Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Postgrado

Control of a tibio-femoral virtual joint by means of a hybrid EEG-EMG scheme

By:

Eng. Luis Antonio Mercado Cerda

in order to obtain the

Degree of Master of Science

in Electrical Engineering

San Nicolás de los Garza, Nuevo León September 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Postgrado

Control of a tibio-femoral virtual joint by means of a hybrid EEG-EMG scheme

By:

Eng. Luis Antonio Mercado Cerda

in order to obtain the

Master Degree of Science

in Electrical Engineering

San Nicolás de los Garza, Nuevo León September 2014

Universidad Autónoma de Nuevo León

Facultad de Ingeniería Mecánica y Eléctrica

Subdirección de Postgrado

Los miembros del Comité de Tesis recomendamos que la Tesis "Control of a tibio-femoral virtual joint by means of a hybrid EEG-EMG scheme", realizada por el alumno Ing. Luis Antonio Mercado Cerda, con número de matrícula 1338475, sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Ingeniería Eléctrica con Orientación en Control Automático.

El Comité de Tesis

Dra. Griselda Quiroz Compeán
Asesora

Dr. Luis Martín Torres Treviño
Revisor

Dr. David Gutiérrez Ruiz
Revisor

Vo. Bo.

Dr. Simón Martínez Martínez
Subdirector de Estudios de Postgrado

Acknowledgments

First I would like to thank my thesis advisor, Dr. Griselda Quiróz Compéan, for all her patience, support and specially her guidance in the development and writing of this thesis.

I would like to give my gratitude to Dr. Juan Ángel Rodríguez Liñán and Dr. Luis Martín Torres Treviño for their support, assistance and teachings that formed part of the basis of this thesis. I would also like to thank Dr. Dania Gutiérrez Ruiz for providing her assistance, required equipment and working space for the experiments. Thanks to Raúl Alonso Almada Aguilar for granting his time and services in performing as a test subject for the experiments.

I would like to thank Facultad de Ingeniería Mecánica y Eléctrica (FIME), the Universidad Autónoma de Nuevo León (UANL) and the Consejo Nacional de Ciencia y Tecnología (CONACYT) for granting me the scholarship number 444620 and the means to realize this thesis. Also thanks to Programa de Apoyo a la Investigación Científica y Tecnolígica (PAICYT) through project IT712-11 for granting the financial support to acquire equipment for the experiments.

Also thanks to my friends, Luis Lauro González for kindly helping me in many different ways, to Gustavo González who shared his ideas and knowledge with me. Also thank to my friends Carlos Onofre, Fernando Carranza, David Cabrera, Arely Rodríguez, Nicte Mata, Octavio Zamora, Elizabeth Hernández, Emirette Alonso and César Rebolloso for giving me support and friendship during the development of this thesis.

Finally, I would like to specially thank my parents. Dr. Luis Antonio Mercado Gómez and Ms. Maria Guillermina Cerda Ruiz for their unconditional love and support. For their live teachings and guidance throughout my life.

Thank you God, for granting me the strength, hearing my prayers and blessing me with all the supporting people that surround me.

To my parents.
I hope that my path
chosen for the future
will make you proud.

Contents

List of Figures

List of Tables

Nomenclatures

ApEN	Approximate Entropy
BCI	Brain-Computer Interface
CPS	Cyber Physical Systems
DWT	Discrete Wavelet Transform
ECG	Electrocardiographic
ECoG	Electrocorticography
EEG	Electroencephalographic
EMG	Electromyographic
EOG	Electrooculographic
FFT	Fast Fourier Transform
FSM	Finite-State Machine
hBCI	Hybrid-BCI
KMI	Kinesthetic Motor Imagery
MAV	Mean-Absolute-Value
NMI	Neural-Machine Interface
NNMS	Neural Network of Maximum Sensibility
PD+	Proportional Derivative Plus
PSD	Power Spectral Density
RMS	Root-Mean-Square
RoGO	Robotic Gait Orthosis
SVM	Support Vector Machine
VRE	Virtual Reality Environment
WPD	Wavelet Packet Decomposition

Abstract

Luis Antonio Mercado Cerda.

Candidate for the Degree of Master of Science in Electrical Engineering.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Title of the study:

Control of a tibio-femoral virtual joint by means of a hybrid EEG-EMG scheme

Number of pages: 85.

Abstract

Practical Brain-Computer Interfaces (BCIs) for disabled people should allow them to use all their remaining functionalities as control possibilities. BCIs are systems that connect the brain with external devices to perform that person's volition or intent, regardless if that individual is unable to perform the task due to body impairments. Sometimes these people have residual activity of their muscles; therefore, in this work we fuse electromyographic (EMG) with electroencephalographic (EEG) activity in a framework of the so called "Hybrid-BCI" (hBCI) approach. The EEG signals are used for the classifications of the locomotion activities, meanwhile the superficial EMG will be used to estimate the amplitude of the user intent.

These neural signals are used in the activation of two switching mathematical models of a tibio-femoral joint, which require a kinematic and dynamic study. For the switching between models a multi-level controller is used. The lower level consists of an individual controller for each model that is in charged of the tracking of the desired trajectory and different velocities of a standard human gait cycle. The mid-level uses a combination of a logical operator and a finite state machine for the switching between models. And the highest level consists in a neural network for the classification of the desired activity. All of this is implemented in a virtual representation of a tibiofemoral joint.

Signature of the advisor: ____________________

Dra. Griselda Quiroz Compeán

Chapter 1

Introduction

There are over millions of amputees who's lives have been drastically changed due to their limb losses. The number of individuals with this pathological condition has been increasing as the population grows and as there is an increment in vascular diseases. Over 75% of major amputations were lower-limb, with nearly 17% of lower-limb amputees suffering bilateral amputations [1]. Therefore, there is a continued need to provide this growing population of amputees with a mean to restore the mobility they once had as much as possible.

In the cyber physical systems (CPS) research community has come a great interest in the integration of both cyber systems and biomedical systems. One of the typical CPS is a neural-machine interface (NMI). These interfaces sense control signals from different locations, such as muscles [2], peripheral nerves [3], or the human brain [4] [5]. The acquired signals then control an external device, like a computer cursor, an internet browser, an exoskeleton or a prosthesis. However, the CPS cover a wide range of challenges, involving an accurate deciphering of the acquired neural signal into the user intent, a good control of the device, having a fair number of inputs and having a real-time neural control of the system [6].

NMI systems that manage the brain signals are usually better known as brain-computer interfaces (BCI). There exists two different methods to this brain signal approach: they are the electroencephalography (EEG) and the electrocorticography (ECoG). Although there exists promising results out of the ECoG procedure, this is an invasive method, with its main disadvantage being a surgery required to allocate the electrodes in the brain cortex. EEG, on the contrary, does not involve any invasion of the body. It mainly consists on allocating electrodes over the scalp in order to record the electric signals that come from the brain. This is why it comes to interest the research of non-invasive methods to acquire the neural signals that allow the control of a certain prosthesis.

1.1 Motivation

Although progress has been made in the area of BCI even in recent years, usually these researches revolve around different activities such as cursor movement, or activation of procedures like over-ground/treadmill walking or hand grasping. Also, the paradigms usually used in the experiments require prolonged training and sustained concentration by the subject in a specific thought or action. Furthermore, the current signal processing of the BCI usually consists just on classification. Nevertheless, there is evidence of using brain signals to decode kinematics of the lower limbs.

Along these lines, BCI is a good way to assist disable people, however it cannot be used for prolonged periods of time and usually requires assistance of an expert. For this reason BCI can be enhanced by the addition of another biosignal. This approach is usually called a hybrid-BCI (hBCI), and follows the concept of enhancing the residual capability of the users. Having the BCI and another channel allows to operate different procedures using the channels separately, or grants a more accurate response to a specific process combining the channels. Currently there exist few works using this hybrid approach combining heart rate, muscular or multiple brain signals; but still not all are appointed to grant a complete user intent. Note, however, that there are various types of BCI revolving around two main categories, invasive and non-invasive. Being the non-invasive the area of interest of this research, since it does not require surgical procedure or a state of discomfort to the user.

1.2 Literature Survey

An electromyography (EMG) based NMI developed by Zhang et al. [7] used the neural signals from the muscles to successfully classify user intent activities, from both an able-bodied subject and amputee subject. Their NMI design is focused on two aspects: the deciphering of the activities of sitting and standing, and focus on accurate classification even in presence of sensor failures or disturbances. Although their work shows an improvement for the prosthetic functions for specific user intent activities, it does not include the consideration for other movement task, such as walking, and it does not reach to the concept of user intent in a progressive activity, neither other biosignal is involved.

There is a recent report about the integration of an EEG-based BCI system with a robotic gait orthosis (RoGO) [8]. In the report the authors accomplished a successful treadmill walking of an able-bodied subject and a paraplegic subject. Their focus covers the classification of the brain signals during idling and walking kinesthetic motor imagery (KMI). With this, it can be seen that the model is subject to the activation of those specific activities. The results provided evidence that ambulation using brain signals is possible. In their work, however, the usage of another type of biosignal is limited since an EMG is used solely to discard volitional leg movements by the able-bodied subject.

Similarly there is a study where subjects operate a BCI-VRE (virtual reality environment) system in real time [9]. The operation system, using EEG signals of walking and idling KMI, granted the user to control the ambulation of an avatar. The system proved to be robust over time, and required a short training time, differently for many other works. The study showed promising results that BCI could control lower extremity activation of prosthesis, orthosis, or rehabilitation devices, but just EEG signals were used, it did not involved another biosignal whatsoever.

Seeing that a proper control for normal human gait in rehabilitation devices is needed, several studies have been proposed. In a paper by Bae et al. [10], an algorithm for gait rehabilitation was presented. Their iterative learning algorithm used the cyclic pattern of gait to correct the angular position of the knee in the following strides by using an assistive torque. The authors realized experiments using simulations and a healthy person performing an abnormal gait. The experiments with the individual were carried out using an actuator in an orthosis, meanwhile the simulations were realized by implementing two models of the human knee. The models were design accordingly for the two main phases of a walking stride, the swing phase and the stance phase. For the swing phase the model behaves like a pendulum and for the stance phase like a inverted pendulum. The study made by these authors focused mainly in rehabilitation, it did not involved limb loss and not making use of any biosignal.

Another control system based on motion intent recognition is proposed by Varol et al. [11]. They use Guassian mixture models to recognize the user intent from the information taken from the prosthesis sensors. The intent recognizer switches between the two modes, standing or walking. For that, a comparison between the prosthesis sensors and a database of walking patterns is what determine the switching between the gait modes. Following the study of Varol et al., an extension can be found in [12], where the control architecture of the powered prosthesis has changed to three levels. The high-level is the intent recognizer, the mid-level consists of the controllers for each mode, walking and standing, and the addition of two modes, sitting and stair ascent/descent. In the lower level a close-loop controller is in charge for the joint torque.

It comes to notice that most prosthesis or orthosis are just activated, by previously using a classifier that tells which activity to realize using one biosignal or solely react to a motion condition. Nonetheless, there are resent works that try to provide more volition of the user to the device. Such is the work of Ha et al. [13]. Their work uses EMG measurements, during a non-weight activity. The EMG classifier distinguishes between flexion or extention of the knee joint, and changes the joint torque only during muscular contraction; if the muscle is relax it maintains its position. So it can be seen that the device progressively follows the user's intent.

During ambulatory conditions, the forces and torques required for the gait need to be taken into consideration because it is a weight-activity, supporting the human body while performing a stance phase with one leg. In order to give the user control over the torque required to move the leg, an EMG to torque relation needs to be studied. Ullah et al. [14] showed that there are different types of models to calculate the EMG-torque relation and stated that the relation is non-linear.

As it has been mentioned, Do et al. [8] and King et al. [9] have an EEG classifier, and to be more precise they both use a linear discriminant analysis. Studies have shown there are different approaches for the EEG classification. For example, linear or quadratic discriminant analysis, linear regression [15], neural networks [16], support vector machines [17]. Each classifier has it own advantages and disadvantages, like accuracy, performance or computational requirements. Some of these methods of classification are used in existing hBCI. However, there are just a few examples of the hybrid approach. So it seems that this area of research is still young and with many improvements to develop.

1.3 Problem Description

The current computerized prostheses, using a microcontroller embedded in them along with motion sensors, form a closed loop that approximates to the natural human gait. However, the function of such computerized prosthesis is still limited. Based on mechanical sensing and prediction, they perform activities as walking, sitting, standing, and even stair climbing; nonetheless, they do not provide a complete user intent.

1.4 Hypothesis

The approach we take in this thesis is the creation of a system that combines two neural activities, being them EMG-based and EEG-based, thus a hBCI. This combination provides the necessary information to obtain the user intent in two different aspects:the activity to realize, and the magnitude of such activity.

So, the hypothesis of this research is that there exists a controller for these extracted/classified signals that can provide a time progressive signal instead of a simplified precise operation. This controlled signal does not only give a precise operation, but also a progressive signal that denotes desired positioning, speed, acceleration, and force. The scheme of the two NMI systems combined into the proposed hBCI system can be seen in Figure 1.1.

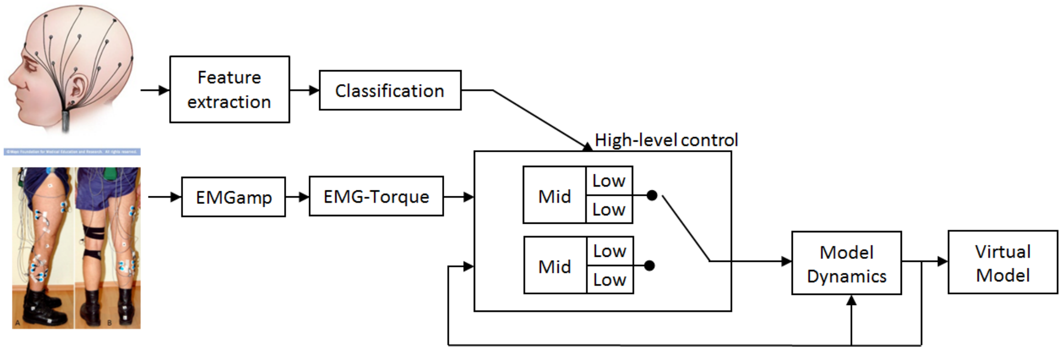


Figure 1.1: Proposed hBCI system.

1.5 Objectives

1.5.1 Main Objective

The main objective is to combine two existing non-invasive NMI systems, that will serve as inputs to a controller of a mathematical model of a prosthetic tibiofemoral joint, which will be represented in a virtual manner.

1.5.2 Particular Objectives

1. Develop a virtual model of a tibiofemoral joint. This is accomplished by making a kinematic and dynamic study of an actual tibiofemoral joint simplified in a two-link model, whose parameters are estimated from a human body and emulated in a 3D representation.

2. Design a multi-level control scheme to solve the trajectory tracking according to the different gait phases and speeds selected by an EEG classifier.

3. Implement the EEG and EMG signal processing to classify different activities and estimate the amplitude of the torque required by the model for a given activity.

1.6 Methodology

1. Literature survey.
2. Mathematical modeling of tibiofemoral joints.
3. Finite-state machine design for the model switching.
4. Segmental analysis to determine nominal parameters.
5. Computerized design of the tibiofemoral joints.
6. Feedback design for the automatic control of the tibiofemoral prosthesis to track the trajectory at different speeds.
7. Acquisition of EMG and EEG measurements during gait for different speeds.
8. Estimation of EMG signal amplitude and develop a scheme to estimate the EMG-torque relation.
9. Pre-processing of EEG signal for feature extraction using power spectral density (PSD) and using a support vector machine (SVM) for the classification of the different speed-walking activities.
10. Perform a set of numerical simulations to evaluate the performance of the hybrid scheme.

1.7 Thesis Contribution

With this thesis we provide a compilation of procedures that grants a hBCI in a computational manner. The contributions of this thesis are as follows:

1. Mathematical models of tibiofemoral joints with nominal parameters.
2. A multi-level control scheme to solve the tracking of the knee trajectory, and switching between mathematical models.
3. An algorithm for the estimation of the EMG-torque relation.
4. An algorithm for the user intent classification of gait during walking at different speeds.
5. A virtual interface in charge of emulating the behavior of the hBCI of a tibiofemoral joint with the proposed tools.

1.8 Thesis Content

The content of this study is organized in logical order as appropriate to explain the procedures used. Chapter 2 revolves around the models used in this study: first it is mentioned what method was used to obtain the kinematic and dynamic models required for the study; this chapter also covers the origins of the parameters used in the models, which derive from equations using a physiological body parameters; the chapter explains the visualization used for feedback, which include the main bones in a tibio-femoral joint. Chapter 3 covers the control area: it specifies the low-level controllers used, and the control scheme used for the transitions between models. Chapter 4 shows everything regarding the neural signals: it is explained what process is required to obtained a useful EMG signal, and the method to classify the mental process using the EEG. Chapter 5 shows the experimental setup followed by the results obtained of all the compilation of procedures, and the discussion about them. Finally, in Chapter 6 the conclusions are presented, as well as future work proposals.

Chapter 2

Models

The human knee is one of the most complex joints in the body. This is due to the set of different ranges of motion it possess. It is possible to model an accurate approximation of this complex joint, but aiming for a high complexity increases the number of equations required for its understanding. Hence this chapter, shows what comprises the knee joint, and how simple models of the knee were studied.

2.1 Anatomy of the knee

Between the hip and ankle joints exist four bones that are the femur, tibia, fibula and patella; which can be seen in Figure 2.1 (edited from [18]). The femur is the longest and strongest bone of the skeleton, and the segment of the body in which is located is named “thigh”. The leg segment has the tibia and the fibula bones. Lastly the patella or kneecap is in charge of protecting the knee. These bones form two types of joints: the patellofemoral joint, which is between the femur and the patella, and the tibiofemoral joint that involves the femur and the tibia. The fibula and femur have no contact with each other so they do not form a joint. Entirely, the knee joint has multiple tasks: it facilitates positions and movements of the body, aids in the conservation of momentum, and provides the necessary moments for activities involving the lower limb. Furthermore, the knee transmits loads during different activities, like sitting/standing, stair ascent/descent, and walking [19]. The complex activity of walking, can be described by understanding the human gait pattern or what is known as the gait cycle.

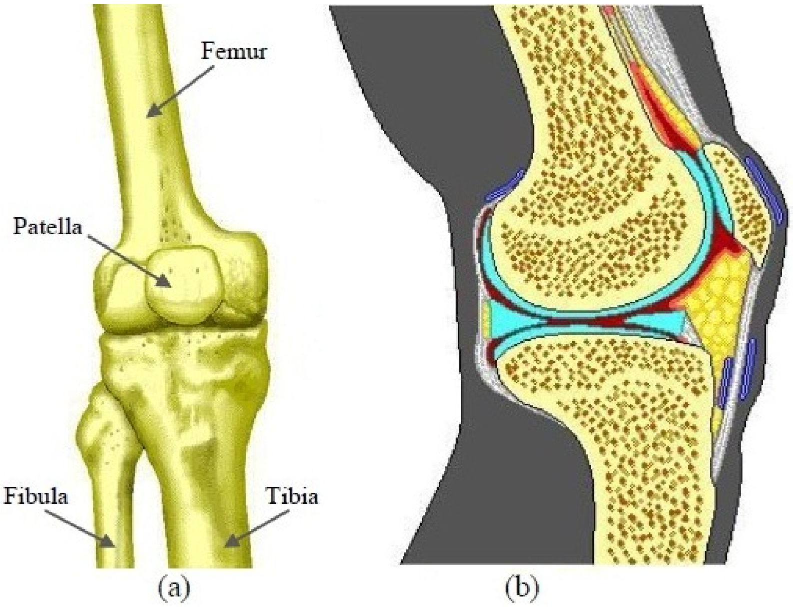


Figure 2.1: The skeletal (a) anterior and (b) transversal cut views of the knee.

2.2 Gait Cycle

The gait cycle consists of the repetition of certain events and phases. One gait cycle begins with heel contact of either foot and finishes with heel contact of that same foot. The cycle is divided in two phases: the stance phase and the swing phase. When one phase ends the other phase starts, making this way the cycle. Since phases and events repeat independently of time, the cycle is usually described in percentage, as seen in Figure 2.2 (edited from [20]).

The stance phase is the period when the foot is making contact with the surface, hence is the one providing support. It begins at 0% with the heel contact of either foot and concludes around 60% with toe-off of that foot. This phase is subdivided in four parts: heel contact, mid-stance, active propulsion, and passive propulsion. These four parts follow the sequence in which they were listed. Using the right foot as reference, right stance phase begins with heel contact of the right foot. At this time, there exists what is called initial double limb support, in which both feet are making contact with the surface.

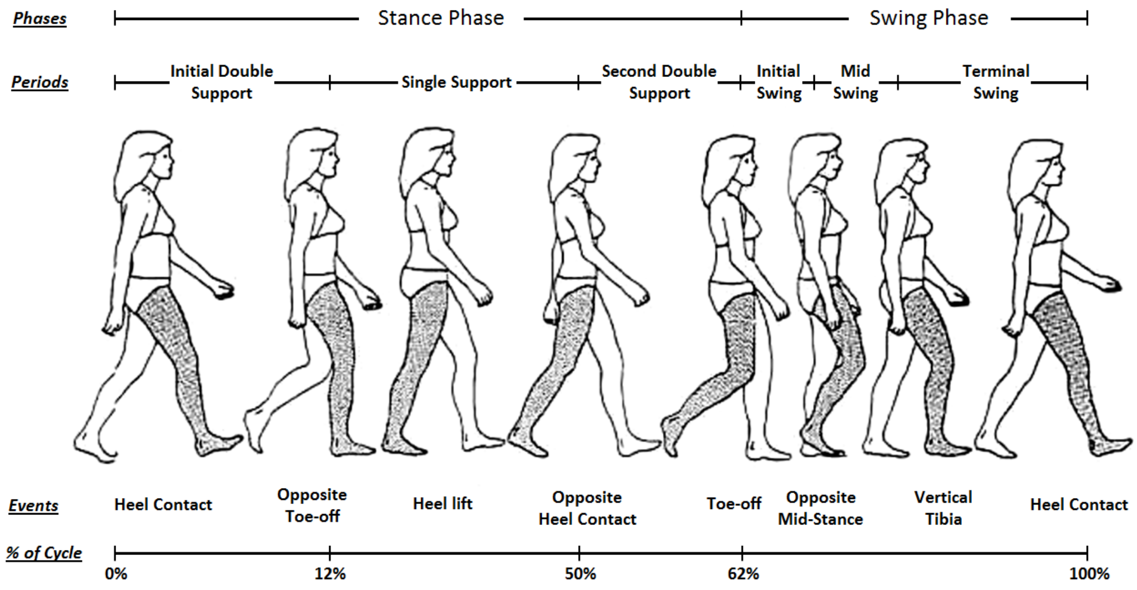


Figure 2.2: Gait cycle (edited from [20]).

When left toe-off occurs, this marks the end of the double support and the beginning of right mid-stance. Right mid-stance continues until there is a heel lift of that same foot. Right active propulsion occurs between heel lift of the right foot and heel contact of the left foot. During mid-stance and active propulsion there is just one limb on the ground, so single limb support takes action. After heel contact of the left foot, right passive propulsion begins, and since both feet are on the ground, there is the terminal double limb support. Finishing right passive propulsion with toe-off of the right foot, will also end the right stance phase.

The role of the swing phase consists on preparing the lower limb to begin the cycle again, taking around 40% of the remaining percentage. The swing phase is also subdivided into parts: initial swing, mid-swing, and terminal swing. Initial swing begins at the moment stance phase ended, at the event of toe-off of either foot. This will continue until mid-stance of the opposite foot, marking the beginning of mid-swing. When the tibia of the swinging limb reaches a vertical position, the mid-swing will end and the terminal swing will begin. During terminal swing, the swinging limb begins to make preparations for the heel contact of that foot, marking the end of the phase and one entire gait cycle.

During gait, the knee almost reaches its full extension at heel contact. When approaching mid-stance, the knee begins to flex reaching the peak of this phase. After mid-stance, during the active and passive propulsion, the knee is again almost completely extended. During the swing phase, the knee flexes in initial and mid-swing reaching the peak of the phase. To finish the gait cycle, the knee extends again to prepare the following heel contact in terminal swing. With this said, it can be noticed that the knee joint has angular displacements at these motion segments during walking. For these motions, a basic understanding of kinematics is required.

2.3 Kinematics

Kinematics is the branch of mechanics that describes the motion of points or objects without consideration of what causes it. In the body, kinematics describe the motion in three different planes, defined below and seen in Figure 2.3 (taken from [21]).

Coronal (or Frontal) Plane: plane that divides the body into front and back.

Sagittal (or Lateral) Plane: plane that divides the body into left and right sides.

Transverse (or Axial) Plane: plane splitting the body into upper and lower parts.

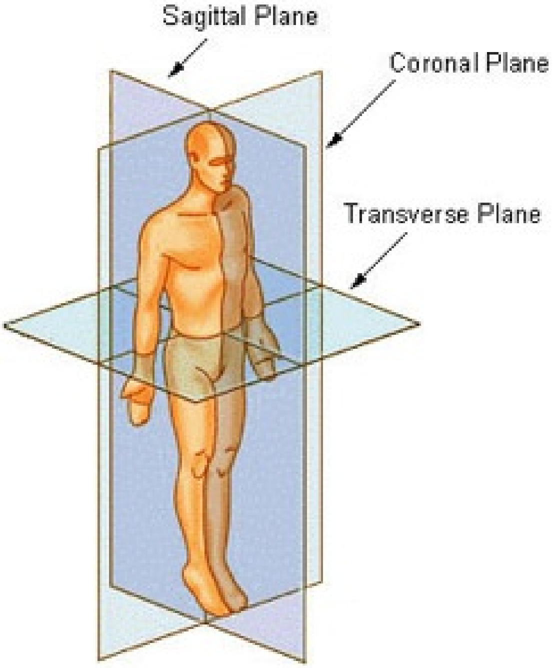


Figure 2.3: Anatomical planes.

2.3.1 Movement

The tibiofemoral joint in the knee shows movements in the three different planes mentioned previously for a combination of six-degrees of freedom: three translations (anterior-posterior, medial-lateral, proximal-distal), and three rotations (flexion-extension, internal-external, and varus-valgus or abduction-adduction). This can be seen in Figure 2.4 (taken from [19]).

Although there exist six-degrees of freedom in the tibiofemoral joint, most are limited to a few degrees of motion. The internal-external rotation reaches its maximum around 30° to 40° of flexion, meanwhile abduction-adduction has just a few degrees when reaching about 30° of flexion. With this, it can be seen that both motions depend on the flexion-extension on the sagittal plane, which has the greatest range of motion. During gait, the abduction-adduction and internal-external rotations present fluctuation of around 10° during full flexion [22]. So in this research, for simplicity, just the rotation in the sagittal plane is considered.

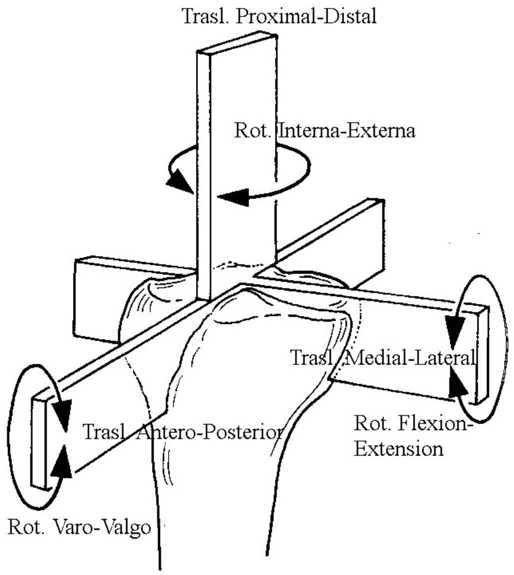


Figure 2.4: The axes and planes of rotation of the biological knee joint.

2.3.2 Forward Kinematics

A kinematic model is comprise of consecutive series of rigid bodies, or links, connected by joints. Forward kinematics specify the joint parameters and, using kinematic equations, calculates the composition of the chain and locates the position of the end-effector. Forward Kinematics finds the relations between the articular coordinates $q \in \mathbb{R}^n$ and Cartesian coordinates $[x, y, z] \in \mathbb{R}^3$ to find the location of end-effector [23].

As mentioned before, in this research is just covered the motion in the sagittal plane, and the tibiofemoral joint can be seen as a 2-link chain. However during the two main phases of gait (stance and swing phase) the tibiofemoral joint behaves differently. During stance phase, the foot is fixed to the ground and the purpose of the limb's movement is for the upper body advancement. In the swing phase, the entire lower limb moves forward with the hip as reference point. Since the behavior is different in the two phases, each has its own model (Figure 2.5). The parameters for each model in this research are supposed to come from the human body.

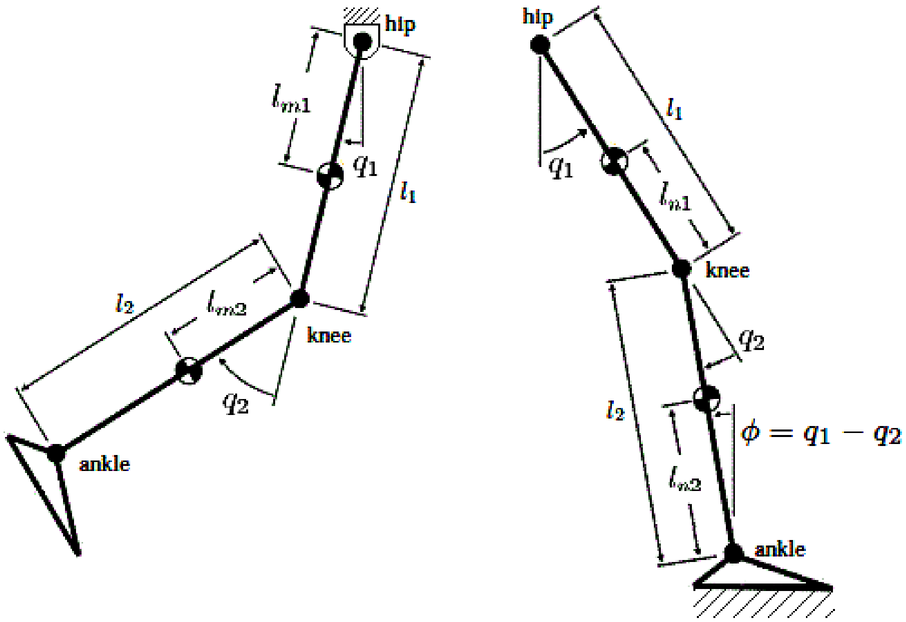


Figure 2.5: Swing model: two links in a double pendulum (left). Stance model: fixed to the ground (right).

Thus the forward kinematics model with respect to the fixed point and the centers of mass of each link for the stance phase is as follows:

$$\begin{bmatrix} x_{St1} \\ y_{St1} \end{bmatrix} = \begin{bmatrix} -l_{n2}\sin(\phi) \\ l_{n2}\cos(\phi) \end{bmatrix}, \quad \text{and} \tag{2.1}$$

$$\begin{bmatrix} x_{St2} \\ y_{St2} \end{bmatrix} = \begin{bmatrix} -l_2\sin(\phi) - l_{n1}\sin(\phi + q_2) \\ l_2\cos(\phi) + l_{n1}\cos(\phi + q_2) \end{bmatrix}. \tag{2.2}$$

For the swing phase, they are given by:

$$\begin{bmatrix} x_{Sw1} \\ y_{Sw1} \end{bmatrix} = \begin{bmatrix} l_{m1}\sin(q_1) \\ -l_{m1}\cos(q_1) \end{bmatrix}, \quad \text{and} \tag{2.3}$$

$$\begin{bmatrix} x_{Sw2} \\ y_{Sw2} \end{bmatrix} = \begin{bmatrix} l_1\sin(q_1) + l_{m2}\sin(q_1 - q_2) \\ -l_1\cos(q_1) - l_{m2}\cos(q_1 - q_2) \end{bmatrix}. \tag{2.4}$$

The parameters of the models are defined in Table 2.1:

Link	Phase	Meaning	Notation
Femur	Stance	Cartesian coordinates of center of mass	x_{St2}, y_{St2}
		Length of center of mass	l_{n1}
	Swing	Cartesian coordinates of center of mass	x_{Sw1}, y_{Sw1}
		Length of center of mass	l_{m1}
		Length of the link	l_1
Tibia	Stance	Cartesian coordinates of center of mass	x_{St1}, y_{St1}
		Length of center of mass	l_{n2}
	Swing	Cartesian coordinates of center of mass	x_{Sw2}, y_{Sw2}
		Length of center of mass	l_{m2}
		Length of the link	l_2
		Angular displacement in the hip (degrees)	q_1
		Angular displacement in the knee (degrees)	q_2
		Calculated angular displacement in the ankle where: $\phi = q_1 - q_2$	ϕ

Table 2.1: Parameters used in the models.

So far in this section the motions of the joint have been described, but it has been done without involving any forces or masses. Since these motions present accelerations and decelerations, there exist forces and moments acting on the body. For these reasons kinetics needs to be applied.

2.4 Dynamic Model

Kinetics is the branch of mechanics that studies the forces and torques that affect a body or object. Kinetics involves two studies: static analysis (during equilibrium), and dynamic analysis (during motion). Since the study in this research involves the walking gait, a dynamic analysis is required. A standard method to obtain the dynamic model is the Euler-Lagrange method, that matches the well-known robot dynamics form [23]:

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q), \tag{2.5}$$

where $q \in \mathbb{R}^n$ defines the generalized coordinates vector or articular position, $\dot{q} \in \mathbb{R}^n$ is the vector of articular velocities, $\ddot{q} \in \mathbb{R}^n$ is the vector of articular accelerations, $M(q) \in \mathbb{R}^{n\times n}$ is a symmetric positive definite inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ is the matrix of centripetal and Coriolis torques, $g(q) \in \mathbb{R}^n$ is the vector of gravitational torques obtained as the gradient of the robot potential energy $U(q)$, i.e., $g(q) = \frac{\partial U(q)}{\partial q}$, and n is the number of degrees of freedom of the model.

In the following subsections, the procedure to obtain the dynamic models used in stance phase and swing phase is shown.

2.4.1 Stance Model

To obtain the stance phase dynamic model, the process starts by using (2.1) and (2.2) from Section 2.3.2 and use differential kinematics to obtain the lineal velocity of the center of mass of each link:

$$v_{St1} = \frac{d}{dt}\begin{bmatrix} x_{St1} \\ y_{St1} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} -l_{n2}\sin(\phi) \\ l_{n2}\cos(\phi) \end{bmatrix} = \begin{bmatrix} -l_{n2}\cos(\phi)\dot{\phi} \\ -l_{n2}\sin(\phi)\dot{\phi} \end{bmatrix}, \quad \text{and} \tag{2.6}$$

$$\begin{aligned} v_{St2} &= \frac{d}{dt}\begin{bmatrix} x_{St2} \\ y_{St2} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} -l_2\sin(\phi) - l_{n1}\sin(\phi + q_2) \\ l_2\cos(\phi) + l_{n1}\cos(\phi + q_2) \end{bmatrix} \\ &= \begin{bmatrix} -l_2\cos(\phi)\dot{\phi} - l_{n1}\cos(\phi + q_2)\dot{\phi} & -l_{n1}\cos(\phi + q_2)\dot{q}_2 \\ -l_2\sin(\phi)\dot{\phi} - l_{n1}\sin(\phi + q_2)\dot{\phi} & -l_{n1}\sin(\phi + q_2)\dot{q}_2 \end{bmatrix}. \end{aligned} \tag{2.7}$$

With these equations, the respective speeds of each link are given by:

$$||v_{St1}||^2 = l_{n2}^2\dot{\phi}^2, \quad \text{and} \tag{2.8}$$

$$||v_{St2}||^2 = l_2^2\dot{\phi}^2 + 2l_2 l_{n1}\cos(q_2)(\dot{\phi} + \dot{q}_2)\dot{\phi} + l_{n1}^2(\dot{\phi} + \dot{q}_2)^2. \tag{2.9}$$

The next step consists on obtaining the kinetic and potential energies. The kinetic energy of a whole system is basically the summation of the kinetic energy of each link:

$$K_{St}(q, \dot{q}) = \frac{1}{2}m_2||v_{St1}^2|| + \frac{1}{2}I_2\dot{\phi}^2 + \frac{1}{2}m_1||v_{St2}^2|| + \frac{1}{2}I_1(\dot{\phi}_2 + \dot{q}_2)^2, \tag{2.10}$$

where the notation I_1 is for the inertia of the femur and I_2 for the tibia. As for the total of the potential energy of the system, it can be obtained as follows:

$$U_{St}(q) = m_2 g l_{n2}\cos(\phi) + m_1 g l_2\cos(\phi) + m_1 g l_{n1}\cos(\phi + q_2). \tag{2.11}$$

The Lagrangian $L(q, \dot{q})$ is defined by the difference between the kinetic and potential energy:

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q). \tag{2.12}$$

The movement equations of Euler-Lagrange of a model are given by:

$$\tau = \frac{d}{dt}\left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}}\right] - \frac{\partial L(q, \dot{q})}{\partial q}. \tag{2.13}$$

After proper calculations, found in Appendix A, the obtained dynamic model of the stance phase is given by:

$$\tau_{St} = M_{St}(q)\ddot{q} + C_{St}(q, \dot{q})\dot{q} + g_{St}(q), \tag{2.14}$$

with

$$M_{St} = \begin{bmatrix} m_2 l_{n2}^2 + m_1 l_2^2 + 2m_1 l_2 l_{n1} cos(q_2) + m_1 l_{n1}^2 + I_1 + I_2 & m_1 l_{n1}^2 + m_1 l_2 l_{n1} cos(q_2) + I_1 \\ m_1 l_{n1}^2 + m_1 l_2 l_{n1} cos(q_2) + I_1 & m_1 l_{n1}^2 + I_1 \end{bmatrix},$$

$$C_{St} = \begin{bmatrix} -2m_1 l_2 l_{n1} sin(q_2)\dot{q}_2 & -m_1 l_2 l_{n1} sin(q_2)\dot{q}_2 \\ m_1 l_2 l_{n1} sin(q_2)\dot{\phi} & 0 \end{bmatrix},$$

$$g_{St} = \begin{bmatrix} -m_2 g l_{n2} sin(\phi) - m_1 g l_2 sin(\phi) - m_1 g l_{n1} sin(\phi + q_2) \\ -m_1 g l_{n1} sin(\phi + q_2) \end{bmatrix},$$

where q, $\dot{q}$, $\ddot{q} \in \mathbb{R}^2$.

2.4.2 Swing Model

In a similar way, to obtain the swing phase dynamic model, the process starts by using (2.3) and (2.4) from Section 2.3.2, together with differential kinematics, to obtain the lineal velocity of the center of mass of each link:

$$v_{Sw1} = \frac{d}{d}\begin{bmatrix} x_{Sw1} \\ y_{Sw1} \end{bmatrix} = \frac{d}{dt}\begin{bmatrix} l_{m1}\sin(q_1) \\ -l_{m1}\cos(q_1) \end{bmatrix} = \begin{bmatrix} l_{m1}\cos(q_1)q_1 \\ l_{m1}\sin(q_1)q_1 \end{bmatrix}, \tag{2.15}$$

$$\begin{aligned} v_{Sw2} = \frac{d}{d}\begin{bmatrix} x_{Sw2} \\ y_{Sw2} \end{bmatrix} &= \frac{d}{d}\begin{bmatrix} l_1\sin(q_1) + l_{m2}\sin(q_1 - q_2) \\ -l_1\cos(q_1) - l_{m2}\cos(q_1 - q_2) \end{bmatrix} \\ &= \begin{bmatrix} l_1\cos(q_1)\dot{q}_1 + l_{m2}\cos(q_2 - q_1)\dot{q}_1 & -l_{m2}\cos(q_2 - q_1)\dot{q}_2 \\ l_1\sin(q_1)\dot{q}_1 - l_{m2}\sin(q_2 - q_1)\dot{q}_1 & l_{m2}\sin(q_2 - q_1)\dot{q}_2 \end{bmatrix}, \end{aligned} \tag{2.16}$$

where the speeds of each link are, respectively

$$||v_{Sw1}||^2 = l_{m1}^2\dot{q}_1^2, \quad \text{and} \tag{2.17}$$

$$||v_{Sw2}||^2 = l_1^2\dot{q}_1^2 + l_{m2}^2\dot{q}_1^2 + 2l_1l_{m2}\cos(q_2)\dot{q}_1^2 + l_{m2}^2\dot{q}_2^2 - 2l_1l_{m2}\cos(q_2)\dot{q}_1\dot{q}_2 - 2l_{m2}^2\dot{q}_1\dot{q}_2. \tag{2.18}$$

As in the previous subsection, the next step consists in obtaining the total kinetic and potential energies. The kinetic energy of each link and the potential energy are given by

$$K_{Sw}(q,\dot{q}) = \frac{1}{2}m_1||v_{Sw1}^2|| + \frac{1}{2}I_1\dot{q}_1^2 + \frac{1}{2}m_2||v_{Sw2}^2|| + \frac{1}{2}I_2(\dot{q}_1+\dot{q}_2)^2, \quad \text{and} \tag{2.19}$$

$$U_{Sw}(q) = m_1gl_{m1}(1-\cos(q_1)) + m_2gl_1(1-\cos(q_1)) + m_2gl_{m2}(1-\cos(q_2-q_1)), \tag{2.20}$$

respectively. Again, as in the stance phase, the Lagrangian in (2.12) and the Euler-Lagrange movement in (2.13) have to be used to calculate the dynamic model of the swing phase. After performing the calculations, which are available at Appendix A, the obtained dynamic model is given by

$$\tau_{Sw} = M_{Sw}(q)\ddot{q} + C_{Sw}(q,\dot{q})\dot{q} + g_{Sw}(q), \tag{2.21}$$

with

$$M_{Sw} = \begin{bmatrix} m_1l_{m1}^2 + m_2l_1^2 + m_2l_{m2}^2 + 2m_2l_1l_{m2}cos(q_2) + I_1 + I_2 & I_2 - m_2l_1l_{m2}cos(q_2) - m_2l_{m2}^2 \\ I_2 - m_2l_1l_{m2}cos(q_2) - m_2l_{m2}^2 & m_2l_{m2}^2 + I_2 \end{bmatrix},$$

$$C_{Sw} = \begin{bmatrix} -2m_2l_1l_{m2}sin(q_2)\dot{q}_2 & m_2l_1l_{m2}sin(q_2)\dot{q}_2 \\ m_2l_1l_{m2}sin(q_2)\dot{q}_1 & 0 \end{bmatrix},$$

$$g_{Sw} = \begin{bmatrix} m_1gl_{m1}sin(q_1) + m_2gl_1sin(q_1) - m_2gl_{m2}sin(q_2-q_1) \\ m_2gl_{m2}sin(q_2-q_1) \end{bmatrix},$$

where q, $\dot{q}$, $\ddot{q} \in \mathbb{R}^2$.

After calculating the dynamic model, a numerical model is necessary for simulation and practical purposes [23]. For this reason, the nominal values of the parameters for the rigid bodies or links are required, and the method to define them is found in the next section.

2.5 Parameters

There exist a procedure called segmental analysis [24] which considers the body as being composed of different rigid bodies or segments (Figure 2.6), and then it estimates the mass and center of mass (CM) location of each segment to calculate the corresponding CM of the whole body. It is shown in Table 2.2, how this analysis provides the calculation of segmental masses and CM locations. The masses are calculated after obtaining the segment weight, which comes from the use of the entire weight of the individual (F_w) in the given formula. The CM location is calculated based on a percentage of the length of the segment and it is allocated from a reference point or proximal end.

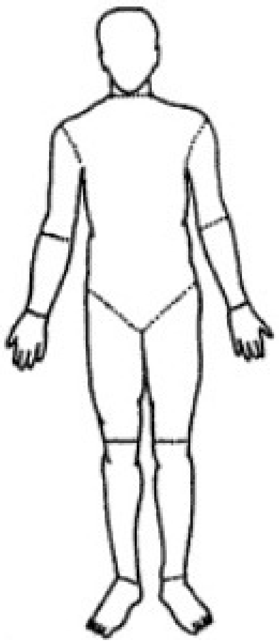


Figure 2.6: The body segment organization, which is comprised by 14 segments and includes the trunk, upper arms, forearms, hands, thighs, shanks, and feet.

Segment	Weight (N)	CM location (%)	Proximal end of segment
Thigh	$0.127F_w$-14.82	39.8	Hip joint
Shank	$0.044F_w$-1.75	41.3	Knee joint
Foot	$0.009F_w$+2.48	40.0	Heel

Table 2.2: Estimating body segment weights and locations of CM. Edited from [24].

In addition to CM location and mass calculations, also segmental moments of inertia need to be considered. The values of inertia are usually reported as mean values (Table 2.3) as there exists low variability between the different specimens studied.

Segment	Sagittal	Coronal	Transverse
Thigh	0.1157	0.1137	0.0224
Shank	0.0392	0.0391	0.0029
Foot	0.0030	0.0034	0.0007

Table 2.3: Segmental moments of inertia ($kg \cdot m^2$), edited from [24].

For this research, a test subject of height of 1.70 meters and a mass of 70 Kg is considered. Therefore, the nominal values of the parameters used for the numerical simulation are shown in Table 2.4. Note that this values can be adjusted to represent another different test subject.

Link	Parameter	Value
Femur	m_1	7.3793 Kg
	l_1	0.52 m
	l_{n1}	0.3130 m
	l_{m1}	0.2070 m
	I_1	0.1157

Link	Parameter	Value
Tibia	m_2	2.9016 Kg
	l_2	0.42 m
	l_{n2}	0.2352 m
	l_{m2}	0.1848 m
	I_2	0.0392

Table 2.4: Nominal parameters used in this research.

2.6 Interface

For a visual aid regarding the movement of the tibiofemoral joint, a 3D model (Figure 2.7) available online at the University of Brussels was used [25]. The reason behind the usage of an accurate modeling of the joint instead of simple lines on a grid is to allow a better understanding of the joint behavior, avoiding using a hinge conceptualization. The ultimate intention is to apply the control scheme, to be described in the next chapter, in a simulation environment using an anthropomorphic mechanism of the tibiofemoral joint. To manipulate this model, adjustments of its parameters had to be made. Nothing was done to compromise the structure of the model, solely the centers of rotation were changed. This was done so that the rotations were similar to that of the human tibiofemoral joint.

One of the features of this model is the possibility of changing the position of the different structures using simple geometry considerations as used in forward kinematics. This is due to the availability of using different input variables, making possible translations and rotations in all different axes. In chapter 5, this interface is integrated to the models for the understanding of the motion of the joint.

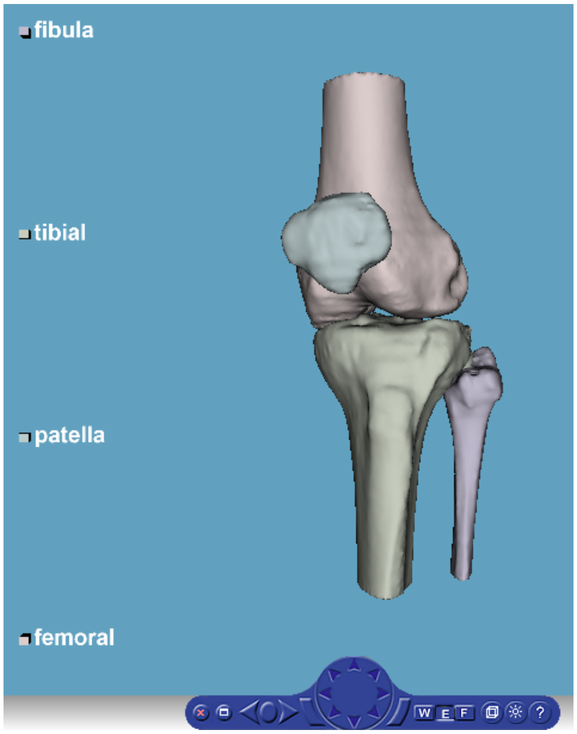


Figure 2.7: 3D virtual knee available in the VAKHUM project [25].

Chapter 3

Multi-level Control

In order to provide a user intent alongside proper control of the gait trajectories, here a multi-level control is proposed. It consists of three levels: low, mid and high. The high-level is used to recognize the user intent. The mid-level uses a finite state machine to distinguish between the two main phases of the gait. The low-level is composed by the trajectory controllers.

3.1 Low-Level

The control problem or control objective to accomplish is the tracking of the angular trajectories performed by a knee joint on a normal gait. To accomplish it, objective a PD+ controller was used, which is based on the position controller of Takegaki et al. [26]. This is the most simple and popular control scheme that can be used. The PD+ controller basically consists in a position controller with the addition of a dynamic compensation loop [27]. This approach has been used by Rune et al. [28] to control a rigid body, and by Ludan et al. [29] for a biped line walking robot.

3.1.1 PD+ Controller

The PD+ control with gravity compensation is an algorithm that includes proportional control of the position error $\tilde{q}$, proportional control of the velocity error $\dot{\tilde{q}}$ plus the complete robot dynamics. In the structure of this control scheme it is also involved the desired tracking trajectory, velocity and acceleration. The PD+

control is given by the following equation:

$$\tau_{pd+} = K_p\tilde{q} + K_v\dot{\tilde{q}} + M(q)\ddot{q}_d + C(q,\dot{q})\dot{q}_d + g(q), \tag{3.1}$$

where τ_{pd+} is the torque of the control signal; q, $\dot{q}$ and $\ddot{q}$ are the angular position velocity and acceleration; q_d, $\dot{q}_d$ and $\ddot{q}_d$ stand for the desired respective values. The errors are then defined as $\tilde{q} = q_d - q$ and $\dot{\tilde{q}} = \dot{q}_d - \dot{q}$. K_p and K_v are the proportional and derivative gains, respectively, and $M(\cdot)$, $C(\cdot)$, and $g(\cdot)$ are the same as in (2.5).

The implementation of the PD+ control scheme requires of the exact knowledge of the dynamic model of the robot, which means that the exact numeric values of its dynamic parameters are known [23]. Hence, taking the dynamic parameters from Section 2.4, we obtain the control equations for the stance and swing phases.

Figure 3.1 shows the diagram of the PD+ controller of the stance phase and its control equation is given by

$$\tau_{PD+St} = K_{St,p}\tilde{q} + K_{St,v}\dot{\tilde{q}} + M_{St}(q)\ddot{q}_d + C_{St}(q,\dot{q})\dot{q}_d + g_{St}(q), \tag{3.2}$$

where the subscript St stands for stance phase and the proportional and derivative gains have the following values:

$$K_{St,p} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \quad \text{and} \quad K_{St,v} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}.$$

Similarly, Figure 3.2 shows the controller for swing phase, and the control equation is as follows:

$$\tau_{PD+Sw} = K_{Sw,p}\tilde{q} + K_{Sw,v}\dot{\tilde{q}} + M_{Sw}(q)\ddot{q}_d + C_{Sw}(q,\dot{q})\dot{q}_d + g_{Sw}(q), \tag{3.3}$$

where the subscript Sw stands for swing phase and the proportional and derivative gains have the following values:

$$K_{Sw,p} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}, \quad \text{and} \quad K_{Sw,v} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}.$$

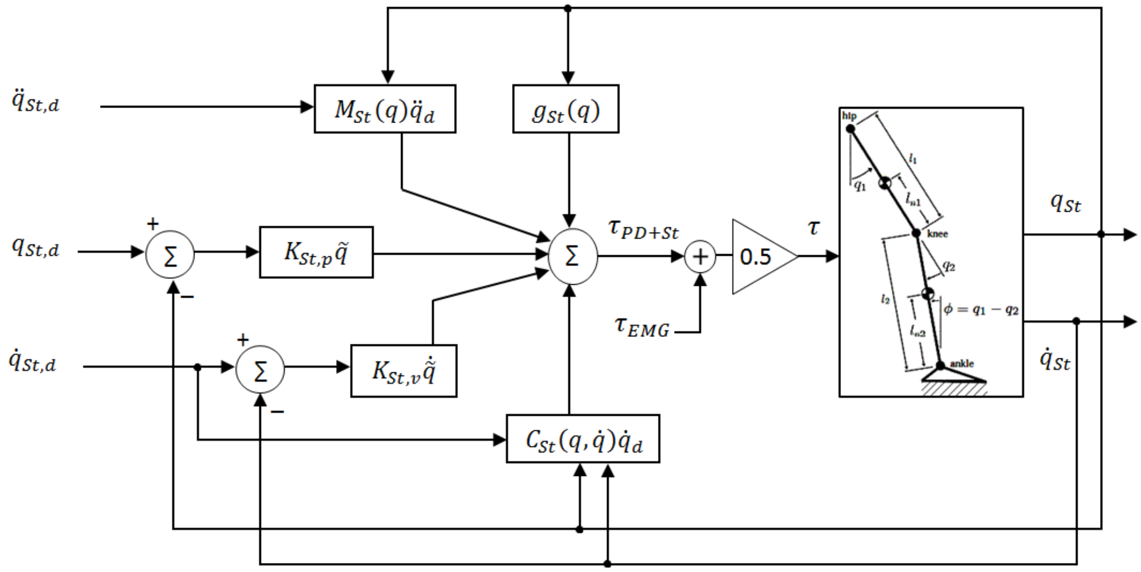


Figure 3.1: Block diagram of PD+ controller of the stance phase.

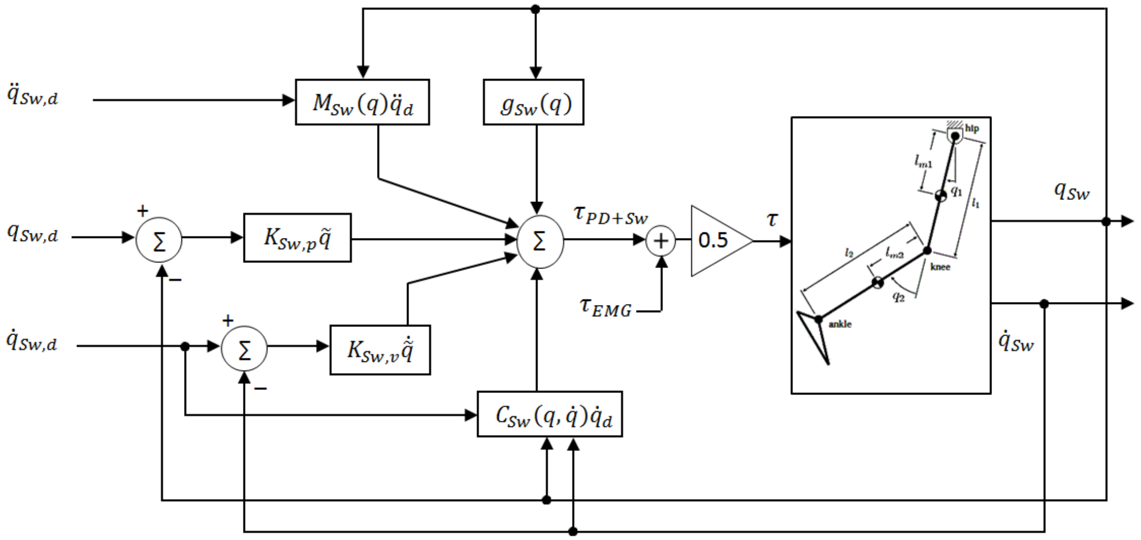


Figure 3.2: Block diagram of PD+ controller of the swing phase.

In Figures 3.1 and 3.2, it can be seen that the models' input τ is an average between the controller τ_{PD+} and the EMG-torque relation τ_{EMG}. Signal τ_{EMG} comes from the processing of the signal of the muscles using EMG seen in Chapter 4. This signal is the torque that the user is providing with his/her musclar activity. The average is taken because both the low-level controller and the EMG-torque relation are aiming for the same amount torque.

3.1.2 Equilibrium Point

The closed loop equation that relates the dynamic model (2.5) and the PD+ control scheme (3.1), expressed directly in state variables that define the trajectory control problem, is given by

$$\frac{d}{dt}\begin{bmatrix}\tilde{q}\\ \dot{\tilde{q}}\end{bmatrix} = \begin{bmatrix}\dot{\tilde{q}}\\ -M^{-1}(q)\left[K_p\tilde{q} + K_v\dot{\tilde{q}} + C(q,\dot{q})\dot{\tilde{q}}\right]\end{bmatrix} \tag{3.4}$$

$$= \begin{bmatrix}\dot{\tilde{q}}\\ -M^{-1}(q_d-\tilde{q})\left[K_p\tilde{q} + K_v\dot{\tilde{q}} + C(q_d-\tilde{q},\dot{q}_d-\dot{\tilde{q}})\dot{\tilde{q}}\right]\end{bmatrix}, \tag{3.5}$$

which results in a nonlinear differential equation of first order, non autonomous, and with an equilibrium point in the origin $\left[\tilde{q},\ \dot{\tilde{q}}\right]^T = 0\ \in \mathbb{R}^{2n}$. The analysis of existence and uniqueness of the equilibrium point $\left[\tilde{q},\ \dot{\tilde{q}}\right]^T = [0\ 0]^T$ is performed as follows [23]:

- Since the identity matrix $I \in \mathbb{R}^{n\times n}$ is a positive definite diagonal matrix, then $\dot{\tilde{q}} = I\dot{\tilde{q}} = 0 \Longleftrightarrow \dot{\tilde{q}} = 0$, for the first component of the closed loop in (3.4).
- For the second component of (3.4) the following considerations are taken:
  - Referring the property of the inertial effect, the inertia matrix $M(q) \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, i.e. $M(q) > 0$, and $M(q) = M^T(q)$. Because of this, the inverse matrix $M^{-1}(q) \in \mathbb{R}^{n\times n}$ exists and it is a symmetric positive definite matrix $\big(M^{-1}(q) > 0$, and $M^{-1}(q) = M^{-T}(q)\big)$.
  - By design, the proportional K_p and derivative K_v gains are positive definite matrices.

- Note that $C(q_d - \tilde{q}, \dot{q}_d - \dot{\tilde{q}})\dot{\tilde{q}} = 0 \in \mathbb{R}^n$ because the first component in (3.5) $\dot{\tilde{q}} = 0$.
- On the other hand, $K_p\tilde{q} = 0 \Leftrightarrow \tilde{q} = 0$, since the proportional gain is a positive definite matrix.

Thus, the equilibrium point in (3.5) exists and is unique, on the condition that the proportional and derivative gains of the PD+ control (3.1) are designed as positive definite matrices.

3.1.3 Stability

The demonstration of the stability, in the Lyapunov sense, of the equilibrium point in (3.5) is developed next.

Consider the following proposed candidate Lyapunov function:

$$V(t, \tilde{q}, \dot{\tilde{q}}) = \frac{1}{2}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{1}{2}\tilde{q}^T K_p \tilde{q}. \tag{3.6}$$

This is a positive definite function since the inertia and proportional gain matrices are positive definite. The temporal derivate of (3.6) takes the following form:

$$\dot{V}(t, \tilde{q}, \dot{\tilde{q}}) = \dot{\tilde{q}}^T M(q)\ddot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T \dot{M}(q)\dot{\tilde{q}} + \tilde{q}^T K_p \dot{\tilde{q}}. \tag{3.7}$$

Replacing the acceleration from the second component of (3.4) we get that:

$$\dot{V}(t, \tilde{q}, \dot{\tilde{q}}) = -\dot{\tilde{q}}^T K_p \tilde{q} - \dot{\tilde{q}}^T K_v \dot{\tilde{q}} - \dot{\tilde{q}}^T C(q, \dot{q})\dot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T \dot{M}(q)\dot{\tilde{q}} + \tilde{q}^T K_p \dot{\tilde{q}}. \tag{3.8}$$

Using the antisymmetric property, where the centripetal forces and Coriolis matrix $C(q, \dot{q})$, and the derivate in time of the inertia matrix $\dot{M}(q)$ satisfy

$$\frac{1}{2}\dot{q}\left[\dot{M}(q) - 2C(q, \dot{q})\right]\dot{q} \equiv 0,$$

i.e., the resulting matrix $\left[\dot{M}(q) - 2C(q, \dot{q})\right]$ is a antisymmetric matrix. Taking all the necessary considerations, the temporal derivate of the candidate function is given by

$$\dot{V}(t, \tilde{q}, \dot{\tilde{q}}) = -\dot{\tilde{q}}^T K_v \dot{\tilde{q}} \leq 0, \tag{3.9}$$

which satisfies the Lyapunov stability theorem, demonstrating the stability in the equilibrium point of (3.4).

The low-level controllers, that have been proved to have an equilibrium point and stability, are focused on the trajectories of each of the main phases, as seen in Figure 3.3. To distinguish between the two phases, another controller is required. Hence a mid-level controller is applied.

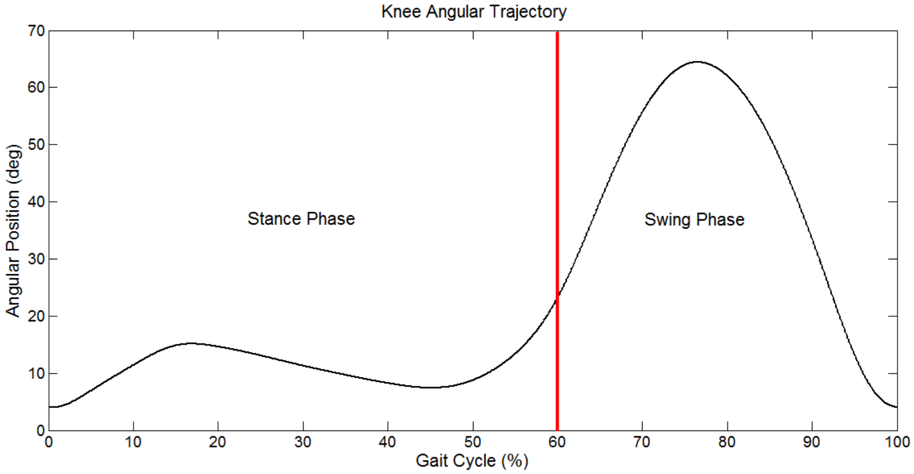


Figure 3.3: Knee angular trajectory, with marked transition point.

3.2 Mid-Level

As mentioned before the gate cycle consists of the repetition of phases. The duty of the mid-level is to switch the low-level controllers accordingly to the two main phases of gait: stance and swing phase. However, in each phase several conditions repeat, as seen in Figure 3.4. Therefore, a finite state machine was implemented. A finite-state machine (FSM), or simply a state machine, is a model used for sequential logics. It is sequential because it can be just in one of a finite number of states at each time. When certain condition is met, it switches from the current state to another one; this is known as transition. FSM has been thoroughly used in the literature, e.g. in Varol et al. [11] and Au et al. [30], who also apply it to walking and other activities.

Regarding the gait cycle, the conditions that repeat in each phase were analyzed and given to the corresponding state transitions (see Figure 3.5). The angular trajectory of the hip, $q1$, in the stance phase begins around 22° and starts to descend as the phase continues, reaching a peak near 50% of the cycle and finishing around 60% with a value of $-7.56°$. After the stance phase concludes, the swing phase commences. It crosses zero and reaches a peak before 90%, then descends a little to finish with the initial value of 22° for the stance phase to start anew, as seen in Figure 3.4.

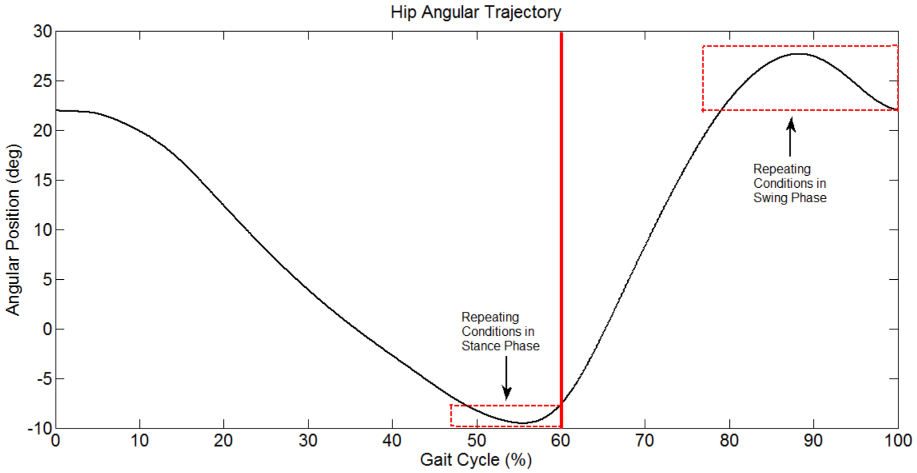


Figure 3.4: Angular trajectories of the hip.

It can be noted that the finishing conditions of both phases are repeated within each phase. For this reason, in this work two states are given to the stance phase and other two to the swing phase, as seen in Figure 3.5. The two states of the stance phase are named: St for "stance" and PSw for "pre-swing". The condition for the transition between these two states is to underpass the value of $-7.56°$, as it is the repeated finishing value of that main phase. In swing phase, the states are denoted as: Sw for "swing" and PSt for "pre-stance". Here, the condition is to overpass the repeated finishing value of the main phase, which in this particular case is 22°. (Note that these names have nothing to do with the previously mentioned gait phases, parts and events). A fifth state, called IDLE, was added. A default transition is placed in this state, to define it as the first state the system starts in.

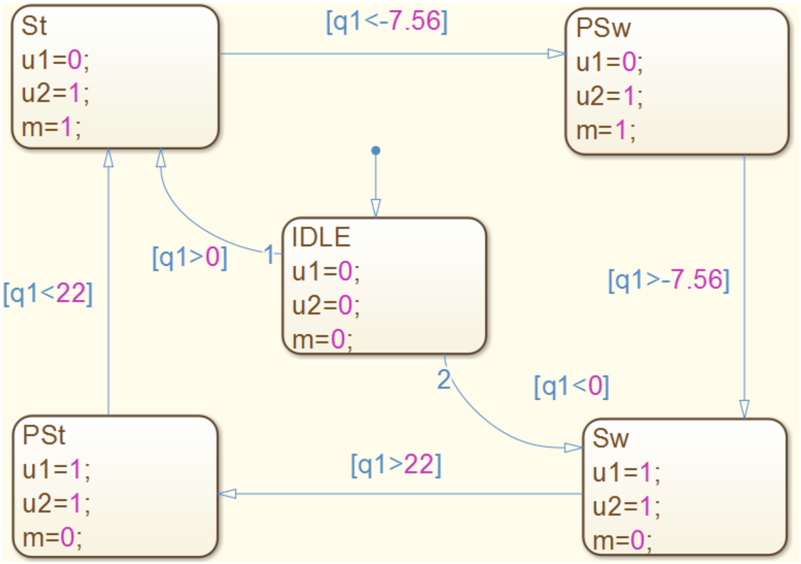


Figure 3.5: Finite state machine with its states and transitions.

In the FSM, each state has the output parameters, "m" that serves as a trigger for a specific block inside the models; and "u1" and "u2" that are specific logical values that serve as inputs to a logical operator. The chosen logical operator is the conditional construct (Figure 3.6) or if-then construct (sometimes called if-then-else). The operator evaluates the inputs and selects the appropriate statement. One of the statements of the logical operator is to give no movement, which would be the idle state. The other statements consist on choosing the low-level controller of the stance or swing phase.

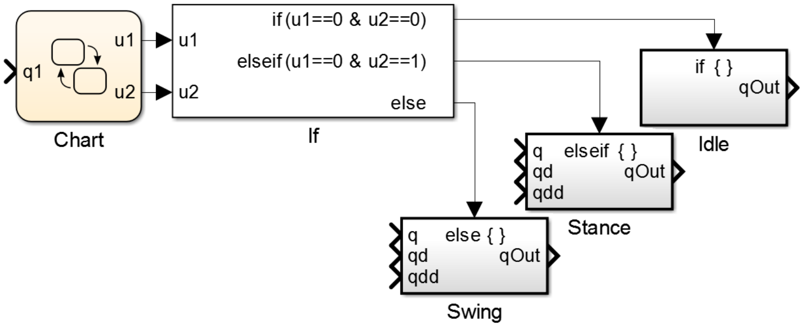


Figure 3.6: State machine and logical operator.

3.3 High-Level

The high-level control is composed primely of a support vector machine (SVM) that classifies the brain signals from the EEG into two categories. The categories consist of the activity of walking at different speeds. The SVM will be explained in Chapter 4. However, the purpose of this level of control will be explained next.

As it was mentioned in Section 2.2, the gait cycle is managed in terms of a percentage, therefore a defined trajectory of the tibiofemoral joint movement can be used. The defined trajectories are given in samples, not involving time, based on average angular displacements as defined in the literature. In this study, it is considered that the 100% of the gait cycle is completed in two sets of time, 1.13 and 1.43 seconds, as to establish the different walking speeds. In the next chapter, it is shown the classification of the activities using the SVM. After the classification, similar to the mid-level, a logical operator evaluating these results executes the different activities. The statements consist mainly in providing the defined trajectories required to have the two walking speeds.

Chapter 4

EEG & EMG Signals

This chapter is mainly focused on explaining the procedure and methods to handle the EEG and EMG signals. Each has its own acquisition method and pre-processing of the signal. This chapter also explains the SVM, which is used for the classification of the EEG to reproduce the intention of the user, as well as the neural network of maximum sensibility (NNMS) used to find the EMG-torque relationship that provides the magnitude of the user's motion.

4.1 Electroencephalography

EEG is the sensing of electrical signals from the brain using electrodes on the surface of the scalp. This approach is the most used when implementing BCI [31]. There exists other methods to acquire the neural activity of the brain. These other methods instead of electrical, rely on magnetic or metabolic activity. Nonetheless, these methods tend to be really expensive on the devices they use, work better for slow processes, or are not suitable for ambulatory conditions [32].

The location of the electrodes and their names are specified by the international 10-20 system for most medical, clinical or research applications, which use 19 recording electrodes. In some occasions less electrodes are used depending on the implementation. Also there exist the 10-10 and 10-5 systems that consider high-density arrays of sensors and can allow for more than 300 electrodes [33].

The electrical discharge that the electrodes record from the scalp lie withing the amplitudes of ±10 to ±100 μV. However, the brain activity is usually described in terms of different rhythms. The brain rhythms are divided in bands of frequencies that are called delta (δ), theta (θ), alpha (α), beta (β) and gamma (γ), listed from lower to higher frequencies [34]. Frequently, these bands are extracted from the EEG signal using spectral methods.

4.1.1 Feature Extraction

Feature extraction consists of extracting main characteristics from the raw data of some process, which they become a representation of the signal itself [31]. These characteristics are stored in what is known as *feature vector*. In our case, the feature vectors are the inputs to the classifiers, because they posses the most relevant information of the data. There exists several methods on feature extraction for EEG, like wavelet packet decomposition (WPD), approximate entropy (ApEn), discrete wavelet transform (DWT) or power spectral density (PSD), to mention some [31, 35].

In this research, the fast fourier transform (FFT) is used to obtain the frequency components of the EEG signal. Since the main interest here is in the gait, the FFT is used to find the frequency bands α and β, which are the frequencies that are related to motion, either real or imaginary [9, 17, 31]. The frequency band α has the range between $8 - 14$ Hz and β between $15 - 30$ Hz. After converting the signal from the time domain to the frequency domain, the range of the spectrum covering the α and β bands is taken to make up the feature vector. Depending on how many channels are decided to use on the EEG acquisition, the number of features will increase. Therefore, it is recommended to adjust the inputs of the classifier accordingly.

4.1.2 Task Classification

A SVM classifies data by finding an appropriate ρ-dimensional hyperplane that splits the data points into two classes. Originally, the basic idea of the SVM was to find the optimal hyperplane for linearly separable patterns, but often the data sets are not linearly separable. In order for a SVM to perform a non-linear classification, the non-linear decision hyperplane is introduced by mapping the inputs into a higher dimensional space using what is known as *kernel tricks.* This section shows how to form a linear SVM followed by a non-linear SVM [36, 37].

4.1.2.1 Linear SVM

Consider the training data $\{(x_i, y_i) \mid x_i \in \mathbb{R}^\rho,\ y_i \in \{1, -1\}\}_{i=1}^{N}$, where x_i is the ith input pattern, y_i is the respective category, which is either 1 or -1, and N is the total number of examples in the training set. It is desired to separate those categories with a hyperplane, which can be obtained with a set of data points that satisfy the following equation:

$$w \cdot x - b = 0, \tag{4.1}$$

where x is an input vector, w is an adjustable weight vector, $(\cdot)$ denotes the dot product, and b is a bias. The distance from the origin to the hyperplane is given by $\frac{b}{||w||}$ along w. Since the training data is linearly separable, its is possible to select two hyperplanes that separate the data with no data points in-between them.

These hyperplanes can be described by the following equations:

$$w \cdot x - b = +1, \quad \text{and} \tag{4.2}$$

$$w \cdot x - b = -1. \tag{4.3}$$

However, an infinite number of hyperplanes can be found. Hence, the SVM has to obtain an optimum hyperplane which maximizes the margin of separation between the two categories, as seen in the example of Figure 4.1.

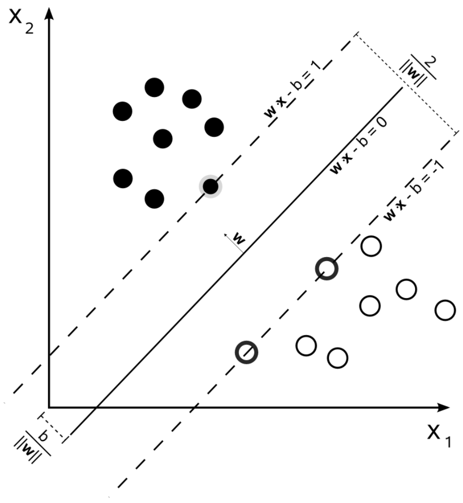


Figure 4.1: Optimum hyperplane with maximized margin of separation between two classes.

Let x_+ and x_- be data points that satisfy (4.2) and (4.3), respectively. These are called support vectors. Note that these equations imply that

$$w \cdot (x_+ - x_-) = 2. \tag{4.4}$$

Therefore, after normalization, the margin is given by

$$m = \frac{w \cdot (x_+ - x_-)}{||w||} = \frac{2}{||w||}. \tag{4.5}$$

Maximizing the margin is equivalent to minimizing $||w||$. As the data points (x_i, y_i) have to be prevented for falling in the margin, a constraint is required such that

$$w \cdot x_i - b \begin{cases} \geq +1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{cases},$$

which can be rewritten as

$$y_i(w \cdot x_i - b) \geq 1, \text{ for all } 1 \leq i \leq N. \tag{4.6}$$

The optimization problem of minimizing $||w||$ is difficult to solve because it depends on the norm of w, which involves a square root. By replacing $||w||$ with $\frac{1}{2}||w||^2$ the optimization problem is simplified without changing the solution, since both quantities are monotonically related [31]. This is known as the primal optimization problem that is formulated as

$$\min_{w} \frac{1}{2}||w||^2 \text{ subject to } y_i(w \cdot x_i - b) \geq 1, \forall i \tag{4.7}$$

This is a quadratic programing optimization problem subject to linear constraints with a unique minimum. It is common in optimization theory to convert from primal to dual form and then solve the latter instead. Using the Lagrangian multipliers $\alpha_i \geq 0$, the previous constrained problem can be expressed as

$$\begin{aligned} L_p &= \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i \left[y_i \left(w \cdot x_i - b\right) - 1\right] \\ &= \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i y_i \left(w \cdot x_i - b\right) + \sum_{i=1}^{N} \alpha_i \\ &= \frac{1}{2} w \cdot w - w \sum_{i=1}^{N} \alpha_i y_i x_i - b \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i . \end{aligned} \tag{4.8}$$

The optimal solution is a saddle point, which minimizes L_p by assigning zero to $\frac{\partial L_p}{\partial w}$ and $\frac{\partial L_p}{\partial b}$:

$$\begin{aligned} \frac{\partial L_p}{\partial w} = 0 \quad &\Rightarrow \quad w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \\ &\Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i y_i x_i \ , \end{aligned} \tag{4.9}$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0. \tag{4.10}$$

Substituting (4.9) and (4.10) into L_p, we obtain its dual form:

$$\begin{aligned} L_d &= \frac{1}{2} w \cdot w - w \sum_{i=1}^{N} \alpha_i y_i x_i + \sum_{i=1}^{N} \alpha_i \\ &= -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) + \sum_{i=1}^{N} \alpha_i \ , \end{aligned} \tag{4.11}$$

subject to: $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i \geq 0, \forall i$. The objective function of the dual problem is maximized in α_i, since L_d only depends on it. And although this also corresponds to a quadratic optimization problem, its complexity depends only on the sample size

n. By obtaining the optimum Lagrange multipliers, it is possible to compute the optimum weight vector through 4.9 entirely in terms of training data. To compute the optimum bias, the optimum weight vector is used in the following equation:

$$y_i(w{\cdot}x_i - b) = 1, \tag{4.12}$$

with $(x_i, y_i) \in SV$, where SV is a set of support vectors. Using (4.9) in (4.12), it can be found that

$$w \cdot x_i - b = \frac{1}{y_i} = y_i \Longrightarrow b = -y_i + \sum_{j=1}^{N} \alpha_j y_j x_j x_i. \tag{4.13}$$

The bias can be obtained using just one support vector. However, for numerical stability and to be more robust, an average of all γ_{SV} support vectors is used to compute the bias:

$$b = \frac{1}{\gamma_{SV}} \sum_{i=1}^{\gamma_{SV}} \left(-y_i + \sum_{j=1}^{N} \alpha_j y_j x_j x_i \right). \tag{4.14}$$

4.1.2.2 Non-linear SVM

To treat nonlinearly separable datasets, the SVM transforms the original input x_i using a nonlinear transformation or mapping, into a high-dimensional feature space. Applying a linear operator in a higher dimensional feature space is equivalent to apply a nonlinear operation into the input space. Let $\varphi(x_i)$ be a basis function that performs a nonlinear transformation, which then defines the kernel function $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$. In (4.15) it can be seen that the data points appear just as inner products:

$$\begin{aligned}
&\max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \right\} \\
&= \max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \right\} \\
&= \max_{\alpha} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \right\},
\end{aligned} \tag{4.15}$$

subject to $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i \geq 0, \forall i$. However, as long as the inner product in the feature space is calculated, the use of the kernel function to carry out φ explicitly is not required. This is known as the kernel trick. Some common kernel functions are shown in Table 4.1.

Type of Kernel	$K(x_i, x_j)$
Polynomial (homogeneous)	$(x_i \cdot x_j)^d$
Polynomial (non-homogeneous)	$(x_i \cdot x_j + 1)^d$
Gaussian radial basis function (rbf)	$exp\left(-\frac{\|\|x_i - x_j\|\|^2}{2\sigma^2}\right)$
Sigmoid	$\tanh(\kappa x_i \cdot x_j - c)$ for some $\kappa, c > 0$

Table 4.1: Common kernel functions [31,36,37].

4.1.2.3 EEG Classification

In Section 3.3, it was mentioned that the high-level controller consists solely of the SVM working as a classifier. The EEG signals are considered the input of the SVM, but as mentioned in Section 4.1.1, the signal needs to passed through the process of feature extraction. After such process, there exist different input signals, which then will be classified by the SVM into two specific tasks defined in this study as *speed 1* and *speed 2*.

4.2 Electromyography

EMG is the recording of the electrical activity produced by the muscles. It can be either invasive or non-invasive, which correspond to intramuscular EMG or surface EMG, respectively [38]. For the intramuscular EMG, a hollow needle with fine wires is inserted in the studied muscle. Although this method is more accurate regarding the muscular signal, several measurements have to be done because the needle insertion just gives a local picture of the muscle. Surface EMG on the contrary gives a general picture of the muscle. However, one has to be careful with the location of the sensors, and the inter-electrode spacing. If placed near the tendon origin, innervation zone, or where the muscle diameter is small, the signal will yield a lower amplitude. Along these lines, a sensor located on a set of muscles and having a large spacing between its electrodes, may detect cross-talk. [1] In this research, the non-invasive approach is used.

The acquisition of the signal is achieved by using surface electrodes. As mentioned before, there needs to be a reasonable spacing between them. These electrodes provide the signal to a differential amplifier, which removes a common signal and amplifies the differential between them. With the exception of the amplifier bandpass, an unprocessed and unfiltered signal is called raw EMG signal. In a surface EMG several phenomena can be appreciated, as seen in Figure 4.2 (edited from [38]). The magnitude of the signal usually has a range between ±1.5 mV, meanwhile the average baseline is around 1 or 2 μV. The bandwidth of the muscular signal is between 0 and 500 Hz. However, the range with more relevant power is between 10 to 250 Hz. As it can be seen, some frequencies are not taken in consideration because they are usually electrical noise. How to treat these noises and other considerations are shown in the subsection of EMG processing. Next, the required filters and the common established processing of the EMG signal are explained.

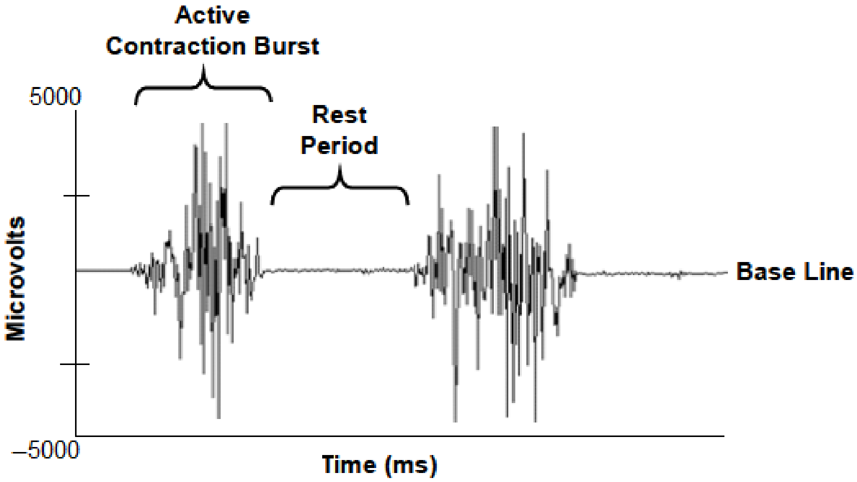


Figure 4.2: Example of a raw surface EMG signal.

[1] Cross-talk is misinterpreting the activity of the studied muscle by sensing another muscles in the proximity.

4.2.1 EMG Processing

The EMG signal, by itself, provides valuable information, but in its raw form it is not very useful. The signal has to be processed in order to obtain relevant information that can be used for scientific purposes. The common and simple method of processing an EMG signal consist of a rectification followed by a smoothening of the signal. There are other methods that have more steps, but the standard method of two steps is well accepted. In this research three steps are taken in consideration: the two well-known steps with an additional preliminary step of noise rejection.

4.2.1.1 Noise Rejection

The frequencies of interest range between 10 and 250 Hz. This is because between 0 and 20 Hz usually exists motion artifacts or some other types of noise [38, 39]. To eliminate these, noises a high-pass filter is used with a cutoff frequency between $10 - 20$ Hz. An in-between frequency is selected because cutting at 10 Hz shows good elimination of the noise, but raising the cutoff frequency increases the loss of relevant signal. For simplicity a high-pass Butterworth filter of order 5 is used. The Butterworth filter is one of the more basic of all electronic filters. It is designed to give the most possible plain response until the cut-frequency. The magnitude response function of the Butterworth low-pass filter has the following form:

$$|H_c(j\omega)|^2 = \frac{1}{1 + (\frac{j\omega}{j\omega_c})^{2r}}, \tag{4.16}$$

where ω is the frequency (rad/s), ω_c is the cutoff frequency (rad/s), and r is the order of the filter. The frequency response function of the Butterworth filter involves complex numbers since it is a function of $j\omega$. Thus, the magnitude-squared function is the product of the response function pairs $H_c(s)$ and $H_c(-s)$ (where $s = j\omega$):

$$H_c(s) \cdot H_c(-s) = \frac{1}{1 + (\frac{s}{j\omega_c})^{2r}}. \tag{4.17}$$

The poles of this expression occur on a circle of radius ω_c at equally spaced points. The transfer function itself will be specified just by the poles in the negative real half-plane of s. The k-th pole s is specified by

$$s_k = w_c e^{\frac{j(2k+r-1)\pi}{2r}}. \tag{4.18}$$

The transfer function may be written in terms of these poles as

$$H(s) = \frac{G_0}{\prod_{k=1}^{r}(s - s_k)/\omega_c}. \tag{4.19}$$

The denominator is a Butterworth polynomial in s, $B_r(s)$. The Butterworth polynomials may be written in complex form, but they are usually written with real coefficients by multiplying pole pairs which are complex conjugates, such as s_1 and s_r. The polynomials are normalized by setting $\omega_c = 1$. Therefore, the normalized Butterworth polynomials have the general form

$$B_r(s) = \prod_{k=1}^{\frac{r}{2}} \left[s^2 - 2s\cos\left(\frac{2k+r-1}{2r}\pi\right) + 1 \right], \tag{4.20}$$

for even order, and

$$B_r(s) = (s+1)\prod_{k=1}^{\frac{r}{2}} \left[s^2 - 2s\cos\left(\frac{2k+r-1}{2r}\pi\right) + 1 \right], \tag{4.21}$$

for odd order. The “low pass” behavior of these functions is driven by (ω/ω_c). As ω increases up to and over ω_c, the denominator of H_c becomes larger and H_c itself becomes smaller. It should not be too farfetched to attempt a magnitude response function with a high pass behavior in the following form:

$$|H_c(j\omega)| = \frac{1}{\sqrt{1 + (\frac{\omega_c}{\omega})^{2r}}}. \tag{4.22}$$

Therefore the design of a high-pass filter is usually achieved by designing a low-pass filter of the desired class, and then transforming the resulting filter by substituting the frequency-domain transfer function $H(s)$ with the relevant frequency transformation, which in this case is the low-pass to high-pass transformation, $s \rightarrow \frac{\omega_c}{s}$.

4.2.1.2 Rectification

The rectification translates the EMG signal to positive values. There exists two types of rectification: the half-wave rectification and the full-wave rectification.

The half-wave rectification deletes the EMG signal with negative values, thus losing some information. The full-wave rectification takes the negative values and converts them to positive. This last method is preferred because all the signal is conserved.

4.2.1.3 Linear Envelope

A smoothening process is done to quantify the energy of the muscle, its intensity or amplitude of the signal. To do that, the smoothening process changes the steep spikes of the signal into a “smooth” linear envelope. There exists three methods to perform the smoothening: two of them are classified as moving average, i.e. the mean-absolute-value (MAV) and root-mean-square (RMS) [38, 40], which are given by

$$MAV_t = \frac{1}{\sigma} \sum_{i=t-\sigma+1}^{t} |x_i| \,, \tag{4.23}$$

$$RMS_t = \sqrt{\frac{1}{\sigma} \sum_{i=t-\sigma+1}^{t} x_i^2} \,, \tag{4.24}$$

respectively. Where V is the length of the smoothening window, t is the initial time, and x_i is the signal being smoothed.

These methods include a signal rectification of their own, so the use of the rectification step is not required. In MAV, the absolute value of x_i is used. Meanwhile in RMS, the squared value is in charge of changing to positive the negative values. An appropriate length of the smoothening window has a range between 50 to 100 ms, for the study of both fast or slow activities.

Finally, there exists a simpler method which solely involves the usage of a low-pass filter, as explained in Section 4.2.1.1. It can be any type of the well-known common filters: Butterworth, Chebyshev I or II, Elliptic, Thompson or Bessel. Any of these methods can be used to smoothen the signal [41].

4.2.2 Torque Relation

Usually artificial neural networks are inspired from nature. This is because the artificial structure tries to emulate some natural function. There are many types of artificial neural networks, but the one used in this research was the NNMS. The NNMS is inspired by the biological theory of functional systems [42]. Basically this theory explains that there are several responses from a set of neural structures after a stimulus is applied. Some structures of the set become more excited than others, but usually just one gives a response. In some cases, a group of stimulated structures may contribute to the response. In both cases, this theory shows that the intensity to which every structure responds generates a differentiation.

4.2.2.1 Neural Network of Maximum Sensibility

The architecture of this neural network basically consists on two inner layers as seen in Figure 4.3. The first inner layer is in charge of detecting and classifying the stimulus that was propagated by the input layer. All the neurons on the first inner layer are fired, but just one that surpasses the sensibility margin generates a maximum sensibility state. After there is a preferred neuron, only that one gives the output response that would be saved in the second inner layer. If by any chance, there is no preferred neuron, meaning, non passed the sensibility margin, the contribution of all neurons is required to provide a response.

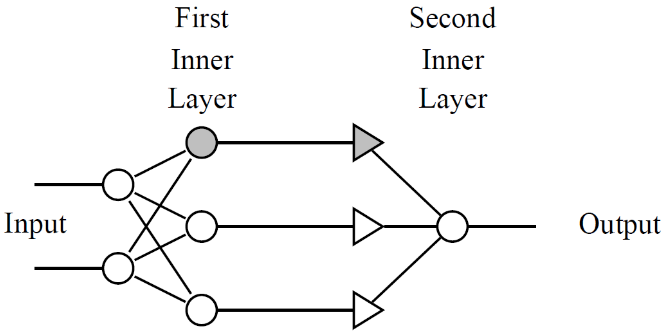


Figure 4.3: Architecture of Neural Network of Maximum Sensibility [42].

To implement the theory, the mathematical model of such architecture is required. The first inner layer has on all its neurons a Gaussian activation function (G) seen in the following equation:

$$G(x, \lambda, cm) = e^{-((x-cm)/\lambda)^2}, \tag{4.25}$$

where x is the input, λ is the sensibility control that is in charge of the width of G on each neuron, and cm is the mean of the Gaussian function. Usually, a value of $cm = 0$ is used on Gaussian activations functions in NNMS. The purpose of the activation function is to define the distance between the input signal and the weights of the neurons in the first layer, as seen in the following equation:

$$SN_{ne} = G\left(\sqrt{\sum\nolimits_{ne=1}^{Ne}(W_{ne,in} - X_i)^2}, \lambda, 0\right), \tag{4.26}$$

where SN is the activation value, $in = 1, 2, ..., I$ is the total of inputs and $ne = 1, 2, ..., Ne$ is the total of neurons.

The matrix W represents the weights on the first inner layer and vector X represents the i input nodes. Basically, all neurons have a certain degree of activation. However, a preferred neuron is selected by determining the most excited one. To determine it, the sensibility margin (sm) has to be surpassed and with this, the maximum sensibility state is also determined, i.e,

$$max\{SN_1, SN_2, SN_3, ..., SN_{Ne}\} > sm. \tag{4.27}$$

If a maximum sensibility state is provided by a single neuron, just one corresponding neuron of the second inner layer will provide information to the output neurons, i.e.,

$$O_j \leftarrow AC_{wn,j} * SN_{wn}, \tag{4.28}$$

where $j = 1, 2, ..., O$ is the total of outputs and wn is the winner neuron location. However, if no maximum sensibility state is reached, then all activated neurons contribute to form an output response:

$$O_j \leftarrow \frac{\sum_{ne=1}^{Ne}(AC_{ne,j} * SN_{ne})}{\sum_{ne=1}^{Ne} SN_{ne}}. \tag{4.29}$$

Each neuron possesses a “use weight” value that represents the usage or utility of the knowledge or information of the neuron. All neurons are set with an initial value and it decreases by a “forgetfulness” factor as the learning mechanism proceeds. This value, however, gets reset to the initial value if the neuron is activated. The previously mentioned learning mechanism is also defined by the maximum sensibility state and has two ways to learn. In the first way, if there is a maximum sensibility state, the winning neuron is destined to update its value to the average of the new input (X) and desired output (DO) patterns with the previously stored weights, i.e.,

$$W_{wn,in} \leftarrow (W_{wn,in} + X_{in})/2, \tag{4.30}$$

$$AC_{wn,j} \leftarrow (AC_{wn,j} + DO_j)/2, \tag{4.31}$$

where $in = 1, 2, ..., I$ is the total of inputs, $j = 1, 2, ..., O$ is the total of outputs and wn is winner neuron.

The other way of learning happens when the maximum sensibility state is off or not reached. Such event happens when the new information has a high distance from the weights of the input neurons. For this unknown information, a new category is created to represent this input signal. To do this, a new neuron with a low “use weight” value is selected to store the new values directly:

$$W_{nn,in} \leftarrow X_{in}, \tag{4.32}$$

$$AC_{nn,j} \leftarrow DO_j, \tag{4.33}$$

where $in = 1, 2, ..., I$ is the total of inputs, $j = 1, 2, ..., O$ is the total of outputs and nn is new neuron.

In Section 3.1.1 it was mentioned that the mathematical model of the tibio-femoral joint has an input that comes from an average between the torque of the controller and a torque from the EMG-torque relation. Such relation between the muscles and a given torque, in this particular research, is implemented by a NNMS. The following subsection covers how the NNMS is handled to achieve it.

4.2.2.2 EMG-Torque Relation

Three main parts are required in order to relate the EMG signal to the torque required by the model to execute the desired trajectories: the linear envelope of the EMG signal, the torque used by the model, and a system to relate both signals (in this case a NNMS). It was discussed in Subsection 4.2.1 how to obtain the linear envelope of the EMG signal. In this research, two linear envelopes are considered: one for the flexor muscle and another for the extensor muscle. Since these are muscular signals coming from the user, they become the inputs of the NNMS. The aim of the NNMS is to find the relation between the muscular signals and the required torque, therefore the torque is the output of the NNMS. The values of the inputs and outputs of the SVM, and the neural network used in this research are better explained in Chapter 5, where the results are shown along with their discussion.

Chapter 5

Experiment and Results

This chapter explains the experiment setup and shows the results of the procedures and simulations done for this research, which includes: the processing of the EEG neural signals and the classification using the SVM, the behavior of the low-level controller on each main phases of the gait cycle, and the torques that the models need to perform the desired trajectory. These are used in the EMG-torque relation, where the processing of the neuro-muscular signal is also discussed. Finally the integration of all these procedures is presented to form the complete hybrid scheme.

5.1 Experiment Setup

5.1.1 Procedure

A 25 years-old healthy participant, with no history of either neurological anomalies or lower limb pathology, was asked to walk in a XTERRA trail racer 3.0 treadmill at two different walking speeds. The first speed was settled according to what the participant defined as a comfortable walking speed before starting the acquisition of the data. An estimate of the speed in gait cycles is around 1.43 seconds per cycle. The second speed was also considered comfortable to the participant, with the difference of being a higher speed that yet was not considered jogging. This speed was around 1.13 seconds per cycle. The recording started after the subject established a comfortable pace at the first walking speed. Then, the procedure consisted in mainly two speed transitions. After walking one minute at the first speed, the subject was informed verbally of the change in speed. Then, the speed was gradually increased to the second walking speed, and as the speed was reached, the subject was asked to blink three times to mark a reference point in the EEG data. After a minute passed during the second walking speed, again the subject was informed of the gradual decrease in speed until reaching the previous speed, and the three time blinking for marking were recorded as well. The acquisition ended after another minute at the first speed had passed and after around 10 to 15 seconds of resting. This procedure was repeated for a set of 10 repetitions, with comfortable resting intervals for the subject.

5.1.2 EEG Recordings

The EEG neural signals were acquired using the B-Alert X10 system from Advanced Brain Monitoring Ltd. This device provides a wireless acquisition of EEG, electrooculographic (EOG) and electrocardiographic (ECG) signals. The system acquires nine channels of monopolar EEG recordings with a linked mastoid reference. The 10th channel is a programmable gain option that can be used for EOG or ECG. The EEG sensors are located at the frontal (Fz, F3 and F4), central (Cz, C3 and C4) and parietal-occipital (POz, P3, and P4) regions, as shown in Figure 5.1. Figure 5.2 shows the subject wearing the system. The headset collects the neural signals of the user with a sampling frequency of 256 Hz. More information about the system can be found in [43].

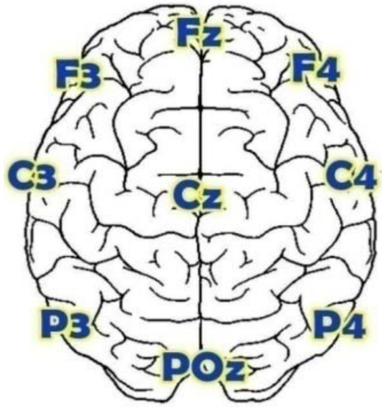
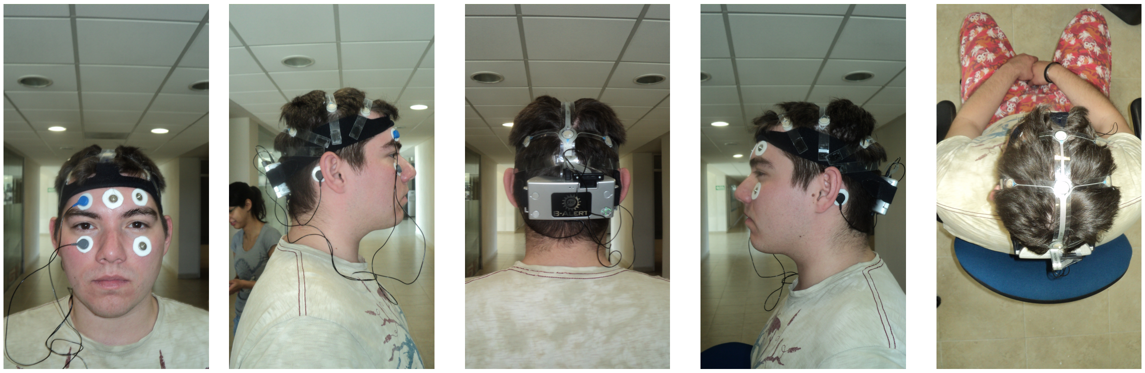


Figure 5.1: Position of the EEG electrodes [43].



(a) Frontal. (b) Right side. (c) Back side. (d) Left side. (e) Top view.

Figure 5.2: Subject wearing the B-Alert X10 sensor headset.

During the recording of the procedure, the usage of the EOG was used as reference of the recording. The marking blinks after reaching the different speeds in transitions determined the beginning and the end of the recording at a given speed. Also all nine channels were used during the recording, however just one main sensor was chosen for analysis, i.e. C3, since it is allocated in the area for real or imaginary movement of the right side of the body, which is of particular interest for the right knee movement. Figure 5.3, shows a recording of an experiment.

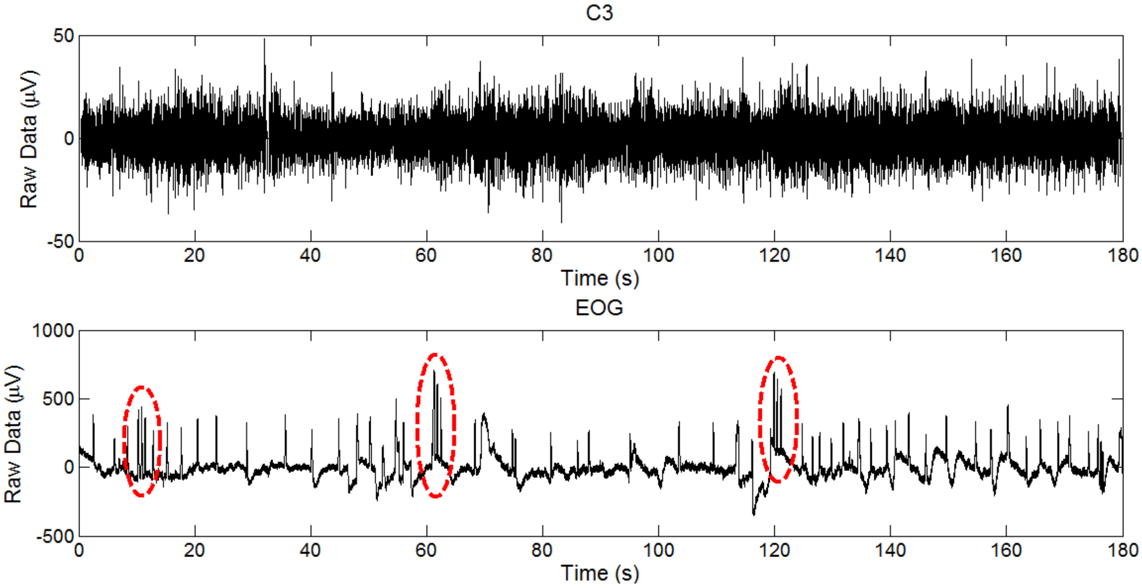
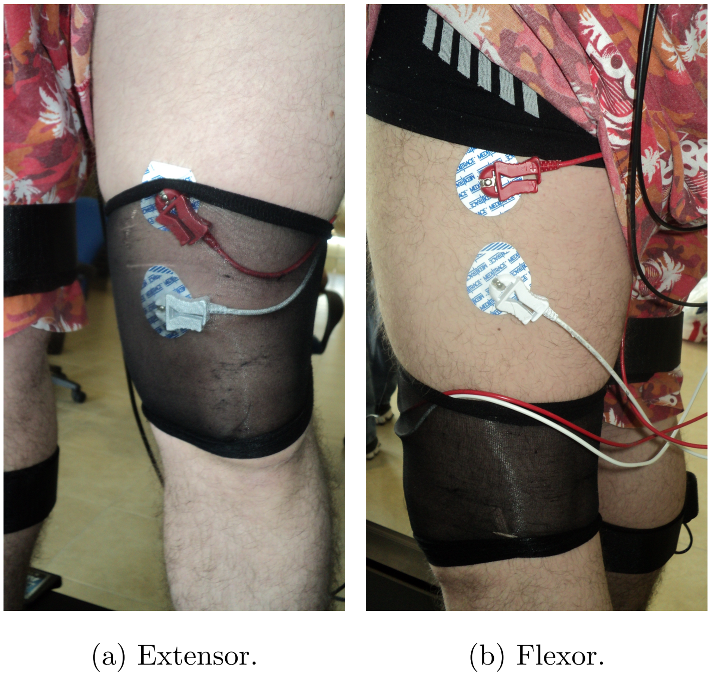


Figure 5.3: Recordings of sensor C3 (above) and EOG (below) with marking blinks.

5.1.3 EMG Recordings

The EMG signals were acquired using the MP36 system from BIOPAC. EL503 disposable electrodes were placed, in a flexor and extensor muscles. The chosen muscles were the quadriceps femoris for flexion and vastus medialis oblique for extension, as they have a fair activation power and period during the gait cycle [38]. Figure 5.4 shows the subject with electrodes placed over both muscles. More details of the hardware can be found in [44].



(a) Extensor. (b) Flexor.

Figure 5.4: Subject wearing the EL503 electrodes connected with pinch leads to the MP36 system.

The recording of the signals was performed using a sampling frequency of 1000 Hz. A recording of both muscles can be seen in Figure 5.5.

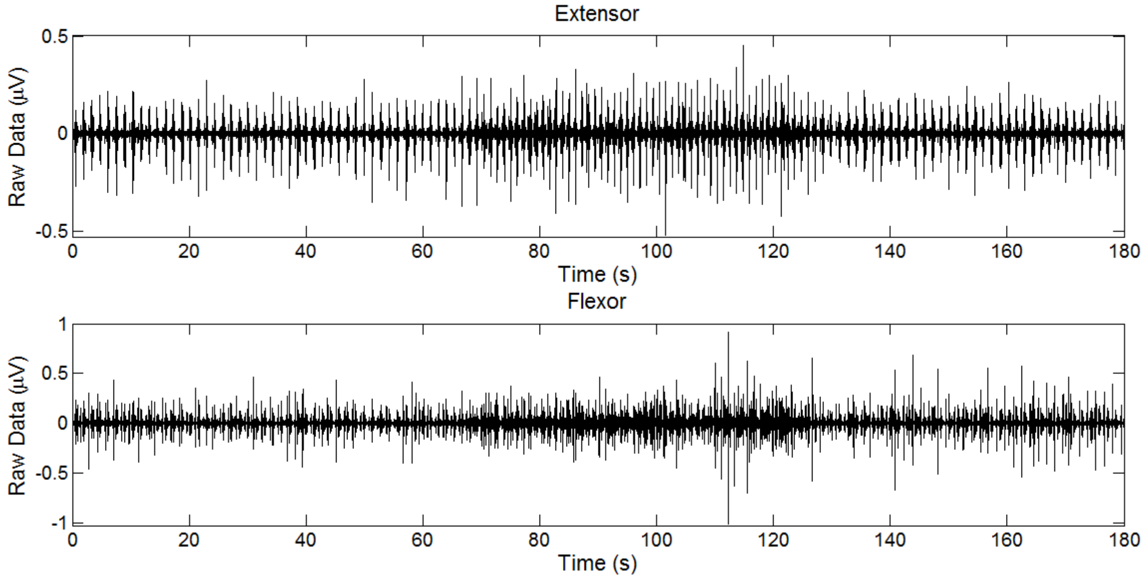


Figure 5.5: EMG recordings of extensor (above) and flexor (below) muscles.

5.2 Results

The hybrid scheme proposed in this study is presented in Figure 1.1, which is now repeated below to facilitate its view.

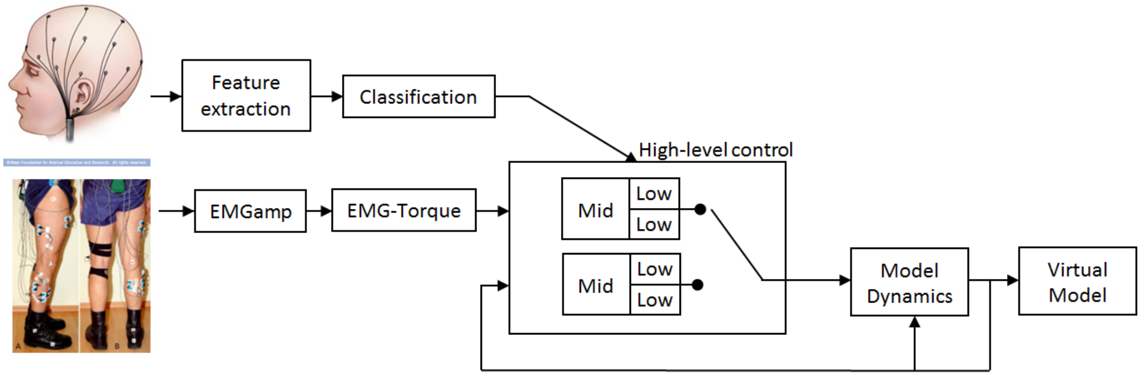


Figure 5.6: Proposed hBCI system.

The numerical implementation of all the procedures was realized using MATLAB® and Simulink®. Based to this approach, the results of each stage are presented in the following manner:

- **EEG:** preprocessing of the signal, feature extraction, and classification.
- **Controllers:** trajectories of stance and swing phase.
- **EMG:** preprocessing of the signal, linear envelope, and reaching EMG-torque relation.
- **Numerical implementation:** collective results of each previous stage to control a virtual tibio-femoral joint.

5.2.1 EEG

The preprocessing of the signal is performed by the B-Alert X10 system directly. It performs a bandpass filter with the high-pass cutoff frequency of 0.1 Hz and the low-pass cutoff frequency of 100 Hz. An example of a power spectrum can be seen in Figure 5.7.

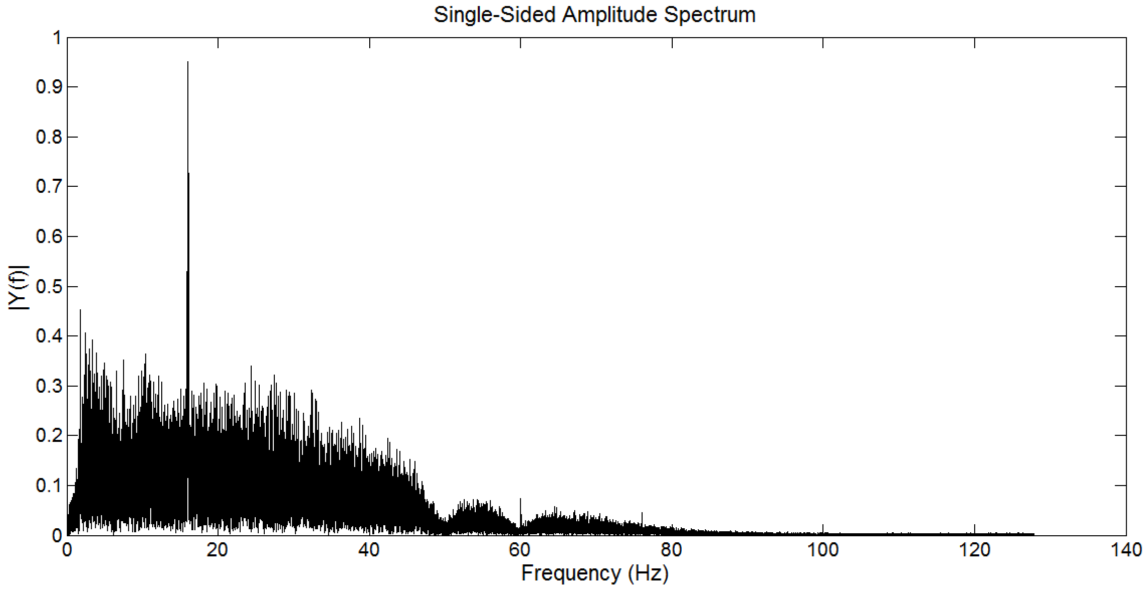


Figure 5.7: Example of Raw EEG power spectrum.

The EEG processing is performed in two sub-stages: the estimation of the power spectrum using the FFT, and its classification using the SVM. This exemplified in Figure 5.8, and a more descriptive diagram can be seen in Appendix B.

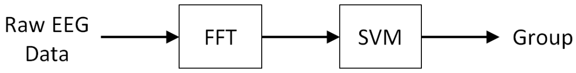


Figure 5.8: Diagram of EEG processing.

5.2.1.1 Feature Extraction

As mentioned in Subsection 4.1.1, a feature vector is created taking certain characteristics of the data. To extract the feature vectors from the raw EEG data, the signal was divided into non-overlapping windows of hundred samples each. This was performed through a *Buffer* Simulink block. For each window the power spectrum was estimated using the *FFT* block. The magnitude of the frequencies fitting the bands α and β were taken out of the spectrum of each window to form the feature vector consisting of 12 features. A code was used in a *MATLAB Function* block to take the respective frequencies. The coding and Simulink blocks can be seen in Figures B.1 and B.2. The power spectra of several windows of a signal are displayed in Figure 5.9 for the α and β bands.

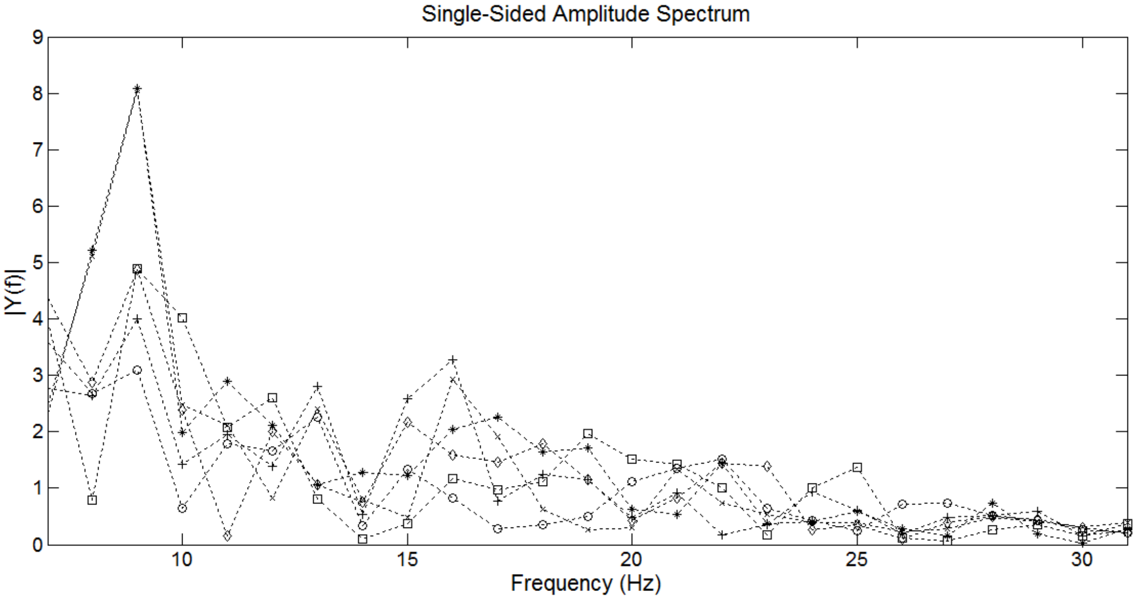


Figure 5.9: Power spectra of several windows of the signal.

5.2.1.2 Classification

Three data sets were selected from the 10 data sets from the experimental procedure. The selection was based on the easy appreciation of the blinking markers on the EOG signal. According to the feature extraction, using a hundred sample window, the three data sets generated a total of 1424 feature vectors. These vectors were used as the training data for the SVM. The training of the SVM was performed separately in the MATLAB command window, with the command *svmtrain*. This command generates the structure, *svmstruct*, that contains the information about the trained SVM classifier. Such structure was generated with the specification of the kernel function *rbf*, that stands for Gaussian radial basis function.

The classification of the data is performed by the MATLAB command *svmclassify*. The command classifies data using the information in the structure *svmstruct*. The command *svmclassify* was applied in the code of the *MATLAB Function* block in Simulink to perform the classification. The command can be seen in Figure B.2 in Appendix B. In Section 4.1.2 was mentioned that classification made by the SVM splits the data into two classes or groups, named 1 and -1. Here, group 1 belongs to *speed 1* and group -1 belongs to *speed 2*. The performance of the SVM with the three selected data sets can be seen in Figure 5.10, and the classification error is in Table 5.1.

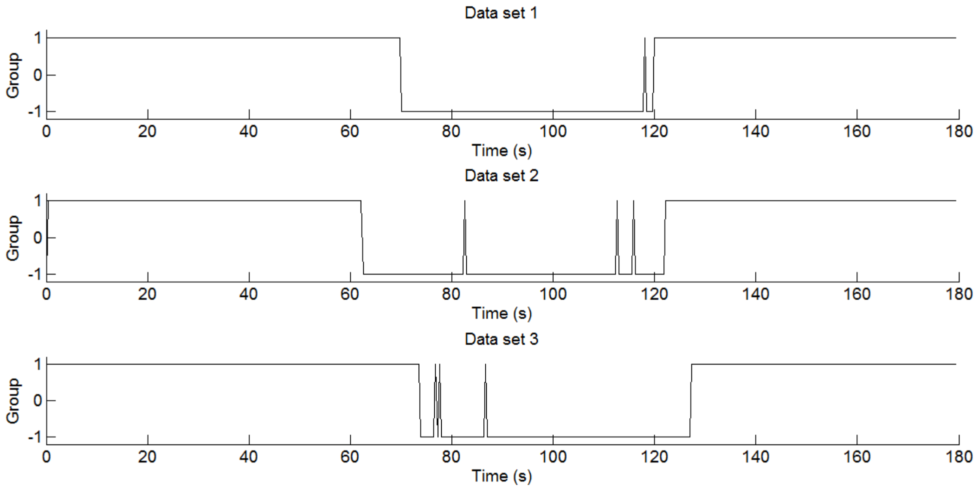


Figure 5.10: SVM Classification.

Experiment	Mean Square Error
1	0.0083
2	0.0356
3	0.0244

Table 5.1: Mean Square Error of Experiments 1, 2, and 3.

5.2.2 Controllers

In order to accomplish the control problem or control objective, which is the tracking of the angular trajectories, some terms have to be settled. Here, two low-level controllers were tested, one for the stance phase and another for the swing phase. The models and parameters of both phases are based on a lower limb from the proposed test subject with the specifications included in Table 2.4 from Section 2.5.

As previously mentioned in Section 2.4, the stance phase model is given by (2.14) and the control equation of such phase is given by (3.2), as mentioned in Section 3.1.1. Using those equations, parameters and gains, the closed-loop system can be made, as seen in Figure 5.11.

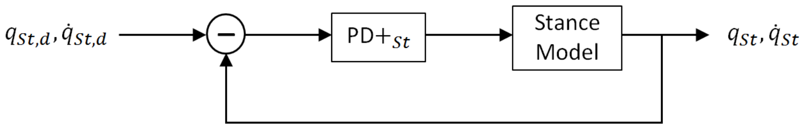


Figure 5.11: Stance phase closed-loop system diagram.

A more specific visualization of this closed-loop can be seen in Figure B.3, where the $PD+_{St}$ is composed of the proportional and derivative gains, and three *Interpreted MATLAB Function* blocks that have the required matrices from the model. Also, the stance phase model is composed of different Simulink blocks that implement (2.14), which provides the required outputs for the feedback of the closed-loop system. Figure 5.12 shows the result of the controller for the trajectory tracking, and Figure 5.13 shows the amount of torque required for the model to accomplish the desired trajectory.

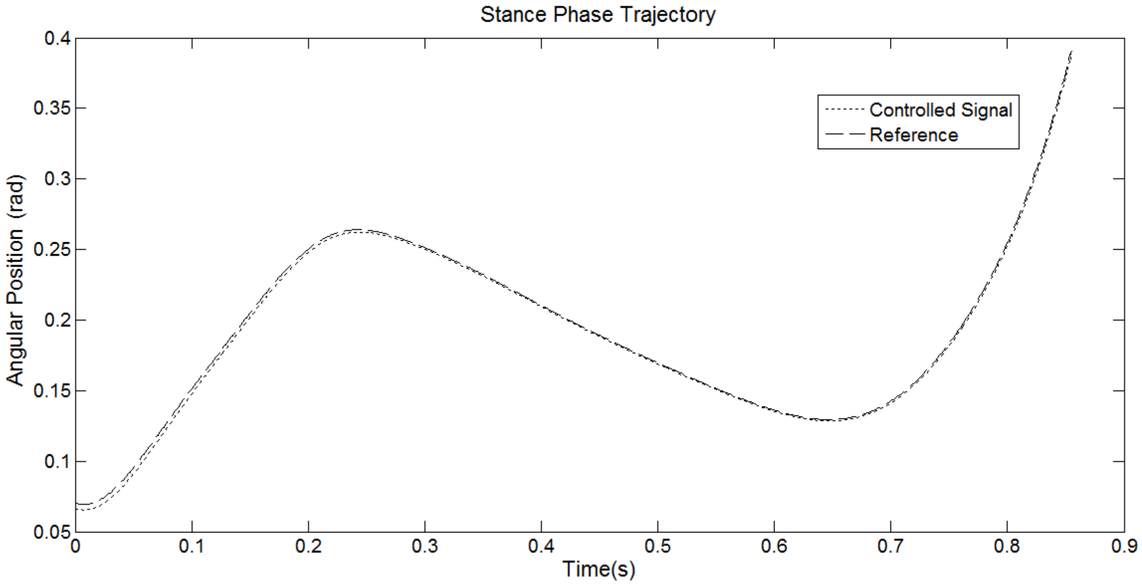


Figure 5.12: Trajectory tracking on the stance phase.

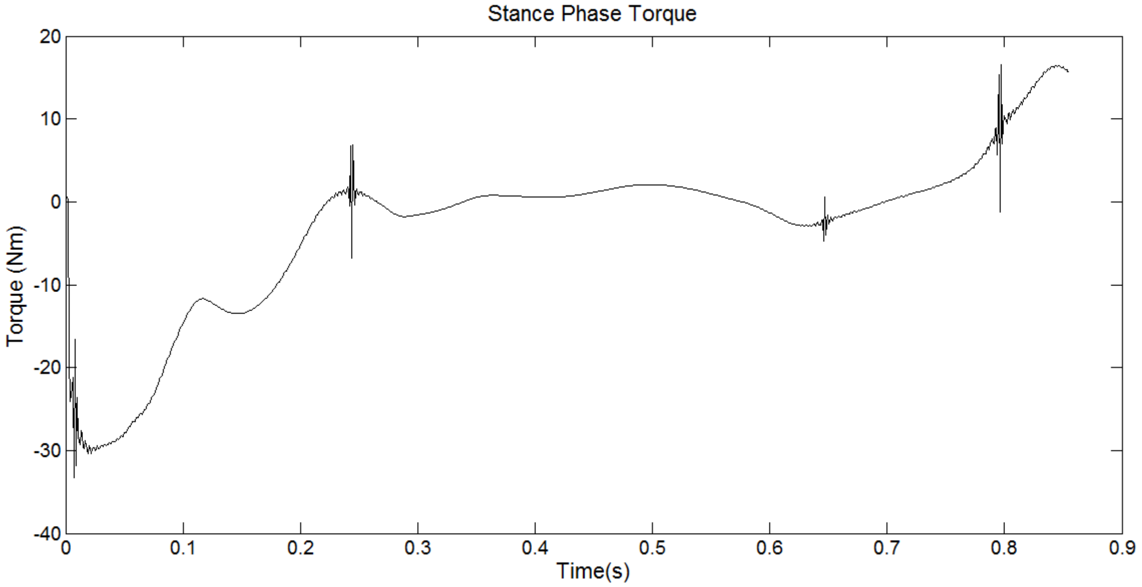


Figure 5.13: Required torque for stance phase.

In the same way, the swing phase model is given by (2.21) and the control equation of such phase is given by (3.3). Using those equations, parameters and gains, the closed-loop system can be calculated, as seen in Figure 5.14.

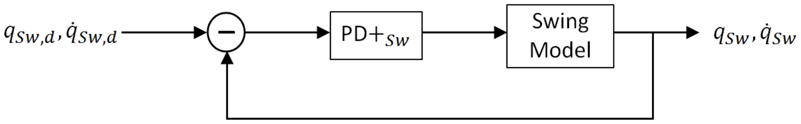


Figure 5.14: Swing phase closed-loop system diagram.

Similar to the stance phase closed-loop system, a more detail version for the swing system is found in Figure B.4. There, the $PD+_{Sw}$ and the swing phase model use the same types of Simulink blocks, but for the swing in (2.21) and (3.3). The result of the controller for the trajectory tracking can be seen in Figure 5.15. Furthermore, Figure 5.16 shows the amount of torque required for the model to accomplish the desired trajectory.

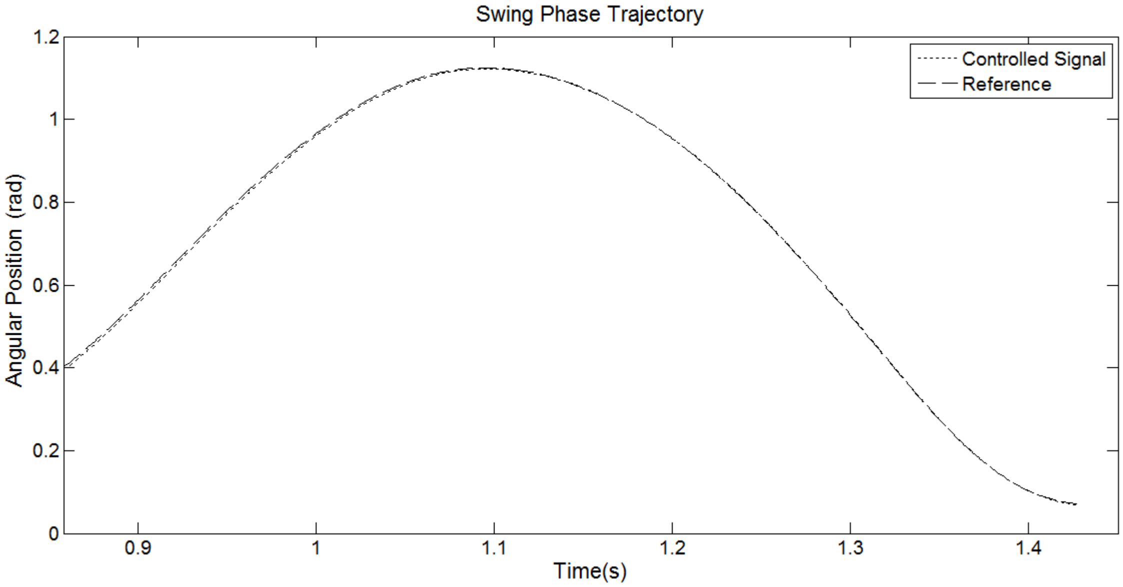


Figure 5.15: Trajectory tracking on the swing phase.

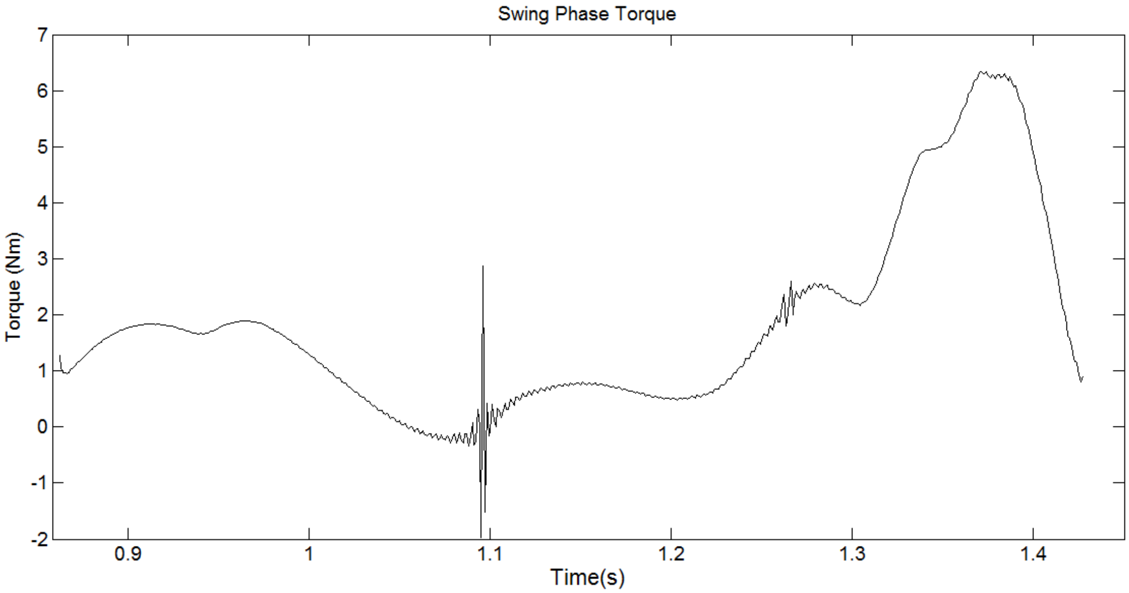


Figure 5.16: Required torque for swing phase.

5.2.3 EMG

As mentioned at the end of Subsection 3.1.1, the input of the models is τ. This input is composed of the average between the controller, τ_{PD+}, and the EMG-torque relation, τ_{EMG}. Obtaining τ_{EMG} requires the torques of both stance and swing phases, which are shown in Figures 5.13 and 5.16.

5.2.3.1 Linear Envelope

The steps described in Section 4.2.1 are applied to the acquired signal from Section 5.1.3. First, a noise rejection filter is used to remove the noise mainly from movement artifacts. The filter used was a high-pass type and is given by (4.22), where the sampling frequency w=1000 Hz, the cutoff frequency ω_c=15 Hz and the filter order n=5. These parameters were applied in the *Highpass Filter* Simulink block to perform the noise rejection. In Figure 5.17, it can be seen the noise rejection in the spectrum of the signal and the comparison between the filtered and raw signals.

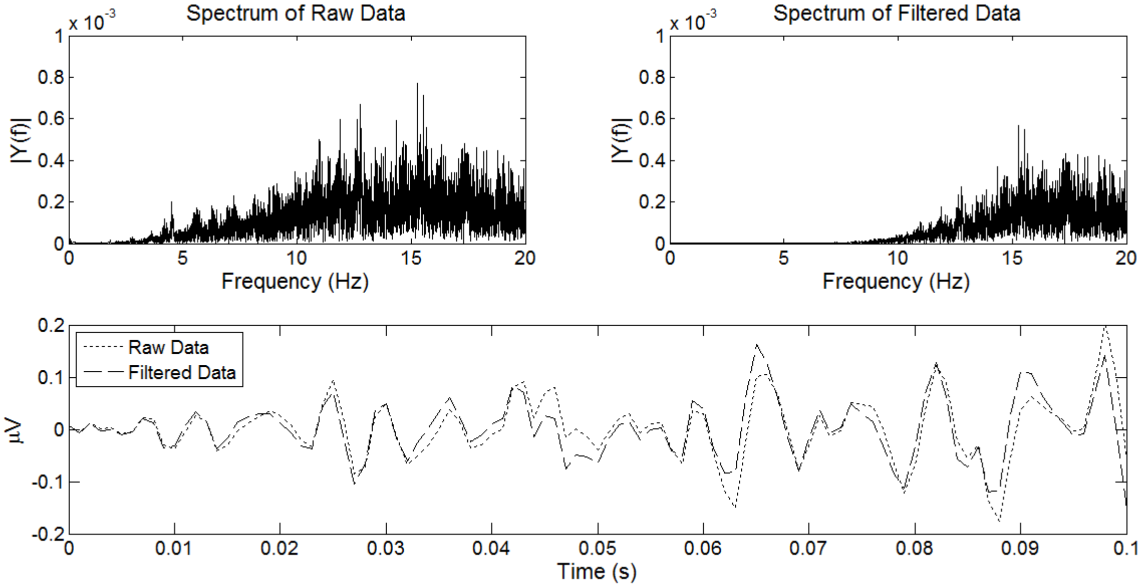


Figure 5.17: Raw EMG and filtered spectra (above). Post and pre-filter comparison (below).

The next step is to perform the simple task of converting all negative amplitudes of the signal to positive. This is achieved with the rectification of the filtered signal using the *Abs* Simulink block. The last step is to obtain the linear envelope with the rectified signal. In this research we chose the simplest between the several methods to obtain the envelope, that being the low-pass filter seen in (4.16), where the sampling frequency ω=1000 Hz, the ω_c=5 Hz and the filter order n=5. These parameters are applied in the *Lowpass Filter* Simulink block to obtain the linear envelope. This is shown in Figure 5.18, and the implementation of the mentioned blocks can be seen in Appendix B.5. The signal rectification and the linear envelopes for both muscles can be seen in Figure 5.19.

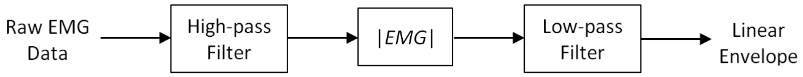


Figure 5.18: Diagram of EMG signal processing.

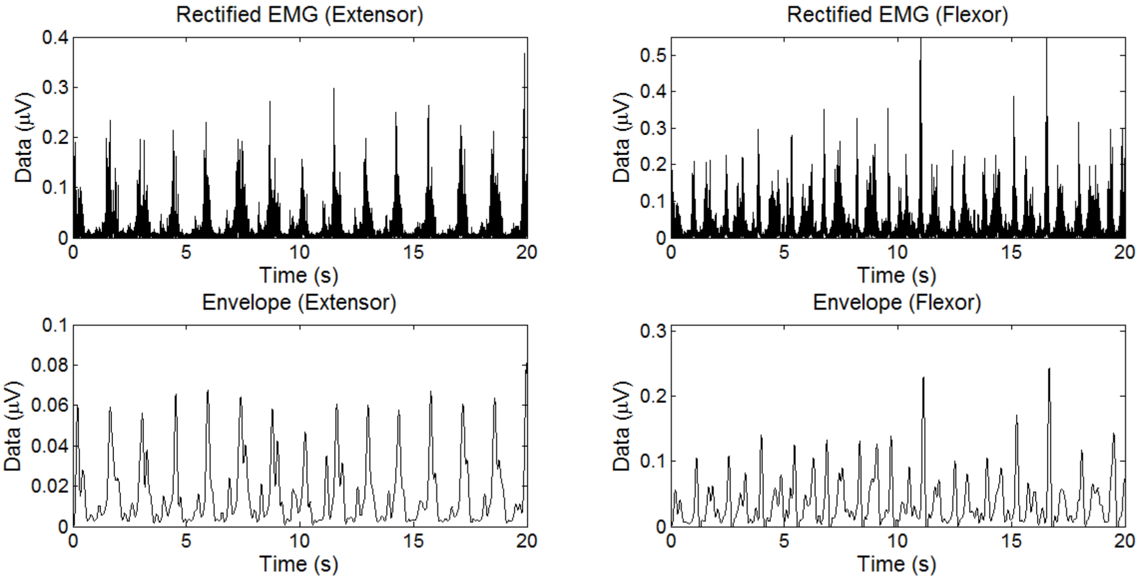


Figure 5.19: Rectified signals (above) and envelopes (below).

5.2.3.2 EMG-Torque Relation

It is mentioned in Subsection 4.2.2.2 the process in which a NNMS is used to find the mapping of the EMG linear envelopes to the corresponding amount of torque needed by the model. The training of the neural network was performed in MATLAB using the algorithm described in [42]. The pseudo-code can be found in Appendix B. The training of the neural network provides the parameters W and AC that are the weight and activation vectors respectively. Also it granted the values to de-normalize the output signal into the required torque range.

Different architectures can be used in a neural network, but the one chosen involved the least number of inputs regarding four different concepts. Thus using a single input from: the extensor signal, the flexor signal, a previous sample of the model's input, and a feedback of the network itself. The output of the neural network is the τ_{EMG} that takes part of the average to obtain the model's input, τ. The number of a hundred neurons was selected because it is considered a fair amount of neurons for this process. The parameters λ and sm were selected out of 81 manual tests where there were compared the different values of λ and sm, ranging from 0.1 to 0.9, for both parameters. The previous idea is the experimental design used to select the learning parameters. The selected λ and sm had the minimum mean square error of 5.0717×10^{-4}. The neural network's specifications are shown in Table 5.2.

Parameter	Value
# of Inputs	4
# of Outputs	1
# of Neurons	100
λ	0.3
sm	0.7

Table 5.2: Neural network specifications for EMG-torque relation.

In Figure 5.20, the general structure of this neural network is shown. In Appendix B the pseudo-code for the application of the NNMS is presented. The code is implemented through the *MATLAB Function* block. Figure 5.21 shows the resulting comparison between the desired torque (obtained from Section 5.2.2) and the best approximation done by the neural network.

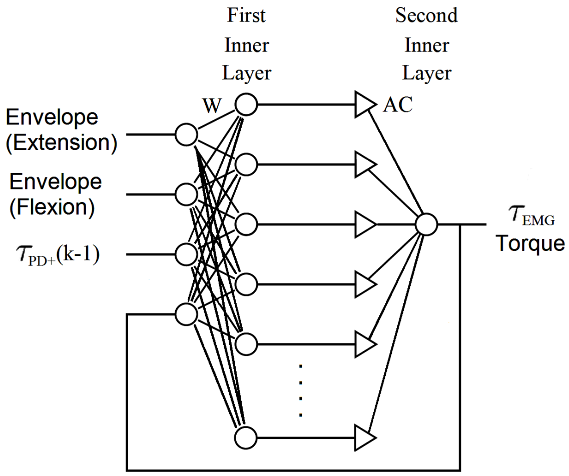


Figure 5.20: Structure of the NNMS to predict torque approximation.

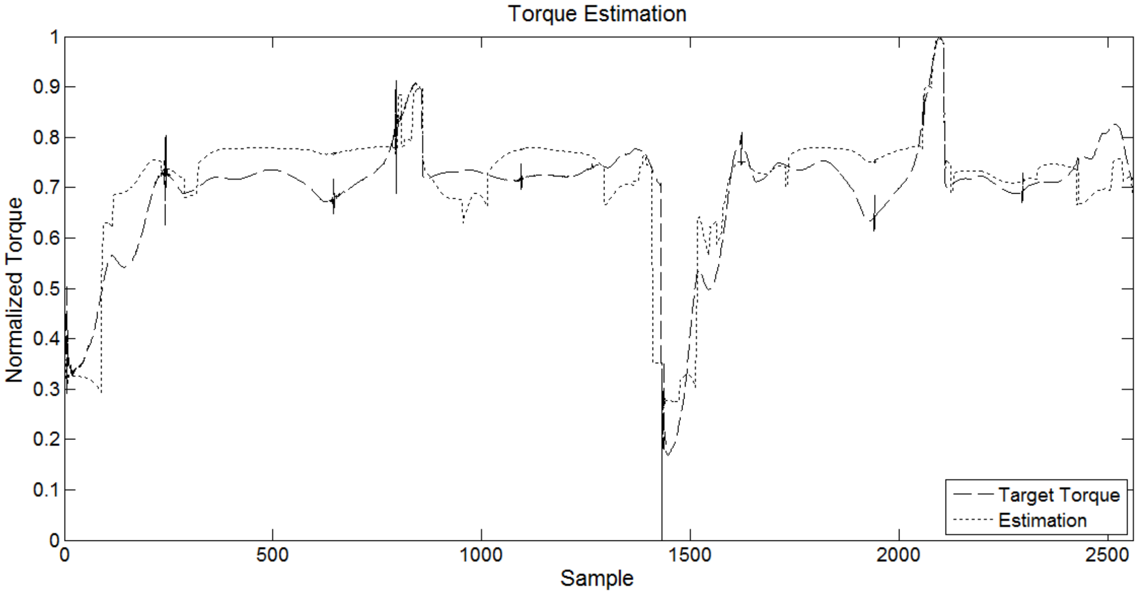


Figure 5.21: EMG-torque relations for both speeds in transition.

5.2.4 hBCI to control a virtual joint

Based on our hypothesis, we created a multi-level controller for the recorded signals that provides a time progressive signal instead of a simplified precise operation. Figure 5.22 shows the diagram of the hBCI. Such diagram corresponds to the integration of all previous stages: it has the processing of EEG and EMG signals, the controllers of the set of models, and it ends with the visualization of the tibio-femoral joint using the 3D model from [25].

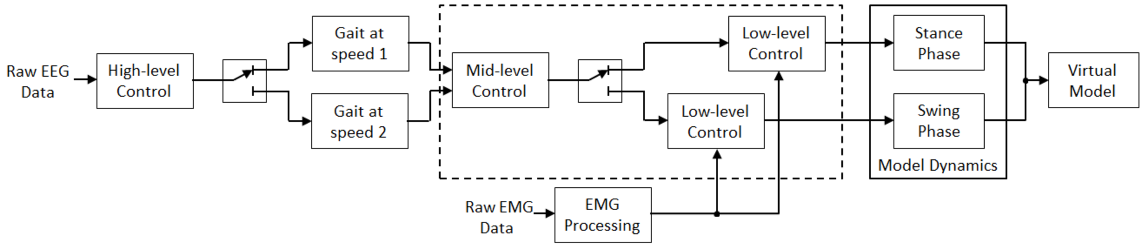
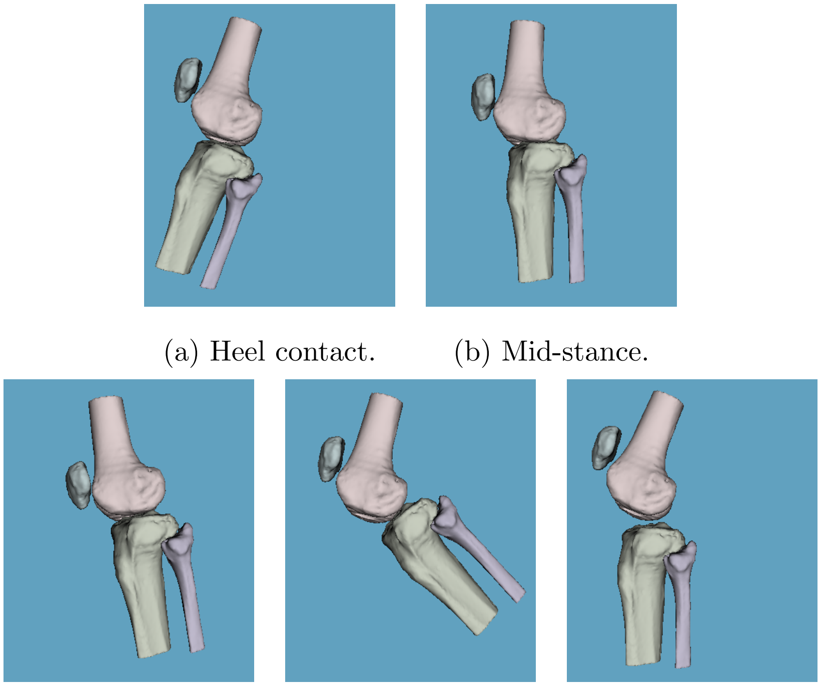


Figure 5.22: hBCI system.

The resulting trajectory of all the hBCI integration is shown next. Figure 5.24 shows the trajectory of a gait cycle along with the tracking performed by the low-level controllers. This Figure can be related to the different events in Figures 5.23 and 5.25.



(a) Heel contact. (b) Mid-stance.

(c) Initial swing. (d) Mid-swing. (e) Terminal swing.

Figure 5.23: 3D model of knee sagittal view.

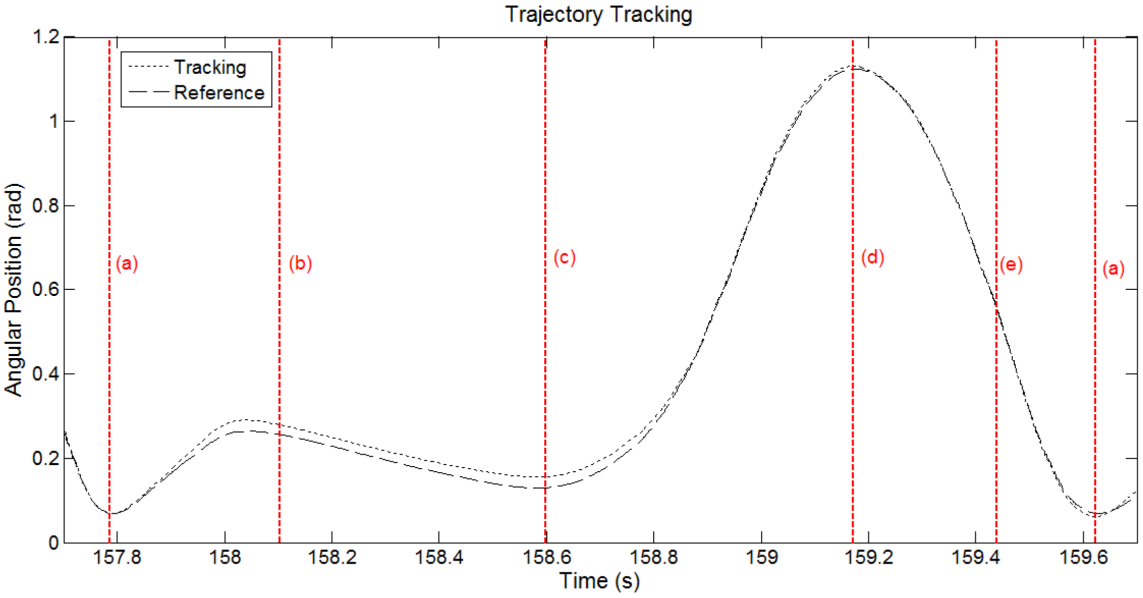
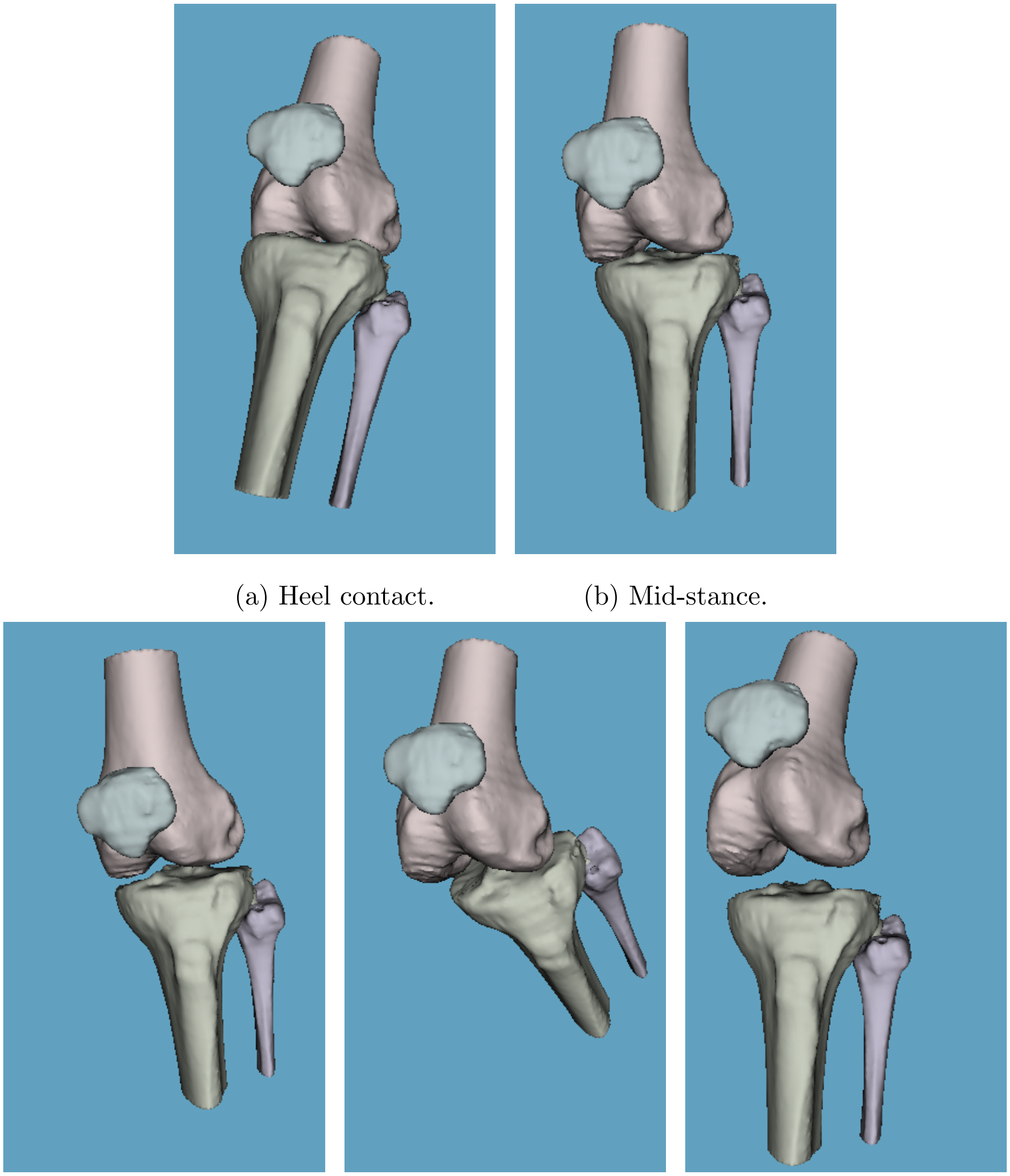


Figure 5.24: Trajectory tracking with five referenced events.



(a) Heel contact.

(b) Mid-stance.

(c) Initial swing.

(d) Mid-swing.

(e) Terminal swing.

Figure 5.25: 3D model of knee in perspective considering different angular positions.

Figure 5.26 shows four gait cycles in both speeds and the transition between them.

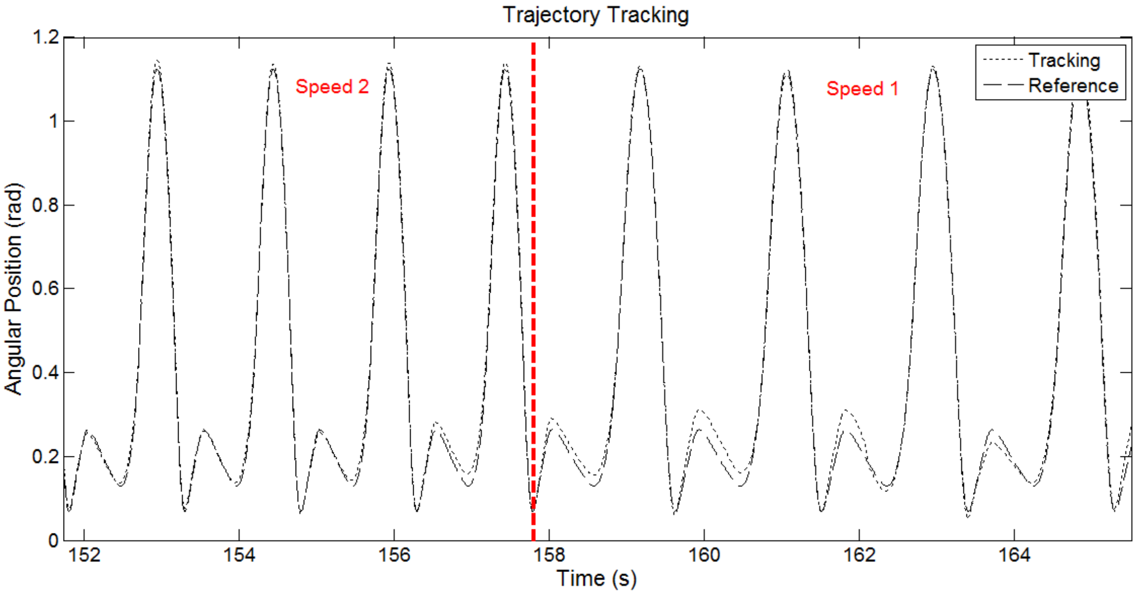


Figure 5.26: Trajectory tracking and transitions.

Chapter 6

Conclusions and Future Work

The main contribution of this research consists on proving that it is possible to integrate different models into a global process that uses simultaneously two measurements to perform the user intent in a virtual joint representation. Along this lines, it can be noted that most of the procedures or techniques used in this study were the simplest. Hence, further improvement can be achieved.

The study of the kinematic and dynamic models of the tibiofemoral joint was thought to be interpreted as just one model. However, under further investigation it was noted that under the different phases of walking the joint behaves differently; thus two sets of models were studied. These were known as stance and swing model. Each model was considered in a simplistic manner, involving just two links for the joint. This allowed to have the tibiofemoral joint as a single actuation point, not requiring other complex knee movements. To achieve a more realistic interpretation of the human body, a more complex model of the tibiofemoral joint could be implemented. The complex model will grant more degrees of freedom and possible movement in all three anatomical planes.

After the models were specified, each had to be controlled to perform the angular trajectory of the joint during the gait cycle. For this a PD+ controller was used, since is the most simplest control scheme for trajectory tracking. Using more powerful control techniques, which could give robust results, can be limited in practical use due to the high complexity and computational work. Usually dynamic control is focused on the trajectory; however, there exist different types of controllers focused on the forces, torques, or impedances of the model. Moreover, it came in consideration that a trajectory controller was not entirely required for the swing phase. This can be substituted by a position controller, since it just consist on moving the limb to the front rather than moving all the superior part of the body.

Knowing that there exist repetitive conditions in the gait cycle, a finite state machine was used alongside a conditional logical operator to perform the mid-level control, allowing then to make transitions between the controlled models. This method allowed a very fast and easy transitioning between models. However, the trajectory transitions could be controlled in a more advance manner. Also the finite state machine could provide assistance in assigning specific sets of trajectories for each sub-phase of the gait cycle.

The high-level control consists also on a conditional logical operator, but it greatly depends on the EEG classification, done by the SVM. In a general way, the SVM is a linear classifier, but with a low increase in the classifier's complexity it can perform a non-linear classification. There exists other methods for classifying EEG signals, like random forest, linear or quadratic discriminant analysis, or also artificial neural networks. Some of the mentioned methods have good performance; however, those that use a high complexity tend to be unstable. The SVM, although considered a slow classifier, is chosen for its simplicity, versatility, and robustness. Note however that the other types of classifiers work by separating a pair or several activities. In [45] decoding the kinematics of walking directly from the EEG recording was proposed. This allows a more complete user intent into the execution of the system, rather than just task classification.

The feature vectors, required by the SVM for the classification, were extracted from the EEG signal with the FFT. Although this method behaves rapidly and grants the power spectra of the signal, there are also several ways to obtain a feature vector. Methods like log variance, WPD, ApEn, and DWT. These methods give another type of features, like time-frequency features, variance, autoregressive, or adaptive autoregressive parameters.

The EMG-Torque relation grants a user interaction with the system. Before the relation takes place, the EMG signal also needed to be treated into a more simpler signal known as the linear envelope. The methods to acquire this signal are the MAV, RMS, and the usage of a low-pass filter. When processing the signal offline any of these methods can be used. However, their computational time required for calculations needs to be taken in consideration when realizing an online processing. Moreover, the processing of the signal can be handled in a more efficient way. In [40], for example, they used an advance EMG amplitude estimation with six stages. This procedure gives a signal with better characteristics than the simpler method here applied.

The relation, between the EMG signal and the torque required by the models to move in the desired trajectories, was done using a NNMS. The NNMS has different parameters to take in consideration, like the number of neurons, the sensibility control and the margin of sensibility. The chosen parameters gave a very fair relation between the processed EMG signal and the required torque. However, there exists a variety of linear and non-linear models that can be used, also depending on what type of physical activity to be studied, can be either static or dynamic.

The system of this thesis operated completely offline. Adapting the hBCI to operate online will be desirable for real time applications. Also it is necessary that the online process could be performed in a complete wireless manner. This way it can be applied to ambulatory conditions in different environments.

Appendix A

Equation Development

Lagrangian equation for Stance Phase:

$$\begin{aligned}
L_{St}(q,\dot{q}) =& K_{St}(q,\dot{q}) - U_{St}(q) \\
=& \frac{1}{2}m_2 l_{n2}^2\dot{\phi}^2 + \frac{1}{2}I_2\dot{\phi}^2 + \frac{1}{2}m_1 l_2^2\dot{\phi}^2 + m_1 l_2 l_{n1}\cos(q_2)\dot{\phi}^2 \\
&+ m_1 l_2 l_{n1}\cos(q_2)\dot{\phi}\dot{q}_2 + \frac{1}{2}m_1 l_{n1}^2\dot{\phi}^2 + m_1 l_{n2}^2\dot{\phi}\dot{q}_2 + \frac{1}{2}m_1 l_{n1}^2\dot{q}_2^2 \\
&+ \frac{1}{2}I_1\dot{\phi}^2 + I_1\dot{\phi}\dot{q}_2 + \frac{1}{2}I_2\dot{q}_2^2 - m_2 g l_{n2}\cos(\phi) \\
&- m_1 g l_2\cos(\phi) - m_1 g l_{n1}\cos(\phi + q_2)
\end{aligned} \tag{A.1}$$

Using the movement equations of Euler-Lagrange:

$$\begin{aligned}
\tau_{St1} =& \frac{d}{dt}\left[\frac{\partial L_{St}(q,\dot{q})}{\partial\dot{\phi}}\right] - \frac{\partial L_{St}(q,\dot{q})}{\partial\phi} \\
\frac{\partial L_{St}(q,\dot{q})}{\partial\dot{\phi}} =& m_2 l_{n2}^2\dot{\phi} + I_2\dot{\phi} + m_1 l_2^2\dot{\phi} + 2m_1 l_2 l_{n1}\cos(q_2)\dot{\phi} \\
&+ m_1 l_2 l_{n1}\cos(q_2)\dot{q}_2 + m_1 l_{n1}^2\dot{\phi} + m_1 l_{n1}^2\dot{q}_2 + I_1\dot{\phi} + I_1\dot{q}_2 \\
\frac{d}{dt}\left[\frac{\partial L_{St}(q,\dot{q})}{\partial\dot{\phi}}\right] =& m_2 l_{n2}^2\ddot{\phi} + I_2\ddot{\phi} + m_1 l_2^2\ddot{\phi} + 2m_1 l_2 l_{n1}\cos(q_2)\ddot{\phi} \\
&- m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}\dot{q}_2 + m_1 l_2 l_{n1}\cos(q_2)\ddot{q}_2 \\
&- m_1 l_2 l_{n1}\sin(q_2)\dot{q}_2^2 + m_1 l_{n1}^2\ddot{\phi} + m_1 l_{n1}^2\ddot{q}_2 + I_1\ddot{\phi} + I_1\ddot{q}_2 \\
\frac{\partial L_{St}(q,\dot{q})}{\partial\dot{\phi}} =& m_2 g l_{n2}\sin(\phi) + m_1 g l_2\sin(\phi) + m_1 g l_{n1}\sin(\phi + q_2)
\end{aligned} \tag{A.2}$$

$$
\begin{aligned}
\tau_{St2} =& \frac{d}{dt}\left[\frac{\partial L_{St}(q,\dot{q})}{\partial \dot{q}_2}\right] - \frac{\partial L_{St}(q,\dot{q})}{\partial q_2} \\
\frac{\partial L_{St}(q,\dot{q})}{\partial \dot{q}_2} =& m_1 l_2 l_{n1}\cos(q_2)\dot{\phi} + m_1 l_{n1}^2\dot{\phi} + m_1 l_{n1}^2\dot{q}_2 + I_1\dot{\phi} + I_1\dot{q}_2 \\
\frac{d}{dt}\left[\frac{\partial L_{St}(q,\dot{q})}{\partial \dot{q}_2}\right] =& m_1 l_2 l_{n1}\cos(q_2)\ddot{\phi} - m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}\dot{q}_2 \\
&+ m_1 l_{n1}^2\ddot{\phi} + m_1 l_{n1}^2\ddot{q}_2 + I_1\ddot{\phi} + I_1\ddot{q}_2 \\
\frac{\partial L_{St}(q,\dot{q})}{\partial q_2} =& - m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}^2 - m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}\dot{q}_2 \\
&+ m_1 g l_{n1}\sin(\phi + q_2)
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
\tau_{St1} &= \left[m_2 l_{n2}^2 + m_1 l_2^2 + 2m_1 l_2 l_{n1}\cos(q_2) + m_1 l_{n1}^2 + I_1 + I_2\right]\ddot{\phi} \\
&+ \left[m_1 l_2 l_{n1}\cos(q_2) + m_1 l_{n1}^2 + I_1\right]\ddot{q}_2 \\
&- 2m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}\dot{q}_2 - m_1 l_2 l_{n1}\sin(q_2)\dot{q}_2^2 \\
&- m_2 g l_{n2}\sin(\phi) - m_1 g l_2\sin(\phi) - m_1 g l_{n1}\sin(\phi + q_2) \\
\tau_{St2} &= \left[m_1 l_2 l_{n1}\cos(q_2) + m_1 l_{n1}^2 + I_1\right]\ddot{\phi} + \left[m_1 l_{n1}^2 + I_1\right]\ddot{q}_2 \\
&+ m_1 l_2 l_{n1}\sin(q_2)\dot{\phi}^2 - m_1 g l_{n1}\sin(\phi + q_2)
\end{aligned}
\tag{A.4}
$$

Lagrangian equation for Swing Phase:

$$\begin{aligned}
L_{Sw}(q,\dot{q}) =& K_{Sw}(q,\dot{q}) - U_{Sw}(q) \\
=& \frac{1}{2}m_1 l_{m1}^2 \dot{q}_1^2 + \frac{1}{2}I_1\dot{q}_1^2 + \frac{1}{2}m_2 l_1^2 \dot{q}_1^2 + \frac{1}{2}m_2 l_{m2}^2 \dot{q}_1^2 \\
&+ m_2 l_1 l_{m2}\cos(q_2)\dot{q}_1^2 + \frac{1}{2}m_2 l_{m2}^2 \dot{q}_2^2 - m_2 l_1 l_{m2}\cos(q_2)\dot{q}_1\dot{q}_2 \\
&- m_2 l_{m2}^2 \dot{q}_1\dot{q}_2 + \frac{1}{2}I_2\dot{q}_1^2 + \frac{1}{2}I_2\dot{q}_1\dot{q}_2 + \frac{1}{2}I_2\dot{q}_2^2 - m_1 g l_{m1} \\
&+ m_1 g l_{m1}\cos(q_1) - m_2 g l_1 - m_2 g l_{m2} \\
&+ m_2 g l_1 \cos(q_1) + m_2 g l_{m2}\cos(q_2 - q_1)
\end{aligned} \tag{A.5}$$

Using the movement equations of Euler-Lagrange:

$$\begin{aligned}
\tau_{Sw1} =& \frac{d}{dt}\left[\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_1}\right] - \frac{\partial L_{Sw}(q,\dot{q})}{\partial q_1} \\
\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_1} =& m_1 l_{m1}^2 \dot{q}_1 + I_1\dot{q}_1 + m_2 l_1^2 \dot{q}_1 + m_2 l_{m2}^2 \dot{q}_1 \\
&+ 2m_2 l_1 l_{m2}\cos(q_2)\dot{q}_1 - m_2 l_1 l_{m2}\cos(q_2)\dot{q}_2 \\
&- m_2 l_{m2}^2 \dot{q}_2 + I_2\dot{q}_1 + \frac{1}{2}I_2\dot{q}_2 \\
\frac{d}{dt}\left[\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_1}\right] =& m_1 l_{m1}^2 \ddot{q}_1 + I_1\ddot{q}_1 + m_2 l_1^2 \ddot{q}_1 + m_2 l_{m2}^2 \ddot{q}_1 \\
&+ 2m_2 l_1 l_{m2}\cos(q_2)\ddot{q}_1 - 2m_2 l_1 l_{m2}\sin(q_2)\dot{q}_1\dot{q}_2 \\
&- m_2 l_1 l_{m2}\cos(q_2)\ddot{q}_2 + m_2 l_1 l_{m2}\sin(q_2)\dot{q}_2^2 \\
&- m_2 l_{m2}^2 \ddot{q}_2 + I_2\ddot{q}_1 + \frac{1}{2}I_2\ddot{q}_2 \\
\frac{\partial L_{Sw}(q,\dot{q})}{\partial q_1} =& - m_1 g l_{m1}\sin(q_1) - m_2 g l_1 \sin(q_1) + m_2 g l_{m2}\sin(q_2 - q_1)
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
\tau_{Sw2} =& \frac{d}{dt}\left[\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_2}\right] - \frac{\partial L_{Sw}(q,\dot{q})}{\partial q_2} \\
\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_2} =& m_2 l_{m2}^2 \dot{q}_2 - m_2 l_1 l_{m2}\cos(q_2)\dot{q}_1 - m_2 l_{m2}^2 \dot{q}_1 + \frac{1}{2}I_2\dot{q}_1 + I_2\dot{q}_2 \\
\frac{d}{dt}\left[\frac{\partial L_{Sw}(q,\dot{q})}{\partial \dot{q}_2}\right] =& m_2 l_{m2}^2 \ddot{q}_2 - m_2 l_1 l_{m2}\cos(q_2)\ddot{q}_1 + m_2 l_1 l_{m2}\sin(q_2)\dot{q}_1\dot{q}_2 \\
&- m_2 l_{m2}^2 \ddot{q}_1 + \frac{1}{2}I_2\ddot{q}_1 + I_2\ddot{q}_2 \\
\frac{\partial L_{Sw}(q,\dot{q})}{\partial q_2} =& - m_2 l_1 l_{m2}\sin(q_2)\dot{q}_1^2 + m_2 l_1 l_{m2}\sin(q_2)\dot{q}_1\dot{q}_2 \\
&- m_2 g l_{m2}\sin(q_2 - q_1)
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
\tau_{Sw1} = &\left[m_1 l_{m1}^2 + m_2 l_1 2 + m_2 l_{m2}^2 + 2 m_2 l_1 l_{m2} \cos(q_2) + I_1 + I_2\right] \ddot{q}_1 \\
&+ \left[\frac{1}{2} I_2 - m_2 l_1 l_{m2} \cos(q_2) - m_2 l_{m2}^2\right] \ddot{q}_2 \\
&- 2 m_2 l_1 l_{m2} \sin(q_2) \dot{q}_1 \dot{q}_2 + m_2 l_1 l_{m2} \sin(q_2) \dot{q}_2^2 \\
&+ m_1 g l_{m1} \sin(q_1) + m_2 g l_1 \sin(q_1) - m_2 g l_{m2} \sin(q_2 - q_1) \qquad \text{(A.8)} \\
\tau_{Sw2} = &\left[\frac{1}{2} I_2 - m_2 l_1 l_{m2} \cos(q_2) - m_2 l_{m2}^2\right] \ddot{q}_1 \\
&+ \left[m_2 l_{m2}^2 + I_2\right] \ddot{q}_2 \\
&+ m_2 l_1 l_{m2} \sin(q_2) \dot{q}_2^2 + m_2 g l_{m2} \sin(q_2 - q_1)
\end{aligned}$$

Appendix B

Simulink Blocks

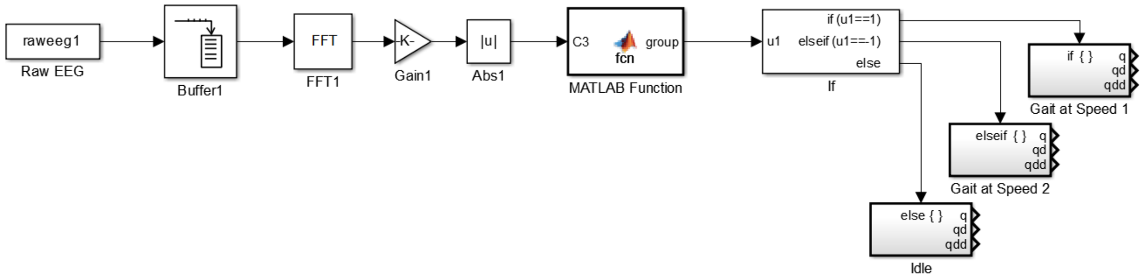
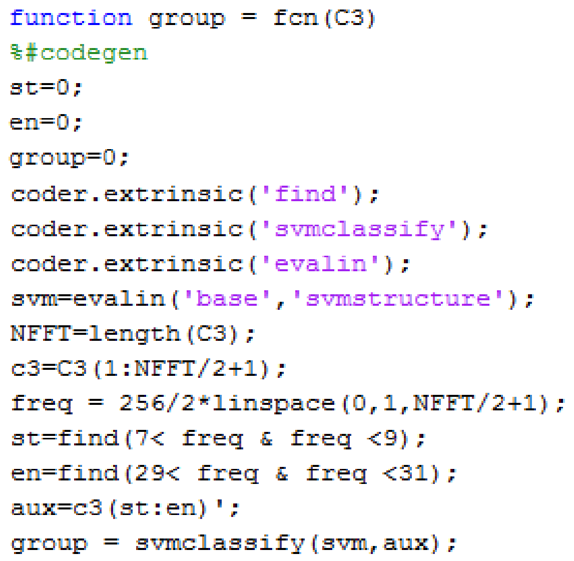


Figure B.1: Simulink Blocks for EEG processing.



```
function group = fcn(C3)
%#codegen
st=0;
en=0;
group=0;
coder.extrinsic('find');
coder.extrinsic('svmclassify');
coder.extrinsic('evalin');
svm=evalin('base','svmstructure');
NFFT=length(C3);
c3=C3(1:NFFT/2+1);
freq = 256/2*linspace(0,1,NFFT/2+1);
st=find(7< freq & freq <9);
en=find(29< freq & freq <31);
aux=c3(st:en)';
group = svmclassify(svm,aux);
```

Figure B.2: Code in MATLAB Function block for extracting the frequency bands and applying the SVM classification.

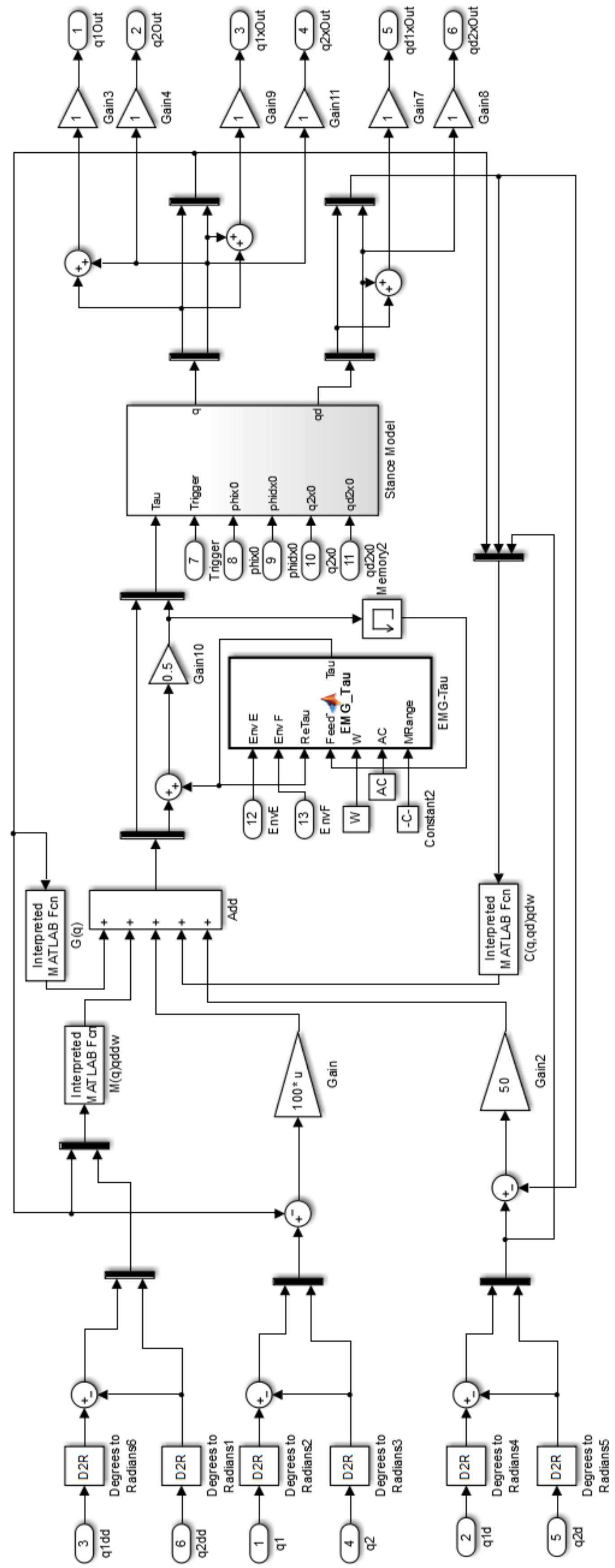


Figure B.3: Stance phase closed-loop system diagram with Simulink Blocks.

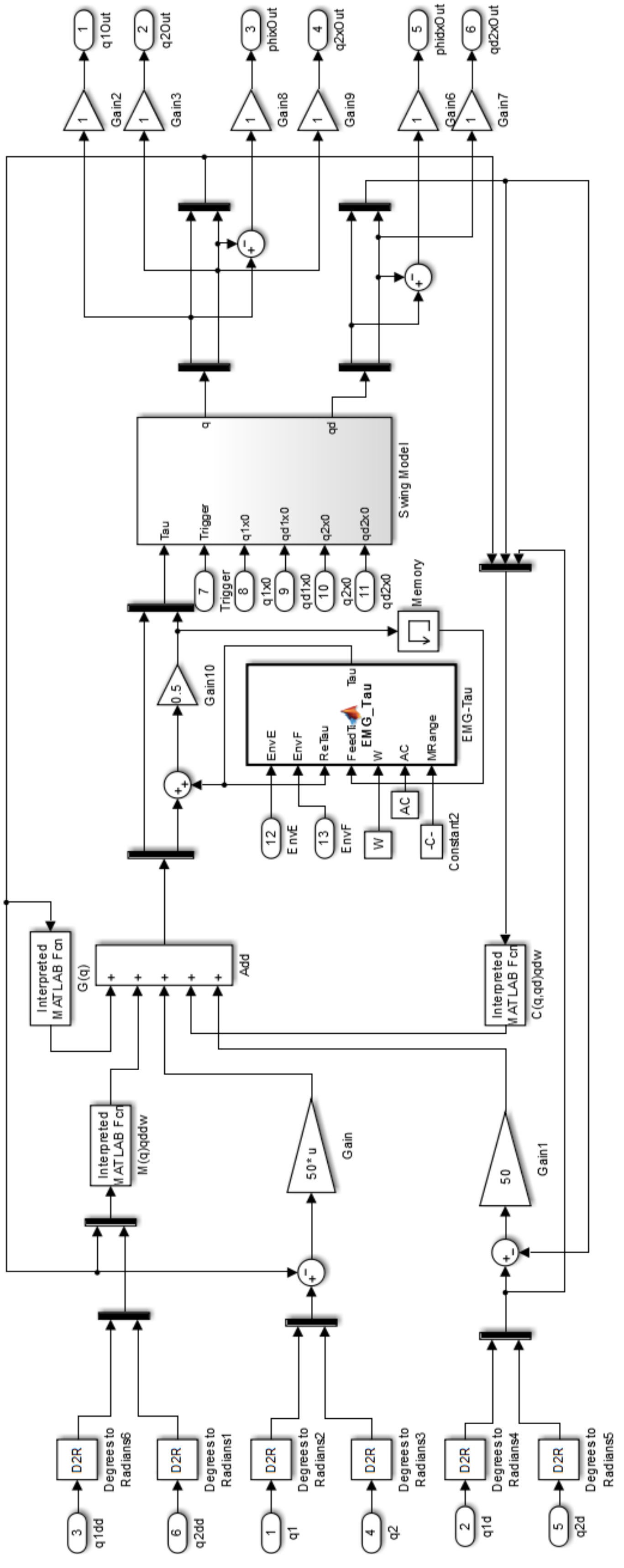


Figure B.4: Swing phase closed-loop system diagram with Simulink Blocks.

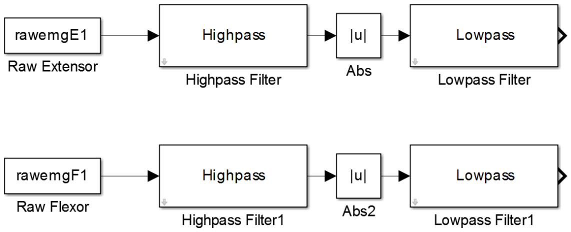


Figure B.5: High-pass filter, rectification and Low-pass filter applied to the Extensor and Flexor signals.

Learning Pseudo-Code

```
LearningFunctionNNMS(X,Ye,W,AC,sm,Lambda,U)
  Maximum Value and Position in SN;
  If Maximum Value is >= sm
    for pr=1, Total of inputs
      W(Position,pr) = (W(Position,pr)+X(pr))/2;
    end
    for o=1, Total of outputs
      AC(Position,o)=(AC(Position,o)+yo(o))/2;
    end
  else
    if (CN+1) > Maximum value of neurons avaliable to use
    Minimum Value and Position of the U
    for pr=1, Total inputs
      (Position, pr) = X(pr)
    end
    for o=1, Total of outputs
      AC(Position,o)=yo(o);
    end
  elseif
    New neuron = Number of neurons actualy used + 1;
    for pr=1, Total inputs
```

```
      W(NewNeuron, pr) = X(pr)
    end
    for o=1, Total of outputs
      AC(NewNeuron,o)=yo(o);
    end
  end
end
```

Application Pseudo-Code

```
ApplicationFunctionNNMS(X,W,AC,sm)
  for n=1,Number of Neurons
    S=0;
    for i=1,Total of inputs
      S=S+(x(i)-W(n,i)^2;
    end
    //fa = activation function
    SN(n)=fa(sqrt(S),alpha);
  end
  Maximum Value and Position in SN;
  if MaxValue > Maximum Sensibility Value
    for o=1, Total of Outputs
      y(o)=AC(pos,o)*SN(pos)
    end
  else
    for o=1, Total of outputs
      s1=0;
      s2=0;
      for n=1, CN
        s1=SN(n)*AC(n,o)+s1;
        s2=SN(n)+s2;
      end
      y(o)=s1/s2;
    end
  end
Output Value of the NNMS = y
```

Parameter	Description
α	Activation function parameter
sm	Maximum sensibility value

Table B.1: Parameters of the neural network.

Variables	Description
X	Input Pattern
Ye	Output desired to learn
W	Weights of first layer
SN	Second layer output
AC	Output value

Table B.2: Variables of the neural network.

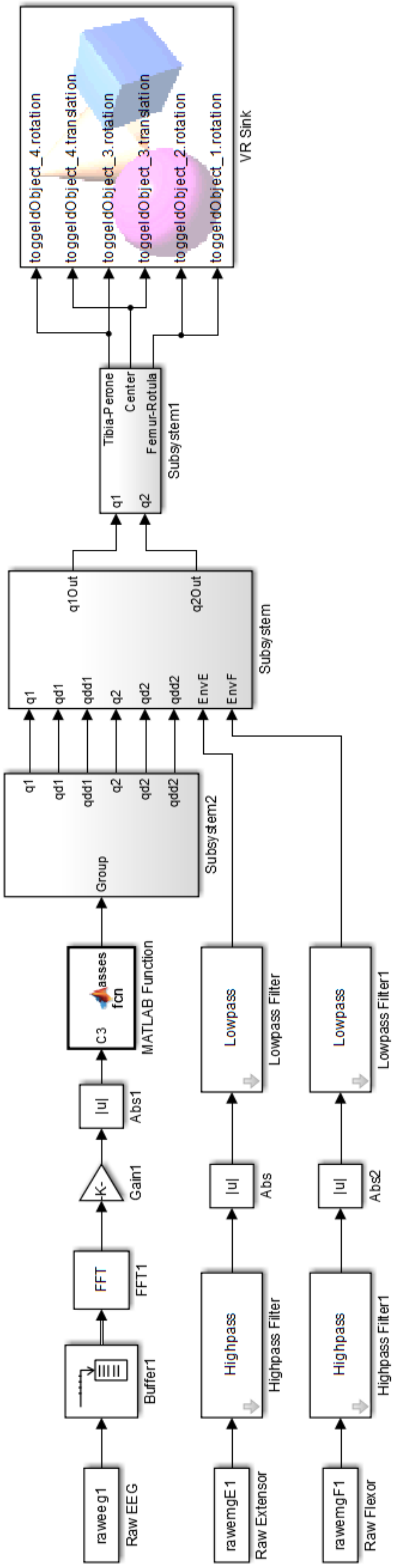


Figure B.6: hBCI system.

Bibliography

[1] Benjamin K. Potter and Charles R. Scoville. Amputation is not isolated: An overview of the us army amputee patient care program and associated amputee injuries. *Journal of the American Academy of Orthopaedic Surgeons*, 14:S118–S190, 2006.

[2] He Huang, Todd A. Kuiken, and Robert D. Lipschutz. A strategy for identifying locomotion modes using surface electromyography. *IEEE Transactions on Biomedical Engineering*, 56:65–73, 2009.

[3] D.M. Durand, H.J. Park, and B. Wodlinger. Localization and control of activity in peripheral nerves. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 3352–3354, 2008.

[4] Leigh R. Hochberg, Mijail D. Serruya, Gerhard M. Friehs, Jon A. Mukand, Maryam Saleh, Abraham H. Caplan, Almut Branner, David Chen, Richard D. Penn, and John P. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, 2006.

[5] Eric C. Leuthardt, Kai J. Miller, Gerwin Schalk, Rajesh P. N. Rao, and Jeffrey G. Ojemann. Electrocorticography-based brain computer interface-the seattle experience. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14:194–198, 2006.

[6] Fan Zhang, Will DiSanto, Jin Ren, Zhi Dou, Qing Yang, and He Huang. A novel cps system for ecaluating a neural-machine interface for artificial legs. In

IEEE/ACM Second International Conference on Cyber-Physical Systems, pages 67–76, 2011.

[7] Xiaorong Zhang, Yuhong Liu, Fan Zhang, Jin Ren, Yan Lindsay Sun, Qing Yang, and He Huang. On design and implementation of neural-machine interface for artificial legs. *IEEE Transactions on Industrial Informatics*, 8:418–429, 2012.

[8] An H. Do, Po T. Wang, Christine E. King, Sophia N. Chun, and Zoran Nenadic. Brain-computer interface controlled robotic gait orthosis. *Journal of NeuroEngineering and Rehabilitation*, 10:1–9, 2013.

[9] Christine E. King, Po T. Wang, Luis A. Chui, An H. Do, and Zoran Nenadic. Operation of a brain-computer interface walking simulator for individuals with spinal cord injury. *Journal of NeuroEngineering and Rehabilitation*, 10:1–14, 2013.

[10] Joonbum Bae and Masayoshi Tomizuka. A gait rehabilitation strategy inspired by an iterative learning algorithm. *Mechatronics*, 22:213–221, 2012.

[11] Huseyin Atakan Varol, Frank Sup, and Michael Goldfarb. Real-time gait mode intent recognition of a powered knee and ankle prosthesis for standing and walking. In *Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 66–72, 2008.

[12] Huseyin Atakan Varol, Frank Sup, and Michael Goldfarb. Powered sit-to-stand and assistive stand-to-sit framework for a powered transfemoral prosthesis. In *Proceedings of the 2009 IEEE 11th International Conference on Rehabilitation Robotics*, pages 645–651, 2009.

[13] Kevin H. Ha, Huseyin Atakan Varol, and Michael Goldfarb. Volitional control of a prosthetic knee using surface electromyography. *IEEE Transactions on Biomedical Engineering*, 58:144–151, 2011.

[14] Khalil Ullah, Asif Khan, Ihtesham ul Islam, and Mohammad A. U. Khan. Electromyographic (emg) signal to joint torque processing and effect of various fac-

tors on emg to torque model. *Journal of Engineering and Technology Research*, 3:330–341, 2011.

[15] Dennis J. McFarland, A. Todd Lefkowicz, and Jonathan R. Wolpaw. Design and operation of an eeg-based brain-computer interface with digital signal processing technology. *Behavior Research Methods, Instruments, & Computers*, 29:337–345, 1997.

[16] Nai-Jen Huan and Ramaswamy Palaniappan. Neural network classication of autoregressive features from electroencephalogram signals for braincomputer interface design. *Journal of Neural Engineering*, 1:142–150, 2004.

[17] Elly Gysels and Philippe Renevey Patrick Celka. Svm-based recursive feature elimination to compare phase synchronization computed from broadband and narrowband eeg signals in braincomputer interfaces. *Signal Processing*, 85:2178–2189, 2005.

[18] Roozbeh Borjian. Design, modeling, and control of an active prosthetic knee. Master's thesis, University of Waterloo, Waterloo, Ontario, Canada, 2008.

[19] Margareta Nordin and Victor H. Frankel. *Basic biomechanics of the musculoskeletal system*. Lippincott Williams and Wilkins, 3 edition, 2001.

[20] Jessica Rose and James G. Gamble. *Human Walking*. Philadelphia Lippincott Williams & Wilkins, 3 edition, 2006.

[21] SEER Training Modules. Anatomical terminology, Feb 2014. http://training.seer.cancer.gov/anatomy/body/terminology.html.

[22] Margareta Nordin and Victor H. Frankel. *Basic biomechanics of the musculoskeletal system*. Lippincott Williams and Wilkins, 4 edition, 2012.

[23] Fernando Reyes Cortés. *Robótica - Control De Robots Manipuladores*. Alfaomega, 1 edition, 2012.

[24] Roger M. Edoka. *Neuromechanics of Human Movement.* Human Kinetics, 4 edition, 2008.

[25] VAKHUM Project. 3d tibiofemoral joint, Feb 2014. http://www.ulb.ac.be/project/vakhum/index.html.

[26] Morikazy Takegaki and Suguru Arimoto. A new feedback method for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103:119–125, 1981.

[27] Brad Paden and Ravi Panja. Globally asymptotically stable pd+ controller for robot manipulators. *International Journal of Control*, 47:1697–1712, 1988.

[28] Rune Schlanbusch, Esten Grøtli, Antonio Loría, and Per Johan Nicklasson. Hybrid attitude tracking of output feedback controlled rigid bodies. In *2011 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 5479–5484, 2011.

[29] Ludan Wang, Sheng Cheng, Haojun Guan, and Jianwei Zhang. Design, modeling and control of a line-walking robot for inspection of power transmission lines. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1990–1995, 2009.

[30] Samuel Au, Max Berniker, and Hugh Herr. Powered ankle-foot prosthesis to assist level-ground and stair-descent gaits. *Neural Networks*, 21:654–666, 2008.

[31] Hamzah S. AlZu'bi. Preprocessing and feature extraction for asynchronous multi-class noninvasive brain computer interface based on eeg signal. Master's thesis, Masdar Institute of Science and Technology, 2011.

[32] J. d. R. Millán, R. Rupp, G. R. Müller-Putz, R. Murray-Smith, C. Giugliemma, M. Tangermann, C. Vidaurre, F. Cincotti, A. Kübler, R. Leeb, C. Neuper, K.-R. Müller, and D. Mattia. Combining braincomputer interfaces and assistive technologies: state-of-the-art and challenges. *Frontiers in Neuroscience*, 4:1–15, 2010.

[33] Valer Jurcak, Daisuke Tsuzuki, and Ippeita Dan. 10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems. *NeuroImage*, 34:1600–1611, 2007.

[34] Saeid Sanei and J.A. Chambers. *EEG Signal Processing*. John Wiley & Sons Ltd.

[35] Chunmei Wang, Junzhong Zou, Jian Zhang, Min Wang, and Rubin Wang. Feature extraction and recognition of epileptiform activity in eeg by combining pca with apen. *Cognitive Neurodynamics*, 4:233–240, 2010.

[36] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 2 edition, 1998.

[37] Vladimir N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10:988–999, 1999.

[38] Peter Konrad. The abc of emg, 2005.

[39] Carlo J. De Luca. Practicum on the use of surface emg signals in movement sciences. Technical report, Delsys Inc, 2008.

[40] Edward A. Clancy, Stéphane Bouchard, and Denis Rancourt. Estimation and application of electromyogram (emg) amplitude during dynamic contractions. *IEEE Engineering in Medicine and Biology Magazine*, 20:47–54, 2011.

[41] Chanderpal Sharma, Manoj Duhan, and Dinesh Bhatia. Filter optimization of emg signal using matlab.

[42] Gustavo González Sanmiguel, Luis Lauro Gonzalez, Luis M. Torres-Trevi no, and César Guerra. On-line learning in an embedded maximum sensibility neural network. In *2012 11th Mexican International Conference on Artificial Intelligence (MICAI)*, pages 75–79, 2012.

[43] BIOPAC Systems, INC. *B-Alert X10 Setup Manual*.

[44] BIOPAC Systems, INC. *BSL Harware Guide*, december 2013.

[45] Alessandro Presacco, Larry W. Forrester, and Jose L. Contreras-Vidal. Decoding intra-limb and inter-limb kinematics during treadmill walking from scalp electroencephalographic (eeg) signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20:212–219, 2012.