On-line decomposition of iEMG signals using GPU-implemented Bayesian filtering

Tianyi Yu

UNIVERSITE BRETAGNE LOIRE MATHSTIC

CENTRALE NANTES

THESE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : ***Génie informatique, automatique et traitement du signal, section CNU61***

Par
Tianyi YU

Décomposition en temps réel de signaux iEMG: filtrage bayésien implémenté sur GPU

Thèse présentée et soutenue à Nantes, le 28 janvier 2019
Unité de recherche : Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

M. Philippe RAVIER Maître de conférence, HDR, Université d'Orléans
M. Fabien CAMPILLO Directeur de recherche, INRIA de Montpellier

Composition du Jury :

Président :	Mme Zohra CHERFI-BOULANGER	Professeur, Université de technologie de Compiègne
Examinateurs :	M. Dario FARINA	Professeur, Imperial College London
	M. Yannick AOUSTIN	Professeur, Université de Nantes
	M. Eric LE CARPENTIER	Maître de conférence, Ecole Centrale de Nantes
Dir. de thèse :	M. Yannick AOUSTIN	Professeur, Université de Nantes
Co-dir. de thèse :	M. Eric LE CARPENTIER	Maître de conférence, Ecole Centrale de Nantes

Acknowledgment

Firstly, I would like to express my sincere gratitude to my advisors Mr. Yannick Aoustin and Mr. Eric Le Carpentier for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. They did not only give me advises for my research and study, but also explained me the difference cultures between Chine and France and helped me in my life.

Besides my advisors, I am grateful to the rest of my thesis committee: M. Philippe Ravier, M. Fabien Campillo, Mme Zohra Cherfi-Boulanger and M. Dario Farina, for their insightful comments and encouragement, but also for the questions which inspired me to widen my research from various perspectives. It was fantastic to have the opportunity to communicate with you.

Finally, I would like to thank my parents and all my friends who have worked with me, helped me and encouraged me in my Ph.D. Thanks all of you!

Contents

1 Introduction **13**
- 1.1 Project . . . 13
- 1.2 Plan of thesis . . . 15
- 1.3 Table of notations . . . 15

2 Research Background **17**
- 2.1 Introduction . . . 17
- 2.2 Electromyographic signals . . . 17
  - 2.2.1 Physiological model of EMG signal . . . 17
  - 2.2.2 Anatomy of motor unit . . . 20
- 2.3 EMG signals decomposition . . . 20
  - 2.3.1 EMG signals acquisition . . . 21
  - 2.3.2 iEMG signals decomposition . . . 23
  - 2.3.3 sEMG signals decomposition . . . 28
- 2.4 Parallel computation with graphics processing unit . . . 31
- 2.5 Discussion and conclusion . . . 34

3 Hidden Markov model **35**
- 3.1 Introduction . . . 35
- 3.2 Modelling of EMG . . . 35
- 3.3 State vector . . . 37
- 3.4 Transition model . . . 38
  - 3.4.1 Recruitment model . . . 38
  - 3.4.2 Renewal model for active spike trains . . . 39
- 3.5 Observation model . . . 40
- 3.6 Discussion and conclusion . . . 41

4 Bayes filter **43**
- 4.1 Introduction . . . 43
- 4.2 Principle . . . 43
- 4.3 Estimation of inter-spike law parameters . . . 44
- 4.4 Estimation of impulse responses . . . 46
  - 4.4.1 Kalman filter . . . 46
  - 4.4.2 Least mean square filter . . . 46
  - 4.4.3 Normalized least mean square filter . . . 47
- 4.5 Posterior probability of scenario . . . 50
- 4.6 Tracking . . . 51
- 4.7 Bayes estimator . . . 51
- 4.8 Initialisation . . . 52
- 4.9 Algorithm . . . 52
- 4.10 Discussion and conclusion . . . 52

5 Acceleration of decomposition **55**
5.1 Introduction . . . 55
5.2 Path pruning . . . 55
5.2.1 Limiting the number of kept paths . . . 55
5.2.2 Pruning based on activity detection . . . 56
5.2.3 Simultaneous spikes interdiction . . . 57
5.3 Parallelism analysis . . . 57
5.3.1 Data parallelism . . . 58
5.3.2 Task parallelism . . . 59
5.4 Task analysis . . . 60
5.4.1 Parallel sorting . . . 60
5.4.2 Indexes of bifurcation . . . 63
5.5 Parallel structure . . . 63
5.6 Discussion and conclusion . . . 65

6 Signals and preprocessing **67**
6.1 Introduction . . . 67
6.2 Signal preprocessing . . . 67
6.2.1 Signal pre-filtering . . . 67
6.2.2 MUAPs clipping . . . 68
6.3 Experimental and simulation protocols . . . 69
6.3.1 Signals . . . 69
6.3.2 Indexes of performance and complexity . . . 70
6.4 Discussion and conclusion . . . 70

7 Results **71**
7.1 Introduction . . . 71
7.2 Performance . . . 71
7.2.1 Simulated signals . . . 71
7.2.2 Experimental signals . . . 74
7.3 Decomposition velocity . . . 80
7.3.1 Experimental signals . . . 80
7.3.2 Simulated signals . . . 83
7.4 Discussion and conclusion . . . 84

8 Conclusion and perspectives **85**
8.1 Conclusion . . . 85
8.2 Perspectives . . . 86

9 Appendix **87**
9.1 Proof of the inter-spike law parameters estimation . . . 87
9.2 From Kalman filter to the least-mean-square filter . . . 88
9.3 Proof of normalised least-mean-square filter . . . 90
9.3.1 The first case: no MU firing . . . 90
9.3.2 The second case: there is a MU firing . . . 90
9.3.3 The third case: there are more than one MU firing . . . 92

List of Tables

1.1 Main notations 16

2.1 Morphological features of MUAP shapes 26

7.1 Decomposition performance for the simulated signal with 10 MUs 73
7.2 Decomposition performance for experimental signals: 'TA' and 'ADM' respectively represent signals from the tibialis anterior and abductor digiti minimi; 'Position' represents the level of abduction scaled equiangularly from the full adduction (0, not included) to the full abduction (5); 'Nb MUs' is the maximal number of MUs concurrently active in the signal; 'NB spikes' represents the overall number of spikes in the signal; and 'Sup.' is the percentage of superposition; 'Sens.' and 'Pred.' represent respectively the global sensitivity and predictivity as estimated by comparison with the manual expert decomposition. 74
7.3 Decomposition performance for an experimental signal detected from the TA with 8 MUs . 77
7.4 Decomposition performance for an experimental signal (see Figure 7.10) from ADM set . 78
7.5 Decomposition time of experimental signals recorded from muscle TA 80
7.6 Decomposition time of experimental signals recorded from muscle ADM: '10 kHz' and '5 kHz' indicates the sampling frequency. 82
7.7 Delay of experimental signals recorded from muscle TA: the signal index corresponds to the signals presented in table 7.5 83
7.8 Decomposition performance of simulated signals 83

List of Figures

1.1 Robotic hand of the laboratory LS2N [1] 14

2.1 Central nervous system and the functions of spinal nerves [2] 18

2.2 Anatomical characteristics fo MU: MNs that innervate the individual muscle are termed motor nucleus or motor neuron pool and cluster in either the anterior (ventral) horn of the spinal cord. Every MN innervates several muscle fibers. The muscle fibers of a muscle unit intermingle with other muscle units, but do not extend to other muscles nearby. (Copyright©2001 Benjamin Cummings, an imprint of Addison Wesley Longman, Inc.) . . 19

2.3 High-density sEMG recordings from a human hand muscle [34]. (A)sEMG signals recorded with an electrode array (5 columns $\times$ 13 rows) placed over the abductor pollicis brevis muscle of a healthy man as he sustained an isometric contraction at 10% of the maximal voluntary contraction force. (B) Distribution of multichannel sEMG amplitude at the instant indicated by the red dots and vertical lines in A. (C) Eight MUAPs detected by the array electrodes. 21

2.4 Different types of needle electrodes [40]:(a)Single fiber electrode (b) Concentric electrode (c) Monopolar electrode (d) Macro electrode 22

2.5 Example of iEMG signal was acquired using 25G wire electrodes (A-M Systems, Carlsborg, WA, USA) made of Teflon coated stainless steel with a diameter of 0.05 mm. The numbers denote indexes of MU. 23

2.6 iEMG signal decomposition[46, 47]: α MNs discharge successively in the spinal cord, then sending APs to related muscle fibers to formulate the MUAP trains. The summation of MUAP trans, named 'Raw EMG signal' in the figure, is detected by the needle electrode or wire electrode. The EMG signal is decomposed in the form of several individual MUAP trains corresponding to each active MU discharge. 24

2.7 Several similar segments of iEMG signal and the signals filtered by the 1st order and the 2nd order low-pass differentiating filter [53]. Differences between MUAPs were highlighted. 25

2.8 The multi-scale model of movement generation [86]. From left to right: P activation primitives are shared by N motor neuron pools $(P < N)$. Each MN pool receives the linear combination of the P activation primitives as input and transforms this input into spike trains that drive the innervated muscle (thus, the scheme presents N muscles). The N muscles contribute to $M(M > N)$ EMG channels by the trains of MUAPs. 29

2.9 A: The quadrifilar needle sensor with larger dimensions containing the 4 pins that detect the sEMG signals. B: Differential combinations that produce 4 channels of sEMG signals. . . 31

2.10 A: The five-pin surface EMG sensor attached above the First Dorsal Interosseous muscle in the hand. B: Top and bottom views of the sensor. The four pins on the corner of a square are spaced 3.6 mm apart. 32

2.11 The development of language GPU [96]: In the early 2000s, we had to write the application in shading language. Latter, several academic and third-party languages that abstracted away the graphics appeared. Since 2007, following the step of Nvidia, all GPU manufactories launched eventually the GPUs programmable in common program language for the general-purposed computation. 32

3.1 The linear model of iEMG signal: The spike train, denoted by $U_i[n]$, is filtered by the corresponding impulse response h_i. We obtain the MUAP trains. The summation of the active MUAP trains is the iEMG signal, represented by $Y[n]$ in the figure. 36
3.2 Illustration of the relationship between the spike train $U[n]$ and corresponding sawtooth sequence $T[n]$; Illustration of MU deactivation/activation events. Time between subsequent spikes was shortened for illustration purposes; in reality, it comprises hundreds of time instants. Moreover, Δ represent the length of inter-spike interval and N is its index. 37
3.3 Example of the discrete Weibull distribution type I with different parameters 40
3.4 Hidden Markov model 41

4.1 Inter-spike law parameters estimation: The first line is the given spike train $U[n]$; The second line is the estimated location parameter sequence of the discrete Weibull distribution type I; The third line is the estimated concentration one; The forth line is the firing rate. The red horizontal straight line denotes the real value, while the black oscillating line represents the estimated value. 45
4.2 Interpretation of $F[n]$: Red and blue dashed lines represent respectively two MUAP waveforms. Part of them is superimposed (from $n = 10$ to $n = 21$). Black line is the simulated iEMG signal, which is the temporal summation of the two MUAP waveforms. If $n < 5$ or $n \geq 26$, $F[n]$ is a empty set; If $5 \leq n < 10$, $F[n] = \{1\}$; If $10 \leq n < 21$, $F[n] = \{1, 2\}$; If $21 \leq n < 26$, $F[n] = \{2\}$. 48
4.3 Comparison of MUAP shapes estimation using three different filters: Kalman filter, LMS filter and NLMS filter. SNR=20 dB (left); SNR=10 dB (right). Misalignment (in dB) is defined as $20 \log_{10}(\frac{\|H[n]-\hat{H}^{[n}_{S^n}\|_2}{\|H[n]\|_2})$. 49

5.1 Example of iEMG segmentation. Segments are detected using certain threshold and shifted in time to the left by l_{pd} due to the use of future samples. Bifurcations containing impulses are forbidden while $Z[n] = 0$. 56
5.2 Two close cases of MUAP superposition: (a) - exact superposition of two spikes, a case considered rare and thus excluded from the search; (b) - a close superposition case (Δt denotes the sampling period). 57
5.3 The CUDA concept of a grid of blocks [96]. Each block consists of a set of threads that can communicate and cooperate. Each thread uses its block index in combination with its thread index to identify its position in the global grid. 59
5.4 Concurrency example: There are five concurrency examples: 1. Serial execution. 2. Concurrency execution between a kernel function and the memory copy from device to host. 3. Concurrency execution between a kernel function and two types of memory copies (DH: from device to host and HD: from host to device). 4. Concurrency execution of a kernel function, two types of memory copies and CPU. 5. Concurrency execution of three kernel functions, two types of memory copies and CPU. 60
5.5 Two sets of operations of the shared-address-space parallel formulation of quick sort [119] 61
5.6 Bitonic sorting of a sequence with 16 elements [119] (a) The sequence is converted three times to a bitonic sequence; (b) A bitonic sequence is sorted to a monotonically increasing sequence. 62
5.7 Parallel structure of iEMG signal decomposition algorithm 64

6.1 Differentiation (a) The iEMG signal before differentiation; (b) The iEMG signal after differentiation. 68
6.2 MUAP waveforms clipping (a) MUAP waveforms before clipping; (b) MUAP waveforms after clipping 68
6.3 Acquisition of experimental signals from the abductor digiti minimi muscle 69

7.1 Comparison of automatic decomposition (crosses, 'x') and actual results (points, '.') in the upper panel and the simulated signal with 10 MUs in the lower panel. 72
7.2 An extraction of the simulated signal decomposition shown in figure 7.1; circles '$\circ$' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions. 72
7.3 Initial MUAP shapes (actual MUAP shapes contaminated by the noise) and their actual MUAP shapes for the signal presented in Figure 7.1 73
7.4 Normalised misalignment of estimated MUAP shapes 74
7.5 Firing rates for the simulated iEMG (see figure 7.1): the dashed lines (empirical) represent the actual firing rates; continuous lines (estimated) represent the firing rates calculated via the estimated parameters of discrete Weibull distribution as described in section 4.3. 75
7.6 Comparison of automatic (crosses, 'x') and reference (points, '.') decompositions (upper panel) and the experimental signal from TA, 30% MVC (lower panel). 76
7.7 An extract of the experimental signal decomposition shown in figure 7.6; circles '$\circ$' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions. 76
7.8 Eight MUAP shapes (manually-extracted dictionary) for the signal presented in Figure 7.6, and a comparison between the 2nd one and the 3rd one. 77
7.9 Firing rates for the iEMG from TA set (see figure 7.6): the dashed lines (empirical) represent the firing rates estimated using reference decomposition; continuous lines (estimated) represent the firing rates calculated via the estimated parameters of discrete Weibull distribution as described in section 4.3. 78
7.10 Comparison of automatic (crosses, 'x') and reference (points, '.') decompositions (upper panel) and corresponding experimental signal from ADM, in position 5, corresponding to approximately 45 degrees of abduction (lower panel). 79
7.11 An extract of the experimental signal decomposition shown in figure 7.10; circles '$\circ$' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions. 79
7.12 Initial MUAP shapes (manually-extracted dictionary) and their final estimations for the signal presented in Figure 7.10 80
7.13 Firing rates of the experimental signal with 7 MUs detected from ADM in the abduction position '5' 81
7.14 Delay of experimental signal (recorded from muscle TA, with 8MUs and 30% MVC) decomposition with 256 paths 82
7.15 Example of a simulated signal with 10 MUs: The recruitment profile is shown in the upper panel. Corresponding simulated signal with 10 dB SNR is depicted in the lower panel. . . 83

1

Introduction

1.1 Project

My work of phD thesis is developed in the group ReV (Robotique Et Vivant) and the group SIMS (Signal, Image et Son) of the laboratory LS2N (Laboratoire des Sciences du Numérique de Nantes). In our team, I am supervised by Mr. Yannick Aoustin and Mr. Eric Le Carpentier, specialists in the robotics and signal processing, researching in the domain of biomedical.

The final objective of our team research is to control precisely the active prosthetic devices for amputees with electromyographic (EMG) signals. The active prosthetic devices that we study is the robotic hand of the laboratory LS2N. It is shown in figure 1.1. The LS2N hand is an underactuated robotic hand, constructed by Alpes Instruments (Grenoble) for LS2N. This robotic hand has an underactuated mechanism, which allows to obtain 15 degree of freedoms (DOFs) by using only six actuators. The underactuation mechanism provides several advantages, such as: low weight, low cost of power, and easy control. When the robotic hand does not contact an object, its fingers move as a one DOF serial chain. When it grasps an object, a particular mechanical system allows its fingers to adapt to the shape of the object, acting similarly to a human hand.

Precise control of the active prosthetic devices for amputees with electromyographic (EMG) signals is a very huge and difficult task. The realisation of this work is divided in three steps:

— Extraction of critical information in the signals of muscle, EMG signals.
— Understanding and quantification of the correlation between muscle signals information and the kinematic coefficient of the movement
— Command of the robotic hand to produce a prediscribed movement.

Due to the time restriction of prosthetic control, the first step and the third step should be executed in a real time manner.

A sequential decomposition algorithm based on a Hidden Markov Model of the EMG, that used Bayesian filtering to estimate the unknown parameters of discharge series of motor units was previously proposed in the laboratory LS2N. This algorithm has successfully decomposed the experimental iEMG signal with four motor units. However, the proposed algorithm requires a high time consuming.

In the work of my phD, we firstly validated the proposed algorithm in a serial structure. We proposed some modifications for the activation process of the recruitment model in Hidden Markov Model and implemented two signal pre-processing techniques to improve the performance of the algorithm. Then, we realized a GPU-oriented implementation of this algorithm, as well as the modifications applied to the original model in order to achieve a real-time performance. Specifically, we proposed a replacement of the

Figure 1.1 – Robotic hand of the laboratory LS2N [1]

originally proposed Kalman filter by a least mean square filter with a significant reduction of computational load. Moreover, we introduced two heuristic-based techniques of branch discarding in order to simplify the problem of optimal spike sequence search. Then, an optimal parallelization of the algorithm is presented, along with details of its implementation on graphics processing unit (GPU). Now, We have achieved the decomposition of 10 experimental iEMG signals acquired from two different muscles, respectively by fine wire electrodes and needle electrodes.

During my phD, Mr. Dario Farina gave us some precious advices for our work and provided us several iEMG signals measured by fine wire electrodes. Moreover, Mr. Clement Huneau and his colleague also offered us some iEMG signals measured by needle electrode. These signals and the ones measured by ourselves were successfully decomposed. Our algorithm was validated on various signals from different laboratories, proving its stability, wide applicability and high efficiency.

1.2 Plan of thesis

The manuscript of thesis is organised as following:

— In the chapter 2, we would like to introduce the background of the research, including the basic anatomic knowledge of motor unit (MU), the physiological model to generate the electromyographic (EMG) signals, the EMG signals acquisition and decomposition. Moreover, the development of the parallel computation will also be presented.

— In the chapter 3, the process of derivation from the linear convolution model of EMG signal to the hidden Markov model (HMM) are illustrated.

— In the chapter 4, the Bayes filter is applied to decompose the iEMG signal. It jointly estimate the action potential of MU (MUAP), the firing statistics parameters and the spike trains.

— In the chapter 5, we implement the decomposition algorithm into the parallel computation in order to realise the real-time decomposition. Some heuristic measure are also given in this chapter.

— Chapters 6 and 7 present some methods to pre-process the signal, the simulated and experimental signals, and their results of decomposition evaluated in terms of the complexity, the performance and the speed.

1.3 Table of notations

The main notations used in the thesis are shown in table 1.1.

Table 1.1 – Main notations

Y	The iEMG signal
Ω	The set of indexes of all motor units
A	The set of indexes of active motor units
h	Impulse response
U	Spike trains
W	Noise
H	The vector of motor unit action potentials shapes
ℓ_{IR}	The maximum motor unit action potentials length
T	The sawtooth sequences
S	The activation scenario
Δ	The inter-spike interval
$\Theta = [t_0, \beta]$	The vector containing discrete Weibull distribution parameters: the location parameter and the concentration parameter
t_R	The shifting parameter of discrete Weibull distribution, that is the refractory period
t_{I}	The maximum active time without spike in the recruitment model
λ	The activation probability in the recruitment model
n_{path}	The number of kept scenarios (paths)
τ	The active time index in the inter-spike law parameters estimation
v_{S^n}	The variance of innovation in the impulse response estimation
V	The variance of noise
$\tilde{v}$	The ratio of the variance of innovation to the variance of noise
m_Δ	The expectation of inter-spike interval
Z	The segmentation sequence
Sup	The superposition percentage
$\mathsf{E}(.)$	Expectation
$\mathsf{Pr}(.)$	Probability
w.p.	with probability
$Y[n]$	The iEMG signal at time index n
Y^n	The vector containing the signal from time index 1 to n
$\|^n$	Given Y^n
$\mathsf{Pr}(T[n] = t[n])$	The probability of the sawtooth sequences at time index n being equal to a value $t[n]$. For all elements of the state vector, the uppercase symbols denote random variables, while the lowercase ones stand for their values.

2

Research Background

2.1 Introduction

Our research focuses on the on-line decomposition of EMG signals in the parallel computation. Therefore, in this chapter, we will introduce the background knowledge related to EMG signals decomposition and parallel computation. The first section shows a basic introduction of EMG signals containing the anatomical characteristics of MU which reveals the anatomical relations between motor neuron (MN) and the muscle fibers that it innervates, and the physiological model of EMG signals which describes the physiological procedure of generating EMG signals. In the second section, we present EMG signals decomposition, including the various protocols of EMG signals acquisition and several EMG decomposition methods corresponding to them proposed in recent years, in the both off-line and on-line manners. Finally, we will make a review of the development of the parallel computation in Graphics Processing Unit (GPU).

2.2 Electromyographic signals

The movement of humans is the kinematic manifestation of the muscle activities, while the EMG signal is the electrical expression of skeletal muscle fibers during a muscle contraction. They are controlled by the active motor unit (MU) populations, which are the smallest voluntary units in a movement. Each MU comprises two components: the motor neuron (MN) and the muscle fibers that its axon innervates, referred as to the muscle unit.

2.2.1 Physiological model of EMG signal

Human movement is controlled by the activity of cells in neural system, which is composed of the central (CNS) and peripheral (PNS) nervous system. The CNS consists of two major structures, the brain and the spinal cord, which is in charge of integrating information it receives, coordinating and influencing the activity of all parts of the bodies. The brain acting as the major processing unit of the nervous system receives sensory information and command body to make reactions. The spinal cord is not only the bridge between the brain and PNS, but also can accomplish some basic reflex.

The PNS consisting of the nerves and ganglia outside the brain and spinal cord, including the nerve roots, dorsal root ganglia, brachial and lumbosacral plexuses, and peripheral nerves, connects the CNS to other parts of body. Complex peripheral nerves are two-way conduits: Efferent motor information travels from the spinal cord to the muscles, while afferent sensory information travels from the periphery to the

spinal cord. Efferent motor signals travel from the anterior horn cells (α MNs), which are lower MNs under the control of the corticospinal tracts, into peripheral nerves by way of ventral roots, finally reaching the innervated muscle units. Figure 2.1 shows different positions of spinal cord controlling various type of muscles. Compared to the direct measure in the efferent peripheral nerve fibers, the muscle signals as a result of the neural informations spatially spreading from the PNS, have a relatively small number of physiological sources per unit volume and still reveals the same level information on the neural activities.

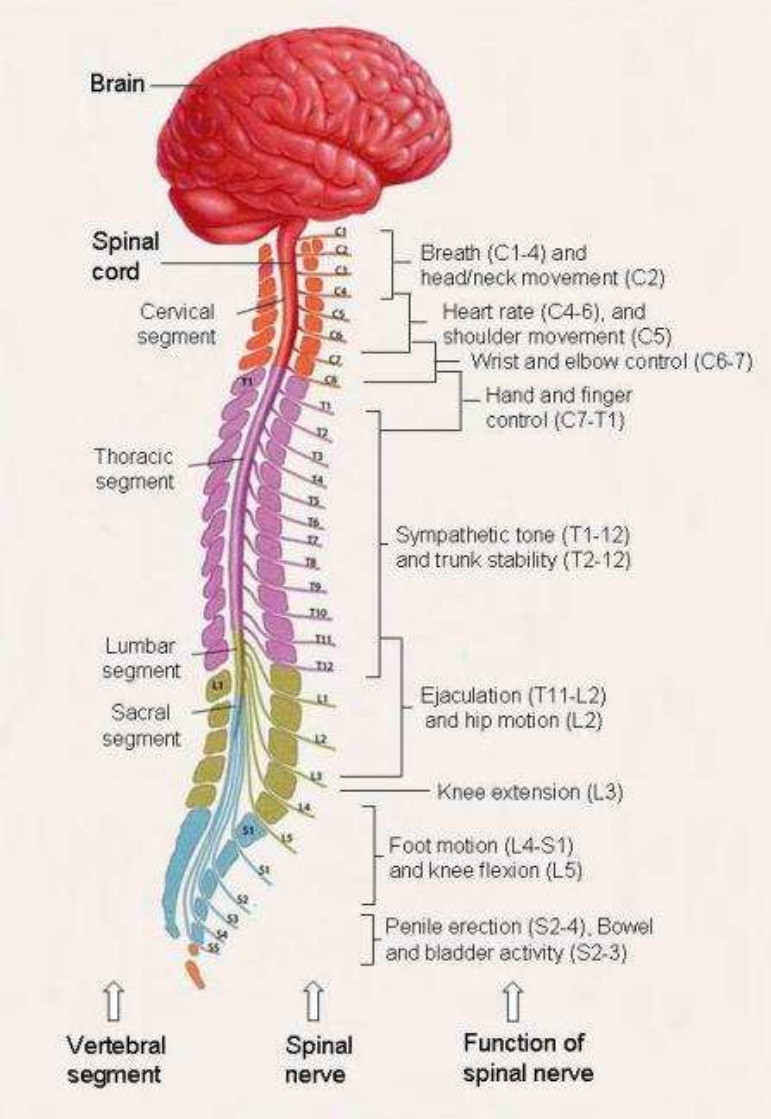


Figure 2.1 – Central nervous system and the functions of spinal nerves [2]

Although, due to the high complexity of neural systems, the mechanism of movement manipulation are still not completely understood. The EMG signal, as the electrical manifestation of muscle activities, reflects the activity of α MNs in the spinal cord via the peripheral nerves conduction, thus offers us the probability to exploit the neuromuscular system.

The CNS takes charge of recruiting MNs in the spinal cord, in the order of from the smallest to the biggest MU [3], based on the size of the load. The MUs are normally divided into four categories [4] based on their physiological properties: The first three types of MU: fast fatigable (FF), fast intermediate (FI)

and fast fatigue resistant (FR), are all recruited in the movement with fast speed, but with different levels of force. The lower level force is related to the higher resistance of fatigue. The forth type of MU, slow oxidative (SO), recruited during the slow contraction with low force shows a extremely high resistance of fatigue.

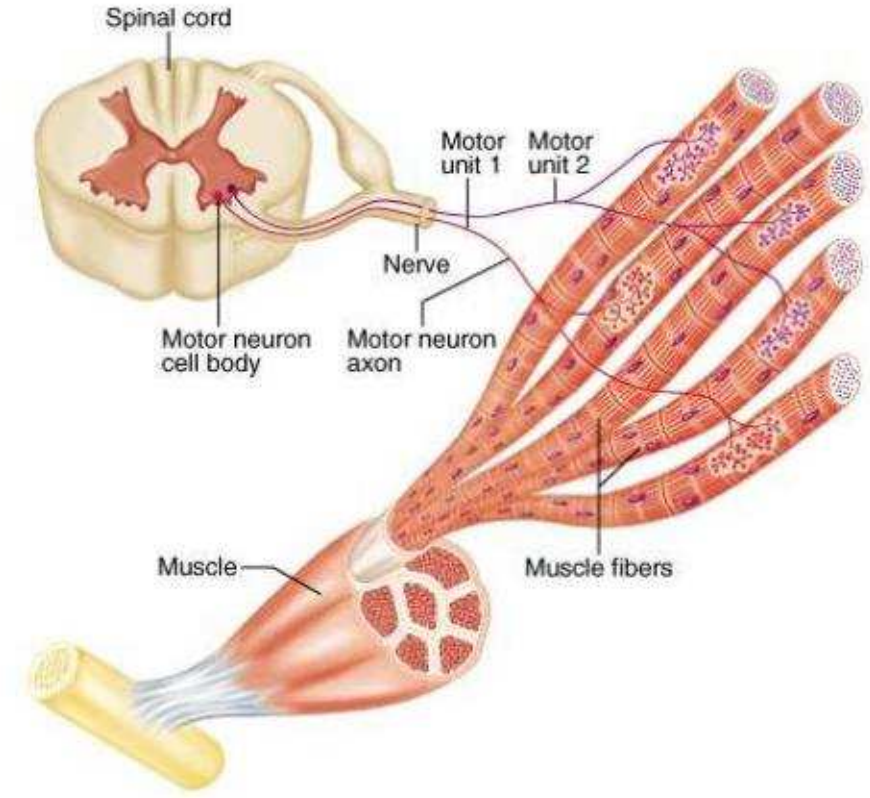


Figure 2.2 – Anatomical characteristics fo MU: MNs that innervate the individual muscle are termed motor nucleus or motor neuron pool and cluster in either the anterior (ventral) horn of the spinal cord. Every MN innervates several muscle fibers. The muscle fibers of a muscle unit intermingle with other muscle units, but do not extend to other muscles nearby. (Copyright©2001 Benjamin Cummings, an imprint of Addison Wesley Longman, Inc.)

A MU consists of an α MN in the spinal cord and the muscle fibers it innervates. Once a MU receives the order from the higher level neural system, its α MN immediately discharges, sending action potentials (AP) propagating along its axon until reaching the neuromuscular junction where they activate sodium channels of muscle cell membrane. Latter provokes a muscle cell's action potential which propagates along the fiber causing its contraction. The innervating process is illustrated in figure 2.2.

Since a α MN cannot discharge all the time, its successive discharges and equilibrias formulate the sequence of action potentials, usually referred as the spike train. Intervals between discharges are under several physiological constrains. Firstly, the discharge patten (firing patten) of an α MN, whose frequence is termed firing rate, exhibits a specific rhythm and regularity, especially during static contractions, thus can be described by the statistic law. Secondly, intervals between discharges are usually not shorter than a certain duration called refractory period, which is in the order of 30 ms [5, 6, 7].

In a MU, the resulting AP discharged by α MN propagates along the membrane of a muscle fiber can be acknowledged as a temporally charging voltage, termed muscle fiber potential [8]. These muscle fiber potentials belonging to the same MU is not time aligned owing to the different location of end-plates. The spatial and temporal summation of these muscle fiber potentials in a MU is called motor unit action potential (MUAP). In order to maintain the muscle contraction force, α MNs discharge repetitively and generate the MUAP trains, where every MUAP is positioned at times of its α MN discharge. A detected EMG signal is the algebraic summation of these generated MUAP trains and the background interference containing the instrument noise and artifacts.

2.2.2 Anatomy of motor unit

MNs that innervate the individual muscle are termed motor nucleus or motor neuron pool and cluster in either the anterior (ventral) horn of the spinal cord or the brain stem [9, 10], as illustrated in figure 2.2. Motor nucleus of proximal muscles are located more ventral and lateral than those of distal muscles in the transverse section of spinal cord, and motor nucleus of anterior muscle are more lateral than those of posterior muscles [11].

The number of MNs in a motor nucleus, the same as the number of MUs that innervate a muscle, ranges from a few tens to several hundred, which is difficult to calculate due to the limitation of available methods. One of the methods is to retrograde transport of horseradish peroxidase (HRP) either by injection into a target muscle or by exposing the cut nerve to the tracer [12], whose difficulty is to deliver sufficient HRP to the target muscle without involving muscles nearby. Another method is human cadavers dissection, which is limited by the assumption: the proportion of large-diameter axons are efferent fibers. Compared to anatomical methods, an electrophysiological method proposed is to measure the amplitude of muscle potentials caused by stimulating the peripheral nerves [13]. The estimates of electrophysiological method are typically lower than those by anatomical methods.

The muscle unit comprises several muscle fibers. The average innervation number, defined as the number of muscle fibers innervated by a single MN, ranges from five to thousand [14, 15]. This average depends on the size of different muscles. Moreover, the innervation number varies within a muscle, with low-threshold MUs containing lower values [16]. Due to the high correlation between the innervation number and maximal force of target muscle, the range of innervation number in a muscle can be estimated by the force [16].

The muscle fibers of a muscle unit occupy a sub-volume of the muscle [17] and intermingle with other muscle units but do not extend to other muscles nearby [18], as depicted in figure 2.2. Some experiments, such as [19, 20], prove that each muscle unit takes only little fraction of the muscle. Therefore, an interesting issue is the transmission force from the contractile proteins of muscles units to the skeleton modified by the connective issue structures. Since muscle units innervated by different MNs are located in different discrete compartments in a muscle, the muscle can be divided into several distinct regions based on different physiological functions [21]. However, subsequent work demonstrates the properties of human MUs are distributed continuously within a motor nucleus [22, 23] and are not divided into distinct groups. Normally, the types of MUs are distinguished with respect to the threshold of force.

2.3 EMG signals decomposition

As depicted in the physiological model, the EMG signal reflects the activity of α MN, thus providing us an insight view of the neural system. Informations regarding MUAP waveforms and MU firing patterns extracted from EMG signals during muscle contractions are widely used in the different domains, for example: MUAP waveforms are used in the diagnosis of neuromuscular disorders [24, 25, 26] and in the estimation of muscle architecture [27]; MU firing patterns are applied for the investigation of central strategies for motor control [28, 29] as well as for creating human-machine interfaces [30, 31, 32]. Therefore, the identification of individual MN spike trains from the EMG, termed EMG decomposition [33] is an indispensable issue. In

recent decades, a great mass of algorithms in a manual, semi-automatic or automatic ways were proposed to resolve this problem. We will talk about them in the following parts of this section.

2.3.1 EMG signals acquisition

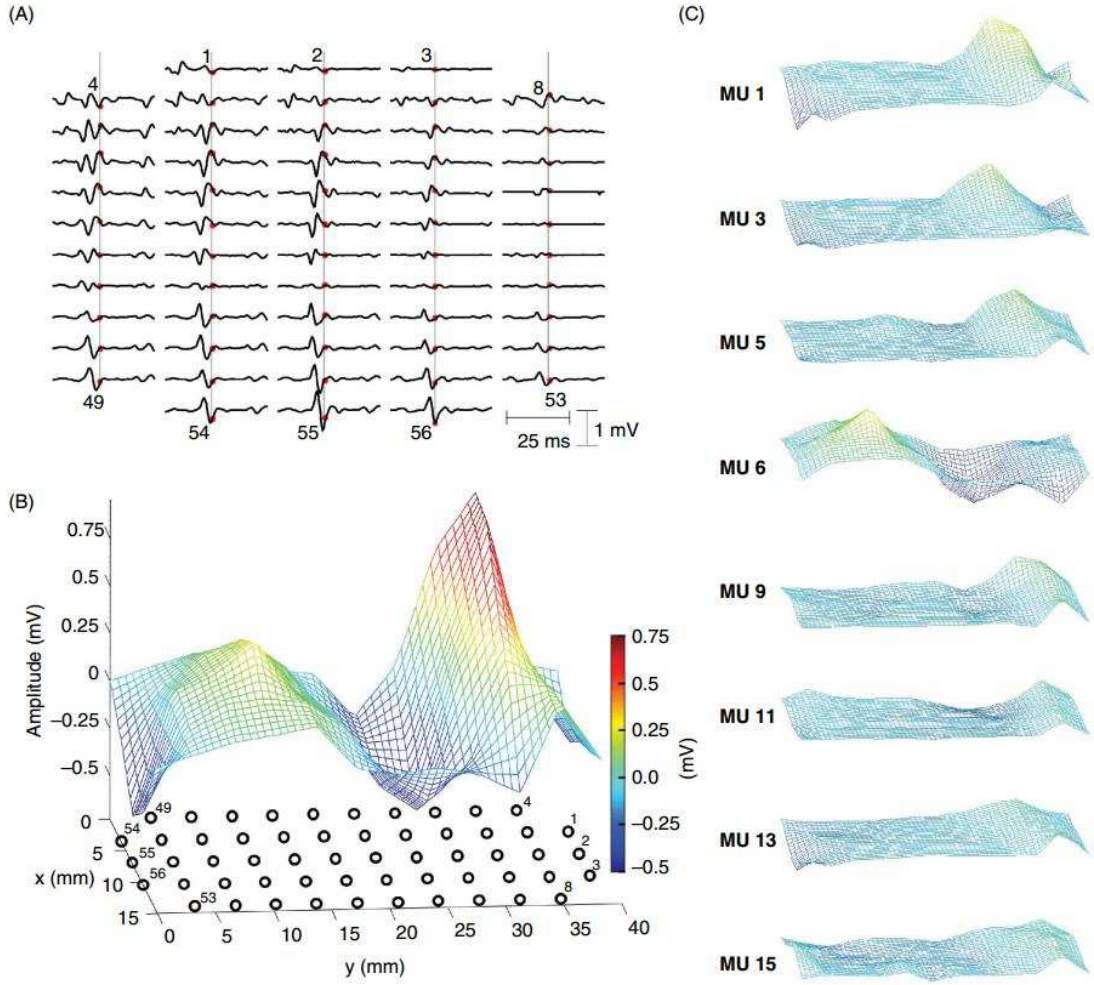


Figure 2.3 – High-density sEMG recordings from a human hand muscle [34]. (A)sEMG signals recorded with an electrode array (5 columns $\times$ 13 rows) placed over the abductor pollicis brevis muscle of a healthy man as he sustained an isometric contraction at 10% of the maximal voluntary contraction force. (B) Distribution of multichannel sEMG amplitude at the instant indicated by the red dots and vertical lines in A. (C) Eight MUAPs detected by the array electrodes.

The beginning step of the EMG signals decomposition is the EMG signals acquisition. An improper EMG signal cannot be decomposed accurately even by the most efficient decomposition method. The type and position of electrodes, the contraction level and the fatigue state of the muscle, as well as the artifacts controlling by the operator, affect the decomposability of a detected signal.

EMG signals are recorded using either surface (surface EMG, sEMG) or intramuscular electrodes (intramuscular EMG, iEMG). sEMG electrodes are usually placed on the skin overlying the target muscle. It has several advantages, such as: non invasive, convenient to use and non pain felling for the subject. However, the volume conductor of the overlying muscles and other subcutaneous tissues acting as a low-pass

filter whose selectivity depends on the distance between the electrodes and the source limits the bandwidth of sEMG signals. Moreover, due to the large pick-up area of surface electrode, it is influenced also by the interference of adjacent muscles (cross-talk) makes the decomposition more challenging. Thus, compared to the iEMG ones, the sEMG signals detect the activity containing more MUs, which overlap mutually and cause cancellations of amplitude, formulating a highly complex signal pattern that is difficult to interpret. In most of sEMG decomposition algorithms [35, 36, 37, 38, 39] proposed recently, the sEMG signal acquisition is always executed in a fashion of multiple electrodes arranged in a two-dimensional arrays in order to obtain the spatial variability of MUAP shapes.

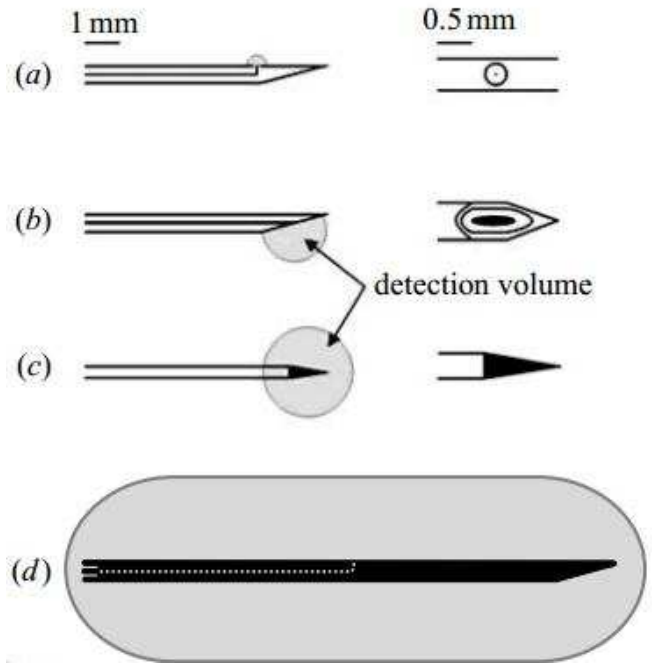


Figure 2.4 – Different types of needle electrodes [40]:(a)Single fiber electrode (b) Concentric electrode (c) Monopolar electrode (d) Macro electrode

Intramuscular electrodes are more spatially selective [41] than the surface ones. The indwelling electrodes inserted directly into a muscle are able to detect the EMG activity in deep area. An iEMG signal is usually produced by a limited number of MUs, with MUAPs distinct from the noise, although superimposed with each other in time. However, some disadvantages are associated with it [40]. Firstly, owing to the invasive property, the iEMG signals acquisition always requires the human interactive. The inserted needle damages some muscle fibers nearby and causes a small local oedema. Sometimes, it could cause unintentional damage of important structures. Secondly, the iEMG signals record only a small number of MUs closed to the detection site. Thirdly, due to the small volume detected by iEMG electrodes, it is difficult to replace the electrode in the same position, thus to repeat the experimentation before. Despite theses limitations, iEMG signals decomposition is attractive for a great many researchers.

Due to the high selectivity of the iEMG signals, many researchers prefer it to offer an accurate decomposition. Therefore, different needles were proposed to measure the iEMG signal to adapt to specific algorithms for information extraction. As one of the earliest proposed needle electrodes, the concentric needle electrode [42] detects signals between the tip of a wire insulated in the cannula and the cannula.

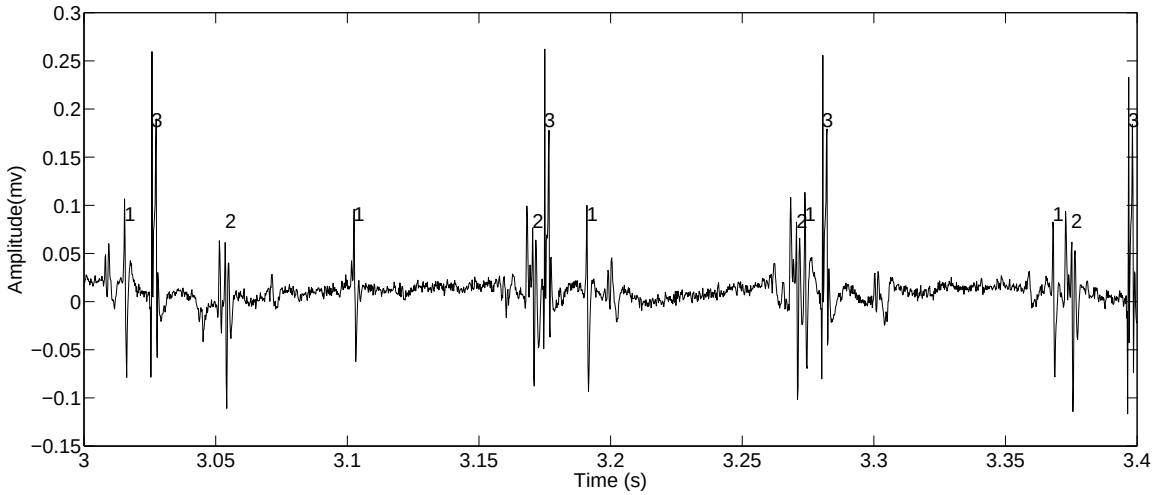


Figure 2.5 – Example of iEMG signal was acquired using 25G wire electrodes (A-M Systems, Carlsborg, WA, USA) made of Teflon coated stainless steel with a diameter of 0.05 mm. The numbers denote indexes of MU.

To fulfil different requirements, some others needle were proposed in recent years, such as: single fiber electrode and monopolar electrode, as figure 2.4 shown. Sometimes, if informations related to MU size or fiber spatial distribution are required, the macro electrode records a more broadly detected signals from the cannula surface.

During the iEMG signal acquisition, slight movements of the centric needle are often inevitable, causing not only the pain of the subject but also the modification of MUAP shapes in the detected signals. To overcome this limitation, wire electrodes [43] were proposed, which are usually appreciated in the long period recording or the recording of movement. The wire is placed in the cannula of a needle and bent at the tip. The needle is inserted into the muscle then removed to make the wire stay in the muscle. Once placed into muscle, its position cannot be adjusted. The wire electrode allows the strong contractions without the discomfort feeling. It shows a stabler property versus the needle ones. An example of iEMG signal aquaired using 25G wire electrodes (A-M Systems, Carlsborg, WA, USA) made of Teflon coated stainless steel with diameter of 0.05 mm, is shown in figure 2.5. Wire electrodes are difficult to arranged in geometrical sites in a single system to extract the spatial informations of the detected muscle. Then, longitudinal intra-fascicular electrodes (LIFE) [44] were proposed to resolve this problem, which are fine wire electrodes implemented into peripheral nerves. The recent generation of this system consists in thin-film LIFE [45]. These systems allow the multi-channel iEMG signal acquisition in a single insertion and the repeat experimentations in the same detected site.

2.3.2 iEMG signals decomposition

As depicted in figure 2.6, α MNs discharge successively in the spinal cord, then sending APs to related muscle fibers to formulate the MUAP trains. The summation of MUAP trans, named 'Raw EMG signal' in the figure, is detected by a needle electrode or a wire electrode. The EMG signal is decomposed in the form of several individual MUAP trains corresponding to each active MU discharge. The individual MUAP train is the convolution of MUAP shapes of each MU and the spike trains, which is the discharge patten of α MN in subsection 2.2.1. In the rest of this subsection, we will present various decomposition algorithms, which were proposed to realise the complete or incomplete decomposition of iEMG signals.

The procedures for iEMG signal decomposition have been progressively improved from methods strongly based on the manual intervention of an operator [48, 49, 50] to semi-automatic and then fully automatic

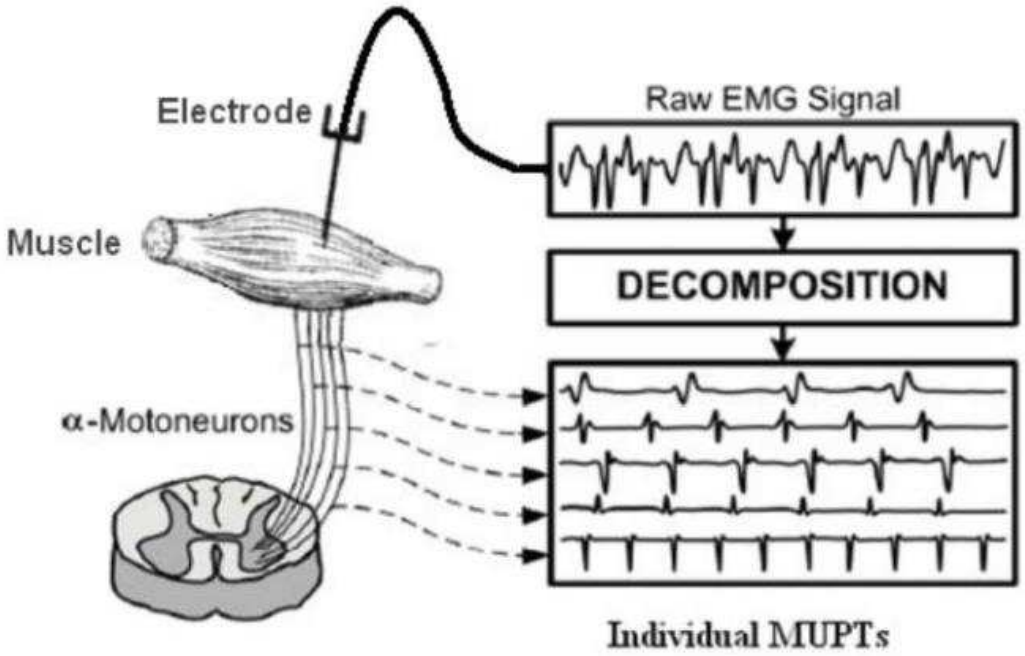


Figure 2.6 – iEMG signal decomposition[46, 47]: α MNs discharge successively in the spinal cord, then sending APs to related muscle fibers to formulate the MUAP trains. The summation of MUAP trans, named 'Raw EMG signal' in the figure, is detected by the needle electrode or wire electrode. The EMG signal is decomposed in the form of several individual MUAP trains corresponding to each active MU discharge.

methods. The initial manual methods rely on the visual inspection of similar MUAP shapes matching in an EMG signal shown on an oscilloscope or plotted on a grid paper. These methods are time consuming and cannot achieve an ideal decomposition, especially in the case of a great many superposed MUAP shapes in the signal. And the quality of decomposition always depends on the experience of operators. With respect to the manual methods, semi-automatic and fully automatic methods are more powerful. Some of techniques are used in both of them. In most of semi-automatic and fully automatic algorithms, the process of MUAP trains identification is implemented by pattern recognition techniques. More recently, blind source separation approaches have been proposed to decompose multi-channel iEMG signals [31, 51], acquired by thin-film LIFE. This kind of approach is appreciated as they are minimally influenced by the MUAP overlap rate.

Pattern recognition methods

Firstly, we present the outline of semi-automatic and fully automatic algorithms based on pattern recognition techniques. These decomposition procedures normally includ the following steps [52] after the acquisition of signals:

— Signal preprocessing
— Signal segmentation: MUAP detection
— Feature extraction
— Clustering and supervised classification of detected MUAP shapes
— Superposition resolving
— Estimation of spike trains statistic parameters (firing rates) and MUAP templates

The first step signal preprocessing is to decrease the influence of background noise and low frequency information, including some small dimension MUAPs taken as noise, by filtering the signal. Moreover, it is also applied to sharpen MUAP shapes and increase the dissimilarity of MUAP shapes, as the example illustrated in figure 2.7. Thus, signal preprocessing improves the MUAP detection and classification. Band-pass filters or low-pass differentiating filters [53], which are easy to implement and fast to execute, are commonly used in this step. These filters suppress the baseline of noise but without attenuating greatly the MUAP amplitude, which simplifies the signal segmentation. Besides the two filters, some others complex wavelets [54, 55] and empirical methods [56] coasting more execution time were applied to remove the noise in the signal. They may work better than the previous two filters. However, their performances depend largely on some pre-defined parameters by users, such as the de-noising threshold or the mother wavelet of the wavelet-based methods.

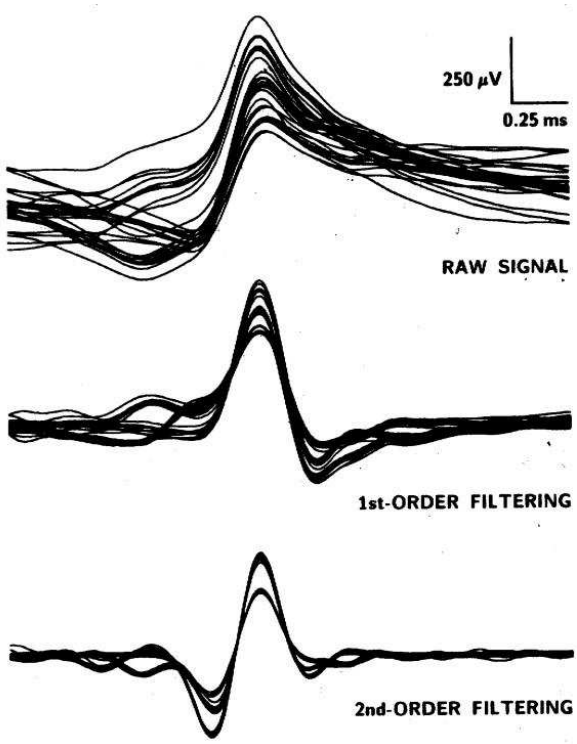


Figure 2.7 – Several similar segments of iEMG signal and the signals filtered by the 1st order and the 2nd order low-pass differentiating filter [53]. Differences between MUAPs were highlighted.

The objective of the second step, signal segmentation, is to separate the noise and the single or super-imposed MUAP segments, thus to detect all activities of MUs. Normally, a threshold crossing the whole iEMG signal is profited. The local peaks of filtered signals exceeding this threshold are taken as the candidate segmentation positions. A window is used to center these peaks and extract the signals nearby to formulate MUAP segmentations. The threshold can be artificially pre-defined as a positive value [46, 47], or can be calculated based on the characteristic of the filtered iEMG signal [53, 57, 58, 59, 60], such as the maximum absolute value, the mean of absolute value and the standard deviation of signal. The length of window can be variable [55, 57], which is adjusted regarding to the duration of MUAPs, or constant, normally chosen as 2.5 ms [61, 62] or 6 ms [58, 59]. The shorter window length can simplify the procedure of clustering and classification, but cause multi-detection of some long MUAP segmentations and the extra fusion procedure, while the longer window length can keep more information of MUAP segmentation, but increase the decomposition time.

In the step of feature extraction, a feature vector is composed of a number of features extracted from MUAP segmentations obtained in the previous step. The dimension of feature vector is typically not too large to require a large mass of computation resource, nor too small to accurately classify these MUAP segmentations. Ideally, the feature vector contains a few number of uncorrelated features which desire less time to calculate and are sensitive to discriminate MUAP segmentations belonging to various MU classes. Moreover, these features should decrease the effect of MUAP superposition and have a tolerance of the variability of MUAP shapes. Until now, different features were extracted to represent the detected MUAPs, improve the accurancy of decomposition and reduce the time execution. These features can be principally divided into time-domain, frequency-domain, wavelet domain features, such as: first and second derivative time samples [46, 47, 61, 62], variance [63], mean frequence and mean power [64], total power and variance of central frequence [65], power spectrum and Fourier transform coefficient [53], and wave coefficient [55, 66]. In addition, besides different features in time-domain, frequency-domain or wavelet domain, some morphological features are involved in the decomposition algorithm because they are easier to calculate. Morphological features are briefly presented in table 2.1.

Table 2.1 – Morphological features of MUAP shapes

Name	Description
Duration	The time between the onset and the termination of an MUAP.
Peak-to-peak amplitude	The difference from the maximal to the minimal peak
Number of phases	The number of baseline crossing plus 1
Number of turns	A turn is a positive or negative peak that is separated from a previous and a following peak of opposite polarity.
MUAP area	The integration of the rectified MUAP over its duration.
Maximum positive (or negative) peak amplitude	the maximum positive (or negative) peak in the MUAP segmentation

Both of clustering and supervised classification is the task for partitioning the MUAP segmentations represented by the feature vector into various different MU classes according to their similarities and correlations. Generally, we define clustering as the first phase of supervised classification. The several beginning seconds of signal are usually taken for the clustering phase as the training sample of supervised classification. Although, in some algorithm, there is only clustering, no supervised classification. Regarding to the aspect of initial condition, no prior knowledge is offered in the clustering, while some informations obtained in the clustering, which is typically the pre-classified MUAP containing the number of MUAP and basic features of MUAP shapes, are initially provided for the supervised classification.

For the clustering, the main difficulty is finding the optimum number of clustering with the interference of MUAP superposition. Some popular clustering algorithms are applied in this phase: K-means, different types of support vector machine, fuzzy c-means and the hierarchical algorithm. Moreover, some decomposition methods combines the clustering with a mathematical model in order to find the spike train of each MU. [67] use the combination of MUAP shapes and MU firing patten provided by a spike train constrained Viterbi algorithm to find the best combination for each MUAP segmentation.

In the supervised classification phase, the variability of MUAP shapes caused by the movement of needle inserted in the muscle, by the neuromuscular jitter or by the fatigue of muscle, can increase a large number of missed classification error, thus reduce the decomposition accuracy. To resolve this problem, the firing rate pattern of MU is often taken as the regularity restriction for the classification. It is used passively in the way of testing the regularity to fill the gap or eliminate the peak of MUAP trains, or actively with the MUAP shapes to determinate the certainty in assigning a MUAP into its MUAP trains. However, two disadvantages limit the implementation of firing rate pattern. The first one is the double discharge of MU (firing twice closely in time). Second, a relative stable stochastic process of firing rate patten is normally in the case of slow change of force. In the case of fast change of force, the firing rates are largely time-varying and difficult to trace.

EMG signal decomposition employs several classification techniques in a variety of algorithms. The maximum a posterior classifier is applied in the algorithm, named Pricision Decomposition 1 (PD1), of [46, 47]. This system was able to decompose iEMG signal less than 8 MUs with an accurancy of 60 to 70% in an automatic mode. And it provides a way of human interaction to reach higher accuracy. To improve the performance of PD1, the artificial intelligence-based maximum a posteriori classifier is used in [68, 69, 70], termed Pricision Decomposition 2 (PD2). PD2 set automatically the parameters in PD1 based on the statistic characters of iEMG signal with a knowledge-based artificial intelligence framework. Moreover, with respect to PD1, this system can resolve complex superimposition. Both PD1 and PD2 use multi-channel iEMG signals detected by the special quadrifilar needle or wire sensor. Pricision Decomposition 3 (PD3) was proposed in [71] but for sEMG signal, which will be presented in subsection 2.3.3. Some algorithms prefer the fuzzy logic-based classifiers. [61] use a fuzzy k-nearest neighbor classifier for the MUAPs supervised classification after the initialisation clustering phase. Certainty-based classifiers [72], which combine MUAP shapes and MU firing rate patten, make an evaluation of certainty to assign a MUAP to one of MU classes. It classifies the MUAP to a class if the largest certainty of related class is greater than the certainty assignment threshold. Otherwise, this MUAP segmentation is unassigned and taken as the superposition. Moreover, multiple classifier systems [73, 74] were built to achieve a higher accuracy. [73] was composed of adaptive certainty-based classifier, adaptive fuzzy k-nearest neighbor classifier and adaptive matched template filter classifier. This classifier fusion system showed a better classification performance, especially regarding to eliminate classification errors.

Two or more different MUs fire simultaneously or within a sufficient short time less than the length of MUAP shapes, which is the superposition of MUAPs. The peel-off method [46, 75, 58, 76] based on MUAP shapes matching is the simplest approach to resolve the superposition problem. By measuring the similarity form between the superimposed MUAP segmentation and the MUAP template, we select the most probable template and then the superimposed MUAP segmentation subtract this template. The resulting new form repeats this process until reaches the stop criterion. Another method is the modelling-based approach [77, 78] which perform better than the peel-off method but is more time-consuming. The goal of this method is to find a sum of convolution between MUAP shapes and the firing time to minimize the criterion:

$$e = \|MUAP_{SUP} - \sum_i h_i * U_i\|^2 \tag{2.1}$$

Where $MUAP_{SUP}$ is the superimposed MUAPs, h is the MUAP shapes, U is the spike train, i denotes the index of MU and e represents their square of difference. The information of MU firing rate patten is sometimes taken as a reference in the superposition analysis.

When we complete the iEMG decomposition, the spike train or MUAP trains of each MU is identified. Estimations of MUAP templates and MU firing patten statistics are the final step for the various following applications. During the process of MUAP shapes estimation, several methods, such as mean [79, 80], median [80] and interference cancelling averaging technique [53], were applied to reduce the influence of the noise and MUAP shapes of other MUs because of improper classifications. If the number of MUAP segmentations classified to a train is large enough, mean estimation shows a good performance and provide a better SNR than other estimation. Otherwise, if the number is small, the median or median trimmed mean averaging estimations are preferred because they can reduce the interference of MUAP shapes of other MUs because of improper classifications. For the estimation of MU firing patten statistics parameters, once the spike train is offered, the firing rates can be calculated by the number of firing spikes in a second.

These algorithm presented before are typically evaluated by the MUAP shapes estimation and the accuracy spike trains, which corresponds to the miss assignments and over assignments of spikes. However, in these algorithms, their resolved problems (with or without MUAP shapes superposition), their validations on simulated and experimental signals with different durations, number of MUs, percentage of superposition, similarities of MUAP shapes and ratios of signal to noise (SNR), and their different assessment indexes make the comparision of performance evaluations difficult. There is not a well-known conventional evaluation criterion applied in all papers, although a few practical assessment indexes were proposed in

[81].

Most of the algorithms presented are fully automatic methods, except PD1 [46, 47] and EMGlab [79] which are semi-automatic and need human intervention, especially EMGlab, offering an Graphical User Interface for users. And we notice that almost all pattern recognition methods decompose iEMG signal in a non-sequential manner, which cannot carry out the real-time decomposition.

Blind source separation approaches

Except these pattern recognition methods, blind source separation approaches [31, 51] applied for the iEMG decomposition were proposed in the last few years. The most famous method Convolution Kernal Compensation (CKC) of [35], by wihch a convolution matrix comprising the information of MUAP shapes is compensated, designed for the decomposition of multi-channel sEMG signals can be also used for multi-channel iEMG signals decomposition. We will present it in subsection 2.3.3. In [31], an extended measurements was used in the formula 2.2 the same as the extended convolution model proposed in [35], in order to describe the conditions under which the assumptions of the convolutive blind separation model are satisfied. Then, it proposed an approach of convolutive sphering of the observations which is based on [82], followed by an iterative extraction of the sources. In [51], another blind source separation approache is presented. The cyclostationary properties of MUAP trains are used to decompose the iEMG signals.

With respect to the pattern recognition algorithm, blind source separation approaches are generally applied for the multi-channels iEMG signals and usually have a relative high resistance for the interference of MUAP shapes variability. Due to the large amount of the multi-channels iEMG signals, they need more time to process signals. Moreover, their performance strongly depends on the number of available channels [51]

On-line decomposition method analysis

In [83], a real-time decomposition method for the signal channel iEMG signal based on the pattern recognition techniques was proposed. This method is composed of on-line single-pass density-based clustering and adaptive classification of bivariate features containing the root mean square and the difference absolute standard deviation, using the concept of potential measure. The superposition problem which is the most complex and time-consuming problem in the decomposition was not taken into consideration in this paper. This algorithm achieves a high accuracy in the classification of the non-supervised MUAP segmentations with a speed of 200 ms signal decomposed within 21 to 97 ms. However, it is limited for the movement with high maximal voluntary contraction force (MVC), in which the iEMG signals with more supervised MUAPs segmentations is generated.

In [84, 85], a new algorithm that allows a full decomposition of single-channel iEMG signals produced during dynamic contractions at moderate force levels was proposed. The algorithm is based on a Markov model of the iEMG, which takes into account the varying number of active MUs and the regularity of their spike trains. Joint superposition resolution, MUAPs shape updates and firing statistics estimation are achieved by applying the Bayes filtering. Sliding window approach made the algorithm adaptive to variations of contraction forces and the variability of MUAP shapes.

Although this algorithm needs long time to complete the decomposition, it shows a potential to realise the on-line decomposition. The sequential decomposition way in this algorithm is a necessary condition for the real-time decomposition. Besides, because of its parallel structure, this algorithm can be efficiently accelerated by parallel computation implementation in GPU.

2.3.3 sEMG signals decomposition

Similar to the iEMG signal, sEMG signal is also a measure of muscle activities and thus, an output of the MN spike trains, containing the information on both central and peripheral neural mechanisms for movement generation. However, compared to the iEMG signal, sEMG signal has much higher complexity

due to the superimposition of sources of neural informations at different levels and in mixing unknown process, as shown in figure 2.8. Therefore, the extraction of sEMG information which is taken as the source separation focus on different scales [86]. Figure 2.8 shows the multi-scales form left to right. Firstly, sources in the sEMG signal can be viewed as the high-level commands of the coordinated activity of multiple muscles, where the sEMG is interpreted as the complex muscle activation patterns [87, 88]. Secondly, at a smaller scale, source is considered as the neural activation to different muscles, where the sEMG is processed to identify the activity of target individual muscle or muscle compartments [89, 90]. At this scale, the great challenge is that the record sEMG is usually the mixture of the activities of many concurrently active muscle compartments and neighbor muscles, termed as EMG crosstalk. Finally, at the individual muscle level, source is taken as the activation of the smallest functional units (MUs), where the sEMG is identified in the form of spike trains of each MU and then, the correlated discharge rates and MUAP shapes are estimated [35, 91]. The decomposition of sEMG at this scale is to extract the information form the α MN to the target muscle and thus, provides a definitive insight into the net output from the CNS and PNS. In the decomposition, since the MUAP shapes vary with respect to many factors, such as the subject anatomy, muscle fatigue and etc, they are usually unknown a priori. Moreover, the volume conductor, defined as the tissue separating the muscle fibers and the recording electrode, works as a low-pass spatial filter, where all the signal collected is low-pass filtered with reference to the distance between the muscle fibers and the electrodes. Despite these restrictions limit the sEMG decomposition, some algorithms were proposed to identify the spike trains of MUs in an efficient and robust way, as it presented in [35, 92, 71, 91].

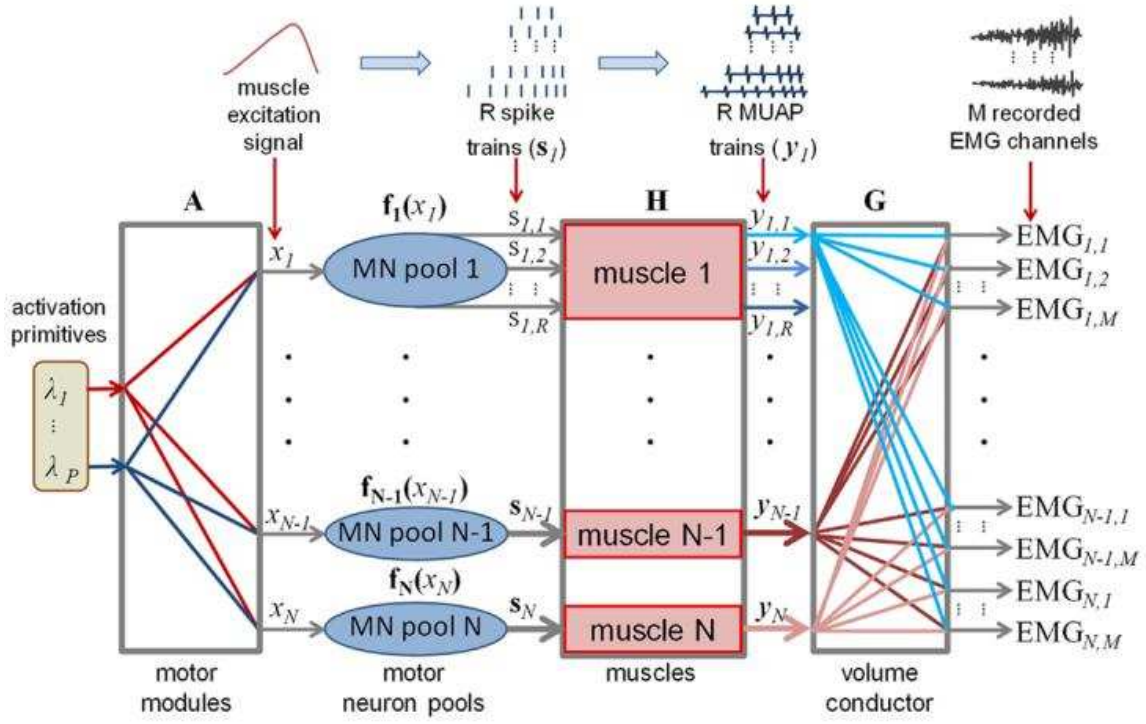


Figure 2.8 – The multi-scale model of movement generation [86]. From left to right: P activation primitives are shared by N motor neuron pools $(P < N)$. Each MN pool receives the linear combination of the P activation primitives as input and transforms this input into spike trains that drive the innervated muscle (thus, the scheme presents N muscles). The N muscles contribute to $M(M > N)$ EMG channels by the trains of MUAPs.

To exploit the spatial variability of the detected MUAPs in the sEMG signal, the surface electrodes are always arranged in to two dimensional arrays on the target muscle. Therefore, the multi-channel sEMG signal recorded in the grid electrodes were modeled as the correlated instantaneuos or convolution data models. In the instantaneuos model, the MUAPs trains recorded in the different locations were assumed to vary only in the amplitude. For the i-th MU recorded in the j-th electrode, we have the formula: $\mathbf{MUAP}_{ij} = g_{ij} \times \mathbf{MUAP}_{st}$, where g_{ij} is a scalar weight and $\mathbf{MUAP}_{st}$ stands for the standard constant MUAP shape of the i-th MU. Different with the instantaneuos model, the convolution model assumed that the discharge timings of each MU, that is, the spike train, were shared in the recording of different electrodes, while the detected MUAP shapes were taken as the mixing process itself. It is evident that the convolution model is much more general, realistic and closer to the physiological MUAP model, exhibiting the spatial variability of MUAP shapes.

Both of the instantaneous model and the convolution model can be modeled by the formula:

$$\mathbf{EMG}(n) = G\ H\ \bar{s}(n)\ +\ w(n) \tag{2.2}$$

where $\mathbf{EMG}(n) = [\mathbf{EMG}_1(n), \cdots, \mathbf{EMG}_{NM}(n)]$ denotes a vector containing NM channel sEMG signals, and $w(n)$ is the vector of noise. The vector $\bar{s}(n) = [s_1(n), s_1(n-1), \cdots, s_1(n-L), s_2(n), \cdots, s_{NR}(n), \cdots, s_{NR}(n-L)]$ contains L consecutive samples of R MU spike trains in the N MN pools (motor nucleus). In the case of instantaneous model, G is a $NM \times NR$ matrix comprising the scalar weight of MUAP and H is a block diagonal matrix composed of the NR MUAPs. The formula H is shown:

$$H = \begin{bmatrix} \mathbf{MUAP}_1(n) & 0 & \cdots & 0 \\ 0 & \mathbf{MUAP}_2(n) & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{MUAP}_{NR}(n) \end{bmatrix} \tag{2.3}$$

In the case of convolution model, the matrix G and H can be modeled by a matrix containing the MUAPs shapes of all MUs recorded in all electrodes:

$$GH = \begin{bmatrix} \mathbf{MUAP}_{1,1}(n) & \mathbf{MUAP}_{1,2}(n) & \cdots & \mathbf{MUAP}_{1,NR}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{MUAP}_{NM,1}(n) & \mathbf{MUAP}_{NM,2}(n) & \cdots & \mathbf{MUAP}_{NM,NR}(n) \end{bmatrix} \tag{2.4}$$

Based on the convolution model, some algorithms proposed for the sEMG signals decomposition reveal a good performance, especially the algorithm of Convolution Kernal Compensation (CKC) in [35] based on blind source separation techiniques. The CKC estimator is principally based on the computationally attractive linear minimum mean square (LMMSE) estimator, which is Bayesian optimal for linear mixing system, and can estimate the spike trains of individual MU without the calculation of the mixing matrix GH. The LMMSE estimator assumes that the mean of the first two moments of the source signals (spike trains) and their cross-correlation with the observation signals are known in priori. Some supervised ways involving the human intervention were proposed before to overcome this problem. In the CKC estimator, the mixing matrix GH is compensated by the calculating the square of Mahalanobis distance of the extending vector $\bar{s}(t)$ and thus, find the indexes and time instant of MUs arising the spike in a segmentation of observed signal. Then, according to the analyze of the superposition MUAPs probability, some moments of source signals are estimated. Thus, the condition a priori of LMMSE is fulfilled in this automatic manner in the CKC estimator. The performance of this algorithm shows a high accuracy and robustness. It allows weak correlations between MUs. Moreover, there is no parameter to determine a priori in the decomposition.

Some ameliorations and validations of CKC estimator, such as identification of MU behavior in pathological tremor, were proposed in [36, 37, 38, 39]. An algorithm proposed in [93] combining the the K-mean clustering method and a modified CKC method for multichannel sEMG decomposition were validate in the signals more than 10 MUs with a high accuracy. Moreover, a real-time decomposition algorithm based

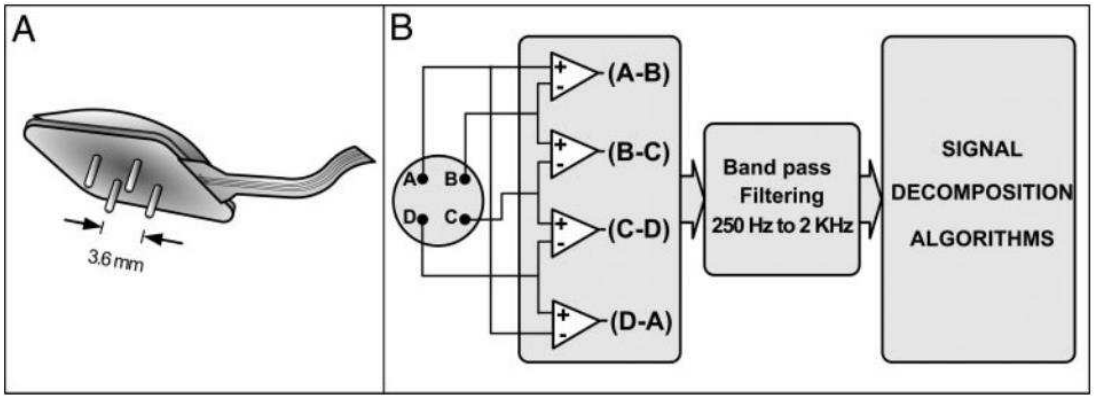


Figure 2.9 – A: The quadrifilar needle sensor with larger dimensions containing the 4 pins that detect the sEMG signals. B: Differential combinations that produce 4 channels of sEMG signals.

on CKC estimator is presented in [92]. Compared to the off-line version, the real-time one begins with a 3 s batch processing of sEMG signals and then, a tracking formula implemented in the CKC estimator to realize the recursively estimation. The results of experimental signals decomposition reveal that it is unable to detect some source trains compared to the off-line one and has a poor performance in the beginning of decomposition. In addition, it is evident that the first 3 s signal is not identified in a real-time way. But the decomposition results after 3 s are relatively high coincidence with the off-line estimation.

Besides the series of CKC methods, there are also lots of algorithms, which prefer the pattern recognition techniques similar to the iEMG signal decomposition, but using some special dedicated surface electrodes to detect sEMG signals with more discriminating MUAP shapes [71, 91], or making a similar assumption [94, 95]. In [71], Precision decomposition 3 (PD3) was proposed for sEMG signal decomposition, while PD1 and PD2 systems were designed for the iEMG. With respect to PD2, PD3 system does not show an evident improvement. Both of them use a knowledge-based artificial intelligence framework to automatically decide the parameters based on the processing signals and to resolve the superposition. The greatest difference between them is that there are more differential electrodes in PD3, as depicted in figure 2.9, to offer a specific version of signals. Latter, another special detector was presented in [91], as shown in figure 2.10. This five-pin surface sensor can detect sEMG signals with more discriminating MUAP shapes. Then, a new Precision decomposition system containing two phase PD-IPUS and PD-IGAT was proposed. PD-IPUS is almost the same system as PD3 presented in [71] to identify the MUAP segmentation without significant superposition and update the time-varying variations of MUAP shapes, while PD-IGAT comprises a MUAP template-matching procedure and an iterative MUAP discrimination analysis to resolve the superposition. As presented in [91], experimental signals detected from five different muscles were decomposed with average accuracy 92.5% during the isometrical contraction at force level ranging up to 100% MVC, which needed a long processing time.

2.4 Parallel computation with graphics processing unit

In the last few ten years, we have entered to the epoch of GPU computing. The GPU computation taking a relative important place in the field of high performance computing (HPC) is applied in a great number of applications with substantial parallelism to achieve superior efficiency. In this section, we will present simply the development of GPU computing and its basic knowledge.

When we talk about GPU computation, we normally make a comparison with Central Processing Unit (CPU) that is highly optimised to execute orderly a series of operations. One of the important index to

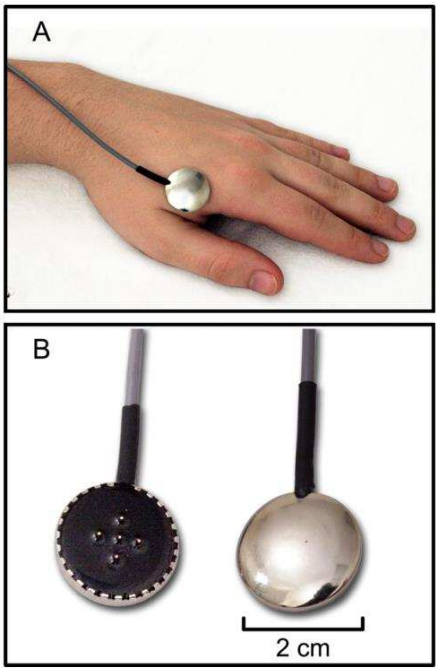


Figure 2.10 – A: The five-pin surface EMG sensor attached above the First Dorsal Interosseous muscle in the hand. B: Top and bottom views of the sensor. The four pins on the corner of a square are spaced 3.6 mm apart.

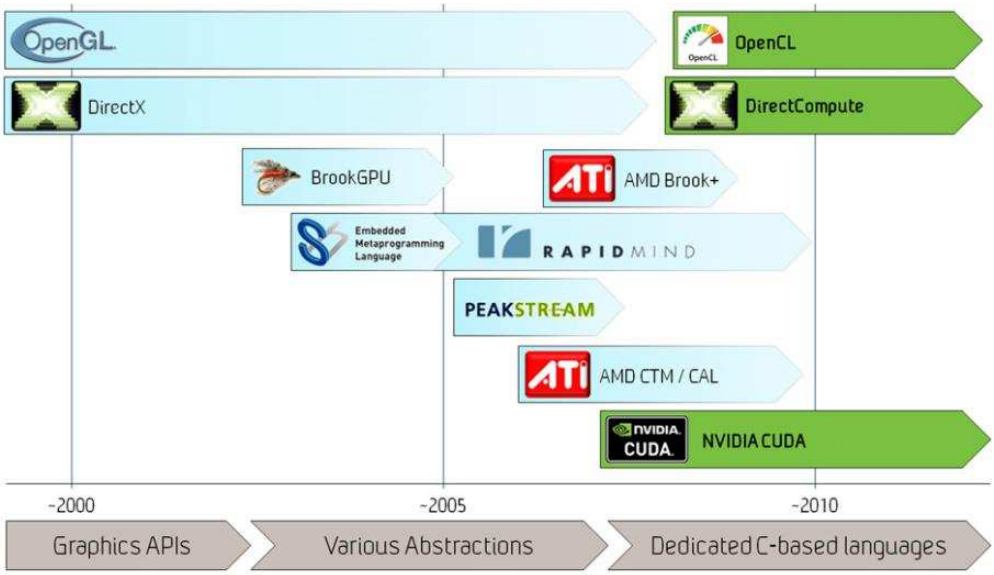


Figure 2.11 – The development of language GPU [96]: In the early 2000s, we had to write the application in shading language. Latter, several academic and third-party languages that abstracted away the graphics appeared. Since 2007, following the step of Nvidia, all GPU manufactories launched eventually the GPUs programmable in common program language for the general-purposed computation.

evaluate the performance of CPU was its frequency. If we double the frequency of CPU, we double its performance. In a long time, we ameliorated the performance of CPU by increasing its frequency until we hit the 'power wall'. In fact, if you fix the voltage, the power consumption of a CPU is approximately the cube of its frequency (clock rate). When we reach a clock rate at around 4 GHz [97], we cannot cool sufficiently the heat of chip converted from such power. Due to the single-core CPU had reached its ceiling, multi-core CPU was designed to increase performance. Meanwhile, the GPU drew more and more attention due to its massive parallelism.

In the early 2000s, GPU was designed to produce a color for every pixel on the screen using programmable arithmetic units known as pixel shaders. The possessed programmable pipelines, by which the user writes a single thread program to making the GPU draw multiple pixels in parallel, attracted the interest of many researchers to exploit the possibility of using graphics hardware for the general-purposed computation. However, in the graphic program, several disadvantages limit the application of the general-purposed computation in the earlier GPU. First of all, the graphic program must be written in shading language, such as Cg or High-Level Shading language (HLSL) [98]. The users should store their data in graphics textures and execute computations by calling OpenGL or DirectX functions, which was coded with shading language. Thus, before enjoying the massive parallel computation power of GPU, the users should learn the knowledge of computer graphics and shading languages. Secondly, there were serious limitations on how and where the user could write results to memory. Finally, it was impossible to terminate the graphic program when we got an unexpected results. There weren't any reliable debugging methods.

With the Nvidia GeForce 8800 introduced in 2006, the first unified graphics and computing GPU architecture was programmable in CUDA C with the CUDA parallel computing model, as well as using DX10 and OpenGL. NVIDIA took industry standard C and added a relatively small number of keywords in order to harness some of the special features of the CUDA Architecture. Latter, Tesla T8 C870, as the first GPU computing system programmed in CUDA C with CUDA was launched in 2007. Compared to the earlier GPU, the CUDA architecture comprised an unifed shader pipeline [99], which allows every arithmetic logic unit (ALU) on the chip to be marshalled by a program intending to perform general-purpose computations. These significant milestones in the development of Nvidia GPU technology, as well as the GPU computing, signified that users were no longer required to have any knowledge of the shading language. As shown in figure 2.11, since 2007, following the step of Nvidia, other GPU manufactories released general-purpose languages for the GPU programming [96]. Nowadays, the most popular GPU programming language are NVIDIA CUDA, DirectCompute, and OpenCL. After the first shoot for language C, CUDA as a parallel computing platform and application programming interface (API) model created by Nvidia released the restriction for other programming language, such as: C++, Fortran and Matlab.

There are three major GPU companies: Intel, Nvidia and AMD for the Personal Computer (PC) market today. Intel is the largest one, but focusing on the integrated and low-performance GPU market, while Nvidia and AMD are the major supplier for the high-performance market. In the academic and industrial field, Nvidia has a dominant position due to the community support, despite the GPU of AMD shows the same excellent performance. Therefore, we focus on GPU of Nvidia in the thesis.

There are three series of Nvidia GPU termed respectively Tesla, Quadro and GeForce. Quadro is designed for the professional workstation; Tesla is used for the scientific computation; And Geforce aim at the market of gaming. Although some laboratories prefer Geforce to execute the scientific computation due to its low price and high power. The computation stability of Geforce cannot be compared with Tesla. After the first generation of GPU designed specially for General-purpose computing on graphics processing units (GPGPU) [100] with the architecture named Tesla, the same name as the series, Nvidia cooperation developed gradually GPUs with Fermi, Kepler, Maxwell and Pascal architectures for GPUs. With the improvement of GPU architecture, a huge leap forward in power efficiency is offered and a great many features were developed. The GPU power efficiency reached the level of TFlop of double precision, Where TFlop refers to the capability of a processor to calculate one trillion floating-point operations per second. Various type of GPU memories containing the register, shared memory, constant memory and global memory breaks the memory wall thus reduce the consummation of memory transition. Multi-streams allow the si-

multaneous execution of kernel functions (GPU program) [101]. Dynamic Parallelism technique adds the capability for the GPU to generate new work for itself without involving CPU. And NVIDIA GPUDirect [102] technique allows that GPUs in different servers located across a network communicate directly with each other without needing to go to CPU/system memory. A great number of other features which increase GPU utilization and simplify parallel program design will be not presented here.

Since the incredible performance of GPGPU in the program with massive parallelism, this technique has been applied in a variety of applications [99]. In the domain of medical imaging, TechniScan system has developed a promising, three-dimensional, ultrasound imaging method. Due to its tremendous computations, TechniScan system was difficult put into practice. The GPGPU with high performance GPU resolve easily this problem. Moreover, the GPGPU techniques are wildly used in the computational fluid dynamics, environment science and etc. In this thesis, we will present the GPGPU applied in the domain of biomedical, the decomposition of iEMG signals.

2.5 Discussion and conclusion

Researchs on prosthetic commands by EMG signals representing one of the means to fight against the effects of ageing and handicaps, draw more and more attentions of researcher fellows. Some remarkable advances in prosthetic control based on EMG-based surface classification methods are shown in recent years [103]. However, the decomposition of EMG signals and in particular of iEMG signals can be a promising study to obtain the effective information of neural system, which controls the muscle activities, in order to perform more complex tasks of capture and manipulation than the currently research.

Most of the current iEMG decomposition methods are based on MUAPs detection and clustering. Some of these do not take the temporal superimposition of MUAP waveforms into account [104, 83], leading to underestimation of the discharge rates, especially at higher contraction forces, when most of the MUAPs are superimposed. Other algorithms (e.g., [57, 105, 106]) achieve complete decomposition in an off-line manner by first identifying and clustering the non-overlapped MUAPs and then iteratively searching for their occurrences in the superpositions. More recently, blind source separation approaches have been proposed to decompose multi-channel EMG signals [31, 35, 92] but their performance strongly depends on the number of available channels [51].

A sequential decomposition algorithm based on a Hidden Markov Model of the EMG, that used Bayesian filtering to estimate the unknown parameters of discharge series of motor units was previously proposed in the laboratory LS2N. This algorithm respects the on-line decomposition manner. However, it requires a high time consuming. Thus, we improved, simplified and accelerated this algorithm to achieve a real time decomposition with high quality performance. In the following chapters, we will present precisely the proposed real time decomposition method.

3

Hidden Markov model

3.1 Introduction

In this chapter, the physiological model of iEMG signal is modelled by a linear convolution model proposed in [107, 108]. Then, a Hidden Markov Model of iEMG proposed in [84, 85] is presented, in which each motor neurone spike train is modeled as a renewal process with inter-spike intervals following discrete Weibull law. The proposed model also incorporates motor unit activation and inactivation making the algorithm adaptable to the recruitment process. With respect to the recruitment model proposed in [84], the MU activation process is modified.

3.2 Modelling of EMG

As presented in the physiological model in section 2.2.1, once a MU is recruited by the higher level neural system, its α MN discharges immediately. Its APs are sent to the neuromuscular junction where they activate sodium channels of muscle cell membrane, via its axon. Then the corresponding muscle cell's APs propagate along its muscle fibers to generate a contraction. The spatial and temporal summation of these muscle fiber potentials in a MU is MUAP.

In the mathematics model, the discharge pattern of a MN is modelled as a sparse sequence comprising 0 and 1, where 1 denotes the discharge and 0 represents the other case. MUAP shapes are modelled as the impulse responses. According to the linear additional property shown in the physiological model, [107, 108] proposed a linear model. The iEMG signal is modelled as the sum of the convolution between spike trains and their proper impulse responses and then is added with the noise:

$$Y[n] = \sum_{i \in A[n]} (h_i * U_i)[n] + W[n] \tag{3.1}$$

— Y denotes the observed signal, the iEMG signal;
— U, a discrete binary sequence made up of ones and zeros, represents the spike train;
— h, a vector with finite scalar elements, is the impulse response, representing the action potential of motor unit, referred as MUAP in the physiological model;
— W is the noise.
— A is the set of active MU indexes.
— n represents the time index or time instant.

— i denotes the MU index;

An example of the linear model of iEMG signal is illustrated in figure 3.1. A motor nucleus or motor neuron pool containing multiple MUs that innervate the same muscle. When a contraction is generated in this muscle, some MUs are recruited, such as the first, i-th and p-th one in the figure, while others are inactive, such as the second one. Each active MU generates a spike train $U_i[n]$, then filtered by the impulse response h_i to formulate the MUAP trains. The sum of all MUAP trains is the iEMG signal. In the final, due to the instrument or the artifacts, the signal is inevitably contaminated by the noise.

In the physiological model, the time is continuous, whereas the time in the linear convolution model is discrete due to the sampling. Thus, in the formula 3.1, the time represented by discrete time index n and sequences Y, U, and W are all discrete.

To simplify this model, some assumptions are proposed:

— W is an independent and identically distributed white noise, with a unknown constant variance v;
— All the sequences of spike train $(U_i)_{i\in\Omega}$ and the sequence of noise W are mutually independent, where Ω is the set of all MU indexes, including the active and inactive ones;
— $(h_i)_{i\in\Omega}$ are unknown. A rough initial forms of each MUAP is offered.

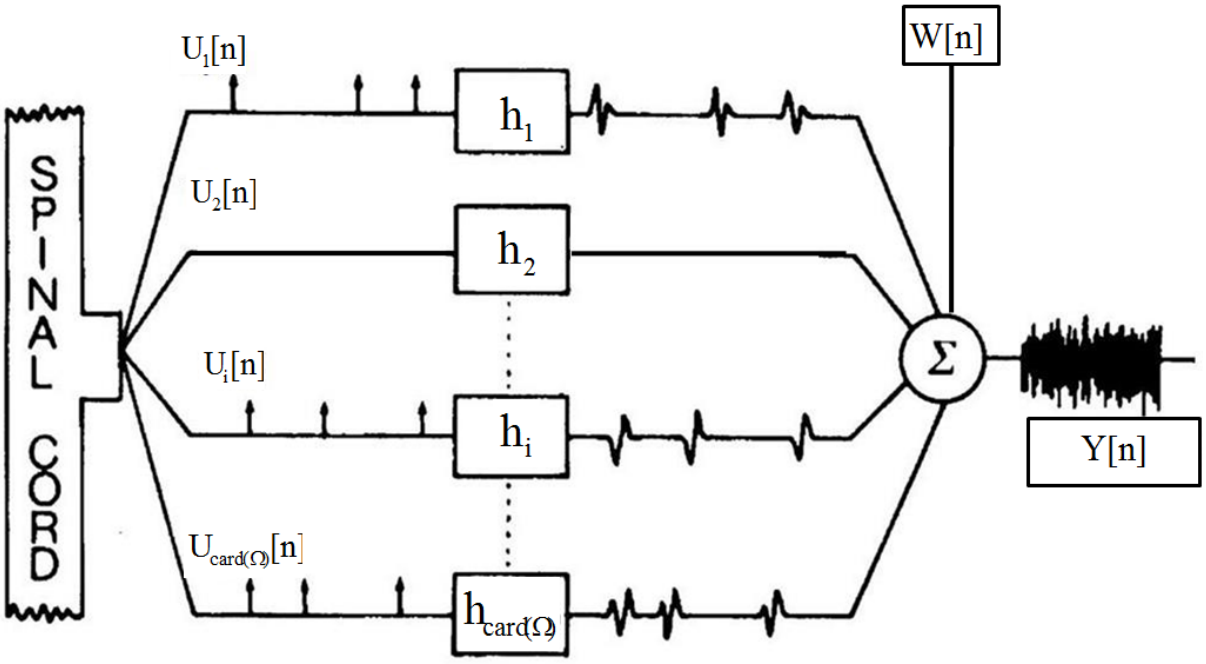


Figure 3.1 – The linear model of iEMG signal: The spike train, denoted by $U_i[n]$, is filtered by the corresponding impulse response h_i. We obtain the MUAP trains. The summation of the active MUAP trains is the iEMG signal, represented by $Y[n]$ in the figure.

Then the decomposition problem can be briefly explained as: Given rough initial forms of impulse responses $h_{i\in\Omega}[0]$, we estimate the set of active MU indexes $A[n]$, the spike train $(U_i[n])_{i\in A[n]}$ and the impulse response $(h_i)_{i\in\Omega}$ with the known observed iEMG signal $Y[n]$ at every time index n. In chapter 4, this decomposition problem will be solved using a Bayes filter built on a HMM [84, 85] derived from this linear convolution model. Moreover, different with the model in [85], a modification of recruitment model is proposed in section 3.4.

3.3 State vector

As the statement in the section 3.2, the spike train $(U_i[n])_{i\in A[n]}$ of the i-th active MU is a binary 0-1 sequence, defined as following:

$$U_i[n] = \begin{cases} 1 & \text{if the } i\text{-th active MU fires} \\ 0 & \text{others} \end{cases} \tag{3.2}$$

Then, we introduce a discrete sawtooth sequence $(T_i[n])_{i\in A[n]}$ related to $(U_i[n])_{i\in A[n]}$, that characterizes the time passed since the previous spike:

$$T_i[n] = \begin{cases} 0 & \text{if } U_i[n] = 1 \\ T_i[n-1] + 1 & \text{if } U_i[n] = 0 \end{cases} \tag{3.3}$$

The distance between two adjacent spikes is termed the inter-spike interval, noted Δ.

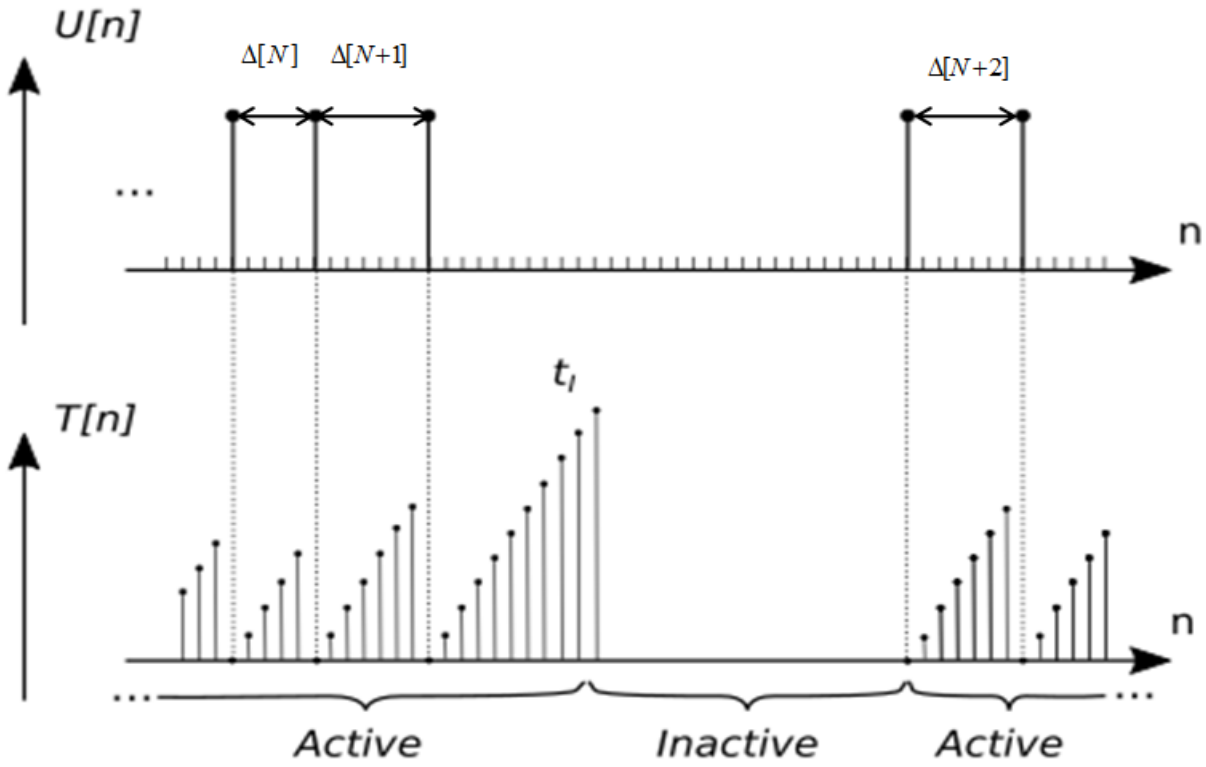


Figure 3.2 – Illustration of the relationship between the spike train $U[n]$ and corresponding sawtooth sequence $T[n]$; Illustration of MU deactivation/activation events. Time between subsequent spikes was shortened for illustration purposes; in reality, it comprises hundreds of time instants. Moreover, Δ represent the length of inter-spike interval and N is its index.

An example is illustrated in the figure 3.2. The state of MU follows an active-inactive-active process. In the active state, the value of $t[n]$ increases with the growing length of inter-spike interval and returns to zero until the next spike.

In addition, we noted that if the i-th MU is inactive, that is $i \notin A[n]$, both $U_i[n]$ and $T_i[n]$ is meaningless, thus not a number. We define the active scenario $S[n]$ as the conjoint vector of $A[n]$ and $(T_i[n])_{i\in A[n]}$.

In the iEMG signal decomposition, one of the most important results is the average number of spikes per time unit $\mathsf{E}\{U_i[n]\}$, termed firing rate or discharge rate, which can be calculated by the inverse of the

expectation of inter-spike interval $\mathsf{E}\{\Delta_i[N]\}$. The inter-spike law parameters $\Theta_i[n]$ are proposed as a parameterized representation of the stochastic sequence $\Delta_i[N]_{N\in\mathbb{Z}}$. This sequence is assumed to be independent and identically distributed. Therefore, $\Theta_i[n]$ is considered as constant over time (its tracking algorithm will be presented in section 4.6):

$$\Theta_i[n] \equiv \Theta_i^* \tag{3.4}$$

$H_i[n]$ is taken as a $1 \times \ell_{\text{IR}}$ vector, containing the possibly non-zero coefficients of h_i, where ℓ_{IR} is the maximum MUAP length. It is firstly supposed to be a constant vector with the formula (its tracking algorithm will be presented in section 4.6):

$$H_i[n] \equiv H_i^* \tag{3.5}$$

Then, the state vectors of HMM are formulated as following:

— $S[n] = (A[n], (T_i[n])_{i\in A[n]})$ the activation scenario is a $(2\times\text{card}(A[n]))$-dimensional column vector, containing the set of active MU indexes and their values of sawtooth sequences, where $\text{card}(A[n])$ denotes the number of active MUs;

— $H[n] = (H_i[n])_{i\in\Omega}$ is a $(\ell_{\text{IR}} \times \text{card}(\Omega))$-dimensional column vector, composed of the coefficients of all MUAP shapes, where $\text{card}(\Omega)$ denotes the number of all MUs in the process;

— $\Theta[n] = (\Theta_i[n])_{i\in\Omega}$ is a $(2 \times \text{card}(\Omega))$-dimensional column vector, made up of the inter-spike law parameters of all MUs.

The Markovian properties of state vectors are explained in the next section 3.4, as well as the dimension of state vector $\Theta[n]$.

3.4 Transition model

Since $H_i[n]$ and $\Theta_i[n]$ are constant vectors, as shown in formulas (3.5) and (3.4), the transition probability distributions for $H[n]$ and $\Theta[n]$ are:

$$\begin{aligned} H_i[n+1] &= H_i[n] \\ \Theta_i[n+1] &= \Theta_i[n] \end{aligned} \tag{3.6}$$

Although state vectors $H[n]$ and $\Theta[n]$ are considered stationary, practically it is never the case. An adaptation to steady changes of these parameters will be introduced later (see section 4.6).

Transition laws of the activation scenario $S[n]$ relies on the recruitment model containing two transition models, respectively related to the two components $A[n]$ and $(T_i[n+1])_{i\in A[n]}$, and the renewal model of spike trains, which takes the regularity of the trains $(T_i[n+1])_{i\in A[n]}$ into account. The recruitment model and the renewal model are respectively presented in the following subsections.

3.4.1 Recruitment model

As it was mentioned above, the regulation of muscle contraction force is achieved by concurrent modulation of MN firing frequencies (rate coding) and recruitment. Recruitment mechanism is introduced into our model using vector $A[n]$ containing the indexes of all MUs that are active at the time instant n. It has the following transition law:

— the probability distribution of $A[n+1]$ given $S[n]$ which is derived from an heuristic recruiting model. With the help of two tuning parameters t_{I} and λ:

  — if some sojourn times $T_i[n]$ reach a maximum time t_{I}, their corresponding indexes will be dropped from $A[n+1]$;

  — otherwise, an inactive motor unit is randomly picked among idle MUs and its index will be added in $A[n+1]$ with probability $\frac{\lambda}{\text{card}(A[n])}$ (that is to say, there is no activation with probability $1-\lambda$), where $\text{card}(A[n])$ denotes the number of active MUs indexes in $A[n]$.

— the probability distribution of $(T_i[n+1])_{i\in A[n+1]}$ given $A[n+1]$, $S[n]$ and $\Theta[n]$, assuming the independence between MUs:
  — if $i \notin A[n+1]$ and $i \in A[n]$ (deactivation), $T_i[n+1]$ is dropped from the state vector;
  — if $i \in A[n+1] \cap A[n]$ (keep activation), the distribution of $T_i[n+1]$ only depends on $T_i[n]$ and $\Theta_i[n]$. $T_i[n+1]|T_i[n], \Theta_i[n]$ is driven by the transition distribution of $(T_i[n+1])_{i\in A[n]}$ in the renewal model, which would be presented in subsection 3.4.2;
  — if $i \in A[n+1]\backslash A[n]$ (activation), $T_i[n+1]$ is inserted in $T[n+1]$ with initial value 0 and the probability of $T_i[n+1]|\Theta_i[n]$ is set to be 1 due to the lack of previous firing time.

The described recruitment model is depicted in figure 3.2. In the lower panel $T_i[n]$, if the $t_i[n]$ reaches the maximum time t_1, the state of MU turn to inactive; if not, we active randomly an inactive MU with with probability $\frac{\lambda}{\text{card}(\bar{A}[n])}$ and $T_i[n+1] = 0$ inserted in the state vector $T[n+1]$.

The recruitment model can be simply expressed by the formula:

$$A[n+1] = \begin{cases} A[n] \setminus i & \text{w.p. 1, for all } T_i[n] = t_I \\ A[n] \cup i & \text{w.p. } \frac{\lambda}{\text{card}(\bar{A}[n])}, \text{ if i} \notin A[n] \\ A[n] & \text{w.p. } 1-\lambda \end{cases} \tag{3.7}$$

where w.p. means 'with the probability' and $\bar{A}[n]$ denotes the set of inactive MU indexes.

3.4.2 Renewal model for active spike trains

The renewal model is proposed for the transition distribution of $(T_i[n+1])_{i\in A[n]}$ with $i \in A[n+1] \cap A[n]$, meaning that the i-th MU keeps activation.

As the assumptions stated in the sections 3.1 and 3.3, the spike trains are mutually independent between MUs and the inter-spike interval sequence $\Delta_i[N]_{N\in\mathbb{Z}}$ of every active MU is independent and identically distributed. And it is evident that $\Delta_i[N]_{N\in\mathbb{Z}}$ is a discrete sequence, as well as $T_i[n]$. Thus, for all active MU, we have the probability of mass function (PMF) of inter-spike intervals $\mathsf{Pr}(\Delta_i = t|\ \Theta_i^*)$, where t is a positive natural number. Its cumulative distribution function (CDF), representing the probability of all inter-spike intervals smaller than t, is:

$$F(t) = \sum_{\tau=1}^{t} \mathsf{Pr}(\Delta_i = \tau|\ \Theta_i^*) \tag{3.8}$$

The reliability function is the complement of CDF, representing the probability of survival with the formula:

$$s(t) = 1 - F(t-1) = \sum_{\tau=t}^{\infty} \mathsf{Pr}(\Delta_i = \tau|\ \Theta_i^*) \tag{3.9}$$

The hazard rate [109] is defined as the event rate at time t conditional on survival until time t or later :

$$r(t) = \mathsf{Pr}(\Delta_i = t|\ \Delta_i \geq t, \Theta_i^*) = \frac{\mathsf{Pr}(\Delta_i = t|\ \Theta_i^*)}{s(t)} \tag{3.10}$$

And it can be characterized by its mean value m:

$$m(\Theta_i^*) = \mathsf{E}(\Delta_i|\ \Theta_i^*)$$

where $\mathsf{E}(.)$ stands for the expectation.

As it shown in [84, 85], the sawtooth sequence $T_i[n]$ is a Markovian chain:

$$\mathsf{Pr}(T[n+1] = t|\ T^n) = \mathsf{Pr}(T[n+1] = t|\ T[n]) \tag{3.11}$$

where the exponent ^{n} means "from 1 to n" (e.g. $Y^n = [Y[1], Y[2], ..., Y[n]]$). Thus, we have the transition distribution, for all $i \in A[n+1] \cap A[n]$:

$$T_i[n+1] = \begin{cases} 0 & \text{w.p. } r(T_i[n]+1, \Theta_i) \\ T_i[n]+1 & \text{w.p. } 1 - r(T_i[n]+1, \Theta_i) \end{cases} \tag{3.12}$$

The invariant distribution is:

$$\mathsf{Pr}(T_i[n] = t|\ \Theta_i[n]) = \frac{s(t+1, \Theta_i[n])}{m(\Theta_i[n])} \tag{3.13}$$

The distribution of inter-spike intervals, obtained from the decomposition of iEMG signals decomposed manually by expert, respects the discrete Weibull distribution type I. Besides, almost all the inter-spikes intervals are more than a known refractory period t_R, which is approximately 30 ms [5, 6]. In the HMM, the refractory period t_R is taken as a known constant, while the other two parameters of the discrete Weibull distribution type I: the location one t_0 and the concentration one β of each MU are components of the state vector Θ. As shown in the figure 3.3, t_0 decides the most probable length of inter-spike intervals and β represents the concentration to t_0.

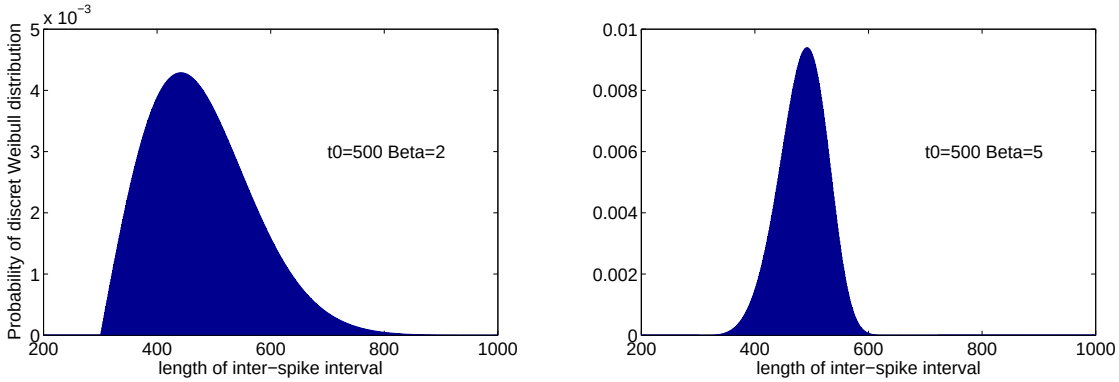


Figure 3.3 – Example of the discrete Weibull distribution type I with different parameters

The discrete Weibull distribution type I[110] for the interspike interval is, for all $t > t_R$:

$$\mathsf{Pr}(\Delta_i = t|\ \Theta_i = (t_0, \beta)) = \exp\left[-\left(\frac{t-1-t_R}{t_0-t_R}\right)^{\beta}\right] - \exp\left[-\left(\frac{t-t_R}{t_0-t_R}\right)^{\beta}\right] \tag{3.14}$$

Then, the formulas of the hazard rate and the reliability function are easy to calculate:

$$\begin{aligned} r(t) &= 1 - \exp[(\frac{t-1-t_R}{t_0-t_R})^{\beta} - (\frac{t-t_R}{t_0-t_R})^{\beta}] \\ s(t) &= \exp[-(\frac{t-1-t_R}{t_0-t_R})^{\beta}] \end{aligned} \tag{3.15}$$

Moreover, an approximation of the firing rates [85] calculated with the parameters of discrete Weibull distribution type I is given:

$$\mathsf{E}(U_i[n]|\ \Theta) \approx \frac{1}{(t_{0i}-t_R)\Gamma(1+\frac{1}{\beta_i}) + t_R} \tag{3.16}$$

3.5 Observation model

The observation equation, which can be directly derived from its linear model (3.1), becomes:

$$Y[n] = \sum_{i \in \Omega} \varphi_i(S[n])\, H_i[n] + W[n] \tag{3.17}$$

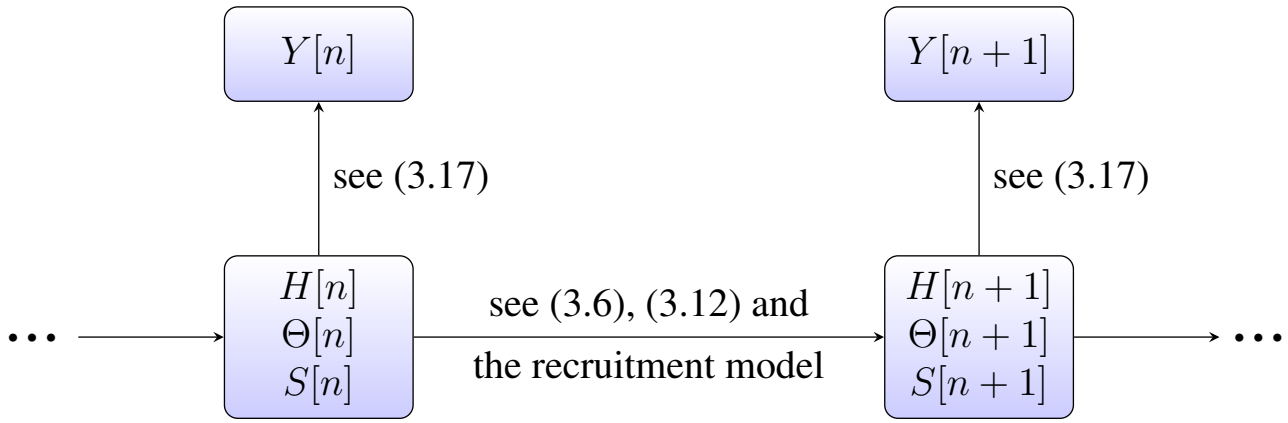


Figure 3.4 – Hidden Markov model

where for all $s = (a, (t_j)_{j\in a})$, $\varphi_i(s)$ is a row vector of size ℓ_{IR} with all components equal to zero, except, if $i \in a$ and $t_i < \ell_{\text{IR}}$, the component in position $t_i + 1$ has value 1. For an active MU $i \in a[n]$, if there is a spike at time index n, that is $u_i[n] = 1$ and $t_i[n] = 0$, the row vector $\varphi_i(s_i[n])$ is a ℓ_{IR} vector filled with zeros, except the first component equal to one. Then, with the growing time index n, the one will propagate through the row vector $\varphi_i(s_i[n])$.

Based on HMM proposed in this chapter, figure 3.4 demonstrates its temporal evolution.

3.6 Discussion and conclusion

The iEMG signal is modelled as a HMM, whose state vectors demonstrate the Markovian property, laying the foundation for the recursive decomposition presented in chapter 4.

We should notice that the discrete Weibull distribution type I is not the one and only describing the distribution of inter-spike intervals. Some others distributions also can make it, but with more complex formulas. Furthermore, with respect to [85], we modified the activation process of recruitment model, which makes the Bayes filter presented in chapter 4 more robust.

4

Bayes filter

4.1 Introduction

In this chapter, a Bayes filter will be proposed in order to estimate recursively the state vector of the HMM based on the propagating posterior probability. The expected value of inter-spike law parameters Θ is approximated using a recursive maximum likelihood estimation. And the impulsed response $H[n]$ can be estimated by the Kalman filter. Moreover, an observation prediction and its variance are also offered. Due to the high time consuming of Kalman filter, a least-mean-square (LMS) filter and a normalised least-mean-square (NLMS) filter were proposed to replace it. Both of them show a good performance on both the quality of estimated impulse responses and the speed of optimisation.

As stated in the section 3.3, both inter-spike law parameters and impulse responses are time varying. Their tracking methods are also introduced. Furthermore, Bayes estimators were applied to calculate the output results, based on several most probable state vectors. Finally, algorithm 1 is given to illustrated the whole decomposition process.

4.2 Principle

The Bayes filter propagates the posterior probability law of the state sequence of an HMM along time. In the sequel, the exponent ^{n} means "from 1 to n" (e.g. $Y^n = [Y[1], Y[2], ..., Y[n]]$), and the exponent $^{|n}$ means "given the data Y^n". For a growing time index n, we have to compute:

— The posterior probability density function (PDF) of $\Theta[n]$ given S^n, Y^n, H. Obviously, except S^n, observation Y^n and impulse response H do not bring any information about $\Theta[n]$. Furthermore, due to MUs independence, its PDF is the product of the PDF of $\Theta_i[n]$ given S^n.

$$\mathsf{Pr}(\Theta[n] \mid S^n, Y^n, H) = \mathsf{Pr}(\Theta[n] \mid S^n) = \prod_{i \in A[n]} \mathsf{Pr}(\Theta_i[n] \mid S^n) \tag{4.1}$$

In the approximate Bayes estimator that we will propose in the following, only the computation of the expected value of Θ_i given S^n is needed. It will be approximated by using a recursive maximum likelihood estimation and noted $\hat{\theta}_{i,S^n}$ (see section 4.3)

— The PDF of $H[n]$ given S^n and Y^n. Thanks to the marginalization principle [111], this PDF is Gaussian and is computed using Kalman filter. The mean and the variance of this PDF will be denoted $\hat{H}_{S^n}^{|n}$ and P_{S^n}. Furthermore, the Kalman filter provides an observation prediction noted as $\hat{Y}_{S^n}^{|n-1}$, as well as its variance noted as v_{S^n} (see subsection 4.4.1).

Due to the big dimension of covariance matrix P_{S^n}, the Kalman filter shows a high time-consuming property. To reduce the complexity of optimization, a LMS filter and a NLMS filter are proposed, which will be introduced in subsections 4.4.2 and 4.4.3.

— The PMF of S^n given Y^n (see subsection 4.5). We should notice that it is impossible to process all possible values of S^n, since their number increases exponentially as the time index n grows.
For example: We suppose that we have two active MUs at time index from 1 to 3, that is $A[n] = \{1, 2\}$, for all $n \in \{1, 2, 3\}$. We initialize the value of $T[1]$ with two random numbers more than the refractory time for the two active MUs, such as: $T[1] = \{612, 534\}$. The refractory time is set to 300. Thus, $S[1] = (\{612, 534\}, \{1, 2\})$. All possible values of S^3 are shown as following:

$$\begin{aligned}
&1.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{613, 535\}, \{1, 2\}), S[3] = (\{614, 536\}, \{1, 2\}) \\
&2.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{613, 535\}, \{1, 2\}), S[3] = (\{0, 536\}, \{1, 2\}) \\
&3.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{613, 535\}, \{1, 2\}), S[3] = (\{614, 0\}, \{1, 2\}) \\
&4.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{0, 535\}, \{1, 2\}), S[3] = (\{1, 536\}, \{1, 2\}) \\
&5.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{0, 535\}, \{1, 2\}), S[3] = (\{1, 0\}, \{1, 2\}) \\
&6.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{613, 0\}, \{1, 2\}), S[3] = (\{614, 1\}, \{1, 2\}) \\
&7.\ S[1] = (\{612, 534\}, \{1, 2\}), S[2] = (\{613, 0\}, \{1, 2\}), S[3] = (\{0, 1\}, \{1, 2\})
\end{aligned} \tag{4.2}$$

In this case of constant $A[n]$ and two active MUs, we have seven possible values of S^3. If we take the variation of $A[n]$ with increasing n, there will be more possible values of S^n.
Due to the limitation of computation power and memory of our machine, we cannot process such great many possible values of S^n. The n_{path} most probable paths are kept at every time index, where n_{path} is chosen according to the available computation power.

4.3 Estimation of inter-spike law parameters

To estimate the inter-spike law parameters (parameters of the discrete Weibull distribution), a recursive maximum likelihood (RML) estimator is implemented. The likelihood is optimized iteratively by the quasi-Newton method. The ML of $\hat{\theta}_{i,S^n}$ is:

$$\hat{\theta}_{i,S^n} = \arg\min_{\theta} \underbrace{-\frac{1}{n} \ln \mathsf{Pr}(S^n = s^n | \ \Theta_i = \theta_i)}_{J_{i,S^n}(\theta)} \tag{4.3}$$

After the optimization, we have the following results.

For all $n \geq 1$, if $i \in A[n] \cap A[n-1]$, we define an active time index τ:

$$\tau_{i,S^n} = \begin{cases} \tau_{i,S^{n-1}} + 1 & \text{if } i \in A[n] \\ \tau_{i,S^{n-1}} & \text{if } i \notin A[n] \end{cases} \tag{4.4}$$

Then, for the estimation of an i-th MU inter-spike law parameters vector θ, we have:

$$\hat{\theta}_{i,S^n} = \hat{\theta}_{i,S^{n-1}} - \frac{1}{\tau_{i,S^n}} G^{-1}_{i,S^n} Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}}) \tag{4.5}$$

$$\begin{aligned} G_{i,S^n} =& \frac{1}{\tau_{i,S^n}} [Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})][Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})]^\top + \\ & (1 - \frac{1}{\tau_{i,S^n}}) \, G_{i,S^{n-1}} \end{aligned} \tag{4.6}$$

Where G_{i,S^n} is the approximate Hessian matrix of the maximum likelihood criterion $Q_{i,S^n}(\theta)$ at the current estimate, and $Q'_{i,S^n}(\theta)$ is its gradient with:

$$Q_{i,S^n}(\theta) = \begin{cases} -\ln r(t_i[n]+1,\theta) & \text{if } t_i[n+1]=0 \\ -\ln\left(1-r(t_i[n]+1,\theta)\right) & \text{if } t_i[n+1]=t_i[n]+1 \end{cases}$$

If $i \notin A[n] \cap A[n-1]$, we have:

$$\begin{cases} \hat{\theta}_{i,S^n} = \hat{\theta}_{i,S^{n-1}} \\ G_{i,S^n} = G_{i,S^{n-1}} \end{cases} \tag{4.7}$$

We would like to take the estimated inter-spike law parameters of the last active period as the initial value of the next active period.

The justification of estimation procedure is shown in the appendix 9.1.

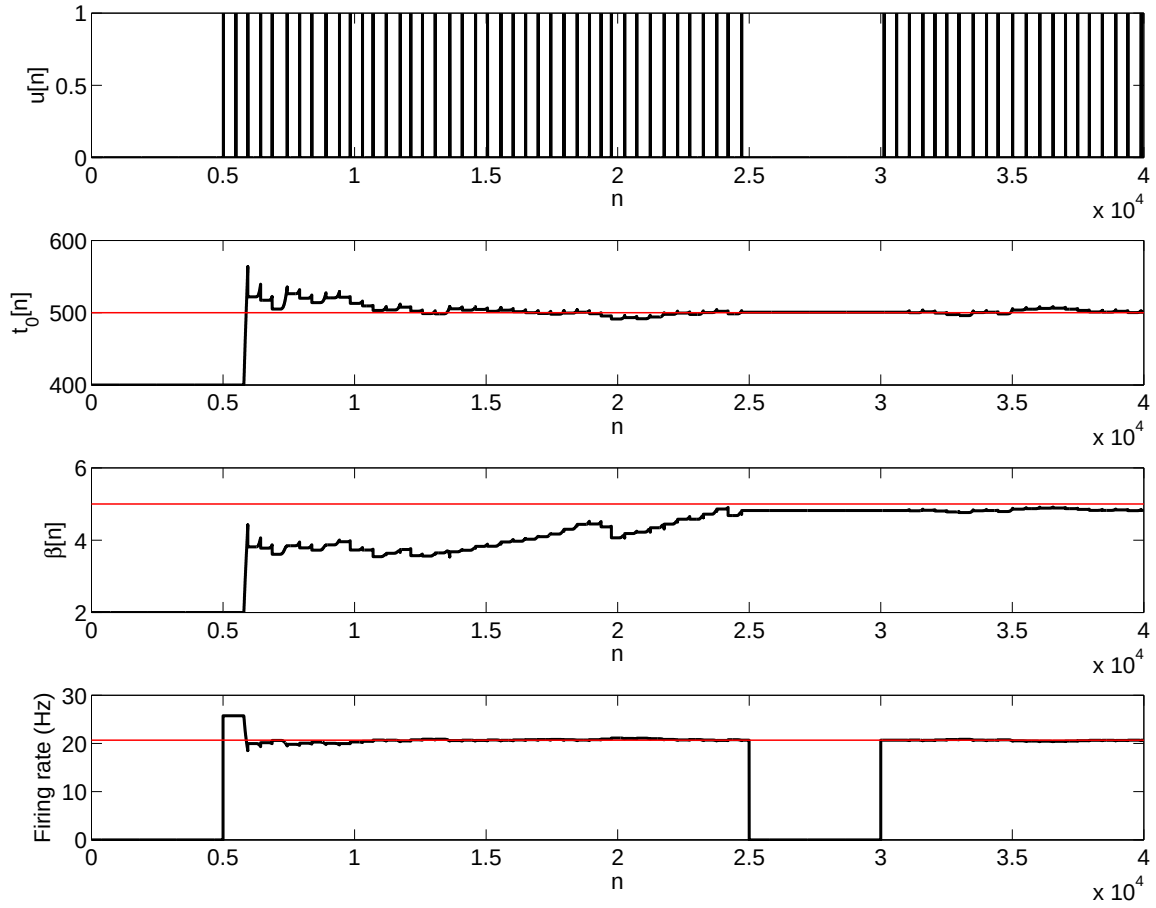


Figure 4.1 – Inter-spike law parameters estimation: The first line is the given spike train $U[n]$; The second line is the estimated location parameter sequence of the discrete Weibull distribution type I; The third line is the estimated concentration one; The forth line is the firing rate. The red horizontal straight line denotes the real value, while the black oscillating line represents the estimated value.

An example of inter-spike law parameters estimation is given in figure 4.1. In the first line, the spike train $u[n]$ is generated with the discrete Weibull distribution type I parameters: $t_0 = 500$ and $\beta = 5$. There are two active periods in this processus. Based on the spike train $u[n]$, we can estimate the parameters

$\hat{\theta}_{S^n} = (t_0[n], \beta[n])$ using the formula (4.5) and (4.7). Then, the firing rate at every time index n can be calculated with the formula (3.16). As depicted in figure 4.1, the estimated inter-spike law parameters of the first active period is taken as the initial value of the second active period, which accelerates the convergence.

4.4 Estimation of impulse responses

The estimation of impulse response is modelled as the optimisation of linear time-invariant systems. Due to its excellent performance, Kalman filter is usually used to resolve this problem [112]. However, its high complexity requires more time. In order to accelerate the estimation procedure, we propose two less time consuming filters: LMS filter and NLMS filter. All of these filters will be presented in the following parts of this section.

4.4.1 Kalman filter

Given S^n, the Markov model for impulse responses reduces, for all $n \geq 1$:

$$\begin{cases} H[n+1] = H[n] \\ Y[n] = \sum_{i \in \Omega} \varphi_i(S[n])\, H_i[n] + W[n] \end{cases} \tag{4.8}$$

Where $W[n]$ assumed to be an independent and identically distributed Gaussian white noise with variance v. $H[1]$ and $H[n]|\ S^n, Y^n$ are Gaussian with means $\hat{H}^{|n}_{S^n}$ and covariance matrices P_{S^n}. $Y[n]|\ S^n, Y^{n-1}$ is Gaussian with mean $\hat{Y}^{|n-1}_{S^n}$ and variance v_{S^n}. These means and variances can be estimated recursively by Kalman filter. With the initial prior $\hat{H}^{|0}_{S^0}$ and P_{S^0}, we have, for all $n \geq 1$:

— Prediction of observation:

$$\begin{aligned} \hat{Y}^{|n-1}_{S^n} &= \psi(S[n])\ \hat{H}^{|n-1}_{S^{n-1}} \\ v_{S^n} &= \psi(S[n])\ P_{S^{n-1}}\ \psi(S[n])^\top + v \end{aligned} \tag{4.9}$$

— Estimation of state:

$$\begin{aligned} K_{S^n} &= P_{S^{n-1}}\ \psi(S[n])^\top\ v^{-1}_{S^n} \\ \hat{H}^{|n}_{S^n} &= \hat{H}^{|n-1}_{S^{n-1}} + K_{S^n}\ (Y[n] - \hat{Y}^{|n-1}_{S^n}) \\ P_{S^n} &= P_{S^{n-1}} - K_{S^n}\ v_{S^n}\ K^\top_{S^n} \end{aligned} \tag{4.10}$$

Where K_{S^n} is the Kalman gain and $\psi(s) = [\varphi_1(s), ..., \varphi_{\text{card}(a[n])}(s)]$.

Moreover, the variance v of noise is unknown. An heuristic approach is proposed to estimate it with the square of the residual $Y[n] - \psi(S[n])\hat{H}^{|n}_{S^n}$.

$$\hat{V}^{|n}_{S^n} = (1 - \frac{1}{n})\hat{V}^{|n-1}_{S^{n-1}} + \frac{1}{n}(Y[n] - \psi(S[n])\ \hat{H}^{|n}_{S^n})^2 \tag{4.11}$$

And the global estimation is:

$$\hat{V}^{|n} = \sum_{S^n} \hat{V}^{|n}_{S^n}\ \mathsf{Pr}^{|n}(S^n = s^n) \tag{4.12}$$

Where $\hat{V}^{|n}$ replaces v in the formula (4.9).

4.4.2 Least mean square filter

In the Kalman filter, the dimension of covariance matrix of impulse responses P_{S^n} is $O((\text{card}(\Omega) \times \ell_{\text{IR}}) \times (\text{card}(\Omega) \times \ell_{\text{IR}}))$, where $\text{card}(\Omega)$ denotes the number of MUs, including both active and inactive

MUs. Thus, such a big dimension of covariance matrix requires a large computational power. The LMS filter is proposed to replace the Kalman filter to accelerate the estimation. The transition from Kalman filter to the LMS filter is justified in the appendix 9.2.

According to the Kalman filter, we obtain:

$$\begin{aligned} P_{S^n} &= \frac{\hat{V}^{|n}}{n} R_{S^n}^{-1} \\ R_{S^n} &= \frac{1}{n} \sum_{k=1}^{n} \psi(S[k])^\top \psi(S[n]) \end{aligned} \tag{4.13}$$

In the LMS filter, the matrix R_{S^n} is approximated by a diagonal matrix that comprises the firing rates of every motor unit, which greatly simplifies the procedure of impulse responses estimation. With the rough initial prior $\hat{H}_{S^0}^{|0}$, for all $n \geq 1$, we have the formula of the LMS filter:

$$\begin{aligned} \epsilon[n] &= Y[n] - \psi(S[n])\ \hat{H}_{S^{n-1}}^{|n-1} \\ m_{\Delta,i}[n] &= \frac{\sum_j \Delta_i[j]}{\text{card}(\Delta_i)} \\ \tilde{v}[n] &= 1 + \sum_i m_{\Delta,i}[n]\ \varphi_i(S[n])\varphi_i(S[n])^\top \\ \hat{H}_{i,S^n}^{|n} &= \hat{H}_{i,S^{n-1}}^{|n-1} + \frac{m_{\Delta,i}[n]\varphi_i(S[n])\epsilon[n]}{n\ \tilde{v}[n]} \end{aligned} \tag{4.14}$$

Where $\Delta_i[j]$ denotes the j-th inter-spike interval of the i-th motor units; $\text{card}(\Delta_i)$ is the number of inter-spike intervals of the i-th motor units; $m_{\Delta,i}[n]$ is the expectation value of the inter-spike intervals of the i-th motor unit at the time index n; and $\tilde{v}[n]$ represents the ratio of the variance of innovation v_{S^n} to the variance of noise $\hat{V}^{|n}$.

The prediction of observation $\hat{Y}_{S^n}^{|n-1}$ is the same as the formula (4.9) and the prediction of the variance of innovation v_{S^n} is:

$$v_{S^n} = \tilde{v}[n]\ \hat{V}^{|n}. \tag{4.15}$$

4.4.3 Normalized least mean square filter

A NLMS filter is also proposed to reduce the processing time of impulse responses estimation. Although it is a little more complex than the LMS filter. Compared to the Kalman filter, it shows an impressive acceleration and a competitive performance.

First of all, we define the set $F[n]$ containing indexes of MUs that exhibit their MUAP shapes at time index n. An example is given in figure 4.2. In this example, part of two MUAP waveforms is superimposed. $F[n]$ equalling to the empty set, the set containing one index or two indexes, depends on the number of MUAPs at time index n. Evidently, $F[n] \subseteq A[n]$.

Then, the impulse responses optimization problem is described as following:

$$\begin{aligned} H_i[n] &= H_i[n-1] + W_{H_i}[n],\ \ \forall i \in \Omega \\ Y[n] &= \sum_{i \in \Omega} \varphi_i(S[n])\ H_i[n] + W[n] \end{aligned} \tag{4.16}$$

As known, the impulse responses are time varying. We assume that $H_i[n]$ is a random vector, where i denotes the index of MU. A sight change $W_{H_i}[n]$, which is a zero-mean white Gaussian noise signal vector, occurs at every time index. $W_{H_i}[n]$ is uncorrelated with $H_i[n-1]$ and its correlation matrix is assumed to be $R_{W_{H_i}} = \sigma_{W_{H_i}}^2\ I_{\ell_{\text{IR}}}$, where $I_{\ell_{\text{IR}}}$ is a $\ell_{\text{IR}} \times \ell_{\text{IR}}$ identity matrix. Furthermore, to simplify the formula, we define $X_i^\top[n] = \varphi_i(S[n])$ in this subsection.

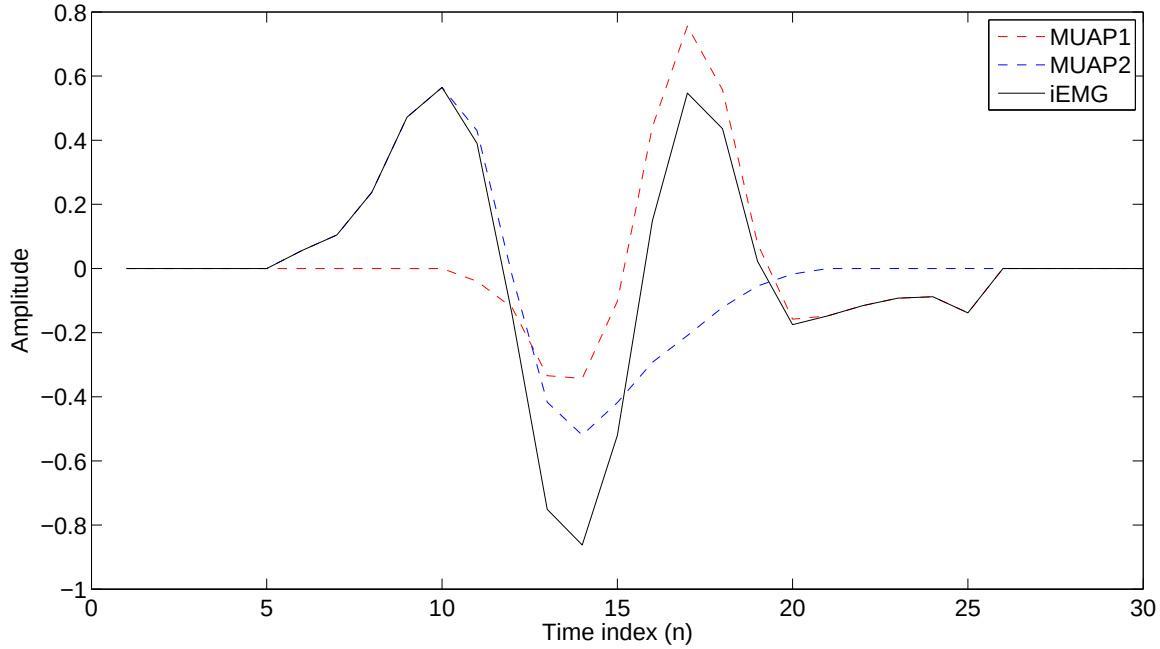


Figure 4.2 – Interpretation of $F[n]$: Red and blue dashed lines represent respectively two MUAP waveforms. Part of them is superimposed (from $n = 10$ to $n = 21$). Black line is the simulated iEMG signal, which is the temporal summation of the two MUAP waveforms. If $n < 5$ or $n \geq 26$, $F[n]$ is a empty set; If $5 \leq n < 10$, $F[n] = \{1\}$; If $10 \leq n < 21$, $F[n] = \{1, 2\}$; If $21 \leq n < 26$, $F[n] = \{2\}$.

If $i \notin F[n]$, we do not update their impulse responses:

$$\hat{H}^{|n}_{i,S^n} = \hat{H}^{|n-1}_{i,S^{n-1}} \tag{4.17}$$

If $i \in F[n]$, we define the conventional NLMS update formula [113]:

$$\hat{H}^{|n}_{F[n],S^n} = \hat{H}^{|n-1}_{F[n-1],S^{n-1}} + \frac{\mu[n]}{X^{\top}_{F[n]}[n]X_{F[n]}[n] + \delta} X_{F[n]}[n]e[n] \tag{4.18}$$

where $\mu[n]$ is the normalized step-size parameter, δ is the regularization term which prevents denominator to be too small, $X_{F[N]}[n]$ is a card$(F[n]) \times \ell_{\text{IR}}$ dimensional vector comprising all $(X_i[n])_{i \in F[n]}$, the estimated impulse response defined as $\hat{H}^{|n}_{F[n],S^n}$, is a card$(F[n]) \times \ell_{\text{IR}}$ dimensional vector comprising all $(\hat{H}^{|n-1}_{i,S^{n-1}})_{i \in F[n]}$, and $e[n]$ is the a priori error of this adaptive filter:

$$e[n] = y[n] - X^{\top}_{F[n]}[n]\hat{H}^{|n-1}_{F[n-1],S^{n-1}} \tag{4.19}$$

The $\mu[n]$ and δ is two variables to determine. We define $z_{F[n]}[n] = H_{F[n]}[n] - \hat{H}^{|n}_{F[n],S^n}$. We proposed the objective function as following:

$$(\mu[n], \delta) = \underset{\mu[n],\delta}{\arg\min}\, \mathsf{E}(\|z_{F[n]}[n]\|^2_2) \tag{4.20}$$

Based on this function, with the derivation procedure justified in the appendix 9.3, we obtain a recursive

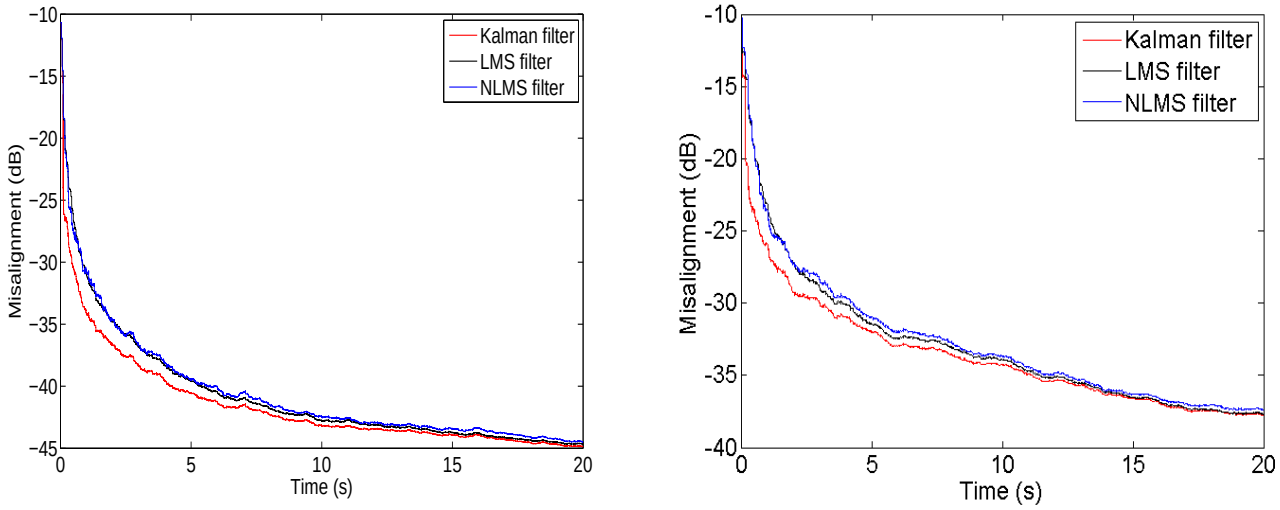


Figure 4.3 – Comparison of MUAP shapes estimation using three different filters: Kalman filter, LMS filter and NLMS filter. SNR=20 dB (left); SNR=10 dB (right). Misalignment (in dB) is defined as $20\log_{10}(\frac{\|H[n]-\hat{H}^{|n}_{S^n}\|_2}{\|H[n]\|_2})$.

optimization formula:

$$\begin{aligned}
e[n] =& y[n] - X^{\top}_{F[n]}[n]\hat{H}^{|n-1}_{F[n-1],S^{n-1}} \\
\hat{H}^{|n}_{F[n],S^n} =& \hat{H}^{|n-1}_{F[n-1],S^{n-1}} + \frac{\mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell_{\text{IR}}\sigma^2_{W_{H_{F[n]}}}}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma^2_{X_{F[n]}}(\sigma^2_{W_{H_{F[n]}}} + \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2))} X_{F[n]}[n]e[n] \\
\mathsf{E}(\|z_{F[n]}[n]\|_2^2) =& \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell_{\text{IR}}\sigma^2_{W_{H_{F[n]}}} \\
& - \frac{\sigma^2_{X_{F[n]}}(\ell_{\text{IR}}\sigma^2_{W_{H_{F[n]}}} + \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2)^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma^2_{X_{F[n]}}\mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell^2_{\text{IR}}\sigma^2_{X_{F[n]}}\sigma^2_{W_{H_{F[n]}}}}
\end{aligned} \tag{4.21}$$

where $\sigma^2_{X_{F[n]}}$ is the variance of $X_{F[n]}$.

In a general manner [114], the variance $\sigma^2[n]$, such as: $\sigma^2_{X_i}[n]$, $\sigma^2_{W_{H_i}}[n]$ and $\hat{V}^{|n}[n]$, denote the power of sequence, and can be updated by the smooth factor:

$$\sigma^2[n] = \lambda_\sigma\sigma^2[n-1] + (1-\lambda_\sigma)\sigma^2[n] \tag{4.22}$$

where $\lambda_\sigma = 1 - \frac{1}{K\ell_{\text{IR}}}$, with $K > 1$.

As mentioned before, the complexity of Kalman filter is the highest among the three filters, much higher than the others, while the LMS filter has the least algebraic operations. Thus, the LMS filter has the fastest processing speed.

Performances of the Kalman filter, the LMS filter and the NLMS filter were evaluated with simulations. A 20 s simulated signal of 5 motor units was generated by the HMM model with the constant impulse responses $H[n]$. Given the scenario S^n and rough initial impulse responses $\hat{H}^{|0}_{S^0}$, the three filters were respectively used to identify $H[n]$. The measure of performance was the normalized misalignment (in dB), defined as $20\log_{10}[\|H[n] - \hat{H}^{|n}_{S^n}\|_2/\|H[n]\|_2]$. Figure 4.3 shows the misalignment of the three filter algorithms. We have studied the performance of three filter for the signals with different signal-to-noise ratio (SNR). As shown in 4.3, the left one is 20 dB, while the right one is 10 dB. For the optimization of MUAP shapes in the same signal, filters have almost the same performance. The Kalman filter has a slight advantage on the convergence rate, which is negligible compared to its high time consuming.

Therefore, considering the processing speed and the performance, the LMS filter was selected to estimate the impulse responses.

4.5 Posterior probability of scenario

For each active MU i, the bifurcation of the sawtooth sequence are $t_i^{n+1} = \{t_i^n, t_i[n]+1\}$ or $t_i^{n+1} = \{t_i^n, 0\}$, if $t_i^n > t_R$; The sawtooth sequence is $t_i^{n+1} = \{t_i^n, t_i[n]+1\}$ if $t_i^n \leq t_R$. Therefore, the total number of all bifurcation sequences, as well as the number of posterior probabilities of scenario, varies from 1 to $2^{\text{card}(A[n+1])}$ at the time index n, where card$(A[n+1])$ denotes the number of elements in the $A[n+1]$.

The posterior probability recursion is derived by means of an update-prediction scheme. According to the Bayes rule, for all possible realizations s^n of S^n, the update step is:

$$\mathsf{Pr}^{|n}(S^n = s^n) \propto \mathsf{Pr}^{|n-1}(S^n = s^n)\; g(Y[n] - \hat{Y}_{s^n}^{|n-1}, v_{s^n}) \tag{4.23}$$

Where $g(.\,, v)$ is the zero-mean and variance v Gaussian PDF. The prediction step is:

$$\begin{aligned}\mathsf{Pr}^{|n}(S^{n+1} = s^{n+1}) =& \mathsf{Pr}^{|n}(S^n = s^n) \times \mathsf{Pr}(A[n+1] = a[n+1] \mid S[n] = s[n]) \times \\ & \prod_{i \in A[n+1]} \mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n = s^n)\end{aligned} \tag{4.24}$$

Where $\mathsf{Pr}(A[n+1] = a[n+1] \mid S[n])$ is the transition probability of the recruitment model. The elements $\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n)$, for all $i \in A[n+1]$, are differently calculated:

If $i \in A[n+1] \cap A[n]$, meaning that the MU is already active, using the total probability formula and the Markov assumption, we have:

$$\begin{aligned}\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n) &= \mathsf{E}(\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid \Theta_i[n], S^n) \mid S^n) \\ &= \mathsf{E}(\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid T_i[n], \Theta_i[n]) \mid S^n)\end{aligned} \tag{4.25}$$

With regard of the transition law of sawtooth sequence, we have:

$$\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n) = \begin{cases} \mathsf{E}(r(T_i[n]+1, \Theta_i[n]) \mid S^n) & \text{if } t_i[n+1] = 0 \\ 1 - \mathsf{E}(r(T_i[n]+1, \Theta_i[n]) \mid S^n) & \text{if } t_i[n+1] = T_i[n]+1 \\ 0 & \text{otherwise} \end{cases} \tag{4.26}$$

We make a strong assumption that the expected hazard rate of the random parameter is the hazard rate of the expected parameter.

$$\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n) \approx \begin{cases} r(T_i[n]+1, \mathsf{E}(\Theta_i[n+1] \mid S^n)) & \text{if } t_i[n+1] = 0 \\ 1 - r(T_i[n]+1, \mathsf{E}(\Theta_i[n+1] \mid S^n)) & \text{if } t_i[n+1] = T_i[n]+1 \\ 0 & \text{otherwise} \end{cases} \tag{4.27}$$

And the expected parameter is approximated by its RML estimate $\hat{\theta}_{i,S^n}$ provided in section 4.3.

$$\mathsf{Pr}(T_i[n+1] = t_i[n+1] \mid S^n) \approx \begin{cases} r(T_i[n]+1, \hat{\theta}_{i,S^n}) & \text{if } t_i[n+1] = 0 \\ 1 - r(T_i[n]+1, \hat{\theta}_{i,S^n}) & \text{if } t_i[n+1] = T_i[n]+1 \\ 0 & \text{otherwise} \end{cases} \tag{4.28}$$

If $i \in A[n+1] \backslash A[n]$, meaning that the MU is activated at the time index $n+1$ with $T_i[n+1] = 0$ (as presented in subsection 3.4.1), the transition law $T_i[n+1] \mid \Theta_i$ is not influenced by the inter-spike distribution parameters due to the lack of the information about them. Thus, we set:

$$\mathsf{Pr}(T_i[n+1] = 0 \mid S^n) = 1 \tag{4.29}$$

4.6 Tracking

The inter-spike laws parameters and the impulse responses are known to be time varying. To set the adaptivity of Bayes filter, we introduce the window length sequence $\ell[n]$ growing from 1 to the maximum window length ℓ_∞ with the recursion formula [115]:

$$\begin{cases} \ell[1] = 1 \\ \ell[n+1] = (1 - \frac{1}{\ell_\infty})\, \ell[n] + 1 \end{cases} \tag{4.30}$$

Where the maximum window length ℓ_∞ is related to the desired adaptivity. If $\ell_\infty = +\infty$, there is no tracking.

The formula of the estimated impulse response (4.14) becomes:

$$\hat{H}^{|n}_{i,S^n} = (1 - \frac{1}{\ell[n]})\hat{H}^{|n-1}_{i,S^{n-1}} + \frac{m_{\Delta,i}[n]\varphi_i(S[n])\epsilon[n]}{\ell[n]\ \tilde{v}[n]} \tag{4.31}$$

With the adaptivity coefficients, the formula of the estimated variance of noise (4.11) is rewritten as:

$$\hat{V}^{|n}_{S^n} = (1 - \frac{1}{\ell[n]})\hat{V}^{|n-1}_{S^{n-1}} + \frac{1}{\ell[n]}(Y[n] - \psi(S[n])\ \hat{H}^{|n}_{S^n})^2 \tag{4.32}$$

For the inter-spike law parameters estimation, considering the activation-inactivation of each MU, the window length sequence is slightly different:

$$\tau_{i,S^n} = \begin{cases} (1 - \frac{1}{\ell_\infty})\ \tau_{i,S^{n-1}} + 1 & \text{if } i \in A[n] \\ \tau_{i,S^{n-1}} & \text{if } i \notin A[n] \end{cases} \tag{4.33}$$

We replace the active index (4.4) by the adaptive formula (4.33).

4.7 Bayes estimator

During the decomposition, at every time index the algorithm chooses n_{path} most probable paths and discards the rest. The probabilities of the retained paths are used as weights to derive the Bayes state estimators, that is:

— A marginal maximum a posteriori (MAP) estimator for the sawtooth sequences and the set of active MUs. The most probable path provides the output spike trains:

$$\hat{S}^{n|n} = \arg\max_{s^n} \mathsf{Pr}^{|n}(S^n = s^n) \tag{4.34}$$

— The minimum mean square error estimator for the impulse responses and the interspike law parameters. Using the total expectation formula:

$$\begin{aligned} \hat{H}^{|n} =& \mathsf{E}^{|n}(H) = \mathsf{E}^{|n}(\mathsf{E}^{|n}(H \mid S^n)) \\ =& \sum_{s^n} \underbrace{\mathsf{E}^{|n}(H \mid S^n = s^n)}_{\hat{H}^{|n}_{s^n}}\ \mathsf{Pr}^{|n}(S^n = s^n) \end{aligned} \tag{4.35}$$

$$\hat{\Theta}^{|n}_i = \sum_{s^n} \underbrace{\mathsf{E}^{|n}(\Theta_i \mid S^n = s^n)}_{\hat{\theta}_{i,s^n}}\ \mathsf{Pr}^{|n}(S^n = s^n) \tag{4.36}$$

4.8 Initialisation

At the beginning of the decomposition, we assume that there is no active MUs. Therefore, the set of active MUs indexes $A[1]$ and the sawtooth sequence $T[1]$ are empty. Initial rough estimates of impulse responses $\hat{H}_{S^1}^{|0}$ are manually or automatically extracted using other techniques, i.e proposed in [57, 53, 116]. Expectations of the inter-spike intervals $m_{\Delta,i}[n]$, for all $i \in \Omega$, are initialized as $3t_R$. An initial estimation of the noise variance $\hat{V}_{s^0}^{|0}$ is made using a signal extract containing no spikes.

The initial inter-spike interval distribution law parameters of active MUs $\hat{\theta}_{i,S^0}$ are composed of t_0 (typically $3t_R \sim 4t_R$) and β (typically $2 \sim 4$) according to the our experience.

Finally, n_{path} initial S^1 are weighted with the same initial probability $\mathsf{Pr}^{|0}(S^1 = s^1)$.

4.9 Algorithm

An outline of the proposed decomposition method is presented in Algorithm 1.

4.10 Discussion and conclusion

The iEMG signal modelled by the HMM, is decomposed by the Bayes filter with the propagating posterior probability. The sequential recursive manner of decomposition reveals a potential of on-line decomposition. However, similar to the algorithm of [67] presented in subsection 2.3.2, this algorithm shows a property of high time-consuming. Although the Kalman filter with big complexity is replaced by the LMS filter, the real time decomposition criterion still cannot be fulfilled in a serial way. Our algorithm shows a parallel structure, which can be accelerated in the parallel environment. Some techniques to simplify the decomposition process and the parallel implementation is introduced in chapter 5.

Algorithm 1 Full estimation process

Initialize $\hat{H}^{|0}_{s^0}$, $m_\Delta[0]$, $\hat{V}^{|0}_{s^0}$, $\hat{\theta}_{i,s^0}$, G_{i,s^0} and s^1
for all initial $s[1]$ **do**
  Initialize $\mathsf{Pr}^{|0}(S^1 = s^1)$
  Predict $\hat{Y}^{|0}_{s^1}$ and v_{s^1} with (4.9)
end for
for all $n \geq 1$ **do**
  new data $Y[n]$
  for all s^n **do**
    Compute the posterior $\mathsf{Pr}^{|n}(S^n = s^n)$ with (4.23)
  end for
  Select and keep the n_{path} most probable paths
  for all s^n **do**
    Update $\hat{H}^{|n}_{s^n}$ with (4.14) (corrected by (4.31))
    Update $\hat{V}^{|n}_{s^n}$ with (4.11) and (4.32)
  end for
  for all s^n **do**
    if source $i \notin a[n] \cap a[n-1]$ **then**
      Update $\hat{\theta}_{i,s^n}$ and G_{i,s^n} with (4.7)
    else
      Update $\hat{\theta}_{i,s^n}$ and G_{i,s^n} with (4.5) and (4.6)
    end if
  end for
  Compute $\hat{H}^{|n}$, $\hat{\Theta}^{|n}_i$ and $\hat{V}^{|n}$ with (4.35), (4.36), and (4.12)
  for all s^n **do**
    if $t_i[n] = t_I$ **then**
      Drop $t_i[n]$ from $t[n]$ and update $a[n+1]$
    else
      Draw α from a uniform distribution $[0, 1]$
      if $\alpha < \lambda$ **then**
        Activate a random source from $\bar{a}[n]$
        Initialize its value in $t[n+1]$ and update $a[n+1]$
      end if
    end if
    Compute $\mathsf{Pr}(A[n+1] = a[n+1] \mid S[n] = s[n])$
  end for
  for all kept (t^n, a^{n+1}) **do**
    for all possible forks t^{n+1} from t^n **do**
      Compute $\mathsf{Pr}^{|n}(S^{n+1} = s^{n+1})$ with (4.24) and (4.27)
    end for
    Predict $\hat{Y}^{|n}_{s^{n+1}}$ and $v_{s^{n+1}}$ with (4.9), (4.14) and (4.15)
  end for
end for

5

Acceleration of decomposition

5.1 Introduction

We model the iEMG signal by the HMM presented in chapter 3 and decompose it with the Bayes filter introduced in chapter 4. The proposed algorithm shows a good performance, which will be demonstrated in 7.2. However, the high accuracy of decomposition by this algorithm requires a large number of kept paths, meaning tremendous calculated quantities and time-consuming. Sometimes, we have to spend several hours, even more than one day, to decompose a complex signal, despite in a fully automatic way.

In this chapter, we will accelerate the decomposition velocity by simplifying the algorithm in order to reduce the calculated quantities and implementing the algorithm into GPU computing environment. Firstly, some heuristic operations are proposed to reduce the processed signal and the bifurcated paths. Then, we establish a parallel computing model of our algorithm: we analyze the parallelism of our algorithm in terms of data parallelism and task parallelism; some special tasks are taken into consideration in a parallel computing manner to save the processing time; we show the outline of parallel structure of our algorithm.

5.2 Path pruning

As it was previously shown in section 4, the number of possible scenarios for S^n grows exponentially with time. Thus, an exhaustive search for the optimal scenario is not possible. In this section we present several means of discarding unnecessary or inconsistent scenarios to avoid the exhaustive search [1].

5.2.1 Limiting the number of kept paths

As stated in section 4.2, limiting the number of kept paths is a conventional operation to reduce the calculated quantities. At each step n, the algorithm generates all possible evolutions of $S[n]$ by bifurcating all paths since the previous step $n-1$ according to the transition law (3.12) and the recruitment model presented in subsection 3.4.1. Then, in order to prevent the exponential growth in number of paths, it chooses only n_{path} of them that have the largest values of posterior probability (4.23). The value of n_{path} should be chosen as a trade-off between the computational complexity and the sub-optimality of the solution.

1. I want to state that the original idea of subsections 5.2.2 and 5.2.3 was proposed by my colleague Konstantin Akhmadeev.

5.2.2 Pruning based on activity detection

An iEMG signal, especially during low-force contractions, comprises the single or superimposed MUAP segments and a great many long segments containing only noise. Therefore, there is no need to make the bifurcations of S^n in the segments containing only noise.

The measure to avoid the bifurcation of S^n in the segments containing only noise is the same as the signal segmentation, one of the procedures of the iEMG signal decomposition with the pattern recognition methods, presented in subsection 2.3.2. Their objective is to separate the noise and the single or superimposed MUAP segments. The local peaks of signals exceeding a threshold, which crosses the whole signal, are taken as the candidate segmentation positions.

Thus, in our sequential decomposition algorithm, the segment from $n+1$ to $n+l_{pd}$ containing the MUAP activities, where l_{pd} denotes the length of pre-detection, is determined by the pre-detection observation $Y[n+1:n+l_{pd}]$. l_{pd} is often related to the maximum MUAPs length ℓ_{IR} and is typically set to $\frac{\ell_{\text{IR}}}{2}$. If a point of discrete sequence $Y[n+1:n+l_{pd}]$ is more than the threshold, S^n bifurcates; otherwise, S^n does not bifurcate. We introduce $Z[n]$ which represents the output of an pre-detection function $z\left(Y[n+1:n+l_{pd}]\right)$ that returns "1" if it detects the presence of MUAPs in the upcoming signal and "0" if does not. If $Z[n]=1$, meaning that there is probably a spike, S^n bifurcates; otherwise, S^n does not bifurcate. Moreover, one thing should be noticed is that if there is a MUAP in the segment, the bifurcation of S^n always takes place in the point of beginning of MUAP. In the end parts of MUAP segmentation, S^n does not need to bifurcate.

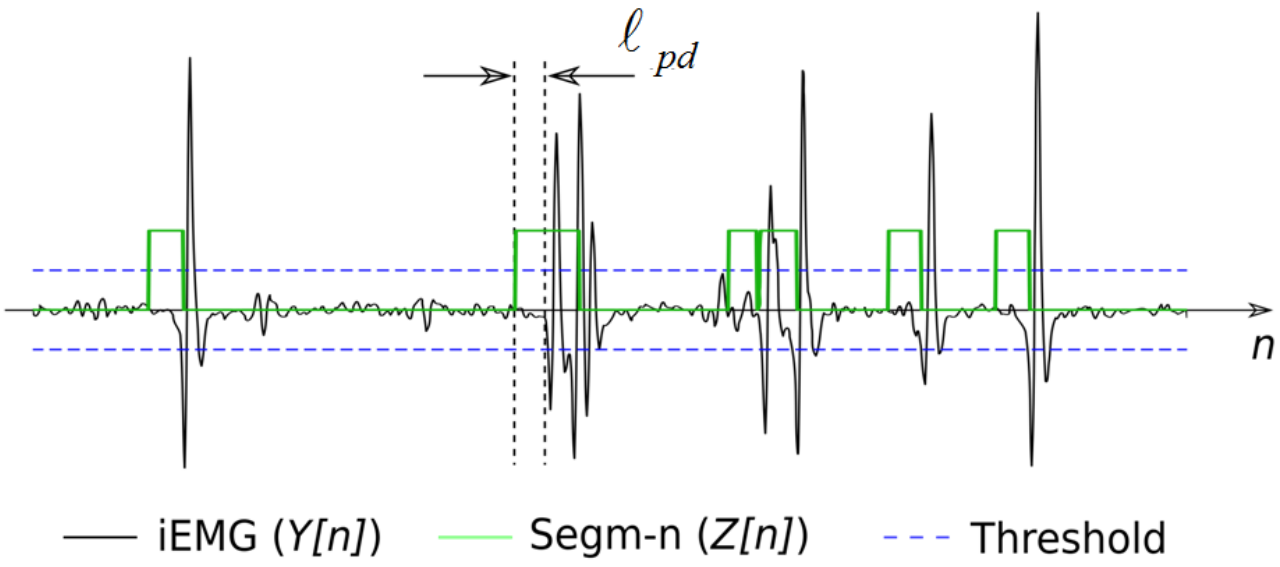


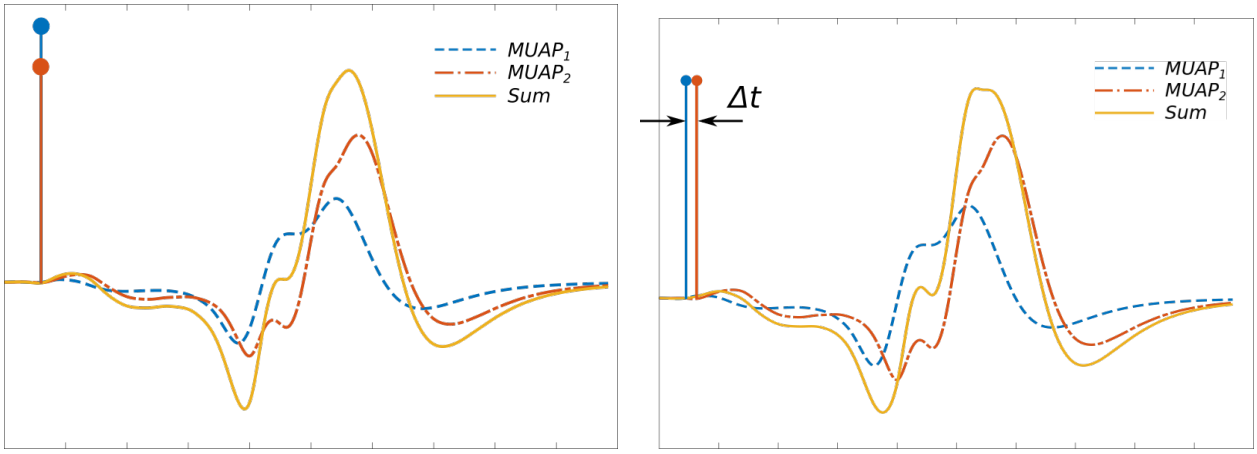
Figure 5.1 – Example of iEMG segmentation. Segments are detected using certain threshold and shifted in time to the left by l_{pd} due to the use of future samples. Bifurcations containing impulses are forbidden while $Z[n]=0$.

An exact implementation of the function $z\left(Y[n+1:n+l_{pd}]\right)$ is beyond the scope of this thesis. Several various segmentation methods are presented in subsection 2.3.2 and any convenient EMG segmentation methods can be used. In our implementation, an adaptive spike-detection threshold derived from [53] was used. An example is given in figure 5.1. The green line denotes the sequence $Z[n]$, have a left shifting compared to the position of MUAP shapes because there is no need to bifurcate in the end part of MUAP segmentations.

5.2.3 Simultaneous spikes interdiction

The simultaneous occurrence of two or more spikes at exactly the same time instant is highly improbable. As an example, considering a sampling frequency of 10 kHz and ten active MUs with mean ISIs of 100 ms, the probability of having more spikes at an instant of time, given that there is already one, is $1-(1-1/1000)(1-2/1000)...(1-9/1000) = 0.0441$. Thus, we consider that the negative impact of this heuristic on the solution can be negligible compared to the gain in computation speed.

The impact is illustrated in Figure 5.2 where an exact superposition (a) can be resolved as its closest possible version (b). The resulting deterioration of MUAP estimates is considered not significant, since the observations in both cases are mostly identical. The gain in computation speed is reached due to the fact that the maximal number of possible branches at step n reduces from $n_{\text{path}} \times 2^{\text{card}(A[n+1])}$ to $n_{\text{path}} \times (\text{card}(A[n+1])+1)$.



(a) (b)

Figure 5.2 – Two close cases of MUAP superposition: (a) - exact superposition of two spikes, a case considered rare and thus excluded from the search; (b) - a close superposition case (Δt denotes the sampling period).

.

We also note that this heuristic does not affect the algorithm capability of decomposing the MUAP superpositions. Only few superposition cases out of many will be actually influenced by it, resulting in a slight deterioration of the MUAPs estimates. The magnitude of this effect is inversely proportional to the sampling frequency.

5.3 Parallelism analysis

The GPU, a multi-core processor, supports thousands of threads running at the same time [117], which typically handled the computation for computer graphics before. Due to tight resource constraints of program modelling of GPU, it is difficult to realize the computation on it. In recent years, with the development of the GPU architectures, GPU program modelling can be programmed by the traditional programming language, such as: C and C++, and it widely equips on personal computers. In this section, we analyze the parallelism of the iEMG signal decomposition model and then implement it into the parallel computation on a GPU.

Based on the HMM model and Bayes filter established in chapters 3 and 4, the serial structure of iEMG signal decomposition at the time index n, for all $n \geq 1$, can be defined as following:

1. Data transmission: the iEMG signal $Y[n]$.
2. Calculation of posterior probabilities $\mathsf{Pr}^{|n}(S^n = s^n)$ of scenarios with formula (4.23).

3. Sorting the posterior probabilities of scenarios and keeping the n_{path} most probable scenarios.
4. Update of the inter-spike law parameters $(\hat{\theta}_{i,S^n})_{i\in\omega}$ with formulas (4.5), (4.6), and (4.33).
5. Update of the impulse responses $(\hat{H}^{|n}_{S^n})_{i\in\omega}$ and the variance of noise $\hat{V}^{|n}$ with formulas (4.14), (4.31), (4.32), and (4.12).
6. Activation and inactivation of MUs with respect to the recruitment model in subsection 3.4.1.
7. Bifurcation of the scenarios and calculation of the priori probabilities $\mathsf{Pr}^{|n}(S^{n+1} = s^{n+1})$ of the scenarios with formulas (4.24), (4.28), and (4.29).
8. Prediction of the observed signal $\hat{Y}^{|n}_{S^{n+1}}$ and of the variance of the innovation $v_{S^{n+1}}$ with formulas (4.9) and (4.15)
9. Data transmission: the state vector at time index n.

The estimation of state sequence computed recursively in the Bayes filter can be roughly interpreted as a loop-based pattern [97]. The performance of a loop-based pattern implemented in the parallel computing structure varies in terms of the dependencies between loop iterations and the work partition between the available processors.

In the recursive estimation process of our algorithm, the strong dependencies are always demonstrated between loop iterations. Since the calculation of the posterior probability of sawtooth sequences $\mathsf{Pr}^{|n}(S^n = s^n)$ at the time index n depends on the results of the prior probability $\mathsf{Pr}^{|n-1}(S^n = s^n)$ and the estimation of the state vector at time index n also depends on its estimated value at time index $n-1$, it is impossible to remove the dependencies between loop iterations. Therefore we must calculate them in strictly sequential manner.

However, in each iteration, the decomposition process can be separated into a number of single tasks (kernel functions) executed in parallel. In each task, the data can be processed in parallel. In the following sections, we will analyze the structure of the decomposition algorithm to minimize communication between processors and to maximize the use of on-chip resources.

5.3.1 Data parallelism

The kernel function, referred to as a GPU program launched in a data-parallel fashion, is executed in a grid consists of several blocks, as illustrated in figure 5.3. Each block comprising a great many threads. Threads of the same block can synchronize and cooperate using fast shared memory[96]. The concept of block and thread is closely related to the hardware of GPU. The program in a block runs in a multiprocessor, while a multiprocessor can execute program of several blocks. The grid and block dimensions can be three dimensional, which determines the number of thread. GPU executes instructions in a single instruction multiple data (SIMD) fashion, usually in a warp containing 32 threads, which means that a signal operation can be executed simultaneously in several threads for multiple data.

Data parallelism is a form of parallelization based on data. It focuses on the distribution of data in the different processors that execute the same operation in parallel [97]:

— Paths on parallel: Before the bifurcation of sawtooth sequences $\mathbf{T}[n]$, there are n_{path} paths which are mutually independent. After the bifurcation, all new paths remain independent. So calculations in all paths could be implemented in the parallel structure with less communication between them.
— Motor units on parallel: According to the hypothesis of the HMM, there is no dependency between any two MUs. Therefore, in every path, the calculation of all MUs can be executed simultaneously.
— Operation on parallel: In every single task, for example: estimation of inter-spike law parameters and estimation of impulse responses, lots of operations as sum of vector or matrix multiplication can be calculated in parallel.

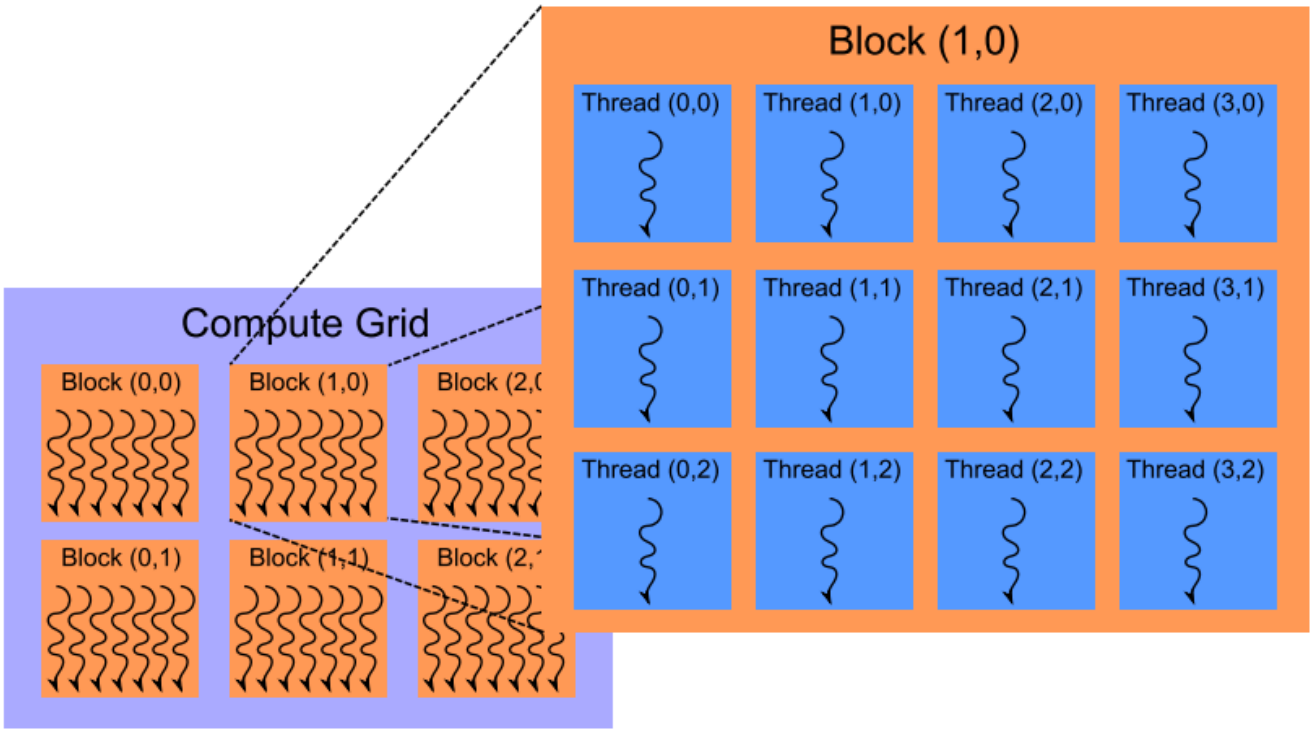


Figure 5.3 – The CUDA concept of a grid of blocks [96]. Each block consists of a set of threads that can communicate and cooperate. Each thread uses its block index in combination with its thread index to identify its position in the global grid.

5.3.2 Task parallelism

Task parallelism is another parallelization that contrasts data parallelism [97]. Rather than simultaneously computing the same function on several data elements in data parallelism, task parallelism consists in performing two or more completely different tasks in parallel. In the structure of iEMG signal decomposition, the simultaneous execution of tasks is limited by the dependences between them.

In each iteration, the data transmission takes place twice: data transmission of observed signal $Y[n]$ from host (CPU) to device (GPU) (task 1) and data transmission of state vector from device to host (task 9). Some parallel computing architectures, e.g. CUDA (Compute Unified Device Architecture), support simultaneous kernel execution and two types of memory copy. The two data transmissions can overlap the other kernel functions. As a result, the time for data transmissions is covered by other calculations in the GPU.

In addition, some parallel computing architectures, such as the Nvidia's GPUs whose micro-architecture is issued after "Fermi", support concurrent kernel execution [101, 99], where different small kernels of the same application context can be executed at the same time to ensure the full use of the GPU resources. According to the structure of the Bayes filter presented in section 4.2, the PDFs of $\Theta[n]$ and $H[n]$ do not depend on each other. Therefore, in every loop, the tasks related to the estimate of the inter-spike law parameters $\hat{\theta}_{i,S^n}$ can be executed simultaneously with the ones related to impulse responses $\hat{H}_{S^n}^{|n}$. Tasks 4 and 5, as well as tasks 7 and 8, can be calculated at the same time.

Some concurrency example is given in figure 5.4, where blocks in different colors represents respectively the two types of data transmissions: DH: from device to host, and HD: from host to device, the kernel functions of GPU and programs of CPU. Five concurrency cases denotes the serial execution and various types of overlap executions. The overlap execution in the white dashed frame of the fifth concurrency example is similar to our parallel task execution in each iteration n. Kernel functions respectively related to the inter-spike law parameters and impulses responses are executed at the same time, while the

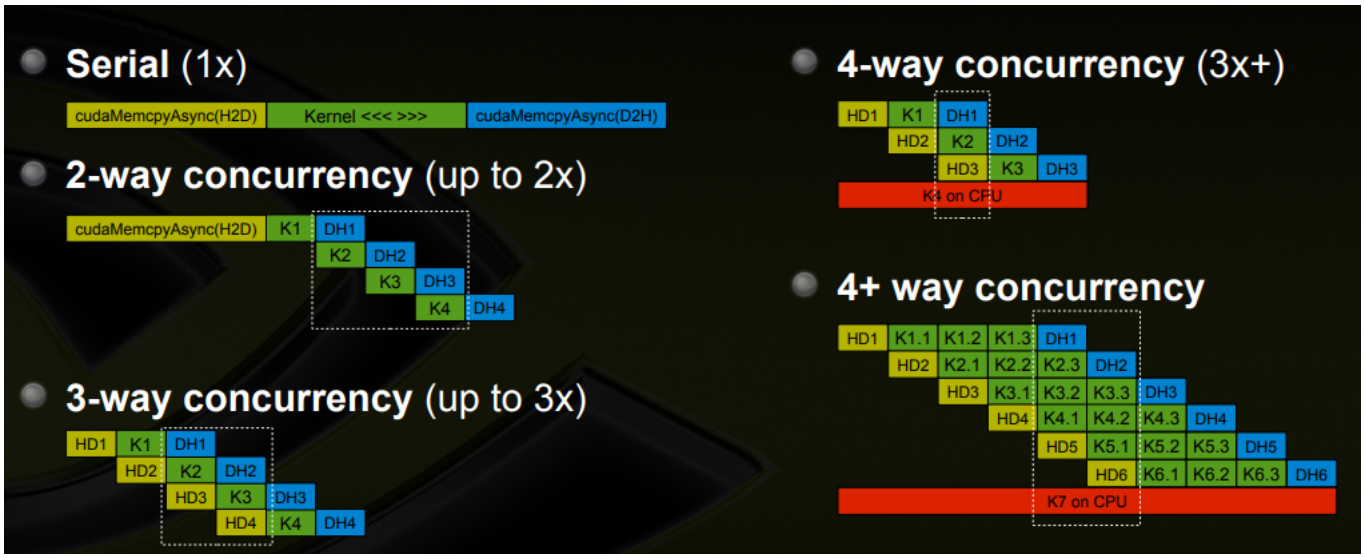


Figure 5.4 – Concurrency example: There are five concurrency examples: 1. Serial execution. 2. Concurrency execution between a kernel function and the memory copy from device to host. 3. Concurrency execution between a kernel function and two types of memory copies (DH: from device to host and HD: from host to device). 4. Concurrency execution of a kernel function, two types of memory copies and CPU. 5. Concurrency execution of three kernel functions, two types of memory copies and CPU.

data transmissions are covered.

5.4 Task analysis

To accelerate the decomposition, the algorithm will be implemented in the parallel calculation in GPU. Some of these tasks need to be analyzed in the parallel environment: In every iteration, tasks 1 and 9 are data transmissions. Tasks 2, 4, 5, 6, and 8 are ordinary small-size parallel computing problems, whereas task 3 consists of sorting the posterior probabilities of scenarios and keep the n_{path} most probable paths which is related to a classic parallel sorting problem. Task 7 (bifurcation of sawtooth sequences) which changes the size of parallel structure also deserves consideration.

5.4.1 Parallel sorting

After the bifurcation of sawtooth sequences, based on the transition distribution presented in subsections 3.4.1 and 3.4.2, there are usually at most $n_{\text{path}} \times 2^{\text{card}(A[n+1])}$ paths. The size of parallel sorting problem varies from n_{path} to $n_{\text{path}} \times 2^{\text{card}(A[n+1])}$. With the assumption of rareness of simultaneous spikes presented in subsection 5.2.3, the maximum number of bifurcations reduces to $n_{\text{path}} \times (\text{card}(A[n+1]) + 1)$. Thus, task 3 is a classic sorting problem.

Sorting is a computational building block of fundamental importance and is one of the most widely studied algorithmic problems. Several sorting algorithms were proposed to resolve this problem in an efficient way, such as: the comparison-based sorting algorithms including bubble sort, merge sort, quick sort, heap sort, etc, and non-comparison-based sorting algorithms including counting sort, bucket sort, and radix sort. The limitation of the comparison-based sorting algorithms performance is that they have an average-case lower bound of $O(n \log n)$ comparison operations [118], which is known as linearithmic time. Whereas, some non-comparison-based sorting algorithms can achieve $O(n)$ performance by using operations other than comparisons. In recent decades, with the development of parallel computing, the parallelization of sorting algorithm reduces greatly the time complexity. It allows some parallel comparison-

based sorting algorithms break the limitation of performance $O(n \log n)$ in the time complexity. In the following parts of this subsection, we will analyze a few parallel sorting algorithms and select the most suitable for our problem.

Quick sort

Quick sort as one of the most common sorting algorithms for sequential computers, due to its simplicity and low overhead, has an average complexity of $O(n \log n)$. Quick sort is a divide-and-conquer algorithm that sorts a sequence by recursively dividing it into smaller sub-sequences until a sorted sequence is obtained.

There are several parallel formalisations of quick sort. The most naive idea is sorting completely independent sub-sequences in parallel. Therefore, parallelising quick sort is to execute it initially on a single process; then, sorting the sub-sequences in parallel after each time of sequence or sub-sequences partitions until the sub-sequences cannot be further partitioned. If we have n processors, the time complexity of this formulation of quick sort has a potential to reach $O(\log n)$. However, its performance is limited because it performs the partitioning step serially [119]. Another more important problem is the choice of pivot to partition the sequence or sub-sequences.

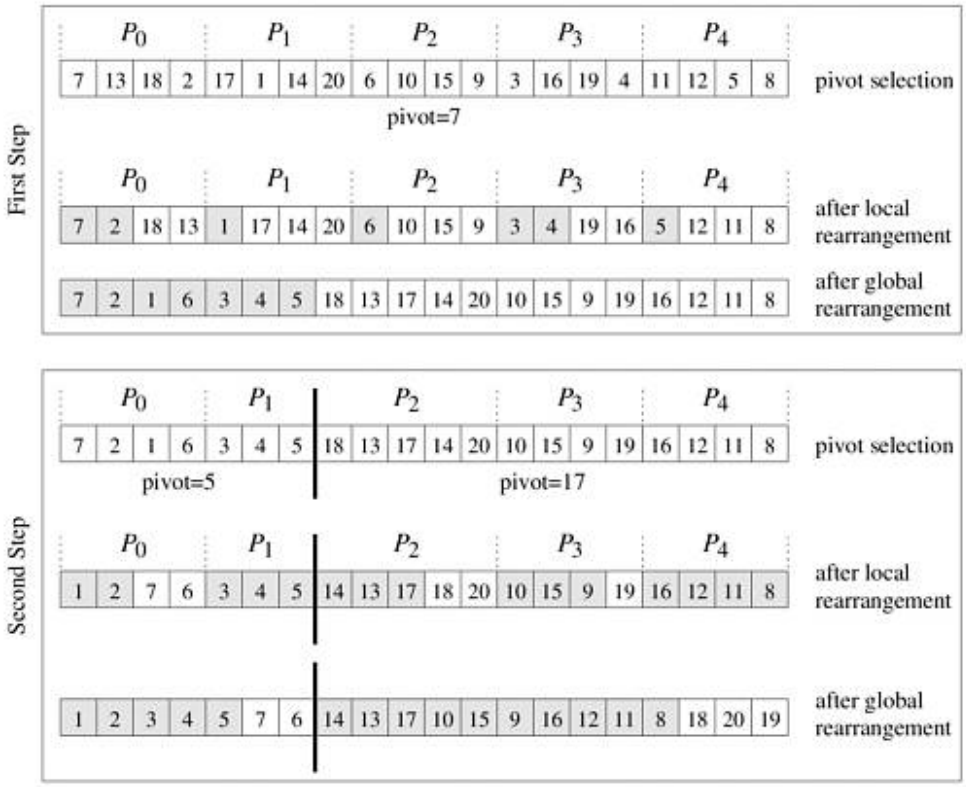


Figure 5.5 – Two sets of operations of the shared-address-space parallel formulation of quick sort [119]

With respect to other formulations such as: parallel formulation of parallel random access machine (PRAM), the shared-address-space parallel formulation of quick sort is more realistic. We divide the n elements of the sequence into p processors. Then, we start a recursive operations as following: Firstly, a pivot is selected; Secondly, p processors separate their sub-sequences simultaneously to two sequences with elements respectively more than and less than pivot; Thirdly, we gather sequences in every processor with small elements to some processors, while sequences with large elements are placed on the other processors. The three operations refer to as pivot selection, local arrangement and global arrangement. We repeat the

three operations until the sequence is sorted. As illustrated in figure 5.5, the first and second loops of operations are given to sort a sequence with 20 elements by 5 processors.

If we ignore the time for the movement of the sub-sequences elements in the operations of local and global arrangements, the most important problem influencing the execution time is the selection of pivot for every sub-sequences. Due to the great variability of our posterior probabilities distribution, this parallel formulation of quick sort is difficult to implement in our algorithm.

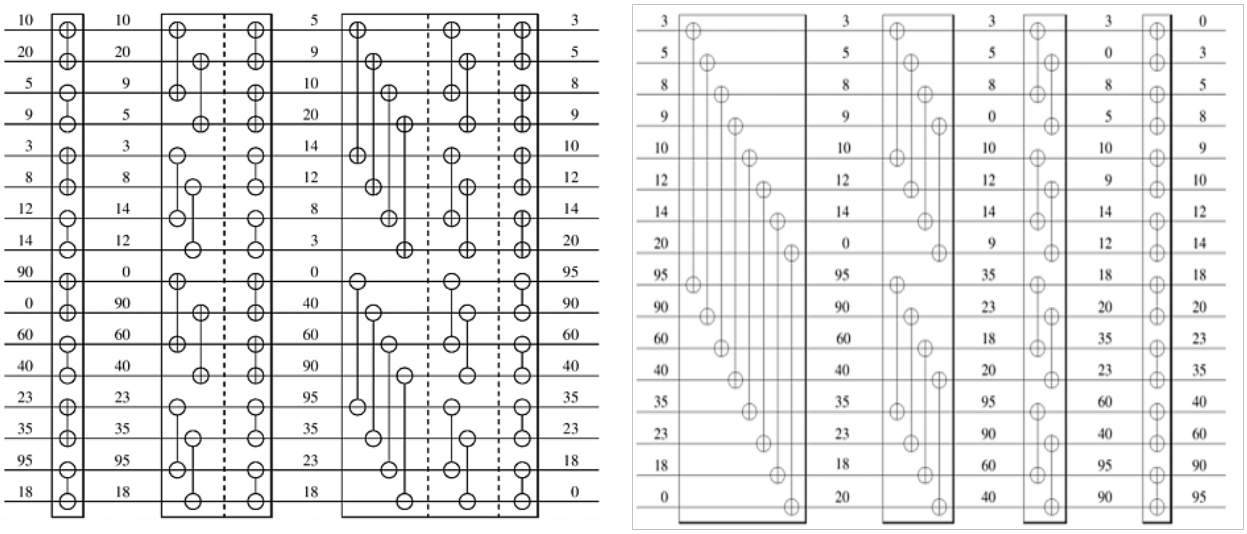
Odd-even sort

An odd-even sort or odd-even transposition sort is developed originally for use on parallel processors with local interconnections. The odd-even algorithm sorts n elements in n phases (n is even), each of which requires $\frac{n}{2}$ compare-exchange operations. This algorithm contains two phases, termed the odd and even phases. In the odd phase, elements with odd indices are compared with their right adjacent elements, and if they are not in an expected order they are exchanged. In the even phase, elements on the even position act the same operation as in the odd phase. The sequence will be sorted after n phases of odd-even exchanges. In each phase, the odd-even sort needs $O(n)$ comparisons. And it requires $O(n)$ phases. Thus, its sequential complexity is $O(n^2)$.

Parallelization of odd-even sort is simple. $\frac{n}{2}$ operators make the compared-exchange simultaneously in each odd-even phase, which requires time $O(1)$. Therefore, in the parallel environment, with $O(n)$ processors, the time complexity of odd-even sort is $O(n)$.

Bitonic sorting

Bitonic sort, which focuses on converting a bitonic sequence comprising a monotonically increasing sequence and a monotonically decreasing sequence into a sequence in an expected order, is a comparison-based sorting algorithm. The data-independent of compared-exchange operation in bitonic sort makes it one of the fastest and suitable parallel sorting algorithms [120].



(a) (b)

Figure 5.6 – Bitonic sorting of a sequence with 16 elements [119] (a) The sequence is converted three times to a bitonic sequence; (b) A bitonic sequence is sorted to a monotonically increasing sequence.

As shown in figure 5.6, a sequence with 16 elements is sorted. The idea is to divide the whole sequence into several bitonic sequences, then sort them to ordered sequences to form new bitonic sequences. In the first step, 16 elements can be considered as 8 bitonic sequences. We can sort the 8 bitonic sequences to 8 ordered sequence. In the second step, the previous 8 sorted sequences form 4 bitonic sequences. Then, we

sort the 4 bitonic sequences to 4 sorted sequence. After that, we do two times the same operation as before. A monotonically increasing sequence is obtained.

The time complexity of bitonic sorting is $O(n\ (\log_2 n)^2)$ in the serial computing. However, in the parallel computing, if we complete $\frac{n}{2}$ compared-exchange operations at the same time, the time complexity turns to $O((\log_2 n)^2)$ [121].

We notice that in every step of bitonic sorting, some small bitonic sub-sequences are arranged to sorted sub-sequences to form new bitonic sequence. In task 3, the final objective is to keep the n_{path} most probable scenarios. Therefore, in the bitonic sequence, if the size of sorted sub-sequences is more than n_{path}, we keep the n_{path} biggest values, then form the new bitonic sub-sequences. This measure removes part of unnecessary work to sort less probable scenarios.

The parallel version of other non-comparison-based sorting algorithms, such as bucket sort and radix sort [122, 123], which has a relative high demand of memory space, is not suitable for our algorithm. Thus, we choose the bitonic sorting for the task 3 from these parallel comparison-based sorting algorithms due to its easy implementation and high parallelism. Moreover, with the increasing the number of MUs and scenarios, if the dimension of the posterior probability sequence is too large, some hybrid sorting algorithms [122, 124, 125] normally composed of the merge sort and the bitonic sorting (or the radix sort), can be applied, which is beyond the scope of this thesis.

5.4.2 Indexes of bifurcation

Path $S[n]$ bifurcates in at most $A[n+1]+1$ different ways giving an overall number of $n_{\text{path}} \times (A[n+1]+1)$ of new paths. After the parallel sorting, we only keep the n_{path} most probable new paths at time index $n+1$. To avoid the memory allocation and initialization of each bifurcation originated from one path, indexing is used.

Here is an example for two active motor neurons, which gives a two-dimensional vector $\mathbf{T}[n]$ and three possible bifurcations (the used values are arbitrary):

$$\text{if } \mathbf{T}[n] = \begin{bmatrix}450\\635\end{bmatrix}, \quad \mathbf{T}[n+1] \in \left\{ \begin{bmatrix}451\\636\end{bmatrix}, \begin{bmatrix}0\\636\end{bmatrix}, \begin{bmatrix}451\\0\end{bmatrix} \right\} \tag{5.1}$$

Each i-th motor unit can either not fire at time index $n+1$ ($T_i[n+1] = T_i[n]+1$) or fire if ready ($T_i[n+1]=0$). Therefore, a binary code can be associated to each configuration in $\mathbf{T}$.

$$\mathbf{T}[n+1] \mapsto \begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 0\end{bmatrix}; \tag{5.2}$$

This code is unique for each bifurcation within a scenario.

Therefore, in task 7, we initialize the indexes instead of the bifurcation. After sorting the bifurcations and keeping the n_{path} most probable paths at time index $n+1$, according to the unique index of every bifurcation kept, we initialize the new scenarios.

5.5 Parallel structure

The virtual observation $Z[n]$ presented in section 5.2.2 defines which bifurcations are possible at the time index n. $Z[n]=0$ means that there is no spike at time index n, that is, $T[n] = T[n-1]+1$ and $\hat{Y}_{S^n}^{|n-1} = 0$. Hence, we do not need to bifurcate scenarios (task 7) and predict $\hat{Y}_{S^n}^{|n-1}$ (task 8) at time index $n-1$. For the next time index, sorting the posterior probabilities of scenarios and keeping the n_{path} most probable scenarios (Task 3) is skipped, because after the bifurcation, the number of scenarios does not change. Moreover, we do not need to update the impulse responses (Task 5), if the length of pre-detection l_{pd} is equal to or more than ℓ_{IR}.

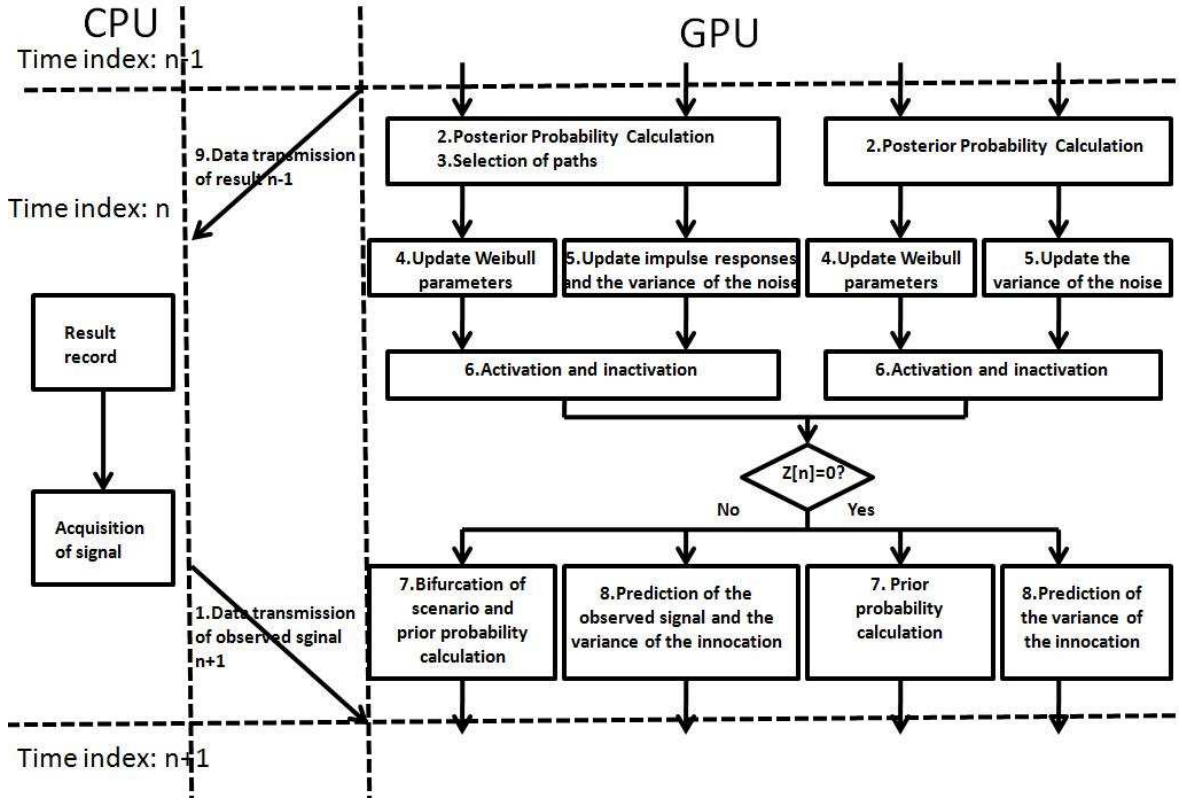


Figure 5.7 – Parallel structure of iEMG signal decomposition algorithm

Considering the parallelism analysis presented above, we obtain the parallel structure illustrated in schema 5.7.

The work complexity of this algorithm is the complexity of bitonic sorting in the serial structure $O(n\,(\log_2 n)^2)$, where n here is the number of new bifurcated scenarios. Its time complexity is $O((\log_2 n)^2)$. Based on the subsection 4.5 and 5.2.3, the value of n can be calculated with the formula $n_{path} \times (\text{card}(A[n]) + 1)$. Therefore, the execution time of algorithm depends on the number of active MUs in the signal and the number of scenarios to decompose the signal. Furthermore, as schema 5.7 shown, the work complexity of algorithm also influenced by the value of $Z[n]$. $Z[n] = 0$ means that the signal only contains noise. We only need to update the inter-spike law parameters, the MUAP shapes and the probability of each scenario. With respect to the case $Z[n] \neq 0$, we have less kernel function executions. Thus, the ratio of signal containing spikes to signal only containing noise and the pre-detection parameters l_{pd} also influence the decomposition time of algorithm. Moreover, the computation power of the GPU including the architecture of GPU, the number of CUDA cores and the dimension of memory space is one of the most important factors for the real-time decomposition.

Due to the execution time of algorithm influenced by various factors, it is difficult to define quantitatively the upper limit of the sources in the signal for the real-time decomposition. We can briefly summarise that more sources number meaning more complicate signals, more scenarios number meaning more precise decomposition and bigger ratio of signal containing spikes to signal only containing noise and the pre-detection parameters l_{pd}, requires more execution time or computation power. And the discussion of decomposition velocity without considering the decomposition performance is meaningless. A quantitative analysis with more details will be given in subsection 7.3.2.

5.6 Discussion and conclusion

To accelerate the decomposition algorithm, we utilize several techniques including: limit the number of scenarios, forbidden the bifurcations in the signal segments only containing the noise and the simultaneous spikes. The algorithm is implemented in the parallel environment, by the means of parallelism analysis in terms of data and task (kernel functions), and task parallel analysis. The latency is covered by the overlap of kernel function and the resource of GPU is used in maximum. In the following chapter 6, some signal pre-processing techniques to improve the performance of algorithm, and experimental and simulation protocols for the signal acquisition and results evaluation will be presented.

6

Signals and preprocessing

6.1 Introduction

In this chapter, two means of signal preprocessing, including signal pre-filtering and MUAP waveforms clipping, are firstly introduced to improve the performance of the proposed algorithm. The signal pre-filtering is to suppress the baseline of noise and sharpen the MUAP waveform, thus highlights the MUAP segmentations. The MUAP waveforms clipping cuts the low activity parts of MUAP shapes in order to adapt the MUAP form to the Bayes filter in chapter 4.

Then, we present the experimental and simulation protocols including the formulation of simulation signals, the acquisition of experimental signals, and indexes to evaluate the complexity of signals and to evaluate the performance of decomposition results. For the simulation signals, the simulated laws and the range of related parameters are defined. For the experimental signal, the recorded electrodes, the target muscles and all parameters concerning the acquired signals are introduced.

6.2 Signal preprocessing

6.2.1 Signal pre-filtering

Since the muscle tissue acts as a low-pass filter [126], the discharge running along a muscle fiber produces MUAP with slowly varying edges and a quickly varying intermediate part. Thus, the "differentiator" [127, 128], widely used in biological signal to suppress the baseline of noise but without attenuating greatly the MUAP amplitude, is applied to the original signal to obtain sharpened spikes [53]:

$$X[n] = Y[n+1] - Y[n-1] \tag{6.1}$$

Where $Y[n]$ is the iEMG and $X[n]$ is the filtered signal. As described in subsection 2.3.2, this filter decreases greatly the influence of background noise, containing small dimension MUAP shapes with low frequency. It makes the boundary between the MUAP segmentations and the signals only containing noise much clear, thus simplifies the process of pre-detection presented in subsection 5.2.2.

An example is shown in Figure 6.1. The figure 6.1(a) is the raw iEMG and the Figure 6.1(b) is the filtered signal. We observe that the differentiation highlights the spikes in the signal and suppresses the noise in the background.

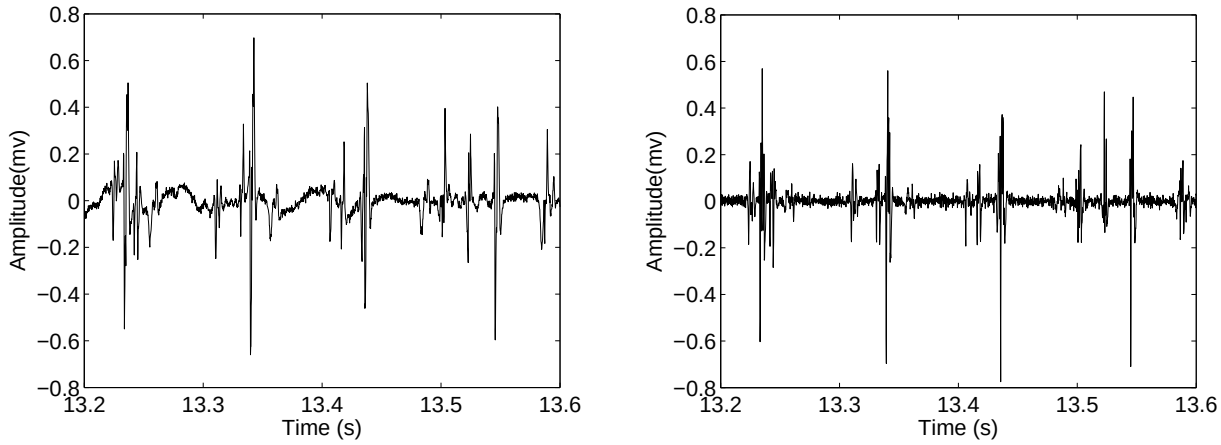


Figure 6.1 – Differentiation (a) The iEMG signal before differentiation; (b) The iEMG signal after differentiation.

6.2.2 MUAPs clipping

In section 4.4, one of the initial conditions for the MUAPs shape estimation is the initial prior $\hat{H}_{s^0}^{|0}$. Although the estimation of initial prior $\hat{H}_{s^0}^{|0}$ is beyond the scope of this paper, their waveforms are similar to these shown in figure 6.2(a).

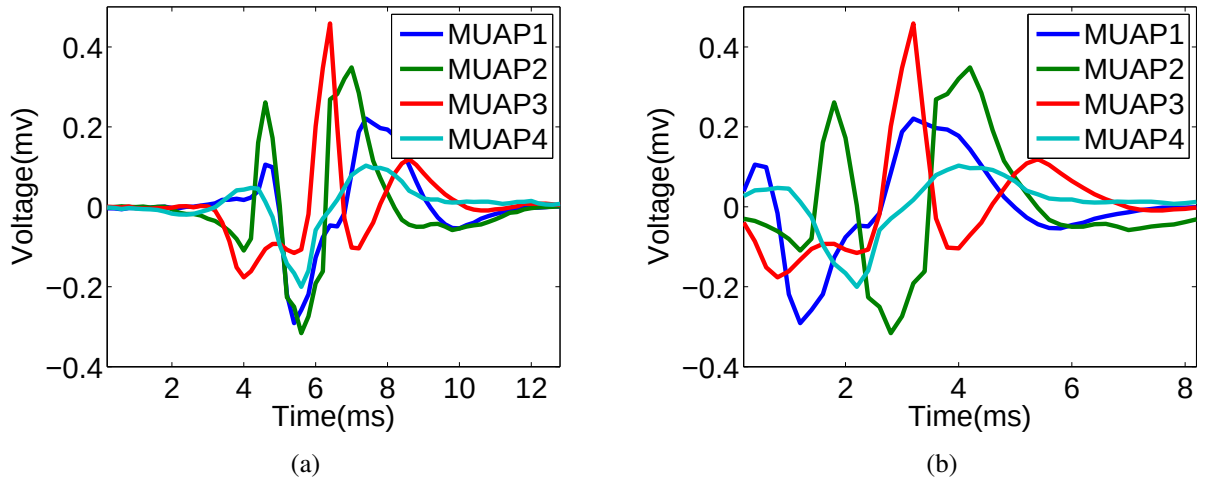


Figure 6.2 – MUAP waveforms clipping (a) MUAP waveforms before clipping; (b) MUAP waveforms after clipping

To ease the decomposition algorithm, which acts with a limited number n_{path} of possible activation scenarios, we have to select the area of high activity in the MUAPs by removing the almost zero side parts. We use a method inspired by the segmentation method used in [53] to clip the MUAPs, that is to say that we remove the side parts whose absolute value is lower than the threshold α, solution of the implicit formula:

$$\alpha = c \ \text{std}\{x[n], 1 \leq n \leq N \mid x[n] < \alpha\} \tag{6.2}$$

Where std means standard deviation. Typically, the authors of [53] proposed to use 0.2 second of signal ($N = 2000$) and $c = 1.5 \sim 3.5$. Figure 6.2(b) shows the result of MUAPs clipping.

This MUAP waveforms clipping measure demonstrates two advantages: Firstly, after clipping, as illustrated in figure 6.2(b), the beginning parts of different MUAPs have a dispersion value. Therefore, the

posterior probabilities for the bifurcated scenarios calculated with the formula (4.23) are more scattered, which prohibits the proper scenario associating to the low posterior probability. Secondly, this measure reduces the length of MUAP templates, thus reduces the length of pre-detection l_{pd} presented in subsection 5.2.2 and accelerates the decomposition process.

6.3 Experimental and simulation protocols

6.3.1 Signals

The algorithm was evaluated using simulated and experimental iEMG signals. Simulated signals were generated by the described Markov model with sampling frequency of 5 kHz and duration 20 s. MUAP shapes for the simulations were obtained from experimental iEMG signals that were manually decomposed. The value of the refractory period was chosen to be 30 ms. For the Weibull distribution, the location parameter t_0 ranged from 60 ms to 90 ms and the concentration parameter β ranged from 2 to 6. The SNR (Signal to Noise Ratio) was set to 10 dB. Three groups of signals were simulated with respectively 6, 8 and 10 MUs. In each group, there were 10 signals.

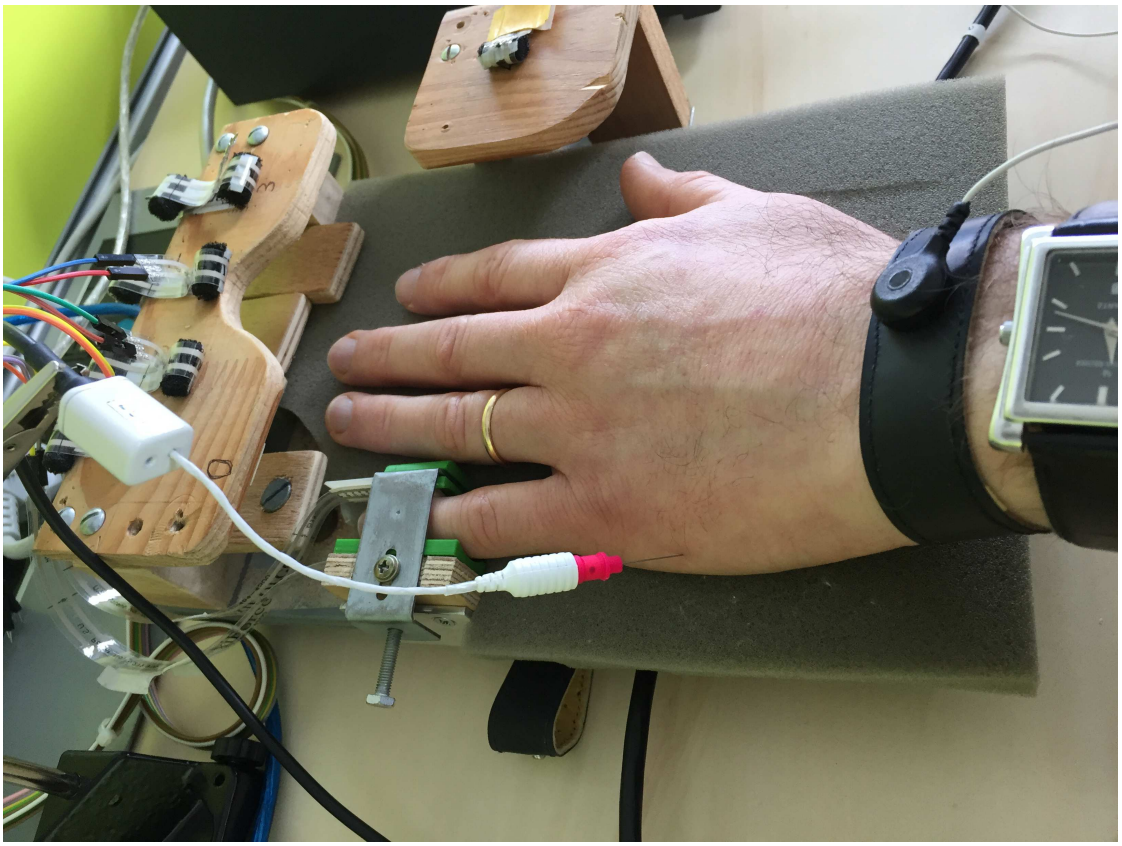


Figure 6.3 – Acquisition of experimental signals from the abductor digiti minimi muscle

Experimental signals were acquired from two muscles. A set of fine-wire experimental iEMG, provided by Prof. Dario Farina and his team of Imperial College London , was acquired from the tibialis anterior (TA) muscle of a 26 years-old healthy man. The subject was asked to perform three trials of an isometric force tracking task of 24 s duration. The force profile was trapezoidal with target force set to 30% of the maximal voluntary contraction (MVC). The wire electrodes used for these recordings were made of Teflon coated stainless steel (50 um diameter; A-M Systems, Carlsborg, WA, USA) and inserted into the muscle with 25G needles.

As shown in figure 6.3, a second set of experimental signals, measured by Prof. Yann Pereon from CHU Nantes, was acquired from the abductor digiti minimi (ADM) muscle of a healthy 28 years-old man with a needle electrode (30G Myoline disposable concentric needle) inserted in the muscle belly at a depth of 4-6 mm. The full abduction range of the little finger (45 degrees) was divided into five equiangular positions (including the maximal abduction and excluding full adduction). The subject was asked to perform a muscle contraction at low muscle activation level (subjective) in each position while being given a visual feedback on the recorded signal.

The signals from TA and ADM were amplified, band-pass filtered between 100 Hz and 4.4 kHz and sampled at a frequency of 10kHz (OTBioelettronica MEBA amplifier). The signals from TA were subsequently down-sampled to 5 kHz.

Since the algorithm initialized all MUs as inactive, the onset of the experimental signal was associated to a recruitment of all MUs at once, which does not prevent the algorithm from converging to the correct state, as will be shown later. Algorithm 1 was applied to decode the simulated and the experimental signals. The activation probability λ and the maximum time t_I were respectively set to 0.03 and $7t_R$; The window length corresponding to the adaptivity was 1.4 s. The number of selected paths was set to 128, 192, 256 and 384 for simulated and experimental signals.

6.3.2 Indexes of performance and complexity

Automatic decomposition results were evaluated in terms of spike train similarity relatively to the reference decomposition. In the case of experimental signals, the reference was an expert-provided manual decomposition using EMGLAB [75]. In case of simulated signals, the exact spike trains used in the simulation were known.

First, to characterize the decomposition problem complexity, we introduce a superposition percentage Sup:

$$\text{Sup} = \frac{Nb_{sup}}{Nb_{spikes}} \tag{6.3}$$

where Nb_{spikes} is the overall number of spikes in the reference train; Nb_{sup} is the number of spikes that are involved in superpositions. We considered a MUAP superimposed with others if there was at least one other MUAP closer than 3 ms to it. This value was chosen as half of the average MUAP duration that we usually observed in experimental signals.

In order to quantitatively evaluate the decomposition results, we used global sensitivity and global positive predictivity values, defined as following. A MUAP was considered correctly identified (true positive) if the reference train contained a spike from the same MU within a margin of 1 ms around it. Thus, global sensitivity was defined as the overall number of correctly identified MUAPs from all MUs, divided by the overall number of spikes in the reference decomposition. Global positive predictivity was the number of correctly identified spikes divided by the overall number of spikes in the decomposition under evaluation.

We also performed an individual analysis of each MUAP train, using "classification phase" indexes proposed in [81]. These indexes included sensitivity, specificity and accuracy, as they are defined in [81].

6.4 Discussion and conclusion

Two pre-processing techniques suppress the baseline of noise and highlight the MUAP waveforms in the iEMG signal, which can improve the performance of the proposed algorithm. Simulated signals and two sets of experimental signals detected from two muscles by different types of electrodes can test the efficiency, adaptability and robustness of the algorithm. The indexes of performance and complexity evaluate the signal and the decomposition algorithm in various aspects. The decomposition results will be shown in chapter 7.

7

Results

7.1 Introduction

In this chapter, we will present the validation results of our parallel computation algorithm on simulated and experimental signals. The performance of the algorithm is evaluated in terms of the decomposition quality in section 7.2 and the decomposition velocity in section 7.3. In the evaluation of decomposition quality, the decomposition results comprise the detection and classification of MUAPs, the estimated firing rates, and the optimized MUAP waveforms. The HMM, driven by the transition models including the recruitment model and the renewal model, and the Bayes filter are verified by these decompositions. For the decomposition velocity, experimental and simulated signals with various MU numbers are decomposed by the parallel computation algorithm. The decomposition accuracy, the velocity and the relation between them are exhibited. Moreover, an analysis of decomposition velocity in detail will be provided.

7.2 Performance

In this section, we have decomposed dozens of simulated and experimental signals in order to analyze the performance of our parallel computing algorithm. We cannot show all the decomposition results in detail. Thus, we would like to demonstrate exhaustive results of three signals: one simulated and two experimental signals. The simulated signal with 10 MUs is generated by the described HMM with sampling frequency of 5 kHz and duration 20 s. It shows a relative complex recruitment manner. The results are fully shown in subsection 7.2.1. Other simulated signals results will be presented in subsection 7.3.2 to reveal the relation between the decomposition accuracy and the execution time of the algorithm. The global view of all experimental signal decomposition results are provided in table 7.2, while the detailed results of two experimental signals, respectively detected from TA muscle and ADM muscle, are illustrated in subsection 7.2.2.

7.2.1 Simulated signals

According to the HMM described in chapter 3, we simulated an iEMG signal with 10 MUs, sampling frequency of 5 kHz, and duration 20 s. There are 1821 MUAPs (or spikes) in this signal. 498 of them are involved in the superposition. The superposition percentage, noted as Sup, equals to 27.35%. We decomposed this signal by the proposed algorithm with 384 scenarios in the parallel computing environment. The length of maximum tracking window is set to 0.6 s.

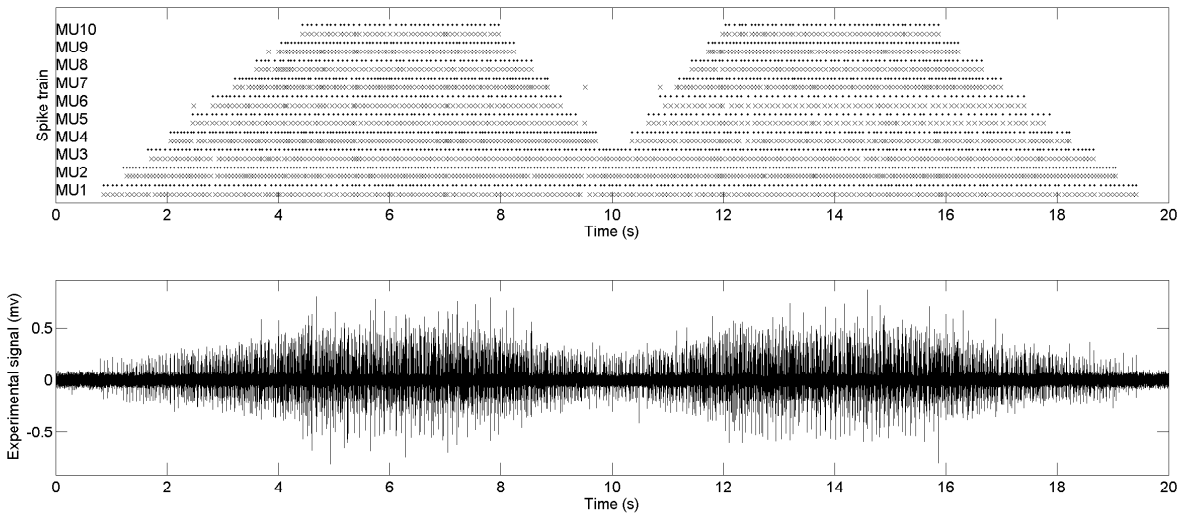


Figure 7.1 – Comparison of automatic decomposition (crosses, 'x') and actual results (points, '.') in the upper panel and the simulated signal with 10 MUs in the lower panel.

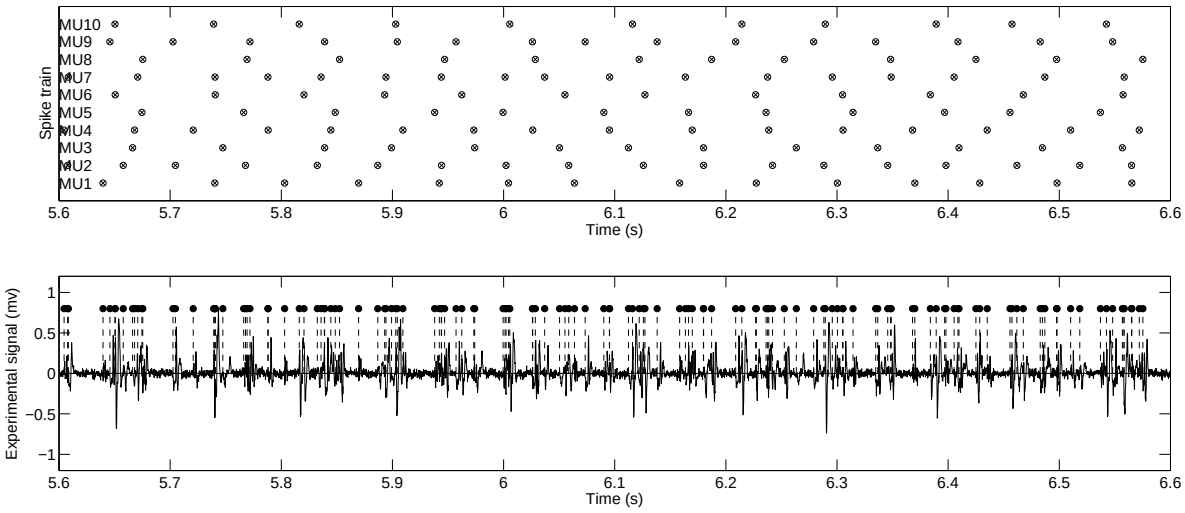


Figure 7.2 – An extraction of the simulated signal decomposition shown in figure 7.1; circles '∘' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions.

As shown in figure 7.1, a global version of the simulated signal with 10 MUs is given. In the upper panel, we compare the automatic MU classification results with the actual spike trains of each MU. We can also learn the recruitment manner of this signal: 10 MUs are eventually recruited. 6 MUs of them have a process activation-deactivation-activation. We validate the proposed algorithm in the signal with this relative complex recruitment manner, in order to test the recruitment model presented in subsection 3.4.1, as well as the estimation of inter-spike law parameters introduced in section 4.3. The lower panel of figure 7.1 shows this simulated signal.

We extract one second of MUAP classification results of the decomposition result. The segment of signal from 5.6 s to 6.6 s is one of the most complex parts of signals, where all of 10 MUs are active. This detailed view is shown in figure 7.2. In the upper panel, the cross represents automatic decomposition results, while the circle denotes the actual one. We note that all the decomposition results are coincidence

Table 7.1 – Decomposition performance for the simulated signal with 10 MUs

MU	Sens.	Pred.	Sens.	Spec.	Acc.
Detection	96.05	93.63	-	-	-
MU1	-	-	96.94	99.28	98.98
MU2	-	-	94.00	98.66	97.87
MU3	-	-	95.71	99.41	98.93
MU4	-	-	95.65	99.16	98.70
MU5	-	-	97.40	99.50	99.32
MU6	-	-	94.93	99.20	98.87
MU7	-	-	93.85	98.63	98.15
MU8	-	-	97.71	99.63	99.49
MU9	-	-	98.53	99.38	99.32
MU10	-	-	100.00	99.58	99.60

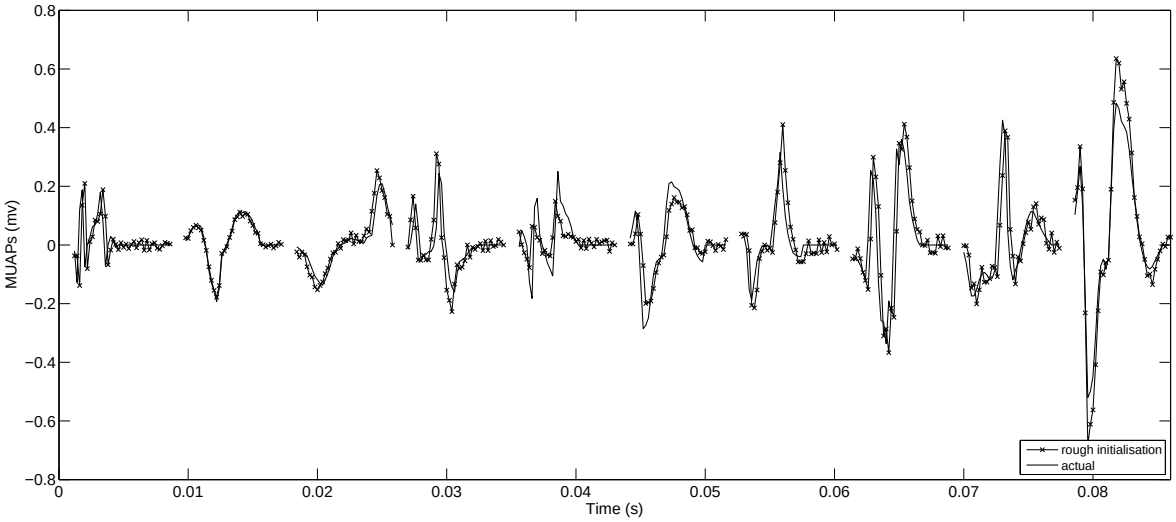


Figure 7.3 – Initial MUAP shapes (actual MUAP shapes contaminated by the noise) and their actual MUAP shapes for the signal presented in Figure 7.1

with the actual ones. In the lower panel, the dashed lines represent the related position of MU spikes, shown in upper panel, in the simulated signal. We find that there are a great number of superpositions, comprising 6 MUAPs at most, in this segment of signal. These superpositions are all perfectly resolved. More specific quantitative performance indexes are given in table 7.1. All the performance indexes are more than 93%.

The rough initial waveforms of MUAPs, which is the actual MUAP shapes contaminated by the noise, is illustrated in figure 7.3, as well as their actual waveforms. Compared to the actual ones, the initial ones show a great difference. Our algorithm also offers the MUAP shapes optimisation during the decomposition. Since the difference between the final estimated MUAP waveforms and the actual ones cannot be distinguished visually, we do not show the final estimated ones in this figure. However, the normalised misalignment of estimated MUAP shapes (in dB), defined as $20\log_{10}[\|H[n]-\hat{H}^{|n}_{S^n}\|_2/\|H[n]\|_2]$, is depicted in figure 7.4. The LMS filter presented in subsection 4.4.2 optimizes efficiently the MUAP shapes. The convergence process only takes 6 s, despite the activation of the 10th MU in the 4 s.

Figure 7.5 shows the firing rates of each MU. The full lines represent the estimated firing rates, calculated with the estimated inter-spike law parameters with the formula (3.16), while the dashed lines denote the actual firing rates. All the estimated one catches quickly the actual one due to the tracking factor pre-

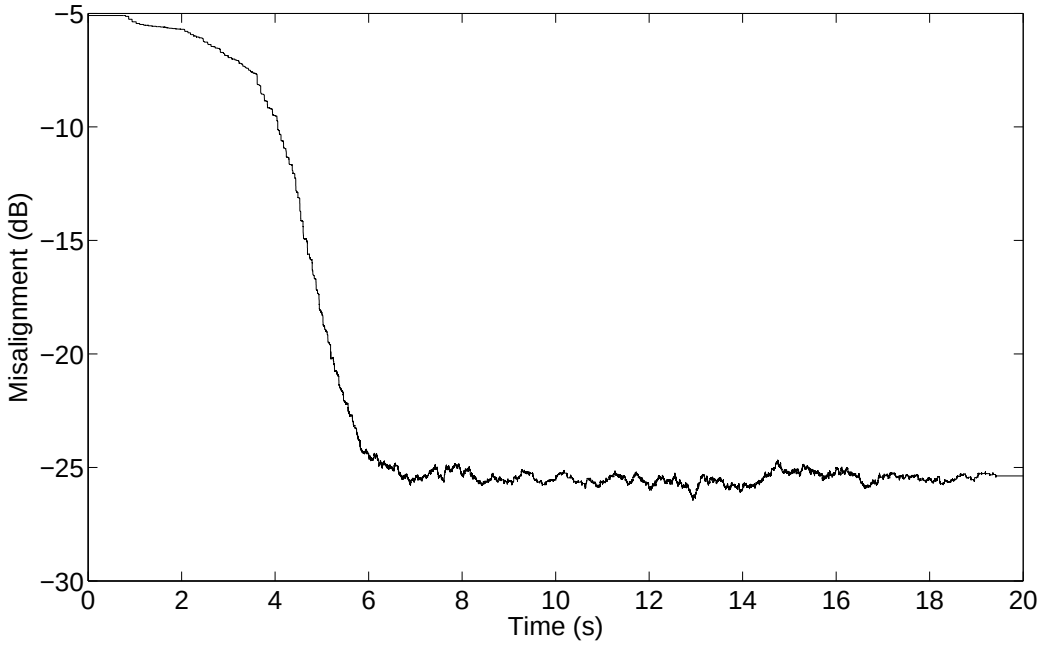


Figure 7.4 – Normalised misalignment of estimated MUAP shapes

sented in section 4.6. We should notice that the two activation segments of the 4th, 5th, and 6th MU are simulated with different actual firing rates. Their estimated firing rates also show a good convergence.

7.2.2 Experimental signals

Table 7.2 – Decomposition performance for experimental signals: 'TA' and 'ADM' respectively represent signals from the tibialis anterior and abductor digiti minimi; 'Position' represents the level of abduction scaled equiangularly from the full adduction (0, not included) to the full abduction (5); 'Nb MUs' is the maximal number of MUs concurrently active in the signal; 'NB spikes' represents the overall number of spikes in the signal; and 'Sup.' is the percentage of superposition; 'Sens.' and 'Pred.' represent respectively the global sensitivity and predictivity as estimated by comparison with the manual expert decomposition.

Index	Muscle	Duration (s)	Force (MVC%)	Position	Nb MUs	Nb spikes	Sup.(%)	Sens. (%)	Pred.(%)
1	TA	24	20	-	5	873	18.10	91.75	90.61
2	TA	24	20	-	5	936	18.38	95.83	94.72
3	TA	24	20	-	6	933	17.15	94.53	93.43
4	TA	24	30	-	7	1176	22.28	88.78	85.71
5	TA	24	30	-	8	1295	28.96	88.34	86.68
6	ADM	5	-	1	2	61	6.56	100	100
7	ADM	5	-	2	5	153	15.03	96.73	92.50
8	ADM	5	-	3	6	192	21.35	96.88	92.54
9	ADM	5	-	4	6	281	23.13	90.10	90.14
10	ADM	5	-	5	7	371	28.84	92.99	92.74

Ten experimental signals (three recorded from the TA at 20% MVC, two recorded from the TA at 30% MVC, and five detected from the ADM muscle in the five different abduction positions) were automatically decomposed. As shown in Table 7.2, for these signals, the number of MU ranged from two to eight and the

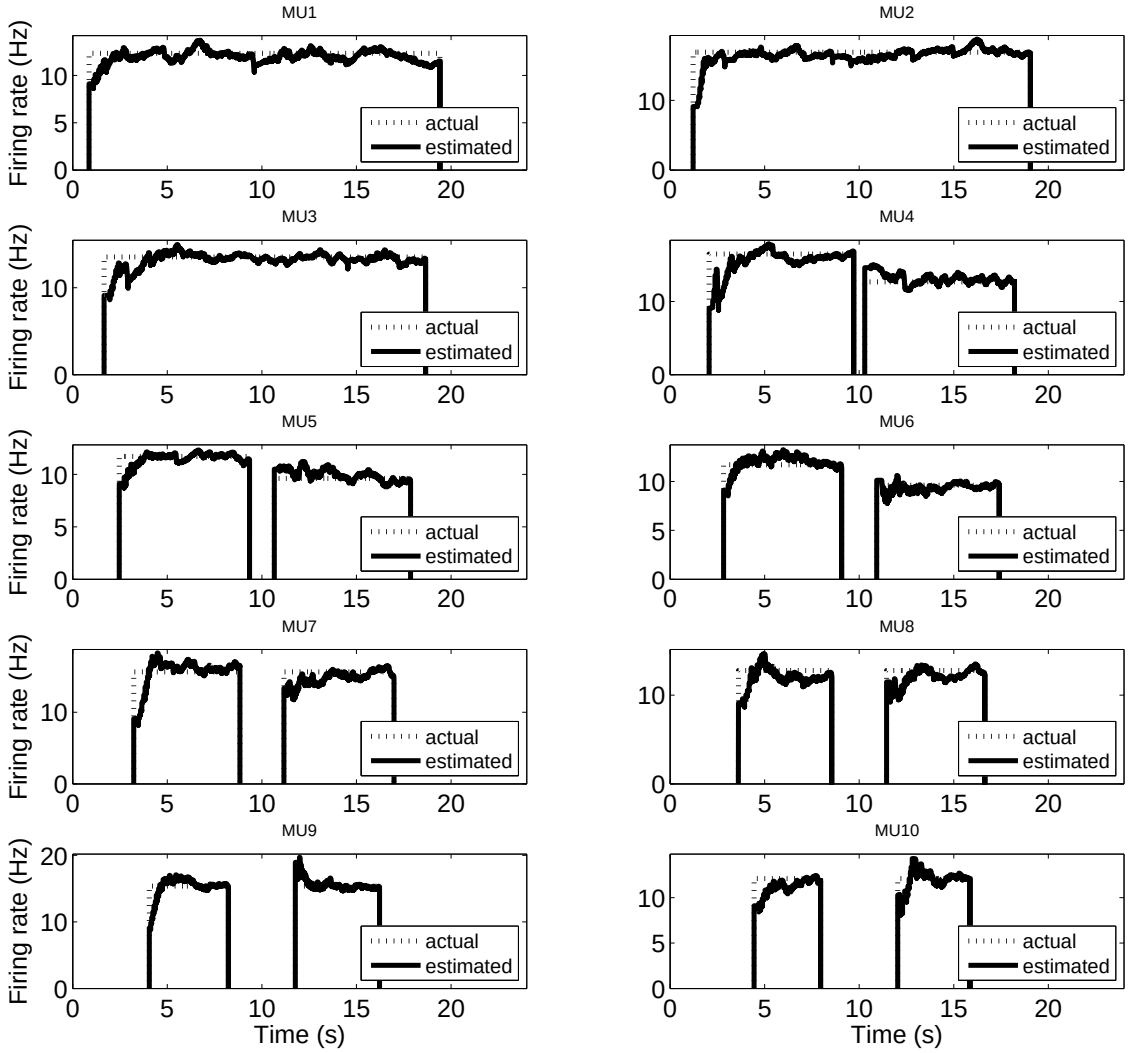


Figure 7.5 – Firing rates for the simulated iEMG (see figure 7.1): the dashed lines (empirical) represent the actual firing rates; continuous lines (estimated) represent the firing rates calculated via the estimated parameters of discrete Weibull distribution as described in section 4.3.

percentage of superposition ranged from 6.56% to 28.96%. Except the two signals recorded from muscle TA with 30% MVC, both the global sensitivity and predictivity of other signals were above 90%. The global sensitivity and predictivity of these two complex signals were more than 85%.

Detailed results of the decomposition are illustrated and analyzed in the following parts for two representative signals: a fine-wire signal from the TA with 8 MUs and a needle signal from the ADM muscle with 7 MUs.

Experimental signal with 8 MUs from the TA

Figure 7.6 provides a global view of the decomposition results. In the upper panel, the activation zone of each MU in the decomposition algorithm is correlated with the manual reference, which proves that the recruitment model presented in subsection 3.4.1 is accurate.

To provide a detailed view of the decomposition results, we have chosen a two seconds extract of the

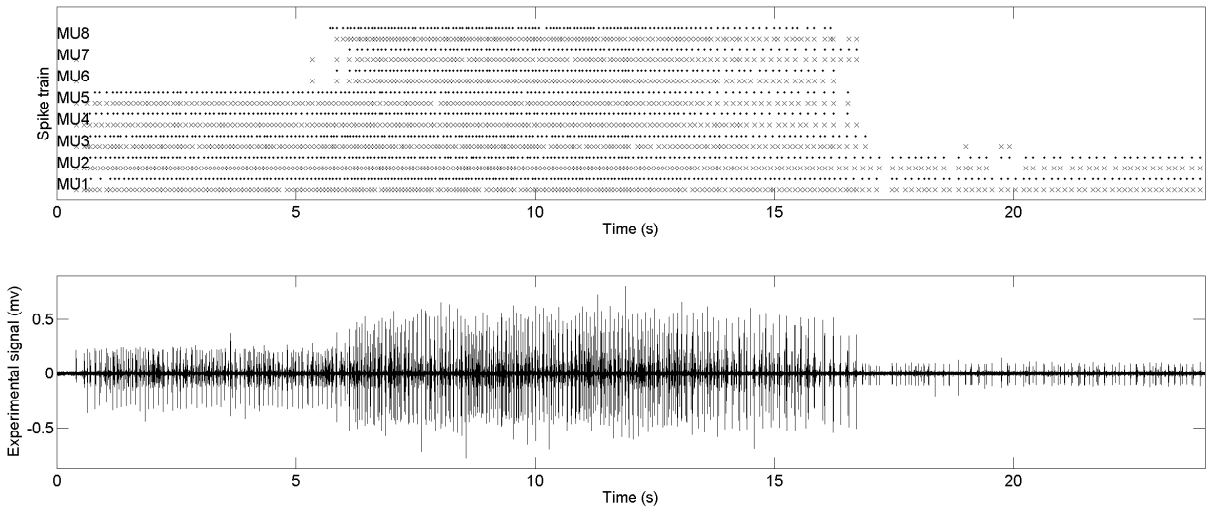


Figure 7.6 – Comparison of automatic (crosses, 'x') and reference (points, '.') decompositions (upper panel) and the experimental signal from TA, 30% MVC (lower panel).

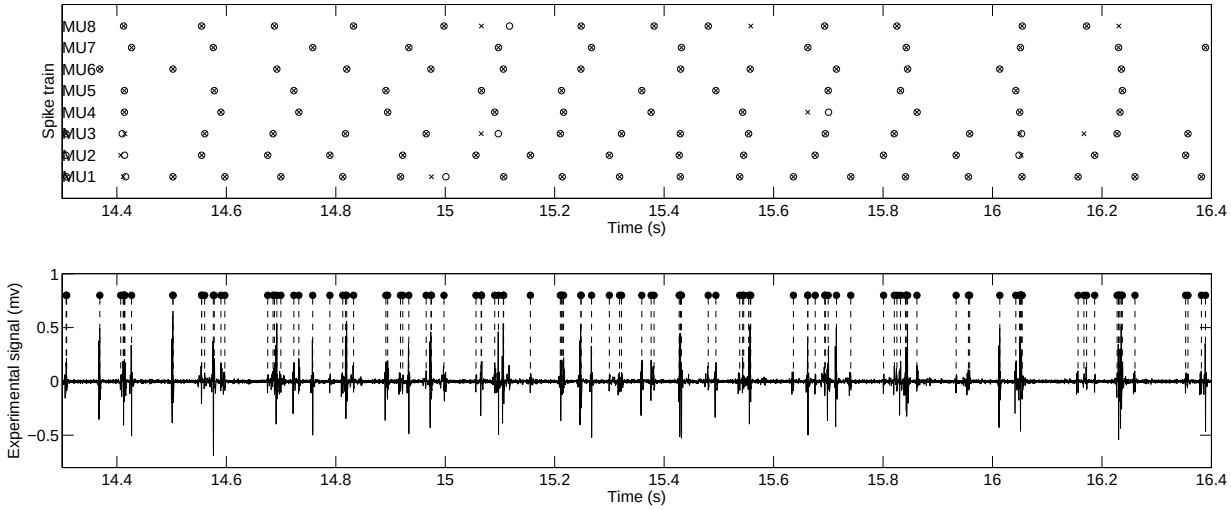


Figure 7.7 – An extract of the experimental signal decomposition shown in figure 7.6; circles '○' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions.

signal, containing the most challenging superposition cases (see figure 7.7). The algorithm performed generally well, successfully processing several complex superpositions. Due to the high complexity of signal, there are also a few mistakes in the classification. Moreover, it is also worth noting that a spike detection with limited precision, which occurred at 14.4 s and 16.05 s (see upper panel of figure 7.7).

For the classification phase, the individual (per MU) performance indexes are shown in table 7.3. Figure 7.8 illustrates MUAP waveforms of eight MUs. The last one is the comparison of MUAP waveforms between the 2nd one and the 3rd one. According to figure 7.8, we analyze the performance indexes in table 7.3. The reason for the lower sensitivity of the 2nd and 3rd MU is that they have the smaller amplitude of MUAP, compared to the others. Generally, this can lead to its complete masking in the superpositions. Furthermore, their MUAP waveforms are similar. Their maximum absolute difference between amplitudes is less than 0.1 mv, while the maximum absolute noise is 0.0348 mv and the standard deviation of noise

is 0.0074 mv. Thus, their classification is influenced by the noise and they switch occasionally with each other, as shown in figure 7.6 (two cases occurred at 20 s). With respect to the two MUs, others are well classified. Globally, the algorithm succeeded in tracking and decomposing the MUs.

Table 7.3 – Decomposition performance for an experimental signal detected from the TA with 8 MUs

MU	Sens.	Spec.	Acc.
MU1	91.53	97.87	96.54
MU2	83.26	96.19	93.85
MU3	73.66	95.54	92.26
MU4	91.87	98.42	97.53
MU5	95.36	99.40	98.88
MU6	97.22	99.52	99.31
MU7	94.95	99.43	99.05
MU8	86.89	96.47	95.49

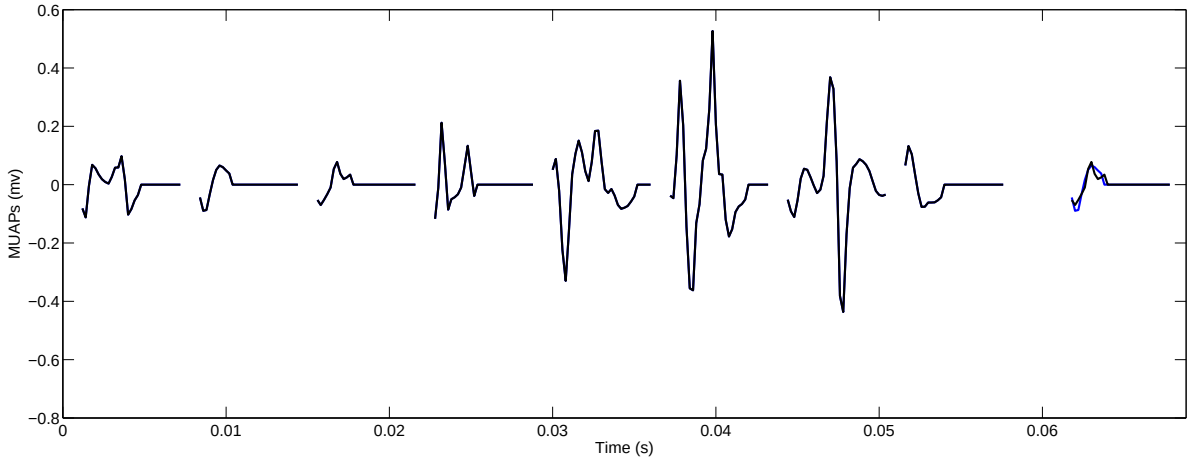


Figure 7.8 – Eight MUAP shapes (manually-extracted dictionary) for the signal presented in Figure 7.6, and a comparison between the 2nd one and the 3rd one.

As shown in section 4.3, the algorithm recursively estimates the parameters of the inter-spike intervals distribution. Figure 7.9 shows the corresponding results for each MU in the signal shown in Figure 7.6. Empirical firing rates were estimated as the inverse of the moving average of subsequent inter-spike intervals in the reference decomposition. The estimated firing rates were calculated via parameters t_0 and β using the approximation formula (3.16). The algorithm successfully tracked the changes in firing rates.

Experimental signal with 7 MUs from the ADM muscle

Figure 7.10 shows an example of a 5 s signal recorded during maximal abduction of the little finger (approximately 45 degrees) and its decomposition results. Since the algorithm initializes all the MUs as inactive, the beginning of the signal is associated to the simultaneous activation of all MUs. We note that the algorithm successfully handled this condition, as well as sporadic activation of the 7th MU.

One second extracted from this signal is presented in figure 7.11. All superposition cases, including complex ones (see signal at 3.1 s and at 3.85 s), were successfully decomposed.

The quantitative evaluation is provided in table 7.4. Compared to other MUs, the first is decomposed with relatively low specificity. Similarly to the previous case, this was due to the small size of its MUAP

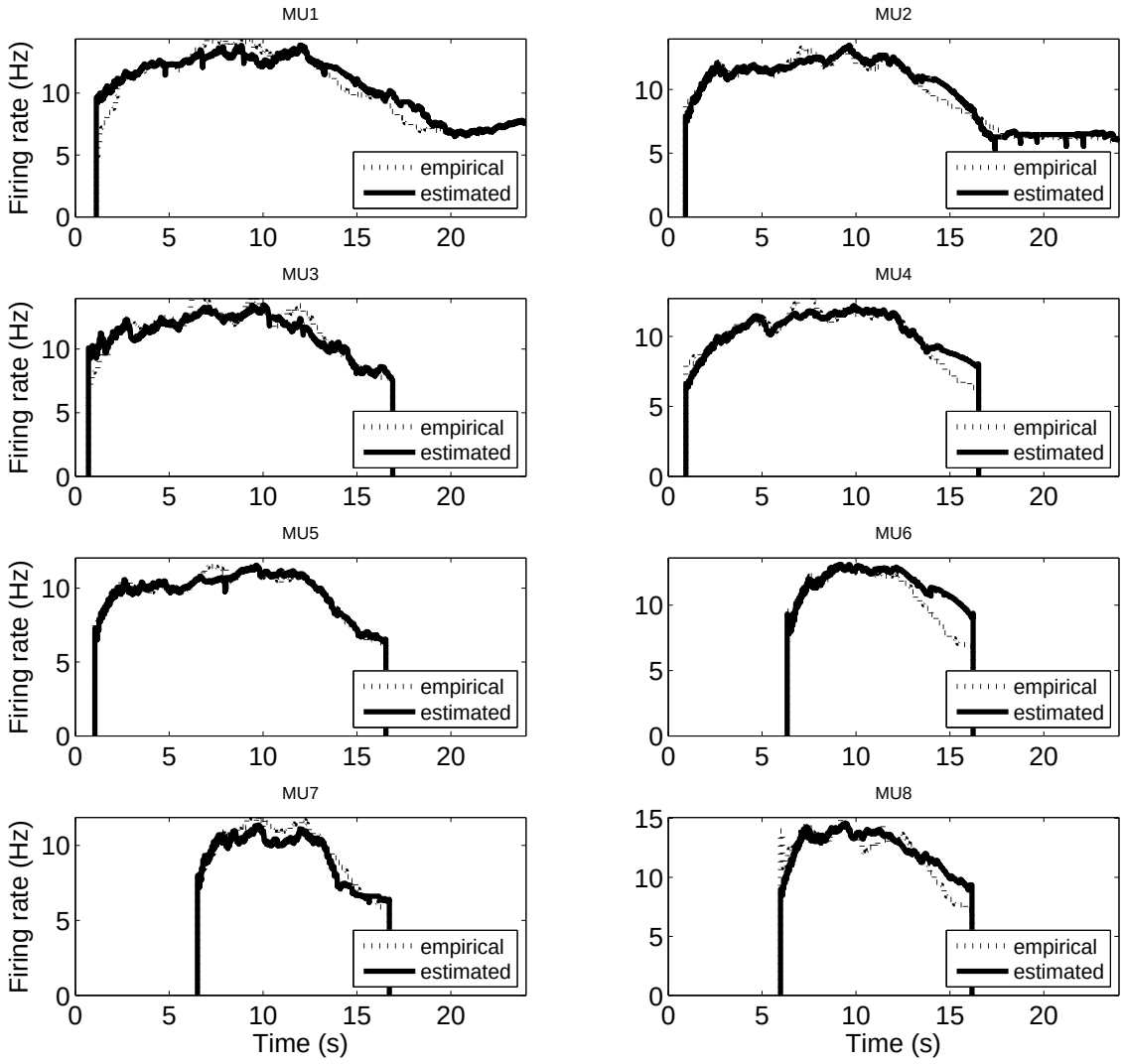


Figure 7.9 – Firing rates for the iEMG from TA set (see figure 7.6): the dashed lines (empirical) represent the firing rates estimated using reference decomposition; continuous lines (estimated) represent the firing rates calculated via the estimated parameters of discrete Weibull distribution as described in section 4.3.

Table 7.4 – Decomposition performance for an experimental signal (see Figure 7.10) from ADM set

MU	Sens.	Spec.	Acc.
MU1	81.69	97.29	94.26
MU2	90.48	97.63	96.37
MU3	98.33	99.31	99.14
MU4	91.53	97.98	96.91
MU5	100	100	100
MU6	97.56	99.35	99.14
MU7	100	99.37	97.42

and its similarity to that of the second MU. This is illustrated in figure 7.12 providing the corresponding dictionary of MUAP shapes and their final estimations by the algorithm. It can be noted that the decompo-

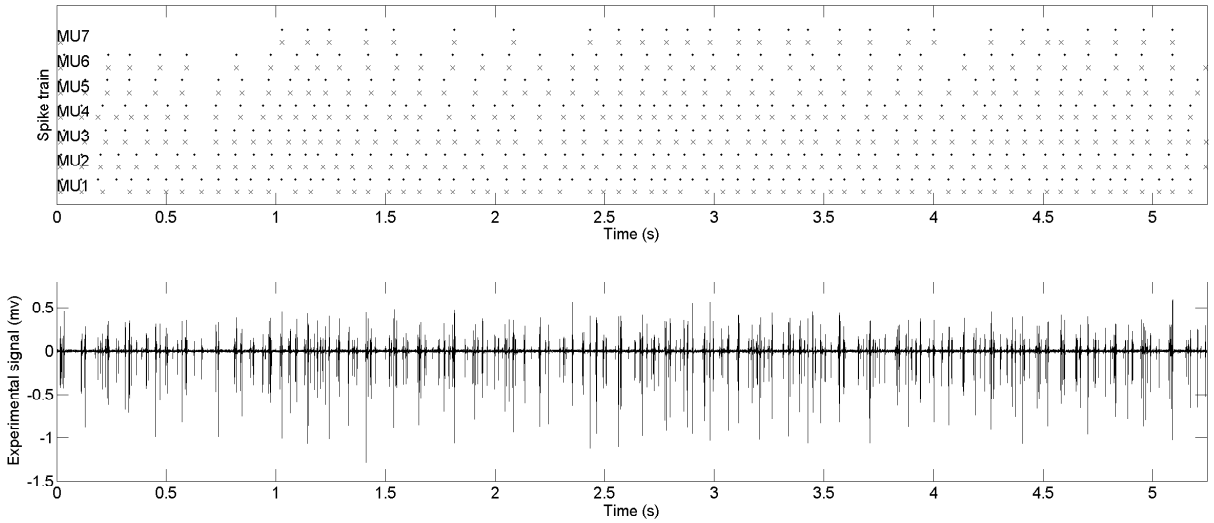


Figure 7.10 – Comparison of automatic (crosses, 'x') and reference (points, '.') decompositions (upper panel) and corresponding experimental signal from ADM, in position 5, corresponding to approximately 45 degrees of abduction (lower panel).

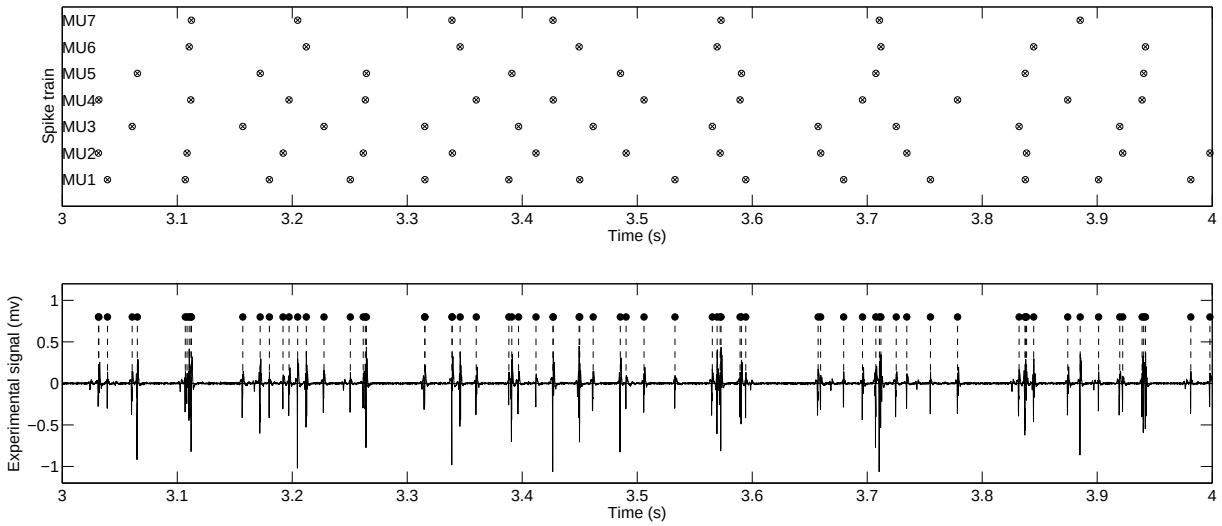


Figure 7.11 – An extract of the experimental signal decomposition shown in figure 7.10; circles '∘' and crosses 'x' represent respectively the spikes from the reference and automatic decompositions.

sition errors for the first MU mostly occur in the first half of the signal (figure 7.10), while the remaining part of the signal is decomposed correctly. This case demonstrates the ability of the algorithm to converge to the correct solution over time due to the recursive upgrades of the inter-spike interval statistics and the impulse responses.

Figure 7.13 shows the firing rates estimated from the reference decomposition ('empirical') and recursively by the algorithm ('estimated'). In this example, at first the firing rate of the 1st MU is underestimated due to several false negatives. However, the estimates recovers later, establishing a correct tracking, in consistence with the results presented in figure 7.10.

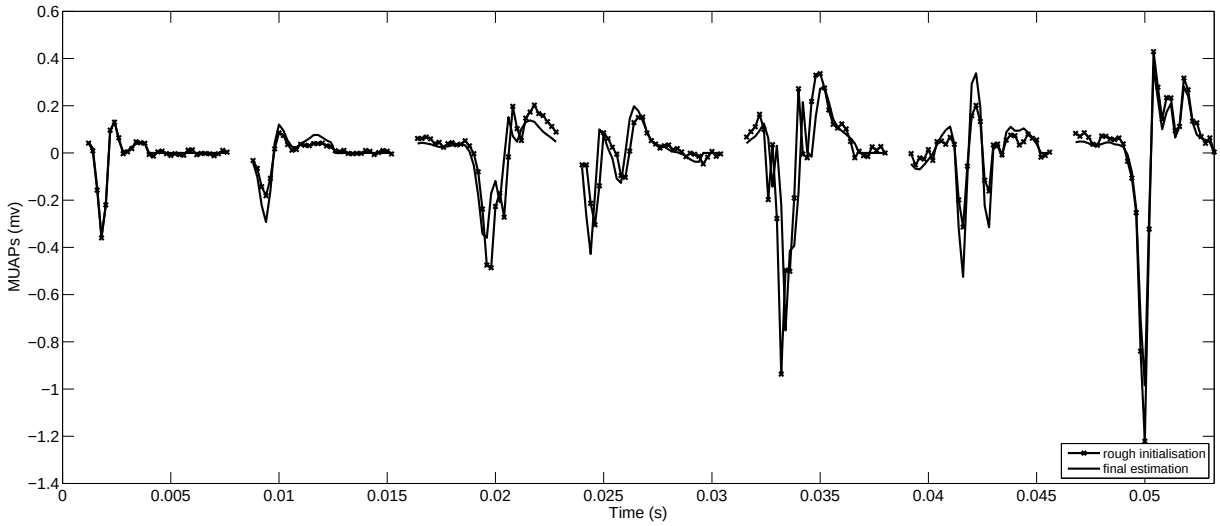


Figure 7.12 – Initial MUAP shapes (manually-extracted dictionary) and their final estimations for the signal presented in Figure 7.10

7.3 Decomposition velocity

To evaluate the decomposition velocity, the simulated signals described in subsection 6.3.1 and all 10 experimental signals presented in 7.2.2 were decomposed on a Nvidia Tesla K80 GPU card using double-precision floating-point format. Their execution time and related results are shown in the following of this section.

7.3.1 Experimental signals

Table 7.5 – Decomposition time of experimental signals recorded from muscle TA

Index	Duration (s)	Force (MVC%)	Nb MUs	Nb spikes	Sup.(%)	Nb paths	Time(s)
1	24	20	5	873	18.10	512	24.93
						384	21.32
						256	18.38
2	24	20	5	936	18.38	512	26.30
						384	22.58
						256	19.68
3	24	20	6	933	17.15	512	20.95
						384	18.55
						256	16.42
4	24	30	7	1176	22.28	512	26.12
						384	23.56
						256	20.16
5	24	30	8	1295	28.96	512	26.78
						384	23.31
						256	20.70

Decomposition time of experimental signals recorded from muscle TA was shown in table 7.5. Signals were decomposed by the algorithm with 256, 384 and 512 paths. The execution time of the algorithm is

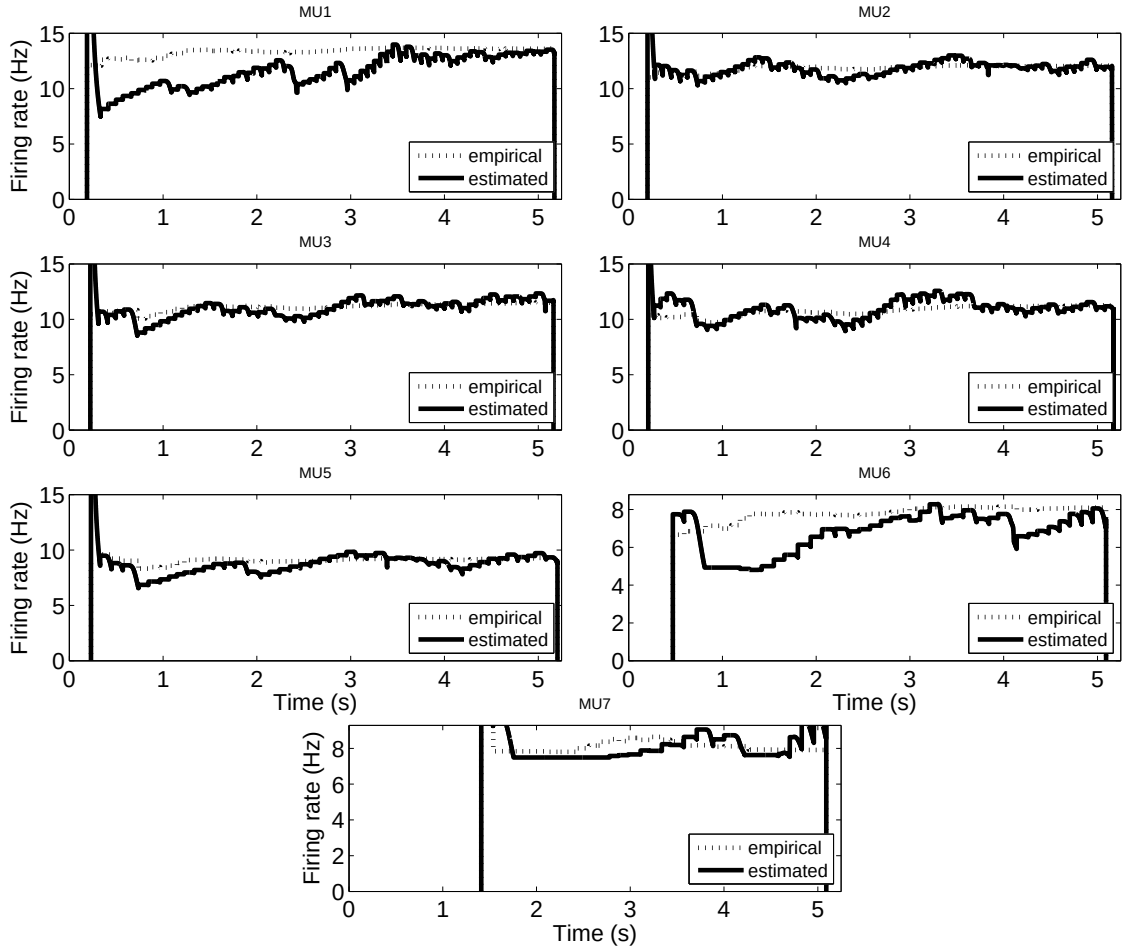


Figure 7.13 – Firing rates of the experimental signal with 7 MUs detected from ADM in the abduction position '5'

proportional to the number of paths. Because a large number of paths means large complexity of algorithm. We notice that all signals of set TA can be decomposed in real time with 384 paths.

In table 7.5, we do not give the related performance for various numbers of paths. Because our decomposition algorithm comprising a random process determines that for a signal decomposition, even with low number of paths, we have the probability to get a good result. For these experimental signals with different numbers of MUs and various length of MUAP waveforms, we cannot reveal the relation of the number of paths, the time execution and the decomposition performance. However, it will be much easier for the simulated decomposition. Thus, a more detailed discussion is given in subsection 7.3.2.

As demonstrated in table 7.6, two experimental signals recorded from muscle ADM, sampled with frequencies 10 kHz and 5 kHz, were decomposed by our algorithm with 256, 384, and 512 paths. The two signals are the most complex among all five signals of set ADM. The decomposition time is proportional to the sampling frequency of signal. In fact, in the low sampling frequency of signal, there are small number of sampled signals to process, thus, low execution time.

Moreover, we notice that the decomposition of signal 9 (index 9) sampled with frequency 5 kHz can be realised in real time with 256 paths, while the decomposition of signal 10 sampled with frequency 5 kHz cannot decomposed in real time with more than 256 paths. Compared to the decomposition of signals detected from muscle TA, the decomposition velocity of set ADM seems to be slower. The reasons are as following: Firstly, the signals of set ADM contain more numbers of MUAPs in every second than the

Table 7.6 – Decomposition time of experimental signals recorded from muscle ADM: '10 kHz' and '5 kHz' indicates the sampling frequency.

Index	Duration (s)	Position	Nb MUs	Nb spikes	Sup.(%)	Nb paths	Time (s)	
							10 kHz	5 kHz
9	5	4	6	281	23.13	512	11.09	6.58
						384	9.67	5.80
						256	8.49	4.92
10	5	5	7	371	28.84	512	12.32	7.60
						384	10.89	6.70
						256	9.36	5.66

TA ones, which means that more signal segments need to make the bifurcation and complete its related tasks in the recursive estimation, as presented in section 5.5, thus more time consumption. Secondly, with respect to the set TA, the duration of signals in set ADM is much shorter, only 5 seconds. The time spent on the allocation of memory in GPU and the memory copy of the initial variables from CPU to GPU, has a nonnegligible influence for the signal decomposition with short duration, especially in the case of high number of paths.

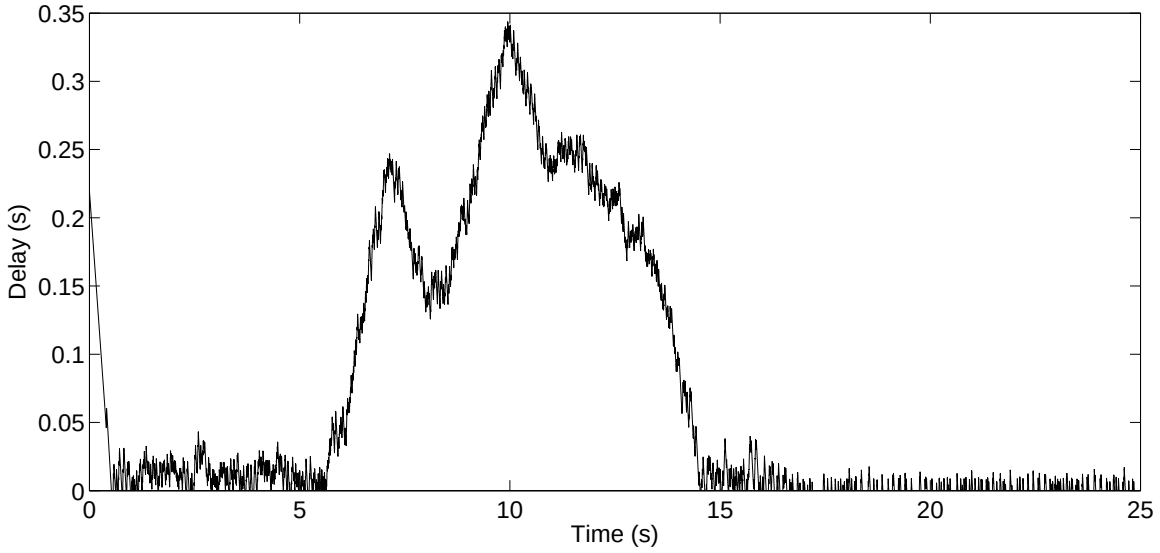


Figure 7.14 – Delay of experimental signal (recorded from muscle TA, with 8MUs and 30% MVC) decomposition with 256 paths

The decomposition time shown in tables 7.5 and 7.6 is the absolute execution time, without considering the sample timing of observation signal. If we take consideration of the sample timing of observation signal, we can measure the latency of our decomposition algorithm at every sample timing. An example of the latency decomposition of experimental signal is given in figure 7.14. This signal with 8 MUs is the one presented in subsection 7.2.2. In figure 7.14, it was decomposed with 256 paths. Since we need to allocate the memory of initial variable in CPU and GPU, and copy the memory from GPU to CPU, the decomposition has a latency about 200 ms in the beginning. Then, the variation of decomposition latency corresponds to the firing rates of signals, illustrated in figure 7.9. The maximum decomposition latency is 343.6 ms.

The threshold of latency for the real time controlling of a device, such as the active prosthetic devices, is 250 ms [129]. Evidently, the maximum latency in figure 7.14 exceeds this threshold. In table 7.7, we give the maximum decomposition latency of other signals, recorded from muscle TA and decomposed with

Table 7.7 – Delay of experimental signals recorded from muscle TA: the signal index corresponds to the signals presented in table 7.5

Index	1	2	3	4	5	5
Nb paths	256	256	256	256	256	192
Max Delay (ms)	29.4	27.2	22.4	149.5	343.6	44.8

256 paths. The signal index indicates the same signal presented in table 7.5. The maximum decomposition latencies of the first four signals, decomposed with 256 paths, respect the threshold of device controlling, as well as the 5th one decomposed with 192 paths.

7.3.2 Simulated signals

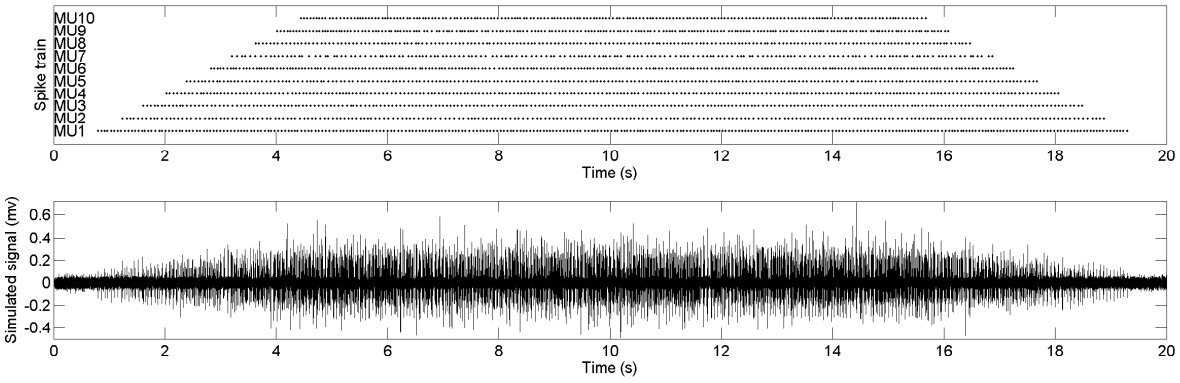


Figure 7.15 – Example of a simulated signal with 10 MUs: The recruitment profile is shown in the upper panel. Corresponding simulated signal with 10 dB SNR is depicted in the lower panel.

In order to reveal the relation of the number of paths, the decomposition time and the decomposition performance, a great number of simulated signals were decomposed. As presented in subsection 6.3.1, the parameters of simulated signals were set. Their recruitment profile is shown in the upper panel of Figure 7.15, while an example of a signal with 10 MUs is provided in the lower panel. Moreover, the maximum MUAP length after clipping presented in subsection 6.2.2 is set to 7.6 ms.

Table 7.8 – Decomposition performance of simulated signals

Nb MUs	Nb sup-spikes	Nb spikes	Sup.(%)	Nb paths	Sens. (%)	Pred. (%)	Time(s)
10	645.20±39.90	2093.10±80.59	30.83±1.91	384	94.98±2.51	92.26±2.60	30.34±0.84
				256	92.45±3.10	89.30±2.80	25.62±0.74
				128	83.32±6.49	80.42±5.16	19.95±0.45
8	446.40±28.52	1769.80±59.44	25.22±1.61	384	97.55±1.77	96.21±2.26	26.29±0.47
				256	96.43±2.14	94.71±2.65	23.31±0.41
				192	95.35±2.78	93.21±3.58	19.95±0.37
6	291.70±15.27	1469.80±52.49	19.85±1.04	384	99.06±1.07	98.44±1.55	23.75±0.56
				256	98.86±1.23	98.18±1.72	19.97±0.45

We note that the mean values of global sensitivity and predictivity (table 7.8) decrease for signals with larger number of active MUs. This is due to the fact that the decomposition task becomes more complex in terms of overall number of spikes and of the superposition percentage. We also note the increase in

standard deviations of the performance indexes, demonstrating the proportionality between task complexity and decrease in performance of the algorithm.

The execution time becomes large with the increasing number of paths. The signal with 10 MUs, 8 MUs, and 6 MUs can be decomposed in real time, with respectively 128 paths, 192 paths and 256 paths. Although other decompositions with larger number of paths cannot realise the on-line decomposition, they can be also completed in a relatively short time, with higher accuracy. Thus, the number of paths n_{path}, as a parameter determined by users, should be chosen as a trade-off between the computational complexity and the sub-optimality of the solution.

7.4 Discussion and conclusion

As shown in this chapter, we have completed the decomposition of 10 experimental iEMG signals acquired from two different muscles, respectively by fine wire electrodes and needle electrodes. The number of MUs ranges from 2 to 8. The percentage of superposition, representing the complexity of iEMG signal, ranges from 6.56 $\%$ to 28.84 $\%$. The accuracies of almost all experimental iEMG signals are more than 90 $\%$, except two signals at 30 $\%$ MVC (more than 85 $\%$). We realized the real-time decomposition for all these experimental signals by the parallel implementation.

Furthermore, we decomposed dozens of simulated signals, whose results, in terms of decomposition performance and velocity, revealing that the decomposition accuracy and robustness are inversely correlated to the decomposition velocity. We need to find the a balance point to achieve the real time decomposition with high quality performance.

8

Conclusion and perspectives

8.1 Conclusion

We have validated the iEMG decomposition algorithm proposed in [84, 85]. The modification of activation process of recruitment model was proposed and several iEMG signal preprocessing techniques were applied. Then, we accelerated this decomposition algorithm: Firstly, we replace the Kalman filter by a more computationally efficient LMS filter for the MUAPs estimation procedure in Bayes filter. Direct comparisons on simulated signal have shown no significant discrepancy between the two estimators. Secondly, we introduced two heuristics helping to reduce number of bifurcation by discarding highly improbable paths: detector of active segments and by prohibition of simultaneous firings. Thirdly, we have shown the parallel computation implementation of the algorithm in the GPU. The proposed algorithm can be appropriately broken down into a number of elementary tasks, each involving simultaneous processing of a large amount of data. Some tasks were parallelized in time due to their inter-independence. Additionally, efficient parallel sorting and GPU memory management were presented.

The proposed parallel computing algorithm was validated using a great many simulated signals and ten experimental iEMG signals acquired from two muscles (TA and ADM) by fine wire and needle electrodes. The global sensitivity and predictivity of almost all experimental signals are more than 90%, except two signals recorded from TA with 30% MVC having the global performance index more than 85%. We can realize the real time decomposition for the simulated and experimental signals by our algorithm with a proper number of paths. The number of paths n_{path}, determined by users, is always selected as a trade-off between the computational complexity and the sub-optimality of the solution.

Our parallel decomposition algorithm is the first one that realizes the real time full decomposition of single channel iEMG signal with number of MUs up to 10, where full decomposition means resolving the superposition problem. For the signals with more than 10 MUs, we can also decompose them quickly, but not reaching the real time level.

This is an important progress to the precise control of active prosthetic devices for amputees with EMG signals. We have almost finished the first step: extraction of critical information in EMG signals in real-time. Although there are still some limitations for our parallel computation algorithm, it allows us to start exploiting the correlation between muscle signals information and the kinematic coefficient of the movement.

8.2 Perspectives

In our parallel computation algorithm, some parameters, such as: the length of window for the tracking, the number of scenarios and so on, which need to be pre-defied before decomposition, influence greatly the decomposition performance and velocity. The prior knowledge for these parameters is crucial for the decomposition. In the future, we would like to propose the estimation algorithm for them. For example, the length of window for the tracking may increase or decrease according to the derivation of firing rates variation.

Other possible limitations of the algorithm arise from large differences of amplitudes between MUAPs (masking of small action potentials) and from the similar MUAP waveforms (occasional switching between similar units). A multichannel version of the presented algorithm may resolve this problem. We briefly note that it can be obtained by an appropriate expansion of the observation model (3.17) which will only affect the MUAP estimation procedure. Or the multichannel signals can be separately decomposed in multi-GPUs by our algorithm and then make a fusion of their results.

There is also a certain limit to the number of MUs that can be simultaneously tracked by the algorithm in the real-time operation. These limits may be overcome in future by a better hardware or further simplifications of mathematical model. Furthermore, we can also reduce the search space using other heuristics or a more elaborate activity detection algorithm.

Based on the proposed parallel computation algorithm, we could continue to step towards the precise control of active prosthetic devices for amputees with EMG signals. In the future, we will make the correlation between the decomposition results and the kinematic parameters of the movement.

9

Appendix

9.1 Proof of the inter-spike law parameters estimation

To estimate the discrete Weibull distribution parameters, a maximum likelihood (ML) estimator is proposed in [130]. In [84], an online ML estimator is implemented. The likelihood is optimized iteratively by the quasi-Newton method. The ML of $\hat{\theta}_{i,S^n}$ is:

$$\hat{\theta}_{i,S^n} = \arg\min_{\theta} \underbrace{-\frac{1}{n} \ln \mathsf{Pr}(S^n = s^n | \Theta_i = \theta_i)}_{J_{i,S^n}(\theta)} \tag{9.1}$$

$J_{i,S^n}(\theta)$ is the objective function to minimize that is the great difference of this discrete Weibull distribution on-line ML estimator with others. In the classic discrete Weibull distribution ML estimators, $-\frac{1}{n} \ln \mathsf{Pr}(\triangle^n = l^n | \Theta_i = \theta_i)$ is usually interpreted as the objective function where $\triangle^n$ denotes the inter-spikes. The change of objective function leads to a practical on-line implementation.

Considering the Markov chain property of sawtooth sequences, the objective function $J_{i,S^n}(\theta)$ can be written for all $n \geq 1$:

$$\begin{aligned} J_{i,S^n}(\theta) =& \frac{1}{n} J_{i,S^1}(\theta) + \frac{1}{n} \sum_{k=2}^{n} Q_{i,S^n}(\theta) \\ J_{i,S^1}(\theta) =& -\ln \mathsf{Pr}(S[1] = s[1] | \Theta_i = \theta_i) \\ Q_{i,S^n}(\theta) =& -\ln \mathsf{Pr}(S[k] = s[k] | \Theta_i = \theta_i, S[k-1] = s[k-1]) \end{aligned} \tag{9.2}$$

where $J_{i,S^1}(\theta)$ and $Q_{S^n}(\theta)$ are computed by the formulas (3.12) and (3.13). The gradients of the two members of objective function are directly computed with the transition probability of the discrete Weibull law. The objective function is recursively minimized by a stochastic gradient update:

$$\hat{\theta}_{i,S^n} = \hat{\theta}_{i,S^{n-1}} - \frac{1}{n} G^{-1}_{i,S^n} Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}}) \tag{9.3}$$

Where G_{i,S^n} is an approximation of Hessian matrix obtained through the Fisher information matrix of the transition probability law [131], $Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})$ is the partial derivative Q_{i,S^n} with respect to the components of $\hat{\theta}_{i,S^{n-1}}$. For all $n \geq 2$, we have:

$$\begin{aligned} G_{i,S^n} =& \frac{1}{n}[Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})][Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})]^\top + \\ & (1 - \frac{1}{n}) G_{i,S^{n-1}} \end{aligned} \tag{9.4}$$

We notice that the approximation of Hessian matrix G_{i,S^n} should be invertible in (9.3). If the matrix G_{i,S^n} is not invertible, we update the inter-spike law parameters $\hat{\theta}_{i,S^n} = [\hat{t}_{0,i,S^n}; \hat{\beta}_{i,S^n}]$ using following formula:

$$\begin{aligned} \hat{t}_{0,i,S^n} =& \hat{t}_{0,i,S^{n-1}} - \frac{1}{n} \frac{Q'_{i,S^n}(\hat{t}_{0,i,S^{n-1}})}{G_{i,S^n}(1,1)} \\ \hat{\beta}_{i,S^n} =& \hat{\beta}_{i,S^{n-1}} \end{aligned} \tag{9.5}$$

Where $Q'_{i,S^n}(\hat{t}_{0,i,S^{n-1}})$ is the partial derivative Q_{i,S^n} with respect to the components of $\hat{t}_{0,i,S^{n-1}}$, and $G_{i,S^n}(1,1)$ is the first element of the approximation of Hessian matrix G_{i,S^n}.

The estimation procedure should take into consideration the activation and deactivation of the source. In order to do that, we replace index n by an active time index τ with following definition:

$$\tau_{i,S^n} = \begin{cases} \tau_{i,S^{n-1}} + 1 & \text{if } i \in A[n] \\ \tau_{i,S^{n-1}} & \text{if } i \notin A[n] \end{cases} \tag{9.6}$$

Moreover, we define that the estimated inter-spike law parameters and the approximation of Hessian matrix keep the same value as in previous time instant if the source is not active.

In conclusion, for all $n \geq 1$, we have:

— if $i \notin A[n] \cap A[n-1]$,

$$\begin{cases} \hat{\theta}_{i,S^n} = \hat{\theta}_{i,S^{n-1}} \\ G_{i,S^n} = G_{i,S^{n-1}} \end{cases} \tag{9.7}$$

— if $i \in A[n] \cap A[n-1]$ and G_{i,S^n} is invertible,

$$\begin{aligned} \hat{\theta}_{i,S^n} =& \hat{\theta}_{i,S^{n-1}} - \frac{1}{\tau_{i,S^n}} G^{-1}_{i,S^n} Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}}) \\ G_{i,S^n} =& \frac{1}{\tau_{i,S^n}} [Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})][Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})]^\top + \\ & (1 - \frac{1}{\tau_{i,S^n}}) G_{i,S^{n-1}} \end{aligned} \tag{9.8}$$

— if $i \in A[n] \cap A[n-1]$ and G_{i,S^n} is not invertible,

$$\begin{aligned} \hat{t}_{0,i,S^n} =& \hat{t}_{0,i,S^{n-1}} - \frac{1}{\tau_{i,S^n}} \frac{Q'_{i,S^n}(\hat{t}_{0,i,S^{n-1}})}{G_{i,S^n}(1,1)} \\ \hat{\beta}_{i,S^n} =& \hat{\beta}_{i,S^{n-1}} \\ G_{i,S^n} =& \frac{1}{\tau_{i,S^n}} [Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})][Q'_{i,S^n}(\hat{\theta}_{i,S^{n-1}})]^\top + \\ & (1 - \frac{1}{\tau_{i,S^n}}) G_{i,S^{n-1}} \end{aligned} \tag{9.9}$$

9.2 From Kalman filter to the least-mean-square filter

Kalman filter, originally used for MUAPs estimation, can be replaced by an LMS filter under specific assumptions. Let us consider the state covariance matrix from (4.10):

$$\begin{aligned} P_{S^n} =& P_{S^{n-1}} - K_{S^n} \, v_{S^n} \, K^\top_{S^n} \\ =& P_{S^{n-1}} - P_{S^{n-1}} \, \psi(S[n])^\top \, v^{-1}_{S^n} \, v_{S^n} \\ & (P_{S^{n-1}} \, \psi(S[n])^\top \, v^{-1}_{S^n})^\top \\ =& P_{S^{n-1}} - P_{S^{n-1}} \, \psi(S[n])^\top \, v^{-1}_{S^n} \, \psi(S[n]) P_{S^{n-1}} \end{aligned} \tag{9.10}$$

Applying the Woodbury matrix identity:

$$[A + BCD]^{-1} = A^{-1} - A^{-1}B[DA^{-1}B + C^{-1}]^{-1}DA^{-1} \tag{9.11}$$

to (9.10), we obtain:

$$\begin{aligned} P_{S^n}^{-1} =& P_{S^{n-1}}^{-1} + \psi(S[n])^\top \left(v_{S^n} - \psi(S[n])\, P_{S^{n-1}}\, \psi(S[n])^\top\right) \\ & \psi(S[n]) \end{aligned} \tag{9.12}$$

This can be simplified using expression (4.9) for the variance of innovation:

$$P_{S^n}^{-1} = P_{S^{n-1}}^{-1} + \psi(S[n])^\top\, v^{-1}\, \psi(S[n]) \tag{9.13}$$

where v is the variance of measurement noise $\hat{V}^{|n}$ estimated using (4.11) and (4.12).

Finally, we have:

$$\begin{aligned} P_{S^n} &= \frac{\hat{V}^{|n}}{n}\, R_{S^n}^{-1} \\ R_{S^n} &= \frac{1}{n} \sum_{k=1}^{n} \psi(S[k])^\top \psi(S[n]) \end{aligned} \tag{9.14}$$

where R_{S^n} can be approximated by a constructed made of $\text{card}(\Omega) \times \text{card}(\Omega)$ blocks R_{i,j,S^n} with dimension $O(\ell_{\text{IR}} \times \ell_{\text{IR}})$:

$$R_{i,i,S^n} = \begin{bmatrix} \xi_{i,S^n} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \xi_{i,S^n} \end{bmatrix} \tag{9.15}$$

$$R_{i,j,S^n} = \begin{bmatrix} \xi_{i,S^n}\xi_{j,S^n} & \cdots & \xi_{i,S^n}\xi_{j,S^n} \\ \vdots & \ddots & \vdots \\ \xi_{i,S^n}\xi_{j,S^n} & \cdots & \xi_{i,S^n}\xi_{j,S^n} \end{bmatrix} \tag{9.16}$$

where ξ_{i,S^n} is the firing rate of i-th motor unit, which is the inverse of its inter-spike interval (ISI) expected value. We can notice that $\forall i, j \in \Omega,\ \xi_{i,S^n}\xi_{j,S^n} \ll \xi_{i,S^n}$ and $\xi_{i,S^n}\xi_{j,S^n} \ll \xi_{j,S^n}$. Therefore, if $i \neq j$, R_{i,j,S^n} can be approximated by a zero-matrix, R_{S^n} can be approximated by diagonal matrix:

$$R_{S^n} = \begin{bmatrix} \xi_{1,S^n} & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \xi_{1,S^n} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & \xi_{\Omega,S^n} & \cdots & 0 \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & \xi_{\Omega,S^n} \end{bmatrix} \tag{9.17}$$

Having the approximation of P_{S^n} with the formulas (9.14) and (9.17), we can derive an expression for LMS filter from that of the Kalman filter (4.10). With a rough initial prior $\hat{H}_{S^0}^{|0}$, for all $n \geq 1$, we have:

$$\begin{aligned} \epsilon[n] &= Y[n] - \psi(S[n])\, \hat{H}_{S^{n-1}}^{|n-1} \\ m_{\Delta,i}[n] &= \frac{\sum_j \Delta_i[j]}{\text{card}(\Delta_i)} \\ \tilde{v}[n] &= 1 + \sum_i m_{\Delta,i}[n]\, \varphi_i(S[n])\varphi_i(S[n])^\top \\ \hat{H}_{i,S^n}^{|n} &= \hat{H}_{i,S^{n-1}}^{|n-1} + \frac{m_{\Delta,i}[n]\varphi_i(S[n])\epsilon[n]}{n\, \tilde{v}[n]} \end{aligned} \tag{9.18}$$

where $\Delta_i[j]$ denotes the j-th ISI of the i-th motor unit; card(Δ_i) is the number of the ISIs for the i-th motor unit; $m_{\Delta,i}[n]$ is the expected ISI for i-th motor unit at time index n; and $\tilde{v}[n]$ represents the ratio of the variance of innovation v_{S^n} to the variance of noise $\hat{V}^{|n}$. And the prediction of the variance of innovation v_{S^n} is:

$$v_{S^n} = \tilde{v}[n]\ \hat{V}^{|n}. \tag{9.19}$$

In order to make this filter adaptive to the changes in MUAPs forms, time index n can be replaced by a forgetting factor $l[n]$.

9.3 Proof of normalised least-mean-square filter

As LMS filter, a NLMS filter is proposed to reduce the processing time of impulse responses estimation. Despite this method is more complex than the LMS filter, it also shows a good performance.

The impulse responses optimization problem is described as following:

$$\begin{aligned} H_i[n] &= H_i[n-1] + W_{H_i}[n],\ \ \forall i \in \Omega \\ Y[n] &= \sum_{i\in\Omega} \varphi_i(S[n])\, H_i[n] + W[n] \end{aligned} \tag{9.20}$$

As known, the impulse responses are time varying. We assume that $H_i[n]$ is a random vector, where i denotes the index of MU. A sight change $W_{H_i}[n]$, which is a zero-mean white Gaussian noise signal vector, occurs at every time index. $W_{H_i}[n]$ is uncorrelated with $H_i[n-1]$ and its correlation matrix is assumed to be $R_{W_{H_i}} = \sigma^2_{W_{H_i}}\, I_{\ell_{\text{IR}}}$, where $I_{\ell_{\text{IR}}}$ is a $\ell_{\text{IR}} \times \ell_{\text{IR}}$ identity matrix.

9.3.1 The first case: no MU firing

If there is not MU firing at time index n, we have:

$$\hat{H}^{|n}_{i,S^n} = \hat{H}^{|n-1}_{i,S^{n-1}},\ \ \forall i \in \Omega \tag{9.21}$$

9.3.2 The second case: there is a MU firing

In this case, there is not superimposed MUAP shapes at time index n. Only the i-th MU is firing. Impulse responses of other MUs are updated with the formula (9.21). The derivation process of the i-th MU impulse responses update based on [132] is shown in the rest of this subsection.

To simplify the formula, we define $X_i^\top[n] = \varphi_i(S[n])$. As the Kalman filter and the LMS filter, $\hat{H}^{|n}_{i,S^n}$ is defined as the estimated impulse response of the i-th MU. Thus, We define the conventional NLMS update formula [113]:

$$\hat{H}^{|n}_{i,S^n} = \hat{H}^{|n-1}_{i,S^{n-1}} + \frac{\mu[n]}{X_i^\top[n]X_i[n] + \delta}\, X_i[n]e[n] \tag{9.22}$$

where $\mu[n]$ is the normalised step-size parameter, δ is the regularization term which avoids denominator to be too small, and $e[n]$ is the a priori error of this adapt filter:

$$e[n] = y[n] - X_i^\top[n]\hat{H}^{|n-1}_{i,S^{n-1}} \tag{9.23}$$

Then, we define the a posteriori misalignment as $z_i[n] = H_i[n] - \hat{H}^{|n}_{i,S^n}$. Therefore, according to the

formula (9.22), we have:

$$\begin{aligned} H_i[n] - \hat{H}^{|n}_{i,S^n} &= H_i[n] - (\hat{H}^{|n-1}_{i,S^{n-1}} + \frac{\mu[n]}{X_i^\top[n]X_i[n]+\delta}\, X_i[n]e[n]) \\ z_i[n] &= H_i[n-1] + W_{H_i}[n] - \hat{H}^{|n-1}_{i,S^{n-1}} - \frac{\mu[n]}{X_i^\top[n]X_i[n]+\delta}\, X_i[n]e[n] \\ z_i[n] &= z_i[n-1] + W_{H_i}[n] - \frac{\mu[n]}{X_i^\top[n]X_i[n]+\delta}\, X_i[n]e[n] \end{aligned} \tag{9.24}$$

Our objective is to minimize the $z_i[n]$ by estimating $\mu[n]$ and δ. It is known that $\mathsf{E}(X_i^\top[n]X_i[n]) = \ell_{\text{IR}}\sigma^2_{X_i}$. Thus, for a $\ell_{\text{IR}} \gg 1$, we have $X_i^\top[n]X_i[n] \approx \ell_{\text{IR}}\sigma^2_{X_i}$ [133]. We notice that the term $\frac{\mu[n]}{X_i^\top[n]X_i[n]+\delta} \approx \frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}$ contains two determined parameters.

Under these conditions, we take the expectations of ℓ_2 norm in both sides of formula (9.24), then remove the uncorrelated terms. We obtain:

$$\begin{aligned} \mathsf{E}(\|z_i[n]\|_2^2) =& \mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma^2_{W_{H_i}} - \frac{2\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}\mathsf{E}(X_i^\top[n]z_i[n-1]e(n)) \\ & - \frac{2\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}\mathsf{E}(X_i^\top[n]W_{H_i}[n]e(n)) + \frac{(\mu[n])^2}{(\ell_{\text{IR}}\sigma^2_{X_i}+\delta)^2}\mathsf{E}(e^2[n]X_i^\top[n]X_i[n]) \\ \mathsf{E}(\|z_i[n]\|_2^2) =& \mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma^2_{W_{H_i}} - \frac{2\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}(\mathsf{E}(X_i^\top[n]z_i[n-1]e(n)) \\ & + \mathsf{E}(X_i^\top[n]W_{H_i}[n]e(n))) + (\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta})^2\mathsf{E}(e^2[n]X_i^\top[n]X_i[n]) \end{aligned} \tag{9.25}$$

The problem turn to find a proper $\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}$ to minimise $\mathsf{E}(\|z_i[n]\|_2^2)$. We find easily that the right sides of formula (9.25) is a polynomial by taking $\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}$ as an unknown variable.

$$\begin{aligned} \frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta} = \underset{\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}}{\arg\min}\ & \mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma^2_{W_{H_i}} - \frac{2\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta}(\mathsf{E}(X_i^\top[n]z_i[n-1]e(n)) \\ & + \mathsf{E}(X_i^\top[n]W_{H_i}[n]e(n))) + (\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta})^2\mathsf{E}(e^2[n]X_i^\top[n]X_i[n]) \end{aligned} \tag{9.26}$$

Thus, we obtain:

$$\frac{\mu[n]}{\ell_{\text{IR}}\sigma^2_{X_i}+\delta} = \frac{\mathsf{E}(X_i^\top[n]z_i[n-1]e(n)) + \mathsf{E}(X_i^\top[n]W_{H_i}[n]e(n))}{\mathsf{E}(e^2[n]X_i^\top[n]X_i[n])} \tag{9.27}$$

Now, we focus on the three terms in the right side of formula (9.27). First, we develop the formula of priori error (9.23):

$$\begin{aligned} e[n] =& y[n] - X_i^\top[n]\hat{H}^{|n-1}_{i,S^{n-1}} \\ e[n] =& X_i^\top[n]H_i[n] + W[n] - X_i^\top[n]\hat{H}^{|n-1}_{i,S^{n-1}} \\ e[n] =& X_i^\top[n]z_i[n-1] + X_i^\top[n]W_{H_i}[n] + W[n] \end{aligned} \tag{9.28}$$

Next, we introduce this formula to these three terms. The input signal $X_i^\top[n]$, $W_{H_i}[n]$ representing variation of $H_i[n]$ and $z_i[n-1]$ the misalignment are manually uncorrelated. Furthermore, as presented in HMM model (Section 3.5), $X_i[n]$ is a vector composed of 0 and 1, and there is at most one component equalling to 1. Thus, we find that $\mathsf{E}(X_i[n]\, X_i^\top[n]) = \sigma^2_{X_i} I_{\ell_{\text{IR}}}$ and the correlation matrix of $z_i[n-1]$ is a matrix diagonal. We assume that the correlation matrix of $W_{H_i}[n]$ is also a matrix diagonal. Then, we obtain the following

formula:

$$\begin{aligned}\mathsf{E}(X_i^\top[n]z_i[n-1]e(n)) =&\mathsf{E}(z_i^\top[n-1]X_i[n]X_i^\top[n]z_i[n-1])\\ =&\mathsf{E}(\mathrm{tr}(z_i[n-1]z_i^\top[n-1]X_i[n]X_i^\top[n]))\\ \approx&\mathrm{tr}(\mathsf{E}(z_i[n-1]z_i^\top[n-1])\mathsf{E}(X_i[n]X_i^\top[n]))\\ \approx&\sigma_{X_i}^2\mathsf{E}(\|z_i[n-1]\|_2^2)\end{aligned} \tag{9.29}$$

In the same manner, we have:

$$\mathsf{E}(X_i^\top[n]W_{H_i}[n]e(n) \approx \ell_{\text{IR}}\sigma_{X_i}^2\sigma_{W_{H_i}}^2 \tag{9.30}$$

Based on the Gaussian moment factoring theorem, the denominator of the right side of formula (9.27) can be approximated:

$$\mathsf{E}(e^2[n]X_i^\top[n]X_i[n]) \approx \ell_{\text{IR}}\sigma_{X_i}^2\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_i}^4(\sigma_{W_{H_i}}^2 + \mathsf{E}(\|z_i[n-1]\|_2^2)) \tag{9.31}$$

We substitute formulas (9.29), (9.30) and (9.31) into (9.27):

$$\begin{aligned}\frac{\mu[n]}{\ell_{\text{IR}}\sigma_{X_i}^2+\delta} =&\frac{\sigma_{X_i}^2\mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{X_i}^2\sigma_{W_{H_i}}^2}{\ell_{\text{IR}}\sigma_{X_i}^2\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_i}^4(\sigma_{W_{H_i}}^2 + \mathsf{E}(\|z_i[n-1]\|_2^2))}\\ \frac{\mu[n]}{\ell_{\text{IR}}\sigma_{X_i}^2+\delta} =&\frac{\mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{W_{H_i}}^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_i}^2(\sigma_{W_{H_i}}^2 + \mathsf{E}(\|z_i[n-1]\|_2^2))}\end{aligned} \tag{9.32}$$

where $\hat{V}^{|n}$ is the variance of noise in the formula (4.12).

Finally, according to formulas (9.22), (9.23), (9.25) and (9.32), we obtain a recursive optimization formula:

$$\begin{aligned}e[n] =&y[n] - X_i^\top[n]\hat{H}_{i,S^{n-1}}^{|n-1}\\ \hat{H}_{i,S^n}^{|n} =&\hat{H}_{i,S^{n-1}}^{|n-1} + \frac{\mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{W_{H_i}}^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_i}^2(\sigma_{W_{H_i}}^2 + \mathsf{E}(\|z_i[n-1]\|_2^2))}X_i[n]e[n]\\ \mathsf{E}(\|z_i[n]\|_2^2) =&\mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{W_{H_i}}^2 - \frac{\sigma_{X_i}^2(\ell_{\text{IR}}\sigma_{W_{H_i}}^2 + \mathsf{E}(\|z_i[n-1]\|_2^2)^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_i}^2\mathsf{E}(\|z_i[n-1]\|_2^2) + \ell_{\text{IR}}^2\sigma_{X_i}^2\sigma_{W_{H_i}}^2}\end{aligned} \tag{9.33}$$

9.3.3 The third case: there are more than one MU firing

This is the case of superposition. We define the set $F[n]$ containing all MUs firing at time index n. Thus, the input signal defined as $X_{F[N]}[n]$, is a $\text{card}(F[n]) \times \ell_{\text{IR}}$ dimensional vector comprising all $(X_i[n])_{i\in F[n]}$ and the estimated impulse response defined as $\hat{H}_{F[n],S^n}^{|n}$, is a $\text{card}(F[n]) \times \ell_{\text{IR}}$ dimensional vector comprising all $(\hat{H}_{i,S^{n-1}}^{|n-1})_{i\in F[n]}$. The conventional NLMS update formula turns to:

$$\hat{H}_{F[n],S^n}^{|n} = \hat{H}_{F[n-1],S^{n-1}}^{|n-1} + \frac{\mu[n]}{X_{F[N]}^\top[n]X_{F[N]}[n]+\delta}X_{F[N]}[n]e[n] \tag{9.34}$$

Moreover, the priori error and the the expectations of posteriori misalignment ℓ_2 norm turn to:

$$\begin{aligned}e[n] &= y[n] - X_{F[N]}^\top[n]\hat{H}_{F[n-1],S^{n-1}}^{|n-1}\\ \mathsf{E}(\|z_{F[n]}[n]\|_2^2) &= \sum_{i\in F[n]}\mathsf{E}(\|z_i[n]\|_2^2)\end{aligned} \tag{9.35}$$

The procedure of derivation is almost the same as the subsection 9.3.2, except the formula $\mathsf{E}(X_i[n]\,X_i^\top[n]) = \sigma_{X_i}^2 I_{\ell_{\text{IR}}}$. In the case of superposition, we have $\mathsf{E}(X_{F[n]}[n]\;X_{F[n]}^\top[n]) \approx \sigma_{X_{F[n]}}^2 I_{\ell_{\text{IR}}}$. As presented in HMM

model (Section 3.5), $X_{F[n]}[n]$ is a vector composed of 0 and 1, and there is at most card$(F[n])$ components equalling to 1. Similar to the matrix presented in (9.17), $X_{F[n]}[n]\ X_{F[n]}^{\top}[n]$ is a dominant matrix, thus can be approximated by a diagonal matrix.

Finally, the recursive optimization formula of the third case is almost the same as the formula (9.33), only replace the notation i by $F[n]$.

$$\begin{aligned}
e[n] =& y[n] - X_{F[n]}^{\top}[n]\hat{H}_{F[n-1],S^{n-1}}^{|n-1} \\
\hat{H}_{F[n],S^{n}}^{|n} =& \hat{H}_{F[n-1],S^{n-1}}^{|n-1} + \frac{\mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{W_{H_{F[n]}}}^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_{F[n]}}^2(\sigma_{W_{H_{F[n]}}}^2 + \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2))}\ X_{F[n]}[n]e[n] \\
\mathsf{E}(\|z_{F[n]}[n]\|_2^2) =& \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell_{\text{IR}}\sigma_{W_{H_{F[n]}}}^2 \\
& - \frac{\sigma_{X_{F[n]}}^2(\ell_{\text{IR}}\sigma_{W_{H_{F[n]}}}^2 + \mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2)^2}{\ell_{\text{IR}}\hat{V}^{|n} + \ell_{\text{IR}}\sigma_{X_{F[n]}}^2\mathsf{E}(\|z_{F[n-1]}[n-1]\|_2^2) + \ell_{\text{IR}}^2\sigma_{X_{F[n]}}^2\sigma_{W_{H_{F[n]}}}^2}
\end{aligned} \tag{9.36}$$

In a general manner [114], the variance $\sigma^2[n]$, such as: $\sigma_{X_i}^2[n]$, $\sigma_{W_{H_i}}^2[n]$ and $\hat{V}^{|n}[n]$, denotes the power of sequence, and can be update by the smooth factor:

$$\sigma^2[n] = \lambda_\sigma\sigma^2[n-1] + (1-\lambda_\sigma)\sigma^2[n] \tag{9.37}$$

where $\lambda_\sigma = 1 - \frac{1}{K\ell_{\text{IR}}}$, with $K > 1$.

Bibliography

[1] E Matheson, Y Aoustin, E Le Carpentier, A Leon, and J Perrin. Anthropomorphic underactuated hand with 15 joints. In *New Trends in Medical and Service Robots*, pages 277–295. Springer, 2016. 9, 14

[2] Per Brodal. *The central nervous system: structure and function*. Oxford University Press, 2004. 9, 18

[3] HS Milner-Brown, RB Stein, and R_ Yemm. The orderly recruitment of human motor units during voluntary isometric contractions. *The Journal of physiology*, 230(2):359–370, 1973. 18

[4] RE Burke, DN Levine, P Tsairis, and FE 3rd Zajac. Physiological types and histochemical profiles in motor units of the cat gastrocnemius. *The Journal of physiology*, 234(3):723–748, 1973. 18

[5] RP Betts, DM Johnston, and BH Brown. Nerve fibre velocity and refractory period distributions in nerve trunks. *J. of Neurology, Neurosurgery & Psychiatry*, 39(7):694–700, 1976. 19, 40

[6] J Kimura, T Yamada, and RL Rodnitzky. Refractory period of human motor nerve fibres. *J. of Neurology, Neurosurgery & Psychiatry*, 41(9):784–790, 1978. 19, 40

[7] David Hampel and Petr Lansky. On the estimation of refractory period. *J. of Neuroscience Methods*, 171(2):288 – 295, 2008. 19

[8] D Farina, R Merletti, and DF Stegeman. Biophysics of the generation of emg signals. *Electromyography: physiology, engineering, and noninvasive applications*, pages 81–105, 2004. 20

[9] RV Routal and GP Pal. A study of motoneuron groups and motor columns of the human spinal cord. *The Journal of Anatomy*, 195(2):211–224, 1999. 20

[10] BE Tomlinson and Dorothy Irving. The numbers of limb motor neurons in the human lumbosacral cord throughout life. *Journal of the neurological sciences*, 34(2):213–219, 1977. 20

[11] Jeremy S Dasen, Jeh-Ping Liu, and Thomas M Jessell. Motor neuron columnar fate imposed by sequential phases of hox-c activity. *Nature*, 425(6961):926, 2003. 20

[12] Thomas ME Brushart. Preferential motor reinnervation: a sequential double-labeling study. *Restorative neurology and neuroscience*, 1(3, 4):281–287, 1990. 20

[13] Alan J McComas. Motor unit estimation: anxieties and achievements. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine*, 18(4):369–379, 1995. 20

[14] Charles E Blevins. Innervation patterns of the human stapedius muscle. *Archives of Otolaryngology*, 86(2):136–142, 1967. 20

[15] MICHELE Torre. Nombre et dimensions des unités motrices dans les muscles extrinsèques de l'œil et, en général, dans les muscles squélettiques reliés à des organes de sens. *Schweiz Arch Neurol Psychiatr*, 72(1-2):362–376, 1953. 20

[16] KENRO Kanda and K Hashizume. Factors causing difference in force output among motor units in the rat medial gastrocnemius muscle. *The Journal of physiology*, 448(1):677–695, 1992. 20

[17] Miriam Young, Angelika Paul, Judith Rodda, Marilyn Duxson, and Philip Sheard. Examination of intrafascicular muscle fiber terminations: Implications for tension delivery in series-fibered muscles. *Journal of morphology*, 245(2):130–145, 2000. 20

[18] WA Weijs, PJW Jüch, SHS Kwa, and JAM Korfage. Motor unit territories and fiber types in rabbit masseter muscle. *Journal of dental research*, 72(11):1491–1498, 1993. 20

[19] A John Harris, Marilyn J Duxson, Jane E Butler, Paul W Hodges, Janet L Taylor, and Simon C Gandevia. Muscle fiber and motor unit behavior in the longest human skeletal muscle. *Journal of Neuroscience*, 25(37):8528–8533, 2005. 20

[20] Taian MM Vieira, Ian D Loram, Silvia Muceli, Roberto Merletti, and Dario Farina. Postural activation of the human medial gastrocnemius muscle: are the muscle units spatially localised? *The Journal of physiology*, 589(2):431–443, 2011. 20

[21] STEPHAN Riek and PARVEEN Bawa. Recruitment of motor units in human forearm extensors. *Journal of Neurophysiology*, 68(1):100–108, 1992. 20

[22] PA McNulty, KJ Falland, and VG Macefield. Comparison of contractile properties of single motor units in human intrinsic and extrinsic finger muscles. *The Journal of physiology*, 526(2):445–456, 2000. 20

[23] CS Klein, CK Häger-Ross, and CK Thomas. Fatigue properties of human thenar motor units paralysed by chronic spinal cord injury. *The Journal of physiology*, 573(1):161–171, 2006. 20

[24] Nikolic Miki. *Detailed analysis of clinical electromyography signals: EMG decomposition, findings and firing pattern analysis in controls and patients with myopathy and amytrophic lateral sclerosis*. PhD thesis, University of Copenhagen, 2001. 20

[25] Timothy J. Doherty and Daniel W. Stashuk. Decomposition-based quantitative electromyography: methods and initial normative data in five muscles. *Muscle Nerve*, 28:204–211, 2003. 20

[26] T. Kamali, R. Boostani, and H. Parsaei. A multi–classifier approach to MUAP classification for diagnosis of neuromuscular disorders. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 22:191–200, 2014. 20

[27] Zoia C. Lateva, Kevin C. McGill, and M. Elise Johanson. The innervation and organization of motor units in a series-fibered human muscle: the brachioradialis. *European J. of Applied Physiology*, 108:1530–1541, 2010. 20

[28] C. J. De Luca and Z. Erim. Common drive of motor units in regulation of muscle force. *Trends in Neurosciences*, 17:299–305, 1994. 20

[29] A. Adam and C. J. De Luca. Recruitment order of motor units in human vastus lateralis muscle is maintained during fatiguing contractions. *Trends in Neurosciences*, 90:2919–2927, 2003. 20

[30] D. Farina and A. Holobar. Human machine interfacing by decoding the surface electromyogram. *IEEE Signal Processing Magazine*, 32:115–120, 2015. 20

[31] Francesco Negro, Silvia Muceli, Margherita Castronovo, and Dario Farina. Multi-channel intramuscular and surface EMG decomposition by convolutive blind source separation. *J. of Neural Engineering*, 13(2), 2016. 20, 24, 28, 34

[32] Dario Farina, Ivan Vujaklija, and Massimo Sartori. Man/machine interface based on the discharge timings of spinal motor neurons after targeted muscle reinnervation. *Nature biomedical engineering*, 1, 2017. 20

[33] B. Mambrito and C. De Luca. A Technique for the Detection, Decomposition and Analysis of the EMG Signal. *Electroencephalography and Clinical Neurophysiology*, 58:175–188, 1984. 20

[34] C.J. C.J. Heckman and R.M. Enoka. Motor unit. *Compr Physiol*, 2(4):2629–2682, 2012. 9, 21

[35] Ales Holobar and Damjan Zazula. Multichannel blind source separation using convolution kernel compensation. *IEEE Transactions on Signal Processing*, 55(9):4487–4496, 2007. 22, 28, 29, 30, 34

[36] Aleš Holobar and Damjan Zazula. Gradient convolution kernel compensation applied to surface electromyograms. In *International Conference on Independent Component Analysis and Signal Separation*, pages 617–624. Springer, 2007. 22, 30

[37] Aleš Holobar, Dario Farina, Marco Gazzoni, Roberto Merletti, and Damjan Zazula. Estimating motor unit discharge patterns from high-density surface electromyogram. *Clinical Neurophysiology*, 120(3):551–562, 2009. 22, 30

[38] A Holobar, V Glaser, JA Gallego, Jakob Lund Dideriksen, and D Farina. Non-invasive characterization of motor unit behaviour in pathological tremor. *Journal of neural engineering*, 9(5):056011, 2012. 22, 30

[39] A Holobar, MA Minetto, and D Farina. Accurate identification of motor unit discharge patterns from high-density surface emg and validation with a novel signal-based performance metric. *Journal of neural engineering*, 11(1):016008, 2014. 22, 30

[40] Roberto Merletti and Dario Farina. Analysis of intramuscular electromyogram signals. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1887):357–368, 2009. 9, 22

[41] R. Merletti and D. Farina. Analysis of intramuscular electromyogram signals. *Philosophical Trans. - Royal Society. Biological sciences*, 367:357–368, 2009. 22

[42] Edgar D Adrian and Detlev W Bronk. The discharge of impulses in motor nerve fibres. *The Journal of physiology*, 67(2):9–151, 1929. 22

[43] John V Basmajian. Muscles alive. their functions revealed by electromyography. *Academic Medicine*, 37(8):802, 1962. 23

[44] Mark S Malagodi, Kenneth W Horch, and Andrew A Schoenberg. An intrafascicular electrode for recording of action potentials in peripheral nerves. *Annals of biomedical engineering*, 17(4):397–410, 1989. 23

[45] Dario Farina, Ken Yoshida, Thomas Stieglitz, and Klaus Peter Koch. Multichannel thin-film electrode for intramuscular electromyographic recordings. *Journal of Applied Physiology*, 104(3):821–827, 2008. 23

[46] Ronald S LeFever and Carlo J De Luca. A procedure for decomposing the myoelectric signal into its constituent action potentials-part i: technique, theory, and implementation. *IEEE transactions on biomedical engineering*, (3):149–157, 1982. 9, 24, 25, 26, 27, 28

[47] Ronald S Lefever, Alan P Xenakis, and Carlo J De Luca. A procedure for decomposing the myoelectric signal into its constituent action potentials-part ii: execution and test for accuracy. *IEEE transactions on biomedical engineering*, (3):158–164, 1982. 9, 24, 25, 26, 27, 28

[48] V. J. Prochazka, B. Conrad, and F. Sindermann. A neuroelectric signal recognition system. *Electroencephalography and Clinical Neurophysiology*, 32(1):95–97, 1972. 23

[49] V. J. Prochazka and H. H. Kornhuber. On-line multi-unit sorting with resolution of superposition potentials. *Electroencephalography and Clinical Neurophysiology*, 34(1):91–93, 1973. 23

[50] J. F. Vibert and J. Costa. Spike separation in multiunit records: A multivariate analysis of spike descriptive parameters. *Electroencephalography and Clinical Neurophysiology*, 47(2):172–182, 1979. 23

[51] Julien Roussel, Philippe Ravier, and Michel Haritopoulos. Decomposition of Multi-Channel Intramuscular EMG Signals by Cyclostationary-Based Blind Source Separation. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 25(11):2035–2045, 2017. 24, 28, 34

[52] Hossein Parsaei, Daniel W Stashuk, Sarbast Rasheed, Charles Farkas, and Andrew Hamilton-Wright. Intramuscular emg signal decomposition. *Critical Reviews™ in Biomedical Engineering*, 38(5), 2010. 24

[53] Kevin C. Mcgill, Kenneth L. Cummins, and Leslie J. Dorfman. Automatic decomposition of the clinical electromyogram. *IEEE Trans. on biomedical engineering*, 32(7), 1985. 9, 25, 26, 27, 52, 56, 67, 68

[54] Jianjun Fang, Gyan C Agarwal, and Bhagwan T Shahani. Decomposition of multiunit electromyographic signals. *IEEE transactions on biomedical engineering*, 46(6):685–697, 1999. 25

[55] Xiaomei Ren, Xiao Hu, Zhizhong Wang, and Zhiguo Yan. Muap extraction and classification based on wavelet transform and ica for emg decomposition. *Medical and Biological Engineering and Computing*, 44(5):371, 2006. 25, 26

[56] Adriano O Andrade, Slawomir Nasuto, Peter Kyberd, Catherine M Sweeney-Reed, and FR Van Kanijn. Emg signal filtering based on empirical mode decomposition. *Biomedical Signal Processing and Control*, 1(1):44–55, 2006. 25

[57] J.R. Florestal, Pierre A. Mathieu, and Armando Malanda. Automated Decomposition of Intramuscular Electromyographic Signals. *IEEE Trans. on Biomedical Engineering*, 53(5):832–839, 2006. 25, 34, 52

[58] Christos D Katsis, Yorgos Goletsis, Aristidis Likas, Dimitrios I Fotiadis, and Ioannis Sarmas. A novel method for automated emg decomposition and muap classification. *Artificial Intelligence in Medicine*, 37(1):55–64, 2006. 25, 27

[59] Christos D Katsis, Themis P Exarchos, Costas Papaloukas, Yorgos Goletsis, Dimitrios I Fotiadis, and Ioannis Sarmas. A two-stage method for muap classification based on emg decomposition. *Computers in Biology and Medicine*, 37(9):1232–1240, 2007. 25

[60] Z. Erim and Winsean Lin. Decomposition of Intramuscular EMG Signals Using a Heuristic Fuzzy Expert System. *IEEE Transactions on Biomedical Engineering*, 55(9):2180–2189, September 2008. 25

[61] Sarbast Rasheed, Daniel Stashuk, and Mohamed Kamel. Adaptive fuzzy k-nn classifier for emg signal decomposition. *Medical engineering & physics*, 28(7):694–709, 2006. 25, 26, 27

[62] Sarbast Rasheed, Daniel W Stashuk, and Mohamed S Kamel. A hybrid classifier fusion approach for motor unit potential classification during emg signal decomposition. *IEEE Transactions on Biomedical Engineering*, 54(9):1715–1721, 2007. 25, 26

[63] Dennis Tkach, He Huang, and Todd A Kuiken. Study of stability of time-domain features for electromyographic pattern recognition. *Journal of neuroengineering and rehabilitation*, 7(1):21, 2010. 26

[64] Mohammadreza Asghari Oskoei, Huosheng Hu, et al. Support vector machine-based classification scheme for myoelectric control applied to upper limb. *IEEE Trans. Biomed. Engineering*, 55(8):1956–1965, 2008. 26

[65] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for emg signal classification. *Expert Systems with Applications*, 39(8):7420–7431, 2012. 26

[66] Xiaomei Ren, Zhiguo Yan, Zhizhong Wang, and Xiao Hu. Noise reduction based on ica decomposition and wavelet transform for the extraction of motor unit action potentials. *Journal of neuroscience methods*, 158(2):313–322, 2006. 26

[67] Richard Gut and George S Moschytz. High-precision emg signal decomposition using communication techniques. *IEEE transactions on signal processing*, 48(9):2487–2494, 2000. 26, 52

[68] S Hamid Nawab. Integrated processing and understanding signals. *Symbolic and knowledge-based signal processing*, pages 251–285, 1992. 27

[69] S Hamid Nawab, R Wotiz, and CJ De Luca. Improved resolution of pulse superpositions in a knowledge-based system emg decomposition. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 1, pages 69–71. IEEE, 2004. 27

[70] SH Nawab, R Wotiz, and CJ De Luca. Resolving emg pulse superpositions via utility maximization. *Proc 8th World Multiconf Systemics Cybernetics Informatics, Orlando, FL*, pages 233–236, 2004. 27

[71] Carlo J De Luca, Alexander Adam, Robert Wotiz, L Donald Gilmore, and S Hamid Nawab. Decomposition of surface emg signals. *Journal of neurophysiology*, 96(3):1646–1657, 2006. 27, 29, 31

[72] Sarbast Rasheed, Daniel Stashuk, and Mohamed Kamel. An interactive environment for motor unit potential classification using certainty-based classifiers. *Simulation Modelling Practice and Theory*, 16(9):1293–1311, 2008. 27

[73] Sarbast Rasheed. *A multiclassifier approach to motor unit potential classification for EMG signal decomposition*. PhD thesis, University of Waterloo, 2006. 27

[74] Tahereh Kamali, Reza Boostani, and Hossein Parsaei. A multi-classifier approach to muap classification for diagnosis of neuromuscular disorders. *IEEE transactions on neural systems and rehabilitation engineering*, 22(1):191–200, 2014. 27

[75] K. McGill, Z. Lateva, and H. Marateb. EMGLAB: An interactive EMG decomposition program. *J. of Neuroscience Methods*, 149(2):121–133, 2005. 27, 70

[76] Marateb H.R., Muceli S., Mcgill K.C., Merletti R., and Farina D. Robust decomposition of single-channel intramuscular EMG signals at low force levels. *IEEE Trans. on Automatic Control*, 8(6):1–13, 2011. 27

[77] Joël R Florestal, Pierre A Mathieu, and Réjean Plamondon. A genetic algorithm for the resolution of superimposed motor unit action potentials. *IEEE Transactions on Biomedical Engineering*, 54(12):2163–2171, 2007. 27

[78] Hamid Reza Marateb and Kevin C McGill. Resolving superimposed muaps using particle swarm optimization. *IEEE Transactions on Biomedical Engineering*, 56(3):916–919, 2009. 27

[79] K.C. McGill, Z.C. Lateva, and H.R. Marateb. EMGLAB: An interactive EMG decomposition program. *J. of Neuroscience Methods*, 149(2):121–133, December 2005. 27, 28

[80] Daniel W Stashuk. Mean, median and mode estimation of motor unit action potential templates. In *Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the IEEE*, volume 4, pages 1498–1499. IEEE, 1996. 27

[81] D. Farina, R. Colombo, R. Merletti, and H. Baare Olsen. Evaluation of intra-muscular EMG signal decomposition algorithms. *J. of Electromyography and Kinesiology*, 11:175–187, 2001. 28, 70

[82] Johan Thomas, Yannick Deville, and Shahram Hosseini. Time-domain fast fixed-point algorithms for convolutive ica. *IEEE Signal Processing Letters*, 13(4):228–231, 2006. 28

[83] Saeed Karimimehr, Hamid Reza Marateb, Silvia Muceli, Marjan Mansourian, Miguel Angel Mananas, and Dario Farina. A Real-Time Method for Decoding the Neural Drive to Muscles Using Single-Channel Intra-Muscular EMG Recordings. *Int. J. of Neural Systems*, 27(6):1750025, 2017. 28, 34

[84] Jonathan Monsifrot, Eric Le Carpentier, Yannick Aoustin, and Dario Farina. Sequential Decoding of Intramuscular EMG Signals via Estimation of a Markov Model. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 22(5):1030–40, 2014. 28, 35, 36, 39, 85, 87

[85] Monsifrot Jonathan. *modélisation de signaux électromyographiques par des processus de renouvellement, Filtre bayésien pour l'estimation séquentielle de paramètres à destination de la commande d'une prothése d'avant-bras*. PhD thesis, Ecole Centrale Nantes, 2013. 28, 35, 36, 39, 40, 41, 85

[86] A Holobar and D Farina. Blind source identification from the multichannel surface electromyogram. *Physiological measurement*, 35(7):R143, 2014. 9, 29

[87] Leonardo Gizzi, Jørgen Feldbæk Nielsen, Francesco Felici, Juan C Moreno, José L Pons, and Dario Farina. Motor modules in robot-aided walking. *Journal of neuroengineering and rehabilitation*, 9(1):76, 2012. 29

[88] Silvia Muceli, Andreas Trøllund Boye, Andrea d'Avella, and Dario Farina. Identifying representative synergy matrices for describing muscular activation patterns during multidirectional reaching in the horizontal plane. *Journal of neurophysiology*, 103(3):1532–1542, 2010. 29

[89] Dario Farina, Cédric Févotte, Christian Doncarli, and Roberto Merletti. Blind separation of linear instantaneous mixtures of nonstationary surface myoelectric signals. *IEEE Transactions on Biomedical Engineering*, 51(9):1555–1567, 2004. 29

[90] Dario Farina, Marie-Françoise Lucas, and Christian Doncarli. Optimized wavelets for blind separation of nonstationary surface myoelectric signals. *IEEE Transactions on Biomedical Engineering*, 55(1):78–86, 2008. 29

[91] S Hamid Nawab, Shey-Sheen Chang, and Carlo J De Luca. High-yield decomposition of surface emg signals. *Clinical neurophysiology*, 121(10):1602–1615, 2010. 29, 31

[92] Vojko Glaser, Ales Holobar, and Damjan Zazula. Real-Time Motor Unit Identification From High-Density Surface EMG. *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, 21(6):949–958, 2013. 29, 31, 34

[93] Yong Ning, Xiangjun Zhu, Shanan Zhu, and Yingchun Zhang. Surface emg decomposition based on k-means clustering and convolution kernel compensation. *IEEE journal of biomedical and health informatics*, 19(2):471–477, 2015. 30

[94] Ivan Gligorijević, Maarten De Vos, Joleen H Blok, Bogdan Mijović, Johannes P van Dijk, and Sabine Van Huffel. Automated way to obtain motor units' signatures and estimate their firing patterns during voluntary contractions using hd-semg. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4090–4093. IEEE, 2011. 31

[95] Ivan Gligorijević, Johannes P van Dijk, Bogdan Mijović, Sabine Van Huffel, Joleen H Blok, and Maarten De Vos. A new and fast approach towards semg decomposition. *Medical & biological engineering & computing*, 51(5):593–605, 2013. 31

[96] André R Brodtkorb, Trond R Hagen, and Martin L Sætra. Graphics processing unit (gpu) programming strategies and trends in gpu computing. *Journal of Parallel and Distributed Computing*, 73(1):4–13, 2013. 10, 32, 33, 58, 59

[97] Shane Cook. *CUDA programming: a developer's guide to parallel computing with GPUs*. Newnes, 2012. 33, 58, 59

[98] John Nickolls and William J Dally. The gpu computing era. *IEEE micro*, 30(2), 2010. 33

[99] Jason Sanders and Edward Kandrot. *CUDA by example: an introduction to general-purpose GPU programming*. Addison-Wesley Professional, 2010. 33, 34, 59

[100] David Luebke and M Harris. General-purpose computation on graphics hardware. In *Workshop, SIGGRAPH*, 2004. 33

[101] NVIDIA Corporation. Whitepaper nvidia's next generation cuda compute architecture: Fermi. `https://www.nvidia.com/content/PDF/fermi_white_papers/NVIDIA_Fermi_Compute_Architecture_Whitepaper.pdf`, 2009. 34, 59

[102] NVIDIA Corporation. Whitepaper nvidia's next generation cuda compute architecture: Kepler gk110. `https://www.nvidia.com/content/PDF/kepler/NVIDIA-kepler-GK110-architecture-whitepaper.pdf`, 2012. 34

[103] Konstantin Akhmadeev, Elena Rampone, Tianyi Yu, Yannick Aoustin, and Eric Le Carpentier. A testing system for a real-time gesture classification using surface emg. *IFAC-PapersOnLine*, 50(1):11498–11503, 2017. 34

[104] D. Stashuk and H. DeBruin. Automatic decomposition of selective needle-detected myoelectric signals. *IEEE Trans. on Biomedical Engineering*, 35:1–10, 1988. 34

[105] J.R. Florestal, P.A. Mathieu, and K.C. McGill. Automatic decomposition of multichannel intramuscular EMG signals. *J. of Electromyography and Kinesiology*, 19:1–9, 2009. 34

[106] H. R. Marateb, K. C. McGill, and A Holobar. Robust decomposition of single-channel intramuscular EMG signals at low force levels. *J. of Neural Engineering*, 8(6):066015, 2011. 34

[107] D. Farina, A. Crosetti, and R. Merletti. A model for the generation of synthetic intramuscular EMG signals to test decomposition algorithms. *IEEE Trans. on Biomedical Engineering*, 48(1):66–77, January 2001. 35

[108] Dan Stashuk. EMG signal decomposition: how can it be accomplished and used? *J. of Electromyography and Kinesiology*, 11(3):151–173, 2001. 35

[109] Vlad Barbu and Nikolaos Limnios. Reliability theory for discrete-time semi-Markov systems. In *Semi-Markov Chains and Hidden Semi-Markov Models toward Applications*, volume 191 of *Lecture Notes in Statistics*, pages 1–30. Springer New York, 2008. 39

[110] T. Nakagawa and S. Osaki. The Discrete Weibull Distribution. *IEEE Trans. on Reliability*, R-24(5):300–301, December 1975. 40

[111] T. Schon, F. Gustafsson, and P.-J. Nordlund. Marginalized particle filters for mixed linear nonlinear state-space models. *IEEE Trans. on Signal Processing*, 53:2279–2289, 2005. 43

[112] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960. 46

[113] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2008. 48, 90

[114] Constantin Paleologu, Silviu Ciochina, and Jacob Benesty. Variable step-size nlms algorithm for under-modeling acoustic echo cancellation. *IEEE Signal Processing Letters*, 15:5–8, 2008. 49, 93

[115] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification*. The MIT Press, Massachusetts and London, 1983. 51

[116] C.D. Katsis, Y. Goletsis, A. Likas, D.I. Fotiadis, and I. Sarmas. A novel method for automated EMG decomposition and MUAP classification. *Artificial Intelligence in Medicine*, 37:55–64, 2006. 52

[117] Yao Zhang and John D. Owens. A quantitative performance analysis model for gpu architectures. In *2011 IEEE 17th International Symposium on High Performance Computer Architecture*, pages 382–393, February 2001. 57

[118] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009. 60

[119] Ananth Grama, Vipin Kumar, Anshul Gupta, and George Karypis. *Introduction to parallel computing*. Pearson Education, 2003. 10, 61, 62

[120] Bilal Jan, Bartolomeo Montrucchio, Carlo Ragusa, Fiaz Gul Khan, and Omar Khan. Fast parallel sorting algorithms on gpus. *International Journal of Distributed and Parallel Systems*, 3(6):107, 2012. 62

[121] Alexander Greb and Gabriel Zachmann. Gpu-abisort: optimal parallel sorting on stream architectures. In *Proceedings 20th IEEE International Parallel and Distributed Processing Symposium*, 2006. 63

[122] Nadathur Satish, Mark Harris, and Michael Garland. Designing efficient sorting algorithms for manycore gpus. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–10. IEEE, 2009. 63

[123] Keliang Zhang and Baifeng Wu. A novel parallel approach of radix sort with bucket partition preprocess. In *2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems*, pages 989–994. IEEE, 2012. 63

[124] Xiaochun Ye, Dongrui Fan, Wei Lin, Nan Yuan, and Paolo Ienne. High performance comparison-based sorting algorithm on many-core gpus. In *Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on*, pages 1–10. IEEE, 2010. 63

[125] Sam White, Niels Verosky, and Tia Newhall. A cuda-mpi hybrid bitonic sorting algorithm for gpu clusters. In *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on*, pages 588–589. IEEE, 2012. 63

[126] P. Rosenfalck. Intra- and extracellular potential fields of active nerve and muscle fibers. *Acta Physiol. Scand.*, 61:1–96, 1964. 67

[127] S. Andreassen and A. Rosenfalck. Recording from a single motor unit during strong effort. *IEEE Trans. on biomedical engineering*, 25:501–508, 1978. 67

[128] S. Usui and I. Amidror. Digital low-pass differentiation for biological signal processing. *IEEE Trans. on biomedical engineering*, 29:686–693, 1982. 67

[129] Mohammadreza Asghari Oskoei and Huosheng Hu. Myoelectric control systems—a survey. *Biomedical Signal Processing and Control*, 2(4):275–294, 2007. 82

[130] K. Kulasekera. Approximate mle's of the parameters of a discrete weibull distribution with type-i censored data. *Microelectronics Reliability*, 34(7):1185–1188, 1994. 87

[131] S. Yi, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic search using the natural gradient. In *In Proc. of the 26th Ann. Int. Conf on Machine Learning*, pages 1161–1168, 2009. 87

[132] Silviu Ciochină, Constantin Paleologu, and Jacob Benesty. An optimized nlms algorithm for system identification. *Signal Processing*, 118:115–121, 2016. 90

[133] Jacob Benesty, Hernan Rey, Leonardo Rey Vega, and Sara Tressens. A nonparametric vss nlms algorithm. *IEEE Signal Processing Letters*, 13(10):581–584, 2006. 91

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

CENTRALE NANTES

Titre : Décomposition en temps réel de signaux iEMG: filtrage bayésien implémenté sur GPU

Mots clés : Markov modèle caché, filtrage bayésien, calcul parallèle , décomposition en temps réel

Résumé : Un algorithme de décomposition des unités motrices constituant un signal électro-myographiques intramusculaires (iEMG) a été proposé au laboratoire LS2N. Il s'agit d'un filtrage bayésien estimant l'état d'un modèle de Markov caché. Cet algorithme demande beaucoup de temps d'execution, même pour un signal ne contenant que 4 unités motrices.
Dans notre travail, nous avons d'abord validé cet algorithme dans une structure série. Nous avons proposé quelques modifications pour le modèle de recrutement des unités motrices et implémenté deux techniques de pré-traitement pour améliorer la performance de l'algorithme. Le banc de filtres de Kalman a été remplacé par un banc de filtre LMS. Le filtre global consiste en l'examen de divers scénarios arborescents d'activation des unités motrices: on a introduit deux techniques heuristiques pour élaguer les divers scénarios. On a réalisé l'implémentation GPU de cet algorithme à structure parallèle intrinsèque
On a réussi la décomposition de 10 signaux expérimentaux enregistrés sur deux muscules, respectivement avec électrode aiguille et électrode filaire. Le nombre d'unités motrices est de 2 à 8. Le pourcentage de superposition des potentiels d'unité motrice, qui représente la complexité de signal, varie de 6.56 % à 28.84 %. La précision de décomposition de tous les signaux sont plus que 90 %, sauf deux signaux en 30 % MVC , sauf pour deux signaux qui sont à 30 % MVC et dont la précision de décomposition est supérieure à 85%. Nous sommes les premiers à réaliser la décomposition en temps réel pour un signal constitué de 10 unités motrices.

Title : On-line decomposition of iEMG signals using GPU-implemented Bayesian filtering

Keywords : Hidden Markov models, Bayesian methods, Recursive estimation, Electromyography decomposition, parallel computation, real-time decomposition.

Abstract :A sequential decomposition algorithm based on a Hidden Markov Model of the EMG, that used Bayesian filtering to estimate the unknown parameters of discharge series of motor units was previously proposed in the laboratory LS2N. This algorithm has successfully decomposed the experimental iEMG signal with four motor units. However, the proposed algorithm demands a high time consuming.
In this work, we firstly validated the proposed algorithm in a serial structure. We proposed some modifications for the activation process of the recruitment model in Hidden Markov Model and implemented two signal pre-processing techniques to improve the performance of the algorithm. Then, we realized a GPU-oriented implementation of this algorithm, as well as the modifications applied to the original model in order to achieve a real-time performance.
We have achieved the decomposition of 10 experimental iEMG signals acquired from two different muscles, respectively by fine wire electrodes and needle electrodes. The number of motor units ranges from 2 to 8. The percentage of superposition, representing the complexity of iEMG signal, ranges from 6.56 % to 28.84 %. The accuracies of almost all experimental iEMG signals are more than 90 %, except two signals at 30 % MVC (more than 85 %). Moreover, we realized the real-time decomposition for all these experimental signals by the parallel implementation. We are the first one that realizes the real time full decomposition of single channel iEMG signal with number of MUs up to 10, where full decomposition means resolving the superposition problem. For the signals with more than 10 MUs, we can also decompose them quickly, but not reaching the real time level.