting potential. Third, there will be an *efflux* of potassium ions according to its equilibrium potential being *below* the resting potential.

To balance these last two effects, the membrane contains *ion pumps* of which the most important one in this context is the Na^+/K^+ pump. Each "stroke" forces three sodium ions *out* of the cell and two potassium ions *into* the cell. Hence, the ion concentrations and thereby the membrane potential are kept at constant levels.

1.1.2 The Hodgkin-Huxley action potential

With the knowledge about the role of sodium, but without knowing about the ion channels which were discovered much later, Hodgkin and Huxley began a series of experiments where they tested different conditions and measured the resulting membrane potential.

After a tremendous amount of experimental work and manual calculations, they were in 1952 able to present a mathematical model of the depolarization of the cell membrane. For this effort they received the Nobel prize in 1963 (shared with John C. Eccles).

Hodgkin and Huxley showed that upon depolarizing the cell membrane, the permeability to sodium suddenly increases which allows the voltage and concentration gradients to push sodium ions into the cell. The sodium influx increases the membrane potential in an effort to reach the equilibrium potential of the sodium ions (55 mV), see Figure. 1.3.

Before equilibrium is reached, however, the permeability to potassium increases which allows the voltage and concentration gradients to push potassium ions out of the cell. The potassium efflux counteracts the sodium influx which causes the membrane potential to peak around 50 mV and slowly decrease. Today, we know that these effects are excerted in *ion channels* that actively open and close according to a certain set of rules.

The depolarization also triggers the adjacent parts of the cell membrane, thus, causing it to spread and conduct along the fiber. The effect is very similar to when lining up dominos close to each other and tipping one of them over. The "depolarized" domino hits the neighboring domino and tips it over which causes the next one to do the same and so on until the last domino has fallen.

To be able to do this again, all dominos need to be erected and lined up. This is quite the same as in the nerve fiber where the ion channels and the ion concentration need to be restored before a new AP may be conducted, called *repolarization*.

The speed with which the AP propagate, the *conduction velocity*, is mainly dependent on the square root of the fiber diameter. In the case of the squid giant axon this means a conduction velocity of about 30 m/s; a quite remarkable value.

There is an evident trade-off here between transmission delay on one hand and "bulkiness" on the other. Since space is at premium for any organism, the enormous size of this axon betrays something about the *importance* of a short transmission delay in this case. A closer look would consequently disclose that this axon is part of the squid's jet-propulsion system; a quite reasonably function to prioritize considering the obvious drawbacks of becoming a prey.

1.1.3 Myelination

The propagation of the AP depends on the coordinated action of both the active current that flows through the voltage-dependent ion channels as well as the passive current that continuously flows through the cell membrane. As a consequence, the conduction velocity is determined by both these factors.

One way of improving the passive current flow is to increase the diameter of the axon because this reduces the internal resistance of the axon. The consequent increase in conduction velocity presumably explains why invertebrates such as the

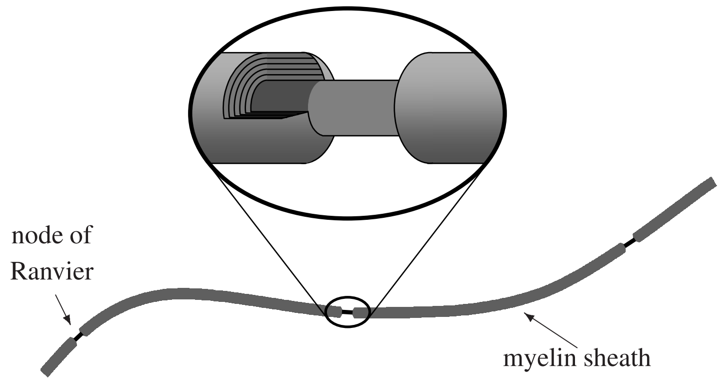


Figure 1.4: A myelinated nerve fiber where one Ranvier node is enlarged clearly showing the multiple layers of the myelin sheath around the cell membrane. At the nodes, the cell membrane is exposed to the extracellular medium which allows the AP to be regenerated.

squid has evolved giant axons.

Another way of improving the passive current flow is to insulate the axonal membrane because this effectively reduces the current leakage, thereby increasing the distance of passive current flow along the axon. In vertebrates, this strategy has resulted in *myelination* which is a more cost efficient solution to the delay/bulk trade-off than the increased-diameter approach.

With the myelination strategy, the time-consuming AP generation in an adjacent segment, denoted as *regeneration*, may be done at discrete points along the axon instead of more or less continuously. In principal, every regeneration takes a certain amount of time, so, by extending the distances between the regeneration points, the AP is forced to propagate in "jumps" and the conduction velocity is increased.

The regeneration takes place in between the insulated parts at the so-called *nodes of Ranvier* where the axon is exposed to extracellular medium, see Figure 1.4. The internode distance is a balance between conduction velocity and reliability; a longer distance yields increased velocity but decreased realiability of the regeneration. Typically, the nodes are separated by a distance corresponding to about a hundred axon diameters which results in conduction velocities ranging from 6 m/s to more than 130 m/s.[2]

Vertebrates thus have a rich variety of nerve fibers to choose from in order to find the best trade-off between transmission delay on one hand and the space occu-

[2] As a rule of thumb, the conduction velocity [m/s] is given by multiplying the fiber diameter [μm] with 6 [24, p. 72].

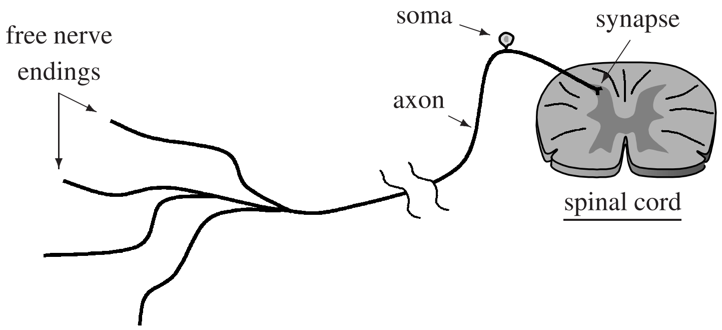


Figure 1.5: A nociceptor with the most important parts indicated.

pation on the other. What's best in the particular case depends on what information that is to be transmitted and the importance of a short transmission delay.

1.1.4 Sensory organs and receptors

Highly specialized nerve endings, called *receptors*, convert physical events into encoded messages that are passed on for further processing.

Using only five basic senses, somatic sensation (mechanical, thermal, or chemical stimuli), vision (photons), audition (sound waves), vestibular sensation (head movements), and chemical senses (taste/odour), we are able to get a good perception about our surrounding environment.

Presenting all these sensory systems would, however, be a daunting challenge. This presentation will therefore be limited to the somatic sensory system. In particular, only the receptors responsible for reporting temperature and painful stimuli will be discussed.

These receptors are called *nociceptors* (*noceo*, Latin: 'do harm') because they primarily detect noxious stimuli. Basically, the nociceptors are constituted by *free nerve endings*, see Figure 1.5, of either slowly conducting myelinated (Aδ) nerve fibers (about 20 m/s) or even slower unmyelinated (C) nerve fibers (less than 2 m/s).

Not surprisingly, the Aδ nociceptors respond to dangerously intense mechanical or mechanothermal stimuli that require an urgent response whereas, in general, the C nociceptors respond to less urgent sensations such as thermal, mechanical, or chemical stimuli.

An important discovery was the finding that nociception actually involves specialized neurons, not simply excessive discharges of the neurons that respond to normal stimulus intensities. Regardless of the discharge rate of non-nociceptors, the stimulus is normally not perceived as painful whereas, in contrast, already low discharge rates of a population of nociceptors actually is.

There are two characteristics of nociceptors that also make them different from other sensors, namely: *sensitization*, and *modulation*. Sensitization is caused by tissue damage and the release of various substances that bring normally silent nociceptors into a sensitized state. The resulting phenomenon is *hyperalgesia* when even light touch could be painful. This may, for example, be experienced after being exposed to the sun for too long.

The modulation of pain is not completely understood but it is clear that neurons in the spinal cord have the ability to inhibit the relaying of the nociceptor APs to higher neurons and ultimately our consciousness. An everyday example of this is the ability to reduce the sensation of sharp pain by activating mechano receptors. For example, by gently blowing towards an injured site, the pain is relieved.

The perhaps most intriguing finding in this latter respect was the finding of endogeneous opioids (the endorphins belong to this family) and opiate sensitive regions in both the central and the peripheral nervous system. It was now clear that pain may be relieved in two independent ways: disabling the nociceptors ("silencing the source") and/or down-modulating the transmission ("cutting the wire").

1.2 Recording C-fiber APs

In order to further increase the knowledge of the functional properties of the C nociceptors, it was necessary to obtain recordings of the emitted APs. A task that was not easily accomplished.

First, the fibers are bundled together into *fascicles* with a thick insulating connective tissue, the *perineurium*, surrounding it, see Figure 1.6. Second, the fascicles are bundled together with blood vessels and enwrapped in a connective tissue sheath, the *epineurium*. Third, using a thin needle electrode, it is indeed possible to penetrate the nerve and position the electrode in a fascicle. APs from a large number of nerve fibers are recorded, however, and individual studies of the APs originating from a particular nerve fiber is difficult.

1.2.1 The marking phenomenon

To overcome this problem, Hallin and Torebjörk introduced a method that shows the excitation of a C-fiber by utilizing the so-called *marking phenomenon*. The phenomenon stems from the slight decrease of a fiber's conduction velocity after an AP has been conducted. The conduction velocity then slowly returns to its initial value.

The principle of the method is to apply an electrical impulse repetitively, at a

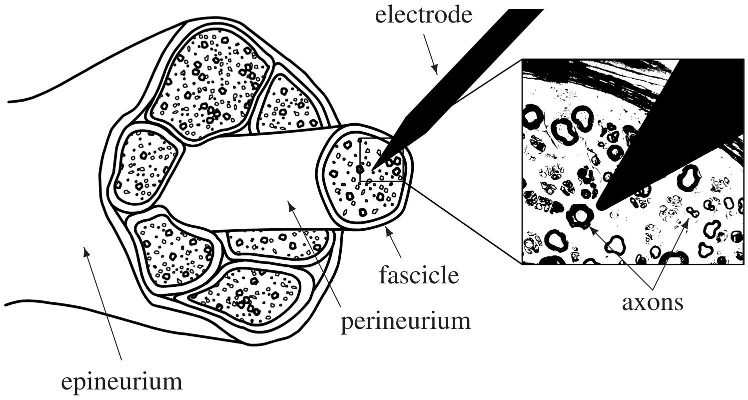


Figure 1.6: A needle electrode inserted into a nerve with the most important parts identified. As the enlargement shows, the electrode records from several fibers at the same time due to its large size compared to the nerve axons.

constant, low frequency (0.25 Hz), into the innervation territory of the C-fiber under study, see Figure 1.7. For each impulse, a single AP is evoked and appears in the recording after a certain latency. To document the response characteristics of the C-fiber, a physiological test stimulus (e.g., mechanical, temperature, or chemical) is applied into the receptive field of the fiber. If such a stimulus generates additional APs, the conduction velocity of the affected fiber decreases. The following APs excited by the repetitive stimuli thereby show a noticeable increase in latency.

This change in latency is used as a *marker* to indicate that the C-fiber responded to the applied physiological stimulus. In addition, the magnitude of the latency increase provides information about the number of APs that were generated by the test stimulus.

To enhance the efficiency of these experiments, a computer-supported recording system is used that both emits the repetitive stimuli and records the responses. Often, several fibers are activated and recorded simultaneously, but due to differences in conduction velocity of the individual C-fibers, the APs are spaced in time. Using the marking phenomenon, it is thus possible to discriminate and classify separate C-fibers by examining their characteristic latency responses.

1.2.2 Analyzing the signal

Previously, the analysis of the recorded traces was carried out manually which was very time consuming. To facilitate the analysis, a computer program that de-

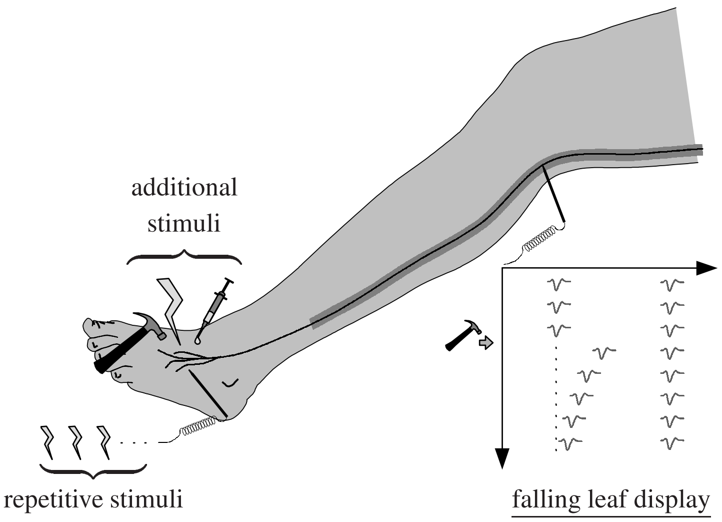


Figure 1.7: The recording setup. Repetitive, electrical impulses are delivered through a needle electrode inserted into the skin of the foot. Each triggered nerve fiber emits a single AP that is recorded at the knee via an electrode inserted into the corresponding nerve. In this case, there are two simultaneously active fibers which APs are recorded after a certain latency. The recorded APs for a particular triggering impulse are displayed from left to right in the falling leaf display. Successive responses are displayed in traces from top to bottom. To study the characteristics of a particular nerve fiber, an additional stimulus (e.g., mechanical, electrical, or chemical) is applied into the innervation area. The fibers responding to the additional stimulus are easily detected by their delayed response to the repetitive stimuli. In this case, the mechanical stimulus applied in between trace three and four triggered the first fiber as shown by its increased latency. Following this, the fiber recovers gradually as indicated by the APs returning to the latency prior to the activation (indicated by a dotted line). The right fiber did not respond to the mechanical stimulus and, hence, its latency was retained throughout the recording.

tects the APs, discriminates between APs originating from different C-fibers, and estimates latency shifts and recovery constants quantitatively was developed, see Figure 1.8. The most important aspects of the analysis are the detection and discrimination of the APs. Once this is accomplished, it is straight-forward to fit a parametrical model to the data in order to obtain the sought parameters.

(a) (b) (c)

Figure 1.8: Sample results from the three-step analysis algorithm: (a) the detected APs, (b) the resulting five tracks after tracking, and (c) the final trajectories obtained by fitting an exponential function to the tracks.

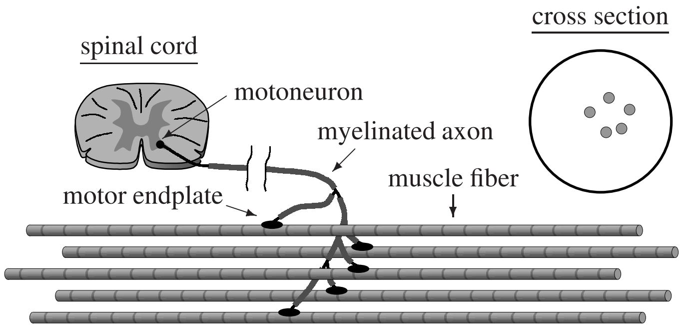


Figure 1.9: The motor unit consists of the motoneuron in the spinal cord, the myelinated axon, the motor endplate, and the innervated muscle fibers.

1.3 The constitution and function of muscles

As noted already by Galvani and Volta, a muscle contraction is initiated if the corresponding nerve is stimulated by an electrical impulse. Further experiments showed that the muscles actually shared many of the electrical characteristics of the nerves. It was discovered that the muscles too may be triggered by an electrical impulse and conduct an AP after being triggered. The latter was demonstrated by using a second frog leg as a detecting device and attaching its nerve to a muscle of the first leg.

1.3.1 The motor unit

Later anatomical studies also revealed that the muscles, similar to the nerves, are constituted by individual fibers where each fiber is a single cell. The muscle cell membranes also share most of the properties of the nerve cell membranes. The

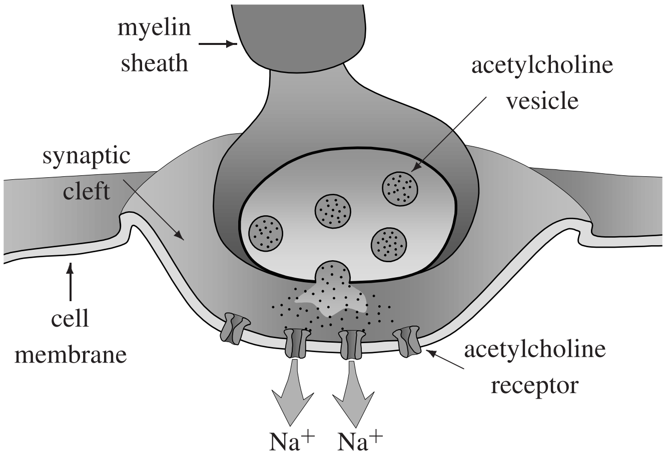


Figure 1.10: The neuromuscular junction, or motor endplate. When the AP arrives through the myelinated axon, it causes the vesicles to empty their acetylcholine molecules into the synaptic cleft. The neurotransmitter diffuses across the cleft and binds to the corresponding receptor. This activates the receptor and lets sodium ions (Na^+) pass through which depolarizes the cell membrane. If the depolarization is large enough, an AP is triggered.

results from the study of the squid giant axon are in general applicable to muscle fibers as well. To mention a few examples, the Hodgkin-Huxley model may be used to model muscle fiber APs and the conduction velocity of the muscle fibers increase with diameter as is the case for nerve fibers. In conclusion, what has been said above about the electrical characteristics of nerve fibers is in principal valid for muscle fibers as well.

The extension of the fibers differ, however. Muscle fibers are limited to the muscle whereas nerve fibers extends from a *motoneuron* in the spinal cord, via a peripheral nerve, all the way to the *innervated* muscle, see Figure 1.9.

The muscle is organized into functional groups of fibers each controlled by the same motoneuron. This constellation constitutes the smallest functional unit of the muscle and is called a *motor unit* (MU). With its axon, the motoneuron innervates its muscle fibers (typically a few hundred) via the *neuromuscular junction* (NMJ), called the *motor endplate*, one for each muscle fiber.

The different MUs are intermingeled with each other and have their muscle fibers randomly distributed. Thus, a particular fiber is in general surrounded by fibers belonging to other MUs.

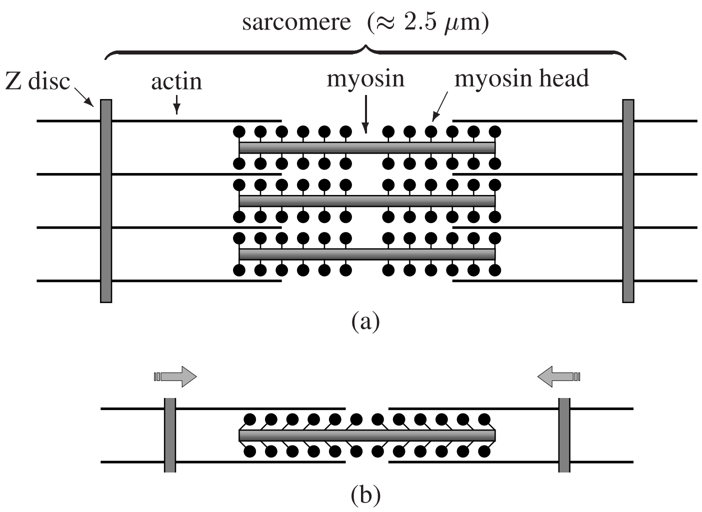


Figure 1.11: The principal constitution of the myofibrils that make up the bulk of the muscle fiber. (a) Each myofibril is made up of long chains of sarcomeres that are attached through the Z discs and consists of actin and myosin filaments. (b) During a contraction, the myosin heads adhere to the actin and undergo a conformational change that forces the Z discs together.

1.3.2 The motor endplate

In contrast to nerve and muscle fibers where the information is carried by electrical events, the APs, the motor endplate use chemical events, neurotransmitters, as information carriers.

Whenever an AP enters the synapse at the nerve terminal, the transmitter substance (acetylcholine) is released and diffuses across the *synaptic cleft* to receptors at the muscle fiber membrane, see Figure 1.10. When the transmitter binds to the receptors, sodium channels open and the cell membrane is depolarized, thereby initiating an AP.

Because of the construction of the motor endplate, there is a delay ($\approx$ 0.5 ms) associated with the arrival of the nerve impulse at the terminal and its further excitation of the muscle fiber. Moreover, this delay has a certain variability, the *jitter*, which is stochastic (standard deviation about 5-40 μs) and may change with disease. Hence, measuring the jitter is a valuable tool when studying the functional properties of the MU.

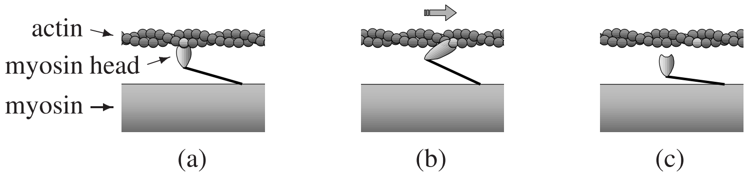


Figure 1.12: A detailed view of a contraction according to the sliding filament theory. (a) The myosin head bind to the actin. (b) The myosin undergoes a change in shape which forces the actin to the right (the power stroke). (c) Through an energy consuming process, the head is released from the actin and its shape is restored.

1.3.3 Muscle contraction

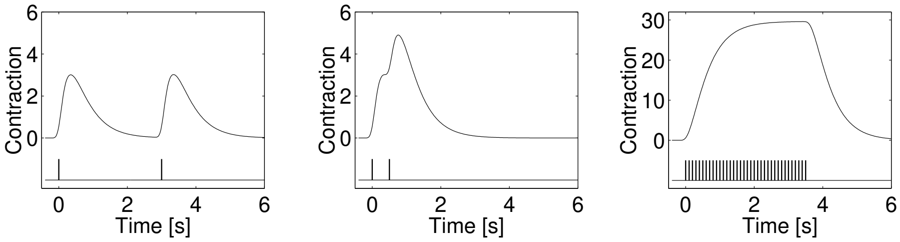
The depolarization of the muscle cell membrane initiates mechanical changes in the muscle fiber that makes it shorter through a minuscule machinery. The bulk of each muscle fiber constists of bundled *myofibrils* that are made up of *myosin* and *actin* filaments, see Figure 1.11. The filaments are organized into *sarcomeres*, separated by the *Z discs*, and work like a rack where the heads on the myosin filaments act as cogs that cling on to the actin filaments.

The conduction of the AP causes calcium (Ca^{2+}) to be released which, according to the sliding filament model, results in the following steps, see Figure 1.12:

1. The heads of the myosin bind to the actin filaments.
2. The heads are bended which causes a contraction by sliding the filaments past each other, the *power stroke*.
3. By consuming energy, the heads are detached and straightened.
4. The process is repeated as long as there are calcium and energy available.

Whenever the membrane potential returns to its resting level, the release of calcium ceases and the free calcium is quickly removed by efficient ion pumps. If this does not work for some reason, the muscle will be "stiff" and unable to relax. *Rigor Mortis* is the extreme of this situation. The free calcium causes the myosin to cling to the actin and contract but is unable to detach due to lack of energy. This condition remains until the filaments disintegrates in the decomposition process.

Due to the construction of the contraction machinery, a single AP causes a *twitch* in the muscle fibers of the corresponding MU that consists of a quick increase in muscle force followed by a somewhat slower return to zero, see Figure 1.13. By emitting a series of APs, the twitches summate to a smooth continuous contraction producing a higher and more sustained force. By activating other MUs as well, the muscle force may be increased even more.



(a) (b) (c)

Figure 1.13: Each AP causes a quick increase in force excerted by the fibers within the MU followed by a slow relaxation. These twitches integrate into a contraction that is proportional to the triggering frequency. The diagrams show the contraction (arbitrary units) when (a) the APs are three seconds apart, (b) the APs are 0.5 s (2 Hz) apart, and (c) the APs are 0.1 s (10 Hz) apart. The times of arrival of the APs are indicated by vertical bars.

In each muscle, there are 100-500 MUs that are working in parallell to provide the correct muscle force. They are independently controlled by their individual motoneurons through frequency modulation and increasing muscle force is obtained by increasing the stimulation frequency of already active MUs and by recruitment of new ones.

1.4 The line source model

From the first evident demonstration of the electrical activity, the *electromyogram* (EMG), in contracting muscles, it has been clear that a good understanding of the underlying processes is vital in order to interpret the EMG correctly. Electromyography, in interplay with various anatomical techniques, has provided much of the present knowledge of the structural organization and the nervous control of muscle.

A vital tool in gaining this knowledge has been, and still is, modeling and simulation. A simple and yet reasonably accurate model is the *line source model* that is obtained by considering the AP as a convolution of a weighting function and a transmembrane current lumped to the center of the muscle fiber. The weighting function depends on the used recording electrode whereas the transmembrane current depends on the particular muscle fiber. The resulting EMG signal is then simply the sum of the contributions from all muscle fibers in the recorded muscle.

With the increased availability of digital computers, conducting simulations has been simplified considerably. Computerized simulations require, however, discrete-time models where the time axis consists of discrete points. This is different from the real world where the time is continuous

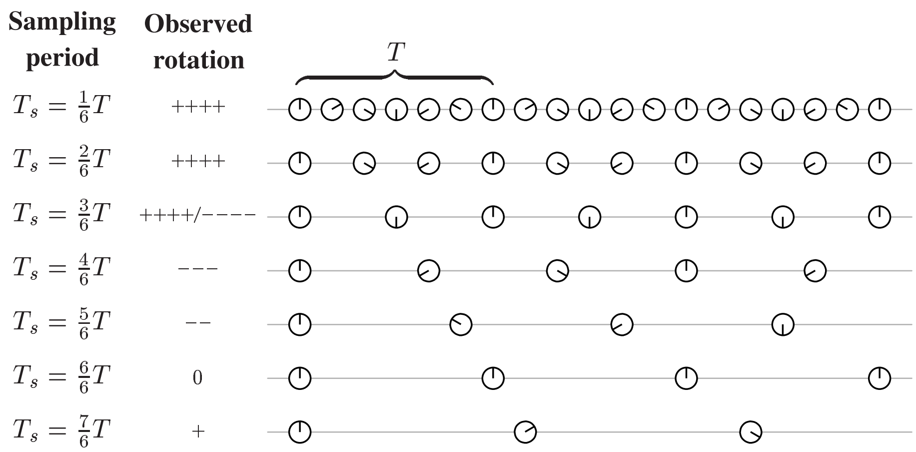


Figure 1.14: The importance of selecting an appropriate sampling period to avoid aliasing is illustrated using a rotating wheel. Unless the sampling period is short enough with respect to the period T, in which the wheel completes one full revolution, the samples constitute a "skewed" view of the continuous-time reality. In the figure, the observed direction of the rotation is indicated by '$+$' for clockwise (cw) rotation and '$-$' for counter-clockwise (ccw) rotation, respectively. The speed is symbolized by the number of repititions of the direction indicator ('0' for stationary rotation). As the figure shows, the correct direction and speed of rotation may be observed for sampling periods $T_s < \frac{1}{2}T$. For $T_s = \frac{1}{2}T$, the correct speed may indeed be observed, but the direction is ambiguous. Using longer sampling periods, the rotation observed in the samples no longer reflects the original rotation. Denoting the original rotating speed as very fast cw*, the observed rotation pass through, in sequence,* fast ccw*,* ccw*,* stationary*, and* slowly cw *as the sampling period increases ($T_s > \frac{1}{2}T$). In these cases, the observed rotation is an "alias" of the original rotation; hence, the term* aliasing *for this type of distorsion.*

1.4.1 Sampling and aliasing

The transformation, or *discretization*, from continuous time to discrete time is often referred to as *sampling* because the continuous-time model is sampled (measured) at discrete points in time. It is important that this is done sufficiently often to also catch the fastest changing characteristics. If not, the discrete-time representation become distorted through what is called *aliasing*. A good example of aliasing is in movie sequences with accelerating cars where the wheels (preferrable with spokes) are visible. First, the wheels (correctly) look like they are rotating faster and faster. Then, the rotation (falsely) looks like it rotates backwards or even stops. In this case, the sampling frequency (frame rate) is too low to catch the characteristics of

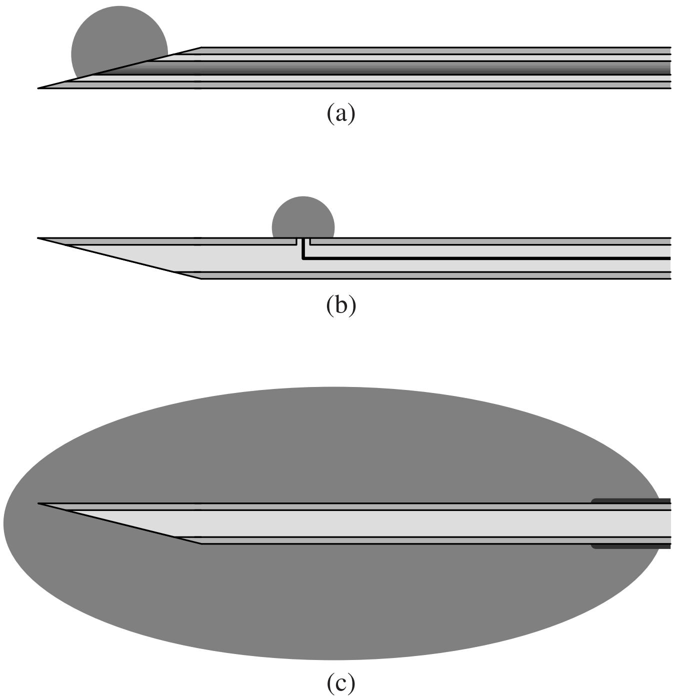


Figure 1.15: The three most common electrode types with their pick-up distance indicated: (a) the CN electrode, (b) the SF electrode, and (c) the Macro electrode.

the quickly rotating wheels.

Figure 1.14 shows this phenomenon from a little different perspective. Here, the rotational speed is constant but the sampling frequency is changed. However, the same observations as in the example above are illustrated.

Addressing the aliasing in the discretization process is an important issue. It can be done in two ways: either the discretization frequency is selected high enough with respect to the fastest changing characteristics, or these characterics are removed by prefiltering before the actual discretization. In the example above with the accelerating car, these alternatives correspond to either increasing the frame rate of the camera, or covering the wheels using smooth hub caps prior to the shooting.

1.4.2 Electrode types

The EMG signal may be acquired using either surface electrodes that are attached on the skin surface or, needle electrodes that are inserted transcutaneously into the muscle. In this thesis, only the three most common needle electrodes are regarded, see Figure 1.15: the *concentric needle* (CN) electrode, the *single fiber* (SF) elec-

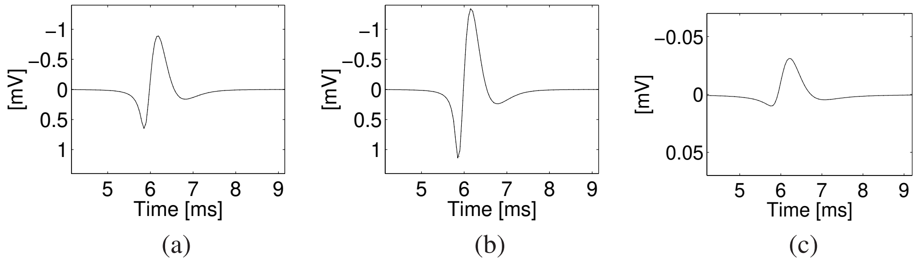


Figure 1.16: Examples of three APs as recorded with (a) the CN electrode, (b) the SF electrode, and (c) the Macro electrode, respectively. Note the different scale of the Macro AP.

trode, and the *Macro* electrode.

The CN electrode is a standard electrode used in clinical routine throughout the world. It has an elliptical recording area (150×580 μm) in the beveled tip of a steel cannula (diameter 0.45 mm). Often, the cannula is used as a reference electrode to obtain bipolar recordings. The electrode sums the electrical field from a number of muscle fibers in a given MU to obtain a *motor unit action potential* (MUAP). The shape of the MUAP reflects the number of muscle fibers, the synchronicity of their APs, and their concentration within the pick-up distance.

By using different electrodes, different resolutions can be obtained. For example, it is possible to record APs from individual muscle fibers using the *single fiber* (SF) electrode [86] [28]. This gives the possibility to study in detail some of the physiological characteristics involved in the generation of the EMG signal. Similarly, it is possible to record APs from entire MUs through the *Macro* electrode [79]. This electrode provides a broader view that, in some cases, is more suitable for diagnosis [71].

The SF electrode is constructed to selectively record APs from individual muscle fibers. The recording surface is a wire (diameter 25 μm) exposed at a side port of the electrode with the cannula as a reference. It is used to assess fiber density (FD) and to measure the jitter in the motor endplate.

The Macro electrode is the opposite of the SF electrode in that it records APs from the entire MU. Its recording area is the exposed tip (length 15 mm) of an insulated cannula with a distant electrode as a reference. This results in a broader view that may be used to estimate the total size of the MU and is in some cases more suitable for diagnosis than what is provided by the CN electrode.

1.4.3 Simulated APs

Using the line source model, fast and accurate simulations of different electrodes, muscle fibers, and electrode positions are readily obtained. As an example of this, Figure 1.16 shows three different APs as if recorded by the three most common electrode types. Note that the y axis increases downwards in line with the convention of displaying EMG signals.

The diameter of the simulated fiber was selected to 55 μm corresponding to a conduction velocity of 3.7 m/s. The electrodes were oriented perpendicular to the muscle fiber where the SF electrode and the CN electrode were positioned at a radial distance (center-to-center) of 100 μm.

The Macro electrode was positioned close to the fiber at a radial distance of 255 μm (a little more than the radius of the cannula plus the radius of the fiber).

1.5 The compound AP of the MU

By combining the line source model with knowledge about the constitution of the MU, compound APs of the MU is readily obtained.

First a MU is generated with a certain number of fibers having certain diameters and being distributed over a certain area, etc, in accordance with anatomical knowledge. The number of fibers present in a particular area is referred to as the *mean fiber concentration* (MFC).

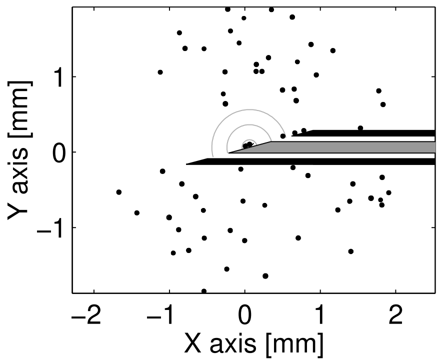
Then, the insertion of the selected recording electrode is imagined. The "shuffling effect" of the beveled tip is simulated by moving the affected fibers to the upper side of the electrode, see Figure 1.17 (a).

Finally, the line source model is applied to each muscle fiber, taking the relative electrode position into account. The resulting individual APs are summed to obtain the compound AP of the MU for the particular electrode selected, see Figure 1.17 (b).

1.5.1 Changes in disease

The constitution and function of the MU may change in disease. The exact effect is of course individual for each disease depending on which category it represents; myopathies, neuropathies, or diseases in the neuromuscular junction.

Myopathies cause a loss or impairment of muscle fibers. In general the number of fibers within the MU decrease, but fibers may actually split longitudinally or regenerate in a compensatory process to meet the loss of fibers. Moreover, the fiber diameter variation is often found to be increased, with abnormally small and large muscle fibers.



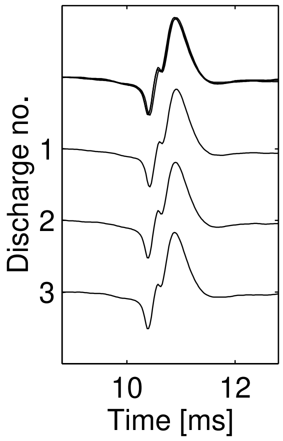


Figure 1.17: Example of a simulated MUAP obtained with a concentric needle for a normal healthy muscle (MFC = 5 fibers/mm^2). (a) A cross section of the MU where the fibers and the electrode are shown. The semi circles indicate the distances 100, 300, and 500 μm, respectively. (b) The simulated MUAP for three discharges. On top, all discharges are drawn superimposed.

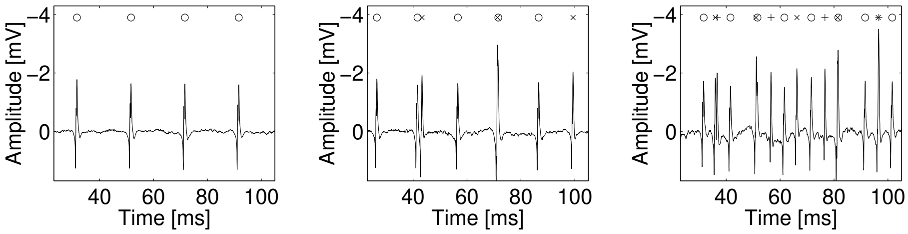
Neuropathies cause a loss of motoneurons or axons. In a compensatory process, surviving axons branch off and reinnervate the orphaned muscle fibers by *collateral sprouting*. Hence, the number of MUs decreases but their size in terms of number of fibers increases. Other findings may be a change in diameter of individual fibers and a temporarily increased jitter of recently reinnervated fibers.

Junctional diseases affect various key components that are vital for the function of the MU. If, e.g., the acetylcholine receptors in the motor endplate are reduced in number, the function of the muscle may be severely impaired. This may cause the variability of the synaptic delay, the jitter, to increase, or the triggering to sometimes fail; a condition called *blocking*.

As a rule of thumb, myopathies are manifested by small MUAPs and often the components of the MUAPs are "spread out" in time. Neuropathies are manifested by large MUAPs with multiple peaks corresponding to the increased number of fibers. The junctional diseases, finally, may often be diagnosed by the differences in the MUAP over time through studies of, e.g., the jitter and/or the blocking.

1.6 EMG analysis

Electromyography is today a common and efficient diagnostic tool that allows the clinician to follow changes in the neuromuscular system caused by disease pro-



(a) (b) (c)

Figure 1.18: Intermingled MUAPs at various force levels with the firing frequency exaggerated for clarity. The number of recruited MUs in each diagram is (a) one, (b) two, and (c) three. The position of the peak is marked ($\circ$ $\times$ $+$) for each individual MUAP.

cesses.

The measured EMG signal is the summed contribution from all active muscle fibers within the entire muscle. Depending on the electrode, distant fibers contribute in a varying extent. The SF electrode, for example, effectively attenuates the contribution from distant fibers owing to the arrangement of the recording surface positioned within the reference.

The EMG signal is made up of the compound APs from each MU, see Figure 1.18. Because the muscle force is regulated by frequency modulation and recruitment/decruitment of MUs, the number of active MUs and their repetition frequency is varying with force level. Thus information regarding multiple MUs can be retrieved from a single EMG signal.

By *decomposing* the EMG signal and separating compound APs originating from different MUs, the analysis is often simplified. Each compound AP may then be analyzed separately which may include an effort to gain knowledge about the MU's constitution (e.g., through the MFC) or functional properties (e.g., through the jitter).

1.6.1 Analyzing the MUAPs

If possible, a complete decomposition of the MUAP into its constituent muscle fiber APs would add considerable information about normal and diseased MUs. The number of fibers within the pick-up distance of the electrode, for example, would be directly given by the number of AP components. Moreover, the jitter would be readily estimated through the variability in timing of the individual APs in subsequent discharges.

Deriving such an algorithm would be a daunting challenge, however. Perhaps is it even impossible unless sophisticated multichannel electrodes are adopted.

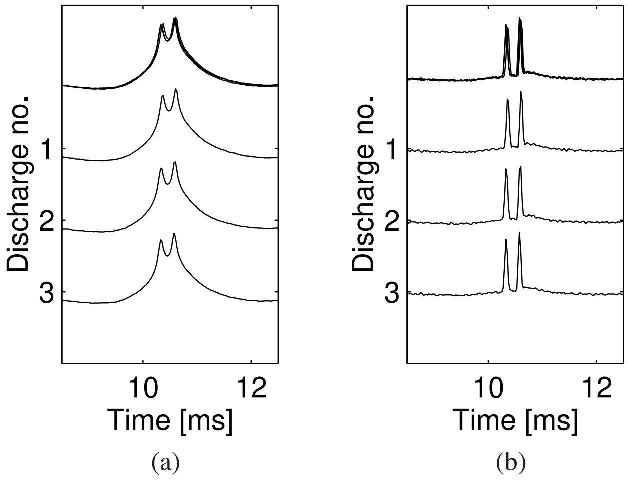


Figure 1.19: Example of applying the deconvoluting prefilters to three discharges of the MUAP in Figure 1.17 (b). (a) The partially deconvolved signal where each individual AP is monophasic. The diagram shows two active APs within the pick-up distance. (b) The fully deconvolved signal where each AP is transformed to a narrow impulse. In this signal, the two active APs are clearly visible. On top, all discharges are drawn superimposed.

Instead, a simpler approach is sought. To discriminate between myopathies and neuropathies, for example, it is sufficient to know the MFC which is directly proportional to the number of fibers within a certain radius (e.g., the pick-up distance).

A scaled version of this number works just as well and may be retrieved easily by assuring a direct relation between the number of fibers and some measurable quality of the analyzed signal. For example, knowing the amplitude of the compound AP would be sufficient in order to discriminate between myopathies and neuropathies if it was directly proportional to the number of fibers within the pick-up distance.

Normally, however, this is not the case because APs have positive and negative phases that make the sum unpredictable. This is called phase *canceling*. By prefiltering the compound AP with a filter that removes the multiphasic shape, a signal more suitable for diagnosis may be obtained, see Figure 1.19 (a). This signal is called the *partially deconvolved* MUAP because it removes the multiphasic shape only and keeps the “smearing” effect.

In the general case, the amplitude of the prefiltered compound AP is, however, not directly proportional to the number of fibers within the pick-up distance because the individual APs are distributed in time. Hence, the number of fibers may be increased without affecting the amplitude.

A better measure in this respect might be the area because the area is always adding up regardless of the position in time. This requires that the signal being measured is monophasic.

To measure the jitter, there are two things to consider: the individual APs must be distinguishable, and neighboring APs must not disturb the localization. Both these requirements are met if the APs could be transformed into narrow impulses. Again, using a suitable prefilter this could be accomplished, see Figure 1.19 (b). This signal is called the *fully deconvolved* MUAP because it seeks to remove all the shaping of the APs while it keeps the amplitude and timing information.

1.7 Further reading

To obtain more information on many of the issues presented above the books listed below are good starting points.

- D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, and J. O. McNamara, *Neuroscience*, Sunderland, MA: Sinauer Associates, Inc., 1997

- E. R. Kandel and J. H. Schwartz, *Principles of Neural Science*, New York, NY: Elsevier North Holland, Inc., 1981

- S. Deutsch and A. Deutsch, *Understanding the Nervous System: An Engineering Perspective*, New York, NY: IEEE Press, 1993

- B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson *Molecular Biology of the Cell*, second edition, New York, NY: Garland Publishing, Inc., 1989

1.8 Objective of this thesis

The present thesis provides our perspective of the signal processing challenges arising in the neurophysiological field. The overall objective of this work is to develop methods that extract information from neural and muscular signals, not available by existing methods.

In particular, we address three principal categories within this field and refer to these as: *modeling*; *prefiltering and parameter assessment*; and *data detection and classification*. These are further presented below.

1.8.1 Modeling

Without doubt, the key tool in neurophysiology is the acquisition of signals and events that are used for diagnosis and research. A basic requirement is here the knowledge of the relationships that connect a certain signal property with a certain neurophysiological condition. As in virtually any other area, one efficient way in gaining this knowledge is by means of experiments followed by modeling and simulation.

The most important model presented herein is the modified line source model that models the recording of APs from a muscle fiber. The model has been implemented in a simulation program [83] that may contribute to new insights into the underlying processes of the generation of the EMG signal. This may in the longer perspective lead to improved diagnostic methods.

In a shorter perspective, this model can be used for research and educational purposes since it enables fast and accurate simulations of different anatomical conditions. Within a few seconds, the corresponding APs may be generated and visualized. Furthermore, benchmarks on different analysis algorithms may be carried out.

1.8.2 Prefiltering and parameter assessment

Within the daily routine, in particular, continual efforts are being made to make the examinations more efficient and to improve the reliability in diagnosis. Most of these efforts concern algorithms that are characterized by prefiltering (often using simple filters) followed by an assessment of the parameters to be used for diagnosis.

Based on a Wiener filter design, we present a more elaborate prefiltering method that strives at optimizing the filtered signal for the subsequent parameter assessment algorithm. Depending on the situation, the optimal prefilter will be different.

The current results from this approach indicate several advantages. First, the variability of the resulting parameter estimates decrease because the parameters are assessed using a signal that is more suitable for the task. Second, the need for different electrodes for examinations of various types may be reduced by applying specifically designed prefilters to enhance different features in the signal. Third, the Wiener filter design method provides a good intuitive coupling between the design variables and the properties of the resulting filter, which simplifies the tuning process.

1.8.3 Data detection and classification

When studying, e.g., the membrane properties of nerve C-fibers, APs originating from a specific C-fiber must be detected and identified. With the special recording technique being used, the measured parameters form clusters in the parameter space. These clusters have (possibly) time-variant centroids, which effectively disqualify most clustering algorithms.

By considering the problem from a target tracking point of view, however, the APs originating from a particular C-fiber are tracked in subsequent responses to electrical stimulation. With this approach, an algorithm has been designed that has made the previously time-consuming, manual analysis much more efficient [57] [44].

This algorithm is a general implementation of a target tracker; only the tracking parameters and the predictor are coupled with the particular problem. Thus, this approach may be applied to virtually any problem where events occurring in multiple detections need to be classified.

1.9 Outline of the thesis

This thesis is divided into two major parts, *Methods* and *Applications*, respectively, of which the latter constitutes the main contribution.

Part I consists of Chapter 2-5 and is essentially a recapitulation of the most important methods used in the applications. These chapters have a disposition governed by the applications part, but are otherwise self-contained and may be read independently of each other.

In Chapter 2 an asynchronous matched filter (MF) detector needed for the application described in Chapter 9 is derived and analyzed. Because the assessment of important performance properties are crucial for the tuning of the algorithms in the applications chapter, this chapter provides the required framework that is then examplified on a hypothetical test signal.

Chapter 3 presents how a Kalman filter based on a continuous-time model is designed and tuned. Kalman filters are used by the multiple hypothesis tracking (MHT) algorithm, c. f. Chapter 4, where they provide predictions and assist in the track evaluation. Because it is important with good estimates in this regard, this chapter presents a consistency analysis procedure as well as an minimum mean squared error estimate of the initial values.

Chapter 4 provides a general description of multiple target tracking. Some common tracking algorithms are mentioned where the multiple hypothesis tracking (MHT) method receives most attention because it is used in the applications of Chapter 8 and Chapter 9.

Deconvolution using Wiener filters is the topic of Chapter 5 and this method is later used in Chapter 8 to filter and refine EMG signals. The implementation selected for this thesis is presented as well as an illustrative example.

The first chapter of Part II, Chapter 6, describes a modified line source model that can be used to simulate APs fast and accurately. The importance of avoiding aliasing when discretizing the model is stressed and the consequences of not doing so are shown through simulations. Moreover, through a simple transformation of the electrode specific weighting functions, a finite muscle fiber length is easily incorporated into the model.

In Chapter 7 represents a brief introduction to simulation of entire MUs. The anatomical and physiological assumptions and parameters used in this thesis are presented. Moreover, the changes in the MU induced by disease are mentioned and exemplified with cross sections and resulting compound APs of simulated MUs.

Chapter 8 explores the possibility to improve the assessment of muscle fiber concentration (MFC) and jitter by combining a MUAP obtained with a CN electrode with Wiener filter deconvolution, c. f. Chapter 5. The performance is compared both to a method currently used in clinical routine as well as to a recently proposed method for improving the jitter estimations. To discriminate between individual APs that originate from different muscle fibers, the MHT/Kalman tracking algorithm is used, c. f. Chapter 3 and Chapter 4.

Detection and classification of nerve APs is described in Chapter 9. The classification is implemented as a target tracking problem where the MHT/Kalman tracking method is used, c. f. Chapter 3 and Chapter 4. Examples on data obtained from real recordings are presented.

Chapter 10, finally, ends this thesis with some concluding remarks and presentation of possible work in the future.

1.9.1 Contributions of the author[3]

Chapter 9 was presented at

> Björn Hansson, Clemens Forster, and Erik Torebjörk, "Matched Filtering and Multiple Hypothesis Tracking Applied to C-fiber Action Potentials Recorded in Human Nerves," in *Proceedings of the SPIE Conference (AeroSense '98), Signal and Data Processing of Small Targets*, volume 3373, Orlando, FL, September 1998, pp. 582–593.

It is also described in

> Björn Hammarberg (Hansson), Clemens Forster, and Erik Torebjörk, "Parameter Estimation of Human Nerve C-Fibers using Matched Filtering and Multiple Hypothesis Tracking," *IEEE Transactions on Biomedical Engineering*, volume 49, no. 4, April 2002.

[3]When studying the author lists, it might be practical to know that, during the course of this work, the author got married and changed his last name from Hansson to Hammarberg.

The muscle fiber model in Chapter 6 is described in

> Björn Hammarberg (Hansson) and Erik Stålberg, "Insights into the Line Source Model for Improved Muscle Action Potential Modeling," submitted to *IEEE Transactions on Biomedical Engineering*.

The motor unit model of Chapter 7 in combination with the line source model is described in

> Lars Karlsson, Björn Hammarberg, and Erik Stålberg, "A Muscle Model to Study Electromyographic Signals," submitted to *Computer Methods and Programs in Biomedicine*.

The work in Chapter 8 will be presented in

> Björn Hammarberg, Mikael Sternad and Erik Stålberg, "A Wiener Filter Approach to Electromyography," in preparation.

In addition to the scientific work listed above, the author has contributed to the following papers that are connected to this thesis to a various degree:

- C. Weidner, M. Schmelz, R. Schmidt, B. Hansson, H. O. Handwerker, and H. E. Torebjörk, "Functional Attributes Discriminating Mechano-Insensitive and Mechano-Responsive C Nociceptors in Human Skin," *the Journal of Neuroscience*, volume 19, no. 22, pp. 10184–10190, November 1999.

- A. Sandberg, B. Hansson, and E. Stålberg, "Comparison between concentric needle EMG and macro EMG in patients with a history of polio," *Clinical Neurophysiology*, volume 110, no. 11, pp. 1900–1908, November 1999.

- C. Weidner, M. Schmelz, R. Schmidt, B. Hammarberg, K. Ørstavik, M. Hilliges, H. E. Torebjörk, and H. O. Handwerker, "Neural Signal Processing – the Underestimated Contribution of Peripheral Human C-fibers," submitted to *the Journal of Neuroscience*.

1.9.2 Financial support

We gratefully acknowledge the financial support obtained from the Swedish Medical Research Counsil (ES, 135) and (ET, 5206), the Deutsche Forschungsgemeinschaft (SFB 353), and a Max Planck Research Award to Prof. Erik Torebjörk.

Part I

Methods

CHAPTER 2

Matched filter detection

DETECTING signals hidden in high levels of noise is a delicate task. If the signal is a member of a set of signals with known shapes and if the color of the noise is known, then *matched filtering* constitutes a standard signal processing technique for optimally enhancing and detecting the signal [19].

In this chapter, a discrete-time matched filter (MF) detector will be derived and analyzed from a general point of view. The task of the MF detector is to analyze the given signal and to *report* any possible occurences of the sought signal to some post-processing algorithm. Discussion of application specific issues is postponed to Chapter 9 where matched filtering is used to detect action potentials recorded in a human nerve.

First, a synchronous MF detector (SMFD) is derived and analyzed from a theoretical perspective. An asynchronous detector which is needed for the application in Chapter 9 is then derived.

Abandoning the assumption of synchronous detection affects the performance expressions, however, and a thorough theoretical analysis is not tractable. Instead, the performance of the asynchronous MF detector (AMFD) is analyzed through *Monte Carlo simulations*.

2.1 Derivation

The expressions for the synchronous MF detector, derived next, form a foundation to the succeeding analysis of the asynchronous MF detector.

2.1.1 A synchronous MF detector

The detection problem can be described by the following question: Is there a signal of known *shape* present in the recorded data? Under certain idealized conditions, the answer is given by the MF detector. The derivation of the SMFD in this section is based on the following key assumptions:

Assumption 2.1 *Finite duration – The signal has a finite length or duration.*

Assumption 2.2 *Known signal shape – The signal amplitude is unknown but its shape is known and is described by a signal template.*

Assumption 2.3 *Single signal – There is* at most *one signal in the recorded data.*

Assumption 2.4 *Known arrival time – The arrival time of the signal is known and the detector can test for the signal's presence at the most favorable time instant.*

Assumption 2.5 *Stationary, additive noise – In addition to the possible signal, the data contains stationary, additive, zero-mean noise that represents anything that cannot be described by a scaled signal template.*

Assumption 2.6 *Synchronous sampling – If the data has its origin in continuous time, the sampling is synchronized with the signal so that the sampled signal is equal in shape for all occurences.*

With the assumptions above, the data may be described by a sum of a signal component and a noise component. Denoting the recorded discrete-time data by $\mathbf{z}(n)$, we may write

$$\mathbf{z}(n) \triangleq \mathbf{s}(n) + \boldsymbol{\eta}(n) \tag{2.1}$$

where $\mathbf{z}(n)$, $\mathbf{s}(n)$, and $\boldsymbol{\eta}(n)$ constitute samples of the data, the signal, and the noise, respectively. They are all p-dimensional vectors in line with Assumption 2.1 and are defined as

$$\mathbf{z}(n) \triangleq \begin{pmatrix} z(n) & z(n-1) & \dots & z(n-p+1) \end{pmatrix}^T \tag{2.2}$$

$$\mathbf{s}(n) \triangleq \begin{pmatrix} s(n) & s(n-1) & \dots & s(n-p+1) \end{pmatrix}^T \tag{2.3}$$

$$\boldsymbol{\eta}(n) \triangleq \begin{pmatrix} \eta(n) & \eta(n-1) & \dots & \eta(n-p+1) \end{pmatrix}^T \tag{2.4}$$

where ^{T} denotes the vector transpose. The noise $\boldsymbol{\eta}(n)$ is assumed to be stationary and zero mean according to Assumption 2.5.

A p-dimensional normalized *template* $\mathbf{s}$ and an *amplitude* μ are introduced to describe a single signal present in the recording at a particular *arrival time* n_a in line with Assumption 2.1-2.4

$$\mathbf{s}(n_a) \stackrel{\triangle}{=} \mu\mathbf{s} \quad . \tag{2.5}$$

This template is valid for all arrival times under Assumption 2.2 and, in addition, for all sampled continuous-time signals under Assumption 2.6.

The objective is to determine whether the desired signal is present or not in the snapshot $\mathbf{z}(n_a)$ which may be described by either of the two hypotheses: H_0 or H_1. Under hypothesis H_0, the data is assumed to contain noise only ($\mu = 0$), whereas under hypothesis H_1, the data is assumed to contain exactly one signal as well as the noise ($\mu > 0$). This is summarized by

$$H_0 \quad : \quad \mu = 0 \tag{2.6}$$

$$H_1 \quad : \quad \mu > 0 \tag{2.7}$$

where the actual signal detection is then equivalent to determining which hypothesis that applies to the recorded data.

From the definition in (2.5), it follows that all signal energy has been received in $\mathbf{z}(n)$ at the arrival time n_a. Hence, all information about the presence of the signal $\mu\mathbf{s}$ is then available. This information is disturbed, however, by the noise $\boldsymbol{\eta}(n)$. If it were possible to attenuate the noise and/or amplify the signal, the detection situation would be improved.

The MF approach addresses this problem by considering the distribution of the signal energy in time and preprocessing the measurement vector $\mathbf{z}(n)$ using a linear filter prior to the actual detection. If we introduce the filter impulse response

$$\mathbf{h} \stackrel{\triangle}{=} \begin{pmatrix} h(0) & h(1) & \ldots & h(p-1) \end{pmatrix}^T \quad , \tag{2.8}$$

the filter output is

$$m(n) = \sum_{i=0}^{p-1} h(i)z(n-i) = \mathbf{h}^T\mathbf{z}(n) \quad . \tag{2.9}$$

As shown in [72], the filter output $m(n_a)$ is, under the assumptions stated above, a *sufficient statistic* for μ, i.e., it contains all information necessary to optimally deciding whether the signal is present ($\mu > 0$) or not ($\mu = 0$). The decision that the prescribed signal template $\mathbf{s}$ is present is then made whenever $m(n_a)$ exceeds a given threshold level m_0. With a properly designed filter, this threshold test is *uniformly most powerful* for testing H_0 versus H_1 and the corresponding filter design is equivalent with maximizing the signal-to-noise ratio (SNR_{mf}) at the time n_a [19]. The resulting optimal impulse response $\mathbf{h}_o$ is then said to consitute a *matched filter*.

Both $\mathbf{h}_o$ and the resulting SNR_{mf} may be calculated in terms of the known signal template vector $\mathbf{s}$ and the (assumed known) $p|p$ covariance matrix $\mathbf{R}_{\boldsymbol{\eta\eta}}$ of the noise $\boldsymbol{\eta}(n)$ according to [19] [72]

$$\mathbf{h}_o = \frac{\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}} \tag{2.10}$$

$$SNR_{mf} = \mu^2\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s} \ . \tag{2.11}$$

For the special case where the noise is white ($\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1} = (\sigma^2\mathbf{I})^{-1} = \frac{1}{\sigma^2}\mathbf{I}$), the expressions (2.10) and (2.11) are simplified to

$$\mathbf{h}_o = \frac{\mathbf{s}}{\sqrt{\sigma^2\mathbf{s}^T\mathbf{s}}} \tag{2.12}$$

$$SNR_{mf} = \frac{\mu^2}{\sigma^2}\mathbf{s}^T\mathbf{s} \ . \tag{2.13}$$

Combining (2.5) and (2.9)-(2.11), we obtain some interesting properties of this matched filter, namely

$$E(m(n_a)) = \frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}}E(\mathbf{z}(n_a)) \tag{2.14}$$

$$\stackrel{H_0}{=} \frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}}E(\boldsymbol{\eta}(n_a)) = 0 \tag{2.15}$$

while

$$\begin{aligned} E(m(n_a)) &\stackrel{H_1}{=} \frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}}E(\mathbf{s}(n_a)+\boldsymbol{\eta}(n_a)) \\ &= \frac{\mu\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}} = \sqrt{SNR_{mf}} \propto \mu \end{aligned} \tag{2.16}$$

where $E(\cdot)$ is the expectation with respect to $\boldsymbol{\eta}(n)$. These relations will be utilized in Chapter 9 to estimate the amplitude. The variance of $m(n_a)$ is identical under the two hypotheses as shown by

$$V(m(n_a)) \triangleq E\left(m(n_a) - E(m(n_a))\right)^2 \tag{2.17}$$

$$= E\left(\frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\boldsymbol{\eta}(n_a)}{\sqrt{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}}\right)^2$$

$$= \frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}E(\boldsymbol{\eta}(n_a)\boldsymbol{\eta}^T(n_a))\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}} = \frac{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}}{\mathbf{s}^T\mathbf{R}_{\boldsymbol{\eta\eta}}^{-1}\mathbf{s}} = 1\ . \tag{2.18}$$

2.1.2 An asynchronous MF detector

In the previous section a synchronous MF detector was derived, i.e. the arrival time n_a, where all signal energy is received and the threshold test is carried out, is known to the detector. In the practical cases considered in this thesis, n_a is *not* known, however. Therefore, in addition to merely decide whether the signal is *present* or not, the detector also has to produce an estimate $\hat{n}_a$ of the true, unknown arrival time n_a. In this section the trivial threshold detector presented above is extended to handle the asynchronous detection.

The standard procedure when the arrival time n_a is unknown is to estimate it from the received data. The maximum likelihood (ML) estimate of the arrival time is the point of time that maximizes the MF output $m(n)$ within an interval containing the signal. The main drawback with this method is that strong noise peaks may prevent weak signals from being detected properly. This performance loss can be evaded if, like in this thesis, all indications of a possible detection are reported and post-processed, where the reports are classified into false alarms and actual detections. Assuming that the post-processing is correct, the detection of weak signals are improved compared to the ML approach.

In most practical cases, the MF output is of lowpass character and a simple peak finding algorithm that reports all positive peaks above the detection threshold m_0 can be used as an asynchronous MF detector. In this thesis, this detector is selected because of its implementation simplicity and resemblence to the synchronous MF detector that is well-known and accurate. The drawback is that analyzing its performance in a formal manner is difficult. Due to its nonlinear processing of the MF output, expressions for its statistical performance are therefore not tractable and Monte Carlo simulations are used instead.

Despite the non-tractable statistical performance of the AMFD, some conclusions about its mean and variance may be drawn from its properties. First, note that the noise may cause the peak observed in the MF output to be shifted in time. Thus, there is always a non-zero probability that the peak found in the MF output is at some sample other than n_a. Because this happen more often if the noise level is high compared to the signal, the mentioned probability increases as the SNR decreases.

The statistical performance of the AMFD is thus different from the SMFD. The mean of the MF output in the SMFD is (per definition) calculated solely from the sample values at the arrival time n_a. In contrast, the AMFD may report a sample $\hat{n}_a \neq n_a$ due to the peak being shifted by the noise as discussed above. Because the AMFD reports the position of the peak, i.e., a maximum in the MF output, $\mathrm{m}(\hat{n}_a) \geq \mathrm{m}(n_a)$. This means that the mean of the reported MF peak output is larger for the AMFD compared to the mean of the MF peak output of the SMFD. For small SNRs, the difference is larger because the MF output peak is misplaced more often, see Figure 2.2 below.

Consequently, the variance of the AMFD MF peak output is decreased compared to the SMFD MF peak output because large neighboring values are reported instead of smaller values at the true arrival time n_a. As for the mean, the effect is more pronounced at low SNR levels; the peak value is then larger than $m(n_a)$ more often.

From the discussion above, it is anticipated that the performance of the AMFD is different to the SMFD. In the next section the performance is evaluated further through Monte Carlo simulations.

2.2 Performance

In this section the performance of the two MF detectors will be analyzed and characterized. This type of knowledge is important when applying the AMFD to the applications in mind for the thesis. When possible, analytical expressions will be presented and analyzed. For the other cases, Monte Carlo simulations will be used. Especially the performance analysis of the nonlinear AMFD depends extensively on Monte Carlo simulations due to the difficult expressions concerning this detector.

2.2.1 Simulation setup

Because the simulations performed in this chapter should form a foundation to Part II, it is important that the test signal selected for the simulations in this chapter reflects a wide variety of signals.

Most real signals have their power concentrated to a certain frequency range which is of either lowpass, bandpass, or highpass character. Since the phase of the signal disappears in the matched filtering, it is of subordinary interest in this

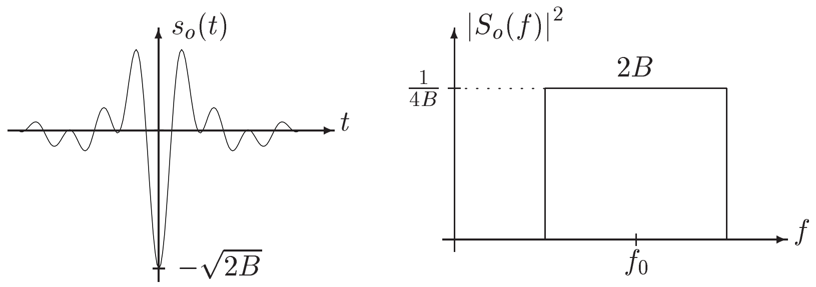


Figure 2.1: The signal to be detected (left) and its power spectrum (right). Depending on the parameters f_0 and B, a matched filter based on this signal characterizes any matched filter of lowpass, bandpass, or highpass character.

respect.

A suitable choice for the test signal is, thus, a parameterized signal that has its power uniformily distributed over a certain interval controlled by its parameters. Such a signal is depicted in Figure 2.1 and described by

$$s_o(t) \triangleq \begin{cases} -\sqrt{2\gamma^{-1} f_0} & : \quad t = 0 \\ -\frac{\sin(2\pi\gamma^{-1} f_0 t)\,\cos(2\pi f_0 t)}{\pi\sqrt{2\gamma^{-1} f_0}\, t} & : \quad \text{otherwise} \end{cases} \tag{2.19}$$

$$\gamma \triangleq \frac{f_0}{B}, \quad \gamma \geq 1 \tag{2.20}$$

where γ is the *bandwidth ratio* that characterizes the test signal, f_0 is the *center frequency*, and B is the *bandwidth*. Defining the test signal in this way, the MF performance of any signal with (near-) uniform spectrum is possible to characterize through the bandwidth ratio γ. The center frequency f_0 is then just a scaling in time. Consequently, the conclusions drawn using this test signal are widely applicable in practice.

The test signal $s_o(t)$ has an infinite support. It is, however, approximated by a truncated replica. Using the sampling period T_s and the truncation time T_0, the signal template $\mathbf{s}$ used in the simulations below is defined as

$$\mathbf{s} \triangleq \begin{pmatrix} s_o(T_0) & s_o(T_0 - T_s) & \ldots & s_o(-T_0 + T_s) & s_o(-T_0) \end{pmatrix}^T \tag{2.21}$$

in accordance with (2.3) and (2.5). The truncation time T_0 is set to

$$T_0 \triangleq \frac{\lfloor 3\gamma + \frac{1}{2} \rfloor + \frac{1}{2}}{2f_0} \tag{2.22}$$

where $\lfloor \cdot \rfloor$ means truncation to the nearest integer less than or equal to the argument.

Assuming $\boldsymbol{\eta}(n)$ to be white noise with unit variance, the optimal impulse response $\mathbf{h}_o$ of the matched filter is defined by (2.12), with $\sigma^2 = 1$. This impulse response of the filter is equal to the signal template $\mathbf{s}$, because the signal template is symmetrical and has unit energy, see the spectrum of $s_o(t)$ in Figure 2.1. This is formalized by

$$\sigma^2 = 1 \tag{2.23}$$

$$\mathbf{h}_o = \frac{\mathbf{s}}{\sqrt{\sigma^2 \mathbf{s}^T \mathbf{s}}} = \mathbf{s} \ . \tag{2.24}$$

Throughout this section, simulations are carried out using the bandwidth ratios $\gamma = 1, 1.25, 1.5, 1.72, 2$. These selected ratios span the most interesting part of the possible bandwidth ratios, see Section 2.2.4. The value 1.72 is studied for reasons that will become clear in Chapter 9.

The data used in the simulations are generated by adding a scaled signal template $\mu\mathbf{s}$ to the noisy data set and the amplitude factor μ is set to yield different SNR levels. As stated in Section 2.1.2, the arrival time n_a is unknown to the AMFD and the positions of the peaks in the MF output are reported as estimates of n_a.

Because the noise may translate a peak originating from a signal and because the noise itself may give rise to a peak, it is ambiguous whether a reported peak is due to the presence of a signal or is derived from the noise. Without resolving this ambiguity, the *analysis* of the simulation results is impossible to carry out because we must be able to count the number of detections and the number of false alarms in order to produce the statistics. Without knowing which of these two groups a particular reported peak relates to, the analysis is impossible.

Analogously with the SMFD, we define a detection to be a reported peak at $\hat{n}_a = n_a$ when a signal is present and a false alarm to be a reported peak when no signal is present. But, because the signal is longer than one sample, it is sensible to also consider a report $\hat{n}_a \neq n_a$ as a detection if a signal is present and the reported peak is sufficiently close to the true arrival time. Thus, if a signal is present, there is a point in time where a report changes from being considered as a false alarm to being considered as a detection.

To find the dividing line and to dissolve the ambiguity above, two main assumptions are made:

Assumption 2.7 *Within a received data set containing a single signal, there exists a single interval such that the characteristics of the data within the interval are in compliance with hypothesis H_1 (signal and noise) and the characteristics of the data outside the interval are in compliance with hypothesis H_0 (noise only).*

Assumption 2.8 *The detector reports are correctly classified into false alarms and detections by some post-processing algorithm.*

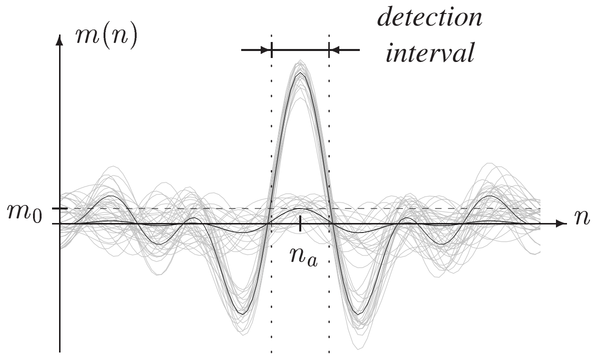


Figure 2.2: The output from the matched filter for two SNR levels, 0 dB and 10 dB (gray). The outputs when no noise is present are also shown (solid) as well as the detection threshold used (dashed). To solve the ambiguity of a false alarm and a faulty localized detection, a detection interval is introduced. Only reports within the interval may be considered as detections. Any single report within the interval is always considered as a detection. If there are several reports, the one closest to n_a is considered as a detection.

Now, under Assumption 2.7 there exists a *detection interval* such that any single report within the interval should be considered as a detection and all reports outside the interval should be considered as false alarms.[1]

If several reports are made within the interval, however, there is a problem of which report should be considered as the detection.[2] Under Assumption 2.8, it is most sensible to select the report closest to the true arrival time n_a as the actual detection because this choice yields the smallest error in both amplitude and estimated arrival time. This should be as close as possible to the net result of the combined efforts of the detector and the assumed post-processing algorithm.

There are, however, two sides of the problem of selecting the detection interval. On one hand, including parts where the true MF output, $\mathbf{h}^T\mathbf{s}(n)$, is below zero makes no sense and would impair the performance measure because the signal's presence counteracts the detection in these regions. On the other hand, Assumption 2.7 is violated by not including such regions because they are incompatible with hypothesis H_0 (noise only). The latter alternative is a safer choice since the corresponding assumptions are slightly pessimistic rather than optimistic.[3]

[1]Due to symmetry reasons, such an interval must be centered around n_a.

[2]Because we are interested in the arrival time n_a and because each occurence of the signal, naturally, arrives precisely one time, each signal occurence must correspond to at most one detection.

[3]It is worth noting that the additional positive peaks in the true MF output, see Figure 2.2, also break Assumption 2.7 if the detection interval is narrowed to include the main peak only. For the signals considered in the actual application in Chapter 9, however, these peaks do not exist and are therefore not considered any further.

The endpoints of the detection interval should thus be set to the first zero-crossings counting from the true arrival time n_a, c. f. Figure 2.2. To simplify the analysis, however, all simulations comparing different bandwidth ratios γ use the same interval length of 13 samples (0.42 ms).

Unless stated otherwise, the simulations are performed with 1000 Monte Carlo runs, with a detection threshold set to $m_0 = 1$, and a white noise with unit variance. The sampling frequency is selected to 31.25 kHz, because that is the sampling frequency set by the application in Chapter 9 where the AMFD is used. The simulation parameters presented in this section are summarized in Table 2.1.

Table 2.1: MF parameters used in the simulations unless stated otherwise

Description	*Parameter*		*Value*
Number of runs	N		1000
Bandwidth ratios	$\{\gamma\}$		$\{1, 1.25, 1.5, 1.72, 2\}$
Noise variance	σ^2		1
Detection threshold	m_0		1
Length of detection interval	L	[samples]	13
Sampling frequency	f_s	[kHz]	31.25
Center frequency	f_0	[kHz]	1.15

2.2.2 Detection and false-alarm probability

The synchronous detector

Using (2.9) and (2.10), and assuming that the noise $\boldsymbol{\eta}(n)$ is Gaussian, the MF peak output $m(n_a)$ of the SMFD is a stochastic variable having a Gaussian distribution with mean as described by (2.15) and (2.16), and with unit variance as described by (2.18). The false-alarm probability P_{FA} and the detection probability P_D may thus be calculated by means of the decision threshold m_0 using the Gaussian distribution density function $\Phi(\cdot)$, see Figure 2.3,

$$P_{FA} = 1 - \Phi(m_0) \tag{2.25}$$

$$P_D = 1 - \Phi\Big(m_0 - \sqrt{SNR_{mf}}\Big) . \tag{2.26}$$

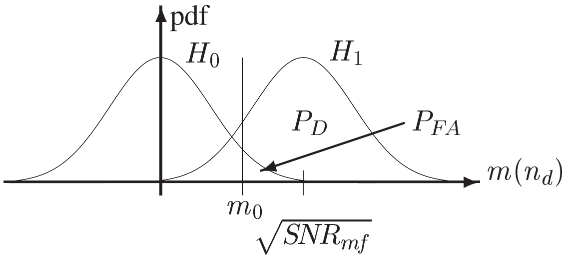


Figure 2.3: This figure shows the probability density functions (pdf:s) of the two hypotheses. Under the hypothesis H_0, the data contains zero mean noise only and the corresponding pdf is centered around zero. Under the hypothesis H_1, the expectation of the MF output is equal to $\sqrt{SNR_{mf}}$ and the pdf is centered around that value. The false alarm probability P_{FA} and the detection probability P_D are equal to the area under their respective pdf above the detection threshold m_0, as shown in the figure.

The false-alarm probability P_{FA} depends on the detection threshold m_0 only, meaning that the SMFD has a constant false-alarm rate (CFAR). This is a desirable detector property in most applications.

The asynchronous detector

For the AMFD, the performance is difficult to analyze analytically. Instead, the performance is analyzed through Monte Carlo simulations and compared to the performance of the SMFD.

There is a major problem in doing this, however, and finding some "general" performance valid for any detection threshold is in principle impossible without simulating every detection threshold setting of interest. The problem is that there may be several reports within the detection interval and, hence, the detection threshold determines which report that is selected as the detection. The result is that the estimated arrival time $\hat{n}_a$ is detection threshold dependent and thus needs to be known before any statistics on the AMFD output $m(\hat{n}_a)$ may be computed.

Fortunately, multiple reports within the detection interval are rare ($< 0.5\%$). Therefore, the mean, the variance, and the pdf of the AMFD output $m(\hat{n}_a)$ are calculated for the maximum peak within the detection interval. This is, in essence, equal to the ML approach discussed in Section 2.1.2. Because the multiple reports are so rare, the ML statistics is almost identical to the statistics of the actual detections for a particular detection threshold.

From Section 2.1.2 we expect an increased mean and decreased variance of the AMFD output $m(\hat{n}_a)$ at low SNR levels compared to the SMFD output $m(n_a)$. As the SNR increases, the arrival time estimate $\hat{n}_a$ improves and the performance of

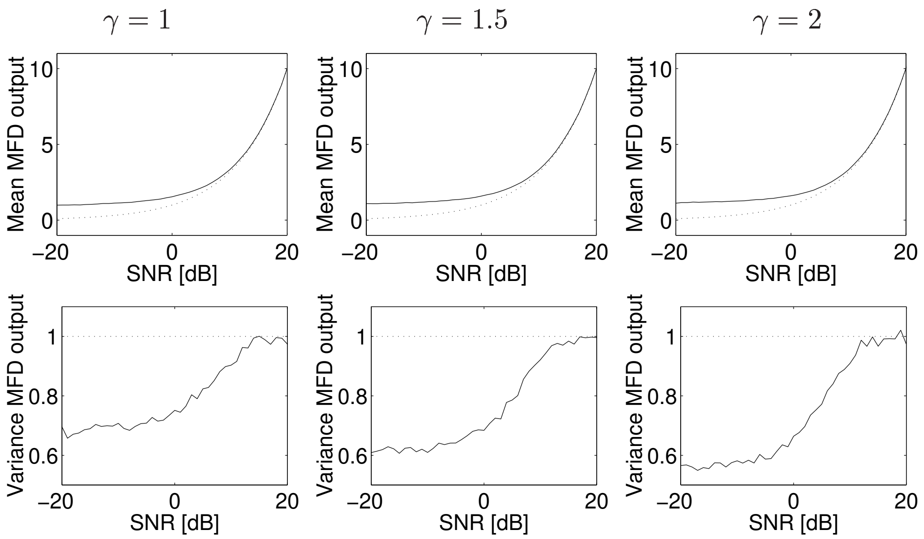


Figure 2.4: The figure shows the mean and the variance of the AMFD (solid) and the SMFD (dotted) output as a function of the SNR for different bandwidth ratios. The AMFD data was estimated from 10^4 Monte Carlo runs.

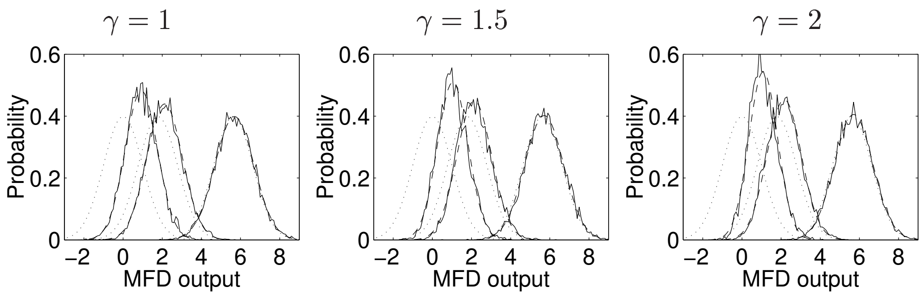


Figure 2.5: The diagrams show the estimated probability density functions of the AMFD (solid) output for different bandwidth ratios and different SNR levels. The selected SNR levels are no signal ($-\infty$ dB), 5 dB, and 15 dB. The theoretical values of the SMFD are shown (dotted) as well as a Gaussian pdf with mean and variance corresponding to the AMFD output (dashed).

the AMFD should approach the performance of the SMFD.

The results in Figure 2.4 from 10^4 Monte Carlo runs clearly show this expected behavior. At low SNR levels, the mean of $m(\hat{n}_a)$ is about one for the AMFD compared to zero for the SMFD. The variance of $m(\hat{n}_a)$ is about 0.6 for the AMFD compared to 1 for the SMFD. As the SNR increases, the differences between the AMFD and the SMFD decreases asymptotically to zero.

In Figure 2.5 the estimated pdf of the AMFD output $m(\hat{n}_a)$ is shown. As expected, the pdf of $m(\hat{n}_a)$ differs from the pdf of $m(n_a)$ at low SNR levels. The figure indicates, however, that despite the AMFD output not being Gaussian, the pdf of $m(\hat{n}_a)$ may be approximated by a Gaussian pdf if the actual mean and variance are taken into account. At high SNR levels it coincides with the $\mathcal{N}(\sqrt{SNR_{mf}}, 1)$ pdf of the SMFD.

An approximate pdf of the AMFD output $m(\hat{n}_a)$, using estimated values of its mean ($\hat{E}(m(\hat{n}_a))$) and variance ($\hat{V}(m(\hat{n}_a))$), may thus be defined as

$$\tilde{p}_{m(\hat{n}_a)}(m,\kappa) \triangleq \frac{1}{\sqrt{\hat{V}(m(\hat{n}_a))}}\, p_{\mathcal{N}}\left(\frac{m - \hat{E}(m(\hat{n}_a))}{\sqrt{\hat{V}(m(\hat{n}_a))}}\right)\Bigg|_{SNR_{mf}=\kappa} \tag{2.27}$$

where $p_{\mathcal{N}}(\cdot)$ is the pdf of Gaussian $\mathcal{N}(0,1)$ distribution, i.e. the same as the SMFD output when no signal is present.

Having the approximate pdf of $m(\hat{n}_a)$, the approximate false-alarm probability and detection probability of the AMFD may be defined. Denoting these probabilities by $\widetilde{P}_{FA}$ and $\widetilde{P}_D$, respectively, we obtain

$$\begin{aligned}
\widetilde{P}_{FA}(m_0) &= \int_{m_0}^{\infty} \tilde{p}_{m(\hat{n}_a)}(m,0)dm = 1 - \int_{-\infty}^{m_0} \tilde{p}_{m(\hat{n}_a)}(m,0)dm \\
&= 1 - \Phi\left(\frac{m_0 - \hat{E}(m(\hat{n}_a))}{\sqrt{\hat{V}(m(\hat{n}_a))}}\right)\Bigg|_{SNR_{mf}=0} \qquad (2.28) \\
\widetilde{P}_{D}(m_0,\kappa) &= \int_{m_0}^{\infty} \tilde{p}_{m(\hat{n}_a)}(m,\kappa)dm = 1 - \int_{-\infty}^{m_0} \tilde{p}_{m(\hat{n}_a)}(m,\kappa)dm \\
&= 1 - \Phi\left(\frac{m_0 - \hat{E}(m(\hat{n}_a))}{\sqrt{\hat{V}(m(\hat{n}_a))}}\right)\Bigg|_{SNR_{mf}=\kappa} \qquad (2.29)
\end{aligned}$$

where the dependence on the the detection threshold m_0 and the SNR_{mf} is explicitly indicated. Note that the AMFD is a CFAR detector like the SMFD.

Figure 2.6 shows that, while the AMFD output is not exactly Gaussian, the expressions in (2.28) and (2.29) are good approximations to the actual false-alarm probability and detection probability. The figure also shows that the detection probability of the AMFD coincide with the corresponding probability of the SMFD at high SNR levels.

In the applications considered in this thesis, a key aspect of the detector is its false-alarm probability. Therefore, in order to compare the performance of the two detectors, it seems reasonable that the threshold used in the AMFD should yield the same false-alarm probability as the threshold used in the SMFD.

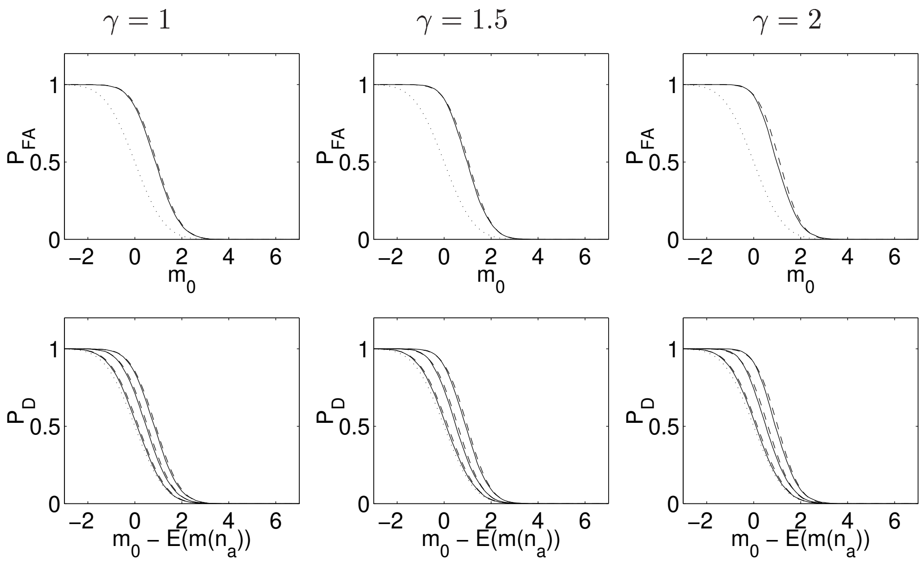


Figure 2.6: **Top:** *The diagrams show the estimated false-alarm probability of the AMFD (solid) as a function of the threshold value. The theoretical false-alarm probability of the SMFD are shown (dotted) as well as the approximate false-alarm probability $\widetilde{P}_{FA}$ of the AMFD (dashed) for the SNR levels -20 dB, 0 dB, and 10 dB, respectively.*
Bottom: *The diagrams show the estimated detection probability of the AMFD (solid) as a function of the threshold value subtracted with the $\sqrt{(}SNR_{mf})$, see (2.26). The theoretical detection probability of the SMFD is shown (dotted) as well as the approximate detection probability $\widetilde{P}_D$ of the AMFD (dashed).*

Combining (2.25) and (2.28), we see that this is accomplished by defining the detection threshold $\tilde{m}_0$ of the AMFD as a linear transformation of the detection threshold m_0 of the SMFD according to

$$\tilde{m}_0 \triangleq \sigma_m m_0 + \tilde{m} \tag{2.30}$$

where $\tilde{m}$ and σ_m are the mean and standard deviation of $m(\hat{n}_a)$ when no signal is present. The mean and the variance for different bandwidth ratios γ are shown in Figure 2.7.

In Figure 2.8 and Figure 2.9 the performance of the AMFD is compared to the SMFD. Figure 2.8 shows the estimated receiver operating characteristics (ROC) curves of the AMFD for different bandwidth ratios and the SNR levels -20 dB, 0 dB, and 10 dB, respectively. As can be seen in the figure, the AMFD performs in general well but has a slight performance loss compared to the SMFD, due to the unknown arrival time n_a.

The diagrams in Figure 2.9 compare the detection probability of the AMFD

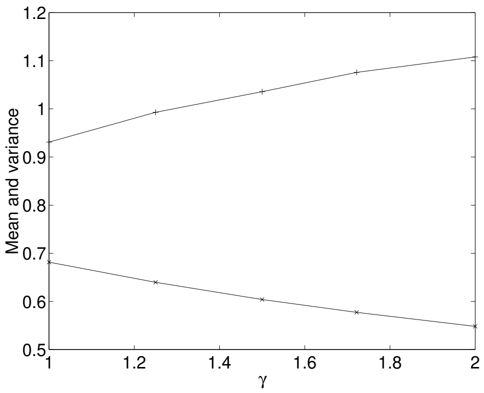


Figure 2.7: The figure shows the mean (+) and the variance (×) of $m(\hat{n}_a)$ when no signal is present as a function of the bandwidth ratio γ.

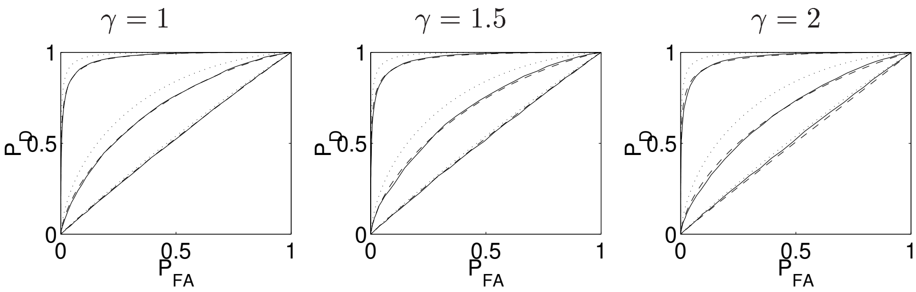


Figure 2.8: The estimated receiver operating characteristics (ROC) curves of the AMFD are shown (solid) for different bandwidth ratios and the SNR levels -20 dB, 0 dB, and 10 dB. The theoretical ROC curves of the SMFD are also shown (dotted) as well as the approximate ROC curves of the AMFD (dashed).

with the SMFD as a function of the SNR for different false-alarm probabilities and bandwidth ratios. The threshold m_0 of the SMFD was set to achieve the false-alarm probabilities 0.2, 0.05, and 0.001, corresponding to $m_0 = 0.84, 1.6, 3.1$, respectively. The detection threshold transformation (2.30) is used to yield equal false-alarm probabilities and we see that the performance loss is limited to less than 3 dB. At higher thresholds and higher SNR levels, the results suggests that the two detectors are comparable in performance.

A note of precaution is, however, necessary here. At closer inspection, the detection threshold transformation (2.30) results in an AMFD threshold $\tilde{m}_0$ less than the SMFD threshold m_0 for large threshold values. This suggests that a performance *gain* is expected when $m_0 > \frac{\tilde{m}}{1-\sigma_m}$ which is about 5 for the parameters $\tilde{m}$ and σ_m considered here. This is of course not possible and the detection threshold

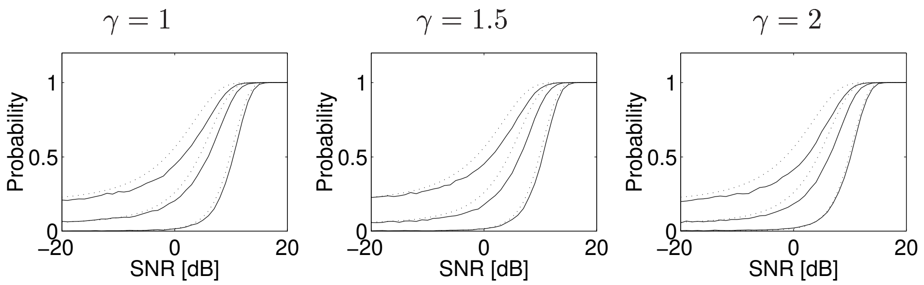


Figure 2.9: The diagrams show the estimated detection probability of the AMFD (solid) as a function of the SNR for different false-alarm probabilities and bandwidth ratios. The theoretical values of the SMFD are shown (dotted). The threshold m_0 of the SMFD was set to achieve the false-alarm probabilities $0.2, 0.05$, and 0.001, corresponding to $m_0 = 0.84, 1.6, 3.1$, respectively, whereas the threshold $\tilde{m}_0$ of the AMFD was set by the transformation (2.30) to yield equal false-alarm probabilities.

transformation in (2.30) should be used with this in mind. The error stems from the approximation in (2.28) that over-estimates the false-alarm probability slightly, see Figure 2.6. The transformation in (2.30) then yields a false-alarm probability of the AMFD that is less than the false-alarm probability of the SMFD and the result is a faulty increase in the detection probability of the AMFD. The correct transformation yields $\tilde{m}_0 > m_0 \; \forall \; m_0$ with an asymptotically decreasing difference for increasing m_0.

Hence, the performance diagrams in Figure 2.9 may be misleading, due to the faulty transformation, for the high threshold case where the AMFD seem to be comparable with the SMFD in performance. An alternative would be to set the threshold $\tilde{m}_0$ so that the *estimated* false-alarm probability is equal to the false-alarm probability of the SMFD. This would, however, require a lot more Monte Carlo runs than the 1000 runs used here. Since the gain of acquiring this information is more academical than useful, it has not been pursued further. In any case, the results indicate that the performance loss is limited to less than 3 dB which is assuring.

Because the AMFD is applied to an "on-going" signal, there is a parameter of more interest than the false-alarm probability, namely the false-alarm *intensity*. This entity represents the number of false alarms per unit time and is valuable for the post-processing algorithm mentioned in Assumption 2.8. In Chapter 9, for example, it is used in a Bayesian approach to this false-alarm/detection classification problem.

From 10^6 Monte Carlo runs, Figure 2.10 shows the estimated false-alarm intensity β when using the detection threshold $m_0 = -\infty$ corresponding to the false-

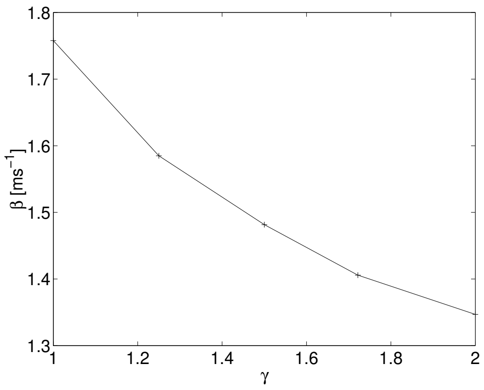


Figure 2.10: The figure shows the estimated false-alarm intensity as a function of the bandwidth ratio γ based on 10^6 Monte Carlo runs using the detection threshold $m_0 = -\infty$. This threshold yields the maximum false-alarm intensity because $\widetilde{P}_{FA} = 1$.

alarm probability $\widetilde{P}_{FA} = 1$ which yields the maximum false-alarm intensity. As the figure shows, β ranges from 1.3 to 1.8 in this case, which means that, on average, there is about one to two false alarm every millisecond. To find the false-alarm intensity for some other detection threshold, the intensity shown in Figure 2.10 is multiplied with the false-alarm probability $\widetilde{P}_{FA}$ because the false-alarm intensity is directly proportional to the false-alarm probability.

2.2.3 Accuracy

In most applications, the accuracy of the detector is important. In this thesis the variance of the reported arrival time $\hat{n}_a$ is considered. Clearly, the variance of the synchronous MF detector is zero. The variance of the asynchronous detector is found by Monte Carlo simulations due to the difficulties in finding analytical expressions for this detector.

Estimating the variance of the arrival time $\hat{n}_a$ from the MF output directly would yield poor estimates, however, because the selected sampling period is long compared to the variance of the asynchronous detector. To obtain a better estimate, the MF output is oversampled ten times through interpolation prior to the detection and the arrival time variance estimation.

In Figure 2.11 the variance of the asynchronous detector using the five different bandwidth ratios is shown. As the figure shows, the variance is small for high SNRs and increases when the SNR decreases. There is no noticeable difference between the different bandwidth ratios. As expected, the variance approaches the variance of a uniform distribution over the detection interval when the SNR is so low that

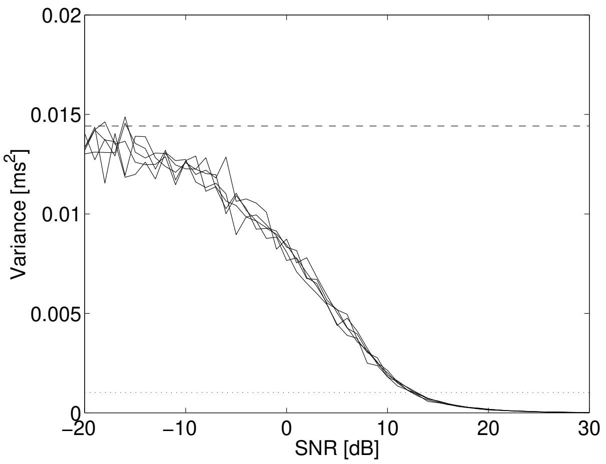


Figure 2.11: The figure shows the estimated variance (solid) of the arrival times for the five different bandwidth ratios whith the MF output oversampled ten times prior to the detection. For decreasing SNR levels, the variance increases asymptotically towards the variance of a uniform distribution over the detection interval (dashed). For comparison, the squared (original) sample period ($1/f_s^2$) is shown (dotted). See also Figure 2.2.

no actual detections can be made. Instead, the reports considered as detections are false alarms that, of course, are uniformily distributed over the interval.

Remark Without interpolating the MF output, as done in this simulation, the variance of the AMFD is always larger than $\frac{1}{12f_s^2}$ if the input signals are asynchronously sampled from continuous time.

2.2.4 Resolution

When deriving the MF detector, it was assumed that at most one signal was present in the recorded data. In particular, it was assumed that no signals were overlapping. For the applications considered in this thesis, no such guarantee exists, however, and it is important to analyze the detector's resolution capabilities, i.e. how much must two different signals be separated in time in order for the detector to report two detections? This question will be answered below.

Again, an analytical analysis is difficult and the resolution capabilities have been obtained through Monte Carlo simulations. It will be shown, however, that Rayleigh's resolution criterion may also be used in an appropriate manner.

The simulation setup was almost identical to the one described in Section 2.2.1. The only differences were that *two* signals were added to the data set. Moreover, if

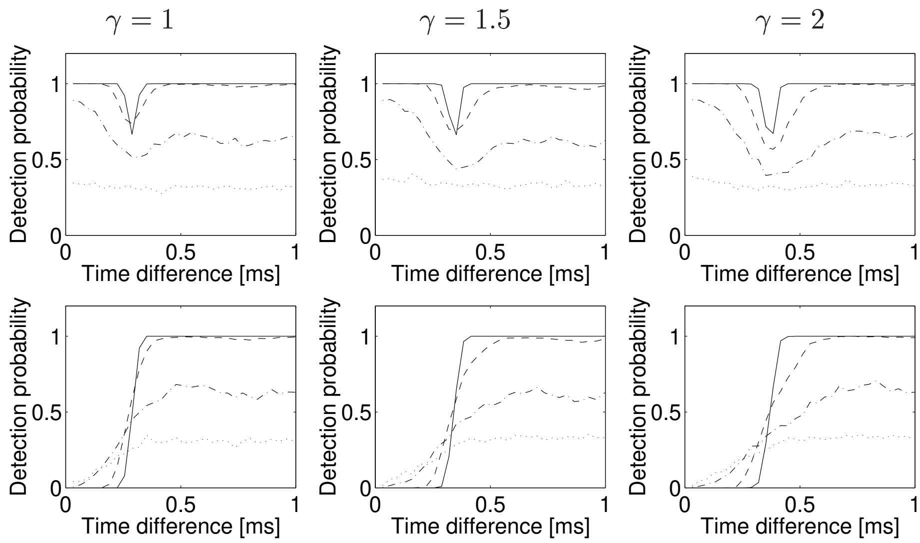


Figure 2.12: The diagrams show the estimated probabilities of detecting one signal (top) and two identical signals (bottom) as a function of the time separation at different SNRs, -20 dB (dotted), 0 dB (dash dotted), 10 dB (dashed), and 20 dB (solid).

both signals were competing for a single detection, at small time differences where the detection intervals overlap, the detection was assigned to the first signal. This way, the detection probability of the first signal is a measure of the influence (if any) that overlapping signals may have on the overall detection probability as analyzed in Section 2.2.2. The detection probability of the second signal is a measure of the probability of detecting *both* signals and, thus, a direct measure of the resolution capability.

Figure 2.12 shows the probability of detecting one and two signals at different SNRs. As expected, the probability of detecting two signals decreases to zero for small time differences. The probability of detecting one signal increases instead, because the signals add and interfere constructively. Surprisingly, there is a dip in the detection probability between 0.2 ms and 0.4 ms. The reason is that the two signals interfere destructively with one another in this interval. The figure suggests that the resolution of the AMFD is about 0.3-0.4 ms.

Based on the results in Figure 2.12 and practical experience, we have found that the resolution capability of the AMFD may successfully be described by the formula

$$\Delta_{\text{res}} \triangleq \arg \left\{ \widehat{P}_{FA}^{(2)}(\Delta t_a) \Big|_{\text{SNR}=20 \text{ dB}} = 0.85 \right\} \tag{2.31}$$

where $\Delta t_a = \Delta n_a / f_s$ is the time difference and $\widehat{P}_{FA}^{(2)}(\Delta t_a)$ is the estimated false-

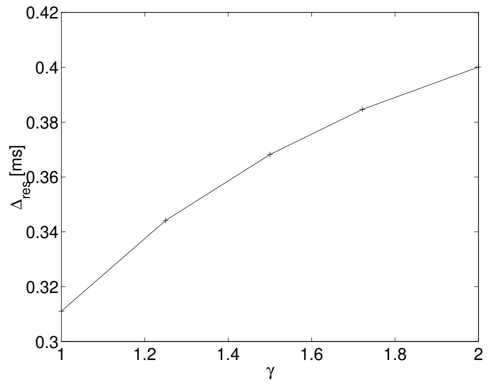


Figure 2.13: The figure shows the estimated resolution dependency of the bandwidth ratio γ. Over the interval, the resoloution ranges from 0.3 ms to 0.4 ms (10-13 samples).

alarm probability of the second signal using the sigmoid model

$$\widehat{P}_{FA}^{(2)}(\Delta t_a) \quad \triangleq \quad \frac{1}{1+e^{-C_1\Delta t_a + C_0}} \tag{2.32}$$

where C_0 and C_1 are adjusted to the measured curves for 20 dB SNR using the simplex method.

In Figure 2.13 the resulting resolution estimates of the AMFD are shown for the different bandwidth ratios γ. As the figure shows, the resolution ranges from 0.3 ms to 0.4 ms (10-13 samples).

Using Rayleigh's resolution criterion, instead, the resolution of the asynchronous detector may be calculated analytically.[4] Figure 2.14 shows the Rayleigh resolution as a function of the bandwidth ratio together with the resolution definition in (2.31). As the figure shows, there is a good agreement between the two different resolution criterias and the Rayleigh resolution may indeed be used instead.

Figure 2.14 also shows the sidelobe rejection, both the primary and the global, because it influences the potential problem of detecting the same signal several times, one time for each sidelobe. The primary sidelobe rejection is defined as the relative amplitude difference of the main peak and the *first* sidelobe, whereas the global sidelobe rejection is defined as the relative amplitude difference of the main peak and the sidelobe with the *highest amplitude*. As shown in the figure, the global sidelobe rejection is close to unity for the bandwidth ratios studied here which suggests that multiple detections are rare in these cases.

[4]The resolution according to Rayleigh is the time difference of the main peak and the first minimum of the MF output.

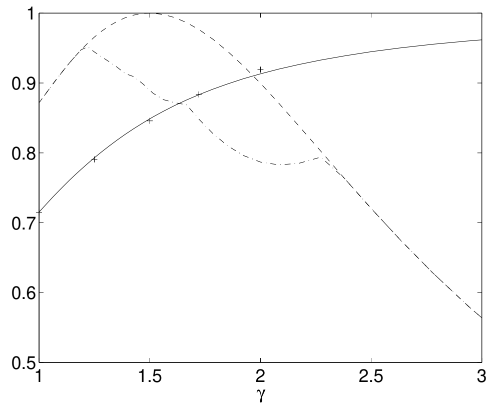


Figure 2.14: The figure shows the product $2f_0\Delta t_R$ (solid) where Δt_R is the Rayleigh resolution, the primary sidelobe rejection (dashed), and the global sidelobe rejection (dash-dotted). Also included is the estimated resolution based on Monte Carlo simulations yielding a detection probability $P_D = 0.85$ (+).

2.3 Concluding remarks

In this chapter, both the synchronous and the asynchronous matched filter detectors were derived and analyzed. Despite the small difference between the SMFD and the AMFD, an analytical analysis of the AMFD was not tractable due to the nonlinear arrival time estimator. Instead Monte Carlo simulations were used to evaluate the performance of the AMFD and comparing it to the SMFD.

As shown above, the performance of the AMFD is similar to the SMFD with at most a 3 dB performance loss in detection probability.

The simulation methods presented here provides a framework for assessing the performance criterias such as false-alarm intensity, arrival-time accuracy, and detection resolution in the application in Chapter 9. The results provided by the simulations in this chapter are promising and applying the AMFD to a real scenario seem feasible.

CHAPTER 3

Kalman filtering and prediction

PREDICTING noisy signals generated by a dynamic system is a task found in a large set of practical applications. There are different ways of solving this problem and in this chapter we look at the well-known Kalman filter approach where the predictor is implemented simply as a recursive, linear filter with time-varying gain.

The Kalman filter [77] [52] [46] has had an enormous act on signal processing and technological development for at least two major reasons. First, a distinctive feature of Kalman filters is their *state-space* formulation and recursive implementation. This reduces the storage requirements and improves the computational efficiency because only the last estimate is needed at each step of the filtering process.

Second, the Kalman filter is the linear filter that produces the minimum mean squared error (MMSE) estimate given all past data. If the system that generates the measurement signal is linear and the noise is Gaussian, then the Kalman filter is the optimal MMSE estimator among all filters; both linear and nonlinear.

The Kalman filter solves problems of smoothing, filtering, and prediction in a simple, but stringent, form. In the Kalman filter realm, these estimates are just special cases of the general solution. If the process under study is stationary then the Kalman filter, with time-varying gain, converges towards an ordinary constant filter that is easy to implement.

The design of the Kalman filter algorithm is based on a model with two equations: the process equation and the measurement equation. The former describes the dynamics of and the inputs to the process (or system) that we are interested in and the latter describes the way measurements are collected from the process along with possible inputs. Both these equations/models may be time varying.

Often, the inputs in these models are signals regarded as noise with known second order properties; process noise and measurement noise, respectively. The Kalman filter algorithm is determined by the possibly time-varying covariance matrices of these noises and the dynamics of the linear state space model.

Before the Kalman algorithm can be applied to data, the initial values of the state estimates need to be set. This issue is often omitted in the literature by just setting the intial values to their expectations since this yields *unbiased* estimates. In a practical situation, this method may be limited by either unknown expectations which forces the designer to preset the initial values to some ad hoc value that is “suitable”, or initial values that are very uncertain which yield a large initial variance. Either of these two scenarios may result in non-MMSE and even biased estimates. Often, however, this aspect is neglected because the effects of any initialization errors decay exponentially with time for detectable systems. Clearly, this approach does not work very well in applications where the initial estimates must be as correct as possible.

One such application is target tracking, see Chapter 9, where correct initialization is of vital importance to the performance. If the initial values are very uncertain or even unknown, the best performance is attained by using a number of the first measurements to find an MMSE state estimate as the starting point for the Kalman filter.

Another important property of the Kalman filter in this regard is its consistency [12]. In demanding applications where optimal performance is required, the consistency of the filter is crucial.

In this chapter, prediction through Kalman filtering is presented. The Kalman filter algorithm for time-invariant models is presented as well as how to derive a discrete-time linear stochastic model from a continuous-time model. Discussion of any application specific issues is deferred to Part II. The model used in the numerical example section, however, is similar to the one used in Chapter 9 because this model is both simple to understand and complex enough to show the main ideas of this chapter. We show that it is fairly simple to properly initiate the Kalman filter and to assure its consistency.

3.1 State space models

The mathematical formulation of Kalman filters are in state space form. Using these concepts, a recursive implementation suitable for digital computers is straight-forward to derive.

Because many practical applications involve a continuous-time system, we recapitulate the derivation of the discrete-time dynamic model needed by the Kalman algorithm from a continuous-time model. In cases where the dynamics actually takes place in discrete time, this step is omitted and the discrete-time process model is formulated directly.

Further, we assume a true discrete-time measurement acquisition system because this is often the case in practical applications. If necessary, the process-model derivation is simple to generalize to continuous-time measurement models.

3.1.1 The continuous-time process equation

A linear time-invariant system is assumed to be described in state space form as

$$\mathbf{dx}_c(t) \quad = \quad \mathbf{A}\mathbf{x}_c(t)\,dt + \mathbf{G}\,\mathbf{dw}(t) \tag{3.1}$$

where $\mathbf{x}_c(t)$ is a n_x-dimensional continuous-time state vector consisting of the states $x_1^{(c)}(t), x_2^{(c)}(t), \ldots, x_{n_x}^{(c)}(t)$. The n_w-dimensional entity $\mathbf{dw}(t)$ is the *Wiener increment* [8] of an n_w-dimensional Wiener process $\mathbf{w}(t)$.[1] The $n_x|n_x$-dimensional matrix $\mathbf{A}$ is the state transition matrix specifying the dynamics of the system. The $n_x|n_w$-dimensional gain matrix $\mathbf{G}$ reflects the amount of influence the noise has on each state.

The incremental covariance matrix $\mathbf{\Lambda}(t)\,dt$ of the Wiener increment $\mathbf{dw}(t)$ is defined as

$$\mathbf{\Lambda}(t)\,dt \stackrel{\triangle}{=} E(\mathbf{dw}(t)\mathbf{dw}^T(t)) \quad . \tag{3.2}$$

3.1.2 The discrete-time process equation

Often, signal processing is performed on a digital computer in which case discrete-time models must be used. By first solving the stochastic differential equation (3.1) of the continuous-time process model and then discretizing the solution, we obtain the discrete-time process model.

Solving (3.1) yields [9]

$$\begin{aligned}
\mathbf{dx}_c(t) &= \mathbf{A}\mathbf{x}_c(t)\,dt + \mathbf{G}\,\mathbf{dw}(t) \\
e^{-\mathbf{A}t}\left(\mathbf{dx}_c(t) - \mathbf{A}\mathbf{x}_c(t)\,dt\right) &= e^{-\mathbf{A}t}\mathbf{G}\,\mathbf{dw}(t) \\
\mathbf{d}\left(e^{-\mathbf{A}t}\mathbf{x}_c(t)\right) &= e^{-\mathbf{A}t}\mathbf{G}\,\mathbf{dw}(t) \\
e^{-\mathbf{A}t}\mathbf{x}_c(t) &= C + \int^t \mathbf{e}^{-\mathbf{A}\tau}\mathbf{G}\,\mathbf{dw}(\tau) \\
\mathbf{x}_c(t) &= Ce^{\mathbf{A}t} + \int^t \mathbf{e}^{\mathbf{A}(t-\tau)}\mathbf{G}\,\mathbf{dw}(\tau) \quad .
\end{aligned} \tag{3.3}$$

[1] The stochastic differential equation (3.1) may be interpreted (with some care) as $\dot{\mathbf{x}}_c(t) = \mathbf{A}\mathbf{x}_c(t) + \mathbf{G}\mathbf{e}(t)$, where $\mathbf{e}(t)$ is a n_w-dimensional continuous-time white noise process.

Solving for the constant C using an arbitrary initial time t_0 we obtain

$$\mathbf{x}_c(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_c(t_0) + \int_{t_0}^{t} \mathbf{e}^{\mathbf{A}(t-\tau)}\mathbf{G}\,\mathbf{dw}(\tau) \tag{3.4}$$

$$= e^{\mathbf{A}(t-t_0)}\mathbf{x}_c(t_0) + \int_{0}^{t-t_0} \mathbf{e}^{\mathbf{A}(t-t_0-\tau)}\mathbf{G}\,\mathbf{dw}(t_0+\tau) \ . \tag{3.5}$$

Using the sampling period T and selecting $t_0 = 0$, the state at time $(k+1)T$ is

$$\begin{aligned} \mathbf{x}_c((k+1)T) &= e^{\mathbf{A}(k+1)T}\mathbf{x}_c(0) + \int_0^{(k+1)T} \mathbf{e}^{\mathbf{A}((k+1)T-\tau)}\mathbf{G}\,\mathbf{dw}(\tau) \qquad (3.6) \\ &= e^{\mathbf{A}T}e^{\mathbf{A}kT}\mathbf{x}_c(0) + e^{\mathbf{A}T}\int_0^{kT} \mathbf{e}^{\mathbf{A}(kT-\tau)}\mathbf{G}\,\mathbf{dw}(\tau) \\ &\quad + \int_{kT}^{kT+T} \mathbf{e}^{\mathbf{A}(kT+T-\tau)}\mathbf{G}\,\mathbf{dw}(\tau) \\ &= e^{\mathbf{A}T}\mathbf{x}_c(kT) + \int_0^{T} \mathbf{e}^{\mathbf{A}(T-\tau)}\mathbf{G}\,\mathbf{dw}(kT+\tau) \qquad (3.7) \end{aligned}$$

which is a complete discrete-time representation of the continuous-time differential equation (3.1). To simplify the notation, we then write the discrete-time process model at sample k as

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{v}_1(k) \tag{3.8}$$

where

$$\mathbf{x}(k) \triangleq \mathbf{x}_c(kT) \tag{3.9}$$

$$\mathbf{F} \triangleq e^{\mathbf{A}T} \tag{3.10}$$

$$\mathbf{v}_1(k) \triangleq \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{G}\,\mathbf{dw}(kT+\tau) \tag{3.11}$$

$$\mathbf{Q}_1(k) \triangleq E(\mathbf{v}_1(k)\mathbf{v}_1^T(k)) \ . \tag{3.12}$$

Here, the transition matrix $\mathbf{F}$ describes the discrete-time system dynamics[2] and the n_x-dimensional vector $\mathbf{v}_1(k)$ represents process noise[3] modeled as independent zero-mean, white-noise processes with covariance matrix $\mathbf{Q}_1(k)$.

[2]The matrix $e^{\mathbf{A}T}$ may be derived via, for example, the series expansion $e^{\mathbf{A}T} = \mathbf{I} + \mathbf{A}T + \frac{1}{2!}(\mathbf{A}T)^2 + \ldots$. See [9] for details.

[3]In equation (3.11), the entity $\mathbf{dw}(kT+\tau)$ may be interpreted as $\mathbf{e}(kT+\tau)d\tau$, where $\mathbf{e}(t)$ is a n_x-dimensional continuous-time white noise process.

3.1.3 The discrete-time measurement equation

The measurement model describes how the measurements are collected. We assume that n_y noisy linear combinations of $\mathbf{x}(k)$ are measurable at each discrete time. Thus, the measurement is here modeled as

$$\mathbf{y}(k) \stackrel{\triangle}{=} \mathbf{H}\mathbf{x}(k) + \mathbf{v}_2(k) \tag{3.13}$$

where $\mathbf{y}(k)$ is a n_y-dimensional measurement vector, $\mathbf{H}$ is a $n_y|n_x$-dimensional mixing matrix that describes the amount each state in $\mathbf{x}(k)$ influences each element of the measurement vector. The n_y-dimensional vector $\mathbf{v}_2(k)$ is the measurement noise modeled as independent zero-mean white-noise processes with a known $n_y|n_y$ covariance matrix

$$\mathbf{Q}_2(k) \stackrel{\triangle}{=} E(\mathbf{v}_2(k)\mathbf{v}_2^T(k)) \ . \tag{3.14}$$

3.2 The Kalman filter

Here, the Kalman algorithm for models of type (3.8)-(3.14) is presented as well as some comments on how to initiate it and evaluate its consistency with the data. It is in the following assumed that the noises $\mathbf{v}_1(k)$ and $\mathbf{v}_2(l)$ are statistically independent except for $k = l$ where their mutual covariance matrix is

$$\mathbf{Q}_{12}(k) \stackrel{\triangle}{=} E(\mathbf{v}_1(k)\mathbf{v}_2^T(k)) \ . \tag{3.15}$$

3.2.1 Kalman filter algorithm

Assuming that the state space model (3.8)-(3.14) is an accurate description of the true system, the Kalman filter is a linear estimator that provides the MMSE filtered estimate $\hat{\mathbf{x}}(k|k)$ and the MMSE one-step prediction $\hat{\mathbf{x}}(k+1|k)$, c. f. [4] [46],

$$\begin{aligned}
\tilde{\mathbf{y}}(k) &= \mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k-1) && (3.16)\\
\mathbf{S}(k) &= \mathbf{H}\mathbf{P}(k|k-1)\mathbf{H}^T + \mathbf{Q}_2(k) && (3.17)\\
\mathbf{K}_f(k) &= \mathbf{P}(k|k-1)\mathbf{H}^T\mathbf{S}^{-1}(k) && (3.18)\\
\hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + \mathbf{K}_f(k)\tilde{\mathbf{y}}(k) && (3.19)\\
\hat{\mathbf{x}}(k+1|k) &= \mathbf{F}\hat{\mathbf{x}}(k|k) + \mathbf{Q}_{12}(k)\mathbf{Q}_2^{-1}(k)(\mathbf{y}(k) - \mathbf{H}\hat{\mathbf{x}}(k|k)) && (3.20)\\
\mathbf{P}(k|k) &= (\mathbf{I} - \mathbf{K}_f(k)\mathbf{H})\mathbf{P}(k|k-1) && (3.21)\\
\mathbf{P}(k+1|k) &= (\mathbf{F} - \mathbf{Q}_{12}(k)\mathbf{Q}_2^{-1}(k)\mathbf{H})\mathbf{P}(k|k)(\mathbf{F} - \mathbf{Q}_{12}(k)\mathbf{Q}_2^{-1}(k)\mathbf{H})^T \\
&\quad + \mathbf{Q}_1(k) - \mathbf{Q}_{12}(k)\mathbf{Q}_2^{-1}(k)\mathbf{Q}_{12}^T(k) && (3.22)
\end{aligned}$$

where $\mathbf{I}$ is the $n_x|n_x$ identity matrix, $\tilde{\mathbf{y}}(k)$ is the measurement prediction error, called the *innovation*, and $\mathbf{K}_f(k)$ is the Kalman filter gain. The $n_x|n_x$ matrix $\mathbf{P}(k|k)$, $\mathbf{P}(k+1|k)$, and the $n_y|n_y$ matrix $\mathbf{S}(k)$ are the covariance matrices for the state filter error, the one-step state prediction error, and the measurement prediction error, respectively. They are defined as

$$\mathbf{P}(k+i|k) \triangleq E(\tilde{\mathbf{x}}(k+i|k)\tilde{\mathbf{x}}^T(k+i|k)), \quad i=0,1 \tag{3.23}$$

$$\mathbf{S}(k) \triangleq E(\tilde{\mathbf{y}}(k)\tilde{\mathbf{y}}^T(k)) \ . \tag{3.24}$$

where

$$\tilde{\mathbf{x}}(k+i|k) \triangleq \mathbf{x}(k+i) - \hat{\mathbf{x}}(k+i|k) \ . \tag{3.25}$$

Remark If $\mathbf{Q}_{12}(k) = \mathbf{0}$, the time-update equations (3.20) and (3.22) are simplified considerably and it is then not required that $\mathbf{Q}_2^{-1}(k)$ exists. If, however, $\mathbf{Q}_2^{-1}(k)$ does not exist, then $\mathbf{HP}(k|k-1)\mathbf{H}^T$ must not be zero for $\mathbf{S}^{-1}(k)$ to exist.

Remark If $\mathbf{Q}_{12}(k) \neq \mathbf{0}$ and $\mathbf{Q}_2^{-1}(k)$ does not exist, only the filtered estimate $\hat{\mathbf{x}}(k|k)$ and its covariance matrix $\mathbf{P}(k|k)$ become non-computable. By rewriting the recursions and formulate them using the one-step prediction $\hat{\mathbf{x}}(k+1|k)$ and its covariance matrix $\mathbf{P}(k+1|k)$ only, these entities are still computable.

Square-root Kalman filtering algorithms

The basic form of the Kalman filter algorithm as presented above suffers from a numerical instability problem. The recursive formula in (3.21) may result in the filter covariance matrix $\mathbf{P}(k|k)$ not being nonnegative definite, as required. To overcome these numerical problems, several modifications have been presented in the literature [96]. One method is the so-called *square-root filtering* method in which the square root of the covariance matrix $\mathbf{P}^{1/2}(k|k)$ is propagated instead. The actual covariance matrix is then calculated at each filtering step from the square root with

$$\mathbf{P}(k|k) \triangleq \mathbf{P}^{1/2}(k|k)\mathbf{P}^{T/2}(k|k) \tag{3.26}$$

which is always nonnegative definite.

Another advantage of this method is that the *conditioning number* of the matrix to be propagated is improved compared to the original algorithm, see [46] for details.

3.2.2 Initialization

Before the use of the recursion (3.16)-(3.22), the initial state estimate and its covariance matrix need to be set properly to achieve optimal performance. As mentioned above, there are three alternatives for this. The initial values are either:

- estimated from the data,
- set to their expectations with a possibly infinite covariance matrix, or
- set to some "safe" ad hoc values.

In demanding applications in practice, the first method is preferred unless the expectations are accurately known.

An elegant combination of the two first alternatives is presented in [6] where a modified Kalman filter is introduced that handles both finite and infinite initial covariance matrices as well as combinations of these. This filter may be viewed as being an augmented Kalman filter where the augmented part considers the initially infinite covariance matrix estimates. The most interesting property of this filter is that it reduces to the ordinary Kalman filter when enough data has been collected; the augmented part of the filter is in effect during the intitial transient phase only.

One drawback of this elegant approach is the elaborate calculations that are difficult to justify for small system models with a short initial transition phase. For system models with scalar measurements, however, the modified Kalman filter is only slightly more complex than the ordinary one [54].

In cases with multi-dimensional measurements, unknown expectations, and a short initial transition phase, a direct estimation from data may be preferable. Given a particular "starting" time k_0 and the first l samples of the measurements $\mathbf{y}(k)$, it is possible to find an MMSE estimate $\hat{\mathbf{x}}(k_0 + l - 1|k_0 + l - 1)$ and its covariance matrix $\mathbf{P}(k_0+l-1|k_0+l-1)$ if $l \geq l_0$ where l_0 is the number of measurements required for the system to be observable from $\mathbf{y}(k)$. See Appendix 3.A for details.

3.2.3 Consistency

If the assumed model is suitably close to the real system, a properly tuned Kalman filter produces (close to) consistent estimates. Checking the consistency during the tuning is often advisable because it may provide additional information valuable to the tuning process [12].

For the Kalman filter to provide consistent estimates, the following conditions have to be fulfilled:

1. The filter is unbiased, i.e. the state errors are zero mean.

2. The covariance matrices calculated by the filter are equal to the covariance matrices of the actual state and measurement errors.

3. The innovations are zero mean and white.

4. The initial state and its covariance matrix are set to their expectations or unbiased estimates of these entities.

In [12], the following tests are based on N Monte Carlo simulations and used to test the consistency of a Kalman filter applied to measurements produced by a model:

1. **Normalized (state) estimation error (NEE) –** Consistency of the filtered state and its covariance matrix may be tested using

$$\bar{\epsilon}_x(k) \quad \triangleq \quad \frac{1}{N}\sum_{i=1}^{N}\tilde{\mathbf{x}}_i^T(k|k)(\operatorname{diag}\mathbf{P}_i(k|k))^{-1/2} \tag{3.27}$$

which is $\mathcal{N}(0, \frac{1}{\sqrt{N}})$ distributed under the hypothesis that condition 1 and 2 are true.

2. **Normalized (state) estimation error squared (NEES) –** Another test of the consistency of the filtered state and its covariance matrix is

$$\bar{\epsilon}_{xx}(k) \quad \triangleq \quad \frac{1}{N}\sum_{i=1}^{N}\tilde{\mathbf{x}}_i^T(k|k)\mathbf{P}_i^{-1}(k|k)\tilde{\mathbf{x}}_i(k|k) \tag{3.28}$$

where $N\bar{\epsilon}_{xx}(k)$ is χ^2 distributed with Nn_x degrees of freedom under the hypothesis that condition 1 and 2 are true.

3. **Normalized innovation squared (NIS) –** Consistency of the innovations and their covariance matrix may be tested using

$$\bar{\epsilon}_{yy}(k) \quad \triangleq \quad \frac{1}{N}\sum_{i=1}^{N}\tilde{\mathbf{y}}_i^T(k)\mathbf{S}_i^{-1}(k)\tilde{\mathbf{y}}_i(k) \tag{3.29}$$

where $N\bar{\epsilon}_{yy}(k)$ is χ^2 distributed with Nn_y degrees of freedom under the hypothesis that condition 1 and 2 are true.

4. **Innovations autocorrelation (IAC) –** Whiteness of the innovations is tested using

$$\bar{\rho}(k,j) \quad \triangleq \quad \sum_{i=1}^{N}\tilde{\mathbf{y}}_i^T(k)\tilde{\mathbf{y}}_i(j)\left[\sum_{i=1}^{N}\tilde{\mathbf{y}}_i^T(k)\tilde{\mathbf{y}}_i(k)\sum_{i=1}^{N}\tilde{\mathbf{y}}_i^T(j)\tilde{\mathbf{y}}_i(j)\right]^{-1/2} \tag{3.30}$$

which may, for N large enough, be approximated with a normal distribution with zero mean under the hypothesis that condition 3 is true.

There are cases where Monte Carlo simulations are not possible to use. In [12], a real time consistency test is constructed for the innovations by using their assumed ergodicity and taking the time averages instead of the ensemble averages.

The consistency tests presented above present a challenge when applying them to an actual application. One may easily end up in a catch-22 scenario where a hypothesized system model needs to be correct in order to test whether it is (sufficiently) correct for the corresponding Kalman filter to provide consistent estimates. Often, however, a (reasonably) correct dynamic model is available, but it may be nonlinear, or of very large dimension and complicated. In these cases, there may be a desire to use a simpler model in the actual Kalman filter implementation. The consistency tests are then important tools when evaluating a filter design based on the approximate system model. Using the consistency tests, a stringent evaluation of the validity of the approximate system model for designing a Kalman filter estimator is possible.

The basic requirement for performing the consistency tests is that the actual true states are available. This means that even if the system model is totally unknown, the consistency tests may be used if there are alternative ways to acquire this information. To improve the statistics, it is preferable to be able to repeat the scenario a number of times. A particular scenario may be a device that, due to economical constraints for example, uses noisy sensors to measure a subset of the available states. During the development in the lab, however, the additional cost to measure all states using accurate sensors is often minimal. Then, the consistency tests – where the high-quality measurements are used as the true states – may be applied as described above to achieve the "optimal" filter design.

3.3 A numerical example

This section illustrates the modeling, initiation, and consistency checks outlined above. The model used is very similar to the one used in Chapter 9 but is simplified somewhat and uses other parameter values that are better suited for this presentation.

3.3.1 Example model

Assume a simple combined system where the first subsystem, of second order, is a first order lowpass filter applied to a Wiener process and the second subsystem, of first order, represents a separate Wiener process. Hence, using the state space formalism c. f. Section 3.1 and (3.1), we assume in this example a state transition matrix $\mathbf{A}$ and gain matrix $\mathbf{G}$ defined as

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\alpha & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.31}$$

$$\mathbf{G} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.32}$$

where α determines the time constant of the lowpass filter in the first subsystem.

The incremental covariance matrix $\mathbf{\Lambda}(t)\,dt$ of the Wiener increment vector $\mathbf{dw}(t)$ is assumed to be time-invariant and diagonal as described as

$$\mathbf{\Lambda}(t)\,dt \triangleq E(\mathbf{dw}(t)\mathbf{dw}^T(t)) = \begin{pmatrix} \sigma^2_{\lambda_1} & 0 \\ 0 & \sigma^2_{\lambda_2} \end{pmatrix} dt \ . \tag{3.33}$$

Sampling this continuous-time process model with the sampling period T, c. f. Section 3.1.2, the discrete-time process model (3.8) is obtained, c. f. Appendix 3.B and Appendix 3.C, as

$$\mathbf{F} = \begin{pmatrix} 1 & \alpha^{-1}(1-e^{-\alpha T}) & 0 \\ 0 & e^{-\alpha T} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.34}$$

$$\mathbf{Q}_1 = \mathbf{Q}_1(k) = \begin{pmatrix} q^{(1)}_{11} & q^{(1)}_{12} & 0 \\ q^{(1)}_{21} & q^{(1)}_{22} & 0 \\ 0 & 0 & q^{(1)}_{33} \end{pmatrix} \mathbf{\Sigma}_0 \ . \tag{3.35}$$

where the elements of $\mathbf{Q}_1$ are specified by (3.87)-(3.92).

Finally, a simple measurement model is assumed where the first state of each submodel is measured in discrete time with additive noise having a constant covariance matrix. The process noise and the measurement noise are assumed independent. This is summarized by

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.36}$$

$$\mathbf{Q}_2 = \mathbf{Q}_2(k) = \begin{pmatrix} q^{(2)}_{11} & 0 \\ 0 & q^{(2)}_{22} \end{pmatrix} \tag{3.37}$$

$$\mathbf{Q}_{12} = \mathbf{Q}_{12}(k) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \ . \tag{3.38}$$

Moreover, we assume that the initial states $\mathbf{x}(k_0 - 1)$ are completely unknown or, equivalently, that the eigenvalues of the initial covariance matrix $\mathbf{P}(k_0 - 1)$ are infinite.

Unless stated otherwise, the simulations were performed with 1000 Monte Carlo runs, with a sampling period of 4 s. The other model parameters used in this section are summarized in Table 3.1.

Table 3.1: Parameters used in the simulations unless stated otherwise

Description	*Parameter*		*Value*
Number of runs	N		1000
Sample period	T	[s]	4
Integration constant (true)	α_0	[s^{-1}]	0.1
Integration constant (mod)	α	[s^{-1}]	$1.5\alpha_0$
Process noise variance	$\sigma^2_{\lambda_1}$		0.001
	$\sigma^2_{\lambda_2}$		0.1
Measurement noise variance	$q^{(2)}_{11}$		0.5
	$q^{(2)}_{22}$		1
Significance level	$1 - \vartheta$		0.95
Initial state, mean	$E(x_1(k_0))$		30
	$E(x_2(k_0))$		0
	$E(x_3(k_0))$		8
Initial state, variance	$V(x_1(k_0))$		0.01
	$V(x_2(k_0))$		10^{-6}
	$V(x_3(k_0))$		9

3.3.2 Initialization

Before the Kalman algorithm may be applied to the example model given above, the initial states must be set to their appropriate values using one of the techniques described in Section 3.2.2. For the model considered here, with an invertible state transition matrix $\mathbf{F}$, the initialization is particularly simple and may be estimated directly from data.

In Appendix 3.D, the MMSE estimate of the state vector $\hat{\mathbf{x}}(k_0 + 1|k_0 + 1)$ is derived. Using this estimate and the true model above, the results presented in Table 3.2 are obtained. As shown in the table, the estimate yield the expected MSE.

Table 3.2: MSE of the initial MMSE estimate of the state vector with 95% confidence intervals

Entity	*True Value*	*MMSE Estimate*	
$\hat{x}_1(k_0 + 1\|k_0 + 1)$	0.500	0.485	$[0.457, 0.545]$
$\hat{x}_2(k_0 + 1\|k_0 + 1)$	0.0425	0.0411	$[0.0389, 0.0464]$
$\hat{x}_3(k_0 + 1\|k_0 + 1)$	0.583	0.581	$[0.533, 0.636]$
$\hat{x}_1(k_0 + 2\|k_0 + 1)$	1.65	1.60	$[1.51, 1.80]$
$\hat{x}_2(k_0 + 2\|k_0 + 1)$	0.0219	0.0218	$[0.0200, 0.0238]$
$\hat{x}_3(k_0 + 2\|k_0 + 1)$	0.983	1.00	$[0.899, 1.07]$

3.3.3 Consistency

Now, having initialized the Kalman filter properly, it is important to check that it is consistent with the data, i.e., that the innovations are zero mean and white, that the filtered estimates are unbiased, and that the errors in the filtered estimates actually correspond to the calculated covariance matrices, see Section 3.2.3 for details. Note that the filter must be properly initialized in order to provide consistent estimations.

With the model described above, simulation with 1000 Monte Carlo runs were performed with MMSE initialization where an example realization is shown in Figure 3.1. Despite the α parameter being wrong, see Table 3.1, it is difficult to see that the performance of the Kalman filter is suboptimal.

Using the consistency tests described in Section 3.2.3, however, the suboptimal performance is easily recognized as shown in Figure 3.2. In the figure the tests are presented with a (95%) confidence interval.

Figure 3.3 shows the prediction MSE as well as the initial state estimate MSE from the two simulations. As shown, the prediction MSE of the first subsystem is first large because of the difficulty in estimating the derivate in the first subsystem correctly, but decreases quickly to a low level. Surprisingly, the initial prediction MSE is larger if the correct model is used.

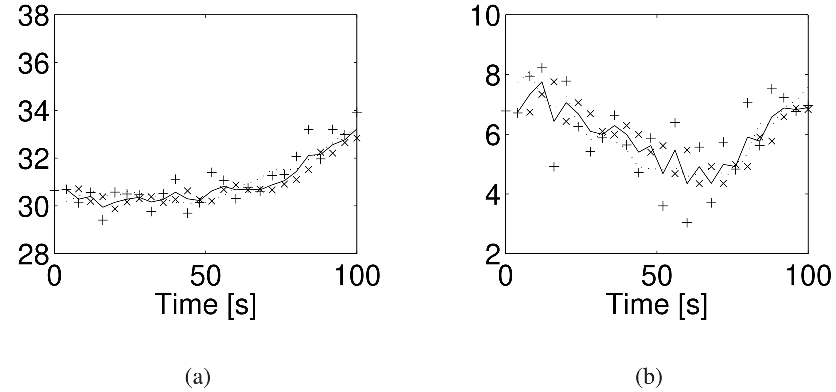


Figure 3.1: Sample results from one simulation run of the example model with erroneous α parameter. The diagrams show the true output $\mathbf{h}_i\mathbf{x}(k)$ (dotted), the actual measured output $y_i(k)$ ($+$), the filtered output $\mathbf{h}_i\hat{\mathbf{x}}(k|k)$ (solid), and the predicted output $\mathbf{h}_i\hat{\mathbf{x}}(k|k-1)$ ($\times$), where $\mathbf{h}_i$ is the ith row in $\mathbf{H}$. In each diagram, we see (a) element one of the measurement using the MMSE initialization, (b) element two of the measurement using the MMSE initialization,

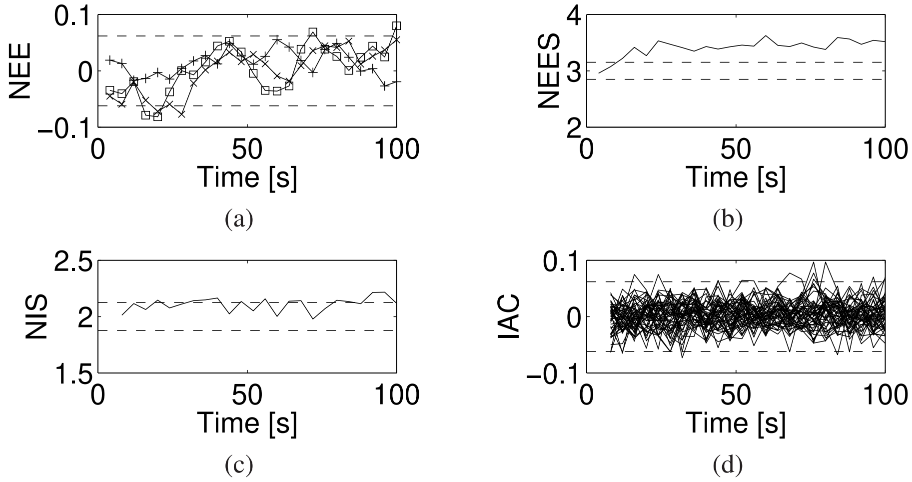
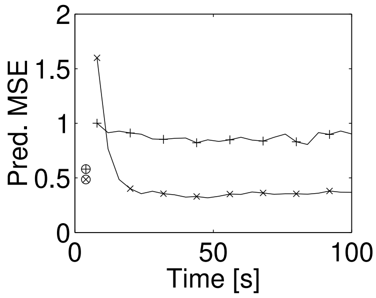
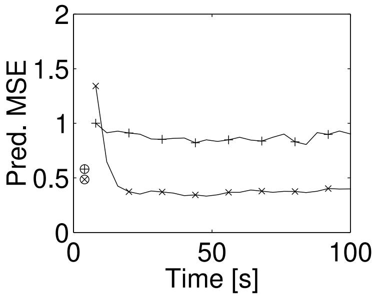


Figure 3.2: Consistency tests (a) average NEE test, (b) average NEES test, (c) average NIS test, and (d) sample IAC test.



(a)



(b)

Figure 3.3: Prediction MSE (solid) and initial state estimate error ($\circ$) of the measurable states $\hat{x}_1(\cdot|k)$ ($\times$) and $\hat{x}_3(\cdot|k)$ ($+$). Each diagram shows (a) MMSE initialization with the correct model and (b) MMSE initialization with the erroneous model.

3.4 Extensions

Paramount to the success of the Kalman filter is the assumption of a known linear system model. Many practical applications, however, do not comply with this requirement. In some cases, it is possible to derive or *design* a model that corresponds well with the actual system. This step might involve techniques such as identification of the system at hand and/or linearization of a given nonlinear system model.

In other cases, a linear model with known parameters is difficult or even impossible to achieve due to practical considerations. In the simulations, for example, the assumed model contains the model parameter α. If this parameter is unknown and the performance drops when using a fixed assumed value that is too large, we may want to extend the estimator to include estimation of this parameter as well. The problem with this solution is that the resulting model is nonlinear and cannot be used in the Kalman filter algorithm.

In the literature, there are a number of approaches to the problem of model nonlinearities and unknown parameters of which we mention three here:

Extended Kalman Filter The idea of the extended Kalman filter (EKF) [12] [76] is to linearize a nonlinear model through a Taylor series expansion and use the linearized model for the calculation of the filter gain and the time update of the covariance matrix. The main drawbacks with this approach are that the EKF is not an optimal filter and that there is no guarantee that it provides any useful estimates.

Multiple Model Approach Using a sufficiently large bank of models, the idea of the multiple model (MM) approach [12] is to span a model space in which the correct model is assumed to reside. The produced estimate and its covariance matrix are weighted sums of the corresponding entities of the individual models. If the correct model is among the set of the models considered and the same model has been in effect all the time, this estimate and its covariance matrix are correct.

Interacting Multiple Model Algorithm The idea of the interacting multiple model (IMM) algorithm [12] [56] is to handle the cases where the active model changes over time while limiting the storage requirements. This method is suboptimal, however, because an optimal approach requires exponentially increasing storage. The main feature of the IMM algorithm is its finite storage requirement while having approximately the same computational complexity as the MM approach.

3.5 Concluding remarks

In this chapter, a review of the Kalman filter was presented in the context of prediction. The structure of the required process and measurement model were presented where the discrete-time process model was derived from a continuous-time model. Techniques on estimating initial values from the data and evaluating the consistency of the Kalman filter were also discussed.

The often neglected issue of proper initialization was pursued and a simple method for initialization the Kalman filter using an MMSE estimate when the initial values are totally unknown was presented. A more general method that may perform better than the MMSE estimate when the distribution of the initial values are partially known was also briefly addressed.

In high-performance applications where an optimal Kalman filter is required, the initialization and consistency tools described in this chapter are of vital importance, if not crucial, to the result.

Appendix 3.A Kalman filter initialization from data

Assuming that the first measurements are available from time k_0, the objective is to find an MMSE state estimate $\hat{\mathbf{x}}(k_0+l-1|k_0+l-1)$ and its covariance matrix $\mathbf{P}(k_0+l-1|k_0+l-1)$.

If the system is observable from the measurements $\mathbf{y}(k)$, it is then possible to find the initial state estimate $\hat{\mathbf{x}}(k_0|k_0+l-1)$ and its covariance matrix $\mathbf{P}(k_0|k_0+l-1)$ using the l first measurements with l sufficiently large. By applying the fixed interval smoothing algorithm [4] with the interval length l, the sought estimates can be obtained. If, however, the inverse transition matrix $\mathbf{F}^{-1}$ exists, the desired estimates can be obtained directly, c. f. (3.74) and (3.75), as shown below.

The measurements $\mathbf{y}(k)$ for $k = k_0, k_0+1, \ldots, k_0+l-1$ are given by (3.8) and (3.13)

$$\begin{aligned}
\mathbf{y}(k_0) &= \mathbf{H}\mathbf{x}(k_0) + \mathbf{v}_2(k_0) && (3.39)\\
\mathbf{y}(k_0+1) &= \mathbf{H}\mathbf{x}(k_0+1) + \mathbf{v}_2(k_0+1) &&\\
&= \mathbf{H}\mathbf{F}\mathbf{x}(k_0) + \mathbf{H}\mathbf{v}_1(k_0) + \mathbf{v}_2(k_0+1) && (3.40)\\
\mathbf{y}(k_0+2) &= \mathbf{H}\mathbf{x}(k_0+2) + \mathbf{v}_2(k_0+2) &&\\
&= \mathbf{H}\mathbf{F}^2\mathbf{x}(k_0) + \mathbf{H}(\mathbf{F}\mathbf{v}_1(k_0) + \mathbf{v}_1(k_0+1)) + \mathbf{v}_2(k_0+2) && (3.41)\\
&\ \ \vdots &&\\
\mathbf{y}(k_0+l-1) &= \mathbf{H}\mathbf{x}(k_0+l-1) + \mathbf{v}_2(k_0+l-1) && (3.42)\\
&= \mathbf{H}\mathbf{F}^{l-1}\mathbf{x}(k_0) &&\\
&\quad + \mathbf{H}\sum_{i=1}^{l-1}\mathbf{F}^{l-1-i}\mathbf{v}_1(k_0+i-1) + \mathbf{v}_2(k_0+l-1)\ . && (3.43)
\end{aligned}$$

This is a system of ln_y equations in the n_x unknowns $\mathbf{x}(k_0)$ and may be written as

$$\mathcal{Y} = \mathcal{O}_l\mathbf{x}(k_0) + \mathcal{E} \qquad (3.44)$$

where the measurement set vector $\mathcal{Y}$, the partial observability matrix $\mathcal{O}_l$ [50], and the residual $\mathcal{E}$ are defined as

$$\mathcal{Y} \triangleq \begin{pmatrix} \mathbf{y}(k_0) \\ \mathbf{y}(k_0+1) \\ \vdots \\ \mathbf{y}(k_0+l-1) \end{pmatrix} \tag{3.45}$$

$$\mathcal{O}_l \triangleq \begin{pmatrix} \mathbf{H} \\ \mathbf{HF} \\ \vdots \\ \mathbf{HF}^{l-1} \end{pmatrix} \tag{3.46}$$

$$\mathcal{E} \triangleq \mathcal{L}\mathcal{V}_1 + \mathcal{V}_2 \tag{3.47}$$

where

$$\mathcal{L} \triangleq \begin{pmatrix} \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{HF} & \mathbf{H} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{HF}^{l-2} & \mathbf{HF}^{l-3} & \cdots & \mathbf{H} & \mathbf{0} \end{pmatrix} \tag{3.48}$$

$$\mathcal{V}_1 \triangleq \begin{pmatrix} \mathbf{v}_1(k_0) \\ \mathbf{v}_1(k_0+1) \\ \vdots \\ \mathbf{v}_1(k_0+l-1) \end{pmatrix} \tag{3.49}$$

$$\mathcal{V}_2 \triangleq \begin{pmatrix} \mathbf{v}_2(k_0) \\ \mathbf{v}_2(k_0+1) \\ \vdots \\ \mathbf{v}_2(k_0+l-1) \end{pmatrix} . \tag{3.50}$$

For (3.44) to be solvable, $\mathcal{O}_l$ must have full rank. A sufficient condition for this is that the system is observable from $\mathbf{y}(k)$ and that $l \geq l_0$ where l_0 is defined as

$$l_0 \triangleq \inf\{l : \operatorname{rank}\mathcal{O}_l = n_x\} . \tag{3.51}$$

Through *prewhitening* of the residual $\mathcal{E}$, (3.44) can be reformulated as

$$\bar{\mathcal{Y}} = \mathcal{M}\mathbf{x}(k_0) + \omega \tag{3.52}$$

where $\bar{\mathcal{Y}}$, $\mathcal{M}$, and ω are defined as[4]

$$\begin{aligned}
\bar{\mathcal{Y}} &\triangleq \mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{Y} && (3.53)\\
\mathcal{M} &\triangleq \mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l && (3.54)\\
\omega &\triangleq \mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{E}\ . && (3.55)
\end{aligned}$$

where $\mathcal{R}_{\mathcal{E}}$ is the covariance matrix of the residual $\mathcal{E}$ defined as

$$\mathcal{R}_{\mathcal{E}} \triangleq E(\mathcal{E}\mathcal{E}^T) = \mathcal{L}\mathcal{Q}_1\mathcal{L}^T + \mathcal{L}\mathcal{Q}_{12} + \mathcal{Q}_{12}^T\mathcal{L}^T + \mathcal{Q}_2 \qquad (3.56)$$

and

$$\begin{aligned}
\mathcal{Q}_1 &\triangleq E(\mathcal{V}_1\mathcal{V}_1^T) = \begin{pmatrix} \mathbf{Q}_1(k_0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Q}_1(k_0+l-1) \end{pmatrix} && (3.57)\\
\mathcal{Q}_{12} &\triangleq E(\mathcal{V}_1\mathcal{V}_2^T) = \begin{pmatrix} \mathbf{Q}_{12}(k_0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Q}_{12}(k_0+l-1) \end{pmatrix} && (3.58)\\
\mathcal{Q}_2 &\triangleq E(\mathcal{V}_2\mathcal{V}_2^T) = \begin{pmatrix} \mathbf{Q}_2(k_0) & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Q}_2(k_0+l-1) \end{pmatrix}\ . && (3.59)
\end{aligned}$$

Now, the MMSE estimate of $\mathbf{x}(k_0)$ is readily computed by solving (3.52) and, hence, (3.44). In the general case $\mathcal{M}$ is not square resulting in the solution

$$\begin{aligned}
\hat{\mathbf{x}}(k_0|k_0+l-1) &\triangleq E\left(\mathbf{x}(k_0)|k_0+l-1\right) && (3.60)\\
&= E\left((\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T(\bar{\mathcal{Y}}-\omega)|k_0+l-1\right)\\
&= (\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T\bar{\mathcal{Y}}\\
&= ((\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l)^T\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l)^{-1}(\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l)^T\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{Y}\\
&= (\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{O}_l)^{-1}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{Y}\ . && (3.61)
\end{aligned}$$

[4]Note that the covariance matrix of ω is the identity matrix due to the prewhitening with $\mathcal{R}_{\mathcal{E}}^{-1/2}$.

With $\mathbf{x}(k_0)$ as given by (3.52), the covariance matrix $\mathbf{P}(k_0|k_0+l-1)$ is given by

$$\begin{aligned}\mathbf{P}(k_0|k_0+l-1) &\stackrel{\triangle}{=} E(\tilde{\mathbf{x}}(k_0|k_0+l-1)\tilde{\mathbf{x}}^T(k_0|k_0+l-1)) \quad (3.62)\\ &= (\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T E(\omega\omega^T)\mathcal{M}(\mathcal{M}^T\mathcal{M})^{-1}\\ &= (\mathcal{M}^T\mathcal{M})^{-1}\mathcal{M}^T\mathcal{M}(\mathcal{M}^T\mathcal{M})^{-1}\\ &= (\mathcal{M}^T\mathcal{M})^{-1}\\ &= ((\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l)^T\mathcal{R}_{\mathcal{E}}^{-1/2}\mathcal{O}_l)^{-1}\\ &= (\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{O}_l)^{-1} \quad (3.63)\end{aligned}$$

where

$$\tilde{\mathbf{x}}(k_0|k_0+l-1) \stackrel{\triangle}{=} \mathbf{x}(k_0)-\hat{\mathbf{x}}(k_0|k_0+l-1) \ . \quad (3.64)$$

In the special case where $\mathcal{O}_l$ is square (3.60) reduces to

$$\begin{aligned}\hat{\mathbf{x}}(k_0|k_0+l-1) &= (\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{O}_l)^{-1}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{Y} \quad (3.65)\\ &= \mathcal{O}_l^{-1}\mathcal{R}_{\mathcal{E}}\mathcal{O}_l^{-T}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{Y}\\ &= \mathcal{O}_l^{-1}\mathcal{Y} \quad (3.66)\end{aligned}$$

with the covariance matrix

$$\begin{aligned}\mathbf{P}(k_0|k_0+l-1) &= (\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}}^{-1}\mathcal{O}_l)^{-1} \quad (3.67)\\ &= \mathcal{O}_l^{-1}\mathcal{R}_{\mathcal{E}}\mathcal{O}_l^{-T}\\ &= \mathcal{O}_l^{-1}(\mathcal{L}\mathcal{Q}_1\mathcal{L}^T+\mathcal{L}\mathcal{Q}_{12}+\mathcal{Q}_{12}^T\mathcal{L}^T+\mathcal{Q}_2)\mathcal{O}_l^{-T} \ . \quad (3.68)\end{aligned}$$

Remark A necessary condition for the solution (3.60) is that $\mathcal{O}_l$ and $\mathcal{R}_{\mathcal{E}}$ are full rank. For $l \geq l_0$ this is satisified for $\mathcal{O}_l$ but for $\mathcal{R}_{\mathcal{E}}$ it is required that $\mathcal{Q}_2$ is full rank because $\mathcal{L}$ is not.

Remark If $\mathcal{O}_l$ is square, however, the solution is then given by (3.65) and the only necessary condition is $l \geq l_0$ which is easily fulfilled for observable systems.

Applying the fixed interval smoothing algorithm [4] using $\hat{\mathbf{x}}(k_0|k_0+l-1)$ and $\mathbf{P}(k_0|k_0+l-1)$ as initial estimates, the sought entities $\hat{\mathbf{x}}(k_0+l-1|k_0+l-1)$ and $\mathbf{P}(k_0+l-1|k_0+l-1)$ are obtained.

If $\mathbf{F}^{-1}$ exist, however, the sought entities can be obtained directly. Combining (3.42) and (3.43), we may rewrite (3.44) to yield

$$\begin{aligned}\mathcal{Y} &= \mathcal{O}_l\mathbf{F}^{-(l-1)}\left(\mathbf{x}(k_0+l-1)-\sum_{i=1}^{l-1}\mathbf{F}^{l-1-i}\mathbf{v}_1(k_0+i-1)\right)+\mathcal{E}\\ &= \mathcal{O}_l\mathbf{F}^{-(l-1)}\mathbf{x}(k_0+l-1)-\mathcal{O}_l\mathcal{F}\mathcal{V}_1+\mathcal{E} \quad (3.69)\end{aligned}$$

where

$$\mathcal{F} \triangleq \begin{pmatrix} \mathbf{F}^{-1} & \mathbf{F}^{-2} & \cdots & \mathbf{F}^{-l+1} & \mathbf{0} \end{pmatrix} . \tag{3.70}$$

From (3.47) and (3.69) we can define a new residual $\mathcal{E}'$

$$\mathcal{E}' \triangleq \mathcal{U}\mathcal{V}_1 + \mathcal{V}_2 \tag{3.71}$$

where

$$\mathcal{U} \triangleq \mathcal{L} - \mathcal{O}_l\mathcal{F} \tag{3.72}$$

$$= -\begin{pmatrix} \mathbf{HF}^{-1} & \mathbf{HF}^{-2} & \cdots & \mathbf{HF}^{-l+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{HF}^{-1} & \cdots & \mathbf{HF}^{-l+2} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{HF}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix} . \tag{3.73}$$

By proceeding analogously as when deriving $\hat{\mathbf{x}}(k_0|k_0+l-1)$ in (3.60) and $\mathbf{P}(k_0|k_0+l-1)$ in (3.62), the MMSE estimate of $\mathbf{x}(k_0+l-1)$ is

$$\hat{\mathbf{x}}(k_0+l-1|k_0+l-1) = \mathbf{F}^{l-1}(\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{O}_l)^{-1}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{Y} \tag{3.74}$$

with the covariance matrix

$$\mathbf{P}(k_0+l-1|k_0+l-1) = \mathbf{F}^{l-1}(\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{O}_l)^{-1}(\mathbf{F}^{l-1})^T \tag{3.75}$$

where $\mathcal{R}_{\mathcal{E}'}$ is the covariance matrix of the residual $\mathcal{E}'$.

In the special case where $\mathcal{O}_l$ is square (3.74) reduces to

$$\begin{aligned} \hat{\mathbf{x}}(k_0+l-1|k_0+l-1) &= \mathbf{F}^{l-1}(\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{O}_l)^{-1}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{Y} && (3.76) \\ &= \mathbf{F}^{l-1}\mathcal{O}_l^{-1}\mathcal{R}_{\mathcal{E}'}\mathcal{O}_l^{-T}\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{Y} \\ &= \mathbf{F}^{l-1}\mathcal{O}_l^{-1}\mathcal{Y} && (3.77) \\ &\equiv \mathbf{F}^{l-1}\hat{\mathbf{x}}(k_0|k_0+l-1) && (3.78) \end{aligned}$$

with the covariance matrix

$$\begin{aligned} \mathbf{P}(k_0+l-1|k_0+l-1) &= \mathbf{F}^{l-1}(\mathcal{O}_l^T\mathcal{R}_{\mathcal{E}'}^{-1}\mathcal{O}_l)^{-1}(\mathbf{F}^{l-1})^T && (3.79) \\ &= \mathbf{F}^{l-1}\mathcal{O}_l^{-1}\mathcal{R}_{\mathcal{E}'}\mathcal{O}_l^{-T}(\mathbf{F}^{l-1})^T \end{aligned}$$

$$= \mathbf{F}^{l-1}\mathcal{O}_l^{-1}(\mathcal{U}\mathcal{Q}_1\mathcal{U}^T + \mathcal{U}\mathcal{Q}_{12} + \mathcal{Q}_{12}^T\mathcal{U}^T + \mathcal{Q}_2)\mathcal{O}_l^{-T}(\mathbf{F}^{l-1})^T . \tag{3.80}$$

Remark Note that with $\mathcal{O}_l$ square, the estimate $\hat{\mathbf{x}}(k_0+l-1|k_0+l-1)$ is directly given by $\hat{\mathbf{x}}(k_0|k_0+l-1)$ and the state transition matrix $\mathbf{F}$ only, c. f. (3.66) and (3.77). This will be true also when $\mathbf{F}$ is not invertible.

Remark If $\mathcal{O}_l$ is not square, however, the contribution of each measurement $\mathbf{y}(k)$ to the estimate $\hat{\mathbf{x}}(\cdot|k_0+l-1)$ is weighted by the noise covariance matrices, the mixing matrix $\mathbf{H}$, and the state transition matrix $\mathbf{F}$.

Appendix 3.B Derivation of the discrete-time state transition matrix

To calculate the discrete-time state transition matrix $e^{\mathbf{A}T}$ from the continuous-time counterpart $\mathbf{A}$, several methods may be used, e.g., series expansion or Laplace transform [9]. With the continuous-time state transition matrix in (3.31)

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\alpha & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.81}$$

a series expansion is suitable to wit

$$\begin{aligned} e^{\mathbf{A}T} &= \mathbf{I} + \mathbf{A}T + \frac{1}{2!}(\mathbf{A}T)^2 + \dots \qquad (3.82) \\ &= \begin{pmatrix} 1 & T - \frac{1}{2}\alpha T^2 + \frac{1}{6}\alpha^2 T^3 - \dots & 0 \\ 0 & 1 - \alpha T + \frac{1}{2}\alpha^2 T^2 - \frac{1}{6}\alpha^3 T^3 + \dots & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & \alpha^{-1}(1-e^{-\alpha T}) & 0 \\ 0 & e^{-\alpha T} & 0 \\ 0 & 0 & 1 \end{pmatrix} . \qquad (3.83) \end{aligned}$$

Appendix 3.C Derivation of the discrete-time process noise covariance matrix

From (3.11) and (3.12) we have for the process noise covariance matrix $\mathbf{Q}_1(k)$

$$\begin{aligned} \mathbf{Q}_1(k) &\triangleq E(\mathbf{v}_1(k)\mathbf{v}_1^T(k)) \\ &= E\int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{G}\,\mathbf{dw}(kT+\tau)\int_0^T \Big(e^{\mathbf{A}(T-\nu)}\mathbf{G}\,\mathbf{dw}(kT+\nu)\Big)^T \end{aligned} \tag{3.84}$$

where $\mathbf{A}$ and $\mathbf{G}$ are defined in (3.31) and (3.32), respectively.

Since $\mathbf{dw}(\nu)$ and $\mathbf{dw}(\tau)$ are independent for $\nu \neq \tau$, we obtain

$$
\begin{aligned}
\mathbf{Q}_1(k) &= E\int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{G}\,\mathbf{dw}(kT+\tau)\mathbf{dw}^T(kT+\tau)\,\mathbf{G}^T e^{\mathbf{A}^T(T-\tau)} \\
&= \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{G}\mathbf{\Lambda}(kT+\tau)\mathbf{G}^T e^{\mathbf{A}^T(T-\tau)}\,d\tau \\
&= \int_0^T e^{\mathbf{A}(T-\tau)} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \sigma_{\lambda_1}^2 & 0 \\ 0 & \sigma_{\lambda_2}^2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau \\
&= \int_0^T e^{\mathbf{A}(T-\tau)} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_{\lambda_1}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_2}^2 \end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau \quad . \qquad (3.85)
\end{aligned}
$$

Setting $a(\tau) = e^{-\alpha(T-\tau)}$ and $b(\tau) = \alpha^{-1}(1 - e^{-\alpha(T-\tau)})$ we get

$$
\begin{aligned}
\mathbf{Q}_1(k) &= \int_0^T \begin{pmatrix} 1 & b(\tau) & 0 \\ 0 & a(\tau) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sigma_{\lambda_1}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_2}^2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ b(\tau) & a(\tau) & 0 \\ 0 & 0 & 1 \end{pmatrix} d\tau \\
&= \int_0^T \begin{pmatrix} b^2(\tau)\sigma_{\lambda_1}^2 & a(\tau)b(\tau)\sigma_{\lambda_1}^2 & 0 \\ a(\tau)b(\tau)\sigma_{\lambda_1}^2 & a^2(\tau)\sigma_{\lambda_1}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_2}^2 \end{pmatrix} d\tau \\
&= \int_0^T \begin{pmatrix} b^2(\tau) & a(\tau)b(\tau) & 0 \\ a(\tau)b(\tau) & a^2(\tau) & 0 \\ 0 & 0 & 1 \end{pmatrix} d\tau\,\mathbf{\Sigma}_0 \qquad (3.86)
\end{aligned}
$$

where the matrix $\mathbf{\Sigma}_0$ is defined as

$$
\mathbf{\Sigma}_0 \triangleq \begin{pmatrix} \sigma_{\lambda_1}^2 & 0 & 0 \\ 0 & \sigma_{\lambda_1}^2 & 0 \\ 0 & 0 & \sigma_{\lambda_2}^2 \end{pmatrix} \quad . \qquad (3.87)
$$

Evaluating the integral in (3.86) we obtain

$$
\mathbf{Q}_1(k) = \begin{pmatrix} q_{11}^{(1)} & q_{12}^{(1)} & 0 \\ q_{21}^{(1)} & q_{22}^{(1)} & 0 \\ 0 & 0 & q_{33}^{(1)} \end{pmatrix} \mathbf{\Sigma}_0 \qquad (3.88)
$$

where

$$
\begin{aligned}
q_{11}^{(1)} &= \frac{1}{2\alpha^3}\left(2\alpha T - 3 + 4e^{-\alpha T} - e^{-2\alpha T}\right) && (3.89)\\
q_{12}^{(1)} &= q_{21}^{(1)} = \frac{1}{2\alpha^2}\left(1 - e^{-\alpha T}\right)^2 && (3.90)\\
q_{22}^{(1)} &= \frac{1}{2\alpha}(1 - e^{-2\alpha T}) && (3.91)\\
q_{33}^{(1)} &= T && (3.92)
\end{aligned}
$$

which concludes the derivation of the discrete-time process noise covariance matrix $\mathbf{Q}_1(k)$.

Appendix 3.D Derivation of the MMSE estimate of the initial state vector

Here, the MMSE estimate of the state vector $\hat{\mathbf{x}}(k_0 + l - 1|k_0 + l - 1)$ is derived for the compound system given by the model (3.31)-(3.38) using the expressions presented in Appendix 3.A with $l = 2$. Deriving $\mathcal{O}_2$ as defined in (3.46) using $b = \alpha^{-1}(1 - e^{-\alpha T})$, we obtain

$$
\mathcal{O}_2 \triangleq \begin{pmatrix} \mathbf{H} \\ \mathbf{HF} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & b & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.93}
$$

which clearly is full rang as required ($\alpha \neq 0$). This means that using two meausurements $\mathbf{y}(k)$, the MMSE estimate may be calculated. Furthermore, the state transition matrix $\mathbf{F}$ is invertible, so the expressions (3.69)-(3.75) may be applied.

CHAPTER 4
Multiple target tracking

IN applications where conditions change over time, there is often a need for some type of tracking, i.e. monitoring the changing conditions. One solution to this type of problems is represented by target tracking techniques, or multiple target tracking (MTT) techniques if several targets (conditions) are to be tracked simultaneously.

Essentially, MTT is a source separation technique where the source signals are decomposed to allow individual analysis of the source characteristics. The MTT techniques may therefore successfully be used in other areas than the ones they first were designed for, e.g. radar surveillance and air traffic control.

In this chapter the general idea of MTT is presented together with three common trackers where the focus is on the multiple hypothesis tracking algorithm because it is later used in the applications in Part II. Moreover, a discussion of the applicational aspects is included.

4.1 Target tracking preliminaries

The perhaps most well-known application of MTT is radar surveillence, for example air traffic control. The purpose is then to monitor the movements of a set of aircrafts by forming a *track* for each aircraft, or *target* in MTT terminology.[1] The track is made up of *observations* where each observation is a description of the target's state (e. g. position) at a particular time instant. The observations are

[1]Because of the close connection between tracks and targets, these words will be used almost synonymously.

collected by periodically scanning the monitored region, analyzing each *scan*, and reporting possible target observations. In conclusion, the MTT system thus periodically scans the region to monitor, collects any observations, and forms target tracks.

In a wider view, the MTT system, like any tracking system for that matter, monitors (in discrete time) how a set of conditions change over time. From a measured signal, possibly non-scalar observations of the status of the monitored conditions (tracks) are derived. The main problem is that which condition (track) a certain observation contains information about, the so-called *observation-to-track association*, is not known a priori. The major complexity of a MTT system is also found in the association algorithms intended to unravel this problem. The problem is very hard, however, and in the literature there are numerous approaches described. In the following sections three of the most common ones are presented.

Before going into the details of the observation-to-track association, the actual derivation of the observations is discussed. For each scan, the set of observations are derived from collected data retrieved by the scanning device using a *detector*. The data may be collected from a single sensor or multiple sensors, but multiple sensors may be treated as a single multidimensional sensor if they are independent of each other. Two sensors measuring height and distance, for example, may be treated as a single twodimensional sensor.

Often, the sequence of observations contains errors that the tracker should handle in a robust way. The most common errors stem from *clutter* and missed detections. Clutter is a collective term of all erroneous observations that the detector reports to the tracker. It is often a major problem because it produces false target observations that place an upper limit on the performance of the tracking system.

The other type of error, missed detections, is often influenced by the noise in the collected data. Clearly, this also sets an upper limit on the performance and the tracking system is required to be robust to these errors.

In this thesis a special MTT problem is presented, which actually is a source separation problem. In this case there is a single sensor and multiple targets. The data contains signals (target echoes) that are detected using a detector described in Chapter 2. The signals are emitted from sources (objects) and the tracker's objective is to discriminate between the detected signals and decide from which source each signal originates.

A block diagram of a general tracker needed for this problem is shown in Figure 4.1. The measured data collected from the sensor is fed to the detector that reports all potential locations of the signals. The reports are sent to the tracker that associates the observations to targets. To facilitate the association process, the tracker may use a predictor that generates a predicted observation for each track formed by the tracker. In a causal tracking system, the predictor uses earlier detec-

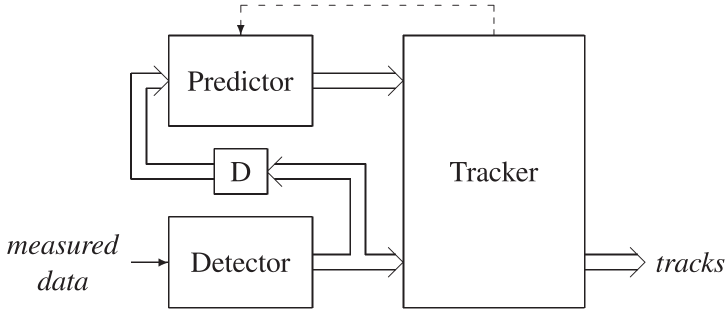


Figure 4.1: A block diagram of an MTT system. Data flow is indicated by solid lines; thin for single data and thick for multiple data. Control flow is indicated by dashed lines.

tions only, which is symbolized by the delay (D) step in the figure. In this way, the association of the observations to targets are guided using a model of the target's characteristics.

Because the tracker forms tracks by associating observations with targets, the tracker block in Figure 4.1 is often called an *association filter* or *correlation filter*. Throughout this presentaion, tracker and association filter is used synonymously. In the following sections the contents of the tracker block is presented for three common types of association filters.

4.2 Nearest neighbor tracking

The simplest association filter possible is the nearest neighbor (NN) association [13]. Each target is associated with the observation closest to the target's predicted observation. This type of tracker therefore needs a distance measure in addition to the predictor. In the simplest case, where the sensor is one dimensional, the distance measure is trivial. When using multi dimensional sensors, the distance measure is often straight-forward to derive if the dimensions of the observations are of the same kind, e.g. spatial position. In other cases there may be some difficulties in deriving a distance measure, but often a suitable measure can be found.

The main advantage of the NN tracker is its simplicity and, hence, its computational speed. Therefore, the NN tracker has been very popular in radar surveillance applications where the computational requirements have been demanding. Due to computational improvements of other trackers and the introduction of more powerful computer systems, the NN tracker is more rarely used in modern MTT systems.

The disadvantages of the NN tracker are poor performance in the presence of false alarms and for closely spaced targets. Moreover, the demand on the predictor is higher compared to other trackers, since it is cruical that the distance between the correct observation and the predicted observation, i.e. the prediction error, is small.

4.3 Joint probabilistic data association tracking

The joint probabilistic data association (JPDA) [11] tracker differs from the NN tracker in several respects. The NN tracker uses a so-called *hard-decision* correlation logic, whereas the JPDA tracker uses a *soft-decision* correlation logic. This means that the JPDA tracker does not pick out *a particular* observation and associates it with a target as the NN tracker does, which is a irrevocable (hard) decision. Instead, the JPDA tracker associates *all* observations with a target and produces a virtual observation that is a mix of the associated observations weighted by their probability of originating from the target in question. Therefore, the JPDA tracker has a good resistance to clutter and missed detections.

The disadvantages of the JPDA tracker is its tendency of coalescing or swapping neighboring tracks and its high computational complexity. The latter is becoming less of a problem with more effective algorithms and more powerful computer systems, however.

Even if the JPDA tracker does not explicitly make use of a predictor, it still needs a target model in order to compute the probabilities of an observation originating from a particular target. Such a model may be interpreted as a predictor in accordance with Figure 4.1.

4.4 Multiple hypothesis tracking

Multiple hypothesis tracking (MHT) [69] is recognized as the theoretically best approach to multitarget tracking problems. In applications with heavy clutter and high *traffic densities*, the performance of MHT is outstanding compared to other methods, e.g. nearest neighbor (NN) correlation or joint probabilistic data association (JPDA) [21]. The brief presentation given in this section is influenced by Blackman and interested readers are referred to his books [13] and [16].

Both MHT and JPDA are Bayesian probabilistic approaches to the tracking problem. The MHT method uses a hard-decision correlation logic, however, and all possible combinations of associating observations with targets are enumerated and ordered according to their probability of being correct. Because of this, the correct partitioning of the collected data are always present and, hence, the MHT tracker outperforms any other method if the memory and computational requirements can be met.

The computational requirements are, however, enormous and increase exponentially with the complexity of the tracking situation. This is the major disadvantage of the MHT tracker and has often limited its use in practical cases, especially in on-line applications.

By arranging the storage of observations in a clever way, limiting the enumeration, and keeping only the most probable partitions, a suboptimal algorithm may be derived. Today, such an algorithm is a good choice for a practical tracking system, owing to the increased power of computing systems.

4.4.1 Overview of the MHT tracker

For each scan in the recorded data, the observations are collected by the tracking system. At a given time and with a given set of observations, there are several plausible ways to combine the observations into tracks. Instead of choosing only the most probable partitioning after each scan, the MHT tracker generates a number of different partionings, so-called *hypotheses*, to be evaluated later when more data are received. Thus, the probability of choosing the correct partitioning of the data into tracks and false alarms is increased.

Because there will, in general, be several hypotheses available, a way of picking the "best" one is needed. With the Bayesian approach, the hypotheses are ranked by their a posteriori probability. Any monotonic function of the probability may, of course, be used to rank the hypothesis. Often, the logarithm is used and scaled in such a way that a partitioning of the data into just false alarms yields the value zero. This ranking function is referred to as the *hypothesis score*.

Similar to the JPDA tracker, the MHT tracker does not explicitly need a predictor. Unless an ad hoc ranking function is used, however, the information needed to derive the score is often sufficient to produce a predictor and, thus, the scoring function may be regarded as a predictor in accordance with Figure 4.1. Moreover, when the hypothesis generation is limited using the gating technique, see Section 4.4.4, a predictor is really needed. Therefore, it is fair to say that the predictor is an important part of the MHT tracking system.

In Figure 4.2 a block diagram of the MHT algorithm is presented. For each trace, the control logic iterates over the current track set. The track currently selected is fed to the predictor where the expected target observation is calculated. A *gate* is formed around the prediction and all observations (in the current trace) that fall outside the gate are discarded, see Section 4.4.4. The current track and the gated observations are used by the track generator to both create updates of the track and to create entirely new tracks. The tracks produced by the track generator are then fed to the hypothesis generator as well as stored for later use in the processing of the observations of the next trace.

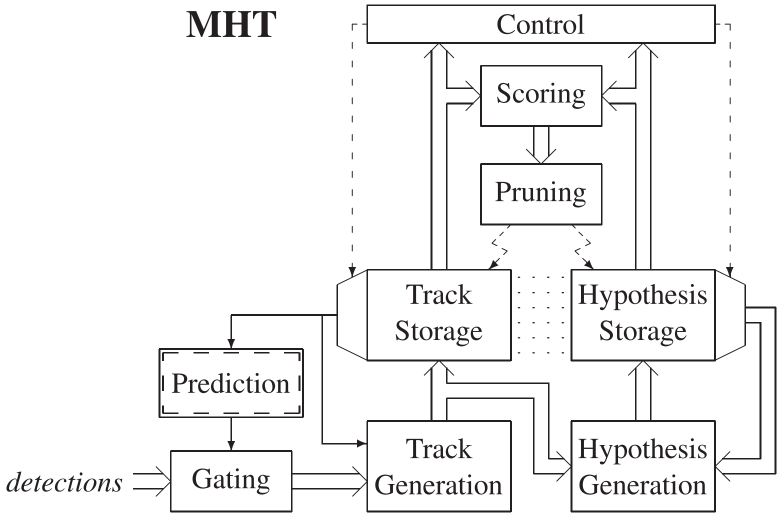


Figure 4.2: A block diagram of the MHT algorithm. Data flows are indicated by solid lines; thin for single data and thick for multiple data. Control flows are indicated by dashed lines whereas other intimate relations are indicated by dotted lines.

At the same time as the current track was selected, the control logic also selected all hypotheses containing that track. These hypotheses are used together with the generated tracks to form updated hypotheses. The generated hypotheses are then stored in the hypothesis storage.

When the track iteration is completed, the scores of all hypotheses are calculated and used to *prune* hypotheses (and tracks) associated with a low score, see Section 4.4.4. The process is then repeated for the observations in the next scan.

Note that the MHT algorithm structure does not depend on a particular predictor. The same algorithm may, in principle, be used for any tracking problem. Only the predictor needs to be changed.

4.4.2 Forming and ranking hypotheses

As mentioned above, the basic idea of the MHT method is to enumerate all possible combinations of observation to track associations. A hypothesis therefore consists of a hypothesized set of tracks containing a hypothesized set of observations. Within each hypothesis, an observation can be associated to at most one track. If an observation is not assigned to any track, it is considered as a false alarm or clutter.

With this strategy, any track may be a part of several hypotheses and an efficient enumeration of all possible partioning combinations is obtained by focusing on the tracks instead of the hypotheses, see [13] for details.

Either way, the enumeration of all possible combinations give rise to a vast number of hypotheses. In order to rank all these and to find the "best" one, a ranking function is needed. Using the Bayesian approach, the ranking function is the a posteriori probabilty of each hypothesis. If the a posteriori probability is a product of several terms, it is beneficial to use the logarithm of it instead, in which case it is called the *score*. Often, the score is scaled in such a way that an all-false-alarm partitioning of the observations yields a score of zero.

Assuming that the tracks are independent of each other, the score of a hypothesis at scan k may be expressed as

$$L(k) = \sum_{i=1}^{n_k} L_i(k) + \tilde{L} \tag{4.1}$$

where $L_i(k)$ is called the *track score*, $\tilde{L}$ is the *residual hypothesis score* due to track termination (see Section 4.4.3), and n_k is the number of targets (tracks) associated with this hypothesis at scan k.

As suggested in [15], the track score can often be calculated recursively as

$$L_i(k) = L_i(k-1) + \Delta L_i(k) \tag{4.2}$$

$$\Delta L_i(k) = \begin{cases} L_i^{(0)}(k), & \text{not updated} \\ L_i^{(1)}(k), & \text{updated} \end{cases} \tag{4.3}$$

where $L_i^{(\cdot)}(k)$ describe the (different) scores that are used depending on whether the track was updated or not. By using (4.1)-(4.3) the hypothesis score may be calculated efficiently and this improves the performance of the tracker.

4.4.3 Track stages

As the tracking proceeds, the tracks may be in any of five possible track stages: potential, tentative, confirmed, deleted, and terminated. The last stage is not used in [13], but is an important addition when the tracking principle is used for discrimination or source separation problems. The presentation also becomes more consistent. Below, the track stages are explained further.

Potential

A single observation not associated to any existing track could either be a false alarm (or clutter) or be the first detection of a new target. While awaiting more data in order to solve this discrepancy, the observation is associated to a so-called *potential* track.

Tentative

As a potential track is assigned an additional observation, there may still be uncertanties whether the assigned observations originate from a true target or if they are a coincidential collection of false alarms. Until the track has been assigned enough observations in order to solve this discrepancy, the track is called a *tentative* track.

Confirmed

When a tentative track has been assigned sufficiently many observations in order to assure that they originate from a true target, the track is called *confirmed*.

Deleted

When a tentative track has been assigned several observations it may be apparent that they cannot originate from any real target. In this case the track is *deleted* and the assigned observations are considered as false alarms.

Terminated

As a confirmed target does not get updated with new observations, the question arises whether the target has disappeared or if its detection has failed. When sufficiently many detections have been missed in order to assure that the target has disappeared, the track is *terminated*. Hence, it is disabled from further updates and its track length is set to the position of the last update. Note that the score of the track still contributes to the hypotheses score. It is therefore beneficial to terminate tracks because short tracks have a higher score than longer un-updated ones.

The problem of track termination is the possibility of premature termination, i.e., there is a chance that the track would have been updated in the next trace if it hadn't been terminated. The track termination must be delayed sufficiently long so that any additional updates still would result in a lower score than if the track is assumed to be ended at its last update.

4.4.4 Hypothesis limiting techniques

Gating

To restrict the number of hypotheses processed by the tracking system, an obvious way would be to avoid generating unlikely hypotheses. As hypotheses are generated by the observation-to-track associations, only the likely ones should be allowed to be introduced.

By adopting the *gating* technique, a *gate* is formed around the predicted observation and only those observations that fall within the gate, i.e., are close enough to the predicted observation, are considered for association to the track in question.

Analogous to the NN tracker, see Section 4.2, a distance measure is needed. Often this is readily given by the predictor. For example, using a Kalman filter, the normalized distance between the observation and the predicted observation may be defined as

$$d^2(k) \quad \triangleq \quad \tilde{\mathbf{y}}^T(k)\mathbf{S}^{-1}(k)\tilde{\mathbf{y}}(k) \tag{4.4}$$

where the residual vector $\tilde{\mathbf{y}}(k)$ is defined according to (3.16) and its inverse covariance matrix $\mathbf{S}^{-1}(k)$ by (3.17).[2]

The gating condition is then simply

$$d^2(k) \leq G \tag{4.5}$$

where the limit G is set to reflect the trade-off between limiting the hypothesis generation and achieving robust performance.

It is highly recommended to use gating in order to achieve a practical tracking implementation. The gating technique is efficient in limiting the number of hypotheses generated while it has a low (negative) impact on the tracking performance.

Clustering

Since the computational complexity grows exponentially with the number of tracks, the most effective way to reduce it is to separate non-interacting hypotheses into independent *clusters* [10] [69] (not shown in Figure 4.2). Instead of solving one large tracking problem, a number of smaller tracking problems are then solved separately for each cluster of hypotheses.

Combined with gating, the *clustering* technique may reduce the computational complexity of the tracker dramatically. For example, two targets with non-overlapping trajectories may be treated separately. Assuming that the tracking of each

[2] The term $d^2(k)$ is actually a squared distance, but, for convenience, it is often referred to as simply a distance.

target generates 3 hypotheses for each scan. If both gating and clustering are used, the two targets are treated separately yielding a total of six $(3 + 3)$ hypotheses. If, however, only gating is used, the result is a total of nine (3×3) hypotheses generated for each scan. If neither is used, the number of hypotheses generated is even larger.

Using only the clustering technique is still useful, but not as effective as when combined with gating. Without gating, all clusters are merged for each new scan because each observation is then assigned to all tracks. Hence, all tracks interact and must be part of the same single cluster. Not until sufficiently many unlikely hypotheses are discarded, in which case the interfering assignments are discarded as well, the two targets may be handled in separate clusters. Therefore, clustering without gating, at best, results in a periodical merging and clustering pattern that deteriorates the performance considerably.

Pruning

As mentioned above, the MHT method defers the evaluation of the hypotheses until more data is received in order to make a better decision. It is however, not desirable to keep unlikely hypotheses for extensive periods of time as it slows down the tracking process and increases the storage requirements.

Thus, unlikely hypotheses are discarded in a process called *pruning* where the low probability hypotheses are deleted. There are several possibilities to choose the set of hypotheses to be removed. In general, the pruning strategies can be divided into two major groups. In the first group, the hypotheses are pruned based on their score only. This is the simplest pruning technique, but the performance is in some cases poor due to sudden changes in tracking decisions.

In the other group, the hypotheses are pruned based on their interrelationships as well as on their score. This technique is often called *root node pruning* [14], because the relations between the hypotheses may be organized into a tree-like structure.

Using root node pruning, the node hypothesis that is the ancestor N scans ago to the currently best hypothesis is identified. This node is called the *root node* and it constitute the new root of the hypothesis tree. All hypotheses that do not belong to this tree are deleted. This way, the tracks in the remaining hypotheses differ only for the N last scans and this pruning method is often termed N*-backscan pruning*.

The root node pruning performs well if N is large enough compared to the dynamics of the tracking situation. Otherwise rapid track changes may be lost. The storage requirements can be exhaustive, though, because the hypotheses tree often grows rapidly. In practice, the value of N is seldom greater than, say, three to five.

Combining

Yet another technique to limit the number of hypotheses is the *combining* technique in which similar tracks and hypotheses are combined to decrease the number of hypotheses managed.

In a classical tracking scenario, when the history is used only to improve the certainties of the present situation, this is a good candidate for hypothesis limitation. For example, two tracks may differ only with respect to their starting points several scans ago. It is, thus, reasonable to believe that this difference is no longer important for the future data processing and that the tracks could be combined into one.

If, however, the tracking is used in a source separation scenario, the combining technique is, clearly, not necessarily a good technique. If combining is used, it is important that the hypothesis score remains unchanged after track combining.

4.5 Concluding remarks

This chapter introduced the concept of multiple target tracking in general and of multiple hypothesis tracking (MHT) in particular.

The MHT method is based on a Bayesian statistical approach to the tracking problem that provides the optimal tracking result if its computational and memory requirements are met, and if the statistical model correctly describes the situation at hand.

Due to the often exponential increase in number of hypotheses generated, a suboptimal method is in general used that limits the number of stored and evaluated hypotheses. This chapter discussed several such variants.

Moreover, by cleverly storing and evaluating the hypotheses the storage and computational requirements may be lowered even further. These improvements in combination with the increased power of computing systems makes the MHT method a good tracking solution even for on-line systems.

CHAPTER 5

Wiener-filter deconvolution

FINDING the input signal to a particular system when only a noisy output signal is available, commonly referred to as *deconvolution*, is a problem often encountered in practice. If the second order properties of the noise and the transfer function of the system are known, then *Wiener filtering* [51] [99] is a well-known method to optimally estimate the input signal in a mean square sense.

Wiener filters may be designed to estimate any signal related in a known way to a measurement. The only requirements are that the underlying dynamic system is linear, that both the signal of interest and the measurements are stationary and correlated, and that their *cross spectrum* is known [76, pp. 183-189].

The Wiener filter may be applied to the measurement signal in the time domain or in the frequency domain. For our purposes in Chapter 8, we prefer to use off-line data in a batch approach and to apply the Wiener filter in the frequency domain, because non-causal, sampling-frequency independent filters are easier to formulate in this way.

The Wiener filter design is considered here for discrete-time systems only. Owing to the choice of working in the frequency domain, a generalization to continuous-time systems is simple to derive, if required. In particular, we consider a signal consisting of impulses with finite energy in additive noise with finite power because this is what is encountered in the application presented in Chapter 8.

In this chapter the basis of Wiener filtering is illustrated. This includes a brief analysis of the problem that arises when mixing energy and effect signals. Our approach to this problem as well as other implementation issues are addressed followed by a simulation where a compound signal is deconvolved in order to discern the individual components.

5.1 The discrete-time Wiener filter

The objective of the discrete-time Wiener filter design is to find the linear, time-invariant filter $G(q)$ that minimizes the mean square error (MSE) criterion

$$J \stackrel{\triangle}{=} E(||s(n) - G(q)y(n)||^2) \tag{5.1}$$

where q is the forward shift operator and $s(n)$ is the signal to be estimated from the measured signal $y(n)$.

5.1.1 The unrealizable Wiener filter

The filter $G(q)$ that minimizes the criterion in (5.1) consitutes the *optimal (linear) mean square estimator* of $s(n)$ given $y(n)$. It's Z-transform $G(\zeta)$ is given by

$$G(\zeta) = \phi_{sy}(\zeta)\phi_y^{-1}(\zeta) \tag{5.2}$$

where $\phi_{sy}(\zeta)$ is the cross spectrum between $s(n)$ and $y(n)$, while $\phi_y(\zeta)$ is the spectrum of the measured signal $y(n)$ [76]. This filter is, however, non-causal in that it uses future data to calculate the estimate of $s(n)$ at a particular time instant. Its impulse response is in general infinite both forward and backward in time, and it is therefore often referred to as the *unrealizable Wiener filter*.

5.1.2 The realizable Wiener filter

In applications where causality is required, an MSE-optimal estimator may be derived under a causality constraint. The resulting *realizable Wiener filter* will have lower performance (higher MSE) than the theoretical performance obtained by the non-causal estimator.

For non-causal filters where the magnitude of the impulse response decays sufficiently fast, an appropriate filter can be obtained through truncation and time shifting of the impulse response.[1] The performance of the resulting truncated approximate Wiener filter is often comparable with that of the unrealizable filter if the truncation point is carefully selected.

A corresponding *processing delay*, or *smoothing lag* can also be included in a Wiener estimator derived under causality constraints [2] [90]. The MSE performance of such an estimate, $\hat{s}(n-n_s|n)$, approaches that of the unrealizable Wiener filter as the processing delay n_s is allowed to increase.

[1] Note that this estimator is not identical to the Wiener filter derived under causality constraints.

5.1.3 Deconvolution in the time domain

With an appropriate Wiener filter $G(q)$ available, the estimate of $s(n)$ is simply

$$\hat{s}(n - n_s|n) \quad = \quad G(q)y(n) \tag{5.3}$$

where $n_s \geq 0$ is the processing delay introduced due to the required causality of the filter $G(q)$.

5.1.4 Deconvolution in the frequency domain

A batch of collected data may be transformed into the frequency domain, where the Wiener filter can be applied off-line. The Wiener estimate of $S(\zeta)$ is then given by

$$\hat{S}(\zeta) \quad = \quad G(\zeta)Y(\zeta) \tag{5.4}$$

where $\hat{S}(\zeta)$ and $Y(\zeta)$ are the Z-transforms of $\hat{s}(n)$ and $y(n)$, respectively, while $G(\zeta)$ is given by (5.2).

This batch approach has similarities with the truncation and time shifting approach to the non-causality problem, see Section 5.1.2. The processing delay introduced by the batch approach is, however, not a constant; it is largest for past data and is virtually zero for the most recent data within the batch.

5.2 Implementation issues

To design and apply the Wiener filter in the frequency domain, it is sufficient to consider each spectrum on the unit circle only, the *spectral density*. The discrete-time Fourier transform (DFT) can therefore be used for the transformation between the time and frequency domains. Use of the FFT algorithm enables the design of a computationally fast implementation of the Wiener filtering.

5.2.1 The discrete-time Fourier transform

Discrete-time Fourier transforms (DFTs) presented in the literature are often without physical units as well as being a sampling-frequency dependent, scaled version of the continuous-time counterpart. In this thesis, both these aspects are very important and we therefore present a DFT below that has the appropriate units and also directly corresponds to the continuous-time counterpart.

From calculus we know that for a particular signal $x_c(t)$ measured in [mV] with t being the time in [ms], the following holds

$$x_c(t) = \int_{-\infty}^{\infty} X_c(f) e^{i2\pi ft}\, df \quad \text{[mV]} \tag{5.5}$$

$$X_c(f) = \int_{-\infty}^{\infty} x_c(t) e^{-i2\pi ft}\, dt \quad \text{[mV/kHz]} \tag{5.6}$$

where f is the frequency in [kHz]. Then $X_c(f)$ is called the *Fourier transform* of $x_c(t)$ and $x_c(t)$ is called the *inverse Fourier transform* of $X_c(f)$.

If the Nyquist criterion is fulfilled for a signal (batch) of length T [ms], the continuous-time Fourier transform may then be approximated using $N = \frac{T}{T_s}$ discrete-time samples $x_c(nT_s), n = 0, 1, 2, \ldots, N-1$ where T_s is the sampling period. This may be summarized as

$$\hat{x}_c(t) \approx \sum_{n=0}^{N-1} X(n/T) e^{i2\pi tn/T} \frac{1}{T} \quad \text{[mV]} \tag{5.7}$$

$$\hat{X}_c(f) \approx X(f) = \sum_{n=0}^{N-1} \mathcal{T}_N(n) x_c(nT_s) e^{-i2\pi fnT_s}\, T_s \quad \text{[mV/kHz]} \tag{5.8}$$

where $\mathcal{T}_N(n)$ is an N-point Hanning window [48, pp. 146-148] normalized to unit power. This window may, however, be omitted for signal waveforms completely accommodated within the selected data window (batch).

5.2.2 Filter implementation

Using the FFT algorithm, a computationally fast approximate Wiener filter was implemented as follows ($\Omega \triangleq e^{i2\pi f/f_s}$):

1. *Transform to the frequency domain* – Use (5.8) to transform the measured signal $y(n)$ and obtain $Y(\Omega)$ in the frequency domain.

2. *Apply the Wiener filter* – Create $\hat{S}(\Omega)$ in the frequency domain by multiplying the Wiener filter $G(\Omega)$ with the signal $Y(\Omega)$.

3. *Inverse transform to the time domain* – Use (5.7) to transform $\hat{S}(\Omega)$ back to the time domain and obtain the desired signal estimate $\hat{s}(n)$.

This procedure is summarized by the expression

$$\hat{s}(n) = \mathcal{F}^{-1}\left\{G(\Omega)\mathcal{F}\left\{y(n)\right\}\right\} \tag{5.9}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the (discrete-time) Fourier transform and the inverse (discrete-time) Fourier transform, respectively. These are implemented using the FFT algorithm in accordance with (5.8) and (5.7) where the vectors containing the samples are padded with zeros to their double length. The latter operation avoids circular convolution.

5.3 Numerical examples

This section demonstrates Wiener filtering through a numerical example and also discusses the problem of tuning a Wiener filter when the signal consists of a single pulse with a finite energy.

5.3.1 Measurement model

In the Wiener filter design, the spectral density of the measured signal and the cross spectral density between the measured signal and the signal to be estimated are needed. Here we present the measurement model from which these entities may be derived.

Assume the measurement system may be described by

$$y(n) = H(q)s(n) + v(n) \quad [\text{mV}] \tag{5.10}$$

$$s(n) = F(q)x(n) \tag{5.11}$$

$$v(n) = \frac{C(q)}{D(q)}e(n) \quad [\text{mV}] \tag{5.12}$$

where, as above, $y(n)$ is the measured signal and $s(n)$ is the signal to estimate. Moreover, $F(\zeta)$ and $H(\zeta)$ are transfer functions that describe how to generate $s(n)$ from the input $x(n)$ and $y(n)$ from $s(n)$, respectively. The entities $C(\zeta)$ and $D(\zeta)$ are polynomials that define the additive measurement noise $v(n)$ where $e(n)$ represents white, zero-mean Gaussian noise.

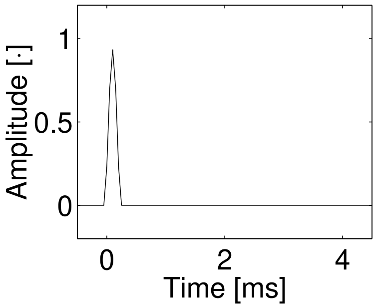
The model (5.10) may be transformed to the frequency domain

$$Y(\Omega) = H(\Omega)S(\Omega) + V(\Omega) \quad [\text{mV ms}] \tag{5.13}$$

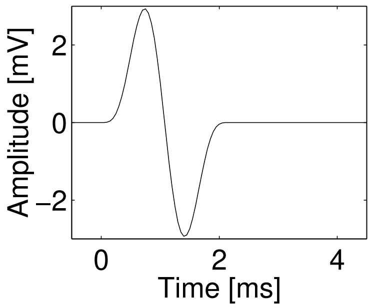
$$S(\Omega) = F(\Omega)X(\Omega) \tag{5.14}$$

$$V(\Omega) = \frac{C(\Omega)}{D(\Omega)}\sqrt{\eta T} \quad [\text{mV ms}] \tag{5.15}$$

where $Y(\Omega)$ is the DFT of $y(n)$, $S(\Omega)$ is the DFT of $s(n)$, etc. The white noise process $e(n)$ has the spectral power density η. These models are specified further below.



(a)



(b)

Figure 5.1: Sample results from the simulation that shows (a) the signal $s(n)$ as a response to a single impulse in $x(n)$ and (b) the measured output $y(n)$ without noise as a response to a single impulse in $x(n)$.

5.3.2 Signal model

For the examples in this section we assume signal models with the following impulse responses

$$f(n) \triangleq \begin{cases} f_s \mathcal{T}_{N_f}(n), & 0 \le n \le N_f - 1 \\ 0, & \text{otherwise} \end{cases} \quad \text{[1/ms]} \tag{5.16}$$

$$h(n) \triangleq \begin{cases} f_s \sin\left(\frac{2\pi n}{N_h - 1}\right) \mathcal{T}_{N_h}(n), & 0 \le n \le N_h - 1 \\ 0, & \text{otherwise} \end{cases} \quad \text{[mV/ms]} \tag{5.17}$$

where $N_f = 5$, $N_h = 40$, $F(\Omega) = \mathcal{F}\{f(n)\}$ and $H(\Omega) = \mathcal{F}\{h(n)\}$, see also Figure 5.1.

5.3.3 Noise model

The following second order ARMA noise model is used in the simulations below

$$C(\zeta) = 0.04(\zeta^2 - \zeta + 0.25) \tag{5.18}$$
$$D(\zeta) = \zeta^2 - 1.8\zeta + 0.81 \quad . \tag{5.19}$$

The spectral power density of the noise, $P_v(\Omega)$ is then

$$P_v(\zeta) = \left|\frac{C(\zeta)}{D(\zeta)}\right|^2 \eta \quad \text{[mV}^2\text{/kHz]} \quad . \tag{5.20}$$

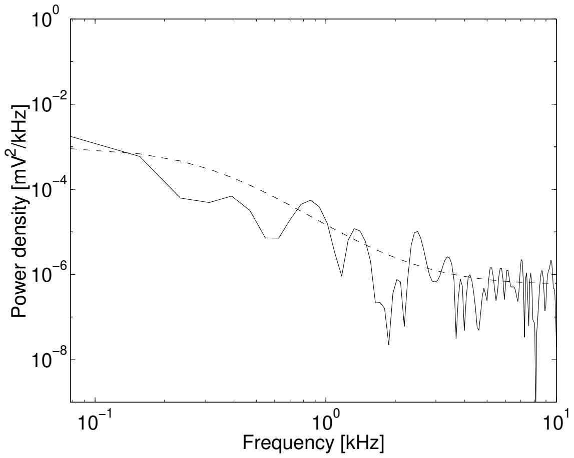


Figure 5.2: The spectral power density of the noise model (dashed) with an example realization (solid) included.

5.3.4 Deconvolution in the frequency domain

To find the optimal Wiener filter $G(\Omega)$ on the unit circle using (5.2), the cross spectral density $\phi_{sy}(\Omega)$ and the spectral density $\phi_y(\Omega)$ are required.

However, there is one major problem in this example, which will also appear in the realistic setting of Chapter 8: The deterministic signals are of finite *energy*, referred to as *energy signals*, while the stochastic noise is of finite *power*, referred to as a *power signal*. Albeit the difference is subtle, its implications are of major importance.

To exemplify this, assume first that the design is based on the spectral *energy* density. This would result in a filter design that erroneously regards an increased batch length (all other parameters being unchanged) as an increase of the noise level. The total noise energy within the data batch will then increase while the signal energy remains constant.

By basing the design on the spectral *power* density instead, the problem is the opposite. This would result in a filter design that regards an increased batch length as a decrease in signal strength of the energy signals, while the noise power remains constant.[2]

We approach this problem by finding the largest batch T_v within which the signal $s(n)$ actually (significantly) affects the output $y(n)$ at some time instant. With a finite-length signal generated by the signal model, T_v is also finite and

[2]The result of this on the Wiener filter design becomes exactly the same as when spectral energy density is considered.

at most twice the duration of the signal.[3] In cases where it is difficult to find a suitable criterion for the definition of T_v it may simply be regarded as a design variable. Note that the parameter T_v is set to a *fixed* value that depends on a subtle relationship between the signal and the noise. It does in no way depend on the batch length T.

As shown in Figure 5.1 (b), the length of the output signal is in this example about 2 ms. We therefore use a T_v twice of this value, or

$$T_v \triangleq 4 \quad [\text{ms}] \ . \tag{5.21}$$

With the model (5.10), the spectral energy densities become

$$\phi_{sy}(\Omega) = X(\Omega)F(\Omega)X^*(\Omega)F^*(\Omega)H^*(\Omega) \tag{5.22}$$

$$\phi_y(\Omega) = |X(\Omega)F(\Omega)H(\Omega)|^2 + P_e(\Omega)T_v \tag{5.23}$$

which yields the Wiener filter using (5.2)

$$G(\Omega) = \frac{X(\Omega)F(\Omega)(X(\Omega)F(\Omega)H(\Omega))^*}{|X(\Omega)F(\Omega)H(\Omega)|^2 + P_e(\Omega)T_v} \ . \tag{5.24}$$

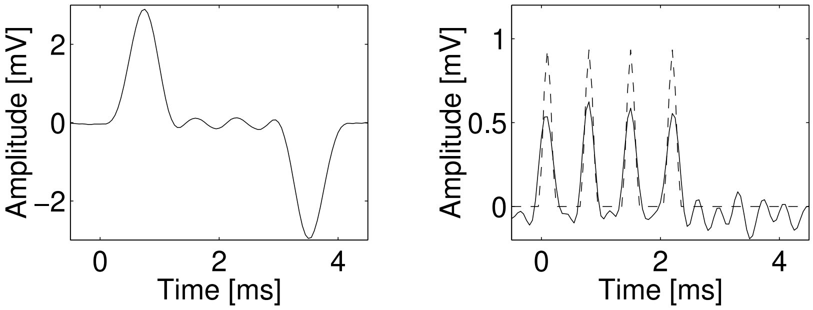
In Figure 5.3 (a), an example simulation is shown ($\eta = 10^{-3}$ mV^2/kHz) where four responses are placed in such a way that the negative and positive peaks of the responses in the middle cancel each other. It is not obvious from the figure that there are four responses active in the measured output $y(n)$.

By deconvolving $y(n)$ using the Wiener filter derived above to estimate the signal $s(n)$, the four peaks clearly appears, see Figure 5.3 (b). Because there is noise present in the measured output, the Wiener filter is forced to decrease the gain at the frequencies where the noise is dominant. This is manifested by the decreased amplitude of the estimated signal $\hat{s}(n)$ compared to the true signal $s(n)$ as shown in the figure. Moreover, the base line is disturbed by the amplified noise.

5.4 Concluding remarks

This chapter briefly presented the Wiener filter. The focus was on Wiener filtering in the frequency domain where the FFT algorithm may be used to create a computationally fast implementation. This implementation constitutes an approximation of the unrealizable (non-causal) filter.

[3] If the largest batch in this respect is longer than twice the duration of the signal, the filter is not optimal because it allows samples containing noise only to affect the filtered output.



(a) (b)

Figure 5.3: An example of deconvolution using a Wiener filter. (a) The measured signal $y(n)$ is shown where the individual responses are placed to maximally cancel each other. The responses in the middle are almost completely cancelled whereas the leftmost and rightmost responses are only partially cancelled. (b) By applying the Wiener filter, the four peaks of the signal $s(n)$ (dashed) appears clearly in the estimated signal $\hat{s}(n)$ (solid) albeit with a lower amplitude and disturbed base line.

Because non-causal Wiener filters are handled easily in the frequency domain, the realizable Wiener filter was mentioned only briefly. Similarily, Wiener filtering in the time domain was not considered more than conceptually.

There also exist several good methods of Wiener filtering in the time domain. See, e.g., [2] [90] [3]. These filters may be extended to incorporate an averaged MSE minimization criterion, c. f. (5.1), applied to a *class* of models. This approach may be used for robust Wiener filter design when there are model uncertainties.

Due to the mixing of finite energy signals with finite power noise as considered in this chapter, a scalar design parameter, here denoted T_v, must inevitably be introduced.

Part II

Applications

CHAPTER 6

Action potential model of a muscle fiber

RELIABILITY and efficiency when diagnosing muscle and nerve disorders using *electromyography* (EMG) is essential. To improve both aspects, the relationship between the physiological conditions and the electric signal of the muscle upon activation must be accurately known. A mathematical model has several advantages in this respect. Besides being a compact description of the present knowledge, it may also be used for simulations to obtain new insights. Also, and perhaps more intriguing, a mathematical description makes it conceivable to use signal processing methods for "digital dissection", where the physiological parameters are presented to the clinician.

The knowledge about the electrical events in the muscle fiber membrane and the volume conduction characteristics of muscle tissue has made it possible to simulate the APs as if they were recorded with a particular electrode. Without simplification, however, volume conduction theory [22] [20] [70] leads to complex and time-consuming calculations. Therefore, many of the first simulations assumed an infinite volume conductor and approximated each of the two spreading depolarizations by a moving dipole or tripole [40] [30] [17].

Because the dipole and tripole models have some shortcomings [62], other approximations were sought. The *line source model* is obtained by considering the AP as a convolution of a weighting function and a transmembrane current lumped to the center of the muscle fiber [5]. This is a very powerful model that can simulate, e.g., finite volume conductors and finite length fibers [37]. Unfortunately, its complexity makes it unsuitable for fast simulations.

If the volume conductor is assumed infinite, the line source model is simplified considerably while retaining a reasonable accuracy compared with volume conduction theory for most types of muscle fibers.[1]

With the development of the first mathematical descriptions of the membrane potential [47] [20], realistic simulations were simplified. An efficient implementation of the line source model was presented in [62] where the convolution of the transmembrane current and the electrode weighting function are calculated in the frequency domain using the FFT algorithm. *Aliasing* may arise, however, if the Nyquist criterion is violated when the line source model is discretized. This crucial aspect has not been properly addressed in the literature. In this paper, we identify the sources of aliasing and present a design variable that simplifies the selection of a proper discretization frequency.

Often, fast simulations are needed. We meet this requirement by introducing an *anti-aliasing filter* that lowers the required dicretization frequency considerably. This makes it possible to improve the performance of the simulations by either decreasing the calculation time, increasing the accuracy, or attaining both objectives.

Recently, the importance of the finite length of the muscle fibers has been studied [42] [38] [26]. Assuming a finite muscle fiber length in a line source model is straight-forward but has to be done properly to avoid the introduction of artifacts in the model output. In [26], this issue is properly addressed but the discontinuities are handled separately from the convolution. We take this one step further and show that it is possible to account for the finite fiber length by a simple transformation of the weighting function. The calculations may then be carried out in one step which makes the calculations simpler.

The focus of this paper is on modeling of single muscle fiber APs, with simulation of entire MUs and different measuring electrodes in mind. The efficiency of the calculations is therefore a very important aspect in which the anti-aliasing filter and the weighting function transformation play a central role.

In addition to simulation, accurate models may be used in filter design, in particular the deconvolution of the EMG signal described in Chapter 8. Describing the model in a transfer function form is beneficial in this perspective. The ultimate goal would then be the situation where the EMG signal, obtained with the concentric needle electrode, is decomposed into single muscle fiber AP components, using model-based deconvolution.

The chapter is organized as follows. In Section 6.1 the continuous-time line source model is presented and the effects arising from having a finite muscle fiber length are derived. In Section 6.2 the model is discretized into a discrete-time model. The cause of the aliasing errors is presented and an anti-aliasing filter is introduced. In Section 6.3 the three most common electrode types are presented.

[1] The Purkinje fibers of the heart have physical dimensions that break several assumptions of the model and may lead to significant errors [45].

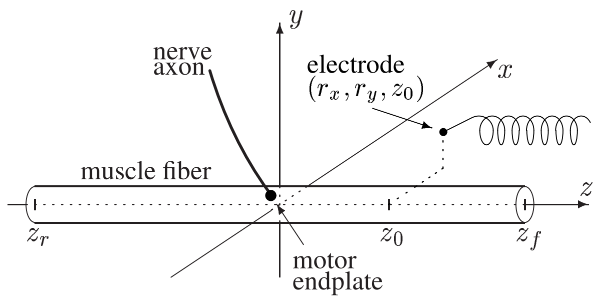


Figure 6.1: The nerve impulse is conducted along the peripheral nerve axon and triggers the muscle fiber at the motor endplate. This results in a depolarization of the muscle fiber membrane that, in turn, causes a transmembrane current to appear and spread towards the two fiber endings where it vanishes. This distribution in both time and space of the transmembrane current is then measured by the electrode.

Simulation results are presented in Section 6.4 followed by a concluding discussion in Section 6.5.

6.1 A continuous-time model

As mentioned above, the muscle fiber receives a nerve impulse from the axon and becomes depolarized, see Figure 6.1. The depolarization generates a transmembrane current that originates at the motor end plate and propagates along the fiber towards the two tendons. Assuming the fiber properties are constant along the fiber, the resulting potential measured by a particular electrode is obtained by convolving the transmembrane current with an electrode dependent weighting function. The measured AP, denoted $\phi_c(t)$, from one fiber may thus be modeled by

$$\phi_c(t) = W_c(p) I_c(p) x_c(t) \tag{6.1}$$

or, in the frequency domain,

$$\Phi_c(s) = W_c(s) I_c(s) X_c(s) \tag{6.2}$$

where p is the derivative operator ($px_c(t) = \frac{d}{dt}x_c(t)$), $W_c(\cdot)$ is the electrode dependent weighting function, and $I_c(\cdot)$ is the transmembrane current. The impulse train delivered by the axon is described by $x_c(t)$ and $X_c(s)$ in the time domain and the frequency domain, respectively. This model, explained in more detail below, is illustrated in Figure 6.2.

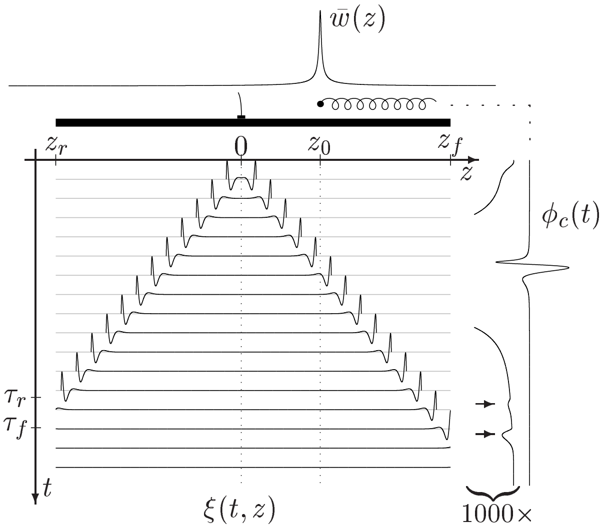
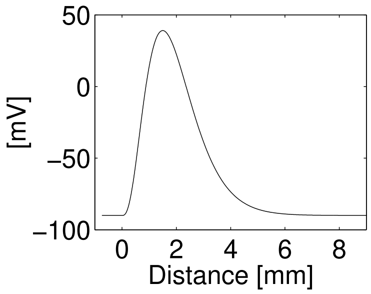


Figure 6.2: The generation of an AP, denoted $\phi_c(t)$, using the line source model is illustrated in a two-dimensional spatio-temporal view. The spatial dimension is along the axial direction of the muscle fiber and is drawn from left to right in the figure. The temporal dimension is drawn from top to bottom where time increases downwards. At $t = 0$, a depolarization is initiated at the motor end plate (z = 0) causing two transmembrane currents to move towards the fiber endings where they vanish at $t = \tau_f$ $(z = z_f)$ and $t = \tau_r$ $(z = z_r)$, respectively. The electrode is positioned at $z = z_0$ and records the resulting AP, i.e., the time-space transmembrane current distribution, $\xi(t, z)$, weighted by the electrode characteristic spatial weighting function, $\bar{w}(z)$. Next to the resulting AP is an enlargement of it with the central part excluded. The end effects, or terminal waves, originating from the two muscle-tendon junctions are indicated by arrows. Note the absence of a corresponding motor end plate effect at $t = 0$.

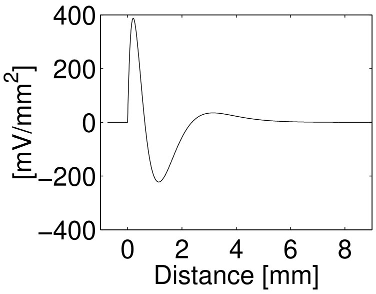
To simplify the notation and the calculations below, we assume that $x_c(t)$ contains only one pulse at $t = 0$, i.e., $x_c(t) = \delta(t)$, where $\delta(t)$ is the Dirac function. Extending this to the general case is straight-forward.

6.1.1 The temporal transmembrane current

We derive the temporal transmembrane current $I_c(s)$ in (6.2) using the core conduction model [5]. According to this model the spatial instantaneous transmem-



(a)



(b)

Figure 6.3: The instantaneous spatial distribution along the fiber of (a) the intracellular potential and (b) its second derivative.

brane current per unit length of fiber, $\bar{i}_m(z)$, is calculated as[2]

$$\bar{i}_m(z) = \frac{\sigma_i \pi d^2}{4} \frac{d^2 V(z)}{dz^2} \quad [\mu\text{A/mm}] \tag{6.3}$$

where σ_i is the intracellular conductivity [S/m], d is the fiber diameter [mm], and $V(z)$ is the intracellular potential [mV] as a function of the distance z [mm] from the depolarization point.

When the transmembrane current $\bar{i}_m(z)$ in (6.3) moves to the left with the fiber's conduction velocity c, see Figure 6.2, the temporal transmembrane current $i_c(t)$ may be observed at the point $z = 0$ according to

$$i_c(t) = \bar{i}_m(ct) = \frac{\sigma_i \pi d^2}{4} \left. \frac{d^2 V(z)}{dz^2} \right|_{z=ct} \quad [\mu\text{A/mm}] \ . \tag{6.4}$$

Following [62] to evaluate the expression (6.4), we adopt a simple model of the intracellular potential and its second derivative, see Figure 6.3,

$$V(z) \overset{\triangle}{=} 768 z^3 e^{-2z} - 90 \quad [\text{mV}], \quad z \geq 0 \tag{6.5}$$

$$\frac{d^2 V(z)}{dz^2} = 768(6z - 12z^2 + 4z^3) e^{-2z} \quad [\text{mV/mm}^2] \ . \tag{6.6}$$

Inserting (6.6) into (6.4) and applying the Laplace transform, the sought transmembrane current in the frequency domain, $I_c(s)$, is

$$I_c(s) = \mathcal{L}\{i_c(t)\} = 768 \frac{\sigma_i \pi d^2}{4} \frac{6cs^2}{(s + 2c)^4} \ . \tag{6.7}$$

[2]The deviation from the volume conductor model is around one percent [5].

As is generally accepted, we assume a linear relationship between conduction velocity and fiber diameter, using a relation proposed in [62]

$$c \triangleq 3.7 + 50(d - 55 \cdot 10^{-3}) \quad \text{[m/s]} \ . \tag{6.8}$$

6.1.2 The temporal weighting function

The temporal weighting function $w_c(t)$ corresponding to $W_c(s)$ in (6.2) is derived from the spatial weighting function $\bar{w}(z)$, here assumed known. The function $\bar{w}(z)$ is electrode dependent and describes the potential produced by a unit current source located on a straight line at a certain radial distance, see Section 6.3.

The potential recorded by the electrode is then the spatially weighted summation of the time dependent transmembrane current distribution along the fiber weighted by the spatial weighting function $\bar{w}(z)$, see Figure 6.2. Denoting the time-space distribution of the transmembrane current by $\xi(t, z)$, the recorded AP $\phi_c(t)$ is

$$\phi_c(t) = \int_{-\infty}^{\infty} \bar{w}(z)\xi(t, z)dz \ . \tag{6.9}$$

The current distribution $\xi(t, z)$ is derived by first extending $i_c(t)$ in (6.4) to include all points along the fiber and not only $z = 0$. After this, the effects of the finite length of the muscle fiber are considered.

Figure 6.2 shows $\xi(t, z)$ graphically where two transmembrane currents are generated at the motor end plate and move in opposite directions with a constant velocity c. The transmembrane current that moves in the forward direction (increasing z) is denoted $i_f(t, z)$ and the transmembrane current that moves in the reverse direction (decreasing z) is denoted $i_r(t, z)$. Both currents vanish at the fiber endings where the depolarization regeneration ceases. Obviously, these two functions may be formulated as

$$i_f(t, z) = i_c\left(t - \frac{z}{c}\right), \quad 0 \leq z \leq z_f \tag{6.10}$$

$$i_r(t, z) = i_c\left(t + \frac{z}{c}\right), \quad z_r \leq z \leq 0 \tag{6.11}$$

where both $i_f(t, 0)$ and $i_r(t, 0)$ are equal to $i_c(t)$ in accordance with (6.4).

To obtain the final expression of $\xi(t, z)$, the constraints in (6.10) and (6.11) are removed by using the Heaviside function $\theta(\cdot)$. Moreover, any possible end effects taking place in the discontinuities at the motor end plate and at the two fiber endings, respectively, are considered. Therefore, the time-space distribution of the transmembrane current $\xi(t,z)$ is defined as

$$\begin{aligned}
\xi(t,z) \;\stackrel{\triangle}{=}\; & i_f(t,z)(1-\theta(z-z_f)) \\
& + i_r(t,z)(1-\theta(-z+z_r)) \\
& + \psi_0(t)\delta(z) \\
& + \psi_1(t)\delta(z-z_f) \\
& + \psi_2(t)\delta(z-z_r) \qquad (6.12)
\end{aligned}$$

where the $\psi_i(t)$:s, $i = 0, 1, 2$, account for the discontinuities.

To obtain the temporal weighting function $w_c(t)$, and hence $W_c(s)$ in (6.2), (6.12) may be used to expand (6.9) through the substitution $z = ct$, see Appendix 6.A. Then, the measured potential $\phi_c(t)$ turns out to be a convolution between the transmembrane current $i_c(t)$ and the desired temporal weighting function $w_c(t)$ according to

$$\phi_c(t) = i_c(t) * w_c(t) \stackrel{\triangle}{=} \int_0^\infty w_c(\tau) i_c(t-\tau) d\tau \quad . \qquad (6.13)$$

Through the derivation of $w_c(t)$, one actually obtains one weighting function in each direction, $w_f(t)$ and $w_r(t)$, because $i_f(t,z)$ is affected by $\bar{w}(z)$ for $z \geq 0$ only and $i_r(t,z)$ is affected by $\bar{w}(z)$ for $z \leq 0$ only, see Appendix 6.A and Figure 6.4. For interpretational reasons, see below, this division is kept throughout this paper. The two functions are related to $w_c(t)$ as

$$\begin{aligned}
w_c(t) &\stackrel{\triangle}{=} w_f(t) + w_r(t) && (6.14) \\
w_f(t) &= c(\bar{w}(ct) - \bar{w}(0))(1-\theta(t-\tau_f)) \\
&\quad + c(\bar{w}(z_f) - \bar{w}(0))\theta(t-\tau_f) && (6.15) \\
w_r(t) &= c(\bar{w}(-ct) - \bar{w}(0))(1-\theta(t-\tau_r)) \\
&\quad + c(\bar{w}(z_r) - \bar{w}(0))\theta(t-\tau_r) && (6.16) \\
\tau_f &= \frac{z_f}{c} \;\; ; \quad \tau_r = \frac{-z_r}{c} \quad . && (6.17)
\end{aligned}$$

Owing to the division of the weighting function, an intuitive interpretation of the transformation of a spatial weighting function $\bar{w}(z)$ to a temporal weighting function $w_c(t)$ is possible as follows, see Figure 6.2 and Figure 6.4. The spatial weighting function $\bar{w}(z)$ is translated vertically so that it is zero at the motor end plate ($\bar{w}(0) + \textit{const} = 0$). Then, the value at each fiber ending is extrapolated to infinity ($\bar{w}(z) = \bar{w}(z_f)\ \forall z \geq z_f$, $\bar{w}(z) = \bar{w}(z_r)\ \forall z \leq z_r$). The part to the left of the motor end plate ($\{\bar{w}(z) : z < 0\}$) is "folded" to the right and added to the part

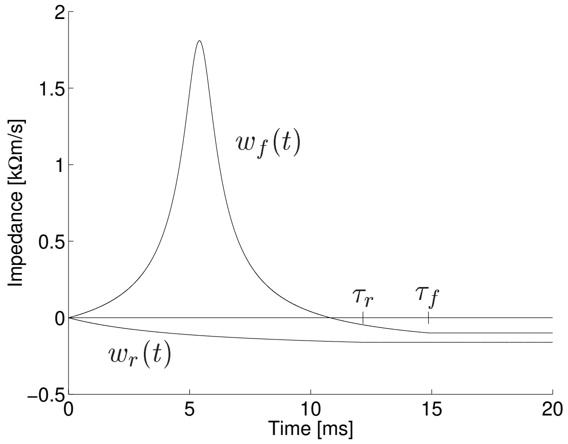


Figure 6.4: Example of a temporal weighting function in the forward $w_f(t)$ and the reverse $w_r(t)$ directions for a point electrode at a radial distance of 1 mm. The conduction velocity is set to 3.7 m/s and the length of the muscle fiber is 100 mm. Due to the non-centered end plate position, the arrival times (τ_f and τ_r) at the fiber tendon differ. Note that each weighting function is constant for $t > \tau_f$, and $t > \tau_r$, respectively.

to the right of the motor end plate ($\{\bar{w}(z) : z > 0\}$). The transformation is then completed by the substitution $z = ct$.

The temporal weighting function $W_c(s)$ in the frequency domain is then obtained by taking the Laplace transform of the weighting function $w_c(t)$ in (6.14)

$$\begin{aligned} W_c(s) \quad = \quad & c\Bigg(\mathcal{L}\left\{(\bar{w}(ct) - \bar{w}(0))(1 - \theta(t - \tau_f))\right\} \\ & + (\bar{w}(z_f) - \bar{w}(0))\frac{1}{s}e^{-\tau_f s} \\ & + \mathcal{L}\left\{(\bar{w}(-ct) - \bar{w}(0))(1 - \theta(t - \tau_r))\right\} \\ & + (\bar{w}(z_r) - \bar{w}(0))\frac{1}{s}e^{-\tau_r s}\Bigg) \quad . \end{aligned} \tag{6.18}$$

This concludes the derivation of the continuous-time representation of a measured AP. This representation is most useful for understanding the underlying relations of the AP generation. For simulations and signal processing, it is convenient to use the discrete-time representation derived next.

6.2 A discrete-time model

In this section a discrete-time representation of the line source model is derived through impulse-invariant discretization (sampling) of the continuous-time model (6.1). This approach is straight-forward and well known [63] [26]. Using a discretization interval T, the discrete-time model may be written as

$$\phi(n) = W(q)I(q)x(n) \tag{6.19}$$

where $\phi(n)$ is the discretized potential ($\approx \phi_c(nT)$), q is the forward shift operator ($qx(n) = x(n+1)$), $W(q)$ is the electrode dependent weighting function, $I(q)$ models the transmembrane current, and $x(n)$ is the impulse train delivered by the axon.

To avoid aliasing when discretizing the continuous-time model functions, the Nyquist criterion has to be fulfilled for these functions. The aliasing issues, however, have not been addressed properly in the literature. When addressed, it has been the bandwidth of the *resulting AP* that has been considered in the Nyquist criterion. This is not correct because, in the discrete-time line source model, the transmembrane current and the weighting function are discretized *prior* to the convolution that generates the resulting AP. In essence, the discrete-time AP should not be regarded as sampled since it is generated in discrete time directly. As shown in Section 6.4.2, the transmembrane current has a significant high-frequency content that cannot be seen in the resulting AP.

The most important consequence of the aliasing is the appearance of a static error in the resulting AP. When modeling entire MUs with as many as 500 fibers or even more, it is very important that the static errors are vanishingly small. Even small magnitude errors can, in such a case, be amplified to a dominant source of error. This in combination with the demand for fast simulations causes a tradeoff between accuracy and low discretization frequency.

By properly addressing these issues, a discrete-time model is possible to derive that is optimized for accurate and computationally efficient computer based modeling of muscle fiber APs. Our approach, obvious when sampling real signals, is to use anti-aliasing prefiltering within the model, which allows significantly lower discretization frequencies. To understand this, assume, for example, that we have an AP with a 5 kHz bandwidth. In general, this corresponds to a transmembrane current with a bandwidth of 20 kHz and a weighting function with a bandwidth of 5 kHz. Hence, despite the bandwidth of the AP being 5 kHz in this example, aliasing-free discretization requires a discretization frequency of 40 kHz. Using an anti-aliasing filter to remove the excess bandwidth of the transmembrane current, the discretization frequency may be lowered by as much as a factor four to 10 kHz.

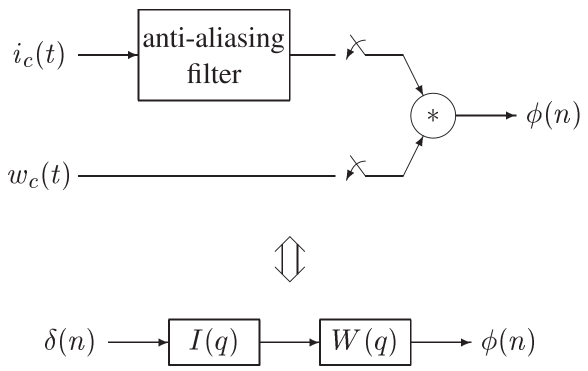


Figure 6.5: A block diagram of the discretized model.

This indicates that the anti-aliasing filter improves the efficiency while retaining the accuracy.

The anti-aliasing filter has to be implemented in the continuous-time model, however, and the filtered model is then discretized to obtain the final discrete-time model, see Fig 6.5. We shall use a Bessel filter of the second order as an anti-aliasing filter, see Appendix 6.B.

6.2.1 The transmembrane current model

After prefiltering the continuous-time transmembrane current (6.4) using an anti-aliasing filter, the actual discretization of the resulting continuous-time model is straight-forward. We derive the discrete-time transmembrane current per unit length of fiber, $i(n)$, through impulse-invariant discretization of the prefiltered analog using a discretization interval T as described by

$$i(n) = \frac{\sigma_i \pi d^2}{4} \left. \frac{d^2 V_H(z)}{dz^2} \right|_{z=cnT} \quad [\mu\text{A/mm}] \tag{6.20}$$

where $\frac{d^2 V_H(z)}{dz^2}$ is given in Appendix 6.B.

By taking the Z-transform of (6.20), the impulse-invariant discretization of the transmembrane current per unit length of fiber in the frequency domain is obtained, see Appendix 6.C.

6.2.2 The weighting function model

As mentioned above, the weighting function has the lowest bandwidth (similar to the bandwidth of $\phi(n)$) of the two generating model functions. This means that if the discretization frequency is sufficiently high to accomodate the bandwidth of the continuous-time AP, the aliasing errors in the discretization of the weighting function is small. Since this is the normal case, no anti-aliasing filter for the weighting function is presented. The discrete-time weighting function is then directly derived through impulse-invariant discretization of the continuous-time model $w_c(t)$ in (6.14) using a discretization interval T as described by

$$w(n) = w_c(t)|_{t=nT} \quad [\mathrm{k}\Omega\mathrm{m/s}] \ . \tag{6.21}$$

To find the frequency domain weighting function $W(\zeta)$ corresponding to the continuous-time $W_c(s)$ in (6.18), we approximate the continuous time shift $e^{-\tau s}$ with q^{-k} where k is the number of samples approximating the time shift τ. The weighting function $W_c(s)$ is thus in discrete time approximated with

$$\begin{aligned} W(\zeta) \quad = \quad & c\Bigg(\mathcal{Z}\left\{(\bar{w}(cnT) - \bar{w}(0))(1-\theta(cnT-\tau_f))\right\} \\ & + (\bar{w}(z_f) - \bar{w}(0))\frac{1}{\zeta - 1}\zeta^{-k_1} \\ & + \mathcal{Z}\left\{(\bar{w}(cnT) - \bar{w}(0))(1-\theta(cnT-\tau_r))\right\} \\ & + (\bar{w}(z_r) - \bar{w}(0))\frac{1}{\zeta - 1}\zeta^{-k_2}\Bigg) \ . \end{aligned} \tag{6.22}$$

6.3 Electrode characteristic weighting functions

In the simulations, see Section 6.4, we present the performance of the line source model. Here we recapitulate the weighting functions of the three most common electrodes. These were assumed known in Section 6.1 and 6.2.

The physical dimensions of standard *single fiber* (SF), *concentric needle* (CN), and *Macro* electrodes are listed in Table 6.1.

6.3.1 Impedance of a point electrode

In an infinite medium with cylindrical anisotropy, a current source ι_g located at the origin produces a potential $\phi(x,y,z)$ in a point electrode positioned at (x,y,z), see Figure 6.1, by [5] [63]

$$\phi(x,y,z) = Z(x,y,z)\iota_g \tag{6.23}$$

$$Z(x,y,z) = \frac{1}{4\pi\sigma_r}\frac{1}{\sqrt{K(x^2+y^2)+z^2}} \quad [\mathrm{k}\Omega\mathrm{m/s}] \tag{6.24}$$

where $Z(x,y,z)$ is the impedance of the point electrode, σ_r is the radial conductivity, σ_z is the axial conductivity, and $K \triangleq \frac{\sigma_z}{\sigma_r}$ is the anisotropy ratio.

Table 6.1: Electrode specifications

Description	*Parameter*		*Value*
Needle radius	R_n	[mm]	0.225
Needle inner radius	R_e	[mm]	0.150
CN core radius	R_c	[mm]	0.075
CN recording angle	θ_{CN}	[rad]	$\frac{\pi}{12}$
CN cannula length	L_{CN}	[mm]	20
Macro cannula length	L_M	[mm]	15
SF core radius	R_{SF}	[mm]	0.0125
SF recording position	x_{SF}	[mm]	$\frac{1}{2}L_M$

6.3.2 The electrode weighting functions

Assuming that a muscle fiber may be approximated by a straight and cylindrical entity, and that the potential produced in a particular electrode is the average of the potential present on the recording surface, the following three models were derived: the Macro electrode, the SF electrode, and the CN electrode. The latter two are bipolar and use the cannula as reference. More accurate models exist [66] [92], but their complexity and computational demand often outweigh their improvement in accuracy.

The three weighting functions below are derived using the electrode specifications listed in Table 6.1 and assuming the electrode is held perpendicular to the fiber.

The macro electrode and the cannula reference

Approximating the cylindrical recording surface with a thin wire of length L, the weighting functions of the Macro electrode ($L = L_M$) and the concentric needle cannula ($L = L_{CN}$) are the average over its length [40] [61], see Fig 6.6. For the

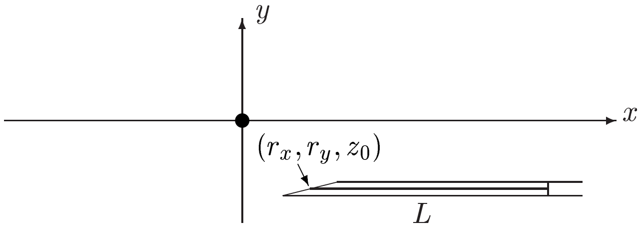


Figure 6.6: The Macro electrode and the cannula of the CN electrode are approximated by a wire (thick line) of length $L = L_M$ and $L = L_{CN}$, respectively. The left end of the wire is at the position (r_x, r_y). The muscle fiber is perpendicular to the needle and located at the origin.

Macro electrode, we thus obtain

$$\bar{w}_M(z) \stackrel{\triangle}{=} \frac{1}{L_M}\int_{L_M} Z\,dL \tag{6.25}$$

$$= \frac{1}{4\pi\sigma_r\sqrt{K}L_M} \ln\frac{\sqrt{(r_x+L_M)^2+B}+r_x+L_M}{\sqrt{r_x^2+B}+r_x} \quad [\mathrm{k}\Omega\mathrm{m/s}] \tag{6.26}$$

with $B = r_y^2 + \frac{1}{K}(z - z_0)^2$.

The weighting function, $\bar{w}_{CNc}(z)$, of the CN electrode cannula is calculated analogously using

$$\bar{w}_{CNc}(z) \stackrel{\triangle}{=} \frac{1}{L_{CN}}\int_{L_{CN}} Z\,dL \quad . \tag{6.27}$$

The single fiber electrode

The active surface of the SF electrode is located in a side port 7.5 mm from the tip of the electrode. Neglecting the so-called *wall effect* [62], the weighting function of the active surface is approximated by a single point due to its dimunitive dimension as described by

$$\bar{w}_{SFa}(z) \stackrel{\triangle}{=} Z(-(r_x + x_{SF}), -(r_y + R_n), z - z_0) \tag{6.28}$$

$$= \frac{1}{4\pi\sigma_r\sqrt{Kr^2 + (z-z_0)^2}} \quad [\mathrm{k}\Omega\mathrm{m/s}] \tag{6.29}$$

where $r = \sqrt{(r_x + x_{SF})^2 + (r_y + R_{SF})^2}$ is the radial distance translated by the off-tip position of the SF core, see Table 6.1.

The final weighting function of the bipolar SF electrode is

$$\bar{w}_{SF}(z) \stackrel{\triangle}{=} \bar{w}_{SFa}(z) - \bar{w}_{M}(z) \quad . \tag{6.30}$$

The concentric needle electrode

Approximating the surface of the central core of the CN electrode by $2N_c + 1$ parallell wires, see Figure 6.7, the weighting function, $\bar{w}_{CNa}(z)$, of the active surface of the CN electrode is [63]

$$\begin{aligned}
\bar{w}_{CNa}(z) &\stackrel{\triangle}{=} \frac{1}{S}\iint_S Z\,dS && (6.31)\\
&\approx \frac{1}{2N_c+1}\sum_{k=-N_c}^{N_c}\frac{1}{8\pi\sigma_r\sqrt{K}\Delta u_k} \\
&\ln\frac{\sqrt{(r_u+\Delta u_k)^2+B_k}+r_u+\Delta u_k}{\sqrt{(r_u-\Delta u_k)^2+B_k}+r_u-\Delta u_k} \quad [\text{k}\Omega\text{m/s}] && (6.32)\\
B_k &= r_v^2 + \frac{1}{K}(z-(\Delta z_k+z_0))^2 && (6.33)\\
\Delta z_k &= R_c\frac{k}{N_c+1}, \quad k=-N_c,\ldots,N_c && (6.34)\\
\Delta u_k &= \frac{R_c}{\sin(\theta_{CN})}\cos(\arcsin(\frac{k}{N_c+1})) && (6.35)\\
r_u &= r_x\cos\theta_{CN} + r_y\sin\theta_{CN} && (6.36)\\
r_v &= -r_x\sin\theta_{CN} + r_y\cos\theta_{CN} && (6.37)
\end{aligned}$$

where θ_{CN} is the angle of the beveled tip of the CN electrode.

The weighting function of the bipolar CN electrode is [60]

$$\bar{w}_{CN}(z) \stackrel{\triangle}{=} \bar{w}_{CNa}(z) - \bar{w}_{CNc}(z) \quad . \tag{6.38}$$

6.4 Simulation results

In this section, the characteristics of the model is visualized through simulations. We present some sample APs, the frequency distribution and bandwidth of the model components, different aliasing effects, and the positional dependence of the AP shape. In all simulations, unless stated differently, the parameter settings used are found in Table 6.2.

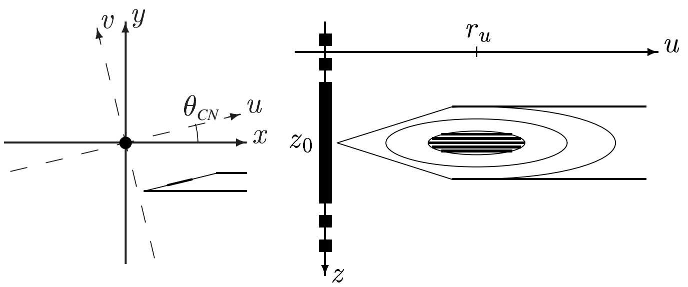


Figure 6.7: The core of the CN electrode is approximated by five parallel wires (thick lines). To simplify the calculations, an alternate coordinate system (u, v) is introduced. It is rotated by the angle θ_{CN} and the u-axis is parallel with the recording surface on the beveled tip of the CN electrode.

Table 6.2: Line source model parameters for simulations

Description	*Parameter*		*Value*
Fiber diameter	d	[mm]	$55 \cdot 10^{-3}$
Conduction velocity	c	[m/s]	3.7
Radial distance	r	[mm]	0.1
Axial distance	z_0	[mm]	20
Filter cut-off frequency	f_0	[kHz]	5.8
Intracellular conductivity	σ_i	[S/m]	1.01
Axial conductivity	σ_z	[S/m]	0.33
Radial conductivity	σ_r	[S/m]	0.063
CN core approximation size	N_c		2

6.4.1 Action potential simulation

The most time-consuming part of the line source model evaluation is the convolution of the transmembrane current and the electrode characteristic weighting function. For fast evaluation of the convolution, it is calculated in the frequency domain via the FFT algorithm [62]. When doing that, both the weighting function and the transmembrane current must be discretized using the same discretizing frequency and must be padded with zeroes to their double length. The latter operation is to avoid circular convolution. After taking the inverse FFT, the resulting potential is

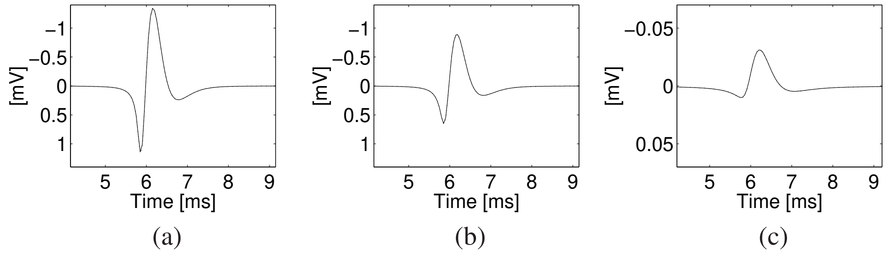


Figure 6.8: Simulations of an AP as recorded with (a) the SF electrode (SFAP), (b) the CN electrode (CNAP), and (c) the Macro electrode (Macro AP). Note the different scale of the Macro AP. The SF recording yields the highest amplitude, whereas the peak-to-peak rise time is about the same, see text.

truncated to the length of the original vectors.

In Figure 6.8 three APs calculated using this technique are shown. Each is a sample of an AP recorded by the SF electrode (SFAP), the CN electrode (CNAP), and the Macro electrode (Macro AP), respectively. As expected, the SFAP has the largest amplitude and the most pronounced phases. Moreover, the peak-to-peak rise time is the same for the SFAP and the CNAP. The rise time is slightly larger for the Macro AP because the radius of the needle forces a longer distance between the fiber and the electrode.

6.4.2 Frequency distribution

As stated in Section 6.2, the large bandwidth of the transmembrane current complicates the choice of discretization frequency because the Nyquist criterion cannot be applied to the AP. Instead, the bandwidth of the transmembrane current and the weighting function introduce constraints on the needed discretization frequency.

As a guide for selecting the correct discretization frequency, the *normalized cumulative power* (NCP), $\varrho(f)$, is defined as

$$\varrho(f) \triangleq \frac{1}{\int_0^\infty P(\nu)d\nu} \int_0^f P(\nu)d\nu \tag{6.39}$$

where $P(\nu)$ is the power spectrum of the function under consideration, e.g., $P(\nu) = |I(j2\pi\nu)|^2$ for the transmembrane current.

In Figure 6.9 the NCP is shown for the transmembrane current and the weighting functions. With the NCP of a SFAP included in the figure, the bandwidth difference, as stated in Section 6.2, of the transmembrane current and the resulting AP is clear.[3] In Figure 6.9 (a) we note that, compared to the resulting AP, the trans-

[3]The SFAP is selected for inclusion because it has the highest bandwidth of the three AP types.

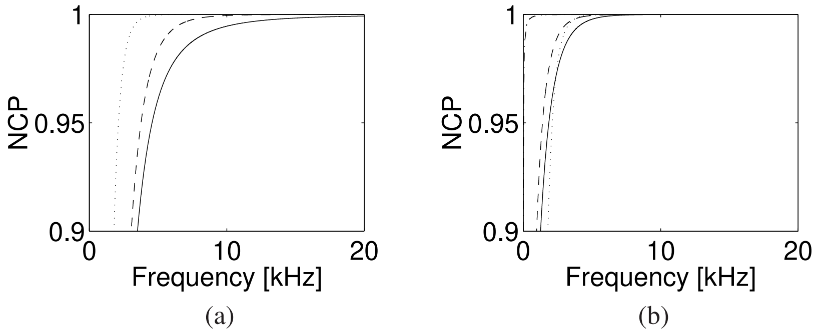


Figure 6.9: (a) The normalized cumulative power (NCP) plot of a transmembrane current both with (dashed) and without (solid) the use of an anti-aliasing Bessel filter. For comparison, the NCP of a SFAP is included (dotted). As can be seen, the unfiltered transmembrane current has substantially more power at higher frequencies compared to the resulting AP.
(b) The NCP plot of the weighting functions of a SF electrode (solid), a CN electrode (dashed), and a Macro electrode (dash-dotted). For comparison, the NCP of a SFAP is included (dotted). Obviously, all four have their power concentrated to low frequencies.

membrane current has a slower increase in the NCP. This means that its spectrum has a slower rolloff, i.e., longer “tails”. The bandwidth decrease and the rolloff improvement of the transmembrane current when applying the anti-aliasing filter is significant. The anti-aliasing filter thus decrease the *Nyquist frequency*, i.e., half of the required discretization frequency, substantially.

Figure 6.9 (b) shows that both bandwidth and rolloff of the weighting function of the SF electrode and the SFAP are similar. The CN electrode bandwidth is slightly lower and the Macro electrode bandwidth is significantly lower. Therefore, it should be sufficient to low-pass filter the transmembrane current only in order to avoid aliasing while using a discretization frequency equal to the sampling frequency of real APs.

The most important usage of the NCP is to calculate the frequency below which a certain amount of the power is found. We call this frequency the *cumulative cut-off frequency* (CCF) and it is defined as

$$\varsigma_\gamma \stackrel{\triangle}{=} \nu : [\varrho(\nu) = \gamma], \quad 0 \leq \gamma \leq 1 \tag{6.40}$$

where $\gamma = 0.999$ has shown to provide a reasonable estimate of the required Nyquist frequency.

Using the CCF, it is evident that the Nyquist frequency must not be set according to the bandwidth of the resulting AP. For example, a SFAP with $\varsigma_{0.999} = 4.2$ kHz may correspond to a transmembrane current with $\varsigma_{0.999} = 18$ kHz. If

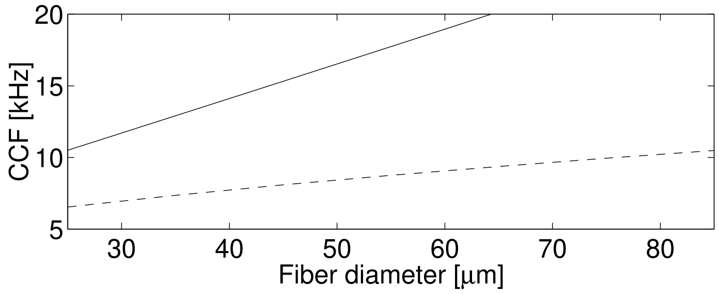


Figure 6.10: The CCF, $\varsigma_{0.999}$, below which 0.999 of the total power is found, of the transmembrane current plotted as a function of fiber diameter, and consequently, as a function of conduction velocity. As shown, the CCF is lowered considerably when the anti-aliasing Bessel filter is used (dashed) as compared to the non-filtered transmembrane current (solid).

the Nyquist frequency of the discretization were set by the bandwidth of the resulting AP, considerable aliasing would occur if no anti-aliasing filter was used. This is particularly important when simulating entire MU potentials acquired with a Macro electrode because the aliasing errors are "amplified" due to the summed contribution from each fiber [61].

Using the Bessel filter, the $\varsigma_{0.999} = 8.8$ kHz suggesting that a discretization frequency of 20 kHz is sufficient to avoid aliasing.

From (6.7) it is apparent that the bandwidth is increased when the conduction velocity increases. Due to the relation between the fiber diameter and conduction velocity, see (6.8), the bandwidth depends on the fiber diameter as well. This dependence is shown in Figure 6.10, where the CCF, $\varsigma_{0.999}$, of the transmembrane current is plotted as a function of fiber diameter in the range 25 μm to 85 μm.[4]

All three weighting functions in Figure 6.9 (b), have their power localized to low frequencies. For the SF electrode $\varsigma_{0.999} = 5.9$ kHz, for the CN electrode $\varsigma_{0.999} = 4.3$ kHz, and for the Macro electrode $\varsigma_{0.999} = 0.74$ kHz. In Figure 6.11, $\varsigma_{0.999}$ of the three weighting functions are shown when the radial distance ranges from 0.05 mm to 2.5 mm. The conduction velocity dependence is visualized with a shaded region where the conduction velocity ranges from 2.2 m/s to 5.2 m/s. The results in the figure suggest that a discretization frequency of about 20 kHz is sufficient in all cases except when using the SF electrode at small radial distances. In these cases, however, the errors are small compared to the AP and may normally be neglected.

[4]Consequently, conduction velocity ranges from 2.2 m/s to 5.2 m/s.

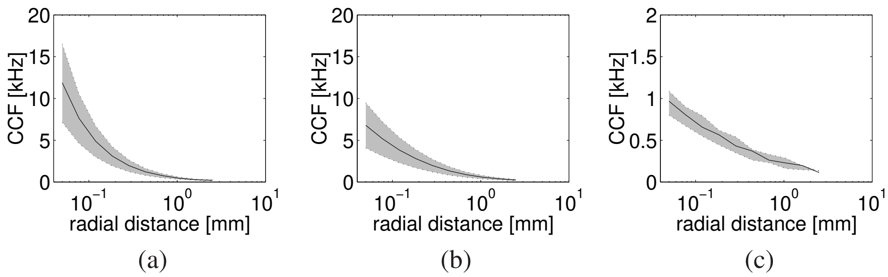


Figure 6.11: The CCF, $\varsigma_{0.999}$, below which 0.999 of the total power of the weighting functions of (a) a SF electrode, (b) a CN electrode, and (c) a Macro electrode is found plotted as a function of radial distance. As NCP frequency depends on the conduction velocity as well, that dependence is shown as a shaded region (2.2-5.2 m/s) where the conduction velocity 3.7 m/s is shown (solid). Note the different scale of the Macro diagram.

6.4.3 Aliasing effects

In Figure 6.12, the aliasing effects are shown as a function of discretization frequency. Note that these particular aliasing effects cannot be found in recorded APs because these effects originate from the discretization of the transmembrane current (mainly) and the weighting function. In recorded APs this never occurs because the convolution takes place in continuous time and only the resulting AP is discretized.

The normalized statical error (NE) in Figure 6.12 is the DC component due to aliasing multiplied by a normalization factor. The normalization factor is the radial distance divided by a nominal distance of 0.1 mm and accounts for the linear increase of the number of muscle fibers with increasing radial distance.

The figure shows that the statical error is about a factor 100 smaller when the anti-aliasing prefiltering is used compared to when not using it. It is clear from the figure what advantage such a filter has to the discretization frequency requirements.

Moreover, the importance of the aliasing issues when simulating Macro APs is clear. The NE is both larger for distant fibers and amplified due to the large number of contributing fibers.

6.4.4 Action potentials recorded by a point electrode

As an example of how the shape of an AP may appear when recorded at different radial and axial distances, Figure 6.13 shows a collection of APs simulated for a single fiber with the recording electrode at different locations. Note how the APs become smoother as the radial distance increases and how the shape and number of phases change when the axial position changes.

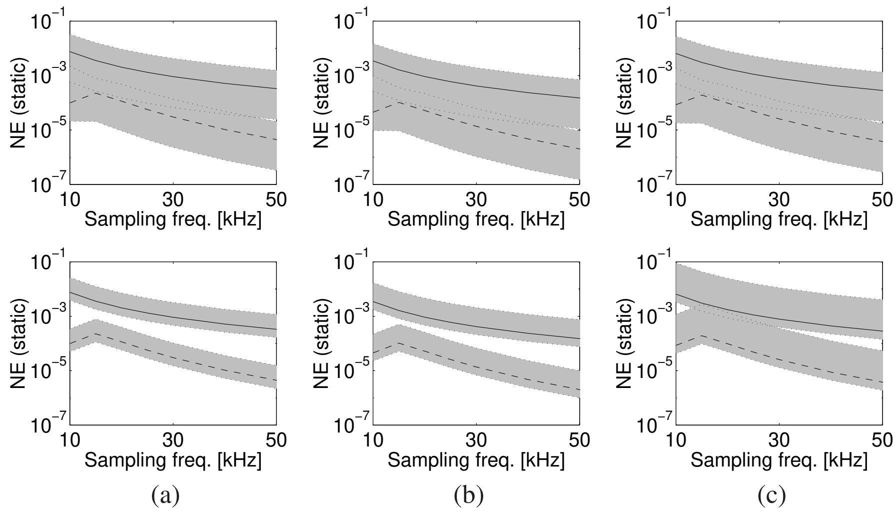


Figure 6.12: The figure shows the normalized statical error as a function of selected sampling frequency for (a) the SF electrode, (b) the CN electrode, and (c) the Macro electrode. The dependence of the fiber diameter (and conduction velocity) is shown (top) as well as the dependence of the radial distance (bottom). The fiber diameter ranges from 25 μm to 85 μm. The radial distance ranges from 0.05 mm to the pick-up distance of each electrode (0.35 mm, 0.5 mm, and 1.45 mm, respectively). In all figures, the result is shown when no anti-aliasing filter is used (solid) and when a second order Bessel filter is used (dashed). The 3.7 m/s and 0.1 mm values are plotted in each range. The NE increases if any of d or r increases.

6.5 Discussion

Insights into the line source model were presented using a signal processing approach. With this perspective, several improvements to the original model were possible to derive. The two most important of these are the anti-aliasing filter and the finite-length adjustment.

Regarding the anti-aliasing filter, there has been little attention in the the literature to the aliasing issues when discretizing the model. The referrals made have incorrectly considered the resulting AP in the Nyquist criterion. In this paper, we showed that this may lead to significant aliasing because the bandwidth of the transmebrane current is in general much higher than the bandwidth of the resulting AP.

We presented an approach where an anti-aliasing second order Bessel filter is applied to the transmembrane current. This way the excess bandwidth is decreased and may be used to speed up the simulations, improving the accuracy, or a mix

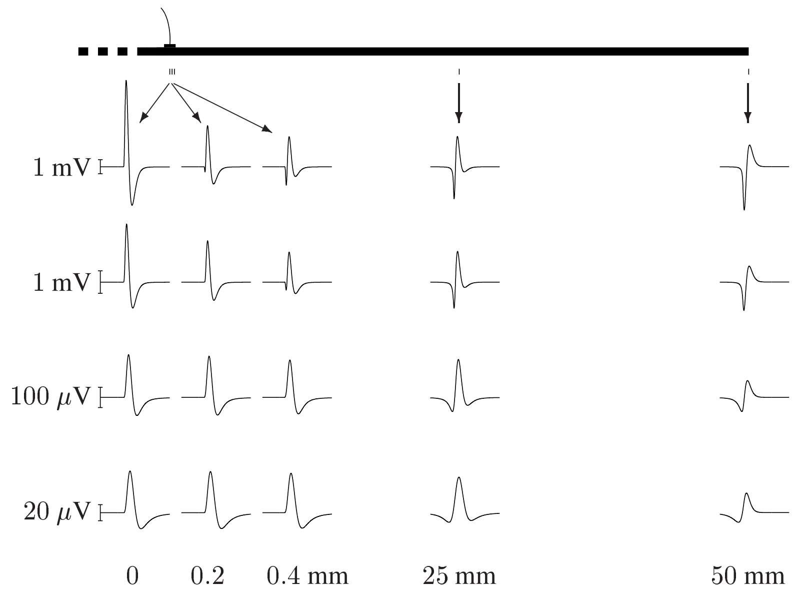


Figure 6.13: The APs as they would be recorded by a point electrode at different axial positions along and radial distances from a 100 mm long muscle fiber with its motor end plate located in the middle. The axial positions are (left to right) 0, 0.2, 0.4, 25 and 50 mm. The radial distances are (top to bottom) 0.05, 0.1, 0.5, and 1 mm. The second last row is enlarged 10 times, and the last row is enlarged 50 times.

of these. Simulations were used to analyze the aliasing effects and to show the improvements the anti-aliasing filter offers.

Regarding the finite length of the muscle fibers, we presented a simple adjustement to the electrode specific weighting function that improves the efficiency of the simulations by allowing the generated AP being described by a single convolution operation.

Also presented were sample APs using the three common electrode types: the SF electrode, the CN electrode, and the Macro electrode. Moreover, we presented a transfer function representation of the transmembrane current that makes it possible to use sophisticated signal processing methods on the EMG signal.

The derived model seems to be reasonably accurate with respect to real APs [53]. In [41] the APs from a single MU were simulated using a similar model with good agreement between simulations and measurements. The simulations carried out with our model correspond well with what medical experise consider an accurate result. This, in combination with the results in [41] makes us confident of the accuracy of the model.

Despite this, there are some aspects of the line source model that need further comments. First, the used intracellular potential is based on a single work presented in 1969 [70] and its accuracy is not fully known. Several variants of this potential have been used, e.g., [62] and [39], suggesting there are some imperfections. Second, the volume characteristics is not fully known and the values used for the radial, axial, and intracellular conductivity are uncertain. Third finally, decreasing the conduction velocity while keeping the diameter constant does not increase the amplitude as expected. Experience has shown, however, that these deficiencies are in most cases negligible.

Instead, the advantages of using this model in this paper are numerous and here we mention two. First, the line source model is computationally fast and is well suited for simulations of large number of fibers and even entire muscles. Such simulations may contribute to a deeper understanding of the correspondance between the EMG signal and the different neuromuscular disorders. This would improve both research and education as well as clinical work. Second, an anti-aliasing filter is incorporated in the model of the membrane current. With this filter, the discretization frequency may be lowered without affecting the accuracy, and hence, further improving the computational speed.

In conclusion, the presented line source model is well suited for fast and accurate AP simulations. With the transfer function formalism of the transmembrane current, an “electrical biopsy” is conceivable. In the near future, the model will be used for simulating MUs as well as entire muscles. A simulation tool [53] is under development and has already shown to be very promising.

Appendix 6.A Derivation of the temporal weighting function

The time-space distribution of the transmembrane current $\xi(t,z)$ is defined in (6.12). To solve for the end effects $\psi_i(t)$, we use that the average transmembrane current along the fiber should be equal to zero [38]. Therefore,

$$\begin{aligned} 0 &= \int_{-\infty}^{\infty} \xi(t,z)dz \qquad (6.41) \\ &= \int_{0}^{z_f} i_f(t,z)dz + \int_{z_r}^{0} i_r(t,z)dz \\ &\quad + \psi_0(t) + \psi_1(t) + \psi_2(t) \quad . \qquad (6.42) \end{aligned}$$

To solve (6.42), two observations are made. First, an end effect is local in its nature. Second, as such, it should be independent of fiber properties at a remote location. The three end effects should thus depend on the end plate position, the first fiber ending, and the second fiber ending, respectively.

By extending (6.42), the integrals may be written as

$$\begin{aligned} 0 &= \int_{0}^{\infty} i_f(t,z)dz - \int_{z_f}^{\infty} i_f(t,z)dz \\ &\quad + \int_{-\infty}^{0} i_r(t,z)dz - \int_{-\infty}^{z_r} i_r(t,z)dz \\ &\quad + \psi_0(t) + \psi_1(t) + \psi_2(t) \quad . \qquad (6.43) \end{aligned}$$

With the observations above, we then conclude that[5]

$$\psi_0(t) = -\int_{0}^{\infty} i_f(t,z)dz - \int_{-\infty}^{0} i_r(t,z)dz \qquad (6.44)$$

$$\psi_1(t) = \int_{z_f}^{\infty} i_f(t,z)dz \qquad (6.45)$$

$$\psi_2(t) = \int_{-\infty}^{z_r} i_r(t,z)dz \quad . \qquad (6.46)$$

The final AP is obtained by summing all contributions along the fiber using the

[5] An equivalent result may be obtained by applying Kirchoff's laws to the core conductor model or by taking the approach described in [26].

weighting function $\bar{w}(z)$ of the measuring electrode

$$\phi_c(t) = \int_{-\infty}^{\infty} \bar{w}(z)\xi(t,z)dz \tag{6.47}$$

$$= \int_0^{z_f} \bar{w}(z)i_f(t,z)dz + \int_{z_r}^0 \bar{w}(z)i_r(t,z)dz + \bar{w}(0)\psi_0(t) + \bar{w}(z_f)\psi_1(t) + \bar{w}(z_r)\psi_2(t) \ . \tag{6.48}$$

Inserting $\psi_i(t)$ from (6.44)-(6.46), $i_f(t,z)$, and $i_r(t,z)$, we can develop the expression further. Then, we set $z = c\tau$ and obtain

$$\begin{aligned}\phi_c(t) = {} & c\int_0^{\tau_f} (\bar{w}(c\tau) - \bar{w}(0))i_c(t-\tau)d\tau \\ & + c\int_0^{\tau_r} (\bar{w}(-c\tau) - \bar{w}(0))i_c(t-\tau)d\tau \\ & + c(\bar{w}(z_f) - \bar{w}(0))\int_{\tau_f}^{\infty} i_c(t-\tau)d\tau \\ & + c(\bar{w}(z_r) - \bar{w}(0))\int_{\tau_r}^{\infty} i_c(t-\tau)d\tau\end{aligned} \tag{6.49}$$

where

$$\tau_f = \frac{z_f}{c} \ ; \quad \tau_r = \frac{-z_r}{c} \ . \tag{6.50}$$

Appendix 6.B A Bessel filter as an anti-aliasing filter

Discretizing the second derivative of the intracellular potential, $d^2V(z)/dz^2|_{z=ct}$, in (6.6), setting $t = nT$, may yield aliasing. By using an anti-aliasing filter, however, this can be reduced. A second order, pole only filter may be written in the frequency domain on the form

$$H(s) = \frac{a_2^2}{s^2 + a_1 s + a_2^2} \tag{6.51}$$

where the coefficients a_i determine the filter characteristics.

Applying the filter $H(s)$ in (6.51) to the second derivative of the intracellular potential $d^2V(z)/dz^2|_{z=ct}$ we obtain in the frequency domain

$$\mathcal{L}\left\{ \left.\frac{d^2V_H(z)}{dz^2}\right|_{z=ct} \right\} = \mathcal{L}\left\{ \left.\frac{d^2V(z)}{dz^2}\right|_{z=ct} \right\} H(s) \tag{6.52}$$

$$= 768\frac{6cs^2}{(s+2c)^4}\frac{a_2^2}{s^2 + a_1 s + a_2^2}. \tag{6.53}$$

Taking the inverse Laplace transform, the filtered second derivative of the intracellular potential in the time domain is

$$
\begin{aligned}
\frac{d^2V_H(z)}{dz^2} &= 2304 a_2 \alpha v \left(\frac{z}{c}\right) && (6.54)\\
v(t) &= C_0 e^{-\alpha t} + C_1 t e^{-\alpha t} \\
&\quad + \frac{1}{2} C_2 t^2 e^{-\alpha t} + \frac{1}{6} C_3 t^3 e^{-\alpha t} \\
&\quad + \left(A\cos(\beta t) + \frac{B - \frac{1}{2}a_1 A}{\beta}\sin(\beta t)\right) e^{-\frac{a_1}{2}t}, \quad t \geq 0 && (6.55)
\end{aligned}
$$

where

$$
\begin{aligned}
\alpha &= 2c \;\; ; \quad \gamma = \alpha^2 - a_1\alpha + a_2 && (6.56)\\
C_0 &= \frac{a_1\alpha^4 - 4a_2\alpha^3 + 4a_2^2\alpha - a_1 a_2^2}{\gamma^4} && (6.57)\\
C_1 &= \frac{a_1\alpha^3 - 3a_2\alpha^2 + a_2^2}{\gamma^3} && (6.58)\\
C_2 &= \frac{a_1\alpha^2 - 2a_2\alpha}{\gamma^2} && (6.59)\\
C_3 &= \frac{\alpha^2}{\gamma} && (6.60)\\
A &= -C_0 \;\; ; \quad B = (\alpha - a_1)C_0 - C_1 && (6.61)\\
\beta &= \sqrt{a_2 - \frac{1}{4}a_1^2} \;\; . && (6.62)
\end{aligned}
$$

A Bessel filter with cut-off frequency f_0 is given by

$$
\begin{aligned}
\omega &= 1.27 \;\; ; \quad \zeta = 0.87 && (6.63)\\
\omega_B &= 2\pi f_0 \quad \text{[krad/s]} && (6.64)\\
a_1 &= 2\zeta\omega\omega_B \quad \text{[krad/s]} && (6.65)\\
a_2 &= \omega^2\omega_B^2 \quad \text{[krad}^2\text{/s}^2\text{]} \;\; . && (6.66)
\end{aligned}
$$

Appendix 6.C The discrete-time transmembrane current in the frequency domain

When using a sufficiently high discretization frequency, the aliasing errors are negligible. By direct impulse-invariant discretization of (6.4) and (6.6), the discrete-time transmembrane current in the frequency domain through the Z-transform is

$$I(\zeta) = 768\frac{\sigma_i \pi d^2}{4}\frac{cT(6-12cT+4c^2T^2)e^{-2cT}\zeta B(\zeta)}{(\zeta - e^{-2cT})^4} \tag{6.67}$$

where the monic second-degree polynomial $B(\zeta)$ have the roots

$$\zeta_{1,2} \quad = \quad \frac{6-4(2\pm\sqrt{3})c^2T^2}{6-12cT+4c^2T^2}e^{-2cT} \quad . \tag{6.68}$$

Often, however, an anti-aliasing filter is needed. In this case, $I(\zeta)$ is acquired by taking the Z-transform of (6.20), which yields

$$\begin{aligned} I(\zeta) \quad = \quad & 2304 a_2 \alpha \frac{\sigma_i \pi d^2}{4}\Bigg(C_0 \frac{\zeta}{\zeta - e^{-\alpha T}} \\ & + C_1 T e^{-\alpha T}\frac{\zeta}{(\zeta - e^{-\alpha T})^2} \\ & + \frac{1}{2}C_2 T^2 e^{-\alpha T}\frac{(\zeta + e^{-\alpha T})\zeta}{(\zeta - e^{-\alpha T})^3} \\ & + \frac{1}{6}C_3 T^3 e^{-\alpha T}\frac{(\zeta^2 + 4e^{-\alpha T}\zeta + e^{-2\alpha T})\zeta}{(\zeta - e^{-\alpha T})^4} \\ & + \frac{B - \frac{1}{2}a_1 A}{\beta}\frac{\zeta e^{-\frac{a_1}{2}T}\sin\beta T)}{\zeta^2 - 2\zeta e^{-\frac{a_1}{2}T}\cos\beta T + e^{-a_1 T}} \\ & + A\frac{(\zeta - e^{-\frac{a_1}{2}T}\cos\beta T)\zeta}{\zeta^2 - 2\zeta e^{-\frac{a_1}{2}T}\cos\beta T + e^{-a_1 T}}\Bigg) \end{aligned} \tag{6.69}$$

where the coefficients are as in Appendix 6.B.

CHAPTER 7

Simulation of compound action potentials of a motor unit

THE smallest functional unit of a muscle is called a *motor unit* (MU). It is constituted by a set of muscle fibers, their innervating motoneuron in the spinal cord, and the interconnecting myelinated axon [85].

Simulating compound APs of a MU is thus obtained by summating individual muscle-fiber APs with respect to the selected recording electrode [53]. A virtual MU is constructed in compliance with anatomical knowledge and by the use of the line source model, c. f. Chapter 6. The simulation is fast and the resulting compound AP corresponds well to real recordings. A straight-forward three-step procedure may be designed as follows.

First, a MU with the appropriate number, diameter, distribution, etc, of muscle fibers is generated in accordance with the anatomic conditions to study. Then, the electrode selected for the investigation is imaginally inserted into the generated MU where the “ploughing effect” of the beveled tip is taken into account. Finally, the line source model is applied to each muscle fiber and their contributions are added to form the compound AP.

This chapter is organized to describe these steps in some detail and to show samples of simulated compound APs of different MUs. In Section 7.1, the anatomy of a normal MU is discussed as well the changes induced by disease. Section 7.2 presents the assumptions made regarding the insertion of the electrode. The line-source model is briefly summarized in Section 7.3. Finally, Section 7.4 presents sample simulation results.

7.1 Generating the MU

To generate realistic MUs and their compound APs, it is important that the constitution of the modelled MU reflects the constitution of real MUs. Below, the most fundamental aspects are presented along with the assumptions used in this thesis.

Depending on muscle, the fibers within each MU may be about 10 to nearly 2000 (typically a few hundred) in number [82] and randomly distributed in a cross section of about 2-10 mm in diameter [81]. The fiber diameters range from 5 to 90 μm where the mean diameter depends on muscle [25, pp. 475]. The conduction velocity depends on the fiber diameter and is of the order 1.5-6.5 m/s [78]. For bipennate muscles, the endplates are located in well-defined zones [7] whereas a more complex pattern is observed in multipennate muscles. The standard deviation of the variability in the delay of an individual motor endplate is ranging between 5 and 40 μs [86].

The total number of fibers and the total area of the MU are related through the *mean fiber concentration* (MFC) that specify the number of fibers found, on average, in an area of one square millimeter. In a normal and healthy muscle the MFC is about 5 fibers/mm^2 [83].

The default parameters used in this thesis are presented in Table 7.1.

Table 7.1: Default MU parameters

Description	*Value*	
Diameter of MU territory	6	mm
Number of muscle fibers	141	
MFC	5	fibers/mm^2
Diameter distribution (Gaussian), mean	55	μm
Diameter distribution (Gaussian), SD	3	μm
Endplate distribution (Rectangular), mean	0	mm
Endplate distribution (Rectangular), range	± 1	mm
Jitter distribution (Gaussian), mean	0	μs
Jitter distribution (Gaussian), SD	15	μs

7.1.1 Testing the model

The model has been tested in a number of ways in order to assess its functionality and its similarity with real EMG recordings. The performance seems to be excellent as reported in [83] and [84]. The obtained signals are reasonably realistic, and certainly sufficiently similar to real recordings to be used to study relationship between muscle parameters and recorded signals.

7.1.2 Changes in disease

The number of muscle fibers in a given MU is affected by disease. It may decrease in primary muscle disorders, or it may increase as a compensatory phenomenon when individual axons disappear due to nerve damage. Surviving axon then produce new branches and contact those muscle fibers that have lost contact with the original nerve axon. Thus, both the number and the cross-sectional distribution of the muscle fibers within a particular MU change with disease [85].

The changes in neuromuscular disorders may be divided into three basic categories:

Myopathies These diseases cause a loss or impairment of muscle fibers. In general the number of fibers within the MU decrease and fat or fibrous tissue is increased. There is an increased fiber diameter variation, generation of new fibers, and sometimes longitudinal splitting of muscle fibers.

Neuropathies These diseases cause a loss of motoneurons or axons. In a compensatory process, neighboring axons branch off and reinnervate the orphaned muscle fibers through *collateral sprouting*. Hence, the number of MUs decreases but their size in terms of number of fibers increases. The individual fibers may show an increased fiber diameter variation.

Neuromuscular junction diseases These diseases affect various key components that are vital for the function of the MU. If, e.g., the acetylcholine receptors in the motor endplate are reduced in number, the function of the muscle may be severely impaired. This may cause the variability of the synaptic delay, the jitter, to increase, or the triggering to sometimes fail, a condition called *impulse blocking*.

As a rule of thumb, myopathies are manifested by small MUAPs and often the components of the MUAPs are dispersed in time. Neuropathies are manifested by large MUAPs with multiple peaks corresponding to the increased number of fibers. The neuromuscular transmission diseases may often be diagnosed by the variation in the MUAP in consecutive discharges through studies of, e.g., the jitter and/or the blocking.

Principally, the random distribution of the muscle fibers in normal, healthy muscle become more and more grouped as the level of myopathy/neuropathy increases. As shown in [84] parameters such as amplitude and area, for example, depend linearly on the MFC both with random and grouped distribution, albeit with a different slope coefficient.

Therefore, in order to simplify the simulations of myopathic/neurogenic MUs, the muscle fibers are distributed randomly in the simulated MUs where only the MFC reflects the level of myopathy/neuropathy. This seems to be a reasonable simplification as long as only relative comparisons are being made.

7.2 Insertion of the electrode

The selected recording electrode, c. f. Section 6.3, is assumed to be inserted perpendicularly to the fiber direction. In real muscle, the beveled tip of the electrode causes a ploughing effect that moves the fibers to the side. This is incorporated into the model by moving the fibers situated in the area occupied by the electrode.

7.3 Applying the line-source model

The line-source model, c. f. Chapter 6, is applied to each fiber within the MU where the position of the electrode with respect to each muscle fiber is considered. The individual contributions are then summed to form the compound AP.

In this thesis, we neglect the so-called *wall effect* [27], observed as an amplification of the AP amplitude in recordings with the SF electrode positioned close to the fiber. The *shadow effect*, manifested by a decrease in amplitude and change of shape of muscle fibers located on the back of the electrode (SF and CN electrodes only) [88] is approximated by only summing the contribution from the muscle fibers located in the semicircle in front of the active surface.

7.4 Simulations

As examples of how the model may be used, three sample MUs are generated in this section. The MU parameters are in accordance with Table 7.1 except for the MFC value that is different for each generated MU.

The MFC value was selected as 5, 2, and 10 fibers/mm^2, to simulate a normal, myopathic, and neurogenic MU, respectively. The CN electrode was selected as

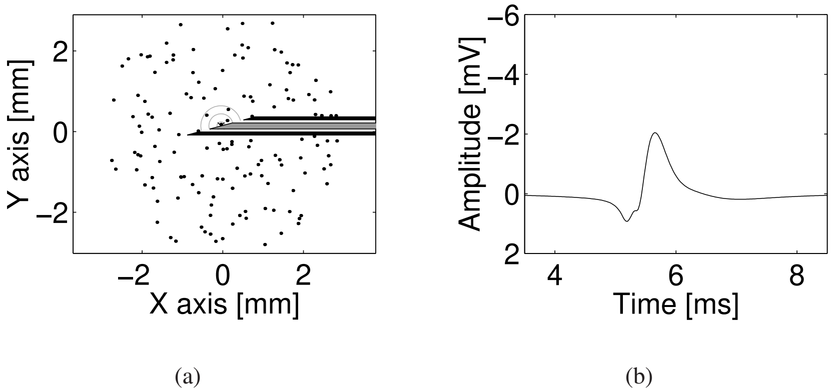


Figure 7.1: A normal MU simulated with MFC = 5 fibers/mm^2 and as if recorded using a CN electrode. The diagrams show (a) the cross section of the MU with the electrode inserted, and (b) the resulting MUAP as recorded by the CN electrode. The radial distances 100, 300, and 500 μm are indicated by semicircles.

recording electrode.

7.4.1 A normal MU

Figure 7.1 shows a sample result from a simulation of a normal MU with MFC = 5 fibers/mm^2. The selected CN electrode is shown inserted in the cross section of the MU. The recording through the CN electrode produces a MUAP as shown in the figure.

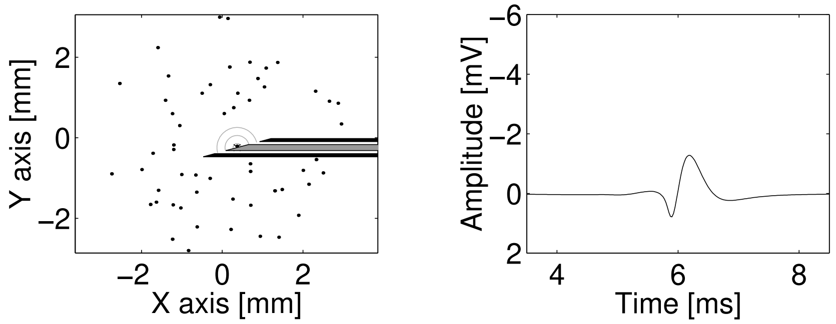
7.4.2 A myopathic MU

In Figure 7.2 a sample result from a simplified simulation of a myopathic MU where MFC = 2 fibers/mm^2. In a real myopathic MU, the diameter of the fibers may change and even produce clusters of fibers through longitudinal splitting, see Section 7.1.2.

The selected CN electrode is shown inserted in the cross section of the MU and the resulting MUAP is shown in the figure. Note that the amplitude is reduced compared to the normal MU.

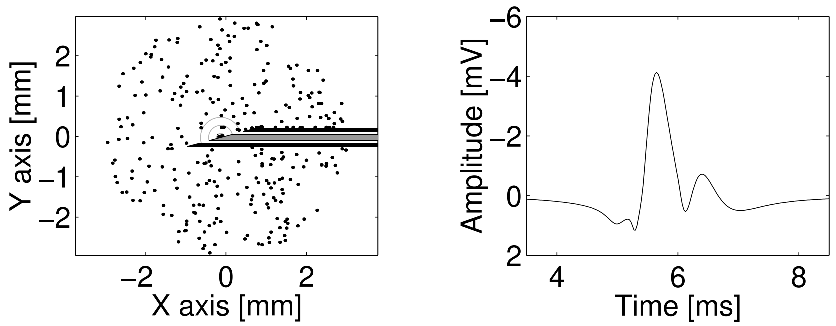
7.4.3 A neurogenic MU

A sample result from a simplified simulation of a neurogenic MU is shown in Figure 7.3 where MFC = 10 fibers/mm^2. In a real neurogenic MU, the fibers are



(a) (b)

Figure 7.2: A simplified simulation of a myopathic MU where MFC = 2 fibers/mm^2. The resulting compound AP is generated as if recorded by a CN electrode. The diagrams show (a) the cross section of the MU with the electrode inserted, and (b) the resulting MUAP as recorded by the CN electrode. The radial distances 100, 300, and 500 μm are indicated by semicircles.



(a) (b)

Figure 7.3: A simplified simulation of a neurogenic MU where MFC = 10 fibers/mm^2. The resulting compound AP is generated as if recorded by a CN electrode. The diagrams show (a) the cross section of the MU with the electrode inserted, and (b) the resulting MUAP as recorded by the CN electrode. The radial distances 100, 300, and 500 μm are indicated by semicircles.

grouped in clusters as an effect of the reinnervation which also affects the jitter, see Section 7.1.2.

The selected CN electrode is shown inserted in the cross section of the MU and the resulting MUAP is shown in the figure. Note the increased amplitude as compared to the normal MU as well as the extra peak at $t = 6.5$ ms. Note also the clustering of fibers close to the cannula on its upper side which is a result of the ploughing effect.

7.5 Discussion

A model for generating MUs and simulating the compound AP is presented. A number of parameters corresponding to the constitution and function of the MU may be changed in order to simulate normal and diseased muscle.

As shown in a related paper [84], the generated EMG signals obtained with the three electrodes mentioned here are reasonably similar to the compound APs obtained in real recordings. Expected changes in the signal is obtained when different parameter values are manipulated. When it comes to the simulation of some pathological situations, the model allows simulation of the most commonly encountered situations [84].

To simplify the simulation of abnormal MUs in this presentation, only the MFC value was changed to yield myopathic and neurogenic MUs. The validity of this simplification is supported by the findings in [84] where parameters such as amplitude and area, for example, depend linearly on the MFC both when appropriately modeling the abnormal MU and when assuming a random distribution. The only difference is the slope coefficient which should not matter as long as the model is used for comparing relative differences only.

In the more elaborate study described in [84], this model has been used in situations where fiber diameter varaition and grouped distribution of fibers have been included as well.

CHAPTER 8

Deconvolving motor unit action potentials

DIAGNOSIS of muscle and nerve disorders is often related to details in the constitution and function of the *motor unit* (MU). The distribution and number of muscle fibers, for example, in a given MU are affected by disease [85] and, hence, provides valuable information in this respect. Functional studies of the neuromuscular transmission is yet another example.

One way to retrieve this type of information is to extract a small piece of muscle tissue for inspection in the microscope, so-called *muscle biopsy*. Another way, less invasive, is to measure the electrical activity in the muscle, the so-called *electromyogram* (EMG), that is generated upon activation of the muscle.

Using a special recording electrode, the EMG may be retrieved and used as input to a computer for signal processing and analysis. Depending on the electrode used, see Chapter 6, different aspects and "resolutions" of the underlying information is obtained [80].

The *concentric needle* (CN) electrode [1] [85] is used to acquire the so-called *motor-unit action potential* (MUAP). The shape and amplitude of the MUAP reflects the number of muscle fibers, the synchronicity of the *action potentials* (APs), and muscle fiber concentration within the pick-up distance. Much is known about the relationship between the characteristics of the MUAP (shape, amplitude, etc) and the underlying type of disorder [85]. With this knowledge, accurate diagnostic methods have successfully been derived and are used in clinical routine throughout the world.

The different approaches to improve the EMG signal for analysis may be divided into two categories. In the first category, the characteristics of the electrode is changed in order to "process" the EMG signal already in the muscle. The *single* *fiber* (SF) electrode [28] and the *Macro* electrode [79] are examples within this category, and are used in different situations [80]. To further improve the EMG methods, alternative electrode types could be developed with, e.g., multiple recording surfaces [100] and even with signal (pre-) processing integrated into the recording electrode [49].

In the second category, signal processing is applied on the already recorded signal. An obvious example in this category is filtering in order to reduce disturbances and irrelevant signal properties. A note of precausion is necessary here, though. One important factor to have in mind is that the level of success of a signal processing method crucially depends on the quality of the recorded signal and, hence, the prioritizations made in the first category. If information is lost during the retrieval of the signal, it cannot be restored.

In this chapter we explore the possibility of discerning the individual AP components of a MUAP acquired with the CN electrode. By doing this, the applicability of CN EMG in clinical routine may be improved and widened. In particular, we consider non-causal deconvolution filters with two different levels of deconvolution where the *partial* deconvolution filter only takes the membrane properties of the muscle fiber into account and produces a signal where each individual AP component is transformed into a *monophasic* impulse that is strictly positive with a single maximum. The *full* deconvolution filter takes the electrode characteristics into account as well and produces a signal with narrow (monophasic) impulses, one for each individual AP.

There are two main advantages of this approach. First, because the monophasic signals effectively remedy the canceling that may occur in the original EMG signal, we anticipate our approach to show an improved performance in estimating, e.g., the number of fibers or the *mean fiber concentration* (MFC), compared to an approach using the original multiphasic signal. Second, we anticipate the fully deconvolved signal to further contribute to this through its (possible) resolution of individual fibers. This may be advantageous, not only for measuring MFC but maybe also for jitter measurements.

This chapter pursues these issues further and is organized as follows. In Section 8.1, the prerequisites for recording the EMG signal and obtaining the MUAPs from the EMG signal are described. For the deconvolution, we have selected the Wiener filter, c. f. Chapter 5. The required signal and noise models are the topics of Section 8.2. In Section 8.3, three examples of the deconvoluted signals are presented and compared to highpass filtering. Different approaches to the estimation of the muscle fiber concentration within the MU are presented in Section 8.4 along with simulations of their performance. In Section 8.5, the fully deconvolved MUAP is used for estimating the jitter where the MHT method, c. f. Chapter 4, and Kalman filtering, c. f. Chapter 3, are used to discriminate between APs originating from different fibers.

8.1 Prerequisites

To successfully record the EMG signal and extract the desired MUAPs, several practical issues need to be solved. In this section we state our assumptions regarding the most important ones.

8.1.1 Recording the EMG

Prior to the sampling, the EMG signal must be prefiltered using a bandpass filter in hardware. The very-low-frequency highpass filter (VLF-HP) removes the DC levels and the low frequency components dominated by the activity from distant muscle fibers not of interest for the recording. The high-frequency lowpass filter constitutes an anti-aliasing lowpass filter (AA-LP) that removes the frequencies above the Nyquist frequency.

For clarity and to simplify the design of the algorithms, we make the following assumptions regarding the EMG signal:

Assumption 8.1 *Filter Order – The VLF-HP filter is of second order and the AA-LP filter is of fourth order.*

Assumption 8.2 *Additive Noise – All signal energy not originating from the MUAP is regarded as noise.*

Assumption 8.3 *No quantization Errors – The quantization errors imposed by the analog-to-digital converter (ADC) are negligible compared to the noise.*

Assumption 8.4 *Linear System – The recorded signal is not subject to nonlinear distorsion, e.g., amplifier overload, aliasing, nonlinear hardware, etc.*

8.1.2 Extracting the MUAPs

In the recorded EMG signal, several MUs may be present simultaneously and occasionally overlap. The complexity of the EMG signal, i.e., the number of MUs simultaneously active and their discharge rate, increases with increased muscle force level [59]. There are algorithms available that decompose EMG signals of different complexity into individual MUAPs [33] [58] and we do not pursue this issue here.

Based on these considerations we assume the following regarding the MUAP extraction:

Assumption 8.5 *Perfect MUAP Extraction – The decomposed MUAPs are not distorted by MUAPs originating from other MUs.*

8.2 Algorithm overview

Information about the MU in terms of number of fibers and the temporal dispersion of the APs is important for diagnosis of disease. This information is not directly obtainable from the MUAP for two main reasons:

1. *Cancellation* – The individual APs are multiphasic and their contributions may partly cancel in the summation MUAP.

2. *"Wideness"* – The width of the individual APs are large compared to their separation in time. Thus, the APs normally overlap which complicates the study of individual APs.

To improve upon this situation, different types of lowpass and highpass filters and/or different types of electrodes have been applied to the acquired EMG signal.

Filtering approaches have been applied to the CN MUAP to study its stability at consecutive discharges [65]. Lately, filtering approaches applied to the CN MUAP have attained an increased interest due to a desire to replace the costly SF electrode with the much cheaper CN electrode for jitter measurements [87] [32]. In addition to the economical advantages, there is the sanitary advantage with the disposable CN electrodes available, as well as the practical advantage of using one type of electrode only.

The traditional approaches to improving the MUAP signal have been to first determine the filter type and then set out to tune its parameters to optimize its (constrained) performance. We generalize upon this idea and seek the optimal *linear* filter. This imposes two problems. First, since the filtered signal is input to a final, possibly nonlinear, processing step, it may be impossible to find a quantitative measure of the filter's optimality. Second, we seek a model based filter design so that, once we have found a proper model, we are able to produce filters for virtually any MUAP without the need of an extensive design procedure.

Based on these considerations, we have selected the Wiener filter design method, see Chapter 5, not primarily because the of its mean squared error (MSE) optimality but rather because there is a clear and intuitive correspondance between the tuning considerations and the produced output. The entities used in the tuning of the Wiener deconvolution filter is a signal model and a noise model, whose parameters are to be set to give a filter with desired performance.

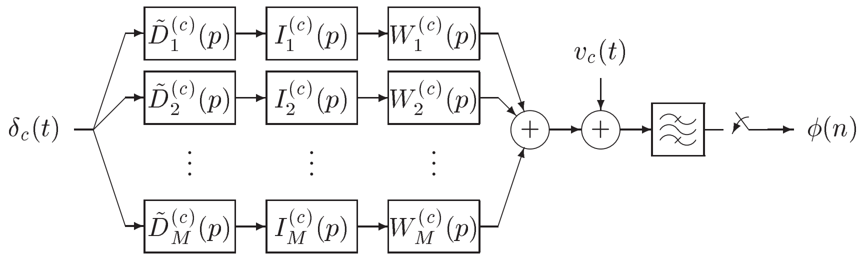


Figure 8.1: A block diagram of the synthesis model.

Depending on the application, we apply the deconvolution provided by the Wiener filter in two different levels that we call *partial* and *full* deconvolution, see below. For this, we need a model of the signal generation, or *synthesis*. By combining the ideas from Chapter 6 and Chapter 7, we have compiled the synthesis model presented next.

8.2.1 Synthesis model

The MUAP measured by the recording equipment is the summed contribution of all active muscle fibers within the MU, c. f. Chapter 7. Each individual fiber produces an AP that may be modelled as a convolution of a muscle fiber membrane current $i_c(t)$ and a positional and electrode dependent weighting function $w_c(t)$, c. f. Chapter 6. The fibers in the MU are all triggered by a single axon via the motor endplate where the trigger impulse is given a variable delay stochastically, the jitter. The AP is further "delayed" by an amount that depends on the axial position of the motor endplate relative to the recording electrode and the conduction velocity of the fiber.

In addition to the MUAP under study, the measurement contains thermal noise and disturbances as well as distant MUAPs, or *background activity*, collectively referred to as noise. Under Assumption 8.5, we do not need to account for other high amplitude MUAPs close to the electrode.

These considerations are presented in Figure 8.1, where p is the derivative operator and $\tilde{D}_m^{(c)}(p)$ is a stochastic delay operator of fiber m. It accounts for the stochastic delay in the motor endplate and the additional, deterministic "delay" described above. The transfer functions $I_m^{(c)}(s)$ and $W_m^{(c)}(s)$ model the transmembrane current and the electrode and positional dependent weighting function, respectively. All contributions are summed together along with the continuous-time noise $v_c(t)$. Finally, the recording equipment applies a bandpass filter in hardware prior to the sampling of the discrete-time MUAP $\phi(n)$.

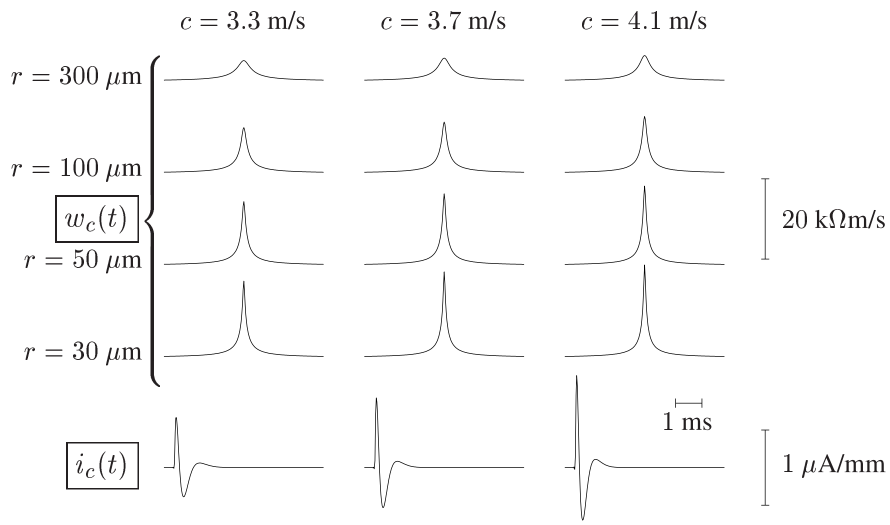


Figure 8.2: The figure illustrates the transmembrane current $i_c(t)$ and the CN electrode weighting function $w_c(t)$ for different conduction velocities c and radial distances r. The transmembrane current depends on conduction velocity alone whereas the weighting function depends on the radial distance as well.

This may be summarized in the time domain as

$$\phi(n) \stackrel{\triangle}{=} \phi_c(nT_s) \tag{8.1}$$

$$\phi_c(t) = H_{BP}(p)\left\{\sum_{m=1}^{M} W_m^{(c)}(p) I_m^{(c)}(p) \tilde{D}_m^{(c)}(p)\delta_c(t) + v_c(t)\right\} \tag{8.2}$$

where T_s is the sampling period and $H_{BP}(s)$ is the hardware bandpass filter.

In the frequency domain the above model becomes

$$\Phi(\zeta) = \Phi_c(f_s \operatorname{Log} \zeta) \tag{8.3}$$

$$\Phi_c(s) = H_{BP}(s)\left\{\sum_{m=1}^{M} W_m^{(c)}(s) I_m^{(c)}(s) \tilde{D}_m^{(c)}(s) + V_c(s)\right\} \tag{8.4}$$

where we in (8.3) have assumed that the bandpass filter $H_{BP}(s)$ removes all signal energy above the Nyquist frequency $f_s/2$ where f_s is the sampling frequency.

Figure 8.2 illustrates the transmembrane current and the CN electrode weighting function. As the figure shows, the amplitude of the transmembrane current increases as the conduction velocity (fiber diameter) increases. Moreover, we see that the weighting function "flattens out" when the radial distance increases.

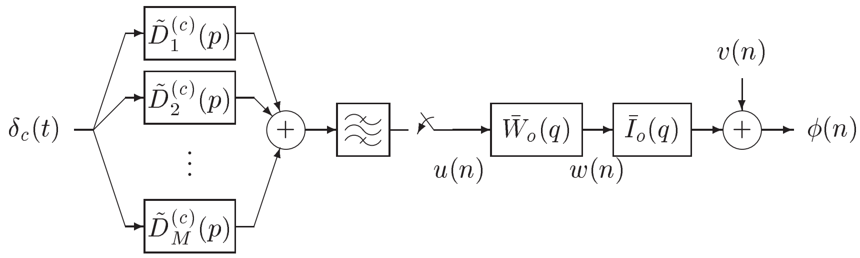


Figure 8.3: A block diagram of the deconvolution model.

8.2.2 Deconvolution model

While the model (8.1)-(8.4) is a good approximation of the practical situation, it is not very helpful in designing a Wiener filter to deconvolve the measured signal $\phi(n)$ because the individual fibers have different model characteristics. Using a filter bank approach, where each filter is tuned to enhance the contribution from a certain fiber and suppress the contribution from all others, is not suitable because the transfer functions, $I_m^{(c)}(s)$ and $W_m^{(c)}(s)$, are too similar between the fibers, see Figure 8.2. Therefore, we need to find an approximative model that is more suitable for our goal.

First, however, we make the observation that deconvolving the measured signal $\phi(n)$ using a Wiener filter is principally no different from filtering $\phi(n)$ with any other linear filter. Hence, it is possible to design a Wiener filter that is as good as *any* other linear filter in providing the desired output. The only requirement is that we should be able to replace the M different muscle fiber models $I_m^{(c)}(s)$ and $W_m^{(c)}(s)$ with "optimal" *design* models $\bar{I}_o(\zeta)$ and $\bar{W}_o(\zeta)$ as shown in Figure 8.3. The order of the two models are the opposite from Figure 8.1 because we later need $w(n)$. The order reversal is possible since the models are time invariant.

This is summarized in the time domain by

$$\phi(n) \approx \bar{I}_o(q)w(n) + v(n) \tag{8.5}$$

$$w(n) \triangleq \bar{W}_o(q)u(n) \tag{8.6}$$

$$u(n) \triangleq u_c(nT_s) \tag{8.7}$$

$$u_c(t) \triangleq H_{BP}(p) \sum_{m=1}^{M} \tilde{D}_m^{(c)} \delta_c(t) \tag{8.8}$$

where $v(n)$ is the discrete-time counterpart of $v_c(t)$, $u(n)$ is a (possibly) non-equidistant pulse train and $w(n)$ is typically less "spiky".

In the frequency domain we obtain

$$\begin{aligned}
\Phi(\zeta) &\approx \bar{I}_o(\zeta)W(\zeta) + V(\zeta) && (8.9)\\
W(\zeta) &= \bar{W}_o(\zeta)U(\zeta) && (8.10)\\
U(\zeta) &= U_c(f_s \operatorname{Log} \zeta) && (8.11)\\
U_c(s) &= H_{BP}(s) \sum_{m=1}^{M} \tilde{D}_m^{(c)}(s) \quad . && (8.12)
\end{aligned}$$

From a modeling perspective, this model could seem as a crude approximation to the real situation. Whether this is true or not is here irrelevant because the model (8.5)-(8.12) is only used for the *design* of the desired filter and not for modeling the measured signal $\phi(n)$. Moreover, if there exists a linear filter with optimal performance, it can be formulated as a Wiener filter by defining the design models appropriately.

In this particular case, this approach is particularly advantegous because the models for different muscle fibers are quite similar, as shown in Figure 8.2. By formulating the design models as a nominal model complemented with an uncertainty model, robust filtering provide a stringent way of obtaining the optimal filter [90] [64].

8.2.3 Noise model

The noise model is based partly on physical knowledge about the recordings and partly on identification from real signals. From this analysis, we have found it suitable to divide the noise model into five parts.

1. *Very low frequency highpass (VLF-HP) filter* – This part models the highpass filter in the recording equipment. In accordance with Assumption 8.1, we assume a second order model.

2. *Low frequency lowpass (LF-LP) filter* – This part models the LF components in the measured signal. These disturbances are most remarkably noticeable and are important to take into account. We use a second order LF-LP filter for this purpose.

3. *Resonance peak (RP) filter* – This part models the hum from the power lines and is simply a second order resonance peak.

4. *Medium frequency lowpass (MF-LP) filter* – This part models the roll-off of the background activity. It is modeled using a first order, pole only MF-LP filter.

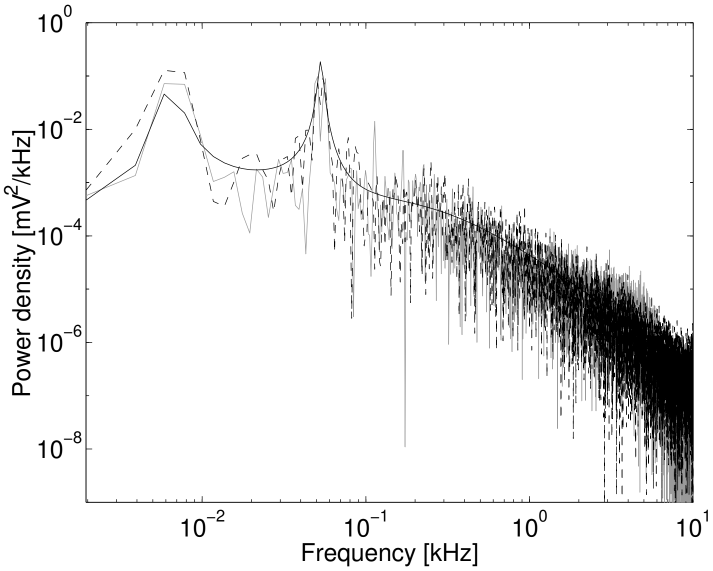


Figure 8.4: The spectral power density of the noise retrieved from a real recording (gray) and the noise model (solid). An example realization (dashed) is also included. As the figure shows, the noise has mainly a lowpass character. In addition, the spectrum contains two peaks of which the first one are the remains of the VLF components and the second one is hum from the powerlines.

5. *High frequency lowpass (HF-LP) filter* – This part models the roll-off present in the analyzed signal. It is modeled using a fourth order, pole only HF-LP filter.

These considerations may be summarized in the following model

$$H_v(\zeta) \triangleq H_1(\zeta)H_2(\zeta)H_3(\zeta)H_4(\zeta)H_5(\zeta) \tag{8.13}$$

where $H_v(\zeta)$ is the transfer function of the noise model and $H_i(\zeta)$ corresponds to each of the filters listed above.

The spectral power density $P_v(\zeta)$ is then calculated using

$$P_v(\zeta) \triangleq |H_v(\zeta)|^2\eta \quad [\text{mV}^2/\text{kHz}] \tag{8.14}$$

where η is the power density of the noise.

Figure 8.4 shows the spectral power density of the noise $v(n)$ (retrieved from a real recording) and the tuned noise model $P_v(\zeta)$ using the model parameters listed in Table 8.1. An example realization is also included. Clearly, there is a good agreement between the model and the measured noise in this particular recording. For other recordings, the noise model may need to be retuned, but the structure of the model is expected to be widely applicable.

Table 8.1: Parameters for the continuous-time noise model

Description	*Parameter*		*Value*
Noise power density	η	[mV^2/kHz]	0.001
VLF-HP zero (2nd order)	$\frac{1}{2\pi}s_{B_1}$	[kHz]	0
VLF-HP pole (2nd order)	$\frac{1}{2\pi}s_{A_1}$	[kHz]	-0.0006
LF-LP zero (2nd order)	$\frac{1}{2\pi}s_{B_2}$	[kHz]	-0.005
LF-LP pole (2nd order)	$\frac{1}{2\pi}s_{A_2}$	[kHz]	$-0.0003 \pm 0.0065i$
RP zero (2nd order)	$\frac{1}{2\pi}s_{B_3}$	[kHz]	$-0.032 \pm 0.053i$
RP pole (2nd order)	$\frac{1}{2\pi}s_{A_3}$	[kHz]	$-0.0018 \pm 0.053i$
MF-LP pole (1st order)	$\frac{1}{2\pi}s_{A_4}$	[kHz]	-0.3
HF-LP pole (4th order)	$\frac{1}{2\pi}s_{A_5}$	[kHz]	-8

8.2.4 Wiener filter tuning

Based on the design models, $\bar{I}_o(\zeta)$ and $\bar{W}_o(\zeta)$, two different Wiener filters, $G_i(\zeta)$ and $G_{iw}(\zeta)$, may be derived, depending on which of the two inputs $u(n)$ and $w(n)$ we want to estimate, c. f. Chapter 5. By defining

$$\hat{w}(n) \stackrel{\triangle}{=} G_i(q)\phi(n) \tag{8.15}$$

$$\hat{u}(n) \stackrel{\triangle}{=} G_{iw}(q)\phi(n) \tag{8.16}$$

where q is the forward shift operator, we call $\hat{w}(n)$ the *partially deconvolved signal* and $\hat{u}(n)$ the *fully deconvolved signal*.

The two Wiener filters $G_i(\zeta)$ and $G_{iw}(\zeta)$ that provide MSE-optimal estimates of (8.15) and (8.16) are then given by, see Chapter 5,

$$G_i(\zeta) = \begin{cases} 0, & \zeta = 1 \\ \frac{\bar{W}_o(\zeta)\bar{W}_o^*(\zeta)\bar{I}_o^*(\zeta)}{|\bar{W}_o(\zeta)\bar{I}_o(\zeta)|^2 + P_v(\zeta)T_v}, & \text{otherwise} \end{cases} \quad [\text{k}\Omega \text{ mm/(mV ms)}] \tag{8.17}$$

$$G_{iw}(\zeta) = \begin{cases} 0, & \zeta = 1 \\ \frac{\bar{W}_o^*(\zeta)\bar{I}_o^*(\zeta)}{|\bar{W}_o(\zeta)\bar{I}_o(\zeta)|^2 + P_v(\zeta)T_v}, & \text{otherwise} \end{cases} \quad [1/(\text{mV ms})] \tag{8.18}$$

where * denotes the complex conjugate.

For the work presented in this chapter, we use a pragmatic approach to find reasonably correct design models $\bar{I}_o(\zeta)$ and $\bar{W}_o(\zeta)$. We simply assume $\bar{I}_o(\zeta) \in I(\zeta)$ and $\bar{W}_o(\zeta) \in W'(\zeta)$ where $W'(\zeta)$ is a modified version, see Appendix 8.A, of $W(\zeta)$ defined in Chapter 6, c. f. (6.22). We then test several realizations of $I(\zeta)$ and $W'(\zeta)$, design the Wiener filters, apply them to a set of test APs that span the range of fibers under study, and select the realizations that yields the best overall performance in terms of narrow monophasic signals in the fully deconvolved signal $\hat{u}(n)$ and no "strange" effects in the partially deconvolved signal $\hat{w}(n)$. See Appendix 8.A for details.

With this approach, we have selected

$$r_o = 30 \quad [\mu\text{m}] \tag{8.19}$$

$$c_o = 3.9 \quad [\text{m/s}] \tag{8.20}$$

$$d_o = 59 \quad [\mu\text{m}] \tag{8.21}$$

where r_o is the radial distance, c_o is the conduction velocity, and the fiber diameter d_o is given by c_o through the linear relation (6.8).

In a practical application where (near) optimal performance is strived for, a more thorough tuning of the models $\bar{W}_o(\zeta)$ and $\bar{I}_o(\zeta)$ may be needed. With the framework given by the Wiener filter design, an intuitive feedback of the characteristics of a certain model is available which simplifies the tuning process.

Figure 8.5 shows the spectral energy density of the design models $\bar{I}_o(\zeta)$ and $\bar{W}_o(\zeta)$, and the resulting Wiener filters $G_i(\zeta)$ and $G_{iw}(\zeta)$ for the selected noise level $5 \cdot 10^{-3}$ mV^2/kHz, conduction velocity $c = 3.3$, and radial distance $r = 100$. As the figure shows, the filters amplify high frequencies to enhance the information present there while low frequencies are attenuated because the noise is dominating for these frequencies.

8.2.5 Refinements

Under ideal circumstances, the fully deconvolved signal $\hat{u}(n)$ is a sum of stochastically delayed impulses. The spectrum of such a signal is constant for all frequencies but due to the bandpass filter, see Figure 8.1, the spectrum is zero above the Nyquist frequency. Hence, if we consider the ideal case with no noise, correct

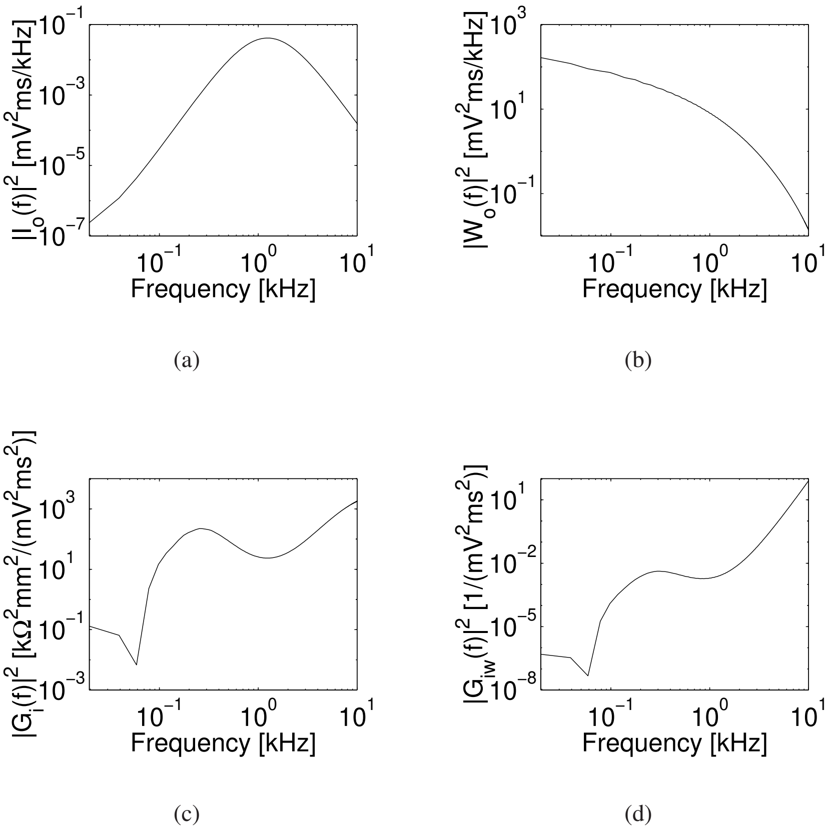


Figure 8.5: The figure shows the spectral energy density of the design models (a) $\bar{I}_o(\zeta)$ and (b) $\bar{W}_o(\zeta)$, and the resulting Wiener filters (c) $G_i(\zeta)$ and (d) $G_{iw}(\zeta)$.

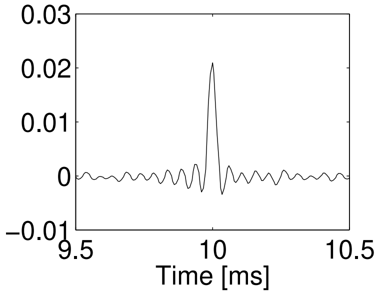
model, optimal bandpass filter, etc, the fully deconvolved signal $\hat{u}(n)$ is given by

$$\hat{u}(n) = \sum_{m=1}^{M} \tilde{D}_m^{(c)}(p) \frac{\sin(2\pi B t)}{\pi t}\bigg|_{t=nT_s} \tag{8.22}$$

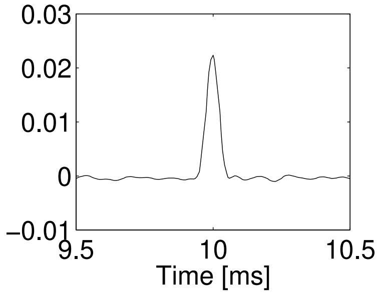
$$= \sum_{m=1}^{M} \frac{\sin(2\pi B(nT_s + \tilde{\Delta}_m)}{\pi n T_s} \tag{8.23}$$

where $\tilde{\Delta}_m$ is a continuous stochastic variable with mean and variance given by the stochastic delay operator $\tilde{D}_m^{(c)}(p)$.

As (8.23) and Figure 8.6 (a) show, the fully deconvolved signal $\hat{u}(n)$ contains a lot of erroneous side lobes because $\tilde{\Delta}_m$ is in general not a multiple of the sampling period T_s. To alleviate this effect, we adjust the corresponding Wiener filter



(a)



(b)

Figure 8.6: The fully deconvolved signal of an AP obtained with the CN electrode (a) without frequency domain windowing, and (b) Hanning window applied in the frequency domain. The conduction velocity was $c = 3.9$ m/s, the radial distance was $r = 30$ μm, and the noise power density was $\eta = 0.001$ mV^2/kHz.

by applying a Hanning window in the frequency domain. Hence, we replace the Wiener filter $G_{iw}(\zeta)$ of the full deconvolution with

$$G'_{iw}(\zeta) \quad \stackrel{\triangle}{=} \quad \mathcal{T}'(\zeta)G_{iw}(\zeta) \quad [1/(\text{mV ms})] \tag{8.24}$$

where the Hanning window $\mathcal{T}'(\zeta)$ is normalized to unit energy and has its maximum for $\zeta = 1$. As Figure 8.6 (b) shows, this refinement reduces the side lobes effectively but at the cost of a wider peak.

8.2.6 Simulation parameters

All simulations in this chapter uses the model parameters listed in Table 8.2 unless stated otherwise. The parameters not listed in the table were identical to those used in Chapter 7.

8.3 Deconvolution

In this section we show the characteristics of the deconvolution approach presented above when applied to modelled APs obtained with a CN electrode. The performance is compared to an approach similar to the one described in [32] where a highpass filter was applied to a MUAP in order to improve its resolution.

Three different fibers were simulated with conduction velocity c = 3.3, 3.7, and 4.1 m/s, respectively. The results when applying a highpass filter, the partial

Table 8.2: Deconvolution parameters for simulation

Description	*Parameter*		*Value*
Sample frequency	f_s	[kHz]	40
Noise power density	η	[mV^2/kHz]	0.005
Length of fiber	L_f	[mm]	150
Absolute jitter (SD)	σ_j	[μs]	15
Conduction velocity	c	[m/s]	3.7
Radial distance	r	[μm]	100
Axial position	z_0	[mm]	40
Radial distance, deconvolution	$\bar{r}_o$	[μm]	30
Conduction velocity, deconvolution	$\bar{c}_o$	[m/s]	3.9

deconvolution filter, and the full deconvolution filter to these CNAPs are presented below.

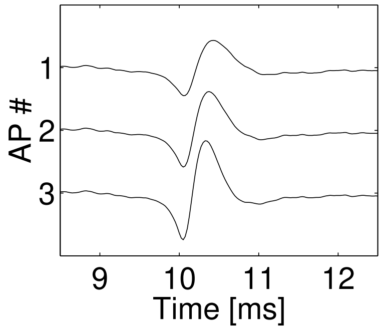
8.3.1 Highpass filtered CNAP

The main advantages of using the SF electrode compared to the CN electrode is that the recorded APs superimpose to a lesser extent (shorter width) and suppress distant APs more efficiently. Hence, the SF signal is directly suitable to estimate the neuromuscular jitter because relatively simple algorithms may be used for the analysis.

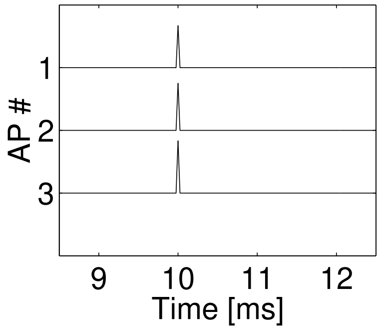
The advantage of the CN electrode, on the other hand, is its low price and that it is disposable. To improve the suitability for jitter measurements of the APs obtained using a CN electrode, a highpass filter may be applied that shortens the width of the recorded APs as well as suppresses more distant fibers [87] [32].

In Figure 8.7 this approach is presented on three different APs obtained using the CN electrode. The figure shows the original AP and two highpass filtered versions using second order highpass Butterworth filters with cut-off frequencies at 500 Hz and 2 kHz, respectively. As a comparison, the "ideal" result where each AP corresponds to a single impulse is included.

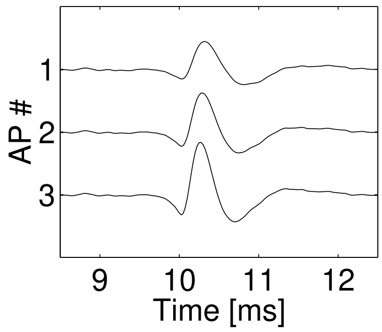
As the figure shows, the highpass filters indeed shorten the width of the APs as desired, but they also add new phases to the signal (both a dip and an artificial peak). In a more complex situation where several APs are recorded, cancellation



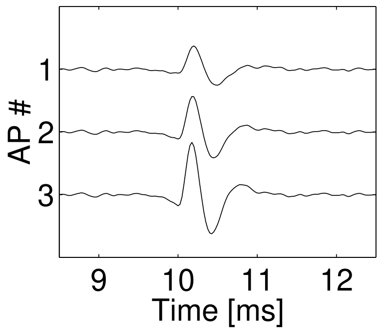
(a)



(b)



(c)



(d)

Figure 8.7: Sample results from a simulation of three APs with the conduction velocities c = 3.3, 3.7, and 4.1 m/s, respectively. In each diagram, we see (a) the unfiltered CNAP, (b) the impulses we want to estimate, (c) the highpass filtered CNAP using a cut-off frequency of 500 Hz, and (d) the highpass filtered CNAP using a cut-off frequency of 2 kHz. Both highpass filters were second order Butterworth filters.

may occur unless the APs are sufficiently separated in time. This place an upper limit on the suitability of this approach.

8.3.2 Partially deconvolved CNAP

Considering the multiphasic shape of the APs and the accompanying risk of cancellation, we know from Chapter 6 that the transmembrane current is the source of this. By separating and removing the contribution of the transmembrane current, a

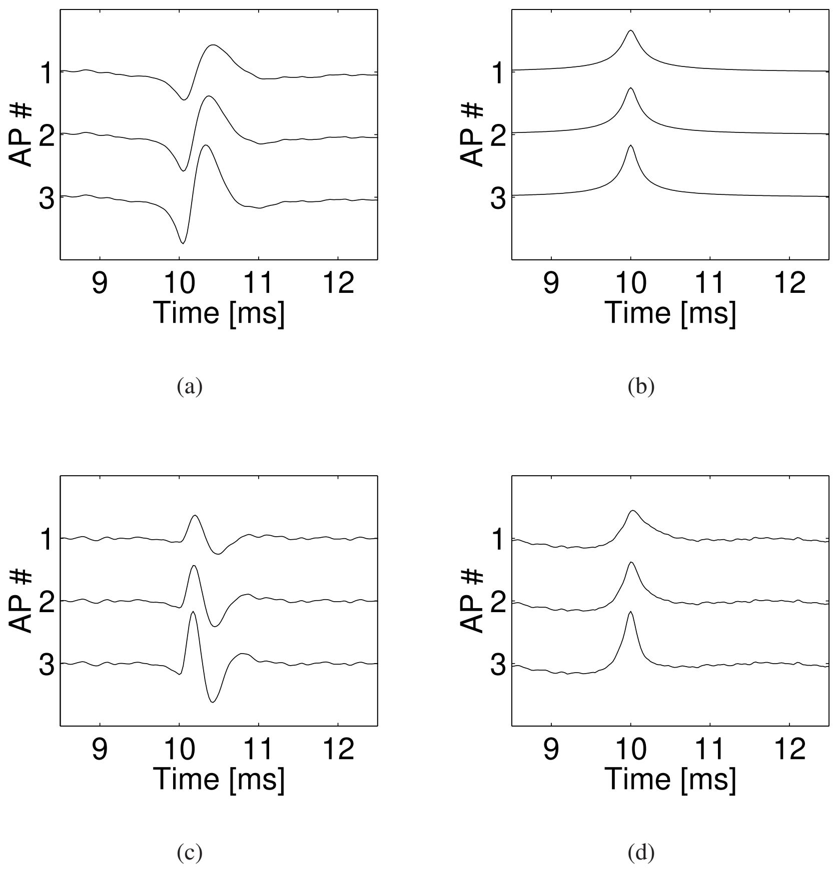


Figure 8.8: Sample results from a simulation of three APs with the conduction velocities c = 3.3, 3.7, and 4.1 m/s, respectively. In each diagram, we see (a) the unfiltered CNAP, (b) the weighting functions we want to estimate, (c) the highpass filtered CNAP using a second order Butterworth filter with a cut-off frequency of 2 kHz, and (d) the partially deconvolved signal $\hat{w}(n)$.

monophasic signal arises instead.

Our approach to this is the partial deconvolution described above where a Wiener filter is designed to deconvolve the APs and remove the contribution of the transmembrane current.

In Figure 8.8 the three different APs obtained using the CN electrode are shown together with the partially deconvolved signal where the highpass filtered (2 kHz) AP from Figure 8.7 is included as a comparison. The figure also shows the “ideal” result with the weighting function corresponding to each AP.

As the figure shows, all deconvolved signals are similar to the ideal result despite the Wiener filter is not optimally tuned to any of the APs. Moreover, the width is comparable with the highpass-filter approach while a monophasic output is produced. The slow decay is a problem, however, because it disturbs the jitter estimates unless the APs are sufficiently separated.

8.3.3 Fully deconvolved CNAP

By also removing the contribution of the weighting function, a narrow impulse is, ideally, obtained. In practice, however, the recording is bandpass filtered, the individual APs originate from fibers at different distances, and noise is added. When taking these sources into account to produce a robust output, the impulse becomes wider.

In Figure 8.9 the three different APs obtained using the CN electrode are shown together with the fully deconvolved signal where the highpass filtered (2 kHz) AP from Figure 8.7 is included as a comparison. The figure also shows the "ideal" result with the impulses corresponding to each AP.

As the figure shows, all deconvolved signals are similar to the ideal result despite the Wiener filter is not optimally tuned. Moreover, compared to the highpass filter and the partial deconvolution approaches, the width is further reduced while a monophasic output is produced. Except for AP #1, all impulses quickly return to zero which is favourable when measuring the jitter as well as calculating the number of fibers.

By comparing AP #1 in Figure 8.9 (c) and (d), we see that the highpass filter output is not as noisy as the fully deconvolved signal. If desired, the latter signal may be improved by retuning the Wiener filter using an assumed conduction velocity of, e.g., 3.3 m/s instead. This yield a fully deconvolved AP that is monophasic and estimates the main peak as good as the highpass filtered AP. If further improvements are desired, the assumed electrode distance may be retuned as well.

The price paid for this improvement is that the performance on the other APs becomes worse but this is still better in terms of cancellation and peak width than corresponding results of the highpass filter. This suggests that the deconvolution approach is potentially superior to the highpass filter approach. The characteristics that is important (optimal overall performance, or optimal performance for a certain type of APs) controls what optimality criterion to select. The Wiener filter design then produces the linear filter that fulfills this criterion the best.

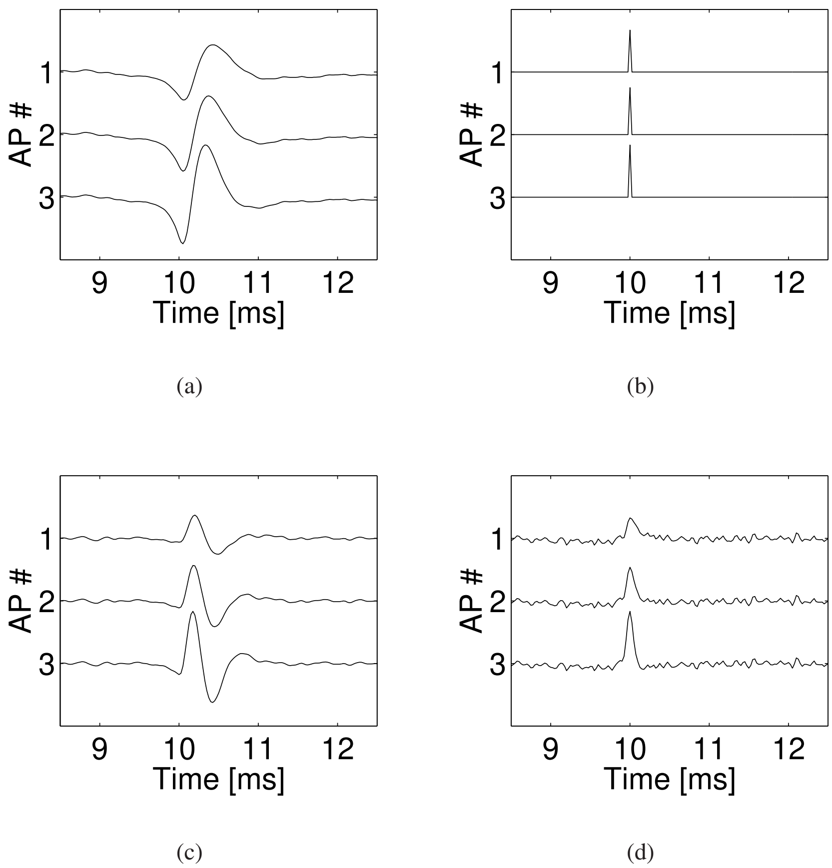


Figure 8.9: Sample results from a simulation of three APs with the conduction velocities c = 3.3, 3.7, and 4.1 m/s, respectively. In each diagram, we see (a) the unfiltered CNAP, (b) the impulses we want to estimate, (c) the highpass filtered CNAP using a second order Butterworth filter with a cut-off frequency of 2 kHz, and (d) the fully deconvolved signal $\hat{u}(n)$.

8.3.4 Deconvolution of a MUAP

Figure 8.10 shows the results of processing the MUAPs from four discharges of a simulated MU with twice as many fibers per square millimeter as normal. As shown in Figure 8.10 (a), the MUAP shape is varying noticeable due to the jitter in the constituent AP components and an experienced eye can see that at least three dominant peaks are present. From Figure 8.10 (c) we can see that, in fact, four peaks prevail in the MUAP.

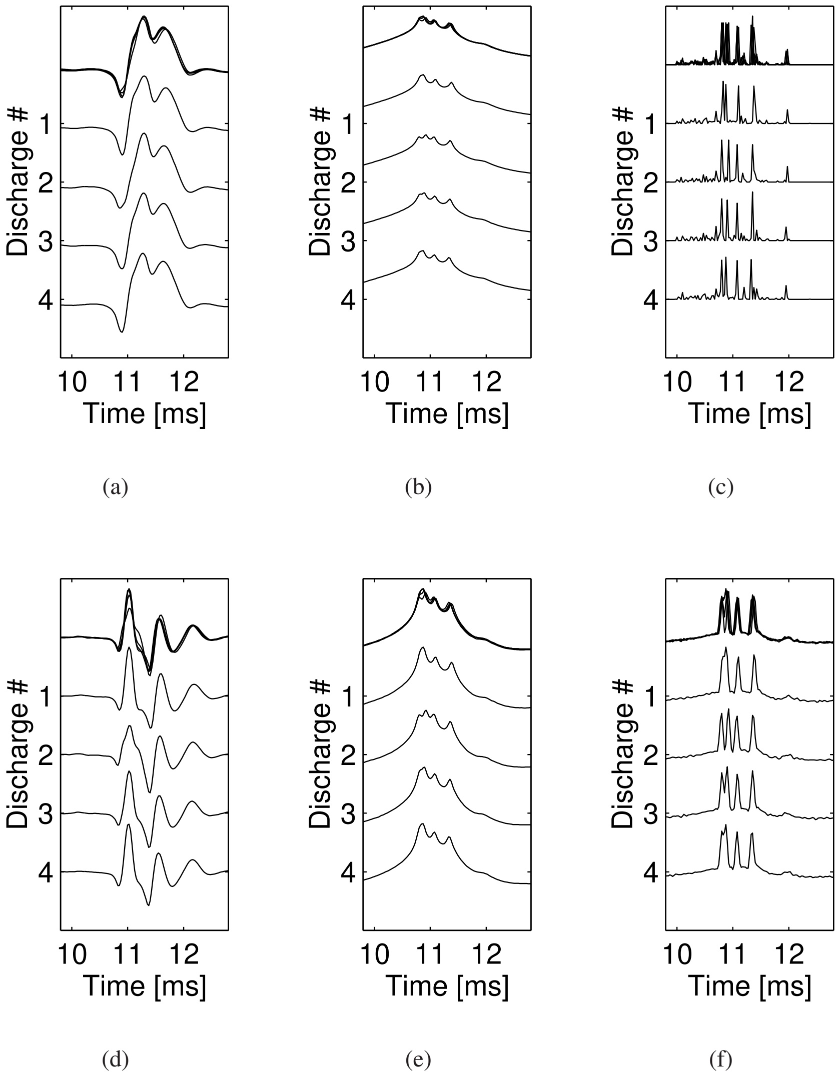


Figure 8.10: Sample results from a simulation of a MU with twice as many fibers as normal where the MUAPs of four discharges were obtained using a CN electrode. The four MUAPs are superimposed at the top of each diagram and shown individually below in subsequent traces. In each diagram, we see (a) the unfiltered MUAP, (b) the ideal weighting functions, (c) the ideal impulses, (d) the MUAP filtered using a second order highpass Butterworth filter with a cut-off frequency of 2 kHz, (e) the partially deconvolved signal $\hat{w}(n)$*, and (f) the fully deconvolved signal* $\hat{u}(n)$*.*

Using the highpass filtered MUAP, Figure 8.10 (d), only two fibers are clearly seen (the third peak is an artifact, see Section 8.3.1). Using the information in the jitter, an experienced eye can see that the first peak is at least a double (actually a triple) peak and that the second peak is indeed single.

In Figure 8.10 (e) and (f), however, all four peaks may be noticed where the latter diagram yields the most clear picture of the number of peaks and their whereabouts.

8.4 Estimating mean fiber concentration

The number of muscle fibers and their distribution within a MU is an important factor in diagnosis. This section explores the possibility of estimating the MFC using a MUAP obtained with a CN electrode. The MFC denotes the number of fibers present (on average) within the area 1 mm^2 and is measured in fibers/mm^2.

A similar, but different, measure is the *fiber density* (FD) parameter [86]. There are, however, two main drawbacks with the FD parameter: the requirement of a SF electrode and the manual fiber-counting procedure. We seek instead a fully automatic method that accurately obtains the corresponding information from a CN MUAP.

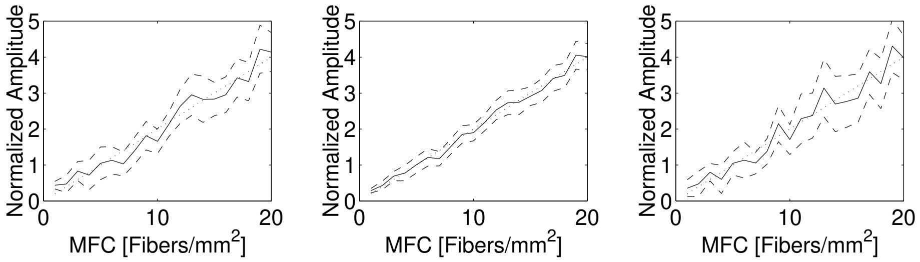
The feasibility of the CN MUAP (bandpass filtered 20 Hz-10 kHz), the partially deconvolved MUAP, and the fully deconvolved MUAP for estimating the MFC is explored below through Monte-Carlo simulations. As estimates of the MFC, the peak-to-peak amplitude, the absolute area, and the number of peaks are tested. We refer to these three entities as the *measurement variables*.

In each simulation, the mean and the variance of the measurement variable were calculated using 30 discharges obtained from 20 random MUs. The MUs were generated as described in Chapter 7.

The mean is expected to increase linearly with the MFC (up to a possible saturation point, see Figure 8.13) whereas the variance is expected to be directly proportional to the MFC. Both these expectations also corresponded well with the acquired data, see below.

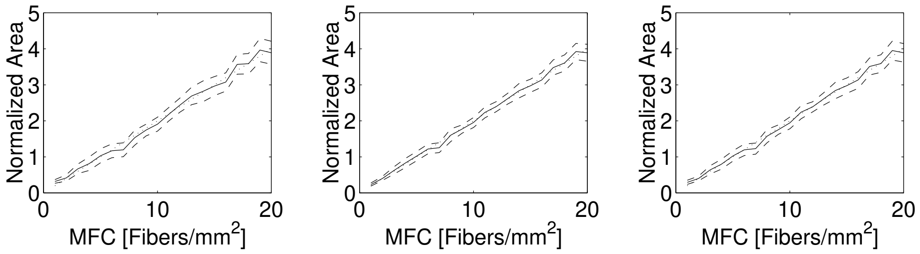
To be able to compare the performances of the three input signals listed above, the linear assumption was used to make a linear regression of the data. With the regression line, each measurement variable was normalized to yield a value of 1 at MFC = 5 fibers/mm^2 (corresponding to a normal healthy person).

Figure 8.11 shows the normalized mean *peak-to-peak amplitude* with the 95% confidence intervals included for each of the three input signals. As the figure shows, the assumption that the data increases linearly with the MFC seems to be reasonable.



(a) (b) (c)

Figure 8.11: The normalized mean peak-to-peak amplitude (solid) with the 95% confidence intervals included (dashed) for each of the three input signals (a) the CN MUAP $\phi(n)$ after bandpass filtering (20 Hz-10 kHz), (b) the partially deconvolved signal $\hat{w}(n)$, and (c) the fully deconvolved signal $\hat{u}(n)$. The normalized amplitude was set to 1 for MFC = 5 using the linear regression (dotted) of the data.

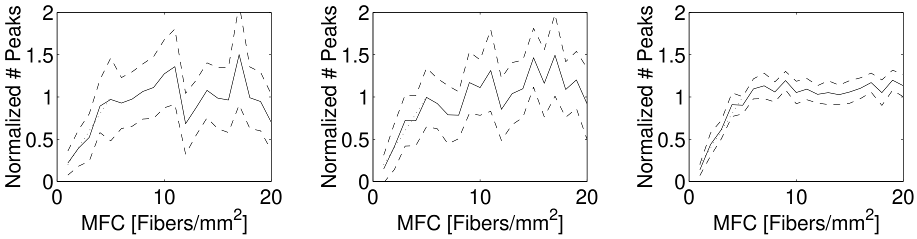


(a) (b) (c)

Figure 8.12: The normalized mean absolute area (solid) with the 95% confidence intervals included (dashed) for each of the three input signals (a) the CN MUAP $\phi(n)$ after bandpass filtering (20 Hz-10 kHz), (b) the partially deconvolved signal $\hat{w}(n)$, and (c) the fully deconvolved signal $\hat{u}(n)$. The normalized amplitude was set to 1 for MFC = 5 using the linear regression (dotted) of the data.

Owing to the normalization, the advantage of the partially deconvolved signal when estimating the MFC through the peak-to-peak amplitude is clearly visible. The slightly worse performance of the CN MUAP may be explained by cancellation whereas the equivalent performance of the fully deconvolved MUAP may be explained by its unsuitability for peak-to-peak amplitude measurements.

Figure 8.12 shows the normalized mean *absolute area* with the 95% confidence



(a) (b) (c)

Figure 8.13: The normalized mean number of peaks (solid) with the 95% confidence intervals included (dashed) for each of the three input signals (a) the CN MUAP $\phi(n)$ after bandpass filtering (20 Hz-10 kHz), (b) the partially deconvolved signal $\hat{w}(n)$, and (c) the fully deconvolved signal $\hat{u}(n)$. The normalized amplitude was set to 1 for MFC = 5 using the linear regression (dotted) of the linear portion (MFC $\leq$ 5) of the data.

intervals included for each of the three input signals. As the figure shows, the assumption that the data increases linearly with the MFC seems to be reasonable here as well.

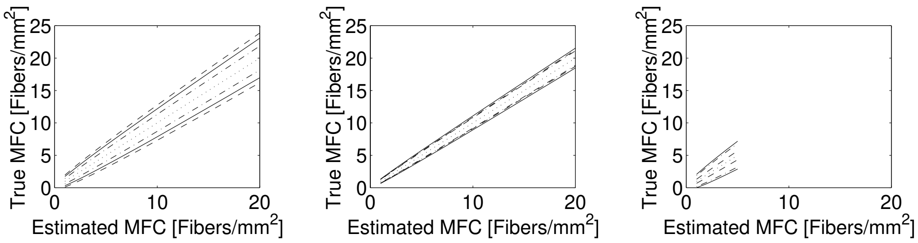
All input signals provided small confidence intervals which suggests that the area is a good parameter when estimating the MFC. There are no clear differences between the three, but the partially deconvolved signal seems to provide a slightly smaller variance than the other two.

In the final simulation, the number of peaks was assessed. In order to count only fiber peaks and not the noise peaks, a threshold was used and individually tuned for each input signal. Only the peaks above this threshold were counted.

Figure 8.13 shows the normalized mean *number of peaks* with the 95% confidence intervals included for each of the three input signals. As the figure shows, the measurements saturate at MFC $\approx$ 5 fibers/mm^2 but seems to increase linearly below that value. Here, only the fully deconvolved signal, Figure 8.13 (c), provides a small variance.

Figure 8.14 shows the correspondance between the estimated and the true MFC with the 95% confidence intervals for the three input signals and the three measurement variables. The results are calculated using the linear regressions of the mean and the variance. Hence, the performance of the peak-counting approach is only available for MFC $\leq$ 5.

With these diagrams, a quantitative comparison of the different approaches may be carried out. In Figure 8.14 (a), for example, the differences in Figure 8.11 are now clearly seen; the partially deconvolved signal provides the best MFC esti-



(a) (b) (c)

Figure 8.14: The correspondance between the estimated MFC and the true MFC using the linear regression (dotted) with the 95% confidence intervals included for the CN MUAP (solid), the partially deconvolved signal (dash-dotted), and the fully deconvolved signal (dashed). Each diagram show the results using (a) the normalized mean peak-to-peak amplitude, (b) the normalized mean absolute area, and (c) the normalized mean number of peaks. The confidence intervals are calculated using the linear regressions of the mean and the variance.

mate whereas the fully deconvolved signal provides the worst.

As Figure 8.14 (b) shows, the performance of the partially deconvolved signal is only slightly better than the other two when the area is used, whereas the performance of the fully deconvolved signal is clearly better than the other two when the number of peaks is measured as shown in Figure 8.14 (c).

In conclusion, the mean absolute area seems to be the best measurement variable to use. When it comes to selecting the input signal, the results suggests that any of the three may be selected.

8.5 Estimating jitter

By measuring the timing variability, the jitter, of individual APs, functional studies of the motor endplates may be performed. Unless axonal stimulation is used, the *absolute* jitter in individual muscle fibers can not be estimated. Instead, the *relative* jitter is measured by observing the variability of the *inter-potential intervals* (IPIs) in pairs of APs [86].

Often, the jitter is expressed using the *mean consecutive difference* (MCD) because of its robustness against outliers and trends [29]. It is defined as

$$\mathrm{MCD} \triangleq \frac{1}{J-1}\sum_{j=1}^{J-1}|\mathrm{IPI}_{j+1}-\mathrm{IPI}_j| \tag{8.25}$$

where IPI_j is the IPI of a pair of APs in discharge j out of a total of J discharges. For Gaussian jitter without trends, the expectation of the MCD is

$$E(\text{MCD}) = \frac{2}{\sqrt{\pi}}\sqrt{\sigma_1^2 + \sigma_2^2} \tag{8.26}$$

where σ_1^2 and σ_2^2 are the variances of the absolute jitter of the two paired APs.

To be able to perform these calculations, the APs must first be detected and localized in order to obtain their (relative) latency. Then, the detections need to be classified, or discriminated, in order to assure that the IPIs are calculated on the same pair of muscle fibers for all discharges.

Once all of this is done, the actual jitter calculations are straight-forward. This section is therefore focused on the localization and discrimination of the APs of which the former is presented next.

8.5.1 Peak localization

In this section the performance of the peak localization algorithm is analyzed. To obtain good jitter estimates, it is important that the localization of the APs is accurate. The algorithm used here, see below, is the same as currently used in clinical routine. The algorithm is tested on three types of signals: a bandpass filtered (500 Hz - 10 kHz) SF compound AP, a highpass filtered (2 kHz) CN MUAP, and a fully deconvolved CN MUAP.

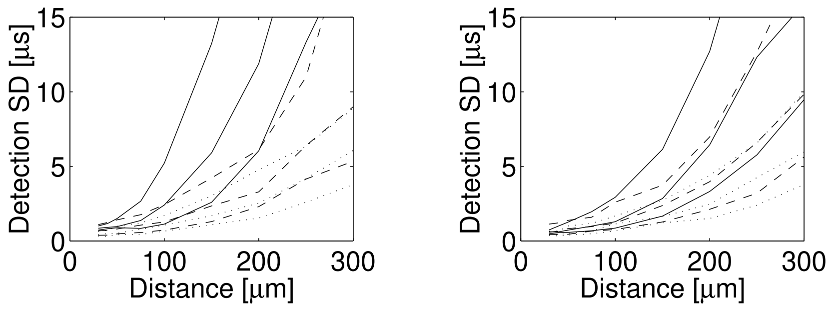
The requirements are high; in a normal healthy person the absolute jitter has a standard deviation of about 15 μs with lower values below 5 μs [86]. Hence, the standard deviation of the localization error must be significantly lower in order to estimate the jitter accurately. With a sampling period of 25 μs ($f_s = 40$ kHz), the sampled data must be further processed in order to achieve the required accuracy.

The algorithm used here localizes the APs in two steps: First, a coarse estimation is performed by finding peaks in the sampled data. Then, these estimates are refined by fitting a second degree polynomial to three samples of the peak and use the (mathematical) point in time where the polynomial has its maximum as the final estimate. With this algorithm, it is possible to achieve a standard deviation of the localization error below 2 μs if the input signal quality is good enough.

There are mainly two sources of error: the noise that disturbs the localization refinement and the presence of nearby APs that displace the observed peak from its true position. Below, these two sources of error will be analyzed in more detail.

The noise-induced localization error

Here, the the first part of the localization error is explored in further detail. In Figure 8.15 the standard deviation of the noise-induced localization error is shown



(a) (b)

Figure 8.15: The standard deviation of the peak localization error when the algorithm is applied to the SF compound AP (dotted), the highpass-filtered CN MUAP (dashed), and the fully deconvolved signal (solid). For each radial distance, three sets of 100 APs were simulated with conduction velocity c = 3.3, 3.7, and 4.1 m/s, respectively. The noise power density used was $\eta = 0.001$ mV^2/kHz. In each diagram, the SF and highpass-filter performances were estimated under identical conditions, whereas the performance of the fully deconvolved signal was estimated (a) directly, and (b) after a bandwidth reduction to 50% (± 10 kHz).

as a function of the radial distance for three fibers with conduction velocities c = 3.3, 3.7, and 4.1 m/s, respectively. As expected, the error increases as the radial distance increases but, as Figure 8.15 (a) shows, the error for the fully deconvolved signal is significantly higher than the other two signals.

This behavior arises, in combination, because the localization-refinement algorithm is noise sensitive and because this was not regarded in the tuning of the Wiener filters. One way to alleviate this is to limit the bandwidth as to prevent the high-frequency noise from disturbing the estimation. The drawback is that the resolution capability decreases correspondingly.

Figure 8.15 (b) shows the localization error when the bandwidth was reduced to 50% of the original. Hence, the bandwidth is ± 10 kHz and, as the figure shows, the performance is improved because the noise above 10 kHz is discarded. The signal energy above this frequency is also discarded, of course, which is not desirable. By properly addressing these aspects during the tuning of the Wiener filter, the performance may be improved further.

The interference-induced localization error

Here, the second part of the localization error is explored by analyzing the peak displacement due to the presence of another AP and its impact on the actual MCD

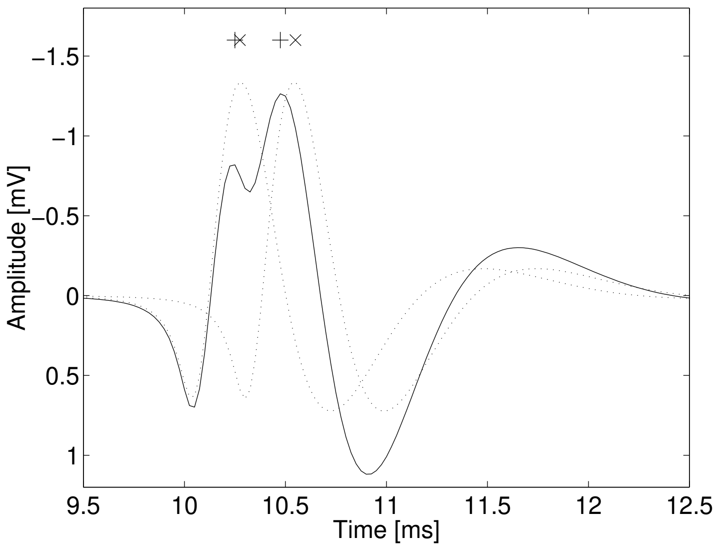


Figure 8.16: En example of the peak displacement taking place in the compound AP (solid) when the two constituent APs (dotted) are closely spaced in time. The peaks in the compound AP (+) are displaced to the left (negative displacement) compared to the peaks in the individual APs (×). (The APs were obtained with the SF electrode.)

measurement, see Figure 8.16.

The same setup as above was used with the difference that a completely noise-free signal was used. Note, however, that the Wiener filters were still tuned using the same noise level as above. An additional AP with the same properties as the AP under study were added where the time separation of the two APs ranged from -800 μs to +800 μs. A negative time difference means that the other peak arrives *before* the peak under study. The displacement of the peak due to the presence of the other AP was then estimated by comparing the peak position with and without the presence of the other peak. The bandwidth reduction to 50% and the same refinement algorithm as described above were used in all estimations.

The peak displacements found for each of the three approaches are shown in Figure 8.17-8.19. A positive displacement means that the peak is delayed compared to its undisturbed position. For small time differencies only one peak was observable which rendered the analysis impossible.

Figure 8.17 shows the displacement taking place in the SF compound AP. The region where only one peak is discernable ranges from approximately ± 200 μs to ± 400 μs depending on the radial distance.

Figure 8.18 shows the displacement taking place in the fully deconvolved MUAP.

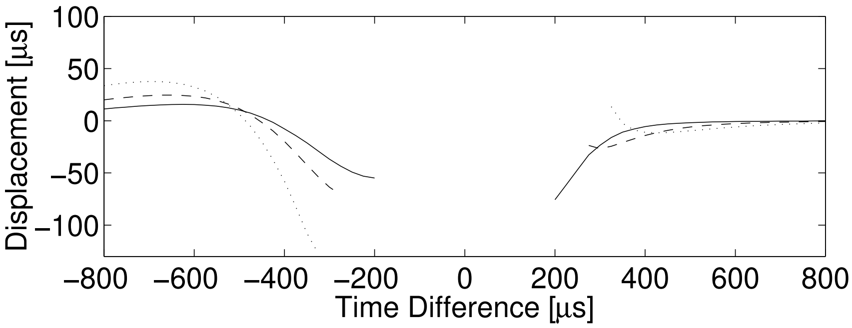


Figure 8.17: Displacement of an AP obtained with the SF electrode when a neighboring AP is present. The displacement is shown as a function of the time separation for the radial distances $r = 30, 150$, and 300 μs as indicated by the (solid), (dashed), and (dotted) lines, respectively. All parameters were identical for the two APs.

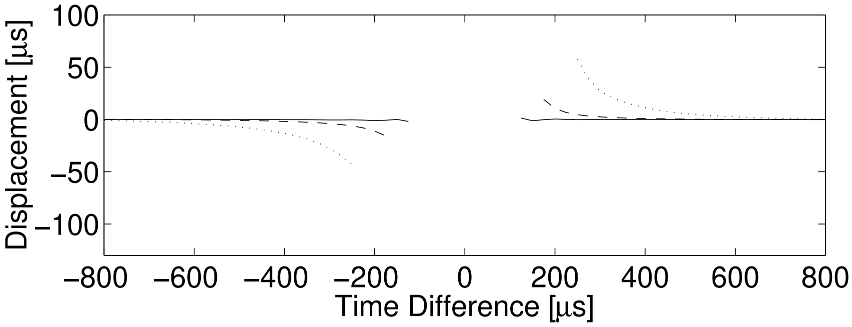


Figure 8.18: Displacement of an AP obtained with the CN electrode when a neighboring AP is present and the full deconvolution is applied. The displacement is shown as a function of the time separation for the radial distances $r = 30, 150$, and 300 μs as indicated by the (solid), (dashed), and (dotted) lines, respectively. All parameters were identical for the two APs.

The region where only one peak is discernable ranges from approximately ± 120 μs to ± 250 μs depending on the radial distance.

This is an improvement compared to the performance in Figure 8.17. Both the one-peak region and the magnitude of the displacement are smaller. For small radial distances the displacement is even approximately zero.

Figure 8.19 shows the displacement taking place in the highpass filtered MUAP. The region where only one peak is discernable ranges from approximately ± 150 μs to ± 400 μs depending on the radial distance.

This result is somewhere in between that of Figure 8.17 and Figure 8.18 except

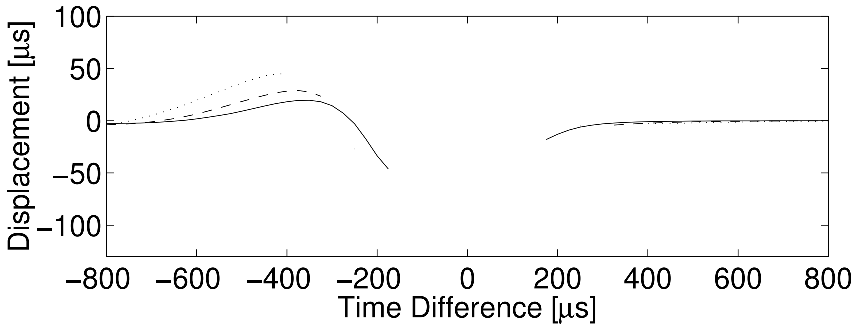


Figure 8.19: Displacement of an AP obtained with the CN electrode when a neighboring AP is present and the highpass filter is applied. The displacement is shown as a function of the time separation for the radial distances $r = 30, 150$, and 300 μs as indicated by the (solid), (dashed), and (dotted) lines, respectively. All parameters were identical for the two APs.

for positive time differencies where the displacement is practically zero for many APs in this case.

With these results, we can anticipate that the jitter estimation will be disturbed by neighboring APs. In Figure 8.20 the results of measuring the MCD of two APs for different time separations are shown. With the absolute jitter 15 μs, the exptected MCD is about 24 μs as indicated in the figure.

The advantage of the narrow impulses in the fully deconvolved signal is clearly shown. Already from 300 μs, the fully deconvolved CN MUAP yields stable estimates as opposed to 1000 μs for the highpass filtered CN MUAP and (possibly) 2000 μs for the SF compound AP.

By comparing, respectively, Figure 8.20 (a)-(c) with Figure 8.20 (d)-(f), it is clear that the jitter measured in the SF signal is biased albeit with a low variance. In contrast, the other algorithms yields less biased estimates but with somewhat higher variance.

In conclusion, the results suggest that the fully deconvolved MUAP may successfully be used for jitter estimation. If the variance of the localization error with this input may be reduced further to a level comparable with that of the SF input, this approach may well outperform the other two algorithms.

8.5.2 Peak classification

To discriminate between APs originating from different muscle fibers, we propose classification through the use of a MHT/Kalman tracker, c. f. Chapter 4 and Chapter 3, because it is based on a Bayesian, statistical approach. As such, it is able to

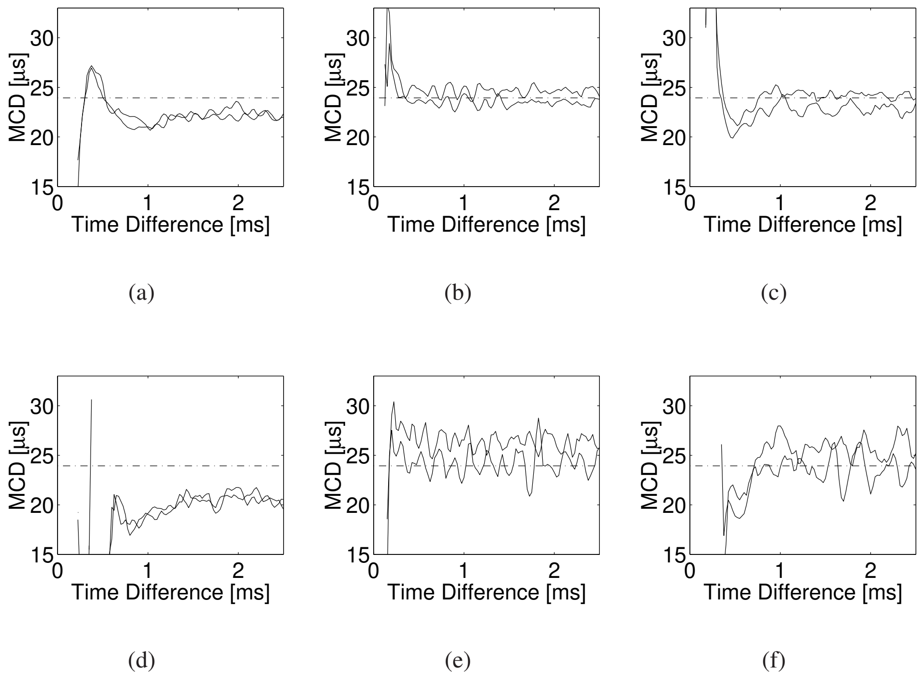


Figure 8.20: The diagrams show two simulation runs where the jitter MCD is estimated for two fibers at varying mean inter-peak intervals. Each diagram represents (a) SF potential, (b) fully deconvolved signal (50% bandwidth reduction), and (c) highpass filter CN potential. (d)-(f) Analogous with (a)-(c), but the amplitude for the second AP was reduced to 50%.

provide the maximum a posteriori classification of the available APs under certain conditions, see Section 4.4. The most important aspect is the statistical model because it is used by the MHT algorithm to differentiate between a *good* classification and a *bad* one.

The required Kalman filter is simple in this case; it consists of two integrating subsystems driven by white noise. The subsystems model the AP latency and the amplitude, respectively.

We anticipate this algorithm to provide good AP classifications if properly tuned. As a mere "proof of concept", a test run with no special tuning is shown in Figure 8.21 where the individual APs in 20 discharges of a fully deconvolved MUAP are detected and classified. The detector was a simple peak detector that reported all peaks above a certain threshold with the same resolution as the sampling period (25 μs). (No polynomial was fitted to the peaks in order to refine the localization.)

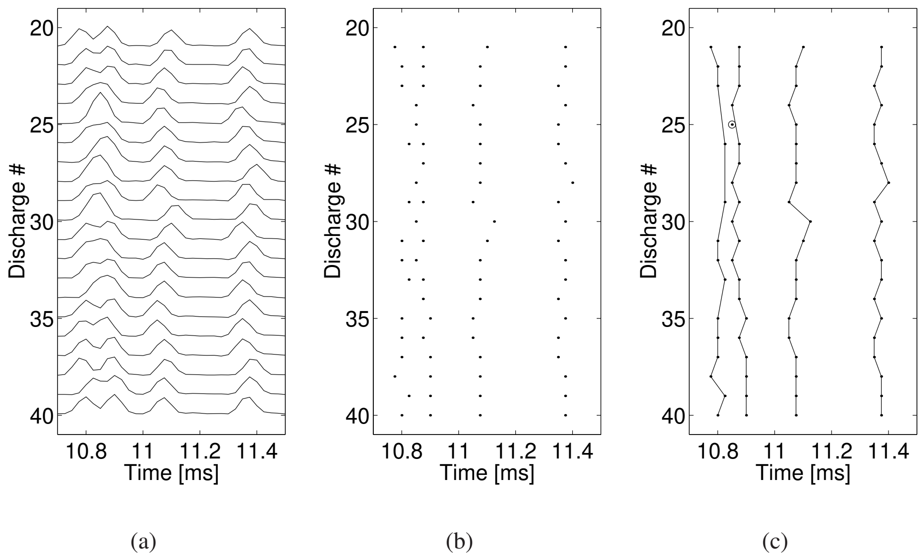


Figure 8.21: A sample result of using the MHT/Kalman algorithm to discriminate and classify the detected APs. The diagrams show (a) the fully deconvolved signal, (b) the detected peaks, and (c) the found tracks. Detections not assigned to any track are marked (∘).

As the figure shows, the tracking algorithm performs well in this example despite no particular tuning was carried out. This suggests that a performance "reserve" is available that may be used in more complex scenarios with false spikes and other disturbances. Note that one observation in discharge #25 (marked in the figure) was not assigned to any track. That is actually perfectly acceptable because it is clearly a double peak (wider and higher) where the two individual peaks could not be resolved, see Figure 8.21 (a). Optimally, all such peaks should be discarded and not assigned to any track because the succeeding jitter estimate is impaired by such spikes.

8.6 Discussion

The possibility of using Wiener filters to improve the diagnostical value of the MUAP obtained with a CN electrode was explored. To design and tune the Wiener filter a noise model and a design model are required. Depending on the choice of criterion, the design model become different.

By assuming that the design model were to be found among the set of AP models, a heuristic tuning approach was used where a set of Wiener filters were tested on a set of APs and the filter with the best overall performance was selected. It was considered important that the output was monophasic and that the peaks were as narrow as possible without artifacts being generated.

Two Wiener filters were designed which allowed two levels of deconvolution, namely: partial deconvolution and full deconvolution. Both were shown to produce monophasic signals when applied to MUAPs. This was different from both the original MUAP as well as a highpass filtered MUAP. The latter was included as a comparison because it has been suggested [87] [32] that by applying a highpass filter to the CN signal it may provide the same type of information as if a SF signal was used.

Certainly, the width of the APs become less wide by applying the highpass filter, but the filtration produces additional phases as artifacts that may erroneously be interpreted as a resolved peak. Nevertheless, the extra phases introduced disturb the analysis of nearby APs. A monophasic signal, in contrast, has a minimal impact on nearby APs. This was also shown where a set of four discharged MUAPs with four dominating peaks were targeted with the highpass filter, the partial deconvolution filter, and the full deconvolution filter, respectively. The highpass filter resolved two peaks only and produced a noticeable artifact in addition. By using the jitter information, an experienced eye could see that the first peak was a multiple. The deconvolved signals, on the other hand, clearly resolved all four peaks in most of the discharges.

In Section 8.4 the possibility of estimating the mean fiber concentration (MFC) using a CN MUAP was explored. The MUAP was processed in three ways: bandpass filtered (20 Hz-10 kHz) as in today's clinical routine, partially deconvolved using the $G_i(\zeta)$ Wiener filter, and fully deconvolved using the $G_{iw}(\zeta)$ Wiener filter. Three MFC estimates were analyzed: the peak-to-peak amplitude, the absolute area, and the number of peaks.

All in all, the area provided the best performance of the three estimates with comparable results for all three MUAP variants. This suggests that the area should be used when estimating the MFC and that the MUAP preprocessing is not so important. Further studies should be carried out, however, because pathological MUs may yield different results. Both myogenic and neurogenic MUs, for example, have an inhomogeneous distribution of their muscle fibers and this may affect the MFC estimates differently than the homogeneous MUs analyzed in this chapter.

In Section 8.5 the characteristics of three jitter estimation approaches were analyzed. The selected approaches differed in which input signal that was used and were: a bandpass filtered (500 Hz-10 kHz) SF compound AP as used in today's clinical routine; the fully deconvolved CN MUAP as presented in this chapter; and a highpass filtered (2 kHz) CN MUAP as suggested in [32].

The SF signal was shown to be clearly more accurate than the other two, especially the fully deconvolved signal, in localizing the peaks. By a 50% bandwidth reduction of the fully deconvolved signal the performance improved. By properly tuning the Wiener filter, it is possible to improve this even further; at least to a level comparable with the performance of the highpass filtered MUAP because both filters are linear.

The peak displacement caused by a neighboring AP showed a different picture. In this case, the roles were changed and the SF signal performed worst and the fully deconvolved signal performed best. Here, the benefit of the monophasic and narrow shape provided by the fully deconvolution filter becomes clear; the APs may be more closely spaced without causing interference.

By calculating the MCD for two APs at different spacings in time, it was shown that the peak displacement is just as important a factor when estimating the jitter. To obtain good jitter estimates through the MCD, the individual APs, when using the SF signal or the HP filtered CN signal, must be separated at least 1 ms (possibly more with the SF signal) compared to 300 μs for the fully deconvolved signal.

For MUAPs with more than two detected APs, the detection needs to be classified so that the IPIs are calculated on the same pair of muscle fibers for all discharges.

We propose the use of the MHT/Kalman tracking method because it provides the optimal maximum-a-posteriori classification if the used statistical model is correct and enough data storage is available. Even if these requirements are not completely satisfied, the performance is good.

A proof of concept was presented where a MUAP with four detected APs was processed. Without any special tuning, the tracking result was good and the APs was successfully tracked.

With a spacing of 100 μs, the APs sometimes fused and formed a single peak. Preferrably, all such peaks should not be assigned to any track. One way to do this is to discard *all* observations in a certain region where there are fewer observations than expected. For example, in Figure 8.21, there are several traces with two observations at latencies about 10.8 ms. Hence, it is likely that a single observation in this region is due to an unsuccessful resolution. Hence, no observation with a latency of about 10.8 ms should be assigned to a track unless there are two observations available. The worst case scenario is that there are occasions where one AP really is absent (blocking, for example) and the algorithm discards correct observations. This yields only a correspondingly higher variance in the jitter estimate (of closely spaced APs) but seems to be a reasonable price to pay.

The conclusions from this analysis is that the proposed prefiltering may improve the usability of the MUAPs in assessing the MFC. Moreover, the results suggest that the jitter may be estimated from the concentric needle EMG with an accuracy comparable with the traditional single fiber EMG method. To test the validity and implications of these conclusions, further studies are needed.

Appendix 8.A Finding the optimal Wiener filters

Here we present a straight-forward procedure to find appropriate Wiener filters by simply testing a range of different Wiener filters on a range of different APs. For the applications considered in this chapter, the *overall* performance of the Wiener filters is important.

The full deconvolution must provide a monophasic signal with as narrow as possible a spike. No significant artifacts are allowed in the signal because this would spoil the deconvolution of the MUAPs. Also, the range of tested APs must reflect the range of the APs in the intended MUAPs.

The weighting function used in the tuning, $W'(\zeta)$, is a modified version of $W(\zeta)$ from Chapter 6. There are two modifications introduced. First, the fiber length is assumed to be infinite because the actual value is unknown and may also differ among the fibers. Second, the weighting function is non-causal with its peak at $t = 0$ around which it is symmetrical. In principle, it is equivalent with an infinite muscle fiber that was triggered in the infinite past (t = -∞) and where the transmembrane current arrives at the electrode at $t = 0$. For obvious reasons, this infinite weighting function is truncated and transformed into the frequency domain to form $W'(\zeta)$.

By testing different realization of both the APs and the Wiener filters, we have concluded that the design models $\bar{I}_o(\zeta)$ and $\bar{W}_o(\zeta)$ defined by the parameter settings

$$c_o = 3.9 \quad [\text{m/s}] \tag{8.27}$$
$$r_o = 30 \quad [\mu\text{m}] \tag{8.28}$$

where c_o is the conduction velocity and r_o is the radial distance, provide the Wiener filters with the best overall performance in terms of number of phases and amount of artifacts.

Below, we motivate this selection with two figures where, respectively, the conduction velocity is varied while keeping the radial distance constant, and vice versa.

The result when varying the conduction velocity is shown in Figure 8.22 where the full deconvolution filter $G_{iw}(\zeta)$ is applied to an AP $\phi(n)$ obtained with the CN electrode. In each column the same AP was used as input to the different Wiener filters while in each row the same Wiener filter was applied to the different APs. Hence, to find the Wiener filter that yields the best overall performance, the row with the best overall performance should be selected and the corresponding conduction velocity c_o identified.

From the figure, it can be seen that the second row from the top only contain

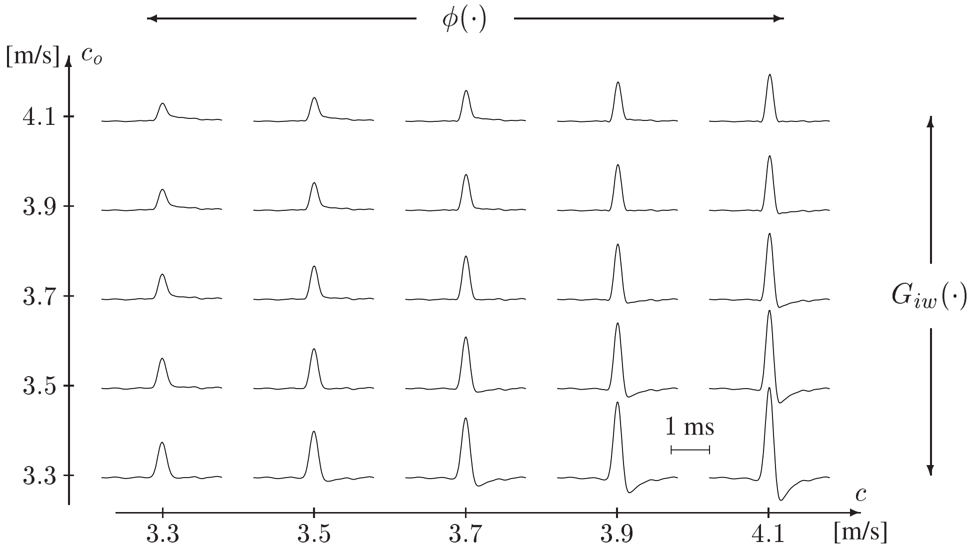


Figure 8.22: The result when applying the full deconvolution filter $G_{iw}(\cdot)$ on a single AP $\phi(\cdot)$ while varying both the conduction velocity c for the synthesis as well as the conduction velocity c_o for the deconvolution. Hence, the same AP was used in each column while the Wiener filter was varied, whereas the same Wiener filter was used in each row while the AP was varied. For example, the deconvolution result found in the second column from the left and the third row from the bottom was synthesized *using the conduction velocity $c = 3.5$ m/s while it was* deconvolved *using the conduction velocity $c = 3.7$ m/s. The radial distance was equal for both synthesis and deconvolution and selected to $r = r_o = 100$ μm.*

monophasic signals that are reasonably narrow. Thus, selecting $c_o = 3.9$ m/s seems reasonable.

Analogously, Figure 8.23 shows the same principle but for varying radial distance instead. In the figure, the first row from the bottom only contain monophasic signals. Thus, selecting $r_o = 30$ μm, as stated above, seems reasonable.

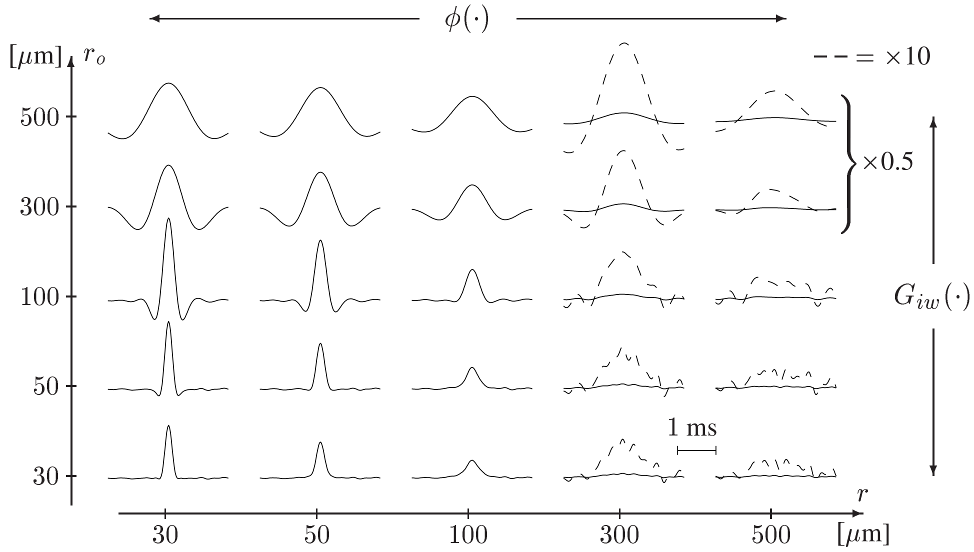


Figure 8.23: The result when applying the full deconvolution filter $G_{iw}(\cdot)$ on a single AP $\phi(\cdot)$ while varying both the radial distance r for the synthesis as well as the radial distance r_o for the deconvolution. Hence, the same AP was used in each column while the Wiener filter was varied, whereas the same Wiener filter was used in each row while the AP was varied. For example, the deconvolution result found in the third column from the left and the first row from the bottom was synthesized using the radial distance $r = 100$ μm while it was deconvolved using the radial distance $r_o = 30$ μm. The conduction velocity was equal for both synthesis and deconvolution and selected to $c = c_o = 3.9$ m/s. Note that the two topmost rows are reduced in magnitude with a factor of two. Moreover, the results in the two rightmost columns are also shown magnified (dashed) with a factor of ten (five for the two topmost/rightmost results).

CHAPTER 9

Detecting and discriminating C-fiber action potentials

IMPROVING signal processing in an application where traditional tools are inadequate often generates many new challenges. We will here describe such an application, arising from the need to examine the stimulus-response characteristics of peripheral unmyelinated (C) fibers in human skin nerves. Key tools for the solution were to be found in a seemingly unrelated area, namely radar tracking of multiple targets.

The action potentials (APs) of the C-axons are recorded through a thin needle electrode inserted transcutaneously into the peroneal nerve of an awake human subject [94] [43]. The APs may be detected as extracellular spikes in this recording. Yet, the signal-to-noise ratio (SNR) is rather poor and the amplitude of some APs is of the same order of magnitude as the peaks of the noise [35]. In such situations special methods are required for the detection.

Neuronal activity is evoked by applying sensory stimuli in the skin area innervated by the fiber of interest. One problem, however, is that APs originating from other fibers are also recorded by the electrode.[1] Their presence obstructs the examination of the stimulus-response characteristics as it is virtually impossible to decide which APs originate from the fiber under study.

To overcome this problem, Hallin and Torebjörk introduced a method that shows the excitation of a C-fiber by utilizing the so-called *marking phenomenon* [95]. The phenomenon stems from the slight decrease of a fiber's conduction velocity after an AP has been conducted. The conduction velocity then slowly returns to its initial value.

The principle of the method is to apply a distinct stimulus repetitively, at a low

[1] These fibers could be either co-excited by the applied stimuli or spontaneously active.

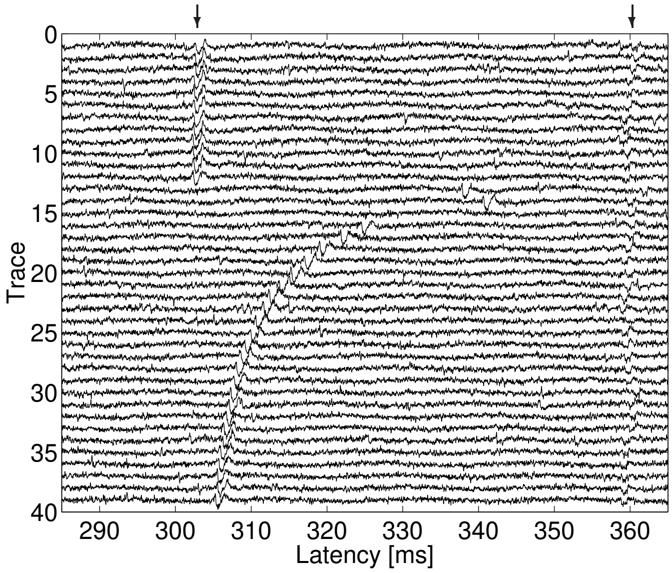


Figure 9.1: The APs of two C-fibers with latencies of about 303 ms and 360 ms, respectively, are shown. The responses were excited by electrical stimuli delivered at 0.25 Hz into the skin innervation territory of the fibers. Successive responses are displayed in traces from top to bottom. At trace 13, the left unit was activated in response to a mechanical stimulus causing a decreased conduction velocity, i.e., increased latency. Following this, the conduction velocity recovered gradually as indicated by the APs returning to the latency prior to the activation. The right unit did not respond to the mechanical stimulus and, hence, its latency was retained throughout the recording.

frequency (0.25 Hz), into the innervation territory of the C-fiber under study. For each impulse, one single AP is evoked and appears in the recording after a certain latency (Figure 9.1, at 303 ms). To document the response characteristics of the C-fiber, a physiological test stimulus (e.g., mechanical, temperature, or chemical) is applied into the receptive field of the fiber. If such a stimulus generates additional APs, the conduction velocity of the affected fiber decreases. Hence, the AP excited by the repetitive stimuli shows a noticeable increase in latency (Figure 9.1, trace 13 to 40). This change in latency is used as a *marker* to indicate that the C-unit responded to the applied physiological stimulus [74]. In addition, the latency increase provides information about the number of APs that were generated by the test stimulus [73].

To enhance the efficiency of these experiments, a computer-supported recording system is used [34] that both emits the repetitive stimuli and records the responses. For the repetitive stimuli, electrical impulses are used because they are both distinct and excite all different kinds of nerve fibers. The impulses are delivered through needle electrodes positioned in the innervation area of the fiber of interest. Often, several fibers are co-activated and recorded simultaneously, but due to differences in conduction velocity of the individual C-fibers, the APs are spaced in time in the recorded signal (Figure 9.1, at 303 ms and 360 ms). Using the marking phenomenon, it is thus possible to identify separate C-fibers and to examine their characteristic latency responses.

In the human skin nerves, different types of C-fibers exist [75].[2] Recently, it has become evident that the latency increase, due to a particular number of additionally triggered APs, and the time course of the recovery differ in different classes of C-fibers [97]. This finding is intriguing because it may promote new insights into differential properties of membranes in different C-fiber classes in humans.

Previously, the analysis of the recorded traces was carried out manually; a very time consuming task. Therefore, a computer program that detects the APs, discriminates between APs originating from different C-fibers, and estimates latency shifts and recovery constants quantitatively has been developed [57] [44]. The application is based on the signal processing algorithms presented in Chapter 2, 3, and 4.

9.1 Algorithm overview

To study the characteristics of the latency time course, two major problems need to be solved: the detection of APs in noisy recordings and the discrimination of APs originating from different C-fibers.

Practical applicability was the most important issue considered during the development of the signal processing algorithms. Hence, *optimality* was not strived for per se as it often means poor performance if key assumptions become invalid. Instead, the goal was to create an analysis tool that is easy to operate and quickly yields results comparable with a manual analysis. By meeting this goal, valuable time would be saved for the benefit of the evaluation work.

The signal processing approach we decided upon analyzes the C-fiber recordings in three steps (see Figure 9.2 and Figure 9.3):

1. *Detection* – Signal detection in noise is a problem with well-known solutions. With the considerations mentioned above in mind, the matched filter

[2] There are several classes of afferents carrying sensory information to the central nervous system and there are also several classes of sympathetic efferents regulating for instance sweating, piloerection, and vasoconstriction in the skin.

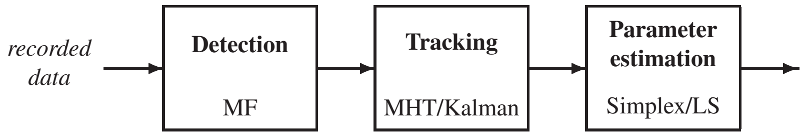


Figure 9.2: A block diagram of the algorithm. The APs in the recorded data are first detected using a MF. Following this, the APs originating from the same sensory unit are grouped together by means of target tracking using MHT with a Kalman filter predictor. In the last step, the found tracks are used to estimate some important parameters using a combined Simplex and LS algorithm.

(MF) was the detector of choice because it is both simple and robust, c. f. Chapter 2.

2. *Tracking* – The discrimination step is crucial for the success of this work and it presented an interesting challenge. We found that a reliable algorithm can be derived by exploiting the marking phenomenon and tracking the APs of a particular C-fiber in the responses to the repetitive electrical stimulus; hence, regarding the discrimination problem as a target tracking problem. Due to the vital importance of this step, the multiple hypothesis tracking (MHT) method [13] combined with a Kalman filter [46] predictor was selected because it is commonly considered as the best tracking algorithm, c. f. Chapter 4 and Chapter 3.

3. *Parameter estimation* – Once the time course of the latency corresponding to a particular C-fiber unit is isolated, a parametric model may be fitted to the data. The current application adjusts an exponential function to the measured latencies using a combination of the Nelder-Mead simplex algorithm [55] and the least squares (LS) method.

This chapter is organized as a description of the three main steps of the algorithm. Section 9.2 describes how the detection is done using a MF, c. f. Chapter 2. Section 9.3 deals with the tracking of the APs using the MHT method, c. f. Chapter 4, and Kalman filtering, c. f. Chapter 3, and describes how the AP amplitude is incorporated into the tracking algorithm. Section 9.4 describes the latency model and the parameter estimation. Section 9.5 presents the characteristics and performance of the algorithms through simulations and theoretical analysis. Finally, Section 9.6 illustrates the performance of the application on actual recordings obtained from awake human subjects.

(a) (b) (c)

Figure 9.3: Sample results from the three steps of the algorithm: (a) the detected APs, (b) the resulting five tracks after applying the tracking algorithm, and (c) the final trajectories obtained by fitting an exponential function to the tracks.

9.2 Detection

We seek a simple and robust detector with good, not necessarily optimal, performance that helps us reaching the goal of simplifying the analysis of the C-fiber recordings. With this in mind, the most important aspect is to design a *working*, *fast*, and *user friendly* application. This means that we cannot take all real life details into account. Instead, based on numerous test runs and a profound knowledge about this type of signals, we have primarily accounted for factors with a large improvement-cost ratio with respect to the *overall* performance. For example, factors that have a moderate improvement on the overall result, but requires extensive tuning or *exact* knowledge of some parameter have been discarded for obvious reasons.

With these trade-offs in mind, we have decided to make the following four key assumptions:

Assumption 9.1 *Constant AP shape – All APs have the same shape and only the amplitude of the APs differ among the units.*

Assumption 9.2 *White noise – The noise is uncorrelated.*

Assumption 9.3 *Semi-stationary noise – The noise variance may change from trace to trace, but is constant within each individual trace.*

Assumption 9.4 *"Sparse" recordings – The energy of any APs in each trace is negligible compared to the noise energy.*

Obviously, there is a huge advantage of making these assumptions in terms of practical and implementational simplicity as well as computational requirements.

With these assumptions a static MF detector may be used that does not require the AP shape and the color of the noise to be estimated from the data to be analyzed. With Assumption 9.3-9.4, the noise variance is still allowed to change during the experiment while simplifying the design of the detector. With these two assumptions, *invariance* to the unknown noise variance, see [72], is simply accomplished by using an estimate of the noise variance that is estimated from the entire trace where the contribution of the APs is neglected.[3]

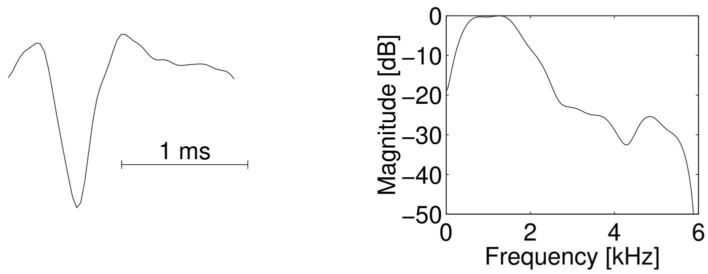
The (theoretical) drawbacks of using these assumptions weigh light in this perspective because we know from experience that these assumptions correspond quite well to the real situation and that the influence of deviations from these assumptins on the overall performance is small. Differencies in AP shape, for example, are manifested only as a reduced SNR in the MF output which is easily accounted for by lowering the detection threshold. The same goes for colored noise, but has the additional effect of increasing the MF output variance. The latter is easily accounted for by properly tuning the succeeding tracking algorithm. Assumption 9.4, finally, because any noticeable AP energy is approximately constant over the traces, the effect in such a case is an over-estimated noise variance that reduces the MF output and is easily accounted for as above. Note, however, that the SNR of the MF output, assuming that the APs and the noise are independent, is *not* changed in this case since the noise is scaled by the same amount.

9.2.1 Tuning

In order to derive, or tune, the AMFD we need to know the sought signal and the color of the noise in advance. The latter is elegantly settled under Assumption 9.2 whereas the former is met by noting that under Assumption 9.1 any already recorded AP may be used as a signal template.

When such a template was extracted from earlier recordings, several manually detected APs were aligned and averaged to reduce the influence of the noise and the background activity. The averaged potential was then lowpass filtered to further reduce the disturbances in the frequency region where no AP energy is expected. Figure 9.4 shows the resulting signal template and the corresponding Bode plot.

[3] In the general case, the detector criterion needs to be invariant to scaling of the measurements and rotation of the measurements in the subspace orthogonal to the signal subspace. In the case considered here, however, the number of samples used for the estimate is so large and the AP contribution is so small that the estimated noise variance may be treated as a "known" parameter in this regard.



(a) (b)

Figure 9.4: (a) The template of the C-fiber action potential to be detected, i.e. the time-reversed impulse response of the MF, and (b) the corresponding Bode magnitude plot (right).

9.2.2 Noise variance estimation

Although the color of the noise is assumed known (white in this case), its variance is unknown and need to be estimated from the recorded data. The recordings may, however, also contain hum from the power supply and surrounding equipment that yield a bias if not accounted for. By first removing the hum, using a notch filter, a simple maximum likelihood (ML) variance estimator may be used.

Under Assumption 9.2-9.4 and assuming that the notch is infinitely narrow, the ML estimate of the noise variance $\hat{\sigma}^2$ may be calculated using the notched data samples $w_n(t)$ in one trace of length N as

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{t=0}^{N-1} w_n^2(t) \quad . \tag{9.1}$$

This is still a biased estimate,[4] but the bias decreases as the width of the notch decreases. As we use a narrow notch filter at 50 Hz, the influence of the bias may be neglected and is not considered further.

Using the ML estimate (9.1) of the noise variance, the impulse response and the SNR of the matched filter are given by (2.12) and (2.13) with σ exchanged by $\hat{\sigma}$.[5]

[4]Noise energy is of course removed by the notch filter and some hum energy is still present.

[5]Introducing the ML estimate of the variance formally implies that a Kelly test [89] should be used instead of the generalized likelihood ratio test (GLRT) used here. Moreover, the MF output then has a t-distribution instead of a normal distribution. However, the number of data points used in the estimations is so large that these effects have a negligible effect on the result.

9.2.3 Action potential detection

During analysis, the MF output of each trace is normalized by its noise variance to yield a measure of the square root of the instantaneous SNR.[6] For stationary noise, this measure is directly proportional to the amplitude of the AP. The benefits of this design is threefold. First, the detector has a constant false alarm rate (CFAR). Second, the selection of the detection threshold is simple and may be set by the operator according to the SNR level of the APs being analyzed. Third, the MF output is a scaled amplitude estimate and can be used to improve the tracking performance, see below.

The sampling frequency used is 31.25 kHz and the spectral content of the APs is concentrated to the frequency range between 500 Hz and 1500 Hz. The matched filter constitutes an optimized bandpass filter with passband in this frequency range. The filter output is smooth and the time instants where APs are present may be obtained using a simple peak finding algorithm that reports all peaks above the given threshold m_0. Such asynchronous detectors were discussed in Chapter 2.

Owing to the bandpass character of this filter, see Figure 9.4, it is immaterial whether the raw data or the notched data is used as input because the influence of the hum is negligible in either case. When validating the overall situation and taking all different aspects, theoretical as well as practical, into account, we have sided with using the raw data as input to the MF detector.

In Section 9.5 the performance of this MF detector is analyzed in terms of resolution, variance, detection probability, and false-alarm probability.

9.3 Discrimination

Finding a reliable algorithm to discriminate the detected APs was a challenge because traditional methods were not suitable. A pattern matching technique, for example, would have poor performance due to the diminutive differences of the AP shape in these recordings. Moreover, such a technique would disregard the latency information provided by the exploitation of the marking phenomenon.

A clustering algorithm, on the other hand, would indeed use this information, but the dynamics of active C-fibers would be difficult, if not impossible, to handle.

An experienced eye, however, easily takes full advantage of the marking phenomenon and exploits the latency information in subsequent traces by forming tracks of APs that belong to each other. We decided to mimic this by solving the association problem as a tracking problem using the MHT method with Kalman

[6] This is the SNR in the MF output used in the likelihood test and is almost four times larger than the common measure $SNR_{pp} \triangleq \left(\frac{\text{minimum AP peak-to-peak amplitude}}{\text{noise root-mean-square value}}\right)^2$.

filter prediction, see below.

Although the tracks may be found using the latency only, the AP amplitude is utilized to improve the tracking performance. In general, the AP amplitude is different for APs originating from different C-fiber units because the distances between the recording electrode and the nerve fibers differ. Using the scaled amplitude estimate that the MF detector provides simplifies the design and has performed well.

9.3.1 Tracking algorithm – multiple hypothesis tracking

The performance of the tracking algorithm is of vital importance and, hence, we selected the MHT [69] method because it is recognized as the theoretically best approach to multitarget tracking problems, c. f. Chapter 4. It's performance is superior to other methods in applications with heavy *clutter* and high *traffic densities*. In our context, clutter would be spurious APs and false detections that are uncorrelated with other APs. High traffic densities means that different C-fibers have very similar latencies.

As mentioned in Section 4.4, a model of how the AP latencies change from trace to trace is needed to evaluate the probability of each hypothesis. Here, all this information is provided by the prediction algorithm and the actual scoring algorithm is simplified, see below.

Scoring algorithm

Using the Bayesian approach mentioned in Section 4.4.2, the score for a particular hypothesis is the logarithm of its a posteriori probability scaled in such a way that an all-false-alarm partitioning of the observations yields a zero score.

To calculate the hypothesis score, we then need the score $L_i(k)$ for each track i in trace k. The general expressions in (4.2) and (4.3) here become

$$L_i(k) = L_i(k-1) + \Delta L_i(k), \quad k > k_0 + 1 \tag{9.2}$$

$$\Delta L_i(k) \triangleq \begin{cases} \ln(1-\hat{P}_D(k)), & \text{not updated} \\ \ln\left[\frac{\hat{P}_D(k)}{\beta_{FT} 2\pi\sqrt{\det \mathbf{S}_i(k)}}\right] - d_i^2(k), & \text{updated} \end{cases} \tag{9.3}$$

with initial values

$$L_i(k_0) \triangleq \ln\left[1 + \frac{\beta_{NT}}{\beta_{FT}}\right] \tag{9.4}$$

$$L_i(k_0+1) \triangleq \ln\frac{\beta_{NT}}{\beta_{FT}} + \ln\left[\frac{\hat{P}_D(k)}{\beta_{FT} 2\pi\sqrt{\det \mathbf{S}_i(k_0+1)}}\right] - \ln d_i^2(k_0+1) \tag{9.5}$$

where $\hat{P}_D(k)$ is the estimated probability of detection, β_{FT} and β_{NT} are the false alarm and the new source density, respectively, and $d_i^2(k)$ is the normalized distance for the observation of track i in trace k defined in (4.4). The matrix $\mathbf{S}_i(k)$ is the measurement prediction error covariance matrix for the observation of track i in trace k. It is obtained from the Kalman predictor.

As the tracking filter needs one update (two observations) to initialize, all potential tracks with a missed detection are rejected. This results in a simplified score calculation of tentative tracks as given by (9.5). Also, experience has shown that better performance is achieved when $\ln d_i^2(k_0+1)$ is used in (9.3) instead of $d_i^2(k_0+1)$. This gives a slight preference to tracks with a stable latency.

9.3.2 Prediction algorithm – Kalman filter

Fundamental in any tracking system is the track prediction and filtering. The two major alternative methods are the Kalman filter [46] and the filter based on interacting multiple models (IMM) [56]. The IMM method is an extension of the multiple model (MM) approach to handle model switching and often provides the best performance. The IMM method is suboptimal, however, and in our application no model switches occur. This suggests that the MM method would give the best performance. At present, the Kalman filter is used, but extending this to an MM based filtering method is straight-forward if required, see Section 3.4.

In addition to this, the Kalman filter has a number of other advantages for use in tracking applications including:

- The state-space form is used to describe the Kalman filter and gives a recursive implementation that reduces the computational demand.

- The characteristics of the filter is altered by changing only a few key parameters. This means that the same filter may be used for varying target and measurement environments.

- The Kalman filter can be used in applications with varying sampling interval and missed detections.

- The innovations covariance matrix provides a convenient measure of the estimation accuracy required to perform the gating and correlation functions of the MHT algorithm accurately.

The target model consists of two submodels, the process model and the measurement model, see Chapter 3. In the context of the application presented in this chapter, the process model describes the target dynamic process and describes how the latency and the matched filter peak output change from scan to scan. The measurement model captures the effects of the measurement system as the addition of measurement noise, for example.

As the process under study is a real physical process assumed to be working in continuous-time, a model is based on this assumption. To incorporate such a model into the tracking process, a discrete-time model is produced by sampling the continuous-time model. The measurements, however, are modeled as a discrete-time process directly.

Process model

From Figure 9.1 it seems reasonable to assume that the measured latency, denoted $y_1(t)$, may be modeled by

$$y_1(t) \triangleq \bar{y}_1(t) + \varepsilon(t) \tag{9.6}$$

$$\bar{y}_1(t) \triangleq y_0 + Ae^{-\alpha_0(t-t_0)}, \quad t \geq t_0 \tag{9.7}$$

where $\bar{y}_1(t)$ is the *true* but unknown latency, $\varepsilon(t)$ is the residual consisting of model and measurement errors, y_0 is the latency at steady state, A is the latency shift due to stimulation, α_0 is the recovery coefficient, and t_0 is the time of excitation, here assumed identical with the start of the track/recovery.

By selecting the true latency $\bar{y}_1(t)$ and its derivative $\dot{\bar{y}}_1(t)$ as state variables, (9.7) may be written in state space form as

$$\begin{pmatrix} \dot{\bar{y}}_1(t) \\ \ddot{\bar{y}}_1(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & -\alpha_0 \end{pmatrix} \begin{pmatrix} \bar{y}_1(t) \\ \dot{\bar{y}}_1(t) \end{pmatrix}, \quad t \geq t_0 \tag{9.8}$$

$$\bar{y}_1(t_0) = y_0 + A \tag{9.9}$$

$$\dot{\bar{y}}_1(t_0) = -\alpha_0 A \tag{9.10}$$

which is an initial value problem with (9.7) as its solution. It is simple to augment this deterministic state space model with an additional state that describes the AP amplitude assumed to be constant.

To account for model errors like errors in the unknown parameter α_0 and slow variations in the latency and the AP amplitude, it is conventional to add noise terms to the uncertain states [12]. The *deterministic* continuous-time state space model (9.8) augmented with the AP amplitude is thus replaced by a *stochastic* continuous-time state space model, c. f. Section 3.1

$$\mathbf{dx}_c(t) = \mathbf{A}\mathbf{x}_c(t)\,dt + \mathbf{G}\,\mathbf{dw}(t), \quad t \geq t_0 \tag{9.11}$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\alpha & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{9.12}$$

$$\mathbf{G} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{9.13}$$

where $\mathbf{x}_c(t)$ is the three-dimensional continuous-time state vector made up of the latency $\bar{y}_1(t)$, its derivative $\dot{\bar{y}}_1(t)$, and the MF output $\sqrt{SNR_{mf}}$. The two-dimensional entity $\mathbf{dw}(t)$ is the Wiener increment of the two-dimensional Wiener process $\mathbf{w}(t)$ and the matrix $\mathbf{A}$ is the state transition matrix, in which α should be as close as possible to the true recovery coefficient α_0. This model is very similar to the one used in Chapter 3, but the modeling of the process noise is different, see below.

The latency part of the model could also be viewed as a first order system driven by an exponentially autocorrelated zero mean process noise. Note that the limit $\alpha = 0$ results in the common first order white noise acceleration model described in [12].

The main difference between the model used in this chapter and the example model in Section 3.3 is the modeling of the process noise for the first subsystem that models the latency. Because the latency model is the stochastic state-space version of the model in (9.7), the process noise here accounting for the latency modeling errors are largest at the start of the decay and are almost zero at steady state. This is represented by defining the incremental variance of the first element of $\mathbf{dw}(t)$, denoted $dw_1(t)$, as

$$\begin{aligned} \sigma^2_{\lambda_1}(t)\,dt &\stackrel{\triangle}{=} E\,(dw_1(t))^2 \\ &= \sigma^2_{\lambda_1}(1 + e^{\beta_0 - \beta_1(t-t_0)})\,dt, \quad t \geq t_0 \end{aligned} \tag{9.14}$$

where β_i are tuning parameters. This results in a process noise that is high at onset and decreases exponentially to its minimum. Note that this affects the latency part of the system only. For the modeling of the amplitude estimate, a constant incremental variance $\sigma^2_{\lambda_2} dt$ is used because we assume that the amplitude is constant or is varying slowly.

Consequently, the incremental covariance matrix $\mathbf{\Lambda}(t)\,dt$ of the Wiener increment $\mathbf{dw}(t)$ is dynamic and is given by

$$\mathbf{\Lambda}(t)\,dt \stackrel{\triangle}{=} E\,\mathbf{dw}(t)\mathbf{dw}^T(t) = \begin{pmatrix} \sigma^2_{\lambda_1}(t) & 0 \\ 0 & \sigma^2_{\lambda_2} \end{pmatrix} dt \quad . \tag{9.15}$$

The discrete-time process model

The utilization of the marking phenomenon, i.e., emitting electrical impulses periodically and measuring the latency of the evoked APs, may be interpreted as a sampling of the (continuous-time) latency of the excited units with a sampling interval equal to the period of the emitted stimuli. The time difference between the emission of the impulse and the detection of the evoked AP of a particular C-unit is then a measurement of the current latency of this unit. Any additional APs manifest themselves by increased time differences corresponding to an increased C-unit latency in accordance with the marking phenomenon.

By sampling the continuous-time model, see Section 3.1.2, we obtain the process model in trace k as

$$\mathbf{x}(k+1) \quad = \quad \mathbf{F}\mathbf{x}(k) + \mathbf{v}_1(k), \quad k \geq k_0 \stackrel{\triangle}{=} \frac{t_0}{T} \tag{9.16}$$

where T is the period of the repetitive electrical stimulation. The matrix $\mathbf{F}$ is the transition matrix, $\mathbf{v}_1(k)$ is a three-dimensional vector that represents the process noise modeled as zero-mean, white-noise Gaussian processes with covariance matrix $\mathbf{Q}_1(k)$. These entities are defined in (3.9)-(3.12) which in this case become, see Appendix 3.B and Appendix 9.A,

$$\mathbf{F} \quad = \quad \begin{pmatrix} 1 & \alpha^{-1}(1-e^{-\alpha T}) & 0 \\ 0 & e^{-\alpha T} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{9.17}$$

$$\mathbf{Q}_1(k) \quad = \quad \begin{pmatrix} q^{(1)}_{11} & q^{(1)}_{12} & 0 \\ q^{(1)}_{21} & q^{(1)}_{22} & 0 \\ 0 & 0 & q^{(1)}_{33} \end{pmatrix} \Sigma_0 + \begin{pmatrix} \rho_{11} & \rho_{12} & 0 \\ \rho_{21} & \rho_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \Sigma(kT) \tag{9.18}$$

where $q^{(1)}_{ij}$, Σ_0, ρ_{ij}, and $\Sigma(kT)$ are given in Appendix 9.A.

Measurement model

The measurement model is identical to the one given in Section 3.3 which is simply

$$\mathbf{y}(k) = \mathbf{H}\mathbf{x}(k) + \mathbf{v}_2(k), \quad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{9.19}$$

where $\mathbf{y}(k)$ is a two-dimensional measurement vector containing the measured latency and the MF output. The two-dimensional vector $\mathbf{v}_2(k)$ is the measurement noise, modeled as independent zero-mean white-noise Gaussian processes with a known constant $2|2$ covariance matrix given by

$$\mathbf{Q}_2 = \begin{pmatrix} q_{11}^{(2)} & 0 \\ 0 & q_{22}^{(2)} \end{pmatrix} . \tag{9.20}$$

Finally, the process noise and the measurement noise are assumed independent which means that

$$\mathbf{Q}_{12} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} . \tag{9.21}$$

Initialization and consistency

Initial estimates of the state and its covariance matrix need to be found before applying the Kalman algorithm. Using two of the first measurements it is possible to find an MMSE estimate of the initial values, see Section 3.2.2 for details. To be able to use the gating technique, see Section 4.4.4, intermediate estimates are used as presented in Appendix 9.B.

Moreover, the model parameters need to be tuned to yield consistent estimates. It is advisable to check this using the methods described in Chapter 3.

9.3.3 Detection-probability estimation

The detection probability P_D may change during the experiment and needs to be estimated for each target and trace. Since the detection thresholds used in practice are $\tilde{m}_0 \approx 5$ and there seems to be a minimal difference between the synchronous MF detector (SMFD) and AMFD performance for such thresholds, c. f. Figure 9.8, we use the SMFD expression in (2.26) to estimate the instantaneous detection probability $\hat{p}_D(k)$ in trace k where the MF output is used as an estimate of the square root of the instantaneous SNR. Hence, we have defined $\hat{p}_D(k)$ as

$$\hat{p}_D(k) \stackrel{\triangle}{=} 1 - \Phi(\tilde{m}_0 - \mathbf{h}_2^T \hat{\mathbf{x}}(k|k)), \quad k > k_0 \tag{9.22}$$

where $\tilde{m}_0$ is selected by the operator, h_2^T is the second row of the measurement matrix H and $\hat{\mathbf{x}}(k|k)$ is provided by the Kalman filter. Thus, $\mathbf{h}_2^T \hat{\mathbf{x}}(k|k)$ is the Kalman filter estimate of the square root of the instantaneous SNR.

The consequence of this choice is that low SNR APs get their detection probability over-estimated which, in turn, lead to a larger "cost" for missed detections. This has, however, not been a problem in practice.

By lowpass filtering the instantaneous detection-probability estimate, the detection-probability estimate is improved in terms of tracking performance. Since the detection-probability estimate is non-optimal and because extreme detection probabilities yield bad tracking, the operator may also specify minimal and maximal values for the detection probability estimate. This is summarized by

$$\bar{P}_D(k) \stackrel{\triangle}{=} (1 - \lambda_{P_D})\hat{P}_D(k-1) + \lambda_{P_D}\hat{p}_D(k) \tag{9.23}$$

$$\hat{P}_D(k) \stackrel{\triangle}{=} \max(P_{D,min}, \min(P_{D,max}, \bar{P}_D(k))), \quad k > k_0 \tag{9.24}$$

where λ_{P_D} is a forgetting factor and $\hat{P}_D(k_0)$ may set by the operator. This estimate has, however, a positive bias for small APs as the MF output values below the threshold do not contribute. The bias may be calculated and the estimate corrected, but it is believed to be of small importance (to the tracking result) and is therefore not considered.

The actual detection probability, from the tracker's perspective, is the mutual probability of the MF detection and the gating. Normally, only large gates are considered and the effect of the gating is consequently neglected.

9.4 Parameter estimation

Estimating model parameters of a particular model once the data is available is, in principle, straight-forward. However, the estimation is complicated somewhat due to the nonlinear data model used here. In the model (9.7) the parameter α_0 enters nonlinearly, whereas the two parameters y_0 and A enter linearly. Our approach is an iterative method in which the nonlinear term is estimated using the simplex method. For each step in the simplex algorithm, the two linear terms are estimated through the LS method and the error norm is returned.

9.4.1 Model function

The model function $\bar{\boldsymbol{\varphi}}(\boldsymbol{\theta})$ and the data $\boldsymbol{\varphi}$ are defined by

$$\bar{\boldsymbol{\varphi}}(\boldsymbol{\theta}) = \begin{pmatrix} \bar{y}_1(k_0, \boldsymbol{\theta}) & \dots & \bar{y}_1(k_{N-1}, \boldsymbol{\theta}) \end{pmatrix}^T \tag{9.25}$$

$$\boldsymbol{\varphi} = \begin{pmatrix} y_1(k_0) & \dots & y_1(k_{N-1}) \end{pmatrix}^T \tag{9.26}$$

$$\bar{y}_1(k, \boldsymbol{\theta}) = y_0 + Ae^{-\alpha(k-k_0)T}, \quad k \in K \tag{9.27}$$

$$\boldsymbol{\theta} \triangleq \begin{pmatrix} \alpha & y_0 & A \end{pmatrix}^T \tag{9.28}$$

$$K \triangleq \{k_0, k_1, \dots, k_{N-1}\} \tag{9.29}$$

where $\bar{y}_1(k, \boldsymbol{\theta})$ is a parametrized model of the latency defined in accordance with the assumed time course of the latency in (9.7). Each element $y_1(k_i)$ is the measured latency (the first element of the measurement vector $\mathbf{y}(k_i)$) of the AP assigned to this track in trace number k_i. The set K contains all trace numbers for which an AP has been assigned to this track and N is the number of assignments.

The set of (nonlinear) regression equations may then be written

$$\boldsymbol{\varphi} = \bar{\boldsymbol{\varphi}}(\boldsymbol{\theta}) + \boldsymbol{\varepsilon} \tag{9.30}$$

where $\boldsymbol{\varepsilon}$ is an N-dimensional residual column vector. This system of equations is solved to find the parameter estimate $\hat{\boldsymbol{\theta}}$ that minimizes $\frac{1}{N}\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$.

In the model (9.27) the parameter α enters nonlinearly, whereas the two parameters y_0 and A enter linearly. The parameters is estimated by an iterative method in which the nonlinear term is estimated using the simplex method [23] [55]. For each step in the simplex algorithm, the two linear terms are estimated through the LS method and the error norm is returned. The simplex method thus seeks the parameter α through

$$\hat{\alpha} = \arg\min_{\alpha} \sum_{k \in K} \left| y_1(k) - \bar{y}_1(k, \alpha | \hat{y}_0, \hat{A}) \right|^2 . \tag{9.31}$$

The two linear terms $\hat{y}_0$ and $\hat{A}$ are estimated with the LS method by solving the over-determined system of equations

$$\mathbf{J}_{lin}(\alpha) \begin{pmatrix} \hat{y}_0 \\ \hat{A} \end{pmatrix} = \boldsymbol{\varphi} \tag{9.32}$$

where $\mathbf{J}_{lin}$ is found by deriving the Jacobian [68] of the parametrized model. The

Jacobian is defined by

$$\mathbf{J}(\boldsymbol{\theta}) \triangleq \frac{\partial}{\partial \boldsymbol{\theta}^T}\bar{\boldsymbol{\varphi}}(\boldsymbol{\theta}) \equiv \begin{pmatrix} \mathbf{j}_\alpha(\alpha, A) & \mathbf{J}_{lin}(\alpha) \end{pmatrix} \tag{9.33}$$

$$\mathbf{j}_\alpha(\alpha, A) = \begin{pmatrix} 0 \\ -(k_1 - k_0)TAe^{-\alpha(k_1-k_0)T} \\ \vdots \\ -(k_{N-1} - k_0)TAe^{-\alpha(k_{N-1}-k_0)T} \end{pmatrix} \tag{9.34}$$

$$\mathbf{J}_{lin}(\alpha) = \begin{pmatrix} 1 & 1 \\ 1 & e^{-\alpha(k_1-k_0)T} \\ \vdots & \vdots \\ 1 & e^{-\alpha(k_{N-1}-k_0)T} \end{pmatrix}. \tag{9.35}$$

The initial value of the recovery constant α is set to the slope coefficient of the linear regression of the data.[7]

9.4.2 Confidence intervals

In [36], confidence intervals of the estimates are derived for a parametrized vector function $\bar{\boldsymbol{\varphi}}(\boldsymbol{\theta}_0)$ where $\boldsymbol{\theta}_0$ represents the true parameter vector. Given a parameter estimate $\hat{\boldsymbol{\theta}}$ and some scalar function $\phi(\boldsymbol{\theta})$, a confidence interval I_ϑ with confidence level ϑ of the scalar $\phi(\boldsymbol{\theta}_0)$ is

$$I_\vartheta = \{\phi(\boldsymbol{\theta}) \mid P(\phi(\boldsymbol{\theta}_0) \in I_\vartheta) = 1 - \vartheta\} \tag{9.36}$$

$$= \left[\phi(\hat{\boldsymbol{\theta}}) \pm t_{N-3;1-\vartheta/2}\sqrt{s^2\hat{\Phi}\hat{\mathbf{C}}\hat{\Phi}^T}\right] \tag{9.37}$$

$$\hat{\mathbf{C}} \triangleq \left(\mathbf{J}^T(\hat{\boldsymbol{\theta}})\mathbf{J}(\hat{\boldsymbol{\theta}})\right)^{-1} \tag{9.38}$$

$$\hat{\Phi} \triangleq \left.\frac{\partial}{\partial \boldsymbol{\theta}^T}\phi(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \tag{9.39}$$

$$s^2 = \frac{1}{N-3}\left(\boldsymbol{\varphi} - \hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\theta}})\right)^T\left(\boldsymbol{\varphi} - \hat{\boldsymbol{\varphi}}(\hat{\boldsymbol{\theta}})\right) \tag{9.40}$$

where $t_{N-3;1-\vartheta/2}$ denotes the upper $1 - \vartheta/2$ critical point of the t-distribution with $N - 3$ degrees of freedom. The confidence interval of the estimates is thus

[7] This results in a good initial estimate for short (with respect to the recovery constant) data sets, but has the drawback of yielding worse estimates for long data sets.

calculated as

$$I_{\vartheta}^{(y_0)} = \hat{y}_0 \pm t_{N-3;1-\vartheta/2}\sqrt{s^2\hat{c}_{11}} \tag{9.41}$$
$$I_{\vartheta}^{(A)} = \hat{A} \pm t_{N-3;1-\vartheta/2}\sqrt{s^2\hat{c}_{22}} \tag{9.42}$$
$$I_{\vartheta}^{(\alpha)} = \hat{\alpha} \pm t_{N-3;1-\vartheta/2}\sqrt{s^2\hat{c}_{33}} \tag{9.43}$$

where $\hat{c}_{ii}$ are the diagonal elements of $\hat{\mathbf{C}}$ in (9.38). Note that the matrix $\hat{\mathbf{C}}$ may be badly conditioned and that the columns in $\mathbf{J}(\hat{\boldsymbol{\theta}})$ could need a rescaling. Unfortunately, the formulas above are only asymptotically valid and the above confidence intervals estimates are larger than the actual confidence intervals for the relatively short track segments used in the application at hand.

9.5 Simulation results

The framework presented in Chapter 2 is used below to assess the performance and characteristics of the MF detector as its properties are essential to the tuning of the Kalman filter and also affect the tuning of the MHT algorithm.

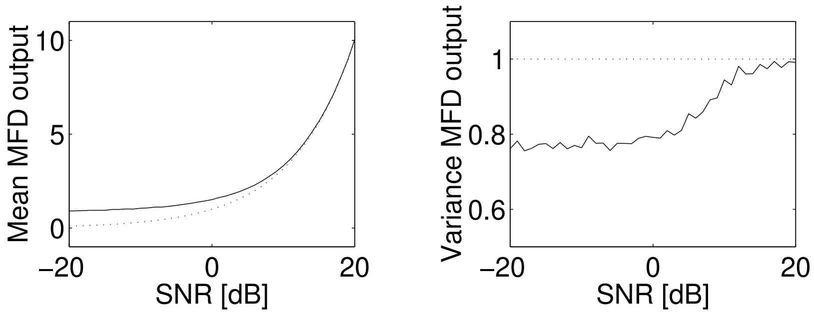
9.5.1 The matched-filter detector

Using the techniques described in Section 2.2, we here present the performance of the AMFD tuned with a signal template acquired from real recordings as described in Section 9.2.1. Throughout the simulations of the matched filter, 1000 Monte Carlo runs were performed with the detection threshold set to $m_0 = 1$ unless stated differently.

To find the bandwidth ratio of the signal template, see (2.20) in Section 2.2.1, we use the following definitions for the bandwidth B and the center frequency f_0

$$S_{norm}(f) \triangleq \frac{S(f)}{\max_f S(f)} \tag{9.44}$$
$$2B \triangleq \int_0^{\infty} |S_{norm}(f)|^2 \, df \tag{9.45}$$
$$f_0 \triangleq \frac{1}{2B}\int_0^{\infty} f|S_{norm}(f)|^2 \, df \tag{9.46}$$

where $S(f)$ is the spectrum of the signal template in Figure 9.4. The bandwidth B as defined above is often called the *noise bandwidth*. To find the spectrum $S(f)$ we take the acquired signal template, pad it with zeros to a length of 1024 samples,



(a) (b)

Figure 9.5: (a) The estimated mean value of the AMFD output (solid) and the theoretical mean value of the SMFD output (dotted) as a function of the SNR. (b) The estimated variance of the AMFD output and the theoretical variance of the SMFD output (dotted) as a function of the SNR. The results were obtained from 10^4 Monte Carlo runs.

and take the discrete Fourier transform. Using this technique, we obtain

$$B = 0.667 \quad \text{[kHz]} \tag{9.47}$$

$$f_0 = 1.15 \quad \text{[kHz]} \tag{9.48}$$

$$\gamma = \frac{f_0}{B} = 1.72 \ . \tag{9.49}$$

Detection and false-alarm probability

In several of the equations constituting the MHT algorithm, the detection and false alarm probabilities are key parameters. Here we present the performance of the AMFD compared to the SMFD with respect to detection and false alarm probability to expose any performance loss due to the unknown arrival time. Most results are actually similar to the results from the generic AMFD presented in Chapter 2 despite the signal templates being different. Hence, we present only the properties that are different and of interest to this application.

The diagrams in Figure 9.5, from 10^4 Monte Carlo runs, clearly show the results expected from Chapter 2, c. f. Figure 2.4 ($\gamma = 1.72$). The main difference is that for the AMFD used here, the variance for small SNRs is larger, i.e., closer to the SMFD variance (=1), compared to the corresponding AMFD ($\gamma = 1.72$) from Chapter 2.

Figure 9.6 is a replica of Figure 2.7 where the estimates, using 10^4 Monte Carlo runs, of the mean and variance of the AMFD used here when no signal is present are inserted. The result is an improvement compared to the generic AMFD presented in Chapter 2; the mean is closer to zero and the variance is closer to one.

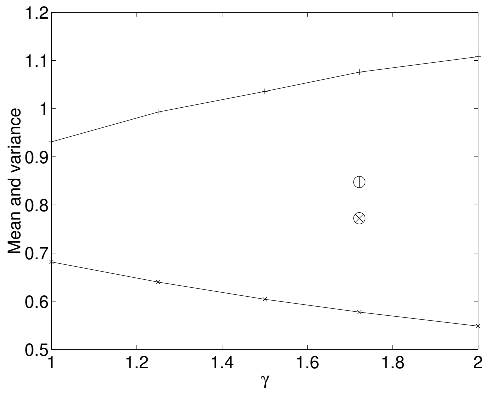


Figure 9.6: This figure is a replica of Figure 2.7 where the estimates, using 10^4 Monte Carlo runs, of the mean ($+$) and variance ($\times$) of $m(\hat{n}_a)$ when no signal is present are plotted as a function of the bandwidth ratio γ. The values for the AMFD used here are marked ($\circ$).

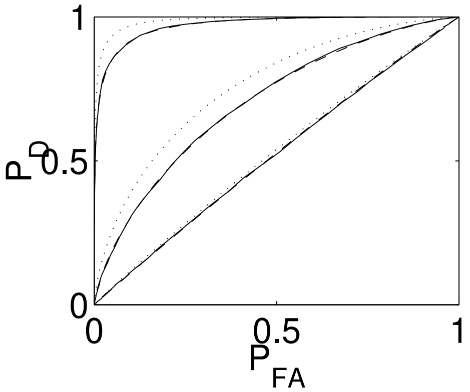


Figure 9.7: The figure shows the estimated receiver operating characteristics (ROC) curve of the AMFD (solid), the ROC curve using the approximative false alarm and detection probabilities of the AMFD (dashed), and the theoretical ROC curve of the SMFD (dotted) for the SNR levels -20 dB, 0 dB, and 10 dB (curves right to left).

In Figure 9.7 the estimated receiver operating characteristics (ROC) curves of the AMFD for the SNR levels -20 dB, 0 dB, and 10 dB are shown. As can be seen in the figure, the AMFD performs in general well but has a slight performance loss compared to the SMFD, due to the unknown arrival time n_a.

Figure 9.8 compares the detection probability of the AMFD with the SMFD as a function of the SNR for different false-alarm probabilities. The threshold

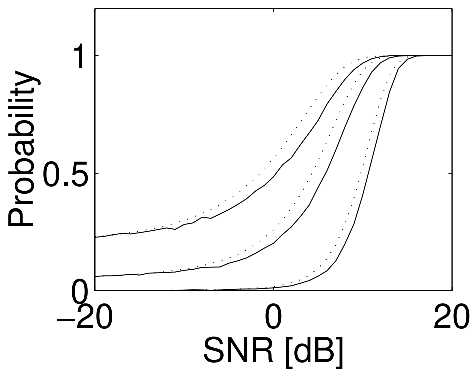


Figure 9.8: The figure shows the estimated detection probability of the AMFD (solid) and the theoretical value of the SMFD (dotted) as a function of the SNR. The threshold m_0 of the SMFD was set to achieve the false-alarm probabilities (top to bottom) 0.2, 0.05, and 0.001, corresponding to $m_0 = 0.84, 1.6, 3.1$, respectively, whereas the threshold $\tilde{m}_0$ of the AMFD was set by the transformation (2.30) to yield equal false-alarm probabilities.

m_0 of the SMFD was set to achieve the false-alarm probabilities 0.2, 0.05, and 0.001, corresponding to $m_0 = 0.84, 1.6, 3.1$, respectively. The detection threshold transformation (2.30) is used to yield equal false-alarm probabilities and we see that the performance loss is limited to less than 3 dB which is acceptable for this application. At higher thresholds and higher SNR levels, the results suggests even that the two detectors are comparable in performance. A note of precaution is, however, necessary here; see Section 2.2 for details.

In Figure 9.9 the false-alarm intensity of the AMFD used here is shown and compared to the AMFDs in Chapter 2. The figure is a replica of Figure 2.10 where the estimated false-alarm intensity based on 10^6 Monte Carlo runs for the AMFD used here is inserted using the detection threshold $m_0 = -\infty$. This threshold yields the maximum false-alarm intensity because $\widetilde{P}_{FA} = 1$. As the figure clearly shows, the false-alarm intensity is almost 50 % higher because the MF used here does not totally suppress the frequencies above $f_0 + B$, c. f. Figure 9.4, which was the case in Chapter 2.

Accuracy

To tune the Kalman filter properly, we need to know the variance of the latency measurements, i.e. the accuracy of the AMFD. Using the techniques described in Section 2.2.3, the accuracy of the AMFD is investigated here.

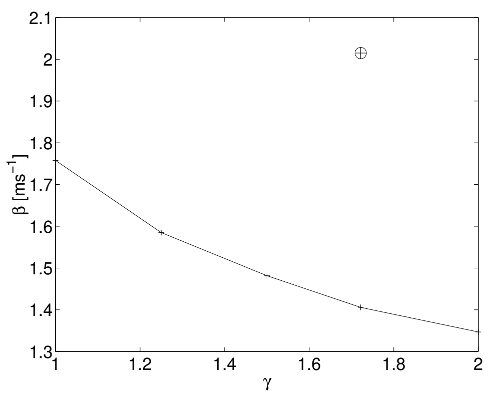


Figure 9.9: This figure is a replica of Figure 2.10 where the estimated false-alarm intensity based on 10^6 Monte Carlo runs are shown as a function of the bandwidth ratio γ using the detection threshold $m_0 = -\infty$. This threshold yields the maximum false-alarm intensity because $\widetilde{P}_{FA} = 1$. The false-alarm intensity for the AMFD used here is marked ($\circ$).

When investigating histograms of the error in the estimated arrival time, it seemed reasonable, for high SNR levels, to assume that it is Gaussian and zero mean. For smaller SNRs, the error becomes more and more uniformly distributed over the detection interval.

Figure 9.10 clearly shows this behavior where the variance is small for high SNRs and increases when the SNR decreases. As expected, the variance approaches the variance of a uniform distribution over the simulated interval.

Resolution

An important property of the AMFD is its resolution capability because it influences the robustness of the tracking of the APs. Analogous to Section 2.2.4, we here present the resolution capabilities of the AMFD used here.

Figure 9.11 (b) shows the probability of resolving both signals. As expected, this probability decreases to zero for small time differences. The probability of detecting at least one AP increases instead, because the APs add and interfere constructively. In between, there is a dip in the detection probability between 0.2 ms and 0.4 ms (for both APs) where the two APs interfere destructively with one another.

From this, we conclude that the tracker must be robust with respect to changes in the detection probability at track crossings. If not, tracks may be lost or swapped.

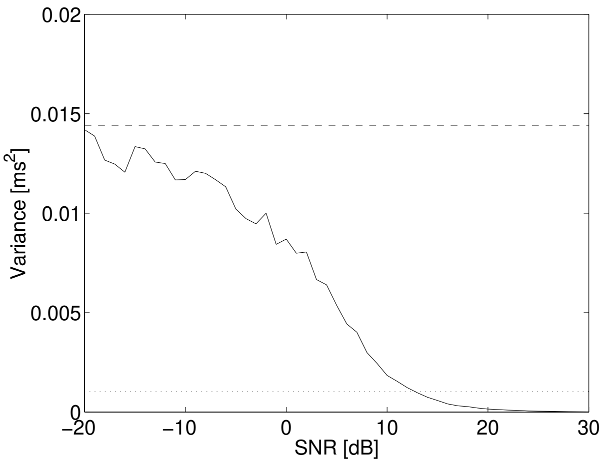


Figure 9.10: The figure shows the estimated variance (solid) of the arrival time with the MF output oversampled ten times prior to the detection. For decreasing SNR levels, the variance increases asymptotically towards the variance of a uniform distribution over the detection interval (dashed). For comparison, the squared (original) sample period ($1/f_s^2$) is shown (dotted).

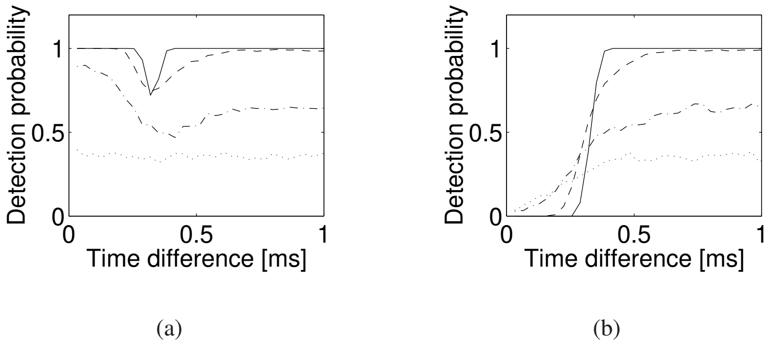


Figure 9.11: Two identical signals are recorded. The diagrams show the estimated probabilities of detecting (a) at least one signal and (b) both signals for different time separations and SNR levels, -20 dB (dotted), 0 dB (dash dotted), 10 dB (dashed), and 20 dB (solid).

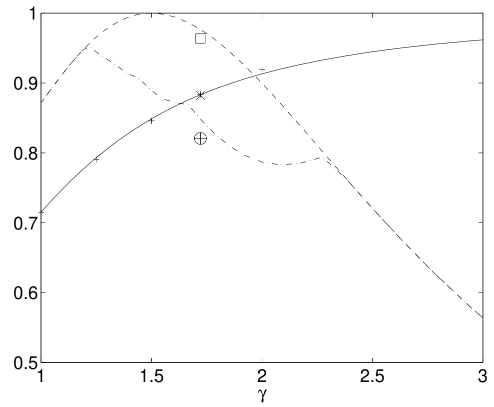


Figure 9.12: This figure is a replica of Figure 2.12 showing the product $2f_0\Delta t_R$ (solid) where Δt_R is the Rayleigh resolution, the primary sidelobe rejection (dashed), and global sidelobe rejection (dash-dotted). Also included is the estimated resolution based on Monte Carlo simulations yielding a detection probability $P_D = 0.85$ (+) with the one for the AMFD used here marked (∘). The Rayleigh resolution (×) and the global sidelobe rejection (□) for the AMFD used here are also shown.

Applying the resolution definition (2.31), we here obtain

$$\Delta_{lat} = 0.36 \quad [\text{ms}] \ . \tag{9.50}$$

In Figure 9.12 the resolution is compared between the AMFD used here and the AMFD used in Chapter 2. As the figure shows, the Rayleigh resolution for the AMFD used here and for the AMFD used in Chapter 2 are identical. The $P_D = 0.85$ resolution, however, is significantly lower. Moreover, the global sidelobe rejection of the AMFD used here is close to the primary sidelobe rejection of the AMFD used in Chapter 2.

9.6 Experimental results

As stated in Section 9.1, the recordings are processed in three steps by detecting the APs in the recorded traces, tracking their latency changes, and estimating their latency parameters.

This section presents the performance of these steps when applied to real recordings from human subjects. First, it is made clear that the MF output has an improved detection capability compared to the original recording. Following that, the performance of the MHT tracker is illustrated in two different scenarios, one

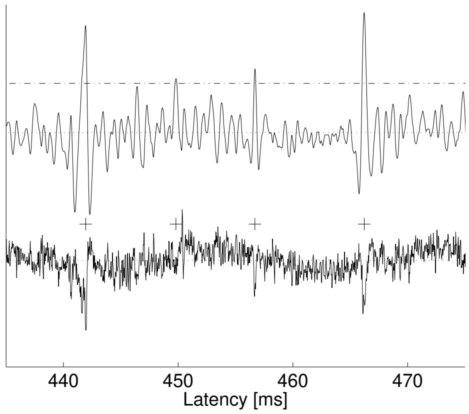


Figure 9.13: The output of the matched filter (top) when applied on a part of scan 5 in the recording in Figure 9.17 (bottom). With the same threshold setting, four detections are reported (+). Note also the 50 Hz hum in the recorded data.

simpler and one harder. Finally, the estimated parameters of one of the active units found by the tracker are presented.

9.6.1 The matched-filter detector

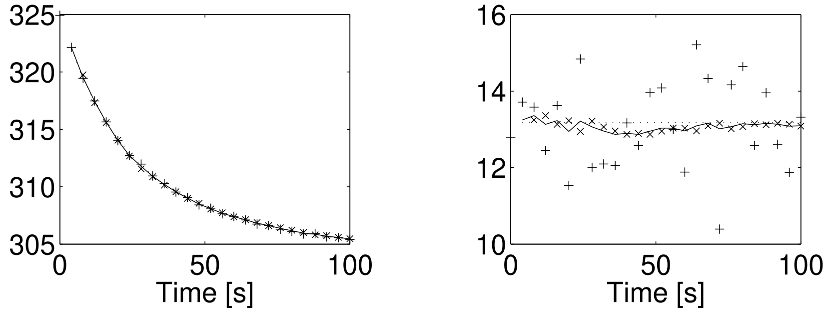
Despite the erroneous assumptions when deriving the AMFD, the anticipated reduction in detection performance has not been a problem in practice. The AMFD is robust in terms of different AP shapes and there is no need for retuning it for different recordings. In addition, it has been incorporated in the on-line data acquisition system to facilitate the experiments.

In Figure 9.13 a part of the fifth scan in Figure 9.17 is shown. As may be seen from Figure 9.13, the MF output is an improvement to the original data. With the detections from several scans (see Figure 9.17), it may be concluded that the four reported peaks (marked with +) could correspond to four AP detections and that even the low amplitude APs are detected without decreasing the detection threshold down to the noise level.

From Figure 9.13 it is clear that a notch filter must be used to remove the 50 Hz hum in order to avoid a biased noise variance estimate (9.1).

9.6.2 The MHT/Kalman tracker

The most critical part of the overall system is the tracking of the different C-units. For this application to be useful, the number of tracking errors has to be small. Some errors are allowed because erroneous assignments may be corrected by the



(a) (b)

Figure 9.14: The diagrams show the exponentially modeled ("true") output $\mathbf{h}_i\mathbf{x}(k)$ (dotted), the actual measured output $y_i(k)$ (+), the filtered output $\mathbf{h}_i\hat{\mathbf{x}}(k|k)$ (solid), and the predicted output $\mathbf{h}_i\hat{\mathbf{x}}(k|k-1)$ (×), where $\mathbf{h}_i$ is the ith row in $\mathbf{H}$. In each diagram, we see (a) the latency of the measurement using the MMSE initialization, and (b) the MF output of the measurement using the MMSE initialization.

operator prior to the statistical analysis. Below, two examples are presented: one simple and one more complex.

A two unit recording

This basic example involves two well-separated units: one with high and one with low amplitude, see Figure 9.1. After matched filtering, we obtain the data shown in Figure 9.16 where each dot represents a MF output above the selected detection threshold $\tau = 5$. First, both units were inactive and their latencies were constant at about 303 ms and 360 ms, respectively. At trace 13, the high amplitude unit was activated by a mechanical stimulus and its latency increased dramatically. The latency then slowly recovered to the level before the activation.

To get the best tracking result possible, the Kalman filter settings used on this recording were tuned to their "optimal" values for this particular unit using the consistency analysis described in [12]. The "true" states needed for this were derived by fitting an exponential model similar to (9.6) to the active unit's recovery trajectory, see Figure 9.14. In order to obtain uncorrelated residuals, however, it was necessary to use two exponential terms. The Kalman filter was subjectively tuned to give as consistent estimates as possible and the following parameters were

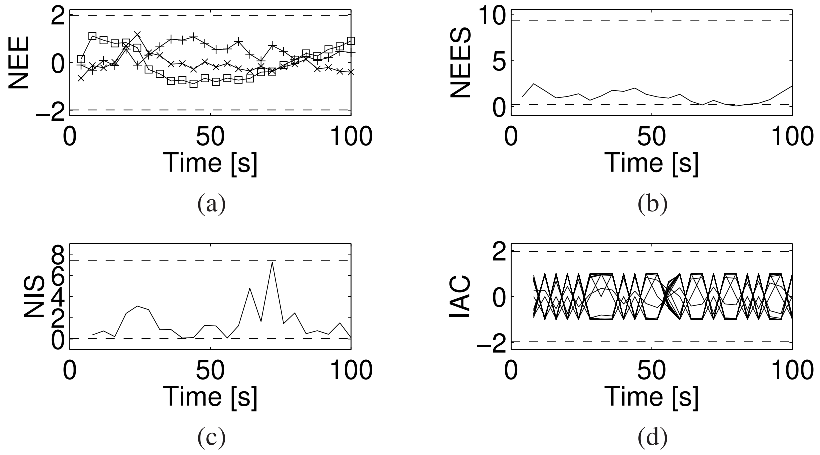


Figure 9.15: Consistency tests, see Section 3.2.3, for the target model on the active unit's recovery trajectory in Figure 9.16. The tests are (a) the average NEE test, (b) the average NEES test, (c) the average NIS test, and (d) the sample IAC test.

selected in the model (9.11)-(9.21) as

$$\begin{array}{rcll rcll}
\alpha & = & 0.0375 & [\mathrm{ms}^{-1}] & \beta_1 & = & 0.05 & [\mathrm{ms}^{-1}] \\
\sigma^2_{\lambda_1} & = & 2\cdot 10^{-5} & [\mathrm{ms}^{-1}] & q^{(2)}_{11} & = & 0.05 & [\mathrm{ms}^2] \\
\sigma^2_{\lambda_2} & = & 10^{-7} & [\mathrm{ms}^{-1}] & q^{(2)}_{22} & = & 1 & . \\
\beta_0 & = & 3 & & & & &
\end{array} \tag{9.51}$$

The consistency tests, see Section 3.2.3, using these parameters are shown in Figure 9.15. As the figure shows, the model passes the tests, but the confidence intervals are so large that its value is limited. Using a qualitative analysis, however, we see that the NEES and NIS tests are close to rejection around 70 s. From the realisation in Figure 9.14(b), we see also that the residuals in the amplitude model are increased which may explain this behavior.

In principle, a robust Kalman filter setting that is optimal for all units is desired. Experience has shown, however, that this is not crucial to the performance (see, for example, below).

As shown in Figure 9.16, the algorithm successfully tracked both units. Note that the track of the low amplitude unit was not lost despite that APs were missed in some traces.

Note also the two high amplitude AP detections in trace 13 and 14 not assigned to any track. They probably originated from the high amplitude unit, but because they were found in the traces directly after the activation of the unit, it is not clear whether they corresponded to the electrical impulses, the mechanical stimulus, or

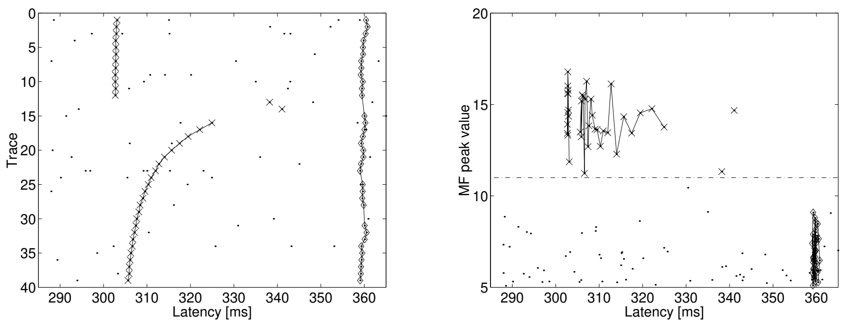


Figure 9.16: The figure shows the confirmed tracks of the tracker when applied to the two C-fiber unit recording in Figure 9.1 ($SNR_{pp} \approx 6.8 = 8.3\ dB$). The two perspectives show the time course of the latencies of the units (left) and the amplitude information of the detected APs where the difference between the two units is clearly seen (right). The APs with an MF peak output above 11 (dash-dotted, right) are marked ($\times$) as well as the APs associated with the unit to the right ($\diamond$). For comments on the two marked APs in trace 13 and 14, see the text.

some spurious after effects of the mechanical stimulus. Hence, this result is considered as “optimal” or at least near “optimal”.

If only the high amplitude unit had been of interest, the tracking would have been simplified by choosing a higher threshold. For example, using $\tau = 11$, only the $\times$-marked APs in the diagrams of Figure 9.16 would have been detected and processed.

The good result despite the low threshold is a strength that is important in more complex situations.

A multi-unit recording

This example is more complex and more realistic and involves several C-units with crossing trajectories. Several of the units were inactive during the recording, and one of the units (drawn with a thick line) was strongly activated between trace 11 and trace 12, see Figure 9.17. Moreover, there were two spontaneously active sympathetic C-units that may be recognized by their more irregular behavior.

It is evident from Figure 9.17 that a good tracking result was obtained for the activated unit as well as for the inactive ones. In this example, no parameter estimation was performed to tune the algorithms. Instead, the algorithm parameters were identical to the ones used in the previous example.

Moreover, the importance of the amplitude information should be obvious as the latency trajectory of the active unit crossed the two sympathetic units, but their amplitudes differed.

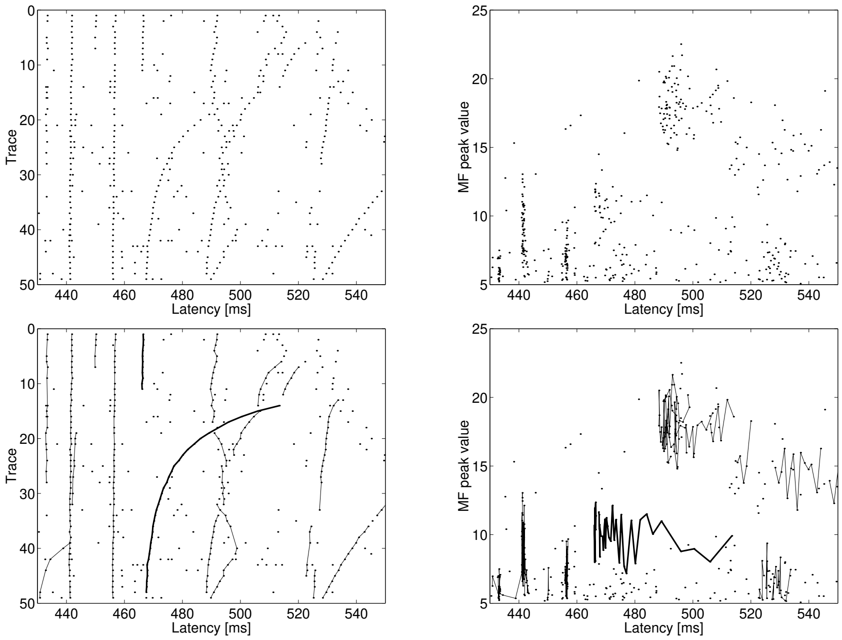


Figure 9.17: A result of the current algorithm where several units were recorded with crossing trajectories ($SNR_{pp} \approx 4.0 = 6.0\ dB$). Both the time course (top left) and the amplitude information (top right) of the detected APs are shown. The tracking result is included and the activated unit is marked with a thick line (bottom left and bottom right). The method handled crossing tracks well because the amplitudes of the APs differed.

Note also the tracker's ability to discriminate between the two tracks at about 440 ms (trace 20-30) despite their closeness and low SNR levels.

Finally, note the lost track between trace 45 and 46 (latency 440 ms). This tracking error may, however, be corrected by the operator prior to the parameter estimation.

From the perspective of the researchers carrying out these experiments, this is a really good result. The otherwise manually task of discriminating the APs is here quickly performed automatically and reliably within minutes.

9.6.3 The parameter estimation

Fitting an exponential curve to a particular track is straight-forward once the data is discriminated and this issue is not pursued in any detail. As an example, however, the results of applying the parameter estimation methods on the latency data of the active unit's trajectory in Figure 9.16 are presented in Table 9.1 along with confidence intervals.

Table 9.1: Parameter estimation of the active unit in Figure 9.16. The residuals turned out to be neither white nor Gaussian.

Parameter		*Value*	*Conf. interval*
y_0	[ms]	305.4	$[305.1, 305.7]$
A	[ms]	19.44	$[19.06, 19.82]$
α	$[\text{ms}^{-1}]$	0.0390	$[0.0369, 0.0411]$
s^2	$[\text{ms}^2]$	0.026	N/A

9.7 Discussion

An application of matched filtering and the MHT method to estimate parameters of human nerve C-fiber APs was presented. The objective of the algorithm was to automate the classification of recorded APs in order to increase the efficiency of the analysis of these recordings.

The APs were successfully detected by means of an MF constituting a ML-CFAR detector. Even APs with amplitudes of the same order of magnitude as the peaks of the noise were detected correctly with a reasonably low false alarm rate. This is an improvement to the previously used nonlinear "noise-cut" filter [93] that amplified only the part of a signal that was above a certain magnitude and thus removed the middle, noisy portion of the signal. Consequently, it did not work well for APs that were of the same order of magnitude as that of the noise itself.

The discrimination between APs originating from different C-fiber units was carried out using the MHT method as described in [13]. Only some minor changes were introduced to adapt the method to the application described. The results corresponded well with what an experienced physician considers to be a "correct" result. Several years of experience also show that the classification is in general correct and that the need for operator intervention is limited.

The estimation of the model parameters was straight-forward using an exponential decay model. The combination of the simplex method and the LS method worked well even for short recovery segments. The estimated parameters have shown to be of limited value to the physician, however.

In spite of the good performance, there are some aspects that could be improved. In the MF derivation, for example, the noise was assumed to be white. As this is not the case in real recordings, the color of the noise should have been considered when the MF impulse response was tuned. An obvious disadvantage with considering the noise color is the possible requirement on estimating it from the data for each experiment which would be very time consuming. Despite this deficiency, the static MF detector has shown to perform well on real recordings as well as to be robust to a wide range of AP shapes. The practical advantages with such a simple and robust detector should be clear.

The most obvious drawback of the MHT algorithm is its exhaustive computational and storage requirements. In the cases considered so far, these requirements have, however, not been a limiting factor. If they were, the track oriented MHT implementation could be used as its computational and storage requirements are less exhaustive [15] [98].

The reason for choosing the MHT method in spite of its complexity is that it is considered to be one of the best tracking methods regarding to performance. Other methods have various shortcomings, for example, the low track maintenance of the nearest neighbor (NN) association method, and the typical track switching behavior of the joint probabilistic data association (JPDA) method are not acceptable [21].

Another drawback of the current implementation is its dependency on errors in the model parameter α that represents an a priori value of the recovery time constant of the latency. It is reasonable to believe, however, that introducing robust filtering methods [91] [64] or replacing the Kalman predictor with an multiple model (MM) based predictor would make the filtering less sensitive to differences in the recovery constant. Implementing the MM approach is probably the most straight-forward because the number of observations to initiate the predictor is kept low. Otherwise, some iterative initialization procedure would have to be included.

In summary, the goal of simplifying and improving the efficiency of the analysis of the human nerve C-fiber recordings was reached. The computer application developed based on the algorithms presented in this paper has already shown to be useful to the research team. In the practical cases considered so far, the overall performance of the application has been excellent.

Appendix 9.A Derivation of the discrete-time process-noise covariance matrix

The covariance matrix $\mathbf{Q}_1(k)$ of the sampled process noise is defined by (3.11) and (3.12). Proceeding as in Appendix 3.C with $\mathbf{A}$ and $\mathbf{G}$ as defined in (9.12) and (9.13), respectively, we obtain

$$\begin{aligned}\mathbf{Q}_1(k) &\triangleq E\,\mathbf{v}_1(k)\mathbf{v}_1^T(k)\\ &= \int_0^T e^{\mathbf{A}(T-\tau)}\mathbf{G}\boldsymbol{\Lambda}(kT+\tau)\mathbf{G}^T e^{\mathbf{A}^T(T-\tau)}\,d\tau\\ &= \int_0^T e^{\mathbf{A}(T-\tau)}\begin{pmatrix}0 & 0 & 0\\ 0 & \sigma_{\lambda_1}^2(kT+\tau) & 0\\ 0 & 0 & \sigma_{\lambda_2}^2\end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau\ .\end{aligned} \tag{9.52}$$

Inserting the expression for $\sigma_{\lambda_1}^2(t)$, c. f. (9.14), we may write

$$\begin{aligned}\mathbf{Q}_1(k) &= \int_0^T e^{\mathbf{A}(T-\tau)}\begin{pmatrix}0 & 0 & 0\\ 0 & \sigma_{\lambda_1}^2(1+e^{\beta_0-\beta_1(kT+\tau-t_0)}) & 0\\ 0 & 0 & \sigma_{\lambda_2}^2\end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau\\ &= \int_0^T e^{\mathbf{A}(T-\tau)}\begin{pmatrix}0 & 0 & 0\\ 0 & \sigma_{\lambda_1}^2 & 0\\ 0 & 0 & \sigma_{\lambda_2}^2\end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau\\ &\quad+\int_0^T e^{\mathbf{A}(T-\tau)}\begin{pmatrix}0 & 0 & 0\\ 0 & \sigma_{\lambda_1}^2 e^{\beta_0-\beta_1(kT+\tau-k_0T)} & 0\\ 0 & 0 & 0\end{pmatrix} e^{\mathbf{A}^T(T-\tau)}\,d\tau\\ &= \begin{pmatrix}q_{11}^{(1)} & q_{12}^{(1)} & 0\\ q_{21}^{(1)} & q_{22}^{(1)} & 0\\ 0 & 0 & q_{33}^{(1)}\end{pmatrix}\Sigma_0\\ &\quad+\int_0^T\begin{pmatrix}b^2(\tau)e^{-\beta_1\tau} & a(\tau)b(\tau)e^{-\beta_1\tau} & 0\\ a(\tau)b(\tau)e^{-\beta_1\tau} & a^2(\tau)e^{-\beta_1\tau} & 0\\ 0 & 0 & 0\end{pmatrix} d\tau\,\Sigma(kT)\end{aligned} \tag{9.53}$$

where $q_{ij}^{(1)}$ and Σ_0 are defined in (3.89)-(3.92) and (3.87), respectively. As in Appendix 3.C, $a(\tau) = e^{-\alpha(T-\tau)}$ and $b(\tau) = \alpha^{-1}(1-e^{-\alpha(T-\tau)})$. The matrix $\Sigma(kT)$ is defined as

$$\Sigma(kT) \triangleq \begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}\sigma_{\lambda_1}^2 e^{\beta_0-\beta_1(k-k_0)T}\ . \tag{9.54}$$

Evaluating the second integral we obtain the final result

$$\mathbf{Q}_1(k) = \begin{pmatrix} q_{11}^{(1)} & q_{12}^{(1)} & 0 \\ q_{21}^{(1)} & q_{22}^{(1)} & 0 \\ 0 & 0 & q_{33}^{(1)} \end{pmatrix} \Sigma_0 + \begin{pmatrix} \rho_{11} & \rho_{12} & 0 \\ \rho_{21} & \rho_{22} & 0 \\ 0 & 0 & 0 \end{pmatrix} \Sigma(kT) \quad (9.55)$$

where $q_{ij}^{(1)}$ and Σ_0 are defined in (3.89)-(3.92) and (3.87), respectively, and

$$\rho_{11} = \alpha^{-2} \Big(\beta_1^{-1}(1 - e^{-\beta_1 T}) - 2(\beta_1 - \alpha)^{-1}(e^{-\alpha T} - e^{-\beta_1 T}) + (\beta_1 - 2\alpha)^{-1}(e^{-2\alpha T} - e^{-\beta_1 T}) \Big) \quad (9.56)$$

$$\rho_{12} = \rho_{21} \quad (9.57)$$

$$= \alpha^{-1} \Big((\beta_1 - \alpha)^{-1}(e^{-\alpha T} - e^{-\beta_1 T}) - (\beta_1 - 2\alpha)^{-1}(e^{-2\alpha T} - e^{-\beta_1 T}) \Big) \quad (9.58)$$

$$\rho_{22} = (\beta_1 - 2\alpha)^{-1}(e^{-2\alpha T} - e^{-\beta_1 T}) \quad (9.59)$$

which concludes the derivation of the discrete-time process noise covariance matrix $\mathbf{Q}_1(k)$.

Appendix 9.B Intermediate state-vector estimate and its covariance matrix

The intermediate state vector $\hat{\mathbf{x}}(k_0+1|k_0)$ and its "covariance" matrix $\mathbf{P}(k_0+1|k_0)$ are needed for the gating of the first candidate update, but they can not be properly calculated because the Kalman filter is initialized first at $k = k_0 + 1$. Nevertheless, practical values, although ad hoc, may be calculated through

$$\hat{\mathbf{x}}(k_0 + 1|k_0) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} \mathbf{y}(k_0) \quad (9.60)$$

$$\mathbf{P}(k_0 + 1|k_0) \triangleq \begin{pmatrix} \mathbf{P}_{11} & \mathbf{0}_{21} \\ \mathbf{0}_{12} & \mathbf{P}_{22} \end{pmatrix} \quad (9.61)$$

$$\mathbf{P}_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \frac{(\dot{y}_{max}^{(1)} T)^2}{G} \quad (9.62)$$

$$\mathbf{P}_{22} = 2q_{22}^{(2)} \quad (9.63)$$

where $\dot{y}_{max}^{(1)}$ is the maximum latency derivative that should be allowed in the gating condition and G is the gate size.

Appendix 9.C MHT parameters and tuning

For the interested reader, the MHT parameter settings used in this Chapter are listed in Table 9.2.

Table 9.2: MHT parameters

Description	*Parameter*		*Value*
A priori detection probability	$\hat{P}_D(k_0)$		0.6
Min detection probability	$\hat{P}_{D,min}$		0.5
Max detection probability	$\hat{P}_{D,max}$		0.9999
Forgetting factor	λ_{P_D}		0.1
New target intensity	β_{NT}	[ms^{-1}]	10^{-7}
False target intensity	β_{FT}	[ms^{-1}]	$2.5 \cdot 10^{-5}$
Gate size	G		12
Deletion score	L_{del}		0
Deletion miss count	N_{del}		3
Confirmation score	L_{conf}		6
Root node depth	N_{prune}		3
Max number of hypotheses	N_{max}		16
Latency process variance	$\sigma^2_{\lambda_1}$	[ms^{-1}]	$2 \cdot 10^{-5}$
Amplitude process variance	$\sigma^2_{\lambda_2}$	[ms^{-1}]	10^{-7}
Process noise constant	β_0		3
Process noise decay	β_1		0.05
Latency measurement variance	$q^{(2)}_{11}$	[ms^2]	0.05
Amplitude measurement variance	$q^{(2)}_{11}$		1
Max latency derivative	$\dot{y}^{(1)}_{max}$		3
Recovery constant	α	[ms^{-1}]	0.0375
Stimulation period	T	[ms]	4

CHAPTER 10

Summary and future work

APPLYING a signal processing approach to the analysis of neurophysiological signals has been challenging due to the competition met from the presently available algorithms. These are well-crafted, numerical algorithms, specifically designed for a particular problem, that often already provide a good balance in the trade-off between practical usability on one hand and resulting accuracy on the other. More subtle competition may involve the automation of a time consuming manual analysis because the analysis algorithm must provide at least (nearly) as accurate results as the experienced analyst. Otherwise, the gain in analysis time is quickly consumed by the need for audits and error corrections.

These challenges have resulted in the algorithms presented in this thesis being quite elaborate and complex. General signal processing methods used in the applications are presented in Part I, *Methods*, and consists of Chapter 2-5.

The applications make up the main contribution of this thesis and span over continuous-time and discrete-time modeling, prefiltering through Wiener filter design, matched filter detection, Kalman filtering and prediction, and signal classification/discrimination using multiple target tracking. This contribution is found in Part II, *Applications*, where basically three tasks are addressed. Below, we refer to these as *modeling*, *prefiltering and parameter assessment*, and *data detection and classification*.

10.1 Modeling

In Chapter 6 and Chapter 7, models for simulating the acquisition of the EMG signal are presented. In Chapter 6 the single muscle fiber's contribution, the action potential (AP), to the EMG signal was modelled.

The shape and amplitude of the AP depend on obvious factors such as the position of the electrode in relation to the muscle fiber and the physical dimensions of the electrode. From a diagnostic perspective, factors such as muscle fiber diameter and conduction velocity are most important. Further parameters involve the actual physical characteristics of the fiber and the volume conductor within which the muscle fiber is residing. Examples of these are the length of the muscle fiber, the conductance of the fiber as well as of the medium, and the disturbances caused by the presence of the recording electrode.

In the presented model, the volume conductor was assumed to be infinite and having an anisotropic conductance. Based on a model of the intracellular potential, a transmembrane current lumped to the center of the fiber was derived.

The transmembrane current, assumed traveling along the axis of the muscle fiber, produces an electrical field within the volume conductor. Depending on the selected recording electrode, the electrical field contributes to the resulting AP in a certain way as described by the so-called weighting function.

This type of model is denoted as a *line source model* because it is formulated as a convolution between the transmembrane current and the weighting function. As shown in this thesis, a finite fiber length may be accomodated already in the continuous-time model by a simple adjustment of the weighting function. This has not been addressed in the literature before and is an important improvement to the efficiency of the simulations.

Another novelty introduced in this thesis is to apply an anti-aliasing filter to the continuous-time model prior to discretization. This reduces the aliasing errors and allows the use of a lower discretization frequency. This is an important contribution to the improvement of the efficiency of simulation models.

In addition to the above, there are some aspects and shortcomings worth mentioning. The assumption of an infinite volume conductor is of course questionable but, unless the muscle fiber is located close to the volume conductor's periphery, the error is negligible [39].

The validity of the intracellular potential model used here is uncertain. It is based on a single work [70] and its accuracy is not fully known. Several variants of this potential have been used, e.g., [62] and [39], suggesting there are some imperfections. This is an issue that would benefit from a further study.

No anti-aliasing filter was applied to the weighting function prior to its discretization due to the lack of the required transfer function formulation. This place a lower limit on the admissible discretization frequencies. Certainly, the weighting function could first be discretized using a high frequency, anti-aliasing filtered, and re-discretized at the desired frequency. The practical advantage of such a procedure is probably limited. The most significant advantage of a transfer function formulation of the weighting function is that, combined with the transfer function formulation of the transmembrane current, the APs may be generated in the frequency domain directly. Therefore, the transformation from the time domain to the frequency domain may be omitted.

The weighting functions of the electrodes were derived by averaging the electric field over the recording surface. The mere presence of the electrode, however, alters the electric field which is manifested as a *wall effect* [27] in recordings with a single fiber positioned close to a muscle fiber and as a *shadow effect* in recordings from a fiber situated "behind" the cannula [88].

Neither of these effects were incorporated in the model due to the lack of parametric descriptions of the corresponding weighting functions. In the general case, however, these effects have no important contribution to the resulting AP. The model validation aspects is further discussed below.

In Chapter 7 a simple model of a motor unit (MU) is presented. Based on the correspondance between the mean fiber concentration (MFC) on one hand and the continuum of myopathic-normal-neurogenic muscles on the other, the MFC value alone was used to control the generation of normal, myopathic, and neurogenic MUs. Albeit this is a principally correct model that has support in earlier studies [84], it disregards many of the important processes known to take place in neuromuscular diseases. The approximation was needed, however, in order to make the simulations in Chapter 8 manageable.

An important issue is the general validity of the model (assuming an appropriate simulation of the MU, see [83]). Performing a proper model validation would be difficult. All available experience, however, indicate that the accuracy is good enough for most situations. Compared to the variability encountered in live recordings, these errors are in general negligible.

The perhaps most important advantage with this model is that it is computationally fast and well suited for simulations of large number of fibers (even entire muscles as shown in [83] and [84]). Such simulations may contribute to a deeper understanding of the correspondance between the EMG signal and the different neuromuscular disorders.

10.2 Prefiltering and parameter assessment

Chapter 8 explores the possibility of applying a prefilter to EMG signals obtained with the concentric needle electrode in order to simplify the succeeding parameter assessment. The assessed parameters are intended to be used for diagnosis.

It was assumed that the EMG signal is properly recorded and that its constituent motor unit action potentials (MUAPs) are identified and extracted. This makes it possible to separately analyze the individual MUAPs originating from the same MU.

The assessment was focused on the MFC and the jitter, because these parameters discriminate a vast number of diseases. To simplify their assessment, the MUAP was prefiltered in different ways.

Estimating the MFC is basically an estimation of the number of fibers residing within a certain area. The number of fibers is reflected in the MUAP by the amplitude, the area, or the number of peaks.

The original MUAP is not very suitable for this analysis due to the constituent APs having both positive and negative phases. If a positive phase coincides with a negative phase, *cancellation* occurs where the total amplitude is decreased.

By designing the prefilter appropriately, the originally multiphasic APs may be transformed to a strictly positive signal with a single maximum, referred to as a *monophasic* signal.

Applying this prefilter to the MUAP causes all APs to add constructively, with no cancellation. For the amplitude, the prefiltering clearly provided an improvement of the MFC estimation whereas for the area, the difference was minimal.

The results suggest that the area is the preferred method when assessing the MFC. In clinical routine, however, the amplitude measure may still be preferred because it is sensitive to changes in the time dispersion of the APs. (This aspect has not been discussed in the thesis.) Prefiltering of the MUAP may therefore be advantageous. Further studies will analyze this issue in more detail.

To count the individual fibers, the prefiltered APs must be narrow in order to be discernable. A second filter was designed that transformed the recorded APs into narrow impulses (also monophasic). By applying this second prefilter to the MUAP, the APs are manifested as narrow impulses in the filter output. The performance of this method was good for MFC values below normal (MFC < 5 fibers/mm^2) but saturated for higher fiber concentrations.

A practical and efficient way of designing the prefilters is to use Wiener filter design methods, c. f. Chapter 5. This framework provides a good intuitive coupling between the design variables and the properties of the resulting filter, which simplifies the tuning process.

The second prefilter may also be used in jitter estimations because it often resolves the individual peaks. In addition to merely detect the peak, it must then also be localized very accurately (within a few microseconds). By fitting a second order polynomial to the found peak, a refined localization is possible to achieve.

In contrast to the fiber counting algorithm, however, this localization procedure is quite sensitive to noise. It works well on, e.g., single fiber APs where the signal-to-noise ratio is large but yields poor estimates on the noisy output from the second prefilter.

To alleviate this effect, the bandwidth of the prefilter was reduced to 50%. The drawback of this is that the width of the impulses increases which impair the resolution capacity. In future work the noise requirements will be addressed already at the outset of the filter design.

The width of the constituent components has another, more subtle, importance whether they are the prefiltered APs or the original ones. If two components are closely spaced in time, the peaks in the summation signal are in general closer to each other than the individual peaks of the two components. We refer to this as *peak displacement* and it may cause an erroneous jitter estimate.

As was shown in the chapter, the narrow impulses produced by the second, adjusted prefilter is a benefit in this regard. Using the APs obtained with the single fiber electrode, the jitter estimates were slightly biased ($\pm 2\ \mu$s). Using the prefiltered MUAPs, the bias was comparable for APs spaced less than 300 μs and practically zero for larger separations. Further studies will be made to investigate the validity and implications of this result.

Once the AP components are detected and localized, the jitter may be estimated from the localization data. In the general case, where the absolute timing of the APs is unknown, the *relative* jitter is measured by studying the variability in the inter-peak intervals (IPIs).

Unless only one pair of the APs are detected in each discharge, the detected APs must be classified in order to assure that the IPIs are calculated on the same pairs. As a “proof of concept,” a test was done using the MHT/Kalman tracking algorithm, c. f. Chapter 3 and Chapter 4, which correctly classified the APs in a MUAP with four detectable APs.

An interesting possibility with this tracking algorithm is that an automatic exclusion of unresolvable APs may be built-in. This is a challenging issue for future work.

Most importantly, the proposed methods must be tested on both properly modeled abnormal muscles as well as on real signals. We anticipate that the prefilters will then need to be adjusted due to differences between the model and real muscle. What impact this has on the performance remains to be seen, but a performance similar to what is shown in the thesis is reasonable.

10.3 Data detection and classification

In Chapter 9 an application that automates the detection and classification of nerve fiber APs is described.

The data was recorded by an automatic system that exploits the *marking phenomenon* [95] by repetitively stimulating a set of C-fibers in the foot while recording the emitted responses at the knee. The recorded APs are displayed in subsequent traces from top to bottom in a *falling leaf display*. Any additional stimulus that triggers a C-fiber is manifested by an increased latency.

The APs were detected asynchronously by a matched filter (MF), c. f. Chapter 2, that was tuned using previously recorded data and had a constant false alarm rate. The noise was assumed to be white and its variance was estimated prior to the analysis of each trace. The performance of this detector was adequate in the sense that it was not a limiting factor in the overall performance.

The detected APs were classified using the MHT/Kalman tracking algorithm, c. f. Chapter 3 and Chapter 4, where the differences in latency was used to discriminate between APs originating from different C-fibers. To further improve the performance, especially for crossing tracks, the latency model was augmented with an amplitude estimate.

Tuning the tracker and finding the appropriate parameter settings is somewhat tricky. The simulation methods presented in Chapter 2 and Chapter 3 provide some parameters, e.g., resolution and accuracy of the MF, and the variance of the predictions to mention a few. Other parameters had to be found using trial-and-error.

Once the appropriate parameter settings were found, however, there was no immediate need for retuning. The overall performance was good and provided AP classifications with little need for audit and correction.

The algorithm application has been in use for several years and has proven to make the analysis of these experiments more efficient.

However, the tracking performance depends on the manual tuning of the model parameter α. Although this parameter rarely needs to be changed, it would be preferable to design a predictor that is robust to differences in this parameter. A possible solution is to extend the Kalman predictor to a multiple model predictor.

10.4 Epilog

As demonstrated by the applications presented in this thesis, a signal processing approach to problems within the neurophysiological field may actually be fruitful in spite of the competition presented by the existing, often well-designed algorithms.

With better physiological and anatomical knowledge, more sophisticated recording techniques, and more powerful computers, the future of signal processing within the neurophysiological field look bright and promising.

BIBLIOGRAPHY

[1] A. D. Adrian and D. W. Bronk, "The discharge of impulses in motor nerve fibers. The frequency of discharges in reflex and voluntary contractions," *J. Physiol. (Lond.)*, vol. 67, pp. 119–151, 1929.

[2] A. Ahlén and M. Sternad, "Wiener filter design using polynomial equations," *IEEE Trans. Signal Process.*, vol. 39, no. 11, pp. 2387–2399, November 1991.

[3] A. Ahlén and M. Sternad, "Derivation and design of Wiener filters using polynomial equations," in C. T. Leondes, editor, *Stochastic Techniques in Digital Signal Processing*, pp. 353–418, Academic Press, New York, NY, 1994.

[4] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.

[5] S. Andreassen and A. Rosenfalck, "Relationship of intracellular and extracellular action potential of skeletal muscle fibre," *CRC Crit. Rev. Bioeng.*, vol. 6, pp. 267–306, November 1981.

[6] C. F. Ansley and R. Kohn, "Estimation, filtering, and smoothing in state space models with incompletely specified initial conditions," *Ann. Statist.*, vol. 13, no. 4, pp. 1286–1316, 1985.

[7] S.-M. Aquilonius, H. Askmark, P.-G. Gillberg, S. D. Nandedkar, Y. Olsson, and E. Stålberg, "Topographic localization of motor endplates in cryosections of whole human muscles," *Muscle & Nerve*, vol. 7, pp. 287–293, May 1984.

[8] K. J. Åström, "Stochastic state models," in *Introduction to Stochastic Control Theory*, pp. 44–90, Academic Press, New York, NY, 1970.

[9] K. J. Åström and B. Wittenmark, "Discrete-time systems," in *Computer Controlled Systems, Theory and Design*, pp. 30–76, Prentice Hall, Upper Saddle River, NJ, third edition, 1997.

[10] Y. Bar-Shalom, "Extension of the probabilistic data association filter in multi-target tracking," in *Proc. 5th Symp. on Nonlinear Estimation*, September 1974.

[11] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. New York, NY: Academic Press, 1988.

[12] Y. Bar-Shalom and X.-R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Norwood, MA: Artech House, 1993.

[13] S. S. Blackman, *Multiple-Target Tracking with Radar Applications*. Dedham, MA: Artech House, 1986.

[14] S. S. Blackman, R. J. Dempster, and T. S. Nichols, "Application of multiple hypothesis tracking to multi-radar air defense systems," in D. Liang et al., editors, *Multi-Sensor Multi-Target Data Fusion, Tracking and Identification Techniques for Guidance and Control Applications*, pp. 96–120, NATO AGARD-AG-337, October 1996.

[15] S. S. Blackman, R. J. Dempster, and S. H. Roszkowski, "IMM/MHT applications to radar and IR multitarget tracking," in *Proc. of SPIE, Signal and Data Processing of Small Targets 1997*, vol. 3163, Orlando, FL, October 1997, pp. 429–439.

[16] S. S. Blackman and Robert Popoli, *Design and Analysis of Modern Tracking Systems*. Dedham, MA: Artech House, 1999.

[17] D. C. Boyd, P. D. Lawrence, and P. J. A. Bratty, "On modeling the single motor unit action potential," *IEEE Trans. Biomed. Eng.*, vol. BME-25, pp. 236–243, 1978.

[18] J. H. Breasted, "The Edwin Smith surgical papyrus," *J. Neurosurg.*, vol. 21, pp. 240–244, 1964, Originally published in The Edwin Smith Surgical Papyrus, 2 volumes. The University of Chicago Press, Chicago, 1930.

[19] J. A. Cadzow, "Matched filters," in *Foundations of Digital Signal Processing and Data Analysis*, pp. 442–461, Macmillan, New York, NY, 1987.

[20] J. Clark and R. Plonsey, "The extracellular potential field of the single active nerve fiber in a volume conductor," *Biophys. J.*, vol. 8, pp. 842–864, 1968.

[21] M. de Feo, A. Graziano, R. Miglioli, and A. Farina, "IMMJPDA versus MHT and Kalman filter with NN correlation: performance comparison," *IEE Proc.-Radar, Sonar Navig.*, vol. 144, no. 2, pp. 49–56, 1997.

[22] R. Lorente de Nó, "A study of nerve physiology," in *Analysis of the distribution of action currents of nerve in volume conductors*, vol. 132, chapter XVI, pp. 384–477, Studies from the Rockefeller Institute for Medical Research, 1947.

[23] J. E. Dennis, Jr. and D. J. Woods, *New Computing Environments: Microcomputers in Large-Scale Computing*, pp. 116–122, SIAM, 1987.

[24] S. Deutsch and A. Deutsch, *Understanding the Nervous System: An Engineering Perspective*. New York, NY: IEEE Press, 1993.

[25] V. Dubowitz and M. H. Brooke, *Muscle biopsy: a modern approach*. London: WB Saunders, 1973.

[26] J. Duchêne and J.-Y. Hogrel, "A model of EMG generation," *IEEE Trans. Biomed. Eng.*, vol. BME-47, no. 2, pp. 192–201, February 2000.

[27] J. Ekstedt, "Human single muscle fiber action potentials," *Acta Physiol. Scand.*, pp. 1–96, 1964.

[28] J. Ekstedt, P. Häggqvist, and E. Stålberg, "The construction of needle multi-electrodes for single fiber electromyography," *Electroenceph. Clin. Neurophysiol.*, vol. 27, pp. 540–543, 1969.

[29] J. Ekstedt, G. Nilsson, and E. Stålberg, "Calculation of the electromyographic jitter," *J. Neurol. Neurosurg. Psy.*, vol. 37, no. 5, pp. 526–539, 1974.

[30] J. Ekstedt and E. Stålberg, "How the size of the needle electrode leading-off surface influences the shape of the single muscle fibre action potential in electromyography," *Comp. Prog. Biomed.*, vol. 3, pp. 204–212, 1973.

[31] C. A. Elsberg, "The Edwin Smith surgical papyrus and the diagnosis and treatment of injuries to the skull and spine 5000 years ago," *Ann. med. Hint.*, vol. 8, pp. 271–279, 1981.

[32] M. Ertaş, M. B. Baslo, N. Yildiz, J. Yazici, and A. E. Öge, "Concentric needle electrode for neuromuscular jitter analysis," *Muscle & Nerve*, vol. 23, pp. 715–719, May 2000.

[33] B. Falck, E. Stålberg, S. Stålberg, and M. Åström, "Multi-MUP EMG analysis in clinical routine," *Neurologia i Neurochirurgia Polska*, vol. 30, no. 3, pp. 55–70, 1996.

[34] C. Forster and H. O. Handwerker, "Automatic classification and analysis of microneurographic spike data using a PC/AT," *J. Neurosci. Meth.*, vol. 31, pp. 109–118, 1990.

[35] C. Forster and M. Schmelz, "New developments in microneurography of human C fibers," *News in Physiological Sciences*, vol. 11, pp. 170–175, 1996.

[36] R. A. Gallant, "Univariate nonlinear regression," in *Nonlinear Statistical Models*, chapter 1, pp. 1–122, Wiley, New York, NY, 1987.

[37] T. H. Gootzen. *Muscle Fibre and Motor Unit Action Potentials: A biophysical basis for clinical electromyography*, PhD thesis, University of Nijmegen, Nijmegen, The Netherlands, 1990.

[38] T. H. Gootzen, D. F. Stegeman, and A. van Oosterom, "Finite limb dimensions and finite muscle length in a model for the generation of electromyographic signals," *Electroenceph. Clin. Neurophysiol.*, vol. 81, no. 2, pp. 152–162, April 1991.

[39] T. H. Gootzen, D. F. Stegeman, and A. van Oosterom, "Finite limb dimensions and finite muscle length in a model for the generation of electromyographic signals," *Ann. Biomed. Eng.*, vol. 21, no. 4, pp. 391–399, July 1993.

[40] P. A. M. Griep, K. L. Boon, and D. F. Stegeman, "A study of the motor unit action potential by means of computer simulation," *Biol. Cybern.*, vol. 30, pp. 221–230, 1978.

[41] P. A. M. Griep et al., "Calculation and registration of the same motor unit action potential," *Electroenceph. Clin. Neurophysiol.*, vol. 53, pp. 388–404, 1982.

[42] A. Gydikov and N. Trayanova, "Extracellular potentials of single active muscle fibers : effects of finite fiber length," *Biol. Cybern.*, vol. 53, pp. 363–372, 1986.

[43] R. G. Hallin and H. E. Torebjörk, "Afferent and efferent C units recorded from human skin nerves in situ," *Acta Soc. Med. Ups.*, vol. 75, pp. 277–281, 1970.

[44] B. Hammarberg (Hansson), C. Forster, and E. Torebjörk, "Parameter estimation of human nerve C-fibers using matched filtering and multiple hypothesis tracking," *IEEE Trans. Biomed. Eng.*, April 2002, Accepted for publication.

[45] T. L. Harman, T. F. Liebfried, J. W. Clark, and C. W. Hibbs, "A comparison of two methods for determining the extracellular potential field of an isolated Purkinje strand in a volume conductor," *IEEE Trans. Biomed. Eng.*, vol. BME-22, pp. 174–183, 1972.

[46] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, third edition, 1996.

[47] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol. (Lond.)*, vol. 117, pp. 500–544, 1952.

[48] L. B. Jackson, *Digital Filters and Signal Processing*. Norwell, MA: Kluwer Academic Publishers, second edition, 1989.

[49] J. Ji and K. D. Wise, "An implantable CMOS circuit interface for multiplexed microelectrode recording arrays," *IEEE J. of Solid-State Circuits*, vol. 27, no. 3, pp. 433–443, March 1992.

[50] T. Kailath, *Linear Systems*. Englewood Cliffs, NJ: Prentice Hall, 1980.

[51] T. Kailath, *Lectures on Wiener and Kalman Filtering*. Wien: Springer, 1981.

[52] R. E. Kalman, "A new approach to linear filtering and prediction problems," *ASME Trans. – J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, March 1960.

[53] L. Karlsson, B. Hammarberg, and E. Stålberg, "A muscle model to study electromyographic signals," *Comp. Meth. Prog. Biomed.*, 2002, submitted.

[54] R. Kohn and C. F. Ansley, "Filtering and smoothing algorithms for state space models," *Computers Math. Applic.*, vol. 18, no. 6/7, pp. 515–528, 1989.

[55] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[56] D. Lerro and Y. Bar-Shalom, "Interacting multiple model tracking with target amplitude feature," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 2, pp. 494–508, 1993.

[57] K. Lundin. "A system for analysis of human pain signals using a radar tracking approach," Master's thesis, UPTEC F 98 078, Uppsala University, Uppsala, Sweden, October 1998.

[58] K. C. McGill, K. L. Cummins, and L. J. Dorfman, "Automatic decomposition of the clinical electromyogram," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 7, pp. 470–477, 1985.

[59] S. D. Nandedkar, D. B. Sanders, and E. Stålberg, "Automatic analysis of the electromyographic interference pattern. Part I: Development of quantitative features," *Muscle & Nerve*, vol. 9, no. 5, pp. 431–439, 1986.

[60] S. D. Nandedkar, D. B. Sanders, E. V. Stålberg, and S. Andreassen, "Simulation of concentric needle EMG motor unit action potentials," *Muscle & Nerve*, vol. 11, pp. 151–159, February 1988.

[61] S. D. Nandedkar and E. V. Stålberg, "Simulation of macro EMG motor unit potentials," *Electroenceph. Clin. Neurophysiol.*, vol. 56, pp. 52–62, February 1983.

[62] S. D. Nandedkar and E. V. Stålberg, "Simulation of single muscle fibre action potentials," *Med. Biol. Eng. Comput.*, vol. 21, pp. 158–165, March 1983.

[63] S. D. Nandedkar, E. V. Stålberg, and D. B. Sanders, "Simulation techniques in electromyography," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 10, pp. 775–785, October 1985.

[64] K. Öhrn. *Design of Multivariable Cautious Discrete-Time Wiener Filters: A Probabilistic Approach*, PhD thesis, Uppsala University, Uppsala, Sweden, 1996.

[65] J. Payan, "The blanket principle: a technical note," *Muscle & Nerve*, vol. 1, no. 5, pp. 423–426, 1978.

[66] R. Plonsey, "Dependence of scalar potential measurements on electrode geometry," *Rev. Sci. Instrum.*, vol. 36, 1965.

[67] D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, and J. O. McNamara, *Neuroscience*. Sunderland, MA: Sinauer Associates, 1997.

[68] L. Råde and B. Westergren, *BETA: Mathematics Handbook*. Lund, Sweden: Studentlitteratur, 1989.

[69] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Contr.*, vol. 24, no. 6, pp. 843–854, December 1979.

[70] P. Rosenfalck, "Intra and extracellular potential fields of active nerve and muscle fibers. A physio-mathematical analysis of different models," *Acta Physiol. Scand.*, vol. 321, pp. 1–168, 1969.

[71] A. Sandberg, B. Hansson, and E. Stålberg, "Comparison between concentric needle EMG and macro EMG in patients with a history of polio," *Clin. Neurophysiol.*, vol. 110, no. 11, pp. 1900–1908, November 1999.

[72] L. L. Scharf, "Neyman-pearson detectors," in *Statistical Signal Processing – Detection, Estimation, and Time Series Analysis*, chapter 4, pp. 103–178, Addison-Wesley Publishing Company, Inc., Reading, MA, 1991.

[73] M. Schmelz, C. Forster, R. Schmidt, M. Ringkamp, H. O. Handwerker, and H. E. Torebjörk, "Delayed responses to electrical stimuli reflect C-fiber responsiveness in human microneurography," *Exp. Brain Res.*, vol. 104, pp. 331–336, 1995.

[74] M. Schmelz, R. Schmidt, M. Ringkamp, C. Forster, H. O. Handwerker, and H. E. Torebjörk, "Limitation of sensitization to injured parts of receptive fields in human skin C-nociceptors," *Exp. Brain Res.*, vol. 109, pp. 141–147, 1996.

[75] R. Schmidt, M. Schmelz, C. Forster, M. Ringkamp, H. E. Torebjörk, and H. O. Handwerker, "Novel classes of responsive and unresponsive C nociceptors in human skin," *J. Neurosci.*, vol. 15, pp. 333–341, 1995.

[76] T. Söderström, *Discrete-time stochastic systems: Estimation & Control*. Hertfordshire, UK: Prentice Hall, 1994.

[77] H. W. Sorenson, "Least-squares estimation: from Gauss to Kalman," *IEEE Spectrum*, vol. 7, no. 7, pp. 63–68, July 1970.

[78] E. Stålberg, "Propagation velocity in human muscle fibre in situ (thesis)," *Acta Physiol. Scand.*, vol. 70, no. suppl. 287, pp. 1–112, 1966.

[79] E. Stålberg, "Macro EMG, a new recording technique," *J. Neurol. Neurosurg. Psy.*, vol. 43, pp. 475–482, 1980.

[80] E. Stålberg, "Single fiber EMG, macro EMG, and scanning EMG. New ways of looking at the motor unit," *CRC Crit. Rev. Clin. Neurobiol.*, pp. 125–167, 1986.

[81] E. Stålberg and P. Dioszeghy, "Scanning EMG in normal muscle and in neuromuscular disorders," *Electroenceph. Clin. Neurophysiol.*, vol. 81, pp. 403–416, 1991.

[82] E. Stålberg and B. Falck, "The role of electromyography in neurology," *Electroenceph. Clin. Neurophysiol.*, vol. 103, pp. 579–598, 1997.

[83] E. Stålberg and L. Karlsson, "Simulation of the normal concentric needle electromyogram by using a muscle model," *Clin. Neurophysiol.*, vol. 112, no. 3, pp. 464–471, March 2001.

[84] E. Stålberg and L. Karlsson, "Simulation of EMG in pathological situations," *Clin. Neurophysiol.*, vol. 112, no. 5, pp. 869–878, May 2001.

[85] E. Stålberg, S. D. Nandedkar, D. B. Sanders, and B. Falck, "Quantitative motor unit potential analysis," *J. Clin. Neurophysiol.*, vol. 13, pp. 401–422, 1996.

[86] E. Stålberg and J. V. Trontelj, *Single Fiber Electromyography in Healthy and Diseased Muscle*. New York, NY: Raven Press, second edition, 1994.

[87] D. W. Stashuk, "Detecting single fiber contributions to motor unit action potentials," *Muscle & Nerve*, vol. 22, pp. 218–229, February 1999.

[88] D. Stegeman, T. H. J. M. Gootzen, and M. M. H. J. Theeuwen, "Intramuscular potential changes caused by the presence of the recording EMG needle electrode," *Electroenceph. Clin. Neurophysiol.*, pp. 81–90, 1993.

[89] A. Steinhardt, "Adaptive multisensor detection and estimation," in *Adaptive Radar Detection and Estimation*, pp. 91–160, Wiley, New York, NY, 1992.

[90] M. Sternad and A. Ahlén, "Robust filtering and feedforward control based on probabilistic descriptions of model errors," *Automatica*, vol. 29, no. 3, pp. 661–679, 1993.

[91] M. Sternad, K. Öhrn, and A. Ahlén, "Robust $\mathcal{H}_2$ filtering for structured uncertainty: the performance of probabilistic and minimax schemes," in *Proc. of 3rd European Control Conf.*, vol. 1, Rome, Italy, 1995, pp. 87–92.

[92] M. M. H. J. Theeuwen, T. H. J. M. Gootzen, and D. F. Stegeman, "Muscle electric activity I : A model study on the effect of needle electrodes on single fiber action potentials," *Ann. Biomed. Eng.*, vol. 21, no. 4, pp. 377–389, 1993.

[93] H. E. Torebjörk, "Afferent C units responding to mechanical, thermal and chemical stimuli in human non-glabrous skin," *Acta Physiol. Scand.*, vol. 92, pp. 374–390, 1974.

[94] H. E. Torebjörk and R. G. Hallin, "C-fibre units recorded from human sensory nerve fascicles in situ," *Acta Soc. Med. Ups.*, vol. 75, pp. 81–84, 1970.

[95] H. E. Torebjörk and R. G. Hallin, "Responses in human A and C fibres to repeated electrical intradermal stimulation," *J. Neurol. Neurosurg. Psy.*, vol. 37, pp. 653–664, 1974.

[96] M. H. Verhaegen and P. van Dooren, "Numerical aspects of different Kalman filter implementations," *IEEE Trans. Automat. Contr.*, vol. 31, pp. 907–917, 1986.

[97] C. Weidner, M. Schmelz, B. Hansson, H. O. Handwerker, and H. E. Torebjörk, "Functional attributes discriminating mechano-insensitive and mechano-responsive C nociceptors in human skin," *J. Neurosci.*, vol. 19, pp. 10184–10190, November 1999.

[98] J. R. Werthman, "Step-by-step description of a computationally efficient version of multiple hypothesis tracking," in *Proc. of SPIE, Signal and Data Processing of Small Targets 1992*, vol. 1698, August 1992.

[99] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series*. New York: The Technology Press and Wiley, 1950.

[100] T. H. Yoon, E. J. Hwang, D. Y. Shin, S. I. Park, S. J. Oh, S. C. Jung, H. C. Shin, and S. J. Kim, "A micromachined silicon depth probe for multichannel neural recording," *IEEE Trans. Biomed. Eng.*, vol. BME-47, no. 8, pp. 1082–1087, August 2000.

INDEX