

TRABALHO DE GRADUAÇÃO

**DECODING AND ENCODING OF NEURAL SIGNALS
FOR PERIPHERAL INTERFACES**

Por

Lucas de Levy Oliveira

Brasília, dezembro de 2015



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

**DECODING AND ENCODING OF NEURAL SIGNALS
FOR PERIPHERAL INTERFACES**

Por
Lucas de Levy Oliveira

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Antônio P. L. Bó, ENE/UnB _____
Orientador

Profa. Mariana C. B. Matias, FGA/UnB _____
Co-orientadora

Profa. Christine A. Coste, INRIA Sophia-Antipolis _____
Examinadora externa

Brasília, dezembro de 2015

FICHA CATALOGRÁFICA

OLIVEIRA, LUCAS DE LEVY

Decoding and encoding of neural signals for peripheral interfaces,

[Distrito Federal] 2015.

viii, 68p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2015). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Neuroengenharia

2. Engenharia Biomédica

3. Estimulação neural

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA, L. de L., (2015). Decoding and encoding of neural signals for peripheral interfaces. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº020, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 68.

CESSÃO DE DIREITOS

AUTOR: Lucas de Levy Oliveira

TÍTULO DO TRABALHO DE GRADUAÇÃO: Decoding and encoding of neural signals for peripheral interfaces.

GRAU: Engenheiro

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Lucas de Levy Oliveira

SMPW Quadra 25 Conjunto 03 Lote 7 Casa B - Park Way.

71745-503 Brasília – DF – Brasil.

Dedicatória

Dedico este trabalho ao meu amado tio Gerson Oliveira Jr., que, apesar de todo seu apoio e carinho, infelizmente não pôde acompanhar o término deste projeto.

Dedico, ainda, às amizades que conquistei ao longo dos últimos anos dentro e fora da Universidade, especialmente a minha irmã Ana Clara da Hora, por todo apoio e bons momentos que compartilhamos. Dedico a Anna Carolina Pinheiro, grande amiga e secretária do Departamento de Engenharia Elétrica, pelos conselhos, pelo suporte e, especialmente, pelos momentos de diversão, descontração e alegria abundantes. Dedico, enfim, ao professor Antônio Bó, pela grande amizade construída, por toda ajuda e paciência concedida e pelo grande aprendizado por ele proporcionado nos últimos anos.

Acima de tudo, dedico este trabalho aos meus pais, Rogério de Oliveira e Mônica Machado; aos meus avós, Levy Machado, Maria Aparecida, Gerson Oliveira e Maria Amélia; ao meu irmão, Victor de Levy; aos meus tios, tias, primos e primas; e ao meu amor, Marina Moreira, por todo carinho, amor e suporte emocional recebidos por todos, que foram os mais fundamentais para a realização deste trabalho.

Lucas de Levy Oliveira

Agradecimentos / Acknowledgements

Agradeço, inicialmente, a todos da minha família. Sem vocês, jamais teria tido condições de crescer academicamente e emocionalmente como cresci ao longo dos anos que precederam a realização deste trabalho. Agradeço aos meus pais pelos conselhos frequentes e por sempre confiarem em minhas decisões. Agradeço aos meus avós, tios e primos pelo apoio emocional e pelo orgulho que me fazem sentir pelo que faço. Agradeço ao meu irmão pelas boas risadas, pela preocupação e por sempre estar presente quando preciso. Agradeço, por fim, a minha companheira de aventuras, por sempre me ajudar a ser a melhor versão de mim mesmo e nunca me permitir duvidar de meu potencial.

Agradeço aos meus grandes amigos e colegas de curso e de laboratório por toda ajuda em projetos e em disciplinas, além da forte amizade construída – em especial a Ana Clara da Hora, Anna Carolina Pinheiro, André Vinícius, Bruno Noronha, De Hong Jung, Guilherme Anselmo, Lucas Fonseca, Marcela Carvalho, Matheus Portela, Miguel Paredes (que, inclusive, me ajudou com os procedimentos cirúrgicos das baratas) e Tiago Pimentel. Agradeço à Marienne Narváez e ao professor Emerson Fachin-Martins por toda a ajuda nos experimentos desenvolvidos no LARA. Agradeço, ainda, a Murilo Marinho pela amizade, pela ajuda em diversos aspectos e, inclusive, no desenvolvimento do código para o projeto desenvolvido no LARA.

Agradeço, também, ao corpo administrativo da Universidade de Brasília, por todo o suporte técnico concedido durante a graduação e pelo companheirismo em diversos momentos de dificuldade do curso, em especial aos atendentes da secretaria da engenharia elétrica Lúcio e Natália; porteiros dos Serviços Gerais 11 Elisângela e Sidney; e responsável pela limpeza da Faculdade de Tecnologia Maria José.

Agradeço ao corpo docente da Universidade de Brasília pelas lições de vida compartilhadas e conhecimentos técnicos adquiridos, em especial a Antônio Bó.

I am also grateful for Backyard Brains', NDI Digital's and Ripple's technical support for all the help during the development of this work. I would like to thank for the friendships developed during my time in Johns Hopkins University, specially my friends from the exchange program and my team from Infinite Biomedical Technologies, specially Martin Vilarino, Megan Hodgson, Rahul Kaliki and Nitish Thakor, my mentors and friends.

I am thankful for my friends at the University of Utah who helped me in several aspects during the development of this work, specially Mandi Peterson, Yiman Zhang, Kacey Gao, David Kluger and Katie Aiello. I would like to thank David Warren for the opportunity presented, for all the technical help, for the advices and for being a great host. I would like to thank Zack Kagan for his help during the experiments and all the advices for the project, besides his helpfullness in different other aspects. I would also like to thank David Page and Mitchel Frankel for all the help in providing advices and material for the research.

Lucas de Levy Oliveira

RESUMO

O campo da neuroengenharia cresce a cada dia com estudos promissores sobre suas diferentes áreas, sendo uma delas a estimulação elétrica funcional (FES). Para o avanço desses estudos, diferentes plataformas experimentais biológicas são utilizadas a fim de melhor entender o funcionamento do sistema nervoso e, eventualmente, poder intervir ativamente a fim de recuperar funções perdidas devido a patologias ou acidentes. Além disso, para o melhor estudo da área, diversos pesquisadores se utilizam de artrópodes como insetos. Estas cobaias são muito úteis, devido às suas similaridades nervosas com seres mais complexos e facilidade de manuseio e aquisição. Desta forma, além de serem utilizados em pesquisas que se utilizam de seus tamanhos reduzidos e complexidade biomecânica para realizar tarefas difíceis para a micro-robótica atual, os insetos também são utilizados para estudos introdutórios em neuroengenharia. Neste trabalho, duas plataformas experimentais relativas aos dois cenários descritos para estimulação em nervos são desenvolvidas e testadas – ambas serão aprofundadas brevemente a seguir, a começar pela última.

No Laboratório de Automação e Robótica (LARA da Universidade de Brasília (UnB), foi desenvolvida uma plataforma experimental para implementação de algoritmos de controle de direção para baratas da espécie *Blaberus giganteus*. Esse *setup* contou com o kit de desenvolvimento da Backyard Brains, o RoboRoach, afixado às costas da barata a ser experimentada para estimulação elétrica de nervos de suas antenas. A placa RoboRoach foi interfaceada com um computador pessoal (PC) por meio de protocolo Bluetooth (BT) 4.0, onde o algoritmo de orientação foi implementado e executado. O PC foi também ligado a um sensor de captura de movimento da NDI Digital, o Polaris Spectra. Este sensor, por meio de emissão e recebimento de ondas infravermelhas, provia ao PC informações de posição do marcador passivo afixado à placa nos três eixos de deslocamento.

Desta forma, traçada uma referência retilínea, um algoritmo de orientação simples foi implementado de forma a estimular a antena da barata referente ao lado para a qual esta não deveria ir, fazendo uso de um comportamento evasivo gatilhado pelas antenas das baratas. Assim, a barata poderia ser orientada a seguir uma trajetória retilínea. Um controlador proporcional, ainda, foi implementado de forma que a amplitude de estimulação variasse de acordo com o distanceamento da barata em relação à referência desejada. Para os experimentos, foi utilizada inicialmente uma "barata virtual", referente à utilização de diodos emissores de luz (LEDs) de *debug* presentes na placa, que indicavam qual antena estaria sendo estimulada em determinado instante. Com isso, foi possível validar a eficiência do controlador em si, isolando as variáveis referentes à estimulação da barata. Em seguida, os procedimentos cirúrgicos para experimentação com as baratas foram feitos e os experimentos foram executados.

Os resultados adquiridos dos experimentos mostraram que, apesar do algoritmo de controle se comportar bem em um cenário simulado (da "barata virtual"), houve impedimentos em relação à estimulação dos espécimes. Erros na forma como o *setup* experimental foi desenvolvido e a estimulação foi feita foram encontrados e discutidos. Mesmo assim, resultados satisfatórios no controle de direção das baratas foram obtidos.

O trabalho foi, então, prosseguido no Centro de Neuroengenharia (CNE) da Universidade de Utah. No novo setup experimental, foi desenvolvido um sistema em tempo real utilizando Matlab em um sistema operacional Windows 7. O sistema interfaceava com um *hardware* de nome Grapevine, da Ripple LLC, que provia dados de eletromiografia adquiridos por uma de suas entradas e, além disso, possibilitava modulação de parâmetros de estimulação elétrica em uma de suas saídas. Desta forma, o sistema foi montado em diversas sessões experimentais: inicialmente, para validação do sistema em tempo real, se utilizando de sapos (de gênero *Ptychaden*) e, posteriormente, para desenvolvimento da plataforma experimental e implementação do algoritmo de controle, ratos (Sprague-Dawley). Ambos os grupos de espécimes foram estimulados no nervo ciático utilizando um eletrodo do tipo *hook*, de forma a desencadear uma estimulação de nervo completo (*whole-nerve stimulation*). Os sinais elétricos, assim, ativavam grupos musculares específicos e os sinais elétricos que resultavam em contrações foram gravados e utilizados no sistema de controle. O *setup* contou, ainda, com a denervação de ramificações do nervo ciático, de forma a isolar o tipo de movimento causado pela estimulação, e com a consequente utilização de um sensor de força analógico para posterior comparação *offline* com os sinais elétricos lidos do músculo (*evoked electromyography* – eEMG).

O algoritmo de controle proposto se utilizava de uma curva de recrutamento gerada por meio de variações de amplitude dos sinais de estimulação e o valor máximo absoluto (MAV) das respectivas respostas musculares. Esse procedimento era feito anteriormente à execução do controle, uma vez que este gerava a curva utilizada durante a execução do algoritmo de controle. Este consistia de leituras a cada 20ms (verificados em suas consistências posteriormente) dos sinais de eEMG, cálculo do respectivo MAV e utilização deste na função de controle. Anteriormente a esta, o valor atual de referência era computado, de forma a respeitar a forma trapezoidal desejada, de valores mínimo e máximo entre zero e 50% do valor de saturação da curva de recrutamento. O controle proporcional integral era, então, computado com base no erro entre valor esperado e referência e a variável de saída somada à referência desejada. Uma vez com o valor normalizado final, este era utilizado na função inversa da curva de recrutamento e o valor de amplitude a ser utilizado para estimulação era computado.

Foram adquiridos resultados referentes à consistência do sistema em tempo real e, ainda, à eficiência do controle implementado. Apesar da discussão sobre consistência do período de amostragem se mostrar curta, uma vez que os 20ms esperados foram respeitados com uma pequena faixa de erro, muitos pontos em relação aos resultados obtidos do controle são discutidos. Qualidade da curva de recrutamento, efeitos da exposição do nervo ao ambiente, qualidade do controle utilizando diferentes ganhos proporcionais e integrais, relação entre eEMG lido e força resultante, estratégias de filtragem implementadas, fadiga muscular observada e erros de implementação do controle são discutidos. Por fim, conclui-se que, apesar do sistema experimental (*hardware* e *software*) e do algoritmo de controle poderem ser melhorados no futuro, houve sucesso no desenvolvimento do *setup* e na implementação do algoritmo de controle.

ABSTRACT

This work consists on the development of two experimental setups for the implementation of control algorithms in animals. The first one, developed at the Automation and Control Laboratory at the University of Brasilia, consists on the electrical stimulation of antennal nerves of cockroaches (*Blaberus giganteus*) for direction control based on a retilinear reference. The position of the cockroach was measured by a motion capture system, which communicated with a personal computer (PC) running the control algorithm. The PC communicated with the stimulation board fixed to the cockroach back through Bluetooth 4.0 protocol. The control algorithm implemented was a simple line-follower which steer the cockroach by stimulating the antenna contrary to the desired direction with an amplitude proportional to the error between measured and reference positions. The second experimental setup took place at the Center for Neural Engineering at the University of Utah, where experiments using grass frogs and Sprague-Dawley rats were conducted. For both species, a hook electrode was used for whole-nerve stimulation of the sciatic nerve and wire electrodes were inserted into gastrocnemius muscles in order to record evoked electromyography (eEMG) signals. A proportional integral control algorithm was implemented in order to control muscle activation based on maximum absolute value (MAV) of the eEMG measured and recruitment curves found beforehand. Several aspects were discussed for both experimental setups regarding efficiency of the controllers and problems found in development and implementation. Finally, it is concluded that, even though there are several aspects to be explored in relation to improvements in both projects, the results found were satisfactory for initial trials.

Keywords: neural engineering, neural stimulation, evoked electromyography, proportional integral control

SUMÁRIO

1	INTRODUCTION	1
1.1	CONTEXT	1
1.1.1	NEURAL ENGINEERING AND INSECTS	1
1.1.2	NEURAL ENGINEERING AND FUNCTIONAL ELECTRICAL STIMULATION	2
1.2	GOALS	2
2	LITERATURE REVIEW	3
2.1	NEUROBIOLOGICAL BASIS OF MOVEMENT	3
2.1.1	INSIDE NEURONS	3
2.1.2	BEYOND NEURONS	5
2.2	INTERFACING WITH NEURONS	8
2.2.1	LISTENING TO THE NERVOUS SYSTEM	10
2.2.2	TALKING TO THE NERVOUS SYSTEM	13
2.3	CONTROL OF ARTHROPODS NAVIGATION	17
2.3.1	PRINCIPLES OF NEUROPHYSIOLOGY IN INSECTS	18
2.3.2	INSECTS ARTIFICIAL NAVIGATION USING NEURAL INTERFACES	19
2.4	FES CONTROL USING EVOKED EMG	20
3	INSECT DIRECTION CONTROL USING NEURAL INTERFACES	23
3.1	METHOD	23
3.1.1	ROBOROACH DEVELOPMENT KIT	24
3.1.2	POLARIS SPECTRA	24
3.1.3	DEVELOPING OUR OWN API	26
3.1.4	COCKROACH PREPARATIONS	27
3.1.5	THE EXPERIMENTAL SETUP	28
3.1.6	SIMULATIONS AND EXPERIMENTS	29
3.2	RESULTS	31
3.3	DISCUSSION	33
4	FES CONTROL USING EVOKED EMG	35
4.1	METHOD	35
4.1.1	GRAPEVINE	36
4.1.2	XIPPMEX	38
4.1.3	DEVELOPING THE REAL-TIME SYSTEM	38

4.1.4	CONTROL ALGORITHM	39
4.1.5	FROG AND RAT PREPARATIONS	41
4.1.6	EXPERIMENTS	43
4.2	RESULTS	44
4.3	DISCUSSION	51
5	CONCLUSIONS.....	55
5.1	FINAL CONSIDERATIONS	55
5.2	FUTURE WORK	55
5.2.1	INSECTS NAVIGATION CONTROL USING NEURAL INTERFACES	55
5.2.2	FES CONTROL USING EVOKED EMG.....	56
REFERÊNCIAS BIBLIOGRÁFICAS		58
ANEXOS		65
I	 DESCRIÇÃO DO CONTEÚDO DO CD.....	66
II	 PROGRAMAS UTILIZADOS.....	67

LISTA DE FIGURAS

2.1	The diagram of a common action potential related with channels open.	4
2.2	Different waves found in electromyography signals.	6
2.3	The different types of electrodes in the order they were cited.	9
2.4	Block diagram describing preprocessing stages before decoding implementation.	12
2.5	Electrical stimulation parameters.	14
2.6	Expected effect of generated recruitment curve.	16
2.7	Example of remotely operated navigation control in a cockroach. Adapted from Latif <i>et al</i> [1].	18
2.8	Basic neurophysiology of insects.	19
2.9	Cockroach atop a trackball used for sensing. Adapted from Holzer <i>et al</i> [2].	20
2.10	Evoked EMG waveforms when stimulated by CIT and CFT. Adapted from Binder-Macleod <i>et al</i> [3].	21
3.1	Block diagram describing the methods utilized.	24
3.2	The volume within which positions are acquired. Source: NDI Digital.	25
3.3	Final result from electrode implantation and board fixation procedure.	28
3.4	Arrangement used for the experiments. Polaris Spectra is positioned facing the floor, where a line indicates the desired path, namely, the reference path.	29
3.5	Control algorithm block diagram.	30
3.6	Voltage output measured from board with 1 and 10 pulses.	32
3.7	Voltage output measured from board with pulse-width of 1ms, 10ms and 50ms.	32
3.8	Simulations using the virtual roboroach setup.	32
3.9	Experiments using the real roboroach setup. In blue, the desired path the cockroach should follow. In red, the actual path traveled.	33
4.1	Block diagram of the simplified experimental setup.	36
4.2	Grapevine's NIP and front-ends. Source: Ripple.	37
4.3	Block diagram of the control system implemented.	40
4.4	Experimental setup used in frog experiments.	42
4.5	Experimental setup used in rat experiments.	43
4.6	Force sensor attached to rat foot.	44
4.7	Force sensor (to the left) fixed to rat finger. The ankle position is fixed in order to acquire better measurements.	44

4.8	Loop periods during oscilloscope recording and stimulation. UDP communication was turned on during the execution.	45
4.9	Loop periods during certain experiment on frog. UDP communication was turned on during the execution.	45
4.10	Data recorded during initial script running on second frog experiment.....	46
4.11	Loop periods during certain experiment on rat. UDP communication was turned on during the execution.	46
4.12	Representation of evoked EMG recorded for single-reference (above) and differential electrodes (below).....	47
4.13	Evoked EMG in rat zoomed in; long experiment for fatigue perceiving.	47
4.14	Recruitment curves before and after electrode repositioning.	48
4.15	Respective control output for first recruitment curve in Figure 4.14.	48
4.16	Recruitment curve found in second leg.	49
4.17	Control output before and after moving window filtering.	49
4.18	Evoked EMG during control execution and respective force measured by transducer.....	50
4.19	Control outputs for different k_p gain values.	50
4.20	Control output recorded in last experiments.....	51
4.21	Control output recorded in very last experiment.	51

LISTA DE TABELAS

3.1 Values used during experiments with cockroaches.....	31
--	----

Chapter 1

Introduction

*Biotechnology is not inherently good or bad; it is simply a set of techniques, and we have choices about how we employ them. If we use our scientific superpowers wisely, we can make life better for all living beings—for species that walk and those that fly, slither, scurry, and swim; for the creatures that live in scientific labs and those who run them. So it's time to embrace our role as the dominant force in shaping the planet's future, time to discover what it truly means to be stewards. Then we can all evolve together. — Emily Anthes, *Frankenstein's Cat* [4]*

1.1 Context

As both medicine and rehabilitation engineering advance their knowledge and techniques regarding the amelioration of neurological dysfunctions, no definite cure for certain pathologies and impairments, such as paralysis and amputation, have been developed yet [5, Foreword]. In this sense, a new field arises named *neural engineering*, defined as "an emerging interdisciplinary research area that brings to bear neuroscience and engineering methods to analyze neurological function as well as to design solutions to problems associated with neurological limitations and dysfunction" [6].

1.1.1 Neural Engineering and Insects

Several approaches have been used in order to better understand the nervous system as a whole, from experiments with humans and other mammals, to the usage of less complex animals in experiments. Some of these researches have found in the study of insects a good approach in both exploring nervous system and development of important applications. Hence, the study of insects nervous system has been tied to two main factors: the development of instrumentation and algorithms implementation for interfacing with nervous systems and the usage of these strategies allied with insects natural minimized size and complex biomechanical responses needed for specific tasks.

Regarding the last aspects, the alliance between insects capabilities and recent technology has

resulted in several interesting applications, such as the usage of controlled-flight beetles that could be used as micro air vehicles (MAVs) with better miniaturization, payload and performance [7]; moth swarming for search and rescue in inhospitable areas [8]; and terrestrial insect navigation controlling for disaster sites exploration and search for possible victims [9]. Thus, it is understood nowadays the advantages of these researches, as they allow us to understand aspects of neural engineering applicable to other animals and, also, provide real-world applications.

1.1.2 Neural Engineering and Functional Electrical Stimulation

Neural engineering has seen three main fields emerging throughout the last years: *stereotactic and functional neurosurgery*, *neuromodulation* and *functional electrical stimulation* (FES) [10, p. 1-1]. Though the first two fields bear their great significance, we will focus mainly in the latter for this work.

FES is a method consisting of applying low intensity electrical currents in order to restore or improve specific functions [11, Introduction]. Most of these functions are related to muscle activity control, such as bowel and bladder control, even though some relate to visual and auditory prostheses [10, p. 1-2]. In this work, the control of skeletal muscles will be the main focus. As just noted, FES does not aim at reversing paralysis once the system is turned off; however, in some applications, the user may receive benefits regarding the recall of voluntary muscle functions. Thus, among several other uses, it has been used in patients in order to maintain active muscle activity during physiotherapy sessions.

1.2 Goals

The goal of this work is to explore the neural engineering realm by working with different animals while developing different applications. We propose a navigation control for cockroaches using neural engineering techniques and a motion-capture sensor in order to trace paths on which the insect should thread. Also, a FES system is developed in order to control muscle activation in small animals, such as frogs and rats, by using whole-nerve stimulation and muscle electrical signals acquisition.

Chapter 2

Literature Review

There is only one nature – the division into science and engineering is a human imposition, not a natural one. Indeed, the division is a human failure; it reflects our limited capacity to comprehend the whole. — Bill Wulf

2.1 Neurobiological basis of movement

The union between engineering and neuroscience, although having shown to be beneficial for humankind, requires for one to understand basic principles of the other. Thus, while preoccupied with a world of equations and algorithms, engineers must also plunge into a more biology-centered domain and comprehend principles related to neurons and their properties. In order to act upon the system, the engineer must understand it – as the next sections should be helpful with.

2.1.1 Inside Neurons

One cannot understand the signals that are read in nerves or the brain without having a primary discerning of what a neuron is or does. A neuron – or brain cell – differentiates itself from a usual cell by being excitable or, more specifically, by specializing in the propagation of electrical current [12, p. 193-194]. The opening and closing of certain electrically charged elements channel an electrical stream along the cell. The *dendrites* serve as the neuron inputs and drive the current passing through the *soma* – the cell body – into the *axons*, which are considered the outputs. However, this current is only fired through the soma if the inputs exceed a certain threshold. This flow of charge is usually called an *action potential* – commonly used when referring to the waveform – or *spike* – which represents the action potential in a wider time-scale.

Due to the characteristics of the ionic channels and their activation timings, action potentials have very specific waveforms depending on several factors, from the cell's own properties to the position of the neuron related to the recording instrument [13]. The main qualities of spikes are *amplitude*, *duration* and *firing rate*. All of these characteristics are going to be primal when dealing with the stimulation parameters. For now, it is necessary to understand them carefully.

Amplitude refers to the highest voltage value the action potential achieves within its duration. This duration is named pulse-width, which contemplates, in time, all voltage values from the beginning to the end of the wave. Spikes, however, do not always present themselves in a single firing, but in a *spike train*, which is a burst of several spikes evoked in sequence. The period at which they are discharged is the inverse of the spike frequency.

Another essential attribute of action potentials is their biphasic polarization. As the typical sequence of events takes place, when a conducting region of the neuron is electrically excited, specific channels activate, generating an inward flow of positive charges. This *depolarizes* the cell, which is originally around negative 70mV (the *resting potential*), activating another set of channels responsible for the outflow of positive charges. This creates a steep decrease in voltage (*repolarization*), reaching a negative value even lower than the resting potential [14] – in other words, a *hyperpolarization* – resulting, thus, in the biphasic shape. After all channels are closed, there is an specific delay within which the cell cannot be stimulated, called *refractory period*.

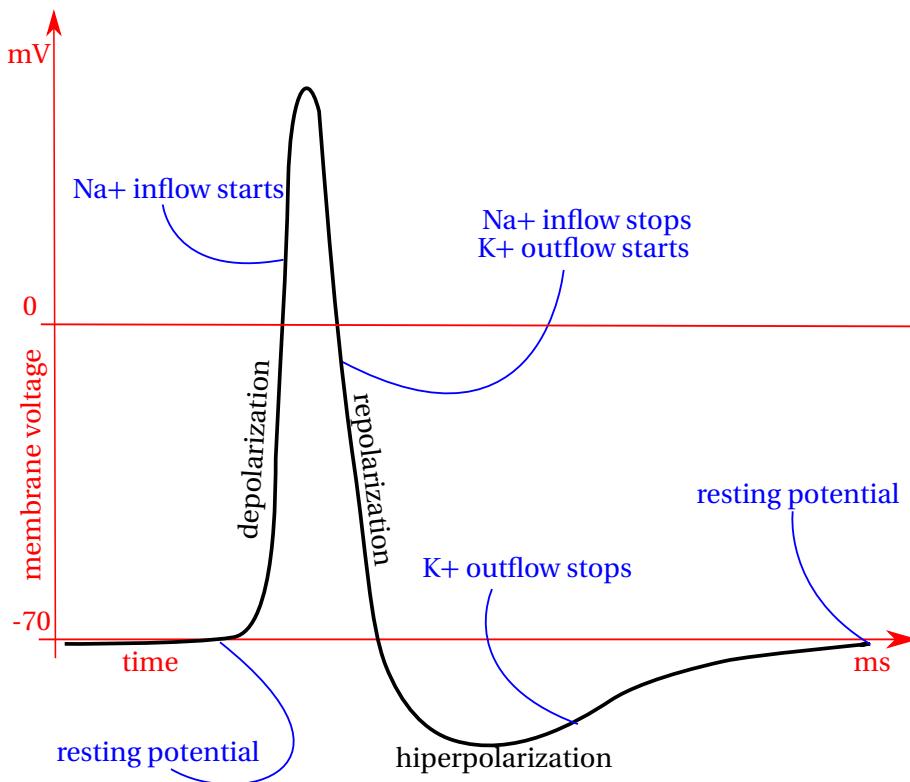


Figure 2.1: The diagram of a common action potential related with channels open.

The action potential then travels along the cell until the end of the axon, where it excites calcium channels, guiding *synaptic vesicles* to the membrane. These vesicles release neurotransmitters that will excite the next cell in the *synaptic cleft* [15, p. 160-166]. This pulse travels along the nerves, from the brain to the nerves and *vice versa*.

Therefore, when dealing with motor and sensory interfaces, there are two well-known pathways followed by the signals: from the *motor cortex* to the *motor units* and from *skin sensors* to the *somatossensory cortex*.

2.1.2 Beyond Neurons

In order to understand the usual pathways studied by neuroengineers, it is important to acquire knowledge about the routes, structures and the signals that interest the engineering realm. Neuronal cells are, hence, zoomed out to nerves, lobes and cortices.

2.1.2.1 The Brain Moves

Starting from the first path aforementioned, most signals that are able to actually move body parts originate primarily from the *premotor cortex*. Within it, the brain designs how movement will be executed, although there are evidences that it is also involved in monitoring of accomplishment and performance [16, p. 255]. Signals are delivered to the motor cortex, which translates movements planned in premotor cortex to a more low-level and specialized context. These signals travel down the brain, passing through the *corticospinal tract* and, finally, the *spinal cord*, where they branch out to different *peripheral nerves* [16, p. 686-690]. The spinal cord plays, thus, the role of a nerval hub – besides being involved in different types of autonomous movements by providing simple circuitries involving both *efferent* and *afferent* pathways [16, p. 558]. Inside these nerves, efferent fibers carry motor-related signals – as opposed to the afferent pathways, which carry sensoring information to the higher-up levels [16, p. 675-678].

Each of these efferent motor fibers, then, innervates different muscle fibers. Every single one of these groups of connections are called *motor units*. At this point, it is interesting to point that, usually, fast-acting and accurately controlled muscles have a large nerve-fibers-to-motor-fibers ratio, whereas slow-acting muscles who do not require precise control have a smaller ratio [16, p. 81]. Once the signal arrives at the muscle, *actin* and *myosin* filaments contract in an intricate mechanism, which generates force and contracts the muscle. The resulting coupled system of tendons and bones generate motion as we know it; however, it will not be thoroughly detailed in this document.

As the neural signals, who originated far back in the premotor cortex, travel through the muscle fiber, a signal can be recorded for analysis — this is called *electromyography* (EMG) [17, p. 179-180]. EMG signals can be divided in two branches: those which are part of a natural process, as being described so far, called *volitional EMG*, and those that are result of electrical stimulation, called *evoked EMG* (eEMG). Even though volitional EMG is described merely by voltage, evoked EMG has specific features which differ from the latter. For example, there are two main components that can be assessed when measuring eEMG. The first one, called *H-wave* (or Hoffman reflex, H-reflex), travels within the same pathways as *muscle spindles* (sensors located inside the muscles that translate length into electrical activity). H-waves are typically easily excited, meaning their threshold for electrical stimulation is much lower. The second main component of EMGs is the *M-wave*, which, opposingly, is excited by higher signals. This wave is usually more closely related to the muscle contracting be-

havior itself, being the one most commonly used in motor analysis. Apart from the other two waves aforementioned, there is a third type, called the *F-wave*, which is generated once the motor fiber is hyperexcited, producing afferent and efferent stimuli, the latter generating the M-wave. The former travels through the nerve back to the spinal cord, where it returns as the F-wave, contracting the muscle after the M-wave does so [18, p. 127].

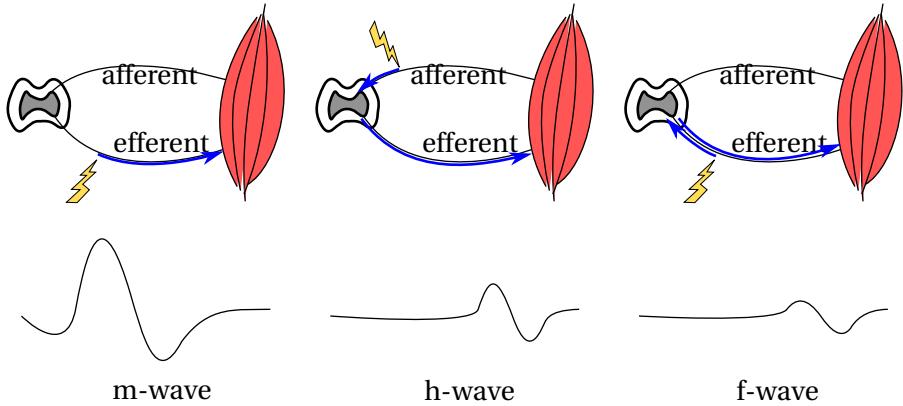


Figure 2.2: Different waves found in electromyography signals.

So far, it has been described the behavior of the motor system when a single signal comes forth. This signal travels through the pathways described and, in the end, generates a single *muscle twitch*, which is a quick contraction of the muscles, which return to the original length due to their passive stiffness. In everyday experiences, however, it is known that this is not the usual performance of members – as they are able to maintain the muscles contracted by long periods until they can no longer be held steady. In order to do so, the neural system has to fire the motor units quickly and repeatedly in a short timespan. As explained above, this form of activity is described as a spike train. The muscle, acting as a low-pass filter, holds the spike for a while once it is excited. Accordingly, in order to maintain the same contraction fashion for some time, the firing rate increases to frequencies close to 50Hz , resulting in what is referred to as *tetanic contraction* – or *tetanus*. Obviously, the comparison with everyday living falls short to explain everything, since the recruitment of different muscles within the same *muscle group* are not being considered. On the other hand, it is a good instrument to ease the comprehension of other factors, such as the muscle's inability to hold contraction for a long time [5, p. 39].

After this prolonged muscle work, the muscle does not present itself as capable of achieving the same amount of contraction than before, in a situation regarded as *muscle fatigue*. Physiologically, it is involved in several situations. Fatigue can occur, for example, by interruption of blood flow, resulting in a decrease of nutrients necessary for muscle contraction. Another example is in prolonged activation and depletion of nutrients which were not consumed by the body – which is a common cause in laboratory experiments [16, p. 82]. In this situation, yet, it is most probably explained by the necessity of engaging in an active muscle process named *relaxation*, which consumes, besides time,

ATP (*adenosine triphosphate*, the substance involved in muscle contraction itself and in general active processes around the body) [19, p. 683]. It is interesting to bring out how muscle fiber types can affect muscle fatigue, being divided by two main kinds: *slow fibers* and *fast fibers* muscle fibers. The first one is characterised by the presence of small fibers and great amounts of capillarity. This results in two main aspects: a smaller generation of force and better resistance against fatigue. In counterpoint, fast fibers are acknowledged as larger – hence, greater force output – and with lesser blood supply, which reveals to be less resistant against fatigue [16, 5, p. 80-81, p. 43]. Other nomenclatures when referring to these fibers are, respectively, *type I* and *type II* or *red* and *white* muscle fibers – the former pair referring to the presence or absence of hemoglobin, which acts as a source of oxygen and, thus, fatigue resistance [20, p. 17].

As fatigue, there is another muscle time-varying characteristic, the *muscle potentiation*, which is caused by increased sensitivity of calcium ionic channels after previous stimulations. This results in a substantial increase of muscle output force and stiffness [21]. All these nonlinear characteristics are truthfully important when designing control interfaces, which will be discussed more deeply further on.

2.1.2.2 The Brain Feels

As there are specific neural pathways for carrying information such as motion, there are – just as important – *sensory pathways* necessary for our everyday activities, such as temperature, pressure, vibration and even pain perception. These specific variables are usually mediated by *exteroceptors*, that is, receptors or skin sensors that perceive external stimuli. The other main group of receptors is named *proprioceptors*, relating to receptors that provide information about the human's own body, such as muscle length, speed, contraction or even joint angle. This type of information is essential for the brain to be able to finely control motion, serving as closed-loop feedback. This distinction, sometimes, may fall into gray areas, as exteroceptors may all the same contribute to motion, for example, in high-temperature reflex arcs [19, p. 147].

There are other types of receptors in the body that should be mentioned, such as visual receptors in eyes and auditory receptors in ears. The receptors mentioned before, nevertheless, are known as *somatic receptors*. The usual pathway for their signals is to travel in afferent neurons through peripheral nerves into the spinal cord (where some of the sensors are processed in basic circuitries) and towards different areas of the nervous system, including: the medulla, pons, mesencephalon, cerebellum, thalamus and, finally, the somatosensory cortex [16, p. 555].

Despite of the long list of sensory receptors present in vertebrates, this reality is a tad different when dealing with insects. While keeping the terminology of exteroceptors and proprioceptors, some sensors vary according to arthropod's inner characteristics, such as the presence of an exoskeleton. Taking the cockroach's antenna (which is a very interesting subject to study, as it will be noted later) as an example, most of the proprioceptors are actually located externally to the body and can act as exteroceptors. In other words, while some *mechanoreceptors*, receptors that react to mechanical changes – such as the *campaniform sensilla* or the *articulated sensory hair* – but only act as exteroceptors [22, p. 78], others can also work as proprioceptors, e.g. *hair plates* [23]. One of the most im-

portant aspects of the cockroach's antenna, in fact, is a characteristic noticeable in different insects: *active sensing*. Because of its innervated nature [24], these insect parts can move in different directions independently and be used in different sorts of behaviors [25]. The number of different actions is surprisingly long (including anemotactic behavior, wall-following and electrostatic field detection), the one with most interest for this work being the *evasive behavior*, which takes place when a cockroach perceives a strange obstacle as a predator – resulting in a quick turn-and-run acting [26].

2.2 Interfacing with Neurons

A lot has been told about the physiology involved in afferent and efferent pathways, motor units, muscles and sensory receptors; therefore, it is time to discuss about how these signals can be recorded, manipulated, generated and used for tightening the interface between man and machine.

It is primarily important to understand the basic terminology used in this sort of interface. *Neural recording*, for example, is the name of the technique used for acquiring neural signals, whereas *neural stimulation* is how machines interface with the nervous system, generating waveforms similar to biological ones and exciting the nerves from an external end. As interfaces vary, these names will keep their main name, as in "EMG recording" or "muscle stimulation". Moving on, this physical interface between machine and human is called an *electrode*, which comes in different sizes and shapes. Usually, there is an explicit trade-off between how invasive the electrode is – how complex is the surgical proceeding to implant it or not – and how selective it can be – namely, how many different neurons are interfaced with a single channel [27].

There are a few types of electrodes that can be mentioned:

- *superficial electrodes*, which provide a less invasive and less selective interface (generating lots of *crosstalk*, i.e., when two different muscles are recorded by the same channel [28]), being placed usually on the skin of the user;
- *hook electrodes*, which are designed as two or more hooks (depending on the number of channels) on which the nerve is "hanged", discarding the electrical noise caused by tissues and the skin-electrode interface that commonly poses as a problem for superficial electrodes users [29], besides having better fixation than the latter [30, p. 45];
- *cuff electrodes*, whose shape follows the nomenclature by covering the nerve, guaranteeing a better-yet-not-enough selectivity;
- *Longitudinal Intrafascicular Electrodes* (LIFE), which are single wires inserted into the nerve in a longitudinal fashion;
- *TIME electrodes*, the *transversal intrafascicular electrodes*, which follow the same logic as the latter mentioned, only transversally [27]; and
- *Utah Electrode Array* (UEA), the one that has become more and more famous by the day due to its increased selectivity, number of electrodes and reach within peripheral nerves [31].

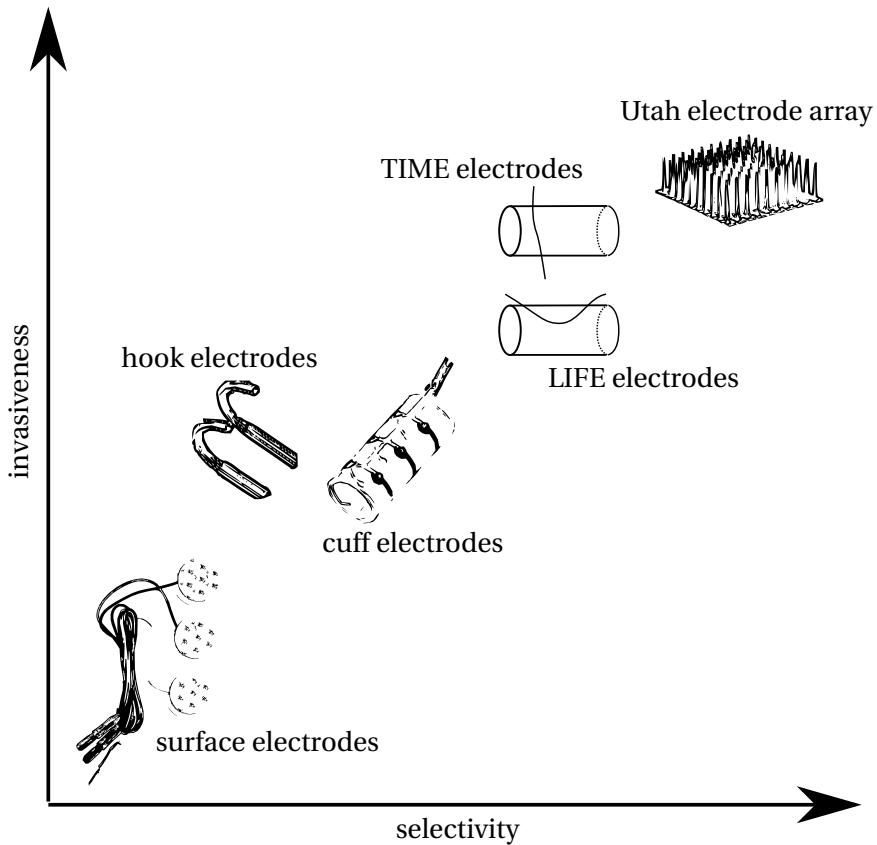


Figure 2.3: The different types of electrodes in the order they were cited.

An important characteristic that accompanies electrode selection when stimulating is *current density*. When delivering an specific amount of charge to the nerve, the gauge through which this current will run is of great importance. In case a high charge is forced upon a narrow-diameter interface, the effect might involve dispersion through heat, which can cause pain to the patient and even harm the tissue, which is not at all advisable. By the same token, if a superficial electrode of broad contact area is selected, for example, the current may be diffused, resulting in a less-than-ideal stimulation threshold [5, p. 209-210]. Another aspect worth mentioning is that both recording and stimulation usually happen in a *single differential configuration*, that is, a channel is always used for either stimulating or recording, whereas at least one channel must be set as reference [32, p. 274]. The signal measured, then, will be the difference between the signals measured by each electrode. In case of EMG (or even *superficial EMG* – sEMG – where wavelengths may be larger than the interelectrode distance), this reference can be used with more channels, allowing a technique named *spatial filtering*, that significantly decreases low frequency artifacts – which will be detailed further on [5, p. 400].

As discussion over the motor system preceded, this very same order will be followed as related techniques are explained, allowing the phisiology discoursed to serve as an example. When interfacing with the motor system, many applications involve recording signals from either the nerves or the muscles themselves, as it follows.

2.2.1 Listening to the Nervous System

Whether one talks about common interfaces as sEMG-interfaced commercial prosthetic hands, more rare practices as targeted muscle reinnervation [33, p. 300], or even more complex procedures as surgical placement of an electrode array into a nerve for controlling a robotic hand, one thing is certain: how signals are read and recorded must be thoroughly understood.

Once the desired electrode has been decided through research and the counter-balance of advantages and disadvantages, the electrode must be placed on the appropriate location. This involves studying the neurophysiology of the species at hand and physical particularities of the subject, as well as the most suited site for the current application. For this goal, all sorts of equipments are used, such as oscilloscopes, muscle stimulators and, as strange as it may sound, audioamplifiers [5, p. 159].

The next main step involves *amplifying* and *filtering* the signal. Since signals recorded from these types of interfaces usually wander from the microvolts to the millivolts ranges, these signals must be magnified in order to be processed by usual electronics. For example, in EMG amplification, a series of *operational amplifiers* must be used in order to attain a good signal quality, usually represented by high input impedance, high common mode rejection ratio and low noise. This series can be further described as the association of voltage followers and instrumentation amplifiers [32, p. 275]. Once the signal is amplified by tens or hundreds of its value, it is valuable to apply usual filters in order to decrease overall noise. One of the filters that may be used, as usual, is a *notch filter* of either 50Hz or 60Hz as the cutoff frequency. However, in case of neural recording, the signal main frequencies lie between specific ranges, such as $150\text{Hz}-15\text{kHz}$, allowing a simple *band-pass filter* to be used [34, p. 483]. In case of EMG, the same band-pass filter can be implemented, however changing its cutt-off frequencies to the range of $1\text{Hz}-3\text{kHz}$ [34, p. 557].

While discussing signal preprocessing, it is worthy mentioning how different external stimuli may interfere with the recorded signal. The stimuli mentioned, in fact, does not need to be of electrical nature only; in other words, these disturbances may happen because of concurrent electrical stimulation or even movement-related aspects. The subject in matter is *stimulation artifacts* and *motion artifacts*. The latter may be present, for example, during sEMG acquisitions, when fixation of the superficial electrodes is not optimal, resulting in distinctive disturbances in the waveform. For this specific case, it has been proposed as solution the light abrasion of the skin surface using a fine sandpaper [35]. However, for artifacts produced by simultaneous electrical stimulation, the approach may involve less physical elements. This will be discussed along with the processing of eEMG signals soon.

Once the signals read are properly amplified and mostly free of noise and artifacts, hopefully by hardware implementation approaches, it is time to convert them to digital data (suitably using an A/D converter) and actually understand their meaning. The two major signal types usually recorded, notwithstanding, have different processing procedures, which will be detailed followingly. Of course, the types being mentioned are neural signals and muscle signals (eEMG), which shall be discussed in order. Neural signals waveforms have been explained earlier in this report, thus, from this stage onward, the plan is to extract those action potential waveforms and start the *decoding* process, which consists in translating the characteristics of the spikes (usually the firing rate) into information that can be used in applications, such as joint angle and skin pressure. The first software implementation

of the input signal is called *spike detecting*, which consists in actually finding the times at which a spike occurred. In order for this to happen, different algorithms can be implemented; yet, a somewhat simplistic approach is usually chosen in spite of the others due to its computational efficiency and ease of implementation: the *threshold detection*. Because of action potential's natural high amplitude peak, a voltage threshold trigger can be used to detect spikes. Thus, once the most proper value for the dataset is set – which can be found either manually or automatically during an offline analysis – the great majority of spikes will be found by this algorithm, since they tend to maintain their characteristical waveform, provided that source location is kept constant [36, p. R57]. This path has been continued by others, who have developed a double-threshold algorithm, consisting of positive and negative thresholds (contemplating both anodic and cathodic polarizations), which may decrease the ratio of false detections, along with filtering implementations [37].

Having found the action potentials or, rather, being able to do so for the following ones, it is time to actually classify the waveforms extracted. This is important since electrode channels are not always capable of accurately pinpointing a single neuron (being addressed thus as an *extracellular recording*), which would provide the waveform that has been made familiar in this document, but rather multiple different signals and waveforms. The general shape of the action potential is known; however, depending on geometry of electrode and distance from recording point, the signals may vary in amplitude or pulse width [38]. Hence, sorting algorithms are implemented, which may rely on two aspects: the overall form of the spike (*template matching*) or specific properties of the signal that are used in order to group them in similar categories, or neuron sources (*feature clustering*).

The first group of algorithms work basically by using predetermined spike templates, found sometimes by creating a "mean waveform" gathered from previous samples and, added to white noise, find the probability that each template is found in the dataset. The second group extracts features from the waveforms, such as *peak time*, *peak-to-peak voltage* or *settling time* and clusters the signals based on these attributes [38]. Other approaches, still within the second group, rely on acquiring the eigenvectors of the waveforms set covariance matrix. This is called *Principal Component Analysis* (PCA), a method that finds the orthogonal basis vectors that follow the largest variation within the data, i.e., an algorithm that automatically selects the best features to use when classifying groups of signals [36]. PCA usually does well in matters of spike sorting; however, it may be not enough in specific cases, such as when the main variation of the signal is in time, not in amplitude. For these situations, algorithms which deal better with time-scaling have been tested and shown to work better, such as the usage of *wavelet coefficients* [39]. Once these features were extracted, different kinds of clustering algorithms can be used, from eyeballing the regions and setting thresholds to algorithms such as *K-means* or even *Superparamagnetic Clustering* (SPC) [39].

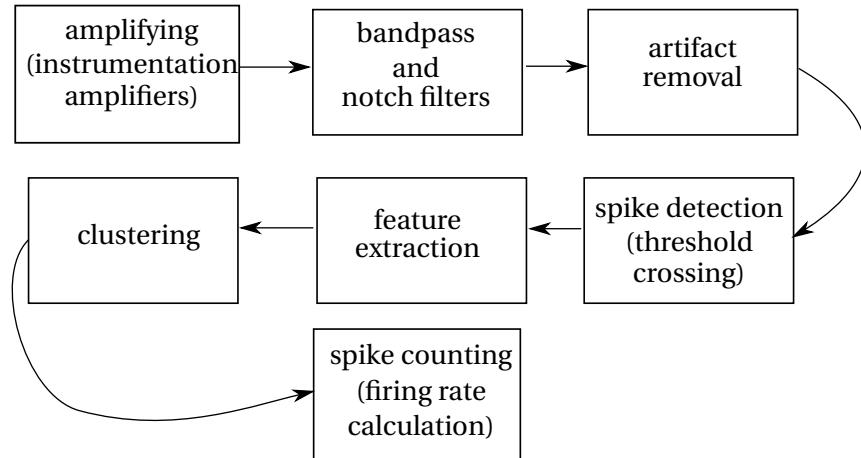


Figure 2.4: Block diagram describing preprocessing stages before decoding implementation.

The following step from having classified the action potential according to their origin is to finally acquire the firing rate of each neuron and try to correlate it with variables to which it is wanted to translate. In other words, an *estimator* that will interpret the firing rate (which can be done by counting the number of spikes within a moving window) and output information about, for example, how contracted is the muscle, whether there is pressure on the skin or not, or even what temperature the body is sensing in an specific place. Followingly, neural signals will be placed on hold, as muscle signals treatment are discussed.

The first thing that should be noted when understanding the procedure of treating evoked EMG signals is that, since it originates from muscles, it will most probably be elicited when a signal generates a contraction. For this to happen, a stimulating signal must travel through the nerve, as it has been discussed before. In other words, when an electrode reads the muscle signals, it may also read the electrical stimulus as well; thus, strategies must be implemented in order to treat this titled artifact. Two known approaches are the usage of *blanking windows* and *finite impulse response filters*. The former one consists on nulling the signal read for an specific period at which the stimulation is present, whereas the latter consists on applying a filter to the whole recorded signal, which can be treated previously by the first approach [40]. While these methods were implemented in software and may be more time costing, other methods have been implemented in hardware, such as the *fast settle triggers* from Grapevine [41, p. 32]

While it is not the case when dealing with evoked EMG, one may notice how discontinuous the volitional EMG signal shows to be; instead of seeming well-behaved as shown in Figure 2.2. In order to achieve the signal mentioned, another processing must take place. After the signal has been properly rectified via either software or hardware (preferably by the latter), a *linear envelope* should be applied to the EMG signal, which will, as the name suggests, interpolate the peaks of signal, creating a smoother curve. Despite of some studies being able to show how to implement this method in a manner that is fast and efficient enough to be used in realtime [42], other methods may still be used online, such as peak-to-peak amplitude and mean amplitude value. Once this waveform is achieved, the am-

plitude of the volitional EMG can be used for implementing control algorithms to activate prosthetics [42], for example.

The procedures aforementioned are the ones which are commonly implemented in order to use decoding and control algorithms. For the next part, methods used for encoding signals, i.e., providing action potentials that can be understood by the nervous system, will be explained.

2.2.2 Talking to the Nervous System

Now that the mysteries behind data acquisition from the nervous system have been made a bit more clear, it is time to understand how one can emulate neural signals into it. As it has been discussed before, action potentials have specific shapes that should be recreated in order to generate an ideal response. Whether this response will go to efferent pathways and stimulate the muscles or to afferent pathways and stimulate the brain in order to modulate specific sensations such as heat and vibration, orientations should be followed in order to obtain better responses. For this goal, a few properties of *neural stimulation* will be discussed next.

One of the first properties of the signal are related to the waveform, such as the signal shape: whether it is rectangular, sinusoidal, triangular, etc. It is shown that the shape must affect more directly in the rising time of the signal, which should be fast enough for the nerve membrane not to accommodate and open the channels. For this purpose, most commonly a *rectangular-shaped stimulus* is preferred. Another feature is related to whether the signal is *bipolar* or *monopolar*, meaning whether the signal changes polarization within its pulse-width (resulting in a bipolar pulse) or not (monopolar pulse). The former is usually chosen due to better comfort and tissue integrity issues, besides preserving the electrode capabilities due to the balancing of charges [5, p. 209]. Another feature that may be added to the waveform is an *interphase delay*, which is represented as an interval between the primary and secondary pulses. This delay may result in a decrease of the threshold separation, meaning a decrease in the slope of the recruitment curve [43], which will be better detailed later on.

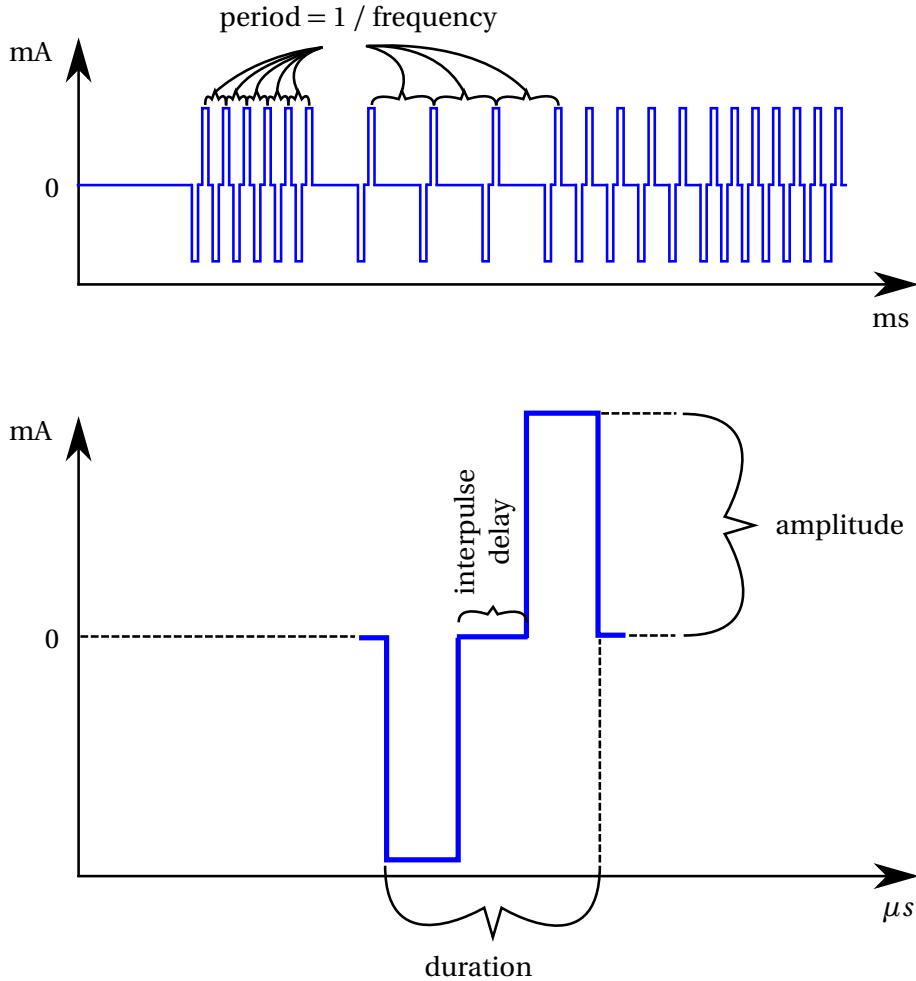


Figure 2.5: Electrical stimulation parameters.

Another property to be considered is the nature of the stimulation itself – namely, the contrast between *current-regulated* and *voltage-regulated stimuli*. Though some commercial electrical stimulators still apply the latter, it is not the most recommended. This is explained by the fact that what actually controls neural signals is electrical charge, which is more directly related with current than voltage. Since the impedance of the system being stimulated is not of resistive nature only, the current delivered will not be adequately. Also, if the impedance between the electrode and the neuron is too high, the charge delivered by voltage-regulated stimulators may be too weak to reach an stimulation threshold – thus, generating no response; by the same token, if the impedance is too small, this may incur in discomfort or even pain to the subject [5, p. 208]. A change of impedance during stimulation, thus, will result in undesired responses.

Followingly, as the signal being used is already settled for, one must decides on which sort of *modulation* will be used. Modulation, in this situation, refers to how changes in an specific variable may be used to affect others. In this case, one must choose between modulating muscle force or sensations by changes in the waveform properties. Two main types of modulation are then presented: *pulse-width modulation* and *amplitude modulation*. Despite that the differences in the sensations or contractions elicited by them are usually very slight, the former is mainly implemented due to its

ease of execution, as generating high resolution in the time axis has generally a more straightforward implementation [44, p. 359].

In case of electrical stimulation on efferent pathways, there are specific matters that should be discussed more thoroughly. Knowingly, efferent pathways stimulation will have direct effect on muscle contraction, but how exactly a specific stimulus may affect the EMG response is a more complex matter. This is more easily explained by *recruitment curves*, which graph the relation between a given stimulation parameter (e.g., pulse-width or amplitude) and amplitude of activation, which can be described by maximum EMG elicited, generated force in newtons, highest peak-to-peak recorded value, etc. Of course, the relation shown can be represented in a three-axis plot, being two of them variables that contribute in stimulation and the third one, generally, once again related to the EMG value [45].

These curves are usually sigmoidal, being divided in three main sections: the *deadzone*, where stimulation does not evoke any contraction; the *high-slope*, in which a behavior that can be approximated to a linear function is found; and *saturation*, the point at which any increase in stimulation will result in no higher activation in the muscle [46]. As noted before, it has been shown that an decrease in the interphase delay value will result in a greater *threshold separation*, meaning that the second section – the high-slope – will demonstrate a steeper slope. In parallel, an increase in this parameter provides a curve with better resolution and, thus, more fitting for applications involving different muscle activations [43]. Another significant usage of the recruitment curve is when the interface being used is that of an electrode array. In these cases, it is not known for sure how well each single channel of the array affects movement or sensation; in order to find out, one can make use of the recruitment curves for each of the electrodes and, based on them, define individual *selectivity indexes*, which should specify how well the targeted pathway is affected compared to the others [47, p. 443].

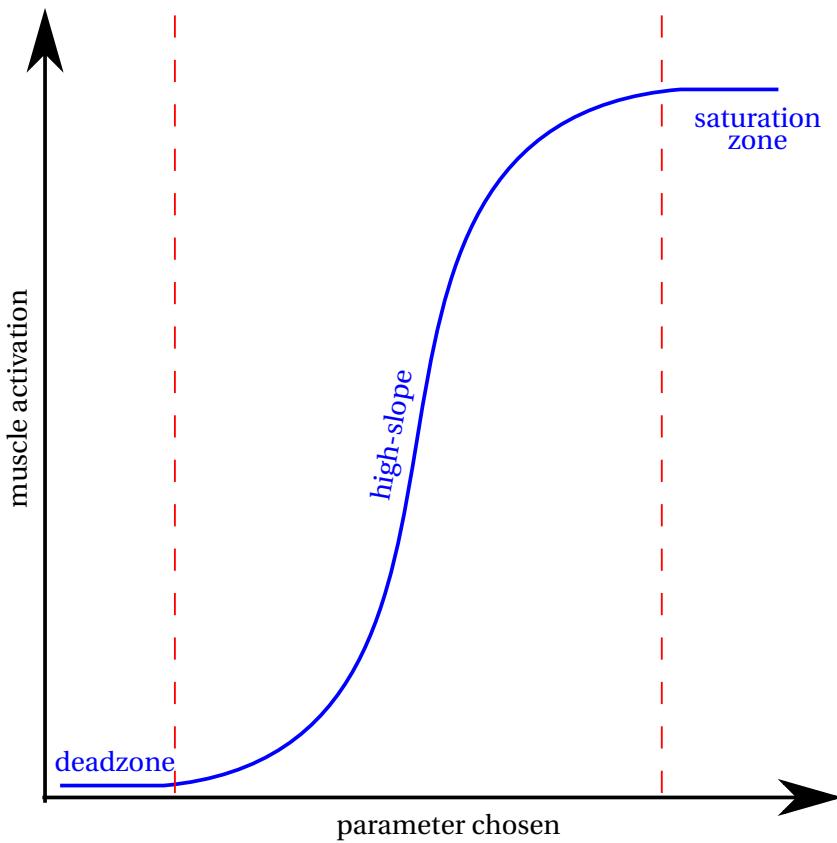


Figure 2.6: Expected effect of generated recruitment curve.

Still when considering recruitment curves, there is an interesting aspect that is worth mentioning related to the difference between physiological activation and artificial stimulation of muscles. During natural activation, the nervous system recruits firstly type I small fibers, recruiting the larger fibers only if more force is needed. However, when dealing with external stimuli, the first motor units to be recruited will be the larger ones, since they have a smaller activation threshold. In terms of recruitment curves, this means that a natural recruitment curve is usually less steep than the one generated by electrical stimulation – which can come in the way of using applications that depend on different values [44, 48, p. 359, p. 159].

Another important aspect that should be considered when stimulating afferent pathways is one that may help deal with different muscle aspects: the *catchlike muscle property*. This property is based on several changes in muscle tetanic contraction characteristics that follow a specific pattern of stimulation, described by an initial burst of two to four high-frequency pulses [49], which has been related to the increase in concentration of sarcoplasmic calcium ions, an important component of muscle contraction [50]. As mentioned, this property has been shown to affect different characteristics, one of which is muscle fatigue itself. It has been shown that it increases substantially resistance against decrease of EMG response after a prolonged time [3], which is mostly explained by the increase of calcium ions, which are closely related to muscle fatigue, as explained before. One more interesting

parameter that is affected by it is muscle potentiation, which are inversely related in matters of muscle strength; in other words, the usage of the catchlike property of the muscle in stimulation results in a smaller overall force during potentiation [21]. Other characteristics have also been related to catch-like muscle property – e.g. fiber type composition of the muscle [51] and muscle contraction [52] – proving its relevance to electrical stimulation.

A lot has been detailed in the electrical stimulation of efferent pathways; however, lots of it can be very useful insofar of explaining the stimulation in afferent pathways. For example, while efferent pathways stimulation is usually modulated by amplitude or pulse-width, one of the most common parameters used in neural encoding for sensations is frequency of stimulation [53, p. 362] Another one of these aspects is analogous to recruitment curves, the named *tuning curves*. They are generated varying specific parameters of stimulation (usually the firing rate) and subjectively relating to the patient's sensations, such as joint angle or pressure [54].

All these aspects that were deeply described will come in very handy when understanding how this work interfaced with subjects from different species and in different ways. A clear discernment of each characteristic of both encoding and decoding were fundamental in the development of the work and, hopefully, it was come across appropriately.

2.3 Control of Arthropods Navigation

Once the basis for understanding neural interface with electrical stimulation of afferent pathways have been settled, studies that rely on these approaches for navigation control of arthropods will be discussed. These studies aim at controlling the direction at which insects and other invertebrates move, either by walking or even flying. In this section, we focus on research involving terrestrial insects, along with the methods proposed and results obtained.

Several studies have been made during the last decades considering neural stimulation of arthropods in order to control movement [55, 7, 56]. Even though our work aims at the implementation of navigation control for insects, most of the works are focused on the development of electronic circuits light enough to be carried by the insect and which affect the overall movement the least possible. As this continuously shows to be a challenge, the navigation control used for arthropods has been mainly restricted to human remote controlling [1], though some researches on control algorithms have been implemented and will be detailed in subsection 2.3.2. In other words, most current researches only make use of navigation control when it comes to assessing the quality of electronic devices proposed. However, a lot can still be learned about how stimulation aimed at motor control has been done.

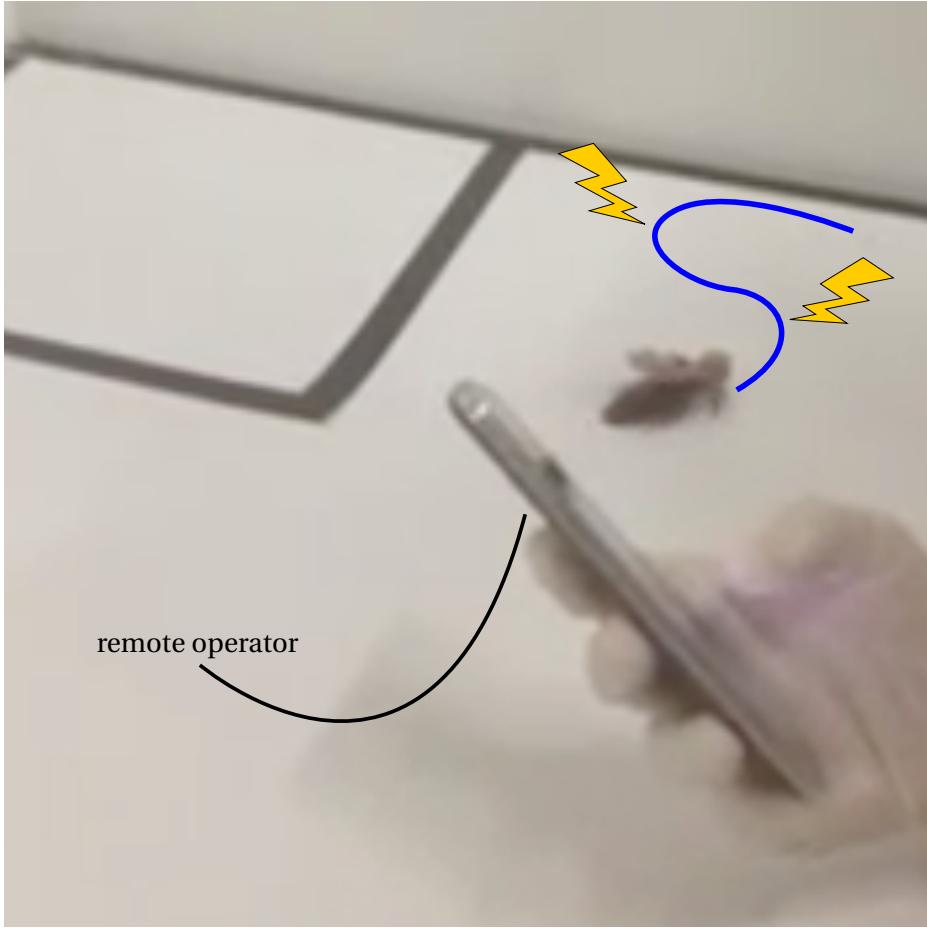


Figure 2.7: Example of remotely operated navigation control in a cockroach. Adapted from Latif *et al* [1].

2.3.1 Principles of Neurophysiology in Insects

Before entering in the navigation control realm, it is interesting to have a deeper understanding on how the nervous system of insects is organized, which is made available in [57] and shall be detailed in sequence.

Insects have simple nervous systems mainly composed by a dorsal *brain* and segmental *ganglia*. Ganglia are groups of interconnected neurons which, generally, represent similar functions. The brain is composed of three main structures: the *protocerebrum*, the *deuterocerebrum* and the *tritocerebrum*. The first one is primarily associated with vision, as it is adjacent to the optic lobes, which is related to the orientation of antennas and biological cycle regulation. The second structure is associated with denser antennal information processing, as olfactory sensing information. The last one, which surrounds the digestory system, is mainly associated with innervation of the labrum (upper lips).

Connected to the brain, there can be found a complex of fused ganglia called *subesophageal ganglion*, which innervate different structures, such as mandibles, maxillae, and labium. In some insects, it also innervates hypopharynx, salivary glands, and neck muscles. The next structures are individual ganglia, which are connected – within each *segment* – by *commisures* and – among different segments

– by intersegmental connectives. Some of the ganglia connected are *thoracic*, which innervate both wings and legs for both afferent and efferent pathways. These are some of the structures primarily involved in locomotion. The *abdominal ganglia*, by the same token, innervate abdominal muscles, which can be used in gait patterns focused on turning the insect around . Finally, the *caudal ganglion* innervate the anus, genitalia and other sensorial receptors present in *cerci*.

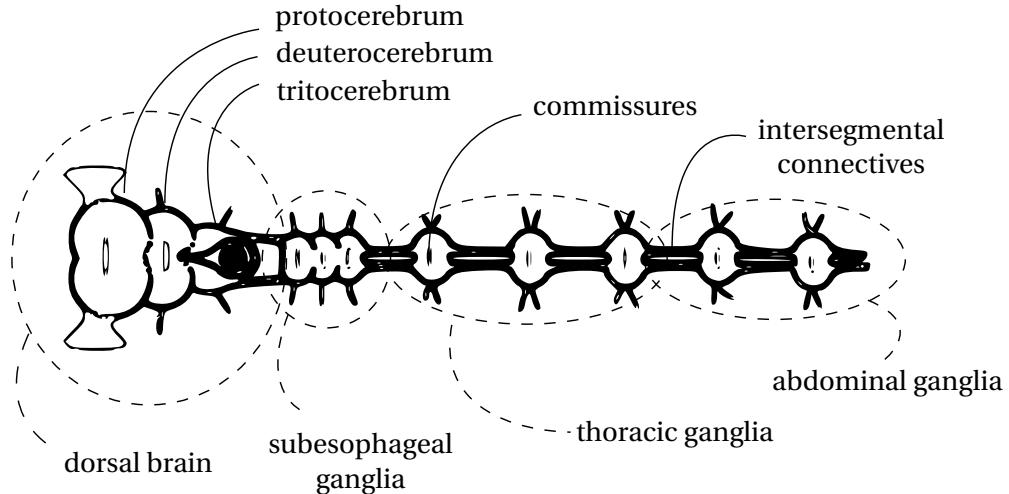


Figure 2.8: Basic neurophysiology of insects.

In case of cockroaches, as the main insect used in our work, the main differences are the fusion of the subesophageal ganglion to the tritocerebrum and the specific nomenclature to the ganglia that innervate the legs: the *prothoracic* and *mesothoracic ganglia*.

2.3.2 Insects Artificial Navigation Using Neural Interfaces

Having understood each role within the nervous system in insects, the strategies used in the last decades are now detailed. Lots of them have been used throughout the years aiming at eliciting stimuli in invertebrates in order to control movement [1, 58]. Some of studies focus on stimulating the insect in specific regions of the Central Nervous System, which evoke coordinated motor activity [59]. These structure, when stimulated, chains a proper sequence of actions that allows the arthropod to execute functionalities such as walking [60]. All the while, several approaches can be followed in order to stimulate these systems, such as the usage of Command Neurons, specific neurons that trigger a whole behavior or part of a motor routine [60, 61]; stimulation of afferent strucutres, e.g. cockroaches hair plates, which set off specific patterns [58]. Another somewhat famous strategy, however, does not focus on the CPG: it consists on taking advantage of the mentioned evasive behavior of some arthropods (more specifically, cockroaches) and stimulate the antennas in order to provoke right and left turns [1, 2, 9].

One of these works placed a cockroach (wearing an electrical stimulation device) on top of an instrumented trackball [2], which allowed the researchers to compare kinematic data based on stimulation parameters. This resulted in valuable lessons about how stimulating the antenna can evoke specific motion and how consistent this motion can be. In fact, these researchers proposed in the end

of the work a line-following setup relying on two light sensors; unfortunately, due to great variance perceived in the motion, the results were not consistent.

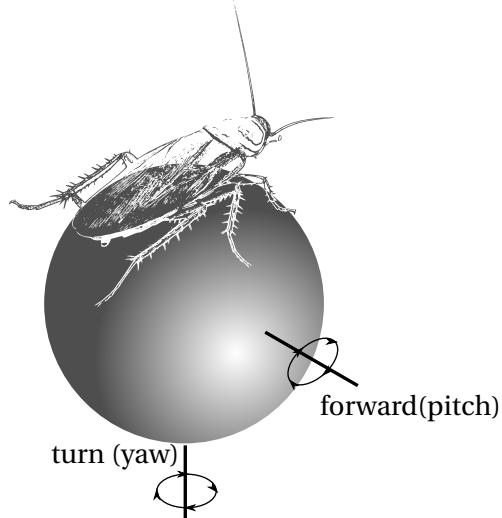


Figure 2.9: Cockroach atop a trackball used for sensing. Adapted from Holzer *et al* [2].

Another work, however, showed to be very efficient in this matter. By using a *Microsoft Kinect* sensor, the system could verify the cockroach's position and orientation in time, allowing very fine control. The control implemented was focused on the angle difference between current orientation and desired orientation towards a waypoint. This way, the control allowed the cockroach to maintain a desired circular reference trajectory [9].

2.4 FES Control Using Evoked EMG

Another application involving the interface with the nervous system is control of movement. As muscle activation is intimately connected to movements, it is the main aspect that should be studied when implementing the control algorithm. In this sense, different variables for measuring the activation may be used, such as generated torque or joint angle; however, the most appropriate this work is evoked EMG. The usage of eEMG is justified by two main reasons: firstly, it provides a cleaner setup, in which sensors are attached in a more reliable way that interferes very little with movements. Another good motive is the assessment of fatigue, that is done more precisely, since eEMG is a direct measurement of muscle activation.

Even though eEMG is the variable chosen in this work, implementations that rely on other variables have been tested, such as in [62], where a *closed-loop proportional and integral* (PI) controller was implemented in order to control output force. This study was successful in dealing with muscle fatigue and potentiation effects by making use of force transducers as sensor inputs and modulating pulse-width in wire electrodes. By surgically dettaching the muscle from the body, the force transducer was connected to it and a wire EMG electrode was inserted into the muscle. The work made use of recruitment modulation, which directly mapped the relation between force and pulse-width.

Using the root locus method, the PI controller was implemented and the results showed it to be efficient in both controlling muscle force and compensating for fatigue effects.

The usage of eEMG recordings in control implementation, however, is the main topic in this section. Some studies have focused on how the variation of specific parameters of stimulation could contribute to reduce fatigue or muscle potentiation. It was found, in [3], that the usage of *catchlike-inducing trains* (CIT) – consisting of an initial high-frequency burst of two to four pulses – increases force response in fatigued and nonfatigued muscles when compared to *constant-frequency trains* (CFT).

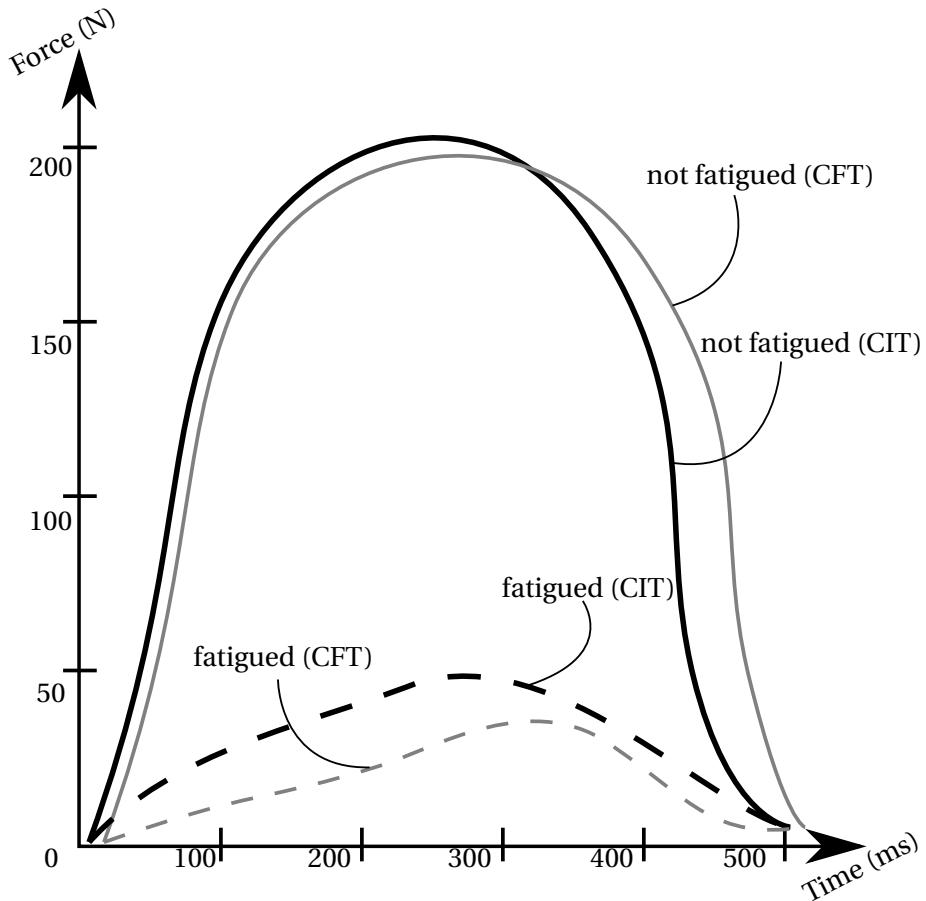


Figure 2.10: Evoked EMG waveforms when stimulated by CIT and CFT. Adapted from Binder-Macleod *et al* [3].

Other approaches aim at controlling eEMG activation itself. Some of which use *Kalman Filter* (KF) for estimating either torque or stimulation level. Truthfully, in [63], a KF-estimator was used for both purposes. The first estimator implemented used evoked EMG recordings and measured torque to find the parameters for modelling activation, which creates a relation between desired torque and measured torque. The second one outputs stimulation level based on past stimulations and eEMG recorded, creating a model that linked desired eEMG and recorded eEMG. This approach resulted in efficient control accuracy with different torque references; in addition, it demonstrated good results as the muscle fatigue became evident.

A similar approach was used in [64]: by making use of a Hammerstein model and applying a Kalman Filter estimator, a predictive control was implemented. Firstly, the estimation of the model parameters is made by using eEMG mean absolute value (MAV), last pulse-width used in stimulation and the output control value acquired by the Hammerstein model as input parameters. Followingly, the sum of the current and last output control values is used in a polynomial equation in order to find the pulse-width. This approach led to results indicating successful control accuracy and stability.

Finally, there have been researches on the use of λ -Controllers for torque control using eEMG and contraction of antagonist muscles (co-contraction). In these works, a reference r_{λ_i} is set as the control input and a linear behavior is forced onto the nonlinear recruitment curve by feedbacking the amount of activated motor neurons, λ . In [65], two λ -Controllers were combined in order to feed models that, based on a transformation between muscle activation and calcium dynamics models (σ_i), estimate accelerating torque (\mathcal{T}) and co-contraction torque (\mathcal{T}_{cc}) when multiplied by a function f_i – that relates measured angle and maximum angle. Both \mathcal{T} and \mathcal{T}_{cc} are then used to model current angle, which is fed back into the loop. As the product between σ_i and f_i is nonlinear, modifying extensions are added in order to linearize the torque inputs. This strategy was simulated and tested in healthy subjects, leading to good results considering accelerating torque and co-contraction strength modulation, as well as joint-angle control.

Chapter 3

Insect Direction Control Using Neural Interfaces

The method of science is tried and true. It is not perfect, it's just the best we have. And to abandon it, with its skeptical protocols, is the pathway to a dark age. — Carl Sagan

The following work was inspired by previous researches using BackyardBrains's project named *The RoboRoach*. The goal is to be able to actively control the direction of locomotion of a cockroach using electrical stimulation of the antennal nerves. As explained before in this work, cockroaches use their antennas to perceive the presence of possible predators, sending signals to their thoracic ganglia [66] in order to react and avoid the menace.

The first half of the work presented in this document was developed at the Automation and Control Laboratory (LARA) of the University of Brasilia, being focused on electrical stimulation of cockroach's antennas in order to implement direction control strategies. In Section 3.1, the methods used for hardware and software development are described in detail, along with the experimental procedure and control implementation. Section 3.2 presents data acquired during the experiments run. Finally, in Section 3.3, the results presented are discussed.

3.1 Method

A small stimulator appended to the cockroach head, which connects one stimulating electrode to each of the insect antennas and a reference electrode to its exoskeleton back was used. The stimulator is based on *BackyardBrains's RoboRoach development kit*, which will be further detailed in Subsection 3.1.1. In order to control its navigation, a scheme that can provide the cockroach's position is designed using the motion capture *NDI's Polaris Spectra* system and passive retro-reflective markers. The whole setup, thus, communicates with a central personal computer (PC) – running Ubuntu 14.04 Operating System – through Bluetooth (BT) and Universal Serial Bus (USB) serial protocol (respectively). The PC runs the navigation control algorithm implemented. In sequence, each of these modules will be carefully elaborated in order to clarify doubts on limitations and possibilities supported by the tools.

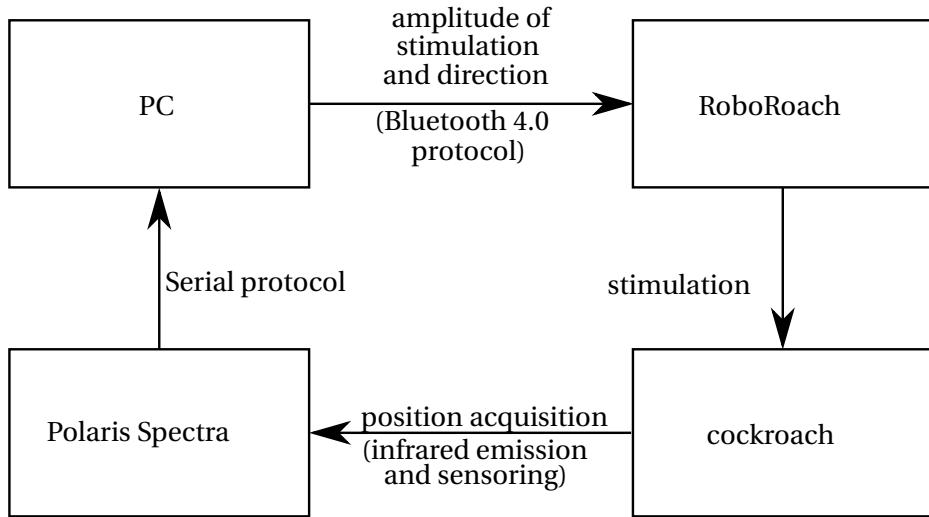


Figure 3.1: Block diagram describing the methods utilized.

3.1.1 RoboRoach Development Kit

"The RoboRoach" is the name given to a system envisioned to be used as a starting point for Neural Engineering enthusiasts and researchers. The system consists of an electronic board small enough not to significantly disturb the cockroach's movements; a three-pin connector, where the system is actually fixated to the insect's exoskeleton; wire electrodes used for reference and stimulation; and a 3V lithium battery. The whole system weight averages 4.4 grams [67].

The electronic board itself does not involve complex hardware implementation, as it is not meant to – due to cost and overall weight reduction purposes. It consists of a simple microcontroller connected, using Serial Peripheral Interface (SPI) protocol, to a BlueRadio BR-LE4.0-S2A low-energy BT module, which implements BT 4.0 communication protocol. There are four Light-Emitting Diodes (LEDs) connected to BlueRadio: two serving as communication status LEDs and one for each stimulation electrodes, meaning to be used for debugging purposes [68].

The stimulation provided by the board is voltage-based and allows the user to change some variables, such as pulse-width, amplitude, frequency and number of pulses. The waveform presented, although rectangular, as usually recommended, is only monopolar (as opposed to bipolar, as explained in Subsection 2.2.2). For each settable parameter, the range that could be found from documentation is from 0 to 255, as contained in one byte.

BackyardBrains also provides all the code used for developing the mobile applications that communicate with the board. The code is documented in their repository [69], which eases the Application Program Interface (API) development procedure soon-to-be explained.

3.1.2 Polaris Spectra

The system for sensing position is NDI Digital's Polaris Spectra, a somewhat portable optical tracking system, which consists of a *Position Sensor*, weighing less than 2kg with dimensions between

613mm of length per 104mm of width and 86mm of height, connected to the *System Control Unit* (SCU) [70]. It provides the user with real-time (up to 60Hz) position and orientation (depending on the tool used) information within a pyramidal volume with dimensions described in Figure 3.2 [71].

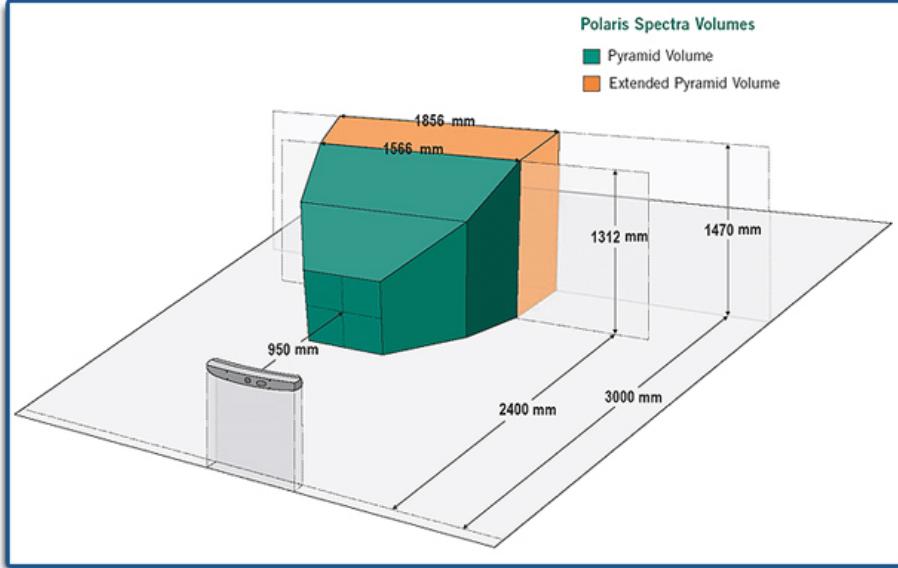


Figure 3.2: The volume within which positions are acquired. Source: NDI Digital.

The Position Sensor consists of an infrared emitter and the infrared sensor itself. A positioning laser can be used in order to align the sensor with the area of interest. It works with two different types of tools: the *active* and the *passive markers*. Active markers, as the name suggests, are powered and emit infrared radiation back to the sensor. This provides better precision while increasing bulkiness of the tool. In order to counteract this disadvantage, passive markers, which consist of spherically-shaped objects that reflect infrared radiation, may be used. The passive markers are divided in two main types: *geometrical tools* and *stray markers*. Geometrical tools must contain at least three individual spherical markers and be arranged in a specific shape, which should be defined within NDI's Guided User Interface (GUI). This allows the system to calculate the orientation of the tool based on an initial setup. The other type, stray markers, are merely the spherical objects themselves, which do not provide information on orientation. The main advantage is the absence of the support that fixates the geometrical shape in geometrical tools, which decreases the tool overall weight and inertia [70].

NDI provides the user with different softwares that can be used in order to configure and test the sensors and the tools. It provides different interfaces with the sensors, from real-time acquisition of the sensor informations to the final calculation of position and orientation of the tools. A great feature of the GUI is the possibility of "listening" to all the serial communication that happens between the SCU and the PC, allowing the user to develop third-party programs using these examples along with the API user's guide provided by the development team.

3.1.3 Developing our own API

In order to use both sensor and actuator, represented by the Polaris Spectra and the RoboRoach, the PC must be able to communicate with them both. Different programming languages were tested for this purpose; however, due to implementation limitations when interfacing with RoboRoach, *Python 3.0* was chosen, which is a fast and programmer-friendly language with great support from the online community. An API for each of the systems was developed and saved in a BitBucket public repository [72] – and will be detailed in sequence.

3.1.3.1 RoboRoach Bluetooth Module API

As just said in Section 3.1.3, the RoboRoach represented the greatest limitation in terms of hardware and software needed for communication. This happened due to the BT module used, a Bluetooth 4.0 low-energy module, which follows a specific and new communication protocol not yet implemented in most languages. The justification for this hardware design lies on its high power efficiency (as it is said to implement an idle mode for the time it is not being used) and easeness of implementation (being selected as the new protocol used in Internet of Things applications) [73]. Thus, it was necessary to develop the code in a rather specific environment: an Ubuntu 14.04 Operating System using a BT 4.0 USB dongle.

The BT 4.0 LE communication protocol is based on Generic Attribute Profile (GATT), a communication profile that divides each feature of the device in a tree-shaped hierarchy. The first level of the tree displays all *services* that are available for interfacing: in this case, the battery service and RoboRoach service, the latter being the one that was dealt with. Each service, then, consists of several *characteristics*. These characteristics may describe different parameters with which the programmer can interface; some of them are configuration parameters, such as stimulation pulse-width, amplitude and frequency, whereas others are acting variables – left and right electrode stimulation. Each characteristic is related to a *parameter* field (still by protocol), which is defined in order to finish the interface. When dealing with configuration parameters, the user must insert the value wanted for the experiment; when stimulating, a simple binary activation may be set to indicate the instant of stimulation. The code repository of BackyardBrains may be helpful when understanding this layout of the protocol [69]. In order to set up the communication and discover the different addresses that were implemented, along with the parameters range values that could be used, different Unix tools were used, such as *hcitool*, *hciconfig* and *gatttool*. Details on the initial procedure for communication can be found in [74].

Having in hands the whole addresses information, a high-level API was developed in order to facilitate the programmer's implementation using simple commands – from configuration settings, such as "change amplitude", to electrode activation, such as "turn right".

3.1.3.2 Polaris Spectra USB Interface API

Before implementing any code, a few steps had to be followed in order to be able to interface with the SCU by the software provided in a CD-ROM by NDI Digital. The installation happened mostly through Ubuntu's Terminal tool, sometimes dealing with hardware connection itself. The procedure is also detailed in the repository [72], under "Polaris" directory.

In terms of API development, it is interesting to note that Polaris Spectra's SCU implements a very straightforward USB serial communication with the computer. As mentioned before, all communication with the proprietary softwares available can be read through specific tools within the GUI, allowing the user to implement the commands by themselves. Even though NDI Digital makes available a thorough API user's guide with detailed information on each function that can be used, the initial communication setup used for stray markers is not entirely explained, which led the implementation to different paths.

In order to design this initial setup, thus, the whole conversation between SCU and PC was mimicked using Python's serial library. Each low-level function described in the API user's guide was accordingly implemented and used in a high-level fashion, implementing two simple functions: one that initialized the system and the communication between both ends, while setting stray markers options, and one that plainly requested from the SCU the last position captured by the sensor for a single stray marker. The code used was based on previous implementations, available in a public repository [75].

3.1.4 Cockroach Preparations

In order to perform artificial control trials, the cockroach must to be prepared for experimentation. All steps were based on BackyardBrains's experimental protocol detailed on their website [76], which had to undergo a few alterations in order to provide better results. The insect species chosen was *Blaberus giganteus*, due to its large size and strength and, also, availability – differently from the famous species that is usually chosen, the *Periplaneta americana*. All surgeries were performed at LARA, with appropriate light and making use of simple small manipulation equipment, such as scissors, tweezers, toothpicks and cotton swabs. Even though there is no specific regulation for experimentation in arthropods [77], a few precautions were taken, such as surgeons and auxiliary staff wearing rubber gloves during the whole procedure and the intercalation of every step of the operation with enough anesthesia and recovery time.

The procedure can be summarized by an initial anesthesia of the insect using ice cubes and cold water [78], followed by the sanding and fixation of the electrode connectors onto the cockroach exoskeleton head using Super Bonder glue. Followingly, a reference electrode is placed underneath the cockroach's wings, being fixated also with super glue. The two stimulating electrodes are then inserted into the residual antenna (which must be previously trimmed in order to enable the insertion). One of the improvements proposed over process efficiency is the usage of a drilling tool in order to increase precision and fixation of reference electrode. Also, an electric sanding tool was used so that the waxy chitin would become more rougher, helping with the connector fixation. The last improve-

ment that was implemented was the fixation of wings and cockroach head in relation to the body, in order to decrease its degrees of motion (allowing a better controllability based on antennal position) and enhance electronic system fixation.

After the procedure, the insect was placed back on its cage for several hours (usually overnight), giving it sufficient time for recovery from surgical procedure and anesthesia.

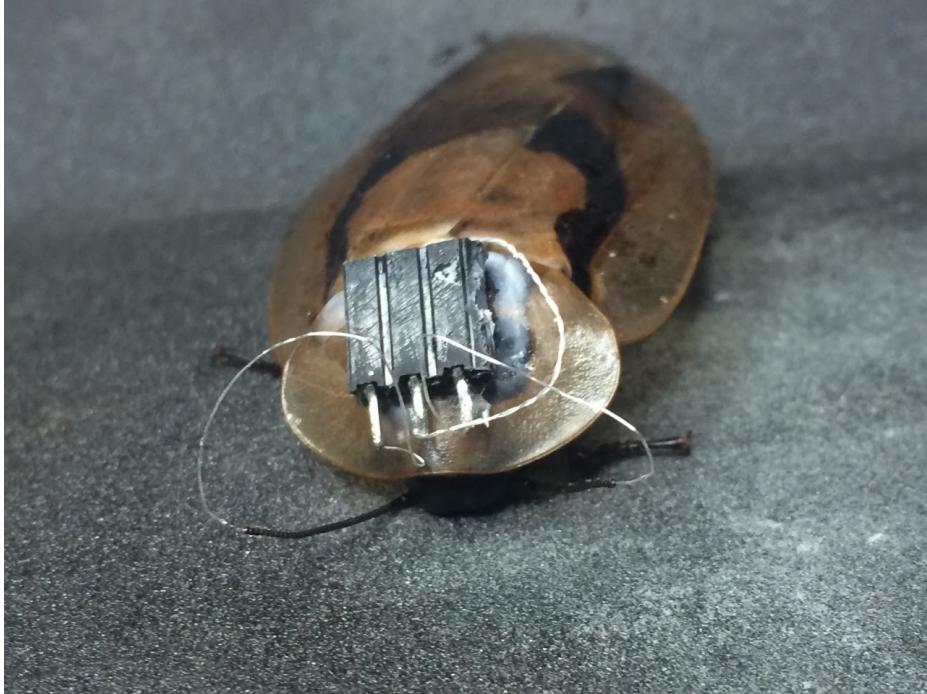


Figure 3.3: Final result from electrode implantation and board fixation procedure.

3.1.5 The Experimental Setup

Having completed the hardware, software and cockroach preparations, the experimental organization that is going to be used during the tests is arranged. For this purpose, the Polaris Spectra is placed on a tripod standing atop a table. The tripod is regulated so the sensor points downwards facing the floor, on which the insect walks. The calibration software provided by NDI Digital is executed so that a path can be traced based on the change of position in only one of the axis. As the cockroach does not move upwards nor downwards, having assured the floor is not crooked in relation to the sensor, it is made sure the cockroach does not move in another axis as well (as represented in the example of Algorithm 3.1, the x-axis).

In order for the tracking to take place appropriately, a marker is attached to the RoboRoach board by gluing it together with a screw, which is inserted in the stray marker, using hot-melt glue. This allows the board to be perceived by the sensor as long as it is within the path length. Also, the path is drawn making sure it is within BT dongle reach while it is following the desired trajectory. The experimental setup can be better pictured in Figure 3.4.



Figure 3.4: Arrangement used for the experiments. Polaris Spectra is positioned facing the floor, where a line indicates the desired path, namely, the reference path.

3.1.6 Simulations and Experiments

Two experimental methods were designed: a simulation, used for verifying the control efficacy, and the experiments using living cockroaches implanted as described before in this work. Followingly, in Subsection 3.1.6.1, the control algorithm implemented is described. In Subsections 3.1.6.2 and 3.1.6.3, these experiments are described.

3.1.6.1 Direction Control Algorithm

As both the sensor and actuators of the control system are complete, it is important to finally add the direction control algorithm itself. In this sense, a plain line follower algorithm control was implemented using a proportional gain that modulated amplitude. The control algorithm implemented is based on the fixation of one of the movement axis (say, for this example, x-axis) around which maximum and minimum limits were imposed. In case the cockroach left these limits, a command would be issued and the antennal stimulation would force the insect to remain within limits. A proportional gain was added according to how much the cockroach overstepped the boundaries; this gain, thus, modulated the amplitude that stimulated the antennas, resulting in a more eccentric turn, as shown in Equations 3.1.

$$\begin{cases} \text{amplitude} = k_p(\text{position.x}_{\text{recorded}} - \text{position.x}_{\text{max}}) \\ \text{amplitude} = k_p(\text{position.x}_{\text{recorded}} - \text{position.x}_{\text{min}}) \end{cases} \quad (3.1)$$

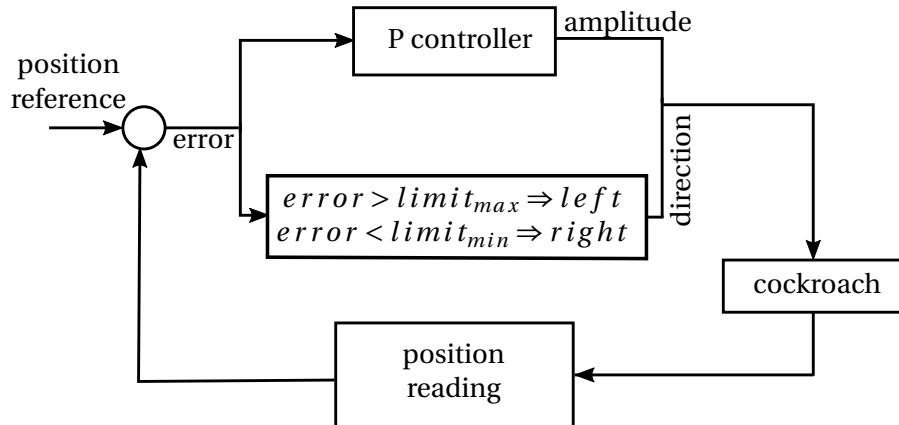


Figure 3.5: Control algorithm block diagram.

where k_p refers to the proportional constant of the control; $position.x_{max}$ and $position.x_{min}$ refer to the maximum and minimum values in the x-axis. The pseudocode for the proposed initial algorithm can be seen in Algorithm 3.1.

Algorithm 3.1 Pseudocode of the initial control algorithm experimented.

```
1: while stop character is not pressed do
2:   get position
3:   if position.x is more than max_limit then
4:     set right_stimulation_amplitude  $\leftarrow K_P * (\text{position.x} - \text{max\_limit})$ 
5:     stimulate(left)
6:   else if position.x is less than min_limit then
7:     set left_stimulation_amplitude  $\leftarrow K_P * (\text{position.x} - \text{min\_limit})$ 
8:     stimulate(right)
9:   end if
10:  end while
```

3.1.6.2 A Virtual Cockroach

As a proof of concept, the navigation control algorithm is tested using what is titled a *virtual cockroach*. The virtual cockroach consists of a human operator holding the board in a way the sensor can capture the stray marker position and simulating the movement of a somewhat well-behaved cockroach. Using the auxiliary status LEDs, which indicate which antenna is being stimulated, the operator can reproduce the activity of a cockroach.

3.1.6.3 A Real Cockroach

The cockroach is initially removed from its cage and placed in a recipient with ice for little time (around 3 minutes), so its metabolism slows down before the experiments, allowing the system to respond more properly. Simple experiments are made in order to test the stimulation response for both

antennas, consisting of stimulating the electrode through the command line in the PC and observing the appropriate result.

Followingly, the cockroach is placed in the beginning of the path desired (it is important to delimitate the beginning and the end, since the algorithm assumed both due to the lack of orientation information). Video of the experiment is recorded along with position information. Several trials are performed for each cockroach, allowing enough time between procedures in order to avoid familiarization with the stimulation signal, which may result in worse responses. The whole procedure is repeated for three more cockroaches. For each, different parameters are tested for pulse-width, frequency of stimulation and maximum amplitude. The values are shown in Table 3.1.

Frequency (Hz)	Pulse-width (ms)	Maximum amplitude (mV)
250	200	750

(a) Experiment 1.

Frequency (Hz)	Pulse-width (ms)	Maximum amplitude (mV)
150	5	750

(b) Experiment 2.

Frequency (Hz)	Pulse-width (ms)	Maximum amplitude (mV)
5	200	750

(c) Experiment 3.

Frequency (Hz)	Pulse-width (ms)	Maximum amplitude (mV)
250	4	750

(d) Experiment 4.

Table 3.1: Values used during experiments with cockroaches.

3.2 Results

As the experiments advanced, different plots are generated, which are related to both efficiency of navigation control algorithm and quality of signal generated by the board used. Firstly, we observe how output tension is delivered by the electrode pins from the board change when varying number of pulses and setting pulse-width at 1 ms .

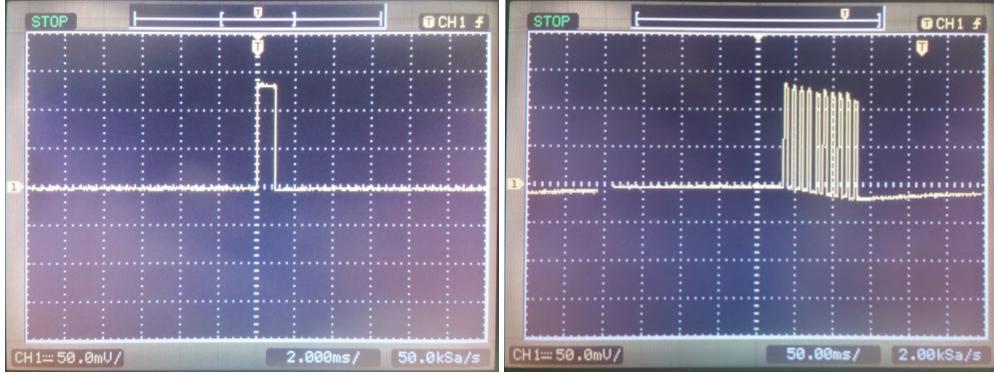


Figure 3.6: Voltage output measured from board with 1 and 10 pulses.

Followingly, how it changes when different pulse-widths are set to the board as the number of pulses is constant at 1.

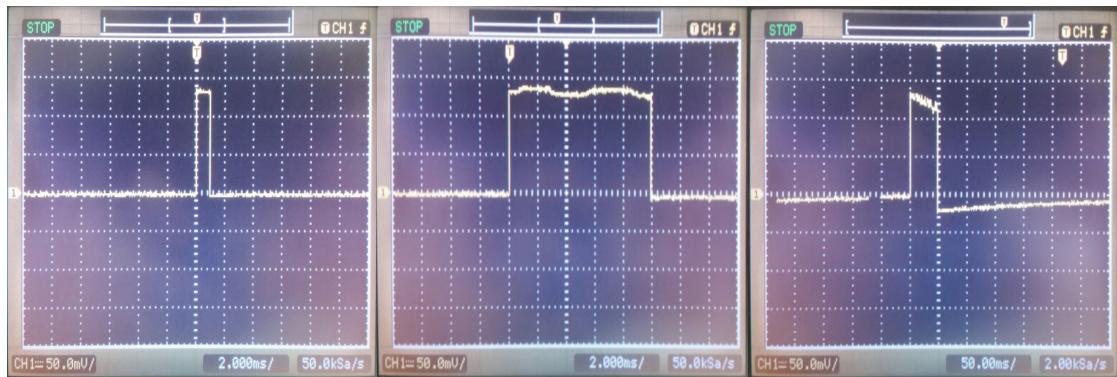
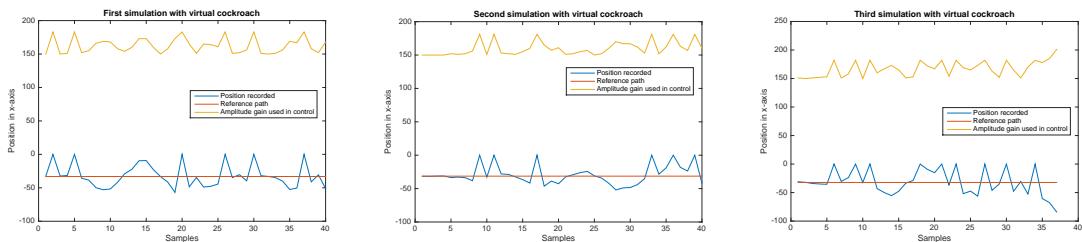


Figure 3.7: Voltage output measured from board with pulse-width of 1ms, 10ms and 50ms.

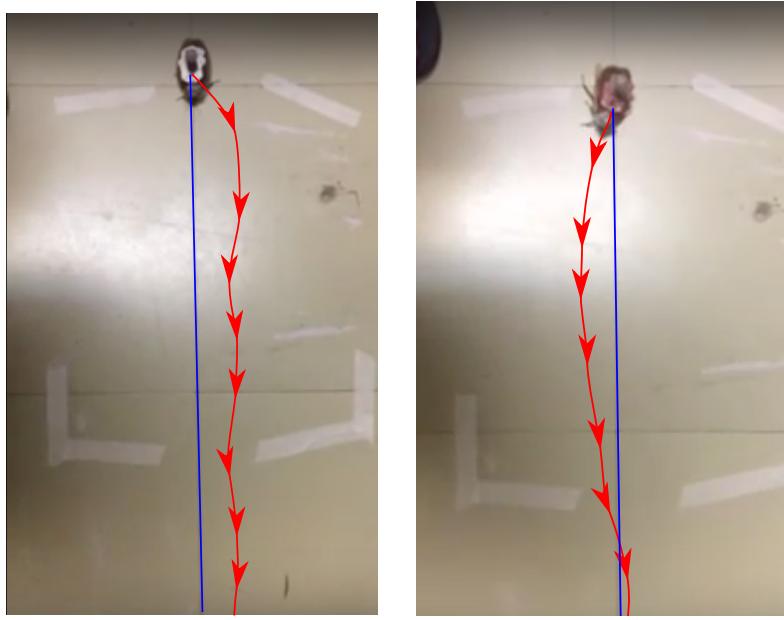
The Figures below provides us with information on position during virtual cockroach experiments.



(a) First simulation with virtual cockroach. (b) Second simulation with virtual cockroach. (c) Third simulation with virtual cockroach.

Figure 3.8: Simulations using the virtual robocockroach setup.

Finally, it is shown to us how the real cockroach wandered around the desired path during experiments.



(a) First experiment with real cockroach.
(b) Second experiment with real cockroach.

Figure 3.9: Experiments using the real roboroach setup. In blue, the desired path the cockroach should follow. In red, the actual path traveled.

3.3 Discussion

There is a lot to be analyzed from both plots and parameters used in the experiments with RoboRoach. Followingly, we will discuss what the results acquired tell us about the performance of the system designed and how it can be improved in future works.

The first point to be analyzed is the output voltage the board describes. As seen in Figures 3.6 and 3.7, the hardware used have certain limitations when it comes to using specific stimulation parameters. For instance, Figure 3.6 shows the offset decay in voltage as the number of pulses parameter is increased. Likewise, Figure 3.7 presents a similar behavior, as the offset decreases when increasing pulse-width values. Both examples provided are enough to illustrate the voltage limitations of the board, which come in the way when trying to modulate parameters such as pulse-width with the control output. For this reason, amplitude modulation was chosen for the experiments.

Moving on from the stimulation to the implemented control itself, it can be observed in Figures 3.8a, 3.8b and 3.8c the navigation control strategy implemented behaved satisfactorily in terms of direction and actuation intensity. This experiment, although not much consistent when considering the subjective character of the operators, provides enough information to validate the strategy implemented, as the experimentation in the cockroach shows to be highly complex.

As for the experiments with real cockroaches, much more information can be extracted. For most of the experiments, as seen in Figures 3.9a and 3.9b, the cockroach presented a behavior far from ideal, most times not responding appropriately to the stimuli. When other stimulation parameters

were implemented, such as Tables 3.1b and 3.1c, the response to stimuli was much more consistent.

Still, whichever parameters chosen, it was observed a high rate of familiarization with the stimulation by the insect, meaning it would react much better to stimuli during the first experiments than during the latter, regardless of the great period between experiments dedicated to avoiding it. One point that was not considered during the experiments is the actual frequency of stimulation. Even though the frequency was set by software to be an specific parameter used by the board during initialization fase of the code, the code implementation did not consider that requesting a new stimulation for the board from the main routine every loop interaction would overwrite the last behavior and restart the stimulation. Since the routine was called in a frequency close to $100Hz$, the stimulation could have been quickly familiarized by the cockroach due to it.

Another issue encountered during the experiments was the lack of an ideal setup. One of the problems was the length of the path: since the reference traced for the experiments was somewhat short in comparison to the distance the cockroach could achieve in a brief period of time, there was little data acquired from successful experiments. Another problem with the setup is the amount of distractions that could drive the cockroach: since it was done in the floor with lots of furniture around it, the insect had several incentives to go to specific ways, such as behind computer racks or towards the wall.

Chapter 4

FES Control Using Evoked EMG

I am very conscious that there is no scientific explanation for the fact that we are conscious. —Andrew Huxley

The following methods refer to experiments held at the Center for Neural Engineering (CNE) of the University of Utah¹. For these experiments, the aim was to study control strategies that could be used with a specific setup in order to control muscle activation from nerve electrical stimulation. In order to do so, a real-time system with a GUI is required having in mind certain features: proper electrical stimulation (along with different parameters modulation, such as amplitude, pulse-width and frequency); one-channel EMG recording (with a high enough sampling rate); a short and consistent loop period (which could vary from 10ms to 20ms); analog interface with a force sensor in order to be able to validate and study the relation between EMG activation and force; data visualization from analog inputs, electrode reading and stipulated reference; control algorithm implementation.

The initial experiments took place using *frogs*, which served as a great starting point for experimental setup testing and troubleshooting. Later on, the experiments used *rats*, on which other experimental setup testings were executed and, finally, control implementation was tested.

In Section 4.1, the development of the real-time system, control implementation, experimental setup and procedures are described. Following, in Section 4.2, results acquired from the experiments are shown. At last, in Section 4.3, the results are explained and discussed.

4.1 Method

The experimental setup makes use of an electrical stimulator named Grapevine, from Ripple LLC, which is detailed in 4.1.1. The Grapevine system communicates with a personal computer (PC) using *User Datagram Protocol* (UDP), where the control algorithm is executed. The Grapevine makes use of front-ends for both stimulation of the nerve and recording of muscle signals. The control algorithm is implemented using a proportional integral controller in order to allow muscle activation to follow a trapezoidal reference previously set.

¹Halfway through the development of the work, the opportunity to upgrade it and finish it at another laboratory was presented.

The code development has a common beginning that should be described for both sets of experimentations. Later on, specific enhancements in each situation are made in the code and hardware so that overall performance is increased. All routines were implemented using *Matlab R2014b* along with Grapevine's API named *Xippmex*.

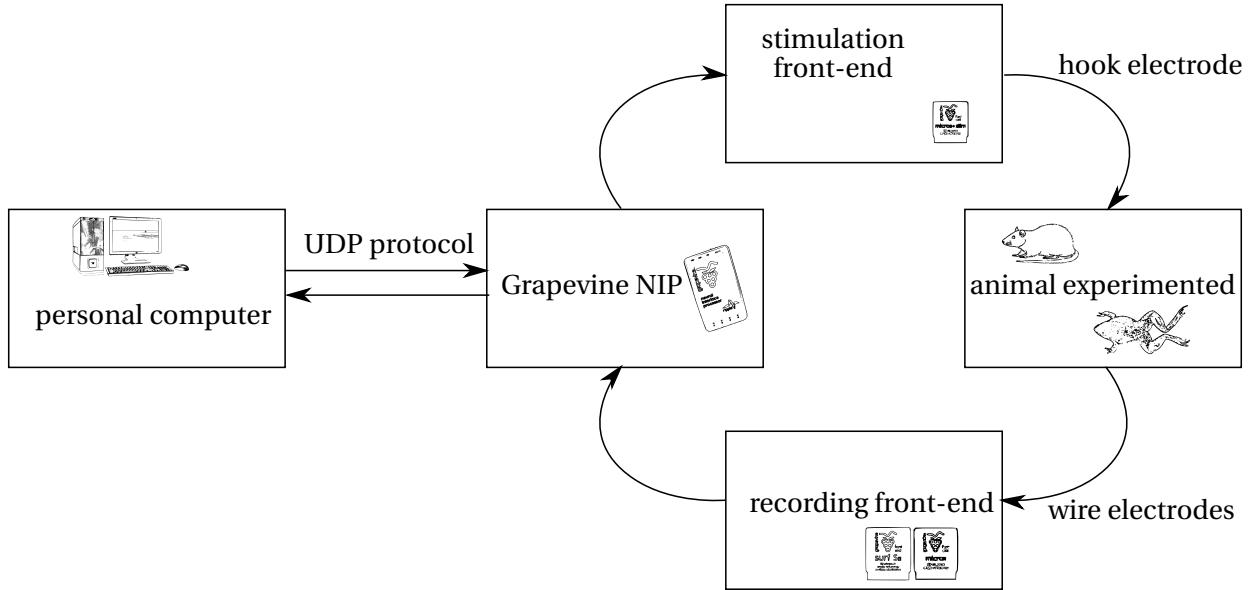


Figure 4.1: Block diagram of the simplified experimental setup.

4.1.1 Grapevine

Grapevine is a hardware system from Ripple LLC used for acquisition of neural or muscular signals and electrical stimulation. Its central computational unit, the largest module in the system, is called *Neural Interface Processor* (NIP). At one side, NIP contains two USB-ports (named "expansion ports"), one auxiliary port, one ethernet port, the power socket and power button [41, p. 8-9]. At the opposing side, it provides four ports that interface with "front ends" (FEs). There are several types of FEs, which are meant to be used for specific applications, e.g., the *EMG front end*, with 16 pairs of differential inputs, ideal for EMG recording at high resolution; the *Analog I/O FE*, which interfaces with analogic sensors and actuators at high sampling rates ($1kS/s$ and $30kS/s$), providing the user with different applications allied to neural acquisition; and, last but not least, the *Micro+Stim FE*, a 32-channel unit made for both recording and current-based stimulation of neural interfaces. All FEs acquire data at high rates and filter out frequencies of no interest, in addition to converting the analogic data to digital in order to decrease noise that could be acquired during the transmission to the NIP [41, p. 13].

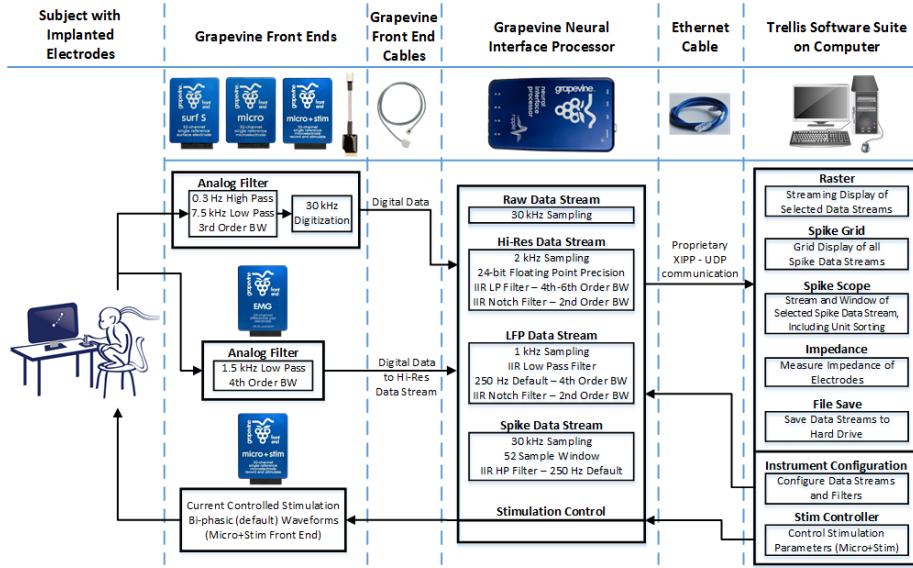


Figure 4.2: Grapevine's NIP and front-ends. Source: Ripple.

For the experiments here presented, the front end used for both recording of muscle signals and electrical stimulation is Micro+Stim. Another FE named *Micro*, which differentiates from Micro+Stim for not possessing stimulating capabilities, is also used in early stages. From Micro+Stim/Micro, the Neural Interface Processor processes the digital data provided and offers the user different acquisition options, such as:

- "raw" – raw signal – a 30kS/s signal discretized in 16 bits;
- "hi-res" – high resolution – same signal sampled at 1kS/s and at a higher 24-bit voltage resolution;
- "lfp" – local field potential – continuous waveform data sampled at 1kS/s, appropriate for electroencephalograms (EEG) and electrocorticograms (ECoG); and
- "spike" – spike waveforms – which returns waveforms from action potentials, found by voltage threshold crossing [41, p. 29].

The procedures require the usage of high-resolution acquisition, since it contains custom filters appropriate for the application at hand. Another important feature of Grapevine and Micro+Stim is the activation of *fast settle triggers*, which broadcast a message to the other front ends at the moment of stimulation so they will stop recording and avoid stimulation artifacts.

Once the digital signal is once again filtered, this time by the NIP, it is sent to the ethernet port in order to interface with the computer. One way to both acquire information from the NIP and also send information about the electrical pulse it should generate is through the proprietary software suite *Trellis*. It provides the user with initial configuration settings for the experiments, along with different other softwares used to visualize data, check electrode impedance and shape the desired waveform for stimulation. For applications that do not involve custom online modulation between

recording and stimulation, Trellis contains all the appropriate softwares. However, since the goal is to implement online real-time control, one must be able to access these features programmatically; hence, Xippmex API should be used.

4.1.2 Xippmex

Xippmex is thus titled based on two main factors: firstly, the protocol implemented between computer and Grapevine, named *Extensible Instrument Processing Protocol* (XIPP); secondly, the MEX format, specific for MATLAB files that invoke C, C++ or Fortran functions. Xippmex allows you to access most of the tools implemented on Trellis through MATLAB scripts [79, p. 3]. Two of the functions of Xippmex, however, deserve special attention: the *cont* function and the *stimseq* function, as they are more frequently used in the code implemented. "Cont" function allows the user to gather continuous data from whichever front end available for streaming. Since Grapevine continuously acquires and stores data within a five-second buffer at specific rates aforementioned, "cont" receives as a parameter the duration that represents from how far back does the user wants to acquire data (in milliseconds). Moreover, other parameters include the identification number for the electrode used, the stream type (as earlier mentioned, "raw", "lfp", "spike", "hi-res") and the timestamp from which the data is delivered [79, p. 11].

Another function that should be contemplated, "stimseq", is used for stimulating the specified electrode using a certain sequence of events. This sequence is determined during code initialization, but can be altered at any time. The parameters that can be set for each electrode are frequency, number of pulses, time at which first stimulus is generated, and the sequence itself. The sequence works as an array of events, each one having as parameters amplitude, polarization, stimulation enabling, fast settle activation, delay and recording enabling. This way, a quadratic pulse of different shapes can be generated, allowing the programmer to have much more flexibility during experiments [79, p. 16-19].

4.1.3 Developing the Real-time System

Before developing the GUI necessary for control implementation and data visualization, a few matters had to be discussed. For example, since the goal of the project was to be able to use a control algorithm in real-time, different approaches for its implementations were studied. In this sense, it was studied the preference between the usage of *xPC Target*, *Real-Time Workshop* or regular MATLAB run in real-time in a *Windows 7* environment. xPC Target relies on a two-computer setup: the host computer, where the main user interface is present, and the target computer, running a real-time operating system and interfacing with the hardware itself. Despite the disadvantage already presented, namely, the need for two computers, specially one running in real-time, xPC Target provides real-time capabilities, with reliable clocks and high performance [80, p. 1-2]. The one-computer-setup derivation of it is called Real-Time Workshop, which runs with both MATLAB and Simulink in a Windows environment. It provides a better performance and ease of implementation [81, p. 1.3]; however, a license is necessary and it was not available at the moment. Hence, it was decided that the GUI could be developed using regular MATLAB functions and scripts, as it has been done before with 60Hz closed-

loop control [82, p. 40-44]. For MATLAB to run in real-time, however, a Process Explorer program was downloaded [83] and used to change the software priority.

Aiming at providing a consistent real-time interval for both the control function (consisting of both acquisition, control algorithm, parameter modulation and stimulation) and the data visualization/logging function, an initial strategy was used based on MATLAB's Timer Objects. These types of MATLAB variables consist on timers to which one can set the period it sets off, the function called whenever the timer reaches zero and other parameters, such as how the queue of events is dealt with. There is, however, a disadvantage to it pointed out by Mathworks Technical Team: when interfacing with external hardwares, the timer may be held for a specific period, which may affect the consistency of the setting-off period. Thus, it was decided to follow a different path, which was based on the serial calling of each function within the period and the implementation of a high-precision delay routine in MATLAB using the tic-toc functions.

Finally, as the code was being developed, it came to light that visualization could be an issue when dealing with real-time. As it was being implemented, the plotting function was demanding a high computational cost, which in several times disrupted the time period consistency. Once concluded that this implementation would not be proper for real-time, it was suggested to use another machine for dealing with plotting, i.e., the host machine would make all calculations for control and deliver the data to the client machine through UDP. For this goal, Java-based functions that dealt with UDP communication were implemented (since the appropriate MATLAB's Toolbox was not available). A crosscable ethernet cable was fabricated, thus, for linking both computers.

In the end, the final GUI was created for control implementation and data visualization/logging. As subproducts of the process, other scripts were also generated focused at different experiments. The first script used basically acquires evoked EMG waves while stimulating in a custom frequency. The second code developed aimed at creating recruitment curves for the animal by varying amplitude, then pulse-width and, finally, frequency of stimulation. Another script was coded in order to modulate amplitude, pulse-width and frequency based on analog readings, such as a force sensor. These scripts were made to serve as preparation before the experiments involving the control algorithm itself. In the final GUI code, other functionalities were implemented, such as the data communication and catchlike inducing patterns. Of course, another script was developed to be used in parallel with the GUI in the client computer, which acquired the data from the UDP channel and plotted it in real-time.

4.1.4 Control Algorithm

The control implemented makes use of the mean absolute value of amplitude acquired in the last time window. This value is compared with a trapezoidal reference, generating an error which acts as the input value for the control function. The control function outputs a normalized value, which is added to the desired reference, and used in the inverse recruitment curve function. Based on the amount of contraction calculated by the control, an amplitude value is found and used in the stimulation, as described in Figure 4.3.

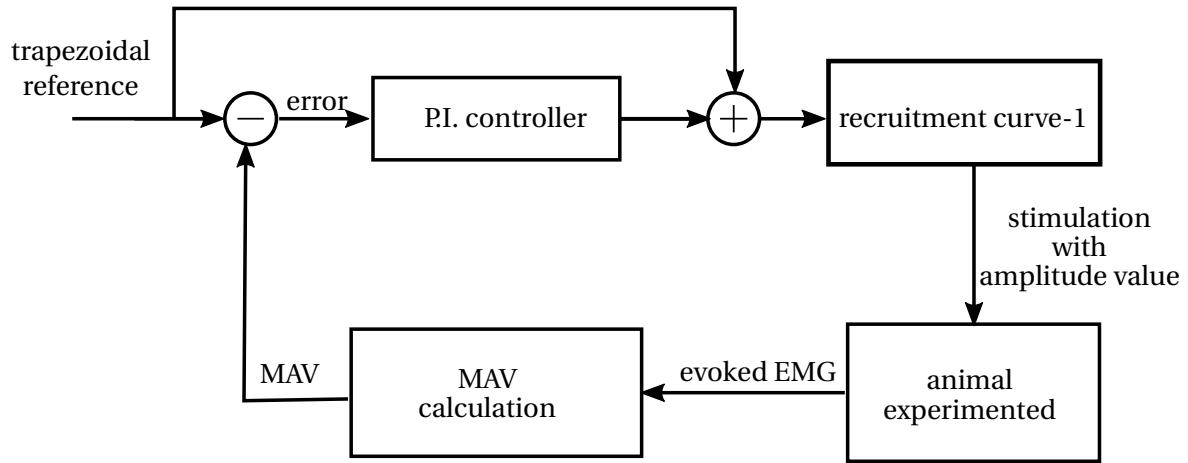


Figure 4.3: Block diagram of the control system implemented.

The recruitment curve, hence, plays an important role in the experimental procedure and its code must be detailed further before thoroughly describing the control function itself. The script for recruitment curve generation runs in three separate stages: the first part consists on varying amplitude of the stimulation (while maintaining pulse-width and frequency) and recording the eEMG signals. Each stimulation lasts for 500ms and there is a delay between stimulations of 2.0s for muscle recovery. The recorded data is saved as well as the mean absolute eEMG values elicited by each stimulation train, forming the second stage. For the third one, the data is used in offline analysis, where the data is plotted and a sigmoidal function is fitted following the equation:

$$y = \frac{F_{max}}{1 + e^{-growthRate(x - \bar{x})}} \quad (4.1)$$

where y represents muscle activation, measured by maximum absolute value, and x represents amplitude. F_{max} , $growthRate$ and \bar{x} are specific parameters of the curve acquired from the curve fitting algorithms of Matlab: maximum value of muscle activation, the slope of the recruitment curve and its mean value.

The control algorithm implemented, then, acquires the current eEMG reference value, which is shaped after a trapezoid – in order to mimic biomechanical movements described by a linear increase in force, a plateau and a linear decrease in force. The maximum reference value is set to 50% of the maximum value of the normalized recruitment curve, so there is enough leeway for increasing stimulation and compensating fatigue. At the beginning of each control period, the last 20ms of recorded eEMG is acquired and the maximum absolute value is computed. This value is compared to the reference, generating an error value, which is used in the control algorithm. The controller implemented is a simple PI control, following the equation:

$$y = k_p (MAV(eEMG)_{reference} - MAV(eEMG)_{measured}) + \\ k_i \int_0^t (MAV(eEMG)_{reference} - MAV(eEMG)_{measured}) dt \quad (4.2)$$

where $MAV(eEMG)$ refers to the mean absolute value of evoked EMG and k_p , k_i denote controller proportional and integral gains, respectively. Also, an anti-windup strategy was added to the algorithm by discarding integral additions as the controller reaches saturation.

The output control value is normalized with the same parameters used by the recruitment curve to values between 0 to 1. This value is added to the reference trapezoidal curve value (which varies from 0 to 0.5) and the respective amplitude is acquired by using the inverse function of Equation 4.1:

$$x = \frac{-\ln\left(\frac{F_{max}}{y} - 1\right)}{growthRate} + \bar{x} \quad (4.3)$$

With the acquired value at hand, the stimulation parameters are overwritten and stimulation is resumed. Algorithm 4.1 represents the procedure just described.

Algorithm 4.1 Pseudocode of the eEMG control algorithm experimented.

```

1: while stop button not pressed do
2:   acquire last 20ms of eEMG
3:   calculate maximum absolute value
4:   acquire last 20ms of analog input
5:   calculate current reference value
6:   calculate error value
7:   calculate control output based on error
8:   add current reference to control output
9:   find parameter based on inverse sigmoidal function
10:  stimulate
11:  delay until loop period reaches 20ms
12: end while

```

4.1.5 Frog and Rat Preparations

The experiments described in this chapter make use of two animal species. The first ones, grass frogs, are used in order to validate real-time capabilities of the system and hardware setup. They are euthanized and made available after undergoing other experimental procedures approved by the University of Utah Institutional Animal Care and Use Committee. Only after the decapitation do the experiments take place, since the nerve activity should be preserved for a few hours following death. The second set of experiments use Sprague-Dawley rats, which included a somewhat similar setup. The rats are used in different experimental routines for other projects, in which the rat is anesthetized using isofluorine in order to be kept alive during the experiments. All procedures were approved by

the University of Utah Institutional Animal Care and Use Committee [84]. Vital signals, such as oxygen concentration and cardiac rhythm, are taken every 15 minutes.

Using tweezers and scissors, the thigh skin of the grassfrog is cut and tissues involving the sciatic nerve are separated using a glass probe in order not to damage the nerve. The sciatic nerve is then placed hanging on a hook electrode (used for whole-nerve stimulation), which is affixedated by a magnetic stand. The crus skin is then cut open in order to surgically expose the muscles and two wire electrodes are placed into *gastrocnemius*, separated from each other by a one-centimeter distance. The hook electrode is connected to a Micro+Stim front-end with $1.3\mu A/step$ resolution and, as the Micro+Stim recording capabilities are considered damaged by the facricant, the wire electrodes are connected to a Micro front-end. It is important to make sure both ground and reference are separated and the fast-settle option within Trellis is turned off, since the artifact should not be significant for this setup. Also, in order to maintain the nerve functioning properly, Ringer's solution is applied to the tissue after each short experiment.



Figure 4.4: Experimental setup used in frog experiments.

For rat experiments, wire EMG electrodes are placed in lateral gastrocnemius (LG), medial gastrocnemius (MG), tibialis anterior (TA) and soleus (Sol) muscles in the previous experiments. In addition, a hook electrode is placed on the sciatic nerve for stimulation. A couple of EMG electrodes from MG is connected to Micro+Stim front-end, as well as the hook electrode. After the experiments, the rat is sacrificed.



Figure 4.5: Experimental setup used in rat experiments.

4.1.6 Experiments

The first set of experiments aim at validating the real-time capabilities of the system during stimulation and recording. Having this goal in mind, initial experiments are performed using a function generator (Wavetek Meterman FG2C Function Generator) at different frequencies and very low amplitude. The function generator is connected to the Micro front ends. The Micro+Stim front-end is connected to an oscilloscope (Tektronix TDS 1012) through a $1k\Omega$ resistor. The initial script for system validation is run and data is saved. Followingly, the system is tested using the frogs described above. The code run is also the script used for testing real-time, recording and stimulating validation. Problems perceived in the Micro+Stim front end during the latter experiments lead to a replacement of the piece. A new Micro+Stim, with $7\mu A/step$ resolution, is used in the following experiments. In order to test its functionalities, the first experiment using a function generator and an oscilloscope is run once again using the Micro+Stim alone, since its recording capabilities were intact.

In sequence, two experimental sessions with rats are made. As noted before, the rat preparation follow the usual procedures listed in [84]; however, for this part, it is proposed stimulating the *common fibular nerve*, a branch of the sciatic nerve that innervates muscles responsible for plantar flexion [85, p. 487]. This way, a force sensor can be fixed to the foot and force data can be acquired. For the first rat, the mentioned branch is stimulated; for the second, due to difficulties involving surgical approaches, the *tibial branch*, which innervates muscles responsible for dorsiflexion, is surgically severed and stimulation is made at the sciatic nerve.

During the first rat session, different setups are tried for recording and stimulating simultaneously, including: using Micro+Stim for both recording and stimulation; using Micro for recording and Micro+Stim for stimulation; and using Micro/Micro+Stim for recording while stimulating with a third-party stimulator (SD9 Square Pulse Stimulator - Grass Technologies). Finally, a Surf D front-end is used, which acts similarly to the EMG front-end described in Subsection 4.1.1.

In the second session, the setup tested is Surf D front-end (for recording) and Micro+Stim front-

end (for stimulation). After testing the setup using the initial script, a recruitment curve is generated with amplitude values using the script described in Subsection 4.1.4. Using the recruitment curve, the control algorithm is initially tested. Afterwards, a force transducer (Force Displacement Transducer Model FT03 - Grass Technologies) is hooked to the left foot by suturing a surgical line into it. The ankle is fixed so the force generated can represent the desired contraction more accurately.

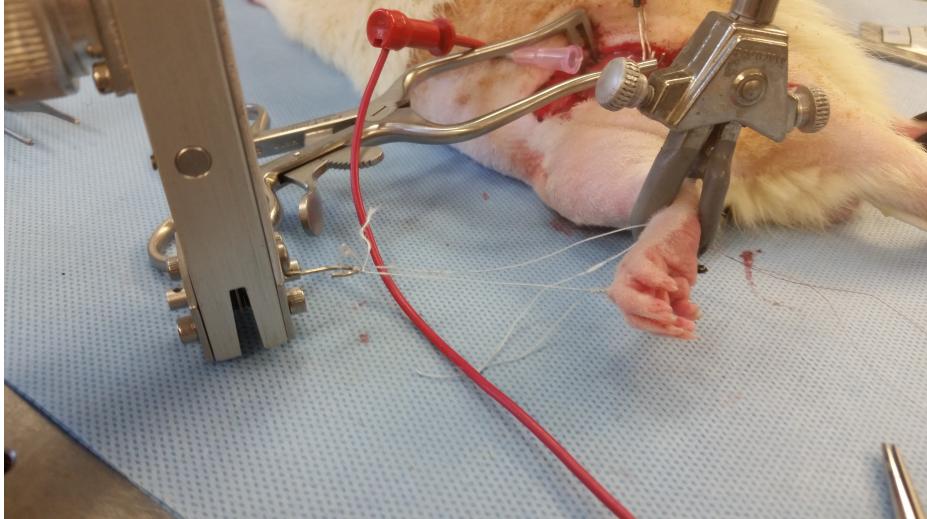


Figure 4.6: Force sensor attached to rat foot.

Figure 4.7: Force sensor (to the left) fixed to rat finger. The ankle position is fixed in order to acquire better measurements.

Different enhancements are proposed during the experimental procedures. One of which relates to the way eEMG is used for control: at first, eEMG is acquired every 20ms and mean absolute value is used as control input. Later on, using a moving window filter of the last 100ms of eEMG data, the mean absolute value of the whole window is taken into account.

4.2 Results

The first results that can be observed are related to first experiments using function generator as input signals and reading output signals with an oscilloscope. Since this experiment was used for validation of code real-time aspects, the timings observed for each interaction are plotted in sequence in Figure 4.8. In this plot, the difference of time recorded between each interaction is shown, as each time dedicated for each function execution: t_{send} refers to time dedicated to data logging and sending; $t_{control}$ refers to time dedicated to data acquisition and control implementation; t_{delay} refers to idle time, namely, time during which no processing is made; t_{gui} refers to time dedicated to GUI commands callbacks.

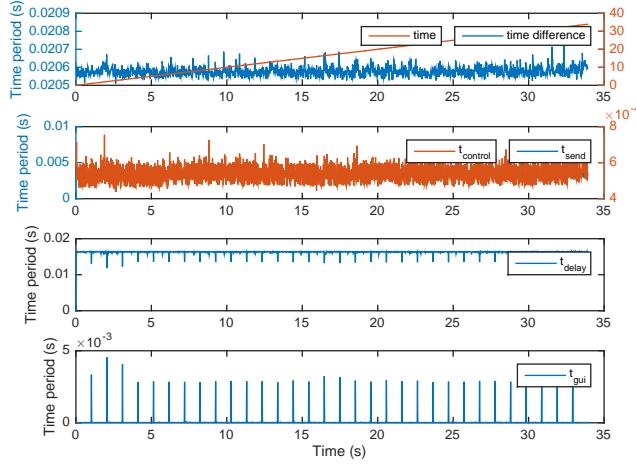


Figure 4.8: Loop periods during oscilloscope recording and stimulation. UDP communication was turned on during the execution.

Followingly, results related to real-time validation and recording of stimulation provided by code are shown. These results were collected during experiments with frogs. Firstly, real-time variables are shown as described earlier for Figure 4.8.

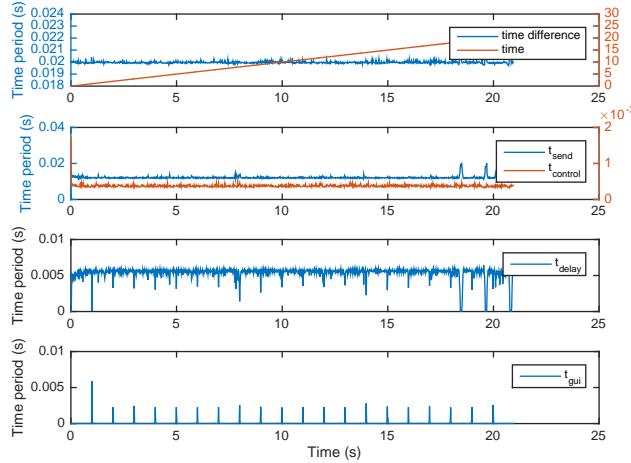


Figure 4.9: Loop periods during certain experiment on frog. UDP communication was turned on during the execution.

In frog experiments, evoked EMG was also recorded. Stimulation parameteres were fixed at a 0.5Hz frequency and $130\mu\text{A}$ of amplitude.

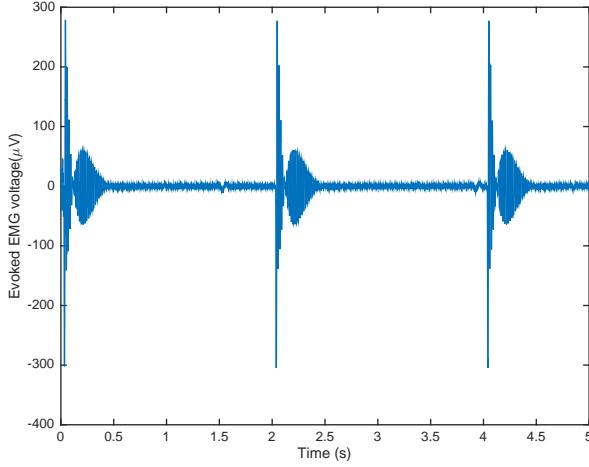


Figure 4.10: Data recorded during initial script running on second frog experiment.

The third set of experiments, as previously mentioned, were made with rats. For the first session of experiments, results are related to real-time consistency and the type of recordings that were compared in the end of the experiment. Time variables are shown below as described before.

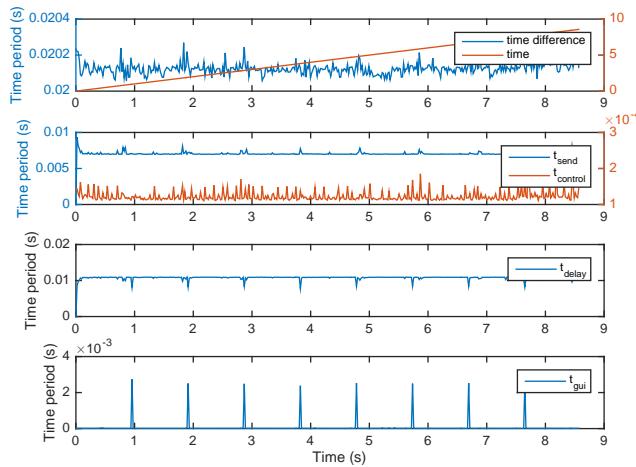


Figure 4.11: Loop periods during certain experiment on rat. UDP communication was turned on during the execution.

Evoked EMG signals were also recorded. As two main setups were tested (single-ended and differential recordings), they are compared in Figure 4.12. For the plots shown below, stimulation was made by the third-party stimulator described before.

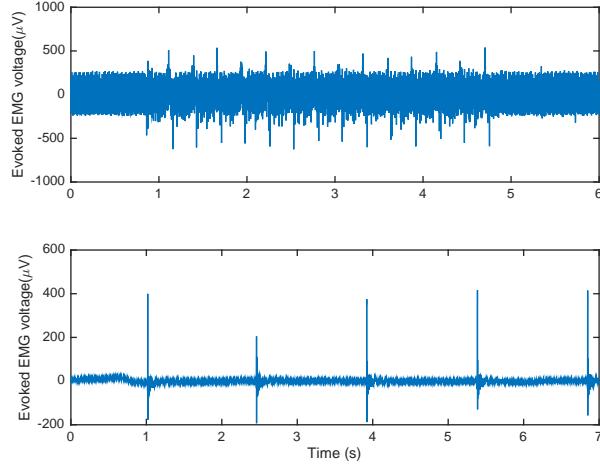


Figure 4.12: Representation of evoked EMG recorded for single-reference (above) and differential electrodes (below).

The second session of experiments have results related to control aspects. Followingly, different plots are shown representing recruitment curve generation and control results. Figure 4.13 shows two plots: one detailing the stimulation artifact found when changing stimulation polarity (done 10s after beginning of experiment) and the other showing behavior of eEMG after a long stimulation period (around 30s).

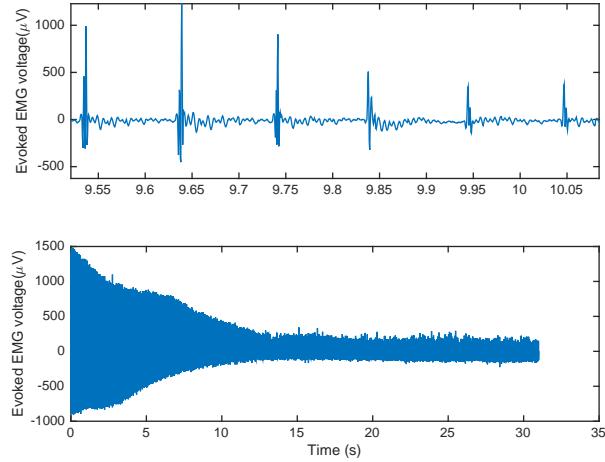


Figure 4.13: Evoked EMG in rat zoomed in; long experiment for fatigue perceiving.

In sequence, two recruitment curves are shown: one recorded after a long time of nerve exposure and one acquired after electrode repositioning. Both recruitment curves were generated by varying amplitude of stimulation.

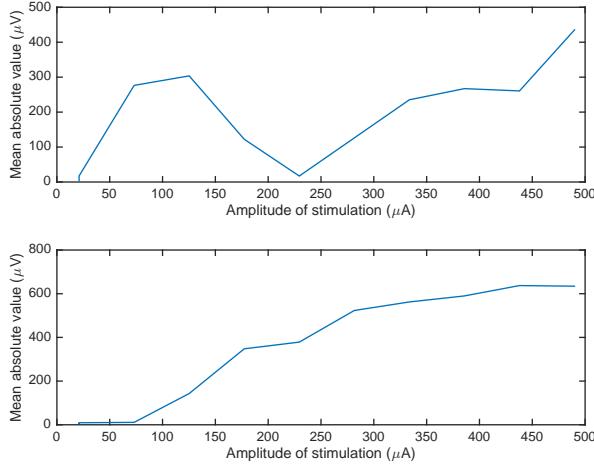


Figure 4.14: Recruitment curves before and after electrode repositioning.

The evoked EMG behavior during the control algorithm execution of the first recruitment curve shown in Figure 4.14 is presented below.

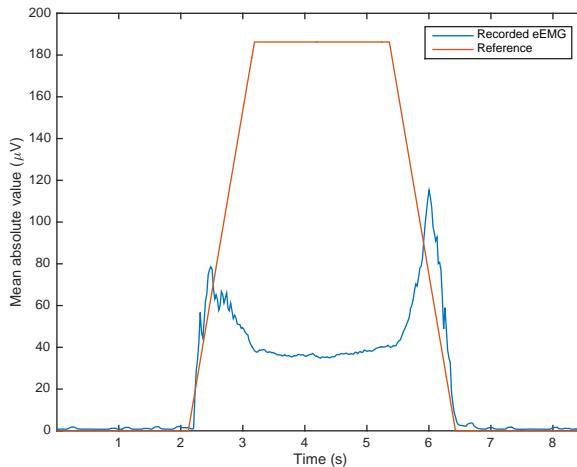


Figure 4.15: Respective control output for first recruitment curve in Figure 4.14.

After the left leg presented a quality decay in signals that will be explained in Section 4.3, the right leg went through the same procedure as the left one; for this part, however, the force transducer was not connected. The tests done were similar: for each recruitment curve generated, different PID values were tested in control. Every time the quality of response was inconsistent, the electrode was repositioned and the recruitment curve was plotted once again. The recruitment curve presented in Figure 4.16 was acquired after changing the rat leg.

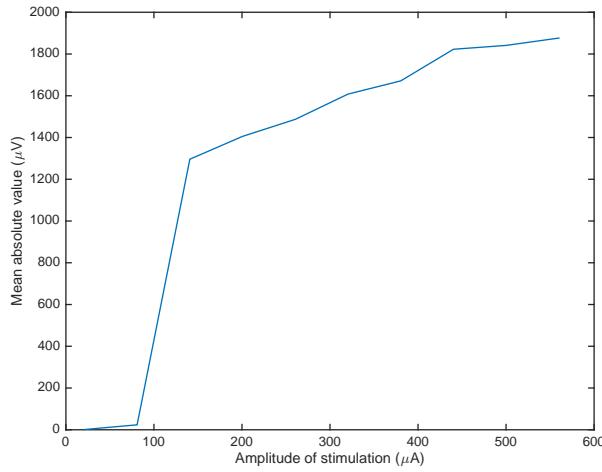


Figure 4.16: Recruitment curve found in second leg.

The following plots show the behavior of two control outputs (with different P constants) before and after applying a moving window filter of 100ms to evoked EMG recorded.

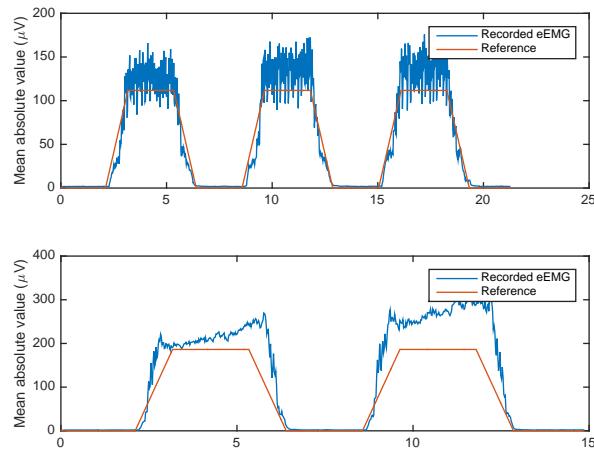


Figure 4.17: Control output before and after moving window filtering.

In the following figure, a plot of evoked EMG is shown atop the respective measurements performed by the force sensor.

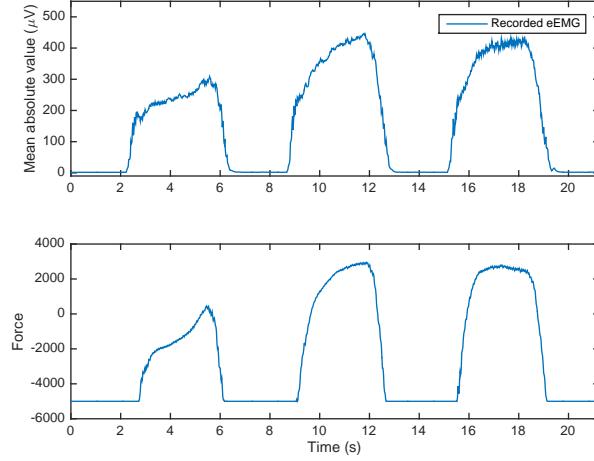


Figure 4.18: Evoked EMG during control execution and respective force measured by transducer.

Figure 4.19 shows control outputs for different k_p and k_i values (k_d is kept constant at zero).

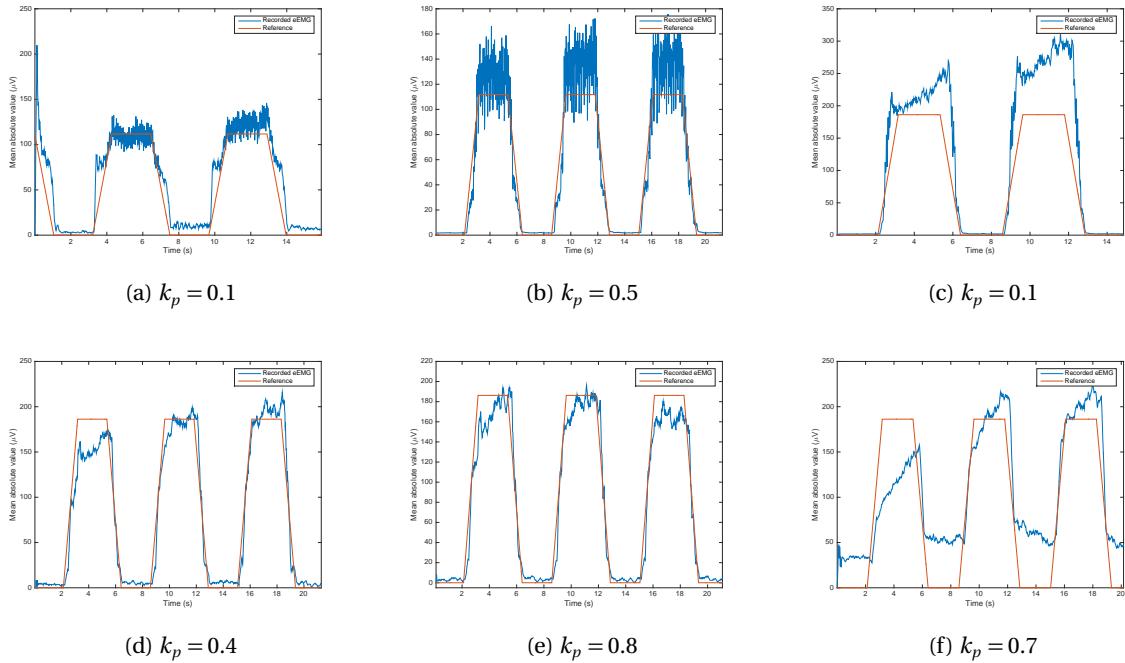


Figure 4.19: Control outputs for different k_p gain values.

These evoked EMG recordings were acquired during the final experiments, using $k_p = 0.7$.

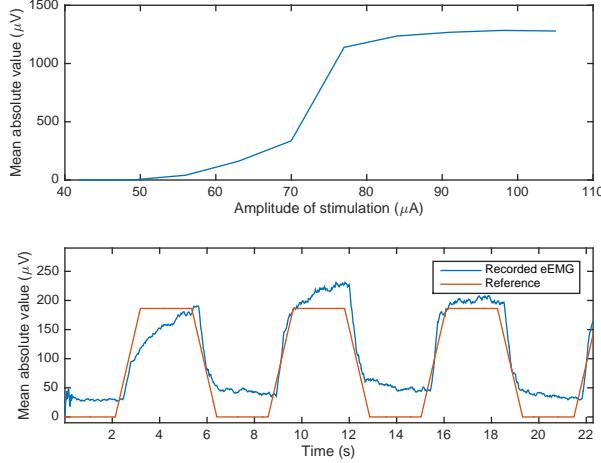


Figure 4.20: Control output recorded in last experiments.

Finally, the last experiment took place and eEMG was recorded while running the control algorithm for more than 60 seconds.

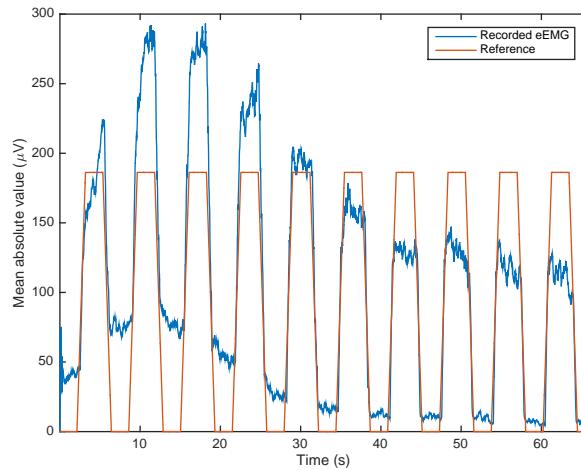


Figure 4.21: Control output recorded in very last experiment.

4.3 Discussion

There is a lot to be analyzed in each result acquired and shown in Section 4.2. In this section, we will describe each of the results and discuss the meaning behind them.

Firstly, time results analysis are shown in Figure 4.8, Figure 4.9 and Figure 4.11. It can be seen that the 20ms loop period required for control implementation is being respected with very little variation. Time used for data acquisition (communication with Grapevine) and control execution is very short, being below 1ms , whereas data logging and UDP communication is clearly seen as bottleneck in execution cost, as it varies around 10ms . Delay time (idle time), implemented in order to keep con-

sistency in overall loop period, is shown to be significant, meaning more routines can be implemented in code, such as control algorithms. The results are very consistent among different experiments and setups, providing enough validation of real-time requirements in simple and complex setups which include biological and nonbiological systems.

During frog experiments, we could acquire some eEMG recordings, which are shown in Figure 4.10. Even though there is a clear correlation between the stimulation times (as frequency of stimulation was set as 0.5 Hz) and the waveforms shown, the signal was considered too noisy for the application. Plus, a great deal of ringing is seen in the signal. At this point, while contacting the manufacturer, we found out the Micro+Stim front-end being used was not compatible with new firmware installed in Grapevine; thus, the piece was returned and a new Micro+Stim was provided, which was experimented in the rat.

The new setup was tested during the first experimental session with the rat. As the focus of this session was to arrange a proper setup for stimulating and recording the signals, several different setups were tried, as mentioned before. However, using the Micro/Micro+Stim front-ends to record eEMG signal was shown to be very noisy, even with reference and recording channel inserted within the same muscle. As we moved to Surf D front-end, which worked using differential signals in relation to a reference positioned in the rat upper body skin, the signal quality increased to a standard that allowed experiments to proceed, as seen in Figure 4.12.

There were some worries at this point about the possibility of the signal recorded being mostly artifact. Even though it was understood that the setup used would not allow a significant artifact (due to distance between stimulation and recording), a few experiments that corroborated the affirmation were run and the corresponding results are depicted in Figure 4.13. As the stimulation polarity changed at 10s , it was believed the polarity of the signal recorded would change as well in case it was a stimulation artifact. However, it can be seen that the greatest peak does not change (even though it decreases in amplitude); in fact, a small oscillation is seen to change before this peak, which is believed to be the stimulus artifact. The other test run is shown in the plot underneath, which represents the continuous stimulation for a long period of time (more than 30s). As it is known, stimulation artifacts are not affected by muscle fatigue, since they are not part of a biological process, just a propagation of the electrical stimulation itself; thus, if the waveforms recorded were all but artifacts, they would not decrease in amplitude after prolonged stimulation. Figure 4.13, nevertheless, shows a different scenario, which reaffirms that the great peak recorded, on which control would be based, represents an M-wave, rather than stimulus artifact.

Once it was concluded that the signal being measured could be used for control, the procedure resumed as described in Subsection 4.1.6. At this point, it is interesting to highlight two interesting results that are presented in Figure 4.14. As explained before, this curve tends to follow the behavior described by the lower plot. However, the upper plot was found several times, usually after long procedures. It was found that, due to nerve exposure to oxygen and continuous stimulation, its propagation properties were being harmed, leading to nerve degradation. This brought forth a great disadvantage of the setup proposed: the usage of a hook electrode for stimulation demands a shorter period of time for experiments. In case this is now followed, the nerve conduction properties will degrade over time.

In order to counteract this, every time control behaved as presented in Figure 4.15, the hook electrode was repositioned further away from the previous local, where tissue damage gradually became visible. The lower plot of Figure 4.14 shows how electrode repositioning increased quality of the recruitment curve.

Another important aspect of the recruitment curves acquired that should be further discussed is shown in Figure 4.16, which was recorded after changing the leg, as described previously in Subsection 4.1.6. In this figure, the recruitment curve is perceived as being very steep, providing a good threshold separation for some applications. However, as the control algorithm implemented wanders around the central point of the curve, assuming different values above and below it, it is important for this section to possess very good resolution, which is not acquired in this case. In order to implement the algorithm control, thus, different portions of the curve were tried. This way, different amplitude values were used for plotting a recruitment curve with good enough resolution for the control algorithm. Finally, a recruitment curve with amplitude values between 42 and $70\mu A$ was found to be good enough for the applications.

Finally, as experiments with the control algorithm advanced, it was noted that the control output was varying too much in a very short amount of time. This led to poor visualization during assessment of controller efficiency. The evoked EMG signal used for the algorithm, thus, was filtered using a moving window of $100ms$, which enhanced the signal regularity, as seen in Figure 4.17, and allowed us to deal better with the parameters of the controller. Also, it was made possible to correlate the relation between evoked EMG and output force, shown in Figure 4.18. As the force produced by the muscle contraction would be an important variable when assessing fatigue compensation, this straightforward relation between both variables provided a good validation of the system.

As for the gain values used when designing the controller, the approach used was somewhat empirical. Firstly, the relation between the proportional section of the controller was assessed. By zeroing k_i in the controller and varying k_p , the evoked EMG response in relation to the reference was evaluated, as shown in Figures 4.19a and 4.19b. After a satisfactory k_p was found, the integral portion was dealt with by changing it to different values. However, there was a bug in the PID controller implementation that prevented the integral term to be correctly implemented, leading to the behavior of a proportional controller. Finally, as the best k_p value was found, the final controller was designed, which can be seen in Figure 4.19f. However, due to bad quality of the recruitment curve generated and fatiguing behavior starting to manifest at this point, the controller did not reach its best performance.

The final experimental results acquired, which are represented by Figure 4.20, unfortunately had a serious issue. This is perceived in the minimum reference values, which are not followed as well as the rest. If the recruitment curve is not plotted correctly, the minimum value for stimulation does not lead to a response close to zero, rather, it leads to an offset that can be seen in the figures. As seen in the recruitment curve shown, the minimum amplitude was close to $40\mu A$, which is considered to be a very high value that already leads to significant muscle activity. This analysis shows how tight the bonds between the recruitment curve and the algorithm control are: if the former is not plotted correctly, there will be serious issues in the latter.

As a final result, Figure 4.21 demonstrate how the algorithm behaves as fatigue manifests. Un-

fortunately, due to the problems regarded in the last paragraph, the control algorithm did not reach its full potential. This plot, however, represents very well how fatigue affects the system and how the steepness of the recruitment curve changes the controller behavior for the minimum reference values.

Chapter 5

Conclusions

5.1 Final Considerations

An experimental setup for the development of a navigation algorithm in cockroaches was successfully developed, along with software and hardware for interfacing with the insect nervous system and acquiring position data. As for the navigation control strategies, a line-follower control algorithm was achieved and results in simulated environment were reasonable; however, due to particularities in the insect nervous system, the experimental results for real cockroaches was most times problematic. Even so, there were situations in which the work functioned with good performance, allowing the cockroach to thread on a straight-line path.

We also developed with success a real-time environment for nerval and muscle interface, allowing the achievement of a FES control system that guides muscle contraction along a desired reference. The system works with good real-time consistency, which is necessary for control algorithm implementations, and interfaces with the nervous system with good performance. However, due to nerval exposure to oxygen, the experimental procedures are somewhat short time-wise. Even so, a proportional control algorithm was implemented using a recruitment curve, providing good results in terms of dynamic and static errors.

5.2 Future Work

As experiments progressed and result analysis advanced, a few improvements surfaced with both easy and more complex implementations. Some of them are referred to software, others to hardware and even to the arrange of experimental setup.

5.2.1 Insects Navigation Control Using Neural Interfaces

Concerning software improvements, the most straightforward strategy that should be implemented is the frequency of stimulation. Since the code runs in loop, it was previously implemented a sleep

function in order to decrease the processing. However, this sleep was not big enough, as it was wrongfully thought that the faster the execution, the better. Thus, it is proposed an implementation similar to that used in the second part of the work: the timing of the functions being called at the beginning of the loop followed by an specific delay subtracted by the initial execution time. This way, the frequency should be more consistent and be set to lower values, such as 10Hz or even 1Hz . Another enhancement that was actually implemented but can still be further explored is the data logging. The lack of proper position data of the insect showed to be a great disadvantage for posterior data analysis; thus, it is imperative for it to be recorded in future experiments.

Other improvements concern software, however, in a more complex algorithmic way: the implementation and testing of different control strategies. Initially, implementation of a PI controller would be beneficial for the overall performance, decreasing steady-state position error. Another control strategy, however, is a bit more complex and is dedicated to controlling orientation of the cockroach: even if the tracking sensor does not provide orientation information, it could be estimated by position difference. Thus, the implementation would follow approaches already tested which led to very good results [9]. Once these implementations are experimented with and the results are analyzed, an interesting approach would be to predetermine the path to be followed by an initial setup, which could be achieved by moving a passive marker around the desired zone. Followingly, the computer would calculate the necessary waypoints for the best interpolation possible and use them in the orientation control. All control strategies should also consider modulating pulse-width instead of amplitude; which could not be made possible due to hardware limitations.

Even with all software implementation enhanced, hardware improvements should be considered. As this was a first experimental version, a commercial hardware was used, which had some limitations already discussed. Hence, it is proposed the design of new hardware with specific features: reduced size and weight for better movement, hardware implementation of biphasic stimulation and current-modulated stimulation. All these characteristics should improve overall performance of the stimulation.

The final suggestions are related to the experimental setup used: the ideal setup should be in a broader space, with the least amount of distractions possible. A surface above ground should be considered, such as a table. The path chosen and implemented on code could be either longer or circular, as done in other works [9], which would allow a greater amount of recorded data for posterior analysis.

In order to advance the research itself, it is also proposed the study of cockroach nervous system and central pattern generators in order to control navigation not by antennal nerves, but by thoracic ganglia. This should enhance the performance of movement and electrode placement and fixation.

5.2.2 FES Control Using Evoked EMG

In terms of software enhancements that could be followingly implemented, it is suggested a proper proportional-integral controller, which should decrease steady-state error and increase overall performance of the controller. In sequence, the controller gains should be parameterized in order to

minimize muscle fatigue. Aiming at enhancing recruitment curve quality and resolution, specific parameters should be tested, such as interphase delay and catchlike-inducing trains.

As for further software advances, it is suggested the programming of the code in a virtual environment more proper for real-time applications, in order to decrease execution loop period and improve control efficiency.

Relating to experimental setup improvements, we propose the usage of a silicon-insulated electrode rather than the hook electrode tested. The former electrode should be able not to displace the nerve away from the animal body, allowing the experiments to run in better conditions regarding nerve exposure and recruitment curve quality.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LATIF, T.; BOZKURT, A. Line following terrestrial insect biobots. In: *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Institute of Electrical & Electronics Engineers (IEEE), 2012. Disponível em: <<http://dx.doi.org/10.1109/EMBC.2012.6346095>>.
- [2] HOLZER, R.; SHIMOYAMA, I. Locomotion control of a bio-robotic system via electric stimulation. In: *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS 97*. Institute of Electrical & Electronics Engineers (IEEE), 1997. Disponível em: <<http://dx.doi.org/10.1109/IROS.1997.656559>>.
- [3] BINDER-MACLEOD, S. A. et al. New look at force-frequency relationship of human skeletal muscle: effects of fatigue. *J. Neurophysiol.*, v. 79, n. 4, p. 1858–1868, Apr 1998.
- [4] ANTHES, E. *Frankenstein's Cat: Cuddling Up to Biotech's Brave New Beasts*. [S.l.]: Scientific American / Farrar, Straus and Giroux, 2014. ISBN 0374534241.
- [5] POPOVIC, D.; SINKJAER, T. *Control of Movement for the Physically Disabled*. First edition. London: Springer-Verlag, 2000.
- [6] DURAND, D. M. What is neural engineering? *J. Neural Eng.*, IOP Publishing, v. 4, n. 4, sep 2006. Disponível em: <<http://dx.doi.org/10.1088/1741-2552/4/4/E01>>.
- [7] SATO, H. et al. A cyborg beetle: Insect flight control through an implantable, tetherless microsystem. In: *2008 IEEE 21st International Conference on Micro Electro Mechanical Systems*. Institute of Electrical & Electronics Engineers (IEEE), 2008. Disponível em: <<http://dx.doi.org/10.1109/MEMSYS.2008.4443618>>.
- [8] VERDERBER, A.; MCKNIGHT, M.; BOZKURT, A. Early metamorphic insertion technology for insect flight behavior monitoring. *Journal of Visualized Experiments*, MyJove Corporation, n. 89, 2014. Disponível em: <<http://dx.doi.org/10.3791/50901>>.
- [9] WHITMIRE, E.; LATIF, T.; BOZKURT, A. Kinect-based system for automated control of terrestrial insect biobots. In: *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. Institute of Electrical & Electronics Engineers (IEEE), 2013. Disponível em: <<http://dx.doi.org/10.1109/EMBC.2013.6609789>>.
- [10] DILORENZO, J. D. B. D. J. *Neuroengineering*. [S.l.]: CRC Press, 2007. ISBN 0849381746.

- [11] TEETER, J. O.; KANTOR, C. *Functional Electrical Stimulation (FES) Resource Guide for Persons With Spinal Cord Injury or Multiple Sclerosis*. [S.l.]: F E S Information Center, 1995. ISBN 1888470038.
- [12] HUDSON, D. L.; COHEN, M. E. Neural signal processing. In: HE, B. (Ed.). *Neural Engineering*. [S.l.]: Kluwer Academic/Plenum Publishers, 2005. cap. 6.
- [13] GOLD, C. On the origin of the extracellular action potential waveform: A modeling study. *Journal of Neurophysiology*, American Physiological Society, v. 95, n. 5, p. 3113–3128, jan 2006. Disponível em: <<http://dx.doi.org/10.1152/jn.00979.2005>>.
- [14] HODGKIN, A. L.; HUXLEY, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, Wiley-Blackwell, v. 117, n. 4, p. 500–544, aug 1952. Disponível em: <<http://dx.doi.org/10.1113/jphysiol.1952.sp004764>>.
- [15] SQUIRE, L. R. et al. *Fundamental Neuroscience*. Third edition. [S.l.]: Elsevier, 2008.
- [16] GUYTON, A. C.; HALL, J. E. *Textbook of Medical Physiology*. Eleventh edition. [S.l.]: Elsevier Saunders, 2006.
- [17] KAMEN, G. Electromyographic kinesiology. In: ROBERTSON, D. e. a. (Ed.). *Research Methods in Biomechanics*. Second edition. Chicago, IL: Human Kinetics Publisher, 2014.
- [18] PAJARO-BLAZQUEZ, M. et al. Challenges in measurement of spasticity in neurological disorders. In: PONS, J. L.; TORRICELLI, D. (Ed.). *Emerging Therapies in Neurorehabilitation*. [S.l.]: Springer, 2014. v. 4.
- [19] KANDEL, E. R.; SCHWARTZ, J. H.; JESSELL, T. M. *Principles of Neural Science*. Fourth edition. NY, United States of America: The McGraw-Hill Companies, 2000.
- [20] KRALJ, A.; BAJD, T. *Functional Electrical Stimulation: Standing and Walking After Spinal Cord Injury*. [S.l.]: CRC Press, 1989.
- [21] BURKE, R. E.; RUDOMIN, P.; ZAJAC, F. E. The effect of activation history on tension production by individual muscle units. *Brain Res.*, v. 109, n. 3, p. 515–529, Jun 1976.
- [22] CAMERON, E. *The Cockroach (Periplaneta americana, L.)*. [S.l.]: William Heinemann, Medical Books, Ltd., London, 1961.
- [23] PRINGLE, J. W. S. Proprioception in insects. *Journal of Experimental Biology*, Company of Biologists, v. 15, n. 4, p. 467–473, 1938. ISSN 0022-0949. Disponível em: <<http://jeb.biologists.org/content/jexbio/15/4/467.full.pdf>>.
- [24] BABA, Y.; COMER, C. M. Antennal motor system of the cockroach, *periplaneta americana*. *Cell Tissue Res*, Springer Science + Business Media, v. 331, n. 3, p. 751–762, jan 2008. Disponível em: <<http://dx.doi.org/10.1007/s00441-007-0545-9>>.
- [25] BELL, W. J. *Searching Behaviour*. [S.l.]: Springer-Science + Business Media, B V, 1990.

- [26] YE, S. et al. The antennal system and cockroach evasive behavior. i. roles for visual and mechanosensory cues in the response. *Journal of Comparative Physiology A*, Springer-Verlag, v. 189, n. 2, p. 89–96, 2003. ISSN 0340-7594.
- [27] BADIA, J. et al. Comparative analysis of transverse intrafascicular multichannel, longitudinal intrafascicular and multipolar cuff electrodes for the selective stimulation of nerve fascicles. *J. Neural Eng.*, IOP Publishing, v. 8, n. 3, p. 036023, may 2011. Disponível em: <<http://dx.doi.org/10.1088/1741-2560/8/3/036023>>.
- [28] STOKES, I. A.; HENRY, S. M.; SINGLE, R. M. Surface EMG electrodes do not accurately record from lumbar multifidus muscles. *Clinical Biomechanics*, Elsevier BV, v. 18, n. 1, p. 9–13, jan 2003. Disponível em: <[http://dx.doi.org/10.1016/S0268-0033\(02\)00140-7](http://dx.doi.org/10.1016/S0268-0033(02)00140-7)>.
- [29] HUIGEN, E.; PEPER, A.; GRIMBERGEN, C. A. Investigation into the origin of the noise of surface electrodes. *Medical & Biological Engineering & Computing*, Springer Science + Business Media, v. 40, n. 3, p. 332–338, may 2002. Disponível em: <<http://dx.doi.org/10.1007/BF02344216>>.
- [30] MOLLER, A. R. *Intraoperative Neurophysiological Monitoring*. [S.l.]: Springer, 2011.
- [31] DOWDEN, B. R. et al. Non-invasive method for selection of electrodes and stimulus parameters for FES applications with intrafascicular arrays. *J. Neural Eng.*, IOP Publishing, v. 9, n. 1, p. 016006, dec 2011. Disponível em: <<http://dx.doi.org/10.1088/1741-2560/9/1/016006>>.
- [32] BARROSO, F. et al. Surface emg in neurorehabilitation and ergonomics: State of the art and future perspectives. In: PONS, J. L.; TORRICELLI, D. (Ed.). *Emerging Therapies in Neurorehabilitation*. [S.l.]: Springer, 2014. v. 4.
- [33] HARGROVE, L.; SCHEME, E.; ENGLEHART, K. Myoelectric prostheses and targeted reinnervation. In: FARINA, D.; JENSEN, W.; AKAY, M. (Ed.). *Instruction to Neural Engineering for Motor Rehabilitation*. [S.l.]: IEEE Press, 2013.
- [34] FOFFANI, G. et al. Decoding sensory stimuli from populations of neurons: Methods for long-term longitudinal studies. In: AKAY, M. (Ed.). *Handbook of Neural Engineering*. [S.l.]: IEEE Press, 2007.
- [35] TAM, H.; WEBSTER, J. G. Minimizing electrode motion artifact by skin abrasion. *IEEE Transactions on Biomedical Engineering*, Institute of Electrical & Electronics Engineers (IEEE), BME-24, n. 2, p. 134–139, mar 1977. Disponível em: <<http://dx.doi.org/10.1109/TBME.1977.326117>>.
- [36] LEWICKI, M. S. A review of methods for spike sorting: the detection and classification of neural action potentials. *Network: Computation in Neural Systems*, v. 9, n. 4, p. R53–R78, 1998. PMID: 10221571. Disponível em: <http://www.tandfonline.com/doi/abs/10.1088/0954-898X_9_4_001>.
- [37] BONFANTI, A. et al. A low-power integrated circuit for analog spike detection and sorting in neural prosthesis systems. In: *2008 IEEE Biomedical Circuits and Systems Conference*. Institute of Electrical & Electronics Engineers (IEEE), 2008. Disponível em: <<http://dx.doi.org/10.1109/BIOCAS.2008.4696923>>.

- [38] FEE, M. S.; MITRA, P. P.; KLEINFELD, D. Automatic sorting of multiple unit neuronal signals in the presence of anisotropic and non-gaussian variability. *Journal of Neuroscience Methods*, Elsevier BV, v. 69, n. 2, p. 175–188, nov 1996. Disponível em: <[http://dx.doi.org/10.1016/S0165-0270\(96\)00050-7](http://dx.doi.org/10.1016/S0165-0270(96)00050-7)>.
- [39] QUIROGA, R. Q.; NADASDY, Z.; BEN-SHAUL, Y. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Computation*, MIT Press - Journals, v. 16, n. 8, p. 1661–1687, aug 2004. Disponível em: <<http://dx.doi.org/10.1162/089976604774201631>>.
- [40] FRIGO, C. et al. EMG signals detection and processing for on-line control of functional electrical stimulation. *Journal of Electromyography and Kinesiology*, Elsevier BV, v. 10, n. 5, p. 351–360, oct 2000. Disponível em: <[http://dx.doi.org/10.1016/S1050-6411\(00\)00026-2](http://dx.doi.org/10.1016/S1050-6411(00)00026-2)>.
- [41] RIPPLE, LLC. *Grapevine's User Manual*. [S.I.], 2015.
- [42] BARZILAY, O.; WOLF, A. A fast implementation for EMG signal linear envelope computation. *Journal of Electromyography and Kinesiology*, Elsevier BV, v. 21, n. 4, p. 678–682, aug 2011. Disponível em: <<http://dx.doi.org/10.1016/j.jelekin.2011.04.004>>.
- [43] GORMAN, P. H.; MORTIMER, J. T. The effect of stimulus parameters on the recruitment characteristics of direct nerve stimulation. *IEEE Transactions on Biomedical Engineering*, Institute of Electrical & Electronics Engineers (IEEE), BME-30, n. 7, p. 407–414, jul 1983. Disponível em: <<http://dx.doi.org/10.1109/TBME.1983.325041>>.
- [44] POPOVIC, D. B.; POPOVIC, M. B. Methods for movement restoration. In: FARINA, D.; JENSEN, W.; AKAY, M. (Ed.). *Instruction to Neural Engineering for Motor Rehabilitation*. [S.I.]: IEEE Press, 2013.
- [45] POPOVIC, D.; BAKER, L.; LOEB, G. Recruitment and comfort of BION implanted electrical stimulation: Implications for FES applications. *IEEE Trans. Neural Syst. Rehabil. Eng.*, Institute of Electrical & Electronics Engineers (IEEE), v. 15, n. 4, p. 577–586, dec 2007. Disponível em: <<http://dx.doi.org/10.1109/TNSRE.2007.909816>>.
- [46] DURFEE, W.; MACLEAN, K. Methods for estimating isometric recruitment curves of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*, Institute of Electrical & Electronics Engineers (IEEE), v. 36, n. 7, p. 654–667, jul 1989. Disponível em: <<http://dx.doi.org/10.1109/10.32097>>.
- [47] JENSEN, W.; HARREBY, K. R. Selectivity of peripheral neural interfaces. In: FARINA, D.; JENSEN, W.; AKAY, M. (Ed.). *Instruction to Neural Engineering for Motor Rehabilitation*. [S.I.]: IEEE Press, 2013.
- [48] DURAND, D. M.; GRILL, W. M.; KIRSCH, R. Electrical stimulation of the neuromuscular system. In: HE, B. (Ed.). *Neural Engineering*. [S.I.]: Kluwer Academic/Plenum Publishers, 2005.
- [49] BURKE, R. E.; RUDOMIN, P.; ZAJAC, F. E. Catch property in single mammalian motor units. *Science*, American Association for the Advancement of Science (AAAS), v. 168, n. 3927, p. 122–124, apr 1970. Disponível em: <<http://dx.doi.org/10.1126/science.168.3927.122>>.

- [50] ABBATE, F. et al. Prolonged force increase following a high-frequency burst is not due to a sustained elevation of [ca2+]. *AJP: Cell Physiology*, American Physiological Society, v. 283, n. 1, p. C42–C47, jul 2002. Disponível em: <<http://dx.doi.org/10.1152/ajpcell.00416.2001>>.
- [51] BEVAN, L. et al. The effect of the stimulation pattern on the fatigue of single motor units in adult cats. *The Journal of Physiology*, Wiley-Blackwell, v. 449, n. 1, p. 85–108, apr 1992. Disponível em: <<http://dx.doi.org/10.1113/jphysiol.1992.sp019076>>.
- [52] BINDER-MACLEOD, S. A.; BARKER, C. B. Use of a catchlike property of human skeletal muscle to reduce fatigue. *Muscle Nerve*, v. 14, n. 9, p. 850–857, Sep 1991.
- [53] MA, T.; GU, Y.-Y.; ZHANG, Y.-T. Circuit models for neural information processing. In: HE, B. (Ed.). *Neural Engineering*. [S.l.]: Kluwer Academic/Plenum Publishers, 2005. cap. 10.
- [54] BUTTS, D. A.; GOLDMAN, M. S. Tuning curves, neuronal variability, and sensory coding. *PLoS Biol.*, v. 4, n. 4, p. e92, Apr 2006.
- [55] VISVANATHAN, K. et al. Flight initiation and directional control of beetles by microthermal stimulation. *Solid-State Sensors, Actuators and Microsystems Workshop*, p. 126–129, 2008.
- [56] SATO, H. et al. Radio-controlled cyborg beetles: A radio-frequency system for insect neural flight control. In: *2009 IEEE 22nd International Conference on Micro Electro Mechanical Systems*. Institute of Electrical & Electronics Engineers (IEEE), 2009. Disponível em: <<http://dx.doi.org/10.1109/MEMSYS.2009.4805357>>.
- [57] MEYER, J. R. *Lecture notes in General Entomology*. [S.l.]: Department of Entomology of NC State University, February 2006.
- [58] WONG, R. K.; PEARSON, K. G. Properties of the trochanteral hair plate and its function in the control of walking in the cockroach. *J. Exp. Biol.*, v. 64, n. 1, p. 233–249, Feb 1976.
- [59] KIEN, J. The initiation and maintenance of walking in the locust: An alternative to the command concept. *Proceedings of the Royal Society B: Biological Sciences*, The Royal Society, v. 219, n. 1215, p. 137–174, sep 1983. Disponível em: <<http://dx.doi.org/10.1098/rspb.1983.0065>>.
- [60] PEARSON, K. G. Common principles of motor control in vertebrates and invertebrates. *Annu. Rev. Neurosci.*, Annual Reviews, v. 16, n. 1, p. 265–297, mar 1993. Disponível em: <<http://dx.doi.org/10.1146/annurev.ne.16.030193.001405>>.
- [61] WYMAN, R. J. et al. The drosophila giant fiber system. In: EATON, R. (Ed.). *Neural Mechanisms of Startle Behavior*. Springer US, 1984. p. 133–161. ISBN 978-1-4899-2288-5. Disponível em: <http://dx.doi.org/10.1007/978-1-4899-2286-1_5>.
- [62] WILHERE, G. F.; CRAGO, P. E.; CHIZECK, H. J. Design and evaluation of a digital closed-loop controller for the regulation of muscle force by recruitment modulation. *IEEE Transactions on Biomedical Engineering*, Institute of Electrical & Electronics Engineers (IEEE), BME-32, n. 9, p. 668–676, sep 1985. Disponível em: <<http://dx.doi.org/10.1109/TBME.1985.325584>>.

- [63] ZHANG, Q.; HAYASHIBE, M.; AZEVEDO-COSTE, C. Evoked electromyography-based closed-loop torque control in functional electrical stimulation. *IEEE Transactions on Biomedical Engineering*, Institute of Electrical & Electronics Engineers (IEEE), v. 60, n. 8, p. 2299–2307, aug 2013. Disponível em: <<http://dx.doi.org/10.1109/TBME.2013.2253777>>.
- [64] LI, Z. et al. Real-time closed-loop FES control of muscle activation with evoked EMG feedback. In: *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*. Institute of Electrical & Electronics Engineers (IEEE), 2015. Disponível em: <<http://dx.doi.org/10.1109/NER.2015.7146700>>.
- [65] KLAUER JORG RAISCH, T. S. C. Nonlinear joint-angle feedback control of electrically simulated and λ -controlled antagonistic muscle pairs. In: *European Control Conference (ECC)*. [S.l.: s.n.], 2013.
- [66] AYALI, A. et al. Sensory feedback in cockroach locomotion: current knowledge and open questions. *J Comp Physiol A*, Springer Science + Business Media, v. 201, n. 9, p. 841–850, nov 2014. Disponível em: <<http://dx.doi.org/10.1007/s00359-014-0968-1>>.
- [67] BACKYARD BRAINS. *The Roboroach*. 2013. Disponível em: <<https://backyardbrains.com/products/roboroach>>.
- [68] BACKYARD BRAINS. *The Roboroach's PCB schematics*. sep 2013. Disponível em: <<https://backyardbrains.com/products/files/RoboRoach.v.1.1b.pdf>>.
- [69] BACKYARD BRAINS. *RoboRoach's online repository*. may 2013. Disponível em: <<https://github.com/BackyardBrains/RoboRoach>>.
- [70] DIGITAL, N. *Medical Polaris Optical Tracking Systems*. 2015. Disponível em: <<http://www.ndigital.com/medical/products/polaris-family/>>.
- [71] DIGITAL, N. *Polaris Spectra's Measurement Volume*. 2015. Disponível em: <<http://www.ndigital.com/medical/products/polaris-family/features/measurement-volume/>>.
- [72] LEVY, L. de. *RoboRoach API online repository*. 2015. Disponível em: <<https://bitbucket.org/lucasdelevy/roboroach>>.
- [73] WEBSITE, B. T. *Bluetooth Low Energy*. 2015. Disponível em: <<http://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>>.
- [74] LEVY, L. de. *Communicating with RoboRoach Board*. 2015. Disponível em: <<http://lucasdelevy.com/index.php/2015/06/04/communicating-with-roboroach-board/>>.
- [75] MARINHO, M. *Polaris Driver*. 2013. Disponível em: <<http://sourceforge.net/projects/polarisdriver/>>.
- [76] BACKYARD BRAINS. *Experiment: Neural Interface Surgery*. 2015. Disponível em: <<https://backyardbrains.com/experiments/roboRoachSurgery>>.

- [77] BRASIL, P. da Republica do. *LEI No. 11.794, DE 8 DE OUTUBRO DE 2008*. 2008. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11794.htm>.
- [78] WIGGLESWORTH, V. B. et al. Do insects feel pain? *Antenna*, n. 1, p. 8–9, 1980.
- [79] RIPPLE, LLC. *Xippmex User Manual*. [S.I.], 2015.
- [80] MATHWORKS, T. *xPC Target For Use with Real-Time Workshop - Selecting Hardware Guide*. [S.I.], 2002.
- [81] MATHWORKS, T. *Real-Time Workshop For Use with SIMULINK*. [S.I.], 1994.
- [82] FRANKEL, M. A. *Control methods of asynchronous intrafascicular multi-electrode stimulation for neuromuscular prostheses*. Tese (Doutorado) — THE UNIVERSITY OF UTAH, 2013.
- [83] MICROSOFT. *Process Explorer v16.05*. 2015. Disponível em: <<https://technet.microsoft.com/en-us/sysinternals/processexplorer.aspx>>.
- [84] SEAH, N. E. et al. Nucleoprotein architectures regulating the directionality of viral integration and excision. *Proceedings of the National Academy of Sciences*, Proceedings of the National Academy of Sciences, v. 111, n. 34, p. 12372–12377, aug 2014. Disponível em: <<http://dx.doi.org/10.1073/pnas.1413019111>>.
- [85] FAAA, R. D. P.; FAAA, A. W. V. P.; FRCR, A. W. M. M. B. F. *Gray's Anatomy for Students: With STUDENT CONSULT Online Access, 3e*. [S.I.]: Churchill Livingstone, 2014. ISBN 0702051314.
- [86] LEVY, L. de. *Grapevine control GUI API online repository*. 2015. Disponível em: <<https://bitbucket.org/lucasdelevy/fns-neuroprosthesis>>.

ANEXOS

- Anexo I: Descrição do CD
- Anexo II: Programas Utilizados

I. DESCRIÇÃO DO CONTEÚDO DO CD

- trabalho_de_graduação.pdf: relatório de Trabalho de Graduação.
- resumo_do_trabalho.pdf: resumo do relatório de Trabalho de Graduação.
- abstract_do_trabalho.pdf: *abstract* do relatório de Trabalho de Graduação.

II. PROGRAMAS UTILIZADOS

Neste capítulo, será explicado o algoritmo em tempo real utilizado para o trabalho descrito no Capítulo 4. Os demais programas dos *setup* experimentais descritos nos capítulos 3 e 4 estão presentes e comentados em seus respectivos repositórios [72, 86].

Para a implementação do algoritmo, é utilizada uma função executada em *loop* até certo parâmetro mudar, como o valor de um botão. Essa função é detalhada em

```
1 function params = periodicFunction(params)
2     global gui_params;
3
4     % Reading from sensors and implementing control
5     control_func_timer = tic;
6     params = readAndStim(params);
7     params.control_func_time = toc(control_func_timer);
8
9     % Logging data into variable and sending to UDP channel
10    send_func_timer = tic;
11    params = logAndSendData(params);
12    params.send_func_time = toc(send_func_timer);
13
14    % Flushing callback buffer every 50*LOOP_PERIOD (1s) and showing time
15    gui_timer = tic;
16    if params.gui_counter == 50
17        gui_params.gui_text.String = sprintf('%.f', params.time);
18        drawnow; % necessary for button callbacks
19        params.gui_counter = 1;
20    else
21        params.gui_counter = params.gui_counter + 1;
22    end
23    params.gui_time = toc(gui_timer);
24
25    % Waiting until next period starts
26    delay_timer = tic;
27    while toc(delay_timer) < params LOOP_PERIOD - ...
28        (params.control_func_time + ...
29        params.send_func_time + ...
30        params.gui_time);
31        nop();
32    end
33    params.delay_time = toc(delay_timer);
34 end
```

Nessa função, **gui_params** são variáveis utilizadas na GUI para interface do usuário com o programa. A função **tic** retorna para **control_func_timer**, **send_func_timer**, **gui_timer** e **delay_timer** os valores de *clock* no momento de sua chamada. A função **toc**, então, utiliza essas variáveis para retornar quanto tempo foi registrado no *clock* desde que o respectivo **tic** foi gravado.

As funções **readAndStim()** e **logAndSendData()** executam as rotinas listadas em seus nomes: aquisição de dados (*read*), estimulação parametrizada (*stim*), gravação de dados (*log*) e envio de dados (*send*). A função nativa do Matlab **drawnow** é utilizada para chamada instantânea de *callbacks* da GUI.

No caso de funções executadas em períodos diferentes, como a rotina que utiliza **drawnow**, um contador é utilizado. Após 50 iterações, o contador é zerado e a rotina é executada. Desta forma, para um período de 20ms , as funções de *callback* são executadas a cada $(20 \cdot 50)\text{ms} = 1\text{s}$.

Após a execução de todas as funções necessárias, um *delay* de alta precisão é utilizado no *loop while*. Com o tempo total de execução (20ms) e os tempos de execução para cada função (presentes nas variáveis **control_func_time**, **send_func_time**, **gui_time**), a diferença é calculada. Essa diferença é utilizada no *delay* de alta precisão, que chama uma função sem operação, **nop()**, que implementa um método simples, como uma adição, ou visualização vazia (**disp("")**).

Desta forma, a função em tempo real é implementada com boa eficiência. É importante notar que um programa de mudança de prioridades do sistema, como o Process Explorer [83], deve ser executado e a prioridade do Matlab deve ser mudada para "Real-time" (24). Assim, a performance do *delay* de alta precisão é máxima.